[CS M51A F14] Solution to Homework 6

TA: Yang Lu (yangluphil@gmail.com)

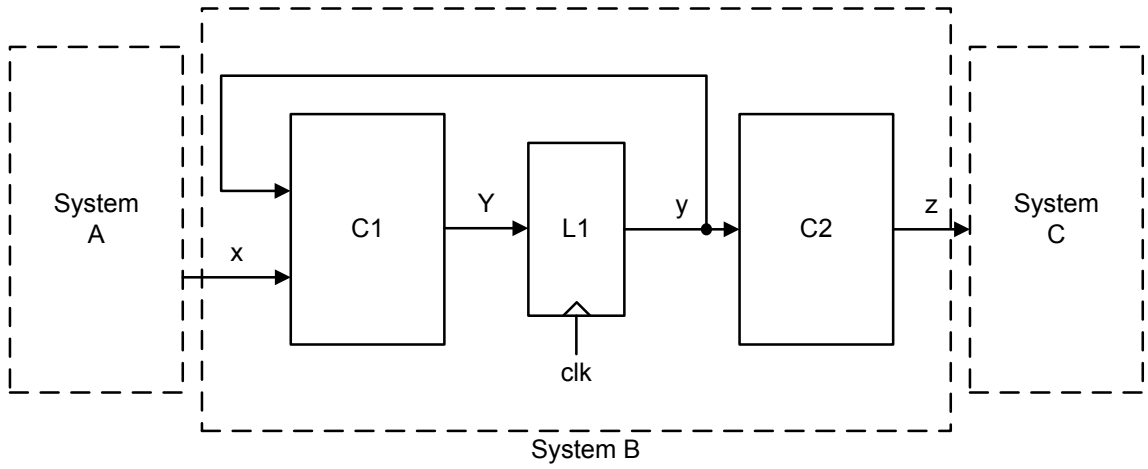
Homework Problems

Exercises

From the book: 9.5, 9.15, 10.7, 10.11, 11.11, 11.13

Problem 1

We would like to analyze the timing required for the following sequential system. All registers in the system are positive edge triggered flip-flops. Propagation delays of all registers are equal at $t_p = 1.5$ ns, and setup times for all registers are also fixed at $t_{su} = 0.4$ ns.



For System A, the delay from the output of the state register to signal x is $d2_A = 2.5$ ns.

For System B, the delay with respect to input x is $d1^x = 3.5$ ns, the delay with respect to state register value y is $d1^y = 4.5$ ns, and the delay of the output combinational logic is $d2_B = 2.5$ ns.

For System C, the delay for signal z to reach the state register of System C is $d1^z = 2.7$ ns.

1. Assume no clock skew for the three systems. What is the minimum clock period required in regard to signal x?

   Solution Regarding signal x:

$$\begin{aligned} & t_{in} + d1^x + t_{suB} \\ = \; & t_{pA} + d2_A + d1^x + t_{suB} \\ = \; & 1.5 + 2.5 + 3.5 + 0.4 \\ = \; & 7.9 \text{ (ns)} \end{aligned}$$

2. What is the minimum clock period required in regard to signal y?

   Solution Regarding signal y, we have:

$$\begin{aligned} & t_{pB} + d1^y + t_{suB} \\ = \; & 1.5 + 4.5 + 0.4 \\ = \; & 6.4 \text{ (ns)} \end{aligned}$$

3. What is the minimum clock period required in regard to signal z?

   Solution Regarding signal z, we get:

$$\begin{aligned} & t_{pB} + d2_B + t_{out} \\ = \; & t_{pB} + d2_B + d1^z + t_{suC} \\ = \; & 1.5 + 2.5 + 2.7 + 0.4 \\ = \; & 7.1 \text{ (ns)} \end{aligned}$$

4. Considering all the three previous values, what is the minimum clock period required for the whole system?

   Solution T_{min} should be larger than all three values, thus the minimum $T_{min} = max\{7.9, 6.4, 7.1\} = 7.9$ (ns).

5. Now, increased distance between the three systems has caused clock skew, and we need to take it into consideration. System C is the closest to the clock source, and the clock arrives at System C the fastest. The clock arrives at System B 0.3 ns later than it arrives at System C, and it arrives at System A 0.6 ns later than System C.

   Of the three clock periods related to signals x, y and z, calculate the adjusted minimum clock period of the affected signals. Also, what is the minimum clock period for the whole system with clock skew?

   Solution Two clock periods are affected by the clock skew, the delay regarding signal x, and signal z. In both cases, the clock arrives faster at the system setup, thus in effect we need longer clock periods to make up for the lost time.

   Therefore, the new delay values are:

$$\begin{aligned} delay_x &= 7.9 + 0.3 = 8.2 \text{ (ns)} \\ delay_y &= 6.4 \text{ (ns) (no change)} \\ delay_z &= 7.1 + 0.3 = 7.4 \text{ (ns)} \end{aligned}$$

   The new $T_{min} = 8.2$ ns.

Problem 2

Complete the following table for the given representations in each part.

	Signed integer x (in decimal)	Representation x_R (in decimal)	Bit-vector $\underline{x}$
a	-25		
b		37	
c			110110

The 'Signed integer' column shows the actual value of the signed integer, and the 'Representation' column holds the value of the bit-vector representation. Assume that any bit vector that is shorter than the given bit length has leading 0s.

1. Two's complement, $n = 7$ bits

   Solution

   For a: Since $25 = 16 + 8 + 1 = 11001_{(2)}$ and $n = 7$, we can get

$$\begin{aligned} \underline{x} &= x' + 1 = 1100110 + 1 = 1100111_{(2)} \\ x_R &= 1100111_{(2)} = 64 + 32 + 4 + 2 + 1 = 103 \end{aligned}$$

   For b: $\underline{x} = 37 = 32 + 4 + 1 = 100101_{(2)}$ and x is also the same value.

   For c: $x = x_R = 0110110_{(2)} = 54$

	Signed integer x (in decimal)	Representation x_R (in decimal)	Bit-vector $\underline{x}$
a	-25	103	1100111
b	37	37	0100101
c	54	54	110110

2. Two's complement, $n = 6$ bits

   Solution

   For a: Since $25 = 16 + 8 + 1 = 11001_{(2)}$ and $n = 6$, we can get

$$\begin{aligned} \underline{x} &= x' + 1 = 100110 + 1 = 100111_{(2)} \\ x_R &= 100111_{(2)} = 32 + 4 + 2 + 1 = 39 \end{aligned}$$

   For b:

$$\begin{aligned} \underline{x} &= 37 = 32 + 4 + 1 = 100101_{(2)} \\ x &= -32 + 4 + 1 = -27 \end{aligned}$$

   For c:

$$\begin{aligned} x &= -32 + 16 + 4 + 2 = -10 \\ x_R &= 110110_{(2)} = 32 + 16 + 4 + 2 = 54 \end{aligned}$$

	Signed integer x (in decimal)	Representation x_R (in decimal)	Bit-vector $\underline{x}$
a	-25	39	100111
b	-27	37	100101
c	-10	54	110110

3. Ones' complement, $n = 6$ bits

   Solution

   For a: Since $25 = 16 + 8 + 1 = 11001_{(2)}$ and $n = 6$, we can get

$$\begin{aligned} \underline{x} &= x' + 1 = 100110 = 100110_{(2)} \\ x_R &= 100110_{(2)} = 32 + 4 + 2 = 38 \end{aligned}$$

   For b:

$$\begin{aligned} \underline{x} &= 37 = 32 + 4 + 1 = 100101_{(2)} \\ x &= (-32 + 4 + 1) + 1 = -26 \end{aligned}$$

For c:

$$\begin{aligned} x_R &= 110110_{(2)} = 32 + 16 + 4 + 2 = 54 \\ x &= (-32 + 16 + 4 + 2) + 1 = -9 \end{aligned}$$

	Signed integer x (in decimal)	Representation x_R (in decimal)	Bit-vector $\underline{x}$
a	-25	38	100110
b	-26	37	100101
c	-9	54	110110

Problem 3

We would like to identify the working signals of an arithmetic unit at work, namely the two's-complement arithmetic unit shown in the textbook at Figure 10.12 (page 297). For the system, fill in the table below according to the given computation in each part. c_0, K_x and K_y are control signals, and z, c_{out}, ovf, $zero$ and sgn are result signals.

1. $z = x + y$

   Solution For addition, $c_0 = K_x = K_y = 0$.

x	y	c_0	K_x	K_y	z	c_{out}	ovf	$zero$	sgn
00000000	00000000	0	0	0	00000000	0	0	1	0
10101010	10100101	0	0	0	01001111	1	1	0	0
10110110	00110011	0	0	0	11101001	0	0	0	1

2. $z = x - y$

   Solution For subtraction of two's complement numbers, $c_0 = 1$, $K_x = 0$, $K_y = 1$.

x	y	c_0	K_x	K_y	z	c_{out}	ovf	$zero$	sgn
00100010	00100010	1	0	1	00000000	1	0	1	0
10010010	01000011	1	0	1	01001111	1	1	0	0
00010011	00110011	1	0	1	11100000	0	0	0	1

Problem 4

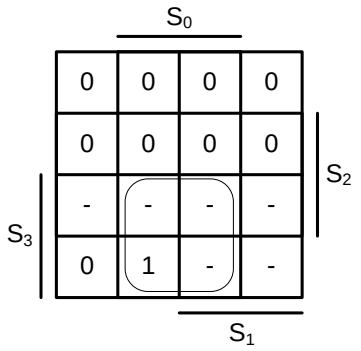
Using a modulo-16 binary counter with parallel inputs explained in the textbook (p. 321) along with logic gates, we would like to implement the following counters.

For each design, minimize the gate logic for the input signals and load signal as much as possible using K-maps.

1. A modulo-10 counter.

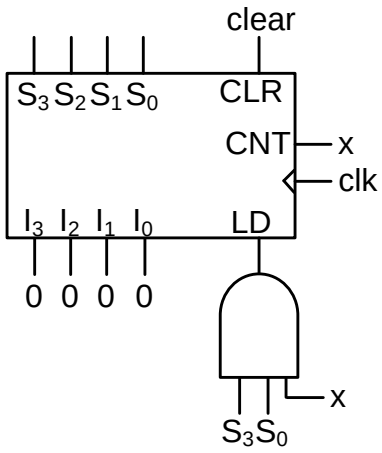
   Solution For a modulo-10 counter, we need to make a jump from 9 to 0, and the values from 10 to 15 are don't-cares as they are never reached.

   The K-map for the LD signal is:



where we can get $LD = xS_3S_0$.

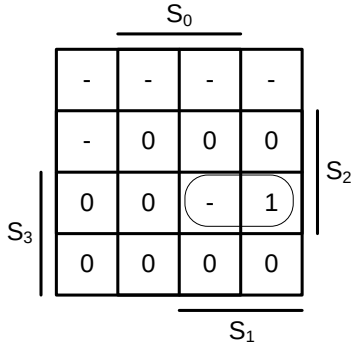
The counter implementation is:



2. A 5-to-14 counter.

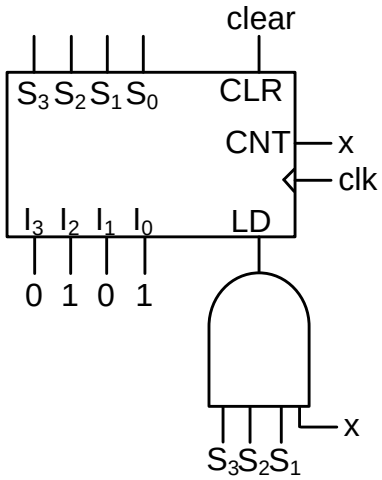
Solution A 5-to-14 counter starts from 5 and jumps from 14 to 5. Values from 0 to 4 and 15 are don't-cares as they are never reached.

The K-map for the LD signal is:



where we can get $LD = xS_3S_2S_1$.

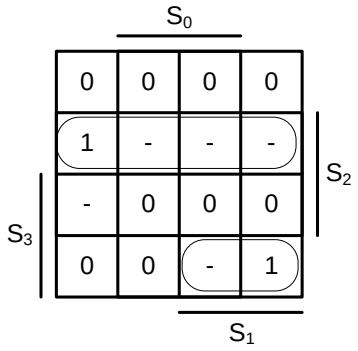
The counter implementation is:



3. A counter which counts in the following sequence: 0, 1, 2, 3, 4, 8, 9, 10, 13, 14, 15

   Solution This counter makes two jumps, from 4 to 8, and 10 to 13. Values 5, 6, 7, 11, and 12 are don't-cares as they are never reached.

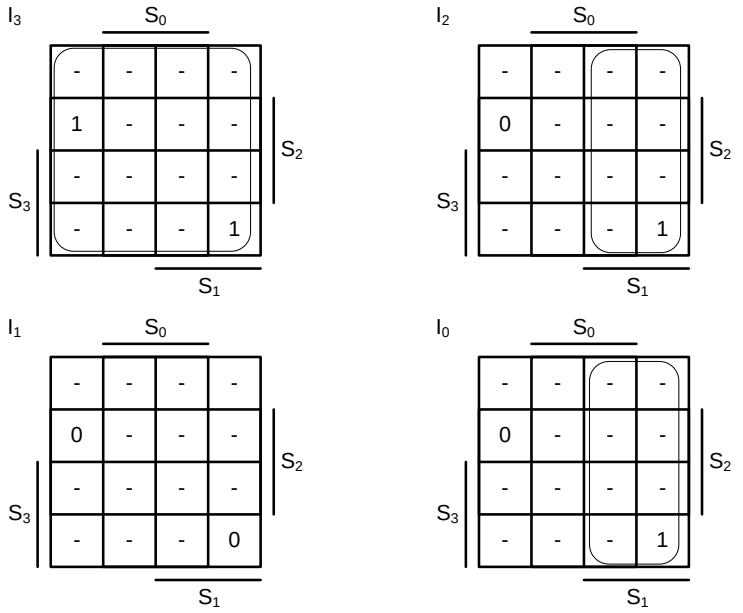
   The K-map for the LD signal is:



   where we can get $LD = x(S_3'S_2 + S_3S_2'S_1)$.

   The values of I_0 to I_3 need to be: $I = \begin{cases} 1000 & \text{if } S = 4, \\ 1101 & \text{if } S = 10, \\ DC & \text{otherwise} \end{cases}$

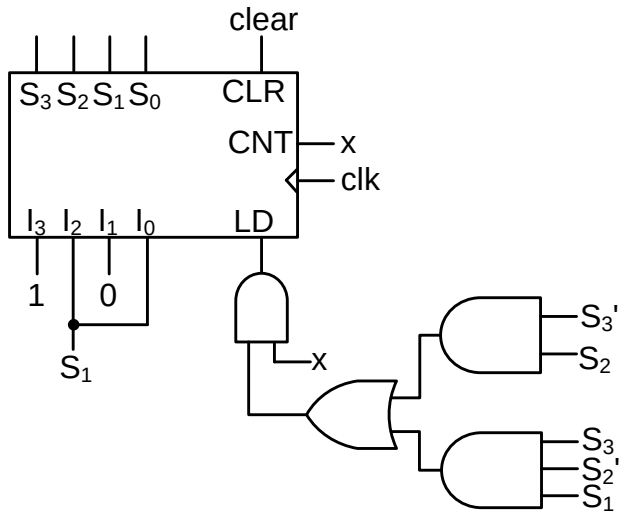
   The K-maps are as follows:



and we can get

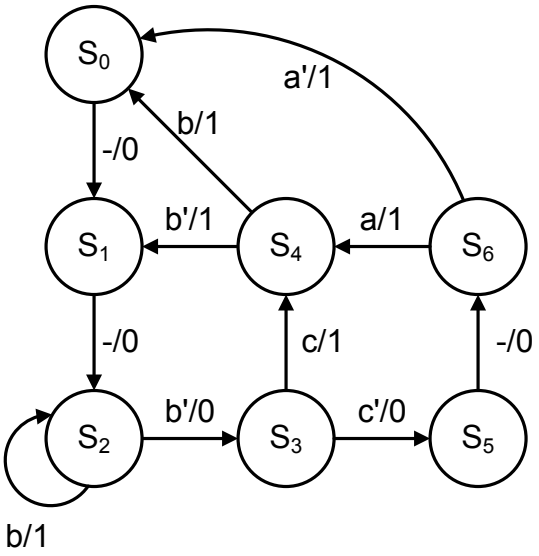
$$\begin{aligned} I_3 &= 1 \\ I_2 &= S_1 \\ I_1 &= 0 \\ I_0 &= S_1 \end{aligned}$$

The counter implementation is:



Problem 5

Implement a sequential system using the standard modulo-16 binary counter with parallel input. The system has three binary inputs a, b, c and one binary output z. The transition and the output functions are specified by the state diagram shown below.



1. Setting $LD = CNT'$, obtain the expression for the input CNT in terms of states (S_0 to S_6) and input signals.

   Solution Looking at the state diagram, we find all cases where the state index increments by 1. This gives us:

   $$CNT = S_0 + S_1 + S_2b' + S_3c + S_5$$

2. Find all load input combinations that we need for this system. For each combination, write the input conditions in terms of states (S_0 to S_6) and input signals. Simplify the conditions as much as possible.

   Solution We utilize parallel loads when the transition is not an increment, i. e. cannot be implemented by setting $CNT = 1$. The states are S_0, S_1, S_2, S_4 and S_5. The expressions we get directly are shown:

   $$I_3I_2I_1I_0 = \begin{cases} 0000 & \text{if} \quad S_4b + S_6a' \\ 0001 & \text{if} \quad S_4b' \\ 0010 & \text{if} \quad S_2b \\ 0100 & \text{if} \quad S_6a \\ 0101 & \text{if} \quad S_3c' \end{cases}$$

   To simplify the expressions, the input can be removed for conditions that are tied to a certain state, since the LD signal is only invoked when $CNT = 0$. For instance, when we are at S_2 and $LD = 1$, b will never be 0 because if $b = 0$, then $S_2b' = 1$ and CNT must be 1 also. For S_4 and S_6, we cannot remove the input variable as the load value is affected by the input signal. For example, at S_4, if $b = 0$ then $I_0 = 1$, if $b = 1$ then $I_0 = 0$.

   With the simplifications in place, we get:

   $$I_3I_2I_1I_0 = \begin{cases} 0000 & \text{if} \quad S_4b + S_6a' \\ 0001 & \text{if} \quad S_4b' \\ 0010 & \text{if} \quad S_2 \\ 0100 & \text{if} \quad S_6a \\ 0101 & \text{if} \quad S_3 \end{cases}$$

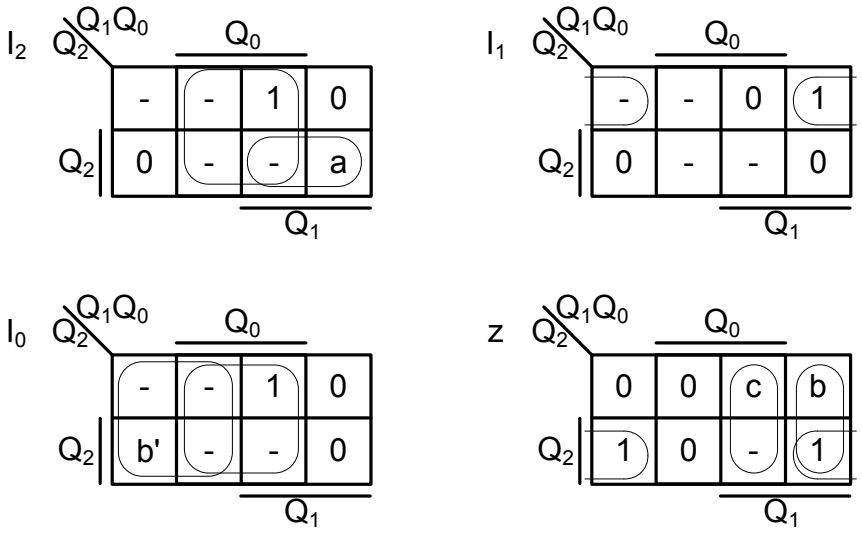
3. Using K-maps, obtain the minimal expressions for I_3 to I_0 and the output z, in terms of counter outputs (Q_3 to Q_0) and input signals.

   Solution We take each I_i as a separate output and build the K-maps with respect to the state values. For example, looking at the answer from the previous section, I_2 needs to be 0 for $S_4b + S_6a'$, S_4b' and S_2, and

it needs to be 1 for $S_6 a$ and S_3. The state number indicates the square on the K-map, and any variables become the values inside the squares. In the index 6 square for I_2, the output needs to be 0 for $a' = 1$ (or $a = 0$) and 1 for $a = 1$, therefore the value that goes here is a.

Similar rules apply for the output z. We can find an output condition for each state in the state diagram, and write a value in the K-map for each one. For example, S_0's output is 0 regardless of the input, so we put a 0 in the square for index 0 in the K-map. At S_3, the output is 0 when $c' = 1$ (or $c = 0$) and 1 when $c = 1$. So we write c in the square for index 3.

Following these principles, we can obtain the K-maps below. We do not need a K-map for I_3 since it is always 0.



Regarding covers that contain variables, we can consider them as special covers that will only cover that variable. They should never cover any 0 squares, and any 1 squares that they cover are considered incomplete, since they will only be valid in cases when the variable is 1. Therefore, all constant 1 squares need to have a purely constant 1 cover to be completely covered.

For instance, in the K-map for z, we will need a separate constant 1 cover for the square at index 6 even if the value at index 4 were to be 0. This is because the b cover is only partially covering index 6. And in the same K-map, if index 0 were to be 1, we still cannot have a cover like Q_0 in the final expression, which is a constant 1 cover going over a variable square. Since variable squares can be 0 or 1 as the input changes, any constant cover should never cover any variables.

The final expressions for each load input and the output is:

$$\begin{aligned} I_3 &= 0 \\ I_2 &= Q_0 + Q_2 Q_1 a \\ I_1 &= Q_2{}' Q_0{}' \\ I_0 &= Q_0 + Q_1{}' b' \\ z &= Q_2 Q_0{}' + Q_1 Q_0{}' b + Q_1 Q_0 c \end{aligned}$$