BAB 1

PENDAHULUAN

1.1 Latar Belakang

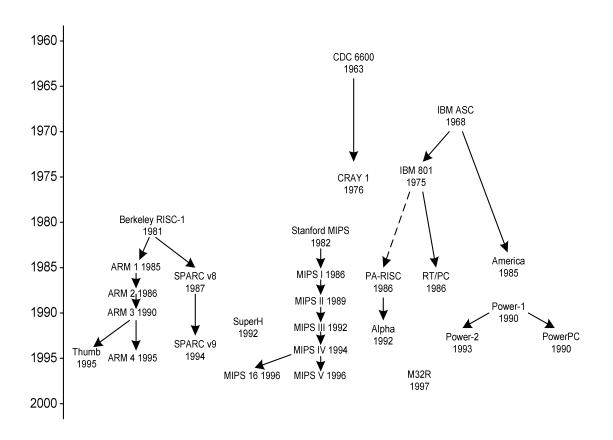
Perkembangan bahasa pemrograman yang semakin pesat memicu lahirnya Bahasa Tingkat Tinggi (*High Level Language*) yang memungkinkan pengekspresian algoritma program menjadi lebih singkat. Hal ini menimbulkan masalah yang disebut *semantic gap*, yaitu perbedaan antara operasi-operasi yang disediakan oleh Bahasa Tingkat Tinggi (BTT) dengan operasi-operasi yang disediakan oleh arsitektur komputer. Masalah ini mengakibatkan terjadinya eksekusi program tidak efisien, program mesin menjadi berukuran besar, dan kompleksitas kompiler meningkat. Untuk mengurangi kesenjangan ini, para perancang prosesor memperbanyak jumlah instruksi, mode pengalamatan, dan jenis *statement* BTT yang diimplementasikan di dalam perangkat keras. Dengan jumlah instruksi yang banyak, maka prosesor dengan arsitektur yang demikian ini disebut CISC (*Complex Instruction Set Computer*). (Stallings, 1997, p121)

Kesulitan yang timbul pada arsitektur CISC adalah pada saat pengoptimalan kode-kode instruksi yang di bangun pada prosesornya. Hal ini mengakibatkan sulitnya meminimalkan ukuran kode tersebut, mengurangi durasi waktu dalam mengeksekusi instruksi, dan meningkatkan *pipelining*. (Stallings, 1997, p134)

Maka pada awal tahun 1980, sebuah kelompok di Universitas California di Berkeley, yang dipimpin oleh David Patterson dan Carlo Sequin mulai merancang prosesor dengan arsitektur RISC (*Reduced Instruction Set Computer*)

yang diberi nama RISC1. Pada waktu yang bersamaan melalui San Francisco Buy di Universitas Stanford, Profesor John Hennessy yang pada saat itu mulai mengembangkan MIPS (*Microprocessor without Interlocking Pipeline Stages*) yang menjadi bacaan dan ide dalam pengembangan prosesor RISC pertama. Industri komputer IBM (*International Business Machine*) muncul dengan IBM 801. (http://cse.stanford.edu/class/sophomore-college/projects-00/risc/mips/index.html, 2003)

Gambar 1.1 menunjukkan silsilah prosesor RISC.



Gambar 1.1 Garis Silsilah Mikroprosesor dengan Arsitektur RISC

Hasil penelitian mendalam berkenaan dengan *semantic gap* telah memberikan inspirasi kepada para peneliti untuk mencari pendekatan secara menyeluruh: yakni untuk membuat arsitektur yang mendukung BTT secara lebih sederhana, bukannya lebih sulit sehingga melahirkan prosesor dengan arsitektur RISC. (Stallings, 1997, p121)

Prosesor dengan arsitektur RISC memiliki tiga karakteristik utama, yaitu *register* dalam jumlah yang besar, perancangan *pipeline* instruksi, dan waktu eksekusi instruksi dalam satu siklus. (http://cse.stanford.edu/class/sophomore-college/projects-00/risc/whatis/index.html, 2003)

Pada arsitektur RISC kesulitan perancangan perangkat keras dilimpahkan pada perancangan perangkat lunaknya. Hal ini dilakukan karena kemajuan teknik kompilasi dalam perangkat lunak yang sangat membantu untuk tujuan tersebut.

Selain pesatnya kemajuan bidang perangkat lunak dewasa ini, kemajuan teknologi pembuatan komponen (perangkat keras) juga terjadi. Hal ini ditunjukkan dengan pesatnya perkembangan kerapatan dalam perancangan arsitektur IC (*Integrated Circuit*), sehingga memungkinkan perancangan *Programmable Logic Device* (PLD) sampai jutaan transistor yang dikenal dengan teknologi *Very Large Scale Integrated* (VLSI). Pendekatan desain VLSI ini menggunakan *gate array*, yang merupakan pola-pola gerbang yang dibangun pada bahan silikon dan berulang sampai ribuan kali, sehingga seluruh *chip* yang dibentuknya mengandung gerbang-gerbang yang sama. Pendekatan baru yang memungkinkan PLD dengan kapasitas yang besar biasanya disebut *Complex Programmable Logic Device* (CPLD) atau *Field-Programmable Gate Array* (FPGA). (Mano dan Kime, 2001, p326)

Dalam perancangannya, sebuah prosesor sederhana akan dibangun berdasarkan pada arsitektur RISC dan diimplementasikan menggunakan FPGA.

1.2 Ruang Lingkup

Ruang lingkup dalam perancangan ini adalah sebagai berikut:

- ◆ Perancangan organisasi (register file, function unit, control unit, pipeline dan interrupt handling) dan arsitektur (jenis-jenis operasi/instruksi, tipe data, format instruksi, register, dan mode pengalamatan) RISC.
- ◆ Mengumpulkan data mengenai kecepatan dan jumlah CLB (*Configurable Logic Block*) yang dibutuhkan untuk membangun prosesor dengan arsitektur RISC pada FPGA.
- Perancangan menggunakan software Xilinx Integrated Software Environment
 (ISE) WebPack seri 5.2i.
- ♦ Implementasi menggunakan FPGA Xilinx Spartan 2 XC2S200-PQ208.

1.3 Tujuan dan Manfaat

Skripsi ini ditujukan sebagai sumber pembelajaran dan penelitian akan perancangan dan pengembangan sistem RISC dan implementasinya menggunakan FPGA pada lingkungan pendidikan. Hal ini dikarenakan pesatnya pengembangan dan penelitian di bidang *embeded system* yang umumnya menggunakan arsitektur RISC. Selain itu ditujukan juga untuk meningkatkan pemahaman dan penggunaan FPGA di lingkungan pendidikan, khususnya Jurusan Sistem Komputer Universitas Bina Nusantara.

Manfaat ke dalam berupa pengalaman mendesain dan membuat arsitektur RISC, memahami lebih jauh tentang arsitektur RISC, dan dengan diimplementasikannya pada FPGA, manfaat yang didapat berupa pengalaman dalam menggunakan FPGA.

Selain itu dijabarkan juga organisasi dan arsitektur RISC serta diberikan data mengenai kecepatan dan kapasitas yang dibutuhkan dalam mengimplementasikan arsitektur RISC pada FPGA sehingga manfaat ke luar berupa informasi yang digunakan untuk mengembangkan sistem RISC lebih jauh.

1.4 Sumber Literatur

Sudah terdapat banyak arsitektur RISC yang dirancang dan dikembangkan menggunakan FPGA, baik merupakan penelitian di bidang pendidikan (terdapat beberapa *sorce code* yang diberikan secara gratis) maupun pengembangan oleh perusahaan-perusahaan yang dijual komersial. Beberapa diantaranya adalah sebagai berikut:

- J32: Dibuat oleh Jan Gray pada tahun 1995. Arsitektur RISC 32 bit klasik (32 register dan 3 operand). Pipeline 4 tahap (IF/RF/EX/WB). On-chip peripheral (bus 32 bit, boot ROM (Read Only Memory), UART, DRAM control). (Gray, 2003)
- XSOC/xr16: Dibuat oleh Jan Gray pada tahun 1998. Menggunakan arsitektur RISC 16 bit. Pipeline sederhana. Bekerja pada frekuensi 33 MHz. Bus dan peripheral berada di dalam chip. (Gray, 1999)

- dari *microcontroller* 8 bit sederhana. Kode binernya cocok dengan *microcontroller* 16C57 produksi Microchip Technology, Inc. Berbagai *tool* yang tersedia untuk pengembangan *software* untuk arsitektur ini menjadikannya menarik untuk tujuan pendidikan dan digunakan pada FPGA atau ASIC (*Application Specific Integrated Circuit*). Tersedia juga *testbench* dan beberapa contoh program. Ditulis oleh Tom Coonan dan disumbangkan ke Free-IP Project pada tahun 2000. (http://www.free-ip.com/risc8/index.html, 2003)
- Free-6502: Berada di bawah Free-IP Project. Cocok dengan 6502 CPU core.

 100% ditulis dengan VHDL (VHSIC (Very High Speed Integrated Circuit)

 Hardware Description Language). Antarmuka bus 8 bit sinkron yang sederhana. Seluruhnya didesain secara sinkron. Registrasi I/O (Input/Output)

 untuk kemudahan penempatan dan integrasi. Mendukung semua instruksi standar 6502. (http://www.free-ip.com/6502/index.html, 2003)
- HC11: Dikeluarkan oleh Green Mountain Computing Systems, Inc. pada tahun 2000. Ditulis sepenuhnya menggunakan VHDL. Membutuhkan 1076 slices dengan frekuensi 31MHz pada Xilinx Virtex 400E dan 2142 LE (Logic Element) dengan frekuensi 32MHz pada Altera APEX 20K100. (http://www.gmvhdl.com/hc11core.html, 2003)
- PicoBlaze: Dikeluarkan oleh Xilinx. Merupakan microcontroller 8 bit.
 Memiliki 16 general-purpose register. Dikeluarkan pada Mei 1999.
 (http://www.xilinx.com/ipcenter/processor_central/picoblaze/, 2003)

- MicroBlaze: Dikeluarkan oleh Xilinx. Prosesor RISC standar 32 bit yang mendukung 32 bit dan 16 bit lebar bus. Memiliki 32 buah register.
 (http://www.xilinx.com/xlnx/xil_prodcat_product.jsp?title=microblaze, 2003)
- Nios: Dikeluarkan oleh Altera. Mendukung 32 bit dan 16 bit variasi arsitektur. Menggunakan 16 bit instruksi. Dapat memasukkan hingga 512 general-purpose register. Menggunakan arsitektur memori Harvard. Pipeline 5 tahap. (http://www.altera.com/, 2003)
- MSL16: Mirip dengan komputer dengan jumlah instruksi minimal yang dijabarkan pada halaman mikroprosesor *Jeff Fox's Forth* tetapi diimplementasikan pada Xilinx FPGA. Jika sistem ini sudah stabil, maka kode sumber VHDL untuk CPU dan semua kodenya akan dikeluarkan sebagai *public domain* CPU yang lengkap. (http://www.cse.cuhk.edu.hk/~phwl/msl16/msl16.html, 2003)
- Silicore SLC1657: Microcontroller RISC 8 bit. Diterapkan pada FPGA dengan VHDL. Digunakan pada aplikasi embedded control seperti: sensor, peralatan kesehatan, pemakai elektronik, sistem otomotif, telekomunikasi, militer, dan kontrol industri. (http://www.silicore.net/, 2003)

1.5 Metodologi Perancangan

Langkah nyata yang akan diterapkan untuk metodologi dalam perancangan adalah menggunakan metodologi umum pembuatan sebuah produk yang biasa digunakan dalam bidang teknik rekayasa atau *engineering* sebagai berikut (Priest, 1998):

a. Batasan (requirement) prosesor

Sesuai dengan tujuan dan manfaat maka prosesor yang akan dibuat harus menggunakan FPGA dengan arsitektur RISC. Jumlah dan jenis instruksi serta jenis data dari prosesor tersebut harus dapat mewakili secara umum dasar sebuah prosesor berarsitektur RISC. Peningkatan kompleksitas prosesor dapat dikembangkan secara bertahap melalui proyek oleh mahasiswa lain yang tertarik pada perancangan RISC menggunakan FPGA.

b. Konsep disain (conceptual design) prosesor

Desain akan dikelompokan dan dibangun berdasarkan fungsi dari masing-masing bagian karena penambahan, pengurangan dan pengembangan komponen dapat dilakukan dengan mudah. Beberapa perancangan seperti SOC/xr16 menggunakan desain berdasarkan fungsi. (Gray, 1999)

c. Detail disain (detailed design) prosesor

Untuk arsitektur, desain dilakukan berdasarkan tahap-tahap rancangan set instruksi, yaitu menentukan jenis instruksi, jenis data, format instruksi, *register*, dan mode pengalamatan.

Sedangkan untuk organisasinya, desain akan dikelompokan menjadi dua bagain utama, yaitu perancangan *datapath* dan *control unit*, dimana perancangan *datapath* dibagi lagi menjadi perancangan *register file* dan *function unit*. *Control unit* berfungsi mengatur aliran data dan instruksi, *register file* terdiri dari beberapa *register* yang berfungsi menampung data dan alamat, serta *function unit* yang disebut juga *Arithmetic Logic Unit* (ALU) berfungsi melakukan pengoperasian data.

Kemudian desain akan diimplementasikan menggunakan modul D2 yang terdiri dari Xilinx FPGA Spartan 2 XC2S200-PQ208 dengan penghasil *clock* 50 MHz dan modul D2IO untuk peralatan *input/output*-nya yang terdiri dari Xilinx CPLD XC95108-PC84, tombol, saklar, dan LED (*Lighting Emmitting Diode*).

d. Pengujian dan evaluasi prosesor

Pengujian dapat dilakukan baik secara *software* maupun *hardware*. Secara *software* dilakukan dengan program simulasi dan secara *hardware* dilakukan dengan membuat program-program pengujian dan hasilnya ditampilkan pada LED.

Sumber-sumber lain untuk penerapan metodologi diperoleh dari internet, buku, dosen, asisten laboratorium perangkat keras, dan teman-teman.

1.6 Sistematika Penulisan

Skripsi ini disusun dalam lima bab sebagai berikut :

Bab satu Pendahuluan, berisi latar belakang pemilihan judul, yaitu dengan membandingkan beberapa teori, hasil penelitian dan parameter bersangkutan lainnya. Melakukan survey literatur tentang kegiatan sejenis yang ada sampai saat ini. Pada bab ini akan dijabarkan pula ruang lingkup dan batasan-batasan yang akan dibahas. Berisi tujuan dan manfaat yang diharapkan serta metodologi untuk mencapai tujuan.

Bab dua Landasan Teori, berisi pemahaman mengenai teori atau metode yang nantinya akan digunakan sebagai bahan analisa, perancangan, dan pengambilan kesimpulan. Beberapa pemahaman yang akan disajikan dibagi menjadi beberapa kelompok besar, yaitu pembahasan mengenai arsitektur RISC, sekilas tentang VHDL, FPGA dan Xilinx.

Bab tiga Strategi Perancangan, berisi tentang analisa yang dilakukan berdasarkan landasan teori dan kemudian dipilih beberapa teori yang digunakan dalam melakukan perancangan, baik perancangan arsitektur maupun organisasi.

Bab empat Implementasi dan Evaluasi, dimana hasil rancangan prosesor RISC pada Bab 3 akan diimplementasikan pada FPGA. Untuk evaluasinya, dibuat beberapa program penguji dan digunakan *switch* dan tombol sebagai masukkan dan LED sebagai keluaran untuk melihat apakah keluaran yang dihasilkan sesuai dengan yang diiginkan.

Bab lima Kesimpulan dan Saran, berisi beberapa bagian-bagian penting yang dihasilkan dimana akan diberikan sebagai kesimpulan, beberapa masalah atau kendala dan pemecahan yang ditemukan serta beberapa poin untuk dikembangkan lebih lanjut akan diberikan sebagai saran.