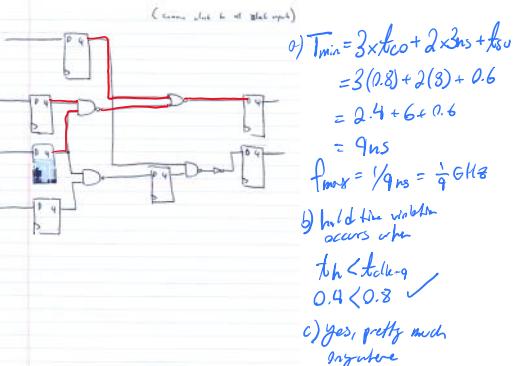


- things for formula sheet
- fsm minimization algo
  - fsm knapping for flipflops
  - mux syntax
  - WANT hold time is less than the total propagation delay
    - o (minimum pd)

1

## MECHTRON 3TB4 — Some Sample Questions

1. Consider the circuit below. NAND gates and NOR gates have delay 3 ns, and NOT gates have delay 2 ns. For the flip-flops, we have  $t_{su} = 0.6$  ns,  $t_{so} = 0.8$  ns, and  $t_h = 0.4$  ns.



- (a) What is the maximum clock frequency for this circuit?  
(b) Would it be possible to have a hold time violation if the output of a flip flop was connected directly to the input of another flip flop?  
(c) Is it possible to have a hold time violation anywhere in the given circuit?

2. You are using 16-bit numbers to represent values in the range  $-100 \leq x < 100$ .

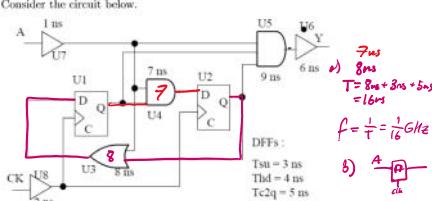
a)  $100_{10} = 1100100_2 = 7\text{ bits for integer values} + 1\text{ bit for sign}$   
should use  $Q(7+1).(16-(7+1))$  for more precision  
 $= Q8.8$

b)  $0111111111111111_2 = 0x-1+1+2+4+8+16+32+64+0.5-0.25+0.125+0.0625$   
 $+ 0.03125+0.015625+7.8125 \times 10^{-3} = 127.9921875$

c)  $0000000000000000_2 = 7.8125 \times 10^{-3}$

- (a) What Q format should you use? Justify your answer.  
(b) What is the largest positive number that your format can represent?  
(c) What is the smallest positive number that your format can represent?

3. Consider the circuit below.



- (a) If the input A is held constant, what is the maximum clock frequency?  
(b) Show how you would add a synchronizer for the asynchronous input A.

4. Implement the function

$$f = (a \cdot b) + (c \cdot \bar{a})$$

using only 2-input LUTs (4-to-1 multiplexers). For full marks, you must use the minimum number of LUTs.

5. Consider the following Verilog code.

Always @ (old or  
newdata[7:0];)  
begin  
end

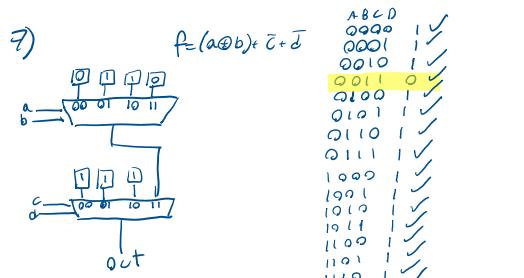
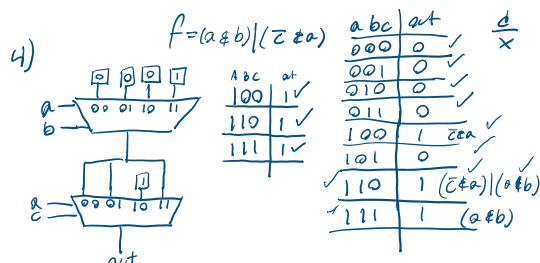
What timing issue does this create?

6. What is the resulting format of multiplying two 16-bit numbers and then performing the following operations: right shift by 15 bits (reserving sign), and then invert the 17th bit. Under what conditions would inverting the 17th bit be significant?

7. Implement the function

$$f = (a \oplus b) \cdot \bar{c} + \bar{d}$$

using only 2-input LUTs (4-to-1 multiplexers). For full marks, you must use the minimum number of LUTs.



9) module register(input clk, reset, newdata[7:0], load);  
Reg [7:0] data;  
always@ (posedge clk or posedge load)  
if (load) data <= newdata;

```

    always@(posedge clk or posedge load)
        if(load) data <= newdata;
        else if(next) data <= $();
endmodule

```

10) module q-to-n ( $i$  [3:0] value, output decimal, wire edges [7:0]);

assign edges = { $i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7]$ };

always @(\*) begin

: If ( $i \geq 4'600$  &  $i \leq 4'61001$ )

decimal <= 1;

else decimal  $\leq \emptyset$ ;

end

## 2nd module

3

8. Assume you wish to implement an FIR filter with the following coefficients and a 16-bit datapath:

$$b_0 = 0.9001, b_1 = -0.6500, b_2 = 0.3000.$$

(a) Which Q format would you use? Justify your answer.  
(b) What is the error in the representation of  $b_2$ ?  
(c) Suppose that you wished to compute  $b_0 + b_1 + b_2$  – does the order that you do the summation make a difference? Justify your answer.

9. Write a Verilog module that implements an 8-bit register with load and synchronous reset. This means that new data should be loaded into the register when the load signal is high and the reset should be synchronous with the clock. You will be marked for syntax.

10. Write a Verilog module that takes as an input a four-bit value and has two outputs: (i) a single bit that is set to 1 if the input corresponds to a value between 0 and 9 in decimal and (ii) an 8-bit value that has the four least significant bits equal to the input and the four most significant bits equal to the reverse of the input. You will be marked for syntax.

11. Design a Moore FSM for a circuit which has a 2-bit input  $X$  and a single-bit output  $Y$ . The output  $Y$  becomes 1 if the cumulative sum of the numbers in  $X$  is a multiple of 3. Show your implementation using D flip-flops and logic gates.

12. An input sequence of 0's and 1's arrives synchronized with the clock. Design a Mealy FSM that recognizes the sequences 000 or 001 (outputs 1 if either of those sequences occur). Show your implementation using D flip-flops and logic gates.

13. Consider the following Moore FSM, with input  $x$  and output  $z$ . Derive an equivalent state machine with the minimum number of states.

Present State	Next State $x = 0$ $x = 1$		Output
0	1	4	0
1	2	5	0
2	2	5	0
3	2	3	1
4	5	3	1
5	6	3	0
6	6	5	0

14. The following code is intended to implement a counter that counts down on positive edges of clock cycles from 7 to 0 (in binary) and then back again to 7 (and repeats this forever). Correct the following Verilog code with the minimum number of changes (you should not have to add or delete lines). The counter is `c3:c2:c1`, with `c3` the MSB.

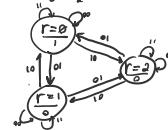
8) a) 16-bit datapath,  
use Q1.15, with 1 being for the sign bit  
& 15 for max allowable precision

$$b) 0.\underline{1}1001100110110$$

11) choice of state: remainder after  $\div 3$

$$\text{ex: } C = 2, S(1m) = 2m \rightarrow C = 2 \quad 3 + 2l \rightarrow r = 1$$

$$\text{ex: Current Sum} = 3 + 0_3 \rightarrow r = 0 \quad 3 + 0_6 \rightarrow r = 1 \\ 3 + 1_2 \rightarrow r = 0 \quad 3 + 1_2 \rightarrow r = 2$$



12)

```

graph TD
    A((A)) -- "in" --> B((B))
    A((A)) -- "out" --> C((C))
    B((B)) -- "out" --> A((A))
    B((B)) -- "in" --> C((C))
    C((C)) -- "in" --> A((A))
    C((C)) -- "out" --> B((B))

```

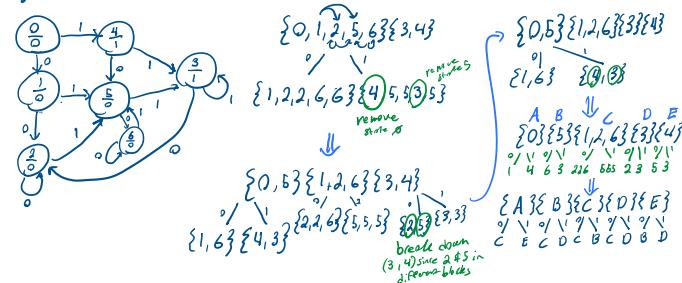
13)

6) c) if you add  $b_0 + b_2$  before  $b_1$ , there will be overflow in the MSB & the number will be misrepresented as negative.

11. Design a Moore FSM for a circuit which has a 2-bit input  $X$  and a single-bit output  $Y$ . The output  $Y$  becomes 1 if the cumulative sum of the numbers in  $X$  is a multiple of 3. Show your implementation using D flip-flops and logic gates.

(1) choice of state : maximum after  $\pm 3$

### III) Change of state: reservoir



```
module top(clk, r, c1, c2, c3);
  input r; clk;
  output c1, c2, c3;

```

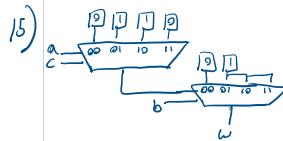
```
  reg c1, c2, c3;
  assign x=x^r;
  wire c;
  my_unit u1(clk, r, x, c1);
  my_unit u2(clk, r, c1, x, c2);
  my_unit u3(clk, r, x, c2, c3);
endmodule
```

```
module my_unit(clk, r, a, b, c);
  input clk, r, a, b;
  output c;
  reg c;
  always @(*)
    begin
      if (r)
        c = a & b;
      else
        c = a | b;
    end
endmodule
```

15. Given the logical expression  $w = (a \oplus c) + b$ , show an implementation using two 4:1 multiplexers. (The symbol  $\oplus$  denotes the xor operation.)

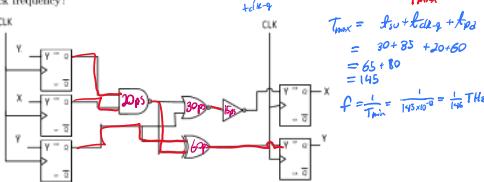
16. True or False.

- In Verilog, the statement `assign f = x|(~y&z);` and the two statements `assign h = and(~y,z);` followed by `assign f=or(h,x);` are equivalent. F
- An 8:1 multiplexer can implement any logical function of three variables. T
- Suppose  $a=4'b0011$ ,  $b=3'b011$ , and  $c=3'b101$ . Then the result of  $(a==b)?b:c;$  is  $011$ . T
- On the FPGA in the lab, the system clock has its own path, partly to aid synchronization. T
- In Verilog, the `initial` block is used to initialize the FPGA. F
- If a variable is not assigned in all possible executions of an `always` statement the synthesis process will fail. F



17. Consider a circuit that has a single input  $x$  and a single output  $z$ . Data arrive serially on  $x$  synchronized with the clock. The output  $z$  should be 1 whenever two consecutive 0's or three consecutive 1's appear. Give a Moore FSM for this system (you do not need to provide a Verilog implementation).

18. For the circuit below, suppose that the delay of a NAND gate is 20 ps, the delay of a NOR gate is 30 ps, the delay of an XOR gate is 60 ps, and the delay of an inverter is 15 ps. The flip-flops have  $t_{m1} = 30$  ps,  $t_h = 20$  ps, and  $t_{f1} = 35$  ps. What is the maximum clock frequency? Added to formula sheet



19. Give Verilog code for a custom ALU that performs the following functions on 4-bit signed integers  $A$  and  $B$ , with a 4-bit signed integer output  $Y$  that is equal to the values in the table below if the operation yields a valid result, and is equal to zero otherwise. A second output  $Z$  is 0 if  $Y$  is a valid output and 1 otherwise. There are two select lines,  $S_0$  and  $S_1$ . Your Verilog code cannot use a multiplier.

Function	$S_1$	$S_0$	$Y$
Invert	0	0	$-A$
Add	0	1	$A + B$
Subtract	1	0	$A - B$
Double	1	1	$2A$

20. Implement a 3-input XOR gate using only a 4-by-1 multiplexer and an inverter.

21. Consider a circuit that has a single input  $x$  and a single output  $z$ . Data arrive serially on  $x$  synchronized with the clock. The output  $z$  should be 1 whenever two consecutive 0's or three consecutive 1's appear. Implement the circuit using a Mealy FSM. Show your implementation using D flip-flops and logic gates.

22. Suppose that  $a = 1011$  is a number in Q2.2 and  $b = 100001$  is a number in Q3.3. Give the result of  $y = a + b$  if  $y$  is represented in Q4.3.

$$\begin{array}{r} 110110 \\ + 100.001 \\ \hline 1010.111 \end{array}$$

Q4.3  $\rightarrow$  Q4.3

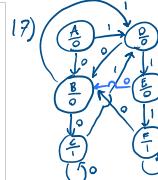
23. One of the following Verilog snippets produces a latch. Which one is it and how can the latch be removed?

Snippet 1:

```
wire A,B;
reg Y;
always @(*)
begin
  Y=A|B;
end
```

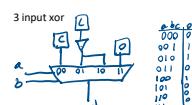
Snippet 2:

```
reg [1:0] x;
reg [1:0] q;
always @(*)
  q=x;
```



19) module ALU( $A, B, S_1, S_0, Y, Z$ )

```
  input [3:0] A, B; output [3:0] Y;
  input S1, S0; output Z;
  case({S1, S0})
    2'b00: assign Y = {~A[3], A[2:0]}; //overflow
    2'b01: assign {Z, Y} = A + B; //Z = carry bit
    2'b10: assign {Z, Y} = A - B;
    2'b11: assign {Z, Y} = A << 1; //double by bit shift
  endcase
endmodule
```



IN	PS	NS	OUT	QUT
0	00	01	0	0000
0	01	00	0	0000
0	01	01	0	0000
0	10	10	0	0000
0	10	00	0	0000
0	10	01	0	0000
1	11	11	0	0000
1	11	00	0	0000
1	11	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	11	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	11	10	0	0000
1	11	00	0	0000
1	11	01	0	0000
1	00	10	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000
1	10	10	0	0000
1	10	00	0	0000
1	10	01	0	0000
1	00	11	0	0000
1	00	00	0	0000
1	00	01	0	0000
1	01	10	0	0000
1	01	00	0	0000
1	01	01	0	0000

Snippet 2:

```
reg [1:0] x;
reg [1:0] q;
always @(*)
begin
    case (x)
        2'b00: q<=2'b01;
        2'b10: q<=2'b10;
    end case
end
```

*need default case otherwise holds value until val changes*

24. Minimize the number of states in the following FSM, where the input is  $x$  and the output is  $y$ .

Present State	Next State		Output
A	A	B	0
B	D	C	1
C	E	D	1
D	E	C	1
E	D	C	0



25. (a) Suppose that in Verilog,  $a=4'b0011$ ,  $b=3'b011$ , and  $c=3'b101$ . What is the result (in binary) of

- i.  $a+b|c$ ;  $b|c = 0b1111 \quad a+b|c = b0011 + b111 = 1010$
- ii.  $(a==b) ? b:c = 3'b011$
- iii.  $\{a,\{2|c\}\} = 0011|10101$

(b) Give an example of how an inferred latch may be produced in Verilog.

```
always@(a) begin
    case(a)
        00: a=1
        01: a=2
    endcase
end
```

*EAZEEEB3EBC, D3ECEC  
F=00*

A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

26. Parts (a) and (b) are unrelated.

(a) Implement the following function using a single 8:1 multiplexer:

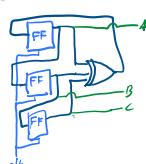
$$Z = BC + AB + A\bar{B}$$

(b) Draw the circuit implemented by the following Verilog module

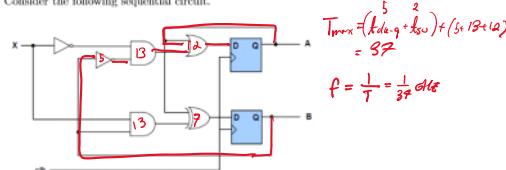
```
module test_module (clk, A, B, C);
    output A, B, C;
    input clk;

    reg A, B, C;

    always @(posedge clk) begin
        C <= B;
        A <= A ^ (B - C);
        B <= A;
    end
endmodule
```



27. Consider the following sequential circuit.



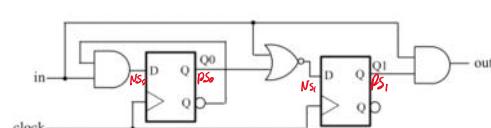
For the flip-flops, we have  $t_{dQ-Q} = 5 \text{ ns}$  and  $t_{su} = 2 \text{ ns}$ . The delays for the gates are: 7 ns (XOR), 12 ns (OR), 13 ns (AND), and 5 ns (NOT). What is the maximum clock frequency for this circuit?

28. Consider the circuit below that implements an FSM.

$$NS_1 = \overline{PS_0 + in}$$

$$NS_0 = \overline{PS_0 \cdot in}$$

8



(a) Give Boolean logic expressions for the next state (both bits) and the output.

(b) Is this a Moore or Mealy FSM? Justify your answer. *not* = *Mealy*

(c) Draw the transition diagram for the FSM.

29. The following Verilog module is intended to implement a function that inverts the even-numbered bits in a 32-bit value. Why won't it work?

```
module invert_even (input [31:0] in, output reg [31:0] out);
    integer i;
    always @(*)
    begin
        for (i=0; i< 31; i=i+1)
            begin
                if (i>2) // can't have modules inside of always
                    begin
                        if (i>2 == 0)
                            not(out[i],in[i]);
                        else
                            buf(out[i],in[i]);
                    end
            end
    endmodule
```

30. True or False

- (a) FIR filters are always stable. *T*
- (b) ASICs are reconfigurable. *F*

and  
endmodule

30. True or False

- (a) FIR filters are always stable. **F** **T**
  - (b) ASICs are reconfigurable.
  - (c) The following Verilog block is executed:
- ```
always @(*) begin
  C <= A;
  C = B;
end
```

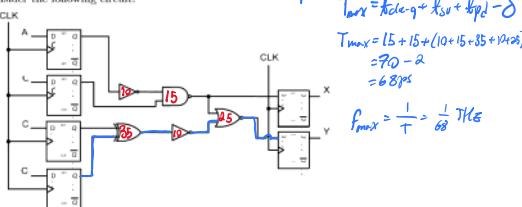
When the block is completed, C will be A.

9

- (d) Verilog is a programming language. **F** **HDL**

31. Give an FSM for a circuit that detects all occurrences of the sequence 11011. You only need to give the state transition diagram.

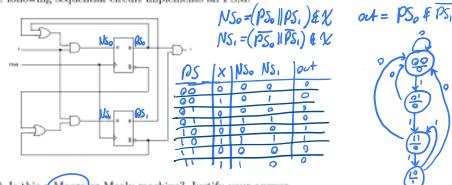
32. Consider the following circuit.



$$\begin{aligned} T_{\text{max}} &= t_{\text{clk}} + t_{\text{XNOR}} + t_{\text{NOT}} - \delta \\ T_{\text{max}} &= 15 + 15 + (10 + 15 + 35 + 10 + 5) \\ &= 90 - 2 \\ &= 68 \text{ ps} \\ P_{\text{max}} &= \frac{1}{T} = \frac{1}{68} \text{ GHz} \end{aligned}$$

The flip-flops have  $t_{\text{clk\_q}} = 15 \text{ ps}$ ,  $t_{\text{prop}} = 15 \text{ ps}$ , and  $t_{\text{q}} = 10 \text{ ps}$ . The propagation delays for the gates are: 15 ps (NAND), 25 ps (NOR), 35 ps (XNOR), and 10 ps (NOT). In addition, there is a clock skew of 2 ps between the flip-flops on the left and the flip-flops on the right (a clock edge arrives later to the flip-flops on the right). Calculate the maximum clock frequency for this circuit.

33. The following sequential circuit implements an FSM:



- (a) Is this Moore or Mealy machine? Justify your answer.

- (b) Give the state transition diagram for this FSM.

10

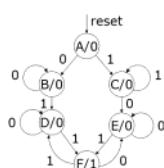
34. Write a Verilog module that has two 8-bit inputs, A and B, one single-bit output C, and a clock input. The period of the output is A clock cycles. Within each period, the output for the first B clock cycles is 1, and for the remaining clock cycles in the period, the output is 0. (You may assume that  $B \leq A$ .)

35. Suppose that  $a = 101001$  is a signed Q3.3 number and  $b = 11$  is an unsigned integer.

- $a = -1.001_2 \rightarrow -2.895$   $b = 3 \rightarrow 8.625$   
(a) What is the result if you multiply a and b? Your answer should give an appropriate representation for the result and the (binary) result in that representation.

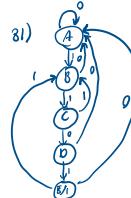
- (b) Write Verilog code that multiplies an arbitrary signed Q3.3 number and a two-bit unsigned integer.

36. Give a minimized FSM for the following Moore FSM:



37. You are to provide an FSM for a serial adder. The two numbers to be added,  $A$  and  $B$ , are serially applied to two single-bit inputs. At each rising clock edge, the next bit of  $A$  and  $B$  is available at the two inputs. The order that the bits are applied to the inputs is from least significant bit to most significant bit. There is a single-bit output. When bit  $n$  of both  $A$  and  $B$  is available at the inputs, bit  $n$  of their sum should be available at the output.  $A$  and  $B$  are of the same size, but the size is unknown. There is an additional single-bit reset input (so there are three inputs in total). When the reset signal is 0, the addition proceeds as described above. When the reset signal is 1, the system goes to the initial state and the output is 0.

- (a) Give a state transition diagram for your FSM.  
(b) Give a circuit (using D flip-flops and logic gates) that implements your FSM.



| PS | X | NS <sub>0</sub> | NS <sub>1</sub> | out |
|----|---|-----------------|-----------------|-----|
| A  | A | B               | 0               | 0   |
| B  | A | C               | 0               | 0   |
| C  | D | B               | 0               | 0   |
| D  | A | E               | 0               | 0   |
| E  | A | B               | 1               | 1   |