

PERTEMUAN 16

RISC DAN CISC

Perbedaan RISC dan CISC

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. Apakah memang RISC lebih lebih baik dari CISC atau sebaliknya. Tetapi tahukah kita dimana sebenarnya letak perbedaan itu. Apakah prosesor dengan instruksi yang lebih sedikit akan lebih baik dari prosesor yang instruksinya kompleks dan lengkap. Apakah memang perbedaan prosesor itu hanya dari banyak atau tidaknya instruksi saja. Bukankah jumlah instruksi tidak berhubungan dengan ke-handal-an suatu prosesor. Pertanyaan-pertanyaan ini yang hendak dijawab melalui tulisan berikut.

Dari segi kecepatannya, Reduced Instruction Set Computer (**RISC**) **lebih cepat** dibandingkan dengan Complex Instruction Set Computer (**CISC**). Ini dikarenakan selain instruksi-instruksi pada RISC lebih mudah untuk diproses, RISC menyederhanakan instruksi. Jumlah instruksi yang dimiliki oleh prosesor RISC kebanyakan berjumlah puluhan ($\pm 30-70$). Contoh: COP8 buatan National Semiconductor memiliki 58 instruksi.

Sedangkan untuk prosesor CISC jumlahnya sudah dalam ratusan (± 100 atau lebih). CISC dirancang untuk meminimumkan jumlah perintah yang diperlukan untuk mengerjakan pekerjaan yang diberikan (Jumlah perintah sedikit tetapi rumit). Konsep CISC menjadikan mesin mudah untuk diprogram dalam bahasa rakitan, tetapi konsep ini menyulitkan dalam penyusunan kompilasi bahasa pemrograman tingkat tinggi.

Dalam CISC banyak terdapat perintah bahasa mesin. Eksekusi Instruksi RISC Waktu eksekusi dapat dirumuskan dengan:

$$\text{Waktu eksekusi} = N \times S \times T$$

Dengan: **N** adalah jumlah perintah.

S adalah jumlah rata-rata langkah per perintah.

T adalah waktu yang diperlukan untuk melaksanakan satu langkah.

- Kecepatan eksekusi dapat ditingkatkan dengan menurunkan nilai dari ketiga variabel di atas.
- Arsitektur CISC berusaha menurunkan nilai N (jumlah perintah), sedangkan

- Arsitektur RISC berusaha menurunkan nilai S dan T.
- Proses pipeline dapat digunakan untuk membuat nilai efektif S mendekati 1 (satu) artinya komputer menyelesaikan satu perintah dalam satu siklus waktu CPU.
- Nilai T dapat diturunkan dengan merancang perintah yang sederhana.

A. CISC (Complex Instruction Set Computing)

Complex Instruction Set Computer (CISC) adalah sebuah arsitektur dari set instruksi dimana setiap instruksi akan menjalankan beberapa operasi tingkat rendah, seperti pengambilan dari memory, operasi aritmetika, dan penyimpanan ke dalam memory, semuanya sekaligus hanya di dalam sebuah instruksi. 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 yang berbeda, melakukan perkalian operasi di unit eksekusi dan kemudian mengembalikan lagi hasilnya ke register yang benar.

Karakteristik CISC yg "sarat informasi" ini memberikan keuntungan di mana ukuran program-program yang dihasilkan akan menjadi relatif lebih kecil, dan penggunaan memory akan semakin berkurang. Karena CISC inilah biaya pembuatan komputer pada saat itu (tahun 1960) menjadi jauh lebih hemat. Memang setelah itu banyak desain yang memberikan hasil yang lebih baik dengan biaya yang lebih rendah, dan juga mengakibatkan pemrograman level tinggi menjadi lebih sederhana, tetapi pada kenyataannya tidaklah selalu demikian.

Contoh-contoh prosesor CISC adalah : System/360, VAX, PDP-11, varian Motorola 68000 , dan CPU AMD dan Intel x86.

1. Karakteristik CISC

Sarat informasi memberikan keuntungan di mana ukuran program-program yang dihasilkan akan menjadi relatif lebih kecil, dan penggunaan memory akan semakin berkurang. Karena CISC inilah biaya pembuatan komputer pada saat itu (tahun 1960) menjadi jauh lebih hemat

Dimaksudkan untuk meminimumkan jumlah perintah yang diperlukan untuk mengerjakan pekerjaan yang diberikan. (Jumlah perintah sedikit tetapi rumit) Konsep CISC menjadikan mesin mudah untuk diprogram dalam bahasa rakitan.

2. Ciri-ciri

- Jumlah instruksi banyak.

- Banyak terdapat perintah bahasa mesin.
- Instruksi lebih kompleks.
- Pengaplikasian CISC yaitu pada AMD dan Intel.

B. RISC (Reduced Instruction Set Computer)

RISC Reduced Instruction Set Computing atau "Komputasi set instruksi yang disederhanakan. Merupakan sebuah arsitektur komputer atau arsitektur komputasi modern dengan instruksi-instruksi dan jenis eksekusi yang paling sederhana. Biasanya digunakan pada komputer berkinerja tinggi seperti komputer vektor.

Bahasa pemrograman memungkinkan programmer dapat mengekspresikan algoritma lebih singkat, lebih memperhatikan rincian, dan mendukung penggunaan pemrograman terstruktur, tetapi ternyata muncul masalah lain yaitu semantic gap, yaitu perbedaan antara operasi-operasi yang disediakan oleh HLL dengan yang disediakan oleh arsitektur komputer, ini ditandai dengan ketidakefisienan eksekusi, program mesin yang berukuran besar, dan kompleksitas kompiler. **Pengaplikasian RISC** yaitu pada CPU Apple

Set-set instruksi yang kompleks tersebut dimaksudkan untuk :

1. Memudahkan pekerjaan kompiler
2. Meningkatkan efisiensi eksekusi, karena operasi yang kompleks dapat diimplementasikan didalam mikrokode.
3. Memberikan dukungan bagi HLL yang lebih kompleks dan canggih.

1. Sejarah RISC

Proyek RISC pertama dibuat oleh IBM, Stanford dan UC –Berkeley pada akhir tahun 70 dan awal tahun 80an. IBM 801, Stanford MIPS, dan Berkeley RISC 1 dan 2 dibuat dengan konsep yang sama sehingga dikenal sebagai RISC.

2. RISC mempunyai karakteristik :

- one cycle execution time : satu putaran eksekusi. Prosessor RISC mempunyai CPI (clock per instruction) atau waktu per instruksi untuk setiap putaran. Hal ini dimaksud untuk mengoptimalkan setiap instruksi pada CPU.
- pipelining: adalah sebuah teknik yang memungkinkan dapat melakukan eksekusi secara simultan. Sehingga proses instruksi lebih efisien.
- large number of registers: Jumlah register yang sangat banyak. RISC di Desain dimaksudkan untuk dapat menampung jumlah register yang sangat banyak untuk mengantisipasi agar tidak terjadi interaksi yang berlebih dengan memory.

3. Ciri-ciri

- Instruksi berukuran tunggal.
- Ukuran yang umum adalah 4 byte.
- Jumlah pengalamatan data sedikit.
- Tidak terdapat pengalamatan tak langsung.
- Tidak terdapat operasi yang menggabungkan operasi load/store dengan operasi aritmatika.
- Tidak terdapat lebih dari satu operand beralamat memori per instruksi.
- Tidak mendukung perataan sembarang bagi data untuk operasi load/ store.
- Jumlah maksimum pemakaian memori manajemen bagi suatu alamat data adalah sebuah instruksi.

C. Perbandingan antara RISC dengan CISC

Cara sederhana untuk melihat kekurangan dan kelebihan dari CISC dan RISC adalah dengan membandingkannya secara langsung. Pada tahap perbandingan ini dicoba dengan menghitung perkalian dua bilangan dalam memori. Memori tersebut terbagi menjadi beberapa lokasi yang diberi nomor 1(baris): 1 (kolom) hingga 6:4. Unit eksekusi bertanggung-jawab untuk semua operasi komputasi. Namun, unit eksekusi hanya

beroperasi untuk data yang sudah disimpan kedalam salah satu dari 6 register (A, B, C, D, E atau F). Misalnya, kita akan melakukan perkalian (product) dua angka, satu di simpan di lokasi 2:3 sedangkan lainnya di lokasi 5:2, kemudian hasil perkalian tersebut dikembalikan lagi ke lokasi 2:3.

1. Menggunakan Pendekatan RISC

Prosesor RISC hanya menggunakan instruksi-instruksi sederhana yang bisa dieksekusi dalam satu siklus. Dengan demikian, instruksi 'MULT' sebagaimana dijelaskan sebelumnya dibagi menjadi tiga instruksi yang berbeda, yaitu "LOAD", yang digunakan untuk memindahkan data dari memori kedalam register, "PROD", yang digunakan untuk melakukan operasi produk (perkalian) dua operand yang berada di dalam register (bukan yang ada di memori) dan "STORE", yang digunakan untuk memindahkan data dari register kembali ke memori. Berikut ini adalah urutan instruksi yang harus dieksekusi agar yang terjadi sama dengan instruksi "MULT" pada prosesor RISC (dalam 4 baris bahasa mesin):

- LOAD A, 2:3
- LOAD B, 5:2
- PROD A, B
- STORE 2:3, A

2. Menggunakan Pendekatan CISC

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.

Sebuah prosesor CISC sudah dilengkapi dengan sebuah instruksi khusus, yang diberi nama MULT. Saat dijalankan, instruksi akan membaca dua nilai dan menyimpannya ke 2 register yang berbeda, melakukan perkalian operand di unit eksekusi dan kemudian mengembalikan lagi hasilnya ke register yang benar. Jadi instruksi-nya cukup satu saja.

- MULT 2:3, 5:2

D. Perbedaan RISC dengan CISC dilihat dari segi instruksinya :

1. RISC (Reduced Instruction Set Computer).

- Menekankan pada perangkat lunak, dengan sedikit transistor.
- Instruksi sederhana bahkan single.

- Load / Store atau memory ke memory bekerja terpisah.
- Ukuran kode besar dan kecepatan lebih tinggi.
- Transistor didalamnya lebih untuk meregister memori.

2. CISC (Complex Instruction Set Computer).

- Lebih menekankan pada perangkat keras, sesuai dengan takdirnya untuk pragramer.
- Memiliki instruksi komplek. Load / Store atau Memori ke Memori bekerjasama.
- Memiliki ukuran kode yang kecil dan kecepatan yang rendah.
- Transistor di dalamnya digunakan untuk menyimpan instruksi – instruksi bersifat komplek.

Contoh-contoh RISC dan CISC:

RISC :

1. Komputer vektor
2. Mikroprosesor Intel 960
3. Itanium (IA64) dari Intel Corporation
4. Power PC dari International Business Machine, dll.

CISC :

1. Prosesor system/360
2. Prosesor VAX
3. Prosesor PDP-11
4. CPU AMD
5. Intel x86, dll.

Kesimpulan

CISC Complex Instruction Set Computer sedangkan RISC merupakan kepanjangan dari Reduced Instruction Set Computer. Chip RISC dibangun mulai pertengahan tahun 1980 sebagai pengganti chip CISC. Pada dasarnya karakteristik CISC yg "sarat informasi" memberikan keuntungan di mana ukuran program-program yang dihasilkan akan menjadi relatif lebih kecil, dan penggunaan memory akan semakin berkurang. Hal inilah yang menyebabkan komputer-komputer pada saat itu memiliki harga yang murah.

Filosofi RISC berada dalam tidak satu pun chip yang menggunakan bahasa instruksi assembly yang complex, seperti yang digunakan di CISC. Untuk itulah, instruksi yang simple dan lebih cepat akan lebih baik daripada besar, complex dan lambat seperti CISC. Keuntungan RISC lainnya karena adanya instruksi yang simple, maka chip RISC hanya memiliki beberapa transistor, yang akan membuat RISC mudah didesain dan murah untuk diproduksi untuk menulis compiler yang powerful. RISC memberikan kemudahan di hardware, namun lebih kompleks di software.

DAFTAR PUSTAKA

<http://rachmatsn.blogspot.com/2013/05/cisc-dan-risc-pada-saat-iniada-dua.html>

<http://maizarti.wordpress.com/2011/02/28/perbedaan-antara-risc-dan-cisc/>

<http://www.slideshare.net/MugiwaraHaqiem/risc-dan-cisc>

<http://id.wikipedia.org/wiki/RISC>