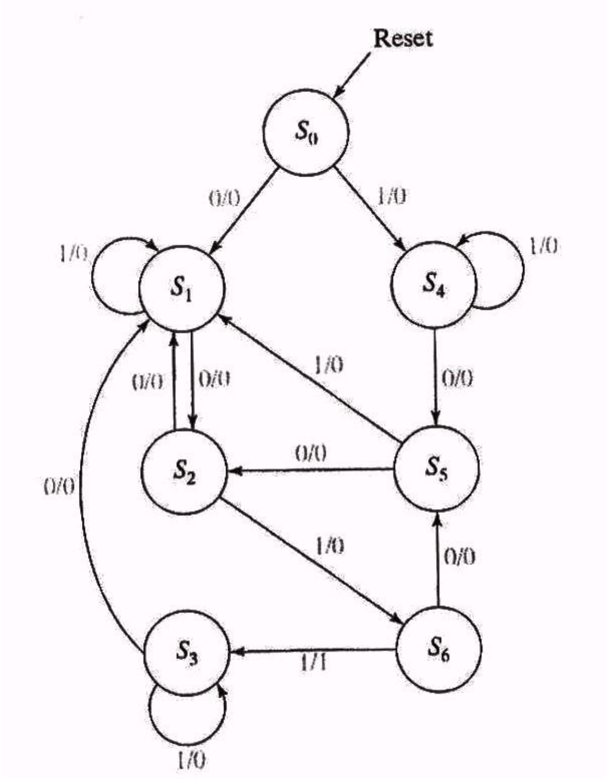
UCLA
Department of Electrical Engineering
EEM16 – Fall 2012
Final Exam Solution
December 10, 2012
(The final contains 7 problems)

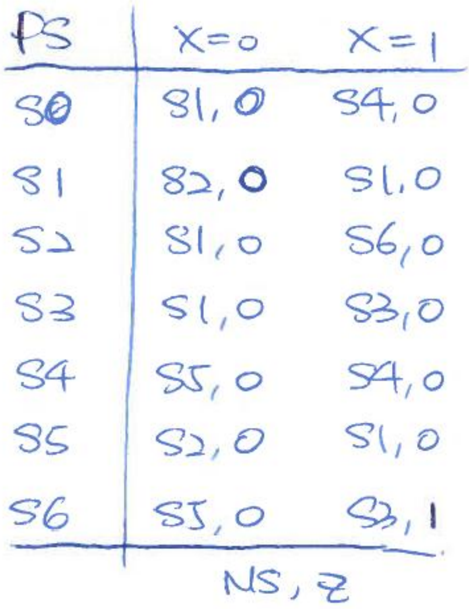
1. Exam is closed book. You are allowed **two 8 ½ x 11" double-sided cheat sheets**.
2. Calculators are allowed.
3. Show the intermediate steps leading to your final solution for each problem.
4. You can use both sides of the sheets to answer questions.

Problem	Points	Your Score	Comments
1	10		
2	15		
3	15		
4	15		
5	15		
6	15		
7	15		
	Total: 100		

Problem 1 *(10 points)*

State Minimization: Given the state diagram in the figure below, determine which states should be combined to determine the reduced state diagram.





PS	X=0	X=1
S0	S1,0	S4,0
S1	S2,0	S1,0
S2	S1,0	S6,0
S3	S1,0	S3,0
S4	S5,0	S4,0
S5	S2,0	S1,0
S6	S5,0	S3,1
	NS, Z	

	G1						G2
P1	(S_0	S_1	S_2	S_3	S_4	S_5)	(S_6)
0	1	1	1	1	1	1	
1	1	1	2	1	1	1	

	G1					G2	G3
P2	(S_0	S_1	S_3	S_4	S_5)	(S_2)	(S_6)
0	1	2	1	1	2		
1	1	1	1	1	1		

	G1			G2		G3	G4
P3	(S_0	S_3	S_4)	(S_1	S_5)	(S_2)	(S_6)
0	2	2	2	3	3	2	2
1	1	1	1	2	2	4	1

⇒ P4 = P3

Reduced state diagram.

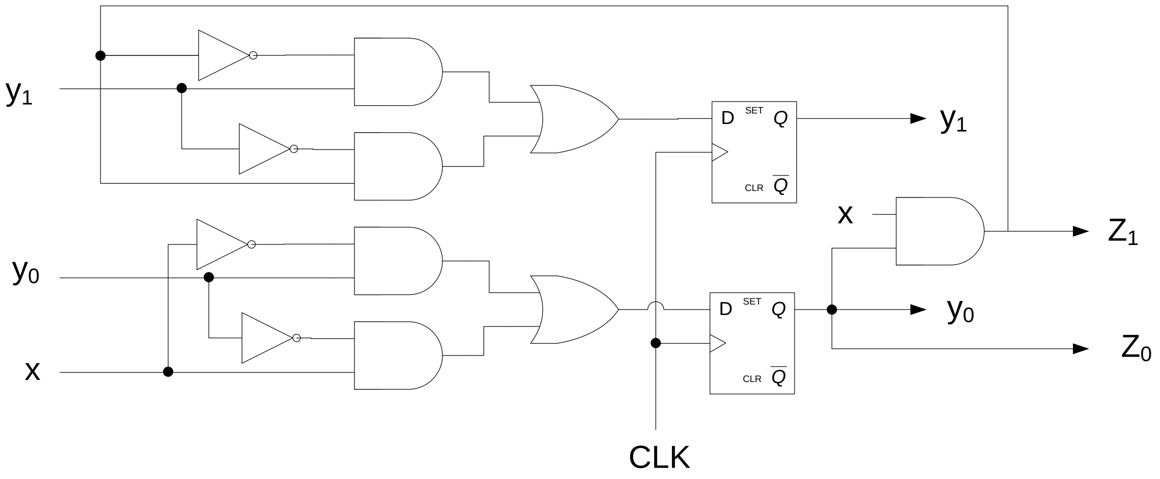
PS	X=0	X=1
S_0	S_1, 0	S_0, 0
S_1	S_2, 0	S_1, 0
S_2	S_1, 0	S_6, 0
S_6	S_1, 0	S_0, 1

Problem 2 *(15 points)*

For the canonical sequential network shown below, determine the following timing factors:

- network setup times: t^x_{su} (net), t^y_{su} (net)
- network hold time: t_h (net)
- network propagation delay: t_p (net)
- minimal period: T_{min}
- maximal clock frequency: f_{max}

for: t_{in} = 2.0ns, t_{out} = 2.5ns, t_p(gate) = 0.5ns, t_{su} (cell) = 1ns, t_p(cell) = 3.0ns and t_h (cell) = 0.5ns.



Using the notation given on Figure 8.15 of the textbook we obtain the following network parameters:

Combinational Networks' delays:

$$\begin{aligned} d1_x &= d1_y = 4 * t_p(\text{gate}) = 2\text{ns} \\ d2 &= 1 * t_p(\text{gate}) = 0.5\text{ns} \end{aligned}$$

Network set-up time:

$$t_{su}^x(\text{net}) = t_{su}^y(\text{net}) = t_{su}(\text{cell}) + d1_x = 1 + 2 = 3\text{ns}$$

Network hold time:

$$t_h(\text{net}) = t_h(\text{cell}) = 0.5\text{ns}$$

Network propagation delay:

$$t_p(\text{net}) = t_p(\text{cell}) + d2 = 3 + 0.5 = 3.5\text{ns}$$

Minimal period:

$$T_{\text{min}} = \max[(t_{in} + t_{su}^x(\text{net})), (t_{su}^y(\text{net}) + t_p(\text{cell})), t_p(\text{net} + t_{out})] = \max[2+3, 3+3, 3.5+2.5] = 6.0\text{ns}$$

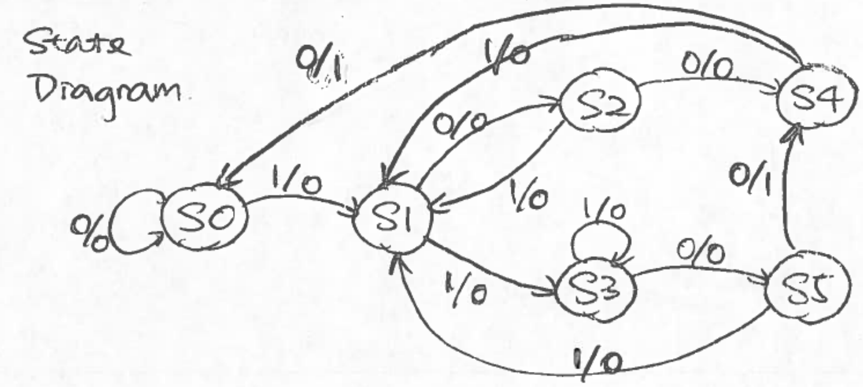
Maximal frequency:

$$f_{\text{max}} = \frac{1}{T_{\text{min}}} = \frac{1}{6.0 \times 10^{-9}} \approx 167\text{MHz}$$

Problem 3 *(15 points)*

Design a pattern recognizer that outputs 1 if the last 4 bits are 1-00, otherwise it outputs 0. Implement it using SR flip-flops and NOR gates.

State Diagram



S0: Init
S1: 1
S2: 1+0
S3: 1+1
S4: 1+0+0
S5: 1+1+0

State Table

PS	X=0	X=1
S0	S0, 0	S1, 0
S1	S2, 0	S3, 0
S2	S4, 0	S1, 0
S3	S5, 0	S3, 0
S4	S0, 1	S1, 0
S5	S4, 1	S1, 0
	NS, Z	

Possible Simplification.

	G1				G2	
P1	(S0	S1	S2	S3)	(S4	S5)
0	1	1	2	2	1	2
1	1	1	1	1	1	1

	G1		G2		G3	G4
P2	(S0	S1)	(S2	S3)	(S4)	(S5)
0	1	2	3	4		
1	1	2	1	2		

P3 (S0)(S1)(S2)(S3)(S4)(S5) ⇒ no simplification needed.

SR Flip Flop.

	SR			
PS	00	01	10	11
0	0	0	1	–
1	1	0	1	–
	NS.			

PS \ NS	0	1
0	0–	10
1	01	–0

Truth Table

	Q(t)				Q(t+1)						
	Q2	Q1	Q0	x	Q2	Q1	Q0	Z	S2R2	S1R1	S0R0
0	0	0	0	0	0	0	0	0	0–	0–	0–
1	0	0	0	1	0	0	1	0	0–	0–	10
2	0	0	1	0	0	1	0	0	0–	10	01
3	0	0	1	1	0	1	1	0	0–	10	–0
4	0	1	0	0	1	0	0	0	10	01	0–
5	0	1	0	1	0	0	1	0	0–	01	10
6	0	1	1	0	1	0	1	0	10	01	–0
7	0	1	1	1	0	1	1	0	0–	–0	–0
8	1	0	0	0	0	0	0	1	01	0–	0–
9	1	0	0	1	0	0	1	0	01	0–	10
10	1	0	1	0	1	0	0	1	–0	0–	01
11	1	0	1	1	0	0	1	0	01	0–	–0

K-maps.

Q2Q1 \ Q0X	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

S2

Q2Q1 \ Q0X	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	–	–	–	–
10	0	0	0	–

S1

Q2Q1 \ Q0X	00	01	11	10
00	0	0	1	1
01	0	0	–	0
11	–	–	–	–
10	0	0	0	0

S0

Q2Q1 \ Q0X	00	01	11	10
00	0	1	–	0
01	0	1	–	–
11	–	–	–	–
10	0	1	–	0

Z

Q2Q1 \ Q0X	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	–	–	–	–
10	1	0	0	1

R2

Q2Q1 \ Q0X	00	01	11	10
00	–	–	–	–
01	0	–	–	0
11	–	–	–	–
10	1	1	1	0

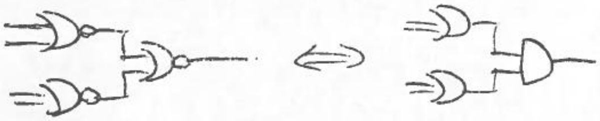
R1

Q2Q1 \ Q0X	00	01	11	10
00	–	–	0	0
01	1	1	0	1
11	–	–	–	–
10	–	–	–	–

R0

Q2Q1 \ Q0X	00	01	11	10
00	–	0	0	1
01	–	0	0	0
11	–	–	–	–
10	–	0	0	1

Two-level NOR-NOR.



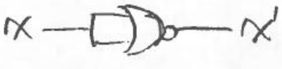
two-level OR-AND
⇒ minPoS.

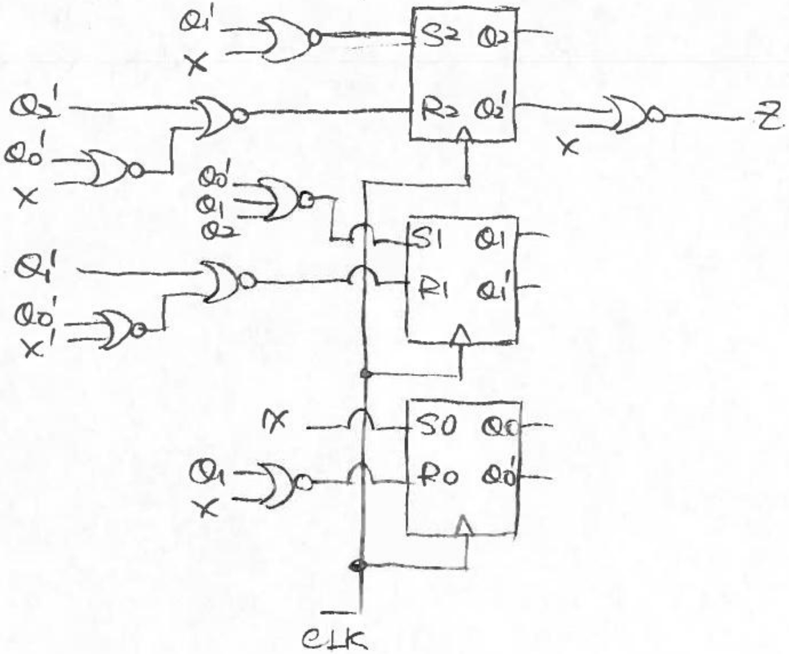
From K-map

$Z = Q_2 \cdot X'$

$$\begin{cases} S2 = Q_1 \cdot X' \\ R2 = Q_2 \cdot (Q_0' + X) \end{cases} \quad \begin{cases} S1 = Q_0 Q_1' Q_2' \\ R1 = Q_1 \cdot (Q_0' + X') \end{cases} \quad \begin{cases} S0 = X \\ R0 = Q_1' \cdot X' \end{cases}$$

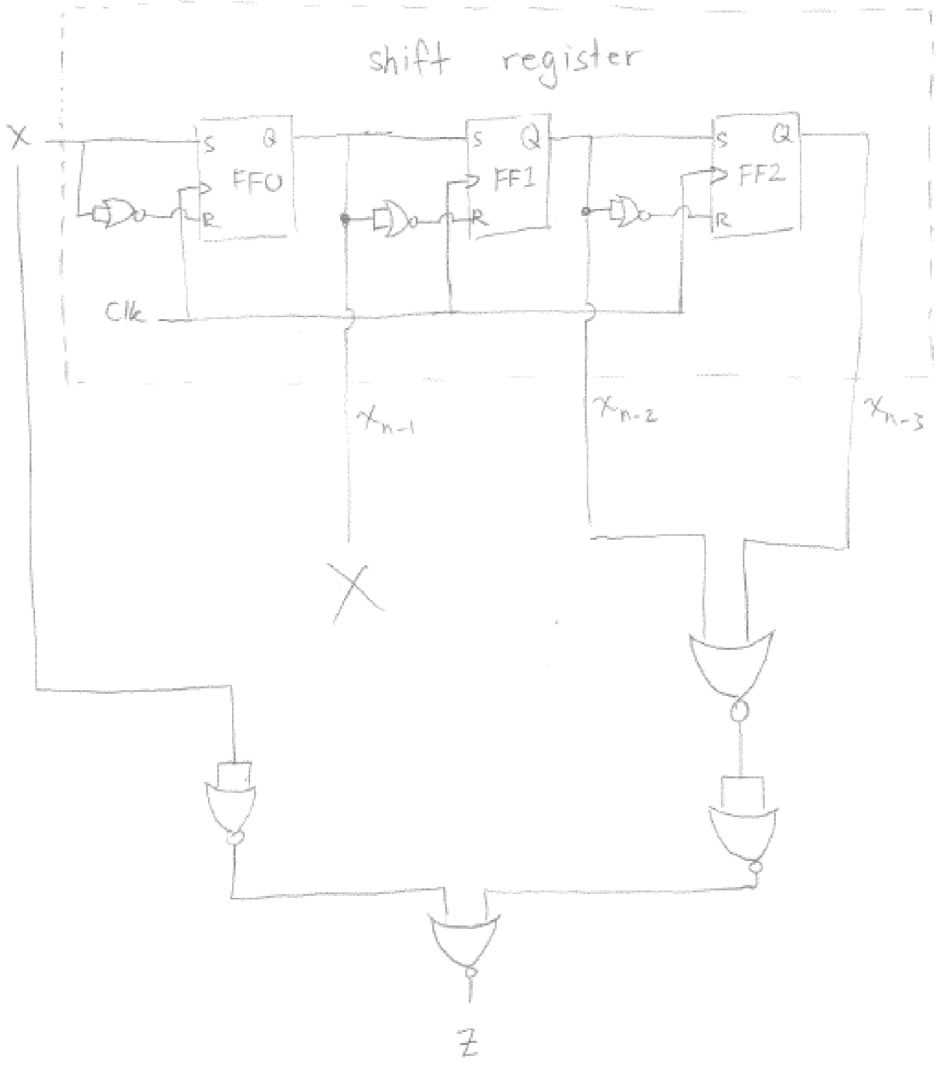
Implementation (complemented inputs are obtained from





Alternate Solution for Prob. 3

Use SR flip-flops to design serial-in parallel-out shift register.

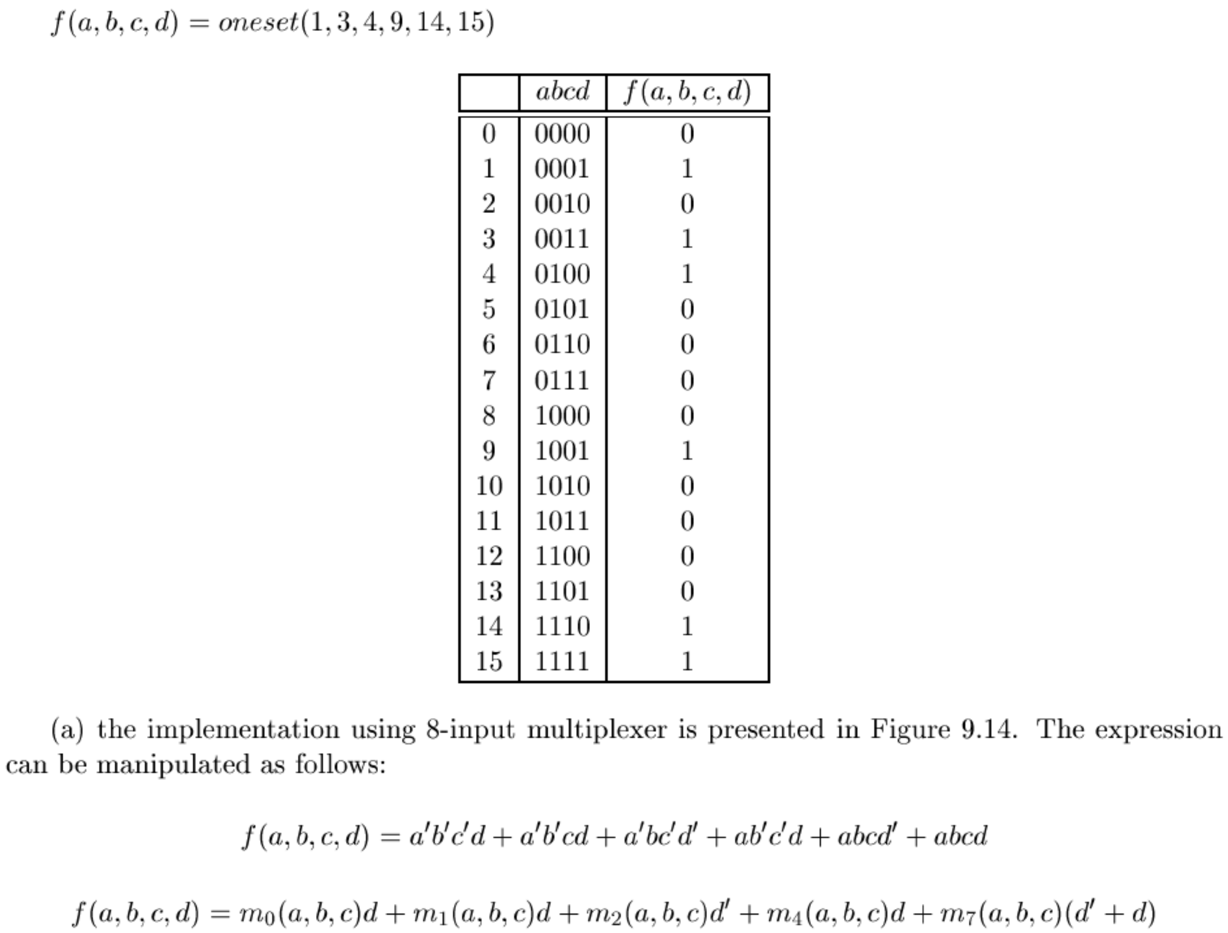


$$Z = x \cdot x_{n-2}' x_{n-3}' = (x' + (x_{n-2} + x_{n-3}))'$$
$$= (x' + (x_{n-2}' x_{n-3}')')' = (x' + ((x_{n-2} + x_{n-3}))')'$$

Problem 4 *(15 points)*

Implement the function $f(a, b, c, d) = one - set(1,3,4,9,14,15)$ using

a. an eight-input multiplexer;
b. a four-input multiplexer and NOR gates (use inputs *a* and *b* as select inputs to the multiplexer, the NOR gates for functions $f(0,0, c, d)$, $f(0,1, c, d)$, $f(1,0, c, d)$ and $f(1,1, c, d)$, and connect the outputs of these networks to the corresponding data input of the multiplexer).



$f(a, b, c, d) = oneset(1, 3, 4, 9, 14, 15)$

	$abcd$	$f(a, b, c, d)$
0	0000	0
1	0001	1
2	0010	0
3	0011	1
4	0100	1
5	0101	0
6	0110	0
7	0111	0
8	1000	0
9	1001	1
10	1010	0
11	1011	0
12	1100	0
13	1101	0
14	1110	1
15	1111	1

(a) the implementation using 8-input multiplexer is presented in Figure 9.14. The expression can be manipulated as follows:

$$f(a, b, c, d) = a'b'c'd + a'b'cd + a'bc'd' + ab'c'd + abcd' + abcd$$

$$f(a, b, c, d) = m_0(a, b, c)d + m_1(a, b, c)d + m_2(a, b, c)d' + m_4(a, b, c)d + m_7(a, b, c)(d' + d)$$

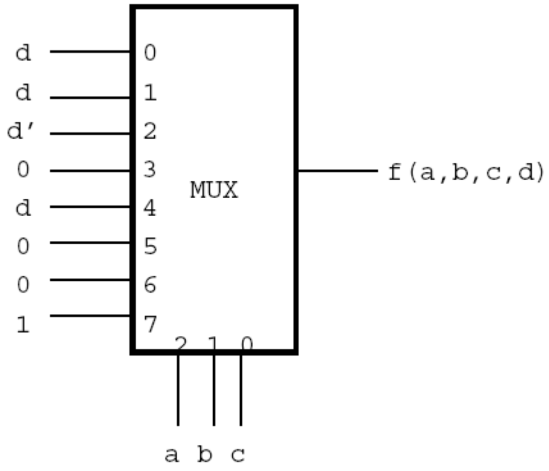


Figure 9.14: Implementation for Exercise 9.15 (a)

(b) the implementation using 4-input multiplexer is presented in Figure 9.15. The expression for this implementation is:

$$f(a,b,c,d) = m_0(a,b)(c'd + cd) + m_1(a,b)c'd' + m_2(a,b)c'd + m_3(a,b)(cd + cd')$$

$$f(a,b,c,d) = m_0(a,b)d + m_1(a,b)(c+d)' + m_2(a,b)(c+d')' + m_3(a,b)c$$

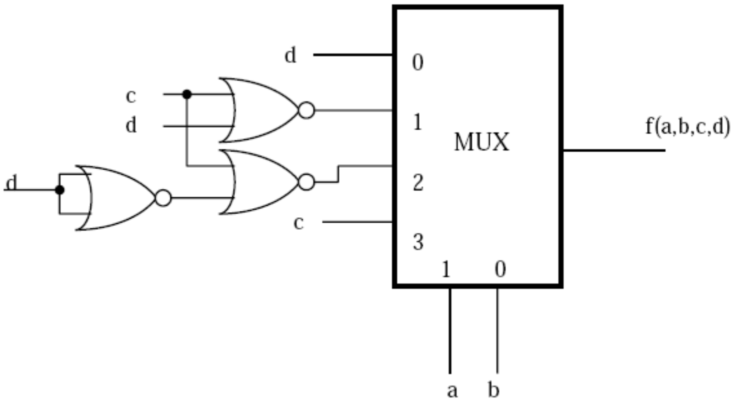
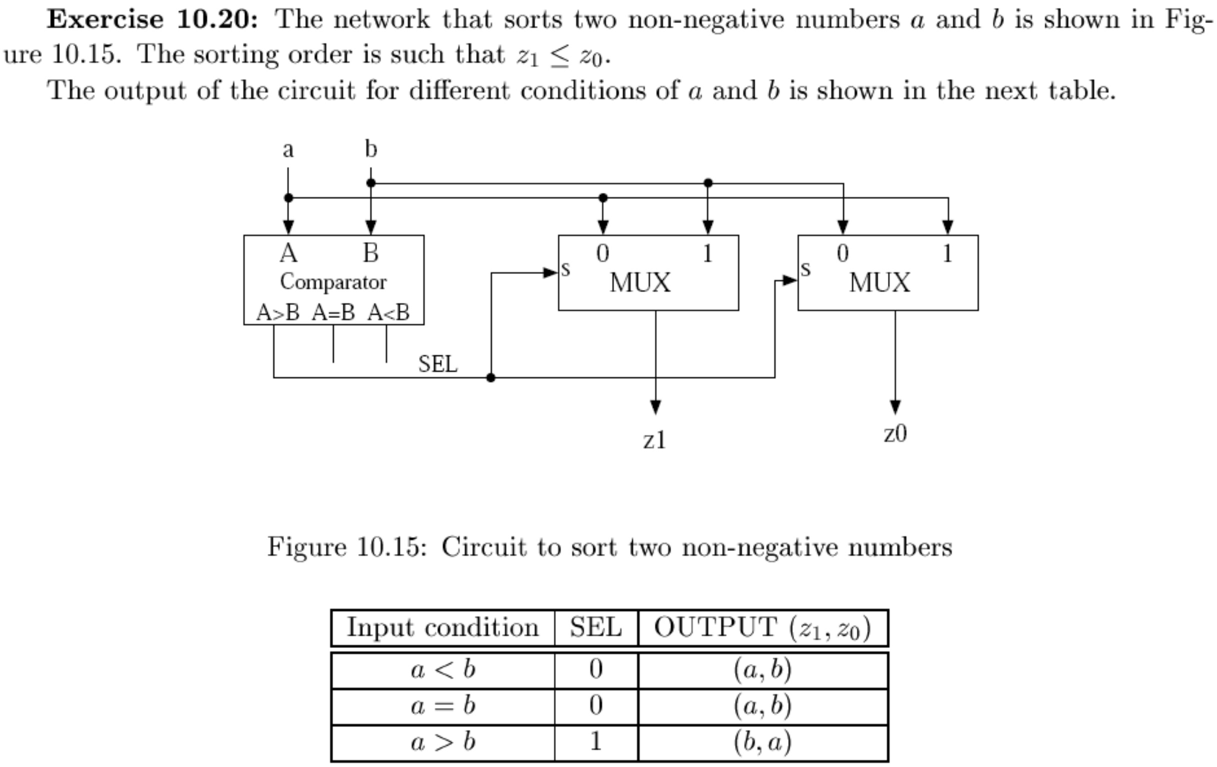


Figure 9.15: Network for Exercise 9.15 (b)

Problem 5 *(15 points)*

Design a network that sorts two nonnegative integers *a* and *b*. Each integer is represented by four bits. You may use only the following modules: 4 x 2-input multiplexers and four-bit comparators. Indicate all inputs on the modules being used.



Exercise 10.20: The network that sorts two non-negative numbers a and b is shown in Figure 10.15. The sorting order is such that $z_1 \leq z_0$.

The output of the circuit for different conditions of a and b is shown in the next table.

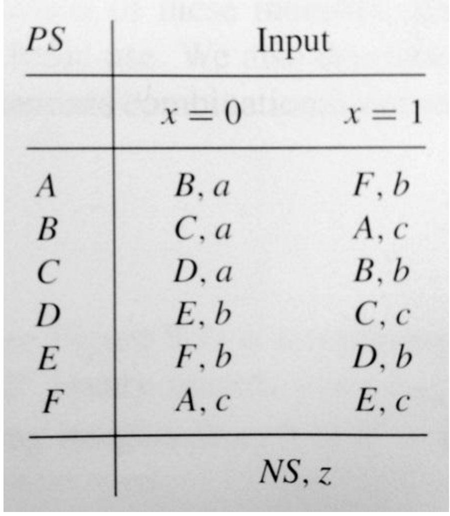
a
b
A
B
Comparator
A>B
A=B
A<B
SEL
s
0
1
MUX
s
0
1
MUX
z1
z0

Figure 10.15: Circuit to sort two non-negative numbers

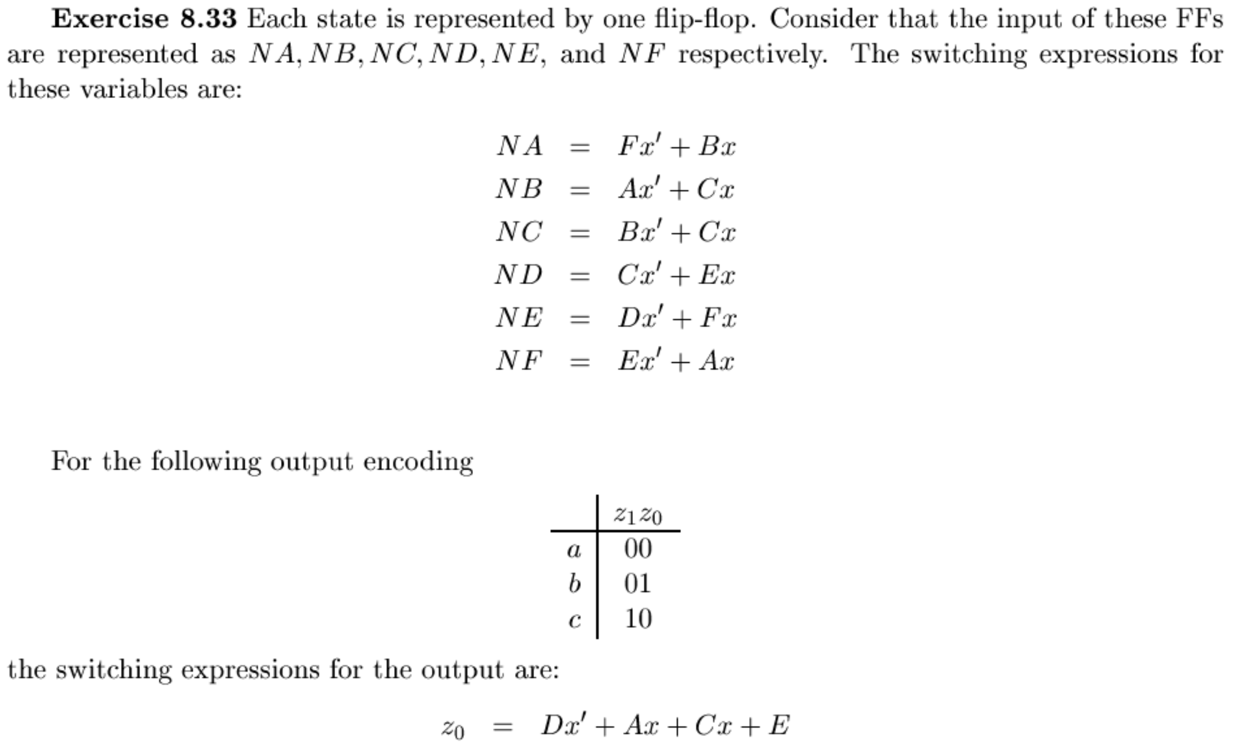
Input condition	SEL	OUTPUT (z_1, z_0)
$a < b$	0	(a, b)
$a = b$	0	(a, b)
$a > b$	1	(b, a)

Problem 6 *(15 points)*

Design a sequential network described by the following state/output table, using the “one flip-flop per state” approach.



PS	Input	
	$x = 0$	$x = 1$
A	B, a	F, b
B	C, a	A, c
C	D, a	B, b
D	E, b	C, c
E	F, b	D, b
F	A, c	E, c
	NS, *z*	



Exercise 8.33 Each state is represented by one flip-flop. Consider that the input of these FFs are represented as NA, NB, NC, ND, NE, and NF respectively. The switching expressions for these variables are:

$$\begin{aligned}
NA &= Fx' + Bx \\
NB &= Ax' + Cx \\
NC &= Bx' + Cx \\
ND &= Cx' + Ex \\
NE &= Dx' + Fx \\
NF &= Ex' + Ax
\end{aligned}$$

For the following output encoding

	z_1z_0
a	00
b	01
c	10

the switching expressions for the output are:

$$z_0 = Dx' + Ax + Cx + E$$

$$z_1 \quad = \quad Bx + Dx + F$$

The network that corresponds to these expressions is shown in Figure 8.38.

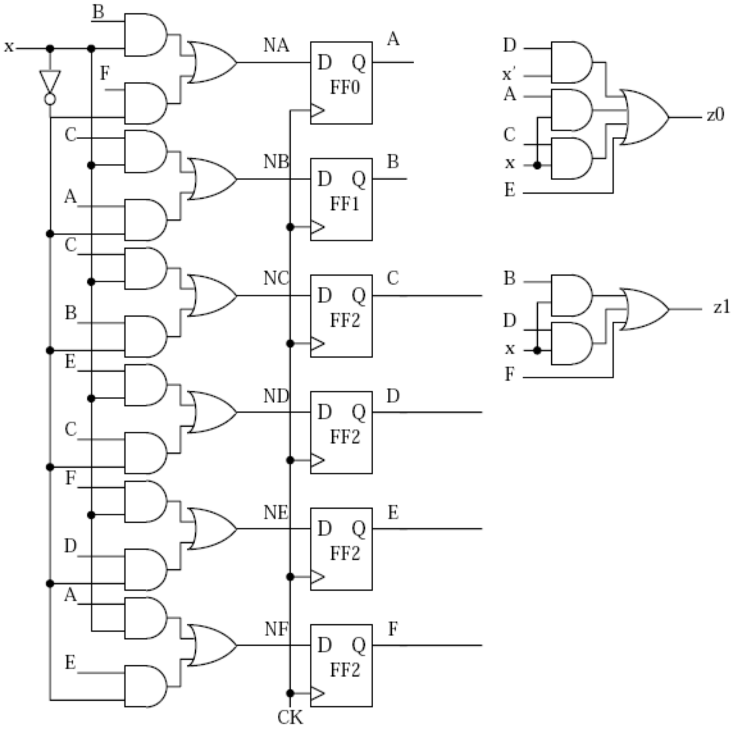


Figure 8.38: Network for Exercise 8.33

Problem 7 *(15 points)*

Using a modulo-16 binary counter with parallel inputs, implement

a. a 4-to-11 counter;
b. a modulo-13 counter;
c. a counter with the following periodical sequence: 0,1,2,3,4,5,8,9,10,11,14,15

Exercise 11.11: Using a modulo-16 binary counter with parallel inputs
(a) 4-to-11 counter.

$$\begin{aligned} CNT &= x \\ LOAD &= Q_3Q_1Q_0x = TC \\ I &= 0100 \end{aligned}$$

The network is shown in Figure 11.14.

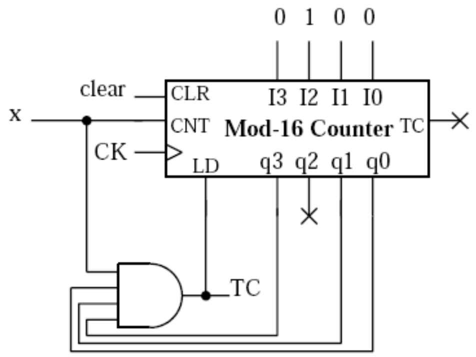


Figure 11.14: 4-to-11 counter - Exercise 11.11(a)

(b) modulo-13 counter.

$$\begin{aligned} CNT &= x \\ LOAD &= Q_3Q_2x = TC \\ I &= 0000 \end{aligned}$$

The network for this counter is shown in Figure 11.15.

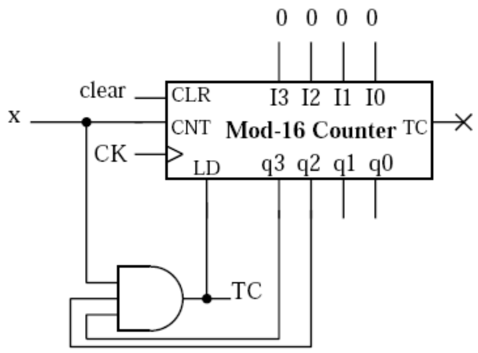


Figure 11.15: Modulo-13 counter - Exercise 11.11(b)

(c) The state diagram is shown in Figure 11.16, with notes associated to the transitions for which the counter must be loaded. From the diagram we obtain the condition for loading on a Kmap as follows:

Q_0

0	0	0	0
0	1	-	-
-	-	0	0
0	0	1	0

Q_2

Q_3

Q_1

that results in the minimal expression:

$$LOAD = (Q_3 Q_2' Q_1 Q_0 + Q_3' Q_2 Q_0)x$$

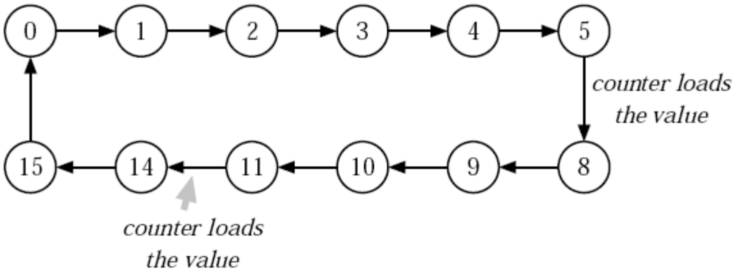


Figure 11.16: State diagram for Exercise 11.11(c)

We use the fact that LOAD overides CNT input to define:

$$CNT = x$$

Considering the inputs I_i as d.c.s for the cases when $LOAD = 0$ and the appropriate next state value for when $LOAD = 1$, and using Kmaps we obtain the expressions:

$$\begin{aligned} I_3 &= 1 \\ I_2 &= I_1 = Q_1 \\ I_0 &= 0 \end{aligned}$$

The network for this system is shown in Figure 11.17.

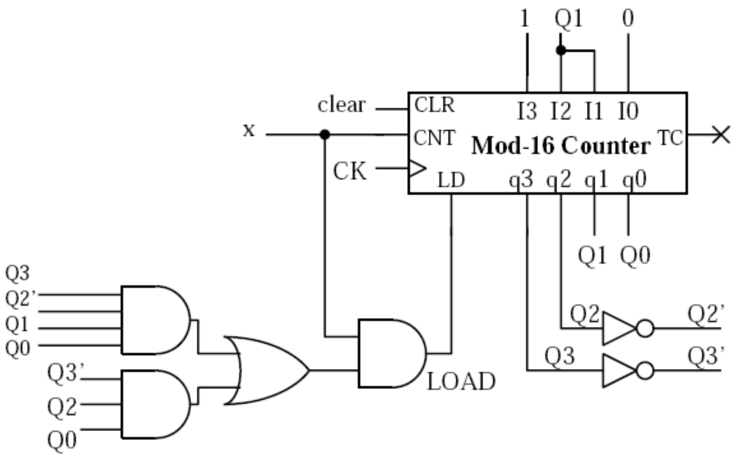


Figure 11.17: Network for Exercise 11.11(c)