Morgan Schulte

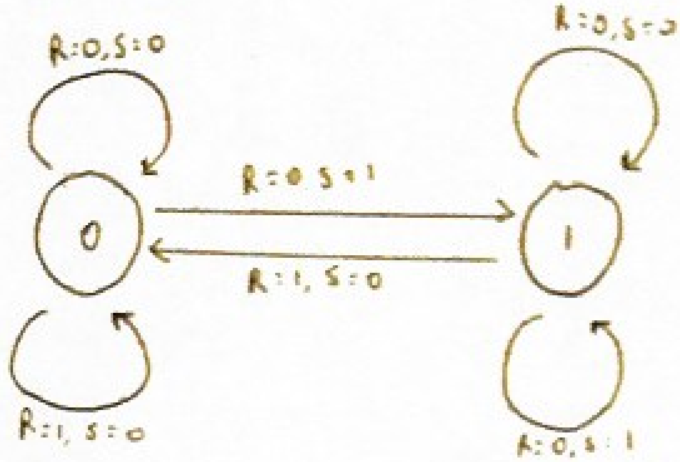
Digital Electronics Supervision 3

1. A synchronous finite state machine has many possible states and has a logical way of transitioning from one state to the next. Such transitions occur on the rising edge of a clock pulse.

- Moore machines are synchronous FSMs in which the next state is determined purely by the current state
- Mealy machines are synchronous FSMs in which the next state depends on both the current state and some input signal/s

2. SR Flip Flop (NOTE: I made a distinction here between the state Z and the output Q because they don't quite match up).

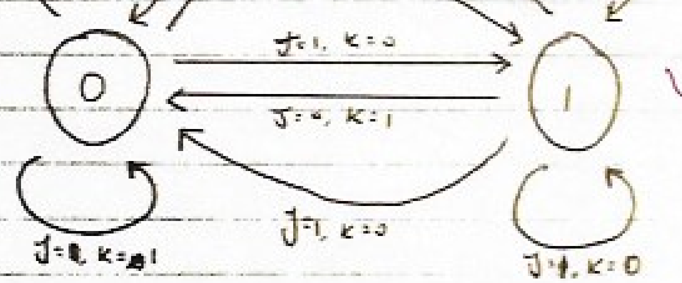
S	R	Z_n	Z_{n+1}	Q	$\overline{Q}$
0	0	0	0	0	1
0	1	0	0	0	1
1	0	0	1	1	0
1	1	0	invalid	1	1
0	0	1	1	1	0
0	1	1	0	0	1
1	0	1	1	1	0
1	1	1	invalid	1	1



Z_n	Z_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

JK Flip Flop

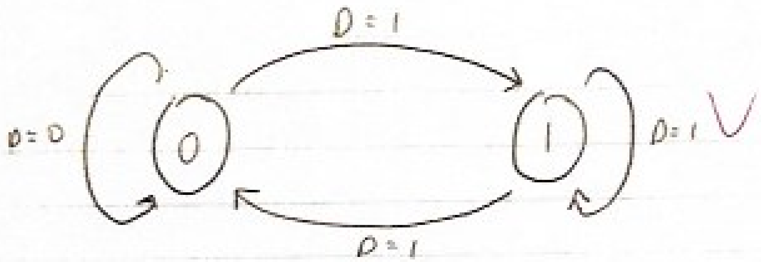
J	K	Z_n	Z_{n+1}	Q	$\bar{Q}$
0	0	0	0	0	1
0	1	0	0	0	1
1	0	0	1	1	0
1	1	0	1	1	0
0	0	1	1	1	0
0	1	1	0	0	1
1	0	1	1	1	0
1	1	1	0	0	1



Z_n	Z_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

D Flip Flop

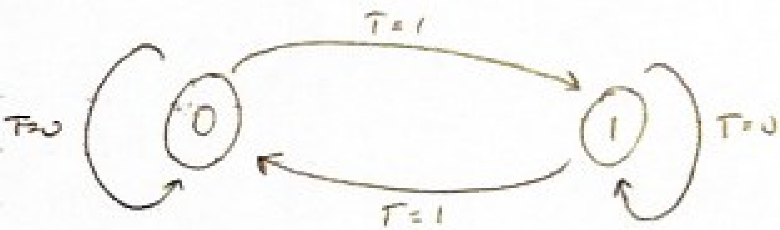
D	Q_n	Q_{n+1}
0	0	0
1	0	1
0	1	0
1	1	1



Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

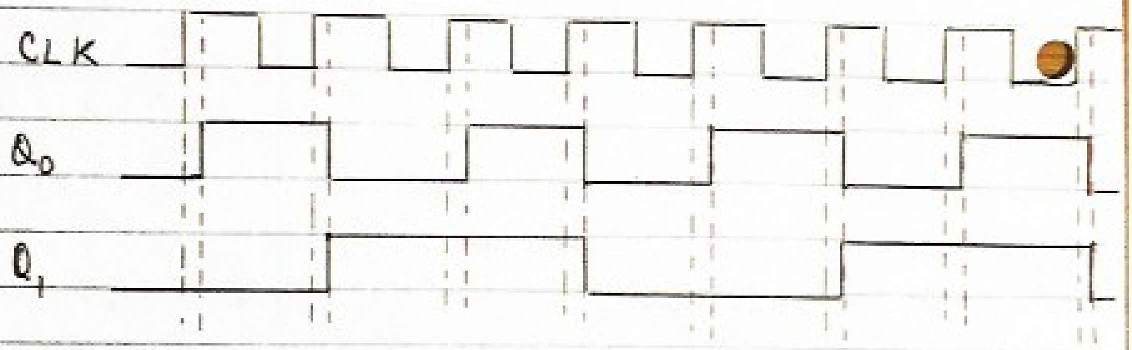
T flip flop

T	Q_n	Q_{n+1}
0	0	0
1	0	1
0	1	1
1	1	0



Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

3a

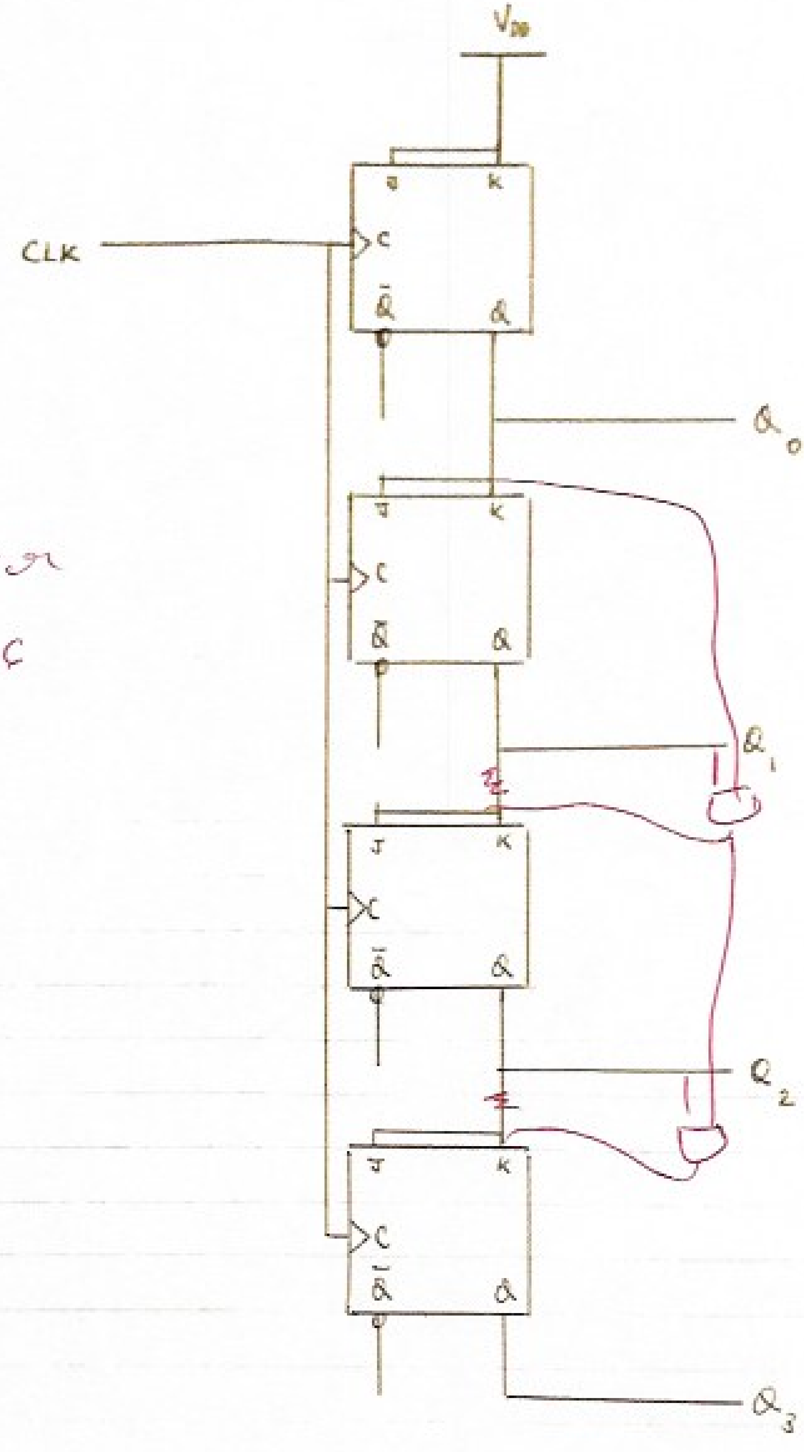


4

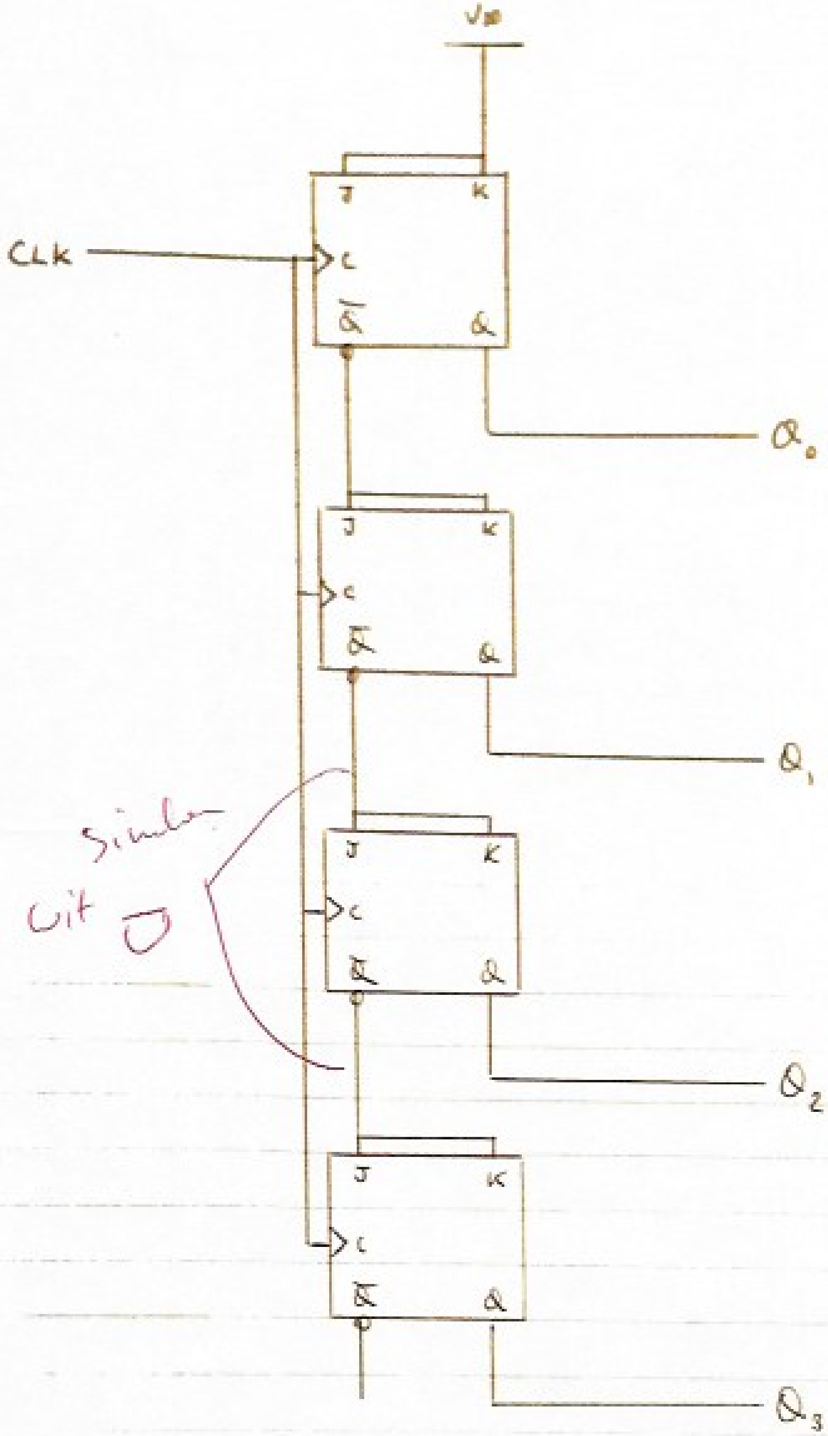
b.

Chron

all 1f



c



F

P	${}_nQ_1$	${}_nQ_0$	${}_{n+1}Q_1$	${}_{n+1}Q_0$	J_1	J_0	K_1	K_0
0	0	0	0	0	0	0	x	x
1	0	0	0	1	0	1	x	x
0	0	1	1	0	1	x	x	1
1	0	1	0	1	0	x	x	0
0	1	0	1	0	x	0	0	x
1	1	0	1	1	x	1	0	x
0	1	1	1	1	x	x	0	0
1	1	1	0	0	x	x	1	1

✓

Assuming Q_0 and Q_1 are both outputs of J-K flip flops, they ~~xxx~~ must have associated values J_0 and K_0, and J_1 and K_1 respectively these values can be filled ~~into~~ in to the above table from the excitation table of a J-K flip flop.

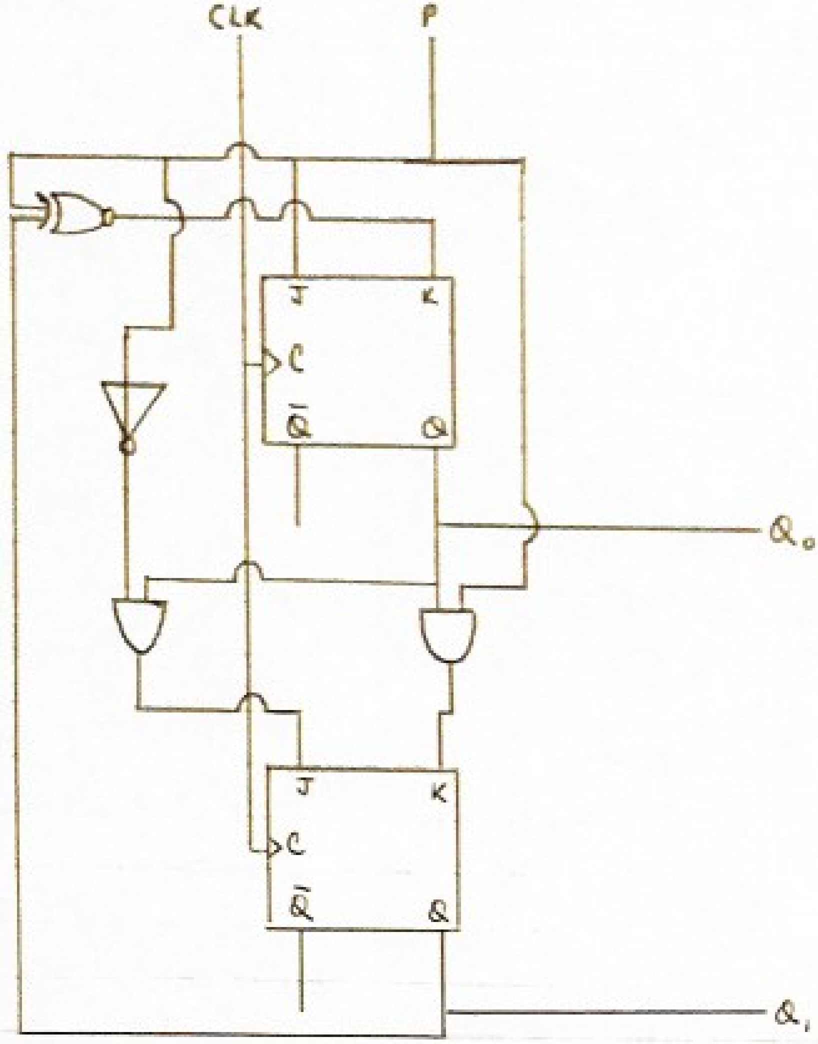
${}_nQ$	${}_{n+1}Q$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

The J_1 column in the top table matches $\bar{P}\cdot{}_nQ_0$

J_0 matches P

K_1 matches $P\cdot{}_nQ_0$ ✓

K_0 matches $\overline{P\oplus{}_nQ_1}$



✓

5.

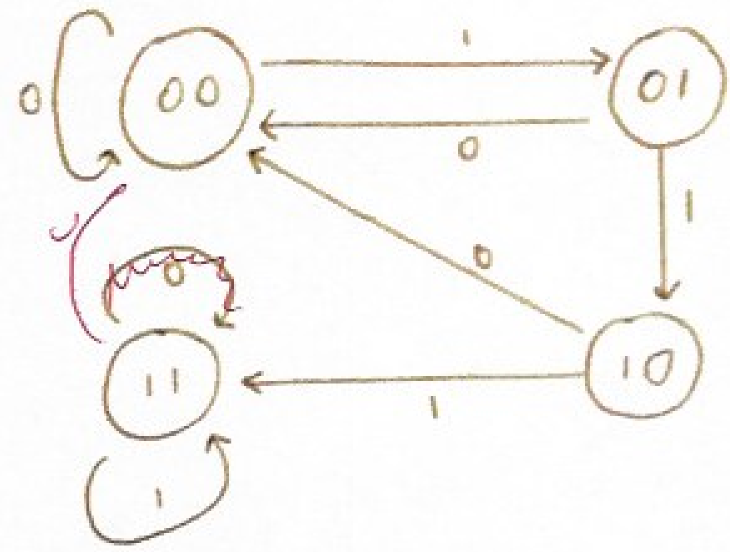
Clock Cycle	Q_1	Q_0
0	0	0
1	0	1
2	1	0
3	0	0

It counts from 0 to 2 (inclusive) in binary and then resets to 0

✓

8

6a.



If the system ends in state 11, then there are at least 3 consecutive 1s. Otherwise, there are not.

b. ~~Mealy~~ Moore

c. Since I don't know whether Q_1 and Q_0 will be the outputs of J-K Flip Flops or D Flip Flops, I will include the corresponding values of J, K, and D with each transition, under the assumption

Yep that we must ~~eith~~ either use the J and the K,

✓ or the D, whichever ends up being simpler.

P	${}_nQ_1$	${}_nQ_0$	${}_{n+1}Q_1$	${}_{n+1}Q_0$	J_1	J_0	K_1	K_0	D_1	D_0
0	0	0	0	0	0	0	X	X	0	0
1	0	0	0	1	0	1	X	X	0	1
0	0	1	0	0	0	X	X	1	0	0
1	0	1	1	0	1	X	X	1	1	0
0	1	0	0	0	X	0	1	X	0	0
1	1	0	1	1	X	1	0	X	1	1
0	1	1	1	1	X	X	0	0	1	1
1	1	1	1	1	X	X	0	0	1	1

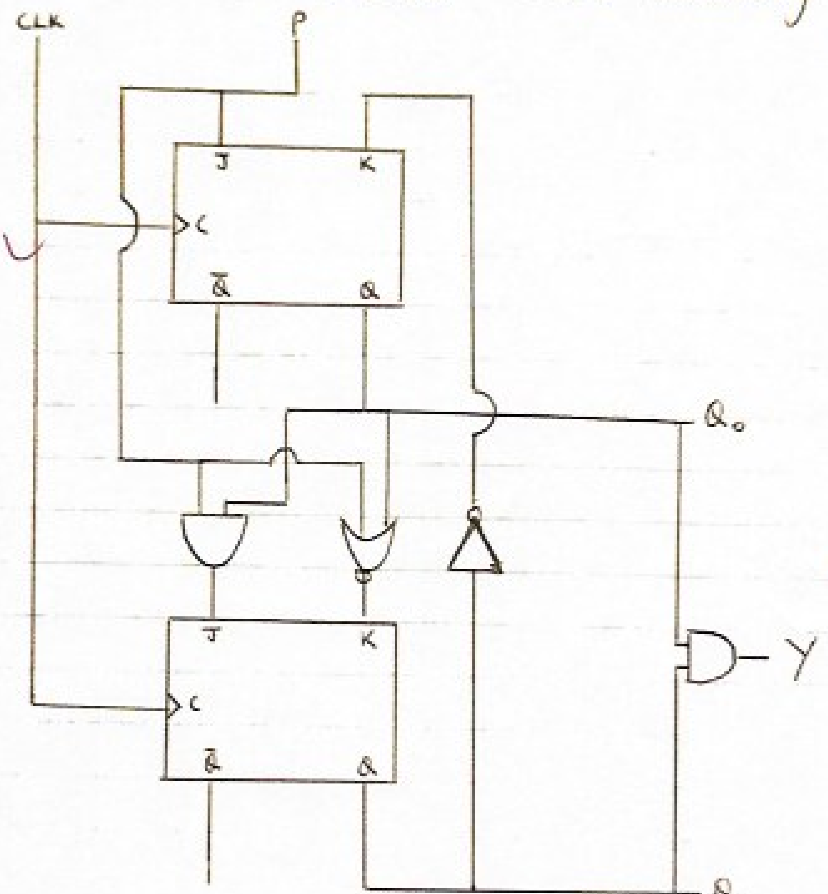
8

The column J_1 matches $P \cdot {}_nQ_0$ ✓
J_0 matches $\overline{P}$ ✓
K_1 matches $\overline{P + {}_nQ_0}$ $\overline{P} + \overline{Q_0}$?
K_0 matches ${}_n\overline{Q_1}$ ✓

D_1 and D_0 do not match any simple combination of P, ${}_nQ_1$ and/or ${}_nQ_0$ which I could see so I think it will be simpler to use 2 J-K flip-flops and no D flip flops. Perhaps this wouldn't be the case if I had used one-hot state encoding.

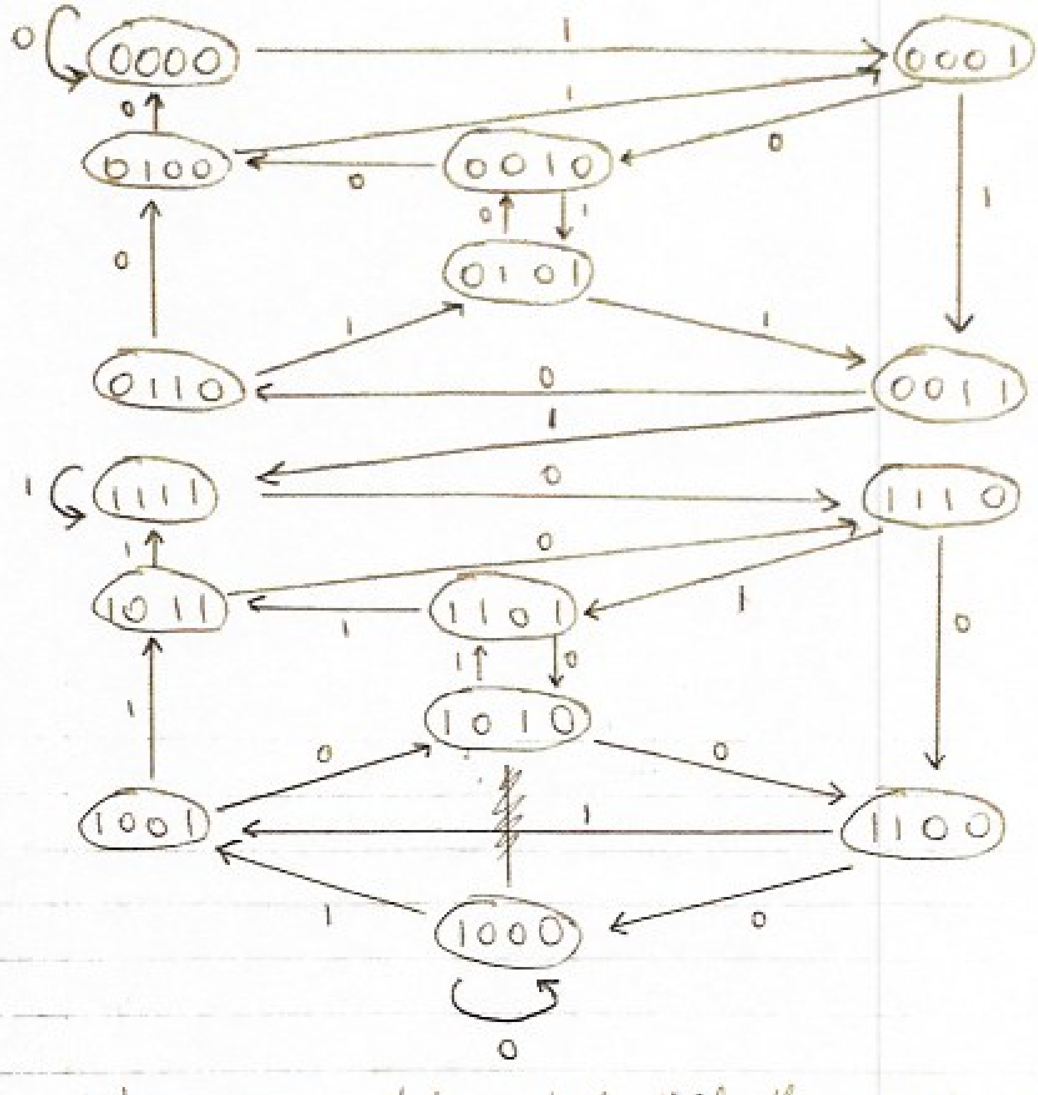
sorry my fault
it should be
a) D ~~only~~ only
b) JK only

✓

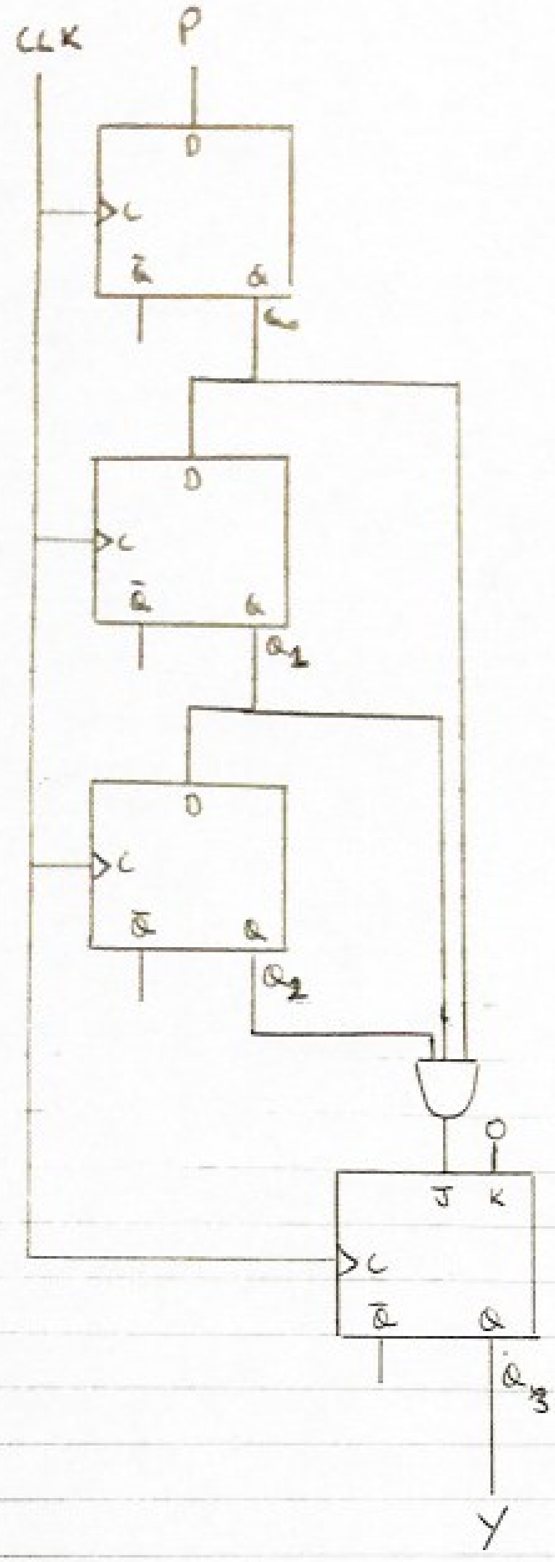


10

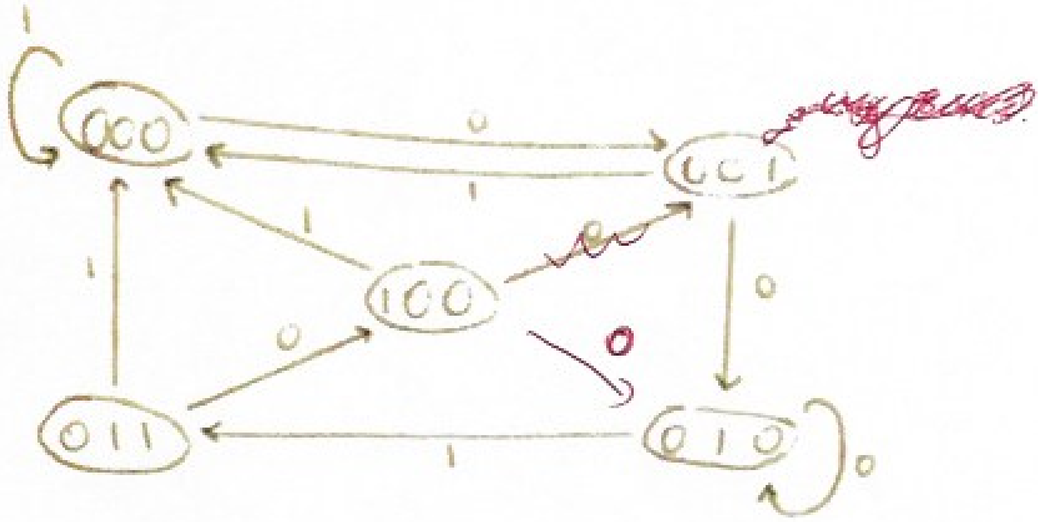
Alternatively a more complicated state diagram does give an easier-to-understand circuit.



where any state in which ~~that~~ the most significant bit is a 1 is a success state.



12

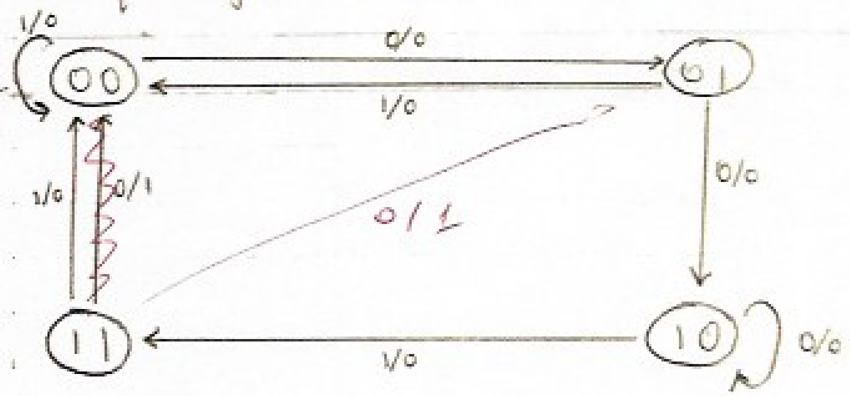
7a. 

Where Q_2 is the output.

b.

Y	$_nQ_2$	$_nQ_1$	$_nQ_0$	$_{n+1}Q_2$	$_{n+1}Q_1$	$_{n+1}Q_0$
0	0	0	0	0	0	1
1	0	0	0	0	0	0
0	0	0	1	0	1	0
1	0	0	1	0	0	0
0	0	1	0	0	1	0
1	0	1	0	0	1	1
0	0	1	1	1	0	0
1	0	1	1	0	0	0
0	1	0	0	0	0	1
1	1	0	0	0	0	0

State 100 is equivalent to state 000 and so is redundant. However, if we remove it, we can no longer take a meaningful output from the machine.

c. 

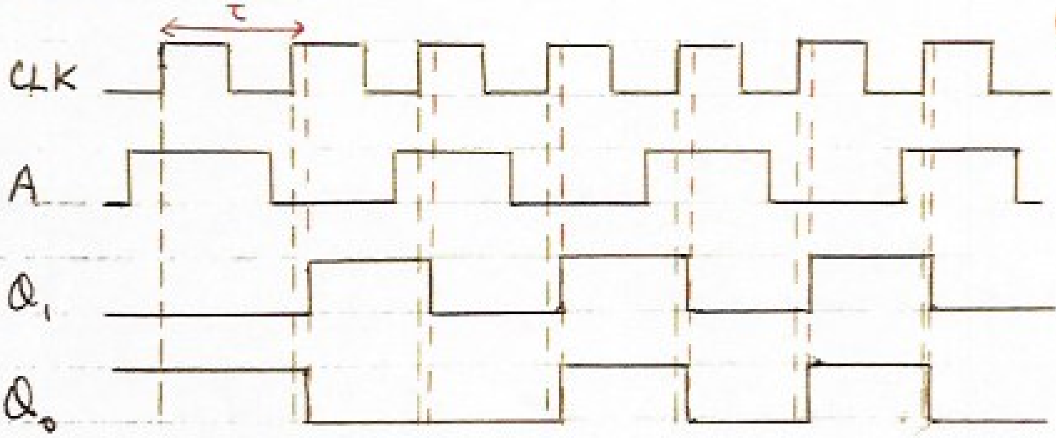
8a.

A	$Q_{n,1}$	$Q_{n,0}$	$Q_{n+1,1}$	$Q_{n+1,0}$	D_1	D_0
0	0	0	0	0	0	0
1	0	0	1	0	1	0
0	0	1	0	0	0	0
1	0	1	1	0	1	0
0	1	0	0	1	0	1
1	1	0	1	1	1	1
0	1	1	0	1	0	1
1	1	1	1	1	1	1

Where D_1 is the required input to Q_1 and D_0 is likewise for Q_0

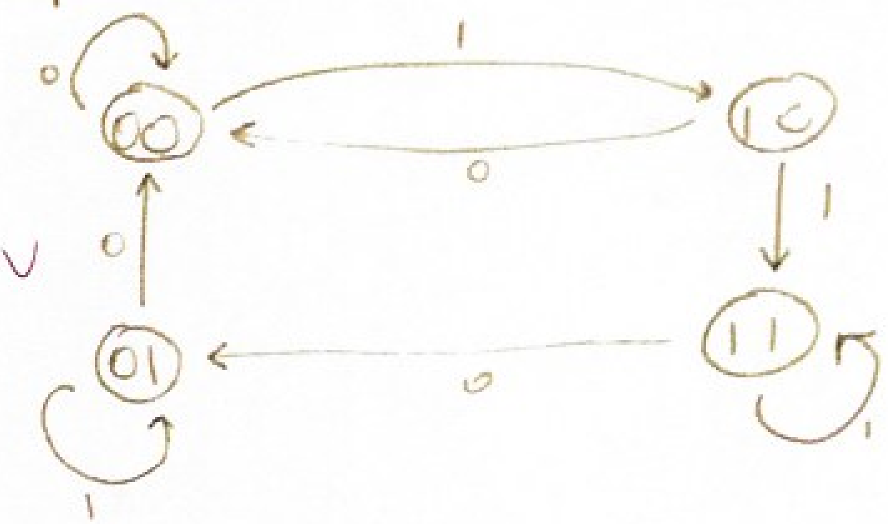
Note that ~~A~~ $D_1 = A$ and $D_0 = Q_{n,1}$
∴ this is a shift register

b. If the clock rate is faster than the transmission delay of the flip flops, this decreases the likelihood of ~~the~~ the dashed transitions, as shown below:



Instead of alternating $01 \to 10 \to 01 \to$
the machine ends up alternating $11 \to 00 \to 11$

c Option 1:



A	${}_mQ_2$	${}_mQ_1$	${}_{m+1}Q_2$	${}_{m+1}Q_1$
0	0	0	0	0
1	0	0	1	0
0	0	1	0	0
1	0	1	0	1
0	1	0	0	0
1	1	0	1	1
0	1	1	0	1
1	1	1	1	1

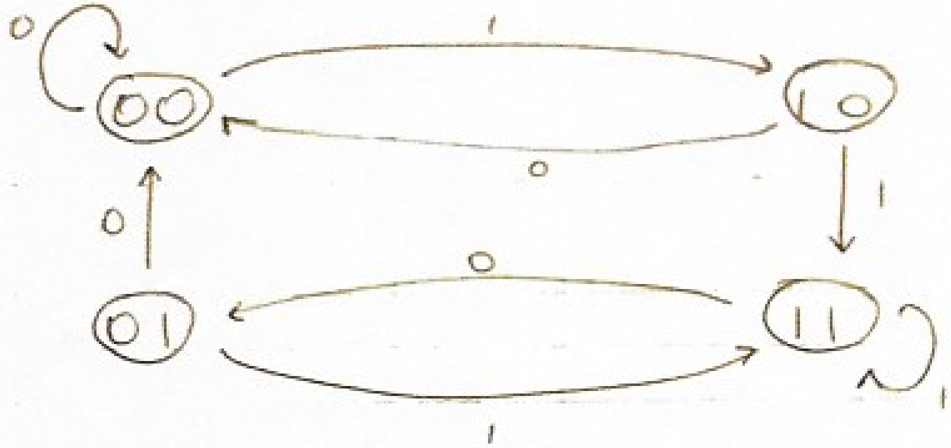
$${}_{m+1}Q_2 = A\left({}_mQ_2 + {}_m\overline{Q}_1\right)$$

1

$${}_{m+1}Q_1 = A\left({}_mQ_2 + {}_mQ_1\right) + {}_mQ_2\,{}_mQ_1$$

15

Option 2

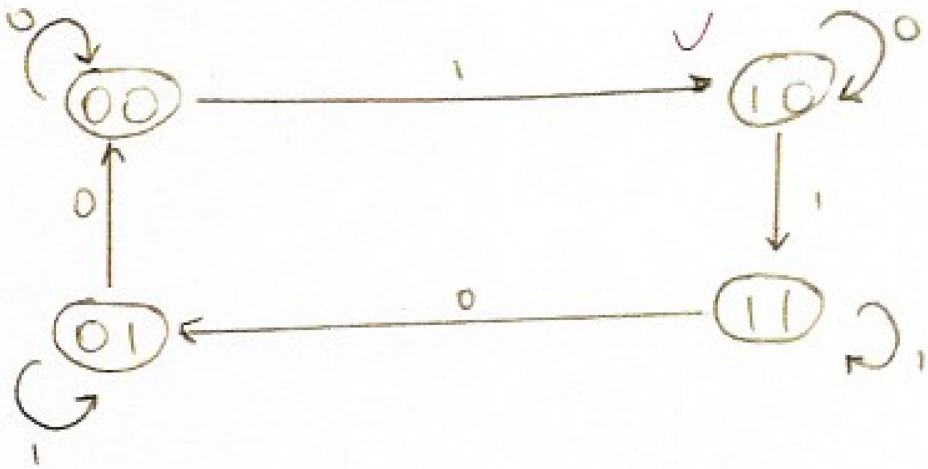


A	$_{m}Q_2$	$_{m}Q_1$	$_{m+1}Q_2$	$_{m+1}Q_1$
0	0	0	0	0
1	0	0	1	0
0	0	1	0	0
1	0	1	1	1
0	1	0	0	0
1	1	0	1	1
0	1	1	0	1
1	1	1	1	1

$$_{m+1}Q_2 = A$$

$$_{m+1}Q_1 = A(_{m}Q_2 + _{m}Q_1) + _{m}Q_2\,_{m}Q_1$$

Option 3

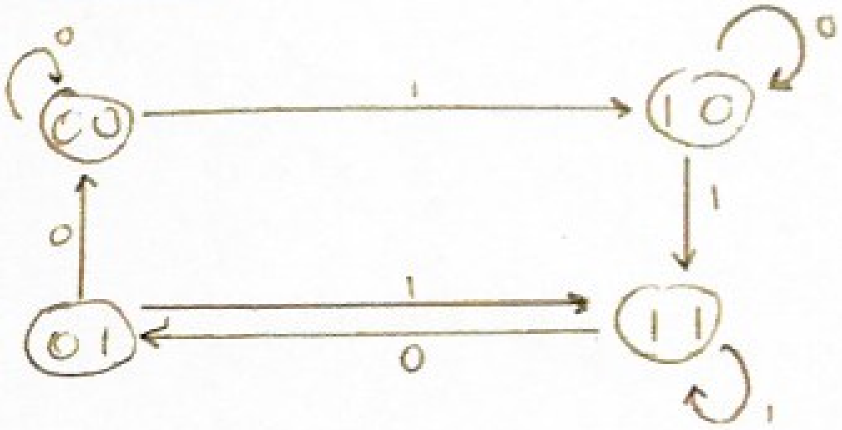


A	$_mQ_2$	$_mQ_1$	$_{m+1}Q_2$	$_{m+1}Q_1$
0	0	0	0	0
1	0	0	1	0
0	0	1	0 ?	0
1	0	1	0	1
0	1	0	0	0
1	1	0	1	1
0	1	1	0	0
1	1	1	0	1

$$_{m+1}Q_2 = \,_m\bar{Q}_2(A + \,_mQ_1) + \bar{A}\,_mQ_2\,_mQ_1$$?

$$_{m+1}Q_1 = \,_mQ_2(A + \,_m\bar{Q}_1) + A\,_m\bar{Q}_2\,_mQ_1$$

Option 4



A	$_mQ_2$	$_mQ_1$	$_{m+1}Q_2$	$_{m+1}Q_1$
0	0	0	0	0
1	0	0	1	0
0	0	1	0	0
1	0	1	1	1
0	1	0	1	0
1	1	0	1	1
0	1	1	0	1
1	1	1	1	1

V

$$_{m+1}Q_2 = \overline{_mQ_2}\,A + {_mQ_2}\,(A + \overline{_mQ_1})^{?} \qquad A + Q_2\overline{Q_1}$$

$$_{m+1}Q_1 = \overline{_mQ_2}\,A\,{_mQ_1} + {_mQ_2}\,(A + {_mQ_1}) \qquad AQ_1 + AQ_2 + Q_2Q_1$$

18