Minimization

Networks and Embedded Software

Module 3.2.4

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

Karnaugh Maps (1)

- Two variables

a

¬a∧¬b	a∧¬b
¬a∧b	a∧b

b

- Three variables

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

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)

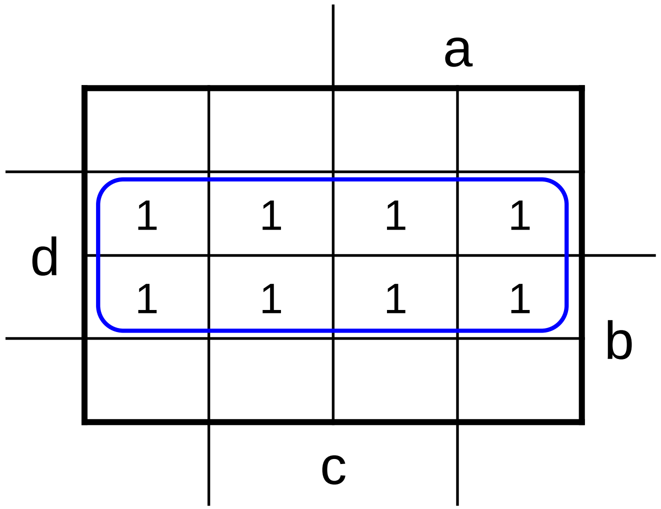
- Example
  - Truth table

a	b	c	φ(a,b,c)	
0	0	0	X	x_0
0	0	1	0	
0	1	0	0	
0	1	1	1	m_0
1	0	0	0	
1	0	1	1	m_1
1	1	0	1	m_2
1	1	1	X	x_1

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

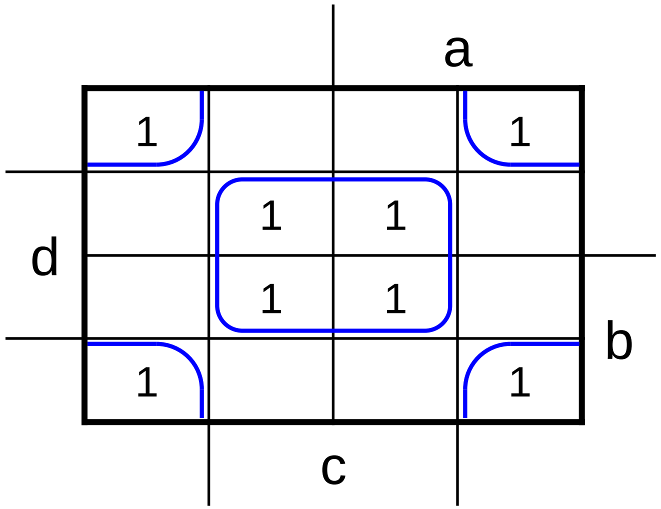
Minimization (3)

- Example
  - Minimise the 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
    $\varphi(a,b,c,d) = d$



Minimization (4)

- Example
  - Minimise the 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
    $\varphi(a,b,c,d) = (c\wedge d) \vee (\neg c\wedge\neg d)$

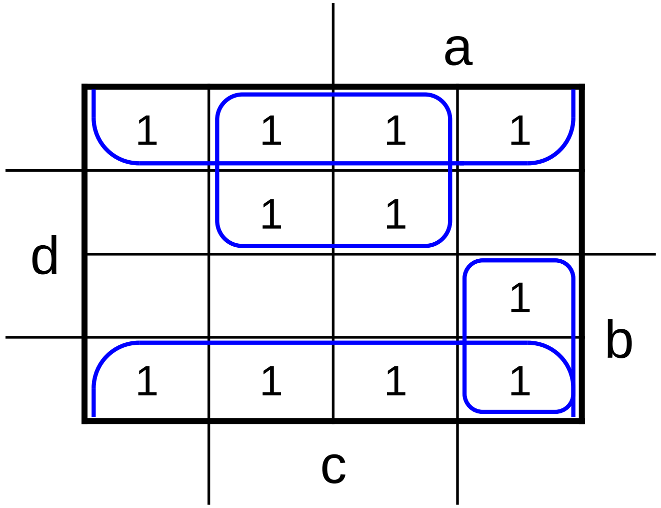


Minimization (5)

- Example
  - Minimise the minterms

    (¬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)
  - Result

    φ(a,b,c,d) =
    ¬d ∨ (¬b∧c) ∨ (a∧b∧¬c)



Minimization (6)

- Don’t-care terms can help to find a better minimization.