1. Fill in each blank or circle the best answer. **[10 points, -2 per wrong or blank answer, min 0]**

a) !(A+B), when expanded into canonical sum-of-products form, has ___1___ minterms.

b) A 4-bit 8 to 1 MUX would require ___3___ select lines.

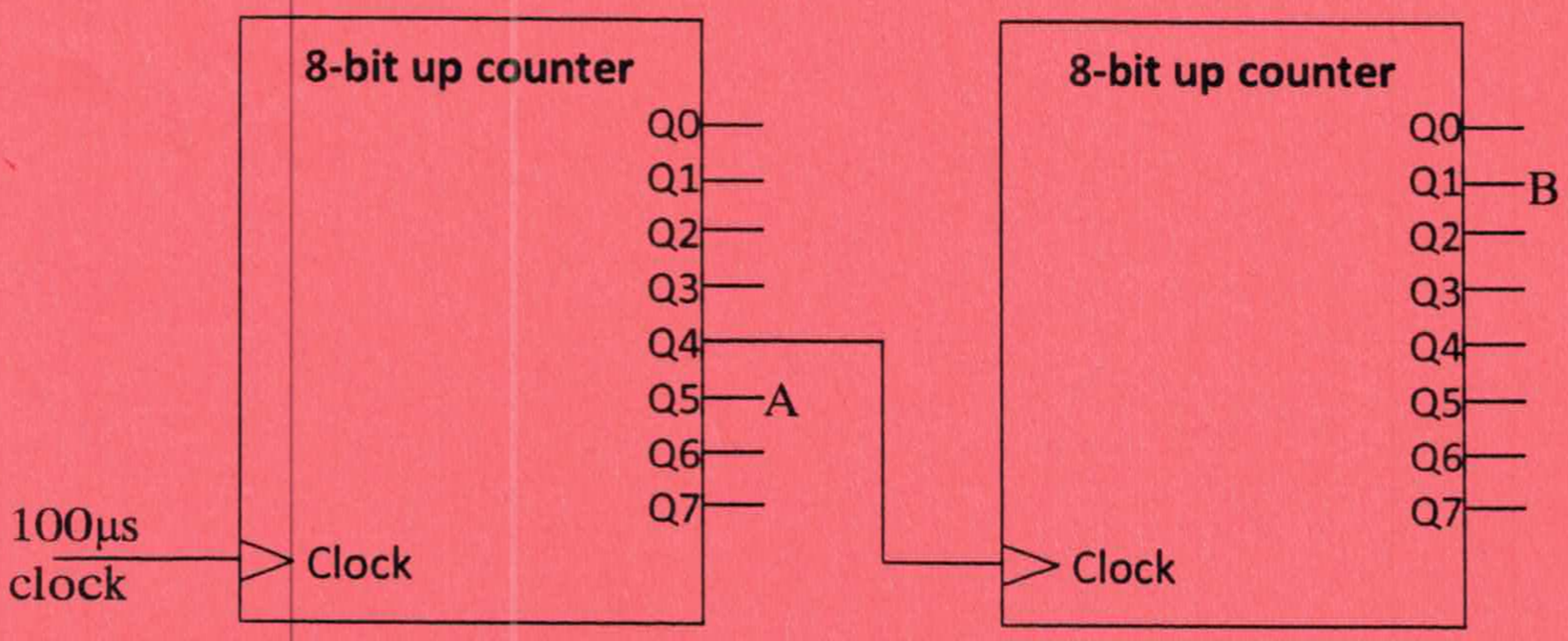
c) You are treating the 8-bit numbers A[7:0] and B[7:0] as unsigned numbers. If you set B[3:0]=A[3:0] and B[7:4]=4'b0000,

B is now equal to A ***plus / minus / times / modulo / divided by*** ***2 / 3 / 4 / 8 / 16 / 32***

d) When building a 1024 by 4 memory out of a square memory of minimum size, the row decoder will have __6__ bits of input while the column MUX will have __4__ bits needed for its select input.

e) In CMOS you'd need at least __10__ transistors to implement a 4-input OR gate.

2. Two 8-bit counters are connected as follows. Assume that these two counters are both working as "counting up" (note that Q7 is the MSB for the output of a counter); also, the period of the input clock signal for the 1st counter (the one on the left-hand side) is equal to ***100 µs***. Calculate the *period* of the output signals at A and B, respectively. *Your answer must be in ms*! **[4]**

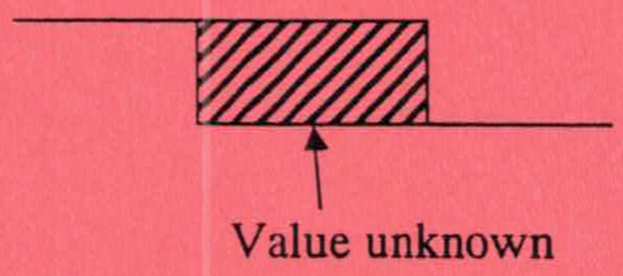


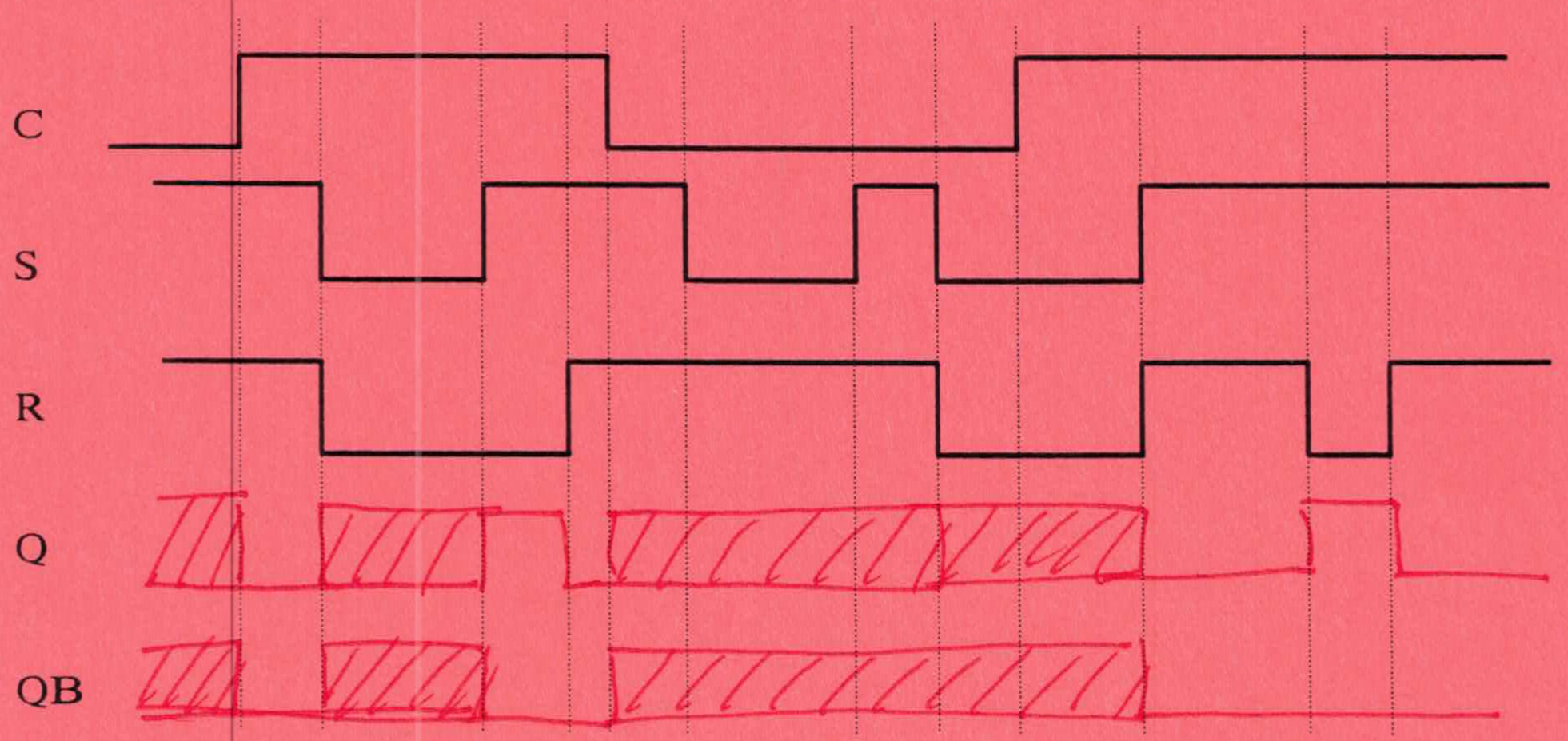

Period of A: __6.4__ ms

Period of B: __12.8__ ms

3. Complete the following timing diagram for an SR-latch with enable. You may assume that the time scale is such that the gate delay is extremely small and your answer should not reflect those delays. Changes shown to be simultaneous are exactly simultaneous. **[5 points]**

   If the value is unknown (or oscillating) at some point, clearly indicate that with hashes (like this)



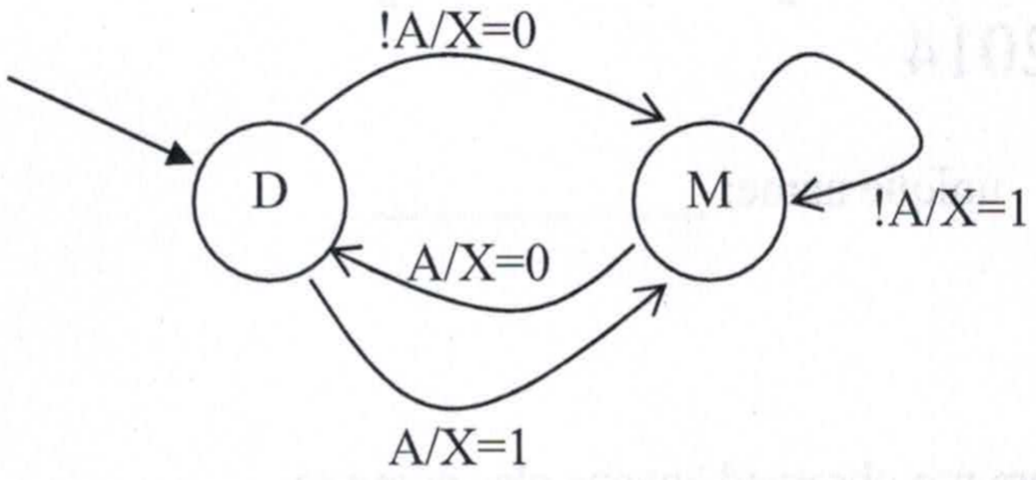

4. Find the minimal sum-of-products using this Kmap. Clearly show your work. **[5]**

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | d | 1 | 0 | 1 |
| 01 | 0 | 1 | 1 | 1 * |
| 11 | 1 * | d | d | 0 |
| 10 | 1 | d | 0 | 1 * |

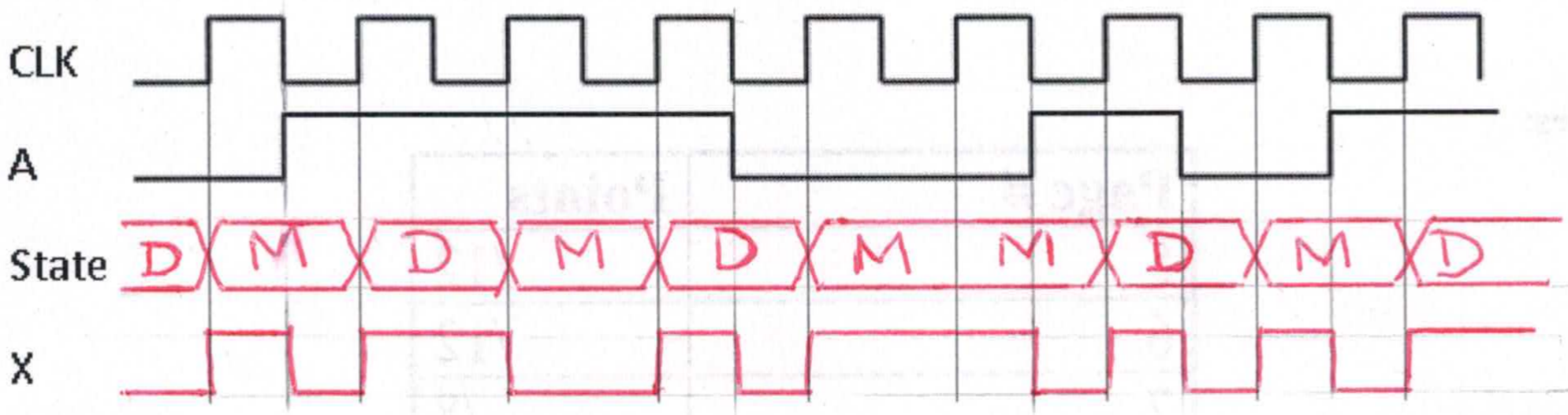
Minimal SoP:

$\bar{B}\bar{D} + A\bar{C} + \bar{C}D + \bar{A}BC$

1. Consider the following Mealy state transition diagram.

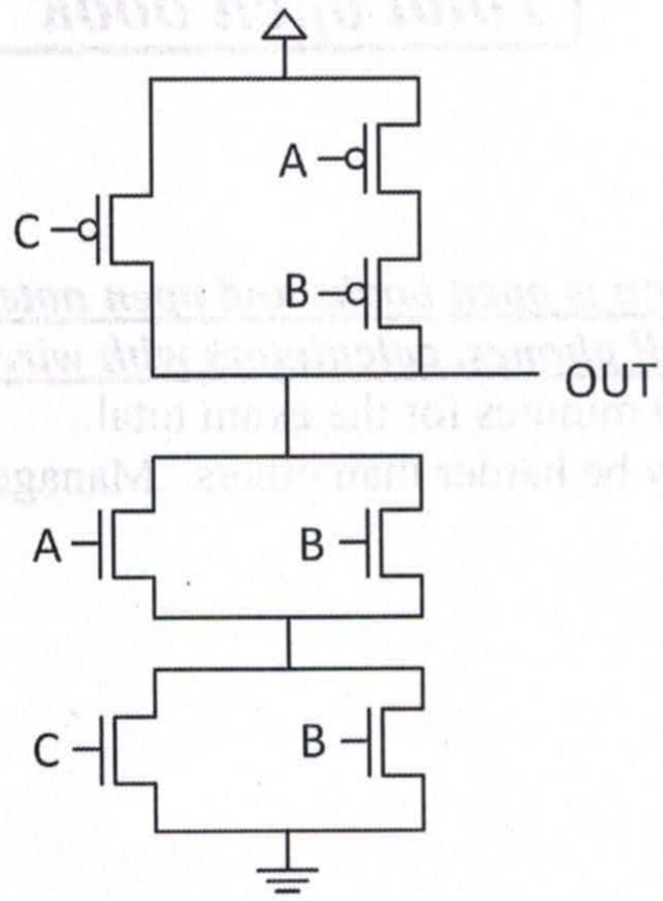



Complete the following timing diagram. Assume that before the first rising edge, the machine is in its initial state. **[6 points]**




2. Consider the transistor diagram below. Fill in the truth table with either "1", "0", "Hi-Z" or "Smoke" (the last if OUT is connected to both Vcc and Ground). **[6 points]**

| A | B | C | OUT |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | Smoke |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | Smoke |
| 1 | 1 | 1 | 0 |



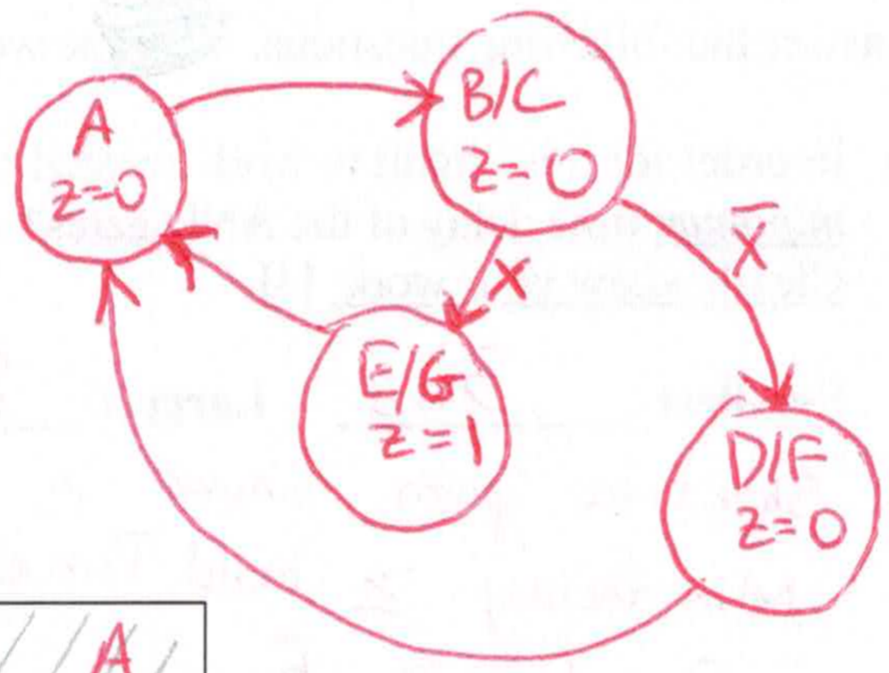
3. The following table describes a state machine.

| Present state | Next state x=0 | Next state x=1 | Output Z |
|---|---|---|---|
| A | B | C | 0 |
| B | D | E | 0 |
| C | F | G | 0 |
| D | A | A | 0 |
| E | A | A | 1 |
| F | A | A | 0 |
| G | A | A | 1 |

Minimize the number of states in this machine and *draw* the state transition diagram which describes this minimized machine. Show your work. **[12 points]**

Merged States: E/G, B/C, D/F

| | | | | | |
|---|---|---|---|---|---|
| A | X | | | | |
| B | X | D E / B C | | | |
| C | X | F G / B C | F G / D E | | |
| D | X | A A / B C | A A / D E | A A / F G | |
| E | A A / A A | X | X | X | X |
| F | X | A A / B C | A A / D E | A A / F G | A A / A A | X |
| | G | A | B | C | D | E |

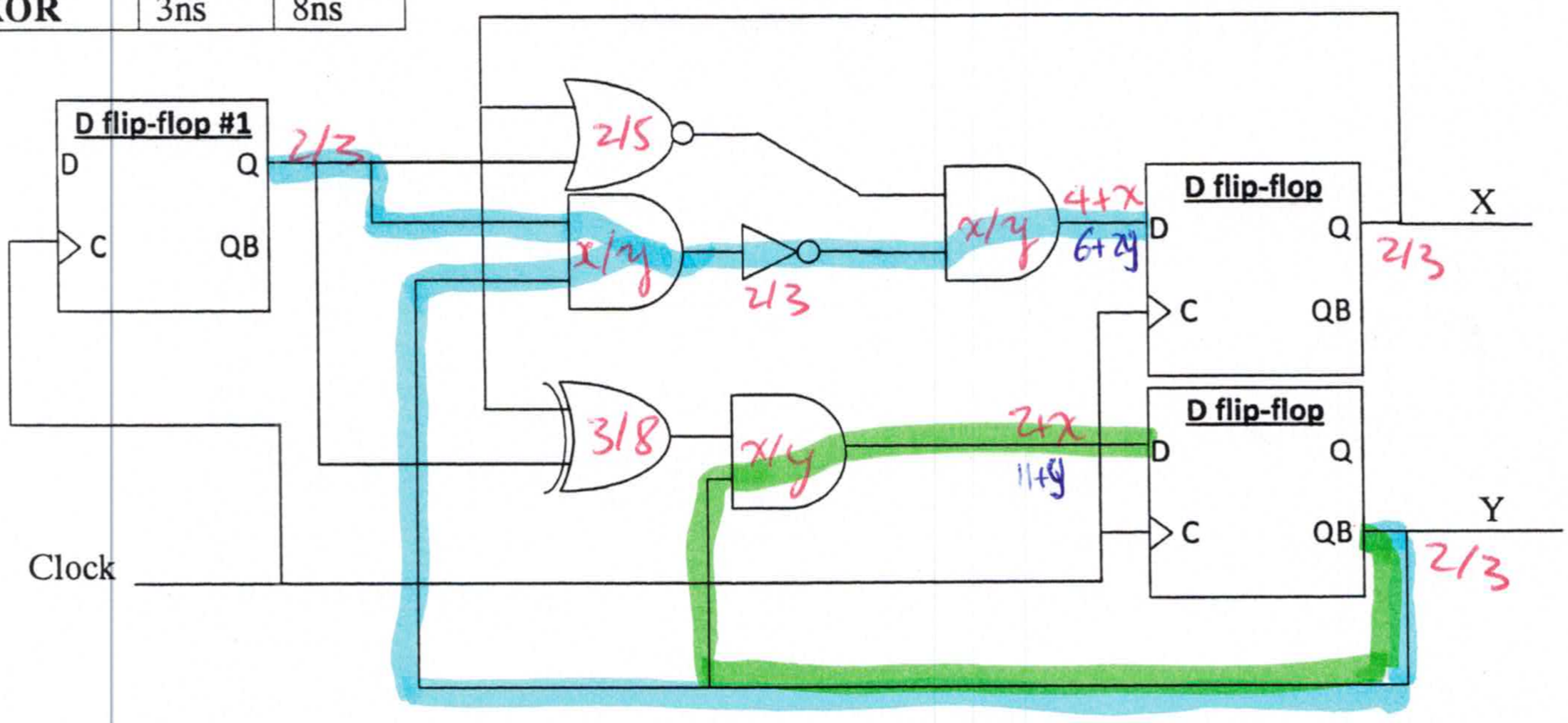



non-equivalent Pairs

Pass 0: AG, BG, CG, DG, FG, AE, BE, CE, DE, EF
Pass 1: AB, AC, AD, AF, BD, BF, CD, CF,
Pass 2: —

| | Min | Max |
|---|---|---|
| **AND** | ??? | ??? |
| **NOR** | 2ns | 5ns |
| **NOT** | 2ns | 3ns |
| **XOR** | 3ns | 8ns |

| **DFF:** | | **Min** | **Max** |
|---|---|---|---|
| | *Clock to Q* | 2ns | 3ns |
| | *Set-up time* | 3 ns | |
| | *Hold time* | 5 ns | |




4. Answer the following questions. Assume we are clocking this circuit at 50MHz. **[8 points]**

a) In order for this circuit to work correctly, what range of values that would be acceptable for the *minimum* time delay of the AND gates? Assume the only options are integers from 1ns to 20ns. Clearly show your work. **[4]**

**Smallest** 3ns **Largest** 20ns.

Min-path shown highlighted in green .... Note: 4+x > 2+x, $\forall x$

Min Delay ≥ Hold Time.

$2+x \geq 5$

$x \geq 3$

Also acceptable to use 5ns as the Largest

b) In order for this circuit to work correctly, what range of values that would be acceptable for the *maximum* time delay of the AND gates? Assume the only options range from 1ns to 10ns. Clearly show your work. **[4]**

**Smallest** 1ns **Largest** 5ns

Also acceptable to use 3ns as the smallest

Clock Period = $1/50 \times 10^{-6}$ s = 20ns.

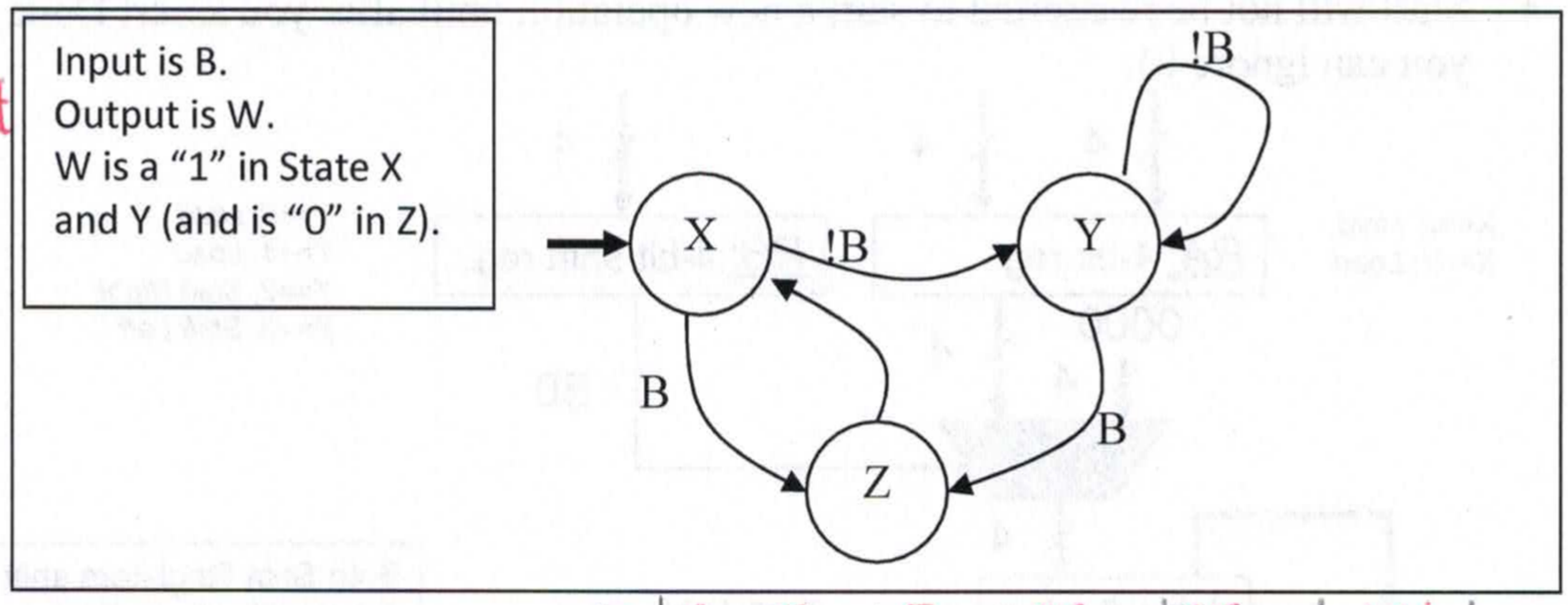
Max-path shown in blue.

Max delay + Tsetup ≤ Clock Period.

Option 1: $6+2y+3 \leq 20 \Rightarrow 2y \leq 11$, $y \leq 5.5$, $y \leq 5$ if only integral values are allowed.

Option 2: $11+y+3 \leq 20 \Rightarrow y \leq 6$

5. Design a state machine which implements the following state transition diagram. Assign state bits S[1:0] as 01 for state X, 00 for state Y, and 11 for state Z. You are to assume that you will never reach the state S[1:0]=10, so you don't care what happens in that case. You must show your work to get any credit! *You only need to compute the next state and output logic, you don't need to draw the gates or flip-flops!* Place your answer where shown, all answers must be in ***minimal*** *sum-of-products* form. **[14 points]**



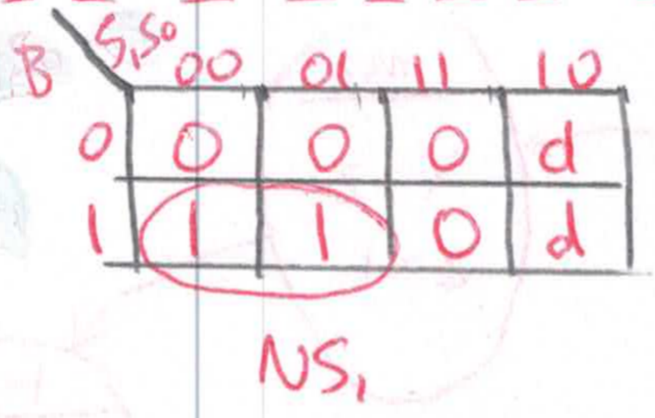

State Assignment

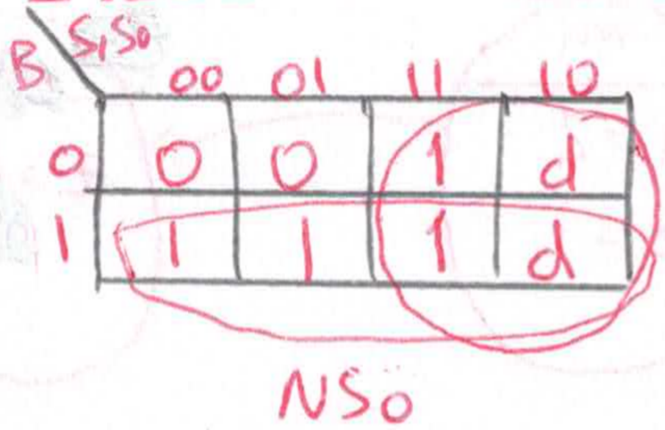
| State | $S_1$ | $S_0$ |
|---|---|---|
| X | 0 | 1 |
| Y | 0 | 0 |
| Z | 1 | 1 |

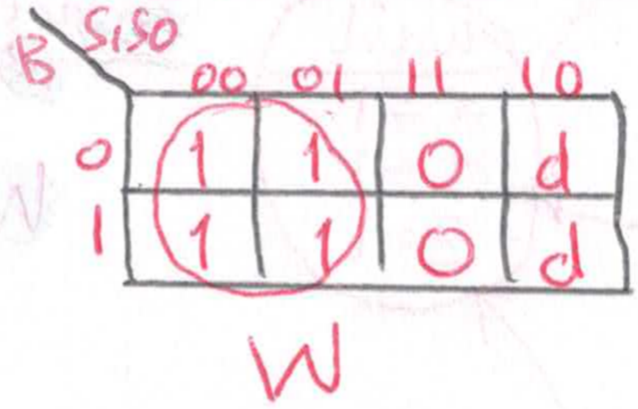
State/Output Table.

| Current | Input B=0 | Input B=1 | Output W |
|---|---|---|---|
| 0 0 | 0 0 | 1 1 | 1 |
| 0 1 | 0 0 | 1 1 | 1 |
| 1 1 | 0 1 | 0 1 | 0 |

| $S_1$ | $S_0$ | B | $NS_1$ | $NS_0$ | W |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | d | d | d |
| 1 | 0 | 1 | d | d | d |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |



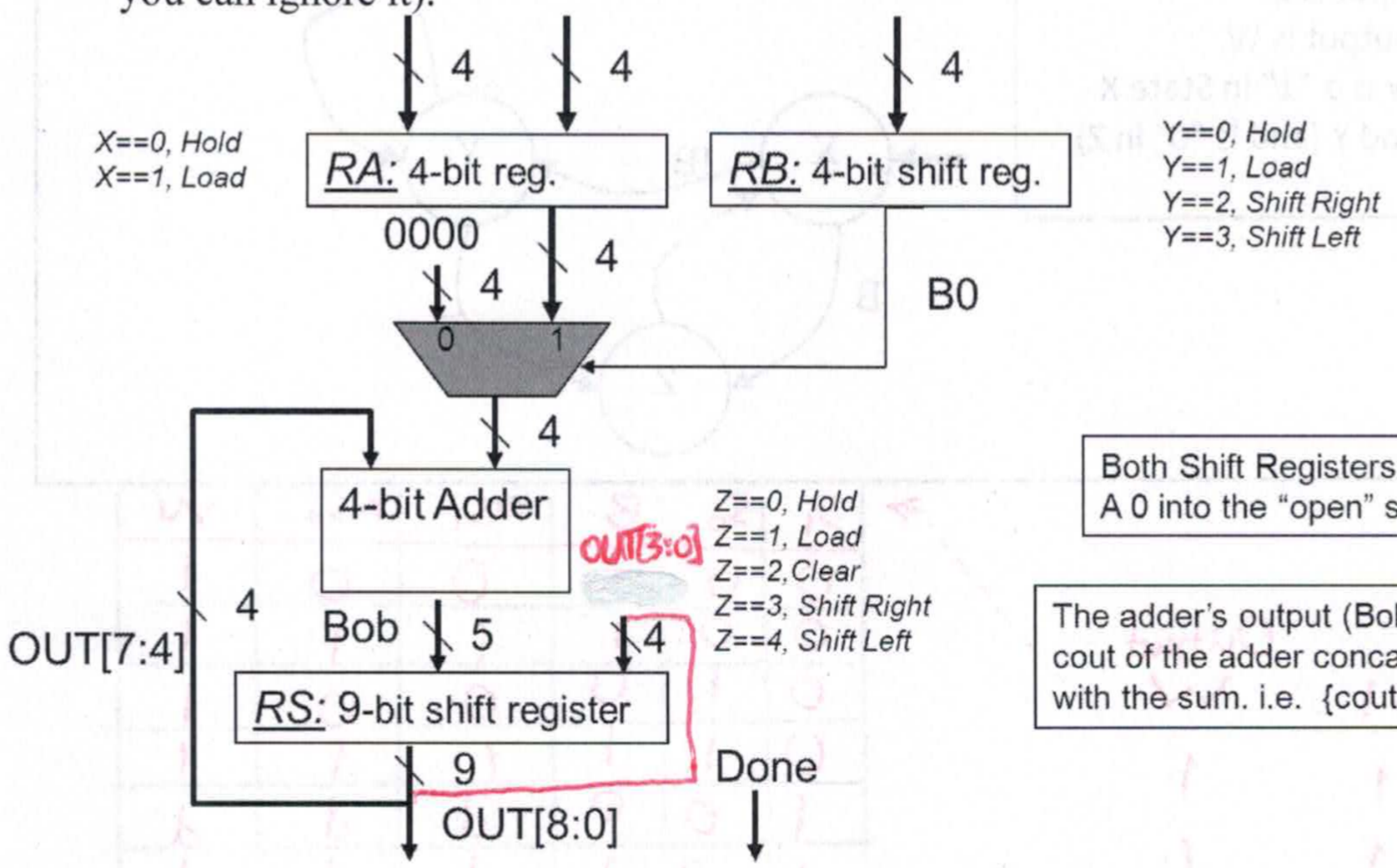

*(Be sure all are in minimal sum-of-products form!)*

NS1= $\overline{S_1}B$

NS0= $S_1 + B$

W= $\overline{S_1}$

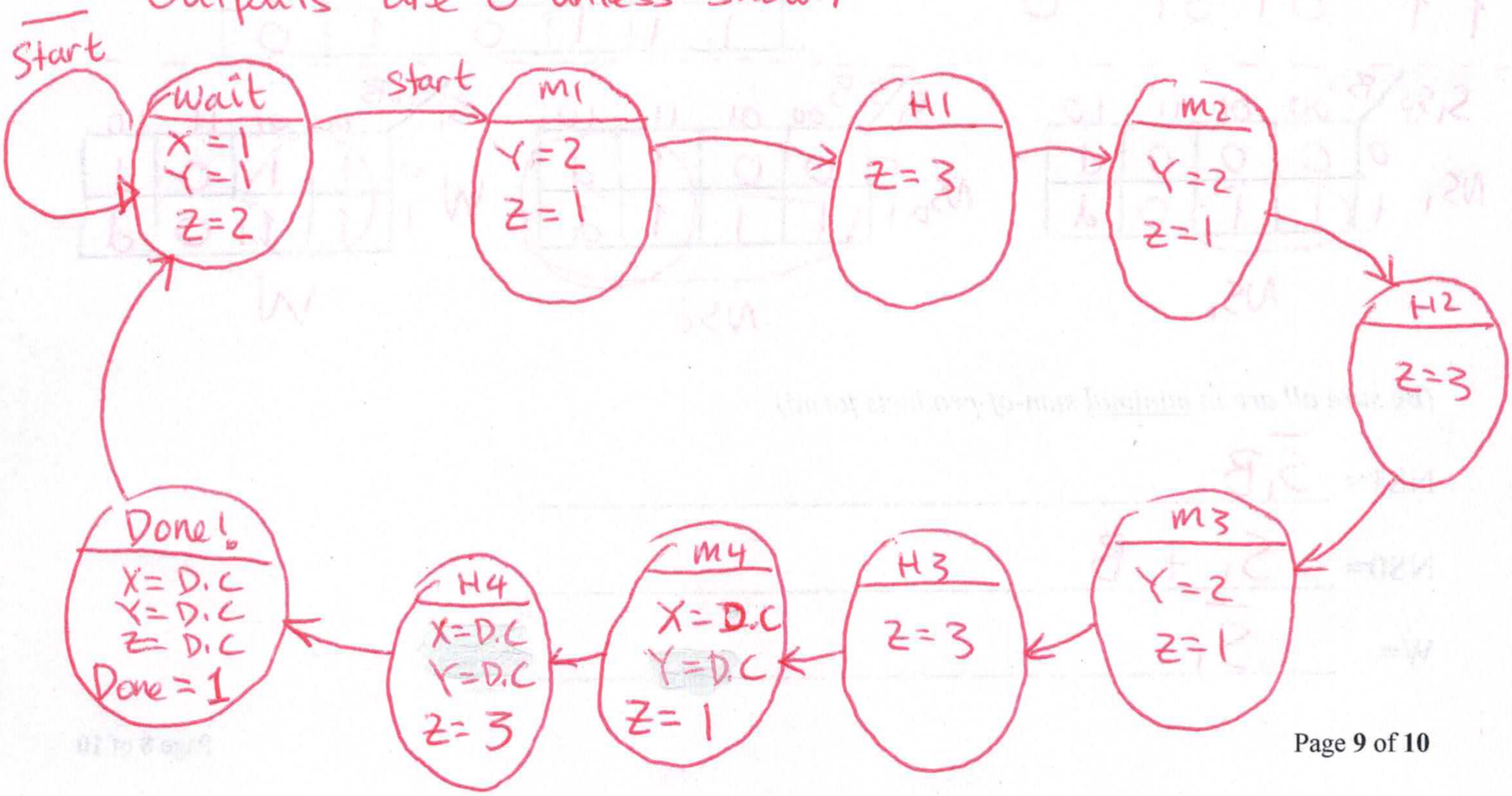
6. The following circuit is an unsigned binary multiplier which computes A*B. Draw the control state transition diagram. **[15 points]**
You are to assume:
   - Start will be asserted for one cycle and that A and B are available on the rising edge that start is asserted
   - On any rising edge Done is asserted, Sum should be correct.
   - Done need only be held high for 1 cycle.
   - Start will not be reasserted to start a new operation until after you assert Done (and if it is, you can ignore it).




Both Shift Registers shift A 0 into the "open" spot.

The adder's output (Bob) is the cout of the adder concatenated with the sum. i.e. {cout,sum[3:0]}

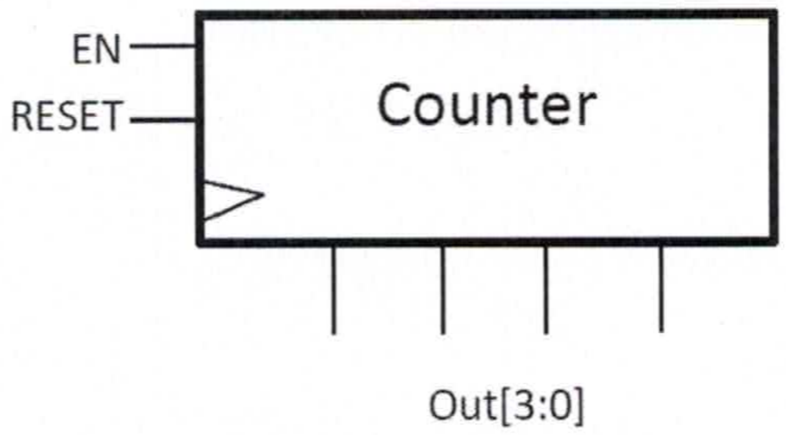
Outputs are 0 unless shown



7. Binary Coded Decimal (BCD) encoding is used to represent decimal numbers. Each digit of the decimal number is binary encoded with 4 bits. For example, 12 will be encoded as 0001 0010 and 45 as 0100 0101.

Using gates and two 4-bit counters of the type to the right, design a two-digit *saturating* BCD counter that counts from 00 to 99. Your BCD counter should have three inputs: clock, reset and enable. On each rising edge:

- If reset is a 1, your BCD counter should go to 00.
- Otherwise, if enable=1 it should increment to the next value (staying at 99 if already at 99).
- Otherwise it should hold its value.




It should have outputs LSD[3:0] which is the least-significant BCD digit and MSD[3:0] which is the most-significant BCD digit. Your grade for this problem will be based on correctness and clarity. Your counters must be clocked using the "clock" input. Assume the supplied 4-bit counter will ignore the EN input if RESET is 1. **[15 points]**

$ENL = enable \cdot \overline{MSD3.MSD0.LSD3.LSD0}$
$RESETL = LSD3.LSD0 \cdot \overline{MSD3.MSD0} + reset$
$ENM = enable.LSD3.LSD0.\overline{MSD3.MSD0}$
$RESETM = reset$

Let: $X = LSD3.LSD0$
$Y = \overline{MSD3.MSD0}$
$ENL = enable.(\bar{X}+Y)$
$RESETL = reset + X.Y$
$ENM = enable.X.Y$
$RESETM = reset$