PERANCANGAN *MULTIPLIER* SEKUENSIAL 8-BIT DENGAN TEKNOLOGI 180NM MENGGUNAKAN PERANGKAT LUNAK ELECTRIC

Brama Yoga Satria[*)], Munawar Agus Riyadi, and Muhammad Arfan

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

[*)]*E-mail: bramayogasatria@gmail.com*

Abstrak

Very Large Scale Integration (VLSI) merupakan proses dari pembuatan sirkuit terpadu atau Integrated Circuit (IC) dengan cara menggabungkan ribuan rangkaian berbasis transistor ke dalam sebuah chip atau prosesor. Dengan adanya VLSI, ukuran dari devais elektronik berbasis transistor dapat dimampatkan agar menghemat area, biaya produksi, dan efek parasitik. Prosesor terdiri dari beberapa blok utama sebagai penunjang kerjanya, salah satu blok yang paling penting yaitu Arithmatic Logic Unit (ALU). Salah satu contoh dari ALU sendiri yaitu adalah multiplier. Multiplier sangat penting untuk banyak dasar proses dari sebuah prosesor. Tujuan dari penelitian ini adalah merancang sebuah multiplier sekuensial 8-bit dengan teknologi 180nm. Multiplier dirancang dengan menggabungkan blok-blok pembangun seperti blok counter, adder, shift register, dan lain-lainnya. Penelitian ini menggunakan perangkat lunak electric untuk mendesain layout dan perangkat lunak LT-Spice untuk menguji fungsional, delay, dan kinerja dari hasil ekstraksi layout. Hasil perancangan ini secara fungsional telah berjalan dengan baik. Multiplier yang dirancang memiliki layout sebesar 3.725.150 lambda^2 dengan nilai delay sebesar 4,428ns. Selain itu, frekuensi maksimum yang digunakan untuk mendapatkan hasil yang benar dari multiplier sekuensial 8-bit yaitu 50MHz.

Kata kunci: ALU, Multiplier, Multiplier Sekuensial

Abstract

VLSI is a process of making integrated circuit or ICs by combining thousands of transistor based circuits into a chip. In the presence of VLSI, the size of transistor based electronic device can be compressed to save area, production costs, and parasitic effects. Processor consists of several main blocks for their work, one of the most important blocks is arithmetic logic units (ALUs). One example of the ALU is a multiplier. Multiplier is very important for many basic processes of a processor. The purpose of this research is to design an 8-bit sequential multiplier with 180nm technology. Multiplier is designed with block building blocks such as counter, adder, shift register, and others. This research uses electric software to design the layout and LT-Spice software to test the functional, delay, and performance of layout extraction results. The design results are functional has been running well. Multiplier designed to have layout of 3.725.150 with delay value equal to 5,375ns and produce performance equal to 20.022.681,25. In addition, the maximum frequency used to obtain the correct result of the 8-bit sequential multiplier is 50MHz.

Keywords: ALU, Multiplier, Sequential Multiplier

1. Pendahuluan

Very Large Scale Integration (VLSI) merupakan proses dari pembuatan sirkuit terpadu atau *Integrated Circuit* (IC) dengan cara menggabungkan ribuan rangkaian berbasis transistor ke dalam sebuah *chip*. Proses ini memungkinkan untuk memasukkan beberapa fungsi dari prosesor seperti *Central Prosessing Unit* (CPU), memori, dan input atau output ke dalam sebuah *chip.* Dengan adanya VLSI, ukuran dari devais elektronik berbasis transistor dapat dimampatkan agar menghemat area, biaya produksi, dan efek parasitik. Salah satu hasil aplikasi yang diterapkan dengan proses VLSI adalah prosesor. Prosesor terdiri dari beberapa blok utama sebagai penunjang kerjanya, salah satu blok yang paling penting yaitu *Arithmatic Logic Unit* (ALU). Blok ini melakukan proses perhitungan seperti penjumlahan, pengurangan, perkalian, pembagian, penggeseran bit dan operasi logika seperti AND, OR, dan lain-lain.

Multiplier merupakan salah satu perangkat keras yang penting dalam komputasi *Digital Signal Processing* (DSP). DSP yang umum diimplementasikan yaitu dimana multiplier berperan penting dalam proses penting seperti

filter digital, komunikasi digital, dan analisis spektral. Banyak DSP sekarang ini menargetkan implementasinya harus bersifat *portable*, *battery-operated systems*, maka dari itu disipasi daya menjadi salah satu kendala utama dalam desain. Penelitian sebelumnya tentang *multiplier* yang dilakukan oleh Vinod Sajjan, B. Sajidha Thabassum, dan Mala Sinnoor dengan judul "Implementation of 4-bit Unsigned Array Multiplier and Baugh Wooley Multiplier using CMOS 180nm Technology and Their Comparative analysis"[1]. *Multiplier* yang dibuat pada penelitian ini memiliki sifat kombinasional dan memiliki lebar data 4-bit menggunakan teknologi 180 nm. Lalu penelitian berjudul "Design of Optimized Wallace Tree Multiplier in Cadence"[2] yang membahas bagaimana merancang *multiplier* 5-bit dengan *software* cadence. Penelitian lain berjudul "8-by-8 Bit Shift/Add Mutiplier" oleh Giovanni D'Aliesio membahas tentang bagaimana merancang *multiplier* 8-bit dengan metode *shift and add* [3]. Kemampatan area devais elektronik merupakan hal yang menarik bagi seorang desainer VLSI. Dituntut untuk dapat merancang banyak sistem elektronik pada substrat yang areanya kecil dengan menggunakan aturan yang berlaku, desainer harus berhadapan dengan masalah-masalah seperti *delay*, area, dan kinerja. Maka dari itu terdapat sebuah tuntutan dimana desainer harus dapat mengoptimalkan area dari substrat yang tersedia. Selain itu, *multiplier* kombinasional dengan lebar data 4-bit dianggap kurang efisien dibanding dengan *multiplier* sekuensial dengan lebar data 8-bit.

Berdasarkan latar belakang tersebut, Penelitian ini mengambil judul "Perancangan Multiplier Sekuensial 8-bit dengan Teknologi 180nm Menggunakan Perangkat Lunak Electric". Perancangan *multiplier* menggunakan teknologi 180nm dengan tegangan sumber sebesar 1,8V serta lebar data 8-bit. Perangkat lunak Electric digunakan untuk membuat desain *layout* multiplier, dan perangkat lunak LT-Spice untuk menguji fungsional, *delay*, dan kinerja dari hasil ekstraksi *layout* multiplier yang dibuat.

2. Metode

2.1. Perancangan *Standard Cell*

Standard cell merupakan suatu standar dimana setiap *cell* mempunyai tinggi dan lebar yang seragam [4][5]. Multiplier sekuensial yang dibuat memiliki area sebesar 3.725.150 lambda^2, akan tetapi area tiap-tiap blok pembangun dari multiplier memiliki nilai yang berbeda-beda. Setiap *standard cell* memiliki jumlah masukan dan fungsi yang berbeda – beda, sehingga koneksi poly-ke-poly dan metal-ke-metal diantara pMOS dan nMOS pada sebuah *standar cell* berbeda - beda. Oleh karena itu, ukuran W transistor tiap *standard cell* berbeda. Hal ini disebabkan untuk menyesuaikan tinggi *standard cell* yang tetap sama dan tetap menjaga agar *time rise* dan *time fall* seimbang [6].

2.2. Alur Logika *Multiplier*

Desain *multiplier* memiliki jalur data yang memetakan proses jalannya data masukan antar blok sampai menjadi hasil proses multiplikasi. Jalannya data di dalam *multiplier* dapat dijelaskan dalam beberapa tahap antara lain:

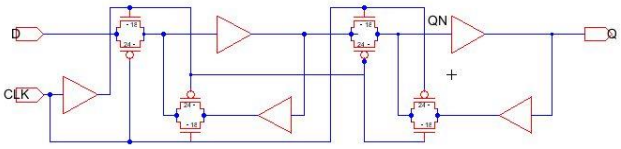
1. Saat *clock initial* terjadi, kondisi rangkaian belum melakukan proses apapun.
2. Pada saat hitungan awal *clock* pertama terjadi, masukan data blok *multiplier* yaitu angka multiplikasi dan blok *process* yang memiliki masukan data awal 0000000000000000_2 akan diproses lalu keluarannya pergi menuju blok *adder*.
3. Keluaran dari kedua blok *multiplier* dan *result* akan diproses oleh blok *adder* lalu dikirim ke blok *result*.
4. Bit terakhir dari 8-bit data blok *multiplicand* akan menjadi penentu hasil keluaran dari blok *result*. Apabila bit terakhir blok *multiplicand* bernilai 0 maka blok *result* akan meneruskan data dari blok *process*, sedangkan apabila bit terakhir blok multiplicand bernilai 1 makan blok *result* akan meneruskan data dari blok blok *adder*.
5. Data masukan yang ada di dalam blok *process* akan berubah menjadi keluaran dari blok *result*. Keluaran inilah merupakan hasil kali data blok *multiplier* dan bit terakhir dari blok *multiplicand*.
6. Saat counter untuk *clock* kedua muncul, Data dari blok *multiplier* dan *multiplicand* mengalami pergeseran ke kiri sebanyak 1-bit.
7. Proses seperti pada poin 2, 3, dan 4 akan terjadi.
8. Ketika *counter* kembali berjalan menjadi *clock* ketiga, data dari blok *multiplier* dan *multiplicand* kembali mengalami pergeseran ke kiri sebanyak 1-bit. Lalu proses pada poin 2, 3, dan 4 akan terjadi kembali.
9. Proses pada poin 7 akan selalu berulang sampai *counter* mencapai *clock* kedelapan.
10. *Multiplier* akan berhenti pada nilai *counter clock* kedelapan dan menampilkan hasil akhir dari proses multiplikasi.

2.3. Perancangan *Multiplier*

Setelah mendesain *standard cell* gerbang logika dari *level* transistor dan *layout* dilanjutkan dengan perancangan *multiplier. Multiplier ini memiliki beberapa blok yang terdiri dari beberapa rangkaian logika. Blok-blok ini juga memiliki spesifikasi dan fungsi tertentu.*

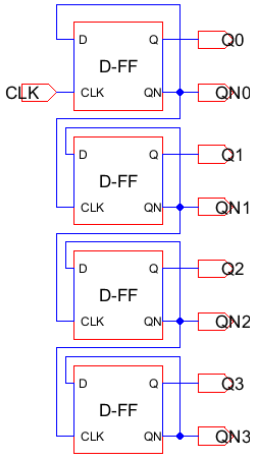
2.3.1. Blok *Counter*

Blok *counter* merupakan blok yang menjadi fungsi penghitung dari angka nol sampai delapan. Karena rangkaian multiplier ini bekerja membutuhkan delapan sinyal *clock*, maka dibutuhkan sebuah penghitung (*counter*) dari angka nol sampai delapan. Disusun dengan menggunakan empat rangkaian D flip-flop, satu rangkaian *tri-state buffer*, dan satu rangkaian gerbang AND.



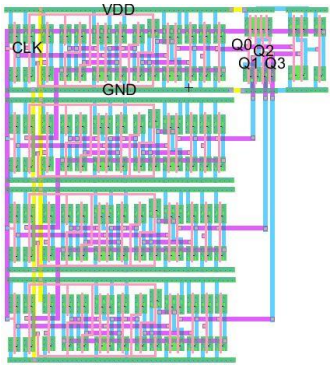
Gambar 1. Skematik DFF 1-bit

Gambar 1 merupakan rangkaian D flip-flop 1-bit dengan masukan D dan CLK serta memiliki keluaran Q dan QN. Setelah itu DFF 1-bit ini akan dirangkai menjadi 4-bit seperti Gambar 2.



Gambar 2. Skematik DFF 4-bit

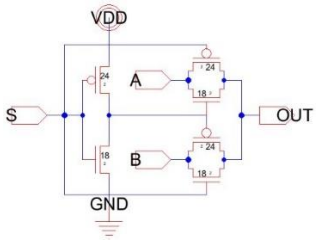
Setelah merancang skematik kemudian dibawa ke perancangan *layout*. Gambar 3 adalah *layout* dari blok *counter*.



Gambar 3. *Layout* blok *counter*

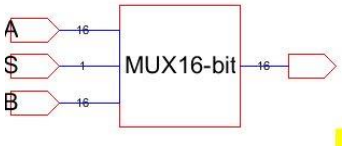
2.3.2. Blok *Process*

Blok *process*, merupakan blok penyimpanan data hasil *dot product*. Blok *process* ini adalah sebuah *register* yang memiliki lebar data 16-bit. Blok ini memiliki masukan awal 0000000000000000_2, setelah itu masukannya dapat berubah. Blok ini disusun oleh multiplekser 2-ke-1 sebanyak 16-bit. Gambar 4 merupakan skematik dari multiplekser 2-ke-1 1-bit.



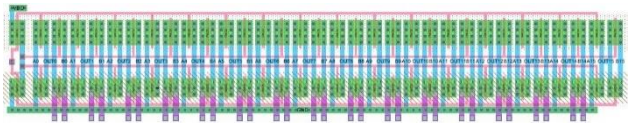
Gambar 4. Skematik Multiplekser 2-ke-1 1-bit

Setelah itu multiplekser 2-ke-1 1-bit akan dirangkai menjadi multiplekser 16-bit seperti pada Gambar 5.



Gambar 5. Skematik Multiplekser 2-ke-1 16-bit

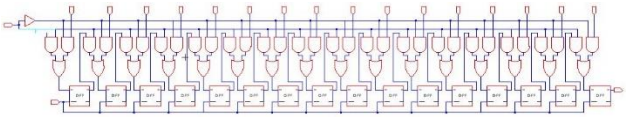
Setelah merancang skematik kemudian dibawa ke*level layout*. Gambar 6 merupakan *layout* dari blok process yang terdiri dari multiplekser 2-ke-1 16-bit.



Gambar 6. *Layout* blok *process*

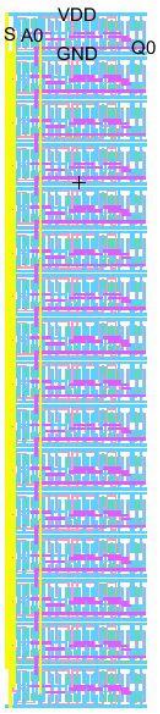
2.3.3. Blok *Multiplier*

Blok *multiplier*, merupakan blok masukan dari angka multiplikasi. Data masukan nantinya akan memiliki lebar data 16-bit dengan format "00000000" A, A sebagai masukan data 8-bit. Blok ini merupakan *shift register* yang dirancang dengan merangkai enam belas D-flip-flop. Gambar 7 merupakan skematik dari *shift register* 16-bit.



Gambar 7. Skematik *shift register* 16-bit

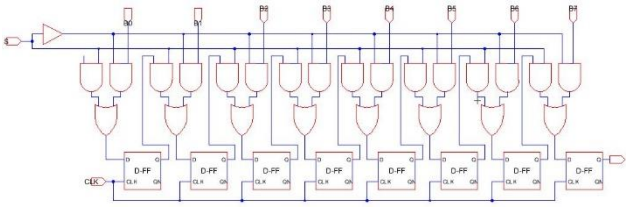
Setelah perancangan level skematik kemudian dibawa ke level *layout.* Gambar 8 merupakan *layout* dari *shift register* 16-bit.



Gmbar 8. *Layout shift register* 16-bit

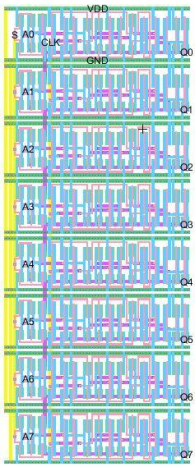
2.3.4. Blok *Multiplicand*

Blok *multiplicand*, merupakan blok masukan dari angka pengali multiplikasi B. Data masukan nantinya akan memiliki lebar data 8-bit. Blok ini dirancang sama persis dengan blok *multiplier*. Gambar 9 merupakan skematik dari *shift register* 8-bit.



Gambar 9. Skematik *shift register* 8-bit

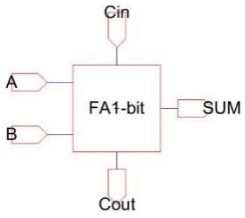
Setelah perancangan level skematik kemudian dibawa ke level *layout.* Gambar 10 merupakan *layout* dari *shift register* 8-bit.



Gambar 10. *Layout shift register* 8-bit

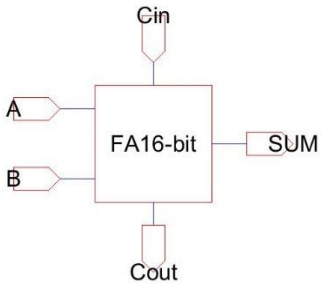
2.3.5. Blok *Adder*

Blok *adder*, merupakan blok yang akan memproses dua masukan dari blok *multiplier* dan blok *process*. Disusun dengan merangkai *full adder* dengan topologi RCA sebanyak 16-bit. Gambar 11 merupakan skematik dari full adder 1-bit.



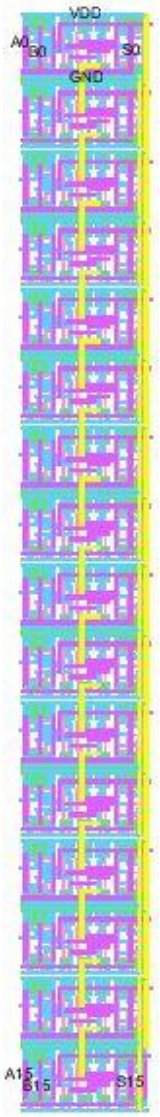
Gambar 11. Skematik *full adder* 1-bit

Setelah itu *full adder* 1-bit akan dirangkai menjadi *full adder* 16-bit seperti pada Gambar 12.



Gambar 12. Skematik *full adder* 16-bit

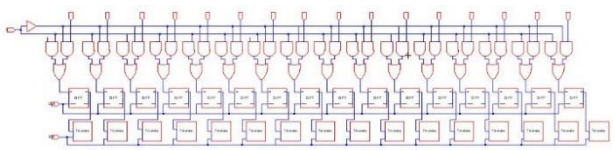
Setelah merancang skematik kemudian dibawa ke*level layout*. Gambar 13 merupakan *layout* dari blok adder yang terdiri dari *full adder* 16-bit.



Gambar 13. *Layout full adder* 16-bit

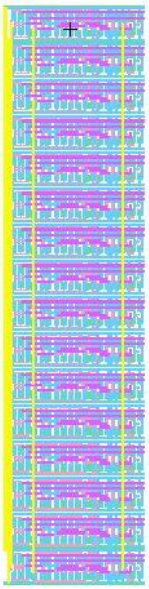
2.3.6. Blok *Result*

Blok *result*, merupakan blok *register* yang menjadi tempat penyimpan hasil keluaran blok *adder*. Disusun oleh enam belas D flip-flop dan *tri-state buffer*. Gambar 14 merupakan skematik dari blok *result.*



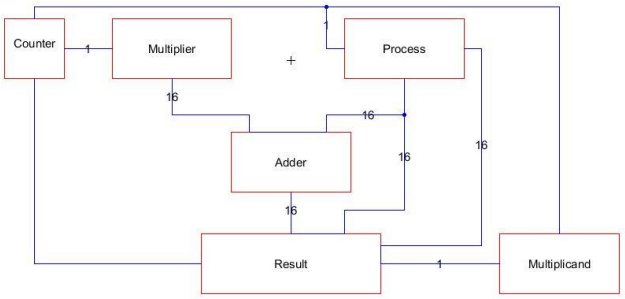
Gambar 14. Skematik blok *result*

Setelah merancang skematik kemudian dibawa ke*level layout*. Gambar 15 merupakan *layout* dari blok *result* yang terdiri dari *register* 16-bit beserta *tri-state buffer.*



Gambar 15. *Layout blok result*

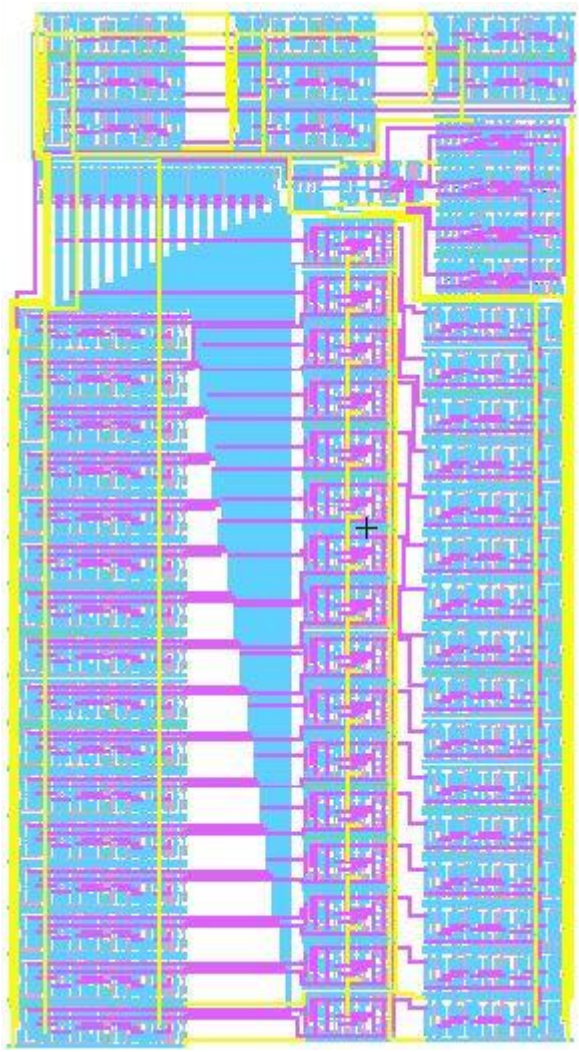
2.3.6. Multiplier



Gambar 16. Skematik *multiplier* sekuensial 8-bit

Gambar 16 merupakan desain skematik *multiplier* skuensial 8-bit. Rangkaian ini menggabung blok-blok fungsi sebelumnya antara lain *adder*, *counter*, *multiplier*, *multiplicand*, *process*, dan *result*. Masukan nilai multiplikasi dapat dimasukkan pada blok multiplier dan *multiplicand* sebanyak 8-bit. Setelah itu, hasil dari multiplikasi dapat dilihat pada blok *result* sebanyak 16-bit.

Setelah perancangan dari *level* skematik di atas, kemudian dibawa ke *level layout*. Berikut hasil desain *layout* dari *multiplier* sekuensial 8-bit



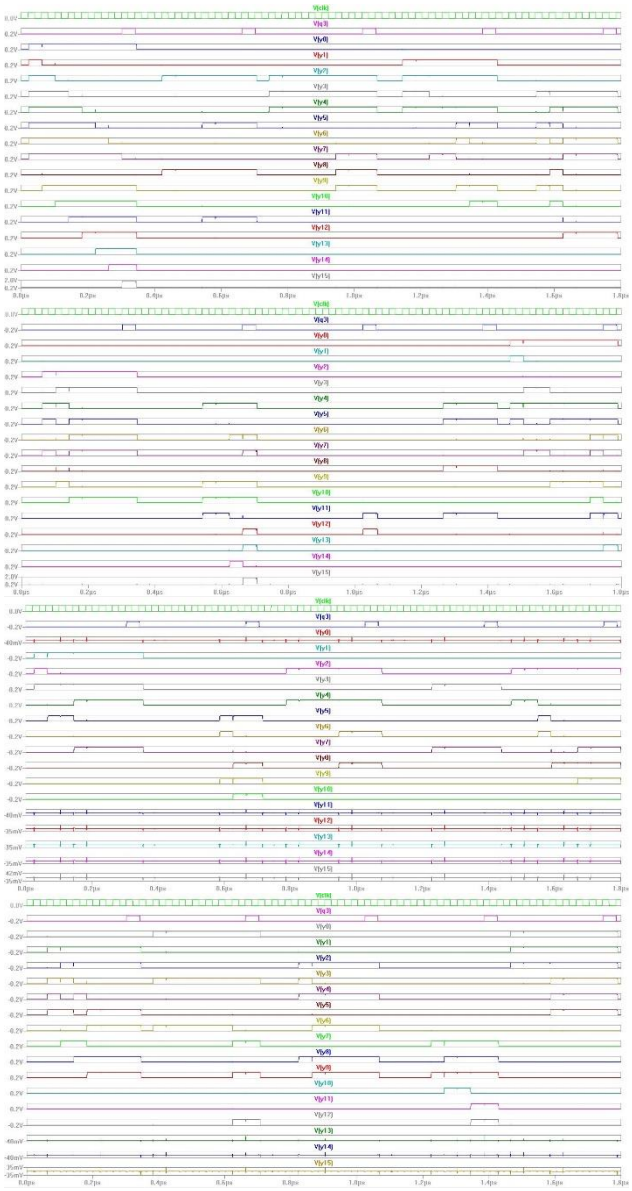
Gambar 17. *Layout multiplier* sekuensial 8-bit

3. Hasil dan Analisa

Setelah pembuatan *layout* selesai dilakukan, tahap berikutnya adalah pemberian *node* sinyal masukan, keluaran, dan *power* untuk menghasilkan variasi *input*. Setelah itu *layout* diekstrak pada perangkat lunak LT-spice untuk dilakukan pengujian fungsional, pengukuran area dan pengukuran *delay*.

3.1. Pengujian Fungsional

Gambar 17 adalah hasil simulasi keluaran *layout multiplier* sekuensial 8-bit. Pengujian dilakukan sebanyak empat kali, dimana satu kali pengujian memiliki lima hasil keluaran. Gelombang hasil keluaran dapat dilihat pada Gambar 18.



Gambar 18. Hasil Pengujian fungsional *multiplier* sekuensial 8-bit

Dapat dilihat gelombang keluaran hasil multiplikasi berada saat *node* Q3 berada di tegangan 1.8V. Pengamatan hasil multiplikasi diaamati dari *node* Y0 sampai Y15. Contohnya saat pengujian pertama masukan bernilai "11111111" (255) untuk *multiplier dan multiplicand.* Hasil dari multiplikasi kedua masukan tersebut yaitu "1111111000000001" (65025). Maka dari itu fungsional *multiplier* sekuensial 8-bit telah berjalan dengan baik. Data pengujian lain dari multiplikasi dapat dilihat pada Tabel 1.

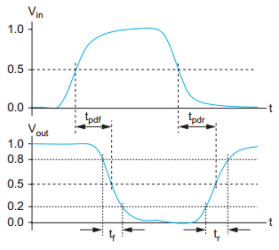
Tabel 1. Pengujian multiplier sekuensial 8-bit

Pengujian	*Multiplier*	*Multiplicand*	Hasil
1	11111111	11111111	1111111000000001
	10000010	00010010	0000100100100100
	00011100	00100001	0000001110011100
	00001111	01101010	0000011000110110
	10011010	00011100	0001000011011000
2	01011010	00001110	0000010011101100
	11100001	11010000	1011011011010000
	00110000	10000000	0001100000000000
	10010011	00010000	0000100100110000
	00110011	11001011	0010100001110001
3	00001110	00001011	0000000010011010
	00010011	01100000	0000011100100000
	00001010	00100010	0000000101010100
	00010001	00001000	0000000010001000
	00010100	00101101	0000001110000100
4	00011101	00011110	0000001101100110
	01001001	01000001	0001001010001001
	01000111	00001100	0000001101010100
	01010000	01011000	0001101110000000
	00000111	00001001	0000000000111111

Dapat diamaati dari semua data pengujian *multiplier*, terlihat semua keluaran multiplikasi sudah sesuai dengan masukan yang diberikan. Sehingga *multiplier* sekuensial 8-bit bekerja dengan benar dan sesuai dengan perancangannya.

3.2. Pengujian Delay

Ketika suatu masukan berubah dan keluaran mempertahankan keadaan lama nya baru setelah ebberapa saat berubah ke keadaan yang baru, inilah yang dinamakan sebuah *delay*. *Delay* ini bisa terjadi karena efek kapasitansi yang terdapat pada gerbang dan masukan sebuah divais, dan jalur koneksi/*wire*. Kecepatan operasi gerbang digital dapat diukur melalui tiga buah parameter, yaitu *rise time* (Tr/waktu naik), *fall time* (waktu turun), dan *propagation delay* [4]. *Propagation delay* ditunjukkan pada Gambar 19.



Gambar 19. *Timing delay* : tpdf,tphr, tr dan tf [4]

- Tpdr = waktu yang dibutuhkan dari *input* Vdd/2 sampai *rising output* VDD/2
- Tpdf = waktu yang dibutuhkan dari *input* Vdd/2 sampai falling *output* VDD/2
- Tpd = rata – rata propagasi *delay* (Tpdr = Tpdf)/2
- Tr = Waktu yang dibutuhkan gelombang untuk naik dari 20% sampai 80% dari keadaan *steady – state*
- Tf = waktu yang dibutuhkan gelombang untuk turun dari 80% sampai 20% dari keadaan *steady - state*

Hasil perhitungan *delay multiplier* sekuensial 8-bit dapat lihat pada Tabel 2.

Tabel 2. Pengujian delay multiplier sekuensial 8-bit

Area (lambda^2)	*Delay*				
	Tr (ns)	Tf (ns)	Tpdr (ns)	Tpdf (ns)	Tpd (ns)
3.725.150	1,783	1,883	6,845	15,743	10,883

Data nilai yang diambil merupakan nilai dalam keadaan *worst case*, yaitu saat keluaran *node* hasil yang terakhir. Dapat dilihat bahwa area dari *layout multiplier* memiliki nilai sebesar 3.725.150 lambda^2. Dengan nilai tersebut *multiplier* menghasilkan nilai *time rise* (tr) sebesar 1,783 ns, *time fall* (tf) sebesar 1,883ns, dan *time propagation delay* (tpd) sebesar 10,883 ns.

Nilai *multiplier* baru mulai mengeluarkan hasil yang benar ketika diberi masukan *clock* sebesar 20ns. Sehingga keluaran *multiplier* baru dapat dikatakan benar setelah 20ns. Oleh karena itu, dapat diperoleh frekuensi maksimum dimana multiplier dapat mengeluarkan hasil operasi dengan benar, yaitu f =1/20ns = 50MHz.

4. Kesimpulan

Multiplier sekuensial 8-bit telah berjalan dengan benar sesuai dengan perancangannya. Dari data yang diambil dalam kondisi *worst case* didapatkan nilai *time propagation delay* (tpd) sebesar 10,883 ns. Selain itu *multiplier* memiliki frekuensi maksimum dimana multiplier dapat mengeluarkan hasil operasi dengan benar, yaitu f = 1/20ns = 50MHz. Untuk penelitian selanjutnya dapat lebih mengoptimalkan desain *multiplier* untuk memperkecil dan dapat menambahkan lebar data multiplikasinya. Selain itu dalam desain *layout*nya dapat menggunakan teknologi yang lebih kecil dari 180nm.

Referensi

[1]. Vinod S., B. Sajidha T., Mala S., "Design and Implementation of 4-Bit Unsigned Array Multiplier and Baugh Wooley Multiplier using CMOS 180nm Technology and Their Comparative Analysis," vol. 4, pp. 276–282, 2015.

[2]. A. Dash, S. Dash, S.K. Mandal, "Design of Optimized Wallace Tree Multiplier," pp. 0975–8887, 2014.

[3]. D'Aliesio G., "8-by-8 Bit Shift/Add Multiplier," 2003.

[4]. N. E. H. Weste and D. M. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, vol. 53, no. 9. 2013.

[5]. R. J. Tocci, *Digital Systems: principles and applications*. Pearson Education India, 1980.

[6]. R. J. Baker, *CMOS: circuit design, layout, and simulation*, vol. 18. John Wiley & Sons, 2011.