The first PAL devices were invented at Monolithic Memories, Inc. (MMI) in 1978 by John Birkner and H. T. Chua. The inventors earned U.S. patent number 4,124,899 for their invention, and MMI rewarded them by buying them a brand new Porsche and Mercedes, respectively! Seeing the value in this technology (PAL devices, not cars), Advanced Micro Devices (AMD) acquired MMI in the early 1980s and became a leading developer and supplier of new PLDs and CPLDs. In 1997, AMD spun off its PLD operations to a subsidiary, Vantis Corporation, which they sold in 1999 to former competitor Lattice Semiconductor.

Some of the best resources for learning about PLD-based design are provided by the PLD manufacturers. Xilinx Corporation, which started in the FPGA business, also makes CPLDs, and publishes a comprehensive *Xilinx Data Book* in paper (San Jose, CA 95124, 1999) and on the Web (`www.xilinx.com`). Similarly, GAL inventor Lattice Semiconductor has a comprehensive *Lattice Data Book* (Hillsboro, OR 97124, 1999) and Web site (`www.latticesemi.com`).

A much more detailed discussion of the internal operation of LSI and VLSI devices, including PLDs, ROMs, and RAMs, can be found in electronics texts such as *Microelectronics*, second edition, by J.Millman and A.Grabel (McGraw-Hill, 1987) and *VLSI Engineering* by Thomas E. Dillinger (Prentice Hall, 1988).

On the technical side of digital design, lots of textbooks cover digital design principles, but only a few cover practical aspects of design. An excellent short book focusing on digital design practices is *The Well-Tempered Digital Design* by Robert B. Seidensticker (Addison-Wesley, 1986). It contains hundreds of readily accessible digital-design "proverbs" in areas ranging from high-level design philosophy to manufacturability.

Drill Problems

5.1 Give three examples of combinational logic circuits that require *billions and billions* of rows to describe in a truth table. For each circuit, describe its inputs and output(s) and indicate exactly how many rows the truth table contains; you need not write out the truth table. (*Hint:* You can find several such circuits right in this chapter.)

5.2 Draw the DeMorgan equivalent symbol for a 74x30 8-input NAND gate.

5.3 Draw the DeMorgan equivalent symbol for a 74x27 3-input NOR gate.

5.4 What's wrong with the signal name "READY′"?

5.5 You may find it annoying to have to keep track of the active levels of all the signals in a logic circuit. Why not use only noninverting gates, so all signals are active high?

5.6 True or false: In bubble-to-bubble logic design, outputs with a bubble can be connected only to inputs with a bubble.

5.7 A digital communication system is being designed with twelve identical network ports. Which type of schematic structure is probably most appropriate for the design?

5.8 Determine the exact maximum propagation delay from IN to OUT of the circuit in Figure X5.8 for both LOW-to-HIGH and HIGH-to-LOW transitions, using the timing information given in Table 5-2. Repeat, using a single worst-case delay number for each gate, and compare and comment on your results.

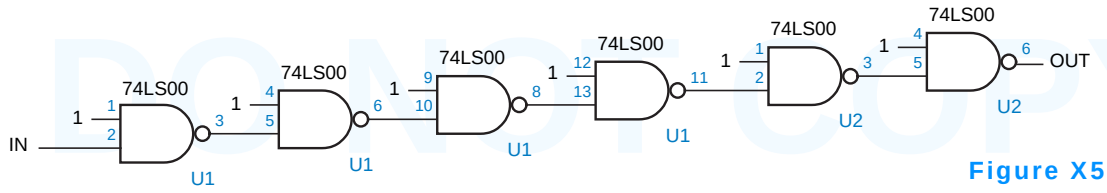


Figure X5.8

5.9 Repeat Drill 5.8, substituting 74HCT00s for the 74LS00s.

5.10 Repeat Drill 5.8, substituting 74LS08s for the 74LS00s.

5.11 Repeat Drill 5.8, substituting 74AHCT02s for the 74LS00s, using constant 0 instead of constant 1 inputs, and using typical rather than maximum timing.

5.12 Estimate the minimum propagation delay from IN to OUT for the circuit shown in Figure X5.12. Justify your answer.

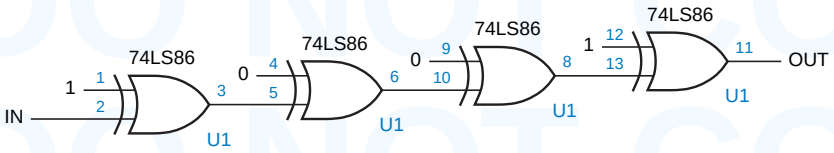


Figure X5.12

5.13 Determine the exact maximum propagation delay from IN to OUT of the circuit in Figure X5.12 for both LOW-to-HIGH and HIGH-to-LOW transitions, using the timing information given in Table 5-2. Repeat, using a single worst-case delay number for each gate, and compare and comment on your results.

5.14 Repeat Drill 5.13, substituting 74HCT86s for the 74LS86s.

5.15 Which would expect to be faster, a decoder with active-high outputs or one with active-low outputs?

5.16 Using the information in Table 5-3 for 74LS components, determine the maximum propagation delay from any input to any output in the 5-to-32 decoder circuit of Figure 5-39. You may use the "worst-case" analysis method.

5.17 Repeat Drill 5.16, performing a detailed analysis for each transition direction, and compare your results.

5.18 Draw the digits created by a 74x49 seven-segment decoder for the nondecimal inputs 1010 through 1111.

5.19 Show how to build each of the following single- or multiple-output logic functions using one or more 74x138 or 74x139 binary decoders and NAND gates. (*Hint:* Each realization should be equivalent to a sum of minterms.)

(a) $F = \Sigma_{X,Y,Z}(2,4,7)$

(b) $F = \prod_{A,B,C}(3,4,5,6,7)$

(c) $F = \Sigma_{A,B,C,D}(2,4,6,14)$

(d) $F = \Sigma_{W,X,Y,Z}(0,1,2,3,5,7,11,13)$

(e) $F = \Sigma_{W,X,Y}(1,3,5,6)$
$G = \Sigma_{W,X,Y}(2,3,4,7)$

(f) $F = \Sigma_{A,B,C}(0,4,6)$
$G = \Sigma_{C,D,E}(1,2)$

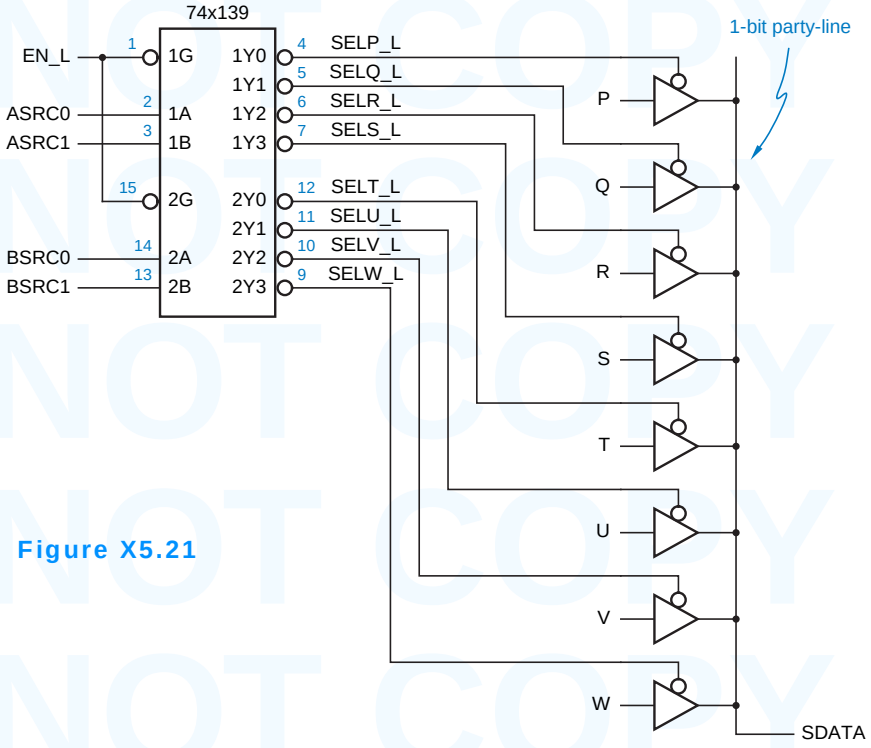


Figure X5.21

5.20 Starting with the logic diagram for the 74x148 priority encoder, write logic equations for its A2_L, A1_L, and A0_L outputs. How do they differ from the "generic" equations given in Section 5.5.1?

5.21 What's terribly wrong with the circuit in Figure X5.21? Suggest a change that eliminates the terrible problem.

5.22 Using the information in Tables 5-2 and 5-3 for 74LS components, determine the maximum propagation delay from any input to any output in the 32-to-1 multiplexer circuit of Figure 5-66. You may use the "worst-case" analysis method.

5.23 Repeat Exercise 5.22 using 74HCT components.

5.24 An *n*-input parity tree can be built with XOR gates in the style of Figure 5-74(a). Under what circumstances does a similar *n*-input parity tree built using XNOR gates perform exactly the same function?

5.25 Using the information in Tables 5-2 and 5-3 for 74LS components, determine the maximum propagation delay from the DU bus to the DC bus in the error-correction circuit of Figure 5-77. You may use the "worst-case" analysis method.

5.26 Repeat Exercise 5.25 using 74HCT components.

5.27 Starting with the equations given in Section 5.9.4, write a complete logic expression for the ALTBOUT output of the 74x85.

5.28 Write an algebraic expression for s_2, the third sum bit of a binary adder, as a function of inputs x_0, x_1, x_2, y_0, y_1, and y_2. Assume that $c_0 = 0$, and do not attempt to "multiply out" or minimize the expression.

5.29 Starting with the logic diagram for the 74x682, write a logic expression for the PGTQ_L output in terms of the inputs.

5.30 Using the information in Table 5-3 for 74LS components, determine the maximum propagation delay from any input to any output of the 16-bit group ripple adder of Figure 5-92. You may use the "worst-case" analysis method.

Exercises

5.31 A possible definition of a BUT gate (Exercise 4.50) is "Y1 is 1 if A1 and B1 are 1 *but* either A2 or B2 is 0; Y2 is defined symmetrically." Write the truth table and find minimal sum-of-products expressions for the BUT-gate outputs. Draw the logic diagram for a NAND-NAND circuit for the expressions, assuming that only uncomplemented inputs are available. You may use gates from 74x00, '04, '10, '20, and '30 packages.

5.32 Find a gate-level design for the BUT gate defined in Exercise 5.31 that uses a minimum number of transistors when realized in CMOS. You may use gates from 74x00, '02, '04, '10, '20, and '30 packages. Write the output expressions (which need not be two-level sums of products), and draw the logic diagram.

5.33 For each circuit in the two preceding exercises, compute the worst-case delay from input to output, using the delay numbers for 74HCT components in Table 5-2. Compare the cost (number of transistors), speed, and input loading of the two designs. Which is better?

5.34 Butify the function $F = \Sigma_{W,X,Y,Z}(3,7,11,12,13,14)$. That is, show how to perform F with a single BUT gate as defined in Exercise 5.31 and a single 2-input OR gate. *butification*

5.35 Suppose that a 74LS138 decoder is connected so that all enable inputs are asserted and C B A = 101. Using the information in Table 5-3 and the '138 internal logic diagram, determine the propagation delay from input to all relevant outputs for each possible single-input change. (*Hint:* There are a total of nine delay numbers, since a change on A, B, or C affects two outputs, and a change on any of the three enable inputs affects one output.)

5.36 Suppose that you are asked to design a new component, a decimal decoder that is optimized for applications in which only decimal input combinations are expected to occur. How can the cost of such a decoder be minimized compared to one that is simply a 4-to-16 decoder with six outputs removed? Write the logic equations for all ten outputs of the minimized decoder, assuming active-high inputs and outputs and no enable inputs.

5.37 How many Karnaugh maps would be required to work Exercise 5.36 using the formal multiple-output minimization procedure described in Section 4.3.8?

5.38 Suppose that a system requires a 5-to-32 binary decoder with a single active-low enable input, similar to Figure 5-39. With the EN1 input pulled HIGH, either the EN2_L or the EN3_L input in the figure could be used as the enable, with the other input grounded. Discuss the pros and cons of using EN2_L versus EN3_L.

5.39 Determine whether the a, b, and c output circuits in the 74x49 seven-segment decoder correspond to minimal product-of-sums expressions for these segments, assuming that the nondecimal input combinations are "don't cares" and BI_L = 1.

Figure X5.40

5.40 Redesign the MSI 74x49 seven-segment decoder so that the digits 6 and 9 have tails as shown in Figure X5.40. Are any of the digit patterns for nondecimal inputs 1010 through 1111 affected by your redesign?

5.41 Starting with the ABEL program in Table 5-22, write a program for a seven-segment decoder with the following enhancements:

- The outputs are all active low.
- Two new inputs, ENHEX and ERRDET, control segment-output decoding.
- If ENHEX = 0, the outputs match the behavior of a 74x49.
- If ENHEX = 1, then the outputs for digits 6 and 9 have tails, and the outputs for digits A–F are controlled by ERRDET.
- If ENHEX = 1 and ERRDET = 0, then the outputs for digits A–F look like the letters A–F, as in the original program.
- If ENHEX = 1 and ERRDET = 1, then digits A–F look like the letter S.

5.42 A famous logic designer decided to quit teaching and make a fortune by licensing the circuit design shown in Figure X5.42.

(a) Label the inputs and outputs of the circuit with appropriate signal names, including active-level indications.

(b) What does the circuit do? Be specific and account for all inputs and outputs.

(c) Draw the logic symbol that would go on the data sheet of this circuit.

(d) Write an ABEL or behavioral VHDL program for the circuit.

(e) With what standard building blocks does the new circuit compete? Do you think it would be successful as an MSI part?

5.43 An FCT three-state buffer drives ten FCT inputs and a 4.7-KΩ pull-up resistor to 5.0 V. When the output changes from LOW to Hi-Z, estimate how long it takes for the FCT inputs to see the output as HIGH. State any assumptions that you make.

5.44 On a three-state bus, ten FCT three-state buffers are driving ten FCT inputs and a 4.7-KΩ pull-up resistor to 5.0 V. Assuming that no other devices are driving the bus, estimate how long the bus signal remains at a valid logic level when an active output enters the Hi-Z state. State any assumptions that you make.

5.45 Design a 10-to-4 encoder with inputs in the 1-out-of-10 code and outputs in BCD.

5.46 Draw the logic diagram for a 16-to-4 encoder using just four 8-input NAND gates. What are the active levels of the inputs and outputs in your design?

5.47 Draw the logic diagram for a circuit that uses the 74x148 to resolve priority among eight active-high inputs, I0–I7, where I7 has the highest priority. The circuit should produce active-high address outputs A2–A0 to indicate the number of the highest-priority asserted input. If no input is asserted, then A2–A0 should be 111 and an IDLE output should be asserted. You may use discrete gates in addition to the ’148. Be sure to name all signals with the proper active levels.

5.48 Draw the logic diagram for a circuit that resolves priority among eight active-low inputs, I0_L–I7_L, where I0_L has the highest priority. The circuit should produce active-high address outputs A2–A0 to indicate the number of the highest-priority asserted input. If at least one input is asserted, then an AVALID output should be asserted. Be sure to name all signals with the proper active levels. This circuit can be built with a single 74x148 and no other gates.

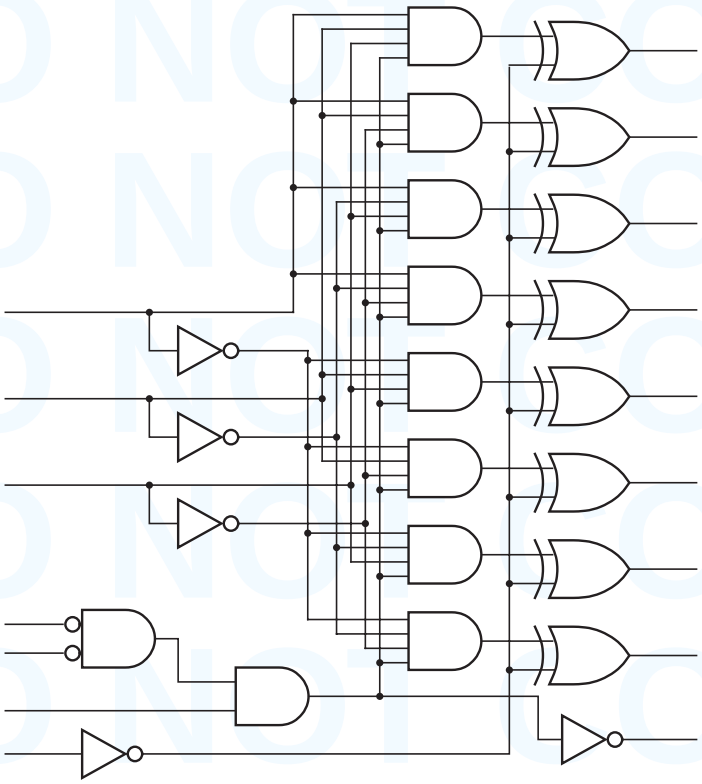


Figure X5.42

5.49 A purpose of Exercise 5.48 was to demonstrate that it is not always possible to maintain consistency in active-level notation unless you are willing to define alternate logic symbols for MSI parts that can be used in different ways. Define an alternate 74x148 symbol that provides this consistency in Exercise 5.48.

5.50 Design a combinational circuit with eight active-low request inputs, R0_L–R7_L, and eight outputs, A2–A0, AVALID, B2–B0, and BVALID. The R0_L–R7_L inputs and A2–A0 and AVALID outputs are defined as in Exercise 5.48. The B2–B0 and BVALID outputs identify the second-highest priority request input that is asserted. You should be able to design this circuit with no more than six SSI and MSI packages, but don't use more than 10 in any case.

5.51 Repeat Exercise 5.50 using ABEL. Does the design fit into a single GAL20V8?

5.52 Repeat Exercise 5.50 using VHDL.

5.53 Create a VHDL type, based on IEEE 1164, that models open-collector outputs, where typing outputs together creates a wired-AND function. You should also model a pull-up resistor element such that if there is no pull-up resistor and no device is driving the bus, then an "unknown" signal is produced. Test your definitions by modeling the circuit in Figure X3.92 for all input combinations, both with and without R1 present.

5.54 Design a 3-input, 5-bit multiplexer that fits in a 24-pin IC package. Write the truth table and draw a logic diagram and logic symbol for your multiplexer.

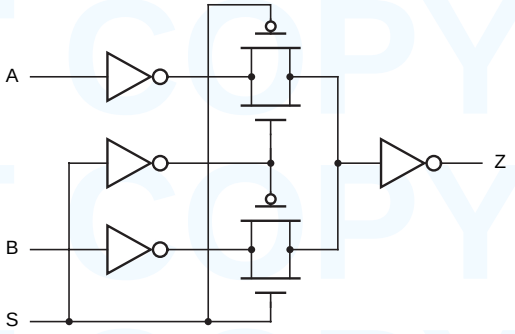


Figure X5.55

5.55 Write the truth table and a logic diagram for the logic function performed by the CMOS circuit in Figure X5.55. (The circuit contains transmission gates, which were introduced in Section 3.7.1.)

5.56 What logic function is performed by the CMOS circuit shown in Figure X5.56?

5.57 A famous logic designer decided to quit teaching and make a fortune by licensing the circuit design shown in Figure X5.57.

(a) Label the inputs and outputs of the circuit with appropriate signal names, including active-level indications.

(b) What does the circuit do? Be specific and account for all inputs and outputs.

(c) Draw the logic symbol that would go on the data sheet of this circuit.

(d) Write an ABEL or behavioral VHDL program for the circuit.

(e) With what standard building blocks does the new circuit compete? Do you think it would be successful as an MSI part?

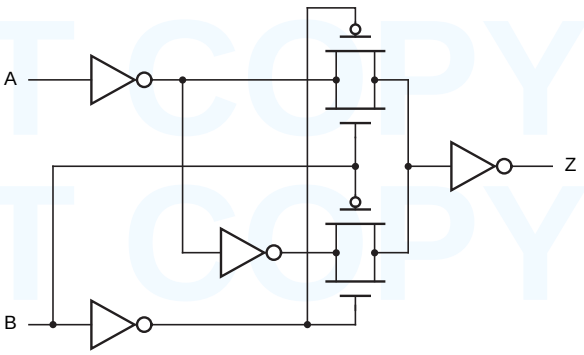
5.58 Write a VHDL program for 74x157 multiplexer with the function table shown in Table 5-35.

5.59 Write a VHDL program for 74x153 multiplexer with the function table shown in Table 5-36.

5.60 Show how to realize the 4-input, 18-bit multiplexer with the functionality described in Table 5-40 using 18 74x151s.

5.61 Show how to realize the 4-input, 18-bit multiplexer with the functionality of Table 5-40 using 9 74x153s and a "code converter" with inputs S2–S0 and outputs C1,C0 such that [C1,C0] = 00–11 when S2–S0 selects A–D, respectively.

Figure X5.56



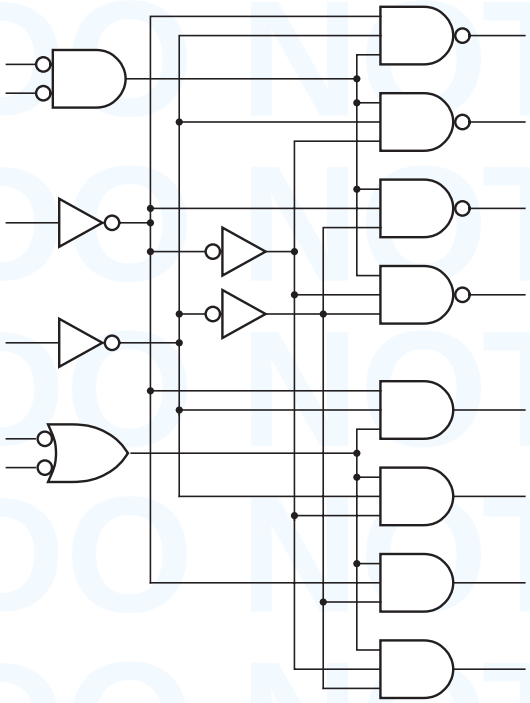


Figure X5.57

5.62 Design a 3-input, 2-output combinational circuit that performs the code conversion specified in the previous exercise, using discrete gates.

5.63 Add a three-state-output control input `OE` to the VHDL multiplexer program in Table 5-43. Your solution should have only one process.

5.64 A 16-bit *barrel shifter* is a combinational logic circuit with 16 data inputs, 16 data outputs, and 4 control inputs. The output word equals the input word, rotated by a number of bit positions specified by the control inputs. For example, if the input word equals ABCDEFGHIJKLMNOP (each letter represents one bit), and the control inputs are 0101 (5), then the output word is FGHIJKLMNOPABCDE. Design a 16-bit barrel shifter using combinational MSI parts discussed in this chapter. Your design should contain 20 or fewer ICs. Do not draw a complete schematic, but sketch and describe your design in general terms and indicate the types and total number of ICs required.

barrel shifter

5.65 Write an ABEL program for the barrel shifter in Exercise 5.64.

5.66 Write a VHDL program for the barrel shifter in Exercise 5.64.

5.67 A digital designer who built the circuit in Figure 5-76 accidentally used 74x00s instead of ’08s in the circuit and found that the circuit still worked, except for a change in the active level of the ERROR signal. How was this possible?

5.68 An odd-parity circuit with 2^n inputs can be built with 2^n-1 XOR gates. Describe two different structures for this circuit, one of which gives a minimum worst-case input to output propagation delay and the other of which gives a maximum. For each structure, state the worst-case number of XOR-gate delays, and describe a situation where that structure might be preferred over the other.

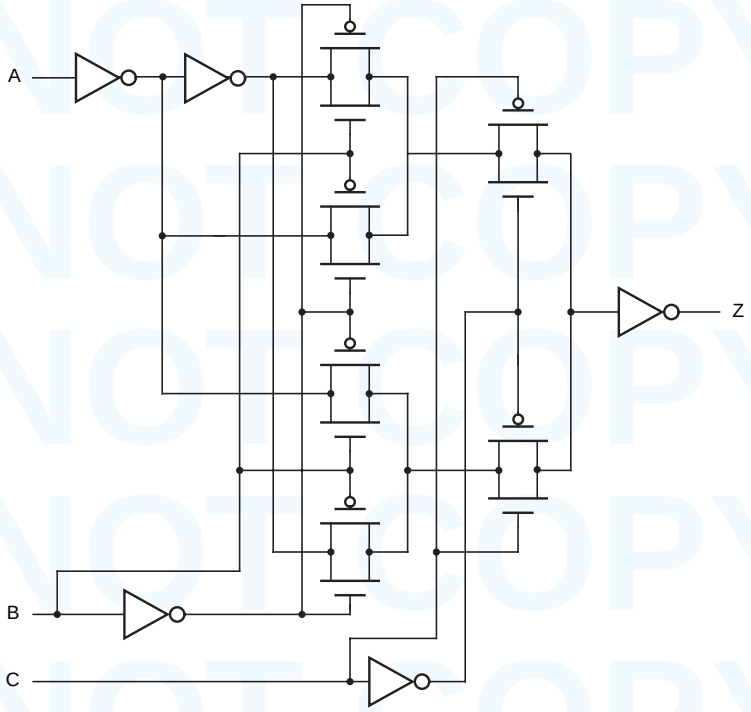


Figure X5.73

5.69 Write a 4-step iterative algorithm corresponding to the iterative comparator circuit of Figure 5-80.

5.70 Write a VHDL program for a 16-bit iterative camparator using the structure of Figure 5-80. Use VHDL's "`generate`" capability.

5.71 Design a 16-bit comparator using five 74x85s in a treelike structure, such that the maximum delay for a comparison equals twice the delay of one 74x85.

5.72 Write a VHDL program for a device with the functionality of a 74x85.

5.73 Write the truth table and a logic diagram for the logic function performed by the CMOS circuit in Figure X5.73.

5.74 Design a comparator similar to the 74x85 that uses the opposite cascading order. That is, to perform a 12-bit comparison, the cascading outputs of the high-order comparator would drive the cascading inputs of the mid-order comparator, and the mid-order outputs would drive the low-order inputs. You needn't do a complete logic design and schematic; a truth table and an application note showing the interconnection for a 12-bit comparison are sufficient.

5.75 Design a 24-bit comparator using three 74x682s and additional gates as required. Your circuit should compare two 24-bit unsigned numbers P and Q and produce two output bits that indicate whether P = Q or P > Q.

5.76 Using the information in Table 5-3, determine the maximum propagation delay from any A or B bus input to any F bus output of the 16-bit carry lookahead adder of Figure 5-96. You may use the "worst-case" analysis method.

5.77 Starting with the logic diagram for the 74x283 in Figure 5-91, write a logic expression for the S2 output in terms of the inputs, and prove that it does indeed equal the third sum bit in a binary addition as advertised. You may assume that $c_0 = 0$ (i.e., ignore c_0).

5.78 Referring to the data sheet of a 74S182 carry lookahead circuit, determine whether or not its outputs match the equations given in Section 5.10.7.

5.79 Estimate the number of product terms in a minimal sum-of-products expression for the c_{32} output of a 32-bit binary adder. Be more specific than "billions and billions," and justify your answer.

5.80 Draw the logic diagram for a 64-bit ALU using sixteen 74x181s and five 74S182s for full carry lookahead (two levels of '182s). For the '181s, you need show only the CIN inputs and G_L and P_L outputs.

5.81 Write a VHDL model for a 74x181 ALU.

5.82 Show how to build all four of the following functions using one SSI package and one 74x138.

$$F1 = X' \cdot Y' \cdot Z' + X \cdot Y \cdot Z \qquad F2 = X' \cdot Y' \cdot Z + X \cdot Y \cdot Z'$$

$$F3 = X' \cdot Y \cdot Z' + X \cdot Y' \cdot Z \qquad F4 = X \cdot Y' \cdot Z' + X' \cdot Y \cdot Z$$

5.83 Determine the worst-case propagation delay of the multiplier in Figure 5-98, assuming that the propagation delay from any adder input to its sum output is twice as long as the delay to the carry output. Repeat, assuming the opposite relationship. If you were designing the adder cell from scratch, which path would you favor with the shortest delay? Is there an optimal balance?

5.84 Repeat the preceding exercise for the multiplier in Figure 5-99.

5.85 Design a customized decoder with the function table in Table X5.85 using MSI and SSI parts. Minimize the number of IC packages in your design.

Table X5.85

CS_L	A2	A1	A0	*Output to Assert*
1	x	x	x	none
0	0	0	x	BILL_L
0	0	x	0	MARY_L
0	0	1	x	JOAN_L
0	0	x	1	PAUL_L
0	1	0	x	ANNA_L
0	1	x	0	FRED_L
0	1	1	x	DAVE_L
0	1	x	1	KATE_L

5.86 Repeat Exercise 5.85 using ABEL and a single GAL16V8.

5.87 Repeat Exercise 5.85 using VHDL.

5.88 Based on the Hamming code used in the VHDL program in Table 5-77, write a VHDL program for a Hamming encoder entity with 4-bit data inputs and 7-bit encoded data outputs.

5.89 Using ABEL and a single GAL16V8, design a customized multiplexer with four 3-bit input buses P, Q, R, T, and three select inputs S2–S0 that choose one of the buses to drive a 3-bit output bus Y according to Table X5.89.

Table X5.89

S2	S1	S0	*Input to Select*
0	0	0	P
0	0	1	P
0	1	0	P
0	1	1	Q
1	0	0	P
1	0	1	P
1	1	0	R
1	1	1	T

5.90 Design a customized multiplexer with four 4-bit input buses P, Q, R, and T, selecting one of the buses to drive a 4-bit output bus Y according to Table X5.89. Use two 74x153s and a code converter that maps the eight possible values on S2–S0 to four select codes for the 74x153s. Choose a code that minimizes the size and propagation delay of the code converter.

5.91 Design a customized multiplexer with five 4-bit input buses A, B, C, D, and E, selecting one of the buses to drive a 4-bit output bus T according to Table X5.91. You may use no more than three MSI and SSI ICs.

Table X5.91

S2	S1	S0	*Input to Select*
0	0	0	A
0	0	1	B
0	1	0	A
0	1	1	C
1	0	0	A
1	0	1	D
1	1	0	A
1	1	1	E

5.92 Repeat Exercise 5.91 using ABEL and one or more PAL/GAL devices from this chapter. Minimize the number and size of the GAL devices.

5.93 Design a 3-bit equality checker with six inputs, SLOT[2–0] and GRANT[2–0], and one active-low output, MATCH_L. The SLOT inputs are connected to fixed values when the circuit is installed in the system, but the GRANT values are changed on a cycle-by-cycle basis during normal operation of the system. Using only SSI and MSI parts that appear in Tables 5-2 and 5-3, design a comparator with the shortest possible maximum propagation delay from GRANT[2–0] to MATCH_L. (*Note:* The author had to solve this problem "in real life" to shave 2 ns off the critical-path delay in a 25-MHz system design.)