EE 121 March 15, 2001
Digital Design Laboratory Handout #36

Midterm Examination #2 Solutions

1. (12 points) *Multiplication.* The unsigned product of 2-bit numbers A_1A_0 and B_1B_0 is a 4-bit result $Y_3Y_2Y_1Y_0$. For example,

$$11_2 \cdot 11_2 = 3_{10} \cdot 3_{10} = 9_{10} = 1001_2 \,.$$

Find efficient Boolean formulas for the product bits $Y_3Y_2Y_1Y_0$ as functions of the bits of the multiplier A_1A_0 and the multiplicand B_1B_0. Hint: the formulas for Y_0 and Y_3 are *very* simple. Remark: the best equations are *not* the minimal sums of products.

There are several approaches to obtaining Boolean equations for multiplication. One method involves studying the usual method for binary multiplication. Another method is to determine the values of the output bits for all 16 input combinations, then use Karnaugh maps to arrive at minimal sums of products.

$$\begin{array}{cccc}
 & & B_1 & B_0 \\
 & \times & A_1 & A_0 \\
\cline{3-4}
 & & A_0B_1 & A_0B_0 \\
+ & A_1B_1 & A_1B_0 & \\
\hline
Y_3 & Y_2 & Y_1 & Y_0
\end{array}$$

	B_1B_0			
A_1A_0	00	01	11	10
00	0000	0000	0000	0000
01	0000	0001	0011	0010
11	0000	0011	1001	0110
10	0000	0010	0110	0100

From the multiplication tableau, we see that each product bit Y_i can be expressed as the sum (XOR) of certain ANDs of bits of the factors together with the carry from the previous column:

$$\begin{aligned}
Y_0 &= A_0 \cdot B_0 \\
Y_1 &= A_0 \cdot B_1 \oplus A_1 \cdot B_0 \\
Y_2 &= A_1 \cdot B_1 \oplus \text{carry} = A_1 \cdot B_1 \oplus (A_1 \cdot A_0 \cdot B_1 \cdot B_0) = A_1 \cdot B_1 \cdot (A_0 \cdot B_0)' \\
Y_3 &= A_1 \cdot A_0 \cdot B_1 \cdot B_0
\end{aligned}$$

We can derive four Karnaugh maps from the above table, resulting in the following minimal sums of products for Y_i:

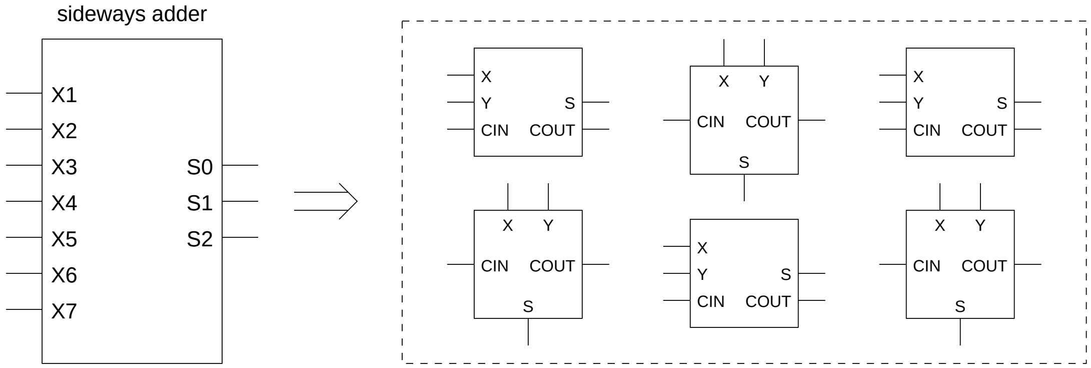
$$\begin{aligned}
Y_0 &= A_0 \cdot B_0 \\
Y_1 &= A_1' \cdot A_0 \cdot B_1 + A_1 \cdot A_0' \cdot B_0 + A_1 \cdot B_1' \cdot B_0 + A_0 \cdot B_1 \cdot B_0' \\
Y_2 &= A_1 \cdot A_0' \cdot B_1 + A_1 \cdot B_1 \cdot B_0' \\
Y_3 &= A_1 \cdot A_0 \cdot B_1 \cdot B_0
\end{aligned}$$

Efficient circuits for calculating all four bits of the product re-use common subexpressions, such as $A_0 \cdot B_0$ and $A_1 \cdot B_1$.

2. (13 points) *Sideways adder.* The *sideways sum* of a signal vector $(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n)$ is the number of input signals that are true; that is,

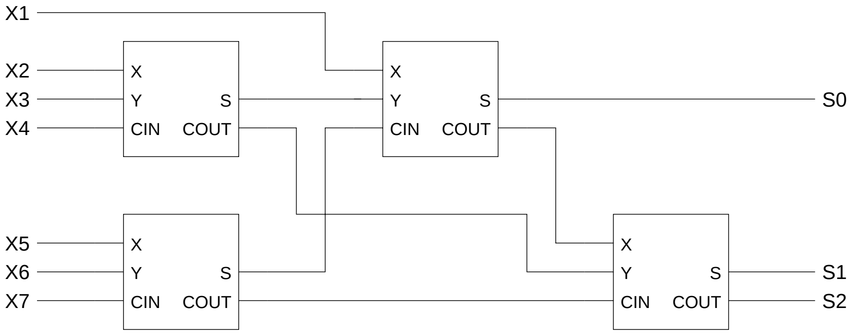
$$\mathsf{S} = \sum_{k=1}^{n} \mathsf{X}_k .$$

Since S has $n+1$ possible values, the sum requires a vector of $m = \lceil \log_2(n+1) \rceil$ bits. For $n = 7$, the sum is represented by $\lceil \log_2(7+1) \rceil = \lceil \log_2 8 \rceil = 3$ bits; that is, $\mathsf{S} = (\mathsf{S}_2, \mathsf{S}_1, \mathsf{S}_0)$.

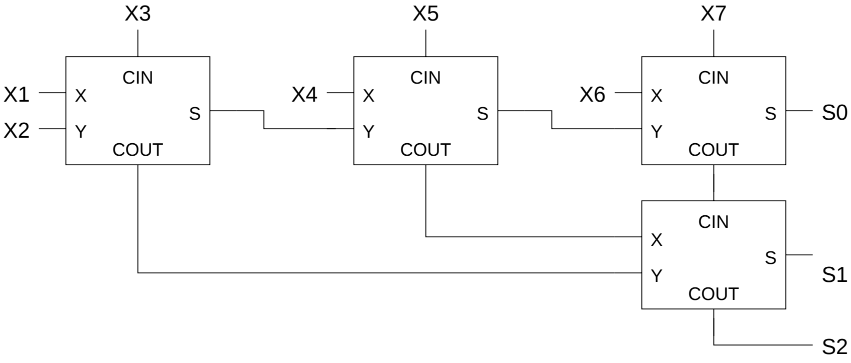


Using only full adders, design a circuit that calculates the sideways sum $(\mathsf{S}_2, \mathsf{S}_1, \mathsf{S}_0)$ of $(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_7)$. You may use either of the two alternative logic symbol for the full adder. (Do not try to wire the full adders inside the box above.) Hint: six full adders are more than enough. Bonus points for minimum number of gates and smallest propagation delay.

Two sideways adder circuits using four full adders are shown below.

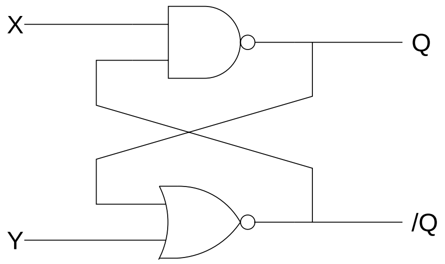


The second circuit is easier to understand and generalize but has larger propagation delay.



3. (20 points) *Latches?*

a. A committee of logic designers could not agree on whether to use NAND gates or NOR gates to build a set-reset latch. The compromise design is shown below.



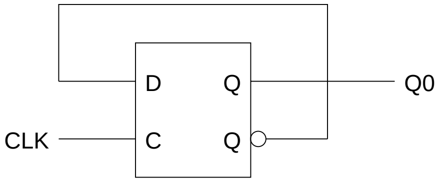
Fill in the function table for this sequential circuit.

X	Y	Q	/Q
0	0	H	L
0	1	H	L
1	0	last Q	last /Q
1	1	H	L

Explain what is wrong with this "latch"?

When X is 0, the output of the NAND gate is high, which forces the output of the NOR gate low. When Y is 1, the output of the NOR gate is low, which forces the output of the NAND gate high. When X is 1 and Y is 0, the circuit reduces to two inverters and retains its previous state. This circuit is a set latch; it can never be reset.

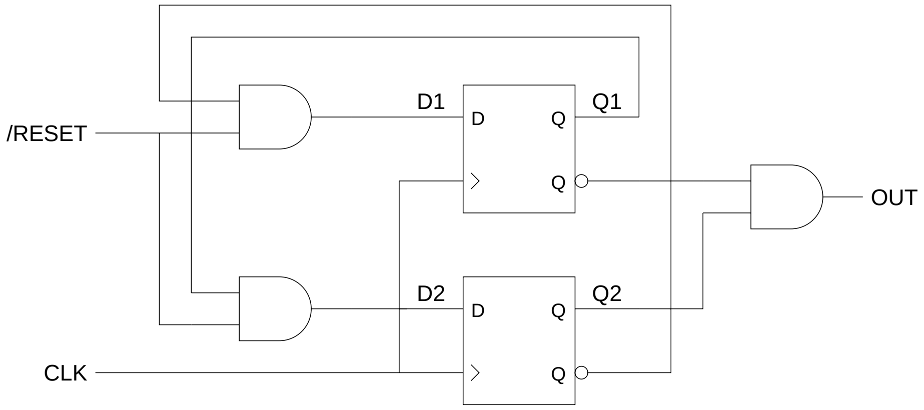
b. The feedback sequential circuit shown below is intended to change state with each clock pulse; in other words, it is supposed to be a "toggle latch."



Explain why this sequential circuit does not work as intended. What actually happens?

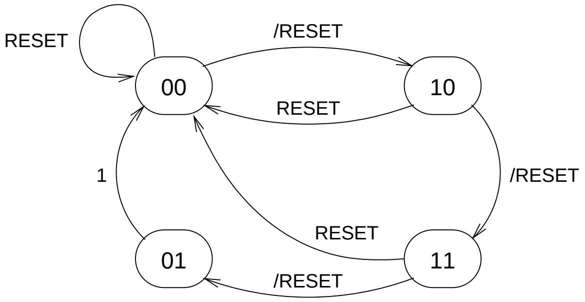
While CLK is high, the latch is transparent, and the circuit behaves like an inverter whose output is connected to its input—the output oscillates or settles at an intermediate voltage. The value latched when CLK goes low depends on how many times the output Q0 has changed while the clock was high.

4. (20 points) *State machine analysis.* Analyze the clocked synchronous state machine below.



a. Draw a state diagram for this state machine.

The state diagram is most easily obtained from the transition/output table below.



This state machine is a 2-bit Johnson (twisted-ring) counter with an active-low reset input.

b. Is this machine a Mealy machine or a Moore machine?

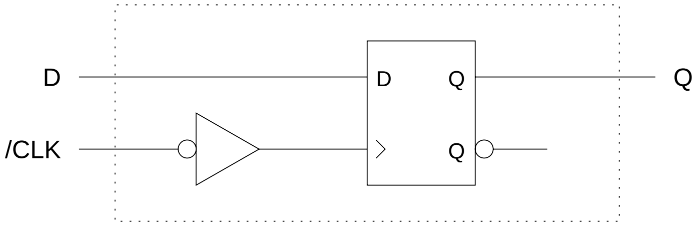
Answer (circle one): Mealy [✓ Moore]

The output $\text{OUT} = \text{Q1}' \cdot \text{Q2}$ is determined by the current state, independent of the input.

c. Fill in the transition/output table for the above state machine.

	/RESET		
Q1 Q2	0	1	OUT
0 0	0 0	1 0	0
0 1	0 0	0 0	1
1 0	0 0	1 1	0
1 1	0 0	0 1	0
	Q1*Q2*		

5. (10 points) A negative edge-triggered D flip-flop can be built from a positive edge-triggered D flip-flop and an inverter, as shown below.



The propagation delays and timing parameters for the inverter and the D flip-flop are given in the following tables.

Inverter		
	min	max
t_{pLH}	5	7
t_{pHL}	3	5

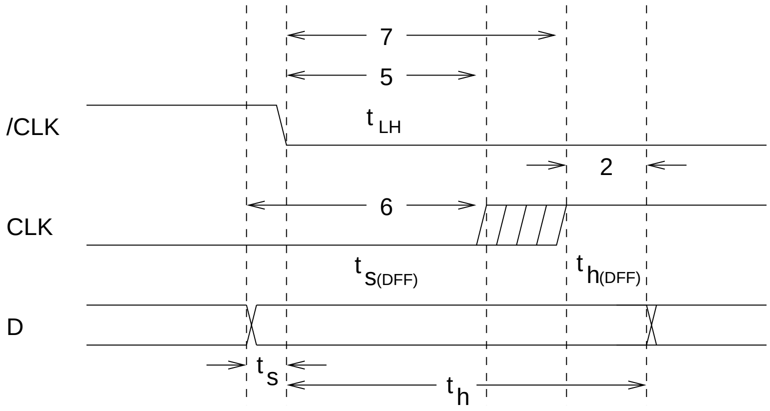
Flip-Flop	
t_{s}	6
t_{h}	2

Find the setup time t_{s} and the hold time t_{h} for this negative edge-triggered flip-flop.

Setup time: $t_{\text{s}} \ = \ t_{\text{s(DFF)}} \ - \ t_{\text{LH(min)}} \ = \ 6 - 5 \ = \ 1$ ns

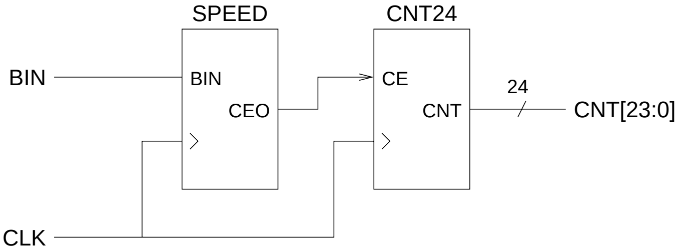
Hold time: $t_{\text{h}} \ = \ t_{\text{h(DFF)}} \ + \ t_{\text{LH(max)}} \ = \ 2 + 7 \ = \ 9$ ns

The following timing diagram shows how to find the setup and hold times of the derived flip-flop from the timing parameters of the original flip-flop.



Because the internal clock signal is a delayed version of the external clock, the hold time is increased by the maximum possible delay of the inverter that produces the internal clock. The worst case setup time is reduced by the minimum possible delay of the inverter. For large enough gate delay, it is possible for the setup time to be negative. If the data input to this logic block were delayed, the hold time would be reduced, perhaps even made negative.

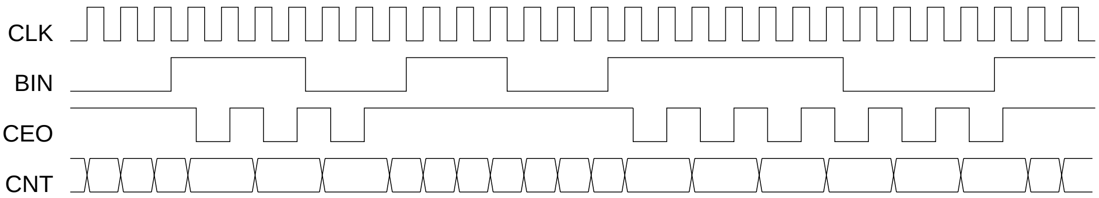
6. (25 points) *Variable-speed counter.* The 24-bit binary counter shown below counts either at full speed or at half speed as determined by the speed-control state machine SPEED.



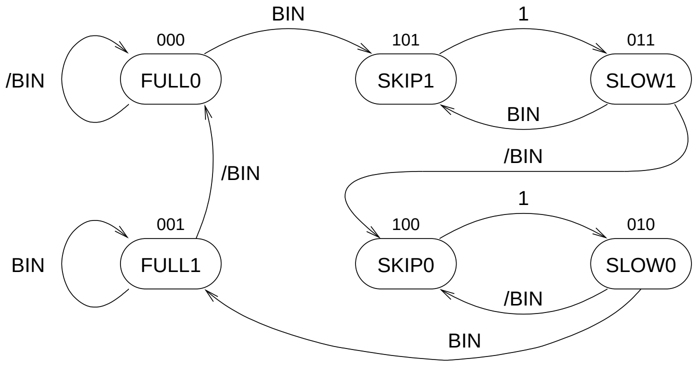
In half-speed mode, SPEED enables counting on every other clock, whereas in full-speed mode, the counter enable CE is always true. The speed is changed by the pushbutton input BIN; each positive pulse of BIN toggles the speed mode, from half to full or from full to half. This input is debounced, and its pulse width is a large number of clock cycles.

a. Draw the state diagram for the SPEED state machine.

The following timing diagram illustrates the desired behavior of the variable-speed counter.



A state diagram for the variable-speed counter is shown below. The initial state is FULL0, which corresponds to counting at full speed while the input signal BIN inactive.



Three bits are needed to represent the state. One state variable can determine CEO, which will simplify the output logic. Another bit of the state is the last observed value of the input BIN. State encodings are shown above the state boxes in the above state diagram.

In the above diagram, BIN is not examined in states SKIP0 or SKIP1, since by assumption BIN will remain high or low for a large number of clock periods. Another state diagram might provide transitions from SKIP1 to SLOW0 when BIN is false and from SKIP0 to FULL1 when BIN is true.

b. For a state assignment of your choice, write the transition/output table.

There are a vast number of state assignments and several reasonable state assignments. We choose a state assignment in which CEO is the complement of state variable Q2 and in which Q0 usually contains the last value of BIN.

		BIN		
State	Q2 Q1 Q0	0	1	CEO
FULL0	0 0 0	0 0 0	1 0 1	1
FULL1	0 0 1	0 0 0	0 0 1	1
SLOW0	0 1 0	1 0 0	0 0 1	1
SLOW1	0 1 1	1 0 0	1 0 1	1
SKIP0	1 0 0	0 1 0	0 1 0	0
SKIP1	1 0 1	0 1 1	0 1 1	0
	1 1 0	0 0 0	0 0 0	0
	1 1 1	0 0 0	0 0 0	0
		Q2* Q1* Q0*		

The last two rows of the transition/output table could be filled in arbitrarily, since they correspond to states that should not occur. To be safe, we have specified zero values for the next state and output.

c. For the state assignment of part b, write the transition and output equations.

The sum-of-products representations of the next state equations are easily obtained from the transition table. Karnaugh maps can be used to find minimal sums of products. The single output equation is very simple, by choice of state assignment.

$$
\begin{aligned}
\mathsf{Q2}\ast &= \mathsf{BIN}\cdot\mathsf{Q2}'\cdot\mathsf{Q1}'\cdot\mathsf{Q0}' \;+\; \mathsf{BIN}'\cdot\mathsf{Q2}'\cdot\mathsf{Q1} \;+\; \mathsf{Q2}'\ast\mathsf{Q1}\cdot\mathsf{Q0} \\
& \quad [\,\mathsf{BIN}\cdot\mathsf{Q2}'\cdot\mathsf{Q1}'\cdot\mathsf{Q0}' \;+\; \mathsf{BIN}'\cdot\mathsf{Q1} \;+\; \mathsf{Q1}\cdot\mathsf{Q0}\,] \\
\mathsf{Q1}\ast &= \mathsf{Q2}\cdot\mathsf{Q1}' \\
& \quad [\,\mathsf{Q2}\,] \\
\mathsf{Q0}\ast &= \mathsf{BIN}\cdot\mathsf{Q2}' \;+\; \mathsf{Q2}\cdot\mathsf{Q1}'\cdot\mathsf{Q0} \\
& \quad [\,\mathsf{BIN}\cdot\mathsf{Q2}' \;+\; \mathsf{Q2}\cdot\mathsf{Q0}\,] \\
\mathsf{CEO} &= \mathsf{Q2}'
\end{aligned}
$$

The formulas in brackets are the slightly simplified transition equations obtained by using don't cares in the last two rows of the transition table.