

Sintesis Rangkaian Logika

Eko Didik Widiyanto (didik@undip.ac.id)

21 Maret 2011

Program Studi Sistem Komputer - Universitas Diponegoro

Artikel ini menjelaskan secara khusus langkah-langkah sintesis untuk membangkitkan rangkaian logika dari deskripsi perilaku fungsional sistem yang diinginkan. Sintesis ini merupakan proses utama dari satu siklus perancangan rangkaian/sistem digital. Proses sintesis ini dimulai dengan menerjemahkan kebutuhan sistem dan mengekspresikannya secara formal dalam tabel kebenaran, kemudian menyatakannya dalam ekspresi SOP/POS, menyederhanakan ekspresi, serta mengimplementasikannya dalam rangkaian logika berupa susunan gerbang-gerbang AND-OR, OR-AND, NAND-NAND atau NOR-NOR.

Artikel ini merupakan ekstraksi dari materi 1-3 mata kuliah TSK-205 Sistem Digital di Program Studi Sistem Komputer, Universitas Diponegoro. Materi 1 membahas tentang sistem digital, sekilas teknologi implementasi sistem dan metodologi perancangan sistem. Materi 2 membahas tentang konsep rangkaian digital, fungsi logika, representasi fungsi logika dengan tabel kebenaran dan ekspresi logika, variabel, ekspresi dan persamaan logika, serta gerbang logika dan rangkaian logika. Materi 3 membahas tentang aljabar Boolean, pembuktian persamaan fungsi dengan aljabar, tabel kebenaran dan diagram Venn, sintesis rangkaian logika, persamaan sum-of-product (SOP) dan product-of-sum (POS), minterm, maxterm, dan bentuk kanonik SOP dan POS, penyederhanaan ekspresi menggunakan aljabar, implementasi rangkaian dengan susunan gerbang AND-OR, OR-AND, NAND-NAND dan NOR-NOR.

Pembahasan disusun menjadi 2 bagian. Bagian pertama membahas dasar-dasar perancangan rangkaian logika. Bagian ini membahas pernyataan kebutuhan fungsional sistem ke dalam deskripsi formal, representasi fungsi logika, penyederhanaan fungsi dan implementasi rangkaian logika. Bagian kedua membahas tentang contoh desain rangkaian logika untuk menjalankan fungsi generator paritas genap dan multiplekser 4 kanal masukan. Contoh desain tersebut menjelaskan aplikasi dari sintesis rangkaian logika.

Kuliah ini diberikan kepada mahasiswa S1 semester 2. Materi online dapat didownload di <http://didik.blog.undip.ac.id>, kategori akademik

Metodologi perancangan sistem digital meliputi penentuan kebutuhan spesifikasi (fungsional) dan konstrain yang harus ditaati, melakukan perancangan, implementasi dan pengujian sistem

Perancangan Rangkaian Logika

Sintesis ini adalah proses untuk membangkitkan rangkaian logika dari deskripsi perilaku fungsional yang diinginkan. Dalam perancangan sistem digital, sintesis merupakan proses mapping rancangan RTL

(*register-transfer level*) yang mendeskripsikan perilaku sistem digital ke dalam netlist level gerbang logika sesuai teknologi chip target yang digunakan. Bagian ini menjabarkan tentang langkah-langkah sintesis tersebut, yaitu merepresentasikan fungsi sistem ke dalam tabel kebenaran dan ekspresi logikanya, menyederhanakan ekspresi logika dan mengimplementasikannya sesuai konstrain teknologi chip digital yang digunakan. Dalam artikel ini, rangkaian logika mengacu pada rangkaian gerbang-gerbang logika sebagai implementasi dari fungsi suatu sistem digital.

Kebutuhan Sistem

Perancangan sistem digital dimulai dengan adanya kebutuhan (*requirement*) sistem. Kebutuhan ini antara lain menjabarkan fungsi apa yang harus dipenuhi oleh sistem. Dalam hal ini, kebutuhan fungsional menggambarkan suatu fungsi logika dari sistem. Kebutuhan ini seringkali dideskripsikan menggunakan bahasa 'verbal'. Contoh deskripsi perilaku fungsional sistem secara verbal adalah:

Dalam sebuah sistem pengontrol terdapat 3 buah saklar untuk mengontrol nyala mesin. Mesin menyala hanya jika saklar 1 tersambung dan salah satu (atau kedua) saklar 2 atau 3 tersambung. Keadaan saklar lainnya, mesin akan mati. Desain rangkaian logika untuk sistem tersebut.

Dari deskripsi di atas terdapat 3 keadaan bebas dan 1 keadaan tak bebas/terikat. Keadaan bebasnya adalah tersambung/terputusnya 3 buah saklar, sedangkan keadaan tak bebas adalah mesin menyala. Keadaan bebas ini dapat dinyatakan sebagai variabel masukan (x_1, x_2, x_3) masing-masing untuk saklar 1, 2 dan 3. Dengan asumsi sistem adalah logika positif, maka x_1, x_2 dan x_3 akan bernilai 1 jika mereka tersambung yang memberikan sinyal tegangan *high* atau sebaliknya. Keadaan tak bebas dinyatakan sebagai variabel keluaran (y atau $f(x_1, x_2, x_3)$), yaitu $y = 1$ saat mesin menyala karena mendapat tegangan *high* atau sebaliknya.

NOTE Dari bahasa verbal di atas sebenarnya kita bisa langsung menyatakan fungsi tersebut sebagai $AND(x_1, OR(x_2, x_3))$ atau dengan ekspresi logika dinyatakan dengan $x_1(x_2 + x_3)$. Urutan langkah-langkah yang akan dijabarkan di bawah ini merupakan penjelasan prosedur formal untuk mensintesis rangkaian logika. Dalam banyak kasus, kebutuhan sistem tidak sesederhana seperti contoh di atas, sehingga penjelasan di bawah dapat menjadi panduan untuk melakukan sintesis.

Dalam beberapa kasus, sintesis ini meliputi proses perancangan dan implementasi sistem dalam rangkaian logika

Implementasi dapat menggunakan susunan gerbang logika AND-OR, OR-AND, NAND-NAND dan NOR-NOR. Di PAL dan CPLD, susunan gerbang AND-OR digunakan

Terdapat 4 cara untuk merepresentasikan suatu fungsi logika, yaitu bahasa verbal, tabel kebenaran, ekspresi logika dan diagram pewaktuan. Deskripsi fungsi dalam tabel kebenaran dan ekspresi logika akan dilakukan sebagai langkah penerjemahan kebutuhan sistem (ekstraksi bahasa verbal). Diagram pewaktuan akan lebih banyak digunakan untuk verifikasi dari rangkaian baik saat simulasi maupun pengujian.

Keadaan bebas berarti bisa mempunyai nilai sebarang, tidak ditentukan oleh sistem. Keadaan tak bebas berarti nilainya tergantung dari fungsi sistem dan masukan yang diberikan ke sistem

Dalam logika positif, tegangan *high* diberikan nilai 1 dan tegangan *low* diberikan nilai 0

Variabel f dan $f(x_1, x_2, x_3)$ selanjutnya menyatakan hal yang sama

Representasi Fungsi

Setelah tiap keadaan dari sistem dinyatakan sebagai variabel, langkah berikutnya adalah menyatakan fungsi tersebut secara formal dalam tabel kebenaran, dan selanjutnya menuliskan ekspresi logikanya. Bentuk ekspresi logika dari fungsi dapat dinyatakan dalam 2 bentuk yaitu SOP (*sum-of-product*) dan POS (*product-of-sum*). Bentuk ini akan menentukan susunan implementasi rangkaian logikanya, yaitu AND-OR untuk SOP dan OR-AND untuk POS.

Tabel Kebenaran

Tabel kebenaran berisi daftar nilai keluaran fungsi logika dari semua kombinasi nilai masukan-masukannya. Fungsi logika Tabel kebenaran untuk menggambarkan fungsi logika sesuai kebutuhan di atas ditunjukkan dalam Tabel 1.

Ekspresi SOP (*Sum-of-Product*)

Ekspresi logika menyatakan fungsi ini menggunakan variabel beserta operatornya. Contoh ekspresi logika adalah $\bar{x}_1 + x_1x_2$. Ekspresi ini seringkali digabungkan dengan ekspresi lain (atau menggunakan satu variabel keluaran) yang membentuk persamaan logika. Contoh persamaan: $y = \bar{x}_1 + x_1x_2$ atau $f(x_1, x_2) = \bar{x}_1 + x_1x_2$.

Untuk sebuah fungsi dengan n buah variabel $f(x_1, x_2, \dots, x_n)$, sebuah *minterm* dari f adalah satu **term perkalian** dari n variabel yang ditampilkan sekali, baik dalam bentuk tidak diinverskan maupun diinverskan.

Tiap baris dari tabel kebenaran membentuk satu buah minterm. Jika diberikan satu baris dalam tabel kebenaran, minterm dibentuk dengan menuliskan variabel x_i jika $x_i = 1$ atau \bar{x}_i jika $x_i = 0$. Notasi m_j merupakan minterm dari baris nomor j di tabel kebenaran. Minterm-minterm untuk fungsi dengan 3 variabel masukan dapat dilihat di Tabel 2. Terlihat bahwa nilai j merupakan nilai desimal dari nilai biner $x_2x_1x_0$. Contoh, jika $x_1 = 0$, $x_2 = 0$, $x_3 = 0$, maka minterm $m_0 = \bar{x}_1\bar{x}_2\bar{x}_3$. Minterm $m_1 = \bar{x}_1\bar{x}_2x_3$ adalah minterm baris kedua ($j = 1$) dengan $x_1 = 0$, $x_2 = 0$, $x_3 = 1$.

Fungsi logika $f(x_1, x_2, x_3)$ dapat dinyatakan dengan ekspresi penjumlahan dari semua minterm di mana tiap minterm di-AND-kan dengan nilai $f(x_1, x_2, x_3)$ yang bersesuaian. Bentuk ekspresi yang dihasilkan disebut ekspresi SOP (*sum-of-product*) karena merupakan penjumlahan dari term-term perkalian. Persamaan SOP dari Tabel 2 dapat dinyatakan sebagai berikut:

Lebih lanjut nanti bisa diwujudkan dengan gerbang NAND-NAND dan NOR-NOR

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Tabel 1: Representasi fungsi logika dalam tabel kebenaran

Perlu pemahaman istilah antara ekspresi dan persamaan logika. Operator yang digunakan adalah $+$ (fungsi OR) dan \cdot (fungsi AND)

Dalam aljabar Boolean, term perkalian adalah fungsi AND dan term penjumlahan adalah fungsi OR

j	x_1	x_2	x_3	minterm m_j	f
0	0	0	0	$m_0 = \bar{x}_1\bar{x}_2\bar{x}_3$	0
1	0	0	1	$m_1 = \bar{x}_1\bar{x}_2x_3$	0
2	0	1	0	$m_2 = \bar{x}_1x_2\bar{x}_3$	0
3	0	1	1	$m_3 = \bar{x}_1x_2x_3$	0
4	1	0	0	$m_4 = x_1\bar{x}_2\bar{x}_3$	0
5	1	0	1	$m_5 = x_1\bar{x}_2x_3$	1
6	1	1	0	$m_6 = x_1x_2\bar{x}_3$	1
7	1	1	1	$m_7 = x_1x_2x_3$	1

Tabel 2: Tabel kebenaran beserta mintermnya untuk tiap baris tabel

$$\begin{aligned}
 f &= m_0 \cdot 0 + m_1 \cdot 0 + m_2 \cdot 0 + m_3 \cdot 0 + m_4 \cdot 0 + m_5 \cdot 1 + m_6 \cdot 1 + m_7 \cdot 1 \\
 &= m_5 + m_6 + m_7 \\
 &= x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3
 \end{aligned}$$

Persamaan di atas dapat dinyatakan dalam notasi kanoniknya sebagai berikut:

$$\begin{aligned}
 f &= m_5 + m_6 + m_7 \\
 &= \underbrace{x_1 \bar{x}_2 x_3}_{101=5} + \underbrace{x_1 x_2 \bar{x}_3}_{110=6} + \underbrace{x_1 x_2 x_3}_{111=7} \\
 &= \sum m(5, 6, 7)
 \end{aligned}$$

Jadi fungsi $f(x_1, x_2, x_3)$ dapat dinyatakan baik dalam ekspresi SOP maupun notasi kanoniknya sebagai berikut:

$$f = x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3 = \sum m(5, 6, 7)$$

Ekspresi POS (Product-of-Sum)

Jika fungsi f dinyatakan dalam suatu tabel kebenaran, maka fungsi inversnya \bar{f} , dapat dinyatakan dengan penjumlahan minterm dengan $\bar{f} = 1$, yaitu di baris di mana $f = 0$

$$\begin{aligned}
 \bar{f} &= m_0 + m_1 + m_2 + m_3 + m_4 \\
 &= \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 \bar{x}_3
 \end{aligned}$$

Fungsi f dapat diperoleh dengan menginverskan \bar{f} dan mengimplementasikan hukum d'Morgan sebagai berikut:

$$\begin{aligned}
 f &= \overline{\bar{f}} = \overline{m_0 + m_1 + m_2 + m_3 + m_4} \\
 &= \overline{\bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 \bar{x}_3} \\
 &= (\overline{\bar{x}_1 \bar{x}_2 \bar{x}_3}) \cdot (\overline{\bar{x}_1 \bar{x}_2 x_3}) \cdot (\overline{\bar{x}_1 x_2 \bar{x}_3}) \cdot (\overline{\bar{x}_1 x_2 x_3}) \cdot (\overline{x_1 \bar{x}_2 \bar{x}_3}) \\
 &= (x_1 + x_2 + x_3) (x_1 + x_2 + \bar{x}_3) (x_1 + \bar{x}_2 + x_3) (x_1 + \bar{x}_2 + \bar{x}_3) (\bar{x}_1 + x_2 + x_3)
 \end{aligned}$$

Hasil persamaan di atas menjadi dasar untuk menyatakan fungsi f dalam bentuk perkalian semua term penjumlahan, *maxterm*.

Sebuah maxterm dari f adalah satu **term penjumlahan** dari semua variabel yang ditampilkan sekali baik dalam bentuk tidak diinverskan maupun diinverskan

Mengingat kembali, dalam aljabar Boolean term perkalian adalah fungsi AND dan term penjumlahan adalah fungsi OR

Tiap baris dari tabel kebenaran membentuk satu buah maxterm. Jika diberikan satu baris dalam tabel kebenaran, maxterm dibentuk dengan memasukkan variabel x_i jika $x_i = 0$ atau \bar{x}_i jika $x_i = 1$. Notasi M_j merupakan maxterm dari baris nomor j di tabel kebenaran. Maxterm-maxterm untuk fungsi dengan 3 variabel masukan dapat dilihat di Tabel 3. Terlihat bahwa nilai j merupakan nilai desimal dari nilai biner $x_2x_1x_0$. Contoh, jika $x_1 = 0, x_2 = 0, x_3 = 0$, maka maxterm $M_0 = x_1 + x_2 + x_3$. Maxterm $M_1 = x_1 + x_2 + \bar{x}_3$ adalah maxterm baris kedua ($j = 1$) dengan $x_1 = 0, x_2 = 0, x_3 = 1$.

Fungsi logika $f(x_1, x_2, x_3)$ dapat dinyatakan dengan ekspresi perkalian dari semua maxterm di mana tiap maxterm di-**OR**-kan dengan nilai $f(x_1, x_2, x_3)$ yang bersesuaian. Bentuk ekspresi yang dihasilkan disebut ekspresi POS (*product-of-sum*) karena merupakan perkalian dari term-term penjumlahan. Persamaan POS dari Tabel 3 dapat dinyatakan sebagai berikut:

$$\begin{aligned} f &= (M_0 + 0) \cdot (M_1 + 0) \cdot (M_2 + 0) \cdot (M_3 + 0) \cdot (M_4 + 0) \cdot (M_5 + 1) \cdot (M_6 + 1) \cdot (M_7 + 1) \\ &= M_0 + M_1 + M_2 + M_3 + M_4 \\ &= (x_1 + x_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3)(x_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_1 + x_2 + x_3) \end{aligned}$$

Persamaan tersebut sama dengan persamaan yang diturunkan dari *double* inversi bentuk SOP di atas. Notasi kanonik dari ekspresi POS adalah sebagai berikut:

$$\begin{aligned} f &= M_0 + M_1 + M_2 + M_3 + M_4 \\ &= \underbrace{(x_1 + x_2 + x_3)}_{000=0} \underbrace{(x_1 + x_2 + \bar{x}_3)}_{001=1} \underbrace{(x_1 + \bar{x}_2 + x_3)}_{010=2} \underbrace{(x_1 + \bar{x}_2 + \bar{x}_3)}_{011=3} \underbrace{(\bar{x}_1 + x_2 + x_3)}_{100=4} \\ &= \prod M(0, 1, 2, 3, 4) \end{aligned}$$

Jadi fungsi $f(x_1, x_2, x_3)$ dapat dinyatakan baik dalam ekspresi POS maupun notasi kanoniknya sebagai berikut:

$$f = (x_1 + x_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3)(x_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_1 + x_2 + x_3) = \prod M(0, 1, 2, 3, 4)$$

Konversi SOP - POS

Dari kedua bentuk ekspresi baik SOP maupun POS untuk menyatakan suatu fungsi logika, dapat dinyatakan bahwa jika suatu fungsi f ditunjukkan dalam suatu tabel kebenaran, maka ekspresi untuk f dapat diperoleh (disintesis) dengan cara:

1. Melihat semua baris dalam tabel dimana $f=1$, atau
2. Melihat semua baris dalam tabel dimana $f=0$

j	x_1	x_2	x_3	maxterm M_j	f
0	0	0	0	$M_0 = x_1 + x_2 + x_3$	0
1	0	0	1	$M_1 = x_1 + x_2 + \bar{x}_3$	0
2	0	1	0	$M_2 = x_1 + \bar{x}_2 + x_3$	0
3	0	1	1	$M_3 = x_1 + \bar{x}_2 + \bar{x}_3$	0
4	1	0	0	$M_4 = \bar{x}_1 + x_2 + x_3$	0
5	1	0	1	$M_5 = \bar{x}_1 + x_2 + \bar{x}_3$	1
6	1	1	0	$M_6 = \bar{x}_1 + \bar{x}_2 + x_3$	1
7	1	1	1	$M_7 = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$	1

Tabel 3: Tabel kebenaran beserta maxtermnya untuk tiap baris tabel

Prinsip dualitas dari fungsi AND dan OR

Pendekatan (1) menggunakan minterm, sedangkan pendekatan (2) menggunakan komplemen dari minterm, yaitu maxterm.

Jika suatu fungsi f diberikan dalam bentuk $\sum m$ atau $\prod M$, maka dengan mudah dapat dicari fungsi f atau inversnya \bar{f} dalam bentuk $\sum m$ atau $\prod M$, sebagai berikut:

Bentuk Asal	Fungsi dan Bentuk yang Diinginkan			
	$f = \sum m$	$f = \prod M$	$\bar{f} = \sum m$	$\bar{f} = \prod M$
$f = \sum m(5, 6, 7)$	-	Nomor yg tdk ada dlm daftar $\prod M(0, 1, 2, 3, 4)$	Nomor yang tdk ada dlm daftar $\sum m(0, 1, 2, 3, 4)$	Nomor yang ada dlm daftar $\prod M(5, 6, 7)$
$f = \prod M(0, 1, 2, 3, 4)$	Nomor yg tdk ada dlm daftar $\sum m(5, 6, 7)$	-	Nomor yang ada dlm daftar $\sum m(0, 1, 2, 3, 4)$	Nomor yg tdk ada dlm daftar $\prod M(5, 6, 7)$

Tabel 4: Tabel konversi bentuk SOP dan POS

Contoh konversi bentuk SOP-POS:

- diberikan fungsi $f(x_1, x_2, x_3) = \sum m(0, 2, 3, 6)$. Fungsi tersebut dapat dinyatakan sebagai $f(x_1, x_2, x_3) = \prod M(1, 4, 5, 7)$. Kedua bentuk persamaan tersebut ekuivalen, menjalankan fungsi yang sama, sebagai berikut:

$$\begin{aligned} \sum m(0, 2, 3, 6) &= \prod M(1, 4, 5, 7) \\ \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_2\bar{x}_3 + \bar{x}_1x_2x_3 + x_1x_2\bar{x}_3 &= (x_1 + x_2 + \bar{x}_3)(\bar{x}_1 + x_2 + x_3)(\bar{x}_1 + x_2 + \bar{x}_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \end{aligned}$$

- diberikan fungsi $f(x_1, x_2, x_3, x_4) = \prod M(2, 5, 6, 7, 10, 11, 13)$. Fungsi tersebut dapat dinyatakan sebagai $f(x_1, x_2, x_3, x_4) = \sum m(0, 1, 3, 4, 8, 9, 12, 14, 15)$. Kedua bentuk persamaan tersebut ekuivalen.

Dalil, Teorema dan Hukum Aljabar Boolean

Dalam aljabar Boolean berlaku dalil, teorema, dan hukum yang dapat digunakan untuk melakukan manipulasi ekspresi logika, misalnya untuk menyederhanakan ekspresi logika. Dalam Tabel 5 dituliskan dalil, teorema dan hukum aljabar dalam 2 kolom yang berpasangan (a dan b). Kolom a dan kolom b adalah dual satu sama lainnya. Jika diberikan sebarang ekspresi logika, dual dari ekspresi tersebut dapat dibentuk dengan mengganti semua $+$ dengan \cdot atau sebaliknya serta mengganti 0 dengan 1 atau sebaliknya. Dual dari pernyataan benar adalah juga benar.

Pembuktian teorema dan hukum dapat dilakukan dengan menggunakan dalil atau teorema lain. Cara yang lain adalah dengan menggunakan tabel kebenaran untuk membandingkan keadaan dua ekspresi.

Dalil merupakan pernyataan yang tidak perlu lagi dibuktikan, ia benar dengan sendirinya

Penjelasan tentang metode pembuktian induktif dan grafis dengan diagram Venn dijelaskan dalam LectureNote materi 3

1a. $0 \cdot 0 = 0$	1b. $1 + 1 = 1$	Tabel 5: Dalil, teorema dan hukum aljabar Boolean	
2a. $1 \cdot 1 = 1$	2b. $0 + 0 = 0$		
3a. $0 \cdot 1 = 1 \cdot 0 = 0$	3b. $1 + 0 = 0 + 1 = 1$		
4a. Jika $x = 0$, maka $\bar{x} = 1$	4b. Jika $x = 1$, maka $\bar{x} = 0$		
5a. $x \cdot 0 = 0$	5b. $x + 1 = 1$		
6a. $x \cdot 1 = x$	6b. $x + 0 = x$		
7a. $x \cdot x = x$	7b. $x + x = x$		
8a. $x \cdot \bar{x} = 0$	8b. $x + \bar{x} = 1$		
9. $\overline{\bar{x}} = x$			
10a. $x \cdot y = y \cdot x$	10b. $x + y = y + x$		→Komutatif
11a. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$	11b. $x + (y + z) = (x + y) + z$		→Asosiatif
12a. $x \cdot (y + z) = x \cdot y + x \cdot z$	12b. $x + y \cdot z = (x + y) \cdot (x + z)$		→Distributif
13a. $x + x \cdot y = x$	13b. $x \cdot (x + y) = x$		→Absorpsi
14a. $x \cdot y + x \cdot \bar{y} = x$	14b. $(x + y) \cdot (x + \bar{y}) = x$		→Penggabungan
15a. $\overline{x \cdot y} = \bar{x} + \bar{y}$	15b. $\overline{x + y} = \bar{x} \cdot \bar{y}$		→DeMorgan
16a. $x + \bar{x} \cdot y = x + y$	16b. $x \cdot (\bar{x} + y) = x \cdot y$		
17a. $x \cdot y + y \cdot z + \bar{x} \cdot z = x \cdot y + \bar{x} \cdot z$	17b. $(x + y) \cdot (y + z) \cdot (\bar{x} + z) = (x + y) \cdot (\bar{x} + z)$		→Konsensus

Pembuktian secara grafis dapat dilakukan dengan menggunakan diagram Venn.

Penyederhanaan Ekspresi

Persamaan SOP atau SOP di atas menyatakan fungsi f secara benar dan unik, namun mungkin tidak menghasilkan implementasi rangkaian yang paling sederhana. Suatu fungsi logika dapat dinyatakan dalam beberapa bentuk ekspresi yang ekuivalen. Proses optimasi memilih salah satu dari rangkaian-rangkaian ekuivalen untuk memenuhi constraint nonfungsional, misalnya area, delay dan *cost*.

Proses penyederhanaan rangkaian logika dapat dilakukan dengan mengurangi *minterm* atau *maxterm* di ekspresi SOP atau POS. Salah satu caranya adalah menggunakan hukum, teorema dan dalil aljabar Boolean, yaitu

- dengan menggunakan hukum 14a ($x \cdot y + x \cdot \bar{y} = x$) untuk ekspresi SOP; dan
- dengan menggunakan hukum 14b ($(x + y) \cdot (x + \bar{y}) = x$) untuk ekspresi POS

Prinsip penyederhanaannya adalah dengan menggabungkan beberapa minterm atau maxterm yang **berbeda hanya di satu variabel** saja dan mengaplikasikan hukum 14a atau 14b.

Persamaan SOP $f = \sum m(5, 6, 7)$ dapat disederhanakan sebagai berikut: Ingat: $x_1x_2x_3 + x_1x_2\bar{x}_3 = x_1x_2x_3$

Rangkaian dengan jumlah gerbang minimal bisa jadi bukan merupakan solusi terbaik, tergantung constraintnya. Misalnya constraint delay

Terdapat beberapa cara penyederhanaan, yaitu menggunakan aljabar Boolean, Karnaugh map (K-map) dan metode Quine-McCluskey

$$\begin{aligned}
 f &= \sum m(5,6,7) = x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3 + x_1x_2x_3 \\
 &= (x_1\bar{x}_2x_3 + x_1x_2x_3) + (x_1x_2x_3 + x_1x_2\bar{x}_3) \\
 &= x_1(\bar{x}_2 + x_2)x_3 + x_1x_2(x_3 + \bar{x}_3) \\
 &= x_1x_3 + x_1x_2
 \end{aligned}$$

Minterm 5 dan 7 hanya berbeda di x_2 dan minterm 6 dan 7 berbeda di x_3 . Minterm 7 ditambahkan agar dapat menyederhanakan minterm 5 dan 6.

Persamaan POS $f = \prod M(0,1,2,3,4)$ dapat disederhanakan sebagai berikut:

$$\begin{aligned}
 f &= \prod M(0,1,2,3,4) = (x_1 + x_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3)(x_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_1 + x_2 + x_3) \\
 &= (x_1 + x_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + \bar{x}_2 + x_3)(x_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_1 + x_2 + x_3) \\
 &= (x_1 + x_2 + x_3\bar{x}_3)(x_1 + \bar{x}_2 + x_3\bar{x}_3)(\bar{x}_1x_1 + x_2 + x_3) \\
 &= (x_1 + x_2)(x_1 + \bar{x}_2)(x_2 + x_3) \\
 &= (x_1 + x_2\bar{x}_2)(x_2 + x_3) = x_1(x_2 + x_3)
 \end{aligned}$$

Persamaan $f = x_1x_2 + x_1x_3$ dan $f = x_1(x_2 + x_3)$ yang diturunkan dari penyederhanaan ekspresi SOP dan POS adalah **ekivalen**.

Implementasi Rangkaian

Rangkaian logika diimplementasikan setelah ekspresi dari fungsi logika disederhanakan. Rangkaian logika tersusun dari gerbang-gerbang logika AND, OR, NAND, NOR dan NOT. Bentuk persamaan SOP dan POS dapat diimplementasikan dengan rangkaian logika 2-level, dengan 4 kombinasi susunan gerbang sebagai berikut:

1. Rangkaian AND-OR

Level 1 rangkaian merupakan fungsi AND. Keluaran level 1 menjadi masukan level 2 rangkaian dengan fungsi OR

2. Rangkaian OR-AND

Level 1 rangkaian diwujudkan dengan gerbang OR, sedangkan level 2 dengan gerbang AND

3. Rangkaian NAND-NAND

Level 1 dan 2 diwujudkan dengan gerbang NAND

4. Rangkaian NOR-NOR

Level 1 dan 2 diwujudkan dengan gerbang NOR

Keempat susunan gerbang tersebut **melakukan fungsi yang sama**, yaitu menghasilkan keluaran yang sama untuk sebarang masukan.

Rangkaian dengan variabel banyak menjadi isu implementasi di FPGA yang menggunakan LUT (Lookup Table) 2 masukan. Hal ini diatasi dengan teknik sintesis multilevel, di mana rangkaian tersusun lebih dari 2 level

Terdapat 2 tipe simbol gerbang: tradisional dan IEEE/IEC. Kedua-duanya digunakan dalam artikel ini

Susunan Gerbang	Diturunkan dari
AND-OR	ekspresi SOP
OR-AND	ekspresi POS
NAND-NAND	ekspresi SOP atau rangkaian AND-OR
NOR-NOR	ekspresi POS atau rangkaian NOR-NOR

Tabel 6: Susunan gerbang rangkaian logika 2 level

Rangkaian AND-OR

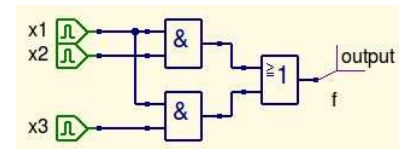
Rangkaian AND-OR dapat diperoleh dari penyederhanaan bentuk ekspresi SOP. Rangkaian AND-OR yang melakukan fungsi logika sesuai spesifikasi ditunjukkan dalam Gambar 1.

Rangkaian OR-AND

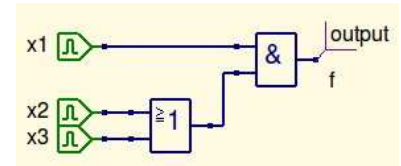
Rangkaian OR-AND dapat diperoleh dari penyederhanaan bentuk ekspresi POS. Rangkaian OR-AND yang melakukan fungsi logika sesuai spesifikasi ditunjukkan dalam Gambar 2.

Rangkaian NAND-NAND

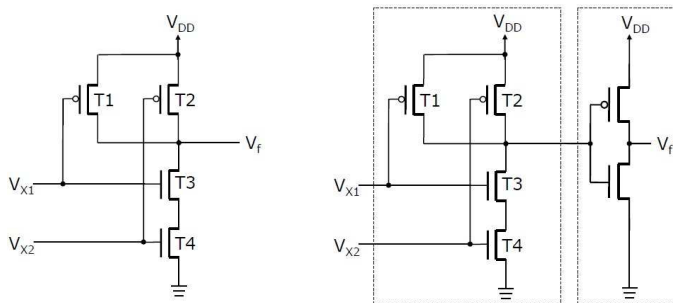
Rangkaian NAND-NAND diperoleh dengan mengaplikasikan teorema DeMorgan di rangkaian AND-OR atau menurunkan dari ekspresi SOP. Penggunaan gerbang NAND-NAND untuk ekspresi SOP lebih disukai. Hal ini disebabkan misalnya saat implementasi dengan



Gambar 1: Rangkaian logika AND-OR dari persamaan $f = x_1x_3 + x_1x_2$



Gambar 2: Rangkaian logika OR-AND dari persamaan $f = x_1(x_2 + x_3)$



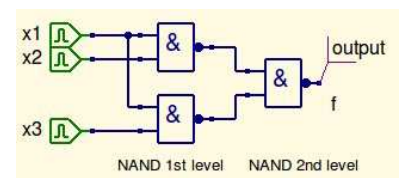
CMOS NAND (4 transistor)

CMOS AND (6 transistor)

teknologi CMOS, gerbang NAND ini lebih sederhana daripada gerbang AND. Gerbang NAND diimplementasikan dengan 4 transistor MOSFET, sedangkan AND dengan 6 transistor.

Rangkaian NAND-NAND dapat dibentuk dari rangkaian AND-OR dengan cara mengganti gerbang AND dan OR dengan NAND sesuai dengan teorema DeMorgan. Dengan mengaplikasikan teorema DeMorgan terhadap persamaan SOP yang telah disederhanakan, didapat persamaan NAND-NAND berikut:

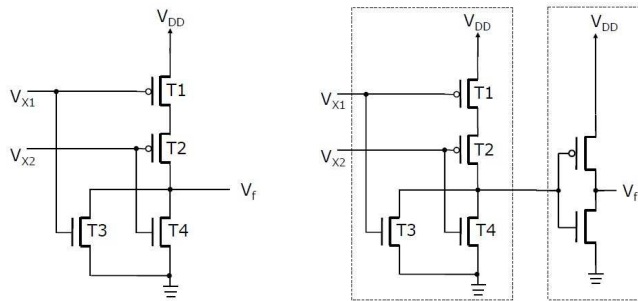
$$\begin{aligned}
 f &= x_1x_2 + x_1x_3 \\
 &= \overline{\overline{x_1x_2} \cdot \overline{x_1x_3}} \\
 &\quad \underbrace{\text{NAND} \quad \text{NAND}} \\
 &\quad \text{NAND-2nd level}
 \end{aligned}$$



Gambar 4: Rangkaian logika NAND-NAND dari persamaan $f = \overline{\overline{x_1x_2} \cdot \overline{x_1x_3}}$

Rangkaian NOR-NOR

Rangkaian NOR-NOR diperoleh dengan mengaplikasikan teorema De-Morgan di rangkaian OR-AND atau menurunkan dari ekspresi POS. Seperti halnya rangkaian NAND-NAND, rangkaian NOR-NOR untuk ekspresi POS lebih disukai daripada OR-AND. Hal ini disebabkan



CMOS NOR (4 transistor) CMOS OR (6 transistor)

karena di implementasi dengan teknologi CMOS, gerbang NOR lebih sederhana daripada gerbang OR. Gerbang NOR diimplementasikan dengan 4 transistor MOSFET, sedangkan OR dengan 6 transistor.

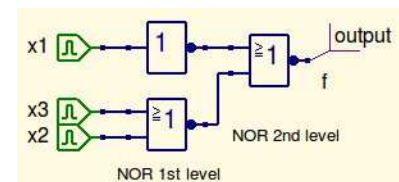
Rangkaian NOR-NOR dapat dibentuk dari rangkaian OR-AND dengan cara mengganti gerbang OR dan AND dengan NOR sesuai dengan teorema DeMorgan. Dengan mengaplikasikan teorema DeMorgan terhadap persamaan POS yang telah disederhanakan, didapat persamaan berikut NOR-NOR berikut:

$$\begin{aligned}
 f &= x_1 (x_2 + x_3) \\
 &= \underbrace{\overline{\overline{x_1}}}_{\text{NOT}} + \underbrace{\overline{(x_2 + x_3)}}_{\text{NOR}} \\
 &\quad \text{NOR-2nd level}
 \end{aligned}$$

Pengujian Fungsional

Dalam tahap pengujian fungsional, semua kemungkinan kondisi masukan diumpangkan ke rangkaian. Keluaran rangkaian dibandingkan dengan keluaran yang diharapkan. Daftar nilai-nilai masukan beserta nilai keluarannya yang diharapkan disebut sebagai *test pattern* (pola uji). Pengujian *exhaustive* akan menguji rangkaian dengan semua kemungkinan nilai masukan. Pengujian tipe ini akan membutuhkan waktu untuk rangkaian dengan masukan banyak dan mempunyai fungsi kompleks. Terdapat upaya untuk membuat pola uji yang optimal. Faktor yang berpengaruh adalah *fault coverage* dari pola uji. Fault coverage menunjukkan persentasi kesalahan yang mungkin dapat terdeteksi oleh pola uji.

Gambar 5: Gerbang NOR lebih sederhana dari OR



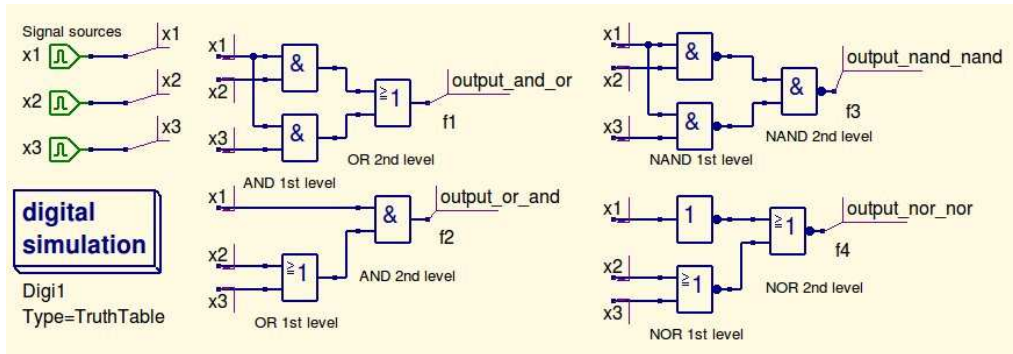
Gambar 6: Rangkaian logika NOR-NOR dari persamaan $f = \overline{\overline{x_1} + (x_2 + x_3)}$

Pola uji dibangkitkan oleh software ATPG (*automatic test pattern generator*) menggunakan algoritma tertentu untuk meningkatkan *fault coverage* dengan jumlah pola uji minimum

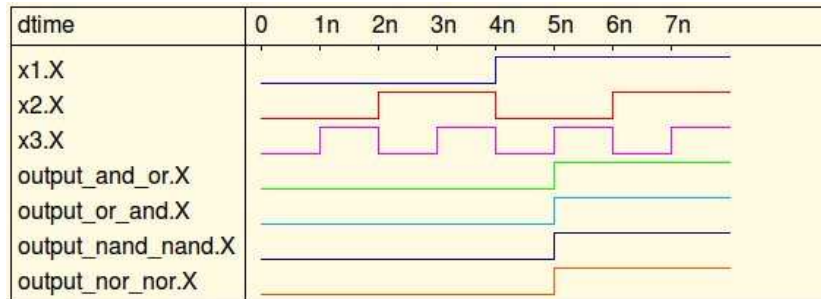
Dalam artikel ini akan dibahas tentang pengujian fungsional rangkaian dengan melakukan analisis menggunakan diagram pewaktuan. Analisis dilakukan dengan memberikan semua kemungkinan masukan dan mengamati keluarannya untuk rangkaian yang telah dibangkitkan (*exhaustive testing*), baik AND-OR, OR-AND, NAND-NAND dan NOR-NOR. Untuk melakukan pengujian, rangkaian diimplementasikan di program Qucs (Quite Universal Circuit Simulator). Keempat bentuk rangkaian disimulasikan dan dianalisis diagram pewaktuannya, seperti diperlihatkan dalam Gambar 7 dan Gambar 8.

Diagram pewaktuan menyatakan perilaku dinamis dari fungsi secara grafis sepanjang waktu t , berupa sinyal (*waveform*) keluaran

Rangkaian disimulasikan dengan program. Simulasi sendiri sebenarnya adalah tahap verifikasi setelah desain, bukan implementasi. Di artikel ini simulasi lebih mengacu ke emulasi dari rangkaian logika nyatanya



Gambar 7: Emulasi rangkaian logika di program simulator Qucs



Gambar 8: Diagram pewaktuan untuk semua rangkaian logika secara *exhaustive*

Contoh Desain

Di bagian ini akan dijabarkan contoh-contoh desain rangkaian logika (sintesis). Langkah-langkah yang dilakukan mengikuti prosedur yang dijelaskan sebelumnya. Langkah sintesis tersebut dapat dirangkum sebagai berikut:

1. Menuliskan spesifikasi formal dalam bentuk tabel kebenaran;
2. Menuliskan ekspresi logika dari tabel kebenaran dalam bentuk SOP atau POS;

3. Menyederhanakan ekspresi logika dengan menggunakan aljabar Boolean;
4. Mengimplementasikan rangkaian logika sesuai teknologi yang akan digunakan, yaitu AND-OR, OR-AND, NAND-NAND atau NOR-NOR;

Rangkaian logika yang akan didesain adalah rangkaian generator paritas dan rangkaian multiplexer 4 kanal.

Rangkaian Generator Paritas

Desain rangkaian generator paritas genap dari data masukan 4-bit!

Deskripsi Formal

Dari kebutuhan desain di atas, terdapat 4 masukan yang dapat dinyatakan sebagai variabel x_1, x_2, x_3, x_4 . Untuk tiap kombinasi nilai variabel masukan, rangkaian yang diinginkan harus mempunyai keluaran 1 jika variabel masukan yang bernilai 1 berjumlah ganjil. Kondisi lainnya, keluaran akan 0. Misalnya jika masukan $\{x_1, x_2, x_3, x_4\}$ diberi nilai $\{0, 0, 1, 0\}$, maka keluaran akan 1, sedangkan jika diberi masukan $\{0, 1, 0, 1\}$, maka keluaran akan 0.

Bentuk formal spesifikasi tersebut di atas secara lengkap dapat dinyatakan dalam tabel kebenaran, seperti ditunjukkan dalam Tabel 7.

Penerjemahan deskripsi kebutuhan dalam deskripsi formal akan menentukan proses desain seterusnya. Interpretasi yang salah dari kebutuhan menyebabkan desain keseluruhan akan salah.

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$	x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
0	0	0	0	0	1	0	0	0	1
0	0	0	1	1	1	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	0	1	1	0	1	0	1	1	1
0	1	0	0	1	1	1	0	0	0
0	1	0	1	0	1	1	0	1	1
0	1	1	0	0	1	1	1	0	1
0	1	1	1	1	1	1	1	1	0

Tabel 7: Spesifikasi formal rangkaian generator paritas genap data 4-bit

Ekspresi Logika

Dari Tabel 7 dapat dituliskan persamaan SOP dan POS sebagai berikut:

$$y = \sum m(1, 2, 4, 7, 8, 11, 13, 14) \text{ atau } y = \prod M(0, 3, 5, 6, 9, 10, 12, 15)$$

Penyederhanaan Ekspresi

Ekspresi SOP akan disederhanakan dengan menggunakan aljabar Boolean. Di sini belum dibahas tentang K-map dan metode Quine-McKluskey

$$\begin{aligned}
 y &= \sum m(1, 2, 4, 7, 8, 11, 13, 14) \\
 &= \bar{x}_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3x_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2x_3x_4 + x_1x_2\bar{x}_3x_4 + x_1x_2x_3\bar{x}_4
 \end{aligned}$$

Persamaan tidak dapat disederhanakan karena **tidak ada minterm** yang hanya mempunyai 1 perbedaan nilai variabel.

Rangkaian SOP standar bisa jadi merupakan rangkaian yang minimal

Rangkaian Logika

Rangkaian logika untuk generator paritas genap diwujudkan dengan rangkaian AND-OR SOP standar, yang terdiri atas 8 gerbang AND 4-masukan, 1 gerbang OR 8-masukan dan 16 gerbang NOT.

Rangkaian Multiplekser (Selektor)

Desain rangkaian multiplekser (selektor) 4 kanal masukan!

Deskripsi Formal

Dalam multiplekser terdapat masukan 4 kanal $\{x_1, x_2, x_3, x_4\}$ dan masukan kontrol 2 bit $\{s_0, s_1\}$. Perilaku multiplekser dapat dituliskan dengan persamaan berikut:

$$y = \begin{cases} x_1 & ; \text{jika } \{s_0, s_1\} = \{0, 0\} \\ x_2 & ; \text{jika } \{s_0, s_1\} = \{0, 1\} \\ x_3 & ; \text{jika } \{s_0, s_1\} = \{1, 0\} \\ x_4 & ; \text{jika } \{s_0, s_1\} = \{1, 1\} \end{cases}$$

Bentuk formal spesifikasi tersebut di atas secara lengkap dapat dinyatakan dalam tabel kebenaran, seperti ditunjukkan dalam Tabel 8 .

x_1	x_2	x_3	x_4	s_0	s_1	$f(x_1, x_2, x_3, x_4, s_0, s_1)$
0	D	D	D	0	0	0
1	D	D	D	0	0	1
D	0	D	D	0	1	0
D	1	D	D	0	1	1
D	D	0	D	1	0	0
D	D	1	D	1	0	1
D	D	D	0	1	1	0
D	D	D	1	1	1	1

Tabel 8: Spesifikasi formal rangkaian multiplekser 4 kanal masukan

Notasi D menyatakan kondisi *don't care*, sebarang nilai masukan tidak akan mempengaruhi keluaran fungsi. Misalnya: jika $\{s_0, s_1\} = \{0, 0\}$ maka nilai f hanya tergantung dari nilai x_1 , sehingga nilai x_2, x_3, x_4 menjadi kondisi *don't care*.

Ekspresi Logika

Persamaan logika SOP dari Tabel 8 dapat dituliskan secara langsung dengan mengambil minterm yang menghasilkan keluaran 1 sebagai berikut:

$$y = f(x_1, x_2, x_3, x_4, s_0, s_1) = x_1\bar{s}_0\bar{s}_1 + x_2\bar{s}_0s_1 + x_3s_0\bar{s}_1 + x_4s_0s_1$$

Ekspresi logika POS dapat dituliskan dengan mengambil maxterm yang menghasilkan keluaran 0 sebagai berikut:

$$y = f(x_1, x_2, x_3, x_4, s_0, s_1) = (x_1 + s_0 + s_1)(x_2 + s_0 + \bar{s}_1)(x_3 + \bar{s}_0 + s_1)(x_4 + \bar{s}_0 + \bar{s}_1)$$

Don't care diabaikan

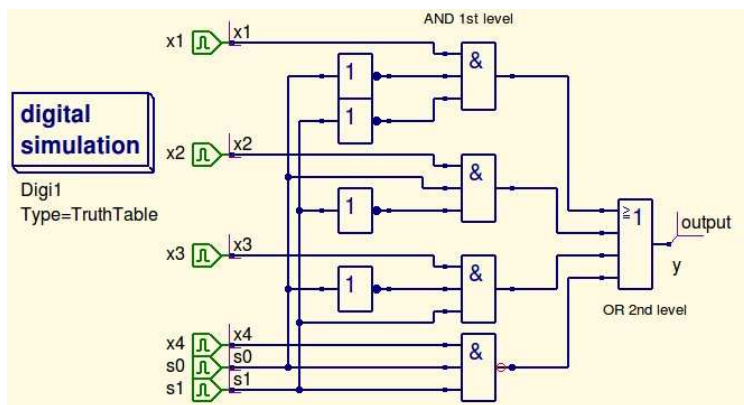
Ingat aturan penulisan maxterm!

Penyederhanaan Ekspresi

Bentuk persamaan SOP dan POS di atas merupakan persamaan paling sederhana dari multiplekser.

Rangkaian Logika

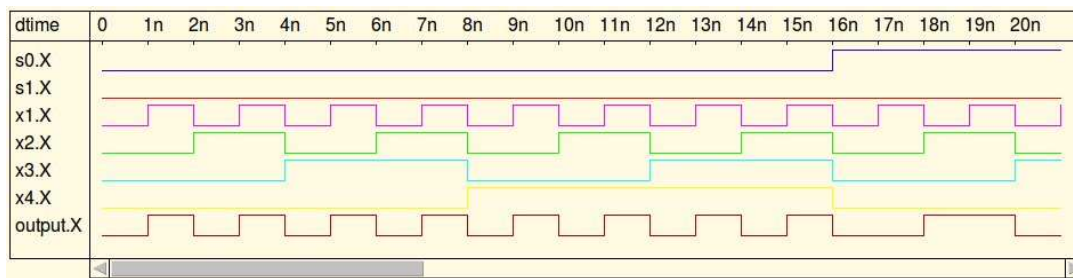
Rangkaian logika multiplekser dapat diwujudkan dengan rangkaian AND-OR, OR-AND, NAND-NAND dan NOR-NOR. Rangkaian multiplekser 4 kanal dengan gerbang AND-OR ditunjukkan Gambar 9. Implementasi dengan susunan gerbang yang lain disediakan untuk latihan.



Gambar 9: Rangkaian multiplekser 4 kanal masukan menggunakan gerbang AND-OR

Analisis Rangkaian

Perilaku rangkaian AND-OR dapat dilihat dari diagram pewaktuan seperti yang ditunjukkan dalam Gambar 10. Dari diagram pewaktuan dapat dilihat bahwa keluaran akan 'mengikuti' masukan sesuai keadaan $\{s_0, s_1\}$, misalnya saat $\{s_0, s_1\} = \{0, 0\}$ sinyal keluaran 'mengikuti' sinyal masukan x_1 .



Gambar 10: Diagram pewaktuan rangkaian multiplexer 4 kanal masukan