

Figure X8.4

- 8.9 Calculate the MTBF of a synchronizer built according to Figure 8-96 using 74F74s, assuming a clock frequency of 25 MHz and an asynchronous transition rate of 1 MHz. Assume that the setup time of an 'F74 is 5 ns and the hold time is zero.
- 8.10 Calculate the MTBF of the synchronizer shown in Figure X8.10, assuming a clock frequency of 25 MHz and an asynchronous transition rate of 1 MHz. Assume that the setup time t_{setup} and the propagation delay t_{pd} from clock to Q or QN in a 74ALS74 are both 10 ns.

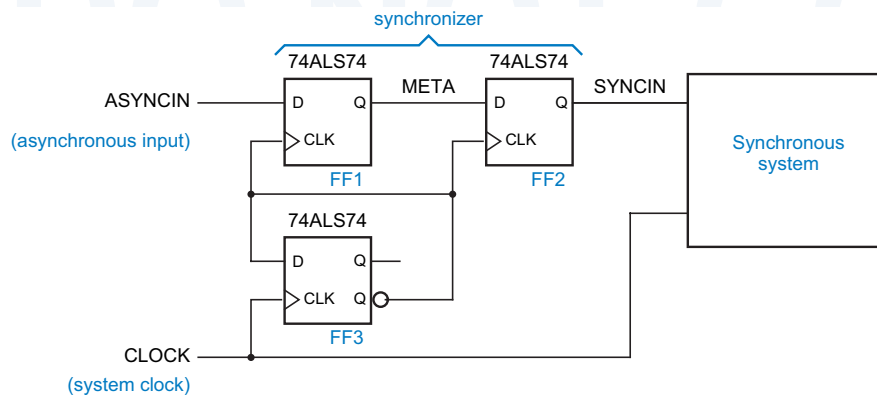


Figure X8.10

Exercises

- 8.11 What does the TTL Data Book have to say about momentarily shorting the outputs of a gate to ground as we do in the switch debounce circuit of Figure 8-5?
- 8.12 Investigate the behavior of the switch debounce circuit of Figure 8-5 if 74HCT04 inverters are used; repeat for 74AC04 inverters.

- 8.13 Suppose you are asked to design a circuit that produces a debounced logic input from an SPST (single-pole, *single-throw*) switch. What inherent problem are you faced with?
- 8.14 Explain why CMOS bus holder circuits don't work well on three-state buses with TTL devices attached. (*Hint:* Consider TTL input characteristics.)
- 8.15 Write a single VHDL program that combines the address latch and latching decoder of Figure 8-16 and Table 8-2. Use the signal name LA[19:0] for the latched address outputs.
- 8.16 Design a 4-bit ripple counter using four D flip-flops and no other components.
- 8.17 What is the maximum propagation delay from clock to output for the 4-bit ripple counter of Exercise 8.16 using 74HCT flip-flops? Repeat, using 74AHCT and 74LS74 flip-flops.
- 8.18 Design a 4-bit ripple *down* counter using four D flip-flops and no other components.
- 8.19 What limits the maximum counting speed of a ripple counter, if you don't insist on being able to read the counter value at all times?
- 8.20 Based on the design approach in Exercise 8.16 and the answer to Exercise 8.19, what is the maximum counting speed (frequency) for a 4-bit ripple counter using 74HCT flip-flops? Repeat, using 74AHCT and 74LS74 flip-flops.
- 8.21 Write a formula for the maximum clock frequency of the synchronous serial binary counter circuit in Figure 8-28. In your formula, let t_{TQ} denote the propagation delay from T to Q in a T flip-flop, t_{setup} the setup time of the EN input to the rising edge of T, and t_{AND} the delay of an AND gate.
- 8.22 Repeat Exercise 8.21 for the synchronous parallel binary counter circuit shown in Figure 8-29, and compare results.
- 8.23 Repeat Exercise 8.21 for an n -bit synchronous serial binary counter.
- 8.24 Repeat Exercise 8.21 for an n -bit synchronous parallel binary counter. Beyond what value of n is your formula no longer valid?
- 8.25 Using a 74x163 4-bit binary counter, design a modulo-11 counter circuit with the counting sequence 3, 4, 5, ..., 12, 13, 3, 4,
- 8.26 Look up the internal logic diagram for a 74x162 synchronous decade counter in a data book, and write its state table in the style of Table 8-11, including its counting behavior in the normally unused states 10–15.
- 8.27 Devise a cascading scheme for 74x163s, analogous to the synchronous parallel counter structure of Figure 8-29, such that the maximum counting speed is the same for any counter with up to 36 bits (nine '163s). Determine the maximum counting speed using worst-case delays from a manufacturer's data book for the '163s and any SSI components used for cascading.
- 8.28 Design a modulo-129 counter using two 74x163s and a single inverter.
- 8.29 Design a clocked synchronous circuit with four inputs, N3, N2, N1, and N0, that represent an integer N in the range 0–15. The circuit has a single output Z that is asserted for exactly N clock ticks during any 16-tick interval (assuming that N is held constant during the interval of observation). (*Hints:* Use combinational logic with a 74x163 set up as a free-running divide-by-16 counter. The ticks in which

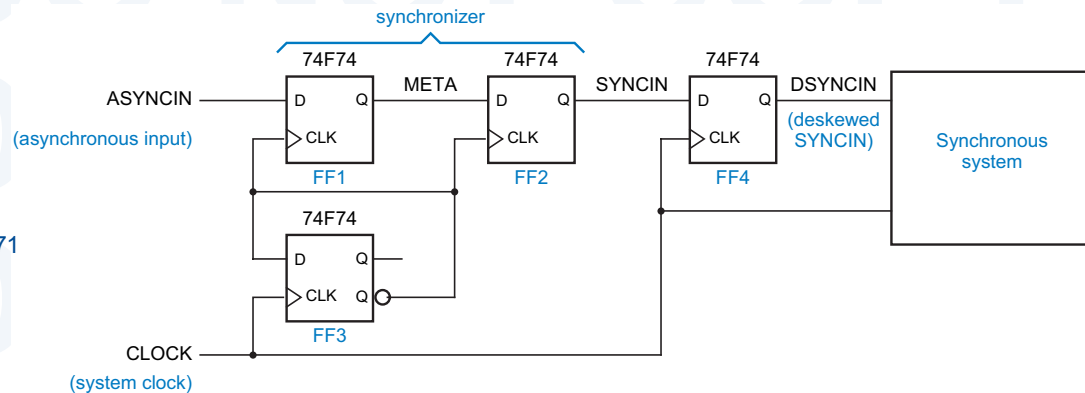
- Z is asserted should be spaced as evenly as possible, that is, every second tick when $N = 8$, every fourth when $N = 4$, and so on.)
- 8.30 Modify the circuit of Exercise 8.29 so that Z produces N transitions in each 16-tick interval. The resulting circuit is called a *binary rate multiplier*, and was once sold as a TTL MSI part, the 7497. (*Hint*: Gate the clock with the level output of the previous circuit.) *binary rate multiplier*
- 8.31 A digital designer (the author!) was asked at the last minute to add new functionality to a PCB that had room for just one more 16-pin IC. The PCB already had a 16-MHz clock signal, MCLK, and a spare microprocessor-controlled select signal, SEL. The designer was asked to provide a new clock signal, UCLK, whose frequency would be 8 MHz or 4 MHz depending on the value of SEL. To make things worse, the PCB had no spare SSI gates, and UCLK was required to have a 50% duty cycle at both frequencies. It took the designer about five minutes to come up with a circuit. Now it's your turn to do the same. (*Hint*: The designer had long considered the 74x163 to be the fundamental building block of tricky sequential-circuit design.)
- 8.32 Design a modulo-16 counter, using one 74x169 and at most one SSI package, with the following counting sequence: 7, 6, 5, 4, 3, 2, 1, 0, 8, 9, 10, 11, 12, 13, 14, 15, 7,
- 8.33 Modify the VHDL program in Table 8-14 so that the type of ports D and Q is STD_LOGIC_VECTOR, including conversion functions as required.
- 8.34 Modify the program in Table 8-16 to use structural VHDL, so it conforms exactly to the circuit in Figure 8-45, including the signal names shown in the figure. Define and use any of the following entities that don't already exist in your VHDL library: AND2, INV, NOR2, OR2, XNOR2, Vdffqpn.
- 8.35 Modify the program in Table 8-17 to use VHDL's generic statement, so that the counter size can be changed using the generic definition.
- 8.36 Design a parallel-to-serial conversion circuit with eight 2.048 Mbps, 32-channel serial links and a single 2.048 MHz, 8-bit, parallel data bus that carries 256 bytes per frame. Each serial link should have the frame format defined in Figure 8-55. Each serial data line SDATA_i should have its own sync signal SYNC_i; the sync pulses should be staggered so that SYNC_i + 1 has a pulse one clock tick after SYNC_i.
Show the timing of the parallel bus and the serial links, and write a table or formula that shows which parallel-bus timeslots are transmitted on which serial links and timeslots. Draw a logic diagram for the circuit using MSI parts from this chapter; you may abbreviate repeated elements (e.g., shift registers), showing only the unique connections to each one.
- 8.37 Repeat Exercise 8.36, assuming that all serial data lines must reference their data to a single, common SYNC signal. How many more chips does this design require?
- 8.38 Show how to enhance the serial-to-parallel circuit of Exercise 8-57 so that the byte received in each timeslot is stored in its own register for 125 μ s, until the next byte from that timeslot is received. Draw the counter and decoding logic for 32 timeslots in detail, as well as the parallel data registers and connections for

- timeslots 31, 0, and 1. Also draw a timing diagram in the style of Figure 8-58 that shows the decoding and data signals associated with timeslots 31, 0, and 1.
- 8.39 Suppose you are asked to design a serial computer, one that moves and processes data one bit at a time. One of the first decisions you must make is which bit to transmit and process first, the LSB or the MSB. Which would you choose, and why?
 - 8.40 Design an 8-bit self-correcting ring counter whose states are 11111110, 11111101, ..., 01111111, using only two SSI/MSI packages.
 - 8.41 Design two different 2-bit, 4-state counters, where each design uses just a single 74x74 package (two edge-triggered D flip-flops) and no other gates.
 - 8.42 Design a 4-bit Johnson counter and decoding for all eight states using just three SSI/MSI packages. Your counter need not be self-correcting.
 - 8.43 Starting with state 0001, write the sequence of states for a 4-bit LFSR counter designed according to Figure 8-68 and Table 8-21.
 - 8.44 Prove that an even number of shift-register outputs must be connected to the odd-parity circuit in an n -bit LFSR counter if it generates a maximum-length sequence. (Note that this is a necessary but not a sufficient requirement. Also, although Table 8-21 is consistent with what you're supposed to prove, simply quoting the table is not a proof!)
 - 8.45 Prove that X_0 must appear on the right-hand side of any LFSR feedback equation that generates a maximum-length sequence. (Note: Assume the LFSR bit ordering and shift direction are as given in the text; that is, the LFSR counter shifts right, toward the X_0 stage.)
 - 8.46 Suppose that an n -bit LFSR counter is designed according to Figure 8-68 and Table 8-21. Prove that if the odd-parity circuit is changed to an even-parity circuit, the resulting circuit is a counter that visits $2^n - 1$ states, including all of the states except 11...11.
 - 8.47 Find a feedback equation for a 3-bit LFSR counter, other than the one given in Table 8-21, that produces a maximum-length sequence.
 - 8.48 Given an n -bit LFSR counter that generates a maximum-length sequence ($2^n - 1$ states), prove that an extra XOR gate and an $n - 1$ input NOR gate connected as suggested in Figure 8-69 produces a counter with 2^n states.
 - 8.49 Prove that a sequence of 2^n states is still obtained if a NAND gate is substituted for a NOR above, but that the state sequence is different.
 - 8.50 Design an iterative circuit for checking the parity of a 16-bit data word with a single even parity bit. Does the order of bit transmission matter?
 - 8.51 Modify the shift-register program in Table 8-23 to provide an asynchronous clear input using a 22V10.
 - 8.52 Write an ABEL program that provides the same functionality as a 74x299 shift register. Show how to fit this function into a single 22V10 or explain why it does not fit.
 - 8.53 In what situations do the ABEL programs in Tables 8-26 and 8-27 give different operational results?

- 8.54 Modify the ABEL program in Table 8-26 so that the phases are always at least two clock ticks long, even if RESTART is asserted at the beginning of a phase. RESET should still take effect immediately.
- 8.55 Repeat the preceding exercise for the program in Table 8-27.
- 8.56 A student proposed to create the timing waveforms of Figure 8-72 by starting with the ABEL program in Table 8-27 and changing the encoding of each of states P1F, P2F, ..., P6F so that the corresponding phase output is 1 instead of 0, so that the phase output is 0 only during the second tick of each phase, as required. Is this a good approach? Comment on the results produced by the ABEL compiler produce when you try this.
- 8.57 The outputs waveforms produced by the ABEL programs in Tables 8-29 and 8-30 are not identical when the RESTART and RUN inputs are changed. Explain the reason for this, and then modify the program in Table 8-30 so that its behavior matches that of Table 8-29.
- 8.58 The ABEL ring-counter implementation in Table 8-26 is not self-synchronizing. For example, describe what happens if the outputs [P1_L..P6_L] are initially all 0, and the RUN input is asserted without ever asserting RESET or RESTART. What other starting states exhibit this kind of non-self-synchronizing behavior? Modify the program so that it *is* self-synchronizing.
- 8.59 Repeat the preceding exercise for the VHDL ring-counter implementation in Table 8-33.
- 8.60 Design an iterative circuit with one input B_i per stage and two boundary outputs X and Y such that $X = 1$ if at least two B_i inputs are 1 and $Y = 1$ if at least two *consecutive* B_i inputs are 1.
- 8.61 Design a combination-lock machine according to the state table of Table 7-14 on page 486 using a single 74x163 counter and combinational logic for the LD_L, CLR_L, and A–D inputs of the '163. Use counter values 0–7 for states A–H.
- 8.62 Write an ABEL program corresponding to the state diagram in Figure 8-84 for the multiplier control unit.
- 8.63 Write a VHDL program corresponding to the state diagram in Figure 8-84 for the multiplier control unit.
- 8.64 Write a VHDL program that performs with the same inputs, outputs, and functions as the multiplier data unit in Figure 8-82.
- 8.65 Write a VHDL program that combines the programs in the two preceding exercises to form a complete 8-bit shift-and-add multiplier.
- 8.66 The text stated that the designer need not worry about any timing problems in the synchronous design of Figure 8-83. Actually, there is one slight worry. Look at the timing specifications for the 74x377 and discuss.
- 8.67 Determine the minimum clock period for the shift-and-add multiplier circuit in Figure 8-83, assuming that the state machine is realized in a single GAL16V8-20 and that the MSI parts are all 74LS TTL. Use worst-case timing information from the tables in this book. For the '194, t_{pd} from clock to any output is 26 ns, and t_s is 20 ns for serial and parallel data inputs and 30 ns for mode-control inputs.

- 8.68 Design a data unit and a control-unit state machine for multiplying 8-bit two's-complement numbers using the algorithm discussed in Section 2.8.
- 8.69 Design a data unit and control-unit state machine for dividing 8-bit unsigned numbers using the shift-and-subtract algorithm discussed in Section 2.9.
- 8.70 Suppose that the SYNCIN signal in Drill 8.10 is connected to a combinational circuit in the synchronous system, which in turn drives the D inputs of 74ALS74 flip-flops that are clocked by CLOCK. What is the maximum allowable propagation delay of the combinational logic?
- 8.71 The circuit in Figure X8.71 includes a deskewing flip-flop so that the synchronized output from the multiple-cycle synchronizer is available as soon as possible after the edge of CLOCK. Ignoring metastability considerations, what is the maximum frequency of CLOCK? Assume that for a 74F74, $t_{\text{setup}} = 5 \text{ ns}$ and $t_{\text{pd}} = 7 \text{ ns}$.

Figure X8.71



- 8.72 Using the maximum clock frequency determined in Exercise 8.71, and assuming an asynchronous transition rate of 4 MHz, determine the synchronizer's MTBF.
- 8.73 Determine the MTBF of the synchronizer in Figure X8.71, assuming an asynchronous transition rate of 4 MHz and a clock frequency of 40 MHz, which is less than the maximum determined in Figure X8.71. In this situation, "synchronizer failure" really occurs only if DSYNCIN is metastable. In other words, SYNCIN may be allowed to be metastable for a short time, as long as it doesn't affect DSYNCIN. This yields a better MTBF.
- 8.74 A famous digital designer devised the circuit shown in Figure X8.74(a), which is supposed to eliminate metastability within one period of a system clock. Circuit M is a memoryless analog voltage detector whose output is 1 if Q is in the metastable state, 0 otherwise. The circuit designer's idea was that if line Q is detected to be in the metastable state when CLOCK goes low, the NAND gate will clear the D flip-flop, which in turn eliminates the metastable output, causing a 0 output from circuit M and thus negating the CLR input of the flip-flop. The circuits are all fast enough that this all happens well before CLOCK goes high again; the expected waveforms are shown in Figure X8.74(b).

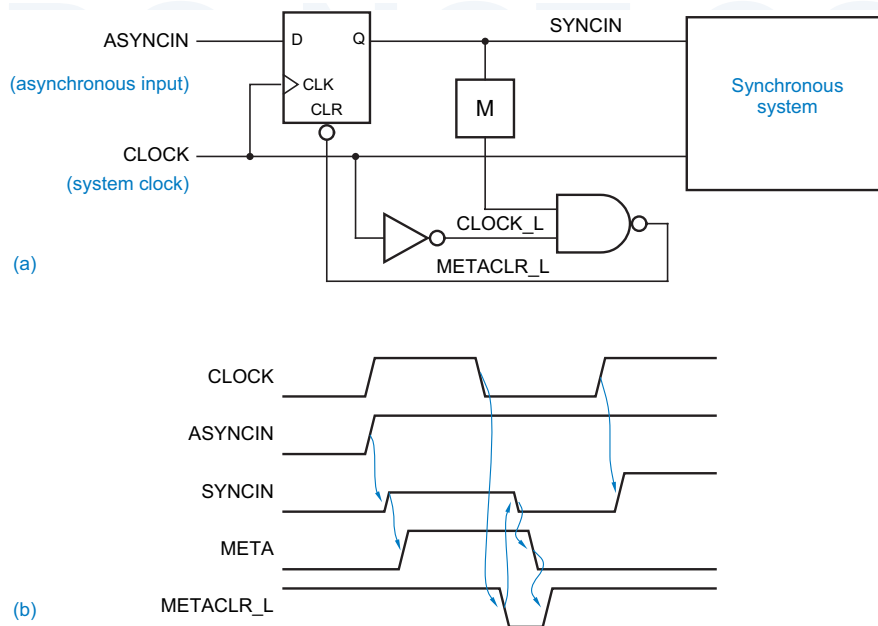


Figure X8.74

Unfortunately, the synchronizer still failed occasionally, and the famous digital designer is now designing pockets for blue jeans. Explain, in detail, how it failed, including a timing diagram.

- 8.75 Look up U.S. patent number 4,999,528, “Metastable-proof flip-flop,” and describe why it doesn’t always work as advertised. (*Hint*: There’s enough information in the abstract to figure out how it can fail.)
- 8.76 In the synchronization circuit of Figures 8-102, 8-104, and Figure 8-106, you can reduce the delay of the transfer of a byte from the RCLK domain to the SCLK domain if you use an earlier version of the SYNC pulse to start the synchronizer. Assuming that you can generate SYNC during any bit of the received byte, which bit should you use to minimize the delay? Also determine whether your solution satisfies the maximum-delay requirements for the circuit. Assume that all the components have 74AHCT timing and that the S-R latch is built from a pair of cross-coupled NOR gates, and show a detailed timing analysis for your answers.
- 8.77 Instead of using a latch in the synchronization control circuit of Figure 8-106, some applications use an edge-triggered D flip-flop as shown in Figure 8-111. Derive the maximum-delay and minimum-delay requirements for this circuit, corresponding to Eqns. 8-1 through 8-3, and discuss whether this approach eases or worsens the delay requirements.

