Minimization

Digital Electronics

by Wolfgang Neff

Minimization (1)

- Truth functions often are very complex
- Minimisation tries to simplify them
- There are several algorithms
  – Karnaugh maps
    - Very descriptive
    - Works only well up to four variables
  – Quine-McCluskey algorithm
    - For more variables
    - Complex and less descriptive

Minimization (2)

- Instruction
  - Get the truth table
  - Make the corresponding Karnaugh map
  - Fill in the Karnaugh terms
  - Find blocks of powers of two (2, 4, 8, ...)
  - Drop variables which are in two regions

Truth Table (1)

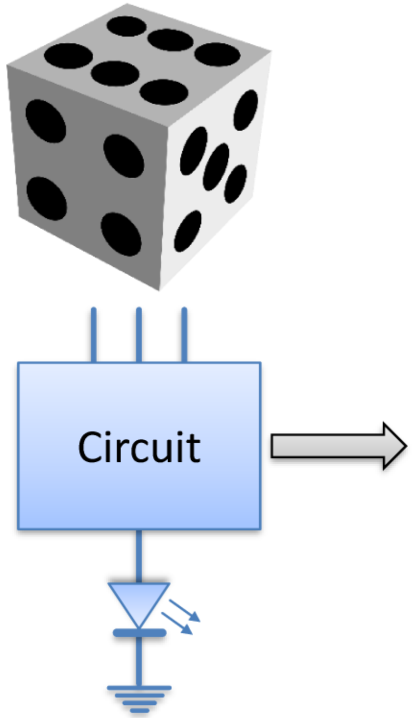
- Get the Truth Table
  - Analyze the problem
  - Find the number of occasions out
  - Find the power of 2 that gives enough occasions
  - Create the corresponding truth table
  - Encode the occasions
  - Determine the result for each line
  - Often there are several possible implementations

Truth Table (2)

- Example: Which face of a dice has even pips?
  - A dice has 6 faces
    - We have 6 occasions
  - An exponent of 3 is enough for 6 occasions
    - We need 3 parameters ($2^2 = 4 \leq 6 \leq 2^3 = 8$)
    - Our circuit has 3 input lines
  - Encoding of the occasions
    - 1 pip → 1, 2 pips → 2 etc.
  - 1 indicates an even number of pips

Truth Table (3)

- Example: even pips (continued)



n	a	b	c	ϕ(a,b,c)	Face
0	0	0	0	X	unused
1	0	0	1	0	1
2	0	1	0	1	2
3	0	1	1	0	3
4	1	0	0	1	4
5	1	0	1	0	5
6	1	1	0	1	6
7	1	1	1	X	unused

Karnaugh Maps (1)

- Two variables

	a	
¬a∧¬b	a∧¬b	
¬a∧b	a∧b	b

- Three variables

		a	a	
¬a∧¬b∧¬c	¬a∧¬b∧c	a∧¬b∧c	a∧¬b∧¬c	
¬a∧b∧¬c	¬a∧b∧c	a∧b∧c	a∧b∧¬c	b
	c	c		

Karnaugh Maps (2)

- Four variables

a

¬a∧¬b∧¬c∧¬d	¬a∧¬b∧c∧¬d	a∧¬b∧c∧¬d	a∧¬b∧¬c∧¬d
¬a∧¬b∧¬c∧d	¬a∧¬b∧c∧d	a∧¬b∧c∧d	a∧¬b∧¬c∧d
¬a∧b∧¬c∧d	¬a∧b∧c∧d	a∧b∧c∧d	a∧b∧¬c∧d
¬a∧b∧¬c∧¬d	¬a∧b∧c∧¬d	a∧b∧c∧¬d	a∧b∧¬c∧¬d

d

b

c

Karnaugh Terms (1)

- Minterms
  - Rows with a result of 1
  - All variables connected by conjunctions
    - Negate variable if they are 0
  - Mark minterms in the map with 1
- Don‘t-care terms
  - Rows with a result of X
  - Mark don’t-care terms in the map with X

Karnaugh Terms (2)

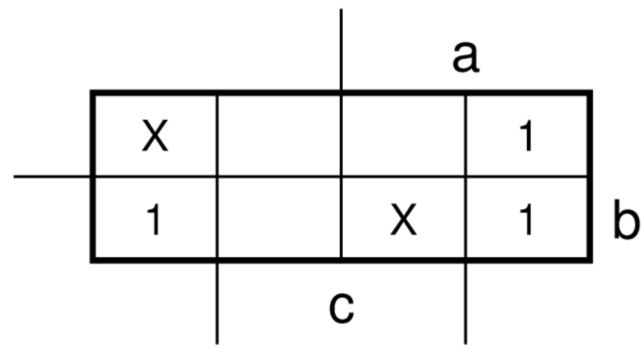
- Finding the Terms
  - Example: even pips

n	a	b	c	ϕ(a,b,c)	
0	0	0	0	X	x_0
1	0	0	1	0	
2	0	1	0	1	m_0
3	0	1	1	0	
4	1	0	0	1	m_1
5	1	0	1	0	
6	1	1	0	1	m_2
7	1	1	1	X	x_1

  - Minterms
    - $m_0 = \neg a \wedge b \wedge \neg c$
    - $m_1 = a \wedge \neg b \wedge \neg c$
    - $m_2 = a \wedge b \wedge \neg c$
  - Don’t-care terms
    - $x_0 = \neg a \wedge \neg b \wedge \neg c$
    - $x_1 = a \wedge b \wedge c$

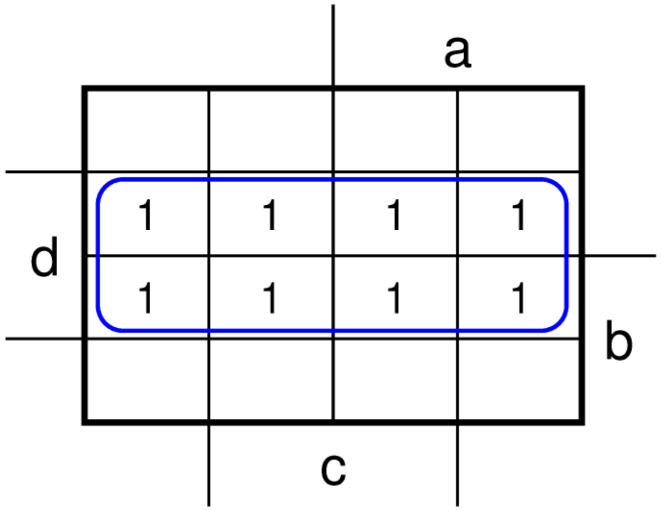
Karnaugh Terms (3)

- Filling the Terms in
  - Example: even pips
    - Minterms
      - $m_0 = \neg a \wedge b \wedge \neg c$
      - $m_1 = a \wedge \neg b \wedge \neg c$
      - $m_2 = a \wedge b \wedge \neg c$
    - Don't-Care Terms
      - $x_0 = \neg a \wedge \neg b \wedge \neg c$
      - $x_1 = a \wedge b \wedge c$



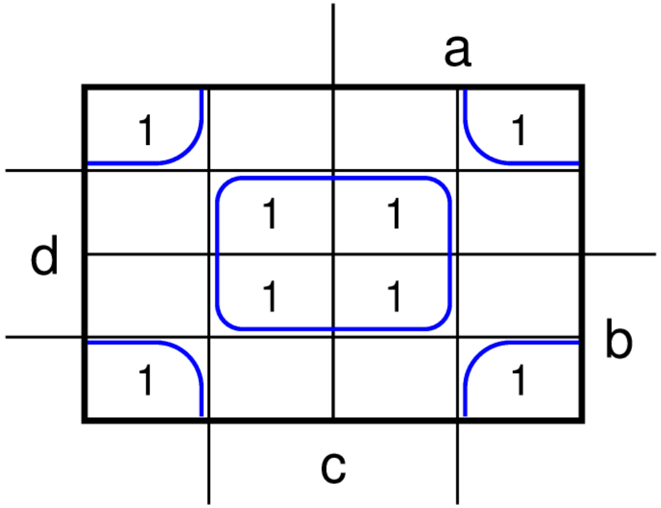
Minimization (3)

- Finding the Blocks
  - Simple blocks
    - Minterms
      $(\neg a \wedge \neg b \wedge \neg c \wedge d)$, $(\neg a \wedge \neg b \wedge c \wedge d)$, $(a \wedge \neg b \wedge c \wedge d)$, $(a \wedge \neg b \wedge \neg c \wedge d)$, $(\neg a \wedge b \wedge \neg c \wedge d)$, $(\neg a \wedge b \wedge c \wedge d)$, $(a \wedge b \wedge c \wedge d)$, $(a \wedge b \wedge \neg c \wedge d)$
    - Result
      $\phi(a,b,c,d) = d$



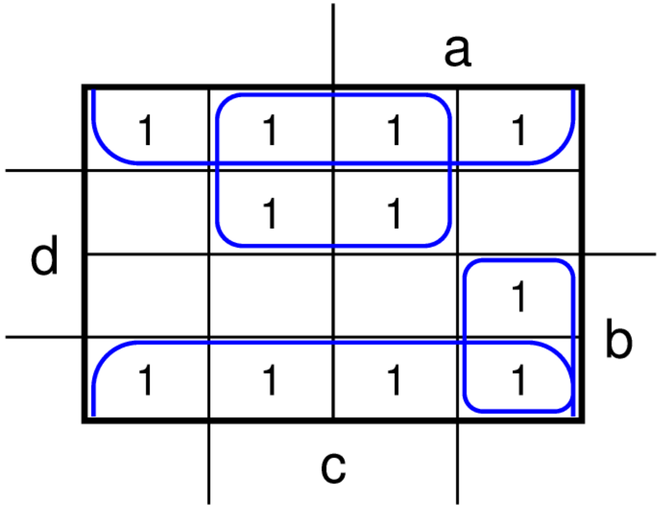
Minimization (4)

- Finding the Blocks (continued)
  - Blocks with border terms
    - Minterms
      $(\neg a \wedge \neg b \wedge \neg c \wedge \neg d)$, $(a \wedge \neg b \wedge \neg c \wedge \neg d)$,
      $(\neg a \wedge \neg b \wedge c \wedge d)$, $(a \wedge \neg b \wedge c \wedge d)$,
      $(\neg a \wedge b \wedge c \wedge d)$, $(a \wedge b \wedge c \wedge d)$,
      $(\neg a \wedge b \wedge \neg c \wedge \neg d)$, $(a \wedge b \wedge \neg c \wedge \neg d)$
    - Result
      $\phi(a,b,c,d) = (c \wedge d) \vee (\neg c \wedge \neg d)$



Minimization (5)

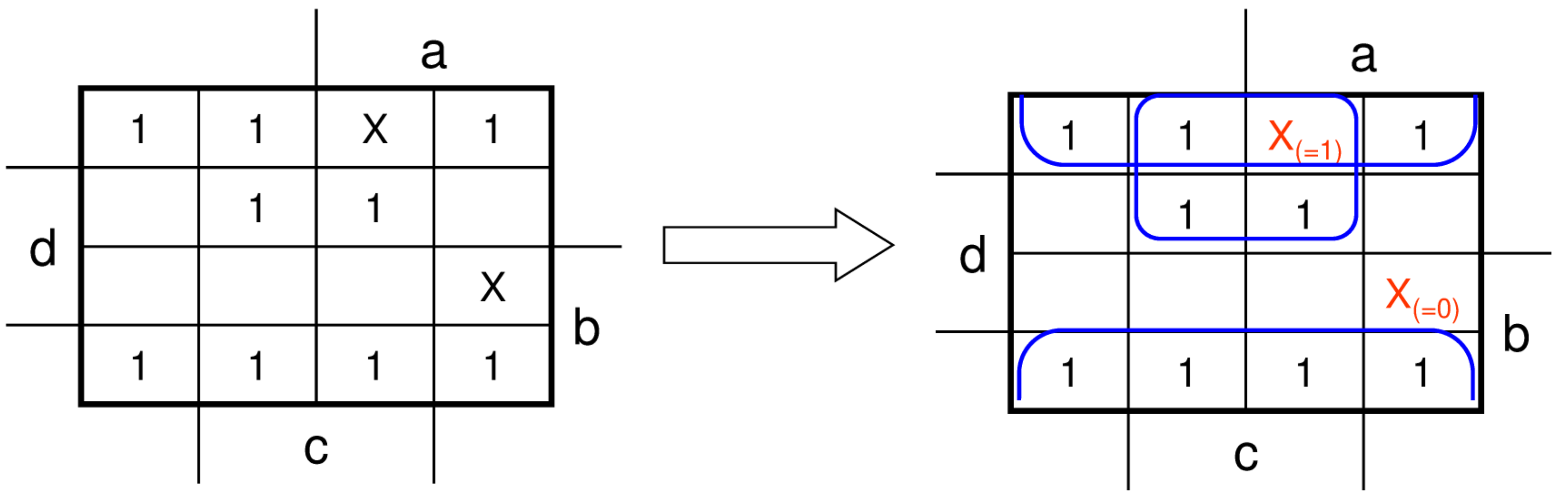
- Finding the Blocks (continued)
  - Blocks with recycled terms
    - Minimise the minterms
      $(\neg a \wedge \neg b \wedge \neg c \wedge \neg d)$, $(\neg a \wedge \neg b \wedge c \wedge \neg d)$, $(a \wedge \neg b \wedge c \wedge \neg d)$, $(a \wedge \neg b \wedge \neg c \wedge \neg d)$, $(\neg a \wedge \neg b \wedge c \wedge d)$, $(a \wedge \neg b \wedge c \wedge d)$, $(a \wedge b \wedge \neg c \wedge d)$, $(\neg a \wedge b \wedge \neg c \wedge \neg d)$, $(\neg a \wedge b \wedge c \wedge \neg d)$, $(a \wedge b \wedge c \wedge \neg d)$, $(a \wedge b \wedge \neg c \wedge \neg d)$



    - Result
      $\phi(a,b,c,d) = \neg d \vee (\neg b \wedge c) \vee (a \wedge b \wedge \neg c)$

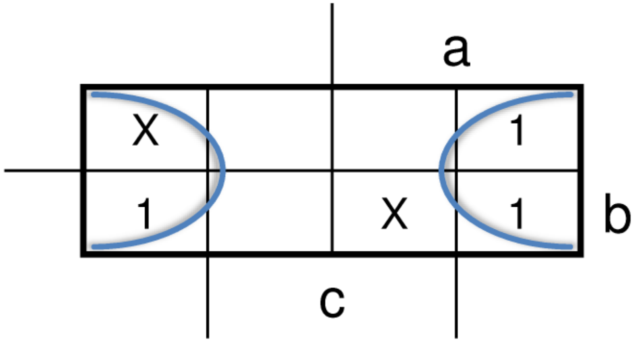
Minimization (6)

- Finding the Blocks (continued)
  - Handling don’t-care terms
    - Helpful for a better minimization.



Minimization (7)

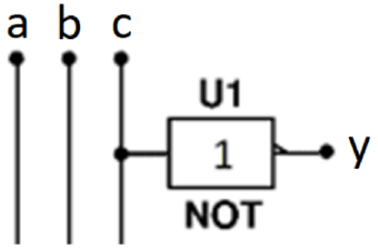
- Finding the Blocks (finished)
  - Example: even pips



  - Switching function

$\phi(a,b,c) = \neg c$

  - Circuit



(yet to come)