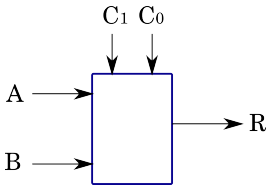
Coursework 1: Combinatorial Circuit Design



The coursework is a hardware design exercise in which you will design a combinatorial circuit and test it with a very simple simulator called Digisim.

The Problem

Design a four-input, one-output digital circuit which operates as a controlled two-bit Boolean function generator. Two of the inputs (C_1, C_0) are used as control inputs and thus can select one of four different Boolean functions of the other two inputs (A, B).

There is a list of student names on the course web page and each has a unique number. If your name does not appear on the list please send me an email and I will issue you a number. Convert your number to base 4 by looking it up in the table given at the end of this handout. Select your functions according to the table below given that the digits of your base four number are $\mathbf{D_3D_2D_1D_0}$. The operator $\oplus$ stands for the XOR (Exclusive OR) operation.

		$\mathbf{C_1C_0}$			
		00	01	10	11
$\mathbf{D_3}$	0	$A \oplus B$			
	1	$(A \oplus B)'$			
	2	1			
	3	0			
$\mathbf{D_2}$	0		A		
	1		A'		
	2		B'		
	3		B		
$\mathbf{D_1}$	0			$A \cdot B$	
	1			$A \cdot B'$	
	2			$A' \cdot B'$	
	3			$A' \cdot B$	
$\mathbf{D_0}$	0				$A + B$
	1				$A' + B$
	2				$A + B'$
	3				$A' + B'$

For example, if my number is 125, then the base-4 equivalent is equal to 1331, thus $\mathbf{D_3} = 1$, $\mathbf{D_2} = 3$, $\mathbf{D_1} = 3$, and $\mathbf{D_0} = 1$. Using these digit values and the above table I get my functions:

Digits	C_1	C_0	Output
$\mathbf{D_3} = 1$	0	0	$(A \oplus B)'$
$\mathbf{D_2} = 3$	0	1	B
$\mathbf{D_1} = 3$	1	0	$A' \cdot B$
$\mathbf{D_0} = 1$	1	1	$A' + B$

NB If due to an oversight you have been allocated number 125 please don't copy out this solution and hand it in but email me and I'll allocate you a different number!

Design Procedure

Once the functions are known, the truth table can be generated and the Karnaugh Map constructed. For the example above this is:

C_1	C_0	A	B	R	Function
0	0	0	0	1	$(A \oplus B)'$
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	1	
0	1	0	0	0	B
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	0	$A' \cdot B$
1	0	0	1	1	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	1	$A' + B$
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	1	

The Karnaugh map can be filled out now (carefully!) and the Boolean expression minimized.

C1,C0 \ A,B	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	1	1	1	0
10	0	1	0	0

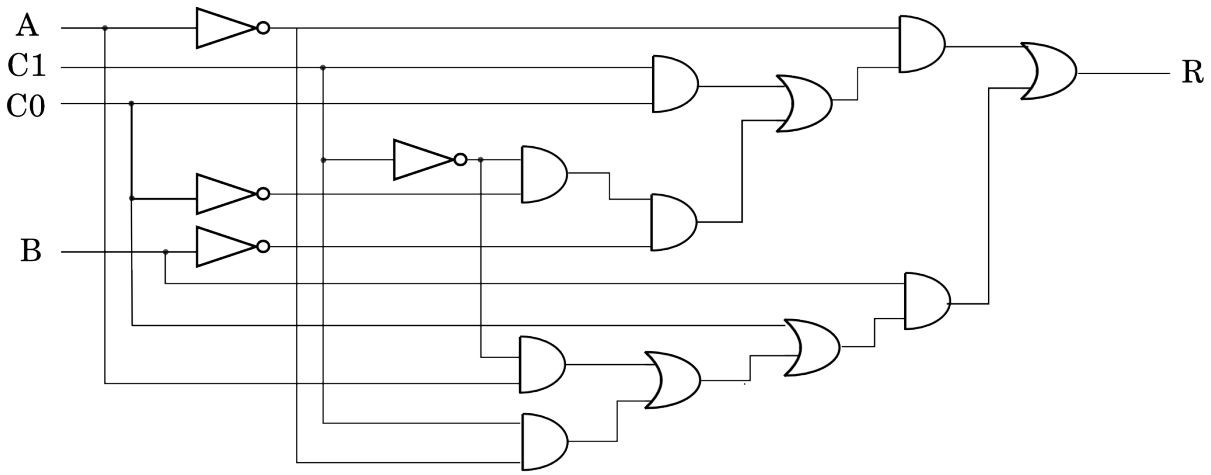
In my case I ended up with a reasonably difficult circuit. There is one term with four, three terms with three, and one term with two literals. All input signals will have to be inverted. The minimized expression is:

$$C_1' \cdot C_0' \cdot A' \cdot B' + C_1' \cdot A \cdot B + C_1 \cdot C_0 \cdot A' + C_1 \cdot A' \cdot B + C_0 \cdot B$$

and factoring it one way, I get:

$$B \cdot (C_0 + C_1' \cdot A + C_1 \cdot A') + A' \cdot (C_1' \cdot C_0' \cdot B' + C_1 \cdot C_0)$$

And the direct circuit implementation is:

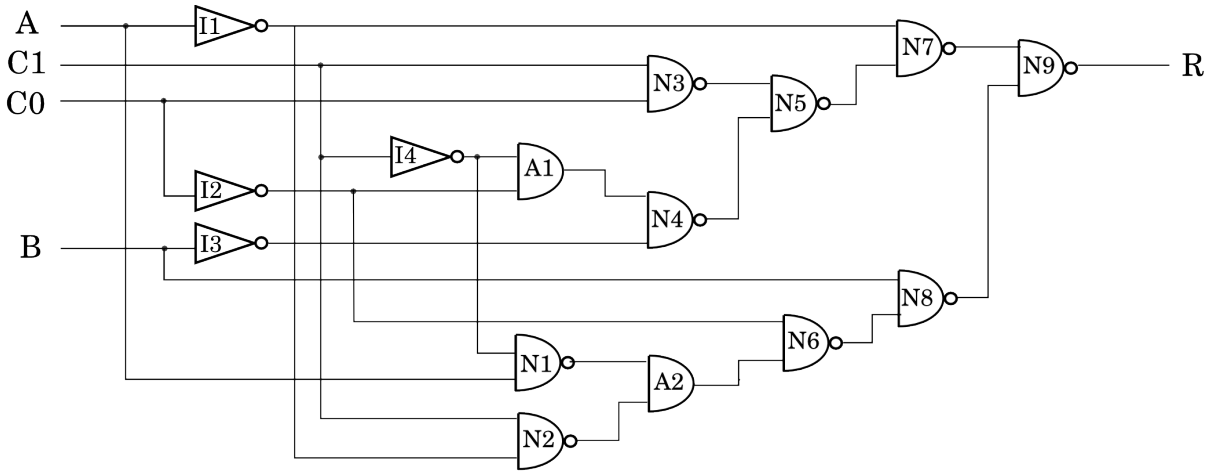


In professional digital circuit design one major objective is to minimise the area of silicon that is needed. In fact Karnaugh map minimisation only partially solves this problem by minimising the number of gates. In practice not all gates take up the same amount of space. In particular AND and OR gates take up more space than NAND and NOR. The table below shows nominal sizes for one/two input gates which are approximately in the size ratios found in practical silicon chips:

Gate	Nominal Size
INVERTER	3
NAND, NOR	4
AND, OR	6
XOR, XNOR	8

We can use these to compare different implementations of our circuit. In the circuit above I have four INVERTERs and 11 AND/OR gates so my size count is 12+66=78.

One way I can try to improve this is by converting the ANDs and ORs to NANDs and NORs using de Morgan's theorem, as we discussed in lecture 4. However, I have another constraint that I need to consider, and that is that I want the delay through the circuit, from inputs to output to be as small as possible. The delay through any one gate is not constant, but depends on the number of gates it is connected to. However, as a simple approximation we will try to minimise the number of gates on the longest path between input and output. So, first off I'll try converting all the OR gates into NAND gates, and I get this circuit:



The circuit has four INVERTERs, two ANDs and nine NANDs so the score is 12 + 12 + 26 = 60, and I have made a considerable saving in silicon area. There are no obvious further improvements. If I convert gates A1 and A2 to NORs the extra two inverters required would increase the area slightly and would also increase the number of gates on the longest path from input to output thus slowing down the circuit. Overall this looks like the best I can achieve without a lot of further work. Now I need to test the circuit.

Creating a wiring list in an xml format.

Digisim is a simple, in house simulator of digital circuits supporting INVERTERs and two input AND, NAND, OR, NOR, XOR and XNOR gates. We need to create a wiring list of the design to input into Digisim. Each wire in the circuit must have a unique name. In this context we use the word "wire" to include all connected branches of a path joining up the gates. We could do this in any way we choose, but to be consistent we will first label all the gates, and then name each wire after the gate from which it originates, or from the input from which it originates. The only exception we will make will be to label the output wire R. I have used a very simple naming convention for the gates/wires on the diagram above (I for INVERTER, N for NAND and A for AND).

The gates are written as functions in the wiring list, and in every case the first argument is the name of the output wire, and the next arguments are the input wires. Thus the top left inverter gate is written INVERTER(I1,A). The wiring list format is self-explanatory. You can write comments at the head or in the body of the design, but not between the pairs of matching tags: <number> </number>; < inputs> </inputs>;< outputs> </outputs> or <gates> </gates>. Anything other than a letter or a digit can be used as a white space. My wiring list is:

```
// My solution to the Hardware Coursework.
// My number is 125 or 1331 in base 4.
// My functions are: (00)=(A eor B)' (01)=B (10)=A'B (11)=A'+B
<circuit>
<number> 125 </number>
<inputs> C1 C0 A B </inputs>
<outputs> R </outputs>
<gates>
    INVERTER(I1, A)
    INVERTER(I2, C0)
    INVERTER(I3, B)
    INVERTER(I4, C1)
    AND(A1, I2, I4)
    AND(A2, N1, N2)
    NAND(N1, I4, A)
    NAND(N2, I1, C1)
    NAND(N3, C1, C0)
    NAND(N4, A1, I3)
    NAND(N5, N3, N4)
    NAND(N6, I2, A2)
    NAND(N7, I1, N5)
    NAND(N8, B, N6)
    NAND(R, N7, N8)
</gates>
</circuit>
```

You can download the java code for Digisim from the course web-page along with my wiring list to check it out. To run it under Windows or Linux you can open a console or terminal (in windows: start → programs → accessories → command prompt, in Linux: Applications → Accessories → Terminal) change to the directory where you have stored Digisim and issue the following commands:

```
>javac Digisim.java
>java Digisim
```

You can also use a more comprehensive java development tool if you wish. Netbeans and Eclipse are available on the laboratory systems. These are useful when you need to write large programmes, but are not necessary for this exercise. When Digisim is running you can open a circuit wiring list in the above text format. It will either report errors to you or print the truth table and the expected output for your function. When you are satisfied that your circuit is correct, (or you have had enough and want to give up!) use the save report option to generate a text file which you should print and hand in with your report.

Once successfully tested you could send your design to a silicon foundry and get the circuit manufactured - but that would be expensive, and you won't get any extra marks for doing so!

Submission

You are required to submit a wiring list of your design, a paper copy of a written report and a paper copy of the Digisim test report. The written report should have a short description of your problem, your design procedures including all the circuits you tried, and your final circuit diagram clearly labeled (ideally this should not exceed four sides of A4). A good grade is given for one working circuit, an excellent one for attempts at minimization of silicon by trying more than one circuit. This piece of work counts for 60% of the hardware continuous assessment.

Electronic: A text file named hardware.txt containing your wiring list.
Paper: Your report of about 4 pages and a print out of the Digisim report.

Conversion of Decimal Numbers to Base 4

	0	1	2	3	4	5	6	7	8	9
000	0000	0001	0002	0003	0010	0011	0012	0013	0020	0021
010	0022	0023	0030	0031	0032	0033	0100	0101	0102	0103
020	0110	0111	0112	0113	0120	0121	0122	0123	0130	0131
030	0132	0133	0200	0201	0202	0203	0210	0211	0212	0213
040	0220	0221	0222	0223	0230	0231	0232	0233	0300	0301
050	0302	0303	0310	0311	0312	0313	0320	0321	0322	0323
060	0330	0331	0332	0333	1000	1001	1002	1003	1010	1011
070	1012	1013	1020	1021	1022	1023	1030	1031	1032	1033
080	1100	1101	1102	1103	1110	1111	1112	1113	1120	1121
090	1122	1123	1130	1131	1132	1133	1200	1201	1202	1203
100	1210	1211	1212	1213	1220	1221	1222	1223	1230	1231
110	1232	1233	1300	1301	1302	1303	1310	1311	1312	1313
120	1320	1321	1322	1323	1330	1331	1332	1333	2000	2001
130	2002	2003	2010	2011	2012	2013	2020	2021	2022	2023
140	2030	2031	2032	2033	2100	2101	2102	2103	2110	2111
150	2112	2113	2120	2121	2122	2123	2130	2131	2132	2133
160	2200	2201	2202	2203	2210	2211	2212	2213	2220	2221

Note on Digisim

The current version of Digisim was written in 2009 and is quite stable, but if you discover any bugs please let me know. One known bug is that occasionally on start up it does not display the menu bar. If this happens just close it by typing ctrl-C at the console and then re-start it. I have not been able to find out why this happens.

Mark Scheme.

This coursework is marked out of 60. A basic mark of 40 will be awarded to any student who completes the exercise correctly. This will be verified by looking at the Digisim truth table, where the expected column should agree exactly with the result (R) column. If the two do not agree exactly, but the circuit is otherwise correct, there is almost certainly a fault in the wiring list. Five marks will be deducted for this case. The markers may test the wiring list submitted through CATE with Digisim to check that the circuit is correct.

Up to 10 extra marks can be awarded for extra work. Examples are:

1. Doing a maxterm implementation as well as the minterm one,
2. Factorising the equations in different ways after using the K-map,
3. Organising the circuit to minimise space or maximise speed,

Five marks are awarded for the quality of the report. Finally up to 5 marks can be awarded for any work showing exceptional merit or originality. No extra credit is given for particularly small circuits as each problem is different.

If the exercise has not been completed marks will be awarded according to the following guidelines:

1. Correctly using the K-map and finding the equation for R gains a pass mark of 24
2. Completing a largely correct wiring list with minor errors gains up to 30