

PERANCANGAN KOMPUTER SEDERHANA DAN IMPLEMENTASINYA PADA FPGA ALTERA FLEX 10K EPF10K70RC240-4

Meicsy E. I. Najooan

Abstrak

Tahap awal perancangan komputer sederhana adalah dengan memahami siklus operasi dari setiap instruksi, organisasi perangkat keras, bus, signal kontrol, dan pewaktu. Perancangan komputer sederhana ini mengacu pada arsitektur sederhana dari suatu sistem komputer yang terdiri dari tiga unit utama, yaitu prosesor, memori dan port masukan keluaran.

Model bahasa pemrograman *hardware* seperti VHDL (*Very high speed integrated circuite Hardware Description Language*) digunakan untuk mendiskripsikan perancangan perangkat keras. Setelah penulisan program selesai, dilakukan kompilasi untuk mendapatkan simulasi dari sistem yang dirancang. Untuk rancangan ini digunakan perangkat lunak kompilator Max+Plus II versi 10.2 dari Altera. Simulasi ini digunakan sebagai pengujian untuk menentukan apakah sistem sudah berjalan dengan baik.

Implementasi dari rancangan ini dengan menggunakan *development board Altera UP2* berbasis chip FPGA Altera FLEX 10K EPF10K70KRC240-4. Pengujian pada perangkat keras dengan menggunakan *memory utilization file* (*.mif). Program bahasa rakitan yang dirancang dapat ditulis disini.

Kata Kunci : Komputer, FPGA, VHDL

I. Pendahuluan

Komputer sederhana terdiri dari tiga unit utama, yakni unit pemroses utama atau *Central Processing Unit (CPU)*, unit memori yang menyimpan instruksi-instruksi program dan data, dan unit masukan/keluaran untuk komunikasi dengan peralatan luar. Ketiga unit ini dihubungkan dengan kumpulan signal-signal digital paralel yang disebut *bus*. Kemampuan utama dari komputer sederhana ini adalah mengambil, mengkodekan dan mengeksekusi perintah yang diberikan.

Untuk mengetahui cara kerja dari komputer sederhana, digunakan *development board* yang dibuat oleh Altera berupa UP2 CPLD Boards dengan VHDL (*Very high speed integrated circuits Hardware Description Language*) suatu bahasa pemrograman hardware yang dapat digunakan untuk merancang komputer sederhana sampai pada pengujian tahap akhir.

II. Tujuan

Adapun tujuan penulisan ini untuk merancang komputer sederhana yang dapat melakukan pengambilan, pengkodean dan eksekusi sesuai dengan perintah. Rancangan ini harus dapat bekerja pada perangkat lunak Altera MaxPlus II serta pada FPGA Altera FLEX 10K EPF10K70RC240-4.

III. Pembatasan Masalah

Pelaksanaan perancangan komputer sederhana ini, dibatasi pada :

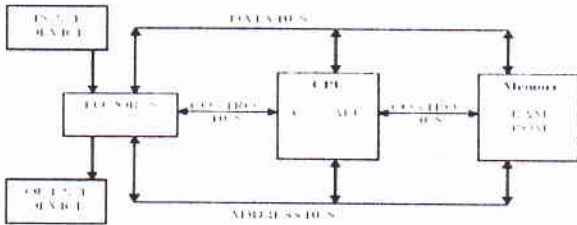
- sistem yang dirancang adalah komputer sederhana dengan jumlah instruksi terbatas dan setiap instruksi terdiri dari 16 bit.
- menggunakan program VHDL untuk disain sistem dan MAX+plus II sebagai *compiler*.
- menggunakan *UP2 Development board* untuk mendapatkan fungsi sistem sebenarnya.

IV. Tinjauan Pustaka

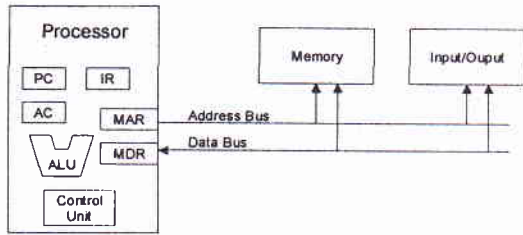
IV.1. Struktur dan Fungsi Komputer

Struktur komputer didefinisikan sebagai cara-cara dari tiap komponen saling terkait. Struktur sebuah komputer secara sederhana, dapat digambarkan dalam diagram blok pada Gambar 1. Sedangkan fungsi komputer didefinisikan sebagai operasi masing-masing komponen sebagai bagian dari struktur.

Di dalam CPU terdapat sejumlah register yang digunakan untuk menyimpan data sementara di dalam prosesor. Register-register seperti PC (*Program Counter*), IR (*Instruction Register*), AC (*Accumulator*), MAR (*Memory Address Register*), dan MDR (*Memory Data Register*) yang dibangun menggunakan *D flip-flop* untuk menyimpan data. ALU digunakan untuk melakukan pengolahan data berupa operasi aritmetik dan *logic*. Operasi dari ALU berupa *add*, *subtract*, dan operasi *logic* seperti *and/or*. Arsitektur dari sistem komputer juga dapat dilihat pada Gambar 2.



Gambar 1. Blok diagram komputer sederhana



Gambar 2. Arsitektur Sistem Komputer

4. 2. Program Komputer dan Instruksi

Sebuah program komputer adalah urutan instruksi yang melakukan operasi tertentu. Instruksi-instruksi ini disimpan dalam memori. Setiap instruksi terdiri dari urutan bit-bit, misalnya satu instruksi berisi 16 bit. Isi dari tiap-tiap bit dikelompokkan seperti 8 bit tertinggi dari instruksi tersebut berisi *opcode*.

Kode operasi instruksi (*opcode*) menunjukkan operasi seperti, *add* atau *subtract* yang akan dilakukan oleh instruksi. Secara khusus, satu instruksi mengirimkan satu set data (*opcode*) melalui ALU untuk melakukan operasi tersebut. Untuk 8 bit terendah dari tiap instruksi berisi bidang alamat dari memori. Tergantung dari kode operasi instruksi tersebut, alamat mungkin menunjuk ke lokasi data atau lokasi instruksi lain. Contoh format instruksi pada Gambar 3.



Gambar 3. Format Instruksi

Beberapa contoh instruksi-instruksi beserta operasi yang dilakukan dapat dilihat sbb:

Instruction Mnemonic	Operation Performed	Opcode Value
ADD <i>address</i>	AC ← AC + contents of memory address	00
STORE <i>address</i>	contents of memory address ← AC	01
LOAD <i>address</i>	AC ← contents of memory address	02
JUMP <i>address</i>	PC ← <i>address</i>	03
JNEG <i>address</i>	If AC < 0 Then PC ← <i>address</i>	04

Contoh program untuk menghitung $A = B + C$, dalam bahasa rakitan.

Assembly Language
Machine Language

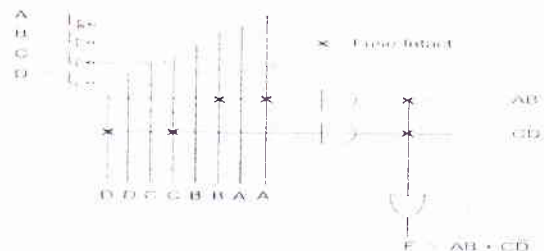
```
LOAD B      0211
ADD C       0012
STORE A     0110
```

Penetapan dari alamat data tidak boleh sama dengan alamat instruksi. Biasanya data disimpan dalam memori setelah semua instruksi dalam satu program. Pada contoh diatas, diasumsikan bahwa program dimulai pada alamat 0 dan ketiga instruksi akan menggunakan alamat mulai dari 0, 1 dan 2.

4.3. Peralatan Logika yang Dapat Diprogram (*Programmable Logic Device / PLD*)

Rangkaian logika digital yang dirancang dapat diimplementasikan dengan *integrated circuit chip* (Chip IC) seperti SSI dan MSI TT, dimana Chip ini akah melakukan fungsi operasi yang tetap sesuai pabrikan. Seorang perancang menghubungkan beberapa tipe chip berbeda untuk membuat rangkaiannya. *Programmable Logic Device/ PLD* adalah salah satu teknologi *programmable logic* dimana operasional fungsi internal IC dapat didefinisikan sendiri oleh perancang (*user*) dengan terlebih dahulu memprogram operasi-operasi yang diinginkan.

Salah satu contoh dari PLD adalah *programmable logic array* (PLA), seperti pada gambar 4. Persamaan logik pertama di-*minimize* dan ditempatkan dalam bentuk *Sum Of Product* (SOP). PLA mempunyai empat input, A, B, C dan D dimana tiap input dihubungkan ke inverter.



Gambar 4. Penggunaan PLA Diimplementasikan Pada Persamaan SOP

Masing-masing titik diimplementasikan dengan AND gate dengan input, dan output gerbang AND dimasukkan ke gerbang OR. Pada ujung dari input gerbang AND diletakkan *fuses* pada input A dan B yang akan menghasilkan perkalian antara A dan B. Hal yang sama juga untuk gerbang AND yang bawah. Jadi dengan adanya teknologi ini, dapat dibuat juga dengan cara yang sama untuk perancangan digital yang sudah lebih rumit

4.4. VHDL (*Very high speed integrated circuit Hardware Description Language*)

VHDL adalah suatu bahasa modular tingkat tinggi yang secara khusus digunakan untuk

logika kombinasional yang kompleks, kelompok-kelompok operasi, *state* mesin dan tabel kebenaran. Penulisan bahasa berupa *file text* yang dapat juga menggunakan *text editor*. Dengan aturan sintaks penulisan, maka setelah penulisan program selesai dilakukan *compile* yang hasilnya berupa simulasi dari bentuk rangkaian logika yang ditulis.

Sebagai contoh misalnya, akan dilakukan penjumlahan, maka *statement* VHDL dapat ditulis :

$$A \leq B + C;$$

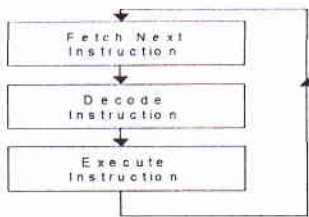
dimana akan secara otomatis membangkitkan *logic circuit* penjumlahan sesuai dengan jumlah bit untuk membangkitkan nilai baru pada A.

V. Perancangan Komputer Sederhana

Fungsi dasar dari komputer adalah eksekusi program. Program yang akan dieksekusi berisi sejumlah instruksi yang tersimpan di dalam memori. *Central Processing Unit* (CPU) melakukan tugas ini dengan cara mengeksekusi suatu program.

5.1. Siklus Pengolahan Instruksi

Proses pengolahan instruksi yang dilakukan oleh CPU atau prosesor dimulai dengan membaca atau mengambil (*fetch*) sebuah instruksi dari memori, mengkodekan (*decodes*) instruksi untuk menentukan operasi apa yang akan dilakukan, kemudian mengeksekusi instruksi. Gambar 5 memperlihatkan tahapan-tahapan tersebut.



Gambar 5. Tahapan Proses Pengolahan Instruksi

State machine sederhana yang disebut control unit mengatur urutan operasi-operasi dalam prosesor. Siklus *fetch*, *decode*, dan *execute* terdapat pada semua jenis prosesor. Implementasi dari siklus ini memerlukan beberapa operasi perpindahan register dan *clock cycles* dalam perancangan.

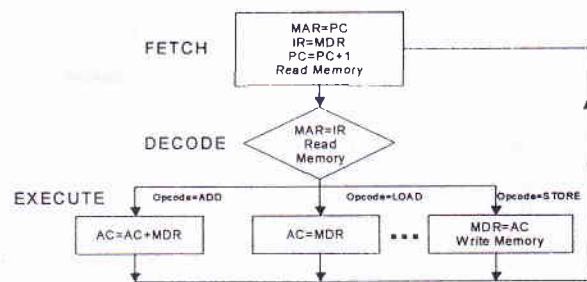
Program Counter (PC) berisi alamat dari instruksi yang baru akan diambil dari memori. Biasanya pengambilan instruksi berikutnya dari memori prosesor harus menambah isi PC. Prosesor kemudian mengirim alamat ini ke memori melalui bus dengan terlebih dahulu memuatnya ke *Memory Address Register* (MAR) dan memulai operasi pembacaan dari memori ke CPU. Selang beberapa waktu (dalam ukuran nano

second) instruksi atau data telah berada pada data bus dan akan masuk ke *Memory Data Register* (MDR).

Eksekusi instruksi memerlukan penambahan siklus memori dimana instruksi biasanya disimpan dalam *Instruction Register* (IR). Kemudian mengambil bagian kode operasi (*opcode*) dari instruksi untuk dikodekan. Eksekusi instruksi akan membutuhkan operasi-operasi tambahan dalam CPU dan juga membutuhkan operasi-operasi tambahan dalam memori.

Accumulator (AC) adalah register utama yang digunakan untuk melakukan perhitungan dan tempat sementara data program dalam prosesor. Setelah selesai mengeksekusi instruksi, prosesor memulai lagi mengambil instruksi berikutnya.

Siklus *fetch*, *decode*, dan *execute* dapat diimplementasikan menggunakan urutan operasi transfer register seperti pada Gambar 6.



Gambar 6. Urutan Operasi Transfer Register untuk komputer sederhana

5.2. Fungsi dan Operasi Prosesor

Untuk menjelaskan fungsi dan operasi prosesor secara detail, bentuk arsitektur komputer sederhana yang akan digunakan seperti pada Gambar 2. Prosesor atau CPU terdiri dari *general-purpose data register* (register data penggunaan umum) yang disebut *accumulator* (AC) dan *program counter* (PC). *Arithmetic logic unit* (ALU) digunakan untuk operasi-operasi aritmetik dan logika.

Berikut ini adalah penjelasan bagaimana fungsi dan operasi dari pengolahan instruksi oleh CPU. Program dan data diasumsikan telah berada di memori utama. Pengambilan instruksi pertama diambil dari memori dengan operasi perpindahan register berikut ini:

MAR = PC Read Memory,

MDR=Instruksi dari memori

IR = MDR

PC = PC + 1

Setelah urutan operasi ini, instruksi sekarang berada di *Instruction Register* (IR). Instruksi ini misalnya salah satu dari instruksi seperti ADD, LOAD atau STORE. Bidang kode operasi (*opcode*) diuji untuk dikodekan sesuai spesifikasi

instruksi mesin. Bidang alamat dari instruksi register berisi alamat dari *operand* atau data yang akan dioperasikan.

VI. IMPLEMENTASI KOMPUTER SEDERHANA PADA FPGA

Tahap akhir perancangan komputer sederhana adalah pengujian pada board Altera UP2 board yang berbasis chip FPGA Altera FLEX 10K EPF10K70RC240-4 dan CPLD Altera Max 7000S EPM7128S. Pengujian dilakukan dengan mengamati timing diagram dan timing analysis untuk memastikan bahwa rancangan dapat diimplementasikan dalam bentuk perangkat keras.

6.1. Board Altera UP2

Board Altera μ P2 memiliki dua *chip* PLD (*Programmable Logic Device*), yaitu FPGA (*Field Programmable Gate Array*) FLEX 10K EPF10K70RC240-4 dan CPLD (*Complex Programmable Logic Device*) MAX 7000S EPM7128S. Dari dua *chip* tersebut hanya *chip* FPGA yang digunakan karena memiliki kapasitas yang lebih tinggi. FPGA adalah *chip* yang berisi gerbang-gerbang logika yang bisa diprogram.

EPF10K70RC240-4 memiliki 70000 gerbang logika yang dikelompokkan menjadi 3744 elemen logika (*logic elemen*) dan 9 blok EAB (*Embedded Array Block*). Elemen logika biasa juga disebut sebagai sel logika (*Logic Cell*) terdiri atas satu LUT (*Look-up Table*), satu *flip-flop* dan jalur-jalur sinyal untuk fungsi *carry* dan *cascade*. EAB memiliki memori 2 KB yang bisa digunakan untuk RAM, ROM serta fungsi-fungsi FIFO (*First-in First-Out*). *Packaging* yang digunakan pada *chip* tersebut adalah jenis PQFP (*Power Quad Flat Package*) dengan 240 buah pin. Tegangan listrik yang dibutuhkan *chip* sebesar 5 Volt.

Board Altera μ P2 menyediakan beberapa fasilitas masukan dan keluaran yang terhubung ke *chip* EPF10K70RC240-4. Fasilitas tersebut adalah sebagai berikut :

➤ Masukan

- Osilator *clock* dengan frekwensi 25,175
- Dua tombol tekan (*push botton*), yaitu FLEX PB1 dan FLEX PB2.
- Delapan buah saklar (*switch*) bernama FLEX_SWITCH.
- Dua digit LED tujuh segmen (FLEX DIGIT)
- Antarmuka untuk *mouse* jenis PS2

➤ Keluaran

- Antarmuka VGA dengan resolusi maksimum 640x480 dan dapat menghasilkan 8 kombinasi warna

➤ Masukan dan Keluaran (*bidirectional*) :

- Tiga buah jalur ekspansi, yaitu FLEX_EXPAN_A, FLEX_EXPAN_B, dan FLEX_EXPAN_C, masing-masing memiliki 60 pin.

Pada gambar 7 memperlihatkan *board* Altera UP2.



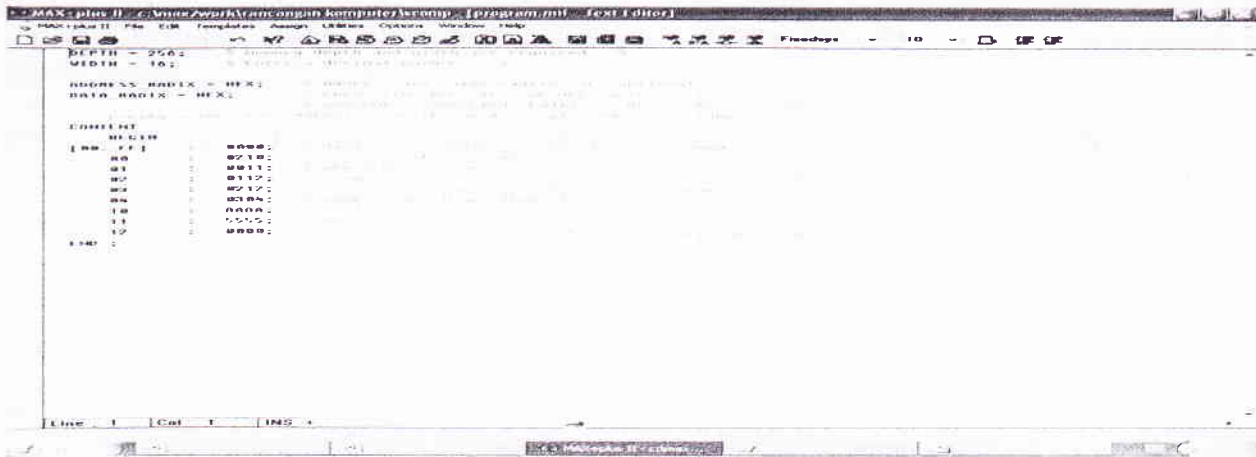
Gambar 7. Altera UP2 CPLD Boards

6.2. Model VHDL

Model VHDL dari rancangan komputer sederhana dimana impelentasinya pada device FLEX 10K20 dan untuk memori RAM menggunakan fungsi LPM_RAM_DQ. Program bahasa mesin seperti pada Gambar 8 dimuat ke memori menggunakan *memory utilization file* (*.mif). Ini menghasilkan 256 kata pada 16-bit memori untuk instruksi dan data. Program mif dapat diedit dan dimuat ke program. Operasi penulisan ke memori dilakukan ketika signal *memory_write* dibuat tinggi (*high*). Device FLEX 10K20, waktu akses memori berada dirange 20-50 ns.

Berdasarkan arsitektur komputer sederhana, program VHDL pertama mendeklarasikan register-register internal untuk prosesor seperti pada Gambar 9. Statement CASE digunakan untuk mengimplementasikan control unit dari state mesin. *State* reset diperlukan untuk inialisasi prosesor. Pada saat reset state, beberapa register diset 0 dan pembacaan memori dimulai pada instruksi pertama. Pada saat ini prosesor akan mulai mengeksekusi instruksi pda loksi 0.

Pada siklus pengambilan instruksi, isi PC akan ditambah dengan 1 dan instruksi akan ditransfer ke IR. Setelah clock signal berikutnya, siklus pengkodean akan dimulai. Pada pengkodean ini, delapan bit terendah dari instruksi yang berada di IR digunakan untuk operasi pembacaan data (*operand*) dari memori. Pada state pengkodean terdapat statement CASE untuk mengkodekan instruksi pada delapan bit teratas yang merupakan *opcode*. Jadi komputer sederhana dapat dibuat sebanyak $2^8 = 256$ instruksi berbeda. Pada program ini hanya diimplementasikan 4 instruksi.



Gambar 8. MIF File berisi Program Komputer Sederhana

```

ENTITY SCOMP IS
PORT(  clock, reset          : IN STD_LOGIC;
      program_counter_out    : OUT STD_LOGIC_VECTOR( 7 DOWNTO 0 );
      register_AC_out        : OUT STD_LOGIC_VECTOR(15 DOWNTO 0 );
      memory_data_register_out : OUT STD_LOGIC_VECTOR(15 DOWNTO 0 ));
END SCOMP;
ARCHITECTURE a OF scomp IS
TYPE STATE_TYPE IS ( reset_pc, fetch, decode, execute_add, execute_load, execute_store,

```

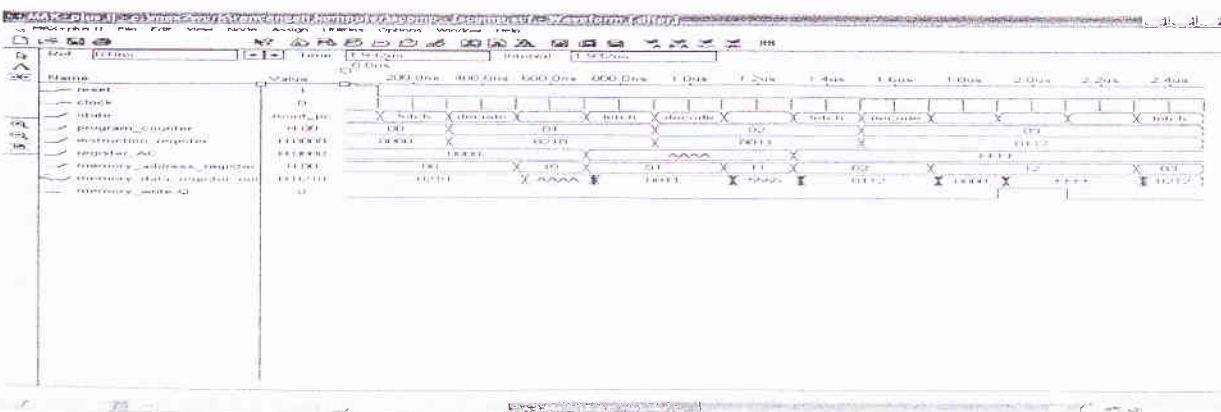
Gambar 9. Deklarasi Register-register Internal

Setelah *clock signal* berikutnya, kontrol berpindahan pada keadaan eksekusi dari spesifik instruksi yang diberikan. Beberapa instruksi dapat dieksekusi dalam satu *clock cycle* dan dapat juga lebih dari satu *clock cycle*. Instruksi-instruksi yang ditulis di memori akan memerlukan lebih dari satu *state* untuk eksekusi karena ada penekanan waktu dari memori. Ini dapat ditunjukkan pada instruksi STORE, alamat memori dan data memerlukan kestabilan sebelum dan sesudah *signal* penulisan memori dibuat tinggi. Oleh karena diperlukan penambahan *state* yang digunakan untuk menghindari kegagalan akses memori dan waktu

yang tidak digunakan. Ketika masing-masing instruksi selesai pada keadaan eksekusi, MAR yang diisi dari PC (MAR=PC) untuk mulai pengambilan instruksi berikutnya.

6.3. Simulasi Rancangan Komputer Sederhana

Rancangan komputer sederhana ini disimulasikan secara perangkat lunak pada perangkat simulator Altera Max+Plus II dan simulasi pada perangkat keras Altera UP2 berbasis FPGA FLEX10K EPF10K70RC240-4, seperti pada Gambar 10.



Gambar 10. Simulasi Rancangan Komputer Sederhana dengan Altera Max+Plus II

Simulasi diatas menunjukkan keadaan signal pewaktu dari program dengan instruksi LOAD, ADD, dan STORE dengan nilai-nilai data yang sesuai dengan contoh program menghitung $A=B+C$. Hasil akhir menunjukkan bahwa memori berisi nilai dari A yang sesuai. Program ini berakhir dengan instruksi JUMP yang akan mengulang program pada alamat 04, terus menerus.

VII. Penutup

7.1. Kesimpulan

Dari perancangan dan implementasi pada UP2 Board untuk komputer sederhana dapat diambil kesimpulan yaitu :

- Rancangan komputer sederhana dapat diimplementasikan dalam bentuk perangkat keras pada FPGA Altera FLEX 10K EPF10K70RC240-4.
- Dalam perancangan prosesor diperlukan pemahaman siklus operasi dari setiap instruksi, organisasi perangkat keras, bus signal control, dan pewaktu.
- Beberapa operasi dapat dilakukan secara paralel.
- Perangkat simulator Altera Max+Plus II, dapat digunakan untuk melakukan analisa terhadap rancangan sebelum diimplementasikan.

- Statement VHDL untuk komputer sederhana didasarkan pada impelemen-tasi state mesin dari siklus pengambilan (*fetch*), pengkodean (*decode*) dan eksekusi (*execute*).

7.2. Saran

Untuk mengembangkan komputer sederhana yang telah dirancang ini, ada beberapa hal yang disarankan

- Instruksi dari komputer sederhana ini, dapat dibuat sebanyak 256.
- State mesin dari setiap siklus harus dibuat secara sistematis dan teliti.

Daftar Pustaka:

1. Altera Max+Plus II, "Getting Started", 1997, Altera Corporation
2. James O. Hamblean And Michael E. Furman, "Rapid Prototyping of Digital Systems : A tutorial approach", Kluwer Academic Publishers, 2001.
3. Richard Tinder, "Digital Engineering Design: A Modern Approach", Prentice-Hall, Inc, 1997
4. William Stallings, "Organisasi dan Arsitektur Komputer: Perancangan Kinerja", Edisi Indonesi, Prentice-Hall, 1996