CSE140: Components and Design Techniques for Digital Systems

Review for Final Exam

Mohsen Imani

Sources: TSR, Katz, Boriello, Vahid, Perkowski

CAPE

Please submit your evaluations !!!!

Sources: TSR, Katz, Boriello, Vahid, Perkowski

RTL design

Use the RTL design process to design a system that has two 4-bit inputs A and B, and a single output OUT.

When the system samples the input, it will get the value of A if a control signal r is 1, B otherwise

The system samples the input and compute the output every clock cycle when a control signal c is 1. The output is:

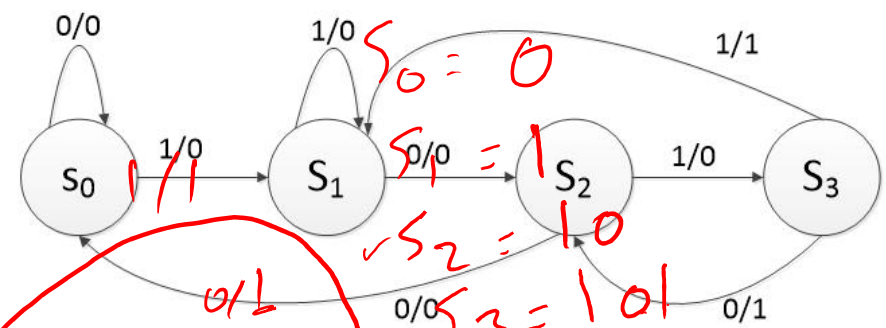
- The stored input times 2, if the input is lower than a 8-bit input L.
- The stored input divided by 8, otherwise

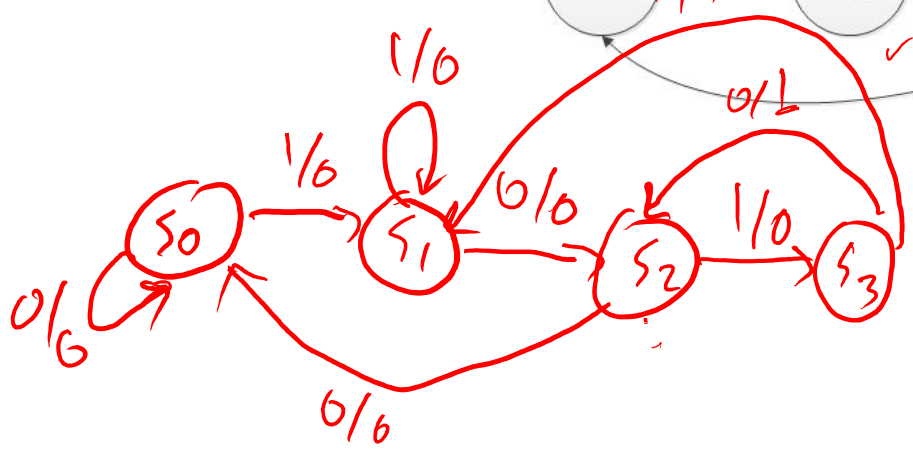
You can assume that once c is switched to 1, r cannot change its value until c is switched back to 0 first

Assume that once we start sampling values, they are either always greater or always lower than L

FSM

- Design a Mealy detector that generates a 1 on its *W output when the sequence of*
- **1010, 1011, or an overlap of the two has been detected on it's A input. Draw the state**
- diagram (FSM) using minimum number of states.
- For example:
- A 10101110011
- W 00010100000





$S_0 = 0$

$S_1 = 1$

$S_2 = 10$

$S_3 = 101$

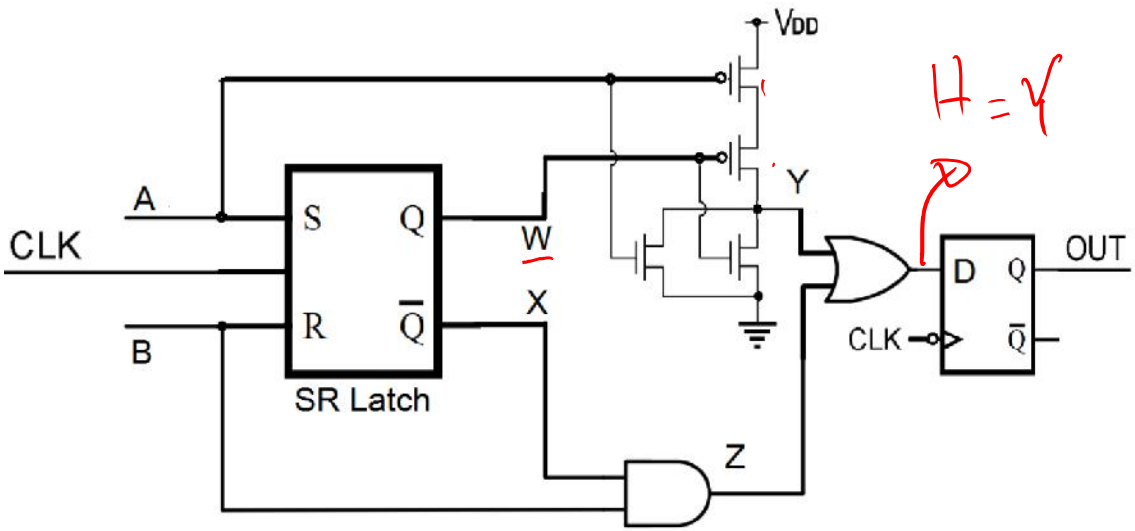
1010

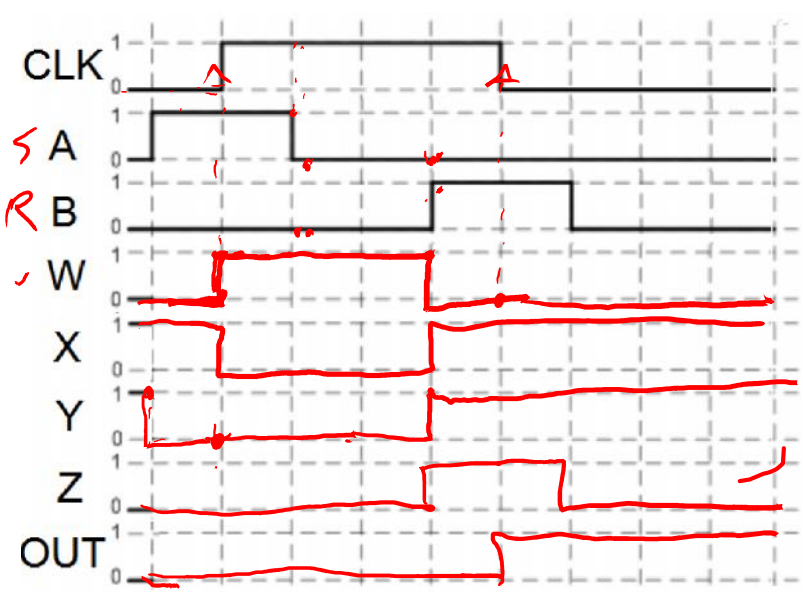
1011

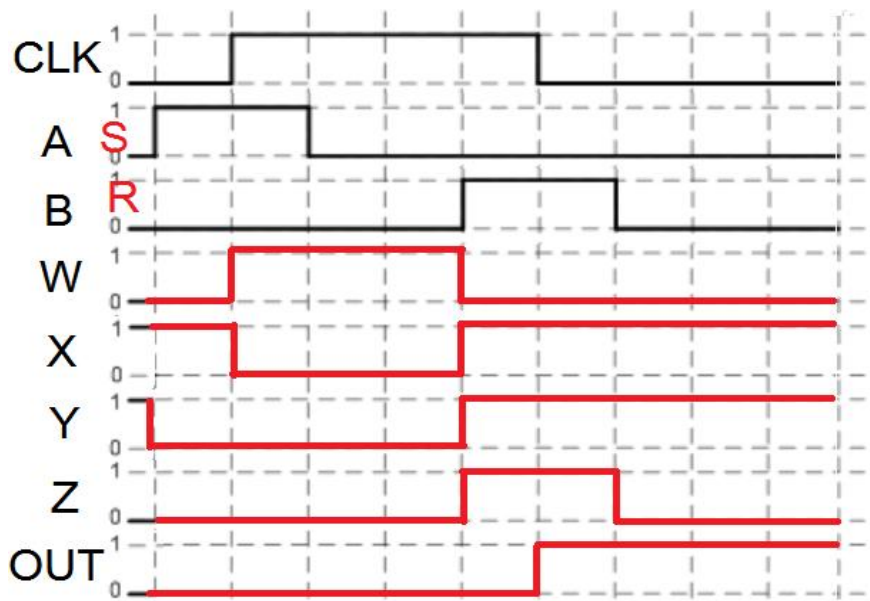
Waveform

W	A	y
0	0	1
0	1	0
1	0	0
1	1	0

NOR







Sources: TSR, Katz, Boriello, Vahid, Perkowski

Timing

You are given the RTL circuit below with Full-Adder (FA), a mux and register built from D flip-flops. The propagation delays of individua components are:

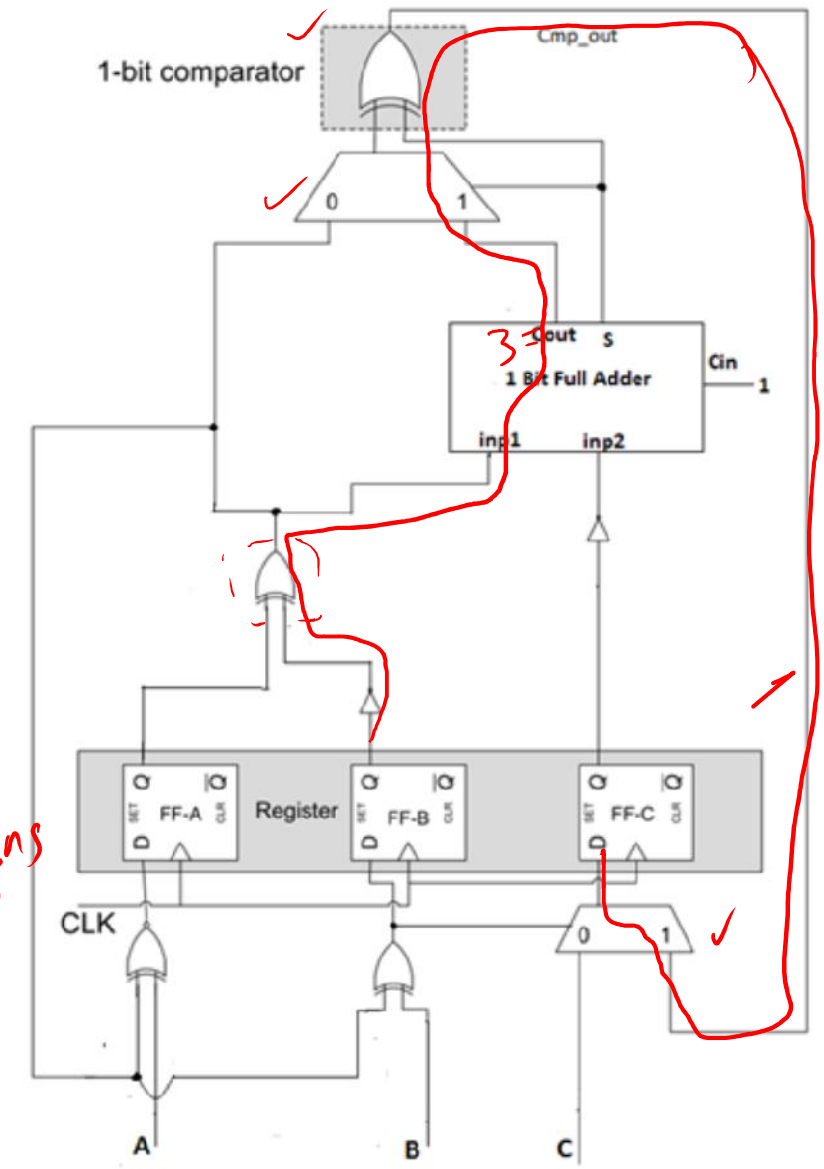
- NOT gate = 0.5ns
- All other logic gates = 1 ns
- Multiplexer delay = 2ns
- Clock->Q (D-FF propagation) delay = 2ns.
- The setup time for a flip-flop = 15ns. Assume zero clock skew.

- What is the fastest clock frequency tha this design can work at ?

$t_{pd} = 0.5 + 1 + 3 \times 1 + 2 + 1 + 2 = 9.5^{ns}$

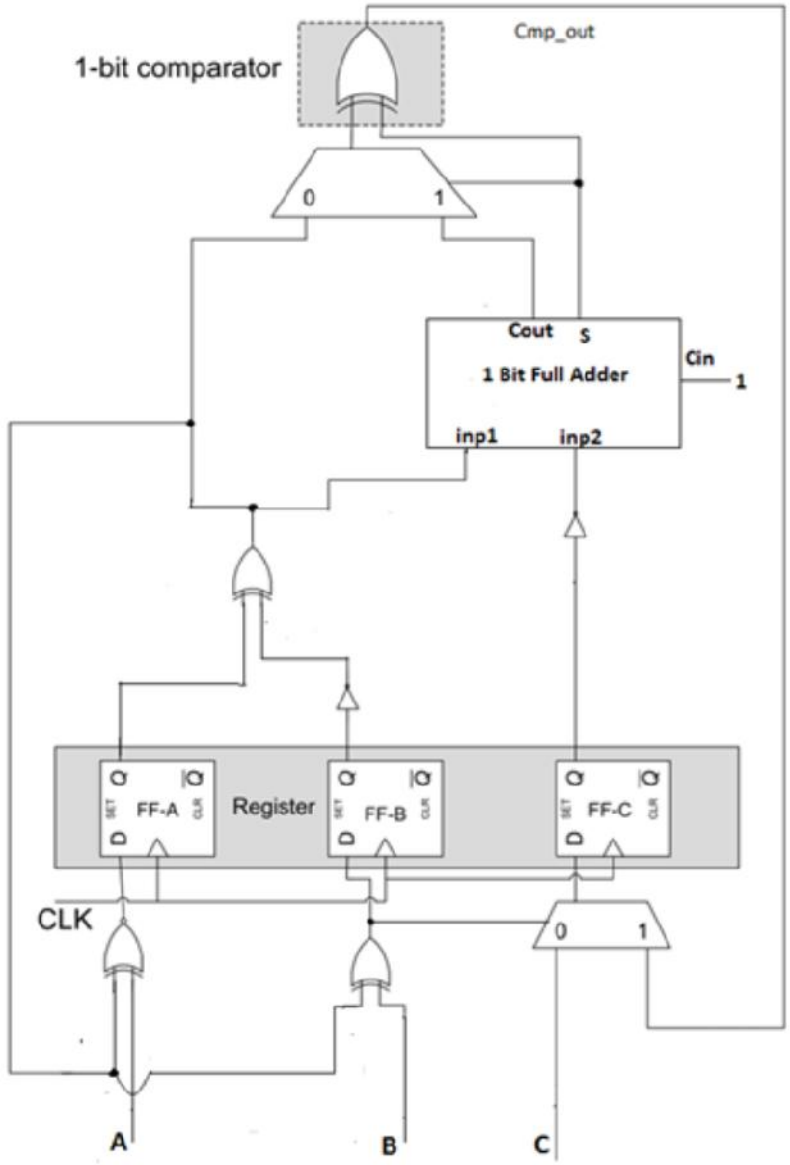
$T_C \geq 9.5 + 2 + 15$

$f = \frac{1}{T_C}$

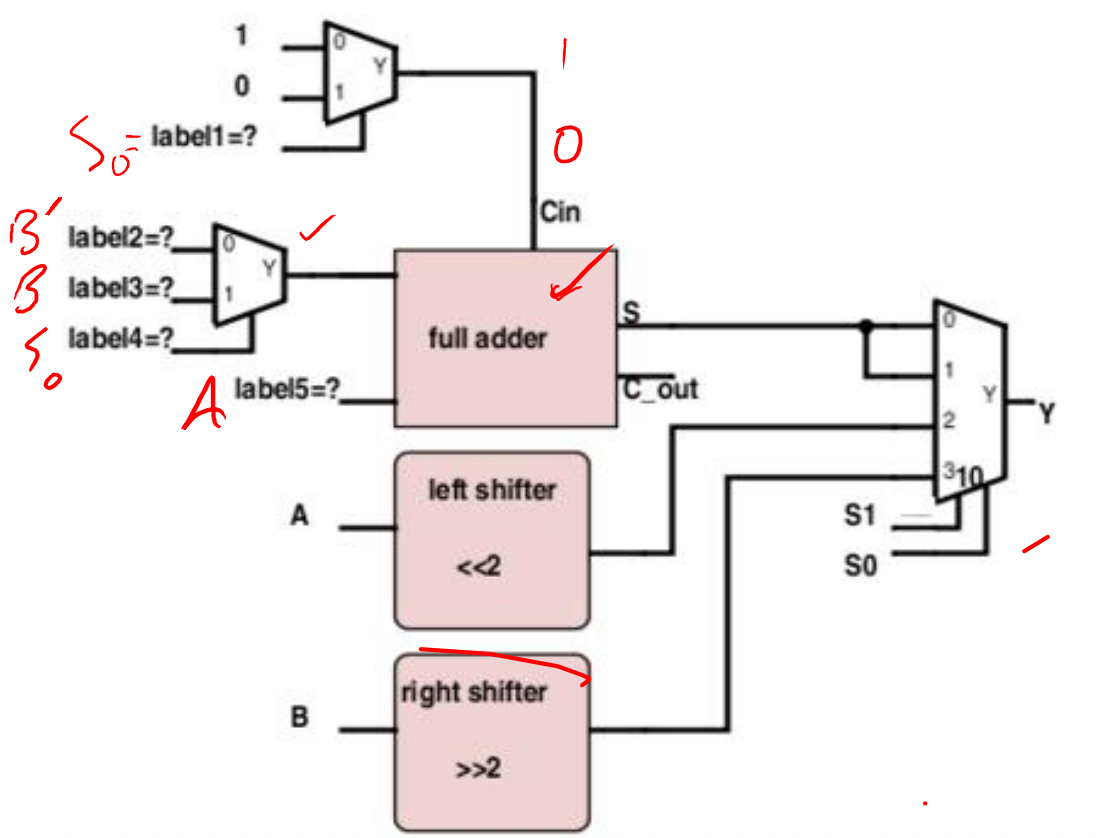


Timing

tpcq+tpd+tsetup<Tc
tpcq = 2ns
tsetup = 15ns
tpd = 0.5 + 1 + 3(1) + 2 + 1 + 2= 9.5ns
Tc > 2 + 15 + 9.5
Tc > 26.5ns
fc = 1/Tc
fc < 1/26.5 GHz



ALU



S1S0	Y
00	A-B
01	A+B
10	operation1=
11	operation2=

label1 =

label2=

label3=

label4=

label5=

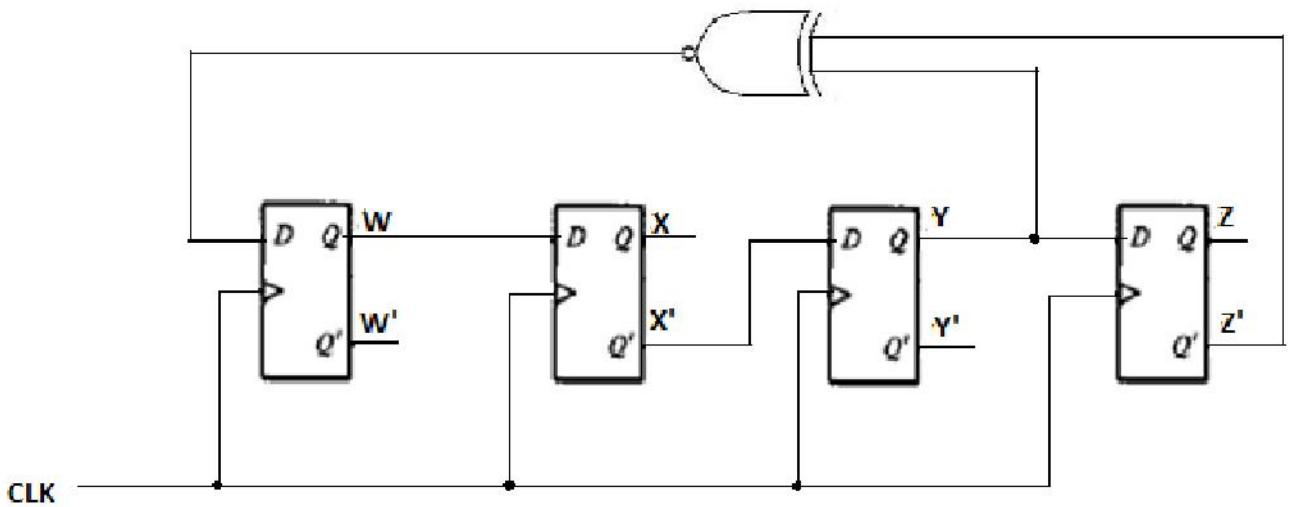
operation1 → A * 4

operation2 → B / 4

Sources: TSR, Katz, Boriello, Vahid, Perkowski

Counter

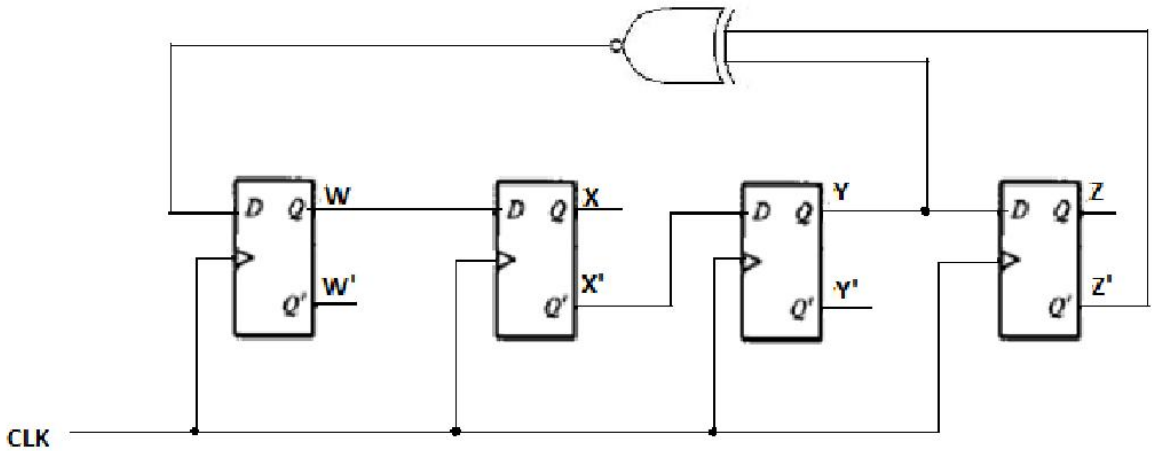
- The counter shown below goes through a repeating sequence starting from 0000.



(a) How many clock cycles does it take before the sequence repeats? List all the transitions in the sequence

0000 -> 0010 -> 1011 -> 0111 -> 0001 -> 1010 -> 1111 -> 0101 -> 1000 -> 0110 -> 1001 -> 1110 -> 1101 -> 1100 -> 0100 -> 0000

Counter



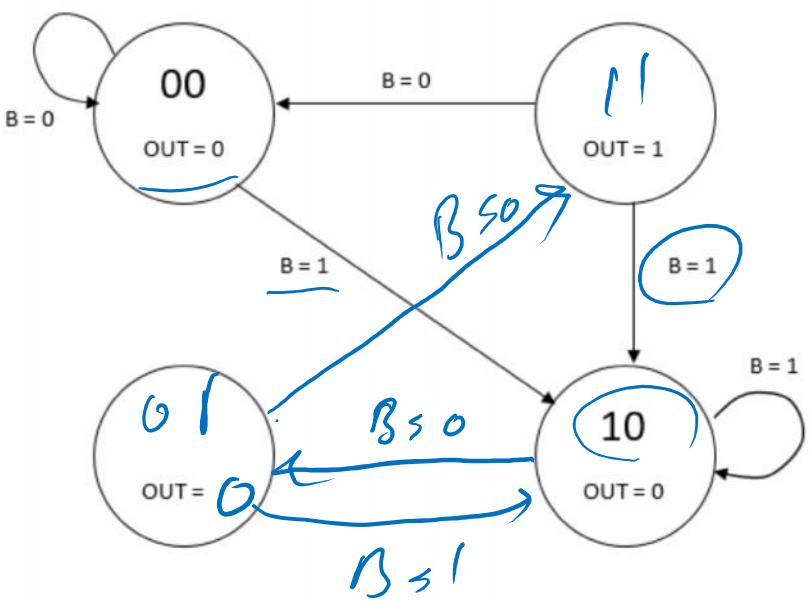
- Derive a Boolean expression that outputs a logic ‘1’ when a palindrome is detected on the current value of “WXYZ”. List all palindromes that you observe in the first 10 transitions of the sequence.
- *A palindrome is a string which reads the same in both directions. For example,10101 is a palindrome whereas 1100 is not as it is 0011 read backwards.*
- (W XNOR Z) AND (X XNOR Y)
- 1111, 0110, 1001

Sources: TSR, Katz, Boriello, Vahid, Perkowski

FSM

The following pattern detector has an input B. Once it detects a pattern, it sets OUT equal to '1'. Use the partially filled out state diagram and state table to do the following:

(a) Fill in the missing entries in the state table and complete the FSM.

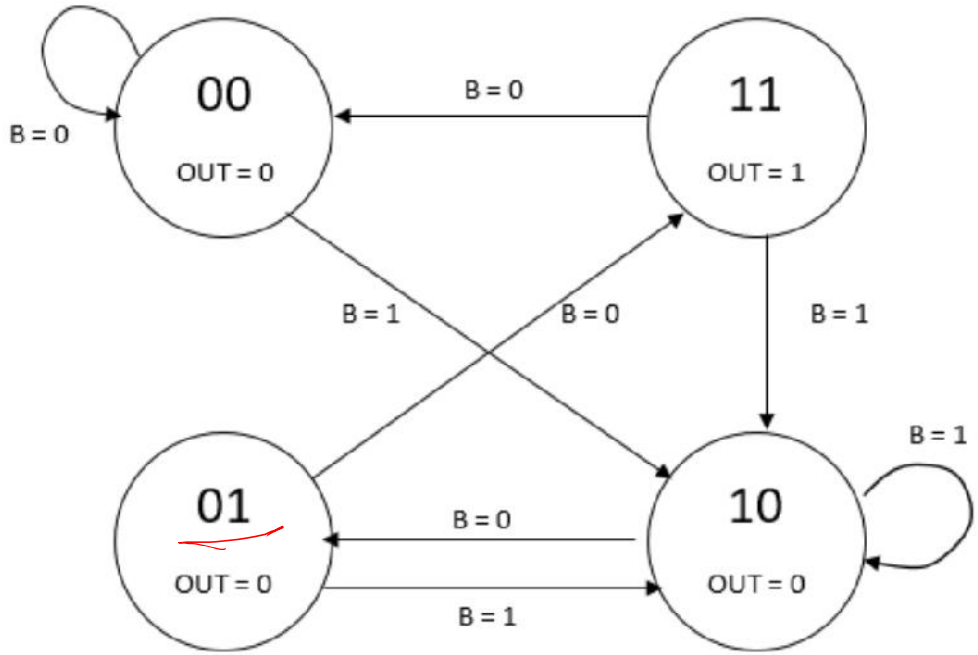


S_1	S_0	B	S_1^+	S_0^+	OUT
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	1

Sources: TSR, Katz, Boriello, Vahid, Perkowski

FSM Solution

- Which pattern is detected by this FSM?

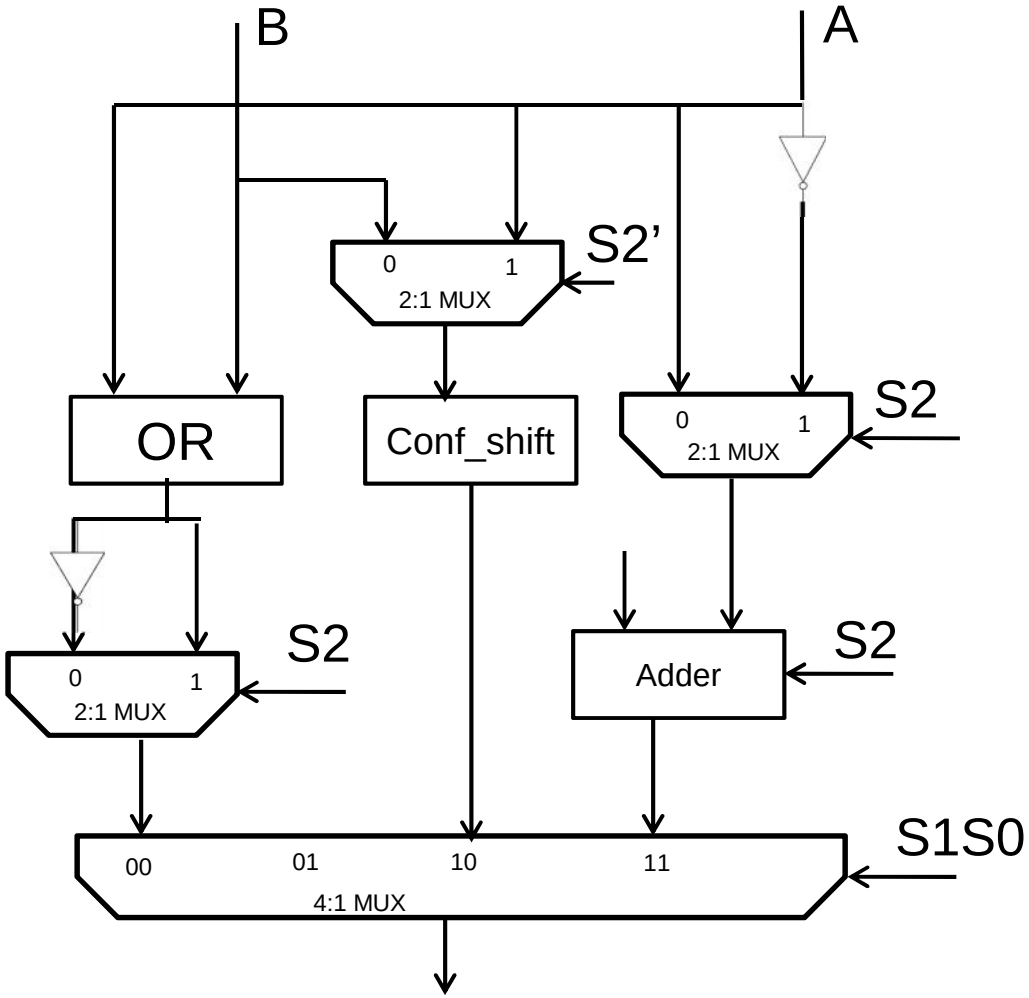


S_1	S_0	B	S_1'	S_0'	OUT
0	0	0	0	0	**0**
0	0	1	1	0	**0**
0	1	0	1	1	0
0	1	1	**1**	**0**	0
1	0	0	0	1	**0**
1	0	1	**1**	**0**	**0**
1	1	0	**0**	**0**	1
1	1	1	**1**	**0**	1

Sources: TSR, Katz, Boriello, Vahid, Perkowski

ALU design

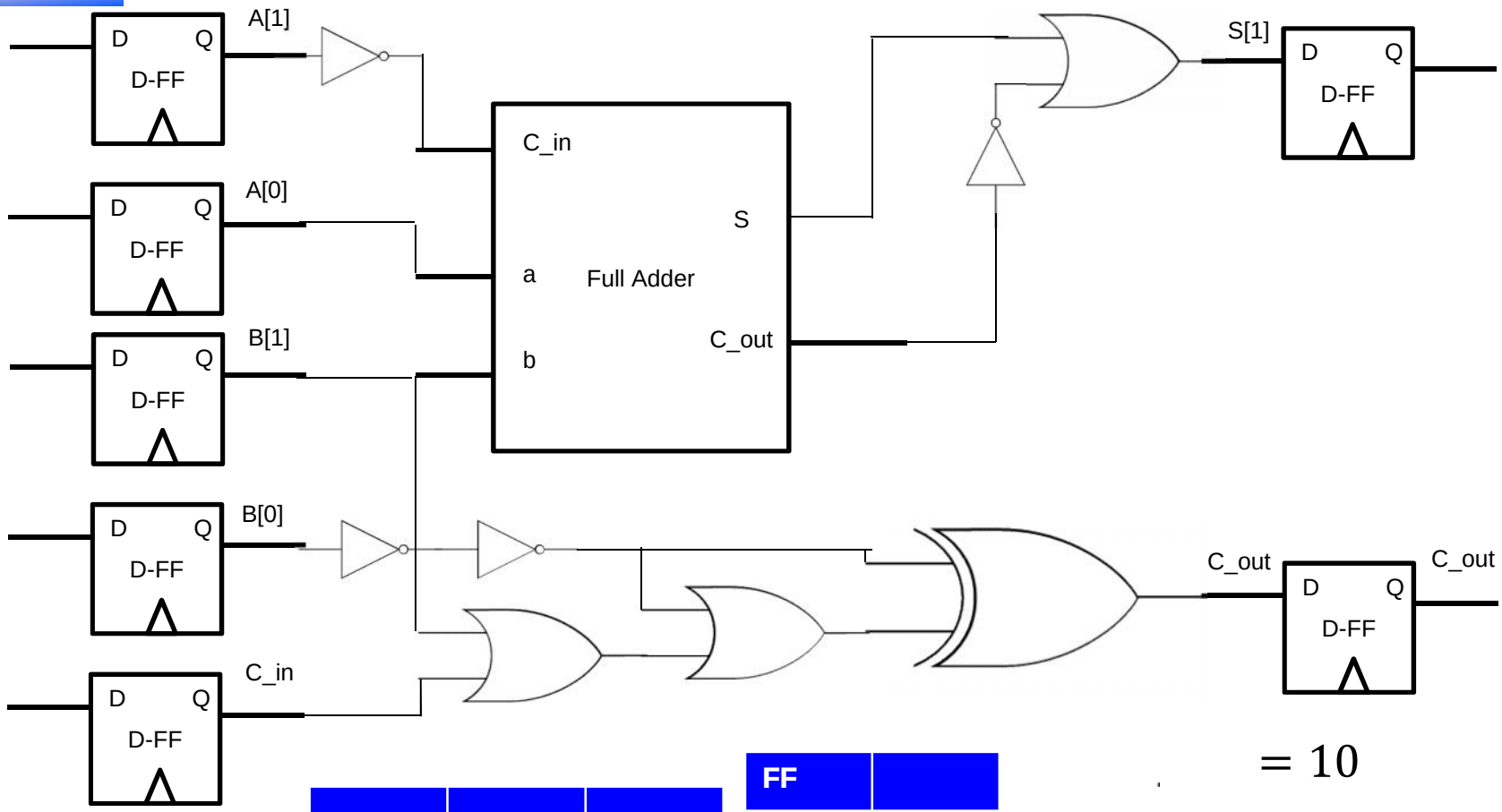
Use the previous conf_shifter (configurable shifter) to design an ALU that implements the following operations. Operands are 4-bit.



S2 S1 S0	OPERATION
000	Bitwise A nor B
001	NOT USED
010	conf_shift_4(A)
011	A + B
100	Bitwise A or B
101	NOT USED
110	conf_shift_4(B)
111	B – A

Sources: TSR, Katz, Boriello, Vahid, Perkowski

Timing constraints



$= 10$

FA	100ns	5ns
OR	25ns	10ns
XOR	80ns	70ns
NOT	10ns	10ns

FF	
	20ns
	30ns
	10ns
	70ns

- Find the maximum frequency
- Check whether there is a hold violation

Sources: TSR, Katz, Boriello, Vahid, Perkowski

Timing constraints

$\geq \quad + \quad + \quad +$

Where $= 10 + 100 + 10 + 25 = 145$

So $= \quad + \quad + \quad + \quad = 20 + 70 + 145 + 10 = 245$ ns

Therefore $= \frac{}{} = \frac{}{} * 10$ Hz $\cong 4$Mhz

$+ \quad < \quad +$

Where $= 10 + 5 = 15$

So:

$30 + 10 < 10 + 15$

$40 < 25$ → Hold time violation !!!!!!

FA	100ns	5ns
OR	25ns	10ns
XOR	80ns	70ns
NOT	10ns	10ns

FF	
	20ns
	30ns
	10ns
	70ns

$= 10$

- Find the maximum frequency
- Check whether there is a hold violation

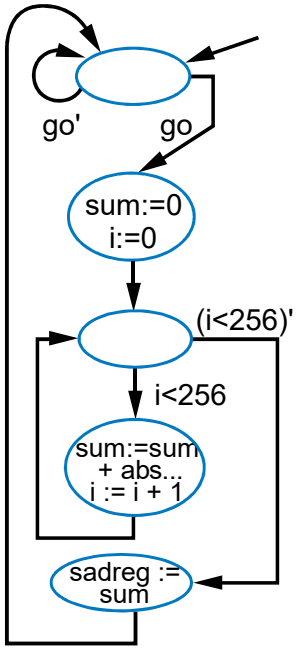
Sources: TSR, Katz, Boriello, Vahid, Perkowski

FSM

```
Inputs byte A[256],B[256]
       bit go
Output int sad
main()
{
  uint sum, short uint i
  while (1) {

      while (!go)
      sum = 0
      i = 0

      while (i < 256) {
         sum = sum + abs(A[i] – B[i])
         i = i + 1
      }
      sad = sum
  }
}
```

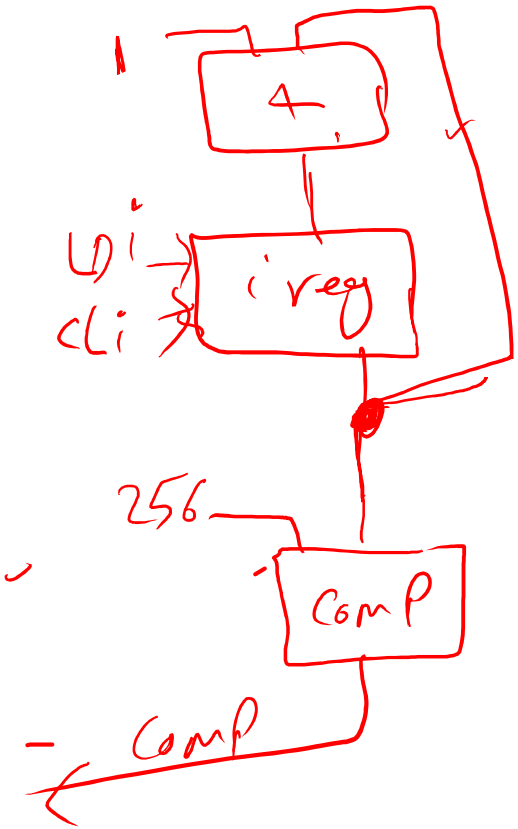


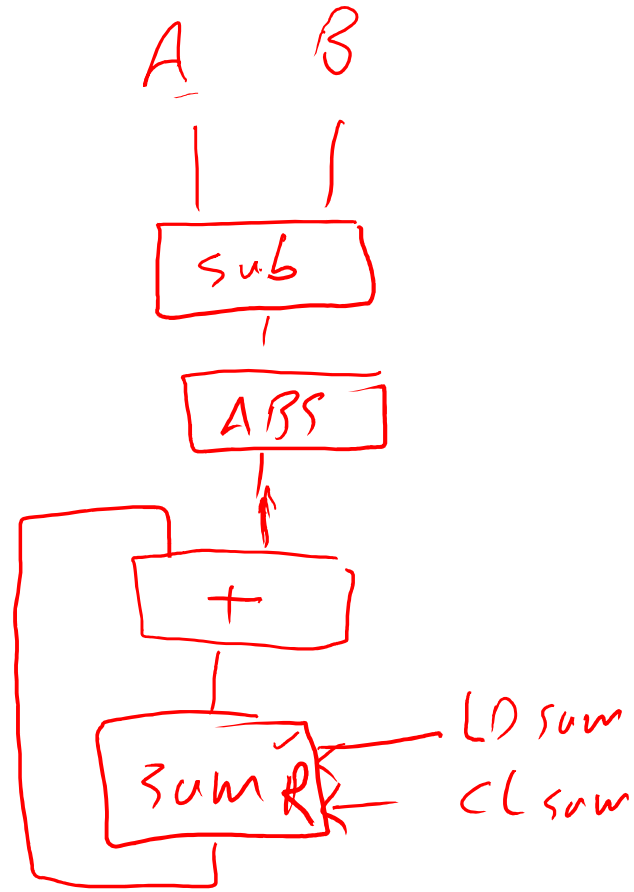
Sources: TSR, Katz, Boriello, Vahid, Perkowski

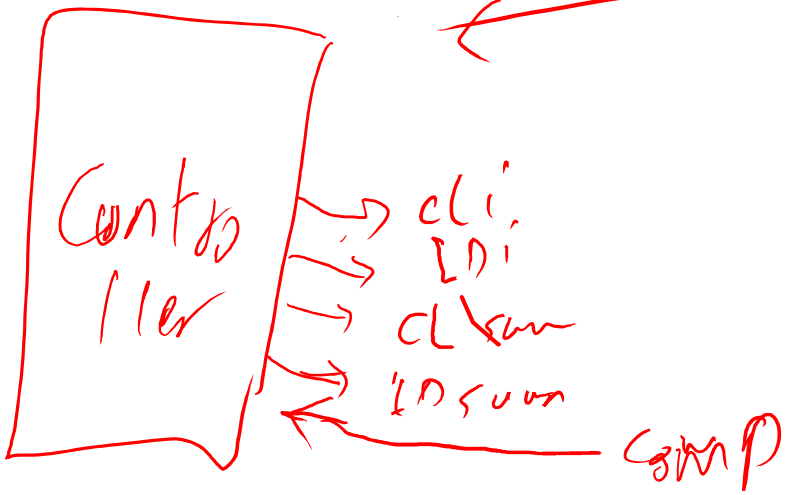
```
Inputs: byte A[256],B[256]
          bit go;
Output: int sad
main()
{
   uint sum; short uint i;
   while (1) {

         while (!go);
         sum = 0;
         i = 0;

         while (i < 256) {
            sum = sum + abs(A[i] – B[i]);
            i = i + 1;
         }
      }  sad = sum;
}
```



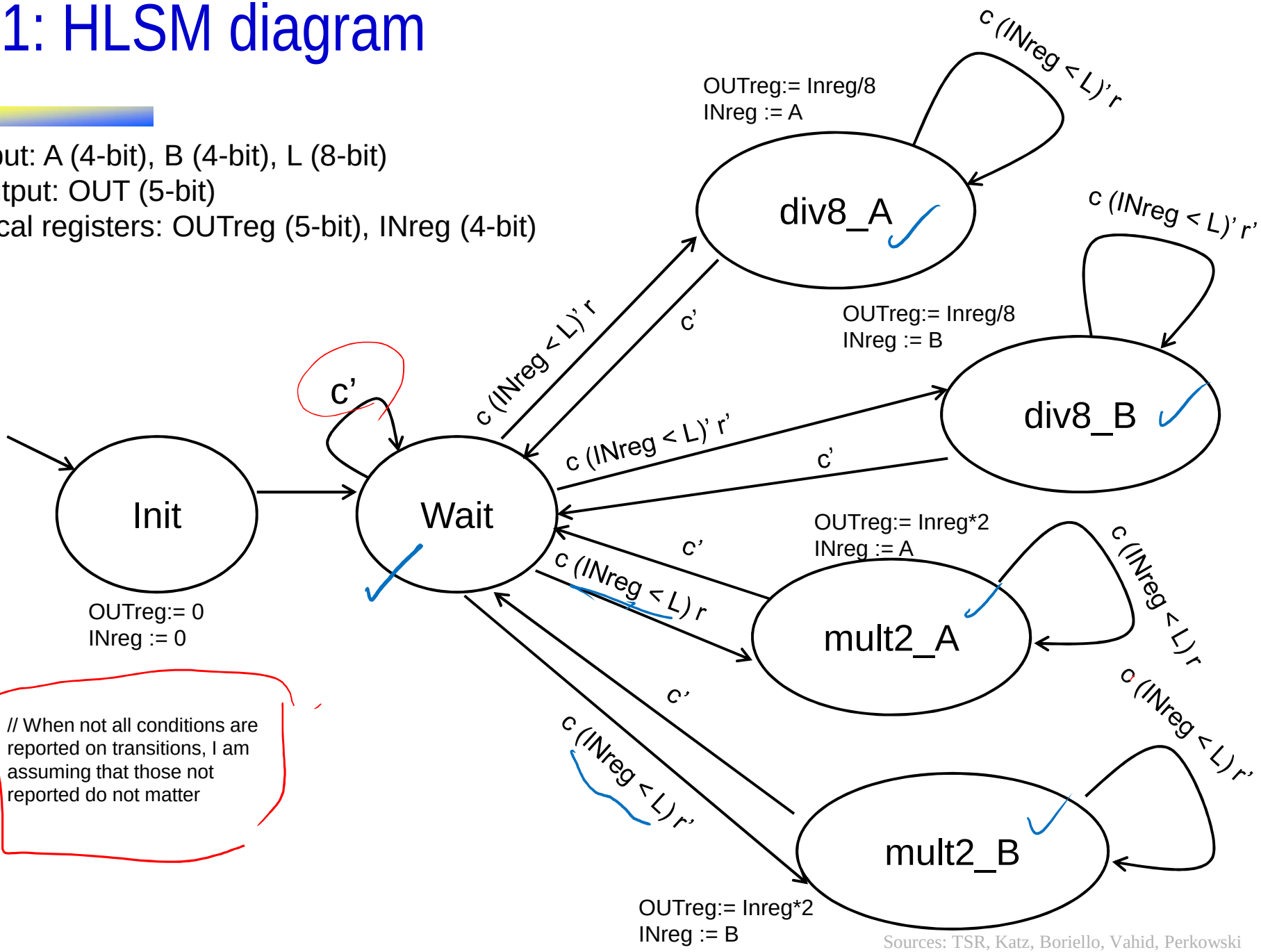




Sources: TSR, Katz, Boriello, Vahid, Perkowski

1: HLSM diagram

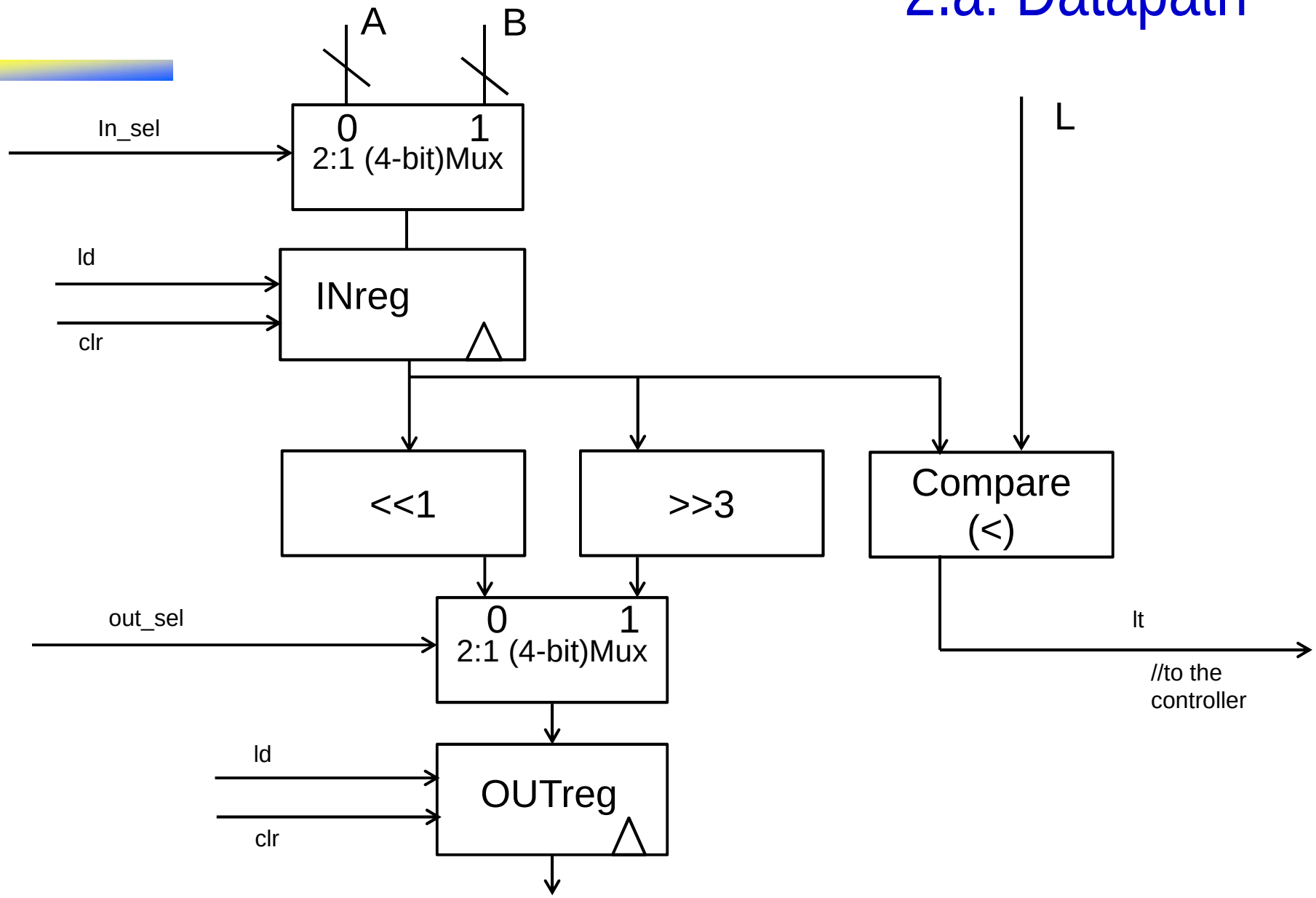
Input: A (4-bit), B (4-bit), L (8-bit)
Output: OUT (5-bit)
Local registers: OUTreg (5-bit), INreg (4-bit)



// When not all conditions are reported on transitions, I am assuming that those not reported do not matter

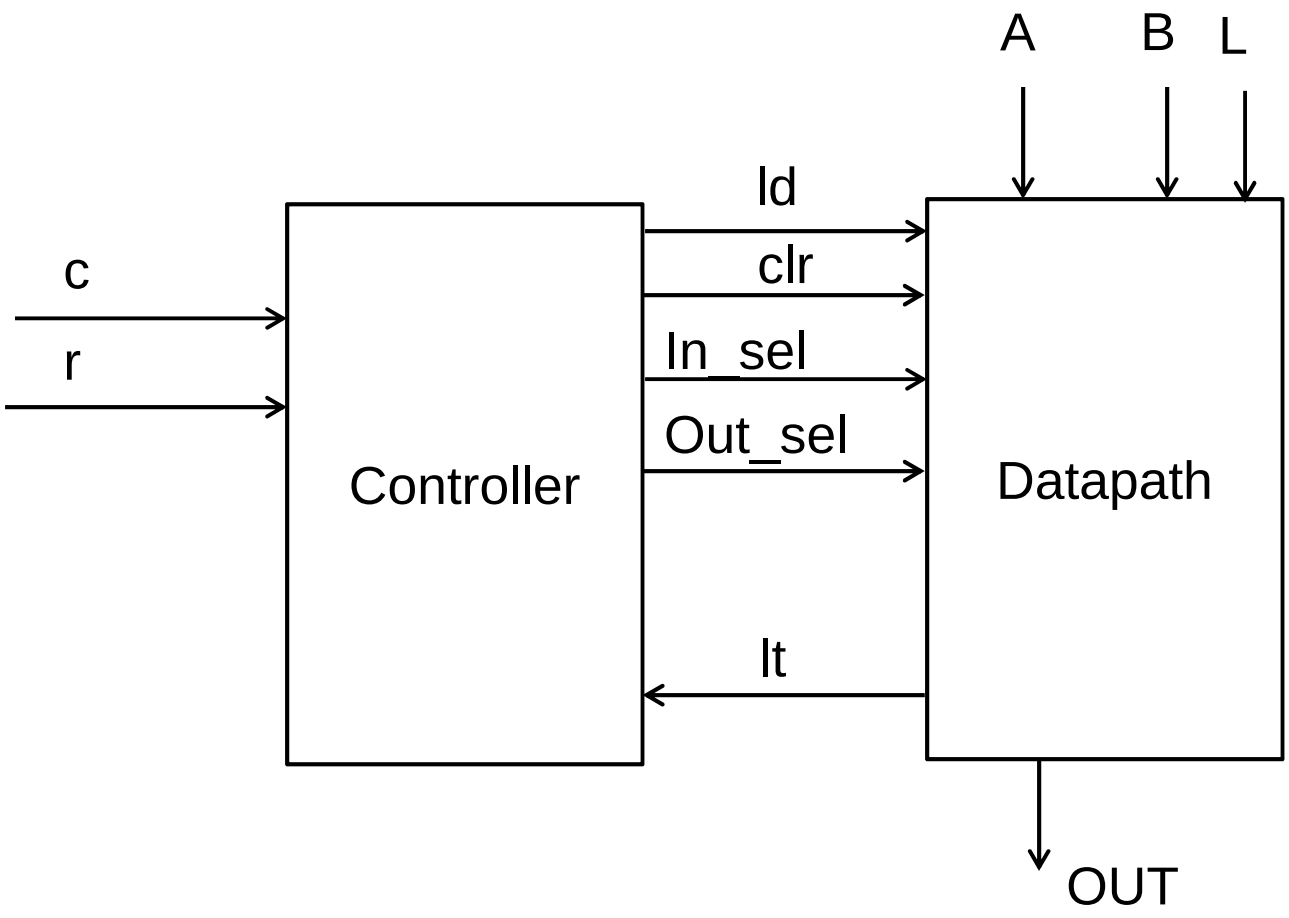
Sources: TSR, Katz, Boriello, Vahid, Perkowski

2.a: Datapath



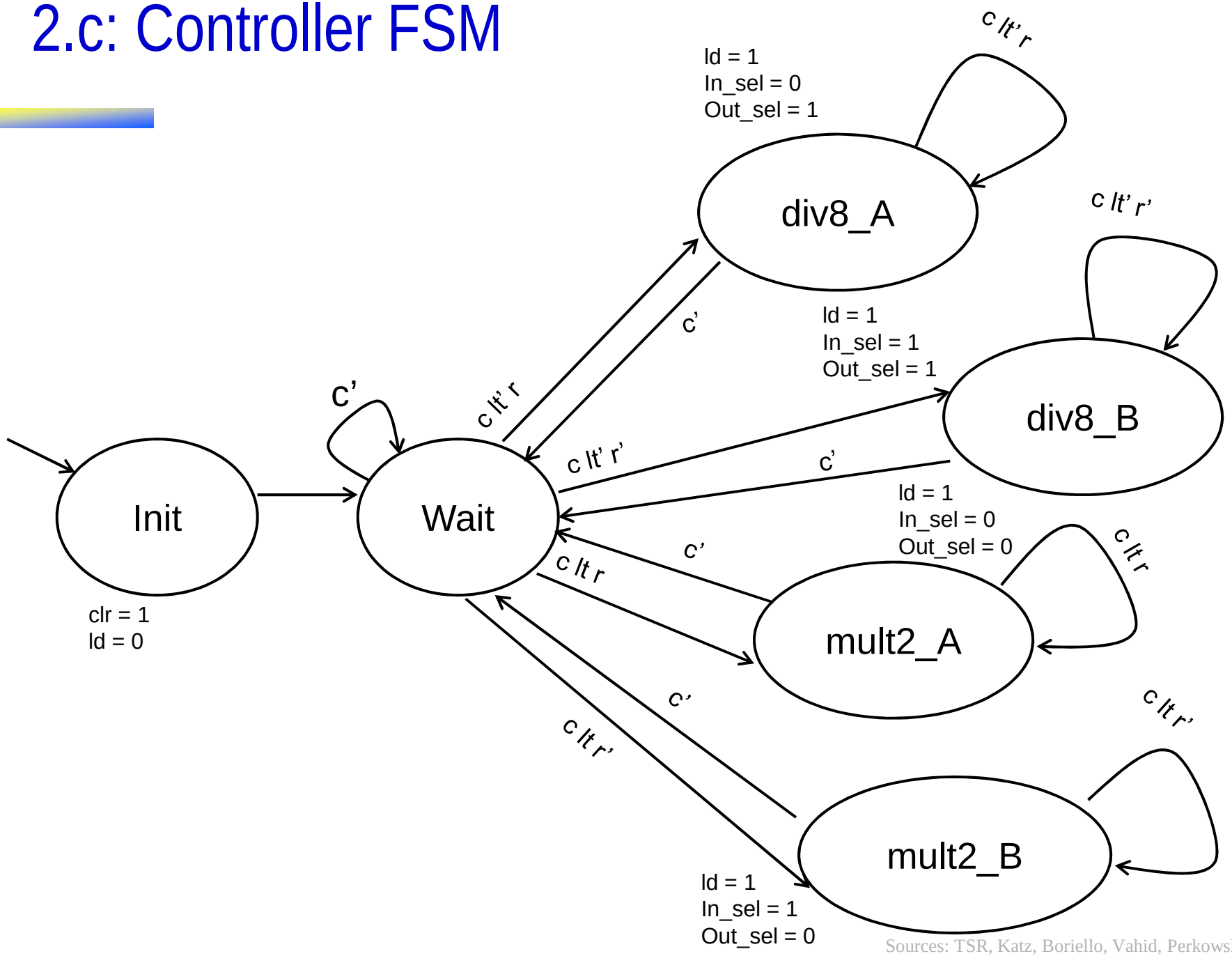
Sources: TSR, Katz, Boriello, Vahid, Perkowski

2.b: Connect datapath and controller



Sources: TSR, Katz, Boriello, Vahid, Perkowski

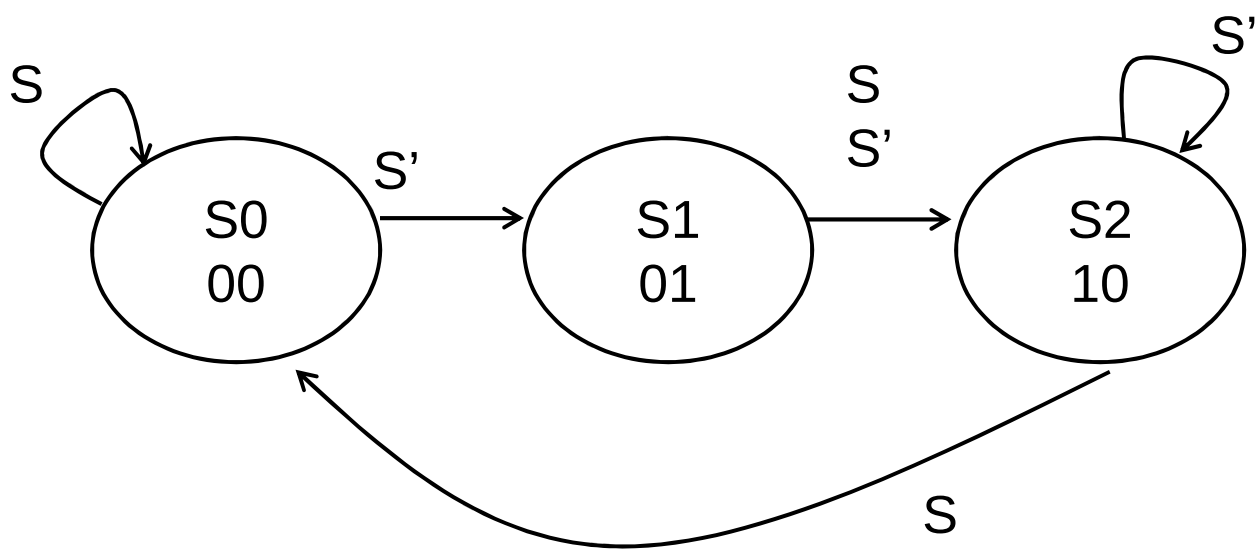
2.c: Controller FSM



Sources: TSR, Katz, Boriello, Vahid, Perkowski

FSM design 1

A FSM has a 1-bit input S and a 2-bit output A[1:0].
Initially the output is 00, and it stays at 00 as long as S is 1. If S is 0, then the output switches to 01 for only one clock cycle, before becoming 10. At this point, if S is 1, then the output will be 00. If S is 0 instead, the output would remain 10.



State and excitation table

State table with assignment

	S=0	S=1
00	01	00
01	10	10
10	10	00

Excitation table

Q1Q0s	D1	D0	A1	A0
000	0	1	0	0
001	0	0	0	0
010	1	0	0	1
011	1	0	0	1
100	1	0	1	0
101	0	0	1	0
110	X	x	x	X
111	x	X	x	x

State assignment:
S0: 00
S1: 01
S2: 10

Kmaps and equations

Excitation table

Q1Q0s	D1	D0	A1	A0
000	0	1	0	0
001	0	0	0	0
010	1	0	0	1
011	1	0	0	1
100	1	0	1	0
101	0	0	1	0
110	X	x	x	X
111	x	X	x	x

s\Q1Q0	00	01	11	10
0	0	1	X	1
1	0	1	X	0

s\Q1Q0	00	01	11	10
0	1	0	X	0
1	0	0	x	0

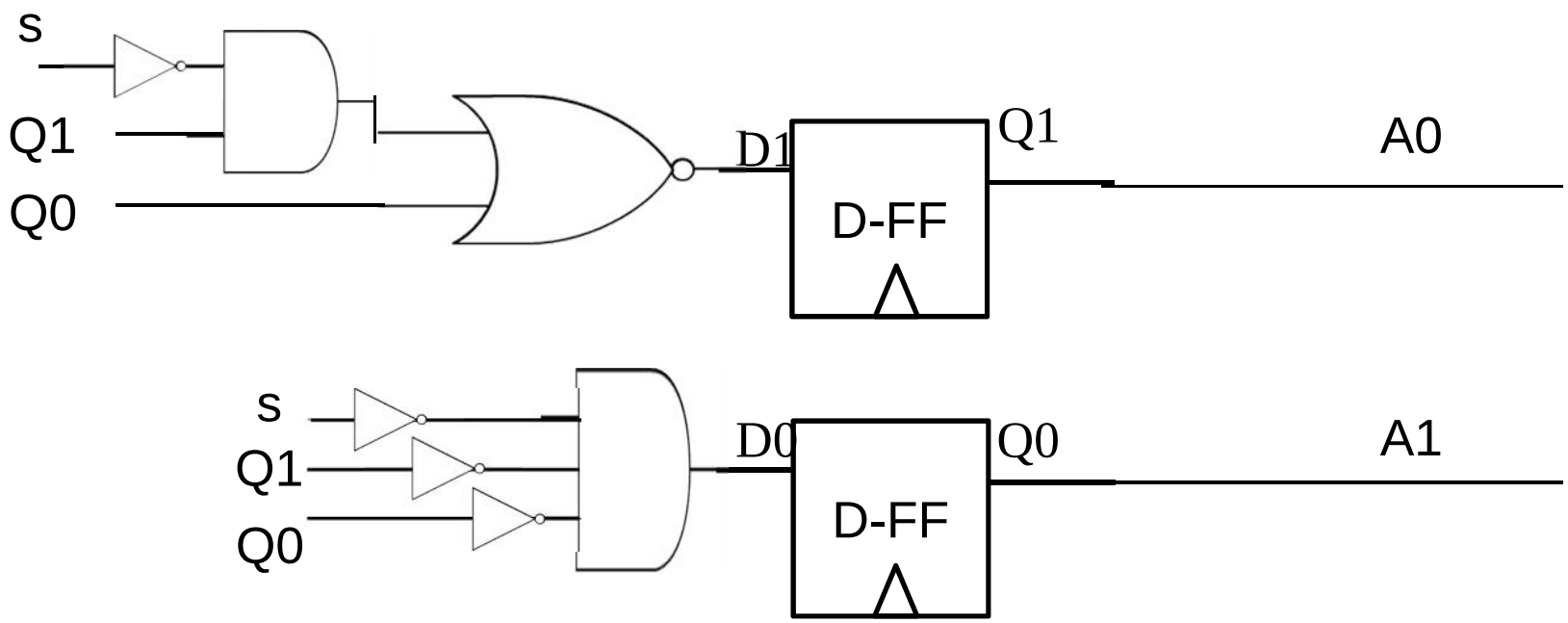
s\Q1Q0	00	01	11	10
0	0	0	x	1
1	0	0	X	1

s\Q1Q0	00	01	11	10
0	0	1	X	0
1	0	1	X	0

D1 = Q0 + s'Q1
D0 = s'Q1'Q0'
A1 = Q1
A0 = Q0

Sources: TSR, Katz, Boriello, Vahid, Perkowski

Circuit

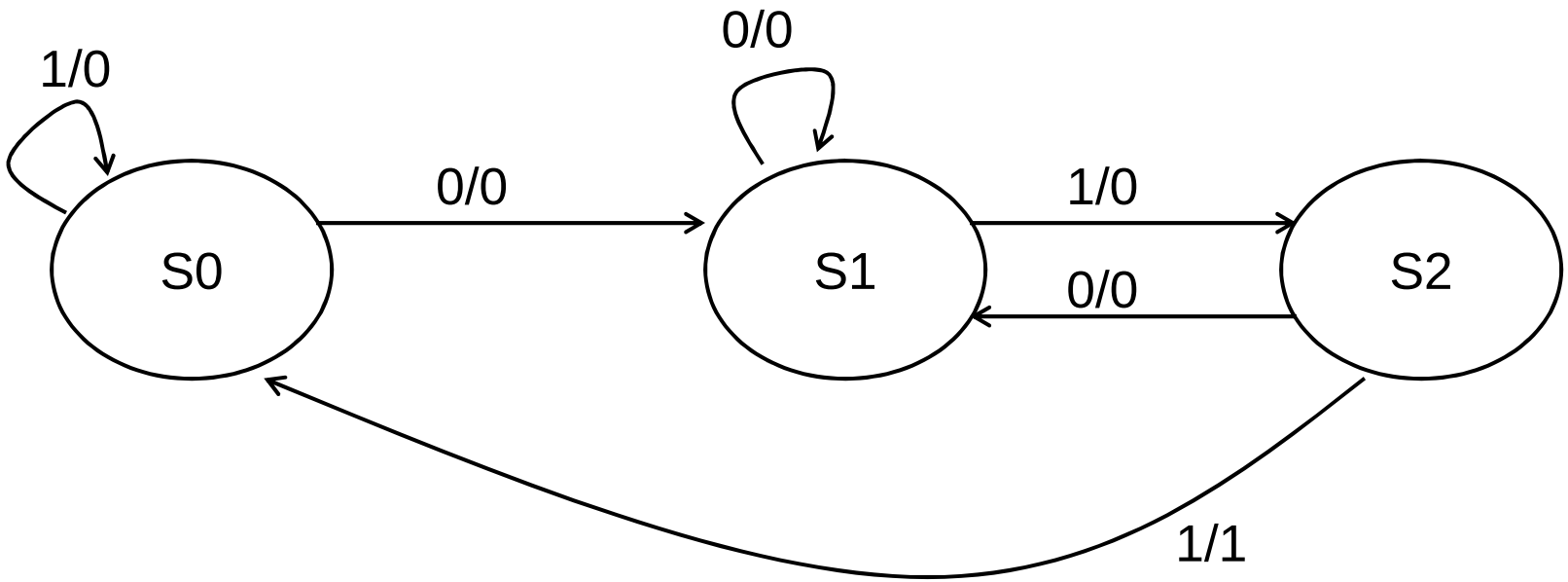


D1 = Q0 + s'Q1
D0 = s'Q1'Q0'
A1 = Q1
A0 = Q0

Sources: TSR, Katz, Boriello, Vahid, Perkowski

FSM design 2

Design a Mealy FSM that recognizes the sequence “011”. The machine has a 1-bit input S and a 1-bit output B. When the sequence is recognized, the output switches to 1 for one clock cycle.



State and excitation table

State table with assignment

	S=0	S=1
00	01, 0	00, 0
01	01, 0	10, 0
10	01, 0	00, 1

Excitation table

Q1Q0S	D1	D0	B
000	0	1	0
001	0	0	0
010	0	1	0
011	1	0	0
100	0	1	0
101	0	0	1
110	x	X	X
111	x	x	x

State assignments:
S0: 00
S1: 01
S2: 10

Kmaps and equations

Excitation table

Q1Q0S	D1	D0	B
000	0	1	0
001	0	0	0
010	0	1	0
011	1	0	0
100	0	1	0
101	0	0	1
110	x	X	X
111	x	x	x

S\Q1Q0	00	01	11	10
0	0	0	X	0
1	0	1	x	0

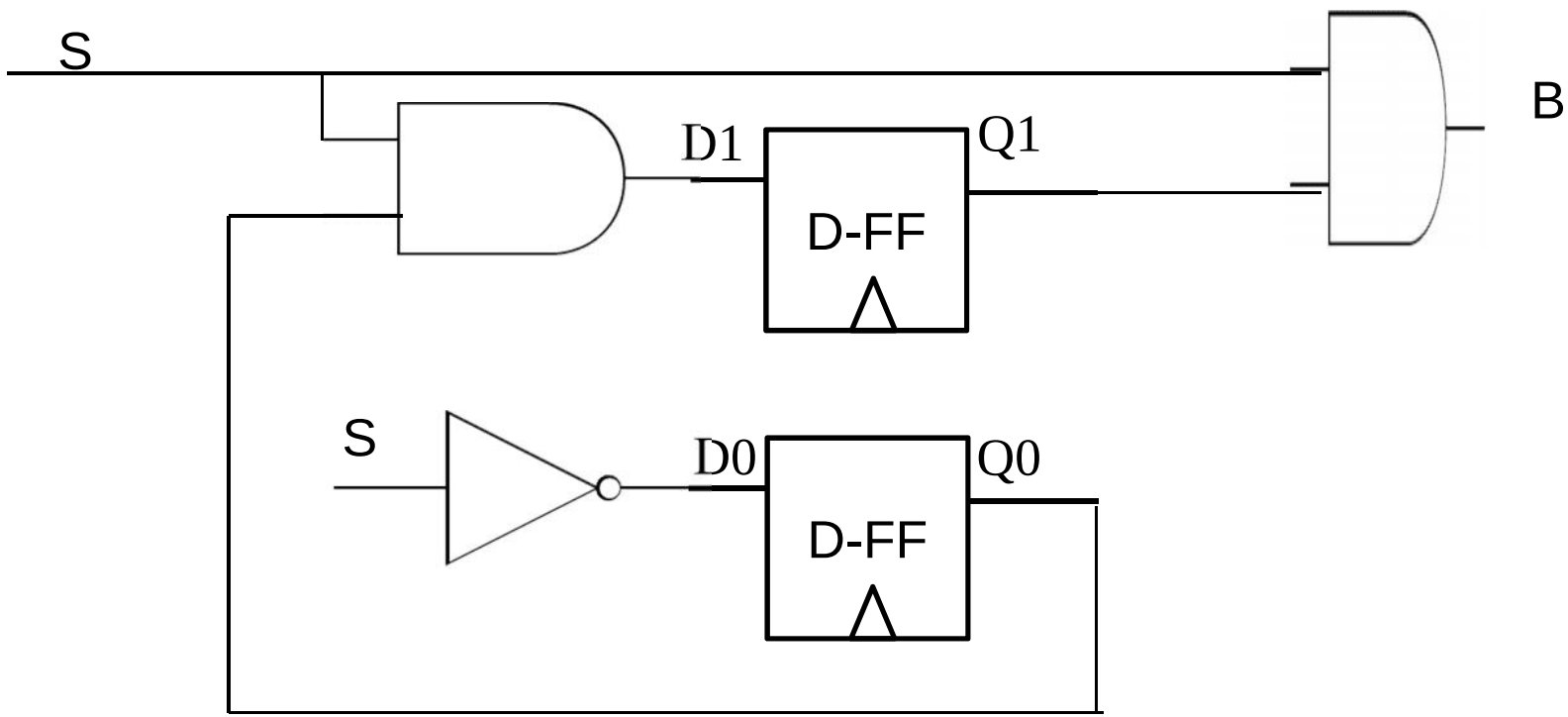
S\Q1Q0	00	01	11	10
0	1	1	X	1
1	0	0	x	0

S\Q1Q0	00	01	11	10
0	0	0	X	0
1	0	0	x	1

D1 = S Q0
D0 = S’
B = S Q1

Sources: TSR, Katz, Boriello, Vahid, Perkowski

Circuit



D1 = S Q0
D0 = S'
B = S Q1

Sources: TSR, Katz, Boriello, Vahid, Perkowski

FSM design 3

Starting from the following characteristic equations for a FSM, derive the FSM diagram

D1 = xQ1'Q0

D0 = x'Q1 + Q0

Y = Q1 + Q0'

Is this a Mealy or a **Moore** machine? (Moore, because the output does not depend on the input x)

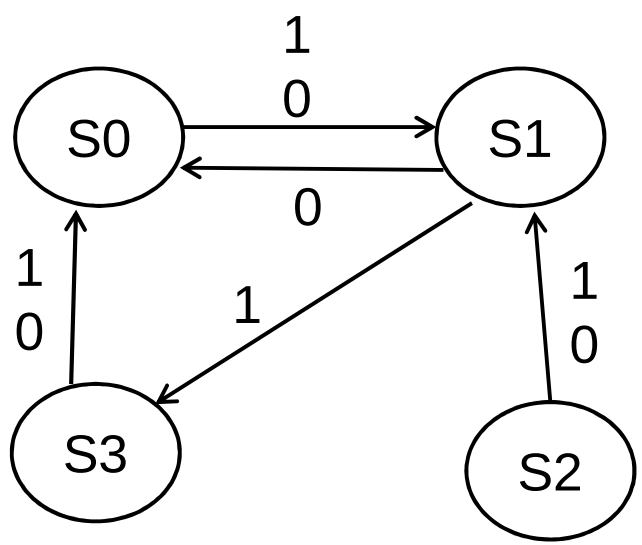
Excitation table

Q1Q0x	D1	D0	Y
000	0	1	1
001	0	1	1
010	0	0	0
011	1	1	0
100	0	1	1
101	0	1	1
110	0	0	1
111	0	0	1

State table with assignment

	x=0	x=1
00	01	01
01	00	11
10	01	01
11	00	00

S0: 00
S1: 01
S2: 10
S3: 11



Sources: TSR, Katz, Boriello, Vahid, Perkowski

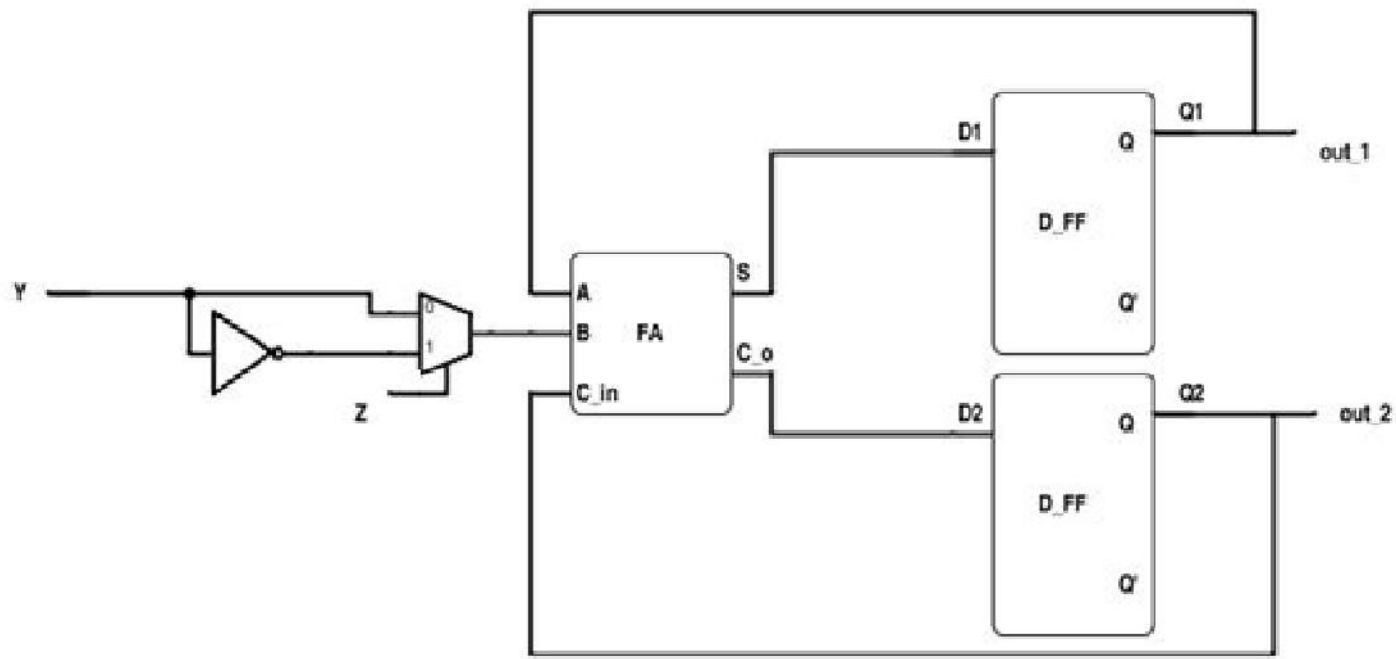
For the given circuit which has two 1 bit inputs (Y, Z) and two outputs (out_1, out_2):

a. Write the state table.

b. Draw the state diagram.

c. Describe the functionality of this FSM.

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

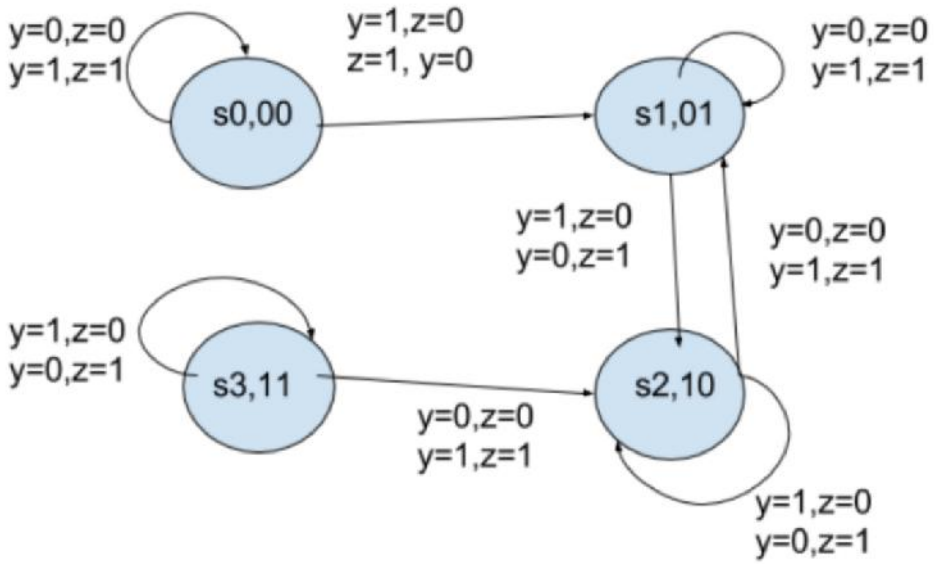


Sources: TSR, Katz, Boriello, Vahid, Perkowski

Y	Z	Q1(t)	Q2(t)	Q1(t+1)	Q2(t+1)	out_1(t)	out_2(t)
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	1
0	0	1	0	1	0	1	0
0	0	1	1	0	1	1	1
0	1	0	0	1	0	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	0	1	0	1
1	0	1	0	0	1	1	0
1	0	1	1	1	1	1	1
1	1	0	0	0	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	0
1	1	1	1	0	1	1	1

Sources: TSR, Katz, Boriello, Vahid, Perkowski

Y	Z	Current State	Next State	out_1	out_2
0	0	S0	S0	0	0
0	0	S1	S1	0	1
0	0	S2	S1	1	0
0	0	S3	S2	1	1
0	1	S0	S1	0	0
0	1	S1	S2	0	1
0	1	S2	S2	0	1
0	1	S3	S3	1	1
1	0	S0	S1	0	0
1	0	S1	S2	0	1
1	0	S2	S2	0	1
1	0	S3	S3	1	1
1	1	S0	S0	0	0
1	1	S1	S1	0	1
1	1	S2	S1	1	0
1	1	S3	S2	1	1



Sources: TSR, Katz, Boriello, Vahid, Perkowski

CSE140 Summary

- Transistors and CMOS technology
- Boolean algebra
- Logic functions, truth tables, circuit representations with basic gates
- 2-level logic minimizations using Kmaps
- Multiplexers and decoders
- ALU components: adders, subtractors (2's complement representation), arithmetic shifters, multiplier, dividers.
- ALU design
- SR latch, level-sensitive SR latch, D-latch, D-FlipFlop
- Registers, counters and shift registers
- **Finite State Machines: Mealy and Moore**
- **Timing Constraints**
- **RTL design**

Sources: TSR, Katz, Boriello, Vahid, Perkowski

CSE140 Summary

We can build computers out of one of the simplest mathematical theories (the Boolean algebra) using the simplest numbering system (zeros and ones)

We can do that because we have MOS transistors, tiny electronic devices that are built to provide an extremely simple behavior: being either ON or OFF.

Good Luck for the Final Exam !

Sources: TSR, Katz, Boriello, Vahid, Perkowski