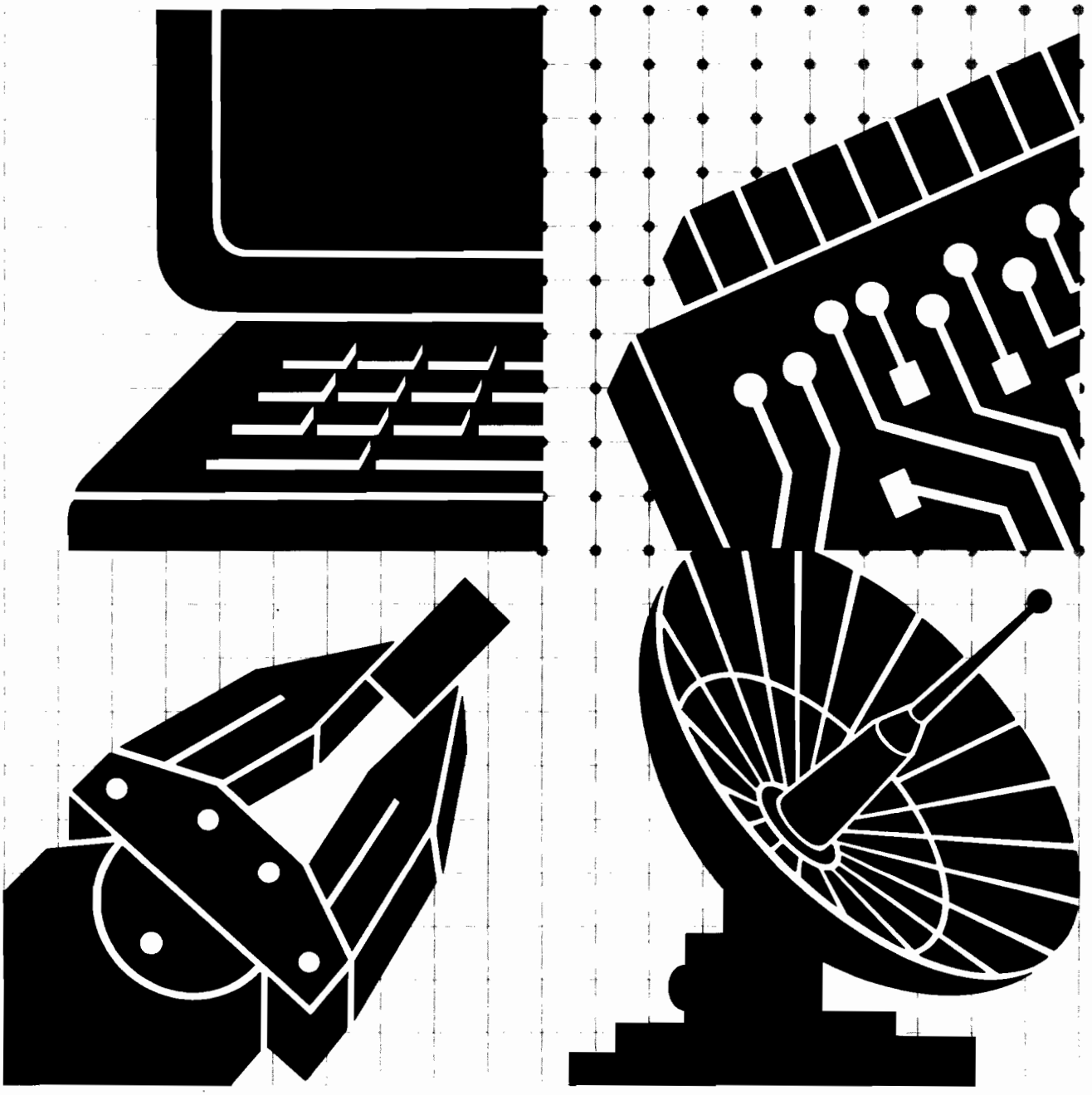
FAIRCHILD

A Schlumberger Company

FAST™ Applications Handbook

FAIRCHILD
A Schlumberger Company

FAST™ Applications Handbook



©1987 Fairchild Semiconductor Corporation, Digital Unit
333 Western Avenue, South Portland, Maine 04106
207/775-8700 TWX 710-221-1980

FAST (Fairchild Advanced Schottky TTL) is a registered trademark of Fairchild Semiconductor Corporation.

FAIRCHILD
A Schlumberger Company

Foreword

The skills required of a digital logic system designer have changed dramatically over the past decade. System clock speeds have increased, LSI/VLSI ICs have become commonplace, and end product quality/reliability have become key design goals.

FAST TTL has become the logic family of choice in these demanding new designs. This handbook was written to assist the engineer in designing in high-speed logic environments.

The Introduction which follows describes FAST from a technology standpoint. Comparisons are made to earlier logic families to identify how FAST is similar to these familiar products and yet offers superior performance and quality to the system designer.

Sections 2 through 8 describe the available SSI/MSI functions. Various, and at times novel, applications are covered. An experienced designer may want to quickly review these chapters.

The remainder of the book is devoted to basic design guidelines for using the high-speed FAST TTL logic family. Section 9, Line Driving and System Design, discusses transmission line effects such as reflections and crosstalk. Techniques for minimizing adverse high frequency effects are covered in detail.

Section 10 covers Fairchild testing procedures. While this section is primarily targeted at quality assurance and incoming test engineers, it is also valuable to the designer. The relationship of how tested parameters can guarantee device functionality/reliability in a particular application is an important concept.

Packaging of integrated circuits is undergoing a dramatic evolution. It is essential that the often overlooked package characteristics be considered early in the design since they can have significant impact on system reliability and product life. Section 11 covers the key package characteristics and examines their importance relative to system design.

The final sections of this handbook contain a product cross reference and index to facilitate finding the proper FAST device for a particular application, and a complete listing of Fairchild Sales Offices and Authorized Distributors.

FAIRCHILD
A Schlumberger Company

Table of Contents

FAST®

FAIRCHILD
A Schlumberger Company

Introduction

FAST, an acronymn for Fairchild Advanced Schottky TTL, is a high-speed, low-power logic family that achieves speeds typically 30% faster than the Schottky family with a corresponding power reduction of approximately 75%. It is fabricated with Fairchild's Isoplanar II process, an advanced oxide isolation technique which produces transistors with very high, well controlled switching speeds, extremely small parasitic capacitances and f_t in excess of 5 GHz.

Since the family is designed to be pin compatible with other TTL families such as Schottky, Low Power Schottky, and standard TTL, existing designs can be easily upgraded. FAST offers significant improvements over the Schottky family in addition to improved speed and power specifications. Other key advantages are higher input threshold levels (improving noise margins), reduced input loading, and increased output drive. The FAST family contains a full complement of circuits for more efficient design capabilities: small scale integration, medium scale integration and large scale integration.

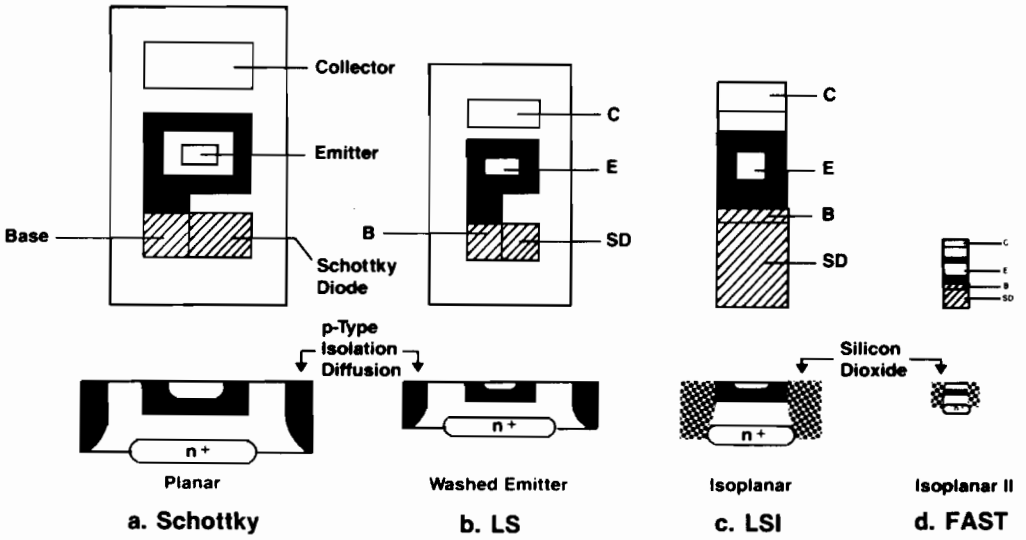
The development of FAST began in 1977 when Fairchild, working with a large military OEM, first applied LSI technology to the manufacture and design of a commodity-type logic family. The result of this work culminated in the introduction of the FAST family in September of 1978, when it was made known to the general public. In 1979, the first FAST samples were delivered to specific customers and development work continued at a very high rate. By 1980, over 1,000,000 devices had actually been shipped to customers and over 35 devices had been introduced. In 1981, the most significant event occurred: Signetics announced a second source agreement with Fairchild Semiconductor. This agreement covered 34 devices and insured a second source position for the FAST family. Many major OEMs have designs committed to the FAST family, either in redesigns or new designs. There are over 175 parts, either introduced or planned for introduction in the FAST family.

Fairchild engineers had some specific design objectives in mind when they developed the FAST logic family. The primary objective was the improvement of the circuit speed-power performance versus earlier TTL families. Another important objective was increasing threshold levels to improve DC noise immunity. Other goals were maintaining or improving the output drive of Schottky for improved line driving capability, and reducing input loading for increasing the overall fanout of the family. Output and input voltage levels, functions and pinouts were standardized to previous TTL families to maintain compatibility.

The primary design consideration was to improve speed while reducing power. The speed of any device is limited by the charge storage in the transistors. The time required to remove this charge is proportional to the capacitance and current available. Thus, to improve the speed, either the internal resistor values must be lowered to increase the available current and therefore remove the charge faster, such as in the 74H family and the Schottky family, or the capacitance must be reduced.

The parasitic capacitance associated with a typical bipolar transistor is proportional to junction area and diffusion depth. Figure 1-1 shows the evolution of the transistor's geometry used in bipolar circuit technology.

Figure 1-1



In Figure 1-1a, the planar Schottky transistor size was determined by the emitter contact alignment tolerance, requiring a larger emitter. The base area was, therefore, larger than the emitter for similar reasons. The base-to-P type isolation spacing is dictated by alignment tolerances, and lateral diffusion techniques determine the size of the collector area surrounding the base region.

In Figure 1-1b, the washed emitter transistor, the elimination of the need for the emitter contact mask was accomplished by etching (washing away) the oxide above the emitter. Thus the emitter needed to be no larger than the actual emitter contact which allowed a reduction in all transistor geometries, thereby reducing the parasitic capacitance.

In Figure 1-1c, this isoplanar process eliminated the P-type isolation diffusion. The lateral cell isolation was performed with silicon dioxide. This minimized the base-collector capacitance associated with that transistor.

In Isoplanar II, Figure 1-1d, further refinements in oxide isolation allowed the use of smaller emitter, base and collector areas thus significantly reducing the capacitances of all three junctions. In addition, the reduction in the diffusion depth allowed a reduction in the diffusion capacitances. The increase in f_t of the Isoplanar II transistor is significant over that of either the planar or the washed emitter. FAST uses internal transistors with f_t in the 5 GHz range. Low Power Schottky and Schottky transistors operate in the 1.6 GHz range.

The speed-power curve shown in Figure 1-2a was used empirically to determine the optimum operating power level for the FAST family. Several internal gates programmed at a variety of power levels were produced on a wafer and the propagation delay of an internal gate for each power level was measured.

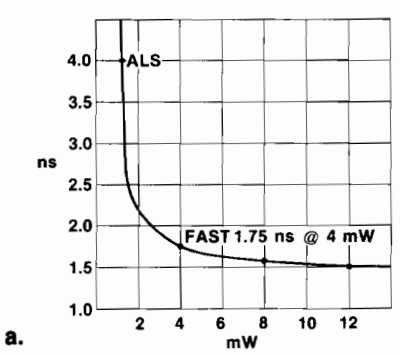
As can be seen readily from the curves, power levels significantly below 4 mW per gate exhibit a dramatic degradation in performance. Power levels significantly above 4 mW, however, appear to have passed the point of diminishing returns with only minor improvements in propagation delay resulting from increased power. It was therefore concluded that the FAST family would be biased at 4 mW. For internal gates, 4 mW achieves 1.75 ns propagation delay. It should be noted that the output stage with its large output transistors necessary to supply the output drive will cause degraded speed. Hence the advantages of FAST are exemplified by MSI and LSI with only minor improvements in SSI.

In Figure 1-2b, the FAST logic family is compared to previous TTL and ECL logic families. Each line groups families with similar technologies. The first line, known as "gold doped," groups together the 7400 and the 74H families into one technology grouping. These saturating logic families can be seen to have a relatively poor speed power curve.

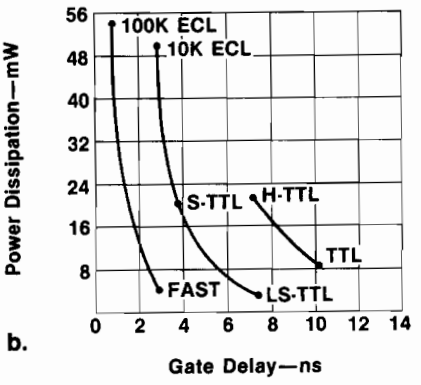
The second curve groups together the Schottky, Low Power Schottky and 10K ECL family. They use non-gold doped, soft saturated (Baker clamped) or current steering logic in order to achieve their speed power performance; however, they still employ the planar technology. The last curve, which shows the FAST family grouped together with its ECL counterpart, the 100K ECL family, employs the isoplanar technology. With FAST isoplanar technology, 3 ns propagation delays at only 4 mW power dissipation are achieved with SSI devices.

To this point we have described the technology and the reasons behind choosing the Isoplanar II technology. Additional improvements were also made in the actual circuitry used to produce the FAST logic family.

Figure 1-2 Speed-Power Product



a.



b.

In addition to the technological improvements over previous TTL families it was recognized that circuit improvements would also provide significant benefits. The schematic of a basic 2-input NAND gate shown in Figure 1-3 portrays the typical input and output circuitry used throughout the FAST family. From the input side to the output side of the circuit, the following description will detail the circuit improvements incorporated in the FAST family.

D5 and D6 are two large-geometry Schottky diodes providing high-speed clamping of input excursions below ground and above V_{CC}. This prevents unwanted activity by parasitic transistors always found in bipolar integrated circuits, and corrects a deficiency frequently observed in the LS family. Note that a similar clamp diode, D12, also appears on the output which limits reflections. The diodes D5, D6 and D12 damp out line reflections at both receiving and transmitting ends.

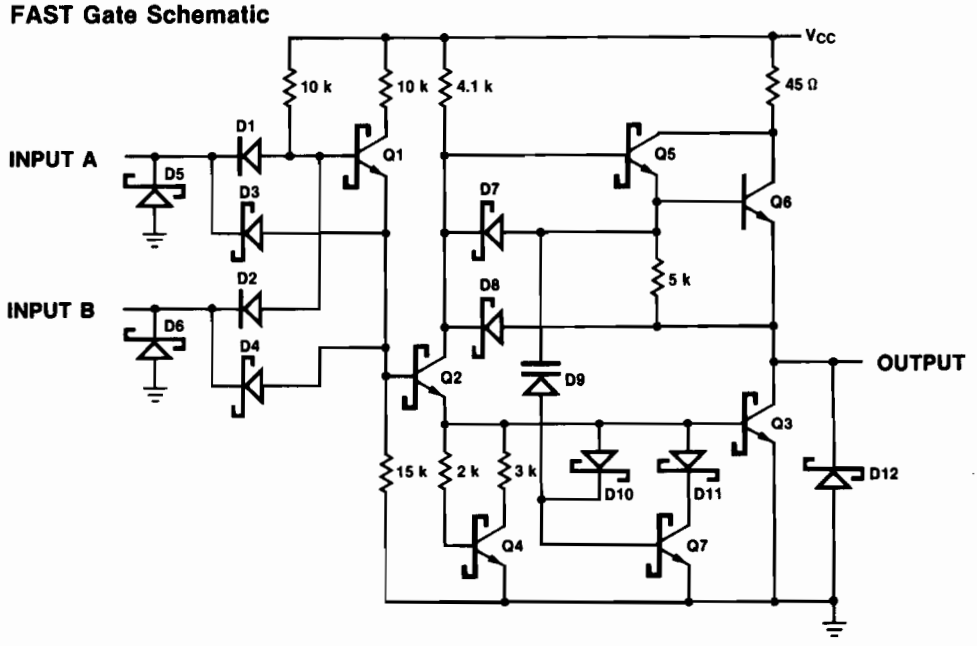
The gating function itself is accomplished with diodes D1 and D2, conventional silicon diodes, not Schottky diodes as in LS. Thus the front end is patterned after DTL instead of employing the multi-emitter transistors which gave TTL its name and were employed by most TTL families prior to LS. The advantage of the diode input structure is evidenced by the return to that structure by the later generation LS providing for improvements in speed and higher maximum input voltages. Low Power Schottky design, however, necessitated employing Schottky diodes of a large geometry to minimize the voltage drop across them, maximizing the input threshold voltage. LS had only two base emitter voltage drops following the gating diodes; therefore, since the voltage across the gating diodes was subtracted from the threshold, the diodes were made physically large to reduce the current density and the forward drops, increasing the threshold to a barely tolerable level. The attendant large junction capacity caused noise spikes to be capacitively coupled into the active circuitry of LS increasing its noise susceptibility.

Three base-emitter drops with three stages of current gain are employed in the FAST family. This not only increases the threshold to 1.5 volts but it also increases the circuit gain to yield a higher fanout than that of previous TTL families.

Diodes D3 and D4 are used to remove the stored charge from the base of Q2 on the negative transition of the input, speeding the positive transition of the output. D7 performs a similar function for Q6, improving the turn-off time of Q6 and speeding the negative transition of the output. Otherwise the removal of the stored charge is limited to the resistors between base and emitter of Q2 and Q6.

The role of D8 is one of increasing the current available at the collector of Q2 which provides base drive to Q3 under dynamic line driving or capacitive discharge conditions. It can be seen that the 4.1k collector resistor for Q2 would limit the current to approximately 1 mA to drive the base of Q3. If the output is HIGH, driving a low dynamic load impedance such as a transmission line or capacitive load, current will be conducted through D8 to the collector of Q2 and

Figure 1-3 Basic FAST Gate Schematic



provide additional base drive to Q3. This is a unique type of bootstrap circuit whereby the energy in the load itself is used as driving energy to discharge the load.

The squaring network consisting of Q4 and its two resistors provides a low dissipation circuit to square the output of the emitter of Q2. This circuit has been employed in previous logic families.

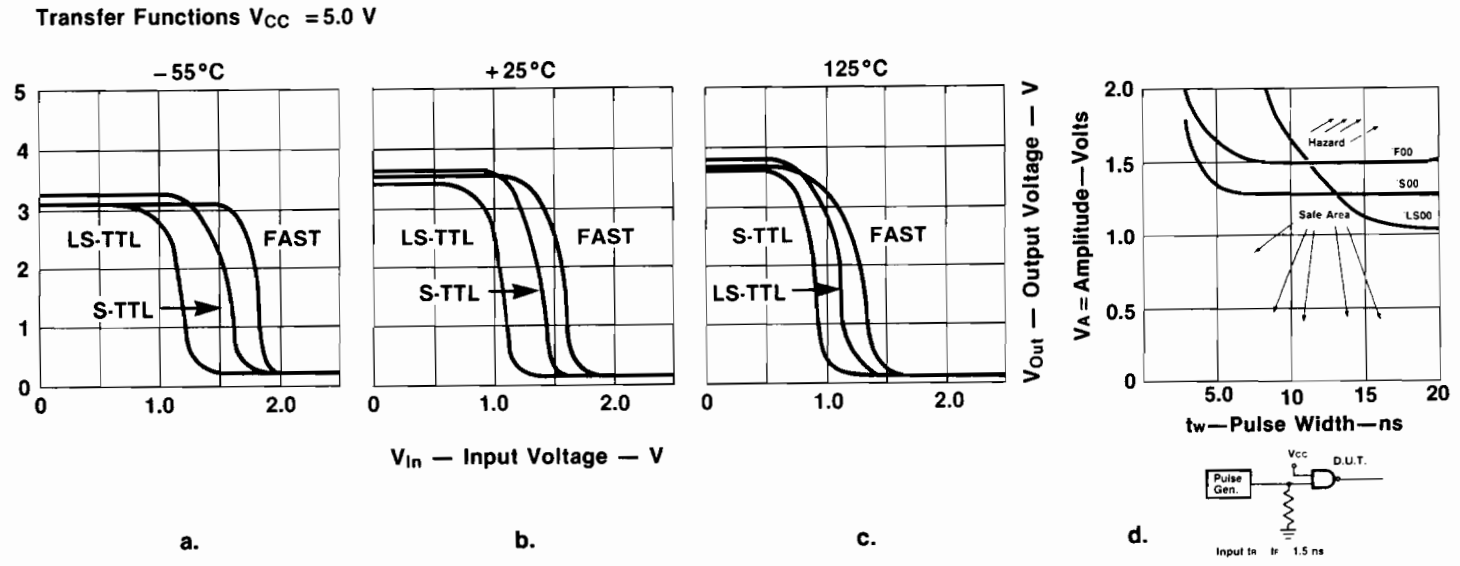
One final circuit which dramatically decreases the rise time of the output is the patented Dynamic Discharge Circuitry comprised of Q7 and D9, D10, and D11. Since the output transistors Q3 and Q6 are required to conduct high currents, they cannot be fabricated with the reduced geometry discussed earlier. Hence, Q3 has a rather large Miller capacitance between base and collector. During the positive transition of the output, current will be provided to the base of Q3 proportional to the output rate of rise and the capacitance of the output transistor. Due to the relatively high impedance of the squaring network, little of this current would be shunted to ground were it not for this dynamic discharge circuit. As the base of Q6 rises, it capacitively couples a current into the base of Q7 which will cause it to turn on and shunt the current provided by the Miller capacitance to ground, thus speeding the positive going transition, which is traditionally the slower in other TTL families. This reduction in skew between LOW-to-HIGH and HIGH-to-LOW transitions minimizes the pulse stretching and shrinking found in other TTL families, a strong point for FAST. This patented circuit is frequently called the "Miller Killer" by those who feel a kinship to the FAST family.

Shown on three transfer curves (Figures 1-4a, 1-4b, 1-4c) is the actual transfer level of FAST compared to Schottky and Low Power Schottky. FAST at 1.5 V has the highest room temperature threshold of any of the families, which compares with Schottky's 1.3 V and Low Power Schottky's 1.0 V. This difference is maintained throughout the military operating temperature range of −55°C to +125°C.

Not only was the DC noise immunity of the FAST logic family improved, but a corresponding increase in the AC noise immunity was also achieved.

Shown in Figure 1-4d is a plot of pulse amplitude versus pulse duration for a positive-going noise pulse that could occur on the input of a logic function. The resulting curve is that combination of pulse width and pulse amplitude which causes the output of the gate under test to drop to 2.0 V. Any combination of amplitude and duration below the line will be rejected by the gate. Any combination of amplitude and duration above the line is the hazard area, because noise could be propagated to the subsequent level of gate. This curve was developed by setting the pulse generator output duration to 20 ns and then increasing the amplitude until the output of the device under test falls to 2.0 V. This gives one data point on the curve for that device type. The input pulse duration is then reduced to 15 ns and the amplitude is ascertained which causes the device under test output to dip to 2.0 V. This gives a second data point.

Figure 1-4 Higher Threshold Levels For Improved Noise Immunity



By reducing the pulse width successively in stages and measuring the input amplitude for each width, enough data points can be obtained to draw a smooth curve that identifies the dividing line between safe and hazardous combinations of noise amplitude and noise duration. FAST exhibits about 200 mV more noise immunity than the Schottky family. The curve on the graph correlates well with the voltage transfer function for 25°C shown above. It shows that the FAST design objective of improved AC as well as DC noise immunity has been met.

Because the circuitry and technological improvements yield well-controlled AC parameters, the FAST family can be specified over extremes of external influences. FAST is the first TTL logic family which does not require derating estimates for worst case design. This has been accomplished by specifying minimum and maximum propagation delays over the operating temperature range and supply voltage with 50 pf loading.

In order to achieve easier correlation with our customers' needs, a change in the actual AC test load was needed. Previously, most TTL families were measured with three serial diodes in parallel with the load capacitor. For the FAST logic family, a 50 pf capacitance in parallel with a 500 Ω resistor is employed. This facilitates fabrication of low capacitance test jigs. It also provides better correlation with customers' measurements of propagation delay. Passive 500 Ω scope probes, which are less expensive and easier to use than the high impedance FET input scope probes, can be employed. This facilitates measurement of the AC performance on automatic test equipment and yields more conservative AC figures than are achieved with the previous AC load technique.

Figure 1-5 shows how load capacitance affects the propagation delay of Low Power Schottky, Schottky and FAST gates, flip-flops, registers and decoders, etc. As would be expected, Low Power Schottky TTL shows greater sensitivity since LS output drive capability is not as great as either Schottky or FAST. Significantly, FAST is less affected than Schottky by load capacity. Figure 1-5b shows propagation delay vs. load capacitance for buffers and line drivers since they are designed for greater output drive.

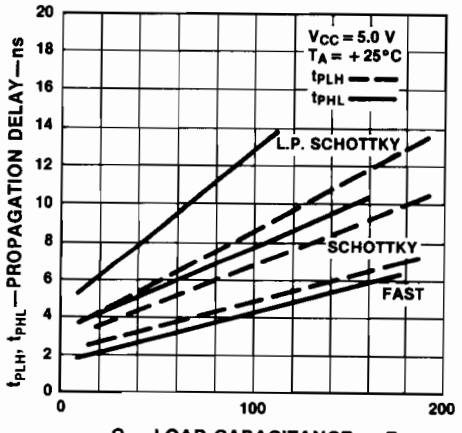
Notice also that for Schottky the HIGH-to-LOW output transition is more affected than its LOW-to-HIGH transition, while for FAST both transitions are equally affected. This shows better balance in the design of the FAST output, and minimizes pulse stretching and compressing.

Designers are cautioned that curves of this type do not apply when the load capacitance is distributed along an interconnection. For example, a 27-foot length of twisted pair wire, having 18 pf load per foot, will have a total capacitance of 500 pf. Such a twisted pair load will have little effect on the delay of a FAST output. The cable delay will be approximately 40 ns.

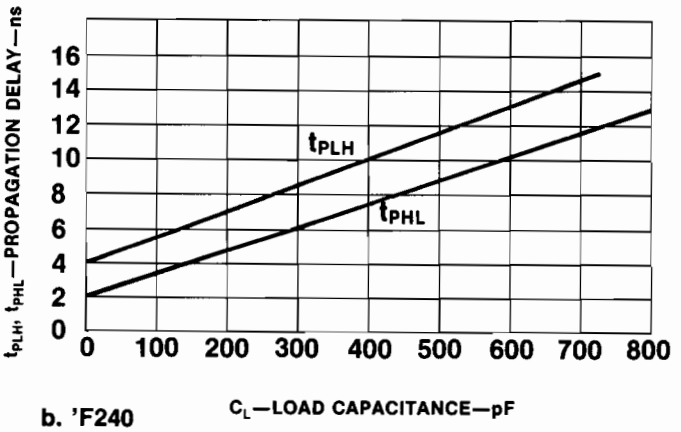
Reduced input loading to improve fanout was one of the additional design considerations for the FAST logic family. FAST and Low Power Schottky have diodes as the input gating elements, therefore, I_{IH} is specified at less than 20 μA. This is primarily a leakage specification. Schottky, employing multi-emitter transistors as the gating elements, has a higher I_{IH} limit.

The breakdown specifications, I_I for FAST and LS, have voltage breakdowns of 7 and 10 V respectively specified at 100μA. This allows direct connection of an input to the

Figure 1-5 Propagation Delay vs. Load Capacitance



a. 'F00



b. 'F240

supply rail for a Logic 1 implementation; Schottky's lower breakdown of 5.5 V requires a current limiting resistor if a Logic 1 supply rail is needed at an input.

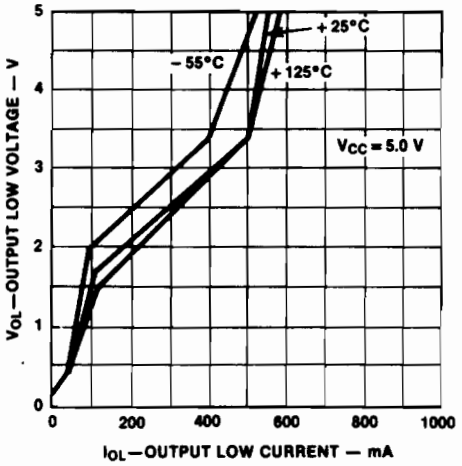
The input low current spec, I_{IL} for FAST, is one-third that of Schottky at 0.6 mA and very close to that of Low Power Schottky at 0.4 mA.

Corresponding to the improved input loading of the FAST logic family is an improvement of the output drive capability. FAST logic family standard outputs have an output sourcing current of 1.0 mA and an output sinking current of 20 mA. This corresponds to Schottky and is considerably higher than Low Power Schottky. This represents a fanout of 33.

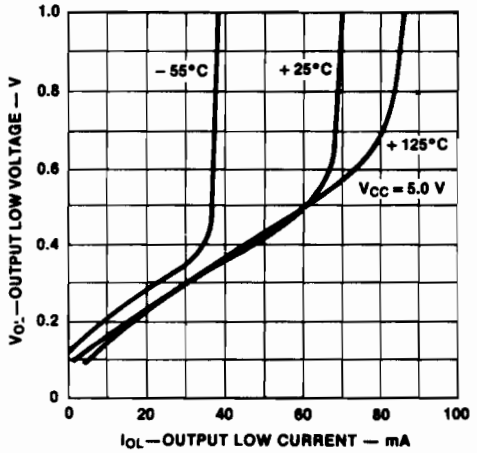
For buffer type outputs in the FAST logic family, such as the 54F/74F240, 54F/74F241 and 54F/74F244, the output drive capability is 64 mA in the LOW state and 15 mA in the HIGH state.

The output short circuit specifications for FAST have been improved over either Schottky or Low Power Schottky. The typical specification for standard outputs is 90 mA with a minimum and maximum specification of 60-150 mA. For buffer type outputs, the target specification is 150 mA with a minimum and maximum specification of 100-225 mA.

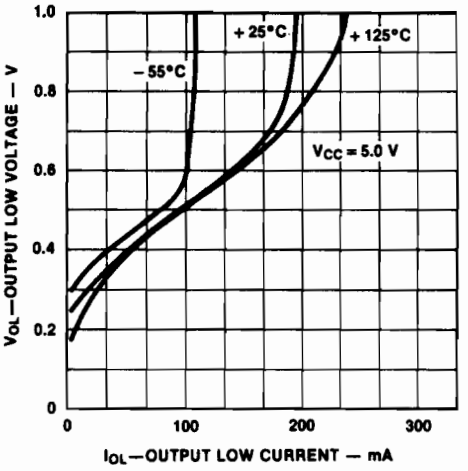
Figure 1-6



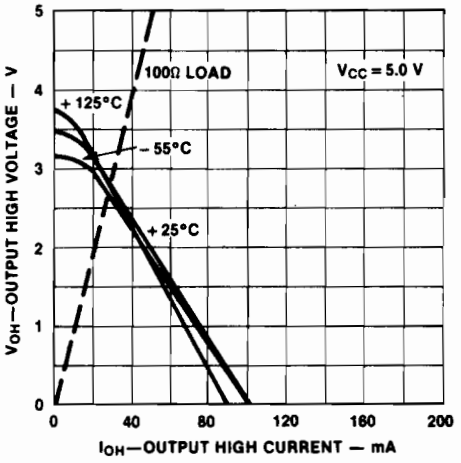
a. Output LOW Characteristics—'F00



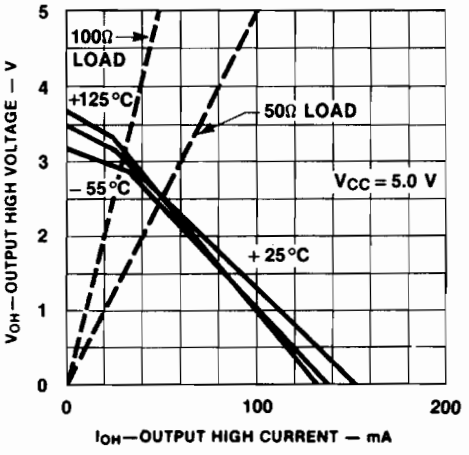
b. Output LOW Characteristics—'F00



c. Output LOW Characteristics—'F244



d. Output HIGH Characteristics—'F00



e. Output HIGH Characteristics—'F244

Figure 1-7 7410 Gate

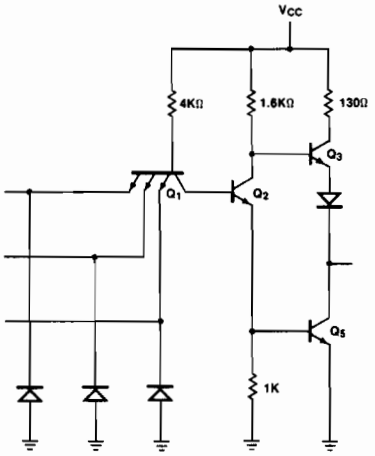


Figure 1-8 74S00 Gate

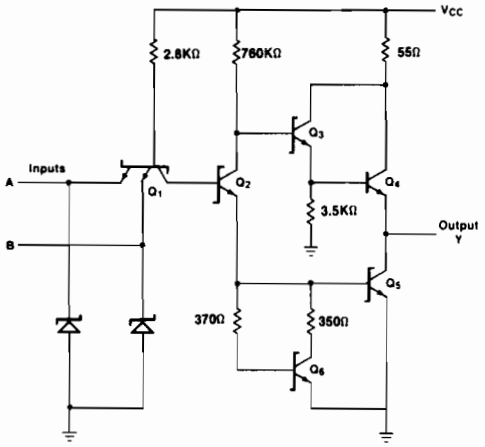
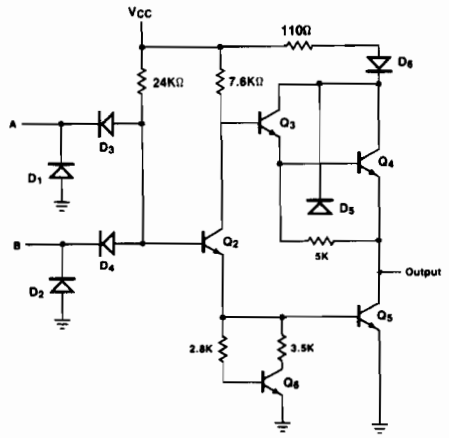


Figure 1-9 74LS00 Gate



Unused Inputs

Theoretically, an unconnected input assumes the HIGH logic level, but practically speaking it is in an undefined logic state because it tends to act as an antenna for noise. Only a few hundred millivolts of noise may cause the unconnected input to go to the logic LOW state. On devices with memory (flip-flops, latches, registers, counters), it is particularly important to terminate unused inputs (MR, PE, PL, CP) properly since a noise spike on these inputs might change the contents of the memory. It is poor design practice to leave unused inputs floating.

If the logic function calls for a LOW input, such as in NOR or OR gates, ground the unused inputs. For a permanent HIGH signal, unused inputs can be tied to V_{CC}. A current limiting resistor, in the range of 1kΩ to 5kΩ, is recommended for emitter-type inputs since these break down at some unspecified voltage above 5.5 V and power supply misadjustment or malfunction can cause damage unless the current is limited. Note that one resistor can serve several inputs, provided only that the cumulative I_{IH} current does not cause the voltage to drop below 2.4 V. Note also that diode-type FAST and LS-TTL inputs have breakdown voltages above 7.0 V and thus protective resistors are not normally required. An unused input may also be tied to a used input having the same logic function, such as NAND or AND gates, provided that the driver can handle the added I_{IH}. This practice is not recommended for diode-type inputs in a noisy environment, since each diode represents a small capacitor and two or more in parallel can act as an entry port for negative spikes superimposed on a HIGH level and cause momentary turn-off of Q2 (Figure 1-9).

Increasing Fanout

To increase fanout, inputs and outputs of gates on the same package may be paralleled. It is advisable to limit the gates being paralleled to those in a single package to avoid large transient supply currents due to different switching times of the gates. This is not detrimental to the devices, but could cause logic problems if the gates are being used as clock drivers.

Open-Collector Outputs

A number of available circuits have no pull-up circuit on the outputs. Open-collector outputs are used for interfacing or for wired-OR (actually wired-AND) functions. The latter is achieved by connecting open-collector outputs together and adding an external pull-up resistor.

The value of the pull-up resistor is determined by considering the fan-out of the OR tie and the number of devices in the OR tie. The pull-up resistor value is chosen from a range between maximum value (established to maintain the required V_{OH} with all the OR-tied outputs HIGH) and a minimum value (established so that the OR tie fan-out is not exceeded when only one output is LOW).

Minimum and Maximum Pull-Up Resistor Values

$$R_{X(MIN)} = \left(\frac{V_{CC(MAX)} - V_{OL}}{I_{OL} - N_2(LOW) \cdot 1.6\ mA}\right)$$

(E1-1)

$$R_{X(MAX)} = \left(\frac{V_{CC(MIN)} - V_{OH}}{N_1 I_{OH} + N_2(HIGH) \cdot 40\mu A}\right)$$

where:

R_X = External pull-up resistor
N_1 = Number of wired-OR outputs
N_2 = Number of input unit loads being driven
$I_{OH} = I_{CEX}$ = Output HIGH leakage current
I_{OL} = LOW level fanout current of driving element
V_{OL} = Output LOW voltage level (0.5V)
V_{OH} = Output HIGH voltage level (2.5V)
V_{CC} = Power Supply Voltage

Example: four 'F524 gate outputs driving four other gates or MSI inputs.

(E1-2)

$$R_{X(MIN)} = \left(\frac{5.5V - 0.5V}{8.0\ mA - 2.4\ mA} = \frac{5.0V}{5.6\ mA}\right) = 893\Omega$$

(E1-3)

$$R_{X(MAX)} = \left(\frac{4.5V - 2.5V}{4 \cdot 250\mu A + 2 \cdot 40\mu A} = \frac{2.0V}{1.08\ mA}\right) = 1852\Omega$$

where:

$N_1 = 4$
$N_2(HIGH) = 4 \cdot 0.5$ U.L. $= 2$ U.L.
$N_2(LOW) = 4 \cdot 0375$ U.L. $= 1.5$ U.L.
$I_{OH} = 250\mu A$
$I_{OL} = 8.0$ mA
$V_{OL} = 0.5V$
$V_{OH} = 2.5V$

Any values of pull-up resistor between 893 and 1852Ω can be used. The lower values yield the fastest speeds while the higher values yield the lowest power dissipation.

Thresholds and Noise Margins

The noise margins most often cited for TTL are obtained by subtracting the guaranteed maximum input HIGH level, V_{IH}, of a driven input from the guaranteed minimum output HIGH level, V_{OH}, of the driving source, and subtracting the guaranteed maximum output LOW level, V_{OL}, of the driver from the guaranteed minimum input LOW level, V_{IL}, of a driven circuit. The guaranteed worst-case values of these parameters vary slightly among the various circuit families and are summarized in Table 1-2. Note that although the 9000 Series V_{IH} and V_{IL} specifications have different limits at different temperatures (see data sheets), they are grouped with the 54/74 family in the table as a matter of convenience. Note also that the V_{OL} limit listed for 74LS is 0.5 V, whereas these circuits are also specified at 0.4 V at a lower level of I_{OL}. Noise margins obtained by the aforementioned subtractions are listed in Tables 1-3 through 1-6, for all combinations of driving and driven circuit types in the various circuit families. Noise margins calculated in this manner are quite conservative, since it is assumed that both the driver output characteristics and the receiver input characteristics are worst-case and that V_{CC} is on the low side for the driver and on the high side for the receiver.

Table 1-2 Parameter Limits

		Military (−55°C to +125°C)				Commercial (0°C to +70°C)				
	Fairchild TTL Families	V_{IL}	V_{IH}	V_{OL}	V_{OH}	V_{IL}	V_{IH}	V_{OL}	V_{OH}	**Units**
TTL	Standard TTL, 9000, 54/74	0.8	2.0	0.4	2.4	0.8	2.0	0.4	2.4	V
FAST	54F/74F	0.8	2.0	0.5	2.4	0.8	2.0	0.5	2.5	V
S-TTL	Schottky TTL, 54S/74S,93S	0.8	2.0	0.5	2.5	0.8	2.0	0.5	2.7	V
LS-TTL	Low Power Schottky TTL, 54LS/74LS	0.7	2.0	0.4	2.5	0.8	2.0	0.5	2.7	V

V_{OL} and V_{OH} are the voltages generated at the output. V_{IL} and V_{IH} are the voltages required at the input to generate the appropriate levels. The numbers given above are guaranteed worst-case values for standard outputs.

Table 1-3 LOW Level Noise Margins (Military)

To / From	TTL	FAST	S-TTL	LS-TTL	Units
TTL	400	400	400	300	mV
FAST	300	300	300	200	mV
S-TTL	300	300	300	200	mV
LS-TTL	400	400	400	300	mV

From V_{OL} to V_{IL}

Table 1-4 HIGH Level Noise Margins (Military)

To / From	TTL	FAST	S-TTL	LS-TTL	Units
TTL	400	400	400	400	mV
FAST	500	500	500	500	mV
S-TTL	500	500	500	500	mV
LS-TTL	500	500	500	500	mV

From V_{OH} to V_{IH}

Table 1-5 LOW Level Noise Margins (Commercial)

To / From	TTL	FAST	S-TTL	LS-TTL	Units
TTL	400	400	400	400	mV
FAST	300	300	300	300	mV
S-TTL	300	300	300	300	mV
LS-TTL	300	300	300	300	mV

From V_{OL} to V_{IL}

Table 1-6 HIGH Level Noise Margins (Commercial)

To / From	TTL	FAST	S-TTL	LS-TTL	Units
TTL	400	400	400	400	mV
FAST	500	500	500	500	mV
S-TTL	700	700	700	700	mV
LS-TTL	700	700	700	700	mV

From V_{OH} to V_{IH}

A more meaningful interpretation of noise margin can be gained by examining the relationship between input and output voltage of a circuit. Figures 1-10, 1-11 and 1-12 show the voltage transfer function of TTL, S-TTL and LS-TTL inverting gates, respectively. The steepest part of a particular curve, where the output changes rapidly for small changes in input, is called the threshold region. Input signals above or below this region cause little or no change in output and thus are of no concern. Problems can occur when an input voltage, whether steady-state, transient or a combination of both, causes an output voltage to rise or fall into the threshold region of its driven loads. Thus, noise of this magnitude can propagate.

The transfer characteristics of Figures 1-10 through 1-12 are essentially steady-state and thus apply for noise disturbances of long duration. For short pulses, however, the finite response time of a circuit has an effect on noise sensitivity. Figure 1-13 illustrates pulse noise immunity of TTL gates of the various families. These data are obtained by applying positive pulses to an otherwise LOW input and noting the combinations of pulse amplitude and duration required to cause the output to fall to 2.0 V, which is the guaranteed input HIGH level for TTL circuits. The curves show that S-TTL responds to the shortest pulses, as might be expected, and that pulse durations greater than about 4.0 ns have essentially the same effect as do input voltage. TTL (7400) is the least sensitive to noise pulses, with H-TTL and LS-TTL responses intermediate between those of 7400 and S-TTL. The flat portion of the various curves shows that LS-TTL is the most sensitive to long duration pulses, while 7400 is least sensitive. This can also be deduced by comparing the transfer functions of Figures 1-10 and 1-12; the LS-TTL threshold regions are nearer the left hand axis, indicating that a lower value of input voltage is required to affect the output voltage than is the case with plain TTL.

Figure 1-10 Voltage Transfer Function of TTL Gates @ – 55°C

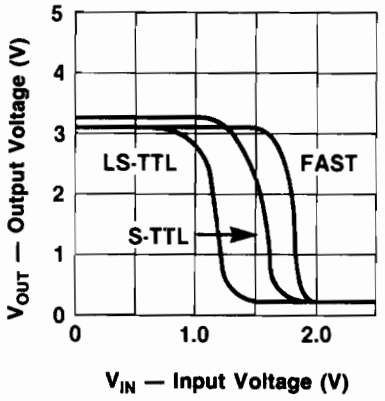


Figure 1-11 Voltage Transfer Function of TTL Gates @25°C

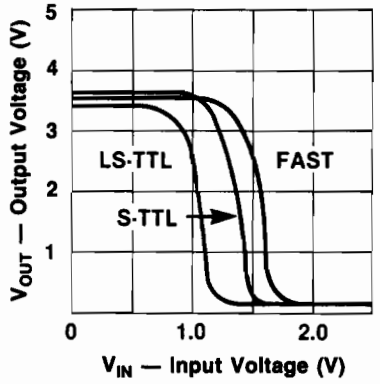


Figure 1-12 Voltage Transfer Functions of TTL Gates @ 125°C

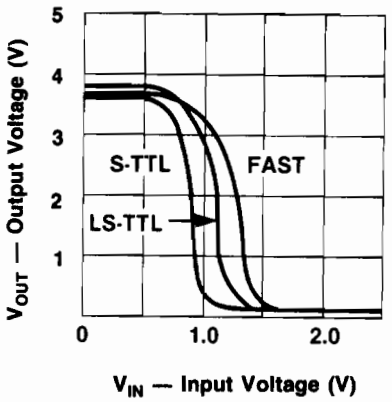
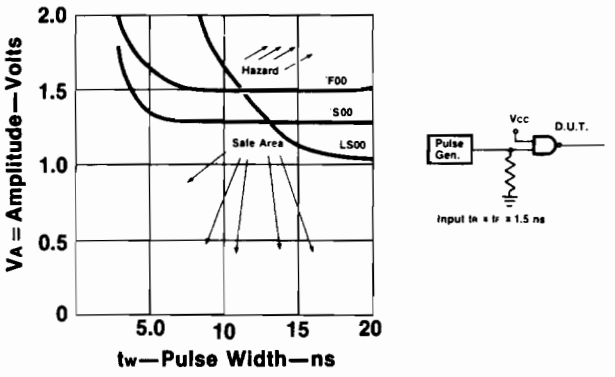


Figure 1-13 Pulse Noise Immunity of TTL Gates



Transmission Lines

Practical transmission lines, cables and strip lines used for TTL interconnections have a characteristic impedance between 30Ω and 150Ω. Thus none of the standard or low power TTL circuits can drive a transmission line, and only FAST is truly capable of driving a 50Ω line under worst case conditions.

These considerations, applicable only when the round trip delay of the line is longer than the rise or fall time of the driving signal ($2t_d > t_r$), do not affect most TTL interconnections. Short interconnections do not behave like a resistive transmission line, but more like a capacitive load. Since the rise time of different TTL outputs is known, the longest interconnection that can be tolerated without causing transmission line effects can easily be calculated and is listed in Table 1-7.

Table 1-7 PC Board Interconnections

TTL Family	Rise Time	Fall Time	Max Interconnection Length
54/74, 54/74LS	6—9 ns	4—6 ns	18 in. (45 cm)
54S/74S	4—6 ns	2—3 ns	9 in. (22.5 cm)
FAST	1.8—2.8 ns	1.6—2.6 ns	7.5 in. (19 cm)

Assuming 1.7 ns/foot propagation speed, typical for epoxy fiberglass PC boards with $\epsilon_r = 4.7$.

Slightly longer interconnections show minimal transmission line effects; the longer the interconnections, the greater the chance that system performance may be degraded due to reflections and ringing. The discussion of transmission line effects gives additional information on transmission line phenomena on longer lines. Good

system operation can generally be obtained by designing around 100Ω lines. A 0.026 inch (0.65 mm) trace on an 0.0625 inch (1.58 mm) epoxy-glass board ($\epsilon_r = 4.7$) with a ground plane on the other side represents a 100Ω line. Wire of 28 to 30 gauge (0.25 mm to 0.30 mm) twisted together forms a twisted pair line with a characteristic impedance of 100Ω to 115Ω. Wire over ground screen (3/4" squares) gives 150Ω to 250Ω impedance with a significant improvement in propagation speed, since the dielectric constant approaches that of air.

Transmission Line Effects

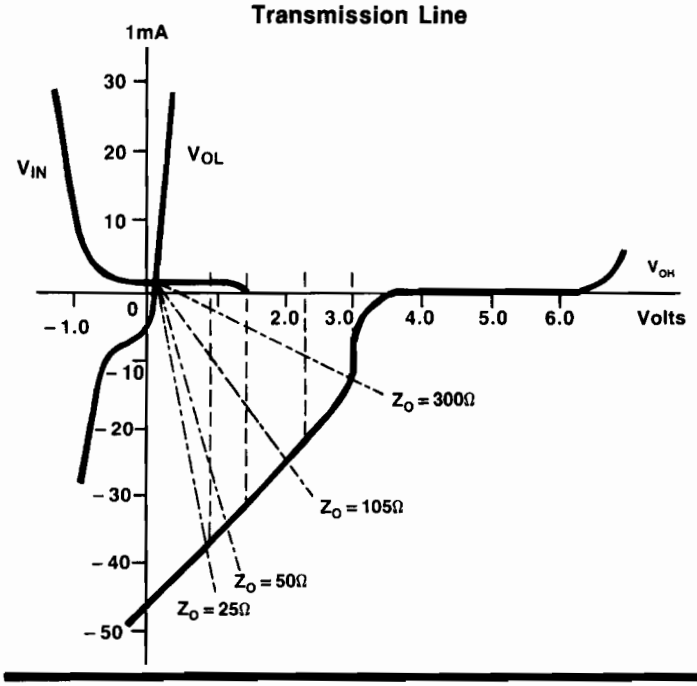
The fast rise and fall times of TTL outputs (2.0 ns to 6.0 ns) produce transmission line effects even with relatively short (<2 ft) interconnections. Consider one TTL device driving another, and the driver switching from the LOW to the HIGH state. If the propagation delay of the interconnection is long compared to the rise time of the signal, the arrangement behaves like a transmission line driven by a generator with a non-linear output impedance. Simple transmission line theory shows that the initial voltage step at the output just after the driver has switched is

(E1-4) $$V_{OUT} = V_E\left(\frac{Z_O}{Z_O + R_O}\right)$$

where Z_O is the characteristic impedance of the line, R_O is the output impedance of the driver, and V_E is the equivalent output voltage source in the driver, i.e., V_{CC} minus the forward drop of the pull-up transistors.

Figure 1-14 shows how the initial voltage step can be determined graphically by superimposing lines of constant impedance on the static input and output characteristics of TTL elements. The constant impedance lines are drawn from the intersection of the V_{IN} and V_{OL} characteristics, which is the quiescent condition preceding a LOW-to-HIGH transition. After this transition the V_{OH} characteristic applies, and the intersection of a particular impedance line with the V_{OH} characteristic determines the initial voltage step. The V_{OH} characteristic shown in Figure 1-14 has an R_O of about 80Ω and V_E of approximately 4.0 V, for calculation purposes.

Figure 1-14 Initial Output Voltage of TTL Driving Transmission Line



This initial voltage step propagates down the line and reflects at the end, assuming the typical case where the line is open-ended or terminated in an impedance greater than its characteristic impedance Z_O. Arriving back at the source, this reflected wave increases V_{OUT}. If the total round-trip delay is larger than the rise time of the driving signal, there is a staircase response at the driver output and anywhere along the line. If one of the loads (gate inputs) is connected to the line close to the driver, the initial output voltage V_{OUT} might not exceed V_{IH}. This input is then undetermined until after the round trip of the transmission line, thus slowing down the response of the system. Figure 1-15 shows the driver output waveform for four different line impedances. For Z_O of 25Ω and 50Ω, the initial voltage step is in the threshold region of a TTL input and the output voltage only rises above the guaranteed 2.0 V V_{IH} level after a reflection returns from the end of the line. If V_{OUT} is increased to >2.0 V by either increasing Z_O or decreasing R_O, additional delay does not occur. R_O is a characteristic of the driver output configuration, varying between the different TTL speed categories. Z_O can be changed by varying the width of the conductor and its distance from ground. Table 1-8 lists the lowest transmission line impedance that can be driven by different TTL devices to insure an initial voltage step of 2.0 V. Note that the worst case value, assuming a +30% tolerance on the current limiting resistor and a −10% tolerance on V_{CC}, is 80% higher than the value for nominal conditions.

A graphical method provides excellent insight into the effects of high-speed digital circuits driving interconnections acting as transmission lines. The method is basically to draw a load line for each input and output situation. Each load line starts at the previous quiescent point, determined where the previous load line intersects the appropriate characteristic. The magnitude of the slope of the load lines is identical and equal to the characteristic impedance of the line, but alternate load lines have opposite signs representing the change in direction of current flow. The points where the load lines intersect the input and output characteristics represent the voltage and current value at the input or output, respectively, for that reflection. The results (Figure 1-16) are shown with and without the input diode and illustrate how the input diode on TTL elements assists in eliminating spurious switching due to reflection.

Table 1-8 Transmission Line Drive Capability

TTL Family or Device	Collector Resistor R Ω	Lowest Transmission Line Impedance Ω				
		Worst Case (R + 30%)		Nominal	Best Case (R − 30%)	
54/74	130	241.4	204.8	136.8	84.6	75.8
54S/74S	55	110.0	92.2	61.1	37.5	33.4
5440/7440	100	185.7	157.5	105.2	65.1	58.3
54S/74S40	25	50.0	41.9	27.7	17.0	15.2
54F/74F00	45	66.2	57.7	40.9	27.6	25.0
54F/74F258	25	36.76	32.0	22.7	15.3	13.9
54F/74F240	15	22.0	19.2	13.6	9.2	8.3
Supply Voltage (V_{CC})		4.50	4.75	5.00	5.25	5.50

Figure 1-15 TTL Driving Transmission Line

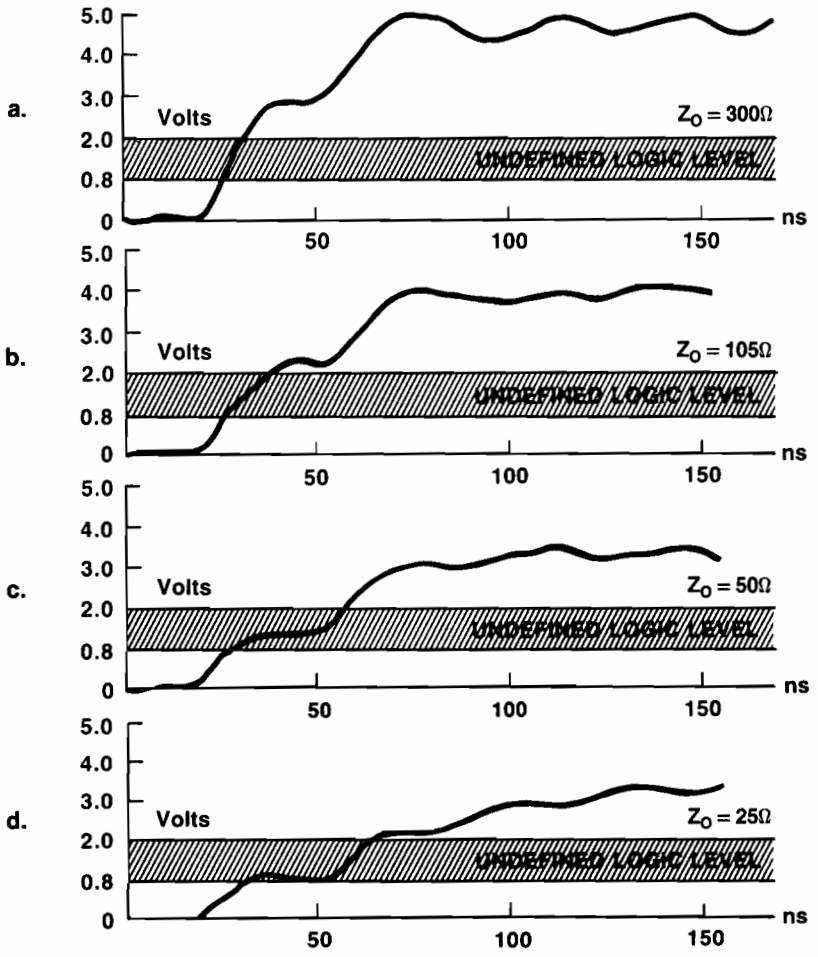
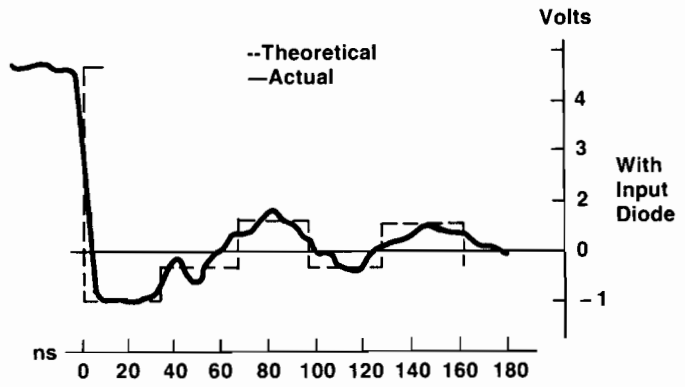
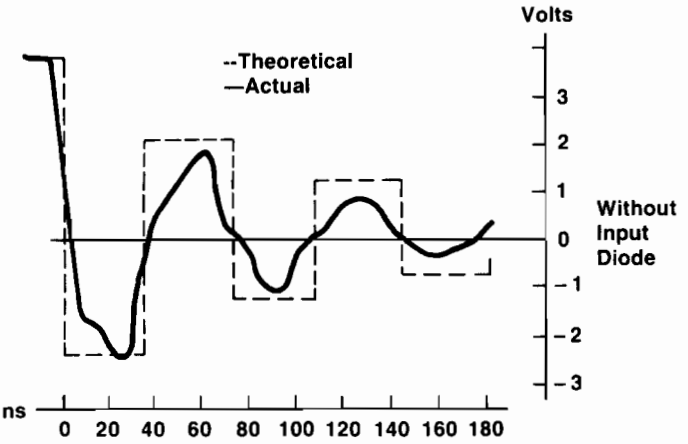


Figure 1-16 Ringing Caused by Reflections



a. With Input Diode



b. Without Input Diode

Decoupling

Decoupling capacitors should be used on every PC card, at least one for every five to ten standard TTL packages, one for every five 74LS and 74S packages, one for every three FAST packages, and one for every one-shot (monostable), line driver and line receiver package. They should be good quality rf capacitors of 0.01 μF to 0.1 μF with short leads. It is particularly important to place good rf capacitors near sequential (bistable) devices. In addition, a larger capacitor (preferably a tantalum capacitor) of 2.0 μF to 20 μF should be included on each card.

Ground

A good ground system is essential for a PC card containing a large number of packages. The ground can either be a good ground bus, or better yet, a ground plane which, incorporated with the V_{CC} supply, forms a transmission line power system. Power transmission systems, which can be attached to a PC card to give an excellent power system without the cost of a multilayer PC card, are commercially available. Ground loops on or off PC cards are to be avoided unless they approximate a ground plane.

Supply Voltage and Temperature

The nominal supply voltage V_{CC} for all TTL circuits is +5.0 V. Commercial grade parts are guaranteed to perform with ±10% supply tolerance (±500 mV) over an ambient temperature range of 0°C to 70°C. Mil grade parts are guaranteed to perform with a ±10% supply tolerance (±500 mV) over an ambient temperature range of −55°C to +125°C.

The actual junction temperature can be calculated by multiplying the power dissipation of the device with the thermal resistance of the package and adding it to the measured ambient temperature T_A or package (case) temperature T_C. For example, a device in Ceramic DIP (Θ_{JA} 100°C/W) dissipates typically 145 mW. At +55°C ambient temperature the junction temperature is:

(E1-5) $T_J = (0.145 \times 100) + 55°C = 70°C$

For package thermal data please refer to Chapter 11. Designers should bear in mind that localized temperatures can rise well above the general ambient in a system enclosure. On a large PC board mounted in a horizontal plane, for example, the local temperature surrounding an IC in the middle of the board can be quite high due to the heating effect of the surrounding packages and the very poor natural convection. Low velocity forced air cooling is usually sufficient to alleviate such localized static air conditions.

Interfacing

All circuits in the Fairchild TTL families, in fact all TTL devices presently manufactured, are compatible, and any TTL output can drive a certain number of TTL inputs. There are only subtle differences in the worst case noise immunity when low power, standard and Schottky TTL circuits are intermixed. Open-collector outputs, however, require a pull-up resistor to drive TTL inputs reliably, as discussed earlier.

While TTL is the dominating logic family, and many systems use TTL exclusively, there are cases where different semiconductor technologies are used in one system, either to improve the performance or to lower the cost, size and power dissipation. The following explains how TTL circuits can interface with DTL, ECL, CMOS and discrete transistors.

Interfacing TTL and DTL—Both DTL and TTL are current sinking families, operating on a +5.0 V supply. They interface perfectly. When TTL drives DTL, one DTL input represents 1 Unit Load in the LOW state, much less than 1 U.L. in the HIGH state. When DTL drives TTL, a 2 kΩ output has a drive capability of 8 U.L., a 6 kΩ output has a drive capability of 4 U.L.

Interfacing TTL and ECL—Mixing ECL and TTL logic families offers the design engineer a new level of freedom and opens the entire VHF frequency spectrum to the advantages of digital measurement, control and logic operation.

The chief advantages of emitter coupled logic are high speed, flexibility, design versatility and transmission line compatibility. But application and interfacing cost problems have traditionally discouraged the use of ECL in many areas, particularly in low cost, less sophisticated systems. Using 10K compensated ECL with new ECL/TTL interface devices and several new interfacing methods promises to extend the advantages of ECL to many low cost systems.

The most practical interfacing method for smaller systems involves using a common supply of +5.0 V to +5.2 V. Care must be exercised with both logic families when using this technique to assure proper bypassing of the power supply to prevent any coupling of noise between circuit families. If only a few 10K ECL packages are designed into a predominantly TTL system the safest method is to use a 0.01 μF miniature ceramic capacitor across each ECL device. This value capacitor has the highest Q, or bypassing efficiency. When larger systems are operated on a common supply, separate power busses to each logic family help prevent problems. Otherwise, good high frequency bypassing techniques are usually sufficient.

10K devices have high input impedance with input pull-down resistors (>20kΩ) to the negative supply. In the TTL to ECL interface circuits in Figure 1-17 it is assumed that the ECL devices have high input impedance.

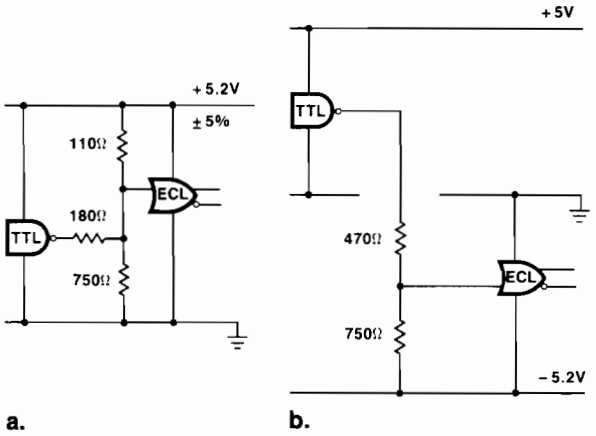
All circuits described operate with ±5% ECL and ±10% TTL supply variations, except those with ECL and TTL on a common supply. In those cases the supply can be ±10% with 10K ECL. All resistors are 1/4 W, ±5% composition type.

TTL to ECL conversion is easily accomplished with resistors, which simultaneously attenuate the TTL signal swing, shift the signal levels, and provide low impedance for damping and immunity to stray noise pick-up. The resistors should be located as near as possible to the ECL circuit for optimum effect. The circuits in Figure 1-17 assume an unloaded TTL gate as the standard TTL source. ECL input impedance is predominantly capacitive (≈3 pF); the net RC time constant of this capacitance with the indicated resistors assures a net propagation delay governed primarily by the TTL signal.

When interfacing between high voltage-swing TTL logic and low voltage-swing ECL logic, the more difficult conversion is from ECL to TTL. This requires a voltage amplifier to build up the 0.8 V logic swing to a minimum of 2.5 V. The circuits shown in Figure 1-18 may be used to interface from ECL to TTL.

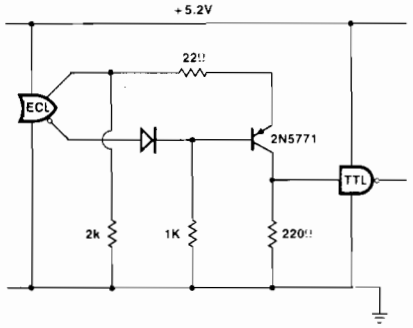
The higher speed converters usually have the lowest fan-out—only one or two TTL gates. This fan-out can be increased simply by adding a TTL buffer gate to the output of the converter. Another option, if ultimate speed is required, is to use additional logic converters.

Figure 1-17 TTL to ECL Conversion

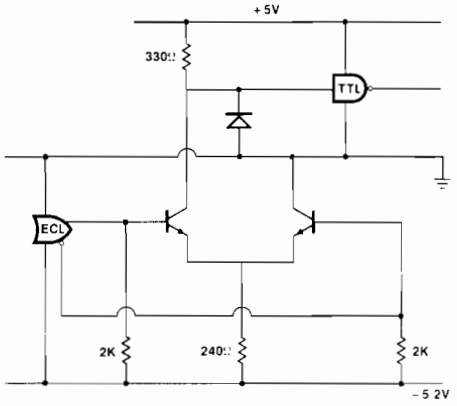


a. b.

Figure 1-18 ECL to TTL Conversion



a. Common Power Supply



b. Separate Power Supplies

Interfacing FAST and CMOS—Due to their wide operating voltage range, CMOS devices will function outside of the standard 5V ± 10% supply levels. For our purposes, only the case where both the FAST and CMOS devices are connected to the same voltage source will be considered.

FAST outputs can sink at least 20 mA in the low state. This is more than adequate to drive CMOS inputs to a valid low level. Due to their output designs, though, FAST outputs are unable to pull CMOS input to above approximately 4.0V. If the CMOS device does not have TTL compatible input levels, the FAST output should be pulled up with a resistor to V_{CC}. The value of this resistor will vary according to the system. Factors that affect the selection of the value are: edge rate—the smaller the resistor, the faster the edge rate; fanout—the smaller the resistor, the greater the fanout; noise margins—the smaller the resistor, the greater the output HIGH noise margin and the smaller the output LOW noise margin. FAST outputs can directly drive TTL-compatible CMOS inputs, such as the inputs on ACT or HCT devices, without pullup resistors.

Most CMOS outputs are capable of directly driving FAST inputs. Be aware, though, that TTL inputs have higher loading specifications than CMOS inputs. Care must be taken to insure that the CMOS outputs are not overloaded by the FAST input loading.

TTL Driving Transistors—Although high voltage, high current ICs are available, it is sometimes necessary to control greater currents or voltages than integrated circuits are capable of handling. When this condition arises, a discrete transistor with sufficient capacity can be driven from a TTL output. Discrete transistors are also used to shift voltages from TTL levels to logic levels for which a standard interface driver is not available.

The two circuits of Figure 1-19 show how TTL can drive npn transistors. The first circuit is the most efficient but requires an open-collector TTL output. The other circuit limits the output current from the TTL totem pole output through a series resistor.

Shifting a TTL Output to Negative Levels—The circuit of Figure 1-20 uses a pnp transistor to shift the TTL output to a negative level. When the TTL output is HIGH, the transistor is cut off and the output voltage is $-V_X$. When the TTL output is LOW, the transistor conducts and the output voltage is

(E1-6) $$-V_X + \frac{R1}{R2}(V_{CC} - 2.0V)$$

if the transistor is not saturated, or slightly positive if the transistor is allowed to saturate.

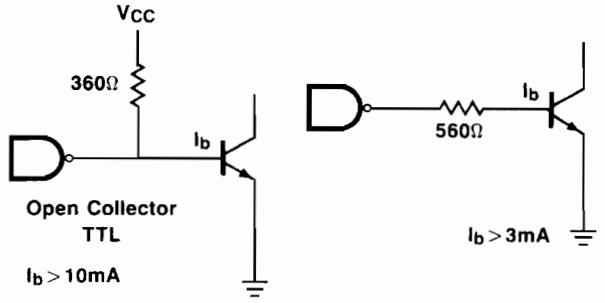
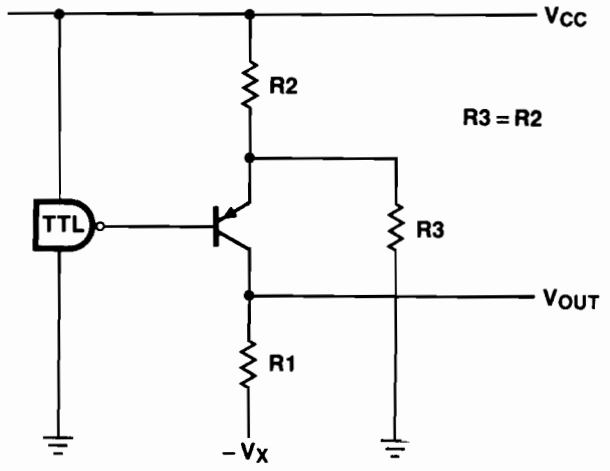


Figure 1-19 TTL Driving npn Transistors

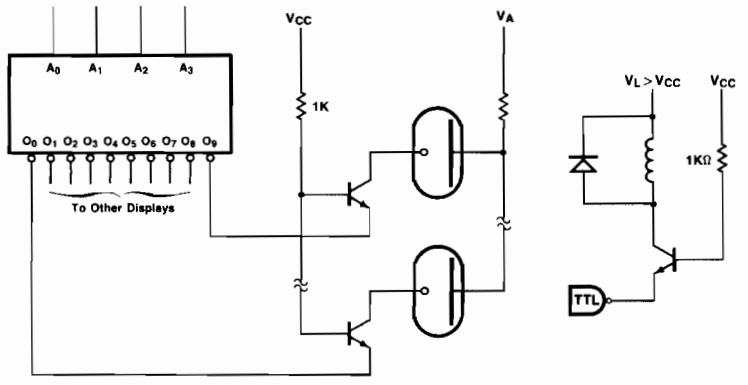
Figure 1-20 pnp Transistor Shifting TTL Output



High Voltage Drivers—A TTL output can be used to drive high voltage, low current loads through the simple, non-inverting circuits shown in Figure 1-21. This can be useful for driving gas discharge displays or small relays, where the TTL output can handle the current but not the voltage. Load current should not exceed $I_{OL}(-4\ mA)$.

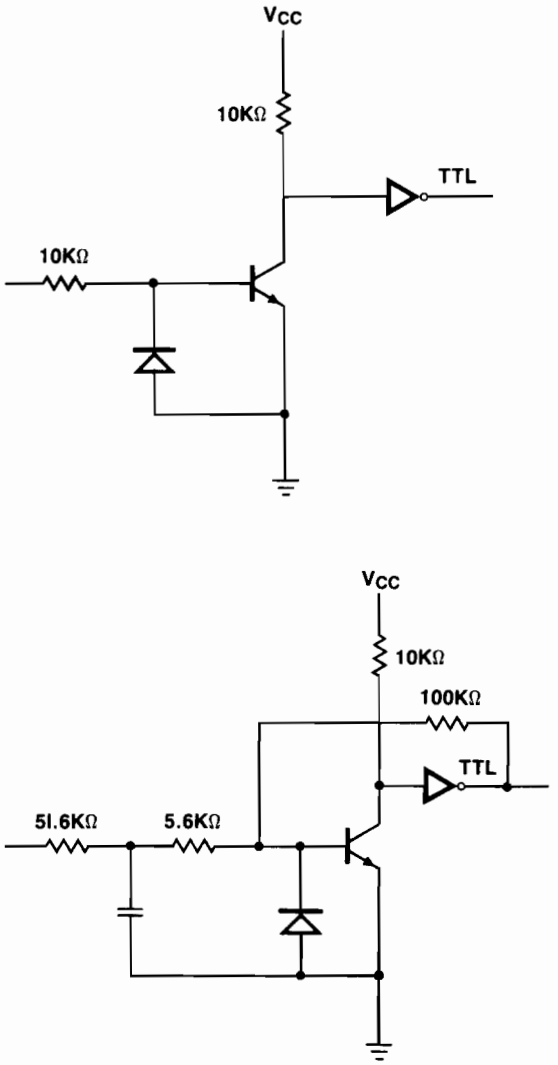
Transistors Driving TTL—It is sometimes difficult to drive the relatively low impedance and narrow voltage range of TTL inputs directly from external sources, particularly in a rough, electrically noisy environment. The circuits shown in Figure 1-22 can handle input signal swings in excess of ± 100 V without harming the circuits. The second circuit has an input RC filter that suppresses noise. Unambiguous TTL voltage levels are generated by the positive feedback (Schmitt trigger) connection.

Figure 1-21 Non-Inverting High Voltage Drivers



a. b.

Figure 1-22 Transistors Driving TTL



TTL Characteristics

Glossary

Currents—Positive current is defined as conventional current flow into a device. Negative current is defined as conventional current flow out of a device. All current limits are specified as absolute values.

I_{CC} **Supply Current**—The current flowing into the V_{CC} supply terminal of a circuit with the specified input conditions and the outputs open. When not specified, input conditions are chosen to guarantee worst case operation.

I_{IH}, I_I **Input HIGH Current**—The current flowing into an input when a specified HIGH voltage is applied.

I_{IL} **Input LOW Current**—The current flowing out of an input when a specified LOW voltage is applied.

I_{OH} **Output HIGH Current**—The current flowing out of the output when it is in the HIGH state. For a turned off open-collector output with a specified HIGH output voltage applied, the I_{OH} is the leakage current.

I_{OL} **Output LOW Current**—The current flowing into an output when it is in the LOW state.

I_{OS} **Output Short Circuit Current**—The current flowing out of a HIGH-state output when that output is short circuited to ground (or other specified potential).

I_{OZH} **Output OFF Current HIGH**—The current flowing into a disabled 3-state output with a specified HIGH output voltage applied.

I_{OZL} **Output OFF Current LOW**—The current flowing out of a disabled 3-state output with a specified LOW output voltage applied.

Voltages—All voltages are referenced to the ground pin. Negative voltage limits are specified as absolute values (i.e., -10.0 V is greater than -1.0 V).

V_{CC} **Supply Voltage**—The range of power supply voltage over which the device is guaranteed to operate within the specified limits.

$V_{CD(Max)}$ **Input Clamp Diode Voltage**—The most negative voltage at an input when a specified current is forced out of that input terminal. This parameter guarantees the integrity of the input diode, intended to clamp negative ringing at the input terminal.

V_{IH} **Input HIGH Voltage**—The range of input voltages that represents a logic HIGH in the system.

$V_{IH(Min)}$ **Minimum Input HIGH Voltage**—The minimum allowed input HIGH in a logic system. This value represents the guaranteed input HIGH threshold for the device.

V_{IL} **Input LOW Voltage**—The range of input voltages that represent a logic LOW in the system.

$V_{IL(Max)}$ **Maximum Input LOW Voltage**—The maximum allowed input LOW in a system. This value represents the guaranteed input LOW threshold for the device.

$V_{OH(Min)}$ **Output HIGH Voltage**—The minimum voltage at an output terminal for the specified output current I_{OH} and at the minimum value of V_{CC}.

$V_{OL(Max)}$ **Output LOW Voltage**—The maximum voltage at an output terminal sinking the maximum specified load current I_{OL}.

V_{T+} **Positive-Going Threshold Voltage**—The input voltage of a variable threshold device (i.e., Schmitt Trigger) that is interpreted as a V_{IH} as the input transition rises from below $V_{T-(Min)}$.

V_{T-} **Negative-Going Threshold Voltage**—The input voltage of a variable threshold device (i.e., Schmitt Trigger) that is interpreted as a V_{IL} as the input transition falls from above $V_{T+(Max)}$.

AC Switching Parameters

f_t **Maximum Transistor Operating Frequency**—The frequency at which the gain of the transistor has dropped by 3 decibels.

f_{max} **Toggle Frequency/Operating Frequency**—The maximum rate at which clock pulses may be applied to a sequential circuit. Above this frequency the device may cease to function.

t_{PLH} **Propagation Delay Time**—The time between the specified reference points, normally 1.5 V on the input and output voltage waveforms, with the output changing from the defined LOW level to the defined HIGH level.

t_{PHL} **Propagation Delay Time**—The time between the specified reference points, normally 1.5 V on the input and output voltage waveforms, with the output changing from the defined HIGH level to the defined LOW level.

t_w **Pulse Width**—The time between 1.5 V amplitude points on the leading and trailing edges of a pulse.

t_h **Hold Time**—The interval immediately following the active transition of the timing pulse (usually the clock pulse) or following the transition of the control input to its latching level, during which interval the data to be recognized must be maintained at the input to ensure its continued recognition. A negative hold time indicates that the correct logic level may be released prior to the active transition of the timing pulse and still be recognized.

t_s **Setup Time**—The interval immediately preceding the active transition of the timing pulse (usually the clock pulse) or preceding the transition of the control input to its latching level, during which interval the data to be recognized must be maintained at the input to ensure its recognition. A negative setup time indicates that the correct logic level may be initiated sometime after the active transition of the timing pulse and still be recognized.

t_{PHZ} **Output Disable Time (of a 3-State Output) from HIGH Level**—The time between the 1.5 V level on the input and a voltage 0.3 V below the steady state output HIGH level with the 3-state output changing from the defined HIGH level to a high impedance (off) state.

t_{PLZ} **Output Disable Time (of a 3-State Output) from LOW Level**—The time between the 1.5 V level on the input and a voltage 0.3 V above the steady state output LOW level with the 3-state output changing from the defined LOW level to a high impedance (off) state.

t_{PZH} **Output Enable Time (of a 3-State Output) to a HIGH Level**—The time between the 1.5 V levels of the input and output voltage waveforms with the 3-state output changing from a high impedance (off) state to a HIGH level.

t_{PZL} **Output Enable Time (of a 3-State Output) to a LOW Level**—The time between the 1.5 V levels of the input and output voltage waveforms with the 3-state output changing from a high impedance (off) state to a LOW level.

t_{rec} **Recovery Time**—The time between the 1.5 V level on the trailing edge of an asynchronous input control pulse and the same level on a synchronous input (clock) pulse such that the device will respond to the synchronous input.

Package Markings

C	Marking code letter indicating that the device is guaranteed to meet the specifications for the commercial temperature range.
D, SD	Package code letter for ceramic Dual In-line and 0.300" wide 24-pin ceramic Dual In-Line packages.
F	Package code letter for ceramic flatpaks.
M	Marking code letter indicating that the device is guaranteed to meet the specifications for the military temperature range.
P, SP	Package code letter for plastic Dual In-line and 0.300" wide 24-pin plastic Dual In-line packages.
QB	Marking code indicating in-house 38510, level B reliability screening (military grade only).
QR	Marking code indicating commercial/industrial reliability screening.
S	Package code letter for small outline surface mount packages.
XC, XM	Shorthand for the commercial or military temperature range specifications or devices; the letter X stands for the code letter of any package in which the device is available.

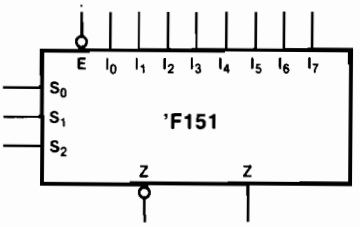
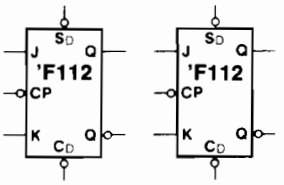
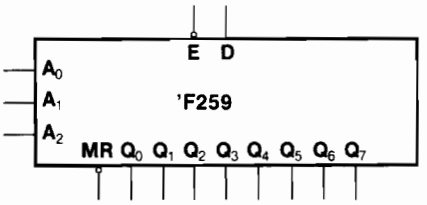
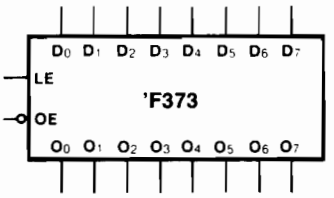
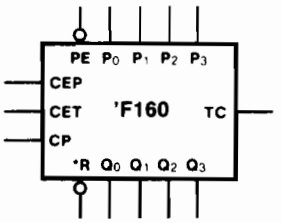
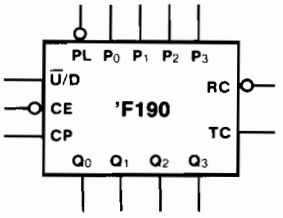
Logic Symbols and Terminology

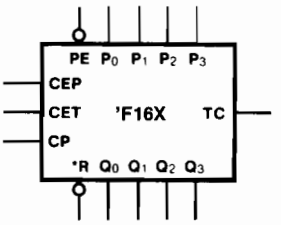
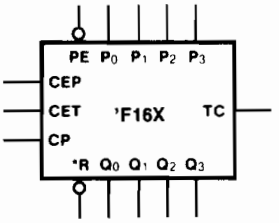
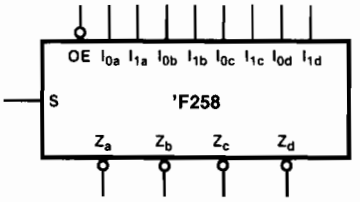
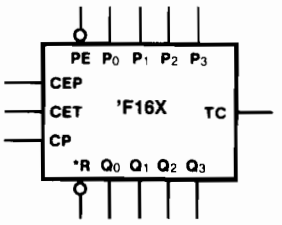
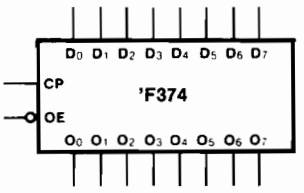
The definitions of LOW and HIGH logic levels are: LOW—a voltage defined by V_{IL}; HIGH—a voltage defined by V_{IH}. A LOW condition represents Logic 0; a HIGH condition, Logic 1.

The logic symbols used to represent the MSI devices follow Mil Std 806B for logic symbols. MSI elements are represented by rectangular blocks with appropriate external AND/OR gates when necessary. A small circle at an external input means that the specific input is active LOW; i.e., it produces the desired function, in conjunction with other inputs, if its voltage is the lower of the two logic levels in the system. A circle at the output indicates that when the function designated is True, the output is LOW. Generally, inputs are at the top and left and outputs appear at the bottom and right of the logic symbol. An exception is the asynchronous Master Reset in some sequential circuits which is always at the left hand bottom corner.

Inputs and outputs are labeled with mnemonic letters as illustrated in Table 1-1. Note that an active LOW function labeled outside of the logic symbol is given a bar over the label, while the same function inside the symbol is labeled without the bar. When several inputs or outputs use the same letter, subscript numbers starting with zero are used in an order natural for device operation.

This nomenclature is used throughout this book and may differ from nomenclature used on other data books (notably early 7400 MSI), where outputs use alphabetic subscripts or use number sequences starting with one.

Label	Meaning	Example
I_X	General term for inputs to combinatorial circuits.	
J,K S,R D	Inputs to JK, SR, and D flip-flops and latches.	
A_X, S_X	Address or Select inputs, used to select an input, output, data route, junction, or memory location.	
LE	Enable, active LOW on all TTL/MSI. A latch can receive new data when its Enable input is in the active state.	
$\overline{PE}$ P	Parallel Enable, a control input used to synchronously load information in parallel into an otherwise autonomous circuit. Parallel data inputs to shift registers and counters.	
$\overline{PL}$	Parallel Load; similar to Parallel Enable except that $\overline{PL}$ overrides the clock and forces parallel loading asynchronously.	

$\overline{MR}$ $\overline{SR}$	Master Reset, synchronously resets all outputs to zero, overriding all other inputs. Synchronous Reset, resets all outputs to zero with active edge of clock.	 *R = $\overline{MR}$ on 'F160/'F161 $\overline{SR}$ on 'F162/'F163
CP CE, CEP, CET	Clock Pulse, generally a HIGH-to-LOW-to-HIGH transition. An active HIGH clock (no circle) means outputs change on LOW-to-HIGH clock transition. Count Enable inputs for counters.	
Z_X, O_X, F_X	General terms for outputs of combinatorial circuits.	
Q_X TC	General term for latch and flip-flop outputs. If they pass through an enable gate before exiting the package, Q or $\overline{Q}$ changes to O or $\overline{O}$. Terminal Count output (1111 for up binary counters, 1001 for up decimal counters, or 0000 for down counters).	
$\overline{OE}$	Output Enable, used to force 3-state outputs into the high impedance state.	

FAST®

FAIRCHILD
A Schlumberger Company

Multiplexers

Introduction

Digital multiplexers are combinatorial (non-memory) devices controlled by a selector address which routes one of many input signals to the output. They can be considered semiconductor equivalents to multiposition switches or stepping switches.

Multiplexers are used for data routing and time division multiplexing and can also generate complex logic functions. For example, the 'F151 and 'F251 8-input/1-pole multiplexers generate any of the 2^{16} different logic functions of four variables; the 'F153, 'F253, 'F352, 'F353 4-input/2-pole multiplexers generate any two of the 256 functions of three variables; the 'F157, 'F158, 'F257, 'F258, 'F398, and 'F399 2-input/4-pole multiplexers generate any four of the sixteen functions of two variables. A single multiplexer package can replace several gate packages, saving printed circuit board area, interconnections, propagation delays, power dissipation, design effort, and component costs. The 'F350 4-input multiplexer is specifically designed to provide a shifting capability.

The examples which follow in this section generally employ the basic circuits of each type ('F151, 'F157, 'F153); however, in many cases the designer may find it beneficial to employ one of the alternate circuits. The 'F352 and 'F353 are the inverted versions of the 'F153 and 'F253 respectively.

'F157 Quad 2-Input Multiplexer

Description and Operation

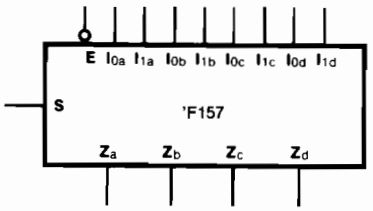


Figure 2-1

Table 2-1 Truth Table

Inputs				Output
$\overline{E}$	S	I_{0X}	I_{1X}	Z_X
H	X	X	X	L
L	H	X	L	L
L	H	X	H	H
L	L	L	X	L
L	L	H	X	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

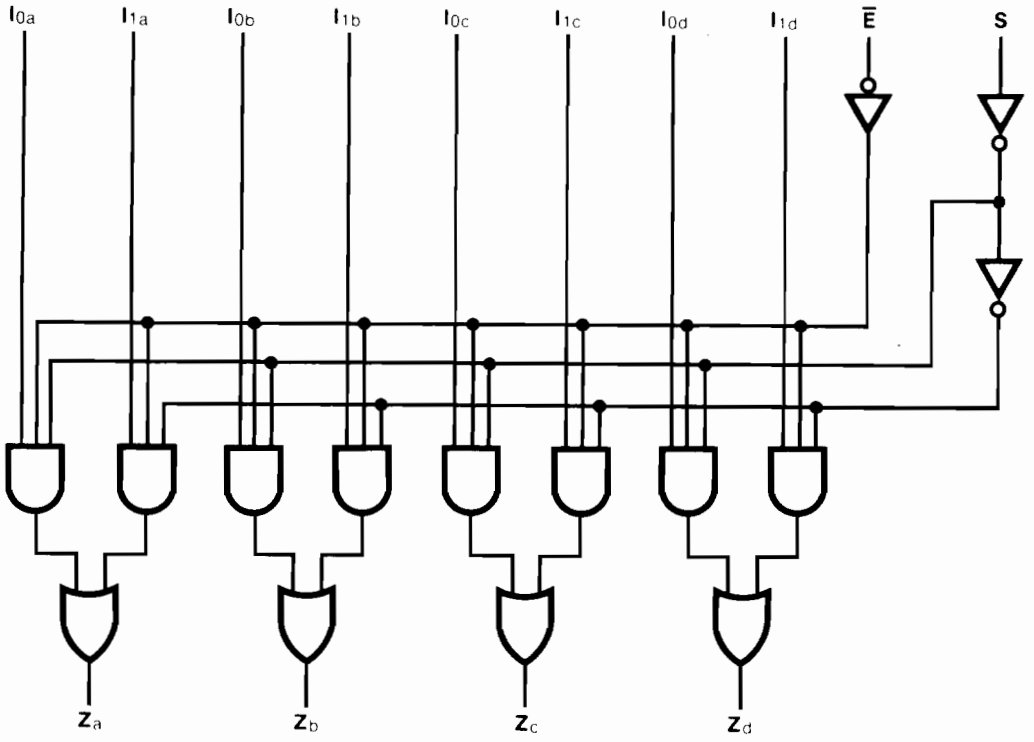


Figure 2-2 Logic Diagram

The 'F157 quad 2-input multiplexer has common input select logic, common active LOW Enable and non-inverting outputs. It allows four bits of data to be switched in parallel to the appropriate outputs from four 2-bit data sources. When the Enable is not active, all the outputs are held LOW.

The 'F158 and 'F258 have inverted outputs; the 'F257 and 'F258 differ in that the 3-state outputs permit bus interfaces. The 'F398 and 'F399 have registers on the output to latch the selected input data; the 'F399 provides only the true outputs, while the 'F398 provides both true and complement outputs. It should be noted that the inverting parts 'F158, 'F258 are considerably faster than their non-inverting counterparts.

Data Routing

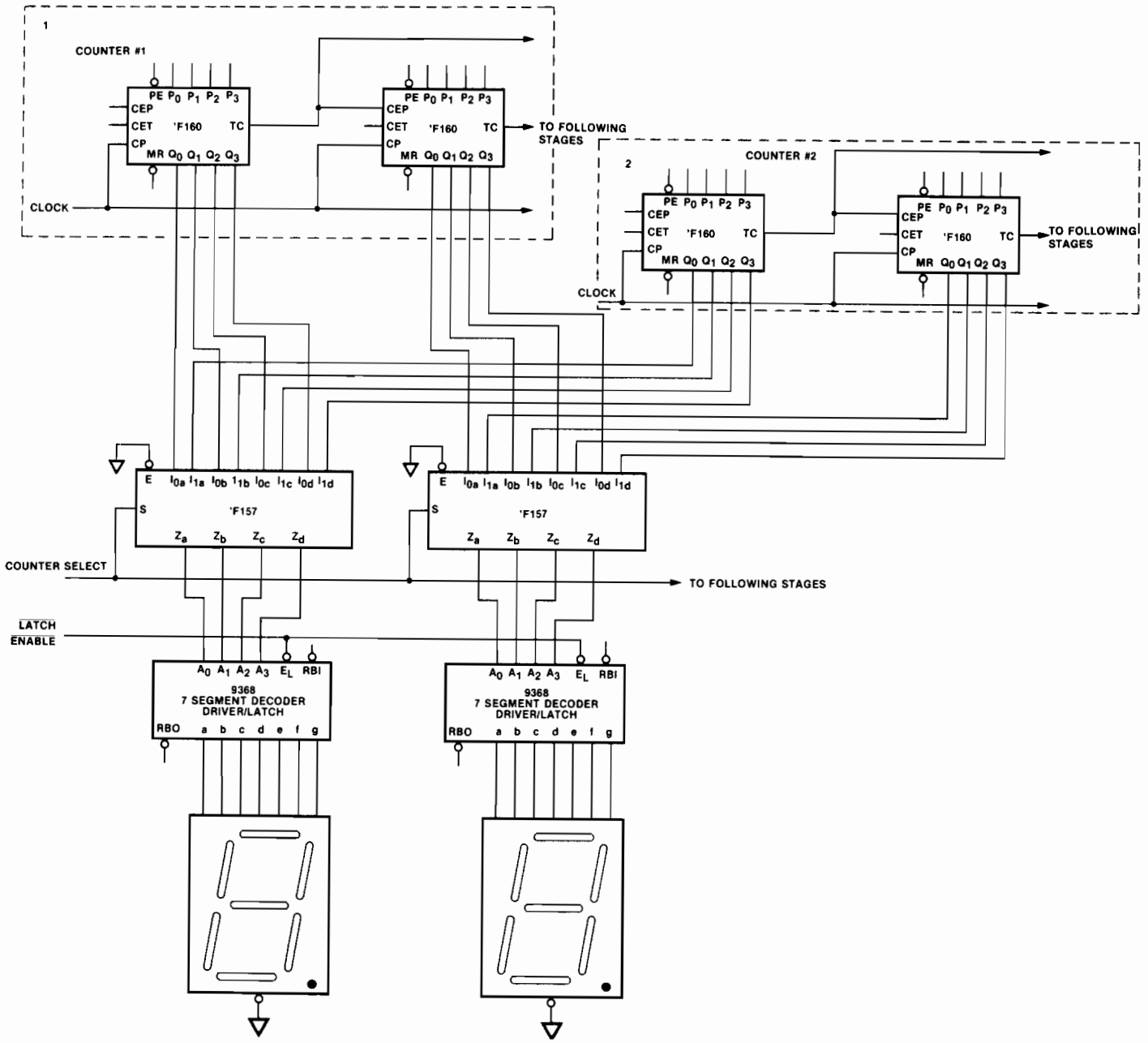


Figure 2-3

Multiplexers usually route data from one of several sources to one destination. One typical application is shown here. This system displays the contents of one of two multidigit BCD counter banks. The 'F157 multiplexers select one of the two counters: when the counter Select line is LOW, counter 1 is selected; when it is HIGH, counter 2 is selected. The multiplexer outputs feed into the 9368 BCD to 7-segment decoder drivers with input latches.

The display follows the selected counter when the Latch Enable input is LOW. When this line is HIGH, the display is no longer affected by input changes, but retains the information that was applied prior to the LOW-to-HIGH transition of the Latch Enable. The 9368 interfaces directly with common cathode LED displays.

Dual 10-Input Multiplexer

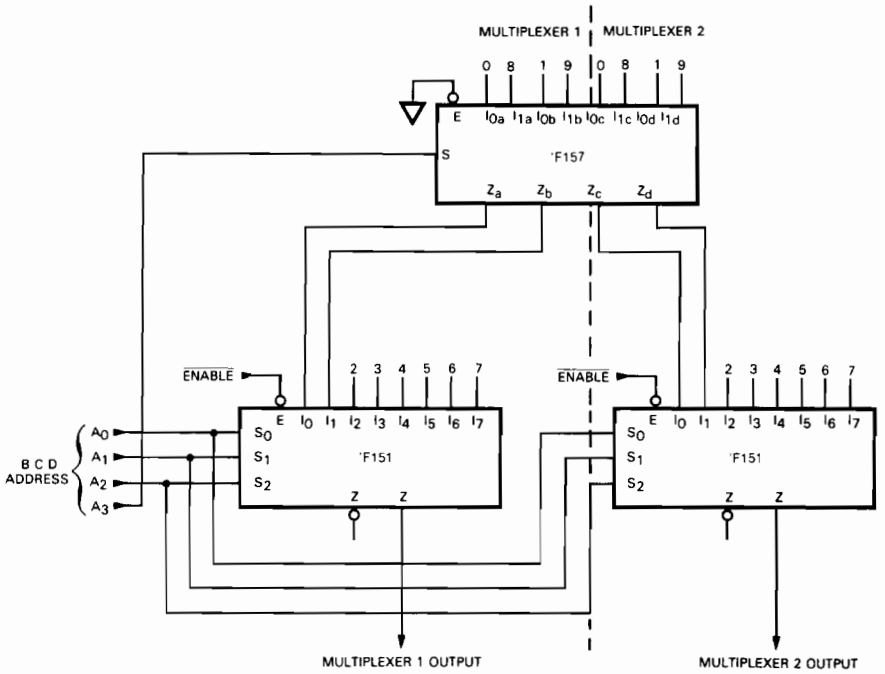


Figure 2-4

This data routing application is a part of a dual 10-input, BCD addressed multiplexer. One 'F157 and two 'F151 8-input multiplexers are used to route two sets of ten inputs to two output lines. Other decade multiplexing circuits are shown later in this section.

'F350 4-Bit Shifter

Description and Operation

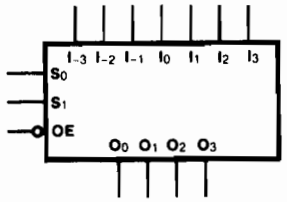


Figure 2-5

Table 2-2 Truth Table

Inputs			Outputs			
$\overline{OE}$	S_1	S_0	O_0	O_1	O_2	O_3
H	X	X	Z	Z	Z	Z
L	L	L	I_0	I_1	I_2	I_3
L	L	H	I_{-1}	I_0	I_1	I_2
L	H	L	I_{-2}	I_{-1}	I_0	I_1
L	H	H	I_{-3}	I_{-2}	I_{-1}	I_0

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = High Impedance

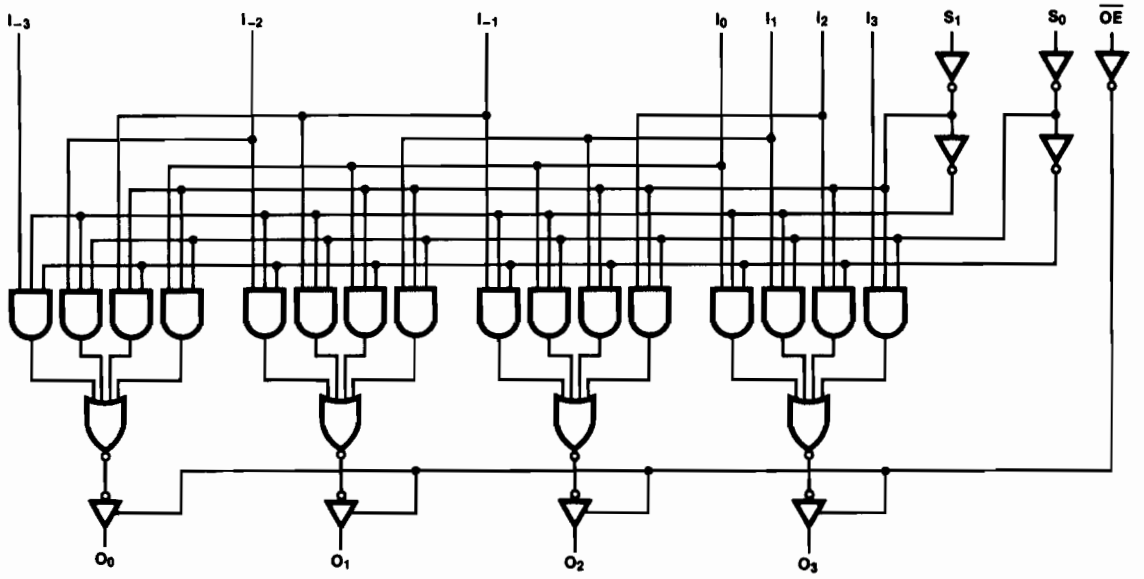


Figure 2-6 Logic Diagram

The 'F350 is a specialized multiplexer that accepts a 4-bit word and shifts it 0, 1, 2 or 3 places, as determined by two Select (S_0, S_1) inputs. For expansion to longer words, three linking inputs are provided for lower-order bits; thus two packages can shift an 8-bit word, four packages a 16-bit word, etc. Shifting by more than three places is accomplished by paralleling the 3-state outputs of different packages and using the Output Enable ($\overline{OE}$) inputs as a third Select level. With appropriate interconnections, the 'F350 can perform zero-backfill, sign-extend or end-around (barrel) shift functions.

'F350 4-Bit Shifter Applications

16-Bit Shift-Up, 0 To 3 Places, Zero Backfill

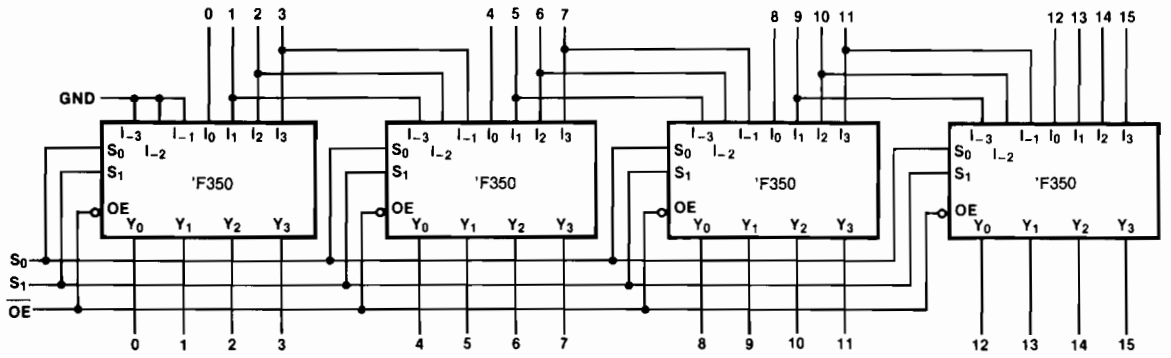


Figure 2-7

This circuit shifts the input 16-bit word to the right by the number of places designated in the table. Zero filling of the LSBs is illustrated here. The order of the bits could be interchanged and the I_{-3}, I_{-2}, and I_{-1} inputs could be connected to I_0 to provide sign extension. Signed twos complement division by 2, 4, or 8 would thus be implemented.

Table 2-3 Function Table

S_1	S_0	Shift Function
L	L	No Shift
L	H	Shift 1 Place
H	L	Shift 2 Places
H	H	Shift 3 Places

H = HIGH Voltage Level
L = LOW Voltage Level

8-Bit End Around Shift, 0 To 7 Places

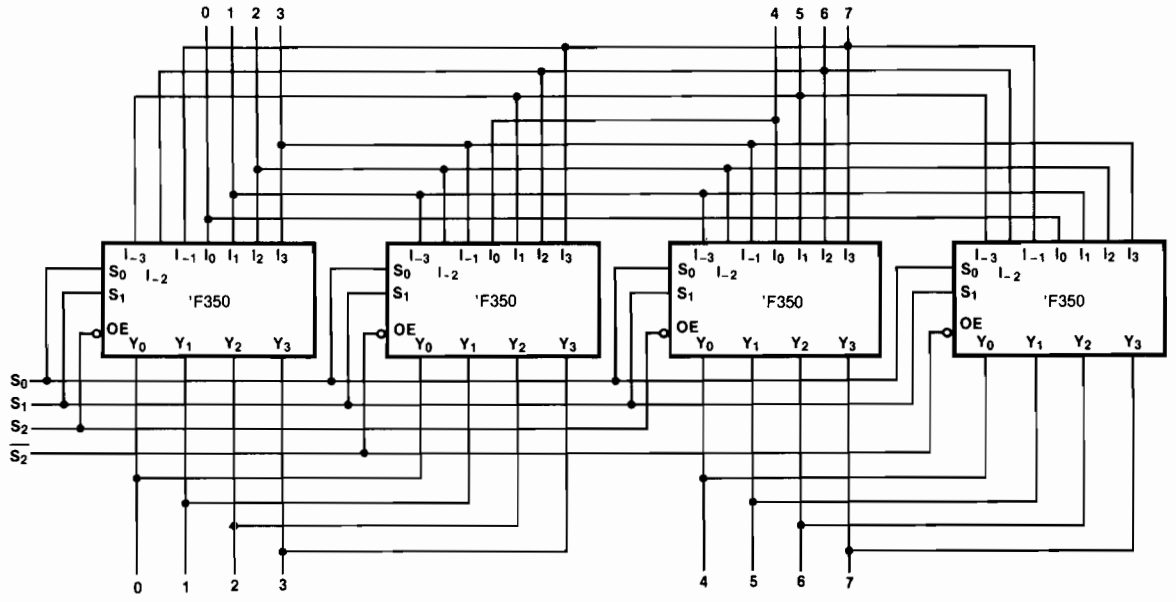


Figure 2-8

Full end around shifting of eight bits can be performed by employing four 'F350s as shown above. The advantage of the multiplexer architecture versus a shift register is that the shifting is dependent only on the internal propagation delays. Thus any length shift can be accomplished within 11 ns worst case.

Table 2-4 Function Table

S_2	S_1	S_0	Shift Function
L	L	L	No Shift
L	L	H	Shift End Around 1
L	H	L	Shift End Around 2
L	H	H	Shift End Around 3
H	L	L	Shift End Around 4
H	L	H	Shift End Around 5
H	H	L	Shift End Around 6
H	H	H	Shift End Around 7

H = HIGH Voltage Level
L = LOW Voltage Level

'F153/'F253/'F353 Dual 4-Input Multiplexer

Description and Operation

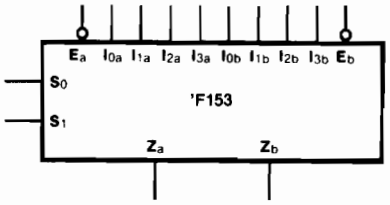


Figure 2-9

Table 2-5 Truth Table

Select Inputs		Inputs (a or b)					Output
S_0	S_1	$\overline{E}$	I_0	I_1	I_2	I_3	Z
X	X	H	X	X	X	X	L
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	L	L
H	H	L	X	X	X	H	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

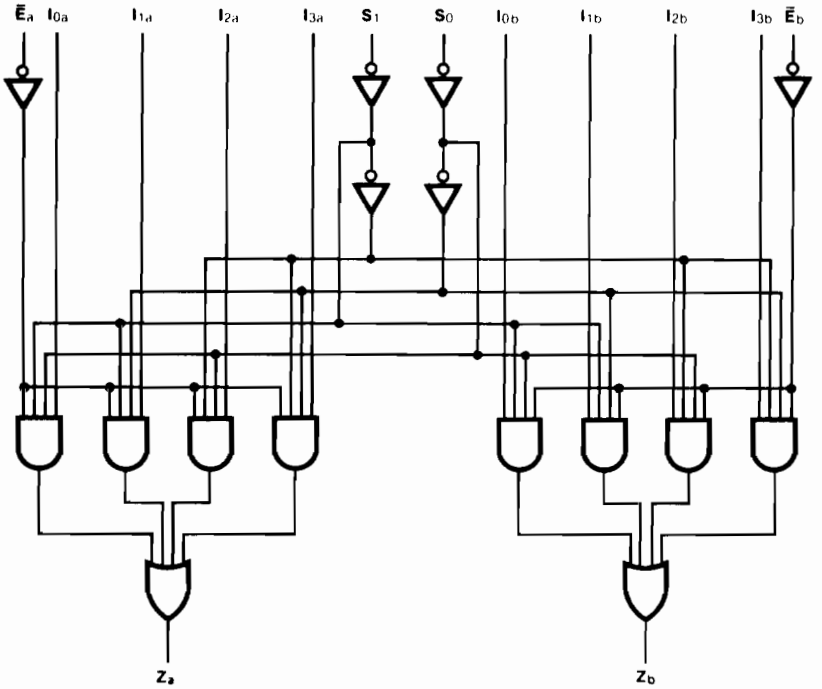


Figure 2-10 Logic Diagram

The 'F153, 'F253 and 'F353 are dual 4-input multiplexers with common Select inputs and separate Enable inputs. They allow two bits of data to be selected from two sets of 4-input sources. The 'F253 has 3-state output control. The 'F353 is equivalent to the 'F253 with inverted outputs.

'F151, 'F251 8-Input Multiplexers

Description and Operation

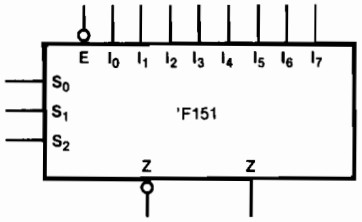


Figure 2-11

Table 2-6 Truth Table

Inputs				Outputs	
$\overline{E}$	S_2	S_1	S_0	$\overline{Z}$	Z
H	X	X	X	H	L
L	L	L	L	$\overline{I}_0$	I_0
L	L	L	H	$\overline{I}_1$	I_1
L	L	H	L	$\overline{I}_2$	I_2
L	L	H	H	$\overline{I}_3$	I_3
L	H	L	L	$\overline{I}_4$	I_4
L	H	L	H	$\overline{I}_5$	I_5
L	H	H	L	$\overline{I}_6$	I_6
L	H	H	H	$\overline{I}_7$	I_7

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

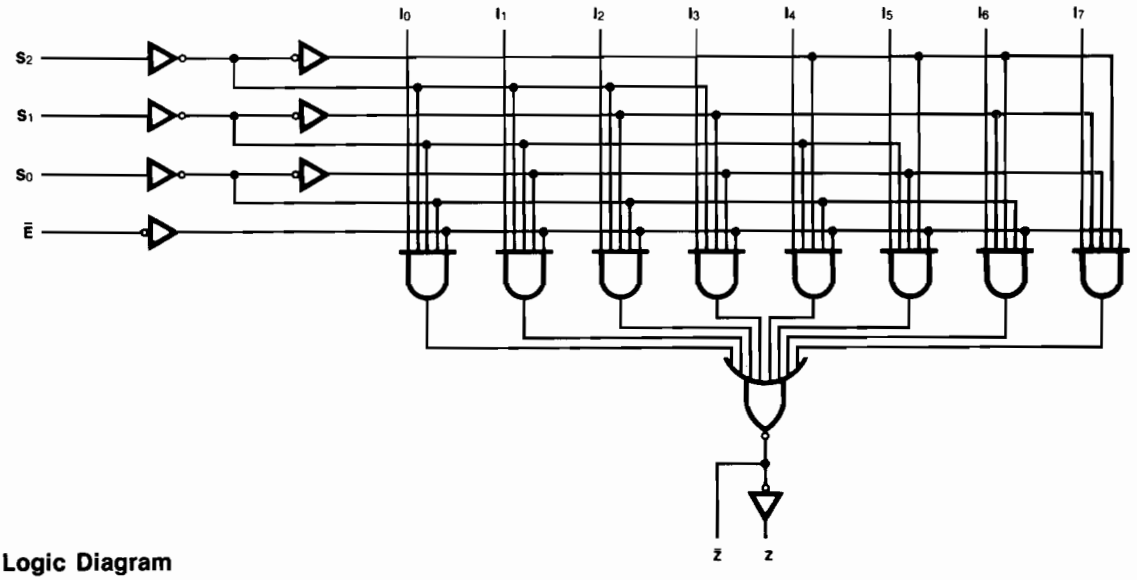
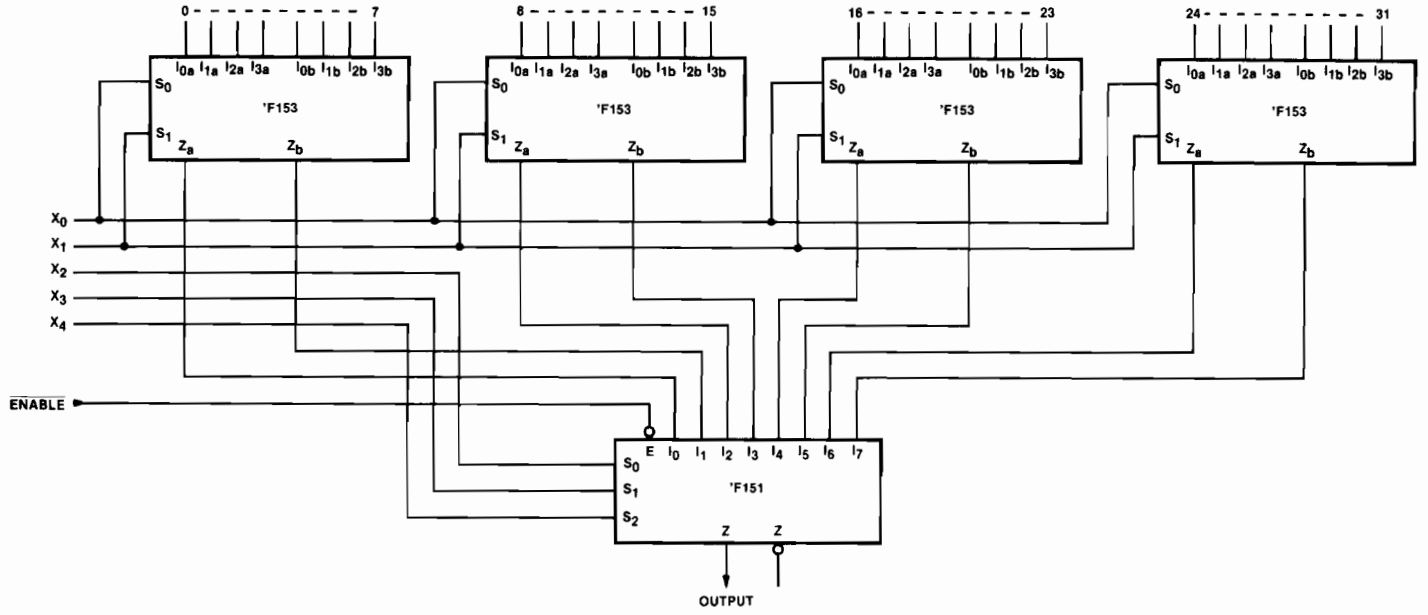


Figure 2-12 Logic Diagram

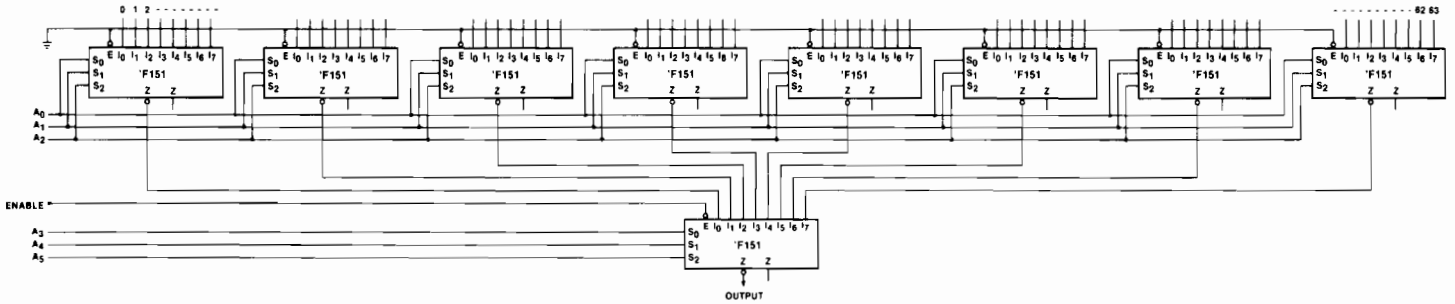
The 'F151 and 'F251 are 8-input multiplexers which select one bit of data from up to eight sources. They have internal select decoding, active LOW Enable and complementary outputs. When the $\overline{E}$ input is active (LOW), data is routed from one particular multiplexer input to the outputs according to the 3-bit code applied to the Select inputs. When the $\overline{E}$ is inactive (HIGH), the Z output is LOW and the $\overline{Z}$ output is HIGH regardless of all other input conditions for the 'F151; both outputs are high impedance (3-state) for the 'F251, permitting multiplexer expansion.

'F151 8-Input Multiplexer Applications

Time Multiplexing/Data Routing



a. 32-Input Multiplexing Circuit



b. 64-Input Multiplexing Circuit

Figure 2-13

Alone, the 'F153 and the 'F151 permit time multiplexing of a maximum of four and eight data lines, respectively. By cascading these devices in two or more levels, the number of inputs can be increased. The circuit in Figure 2-13a above shows two levels of multiplexers cascaded to implement a 32-input multiplexer with a delay of about 12.5 ns. It can be expanded to the 64-input multiplexer shown in Figure 2-13b with a typical delay of 10 ns; worst case over military temperature range 17.5 ns. The 'F151 Enable can be used to gate the selected data out. Note that the negative outputs are used at both levels of the 64-input example to minimize propagation delay. As indicated in the circuit schematic, the assertion output is generated by reinverting the negative output and is therefore slower.

8-Step Control Sequence

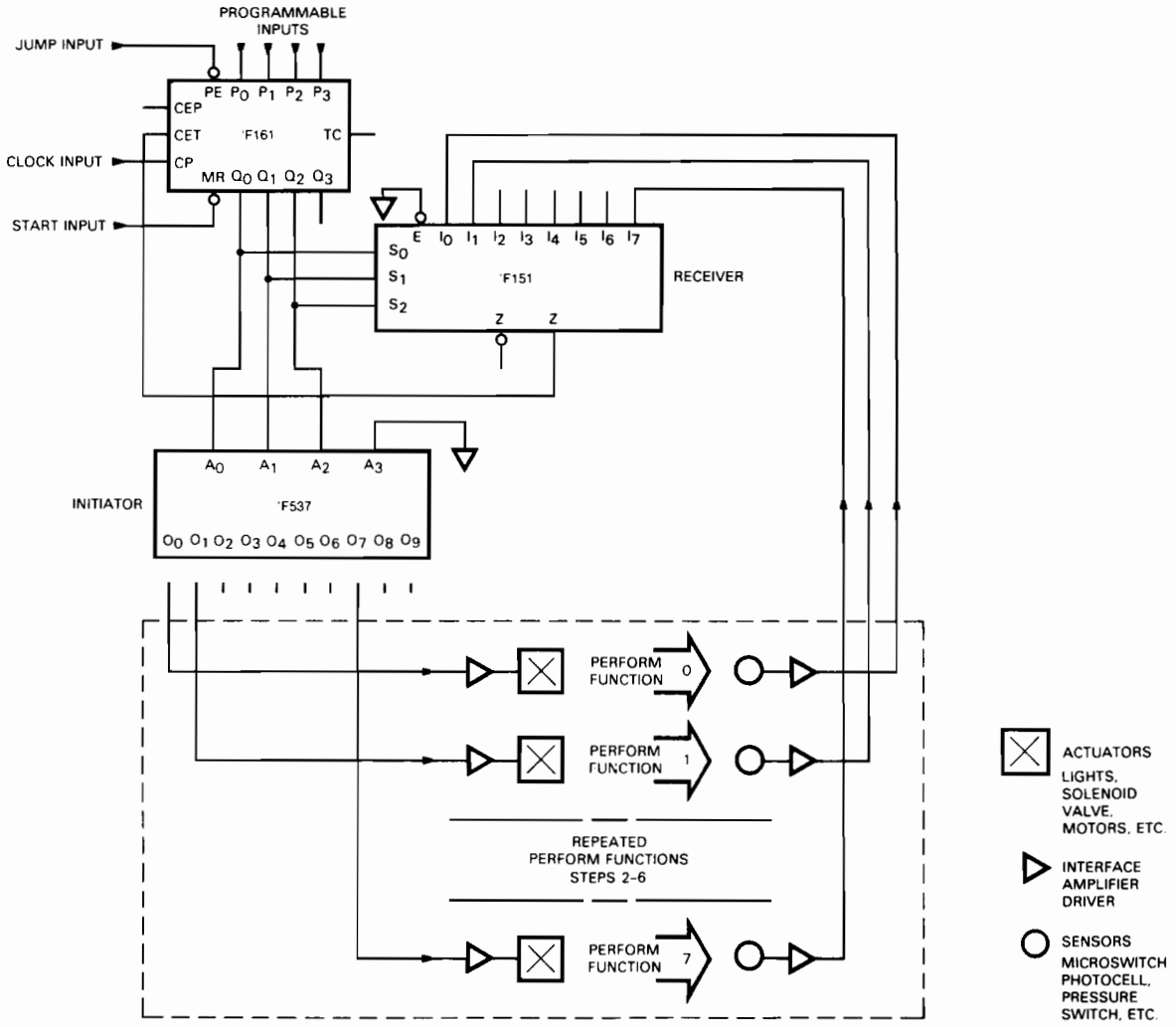


Figure 2-14

In this circuit, the 'F151 multiplexer is used as part of a control sequencer which steps through eight states, each initiating a test. The controller advances only when the test result is positive. It can also cycle through any portion of the sequence or jump, conditionally or unconditionally, to any other step.

The 'F161 counter is the modulo 8 program counter; its state is decoded by the O_0-O_7 outputs of the 'F537 which initiate the program steps. The counter also addresses the 'F151 multiplexer which acts as a receiver. When the 'F151 selected input is LOW, the test indicates that the program step is not yet completed. When the input is HIGH, the test indicates completion of the program step.

The active LOW Start pulse initiates the sequence by resetting the counter. This state is decoded and activates the 'F537 O_0 (step zero). This step lasts until the 'F151 I_0 is activated which in turn activates the count Enable input of the 'F161 counter, causing it to advance on the next LOW-to-HIGH clock transition. The next decoder output is then activated and this state lasts until the 'F151 I_1 is activated, etc.

When the multiplexer inputs are tied HIGH permanently, the sequencer advances to the next state on each subsequent clock pulse. Conditional or unconditional jumps can be made by activating the counter $\overline{PE}$ input with one of the decoder outputs and feeding the destination address into the counter's programmable inputs.

The system can be made asynchronous by using the $\overline{Z}$ output of the multiplexer as the clock source for the counter, which then advances whenever the multiplexer input goes from HIGH to LOW. This system is easily expanded to sixteen or more control steps.

'F251 8-Input Multiplexer Applications

Digital 32-Input Multiplexer

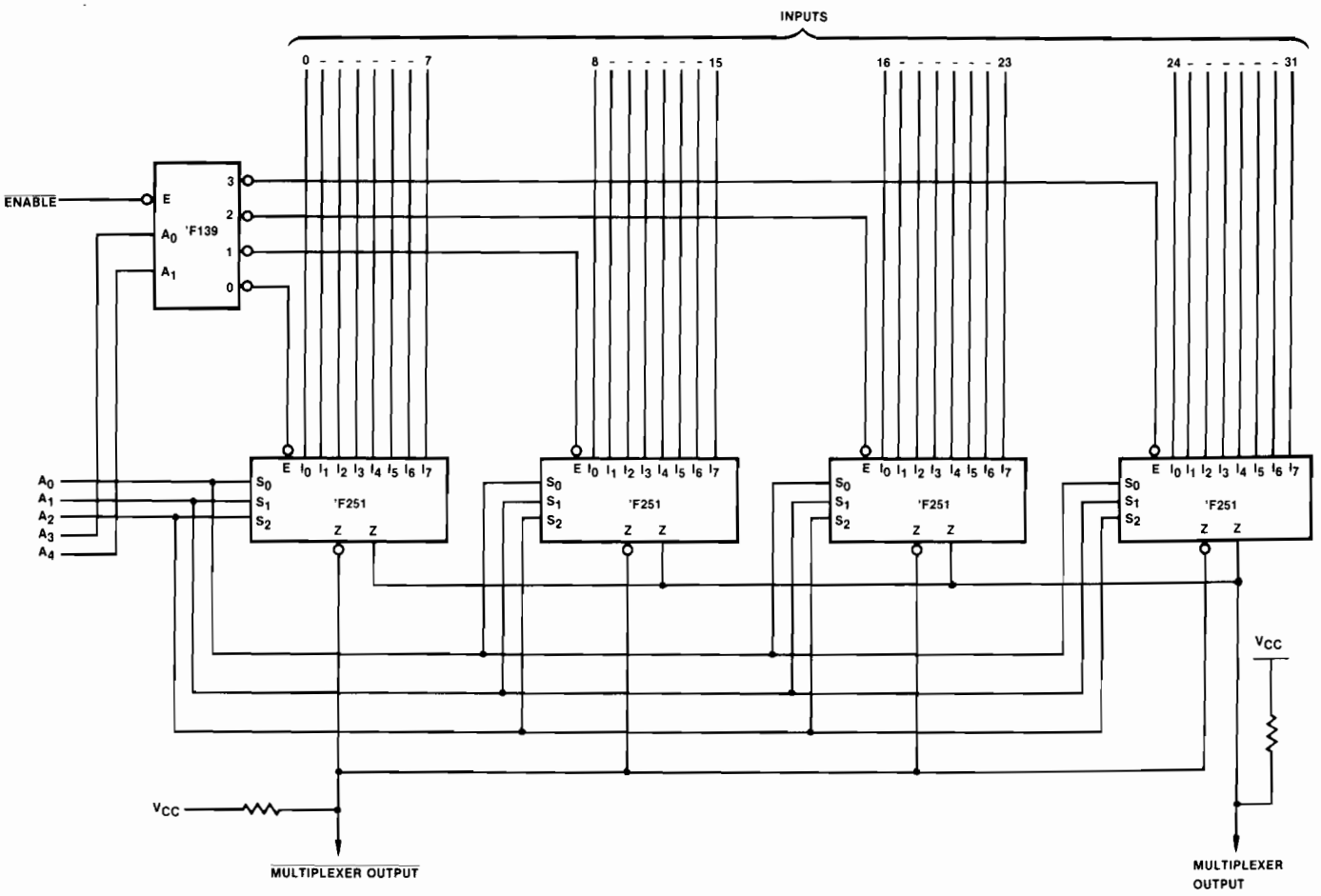


Figure 2-15

This 32-input digital multiplexer uses the 'F251 3-state device with the $\overline{Z}$ and Z outputs OR-tied. The pull-up resistors are optional.

Multiplexers as Function Generators

In most digital systems there are areas, usually in the control section, where a number of inputs generate an output in a highly irregular way. In other words, an unusual function must be generated which is apparently not available as an MSI building block.

In such cases, many designers tend to return to classical methods of logic design with NAND and NOR gates, using Boolean Algebra, Karnaugh maps and Veitch diagrams for logic minimization. Surprisingly enough, multiplexers can simplify these designs.

- The 'F157, 'F158, 'F257, 'F258, 'F398, 'F399 quad 2-input multiplexers can generate any four of the sixteen different functions of two variables.
- The 'F153, 'F253, 'F352, and 'F353 dual 4-input multiplexers can generate any two of the 256 different functions of three variables.
- The 'F151 and 'F251 8-input multiplexers can generate any one of the 65,536 different functions of four variables.

If a function has a certain regularity, adders or a few simple gates are possibly more economical. However, for a completely random function the multiplexer approach is more economical, certainly more compact and flexible, and easier to design.

Function generation with multiplexers is best explained with examples. An 8-input multiplexer such as the 'F151 can obviously generate any possible function of three variables. The desired function is written as a truth table. The variables A, B, and C are applied to the Select inputs S_0, S_1, and S_2 and the eight inputs are connected to either a HIGH or a LOW level, according to the truth table. This method is simple, but inefficient.

Figure 2-16

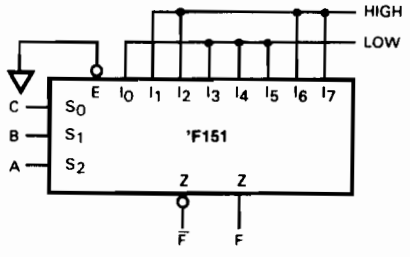


Table 2-7 Truth Table

A	B	C	F
L	L	L	L
L	L	H	H
L	H	L	H
L	H	H	L
H	L	L	L
H	L	H	L
H	H	L	H
H	H	H	H

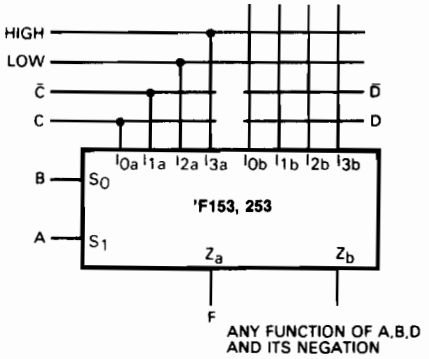
H = HIGH Voltage Levels
L = LOW Voltage Levels

The same function can also be generated by one half of a dual 4-input multiplexer such as the 'F153. For this purpose, the truth table is divided into four blocks as shown. Within each block, inputs A and B are constant, but output F can exhibit any of four characteristics:

- LOW for both input codes independent of C;
- HIGH for both input codes independent of C;
- Identical to C;
- Identical to $\overline{C}$.

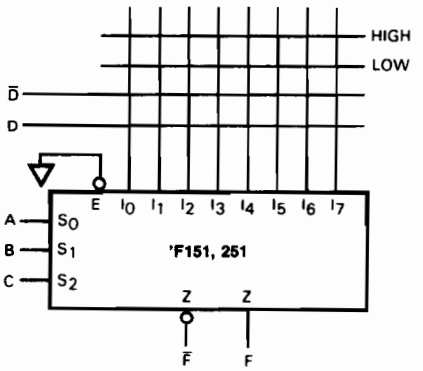
Therefore, the function can be implemented by a 4-input multiplexer, using the input variables A and B as Select inputs S_0 and S_1, and feeding the appropriate input with one of four signals: either a HIGH, a LOW, or the input variables C or $\overline{C}$. The other half of the 'F153 can be used to generate any other function of the variables A, B, and any third variable, not necessarily C.

Figure 2-17



The same reasoning can be applied to a function of four variables:

Figure 2-18



An 8-input multiplexer such as the 'F151 can generate any of the 65,536 (2^{16}) possible functions of the four variables A, B, C, and D.

Full Adder, Full Subtractor

Table 2-8
Full Adder (A plus B plus C_{IN})

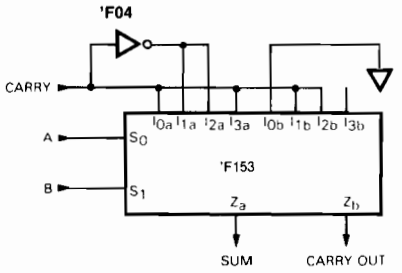
Inputs			Outputs	
A	B	C_{IN}	S	C_O
L	L	L	L	L
L	L	H	H	L
H	L	L	H	L
H	L	H	L	H
L	H	L	H	L
L	H	H	L	H
H	H	L	L	H
H	H	H	H	H

H = HIGH Voltage Levels
L = LOW Voltage Levels

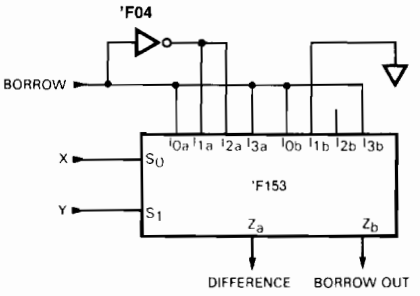
Table 2-9
Full Subtractor (X minus Y minus B_{IN})

Inputs			Outputs	
X	Y	B_{IN}	D	B_{OUT}
L	L	L	L	L
L	L	H	H	H
H	L	L	H	L
H	L	H	L	L
L	H	L	H	H
L	H	H	L	H
H	H	L	L	L
H	H	H	H	H

H = HIGH Voltage Levels
L = LOW Voltage Levels



a.



b.

Figure 2-19

An 'F153 dual 4-input multiplexer can implement any two functions of three variables. Therefore, it can be used as a full adder or as a full subtractor. These circuits demonstrate the versatility of the multiplexer as a function generator. However, the 'F283, 'F583, 'F385 and 'F582 adders are more efficient circuits for adding several bits in parallel. Note the inverted outputs are available.

Other Multiplexer Applications

Event Demultiplexer (Clock Routing)

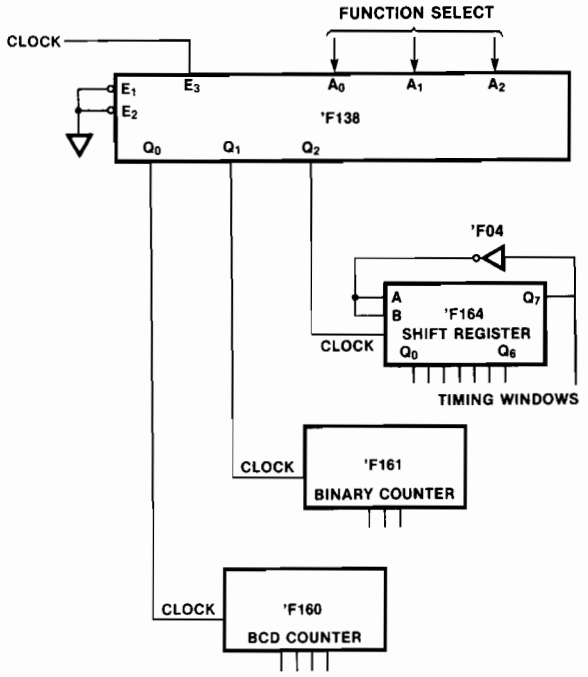


Figure 2-20

This is a circuit which may be used to distinguish up to eight separate events. It consists of an 'F138 decoder which is used to drive other circuit functions at a selected clock rate. In the circuit shown we are utilizing an 'F164 shift register in order to provide window duration timing which may be fed on to any number of applications. Also, an 'F161 binary counter is shown here as well as an 'F160 BCD counter. The outputs of these devices, of course, depend on the address which is presented to the 'F138 decoder/demultiplexer, and the timing is dependent on the clock frequency. Another way of describing this array would be to look at it as a way of distributing and/or dividing the clock. Further timing definition may be output via the counters, but it is intended here to illustrate a method of routing the clock in order to obtain various functions.

Programmable Divider

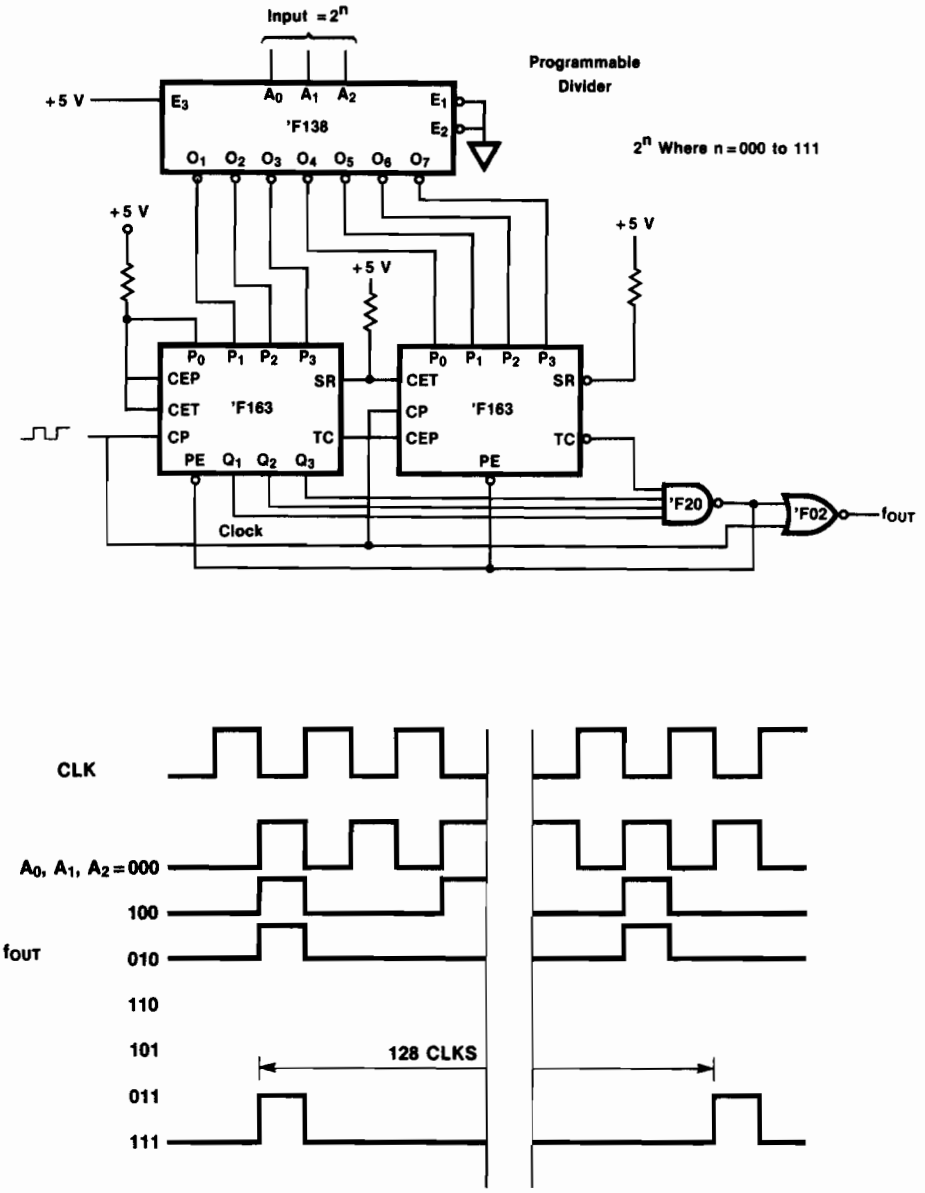


Figure 2-21

This circuit demonstrates a method of forming a programmable divider. It uses one 'F138 to establish the programmed input pattern for two 'F163 synchronous presettable binary counters. This saves on the number of lines required for the division ratio shown. The 'F163s along with an 'F20 NAND gate and an 'F02 NOR gate form the programmed output waveform. In the circuit, eight output waveforms are available. Rates as slow as 1 pulse in 128 clocks are possible. A BCD input to A_0, A_1, and A_2 of zero to seven determines the output pulse repetition rate. The output pulse width is determined by the clock pulse width but due to the use of the 'F02, it is phase shifted 180° with respect to the clock. All unused inputs on the 'F163s must be tied HIGH to prevent erroneous triggering.

FAIRCHILD
A Schlumberger Company

Decoders

Introduction

There are two categories of decoders, logic decoders and display decoder/drivers. Logic decoders are MSI devices controlled by an address. They select and activate a particular output as specified by the address. Display decoders and display decoder/drivers generate numeric codes (e.g., 7-segment) and then provide the codes to a driver or drive the displays directly.

Logic decoders are discussed below and are available in many configurations. They are used extensively in the selective addressing structures of memory systems, for data or clock routing, demultiplexing and as minterm generators in random and control logic.

Display decoder/drivers are comparatively slow devices not applicable to high-speed logic families such as FAST. These decoder/drivers tend to be special interface devices constructed in linear or CMOS technologies, and will not be discussed here.

'F139, 'F539 Dual 1-of-4 Decoder

Description and Operation

The 'F139 is two separate decoders, each designed to accept two binary weighted inputs and provide four mutually exclusive active LOW outputs as shown in the logic diagram. Each decoder can become a 4-output demultiplexer by using the Enable as a data input.

The delay from address to output is equivalent to either two or three gate delays, depending on the order of address bits that are changing simultaneously. The delay from Enable to output is always two gate delays, typically 5ns.

The 'F539 is similar to the 'F139 but offers greater flexibility in that the output polarity can be selected by an additional input to each section, and a separate output Enable is provided to each section, providing expandability through 3-state control. Because of this increased complexity, the 'F539 exhibits greater propagation delays than the 'F139. All applications demonstrate the 'F139. If desired, the 'F539 could be substituted to take advantage of its additional features.

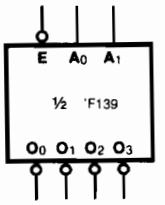


Figure 3-1

Table 3-1 Truth Table

Inputs			Outputs			
$\overline{E}$	A_0	A_1	$\overline{O}_0$	$\overline{O}_1$	$\overline{O}_2$	$\overline{O}_3$
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

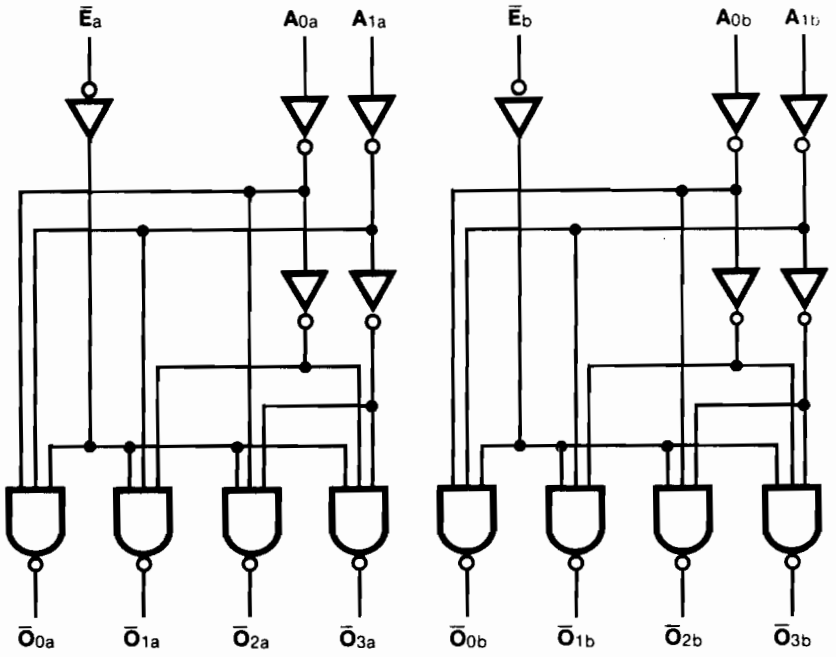


Figure 3-2 Logic Diagram 'F139

'F139 Dual 1-of-4 Decoder Applications

Decoding

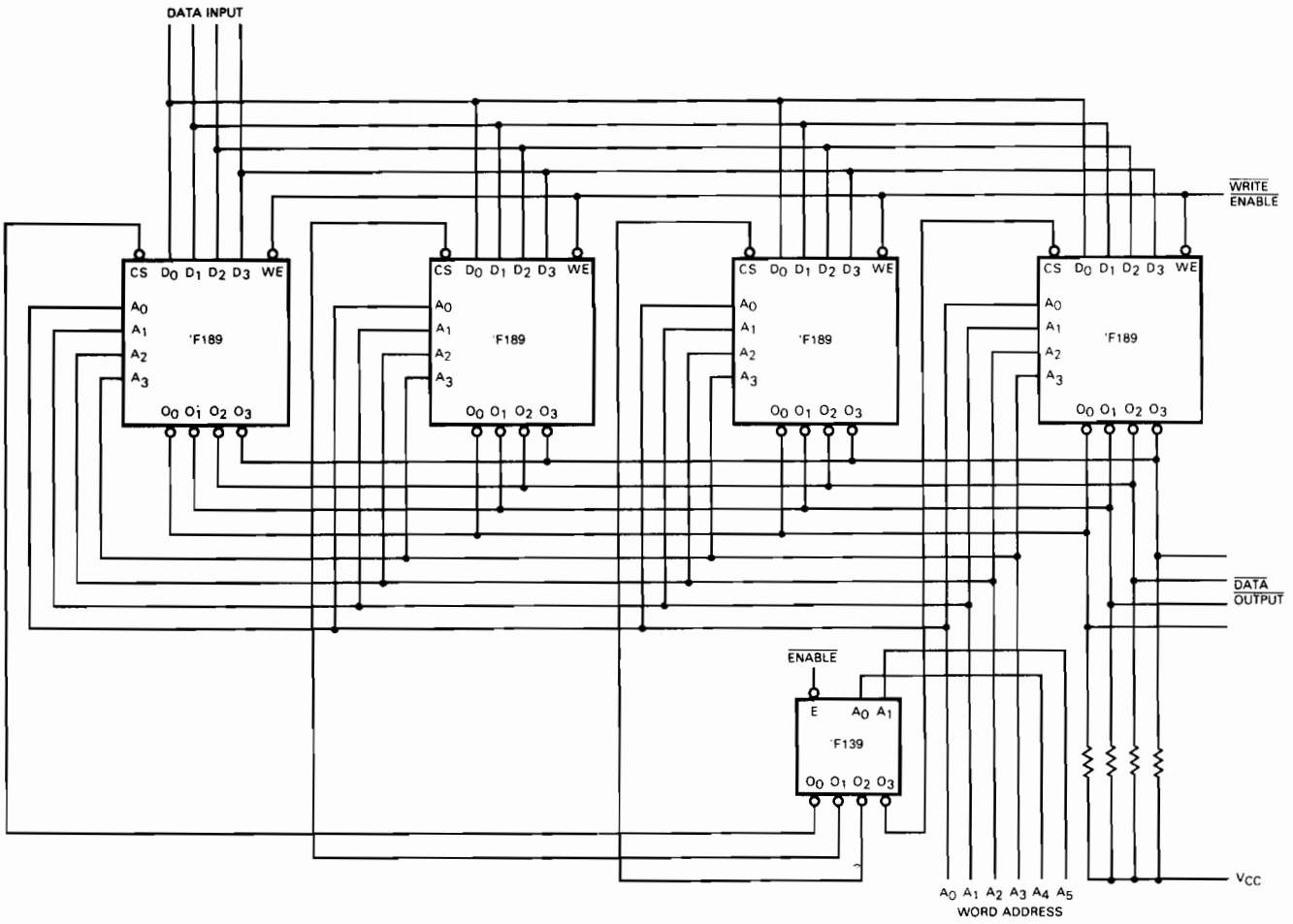


Figure 3-3

The most obvious use of the 'F139 is in logic decoding and memory addressing. As shown, the decoder supplies the extra decoding necessary to address a word in a 64-word memory. One 1-of-4 decoder is used to decode the two most significant bits of memory address and to enable the appropriate memory units. The four least significant bits are decoded on the 'F189. The high fanout capability of the 'F139 allows it to drive sixteen 'F189 memory units with a word length of 64 bits without additional buffers.

Demultiplexing

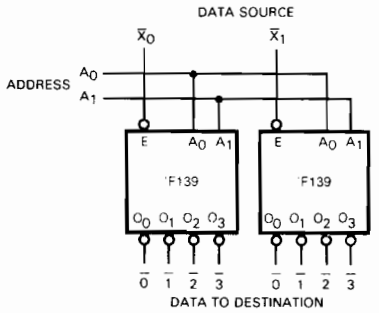


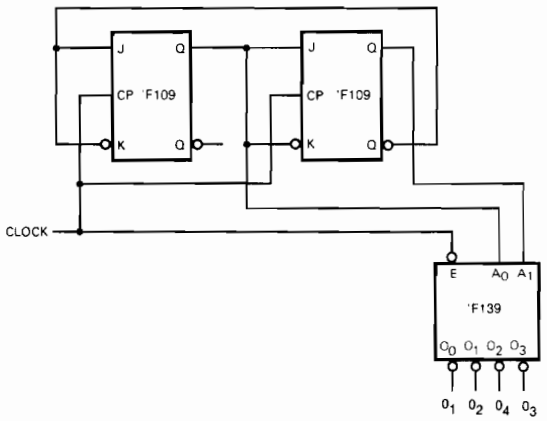
Figure 3-4

Table 3-2 Truth Table

A_0	A_1	Output Selected
0	0	0
0	1	1
1	0	2
1	1	3

The 'F139 can be used as a demultiplexer, routing data from a single source to a destination chosen by the applied address. The data is applied through the Enable and routed without inversion to the output specified by the address inputs A_0 and A_1. All unselected outputs remain HIGH. For example, with both address inputs HIGH, output 3 follows the state of the Enable input: LOW when the Enable is LOW and HIGH when the Enable is HIGH. Demultiplexing can be employed for either data routing or clock distribution. A 2-bit data demultiplexer is shown. Two bits of active LOW data are routed to the outputs selected by the applied address as shown in the table above.

4-Phase Clock Generator



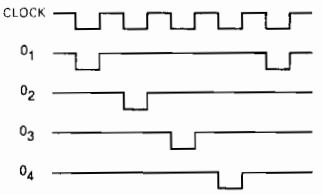
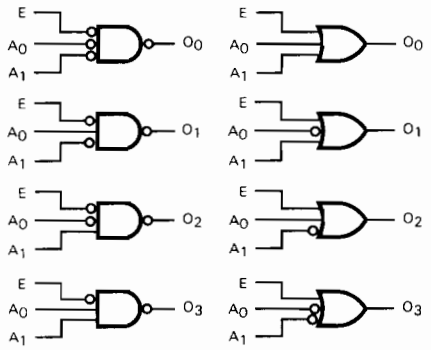


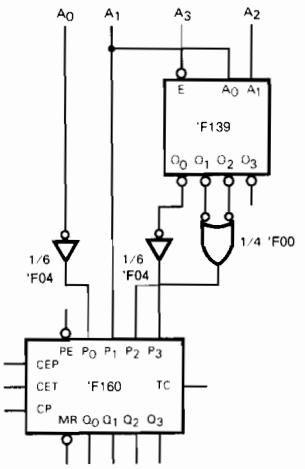
Figure 3-5

Clock demultiplexing for clock distribution and generation is readily accomplished with the 'F139. This is a 4-phase clock generator producing non-overlapping clock pulses for TTL circuitry. Note that the Enable is used as the clock input, eliminating glitches by framing address changes which occur when the flip-flops, registers or counters change state on the rising clock edge.

Function Generation



a.



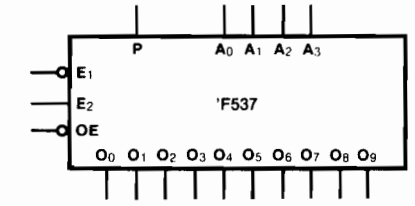
b.

Figure 3-6

Each half of the 'F139 generates all four minterms of two variables. These four minterms are useful in some applications, replacing logic functions and thereby reducing the number of packages required in a logic network. Gate functions which can be replaced by the 'F139 are shown in Figure 3-6a; Figure 3-6b illustrates a nines complement circuit utilizing these gate functions.

'F537 Decoder

Description and Operation



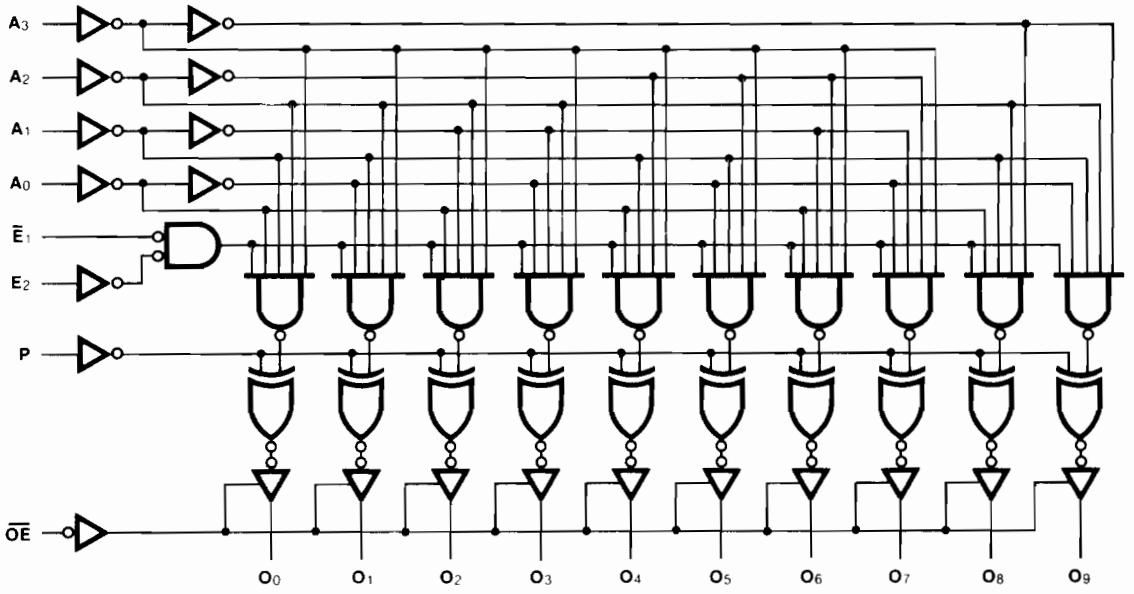


Figure 3-7 Logic Diagram

The 'F537 1-of-10 decoder is a high-speed, complex function integrated circuit suitable for use in high-speed digital equipment. This decoder has four inputs which act as an address to produce an output at the corresponding output terminal, and two input Enable lines (one active HIGH, one active LOW). The Polarity Select Line determines active HIGH or LOW outputs. An active LOW control line enables the 3-state outputs. It has high speed and excellent noise margins with low power consumption.

The 'F537 1-of-10 decoder accepts four active BCD inputs and provides ten mutually exclusive active outputs. All outputs are inactive when binary codes greater than nine are applied to the inputs.

The most significant address input, A_3, may function as active LOW Enable or active HIGH Inhibit input when the circuit is used as a 1-of-8 decoder, or it can be used as the data input for a 8-output demultiplexer.

When the 'F537 is used as a 1-of-8 decoder or demultiplexer, the three address inputs A_0-A_2 can be interchanged and/or considered active LOW, provided the outputs are relabeled appropriately. This may simplify printed circuit board layout.

'F537 Decoder Applications

Demultiplexing

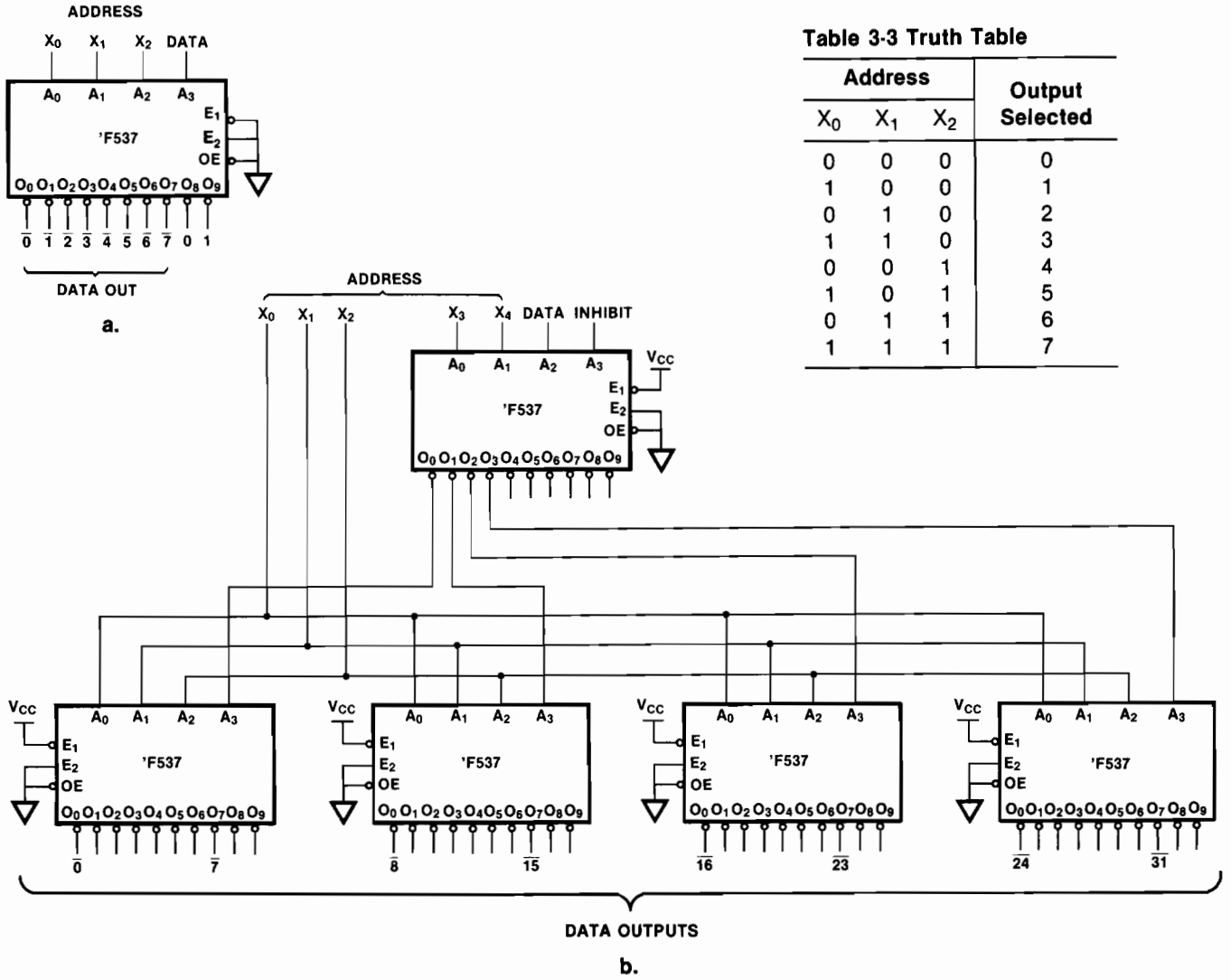


Table 3-3 Truth Table

Address			Output Selected
X_0	X_1	X_2	
0	0	0	0
1	0	0	1
0	1	0	2
1	1	0	3
0	0	1	4
1	0	1	5
0	1	1	6
1	1	1	7

Figure 3-8

The 'F537 decoder can be used as an 8-output demultiplexer (Figure 3-8a). The first three inputs select the appropriate output, and the logic level of the signal on the A_3 terminal determines its polarity. Thus, data on the A_3 input is switched to the output terminal selected by the address, $A_0A_1A_2$. The last two outputs, 8 and 9, are the complements of the first two outputs, 0 and 1; note that data is not inverted when switched from the A_3 terminal to the selected output. The A_2 input of the decoder can also be used as a data input. In this mode, the A_3 terminal becomes an active HIGH Inhibit and inputs A_0 and A_1 are a 2-bit address. Decoder outputs 0, 1, 2 and 3 are the assertion outputs of the demultiplexer and decoder outputs 4, 5, 6 and 7 are corresponding complements.

The multistage decoding scheme (Figure 3-8b) can also be used for demultiplexers requiring a large number of output channels. This design shows a 32-output demultiplexer. One decoder has two inputs to produce four active LOW outputs, which are then used to select one of other four decoders. The remaining inputs of the first decoder are used as Data and Inhibit inputs.

Minterm Generation

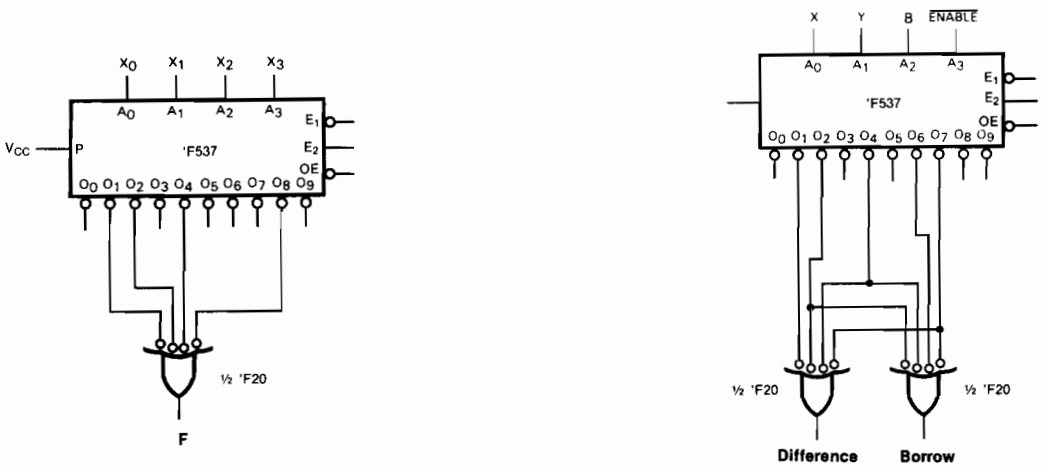


Figure 3-9

The 'F537 decoder can function as an active LOW or HIGH output minterm generator producing the first ten minterms of the sixteen possible from four variables. The appropriate minterms can be summed with the use of an active LOW or HIGH input OR. The state of the P input determines the active states of the outputs.

This technique is suitable for all types of control, sequencing and decoding logic, and can considerably simplify the problem of generating a required sequence of outputs on a set of lines.

The circuits above show: in Figure 3-9a a single one detector (output F is HIGH whenever one and only one of the X_0-X_3 inputs is HIGH); and, in Figure 3-9b a gated full subtractor generating the Difference and the Borrow outputs of variables X-Y-B when the Enable input is LOW. See the Multiplexer section for use of multiplexers as function generators.

Table 3-4 Input-Output Table

Input Sequence				Input Code Value															
A_0	A_1	A_2	A_3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	4	8	0	1	2	3	4	5	6	7	8	9	–	–	–	–	–	–
1	2	8	4	0	1	2	3	8	9	–	–	4	5	6	7	–	–	–	–
1	4	2	8	0	1	4	5	2	3	6	7	8	9	–	–	–	–	–	–
1	4	8	2	0	1	8	9	2	3	–	–	4	5	–	–	6	7	–	–
1	8	2	4	0	1	4	5	8	9	–	–	2	3	6	7	–	–	–	–
1	8	4	2	0	1	8	9	4	5	–	–	2	3	–	–	6	7	–	–
2	1	4	8	0	2	1	3	4	6	5	7	8	–	9	–	–	–	–	–
2	1	8	4	0	2	1	3	8	–	9	–	4	6	5	7	–	–	–	–
2	4	1	8	0	4	1	5	2	6	3	7	8	–	9	–	–	–	–	–
2	4	8	1	0	8	1	9	2	–	3	–	4	–	5	–	6	–	7	–
2	8	1	4	0	4	1	5	8	–	9	–	2	6	3	7	–	–	–	–
2	8	4	1	0	8	1	9	4	–	5	–	2	–	3	–	6	–	7	–
4	1	2	8	0	2	4	6	1	3	5	7	8	–	–	–	9	–	–	–
4	1	8	2	0	2	8	–	1	3	9	–	4	6	–	–	5	7	–	–
4	2	1	8	0	4	2	6	1	5	3	7	8	–	–	–	9	–	–	–
4	2	8	1	0	8	2	–	1	9	3	–	4	–	6	–	5	–	7	–
4	8	1	2	0	4	8	–	1	5	9	–	2	6	–	–	3	7	–	–
4	8	2	1	0	8	4	–	1	9	5	–	2	–	6	–	3	–	7	–
8	1	2	4	0	2	4	6	8	–	–	–	1	3	5	7	9	–	–	–
8	1	4	2	0	2	8	–	4	6	–	–	1	3	9	–	5	7	–	–
8	2	1	4	0	4	2	6	8	–	–	–	1	5	3	7	9	–	–	–
8	2	4	1	0	8	2	–	4	–	6	–	1	9	3	–	5	–	7	–
8	4	1	2	0	4	8	–	2	6	–	–	1	5	9	–	3	7	–	–
8	4	2	1	0	8	4	–	2	–	6	–	1	9	5	–	3	–	7	–
				Activated Outputs															

– = No active output

The 'F537 decodes ten of the possible sixteen minterms of the four variables applied to the inputs A_0-A_3. The table above shows how different groups of minterms can be decoded by changing the assignment of the 1248 input signals to the A_0-A_3 address inputs. Interconnection of outputs can generate many different functions of four variables.

'F138, 'F538, 'F547, and 'F548 1-of-8 Decoder Family

This family of four FAST parts decodes three inputs into one of eight output lines. The 'F138 is the basic 1-of-8 decoder pictured below.

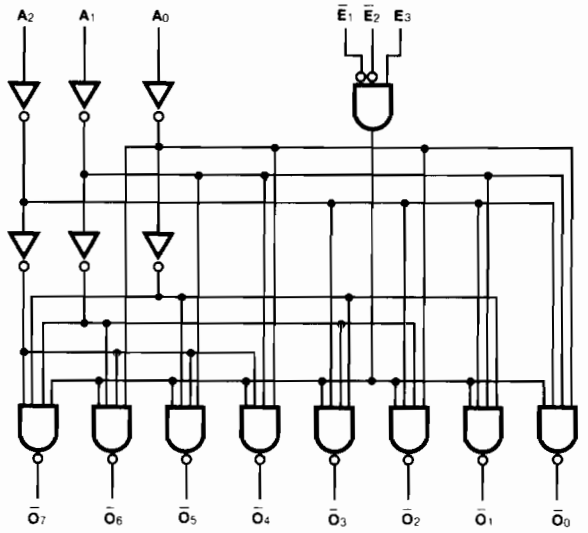


Figure 3-10 Logic Diagram 'F138

Description

The 'F138 high-speed 1-of-8 decoder/multiplexer accepts three binary weighted inputs (A_0, A_1, A_2) and, when enabled, provides eight mutually exclusive active LOW outputs ($\overline{O}_0$-$\overline{O}_7$). The 'F138 features three Enable inputs, two active LOW ($\overline{E}_1$,$\overline{E}_2$) and one active HIGH (E_3). All outputs will be HIGH unless $\overline{E}_1$ and $\overline{E}_2$ are LOW and E_3 is HIGH. This multiple enable function allows easy parallel expansion of the device to a 1-of-32 (5 lines to 32 lines) decoder with just four 'F138 devices and one inverter. The 'F138 can be used as an 8-output demultiplexer by using one of the active LOW Enable inputs as the data input and the other Enable inputs as strobes. The Enable inputs which are not used must be permanently tied to their appropriate active HIGH or active LOW state.

The 'F538 provides additional flexibility in that a 20-pin package is employed to give an additional Enable pin, two output (3-state) control pins, and an input Polarity pin to determine the parity of throughput signal. The 'F538 logic diagram is pictured in Figure 3-11 with a corresponding truth table.

Table 3-5 Truth Table

Inputs						Outputs							
$\overline{E}_1$	$\overline{E}_2$	E_3	A_0	A_1	A_2	$\overline{O}_0$	$\overline{O}_1$	$\overline{O}_2$	$\overline{O}_3$	$\overline{O}_4$	$\overline{O}_5$	$\overline{O}_6$	$\overline{O}_7$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

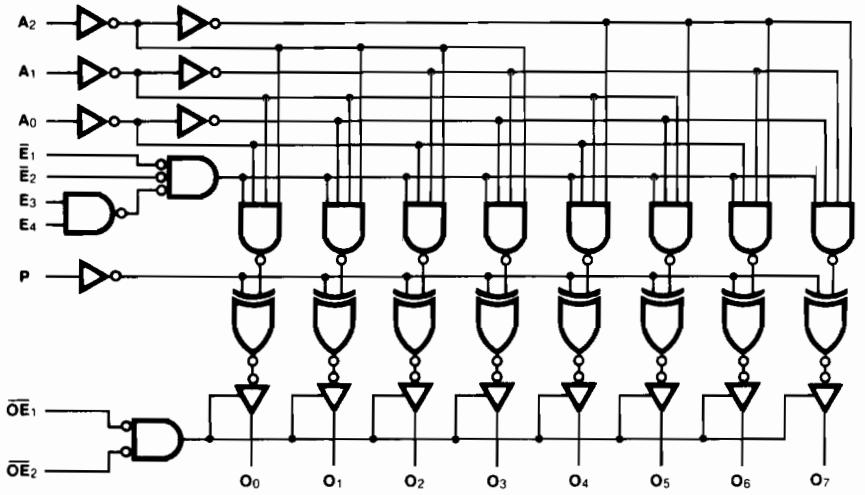


Figure 3-11 Logic Diagram 'F538

Table 3-6 Truth Table

Function	Inputs									Outputs							
	$\overline{OE}_1$	$\overline{OE}_2$	$\overline{E}_1$	$\overline{E}_2$	E_3	E_4	A_2	A_1	A_0	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
High Impedance	H	X	X	X	X	X	X	X	X	Z	Z	Z	Z	Z	Z	Z	Z
	X	H	X	X	X	X	X	X	X	Z	Z	Z	Z	Z	Z	Z	Z
Disable	L	L	H	X	X	X	X	X	X	Outputs Equal P Input							
	L	L	X	H	X	X	X	X	X								
	L	L	X	X	L	X	X	X	X								
	L	L	X	X	X	L	X	X	X								
Active HIGH Output (P = L)	L	L	L	L	H	H	L	L	L	H	L	L	L	L	L	L	L
	L	L	L	L	H	H	L	L	H	L	H	L	L	L	L	L	L
	L	L	L	L	H	H	L	H	L	L	L	H	L	L	L	L	L
	L	L	L	L	H	H	L	H	H	L	L	L	H	L	L	L	L
	L	L	L	L	H	H	H	L	L	L	L	L	L	H	L	L	L
	L	L	L	L	H	H	H	L	H	L	L	L	L	L	H	L	L
	L	L	L	L	H	H	H	H	L	L	L	L	L	L	L	H	L
	L	L	L	L	H	H	H	H	H	L	L	L	L	L	L	L	H
Active LOW Output (P = H)	L	L	L	L	H	H	L	L	L	L	H	H	H	H	H	H	H
	L	L	L	L	H	H	L	L	H	H	L	H	H	H	H	H	H
	L	L	L	L	H	H	L	H	L	H	H	L	H	H	H	H	H
	L	L	L	L	H	H	L	H	H	H	H	H	L	H	H	H	H
	L	L	L	L	H	H	H	L	L	H	H	H	H	L	H	H	H
	L	L	L	L	H	H	H	L	H	H	H	H	H	H	L	H	H
	L	L	L	L	H	H	H	H	L	H	H	H	H	H	H	L	H
	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = High Impedance

Read Only Memory Control

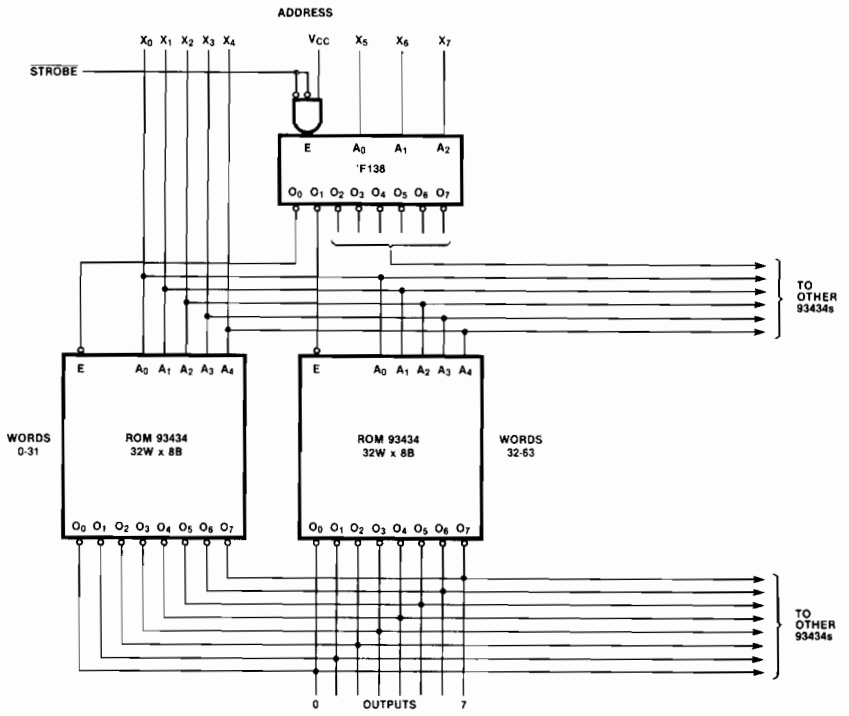
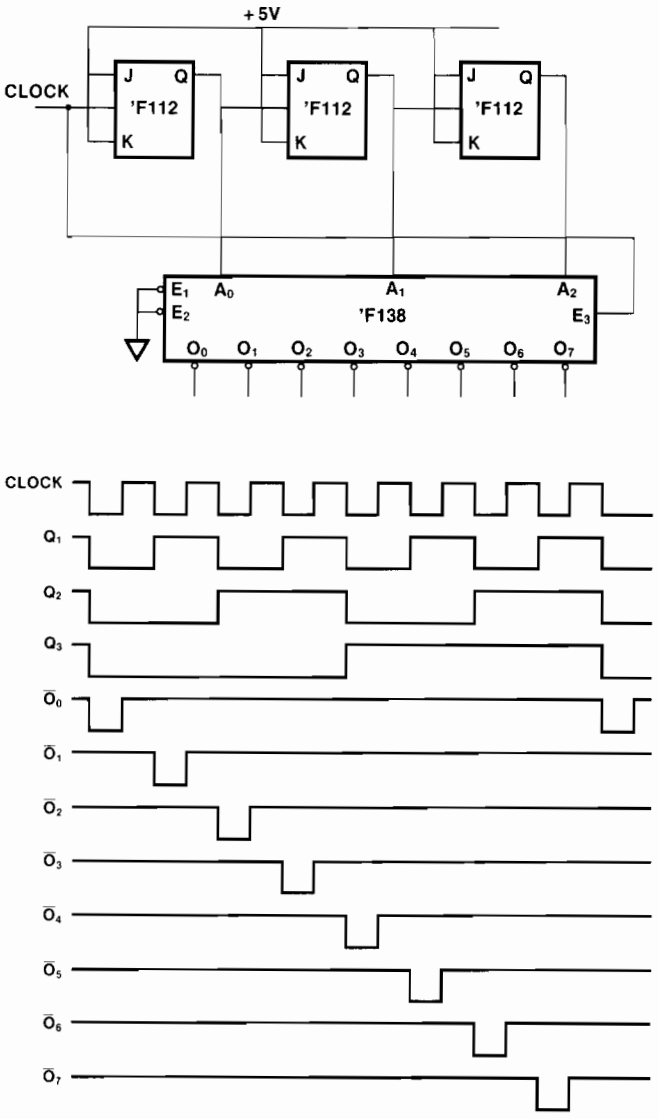


Figure 3-12

An 'F138 decoder can select a particular memory from a group of memories (93434s) comprising a complete stack. A single 'F138 can control a group of eight of these memories. Each memory contains 256 bits arranged in a 32-word x 8-bit format, with the outputs of common digits OR-tied. An output from the 'F138 can drive up to ten 93434 memories, allowing control of a 256-word x 80-bit memory or 20,480 bits of information with a single decoder.

8-Phase Clock Generator



Using an 'F138 1-of-8 decoder/demultiplexer and two 'F112 dual JK negative edge-triggered flip-flop packages, it is possible to build a clock generator which can be used as a synchronous counter, an eight event synchronous timer as well as other applications. In the circuit, the 'F112s have their J-K inputs tied HIGH at a 5V TTL level for continuous toggling. All Q outputs switch LOW on the first negative-going edge of the clock input, the JK flip-flop can only be toggled if both J and K inputs are HIGH before the negative edge of the clock pulse.

The clock pulse is connected to the $\overline{E}_3$ input of the 'F138. In this manner the clock pulse width determines the 'F138 output pulse width by enabling the 'F138 output during the clock low duration time only. In this circuit, $\overline{E}_1$ and $\overline{E}_2$ are tied to ground. A_0, A_1, and A_2 input are binary and the outputs are glitchless.

Figure 3-13

Table 3-7 Truth Table

	Inputs			Outputs							
∅	A_0	A_1	A_2	$\overline{O}_0$	$\overline{O}_1$	$\overline{O}_2$	$\overline{O}_3$	$\overline{O}_4$	$\overline{O}_5$	$\overline{O}_6$	$\overline{O}_7$
0	0	0	0	0	1	1	1	1	1	1	1
1	1	0	0	1	0	1	1	1	1	1	1
2	0	1	0	1	1	0	1	1	1	1	1
3	1	1	0	1	1	1	0	1	1	1	1
4	0	0	1	1	1	1	1	0	1	1	1
5	1	0	1	1	1	1	1	1	0	1	1
6	0	1	1	1	1	1	1	1	1	0	1
7	1	1	1	1	1	1	1	1	1	1	0

Clock - HIGH to LOW

16-Phase Clock Generator

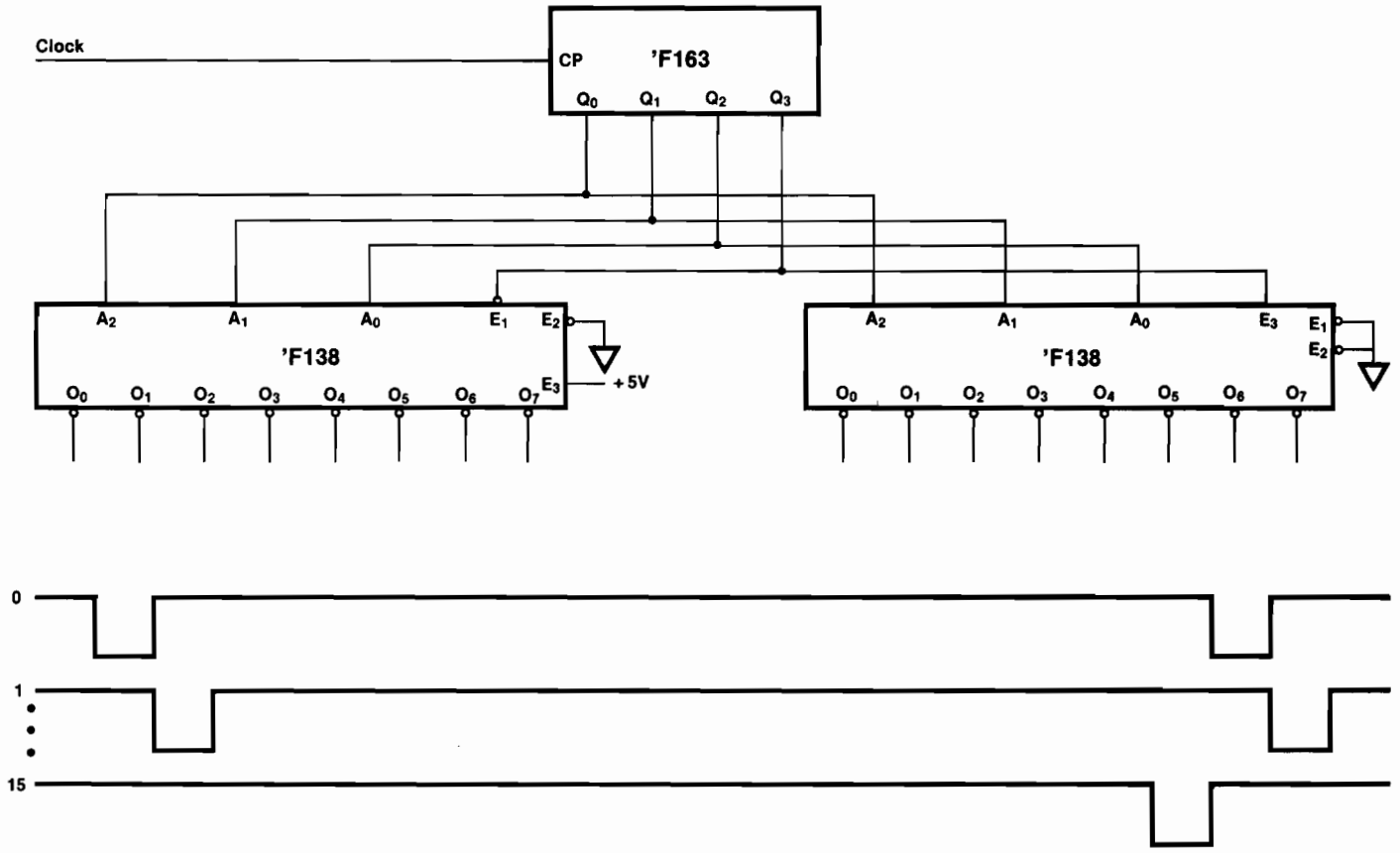


Figure 3-14

This 16-phase clock generator is built using an 'F163 and two 'F138s. The 'F163 flip-flop outputs are clocked and fed in a parallel fashion to two 'F138 circuits. Opposite Enables are utilized on the 'F138s so that during count zero through seven only one decoder is enabled while during count eight through fifteen the second 'F138 is enabled and the first is utilized to establish the one of sixteen. Unused inputs should be tied to their appropriate logic levels.

Programmable Counter

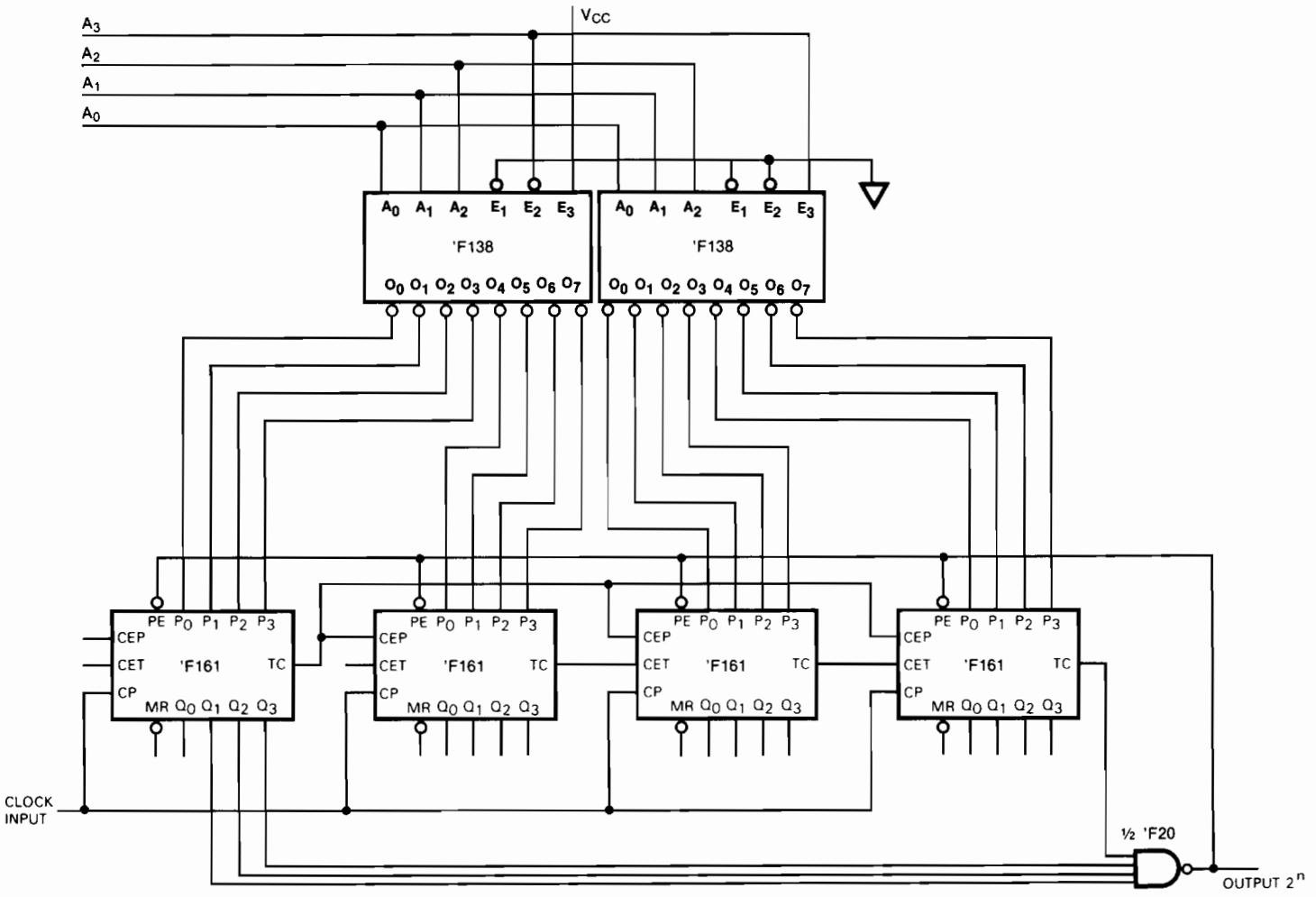


Figure 3-15

A programmable counter can be designed using a 'F138 decoder that counts in modulo 2^n, where n is the programmable input. Shown above is a 'F138 decoder and four 'F161 binary counters capable of counting up to 2^{15}. The input n drives the selected output LOW so that when a parallel load occurs, all HIGHs are written into the register except at the stage represented by the address n. The counter counts pulses and reaches the condition 1111111110000, at which point the terminal count of the last stage goes HIGH. After fourteen additional pulses bring the total to 2^{n-1}, the three remaining inputs to the 'F20 gate are HIGH, and the next clock pulse reloads the counter to its original condition. The circuit thus performs as a 2^n programmable divider.

Expansion to 1-of-32 Decoding

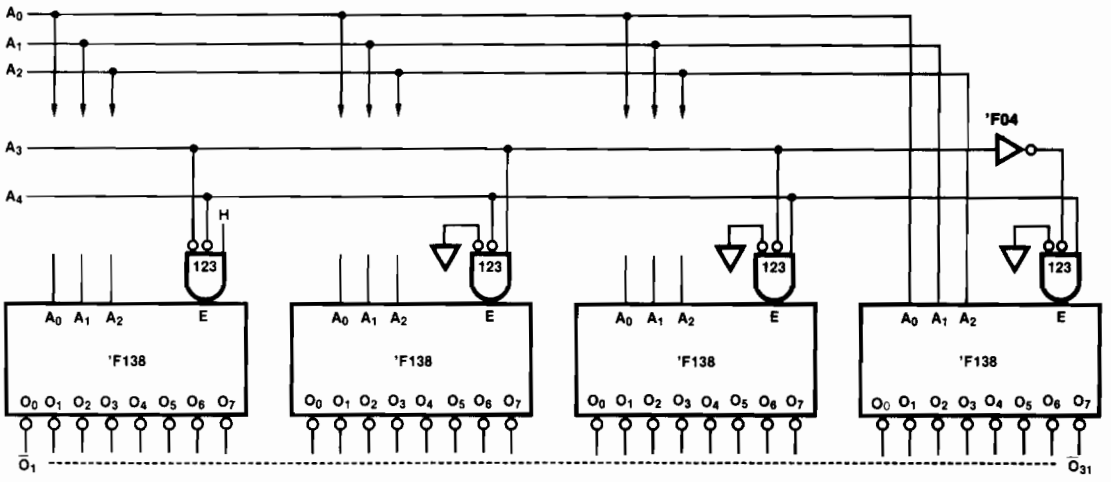


Figure 3-16

Four 'F138s can be expanded to a 1-of-32 decoder, as illustrated above, by the addition of one inverter. Note that this inverter is unnecessary with the 'F538 and 'F548 since they both possess two active HIGH Enables and two active LOW Enables.

Range Gate Strobe Generator

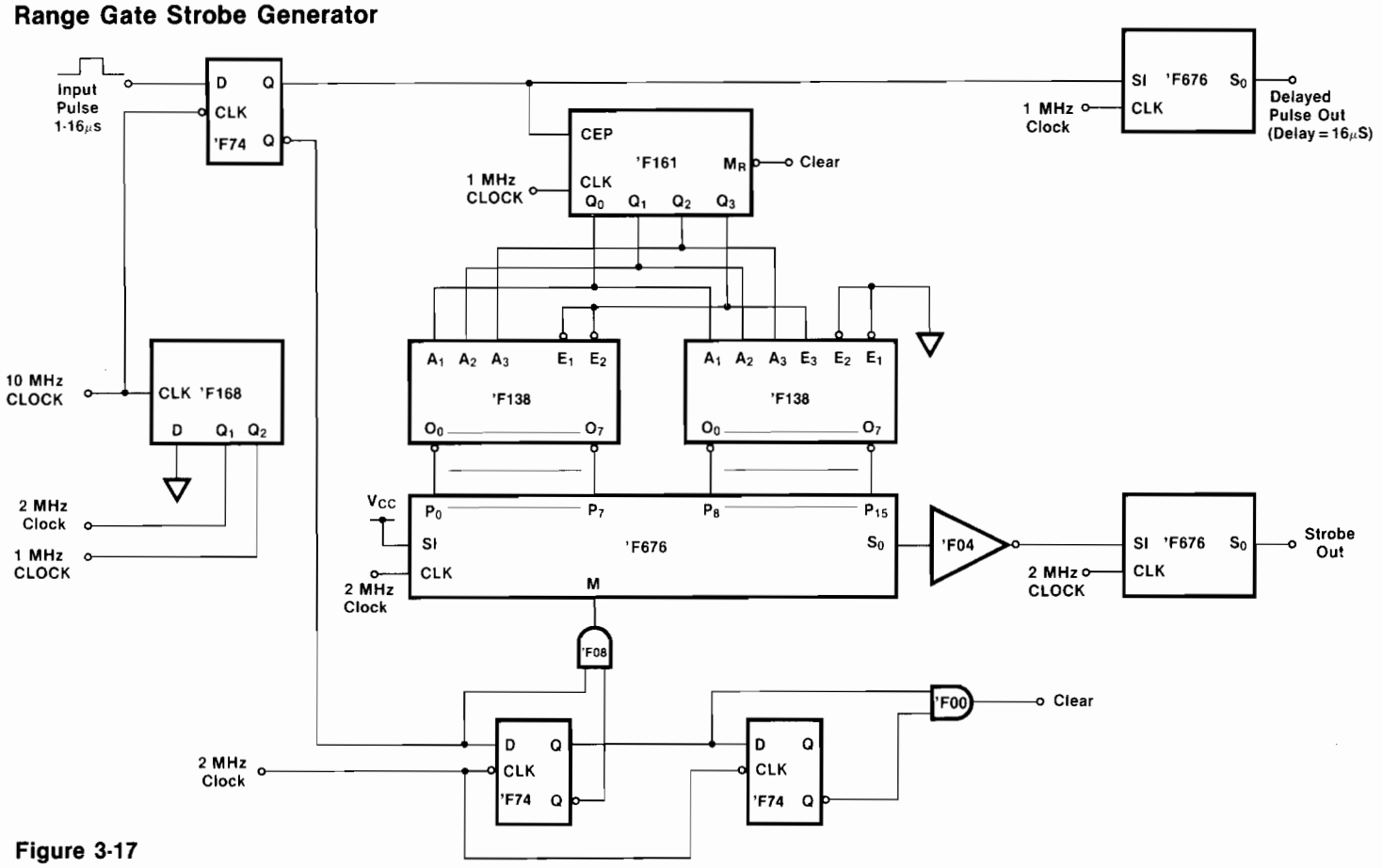


Figure 3-17

This circuit constitutes a method of generating a strobe pulse 500ns wide which is then placed in the center of an incoming pulse of 1 to 16μs in duration. The circuit was designed as a strobe generator for radar range gating. A 10MHz base clock must be provided in order to facillitate generation of a 1 and 2MHz quadrature clock signal and for proper circuit operation. In the circuit two shift registers are present. One is a 16-bit register fed with the 1MHz clock signal which delays the input pulse to 16μs. The second shift register is a 32-bit comprised of two 'F676 registers and clocked at a 2MHz rate. The left-hand 'F676 of the 32-bit register is input with the decoded center count provided by the 'F161 synchronous counter and the 1-of-16 decoder.

The input pulse is resynchronized by an 'F74 D flip-flop and one propagation delay later enables the 'F161 synchronous binary counter by setting the CEP input HIGH. The 'F161 counts at a 1MHz rate, determined by its clock frequency, as long as CEP is held HIGH. As soon as the input drops low and the next positive clock edge occurs, CEP goes LOW and the 'F161 discontinues its count. The input center point, having now been detected, is fed into a 1-of-16 decoder comprised of two 'F138s in a tandem configuration. The decoded output feeds the left-hand 'F676 which cascades with another 'F676 to form the 32-bit shift register. Here the S_1 input is tied to V_{CC} and may be ignored. When the input pulse goes LOW the Q output of the 'F74 signal resynchronizer goes HIGH on the subsequent positive edge of the clock signal. This signal, along with the 2MHz clock, is fed to a one-shot circuit comprised of an 'F74 and an 'F08 AND gate. It can be seen that when the one-shot output goes HIGH the mode select input (M) of the 'F676 is also HIGH. In this condition the information present on the parallel data inputs, which is the center position count data, is entered on the falling edge of the clock pulse input signal $\overline{CP}$. Now the serial output pin is enabled for 500ns. This is passed to the second 'F676 at a 2MHz rate and thereby is presented at the center of the delayed input pulse which appears as the output of the top 'F676.

A clean pulse is derived via the one-shot 'F74 and ensures that the 'F161 is set to all zeros and so is ready for the next input pulse.

It may also be noted here that 'F538s may be substituted for the 'F138s which would allow the designer to eliminate the one 'F04 hex inverter used for signal inversion.

FAIRCHILD
A Schlumberger Company

Encoders

Introduction

Encoders are circuits with many inputs that generate the address of the active input. If a system design guarantees only one encoder input active, the encoder logic is very simple and can be implemented with gates, as shown in Figure 4-1.

If several inputs can be active at one time, a simple encoder would generate the logic OR of their addresses, which is probably undesirable (i.e., inputs 2 and 4 active would generate address 6). A priority encoder generates the address of the active input with the highest priority. The priority is pre-assigned according to the position at the inputs. This chapter describes the 'F148 8-input priority encoder and some applications.

4

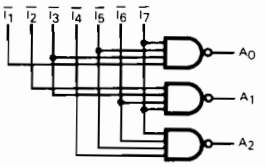


Figure 4-1

'F148 8-Input Priority Encoder

Description

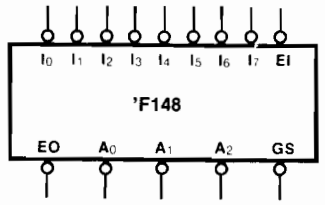


Figure 4-2

Table 4-1 Truth Table

Inputs									Outputs				
$\overline{EI}$	$\overline{I}_0$	$\overline{I}_1$	$\overline{I}_2$	$\overline{I}_3$	$\overline{I}_4$	$\overline{I}_5$	$\overline{I}_6$	$\overline{I}_7$	$\overline{GS}$	$\overline{A}_0$	$\overline{A}_1$	$\overline{A}_2$	$\overline{EO}$
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	H	L	L	H
L	X	X	X	X	X	L	H	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	L	L	L	H	H
L	X	X	L	H	H	H	H	H	L	H	L	H	H
L	X	L	H	H	H	H	H	H	L	L	H	H	H
L	L	H	H	H	H	H	H	H	L	H	H	H	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

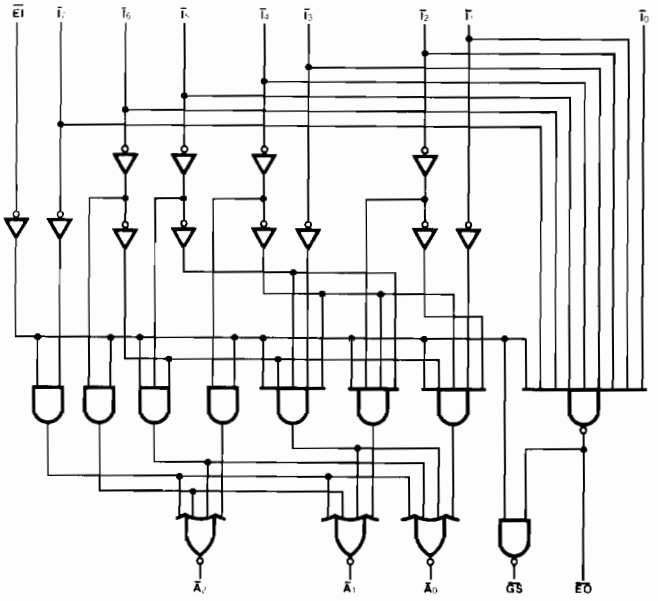
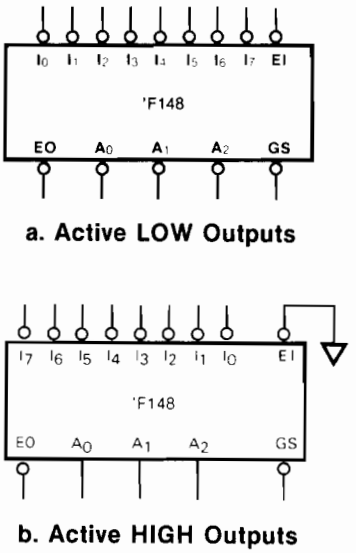


Figure 4-3 Logic Diagram

The 'F148 8-input priority encoder is a multipurpose device useful in a wide variety of applications such as priority encoding, priority control, decimal or binary encoding, code conversion, analog-to-digital and digital-to-analog conversion. A priority encoder can improve computer systems by providing the computer with high-speed hardware priority interrupt capabilities. It is expanded easily through Input and Output Enables to provide priority encoding over many bits.

The logic symbol and truth table for the 'F148 are shown above. The 'F148 accepts eight active LOW inputs ($\overline{I}_0$-$\overline{I}_7$) and produces a binary weighted output code ($\overline{A}_0\overline{A}_1\overline{A}_2$) representing the position of the highest order active input. A priority is thus assigned to each input ($\overline{I}_7$ has the highest priority). Therefore, when two or more inputs are simultaneously active and the group is enabled, the input with the highest priority is encoded and the other inputs are ignored. In addition, all inputs are OR-tied to provide a group signal indicating the presence of any LOW input signal. This group signal is LOW whenever any input is LOW and the encoder is enabled.

Operation



a. Active LOW Outputs

b. Active HIGH Outputs

Figure 4-4

When HIGH, the Enable input ($\overline{EI}$) overrides all inputs and forces all outputs HIGH. The active LOW Enable output indicates that the encoder is enabled and no active signal has been applied. The combination of Enable input and Enable output permits series expansion of priority encoding to many levels with little additional logic. The truth table (Table 4-1) lists all the input combinations and the resulting outputs. All inputs and outputs are active LOW.

The 'F148 priority encoder also can be represented with active HIGH address outputs (Figure 4-3b), but in this case disabling the input generates an address 7. Also, input zero has the highest priority. Because the 'F148 is a combinational network, address glitches can appear during input transitions. Moreover, when all priority inputs are HIGH, a HIGH-to-LOW Enable change can cause a transient on the Group Signal output.

Priority Encoder Expansion

The 'F148 priority encoder can be expanded in series or in parallel. Expansion in series requires a minimum of components while expansion in parallel offers the highest operating speed.

The expanded priority encoder generates the binary address of the highest order priority input in two or more levels. The inputs are applied to what can be considered the first level of priority encoders. Since a binary code is cyclic, the output of each encoder represents potentially the three least significant bits of the output. Series and parallel expansion differ in how the proper encoder and its output code are selected. In both cases, the group signals of the encoders go through a second level of encoding to determine the more significant group. Two levels of encoding are required for up to 64 inputs, and three levels are required for 65 to 512 inputs.

Series Expanded Encoders

In this 16-input expanded encoder, and in the 64-input version shown below, the priority encoders are enabled in series; the Enable output of the most significant encoder is connected to the Enable input of the next less significant encoder. This allows the highest order encoder with an active LOW input to disable all less significant encoders. The selection of the three least significant bits is simple since this particular encoder is the only one with active outputs. All less significant encoders are disabled and all more significant encoders have no active inputs and consequently no active outputs. Therefore the appropriate address outputs can be ORed with active LOW input and output OR gates (ANDs) to generate the least significant bits. The group signals of each first level encoder are encoded, if necessary, to provide the most significant bits of the output. In a series array, only the highest order encoder with an active input has a Group Signal output and its address can be generated with gates.

The disadvantage of series expansion is that (worst case) the Enable signals must ripple through every encoder before a valid output is provided.

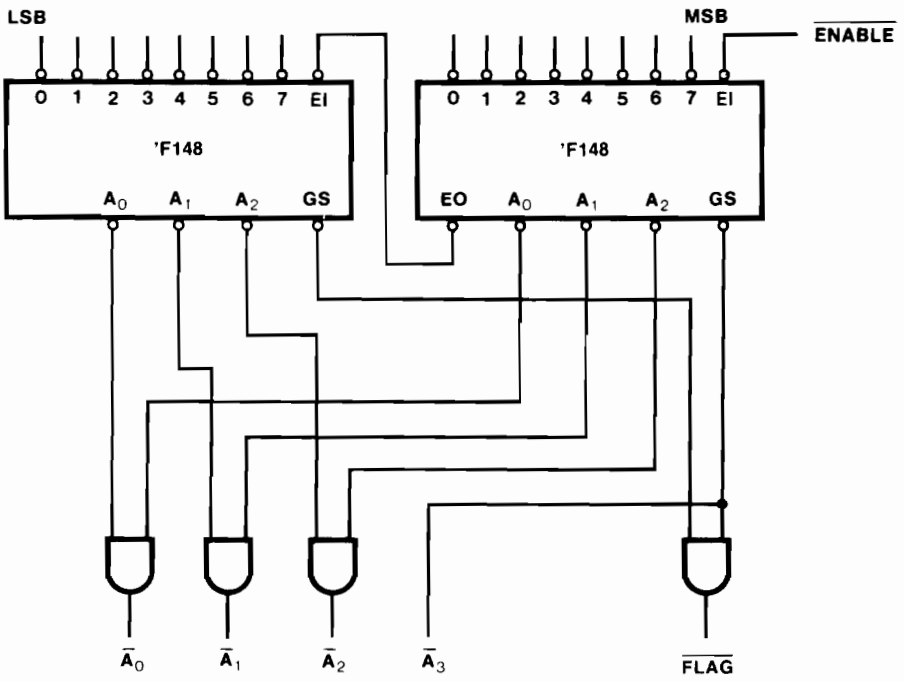


Figure 4-5 16-Input Series Expanded Encoder

64-Input Serial Expanded Encoder

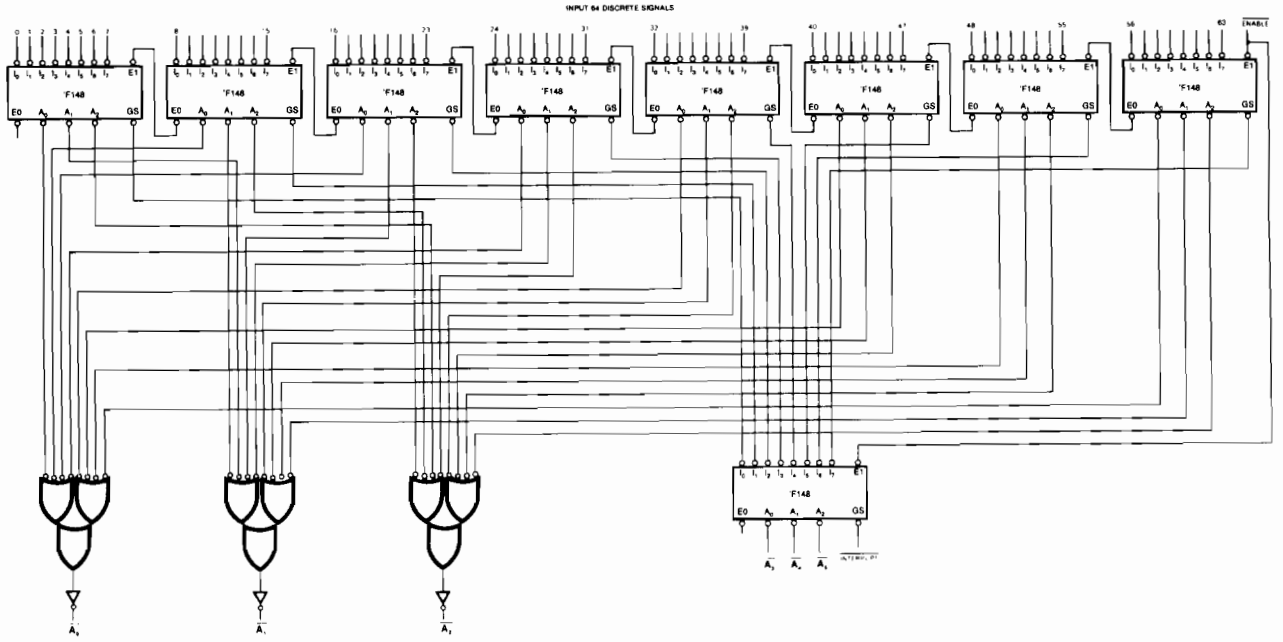


Figure 4-6

64-Input Parallel Expanded Encoder

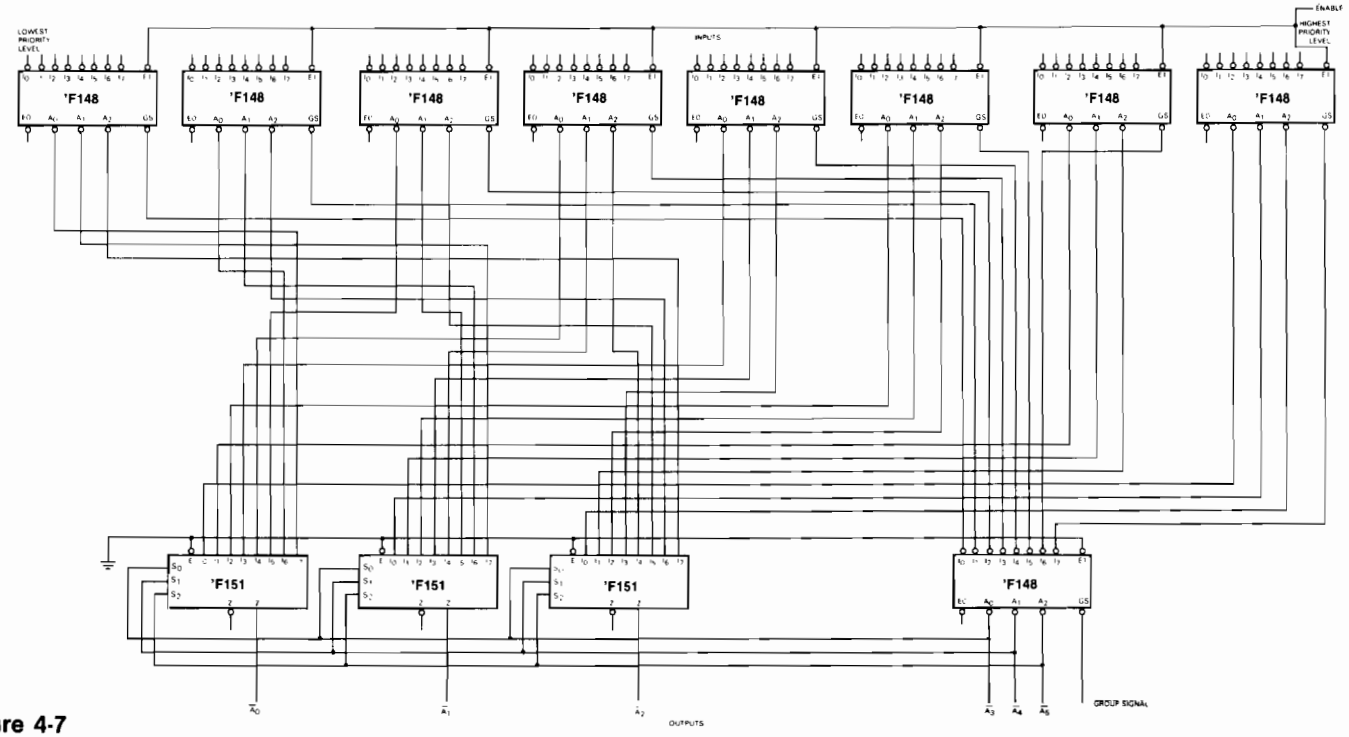
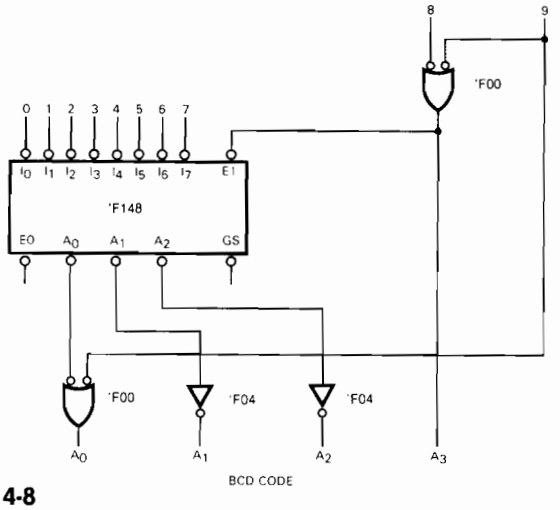


Figure 4-7

A fast expansion method (only 20ns delay) is shown in Figure 4-7 above. Each of the first level priority encoders operates independently. The group signals of each of the encoders are applied to another priority encoder that selects the highest order group signal and provides the most significant bits of the priority address. These bits are supplied as an address to three multiplexers which select the appropriate least significant bits. In this case the second level encoder must be a priority encoder, since more than one group signal can be active.

Decimal (10-Input) BCD 8421 Encoder



A 10-input, decimal-to-BCD 8421 code priority encoder is useful for a decimal keyboard or it can be connected directly to a display decoder.

A 2-input NAND gate disables the 'F148 when inputs $\overline{I}_8$ or $\overline{I}_9$ go LOW and is used to produce the correct output code. When $\overline{I}_8$ or $\overline{I}_9$ are not LOW, the encoder is enabled and encodes inputs $\overline{I}_0$-$\overline{I}_7$ normally. This decimal encoder has active HIGH outputs representing the highest order input. However, just inserting the two inverters in the A_0 and A_3 lines instead of the A_1 and A_2 lines provides active LOW outputs.

Figure 4-8

Decimal (20-Input) BCD 8421 Encoder

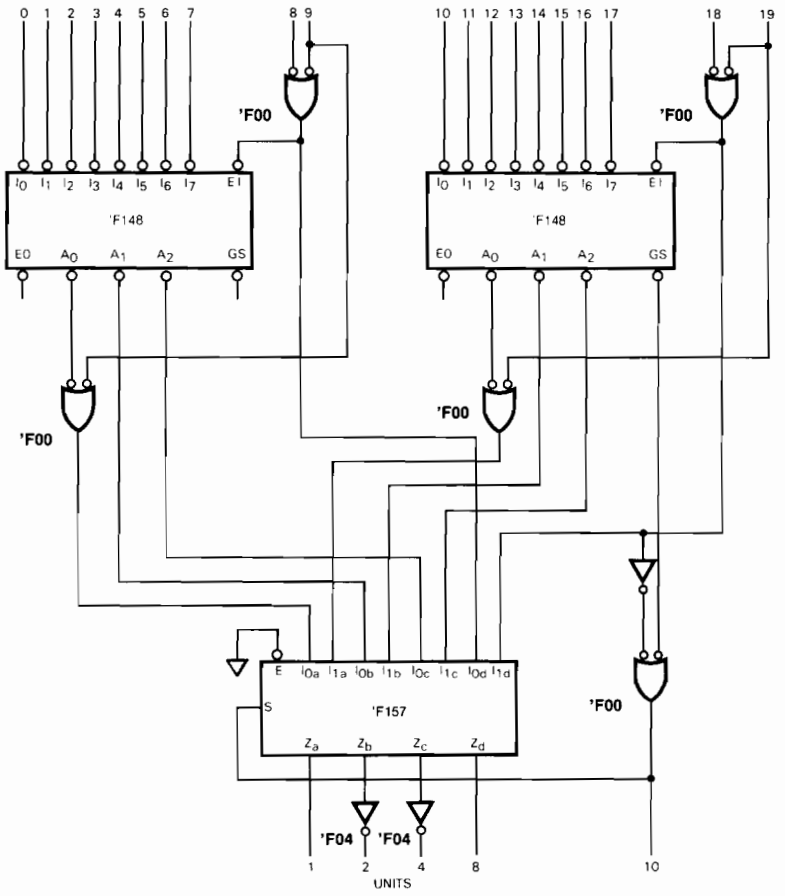


Figure 4-9

A 20-input BCD encoder is formed by generating a Group Signal from the most significant decimal encoder, using this output as the 10s output and also using it to select the proper least significant digit through the 2-input multiplexer ('F157).

Switch Encoder/Debouncer

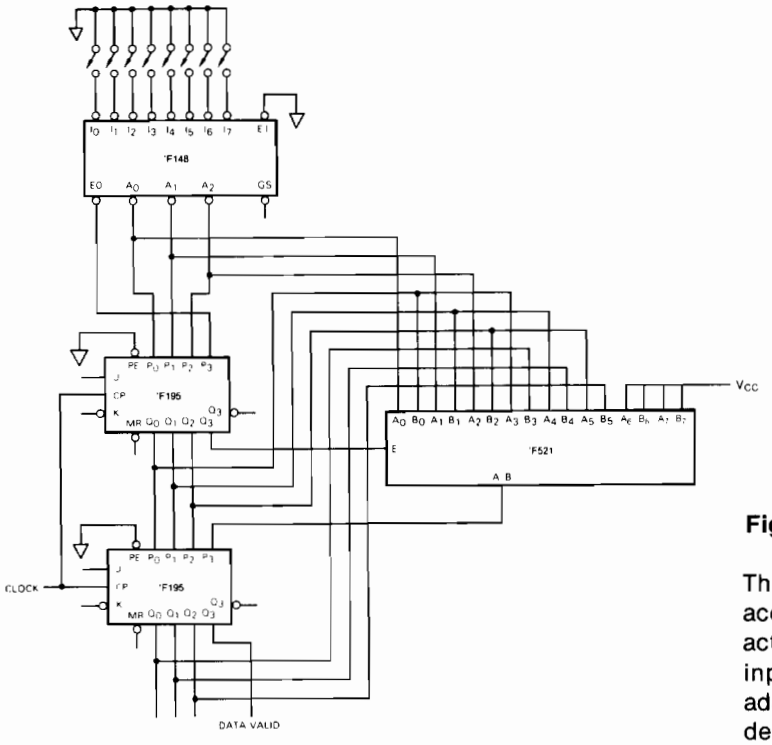


Figure 4-10

Switches or keys must be encoded for data entry into many digital systems. The 'F148 is often most efficient for encoding groups of a moderate number of keys, from eight to sixteen. Other approaches, such as scanning, are more efficient for large keyboard arrays.

In Figure 4-10, the output of the 'F148 is sampled, stored, and compared over several clock periods. A Data Valid signal appears only after the 'F148 outputs and the keys have stabilized. The clock rate supplied to the shift register must be adjusted to the bounce characteristics of the particular switch used so that all switch bounce is ignored.

Two 'F195 shift registers are connected as four 2-bit shift registers so that codes from the 'F148 are stored for two successive clock periods. All outputs are compared and a Data Valid signal appears one clock pulse after identical output addresses have been clocked twice into the 'F195 register. This insures that the output address is correct whenever the Data Valid signal is HIGH.

Linear Priority Encoder

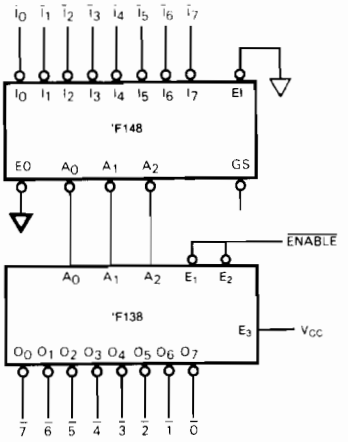


Figure 4-11

The linear encoding network shown in Figure 4-11 accepts eight active LOW inputs and produces a single active LOW output corresponding to the highest order input. The network consists of an 'F148 to establish the address of the highest order input and a 'F537 to decode this address and activate the appropriate output. This method offers a considerable package reduction over discrete linear priority networks and is easily expandable by adding more encoders and decoders. A 16-input encoding network requires only two 'F148s, two 1-of-8 decoders ('F138s), and one gate.

4-Digit BCD To Binary Encoder

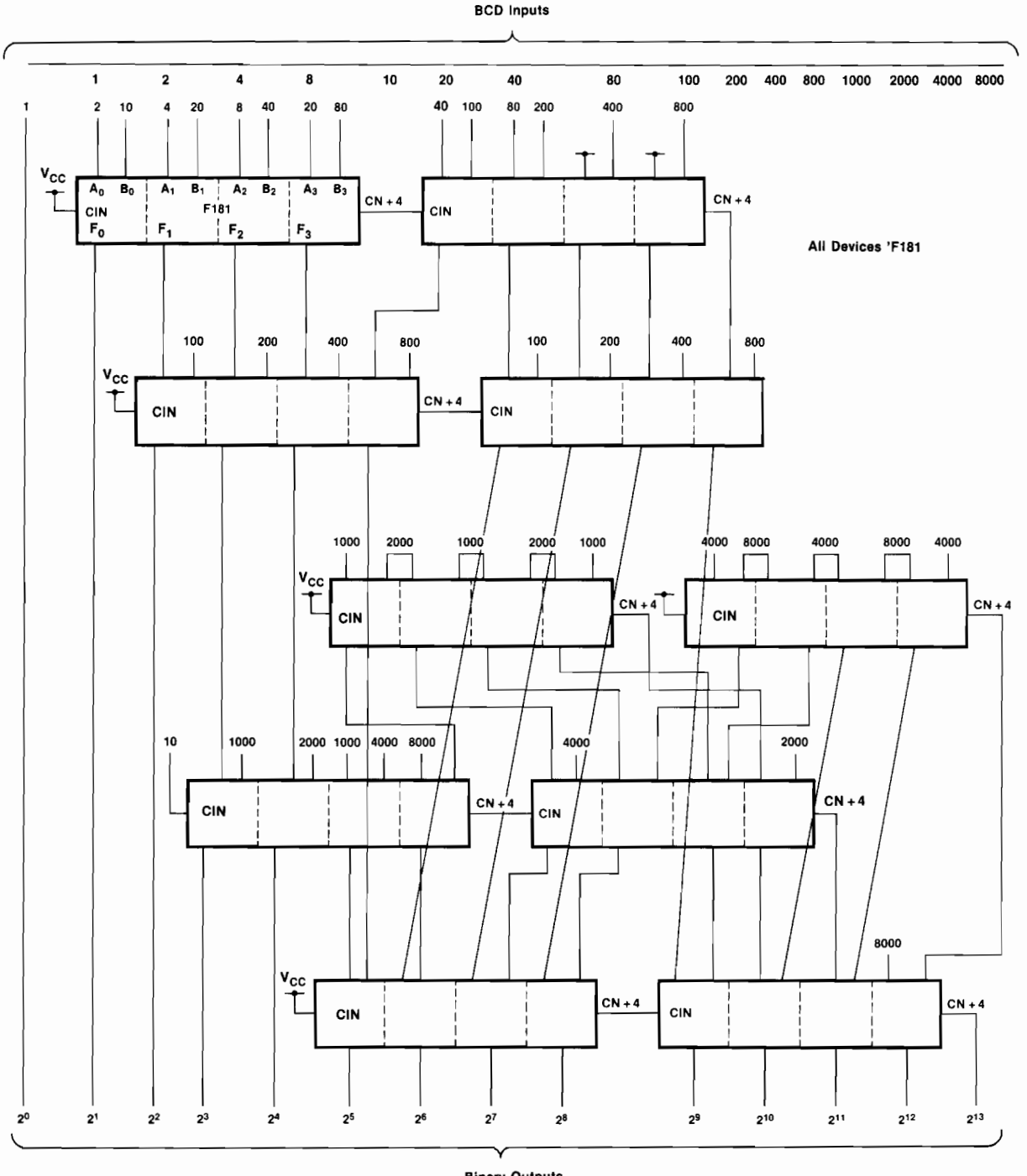
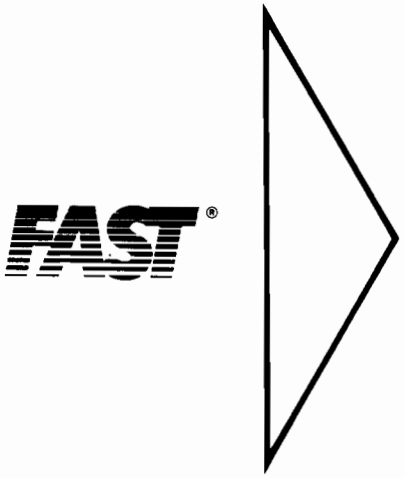


Figure 4-12

Figure 4-12 illustrates a scheme whereby the 'F181 ALU or 'F283 may be used to form a BCD to Binary encoder. In this case four digits have been encoded. However, the circuit could be readily expanded to a larger number of digits if desired. Since either active HIGH or active LOW operation could be selected, the mode control input, M, would be tied HIGH for active LOW operation and tied LOW for active HIGH operation. Note that one of the Carry Ins is not connected to either the HIGH or LOW bus, but rather to the BCD 10. One of the Carry Outs can be ignored since it cannot be active for any legitimate input condition.



FAIRCHILD
A Schlumberger Company

Operators

Introduction

The term "operators" describes a broad category of combinatorial (non-memory) devices which perform logic operations (such as AND, OR, Exclusive-OR, and invert), arithmetic operations (add, subtract, multiply, divide), compare the magnitude of two operands or generate/check parity.

Because operators tend to be used in the heart of digital systems, they strongly influence system design and architecture. It is important to investigate the large number of alternate devices before settling on a system design. These devices represent compromises of speed, cost, part count, and connection complexity. The following points are some major design considerations.

SPEED—Slower systems usually require fewer and less expensive components and are less sensitive to noise. Careful consideration should be given to system performance versus system cost.

CODES—Binary arithmetic is simpler than decimal arithmetic. BCD and Excess-3 codes are preferred for decimal operation. Special codes (BCD and Excess-3 Gray) require extensive conversion before use in arithmetic operations.

NEGATIVE NUMBERS—For addition and subtraction, negative numbers are best represented as complements, one or twos complement in binary notation, nine or tens complement in decimal notation. The easiest to generate are ones and nines complements; however, twos and tens complements permit faster and simpler arithmetic. For multiplication and division, and for human interfacing (input/output), negative numbers are best represented in signed magnitude notation.

VERSATILITY—When several different operations are to be performed, a well designed Arithmetic Logic Unit (ALU) may be able to execute them in sequence. For example, an ALU can count by incrementing or decrementing a register, or it may be used to control a display multiplexer, etc.

Fairchild offers a broad range of operator devices in the FAST family, ranging in complexity from Exclusive-OR gates to multipliers and 4-bit Arithmetic Logic Units (ALUs).

Adders/Subtractors
Multipliers
Comparator Systems
Error Detection/Correction
Code Conversion

Terminal Confusion or How Can Two States Lead to So Many Terms?

The signals used in digital systems are described in several different and sometimes confusing terms. A logic signal can be either ACTIVE (= TRUE) or NOT ACTIVE (= NOT TRUE = FALSE). Digital circuits, on the other hand, are defined for voltage levels that are either HIGH (more positive) or LOW (less positive or more negative). Either of these levels can be considered ACTIVE (TRUE), then the opposite level is NOT ACTIVE (FALSE).

MIL STD 806B has established a clear symbology: the HIGH level is considered ACTIVE unless a small circle ("bubble") at the input or output describes the opposite assignment (LOW = ACTIVE).

In non-arithmetic circuits the symbols "0" and "1" are unnecessary and confusing because some people think that a "1" implies a HIGH level, others think of it as an ACTIVE (TRUE) signal, and some mistakenly think that it must mean both ACTIVE and HIGH.

Thus, this book generally does not use 0 and 1, but uses the terms ACTIVE and NOT ACTIVE for systems descriptions and the terms H and L for circuit descriptions and truth tables.

In arithmetic (binary and BCD) systems, the terms zero and one cannot be avoided, since they have a mathematical significance. They have to be related to the logic terms in a consistent and unambiguous way.

Arithmetic 1 = ACTIVE = TRUE
Arithmetic 0 = NOT ACTIVE = NOT TRUE = FALSE

The rules of MIL STD 806B are then used to describe whether a HIGH level means a "1" (active HIGH, no bubble) or whether a LOW level means a "1" (active LOW, with a bubble at the input or output of the logic symbol).

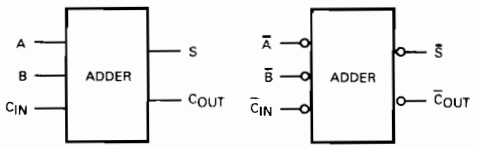


Figure 5-1

For example, a full adder produces sum and carry outputs as a function of the three inputs A, B and C. Truth Table 5-1 describes the electrical function in terms of HIGH and LOW. Table 5-1 and Figure 5-1 also describe this circuit in terms of either active HIGH or active LOW logic levels. Any logic network which performs binary addition or subtraction can be described in terms of active HIGH as well as in terms of active LOW inputs and outputs.

Such equivalence is a basic feature of adder structures and is true regardless of the number of bits and the method of carry propagation. It applies to a single full adder as well as to a complex ALU system.

Carry Signals In Parallel Binary Adders

High-speed digital systems perform addition and subtraction on parallel words of typically 8 to 64 bits. The result of an addition or subtraction at any bit position, however, depends not only on the two operand bits in that position, but also on the less significant operand bits. More specifically, the result depends on the carry from the less significant bit positions.

Ripple Carry

In the simplest scheme, each position receives a potential carry input from the less significant position and passes a potential carry on to the more significant position. Thus the worst case delay for the addition of two n-bit numbers is n-1 carry delays plus one sum delay. This technique is used with simple 4-bit ripple carry adders.

Table 5-1 Functions

Active High Logic Function					Electrical Function					Active Low Logic Function				
A	B	C_{IN}	S	C_{OUT}	A	B	C_{IN}	S	C_{OUT}	$\overline{A}$	$\overline{B}$	$\overline{C}_{IN}$	$\overline{S}$	$\overline{C}_{OUT}$
0	0	0	0	0	L	L	L	L	L	1	1	1	1	1
1	0	0	1	0	H	L	L	H	L	0	1	1	0	1
0	1	0	1	0	L	H	L	H	L	1	0	1	0	1
1	1	0	0	1	H	H	L	L	H	0	0	1	1	0
0	0	1	1	0	L	L	H	H	L	1	1	0	0	1
1	0	1	0	1	H	L	H	L	H	0	1	0	1	0
0	1	1	0	1	L	H	H	L	H	1	0	0	1	0
1	1	1	1	1	H	H	H	H	H	0	0	0	0	0

H = HIGH Voltage Level
L = LOW Voltage Level

Carry Lookahead

Addition and subtraction can be made much faster if more logic is used at each bit position to anticipate the carry into this position instead of waiting for a ripple carry to propagate through all the lower positions. An adder constructed with carry anticipation is called a "carry lookahead adder."

The carry into position 0 is C_0

The carry into position 1 is $C_1 = A_0 \bullet B_0 + C_0 (A_0 + B_0)$

The carry into position 2 is $C_2 = A_1 \bullet B_1 + C_1 (A_1 + B_1)$

If the two auxiliary functions & and V are defined

$$\&_i = A_i B_i$$
$$V_i = A_i + B_i$$

then the carry equations are:

$$C_1 = \&_0 + V_0 C_0$$
$$C_2 = \&_1 + V_1(\&_0 + V_0 C_0)$$
$$C_3 = \&_2 + V_2(\&_1 + V_1\&_0 + V_1 V_0 C_0)$$

or, in general terms:

$$C_{i+1} = \&_i + V_i\&_{i-1} + V_i V_{i-1}\&_{i-2} + V_i V_{i-1} V_{i-2}\&_{i-3} + \ldots$$

The anticipated carry into any position can thus be generated in two gate delays (counting AND/OR/INVERT as one gate delay), one gate delay to generate all the & and V functions, and a second gate delay to generate the anticipated carry. The sum/difference outputs are generated in one additional delay for a total of three gate delays, independent of word length. The auxiliary functions & and V can be interpreted as

& = Carry Generate—AB generates a carry, independent of any incoming carry
V = Carry Propagate—A + B pass on an incoming carry

This "brute force" carry lookahead scheme is conceptually simple, but, due to the large number of interconnections and the heavy loading of the & and V functions, becomes impractical as the word length increases beyond five or six bits.

The same concept, however, can be applied on a higher level by dividing the word into practical blocks of 4-bit lengths, using carry lookahead within each block, generating new auxiliary functions G, Carry Generate and P, Carry Propagate which refer to the whole block. G is obviously the carry out of the most significant position of the block. P is defined as Carry Propagate through the block, i.e., P is True if a carry into the block would result in a carry out of the block. For a block size of four bits,

$$G = \&_3 + V_3\&_2 + V_3 V_2\&_1 + V_3 V_2 V_1\&_0$$
$$P = V_3 V_2 V_1 V_0$$

Neither of these functions is affected by the incoming carry; they will therefore be stable within two gate delays and can be used to supply carry information to the more significant blocks. The carry into block n is:

$$C_n = G_{n-1} + P_{n-1}G_{n-2} + P_{n-1}P_{n-2}G_{n-3} + \ldots$$

This carry in signal is used in the internal carry lookahead structure:

$$C_0 = C_n$$
$$C_1 = \&_0 + V_0 C_n$$
$$C_2 = \&_1 + V_1\&_0 + V_1 V_0 C_n$$
$$C_3 = \&_2 + V_2\&_1 + V_2 V_1\&_0 + V_2 V_1 V_0 C_n$$

The FAST carry lookahead arithmetic logic units, the 'F181 and 'F381, use this 2-level carry lookahead. The 'F381 handles three arithmetic and three logical operations, generating propagate and generate signals for external lookahead carry. The 'F181 has more logic flexibility, which requires four additional pins. The lookahead logic above 4 bits must be contained in a separate device, the 'F182. Only one 'F182 is needed to achieve full carry lookahead across sixteen bits. Beyond 16 bits 'F182s can be cascaded.

Number Representation

In general, adders and ALUs work on binary numbers. Operation in other number systems, such as BCD, Excess-3, etc. is achieved by additional logic and/or additional cycles through the binary adder.

There is only one way to represent positive binary numbers, but negative binary numbers can be represented in three ways.

- Sign Magnitude—The most significant bit indicates the sign (0 = positive, 1 = negative). The remaining bits indicate the magnitude, represented as a positive number.

Sign				LSB	
0	1	1	0	1	= +13
1	1	1	0	1	= −13

This representation is convenient for multiplication and division, and may be desirable for human-oriented input and output, but for addition and subtraction it is inconvenient and rarely used.

- Ones Complement—Negative numbers are bit inversions of their positive equivalents. The most significant bit indicates the sign (0 = positive, 1 = negative). Thus $-A$ is actually represented as $2^n - A - 1$. The ones complement is very easy to form, but it has several drawbacks, notably a double representation for Zero (all Ones or all Zeros)

- Twos Complement—This is the most common representation. It is more difficult to generate than ones complement, but it simplifies addition and subtraction. The twos complement is generated by inverting each bit of the positive number and adding one to the LSB.

Sign				LSB	
0	1	1	0	1	= +13
1	0	0	1	1	= −13

Thus an n-bit word can represent the range from $+(2^{n-1}-1)$ to $-(2^{n-1})$

A 4-bit word can represent the range from 0111 = +7 to 1000 = −8.

Addition and Subtraction of Binary Numbers

Addition of positive numbers is straightforward, but a carry into the sign bit must be prevented and interpreted as overflow. When two negative numbers, or a negative and a positive number are added, the operation depends on the negative number representation. In twos complement methods, addition is straightforward but it must include the sign bit. Any carry out of the sign position is simply ignored.

+14	01110	+ 7	00111	−4	11100
− 7	11001	−14	10010	−3	11101
+ 7	00111	− 7	11001	−7	11001

If ones complement notation is used, the operation is similar, but the carry out of the sign bit must be used as a carry input to the least significant bit (LSB). This is commonly called "end-around carry."

+14	01110	+ 7	00111	−4	11011
− 7	11000	−14	10001	−3	11100
	00110				10111
	+ 1				+ 1
+ 7	00111	− 7	11000	−7	11000

In twos complement subtraction, the arithmetic is performed by inverting (ones complement) the subtrahend and adding, and by forcing a carry into the least significant bit (LSB).

+14	01110	+ 7	00111	− 6	11010
−(+ 7)	−00111	−(+14)	−01110	−(+ 8)	−01000
	01110		00111		11010
	+11000		+10001		+10111
	+ 1		+ 1		+ 1
+ 7	00111	− 7	11001	−14	10010

In ones complement methods, subtraction is performed by inverting (ones complement) the subtrahend and adding, using the Carry Out of the sign position as carry input to the LSB (end-around carry).

+14	01110	+ 7	00111	− 6	11001
−(+ 7)	−00111	−(+14)	−01110	−(+ 8)	−01000
	01110		00111		11001
	+11000		+10001		+10111
	00110				10000
	+ 1				+ 1
+ 7	00111	− 7	11000	−14	10001

It is interesting to note that the Carry Out of the sign position occurs when the result does not change sign; no carry occurs when the sign changes, implying a "borrow."

Overflow

Adding two numbers of the same sign or subtracting two numbers of opposite sign might generate a result which cannot be represented by the given word length. This is overflow. It must be detected and used to initiate some corrective routine. Overflow occurs when the Carry Out of the sign position differs from the Carry In to the sign position.

$$\text{OVERFLOW} = C_S \oplus C_{S+1}$$

When the sign is in the most significant position of a 'F181 or 'F381 ALU, the carry signal into this position is not directly available but must be regenerated in one of several ways. The simplest method is to use the equation

$$C_S = S_S \oplus A_S \oplus B_S \text{ and}$$
$$\text{OVERFLOW} = S_S \oplus A_S \oplus B_S \oplus C_{S+1}$$

Adders/Subtractors

'F283 4-Bit Ripple Carry Adder

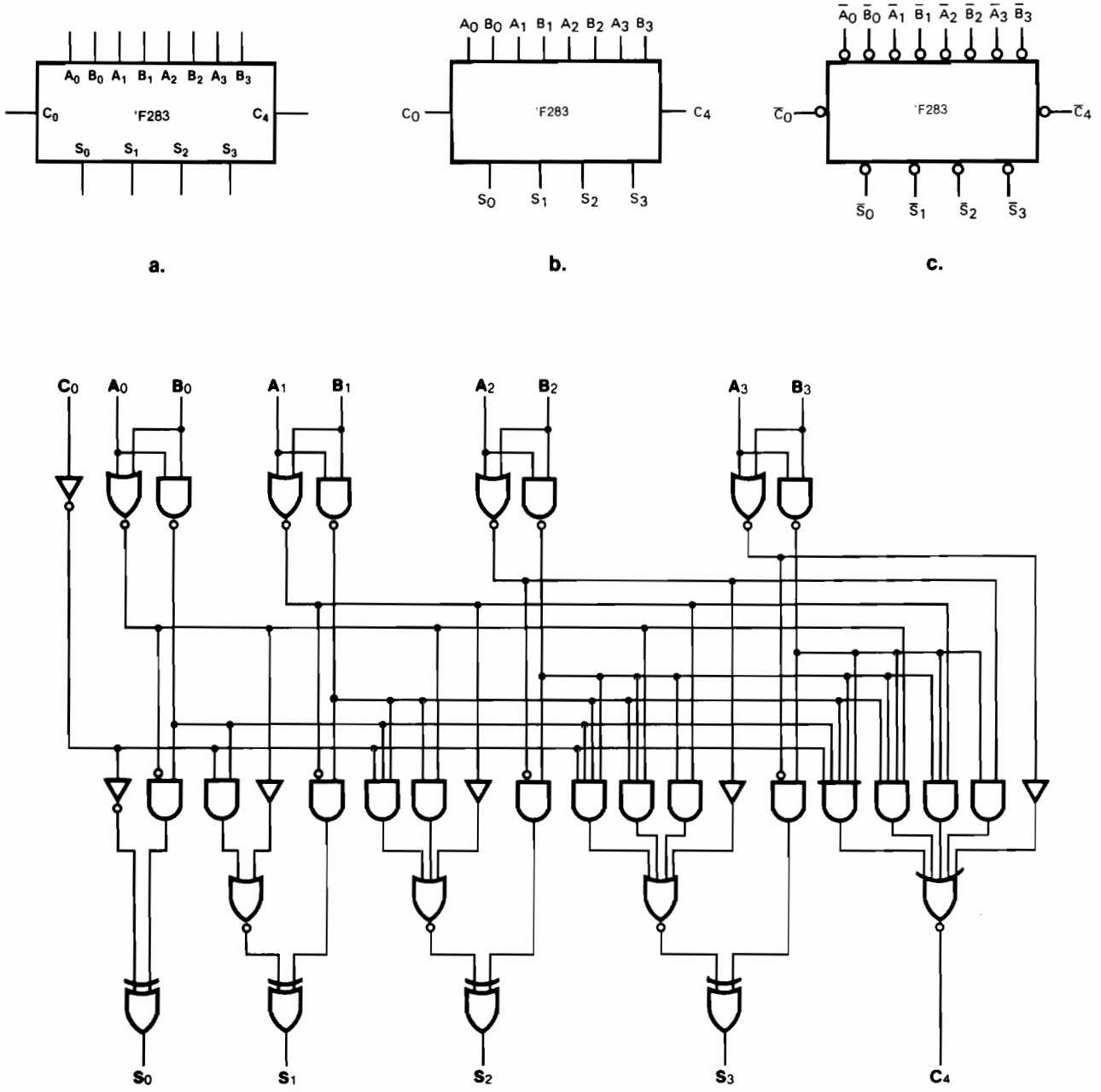


Figure 5-2

As with all other adders, 'F283 operation can be described in terms of both active HIGH (Figure 5-2b) and active LOW (Figure 5-2c) inputs and outputs. Note that with active High inputs, Carry In cannot be left open, but must be held LOW when no Carry In is intended.

The 'F283 4-bit carry adder provides 4-bit addition in a single 16-lead package, which is very useful in applications which do not require the functional versatility of more sophisticated 4-bit Arithmetic Logic Units.

The 'F283 is four full adders. They add four bits of A with four bits of B plus a carry input, generating four sum bits and a carry output. There are no control inputs and the speed is enhanced by the internal lookahead carry structure.

'F283 as a 3-Bit or a 2 + 1-Bit Adder

a. 3-Bit Adder

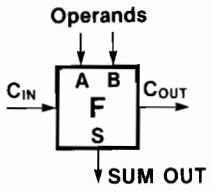
b. 2-Bit and 1-Bit Adders

Figure 5-3

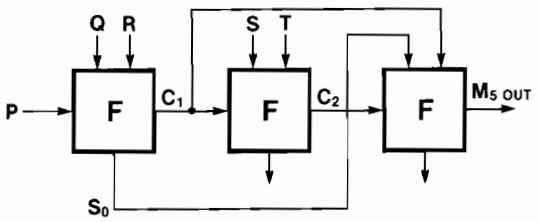
The 'F283 can be connected as a 3-bit adder (Figure 5-3a) with carry input and output by using the least significant data inputs and the carry in, normally tying A_3 and B_3 together and terminating them either HIGH or LOW. The carry out signal is available on the S_3 output.

The 'F283 is useful as two independent adders, (Figure 5-3b) one two bits wide, the other one bit wide. The two least significant bits are used as a 2-bit adder with carry out on the S_2 output. Inputs A_2 and B_2 are tied together and used as carry input for the second adder. The A and B operands of the second adder are applied on the A_3 and B_3 inputs; carry out appears on the C_4 output. All these configurations work with active HIGH as well as active LOW inputs and outputs.

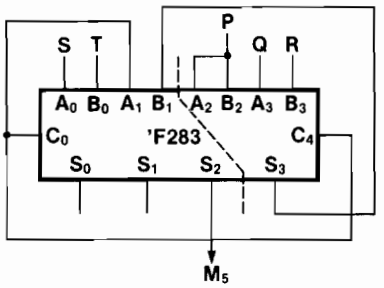
4-Bit Adder Serves as 5-Input Majority Gate



a. Full Adder Symbol



b. 5-Input Majority Function



c. Implementing the 5-Input Majority Function with a 4-Bit Adder

Figure 5-4

A majority gate is one whose output is True if a Majority of its inputs are True, a function also known as voting logic. A full adder acts as a 3-input majority gate since the C_{OUT} is True if any two or all three of the inputs are True.

To obtain a 5-input majority function (Figure 5-4b) inputs P, Q and R are combined in the first adder. Inputs S and T are applied to the second adder, along with the carry C_1 from the first adder. Note that the carry C_2 from the second adder is not sufficient indication that a majority of the inputs are True, since it will be True if only S and T are True but will be False if only P, Q and R are True. But if any two or all three of S_0, C_1 and C_2 are True, then it must be that three or more of the inputs are True. Thus the third adder is used as a majority gate for S_0, C_1 and C_2.

A slight dilemma occurs in trying to implement Figure 5-4b with one MSI package since there is no triple adder available, and in the 4-bit adder only one carry is brought out of the package whereas access to two carries is needed. The dilemma is solved by using one adder of the 4-bit circuit as an input, and an output that is effectively independent of the adder (Figure 5-4c). Input function P connects to both operand inputs of the third adder, which means that its C_{OUT} is always equal to P and is independent of its C_{IN}. Thus the function of the first adder in Figure 5-4b is performed by the fourth adder in Figure 5-4c, whose C_{OUT} is available. The functions of the second and third adders of Figure 5-4b are performed by the first and second adders of Figure 5-4c. The C_{OUT} of this second adder passes through to the Sum output of the third adder to provide the final output.

'F181 Arithmetic Logic Unit

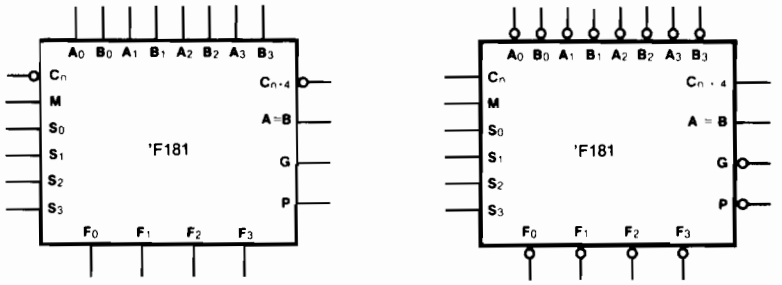


Figure 5-5

The 'F181 is a 4-bit Arithmetic Logic Unit of much higher complexity, versatility and speed than the adders mentioned so far. The 'F181 comes in a 24-lead package and can be used in parallel high-speed binary systems. It accepts four bits of one operand (A) and four bits of a second operand (B) and generates four outputs (F). Operation is controlled by Mode and Select inputs.

This ALU is a parallel 4-bit MSI device that can perform sixteen arithmetic and all sixteen possible logic operations on two 4-bit parallel words. The significant arithmetic operations are add, subtract, pass, increment, decrement, invert and double. The operation is selected by four select lines S_0-S_3 and a mode control line M, which is LOW for arithmetic operations and HIGH for logic operations. The device has a Carry In, a Carry Out for ripple carry cascading of units, and two lookahead auxiliary carry functions, Carry Generate and Carry Propagate for use with the carry lookahead 'F182. An open collector A = B output is also provided that can be AND-tied to the A = B outputs of other ALUs to detect an all HIGH output condition for several units.

'F181 Operation

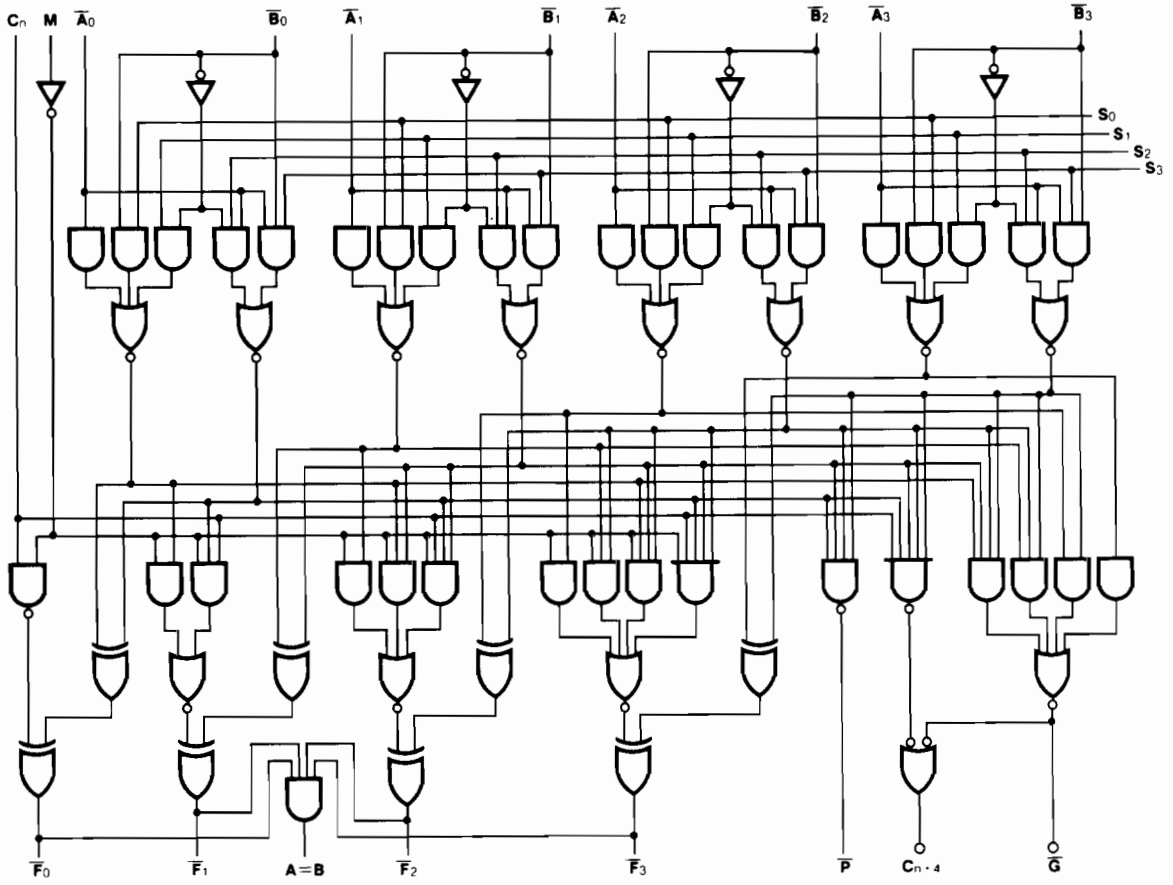


Figure 5-6

In the logic of this ALU, four identical AND/OR networks gate the A and B input operands with the four select lines S_0-S_3 to produce the required first level auxiliary AND and OR functions. These are then used to generate the sum and carry functions. Internal carry lookahead gives high speed. The A = B output is generated by sensing the all-ones condition at the F outputs. When control M is in the HIGH state, carries are inhibited from propagating and logic functions are generated at the outputs. The functions available with the device form a closed set such that inversion of the logic inputs produces a function which is still in the set. Therefore, the device performs the same logic and arithmetic functions in the active HIGH representation as it does in the active LOW representation, but with a different select code. If a mixed representation is employed, the majority of useful functions is still available. The four modes of ALU use and operation tables for each mode are shown in Table 5-2.

Table 5-2 'F181 Operation Tables

S_0	S_1	S_2	S_3	Logic (M = H)	Arithmetic (M = L, C_0 = Inactive)	Arithmetic (M = L, C_0 = Active)
L	L	L	L	$\overline{A}$	A minus 1	A
H	L	L	L	$\overline{A \cdot B}$	A•B minus 1	A•B
L	H	L	L	$\overline{A} + B$	A•$\overline{B}$ minus 1	A•$\overline{B}$
H	H	L	L	Logic '1'	minus 1 (2s comp.)	Zero
L	L	H	L	$\overline{A + B}$	A plus (A + $\overline{B}$)	A plus (A + $\overline{B}$) plus 1
H	L	H	L	$\overline{B}$	A•B plus (A + $\overline{B}$)	A•B plus (A + $\overline{B}$) plus 1
L	H	H	L	$\overline{A \oplus B}$	A minus B minus 1	A minus B
H	H	H	L	A + $\overline{B}$	A + $\overline{B}$	A + $\overline{B}$ plus 1
L	L	L	H	$\overline{A}$•B	A plus (A + B)	A plus (A + B) plus 1
H	L	L	H	A⊕B	A plus B	A plus B plus 1
L	H	L	H	B	A•$\overline{B}$ plus (A + B)	A•$\overline{B}$ plus (A + B) plus 1
H	H	L	H	A + B	A + B	A + B plus 1
L	L	H	H	Logic '0'	A plus A (2 × A)	A plus A (2 × A) plus 1
H	L	H	H	A•$\overline{B}$	A plus A•B	A plus A•B plus 1
L	H	H	H	A•B	A plus A•$\overline{B}$	A plus A•$\overline{B}$ plus 1
H	H	H	H	A	A	A plus 1

a. All input data inverted

S_0	S_1	S_2	S_3	Logic (M = H)	Arithmetic (M = L, C_0 = Inactive)	Arithmetic (M = L, C_0 = Active)
L	L	L	L	$\overline{A}$	A	A plus 1
H	L	L	L	$\overline{A + B}$	A + B	A + B plus 1
L	H	L	L	$\overline{A}$•B	A + $\overline{B}$	A + $\overline{B}$ plus 1
H	H	L	L	Logic '0'	minus 1 (2s comp.)	Zero
L	L	H	L	$\overline{A \cdot B}$	A plus A•$\overline{B}$	A plus A•$\overline{B}$ plus 1
H	L	H	L	$\overline{B}$	A•$\overline{B}$ plus (A + B)	A•B plus (A + B) plus 1
L	H	H	L	A⊕B	A minus B minus 1	A minus B
H	H	H	L	A•$\overline{B}$	A•$\overline{B}$ minus 1	A•$\overline{B}$
L	L	L	H	$\overline{A} + B$	A plus A•B	A plus A•B plus 1
H	L	L	H	$\overline{A \oplus B}$	A plus B	A plus B plus 1
L	H	L	H	B	A•B plus (A + $\overline{B}$)	A•B plus (A + $\overline{B}$) plus 1
H	H	L	H	A•B	A•B minus 1	A•B
L	L	H	H	Logic '1'	A plus A (2 × A)	A plus A (2 × A) plus 1
H	L	H	H	A + $\overline{B}$	A plus (A + B)	A plus (A + B) plus 1
L	H	H	H	A + B	A plus (A + $\overline{B}$)	A plus (A + $\overline{B}$) plus 1
H	H	H	H	A	A minus 1	A

b. All input data true

S_0	S_1	S_2	S_3	Logic (M = H)	Arithmetic (M = L, C_0 = Inactive)	Arithmetic (M = L, C_0 = Active)
L	L	L	L	$\overline{A}$	A minus 1	A
H	L	L	L	$\overline{A} + B$	A•$\overline{B}$ minus 1	A•B
L	H	L	L	$\overline{A \cdot B}$	A•B minus 1	A•B
H	H	L	L	Logic '1'	minus 1 (2s comp.)	Zero
L	L	H	L	$\overline{A}$•B	A plus (A + B)	A plus (A + B) plus 1
H	L	H	L	B	A•$\overline{B}$ plus (A + B)	A•$\overline{B}$ plus (A + B) plus 1
L	H	H	L	A⊕B	A plus B	A plus B plus 1
H	H	H	L	A + $\overline{B}$	A + B	A + B plus 1
L	L	L	H	$\overline{A + B}$	A plus (A + $\overline{B}$)	A plus (A + $\overline{B}$) plus 1
H	L	L	H	$\overline{A \oplus B}$	A minus B minus 1	A minus B
L	H	L	H	$\overline{B}$	A•$\overline{B}$ plus (A + B)	A•$\overline{B}$ plus (A + B) plus 1
H	H	L	H	A•$\overline{B}$	A•$\overline{B}$ minus 1	A•$\overline{B}$
L	L	H	H	Logic '1'	A plus A (2 × A)	A plus A (2 × A) plus 1
H	L	H	H	A + B	A plus (A + $\overline{B}$)	A plus (A + $\overline{B}$) plus 1
L	H	H	H	A + $\overline{B}$	A plus (A + B)	A plus (A + B) plus 1
H	H	H	H	A	A minus 1	A

c. A input data inverted; B input data true

S_0	S_1	S_2	S_3	Logic (M = H)	Arithmetic (M = L, C_0 = Inactive)	Arithmetic (M = L, C_0 = Active)
L	L	L	L	$\overline{A}$	A minus 1	A
H	L	L	L	$\overline{A}$•B	A + $\overline{B}$	A + $\overline{B}$ plus 1
L	H	L	L	$\overline{A + B}$	A + B	A + B plus 1
H	H	L	L	Logic '0'	minus 1 (2s comp.)	Zero
L	L	H	L	$\overline{A} + B$	A plus A•B	A plus A•B plus 1
H	L	H	L	B	A•$\overline{B}$ plus (A + B)	A•$\overline{B}$ plus (A + B) plus 1
L	H	H	L	$\overline{A \oplus B}$	A plus B	A plus B plus 1
H	H	H	L	A•$\overline{B}$	A•$\overline{B}$ minus 1	A•$\overline{B}$
L	L	L	H	$\overline{A}$•$\overline{B}$	A plus A•$\overline{B}$	A plus A•$\overline{B}$ plus 1
H	L	L	H	A⊕B	A minus B minus 1	A minus B
L	H	L	H	$\overline{B}$	A•$\overline{B}$ plus (A + B)	A•$\overline{B}$ plus (A + B) plus 1
H	H	L	H	A•$\overline{B}$	A•$\overline{B}$ minus 1	A•$\overline{B}$
L	L	H	H	Logic '1'	A plus A (2 × A)	A plus A (2 × A) plus 1
H	L	H	H	A + B	A plus (A + $\overline{B}$)	A plus (A + $\overline{B}$) plus 1
L	H	H	H	A + $\overline{B}$	A plus (A + B)	A plus(A + B)plus 1
H	H	H	H	A	A minus 1	A

d. A input data true; B input data inverted

'F182 Carry Lookahead Generator

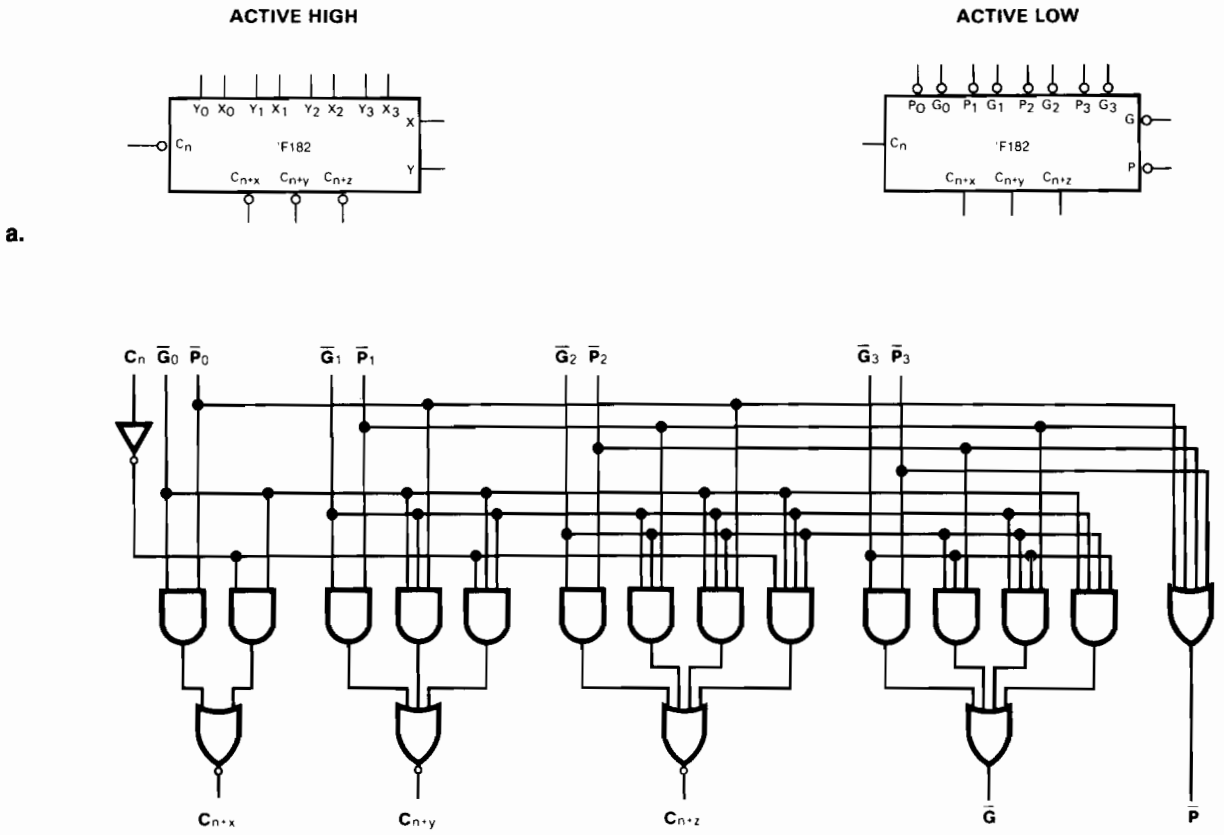


Figure 5-7

The 'F181 ALU can be used in a variety of carry modes. The simplest of these is in a ripple carry mode where the Carry In (C_{in}) of an ALU is driven by the Carry Out signal (C_4) from the previous ALU. This method of propagating the carry is slow for large word lengths but has the advantage that additional carry circuits are not required; if several levels of lookahead are permitted and extra logic is used, the speed of the ALU can be improved. The 'F181 gives the auxiliary carry functions, Carry Generate and Carry Propagate, which can be used with the 'F182 to give complete carry lookahead or ripple block lookahead. In this latter mode, the ALU is split into 16-bit blocks, each with its own lookahead with carries allowed to ripple between the blocks. The 'F182 accepts up to four sets of Carry Generate and Carry Propagate functions and a Carry In and provides the three Carry Out signals required by the ALUs and also the next level auxiliary functions. These auxiliary functions generated by the carry lookahead circuit allow further levels of lookahead. Unfortunately, to satisfy signal polarities, a penalty of two gate delays is incurred for each level of lookahead, and the auxiliary functions are rarely used over more than two levels of lookahead. The logic symbols and logic diagram of the 'F182 carry lookahead circuit are shown above. The auxiliary logic functions in the active HIGH case are not Carry Generate and Carry Propagate—they have been labeled X and Y, respectively. In this logic design the auxiliary functions are used to generate the three Carry Out signals and the two auxiliary functions required for further levels of lookahead. They are connected in the same manner as the active LOW case.

Carry Lookahead

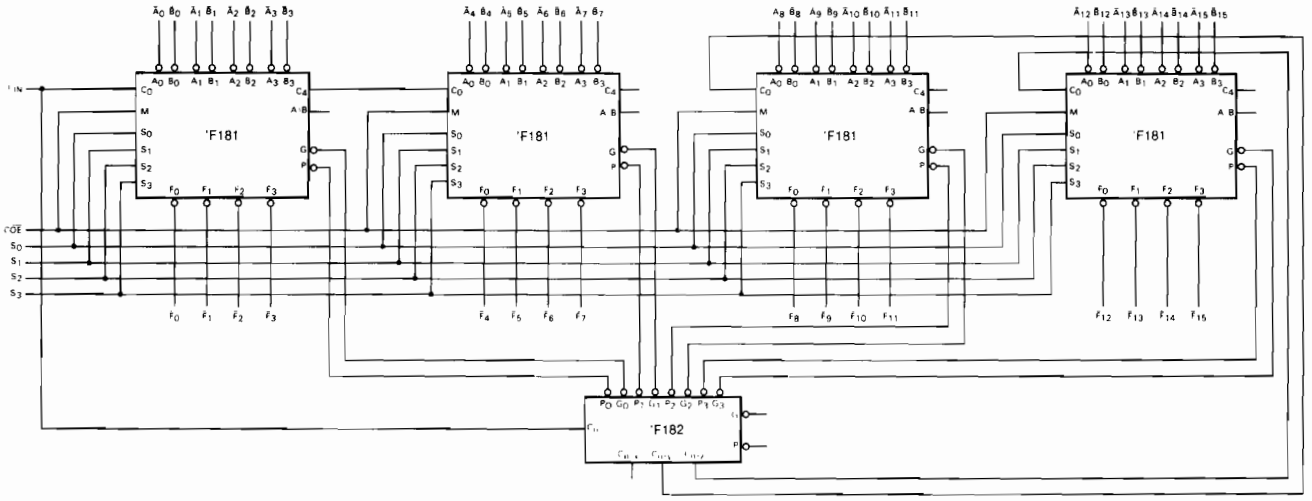
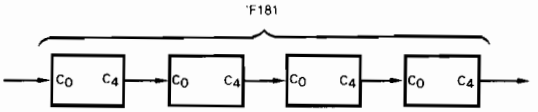


Figure 5-8

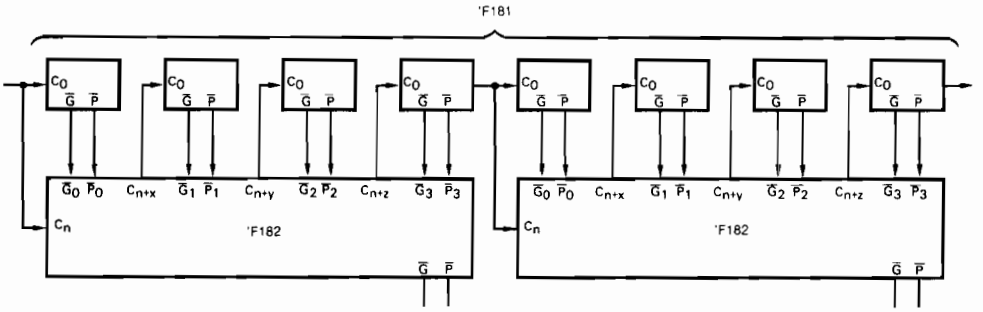
A single carry lookahead circuit is used with four 'F181 ALUs to perform arithmetic operations with complete carry lookahead over 16-bit words. For word lengths of 20 and 24 bits, the fastest speed is achieved by using only a single 'F182 as above and letting the carry ripple through the additional one or two 'F181s. For word lengths of 28 and 32 bits, the fastest speed is achieved by using two 'F182s, constructing two blocks similar to the 16-bit block above and letting the carry ripple from the first block to the second. Only when the word length exceeds 32 bits is there a speed advantage in using three levels of carry lookahead.

'F181 Carry Methods

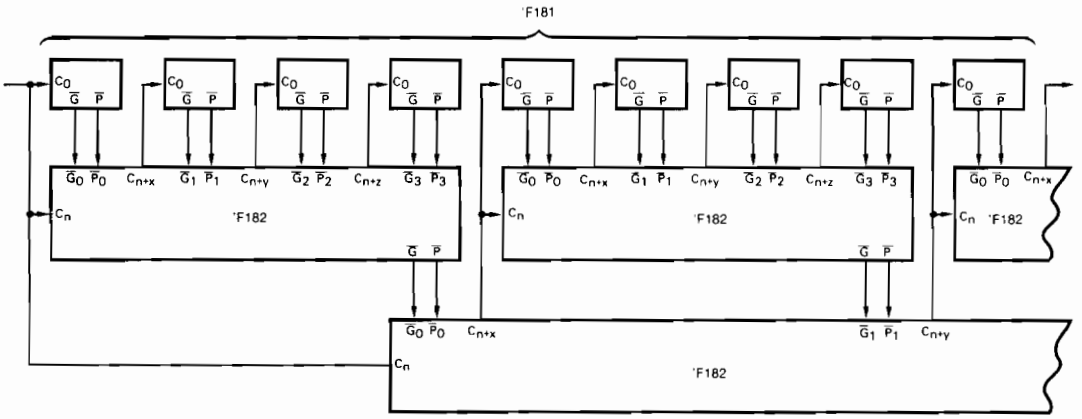
a. Single Level (Internal) Carry Lookahead



b. Two Level Carry Lookahead



c. Three Level Carry Lookahead



d. 32-Bit ALU

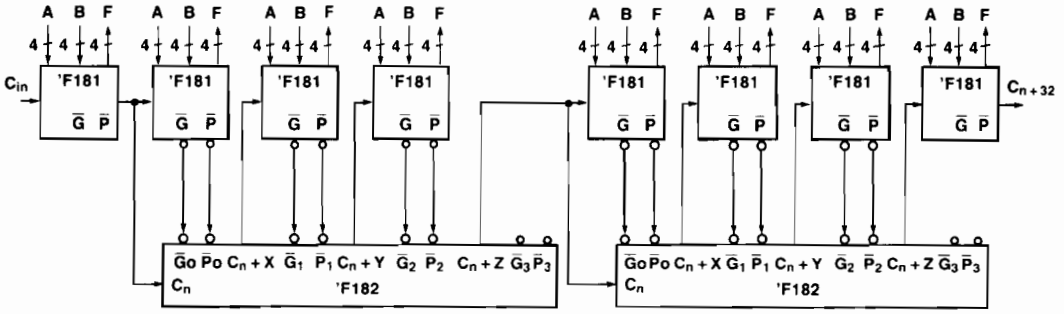


Figure 5-9

There are various methods of using the 'F182 with ALUs to perform arithmetic operations over large word lengths. The use of a small number of carry lookahead packages decreases considerably the delay for typical word lengths used in digital systems.

A faster implementation of the 32-bit ALU is shown in Figure 5-9d. This takes advantage of several faster paths in the 'F181 and 'F182 combination shown.

Table 5-3 Speed Table

		FAST	Schottky	AS
Typical	A + B	26ns	44ns	24ns
	A – B	27.7ns	45ns	26ns
Worst Case	A + B	38ns	55.5ns	37ns
	A – B	40ns	56.5ns	39ns
Total I_{CC} (mA)		381	1,204	1,680

Arithmetic with ALUs

The 'F181 can be used as an arithmetic element in all the common binary number representations. The basic concepts as discussed above also apply to the 'F381. The most difficult number representation is sign magnitude. The 'F181 is more flexible than the 'F381, but additional peripheral logic must be used to decode the desired functions required by the select lines and the Carry In at the first stage. An additional arithmetic operation of the 'F181 as compared to the 'F381 is "double A" multiplication. It is often possible to use the select code on the ALU to perform additional decoding effectively; for example, if a control signal is to select between "add A and B" when S is HIGH and "pass A" when S is LOW, then for the active HIGH case S_0 is tied to S_3 to form S, and S_1 and S_2 are tied LOW. This type of operation (Add or Pass) is useful in multiplication routines.

Comparison Functions Using the 'F181

Several comparison functions can be performed with the 'F181 by using the A = B and C_4 outputs. The A = B output is better described as "F = 0," since this output goes HIGH any time all the F outputs are HIGH. Therefore, the outputs are not only used for comparing A = B during a subtract operation, but also to ascertain that the function outputs are all HIGH after any arithmetic or logic operation. In the PASS operation, the output indicates that one of the operands is equal to zero. In the Exclusive-OR operation, it indicates that the two operands are identical. In the EQUIVALENCE operation, it indicates that the two operands are complementary. For unsigned numbers with the most significant bit positive, the Carry Out of the ALU indicates relative magnitude. Table 5-4 lists the various comparison functions which can be performed in active HIGH and active LOW logic.

Table 5-4 Function Table

Output	State	Operation	Active LOW Logic	Active HIGH Logic
A = B	H	A minus B	A = B	A = (B minus 1)
	H	$\overline{A \oplus B}$	A ≠ B	A = B
	H	A ⊕ B	A = B	A ≠ B
Carry Out	H	A minus B	A ≧ B	A < B
($\overline{C}_4$ for active HIGH operands)	L	A minus B	A < B	A ≧ B
(C_4 for active LOW operands)	H	A minus B minus 1	A > B	A ≦ B
	L	A minus B minus 1	A ≦ B	A > B

4-Bit Accumulator

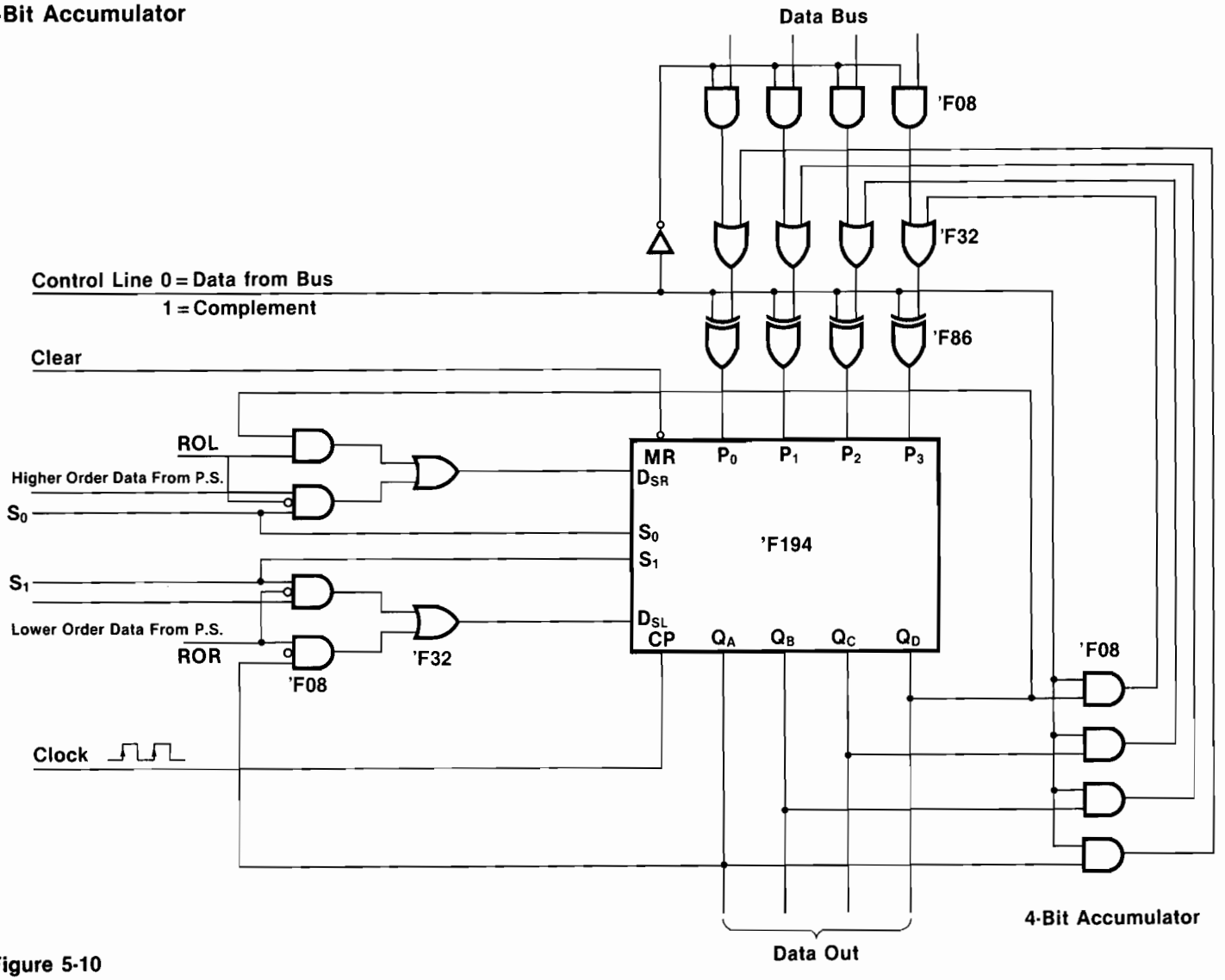


Figure 5-10

The above diagram illustrates a method of connecting the 'F194 4-bit bidirectional shift register as a 4-bit accumulator. The 'F194 can also be cascaded with another register in order to form an 8-bit accumulator. In this manner an accumulator can be created which can operate as an adder or subtractor and can be utilized in a variety of applications.

An 'F08 AND gate is used in conjunction with the 'F86 Exclusive-OR gate and an 'F32 OR gate in order to derive the ones complement. This is achieved by placing the control line at the logic 1 level; in this manner ones complement arithmetic may be performed. All shift and rotate operations occur on the positive edge of the clock pulse and a hold status can also be initiated by placing lows on both mode control inputs (S_0, S_1) simultaneously. Data present at the outputs may be asynchronously cleared by placing a low logic level on the $\overline{MR}$ input.

The schematic diagram indicates how the outputs can be brought in from a previous stage (P.S.) in the case where more than four bits are required.

Table 5-5 Function Table

S_0	S_1	Operation
0	0	Hold
0	1	Shift Left/Rotate Left
1	0	Shift Right/Rotate Right
1	1	Load

Multipliers

16 x 16 Multiply (Low-Cost Solution)

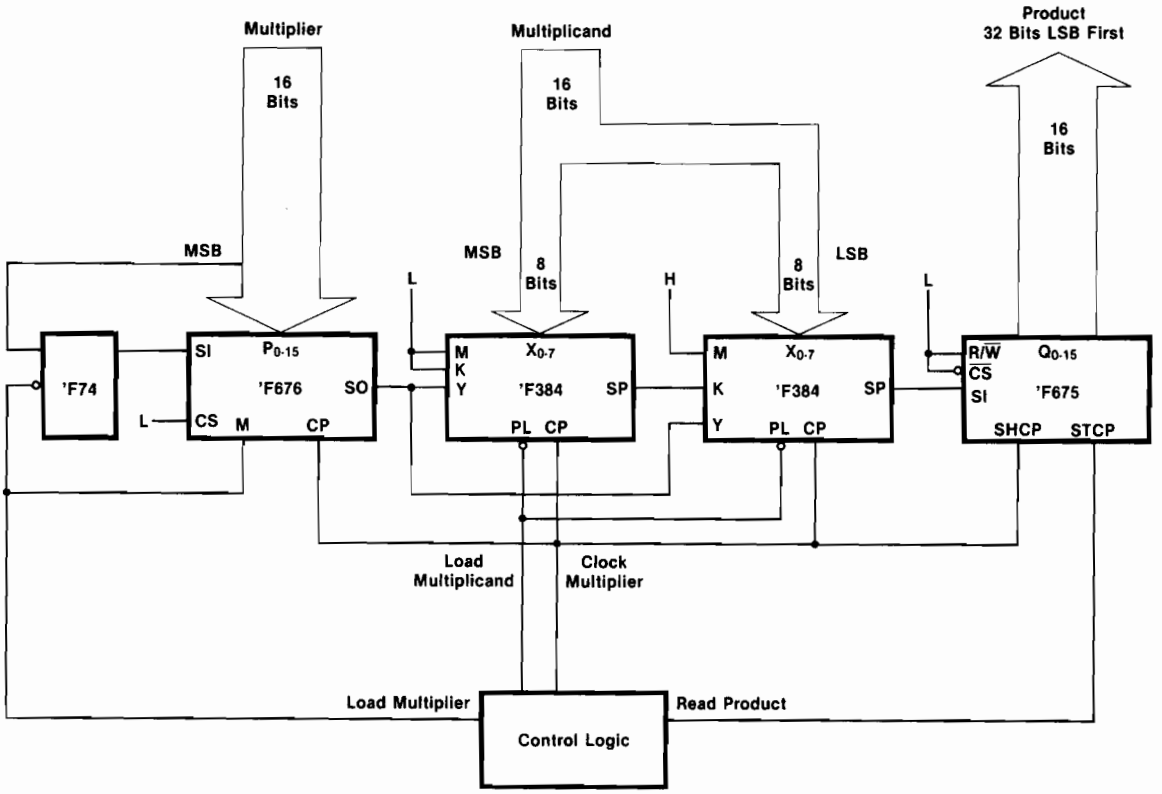


Figure 5-11

Many microprocessor systems do not require the speed or justify the cost of LSI multipliers, and would benefit from a hardware 16 x 16 multiplier. Shown here is a 16 x 16 multiplier that will give a 32-bit result in under 1μs. The 'F384 serial multipliers are connected to form a 16 x 16 multiplier. Thirty-four clock cycles are required to achieve a 32-bit product. The high-speed 'F676 shift register and a 'F74 are used to hold the sign extended multiplier. The 16-bit multiplicand can be applied directly to the 'F384 multipliers. The product holding shift register 'F675 can hold all 32 bits of product in a single chip. Maximum clock frequency is 35MHz over the commercial temperature range.

4 x 4 Multiplier

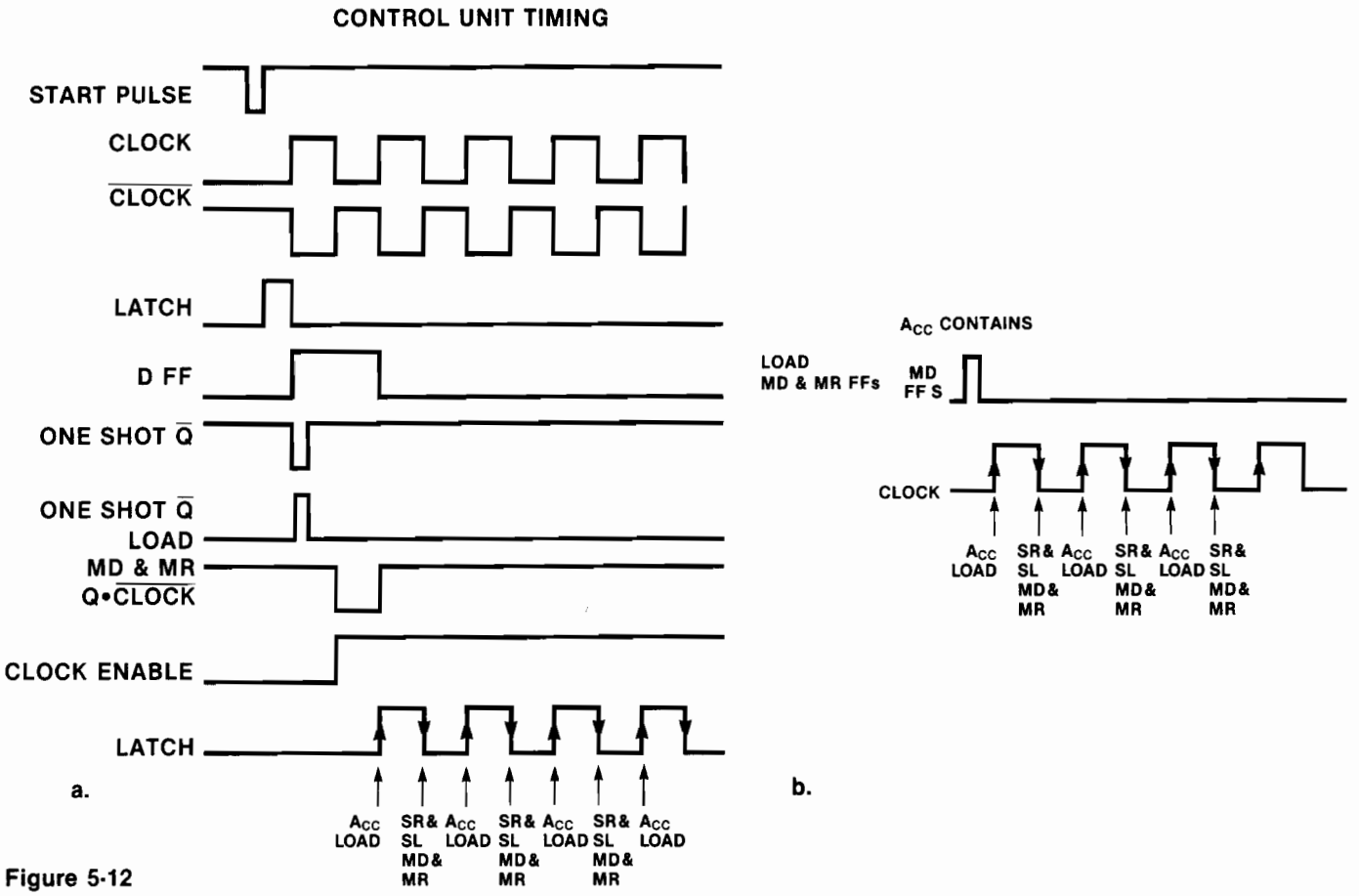


Figure 5-12

This is a description of a multiplier circuit which utilizes the 'F109 JK and the 'F112 JK flip-flops (Figure 5-13). This multiplier can also be expanded to an 8 x 8 or even a 16 x 16 multiplier. However, the number of packages becomes large and the increased power consumption must also be a consideration when replacing the shift sections with MSI shift registers.

In the circuit, before a multiplication can be performed, all JK flip-flops must be initially cleared. This is accomplished by using a 5ns negative-going pulse referred to as the start pulse. This pulse is applied to all direct clear ($\overline{CD}$) flip-flop inputs and causes the Q outputs to reflect a zero or LOW state. Data must be presented on the inputs of the steering gates provided for the multiplication prior to the positive load pulse. When loading occurs, inverted data appears on the direct set ($\overline{SD}$) inputs of the shift left and shift right registers. In the schematic the shift right register is for the multiplier while the shift left register is for the multiplicand. As the clock causes the information to be shifted, it is applied to the 'F283 adders. The adder outputs constitute a partial sum that consists of four bits which are meaningful after the first clock pulse. These bits will appear on the outputs of adder "A." All zeros will appear on the outputs of adder "B." Thus, the "multiplication" has begun. The accumulator is clocked via an 'F00 NAND gate when a HIGH state occurs on the Q output of flip-flip 1. Accumulator loading occurs on negative clock transitions; shift registers also operate on the negative clock.

With a 33MHz clock rate this circuit is capable of performing a 4 x 4 multiplication in five clock pulses or approximately 150ns.

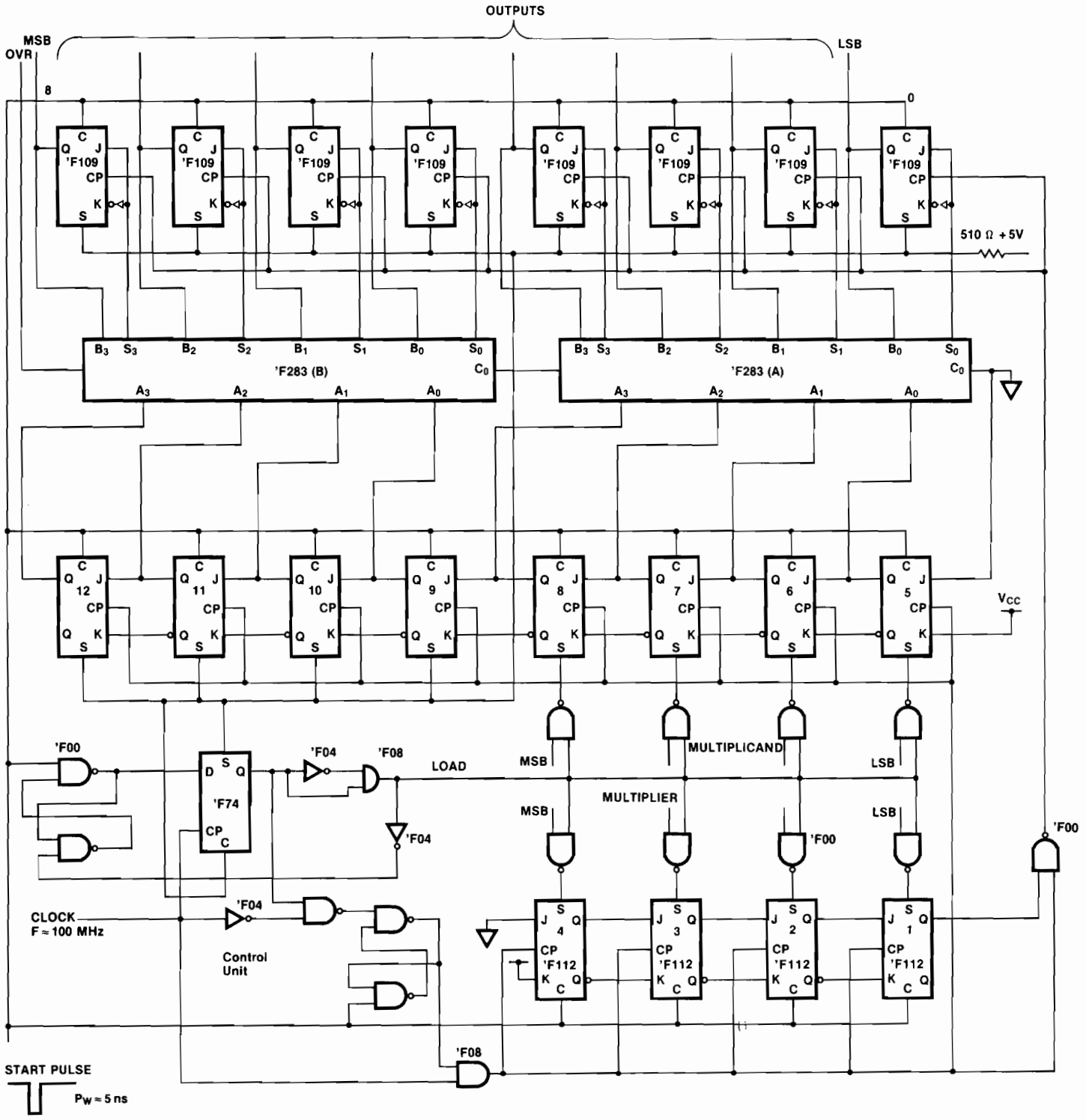


Figure 5-13 4 x 4 Multiplier

Comparator Systems

Comparator systems fall into two classes:

- Identity comparators, which detect whether or not two words are identical;
- Magnitude comparators, which detect which of two words is larger. Magnitude comparators are more complex and tend to be slower.

All comparators are defined in binary terms, but they can obviously be used with BCD or any other monotonic code without change.

Identity Comparators

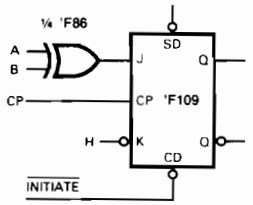


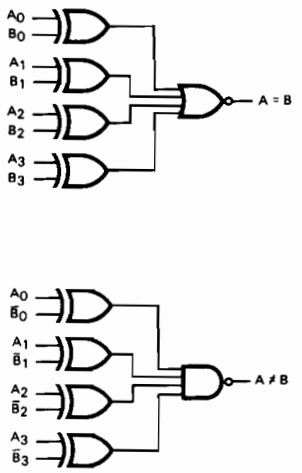
Figure 5-14 Bit Serial Operation

One Exclusive-OR and one flip-flop form a serial identity comparator. The flip-flop must start out reset. As long as the A and B inputs are identical, the output of the Exclusive-OR is LOW, leaving the flip-flop in its reset state. When $A \neq B$ the flip-flop is set and stays set until a new cycle is initiated by asynchronously clearing the flip-flop. The state of Q after the last bit has been clocked indicates the result of the comparison:

Q: $A \neq B$ $\quad\quad$ $\overline{Q}$: $A = B$

Obviously the bit sequence does not affect the identity comparison.

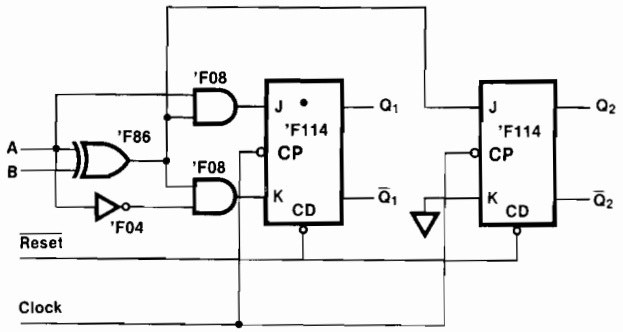
Figure 5-15 Parallel Operation



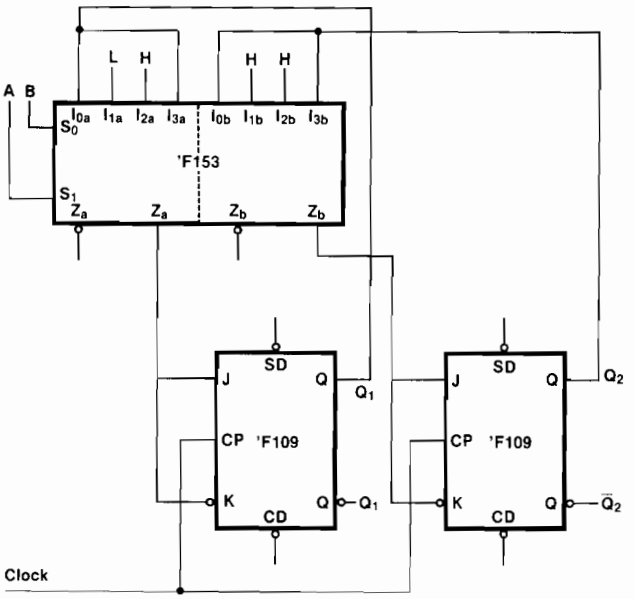
Parallel identity comparison is most efficiently performed with Quad Exclusive-OR gates with outputs NORed or NANDed. The NAND configuration requires opposite polarities of the two operands.

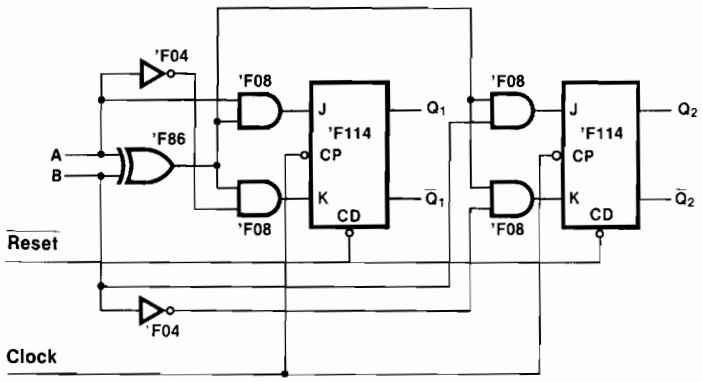
5

Bit Serial Magnitude Comparison, Least Significant Bit First



a. Q_1: A>B Q_2: A≠B
$\overline{Q}_1$: A≤B $\overline{Q}_2$: A=B





b. Q_1: A>B Q_2: A<B
$\overline{Q}_1$: A≤B $\overline{Q}_2$: A≥B

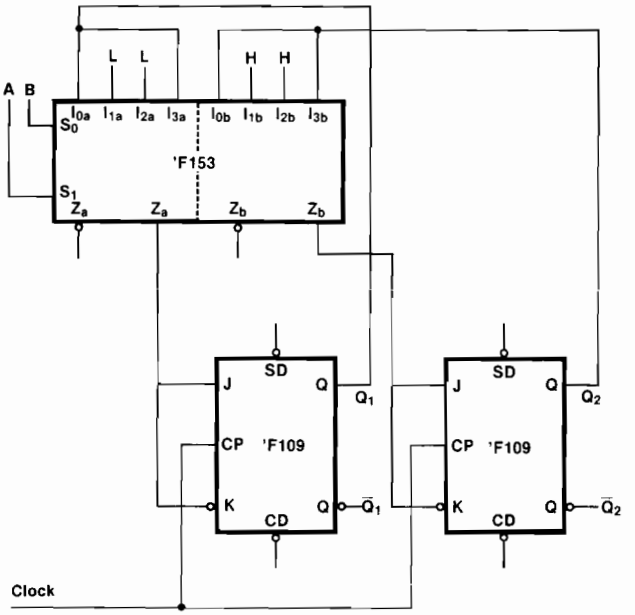


Figure 5-16

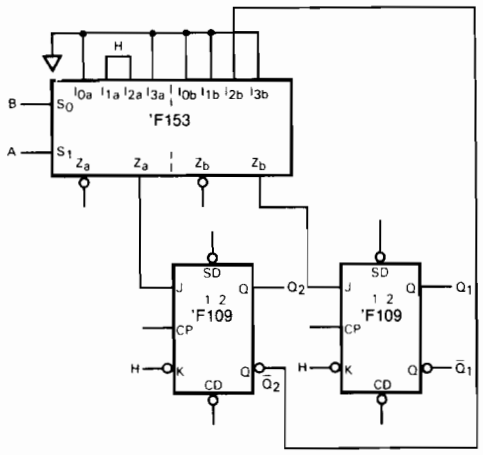
Magnitude comparison discriminates between three possible conditions: A>B, A<B, and A=B, usually encoded on two output signals.

A serial magnitude comparator for LSB first (Figure 5-16a) is most efficiently implemented by either a dual 4-input multiplexer ('F153) and a dual flip-flop ('F109), or by an Exclusive-OR gate and a dual flip-flop ('F114). Note that the 'F109 master/slave flip-flop requires stable inputs during the entire clock LOW period.

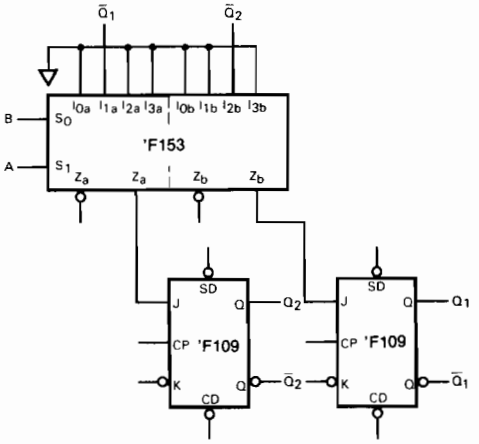
Assuming active HIGH notation,
Q_1 is set by $A \bullet \overline{B}$, reset by $\overline{A} \bullet B$, unaffected by $A \bullet B$ or $\overline{A} \bullet \overline{B}(A = B)$
Q_2 is set by A≠B, unaffected by A=B.

Thus, if both flip-flops start out reset, their state after clocking in the most significant bit indicates the result of the comparison. A slight rearrangement of the same basic circuit (Figure 5-16b) generates a different set of outputs.

Magnitude Comparison, Most Significant Bit First



a. Q_1: A>B Q_2: A≠B
$\overline{Q}_1$: A≤B $\overline{Q}_2$: A=B



b. Q_1: A>B Q_2: A<B
$\overline{Q}_1$: A≤B $\overline{Q}_2$: A≥B

Figure 5-17

Magnitude comparison is also possible when the serial words come in "backward," with their most significant bits first (Figure 5-17a). In this case, the first bit where A differs from B determines the result. This circuit sets Q_1 when $A \bullet \overline{B} \bullet \overline{Q}_2$; i.e., if A>B and all previous bits have been A=B. Q_1 is unaffected under all other conditions. It sets Q_2 if A≠B, but does not reset it until a new comparison is initiated by clearing both flip-flops.

A slight rearrangement of basically the same circuit (Figure 5-17b) generates a different set of outputs.

'F524 8-Bit Registered Comparator

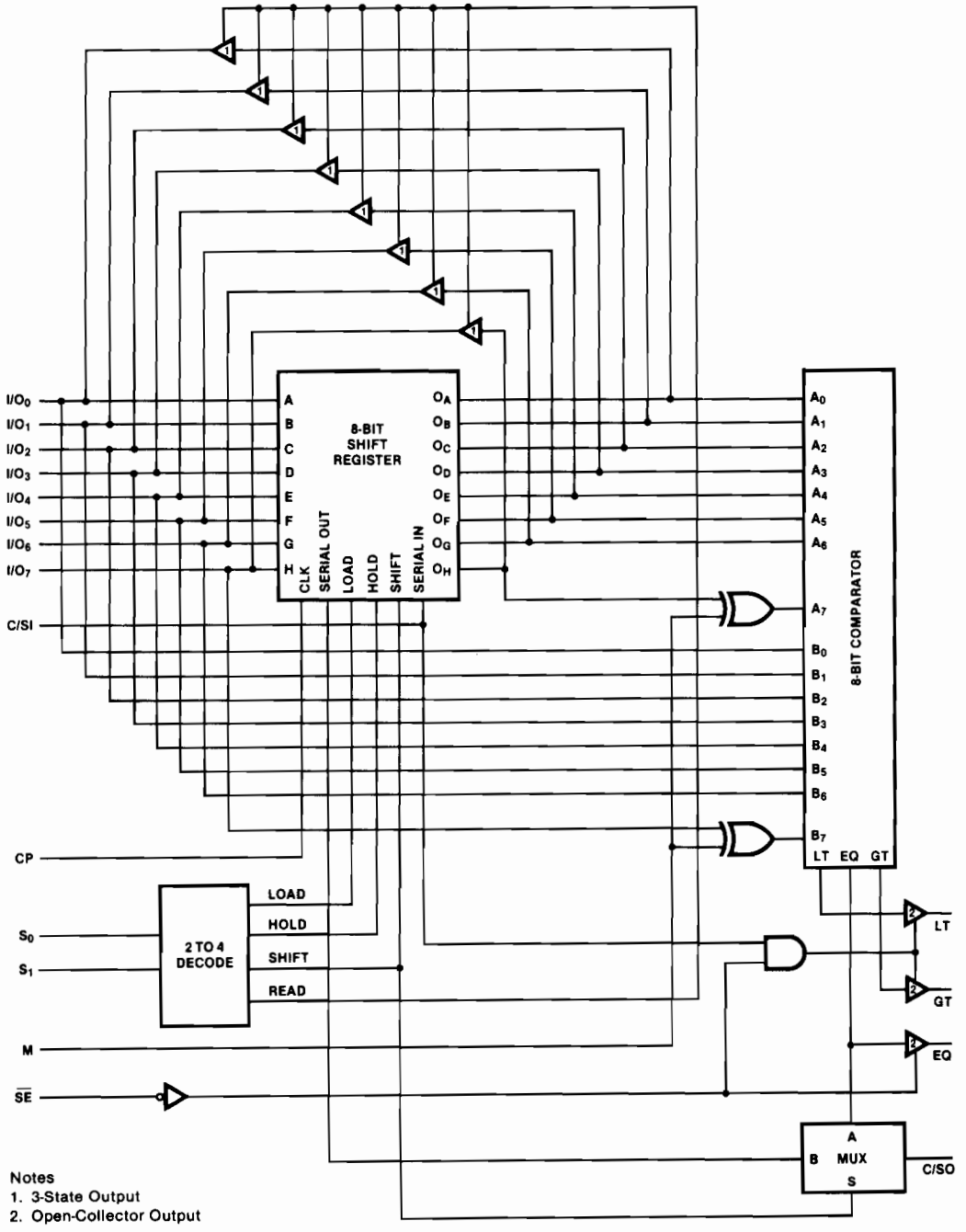


Figure 5-18

The 'F524 is an 8-bit serial/parallel load/read register whose outputs are connected to an 8-bit magnitude comparator. The other side of the comparator is connected to the I/O port. Mode lines S_0 and S_1 control the operation of the register: load, hold, shift and read. Open collector outputs on LT, GT and EQ outputs can be cascaded. Mode input selects magnitude or twos complement comparison.

'F524 Cascading for Comparing Longer Words

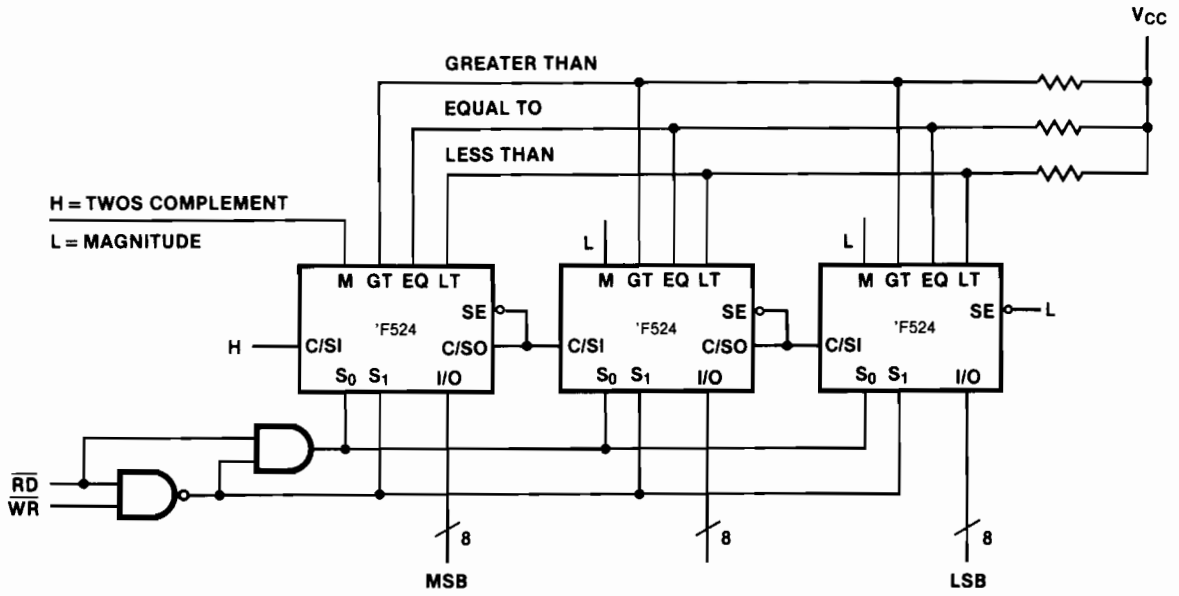


Figure 5-19

This diagram shows the 'F524 cascaded for a 24-bit compare. Note that the $\overline{SE}$ line is tied LOW on the least significant device thereby enabling the least significant output. The cascade line C/SI should be held HIGH on the most significant device. In the case of equality, C/SO will go HIGH lifting $\overline{SE}$ HIGH, disabling its equality output. At the same time the next C/SI input will go HIGH enabling the next most significant device. Thus, the decision is passed down the chain until an inequality is detected or the least significant device decides that the whole word is equal. For cascading n devices the worst case compare is (32.5 + (12(n-2))ns.

Error Detection/Correction

Whenever digital data is transferred from one location to another, there is a probability for error due either to device failure or noise. There are numerous ways to handle errors at the system level. Some systems detect errors and request retransmission of data. In other systems retransmission may be impossible or prohibitively expensive. In such systems the receiving equipment must not only be able to detect, but also correct the error.

Both error detection and error correction rely on the transmission of redundant information. This requires additional bits of data and lowers the overall efficiency of transmission. In parallel systems additional wires, transmitters, and receivers are required, whereas serial transmission systems use additional time to transmit the redundant information. All these methods cannot completely eliminate errors, but as the percentage of redundant data bits increases, the probability of undetected or uncorrected errors decreases.

The simplest and most common method of dealing with errors is the addition of a single extra bit, called a parity bit. The parity bit is chosen such that the total number of ones in the word (counting the parity bit) is odd (in an odd parity system) or even (in an even parity system). Odd parity is generally preferred, since it insures at least one "1" in any word. At the receiving end, the parity of the word is examined. If any single bit in the word was changed, the detector indicates wrong parity. However, if an even number of errors occurs, this simple method cannot detect it. The parity bit provides only single error detecting.

Parity Generation

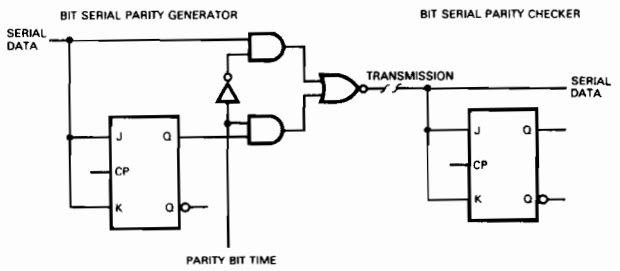


Figure 5-20

In a serial parity generator, a flip-flop is toggled for every "1" in the data word and the state of this flip-flop is inserted as a trailing parity bit. On the receiving side the parity checker has an equivalent flip-flop. Its state is interrogated after the data has been received. Both circuits are easily adapted for odd or even parity systems.

For parallel systems it is necessary to generate the modulo 2 sum of many inputs simultaneously. This requires an array of cascaded Exclusive-OR circuits. The 'F280 9-bit parity checker/generator is specifically designed for this function.

Error Correction[1]—Hamming Codes

A parity bit can only detect single errors. It cannot reliably detect multiple errors and it cannot correct single errors. A single redundant bit does not carry enough information to do so. However, it is possible to add more redundant information to the data, formulated such that errors are not only detected, but also corrected.

A data word containing an error-correcting field of redundant information is called a Hamming code. It uses a series of parity bits generated and arranged so that a unique set of parity errors results from an error in any given bit position. For example, three redundancy bits can have a total of eight different states. Since one of these states must indicate "no error," the other seven states can be used to locate an error in any one of seven transmitted bits. Three of the transmitted bits are the redundancy bits themselves, leaving four data bits in which an error can be uniquely detected, and also corrected. The coding of the parity bits is done conveniently so the pattern of parity errors is the binary address of the bit in error. In general, a Hamming code contains 2^m-1 bits, m of which are the Hamming or check bits, $2^m - m - 1$ are the data bits.

Total Bits	Hamming Bits	Data Bits
7	3	4
15	4	11
31	5	26

Thus three additional parity (Hamming) bits can provide single error correction for 4-bit data words. The seven bits are arranged in the following way:

$P_0P_1D_0P_2D_1D_2D_3$

where D_0, D_1, D_2, D_3 are the four data bits;

P_0 is odd parity over bits D_0, D_1, D_3;

P_1 is odd parity over bits D_0, D_2, D_3;

P_2 is odd parity over bits D_1, D_2, D_3.

At the receiving end the three parity bits are again generated from the data bits using an identical scheme. Then these three parity bits are compared with the three transmitted parity bits. If they all match, there was no single error. If they differ, the pattern of mismatches is interpreted as a binary address of the bit in error.

A practical system avoids the additional comparison and generates the error address (E_{0-2}) by including the received parity bits in the parity check:

E_0 is odd parity over bits P_0, D_0, D_1, D_3;
E_1 is odd parity over bits P_1, D_0, D_2, D_3;
E_2 is odd parity over bits P_2, D_1, D_2, D_3.

This Hamming code can detect and correct single errors, but it will fail on double errors—it would correct the wrong bit. However, if one more overall parity bit is added, it is also possible to detect (but not correct) double errors. When the receiver finds the overall parity check correct and the error address is zero, there was no error. If the overall parity check is wrong and the error address is not zero, there was a single error which can be corrected. However, if the overall parity check is correct, but the error address is not zero, then there was a non-correctable double error.

[1]For a detailed description of the theory behind and the applications of error correcting codes, see Peterson and Weldon, ERROR CORRECTING CODES, Second Edition, The MIT Press, Cambridge, MA, 1972.

Cyclic Checks for Error Detection

Error detection schemes using parity checks are well known. A parity check on a character is called "vertical" parity and a check on corresponding bits of every character in a message (data block) is called "longitudinal" parity. Used together, they provide a satisfactory checking scheme. The measure of protection provided is better than using vertical or longitudinal parity alone. However, the level of redundancy to achieve this protection is relatively high. For example, if there are x bytes in a message each consisting of seven data bits and one parity bit, the ratio of number of check bits to data bits is 1/7.

Another checking scheme exists called polynomial or cyclic coding that can be designed to perform with higher efficiencies than traditional parities. The level of protection achieved with a 16-bit cyclic check is probably satisfactory for most practical purposes. When used with a data block consisting of 7x data bits, ratio of check to data bits is only 16/7x. The ratio approaches a limit of zero as x increases. This high efficiency is inducing designers to incorporate cyclic check schemes in modern data communication and peripheral equipment such as tapes and discs. Theoretical knowledge necessary for cyclic check implementation has existed for several years. However, widespread use has begun only recently in designs using integrated circuits. Because it is relatively new, many designers do not have the needed exposure to cyclic schemes and tend to shy away.

This discussion is intended to familiarize uninitiated readers with the algebraic concepts required to design circuits for implementing cyclic check schemes. Not only are these concepts of value to the hardware designer, but also to the diagnostic programmer who must generate the code to check the implemented logic for validity and failures.

Polynomial Notation and Manipulation

A very convenient way of expressing a bit stream (message) consisting of K bits is to think of it as a polynomial in a dummy variable x with K terms. The bits of the message are the coefficients in the polynomial. Thus, if 100100011011 is the message, it may be written as:

$$M(x) = 1(x)^{11} + 0(x)^{10} + 0(x)^9 + 1(x)^8 + 0(x)^7 + 0(x)^6 + 0(x)^5 + 1(x)^4 + 1(x)^3 + 0(x)^2 + 1(x)^1 + 1(x)^0$$

or

$$M(x) = x^{11} + x^8 + x^4 + x^3 + x + 1$$

To compute the cyclic check on a message, another polynomial P(x), called a generating polynomial, is chosen. The degree "r" of the P(x) is such that it is greater than zero but less than the degree of M(x). Moreover, P(x) has a non-zero coefficient in the x^0 term. Thus it is clear that for a given message length, more than one generating polynomial of desired length can be specified. Fortunately, several accepted standard generating polynomials exist. Most common are CRC-16 and CRC-12 which were originally proposed for the IBM binary synchronous communications.

CRC-16 is a 16-bit check resulting from a generating polynomial $x^{16} + x^{15} + x^2 + 1$ and CRC-12 is a 12-bit check resulting from $x^{12} + x^{11} + x^3 + x^2 + x + 1$. Theory suggests that use of CRC-16 and CRC-12 will catch all messages with an odd number of errors, all with a single error burst of less than 16 or 12 bits respectively and most of the few messages with larger error bursts.

Cyclic check computation involves manipulating M(x) and P(x) using laws of ordinary algebra, except that modulo 2 arithmetic is used. Because modulo arithmetic yields the same result for addition and subtraction, it is necessary only to consider three operations involving polynomials—addition, multiplication and division.

Addition of two polynomials, $x^6 + x^5 + x^2 + 1$ and $x^5 + x^4 + x^3 + x^2$, yields $x^6 + x^4 + x^3 + 1$ as shown below:

$$\begin{array}{rl} x^6 + x^5 + 0 + 0 + x^2 + 0 + 1 &= 1100101 \\ x^5 + x^4 + x^3 + x^2 + 0 + 0 &= 111100 \\ \hline x^6 + 0 + x^4 + x^3 + 0 + 0 + 1 &= 1011001 \end{array}$$

Multiplication of two polynomials, $x^7 + x^6 + x^5 + x^2 + 1$ and $x + 1$, results in $x^8 + x^5 + x^3 + x^2 + x + 1$:

$$\begin{array}{rl} (x^7 + x^6 + x^5 + x^2 + 1)(x + 1) &= (11100101) \times 11 \\ x^8 + x^7 + x^6 + 0 + 0 + x^3 + 0 + x + 0 &= 111001010 \\ x^7 + x^6 + x^5 + 0 + 0 + x^2 + 0 + 1 &= 011100101 \\ \hline x^8 + 0 + 0 + x^5 + 0 + x^3 + x^2 + x + 1 &= 100101111 \end{array}$$

It is interesting to note that multiplication of a polynomial by x^m results in a shifted bit pattern which is identical to the original except for zeros in the lower m positions. For example:

$x^5(x^{11}+x^{10}+x^8+x^4+x^3+x+1) = x^{16}+x^{15}+x^{13}+x^9+x^8+x^6+x^5$
where $x^{11}+x^{10}+x^8+x^4+x^3+x+1 = 110100011011$
and $x^{16}+x^{15}+x^{13}+x^9+x^8+x^6+x^5 = 11010001101100000$

Dividing $x^{13}+x^{11}+x^{10}+x^7+x^4+x^3+x+1$ by $x^6+x^5+x^4+x^3+1$ results in a quotient of $(x^7+x^6+x^5+x^2+x+1)$ and a remainder of (x^4+x^2) as shown below. Practically, it might be easier to divide by longhand if the bit pattern is used rather than the polynomial.

$x^{13}+x^{11}+x^{10}+x^7+x^3+x+1 = 10110010011011$;
$x^6+x^5+x^4+x^3+1 = 1111001$

```
            11100111
          ------------------
11110011  10110010011011
          1111001
          -------
           1000000
           1111001
           -------
            1110010
            1111001
            -------
             1011110
             1111001
             -------
              1001111
              1111001
              -------
               1101101
               1111001
               -------
                10100
```

Thus, $Q(x) = 11100111 = x^7+x^6+x^5+x^2+x+1$;
$R(x) = 10100 = x^4+x^2$

Cyclic Check-Computing Procedure

To compute a check on M(x), a generating polynomial P(x) is chosen as mentioned earlier. Steps involved in check computation are as follows:

a) Message polynomial M(x) is multiplied by x^r where r is the degree of P(x). As noted earlier, this process yields zeros in the lower r positions of M(x). These vacated positions are in preparation for the r check bits that will be appended to the message. Also note that this process does not alter the message bit pattern.

b) The result obtained from step (a) is divided by P(x). This gives a quotient Q(x) and a remainder R(x). The remainder will be r bits or less.

c) The quotient is discarded and the remainder is added to the result of step (a). The remainder is the check. The message with this remainder at the tail end constitutes the transmitted polynomial T(x).

The following example illustrates the computation procedure. Let $M(x) = x^{11}+x^{10}+x^8+x^4+x^3+x+1 = 110100011011$ and $P(x) = x^5+x^4+x^2+1 = 110101$. Thus, r = 5 and $x^rM(x) = x^{16}+x^{15}+x^{13}+x^9+x^8+x^6+x^5 = 11010001101100000$

```
  x^rM(x) = 11010001101100000
  ---------------------------
    P(x) = 110101
```

Carrying this division, Q(x) = 100001100111 and R(x) = 1011.

Transmitted message T(x) is obtained by adding R(x) to $x^rM(x)$:

```
  x^rM(x) = 11010001101100000
    R(x) = 01011
  ---------------------------
    T(x) = 11010001101101011
  ---------------------------
    Data Check
```

Note that transmission occurs from left to right; data thus is unmodified and check bits follow at the end.

Data Validation at the Receiver

The transmitted polynomial arrives at the receiver modified or unmodified depending on whether transmission has encountered errors or not. Clearly, one of the ways by which the receiver can ensure data validity is to recompute the check bits on the message using the same generator polynomial and compare them with the received check bits. If they agree, it is assumed that received data is good.

Alternately, the receiver can divide the complete received polynomial by the same generator polynomial P(x). If there are no errors, it can be shown that this division results in zero remainder. This property can be easily verified by long division of T(x) = 11010001101101011 by P(x) = 110101. If the division results in a non-zero remainder, it can be assumed that T(x) has been modified by errors. This may be verified by introducing error and performing the division. The process of dropping and picking bits can be viewed as adding another polynomial E(x) (error polynomial) to T(x).

For example, if $T'(x) = 10010001101101011$ is received, instead of Tx), $T'(x) = T(x) \otimes E(x)$ can be written where $E(x) = 01010001101101011$. It follows then that if $T'(x)$ is exactly divisible by P(x), the receiver is blind and indicates no errors. This only happens if E(x) is exactly divisible by P(x). Knowing the characteristics of the transmission medium, it is advisable to choose such a generating polynomial that the probability of error patterns occurring that are divisible by P(x) is extremely low. The process of not detecting such errors is somewhat analogous to the erroneous validity indication in normal parity schemes where multiple bit errors may cancel each other's contribution to the check.

Basic Polynomial Divider

Consider longhand division of the polynomial $x^{16} + x^{15} + x^{13} + x^9 + x^8 + x^6 + x^5$, i.e., 11010001101100000, by another polynomial $x^5 + x^4 + x^2 + 1$, i.e., 110101.

```
          100001100111
        ________________________
110101  11010001101100000
          110101
        ________________________
             101101
             110101
           ______________________
             110001
             110101
           ______________________
              100000
              110101
            _____________________
              101010
              110101
            _____________________
               111110
               110101
             ____________________
                 1011
```

From this example, longhand division procedure can be summarized: align the most significant bits of the partial remainder and divisor, borrowing from the dividend as required. This implies aligning the divisor and dividend to start the division process. Then subtract the divisor from the partial product using modulo 2 arithmetic. When all bits in the dividend are processed, the result is the remainder.

Subtraction in modulo 2 of two bits is the same as performing an Exclusive-OR operation and alignment of bits suggests a shift operation. Consider two registers as shown in Figure 5-21.

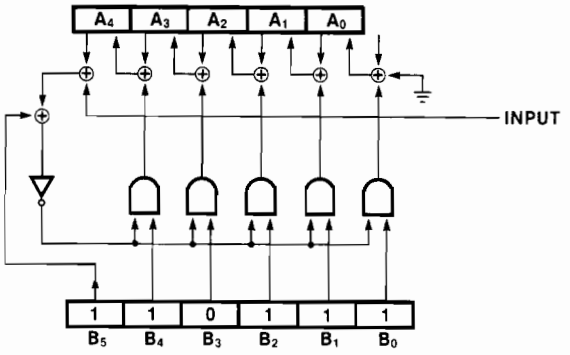


Figure 5-21 Conceptual Polynomial Divider

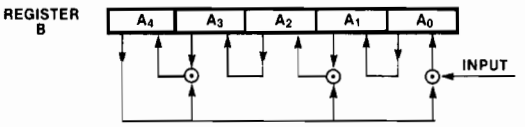


Figure 5-22 Polynomial Divider for $x^6 + x^5 + x^2 + 1$

Assume that register A is initially clear and register B contains 110101, which is the divisor bit pattern. Also imagine that the dividend serially enters the network as input (most significant bit first), in response to a clock signal that operates register A. As long as A_4 is cleared and B_5 is set, the AND gates are inhibited. This establishes a connection between A_4 input and A_3 output, A_3 input and A_2 output, etc. Thus, register A serves as a "shift left" register. When clocked with the dividend as serial input, the most significant bit eventually appears in A_4. At this point, A_4 and B_4 are both set, i.e., the most significant bits of divisor and dividend are aligned.

This alignment enables the AND gates. However, this has no effect on the Exclusive-OR gates with inputs derived from zero bit positions of register B. The "shift left" nature of register A at bit locations fed by these Exclusive-OR gates is preserved. Thus in Figure 5-21, the A_1 input comes from A_0, and A_3 input from A_2. On the other hand, the remaining bit positions receive the result of modulo 2 subtraction between appropriate bits. In summary, when register A is clocked after bit alignment, the partial remainder is loaded into it. If clocking is continued until all dividend bits are processed, the content of register A is the required remainder. Table 5-6 illustrates the register contents through this process; it is instructive to compare it with the long division.

Table 5-6
Bit Patterns Through Division Process

Input	Register				
	A_4	A_3	A_2	A_1	A_0
1	0	0	0	0	1
1	0	0	0	1	1
0	0	0	1	1	0
1	0	1	1	0	1
0	1	1	0	1	0
0	0	0	0	0	1
0	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	1
0	1	0	1	1	0
1	1	1	0	0	0
1	0	0	1	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	1	1	1
0	0	1	0	1	1

A closer examination of Figure 5-21 suggests that it can be greatly simplified. Figure 5-22 shows a functionally identical scheme similar to that used for cyclic checking purposes.

Discussion of basic polynomial division circuits cannot be concluded without further observations. The division algorithm can be implemented by suitable interconnection of shift registers and Exclusive-OR gates. The total number of register positions equals the degree of the divisor polynomial. The total number of Exclusive-OR gates is equal to one less than the number of non-zero terms in the divisor.

Polynomial Divider for Cyclic Checks

But for one drawback, the polynomial divider could be used as a cyclic check generator. Imagine that the dividend polynomial $x^{16}+x^{15}+x^{13}+x^9+x^8+x^6+x^5$ is the result of multiplying $(x^{11}+x^{10}+x^8+x^4+x^3+x+1)$ by x^5, and the divisor $x^5+x^4+x^2+1$ is the generating polynomial. From the cyclic check coding scheme, remember that $x^{11}+x^{10}+x^8+x^4+x^3+x+1$ is the actual data stream. The divider circuit discussed so far does not provide the remainder until the trailing zeros have been processed. Thus, if the remainder is to be appended as a check to the data stream, there is a delay before it is available for transmission. In almost all applications, such a gap between data and check bits is undesirable. This deficiency could easily be rectified if a circuit were possible which could multiply two polynomials while dividing by a third simultaneously.

Polynomial multiplication circuits can be derived using analogous arguments that result in the division circuit. For example, the arrangement shown in Figure 5-23 multiplies an incoming polynomial by $x^6+x^5+x^4+x^3+1$. Fortunately for cyclic check applications multiplication by a single term in the form x^r, where "r" is the degree of the generator polynomial, is sufficient. To implement a multiply by x^5 circuit, only a 5-bit shift register and one Exclusive-OR gate are needed as shown in Figure 5-24.

It is possible to combine the multiplier shown in Figure 5-24 and the divider in Figure 5-21 to implement a simultaneous multiply by x^5 and divide by $x^5+x^4+x^2+1$ circuit as shown in Figure 5-25. As before, Figure 5-25 may be simplified to arrive at Figure 5-26 which can be used as a cyclic check generator for the generating polynomial $P(x) = x^5+x^4+x^2+1$.

Table 5-7 lists the register content as each bit of the dividend (message polynomial) is processed.

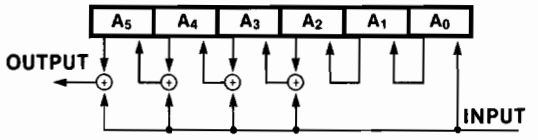


Figure 5-23 Circuit for Multiplying by $(x^6+x^5+x^4+x^3+1)$

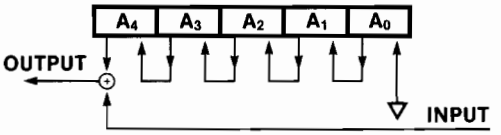


Figure 5-24 Circuit for Multiplying by x^5

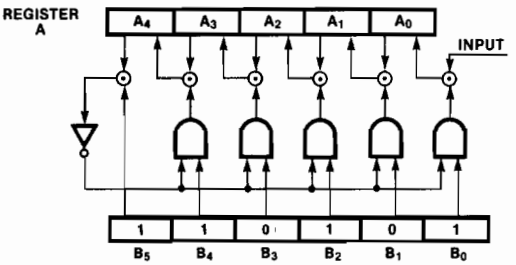


Figure 5-25 Conceptual Cyclic Check Generator

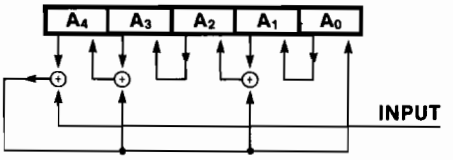


Figure 5-26 Basic Cyclic Check Circuit for P(x) = $x^5+x^4+x^2+1$

Table 5-7
Bit Pattern Through the Check Circuit

Input	Register				
	A_4	A_3	A_2	A_1	A_0
1	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	0
1	0	1	0	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	1	1	1
1	1	1	1	1	0
1	1	1	1	0	0
0	0	1	1	0	1
1	0	1	1	1	1
1	0	1	0	1	1

From Table 5-7, it is clear that the remainder is available as soon as the last data bit is processed. Also note that the quotient bit pattern appears in A_0. If it is desired to transmit the remainder from the register of Figure 5-26, it must be transmitted in a serial fashion. The connections must be established to make the register a straight shift from right to left by disabling the feedback through the Exclusive-OR gates.

Reverse Polynomials

Cyclic checks are often used in magnetic tape systems. Many of these have capabilities to read data in both forward and reverse directions. One of the reasons for this capability is to combat the overhead required to position the tape in front of the data block for a re-read operation in the event of an error. When "data followed by check bits" format is used to write on the tape, the check character is encountered first while reading in the opposite direction and the bit order for the whole block is reversed. Clearly, if the same check circuitry is used for error detection in both directions, erroneous indications are inevitable when reading in the opposite direction. This situation can be avoided by utilizing a reverse polynomial for checking in the opposite direction. The reverse polynomial is obtained by writing a polynomial bit pattern backwards. For example, the bit pattern for CRC-16 (forward) is 11000000000000101, i.e., $x^{16} + x^{15} + x^2 + 1$. The reverse polynomial for this pattern is 10100000000000011 or $x^{16} + x^{14} + x + 1$.

The 'F401 utilizes the concepts outlined above and contains polynomials for CRC-16, CRC-16 Reverse, CRC-12, LRC-8, CRC-CCITT and CRC-CCITT Reverse.

'F402 Expandable Polynomial Generator Checker

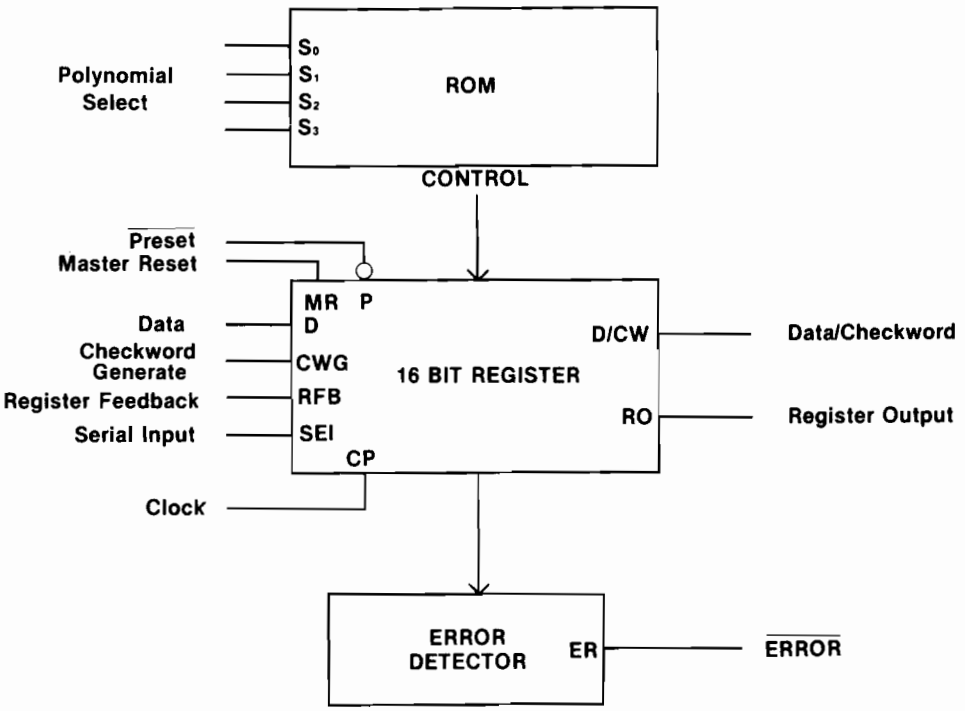


Figure 5-27

The 'F402 Expandable Polynomial Generator Checker is similar to the 'F401 and is expandable to polynomials of the 56th order. Six selectable polynomials and all data path gating is on the chip. The six polynomials contained on the chip are CRC16, CRC-CCITT, Ethernet, Ethernet Residue, 32nd Order, 48th Order and 56th Order. A bypass mode is also included in the selection to disconnect the chip from the feedback path. The 32nd, 48th and 56th order polynomials are all fire code generator polynomials; burst correction circuitry may be implemented with external circuitry.

'F402 for 16th Order Polynomials

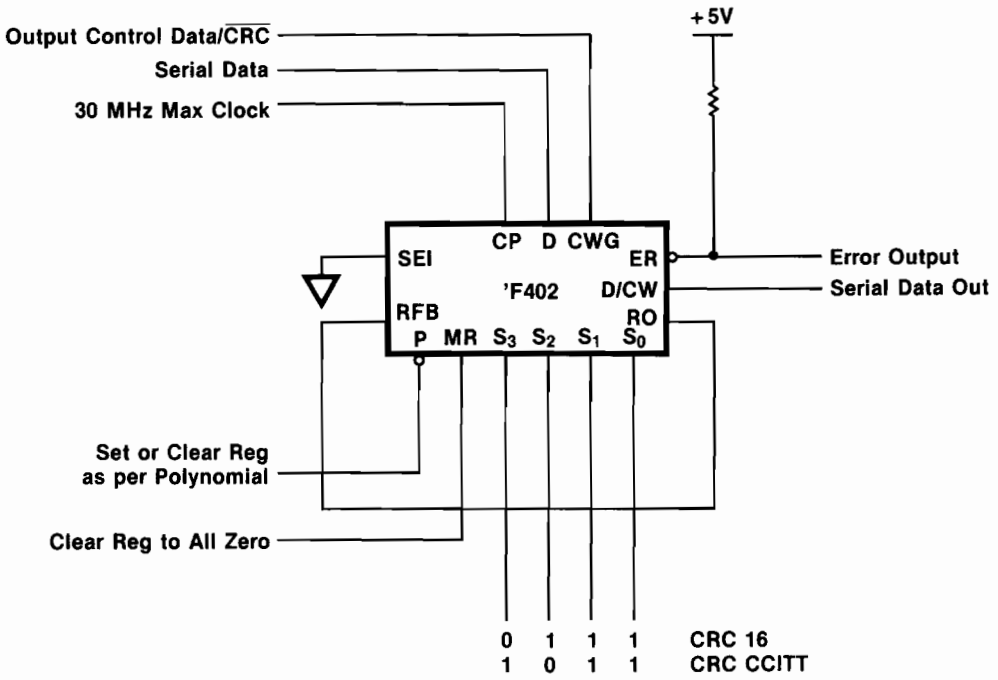


Figure 5-28

This shows the 'F402 in its simplest configuration, CRC16 or CRC-CCITT. The serial enable in line (SEI) is grounded, register out (RO) is connected to the register feedback (RFB), serial data in is applied to the data line (D), and transmitted data is taken from data/check word output (D/CW). The device can be clocked at 30MHz over temperature and voltage limits. Data or check word out is controlled by the check word generate line (CWG). The error flat ($\overline{ER}$) is an open collector output for expansion purposes.

'F402 for 32nd Order Polynomials

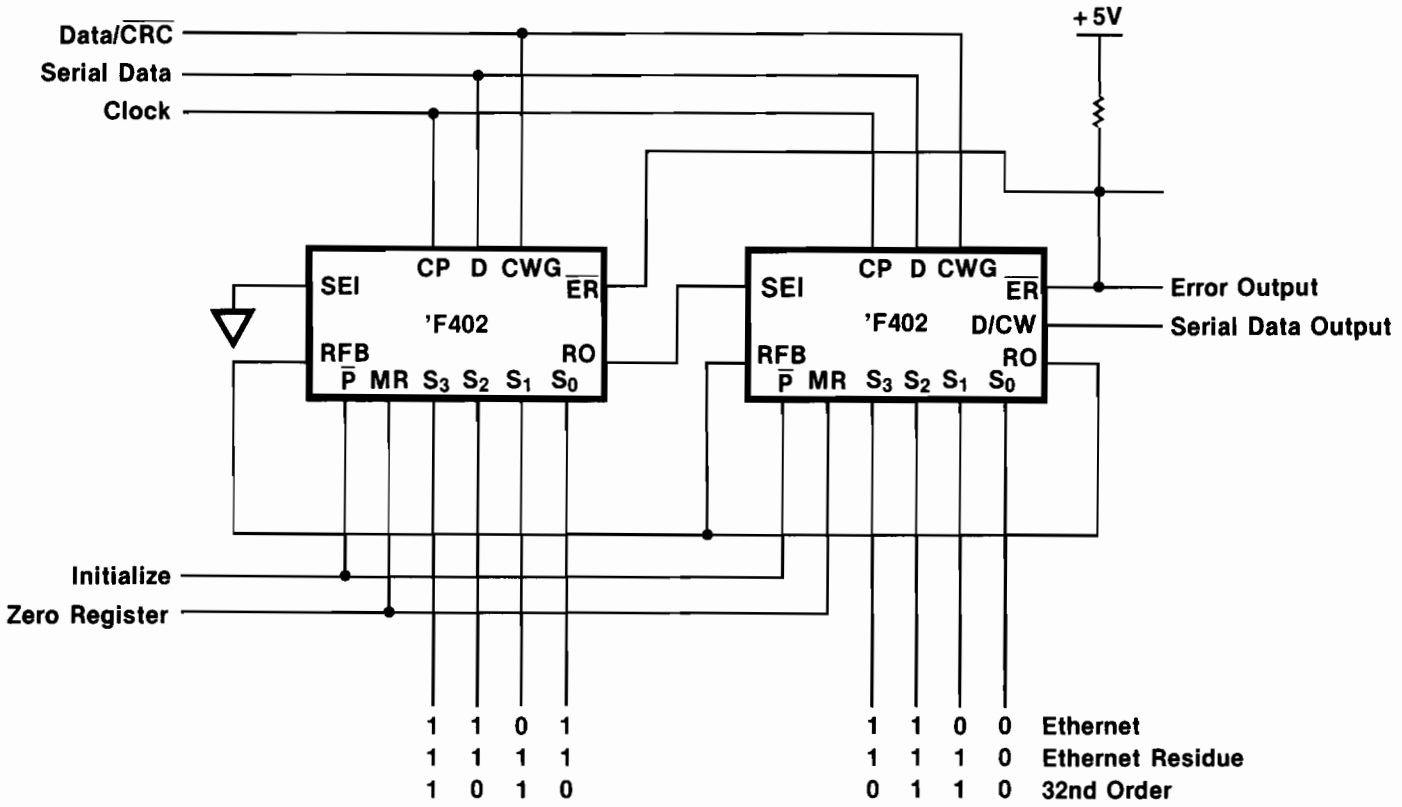


Figure 5-29

This diagram shows the ease of expansion of the 'F402. The register output (RO) of the most significant device is used to drive the register feedback (RFB) of both devices. The register output (RO) of the lower order device connects to the serial enable input (SEI) of the next higher order device; clock, data, check word generate and error flag lines of the respective devices are connected in parallel. A unique feature of the ethernet polynomial is that when the data stream and check word are divided by the polynomial, the result is not zero. To check for errors, the polynomial is changed to the ethernet residue polynomial and clocked once more. The error output will then go HIGH if no transmission error has occurred.

'F402 for 48th Order Polynomials

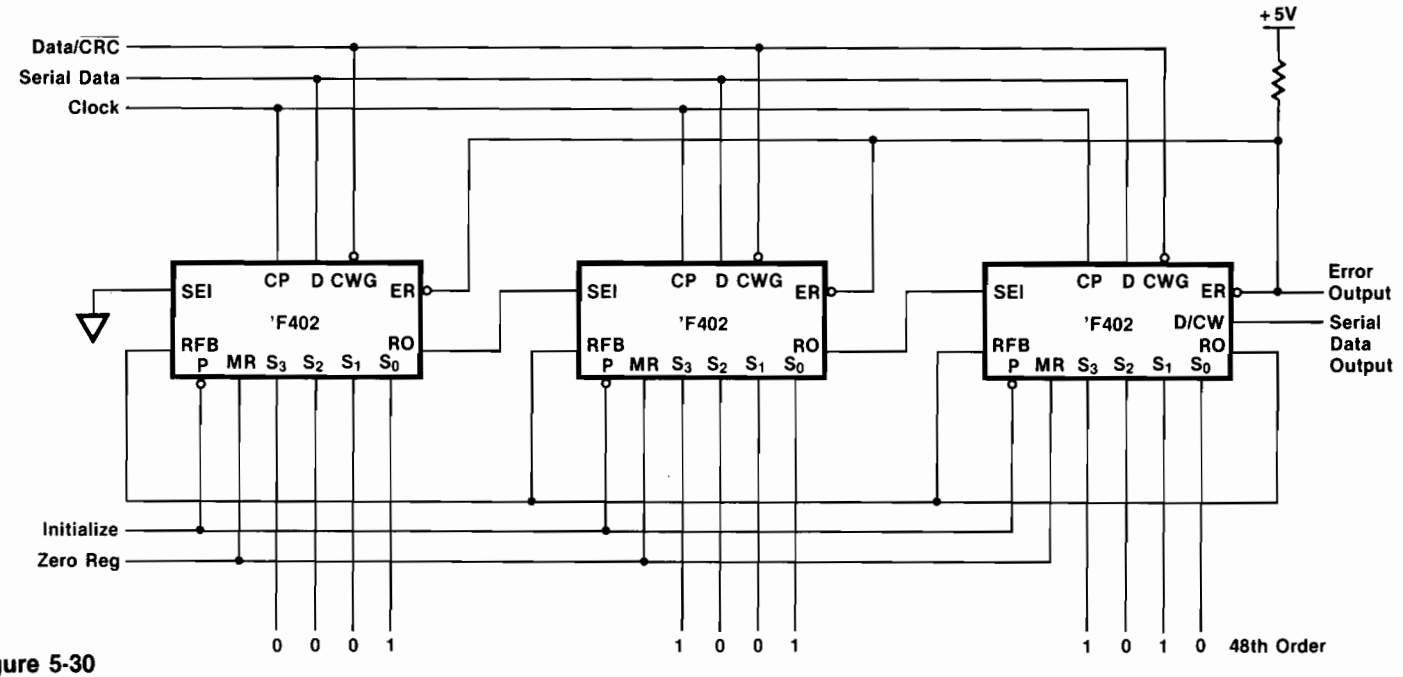


Figure 5-30

In this application another 'F402 is inserted into the feedback chain in a similar manner to the 32nd order configuration.

'F402 for 56th Order Polynomials and Lower

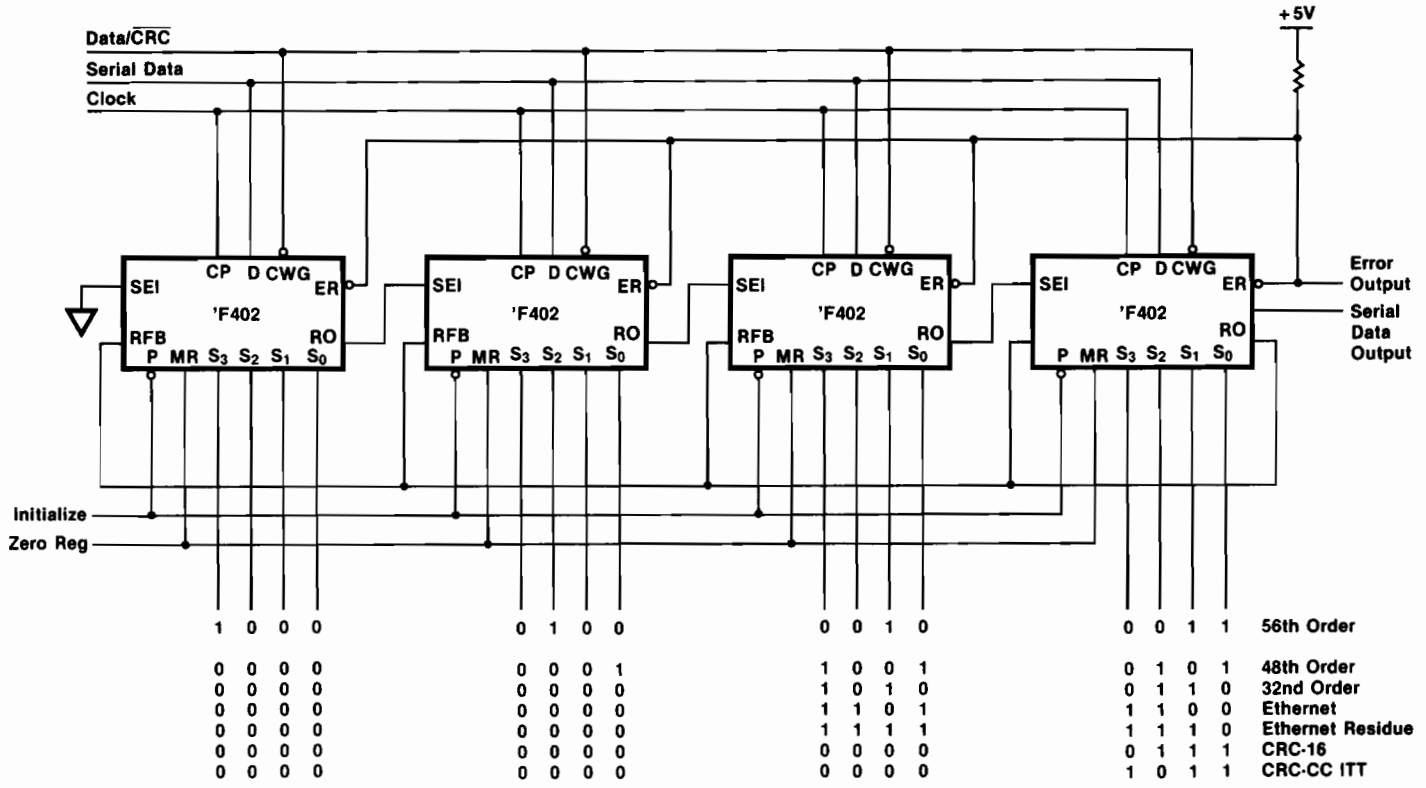


Figure 5-31

This shows a total of four 'F402s connected to form a 56th order polynomial generator checker. Note the 0000 select codes on the lower order registers when used with 48th, 32nd and 16th order polynomials.

Code Conversion

Numbers can be represented by a large variety of codes. The binary code is the most natural, the simplest, and the one most commonly used in high-speed computer systems. For convenience this code is often grouped in 3-bit groups and called an octal code. However, since this code is simply a different way of interpreting the binary code, all the features of the binary code are retained.

Unfortunately, a different numbering system based on the number 10 is in everyday use and mixed numbering systems are also used for some special applications (time, angles, etc.). This ambiguity has created a need for binary-to-BCD (binary coded decimal) and BCD-to-binary converter circuits.

The number of bits and digits involved, the time available and the amount of general purpose (perhaps even microprogrammed) logic available in the system are important factors in selecting one of the many different methods available for code conversion.

Any arbitrary code can be converted into any other arbitrary code by using a Read Only Memory (ROM) as a lookup table. This method is very fast with bipolar ROMs, but in most cases it is unnecessarily expensive, since most codes show some kind of regularity. Cost effective MSI circuits can take advantage of this regularity and provide a more economical solution.

Binary adders are used in high-speed parallel BCD-to-binary conversion. Every bit in a BCD number can be expressed as a binary number, and their sum is the binary equivalent of the whole BCD number. BCD adders such as the 'F583 are available for performing high-speed BCD arithmetic.

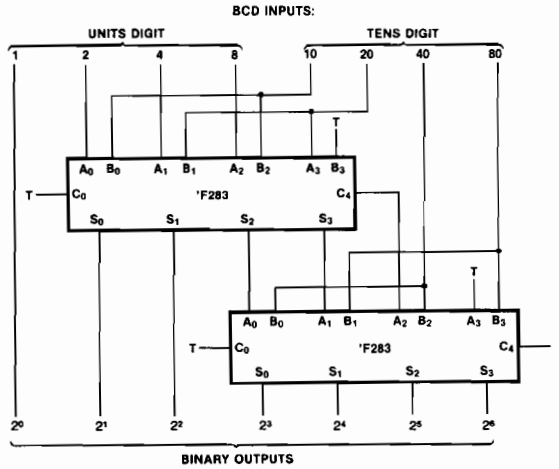


Figure 5-32 2-Digit BCD to 7-Bit Binary Converter

Converting a 2-digit BCD number into a 7-bit binary number is accomplished simply and economically with two 4-bit adders. The necessary interconnections are determined by first expressing each of the weighted BCD bits in terms of numbers that are powers of two.

$80 = 64 + 16 = 2^6 + 2^4$

$40 = 32 + 8 = 2^5 + 2^3$, etc.

Arranging the BCD and binary numbers in an orderly array, as shown in Table 5-8, makes it easy to see which of the BCD inputs must be summed into the various binary outputs. For example, the 2^0 output is the least significant bit of the unit's BCD digit, while inputs 2 and 10 must be summed to produce the 2^1 output. Notice that the 2^3 sum has more than two inputs (8, 10 and 40) and therefore cannot be formed in a single adder stage. Thus, for the 2^3 output, the sum is partially formed in the first adder package and completed in the second, as shown in the connection diagram. Inputs marked with a T must be terminated LOW for active HIGH inputs and terminated HIGH for active LOW inputs.

Table 5-8

Binary Outputs	BCD Inputs: 1	2	4	8	10	20	40	80
2^0	X							
2^1		X			X			
2^2			X			X		
2^3				X	X		X	
2^4						X		X
2^5							X	
2^6								X

4-Digit BCD to Binary Converter

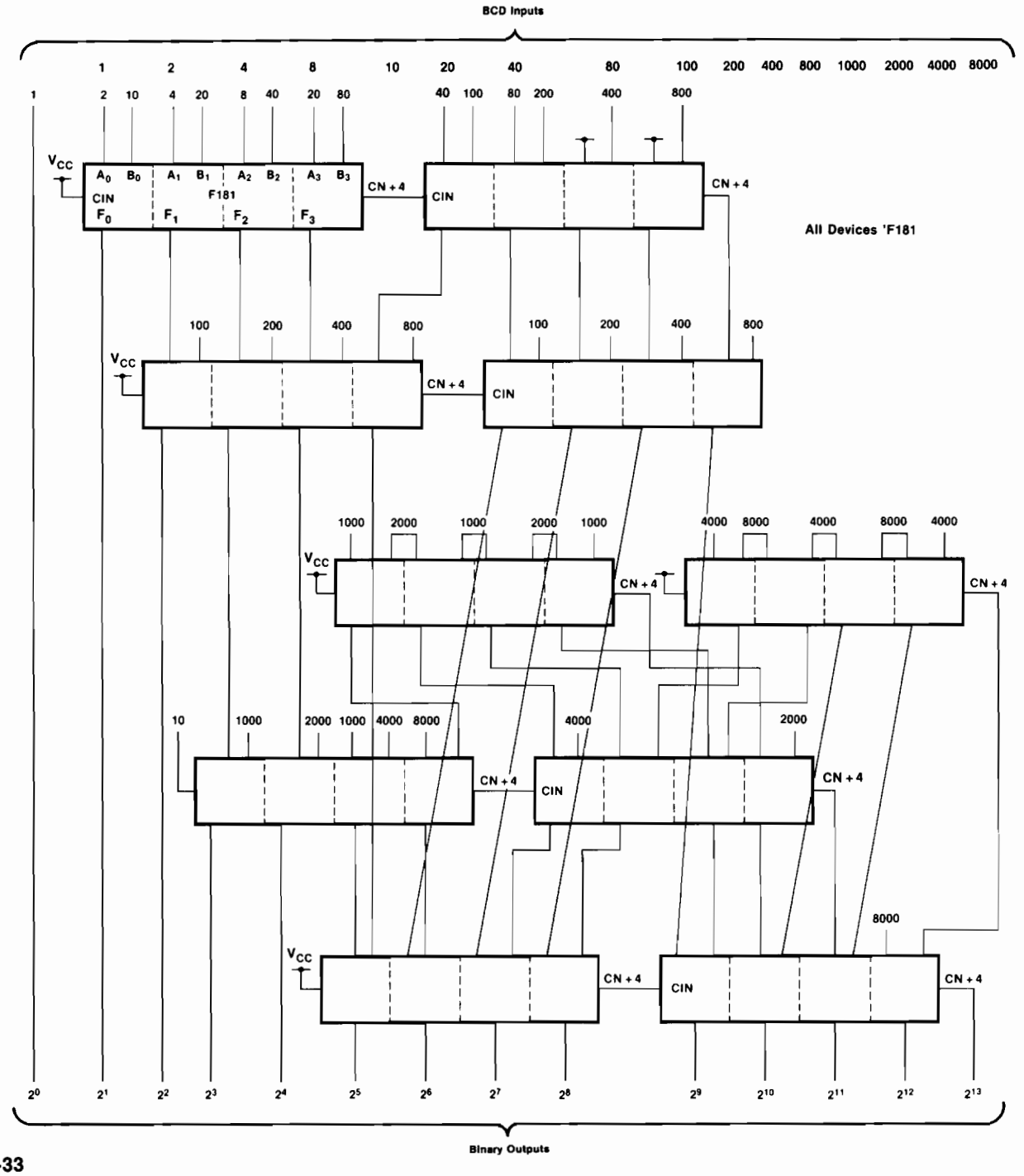


Figure 5-33

Figure 5-33 illustrates a scheme whereby the 'F181 ALU or 'F283 may be used to form a BCD to Binary converter. In this case four digits have been converted. However, the circuit could be readily expanded to a larger number of digits if desired. Since either active HIGH or active LOW operation could be selected, the mode input control, M, would be tied HIGH for active LOW operation and tied LOW for active HIGH operation. Note that one of the Carry Ins is not connected to either the HIGH or LOW bus, but rather to the BCD 10. One of the Carry Outs can be ignored since it cannot be active for any legitimate input condition.

Bit Serial Binary-to-BCD Converter

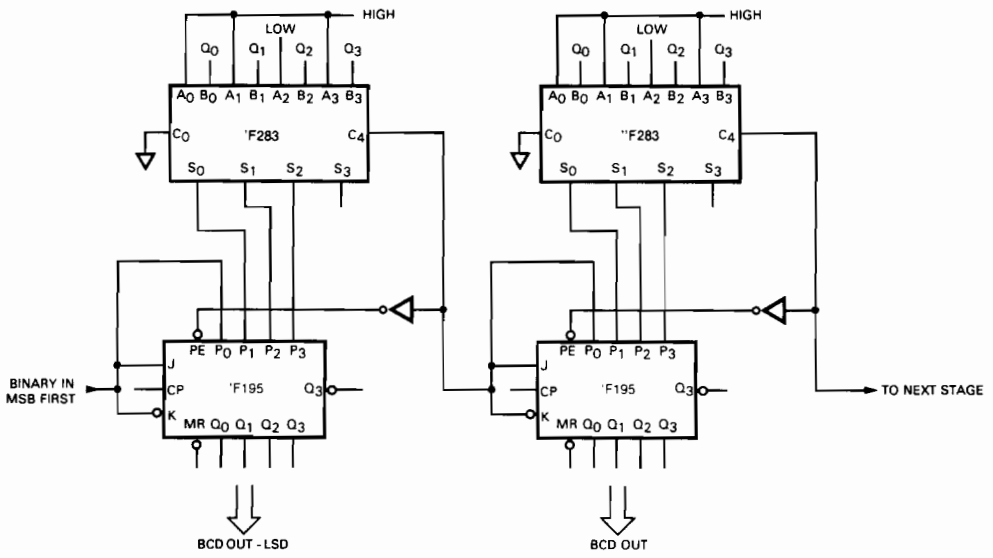


Figure 5-34

The reverse of the BCD-to-binary algorithm is used for binary-to-BCD conversion. The binary word is shifted, most significant bit first, into a shift register consisting of several series-connected 'F195. Each shift doubles the contents of the registers in terms of BCD notation. Therefore, a correction is required whenever any of the 4-bit registers contains a number greater than four, which when shifted generates a non-BCD code. This correction is performed by adding three to the contents of the register and inserting the sum one bit downstream into the parallel data inputs. By adding 11 and then ignoring the most significant bit, the same 4-bit adder also detects whether or not the correction is necessary. A binary number is completely converted when its LSB has been shifted in. This circuit can be used for any number of bits and digits. It requires only one 'F195 4-bit shift register, one 'F283 4-bit adder, and one inverter for each resulting BCD digit.

Serial-In, Serial-Out BCD-to-Binary Converter

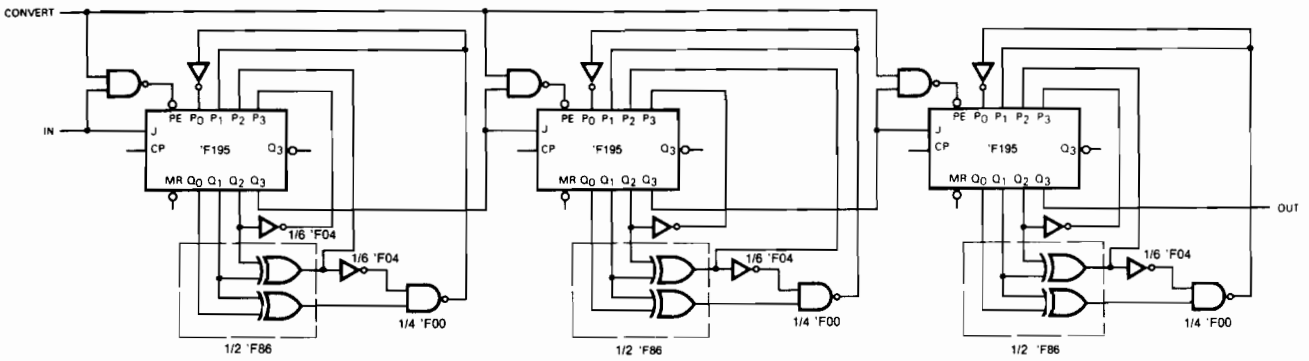


Figure 5-35

A well-known algorithm generates the binary equivalent of a BCD number by repeatedly dividing it by two. The series of least significant bits generated is the binary output, least significant first. This algorithm can be implemented with 'F195 shift registers and some gates or adders.

When a BCD number is stored in the 'F195 shift register with its LSB in the Q_3 stage, a right shift effectively divides it by two. A problem arises if the LSB of the more significant digit is a one, implying a value of 10 with respect to the first digit. Shifting this one into the Q_0 position changes the 10 to an 8, instead of dividing it by 2. To correct for this, a 3 must be subtracted from the new contents of the 'F195 register. The circuit shown provides a gate-minimized implementation of this algorithm using the parallel inputs of the 'F195 for the correction. It converts a 4-digit (9999) BCD number into its 14-bit binary equivalent. Operation is started by bit-serially shifting in the three least significant BCD digits (LSB of the LSD first) while the Convert input is LOW. The actual conversion starts when the three digits have been shifted in and the LSB of the most significant digit is being applied to the serial input. At this point, the Convert input is made HIGH, activating the three correction networks whenever there is a one to be shifted into any of the registers. The next fourteen clock pulses shift out the binary result, LSB first. This circuit can be used for any number of digits. It requires only one 4-bit shift register with a conversion network for each decimal digit except the MSD.

5

BCD-to-Binary Converter Using Counters

Counters can be used for binary-to-BCD or for BCD-to-binary conversion, if sufficient time is available. The code to be converted is loaded into a counter, which is then counted down while another counter is counted up. When the first counter reaches zero the second counter has reached the original numeric value represented in the desired code. This method is easily expanded and can also be used for mixed modulo codes (like 6/10 for minutes and seconds, 36/10 for degrees of angle). It is also very easy to perform accumulation.

Figure 5-36 is a BCD-to-binary converter capable of taking a 5-digit BCD number and converting it into a 17-bit binary number in less than 5ms using a clock generator running at 40MHz. The circuit, as shown, is completely self-contained, including a debounce/edge detect circuit for use with a push button to initiate the conversion. Only eleven integrated circuit packages are required.

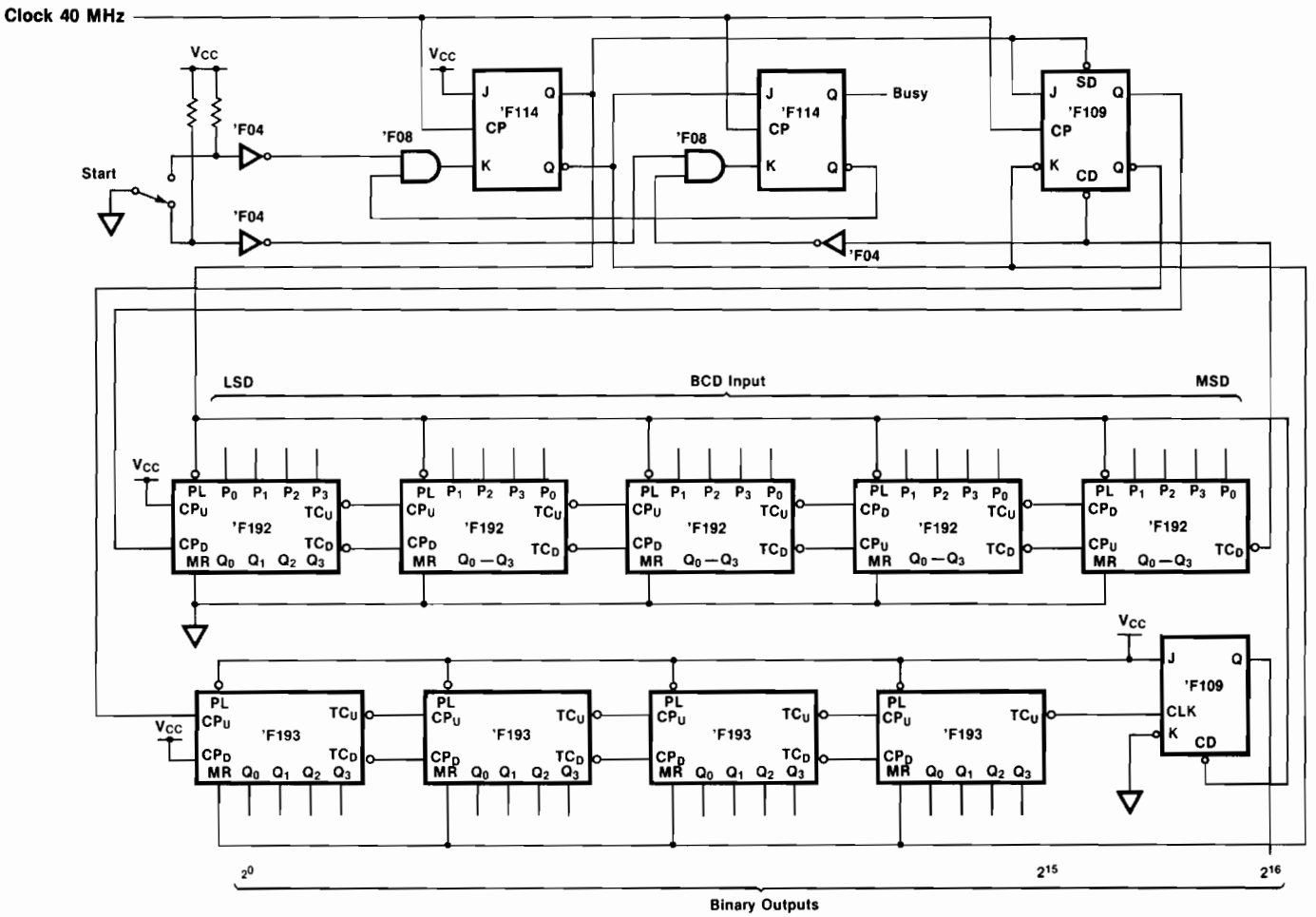


Figure 5-36

A 'F114 dual flip-flop is used to debounce and detect a new closure on the start button and to generate a pulse one clock period wide which loads the BCD number into a decimal down counter and clears a binary up counter. The two 'F114 flip-flops are normally in an idle state with the first flip-flop set (Q = HIGH) and the second reset (Q = LOW). Depressing the start button causes the first flip-flop to reset, activating the asynchronous Load and Reset inputs on the respective counters and asynchronously setting a 'F109 flip-flop used to drive the clock inputs on both counters. While a clear direct signal initially exists at this flip-flop, it quickly disappears as the clock input to the down counter is forced HIGH and the counter is loaded with a new number.

The next clock pulse causes both 'F114 flip-flops to set, removing the counter load and reset signals, generating a "busy" signal, and enabling the 'F109 flip-flop to toggle on subsequent clock pulses. The two counters count, one down and the other up, at half the clock generator frequency until the decimal down counter reaches zero.

After the down counter has reached zero and the 'F109 flip-flop has again reset, the $\overline{TC}_D$ output from the counter goes LOW holding the flip-flop in the reset state and thus locking the counters. $\overline{TC}_D$ = LOW also enables the second 'F114 to reset if the start button has been released. The push button must be released and depressed again to initiate a new conversion.

The converter provides the correct binary output for any number within the range of the counter including zero.

Binary Angle to BCD Converter With Display

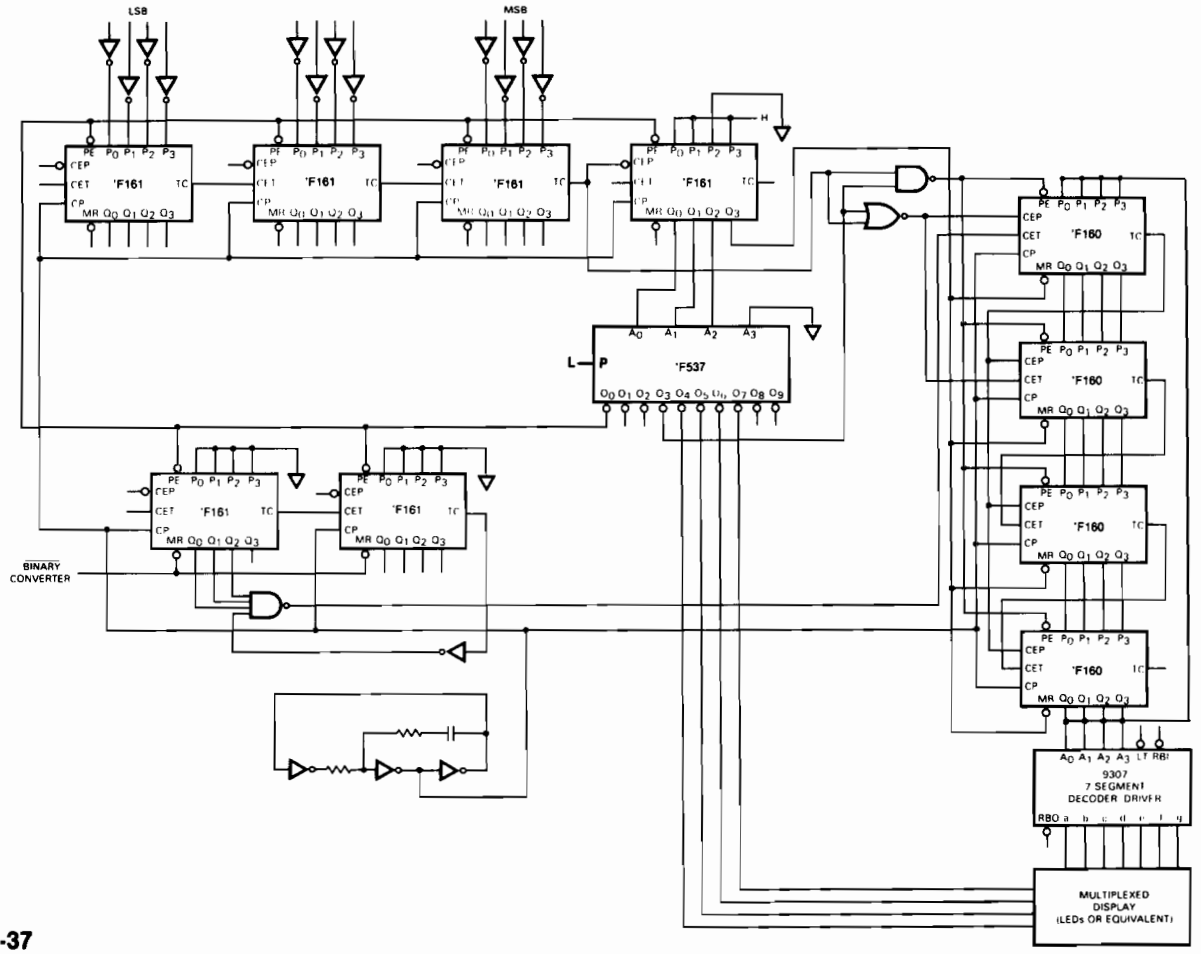


Figure 5-37

This converter can perform a code conversion and display the result continuously in binary coded decimal (BCD). The converter, as shown, operates in either of two modes, binary-to-BCD or binary angle-to-BCD. In the binary angle code, the most significant bit represents 180° (half circle), the next 90° (quarter circle), the third 45° (eighth circle), etc. The converter accepts twelve bits of this code and converts it to degrees and tenths of degrees (0.0° to 359.9°). The converter is a self-contained circuit requiring only seventeen packages capable of running at a 30MHz clock rate. The BCD output is displayed in an economical manner by multiplexing the four digits.

The circuit operates by automatically loading the complement of the input code into the binary counter at the beginning of the cycle while the decimal counter is cleared. The binary counter reaches terminal count after n clock pulses where n = binary input number. In the binary conversion mode, the decade counter also counts n times and thus reaches the BCD equivalent of the binary input. If the ability to convert angles to BCD is not needed, the two lower 'F161s and two gates are not needed.

After the conversion, the BCD data is displayed by multiplexing the digits, most significant digit first. The four stage binary counter shifts the digits through the decimal counter while enabling one display digit at a time by counting 4096 clock pulses per digit. After displaying the least significant digit, the cycle is repeated.

In the binary angle-to-BCD mode, the decimal counter is incremented at a slower rate than the binary counter to adjust for the weights of the binary angle bits entered in the binary counter. The most significant bit in a binary-to-BCD conversion has a weight of 2^{11} or 2048, but in this binary angle-to-BCD conversion it has a weight of 180° or 1800 tenths of a degree. The BCD counter is therefore incremented at 1800/2048, or 225/256 the rate of the binary counter by inhibiting the decimal counter 31 times during every 256 clock pulses. This rate multiplier function is performed by two 'F161s (modulo 256 counter) and two gates decoding every eighth state, except TC of the counter, for a total of 31 states. These evenly distributed inhibit pulses minimize the conversion error.

Gray Code Conversions

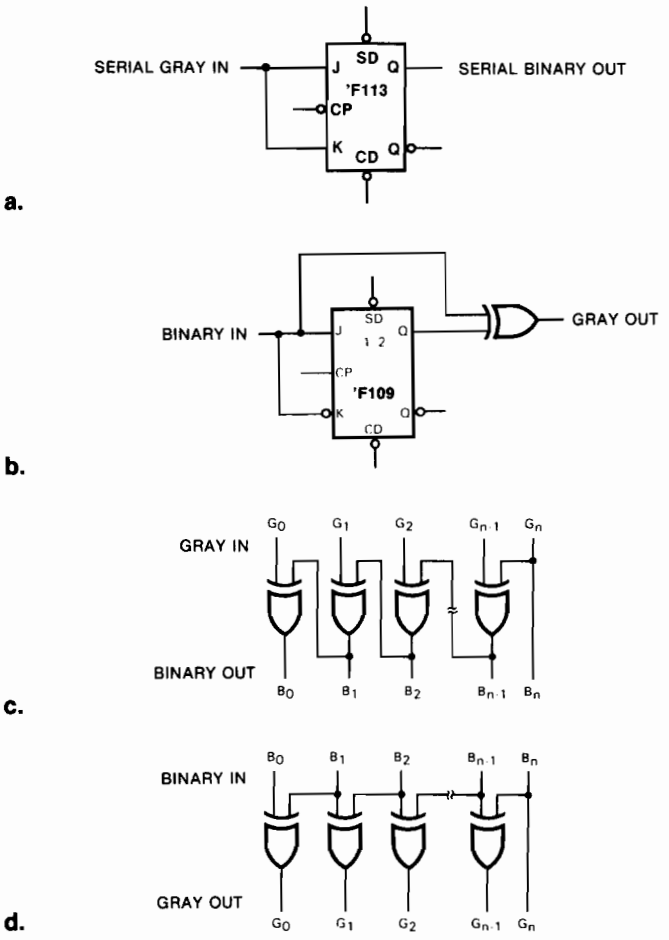


Figure 5-38

Binary codes are not particularly suited for electrical or electro-optical encoder systems (angular position shift encoders, etc.) because a movement from one state to the next often results in more than one bit change; i.e., from seven to eight, the binary code changes from 0111 to 1000. Such bit changes can never really be simultaneous, so the encoder always generates erroneous transient codes when switching between certain positions. This problem is avoided with a Gray code because only one bit changes between adjacent states. The Gray code is a non-weighted code and awkward for other applications. It must be converted to binary or BCD before any arithmetic can be performed.

In Gray-to-binary serial conversion, a flip-flop that toggles for every logic one performs the conversion. The most significant bit, however, must come in first. Gray-to-binary parallel conversion is performed by a series of Exclusive-OR gates.

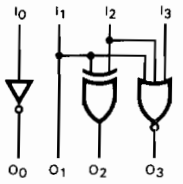
In binary-to-Gray serial conversion, a flip-flop acts as a 1-bit delay element and an Exclusive-OR gate is used between the present and the previous binary bit. Note that in this case, as well as in Gray-to-binary serial conversion, the most significant bit must come in first. Binary-to-Gray parallel conversion is performed by a series of Exclusive-OR gates.

Table 5-9 Excess 3 Gray Code

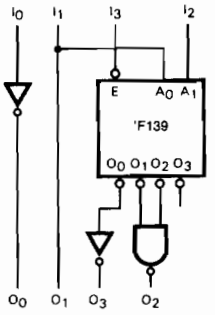
Decimal	Binary	Gray	X3 Binary	X3 Gray
0	0000	0000	0011	0010
1	0001	0001	0100	0110
2	0010	0011	0101	0111
3	0011	0010	0110	0101
4	0100	0110	0111	0100
5	0101	0111	1000	1100
6	0110	0101	1001	1101
7	0111	0100	1010	1111
8	1000	1100	1011	1110
9	1001	1101	1100	1010
10	1010	1111		
11	1011	1110		
12	1100	1010		
13	1101	1011		
14	1110	1001		
15	1111	1000		

Decimal systems use Excess 3 Gray Code because this code has the feature of changing only one bit at a time even on a nine-to-zero transition. Excess 3 Gray Code is detected or generated in the same manner as Gray codes, except adding a three to the binary value for binary-to-Excess 3 conversion and subtracting a three (i.e., adding binary 13) from the binary value for Excess 3-to-binary conversion.

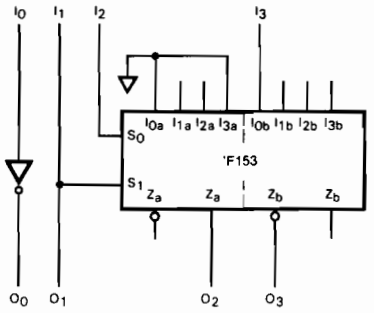
Generating Nines Complements



a.



b.



c.

Figure 5-39

The one complement of a binary number is easily generated by inverting each bit. The equivalent in a decimal (BCD) system, nines complement, is not that easy. These three circuits convert a 1-digit BCD input into its nines complement. They use about one equivalent gate or MSI package per digit (decade).

Controlled Nines Complement Circuit Using Two Gate Packages

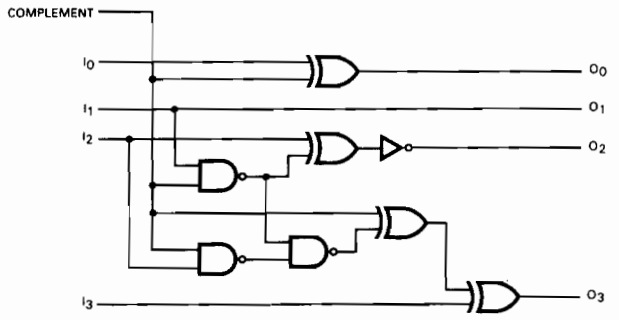


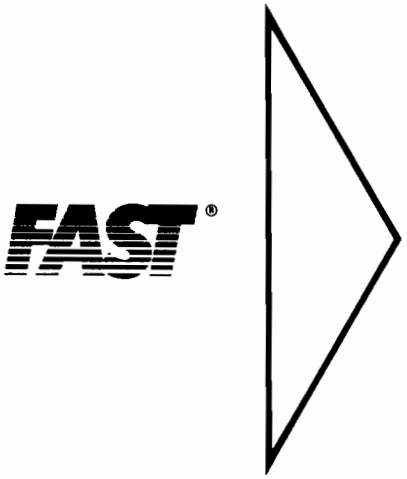
Figure 5-40

Table 5-10 Truth Table

I_0	I_1	I_2	I_3	Compl.	O_0	O_1	O_2	O_3
X	X	X	X	L	I_0	I_1	I_2	I_3
L	L	L	L	H	H	L	L	H
H	L	L	L	H	L	L	L	H
L	H	L	L	H	H	H	H	L
H	H	L	L	H	L	H	H	L
L	L	H	L	H	H	L	H	L
H	L	H	L	H	L	L	H	L
L	H	H	L	H	H	H	L	L
H	H	H	L	H	L	H	L	L
L	L	L	H	H	H	L	L	L
H	L	L	H	H	L	L	L	L

H = HIGH Voltage level
L = LOW Voltage level
X = Immaterial

This controlled nines complement circuit (Figure 5-40), using two gate packages, either generates the nines complement or transfers the BCD inputs through unchanged.



FAIRCHILD
A Schlumberger Company

FIFOs

Introduction

First-in, first-out memory devices are high performance "rubber band" memories that are used to buffer and synchronize information between two asynchronous parts of a system. They can also hold data or commands on a first come, first served basis. Some FIFOs have serial and parallel I/Os, enabling data to be serialized or deserialized as may be required by disk and tape controllers, local area network systems or between systems. Contained in this section is a FIFO controller that allows the use of static RAM as the storage medium, reducing the complexity from 25 packages to four.

6

Using the 'F403 and 'F433 FIFOs

The First-In First-Out (FIFO) memory is a read/write memory which automatically stacks the words in the same order as they were entered and makes them available at the output in the same sequence. In the past, MOS technology has been the dominant manufacturing process for FIFOs. Now, however, there are two new products that utilize advanced Schottky TTL technology, the 'F403 and the 'F433.

Description

The 'F403 ('F433) FIFO is a 16 (64) x 4 parallel/serial memory consisting of the following (Figure 7-1):

- An input register with parallel and serial data inputs as well as control inputs and outputs for input handshaking and expansion;
- A 4-bit wide, 14 (62) word deep fall-through stack with self-contained control logic;
- An output register with parallel and serial data outputs, control inputs for output handshaking and expansion.

Parallel data is entered into the input register by using D_0 through D_3 as data inputs and Parallel Load (PL) as the strobe. A HIGH at the PL input operates the direct set and clear inputs of the input-register flip-flops. The quiescent state of the PL input is LOW.

To enter data serially, the D_S is used as the data input and $\overline{CPSI}$ as the clock. The input register responds to the HIGH-to-LOW clock transition and the quiescent state of the $\overline{CPSI}$ input is LOW. For the $\overline{CPSI}$ to effect shifting, the Input Expand Serial ($\overline{IES}$) input must be LOW.

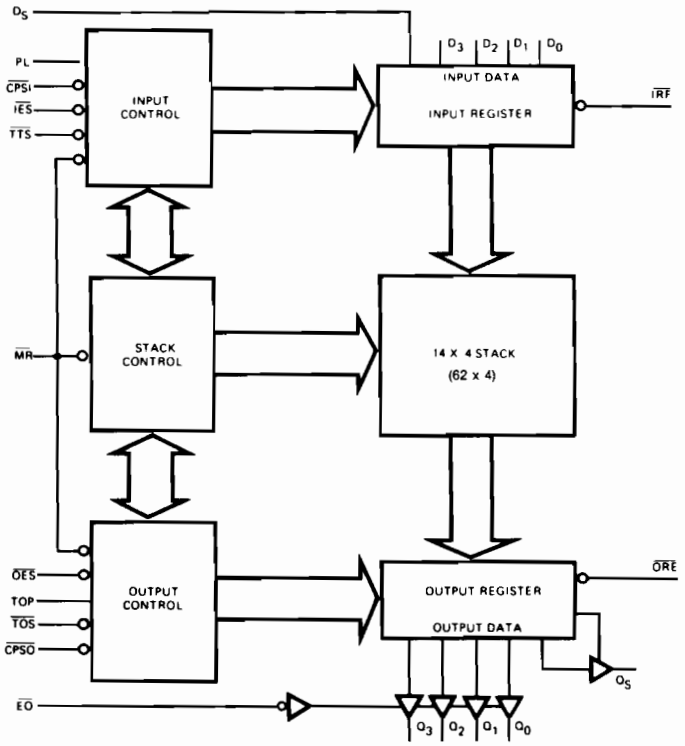
Whenever the input register receives four data bits, whether by serial or parallel entry, the status output signal, Input Register Full ($\overline{IRF}$), goes LOW. If the Transfer to Stack ($\overline{TTS}$) input is activated with a LOW pulse, data from the input register is transferred into the first stack location (provided it is empty). As soon as data is transferred, the control logic attempts to initialize the input register so that it can accept another word; however, the initialization is inhibited until the PL input is LOW. The device is designed so that the $\overline{IRF}$ output can be connected to the $\overline{TTS}$ input. Thus, when a data word is received by the input register, it automatically enters the stack and falls through toward the output, pausing only as needed for an "empty" location.

Normally, the Output Register Empty ($\overline{ORE}$) is LOW, indicating that the output register does not contain valid data. As soon as a data word arrives in the register, the $\overline{ORE}$ output goes HIGH, indicating the presence of valid data. If the Output Enable ($\overline{EO}$) input is LOW, the 3-state buffers are enabled and data is available on the Q_0 through Q_3 outputs.

Data can be extracted either serially or in parallel. Q_s is used for serial data output and $\overline{CPSO}$ for the clock input. The Q_s output is also available through a 3-state buffer and its enabling is controlled internally. Output register shifting occurs on the HIGH-to-LOW transition of the $\overline{CPSO}$ whose quiescent state is LOW. As soon as the last data bit is shifted out, the $\overline{ORE}$ output goes LOW, indicating that the output register is empty.

The inactive state of the $\overline{TOS}$ input is LOW. A HIGH-to-LOW transition on the input causes new data to be loaded from the stack into the output register (provided data is available). The $\overline{ORE}$ output can be connected to the $\overline{TOS}$ input so that as soon as the last bit is shifted out, new data is automatically demanded.

Figure 6-1 'F403 Block Diagram

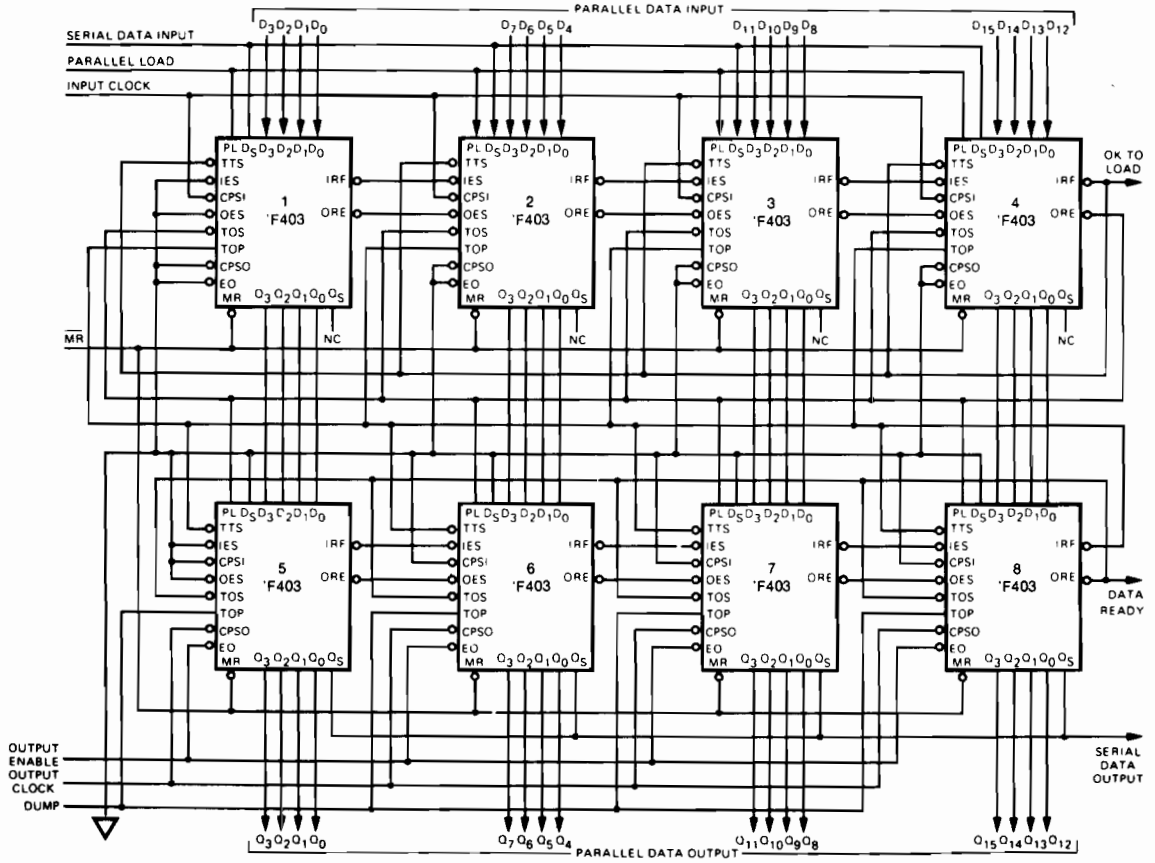


Pin Names

D_0-D_3	Parallel Data Inputs
D_S	Serial Data Input
PL	Parallel Load Input
$\overline{CPSI}$	Serial Input Clock Input (HIGH-to-LOW Triggered)
$\overline{CPSO}$	Serial Output Clock Input (HIGH-to-LOW Triggered)
$\overline{IES}$	Serial Input Enable (Active LOW)
$\overline{TTS}$	Transfer to Stack Input (Active LOW)
$\overline{TOS}$	Transfer Out Serial Input (Active LOW)
TOP	Transfer Out Parallel Input
$\overline{OES}$	Serial Output Enable Input (Active LOW)
$\overline{EO}$	Output Enable Input (Active LOW)
$\overline{MR}$	Master Reset Input (Active LOW)
$\overline{IRF}$	Input Register Full Output (Active LOW)
$\overline{ORE}$	Output Register Empty Output (Active LOW)
Q_0-Q_3	Parallel Data Outputs
Q_S	Serial Data Output

6

Figure 6-2 31 x 16 FIFO Array



The inactive state of the TOP input is HIGH and a LOW-to-HIGH transition causes new data to be loaded into the output register. Moreover, a HIGH level on the TOP input causes the $\overline{ORE}$ to go LOW. The TOP input can be connected to the $\overline{EO}$ input so that the output data can be enabled when $\overline{EO}$ is LOW. When the output is disabled, new data is automatically demanded. It should be noted that the $\overline{TOS}$ input does not affect the $\overline{ORE}$ output.

The FIFO is initialized by a LOW signal on the Master Reset ($\overline{MR}$). This causes the status outputs, $\overline{IRF}$ and $\overline{ORE}$, to assume an empty state; i.e., $\overline{IRF}$ is then HIGH and $\overline{ORE}$ LOW. It is important to remember that the $\overline{MR}$ does not clear all the data flip-flops; it only initializes the control. Specifically, the Q_0 . . . Q_3 outputs are not affected by the Master Reset.

Expansion

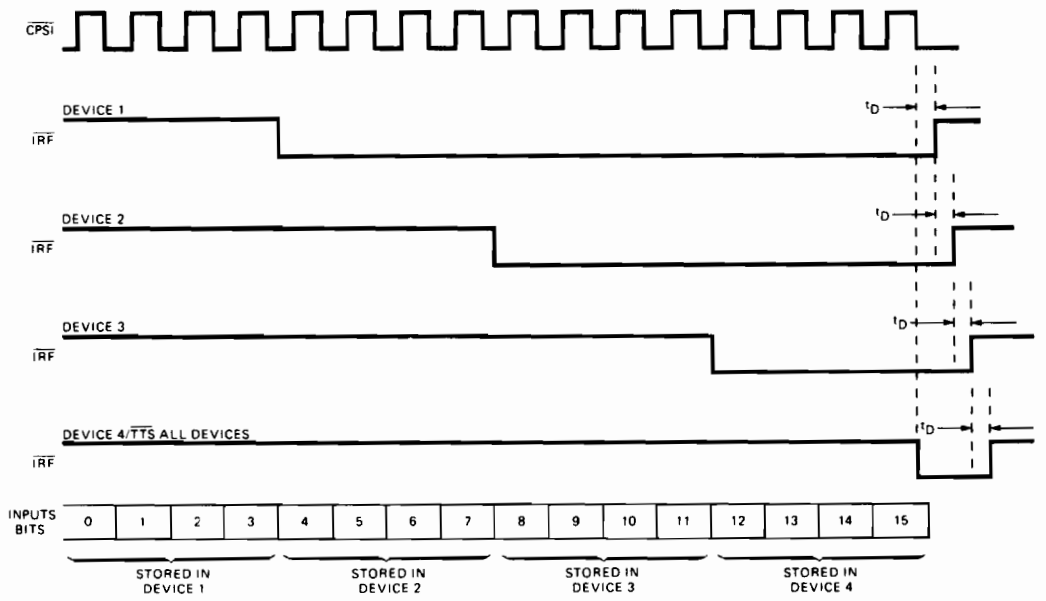
The 'F403/'F433 can be vertically expanded to store more words or horizontally expanded to store longer words (in multiples of four bits) without external logic. Also, the expansion scheme fully preserves the parallel/serial data features. To illustrate the expansion connections, a FIFO array consisting of eight devices is shown in Figure 6-2. If there are m devices in a row and n rows, the array provides (15n + 1) words of storage with 4m bits in each word. The reduction in storage to (15n + 1) words instead of 16n is quite common in such expansion (see explanation following). Data is entered into devices 1 through 4 and extracted from devices 5 through 8.

The D_S inputs of the first four devices are bussed together and serial data is entered on this line. The $\overline{CPSI}$ inputs are also connected together for clocking the serial data. The $\overline{IES}$ input of device 1 is connected to ground, while the $\overline{IES}$ inputs of devices 2, 3, and 4 are each connected to the $\overline{IRF}$ output of the preceding device. The $\overline{IRF}$ output of device 4 feeds into the $\overline{TTS}$ inputs of all four devices.

After initialization by a LOW level on the $\overline{MR}$ input, the $\overline{IRF}$ output of all four devices are HIGH. Under these conditions, only device 1 responds to the $\overline{CPSI}$ because its $\overline{IES}$ input is LOW. The first four clock pulses shift four data bits into the device-1 input register; its $\overline{IRF}$ output then becomes LOW. The first data bit is located in a flip-flop corresponding to the D_0 input of device 1. Control logic inhibits the $\overline{CPSI}$ from further affecting this device.

Because the $\overline{IES}$ input of device 2 is now LOW, the clock starts shifting data into the input register of device 2. On the eighth clock pulse, the $\overline{IRF}$ output of device 2 goes LOW and disables shifting of device 2. This process continues on devices 3 and 4. Therefore, on the 16th clock pulse, the $\overline{IRF}$ output of device 4

Figure 6-3 Serial Data Entry for FIFO Array



becomes LOW and activates the $\overline{TTS}$ inputs of all devices. The stack control logic in each device responds by transferring data into each stack from the respective input register, and the input registers are initialized. Thus the $\overline{IRF}$ outputs of all devices become HIGH once again. An automatic priority scheme assures that if the $\overline{IRF}$ output of device 4 is HIGH, the input registers of all four devices have been initialized. The timing diagram for 16 bits of serial entry into the array is shown in Figure 6-3.

Parallel entry into the array is made with a HIGH level on the PL inputs. The same conditions prevail in the input section that exist after the 16th clock pulse in the serial entry mode. The stack controls do not initialize the input registers until the PL inputs are LOW to assure proper device operation.

Data loaded into the stacks eventually arrives at the output registers of the first four devices. Normally, the $\overline{ORE}$ outputs are LOW due to initialization; however, as soon as data is loaded into each output register, the $\overline{ORE}$ goes HIGH. An automatic priority scheme, similar to the one for data entry, also exists at the output. Thus a HIGH level on the $\overline{ORE}$ output of device 4 guarantees that valid data is present in all the output registers.

The $\overline{ORE}$ output of device 4 is connected to the PL inputs of devices 5 through 8, as well as to the $\overline{TOS}$ inputs of the first four devices. It should be noted that if serial extraction from the output is not desired, the $\overline{TOS}$ inputs can be connected to ground instead. The $\overline{EO}$ inputs of the first four devices are connected to ground; thus the contents of an output register are available on the appropriate outputs.

The HIGH level on the $\overline{ORE}$ output of device 4 activates the PL inputs of devices 5...8, thus forcing the data outputs from each device in the first row into the input register of the corresponding device in the second row. The $\overline{IRF}$ output of device 8 is connected to the TOP inputs of devices 1...4 and to the $\overline{TTS}$ inputs of devices 5...8. Because the PL inputs are HIGH, the $\overline{IRF}$ outputs of devices 5...8 are LOW, therefore establishing a LOW on the TOP inputs of devices 5...8, thus initiating a fall-through action. The stack controls in devices 5...8 initialize their respective registers and the $\overline{IRF}$ outputs go HIGH. An automatic priority scheme is also present at the inputs of devices 5...8. The HIGH on the $\overline{IRF}$ output of device 8 restores the TOP inputs of devices 1...4 to the quiescent state.

If the stacks of devices 5...8 are full, activating the $\overline{TTS}$ inputs by the LOW $\overline{IRF}$ output of device 8 would not initiate a data transfer from the input registers. The $\overline{IRF}$ output of device 8 would remain LOW until the data can be successfully transferred into the stacks. Thus, as long as devices 5...8 are holding 16 words, the $\overline{IRF}$ output of device 8 remains LOW. This also holds the TOP inputs of devices 1...4 LOW. As long as they remain LOW, data cannot be loaded into the output registers from the stacks because a LOW-to-HIGH transition at the TOP inputs is needed to demand new data. Under these circumstances, devices 1...4 temporarily lose the ability to use their output registers and hence can hold only 15 words. As a result, the two rows have a storage capacity of 31 words instead of 32; and, for the general case, the storage capacity of an n-row array is (15n + 1) instead of 16n.

The data loaded into the stacks eventually arrives at the output registers of devices 5...8, at which time the $\overline{ORE}$ outputs go HIGH from the LOW state originally initialized by the $\overline{MR}$ input. The automatic priority scheme is still in effect, and the data from the output can be extracted either in serial or parallel format.

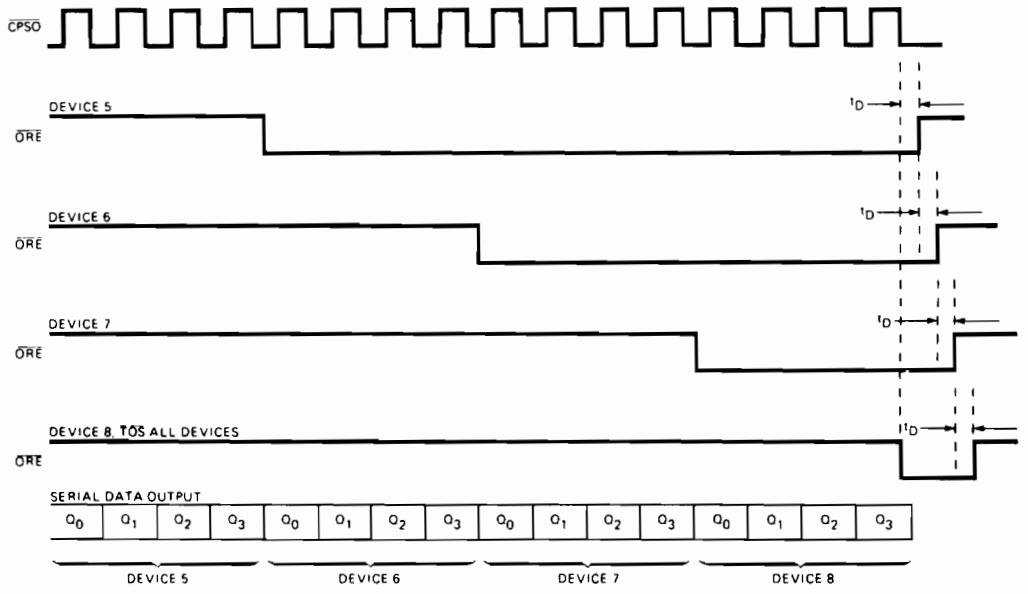
The Q_S outputs of devices 5...8, each available through a 3-state buffer, are connected together and the serial data output from the array appears on this line. The $\overline{CPSO}$ inputs are also connected together and the line driven by the output clock. When there is no valid data in the output register, Q_S is disabled and is therefore in a high impedance state.

The OES input of device 5 is connected to ground and devices 6, 7 and 8 each receives its OES input from the preceding device. As soon as data arrives in the output registers of devices 5...8, the $\overline{ORE}$ outputs go HIGH and the 3-state buffer of device 5 is enabled so that its Q_S output becomes identical to its Q_0 output. The Q_S outputs of devices 5...8 are in a high impedance state. The clock on the $\overline{CPSO}$ input shifts the device-5 output register and data is shifted out in the same bit order as entered at the array input. After the fourth clock pulse, the $\overline{ORE}$ output of device 5 goes LOW and its Q_S output is disabled into the high impedance state.

The $\overline{ORE}$ output of device 5 establishes a LOW on the $\overline{OES}$ input of device 6. This enables its Q_S output buffer and a signal, corresponding to that of the Q_0 output, appears on the serial output line. Device 6 now responds to the clock inputs and, after shifting the data out, its Q_S output goes into a high impedance mode. The LOW on the $\overline{ORE}$ output of device 6 enables device 7. This process continues until the last data bit has been shifted out of device 8, at which time its $\overline{ORE}$ output goes LOW. This activates the $\overline{TOS}$ inputs of devices 5...8 and new data can then be loaded from the stack when available. The timing diagram for 16 bits of serial data extraction is shown in Figure 6-4.

Data can be extracted from the array in parallel by activating the TOP inputs of device 5...8 LOW. New data is loaded into the output registers on the LOW-to-HIGH transition of this input. The TOP and $\overline{EO}$ inputs can be connected together so that data can be automatically extracted.

Figure 6-4 Serial Data Extraction for FIFO Array



Automatic Priority Scheme

Most conventional FIFO designs provide status signals analogous to the $\overline{IRF}$ and $\overline{ORE}$ outputs. However, when these devices are operated in arrays, unit-to-unit delay variations require external gating to avoid transient false-status indications. This is commonly referred to as composite-status signal generation. The design of the 'F403/'F433 FIFO eliminates this problem. An automatic priority feature is built in to assure that a slow device will automatically predominate, irrespective of location in the array.

In Figure 6-2, devices 1 and 5 are defined as "row masters." Devices 2, 3 and 4 are "slaves" to device 1 while devices 6, 7, and 8 are slaves to device 5. The row master is established by sensing the $\overline{IES}$ input during the period when the $\overline{MR}$ input is LOW. Because of the initialization, the $\overline{IRF}$ outputs of all devices are HIGH for a short time after the HIGH-to-LOW transition of the $\overline{MR}$ input. Thus $\overline{IES}$ inputs of all devices except 1 and 5 are HIGH. This condition is sensed by the device logic to establish the row mastership.

All devices in any given row transfer data from their input registers into the corresponding stacks simultaneously. However, no slave can initialize its input register until its $\overline{IES}$ input goes HIGH. Thus initialization starts with the row master and eventually ends at the last slave in the row.

A similar situation occurs at the output registers of all devices in a row. They are loaded simultaneously from corresponding stacks; however, the $\overline{ORE}$ output of a slave cannot go HIGH until its $\overline{OES}$ input is HIGH. Thus the row master is the first to indicate a HIGH on its $\overline{ORE}$ and eventually the slaves will follow. It should be pointed out that this automatic priority scheme reduces the maximum operation speed of the array. If speed is essential, the master-slave hierarchy can be replaced by the traditional composite-status signal-generation scheme, which requires external gating.

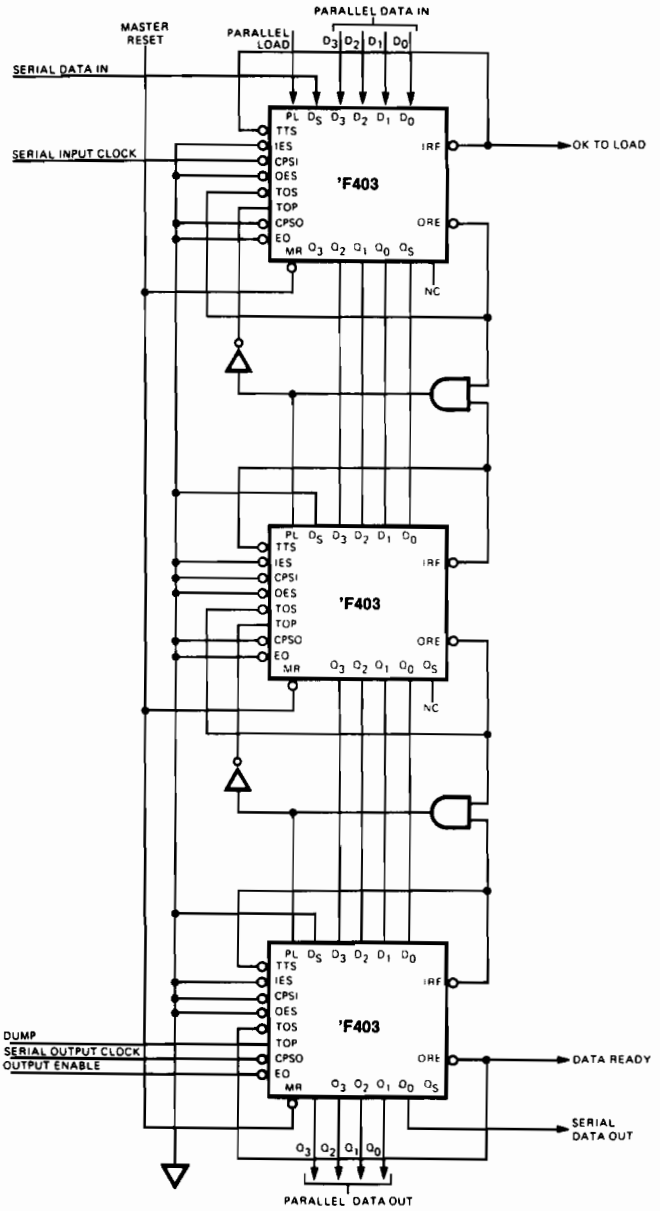
Other Expansion Schemes

The expansion scheme illustrated in Figure 6-2 is quite simple and straightforward. It does not require any external support logic to achieve the desired expansion and retains all the serial/parallel features. However, these advantages are not without sacrifice: one storage location is eliminated at the interface between rows, and the n-row array has storage capacity of 15n + 1 instead of 16n words. Moreover, the automatic priority scheme results in a ripple action from row master to the last slave in that row for the status signaling. This reduces the maximum operation frequency of an array and the inherent speed of the individual devices is not fully utilized.

The 'F403/'F433 FIFO, because of its versatility, can be used to overcome the above disadvantages with minimum external logic. A vertically expanded array, consisting of three FIFOs, yields 48 words of storage (Figure 6-5). After initialization by a LOW level on the $\overline{MR}$ inputs, the $\overline{IRF}$ outputs of all three devices are HIGH and the $\overline{ORE}$ outputs LOW. The AND gates at the row interface are thus disabled. The PL inputs of devices 2 and 3 are LOW. Now, if the input register of device 1 receives four bits of data, then $\overline{IRF}$ output goes LOW. This activates the $\overline{TTS}$ input and the data falls through into the output register of device 1 and the $\overline{ORE}$ output becomes HIGH. Since the $\overline{IRF}$ output of device 2 is HIGH from initialization, the AND gate between devices 1 and 2 is enabled and the PL input of device 2 becomes HIGH. Data from device 1 is loaded into the input register of device 2 causing the $\overline{IRF}$ output of device 2 to go LOW. Moreover, a HIGH level on the PL input of device 2 results in a LOW level on the TOP input of device 1. As a result, the $\overline{ORE}$ output of device 1 also becomes LOW. Either way, the AND gate is disabled and the PL input of device 2 goes LOW and the TOP input of device 1 becomes HIGH.

The LOW level on the $\overline{IRF}$ output of device 2 activates its $\overline{TTS}$ input and initiates a fall-through action; the data appears at the output register. Because the TOP input of device 1 is HIGH, new data arrives at the device-2 output register. When data appears at the output of device 2, the AND gate at the interface of devices 2 and 3 is enabled. By a similar action described above, device 3 takes the data word into its input register and passes it on to the output. Thus, if 16 words are loaded at the input to the array, the 1st word is located in the output and the 16th word is in the input register of device 3. Device 3 is full now and its $\overline{IRF}$ output remains LOW until data is extracted. This LOW level disables the AND gate between devices 2 and 3 and hence any arrival of new data into the output register of device 2 does not activate the PL input of device 3. As new data is received, it is arranged in devices 1 and 2 so that the 17th data word falls into the device 2 output register and the 48th word remains in the input register of device 1. Forty-eight data words fill all devices in the

Figure 6-5 Expansion without Sacrificing a Storage Location at the Interface



array. Under these conditions, the status output is as follows: the $\overline{IRF}$ outputs of devices 1, 2, and 3 are LOW and the $\overline{ORE}$ outputs of devices 1, 2 and 3 HIGH.

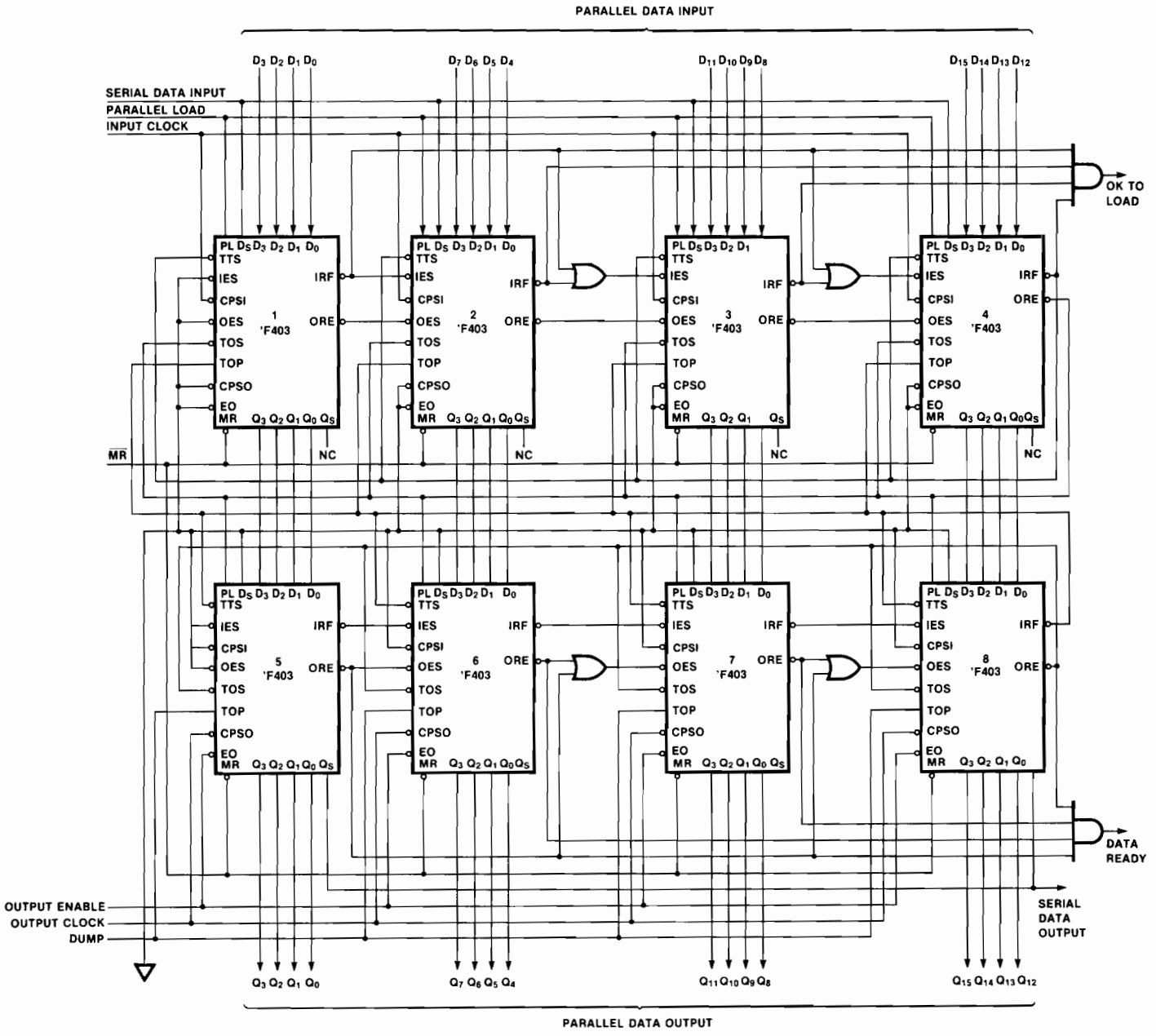
The data extraction takes place when the TOP input of device 3 is activated; normally it is HIGH. To extract data, TOP is made LOW and then HIGH. When the TOP input is LOW, the $\overline{ORE}$ of device 3 goes LOW. When TOP is returned HIGH, data is demanded from the stack.

The internal control in device 3 loads the second data word into the output register and the $\overline{ORE}$ goes HIGH.

The internal control also initiates a fall through action in device 3. Thus, the 16th data word that was located in the input register is transferred into the device 3 stack and the input register is initialized. Thus, the $\overline{IRF}$ output of device 3 becomes HIGH.

The 17th data word is located in the output register of device 2, hence the $\overline{ORE}$ output is HIGH. When the $\overline{IRF}$ output of device 3 becomes HIGH, the AND gate at the interface causes the PL input of device 3 to go HIGH and the TOP input of device 2 to go LOW. The 17th data word then goes into the input register of device 3. The internal control of device 2 initiates fall-through action so that the 18th word falls into the output and the 32nd word is transferred into the stack. This results in a HIGH at the $\overline{IRF}$ output of device 2. Similar action takes place between devices 1 and 2 with the net result that all data has fallen one location creating a vacancy in the input register of device 1. It is now clear that this FIFO array has a 48-word capacity without affecting the serial/parallel data feature at the input or the output. It can then be concluded that if an array of n rows is constructed using the proposed scheme, the effective storage capacity of the FIFO is 16n words.

Figure 6-6 Expansion with Priority Defeated for Faster Operation



The array of Figure 6-6 has all the features and yet operates at a higher speed than the array shown in Figure 6-2. Whenever the $\overline{IRF}$ output of device 1 is HIGH, the $\overline{IES}$ inputs of devices 2, 3 and 4 are also HIGH. Therefore, when the array is initialized by a LOW level on the $\overline{MR}$ inputs, device 1 is the row master and devices 2, 3 and 4 are the slaves. In the second row of devices, the $\overline{IRF}$s and $\overline{IES}$s are interconnected so that device 5 is also a row master and devices 6, 7 and 8 are slaves.

When serial data is entered into the array, device 1 receives the first four bits of data. Devices 2, 3 and 4 do not respond to the clock since all three $\overline{IES}$ inputs are HIGH. After the 4th bit, the $\overline{IRF}$ output of device 1 is LOW. This disables device 1 from responding to the clock and enables device 2 so that the next four bits are entered into device 2. Devices 3 and 4 remain disabled by a HIGH level on the $\overline{IES}$ inputs. After the 8th bit, the $\overline{IRF}$ of device 2 becomes LOW, thus disabling device 2 and enabling device 3. After the 12th bit, the $\overline{IRF}$ output of device 3 is LOW and thus device 4 is enabled. After the 16th bit, the $\overline{IRF}$ output of device 4 is LOW. So far, the serial data entry into this array is identical to that for the array in Figure 6-2.

The LOW level on the $\overline{IRF}$ output of device 4 activates the $\overline{TTS}$ inputs of all 4 devices, causing the transfer of data into the stacks. Although all devices transfer data into the stack simultaneously, device 1 (row master) is the first to initialize its input register. Since devices 2, 3 and 4 are slaves, they need a HIGH on their $\overline{IES}$ inputs for input-register initialization. As soon as the $\overline{IRF}$ output of device 1 goes HIGH due to initialization, the $\overline{IES}$ inputs of devices 2, 3 and 4 become HIGH and their input registers are initialized simultaneously. This is in contrast to Figure 6-2 where device 3 has to wait for device 2 to initialize, etc. The ripple action of input initialization has been overcome by simple gating. The $\overline{IRF}$ outputs of devices 1, 2, 3 and 4 are fed into 4-input AND gates to generate the composite input status. To obtain an indication that the input register of the array is empty, the input register of each device in the first row should be empty.

The $\overline{ORE}$ and $\overline{OES}$ interconnections for the second row are essentially similar to the input section. This gating at the output section eliminates the rippling effect of the output status indication. If the gating arrangement used in Figure 6-5 is incorporated into the array of Figure 6-6, the result is a 32-word x 16-bit FIFO network.

Other FIFO Applications

Disc Controller Serialize/Deserialize Logic

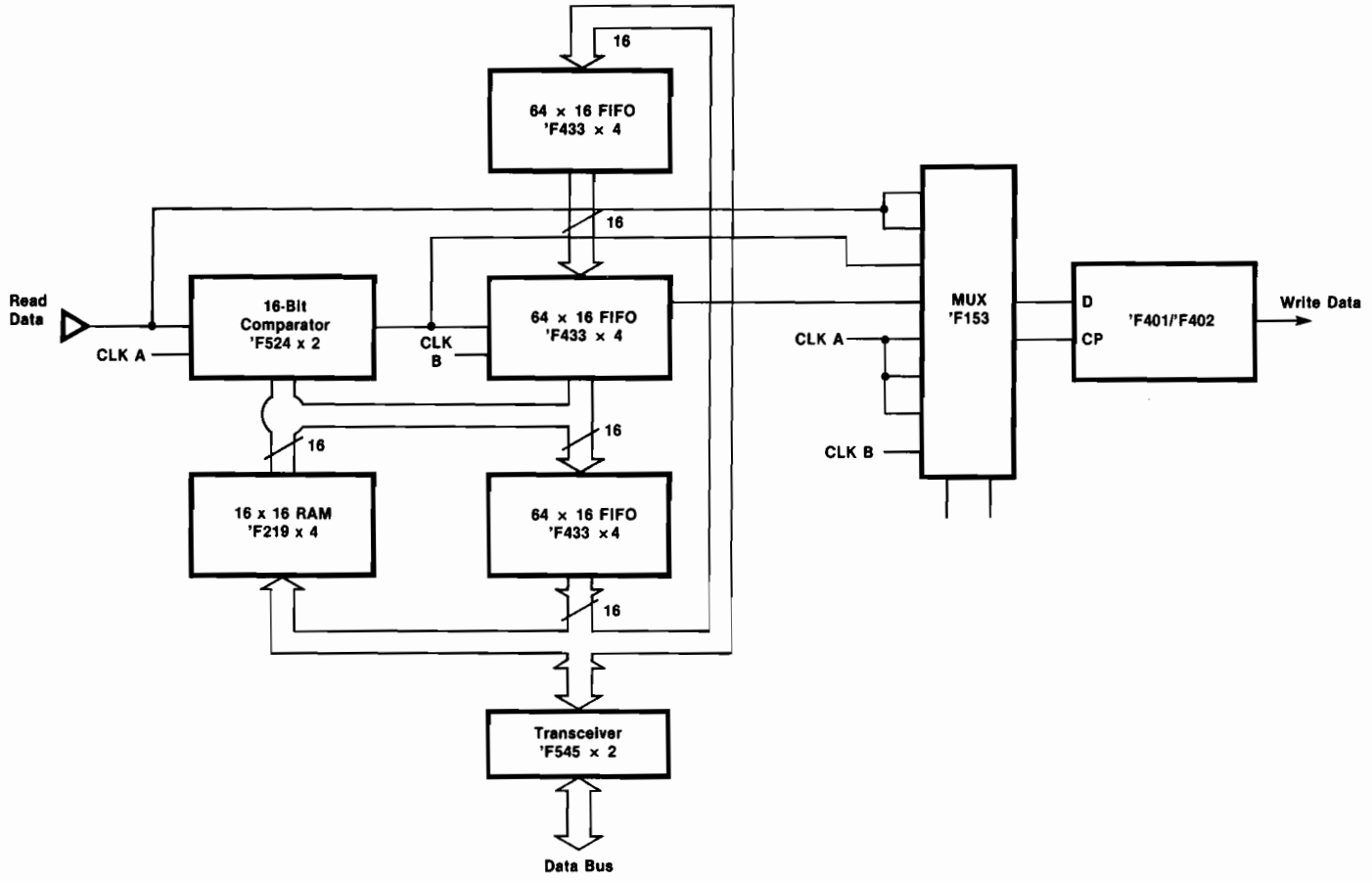


Figure 6-7

The scheme shown in Figure 6-7 transfers data between memory and disc. This scheme utilizes embedded headers and CRC error checking for maximum data integrity. This architecture is usually controlled by a simple state machine for maximum speed, which may also drive a DMA controller. For clarity the control logic has been omitted.

The required sector for reading or writing is identified by comparing the data stream read from disc with the correct header previously stored in the 16x16 scratch pad RAM. The 16-bit comparator is formed by cascading two 'F524 comparators. Disc data is entered serially into the shift register of the 'F524 where it is compared on a bit by bit basis with sequential header data on the I/O lines provided by the header scratch pad RAM formed by four 'F219 16x4 RAM chips. The first match found will be the block synchronization character. When this match is made, the comparison switches to a word by word basis and the rest of the header is then checked and any failure to match on a word will vector the controller to a suitable error routine.

The header block normally contains unit, track, head, sector, block length, encoding format. The header block is separated from the data block by an interblock gap normally of zeros. This allows for write gate turn off when the header is written, and write gate turn on when the data block is written. Filling the interblock gap with zeros gives a smooth data run-in to the data block synchronization character.

The data block begins with a synchronization character followed by a fixed length data field, as defined in the header with a data block CRC appended to the end of the block. This is then followed by another interblock gap to the header of the next sector. For designs that use interchangable media, industry standard formats are used, but for fixed media the choice of format is at the discretion of the designer.

The transfer of data to and from the disc is via a FIFO stack constructed from 'F433 64x4 FIFOs. The stack shown is 256 bytes deep and can be constructed with 8 or 12 devices, 12 devices being used for bit by bit read after write verification.

On a read data transfer, data is entered serially into the FIFO after the data block sync character is detected. The FIFO automatically loads the data onto the next stack location when a word has been entered. Data will ripple forward to the output of the FIFO where it will be transferred to memory in parallel under DMA control. CRC accumulation begins with matching the sync character and all incoming data including the written CRC and inputting to the CRC generator checker. If the block is error free the error flag on the CRC generator checker will go low for one clock period indicating an error free block. The CRC generator checker can be achieved using the 'F401 or the 'F402, depending on the chosen CRC polynomials.

To write data to the disc, the data block synchronization character is first loaded into the FIFO, followed by the parallel transfer of memory data under DMA control. The disc is read until the header of the correct sector to be written is identified, including a correct CRC. The write gate is turned on and a string of zeros is written in the interblock gap. The synchronization character and the data block are shifted out of the FIFO serially to the CRC generator checker. The FIFO stack automatically loads the next data word into the output shift register when the last bit has been output. The stack then automatically ripples forward. The CRC generator checker is held reset until the synchronization character has been output to the disc, and CRC accumulation begins with the first data word. When the entire data block has been passed through the CRC generator-checker, the accumulated CRC is appended to the data stream and written to disc followed by a guard byte of zeros. The write gate is then turned off in the interblock gap.

Data and clocks to the CRC generator checker can be multiplexed with a 'F153 under control of the state machine; 'F545s make ideal data bus drivers for this type of application.

Read after write verification can be accomplished easily with this architecture. If the data block and FIFO are the same length, then the write data can be transferred in parallel automatically to the top of the FIFO stack after the synchronization character has been written. In this manner when the data block is reread, the contents of the FIFO are compared to the data from the disc on a word by word basis. Should the data block be longer than the FIFO stack, the FIFO stack would have to be reloaded from memory before starting the verify sequence.

This architecture can be used at data rates up to 12.5 MHz, with the limiting factor being the maximum serial clock rate of the FIFOs. If external shift registers are used then this can easily run at 30 MHz.

'F411 FIFO Controller

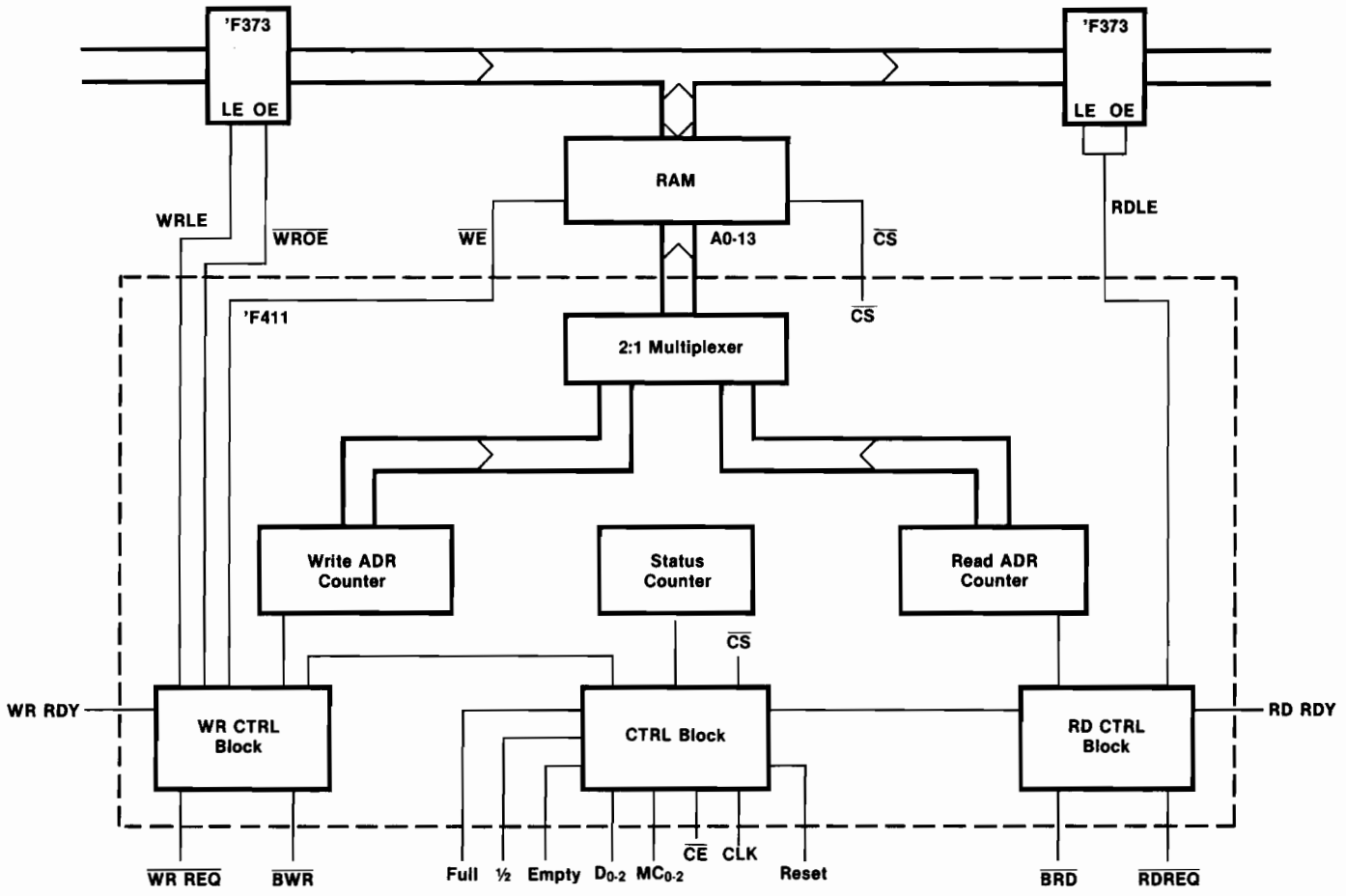


Figure 6-8

The 'F411 FIFO RAM Controller allows a static RAM to be used as a FIFO. In this unidirectional implementation, sufficient 'F373 latches are used to accomodate the word width. There is no restriction on word width as long as the output drive capability of the 'F411 is not exceeded. Buffering can be added if necessary. The static RAM used may be any TTL-compatible device from 128 words to 16K words. All necessary control and timing signals for both the RAM and the buffer latches are provided by the 'F411.

The dashed box shows 'F411 internals:
14-bit read/write address counters;
14-bit up/down status counter;
Read output control signal RDLE;
Write control signals WRLE and WROE.

'F411 Bidirectional FIFO

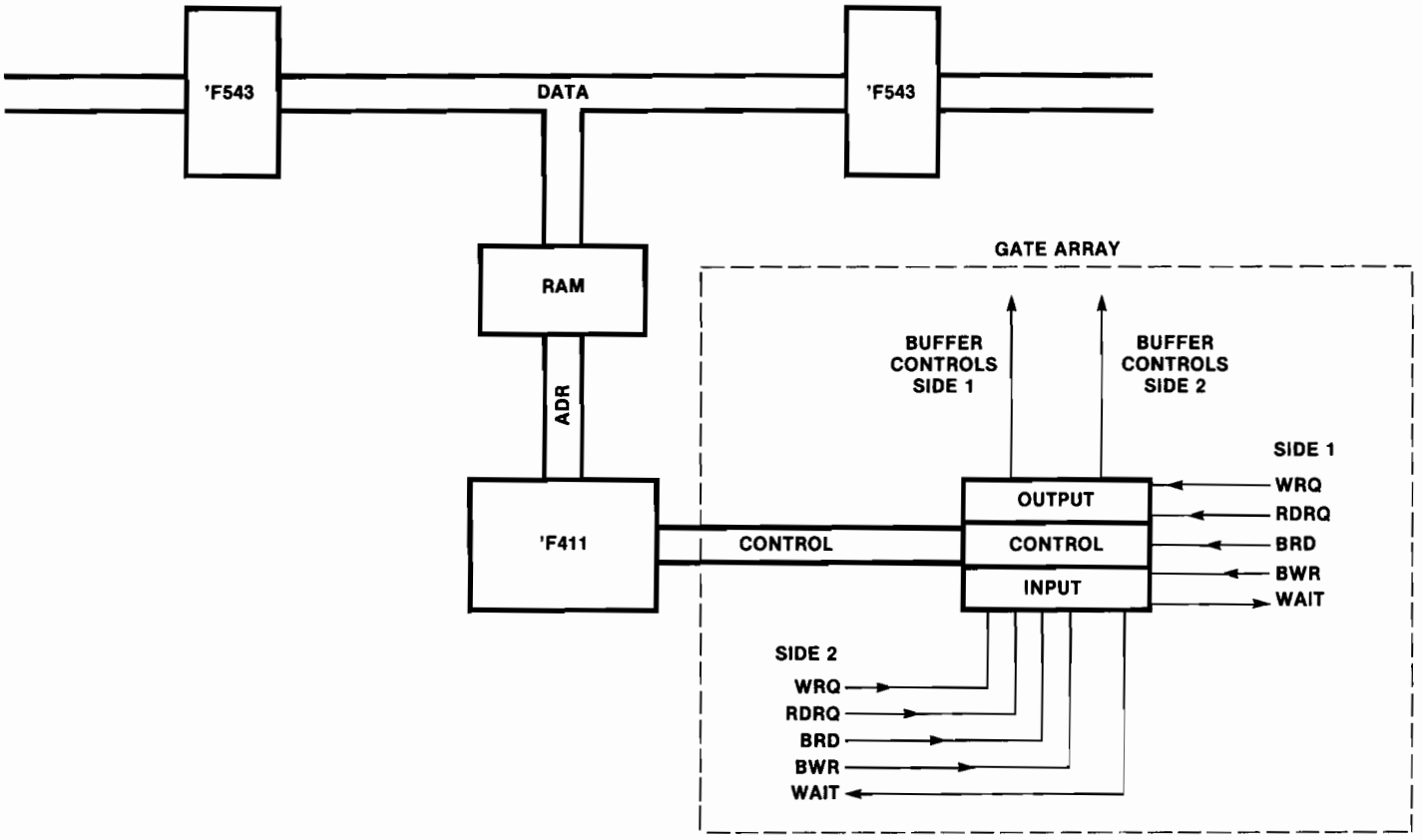


Figure 6-9

In many systems a bidirectional FIFO is required, the general scheme of which is shown in Figure 6-9 and expanded upon in Figures 6-10 through 6-12. In this case 'F543/'F544 latched transceivers and some external SSI logic are used. The control block may be implemented with SSI circuits or programmable logic (for example, Fairchild's FAST-PLAs).

'F411 Bidirectional FIFO Controller—SSI Implementation

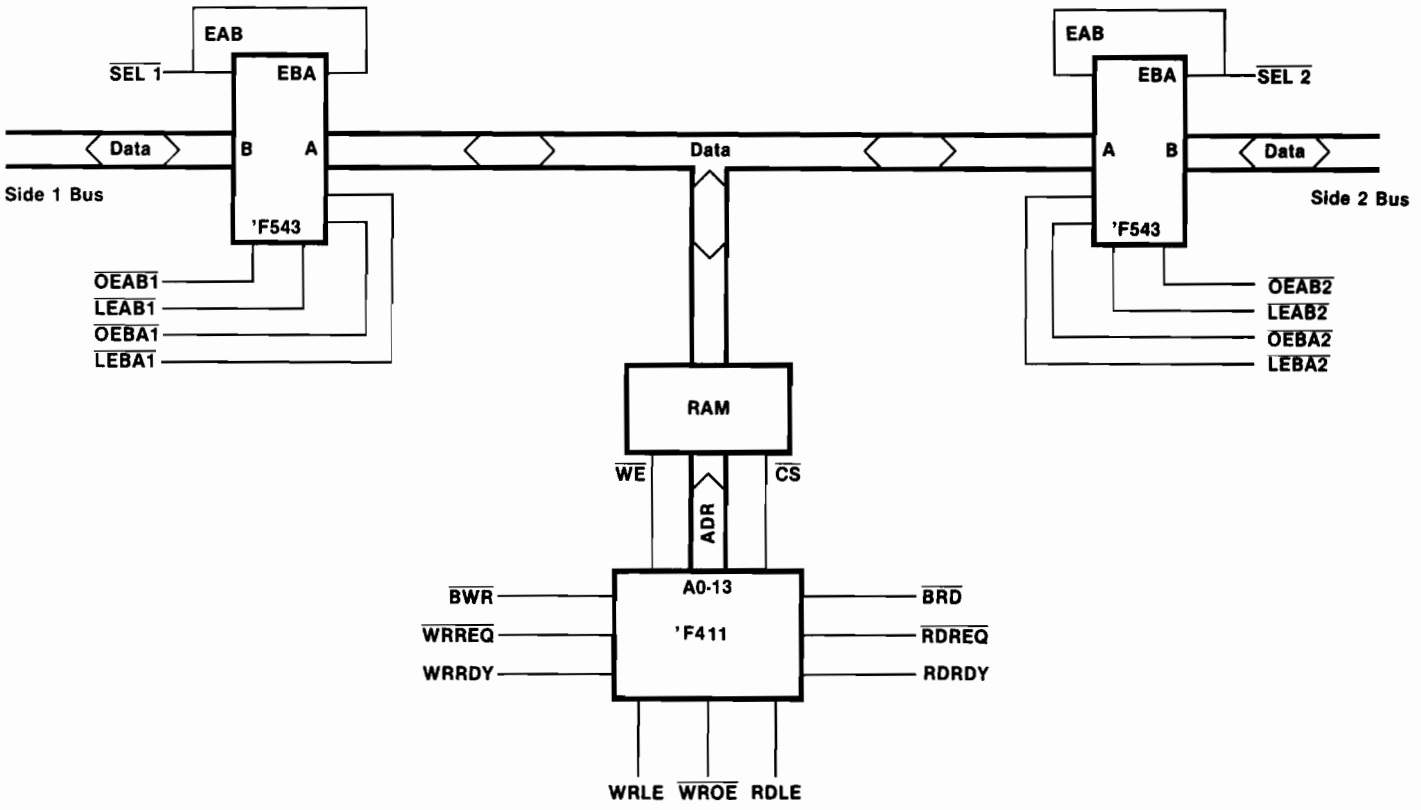


Figure 6-10

In many systems applications such as disk and tape controllers, asynchronous interprocessor communication in multiprocessor systems, a bidirectional FIFO is required. This can be implemented using the 'F411 FIFO controller and 'F543/'F544 latched transceivers with some SSI logic. Bus arbitration is achieved with an exclusive latch (Figure 6-11a); Bus 1 and Bus 2 are scanned for any valid read or write request. A valid request is passed to the 'F411 FIFO controller by the exclusive latch of Figure 6-11a. The 'F543 latched transceiver controls signals via the 'F157 input selector and control signal gating under control of SEL 1 and SEL 2 signals (Figure 6-11b). If both Bus 1 and Bus 2 request access simultaneously, the exclusive latch will access the first to arrive or remain in its last configuration and the appropriate wait signal will be active. The status of latch will not change until the cycle is finished and the request inactive; the latch will then change state and the wait line will be made inactive. A new cycle can then begin. The 'F411 does not generate an RDOE signal so this has to be regenerated as $\overline{OEAB}$ using a 'F32 and the inactive state of RDLE (Figure 6-12). External logic reduces maximum operating speed by 37ns worst case when using the 'F411 FIFO controller in a bidirectional mode.

Bus Arbitration and Input Selector

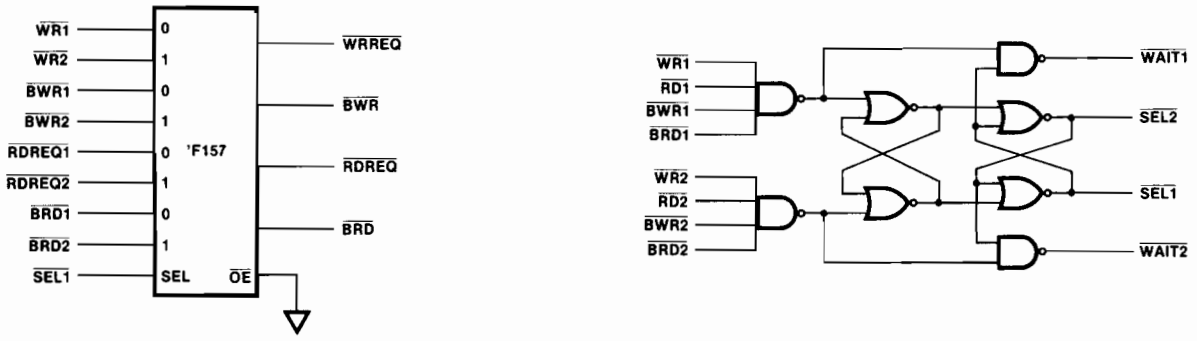


Figure 6-11

6

Buffer Latch Control Signals

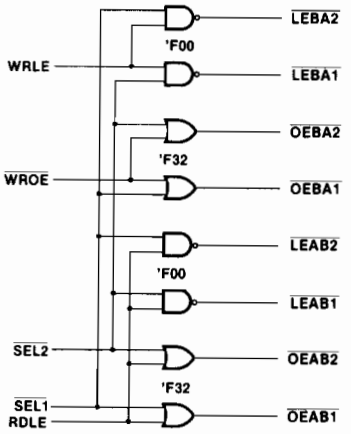


Figure 6-12

FAIRCHILD
A Schlumberger Company

Counters

Introduction

Integrated circuits are used in a wide variety of counting applications in scientific instruments, industrial controls, computers and communications equipment as well as many other areas. Fairchild offers a large selection of FAST counter circuits, differing in complexity, functional versatility, speed, power and cost. All of these FAST counter circuits consist of master/slave flip-flops plus the necessary gating.

It is useful to note, however, that the lowest cost counter does not necessarily ensure the lowest cost system. The versatility of the more complex counters can often save external controls and buffer circuits, thus optimizing both system performance and cost.

Counting with Shift Registers
Fully Synchronous Presettable Counters
Up/Down Counters
Digital Sine Wave Generator
1-of-16 Decoder
Ten Decade Counter
1024 x 1024 Non-Interlaced Video Board

Synchronous Counters with Synchronous Parallel Load
'F160, 'F161, 'F162, 'F163

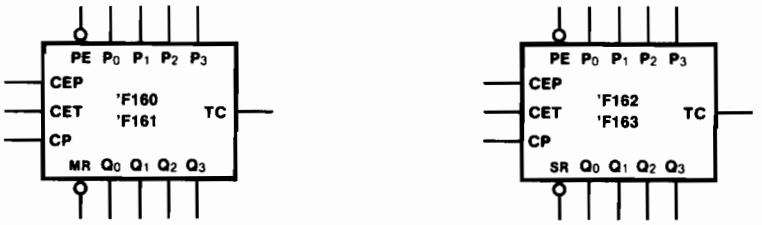


Figure 7-1

The 'F160, 'F162 (BCD) and 'F161, 'F163 (binary) counters offer synchronous counting and synchronous parallel load. The counter flip-flops are positive edge triggered. When the $\overline{PE}$ (active LOW Parallel Enable) input is HIGH and the Count Enable inputs are HIGH, the clock causes the counter to increment. When the $\overline{PE}$ input is LOW, the clock causes the four flip-flops to assume the state of the respective parallel input. Maximum counting frequency is 90MHz, and the delay between the clock edge and any output change is typically only 7.5ns, which is much faster than asynchronous (ripple) counters. Moreover, this synchronous operation can be extended up to 11 packages (44 bits) without additional components by proper use of the decoded Terminal Count (TC) output and the Count Enable inputs.

These counters are ideally suited for operation in synchronous systems. The synchronous parallel load and decoded terminal count make the design of arbitrarily presettable counters easy and free of any race conditions. The synchronous data inputs can be interconnected in such a way that the counter operates either as a shift register or counter controlled by the $\overline{PE}$ input. Data can be shifted out for a multiplexed display or other serial applications.

Synchronous Up/Down Counters
'F190, 'F191, 'F192, 'F193

Figure 7-2

The counters already mentioned are unidirectional, counting up. Some applications require down counters, but often the design can be modified to use an up counter instead, for example, by complementing the parallel input signals. In some cases, particularly in industrial applications, counters that count both up and down are needed.

The simplest bidirectional counters are the 'F192 (BCD decade) and 'F193 (binary) synchronous up/down counters with non-synchronous parallel data entry. The 'F190 and 'F191 are similar except that they contain Terminal Count and Ripple Clock outputs. These counters change state on the LOW-to-HIGH transition of either clock. While counting with one clock input, the other should be held HIGH. Otherwise, the circuit will either count by twos or not at all, depending on the state of the first flip-flop, which cannot toggle as long as either clock input is LOW. Terminal Count Up (code 9 for BCD, code 15 for binary) and Terminal Count Down (code 0 for both counters) are decoded and brought out as active LOW signals. This allows simple cascading of counters but results in an enable ripple delay of about 7.0ns between counters.

The parallel data entry on these counters is asynchronous (independent of the clock), activated only by the active LOW Parallel Load line. This can be an advantage since no clock pulse is required for loading the counter, but parallel load must be timed more carefully than in a synchronous scheme.

Counting with Shift Registers

The FAST devices mentioned so far are designed as counters and are best suited for most counting applications. There are, however, cases where unconventional counting methods are more practical and economical.

Johnson (Moebius) Counters
The twisted-ring (Johnson or Moebius) shift register counter offers simple and glitch-free decoding with 2-input NAND gates. It uses n shift register stages for counting modulo 2n, and therefore is economical only for small values of n. Figures 8-34a and b on page 8-32 show the 'F195 universal 4-bit shift register used as modulo 6 and 8 Johnson counters.

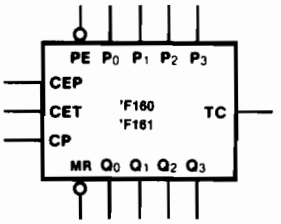
Linear Feedback Shift Register Counters
By applying more sophisticated feedback, the 'F195 can be used to count by any number up to 16, to count up and down, and can easily be programmed, e.g., as a programmable divider for a frequency synthesizer.

Serial Incrementers and Decrementers
In serial incrementer/decrementer systems the functions of data storage and arithmetic operation, normally combined in counters, are completely separated. Data storage is performed in the least expensive way by shift registers. The arithmetic operation is performed by a carry flip-flop and by adders (four bits parallel) or one Exclusive-OR (bit serial).

Serial incrementers are inherently slow, incrementing by one unit for a complete circular data shift. However, they offer several advantages: they can be easily gated to increment or decrement, resulting in up/down counters; their modulo is easily controlled, simplifying the design of mixed modulo counters such as those for timers; they combine naturally with display multiplexers resulting in economical display counters.

'F160/'F161 Counters

Description



Count Enable = CEP•CET•$\overline{PE}$
TC ('F160) = CET•Q_0•$\overline{Q}_1$•$\overline{Q}_2$•Q_3
TC ('F161) = CET•Q_0•Q_1•Q_2•Q_3

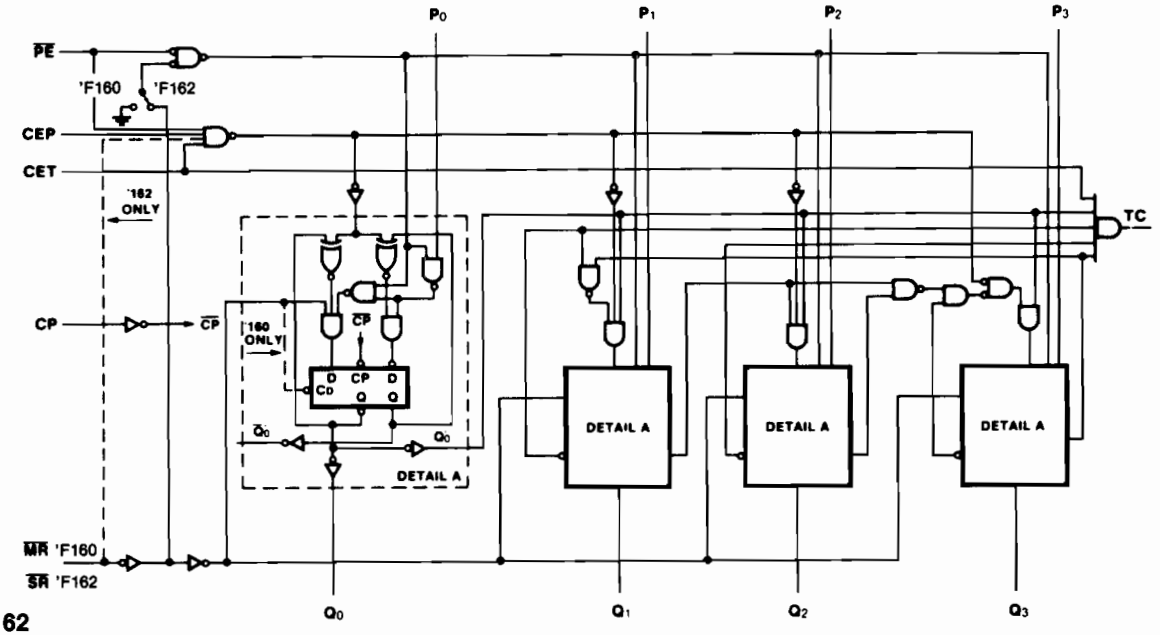
Note: $\overline{PE}$ and $\overline{MR}$ are active LOW inputs. Thus, data on P_{0-3} is loaded into Q_{0-3} by the rising edge of the clock when $\overline{PE}$ is LOW. The counter is asynchronously reset when $\overline{MR}$ is LOW.

Figure 7-3

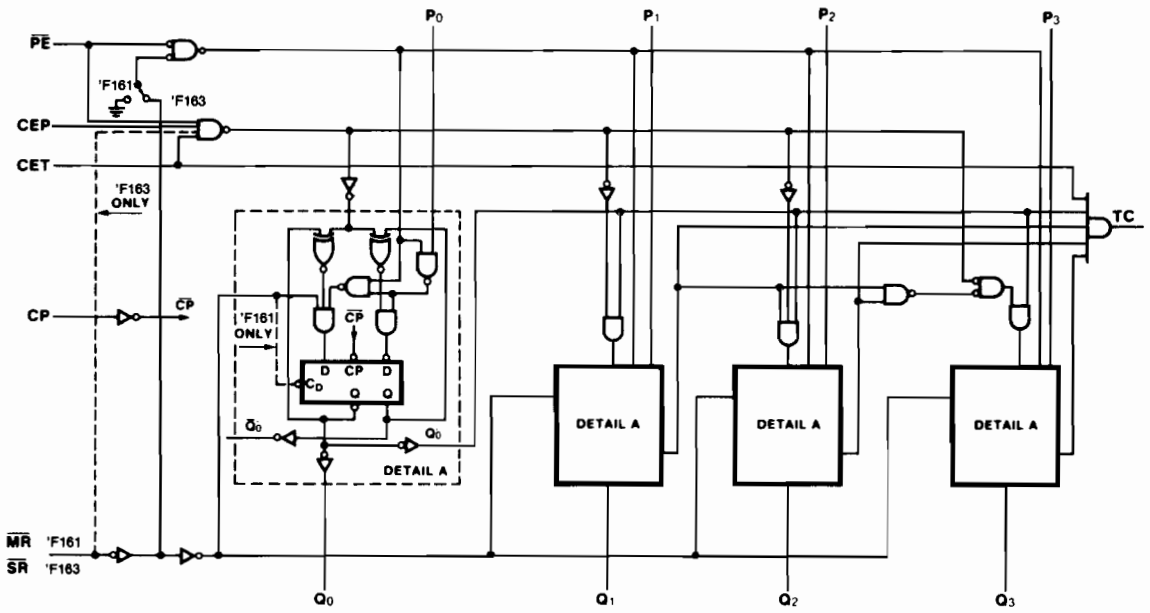
The 'F160 BCD decimal counters and the 'F161 binary hexadecimal counters are multifunction devices with the following features:

- Synchronous parallel loading
- Asynchronous Master Reset
- Assertion output from each stage
- Terminal count activated (HIGH) at count 9 on the 'F160 and at count 15 on the 'F161
- Count Enable Parallel (CEP) input and Count Enable Trickle (CET) input permitting "enable while counting" in high-speed multiple decade counting operations

Operation



a. 'F160/'F162



b. 'F161/'F163

Figure 7-4 Logic Diagrams

Table 7-1 Mode Selection

$\overline{MR}/\overline{SR}$	$\overline{PE}$	CET	CEP	Action on the Rising Clock Edge (⌐)
L	X	X	X	Reset (Clear)
H	L	X	X	Load (P_n—Q_n)
H	H	H	H	Count (Increment)
H	H	L	X	No Change (Hold)
H	H	X	L	No Change (Hold)

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

The 'F160 and 'F162 count modulo-10 in the BCD (8421) sequence. From state 9 (HLLH) they increment to state 0 (LLLL). The clock inputs of all flip-flops are driven in parallel through a clock buffer. Thus all changes of the Q outputs (except due to Master Reset of the 'F160) occur as a result of, and synchronous with, the LOW-to-HIGH transition of the CP input signal. The circuits have four fundamental modes of operation, in order of precedence: asynchronous reset ('F160), synchronous reset ('F162), parallel load, count up and hold. Four control inputs—Master Reset ($\overline{MR}$), Synchronous Reset ($\overline{SR}$), Parallel Enable ($\overline{PE}$), Count Enable Parallel (CEP) and Count Enable Trickle (CET)—determine the mode of operation, as shown in the Mode Select Table 7-1. A LOW signal on $\overline{MR}$ overrides all other inputs and asynchronously forces all outputs LOW. A LOW signal on $\overline{SR}$ overrides counting and parallel loading and allows all outputs to go LOW on the next rising edge of CP. A LOW signal on $\overline{PE}$ overrides counting and allows information on the Parallel Data (P_n) inputs to be loaded into the flip-flops on the next rising edge of CP. With $\overline{PE}$ and $\overline{MR}$ or $\overline{SR}$ HIGH, CEP and CET permit counting when both are HIGH. Conversely, a LOW signal on either CEP or CET inhibits counting.

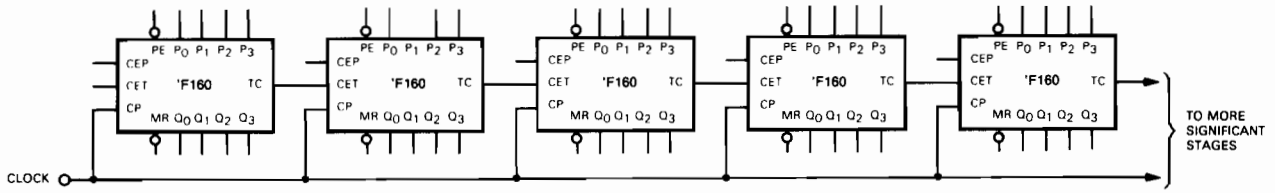
The 'F160 and 'F162 use D-type edge-triggered flip-flops and changing the $\overline{SR}$, $\overline{PE}$, CEP and CET inputs when CP is in either state does not cause errors, provided that the recommended setup and hold times, with respect to the rising edge of CP, are observed.

The Terminal Count (TC) output is HIGH when CET is HIGH and counter is in state 9. To implement synchronous multistage counters, the TC outputs can be used with the CEP and CET inputs in two different ways. The TC output is subject to decoding spikes due to internal race conditions and is therefore not recommended for use as a clock or asynchronous reset for flip-flops, counters or registers. In the 'F160 and 'F162 decade counters, the TC output is fully decoded and can only be HIGH in state 9. If a decade counter is preset to an illegal state, or assumes an illegal state when power is applied, it will return to the normal sequence within two counts, as shown in the State Diagram (Figure 7-6).

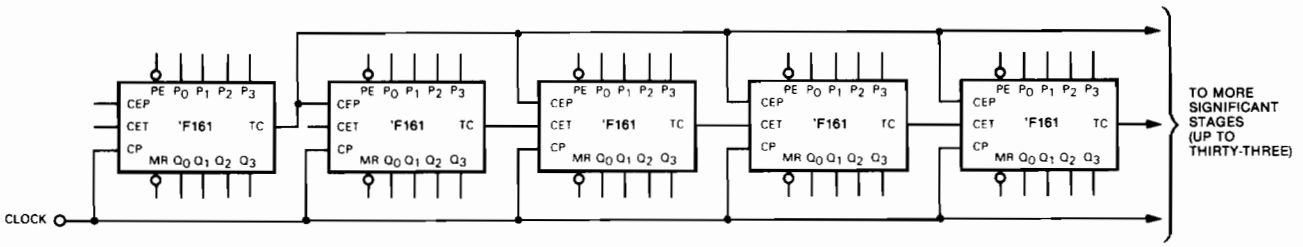
The 'F161 and 'F163 function the same as the 'F160 and 'F162, except that they count in modulo-16 binary sequence. From state 15 (HHHH) they increment to state 0 (LLLL), and the Terminal Count (TC) output is HIGH when CET is HIGH and the counter is in state 15.

To implement synchronous multistage counters (Figure 7-6), the TC outputs can be used with the CEP and CET inputs in two different ways. The TC output is subject to decoding spikes due to internal race conditions and is therefore not recommended for use as a clock or asynchronous reset for flip-flops, counters or registers.

Multistage Counting



a. Slow Multistage Counting Scheme



b. High-Speed Multistage Counting Scheme

Figure 7-5

For multistage counting, all less significant stages must be at their terminal count before the next more significant counter is enabled (Figure 7-5a). This setup requires very few interconnections but has a drawback: the counter chain is fully synchronous, but since it takes time for the enable to ripple through the counter stages, maximum counting speed is reduced. This drawback can be overcome by proper use of the CEP and CET inputs which makes it possible to build a multistage counter (Figure 7-5b) that can operate as fast as a single counter stage.

The advantage of the "enable while counting" method is best seen by assuming all stages except the second and last are in their terminal condition. As the second stage advances to its terminal count, an Enable is allowed to trickle down to the last counter stage, but has the full cycle time of the first counter to reach it. When the TC of the first stage goes active (HIGH), all CEP inputs are activated allowing all stages to count on the next clock. The fan-out of TC (fan-out = 33) limits the configuration in Figure 7-5b to 34 decades.

Programmable Divider

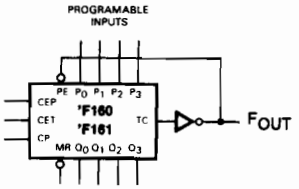


Table 7-2 'F160/'F162

Divide Ratio	Input Required	P_0	P_1	P_2	P_3
2	8	L	L	L	H
3	7	H	H	H	L
4	6	L	H	H	L
5	5	H	L	H	L
6	4	L	L	H	L
7	3	H	H	L	L
8	2	L	H	L	L
9	1	H	L	L	L
10	0	L	L	L	L
11	15	H	H	H	H
12	14	L	H	H	H

L = LOW = Ground
H = HIGH = V_{CC}

Table 7-3 'F161/'F163

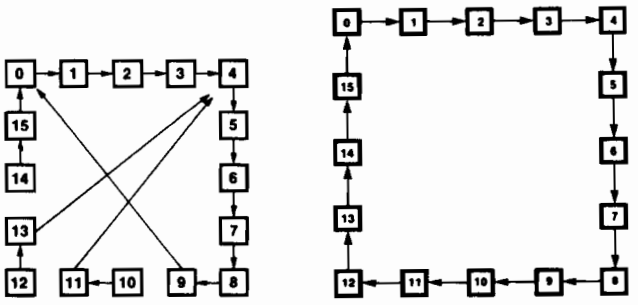
Divide Ratio	Input Required	P_0	P_1	P_2	P_3
2	14	L	H	H	H
3	13	H	L	H	H
4	12	L	L	H	H
5	11	H	H	L	H
6	10	L	H	L	H
7	9	H	L	L	H
8	8	L	L	L	H
9	7	H	H	H	L
10	6	L	H	H	L
11	5	H	L	H	L
12	4	L	L	H	L
13	3	H	H	L	L
14	2	L	H	L	L
15	1	H	L	L	L
16	0	L	L	L	L

L = LOW = Ground
H = HIGH = V_{CC}

The 'F160 and 'F161 can be converted into synchronous programmable counters with modulo 2 to 10 or 2 to 16 respectively, simply by adding a single inverter. The resulting counter actually simplifies certain applications. For example, most applications that seem to require a count either from zero to a predetermined number or from the number down to zero need not be implemented in this way. Instead, the operation can be performed with fewer gates by using a 'F160 (or 'F161) programmed with the complement of the number and allowed to count up until the terminal count is reached. Therefore, if it is necessary to Enable a line for three counts, a 'F160 may be used by entering binary 6 (nines complement of 3) on the parallel inputs and counting up.

The 'F16X counters can be made to divide the incoming clock by utilizing the count sequence internal to the device (Figure 7-6a), as shown in Tables 7-2 and 7-3.

Count Sequence



a. 'F160 b. 'F161

Count Enable = CEP•CET•$\overline{PE}$
TC ('F160) = CET•Q_0•$\overline{Q}_1$•$\overline{Q}_2$•Q_3
TC ('F161) = CET•Q_0•Q_1•Q_2•Q_3
Preset = PE•CP•(rising clock edge)
Reset = MR

The 'F160 can be preset to any state, but will not count beyond 9. If preset to state 10, 11, 12, 13, 14, or 15, it returns to its normal sequence within two clock pulses.

Figure 7-6 State Diagrams

Rotary or Thumbwheel Switch Programming

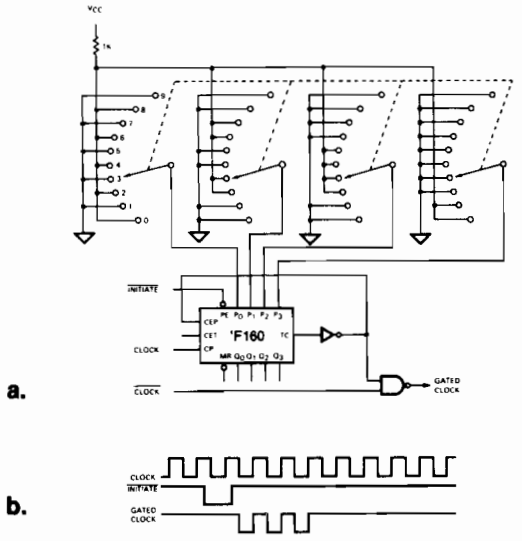
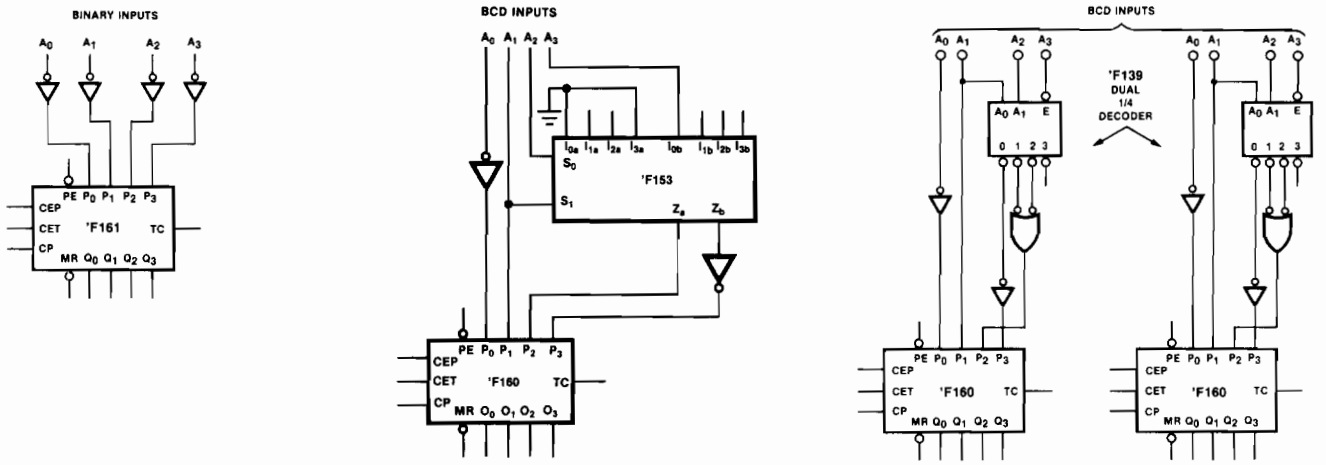


Figure 7-7

This 'F160 is programmed by a 4-pole switch and is used to permit zero to nine pulses to pass through a gate. Various types of readily available thumbwheel switches function well with these devices. An open input appears to be HIGH and must be grounded if a LOW is desired. For this reason, nines complement switches providing closed contacts for HIGH only will not function directly with these devices. A type providing shorts for LOWs must be used. The proper type is sometimes called "nines complement plus complement" or "inverted nines complement" by thumbwheel switch manufacturers.

Generating Complements



a. b. c.

Figure 7-8

These examples show how the complement of a number may be generated with logic circuits. In a binary counter ('F161), the negated outputs of the circuits supplying the number furnish the proper complement. If these outputs are not available, inverters will generate the complement as shown in Figure 7-8a above. In a decimal counter ('F160), the nines complement is generated by a 'F153 dual 4-input multiplexer plus one inverter (Figure 7-8b), or by half of a 'F139 dual 1-of-4 decoder plus suitable gating (Figure 7-8c).

50% Duty Cycle Output Counters

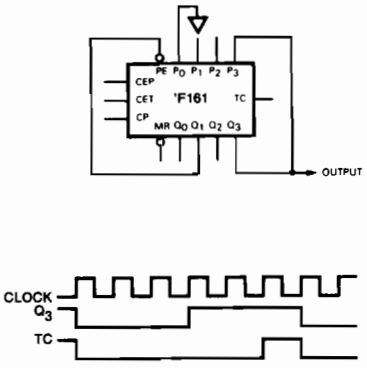
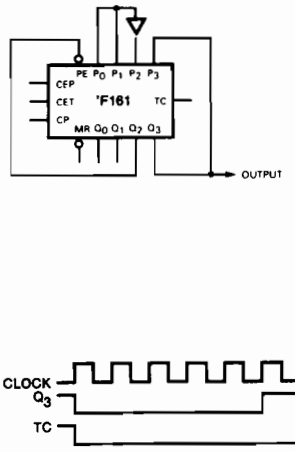


Table 7-4
Count Sequence

Q_0	Q_1	Q_2	Q_3
0	0	0	0
0	1	1	0
1	1	1	0
0	0	0	1
0	1	1	1
1	1	1	1

a. Modulo 6

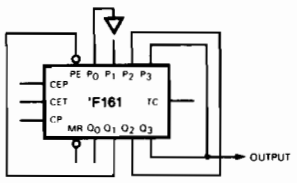
Table 7-5
Count Sequence



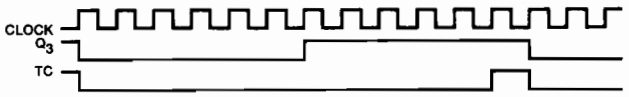
Q_0	Q_1	Q_2	Q_3
0	0	0	0
0	1	0	0
1	1	0	0
0	0	1	0
0	1	1	0
1	1	1	0
0	0	0	1
0	1	0	1
1	1	0	1
0	0	1	1
0	1	1	1
1	1	1	1

b. Modulo 10

Table 7-6
Count Sequence

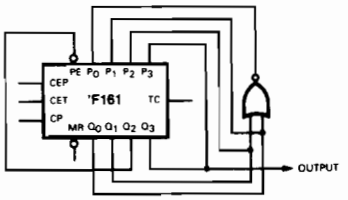


Q_0	Q_1	Q_2	Q_3
0	0	0	0
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0
0	0	0	1
0	0	1	1
1	0	1	1
0	1	1	1
1	1	1	1

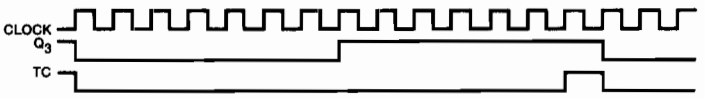


c. Modulo 12

Table 7-7
Count Sequence



Q_0	Q_1	Q_2	Q_3
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0
0	0	0	1
1	0	0	1
0	1	0	1
0	0	1	1
1	0	1	1
0	1	1	1
1	1	1	1



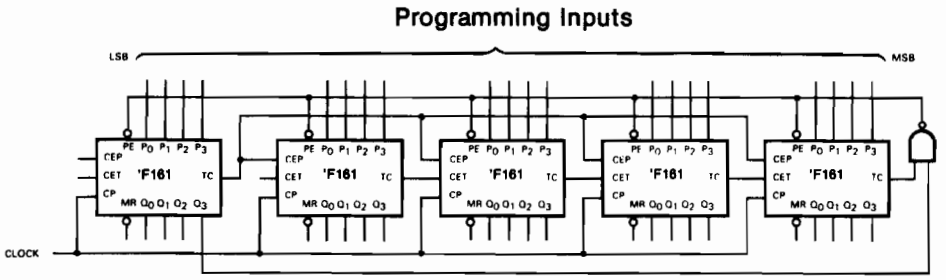
d. Modulo 14

Figure 7-9

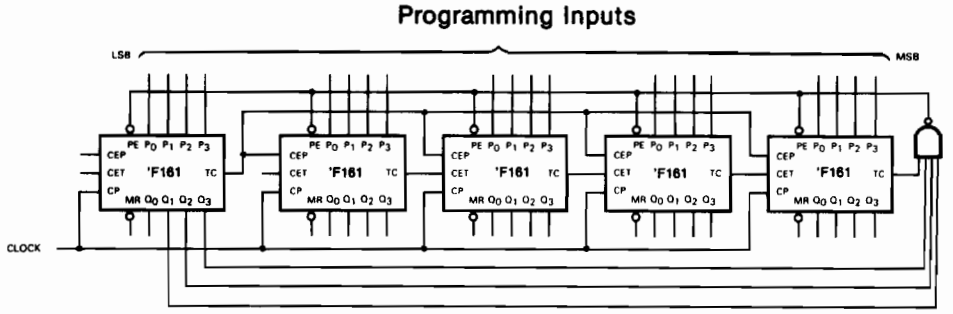
Four circuits which divide by 6, 10, 12, and 14 are shown. The Q_3 output provides a 50% duty cycle output. No additional gates are required except in the divide-by-14 circuit. In addition, all the count sequences start on 0000 and end on 1111, which means the Master Reset ($\overline{MR}$) input and the Terminal Count (TC) output still function properly.

Synchronously parallel loading the 'F161 forces the counter to skip some of the states it would otherwise count through. In each circuit, either the Q_1 or Q_2 output is connected to the active LOW Parallel Enable ($\overline{PE}$) input. Whenever this output is LOW, the counter loads instead of counting on the next clock pulse.

Multistage Programmable Counters



a. Decimal Program with Nines Complement

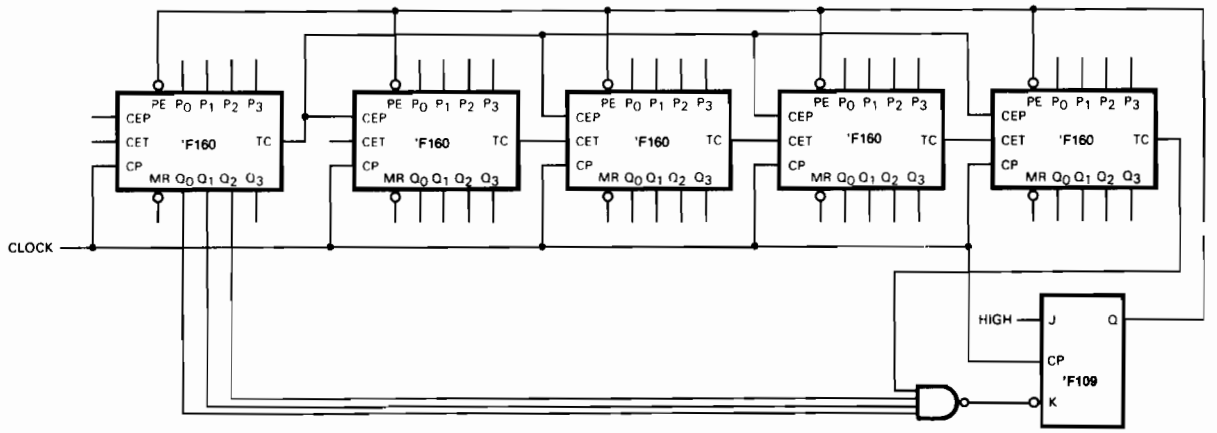


b. Binary Program with Ones Complement

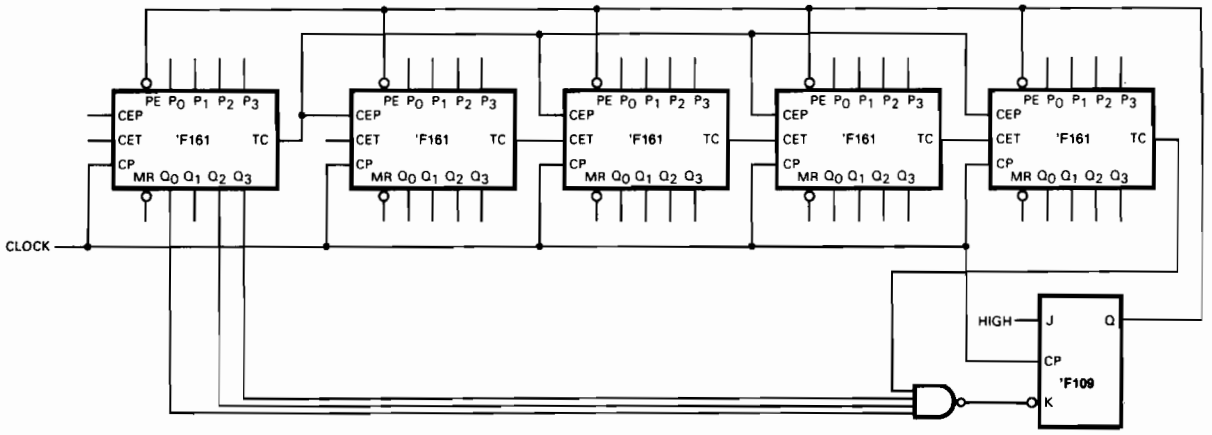
Figure 7-10

In the multistage programmable decimal and binary counters shown above the state prior to Terminal Count (TC – 1) is decoded and activates the $\overline{PE}$ input. Therefore, the next clock pulse does not increment the counter to Terminal Count (all nines for decimal, all ones for binary) but rather loads the program value into the counter. The counters are programmed with the nines or ones complement of the count modulos, instead of the more complicated tens or twos complement used in the conventional approach. The maximum count frequency is limited by the delay in TC decoding and the setup time of the $\overline{PE}$ input. This can be improved with an additional flip-flop as shown in Figure 7-11.

Higher Speed Multistage Programmable Counters



a. Decimal Program with Nines Complement



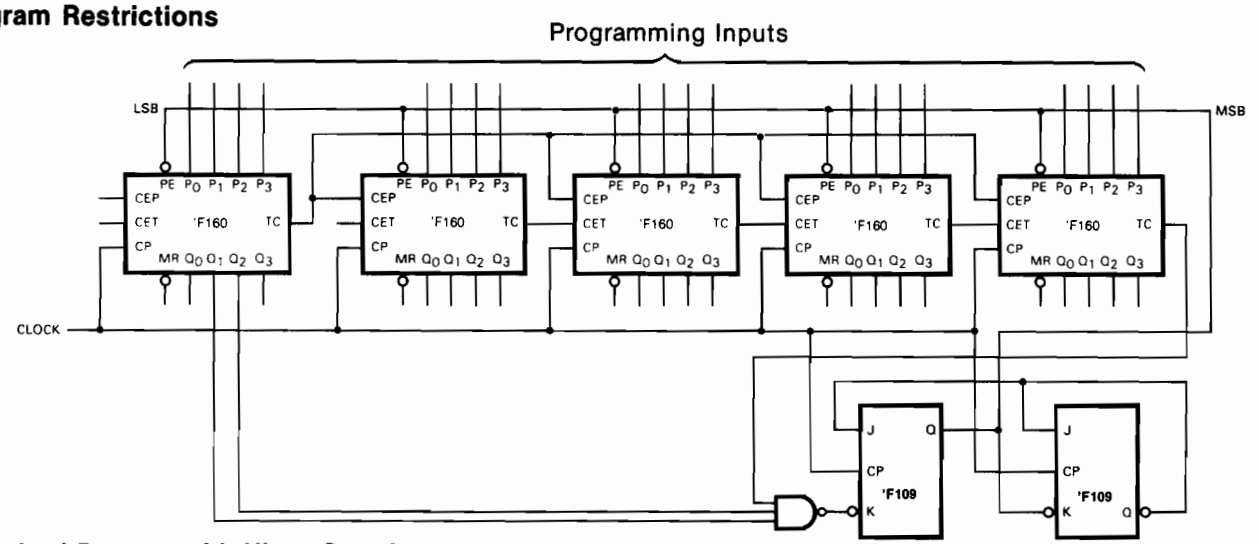
b. Binary Program with Ones Complement

Figure 7-11

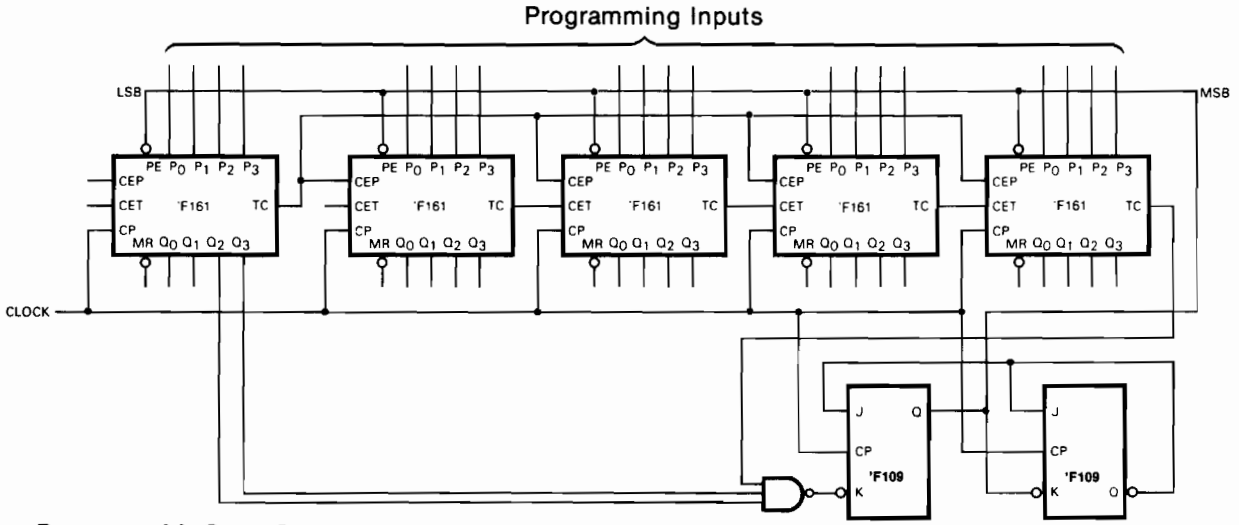
The maximum count frequency of a programmable counter can be improved by decoding the TC – 2 state of the counter and synchronizing this state in a FAST flip-flop such as the 'F109.

The clock pulse that increments the counter to TC – 1 also resets this flip-flop, thus activating the $\overline{PE}$ input. The next clock pulse loads the counter with the program value. Guaranteed count frequency can be as high as 25MHz, limited only by the sum of the t_{PD} of the flip-flop plus the setup time of the $\overline{PE}$ inputs.

Fast Multistage Programmable Counters Without Program Restrictions



a. Decimal Program with Nines Complement



b. Binary Program with Ones Complement

Figure 7-12

The programmable counters described in the previous two examples suffer from a decrease in maximum counting speed when they are programmed with certain unfavorable numbers which do not allow enough time for the decay of the TC ripple chain.

For example, assume that a BCD counter is programmed for modulo 90. The counting sequence is:

MSD	99996	LSD
	99997	
	99998	activates $\overline{PE}$
	99909	nines complement of 90 is loaded
	99910	
	etc.	

The ripple TC output from the MSD must disappear during one clock period (when 99909 has been loaded). If the clock period is shorter than this ripple delay, the next clock pulse reloads and the counter divides by the wrong number. This problem is overcome by a second flip-flop, as shown above.

The dual flip-flop provides additional time for the TC outputs to ripple LOW, since it activates the $\overline{PE}$ signal for two clock pulses instead of one. The two flip-flops form a modulo 3 counter and are normally set. TC – 3 is decoded and activates the reset ($\overline{K}$) input of the first flip-flop. The next clock pulse increments the counter to TC – 2 and resets the first flip-flop. This activates the $\overline{PE}$ inputs and the reset ($\overline{K}$) input of the second flip-flop. The next clock pulse loads the program value into the counter and resets the second flip-flop. Since the first flip-flop remains reset, the following clock pulse loads the counter again and it sets both flip-flops. The next clock pulse increments the counter.

Self-Stopping Counter

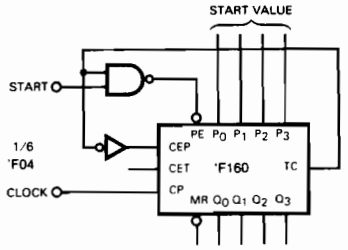


Figure 7-13

An inverter between the TC output and the CEP Enable input is all that is needed to stop a counter at its terminal state. However, the inverter must be connected to CEP, not to the CET Enable input. Since TC is a function of CET, such a connection causes oscillation.

Counter With Three Enables

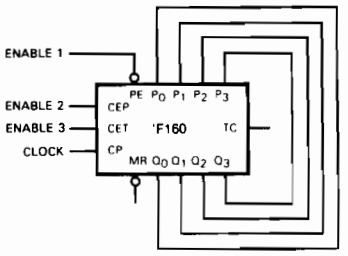


Figure 7-14

If a particular application does not require the programmable feature but does require three Enables on a counter, each Q output should be tied to the corresponding P input as shown, permitting the $\overline{PE}$ input to function as an enable.

High-Speed Programmable Divider Stage

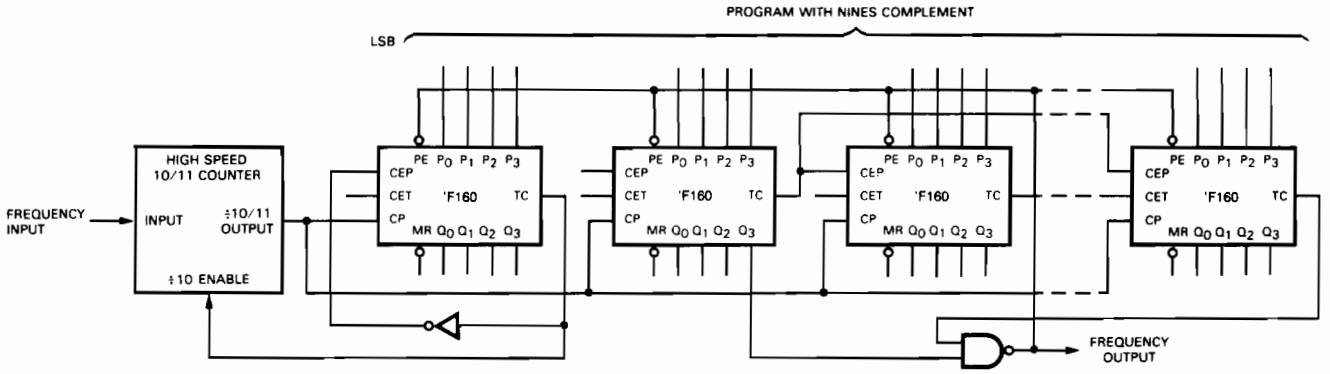


Figure 7-15

The 'F160 and 'F161 can control a higher speed counter that is one of the stages of a programmable divider, extending the operating frequency by a factor of 10. A selectable divide-by-10 or divide-by-11 high-speed counter acts as the clock source for the subsequent counter stages and for a control counter. The control counter and 10/11 counter form the units counter of the programmable divider. After loading the 'F160s, the control counter enables the divide-by-11 mode of the high-speed counter for the duration of the units cycle. When the units cycle is over, the control counter enables the divide-by-10 mode, causing the high-speed counter to become a divide-by-10 counter functioning as a normal decade units counter stage. This technique is known as pulse swallowing and is frequently used in high frequency synthesizers.

Up/Down Counters

The 'F160 and 'F161 combined with gating circuitry can be used as up/down counters, but the 'F192 and 'F193 up/down counters are generally more applicable for this function.

Special Applications

The synchronous parallel load capability of the 'F160 and 'F161 enables these counters to be used in applications not customarily associated with counters. For example, if the $\overline{PE}$ input of the 'F160 ('F161) is permanently grounded, the device behaves like four dual-rank D-input flip-flops, decreasing the number of different parts needed in certain systems.

Resynchronizer

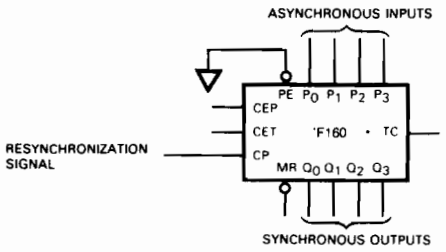


Figure 7-16

A resynchronizer using an 'F160 (or 'F161) as four D-input flip-flops is shown. In this circuit the $\overline{PE}$ input is grounded, and the resynchronizing input is applied to the CP input. In most cases, the 'F195 universal shift register is preferable for this function.

Combination Counter/Serial Registers

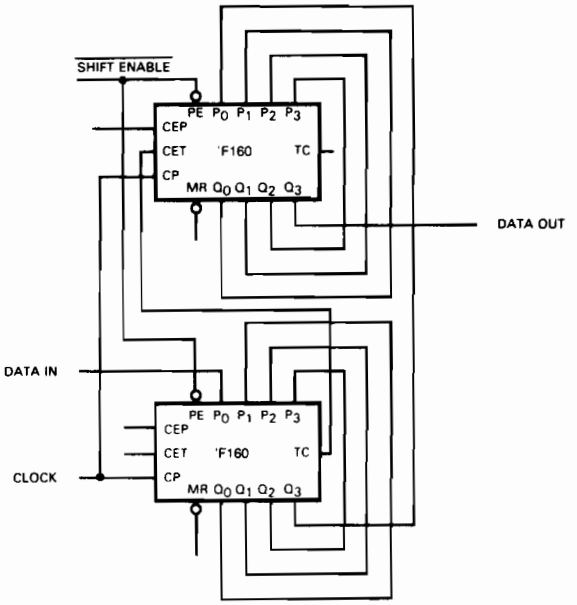


Figure 7-17

The contents of a 'F160 or 'F161 can be shifted out serially. This circuit shows each 'F160 (or 'F161) with Q outputs connected to the next more significant P input for shifting the most significant bit first. When the Shift Enable is HIGH, the 'F160s (or 'F161s) perform the normal counting operation. When the Shift Enable is activated (LOW), the contents of the two decade counters are shifted out serially at the Q_3 output of the second decade counter.

If each 'F160 (or 'F161) output is connected to the next less significant parallel input, the least significant bit is shifted out first. Q_0 of the least significant stage provides the serial output.

Combination Counter/Decade Register

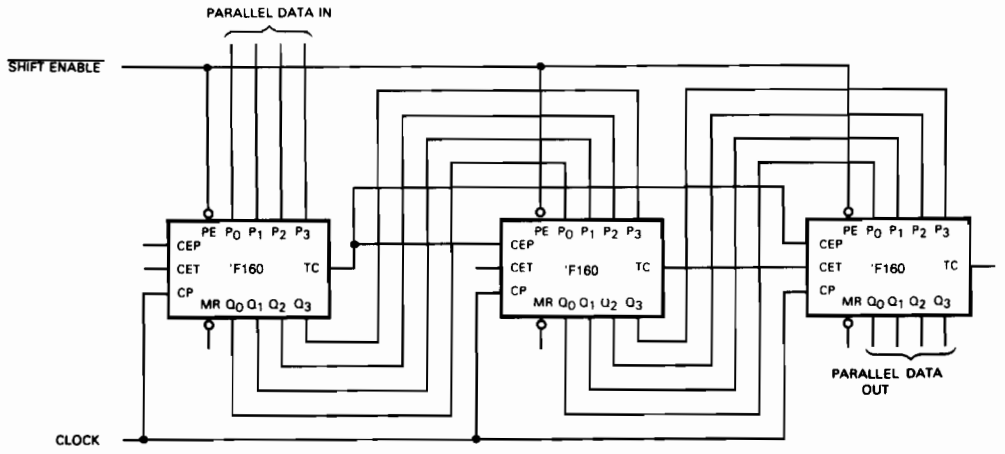


Figure 7-18

The decade counter illustrated shifts data out of each counter in parallel by decade, with the most significant decade shifted first. To shift the least significant decade first, each output is connected to the next less significant decade parallel inputs.

Programmable Counters Starting at Zero

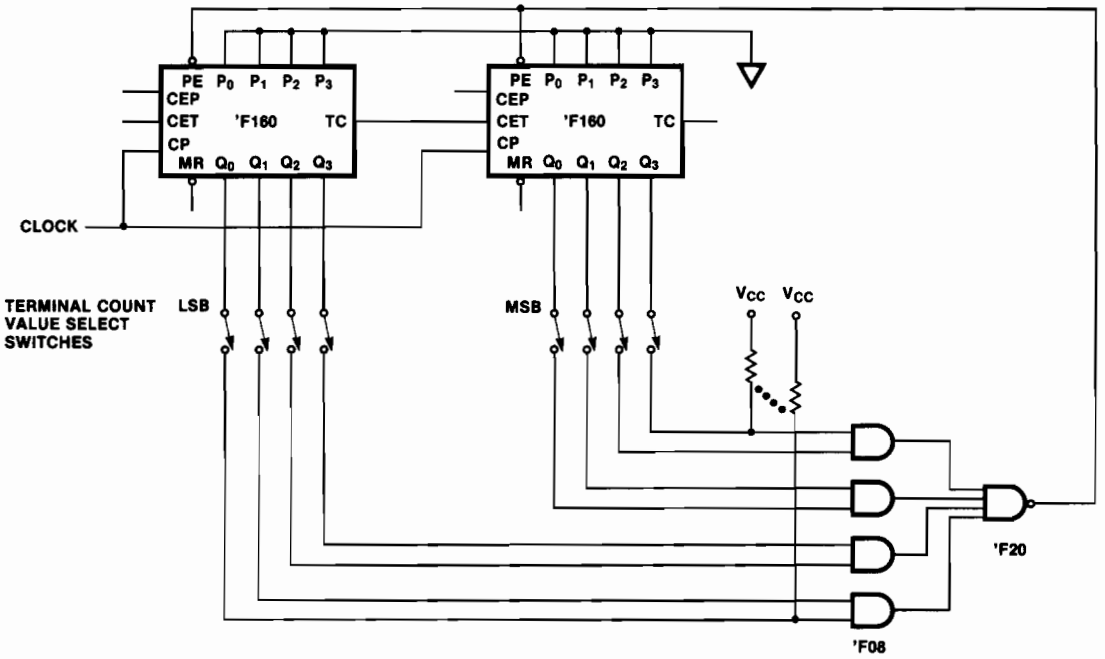
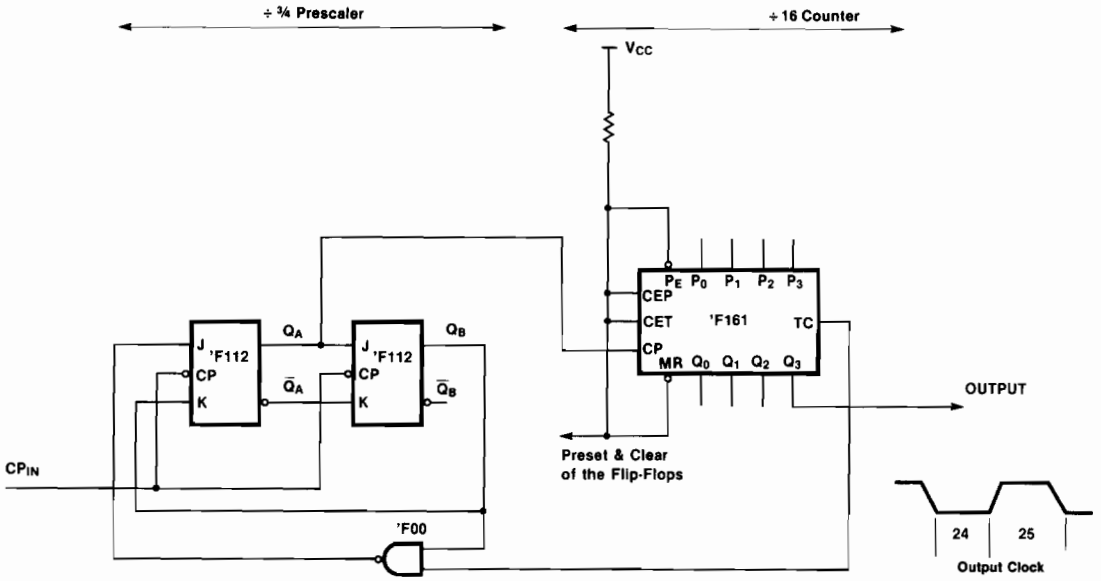


Figure 7-19

Some applications call for counters that start at zero and reset at a predetermined number. This number may be derived from a switch position, from internal logic signals, or it may be permanently fixed. A NAND gate decodes the terminal count, and on the next clock pulse the decoded-output resets the counter through the $\overline{PE}$ inputs. The $\overline{PE}$ inputs make it possible to reset the counters synchronously by grounding all the P inputs. If the asynchronous $\overline{MR}$ input were used to reset the counter, a race condition would result.

Divide-by-49 Circuit



49 = 3x15 + 4x1
Prescaler divides by 3 as long as TC is LOW (15 cycles)
Prescaler divides by 4 when TC is HIGH (1 cycle)

Table 7-8

Q_a	Q_b	Q_a	Q_b
H	L	H	L
H	H	H	H
L	H	L	H
H	L	L	L
etc.		H	L
TC = LOW		TC = HIGH	

Figure 7-20

This figure illustrates a scheme to derive an unusual division from a counter by modifying the prescaler to divide by three or four as a function of the terminal count of the 'F161. The prescaler divides by three for the fifteen cycles when the TC is LOW and divides by four for the remaining cycle when the TC is HIGH. This is one simple example of how to implement a division by 49; many other variations of this scheme are possible. The high speed of the FAST family allows this design to be operated at speeds in excess of 100MHz.

Cyclic D/A Conversion

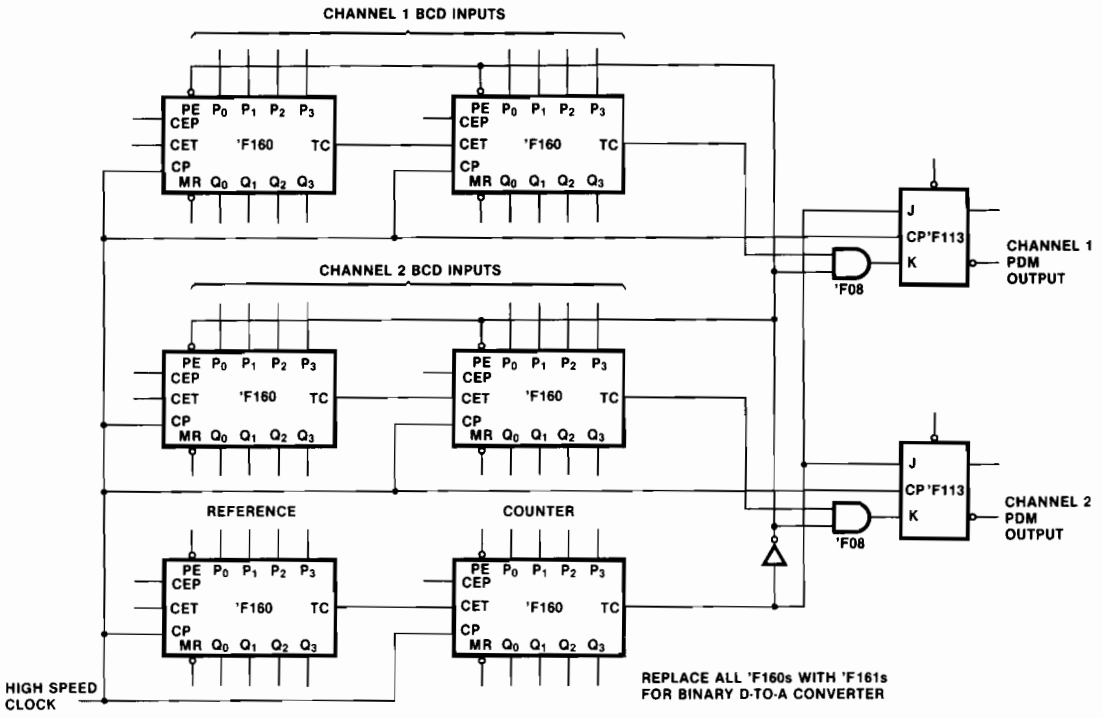


Figure 7-21

This is a schematic of a dual D/A Pulse Division Multiplex (PDM) converter using the 'F160s (or 'F161s). This circuit uses one programmable counter per channel plus one reference counter. The number to be converted is supplied to the parallel inputs of the programmable counter and entered when the reference counter is at Terminal Count (TC active). The programmable counter value is greater than the value of the reference counter by a number of counts equal to the digital input. To produce a PDM output directly proportional to the phase difference of the two counters, the Terminal Counts of the programmable and reference counters alternately set and reset the 'F113 flip-flop. Additional D/A conversion channels are easily obtained by adding a programmable counter and flip-flop for each channel desired. An alternate D/A converter (more economical for multichannel conversion) using the 'F148 priority encoder is included in the encoder section.

'F192/'F193 Up/Down Counters

Description

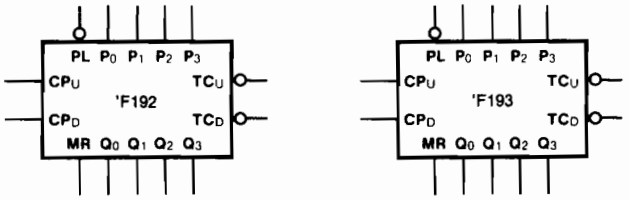


Figure 7-22

The 'F192 is an up/down decade (8421) counter and the 'F193/'F193 is an up/down 4-bit binary counter. Both devices are synchronous dual-clock up/down counters with asynchronous Parallel Load, asynchronous overriding Master Reset and internal Terminal Count logic which allows the counters to be easily cascaded without additional logic. The 'F192 and 'F193 can be used in many up/down counting applications, particularly when the initial count value must be loaded into the counter and multistage counting is required.

Operating Modes and Count Sequences

The 'F192 and 'F193 can be Reset, Preset and can count up and down. The operating modes of the counters are listed and are identical. The only difference is in their count sequences.

Counting is synchronous with the outputs changing state after the LOW-to-HIGH transition of either the count up clock (CP_U) or count down clock (CP_D). The direction of the count is determined by the clock input which is pulsed while the other count input is HIGH. Incorrect counting can occur if both the count up clock and count down clock inputs are LOW simultaneously.

The counters respond to a clock pulse on either input by changing to the next appropriate state of the sequences shown in Figure 7-25. The 'F192 diagram shows the regular BCD (8421) sequence as well as the sequence of states if a code greater than nine is preset into the counter.

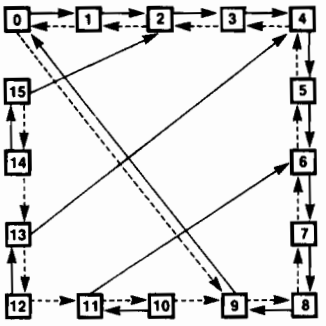
The 'F192 and 'F193 have an asynchronous parallel load capability permitting the counter to be preset. When the Parallel Load (PL) and the Master Reset (MR) inputs are LOW, information present on the parallel data inputs (P_0, P_1, P_2 and P_3) is loaded into the counter and appears on the outputs regardless of the conditions of the clock inputs. When the Parallel Load input goes HIGH, this information is stored in the counter and when the counter is clocked it changes to the next appropriate state in the count sequence. The parallel inputs are inhibited when the Parallel Load is HIGH and have no effect on the counter. A HIGH on the asynchronous Master Reset (MR) input overrides both clocks and Parallel Load and clears the counter. For predictable operation, the Parallel Load and Master Reset must not be deactivated simultaneously.

Table 7-9 Function Table

MR	$\overline{PL}$	CP_U	CP_D	Mode
H	X	X	X	Reset (Asyn.)
L	L	X	X	Preset (Asyn.)
L	H	H	H	No Change
L	H	⌐	H	Count Up
L	H	H	⌐	Count Down

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
⌐ = LOW-to-HIGH Transition

a. 'F192



b. 'F193

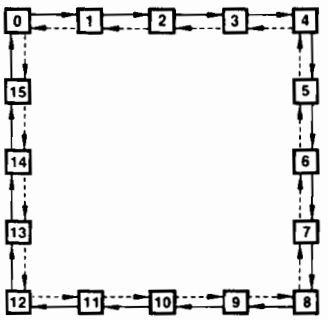
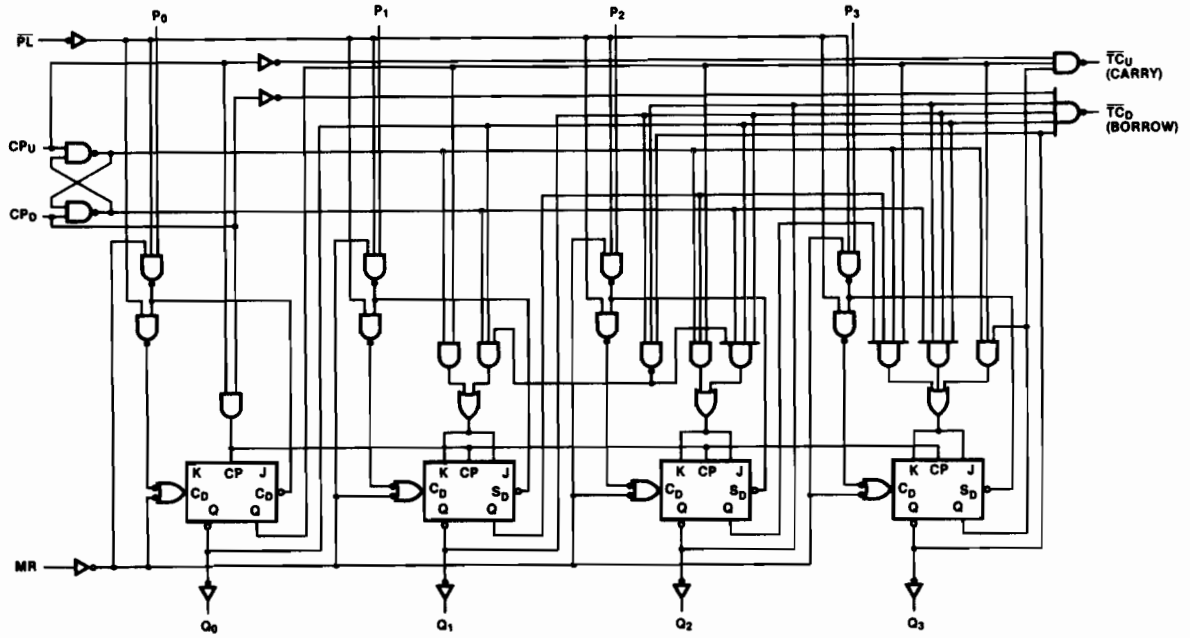
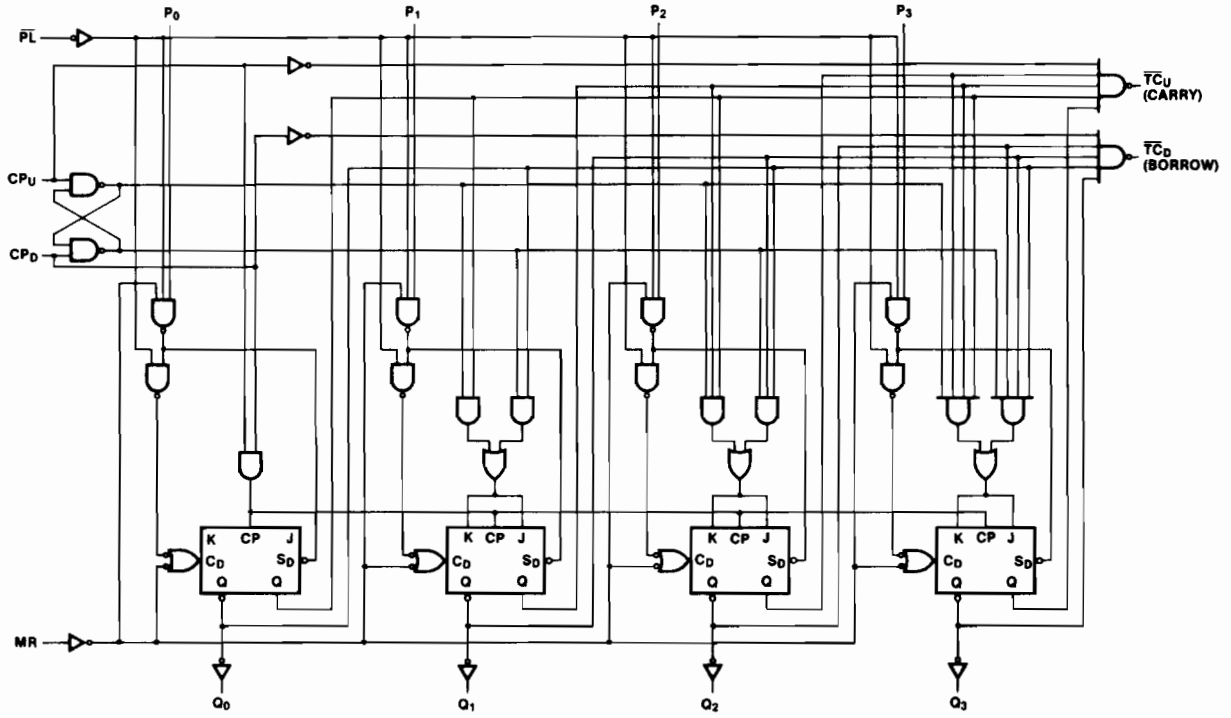


Figure 7-23 State Diagrams

Operation



a. 'F192



b. 'F193

Figure 7-24 Logic Diagrams

Both the 'F192 and 'F193 have four master/slave flip-flops plus steering, Terminal Count decoding and Preset logic. Each flip-flop is designed to toggle after each clock pulse. Counting occurs by steering clock pulses from either the up or down clock input to the appropriate flip-flops. Output changes are coincident, two gate delays after the rising clock edge. The steering logic in the 'F193 allows a particular flip-flop to receive an up clock pulse when all preceding stages are one and to receive a down clock pulse when all preceding stages are zero. The first flip-flop toggles if an up or down clock is received. The 'F192 incorporates slightly

different steering logic to allow decade counting. Each flip-flop is a master/slave toggle flip-flop operating as follows: when the toggle clock input is LOW, the slave is steady but the master is set to the opposite state of the slave; during the LOW-to-HIGH clock transition, the master is disabled so a later change in the slave outputs does not affect the master. Also, the information now in the master is transferred to the slave and appears at the output. When the transfer is completed, the master and slave are steady as long as the clock input remains HIGH. During the HIGH-to-LOW clock transition, the transfer path from master to slave is inhibited, leaving the slave steady in its present state and allowing the master to be set to the opposite state of the slave. Asynchronous Set and Clear inputs on each flip-flop allow the respective flip-flops to be set or cleared independently of the clock inputs.

Terminal Count Logic

The 'F192 and 'F193 have Terminal Count Up ($\overline{TC}_U$) and Terminal Count Down ($\overline{TC}_D$) outputs which allow multistage ripple binary and decade counter operations without additional logic. The Terminal Count Up output is LOW while the up clock input is LOW and the counter is in its highest state (15 for the 'F193, nine for the 'F192). Similarly, the Terminal Count Down output is LOW while the down clock input is LOW and the counter is in state zero. The logic equations for Terminal Count are:

'F192

$$TC_U = Q_0 \bullet \overline{Q}_1 \bullet \overline{Q}_2 \bullet Q_3 \bullet \overline{CP}_U$$
$$TC_D = \overline{Q}_0 \bullet \overline{Q}_1 \bullet \overline{Q}_2 \bullet \overline{Q}_3 \bullet \overline{CP}_D$$

'F193

$$TC_U = Q_0 \bullet Q_1 \bullet Q_2 \bullet Q_3 \bullet \overline{CP}_U$$
$$TC_D = \overline{Q}_0 \bullet \overline{Q}_1 \bullet \overline{Q}_2 \bullet \overline{Q}_3 \bullet \overline{CP}_D$$

Counter Cascading

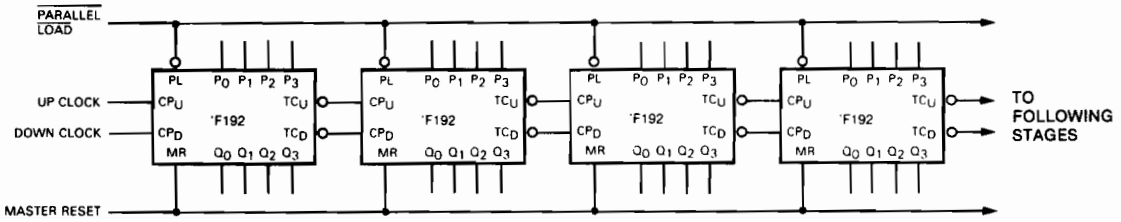


Figure 7-25

The counters are cascaded by feeding the Terminal Count Up output to the up clock input and the Terminal Count Down output to the down clock input of the following (more significant) counter. Therefore, when a 'F193 counter is in state 15 and counting up or in state zero and counting down, a clock pulse will change the counter's state on the rising edge and simultaneously clock the following counter through the appropriate active LOW terminal output. The operation of 'F192 is the same, except when counting up, clocking occurs on gate nine. The delay between the clock input and the Terminal Count output of each counter is two gate delays (typically 7 ns). These delays are cumulative when cascading counters. When a counter is reset, the Terminal Count Down output goes LOW if the down clock is LOW and, conversely, if a counter is preset to its terminal count value, the Terminal Count Up output goes LOW while the clock is LOW.

Light-Controlled Up/Down Counting

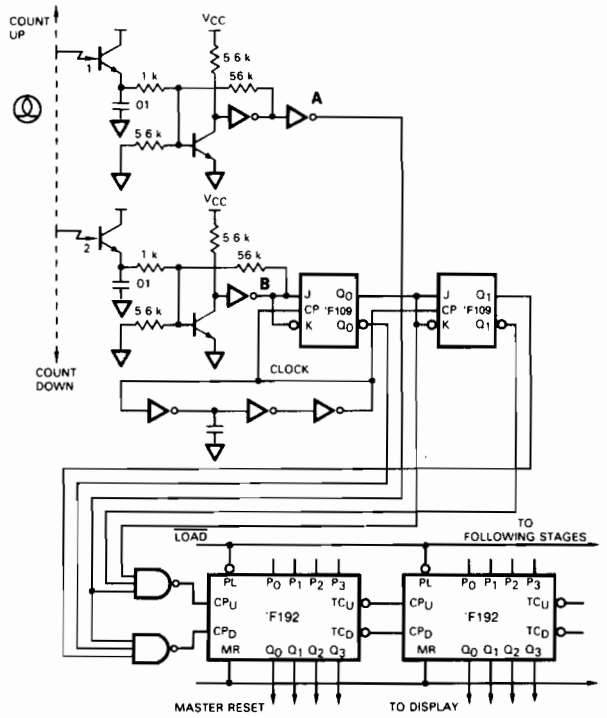


Figure 7-26

Many industrial or scientific applications require a count of objects traveling in different directions. The circuit shown counts moving objects as they move between a light source and phototransistors. This permits a count of objects passing in either direction and allows for reversals in movement or non-uniform movement. Each object passing from bottom to top increments the counter. Each object passing from top to bottom decrements the counter. Any object passing between the light source and the two phototransistors is counted as long as the object is large enough to cover both transistors simultaneously. Hex inverters serve as a clock generator and as phototransistor amplifiers. The dual flip-flop and 3-input NAND gates are used to route the phototransistors' signals to the up/down counters.

When an object moves from bottom to top it covers phototransistor 2 first, bringing line B LOW. This stores a zero in the 2-bit shift register. As the object continues, phototransistor 1 is then covered and brings line A HIGH. As the object moves even further, it uncovers phototransistor 2, bringing line B HIGH again. The next clock pulse loads a 1 into the first bit of the shift register. This one-zero combination in the shift register and HIGH level on line A are decoded and gated with the clock to increment the counter. For an object moving from top to bottom, the sequence is reversed and the counter decremented.

Programmable Divider

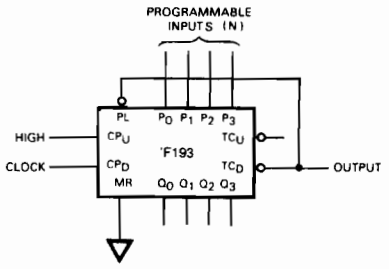


Figure 7-27

Table 7-10 Truth Table

Divide Ratio	Input Required P_0	P_1	P_2	P_3	Decimal Equivalent Of Input
1	H	L	L	L	1
2	L	H	L	L	2
3	H	H	L	L	3
4	L	L	H	L	4
5	H	L	H	L	5
6	L	H	H	L	6
7	H	H	H	L	7
8	L	L	L	H	8
9	H	L	L	H	9
10	L	H	L	H	10
11	H	H	L	H	11
12	L	L	H	H	12
13	H	L	H	H	13
14	L	H	H	H	14
15	H	H	H	H	15

The 'F192 and 'F193 can act as programmable dividers without additional logic. The divide ratio, n, is directly programmable in binary or BCD by using the count down capabilities of either counter. The divider shown operates as follows. The counter counts down until the terminal count value is reached, and when the clock goes LOW again the Terminal Count output goes LOW and starts to load the initial count value into the counter. When the preset number appears on the counter outputs, the Terminal Count Down output disappears and the counter decrements when the clock goes HIGH.

The input clock width must exceed the sum of the Terminal Count delays (two gate delays per counter), the asynchronous load delays and the clock setup time. The Terminal Count output is a short pulse which should be lengthened for some applications before it is applied to other logic. It can be applied to a one-shot or to a flip-flop divide-by-two stage producing a symmetrical output at half the programmed frequency. Although the Terminal Count disappears as soon as a single output changes, the internal delays are such that all flip-flops become preset before Terminal Count disappears. This is because the Preset signal must propagate through both the flip-flop and the Terminal Count gate before the preset signal is removed. Additional delay may be desired between the Terminal Count output and the Parallel Load input and is accomplished with an appropriate number of inverters.

Dead End Counters

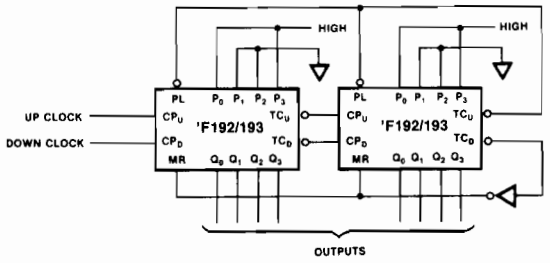


Figure 7-28

Some systems using up/down counters require that underflow or overflow be inhibited. A change from the maximum count to zero in the count up mode or from zero to the maximum count in the count down mode must be prevented. To achieve this limited range operation, the feedback connections illustrated for two 'F192 decade counters are used. The same feedback can be used over any number of stages and can also be used with the 'F193 binary counter. However, 15 must be loaded into the 'F193 to prevent overflow.

The lower limit for counting is established at zero by inverting the Terminal Count Down output and applying it to the Master Reset input. Therefore, when the counter is at state zero and a down clock is applied, Terminal Count Down activates Master Reset during the entire time the clock is LOW, keeping the counter at zero and preventing the counter from decrementing. After the down clock goes HIGH, Master Reset is removed but the counter is still in state zero. The upper limit for counting is established by connecting the Terminal Count up output to the Parallel Load input and applying the terminal count value to the preset inputs.

Single Line Up/Down Control

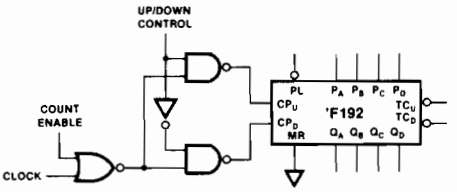


Figure 7-29

Figure 7-29 illustrates the addition of a single line UP/$\overline{\text{DOWN}}$ Control and Enable to the 'F192 and 'F193. All changes in the UP/$\overline{\text{DOWN}}$ Enable should be made while the clock is HIGH.

7

Synchronization and Coincident Pulse Prevention

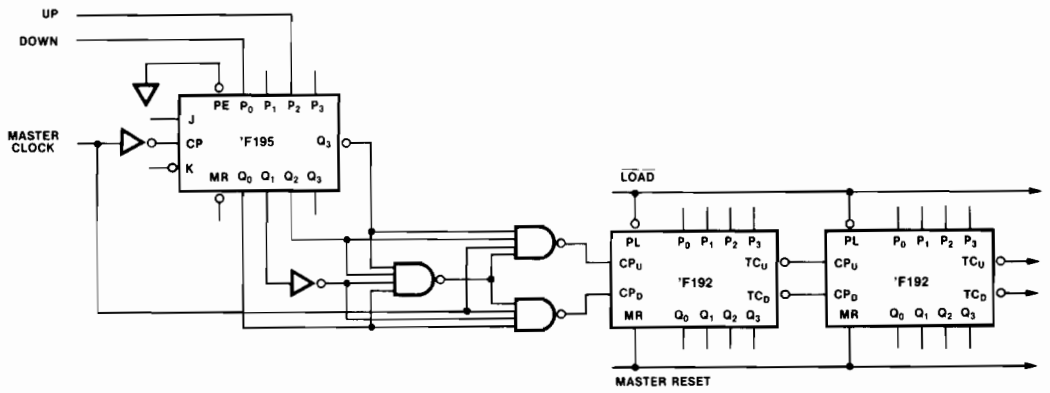


Figure 7-30

A method of synchronizing asynchronous up/down input pulses and avoiding coincident pulses to the counters is illustrated here. The counters increment or decrement when either up or down asynchronous input makes a LOW-to-HIGH transition. If both inputs make the transitions simultaneously, the counters do not decrement or increment. A master clock with a frequency of at least twice the frequency of the asynchronous inputs is needed to avoid the loss of input pulses. The 'F192 counter outputs are synchronized with the master clock.

The asynchronous up/down inputs are fed to a 'F195 4-bit shift register connected to form two independent 2-bit shift registers. The outputs Q_0 and Q_1 reflect the information on the asynchronous down input during two clock periods and the Q_2 and Q_3 outputs reflect information on the up input during the same clock periods. This information is decoded by the three 4-input NAND gates and gated with the clock to produce the 'F192 (or 'F193) up or down pulses. A LOW-to-HIGH transition on the up input results in Q_3 LOW and Q_2 HIGH. This is decoded and gated with the clock to increment the counter. A LOW-to-HIGH transition on the down input results in Q_1 LOW and Q_0 HIGH. This is decoded and gated with the clock to decrement the counter. A 4-input NAND gate disables both clocks to the counter when both transitions have occurred simultaneously.

'F190, 'F191 Synchronous Up/Down Counters

Description

The 'F190 is a synchronous up/down BCD decade counter and the 'F191 is a synchronous up/down 4-bit binary counter. The operating modes of the 'F190 decade counter and the 'F191 binary counter are identical, with the only difference being the count sequences as noted in the state diagrams (see FAST Data Book). Each circuit contains four master/slave flip-flops, with internal gating and steering logic to provide individual present, count-up and count-down operations.

Each circuit has an asynchronous parallel load capability permitting the counter to be preset to any desired number. When the Parallel Load ($\overline{PL}$) input is LOW, information present on the Parallel Data inputs (P_0-P_3) is loaded into the counter and appears on the Q outputs. This operation overrides the counting functions, as indicated in the Mode Select Table.

A HIGH signal on the $\overline{CE}$ input inhibits counting. When $\overline{CE}$ is LOW, internal state changes are initiated synchronously by the LOW-to-HIGH transition of the clock input. The direction of counting is determined by the $\overline{U}$/D input signal, as indicated in the Mode Select Table. When counting is to be enabled, $\overline{CE}$ and $\overline{U}$/D can be changed with the clock in either state, provided only that the recommended setup and hold times are observed.

Two types of outputs are provided as overflow/underflow indicators. The Terminal Count (TC) output is normally LOW and goes HIGH when a circuit reaches zero in the count-down mode or reaches maximum (9 for the 'F190, 15 for the 'F191) in the count-up mode. The TC output will then remain HIGH until a state change occurs, whether by counting or presetting or until $\overline{U}$/D is changed. The TC output should not be used as a clock signal because it is subject to decoding spikes.

The TC signal is also used internally to enable the Ripple Clock ($\overline{RC}$) output. The $\overline{RC}$ output is normally HIGH. When $\overline{CE}$ is LOW and TC is HIGH, the $\overline{RC}$ output will go LOW when the clock next goes LOW and will stay LOW until the clock goes HIGH again. This feature simplifies the design of multi-stage counters, as indicated in Figures 7-31a and b. In Figure 7-31a, each $\overline{RC}$ output is used as the clock input for the next higher stage. This configuration is particularly advantageous when the clock source has a limited drive capability, since it drives only the first stage. To prevent counting in all stages it is only necessary to inhibit the first stage, since a HIGH signal on $\overline{CE}$ inhibits the $\overline{RC}$ output pulse, as indicated in the $\overline{RC}$ Truth Table. A disadvantage of this configuration, in some applications, is the timing skew between state changes in the first and last stages. This represents the cumulative delay of the clock as it ripples through the preceding stages.

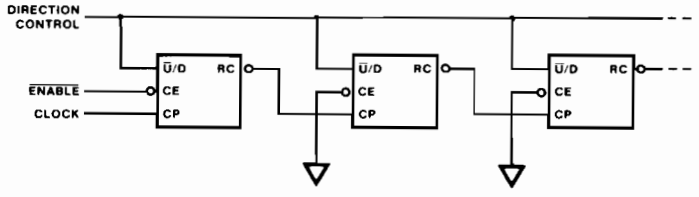
Table 7-11 $\overline{RC}$ Truth Table

Inputs			Output
$\overline{CE}$	TC*	CP	$\overline{RC}$
L	H	⊓	⊓
H	X	X	H
X	L	X	H

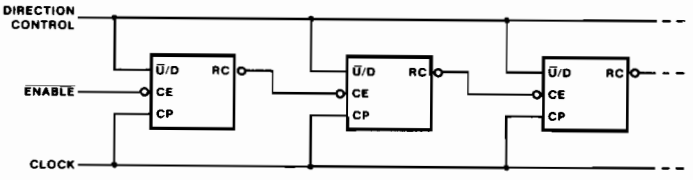
* TC is generated internally
H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

A method of causing state changes to occur simultaneously in all stages is shown in Figure 7-31b. All clock inputs are driven in parallel and the $\overline{RC}$ outputs propagate the carry/borrow signals in ripple fashion. In this configuration the LOW state duration of the clock must be long enough to allow the negative-going edge of the carry/borrow signal to ripple through to the last stage before the clock goes HIGH. There is no such restriction on the HIGH state duration of the clock, since the $\overline{RC}$ output of any package goes HIGH shortly after its CP input goes HIGH.

Figure 7-31



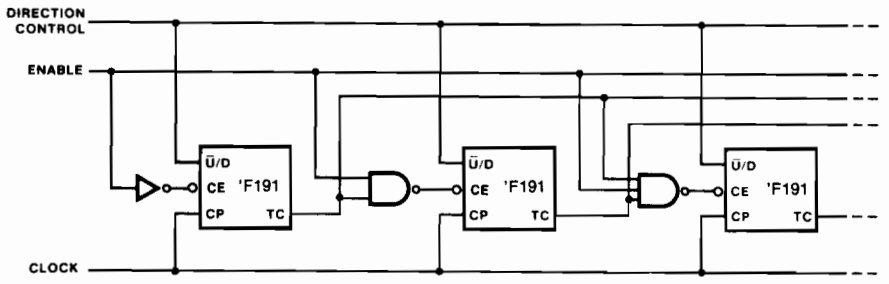
a. n-Stage Counter Using Ripple Clock



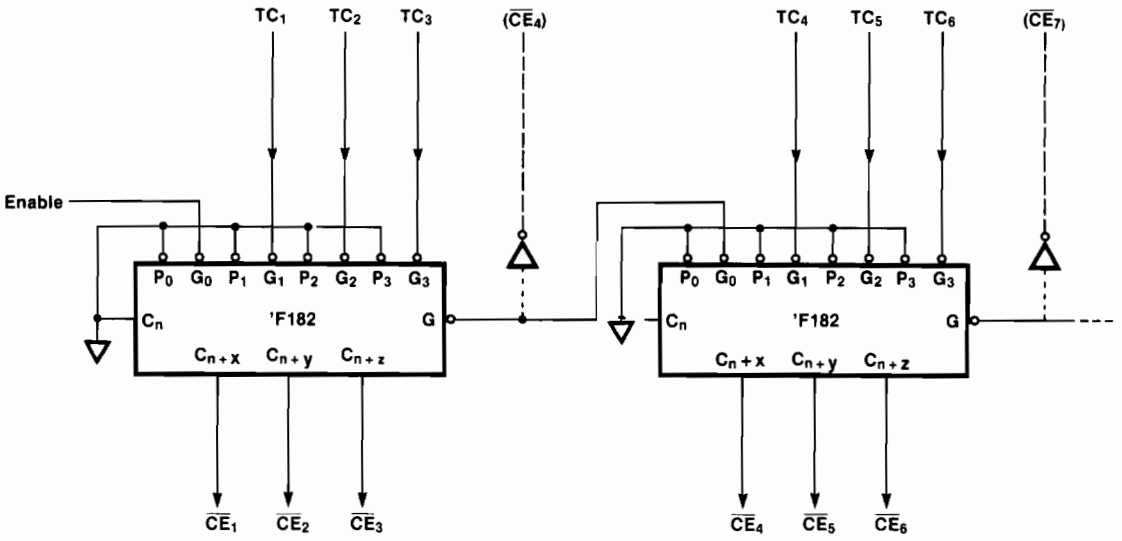
b. Synchronous n-Stage Counter Using Ripple Carry/Borrow

The configuration shown in Figure 7-31c avoids ripple delays and their associated restrictions. The $\overline{CE}$ input for a given stage is formed by combining the TC signals from all the preceding stages. Note that in order to inhibit counting an enable signal must be included in each carry gate. The simple inhibit scheme of Figures 7-31a and b does not apply, because the TC output of a given stage is not affected by its own $\overline{CE}$.

For extended counter strings the logic complexity grows rapidly if look ahead is required. It is possible to employ the 'F182 Carry Lookahead Generator to generate the carry/borrow signals in these cases retaining a low chip count while still retaining the speed advantage of look ahead as illustrated in Figure 7-31d.



c. Synchronous n-Stage Counter with Parallel Gated Carry/Borrow



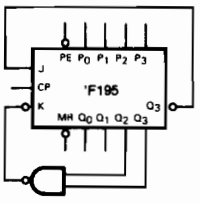
d. Using 'F182 Lookahead Carry Circuits to Generate Carry/Borrow Signals for Synchronous n-Stage Counter

Figure 7-31

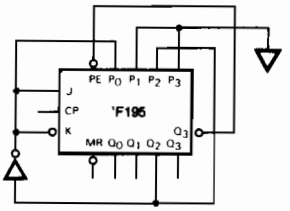
'F195 Shift Register as a Counter

Counting with Shift Registers

a. Modulo 8



b. Modulo 6



c. Modulo 6

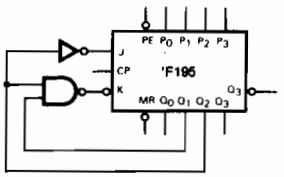


Figure 7-32

Table 7-12
Count Sequence

Q_0	Q_1	Q_2	Q_3
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1

Table 7-13
Count Sequence

Q_0	Q_1	Q_2	Q_3
0	0	0	1
1	0	0	0
1	1	0	0
1	1	1	0
0	1	1	1
0	0	1	0

Table 7-14
Count Sequence

Q_0	Q_1	Q_2	Q_3
0	0	0	1
1	0	0	0
1	1	0	0
1	1	1	0
0	1	1	1
0	0	1	1

The 'F195 4-bit universal shift register can be used for a wide variety of counting circuits including simple counters of different modulo, variable modulo counters and up/down counters. Twisted ring counters offer glitch-free decoding of any individual state with one inverter and one 2-input NAND gate. Decoding any group of adjacent states (2, 3, 4, 5, 6 or 7) is equally simple. The unused states of these counters are nonpersistent; i.e., the counter reverts into its operating loop if accidentally set to an unused state.

Twisted Ring (Johnson or Moebius) Reversible Counters

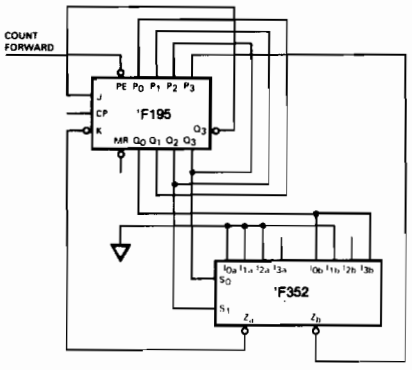


Table 7-15
Count Sequence

Q_0	Q_1	Q_2	Q_3
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1

a. Modulo 8

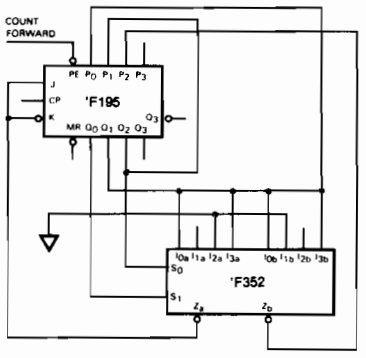


Table 7-16
Count Sequence

Q_0	Q_1	Q_2
0	0	0
1	0	0
1	1	0
1	1	1
0	1	1
0	0	1

b. Modulo 6

Figure 7-33

Twisted ring reversible counters are possible with 'F195 shift registers and 'F352 multiplexers. Individual or adjacent states are easily decoded without glitches with 2-input NAND gates and inverters. Again, all unused states are non-persistent.

Linear Feedback Shift Counters

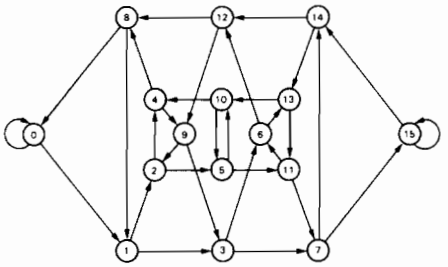


Figure 7-34

The most economical approach to the design of a counter with the 'F195 is the shift counter technique. The general state diagram of a 4-bit feedback shift register is shown. Each state is identified by a decimal number which is the equivalent of the contents of the register taken as a binary number. The first bit in the register is the least significant bit. Each state has two entrances and two exits. One exit results from shifting a zero into the register and the other from shifting in a one. A wide variety of loop sequences can be chosen from the state table by assigning one and zero exits from successive states. Count sequences can be selected based on decoding requirements, simplicity of feedback logic or other system constraints.

There are several feedback shift counter designs using the 'F195. One approach would be to select the desired sequence of states using this general state diagram. Then, the necessary input functions would be developed to produce the selected sequence. On the first attempt to generate the input function, those states not appearing in the desired sequence (except the all-0 and all-1 states) can be considered as "Immaterial" conditions. However, it is necessary to check the states considered as "Immaterial" conditions against the generated input function to see that they do not form any unwanted loops. The all-1 and all-0 states are special cases since they may be persistent states (loops of one state). If an all-1 or an all-0 state is not wanted, this requirement can be taken into consideration on the initial development of the input function by supplying a zero input for the all-1 state and a one for the all-0 state. If the unused states form a loop, it may be necessary to modify the input function to eliminate the secondary loop. Examination of the relationship between states in the desired and undesired loops indicates the required changes in the input function for eliminating the undesired loops. Except for the 2-state loop (1010, 0101), there are several distinct loops for all loop lengths. For example, there are sixteen different loop sequences of length 16. The implementation of the necessary input function is much simpler for some loops than for others.

Feedback Shift Registers

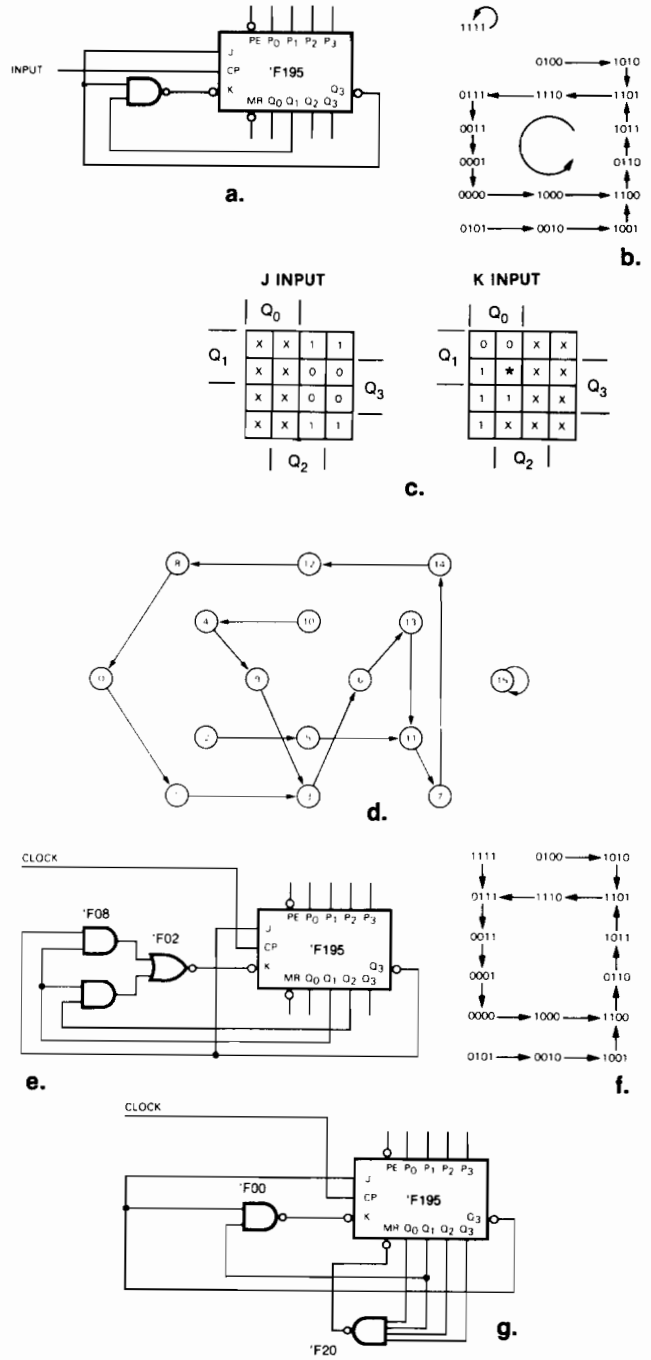


Figure 7-35

An example of this approach to the design of a feedback shift register is illustrated in Figure 7-35a. A 10-state loop is selected, and the Veitch diagrams for the J and $\overline{K}$ inputs are shown (Figure 7-35c) with the resulting state diagram for the simplest gate implementation (Figure 7-35d). With the simplest gate implementation of the main loop, the all-1 state is persistent. This condition can be relieved by several approaches.

A new input function and its gate implementation results in a zero input for all-1 state (Figure 7-35e).

Another method of eliminating the all-1 persistent state is to connect a 4-input gate as shown so that the all-1 condition produces a Master Reset input (Figure 7-35g). At least one stage is then set to zero.

Modulo 15 Counter

Table 7-17
Jump Will Produce Modulo 10 Counter

0	0	0	0
1	0	0	0
0	1	0	0
1	0	1	0
0	1	0	1
0	0	1	0
1	0	0	1
1	1	0	0
0	1	1	0
1	0	1	1
1	1	0	1
1	1	1	0
0	1	1	1
0	0	1	1
0	0	0	1

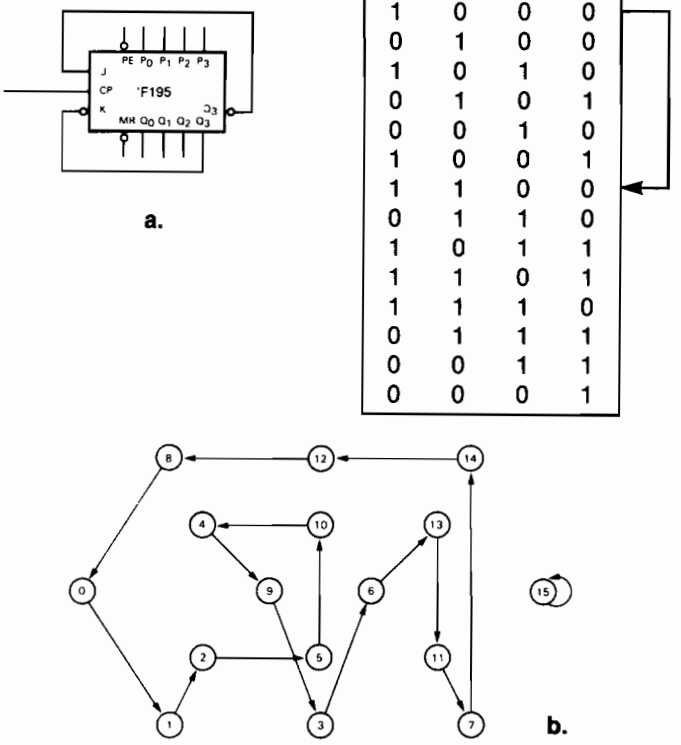


Figure 7-36

Another approach to the generation of a count sequence with a feedback shift register is to first generate a simple sequence such as one obtained by toggling the first stage of an n-bit shift register whenever the last stage is zero. This simple function produces a loop of 2^n-1 states and a single all-1 persistent state for the values of n = 2, 3, 4, 6, 7, 15 and 22. Figure 7-36 shows a 'F195 connected in this manner and the resulting sequence of states.

Examination of this sequence shows that a 10-state counter results if a jump is caused from the 1000 state to the 1100 state. It is only necessary to inhibit the Reset of the first state to obtain this jump. Further examination of the sequence reveals that the inhibit function is simply the Q_2 output. This approach creates exactly the same counter as obtained by the previous methods.

Up/Down Counters

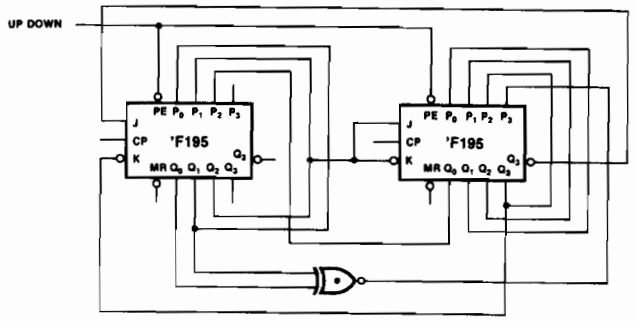
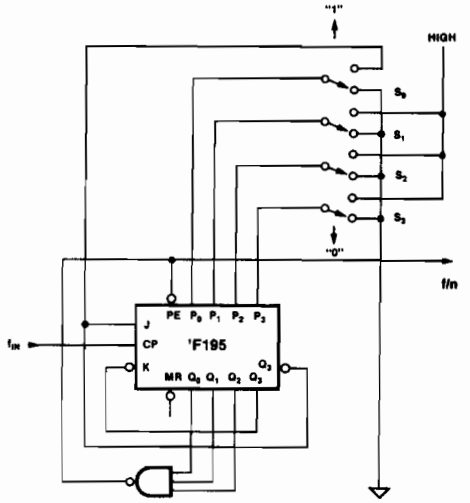


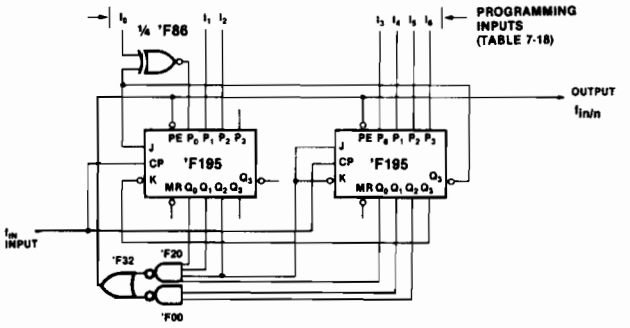
Figure 7-37

An up/down shift register counter is possible by connecting the 'F195 as a right/left shift register (using the synchronous parallel inputs) and supplying a zero to the left shift input when the first two stages are different. This is the reverse of toggling the first stage whenever the last stage is zero. This is a modulo 127 up/down counter using this technique.

Variable Modulo Counters



a.



b.

Figure 7-38

Another use for the parallel input of the 'F195 shift register is illustrated in Figure 7-38 by the variable modulo or divide-by-n counters. Again, the simple feedback (toggle the first stage when the last stage is zero) is used. This time a single gate produces the Parallel Load signal whenever the first three stages contain all ones. The parallel input combination to be loaded into the register is determined by the four switches. Note that this counter can divide by any integer up to and including 16. The divide-by-n counter shown in Figure 7-38b is simply an extension to seven bits of the 4-bit divide-by-n counter. Table 7-19 gives the loading values for this 7-bit counter.

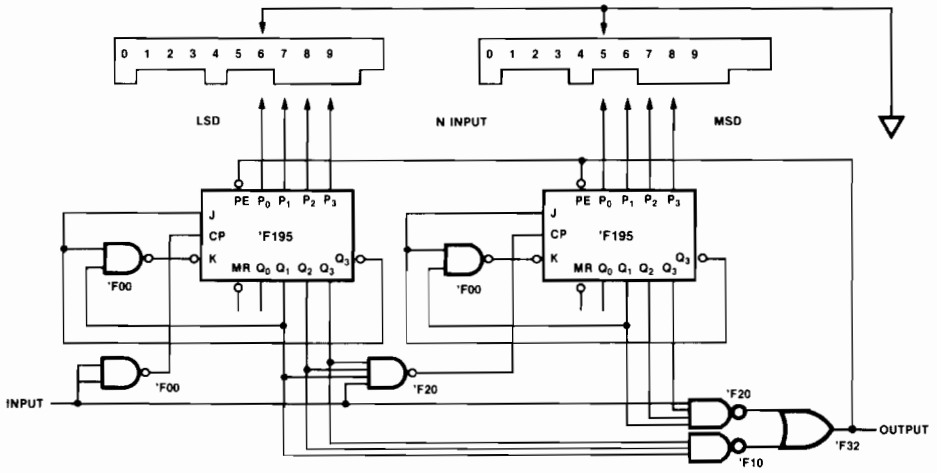
Table 7-18

S_0	S_1	S_2	S_3	N
1	1	1	1	16
0	1	1	1	15
0	0	1	1	14
0	0	0	1	13
0	0	0	0	12
1	0	0	0	11
0	1	0	0	10
1	0	1	0	9
0	1	0	1	8
0	0	1	0	7
1	0	0	1	6
1	1	0	0	5
0	1	1	0	4
1	0	1	1	3
1	1	0	1	2
1	1	1	0	1

Table 7-19
Loading Values, Divide-by-in Counters

n	I_0	I_1	I_2	I_3	I_4	I_5	I_6
128	1	1	1	1	1	1	1
127	0	1	1	1	1	1	1
126	0	0	1	1	1	1	1
125	0	0	0	1	1	1	1
124	0	0	0	0	1	1	1
123	0	0	0	0	0	1	1
122	0	0	0	0	0	0	1
121	0	0	0	0	0	0	0
120	1	0	0	0	0	0	0
119	0	1	0	0	0	0	0
118	1	0	1	0	0	0	0
117	0	1	0	1	0	0	0
116	1	0	1	0	1	0	0
115	0	1	0	1	0	1	0
114	1	0	1	0	1	0	1
113	1	1	0	1	0	1	0
112	0	1	1	0	1	0	1
111	0	0	1	1	0	1	0
110	1	0	0	1	1	0	1
109	1	1	0	0	1	1	0
108	0	1	1	0	0	1	1
107	0	0	1	1	0	0	1
106	0	0	0	1	1	0	0
105	1	0	0	0	1	1	0
104	0	1	0	0	0	1	1
103	0	0	1	0	0	0	1
102	0	0	0	1	0	0	0
101	1	0	0	0	1	0	0
100	0	1	0	0	0	1	0
99	1	0	1	0	0	0	1
98	1	1	0	1	0	0	0
97	0	1	1	0	1	0	0
96	1	0	1	1	0	1	0
95	0	1	0	1	1	0	1
94	0	0	1	0	1	1	0
93	1	0	0	1	0	1	1
92	1	1	0	0	1	0	1
91	1	1	1	0	0	1	0
90	0	1	1	1	0	0	1
89	0	0	1	1	1	0	0
88	1	0	0	1	1	1	0
87	0	1	0	0	1	1	1
86	0	0	1	0	0	1	1
85	0	0	0	1	0	0	1
84	0	0	0	0	1	0	0
83	1	0	0	0	0	1	0
82	0	1	0	0	0	0	1
81	0	0	1	0	0	0	0
80	1	0	0	1	0	0	0
79	0	1	0	0	1	0	0
78	1	0	1	0	0	1	0
77	0	1	0	1	0	0	1
76	0	0	1	0	1	0	0
75	1	0	0	1	0	1	0
74	0	1	0	0	1	0	1
73	0	0	1	0	0	1	0
72	1	0	0	1	0	0	1
71	1	1	0	0	1	0	0
70	0	1	1	0	0	1	0
69	1	0	1	1	0	0	1
68	1	1	0	1	1	0	0
67	0	1	1	0	1	1	0
66	1	0	1	1	0	1	1
65	1	1	0	1	1	0	1
64	1	1	1	0	1	1	0
63	0	1	1	1	0	1	1
62	0	0	1	1	1	0	1
61	0	0	0	1	1	1	0
60	1	0	0	0	1	1	1
59	1	1	0	0	0	1	1
58	1	1	1	0	0	0	1
57	1	1	1	1	0	0	0
56	0	1	1	1	1	0	0
55	1	0	1	1	1	1	0
54	0	1	0	1	1	1	1
53	0	0	1	0	1	1	1
52	0	0	0	1	0	1	1
51	0	0	0	0	1	0	1
50	0	0	0	0	0	1	0
49	1	0	0	0	0	0	1
48	1	1	0	0	0	0	0
47	0	1	1	0	0	0	0
46	1	0	1	1	0	0	0
45	0	1	0	1	1	0	0
44	1	0	1	0	1	1	0
43	0	1	0	1	0	1	1
42	0	0	1	0	1	0	1
41	0	0	0	1	0	1	0
40	1	0	0	0	1	0	1
39	1	1	0	0	0	1	0
38	0	1	1	0	0	0	1
37	0	0	1	1	0	0	0
36	1	0	0	1	1	0	0
35	0	1	0	0	1	1	0
34	1	0	1	0	0	1	1
33	1	1	0	1	0	0	1
32	1	1	1	0	1	0	0
31	0	1	1	1	0	1	0
30	1	0	1	1	1	0	1
29	1	1	0	1	1	1	0
28	0	1	1	0	1	1	1
27	0	0	1	1	0	1	1
26	0	0	0	1	1	0	1
25	0	0	0	0	1	1	0
24	1	0	0	0	0	1	1
23	1	1	0	0	0	0	1
22	1	1	1	0	0	0	0
21	0	1	1	1	0	0	0
20	1	0	1	1	1	0	0
19	0	1	0	1	1	1	0
18	1	0	1	0	1	1	1
17	1	1	0	1	0	1	1
16	1	1	1	0	1	0	1
15	1	1	1	1	0	1	0
14	0	1	1	1	1	0	1
13	0	0	1	1	1	1	0
12	1	0	0	1	1	1	1
11	1	1	0	0	1	1	1
10	1	1	1	0	0	1	1
9	1	1	1	1	0	0	1
8	1	1	1	1	1	0	0
7	0	1	1	1	1	1	0
6	1	0	1	1	1	1	1
5	1	1	0	1	1	1	1
4	1	1	1	0	1	1	1
3	1	1	1	1	0	1	1
2	1	1	1	1	1	0	1

Multistage Program Divider



This circuit divides any number n from 1 to 100. The selected n is one greater than is shown on the slide switches. As an example, the switches show 56, therefore the circuit will divided by 57.

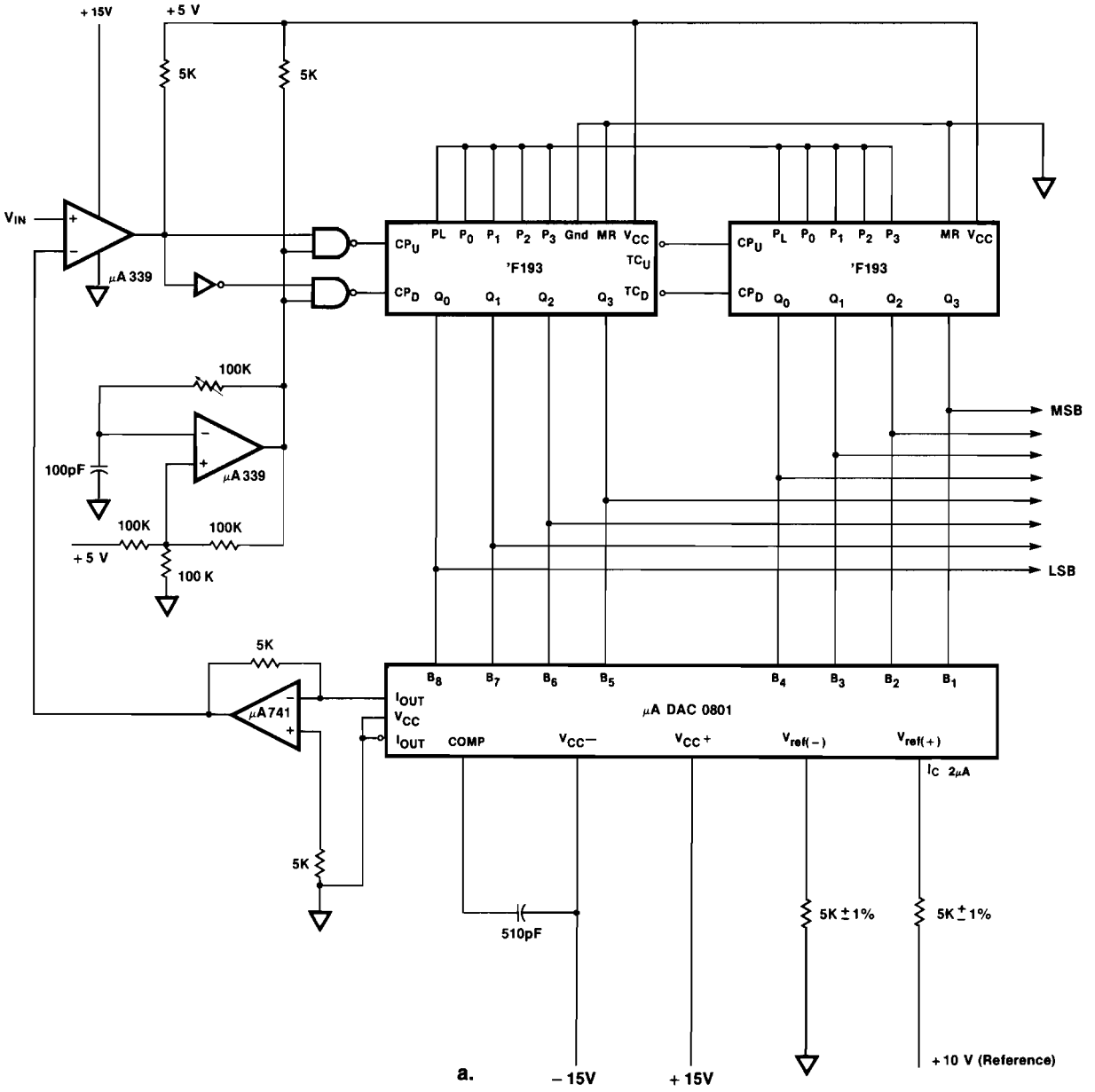
Figure 7-39

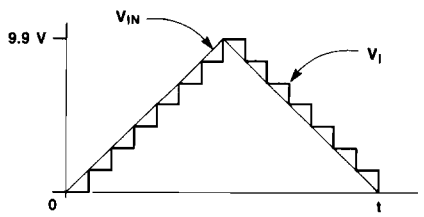
Table 7-20 Counting States

9	0011
8	0001
7	0000
6	1000
5	1100
4	0110
3	1011
2	1101
1	1110
0	0111

Divide-by-n counters are difficult to use in a large number of program divider applications due to the unwieldy nature of the n-input format when large values of a variable input are required. This can be overcome by building the counter in stages. This diagram is a simple, fully synchronous 2-stage decade counter. Each stage counts down modulo 10, with the first stage gating the clock to the second stage when it is in the zero condition. When both stages are in the zero condition, the Parallel Load is enabled, setting the value of n into the two decade stages. Since the decade count 00 is included in the sequence, the counter counts one more than the value of n loaded. Each additional decade stage using this approach requires either much more logic or a reduction of operating speed.

Servo A/D Converter





$0 \leq V_{IN} \geq 10$ V

b.

Figure 7-40

This circuit constitutes essentially a form of phase detection with subsequent error correction. Two μA 339 comparators are used here, one to transform the input analog voltage to digital and the other to produce the proper clock frequency.

For the servo A/D converter to operate properly, a relationship between the analog signal frequency and the digital clock frequency must be maintained. A 100K Ω potentiometer in the clock feedback circuit is provided for trimming the clock frequency. This relationship must exist for the μA 0801 phase detector to correctly follow the analog input signal information.

Then:

$$\frac{E/m}{2^n - 1} = E/m \sin \omega t$$

$$E_{max} = 10V \text{ for DACO8 with } n = 8 \text{ bits}$$

$$\frac{10V}{2^8 - 1} = 10V \sin 2 \pi f_A \Delta t$$

For small angles we can use the relation: $\sin 2 \pi f_A \Delta t \cong 2 \pi f_A \Delta t$

Then:

$$\frac{1}{255} = 2 \pi f_A \Delta t$$

From here:

$$\Delta t = \frac{1}{255 \cdot 2 \pi f_A}$$

But to keep up with the input analog frequency, the digital clock:

$$f_{clock} = \frac{1}{\Delta t}$$

Then:

$$f_{\substack{clock \\ digital}} = 255 \times 2 \times \pi \cdot f_A$$

$$f_{\substack{clock \\ digital}} = 510 \times \pi \times f_A$$

In the circuit, as a difference in phase occurs between the input and the clock signals, the output of the DAC will automatically adjust the feedback voltage level on the input comparator. This is accomplished through a μA 741 operation amplifier utilized here as a current to voltage converter circuit. The count inputs to the 'F193 are thereby toggled alternately by one until phase error correction is realized.

Digital Sine Wave Generator

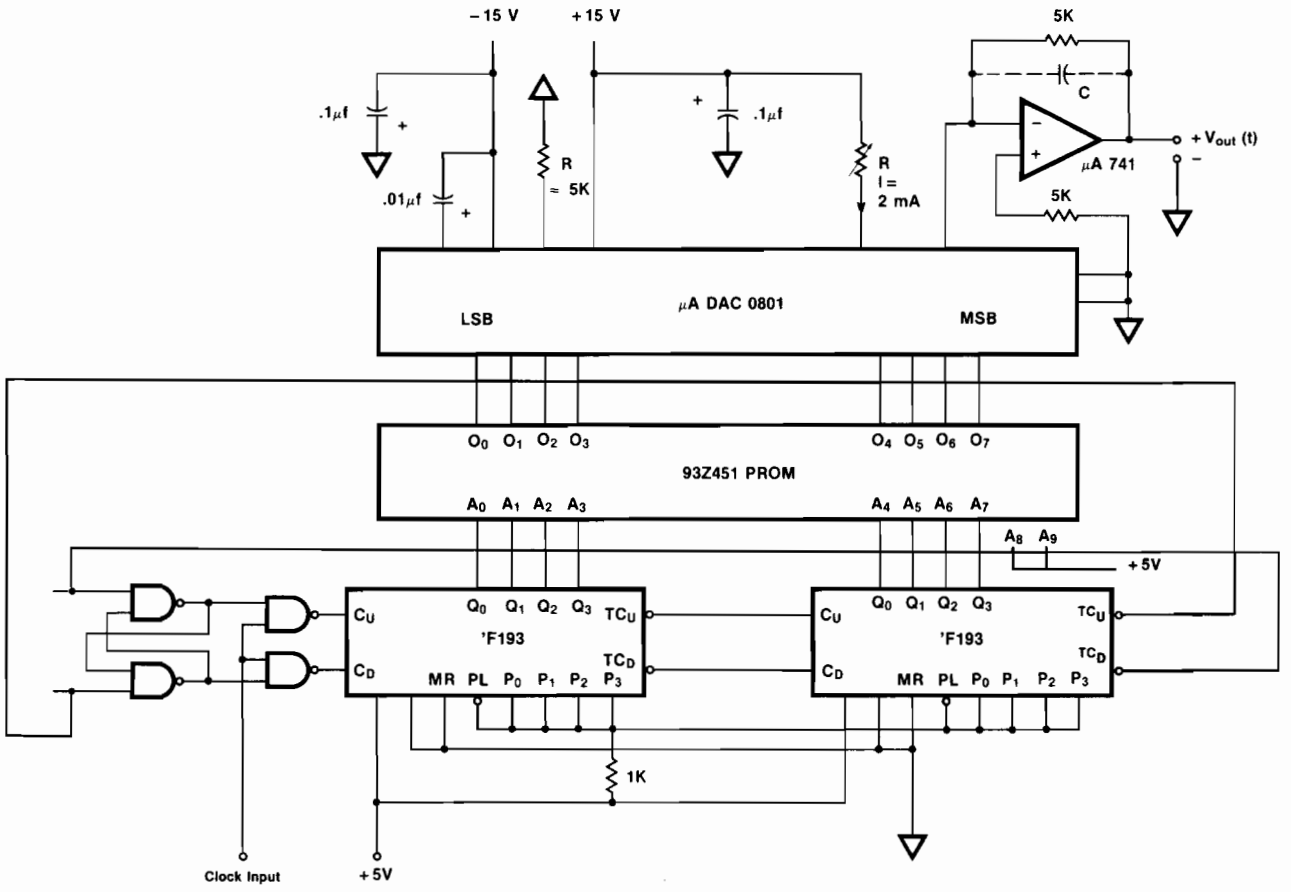


Figure 7-41

In this circuit we have utilized two 'F193 up/down binary counters to generate digital information which is converted to analog by the Fairchild μA 0801 DAC. Also included is the 93Z451 PROM which is programmed for values taken at 360/511 or 0.7° intervals allowing proper sine wave generation. A total of 180 degrees were computed using the following formulas:

0-90 degrees => Sin (N)° + (128) = value

-90-0 degrees => 128 - value = value'

where N = increments of 0.7°

Table 7-21 showing hex values for 0°-180° has been included in order to facilitate PROM programming.

A μA 741 operational amplifier, connected as a voltage follower, is provided at the output of the μA 0801 in order to transform an otherwise high output impedance to a low value, thus improving interface drive capability. By the addition of C (in dashed lines), a LOW pass filter may be used. This can be useful in many cases depending on clock frequencies.

As can be seen on the circuit schematic all unused inputs must be tied to the appropriate logic levels. This will ensure proper operation of the 'F193 counters.

The maximum frequency of the generator is determined by the μA 0801 DAC settling time, and is approximately 11MHz/512 or 21.5KHz.

Table 7-21 Digital Sine Wave Generation PROM Programming Chart

Degree of Angle	Hex Equiv.	Degree of Angle	Hex Equiv.	Degree of Angle	Hex Equiv.	Degree of Angle	Hex Equiv.	Degree of Angle	Hex Equiv.
−90.0	0	−52.0	1B	−14.1	61	23.9	B3	61.9	F0
−89.3	0	−51.3	1C	−13.4	63	24.6	B4	62.6	F0
−88.6	0	−50.6	1D	−12.7	64	25.3	B5	63.3	F1
−87.9	0	−49.9	1E	−12.0	66	26.0	B7	64.0	F2
−87.2	0	−49.2	1F	−11.3	67	26.7	B8	64.7	F3
−86.5	0	−48.5	20	−10.6	69	27.4	BA	65.4	F3
−85.8	0	−47.8	21	−9.9	6A	28.1	BB	66.1	F4
−85.1	01	−47.1	23	−9.2	6C	28.8	BC	66.8	F4
−84.4	01	−46.4	24	−8.5	6D	29.5	BE	67.5	F5
−83.7	01	−45.7	25	−7.7	6F	30.2	BF	68.2	F6
−83.0	01	−45.0	26	−7.0	70	30.9	C1	68.9	F6
−82.3	01	−44.3	27	−6.3	72	31.6	C2	69.6	F7
−81.6	02	−43.6	28	−5.6	74	32.3	C3	70.3	F7
−80.9	02	−42.9	29	−4.9	75	33.0	C5	71.0	F8
−80.2	02	−42.2	2A	−4.2	77	33.7	C6	71.7	F8
−79.5	02	−41.5	2C	−3.5	78	34.4	C7	72.4	F9
−78.8	03	−40.8	2D	−2.8	7A	35.2	C8	73.1	F9
−78.0	03	−40.1	2E	−2.1	7B	35.9	CA	73.8	FA
−77.3	03	−39.4	2F	−1.4	7D	36.6	CB	74.5	FA
−76.6	04	−38.7	30	−0.7	7E	37.3	CC	75.2	FB
−75.9	04	−38.0	32	0	80	38.0	CD	75.9	FB
−75.2	04	−37.3	33	0.7	81	38.7	CF	76.6	FB
−74.5	05	−36.6	34	1.4	82	39.4	D0	77.3	FC
−73.8	05	−35.9	35	2.1	84	40.1	D1	78.0	FC
−73.1	06	−35.2	37	2.8	85	40.8	D2	78.7	FC
−72.4	07	−34.5	38	3.5	87	41.5	D3	79.4	FD
−71.7	07	−33.8	39	4.2	88	42.2	D5	80.1	FD
−71.0	08	−33.1	3A	4.9	8A	42.9	D6	80.8	FD
−70.3	08	−32.4	3C	5.6	8B	43.6	D7	81.5	FD
−69.6	08	−31.7	3D	6.3	8D	44.3	D8	82.3	FE
−68.9	09	−30.9	3E	7.0	8F	45.0	D9	83.0	FE
−68.2	09	−30.2	40	7.7	90	45.7	DA	83.7	FE
−67.5	0A	−29.5	41	8.4	92	46.4	DB	84.4	FE
−66.8	0B	−28.8	43	9.1	93	47.1	DC	85.1	FE
−66.1	0B	−28.1	44	9.8	95	47.8	DE	85.8	FF
−65.4	0C	−27.4	45	10.5	96	48.5	DF	86.5	FF
−64.7	0D	−26.7	47	11.2	98	49.2	E0	87.2	FF
−64.0	0D	−26.0	48	11.9	99	49.9	E1	87.9	FF
−63.3	0E	−25.3	4A	12.6	9B	50.6	E2	88.6	FF
−62.6	0F	−24.6	4B	13.4	9C	51.3	E3	89.3	FF
−61.9	0F	−23.9	4C	14.1	9E	52.0	E4	90.0	FF
−61.2	10	−23.2	4E	14.8	9F	52.7	E5		
−60.5	11	−22.5	4F	15.5	A1	53.4	E5		
−59.8	12	−21.8	51	16.2	A2	54.1	E6		
−59.1	13	−21.1	52	16.9	A4	54.8	E7		
−58.4	13	−20.4	54	17.6	A5	55.5	E8		
−57.7	14	−19.7	55	18.3	A7	56.2	E9		
−57.0	15	−19.0	57	19.0	A8	56.9	EA		
−56.3	16	−18.3	58	19.7	AA	57.6	EB		
−55.6	17	−17.6	5A	20.4	AB	58.3	EC		
−54.9	18	−16.9	5B	21.1	AD	59.0	EC		
−54.1	19	−16.2	5D	21.8	AE	59.8	ED		
−53.4	1A	−15.5	5E	22.5	B0	60.5	EE		
−52.7	1A	−14.8	60	23.2	B1	61.2	EF		

16-Phase Clock Generator

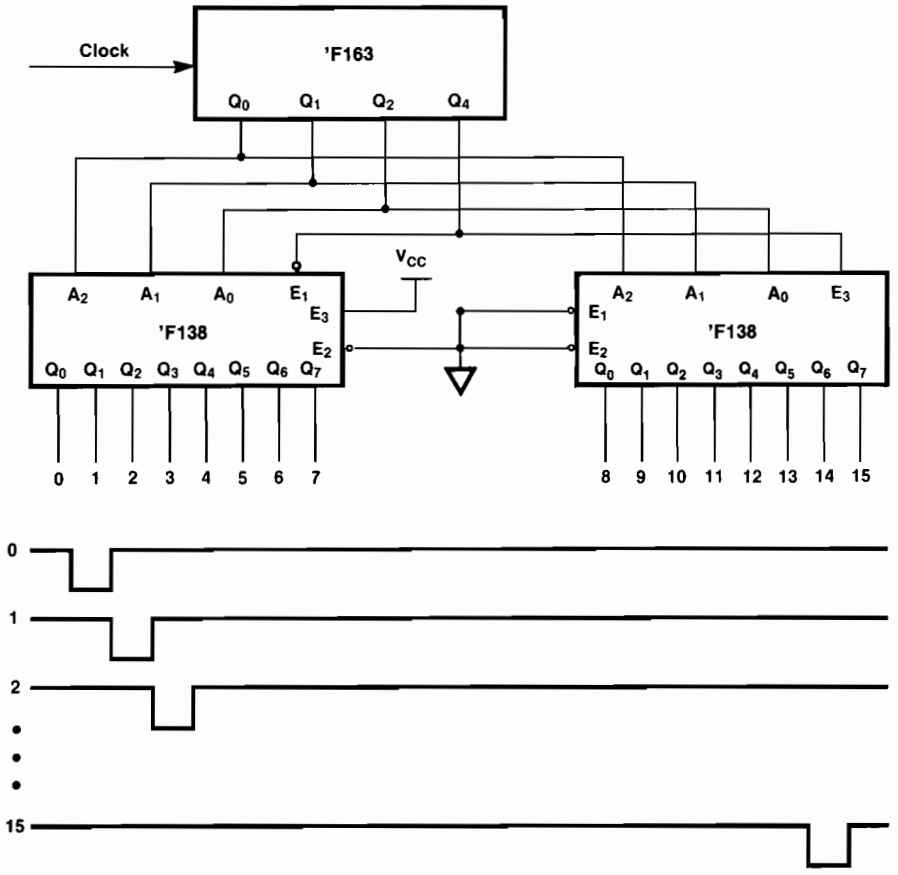


Figure 7-42

Figure 7-42 illustrates a circuit which is a 1-of-16 non-overlapping clock generator using one 'F163 and two 'F138 circuits. The 'F163 counter outputs are fed in parallel to two 'F138 circuits. Opposite enables are utilized on the 'F138s so that during count one through seven only one decoder is enabled while during count eight through sixteen the second 'F138 is enabled.

Ten Decade Counter

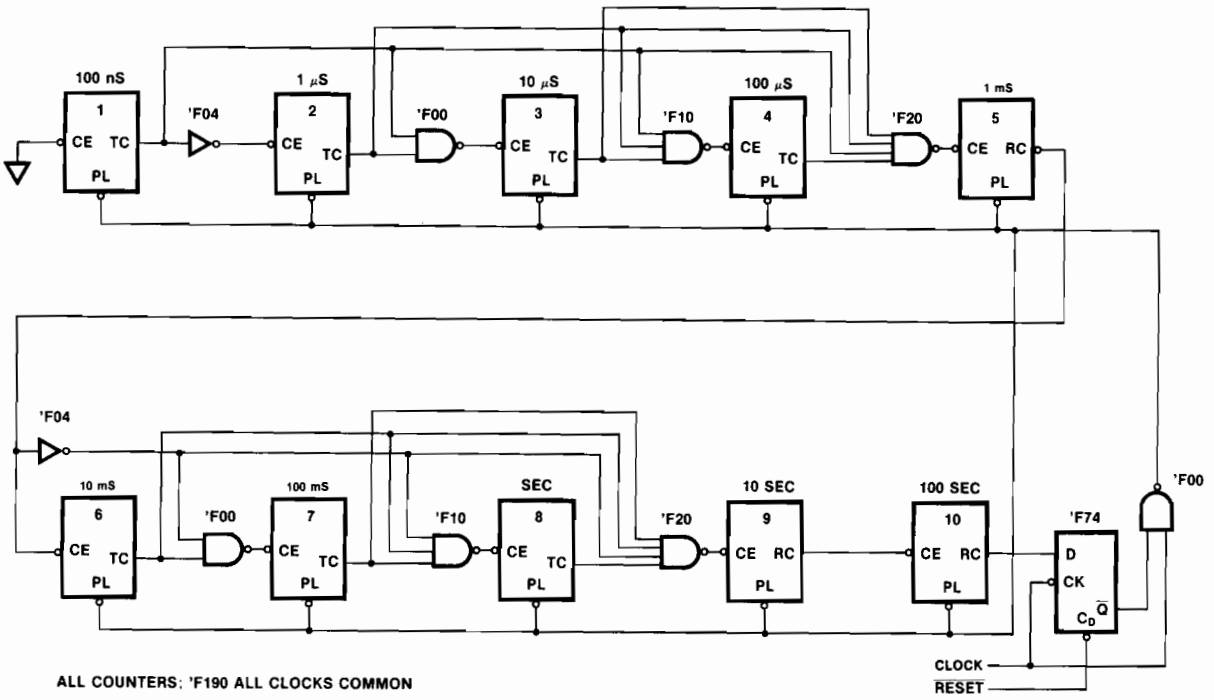


Figure 7-43

The circuit in Figure 7-43 is applicable where precise time intervals are necessary. The parallel inputs may be connected to BCD switch banks and will enable the user to predetermine the cycle time of the circuit. The parallel input data is loaded when reset is initiated or final terminal count is active and the clock goes positive. All counters are 'F190s operating in the down count mode and the clock is fed in parallel to all counters. A 10MHz clock is assumed to be in use in order to facilitate the time division shown, although up to 12.5MHz may be used. Maximum clock rate is established by computing the cumulative propagation delay as follows:

Critical timings:

Device Number			
1	CP to TC	= 14ns	Clock HIGH = 20ns
	Gate Delay	= 6ns	
5	$\overline{CE}$ to $\overline{RC}$	= 8ns	Clock LOW = 36ns
	Gate Delay	= 6ns	
	Gate Delay	= 6ns	
9	$\overline{CE}$ to $\overline{RC}$	= 8ns	
10	$\overline{CE}$ to $\overline{RC}$	= 8ns	

The terminal count (TC) output of Counter 1 is normally LOW, but switches HIGH when the circuit reaches zero in the count down mode. TC will remain HIGH enabling its subsequent gate until the next positive transition of the clock. TC thus enables counter 2 for one count in every 10. This pattern is the case throughout the system.

The $\overline{RC}$ output is utilized on counter 5 in order to facilitate a reduction in gate packages used. Otherwise, an 8-input NAND gate would be necessary in order to enable counter 9. The switch selected data, present on the parallel inputs, is loaded into the counters upon reset and a positive clock transition or upon completion of the previous cycle. When the count is completed $\overline{RC}$ is set true on counter 10. Here $\overline{RC}$ causes an 'F74, connected as a one-shot, to re-initialize the the cycle.

The Chip Enable of Counter 1 can be used as an Enable to the counter. If $\overline{CE}$ on counter 1 is made HIGH all counting will halt, and a LOW causes counting to continue from where it was halted.

With a 10MHz clock and all nines present on the parallel load inputs in the countdown mode a delay of 16.666 minutes is available, with resolution of 100ns.

1024 x 1024 Non-Interlaced Video Board Using FAST and the 7220

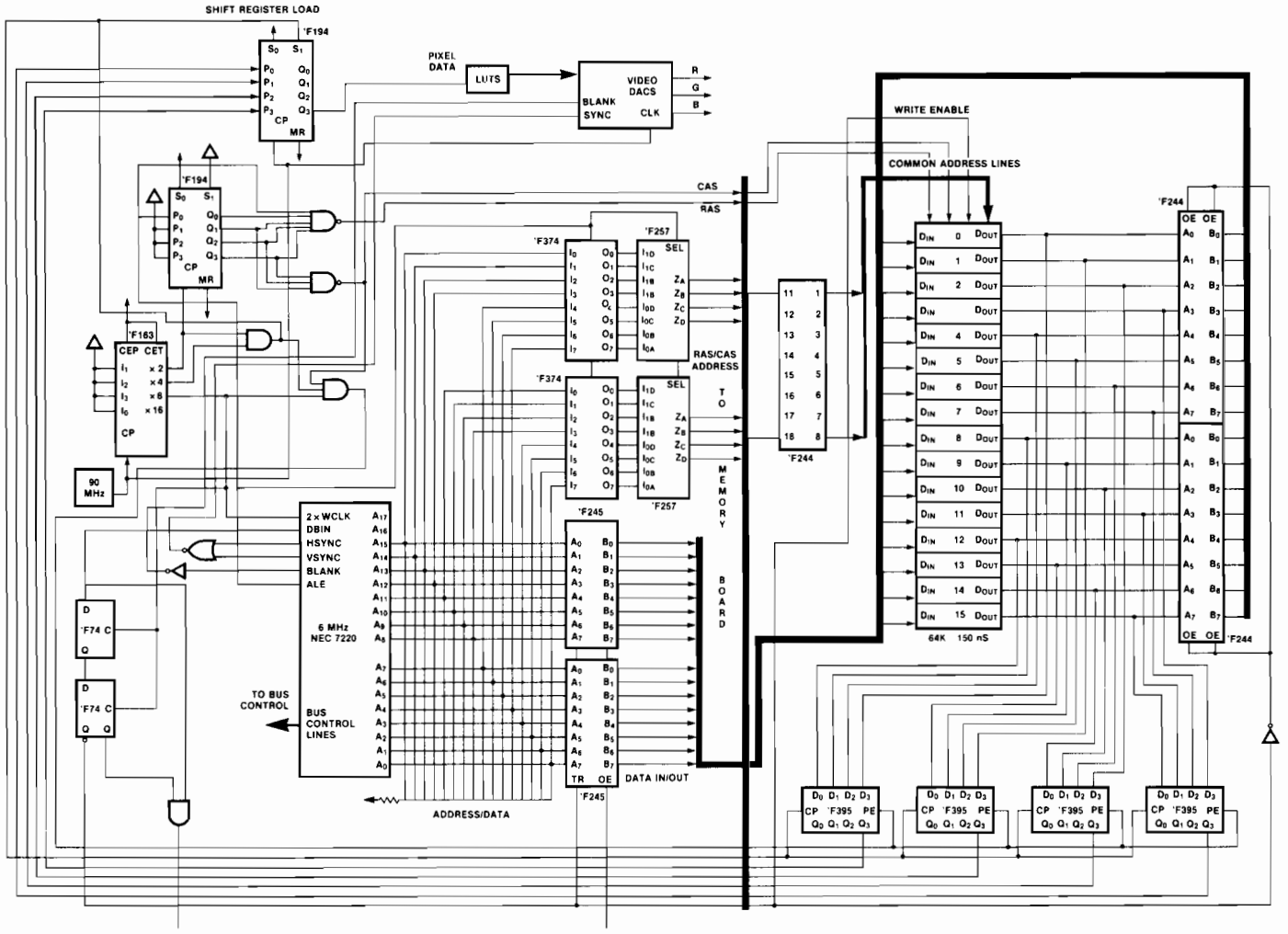
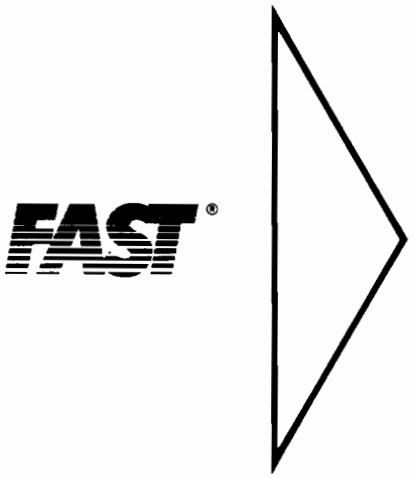


Figure 7-44

The accompanying drawing is a conceptual description of a cost effective, non-interlaced 1024 x 1024 video graphics subsystem using FAST technology. FAST is utilized to attain the minimum pixel speed of 90MHz necessary for an RS-343A compatible monitor. Also used in the design is the 7220 graphics display controller for monitor control signals and frame buffer modification.

This particular implementation is single plane, though there are a number of ways to expand this for multi-plane operation. As previously mentioned, the 90MHz clock rate makes it imperative to use FAST technology. The 90MHz clock generator is used as the system clock.



TTL Small Scale Integration

Introduction

SSI circuits (gates and flip-flops) are useful for extending logic functions available in MSI. They should be used sparingly, essentially as "glue" between MSI circuits, because they require considerably more PC board area, interconnections, power and volume compared to equivalent functions available in MSI circuits.

Gates

FAST offers a wide variety of gate circuits: NAND, NOR, EXCLUSIVE-(N)OR, AND/OR/INVERT, BUFFERS. The functions in this section are described in terms of positive logic. Obviously, a positive NAND gate can be considered a negative NOR gate and a positive NOR can be interpreted as a negative NAND. The normal use of TTL gates is well understood and need not be covered here. The following notes describe miscellaneous uncommon circuits.

Increasing the Voltage Swing of TTL Outputs

When interfacing with MOS devices or discrete transistors, it may be desirable to increase the logic swing out of a TTL output, which (worst case) is only 2.0V. A 1kΩ resistor connected from any TTL output to V_{CC} will pull the output close to V_{CC}, increasing the output swing to more than 4V, while decreasing the available fanout by three unit loads.

Driving Transistors from FAST Gate Outputs

Any gate output can be tied to the base of a grounded emitter transistor via a resistor to provide base drive. An additional pull up resistor tied to V_{CC} increases the base drive.

Edge Detector 'F00

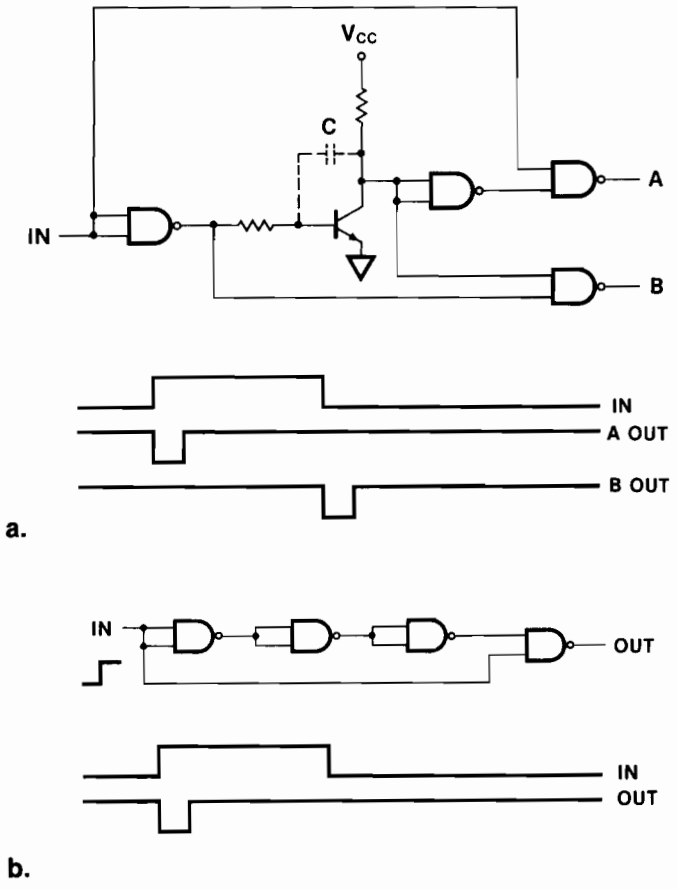


Figure 8-1

The edge detector circuit shown in Figure 8-1a generates a negative-going pulse on output A for each LOW-to-HIGH transition of the input, and generates a negative-going pulse on output B for each HIGH-to-LOW transition of the input. The pulse width is adjustable by varying the Miller capacitance. The transistor can be replaced by an additional NAND gate (Figure 8-1b). In this case, the pulse widths will be determined by the propagation delay of the gates. A single edge detection circuit can also be built using NAND gates. This circuit will generate a negative-going pulse on the LOW-to-HIGH transition of the input. The pulse width can be adjusted by changing the number of inverters. The formula for the pulse width is $(2n + 1)\ t_{pd}$, where t_{pd} is the propagation delay of each inverter. If the circuit is built with NOR gates, the pulse is generated by the HIGH-to-LOW transition of the input.

By using non-inverting delay chains, this circuit can be made to skew the rising or falling edge of an input signal. This is useful in applications where a longer or shorter pulse is needed.

Dual Edge Detector

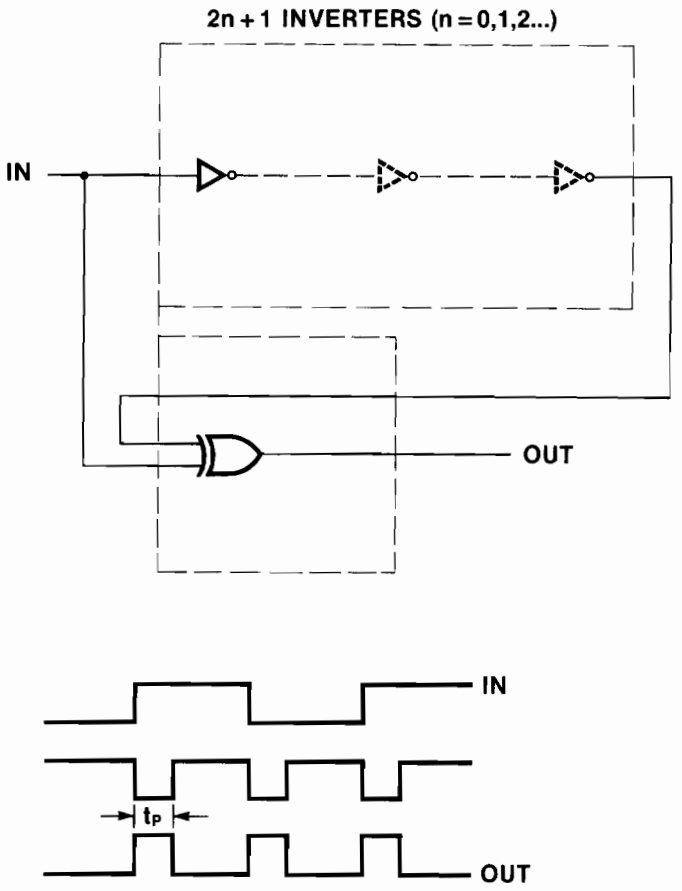


Figure 8-2

One-quarter of an 'F86 quad Exclusive-OR gate with an odd number of inverters provides a circuit generating an output pulse for both a LOW-to-HIGH and a HIGH-to-LOW transition of the input signal. The remaining three Exclusive-OR gates may be used as inverters. This function is useful for regenerating the clock in a self-clocking PDM transmission system. When fed with a square wave input, this circuit acts as a frequency doubler. The pulse width is determined by the number of inverter stages.

Simple and Inexpensive Pulse Phase Demodulator

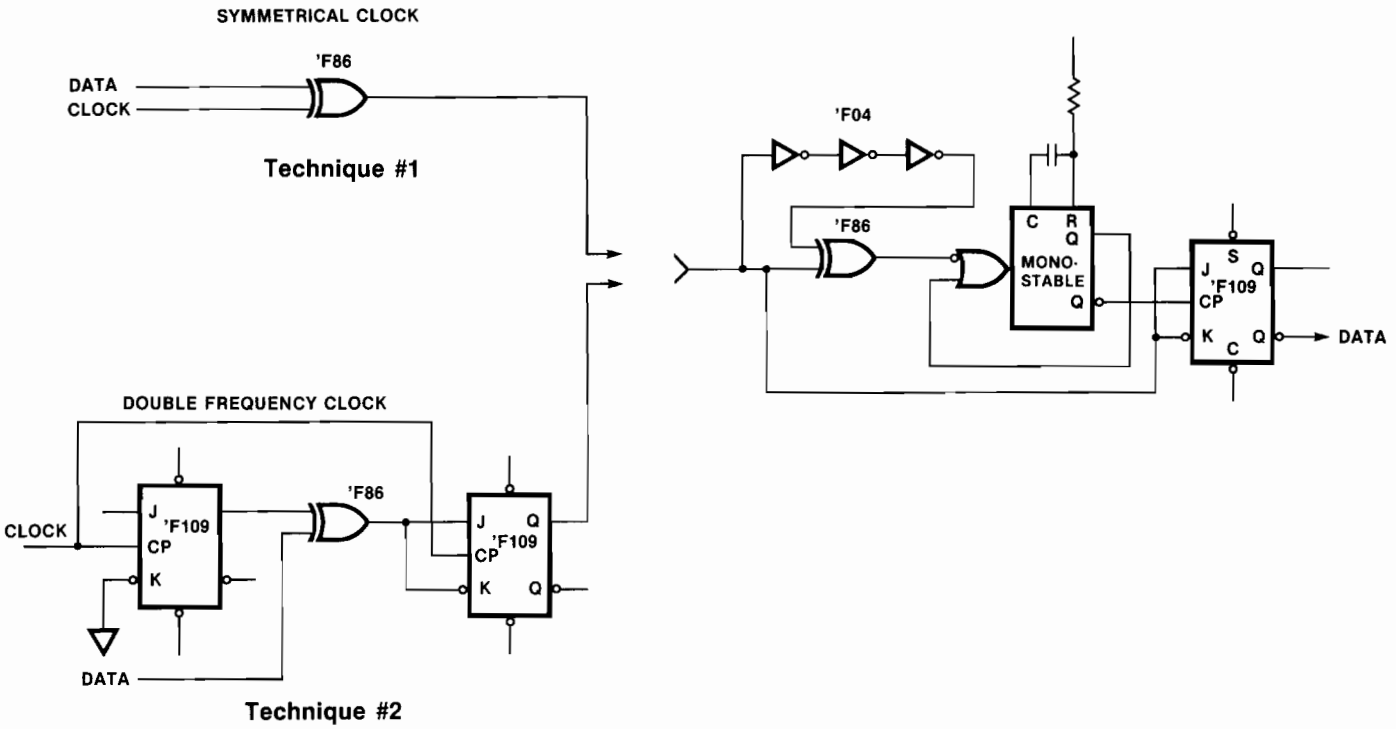


Figure 8-3

Many digital systems require data communication over a single line without access to a common clock. This means that clock pulses and data must be transmitted on the same wire. A common, efficient way to accomplish such transmission is with pulse phase modulation (Manchester code).

In such a code, a zero is represented by a signal level that is HIGH for the first half of the bit time, LOW for the second half. A one is represented by a level that is first LOW, then HIGH. Thus, each bit has at least one transition which is used to resynchronize data at the receiver. Generating this code is very simple, requiring either an Exclusive-OR (Technique #1) and a 50% duty cycle clock, or an Exclusive-OR and a flip-flop triggered by a double frequency clock (Technique #2). Another flip-flop can be used to resynchronize and suppress glitches.

The receiver is considerably more complex, because it must discriminate between the important clock transitions that occur in the middle of every bit time and the incidental transitions that occur between two adjacent opposite bits. A nonretriggerable one-shot with a pulse width set at 75% of a bit time (or anything between 50% and 100%), and triggered by any input level change, performs this discrimination.

Classically, the trigger pulses are generated by a differentiator and a rectifying circuit, involving diodes, capacitors and/or a pulse transformer. This function can be performed most economically by one-half of a 'F04 and one-quarter of a 'F86. The 'F04 inverts and delays the incoming signal. The 'F86 forms the Exclusive-OR of this delayed, inverted signal and the input signal. The output of the 'F86 is a negative pulse with a width that is determined by the delay of the three inverters for either a positive or negative transition of the input signal. The one-shot connected in a non-retriggerable mode triggers only on the clock transition (in the middle of the bit). The trailing edge of the one-shot can be used as a clock input to an edge-triggered flip-flop which recovers the transmitted data. This receiver is synchronized by the first input data change and stays synchronized as long as the one-shot pulse width is between 50% and 100% of the data bit time.

This circuit is very cost effective and practical for data rates up to approximately 10 megabits per second. At lower frequencies, it can be made self-adaptive to varying data rates by controlling the one-shot time, or, better yet, by substituting an oscillator and a programmable counter for the one-shot.

Simple RC Clock Generator

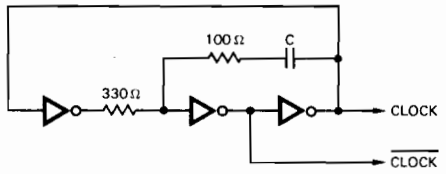


Figure 8-4

Table 8-1

C	f
200 pF	5 MHz
1600 pF	1 MHz
0.018 μF	100 KHz
0.18 μF	10 KHz

The simple TTL clock generator circuit shown provides a clock satisfactory for most simple TTL systems and it always starts oscillating without coaxing. This circuit requires only one-half of a hex inverter package and three passive components—two resistors and a capacitor.

Active LOW Enable

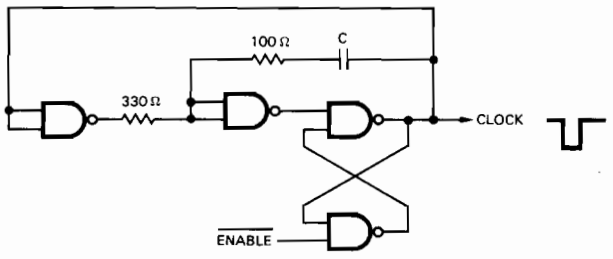


Figure 8-5

An active LOW Enable input can be provided by replacing the 'F04 with a 'F00 quad 2-input NAND gate as shown above. The cross-coupled gates guarantee that clock pulses are not cut short if the Enable input is removed at the wrong time. Once the clock goes LOW, it stays LOW for its full normal width even if the Enable signal is returned HIGH. The clock pulse starts two gate delays (8ns) after the HIGH-to-LOW transition of the $\overline{\text{ENABLE}}$ input.

Single and Dual Flip-Flops

FAST offers a wide variety of single and dual flip-flops with various differences in logic configuration, speed and power.

There are three independent aspects to the selection of flip-flops:

- Logic configuration
- Speed and power
- Clocking scheme

Logic Configuration

JK flip-flops' response to any digital input condition is logically defined according to the truth table below.

Table 8-2

J	K	Q_{n+1}
Not Active	Not Active	Q_n (= No Change)
Active	Not Active	HIGH (= SET)
Not Active	Active	LOW (= RESET)
Active	Active	$\overline{Q}_n$ (= TOGGLE)

Essentially J is a Set input and K is a Reset input, but applying both J and K simultaneously toggles the flip-flop. The flip-flops differ in the logic controlling the J and K condition. Some have no extra logic, others have a JK Enable, some have a multi-input AND gate controlling J and K, and some have an AND/OR structure, accepting data from different sources. Note that some of the inputs are active LOW. The $\overline{\text{K}}$ input is particularly useful since tying J and $\overline{\text{K}}$ together results in a D input.

All present TTL flip-flops have an asynchronous, active LOW Preset or Clear input, and some of them have both.

Clocking Scheme

Perhaps the most important and certainly the most confusing and misunderstood aspect of flip-flops is their clocking scheme. This defines when the flip-flop accepts data and when the outputs change with regard to the clock.

While most MSI circuits follow standard and simple rules (data is accepted a set-up time before the rising edge of the clock, and the outputs change after the rising edge of the clock), flip-flops are not that consistent.

There is a functional difference between edge-triggered and true master/slave flip-flops. In addition, some flip-flops use inverting clock buffers and others do not. As a result, there are four different clocking schemes. Only two are used with FAST devices.

Rising-Edge Triggered

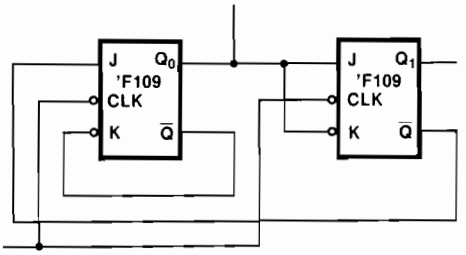
The 'F74 dual flip-flop accepts data a set-up time before the rising edge of the clock (⎍) and changes its output after this rising edge of the clock. Clock pulse width is not critical as long as it exceeds worst case set-up time. This is consistent with the behavior of most MSI devices.

Falling-Edge Triggered

The 'F112/'F113/'F114 dual flip-flops accept data a set-up time before the falling edge of the clock and change their outputs after this falling edge of the clock. Their clock input is therefore shown with an inverting symbol (⎍). Interfacing with most MSI devices requires a clock inverter unless this offset triggering is desired.

Johnson (Moebius) Counters

Modulo 3



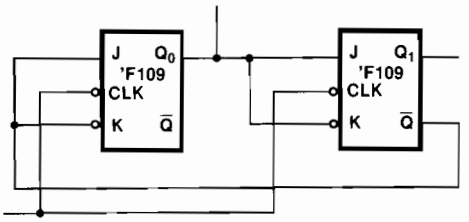
Modulo 3 Count Sequence

Q_0	Q_1
0	0
1	0
0	1
1	1

Figure 8-6a

With this modulo 3 Johnson counter, each state can be easily decoded with a 2-input NAND gate. As shown, the unused state is non-persistent.

Modulo 4



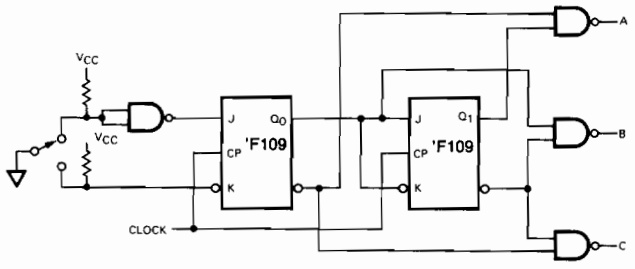
Modulo 4 Count Sequence

Q_0	Q_1
0	0
1	0
1	1
0	1

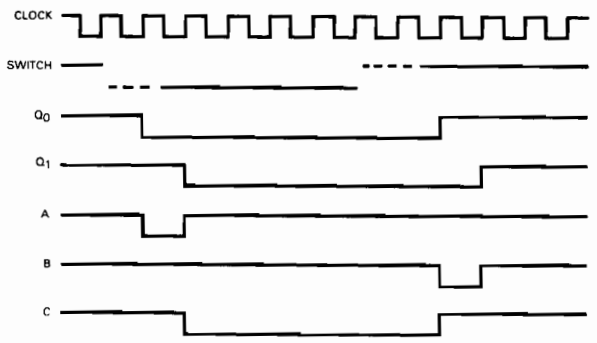
Figure 8-6b

The modulo 4 counter also provides for glitch-free, easy decoding of each state with a 2-input NAND gate.

Switch Bounce Eliminator and Digital Differentiator



a.



b.

Figure 8-7

This circuit eliminates switch bounce and generates active LOW output pulses; one on output A after switch depression and one on output B after switch release, and a bounce-free level output C between them.

Exclusive Latch

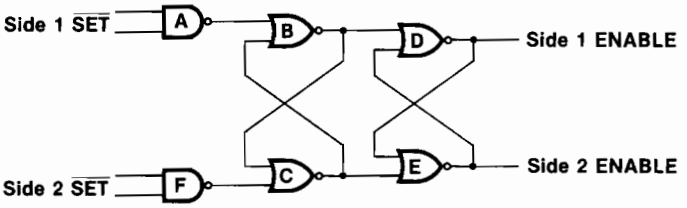


Figure 8-8

The exclusive latch is an SSI implementation of logic necessary for bus arbitration. The output will only change state with a single active input; two active inputs cause the latch to hold current state.

Inductance is Not All Bad

Trace inductance can be used to advantage. Figure 8-9 shows trace inductance used in a lumped constant delay line in a Z80A memory timing application.

For a Z80A running at 4MHz, two areas of critical timing exist when using 64k DRAMs. These are fetch cycle (MI), read access time, and $\overline{RAS}$ precharge during T3 of the opcode fetch cycle. In most production designs, minimum cost is a major design objective along with easy and consistent assembly and test. Maximum speed is the next consideration. A worst case timing analysis on the Z80A shows 240ns maximum access is available during MI (MRQ to t_s D_{in}) and 97.5ns available for $\overline{RAS}$ precharge. If we assume that the CPU and memory are located on different boards, bus drivers, buffers and backplane/PCB trace delays will account for 61ns, thus leaving 179ns available for RAM access. This is well within the 150ns access DRAM timing. However, the $\overline{RAS}$ precharge at 97.5ns is outside the 100ns time required by 150ns DRAMs. By shortening the $\overline{RAS}$ signal we can achieve a precharge of greater than 100ns; $\overline{RAS}$ can be shortened to 144ns minimum (T_hRow + T Col Valid + T_{cas} + 100ns). The simplest method of achieving this is to clock a flip-flop from the leading edge of T3. This will then give a $\overline{RAS}$ precharge time of 160ns maximum and $\overline{RAS}$ of 297ns min and a total cycle time of 457ns. For stability, the timing section should be implemented using a lumped constant LC delay line. This can be implemented as shown in Figure 8-9.

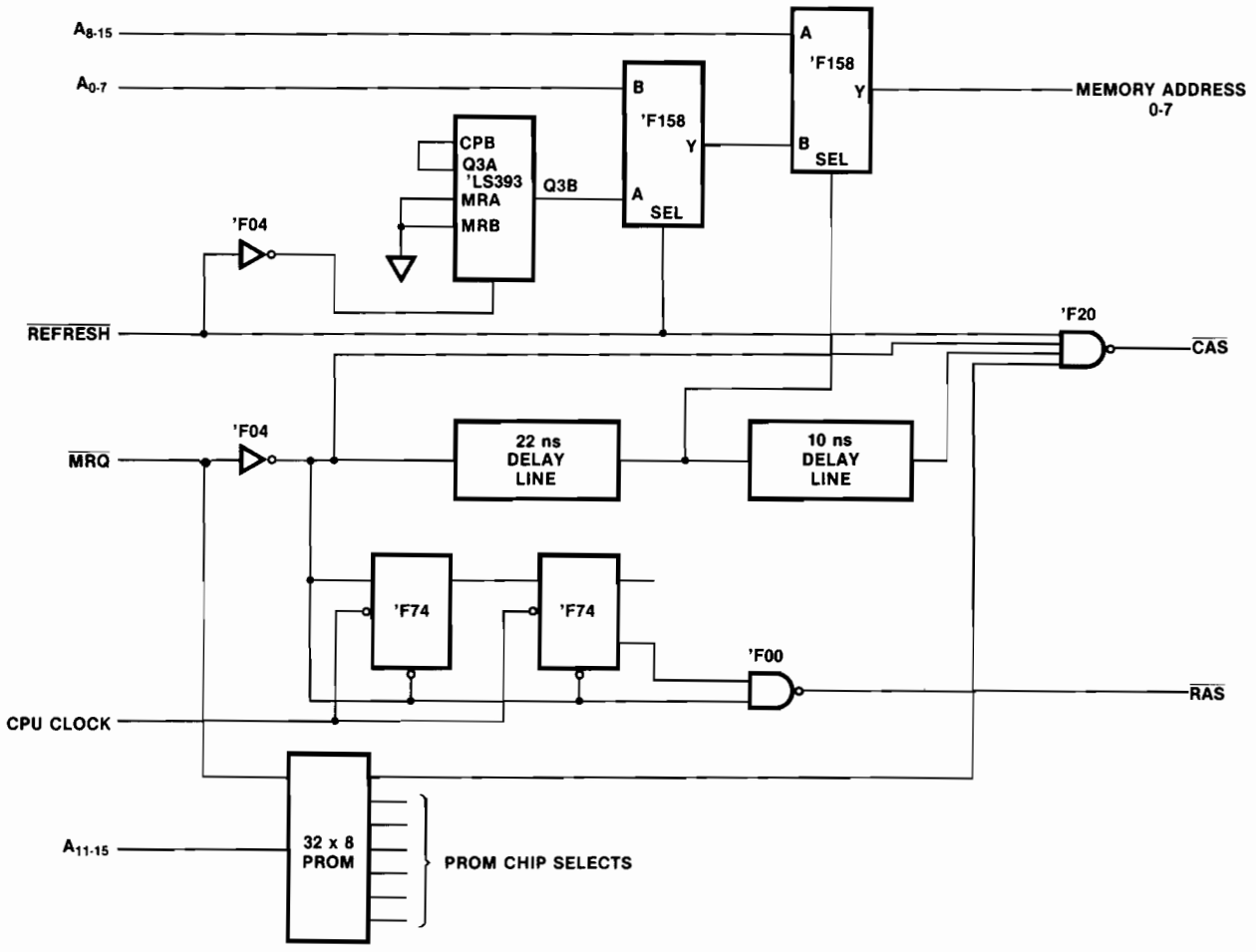


Figure 8-9 Trace Inductance in Memory Timing Application

The timing circuit is comprised of two LC lumped constant delay lines of 22ns and 10ns respectively. The first of these (22ns) delay lines can be made from five sections of 250nH loaded with 82pF. The 250nH inductors are constructed from eight turn spirals of 0.0125" track on a 0.025" grid. The line delay can be calculated from

$$T_S = \sqrt{LC}$$

The dynamic impedance $Z_L = \sqrt{L/C}$

The trace inductance

$$L = .005l \left[l_n(2l/w + t) + 1/2 \right] \ \mu H$$

Where l = trace length, w = trace width, t = trace thickness, all inches.

If a multilayer board is used, then the inductance calculation will change due to the close proximity to ground.

$$L = \left[0.0051 \ \ln \left(\frac{4h}{\alpha} \right) \right] + .00127 \ \ \mu H/inch$$

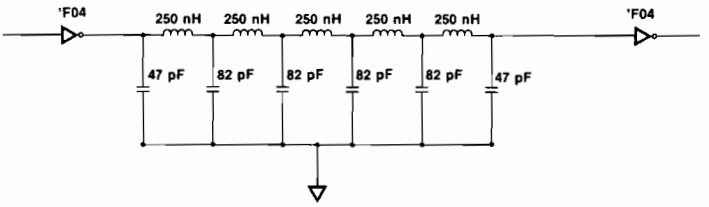
$$where \ \alpha = \sqrt{\frac{4\ wt}{\pi}}$$

h = *dielectric thickness*

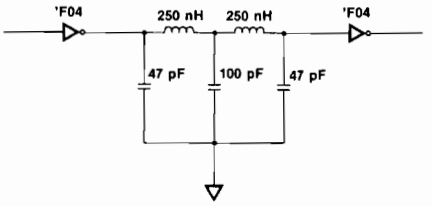
A 250nH inductor is constructed from 7.0 inches of PC trace. When loaded with 82pF this has an impedance of 54Ω. A FAST 'F04 can drive a 50Ω load and switch on incident wave, thus making production design both easy to control and reliable. The temperature coefficient of the timing circuit can be reduced to a very small number by selecting a capacitor to have a negative temperature coefficient, to track the parameter shifts of the FAST gates or to track the timing requirements of the dynamic RAMs used. The printed circuit inductor delay line will occupy more area than a proprietary delay line but is much lower in cost and easier to control or change. Capacitors are also multi-sourced where manufacturers of delay lines are few and their products expensive.

Figure 8-10

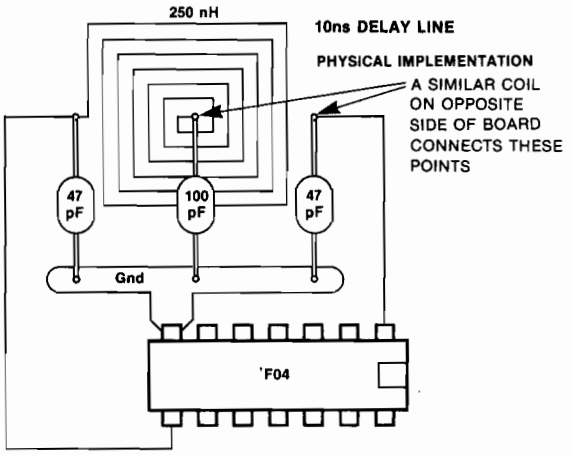
a. 22ns Delay Line

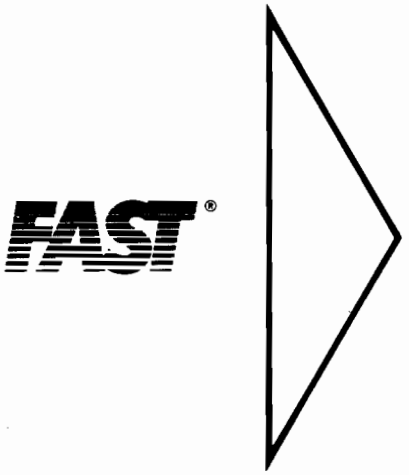


b. 10ns Delay Line



c. 10ns Delay Line





FAIRCHILD
A Schlumberger Company

Line Driving and System Design

Introduction

Successful high-speed system design is dependent on careful system timing design and good board layout. The pitfalls are many and varied, and this section addresses some of those problem areas and simplifies the design requirements. All systems must interconnect signals either by short lines on printed circuit board, long lines on a backplane, twisted pair cables, or coaxial cables, etc. At high frequency, all of these mediums must be treated as transmission lines. Two properties of transmission lines, characteristic impedance (Z_O) and propagation delay (t_{PD}), are of concern. Transmission lines store energy, the magnitude of which is dependent on line length, impedance, applied voltage and source impedance. This stored energy must be dissipated by the terminating device and is also available to be coupled in other circuits by crosstalk. The effects of termination on line reflection and crosstalk are discussed, as well as good board layout practices.

9

Transmission Line Concepts

The interactions between wiring and circuitry in high-speed systems are more easily determined by treating the interconnections as transmission lines. A brief review of basic concepts is presented and simplified methods of analysis are used to examine situations commonly encountered in digital systems. Since the principles and methods apply to any type of logic circuit, normalized pulse amplitudes are used in sample waveforms and calculations.

Simplifying Assumptions

For the great majority of interconnections in digital systems, the resistance of the conductors is much less than the input and output resistance of the circuits. Similarly, the insulating materials have very good dielectric properties. These circumstances allow such factors as attenuation, phase distortion and bandwidth limitations to be ignored. With these simplifications, interconnections can be dealt with in terms of characteristic impedance and propagation delay.

Characteristic Impedance

The two conductors that interconnect a pair of circuits have distributed series inductance and distributed capacitance between them, and thus constitute a transmission line. For any length in which these distributed parameters are constant, the pair of conductors have a characteristic impedance Z_O. Whereas quiescent conditions on the line are determined by the circuits and terminations, Z_O is the ratio of transient voltage to transient current passing by a point on the line when a signal change or other electrical disturbance occurs. The relationship between transient voltage, transient current, characteristic impedance, and the distributed parameters is expressed as follows:

$$\frac{V}{I} = Z_O = \sqrt{\frac{L_O}{C_O}}$$

(E9-1)

where L_O = inductance per unit length, and C_O = capacitance per unit length. Z_O is in ohms, L_O in henries, and C_O in farads.

Propagation Velocity

Propagation velocity (ν) and its reciprocal, delay per unit length (δ), can also be expressed in terms of L_O and C_O. A consistent set of units is nanoseconds, microhenries and picofarads, with a common unit of length.

$$\nu = \frac{1}{\sqrt{L_O C_O}} \qquad \delta = \sqrt{L_O C_O}$$

(E9-2)

Equations 9-1 and 9-2 provide a convenient means of determining the L_O and C_O of a line when delay, length and impedance are known. For a length l and delay T, δ is the ratio T/l. To determine L_O and C_O, combine Equations 9-1 and 9-2.

(E9-3) $L_O = \delta Z_O$

(E9-4) $C_O = \frac{\delta}{Z_O}$

More formal treatments of transmission line characteristics, including loss effects, are available from many sources.

Termination and Reflection

A transmission line with a terminating resistor is shown in Figure 9-1. As indicated, a positive step function voltage travels from left to right. To keep track of reflection polarities, it is convenient to consider the lower conductor as the voltage reference and to think in terms of current flow in the top conductor only. The generator is assumed to have zero internal impedance. The initial current I_1 is determined by V_1 and Z_O.

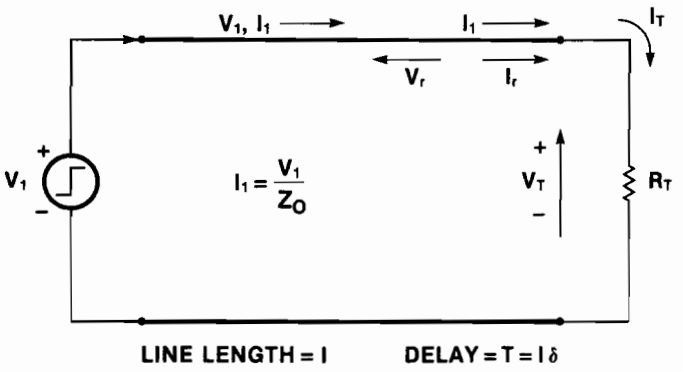


Figure 9-1

If the terminating resistor matches the line impedance, the ratio of voltage to current traveling along the line is matched by the ratio of voltage to current which must, by Ohm's law, always prevail at R_T. From the viewpoint of the voltage step generator, no adjustment of output current is ever required; the situation is as though the transmission line never existed and R_T had been connected directly across the terminals of the generator.

From the R_T viewpoint, the only thing the line did was delay the arrival of the voltage step by the amount of time T.

When R_T is not equal to Z_O, the initial current starting down the line is still determined by V_1 and Z_O but the final steady state current, after all reflections have died out, is determined by V_1 and R_T (ohmic resistance of the line is assumed to be negligible). The ratio of voltage to current in the initial wave is not equal to the ratio of voltage to current demanded by R_T. Therefore, at the instant the initial wave arrives at R_T, another voltage and current wave must be generated so that Ohm's law is satisfied at the line-load interface. This reflected wave, indicated by V_r and I_r in Figure 9-1, starts to return toward the generator. Applying Kirchoff's laws to the end of the line at the instant the initial wave arrives results in the following:

$$I_1 + I_r = I_T = \textit{current into } R_T$$

(E9-5)

Since only one voltage can exist at the end of the line at this instant of time, the following is true:

$$V_1 + V_r = V_T$$

$$\text{thus} \quad I_T = \frac{V_T}{R_T} = \frac{V_1 + V_r}{R_T}$$

$$\text{also} \quad I_1 = \frac{V_1}{Z_O} \text{ and } I_r = -\frac{V_r}{Z_O}$$

(E9-6)

with the minus sign indicating that V_r is moving toward the generator.

Combining the foregoing relationships algebraically and solving for V_r yields a simplified expression in terms of V_1, Z_O and R_T.

$$\frac{V_1}{Z_O} - \frac{V_r}{Z_O} = \frac{V_1 + V_r}{R_T} = \frac{V_1}{R_T} + \frac{V_r}{R_T}$$

(E9-7)

$$V_1 \left(\frac{1}{Z_O} - \frac{1}{R_T}\right) = V_r \left(\frac{1}{R_T} + \frac{1}{Z_O}\right)$$

$$V_r = V_1 \left(\frac{R_T - Z_O}{R_T + Z_O}\right) = \rho_L V_1$$

The term in parentheses is called the coefficient of reflection (ρ_L). With R_T ranging between zero (shorted line) and infinity (open line), the coefficient ranges between -1 and $+1$ respectively. The subscript L indicates that ρ_L refers to the coefficient at the load end of the line.

Equation 9-7 expresses the amount of voltage sent back down the line, and since

$$V_T = V_1 + V_r$$

$$\text{then} \quad V_T = V_1 (1 + \rho_L)$$

(E9-8)

V_T can also be determined from an expression which does not require the preliminary step of calculating ρ_L. Manipulating $(1 + \rho_L)$ results in

$$1 + \rho_L = 1 + \frac{R_T - Z_O}{R_T + Z_O} = 2\left(\frac{R_T}{R_T + Z_O}\right)$$

Substituting in Equation 9-8 gives

$$V_T = 2\left(\frac{R_T}{R_T + Z_O}\right) V_1$$

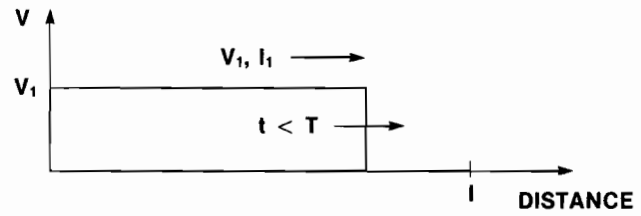
(E9-9)

The foregoing has the same form as a simple voltage divider involving a generator V_1 with internal impedance Z_O driving a load R_T, except that the amplitude of V_T is doubled.

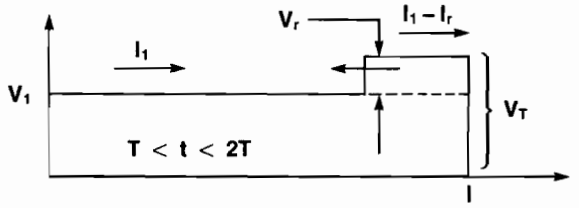
The arrow indicating the direction of V_r in Figure 9-1 correctly indicates the V_r direction of travel, but the direction of I_r flow depends on the V_r polarity. If V_r is positive, I_r flows toward the generator, opposing I_1. This relationship between the polarity of V_r and the direction of I_r can be deduced by noting in Equation 9-7 that if V_r is positive it is because R_T is greater than Z_O. In turn, this means that the initial current I_r is larger than the final quiescent current, dictated by V_1 and R_T. Hence I_r must oppose I_1 to reduce the line current to the final quiescent value. Similar reasoning shows that if V_r is negative, I_r flows in the same direction as I_1.

It is sometimes easier to determine the effect of V_r on line conditions by thinking of it as an independent voltage generator in series with R_T. With this concept, the direction of I_r is immediately apparent; its magnitude, however, is the ratio of V_r to Z_O, i.e., R_T is already accounted for in the magnitude of V_r. The relationships between incident and reflected signals are represented in Figure 9-2 for both cases of mismatch between R_T and Z_O.

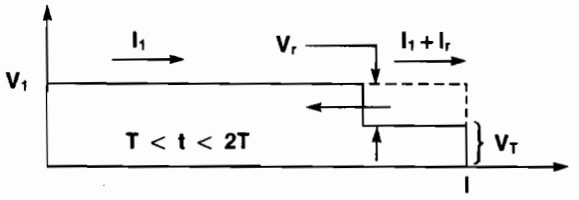
Figure 9-2 Reflections for $R_T \neq Z_O$



a. Incident Wave



b. Reflected Wave for $R_T > Z_O$



c. Reflected Wave for $R_T > Z_O$

The incident wave is shown in Figure 9-2a, before it has reached the end of the line. In Figure 9-2b, a positive V_r is returning to the generator. To the left of V_r the current is still I_1, flowing to the right, while to the right of V_r the net current in the line is the difference between I_1 and I_r. In Figure 9-2c, the reflection coefficient is negative, producing a negative V_r. This, in turn, causes an increase in the amount of current flowing to the right behind the V_r wave.

Source Impedance, Multiple Reflections

When a reflected voltage arrives back at the source (generator), the reflection coefficient at the source determines the response to V_r. The coefficient of reflection at the source is governed by Z_O and the source resistance R_S.

$$\rho_S = \frac{R_S - Z_O}{R_S + Z_O}$$

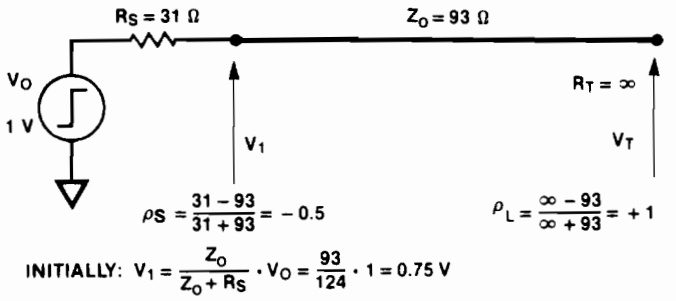
(E9-10)

If the source impedance matches the line impedance, a reflected voltage arriving at the source is not reflected back toward the load end. Voltage and current on the line are stable with the following values.

$$V_T = V_1 + V_r \text{ and } I_T = I_1 - I_r$$

(E9-11)

If neither source impedance nor terminating impedance matches Z_O, multiple reflections occur; the voltage at each end of the line comes closer to the final steady state value with each succeeding reflection. An example of a line mismatched on both ends is shown in Figure 9-3. The source is a step function of 1V amplitude occurring at time t_0. The initial value of V_1 starting down the line is 0.75V due to the voltage divider action of Z_O and R_S. The time scale in the photograph shows that the line delay is approximately 6ns. Since neither end of the line is terminated in its characteristic impedance, multiple reflections occur.



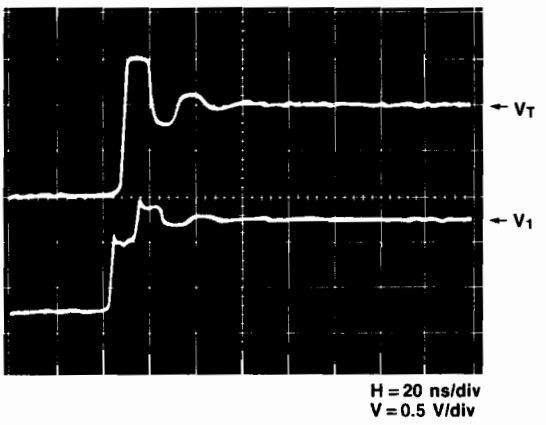


Figure 9-3 Multiple Reflections Due to Mismatch at Load and Source

The amplitude and persistence of the ringing shown in Figure 9-3 become greater with increasing mismatch between the line impedance and source and load impedances. Reducing R_S (Figure 9-3) to 13Ω increases ρ_S to -0.75, and the effects are illustrated in Figure 9-4. The initial value of V_T is 1.8V with a reflection of 0.9V from the open end. When this reflection reaches the source, a reflection of $(0.9)\times(-0.75V)$ starts back toward the open end. Thus, the second increment of voltage arriving at the open end is negative-going. In turn, a negative-going reflection of $(0.9)\times(-0.75V)$ starts back toward the source. This negative increment is again multiplied by $-0.75V$ at the source and returned toward the open end. It can be deduced that the difference in amplitude between the first two positive peaks observed at the open end is

$$V_T - V'_T = (1 + \rho_L)V_1 - (1 + \rho_L)\, V_1\, \rho^2{}_L\, \rho^2{}_S$$

$$= (1 + \rho_L)V_1\, (1 - \rho^2{}_L\, \rho^2{}_S).$$

(E9-12)

The factor $(1-\rho^2{}_L\rho^2{}_S)$ is similar to the damping factor associated with lumped constant circuitry. It expresses the attenuation of successive positive or negative peaks of ringing.

Lattice Diagram

In the presence of multiple reflections, keeping track of the incremental waves on the line and the net voltage at the ends becomes a bookkeeping chore. A convenient and systematic method of indicating the conditions which combines magnitude, polarity and time utilizes a graphic construction called a lattice diagram. A lattice diagram for the line conditions of Figure 9-3 is shown in Figure 9-5.

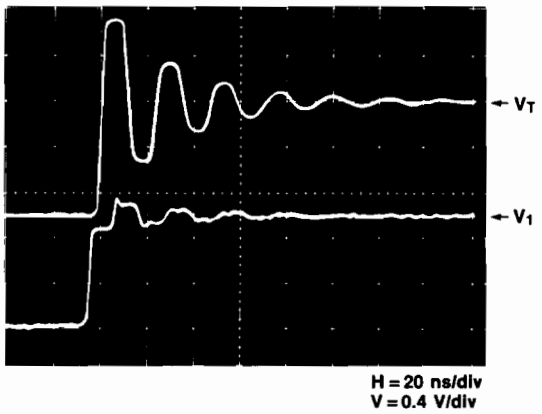


Figure 9-4 Extended Ringing when R_S of Figure 9-3 is Reduced to 13Ω

The vertical lines symbolize discontinuity points, in this case the ends of the line. A time scale is marked off on each line in increments of 2T, starting at t_0 for V_1 and T for V_T. The diagonal lines indicate the incremental voltages traveling between the ends of the line; solid lines are used for positive voltages and dashed lines for negative. It is helpful to write the reflection and transmission multipliers ρ and $(1+\rho)$ at each vertical line, and to tabulate the incremental and net voltages in columns alongside the vertical lines. Both the lattice diagram and the waveform photograph show that V_1 and V_T asymptotically approach 1V, as they must with a 1V source driving an open-ended line.

Shorted Line

The open-ended line in Figure 9-3 has a reflection coefficient of $+1$ and the successive reflections tend toward the steady state conditions of zero line current and a line voltage equal to the source voltage. In contrast, a shorted line has a reflection coefficient of -1 and successive reflections must cause the line conditions to approach the steady state conditions of zero voltage and a line current determined by the source voltage and resistance.

Shorted line conditions are shown in Figure 9-6a with the reflection coefficient at the source end of the line also negative. A negative coefficient at both ends of the line means that any voltage approaching either end of the line is reflected in the opposite polarity. Figure 9-6b shows the response to an input step-function with a duration much longer than the line delay. The initial voltage starting down the line is about $+0.75$ V, which is inverted at the shorted end and returned toward the source as -0.75 V. Arriving back at the source end of the line, this voltage is multiplied by $(1+\rho_S)$, causing a -0.37V net change in V_1. Concurrently, a reflected voltage of $+0.37$V (-0.75V times ρ_S of -0.5) starts back toward the shorted end of the line. The voltage at V_1 is reduced by 50% with each successive round trip of reflections, thus leading to the final condition of zero volts on the line.

When the duration of the input pulse is less than the delay of the line, the reflections observed at the source end of the line constitute a train of negative pulses, as shown in Figure 9-6c. The amplitude decreases by 50% with each successive occurrence as it did in Figure 9-6b.

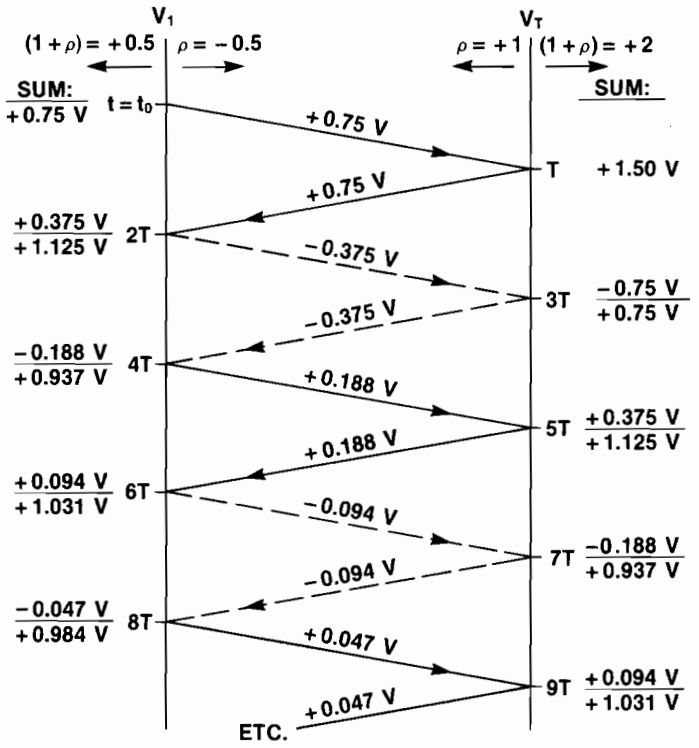


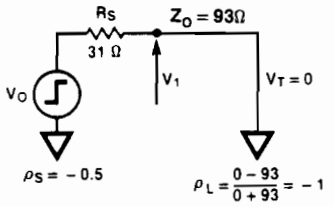
Figure 9-5 Lattice Diagram for the Circuit of Figure 9-3

Series Termination

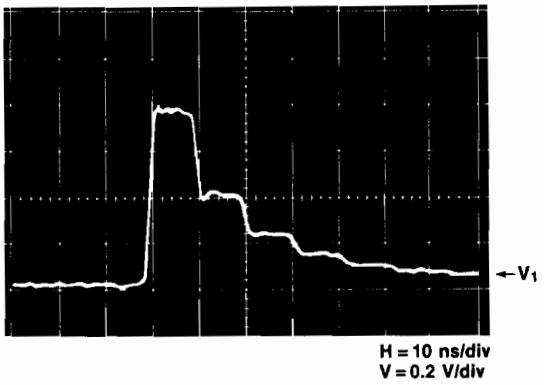
Driving an open-ended line through a source resistance equal to the line impedance is called series termination. It is particularly useful when transmitting signals which originate on a PC board and travel through the back-plane to another board, with the attendant discontinuities, since reflections coming back to the source are absorbed and ringing thereby controlled. Figure 9-7 shows a 93Ω line driven from a 1V generator through a source impedance of 93Ω. The photograph illustrates that the amplitude of the initial signal sent down the line is only half of the generator voltage, while the voltage at the open end of the line is doubled to full amplitude ($1 + \rho_L = 2$). The reflected voltage arriving back at the source raises V_1 to the full amplitude of the generator signal. Since the reflection coefficient at the source is zero, no further changes occur and the line voltage is equal to the generator voltage. Because the initial signal on the line is only half the normal signal swing, the loads must be connected at or near the end of the line to avoid receiving a 2-step input signal.

A TTL output driving a series-terminated line is severely limited in its fanout capabilities due to the IR drop associated with the collective I_{IL} drops of the inputs being driven. For most TTL families other than FAST it should not be considered since either the input currents are so high (TTL, S, H) or the input threshold is very low (LS). In either case the noise margins are severely degraded to the point where the circuit becomes unusable. In FAST, however, the I_{IL} of 0.6mA, if sunk through a resistor of 93Ω used as a series terminating resistor, will reduce the low level noise margin 55.8mV for each standard FAST input driven.

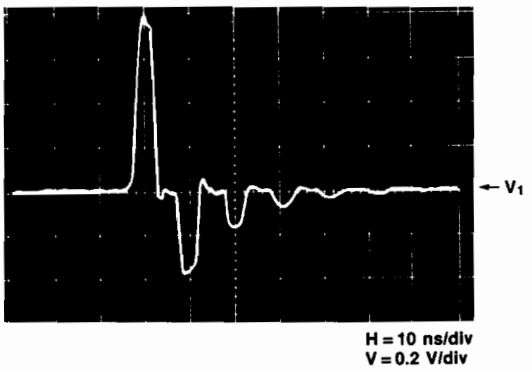
Figure 9-6 Reflections of Long and Short Pulses on a Shorted Line



a. Reflection Coefficients for Shorted Line



b. Input Pulse Duration >> Line Delay



c. Input Pulse Duration < Line Delay

Figure 9-7 Series-Terminated Line Waveforms

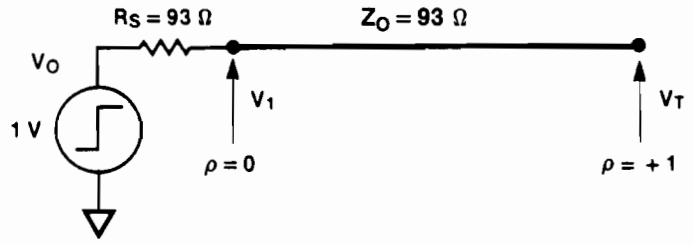
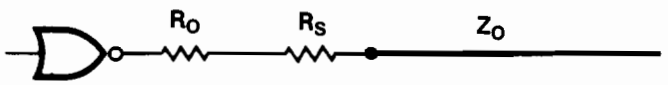


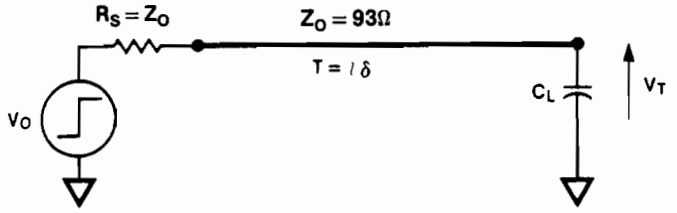
Figure 9-8 TTL Element Driving a Series-Terminated Line



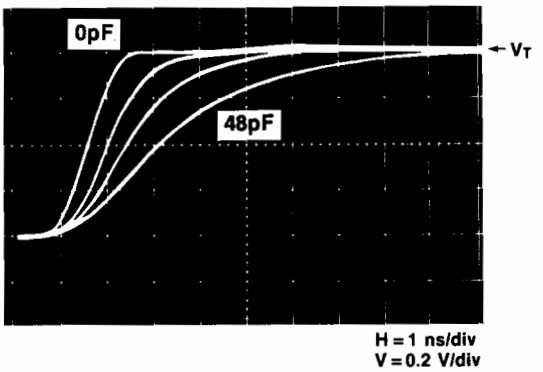
Extra Delay with Termination Capacitance

Designers should consider the effect of the load capacitance at the end of the line when using series termination. Figure 9-9 shows how the output waveform changes with increasing load capacitance. Figure 9-9b shows the effect of load capacitances of 0, 12, 24, 48pF. With no load, the delay between the 50% points of the input and output is just the line delay T. A capacitive load at the end of the line causes an extra delay ΔT due to the increase in rise time of the output signal. The midpoint of the signal swing is a good approximation of the FAST threshold since $V_{OL} = 0.5V$ and $V_{OH} = 2.5V$ and the actual input switching threshold of FAST is 1.5V at 25°C.

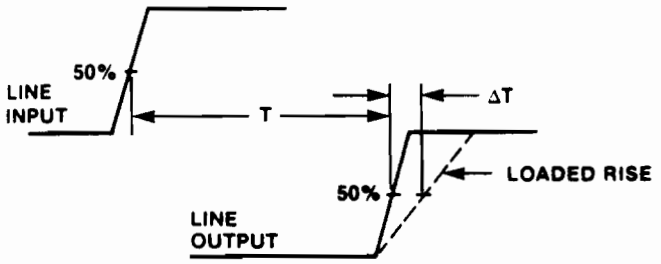
Figure 9-9 Extra Delay with Termination Capacitance



a. Series-Terminated Line with Load Capacitance



b. Output Rise Time Increase with Increasing Load Capacitance

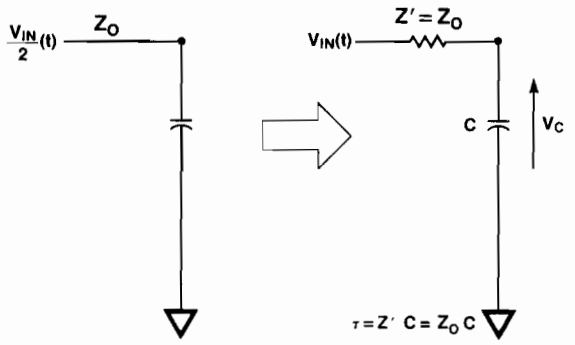


c. Extra Delay ΔT Due to Rise Time Increase

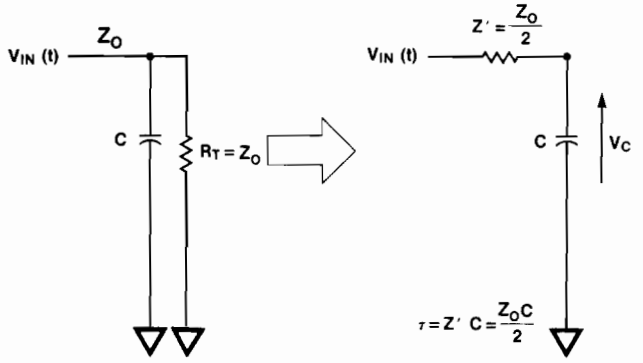
The increase in propagation delay can be calculated by using a ramp approximation for the incident voltage and characterizing the circuit as a fixed impedance in series with the load capacitance, as shown in Figure 9-10. One general solution serves both series and parallel termination cases by using an impedance Z′ and a time constant, τ, defined in Figure 9-10a and 9-10b. Calculated and observed increases in delay time to the 50% point show close agreement when τ is less than half the ramp time. At large ratios of τ/a (where a = ramp time), measured delays exceed calculated values by approximately 7%. Figure 9-11, based on measured values, shows the increase in delay to the 50% point as a function of the Z′C time constant, both normalized to

the 10% to 90% rise time of the input signal. As an example of using the graph, consider a 100Ω series-terminated line with 30pF load capacitance at the end of the line. The 3ns rise time assumed is typical of FAST in an actual line driving application. From Figure 9-10a, Z′ is equal to 100Ω; the ratio Z′C t_r is 1. From the graph, the ratio $\Delta T/t_r$ is 0.8. Thus the increase in the delay to the 50% point of the output waveform is 0.8 t_r, or 2.4ns, which is then added to the no-load line delay T to determine the total delay.

Figure 9-10 Determining the Effect of End-of-Line Capacitance



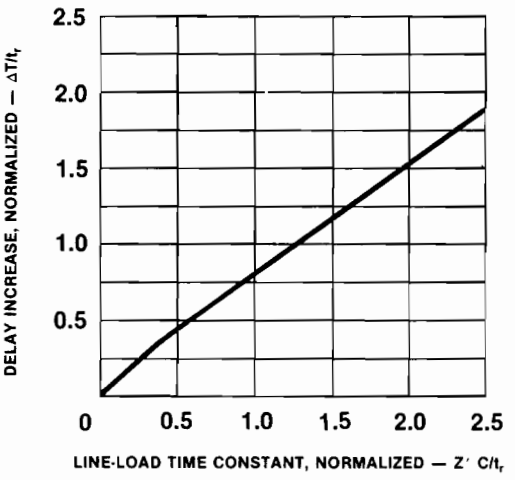
a. Thevenin Equivalent for Series-Terminated Case



b. Thevenin Equivalent for Parallel-Terminated Case

Had the 100Ω line in the foregoing example been parallel rather than series terminated at the end of the line, Z′ would be 50Ω. The added delay would be only 1.35ns with the same 30pF loading at the end. The added delay would be only 0.75ns if the line were 50Ω and parallel-terminated. The various trade-offs involving type of termination, line impedance, and loading are important considerations for critical delay paths.

Figure 9-11 Increase in 50% Point Delay Due to Capacitive Loading at the End of the Line, Normalized to t_r

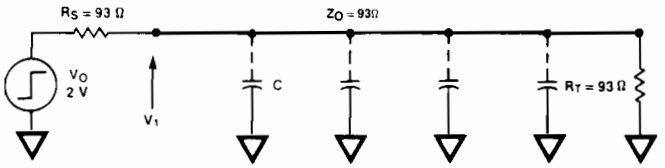


Distributed Loading Effects on Line Characteristics

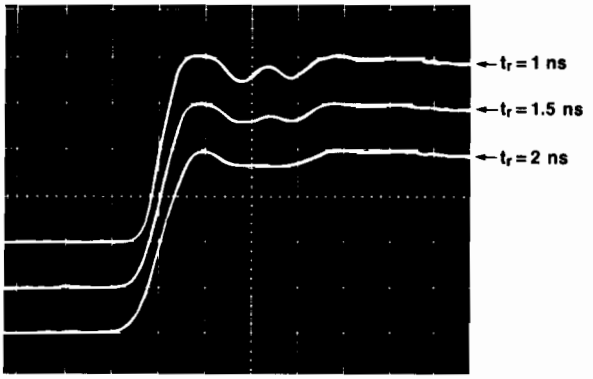
When capacitive loads such as TTL inputs are connected along a transmission line, each one causes a reflection with a polarity opposite to that of the incident wave. Reflections from two adjacent loads tend to overlap if the time required for the incident wave to travel from one load to the next is equal to or less than the signal rise time. Figure 9-12a illustrates an arrangement for observing the effects of capacitive loading, while Figure 9-12b shows an incident wave followed by reflections from two capacitive loads. The two capacitors causing the reflections are separated by a distance requiring a travel time of 1ns. The two reflections return to the source 2ns apart, since it takes 1ns longer for the incident wave to reach the second capacitor and an additional 1ns for the second reflection to travel back to the source. In the upper trace of Figure 9-12b, the input signal rise time is 1ns and there are two distinct reflections, although the trailing edge of the first overlaps the leading edge of the second. The input rise time is longer in the middle trace, causing a greater overlap. In the lower trace, the 2ns input rise time causes the two reflections to merge and appear as a single reflection which is relatively constant (at $\approx -10\%$) for half its duration. This is about the same reflection that would occur if the 93Ω line had a middle section with an impedance reduced to 75Ω.

With a number of capacitors distributed all along the line of Figure 9-12a, the combined reflections modify the observed input waveform as shown in the top trace of Figure 9-12c. The reflections persist for a time equal to the 2-way line delay (15ns), after which the line voltage attains its final value. The waveform suggests a line terminated with a resistance greater than its characteristic impedance ($R_T > Z_O$). This analogy is strengthened by observing the effect of reducing R_T from 93Ω to 75Ω which leads to the middle waveform of Figure 9-12c. Note that the final (steady state) value of the line voltage is reduced by about the same amount as that caused by the capacitive reflections. In the lower trace of Figure 9-12c the source resistance R_S is reduced from 93Ω to 75Ω, restoring both the initial and final line voltage values to the same amplitude as the final value in the upper trace. From the standpoint of providing a desired signal voltage on the line and impedance matching at either end, the effect of distributed capacitive loading can be treated as a reduction in line impedance.

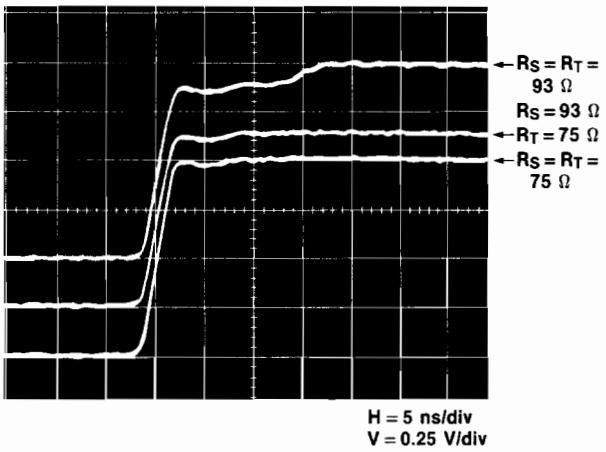
Figure 9-12 Capacitive Reflections and Effects on Line Characteristics



a. Arrangement for Observing Capacitive Loading Effects



b. Capacitive Reflections Merging as Rise Time Increases



c. Matching the Altered Impedance of a Capacitively Loaded Line

The reduced line impedance can be calculated by considering the load capacitance C_L as an increase in the intrinsic line capacitance C_O along that portion of the line where the loads are connected. Denoting this length of line as l, the distributed value C_D of the load capacitance is as follows:

$$C_D = \frac{C_L}{l}$$

C_D is then added to C_O in Equation 9-1 to determine the reduced line impedance Z_O.

$$Z_O' = \sqrt{\frac{L_O}{C_O + C_D}} = \sqrt{\frac{L_O}{C_O\left(1+\frac{C_D}{C_O}\right)}}$$

(E9-13)

$$Z_O' + \frac{\sqrt{\frac{L_O}{C_O}}}{\sqrt{1+\frac{C_D}{C_O}}} = \frac{Z_O}{\sqrt{1+\frac{C_D}{C_O}}}$$

In the example of Figure 9-12c, the total load capacitance (lC_O) is 60pF. Note that the ratio C_D/C_O is the same as C_L/lC_O. The calculated value of the reduced impedance is thus

(E9-14)

$$Z_O' = \frac{93}{\sqrt{1+\frac{33}{60}}} = \frac{93}{\sqrt{1.55}} = 75\Omega$$

This correlates with the results observed in Figure 9-12c when R_T and R_S are reduced to 75Ω.

The distributed load capacitance also increases the line delay, which can be calculated from Equation 9-2.

$$\delta' = \sqrt{L_O(C_O+C_D)} = \sqrt{L_O C_O}\ \sqrt{1+\frac{C_D}{C_O}} = \delta\sqrt{1+\frac{C_D}{C_O}}$$

(E9-15)

The line used in the example of Figure 9-12c has an intrinsic delay of 6ns and a loaded delay of 7.5ns which checks with Equation 9-15.

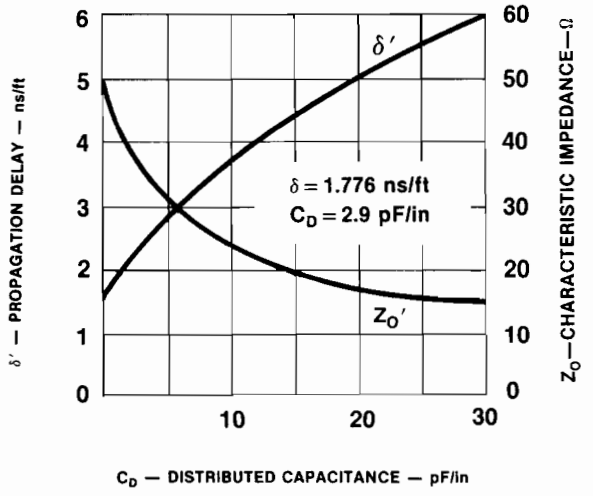
$$l\delta' = l\delta\sqrt{1.55} = 6\sqrt{1.55} = 7.5\ ns$$

(E9-16)

Equation 9-15 can be used to predict the delay for a given line and load. The ratio C_D/C_O (hence the loading effect) can be minimized for a given loading by using a line with a high intrinsic capacitance C_O.

A plot of Z′ and δ′ for a 50Ω line as a function of C_D is shown in Figure 9-13. This figure illustrates that relatively modest amounts of load capacitance will add appreciably to the propagation delay of a line. In addition, the characteristic impedance is reduced significantly.

Figure 9-13 Capacitive Loading Effects on Line Delay and Impedance



Worst case reflections from a capacitively loaded section of transmission line can be accurately predicted by using the modified impedance of Equation 9-9. When a signal originates on an unloaded section of line, the effective reflection coefficient is as follows:

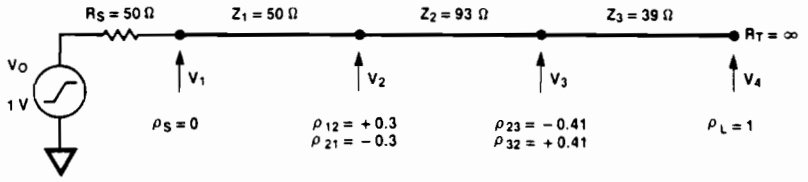
$$\rho = \frac{Z_O' - Z_O}{Z_O' + Z_O} \qquad \textbf{(E9-17)}$$

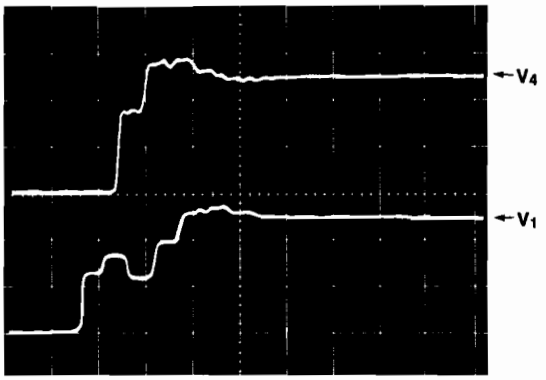
Mismatched Lines

Reflections occur not only from mismatched load and source impedances but also from changes in line impedance. These changes could be caused by bends in coaxial cable, unshielded twisted-pair in contact with metal, or mismatch between PC board traces and backplane wiring. With the coax or twisted-pair, line impedance changes run about 5 to 10% and reflections are usually no problem since the percent reflection is roughly half the percent change in impedance. However, between PC board and backplane wiring, the mismatch can be 2 or 3 to 1. This is illustrated in Figure 9-14 and analyzed in the lattice diagram of Figure 9-15. Line 1 is driven in the series-terminated mode so that reflections coming back to the source are absorbed.

The reflection and transmission at the point where impedances differ are determined by treating the downstream line as though it were a terminating resistor. For the example of Figure 9-14, the reflection coefficient at the intersection of lines 1 and 2 for a signal traveling to the right is as follows:

$$\rho_{12} = \frac{Z_2 - Z_1}{Z_2 + Z_1} = \frac{93 - 50}{143} = +0.3 \qquad \textbf{(E9-18)}$$





H = 20 ns/div
V = 0.4 V/div

Figure 9-14 Reflections from Mismatched Lines

Thus the signal reflected back toward the source and the signal continuing along line 2 are, respectively, as follows:

$$V_{1r} = \rho_{12}\, V_1 = +0.3\, V_1$$

$$V_2 = (1+\rho_{12})\, V_1 = +1.3\, V_1$$

(E9-19)

At the intersection of lines 2 and 3, the reflection coefficient for signals traveling to the right is determined by treating Z_3 as a terminating resistor.

$$\rho_{23} = \frac{Z_3 - Z_2}{Z_3 + Z_2} = \frac{39-93}{132} = -0.41$$

(E9-20)

When V_2 arrives at this point, the reflected and transmitted signals are as follows:

$$V_{2r} = \rho_{23}\, V_2 = -0.41\, V_2$$
$$= (-0.41)\ (1.3)\ V_1)$$
$$= -0.53\, V_1$$

(E9-21a)

$$V_3 = (1+\rho_{23})\, V_2 = 0.59\, V_2$$
$$= (0.59)\ (1.3)\ V_1$$
$$= 0.77\, V_1$$

(E9-21b)

Voltage V_3 is doubled in magnitude when it arrives at the open-ended output, since ρ_L is $+1$. This effectively cancels the voltage divider action between R_S and Z_1.

$$V_4 = (1+\rho_L)\, V_3 = (1+\rho_L)\ (1+\rho_{23})\, V_2$$
$$= (1+\rho_L)\ (1+\rho_{23})\ (1+\rho_{12})\, V_1$$
$$= (1+\rho_L)\ (1+\rho_{23})\ (1+\rho_{12})\, \frac{V_O}{2}$$
$$V_4 = (1+\rho_{23})\ (1+\rho_{12})\, V_O$$

(E9-22)

Thus, Equation 9-22 is the general expression for the initial step of output voltage for three lines when the input is series-terminated and the output is open-ended. Note that the reflection coefficients at the intersections of lines 1 and 2 and lines 2 and 3 in Figure 9-15 have reversed signs for signals traveling to the left. Thus the voltage reflected from the open output and the signal

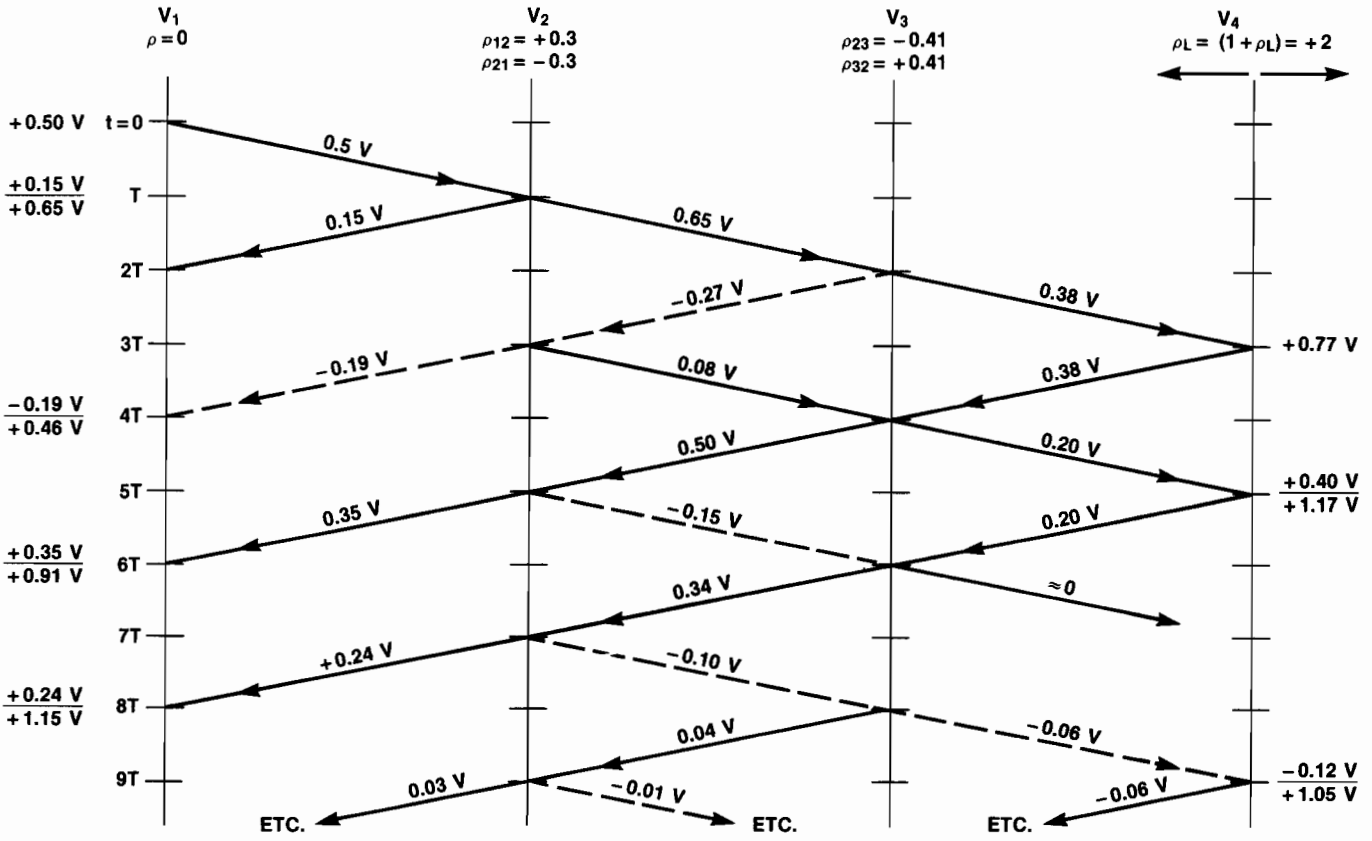


Figure 9-15 Lattice Diagram for the Circuit of Figure 9-14

reflecting back and forth on line 2 both contribute additional increments of output voltage in the same polarity as V_O. Lines 2 and 3 have the same delay time; therefore, the two aforementioned increments arrive at the output simultaneously at time 5T on the lattice diagram (Figure 9-15).

In the general case of series lines with different delay times, the vertical lines on the lattice diagram should be spaced apart in the ratio of the respective delays. Figure 9-16 shows this for a hypothetical case with delay ratios 1:2:3. For a sequence of transmission lines with the highest impedance line in the middle, at least three output voltage increments with the same polarity as V_O occur before one can occur of opposite polarity. On the other hand, if the middle line has the lowest impedance, the polarity of the second increment of output voltage is the opposite of V_O. The third increment of output voltage has the opposite polarity, for the time delay ratios of Figure 9-16.

When transmitting logic signals, it is important that the initial step of line output voltage pass through the threshold region of the receiving circuit, and that the next two increments of output voltage augment the initial step. Thus in a series-terminated sequence of three mismatched lines, the middle line should have the highest impedance.

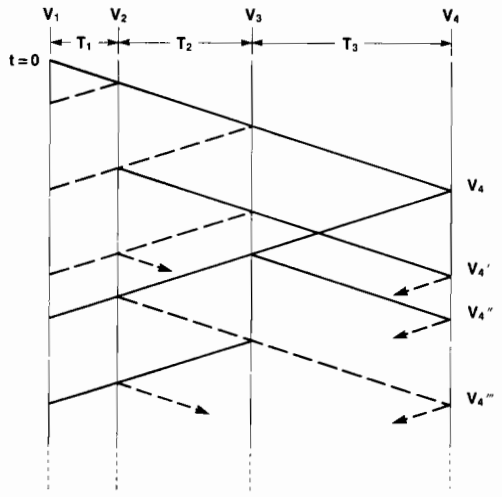


Figure 9-16 Lattice Diagram for Three Lines with Delay Ratios 1:2:3

Rise Time Versus Line Delay

When the 2-way line delay is less than the rise time of the input wave, any reflections generated at the end of the line are returned to the source before the input transition is completed. Assuming that the generator has a finite source resistance, the reflected wave adds algebraically to the input wave while it is still in transition, thereby changing the shape of the input. This effect is illustrated in Figure 9-17, which shows input and output voltages for several comparative values of rise time and line delay.

In Figure 9-17b where the rise time is much shorter than the line delay, V_1 rises to an initial value of 1V. At time T later, V_T rises to 0.5 V, i.e., $1+\rho_L=0.5$. The negative reflection arrives back at the source at time 2T, causing a net change of -0.4V, i.e., $(1+\rho_S)(-0.5)=-0.4$.

The negative coefficient at the source changes the polarity of the other 0.1V of the reflection and returns it to the end of the line, causing V_T to go positive by another 50 mV at time 3T. The remaining 50 mV is inverted and reflected back to the source, where its effect is barely distinguishable as a small negative change at time 4T.

In Figure 9-17c, the input rise time (0 to 100%) is increased to such an extent that the input ramp ends just as the negative reflection arrives back at the source end. Thus the input rise time is equal to 2T.

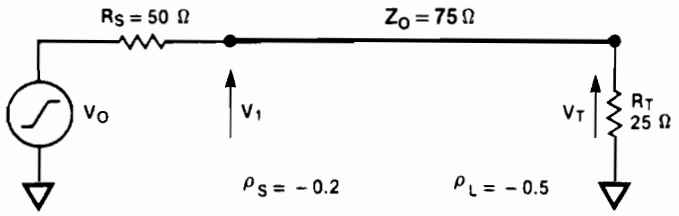
The input rise time is increased to 4T in Figure 9-17d, with the negative reflection causing a noticeable change in input slope at about its midpoint. This change in slope is more visible in the double exposure photo of Figure 9-17e, which shows V_1 (t_r still set for 4T) with and without the negative reflection. The reflection was eliminated by terminating the line in its characteristic impedance.

The net input voltage at any particular time is determined by adding the reflection to the otherwise unaffected input. It must be remembered that the reflection arriving back at the input at a given time is proportional to the input voltage at a time 2T earlier. The value of V_1 in Figure 9-17d can be calculated by starting with the 1V input ramp.

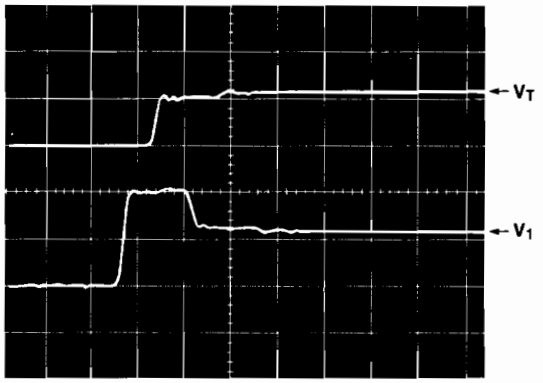
$$V_1 = \frac{1}{t_r} \cdot t \text{ for } 0 \le t \le 4T$$
$$= 1\ V \quad \text{for } t \ge 4T$$

(E9-23)

Figure 9-17 Line Voltages for Various Ratios of Rise Time to Line Delay

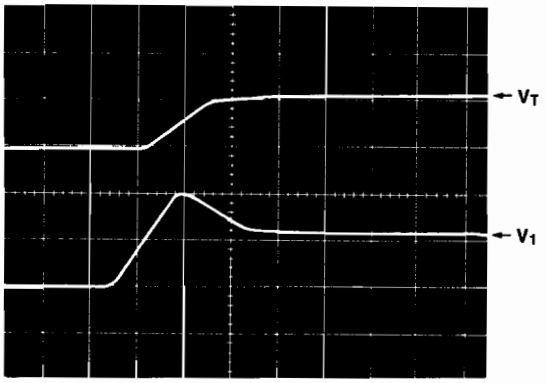


a. Test Arrangement for Rise Time Analysis



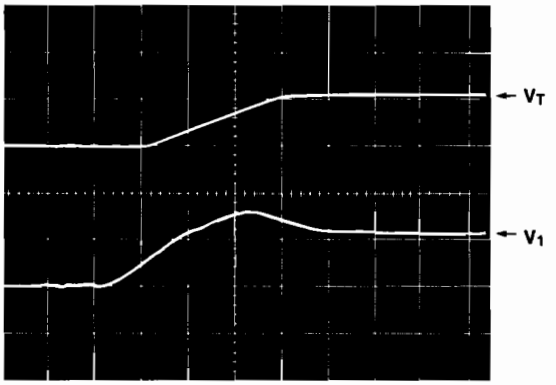
H = 10 ns/div
V = 0.5 V/div

b. Line Voltages for $t_r << T$



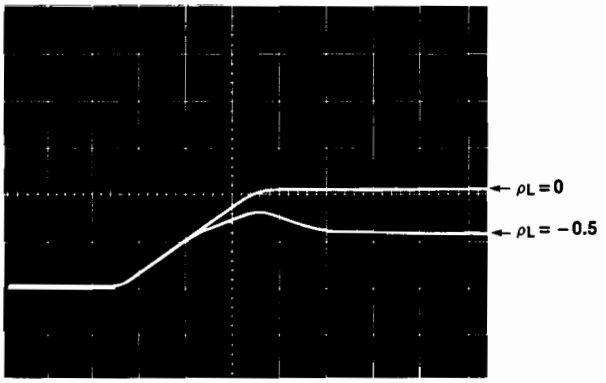
H = 10 ns/div
V = 0.5 V/div

c. Line Voltages for $t_r = 2T$



H = 10 ns/div
V = 0.5 V/div

d. Line Voltages for $t_r = 4T$



H = 10 ns/div
V = 0.5 V/div

e. Input Voltage With and Without Reflection

The reflection from the end of the line is

$$V_r = \frac{\rho_L (t - 2T)}{t_r};$$

(E9-24)

the portion of the reflection that appears at the input is

$$V'_r = \frac{(1 + \rho_S)\ \rho_L\ (t - 2T)}{t_r};$$

(E9-25)

the net value of the input voltage is the sum.

$$V'_1 = \frac{t}{t_r} + \frac{(1 + \rho_S)\ \rho_L\ (t - 2T)}{t_r}$$

(E9-26)

The peak value of the input voltage in Figure 9-17d is determined by substituting values and letting t equal 4T.

$$V'_1 = \frac{(0.8)\ (-0.5)\ (4T - 2T)}{t_r}$$

$$= 1 - .04\ (0.5) = 0.8\ V$$

(E9-27)

After this peak point, the input ramp is no longer increasing but the reflection is still arriving. Hence the net value of the input voltage decreases. In this example, the later reflections are too small to be detected and the input voltage is thus stable after time 6T. For the general case of repeated reflections, the net voltage $V_{1(t)}$ seen at the driven end of the line can be expressed as follows, where the signal caused by the generator is $V_{1(t)}$:

$$V'_{1(t)} = V_{1(t)} \quad \text{for } 0 < t < 2T$$

$$V'_{1(t)} = V_{1(t)} + (1+\rho_S)\,\rho_L\,V_{1(t-2T)} \quad \text{for } 2T < t < 4T$$

$$V'_{1(t)} = V_{1(t)} + (1+\rho_S)\,\rho_L\,V_{1(t-2T)} + (1+\rho_S)\,\rho_S\rho_L^{\,2}\,V_{1(t-4T)} \quad \text{for } 4T < t < 6T$$

$$V'_{1(t)} = V_{1(t)} + (1+\rho_S)\,\rho_L\,V_{1(t-2T)} + (1+\rho_S)\,\rho_S\rho_L^{\,2}\,V_{1(t-4T)} + (1+\rho_S)\,\rho_S^{\,2}\rho_L^{\,3}\,V_{1(t-6T)} \quad \text{for } 6T < t < 8T, \text{ etc.}$$

(E9-28)

The voltage at the output end of the line is expressed in a similar manner.

$$V_{T(t)} = 0 \quad \text{for } 0 < t < T$$

$$V_{T(t)} = (1+\rho_L)\,V_{1(t-T)} \quad \text{for } T < t < 3T$$

$$V_{T(t)} = (1+\rho_L)\,V_{1(t-T)} + (1+\rho_L)\,\rho_S\rho_L\,V_{1(t-3T)} \quad \text{for } 3T < t < 5T$$

$$V_{T(t)} = (1+\rho_L)\,V_{1(t-T)} + (1+\rho_L)\,\rho_S\rho_L\,V_{1(t-3T)} + (1+\rho_L)\,\rho_S^{\,2}\rho_L^{\,2}\,V_{1(t-5T)} \quad \text{for } 5T < t < 7T, \text{ etc.}$$

(E9-29)

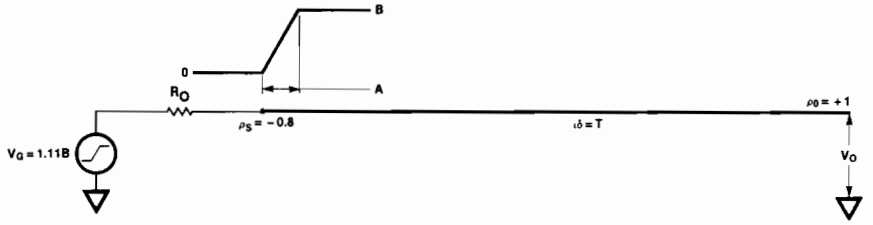
Ringing

Multiple reflections occur on a transmission line when neither the signal source impedance nor the termination (load) impedance matches the line impedance. When the source reflection coefficient ρ_S and the load reflection coefficient ρ_L are of opposite polarity, the reflections alternate in polarity. This causes the signal voltage to oscillate about the final steady state value, commonly recognized as ringing.

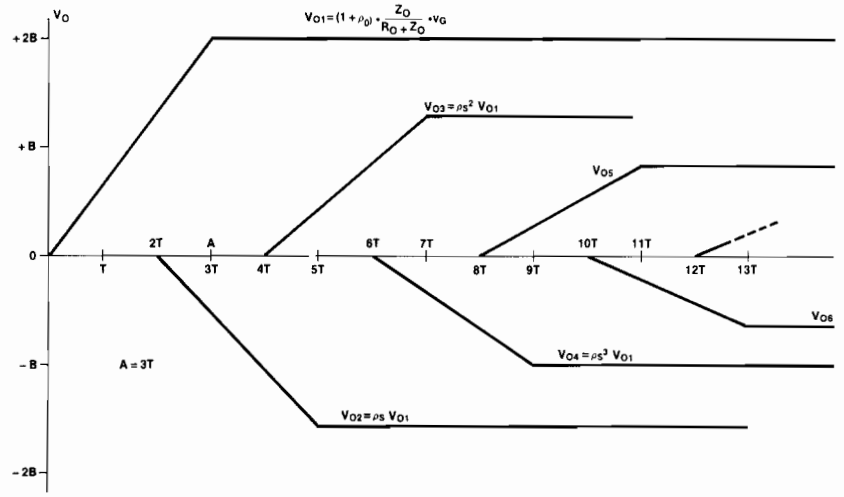
When the signal rise time is long compared to the line delay, the signal shape is distorted because the individual reflections overlap in time. The basic relationships among rise time, line delay, overshoot and undershoot are shown in a simplified diagram, Figure 9-18. The incident wave is a ramp of amplitude B and rise duration A. The reflection coefficient at the open-ended line output is $+1$ and the source reflection coefficient is assumed to be -0.8, i.e., $R_O = Z_O/9$.

Figure 9-18b shows the individual reflections treated separately. Rise time A is assumed to be three times the line delay T. The time scale reference is the line output and the first increment of output voltage V_O rises to 2B in the time interval A. Simultaneously, a positive reflection (not shown) of amplitude B is generated and travels to the source, whereupon it is multiplied by -0.8 and returns toward the end of the line. This negative-going ramp starts at time 2T (twice the line delay) and doubles to -1.6B at time $2T + A$.

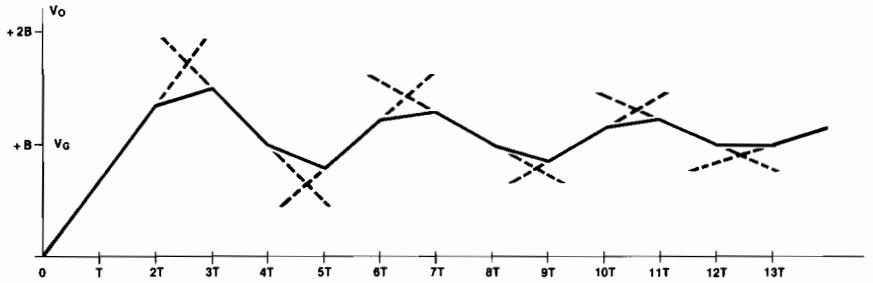
Figure 9-18 Basic Relationships Involved in Ringing



a. Ramp Generator Driving Open-Ended Line



b. Increments of Output Voltage Treated Individually



c. Net Output Signal Determined by Superposition

The negative-going increment also generates a reflection of amplitude $-0.8B$ which makes the round trip to the source and back, appearing at time 4T as a positive ramp rising to $+1.28B$ at time $4T + A$. The process of reflection and re-reflection continues, and each successive increment changes in polarity and has an amplitude of 80% of the preceding increment.

In Figure 9-18c, the output increments are added algebraically by superposition. The starting point of each increment is shifted upward to a voltage value equal to the algebraic sum of the quiescent levels of all the preceding increments (i.e., 0, 2B, 0.4B, 1.68B, etc.). For time intervals when two ramps occur simultaneously, the two linear functions add to produce a third ramp that prevails during the overlap time of the two increments.

It is apparent from the geometric relationships, that if the ramp time A is less than twice the line delay, the first output increment has time to rise to the full 2B amplitude and the second increment reduces the net

output voltage to 0.4B. Conversely, if the line delay is very short compared to the ramp time, the excursions about the final value V_G are small.

Figure 9-18c shows that the peak of each excursion is reached when the earlier of the two constituents ramps reaches its maximum value, with the result that the first peak occurs at time A. This is because the earlier ramp has a greater slope (absolute value) than the one that follows.

Actual waveforms such as produced by TTL do not have a constant slope and do not start and stop as abruptly as the ramp used in the example of Figure 9-18. Predicting the time at which the peaks of overshoot and undershoot occur is not as simple as with ramp excitation. A more rigorous treatment is required, including an expression for the driving waveform which closely simulates its actual shape. In the general case, a peak occurs when the sum of the slopes of the individual signal increment is zero.

Summary

The foregoing discussions are by no means an exhaustive treatment of transmission line characteristics. Rather, they are intended to focus attention on the general methods used to determine the interactions between high-speed logic circuits and their interconnections. Considering an interconnection in terms of distributed rather than lumped inductance and capacitance leads to the line impedance concept, i.e., mismatch between this characteristic impedance and the terminations causes reflections and ringing.

Series termination provides a means of absorbing reflections when it is likely that discontinuities and/or line impedance changes will be encountered. A disadvantage is that the incident wave is only one-half the signal swing, which limits load placement to the end of the line. TTL input capacitance increases the rise time at the end of the line, thus increasing the effective delay. With parallel termination, i.e., at the end of the line, loads can be distributed along the line. TTL input capacitance modifies the line characteristics and should be taken into account when determining line delay.

Line Driving

All interconnects, such as coaxial cable, defined impedance transmission lines and feeders, can be considered as transmission lines, whereas printed circuit traces and hook-up wire tend to be ignored as transmission lines. With any high-speed logic family, all interconnects should be considered as transmission lines, and evaluated as such to see if termination is required. Of the many properties of transmission lines, two are of major interest to us: Z_O (the effective equivalent resistive value that causes zero reflection) and t_{PD} (propagation delay down the transmission line). Both of these parameters are geometry dependent. Here are some common configurations:

Printed Circuit Configurations

h = dielectric thickness
c = trace thickness
L = trace length
K = dielectric thickness between ground planes
b = trace width
ϵ_r = dielectric constant

Figure 9-19 Micro Stripline

$$Z_O = \frac{87}{\sqrt{\epsilon_r + 1.41}} \ln\left(\frac{5.98h}{0.8\,b + c}\right) \ \Omega \qquad \textbf{(E9-30)}$$

$$t_{PD} = 1.017\sqrt{0.475\,\epsilon_r + 0.67}\ \ ns/ft.$$

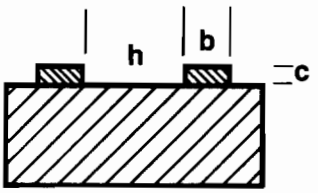
Figure 9-20 Stripline

$$Z_O = \frac{60}{\sqrt{\epsilon_r}} \ln\left(\frac{4K}{0.67\,\pi\,b\left(0.8 + \frac{c}{b}\right)}\right) \Omega$$

$$\textbf{(E9-31)} \qquad t_{PD} = 1.017\sqrt{\epsilon_r}\ \ ns/ft.$$

9

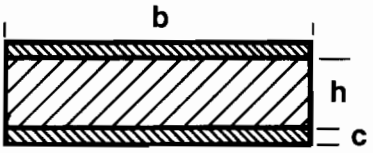
Figure 9-21 Side by Side



$$Z_O = \frac{120}{\sqrt{\epsilon_r}} \ln\left(\frac{\pi h}{b+c}\right) \quad \Omega$$

(E9-32) $$t_{PD} = 1.017\sqrt{0.475\,\epsilon_r + 0.67} \quad ns/ft.$$

Figure 9-22 Flat Parallel Conductors for b>>h and h>>c

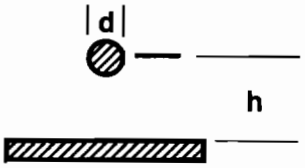


$$Z_O = \frac{377}{\sqrt{\epsilon_r}} \ln\left(\frac{h}{b}\right) \quad \Omega$$

(E9-33) $$t_{PD} = 1.017\sqrt{0.475\,\epsilon_r + 0.67} \quad ns/ft.$$

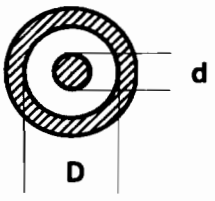
Figure 9-23 Wiring

a. Wire over Ground Plane



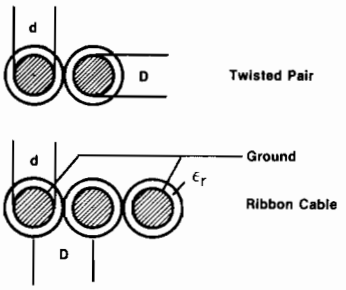
(E9-34) $$Z_O = \frac{60}{\sqrt{\epsilon_r}} \ln\left(\frac{4h}{d}\right) \quad \Omega$$

b. Coaxial Cable



(E9-35) $$Z_O = \frac{60}{\sqrt{\epsilon_r}} \ln\left(\frac{D}{d}\right) \quad \Omega$$

Figure 9-24 Twisted Pair or Ribbon Cable



(E9-36) $$Z_O = \frac{120}{\sqrt{\epsilon_r}} \ln\left(\frac{2D}{d}\right) \quad \Omega$$

All of the above rely on the complex relationship

(E9-37) $$Z_O = \sqrt{\frac{R_O + j\,\omega\,L_O}{G_O + j\,\omega\,C_O}} \quad \Omega$$

and can be simplified to

$$Z_O = \sqrt{\frac{L_O}{C_O}}$$

if we assume

$$G_O = R_O = 0$$

Note that Z_O is real, not complex, appears resistive and is not a function of length.

Also,

(E9-38) $$t_{PD} = \sqrt{L_O C_O}$$

The inductance of PC trace can be determined by the formula

(E9-39) $$L_O = \left[0.0051 \;\; Ln\left(\frac{4h}{\propto}\right)\right] + .00127 \;\; \mu H/inch$$

$$where \propto = \sqrt{\frac{4\,bc}{\pi}}$$

For power and ground planes in a multilayer board, the capacitance of the plane can be calculated by the formula for parallel plates separated by a dielectric:

(E9-40) $$C = 0.2212\,\frac{\epsilon_r A}{h} \quad pF$$

where A = surface area of one plate.

The above formula (E9-40) cannot be used to calculate PC trace capacitance. This must either be measured or an appropriate value may be taken from the following curves.

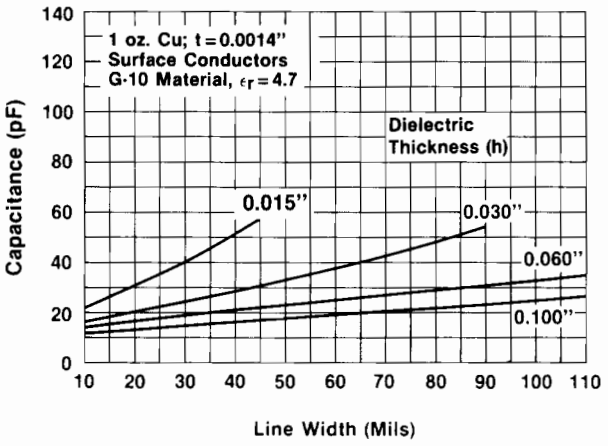


Figure 9-25 Capacitance of Microstriplines

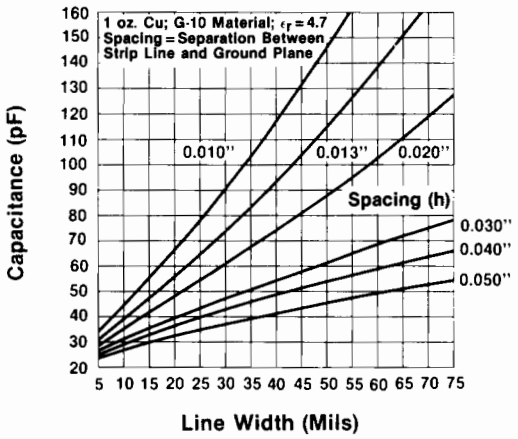


Figure 9-26 Capacitance of Striplines

The impedance of striplines and microstriplines can be found quickly from the following curves. For characteristics of cables, refer to manufacturers' data.

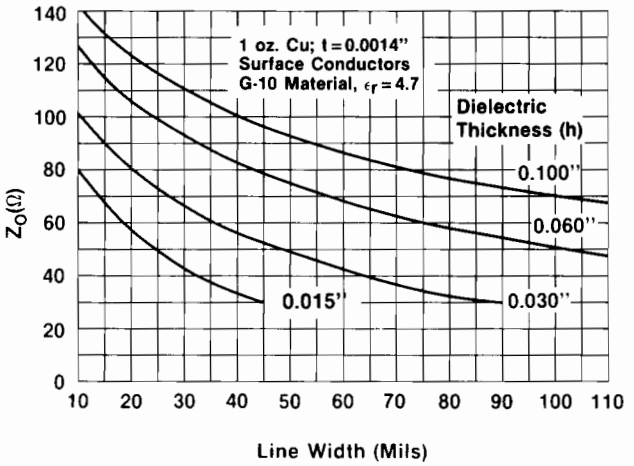


Figure 9-27 Impedance of Microstriplines

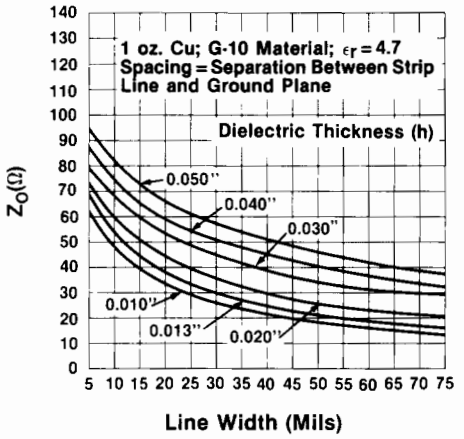


Figure 9-28 Impedance of Striplines

9

Table 9-1
Relative Dielectric Constants of Various Materials

Material	ϵ_r
Air	1.0
Polyethylene foam	1.6
Cellular polyethylene	1.8
Teflon	2.1
Polyethylene	2.3
Polystyrene	2.5
Nylon	3.0
Silicon rubber	3.1
Polyvinylchloride (PVC)	3.5
Epoxy resin	3.6
Delrin	3.7
Epoxy glass	4.7
Mylar	5.0
Polyurethane	7.0

All the above information on impedance and propagation delays are for the circuit interconnect only. The actual impedance and propagation delays will differ from this by the loading effects of gate input and output capacitances, and by any connectors that may be in line. The effective impedance and propagation delay can be determined from the following formula:

$$Z_O' = \frac{Z_O}{\sqrt{1+\left(\frac{C_L}{C_O}\right)}} \ \Omega$$

$$t_{PD} = \sqrt{L_O C_O} \quad \therefore \ t_{PD}' = t_{PD}\sqrt{1+\left(\frac{C_L}{C_O}\right)}$$

(E9-41)

where C_L is the total of all additional loading.

The results of these formulas will frequently give effective impedances of less than half Z_O, and interconnect propagation delays greater than the driving device propagation delays, thus becoming the predominant delay.

Driving Transmission Lines

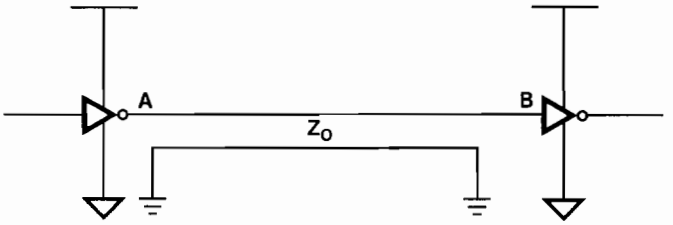


Figure 9-29
1. Unterminated

The maximum length for an unterminated line can be determined by

$$l_{max} = \frac{t_r}{2t_{PD}'} \quad \text{For FAST, } t_r = 3\ ns$$

$\therefore \ l_{max} = 10$ inches for trace on GIO epoxy glass P.C.

(E9-42)

The voltage wave propagated down the transmission line (V step) is the full output drive of the device into Z_O'. Reflections will not be a problem if $l \leqq l_{max}$. Lines longer than l_{max} will be subject to ringing and reflections and will drive the inputs and outputs below ground.

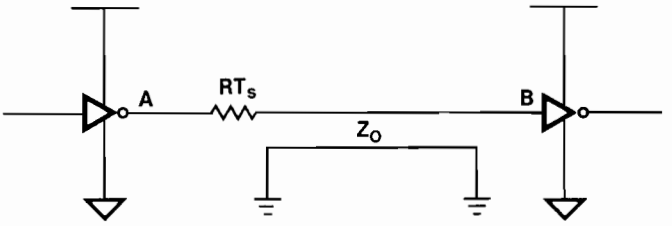


Figure 90-30
2. Series-Terminated

$RT_S = Z_O$

Series termination has limited use in TTL interconnect schemes due to the voltage drop across RT_S in the LOW state, reducing noise margins at the receiver. Series termination is the ideal termination for highly capacitive memory arrays whose DC loadings are minimal. RT_S values of 10 to 50Ω are normally found in these applications.

3. Parallel-Terminated

Four possibilities for parallel termination exist:

A. Z_O' to V_{CC}. This will consume current from V_{CC} when output is LOW;

B. Z_O' to GND. This will consume current from V_{CC} when output is HIGH;

C. Thevenin equivalent termination. This will consume half the current of A and B from the output stage, but will have reduced noise margins, and consume current from V_{CC} with outputs HIGH or LOW. If used on a 3-state bus, this will set the quiescent line voltage to half.

D. AC Termination. An RC termination to GND, $R + X_C = Z_O$, X_C to be less than 2_7 of Z_O' at

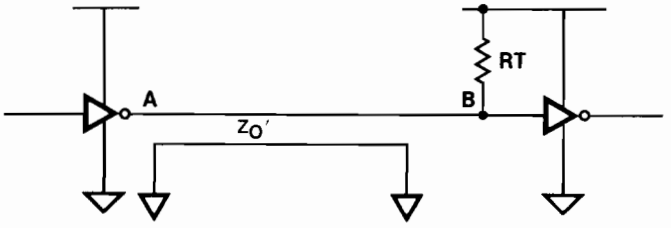
$$f = \frac{1}{2t_r}$$

(E9-43)

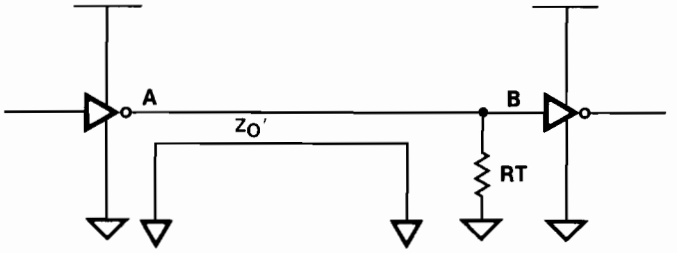
This consumes no DC current with outputs in either state. If this is used on a 3-state bus, then the quiescent voltage on the line can be established at V_{CC} or GND by a high value pull up (down) resistor to the appropriate supply rail.

Parallel-Terminated

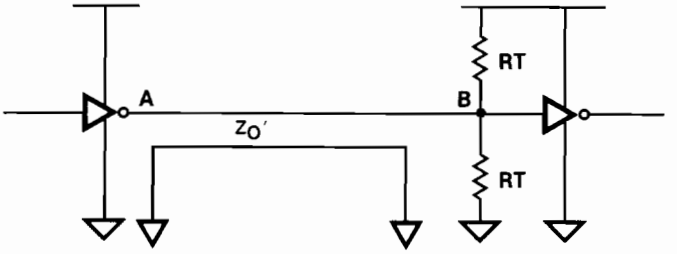
A. RT to V_{CC}
RT = Z_O'



B. RT to GND
RT = Z_O'



C. Thevenin Termination
RT = $2Z_O'$



D. AC Termination to GND
RT + X_{CT} = Z_O'

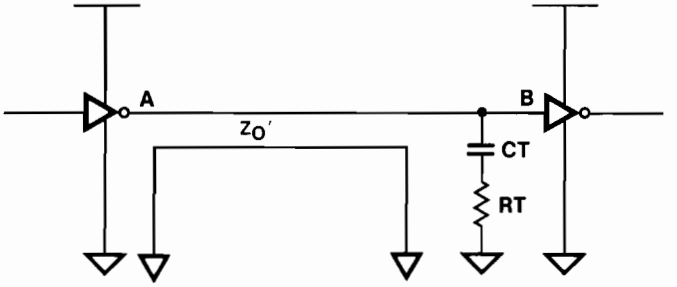


Figure 9-31

Decoupling

Typical Dynamic Impedance of Unbypassed V_{CC} Runs

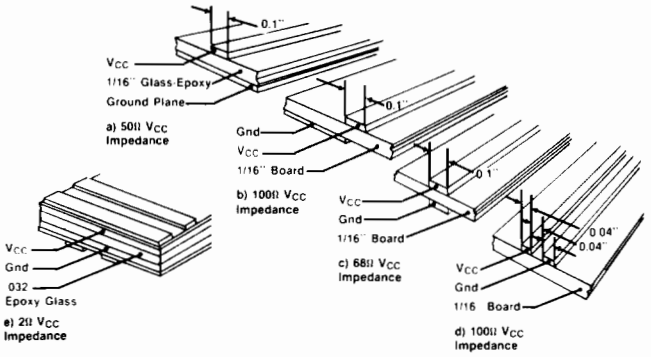


Figure 9-32

This diagram shows several schemes for power and ground distribution on logic boards. Figure 9-32 is a cross-section, with a, b, and c showing a 0.1 inch wide V_{CC} bus and ground on the opposite side. Figure 9-32d shows side-by-side V_{CC} and ground strips, each 0.04 inch wide. Figure 9-32e shows a four layer board with embedded power and ground planes.

In Figure 9-32a, the dynamic impedance of V_{CC} with respect to ground is 50Ω, even though the V_{CC} trace width is generous and there is a complete ground plane. In Figure 9-32b, the ground plane stops just below the edge of the V_{CC} bus and the dynamic impedance doubles to 100Ω. In Figure 9-32c, the ground bus is also 0.1 inch wide and runs along under the V_{CC} bus and exhibits a dynamic impedance of about 68Ω. In Figure 9-32d, the trace widths and spacing are such that the traces can run under a DIP, between two rows of pins. The impedance of the power and ground planes in Figure 9-32e is typically less than 2Ω.

These typical dynamic impedances point out why a sudden current demand due to an IC output switching can cause a momentary reduction in V_{CC}, unless a bypass capacitor is located near the IC.

I_{CC} Drain Due to Line Driving

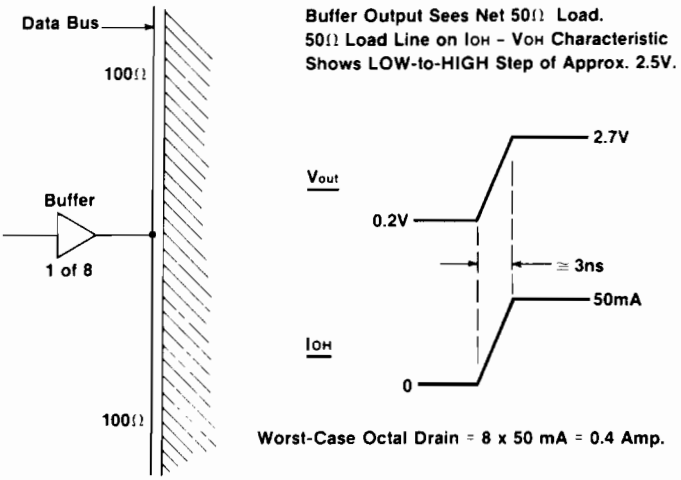


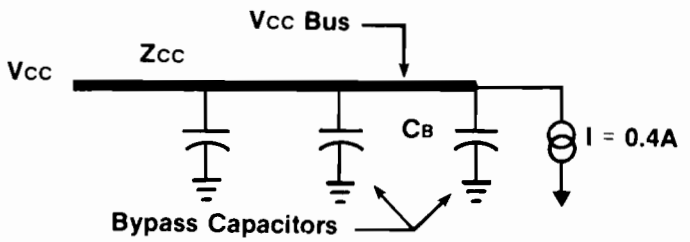
Figure 9-33

This diagram illustrates the sudden demand for current from V_{CC} when a buffer output forces a LOW-to-HIGH transition into the midpoint of a data bus. The sketch shows a wire-over-ground transmission line, but it could also be twisted pair, flat cable or PC interconnect.

The buffer output effectively sees two 100Ω lines in parallel and thus a 50Ω load. For this value of load impedance, the buffer output will force an initial LOW-to-HIGH transition from 0.2V to 2.7V in about 3ns. This net charge of 2.5V into a 50 load causes an output-HIGH current change of 50mA.

If all eight outputs of an octal buffer switch simultaneously, in this application the current demand on V_{CC} would be 0.4 Amp. Clearly, a nearby V_{CC} bypass capacitor is needed to accommodate this demand.

V_{CC} Bypass Capacitor for Octal Driver



$Q = CV$

$I = C\triangle V/\triangle t$

$C = I\triangle t/\triangle V$

$\triangle t = 3 \times 10^{-9}$

Specify V_{CC} Droop = 0.1 V max.

$$C = \frac{0.4 \times 3 \times 10^{-9}}{0.1} = 12 \times 10^{-9} = 0.012\mu F$$

Select $C_B \geq 0.02\mu F$

Place one bypass capacitor near each buffer package. Distribute other bypass capacitors evenly throughout the logic, one capacitor per two packages.

Figure 9-34

A V_{CC} bus with bypass capacitors connected periodically along its length is shown above. Also shown is a current source representing the current demand of the buffer in the preceding application.

The equations illustrate an approximation method of estimating the size of a bypass capacitor based on the current demand, the drop in V_{CC} that can be tolerated and the length of time that the capacitor must supply the charge. While the current demand is known, the other two parameters must be chosen. A V_{CC} droop of 0.1V will not cause any appreciable change in performance, while a time duration of 3ns is long enough for other nearby bypass capacitors to help supply charge. If the current demand continues over a long period of time, charge must be supplied by a very large capacitor on the board. This is the reason for the recommendation that a large capacitor be located where V_{CC} comes onto a board. If the buffers are also located near the connector end of the board, the large capacitor helps supply charge sooner.

Design Considerations

Ground—An Essential Link

With the advent of Fairchild Advanced Schottky Technology (FAST) with considerably faster edge rates and switching times, proper grounding practice has become of primary concern in printed circuit layout. Poor circuit grounding layout techniques may result in crosstalk and slowed switching rates. This reduces overall circuit performance and may necessitate costly redesign. Also when FAST chips are substituted for standard TTL-designed printed circuit boards, faster edge rates can cause noise problems. The source of these problems can be sorted into three categories:

1. V_{CC} droop due to faster load capacitance charging;
2. Coupling via ground paths adjacent to both signal sources and loads; and
3. Crosstalk caused by parallel signal paths.

V_{CC} droop can be remedied with better or more bypassing to ground. The rule here is to place 0.01μF capacitors from V_{CC} to ground for every two FAST circuits used, as near the IC as possible. The other two problems are not as easily corrected, because PC boards may already be manufactured and utilized. In this case, simply replacing TTL circuits with FAST compatible circuits is not always as easy as it may seem, especially on two-sided boards. In this situation IC placement is critical at high speeds. Also when designing high density circuit layout, a ground-plane layer is imperative to provide both a sufficiently low inductance current return path and to provide electromagnetic and electrostatic shielding thus preventing noise problem 2 and reducing, by a large degree, noise problem 3.

Illustrations

Two-Sided PC Board Layout

When considering the two-sided PC board, more than one ground trace is often found in a parallel or non-parallel configuration. For this illustration parallel traces tied together at one end are shown. This arrangement is referred to as a ground comb. The ground comb is placed on one side of the PC board while the signal traces are on the other side, thus the two-sided circuit board.

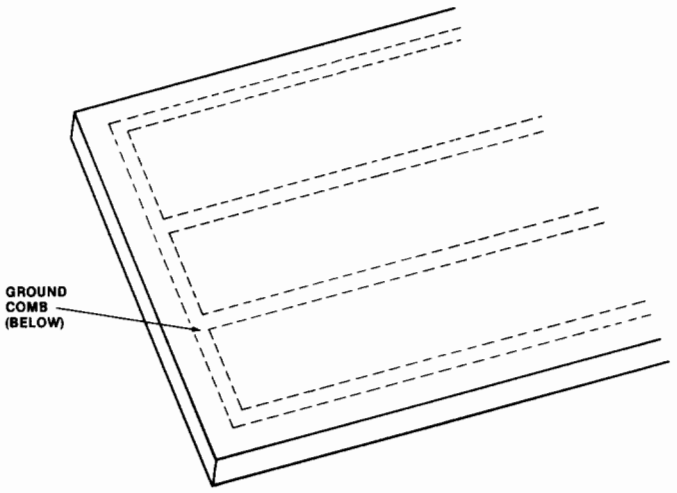


Figure 9-35

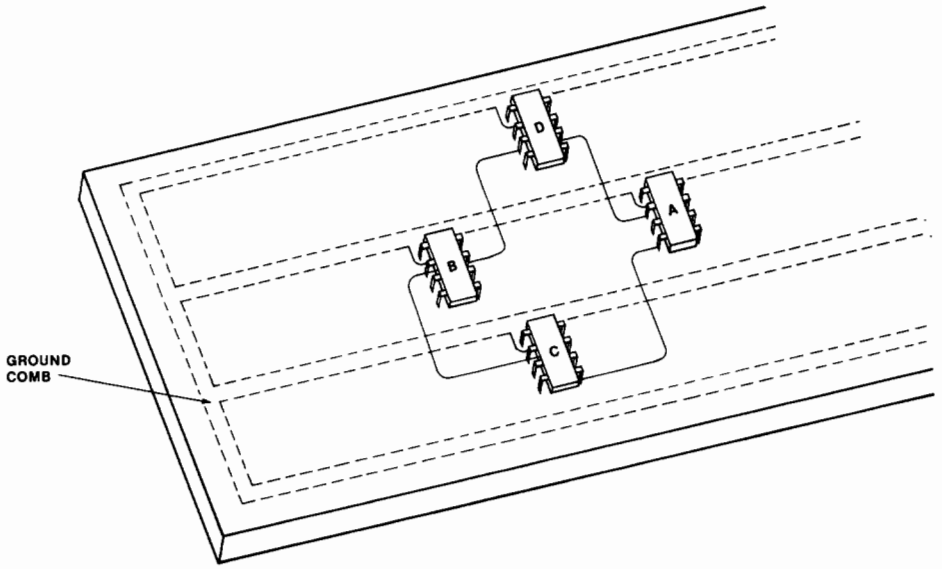


Figure 9-36

Figure 9-36 illustrates how noise is generated even though there is no apparent means of crosstalk between the circuits. If package A has an output which drives package D input and package B output drives package C input, there is no apparent path for crosstalk since mutual signal traces are remotely located. What is significant, and must be emphasized here, is that circuit packages A and B accept their ground link from the same trace. Hence, circuit A may well couple noise to circuit B via the common or shared portion of the trace. This is especially true at high switching speeds.

Ground Trace Coupling

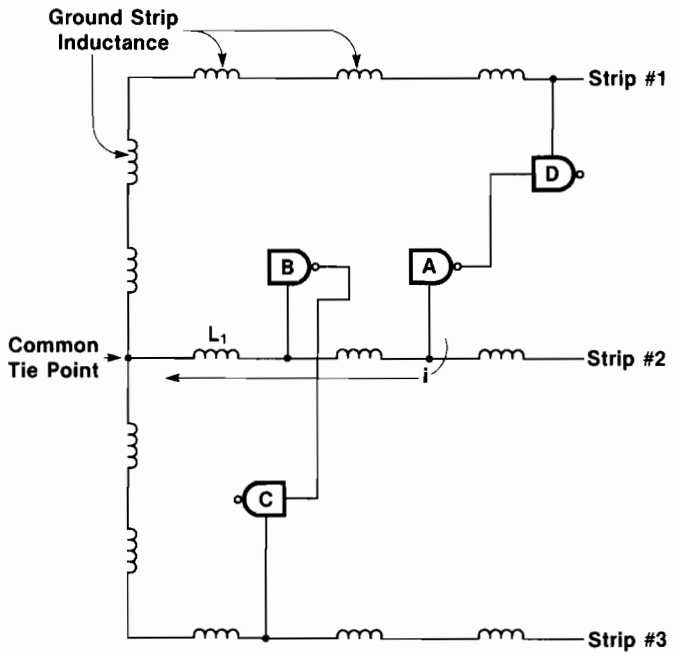


Figure 9-37

Ground trace noise coupling is illustrated by a model circuit in Figure 9-37. With the ground comb configuration, the ground strips may be shown to contain distributed inductance, as is indeed the case. Referring to the above illustration we can see that if we switch gate A from HIGH to LOW, the current for the transition is drawn from ground strip number two. Current flows in the direction indicated by the arrow to the common tie point. It can be seen that gate B shares ground strip number two with gate A from the point where gate B is grounded back to the common tie point. This length is represented by L_1. When A switches states there is a current transient which occurs on the ground strip in the positive direction. This current spike is caused by the ground strip inductance and it is "felt" by gate B. If gate B is in a LOW state (V_{OL}) the spike will appear on the output since gate B's V_{OL} level is with reference to ground. Thus if gate B's ground reference rises momentarily V_{OL} will also rise. Consequently, if gate B is output to another gate (C in the illustration) problems may arise.

Problem

System faults occur if the sum of V_{OL} quiescent level plus current spike amplitude reaches the threshold region of gate C. From this it can be seen that erroneous switching may be transmitted throughout the system. In the illustration the glitch at gate B's output is given by the following formula:

$$L_1 \, (d_i/d_t)$$

(E9-44)

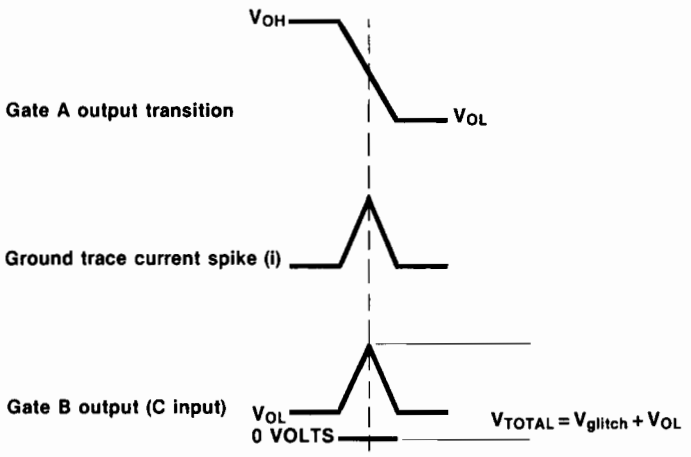


Figure 9-38

Solution

The following sketch (Figure 9-39) shows one method of effectively reducing ground path length when using the ground comb layout. By using topside traces to tie the underside ground comb together we can reduce ground strip distributed inductances. Therefore, current transients are significantly smaller or nonexistent in amplitude. In the application of these topside strips, care need not be exercised in their spacing or arrangement. Parallelism is not of paramount importance either. Another advantage is evident: if one or more of these strips is placed between topside signal traces, crosstalk can be eliminated between those traces.

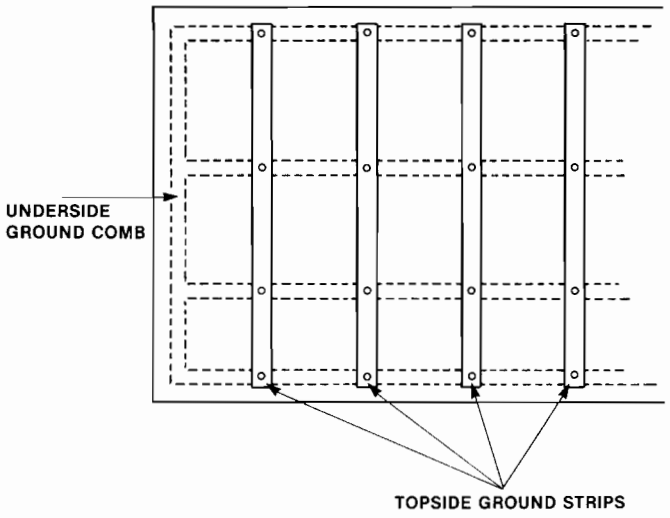


Figure 9-39

Bus Driver Packages

An area which warrants special consideration is bus driver/buffer package placement. Here we refer to products such as the 'F240, 'F241, 'F244, 'F540, 'F827 and 'F828. These units have a minimum of eight outputs. A problem may arise if all eight outputs happen to switch from HIGH to LOW or vice versa at the same time. In this case the chance for a large current transient on the ground circuit is apparent. This is possible even on short runs of ground strip (1 to 2 inches). Here, extra care is advised and it is suggested that buffer/driver groups driving backplanes be segregated to one area in the circuit. This area should have its own ground reference. Ideally it should be a ground plane configuration or contain minimal or negligible length ground trace connections.

General-Purpose Boards (Breadboard)

It is important when breadboarding, creating prototype circuits for evaluation or making special function generators, to use optimum techniques for connecting V_{CC} and ground. Breadboard-type selection is of certain consequence here and should be attended to wisely.

The best choice, when designing with high-speed logic, is board material which has power and ground already connected to circuit trace grids. Boards may offer the designer the option of using IC sockets although these are not recommended for high-speed applications. Socket layout is convenient and may be necessary when special or one-of-a-kind circuits are utilized in initial circuit arrangements. However, when designing with FAST products, consider that sockets increase total lead inductance and interlead capacitance, thus circuit performance may be adversely affected. If boards without ground and power grids must be used or if non-standard pin connections must be accommodated, the use of copper strips or braid is recommended. Copper strip is readily available as shim stock while a brand of solder wick can be used for braid material. Please note here that jumper wires must be avoided because of wire inductance. Wire inductance, like stripline inductance, will slow rise and fall times. Jumpers also promote crosstalk coupling.

Noise Decoupling

As stated earlier under "Ground—An Essential Link", it was noted that the common rule of thumb is to decouple every other FAST package with 0.01μF capacitors. This is fine for most gates in the majority of applications. However, with buffer/driver packages, decoupling should occur at each package since the possibility of all outputs switching coincidentally exists and can cause large loads on V_{CC}.

An alternative to standard decoupled power traces on the two-sided P.C. board is a product called Q/PAC*. Q/PAC is a low impedance, high capacitance power distribution system which uses wide V_{CC} and ground conductors in close proximity with ceramic insulators to effectively represent integral decoupling capacitors. Packages are available in varying lengths and pinout spacings.

* Rogers Corporation
Q/PAC Division, 5750 East McKellips Road
Mesa, AZ 85205 Telephone: 602-830-3370

Crosstalk

Crosstalk is an interference effect of an active signal line on an inactive signal line in close proximity to the active line. There are two forms of crosstalk that are of concern in system design: forward and reverse crosstalk. The causes of both both kinds are similar, but the effects are significantly different. Four possible crosstalk conditions can exist at the inactive receiver: (1) a positive pulse on a LOW, (2) a negative pulse on a LOW, (3) a positive pulse on a HIGH, (4) a negative pulse on a HIGH. Of the four previously mentioned conditions (1) and (4) are of major concern in logic systems, and (2) and (3) are less problematic. Crosstalk is caused by a number of interrelated factors which fall into two groups: mutual impedance and velocity difference.

Mutual impedance is caused by the mutual inductance and mutual capacitance distributed along two signal lines in close proximity. The electrical effects are akin to transformer action with well defined polarities. The induced crosstalk voltage pulse is of opposite polarity to the inducing pulse. Figure 9-40 shows the schematic representation.

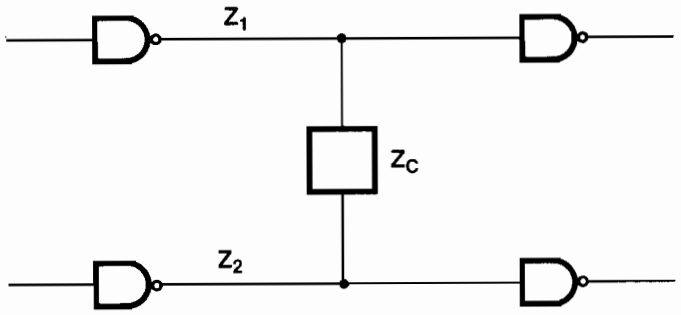


Figure 9-40

Here Z_1 and Z_2 represent the adjacent signal line impedances, and Z_C is the mutual impedance coupling the two signal lines. An equivalent circuit is shown below.

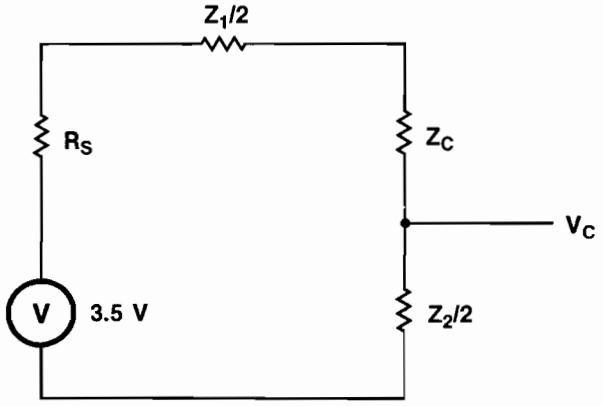


Figure 9-41

R_S is the effective source resistance; for V_{OH}, $R_S = 33\Omega$ and V_{OL}, $R_S = 3\Omega$. These are the typical FAST gate sink and source resistances. V_C is the crosstalk voltage and should be adjusted for polarity. The crosstalk voltage can be calculated with the following simplified formula:

$$V_C = \frac{Z_2/2}{R_S + Z_C + Z_1/2 + Z_2/2} \times V_{OUT}$$

(E9-45)

Velocity differences are caused when a signal propagates along a conductive medium that is in contact with substances of different dielectric constants, i.e., epoxy glass and air in printed circuit board applications. The different dielectric constants of the materials cause the wave propagating at the epoxy glass interface to be travelling slower than the wave at the air interface. This has the effect of generating a pulse that will couple electrostatically into the adjacent signal line and add to the pulse caused by mutual impedance coupling. The velocity difference pulse will have the same rise time as the signal on the active line and its duration will be twice the difference between the arrival of the wave front in air and the wave front in epoxy glass.

Forward Crosstalk

Forward crosstalk is the effect when the active driver and the driver on the non-active line are at the same end: the wave front propagates toward the active and non-active receiver simultaneously. Forward crosstalk is classically attributed almost entirely to velocity differences, but in practice it is a mixture of both velocity difference and mutual impedance effects.

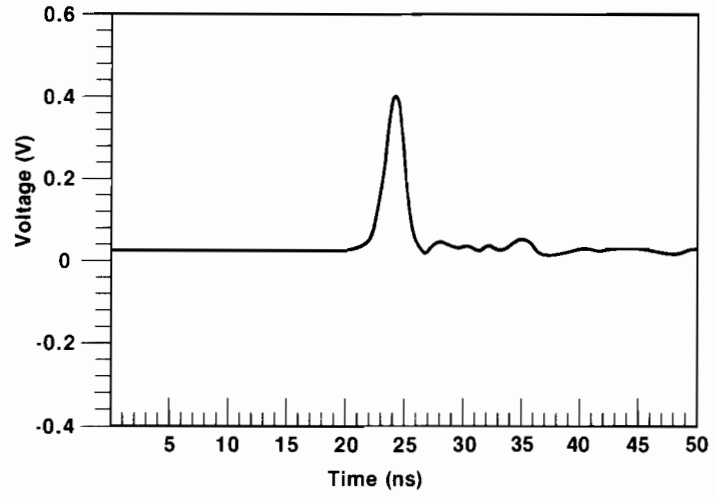
Reverse Crosstalk

Reverse or backward crosstalk is the effect when the active driver and the non-active receiver are at the same end of the signal lines: the wave front propagates toward the active receiver and the non-active driver simultaneously. Reverse crosstalk is due entirely to mutual impedance effects. Forward and reverse crosstalk tests have been performed on both parallel circuit board traces and ribbon cable.

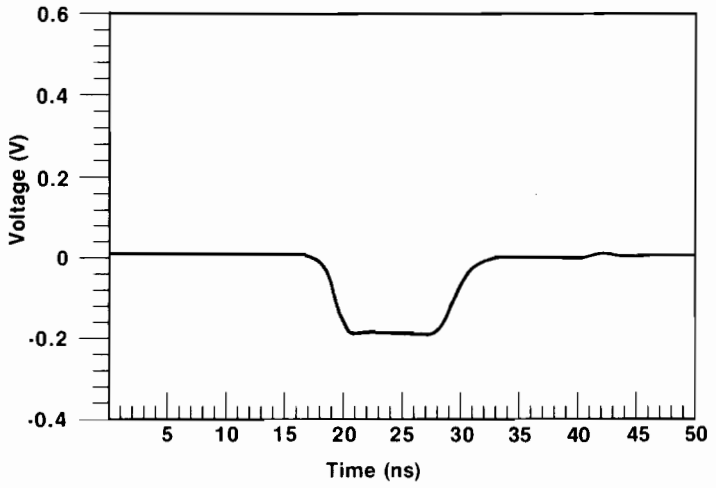
Crosstalk on PC Trace

Crosstalk on printed circuit traces exhibits both velocity difference and mutual impedance effects. This can be seen clearly in Figure 9-42. The jig, two 50Ω parallel traces, 34 inches long and 0.100 inches apart, was characterized using a 5V 3ns rise time signal from a 50Ω source and all traces terminated in 50Ω. Figures 9-43 through 9-50 show the effects of forward and reverse crosstalk on terminated and unterminated cases using the jig of Figure 9-42. All of the cases show no approach to the logic threshold on this test jig; other circuit configuarations and impedances may not act in a similar fashion and crosstalk avoidance procedures may have to be taken.

Figure 9-42 34-Inch, 50Ω Crosstalk Jig

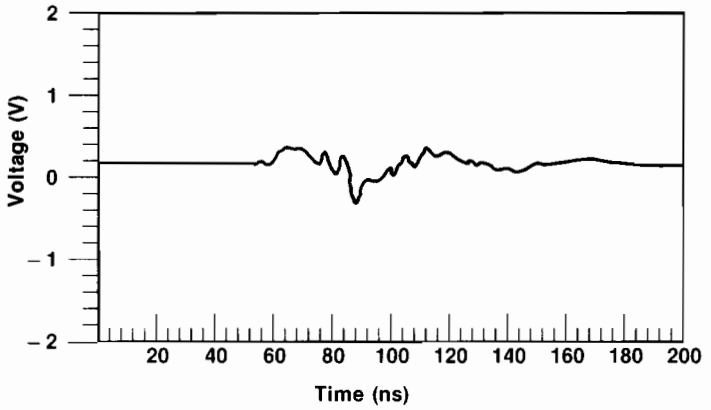


a. Velocity Difference

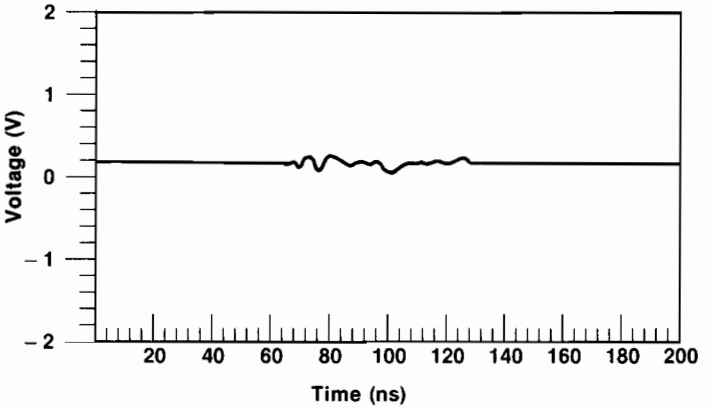


b. Mutual Impedance

Figure 9-43 Reverse PC Board Crosstalk Through 34-Inch, 0.100 Trace Unterminated

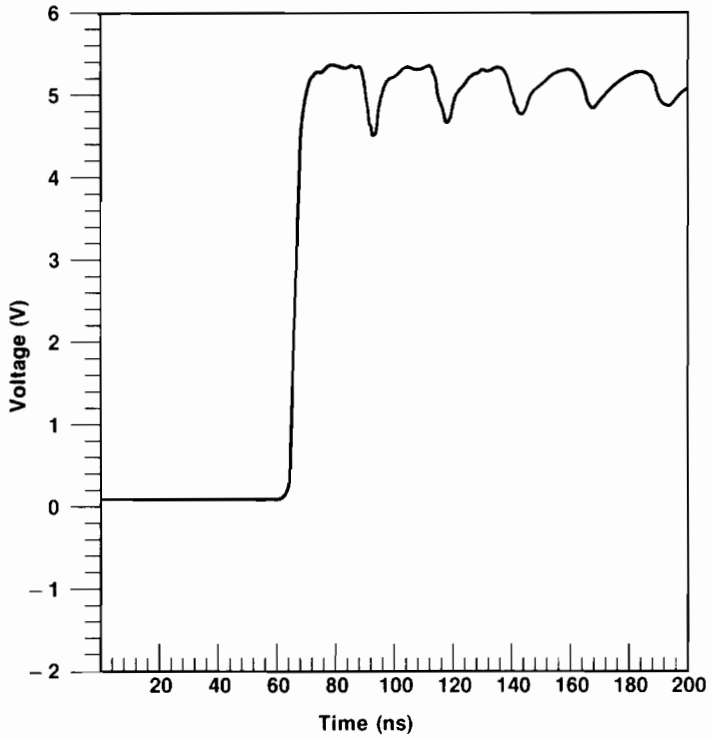


a. Adjacent Receiver

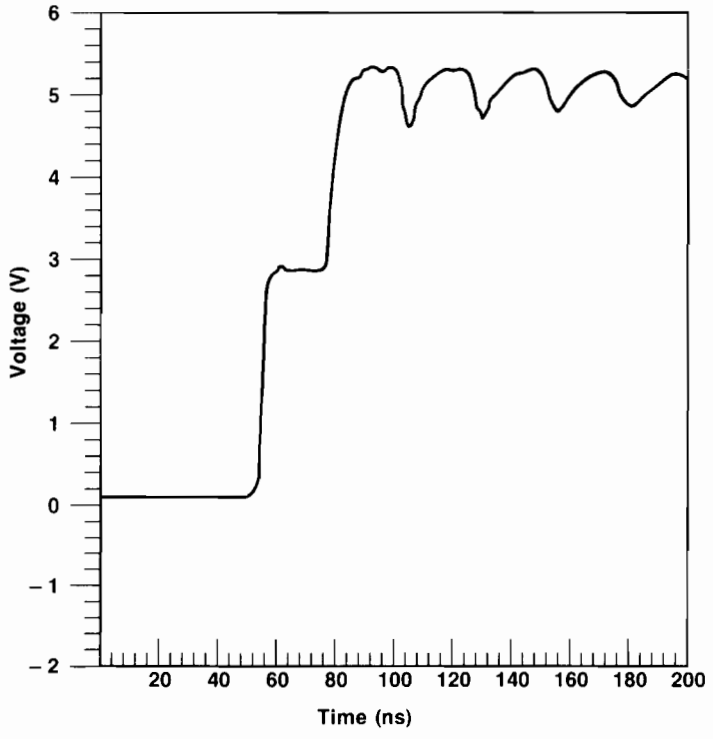


b. Adjacent Driver

Figure 9-43 (continued)

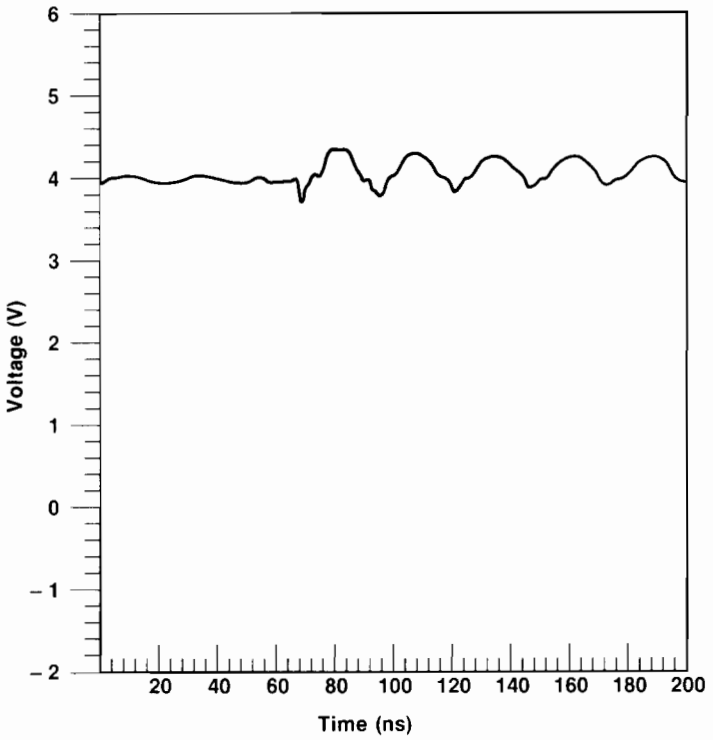


c. Active Receiver

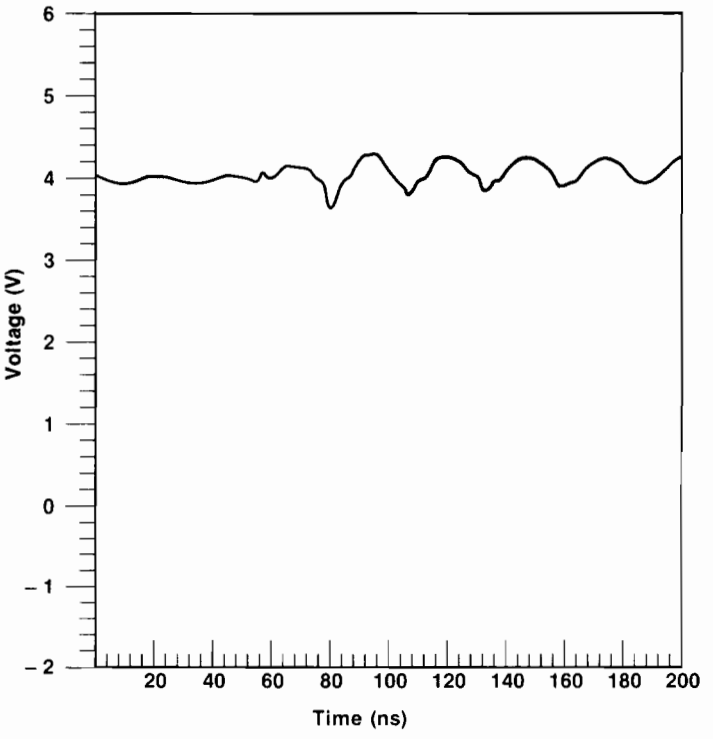


d. Active Driver

Figure 9-44 Forward PC Board Crosstalk Through 34-Inch Trace Unterminated



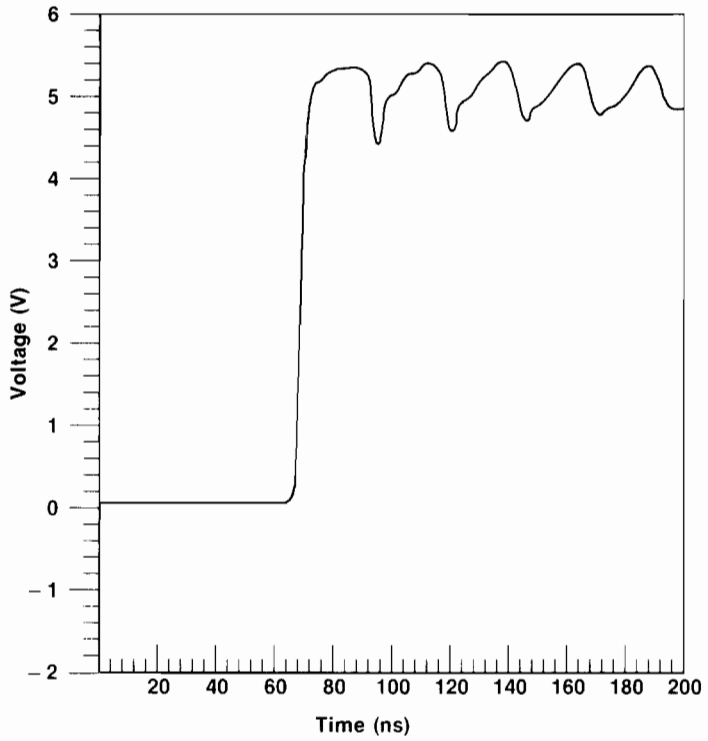
a. Adjacent Receiver



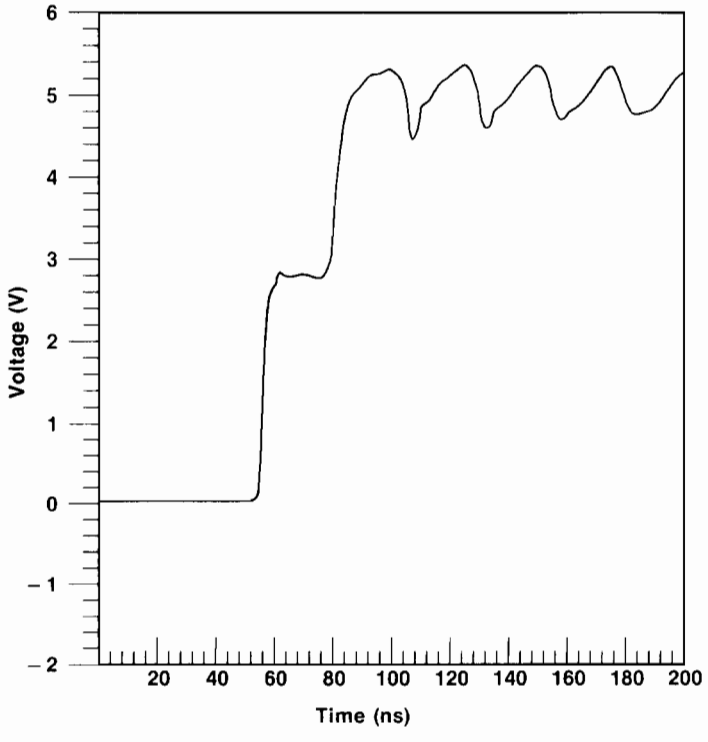
b. Adjacent Driver

9

Figure 9-44 (continued)

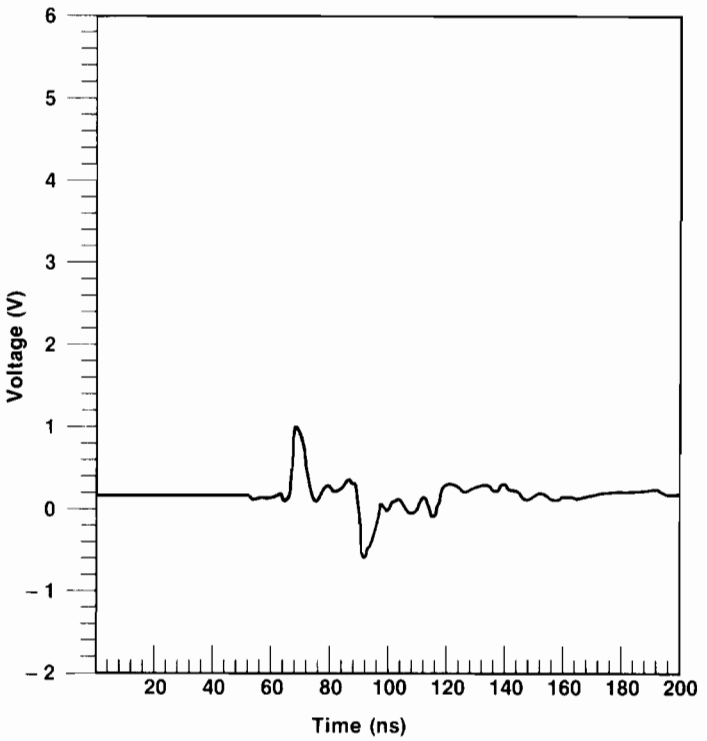


c. Active Receiver

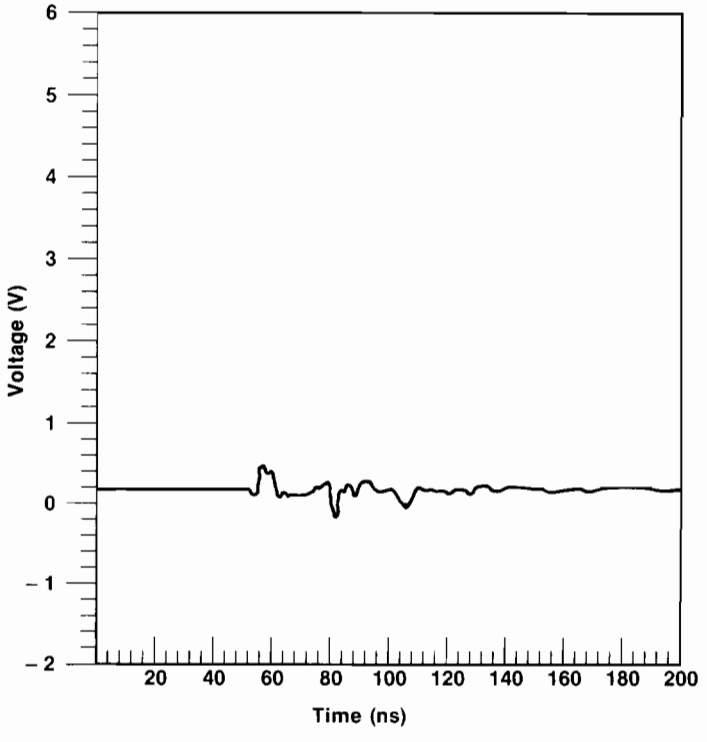


d. Active Driver

Figure 9-45 PC Board Crosstalk Through 36-Inch, 0.100 Trace, Forward with No Termination

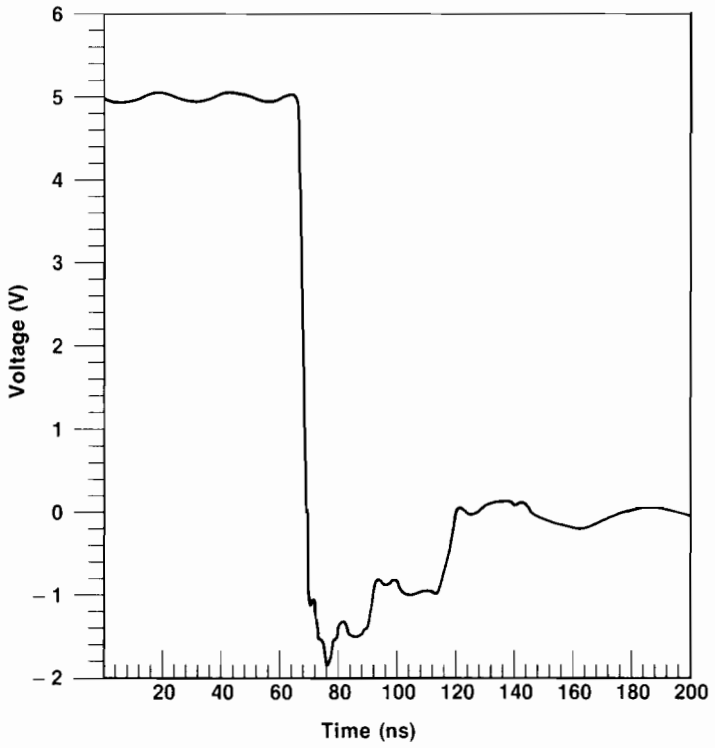


a. Adjacent Receiver

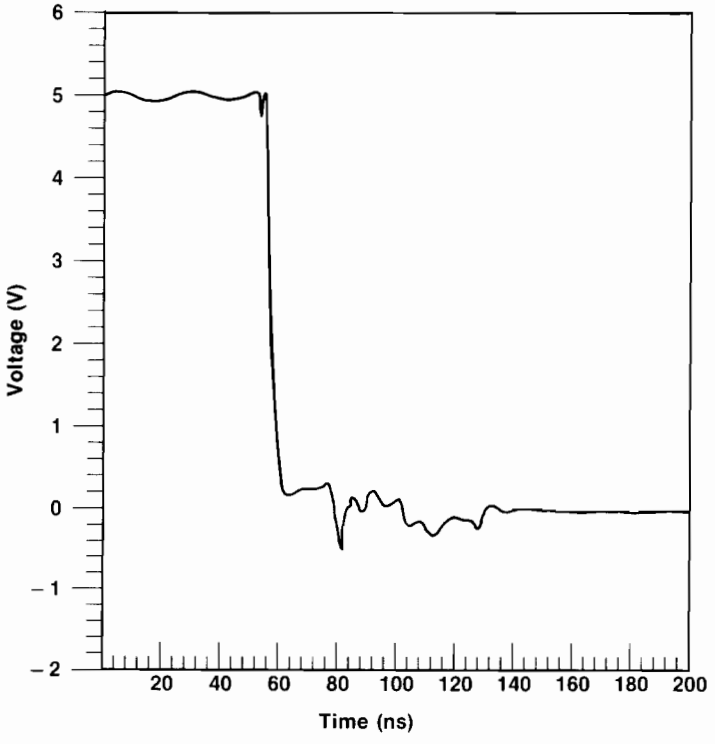


b. Adjacent Driver

Figure 9-45 (continued)

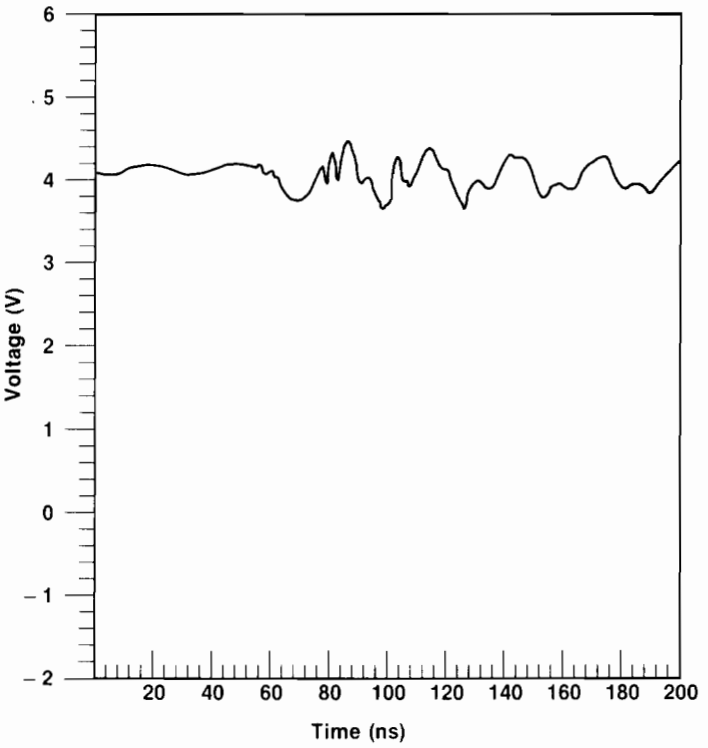


c. Active Receiver

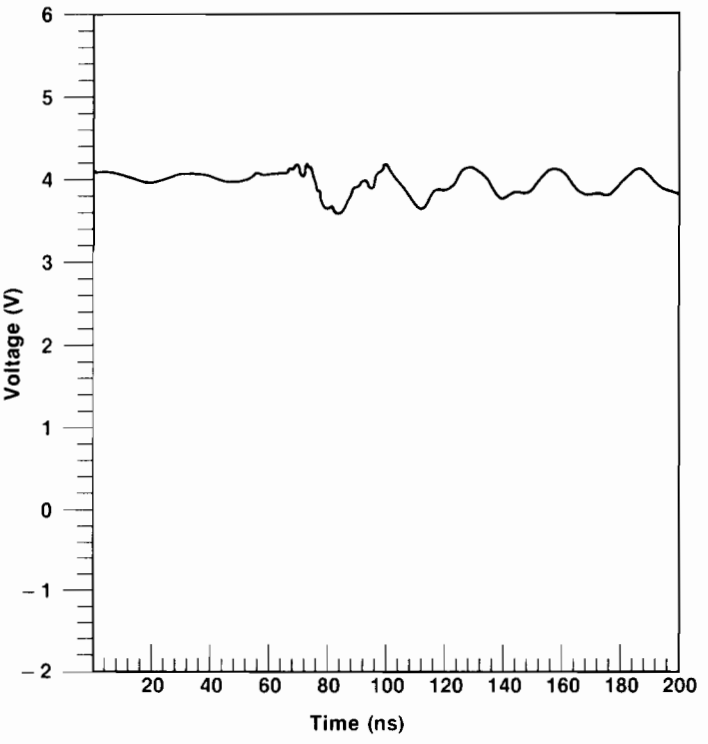


d. Active Driver

Figure 9-46 Reverse PC Board Crosstalk Through 34-Inch, 0.100 Trace Unterminated

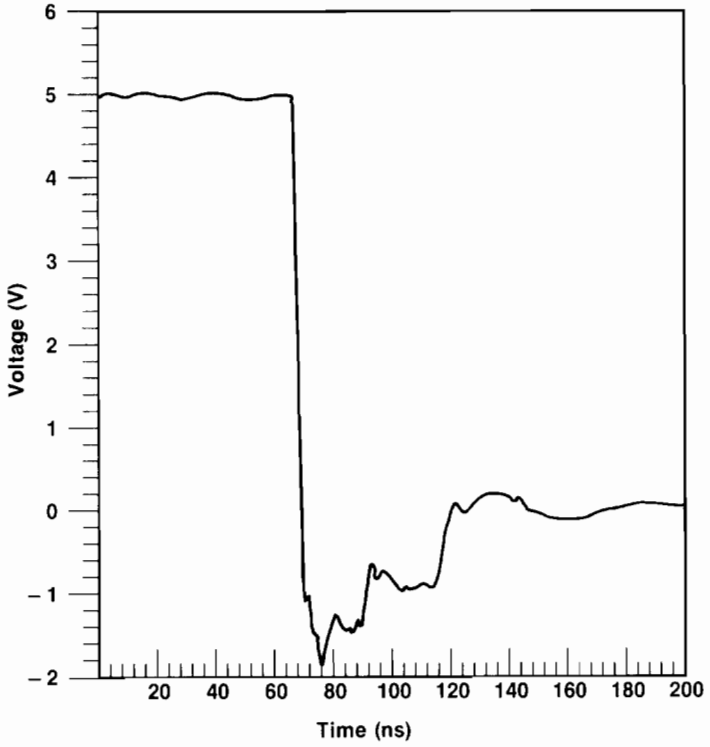


a. Adjacent Receiver

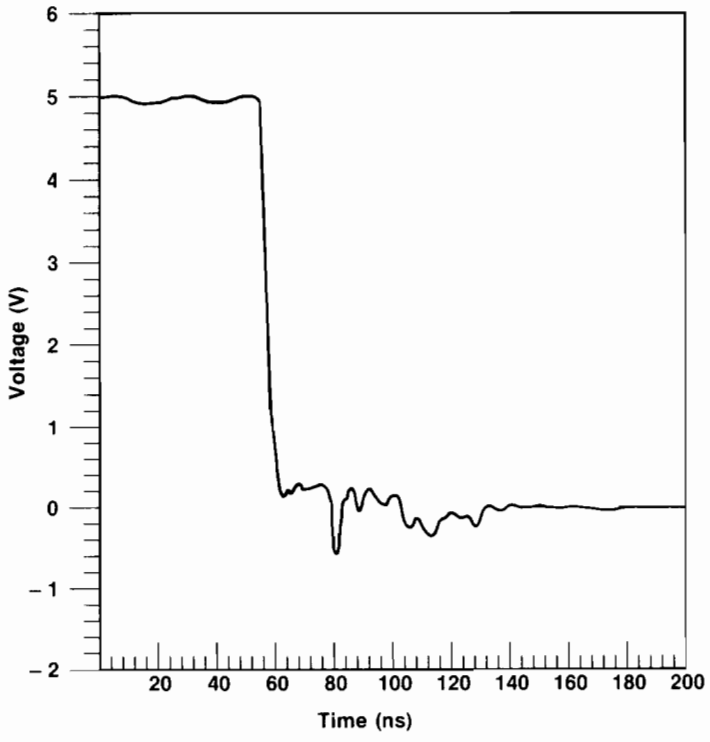


b. Adjacent Driver

Figure 9-46 (continued)

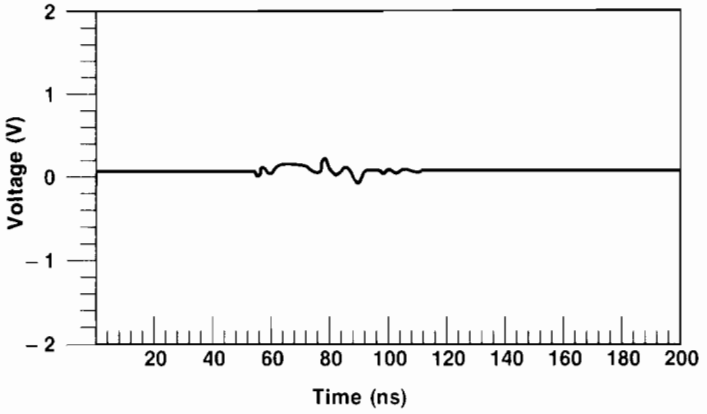


c. Active Receiver

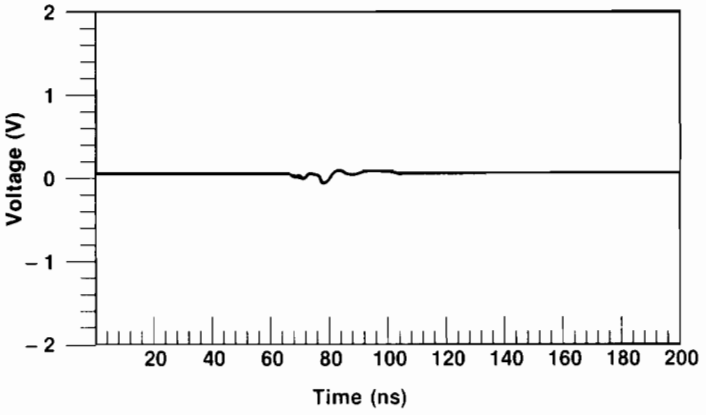


d. Active Driver

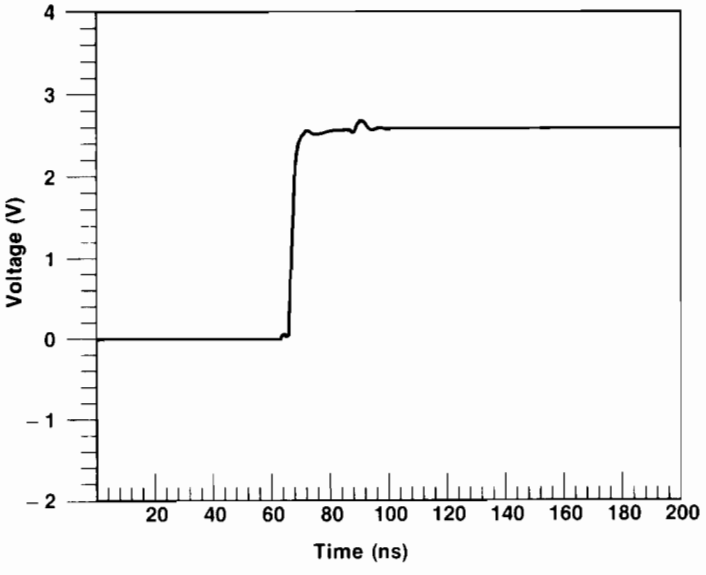
Figure 9-47 Reverse PC Board Crosstalk Through 34-Inch, 0.100 Trace Terminated



a. Adjacent Receiver

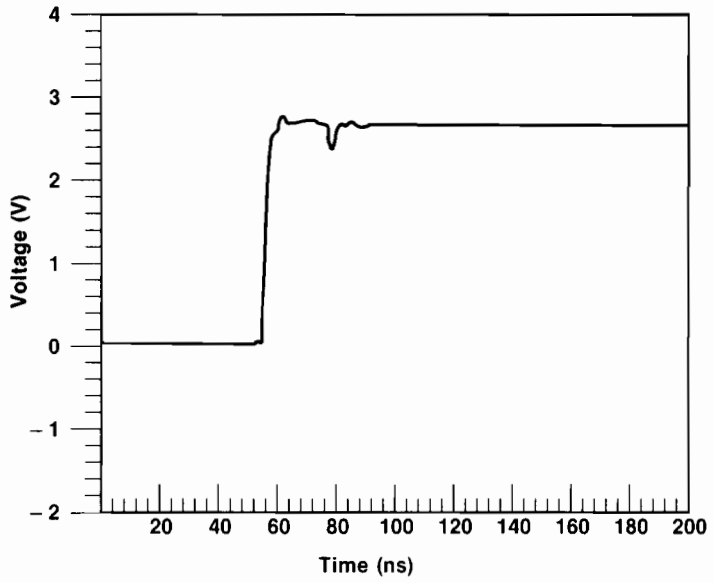


b. Adjacent Driver

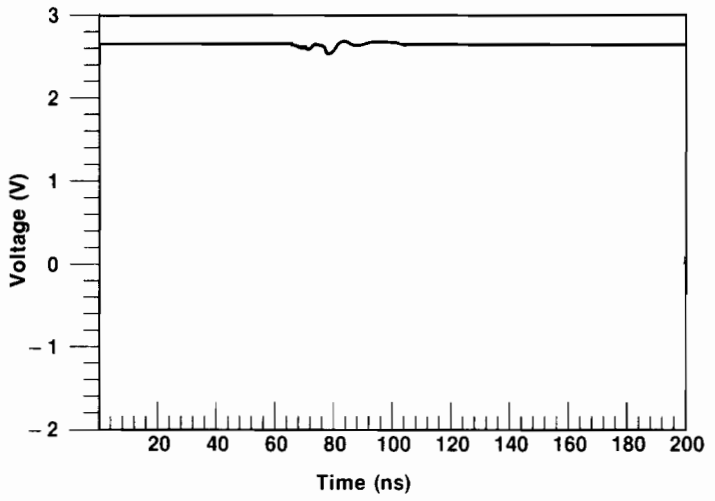


c. Active Receiver

Figure 9-47 (continued)

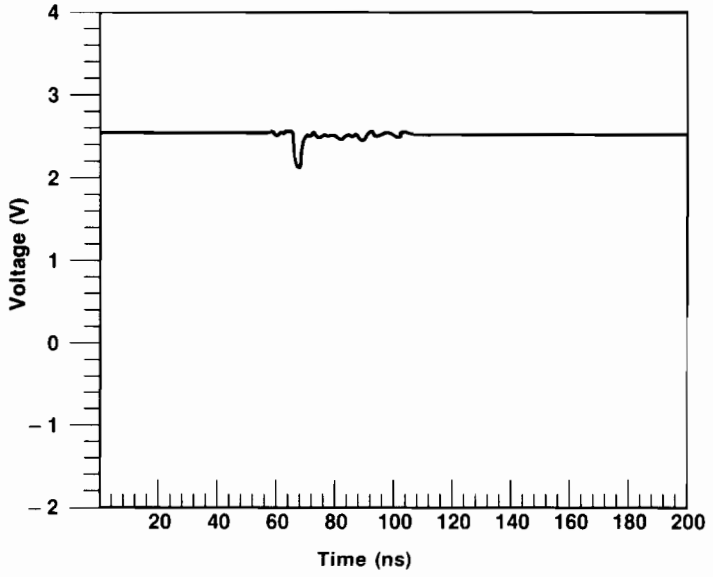


d. Active Driver

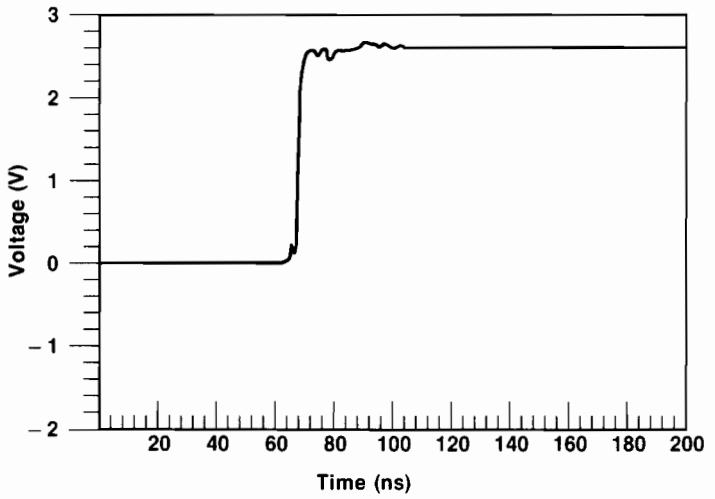


b. Adjacent Driver

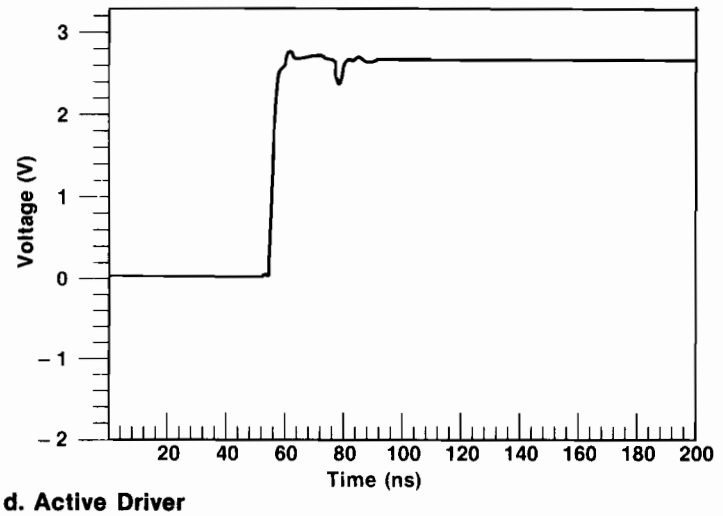
Figure 9-48 Forward PC Board Crosstalk Through 34-Inch Trace Terminated



a. Adjacent Receiver



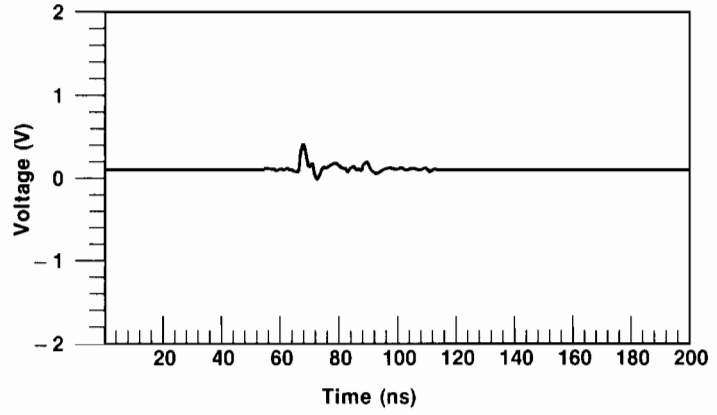
c. Active Receiver



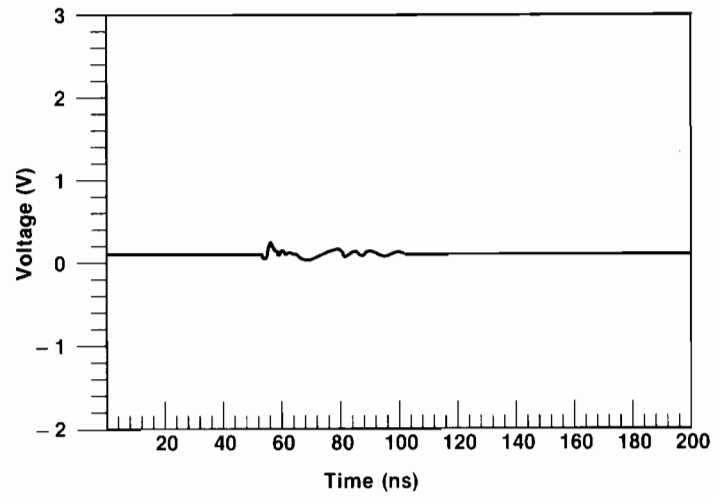
d. Active Driver

9

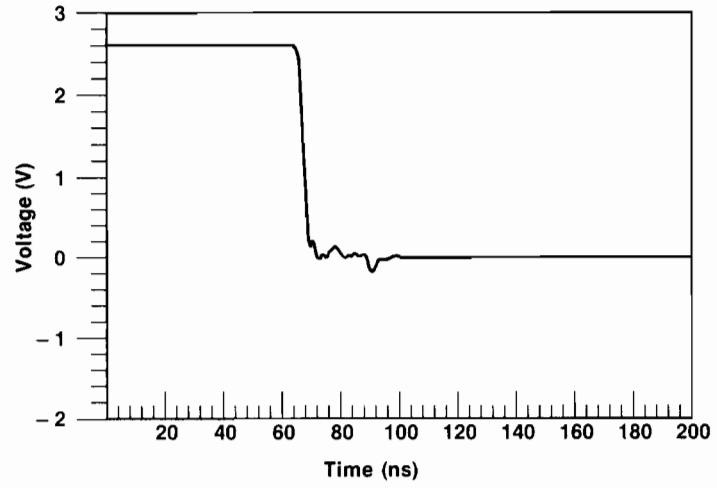
Figure 9-49 Forward PC Board Crosstalk Through 34-Inch Trace with Termination



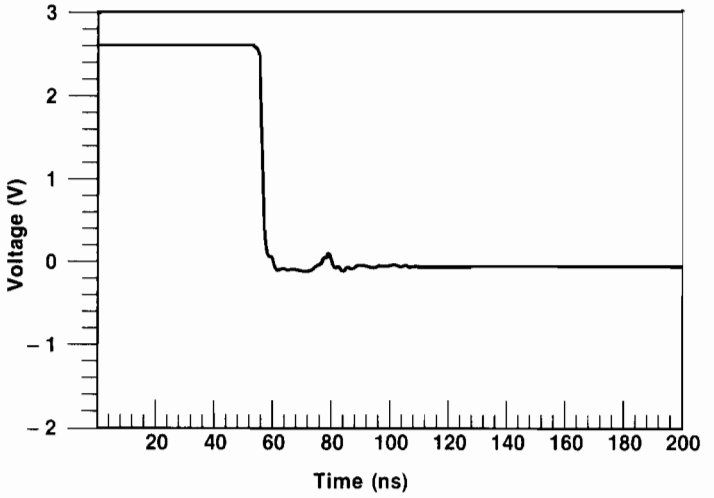
a. Adjacent Receiver



b. Adjacent Driver

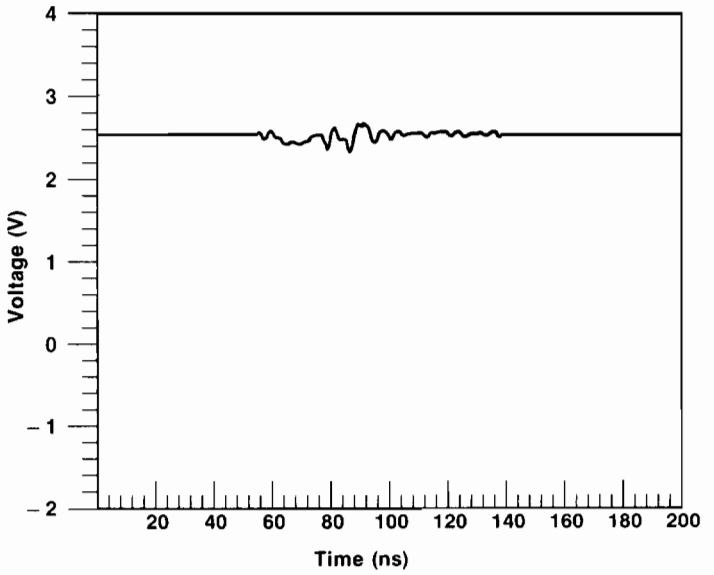


c. Active Receiver



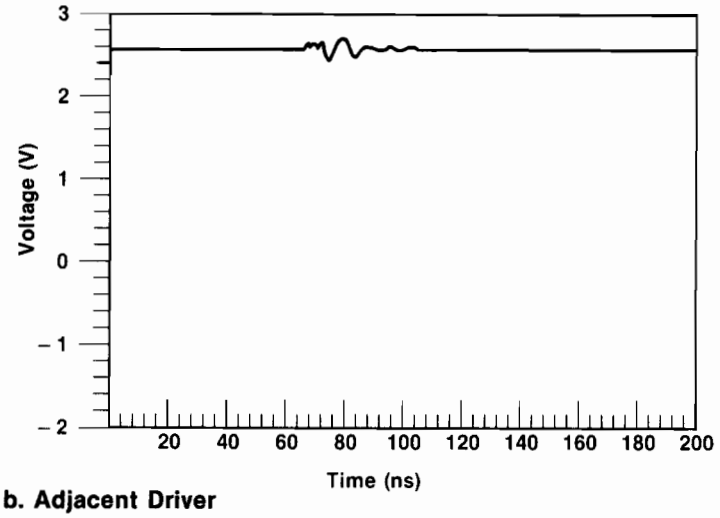
d. Active Driver

Figure 9-50 Reverse PC Board Crosstalk Through 34-Inch, 0.100 Trace Terminated

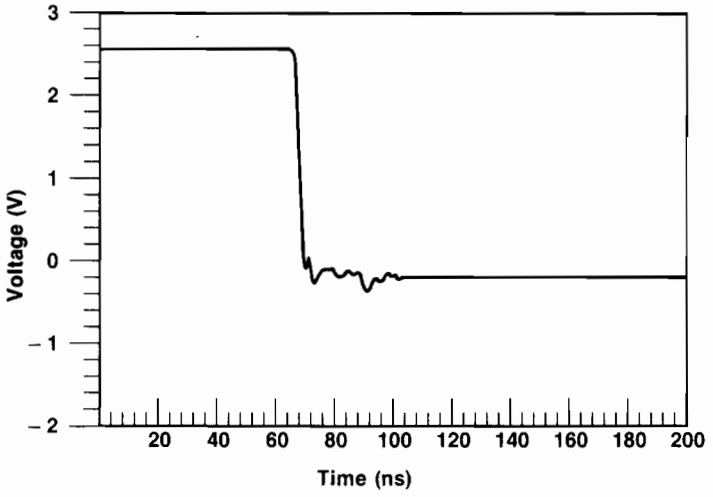


a. Adjacent Receiver

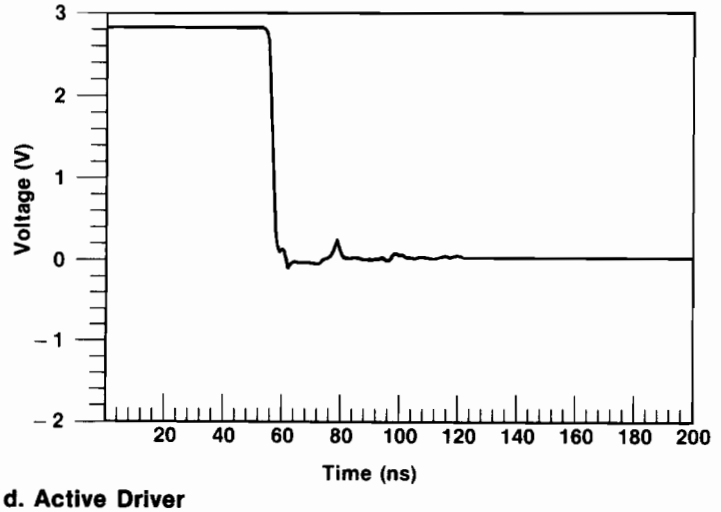
Figure 9-50 (continued)



b. Adjacent Driver



c. Active Receiver

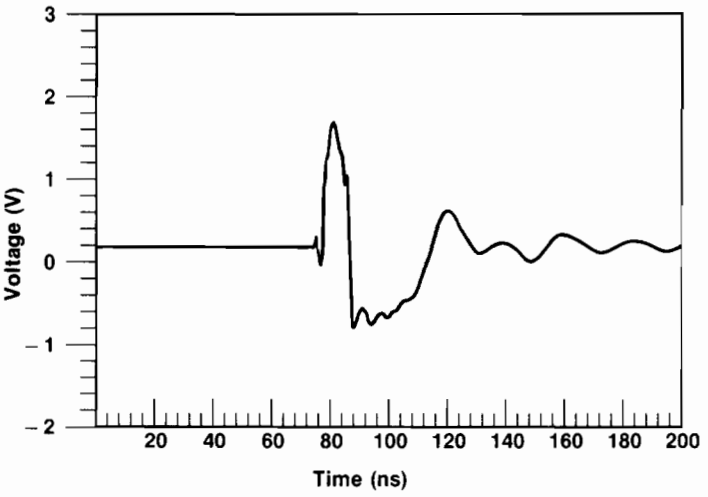


d. Active Driver

Crosstalk on Ribbon Cable

Crosstalk on ribbon cable shows no velocity difference effects—because the cable insulation is a homogeneous medium, all effects are due to mutual impedance. The results of tests on three foot sections of 160Ω ribbon cable are shown in Figures 9-51 through 9-58. From these it can be seen that the unterminated lines exhibit large amounts of ringing due to unterminated energy being transferred between lines. Note also that when the adjacent line is in a HIGH state a charge pump effect occurs, forcing the HIGH output above the V_{CC} supply and into a high impedance state with the output structure turned off and the input exhibiting only leakage currents. This high impedance state causes the current that has been induced into the line to reflect fro both ends and induce crosstalk back into the active line. This action will continue until damped by circuit resistance and leakages. The charge pump effect will leave the adjacent line at around 7V. If this line is then switched low, twice the normal energy is required to switch the line, thus almost doubling the crosstalk generated in the previous case. The terminated lines show the true magnitude of the crosstalk. Note that when the adjacent line is in the LOW state, the crosstalk will cause the driver output to turn off until clamped by the diode in the output structure.

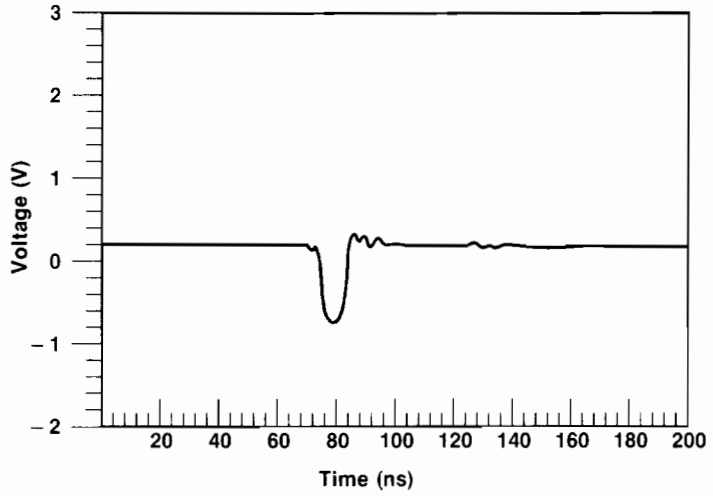
Figure 9-51 Forward Crosstalk Using FAST and 3-Foot Ribbon Cable, Unterminated



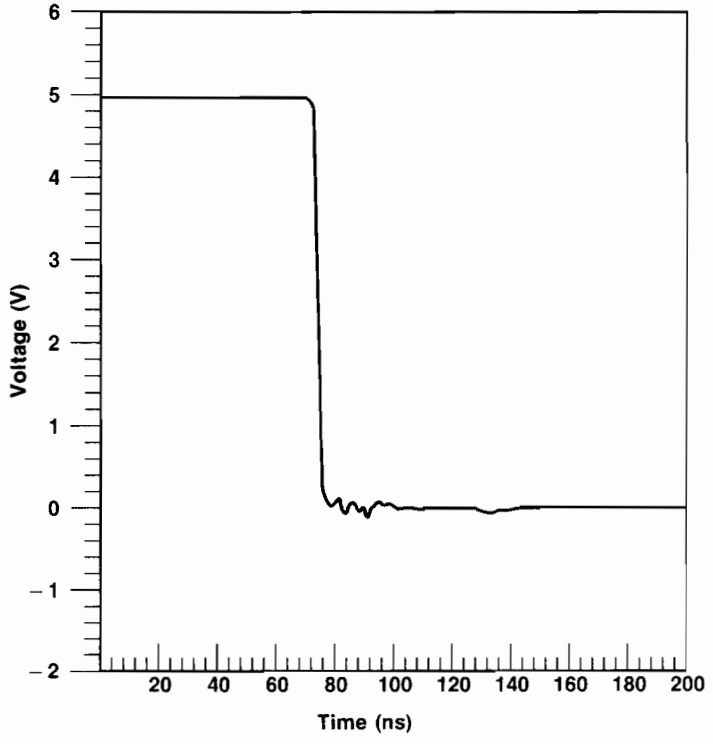
a. Adjacent Receiver

9

Figure 9-51 (continued)

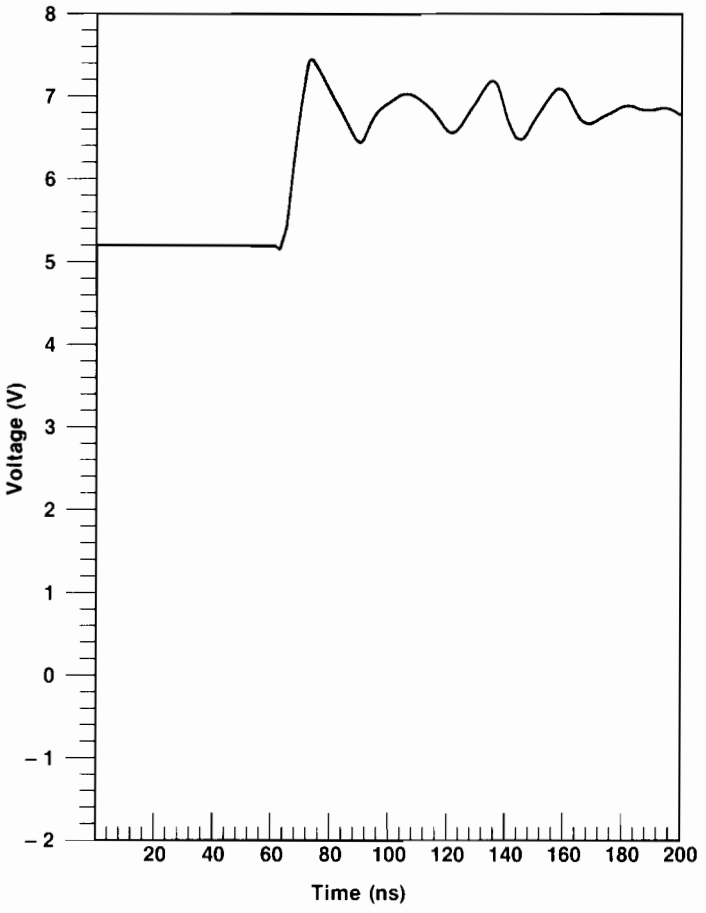


b. Adjacent Driver



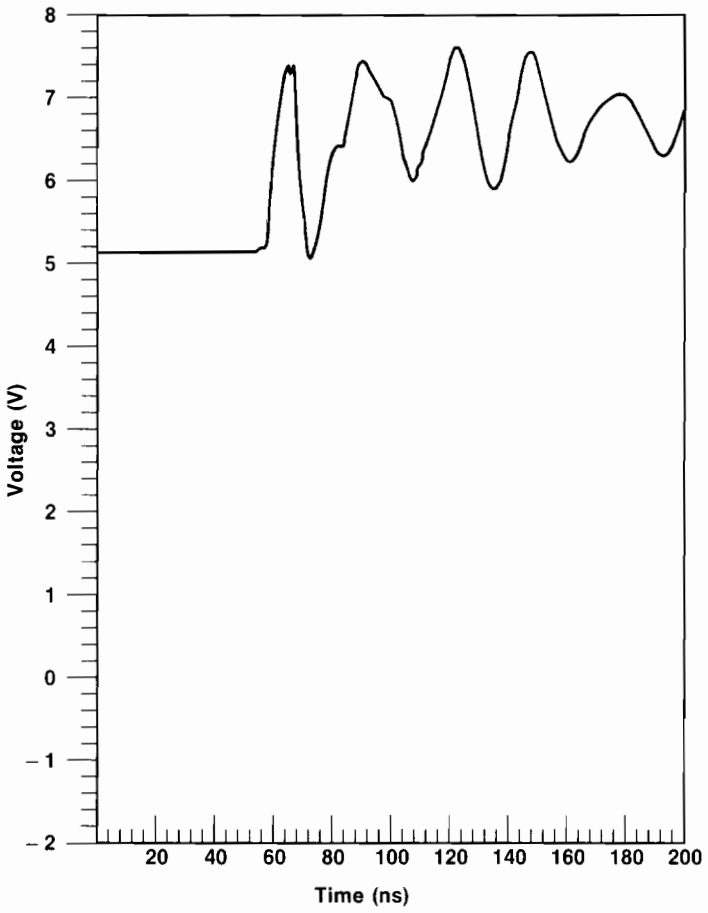
c. Active Driver

Figure 9-52 Forward Crosstalk Using FAST and 3-Foot, 2-Conductor Ribbon Cable, Unterminated

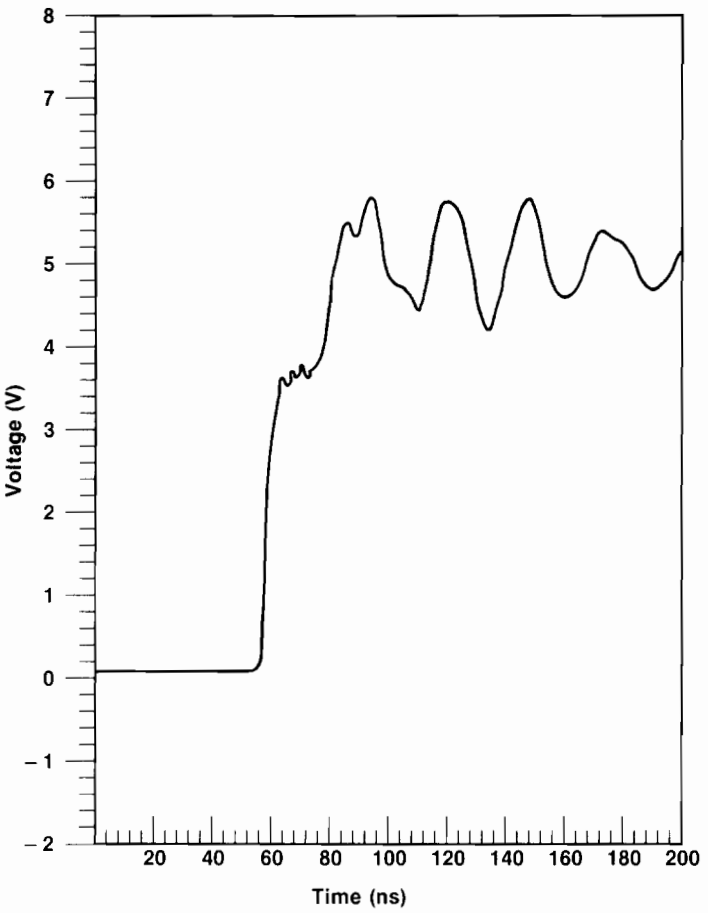


a. Adjacent Receiver

Figure 9-52 (continued)



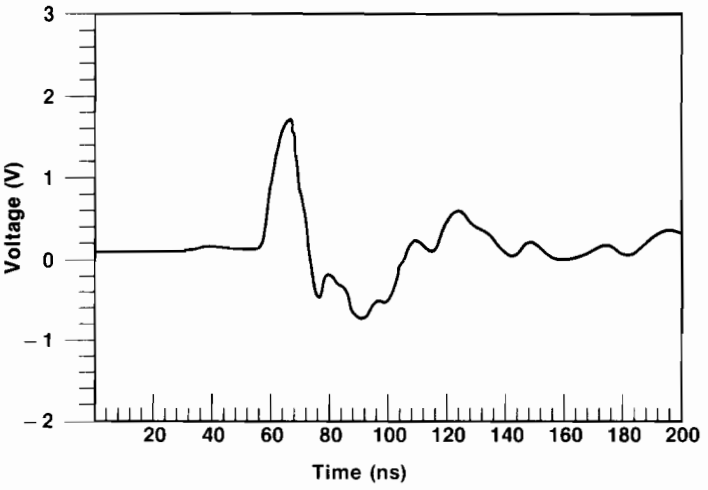
b. Adjacent Driver



c. Active Driver

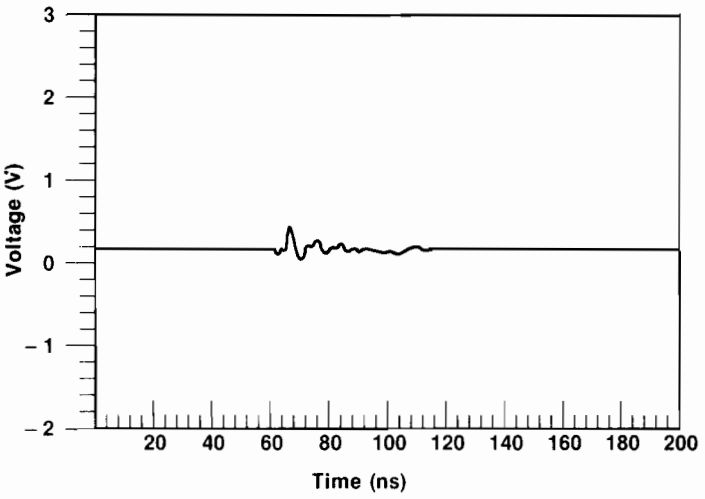
9

Figure 9-53 Reverse Crosstalk Using FAST and 3-Foot, 2-Conductor Ribbon Cable, Unterminated

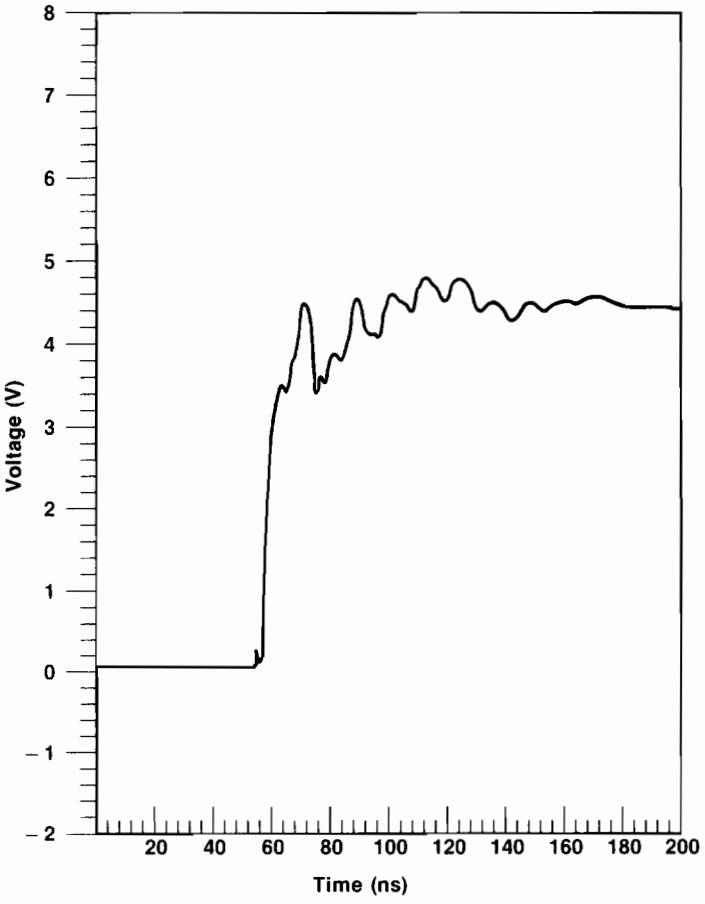


a. Adjacent Receiver

Figure 9-53 (continued)

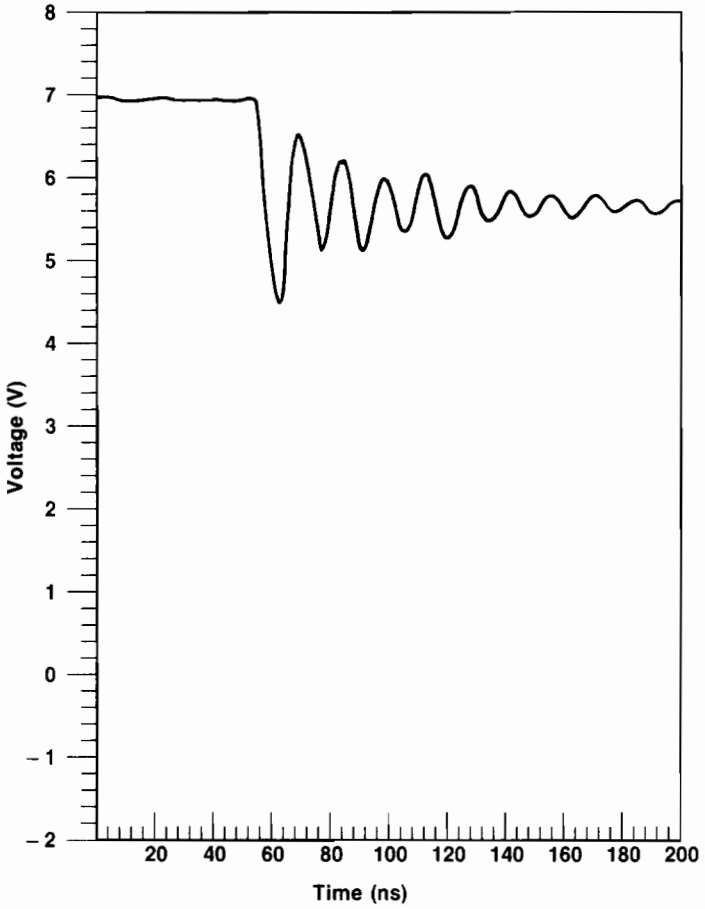


b. Adjacent Driver



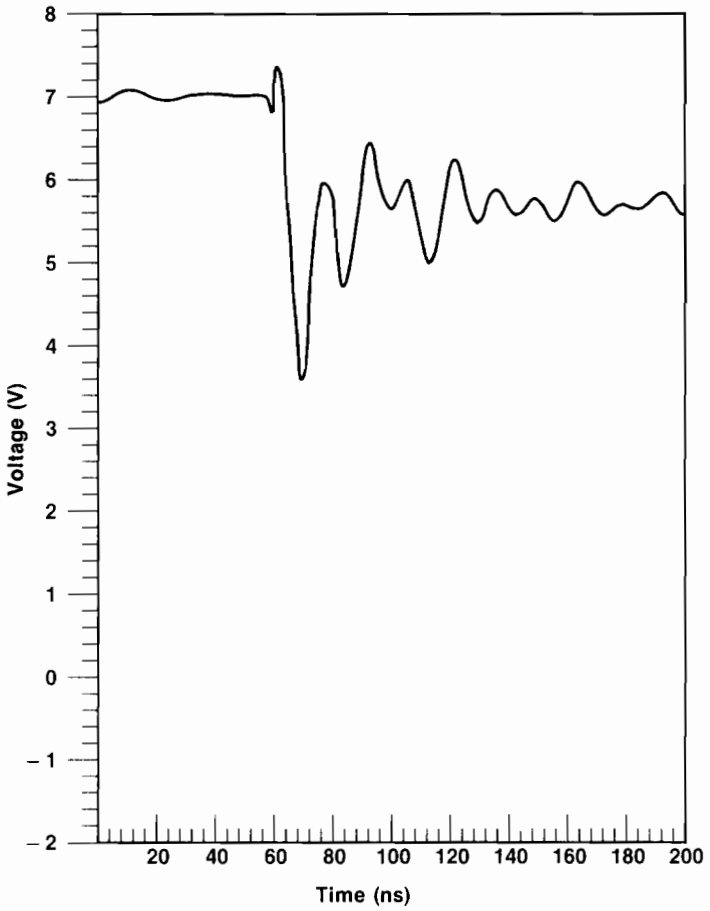
c. Active Driver

Figure 9-54 Reverse Crosstalk Using FAST and 3-Foot, 2-Conductor Ribbon Cable, Unterminated

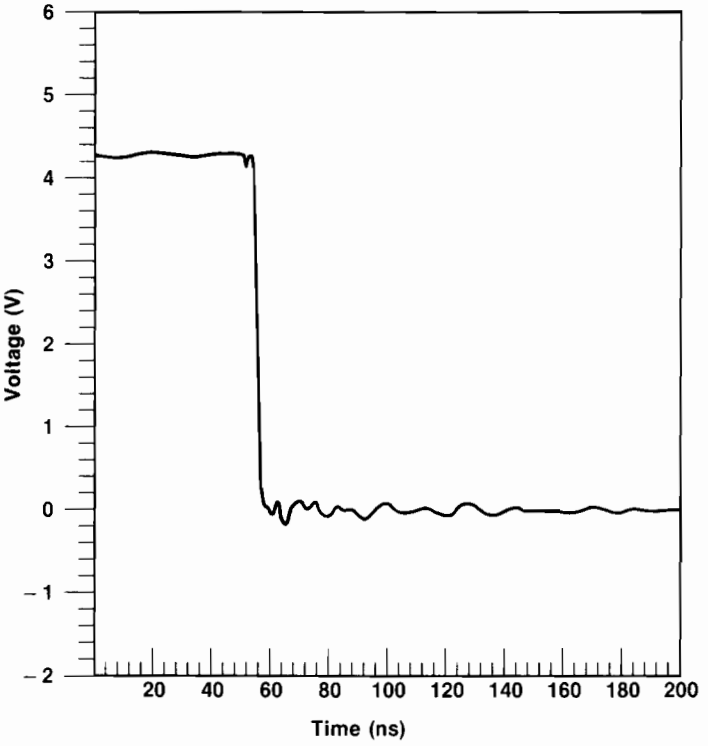


a. Adjacent Receiver

Figure 9-54 (continued)



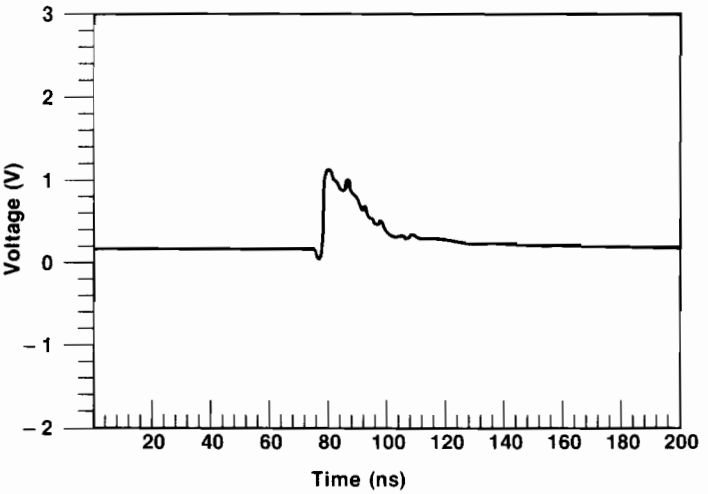
b. Adjacent Driver



c. Active Driver

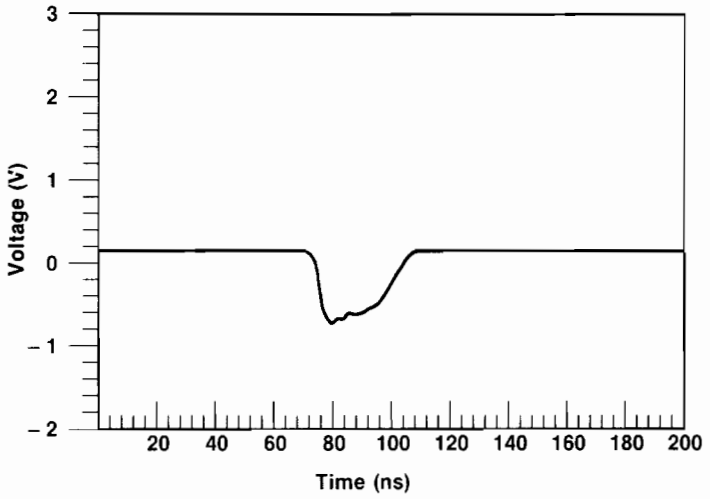
9

Figure 9-55 Forward Crosstalk Using FAST and 3-Foot, 2-Conductor Ribbon Cable, Terminated

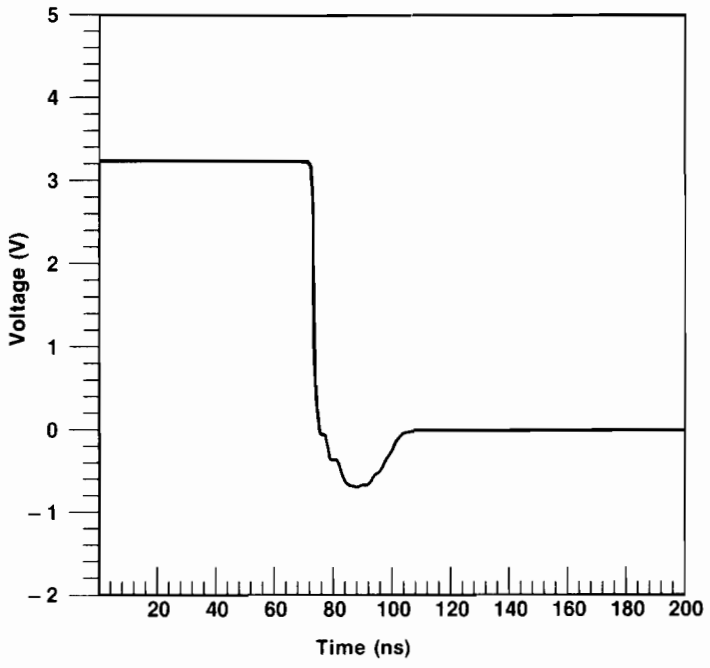


a. Adjacent Receiver

Figure 9-55 (continued)

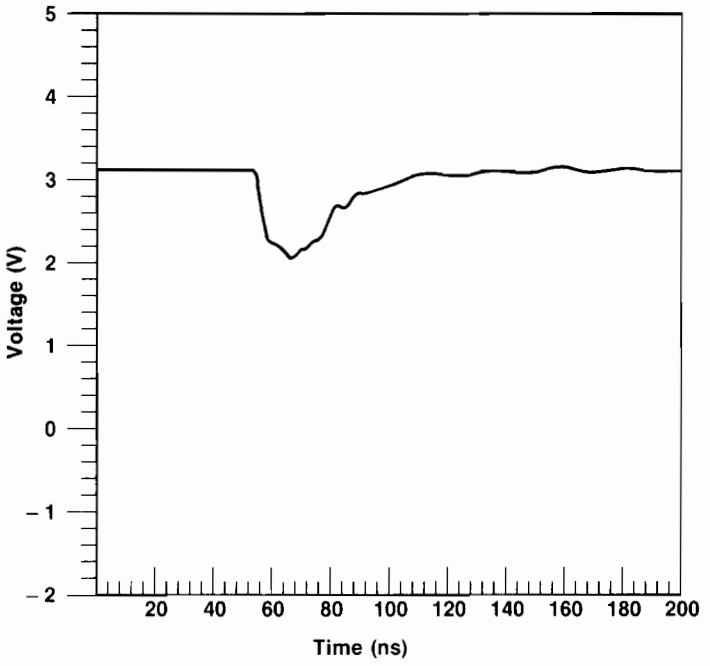


b. Adjacent Driver

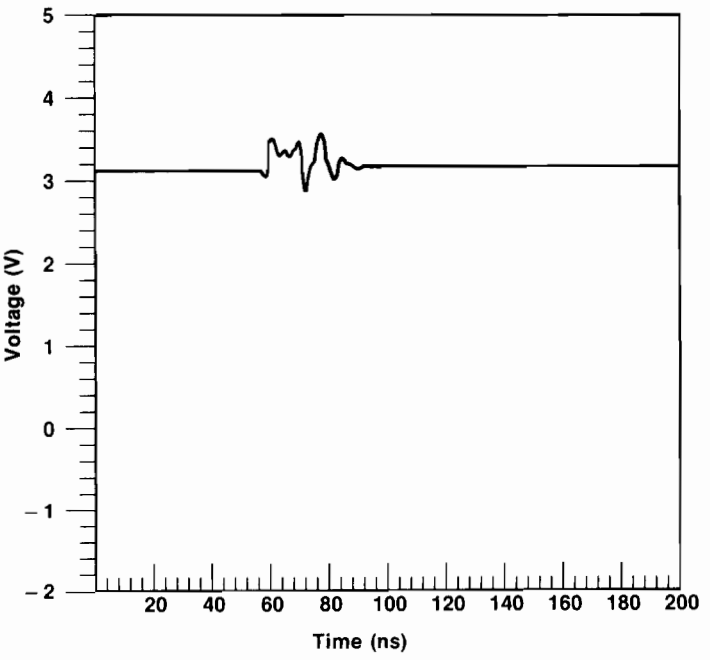


c. Active Driver

Figure 9-56 Reverse Crosstalk Using FAST and 3-Foot, 2-Conductor Ribbon Cable, Terminated

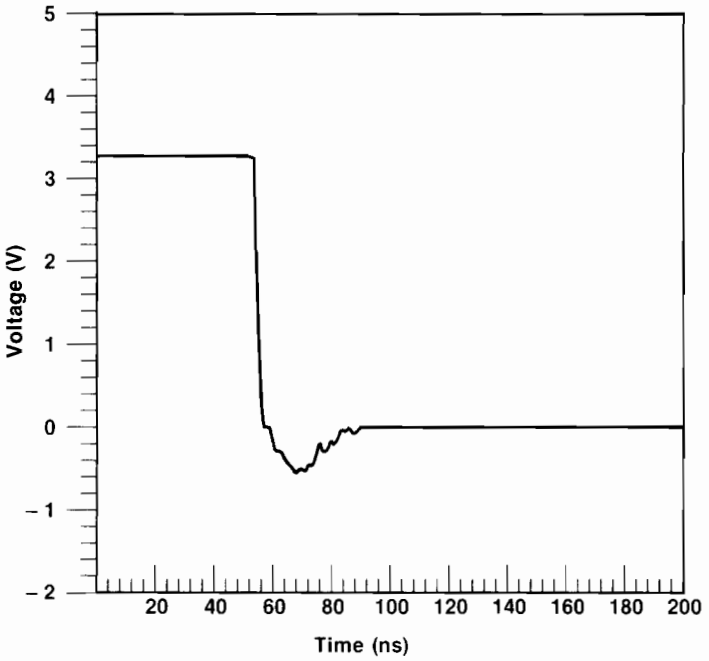


a. Adjacent Receiver

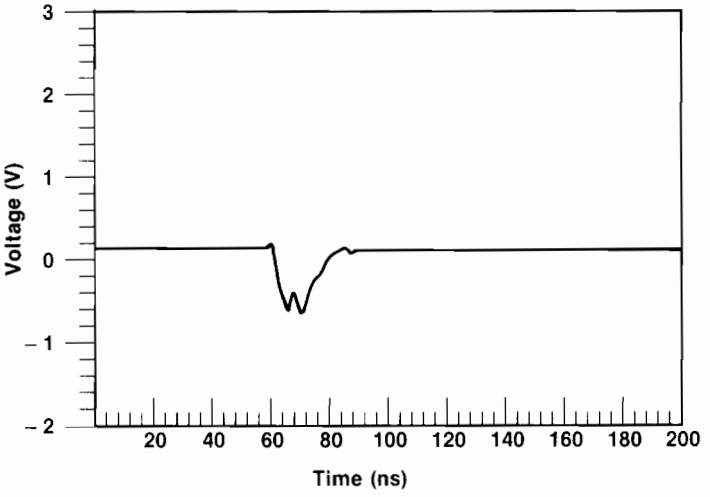


b. Adjacent Driver

Figure 9-56 (continued)

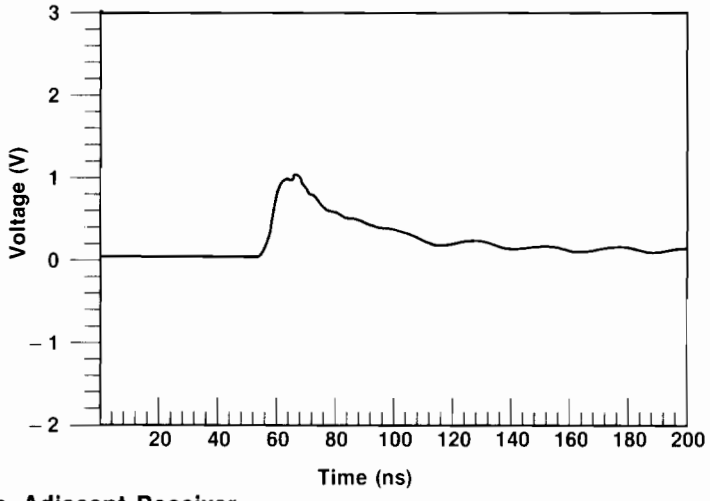


c. Active Driver

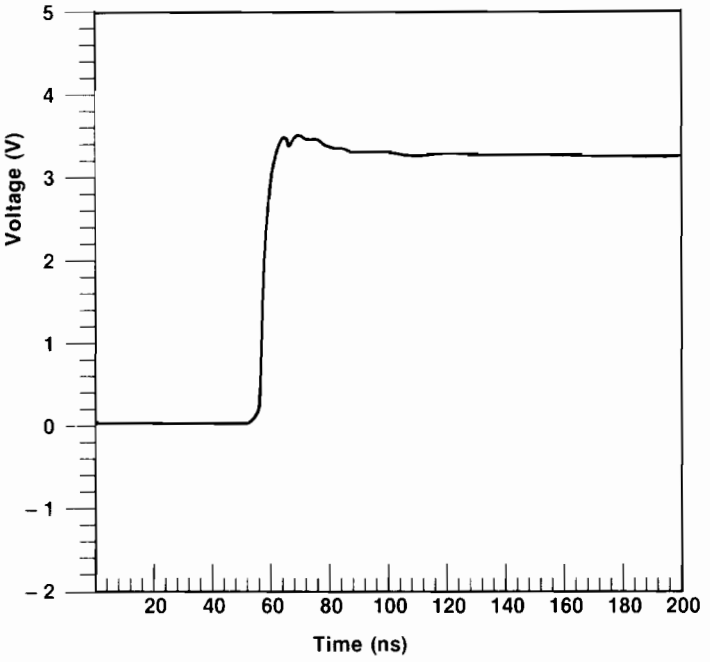


b. Adjacent Driver

Figure 9-57 Reverse Crosstalk Using FAST and 3-Foot, 2-Conductor Ribbon Cable, Terminated



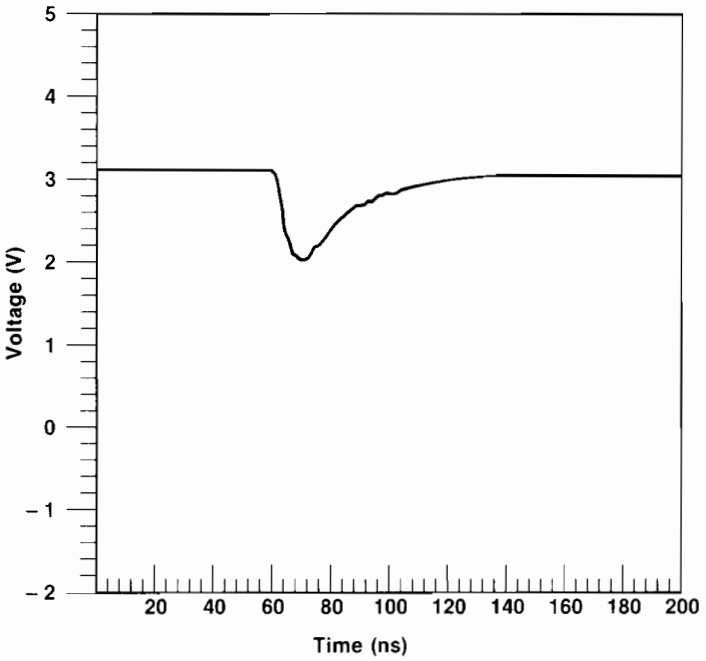
a. Adjacent Receiver



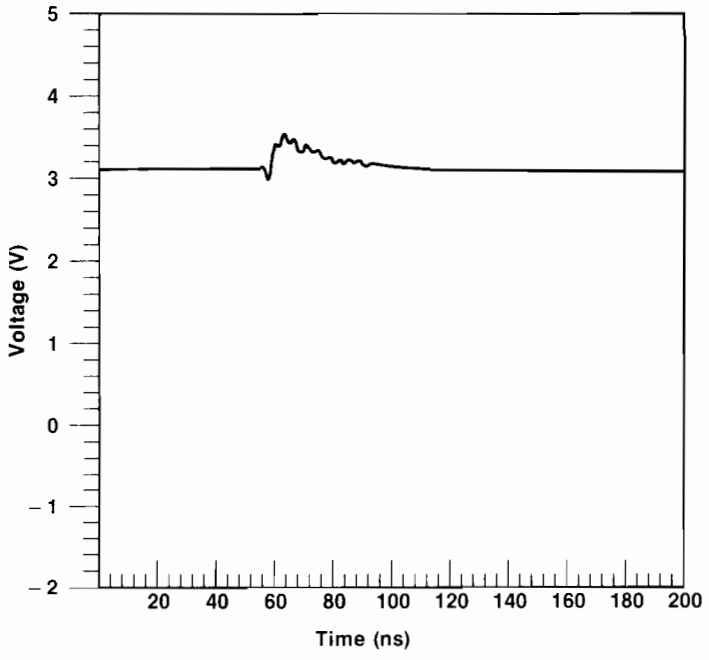
c. Active Driver

9

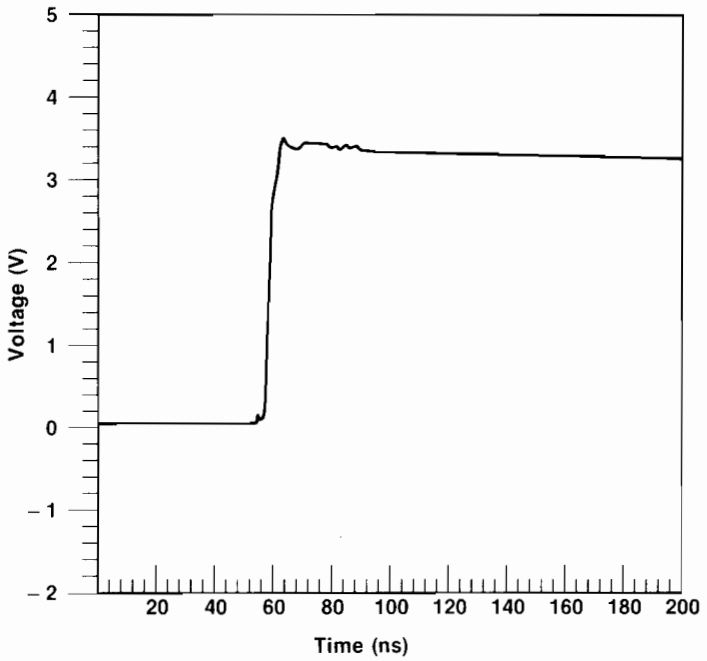
Figure 9-58 Forward Crosstalk Using FAST and 3-Foot, 2-Conductor Ribbon Cable, Terminated



a. Adjacent Receiver



b. Adjacent Driver



c. Active Driver

Recommendations

In order to minimize crosstalk it is necessary to consider the causes during the design of systems. Some preventative measures are as follows:

1. always use maximum allowable spacing between signal lines;
2. minimize spacing between signal lines and ground lines;
3. run ground strips alongside either the cross-talker or the cross-listener and between the two when possible;
4. in backplane and wire-wrap applications use twisted pair for sensitive functions such as clocks, asynchronous set or clear, asynchronous parallel load (especially leading to LS inputs); and
5. for ribbon or flat cabling make every other conductor a ground line.

In the case where systems or boards are already built and problems are encountered, some temporary or quick fixes may be utilized. They are:

1. with printed circuit boards, glue a source of ground, either a wire or a copper strip, alongside the cross-talker or cross-listener—preferably between them;

2. for the backplane or wire-wrap situation, spiral a ground wire around the talker to confine its electromagnetic field or around the listener in order to shield it, or do both;

3. try the split-resistor termination on the offending line (Figure 9-41);

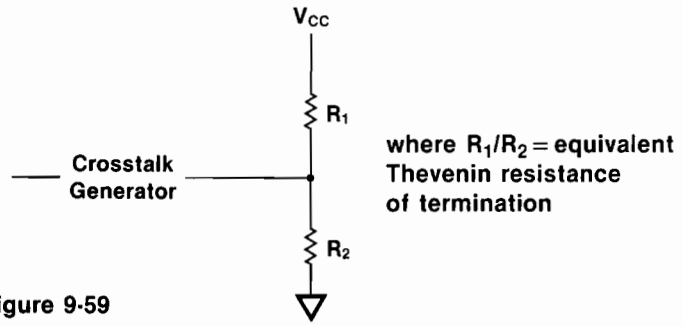


Figure 9-59

4. cut the offending crosstalk trace from the PC board and replace it with a wire. In this method reverse and forward crosstalk can be lessened. The line in this case may be lengthened, thereby increasing propagation delays, but a rerouting of the generating signal line may eliminate the crosstalk.

Termination can be used to reduce the effects of crosstalk. It can be seen here that a little termination is better than no termination.

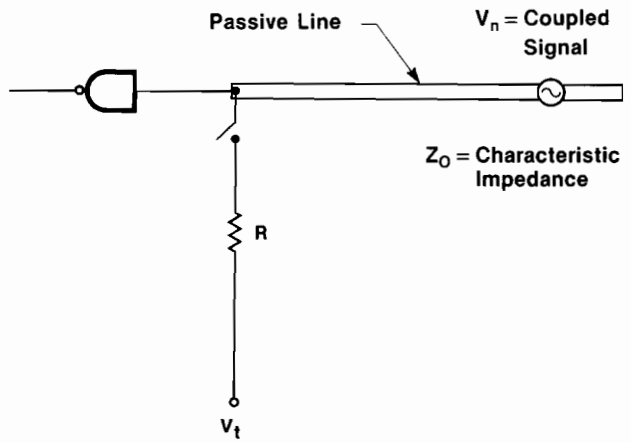


Figure 9-60

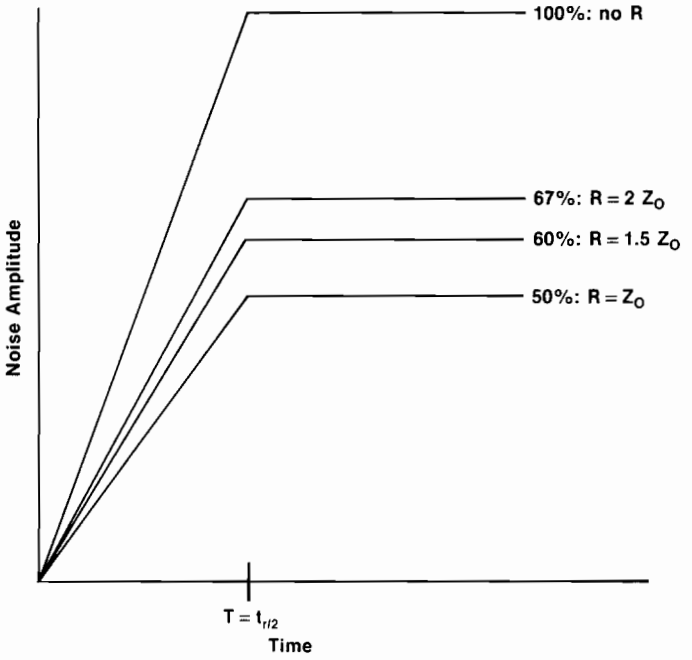


Figure 9-61

Summary

Trace proximity and coupled trace length are the two main factors which affect the amount of reverse crosstalk that occurs. Therefore, if coupled length is long, noise will be at a maximum. For short lengths, noise may appear only as a short spike which can cause difficulties and even system failures.

When two lines do not run between the same points but are in proximity over part of their length, signal propagation time (line delay) along this coupled length is T. If T is long compared to the rise of the signal on the active line, the crosstalk pulse has time to develop its full amplitude. The trailing edge of the noise pulse is caused by the reflection from the driven end of the passive line. When T is half the rise time, the reflection from the driven end of the passive line arrives and begins to pull the noise pulse down just as it reaches full amplitude. Any value of T less than half the rise time of the active signal will cause a reflection to arrive and oppose the noise pulse voltage before it can reach full amplitude. The noise will therefore be lower in amplitude.

The Capacitor

General Information

A capacitor is a component which is capable of storing electrical energy. It consists of conductive plates (electrodes) separated by insulating material which is called the dielectric. A typical formula for determining capacitance is:

$$C = \frac{0.224\ KA}{t}$$

(E9-45)

C = Capacitance (farads)
K = Dielectric constant (Vacuum = 1)
A = Area in square inches
t = Separation between plates in inches (thickness of dielectric)
0.224 = Conversion constant (0.0884 for metric system in cm)

Capacitance—The standard unit of capacitance is the farad. A capacitor has a capacitance of 1 farad when 1 coulomb charges it to 1 volt. One farad is a very large unit and most capacitors have values in the micro (10^{-6}), nano (10^{-9}), or pico farad (10^{-12}) level.

Dielectric Constant—In the formula for capacitance given above, the dielectric constant of a vacuum is arbitrarily chosen as the number 1. Dielectric constants of other materials are then compared to the dielectric constant of a vacuum. Dielectric constants of some typical materials are as follows:

Ruby Mica	7
Glass	10
Ceramic (class 1)	5-450
Ceramic (class 2)	200-12,000
Paper	2.5
Mylar	3
Polystyrene	2.6
Polycarbonate	3
Aluminum Oxide	7
Tantalum Oxide	11

Dielectric Thickness—Capacitance is indirectly proportional to the separation between electrodes. Lower voltage requirements mean thinner dielectrics and greater capacitance per volume.

Area—Capacitance is directly proportional to the area of the electrodes. Since the other variables in the equation are usually set by the performance desired, area is the easiest parameter to modify to obtain a specific capacitance within a material group.

Energy which can be stored in a capacitor is given by the formula:

$$E = 1/2\ CV^2$$

(E9-46)

E = Energy in joules (watts-sec)
V = Applied voltage
C = Capacitance in farads

A capacitor is a reactive component which reacts against a change in potential across it. This is shown by the equation for the linear charge of a capacitor:

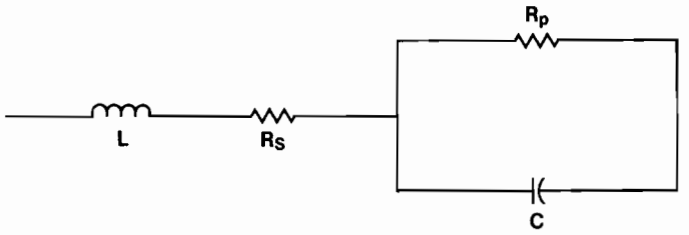
$$I_{ideal} = C\ \frac{dV}{dt}$$

(E9-47)

where
I = Current
C = Capacitance
dV/dt = Slope of voltage transition across capacitor

Thus an infinite current would be required to instantly change the potential across a capacitor, and the amount of current a capacitor can "sink" is given by the above equation.

A capacitor, as a practical device, exhibits not only capacitance but also resistance and inductance. A simplified schematic for the equivalent circuit is:



C = Capacitance L = Inductance
R_S = Series resistance R_p = Parallel resistance

Figure 9-62

All the factors shown above are important in the application of capacitors. The inductance determines the usefulness of the capacitor at high frequency, the parallel resistance affects performance in timing and coupling circuits (normally expressed as Insulation Resistance) and the series resistance is a measure of the loss in the capacitor and is a major factor in Power Factor and/or Dissipation Factor.

Since the insulation resistance (R_p) is normally very high, the total impedance of a capacitor is:

$$Z = \sqrt{R_S^2 + (X_C - X_L)^2}$$

(E9-48)

where
Z = Total impedance
R_S = Series resistance
X_C = Capacitive reactance = $1/2\pi fc$
X_L = Inductive reactance = $2\pi fL$

The variation of a capacitor's impedance with frequency determines its effectiveness in many applications.

Power Factor and Dissipation Factor are often confused since they are both measures of the loss in a capacitor under AC application and are often almost identical in value. In a "perfect" capacitor the current in the capacitor will lead the voltage by 90°.

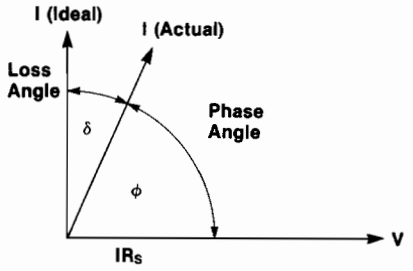


Figure 9-63

In practice the current leads the voltage by some other phase angle due to the series resistance R_s. The complement of this angle is called the loss angle and:
Power Factor (PF) = cos ϕ or sine δ
Dissipation Factor (DF) = tan δ

For small values of δ the tan and sine are essentially equal which has led to the common interchangeability of the two terms in the industry.

The term ESR or Equivalent Series Resistance combines all losses, both series and parallel, in a capacitor at a given frequency so that the equivalent circuit is reduced to a simple R-C series connection.

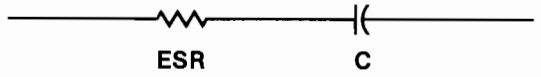


Figure 9-64

$$\text{Dissipation Factor} = \frac{ESR}{X_c} = (2\pi fc)(ESR)$$

(E9-49)

The DF/PF of a capacitor tells what percent of the apparent power input will turn to heat in the capacitor. The watts loss is:

$$\text{Watts Loss} = (2\pi fcE^2)(DF)$$

(E9-50)

Very low values of dissipation factor are expressed as their reciprocal for convenience. These are called the "Q," or Quality factor of capacitors.

Insulation Resistance is the resistance measured across the terminals of a capacitor and consists principally of the parallel resistance R_p shown in the equivalent circuit. As capacitance values and hence the area of dielectric increases, the IR decreases. The product (C x IR or RC) is often specified in ohm farads or commonly megohm microfarads.

Dielectric Strength is an expression of the ability of a material to withstand an electrical stress. Although dielectric strength is ordinarily expressed in volts, it is actually dependent on the thickness of the dielectric and thus is also more generically a function of volts/mil.

Other specialized factors which may be of interest to the user, especially in high voltage applications, are corona and dielectric absorption.

The phenomenon of Dielectric Absorption is exhibited in the following manner: charging current from a steady unidirectional source continues to flow at a gradually decreasing rate into a capacitor of negligible series resistance for some time after the almost instantaneous charge is completed. A steady value proportional to the capacitor parallel resistance is finally reached. The additional charge apparently is absorbed by the dielectric. Conversely, a capacitor does not discharge instantaneously upon application of a short circuit, but drains gradually after the capacitance proper has been discharged. It is common practice to measure the dielectric absorption by determining the "reappearing voltage" which develops across a capacitor at some point in time after it has been fully discharged under short circuit conditions.

Corona is the ionization of air or other vapors which causes them to conduct current. It is especially prevalent in high voltage units but can occur with low

voltages as well where high voltage gradients occur. The energy discharged degrades the performance of the capacitor and can in time cause catastrophic failures.

The usual characteristics that are specified for a capacitor include Capacitance, Dissipation Factor or ESR, Insulation Resistance or Leakage Current and Dielectric Strength. The electrical and environmental parameters that are of most interest with respect to these four basic measurements are temperature, voltage and test frequency. The reference temperature for most capacitor measurements is 25°C. Voltage is dependent on the rating applied by the manufacturer and the test frequency typically depends on the class of product.

As the ambient temperature changes, the dielectric constant and hence the capacitance of many capacitors changes. In general, when the dielectric constant is lower, materials tend to change capacitance less with temperature or with relatively predictable changes that are linear with temperature. High dielectric constant materials tend to have capacitance changes that are non-linear and expressed as percent capacitance change over a temperature range. Increasing temperature usually reduces Insulation Resistance, increases Leakage Current and Power Factor/Dissipation Factor and reduces the voltage rating of the part. Some ceramic capacitors actually exhibit a decrease in DF with increasing temperature. Conversely, reducing temperature normally improves most characteristics.

The effects of applied voltage on capacitors are a prime consideration in use. Capacitance and other parameter changes occur under both AC and DC applied voltages. Even those cases where voltage application does not change the parametric characteristics of a capacitor, the level of voltage applied will determine the life expectancy of the capacitor.

Frequency is the third factor which is of great concern in the application of capacitors. This is an area that is often overlooked by designers. Earlier an equivalent circuit was given for a capacitor. Inductance which is caused by the leads and the electrodes was depicted. As the frequency applied to the capacitor increases it eventually passes through self-resonance and becomes inductive with gradually increasing impedance. Even though a capacitor is beyond the self-resonant point it still blocks DC and has a low impedance and thus is useful in bypass, coupling and many other applications. Care should be taken in feedback, tuning, phase shift and such applications.

Ceramic Capacitors

Ceramic capacitors are the most widely used capacitors. They come in an extremely wide range of mechanical configurations and electrical characteristics. The common mechanical variations are discs, tubulars, feed throughs and monolithics. A 0.01μF disc is about 1/2 inch in diameter while the 0.01μF monolithic chip capacitor is only 0.050" x 0.075" x 0.030". Electrically, ceramic capacitors are broken into two classes. Class 1 ceramic dielectrics are also called temperature compensating ceramics and feature zero TC and other predictable and relatively linear TC bodies. The insulation resistance is high, the losses are low and the parts are essentially unaffected by voltage or frequency and are usually used for tuned circuits, timing applications and other precision circuits.

Where Class 1 ceramics are completely predictable, Class 2 general purpose ceramics are full of surprises for the unsuspecting engineer. Not only does capacitance change with temperature but the "high K" units which are so enticingly small in size may lose 90% of their room temperature capacitance at −55°C. Further care must be exercised when voltage is applied, particularly with monolithic capacitors with their thin dielectrics. AC voltage caused the capacitance to increase and DC voltage causes a capacitance loss. A change in frequency also changes capacitance and DF.

The fact that more ceramic Class 2 capacitors are used than all other types combined proves that the variability of characteristics not only can be overcome by wise selection but can in many cases be an advantage. Considerably more detailed information is given below and a number of articles and booklets are also available on this subject.

The ceramic capacitor is defined as a capacitor manufactured from metallic oxides, sintered at a high temperature. As a general rule, the electrical ceramics used in capacitors are based on complex titanate compounds, principally barium titanate, rare earth titanates, calcium titanates, sodium titanate, etc. Occasionally other materials, such as lead niobiate, may be used.

From a mechanical point of view, ceramic capacitors are manufactured by two basic techniques. One method involves pressing or extruding the ceramic material, firing (sintering) the ceramic and subsequently applying electrodes (typically with silver materials) which are fired onto the ceramic at lower temperatures after the

maturation of the ceramic. This is the method employed in the fabrication of single layer devices. The most common form of single layer capacitors is disc capacitors with radial leads or tubular capacitors which are available with axial leads, radial leads or in feed-through form with both bolt and eyelet types being common. There are specialized versions of pressed ceramic capacitors, such as high voltage cartwheels and double cup high voltage units. These and other types may be considered as jumbo size disc pressed units.

The second method of fabricating ceramic capacitors evolved in recent years as a result of the demand for lower voltages and smaller sizes consistent with the advent of semiconductor usage. The miniaturization in the ceramic capacitor area was made possible through the manufacture of monolithic types of ceramic capacitors. These capacitors are manufactured by mixing the ceramic powder in an organic binder (slurry) and casting it by one technique or another into thin layers typically ranging from about 3 mils in thickness down to 1 mil or thinner.

Metal electrodes are deposited onto the green ceramic layers which are then stacked to form a laminated structure. The metal electrodes are arranged so that their terminations alternate from one edge of the capacitor to another. Upon sintering at high temperature the part becomes a monolithic block which can provide extremely high capacitance values in small mechanical volumes. Figure 9-65 shows a pictorial view of a monolithic ceramic capacitor.

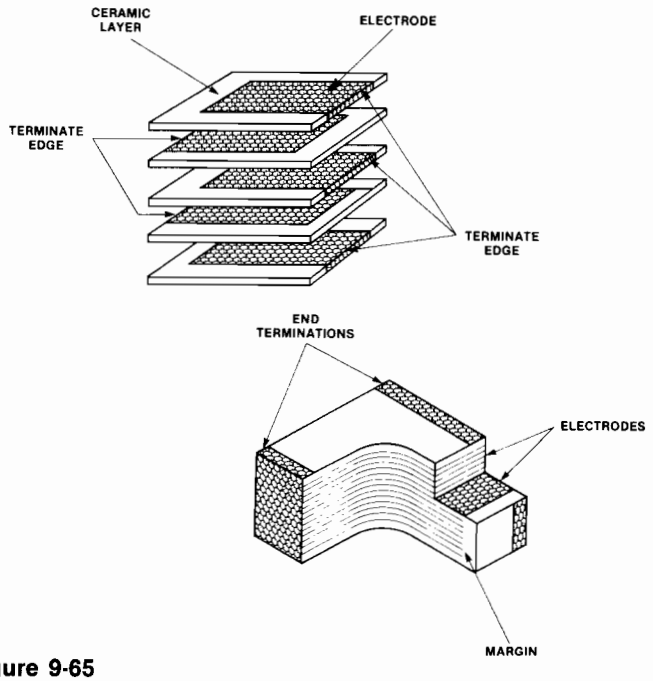


Figure 9-65

While pressed and extruded ceramic capacitors are in general low cost and provide limited capacitance values, monolithic units are typically smaller in size, feature excellent high frequency characteristics because of the small size and provide considerably higher capacitance values with low voltage ratings.

Ceramic capacitors are available in a tremendous variety of characteristics. Electronic Industries Association (EIA) and the military have established categories to help divide the basic characteristics into more easily specified classed. The basic industry specifications for ceramic capacitors is EIA specification RS-198 and as noted in the general section it specifies temperature compensating capacitors as Class 1 capacitors. These are specified by the military under specification MIL-C-20. General purpose capacitors with non-linear temperature coefficients are called Class 2 capacitors by EIA and are specified by the military under MIL-C-11015 and MIL-C-39014. EIA specifications further include a Class 3 category which is defined as reduced titanates.

Class 1 or temperature compensating capacitors are usually made from mixtures of titanates where barium titanate is normally not a major part of the mix. They have predictable temperature coefficients and in general do not have any aging characteristic. Thus they operate in a manner similar to mica capacitors except for the TC which is controllable. Normally the TCs of Class 1 capacitors are deemed to run between P100 and N750. Class 1 extended temperature compensating capacitors are also manufactured in TCs from negative 1400 through negative 5600, however, these may start developing a slight aging characteristic and voltage susceptibility.

Most TC formulations are available in pressed and extruded construction while only NPO (zero TC) is provided by most manufacturers in monolithic construction. NPO ceramics in monolithic capacitors are available in high enough values to cover most applications requiring extreme stability. With the exception of some NPO capacitors, almost all temperature compensating capacitors have a TC curve which is a true curve and not a straight line. The TC tends to become more negative at the cold end than it is from the 25°C reference to +85°C. Both EIA specification RS-198 and military specification MIL-C-20 contain information about curvature. This information is contained in Table 9-2. These charts are based on industry accepted standard TC values.

Table 9-2

TC TOLERANCES (1)

Capacitance in pf	NPO	N030	N080	N150	N220	N330	N470	N750	N1500	N2200
					−55°C to +25°C IN PPM/°C					
10 and over	+30 −75	+30 −80	+30 −90	+30 −105	+30 −120	+60 −180	+60 −210	+120 −340	+250 −670	+500 −1100
					+25°C to +85°C IN PPM/°C					
10 and over	±30	±30	±30	±30	±30	±60	±60	±120	±250	±500
Closest MIL−C−20D Equivalent	CG	HG	LG	PG	RG	SH	TH	UJ	NONE	NONE
EIA Desig.	C0G	S1G	U1G	P2G	R2G	S2H	T2H	U2J	P3K	R3L

(1)Table 9-2 indicates the tolerance available on specific temperature characteristics. It may be noted that limits are established on the basis of measurements at +25°C and +85°C and that TC becomes more negative at low temperature. Wider tolerances are required on low capacitance values because of the effects of stray capacitance.

General purpose ceramic capacitors are called Class 2 capacitors and have become extremely popular because of the high capacitance values available in very small size. Class 2 capacitors are "ferro electric" and vary in capacitance value under the influence of the environmental and electrical operating conditions. Class 2 capacitors are affected by temperature, voltage (both AC and DC), frequency and time. Temperature effects for Class 2 are exhibited as non-linear capacitance changes with temperature.

In specifying capacitance change with temperature, EIA expresses capacitance change over an operating temperature range by a 3-symbol code. The first symbol represents the cold temperature end of the range, the second represents the upper limit of the operating range and a third symbol represents the capacitance change allowed over the operating temperature range. Table 9-3 provides a detailed explanation of the EIA system. As an example, a capacitor with a characteristic X7R would change ±15% over the temperature range −55°C to +125°C and is often identical to military characteristics BX. Parts with characteristics are also sometimes called "K1200" but most manufacturers now use higher dielectric constant than 1200 so the term is now taken to mean only X7R and is commonly called semi-stable material.

Table 9-3

MIL-C-11015D CODE			EIA CODE Percent Capacity Change Over Temperature Range	
Symbol	**Temperature Range**		**RS198**	**Temperature Range**
A	−55°C to +85°C		X7	−55°C to +125°C
B	−55°C to +125°C		X5	−55°C to +85°C
C	−55°C to +150°C		Y5	−30°C to +85°C
			Z5	+10°C to +85°C
Symbol	**Cap. Change Zero Volts**	**Cap. Change Rated Volts**	**Code**	**Per Cent Capacity Change**
R	+15%, −15%	+15%, −40%	D	±3.3%
			E	±4.7%
W	+22%, −56%	+22%, −66%	F	±7.5%
			P	±10%
X	+15%, −15%	+15%, −25%	R	±15%
			S	±22%
Y	+30%, −70%	+30%, −80%	T	+22% −33%
			U	+22% −56%
Z	+20%, −20%	+20%, −30%	V	+22% −82%
Temperature characteristic is specified by combining range and change symbols, for example BR or AW. Specification slash sheets indicate the characteristic applicable to a given style of capacitor.			Example—A capacitor is desired with the capacitance value at 25°C to increase no more than 7.5% or decrease no more than 7.5% from −30°C to +85°C. EIA Code wil be Y5F.	

A Z5U temperature characteristic is also extremely popular. It allows a capacitance change of +22% to −56% over the temperature range of +10°C to +85°C, and is usually made with materials with a dielectric constant in the range of 5000 to 10,000.

Effects of Voltage

Whereas variations in temperature affect all of the parameters of ceramic capacitors, voltage basically affects only the capacitance and dissipation factor. The application of DC voltage reduces both the capacitance and dissipation factor while the application of an AC voltage within a reasonable range tends to increase both capacitance and dissipation factor readings. If a high enough AC voltage is applied, eventually it will reduce capacitance just as a DC voltage will. However, the application of this high an AC voltage is normally not encountered.

Since the magnitude of the effect is dependent on the thickness of the dielectric versus the voltage applied (volts per mil) the curve is based on percent of rated voltage in order to give a basic idea of the order of magnitude of the changes in question. Figure 9-66 shows the effects of AC voltage.

These are of major significance in some applications but are perhaps of even more significance when it comes to measuring the capacitance value and dissipation factor of the capacitors. Capacitor specifications specify the AC voltage at which to measure (normally 1 VAC) and application of the wrong voltage can cause spurious readings. Figure 9-67 gives

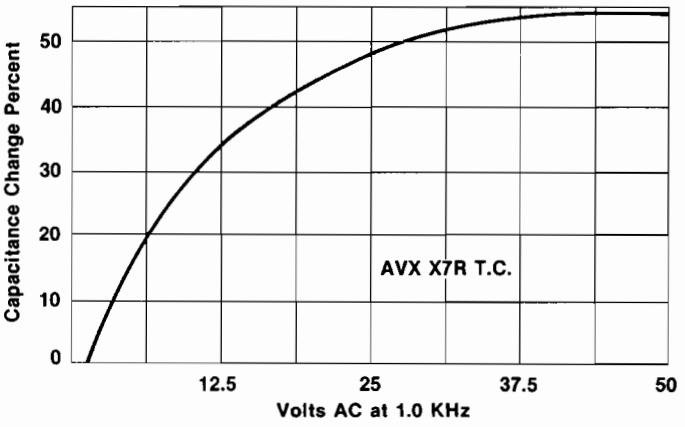


Figure 9-66

the voltage coefficient of dissipation factor for AC based on 1000 cycles reading and 1 kilohertz readings. Applications of different frequencies will affect the percentage changes versus voltages.

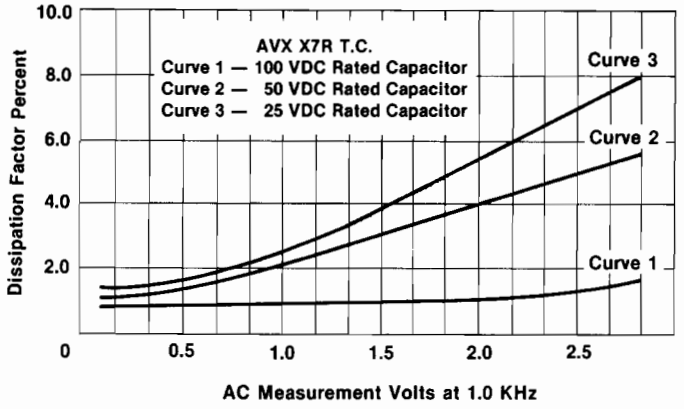


Figure 9-67

The effect of the application of DC voltage is once again dependent on the thickness of the dielectric (volts per mil) and is shown in a similar manner in Figure 9-68. As will be noted in general, the voltage coefficient is more pronounced for higher K dielectrics. These figures are shown for room temperature conditions. Of considerable interest to the user is a combination characteristic known as voltage temperature limit which shows the effects of rated voltage over the operating temperature range. Figure 9-69 shows a capacitor of military specification type BX.

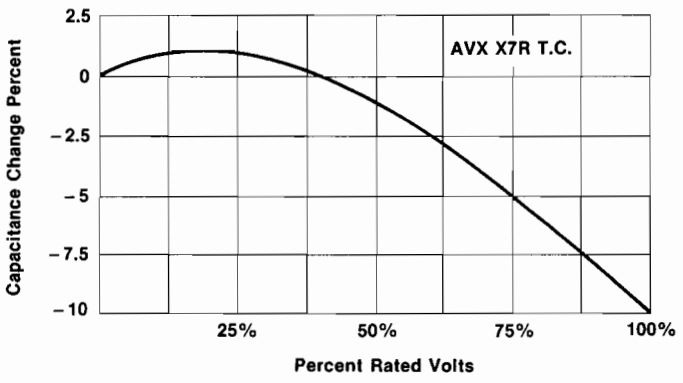


Figure 9-68

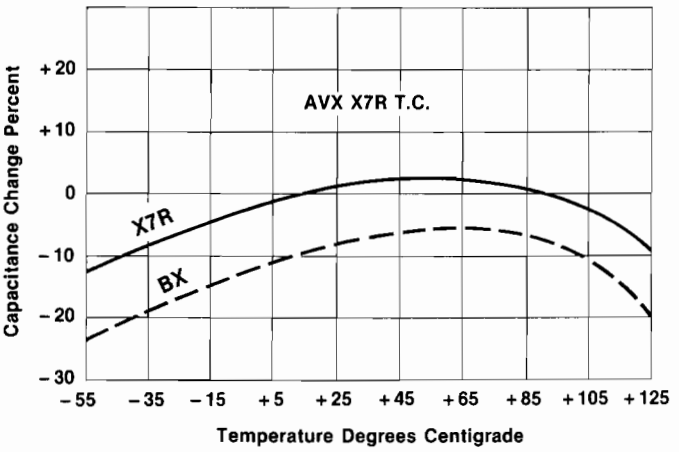


Figure 9-69

Effects of Frequency

Frequency affects capacitance and dissipation factor as is the case with voltage. Curves of capacitance change and dissipation factor change with normal type ceramics are shown in Figure 9-70 and 9-71.

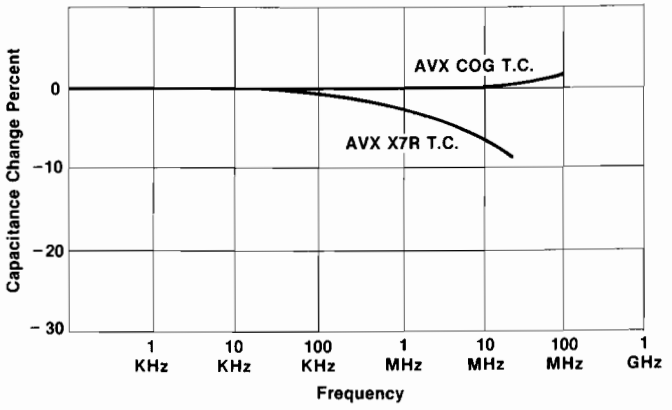


Figure 9-70

Variation of impedance with frequency is an important consideration for decoupling capacitor applications. Lead length, lead configuration and body size all affect the impedance level as well as the ceramic formulation variations.

Special ceramic materials are also made for use at extremely high frequencies.

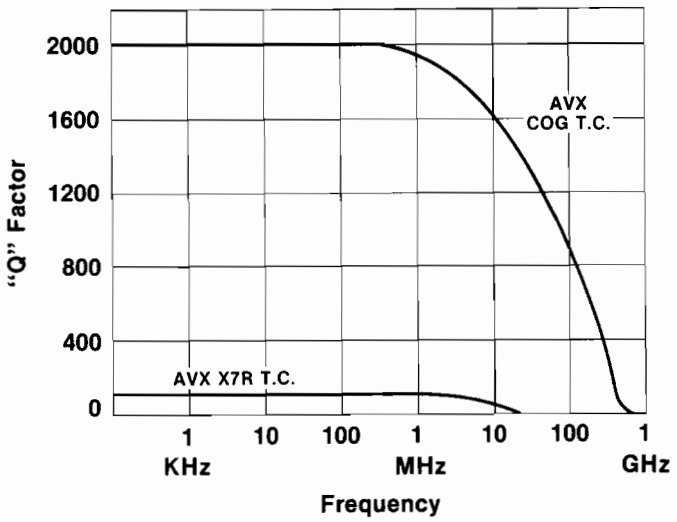
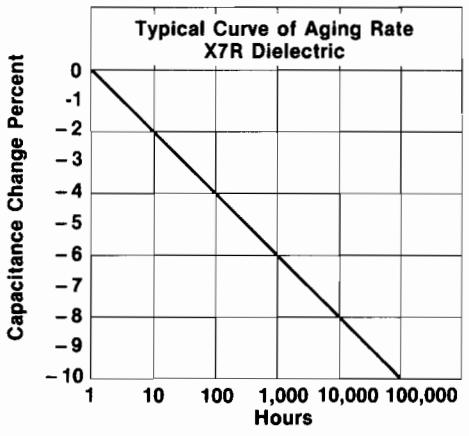


Figure 9-71

Effects of Time

Class 2 ceramic capacitors change capacitance and dissipation factor with time as well as temperature, voltage and frequency. This change with time is known as aging. Aging is caused by a gradual re-alignment of the crystalline structure of the ceramic and produces an exponential loss in capacitance and decrease in dissipation factor versus time. A curve of typical aging rate of semistable ceramic is shown in Figure 9-72 and a table is given showing the aging rates of various dielectrics.



Characteristic	Max Aging Rate %/Decade
A	None
C	1.5
E	5

Figure 9-72

If a ceramic capacitor that has been sitting on the shelf for a period of time is heated above its Curie point (125°C for 4 hours or 150°C for ½ hour will suffice), the part will de-age and return to its initial capacitance and dissipation factor readings. Because the capacitance changes rapidly, immediately after de-aging, the basic capacitance measurements are normally referred to a time period sometime after the de-aging process. Various manufacturers use different time bases but the most popular one is one day or twenty-four hours after "last heat." Aging as noted is expressed in terms of percent per decade with a split in the industry as to whether it should be day decades or hour decades. This is of concern to the user only because the industry only guarantees the capacitor to be within tolerance for two decades after receipt which means a difference between a thousand hours (42 days) depending on the system employed. Permanent change in the aging curve can be caused by the application of voltage and other stresses and if this is of importance to the designer, he should consult the manufacturer for further details. The possible changes in capacitance due to de-aging by heating the unit explain why capacitance changes are allowed after test, such as temperature cycling, moisture resistance, etc., in mil specs. The application of high voltages such as dielectric withstanding voltages also tends to de-age capacitors and once again explains why re-reading of capacitance after 12 or 24 hours is allowed in military specs after dielectric strength tests have been performed.

9

Effects of Mechanical Stress

Ceramic capacitors exhibit some low level piezoelectric reactions under mechanical stress. As a general statement, the piezoelectric output is higher, the higher the dielectric constant of the ceramic. It therefore is often wise to investigate this effect before using high K dielectrics as coupling capacitors in extremely low level applications. Normally for this type of application semistable (X7R, BX, K1200, etc.) material works considerably better.

Reliability

Historically ceramic capacitors have been considered one of the most reliable types of capacitors in use today. The inherent reliability of the ceramic capacitor can be improved by power aging (burn-in) and other types of screening intended to remove early failures. In testing of ceramic capacitors, fairly high dielectric strength tests are applied which will remove units with any gross mechanical defects such as large voids in the dielectric or other mechanical weaknesses in the basic dielectric. Power aging or burn-in not only will

supplement the detection of this type of failure, but also, if properly applied, tend to eliminate early wear-out modes of failure. Assuming that gross mechanical defects and early wear-out types of failures have been reasonably eliminated by dielectric strength test and burn-in, the effects of temperature and voltage can be considered. The approximate formula for the reliability of a ceramic capacitor is:

(E9-51) $$\frac{L_O}{L_t} = \left(\frac{V_t}{V_O}\right)^X \left(\frac{T_t}{T_O}\right)^Y$$

L_o = Operating life
L_t = Test life
V_t = Test voltage
V_o = Operating voltage
T_t = Test temperature and
T_o = Operating temperature in °C
X, Y = See test

Historically in the ceramic capacitor business the exponent X has always been considered as 3. Considerable work has been done to either verify or disprove the "cube law" for ceramic capacitors, and although it has been shown that it is not always accurate, it has been demonstrated to be a reasonable approximation. The exponent Y for temperature effects typically tends to run about 8. Taking some examples, one can see readily that lowering the voltage and the temperature dramatically improves the anticipated performance of ceramic capacitors, pointing out once again the importance of careful design. One factor as far as reliability that is usually not considered is the impedance of the circuit versus the impedance of the test conditions. All standard ceramic test requirements at the present time specify that currents be limited to 50 milliamps or less. With the advent of semi-conducting devices, however, many applications are operating at considerably higher current availabilities in very low impedance circuits. Because of the differences between test conditions and operating conditions failures can sometimes occur in components that would perform reliably under the high impedance test conditions. Quite often consultation with the manufacturer and special screening are desirable for this type of application.

Reprinted with permission of AVX Capacitors.

FAIRCHILD
A Schlumberger Company

Fast Characteristics and Testing

Introduction

This section deals with the testing of devices at maximum operating frequency and the problems that can be encountered in high-speed devices and systems. Ground lift effects cause problems when multiple outputs switch heavy loads at high speeds. The cures for these problems lie in the package selection and board layout.

Mastering the Problems of f_{max} Testing

Introduction

The emphasis in logic designs today is on speed. It is insufficient to make a functional circuit—in addition, that circuit must perform its function at a very high speed. The functional circuit then becomes limited by the weakest link in its clock-dependent synchronous circuit elements. Since PCB technology has been refined to transmit high data rates with ease, the weak link then becomes the frequency performance of the individual logic elements in the functional circuit. A measure of functional performance at the logic device level is found in the parameter f_{max} or "maximum frequency of operation." This specification provides the designer with a measure of performance that can be compared with other logic elements in the same functional circuit to determine suitability or compatibility with the design objectives.

However, the testing of the parameter is difficult since the inherent nature of the test implies that the functional performance of the logic element must be determined at frequencies which can exceed 100MHz. It is a complex problem and is beyond the scope of most commercial Automatic Test Equipment (ATE).

An evaluation of existing test systems reveals that very few consider running their functional drivers at 100MHz. The reason is that, for a driver to operate at that rate, it must have very fast edge rates so as not to degrade as the frequency moves up. Another problem common to test systems is that the formatter logic skews which result from mode changes are large enough to prevent high frequencies from passing through.

Some interesting and yet challenging problems confront the test engineer who tries to test devices in that frequency range. This section presents an overview of the kinds of problems which one may expect and some possible solutions.

Defining the Problem

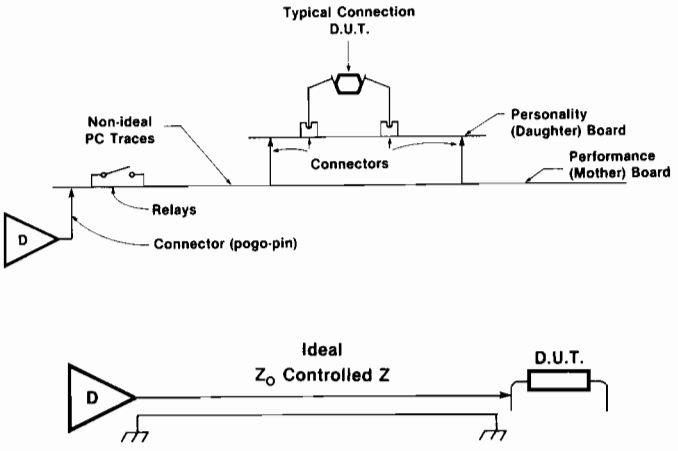
As bipolar technologies produced devices that were able to propagate signals in the 3 to 5ns range, the edge rates were made faster as a direct result of the internal speed enhancements. The problem that arose is best explained by an understanding of the type of signals the devices produced. A Fourier analysis of a 2.5ns edge reveals some interesting frequency components: major magnitudes at 200, 400, and 600MHz (see Figure 10-3). Each of the components combines with the others to produce the resultant square wave.

The prospective test equipment must present some form of interface which will transmit accurately the input stimulus with a minimum of degradation to the D.U.T. This is a complex problem. Transmission of a fast edge rate from a test system driver to a D.U.T. requires a controlled impedance environment. The impedance of the driver must be matched with the transmission line and it must be capable of generating the desired edge rate looking into that impedance.

Crossing the Boundary

Another consideration is that of signal boundaries. A boundary would be any point at which the signal must pass from or through an impedance different from the characteristic impedance of the transmission line. This may be in the form of a connector, po-go pins, open blade contactors on a piece of handling equipment and other not so obvious discontinuities (Figure 10-1). Poor signal boundaries cause the overall bandwidth of the transmission path to degrade resulting in a lower edge rate and hence less energy available to drive the D.U.T. In addition, if the driver impedance is not matched, secondary reflections will result when the reflected signal produced at signal boundaries is not totally absorbed by the source. If the transmission path is short, then the edge could be aberrated by the reflections resulting in severly distorted drive signal.

Figure 10-1



What the Device Being Tested Sees

The signal at the far end of a properly terminated lossless transmission line may be said to be exactly like the input to the line except delayed in time due to finite velocity in the line itself. Unfortunately, lossless lines do not exist and it is seldom possible to terminate an automatic test system properly, therefore the edge shape at the D.U.T. is somewhat aberrated from the source signal. This can result in several undesirable effects. First, if the line is terminated in some lumped capacitance (i.e., D.U.T input capacitance, Printed Circuit to ground capacitance, etc.), the edge rate will degrade. Second, due to finite skin-effect losses, the top and bottom portions of the edge will tend to round out. Third, the unterminated or mainly reactive termination at the end of the line will cause the transmission line current to collapse and produce a resultant increase in the signal amplitude. How much or how little depends on the magnitude and quality of the termination.

Ground Lift Effects

As the frequency of test goes up, the edges begin to resemble a sinusoidal wave. The available energy is reduced significantly and ground lift effects, which may or may not have been troublesome at lower frequency propagation delay testing, now cause significant problems. Since the energy contained in the driving pulse through the threshold region is critical, it becomes apparent that any lift in the ground due to parasitic inductances, crosstalk and inadequate decoupling will potentially cause the device's threshold to appear higher than it actually is. The result is that the device sees less energy for that instant in time. This undesirable effect may cause a functionally good device to malfunction.

To minimize this problem it will be necessary to consider power source impedances as they appear to the device under extreme switching situations. Adequate decoupling in the form of low impedance power planes and low series inductance decoupling capacitors coupled with a large low inductance connection from the D.U.T. to the performance board ground are mandatory to achieve the desired results.

What Can Be Done

Test engineers must consider the above effects when implementing f_{max} testing on automatic testers. A close-to-ideal transmission path must be designed in. It is important to minimize the number of boundaries (i.e. relays, connectors, etc.). The single biggest contributor to distorted high-frequency signals may be the contactor, if it is of an open blade construction. Care must be taken to see that a special contactor with a controlled impedance path is used in these applications.

Figure 10-2 Time Domain Data

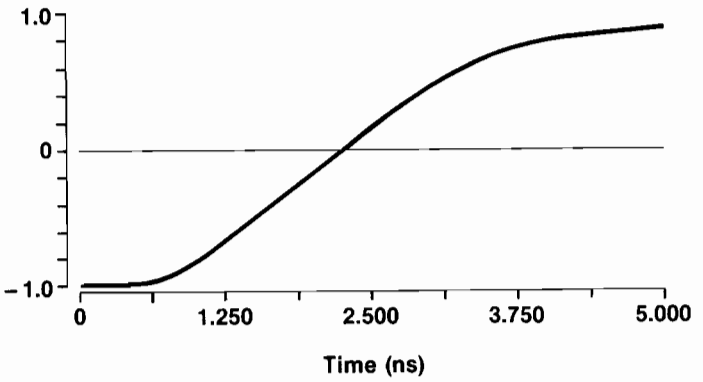
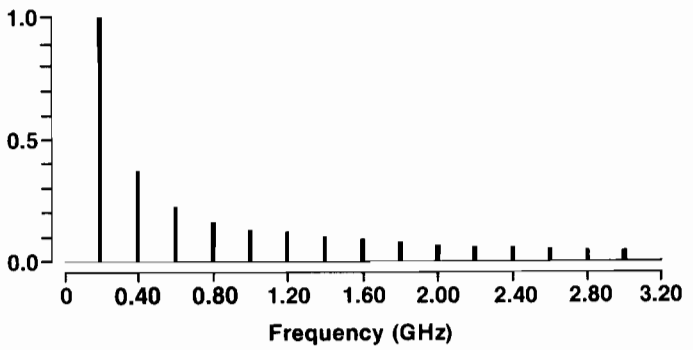


Figure 10-3 Frequency Domain Data



10

Noise Budget: Designing Computer Hardware to Meet System Noise Limits

Introduction

Every logic family has a noise immunity (sometimes called noise margin). In this section it is shown that the allocation of the available noise margin according to the various noise sources is supported by the noise budgeting theory using statistical methods. Noise budgeting implies that proper noise reduction techniques are applied in the design to accommodate the established budget limits as well as possible.

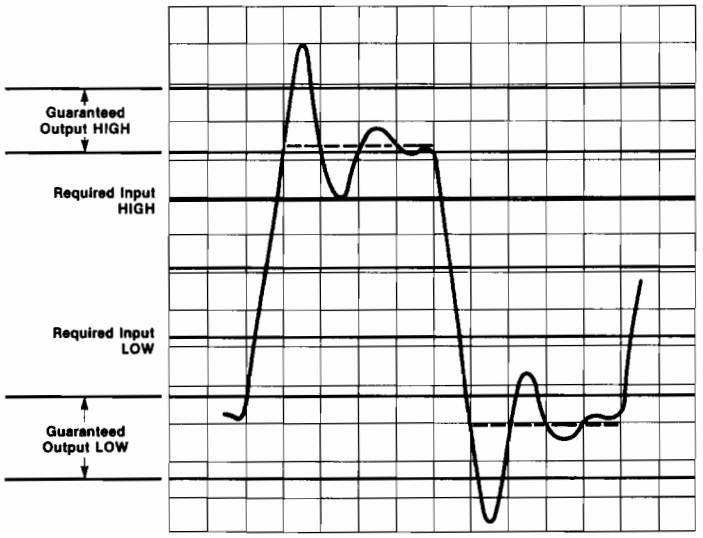


Figure 10-4 Noise

A simple definition is offered here: noise is the undesired part of the signals that are generated. Figure 10-4 above is a typical computer signal with some noise present, the kind present when a line is terminated with a resistance higher than the characteristic impedance. This noise is just barely within acceptable limits for the circuit input so it will not appear at the circuit output or cause any problems as long as no other noises are affecting the circuit. Therefore the noise immunity has not been violated.

Noise Immunity

The noise immunity of a circuit may be defined as the maximum amount of noise on a particular node which will not cause the output level to move to an invalid state. Or, noise immunity is the amount of noise on an input which will not propagate through a worst case chain of gates, causing a flip-flop or other circuit to respond to the resulting transient. Figure 10-5 illustrates both of these ideas. During time B, a wide noise pulse pushes the high down to the input limit for highs and uses up all of the noise margin. During time C, a narrow noise adds to the first pulse and pushes the signal past threshold. This noise could propagate through a chain of gates.

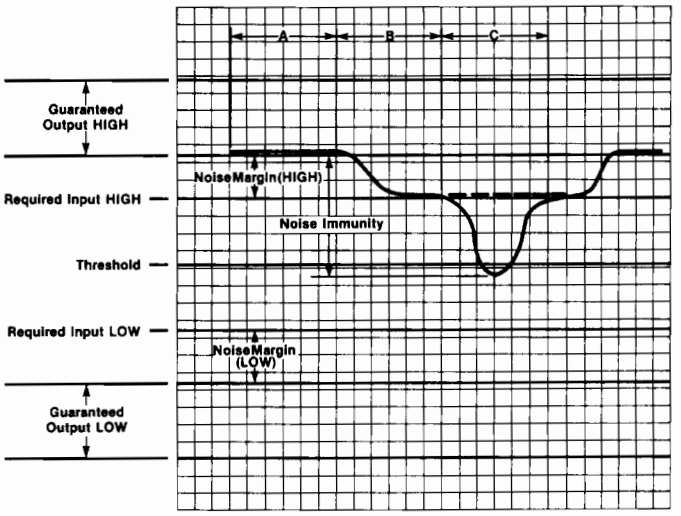


Figure 10-5

In Figure 10-6 the noise pulse is propagating through a chain of gates. Each gate has worst case noise immunity, and depending on conditions, each could amplify the noise pulse as shown here, until it is big enough to violate the noise immunity of a flip-flop and be stored as an error.

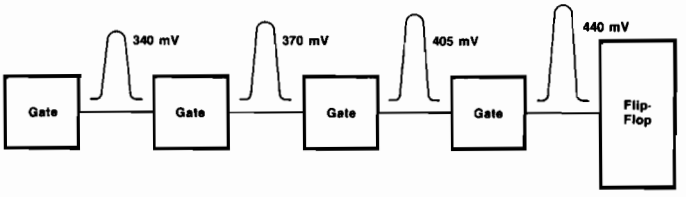


Figure 10-6 Noise Propagating Through a Chain of Gates

Noise Sources

The following list names the most common noise sources:

- External Noise—radiated signals from electromagnetic fields
- Power Supply and Distribution—noise coupled into the system by the power supplies
- Crosstalk—noise induced into signal lines from adjacent signal lines
- Transmission Line Reflections—over and undershoot, ringing caused by improper terminations

There are also certain environmental factors which affect the noise margin, such as:

- Ground Offset—potential difference between two different grounding points in a system
- Ambient Temperature—differences in the temperatures of communicating devices

The Noise Budget

Analyzing the noise sources and relating them to the available noise margins comprise the process of noise budgeting. The noise budget is a table listing the various noise contributing factors with their allocated DC and AC noise limits (Table 10-1).

Notice the use of Effectivity Factors: these factors can usually be determined from the vendor-supplied circuit characteristics and graphs. They indicate the measure by which a particular circuit parameter, such as power supply or temperature variation, affects the output level of the device.

Table 10-1 The Noise Budget

Source	DC Noise	DC Eff.	Equiv. DC	AC Noise	AC Eff.	Equiv. AC	Total Eff. Noise
1. Gnd to Gnd	15 mV	1.0	15 mV	100 mV	0.45	45 mV	60 mV
2. PC Card Crosstalk	—	—	—	82 mV	1.0	82 mV	82 mV
3. Backpanel Crosstalk	—	—	—	74 mV	1.0	74 mV	74 mV
4. V_{CC} Bus	211 mV	0.10	21 mV	90 mV	0.29	26 mV	47 mV
5. Temperature III	12 mV	1.0	12 mV	—	—	—	12 mV
6. SIP Crosstalk	—	—	—	25 mV	1.0	25 mV	25 mV
7. Termination	—	—	—	23 mV	1.0	23 mV	23 mV
8. Wire Untwist	—	—	—	30 mV	1.0	30 mV	30 mV
9. EMI	—	—	—	2 mV	1.0	2 mV	2 mV

The RSS of the total effective noise is 149 mV. This represents the upper 3 sigma point of the noise, or the value that will not be exceeded over 99.86% of the machine for this circuit type.

Theory of Noise Budgeting

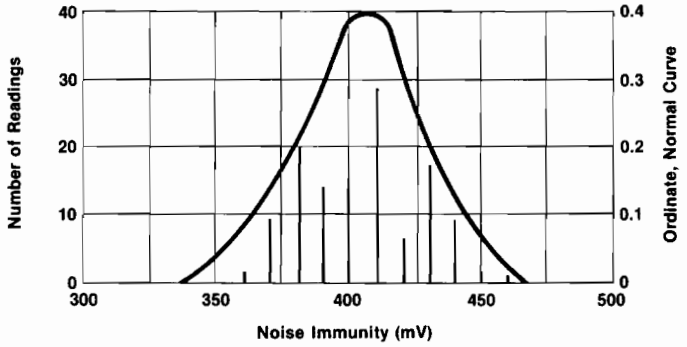


Figure 10-7 Input Noise Immunity of Simple Gates

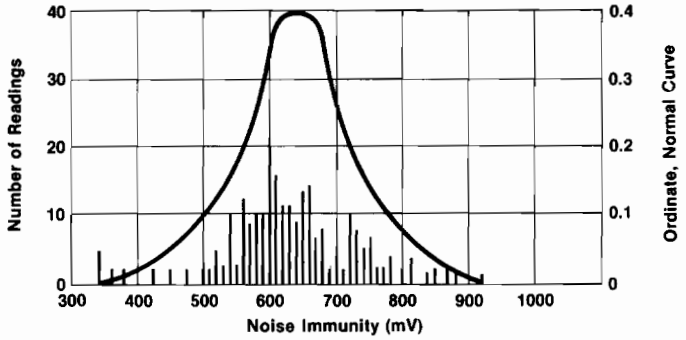


Figure 10-8 Input Noise Immunity of Clocked Circuits

A normal distribution of noise immunities of a particular circuit may be expected. Thus, a plot of the noise immunity of a large sample of circuits at any particular pin, under a given set of test conditions, will follow the common bell curve.

The first consideration is electrical noise in general, since much noise has a Gaussian or normal distribution of instantaneous amplitudes with time.

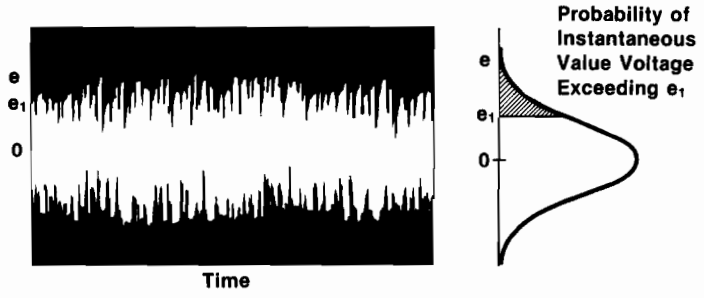


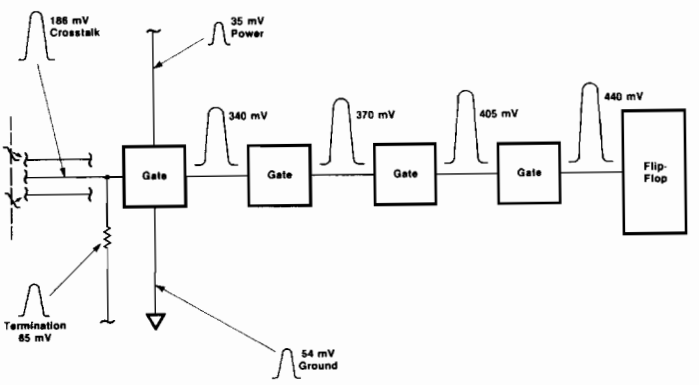
Figure 10-9 Noise Waveform and Gaussian Distribution of Amplitudes

The area under the Gaussian curve (Figure 10-9) represents the probability that a particular event will occur. If we consider a noise value such as e_1, the probability of exceeding that level at any instant in time is shown by the cross-hatched area. A good engineering approximation is that common electrical noise lies within plus or minus three times the root-mean-square (RMS) value of the noise wave. The peak-to-peak voltage is less than six times the RMS for 99.7% of the time.

It so happens that 99.7% of the time represents the area between the 3 sigma points of the Gaussian curve. From this it can be seen that, when measuring random noise with an oscilloscope, the peak-to-peak voltage represents the noise within the 3 sigma limits. Also the 1 sigma limit represents the RMS value of the wave.

In order to make the above theory applicable to our noise budgeting, it must be shown that computer noise is random. Some of the noise voltages are slow variations due to line voltage changes, power supply drift and ripple, temperature changes and differentials. Fast noise pulses (spikes) due to switching transients travel along the lines, reflecting and re-reflecting until they finally die down. The net result of all this is that the various noise sources produce fairly random noise consisting of a great many frequencies. By the time the noise is "summed up" in each individual circuit, the total noise is more random than any of the individual contributors. Although there will be some predominant frequencies at the system clock frequency and its harmonics, noise pulses caused by the clock edges will travel random distances before reaching a particular circuit and this will, in fact, contribute to the randomness of the noise. Therefore, it is safe to say that for most practical purposes, computer noise may be considered to be completely random.

Figure 10-10 Violation of Noise Immunity in Poorly Designed Circuit



Each circuit in a computer is constantly summing up the various noise contributors. Figure 10-10 shows what could happen in a poorly designed machine. Although the noises are random, at this particular instant in time, each source happens to have a noise pulse in the same direction. These have been allowed to become so large that the noise immunity of the circuit has been violated. But events like this are extremely rare, even in a poorly designed machine, due to the extremely low probability of enough noise contributors moving far enough in the same direction at the same time.

In the well-designed machine, all contributors are kept at values low enough that the summing process does not result in noise immunity violations.

If several sources of random noise are present within a circuit and if these sources are effectively in series, the total equivalent noise can be obtained by taking the root of the sum of the squares (RSS) of the individual sources. The formula is:

(E10-1) $$E\ equivalent = \sqrt{E_1^2 + E_2^2 + \ldots E_n^2}$$

where all values are RMS values.

In a circuit where the various noise sources are not directly in series, each noise source voltage may be multiplied by a suitable effectivity factor (see above). The resulting equivalent noises may then be RSS-ed.

It can be shown that peak values of the noise voltages may be RSS-ed to give the peak value of the total effective noise. Therefore, if the peak values of noise are measured from various sources at a particular logic circuit, multiplied by the proper effectivity factors and RSS-ed, the peak value of the effective noise at that circuit is obtained. This peak value will not be exceeded 99.7% of the time.

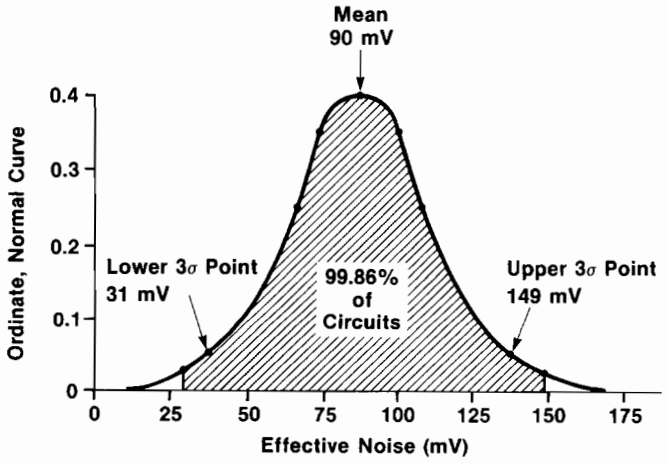


Figure 10-11 Distribution of Noise for Typical Mainframe

Finally, if one measures sufficiently large samples of the noise from all sources, one can determine the upper 3 sigma (worst case) values, apply the appropriate effectivity factors and RSS the results to get the worst case effective noise. This is the value that will not be exceeded over 99.86% of the machine, 99.7% of the time.

Conclusion

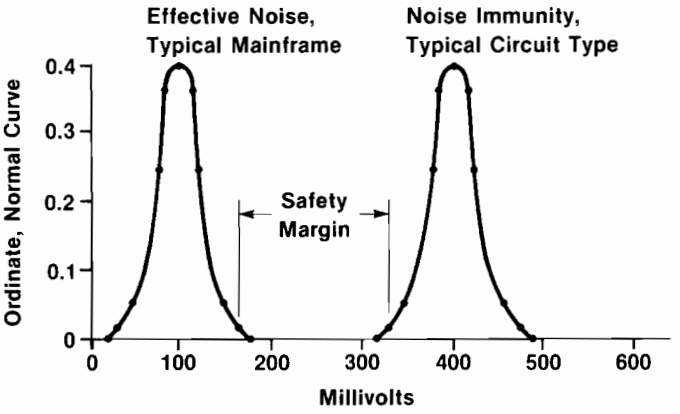


Figure 10-12 Safety Margin Between Effective Noise and Immunity

This graph (Figure 10-12) shows the effective noise for logic boards with a particular circuit family and the noise immunity of a typical circuit type from that family. When a piece of electronic equipment is properly designed, there is a good safety margin between the effective noise and the noise immunity of the circuits.

The adequacy of the safety margin can be confirmed by running tests in which the bus voltages are varied from nominal by some set percentage. These are bus voltage margin tests.

References

1.C.D. Motchenbacher and F.C. Fitchen, *Low Noise Electronic Design,* Wiley, 1973.

Ground Lift Effects on FAST Logic Devices

Introduction

The early bipolar logic devices introduced relatively fast edge rates to the designer and an entirely new way of thinking about circuit design. It soon became apparent that loose wiring practices produced undesirable effects such as malfunction and/or "soft errors." Lead dress, power and grounding systems had to be developed to improve the performance of new designs.

If designers failed to implement these techniques, the outcome was almost inevitably reduced performance. New developments in the digital field not only reduced the propagation time of a signal through a device but also reduced the time required in making the state transition. This presented even more complications to designers who now had to consider such factors as the time elapsed on a signal path going from one device to another, distributive capacitive effects of devices scattered throughout a circuit board and backplane effects.

Fairchild's Advanced Schottky TTL (FAST) produced even greater improvements in speed performance but not without an accompanying collection of new considerations. Terms such as "incident-wave/reflected-wave switching" have become common words among designers of high-performance logic circuits.

To complicate matters further, devices which had been specifically designed to accommodate the microprocessor market began to appear in these high-speed logic families. Octal functions offered increased flexibility (of the products) and were quickly utilized by microcomputer designers. New problems became evident when devices were presented with simultaneous stimuli. The effects of simultaneous switching and some derating factors derived empirically will be discussed in this section.

Understanding the Phenomenon

The effects of simultaneous switching range from increased propagation delay to output glitching, but in most cases the source can be traced to poor grounds and poor decoupling of power sources. However, there are some packaging characteristics which limit the device performance.

Undesirable effects are produced by attempting to transport large amounts of charge from the loads to either the V_{CC} or ground power planes. For an octal device with each of its eight outputs having a full 50 picofarad load connected to it, where ideally there are no current limiting factors, the current could be defined as

(E10-2) $$i(t) = C_t \frac{dv}{dt}$$

Where the dv/dt is large, the i(t) is large also. This means that as the amount of time required to make a transition from one logic state (i.e., V_{OH}) to another (i.e., V_{OL}) is decreased, the current will increase proportionately. If C_t is increased, either by increasing its value at a single node or adding nodes of additional capacity which are undergoing the same transition at the same time, the result is an increased current i(t).

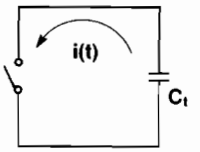


Figure 10-13

In reality, however, there are limiting factors. The capacitance is not really being charged or discharged from a lossless source. The package and PCB also have effects which limit this. All packages and PCB traces high-speed devices. All packages and PCB traces contain some finite inductance. These parameters, combined with collector saturation resistances, limit the charge and discharge currents. Kirchoff's Voltage Law shows how the voltages are distributed around an output that is switching.

(E10-3) $$V = R_s \cdot i(t) = \left(L_1 L_2\right) \frac{di}{dt} = \frac{1}{C_t} \int_0^t idt$$

where C_t = total load capacitance
L_1 = signal inductance
L_2 = ground inductance
R_s = output Tx R_{sat}

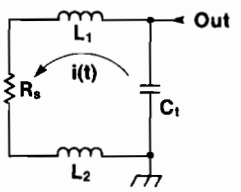


Figure 10-14

Note that these become limiting factors for the peak currents that are developed and that the resulting voltage losses in the elements govern the transition time from one state to another. The inductances L_1 and L_2 represent the inductances present in the output line and the ground connection respectively.

The reason is that dynamic currents flowing between the substrate and absolute ground produce a voltage drop between those two nodes. The result is that the substrate voltage momentarily lifts up above the absolute ground potential. This is not desirable for two reasons: the substrate lifting reduces the available voltage across the phase splitter transistor and collector resistor reducing the base drive momentarily. This in turn decreases the I_{OL} sinking capability of the pull-down transistor, momentarily slowing down the transition time and in effect increasing the propagation delay; if the device has a short propagation delay, i.e., output transition overlaps input stimulus, then a shift in the input threshold can cause a "stepping" on the output waveform, usually in the range of the threshold voltage. Additionally, the voltage produced by the transient current flowing in the ground leg can be transmitted to "quiet" outputs through the substrate, producing spiked outputs on pins that are supposed to be at V_{OL} potential. The dominant spike or pulse is that which is produced by the di/dt collapsing to zero while the "quiet" output is driven below ground momentarily.

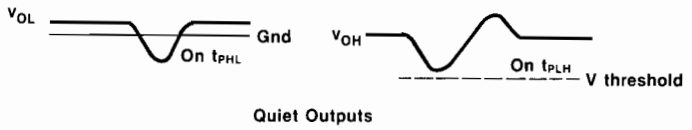


Figure 10-15

The equation for voltage in an inductor

(E10-4) $$V_L = L \cdot \frac{di}{dt}$$

indicates that inductance due to the ground connection must be at its absolute minimum to minimize the voltage induced from the chip substrate to the power supply digital ground. The object here is to reduce the voltage drop from the substrate to the system ground. This voltage is developed across the ground inductance in the system. This inductance is a combination of the package and the PCB trace inductance. Therefore, the use of ground planes is highly recommended because they provide the lowest inductive connection between a package and the power supply.

Reduced performance will also be realized when the demand for instantaneous current is impressed on the V_{CC} pin as would be the case for all or many outputs switching from the V_{OL} to the V_{OH} state. Here the V_{CC} pin would drop in potential, momentarily limiting the drive to external loads. This also shows itself in the form of increased propagation delays.

Derating Product Specifications for Multiple Output Switching

It is assumed that the designer will have already considered the importance of V_{CC} and ground power planes. Decoupling capacitances are determined based on the maximum allowable V_{CC} droop as a result of the instantaneous current demand.

The experiments conducted by Fairchild involved several devices of an octal structure. Some of the devices are of a sequential design and some of a combinatorial nature. The products were connected to a Fairchild PC-163 5-layer PC fixture for this experiment. The fixture was equipped with low-profile machined socket terminals to provide a connection roughly equivalent to soldering the chip to a properly decoupled low inductance PC board.

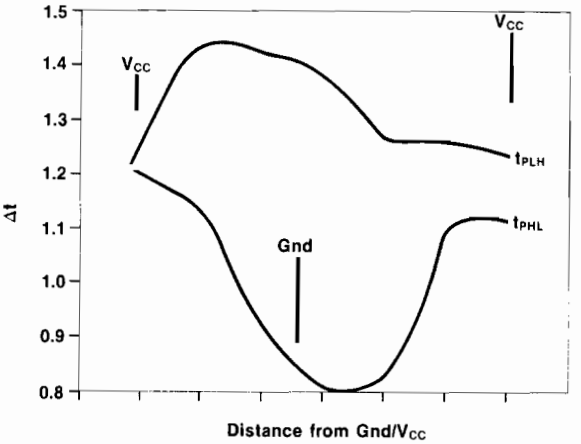


Figure 10-16

The graph in Figure 10-16 illustrates the typical skew that occurred as a result of all outputs switching as a function of distance from the V_{CC} and ground pins on the package. The Δt represented is the difference or skew between individual outputs as each is examined with only that one output switching and then with all eight outputs switching. It should be noted that the top curve is the Δt_{PLH}; the outer two X-axis points represent pins closest to the V_{CC} pin; the lower curve depicts the Δt_{PHL} as a function of distance from the ground pin (X-axis center of graph).

10

Figure 10-17 'F299 t_{PHL} MR-Q7

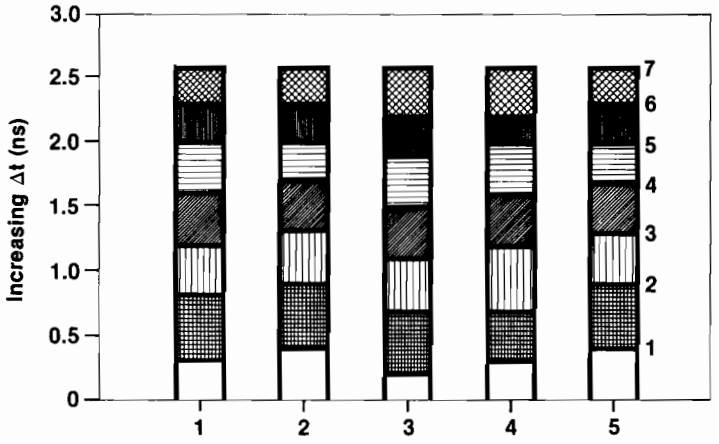


Figure 10-17 illustrates the incremental Δt in nanoseconds experienced on the 'F299. The five devices showed about the same added delay with all outputs switching; however, the incremental delay, device to device, shows some minor variation.

Figure 10-18 'F374 t_{PLH} CP-Q

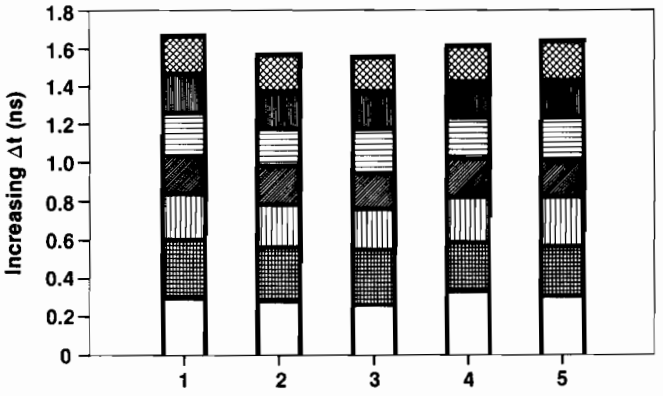


Figure 10-18 illustrates the incremental Δt in nanoseconds experienced on the 'F374. The maximum variation with all outputs switching is about 150 picoseconds.

Figure 10-19 'F245 t_{PLH} A-B

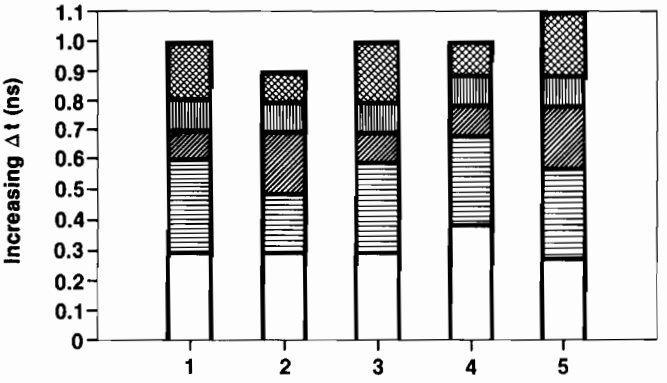


Figure 10-20 'F245 t_{PHL} A-B

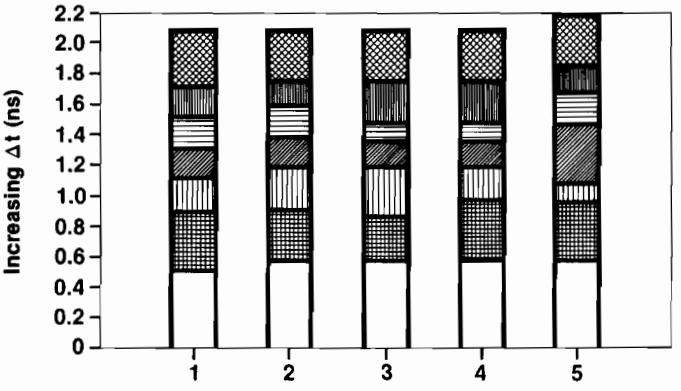


Figure 10-21 'F241 t_{PHL}

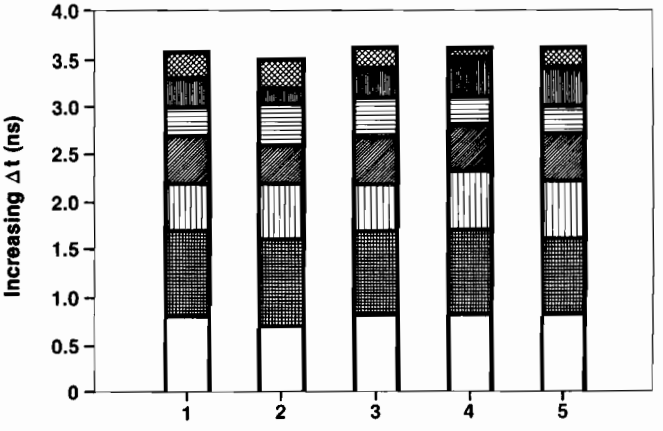


Figure 10-22 'F241 t_{PLH}

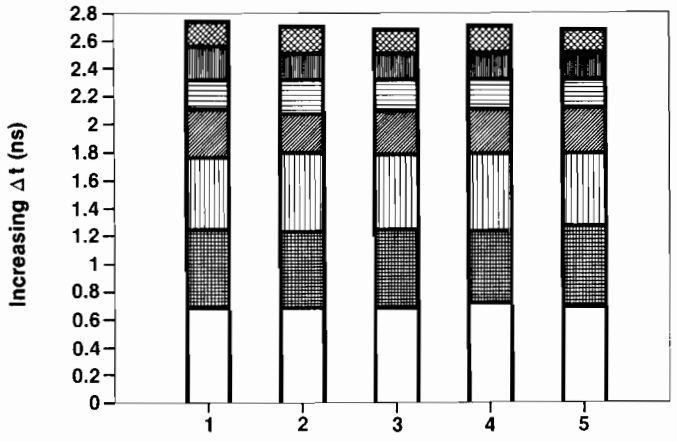


Figure 10-23 'F240 t_{PLH}

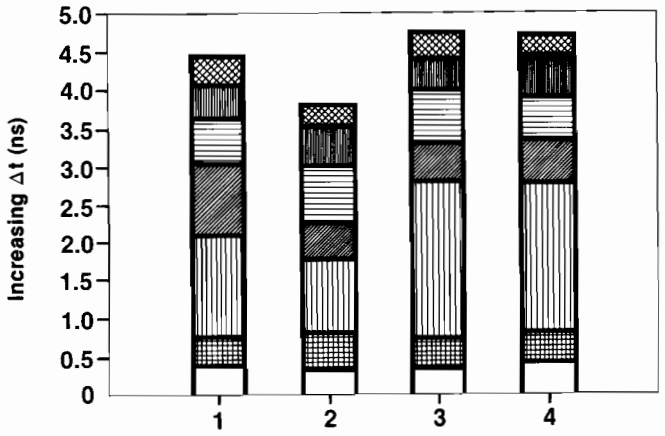
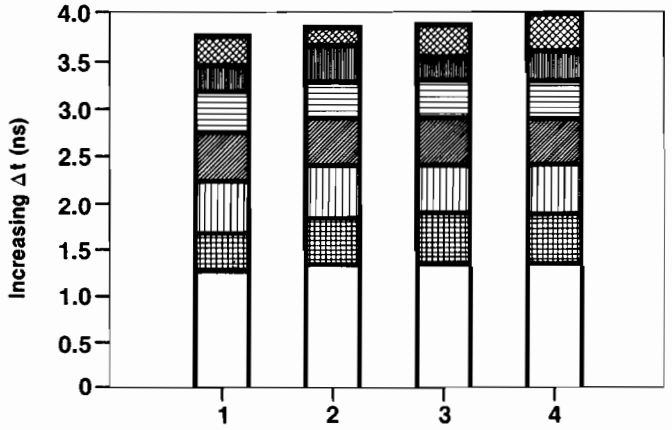


Figure 10-24 'F240 t_{PLH}



Figures 10-19 through 10-24 illustrate the incremental times associated with buffer class devices such as the 'F24X. The times noted were generally more affected by all outputs switching.

Conclusions

The devices, as expected, showed an increase in the Δt between one output and all outputs switching. As the devices were evaluated it was noted that the least gradient was experienced by outputs that were close to the ground pin on the test device. Internally as the individual pull-down transistors turn on, the current di/dt produced by each individual transistor will cause a voltage to be impressed on their individual emitters. The corresponding reduced drive causes an elongated transition time and hence increased propagation time.

The information presented represents the worst case analysis and in most applications performance will be notably better than indicated by the adders.

For non-buffer type FAST devices, it appears as though an adder to the Data Book propagation delays is necessary when multiple output switching is desired. A t_{PHL} adder of n x 400ps (where n = number of additional outputs switching simultaneously) would seem to cover a worst case scenario. The t_{PLH} delta for non buffers would be n x 250ps.

For buffer elements where the dynamic currents are notably higher, an adder of n x 700ps for the t_{PHL} parameter and n x 600ps for the t_{PLH} parameters should be used.

Summary for Dual In Line packages;

Non-buffer devices $t_{PLH} = t_{PLH}(db) + n \times 250ps$
Non-buffer devices $t_{PHL} = t_{PHL}(db) + n \times 400ps$
Buffer devices $t_{PLH} = t_{PLH}(db) + n \times 600ps$
Buffer devices $t_{PHL} = t_{PHL}(db) + n \times 700ps$

db = data book limits @25°C

10

Commercial Test Operations

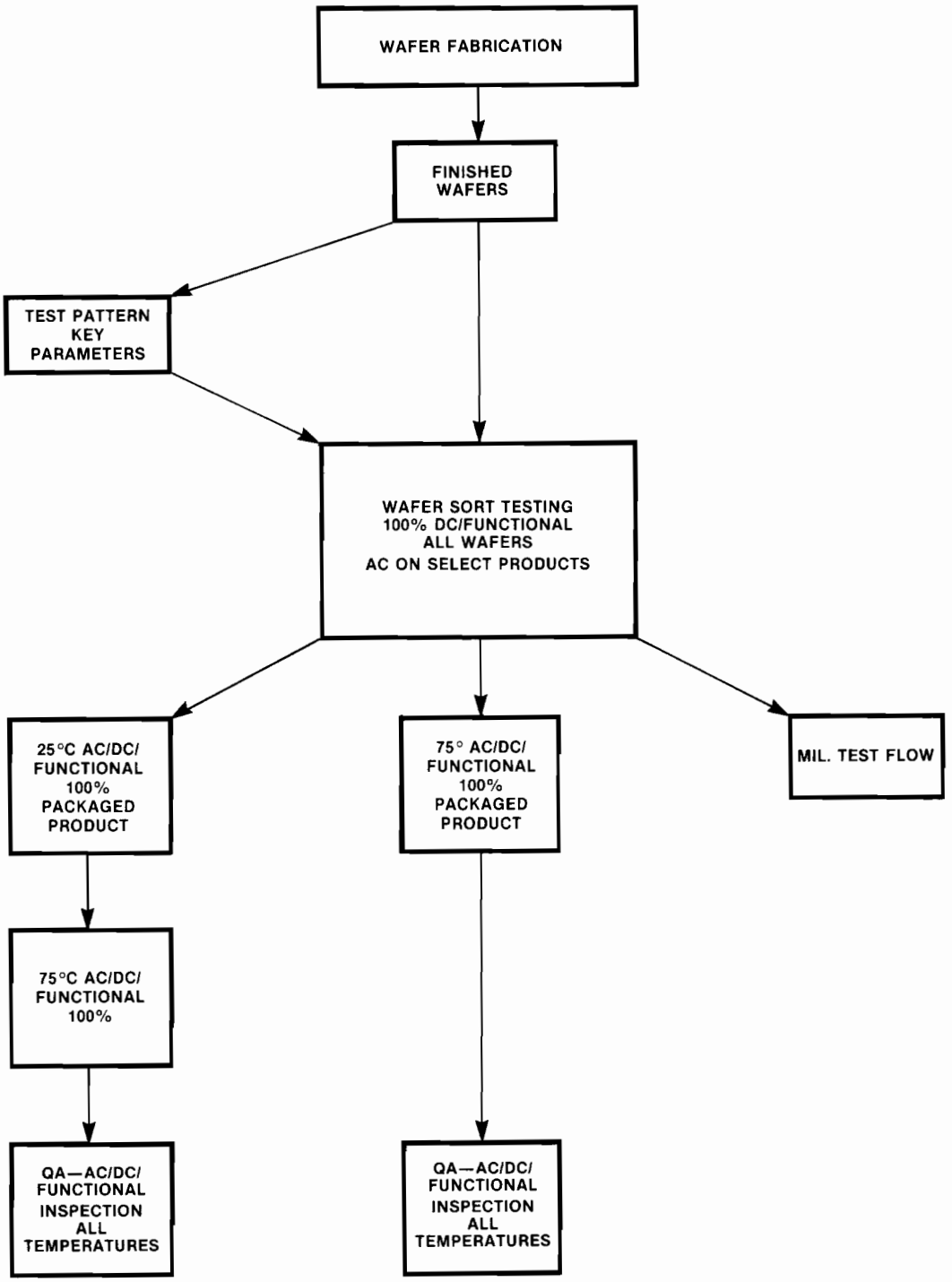


Figure 10-25

The factors described above have been empirically derived from evaluation of a relatively small number of devices. Thus this data should only be used as an aid in system design, and should not be interpreted as absolute, guaranteed values.

As it is likely that significant lot-to-lot variation in these values may occur, use of such values in incoming test programs is not recommended. Test correlation with the manufacturer should only be attempted by use of the industry standard single output switching test method.

Wafer Fabrication
At this level, precise measurements of semiconductor characteristics such as V/I conductivity tests are performed to control the process and provide maximum yields.

Test Pattern Testing
Test patterns exist as a part of all wafers manufactured. The purpose of these test patterns is to provide device engineering with a quantitative measure of the fabrication process. Parameters such as Low Resistance, Beta, High Resistance, Voltage Forward Bias (V_f) of diodes or junctions are measured.

Wafer Sort Testing
In wafer form, each individual die is subjected to rigorous functional and static (DC) measurements. The DC tests include all data book parameters, as well as engineering tests used in manufacturing. Functional tests are performed at $V_{CC} \pm 10\%$ to assure operation over the entire specified operating range.

Packaged Product Testing
After assembly and such finishing operations as lead form, lead plate, and mark, each device is subjected to 100% final electrical test. This testing is performed at +75°C, so as to provide a 5°C guardband above the specified maximum operating temperature of 70°C. At this test operation, all Data Book functional, static, and dynamic specifications are evaluated and guaranteed either by direct test or equivalent methods.

Prior to shipment a statistically significant sample of each lot is subjected to additional electrical testing at 0°C, 25°C, and 70°C by the Quality Control organization. This stringent test procedure insures compliance to all specifications.

Military products are processed in compliance with the requirements of MIL-STD 38510F and MIL-STD 883C.

FAST®

FAIRCHILD
A Schlumberger Company

Packaging Characteristics

Introduction

Three specific aspects of electronic packaging will be discussed: electrical characteristics, thermal considerations and surface mount technology. These are basic guidelines only. They are preceded by a glossary of important terms.

Glossary

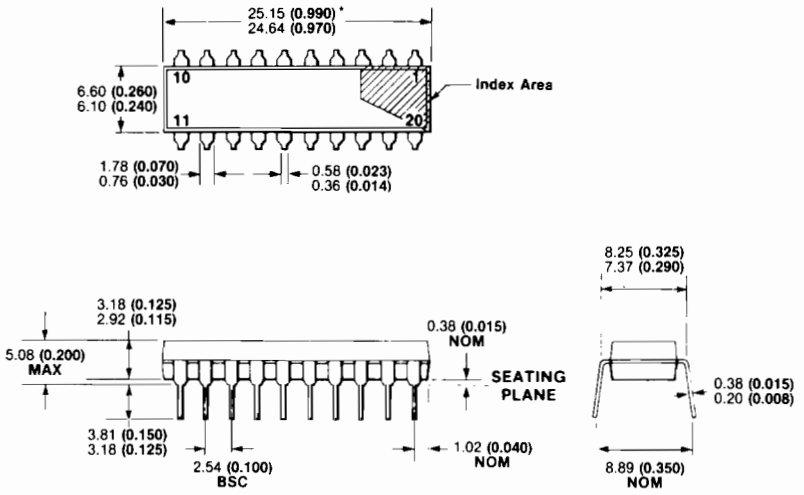


Figure 11-1 20 Lead Plastic Dual In-Line

DIP—The Dual In-line package is the semiconductor workhorse of the 70's and 80's. This component is a through hole mount package and comes either in plastic or ceramic form. The ceramic version is a hermetic package primarily used in military or high-reliability applications, while the plastic package is used in commercial applications. In either format, it consists of a leadframe and some kind of casing for a chip which is attached to the leadframe. The leads are typically on 0.100 inch center and the package is through hole mounted; i.e., mounted in plated through holes in the board or in a socket on the board. A DIP becomes a fairly large package with leads in excess of 40 pins. Mechanical considerations then become critical.

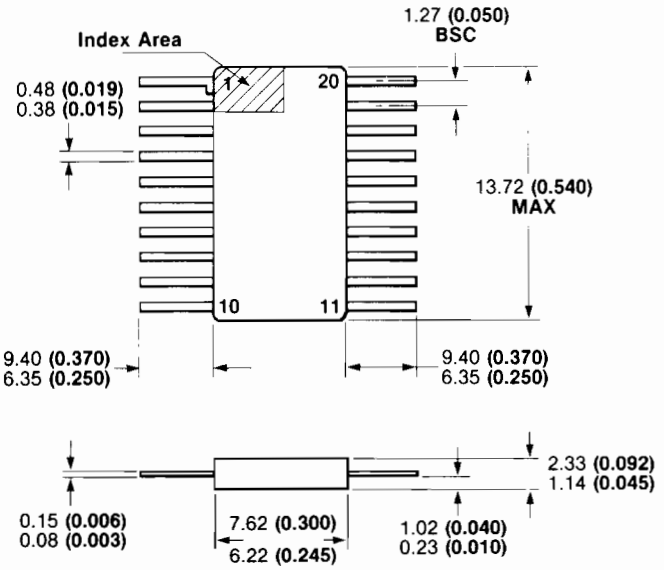


Figure 11-2 20 Lead Ceramic Flatpak

FLATPAK—The flatpak generally has a rectangular ceramic body. The leads are typically on 0.50 inch centers which allows for a reduction in package size and footprint on the printed circuit board as compared with a DIP. The flatpak is a hermetic package with a metal leadframe secured by a glass seal. Flatpaks are also available with leads on all four sides.

SURFACE MOUNT TECHNOLOGY (SMT)—Surface mount packages have terminals or leads which are soldered directly to lands on the surface of the printed circuit board. This technology allows up to a six to one advantage in size and weight over through hole mounting. However, if it is improperly designed, there are penalties in terms of cost or manufacturing. It should be considered that solder is not generally a structural material and has regions of plastic-like behavior in normal use. The following packages are surface mount packages.

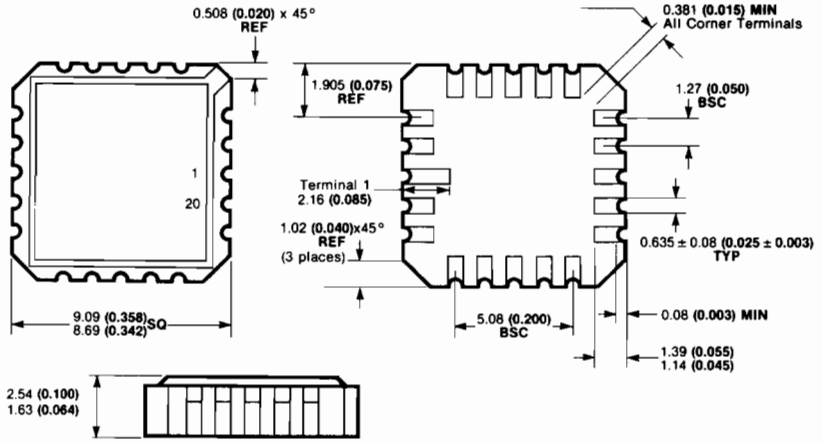


Figure 11-3 20 Terminal Ceramic Leadless Chip Carrier

LCC—The Leadless Chip Carrier is an hermetic ceramic package which generally has leads on 0.040 or 0.050 inch centers but can be as low as 0.010 in extreme applications. Fairchild uses a 0.050 center line package for its military hermetic surface mount devices. Figure 11-3 illustrates a typical leadless chip carrier. It is a 3-layer package and is either glass or solder sealed.

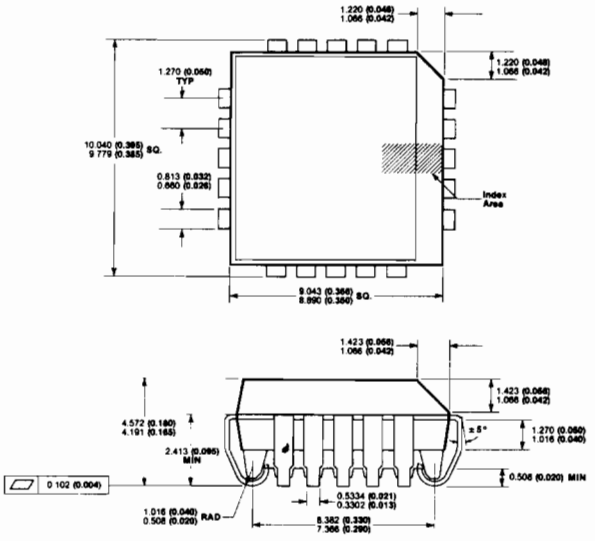


Figure 11-4 20 Lead Plastic Chip Carrier (PCC)

PCC—The Plastic Chip Carrier is a low cost version of the LCC. It has leads which allow compliant bending between the printed circuit board and the chip carrier. Figure 11-4 illustrates a typical plastic leaded chip carrier. Typically the PCC has lower thermal characteristics than the corresponding dual in-line package.

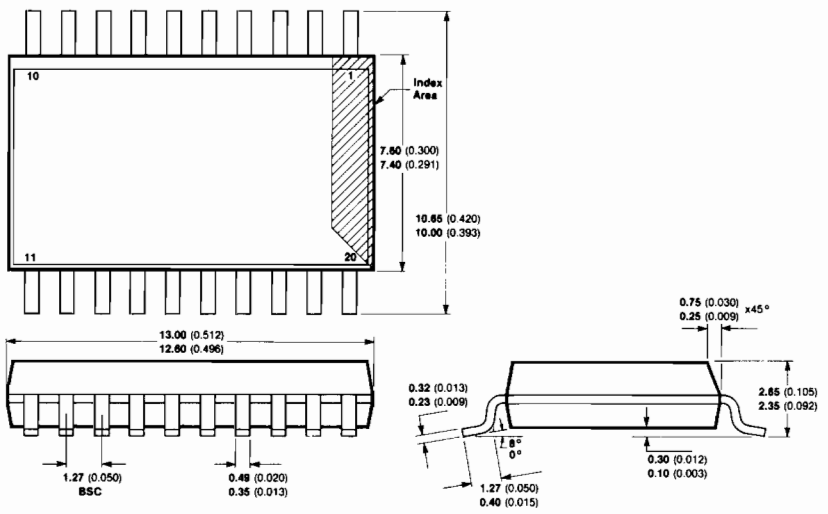


Figure 11-5 20 Lead Small Outline Integrated Circuit (SOIC)

SOIC—The Small Outline Integrated Circuit package is a miniaturized DIP with leads on 0.050 inch centers. Generally leads are bent out in a gull wing configuration so that the SOIC can be surface mounted. No plated through holes on the printed wiring board are necessary for the SOIC to be firmly attached. As with the previous SMT packages, the leads are soldered to the printed circuit board.

PGA/LGA—With lead counts in excess of 68 for VLSI devices, the Pin Grid Array and the Leadless Grid Array use a square array of pins on 0.050 or 0.100 inch centers to reduce the package footprint. The PGA is a through hole mounted package while the LGA is a surface mounted package. However, surface mounting of the LGA requires blind solder joints which can cause reliability or inspection problems.

THERMAL MANAGEMENT—This is the design function where the temperatures of the semiconductor and other devices are analyzed and constrained to be below certain values. Temperature varies depending on the application. Thermal management becomes increasingly important when using surface mount devices because of the high heat density obtained.

ELECTRICAL CHARACTERISTICS—There are several electrical characteristics involved with all packages. As speed and density increase, these characteristics become very important; inductance and DC resistance are calculated and presented in the Electrical Properties Section.

PCB/PWB—Printed Circuit Board or Printed Wiring Board is the substrate on which most electrical packages are mounted and interconnected. With surface mount devices these printed circuit boards become more complex. However, savings may be found at the system level in terms of reduced size, weight and connector functions. There are several different types of printed circuit boards: multilayer circuit boards have more than two layers of interconnection and are somewhat costly; a two-sided circuit board has interconnection patterns on two sides of the board, top and bottom, and none internally. The plated through hole is used for interconnection on most printed circuit boards and is also used to mount through hole components. A buried or blind via is a means of interconnecting two or more layers without drilling through the whole PCB.

Electrical Properties

This section details the electrical properties of the packages used by the Fairchild Digital Unit. Specifically, ground (or power) lead inductance, DC resistance and AC resistance are given. These values are calculated from the package drawings and are not verified by measurement. All measurements are from seating plane to bond pad on chip.

The inductance is calculated from the following three formulas:

$$(1)\ L\ (\mu H) = 0.005 \times \left[ln\left(\frac{2X}{B+C}\right) + 1/2 \right]$$

(Non ferrous materials in rectangular geometries)

(E11-1)

$$(2)\ L\ (\mu H) = 0.005 \times \left[ln\left(\frac{2X}{B+C}\right) + \frac{\mu_r}{4} + 1/4 \right]$$

(Ferrous materials in rectangular geometries)

$$(3)\ L\ (\mu H) = 0.005 \times \left[ln\left(\frac{2X}{r}\right) - 3/4 \right]$$

(Non ferrous materials in cylindrical geometries - wirebonds)

Where X is length, r is radius, B and C are cross sectional dimensions (in inches) and μ_r is the relative permeability. The inductance was measured on packages, then the relative permeability was found to be approximately 20.5. With thermal and mechanical stresses, the relative permeability increases. Note that with increasing frequency, the effect of μ_r or inductance decreases.

The DC resistance is calculated from R = pX/A, where p is material resistivity, X is length and A is the cross-sectional area.

Table 11-1 Worst Case Electrical Properties of Packages

Pins	Package	Inductance (nH)	DC Resistance (Ω)	AC Resistance (Ω)
14	PDIP	5.6	0.09	0.52
14	CDIP—SSI*	6.8	0.14	0.98
14	CDIP—MSI*	7.0	0.14	0.99
14	FLATPACK*	19.7	0.26	2.95
14	SOIC	2.2	0.05	0.15
16	PDIP	6.6	0.09	0.48
16	CDIP—SSI*	15.4	0.16	1.71
16	CDIP—MSI*	14.4	0.16	1.63
16	FLATPACK*	17.1	0.36	4.81
16	SOIC	2.6	0.05	0.17
16	SOIC—WIDE	4.0	0.05	0.17
18	CDIP*	20.5	0.16	2.00
20	PDIP	8.0	0.10	0.64
20	CDIP*	25.6	0.17	2.31
20	FLATPACK*	15.8	0.13	1.16
20	LCC	1.6	0.10	0.32
20	PCC	2.5	0.05	0.16
20	SOIC	4.3	0.05	0.17
24	PDIP*	31.0	0.20	2.45
24	PDIP SLIM	8.5	0.10	0.66
24	CDIP MSI*	29.7	0.18	2.66
24	FLATPACK*	26.8	0.13	1.22
24	CDIP SLIM*	22.5	0.18	2.32
24	SOIC	4.6	0.05	0.18
28	CDIP*	34.3	0.19	2.98
28	PDIP	12.6	0.10	0.73
28	PCC	2.8	0.05	0.16
28	LCC	1.7	0.10	0.34
28	FLATPACK	31.2	0.13	1.33
28	SOIC	5.0	0.05	0.19
40	CDIP (longest lead)*	48.7	0.22	3.94
40	CDIP (shortest lead)*	10.5	0.13	1.08
40	PDIP (longest lead)	19.2	0.10	0.88
40	PDIP (shortest lead)	3.5	0.08	0.36
40	LCC	2.4	0.11	0.17
40	PCC	3.8	0.05	0.17
52	PCC	6.0	0.05	0.19
68	PCC	7.2	0.05	0.20

*Leadframe is Alloy 42

11

Thermal Considerations

Introduction

Thermal considerations play an increasingly important role in electronic packaging. The advent of surface mounted components and technology is making thermal analysis mandatory in many situations. This section will define terms, point out some possible pitfalls for the electronics packager and present recent test data.

Thermal analysis is time-consuming, expensive and, at times, problematic. But without proper analysis, products may exhibit low mean time between failure and sporadic faults. Proper analysis should predict junction temperature, allowing reliability calculations based on the Arrhenius equations. Knowing mean junction temperatures and statistical variations allows more accurate thermal management techniques. If a particular technique is applicable to all but a small number of devices because of high temperatures, thermal analysis allows alternatives to be evaluated wisely. For complex analysis, such as finite difference or finite element work, experienced personnel should be consulted.

Various schemes implemented at Fairchild's Digital Unit and the concept of using a consistent philosophy in all thermal work will be presented. The assumptions made in the thermal work, whether implicit or explicit, will be discussed as appropriate.

The following terms and definitions will be used throughout this section.

THERMAL RESISTANCE—A value representing the resistance in the analogous electrical equations. Thermal resistance is often reported in degrees celsius per watt (C/W). Steady-state thermal parameters are concerned with thermal resistance of several different types: conduction, convection and radiation modes of heat transfer. θ_{JC} and θ_{JA} are examples of thermal resistances.

THERMAL ANALOGY—An easy way to show thermal parameters and equations. The basic thermal equation is $T = q \bullet (Rth)$. Similarities are evident in comparing this equation to Ohm's Law, $V = IR$. T represents the change in temperature, in °C or °F, which, as the forcing function, is analogous to voltage. The power, q, is analogous to current and has the units of power, watts, or BTU per hour. Rth, the thermal resistance, is the analogy to electrical resistance. Other analogies can be found: the thermal capacity is the analogy to capacitance in electronics. Both store energy but in different forms.

HEAT TRANSFER MODES—Three different modes of heat transfer exist:

Conduction—The simplest of the three modes, follows the Fourier equation $q = kA\ \delta T/\delta X$ where k is the thermal conductivity, q is power, A is area and T is the temperature difference a material requires for the heat energy to move from one potential to another.

Convection—The transfer of heat through the use of a gas or liquid gravitational field or through forced movements. Convection is a complex phenomenon because the coefficient h, unlike the "simple" material property, k, is complex and dependent on fluid mechanics and other properties. The convection equation is $T = q \bullet 1/hA$ where h is the convection coefficient and A the surface area. Thus, increasing the surface area decreases the thermal resistance (1/hA). In some calculations, h is explicitly expressed as a function of temperature to a fractional exponent.

Radiation—The transfer of heat using electromagnetics. The amount of radiation is given in accordance with Planck's blackbody law. Radiation transfer between two objects needs no medium for heat transfer, only a difference in temperature between the objects. This is not a strict definition of radiation, but it is convenient for heat transfer. Other considerations involve emissivity, form factors and reflectivity. The equations are not given here but it should be noted that heat transfer is a function of absolute temperature raised to the fourth power.

POWER—The analog to current in Ohm's Law. Power in electrical devices is the electrical power dissipated by the device and is expressed in watts.

TEMPERATURE DIFFERENCE—The difference in temperature between two or more objects, background, etc., in °C or °F.

THERMAL CAPACITANCE—The ability of a thermal circuit to store energy; usually expressed in Joules/°C or BTU/°F. It is the thermal analog of capacitance in electronics. Transient analysis makes extensive use of this parameter to predict the flow of heat as a function of time. The thermal capacitance for semiconductor packaging is not usually specified. It is found by multiplying the specific heat by the density and volume of the material under consideration.

STEADY STATE ANALYSIS—The type of analysis most often performed in thermal management of electronic circuits. Steady state refers to a situation in which heat transfer (and related phenomena) are invariant with time. Analyses can be performed with calculator and pencil, personal computers, or large computers with elaborate finite element (or finite difference) codes. At today's level of electronics, steady state analysis predicts temperatures fairly accurately. As the level of complexity and heat flux increases still further, a less conservative approach taking duty cycles, statistical power distributions and other transient phenomena into consideration may be required.

TRANSIENT ANALYSIS—Thermal analysis which deals with heat transfer, as a function of time. Thermal resistances and capacitances form a thermal network analogous to an electrical RC circuit. This type of analysis is much more expensive and time-consuming than steady state analysis and should only be performed where necessary.

THERMAL CONDUCTIVITY—A material property occasionally expressed in useful units. Several different units are used to express this property: the most common units are w/m°C or Btu/hr.ft.F. The symbol k is used for thermal conductivity in the Fourier Law of Conduction. Thermal conductivity varies with temperature in most cases.

HEAT FLUX—Heat flux is heat per unit area (or volume). High thermal gradients can result from high values of heat flux. Sometimes it is used to evaluate thermal management schemes. The units are w/in^2 or w/in^3.

JUNCTION TEMPERATURE—The temperature of the hottest junction on a component of an integrated circuit. It is very difficult to measure accurately a true junction temperature without the experimental setup or procedure interfering with desired results. The maximum junction temperature for most silicon devices is 175°C unless other factors lower it even further, i.e., bimetallic formations. To complicate the issue further, on LSI devices the hottest junction may change with different uses or programming. However, in most logic circuits, the junction of interest is the output driver circuit or nearby. The proper procedure to measure junction temperature must be evaluated carefully.

REGIONAL JUNCTION TEMPERATURE—Temperature of the region surrounding the junction, generally lower than the maximum junction temperature. The region is defined to be within six equivalent radii of the hottest junction by MIL-STD 883.

CASE TEMPERATURE—The temperature of the hottest area on the outside of the semiconductor package. Generally in ceramic packages this is directly beneath the die. In encapsulated packages such as plastic DIPs, this location is extremely variable and depends on the mounting scheme and airflow.

BOARD TEMPERATURE—The temperature of the substrate directly beneath the semiconductor. Measurement of this temperature is relatively straightforward, and is relevant with high heat flux situations or when attempting to decouple the component from the substrate for thermal analysis.

FREE CONVECTION—Often referred to as still air. Free convection is not the same as still air. Although no fan is used, convection currents can be substantial. Free convection must take place in a gravitational field and utilize a liquid or gas for mass transport (which transports the heat). No mechanical means is used to enhance this mass transport.

FORCED CONVECTION—Convection enhanced or accomplished by mechanical movement of gas or liquid. This type of convection does not require the presence of a gravitational field. Most often a fan is used.

Figure 11-6 Package in Oil Bath

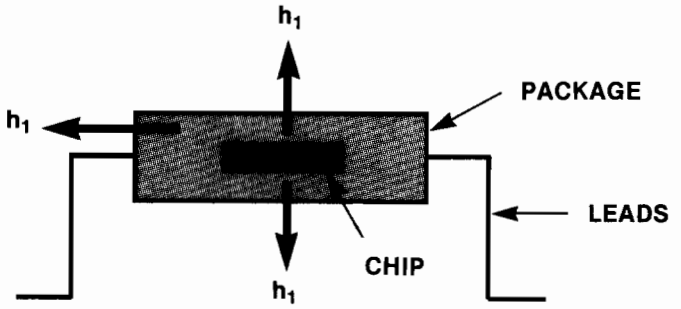
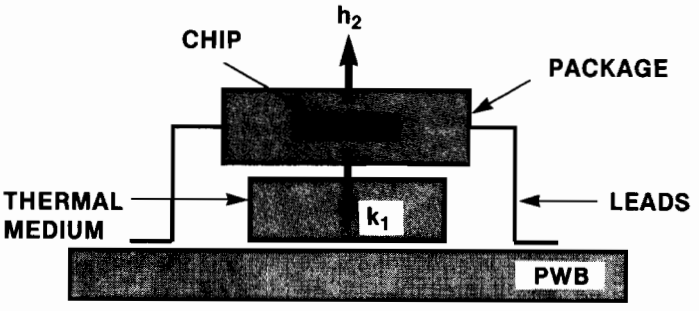


Figure 11-7 Package in Actual Use



11

JUNCTION TO CASE THERMAL RESISTANCE (Θ_{JC})— The junction to case thermal resistance of a semiconductor package is defined as the difference in temperature between the case and the junction, divided by the dissipated power. This parameter is often defined as the best heat transfer that can be accomplished with a particular package. It is also thought that the Θ_{JC} does not vary with mounting conditions. This is untrue. Figure 11-6 shows the heat flows of a package in an oil bath. If the package is mounted as in Figure 11-7 and forced convection applied, then the heat flows change substantially, possibly resulting in a different Θ_{JC}. Note that the convection coefficients h_1 and h_2 are different. Plastic packages show most change. Other variables are discussed below.

JUNCTION TO AMBIENT THERMAL RESISTANCE (Θ_{JA})— The junction to ambient thermal resistance of a semiconductor package is defined as the difference in temperature between the junction and the ambient air, divided by the dissipated power. It is very dependent upon mounting conditions and should be used carefully. The test mounting must be compared to actual use and corrections applied to resolve any differences. Airflow rates must be given. Factors affecting Θ_{JA} are numerous and are discussed more fully below.

Procedures for Thermal Resistance Measurements

Unfortunately, thermal resistance values of different manufacturers do not correlate well. Thus, the experimental conditions should be examined closely to determine if they are similar to system use.

Thermal Resistance—Junction to Case

The junction to case thermal resistance can be measured with several accepted methods. Two of the most common are the heat sink and oil bath procedures. Fairchild utilizes the oil bath procedure which is described below.

Two oil baths are used: one at room temperature (cold) and the other 10 to 20°C higher (hot) depending on the device to be tested. Power and ground are connected and the voltage across a diode on one of the outputs is monitored when biased by a 1mA current. This diode voltage is the Temperature Sensitive Parameter (TSP), the method used to determine the junction temperature. The power is supplied in a cycle with the power on 99.9% of the cycle: 65 milliseconds on, 55 microseconds off. With the power off the diode voltage is measured and displayed.

First the unpowered device is placed in the hot oil bath and allowed to soak. After no further changes are observed in the TSP, the voltage and temperature of the bath are recorded. Then the device is placed in the cold bath and powered up. After the TSP has settled, the diode voltage, the case temperature and the supplied power are recorded. The device is then turned off. After soaking in the cold bath, the TSP and cold bath temperature are recorded again.

The variation of the TSP is calculated from the difference between the measurements taken while the device is unpowered. In this case the TSP is assumed to be linear and has the units of mV/°C. The TSP is used to determine the junction temperature for the powered condition. The thermal resistance is calculated from

(E11-2)
$$\frac{T_J - T_C}{q} = \Theta_{JC}$$

where T_J is the junction temperature, T_C is the case temperature and q is the power.

Thermal Resistance—Junction to Ambient

The junction to ambient thermal resistance is much more procedure/equipment dependent than the junction to case measurements. The actual procedure used is important in deciding if the junction to ambient values apply to a particular packaging scheme.

The apparatus consists of a 6.0 inch (inside diameter) wind tunnel. A thermocouple measures the ambient temperature. A hot wire anemometer is used to determine windspeed.

The printed circuit board is 3.0 x 3.0 inches and is mounted in a nonconducting plastic card guide for thermal isolation from the wind tunnel walls. The method of mounting parts on the PCB depends on the package of the device under test. DIPs are mounted in low profile sockets, while surface mount packages are reflow soldered to the appropriate land pattern. Flatpaks are epoxied onto the board and leads attached to cable interfaces. The surface mount packages are mounted as closely as possible to actual device use because of the critical nature of thermal management in a closely packed system. No special thermal enhancement techniques are used in order to obtain worst case values.

The procedure is similar to the junction to case measurement procedure. The PCB is placed in the wind tunnel with minimal power (less than 1mw) and the TSP and ambient temperature are measured. The devices are then powered up, one at a time. Once the TSP has settled, the voltage and ambient temperature are measured. The device is then shut off and the next device powered until all devices are tested in the free convection mode. The wind tunnel is then turned on and measurements made at 250, 500, 1000 and 1500 LFPM (Linear Feet Per Minute), measured by the anemometer.

After forced air measurements are complete, the PCB and devices are placed in a hot oil bath as in the junction to case procedure. With power off, the TSP is measured to provide the other end point. The TSP is calculated as in the junction to case procedure and junction temperatures calculated for each measurement point. Thermal resistance is calculated from

(E11-3) $$\frac{T_J - T_A}{q} = \Theta_{JA}$$

where T_A is the ambient temperature, T_J is the junction temperature and q is the power as measured at the time of test.

Tabulation of Thermal Resistance

This section will show graphically the thermal resistance measurements made with the above procedures. Some graphs present a "top down" look at the relationship of thermal resistance and pin count for Fairchild parts. Several graphs will detail the relationship of thermal resistance and die area (in square inches unless otherwise noted). The proper use of the graphs will be illustrated with examples.

A note of caution: test results vary for many different reasons. From manufacturer to manufacturer results do not stay constant for a particular device. However, even with the same manufacturer's product, results may vary due to die size, conditions of manufacture, different lots, die attach, redesign or refinement of die, etc. These parameters are extensive and testing continues in order to specify thermal resistance for varying factors.

Figures 11-8 to 11-10 are a top down look at Fairchild thermal resistance by pin count. Figure 11-8 shows Θ_{JA} for PDIPs. Figure 11-9 shows Θ_{JA} for CDIPs and Figure 11-10 illustrates the SOICs. Each figure has thermal resistances for free convection and 500 LFPM.

Figure 11-11 illustrates Θ_{JC} of 20 pin LCCs and varying die area. However, a parameter called extended die area is plotted, not actual die area. Extended die dimensions are the die dimensions plus 0.020 inch added to account for heat spreading. The extended area is merely the X and Y extended dimensions multiplied together. This parameter was one used for LCC computations. The thermal resistance is given by the point corresponding to the appropriate extended area. The two curves on the graph represent minimum and maximum values due to varying conditions of manufacture.

Figure 11-8 Thermal Resistance for PDIP

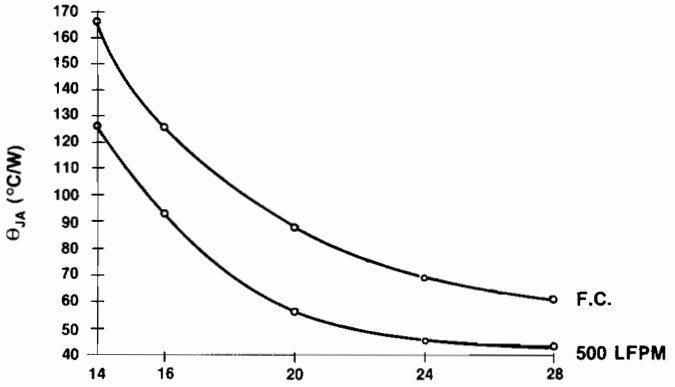
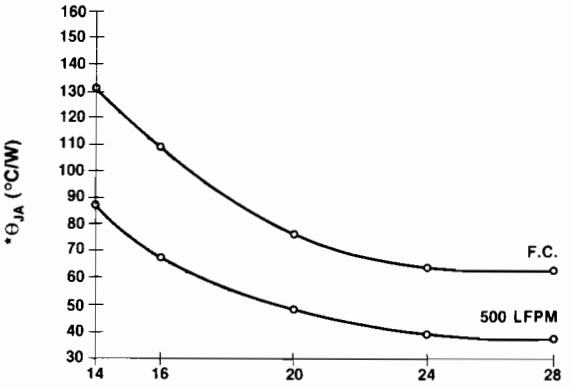


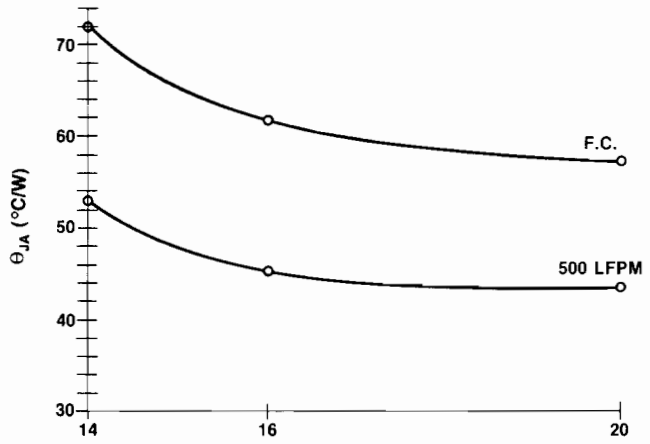
Figure 11-9 Thermal Resistance for CDIP



*Maximum of the means

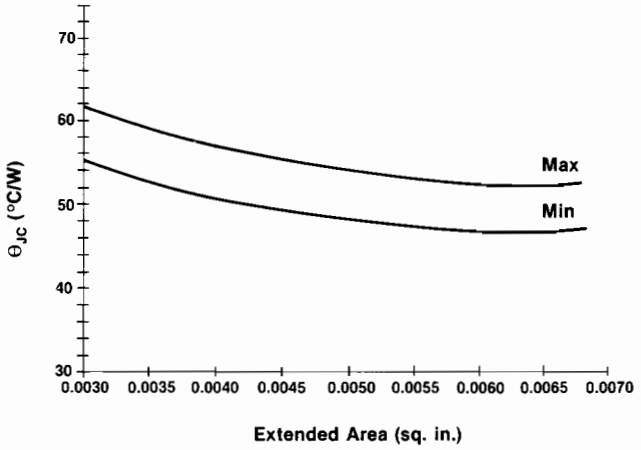
11

Figure 11-10 Thermal Resistance for SOIC



*Maximum of Means

Figure 11-11 Junction to Case Resistance for 20 Terminal LCC



Example: The 'F74 is placed in a leadless chip carrier. Find Θ_{JC}.

Solution: Figure 11-11 is the pertinent graph. The die size of the 'F74 is 0.051 by 0.058. The extended area is 0.071•0.078 = 0.0055 sq. in. The midpoint of the range is 54°C/W. This is the value for Θ_{JC}.

Figure 11-12 through 11-15 give Θ_{JC} as a function of die area, not extended area, for 14, 16, 20 and 24 pin CDIPs. Figure 11-16 and 11-17 give Θ_{JC} as a function of area for 14/16 and 20 pin PDIPs. The 14 and 16 pin PDIPs are essentially the same package and are shown as such.

Example: The 'F373 is placed in a 20 pin DIP. Find Θ_{JC} for both CDIP and PDIP.

Solution: The pertinent graphs are Figures 11-14 and 11-17. The die size of the 'F373 is 0.076 by 0.084 inches and the area is 0.076•0.084 = 0.0064 sq. in. The corresponding Θ_{JC} for PDIP is 33°C/W.

Figure 11-12 Range of Θ_{JC} for 14 Pin CDIPs

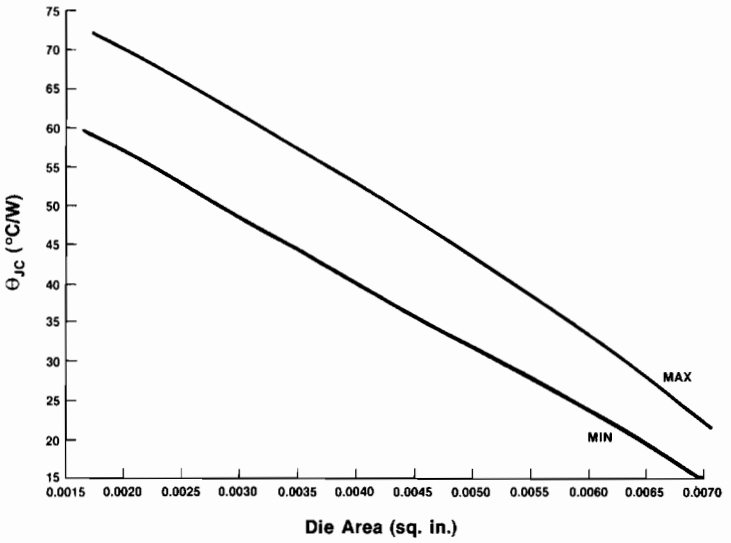


Figure 11-13 Range of Θ_{JC} for 16 Pin CDIPs

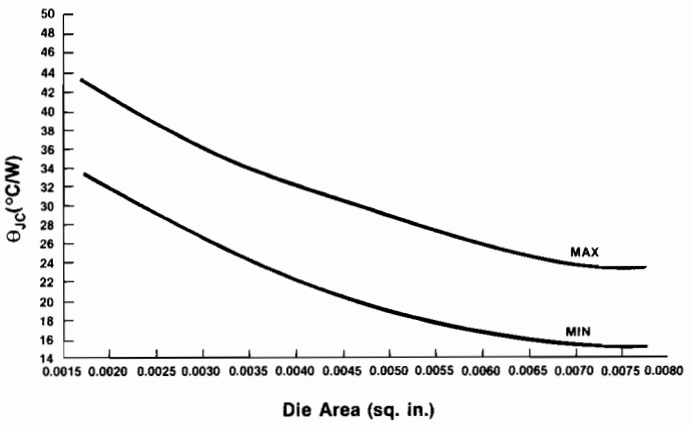


Figure 11-14 Range of Θ_{JC} for 20 Pin CDIPs

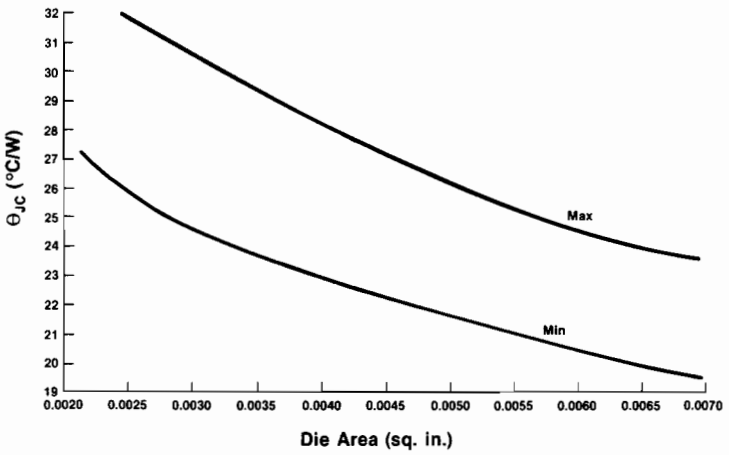


Figure 11-15 Range of Θ_{JC} for 24 Pin CDIPs

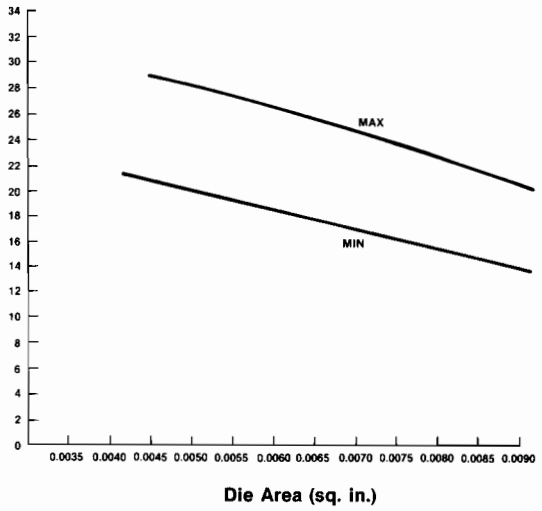


Figure 11-16 Range of Θ_{JC} for 14/16 Pin PDIPs

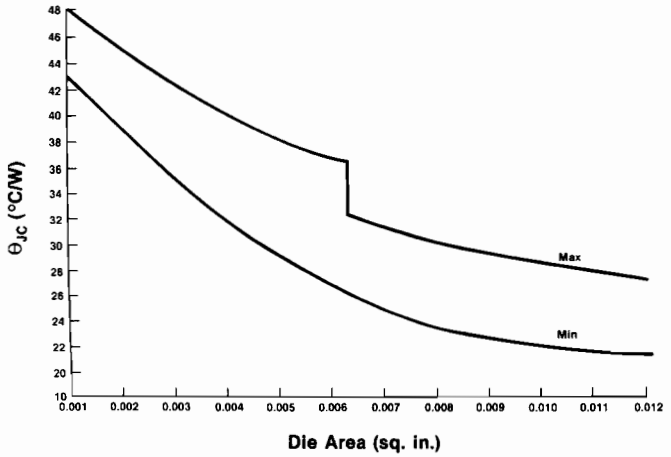
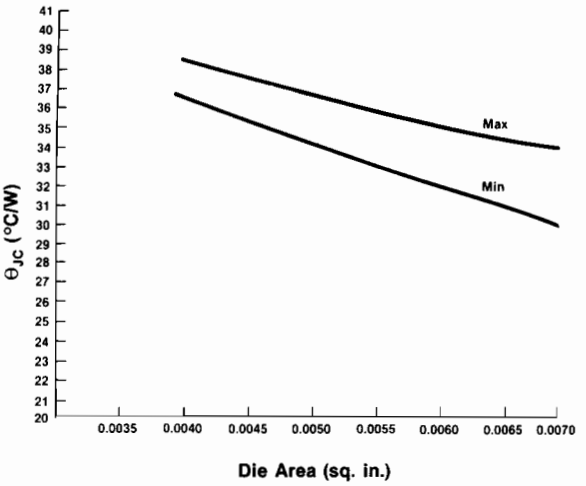


Figure 11-17 Range of Θ_{JC} for 20 Pin PDIPs



Example: The 'F74 is placed in a 14 pin PDIP. Find the Θ_{JC}.

Solution: As mentioned above, the die size is 0.051 by 0.058 and the die area is $0.051 \bullet 0.058 = 0.0030$ sq. in. The pertinent graph is Figure 11-16. Θ_{JC} is 41°C/W. The thermal resistance for 14/16 PDIPs has an odd curve because there are two different paddle sizes for these PDIPs. The larger the paddle, the lower the thermal resistance. If the die is nearly square use the upper portion of the curves.

Figures 11-18 and 11-19 give the Θ_{JA} for 14 and 16 pin CDIPs. They are actually two distinct packages. Figure 11-20 gives the Θ_{JA} for 14/16 pin PDIPs. All of these graphs show thermal resistance for free convection.

11

Example: The 'F74 is packaged in a 14 pin PDIP. Find Θ_{JA} at free convection.

Solution: As mentioned above, the die area is 0.0030 sq. in. The pertinent graph is Figure 11-20. The corresponding Θ_{JA} is 143°C/W.

Figure 11-21 to 11-24 give Θ_{JA} for 14, 16 narrow, 16 wide and 20 wide SOICs as a function of die area. All of these graphs show thermal resistance at free convection.

Figure 11-18 Range of Θ_{JA} for 14 Pin CDIPs

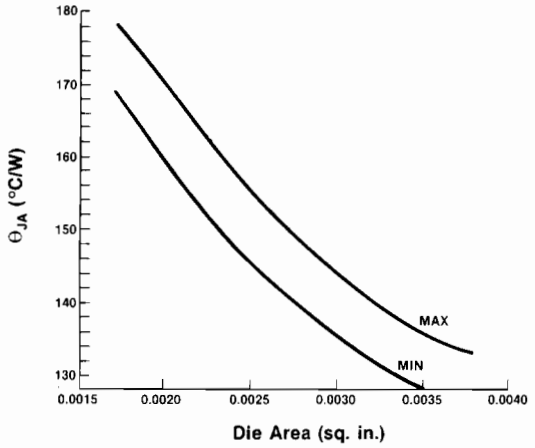


Figure 11-19 Range of Θ_{JA} for 16 Pin CDIPs (Free Convection)

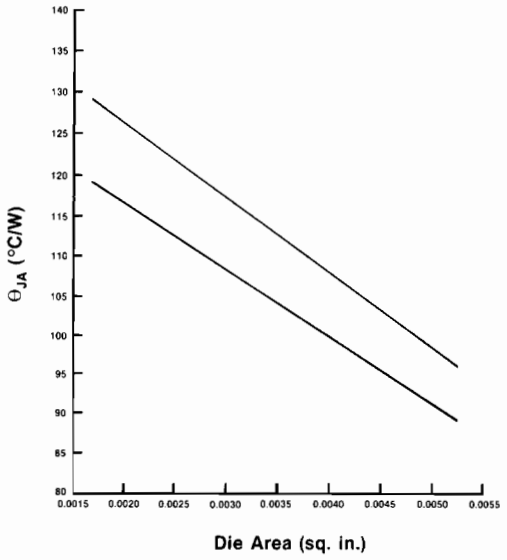


Figure 11-20 Range of Θ_{JA} for 14/16 Pin PDIPs (Free Convection)

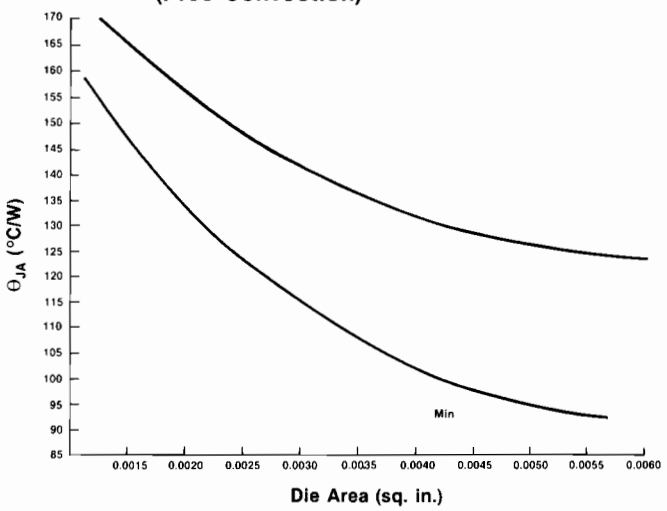


Figure 11-21 Range of Θ_{JA} for 16 Pin CDIPs (Free Convection)

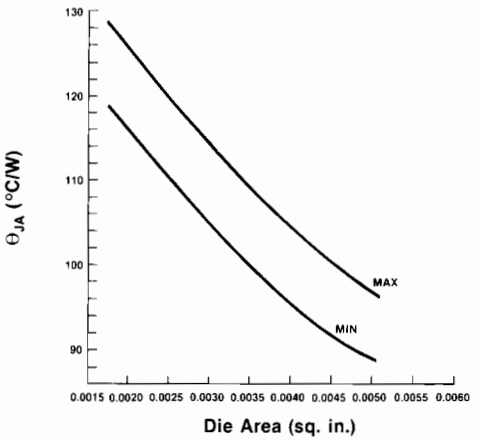
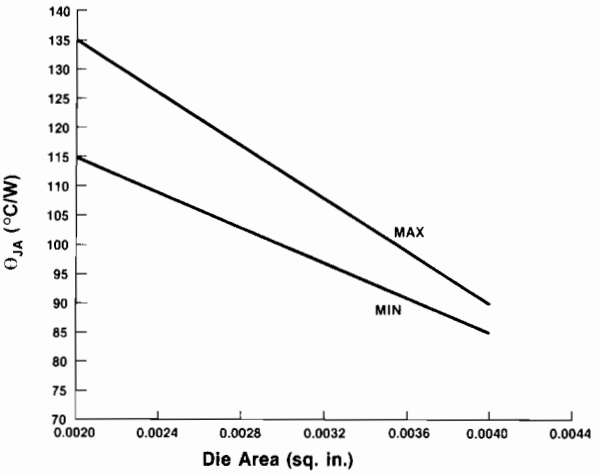


Figure 11-22 Range of Θ_{JA} for 14 Lead SOIC



Example: The 'F373 is packaged in a 20 pin SOIC. Find Θ_{JA} at free convection.

Solution: The die size is 0.0064 sq. in. and the pertinent graph is Figure 11-24. The corresponding Θ_{JA} is 79°C/W.

Figure 11-23 Range of Θ_{JA} for 16 Pin SOIC

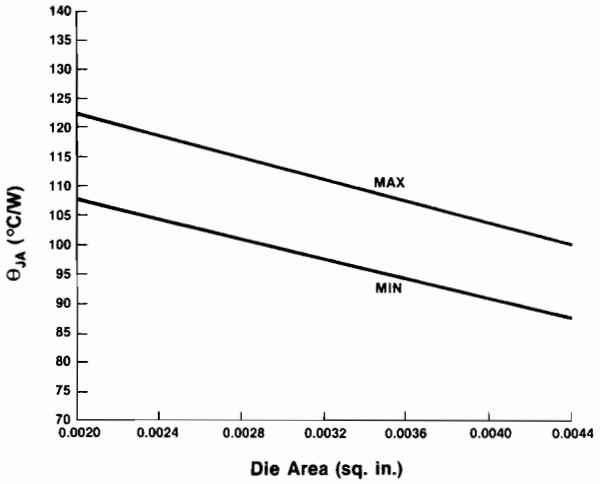


Figure 11-24 Range of Θ_{JA} for 16L Lead SOIC

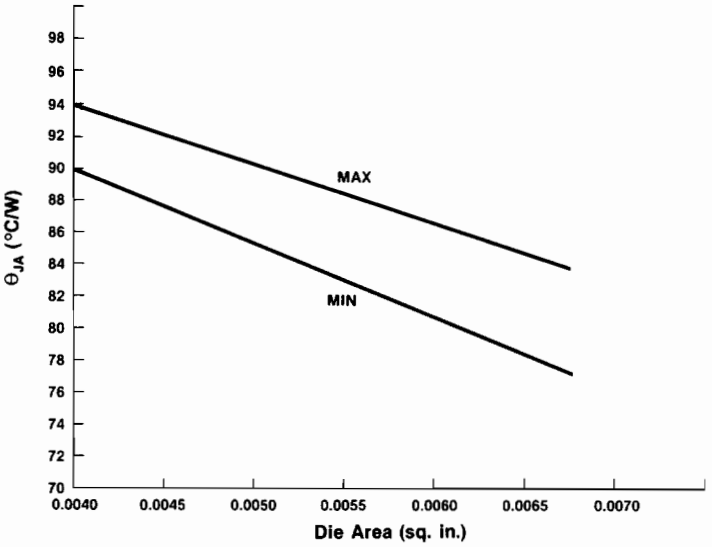
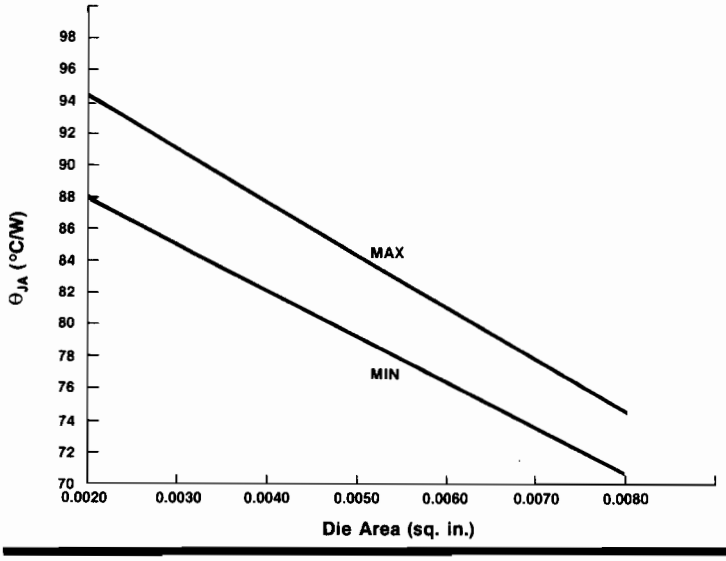


Figure 11-25 Range of Θ_{JA} for 20 Lead SOIC



How to Calculate Power Dissipations

The examples in this section will use FAST devices but the results can be extended to other bipolar devices as well.

Different specifications for military and commercial devices cause different power dissipations for each. Maximum power is used to find the hottest junction possible for a given configuration. Typical power is generally the median power. For either of these calculations, I/O power must be added to show power dissipated within the device when it acts as a current sink for other devices. Omitting the I/O power can cause significant error in junction temperature calculations.

First, typical and maximum power should be calculated from data book specifications. The I_{CC} (typ) specification gives the typical power supply current while the I_{CC} (max) spec gives maximum power supply current. Multiplying I_{CC} (typ) by 5.0 volts gives the typical power delivered. Similarly, multiplying I_{CC} (max) by 5.5 volts gives maximum power delivered.

Examples: 'F74 and the 'F373

'F74

DC Characteristics over Operating Temperature Range (unless otherwise specified)

Symbol	Parameter	54F/74F			Units	Conditions
		Min	Typ	Max		
I_{CC}	Power Supply Current		10.5	16.0	mA	V_{CC} = Max, V_{CP} = 0 V

The typical power is 10.5 mA • 5.0 V = 52.5 mW.
The maximum power is 16.0 mA • 5.5 V = 88.0 mW.

'F373

DC Characteristics over Operating Temperature Range (unless otherwise specified)

Symbol	Parameter	54F/74F			Units	Conditions
		Min	Typ	Max		
I_{CCZ}	Power Supply Current (All Outputs OFF)		38	55	mA	V_{CC} = Max, $\overline{OE}$ = HIGH D_n, LE = Gnd

The typical power is 38 mA • 5.0 V = 190.0 mW.
The maximum power is 55 mA • 5.5 V = 302.5 mW.

Note that some devices may have different specifications for military and commercial use.

The I/O power can be substantial and the exact magnitude depends on the actual use of the device in the system. The value calculated by the method outlined is the maximum and can be as much as 0.5 watt.

Much of this material is contained in Section 3, Ratings, Specifications and Waveforms, of the 1985 FAST Data Book. First the outputs of the device are counted. The number of Unit Loads (U.L.) for an output in the low state is determined. A unit load in this state is defined as 1.6 mA. Often the commercial rating is different than the military rating. The V_{OL} for most devices is 0.5 volts while buffer-powered drivers have a V_{OL} specified at 0.55 volts. Multiply the current by the appropriate voltage to obtain the I/O power.

Examples: 'F74 and 'F373

'F74

Input Loading/Fan-Out: See Section 3 for U.L. definitions

Pin Names	Description	54F/74F(U.L.) HIGH/LOW
D_1, D_2	Data Inputs	0.5/0.375
CP_1, CP_2	Clock Pulse Inputs (Active Rising Edge)	0.5/0.375
$\overline{C}_{D1}$, $\overline{C}_{D2}$	Direct Clear Inputs (Active LOW)	0.5/1.125
$\overline{S}_{D1}$, $\overline{S}_{D2}$	Direct Set Inputs (Active LOW)	0.5/1.125
Q_1, $\overline{Q}_1$, Q_2, $\overline{Q}_2$	Outputs	25/12.5

There are four outputs, each rated at 12.5 unit loads. No special military ratings are given. Accordingly, the power is

12.5 U.L. • 1.6 mA/U.L. • 0.5 V • 4 outputs = 40 mW

This power must be added to both the typical and maximum power calculated above to get the true typical power and true maximum power.

Typical: 52.5 mW + 40 mW = 92.5 mW
Maximum: 88.0 mW + 40 mW = 128.0 mW

'F373

Input Loading/Fan-Out: See Section 3 for U.L. definitions

Pin Names	Description	54F/74F(U.L.) HIGH/LOW
D_0-D_7	Data Inputs	0.5/0.375
LE	Latch Enable Input (Active HIGH)	0.5/0.375
$\overline{OE}$	Output Enable Input (Active LOW)	0.5/0.375
O_0-O_7	3-State Latch Outputs	25/12.5

There are eight outputs rated at 15 U.L. for commercial and 12.5 U.L. for military applications. The power is calculated as:

Military: 12.5 U.L. • 1.6 mA/U.L. • 0.5 • 8 outputs = 80 mW
Commercial: 15 U.L. • 1.6 mA/U.L. • 0.5 • 8 outputs = 96 mW

Again, these results must be added to the appropriate power calculations:

Military (typ): 190 mW + 80 mW = 270 mW
Military (max): 302.5 mW + 80 mW = 382.5 mW
Commercial (typ): 190 mW + 96 mW = 286 mW
Commercial (max): 302.5 + 96 mW = 398.5 mW

If the exact configuration is known and all the outputs will never be used in this manner, appropriate corrections may be made to this procedure.

How to Calculate Junction Temperatures

It is not difficult to calculate junction temperatures with the above information. However, the particulars about the electronic packaging at the board or system level must be known. Further, the junction to ambient thermal resistance is dependent upon the temperature of the cooling air directly around the device. If many devices are used, this can be a complex problem. Another variable is the temperature of the printed circuit board which will impact the junction temperature to some degree. As board populations increase and device powers go still higher, exact data will be needed.

For military systems, the junction to case thermal resistance is used: the hottest temperature on the case can be calculated from the temperature of the printed circuit board using simple thermal analysis programs.

The basic definitions for thermal resistance then enable either the commercial or the military user to determine the desired junction temperatures. For junction to case measurements, knowledge of the case temperature will allow the junction temperature to be calculated by:

$$T_J = (q \bullet \Theta_{JC}) + T_C$$

where T_J = the junction temperature
q = the appropriate power dissipation (see above)
Θ_{JC} = the junction to case thermal resistance
T_C = the hottest temperature on the case (or package)

For junction to ambient resistance measurements, knowledge of the ambient air temperature enables the junction temperature to be calculated by:

$$T_J = (q \bullet \Theta_{JA}) + T_a$$

where T_J = the junction temperature
q = the appropriate power dissipation (see above)
Θ_{JA} = junction to ambient resistance for the windspeed
T_a = the temperature of the ambient air

The following examples illustrate some of the possible situations that may be encountered:

Example: An 'F74 in a plastic DIP is used in a system that incorporats printed circuit boards. Cooling method is by free convection with the boards mounted horizontally. The ambient air is 35°C but the air immediately around the device is 45°C. Find the junction temperature for a typical part and a worst case part.

Solution: The junction to ambient resistance is 140°C/W, the typical power is 92.5 mW and the maximum power is 128 mW (see above). The junction temperatures are calculated as follows:

Typical: $T_J = 0.0925\ W \bullet 140°C/W + 45°C = 58°C$
Maximum: $T_J = 0.128\ W \bullet 140°C/W + 45°C = 62.9°C$ or 63°C

Example: An 'F74 in an LCC is used in a system with copper clad Invar printed circuit boards. The system is used in a satellite with the walls at 72°C. A simulation of the printed circuit board is run and the circuit board directly beneath the device is found to be 85°C. Find the maximum junction temperature.

Solution: The junction to case resistance for the 'F74 in a 20 pin LCC is 54°C/W and the maximum power is 128 mW (see above). The thermal resistance of the solder attachment is computed as follows: assume a 0.006 inch standoff, a pad size of 0.025 by 0.050 inch, and 25% voiding. The thermal resistance of one pad is:

$$Rth = L / k \bullet A$$

where Rth = thermal resistance due to conduction
k = thermal conductivity of solder (1.28 W/in-K)
A = area
L = length

Therefore,

$$Rth = (0.006\ in) / \{[1.28\ W/in\text{-}K)] \bullet [0.025\ in \bullet 0.050\ in \bullet (1 - 0.25)]\}$$

Rth (for one solder post) = 5°C/W, but there are 20 leads in parallel so the total Rth = 5°C/W / 20 leads = 0.25°C/W. The power is 128 mW so the temperature of the case is Temp rise = 128 mW • 25°C/W = 0.25°C. This value is insignificant. Therefore the junction temperature is:

$$T_J = 0.128\ W \bullet 54°C/W + 85°C = 92°C$$

Example: A 'F373 is packaged in a 20 pin CDIP. It is mounted on a printed circuit board on a heat strap. The temperature at the top of the heat strap is 45°C. The heat strap is a piece of copper which the case of the CDIP touches directly. Find the function temperature for both the typical and worst case.

Solution: The heat strap is in direct contact with the case. The junction to case thermal resistance is 23°C/W (see above). The power dissipated in the device is 0.2835 W (maximum) and 0.270 W (typical) as shown above. Therefore the junction temperature is

Maximum: $T_J = 0.3825\ W \bullet 23°C/W + 45°C = 54°C$
Typical: $T_J = 0.270\ W \bullet 23°C/W + 45°C = 51.2°C$

Note that there is a small difference in the typical compared to the maximum junction temperatures. This difference becomes more significant with higher powered chips packaged in PDIPs mounted in free convection environments. Whenever high powered component temperatures are to be used, junction temperatures should be calculated to ensure that failure rates will not be too high.

Reliability Considerations

Calculation of the Mean Time Between Failures (MTBF) is but one of the ways to use the thermal information calculated above. The MTBF is a prime guide for the system manufacturer in determining the reliability of a proposed design or modification to an existing product.

The Arrhenius Equation relates reaction rate to temperature. This is a general equation in chemistry that applies to several types of failure mechanisms in a semiconductor device. The equation is:

$$R = (R_o) \bullet \exp(-[E_a]/kT)$$

where R = the reaction rate
R_o = a constant determined by initial conditions
E_a = the activation energy for the reaction in electron-Volts
k = Boltzman's constant, 8.6E-05 eV/K
T = the absolute temperature in Kelvin (K).

The reaction rate is related to the time to reach a critical level, which for semiconductors is the level at which failure occurred. Knowing the rate can reveal the time necessary to reach the critical level (MTBF).

Comparing the MTBFs of a device at different temperatures is instructive. By appropriate manipulation of the Arrhenius Equation one arrives at:

$$\text{Acceleration Factor (AF)} = \text{MTBF (at T1)} / \text{MTBF (at T2)}$$

$$AF = \exp[(E_a/k)[1/T1-1/T2]]$$

where T1 and T2 are junction temperatures and the acceleration factor is merely the ratio of the two MTBFs.

For the general case, E_a is often 1.0 eV for semiconductor devices where the exact activation energy is unknown. Figure 11-26 shows the plot of the acceleration factor vs temperature with the AF at 25°C equal to 1.000. To find the AF between two temperatures, i.e., at 50°C and 85°C, first find the acceleration factors with reference to 25°C. These are 0.05 (for 50°C) and 0.0015 (for 85°C). Obtain the ratio of the AF to obtain the AF between the two numbers: 0.05/0.0015 = 33.3. Thus, using a device at a junction temperature of 50°C will last 33.3 times as long as using it at 85°C.

This information is often used to establish maximum junction temperatures for semiconductor devices in a system. The maximum junction temperature for all devices is 175°C. This limit is dictated by material considerations of the actual semiconductor die. Most users will derate this limit to obtain a lower MTBF. The actual MTBF (in hours) is a function of the processes and materials used in the fabrication of the device. This number is being constantly increased with improvements in processes and materials.

The recommended junction temperature for plastic devices is 150°C. Fairchild devices in plastic will normally not exceed this temperature. There are some exceptions, as with the rest of the industry, where methods and materials to handle LSI and VLSI components are still being evaluated. Improvements in materials are aiding the trend of lower junction temperatures. The use of a low stress plastic, which generally has poor thermal properties, results in a package that is superior in many respects to a package using plastic with better thermal characteristics. In most cases, derating should be used to ensure a longer system lifetime.

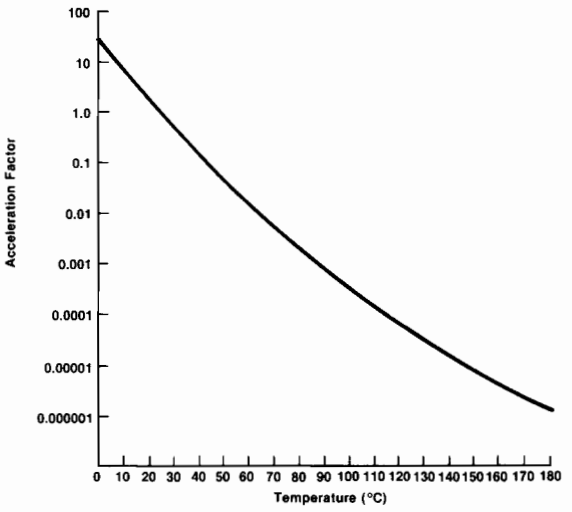


Figure 11-26 Acceleration Factor vs. Temperature

Impact of Surface Mount Technology (SMT)

Surface mount technology is not a new phenomenon. In the late 1960's and early 1970's several major military programs were designed with flatpaks mounted on PCBs in a surface mount configuration. The use of SMT allows a smaller footprint and a more effective interconnection system on the PCB. However, there are pitfalls in the technology that can cause problems if approached improperly. This section will present a philosophy that can help avoid pitfalls in the design and manufacturing process. Specifics such as land sizes, screening practices, pick and place machines, etc., will not be mentioned as these can be found in available references on SMT.

The advent of SMT as a major thrust in packaging technology coincides with the development of new components such as SOIC, PCC, and LCC. These components generally have better materials and technology than the previous DIP packages. They have an advantage because the constraints imposed today upon SMT and electronics packaging in general would quickly limit systems designed with old technology. The new components have significant size and weight savings over the previous through hole mount components. Up to six to one savings in size and weight may be obtained. And as shown previously, electrical performance is much enhanced over the larger DIP and through hole packages. Thus, SMT is used for size and weight reduction and electrical performance improvements.

As mentioned in the previous section, thermal management is as integral a part of the design as PCB layout and selection of components. Correctly designed, these components and assemblies will be reliable and save total system costs.

Surface mount components are not one-for-one replacements with present through hole mount components. Even though PCBs can be more expensive, the cost per function is dramatically lower. Generally, when using SMT, fine line multilayer PCBs are used; using LCCs also requires material changes. The thermal coefficient of expansion mismatch between LCCs and organic PCBs is one problem that must be resolved on a specific system requirements basis. The attachment of the component to the board is important. Most systems use solder as a structural medium in attaching the component. Solder has qualities such as creep and low strength that can cause problems. The solution is to link design to manufacturing and plan ahead to minimize problem areas.

Without using computer-aided design (CAD) and computer-aided engineering (CAE), the design of a surface mount PCB can be difficult. With appropriate routines such as automatic placement, self check design rules and easier types of modeling these difficult areas in design can be resolved rather easily.

VLSI is one major advance that requires SMT. Most through hole components, with the exception of the PGAs, do not lend themselves to large pin counts very readily. However, the surface mount devices for VLSI are larger than the surface mount devices for the mid range (forty to eighty-four) pin counts. These larger devices require focusing on potential problems such as thermal coefficient of expansion mismatches, power cycling and PCB reliability. The design of an SMT system must be thought out completely from design cycle to completion of manufacturing. Manufacturing personnel should be brought in early to help in design and prepare manufacturing plans. Properly designed, a system implemented with surface mount components results in one that can be automated in assembly and testing; meeting requirements for tight packaging.

Automation should play a key role in any surface mount system, especially in high volume manufacture. Most components can be found in reels and tape or tubes for automated placement. However, the designer has to pick components that can be found in the surface mount and automated configurations. Thus input from manufacturing to design and back is critical.

Some considerations for the removal and repair of surface mount devices on a PCB board assembly involve the following:

1. A controlled application of heat to reflow the solder joints. No overheating of the PCB, the device or reflowing of adjacent device solder joints can be tolerated. Thus some method should be used to determine when heating should stop.

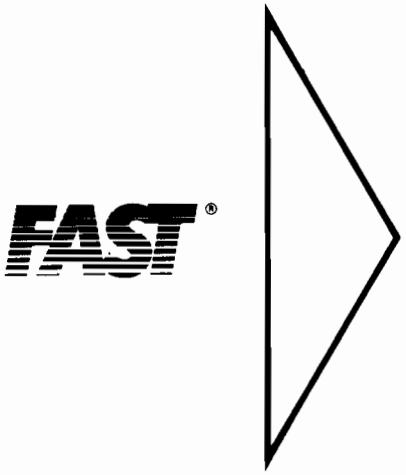
2. Not only do the joints have to be reflowed, but the reflow has to be sensed and the component removed quickly to prevent delamination of the lands from the PCB. It is important that the heating time be kept as short as possible.

3. Prepare the lands and pretin the device before placing it on the board.

4. The positioning of the device on the board is critical: it must be placed at least as accurately as the original device, at most ± 15 mils.

Other types of equipment that are important involve vapor phase soldering, modified wave soldering machines, pick and place machines of several different types and equipment and procedures to clean underneath the device. The space underneath a surface mounted device is small, thus cleaning must be performed satisfactorily or the reliability of the finished assembly will suffer.

These are a few of the considerations of using SMT. For more specific information, refer to several manuals published by professional societies. Properly used, surface mount will enhance electrical performance, reduce size and weight and provide a springboard for automating many assembly tasks previously performed by hand.



FAIRCHILD
A Schlumberger Company

Index

FAST®

Fairchild Semiconductor Sales Offices United States and Canada

Alabama
555 Sparkman Drive, Suite 1030
Huntsville, Alabama 35805
Tel: 205-837-8960

Arizona
9201 N. 25th Avenue, Suite 215
Phoenix, Arizona 85021
Tel: 602-943-2100

California
Auburn Office
3620 Sugarview Road
Meadow Vista, California 95722
Tel: 916-823-6664

Costa Mesa Office
3505 Cadillac Avenue
Suite 0-104
Costa Mesa, California 92626
Tel: 714-241-5900

Cupertino Office
10400 Ridgeview Court
Cupertino, California 95014
Tel: 408-864-6200

Encino Office
15760 Ventura Blvd., Suite 1027
Encino, California 91436
Tel: 818-990-9800

San Diego Office
4355 Ruffin Drive, Suite 100
San Diego, California 92123
Tel: 619-560-1332

Colorado
Colorado Springs Office
102 S. Tejon Street, Suite 1100
Colorado Springs, Colorado 80903
Tel: 303-578-3319

Denver Office
10200 E. Girard, Suite 222, Bldg. B
Denver, Colorado 80231
Tel: 303-695-4927

Connecticut
2440 Whitney Avenue
Hamden, Connecticut 06518
Tel: 203-288-1560

Florida
Deerfield Beach Office
450 Fairway Drive, Suite 107
Deerfield Beach, Florida 33441
Tel: 305-421-3000

Orlando Office
Maitland Colonnades
2301 Lucien Way, Suite 260
Maitland, Florida 32751
Tel: 305-875-0500

St. Petersburg Office
9800 4th Street North, Suite 206
St. Petersburg, Florida 33702
Tel: 813-577-1380

Georgia
3080 Northwoods Circle, Suite 130
Norcross, Georgia 30071
Tel: 404-441-2740

Illinois
500 Park Blvd., Suite 575
Itasca, Illinois 60143
Tel: 312-773-3133

Indiana
7202 N. Shadeland, Room 205
Indianapolis, Indiana 46250
Tel: 317-849-5412

Iowa
373 Collins Road N.E., Suite 200
Cedar Rapids, Iowa 52402
Tel: 319-395-0090

Kansas
Kansas City Office
8600 West 110th Street, Suite 209
Overland Park, Kansas 66210
Tel: 913-451-8374

Maryland
10270 Old Columbia Road, Suite R
Columbia, Maryland 21046
Tel: 301-381-2500

Massachusetts
1432 Main Street
Waltham, Massachusetts 02154
Tel: 617-890-4000

Michigan
21999 Farmington Road
Farmington Hills, Michigan 48024
Tel: 313-478-7400

Minnesota
Minneapolis Office
3600 W. 80th Street, Suite 590
Bloomington, Minnesota 55431
Tel: 612-835-3322

New Jersey
783 Riverview Drive North
Totowa, New Jersey 07512
Tel: 201-256-9006

New Mexico
2900 Louisiana N.E., Suite D
Albuquerque, New Mexico 87110
Tel: 505-884-5601

New York
Endicott Office
421 E. Main Street
Endicott, New York 13760
Tel: 607-757-0200

Fairport Office
830 Cross Keys Office Park
Fairport, New York 14450
Tel: 716-223-7700

Hauppauge Office
300 Wheeler Road
Hauppauge, New York 11788
Tel: 516-348-0900

Poughkeepsie Office
19 Davis Avenue
Poughkeepsie, New York 12603
Tel: 914-473-5730

North Carolina
5970-C Six Forks Road
Raleigh, North Carolina 27609
Tel: 919-848-2420

Ohio
Cleveland Office
6133 Rockside Road, Suite 407
Cleveland, Ohio 44131
Tel: 216-447-9700

Dayton Office
7250 Poe Avenue, Suite 260
Dayton, Ohio 45414
Tel: 513-890-5813

Oregon
6600 S.W. 92nd Avenue, Suite 27
Portland, Oregon 97223
Tel: 503-244-6020

Pennsylvania
Philadelphia Office
Willow Wood Office Center
3901 Commerce Avenue, Suite 110
Willow Grove, Pennsylvania 19090
Tel: 215-657-2711

Texas
Austin Office
8240 Mopac Expressway, Suite 270
Austin, Texas 78759
Tel: 512-346-3990

Dallas Office
1702 North Collins Blvd., Suite 101
Richardson, Texas 75080
Tel: 214-234-3811

Houston Office
9896 Bissonnet-2, Suite 470
Houston, Texas 77036
Tel: 713-771-3547

Utah
Salt Lake City Office
5282 S. 320 West, Suite D120
Murray, Utah 84107
Tel: 801-266-0773

Washington
Seattle Office
11911 N.E. First, Suite 310
Bellevue, Washington 98005
Tel: 206-455-3190

Canada
Montreal Office
3675 Sources Blvd., Suite 109
Dollard des Ormeaux
Quebec H9B 2T6
Tel: 514-683-0883

Ottawa Office
148 Colonnade Road, Unit 13
Nepean, Ontario K2D 0A8
Tel: 613-226-8270

Toronto Regional Office
7 Director Court
Building C, Unit 102
Woodbridge, Ontario L4L 4S5
Tel: 416-746-7120

Fairtech Centers
3505 Cadillac Avenue, Suite 0-103
Costa Mesa, California 92626
Tel: 714-556-TECH

10400 Ridgeview Court
Cupertino, California 95014
Tel: 408-864-6200

Maitland Colonnades
2301 Lucien Way, Suite 260
Maitland, Florida 32751
Tel: 305-875-0500

1432 Main Street
Waltham, Massachusetts 02154
Tel: 617-890-4000

3600 W. 80th Street, Suite 590
Bloomington, Minnesota 55431
Tel: 612-333-TECH

1702 Collins Blvd., Suite 101
Richardson, Texas 75080
Tel: 214-234-3811

Fairchild Semiconductor Sales Offices International

Australia
Fairchild Australia Pty Ltd
366 White Horse Road
Nunawading, Victoria 3131
Tel: (613) 877-5444 Telex: 36496

Austria and Eastern Europe
Fairchild Electronics GmbH
Assmayergasse 60
A-1120 Wien
Austria
Tel: (0222) 85-86-82 Telex: 115096

Brazil
Rua Estacio de Sa, 1144
13.080 Campinas
Sao Paulo, Brazil
Tel: (011) 55-192-416655

France
Fairchild Europe Semiconductor Hqtrs.
12 Place des Etats-Unis
B.P. 655
92542 Montrouge Cedex
Tel: (1) 47-46-61-61 Telex: 201893

Germany
Fairchild Semiconductor GmbH
Flughafen Franchtz Geb. 458
D-6000 Frankfurt/Main 75
Tel: (069) 690-56-13 Telex: 411829

Fairchild Semiconductor GmbH
Oeltzenstrasse 14
D-3000 Hanover
Tel: (0511) 178-44 Telex: 922922

Fairchild Semiconductor GmbH
Poststrasse 37
D-7250 Leonberg
Tel: (07152) 410-26 Telex: 7245711

Fairchild Semiconductor GmbH
Zweigniederlassung Neufahrn
Hans-Braun-Strasse 50
D-8056 Neufahrn
Tel: (08165) 61-80 Telex: 526770

Holland
Fairchild Semiconductor B.V.
Ruysdaelbaan 35
NL-5613 DX-Eindhoven
The Netherlands
Tel: (040) 44-69-09 Telex: 51024

Hong Kong
Fairchild Semiconductor Products
12th Floor, Austin Tower
22-26A Austin Avenue, Tsimshatsui
Kowloon, Hong Kong
Tel: 3-7235256 Telex: 11780-73531

Fairchild Semiconductor (HK) Ltd.
5/F-6/F, San Miguel Bldg.
9-11, Shing Wan Road
Tai Wai, Shatin
NT Hong Kong
Tel: 852-0-6055311
Telex: 852050511

Italy
Fairchild Semiconductor, S.p.A.
Viale Corsica 7
20133 Milano
Tel: (02) 749-12-71 Telex: 330522

Fairchild Semiconductor S.p.A.
Via Francesco Saverio Nitti 11
00191 Roma
Tel: (06) 328-75/48-328-27
Telex: 612046

Japan
Fairchild Semiconductor
Pola Shibuya Bldg.
1-15-21, Shibuya
Shibuya-Ku, Tokyo 150
Tel: (02)-4008351

Fairchild Japan Corporation
Yotsubashi Chuo Bldg.
1-4-26, Shinmachi
Nishi-Ku, Osaka 550
Tel: 06-541-6138/9

Korea
Fairchild Semiconductor Korea Ltd.
10th Floor, Life Bldg.
61 Yuido-Dong, Youngdongpo-Ku
Seoul 150
Tel: 783-3795

Scandinavia
Fairchild Semiconductor AB
Bergsunds Strand 39
S-117 38 Stockholm
Tel: (08) 84-01-70 Telex: 17759

Singapore
Fairchild Semiconductor Pte. Ltd.
74 Bukit Timah Rd. #03-01/02
Boon Siew Bldg.
Singapore 0922
Republic of Singapore
Tel: (65) 258-1944

Switzerland
Fairchild Semiconductor GmbH
Baumackerstr. 46
CH-8050 Zurich
Tel: (01) 311-42-30 Telex: 823285

Taiwan
Fairchild Semiconductor Ltd.
Hsietsu Bldg., Room 502
47 Chung Shan North Road
Sec. 3 Taipei, Taiwan
Tel: 573205

United Kingdom
Fairchild Semiconductor Ltd.
230 High Street
Potters Bar
Hertfordshire EN6 5BU
Tel: (0707) 511-11 Telex: 262835

Fairchild Semiconductor Authorized Distributors Unites States and Canada

Alabama
Hamilton/Avnet Electronics
4940 Research Dr. N.W.
Huntsville, Alabama 35805
Tel: 205-837-7210

Schweber Electronics
4930 Corporate Drive
Huntsville, Alabama 35805
Tel: 205-895-0480

Arizona
Hamilton/Avnet Electronics
505 South Madison Drive
Tempe, Arizona 84281
Tel: 602-231-5100

Schweber Electronics
11049 N. 23rd Drive, Suite 100
Phoenix, Arizona 85029
Tel: 602-997-4874

Wyle Distribution Group
17855 N. Black Canyon Hwy.
Phoenix, Arizona 85023
Tel: 602-866-2888

California
Arrow Electronics
19748 Dearborn Street
Chatsworth, California 91311
Tel: 818-701-7500

Arrow Electronics
1502 Crocker Ave.
Hayward, California 94544
Tel: 415-487-4600

Arrow Electronics
9511 Ridge Haven Court
Viewriego, California 92123
Tel: 619-565-4800

Arrow Electronics
521 Weddell Avenue
Sunnyvale, California 94086
Tel: 408-745-6600

Arrow Electronics
2961 Dow Avenue
Tustin, California 92680
Tel: 714-838-5422

Avnet Electronics
350 McCormick Avenue
Costa Mesa, California 92628
Tel: 714-754-6111 (Orange County)
213-558-2345 (Los Angeles)

Avnet Electronics
20501 Plummer Street
Chatsworth, California 91311
Tel: 818-883-0000

Hamilton/Avnet Electronics
3170 Pullman Avenue
Costa Mesa, California 92626
Tel: 714-641-1850

Hamilton Electro Sales
10912 West Washington Blvd.
Culver City, California 90230
Tel: 213-558-2000

Hamilton/Avnet Electronics
4103 North Gate Blvd.
Sacramento, California 95834
Tel: 916-920-3150

Hamilton/Avnet Electronics
4545 Viewridge Avenue
San Diego, California 92123
Tel: 619-571-7527

Hamilton Electro Sales
1361-B West 190th St.
Gardena, California 90248
Tel: 213-217-6700

Hamilton Electro Sales
9650 Desota Avenue
Chatsworth, California 91311
Tel: 818-700-6500

Hamilton Electro Sales
3002 East G Street
Ontario, California 91764
Tel: 714-989-4602

Hamilton/Avnet Electronics
1175 Bordeaux Drive
Sunnyvale, California 94086
Tel: 408-743-3355

Schweber Electronics
17822 Gillette Avenue
Irvine, California 92714
Tel: 714-863-0200

Schweber Electronics
21139 Victory Blvd.
Canoga Park, California 91303
Tel: 818-999-4702

Schweber Electronics
90 East Tasman Drive
San Jose, California 95134
Tel: 408-946-7171

*Sertech Laboratories
3170 Pullman Dr.
Costa Mesa, California 92626
Tel: 714-754-0666

Wyle Distribution Group
7382 Lampson Ave.
Garden Grove, California 92641

Wyle Distribution Group
26677 Agoura Road
Calabasas, California 91302
Tel: 818-880-9001

Wyle Distribution Group
124 Maryland Street
El Segundo, California 90245
Tel: 213-322-8100

Wyle Distribution Group
17872 Cowan Avenue
Irvine, California 92714
Tel: 714-863-9953

Wyle Distribution Group
Military Product Division
18910 Teller Avenue
Irvine, California 92715
Tel: 714-851-9953

Wyle Distribution Group
11151 Sun Center Drive
Rancho Cordova, California 95670
Tel: 916-638-5282

Wyle Distribution Group
9525 Chesapeake Drive
San Diego, California 92123
Tel: 619-565-9171

Wyle Distribution Group
3000 Bowers Avenue
Santa Clara, California 95051
Tel: 408-727-2500

Zeus Components, Inc.
1130 Hawk Circle
Anaheim, California 92807
Tel: 714-632-6880

Zeus Components, Inc.
1580 Old Oakland Rd. Suite C 205
San Jose, California 95131
Tel: 408-998-5121

Colorado
Arrow Electronics
1390 S. Potomac Street, Suite 136
Aurora, Colorado 80012
Tel: 303-696-1111

Hamilton/Avnet Electronics
8765 E. Orchard Rd., Suite 708
Englewood, Colorado 80111
Tel: 303-740-1000

Schweber Electronics
8955 E. Nichols Avenue
Englewood, Colorado 80112

Wyle Distribution Group
451 East 124th Avenue
Thornton, Colorado 80241
Tel: 303-457-9953

Connecticut
Arrow Electronics
12 Beaumont Road
Wallingford, Connecticut 06492
Tel: 203-265-7741

Hamilton/Avnet Electronics
Commerce Drive, Commerce Park
Danbury, Connecticut 06810
Tel: 203-797-2800

Schweber Electronics
Finance Drive
Commerce Industrial Park
Danbury, Connecticut 06810
Tel: 203-792-3500

Florida
Arrow Electronics
350 Fairway Drive
Deerfield Beach, Florida 33441
Tel: 305-429-8200

Arrow Electronics
1530 Bottlebrush Dr. N.E.
Palm Bay, Florida 32905
Tel: 305-725-1480

*Chip Supply
7725 N. Orange Blossom Trail
Orlando, Florida 32810
Tel: 305-298-7100

Hamilton/Avnet Electronics
6801 N.W. 15th Way
Ft. Lauderdale, Florida 33309
Tel: 305-971-2900

Hamilton/Avnet Electronics
3197 Tech Drive, North
St. Petersburg, Florida 33702
Tel: 813-576-3930

Hamilton/Avnet Electronics
6947 University Blvd.
Winter Park, Florida 32792
Tel: 305-628-3888

Schweber Electronics
317 South North Lake Blvd., Suite 1024
Altamonte Springs, Florida 32701
Tel: 305-331-7555

Schweber Electronics
2830 North 28th Terrace
Hollywood, Florida 33020
Tel: 305-927-0511

Zeus Components
1750 West Broadway, Suite 114
Oviedo, Florida 32765
Tel: 305-365-3000

Georgia
Arrow Electronics
3155 Northwoods Pkwy., Suite A
Norcross, Georgia 30071
Tel: 404-449-8252

Hamilton/Avnet Electronics
5825-D Peachtree Corners East
Norcross, Georgia 30092
Tel: 404-447-7500

*This distributor carries Fairchild die products only.

Schweber Electronics
2979 Pacific Drive, Suite E
Norcross, Georgia 30092
Tel: 404-449-9170

Illinois
Arrow Electronics
2000 Algonquin Road
Schaumburg, Illinois 60195
Tel: 312-397-3440

Hamilton/Avnet Electronics
1130 Thorndale Avenue
Bensenville, Illinois 60106
Tel: 312-860-7780

Schweber Electronics
904 Cambridge Road
Elk Grove Village, Illinois 60007
Tel: 312-364-3750

Indiana
Arrow Electronics
2495 Directors Row, Suite H
Indianapolis, Indiana 46241
Tel: 317-243-9353

Hamilton/Avnet Electronics
485 Gradle Drive
Carmel, Indiana 46032
Tel: 317-844-9333

Iowa
Arrow Electronics
373 Collins Road N.E.
Cedar Rapids, Iowa 52402
Tel: 319-395-7230

Schweber Electronics
5270 N. Park Place N.E.
Cedar Rapids, Iowa 52402
Tel: 319-373-1417

Kansas
Hamilton/Avnet Electronics
9219 Quivira Road
Overland Park, Kansas 66215
Tel: 913-888-8900

Schweber Electronics
10300 W. 103rd St., Suite 103
Overland Park, Kansas 66214
Tel: 913-492-2921

Kentucky
Hamilton/Avnet Electronics
1051-D Newtown Pike
Lexington, Kentucky 40511
Tel: 609-259-1475

Maryland
Arrow Electronics
8300 Guilford Road
Suite H, Rivers Center
Columbia, Maryland 21046
Tel: 301-995-0003

Hamilton/Avnet Electronics
6822 Oak Hall Lane
Columbia, Maryland 21045
Tel: 301-995-3500

Schweber Electronics
9330 Gaither Road
Gaithersburg, Maryland 20877
Tel: 301-840-5900

Zeus Components
8930 Rt. 108
Columbia, Maryland 21045
Tel: 301-997-1118

Massachusetts
Arrow Electronics
One Arrow Drive
Woburn, Massachusetts 01801
Tel: 617-933-8130

Gerber Electronics
128 Carnegie Row
Norwood, Massachusetts 02062
Tel: 617-329-2400

Hamilton/Avnet Electronics
10-D Centennial Drive
Peabody, Massachusetts 01960
Tel: 617-531-7430

Schweber Electronics
25 Wiggins Avenue
Bedford, Massachusetts 01730
Tel: 617-275-5100

*Sertech Laboratories
10-B Centennial Drive
Peabody, Massachusetts 01960
Tel: 617-531-8673

Zeus Components
429 Marrett Rd.
Lexington, Massachusetts 02173
Tel: 617-863-8800

Michigan
Arrow Electronics
755 Phoenix Drive
Ann Arbor, Michigan 48104
Tel: 313-971-8220

Arrow Electronics
3510 Roger B. Chafee, S.E.
Grand Rapids, Michigan 49508
Tel: 616-243-0912

Hamilton/Avnet Electronics
2215 29th Street S.E., Space A5
Grand Rapids, Michigan 49508
Tel: 616-243-8805

Hamilton/Avnet Electronics
32487 Schoolcraft
Livonia, Michigan 48150
Tel: 313-522-4700

Schweber Electronics
12060 Hubbard Avenue
Livonia, Michigan 48150
Tel: 313-525-8100

Minnesota
Arrow Electronics
5230 West 73rd Street
Edina, Minnesota 55435
Tel: 612-830-1800

Hamilton/Avnet Electronics
10300 Bren Road East
Minnetonka, Minnesota 55343
Tel: 612-932-0600

Schweber Electronics
7424 West 78th St.
Edina, Minnesota 55435
Tel: 612-941-5280

Missouri
Arrow Electronics
2380 Schuetz Road
St. Louis, Missouri 63146
Tel: 314-567-6888

Hamilton/Avnet Electronics
13743 Shoreline Court, East
Earth City, Missouri 63045
Tel: 314-344-1200

Schweber Electronics
502 Earth City Expressway
Earth City, Missouri 63045
Tel: 314-739-0526

New Hampshire
Arrow Electronics
1 Perimeter Road
Manchester, New Hampshire 03103
Tel: 603-668-6968

Hamilton/Avnet Electronics
444 E. Industrial Drive
Manchester, New Hampshire 03104
Tel: 603-624-9400

Schweber Electronics
Bedford Farms Building 2
Kilton and South River Roads
Manchester, New Hampshire 03102
Tel: 603-625-2250

New Jersey
Arrow Electronics
6000 Lincoln Drive East
Marlton, New Jersey 08053
Tel: 609-596-8000

Arrow Electronics
2 Industrial Road
Fairfield, New Jersey 07006
Tel: 201-575-5300

Hamilton/Avnet Electronics
10 Industrial Road
Fairfield, New Jersey 07006
Tel: 201-575-3390

Hamilton/Avnet Electronics
#1 Keystone Avenue
Cherry Hill, New Jersey 08003
Tel: 609-424-0100

Schweber Electronics
18 Madison Road
Fairfield, New Jersey 07006
Tel: 201-227-7880

New Mexico
Arrow Electronics
2460 Alamo Avenue S.E.
Albuquerque, New Mexico 87106
Tel: 505-243-4566

Hamilton/Avnet Electronics
2524 Baylor Drive, S.E.
Albuquerque, New Mexico 87106
Tel: 505-765-1500

New York
Arrow Electronics
25 Hub Drive
Melville, New York 11747
Tel: 516-694-6800

Arrow Electronics
20 Oser Avenue
Hauppauge, New York 11787
Tel: 516-231-1000

Arrow Electronics
P.O. Box 370
7705 Maltlage Drive
Liverpool, New York 13088
Tel: 315-652-1000

Arrow Electronics
3375 Brighton-Henrietta
Town Line Road
Rochester, New York 14623
Tel: 716-275-0300

Hamilton/Avnet Electronics
933 Motor Parkway
Hauppauge, New York 11788
Tel: 516-231-9800

Hamilton/Avnet Electronics
333 Metro Park
Rochester, New York 14623
Tel: 716-475-9130

Hamilton/Avnet Export
1065 Country Rd., Suite 211A
Westbury, NY 11590

Hamilton/Avnet Electronics
103 Twin Oaks Drive
Syracuse, New York 13207
Tel: 315-437-2642

*This distributor carries Fairchild die products only.

Schweber Electronics
Jericho Turnpike
Westbury, New York 11590
Tel: 516-334-7474

Schweber Electronics
3 Town Line Circle
Rochester, New York 14623
Tel: 716-424-2222

Summit Distributors, Inc.
916 Main Street
Buffalo, New York 14202
Tel: 716-884-3450

Zeus Components, Inc.
100 Midland Avenue
Port Chester, New York 10573
Tel: 914-937-7400

North Carolina
Arrow Electronics
5240 Greens Dairy Road
Raleigh, North Carolina 27604
Tel: 919-876-3132

Hamilton/Avnet Electronics
3510 Spring Forest Road
Raleigh, North Carolina 27604
Tel: 919-878-0819

Schweber Electronics
5285 North Blvd.
Raleigh, North Carolina 27604
Tel: 919-876-0000

Ohio
Arrow Electronics
7620 McEwen Road
Centerville, Ohio 45459
Tel: 513-435-5563

Arrow Electronics
6238 Cochran Road
Solon, Ohio 44139
Tel: 216-248-3990

Hamilton/Avnet Electronics
954 Senate Drive
Dayton, Ohio 45459
Tel: 513-433-0610

Hamilton/Avnet Electronics
4588 Emery Industrial Parkway
Warrensville Heights, Ohio 44128
Tel: 216-831-3500

Hamilton/Avnet Electronics
777 Brooksedge Blvd.
Westerville, Ohio 43081
Tel: 614-882-7004

Schweber Electronics
23880 Commerce Park Road
Beachwood, Ohio 44122
Tel: 216-464-2970

Schweber Electronics
7865 Paragon Road
Dayton, Ohio 45459
Tel: 513-439-1800

Oklahoma
Arrow Electronics
4719 S. Memorial
Tulsa, Oklahoma 74145
Tel: 918-665-7700

Schweber Electronics
4815 S. Sheridan Rd.
Tulsa, Oklahoma 74145
Tel: 918-622-8000

Oregon
Arrow Electronics
10260 S.W. Nimbus, Suite M3
Tigard, Oregon 97223
Tel: 503-684-1690

Hamilton/Avnet Electronics
6024 S.W. Jean Road
Building C, Suite 10
Lake Oswego, Oregon 97034
Tel: 503-635-8157

Wyle Distribution
5250 N.E. Elam Young Parkway, Suite 600
Hillsboro, Oregon 97124
Tel: 503-640-6000

Pennsylvania
Arrow Electronics
650 Saco Road
Monroeville, Pennsylvania 15146
Tel: 412-856-7000

Schweber Electronics
231 Gibraltor
Horsham, Pennsylvania 19044
Tel: 215-441-0600

Schweber Electronics
1000 R.I.D.C. Plaza, Suite 203
Pittsburgh, Pennsylvania 15238

Texas
Arrow Electronics
2227 W. Braker Lane
Austin, Texas 78758
Tel: 512-835-4180

Arrow Electronics
3220 Commander Drive
Carrolton, Texas 75006
Tel: 214-380-6464

Arrow Electronics
10899 Kinghurst, Suite 100
Houston, Texas 77099
Tel: 713-530-4700

Hamilton/Avnet Electronics
1807 West Braker Lane
Austin, Texas 78758
Tel: 512-837-8911

Hamilton/Avnet Electronics
4850 Wright Road, Suite 190
Stafford, Texas 77477
Tel: 713-240-7733

Hamilton/Avnet Electronics
2111 W. Walnut Hill Lane
Irving, Texas 75062
Tel: 214-659-4111

Schweber Electronics
6300 La Calma Drive, Suite 240
Austin, Texas 78752
Tel: 512-458-8253

Schweber Electronics
4202 Beltway Drive
Dallas, Texas 75234
Tel: 214-661-5010

Schweber Electronics
10625 Richmond, Suite 100
Houston, Texas 77042
Tel: 713-784-3600

Wyle Distribution Group
2120 West Braker Lane, Suite F
Austin, Texas 78758
Tel: 512-834-9957

Wyle Distribution Group
11001 S. Wilcrest, Suite 105
Houston, Texas 77099
Tel: 713-879-9953

Wyle Distribution Group
1810 N. Greenville Rd.
Richardson, Texas 75081
Tel: 214-235-9953

Zeus Components, Inc.
1800 N. Greenville Rd.
Richardson, Texas 75081
Tel: 214-783-7010

Utah
Arrow Electronics
1515 West 2200 South
Salt Lake City, Utah 84119
Tel: 801-972-0404

Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City, Utah 84119
Tel: 801-972-2800

Wyle Distribution Group
1959 South 4130 West, Unit B
Salt Lake City, Utah 84104
Tel: 801-974-9953

Virginia
Arrow Electronics
8002 Discovery Drive
Richmond, Virginia 23285
Tel: 804-282-0413

Washington
Arrow Electronics
14320 N.E. 21st Street
Bellevue, Washington 98005
Tel: 206-643-4800

Hamilton/Avnet Electronics
14212 N.E. 21st Street
Bellevue, Washington 98005
Tel: 206-453-5844

Wyle Distribution Group
1750 132nd Avenue N.E.
Bellevue, Washington 98005
Tel: 206-453-8300

Wisconsin
Arrow Electronics
200 North Patrick Blvd.
Brookfield, Wisconsin 53005
Tel: 414-792-0150

Hamilton/Avnet Electronics
2975 South Moorland Road
New Berlin, Wisconsin 53151
Tel: 414-784-4510

Schweber Electronics
150 Sunnyslope Road, Suite 120
Brookfield, Wisconsin 53005
Tel: 414-784-9020

Canada
Future Electronics Corporation
3220 5th Ave. NE
Calgary, Alberta T2A 5N1
Tel: 403-235-5325

Future Electronics
5312 Calgary Trail
Edmonton, Alberta P6H 4J8
Tel: 403-438-2858

Future Electronics, Inc.
82 St. Regis Crescent North
Downsview, Ontario M3J 1Z3
Tel: 416-638-4771

Future Electronics, Inc.
Baxter Center
1050 Baxter Road
Ottawa, Ontario K2C 3P2
Tel: 613-820-8313

Future Electronics, Inc.
237 Hymus Blvd.
Pointe Claire (Montreal),
Quebec, H9R 5C7
Tel: 514-694-7710

Future Electronics Corporation
1695 Boundary Road
Vancouver B.C. V5K 4X7
Tel: 604-438-5545

Hamilton/Avnet Canada Ltd.
6845 Rexwood Road, Units 3-4-5
Mississauga, Ontario L4V 1R2
Tel: 416-677-7432

Hamilton/Avnet Canada Ltd.
190 Colonnade Road
Nepean, Ontario K2E 7J5
Tel: 613-226-1700

Hamilton/Avnet Canada Ltd.
2795 Halpern Road
St. Laurent, Quebec H4S 1P8
Tel: 514-335-1000

Semad Electronics Ltd.
9045 Cote De Liesse, Suite 101
Dorval, Quebec H9P 2M9
Tel: 514-636-4614

Semad Electronics Ltd.
864 Lady Ellen Place
Ottawa, Ontario K1Z 5M2
Tel: 613-722-6571

Semad Electronics, Ltd.
85 Spy Court
Markham, Ontario L3R 4Z4
Tel: 416-475-8500

FAIRCHILD

A Schlumberger Company

Fairchild cannot assume responsibility for use of any circuitry described other than circuitry entirely embodied in a Fairchild product.

Fairchild reserves the right to make changes in the circuitry or specifications at any time without notice.

Printed in U.S.A. January 1987 305200 50M