RISC,CISC, DAN PROSESOR SUPERSCALAR

Created by

NAMA: FIRDAUS SURYA P

NRP/KELAS: 7107040041/2 EB D4

ELECTRONIC ENGINEERING POLYTECHNIC INSTITUTE OF SURABAYA

Kata Pengantar

Kebutuhan akan ilmu tentang elektronika sangatlah dibutuhkan di era sekarang ini, tidak terkecuali pelajar,mahasiswa,businessman,maupun ibu rumah tangga. Di jaman serba teknologi sekarang ini telah mengubah sendi kehidupan manusia, dimana manusia sekarang menyebutnya sebagai era modern. Tak dapat dihindarkan berbagai teknologi canggih pun telah diciptakan guna mempermudah pekerjaan manusia, menambah ilmu pengetahuan, atau hanya sekedar hiburan belaka. Dan semuanya itu telah di lengkapai dengan berbagai fasilitas dari vendor yang memegang teknologi tersebut, dengan berbagai iklan tentang murahnya biaya pemakaian hingga kemampuan untuk dapat di akses dimanapun.

Akibatnya, badai teknologi inipun juga menunutu para produsen untuk lebih berlomba-lomba dengan bentukan produsen lain, apalagi kalau bukan untuk merebut pasar, demikian ditinjau dari segi ekonomi kemasyarakatan, namun jauh dasi sisi itu yaitu dari sisi teknologi sendiri memungkinkan berkembangnya piranti pendukung teknologi yang juga arusnya disadari atau tidak semakin cepat pula. Tak terkecuali komputer yang disinyalir sebagai piranti paling penting, kompeten dan strategis dalam kemajuan dunia teknologi informasi seperti era sekarang ini.

Dalam makalah ini, saya sebagai penulis menyajikan beberapa ulasan singkat tentang beberapa piranti dalam komputer . Meskipun hanya ulasan singkat, tapi saya berharap makalah ini bias bermanfaat bagi pembaca blog saya, dan para netter semua. Dan mohon postingnya jangan lupa.Trims.

BAB I

PENDAHULUHAN

1.1 Latar Belakang

Terdapat dua prosesor yang saat ini telah kita kenal, yaitu RISC (*Reduce Instruction Set Komputer*) dan CISC (*Complex Instruction Set Computer*). Prosesor CISC memiliki instruksi-instruksi kompleks untuk memudahkan penulisan program bahasa assembly, sedangkan prosesor RISC memiliki instruksi-instruksi sederhana yang dapat dieksekusi dengan cepat untuk menyederhanakan implementasi rangkaian kontrol internal prosesor. Karenanya, prosesor RISC dapat dibuat dalam luasan keping semikonduktor yang relatif lebih sempit dengan jumlah komponen yang lebih sedikit dibanding prosesor CISC. Perbedaan orientasi di antara kedua prosesor ini menyebabkan adanya perbedaan sistem secara keseluruhan, termasuk juga perancangan kompilatornya. Sedangkan prosesor Superscalar umumnya menggunakan beberapa unit fungsional, menciptakan jalur paralel di mana berbagai instruksi yang berbeda dapat dieksekusi secara paralel. Dengan pengaturan tersebut, maka dimungkinkan untuk memulai eksekusi beberapa instruksi secara paralel tiap siklus detak. Tentu saja, eksekusi paralel harus mempertahankan kebenaran logikan program, sehingga hasil yang diperoleh harus sama dengan hasil dari eksekusi secara serial

BAB II

ISI

2.1 Pengertian

2.1.1 CISC (*Complex Instructions Set Computer*) , RISC (Reduce Instructions Set Computer) dan Superscalar

CISC adalah singkatan dari *Complex Intruction Set Computer* dimana prosesor tersebut memiliki set instruksi yang kompleks dan lengkap. CISC sendiri adalah salah satu bentuk arsitektur yang menjalani beberapa instruksi dengan tingkat yang rendah. Misalnya intruksi tingakt rendah tersebut adalah operasi aritmetika, penyimpanan-pengambilan dari memory dll.

CISC memang memiliki instruksi yang complex dan memang dirasa berpengaruh pada kinerjanya yang lebih lambat. CISC menawarkan set intruksi yang powerful, kuat, tangguh, maka tak heran jika CISC memang hanya mengenal bahasa asembly yang sebenarnya ia tujukan bagi para programmer. Oleh karena itu ,CISC hanya memerlukan sedikit instruksi untuk berjalan.

Sistem mikrokontroler selalu terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat lunak ini merupakan deretan perintah atau instruksi yang dijalankan oleh prosesor secara sekuensial. Instruksi itu sendiri sebenarnya adalah bit-bit logik 1 atau 0 (biner) yang ada di memori program. Angka-angka biner ini jika lebarnya 8 bit disebut byte dan jika 16 bit disebut word. Deretan logik biner inilah yang dibaca oleh prosesor sebagai perintah atau instruksi. Supaya lebih singkat, angka biner itu biasanya direpresentasikan dengan bilangan hexa (HEX). Tetapi bagi manusia, menulis program dengan angka biner atau hexa sungguh merepotkan. Sehingga dibuatlah bahasa assembler yang direpresentasikan dengan penyingkatan kata-kata yang cukup dimengerti oleh manusia.

Bahasa assembler ini biasanya diambil dari bahasa Inggris dan presentasinya itu disebut dengan *Mnemonic*. Masing-masing pabrik mikroprosesor melengkapi *chip* buatannya dengan set instruksi yang akan dipakai untuk membuat program.

Biner	Hexa	Mnemonic
10110110	B6	LDAA ...
10010111	97	STAA ...
01001010	4A	DECA ...
10001010	8A	ORAA ...
00100110	26	BNE ...
00000001	01	NOP...

01111110 7E JMP ...

Jadi sebenarnya Tujuan utama dari arsitektur CISC adalah melaksanakan suatu perintah cukup dengan beberapa baris bahasa mesin sedikit mungkin. Hal ini bisa tercapai dengan cara membuat perangkat keras prosesor mampu memahami dan menjalankan beberapa rangkaian operasi. Untuk tujuan contoh kita kali ini, sebuah prosesor CISC sudah dilengkapi dengan sebuah instruksi khusus, yang kita beri nama MULT. Saat dijalankan, instruksi akan membaca dua nilai dan menyimpannya ke 2 register yag berbeda, melakukan perkalian operan di unit eksekusi dan kemudian mengambalikan lagi hasilnya ke register yang benar. Jadi instruksi-nya cukup satu saja

Sedangkan RISC adalah singkatan dari *Reduced Instruction Set Computer* yang artinya prosesor tersebut memiliki set instruksi program yang lebih sedikit. Karena perbedaan keduanya ada pada kata set instruksi yang kompleks atau sederhana (*reduced).* RISC lahir pada pertengahan 1980, kelahirannya ini dilator belakangi untuK CISC. Perbedaan mencolok dari kelahiran RISC ini adalah tidak ditemui pada dirinya instruksi assembly atau yang dikenal dengan bahasa mesin sedangkan itu banyak sekali di jumpai di CISC.

Konsep arsitektur RISC banyak menerapkan proses eksekusi pipeline.

Meskipun jumlah perintah tunggal yang diperlukan untuk melakukan pekerjaan yang diberikan mungkin lebih besar, eksekusi secara pipeline memerlukan waktu yang lebih singkat daripada waktu untuk melakukan pekerjaan yang sama dengan menggunakan perintah yang lebih rumit.

Mesin RISC memerlukan memori yang lebih besar untuk mengakomodasi program yang lebih besar.

IBM 801 adalah prosesor komersial pertama yang menggunakan pendekatan RISC.

Lebih lanjut untuk memahami RISC, diawali dengan tinjauan singkat tentang karakteristik eksekusi instruksi.

Aspek komputasi yang ditinjau dalam merancang mesin RISC adalah sbb.:

>>Operasi-operasi yang dilakukan:

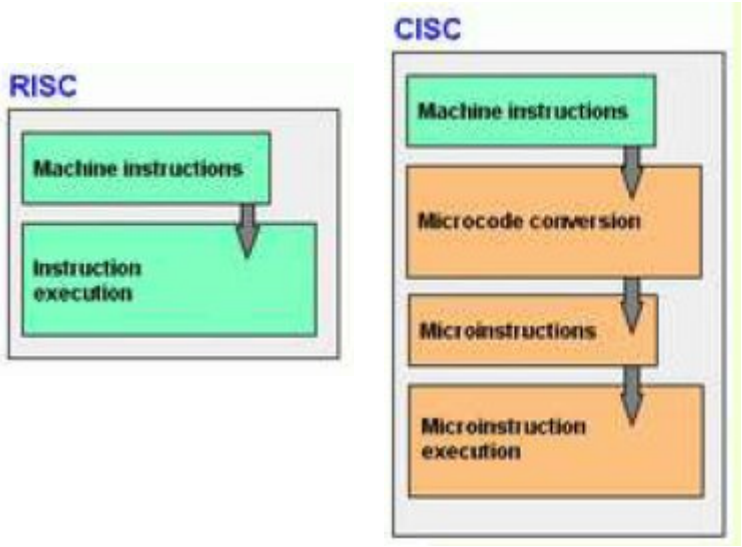
Hal ini menentukan fungsi-fungsi yang akan dilakukan oleh CPU dan interaksinya dengan memori.

>> Operand-operand yang digunakan:

Jenis-jenis operand dan frekuensi pemakaiannya akan menentukan organisasi memori untuk menyimpannya dan mode pengalamatan untuk mengaksesnya.

>> Pengurutan eksekusi:

Hal ini akan menentukan kontrol dan organisasi pipeline.



Salah satu jenis dari arsitektur, dimana superscalar adalah sebuah uniprocessor yang dapat mengeksekusi dua atau lebih operasi scalar dalm bentuk paralel. Merupakan salah satu rancangan untuk meningkatkan kecepatan CPU. Kebanyakan dari komputer saat ini menggunakan mekanisme superscalar ini. Standar pipeline yang digunakan adalah untuk pengolahan bilangan matematika integer (bilangan bulat, bilangan yang tidak memiliki pecahan), kebanyakan CPU juga memiliki kemampuan untuk pengolahan untuk data floating point (bilangan berkoma). Pipeline yang mengolah integer dapat juga digunakan untuk mengolah data bertipe floating point ini, namun untuk aplikasi tertentu, terutama untuk aplikasi keperluan ilmiah CPU yang memiliki kemampuan pengolahan floating point dapat meningkatkan kecepatan prosesnya secara dramatis.

Peristiwa menarik yang bisa dilakukan dengan metoda superscalar ini adalah dalam hal memperkirakan pencabangan instruksi (brach prediction) serta perkiraan eksekusi perintah (speculative execution). Peristiwa ini sangat menguntungkan buat program yang membutuhkan pencabangan dari kelompok intruksi yang dijalankankannya.

Program yang terdiri dari kelompok perintah bercabang ini sering digunakan dalam pemrograman. Contohnya dalam menentukan aktifitas yang dilakukan oleh suatu sistem berdasarkan umur seseorang yang sedang diolahnya, katakanlah jika umur yang bersangkutan lebih dari 18 tahun, maka akan diberlakukan instruksi yang berhubungan dengan umur tersebut, anggaplah seseorang tersebut dianggap telah dewasa, sedangkan untuk kondisi lainnya dianggap belum dewasa. Tentu perlakuannya akan dibedakan sesuai dengan sistem yang sedang dijalankan.

Lalu apa yang dilakukan oleh CPU untuk hal ini? Komputer akan membandingkan nilai umur data yang diperolehnya dengan 18 tahun sehingga komputer dapat menentukan langkah dan

sikap yang harus diambilnya berdasarkan hasil perbandingan tersebut. Sikap yang diambil tentu akan diambil berdasarkan pencabangan yang ada.

Pada CPU yang mendukung perintah pencabangan ini, CPU membutuhkan lumayan banyak clock cycle, mengingat CPU menempatkan semuanya pada pipeline dan menemukan perintah berikutnya yang akan dieksekusinya. Sirkuit untuk branch prediction melakukan pekerjaan ini bekerja sama dengan pipeline, yang dilakukan sebelum proses di ALU dilaksanakan, dan memperkirakan hasil dari pencabangan tersebut.

Jika CPU berfikir bahwa branch akan menuju suatu cabang, biasanya berdasarkan pekerjaan sebelumnya, maka perintah berikutnya sudah dipersiapkan untuk dieksekusi berikut data-datanya, bahkan dengan adanya pipeline ini, bila tidak diperlukan suatu referensi dari instruksi terakhir, maka bisa dilaksanakan dengan segera, karena data dan instruksi yang dibutuhkan telah dipersiapkan sebelumnya..

Dalam hal speculative execution, artinya CPU akan menggunakan melakukan perhitungan pada pipeline yang berbeda berdasarkan kemungkinan yang diperkirakan oleh komputer. Jika kemungkinan yang dilakukan oleh komputer tepat, maka hasilnya sudah bisa diambil langsung dan tinggal melanjutkan perintah berikutnya, sedangkan jika kemungkinan yang diperkirakan oleh komputer tidak tepat, maka akan dilaksanakan kemungkinan lain sesuai dengan logika instruksi tersebut.

Teknik yang digunakan untuk pipeline dan superscalar ini bisa melaksanakan branch prediction dan speculative execution tentunya membutuhkan ekstra transistor yang tidak sedikit untuk hal tersebut.

Sebagai perbandingan, komputer yang membangkitkan pemrosesan pada PC pertama yang dikeluarkan oleh IBM pada mesin 8088 memiliki sekitar 29.000 transistor. Sedangkan pada mesin Pentium III, dengan teknologi superscalar dan superpipeline, mendukung branch prediction, speculative execution serta berbagai kemampuan lainnya memiliki sekitar 7,5 juta transistor. Beberapa CPU terkini lainnya seperti HP 8500 memiliki sekitar 140 juta transistor.

2.2 Perbedaan karakteristik CISC dan RISC serta SUPERSCALAR

CISC dan RISC perbedaannya tidak signifikan jika hanya dilihat dari terminologi set instruksinya yang kompleks atau tidak (*reduced*). Lebih dari itu, RISC dan CISC berbeda dalam filosofi arsitekturnya. Filosofi arsitektur CISC adalah memindahkan kerumitan *software* ke dalam *hardware*. Teknologi pembuatan IC saat ini memungkinkan untuk menamam ribuan bahkan jutaan transistor di dalam satu *dice*. Bermacam-macam instruksi yang mendekati bahasa pemrogram tingkat tinggi dapat dibuat dengan tujuan untuk memudahkan *programmer* membuat programnya. Beberapa prosesor CISC umumnya memiliki *microcode* berupa *firmware* internal di dalam *chip*-nya yang berguna untuk menterjemahkan instruksi makro. Mekanisme ini bisa memperlambat eksekusi instruksi, namun efektif untuk membuat instruksi-instruksi yang kompleks. Untuk aplikasi-aplikasi tertentu yang membutuhkan *singlechip* komputer, prosesor CISC bisa menjadi pilihan.

Karakteristik CISC versus RISC

 Rancangan RISC dapat memperoleh keuntungan dengan mengambil sejumlah feature CISC dan Rancangan CISC dapat memperoleh keuntungan dengan mengambil sejumlah feature RISC.

 Hasilnya adalah bahwa sejumlah rancangan RISC yang terbaru, yang dikenal sebagai PowerPC, tidak lagi “murni” RISC dan rancangan CISC yang terbaru, yang dikenal sebagai Pentium, memiliki beberapa karakteristik RISC.

Ciri-ciri RISC:

 Instruksi berukuran tunggal

 Ukuran yang umum adalah 4 byte.

 Jumlah mode pengalamatan data yang sedikit, biasanya kurang dari lima buah.

 Tidak terdapat pengalamatan tak langsung.

 Tidak terdapat operasi yang menggabungkan operasi load/store dengan operasi aritmetika (misalnya, penambahan dari memori, penambahan ke memori).

Sebaliknya, filosofi arsitektur RISC adalah arsitektur prosesor yang tidak rumit dengan membatasi jumlah instruksi hanya pada instruksi dasar yang diperlukan saja. Kerumitan membuat program dalam bahasa mesin diatasi dengan membuat bahasa program tingkat tinggi dan *compiler* yang sesuai. Karena tidak rumit, teorinya mikroprosesor RISC adalah mikroprosesor yang *low-cost* dalam arti yang sebenarnya. Namun demikian, kelebihan ruang pada prosesor RISC dimanfaatkan untuk membuat sistem-sistem tambahan yang ada pada prosesor modern saat ini. Banyak prosesor RISC yang di dalam *chip*-nya dilengkapi dengan sistem *superscalar*, *pipelining*, *caches memory*, register-register dan sebagainya, yang tujuannya untuk membuat prosesor itu menjadi semakin cepat.

Sudah sering kita mendengar debat yang cukup menarik antara komputer personal IBM dan kompatibelnya yang berlabel Intel Inside dengan komputer Apple yang berlabel PowerPC. Perbedaan utama antara kedua komputer itu ada pada tipe prosesor yang digunakannya. Prosesor PowerPC dari Motorola yang menjadi otak utama komputer Apple Macintosh dipercaya sebagai prosesor RISC, sedangkan Pentium buatan Intel diyakini sebagai prosesor CISC. Kenyataannya komputer personal yang berbasis Intel Pentium saat ini adalah komputer personal yang paling banyak populasinya. Tetapi tidak bisa pungkiri juga bahwa komputer yang berbasis RISC seperti Macintosh, SUN adalah komputer yang handal dengan sistem *pipelining*, *superscalar*, operasi *floating point* dan sebagainya.

Tersedia dari peningkatan kinerja superscalar teknik dibatasi oleh dua bidang utama:

• Tingkat dari hakiki paralel dalam instruksi streaming, yakni terbatasnya jumlah instruksi level parallelism, dan

• Kompleksitas waktu dan biaya yang terkait memberangkatkan dan ketergantungan memeriksa logika.

Binari yang ada telah dijalankan program tahap hakiki paralel. Dalam beberapa kasus petunjuk tidak tergantung pada satu sama lain dan dapat dijalankan secara bersamaan. Dalam kasus lain mereka yang antar-tergantung: satu instruksi dampak baik sumber daya atau hasil lainnya. Petunjuk yang = b + c; d = e + f dapat berjalan secara bersamaan karena tidak ada yang bergantung pada hasil perhitungan lain. Namun, petunjuk yang = b + c; d = a + f mungkin tidak akan runnable secara paralel, tergantung pada urutan petunjuk yang lengkap saat mereka bergerak melalui unit.

Bila jumlah yang dikeluarkan secara simultan petunjuk meningkat, biaya memeriksa dependensi meningkat sangat pesat. Hal ini diperparah oleh kebutuhan untuk memeriksa dependensi di waktu dan menjalankan di CPU jam menilai. Ini termasuk biaya tambahan gerbang logika diperlukan untuk melaksanakan pemeriksaan, dan waktu tunda yang melalui pintu. Penelitian menunjukkan pintu

gerbang biaya dalam beberapa kasus dapat NK pintu, dan biaya keterlambatan k2logn, dimana n adalah jumlah instruksi pada prosesor's set instruksi, dan k adalah jumlah bersamaan menurunkan petunjuk. Dalam matematika, ini disebut sebagai combinatoric masalah melibatkan permutations.

Meski mungkin berisi instruksi streaming tidak antar-instruksi dependensi, superscalar CPU yang sebenarnya harus memeriksa bahwa kemungkinan, karena tidak ada jaminan lain dan kegagalan untuk mendeteksi suatu dependensi akan menghasilkan hasil yang salah.

Tidak peduli bagaimana lanjutan proses yang semikonduktor atau cara cepat kecepatan yang berpindah, ini tempat yang praktis membatasi berapa petunjuk dapat menurunkan secara bersamaan. Meskipun proses kemajuan akan mengijinkan pernah lebih besar jumlah unit fungsional (misalnya, ALUs), beban instruksi memeriksa dependensi sehingga tumbuh pesat yang dicapai superscalar dispatch batas relatif kecil. - Kemungkinan pada urutan lima hingga enam secara bersamaan menurunkan petunjuk.

Namun akhirnya tak terhingga cepat memeriksa ketergantungan pada logika konvensional yang lain superscalar CPU, jika instruksi streaming itu sendiri memiliki banyak dependensi, ini juga akan membatasi speedup mungkin. Dengan demikian tingkat hakiki paralel dalam kode streaming bentuk kedua keterbatasan.

Karakteritik	CISC			RISC		Superskalar	
	IBM 370/168	VAX 11/780	Intel 80486	Motorola 88000	MIPS R4000	IBM RS/System 6000	Intel 80960
Tahun dibuat	1973	1978	7989	1988	7991	1990	1989
Jumlah instruksi	208	303	235	51	94	184	62
Instruksi (Bytes)	2-6	2-57	1-11	4	32	4	4,8
Mode Pengalamatan	4	22	11	3	1	2	11
Jumlah register general-purpose	16	16	8	32	32	32	23-256
Ukuran memori kontrol (Kbits)	420	480	246	-	-	-	-
Ukuran Cache (Kbytes)	64	64	8	16	128	32-64	0,5

BAB III

PENUTUP

Diantara kelebihan dan kekurangan dari arsitektur RISC dan arsitektur CISC sampai sekarang masih menjadi sebuah perdebatan. Ada juga teknologi yang menggabungkan kedua arsitektur tersebut, contohnya : Prosesor Intel dan AMD yang dijual secara komersil sekarang adalah pengembangan dari prosesor x86 yang menggunakan basis prosesor CISC. Lucunya, instruksi set yang didukung oleh kedua prosesor tersebut menggunakan instruksi RISC yang lebih efisien dalam menangani data.

DAFTAR PUSTAKA

>>http://www.electroniclab.com/index.php?option=com_content&view=article&id=30:cisc-vs-risc&catid=9:labmikro&Itemid=11

>>http://www.total.or.id/info.php?kk=Superscalar

>>file:///H:/arkom/Perbandingan%20RISC%20dan%20CISC.htm

>>file:///H:/arkom/i_yha_%20Processor%20Superscalar.htm