Aljabar Boolean
Gerbang Digital

Pertemuan 11/12

(Chapter 11, Schaum Set Theory)

Pendahuluan

- Aljabar boolean, adalah sistem aljabar himpunan atau proposisi yang memenuhi aturan-aturan ekivalen logis.
- Misalkan B dengan operasi + dan *, atau suatu komplemen, dan dua elemen yang beda 0 dan 1 yang didefinisikan pada himpunan atau proposisi, sehingga a,b dan c merupakan elemen B yang mempunyai sifat-sifat identitas, komutatif, distributif dan komplemen.

a+b

+	1	0
1	1	1
0	1	0

a*b

*	1	0
1	1	0
0	0	0

a’

’	1	0
	0	1

x \ y	$\wedge$	0	1
0		0	0
1		0	1

x \ y	$\vee$	0	1
0		0	1
1		1	1

x \ y	$\rightarrow$	0	1
0		1	1
1		0	1

x \ y	$\oplus$	0	1
0		0	1
1		1	0

Figure 1. Truth tables

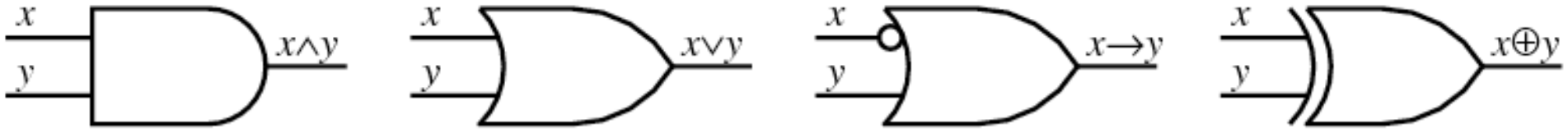


Figure 2. Logic gates

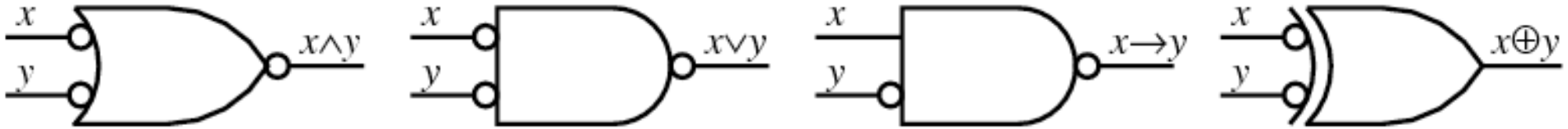


Figure 3. De Morgan equivalents

Figure 4. Venn diagrams

Basis Bilangan

- Biner = 0,1
- Oktal = 0,1,2,3,4,5,6,7
- Desimal = 0,1,2,3,4,5,6,7,8,9
- Hexadesimal = 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Prinsip Dualitas

- Apabila dalam sebuah aljabar boole B, operasi + dipertukarkan dengan * dan sebaliknya, dan 0 dipertukarkan dengan 1 dan sebaliknya.
- (a*1)+(b*0)=b adalah (a+0)*(b+1)=b

Fungsi Boolean

- Dinyatakan dalam bentuk f(x), g(x) dan lainnya.

Pengujian Ekivalen

- Bentuk Standar
- Bentuk Kanonik
  - Bentuk Sum-of-Product

    Sum = gabungan dan product = irisan.
- Masukannya E sebuah ekspresi boole, dan ekspresi SoP ekivalen dengan E
  - Gunakan kaidah2 ekivalen sehingga tersisa bentuk + dan * saja.
  - Gunakan kaidah De Morgan.
  - Gunakan kaidah Distribusi.
  - Gunakan kaidah komutatif.
  - Gunakan kaidah penyerapan dan identitas.

Operasi Logika

Fungsi Boole	Simbol Operator	Nama
F0=0		Null
F1=xy	x*y	and
F2=xy’	x/y	Inhibition
F3=x		Transfer
F4=x’y	y/x	Inhibition

Fungsi Boole	Simbol Operator	Nama
F5=y		Transfer
F6=xy'+x'y	x⊕y	XOR
F7=x+y	x+y	OR
F8=(x+y)'	x↓y	NOR
F9=xy+x'y'	x⊗y	Equivalence

Fungsi Boole	Simbol Operator	Nama
F10=y’	y’	Complement
F11=x + y’	x⊂y	Implication
F12=x’	X’	Complement
F13=x’+y	x ⊃y	Implication
F14=(xy)’	x↑y	NAND
F15=1		Identity

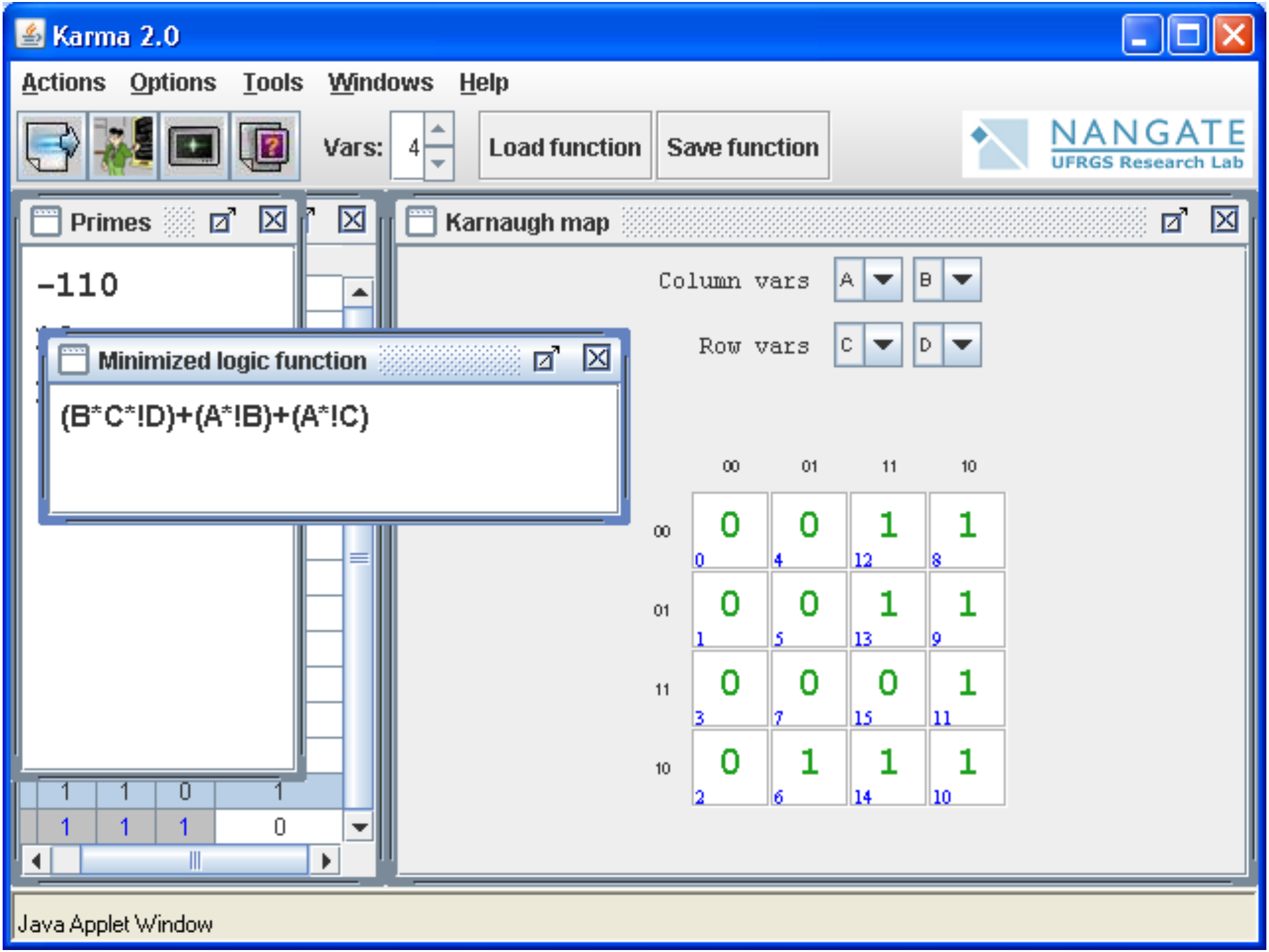
Penyederhanaan

- Aljabar – menggunakan logika ekivalen.
- Peta Karnaugh
- Metode Quine-Mc.Cluskey (untuk kasus empat variabel atau lebih)
- Espresso heuristic
- Metode Petrick's

Peta Karnaugh

- Peta Karnaugh digambarkan dengan kotak bujur sangkar. Setiap kotak merepresentasikan minterm(SoP).
- Jumlah kotak=2 pangkat jumlah variabel.

- The **Karnaugh map**, also known as a **Veitch diagram** (**KV-map** or **K-map** for short), is a tool to facilitate the simplification of Boolean algebra IC expressions. The Karnaugh map reduces the need for extensive calculations by taking advantage of human pattern-recognition and permitting the rapid identification and elimination of potential race hazards.
- The Karnaugh map was invented in 1952 by Edward W. Veitch. It was further developed in 1953 by Maurice Karnaugh, a telecommunications engineer at Bell Labs, to help simplify digital electronic circuits.
- In a Karnaugh map the boolean variables are transferred (generally from a truth table) and ordered according to the principles of Gray code in which only one variable changes in between squares. Once the table is generated and the output possibilities are transcribed, the data is arranged into the largest even group possible and the minterm is generated through the axiom laws of boolean algebra.

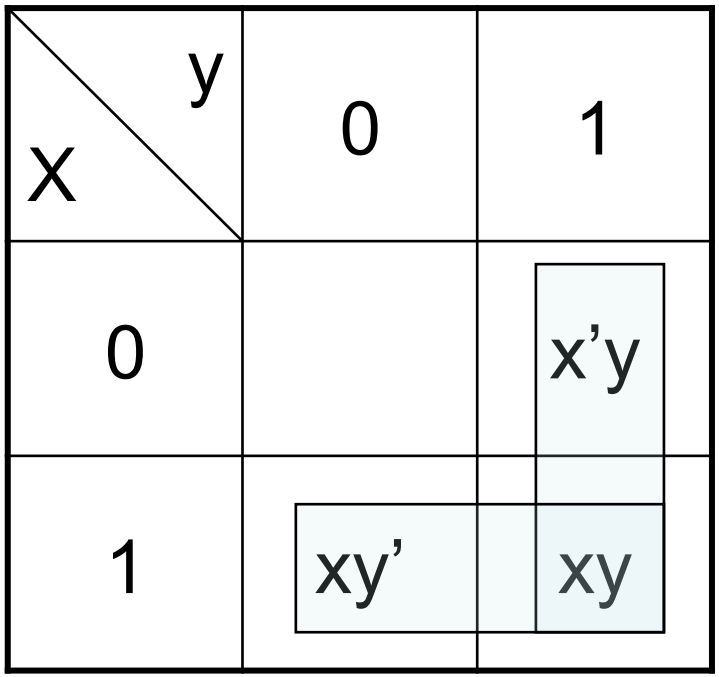

Karma 2.0
Actions
Options
Tools
Windows
Help
Vars:
4
Load function
Save function
NANGATE
UFRGS Research Lab
Primes
-110
Minimized logic function
(B*C*!D)+(A*!B)+(A*!C)
1 1 0 1
1 1 1 0
Karnaugh map
Column vars
A
B
Row vars
C
D
00 01 11 10
00 0 0 1 1
0 4 12 8
01 0 0 1 1
1 5 13 9
11 0 0 0 1
3 7 15 11
10 0 1 1 1
2 6 14 10
Java Applet Window

- Dua variabel

x \ y	0	1
0		x’y
1	xy’	xy

Contoh

•



= x + y

Latihan

- E= xy + xy’
- E= xy + xy’ + x’y’
- E= xy + x’y’

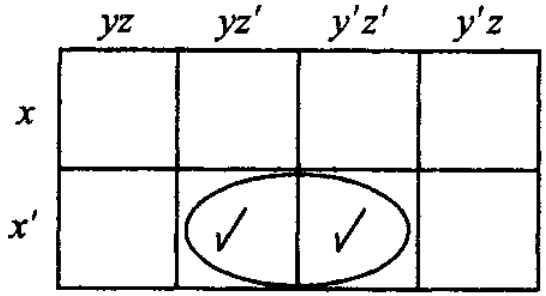
- Tiga variabel

x \ yz	00	01	10	11
0		x’y’z	x’yz’	x’yz
1	xy’z’	xy’z	xyz’	xyz

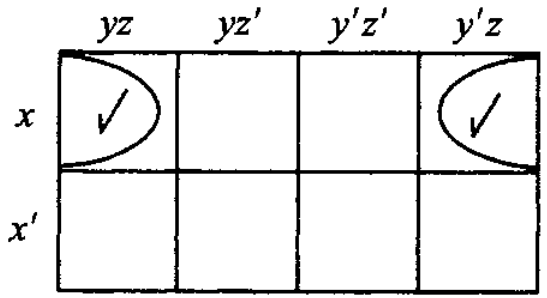
= x + y + z

Latihan

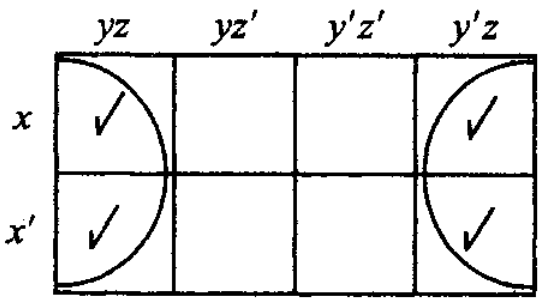
- E= xyz + xyz’ + x’yz’ + x’y’z
- E= xyz + xyz’ + xy’z + x’yz + x’y’z
- E= xyz + xyz’ + x’yz’ + x’y’z + x’y’z’



(*a*)



(*b*)



(*c*)

Metode Quine-Mc.Cluskey

- Dengan peta Karnaugh, untuk empat variabel menjadi lebih kompleks.
- Metode ini akan membantu penyelesaiannya.
- Metode ini terdiri atas dua bagian
  - Menentukan term-term sebagai kandidat.
  - Memilih kandidat untuk mendapatkan ekspresi dengan jumlah literal sedikit.

ESPRESSO Heuristic

- A radically different approach to this issue is followed in the ESPRESSO algorithm, developed by Brayton e.a. at the University of California, Berkeley.[7] Rather than expanding a logic function into minterms, the program manipulates "cubes", representing the product terms in the ON-, DC- and OFF-covers iteratively. Although the minimization result is not guaranteed to be the global minimum, in practice this is very closely approximated, while the solution is always free from redundancy.

- Compared to the other methods, this one is essentially more efficient, reducing memory usage and computation time by several orders of magnitude. Its name reflects the way of instantly making a cup of fresh coffee. There is hardly any restriction to the number of variables, output functions and product terms of a combinational function block. In general, e.g. tens of variables with tens of output functions are readily dealt with.

- The input for ESPRESSO is a function table of the desired functionality; the result is a minimized table, describing either the ON-cover or the OFF-cover of the function, depending on the selected options. By default the product terms will be shared as much as possible by the several output functions, but the program can be instructed to handle each of the output functions separately. This allows for efficient implementation in two-level logic arrays such as a PLA (Programmable Logic Array) or a PAL (Programmable Array Logic).

- The ESPRESSO algorithm proved so successful that it has been incorporated as a standard logic function minimization step into virtually any contemporary logic synthesis tool. For implementing a function in multi-level logic, the minimization result is optimized by factorization and mapped onto the available basic logic cells in the target technology, whether this concerns an FPGA (Field Programmable Gate Array) or an ASIC (Application Specific Integrated Circuit).