Homework 4

CS51A Fall 2018 Solutions- Due date November 9

November 21, 2018

1. A combinational system has four inputs a, b, c, d and one output y which is 1 iff $8a + 4b + 2c + d$ is a prime. Find both SOP and POS formulas and identify which one will have minimal complexity.Design a minimal two-level network to implement this system.

   Solution

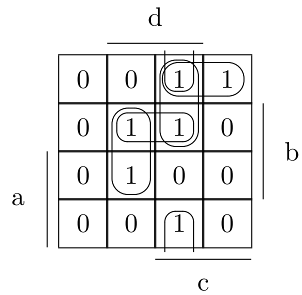
   Input: (a, b, c, d), with $a, b, c, d \in \{0, 1\}$

   Output: $y \in \{0, 1\}$

   Function:

$$y = \begin{cases} 1 & \text{if } (8a + 4b + 2c + d) \text{ is prime} \\ 0 & \text{otherwise} \end{cases}$$

input value	abcd	y
0	0000	0
1	0001	0
2	0010	1
3	0011	1
4	0100	0
5	0101	1
6	0110	0
7	0111	1
8	1000	0
9	1001	0
10	1010	0
11	1011	1
12	1100	0
13	1101	1
14	1110	0
15	1111	0

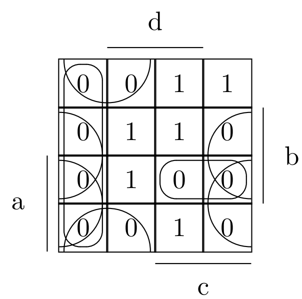


From the Kmap we get the following prime implicants: $a'bd$, $b'cd$, $a'b'c$, $a'cd$, and $bc'd$ of which the essential prime implicants are: $b'cd$, $a'b'c$, and $bc'd$.

A minimal sum of products for function y is:

$$y = b'cd + a'b'c + bc'd + a'cd$$

.



From the Kmap above we get the following prime implicates: $(b'+d), (a'+d), (b+c), (c+d)$ and $(a'+b'+c')$ of which $(c+d)$ prime implicate is not essential. So, the minimal product of sums in this case is:

$$y = (b' + d)(a' + d)(b + c)(a' + b' + c')$$

.

Notice that the cost of the product of sums is lower because it needs only one three input first level gate.

2. Obtain minimal switching expressions for computing residue mod7 of an integer in range 0 to 15.

   Solution

   Input: x in the range $[0, 15]$

   Output: y in the range $[0, 7]$

   Function: $y = x \bmod 7$

   The switching functions are shown in the table

$x = (x_3x_2x_1x_0)$	$y = (y_2y_1y_0)$
0000	000
0001	001
0010	010
0011	011
0100	100
0101	101
0110	110
0111	000
1000	001
1001	010
1010	011
1011	100
1100	101
1101	110
1110	000
1111	001

From K-maps (not shown) we obtain the following minimal switching expressions

$$
\begin{aligned}
y_2 &= (x_3 + x_2)(x_2' + x_1' + x_0')(x_3' + x_1' + x_0)(x_2 + x_1) \\
&= x_2x_1' + x_3x_2'x_1x_0 + x_3'x_2x_0' \\
y_1 &= (x_3 + x_1)(x_1 + x_0)(x_3 + x_2' + x_0')(x_3' + x_2' + x_1')(x_3' + x_1' + x_0') \\
&= x_3x_1'x_0 + x_3'x_2'x_1 + x_3'x_1x_0' + x_2'x_1x_0' \\
y_0 &= (x_3 + x_0)(x_3' + x_2 + x_0')(x_3' + x_1 + x_0')(x_2' + x_1' + x_0)(x_3 + x_2' + x_1') \\
&= x_3x_1'x_0' + x_3x_2'x_0' + x_3x_2x_1x_0 + x_3'x_1'x_0 + x_3'x_2'x_0
\end{aligned}
$$

Pick *SOP* for y_2, *SOP* for y_1 and *POS* for y_0.

3. Design a two-level NOR network that performs four-bit binary to gray code conversion. Minimize each output separately by K-MAP.
Solution

Input: binary code represented as $\underline{b} = (b_3b_2b_1b_0)$, where $b_i \in \{0,1\}$

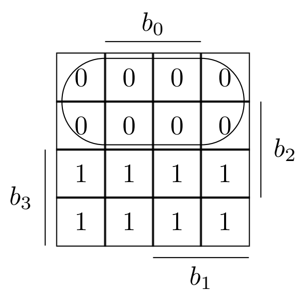
Output: Gray code represented as $\underline{g} = (g_3g_2g_1g_0)$, where $g_i \in \{0,1\}$

Function: g is the Gray code that corresponds to b. The correspondence between binary and Gray codes is shown in the following table:

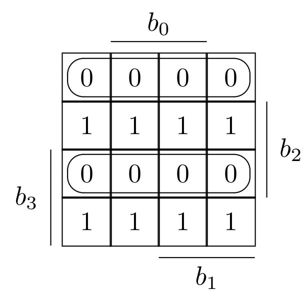
Binary $b_3b_2b_1b_0$	Gray $g_3g_2g_1g_0$
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

The correspoding Kmaps are as follows:

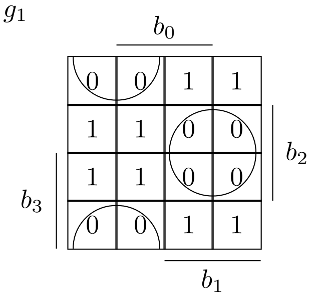
g_3



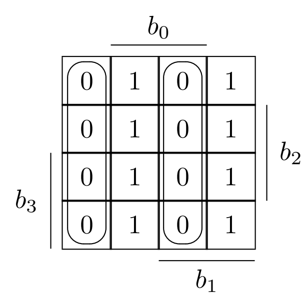
g_2



g_1



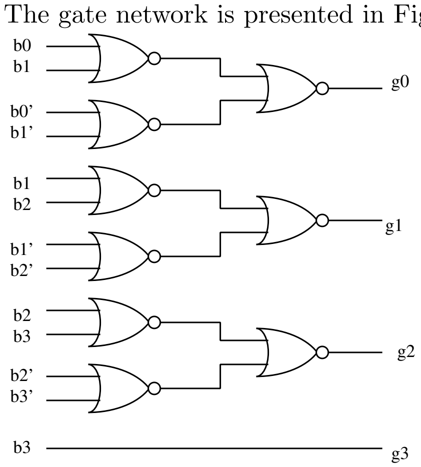
g_0



To obtain a NOR-NOR network we produce the minimal product of sums:

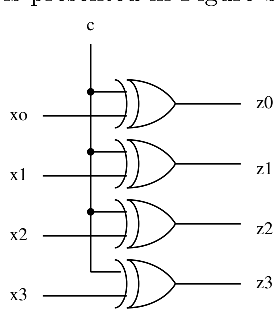
$$\begin{aligned} g_3 &= b_3 \\ g_2 &= (b_2 + b_3)(b_2' + b_3') \\ g_1 &= (b_1 + b_2)(b_1' + b_2') \\ g_0 &= (b_0 + b_1)(b_0' + b_1') \end{aligned}$$

The gate network is presented in Figure below.



4. Design a complementer with only XOR gates which perform $z_i = x_i c_i' + x_i' c_i$ for $0 \leq i \leq 3$ where x_i's and c are network inputs and z_i's are network outputs.
   Solution

   The network that implements a 4-bit complementer using only XOR gates is presented in Figure below.



5. A sequential system is described by the following expresssions, where the state is a bit-vector of four components (s_3, s_2, s_1, s_0):

$$s_0(t+1) = s_3(t) \oplus x(t)$$

$$s_i(t+1) = s_{i-1}(t) \quad \text{for } 1 \leq i \leq 3$$

$$z(t) = s_3(t)$$

Solution

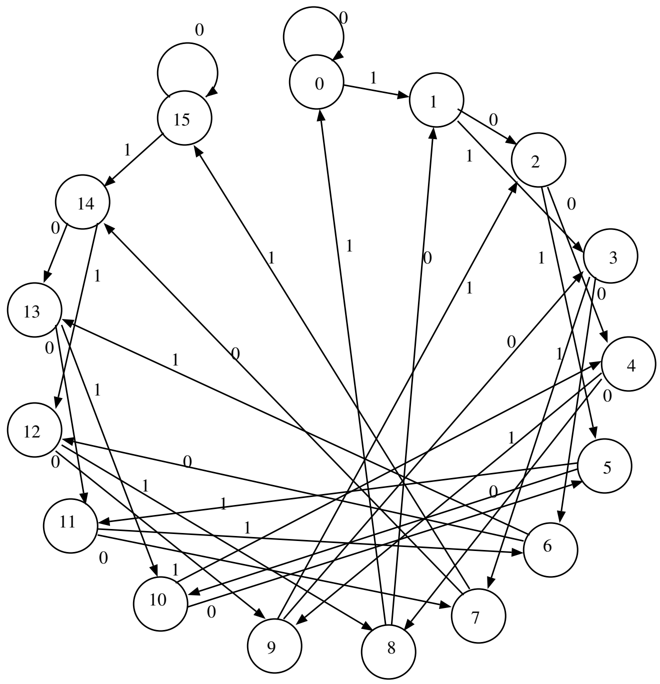
There are sixteen possible states. To obtain the state diagram, we first obtain the state table, by an evaluation of the expressions. To simplify the notation, we label the states with an integer $0 \leq j \leq 15$ whose binary representation is the bit-vector (s_3, s_2, s_1, s_0). The state table is

	Input		
PS	$x = 0$	$x = 1$	
0	0	1	0
1	2	3	0
2	4	5	0
3	6	7	0
4	8	9	0
5	10	11	0
6	12	13	0
7	14	15	0
8	1	0	1
9	3	2	1
10	5	4	1
11	7	6	1
12	9	8	1
13	11	10	1
14	13	12	1
15	15	14	1
	NS		z

Observe that the state transition function is:

$$s(t+1) = (2s(t) + (\left\lfloor \frac{s(t)}{8} \right\rfloor + x(t)) \bmod 2) \bmod 16$$

The corresponding state diagram is shown in



6. A sequential system has an imput with values a, b, and c, and one binary output. The output at time t is 1 if $x(t-3,t) = abca$ and the number of a's in $x(0,t)$ is even. Obtain a "loose" state desciption and minimize the number of states.

<u>Solution</u>

The loose description has eight states as follows:

$$
\begin{aligned}
s(t) &= A \textbf{ if } x(t-1) = a \textbf{ and } \# \text{ of } a's \text{ is even} \\
s(t) &= B \textbf{ if } x(t-1) = a \textbf{ and } \# \text{ of } a's \text{ is odd} \\
s(t) &= C \textbf{ if } x(t-2,t-1) = ab \textbf{ and } \# \text{ of } a's \text{ is even} \\
s(t) &= D \textbf{ if } x(t-2,t-1) = ab \textbf{ and } \# \text{ of } a's \text{ is odd} \\
s(t) &= E \textbf{ if } x(t-3,t-1) = abc \textbf{ and } \# \text{ of } a's \text{ is even} \\
s(t) &= F \textbf{ if } x(t-3,t-1) = abc \textbf{ and } \# \text{ of } a's \text{ is odd} \\
s(t) &= G \textbf{ if } x(t-3,t-1) = other \textbf{ and } \# \text{ of } a's \text{ is even} \\
s(t) &= H \textbf{ if } x(t-3,t-1) = other \textbf{ and } \# \text{ of } a's \text{ is odd}
\end{aligned}
$$

The corresponding state table is

PS	$x=a$	$x=b$	$x=c$
		$Input$	
A	$B,0$	$C,0$	$G,0$
B	$A,0$	$D,0$	$H,0$
C	$B,0$	$G,0$	$E,0$
D	$A,0$	$H,0$	$F,0$
E	$B,0$	$G,0$	$G,0$
F	$A,1$	$H,0$	$H,0$
G	$B,0$	$G,0$	$G,0$
H	$A,0$	$H,0$	$H,0$
		$NS, Output$	

From the table we get

$$P_1 = (A, B, C, D, E, G, H)(F)$$

To obtain P_2, we determine the class of P_1 to which each successor of the states belong.

	1 (A, B, C, D, E, G, H)	2 (F)
a	1 1 1 1 1 1 1	1
b	1 1 1 1 1 1 1	1
c	1 1 1 2 1 1 1	1

Consequently,

$$P_2 = (A, B, C, E, G, H)(D)(F)$$

To obtain P_3, we repeat the process

	1 (A, B, C, E, G, H)	2 (D)	3 (F)
a	1 1 1 1 1 1	1	1
b	1 2 1 1 1 1	1	1
c	1 1 1 1 1 1	3	1

Thus,

$$P_3 = (A, C, E, G, H)(B)(D)(F)$$

And once more,

	1 (A, C, E, G, H)	2 (B)	3 (D)
a	2 2 2 2 1	1	1
b	1 1 1 1 1	3	1
c	1 1 1 1 1	1	4

So

$$P_4 = (A, C, E, G)(H)(B)(D)(F)$$

Still not ready! Once more (hopefully the last)

	1 (A, C, E, G)	2 (B)	3 (D)	4 (F)	5 (H)
a	2 2 2 2	1	1	1	1
b	1 1 1 1	3	5	5	5
c	1 1 1 1	5	4	5	5

So, finally!

$$P = P_5 = P_4 = (A, C, E, G)(B)(D)(F)(H)$$

and the reduced table is

	Input		
PS	$x = a$	$x = b$	$x = c$
A	$B, 0$	$A, 0$	$A, 0$
B	$A, 0$	$D, 0$	$H, 0$
D	$A, 0$	$H, 0$	$F, 0$
F	$A, 1$	$A, 0$	$A, 0$
H	$A, 0$	$H, 0$	$H, 0$
	$NS, Output$		