Makalah Nomor: KNSI-343

# SIMULASI RANGKAIAN DIGITAL MESIN PENJUAL KOPI DENGAN XILINX

Ayu Astariatun<sup>1</sup>, Nelly Sulistyorini<sup>2</sup>, Hartono Siswono<sup>3</sup>

1,2,3 Teknik Elektro, Teknologi Industri, Universitas Gunadarma Jl. Margonda Raya 100, Depok, West Java, Indonesia 1 ayu.astariatun@gmail.com, 2 nelnelly16@yahoo.com, 3 hartono@staff.gunadarma.ac.id

#### **Abstrak**

"Rangkaian Digital Mesin Penjual Kopi" merupakan rangkaian digital yang dirancang untuk membuat vending machine yang dapat mengeluarkan 3 pilihan kopi, yaitu kopi dengan rasa "original", "vanilla" dan "cappucino" dengan memasukkan uang Rp.5000,-. Rangkaian ini dirancang menggunakan gerbang logika dan D flip-flop dengan melalui beberapa tahap diantaranya: identifikasi masalah, state diagram, tabel kebenaran, K-map, IFL (Input Forming Logic), OFL (Output Forming Logic), simulasi dan analisa.

Kata kunci : Identifikasi Masalah, Simulasi, State Diagram

## 1. Pendahuluan

Seiring berkembangnya zaman, teknologi sangat berpengaruh bagi kehidupan manusia. Dengan adanya teknologi, kehidupan manusia menjadi mudah dan praktis. Salah satu teknologi yang berpengaruh di zaman modern ini adalah vending machine, yaitu mesin penjual makanan atau minuman ringan secara praktis. Dengan adanya mesin ini dapat menghilangkan fungsi kasir, karena pembeli cukup memasukkan uang ke dalam mesin dan memilih makanan atau minuman yang ingin dibeli, kemudian mesin otomatis akan mengeluarkan makanan atau minuman yang diinginkan.

Teknologi yang sedang populer ini menginspirasi penulis untuk membuat "Simulasi Rangkaian Digital Mesin Penjual Kopi", yang bertujuan untuk memudahkan semua orang dalam menikmati kopi hangat dengan praktis. Terutama dalam lingkungan perkantoran dan perkuliahan yang membutuhkan hal-hal serba praktis. Piranti ini memiliki 3 pilihan yang juga sebagai keluaran yaitu : Kopi rasa original, kopi rasa vanilla dan kopi rasa cappucino, dengan harga sebesar Rp.5000,-.

## 2. Teori

# 2.1 Gerbang Logika IC Digital

Gerbang Logika adalah rangkaian dengan satu atau lebih dari satu sinyal masukan tetapi hanya menghasilkan satu sinyal berupa tegangan tinggi atau tegangan rendah.

#### 2.1.1 Gerbang AND

Gerbang AND adalah gerbang logika di mana hasil dari jalan keluarnya (*output*) ditentukan

oleh hasil perkalian pada jalan masukan gerbang tersebut. Jika A dan B adalah input dan Y adalah output gerbang AND, maka didapat persamaan berikut:  $Y = A \cdot B$ .

Dan kemungkinan hasil tabel kebenaran yang didapat adalah sebagai berikut :

Tabel 2.1 Tabel Kebenaran Gerbang AND





Gambar 2.1 Gambar Gerbang AND

#### 2.1.2 Gerbang NOT

Gerbang NOT adalah gerbang logika yang berfungsi untuk mengubah nilai pada jalan masukan menjadi nilai komplemennya, oleh karena itu gerbang logika ini dapat disebut juga sebagai pembalik atau *inverter*. Jadi nilai pada jalan keluaran ini berlawanan dengan nilai masukannya, berikut ini adalah tabel kebenaran dan gambar simbol gerbang NOT:

Tabel 2.2 Tabel Kebenaran Gerbang NOT

X	Y
0	1
1	0

Gambar 2.2 Gerbang NOT

## 2.1.3 Gerbang OR

Gerbang OR adalah gerbang logika dimana hasil dari keluarannya (output) ditentukan oleh hasil penambahan pada masukan (input) gerbang tersebut. Jika A dan B sebagai masukan dan Y sebagai hasil dari keluaran maka persamaan yang didapat adalah, Y = A + B.

Tabel 2.3 Tabel Kebenaran Gerbang OR

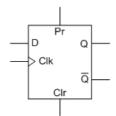
A	В	Y
0	0	0
0	1	1
1	0	1
1	1	1



Gambar 2.3 Gerbang OR

## 2.2 Flip-Flop D

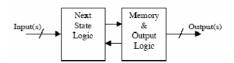
Flip-flop D (*delay* = tunda) merupakan flip-flop yang dapat mencegah nilai D mencapai nilai keluaran sampai berlangsungnya pulsa clock. Flip-flop D sering kali disebut juga sebagai flip-flop tunda, nama ini menggambarkan operasi unit ini. Apapun bentuk input pada input data (D), input tersebut akan tertunda selama satu pulsa detak untuk mencapai output normal (Q). Data dipindahkan ke output pada transisi detak Low ke High.



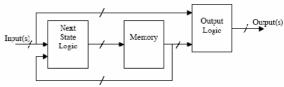
Gambar 2.4 Lambang Logika D Flip-Flop

#### 2.3 State Machine

State machine merupakan konstruksi logika yang digunakan untuk mendefinisikan sifat sistem yang sederhana. *State machine* menggunakan kombinasi *latch* dan gerbang logika untuk sistem yang mempunyai memori. *State machine* di gambarkan dengan *state diagram* yang didefinisikan dari *state transition table*. Contoh *state machine* yang sederhana dapat dilihat pada gambar 2.3



Gambar 2.5 Moore Machine



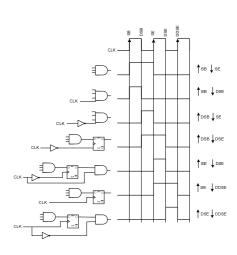
Gambar 2.6 Mealy Machine

## 2.4 Output Forming Logic

Output forming logic hampir sama dengan translasi pada counter akan tetapi Output forming logic mempunyai fungsi lebih untuk perancangan yang glitch-free dan digunakan untuk masukan yang lebih kompleks ke suatu dekoder yang merupakan basis rangkaian digital berdasarkan output.

Sebagai contoh penggunaan 3 bit sebagai kombinasi output dimana 3 bit membangkitkan (generate) single output atau multiple output. Counter mod 6 glitch free dengan keluaran sederetan angka 0, 1, 2, 3, 4 dan 5 dapat dibuat dengan menentukan output forming logic dengan menggunakan peta karnaugh (karnaugh map).

Beberapa rangkaian untuk membangkitkan pulsa untuk *output forming logic* dapat dilihat pada gambar 2.7.



Gambar 2.7 Rangkaian Untuk Membangkitkan Pulsa Untuk *Output Forming Logic* 

Definisi-definis transisi state:

SB : State BeginningSE : State End

DSB : Delayed State Beginning
 DSE : Delayed State End
 DDSE : Double Delayed State End

## 3. Perancangan Sistem

Proses perancangan yang harus dilakukan yaitu, pertama membuat pengidentifikasian masalah, dimana bagian tersebut menentukan input dan output dari rangkaian. Kemudian dilakukan pengidentifikasian keadaan (state) dan dilanjutkan dengan penggambaran diagram keadaan (state diagram). Dari state diagram dapat ditentukan table kebenaran. Dengan bantuan K-map dapatlah dibentuk persamaan aljabar Boolean tersederhana untuk sebagai masukan pada D Flip-flop atau disebut sebagai Input Forming Logic (IFL), setelah menentukan gelombang inputnya dapat ditentukan Output Forming Logic (OFL).

#### 3.1 Identifikasi Masalah

Permasalahan pada mesin penjual kopi dapat dinyatakan sebagai berikut:

• input

✓ Tidak ada kegiatan : TA
✓ Memasukkan uang Rp 5.000,- : D
✓ Uang Kembali : E
✓ Memilih kopi original : F
✓ Memilih kopi cappucino : G
✓ Memilih kopi vanila : H

• output

✓ uang kembali : W
✓ kopi original : X
✓ kopi cappucino : Y
✓ kopi vanilla : Z

## 3.2 Identifikasi State / Keadaan :

#### 1. Keadaan a:

- Tidak ada kegiatan, kembali ke keadaan a.
- Memasukkan uang Rp.5000,-.

## 2. Keadaan b:

- Tidak ada kegiatan, kembali ke keadaan b

 Minta uang kembali ke keadaan c.

- Memilih kopi original ke keadaan d.
- Memilih kopi cappucinno ke keadaan e.
- Memilih kopi vanilla ke keadaan f.

#### 3. Keadaan c:

- Mengeluarkan uang kembali, kembali ke keadaan a

#### 4. Keadaan d:

- Mengeluarkan kopi original, kembali ke keadaan a.

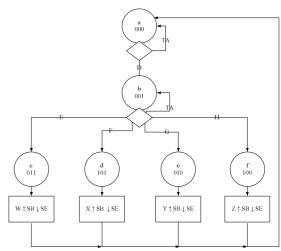
#### 5. Keadaan e:

 Mengeluarkan kopi cappucinno, kembali ke keadaan a

#### 6. Keadaan f:

Mengeluarkan kopi vanilla, kembali ke keadaan a.

## 3.3 Diagram Keadaan (State Diagram)



Gambar 3.1 Diagram Keadaan

## 3.4 Table Kebenaran Rangkaian

Dan tabel kebenarannya adalah sebagai berikut : Tabel 3.1 Tabel Kebenaran Keseluruhan Dari Rangkaian

pre	sent	t	inpu	ıt				next			DA DB	DC		
A	В	С	D	E	F	G	Н	1 IICAE		В		DC	OFL	
0	0 0 0	0	0	x	x	х	х	0	0	0	0	0		
		ľ	1	x	x	х	х	0	0	1			D	x
			х	0	0	0	0	0	0	1			E' F' G' H' + E + F	x
			х	1	x	х	x	0	1	1	F+H	E+G		
0	0	1	х	х	1	х	x	1	0	1				
			х	x	x	1	x	0	1	0				
			х	x	x	х	1	1	0	0				
0	1	0	х	х	х	х	х	0	0	0	0	0	0	Y
0	1	1	х	x	x	х	х	0	0	0	0	0	0	W
1	0	0	x	x	x	x	x	0	0	0	0	0	0	Z
1	0	1	х	x	x	х	x	0	0	0	0	0	0	X
1	1	0	х	x	x	х	x	X	х	х	x	х	x	x
1	1	1	x	x	x	x	x	X	x	x	x	x	x	x

## 3.5 Pembentukan Input Forming Logic (IFL)

 $D_A$ :

C \ AB	00	01	11	10
0	0	0	X	0
1	F+H	0	x	0

#### $D_B$ :

C \ AB	00	01	11	10
0	0	0	х	0
1	E+G	0	X	0

# $D_C$ :

C \ AB	00	01	11	10
0	D	0	x	0
1	E' F' G' H' + E + F	0	x	0

Dengan menggunakan K-map maka diperoleh *Input Forming Logic* sebagai berikut:

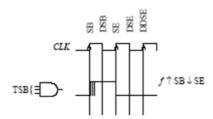
 $D_A = A'B'C (F+H)$ 

 $D_B = A'B'C (E+G)$ 

 $D_C = (A'B'C(E'F'G'H' + E + F) + (A'B'C'D)$ 

## 3.6 Penentuan Output Forming Logic (OFL)

Output forming logic kali ini yang dipilih adalah gelombang " $\uparrow$  SB  $\downarrow$  SE" karena rangkaian untuk menghasilkan output forming logic ini sederhana, yaitu hanya menggunakan gerbang AND dan OR.

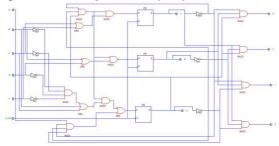


Gambar 3.2 Rangkaian Pembangkit Pulsa OFL  $\uparrow$  SB  $\downarrow$  SE

## 4. Implementasi Sistem

## 4.1 Schematic Editor

Membuat rangkaian sesuai persamaan yang diperoleh dari *Output Forming Logic* :



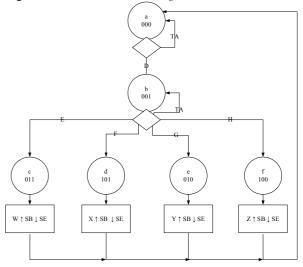
Gambar 4.1 Rangkaian Schematic Editor

Dalam rangakaian ini menggunakan D flip-flop dengan inputan D E F G H dan ouput A B C W X Y Z.

- Pada D flip-flop A, masukannya berupa A'B'C (F+H)
- Pada D flip-flop B, masukannya berupa A'B'C (E+G)
- Pada D flip-flop C, masukannya berupa ( A'B'C (E' F' G' H' + E + F) + (A'B'C'D)
- Pada output W bernilai 011 yaitu A'BC
- Pada output X bernilai 101 yaitu AB'C
- Pada output Y bernilai 010 yaitu A'BC'
- Pada output Z bernilai 100 yaitu AB'C'

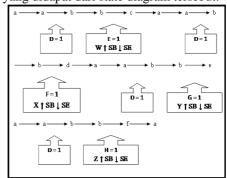
#### 4.2 Simulasi Rangkaian

Sebelum mensimulasikan rangkaian schematic editor kita harus menentukan jalur dari diagram keadaan atau state diagram.



Gambar 4.2 Diagram Keadaan

Dari state diagram tersebut kita buat jalur agar dapat menentukan nilai 1 dan 0 pada input-input ketika mensimulasikan rangkaian *schematic editor*, berikut jalur yang didapat dari state diagram tersebut:



Gambar 4.3 Jalur Rangkaian

## Keterangan:

#### ⇒ Jalur pertama,

Dimulai dari keadaan a, pertama saat di keadaan a terjadi looping yaitu kembali lagi ke keadaan a karena tidak ada kegiatan, selanjutnya dari keadaan a menuju keadaan b dimana menjadikan masukan D bernilai 1 dan masukan-masukan yang lain (E F G

H) bernilai 0, kemudian kembali lagi ke state b (looping) karena tidak ada kegiatan, berawal dari keadaan b lagi menuju ke keadaan c dimana menjadikan masukan E bernilai 1 dan masukan-masukan yang lain bernilai 0 yang mana menghasilkan keluaran W kemudian kembali keadaan a.

#### ⇒ Jalur kedua.

Kembali dimulai dari keadaan a, dan menjalankan jalur sampai keadaan b sama seperti jalur pertama, kemudian dari keadaan b menuju keadaan d dan menjadikan masukan F bernilai 1 dan masukan lain bernilai 0 yang mana menghasilkan keluaran X kemudian kembali ke keadaan a.

## ⇒ Jalur ketiga,

Berawal dari keadaan a lagi dan kembali menjalankan jalur sampai keadaan b sama seperti jalur pertama, kemudian dari keadaan b menuju keadaan e dan menjadikan masukan G bernilai 1 dan masukan lain bernilai 0 yang mana menghasilkan keluaran Y kemudian kembali ke keadaan a.

## ⇒ Jalur keempat,

Berawal dari keadaan a lagi dan kembali menjalankan jalur sampai keadaan b sama seperti jalur pertama, kemudian dari keadaan b menuju keadaan e dan menjadikan masukan H bernilai 1 dan masukan lain bernilai 0 yang mana menghasilkan keluaran Z kemudian kembali ke keadaan a.

Setelah mendapatkan jalur, memberikan nilai logika 0 atau 1 pada input-input, seperti pada gambar sebagai berikut:



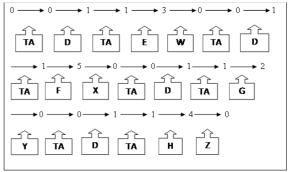
Gambar 4.4 Ketentuan Nilai Logika Input Pada Simulasi

Setelah memberi nilai 1 atau 0 pada input, simulasi dijalankan dan hasilnya seperti gambar sebagai berikut:



Gambar 4.5 Simulasi Rangkaian

Dari hasil simulasi dapat dilihat bahwa rangkaian telah dirancang dengan benar karena outputnya berupa sekuens :



Gambar 4.6 Hasil simulasi

#### 5. Kesimpulan

Proses perancangan dan hasil analisa dari rangkaian mesin penjual kopi ini sesuai dengan hasil yang diharapkan saat perancangan. Maka simulasi rangkaian mesin penjual kopi ini berfungsi dengan baik.

#### 6. Saran

Berdasarkan hasil simulasi dan analisa sampai selesainya penulisan ini, maka untuk melengkapi kekurangan perancangan rangkaian dan simulasi disarankan:

Rangkaian ini dapat dibuat secara sederhana dan praktis dengan menggunakan program xillinx. Karena xillinx dapat menerjemahkan schematic editor menjadi VHDL ( Very High Density Language ) yang berguna untuk memprogram FPGA ( Field Programmable Gate Array ).

## Daftar Pustaka:

- [1] Siswono, Hartono, dkk., *Perancangan Digital Menggunakan Simulator Xillinx*. DEPOK: Universitas Gunadarma, 2004.
- [2] Riris, Veronica., Sistem Indikator Kontrol Pada Vending Machine. Laporan Tugas Akhir. Jurusan Teknik Elektro Universitas Gunadarma, 2006.
- [3] F, Zuldin., Gerbang Logika Dasar Dan Gerbang Logika Lanjutan (Logic Gate), http://zhuldyn.files.wordpress.com/2012/0 4/gerbang-logika-dasar-dan-lanjutan.pdf, 2012, diakses 24 april 2012.
- [4] Syafari, Anjar., Mengenal Gerbang Logika (Logic Gate). http://elearning.gunadarma.ac.id/docmodul/naskah\_kursus\_perancangan\_sistem\_digital/5\_output\_forming\_logic.pdf, 2007, diakses 27 oktober 2007.
- [5] Gian, Melawak., Prinsip Rangkaian Flip-Flop, http://gianalonso.blogspot.com/2010/11/flip-flop-adalah-rangkaian-digital-yang.html, 2010, diakses 01 november 2010.