
The University of Michigan
Department of EECS
EECS 270: Introduction to Logic Design
Professors Sakallah & Brehob

Final Exam Solution
Wednesday December 16, 2009
10:30 a.m.-12:30 p.m.
A thru L: DOW 1013
M thru Q: DOW 1010
R thru T: DOW 1017
V thru Z: DOW 1005

Open Book and Notes

Name: _____

UMID: _____

Instructions:

- No calculators, cell phones, or other electronics may be used.
- Print your name, write your UMID, and sign the honor pledge on this exam.
- The exam consists of 11 problems with the point distribution as indicated in the heading of each problem.
- There are limited opportunities for partial credit. If you find yourself spending too much time on any one question, move on to another question. Use your time wisely.

| | | | |
|---|---|---|---|
| 1. | /20 | 7. | /14 |
| 2. | /7 | 8. | /11 |
| 3. | /7 | 9. | /9 |
| 4. | /4 | 10. | /10 |
| 5. | /4 | 11. | /10 |
| 6. | /4 | Tot: | /100 |

## Honor Pledge:

"I have neither given nor received aid on this exam, nor have I concealed any violations of the Honor Code."

Signature: _____

**1. [Miscellaneous—20 Points]**

a. [1 Point] The minimal SOP and minimal POS forms of a Boolean function have the same number of terms and literals. F TRUE FALSE

b. [2 Points] The product term $p = ACD'E$ is an implicant of $f = A'B'E' + BD'E + ACE$. T TRUE FALSE

c. [2 Points] The function $f = A'B'E' + BD'E + ACE$ covers $p = ADE$. F TRUE FALSE

d. [2 Points] If 8-bit numbers are used, the two's complement integer representation covers the range from $-2^7 = -128$ (most negative) to 127 (most positive).

e. [2 Points] An 8 Mbit RAM with a word size of 8 bits requires 20 address bits. An 8 Mbit RAM with a word size of 1 bit requires 23 address bits.

f. [6 Points] For each of the four functions listed in the following table, indicate the number of minterms, prime implicants, and essential prime implicants.

| | ***n*-input OR** | ***n*-input AND** | ***n*-input XOR** | **2-to-1 MUX** |
|---|---|---|---|---|
| **# minterms** | $2^n - 1$ | 1 | $2^{n-1}$ | 4 |
| **# prime implicants** | $n$ | 1 | $2^{n-1}$ | 3 |
| **# essential prime implicants** | $n$ | 1 | $2^{n-1}$ | 2 |

g. [2 Points] **Modified from final exam F96**A threshold gate with inputs $x_1$, $x_2$ and output $z$ is specified by two real weights $w_1$, $w_2$ and a real threshold $t$. The gate output is equal to logic 1 when the weighted sum of the inputs equals or exceeds the threshold. In other words, $z = 1$ if and only if $w_1 \times x_1 + w_2 \times x_2 \geq t$ where $\times$ and $+$ denote arithmetic multiplication and addition, respectively. Which standard logic gate (AND, OR, NAND, NOR, XOR, XNOR) corresponds to the threshold gate with $w_1 = w_2 = -2$ and $t = -1$?
Answer: NOR

h. [3 Points] Let $G_{j,i}$ and $P_{j,i}$ denote, respectively, the group generate and group propagate signals for the bit range $i$ through $j$ (inclusive). Also, let $c_i$ denote the carry into bit position $i$. Indicate the truth or falsehood of the following statements:

$G_{11,3} = G_{11,6} + P_{11,6}G_{5,3}$ T TRUE FALSE

$G_{9,1} = G_{9,3} + P_{9,2}G_{2,1}$ F TRUE FALSE

$c_5 = G_{4,2} + P_{4,2}c_2$ T TRUE FALSE

**2. [Two's Complement Numbers—7 Points]**

Given the 4-bit two's complement number $X = x_3x_2x_1x_0$ write the simplest SOP expression for each of the following:

a. *odd* (should be 1 when $X$ is odd and 0 otherwise):

*odd* = $x_0$

b. *even* (1 when $X$ is even and 0 otherwise):

*even* = $\overline{x_0}$

c. *positive* (1 when $X$ is positive and 0 otherwise):

*positive* = $\overline{x_3}x_2 + \overline{x_3}x_1 + \overline{x_3}x_0$

d. *negative* (1 when $X$ is negative and 0 otherwise):

*negative* = $\overline{x_3}$

e. *zero* (1 when $X$ is zero and 0 otherwise):

*zero* = $\overline{x_3}\,\overline{x_2}\,\overline{x_1}\,\overline{x_0}$

f. *maximum* (1 when $X$ has its largest possible value and 0 otherwise):

*maximum* = $\overline{x_3}x_2x_1x_0$

g. *minimum* (1 when $X$ has its smallest possible value and 0 otherwise):

*minimum* = $x_3\overline{x_2}\,\overline{x_1}\,\overline{x_0}$

**3. [K-Map Minimization—7 Points]**

Derive a minimal SOP *and* a minimal POS expression for the function $f(a, b, c, d) = \sum_{(a, b, c, d)}(1, 2, 4, 5, 6, 13) + d(7, 9, 10, 11, 12)$. Mark your solution on the K-Maps below.

| *ab* \ *cd* | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 1 |
| 01 | 1 | 1 | d | 1 |
| 11 | d | 1 | 0 | 0 |
| 10 | 0 | d | d | d |



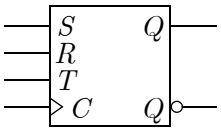
| *ab* \ *cd* | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 0 | 1 |
| 01 | 1 | 1 | d | 1 |
| 11 | d | 1 | 0 | 0 |
| 10 | 0 | d | d | d |



Minimal SOP expression: _____ Minimal POS expression: _____

**4. [Flip-Flop Characteristic Equation—4 Points]**

A new Set-Reset-Toggle (SRT) flip-flop has just been announced by flipflops.com, a start-up company that specializes in internet delivery of storage elements to the electronics and computer industries. The device symbol and function table for this flip-flop are shown below:




| $S$ | $R$ | $T$ | **Function** |
|---|---|---|---|
| X | X | 1 | Toggle (i.e., complement stored value) |
| 1 | 0 | 0 | Set |
| 0 | 1 | 0 | Reset |
| 0 | 0 | 0 | Hold |
| 1 | 1 | 0 | Hold |

Which of the following is its characteristic equation?

a. $Q^+ = TQ' + T'[Q(S \oplus R) + SR']$

b. $Q^+ = (T \oplus Q) + S + R'Q$

c. $Q^+ = T'Q + T[Q'(S + R') + SR']$

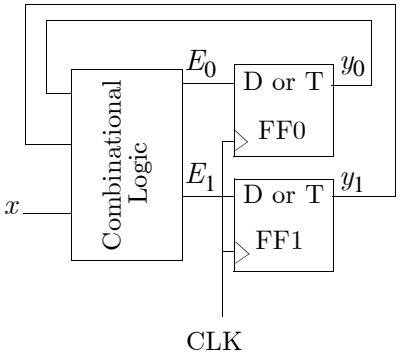
* d. $Q^+ = TQ' + T'[Q(S + R') + SR']$

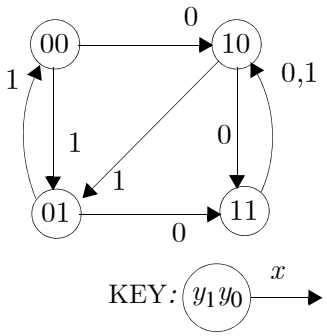
e. $Q^+ = T + [Q(S + R') + SR']$

Write the next-state equation $Q^+ = TQ' + SR'T' + (S'R' + SR)T'Q$ by inspection from the function table and simplify.

**5. [FSM Design—4 Points] Modified from final exam F96**

A four-state FSM design has been handed down to a new digital design engineer with a partial schematic. The desired state diagram of the FSM is known and the excitation functions of the two flip-flops are known, but the type of flip-flops to be used was not recorded. At least it is known that they must be T or D flip-flops because they only have one excitation function each. The excitation functions are: $E_1 = \bar{x} + y_1 y_0$ and $E_0 = x + y_1$, where $x$ is the primary input and $\{y_1, y_0\}$ are the state variables. Determine which flip-flops or combination of flip-flops can be used to implement the specified state diagram.




a. FF0: T, FF1: T

b. FF0: D, FF1: D

* c. FF0: T, FF1: D

d. FF0: D, FF1: T

e. None of the above; the excitation logic is erroneous

From the state diagram we get $y_1^+ = \bar{x} + y_1 y_0$ and $y_0^+ = \overline{y_1} y_0 \bar{x} + y_1 \overline{y_0} + \overline{y_0} x$. The equation for $y_1^+$ matches $E_1$ so FF1 is a D FF. We now check if FF0 is a T FF: $y_0^+ = E_0 \oplus y_0 = (x + y_1) \oplus y_0 = \bar{x}\overline{y_1} y_0 + (x + y_1)\overline{y_0}$ which matches $y_0^+$ from the state diagram, so FF) is a T FF.

**6. [Shift Register—4 Points]**

Let $X = x_3x_2x_1x_0$ denote the contents of register $R$, shown in this circuit, as a two's complement number. After exactly two clock ticks, the contents of $R$ will be:

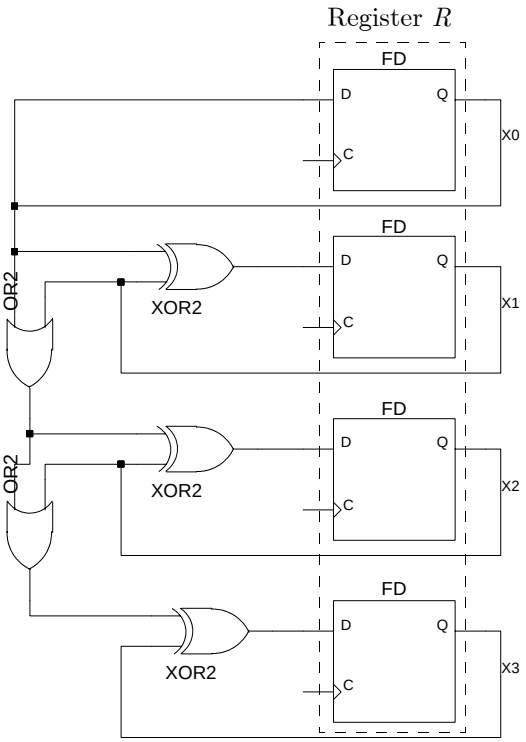
A. $X+1$

B. $X+2$

* C. $X$

D. $-X$

E. $X-1$




$$x_0^+ = x_0$$
$$x_1^+ = x_0 \oplus x_1$$
$$x_2^+ = (x_0 + x_1) \oplus x_2$$
$$x_3^+ = (x_0 + x_1 + x_2) \oplus x_3$$

$$x_0^{++} = x_0^+ = x_0$$
$$x_1^{++} = x_0^+ \oplus x_1^+ = x_0 \oplus x_0 \oplus x_1 = 0 \oplus x_1 = x_1$$
$$x_2^{++} = (x_0^+ + x_1^+) \oplus x_2^+ = (x_0 + (x_0 \oplus x_1)) \oplus (x_0 + x_1) \oplus x_2$$
$$= x_0 \oplus (x_0 \oplus x_1) \oplus x_0(x_0 \oplus x_1) \oplus x_0 \oplus x_1 \oplus x_0x_1 \oplus x_2$$
$$= x_0 \oplus x_0 \oplus x_1 \oplus x_0 \oplus x_0x_1 \oplus x_0 \oplus x_1 \oplus x_0x_1 \oplus x_2$$
$$= x_2$$
$$x_3^{++} = (x_0^+ + x_1^+ + x_2^+) \oplus x_3^+ = \ldots = x_3$$

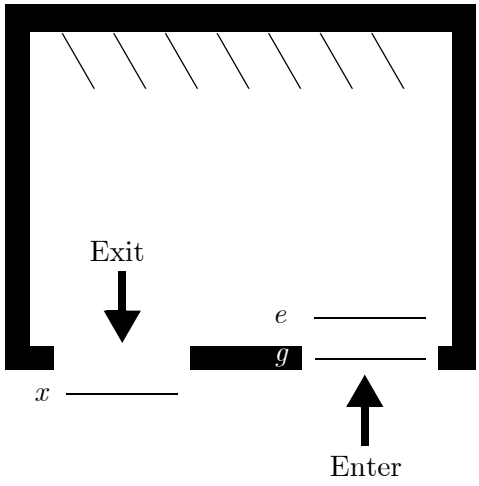
**7. [Sequential Circuit Design—14 Points]**

A controller for a parking garage (whose schematic is shown at right) has two inputs $e$ and $x$ and one Moore output $g$ defined as follows:

$$e = \begin{cases} 1, & \text{if a car has just entered the garage} \\ 0, & \text{otherwise} \end{cases}$$

$$x = \begin{cases} 1, & \text{if a car has just exited the garage} \\ 0, & \text{otherwise} \end{cases}$$

$$g = \begin{cases} 0, & \text{opens the entrance gate} \\ 1, & \text{closes the entrance gate} \end{cases}$$




There is no gate at the exit (parking is free!). The entrance gate should remain open as long as the garage has available parking spaces and should be closed when the garage is full. You should make reasonable assumptions about traffic flow. For example, you may assume that the entrance gate has enough time to close between cars when the garage becomes full (i.e., it is not possible for two cars to pass under the gate while it is closing.)

Assuming that you have access to a 10-bit up-down counter whose count state is denoted by the variable COUNT (ranging from 0 to 1023), and that the garage has 600 spaces, complete the following transition list to capture the required controller behavior. Minimize the transition conditions to receive full credit. *Hint*: You don't need to explicitly enumerate all possible states to capture the required behavior; your job will be much easier if you think in terms of sets of states!

| Present State | Transition Condition | Next State | g |
|---|---|---|---|
| COUNT = 0 | $e$ | COUNT = 1 | 0 |
| COUNT = 0 | $e'$ | COUNT = 0 | 0 |
| 1 ≤ COUNT ≤ 599 | $e'x' + ex$ | COUNT | 0 |
| 1 ≤ COUNT ≤ 599 | $ex'$ | COUNT + 1 | 0 |
| 1 ≤ COUNT ≤ 599 | $e'x$ | COUNT - 1 | 0 |
| COUNT = 600 | $x$ | COUNT = 599 | 1 |
| COUNT = 600 | $x'$ | COUNT = 600 | 1 |
| | | | |
| | | | |
| | | | |

**8. [Tabular Two-Level Minimization—11 Points]**

a. [7 Points] Complete the partially filled-in Quine-McCluskey table below to determine the prime implicants of the 4-variable function $f(a, b, c, d) = \Sigma_{a,b,c,d}(0, 1, 5, 8) + d(2, 9, 13, 14, 15)$ . Place ✓ next to product terms that are subsumed by larger product terms.

| | $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|---|
| ✓ | 0 | 0 | 0 | 0 |
| ✓ | 0 | 0 | 0 | 1 |
| ✓ | 0 | 0 | 1 | 0 |
| ✓ | 1 | 0 | 0 | 0 |
| ✓ | 0 | 1 | 0 | 1 |
| ✓ | 1 | 0 | 0 | 1 |
| ✓ | 1 | 1 | 0 | 1 |
| ✓ | 1 | 1 | 1 | 0 |
| ✓ | 1 | 1 | 1 | 1 |

| | $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|---|
| ✓ | 0 | 0 | 0 | - |
| | 0 | 0 | - | 0 |
| ✓ | - | 0 | 0 | 0 |
| ✓ | 0 | - | 0 | 1 |
| ✓ | - | 0 | 0 | 1 |
| ✓ | 1 | 0 | 0 | - |
| ✓ | - | 1 | 0 | 1 |
| ✓ | 1 | - | 0 | 1 |
| | 1 | 1 | - | 1 |
| | 1 | 1 | 1 | - |

| $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|
| - | 0 | 0 | - |
| - | - | 0 | 1 |

b. [3 Points] For each prime implicant identified in the above table, indicate the on-set minterms that it covers by placing "1" in the appropriate cell of the chart below:

On-set Minterms

| Prime Implicants | 0 | 1 | 5 | 8 |
|---|---|---|---|---|
| $a'b'd'$ | 1 | | | |
| $abd$ | | | | |
| $abc$ | | | | |
| $b'c'$ | 1 | 1 | | 1 |
| $c'd$ | | 1 | 1 | |
| | | | | |

c. [1 Point] The minimal SOP expression for $f$ is: _____ $b'c' + c'd$

**9. [State Minimization—9 Points]**

a. [6 Points] Complete the merger table below to find the equivalent states, if any, of the following state table. In each cell of the merger table:

- Write ✓ to indicate *unconditional equivalence* of the corresponding state pair
- Write ✗ to indicate *unconditional non-equivalence* of the corresponding state pair
- Write $XY$ to indicate that the equivalence/non-equivalence of the corresponding state pair depends on the equivalence/non-equivalence of state pair $XY$.

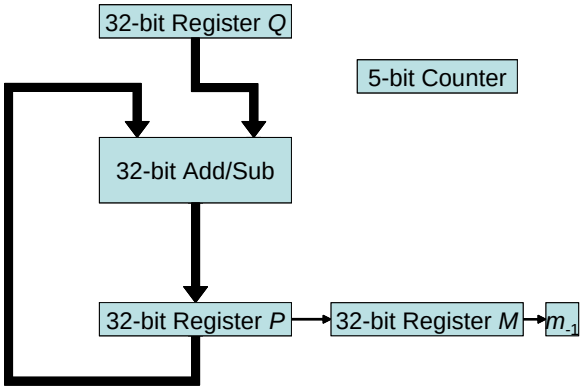
| **Present State** | **Next State, $z$** | |
|---|---|---|
| | $x = 0$ | $x = 1$ |
| $A$ | $A$, 1 | $E$, 0 |
| $B$ | $C$, 0 | $A$, 1 |
| $C$ | $B$, 0 | $C$, 1 |
| $D$ | $B$, 0 | $C$, 1 |
| $E$ | $D$, 0 | $F$, 1 |
| $F$ | $A$, 1 | $B$, 0 |

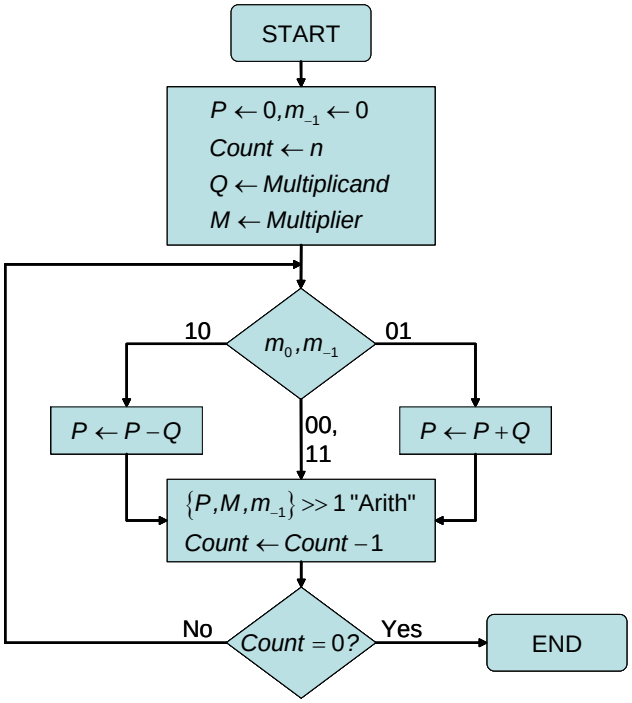
| | $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|---|
| $B$ | ✗ | | | | |
| $C$ | ✗ | AC | | | |
| $D$ | ✗ | BC, AC | ✓ | | |
| $E$ | ✗ | CD, AF | BD, CF | BD, CF | |
| $F$ | BE | ✗ | ✗ | ✗ | ✗ |

b. [3 Points] Equivalent states: _____

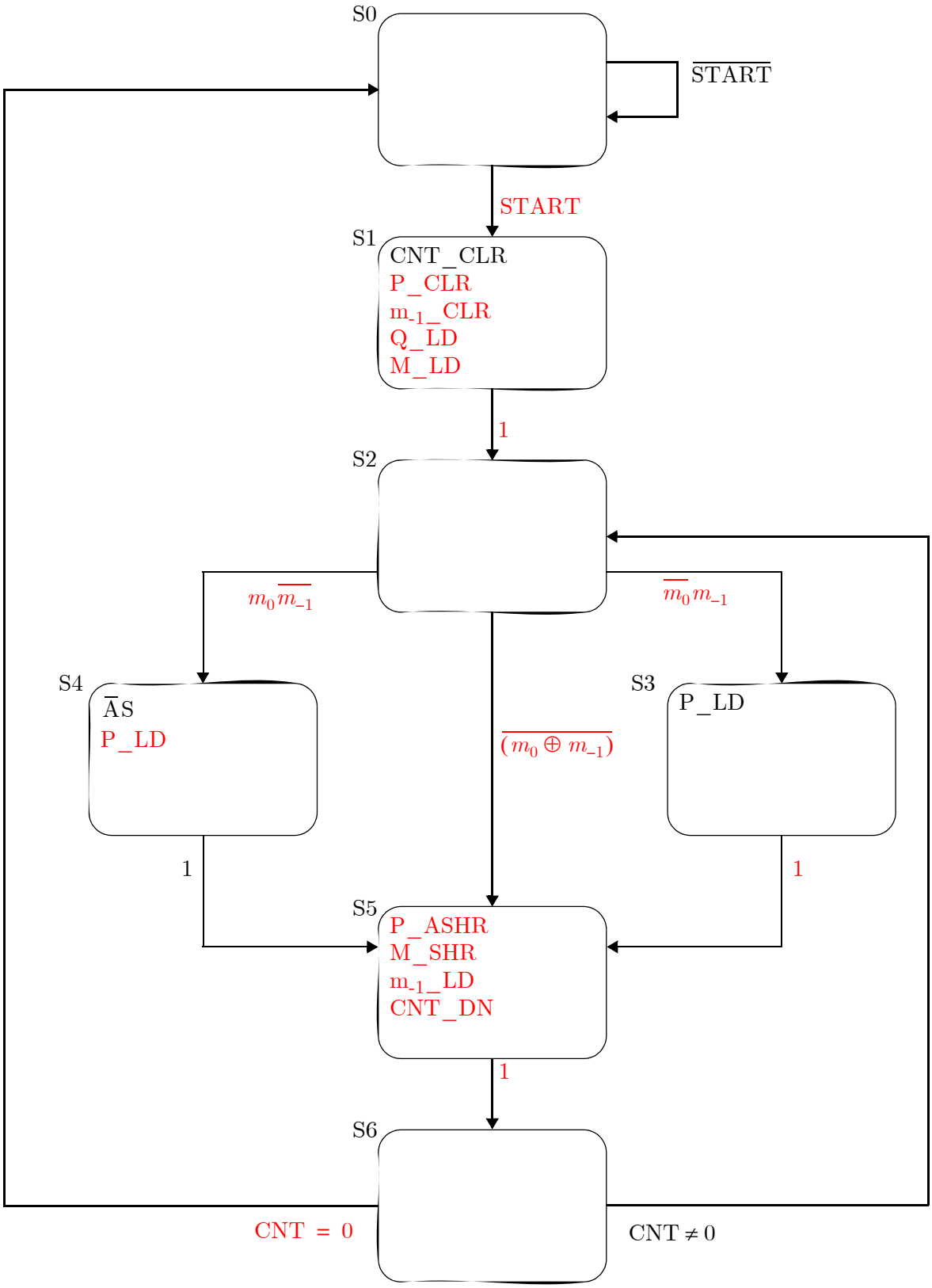
**10. [FSM Controller for Booth Multiplier—10 Points]**

The datapath and algorithm for the Booth multiplier we discussed in class are reproduced below. The capabilities of each datapath component are also listed. For example, the 32-bit accumulator P can be cleared (P_CLR = 1), loaded (P_LD = 1) and arithmetically shifted right (P_ASHR = 1). These are "control" signals that must be set to 1 or 0 at the appropriate times by the finite-state controller that implements the Booth algorithm. A partially filled-in state diagram for such a controller is shown on the next page. Complete this diagram by appropriately labeling the conditions on the state transitions and by listing the control signals that must be set to 1 in each state (you don't have to write the signals that must be set to 0).



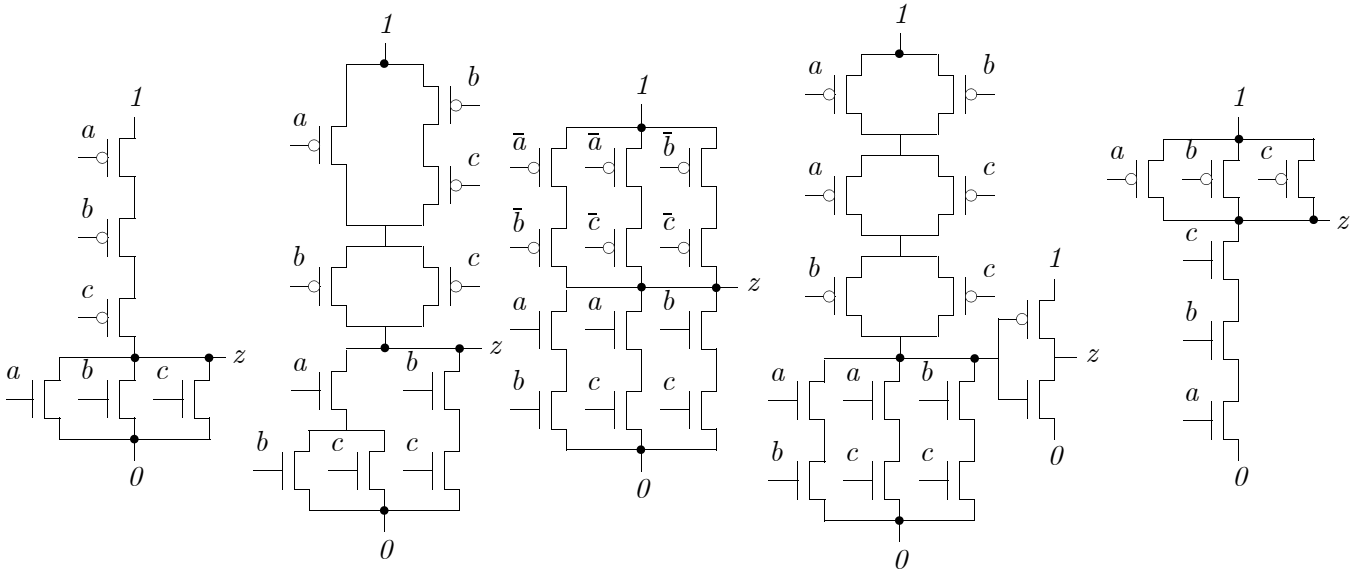




| Datapath Component | Description | Control Outputs | Function |
|---|---|---|---|
| Add/Sub Unit | 32-bit adder/subtractor | $\overline{A}S$ | $\overline{A}S$ = 0: add<br>$\overline{A}S$ = 1: subtract |
| CNT | 5-bit down counter | CNT_CLR<br>CNT_DN | CNT_CLR = 1: Count ← 0<br>CNT_DN = 1: Count ← Count - 1 |
| Q | 32-bit multiplicand register | Q_LD | Q_LD = 1: Q ← Multiplicand |
| M | 32-bit multiplier register | M_LD<br>M_SHR | M_LD = 1: M ← Multiplier<br>M_SHR = 1: M ← M >> 1 (logical) |
| P | 32-bit product accumulator | P_CLR<br>P_LD<br>P_ASHR | P_CLR = 1: P ← 0<br>P_LD = 1: P ← Add/Sub Unit<br>P_ASHR = 1: P ← P >> 1 (arithmetic) |
| $m_{-1}$ | 1-bit flip-flop | $m_{-1}$_CLR<br>$m_{-1}$_LD | $m_{-1}$_CLR = 1: $m_{-1} \leftarrow 0$<br>$m_{-1}$_LD = 1 : $m_{-1} \leftarrow m_0$ |


S0
START
START
S1
CNT_CLR
P_CLR
m-1_CLR
Q_LD
M_LD
1
S2
$m_0\overline{m_{-1}}$
$\overline{m_0}m_{-1}$
$\overline{(m_0 \oplus m_{-1})}$
S4
AS
P_LD
S3
P_LD
1
1
S5
P_ASHR
M_SHR
m-1_LD
CNT_DN
1
S6
CNT = 0
CNT ≠ 0

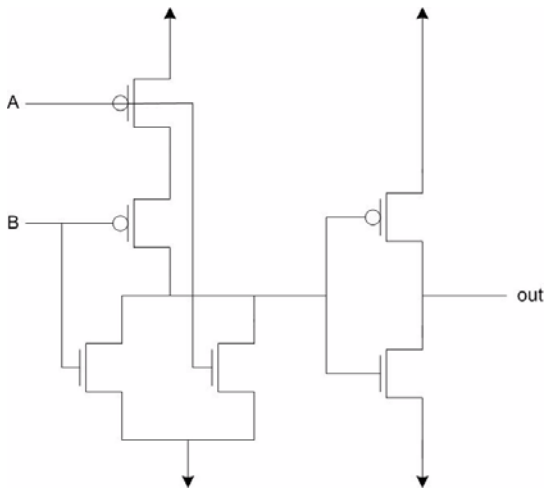
**11. [Analysis of CMOS Circuits—10 Points]**

a. [3 Points] Which of these CMOS circuits implements the following functional specification? $z = 0$ if and only if at least two of the input variables $a$, $b$, $c$ assume the 1 value; $z = 1$ otherwise.

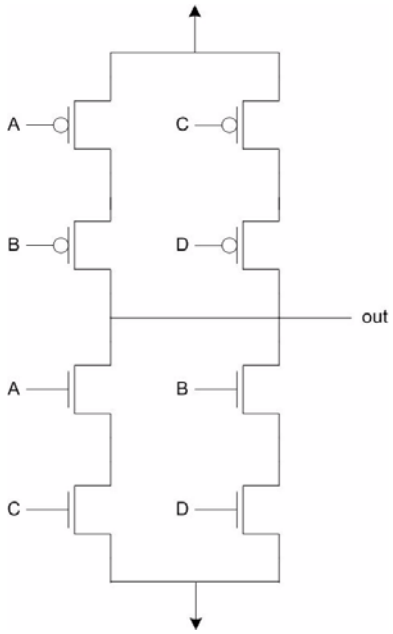



A. B. * C. D. E.

b. [3 Points] For the CMOS circuit at right, express out as a function of A and B.
out = (A'B')' = ((A + B)')' = A + B



c. [4 Points] Analyze the CMOS gate shown here and fill in the "OUT" column of its truth table with "0", "1", "Hi-Z", or "SC" (for short circuit).




| *A* | *B* | *C* | *D* | *OUT* |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 1 | |
| 0 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 0 | |
| 0 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 0 | |
| 0 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | |
| 1 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 0 | |
| 1 | 1 | 1 | 1 | |