PERANCANGAN *LAYOUT* VLSI UNTUK ARSITEKTUR SET INSTRUKSI PADA PROSESOR MULTIMEDIA

VLSI LAYOUT DESIGN FOR INSTRUCTION SET ARCHITECTURE (ISA) ON MULTIMEDIA PROCESSOR

[1]Muhammad Nur Ramadhan Alam, [2]Agung Nugroho Jati, [3]Fairuz Azmi

[1,2,3] Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

[1]madhon@student.telkomuniversity.ac.id, [2]agungnj@telkomuniversity.ac.id, [3]worldliner@telkomuniversity.ac.id

Abstrak

Pada tugas akhir ini telah dirancang sebuah layout VLSI (very large scale IC) untuk sekumpulan set instruksi yang berguna untuk proses pengambilan data audio. Set instruksi yang dirancang memiliki panjang 16 bit. Data audio yang diolah disimpan di ROM. Layout direalisasikan menggunakan perangkat lunak Electric VLSI. Layout yang dirancang mulai dari gerbang – gerbang dasar hingga layout untuk komponen – komponen yang terlibat dalam proses pengambilan data. Layout dirancang menggunakan teknologi 300nm. Dari penggunaan teknologi 300nm dihasilkan ukuran die size untuk setiap komponen dengan satuan ukuran mm^2 (millimeter x millimeter)^2. Ukuran die size yang dihasilkan bervariasi, sesuai dengan nilai lamda $(\lambda)^2$ yang didapat dari pengukuran layout untuk masing – masing komponen. Clock frekuensi yang dibutuhkan prosesor multimedia yang dirancang dari interval 44.1KHz sampaI dengan frekuensi maksimum 250MHz.

Kata kunci : ***Layout VLSI, CMOS, ASIC, ISA, DSP, Prosesorr Multimedia.***

Abstract

In this final project has designed a VLSI (Very Large Scale IC) layout for a set of instruction that useful for fetching audio data. The length of instruction set design has 16 bits. The audio data that processed is stored in ROM. Layout realized using Electric VLSI. Layout designed from the basic gates to layouts for the components involved in fetching data process. Layout designed using 300nm technology. From the use of 300nm technology generated the size of die size for each component with unit size mm^2 (millimeter x millimeter)^2. The size of the resulting die size are different from one to another, according to the lamda $(\lambda)^2$ values obtained from the measurement layout for each component. Clock frequency required for the multimedia processor developed from the 44.1KHz interval up to the maximum frequency 250MHz.

Keywords: ***VLSI Layout Design, CMOS, ASIC, ISA, DSP, Multimedia Processor.***

1. Pendahuluan

Digital Signal Processor (DSP) merupakan satu jenis prosesor dari sekian banyak prosesor yang mengimplementasikan *Harvard Architecture*, yang berkembang dan dikembangkan secara terus menerus. DSP juga merupakan mikroprosesor jenis khusus dengan arsitektur yang dioptimalkan untuk *signal processing*[1]. Telah dibuktikan dengan akselerasi hardware yang dijadikan strategi dari implementasi untuk (DSP) dan *multimedia processing*. Struktur dari unit komputasi di dalam DSP dirancang khusus untuk memungkinkan kinerja tinggi dari suatu operasi secara fleksibel[1]. DSP juga mempunyai beberapa tipe yang beragam dan juga keunggulannya dikembangkan sesuai dengan kebutuhan, terutama dalam pengimplementasiannya secara nyata seperti pengolahan musik[2]. Prosesor Multimedia merupakan salah satu DSP yang menjadi perhatian.

Multimedia processing yang ada sampai saat ini telah berkembang cepat, terutama dalam proses kecepatan memproses suatu operasi atau instruksi, dikarenakan telah dilakukan penambahan instruksi khusus pada arsitektur set instruksi nya, terutama pada mikroprosesor *modern*[3] yang berkembang saat ini. Sebuah set instruksi yang dirancang memungkinkan untuk menulis setiap operasi yang lebih kompleks dari instruksi – instruksi dasar[4].

Instruction Set Architecture (ISA) adalah sebuah antarmuka metafisik antara sistem tingkat rendah dari mesin dan perangkat keras yang berisi semua informasi tentang mesin, yang diperlukan untuk menulis sebuah program untuk mesin[5]. ISA didefinisikan juga sebagai arsitektur yang menjelaskan proses kerja dari instruksi yang ada pada suatu prosesor, dari cara prosesor mengambil data untuk diolah dan menghasilkan suatu keluaran sampai menjelaskan tentang ukuran dari suatu prosesor[5]. Prosesor Multimedia juga membutuhkan ISA yang dapat mendukung *multimedia processing*. *Mixing audio* adalah salah satu dari sekian banyak *multimedia*

processing yang membutuhkan ISA yang cocok dan mendukung untuk melakukan proses *mixing* pada *audio*. *Mixing* merupakan proses dimana sinyal audio dari modul input dijumlahkan dan kemudian diolah oleh *equalizer*[6].

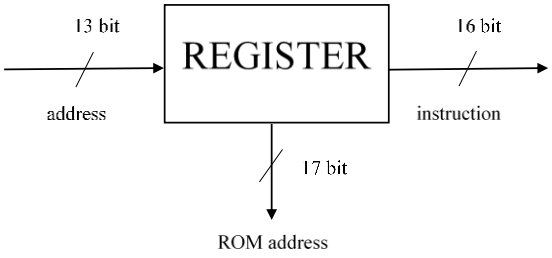
Very Large Scale IC (VLSI) merupakan proses pembuatan IC dengan menggabungkan ribuan bahkan jutaan sirkuit transistor kedalam satu chip kecil. VLSI dikenal juga sebagai teknik miniaturisasi sebuah *microchip* yang berisi jutaan transistor[7]. Semua transistor harus ditempatkan dengan benar dan terhubung sehingga seluruh rangkaian dapat beroperasi pada frekuensi tinggi secara selaras. Konsumsi daya pada suatu sirkuit perlu dikurangi[8]. VLSI desain terkait erat dengan proses fabrikasi. VLSI dapat diimplementasikan untuk setiap *microchip* atau mikroprosesor, karena hal tersebut, Prosesor Multimedia juga dapat diterapkan dalam bentuk layout VLSI, khususnya Prosesor Multimedia yang berkemampuan untuk *mixing audio*.

2. Material dan Perancangan

2.1. Gambaran Umum Sistem

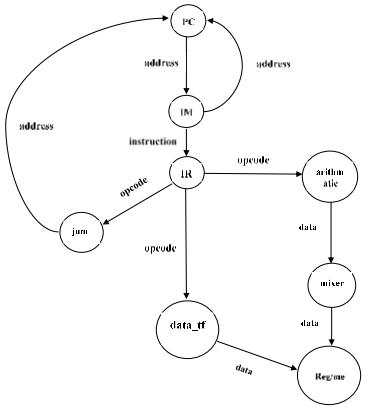
Perancangan layout VLSI untuk ISA 16 bit pada Prosesor Multimedia ini merupakan penelitian Tugas Akhir yang akan direalisasikan menggunakan perangkat Electric VLSI yang nanti nya akan disimulasikan dengan bantuan *FPGA Board Altera Cyclone II* yang bertujuan untuk menggabungkan atau mencampurkan 2 buah audio yang berbeda menjadi sebuah audio utuh.

Dalam prosesnya, data harus diambil untuk diolah serta sekumpulan instruksi yang harus dijalankan. Proses pengambilan data dan instruksi ini akan digambarkan pada diagram blok berikut :



Gambar 1. Diagram Blok fetching

Dari Gambar 1 kemudian diperjelas dengan penjelasan dari FSM (Finite State Machine) guna mengetahui proses yang dilalui pada proses *fetching*. Berikut merupakan FSM untuk proses tersebut :



Gambar 2. Diagram Transisi

Pada Gambar 2 melibatkan 3 komponen dalam proses pengambilan data dan instruksi *(fetching)*. Ketiga komponen tersebut adalah:

a) Program counter
b) Instruction Memory
c) Instruction Register

2.2. Rancangan Set Instruksi

Berikut ini adalah format ISA yang akan digunakan:

Tabel 1. Tabel Format Set Instruksi (ISA)

Load / Store Architecture

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Rc				Address								

Arithmetic Architecture

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Rc				Ra				Rb				0

Jump / Move Architecture

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Address												

Dari format ISA diatas, operasi – operasi yang akan diolah adalah sebagai berikut :

Tabel 2. Tabel Operasi Set Instruksi (ISA)

Jump

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Address												
000			0000000001010												

Load audio A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Rc				Address								
001			0000				000000000								

Load audio B

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Rc				Address								
010			0000				000000000								

Store header

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Rc				Address								
011			0000				000000000								

Store data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Rc				Address								
100			0000				000000000								

Mix audio A dan B

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Rc				Ra				Rb				0
101			0000				0000				0000				0

Play audio A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Address												
110			0000000000000												

Play audio B

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode			Address												
111			0000000000000												

Dari Tabel 1. Dan Tabel 2. Akan didapati deskripsi fungsi-fungsi dari setiap operasi ISA sebagaimana dijelaskan pada Tabel 3. Berikut ini.

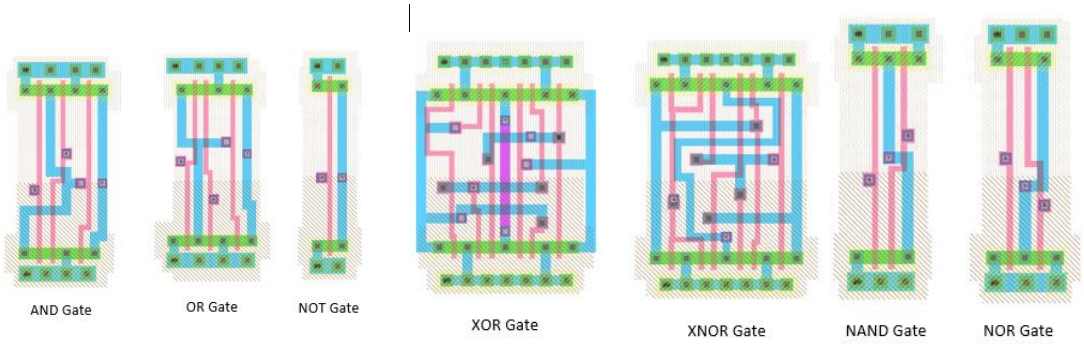
Tabel 3. Deskripsi dari set instruksi

Nama Operasi	**Fungsi**
Store Header	Memasukkan data Header dari audio ke SRAM
Play Audio A	Memainkan audio A
Play Audio B	Memainkan Audio B
Store data audio	mengambil data audio untuk di masukkan ke SRAM
Load data A	Memasukkan data dari ROM A ke register
Load data B	Memasukkan data dari ROM B ke register
Mix	Mencampurkan audio A dengan audio B

2.3. Rancangan Gate

2.3.1 Gerbang dasar

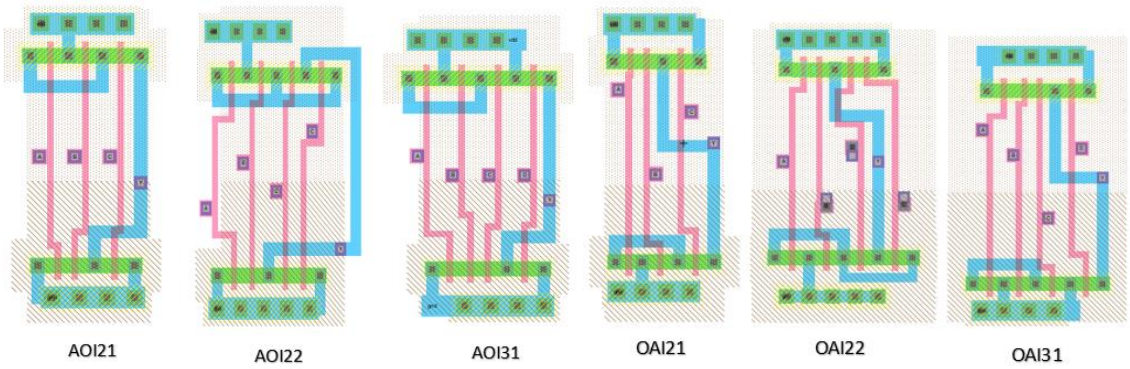
Pada logika transistor, yang termasuk ke dalam gerbang dasar ialah gerbang NAND, NOR dan juga NOT. Dari ketiga gerbang dasar tersebut, dapat dikembangkan lagi menjadi gerbang – gerbang lain nya, seperti AND dan NOT. Untuk perancangan layout dari gerbang – gerbang yang telah disebutkan, terlihat pada Gambar 6.



Gambar 6. Layout untuk Gerbang Dasar

2.3.2 Compound gate

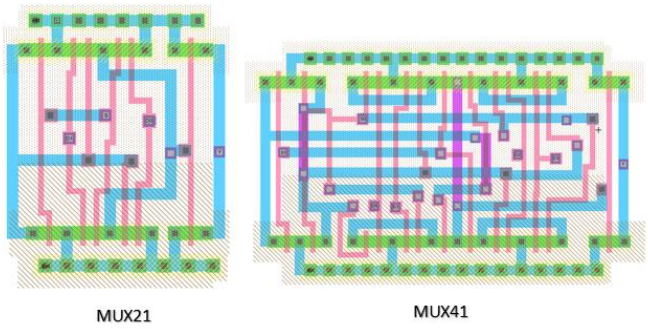
Compound gate yang dimaksud pada perancangan layout ini ialah seperti AOI21, AOI13, AOI22 dan juga OIA21, OAI13, serta OAI22, seperti yang terlihat pada Gambar 7.



Gambar 7. Layout untuk Compound Gate

2.3.3 Multiplexer

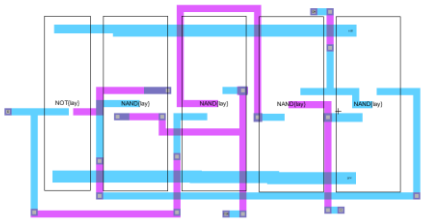
Multiplexer atau yang biasa disebut Mux yang dirancang berfungsi sebagai pemilih atau penyeleksi data, data yang manakah yang akan melewatkan data tersebut untuk diolah. Seperti yang terlihat pada Gambar 9.



Gambar 9. Layout unutk Multiplexer

2.3.4 Flip-flop

Flip – flop berfungsi sebagai penyimpan sementara, flip – flop yang digunakan ialah DFF dan DFF Set Reset (SR). layout yang akan digunakan terlihat pada Gambar 10.

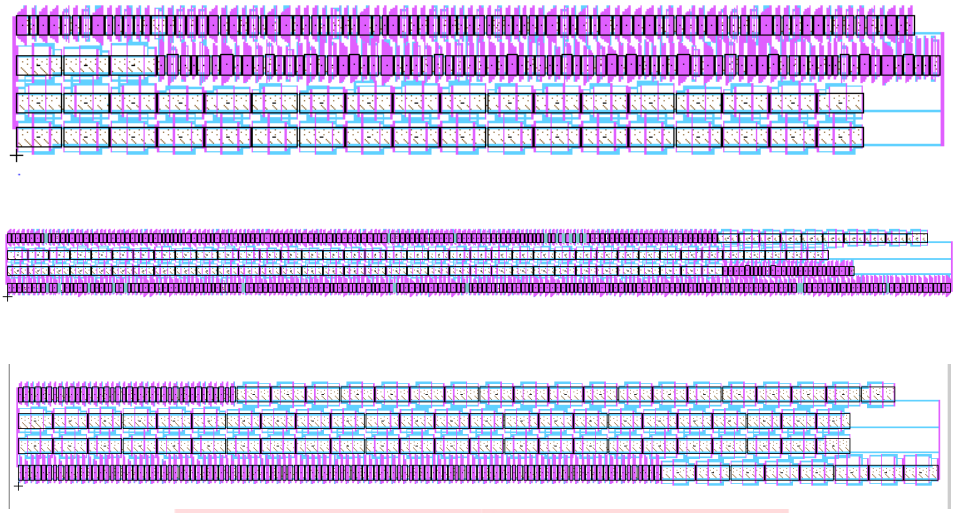


Gambar 10. Layout untuk Flip – flop

Dari layout – layout yang telah dirancang, didapat ukuran – ukuran untuk masing – masing layout. Tabel 4 akan menjelaskan ukuran – ukuran tersebut.

3. Pembahasan

3.1. Realisasi Layout

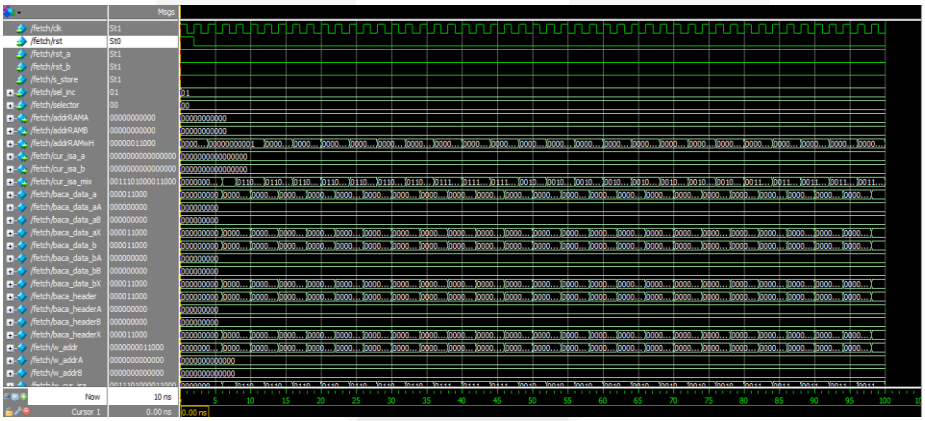


Gambar 11. Layput komponen fetching data

Pada gambar 11, menunjukkan layout dari komponen fetching data yang terdiri dari posisi paling atas ialah program counter dilanjutkan dengan nstruction memory dan juga instruction register.

3.2. Pengujian kebenaran logika dan fungsi

Untuk mengetahui keberhasilan proses fetching data, telah dilakukan proses simulasi menggunakan modelSim Altera untuk mengetahui apakah masing – masing komponen fetching berjalan sesuai fungsi nya masing - masing. Simulasi akan ditampilkan pada gambar 12 berikut.



Gambar 12. Hasil simulasi proses fetching

Tabel 4. Akan mendeskripsikan proses simulasi dari i/o untuk proses fetching data yang ditampilkan [ada gambar 12 diatas.

Tabel 4. Deskripsi proses fetching

input	Masuk ke	Output
1111111111111	Program counter	0000000000000
0000000000000	Instruction memory	16 bit set instruksi
9 bit akhir dari instruksi	Increment	11 bit alamat ROM
16 bit set instruksi	Instruction register	Instruksi yang sedang berjalan

Dari tabel 4 mejelaskan proses fetching data. Proses akan dimulai dari program counter yang akan menerima masukan barupa sekumpulan bit berjumlah 13 bit dan akan mengeluarkan alamat yang menjadi input bagi instruction memory. Keluaran dari program counter akan menjadi alamat selanjut nya yang akan mengakses data instruksi di instruction memory. Instruction register akan menerima masukan dari instruction memory berupa set instruksi yang sedang dieksekusi oleh prosesor.

4. Kesimpulan

Dari hasil pengujian dan analisa yang telah dilakukan, dapat disimpulkan sebagai berikut.

1. Proses fetching melibatkan 3 komponen, yaitu program counter, instruction memory dan instruction register.
2. Dari hasil pengujian, komponen fetching berjalan sesuai fungsi nya masing – masing .
3. Karena komponen fetching berjalan sesuai fungsi yang seharusnya, maka logika untuk masing – masing komponen sesuai dengan seharusnya.

Daftar Pustaka

[1] L. M. Jiji, “Architecture Implementation of Optimized DSP Accelerator with Modified Booth Recoder in FPGA.”

[2] D. Ho, “Towards a RISC Instruction Set Architecture for the 32-bit VLIW DSP Processor Core,” *Reg. 10 Symp. 2014 IEEE*, pp. 404–409, 2014.

[3] R. B. Lee, A. M. Fiskiran, Z. Shi, and X. Yang, “Refining instruction set architecture for high-performance multimedia processing in constrained environments,” *Proc. Int. Conf. Appl. Syst. Archit. Process.*, vol. 2002–Janua, pp. 253–264, 2002.

[4] N. Wirth, “The Design of a RISC Architecture and its Implementation with an FPGA,” pp. 1–24, 2015.

[5] I. I. Ntroduction, “Advanced Low Power RISC Processor Design using MIPS Instruction Set,” pp. 1252–1258, 2015.

[6] M. Uskoković and B. Ivančević, “The implementation of digital audio processors in analog multimedia audio systems,” *Conf. Proc. - ICECom 2003 17th Int. Conf. Appl. Electromagn. Commun.*, no. October, pp. 131–134, 2003.

[7] “What is Very Large Scale Integration (VLSI)? - Definition from WhatIs.com.” [Online]. Available: http://whatis.techtarget.com/definition/Very-Large-Scale-Integration-VLSI. [Accessed: 21-Jul-2017].

[8] “VLSI - Department of Electrical and Computer Engineering.” [Online]. Available: https://www.ece.ncsu.edu/research/ecs/vlsi. [Accessed: 21-Jul-2017].