

Penerapan *Finite State Machine* Untuk Merancang Pengendali Motor Stepper Menggunakan Vhdl

Arief Budijanto
 Universitas Widya Kartika
 Email : arief232@yahoo.com

ABSTRAK

Makalah ini menjelaskan tentang proses pembelajaran mata kuliah perancangan chip digital berbasis proyek yang menerapkan Finite State machine (FSM) sebagai metode untuk merancang pengendali motor stepper menggunakan VHDL. Motor stepper yang digunakan dalam studi kasus ini adalah motor stepper unipolar. Dimana motor stepper tersebut dikendalikan dengan 2 mode, yaitu half-step dan full-step. Pengendalian pada mode full step terdiri dari 2 cara yaitu dengan kendali 1 phase ON dan 2 phase ON. Pengendali motor stepper diimplementasikan pada chip Complex Programmable Logic Device (CPLD) dengan seri EPM3032ALC44-4. Dari hasil simulasi waktu yang dibutuhkan dari input sampai ke output membutuhkan waktu 3 ns.

Kata kunci : Pengendali, Motor Stepper, FSM, VHDL

PENDAHULUAN

Perancangan *Chip Digital* merupakan teknologi yang perkembangannya sangat pesat. Sehingga dalam dunia pendidikan dibidang elektronika dan komputer perlu dikembangkan metode - metode perancangan *chip digital* agar lulusan mahasiswa dapat mendukung perkembangan teknologi perancangan chip. Dalam makalah ini akan dijelaskan tentang perancangan chip digital dengan studi kasus perancangan pengendali motor stepper jenis unipolar yang menerapkan *finite state machine* sebagai metodenya.

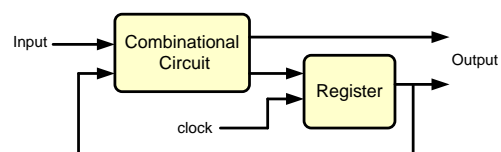
Finite State Machine ini merupakan cara untuk memudahkan perancang dalam membuat *VHDL coding* dengan cara menkonversikan *state diagram* dalam *VHDL coding*. Kendali motor stepper yang dirancang menggunakan *mode half step* dan *full step* yang disimulasik menggunakan perangkat lunak *Max Plus* yang merupakan produk dari *Altera, Inc*.

Finite State Machine

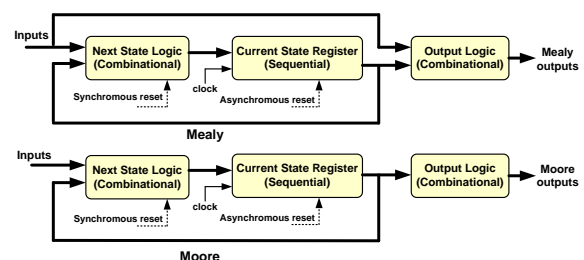
Sistem digital dapat diklasifikasikan sebagai rangkaian kombinasional dan Sekuensial (berurutan). Pada sistem digital kombinasional ini nilai output ditentukan oleh nilai input pada kondisi saat ini. Sedangkan sistem digital sekuensial output tidak hanya tergantung pada kondisi input saat ini, tetapi juga tergantung pada kondisi input sebelumnya, sehingga pada sistem ini dibutuhkan suatu *memory (register)* untuk

menyimpan informasi biner. Diagram blok sistem digital sekuensial diperlihatkan pada gambar 1.

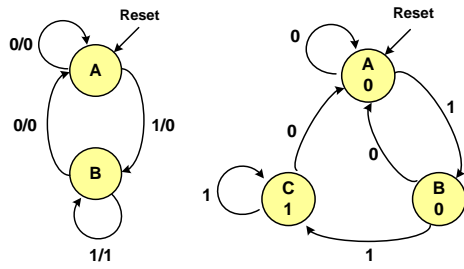
Finite state machine (FSM) dalam perancangan sistem digital merupakan metode yang digunakan untuk merancang rangkaian digital sekuensial. FSM dapat diklasifikasikan sebagai *Mealy Machine* dan *Moore Machine* (Short, 2009). Diagram blok *FSM* ditunjukkan pada gambar 2. Pada *mealy machine*, output merupakan fungsi dari *current state* dan nilai dari inputnya. Sedangkan *moore machine*, output merupakan fungsi dari *current state* saja.



Gambar 1. Diagram Blok Rangkaian sekuensial (Mano & Ciletti, 2013)



Gambar 2. Diagram blok *FSM Mealy & Moore* (smit,1997)



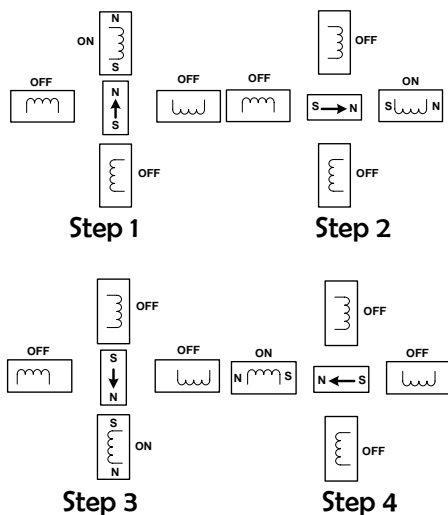
a. Mealy Machine b. Moore Machine
Gambar 3. State Diagram (Short, 2009).

Motor Stepper

Motor *stepper* adalah salah satu jenis motor dc yang dikendalikan dengan pulsa-pulsa digital. Prinsip kerja motor *stepper* adalah bekerja dengan mengubah pulsa elektronik menjadi gerakan mekanis diskrit dimana motor *stepper* bergerak berdasarkan urutan pulsa yang diberikan kepada tiap-tiap koil motor *stepper* tersebut. Pulsa digital yang diberikan pada koil motor *stepper* berupa level logika '1' dan '0' yang ditunjukkan pada tabel 1. dan tabel 2.

Tabel 1. Pulsa digital untuk *mode full step* dengan 1 Phase ON

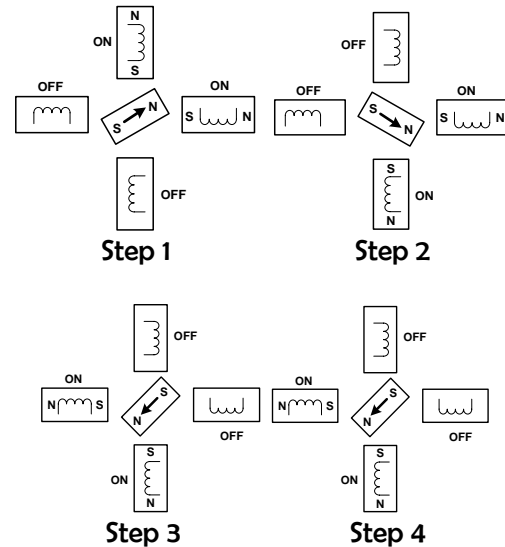
T	Y0	Y1	Y2	Y3
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1



Gambar 4. Putaran Motor Stepper Mode Full Step 1 Phase ON

Tabel 2. Pulsa digital untuk *mode full step* dengan 2 Phase ON

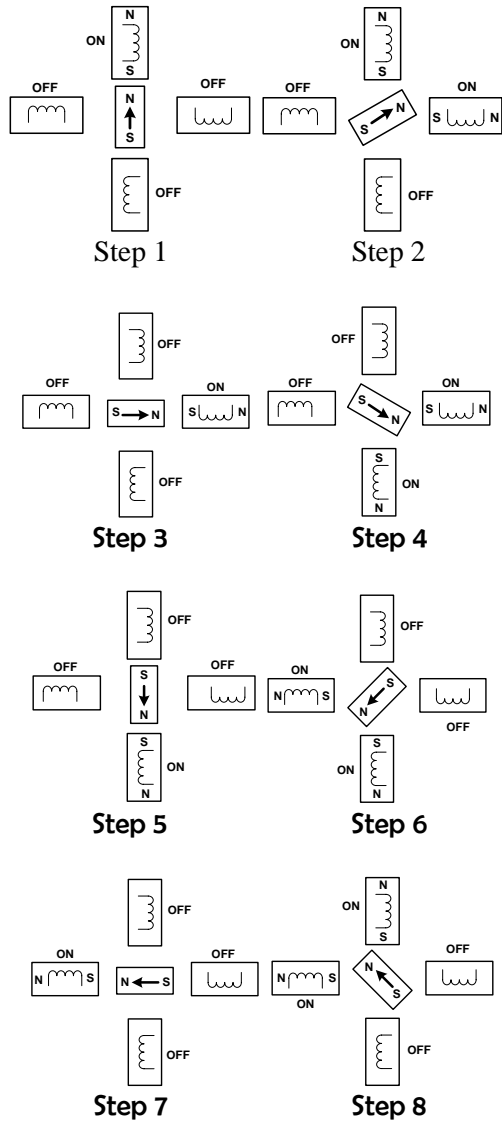
T	Y0	Y1	Y2	Y3
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1



Gambar 5 Putaran Motor Stepper Mode Full Step 2 Phase ON

Tabel 3. Pulsa digital untuk *mode full step*

T	Y0	Y1	Y2	Y3
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1



Gambar 6 Putaran Motor *Stepper Mode Half Step*

METODOLOGI

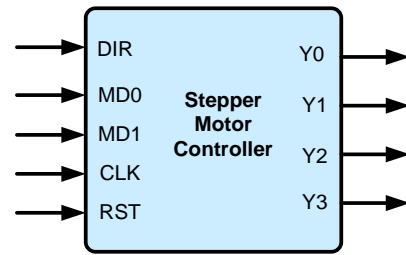
Metodologi perancangan pengendali motor *stepper* dengan *VHDL* meliputi :

1. Desain diagram blok *chip*
2. Tabel fungsi *chip*
3. Desain *FSM* dari tabel fungsi *chip*
4. Implementasi *FSM* dalam *VHDL coding*

Konfigurasi *pin output chip* pengendali motor *stepper* ditunjukkan pada gambar 7. Dimana fungsi pin adalah sebagai berikut:

- DIR** : untuk mengatur arah putaran
- MD0,MD1** : untuk mengatur mode putaran yaitu *half step* atau *full step*
- CLK** : masukan pulsa *clock*

RST : Masukan *pin reset*



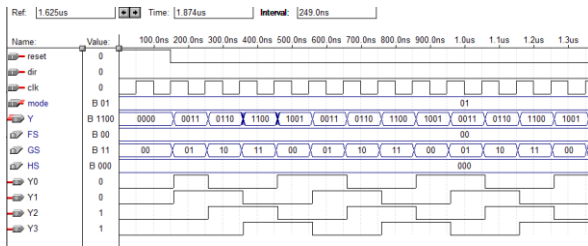
Gambar . 7 Diagram blok pengendali motor *stepper*

Sedangkan fungsi pin nya ditunjukkan pada tabel 4.

Tabel 4. Fungsi *chip*

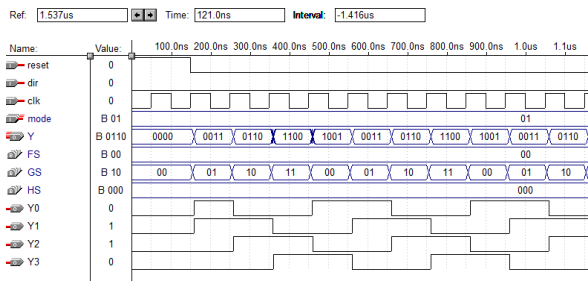
INPUT				OUTPUT
DIR	MD 1	MD 0	RST	MODE
0	0	0	0	Full Step Clockwise 1 Phase On
0	0	1	0	Full Step Clockwise 2 Phase ON
0	1	0	0	Half Step Clockwise
x	1	1	0	Unused
1	0	0	0	Full Step Anti Clockwise 1 Pphase ON
1	0	1	0	Full Step Anti Clockwise 2 Phase ON
1	1	0	0	Half Step Anti Clockwise
x	x	x	1	Reset Condition

Setelah membuat diagram blok dan tabel fungsi *chip*, kemudian dilanjutkan dengan membuat diagram *FSM* pengendali motor *stepper*. Diagram tersebut ditunjukkan pada gambar 8.



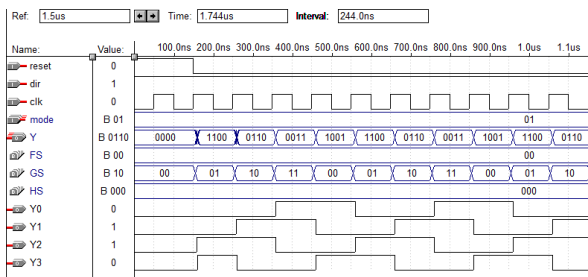
Gambar 12. Deklarasi Entity VHDL

Gambar 13. Merupakan hasil simulasi kendali motor stepper dengan cara *2 phase ON*, yaitu dengan memberikan logika pada pin *mode = "01"* dan *dir = '1'*, sehingga arah putaran motor *stepper* berlawanan dengan arah putaran jarum jam.



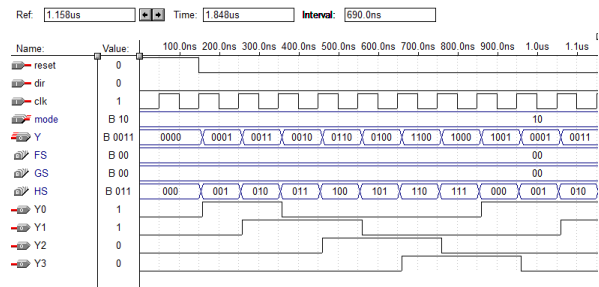
Gambar 13. Deklarasi Entity VHDL

Gambar 14. Merupakan hasil simulasi kendali motor *stepper* dengan *mode = "10"* dan *dir = '0'*, sehingga arah putaran motor *stepper* berlawanan dengan arah putaran jarum jam.



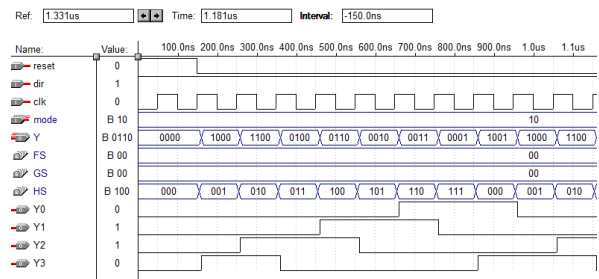
Gambar 14. Deklarasi Entity VHDL

Gambar 15. Merupakan hasil simulasi kendali motor *stepper* dengan metode *half step* dan memberikan logika pada pin *mode = "10"* dan *dir = '0'* sehingga arah putaran motor *stepper* searah dengan arah putaran jarum jam. Tetapi derajat simpangannya lebih kecil setengahnya dari simpangan *full step*.



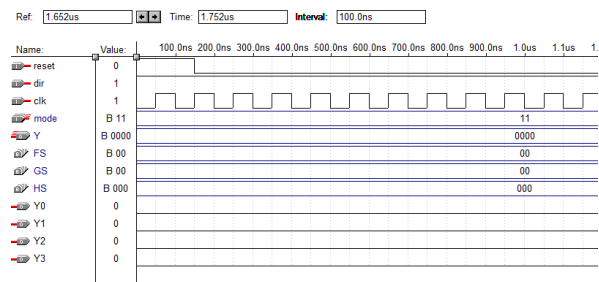
Gambar 15. Deklarasi Entity VHDL

Gambar 16. merupakan hasil simulasi kendali motor stepper dengan metode *half step* dan memberikan logika pada pin *mode = "10"* dan *dir = '1'* sehingga arah putaran motor *stepper* berlawanan dengan arah putaran jarum jam. Tetapi derajat simpangannya lebih kecil setengahnya dari simpangan *full step*.



Gambar 16. Deklarasi Entity VHDL

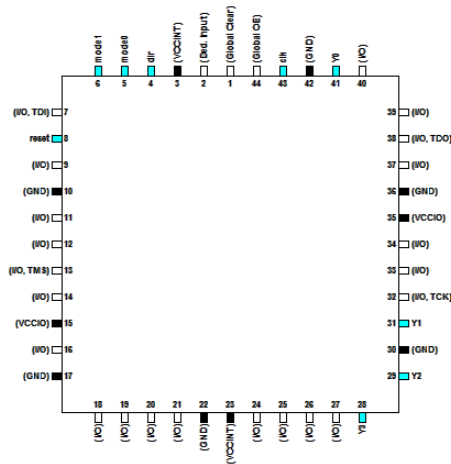
Gambar 17. Merupakan hasil simulasi kendali motor *stepper* dengan *mode = "11"*. Pada kondisi ini output *Y* berlogika "0000" karena kondisi ini memang dirancang tidak digunakan untuk mengendalikan motor *stepper*.



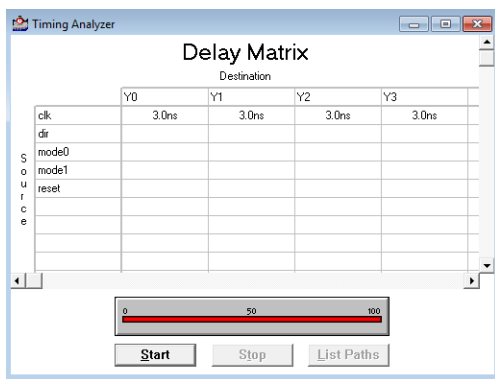
Gambar 17. Deklarasi Entity VHDL

Konstruksi pin chip pengendali motor stepper menggunakan *CPLD* seri *EPM3032ALC44-4* ditunjukkan pada gambar 18. Dan matriks waktu tunda perambatan sinyal digital (*time propagation delay*) dari input menuju output

membutuhkn waktu 3 ns. Matriks waktu tunda ditunjukkan pada gambar 19.



Gambar 18. Konstruksi pin chip Pengendali Motor Stepper



Gambar 19. Matriks Waktu Tunda Perambatan Sinyal Digital

KESIMPULAN

Kesimpulan yang dapat diambil dari simulasi pengendali motor stepper adalah sebagai berikut:

1. Chip pengendali dapat mengendalikan motor stepper dengan metode *full step* dan *half step*.
2. Pada metode full step ada 2 cara pengendalian berdasarkan fasenya, yaitu *1 phase ON* dan *2 phase ON*.
3. Chip diimplementasikan pada komponen CPLD seri EPM3032ALC44-4.
4. Chip membutuhkan waktu tunda perambatan dari input menuju output chip sekitar 3 ns.

DAFTAR PUSTAKA

Rushton, Andrew, (2011), *Vhdl For Logic Synthesis*, John Wiley & Sons, Ltd, United Kingdom.

Platt, Charles, (2013), *Encyclopedia of Electronic Components Volume 1*, O’Reilly Media, Inc. 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Iuliana C, Alin D. P, Adrian G, (2010), *Finite State Machine Design And Vhdl Coding Techniques*, 10th International Conference On Development And Application Systems, Suceava, Romania, May 27-29, pp.273-278.

Kavita S, Abhilasha S, *FPGA Based System Login Security Lock Design Using Finite State Machine*, IOSR Journal of Electronics and Communication Engineering, Volume 5, Issue 3 (Mar. - Apr. 2013), PP 70-75.

Mano, M. Morris, Ciletti, Michael D, (2013), *Digital Design With an Introduction to the Verilog HDL*, Prentice Hall.

Short, Keneth L., (2009), *VHDL for engineers*, Pearson Education, Inc., Upper Saddle River, New Jersey.

Smith, Douglas J., (1997), *HDL Chip Design*, Doone Publications, Madison, AL,USA.