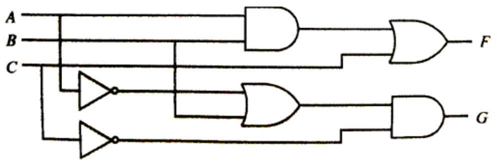
Homework Lecture 1

Exercise 1 (Book Exercise A.1)

Show the logic diagram for an AND gate implemented entirely with NAND gates.

Exercise 2.(Book Exercise A.4)

Construct the truth table for the circuit below.



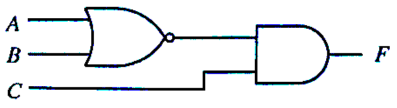
Exercise 3. (Book Exercise A.7)

$f(A,B,C) = A \cdot B \cdot C + \bar{A} \cdot B \cdot \bar{C}$

$g(A,B,C) = (A \oplus C)B$

Are functions *f* and *g* equivalent?

Exercise 4. (Book Exercise A.8)



Give the Boolean equation of this circuit in SOP form.

Exercise 5. (Book Exercise A.14)

Draw a logic diagram that uses a decoder and two OR gates to implement function F and G.

$F(A,B,C) = \bar{A} \cdot B \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C$

$G(A,B,C) = \bar{A} \cdot B \cdot \bar{C} + A \cdot B \cdot C$

Exercise 6.

A property of the GRAY code is that there is exactly 1 bit change between two consecutive numbers.
The GRAY code is also used in the Karnaugh map.
Give the sequence of sixteen consecutive GRAY numbers. Start with number 0000.

Exercise 7. (Book Exercise A.37a) (SOP only)

Use Karnaugh map to simplify function *f* using SOP form.

$$f(A,B,C,D) = \sum m(2,8,10,11) + \sum_d m(0,9)$$

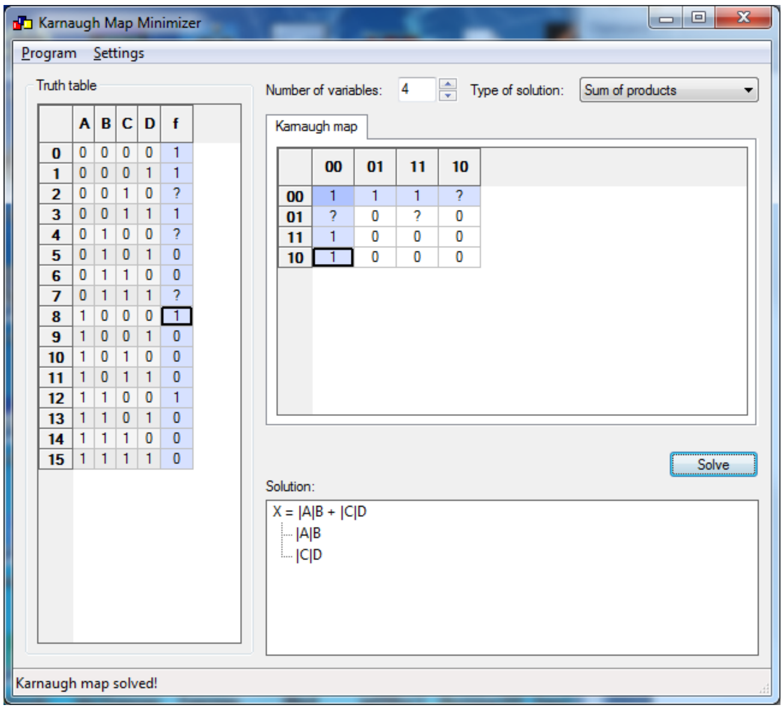
Note: the $\sum$ indicates that it is SUM of minterms (‘m’). The ‘m’ is not always in the specification. I.e. equivalent is:

$$f(A,B,C,D) = \sum (2,8,10,11) + \sum_d (0,9)$$

==

Karnaugh map, more information needed? See
http://www.allaboutcircuits.com/vol_4/chpt_8/7.html

With the “Karnaugh Map Minimizer” you find the relation between the Truth Table, Karnaugh map and a minimal solution in SOP-form (In the course Digital Hardware EE students will also use the POS-form).
http://sourceforge.net/projects/k-map/files/k-map/0.4/



Truth table

	A	B	C	D	f
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	?
3	0	0	1	1	1
4	0	1	0	0	?
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	?
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Number of variables: 4
Type of solution: Sum of products

Karnaugh map

	00	01	11	10
00	1	1	1	?
01	?	0	?	0
11	1	0	0	0
10	1	0	0	0

Solution:

```
X = |A|B + |C|D
 ... |A|B
 ... |C|D
```

Karnaugh map solved!

Notes:
1) ‘?’ is ‘d’ (don’t care in the book)
2) The maximum of variables used in de K-map in this course is 4.
3) Notice that the GRAY code is used in the rows in columns (and not the binary code).
4) **|C** is used instead of $\bar{C}$ (also C’ is used).
5) When R, S and T are inputs, and you have the AND of these inputs. Write this as: $R \cdot S \cdot T$ and not RST (otherwise names can have only one character!!!)

6)

$$f(A,B,C,D = ..$$

The order of the inputs in the function is relevant for the number of the minterm. In the example above D has weight 2^0 and, C with 2^1, ..
The number if the minterm is $A \times 2^3 + B \times 2^2 + C \times 2^1 + D \times 2^0$)

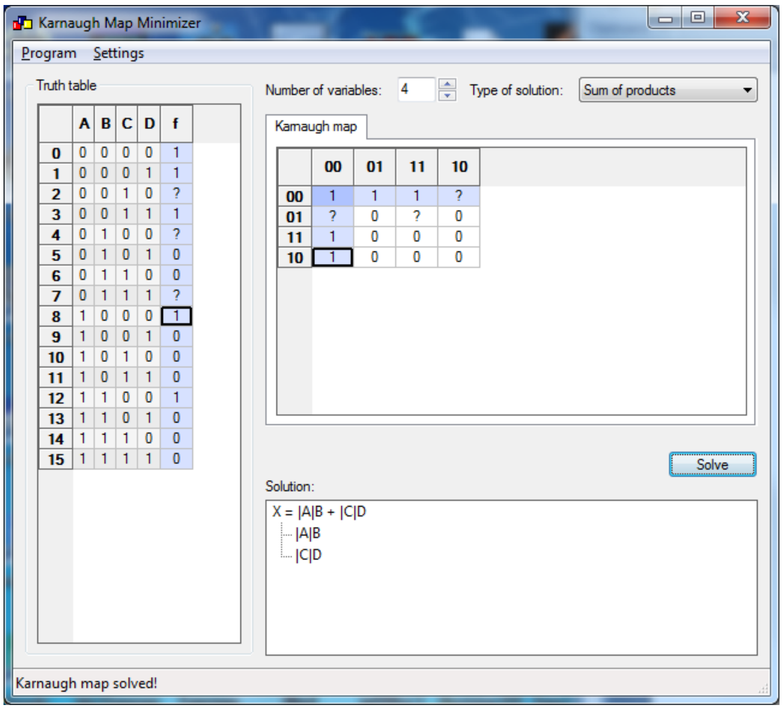
You often see the two Karnaugh map representations for *f(A,B,C,D)* below. Check that the minterms are the same of each corresponding cell.

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

Used in the K-map simulator
We use this mapping

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

Used in the book



Truth table

	A	B	C	D	f
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	?
3	0	0	1	1	1
4	0	1	0	0	?
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	?
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Number of variables: 4
Type of solution: Sum of products

Karnaugh map

	00	01	11	10
00	1	1	1	?
01	?	0	?	0
11	1	0	0	0
10	1	0	0	0

Solution:

```
X = |A|B + |C|D
 ···|A|B
 ···|C|D
```

Karnaugh map solved!

<u>Exercise 8</u>

Simplify the following Boolean functions by means of a four-variable map (minimum SOP form):

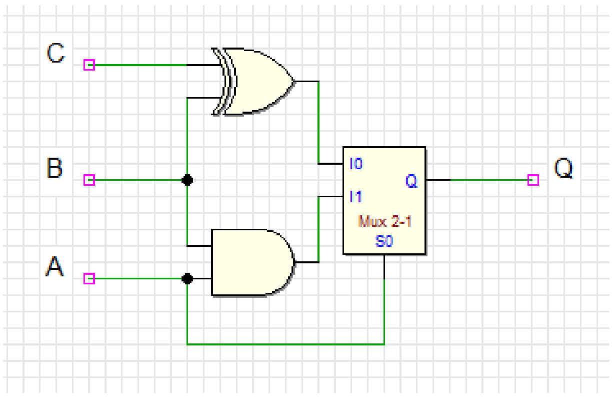
a) $F(A,B,C,D) = \sum m(1,5,9,12,13,15)$
b) $F(A,B,C,D) = \sum m(1,3,9,11,12,13,14,15)$
c) $F(A,B,C,D) = \sum m(0,2,4,5,6,7,8,10,13,15)$

Exercise 9

Simplify the following Boolean functions in Sum-of-Products (SOP)

a) $F(A,B,C,D) = \sum m(0,2)$ and $d = \sum m(8,10,11,15)$
b) $F(A,B,C,D) = \sum m(7,9,13,14)$ and $d = \sum m(0,2,5,6,8,10,11,15)$

Exercise 10

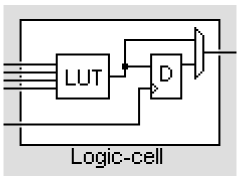


Give a minimum Boolean equation in SOP-form of this system.

Exercise 11

http://www.fpga4fun.com/FPGAinfo2.html

FPGAs are built from one basic "logic-cell", duplicated hundreds or thousands of time. A logic-cell is basically a small lookup table ("LUT"), a D flip-flop and a 2-to-1 mux (to bypass the flip-flop if desired).



The LUT is like a small RAM that can implement any logic function. It has typically a few inputs (4 in the drawing above), so for example an AND gate with 3 inputs, whose result is then OR-ed with another input would fit in one 4-input LUT.

This exercise is about the LUT (so ignore D flip-flop and multiplexer). The LUT is a memory in this example has 4 address lines (left of the LUT), and each memory address has 1 bit (so a “16×1 bit memory”).

In programmable hardware you can fill this memory during programming of the device. The lines to program the LUT are not shown in the picture. E.g. if you fill the memory with all zero’s except address 15 is 1 than it behaves like and AND gate with 4 inputs. Since only when all 4 address lines are 1 the output is 1 (only at that address a 1 is stored in the memory). All the others input combinations result in selecting a memory cell that contains a 0.

How many different combination functions can be implement with a LUT with 4 address lines and 1 output bit?

Exercise 12

A majority function has an output value of 1 if there are more ones than zeros on its inputs else the output is 0. Design a three-input majority function.

Additional (for EE students; “Digital Hardware”): finding the minimal POS-form.

You can find the POS-form with the Karnaugh map with the following procedure:

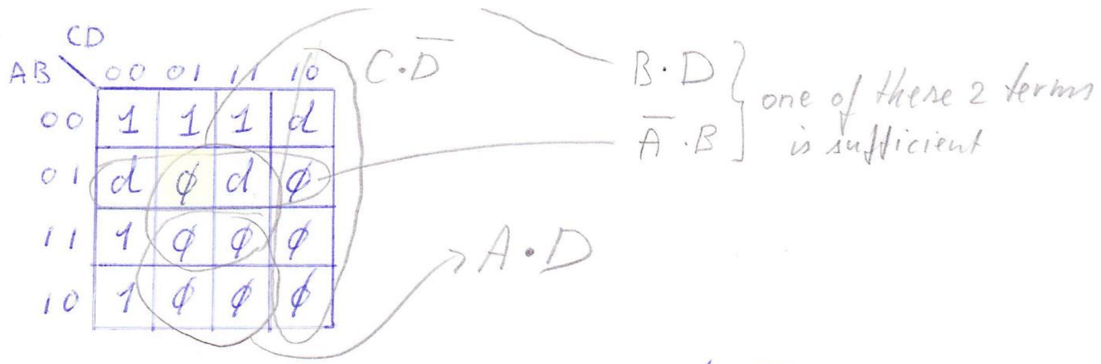
1- Find the minimal SOP-form for the INVERSE (i.e. grouping the zeros instead of the ones).

2- Take the complement (NOT) of the SOP-form

3- Use DeMorgan for the OR term

4- Use deMorgan for the NAND term(s)

The steps are given below for this example.



AB \ CD	00	01	11	10
00	1	1	1	d
01	d	ϕ	d	ϕ
11	1	ϕ	ϕ	ϕ
10	1	ϕ	ϕ	ϕ

1) $\overline{F} = \overline{A}\cdot B + A\cdot D + B\cdot D$ (or $\overline{F} = B\cdot D + A\cdot D + B\cdot D$)

2) $\overline{\overline{F}} = \overline{\overline{A}\cdot B + A\cdot D + B\cdot D}$

3) $F = \overline{\overline{A}\cdot B} \cdot \overline{A\cdot D} \cdot \overline{B\cdot D}$

4) $F = (A+\overline{B})(\overline{A}+\overline{D})(\overline{B}+\overline{D})$