

Desain *Datapath* Arsitektur Komputer MIC-1 8 bit Menggunakan IC74XX

Seprianto Ray Roganda Sianipar¹, Mochammad Hannats Hanafi Ichsan², Eko Setiawan³

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹raysianipar15@gmail.com, ²hanas.hanafi@ub.ac.id, ³Ekosetiawan@ub.ac.id

Abstrak

Arsitektur MIC-1 merupakan salah satu arsitektur komputer paling sederhana yang di dalamnya terdapat kumpulan register seperti *program counter*, *memory buffer register*, *register A*, *register B*, *external input* dan *arithmetic logic unit*. Kumpulan register digunakan sebagai *datapath* untuk jalannya proses eksekusi suatu instruksi. *Integrated circuit* (IC74XX) merupakan *integrated circuit* (IC) series yang digunakan untuk mengimplementasikan rangkaian logika sederhana. Dalam satu kemasan IC74xx terdapat beberapa macam *gate* (gerbang) yang dapat melakukan berbagai macam fungsi meliputi AND, OR, NOT, *adder*, dan *multiplexer*. Pada rangkaian *datapath* arsitektur komputer MIC-1 8 bit menggunakan *switch* dan IC74051. Fungsi *switch* adalah sebagai data *input* yang dimasukkan oleh *user* secara manual dan IC74051 untuk mengolah data *datapath* yang digunakan sebagai *input* bus B. Pada rangkaian *arithmetic logic unit* (ALU) menggunakan IC74LS32, IC74LS04, IC74LS283, IC74LS08, dan IC74LS251N dalam proses aritmatika dan logika. Rangkaian *datapath* dan *arithmetic logic unit* menggunakan daya sebesar 5 volt sebagai sumber daya dan LED sebagai *output*. Pada hasil pengujian fungsional didapatkannya hasil yang sesuai dengan perhitungan manual pada *multiplexer datapath operation* dan mengalami kendala untuk proses bagian penjumlahan pada pengujian non-fungsional.

Kata kunci: *arsitektur MIC-1, datapath, arithmetic logic unit, IC74XX*

Abstract

The architecture of the MIC-1 8 bits is one of the simplest computer architecture in which there are registers such as the program counter, a memory buffer register, register A, register B, an external input and integrated logic unit. The register collection was used as a datapath for the execution of instructions. Intergrated circuit 74XX (IC74XX) is an intergrated circuit (IC) series used to implementation simple logical series. IC74XX, in one set of IC74XX, there are several varieties of gates that can perform a variety of functions involving AND, OR, NOT, adder, and multiplexer. On a series of datapath computer architecture MIC-1 8 bits uses switches and IC74051. The switch function is as a data input entered by the user by hand and IC74051 to process datapath data that is used as a b bus *input*. Arihmetic logic unit (ALU) using IC74LS32, IC74LS04, IC74LS08, IC74LS08, and IC74LS251N in the arithmetic and logic process. Datapath and arithmetic logic unit (ALU) use 5-volt power as resources and LED as output. With functional testing results it obtained according to manual calculations in multiplexer datapath operation and was reduced to non-functional testing part.

Keywords: *MIC-1 architecture, datapath, arithmetic logic unit, IC74XX*

1. PENDAHULUAN

Datapath adalah kumpulan dari register yang berfungsi sebagai jalan dalam proses eksekusi suatu instruksi. *Datapath* menjadi salah satu bagian penting dari perkembangan dan perancangan sebuah arsitektur komputer MIC-1 8 bit. Arsitektur komputer MIC-1 8 bit adalah salah satu arsitektur komputer yang dirancang berdasarkan arsitektur Von

Neumann. Arsitektur ini merupakan salah satu arsitektur komputer paling sederhana yang di dalamnya terdapat *program counter*, *memory buffer register*, *register A*, *register B*, *external input* dan *arithmetic logic unit*. *Program counter* (PC), *memory buffer register* (MBR), *register A*, *register B*, dan *external input* merupakan penyimpanan data atau instruksi sedang diproses yang dikumpulkan menjadi satu untuk membentuk sebuah *datapath*. Sedangkan *arithmetic logic unit* (ALU) adalah

bagian yang bertugas untuk memproses operasi aritmatika maupun logika.

Pada penelitian sebelumnya, Derek telah berhasil *desain arithmetic logic unit* (ALU) pada *central processing unit* (CPU) 8 bit. Penelitian kedua, M. Hannast telah berhasil mensimulasikan arsitektur komputer MIC-1 8 bit pada logism. Penelitian ketiga, Ega telah berhasil menerapkan hasil simulasi *arithmetic logic unit* (ALU) logism menggunakan IC74xx. *Intergrated circuit* 74xx (IC74xx) merupakan *intergrated circuit* digital yang berfungsi sebagai gerbang logika. IC74xx ini memiliki keseimbangan antara kecepatan dan konsumsi daya rendah (*low power*) dibandingkan dengan IC lainnya. Adapun jenis-jenis dari IC74xx antara lain, IC74LS32 untuk IC OR, IC74LS08 untuk IC AND, IC74LS04 untuk IC NOT, IC74LS283 untuk ADDER, dan IC74251 untuk multiplexer.

Penelitian Ega Dewa Insantoro sudah mampu menyusun bagian *arithmetic logic unit* dari arsitektur komputer MIC-1 8 bit menggunakan IC74xx. Pada penelitian tersebut, *datapath* belum terealisasi. *Datapath* merupakan salah satu komponen yang harus ada dalam sebuah arsitektur komputer MIC-1 8 bit. Sehingga pada penelitian ini, penulis akan melakukan desain *datapath* arsitektur komputer MIC-1 8 bit dalam bentuk fisik menggunakan IC74xx.

2. LANDASAN KEPUSTAKAAN

2.1 Arsitektur Komputer

Arsitektur komputer merupakan atribut-atribut suatu sistem komputer yang terkait dengan seorang programmer dan memiliki dampak langsung pada eksekusi logis suatu program. Arsitektur komputer Von Neuman adalah salah satu arsitektur komputer yang menggambarkan komputer menjadi empat bagian utama yaitu, *register*, *unit control*, ALU, dan *I/O*. Arsitektur komputer MIC-1 merupakan arsitektur komputer yang dirancang berdasarkan arsitektur komputer Von Neumann yang terdiri dari *datapath*, ALU, dan *unit control*.

2.2 Arsitektur Komputer MIC-1

Aristektur komputer MIC-1 merupakan arsitektur komputer yang diciptakan oleh Andrew Tanenbaum dirancang sangat sederhana diantara arsitektur MIC lainnya.

Dalam membangun arsitektur komputer MIC-1 8 bit dibutuhkan beberapa komponen seperti *register A*, *register B*, *multiplexer*, *program counter* (PC), *memory byte register* (MBR), dan lain-lain.

2.3 Arithmetic Logic Unit

Arithmetic logic unit (ALU) merupakan salah satu bagian yang akan digunakan dalam membangun *central processing unit* 8 bit. Adapun fungsi arithmetic logic unit (ALU) adalah untuk melaksanakan pemrosesan data dalam bentuk aritmatika maupun logika.

2.4 Datapath

Datapath merupakan kumpulan dari *register* yang digunakan sebagai media penyimpanan data atau instruksi yang sedang diproses. Register yang digunakan dalam membangun *datapath* adalah *program counter*, *memory buffer register*, *register B*, dan *external input*.

2.5 Multiplexer

Multiplexer adalah rangkaian kombinasional yang dirancang khusus untuk seleksi salah satu dari beberapa jalur *input* menjadi satu jalur *output*. Adapun jumlah *input* dan *selector* pada multiplexer terdiri dari $2^n \times 1$ yaitu n merupakan jumlah selector, 2^n merupakan jumlah *input*, dan 1 merupakan jumlah output.

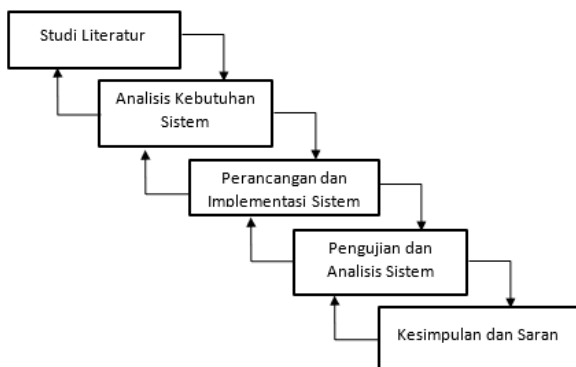
2.5 IC74XX

IC74xx merupakan bagian *intergrated circuit* (IC) series yang digunakan untuk mengimplementasikan rangkaian logika sederhana. Dalam satu kemasan IC74xx terdapat beberapa macam *gate* (gerbang) yang dapat melakukan berbagai macam fungsi meliputi AND, OR, NOT, *adder*, dan *multiplexer* sehingga *pin* (kaki) IC jumlahnya banyak dan bervariasi ada yang 8, 14, 16, 24, dan 40. Pada penelitian ini IC74xx yang akan digunakan meliputi IC74LS32 untuk operasi OR, IC74LS04 untuk operasi NOT, IC74LS08 untuk operasi AND, IC74LS283 untuk *adder*. IC74LS251N dan IC74LS44051 untuk *multiplexer*.

3. Metodologi Penelitian

Metodologi penelitian adalah langkah-langkah yang akan ditempuh dalam penelitian, meliputi studi literatur, analisis kebutuhan sistem, perancangan dan implementasi sistem,

pengujian dan analisis sistem, kesimpulan dan saran.



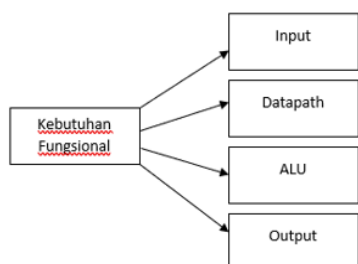
Gambar 1. Diagram Alur Metodologi Penelitian

Berdasarkan gambar 1 menjelaskan pengerjaan penelitian dimulai dengan studi literatur sampai dengan kesimpulan dan saran. Proses ini dilakukan secara berurutan dan tidak dapat di lewati. Setiap tahapan ini memiliki parameter tertentu agar dapat melanjutkan ke tahapan selanjutnya sebagai contoh pada bagian implementasi sistem, ketika berada pada tahapan ini implementasi yang dilakukan mempunyai parameter yang harus dipenuhi yaitu berdasarkan dari bagian perancangan sistem. Ketika bagian implementasi tidak terpenuhi, maka penelitian harus diulang pada bagian perancangan agar sistem dapat berjalan dengan fungsi yang telah ditentukan.

4. ANALISIS KEBUTUHAN SISTEM

4.1 Kebutuhan Fungsional

Kebutuhan fungsional pada penelitian ini menjelaskan kebutuhan yang harus dipenuhi oleh sistem sehingga dapat melaksanakan keseluruhan fungsinya. Pada penelitian ini kebutuhan yang harus dipenuhi yaitu *input*, *datapath*, *arithmetic logic unit*, dan *output*.



Gambar 2. Blok Diagram Kebutuhan Fungsional

Berdasarkan gambar 2 menjelaskan kebutuhan fungsional yang harus dilakukan oleh sistem meliputi sistem dapat menerima *input*, *datapath* dapat mengolah register, ALU dapat melakukan operasi aritmatika atau logika dan sistem dapat menampilkan output.

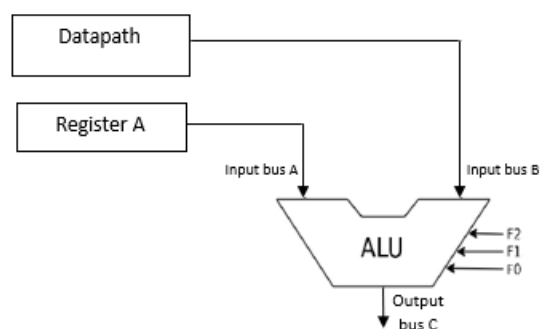
4.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan perangkat keras dan kebutuhan perangkat lunak. Kebutuhan perangkat keras yang digunakan meliputi IC74LS08, IC74LS32, IC74LS251, IC74LS04, IC74051, *switch* dan LED. Pada kebutuhan perangkat lunak menggunakan logism dan eagle.

5. PERANCANGAN DAN IMPLEMENTASI SISTEM

5.1 Perancangan Sistem

Perancangan sistem pada penelitian ini dapat berjalan apabila semua kebutuhan telah terpenuhi baik kebutuhan fungsional maupun kebutuhan non-fungsional. Penelitian ini akan membentuk sebuah arsitektur komputer MIC-1 8 bit yaitu dimulai dengan membuat perancangan *datapath* dan perancangan *arithmetic logic unit* (ALU). Berikut adalah bentuk perancangan dari arsitektur MIC-1 8 bit ditunjukkan pada gambar 3.



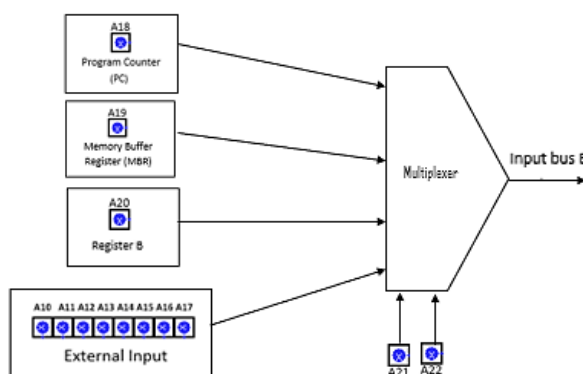
Gambar 3. Blok Diagram Rancangan Arsitektur MIC-1

Berdasarkan gambar 3 menjelaskan proses perancangan yang dilakukan pada arsitektur komputer MIC-1 8 bit. Dapat dilihat, proses perancangan arsitektur komputer MIC-1 8 bit terlebih dahulu melakukan perancangan pada *datapath* untuk menghasilkan *input bus B*. Dilanjutkan perancangan *arithmetic logic unit* dengan menggunakan *input bus A* dan *input bus B* sebagai masukan, dan F2,F1,F0 sebagai

selector. Output dari *arithmetic logic unit* didapat berdasarkan *input selector* yang diberikan. Pada penelitian ini register A di-*input* secara manual oleh *user* untuk digunakan langsung sebagai *input bus A*.

5.1.1 Perancangan Datapath

Perancangan *datapath* adalah perancangan yang dilakukan pada kumpulan *register* untuk digunakan sebagai *input bus B*. Berikut adalah perancangan *datapath* ditunjukkan pada gambar 4.



Gambar 4. Blok Diagram Rancangan Datapath

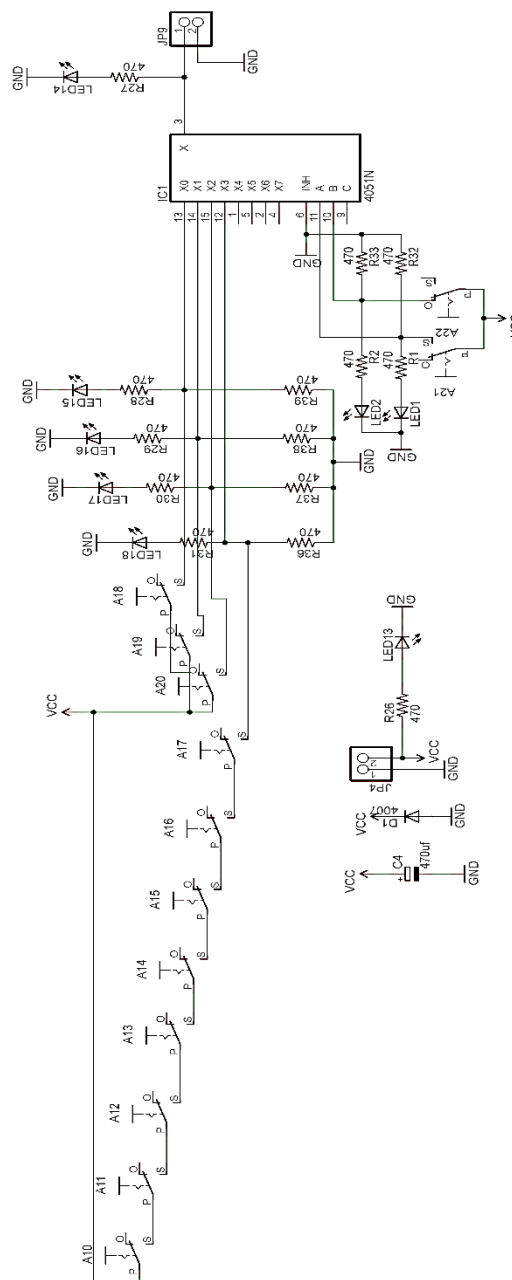
Berdasarkan gambar 4 menjelaskan rancangan *datapath* sebelum pembuatan *schematic*. Rancangan *datapath* tersebut terdiri dari *program counter* (PC), *memory buffer register* (MBR), *register B*, *external input*, dan *multiplexer*. Pada penelitian ini rancangan *datapath* mengimplimentasikan rangkaian sebanyak 8 buah. Penggunaan 8 buah rangkaian diperlukan untuk pengalokasian *memory* sebesar 8 bit, hal tersebut dikarenakan penelitian ini tidak menggunakan *external memory*. Rancangan *datapath* menggunakan *switch* (on/off) pada setiap *register* dan *selector*. Fungsi *switch* (on/off) adalah sebagai data *input* yang akan dimasukkan oleh *user* secara manual.

Pada penelitian ini *multiplexer* rancangan *datapath* digunakan untuk memilih *input* setiap *register* yang akan ditampilkan berdasarkan *multiplexer datapath operation*. Berikut adalah *multiplexer datapath operation* ditunjukkan pada tabel 1.

Tabel 1. Multiplexer Datapath Operation

F0	F1	OUT
0	0	PC
0	1	MBR
1	0	Register B
1	1	External Input

Berdasarkan tabel 1 *multiplexer datapath operation* tersebut terdiri dari F0, F1, dan OUT (output). Pada penelitian ini F0 dan F1 merupakan data *input selector* yang dimasukkan oleh *user* secara manual dan *output* merupakan hasil dari *input selector*. Berikut adalah rancangan *datapath* yang telah dibuat dalam bentuk *schematic* ditunjukkan pada gambar 5.



Gambar 5. Schematic Rancangan Datapath

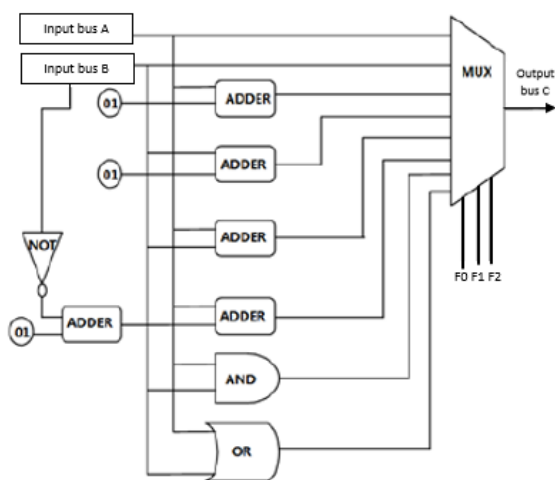
Berdasarkan gambar 5 menjelaskan rancangan *datapath* telah dirancang dalam *schematic*. Dalam *schematic* rancangan *datapath* tersebut terdiri dari *switch* (on/off), VCC sebagai *power supply*, GND sebagai pengaman kebocoran arus, *resistor* sebagai penghambat arus listrik, LED sebagai *output*,

dan *multiplexer*. *Switch* (on/off) digunakan sebagai *input* yang dimasukkan oleh user secara manual dan *multiplexer* berfungsi untuk menyeleksi *input* yang diberikan.

Pada penelitian ini *external input* ditunjukkan pada label A10-A17 terdiri dari 8 buah *switch* yang dirangkai secara seri, dimana jika 8 buah *input switch on* akan menghasilkan nilai 1 dan jika salah satu atau semua dari 8 buah *input switch off* menghasilkan nilai 0. *Program counter* (PC) ditunjukkan pada label A18 terdiri dari 1 buah *switch* (on/off), dimana jika *on* bernilai 1 dan jika *off* bernilai 0. *Memory buffer register* (MBR) ditunjukkan pada label A19 terdiri dari 1 buah *switch* (on/off), dimana jika *on* bernilai 1 dan jika *off* bernilai 0. *Register B* ditunjukkan pada label A20 terdiri dari 1 buah *switch* (on/off), dimana jika *on* bernilai 1 dan jika *off* bernilai 0. Label A21-A22 merupakan *selector* terdiri dari 2 buah *switch* (on/off), dimana jika *on* bernilai 1 dan jika *off* bernilai 0.

5.1.2 Perancangan Arithmetic Logic Unit

Perancangan *arithmetic logic unit* (ALU) digunakan untuk menyelesaikan operasi aritmatika dan logika. Operasi aritmatika terdiri dari penjumlahan dan operasi logika terdiri dari logika AND dan logika OR.



Gambar 6. Blok Diagram Rancangan Arithmetic Logic Unit

Berdasarkan gambar 6 menjelaskan rancangan *arithmetic logic unit* (ALU) sebelum pembuatan *schematic*. Rancangan *arithmetic logic unit* (ALU) tersebut terdiri dari *input bus A*, *input bus B*, gerbang *logic AND*, gerbang *logic OR*, gerbang *logic adder*, gerbang *logic not*, dan *multiplexer*. Pada penelitian rancangan ALU mengimplementasikan rangkaian sebanyak

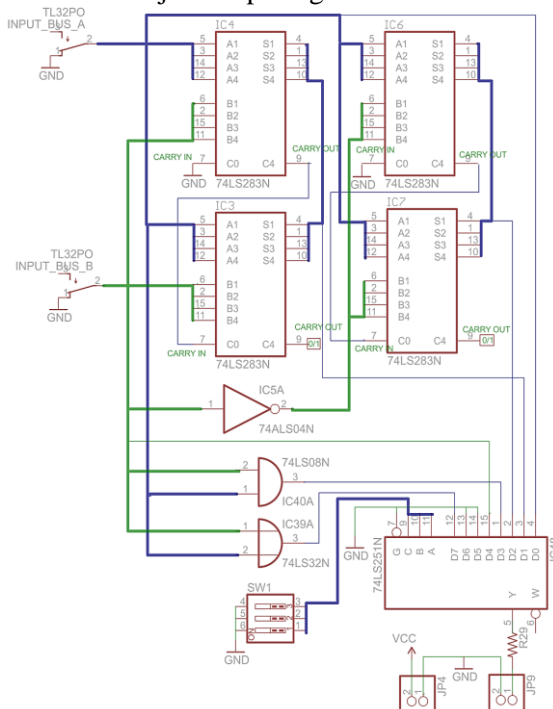
8 buah seperti pada gambar. *Input bus B* didapat berdasarkan perancangan *datapath* dan register A langsung digunakan sebagai *input bus A*.

Pada penelitian ini *multiplexer* rancangan *arithmetic logic unit* digunakan untuk memilih dari delapan buah operasi yang akan ditampilkan berdasarkan *multiplexer operation arithmetic logic unit* (ALU). Berikut adalah *multiplexer operation arithmetic logic unit* (ALU) ditunjukkan pada tabel 2.

Tabel 2. Multiplexer ALU Operation

F0	F1	F2	Output
0	0	0	A
0	0	1	A+B
0	1	0	A+0
0	1	1	AND
1	0	0	B
1	0	1	A+(-B)
1	1	0	B+0
1	1	1	OR

Berdasarkan tabel 2 menjelaskan *multiplexer arithmetic logic unit* (ALU) *operation*. *Multiplexer ALU operation* tersebut terdiri dari F0, F1, dan OUT (output). Pada penelitian ini F0, F1, dan F2 merupakan data *input selector* yang dimasukkan oleh user secara manual dan *output* merupakan hasil dari *input selector*. Berikut adalah rancangan *arithmetic logic unit* yang telah dibuat dalam bentuk *schematic* ditunjukkan pada gambar 7.



Gambar 7. Schematic Rancangan Arithmetic Logic Unit

Berdasarkan gambar 7 menjelaskan rancangan *arithmetic logic unit* telah dirancang dalam schematic. Schematic rancangan *arithmetic logic unit* terdiri dari *input* bus A, *input* bus B, VCC sebagai *power supply*, GND sebagai pengaman kebocoran arus, *resistor* sebagai penghambat arus listrik, LED sebagai *output*, gerbang *logic adder*, gerbang *logic AND*, gerbang *logic NOT*, gerbang *logic OR* dan *multiplexer*.

Operasi aritmatika pada *arithmetic logic unit* menggunakan IC74LS283, IC ini digunakan untuk penjumlahan bilangan pada posisi satuan saja, atau lebih umum bagian LSB-nya saja (LSB singkatan dari Least Significant Bit) karena tidak tersedia terminal masukan untuk menampung terjadinya simpanan dari posisi sebelumnya. Padahal proses penjumlahan pada umumnya melibatkan simpanan. Suatu rangkaian yang memenuhi syarat tersebut dikenal sebagai rangkaian penjumlah penuh (*full adder*). Sehingga rangkaian *full adder* dalam penelitian ini, penulis menggunakan 2 buah IC74LS283 yang dihubungkan secara kaskade untuk menyesuaikan penjumlahan bilangan-bilangan dengan bit yang lebih besar.

Dua bilangan yang dijumlahkan melalui 2 buah IC74LS283 masing-masing adalah $A = A_4 A_3 A_2 A_1$ dan $B = B_4 B_3 B_2 B_1$ yang hasilnya $S = S_4 S_3 S_2 S_1$.

Secara lebih rinci dapat dijelaskan bahwa IC 74LS283 (1) menjumlahkan bilangan 4 bit LSB (Least Significant Bit) yang simpanannya (C4) diumpankan ke masukan C0 pada IC 74LS283 (2). IC 74LS283 (2) menjumlahkan bilangan 4 bit MSB (Most Significant Bit) beserta simpanan yang dihasilkan dari IC 74LS283 (1).

Operasi logika pada *arithmetic logic unit* menggunakan IC74LS32 dan IC74LS08. IC74LS32 merupakan bagian dari IC74xx dirancang berdasarkan *gate* (gerbang) OR yang mempunyai dua atau lebih *input* dan memiliki satu *output*. Apabila salah satu *input* berlogika "1", maka *output* "1". Jika semua *input* berlogika "0", maka *output* akan berlogika "0". Sedangkan IC74LS08 merupakan bagian IC74xx dirancang berdasarkan *gate* (gerbang) AND yang memiliki dua atau lebih *input* dan memiliki satu *output*. *Output* akan berlogika "1" jika semua *input* (A AND B) berlogika "1". Jika salah satu *input* berlogika "0" maka *output* akan berlogika "0".

Pada penelitian ini rancangan *datapath* dan rancangan *arithmetic logic unit* akan saling

berhubungan dalam proses implementasi yaitu, nilai atau *input* yang dimasukkan dalam *input* bus B akan didapatkan melalui proses *datapath*, yang dihubungkan menggunakan kabel pelangi pada B *input* dalam proses implementasinya. Sedangkan *register* A langsung diteruskan menjadi *input* bus A dikarenakan hanya menggunakan *switch* (on/off).

Hal tersebut dilakukan dikarenakan proses dalam rancangan arsitektur komputer MIC-1 *input* bus B didapatkan melalui perancangan *datapath* dan *input* bus A didapat melalui *register* A. *Integrated circuit* 74LS251 (IC74LS251) merupakan gerbang *logic multiplexer* yang berfungsi sebagai ALU operation.

5.2 Implementasi Sistem

Implementasi sistem pada penelitian ini dapat berjalan apabila perancangan sistem telah terpenuhi baik perancangan *datapath* maupun perancangan *arithmetic logic unit* (ALU). Pada penelitian ini, implementasi sistem akan mengimplementasikan rangkaian *datapath* dan rangkaian *arithmetic logic unit* (ALU) menggunakan *integrated circuit* 74xx (IC74xx).

5.2.1 Implementasi Rangkaian Datapath

Implementasi *datapath* akan menjelaskan implementasi dari rancangan *datapath* yang sudah dilakukan sebelumnya. Pada penelitian ini menggunakan konektor daya sebesar 5 V, resistor 470 ohm, *switch* sebanyak 13 buah, LED sebanyak 8 buah, dan *integrated circuit multiplexer*. Dari 13 buah jumlah *switch* akan dibagi pengerjaannya yaitu 8 buah *switch* digunakan sebagai *external input*, 2 buah *switch* digunakan sebagai *selector*, 1 buah *switch* digunakan sebagai *program counter* (PC), 1 buah *switch* digunakan sebagai *memory buffer register* (MBR), dan 1 buah *switch* digunakan sebagai *register* B. Fungsi dari *switch* adalah memberikan nilai *input* yang dimasukkan oleh user secara manual.

Sedangkan dari 8 buah jumlah LED yang digunakan akan dibagi pengerjaannya juga yaitu 2 buah LED ke bagian *switch selector*, 1 buah LED ke bagian *switch external input*, 1 buah LED ke bagian *switch program counter* (PC), 1 buah LED ke bagian *switch memory buffer register* (MBR), 1 buah LED ke bagian *switch register* B, 1 buah LED ke bagian *integrated circuit multiplexer*, dan 1 buah LED ke bagian

konektor daya. Adapun fungsi dari LED yaitu untuk mengetahui bahwa *datapath* sudah terkoneksi dengan daya dan juga mengetahui bahwa jika LED menyala akan menandakan bernilai biner 1 dan jika LED mati menandakan LED bernilai biner 0. Berikut adalah proses rangkaian *datapath* sudah dapat berjalan maupun sudah diimplementasikan ditunjukkan pada gambar 8.



Gambar 8. Rangkaian Datapath

Berdasarkan gambar 8 menjelaskan implementasi rangkaian *datapath*. Setiap rangkaian *datapath* dicetak atau diimplementasikan pada PCB sesuai dengan *schematic* rancangan *datapath* yang ditunjukkan pada gambar 5. Implementasi setiap rangkaian *datapath* tersebut terdiri dari *program counter*, *memory buffer register* (MBR), *selector*, *external input*, LED, VCC, dan multiplexer.

Pada penelitian ini proses implementasi rangkaian *datapath* sesuai dengan perancangan *datapath* yang telah dibuat. *Switch program counter*, *switch memory buffer register* (MBR), *switch register B*, *switch external input*, dan *switch selector* merupakan *input* yang dimasukkan oleh user secara manual bernilai 1 (LED menyala). Jika *switch* mati, menandakan bahwa user belum melakukan *penginputan* nilai (LED mati). Dan *multiplexer* yang terdapat pada rangkaian *datapath* digunakan untuk memilah proses yang ingin ditampilkan.

Proses *program counter* dapat ditampilkan jika kedua *input selector* (*switch 1* dan *switch 2*) dalam keadaan mati atau bernilai 0. Proses *memory buffer register* (MBR) dapat ditampilkan jika *switch 1* bernilai 0 dan *switch 2* bernilai 1. Proses *register B* dapat ditampilkan jika *switch 1* bernilai 1 dan *switch 2* bernilai 0. Dan proses *external input* terdiri dari 8 buah *switch external input*, jika dari 8 buah *switch external input* bernilai 1 maka *input dari external input* adalah bernilai 1, jika salah satu

dari 8 buah *switch external input* bernilai 0 maka *input dari external input* adalah bernilai 0. Proses *external input* dapat ditampilkan jika *switch 1* bernilai 1 dan *switch 2* bernilai 1.

5.2.2 Implementasi Rangkaian ALU

Implementasi rangkaian *arithmetic logic unit* (ALU) merupakan rangkaian yang berfungsi memproses operasi aritmatika dan operasi logika. Setiap rangkaian *arithmetic logic unit* (ALU) dicetak atau diimplementasikan pada PCB sesuai dengan *schematic* rancangan *arithmetic logic unit* yang ditunjukkan pada gambar 7. Implementasi setiap rangkaian *arithmetic logic unit* tersebut terdiri dari *input bus A*, *input bus B*, IC74LS283, IC74HC04, IC74LS08, IC74LS32, dan IC74LS251. Adapun implementasi rangkaian *arithmetic logic unit* dapat dilihat pada gambar 9.



Gambar 9. Rangkaian ALU

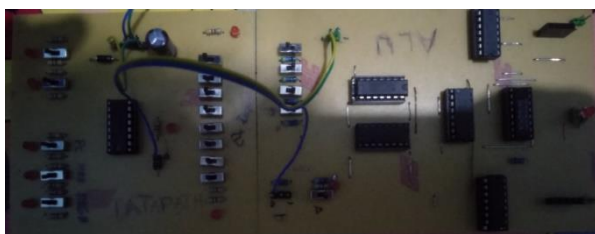
Berdasarkan gambar 9 menjelaskan implementasi rangkaian *arithmetic logic unit* dan. Pada penelitian ini rangkaian *arithmetic logic unit* menggunakan konektor daya sebesar 5 V, resistor sebesar 470 ohm, 5 buah *switch*, 6 buah LED, 2 buah IC adder (IC74LS283), 1 buah IC NOT (IC74HC04), 1 buah IC OR (IC74LS32), 1 buah IC AND (IC74LS08), 1 buah IC *multiplexer* (IC74LS251), dan 10 buah kabel pelangi. Fungsi *switch* adalah *input* bernilai 0 dan 1 yang dimasukkan oleh secara manual

LED yang digunakan dibagi ke beberapa bagian yaitu 3 buah LED akan digunakan sebagai *output switch selector*, 1 buah akan digunakan sebagai *output switch input bus A*, 1 buah akan digunakan sebagai *output multiplexer*, dan 1 buah LED akan digunakan sebagai *output ground*. Sedangkan dari 10 buah kabel pelangi dibagi ke beberapa bagian yaitu 2 buah kabel pelangi terhubung ke port rangkaian utama

arithmetic logic unit akan digunakan sebagai *input* B, 2 buah kabel pelangi terhubung ke port ground, 2 buah pelangi terhubung ke port daya, dan 4 buah kabel pelangi akan dihubungkan ke port rangkaian selanjutnya.

5.2.3 Implementasi Rangkaian *Datapath* dan ALU Saling Terhubung

Implementasi arsitektur komputer MIC-1 merupakan implementasi rangkaian *datapath* dan implementasi rangkaian *arithmetic logic unit* (ALU) saling terhubung. Berikut adalah implementasi rangkaian *datapath* dan *arithmetic logic unit* saling terhubung ditunjukkan pada gambar 10.



Gambar 10. Rangkaian *Datapath* dan ALU Terhubung

Berdasarkan gambar 10 menjelaskan implementasi rangkaian *datapath* dan *arithmetic logic unit* (ALU) sudah dapat saling terhubung. Rangkaian *datapath* terdapat 13 buah switch, 1 buah IC74051, 8 buah LED, dan 3 buah kabel pelangi. Pada rangkaian *datapath* 3 buah switch menginterpretasikan *program counter*, *memory buffer register*, dan *register* B. 2 buah switch menginterpretasikan selector dan 8 buah switch menginterpretasikan external *input*. 8 buah LED yang digunakan berfungsi sebagai output dari *input* nilai yang diberikan. 1 buah IC74051 digunakan untuk memproses *input* nilai yang telah dimasukkan oleh *user* secara manual.

Sedangkan 3 buah kabel pelangi dihubungkan pada port rangkaian *arithmetic logic unit* yang menginterpretasikan *input* bus B. Rangkaian *datapath* dan *arithmetic logic unit* akan dihubungkan pada 7 buah rangkaian kembar untuk mengimplementasikan arsitektur komputer MIC-1 sebesar 8 bit. 7 buah rangkaian kembar adalah rangkaian yang dirancang sama seperti rangkaian *datapath* dan ALU yang menginterpretasikan 7 bit *input* dan 7 bit *output* tanpa menggunakan selector.

6. Pengujian dan Analisis Sistem

6.1 Pengujian Fungsional

Pada penelitian ini pengujian fungsional dilakukan dengan cara membandingkan output implementasi dengan output manual pada *input* bus A, dan *datapath*. Output implementasi merupakan output dari hasil implementasi yang dilakukan oleh peneliti dan output manual merupakan output yang seharusnya ditampilkan pada bagian implementasi.

6.1.1 Pengujian *Input* Bus A

Pengujian *input* bus A dapat dibuktikan menggunakan *multiplexer ALU operation*, yaitu jika selector F0, F1, dan F2 bernilai biner 0 maka output adalah *input* A. Berikut pengujian yang telah dilakukan pada *input* bus A.

Tabel 3 Pengujian *Input* Bus A

Input A	F0	F1	F2	Output (Biner)	Output Manual (Biner)	Output (DEC)	Output Manual (DEC)	Hasil (Ya/Tidak)	Selisih Angka Error	Data Error
0000 0001	0	0	0	0000 0001	0000 0001	1	1	Ya	0	0
0000 0010	0	0	0	0000 0010	0000 0010	2	2	Ya	0	0
0000 0011	0	0	0	0000 0011	0000 0011	3	3	Ya	0	0

6.1.2 Pengujian *Program Counter*

Pengujian *program counter* dapat dibuktikan menggunakan *multiplexer datapath operation*, yaitu jika selector F0 bernilai biner 0 dan F1 bernilai biner 0, maka output adalah *program counter*. Berikut pengujian yang telah dilakukan pada *program counter*.

Tabel 4 Pengujian *Program Counter*

Program Counter (PC)	F0	F1	Output (Biner)	Output Manual (Biner)	Output (DEC)	Output Manual (DEC)	Hasil (Ya/Tidak)	Selisih Angka Error	Data Error
0000 0001	0	0	0000 0001	0000 0001	1	1	Ya	0	0
0000 0010	0	0	0000 0010	0000 0010	2	2	Ya	0	0
0000 0011	0	0	0000 0011	0000 0011	3	3	Ya	0	0

6.1.3 Pengujian *Memory Buffer Register*

Pengujian *memory buffer register* (MBR) dapat dibuktikan menggunakan *multiplexer datapath operation*, yaitu jika selector F0 bernilai biner 0 dan F1 bernilai biner 1, maka output adalah *memory byte register* (MBR). Berikut pengujian yang telah dilakukan pada MBR.

Tabel 5 Pengujian Memory Buffer Register

Memory Buffer Register (MBR)	F0	F1	Output	Output Manual (Biner)	Output (DEC)	Output Manual (DEC)	Hasil (Ya/Tidak)	Selisih Angka Error	Data Error
0000 0001	0	1	0000 0001	0000 0001	1	1	Ya	0	0
0000 0010	0	1	0000 0010	0000 0010	2	2	Ya	0	0
0000 0011	0	1	0000 0011	0000 0011	3	3	Ya	0	0

6.1.3 Pengujian Register B

Pengujian register B dapat dibuktikan menggunakan *multiplexer datapath operation*, yaitu jika selector F0 bernilai biner 1 dan F1 bernilai biner 0, maka output adalah register B. Berikut pengujian yang telah dilakukan pada register B.

Tabel 6 Pengujian Register B

Register B	F0	F1	Output (Biner)	Output Manual (Biner)	Output (DEC)	Output Manual (DEC)	Hasil (Ya/Tidak)	Selisih Angka Error	Data Error
0000 0001	1	0	0000 0001	0000 0001	1	1	Ya	0	0
0000 0010	1	0	0000 0010	0000 0010	2	2	Ya	0	0
0000 0011	1	0	0000 0011	0000 0011	3	3	Ya	0	0

6.1.4 Pengujian External Input

Pengujian external input dapat dibuktikan menggunakan *multiplexer datapath operation*, yaitu jika selector F0 bernilai biner 1 dan F1 bernilai biner 1, maka output adalah external input. Berikut pengujian yang telah dilakukan pada external input.

Tabel 7 Pengujian External Input

External Input	F0	F1	Output (Biner)	Output Manual (Biner)	Output (DEC)	Output Manual (DEC)	Hasil (Ya/Tidak)	Selisih Angka Error	Data Error
0000 0001	1	1	0000 0001	0000 0001	1	1	Ya	0	0
0000 0010	1	1	0000 0010	0000 0010	2	2	Ya	0	0
0000 0011	1	1	0000 0011	0000 0011	3	3	Ya	0	0

6.2 Pengujian Non-Fungsional

Pada penelitian ini pengujian non-fungsional dilakukan dengan cara membandingkan output implementasi dengan output manual pada logika AND, logika OR, A+B, dan A+(-B). Output implementasi merupakan output dari hasil implementasi yang dilakukan oleh peneliti dan output manual merupakan output yang seharusnya ditampilkan pada bagian implementasi.

6.2.1 Pengujian Logika AND

Pengujian logika AND dapat dibuktikan menggunakan *multiplexer ALU operation*, yaitu jika selector F0 bernilai biner 0, F1 bernilai biner 1 dan F2 bernilai biner 1, maka akan melakukan operasi logika AND. Berikut pengujian yang telah dilakukan pada logika AND.

Tabel 8 Pengujian Logika AND

Input bus A	Input bus B	F0	F1	F2	Output (BINER)	Output Manual (BINER)	Output (DEC)	Output Manual (DEC)	Hasil (Ya/Tidak)	Selisih Angka Error	Data Error
0000 0001	0001 0000	0	1	1	0000 0000	0000 0000	0	0	Ya	0	0
0000 0010	0001 0001	0	1	1	0000 0000	0000 0000	0	0	Ya	0	0
0000 0011	0001 0010	0	1	1	0000 0010	0000 0010	2	2	Ya	0	0

6.2.2 Pengujian Logika OR

Pengujian logika OR dapat dibuktikan menggunakan *multiplexer ALU operation*, yaitu jika selector F0 bernilai biner 1, F1 bernilai biner 1 dan F2 bernilai biner 1, maka akan melakukan operasi logika OR. Berikut pengujian yang telah dilakukan pada logika OR.

Tabel 9 Pengujian Logika OR

Input bus A	Input bus B	F0	F1	F2	Output (BINER)	Output Manual (BINER)	Output (DEC)	Output Manual (DEC)	Hasil (Ya/Tidak)	Selisih Angka Error	Data Error
0000 0001	0001 0000	1	1	1	0001 0001	0001 0001	17	17	Ya	0	0
0000 0010	0001 0001	1	1	1	0001 0011	0001 0011	19	19	Ya	0	0
0000 0011	0001 0010	1	1	1	0001 0011	0001 0011	19	19	Ya	0	0

6.2.3 Pengujian A+B

Pengujian aritmatika A+B dapat dibuktikan menggunakan *multiplexer ALU operation*, yaitu jika selector F0 bernilai biner 0, F1 bernilai biner 0 dan F2 bernilai biner 1, maka akan melakukan operasi aritmatika A+B. Berikut pengujian yang telah dilakukan pada A+B.

Tabel 10 Pengujian A+B

Input bus A	Input bus B	F0	F1	F2	Output (BINER)	Output Manual (BINER)	Output (DEC)	Output Manual (DEC)	Hasil (Ya/Tidak)	Selisih Angka Error	Data Error
0000 0000	0001 1111	0	0	1	0001 1111	0001 1111	31	31	Ya	0	0
1111 1000	0000 0011	0	0	1	1111 0011	1111 1010	243	250	Tidak	7	1
1111 0000	0000 0100	0	0	1	1111 0100	1111 0100	244	244	Ya	0	0

6.2.4 Pengujian A+(-B)

Pengujian aritmatika A+(-B) dapat dibuktikan menggunakan *multiplexer ALU operation*, yaitu jika selector F0 bernilai biner 1, F1 bernilai biner 0 dan F2 bernilai biner 1, maka

akan melakukan operasi aritmatika $A+(-B)$. Berikut pengujian yang telah dilakukan pada $A+(-B)$.

Tabel 11 Pengujian $A+(-B)$

Input bus A	Input bus B	F0	F1	F2	Output (BINER)	Output Manual (BINER)	Output (DEC)	Output Manual (DEC)	Hasil (Ya/Tidak)	Selisih Angka Error	Data Error
0000 0001	0000 0001	1	0	1	0000 0001	0000 0000	1	0	Tidak	1	1
0000 0010	0000 0010	1	0	1	0000 0000	0000 0000	0	0	Ya	0	0
0000 0011	0000 0011	1	0	1	0000 0001	0000 0000	1	0	Tidak	1	1

6.3 Analisis Sistem

Analisis sistem dilakukan dengan cara menghitung jumlah selisih angka error dan data error dari tiap pengujian.

7. Kesimpulan Dan Saran

7.1 Kesimpulan

Berdasarkan hasil perancangan sistem, implementasi sistem, pengujian sistem dan analisis sistem penelitian ini dapat disimpulkan sebagai berikut :

1. Dalam desain *datapath* membutuhkan *switch* (on/off), IC74051 dan LED. *Switch* digunakan oleh user untuk memberi masukan nilai 0 atau 1 pada kumpulan register, IC74051 digunakan untuk mengolah data pada kumpulan register dan LED sebagai *output*.
2. Pada desain *datapath* menggunakan 8 buah rangkaian yang sama dan setiap rangkaian saling terhubung untuk menginterpretasikan external memory sebesar 8 bit.
3. Dalam desain *arithmetic logic unit* (ALU) membutuhkan IC gerbang logika untuk memproses aritmatika dan logika. Komponen IC yang dibutuhkan adalah IC74LS283 sebagai gerbang *logic adder*, IC74LS04 sebagai gerbang *logic NOT*, IC74LS32 sebagai gerbang *logic OR*, IC74LS08 sebagai gerbang *logic AND* dan IC74LS251 sebagai gerbang *logic multiplexer*.
4. Pada desain *arithmetic logic unit* (ALU) menggunakan 8 buah rangkaian yang sama dan setiap rangkaian saling terhubung untuk dapat mengelola data sebesar 8 bit yang diterima dari *datapath* sebagai *input bus B*.
5. Fungsionalitas desain *datapath* arsitektur komputer MIC-1 8 bit menggunakan IC74xx belum berjalan sebagaimana mestinya. Hal tersebut dikarenakan dari pengujian yang

dilakukan masih terdapat selisih angka *error* dan data *error* pada operasi aritmatika.

7.1 Saran

Berikut beberapa saran dalam mengembangkan sistem ini :

1. Pada rancangan sistem *datapath* dan alu masih menggunakan 8 buah rangkaian. Dapat dikembangkan menjadi 1 buah rangkaian, salah satunya yaitu menggunakan FPGA.
2. Pada fungsi yang menggunakan adder masih belum sempurna dikarenakan output sistem masih banyak mengalami perbedaan dengan hitungan manual (*error*).

8. Daftar Pustaka

- Ichsan, M. H., & Kurniawan, W. (2017). Design and Implementation 8 bit CPU Architecture on Logisim For Undergraduate Learning Support. International Conference on Sustainable Information Engineering and Technology (SIET).
- Iswantoro, E. D., Ichsan, M. H., & Kurniawan, W. (2019). Desain Arithmetic Logic Unit 8bit untuk Central Processing Unit 8bit. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3, 8481-8487.
- Kurniawan, A. (2010). *Rancang Bangun Perangkat Sistem Pengujian IC Digital*. Depok: Universitas Indonesia.
- Schuurman, D. C. (2017). Step-by-Step Design and Simulation of A Simple CPU Architecture. Denver: Proceeding of the 44th ACM technical symposium on Computer science education.
- Semiconductor Components Industries, LLC. (2019). *74LS04*. Retrieved May 2020, 15, from AllDatasheet: <https://www.alldatasheet.com/datasheet-pdf/pdf/12617/ONSEMI/74LS04.html>
- Stallings, W. (2010). *Computer Organization and Architecture : Designing for Performance*. New Jersey: Pearson Prentice Hall.
- Tanenbaum, A. S. (2005). *Structured Computer Organization*. Amsterdam: Pearson Prentice Hall.