Peta Karnaugh & Rangkaian Multi-Keluaran (Bagian 1)

Kuliah#4 TKC205 Sistem Digital

Eko Didik Widianto

Departemen Teknik Sistem Komputer, Universitas Diponegoro

Umpan Balik

- Sebelumnya dibahas tentang implementasi fungsi logika menjadi suatu rangkaian logika (disebut proses sintesis), baik menggunakan tabel kebenaran, maupun aljabar Boolean
  - Aljabar Boolean: aksioma, teorema, dan hukum
  - Diagram Venn
  - Manipulasi aljabar
  - Sintesis ekspresi logika dari tabel kebenaran
  - Bentuk kanonik: minterm/SOP dan maxterm/POS beserta notasinya
  - Konversi SOP <-> POS
  - Rangkaian AND-OR, OR-AND
  - Rangkaian NAND-NAND, NOR-NOR
- Rangkaian optimal dapat diperoleh dengan **penyederhanaan ekspresi logika secara Aljabar**

Tentang Kuliah

- Dibahas proses sintesis rangkaian logika minimal menggunakan peta Karnaugh untuk menyederhanakan persamaan fungsi logika
  - Peta Karnaugh juga digunakan untuk merancang rangkaian multikeluaran minimal
- Pokok Bahasan:
  - peta Karnaugh: 2 variabel, 3-variabel, 4-variabel, 5-variabel dan 6-variabel
  - strategi minimisasi rangkaian SOP (pengelompokan minterm)
  - kondisi *don't care* dan rangkaian dengan spesifikasi tidak lengkap
  - literal, implicant, *cover*, *cost,* implicant utama dan fungsi minimum
  - implementasi rangkaian logika SOP optimal dengan AND-OR dan/atau NAND-NAND

Kompetensi Dasar

- Setelah mempelajari bab ini, mahasiswa akan mampu:
  1. [C2] memahami prinsip-prinsip penyederhanaan fungsi logika menggunakan peta Karnaugh;
  2. [C3] menggunakan *Don't care* dalam peta Karnaugh;
  3. [C4] mendesain dan menganalisis rangkaian logika SOP minimal (AND-OR atau NAND-NAND) menggunakan peta Karnaugh;
- Link
  - Website: `http://didik.blog.undip.ac.id/2017/03/06/tkc205-sistem-digital-2016-genap/`
  - Email: didik@live.undip.ac.id

Buku Acuan/Referensi

Eko Didik Widianto, Sistem Digital: Analisis, Desain dan Implementasi, Edisi Pertama, Graha Ilmu, 2014 **(Bab 4: Peta Karnaugh dan Rangkaian Multikeluaran)**

- Materi: 4.1 Peta Karnaugh
    - 4.1.1 Representasi Peta Karnaugh
    - 4.1.2 Pengelompokan Minterm
    - 4.1.3-5 K-map Tiga Variabel, Empat Variabel, dan Banyak Variabel
    - 4.1.6 Literal, Implicant, Cover dan Cost
    - 4.1.7 Implicant Utama dan Fungsi Minimum Rangkaian
- Website:
    - http://didik.blog.undip.ac.id/buku/sistem-digital/

Bahasan

Rangkaian Optimal

- Rangkaian optimal
  - *Cost* rangkaian sekecil mungkin: jumlah gerbang (dan transistor), jumlah jalur
  - Fungsional terpenuhi
  - *Constraint* terpenuhi: delay, *fanout* (*driving*), area
- Rangkaian optimal biasanya minimal
- Rangkaian optimal bisa diperoleh dengan teknik:
  1. Penyederhanaan fungsi logika
     - Menggunakan prinsip-prinsip Aljabar Boolean
     - Menggunakan Karnaugh Map
  2. Penggunaan gerbang secara bersama untuk beberapa fungsi sekaligus, membentuk **rangkaian multi-keluaran**

Bahasan

Prinsip Penyederhanaan

- Operasi penyederhanaan adalah mengurangi minterm atau maxterm di ekspresi
  - SOP: menggunakan hukum 14a ($x \cdot y + x \cdot \overline{y} = x$)
  - POS: menggunakan hukum 14b ($(x + y) \cdot (x + \overline{y}) = x$)
- Beberapa minterm atau maxterm dapat digabungkan menggunakan hukum 14a atau 14b jika berbeda hanya di satu variabel saja
  $f(x_1, x_2, x_3) = \overline{x}_1\overline{x}_2x_3 + x_1\overline{x}_2\overline{x}_3 + x_1\overline{x}_2x_3 + x_1x_2\overline{x}_3$
  m_1 dan m_5 berbeda di x_1, dan m_4 dan m_6 berbeda di x_2

$$
\begin{aligned}
f &= \overline{x}_1\overline{x}_2x_3 + x_1\overline{x}_2x_3 + x_1\overline{x}_2\overline{x}_3 + x_1x_2\overline{x}_3 \\
&= (\overline{x}_1 + x_1)\,\overline{x}_2x_3 + x_1(\overline{x}_2 + x_2)\overline{x}_3 \\
&= \overline{x}_2x_3 + x_1\overline{x}_3
\end{aligned}
$$

$f(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(x_1 + \overline{x}_2 + x_3)(x_1 + \overline{x}_2 + \overline{x}_3)(\overline{x}_1 + \overline{x}_2 + \overline{x}_3)$
M_0 dan M_2 berbeda di x_2, dan M_4 dan M_7 berbeda di x_1

$$
\begin{aligned}
f &= ((x_1 + x_3) + x_2\overline{x}_2)\,(x_1\overline{x}_1 + (\overline{x}_2 + \overline{x}_3)) \\
&= (x_1 + x_3)\,(\overline{x}_2 + \overline{x}_3)
\end{aligned}
$$

Peta Karnaugh

- **Peta Karnaugh** (K-map) menyediakan cara sistematik dan grafis untuk mencari rangkaian SOP dan POS minimal
- K-map SOP
  - mengelompokkan minterm-minterm bernilai 1 yang saling berdekatan, yang hanya mempunyai perbedaan di satu variabel saja
  - membentuk rangkaian AND-OR
- K-map POS
  - mengelompokkan Maxterm-Maxterm bernilai 0 yang saling berdekatan
  - membentuk rangkaian OR-AND minimal

Representasi Peta Karnaugh

- K-map juga merupakan **alternatif** untuk menyatakan suatu fungsi logika selain tabel kebenaran dan ekspresi logika
  - K-map disusun atas sel-sel. Satu sel, satu minterm

x_1	x_2	$f(x_1,x_2)$	minterm
0	0	m_0	$\overline{x}_1\overline{x}_2$
0	1	m_1	$\overline{x}_1 x_2$
1	0	m_2	$x_1\overline{x}_2$
1	1	m_3	$x_1 x_2$

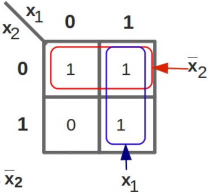
$x_2 \backslash x_1$	0	1
0	m_0	m_2
1	m_1	m_3

Bahasan

Grouping K-Map

- Minterm-minterm yang berdekatan **dapat dikombinasikan** karena mereka hanya berbeda di satu variabel saja, disebut **Grouping**
- Grouping dilakukan dengan melingkari nilai ’1’ yang berdekatan
- Melingkari **dua nilai ’1’ bersama**, berarti **mengeliminasi satu term dan satu variabel** dari ekspresi output
  - Variabel yang dieliminasi adalah **yang mempunyai perbedaan nilai** di grup, vertikal/horizontal
  - Group merah: x_1 dieliminasi, Grup biru: x_2 dieliminasi

x_1	x_2	$f(x_1,x_2)$
0	0	1
0	1	0
1	0	1
1	1	1



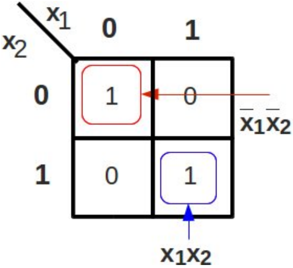
$$f(x_1,x_2) = x_1 + \bar{x}_2$$

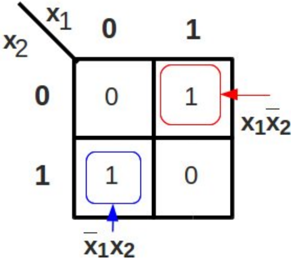
Ketentuan dan Tips Grouping

- Hanya dapat mengkombinasikan nilai 1 yang berdekatan
- Hanya dapat menggabungkan 2^n minterm (1,2,4,8,16, dst)
- Bentuk grup sebesar mungkin
  - grup 2 minterm menghilangkan 1 variabel
  - grup 4 minterm menghilangkan 2 variabel
  - grup 8 minterm menghilangkan 3 variabel
- Group yang sudah dicover oleh group lain **tidak perlu digabungkan lagi**

Contoh Grouping Fungsi 2 Variabel

Sederhanakan: $f(x_1, x_2) = \sum m(0,3)$ dan $f(x_1, x_2) = \sum m(1,2)$

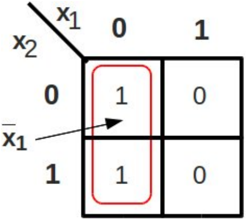


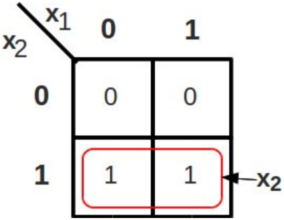


- $f(x_1, x_2) = \sum m(0,3) = \overline{x}_1\overline{x}_2 + x_1 x_2$
  - fungsi SOP tidak dapat disederhanakan
- $f(x_1, x_2) = \sum m(1,2) = x_1\overline{x}_2 + \overline{x}_1 x_2$
  - fungsi SOP tidak dapat disederhanakan

Contoh Grouping Fungsi 2 Variabel

- Sederhanakan: $f(x_1, x_2) = \sum m(0,1)$ dan $f(x_1, x_2) = \sum m(1,3)$

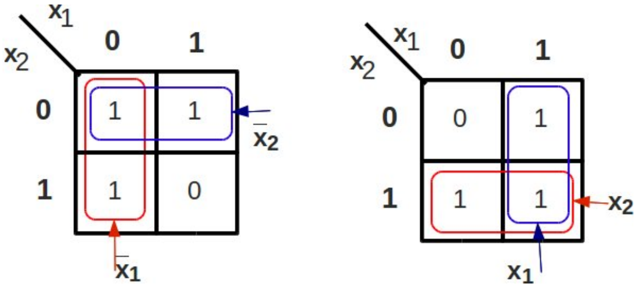




- $f(x_1, x_2) = \sum m(0,1) = \overline{x}_1\overline{x}_2 + \overline{x}_1 x_2 = \overline{x}_1$, $x_2$dieliminisi
- $f(x_1, x_2) = \sum m(1,3) = \overline{x}_1 x_2 + x_1 x_2 = x_2$, $x_1$dieliminasi

Contoh Grouping Fungsi 2 Variabel

- Sederhanakan: $f(x_1, x_2) = \sum m(0,1,2)$ dan $f(x_1, x_2) = \sum m(1,2,3)$

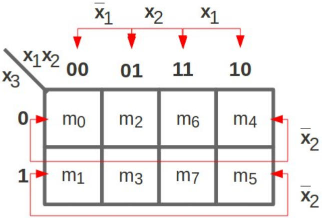


- $f(x_1, x_2) = \sum m(0,1,2) = \overline{x}_1\overline{x}_2 + \overline{x}_1 x_2 + x_1\overline{x}_2 = \overline{x}_1 + \overline{x}_2$
- $f(x_1, x_2) = \sum m(1,2,3) = \overline{x}_1 x_2 + x_1\overline{x}_2 + x_1 x_2 = x_1 + x_2$

K-Map 3 Variabel

- K-map disusun sehingga minterm yang berdekatan hanya mempunyai perbedaan 1 variabel

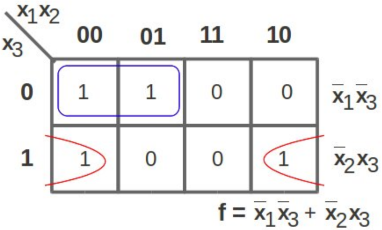
x_1	x_2	x_3	minterm m_j
0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$
0	0	1	$m_1 = \overline{x}_1\overline{x}_2x_3$
0	1	0	$m_2 = \overline{x}_1x_2\overline{x}_3$
0	1	1	$m_3 = \overline{x}_1x_2x_3$
1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$
1	0	1	$m_5 = x_1\overline{x}_2x_3$
1	1	0	$m_6 = x_1x_2\overline{x}_3$
1	1	1	$m_7 = x_1x_2x_3$



Kedua sel tepi berdekatan

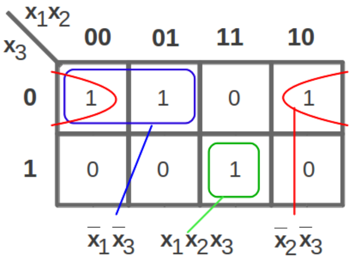
Contoh K-Map 3 Variabel

- Sederhanakan $f(x_1, x_2, x_3) = \sum m(0, 1, 2, 5)$



Contoh K-Map 3 Variabel

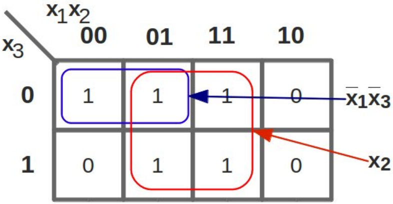
- Sederhanakan $f(x_1, x_2, x_3) = \sum m(0, 2, 4, 7)$



- menghasilkan $f(x_1, x_2, x_3) = \overline{x}_1\overline{x}_3 + \overline{x}_2\overline{x}_3 + x_1x_2x_3$

Contoh K-Map 3 Variabel

- Sederhanakan: $f(x_1, x_2, x_3) = \sum m(1, 3, 5, 7)$,
  $f(x_1, x_2, x_3) = \sum m(0, 2, 3, 6, 7)$



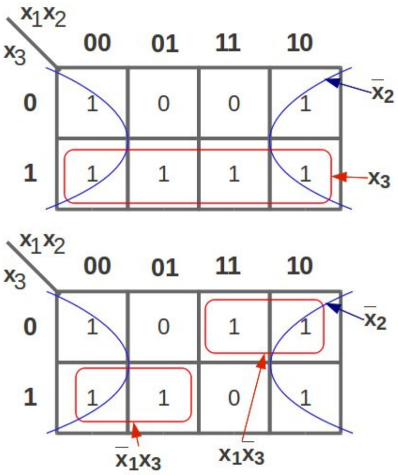
Desain Rangkaian Logika

Dari sebuah K-map, implementasi rangkaian logika bisa mempunyai dua bentuk, yaitu:

1. Jika diinginkan rangkaian logika dengan AND-OR atau NAND-NAND, maka persamaan logika SOP minimal dapat diperoleh dengan mengelompokkan minterm bernilai 1;
2. Jika diinginkan rangkaian logika dengan OR-AND atau NOR-NOR, maka persamaan logika POS minimal dapat diperoleh dengan mengelompokkan Maxterm bernilai 0;

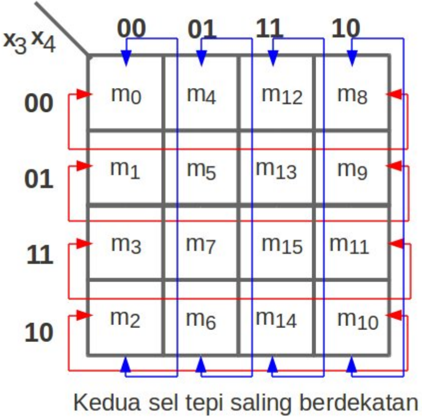
Contoh K-Map 3 Variabel

- Rancang rangkaian NAND-NAND dari fungsi $f(x_1, x_2, x_3) = \sum m(0, 1, 3, 4, 5, 7)$ dan $f(x_1, x_2, x_3) = \prod M(2, 7)$



K-Map 4 Variabel

- Bentuk K-map 4 variabel:



Contoh: Grouping K-Map 4 Variabel

- Sederhanakan $f(x_1, x_2, x_3, x_4) = \sum m(2, 3, 8 - 11, 13)$

x_3x_4 \ x_1x_2	00	01	11	10
00	0	0	0	1
01	0	0	1	1
11	1	0	0	1
10	1	0	0	1

Grouping K-Map 4 Variabel

- Sederhanakan fungsi $f(x_1, x_2, x_3, x_4) = \prod M(0, 2, 4, 8 - 12, 14)$ dengan K-map

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	1	1	1	0
10	0	1	0	0

- Menghasilkan $f(x_1, x_2, x_3, x_4) = \overline{x}_1 x_4 + x_2 x_4 + x_1 \overline{x}_2 x_3$

Umpan Balik: Grouping K-Map 4 Variabel

Sederhanakan:

- $f(x_1, x_2, x_3) = \sum m(3-7, 9, 11, 12-15)$
- $f(x_1, x_2, x_3) = \sum m(0-4, 6, 9, 11, 12, 14)$
- $f(x_1, x_2, x_3) = \sum m(0, 2, 5, 7, 8, 10, 13, 15)$

K-Map 5 Variabel

$x_5=0$

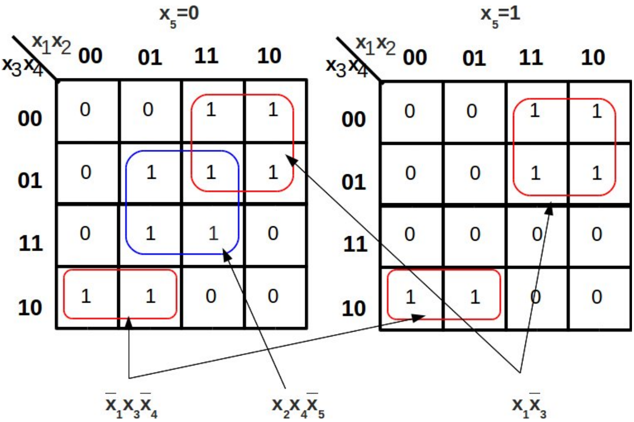
$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	m_0	m_8	m_{24}	m_{16}
01	m_2	m_{10}	m_{26}	m_{18}
11	m_6	m_{14}	m_{30}	m_{22}
10	m_4	m_{12}	m_{28}	m_{20}

$x_5=1$

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	m_1	m_9	m_{25}	m_{17}
01	m_3	m_{11}	m_{27}	m_{19}
11	m_7	m_{15}	m_{31}	m_{23}
10	m_5	m_{13}	m_{29}	m_{21}

Contoh K-map 5 Variabel

- Sederhanakan fungsi $f\,(x_1, x_2, x_3, x_4, x_5) = \sum m(4, 5, 10, 12-14, 16-19, 24-27, 30)$



K-map 6 Variabel

- Bagaimana K-Map 6 Variabel? Tidak berguna dari sudut pandang praktis
  - Akan membutuhkan perangkat CAD, salah satunya bmin http://bukka.eu/bmin/0.5.0
- Contoh: $f(f,e,d,c,b,a) = \sum m(21,23,29,31,53,55,61,63) = ace$

edc \ fba	000	001	011	010	110	111	101	100
000	0	0	0	0	0	0	0	0
001	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0
010	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0
111	0	1	1	0	0	1	1	0
101	0	1	1	0	0	1	1	0
100	0	0	0	0	0	0	0	0

Bahasan

Peta Karnaugh
- Karnaugh Map
- Grouping K-Map
- Literal, Implicant, Cover dan Cost

Ringkasan

Lisensi

Terminologi

- ***Literal*** = variabel di suatu term
  - Contoh: $\overline{x}_1 x_2 x_3 \overline{x}_4$(term dg 4 literal), $x_2 x_3$(term dg 2 literal)
- ***Implicant***: sebarang term bernilai '1' atau grup term bernilai '1' yang dapat digabungkan di K-map
  - minterm adalah *implicant* dasar. Untuk fungsi n-variabel, minterm adalah *implicant* dengan n literal
- ***Prime Implicant***: *implicant* yang tidak bisa digabungkan dengan implicant lain untuk menghilangkan sebuah variabel
  - Literal dalam prime implicant tidak dapat dihapus untuk mendapatkan *implicant* valid
- ***Cover***: suatu himpunan *implicant* yang menghasilkan nilai fungsi '1'
- ***Cost***: jumlah gerbang ditambah jumlah total masukan ke semua gerbang dalam rangkaian logika

Implicant dan Prime Implicant

x_3x_4 \ x_1x_2	00	01	11	10
00	0	0	0	1
01	0	0	1	1
11	1	0	0	1
10	1	0	0	1

- Terdapat **10 implicant** valid
  - 7 buah minterm
  - 1 term 3-literal (grup 2 minterm)
  - 2 term 2-literal (grup 4 minterm)
- Terdapat **3 prime implicant**
  - $x_1\overline{x}_2,\ \overline{x}_2x_3,\ x_1\overline{x}_3x_4$
  - Tidak bisa disederhanakan lagi?
    - Untuk $x_1\overline{x}_2$, jika sebuah literal dihapus menyisakan x_1 atau $\overline{x}2$, padahal $x_1$bukan implicant valid karena {1,1,0,0} menghasilkan $f = 0$

Cover dan Cost

- Cover untuk $f(x_1, x_2, x_3, x_4) = \sum m(2, 3, 8, 9, 10, 11, 13)$
  1. Persamaan dengan semua minterm
  2. $f = x_1\overline{x}_2 + \overline{x}_1\overline{x}_2x_3 + x_1\overline{x}_3x_4$ merupakan cover valid
  3. $f = x_1\overline{x}_2 + \overline{x}_2x_3 + x_1\overline{x}_3x_4$ merupakan cover valid yang berisi *prime implicant*
- Cost untuk setiap cover: (asumsi input utama baik terinvers atau tidak mempunyai cost 0)
  1. jumlah gerbang=7+1, jumlah input semua gerbang=7*4+7*1, total=8+28+7=43
  2. jumlah gerbang=3+1, jumlah input semua gerbang=8+3, total=4+11=15
  3. jumlah gerbang=3+1, jumlah input semua gerbang=7+3, **total=4+10=14**
- Cover yang berisi prime implicant **cenderung menghasilkan** implementasi dengan **cost terendah**

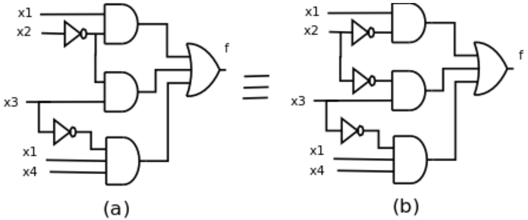
Menghitung Cost Rangkaian

- Fungsi $f = x_1\overline{x}_2 + \overline{x}_2x_3 + x_1\overline{x}_3x_4$
- NOT tidak diperhitungkan

Gerbang	#Gerbang	#Masukan	Keterangan
AND-3	1	$1 \times 3 = 3$	$\rightarrow x_1\overline{x}_3x_4$
AND-2	2	$2 \times 2 = 4$	$\rightarrow x_1\overline{x}_2$ dan $\overline{x}_2x_3$
OR-3	1	$1 \times 3 = 3$	
Total	4	10	

Cost$= 4 + 10 = 14$

Jika Gerbang NOT Diperhitungkan



Gerbang	#Gerbang	#Masukan	Keterangan
AND-3	1	$1 \times 3 = 3$	$\rightarrow x_1 \overline{x}_3 x_4$
AND-2	2	$2 \times 2 = 4$	$\rightarrow x_1 \overline{x}_2$ dan $\overline{x}_2 x_3$
NOT	2	$2 \times 1 = 2$	$\rightarrow$1 masukan, $x_2$dan x_3
OR-3	1	$1 \times 3 = 3$	
Total	6	12	Cost$= 6 + 12 = 18$

Gerbang	#Gerbang	#Masukan	Keterangan
AND-3	1	$1 \times 3 = 3$	$\rightarrow x_1 \overline{x}_3 x_4$
AND-2	2	$2 \times 2 = 4$	$\rightarrow x_1 \overline{x}_2$ dan $\overline{x}_2 x_3$
NOT	3	$3 \times 1 = 3$	$\rightarrow$1 masukan, $x_2$dan x_3
OR-3	1	$1 \times 3 = 3$	
Total	7	13	Cost$= 7 + 13 = 20$

Prime Implicant Esensial dan Non-Esensial

SOP minimum hanya mengandung prime implicant (namun tidak semua prime implicant)

- **Essential**: diperlukan untuk membentuk SOP minimum
- **Nonessensial**: tidak diperlukan untuk SOP minimum, sehingga dapat dihilangkan

$x_3x_4 \backslash x_1x_2$	00	01	11	10
00	0	0	0	1
01	0	1	1	1
11	1	0	0	1
10	1	0	0	1

- Prime implicant: $x_1\overline{x}_2$, $\overline{x}_2x_3$, $x_1\overline{x}_3x_4$ dan $x_2\overline{x}_3x_4$
- Esensial: $x_1\overline{x}_2$, $\overline{x}_2x_3$, dan $x_2\overline{x}_3x_4$
- non-esensial: $x_1\overline{x}_3x_4$
- $f_{min} = x_1\overline{x}_2 + \overline{x}_2x_3 + x_2\overline{x}_3x_4$, $x_1\overline{x}_3x_4$ dihilangkan

Contoh

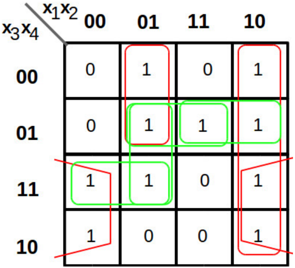
x_3x_4 \ x_1x_2	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	1	1	0	1
10	1	0	0	1

- Prime implicant: $x_1\overline{x}_2$, $\overline{x}_2x_3$, $\overline{x}_1x_2\overline{x}_3$, $\overline{x}_1x_2x_4$ dan $\overline{x}_1x_3x_4$
- Esensial: $x_1\overline{x}_2$, $\overline{x}_2x_3$, dan $\overline{x}_1x_2\overline{x}_3$
- non-esensial: $\overline{x}_1x_2x_4$, $\overline{x}_1x_3x_4$ (harus dipilih salah satu)
- $f_{min} = x_1\overline{x}_2 + \overline{x}_2x_3 + \overline{x}_1x_2\overline{x}_3 + \left\{ \begin{array}{c} \overline{x}_1x_2x_4 \\ \overline{x}_1x_3x_4 \end{array} \right\}$

Langkah Penyederhanaan

- SOP minimum berisi **semua prime implicant esensial** dan beberapa prime implicant non-esensial
- Langkah menemukan rangkaian dengan cost minimum:
  1. Cari semua prime implicant dari f
  2. Cari set prime implicant esensial
  3. Jika set tersebut telah meng-cover semua valuation dimana $f = 1$, maka set ini adalah cover dari f yang diinginkan. Jika tidak, tentukan prime implicant non-esensial yang harus ditambahkan agar minimum
- Menentukan prime implicant non-esensial? *heuristik* (mencoba semua kemungkinan untuk mendapatkan cover dengan cost minimum)

Latihan



- Cari semua prime implicant dari *f*
- Cari set prime implicant esensial
- Cari cover dengan cost terendah dari semua kombinasi prime implicant non-esensial

Ringkasan Kuliah

- Yang telah kita pelajari hari ini:
  - Penyederhanaan fungsi logika menggunakan peta Karnaugh melalui Grouping minterm untuk rangkaian SOP, baik fungsi 2-variabel sampai 6-variabel
  - Terminologi dalam K-map, yaitu implicant, prime implicant (esensial, non-esensial), cover dan cost beserta contoh penggunaan istilah-istilah tersebut
- Yang akan kita pelajari di pertemuan berikutnya adalah penyederhanaan fungsi logika menggunakan peta Karnaugh melalui grouping Maxterm untuk rangkaian POS, fungsi tidak lengkap dan rangkaian multikeluaran

Lisensi