

Perancangan Layout IC CMOS Dengan Efisiensi Ruang Yang Optimal Pada Register Geser SIPO dan INVERTER Sebagai Penggerak Motor Stepper

Harry Sudibyo S.¹ dan Pipit Anggraeni²

¹Departemen Teknik Elektro, Fakultas Teknik Universitas Indonesia, Depok 16424.

²Jurusan Teknik Otomasi Manufaktur dan Mekatronika · POLMAN Bandung, Jl. Kanayakan 21 Bandung
email : harisudi@eng.ui.ac.id

Abstrak

Motor stepper merupakan aktuator yang kita ketahui banyak diaplikasikan dalam sistem komputer. Penggunaannya, terutama untuk disk drive, tidak memerlukan torsi besar dan hanya mengatur sudut-sudut tertentu untuk mendapatkan posisi head yang dikehendaki.

Untuk menggerakkan motor stepper diperlukan masukan data paralel (umumnya ada empat masukan ditambah ground). Masing-masing masukan mendapat sinyal yang berbeda tapi berurutan satu sama lain. Masing-masing masukan mendapat sinyal pulsa yang berasal dari rangkaian driver / switching yang biasanya dibangun dari transistor. Umumnya perintah (sinyal masukan) untuk menggerakkan motor stepper tersebut dikeluarkan oleh komputer dalam bentuk data seri. Sehingga diperlukan satu rangkaian untuk mengubah data seri tersebut menjadi data paralel yang bisa dibaca oleh motor stepper. Rangkaian tersebut adalah SIPO (Serial In Parallel Out).

Model ini mencoba untuk membuat rancangan layout CMOS dari paduan antara rangkaian SIPO dengan driver stepper. Layout SIPO ini dibangun dari beberapa rangkaian D flip-flop. Layout driver kemudian dicoba untuk digabungkan dengan layout SIPO dalam satu layout dalam beberapa alternatif rangkaian.

Dari alternatif-alternatif layout yang dibuat, luas layout dan tanggapan waktunya kemudian dibandingkan untuk mendapatkan hasil yang terbaik. Perancangan layout ini menggunakan software "Magic CAD" untuk menggambar layout rancangan dan "IRSIM" untuk mensimulasikan hasil rancangan. Proses penggambaran layout dan simulasi hasil rancangan ini dilakukan di Lab. Elektronika Fakultas Teknik UI selama bulan Desember 2003 sampai Januari 2004.

Kata Kunci : CMOS, Layout, Stepper

Abstract

Stepper motor is actuator as we knew has a lot of application in computer system field. The applications, especially for disk drive, doesn't need large torque and only set a certain angle to achieve a certain position.

To drive the motor stepper we need parallel data input (usually 4 inputs and 1 ground). Each of inputs gets sequencely different signal. Each of them has pulse signal that comes from driver/switching circuit which be built by transistors. The command (input signal) to drive that motor stepper usually is yielded by computer in serial data form. Then we need one circuit to convert serial data to be a parallel data that can be read by stepper motor. The circuit is called SIPO (Serial In Parallel Out).

This paper is trying to make a CMOS layout design of SIPO and stepper driver combination. The SIPO layout is built from some D flip-flop. Then driver layout is combined with SIPO in one layout within several circuit.

From this several layout alternative, the layout wide and time respon will be compared to get the best result. This layout design use "Magic CAD" software for layout drawing and "IRSIM" to simulate the design result. Layout drawing process and result of design simulation had done in the Laboratory of Electronic of Engineering Faculty of Indonesia University since December 2003 until January 2004.

Keywords : CMOS, Layout, Stepper

1. Pendahuluan

Motor *stepper* banyak digunakan untuk aplikasi-aplikasi yang biasanya cukup menggunakan torsi yang kecil. Penggunaannya juga cukup sederhana dan mudah digunakan untuk aplikasi-aplikasi tertentu yang tidak membutuhkan torsi besar. Dalam hal kecepatan, kecepatan motor *stepper* cukup cepat jika dibandingkan dengan motor DC [1].

Pengontrolan motor *stepper* dapat dilakukan oleh sebuah rangkaian digital bahkan sebuah komputer melalui *serial port* atau *parallel port*. Tetapi pada umumnya *parallel port* banyak digunakan untuk printer. Dengan menggunakan *serial port* dari komputer maka diperlukan perangkat lunak yang nantinya akan mengatur pemberian data dengan suatu urutan-urutan tertentu kepada komponen kontrol motor *stepper* [2]. Permasalahan yang timbul nantinya adalah keluaran dari *serial port* komputer merupakan data seri sedangkan untuk menggerakkan motor *stepper* kita memerlukan sinyal masukan paralel. Maka untuk mengubah data seri dari komputer menjadi data paralel, sehingga dapat digunakan untuk masukan motor *stepper*, digunakan sebuah rangkaian register geser yang dapat mengubah data masukan seri menjadi data keluaran paralel. Register geser ini dapat dibangun dari rangkaian D *flip-flop*.

Tujuan pembahasan ini adalah untuk mendapatkan bentuk *layout* IC CMOS dengan efisiensi ruang yang optimal dari alternatif-alternatif *layout* hasil paduan antara rangkaian penggerak motor *stepper* dengan rangkaian register geser SIPO (*Serial In Parallel Out*) yang digunakan untuk mengubah data seri dari komputer menjadi data paralel untuk memberi masukan data pada motor *stepper*.

Tulisan ini membahas rancangan *layout* IC CMOS untuk rangkaian penggerak motor *stepper* dengan register geser SIPO sebagai perangkat yang mengubah data seri dari keluaran komputer menjadi sinyal paralel untuk masukan motor *stepper*. Tipe motor *stepper* yang digunakan adalah motor

stepper variabel reluktansi 4 masukan yang akan menghasilkan gerakan putar 30° tiap stepnya.

Motor *stepper* ini nantinya bergerak 120° setiap pemberian satu pulsa pada data masukan. Walaupun begitu putarannya dapat dihentikan dengan memberikan data RESET.

Layout yang dibuat bukan merupakan *layout* lengkap dalam arti *layout* dalam tulisan ini hanya terdiri dari *polysilicon*, metal, daerah difusi serta kontak.

2. Motor Stepper

Motor *stepper* merupakan motor listrik yang tidak mempunyai komutator, dimana semua lilitannya merupakan bagian dari stator [2]. Bagian rotornya hanya merupakan magnet permanen. Dengan model motor seperti ini maka motor *stepper* dapat diatur posisinya pada sudut-sudut tertentu dan berputar kearah yang diinginkan, searah jarum jam atau sebaliknya [1].

Motor *stepper* adalah sebuah peralatan digital [3] karena motor *stepper* memiliki kemampuan untuk mengikuti perintah digital secara tepat seperti yang diperintahkan komputer. Motor *stepper* dikendalikan oleh pulsa digital. Deretan pulsa akan diterjemahkan menjadi putaran motor dimana tiap putaran terjadi setiap pemberian pulsa. Setiap satu pulsa sama artinya dengan gerakan putar satu *step*.

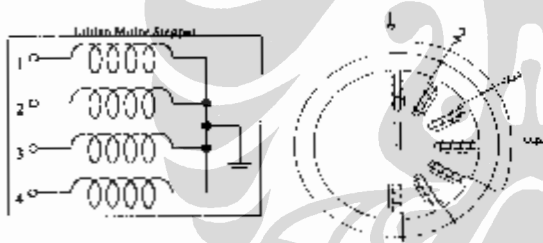
Kecepatan motor *stepper* pada dasarnya ditentukan oleh frekuensi pemberian data pada komutasi lilitannya. Semakin cepat data yang diberikan maka motor *stepper* akan semakin cepat pula berputarnya. Pada kebanyakan motor *stepper*, kecepatan datanya diatur dalam daerah frekuensi audio dan akan menghasilkan putaran yang cukup cepat [1].

Motor *stepper* dibedakan menjadi dua macam berdasarkan magnet yang digunakan, yaitu tipe magnet permanen dan variabel reluktansi. Pada umumnya motor *stepper* yang saat ini banyak digunakan adalah yang mempunyai variabel reluktansi

Cara yang paling mudah untuk membedakan antara tipe motor *stepper* diatas adalah dengan cara memutar rotor menggunakan tangan ketika tidak dihubungkan dengan sumber tegangan [1].

Motor *stepper* yang menggunakan magnet permanen ketika diputar menggunakan tangan akan terasa lebih tersendat karena adanya gaya yang ditimbulkan oleh magnet permanen. Sedangkan pada motor yang menggunakan variabel reluktansi ketika diputar menggunakan tangan maka akan terasa lebih halus karena sisa reluktansinya cukup kecil [1].

Untuk motor *stepper* jenis variabel reluktansi dengan 4 buah lilitan yang pada ujung-ujungnya dijadikan satu dengan *ground*, cara menggerakkan motor ini adalah dengan mengaktifkan masing-masing lilitan sesuai dengan urutannya [1]. Gambar 1 merupakan gambar struktur motor variabel reluktansi



Gambar 1. Motor Variabel Reluktansi [9]

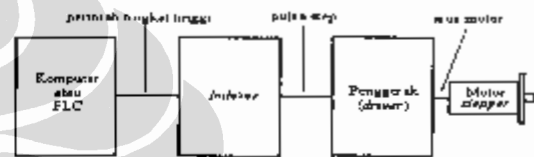
Jika lilitan 1 dilewati oleh arus, lilitan 2, 3, dan 4 mati maka kumparan 1 akan menghasilkan gaya tolak pada rotor dan rotor akan berputar sejauh 30° searah jarum jam. Sehingga kutub rotor akan sejajar dengan kutub dengan lilitan 2.

Jika kondisi ini berulang terus menerus secara berurutan, lilitan 2 dilewati arus kemudian lilitan 3 lalu lilitan 4, maka motor akan berputar mengikuti arah urutan aktifasi lilitan. Urutan pemberian data adalah sebagai berikut:

Lilitan 1: 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
 Lilitan 2: 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
 Lilitan 3: 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
 Lilitan 4: 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0

'1' diartikan bahwa lilitan yang bersangkutan dilewati arus sehingga menghasilkan gaya tolak untuk rotor. Sedangkan '0' diartikan lilitan dalam kondisi mati atau tidak mendapatkan arus.

Sistem motor *stepper* terdiri dari tiga dasar elemen, sering juga digabung dengan beberapa macam antarmuka pengguna (komputer atau PLC) [4]. Jumlah, kecepatan, dan arah putaran sebuah motor *stepper* ditentukan oleh perangkat kontrol digital yang sesuai. Jenis perangkat kontrol digital yang utama adalah pengontrol-pengontrol Penggerak motor. Perangkat ini bekerja seperti pada gambar 2 [3].

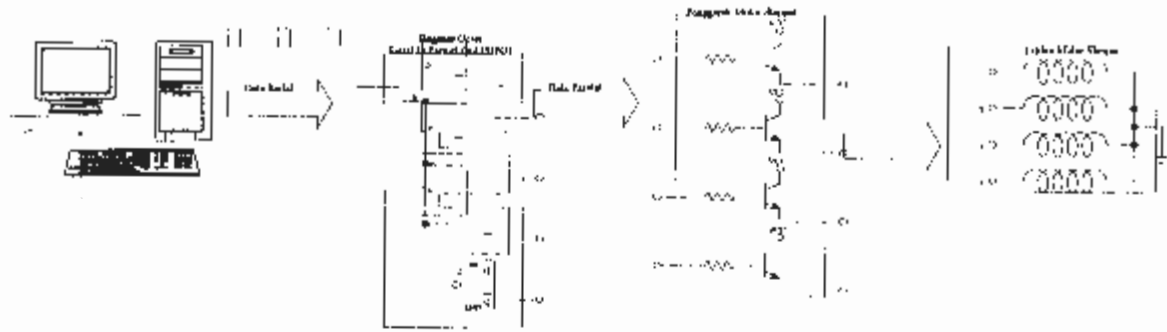


Gambar 2. Sistem Pengontrolan Motor Stepper [3]

Komputer atau PLC (*Programmable Logic Controller*) mengirimkan perintah tingkat tinggi ke *indexer*. *Indexer* (kontrol penghubung) adalah sebuah mikroprosesor yang mampu menghasilkan pulsa step dan sinyal pengarah untuk penggerak (*driver*). Sebagai tambahan, *indexer* diharapkan mampu melakukan beberapa fungsi perintah lain yang memuaskan.

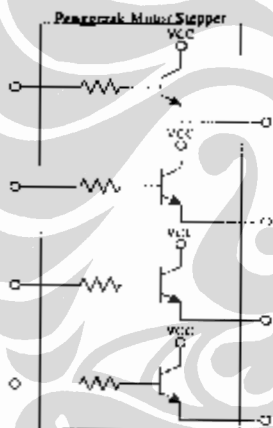
Penggerak (*Driver*) menerima pulsa step dan mengubah sinyal perintah *indexer* menjadi tenaga yang diperlukan untuk menggerakkan kumparan motor.

Motor *stepper* adalah perangkat elektromagnet yang mengubah pulsa digital menjadi putaran poros mekanis. Keuntungan motor *stepper* adalah biaya yang rendah, handal, torsi tinggi saat kecepatan rendah dan konstruksi kokoh. Kekurangan utamanya adalah efek resonansi sering terjadi pada kecepatan rendah dan penurunan torsi dengan semakin tingginya kecepatan. Rangkaian penggerak motor *stepper* pada dasarnya merupakan rangkaian *switching* arus yang mengalir pada lilitan-lilitan motor *stepper*



Gambar 3.
Blok diagram pengontrolan motor stepper

Cara memberikan data pada motor *stepper* adalah dengan mengontrol arah putaran dari motor *stepper*. Penambahan kecepatan pada motor *stepper* dapat dilakukan dengan cara meningkatkan frekuensi data masukan pada rangkaian *switching* arus [2].

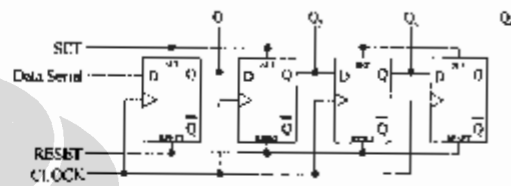


Gambar 4.
Rangkaian *switching* arus [9]

3. Register Geser SIPO (*Serial In Parallel Out*)

Cara memberikan data masukan kepada motor *stepper* dibutuhkan suatu rangkaian yang dapat mengubah masukan data seri dari komputer menjadi keluaran paralel untuk masuk ke rangkaian *switching*. Oleh karena itu, digunakan rangkaian register geser dengan masukan seri yang menghasilkan keluaran paralel. Untuk rangkaian register geser ini digunakan D *flip-flop* [5].

Tabel 1 menunjukkan bagaimana operasi dari register geser 4-bit SIPO bekerja



Gambar 5.
Register Geser 4-bit *Serial In Parallel Out*

Pada dasarnya D *flip-flop* akan menunda data masukan selama 1 *clock*. Apabila keluaran dari D *flip-flop* pertama menjadi data masukan bagi D *flip-flop* kedua maka D *flip-flop* yang kedua ini akan menunda data masukan pertama selama 2 *clock*.

Tabel 1
Operasi register geser 4-bit SIPO

Masukan				Keluaran			
S	R	D	CLK	Q ₁	Q ₂	Q ₃	Q ₄
0	1	X	1	1	1	1	1
1	0	X	2	0	0	0	0
1	1	1	3	0	0	0	0
1	1	0	4	1	0	0	0
1	1	0	5	0	1	0	0
1	1	0	6	0	0	1	0
1	1	0	7	0	0	0	1

SET akan menyetting keluaran Q menjadi 1 saat SET diberi kondisi masukan 0. Sedangkan RESET akan membuat keluaran Q menjadi 0 saat RESET diberi kondisi masukan 0.

Sinyal keluaran dari register geser 4-bit SIPO (Q₁, Q₂, Q₃, Q₄) sesuai dengan urutan pemberian data pada lilitan motor *stepper* variabel reluktansi. Satu pulsa masukan pada data masukan akan menyebabkan

motor berputar 120° . Bila kita menginginkan putaran motor dibawah 120° , maka kita dapat menggunakan sinyal RESET untuk menghentikan putaran motor. Gambar 6. akan memperjelas maksud dari tabel 1 tentang operasi dari register geser 4-bit SIPO.



Gambar 6.

Bentuk gelombang operasi dari register geser 4-bit Serial In Parallel Out (SIPO)

4. Flow Perancangan

Dalam perancangan layout CMOS VLSI ini, dibuat suatu algoritma yang membantu perancangan agar lebih sistematis. Gambar 7 merupakan diagram alir (flow) perancangan.



Gambar 7

Diagram Alir Perancangan

Dimulai dari spesifikasi sistem, yakni penentuan ide tentang apa yang akan dibuat. Dalam perancangan ini spesifikasi sistemnya adalah layout IC CMOS untuk register geser sipo (*serial in paralel out*) 4 masukan sebagai penggerak motor *stepper*.

Dari spesifikasi sistem yang telah ditentukan, kemudian kita buat fungsi logikanya. Fungsi logika ini akan sangat membantu dalam membuat rancangan rangkaian logika untuk sistem.

Dari fungsi logika yang telah kita buat maka dengan mudah kita dapat membuat rangkaian logika. Rangkaian logika ini akan membantu dalam penyusunan rangkaian transistor dimana rangkaian transistor merupakan tahapan perancangan selanjutnya.

Tahapan selanjutnya setelah membuat rangkaian transistor adalah membuat *layout gate-matrix*. *Layout gate-matrix* ini merupakan salah satu metode geometri dalam membantu penyusunan komponen-komponen layout IC CMOS.

Tahapan terakhir yaitu perancangan layout IC CMOS. Perancangan layout ini menggunakan software "Magic CAD" untuk menggambar dan "IRSIM" untuk mensimulasikan hasil rancangan layoutnya.

5. Perancangan

D-flip sering disebut dengan '*Delay flip-flop*' karena piranti ini bersifat menunda (*Latching*) dari keluarannya dan biasanya digunakan sebagai penyimpan.

Dalam penggunaannya D *flip-flop* banyak digunakan dengan menambahkan 2 masukan SET dan RESET. SET akan menyeting keluaran Q menjadi 1 saat SET diberi kondisi masukan 0. Sedangkan RESET akan membuat keluaran Q menjadi 0 saat RESET diberi kondisi masukan 0.



Gambar 8

D *flip-flop* dengan SET dan RESET [6]

Simbol :



Gambar 9

Simbol D flip-flop dengan SET dan RESET

Kondisi SET '0' dan RESET '0' merupakan kondisi yang tidak diijinkan (*prohibited condition*) dan kondisi ini harus dihindari. Dengan kondisi SET '1' dan RESET '1' maka D flip-flop dapat berfungsi menunda data masukan D satu clock ke keluaran Q. Perlu diingat bahwa D flip-flop menggunakan transisi naik (LOW-to-HIGH) dari pulsa clock untuk mentransfer data dari D ke keluaran Q. D flip-flop merupakan peralatan *logic* sekuensial dimana lebih luas lagi mampu digunakan sebagai peralatan memori sementara. Beberapa D flip-flop yang dirangkai bersama dapat membentuk suatu register geser dan register penyimpanan.

Untuk membuat sebuah layout dari rangkaian gerbang logika D-ff maka harus-

Tabel 2.

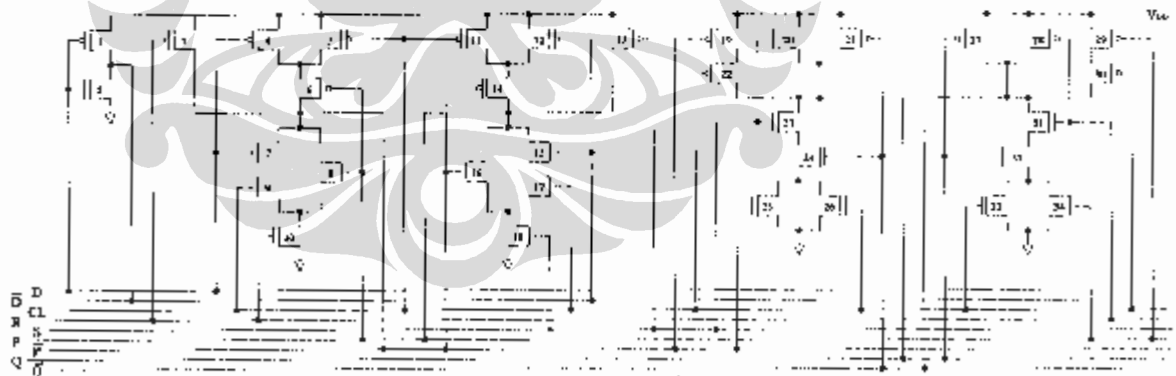
Tabel kebenaran D flip-flop dengan SET dan ET

Masukan				Keluaran	
SET	RESET	CLK	D	Q	Q
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	1	1
1	1	X	1	1	0
1	1	X	0	0	1

dibuat rangkaian transistornya terlebih dahulu. Gambar 10 merupakan rangkaian transistor untuk D-ff.

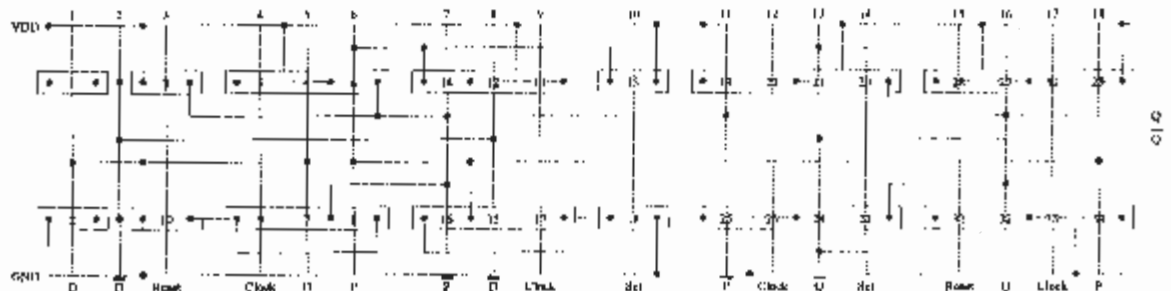
Dari Gambar 10 dapat diketahui bahwa banyaknya transistor yang digunakan untuk membentuk sebuah rangkaian D flip-flop adalah 34 buah transistor yang terdiri atas 17 transistor tipe-p dan 17 transistor tipe-n.

Dari rangkaian transistor untuk D flip-flop yang telah dibuat maka dapat disusun sebuah layout untuk sebuah D flip-flop dengan bantuan *gate matrix design*. Gambar 11 adalah gambar sebuah *gate matrix design style* untuk D flip-flop.



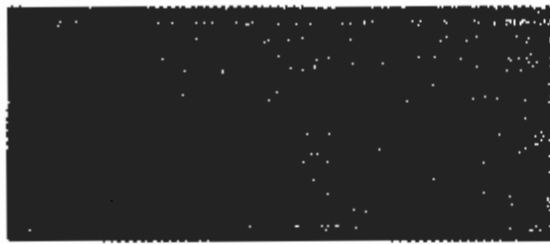
Gambar 10

Rangkaian transistor D flip-flop



Gambar 11

Gate-matrix design style D flip-flop

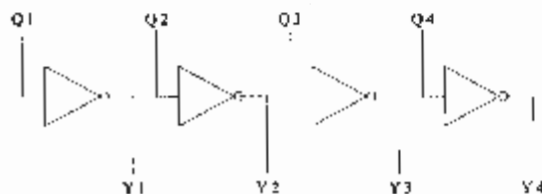


Gambar 12.
Layout D Flip-flop

Dari *gate-matrix design style* yang telah dibuat maka dapat disusun sebuah *layout* untuk sebuah *D flip-flop*. Gambar 12 adalah gambar sebuah *layout* untuk *D flip-flop*

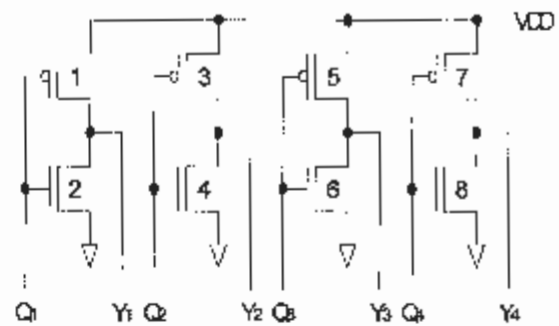
Rangkaian penggerak motor *stepper* pada dasarnya merupakan rangkaian *switching* arus yang mengalir pada lilitan-lilitan motor *stepper*. Maka untuk motor *stepper* 4 masukan dibutuhkan 4 rangkaian penggerak untuk setiap masukannya. Rangkaian penggerak motor *stepper* dapat menggunakan rangkaian *switching* arus dari transistor, seperti terlihat pada gambar 4.

Tetapi pada saat perancangan dipilih rangkaian *switching* arus yang dibangun dari inverter CMOS. Hal ini dilakukan untuk memudahkan sistem catu daya yang digunakan antara rangkaian register geser dan rangkaian *switching* arusnya. Konsekuensi dari perubahan ini adalah bahwa data masukan yang digunakan oleh rangkaian *switching* arus harus berasal dari keluaran negatif SIPO (not Q_1 , not Q_2 , not Q_3 , dan not Q_4) agar menghasilkan sinyal keluaran yang dikehendaki. Gambar 13 merupakan gerbang logika dari rangkaian *inverter* untuk rangkaian *switching* arus.



Gambar 13
Gerbang logika penggerak motor *stepper*

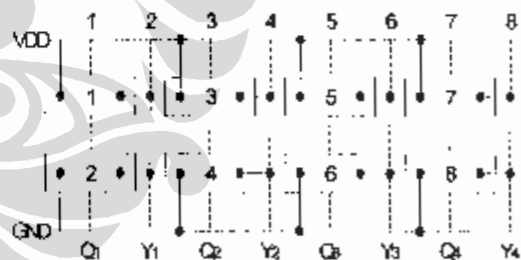
Untuk membuat sebuah *layout* dari rangkaian gerbang logika penggerak motor *stepper* maka kita harus membuat rangkaian transistornya terlebih dahulu. Gambar 14-



Gambar 14
Rangkaian transistor penggerak motor *stepper*

Dari Gambar 14 dapat diketahui bahwa banyaknya transistor yang digunakan untuk membentuk sebuah rangkaian penggerak *stepper* adalah 8 buah transistor yang terdiri atas 4 transistor tipe-p dan 4 transistor tipe-n.

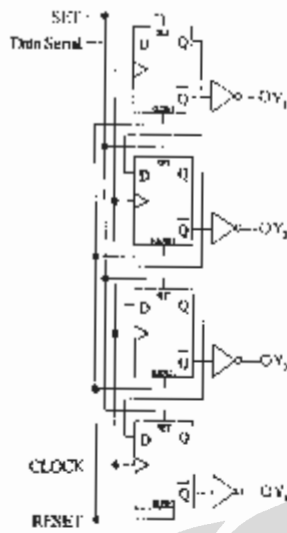
Dari rangkaian transistor untuk penggerak motor *stepper* yang telah dibuat maka dapat disusun sebuah *layout* untuk sebuah penggerak motor *stepper* dengan bantuan *gate matrix design*. Gambar 15 adalah gambar sebuah *gate matrix design style* untuk penggerak motor *stepper*.



Gambar 15
Gate-matrix design style penggerak motor *stepper*

merupakan rangkaian transistor untuk penggerak motor *stepper*.

Dalam perancangan akan buat 3 alternatif susunan dari komponen-komponen penyusun rancangan. Pada alternatif 1, register geser SIPO disusun memanjang ke bawah dengan rangkaian penggerak motor *stepper* (*inverter*) ditempatkan disebelah kanan *D flip-flop*.



Gambar 16.
Susunan Komponen Alternatif 1

Pada alternatif 2 register geser SIPO disusun memanjang ke samping dengan rangkaian penggerak motor stepper (*inverter*) ditempatkan disebelah kanan tiap-tiap D *flip-flop*.



Gambar 17.
Susunan Komponen Alternatif 2

Pada alternatif 3 register geser SIPO disusun memanjang ke samping dengan rangkaian penggerak motor stepper (*inverter*) ditempatkan disebelah kanan register geser SIPO.



Gambar 18.
Susunan Komponen Alternatif 3

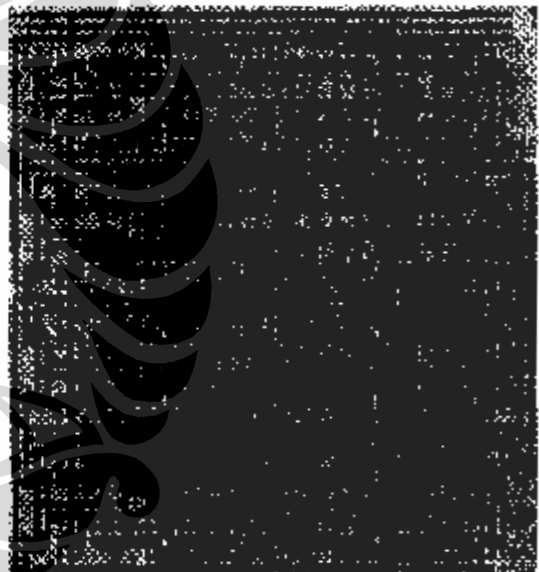
Gambar 19, 20, dan 21 merupakan gambar layout IC CMOS dari alternatif-alternatif susunan komponen yang telah dibuat. Warna biru menunjukkan bagian

metal, coklat adalah bagian p-diff, hijau adalah bagian n-diff, merah adalah bagian polisilikon dan hitam menunjukkan bagian kontak.

Layout alternatif 1 memiliki panjang rangkaian 376 λ dan lebar 380 λ sehingga luas rangkaian dari *layout* alternatif 1 adalah $376 \lambda \times 380 \lambda = 142880 \lambda^2$

Layout alternatif 2 memiliki panjang rangkaian 1424 λ dan lebar 112 λ sehingga luas rangkaian dari *layout* alternatif 2 adalah $1424 \lambda \times 112 \lambda = 159488 \lambda^2$

Layout alternatif 2 memiliki panjang rangkaian 1424 λ dan lebar 112 λ sehingga luas rangkaian dari *layout* alternatif 2 adalah $1412 \lambda \times 139 \lambda = 196268 \lambda^2$



Gambar 19.
Layout Alternatif 1



Gambar 20.
Layout Alternatif 2



Gambar 21.
Layout Alternatif 3

5. Hasil Simulasi dan Analisa

Pada simulasi dengan menggunakan program IRSIM, parameter yang diberikan untuk rancangan yang dibuat dapat dilihat pada tabel 3.

Tabel 3
Kondisi Kerja Yang Direkomendasikan

Frekuensi Clock	25 MHz	
Lebar Pulsa RESET	40 ns	
Lebar Pulsa Clock	20 ns	
Waktu Setup	Data Input	20 ns
	Kondisi inaktif RESET	20 ns

Dari hasil simulasi didapatkan bentuk gelombang keluaran untuk tiap rancangan dapat dilihat pada gambar 22, 23, dan 24. Sedangkan tabel 4, 5, dan 6 merupakan data hasil simulasi dari tanggapan waktunya.

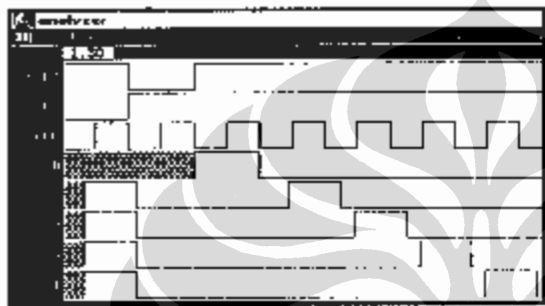


Gambar 22.
Hasil simulasi layout alternatif 1

Tabel 4
Data tanggapan waktu hasil simulasi layout alternatif 1

Parameter	Dari	Ke		Unit
t_{PHL}	RESET	Y ₁	4,51	ns
		Y ₂	4,51	ns
		Y ₃	4,51	ns
		Y ₄	4,51	ns
t_{PHL}	CLOCK	Y ₁	39,2	ns
		Y ₂	39,13	ns
		Y ₃	39,2	ns
		Y ₄	39,13	ns
Waktu Tunda D - Y ₁			19,2	ns
Waktu Tunda Y ₁ - Y ₂			9,38	ns
Waktu Tunda Y ₂ - Y ₃			9,38	ns
Waktu Tunda Y ₃ - Y ₄			9,38	ns
Lebar Pulsa Y ₁			30,16	ns
Lebar Pulsa Y ₂			30,16	ns
Lebar Pulsa Y ₃			30,16	ns
Lebar Pulsa Y ₄			30,16	ns

Apabila dilihat bentuk gelombang keluaran dari hasil simulasi, maka telah didapatkan bentuk gelombang yang sesuai dengan fungsi yang diharapkan. Hanya lebar pulsa dari tiap-tiap keluaran Y menjadi lebih kecil dari lebar pulsa data masukan. Pada alternatif 1, rata-rata lebar pulsa keluaran Y adalah 30,16 ns. Pada alternatif 2, rata-rata lebar pulsa keluaran Y adalah 32,0 ns. Sedangkan pada alternatif 3, lebar pulsa Y memiliki nilai yang bervariasi antara 32,53 ns – 33,88 ns.



Gambar 23. Hasil simulasi layout alternatif 2

Tabel 5 Data tanggapan waktu hasil simulasi layout alternatif 2

Parameter	Dari	Ke	Unit
t_{PHL}	RESET	Y_1	4,66 ns
		Y_2	4,66 ns
		Y_3	4,66 ns
		Y_4	4,66 ns
t_{PLH}	CLOCK	Y_1	38,23 ns
		Y_2	38,23 ns
		Y_3	38,23 ns
		Y_4	38,23 ns
Waktu Tunda D – Y_1		18,23	ns
Waktu Tunda Y_1 – Y_2		7,41	ns
Waktu Tunda Y_2 – Y_3		8,08	ns
Waktu Tunda Y_3 – Y_4		8,74	ns
Lebar Pulsa Y_1		32,0	ns
Lebar Pulsa Y_2		32,0	ns
Lebar Pulsa Y_3		32,0	ns
Lebar Pulsa Y_4		32,0	ns

Pada pulsa keluaran Y terdapat tunda waktu antar keluaran Y. Pada alternatif 1, rata-rata tunda waktu antar keluaran Y adalah 9,38 ns. Pada alternatif 2, rata-rata tunda waktu antar keluaran Y adalah 8,07 ns. Sedangkan pada alternatif 3, tunda waktu antar keluaran Y adalah 5,49 ns.



Gambar 24. Hasil simulasi layout alternatif 3

Sedangkan pada waktu tunda antara data masukan dengan keluaran Y_1 pada alternatif 1 adalah 19,2 ns. Pada alternatif 2, waktu tunda antara data masukan dengan keluaran Y_1 adalah 18,23 ns. Pada alternatif 3, waktu tunda antara data masukan dengan keluaran Y_1 adalah 23,78 ns.

Tabel 6 Data tanggapan waktu hasil simulasi layout alternatif 3

Parameter	Dari	Ke	Unit
t_{PHL}	RESET	Y_1	8,57 ns
		Y_2	7,85 ns
		Y_3	7,13 ns
		Y_4	6,47 ns
t_{PLH}	CLOCK	Y_1	43,78 ns
		Y_2	42,76 ns
		Y_3	41,74 ns
		Y_4	40,81 ns
Waktu Tunda D – Y_1		23,78	ns
Waktu Tunda Y_1 – Y_2		5,1	ns
Waktu Tunda Y_2 – Y_3		5,45	ns
Waktu Tunda Y_3 – Y_4		5,92	ns
Lebar Pulsa Y_1		33,88	ns
Lebar Pulsa Y_2		33,53	ns
Lebar Pulsa Y_3		33,15	ns
Lebar Pulsa Y_4		32,53	ns

6. Kesimpulan

Dari hasil penggambaran layout, dapat disimpulkan beberapa hal sebagai berikut:

Perbandingan Hasil Penggambaran untuk rancangan alternatif 1 luas Layout adalah $142880 \lambda^2$, Alternatif 2 Luas Layout $159488 \lambda^2$ dan Alternatif 3 Luas Layout $196268 \lambda^2$.

- ❖ Rangkaian gabungan dibangun dari paduan register geser SIPO 4 masukan dan rangkaian penggerak.
- ❖ Rangkaian register geser SIPO dibangun dari 4 buah *D flip-flop* dan rangkaian penggerak dibangun dari 4 buah inverter
- ❖ Dari hasil penggambaran *layout*, rancangan *layout* gabungan *D-flip-flop – Driver Stepper* alternatif 1 memiliki efisiensi ruang yang paling baik dibanding rancangan *layout* yang lainnya dengan luas *layout* $142880 \lambda^2$

Sedangkan dari hasil analisa tanggapan waktu, dapat kita simpulkan beberapa hal sebagai berikut:

Kondisi Kerja Yang Direkomendasikan memiliki Frekuensi Clock 25 MHz, Lebar Pulsa RESET 40 ns dan Lebar Pulsa Clock 20 ns serta Waktu Setup Data Input adalah 20 ns sedangkan Kondisi inaktif RESET adalah 20 ns.

Kondisi kerja yang direkomendasikan memiliki nilai yang sama untuk semua rancangan. Hal ini dikarenakan untuk semua rancangan dilakukan simulasi dengan data yang sama.

Perbandingan Tanggapan Waktu Karakteristik Pensaklaran Rancangan

Alternatif 1 t_{PHL} rata-rata dari RESET ke setiap Y adalah 4,51 ns, t_{PHL} rata-rata dari CLOCK ke setiap Y = 39,16 ns Waktu Tunda rata-rata antar Y adalah 9,38 ns

Alternatif 2 t_{PHL} rata-rata dari RESET ke setiap Y adalah 4,66 ns, t_{PHL} rata-rata dari CLOCK ke setiap Y = 38,23 ns dan Waktu Tunda rata-rata antar Y adalah 8,07 ns

Alternatif 3 t_{PHL} rata-rata dari RESET ke setiap Y adalah 7,50 ns, t_{PHL} rata-rata dari CLOCK ke setiap Y = 42,27 ns, Waktu Tunda rata-rata antar Y adalah 5,49 ns

- ❖ Tanggapan waktu t_{PHL} dari RESET untuk bentuk *layout* alternatif 1 memiliki tanggapan waktu yang paling cepat.
- ❖ Tanggapan waktu t_{PHL} dari CLOCK untuk bentuk *layout* alternatif 2

memiliki tanggapan waktu yang paling cepat.

- ❖ Waktu tunda rata-rata antar keluaran Y dari hasil rancangan sekitar 5,49 ns – 9,38 ns. Dengan kondisi waktu tunda lebih kecil dari 1 ms maka motor *stepper* masih dapat dinyatakan memiliki putaran yang halus (*smooth*).

Bila diprioritaskan luas *layout* yang minimal dengan tanggapan waktu yang masih dapat diterima, maka *layout* alternatif 1 memiliki luas yang paling kecil dengan tanggapan waktu yang masih dapat diterima.

Daftar Acuan

1. <http://www.alds.stts.edu/ANALOG/Stepper1.htm>
2. <http://www.alds.stts.edu/ANALOG/Stepper2.htm>
3. webmaster@anaheimautomation.com
4. <http://Uj/ams2000/stepping101.html>
5. Roger L. Tokheim, *Digital Electronics, McGraw-Hill*, 1994
6. Neil H. E. Weste and Kamran Eshraghian, *Principles of CMOS VLSI Design, Addison-Wesley Publishing Company*, 1985
7. Aris Budiarto, Diktat Elektronika Digital II, POLMAN – Bandung, 1995
8. Rizal Rizkiawan, Tutorial Perancangan Hardware, PT. Elex Media Computindo Jakarta, 1996
9. Wolfgang Link, Pengukuran Pengendalian dan Pengaturan dengan PC, PT. Elex Media Computindo Jakarta, 1995
10. Albert Paul Malvino, *Computer Digital Electronics*, Gunadarma Jakarta, 1991.
11. Kwiro Lee, Michael Shur, Tor A. Fjeldly, Trond Ytterdal, *Semiconductor Device Modelling for VLSI, Prentice-hall International Edition*, 1993.
12. Ronald J. Tocci, *Digital System Principles and Application, Prentice-hall Fifth Edition*, 1993.