UNIVERSITY OF CALIFORNIA, RIVERSIDE

Department of Computer Science and Engineering
Department of Electrical Engineering
CS/EE120B – Introduction to Embedded Systems
Final
March 20, 2001

25

Name: Solution Key **Student ID#:**

Please print legibly

Lab Section: 21 (WF 6-10):____ **22** (MW 2-6):____ **23** (TR 6-10):____

(Numbers in parenthesis denote total possible points for question.)

1. Design a T flip-flop using a SR flip-flop and basic gates. (5)

Answer

The next-state table for the T flip-flop is

	Q_{next}	
Q	T = 0	T = 1
0	0	1
1	1	0

Converting the next-state table to the implementation table for the SR flip-flop, we get

	SR	
Q	T = 0	T = 1
0	0 ×	1 0
1	× 0	0 1

1 point for correct implementation table

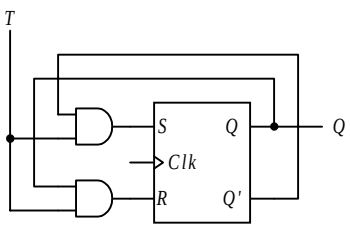
Solution 1:

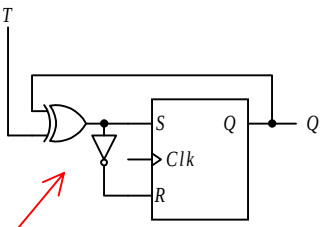
$S = TQ'$
$R = TQ$

Solution 2:

$S = T \oplus Q$
$R = (T \oplus Q)'$

2 points for correct equations





2 points for correct circuit

2. Synthesize the FSM circuit using T flip-flops for the following next-state / output table. (5)

Current State $Q_2Q_1Q_0$	Next State $Q_{2next}\ Q_{1next}\ Q_{0next}$		Output Y
	$C = 0$	$C = 1$	
000	000	001	0
001	001	010	0
010	010	011	0
011	011	100	0
100	100	101	0
101	101	000	1

Answer

The implementation table is as follows:

Current State $Q_2Q_1Q_0$	Implementation $T_2\ T_1\ T_0$	
	$C = 0$	$C = 1$
000	000	001
001	000	011
010	000	001
011	000	111
100	000	001
101	000	101

1 point for correct implementation table

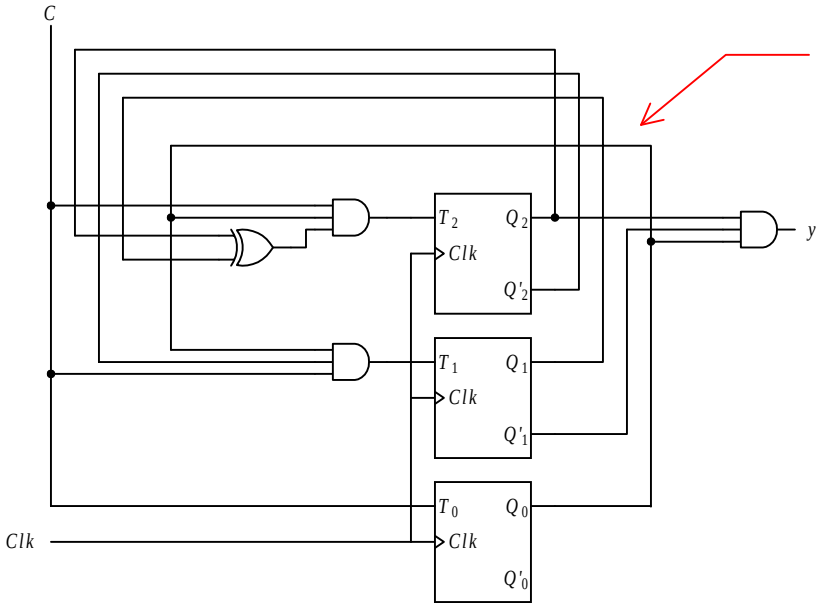
The next-state / output equations are as follows:

$$T_2 = CQ_2'Q_1Q_0 + CQ_2Q_1'Q_0$$
$$= CQ_0(Q_2 \oplus Q_1)$$
$$T_1 = CQ_2'Q_1'Q_0 + CQ_2'Q_1Q_0$$
$$= CQ_2'Q_0$$
$$T_0 = C$$

$$Y = Q_2Q_1'Q_0$$

2 points for the four equations (0.5 points each).

The FSM circuit is as follows:



2 points for the correct circuit.

3. Design a customized datapath and a State-action table for the Mealy FSM that solves the following problem. Your datapath should use as few single functional units and registers as possible. Your FSM should have a *start* and a *done* signal. You should first write the high-level pseudo-code that solves the problem. (5)

   Input (given) a 7-bit data. Output a '1' if there are odd number of 1 bits in the data, otherwise, output a '0'.

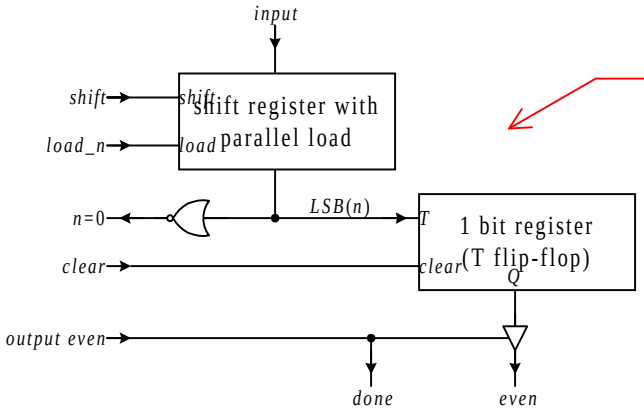
Answer

```
even = 0;
input n;
while n ≠ 0{
     if (LSB(n) == 1)
          even = not even;
     n = n >> 1;
}
output even;
output done;
```

1 point for the pseudo-code

For the datapath, we need:

- one shift register with parallel load for the variable *n*.
- a one bit register (T flip-flop) with load and clear for the variable *even*.



☞ 2 points for this datapath.

☞ 1 point only if the datapath uses more functional units like those shown below.

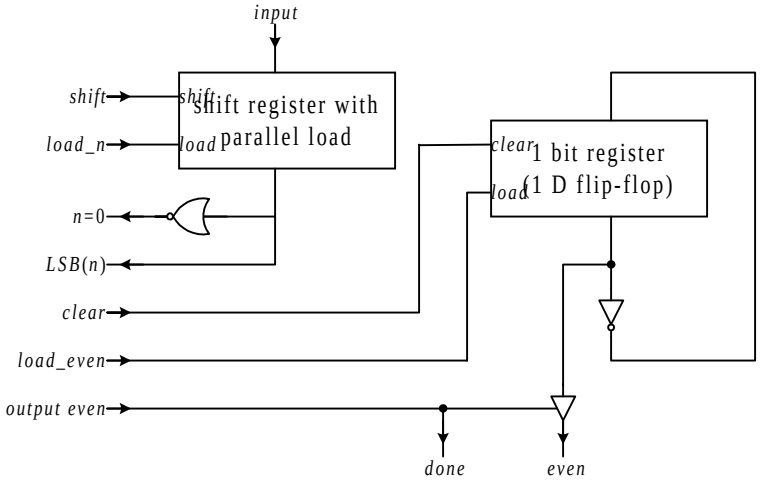
2 points for this state-action table.

The Mealy FSM is as follows:

Current State Q_1 Q_0 Name	Next State [Condition, State]	Control and Datapath Actions [Condition, Actions]
0 0 s_0	$\begin{bmatrix} start = 0, s_0 \\ start = 1, s_1 \end{bmatrix}$	*even* = 0 *done* = 0 *output* = *Z*
0 1 s_1	s_2	*n* = *input*
1 0 s_2	$\begin{bmatrix} n \neq 0, s_2 \\ n = 0, s_3 \end{bmatrix}$	$\begin{bmatrix} \text{LSB}(n) = 1, even = \text{not } even \\ n \neq 0, n = n >> 1 \end{bmatrix}$
1 1 s_3	s_0	*output* = *even* *done* = 1

Other possible datapath:

Datapath 2:

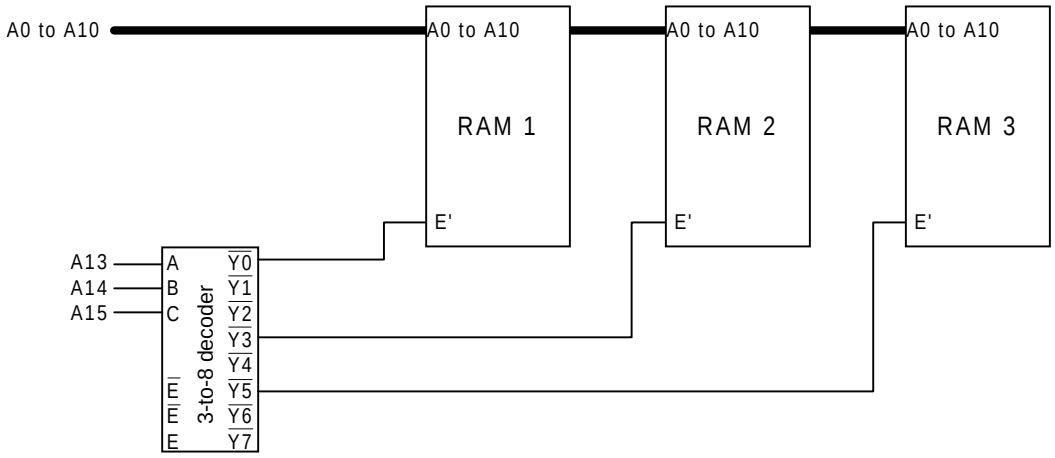


Datapath 3:

Algorithm: Using a counter to count the number of ones and a “divider” to divide by two to see if it is even or odd.

Functional units require: counter, divider, comparator

4. Given the following circuit for a memory system: (a) what are the starting addresses for the three RAM chips? (b) List all the addresses that would access the third location of RAM 1.(5)



Answer

(a)

RAM1 starting address: 0000, 0800, 1000, 1800
RAM2 starting address: 6000, 6800, 7000, 7800
RAM3 starting address: A000, A800, B000, B800

☞ 2 points for 0000, 6000, and A000 only.

☞ 2.5 points for all correct.

Reasoning:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	to decoder			not used		decoded inside RAM										
Y_0	0	0	0	×	×	0	0	0	0	0	0	0	0	0	0	0
Y_3	0	1	1	×	×	0	0	0	0	0	0	0	0	0	0	0
Y_7	1	1	1	×	×	0	0	0	0	0	0	0	0	0	0	0

(b)

Addresses that would access the third location of RAM 1:

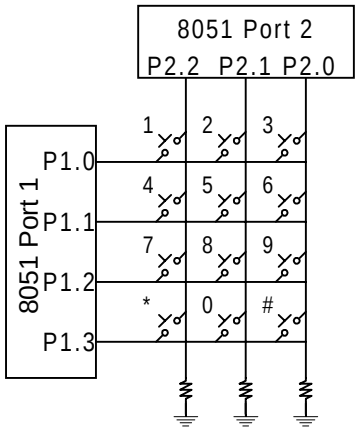
0002
0802
1002
1802

2.5 points for all four correct addresses.

Reasoning:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
to decoder			not used		decoded inside RAM 1										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0

5. Given a 3×4 matrix keypad (like the telephone keypad) and the connections for the keypad to the 8051 as shown below. Note that each intersection of a vertical and horizontal wire is not connected unless the corresponding key is pressed.



a) Which port (P1 or P2) should be the input port and which should be the output port? (1)

b) Which one of the following code segments will return the correct key pressed? (2)

① P1 = 0x10;
if(P2.2 == 0) return '1';

② P2 = 0x06;
if(P1.2 == 0) return '7';

③ P1 = 0x04;
if(P2.1 == 1) return '8';

④ P2 = 0x04;
if(P1.3 == 1) return '*';

⑤ P1 = 0x06;
if(P2.0 == 1) return '6';

c) Write a segment of C code (no longer than 5 lines) that will determine that there is no key pressed and return a space (' '). (2)

Answer

a) P1 is the output port and P2 is the input port.

1 point for this

b) ③

2 points for this

c)

```
P1 = 0x0F;
if((P2.0 == 0) && (P2.1 == 0) && (P2.2 == 0)) return ' ';
```

2 points for this