Embedded Real-Time Systems (AME 3623)
Homework 1 Solutions

February 27, 2005

Question 1

Consider the following function:

A	B	C	D	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

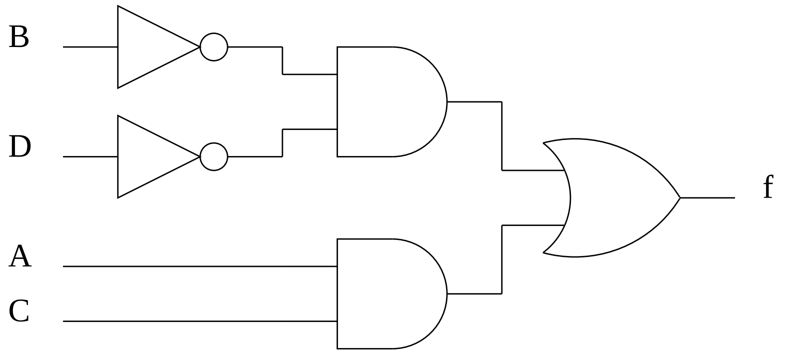
1. (10pts) Show the corresponding Karnaugh map and a set of covering clusters.

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	1	1
10	1	0	1	1

2. (10pts) What is the algebraic description of the reduced circuit?

   $f = \bar{D}\bar{B} + AC$

3. (10pts) Show the reduced circuit.



4. (10pts) Starting with the original algebraic description of the function ($A'B'C'D' + A'B'CD' + ...$), use the definitions and identities discussed in class to prove that it is equivalent to the reduced description that you gave above.

$$\bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D}$$
$$+ABC\bar{D} + A\bar{B}CD + ABCD$$

$$\bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D}$$

$$+A\bar{B}C\bar{D}+ABC\bar{D}+A\bar{B}CD+ABCD \qquad X=X+X$$

$$\bar{B}\bar{D}(\bar{A}\bar{C}+A\bar{C}+\bar{A}C+AC)$$
$$+AC(\bar{B}\bar{D}+B\bar{D}+\bar{B}D+AC) \qquad \text{Distributive Law}$$

$$\bar{B}\bar{D}(1)+AC(1) \qquad X*1=X \text{ and Distributive Law}$$

$$\bar{B}\bar{D}+AC \qquad X*1=X$$

Question 2

Consider the following function:

A	B	C	D	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

1. (10pts) Show the corresponding Karnaugh map and a set of covering clusters.

A

B

	CD \ AB	00	01	11	10
	00	1	0	0	1
D	01	0	0	1	1
D, C	11	0	1	1	1
C	10	0	0	1	1

2. (10pts) What is the algebraic description of the reduced circuit?

   $f = AC + DA + BCD + \bar{B}\bar{C}\bar{D}$

   or

   $f = A(C + D) + BCD + \bar{B}\bar{C}\bar{D}$

Question 3

Consider the following function:

A	B	C	D	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

1. (10pts) Show the corresponding Karnaugh map and a set of covering clusters.

   There are several equivalent coverings of the '1's. Here is one:

				A	
	AB		B		
CD		00	01	11	10
	00	0	1	1	1
	01	1	1	1	1
D	11	1	1	0	1
C	10	1	1	1	1

   This solution involves six clusters. However, there is a much simpler solution if we focus on the '0's:

					A	
		AB		B		
	CD		00	01	11	10
		00	0	1	1	1
		01	1	1	1	1
	D	11	1	1	0	1
C		10	1	1	1	1

2. (10pts) What is the simplest algebraic description of the reduced circuit?

   $C\bar{A} + \bar{C}A + C\bar{D} + \bar{C}D + A\bar{B} + \bar{A}B$ (6 pts)

   which simplifies to:

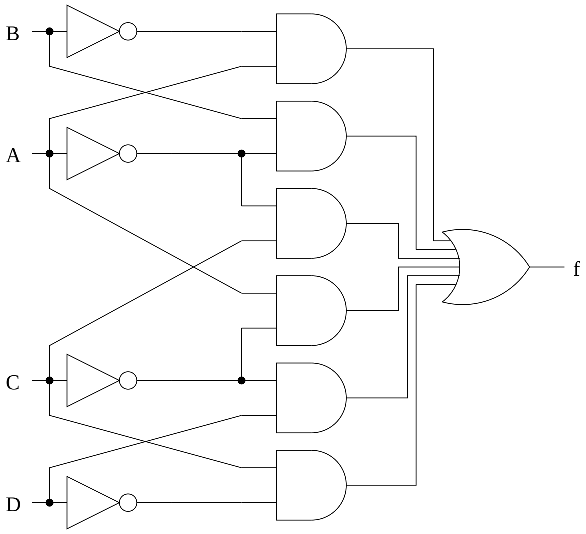
   $(A \bigoplus C) + (C \bigoplus D) + (A \bigoplus B)$ (8 pts)

   But, if we use the modified Karnaugh map, we have:

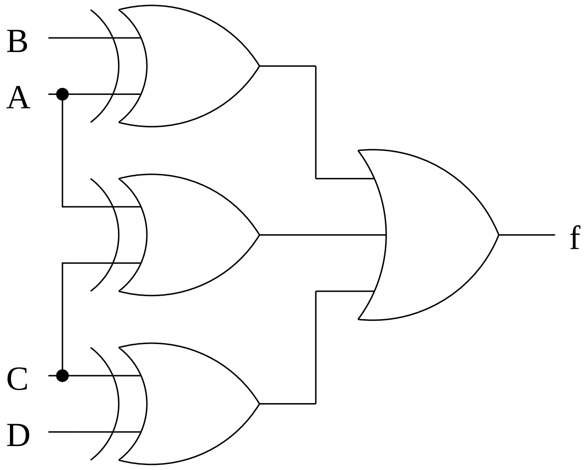
   $\overline{(\bar{A}\bar{B}\bar{C}\bar{D} + ABCD)}$ (10 pts)

3. (10pts) Show the reduced circuit.

Solution 1:

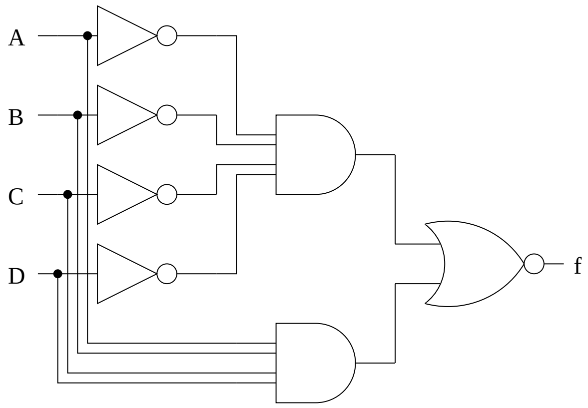


Solution 2:



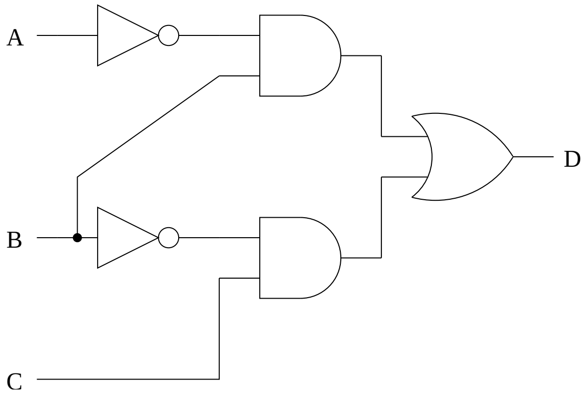
Note: 2-input XOR gates are constructed from 2 AND and 1 OR gate (so this circuit is more expensive to implement than solution 3).

Solution 3:

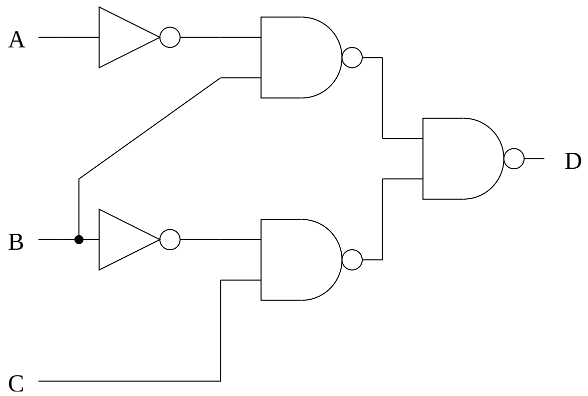


Question 4

From the perspective of the transistor count, it is cheaper to implement a NAND gate than an AND gate (we typically implement an AND gate as a NOT - NAND). Consider the following circuit:



1. (10pts) Redesign this circuit such that it implements the same function, but uses only NAND and NOT gates. Show this new circuit.



2. (10pts) What can you generally conclude about functions that involve a set of AND gates that then provide inputs to an OR gate? Explain.

DeMorgan's law says that we can convert an OR gate into a NOT-AND gate with "NOTed" inputs. These latter NOTs can be moved to the AND gates, converting them to NAND gates. In general, any cascade of AND to OR gates can be converted to a cascade from NAND to NAND without any other changes to the circuit.

Question 5

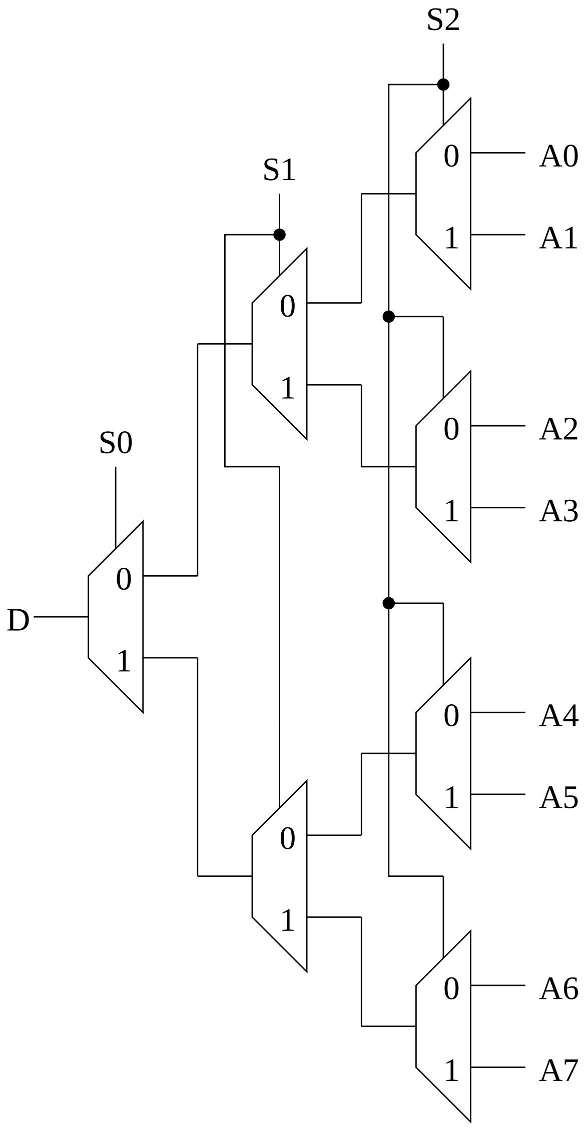
(10pts) Using the algebraic definitions and identities discussed in class, prove the following:

$\overline{(\bar{A}+\bar{B}+\bar{C})} = ABC$

$$
\begin{aligned}
\overline{(\bar{A}+\bar{B}+\bar{C})} & \\
\overline{((\bar{A}+\bar{B})+\bar{C})} & \quad \text{Associative Law} \\
\overline{(\overline{AB}+\bar{C})} & \quad \text{DeMorgan's Law; X} = \overline{\overline{\text{X}}} \\
\overline{(\overline{ABC})} & \quad \text{DeMorgan's Law; X} = \overline{\overline{\text{X}}} \\
ABC & \quad \text{X} = \overline{\overline{\text{X}}}
\end{aligned} \tag{1}
$$

Question 6

(10pts) Using two-output demultiplexers (1 select line) and the basic gates (AND, OR, NOT, NOR, NAND, and XOR), design an eight-output demultiplexer (three select lines).



Question 7

(10pts) The sequential counter that we designed in class using three D-type flip-flops counts from zero to seven, and then rolls over to zero again. Modify this design such that the counter rolls over to zero after reaching six.

We want our counter to produce the following sequence:

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ & \vdots & \end{array}$$

This behavior is the same as that of our 3-bit synchronous counter that we designed in class – except in the transition from 110 to 000. (Recall that we wired our synchronous counter such that X0 changed state on every clock tick; X1 changed state on every clock tick for which X0 was high; and X2 changed state on every clock tick for which both X0 and X1 were high). Our approach will be to first detect the one exceptional case (110). This is accomplished with a sub-circuit that detects this case:

$R = \overline{X0} * X1 * X2.$

Note that since it is never the case that our counter will be in the state 111, we can simplify this to:

$R = X1 * X2.$

When we are in this exceptional condition, we want our circuit to do the following:

- For $X0$: we want to withhold the clock (so that it does not change state).
- For $X1$ and $X2$, we want the clock to be allowed to the corresponding flip-flops so that both will change state.

So: the clock input for flip-flop 0 ($CLK0$) should be as follows:
$CLK0 = \overline{R} * CLK$

For flip-flop 1, the situation is more complex. Here is the truth table for $CLK1$:

R	X0	CLK	CLK1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	x
1	1	1	x

The last two cases are "don't cares" because these cases will never occur (i.e., $R * X0$ will never be true). The corresponding Karnaugh map is:

$X0,\ CLK$ / R	00	01	11	10
			X0	
		CLK		
0	0	0	1	0
1	0	1	x	x

So: $CLK1 = CLK * X0 + CLK * R = CLK * (X0 + R)$

For CLK2, we must also consider the state of X1 (on top of X0, R, and CLK). Here is the truth table:

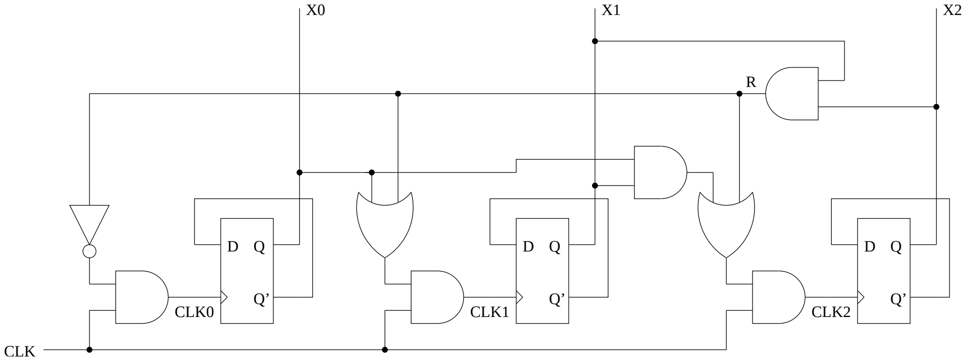
R	X1	X0	CLK	CLK2
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	x
1	0	0	1	x
1	0	1	0	x
1	0	1	1	x
1	1	0	0	0
1	1	0	1	1
1	1	1	0	x
1	1	1	1	x

Again, the “don’t care” cases correspond to those cases that cannot occur. Here is the Karnaugh map:

	$X0,\ CLK$		CLK	X0	
$R,\ X1$		00	01	11	10
	00	0	0	0	0
X1	01	0	0	1	0
R	11	0	1	x	x
	10	x	x	x	x

So: $CLK2 = CLK * X0 * X1 + CLK * R = CLK * (X0 * X1 + R)$

Putting it all together, here is the resulting circuit:



Question 8

How much time did you spend on this homework assignment?