

PERANCANGAN ARITHMATIC LOGIC UNIT (ALU) PADA PROSESOR RISC 16-BIT DENGAN TEKNOLOGI 600NM MENGGUNAKAN ELECTRIC

Rizka Hendrawan^{*)}, Munawar Agus Riyadi, and Darjat

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

^{*)}E-mail: rizkahendrawan@yahoo.com

Abstrak

Teknologi IC merupakan salah satu teknologi yang paling penting dalam dunia elektronik saat ini, karena dapat meminimalkan luas area dari rangkaian elektronik, efek parasitik dan biaya. Divais ini lebih handal dibandingkan dengan komponen diskrit. Salah satu penerapannya ada pada prosesor yang merupakan unit pusat pemrosesan data pada komputer digital dan sistem tertanam untuk keperluan sehari-hari. Tujuan dari Tugas Akhir ini adalah merancang layout *Arithmetic Logic Unit* (ALU) pada prosesor RISC 16-bit pada level transistor. Metode yang digunakan dalam perancangan *Arithmetic Logic Unit* (ALU) ini menggunakan teknologi fabrikasi 600nm dari model C5 MOSIS menggunakan software Electric untuk mendesain layout, serta software LT-spice untuk menguji hasil ekstraksi dari layout. Hasil pengujian perancangan ini menunjukkan bahwa *Arithmetic Logic Unit* (ALU) memiliki 8 fungsi yaitu penjumlahan, pengurangan, AND, OR, XOR, SLL, SRL, SLTU. Secara fungsional, ALU dapat mengeluarkan data sesuai dengan sinyal kontrol. *Time propagation delay* (T_{pd}) yang terjadi pada ALU sebesar 13ns. Nilai ini merupakan rata-rata dari *time propagation delay rise* (T_{pdr}) sebesar 10ns dan *time propagation delay fall* (T_{pdf}) sebesar 16ns. Sedangkan nilai *rise time* (T_r) sebesar 0.22ns dan *fall time* (T_f) sebesar 0.15ns. Oleh karena itu, ALU dapat beroperasi maksimal pada *clock* sebesar 62.5MHz.

Kata kunci: Prosesor, Arithmetic Logic Unit (ALU), teknologi fabrikasi 600nm.

Abstract

IC technology is one of the most important technologies in electronic world now, because IC technology can minimize the area of electronic circuit, parasitic effect and cost. Device based in IC technology is more reliable than the discrete components. One of the implementation is in the processor which is a unit of the data processing center on digital computer and embedded system for daily needs. The aim of this final project is to design the floorplan layout of Arithmetic Logic Unit (ALU) on 16-bit RISC processor in transistor level. The method used in designing Arithmetic Logic Unit (ALU) is using 600nm fabrication technology from C5 MOSIS model and using Electric software to design the layout and also LT-spice software to examine extraction result from the layout. The testing result shows that Arithmetic Logic Unit (ALU) has eight functions: addition, reduction, AND, OR, XOR, SLL, SRL, SLTU. Functionally, ALU can issue the data according to the control signal. Time propagation delay (T_{pd}) that occurs in the ALU is 13ns. This value is the average of time propagation delay rise (T_{pdr}) of 10ns and time propagation delay fall (T_{pdf}) of 16ns. Moreover, the value of rise time (T_r) is 0.22ns and fall time (T_f) is 0.15ns. Therefore, ALU can operate at maximum 62.5MHz clock frequency.

(Times New Roman 10)

Keywords: Processor, Arithmetic Logic Unit (ALU), 600nm fabrication technology.

1. Pendahuluan

Teknologi IC adalah salah satu bidang paling penting dalam dunia elektronik, karena 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 prosesor.

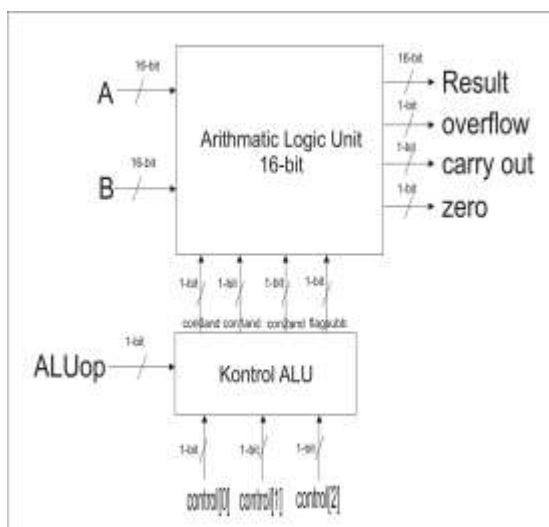
Prosesor merupakan blok utama dari komputer digital dan beberapa sistem tertanam yang digunakan hampir oleh semua alat elektronik dalam kehidupan sehari-hari. Semua proses komputasi seperti pengontrolan dan perhitungan dari divais tersebut dilakukan oleh sebuah prosesor. Penelitian tentang perancangan prosesor pernah dilakukan oleh John Enyeart dengan judul "8-bit MIPS Processor". Prosesor 8-bit ini memiliki lebar data 8-bit,

sehingga maksimal data yang dapat diakses adalah FF8. Selain itu, prosesor ini hanya mampu melakukan 10 instruksi, yaitu penjumlahan, pengurangan, AND, OR, SLT, penjumlahan *immediate, branch, jump, load* dan *store data*[1]. Karena kebutuhan akan kecepatan akses data dan lebar data pada sebuah divais elektronik pada saat ini semakin meningkat, prosesor dengan lebar data 8-bit kurang mampu memenuhi kebutuhan yang ada dibandingkan dengan prosesor dengan lebar data yang lebih besar seperti prosesor 16-bit, sehingga kapasitas register ,akses data, dan jumlah data yang dapat diakses lebih besar dan cepat.

Prosesor terdiri dari beberapa blok utama sebagai penunjang kerja prosesor. Beberapa blok tersebut adalah *control unit*, register, *arithmetic logic unit* (ALU), dan memori. *Arithmetic Logic Unit* (ALU) adalah unit utama dari mikroprosesor yang melakukan proses arimatik seperti penjumlahan, pengurangan, penggeseran bit, perbandingan nilai A dan B, dan juga operasi logika AND, OR, XOR,dll. ALU dalam melakukan operasi aritmatik dikontrol oleh *control unit* dan instruksi dari memori instruksi.

Berdasarkan latar belakang tersebut, Tugas Akhir ini mengambil judul perancangan *Arithmetic Logic Unit* (ALU) prosesor RISC 16-bit yang memiliki 8 buah fungsi seperti penjumlahan, pengurangan, penggeseran bit ke kiri dan kanan, perbandingan nilai 2 buah operand, dan operasi logika AND, OR, dan XOR. Perancangan ini menggunakan teknologi fabrikasi 600nm dari model C5 MOSIS[2] yang memiliki 3 metal teknologi CMOS, dan tegangan sumber sebesar 5V. *Software Electric*[3] digunakan untuk membuat desain layout ALU, dan *software LT-spice*[4] untuk menguji fungsional dan delay dari hasil ekstraksi desain layout ALU yang dibuat.

2. Metode



Gambar 1. Blok diagram ALU 16-bit

Gambar 1 merupakan blok diagram *Arithmetic Logic Unit* 16-bit beserta blok kontrolnya. ALU 16-bit tersusun dari 16 buah ALU 1-bit. Tiap ALU 1-bit terdiri dari blok penyusun seperti *two's complement*, *full adder*, gerbang AND, OR, XOR, SLTU, SLL, SRL, serta 2 buah multiplexer 4-ke-1 dan 1 buah multiplexer 2-ke-1 yang berguna untuk mengontrol keluaran dari ALU 1-bit.

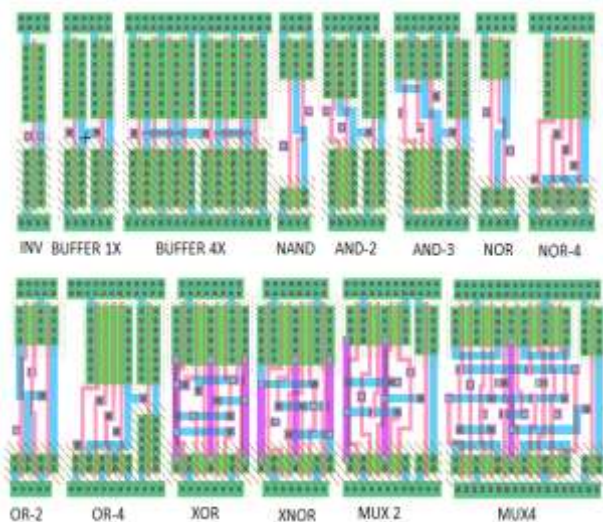
Arithmetic Logic Unit (ALU) ini mempunyai 8 buah fungsi yang dikontrol oleh sinyal keluaran 4-bit dari blok kontrol ALU. Tabel 1 menampilkan 8 fungsi ALU beserta sinyal keluaran 4-bit dari blok kontrol ALU.

Tabel 1. Tabel fungsi dan kontrol ALU

Fungsi	Sinyal Kontrol ALU	Sinyal kontrol two's complement
Penjumlahan	000	0
AND	001	0
Shift Left (SLL)	010	0
XOR	011	0
Pengurangan	100	1
OR	101	0
Shift right(SRL)	110	0
Set Less Than	111	0
Unsigned(SLTU)	111	0

2.1. Standard Cell

Untuk membentuk blok penyusun ALU, diperlukan standard cell/ layout gerbang logika. Gambar 2 merupakan *standard cell* yang digunakan pada perancangan ini sebagai penyusun blok rangkaian ALU.

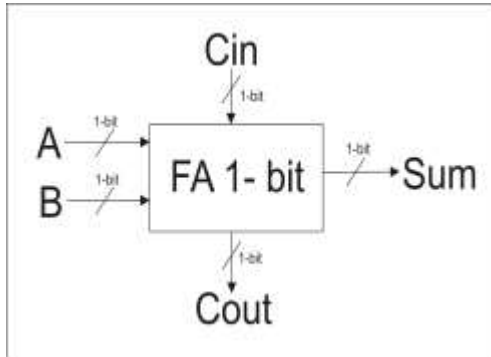


Gambar 2. Standard cell

2.2. Full Adder 1-bit

Gambar 3 merupakan diagram blok full adder 1-bit yang menggunakan metode *manchester carry chain*. Full adder 1-bit ini memiliki 3 buah masukan, yaitu A,B, dan Cin. Juga memiliki 2 buah keluaran, yaitu Sum dan Cout.

Untuk membentuk sebuah full adder 16-bit, full adder 1-bit disusun secara *cascade*. Sehingga Cout akan masuk ke Cin bit sesudahnya.

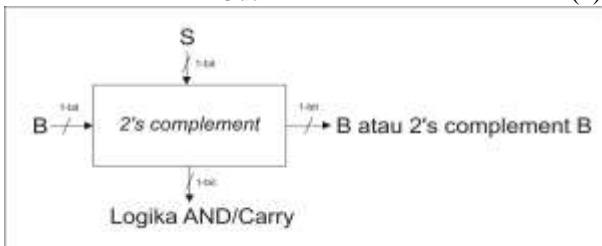


Gambar 3. Blok diagram Full adder 1-bit

2.3. 2's Complement

Gambar 4 merupakan blok diagram 2's complement 1-bit. 2's complement 'ini memiliki dua buah masukan yaitu data B dan S sebagai saklar multiplekser, dan satu buah keluaran yaitu Out. Fungsi logika dari blok diagram pada gambar 4 adalah sebagai berikut :

$$\text{Out} = Bn + 1 \quad (1)$$



Gambar 4. Blok diagram 2's complement

2.4. Set Less Than Unsigned (SLTU)

Gambar 5 merupakan blok diagram dari fungsi SLTU 1-bit. Dimana mempunyai 2 buah masukan, yaitu data A dan B, dan mempunyai 1 buah keluaran yaitu outSLT. Pada fungsi ini, nilai A akan dibandingkan dengan nilai B, jika nilai $A < B$ maka keluaran bernilai '1', selain itu keluaran bernilai '0'.

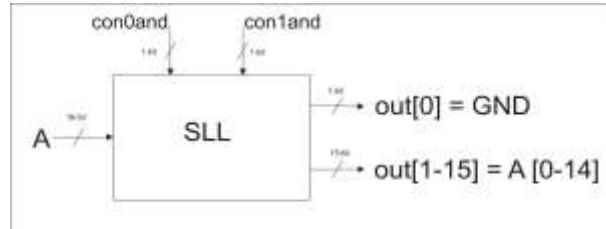


Gambar 5. Blok diagram SLTU 1-bit

2.5. Shift Left Logic (SLL)

Gambar 6 merupakan blok diagram SLL 16-bit. Blok ini memiliki 18 masukan, yaitu data 16-bit dan saklar 2-bit.

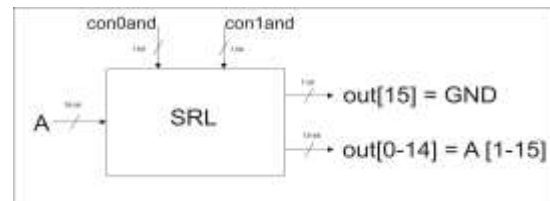
Keluaran pada blok ini adalah hasil SLL 16-bit. Dimana data A[15:0] tiap bit nya digeser ke kiri satu kali, sehingga bit LSB/paling kanan diisi dengan logika '0' dan bit MSB awal hilang dan diganti dengan bit[14].



Gambar 6. Blok diagram SLL 16-bit

2.6. Shift Right Logic (SRL)

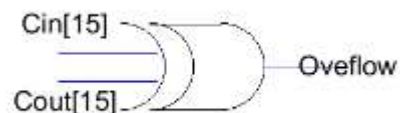
Gambar 7 merupakan blok diagram SRL 16-bit. Blok ini memiliki 18 masukan, yaitu data 16-bit dan saklar multiplekser 2-bit. Keluaran pada blok ini adalah hasil SRL 16-bit. Dimana data A[15:0] tiap bit nya digeser ke kanan satu kali, sehingga bit MSB/paling kiri diisi dengan logika '0' dan bit LSB awal hilang, kemudian diisi dengan bit[1].



Gambar 7. Blok diagram SRL 16-bit

2.7. Overflow

Overflow merupakan sebuah tanda atau *flag* yang hanya diperhatikan saat melakukan operasi penjumlahan/pengurangan dengan bilangan *signed*. Jika overflow bernilai '1', berarti hasil perhitungan yang dilakukan melebihi dari kapasitas bit yang tersedia, dan hasil yang dihasilkan berarti salah. Overflow bisa didapat melalui hasil XOR dari Cin (n-1) dengan Cout (N-1). Gambar 8 merupakan rangkaian *overflow* dengan gerbang xor.

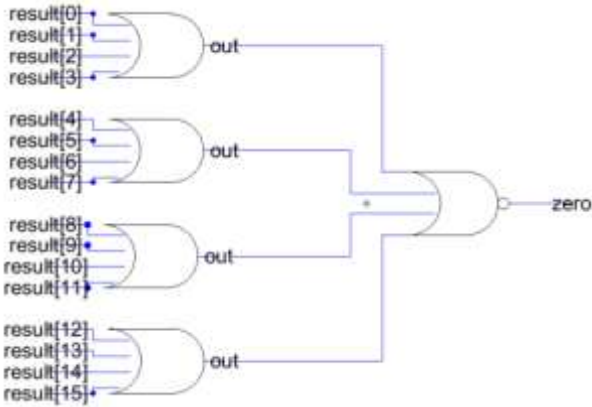


Gambar 8. Overflow dengan gerbang XOR

2.8. Zero Detector

Zero detector ini digunakan untuk mendeteksi apabila semua hasil ALU 16-bit bernilai '0', maka rangkaian ini akan menghasilkan logika '1'. Rangkaian ini disusun dari

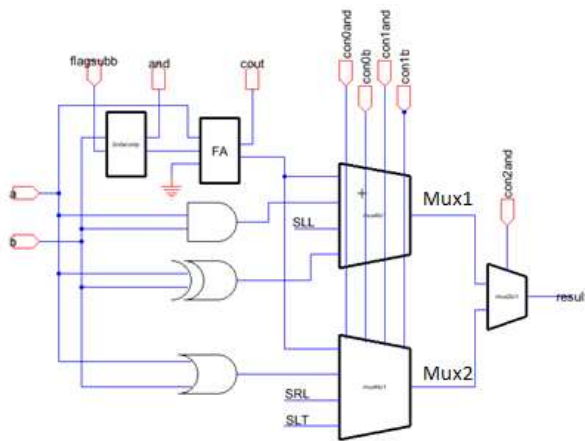
4 buah gerbang OR, dan 1 buah gerbang NOR. Gambar 9 merupakan rangkaian *zero detector* yang digunakan pada ALU 16-bit.



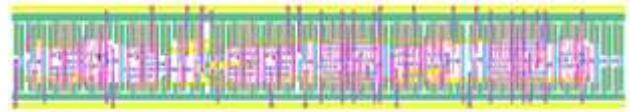
Gambar 9. Gerbang logika zero detector ALU 16-bit

2.9. Arithmetic Logic Unit (ALU) 1-bit

Gambar 10 merupakan rangkaian dari ALU 1-bit yang mempunyai 8 fungsi, terlihat bahwa seluruh komponen ALU akan tetap aktif, tetapi *result* yang dikeluarkan akan dikontrol oleh 3 multiplexer. Sinyal kontrol untuk multiplexer ini datang dari blok kontrol ALU. Gambar 11 merupakan layout dari ALU 1-bit.



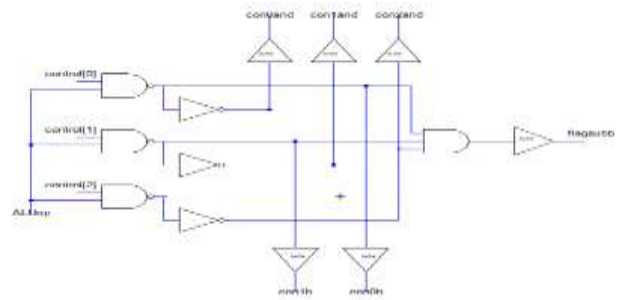
Gambar 10. Blok diagram ALU 1-bit dan control



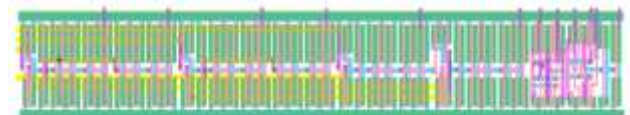
Gambar 11. Layout ALU 1-bit

2.10. Kontrol ALU

Fungsi Kontrol ALU disini adalah meneruskan sinyal kontrol 3-bit dari instruksi dan sinyal *enable* 1-bit ALUop dari *control unit* untuk mengoperasikan multiplexer guna menentukan operasi mana yang akan dilakukan, dan juga memberikan sinyal ke multiplexer *two's complement* untuk memilih apakah keluaran *two's complement* nilai B atau *two's complement* dari B. Gambar 12 merupakan rangkaian dari blok kontrol ALU dan gambar 13 merupakan hasil layout kontrol ALU.



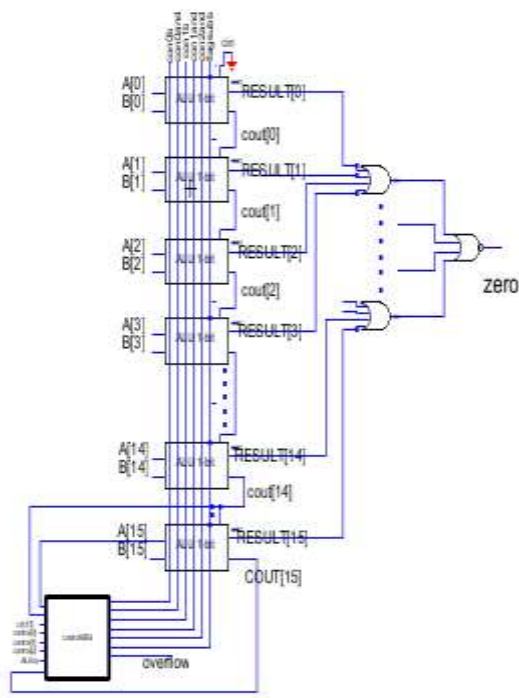
Gambar 12. Rangkaian kontrol ALU



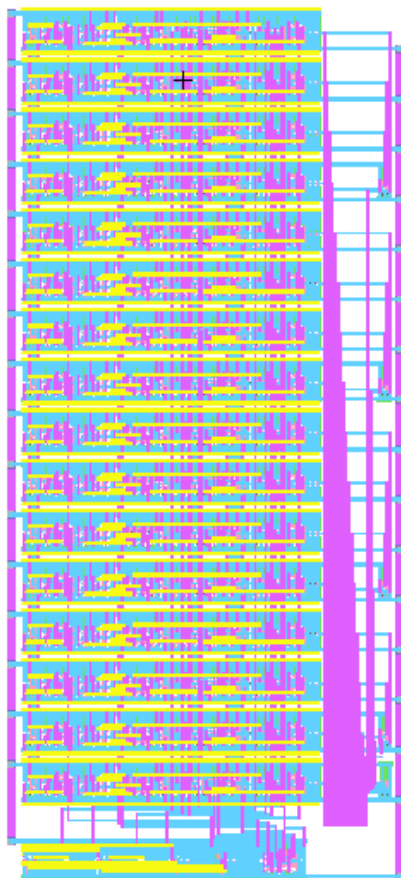
Gambar 13. Layout kontrol ALU

2.11. Arithmetic Logic Unit (ALU) 16-bit

Setelah mendesain ALU 1-bit, untuk membentuk sebuah rangkaian 16-bit ALU, disusun sebanyak 16 buah ALU 1-bit dan saklar tiap multiplexer harus disambung pada seluruh bit, juga *output* carry out bit sebelumnya disambung ke carry in pada bit sesudahnya, dan keluaran logika AND *two's complement* bit sebelumnya disambung dengan masukan AND dan XOR *two's complement* bit berikutnya. Gambar 14 merupakan rangkaian dari ALU 16-bit dan gambar 15 merupakan layout ALU 16-bit



Gambar 14. Rangkaian ALU 16-bit

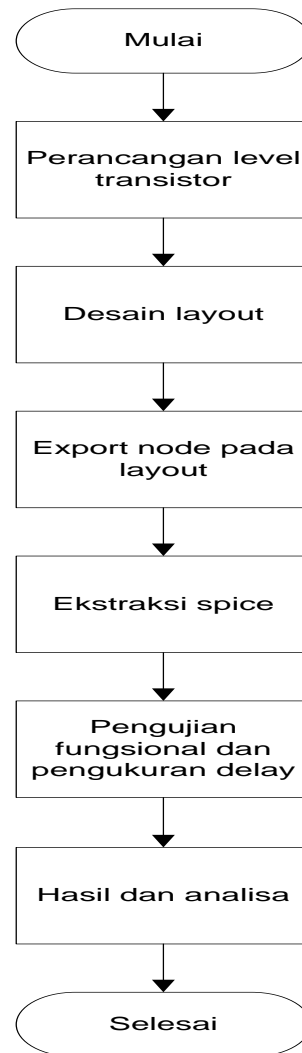


Gambar 15. Layout ALU 16-bit

3. Hasil dan Analisa

Pengujian hasil perancangan menggunakan software LT-spice untuk mengetahui fungsional dan *timing delay* dari hasil ekstraksi layout yang dibuat.

3.1. Metode Pengujian



Gambar 16. Diagram alir metode pengujian

Pada gambar 16 dapat dilihat diagram alir metodologi perancangan dan pengujian tugas akhir 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. Kemudian dilakukan analisa terhadap hasil pengujian fungsional dan pengukuran delay.

3.2. Standard Cell

Pengujian standard cell dilakukan menggunakan fanout yang berbeda-beda untuk tiap cell nya, sesuai dengan berapa gerbang yang akan di *drive* pada blok ALU. Secara fungsional *standard cell* ini sudah mengeluarkan logika yang benar sesuai dengan tabel kebenaran gerbang logikanya. Tabel 2 menampilkan hasil pengukuran *timing delay* pada standard cell.

Tabel 2. Tabel *timing delay standard cell*

Standard Cell	Delay (ns)					
	Beban	Tpdr	Tpdf	Tpd	Tr	Tf
Inverter	62,4ff	0.12	0.09	0.105	0.13	0.082
Buffer 1x	62,4ff	0.19	0.19	0.19	0.134	0.12
Buffer 4x	1000ff	0.33	0.3	0.315	0.353	0.24
NAND	31,2ff	0.08	0.23	0.155	0.07	0.26
AND-2	62,4ff	0.4	0.3	0.35	0.27	0.25
AND-2	62,4ff	0.24	0.17	0.205	0.14	0.09
AND-3	31,2ff	0.25	0.16	0.205	0.12	0.07
NOR	31,2ff	0.24	0.17	0.205	0.29	0.19
NOR-4	31,2ff	0.35	0.32	0.335	0.42	0.39
OR-2	31,2ff	0.3	0.4	0.35	0.16	0.20
OR-4	31,2ff	0.53	0.45	0.49	0.16	0.15
XOR	93,6ff	0.45	0.53	0.49	0.45	0.66
XNOR	31,2ff	0.35	0.32	0.335	0.24	0.35
Mux 2	93,6ff	0.45	0.53	0.49	0.29	0.37
Mux 4	31,2ff	0.72	0.63	0.675	0.26	0.28

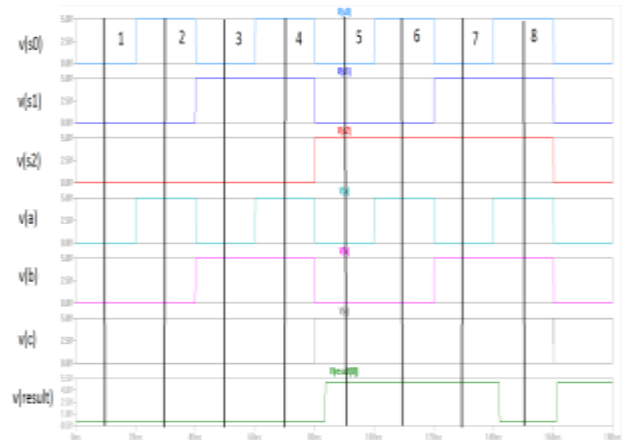
3.3. Blok Rangkaian

Pengujian blok rangkaian menggunakan fanout yang berbeda-beda tiap bloknnya, sesuai dengan penggunaan pada blok ALU. Secara fungsional, blok ini sudah menghasilkan logika yang benar sesuai dengan masukannya. Tabel 3 menampilkan hasil pengukuran *timing delay* blok rangkaian

Tabel 3. Tabel *timing delay blok rangkaian*

Standard Cell		Delay(ns)					
		Beban	Tpdr	Tpdf	Tpd	Tr	Tf
Full	Sum	62,4ff	1.05	1.1	1.075	0.34	0.51
adder	Cout		0.45	0.35	0.41	0.19	0.16
2's	S=0	93,6ff	0.50	0.6	0.55	0.3	0.37
compl	S=1		0.8	1.05	0.925	0.3	0.4
	SLTU	31,2ff	0.55	0.55	0.55	0.15	0.17
	SLL	31,2ff	0.7	0.75	0.725	0.25	0.28
	SRL	31,2ff	0.7	0.75	0.725	0.25	0.28
	Zero detector	31,2ff	0.85	0.28	0.56	0.42	0.28

3.4. Arithmetic Logic Unit (ALU) 1-bit



Gambar 17. Hasil simulasi ALU 1-bit pada LT-spice

Tabel 4. Rangkuman pengujian ALU 1-bit

Pengujian	Perintah	Selektor mux	Input	Result
		(S2S1S0)	(CBA)	
1	Add	000	000	0
2	AND	001	001	0
3	SLL	010	010	0
4	XOR	011	011	0
5	Subb	100	100	1
6	OR	101	101	1
7	SRL	110	110	1
8	SLT	111	111	0

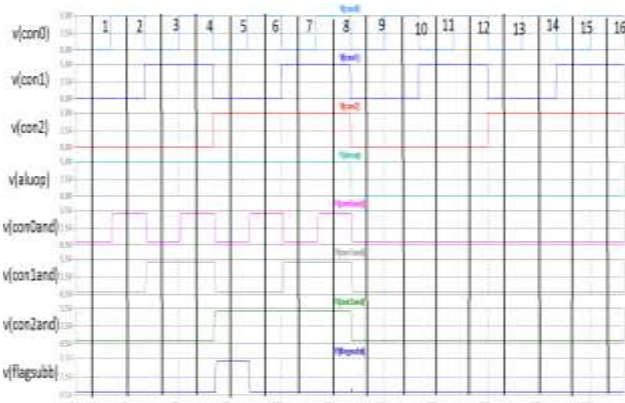
Tabel 5. Tabel Analisis area dan *delay* ALU 1-bit

ALU 1 - bit					
Area (lambda ²)	882 x 132				
Delay(ns)					
Beban	Tpdr	Tpdf	Tpd	Tr	Tf
Clload 62,4ff	3.4	1.9	2.65	0.14	0.1

Gambar 17 merupakan hasil pengujian ALU 1-bit, hasil rangkuman pengujian dapat dilihat pada tabel 4. Secara fungsional, hasil keluaran ALU 1-bit sudah sesuai dengan perancangan yang dilakukan. Tabel 5 adalah hasil pengukuran delay. Delay paling lama terjadi saat pengurangan, karena delay yang terjadi dari blok 2's complement dan blok full adder.

3.5. Kontrol ALU

Hasil simulasi pengujian kontrol ALU dapat dilihat pada gambar 18, sedangkan Tabel 6 merupakan gambar tabel hasil pengujian yang dilakukan sesuai gambar 18. Secara fungsional kontrol ALU sudah bekerja dengan benar, sesuai dengan perancangan yang telah dilakukan. Hasil pengukuran delay kontrol ALU dapat dilihat pada tabel 7. Delay yang diukur adalah pada keadaan *worst case*, yaitu saat flagsubb bernilai '1'.



Gambar 18. Hasil simulasi pengujian kontrol ALU pada LT-spice

Tabel 6. Rangkuman pengujian kontrol ALU

Pengujian	Sinyal kontrol		Output	
	Con2Con1 Con0	ALUop	Output (con0,con 1,con0)	Flagsu bb
1	000	1	000	0
2	001	1	001	0
3	010	1	010	0
4	011	1	011	0
5	100	1	100	1
6	101	1	101	0
7	110	1	110	0
8	111	1	111	0
9	000	0	000	0
10	001	0	000	0
11	010	0	000	0
12	011	0	000	0
13	100	0	000	0
14	101	0	000	0
15	110	0	000	0
16	111	0	000	0

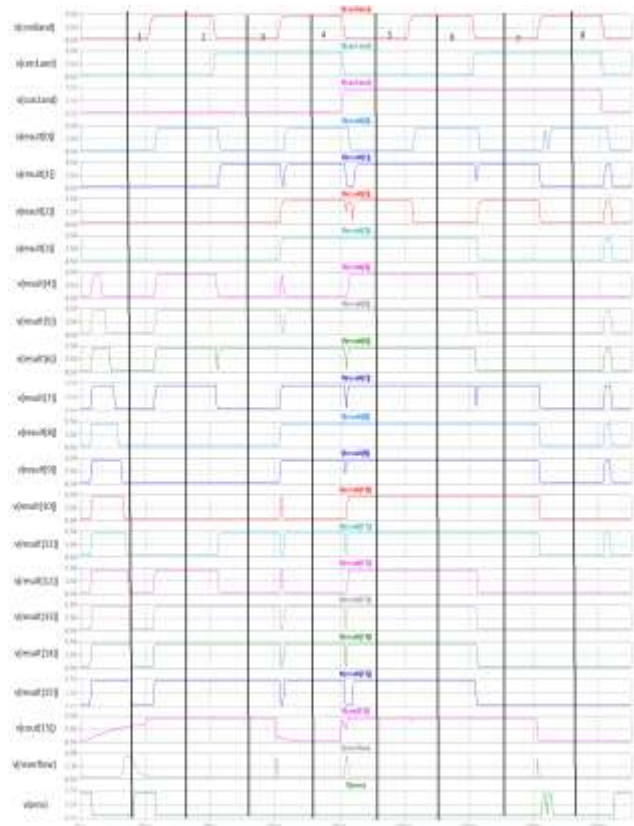
Tabel 7. Tabel Analisis area dan delay Kontrol ALU

ALU kontrol						
Area (lambda ²)						
323 x 124						
Delay(ns)						
Beban	Kontrol	Tpdr	Tpdf	Tpd	Tr	Tf
Cloud	Kontrol	1	1	1	0.36	0.27
1000fF	Flagsubb	1.4	1	1.2	0.36	0.27

Beban yang digunakan sebesar 1000fF. Beban ini disesuaikan dengan fanout dari blok kontrol ALU untuk mengontrol 32 multiplexer.

3.6. Arithmetic Logic Unit (ALU) 16-bit

Gambar 19 merupakan hasil simulasi ALU 16-bit pada LT-spice, dan tabel 8 adalah rangkuman hasil pengujian pada LT-spice. Dari hasil pengujian, dapat dilihat jika ALU 16-bit sudah mengeluarkan logika perhitungan dengan benar dan sesuai sinyal kontrol yang diberikan. Tabel 9 adalah hasil pengukuran delay pada ALU 16-bit. Terlihat jika delay yang terjadi sebesar 13ns. Dari nilai Tpdf dan Tpdr, dapat diambil kesimpulan jika hasil keluaran ALU valid setelah 16ns. Hal tersebut terjadi pada saat kondisi *worst case*, yaitu saat penjumlahan dengan nilai A=0001 B=FFFF serta pengurangan dengan nilai A=FFFF dan B=0001.



Gambar 19. Hasil pengujian ALU 16-bit pada LT-spice

3.7. Arithmetic Logic Unit (ALU) 16-bit Instruksi Bersama

Pengujian bersama ini menggunakan urutan instruksi yang disesuaikan dengan blok sistem pada prosesor RISC 16-bit lainnya. Hanya saja pengujian dilakukan secara terpisah oleh setiap bloknya, sehingga dianggap masukan untuk setiap blok dari keluaran blok lain dianggap benar. Berikut ini adalah urutan instruksi yang digunakan. Tabel 10 adalah urutan instruksi bersama yang dilakukan, hasil simulasi dan tabel rangkuman dapat dilihat pada gambar 20 dan tabel 11.

Tabel 8. Tabel hasil pengujian ALU 16-bit

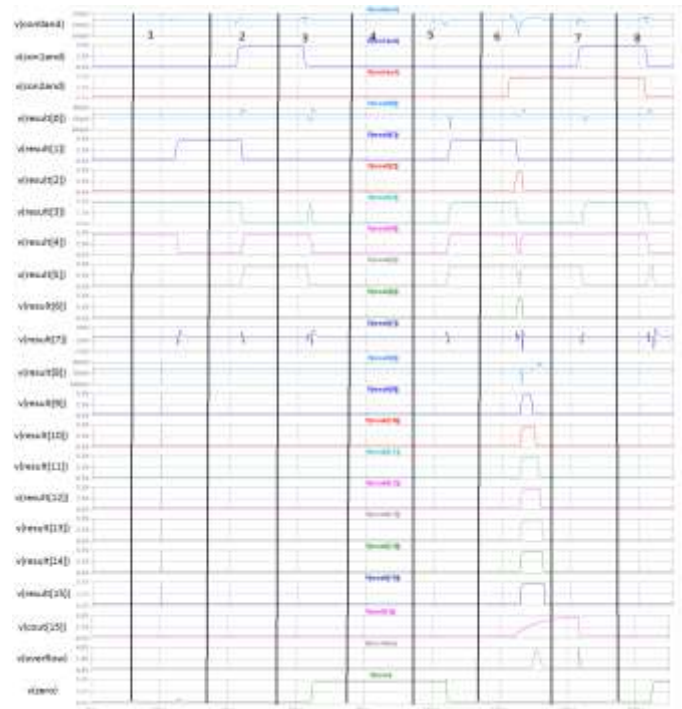
Kontrol	Input		Output			
			Result	Cout	Overflow	Zero
000	A	0001 ₁₆	0000 ₁₆	1	0	1
	B	FFFF ₁₆				
001	A	FFF9 ₁₆	F0F1 ₁₆	Don't care	0	0
	B	F0F3 ₁₆				
010	A	F431 ₁₆	E862 ₁₆	Don't care	0	0
	B	1F0D ₁₆				
011	A	0BEE ₁₆	EBEF ₁₆	Don't care	0	0
	B	E001 ₁₆				
100	A	1111 ₁₆	FFFE ₁₆	1	0	0
	B	0001 ₁₆				
101	A	F0F3 ₁₆	FFFB ₁₆	Don't care	0	0
	B	FFF9 ₁₆				
110	A	1F0D ₁₆	0F86 ₁₆	Don't care	0	0
	B	F431 ₁₆				
111	A	0001 ₁₆	0001 ₁₆	Don't care	0	0
	B	0BEE ₁₆				

Tabel 9. Tabel analisis area dan delay ALU 16-bit

ALU 16-bit						
Area (lambda ²)	1185 x 2542.5					
Delay						
Beban	Tpdr (ns)	Tpdf(ns)	Tpd(ns)	Tr(ns)	Tf(ns)	
Cload 62.4fF	10	16	13	0.22	0.15	

Tabel 10. Tabel instruksi bersama

No	Instruksi	Sinya masukan ALU	
1	ADDI R0, R1, 24 ₁₀	A	0000 ₁₆
		B	0018 ₁₆
		Aluop	0 ₂
		ALU kontrol	000 ₂
2	ADDI R0, R2, 10 ₁₀	A	0000 ₁₆
		B	000A ₁₆
		Aluop	0 ₂
		ALU kontrol	000 ₂
3	SLL R1, R0, R1	A	0018 ₁₆
		B	0000 ₁₆
		Aluop	1 ₂
		ALU kontrol	010 ₂
4	SW R0, R1, 000000	A	0000 ₁₆
		B	0000 ₁₆
		Aluop	0 ₂
		ALU kontrol	000 ₂
5	LW R0, R3, 000000	A	0000 ₁₆
		B	0000 ₁₆
		Aluop	0 ₂
		ALU kontrol	000 ₂
6	ADD R3, R2, R4	A	0030 ₁₆
		B	000A ₁₆
		Aluop	1 ₂
		ALU kontrol	000 ₂
7	SUBB R4, R2, R1	A	003A ₁₆
		B	000A ₁₆
		Aluop	1 ₂
		ALU kontrol	100 ₂
8	SRL R1, R0, R1	A	0030 ₁₆
		B	0000 ₁₆
		Aluop	1 ₂
		ALU kontrol	110 ₂



Gambar 20. Hasil pengujian ALU 16-bit instruksi bersama

Tabel 11. Tabel hasil pengujian ALU instruksi bersama

No	Sinyal kontrol	Input		Output			
				Result	Cout	Overflow	Zero
1	000 ₂	A	0000 ₁₆	0018 ₁₆	Don't care	0	
		B	0018 ₁₆				
2	000 ₂	A	0000 ₁₆	000A ₁₆	Don't care	0	
		B	000A ₁₆				
3	010 ₂	A	0018 ₁₆	0030 ₁₆	Don't care	0	
		B	0000 ₁₆				
4	000 ₂	A	0000 ₁₆	0000 ₁₆	Don't care	1	
		B	0000 ₁₆				
5	000 ₂	A	0000 ₁₆	0000 ₁₆	Don't care	1	
		B	0000 ₁₆				
6	000 ₂	A	0030 ₁₆	003A ₁₆	0	0	
		B	000A ₁₆				
7	100 ₂	A	003A ₁₆	0030 ₁₆	1	0	
		B	000A ₁₆				
8	110 ₂	A	0030 ₁₆	0018 ₁₆	Don't care	0	
		B	0000 ₁₆				

Dari hasil pengujian, dapat dilihat jika ALU 16-bit menghasilkan keluaran yang sesuai dengan perintah yang diberikan. Sehingga ALU 16-bit bekerja dengan benar dan sudah sesuai dengan perancangan ALU 16-bit Prosesor RISC 16-bit.

4. Kesimpulan

Telah dirancang *Arithmetic Logic Unit* (ALU) 16-bit menggunakan teknologi fabrikasi 600nm yang mampu melakukan 8 buah operasi, yaitu penjumlahan, pengurangan, AND, OR, XOR, SLL, SRL, dan SLTU sesuai dengan sinyal kontrol. Dari hasil pengujian didapatkan luas area layout menggunakan perangkat

lunak *Electric* sebesar $355.5 \times 762.75 \text{ um}^2$. *Time propagation delay* (Tpd) yang terjadi pada hasil keluaran ALU sebesar 13ns, sedangkan nilai *time rise* (Tr)=0,22ns dan *time fall* (Tf)=0,15ns. Oleh karena itu, ALU 16-bit dapat beroperasi maksimal pada clock sebesar 62.5MHz.

Referensi

- [1]. J. Enyeart, M. Al-Karawi, and R. Medina, "8-bit MIPS Processor," 2014.
- [2]. MOSIS, "C5 model." [Online]. Available: <http://www.onsemi.com/PowerSolutions/content.do?id=16693>.
- [3]. S. M. Rubin, "Electric (Software)," 2014. [Online]. Available: <http://cmosedu.com/cmos1/electric/electric.htm>. [Accessed: 20-Dec-2014].
- [4]. M. Engelhardt, "LTspice IV (software)." Linear Technology, 2014.
- [5]. D. A. Patterson and C. H. Sequin, "A VLSI RISC.," *IEEE Comput.*, vol. 15, no. 9, pp. 8–21, 1982.
- [6]. R. Jarwal and U. Khire, "4-Bit Arithmetic And Logic Unit Design Using Structural Modelling In VHDL," in *International Journal of Engineering Research and Technology*, 2013, vol. 2, no. 3 (March-2013).
- [7]. P. Gurjar, R. Solanki, P. Kansliwal, and M. Vucha, "VLSI implementation of adders for high speed ALU," in *India Conference (INDICON), 2011 Annual IEEE*, 2011, pp. 1–6.
- [8]. W. L. Pang and M. B. I. Reaz, "Performance evaluation of Manchester carry chain adder for VLSI designer library," in *Proceedings of the 5th WSEAS international conference on Simulation, modelling and optimization*, 2005, pp. 685–689.
- [9]. A.P.Paplinski, "MOS Transistor," 2002. [Online]. Available: <http://www.csse.monash.edu.au/courseware/cse3142/2006/Lnts/C02.pdf>. [Accessed: 25-Mar-2015].
- [10]. N. H.E.Weste and D. M. Harris, *CMOS VLSI DESIGN*. Boston: Addison-Wesley, 2011.