CSSE2010 / CSSE7201 – Introduction to Computer Systems
Answers to Exercises – Week Four
Shift Registers and State Machines

Answers

Some of the questions below are taken from or based on questions in Tanenbaum, Structured Computer Organisation, 5th edition.

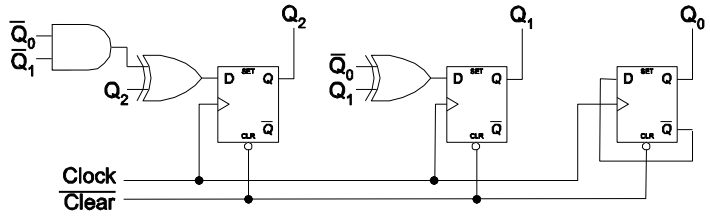
1. Draw the logic diagram for a 3-bit synchronous counter which counts backwards (111→110→101→100→011→010→001→000→111→…). Hint: Start by writing down a truth table of current value and next value as done in the lecture for the 2-bit synchronous counter.

Current Value			Next Value		
Q2	Q1	Q0	D2	D1	D0
0	0	0	1	1	1
0	0	1	0	0	0
0	1	0	0	0	1
0	1	1	0	1	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

$$D0 = \overline{Q0}$$
$$D1 = \overline{Q0}.\overline{Q1} + Q0.Q1 = Q1 \oplus \overline{Q0}$$
$$D2 = \overline{Q0}.\overline{Q1}.\overline{Q2} + (Q0 + Q1).Q2 = Q2 \oplus (\overline{Q0}.\overline{Q1})$$

(We don't expect you to be able to simplify expressions like these. Leaving the answer as sum-of-products form is fine. There are many equally correct expressions and logic diagrams.)

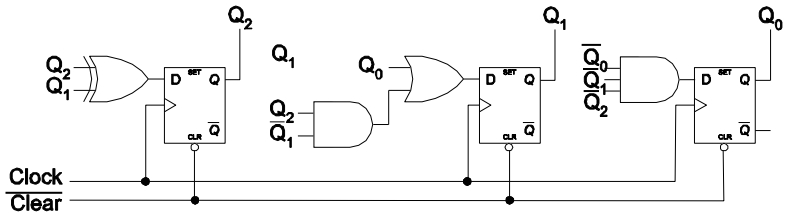
Simplified logic diagram:



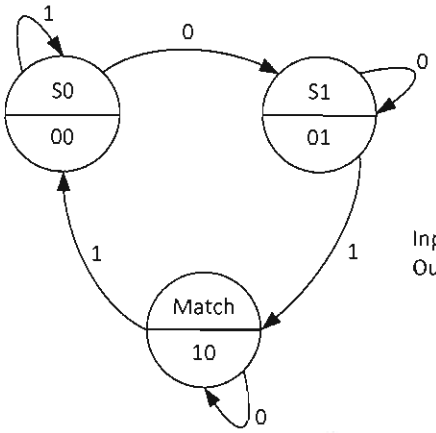
2. Draw the logic diagram for a 3-bit synchronous counter which counts in the sequence 000→001→010→100→110→000→… (i.e. 0,1,2,4,6).

Current Value			Next Value		
Q2	Q1	Q0	D2	D1	D0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	1	0	0
1	0	0	1	1	0
1	1	0	0	0	0

$$D0 = \overline{Q2}.\overline{Q1}.\overline{Q0}$$
$$D1 = Q0 + Q2.\overline{Q1}$$
$$D2 = Q2 \oplus Q1$$


Q2
Q1
Q2
Q1
Q1
Q0
Q0
Q1
Q2
Q0
D
SET
Q
CLR
Clock
Clear

3. For the state diagram below, construct both a one-dimensional and two-dimensional state table.



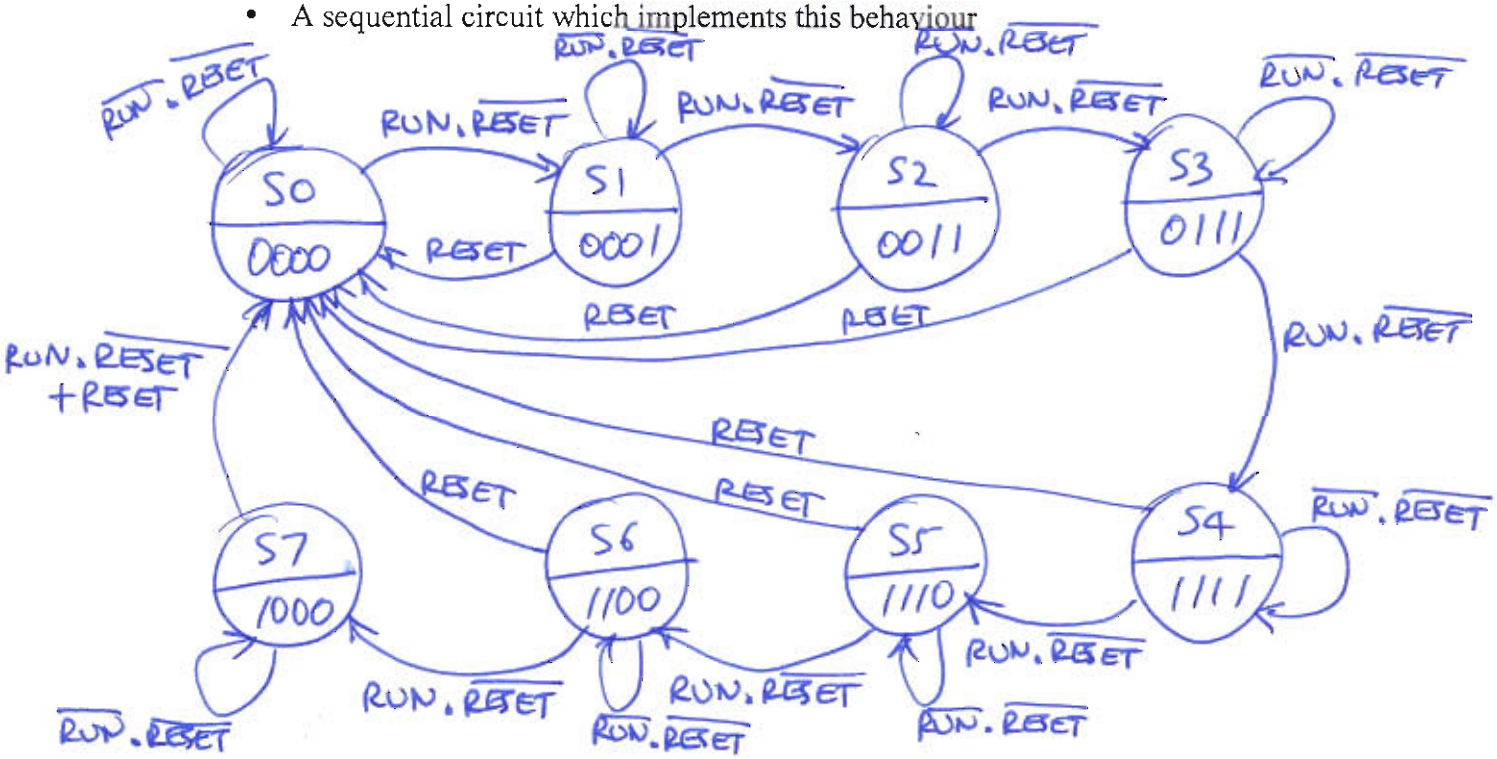
1D

Current State	Input(G)	Next State	Output (X_1X_0)
S0	0	S1	00
S0	1	S0	00
S1	0	S1	01
S1	1	Match	01
Match	0	Match	10
Match	1	S0	10

2D

Current State	Next State G=0	Next State G=1	Output X_1X_0
S0	S1	S0	00
S1	S1	Match	01
Match	Match	S0	10

4. Design a state machine which implements a counter which counts through the output sequence 0000→0001→0011→0111→1111→1110→1100→1000→0000… There are two inputs: RUN and RESET. If RUN is 1 and RESET is 0, the counter counts through the sequence given. If RESET is 1, independent of the value of RUN, the counter returns to output 0000. If both RUN and RESET are 0, the counter output remains the same. Show
   - A state diagram
   - A state table (two dimensional)
   - Your chosen state encoding
   - A sequential circuit which implements this behaviour



State table (2D)

Current State	Next State, Inputs RUN, RESET 00	01	10	11	Output $X_3X_2X_1X_0$
S0	S0	S0	S1	S0	0000
S1	S1	S0	S2	S0	0001
S2	S2	S0	S3	S0	0011
S3	S3	S0	S4	S0	0111
S4	S4	S0	S5	S0	1111
S5	S5	S0	S6	S0	1110
S6	S6	S0	S7	S0	1100
S7	S7	S0	S0	S0	1000

Chosen state encoding
- use 4 flip-flops, choose state = output (simplifies output logic)

2D State table

	Current state $Q_3\ Q_2\ Q_1\ Q_0$	Next state RESET=0, RUN=0 $D_3\ D_2\ D_1\ D_0$	RESET=0, RUN=1 $D_3\ D_2\ D_1\ D_0$	RESET=1 $D_3\ D_2\ D_1\ D_0$
S0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0
S1	0 0 0 1	0 0 0 1	0 0 1 1	0 0 0 0
S2	0 0 1 1	0 0 1 1	0 1 1 1	0 0 0 0
S3	0 1 1 1	0 1 1 1	1 1 1 1	0 0 0 0
S4	1 1 1 1	1 1 1 1	1 1 1 0	0 0 0 0
S5	1 1 1 0	1 1 1 0	1 1 0 0	0 0 0 0
S6	1 1 0 0	1 1 0 0	1 0 0 0	0 0 0 0
S7	1 0 0 0	1 0 0 0	0 0 0 0	0 0 0 0
		State stays same	State advances	Return to S0

Output logic

$$X_3 = Q_3$$
$$X_2 = Q_2$$
$$X_1 = Q_1$$
$$X_0 = Q_0$$

Next state logic

Consider D_0 – value is 0 if RESET is 1

- if RESET is 0
  - if RUN is 0 – value is Q_0 (no change)
  - if RUN is 1 – value is $\overline{Q_3}$

ie $D_0 = \overline{RESET} \cdot (\overline{RUN} \cdot Q_0 + RUN \cdot \overline{Q_3})$

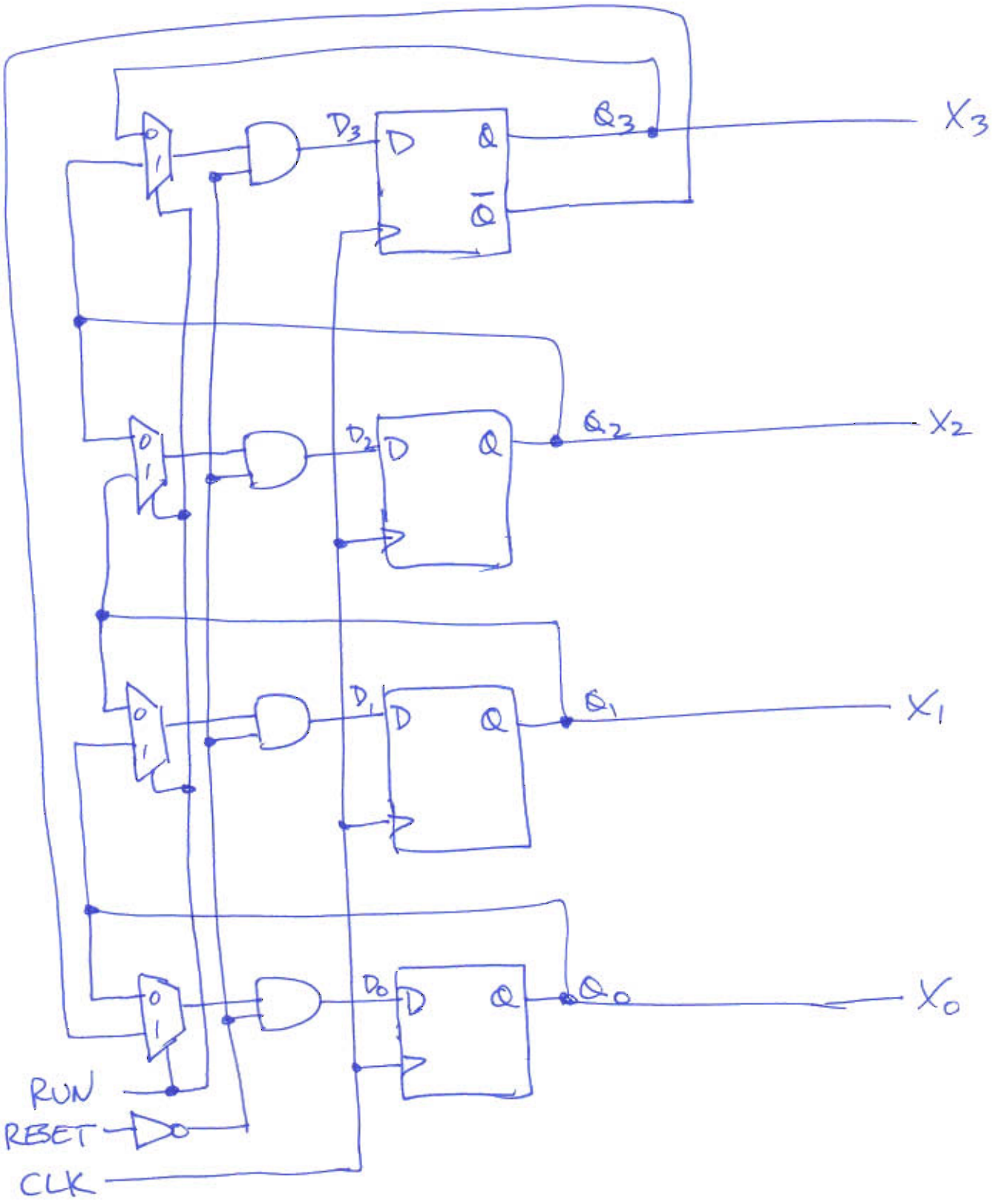
Consider D_1 – value is 0 if RESET is 1

- if RESET is 0
  - if RUN is 0 – value is 0, (Unchanged)
  - if RUN is 1 – value is Q_0

ie $D_1 = \overline{RESET} \cdot (\overline{RUN} \cdot Q_1 + RUN \cdot Q_0)$

Similarly $D_2 = \overline{RESET} \cdot (\overline{RUN} \cdot Q_2 + RUN \cdot Q_1)$

$D_3 = \overline{RESET} \cdot (\overline{RUN} \cdot Q_3 + RUN \cdot Q_2)$


D_3
D
Q
$\bar{Q}$
Q_3
X_3
D_2
Q_2
X_2
D_1
Q_1
X_1
D_0
Q_0
X_0
RUN
RESET
CLK