

# PERANCANGAN CONTROL UNIT, PROGRAM COUNTER (PC), DAN MEMORI INSTRUKSI PADA PROSESSOR RISC 16-BIT DENGAN TEKNOLOGI 600nm MENGGUNAKAN ELECTRIC

M Pramuaji Tri Saputro<sup>\*)</sup>, and Munawar.Darjat.

Jurusan Teknik Elektro, Universitas Diponegoro Semarang  
Jl. Prof. Sudharto, SH, kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>\*)</sup> Email: pramuaji93@gmail.com

## Abstrak

Processor adalah sebuah IC (integrated circuit) yang digunakan sebagai otak atau pengolah utama dalam sebuah sistem komputer. Pada processor terdapat 3 komponen utama yaitu Arithmetic Logic Unit (ALU), Control Unit, dan Memori. Pada penelitian ini dilakukan perancangan dan pengujian rangkaian control unit, program counter (PC) dan memori instruksi (ROM) pada processor 16-bit dengan mengacu pada arsitektur RISC. Pada perancangannya control unit menggunakan prinsip memori dengan masukan 4-bit opcode dan mengeluarkan 9-bit sinyal perintah yang akan mengatur jalannya elemen lain pada processor. Rangkaian program counter (PC) dirancang menggunakan jenis adder manchester yang dihubungkan dengan rangkaian sekuensial D-flipflop untuk media penyimpanan data. Memori instruksi menggunakan jenis memori ROM. Berdasarkan hasil perancangan dan pengujian didapatkan rangkaian control unit yang dapat bekerja pada 3 format instruksi, yaitu format instruksi R, I dan J. Rangkaian control unit memiliki luas area sebesar 125.184  $\mu\text{m}^2$  dan delay propagation maksimal adalah 1,008ns. Rangkaian program counter (PC) dan memori instruksi mengeluarkan 16-bit alamat yang akan didistribusikan ke elemen-elemen dalam processor seperti control unit dan register dengan luas area 923.653  $\mu\text{m}^2$  dan delay propagation maksimal sebesar 4,392 ns yaitu pada instruksi bit-10.

*Kata kunci : processor, ALU, control unit, program counter.*

## Abstract

Processor is an IC (integrated circuit) which is used as the brain or the main processor in a computer system. In the processor, there are 3 main element, namely arithmetic logic unit (ALU), Control Units, and Memory. In this research is to design and test the circuit control units, program counter (PC) and instruction memory (ROM) on a 16-bit processor with a reference to the RISC architecture. In its design, control unit uses the principle of memory with input of 4-bit opcode and releasing 9-bit command signal which will set the course of other elements on the processor. The circuit of the program counter (PC) is designed using adder manchester type connected with the sequence D-flipflop circuit for data storage media. Memory instruction using ROM memory type. Based on the design and testing result obtained that control unit circuit that can work on three instruction formats, formats instruction of R, I and J. The control unit circuit has an area of 125.184  $\mu\text{m}^2$  and maximum propagation delay is 1,008ns. The program counter circuit (PC) and memory instruction releasing 16-bit address that will be distributed to the processor elements such as control units and registers with area of 923.653  $\mu\text{m}^2$  and a maximum propagation delay of 4.392 ns in the 10-bit instruction.

*key word : processor, ALU, control unit, program counter*

### 1. Pendahuluan

Teknologi IC adalah salah satu bidang paling penting dalam dunia elektronik karena dengan teknologi ini dapat meminimalkan luas area dari rangkaian elektronik, efek parasitik dan biaya. Divais ini lebih handal dibandingkan dengan rangkaian yang tersusun dari komponen diskrit. Salah satu penerapan teknologi IC ini ada pada processor. Processor merupakan blok utama dari komputer digital dan beberapa sistem tertanam yang digunakan hampir oleh semua alat elektronik dalam kehidupan sehari - hari.

Semua proses komputasi seperti pengontrolan dan perhitungan dari divais tersebut dilakukan oleh sebuah processor. Untuk itu, didalam processor terdapat beberapa blok utama seperti *Arithmetic Logic Unit*(ALU), register dan memori, dan *control unit* untuk mendukung kerja dari processor. Salah satu penelitian tentang perancangan prosesor pernah dilakukan oleh John Enyeart dengan judul “8-bit MIPS Processor”, prosesor 8-bit ini memiliki lebar bus data 8-bit sehingga banyak data yang dapat diakses adalah FF8. Selain itu, prosesor ini mampu melakukan 10 instruksi, yaitu penjumlahan, pengurangan,

AND, OR, SLT, penjumlahan *immediate, branch, jump, load* dan *store data*. [1]

Kebutuhan akan kecepatan akses data dan lebar data pada sebuah divais elektronik sekarang ini semakin meningkat, oleh karena itu pada Penelitian ini dilakukan perancangan elemen *control unit, program counter (PC)* dan memori instruksi dalam prosesor RISC 16-bit yang memiliki 14 instruksi diantaranya adalah penjumlahan, pengurangan, AND, XOR, OR, *shift left, shift right, set less than unsigned*, penjumlahan *immediate, branch if equal, jump, load word* dan *store word* serta dapat melakukan operasi dengan akses data maksimal FFFFh.

Seperti yang telah dijelaskan di atas bahwa sebuah mikroprosessor terdiri dari 3 elemen penting salah satunya adalah *control unit* atau unit pengendali merupakan elemen paling penting di dalam prosessor, dengan bantuan unit pengendali proses di dalam prosessor akan diatur dan dikendalikan sehingga didapatkan output prosessor yang benar dan valid sesuai dengan instruksi yang diberikan. Karena menggunakan arsitektur RISC control unit memiliki desain yang sederhana serta memiliki hubungan input dan output yang jelas.

Perancangan ini difokuskan pada fungsional dari desain *control unit*, memori instruksi, dan *program counter (PC)* pada prosessor RISC 16 bit. Desain dibuat pada software Electric. Desain *control unit* mengacu pada desain memory ROM sehingga rangkaian kontrol akan mengeluarkan perintah kendali yang tepat untuk setiap instruksi. Desain disimulasikan menggunakan LT Spice untuk mengetahui karakteristik dari rangkaian yang telah didesain.

## 2. Metode

### A. Instruksi

Pada arsitektur RISC, untuk menjalankan fungsi yang sudah dibahas pada sub bab sebelumnya diperlukan sebuah instruksi sepanjang 16-bit untuk memberikan perintah. Instruksi ini memiliki urutan khusus, dimana instruksi tersebut diencode menjadi 4 template yaitu R, I, dan J, dan 1 buah instruksi NOP (*No Operation*). Berikut ini akan dijelaskan mengenai masing – masing instruksi yang ada pada arsitektur prosesor 16-bit sederhana ini : Instruksi format R :

Instruksi R ini digunakan untuk operasi aritmatik yang melibatkan register. Berikut ini adalah operasi aritmatik yang bisa dijalankan:

Tabel 1 operasi pada instruksi R

ALUopp	Instruksi	Keterangan
000	Add	Penjumlahan A + B
001	AND	Input A dioperasi logika AND dengan B
010	Sll( <i>shift left</i> )	Input A digeser ke kiri sebanyak 1 bit
011	Xor	Input A dioperasi logika XOR dengan B
100	Subb	Pengurangan A-B
101	OR	Input A dioperasi logika OR dengna B
110	Srl( <i>shift right</i> )	Input A digeser ke kanan sebanyak 1-bit
111	SLTU ( <i>set less than unsigned</i> )	Membandingkan nilai A dan B. Jika A<B, keluaran = 1. Selain itu keluaran = 0.

Tabel 2 menunjukkan format kode dari instruksi R :

Tabel 2 format kode instruksi R

Opcode	Ra	Rb	Rd	ALUopp
0000 [15-12]	[11-9]	[8-6]	[5-3]	[2-0]

Instruksi format I :

Berbeda dengan instruksi R, Instruksi I hanya memerlukan 1 buah blok data dari Register untuk diproses ALU, kemudian hasilnya disimpan dalam blok Register lainnya. Sehingga instruksi menggunakan 2 register (R1 dan Rd) ditambah 6-bit *immediate*. Input data immediate adalah *input* data yang tertulis langsung pada instruksi, sehingga lebar data ini lebih kecil dari data bus yang digunakan, terbatas pada format instruksi. Agar data immediate dapat diproses oleh ALU, data tersebut harus melalui sign extend terlebih dahulu agar menjadi data 16 bit. Format instruksi I ditunjukkan pada tabel 3 :

Tabel 3 Instruksi I

Opcode	Instruksi(I)	Keterangan
0011	Addi	penjumlahan A + input immediate
1010	LW	menuliskan data dari memori ke register
1011	SW	menyimpan data dari register ke memori
0100	BEQ	melompat beberapa instruksi jika data read register 1 dan 2 bernilai sama

Format *encode* dari instruksi I ditunjukkan pada tabel 4 berikut:

Tabel 4 Format Instruksi I

Opcode	Rb	Rd	Imm_value
[15-12]	[11-9]	[8-6]	[5-0]

Instruksi format J :

Pada format instruksi J hanya 1 buah instruksi yang bisa dieksekusi yaitu perintah “*jump*”, dimana berfungsi untuk melompat ke alamat instruksi yang dikehendaki, jadi disini akan dituliskan alamat yang ingin dituju. Tabel 5 merupakan format instruksi J :

Tabel 5 Format Instruksi J

Opcode	Address	Keterangan
0010 [15-12]	[11-0]	Bit [11-0] merupakan alamat yang akan dituju

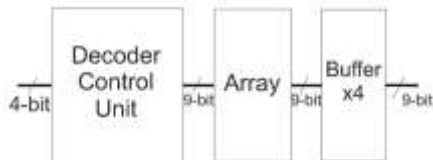
Instruksi format NOP :

Instruksi *no operation* merupakan instruksi yang digunakan untuk memberikan jeda pada prosesor. Sehingga pada instruksi ini prosesor tidak melakukan proses apapun, kecuali melakukan penjumlahan nilai register PC sebagai penunjuk alamat memori instruksi. Format instruksi *no operation* dijelaskan pada tabel 6 berikut :

Tabel 6 Format instruksi NOP

Opcode	NOP instruction set
0010 [15-12]	000000000000 [11-0]

B. Control Unit



Gambar 1 Blok diagram Control unit

Pada perancangan rangkaian *Control unit* dibagi menjadi 3 buah blok rangkaian yaitu decoder *Control unit*, array kontrolunit, dan buffer. Rangkaian decoder *Control unit* digunakan untuk menentukan instruksi yang akan dilakukan berdasarkan *input* 4-bit opcode yang diterima oleh rangkaian *Control unit*. Rangkaian array berisikan sel-sel yang akan aktif berdasarkan seleksi dari *output* decoder *Control unit*, sedangkan yang terakhir blok buffer digunakan untuk memperjelas logika sinyal keluaran dari array.

I/O Rangkaian Control Unit :

Berikut adalah *input* dan *output* rangkaian *Control unit* :

Tabel 7 I/O rangkaian Control unit

Input	Output
4 bit opcode [15:12]	9 bit sinyal kendali

Pada tabel 7 dapat dilihat bahwa *Control unit* atau unit pengendali mendapatkan *input* berupa 4 bit sinyal opcode

yang berasal dari memori instruksi bit [15:12]. Sinyal opcode ini kemudian akan diolah oleh rangkaian *control unit* sehingga menghasilkan 9 bit sinyal kendali yang nantinya akan digunakan untuk mengendalikan blok lain dalam mikroprosesor seperti blok register, memory, dan ALU (*arithmetic control unit*). Tabel 8 menunjukkan 9 bit sinyal kontrol dalam rangkaian *control unit* :

Tabel 8 Bit sinyal kontrol

Bit	Sinyal kontrol
0	Regdst
1	Regwrite
2	Alusrc
3	Aluopp
4	Memwrite
5	Memena
6	Memtoreg
7	Branch
8	Jump

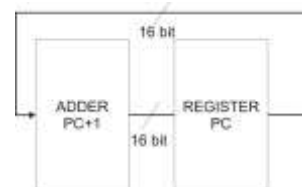
Setelah pemaparan mengenai I/O dan sinyal kontrol pada rangkaian *control unit* berikut akan diperlihatkan hubungan antara input dan output dari rangkaian CU :

Tabel 9 Hubungan *input* dan *output* pada rangkaian *Control unit*

Input (opcode)	Output[8-0]	Format instruksi
0000	000001011	R
0011	000000110	I
0010	100000100	J
1011	000110100	SW
1010	001100110	LW
0100	010000100	BEQ
1111	000000000	NOP

Pada tabel 9 diperlihatkan mengenai hubungan *input* dan *output* rangkaian kontrol unit, apabila rangkaian *control unit* mendapatkan *input* berupa bit opcode dengan logika ‘0000’ maka *output* rangkaian akan berlogika ‘000001011’ yang menandakan rangkaian sedang menjalankan instruksi format R.

C. Clock Counter



Gambar 2 Blok diagram Program Counter

Pada perancangan rangkaian register PC dibagi menjadi 2 buah blok rangkaian yaitu adder PC+1 dan register PC. Rangkaian adder PC+1 digunakan untuk melakukan penjumlahan *output* dari register PC dengan menggunakan prinsip adder manchester carry chain sedangkan rangkaian register PC digunakan untuk melakukan pengalamatan pada memori instruksi. I/O Rangkaian Program Counter (PC) :

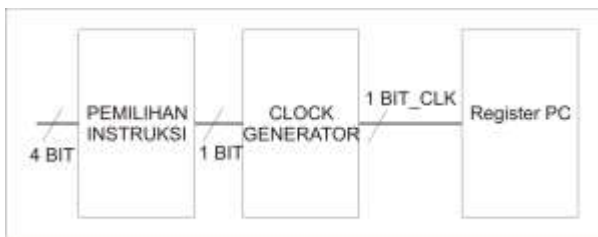
Berikut adalah *input* dan *output* rangkaian Program Counter (PC) :

Tabel 10 I/O Program Counter (PC)

Input	Output
Clock (pulsa positif)	16 bit address

Pada tabel 10 dapat dilihat bahwa *input* rangkaian Program Counter (PC) hanya berupa 1 bit pulsa clock sedangkan *output* rangkaian adalah 16 bit address. Keluaran 16-bit address adalah hasil penjumlahan antara keluaran register PC dan nilai konstan 1 desimal.

D. Program Counter



Gambar 3 Blok Diagram Counter

Pada perancangan rangkaian clock counter dibagi menjadi 2 buah blok rangkaian yaitu blok pemilihan instruksi dan blok clock generator. Perancangan rangkaian counter sendiri menggolongkan instruksi panjang yaitu 4 kali clock global dan instruksi pendek yaitu 2 kali clock global. Blok pemilihan instruksi digunakan untuk melakukan seleksi instruksi mana yang membutuhkan clock panjang atau pendek sedangkan pada block clock generator digunakan untuk pembangkit sinyal clock panjang dan pendek.

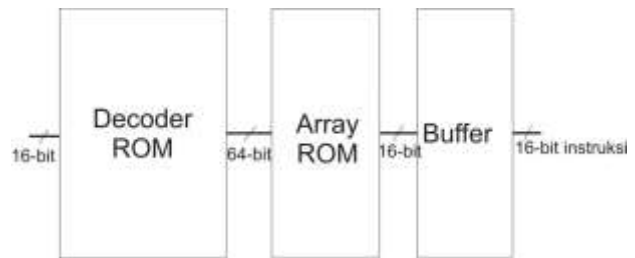
I/O Rangkaian Program Counter (PC) :  
Berikut adalah *input* dan *output* rangkaian clock counter :

Tabel 11 I/O rangkaian Clock Counter

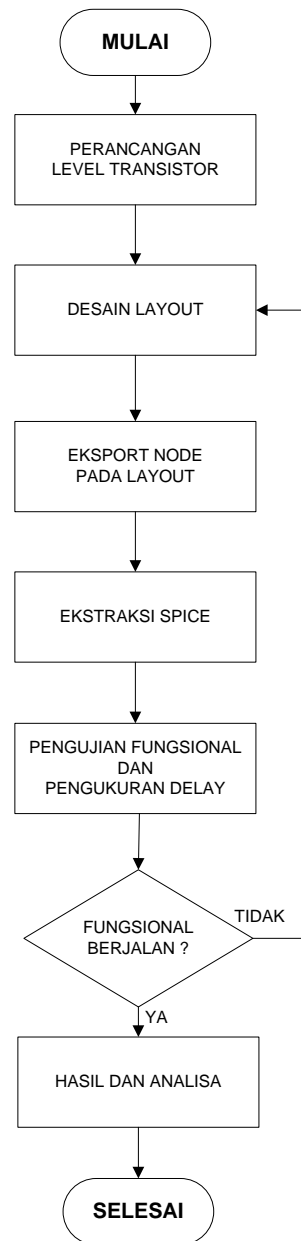
Instruksi	Input (opcode)	Status Clock
R	0000	Panjang
I	0011	Panjang
Jump	0010	Pendek
SW	1011	Panjang
LW	1010	Panjang
BEQ	0100	Pendek
NOP	1111	Pendek

Pada tabel 11 dapat dilihat bahwa *input* rangkaian pemilihan instruksi berjumlah 4 bit opcode dari memori instruksi. Rangkaian pemilihan instruksi ini digunakan untuk menggolongkan instruksi panjang dan pendek. Instruksi yang membutuhkan clock panjang adalah instruksi R, I, SW, dan LW sedangkan instruksi BEQ, NOP, dan Jump membutuhkan clock pendek.

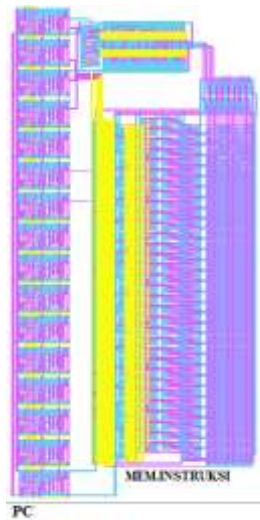
E. Memori Instruksi (ROM)



Gambar 4 Blok diagram memori instruksi



Gambar 5 Diagram alir metode pengujian



Gambar 6 Layout rangkaian program counter (PC) dan memori instruksi

Dapat dilihat pada gambar 6 bahwa perancangan rangkaian *Memori Instruksi* dibagi menjadi 3 buah blok rangkaian yaitu *Decoder Memori Instruksi*, *Array Memori Instruksi*, dan *Buffer*. Rangkaian *decoder Memori Instruksi* digunakan untuk menentukan sel *memori* aktif yang ditunjuk berdasarkan *input address* dari rangkaian register PC, rangkaian *array Memori Instruksi* berisikan bit instruksi yang nilainya sudah ditentukan, sedangkan rangkaian *buffer* digunakan untuk menguatkan sinyal keluaran dari *array Memori Instruksi*. Pada perancangan rangkaian *Memori Instruksi* ini memiliki jumlah sel *memori* sebanyak 64 dengan lebar sel 16-bit.

I/O Rangkaian Memori Instruksi :

Tabel 12 I/O Memori Instruksi

Input	Output
16-bit alamat	16-bit instruksi

Dapat dilihat pada tabel 12 bahwa rangkaian *Memori Instruksi* mendapatkan *input* dari register PC berupa 16-bit alamat dan mengeluarkan 16-bit instruksi sesuai dengan pengalamatan dari register PC. Pada rancangan memori instruksi ini bit address yang digunakan hanya 6 bit LSB dikarenakan kapasitas memori instruksi adalah 64x16 bit.

### 3. Hasil dan Analisa

#### A. Metode Pengujian

Pada gambar 10 dapat dilihat diagram alir metodologi perancangan dan pengujian penelitian ini. Pada awalnya dilakukan perancangan rangkaian pada level transistor. Hal ini digunakan sebagai dasar pembuatan layout rangkaian dengan software Electric. Setelah pembuatan desain layout, tahap berikutnya adalah pemberian node

sinyal input dan output pada layout rangkaian. Selanjutnya hasil layout yang telah diberi node, diekstrak ke dalam software LT-spice untuk dilakukan pengujian fungsional dan delay. Apabila hasil keluaran tidak sesuai dengan fungsional, maka akan dilakukan desain ulang dalam level layout. Bila keluaran sudah sesuai dengan fungsional, selanjutnya dilakukan analisa terhadap hasil pengujian fungsional dan pengukuran delay.

#### B. Pengujian program counter (PC) dan Memori Instruksi

Pengujian fungsional rangkaian program counter (PC) dan memori instruksi seperti pada gambar 11 dilakukan dengan memberikan masukan tunggal yaitu clock. Pulsa masukan clock yang diberikan memiliki periode 20ns dan duty cycle 50%. Pada ROM telah ditentukan beberapa set instruksi diantaranya :

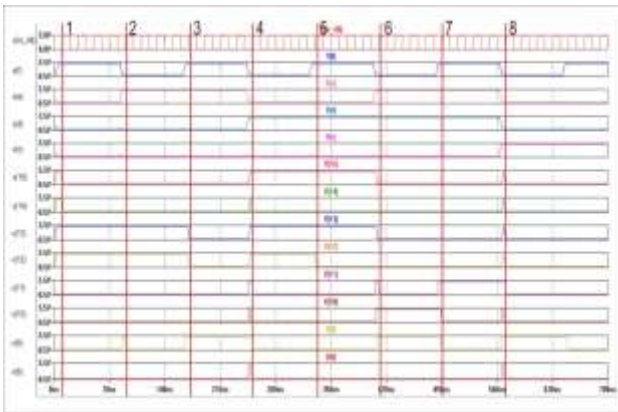
Tabel 13 Set instruksi pada memori instruksi (ROM)

No	Instruksi	Bit instruksi
1	Addi R0,R1,24	0011000001011000
2	Addi R0,R2,10	0011000010001010
3	SLL R1,R0,R1	0000001000001010
4	SW R0,R1,000000	1011000001000000
5	LW R0,R3,000000	1010000011000000
6	ADD R3,R2,R4	0000011010100000
7	SUBB R4,R2,R1	0000100010001100
8	SRL R1,R0,R1	0000001000001110

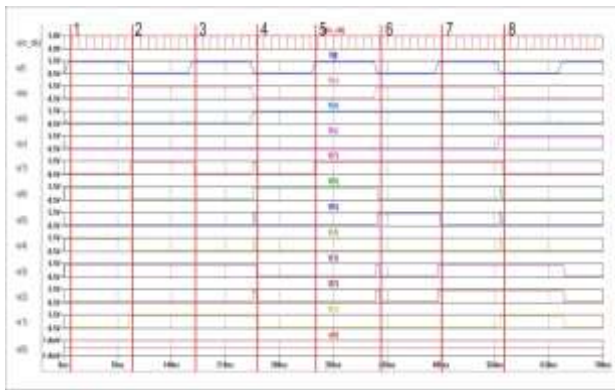
Pada tabel 13 dapat dilihat instruksi pertama bermakna register R1 dimasukan nilai  $24_{10}$  yang berarti keadaan awal register tersebut berisi nilai  $24_{10}$ . Instruksi kedua register R2 dimasukan nilai  $10_{10}$  yang berarti nilai register R2 adalah  $10_{10}$ . Instruksi ketiga register R1 nilainya akan di *shift left* yang berarti nilai register R1 adalah  $48_{10}$ . Instruksi keempat nilai dari register R1 akan disimpan pada memori dengan alamat  $(R0+0_{10})$ . Instruksi kelima isi memori pada blok 0 akan dimuat pada register dengan alamat R3. Instruksi keenam isi register R2 akan ditambah dengan isi register R3 dan hasilnya akan disimpan pada register R4. Instruksi ketujuh isi register R2 ditambah dengan isi register R4 dan hasilnya akan disimpan pada register R1. Instruksi kedelapan isi register R1 akan di *shift right* sehingga register tersebut berisi data  $24_{10}$ . Parameter keberhasilan pengujian fungsional pada blok program counter (PC) dan memori instruksi apabila nilai bit instruksi yang dikeluarkan oleh memori instruksi sama dengan bit instruksi yang tertera pada tabel 13.

Berikut adalah hasil pengujian rangkaian program counter dan memori instruksi dengan spesifikasi pengujian seperti dijelaskan diatas :





Gambar 7 Gelombang hasil simulasi rangkaian program counter (PC) dan memori instruksi [15:8]



Gambar 8 Gelombang hasil simulasi rangkaian program counter (PC) dan memori instruksi [7:0]

Berdasarkan gambar 13 didapatkan tabel pengujian fungsional dan delay seperti berikut :

Pengujian	Bit instruksi															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	1	0	0	0	0	0	1	0	1	1	0	0	0
2	0	0	1	1	0	0	0	0	1	0	0	0	1	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
4	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
5	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0
6	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0
7	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0

Gambar 9 tabel hasil pengujian fungsional rangkaian PC dan memori instruksi

Berdasarkan tabel 14 dapat disimpulkan bahwa pengujian yang dilakukan terhadap rangkaian program counter (PC) dan memori instruksi sesuai dengan fungsional, namun terdapat delay dalam pengujian. Berikut adalah tabel

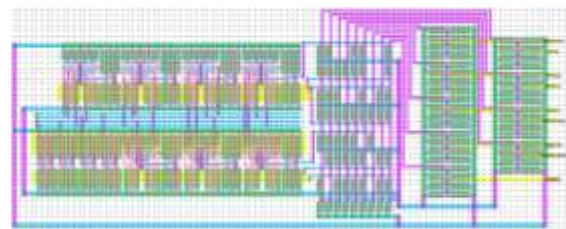
analisis delay dan area dari rangkaian program counter dan memori instruksi :

Tabel 14 analisis delay dan area pada rangkaian program counter (PC) dan memori instruksi

Rangkaian program counter (PC) dan memori instruksi						
Area		3.078.844 lambda				
No	Jenis	$t_{pd}$ (ns)	$t_{pdf}$ (ns)	$t_{pd}$ (ns)	$t_r$ (ns)	$t_f$ (ns)
1	Bestcase	2.867	2.917	2.892	0.978	0.871
2	worstcase	3.117	5.668	4.392	0.292	0.389

Pada tabel 14 dapat dilihat untuk delay propagasi ( $t_{pd}$ ) bestcase adalah 2.892ns yaitu pada keluaran memori instruksi bit-15 dan delay propagasi ( $t_{pd}$ ) worstcase adalah 4.392 yaitu pada keluaran memori instruksi bit-10.

C. Pengujian Rangkaian Control Unit



Gambar 10 Layout rangkaian control unit

Pengujian fungsional layout kontrol unit yang ditunjukkan seperti gambar 10 diatas dilakukan dengan cara memberikan pulsa tegangan masukan sebesar 5vdc pada setiap gate transistor *input* yang ditunjukkan oleh node a,b,c, dan d dengan asumsi bahwa nilai logika *output* dari memori instruksi benar. Berikut adalah gambar gelombang hasil pengujian rangkaian *control unit* beserta penjelasannya :



Gambar 11 Gelombang hasil simulasi rangkaian control unit

Dari gambar 11 dapat dilihat saat pengujian 1, *input* a,b,c,d berlogika '0011' dan *output* rangkaian berlogika '00000110' bahwa instruksi yang dijalankan adalah instruksi I. Pada pengujian 6 masukan berlogika '0000' dan *output* rangkaian berlogika '00001011' bahwa

instruksi yang dijalankan adalah instruksi R. Untuk lebih jelasnya dapat dilihat pada tabel 4.24 :

Tabel 15 pengujian fungsional rangkaian control unit

No	Input (biner)	Output (biner)
1	0011	00000110
2	0011	00000110
3	0000	00001011
4	1011	000110100
5	1010	001100110
6	0000	00001011
7	0000	00001011
8	0000	00001011

Dari tabel 15 dapat dilihat untuk semua pengujian yang dilakukan menghasilkan output rangkaian control unit dengan logika yang benar. Namun output rangkaian tidak dapat langsung merespon perubahan input dikarenakan adanya delay, berikut adalah tabel analisis delay dan area dari rangkaian control unit :

Tabel 16 analisis delay dan area pada rangkaian control unit

Rangkaian control unit						
Area		417.280 lambda				
No	Jenis	$t_{pdf}$ (ns)	$t_{pdf}$ (ns)	$t_{pd}$ (ns)	$t_r$ (ns)	$t_f$ (ns)
1	worstcase	1.087	0.93	1.008	0.082	0.074

Pada tabel 16 dapat dilihat untuk delay propagation (tpd) worstcase adalah 1,008ns yaitu pada keluaran rangkaian control unit Regdst dengan nilai delay rising sebesar 0,082ns dan delay falling 0,074ns.

#### 4. Kesimpulan

Berdasarkan perancangan dan analisis yang dilakukan terhadap rangkaian maka dapat disimpulkan sebagai berikut :

1. Perancangan prosessor menggunakan memori instruksi dan memori data terpisah dikarenakan mengacu pada arsitektur Harvard.

2. Terdapat 3 jenis penggolongan instruksi yaitu instruksi format R, I dan J. Dimana format kode pada tiap-tiap instruksi berbeda.
3. Rangkaian program counter (PC) digunakan untuk memberikan alamat kepada memori instruksi.
4. Terdapat 2 penggolongan pewaktu berdasarkan instruksi yang sedang dijalankan. Pewaktu pendek dengan 2 kali pewaktu global dan pewaktu panjang dengan 4 kali pewaktu global. Instruksi yang membutuhkan pewaktu pendek adalah instruksi BEQ, JUMP dan NOP, sedangkan instruksi yang membutuhkan pewaktu panjang adalah instruksi R, I, LW dan SW.
5. Rangkaian control unit menggunakan prinsip memori. Dengan masukan berupa 4-bit opcode dan mengeluarkan 9-bit perintah yang digunakan untuk mengatur elemen lain di dalam prosessor.
6. Keluaran rangkaian program counter (PC) dan memori instruksi memiliki delay propagation maksimal atau keadaan worstcase sebesar 4,392ns dan delay propagation minimum atau keadaan bestcase sebesar 2,892ns. Delay worstcase terdapat pada instruksi bit-10 dan delay bestcase terdapat pada instruksi bit-15.
7. Keluaran rangkaian control unit memiliki delay maksimal atau keadaan worstcase sebesar 1.008ns yaitu pada Regdst.

#### Referensi

- [1]. R.M. Enyeart.John, Mohammed AL-Karawi, "8-Bit MIPS Processor," Senior Design Report, UNLV, 2014.
- [2]. MOSIS, "C5model," 2009. [Online]. Available: [http://cmosedu.com/cmos1/electric/C5\\_models.txt](http://cmosedu.com/cmos1/electric/C5_models.txt). [Accessed: 10-Feb-2014].
- [3]. S. M.Rubin, "Electric (Software)," 2014. [Online]. Available: <http://cmosedu.com/cmos1/electric/electric.htm>. [Accessed: 14-Feb-2014].
- [4]. M. Engelhardt, "LTspice." Linear Technology, Inc, 2015.
- [5]. H.E. Weste.Neil, dan David Money Harris, "CMOS VLSI DESIGN", Pearson Education, 2011.