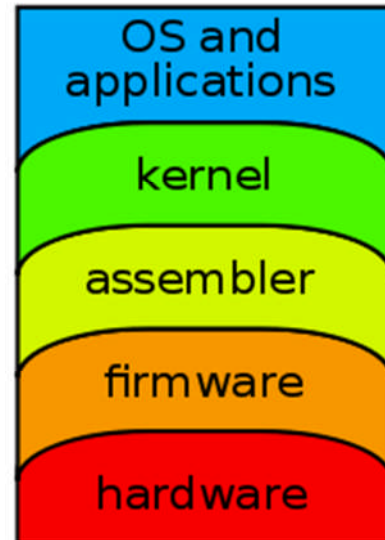


Arsitektur komputer

Abstraksi dari sebuah arsitektur komputer dan hubungannya dengan bagian perangkat keras, *firmware*, *assembler*, kernel, sistem operasi, dan perangkat lunak aplikasinya

Dalam bidang teknik komputer, **arsitektur komputer** adalah konsep perencanaan dan struktur pengoperasian dasar dari suatu sistem komputer. Arsitektur komputer ini merupakan rencana cetak-biru dan deskripsi fungsional dari kebutuhan bagian perangkat keras yang didesain (kecepatan proses dan sistem interkoneksinya). Dalam hal ini, implementasi perencanaan dari masing-masing bagian akan lebih difokuskan terutama, mengenai bagaimana CPU akan bekerja, dan mengenai cara pengaksesan data dan alamat dari dan ke memori cache, RAM, ROM, cakram keras, dll). Beberapa contoh dari arsitektur komputer ini adalah arsitektur von Neumann, CISC, RISC, blue Gene, dll.



Arsitektur komputer juga dapat didefinisikan dan dikategorikan sebagai ilmu dan sekaligus seni mengenai cara interkoneksi komponen-komponen perangkat keras untuk dapat menciptakan sebuah komputer yang memenuhi kebutuhan fungsional, kinerja, dan target biayanya.

Arsitektur komputer ini paling tidak mengandung 3 sub-kategori:

- Set instruksi (ISA)
- Arsitektur mikro dari ISA, dan
- Sistem desain dari seluruh komponen dalam perangkat keras komputer ini

Set Instruksi –ISA

ISA (bahasa Inggris: **Instruction Set**, atau **Instruction Set Architecture** (ISA)) didefinisikan sebagai suatu aspek dalam arsitektur komputer yang dapat dilihat oleh para pemrogram. Secara umum, ISA ini mencakup jenis data yang didukung, jenis instruksi yang dipakai, jenis register, mode pengalamatan, arsitektur memori, penanganan interupsi, eksepsi, dan operasi I/O eksternalnya (jika ada).

ISA merupakan sebuah spesifikasi dari kumpulan semua kode-kode biner (*opcode*) yang diimplementasikan dalam bentuk aslinya (*native form*) dalam sebuah desain prosesor tertentu. Kumpulan *opcode* tersebut, umumnya disebut sebagai **bahasa mesin** (*machine language*) untuk ISA yang bersangkutan. ISA yang populer digunakan adalah set instruksi untuk chip Intel x86, IA-64, IBM PowerPC, Motorola 68000, Sun SPARC, DEC Alpha, dan lain-lain.

ISA kadang-kadang digunakan untuk membedakan kumpulan karakteristik yang disebut di atas dengan mikroarsitektur prosesor, yang merupakan kumpulan teknik desain prosesor untuk mengimplementasikan set instruksi (mencakup *microcode*, *pipeline*, *sistem cache*, manajemen daya, dan lainnya). Komputer-komputer dengan mikroarsitektur berbeda dapat saling berbagi set instruksi yang sama. Sebagai contoh, prosesor Intel Pentium dan prosesor AMD Athlon mengimplementasikan versi yang hampir identik dari set instruksi Intel x86, tetapi jika ditinjau dari desain internalnya, perbedaannya sangat radikal. Konsep ini dapat diperluas untuk ISA-ISA yang unik seperti TIMI yang terdapat dalam IBM System/38 dan IBM IAS/400. TIMI merupakan sebuah ISA yang diimplementasikan sebagai perangkat lunak level rendah yang berfungsi sebagai mesin virtual. TIMI didesain untuk meningkatkan masa hidup sebuah platform dan aplikasi yang ditulis untuknya, sehingga mengizinkan platform tersebut agar dapat dipindahkan ke perangkat keras yang sama sekali berbeda tanpa harus memodifikasi perangkat lunak (kecuali yang berkaitan dengan TIMI). Hal ini membuat IBM dapat memindahkan platform AS/400 dari arsitektur mikroprosesor CISC ke arsitektur mikroprosesor POWER tanpa harus menulis ulang bagian-bagian dari dalam sistem operasi atau perangkat lunak yang diasosiasikan dengannya.

Ketika mendesain mikroarsitektur, para desainer menggunakan Register Transfer Language (RTL) untuk mendefinisikan operasi dari setiap instruksi yang terdapat dalam ISA.

Sebuah ISA juga dapat diemulasikan dalam bentuk perangkat lunak oleh sebuah interpreter. Karena terjadi translasi tambahan yang dibutuhkan untuk melakukan emulasi, hal ini memang menjadikannya lebih lambat jika dibandingkan dengan menjalankan program secara langsung di atas perangkat keras yang mengimplementasikan ISA tersebut. Akhir-akhir ini, banyak vendor ISA atau mikroarsitektur yang baru membuat perangkat lunak emulator yang dapat digunakan oleh para pengembang perangkat lunak sebelum implementasi dalam bentuk perangkat keras dirilis oleh vendor.

Daftar ISA di bawah ini tidak dapat dikatakan komprehensif, mengingat banyaknya arsitektur lama yang tidak digunakan lagi saat ini atau adanya ISA yang baru dibuat oleh para desainer.

ISA yang diimplementasikan dalam bentuk perangkat keras

- Alpha AXP (DEC Alpha)

- ARM (Acorn RISC Machine) (Advanced RISC Machine now ARM Ltd)
- IA-64 (Itanium/Itanium 2)
- MIPS
- Motorola 68k
- PA-RISC (HP Precision Architecture)
- IBM POWER
- IBM PowerPC
- SPARC
- SuperH (Hitachi)
- System/360
- Tricore (Infineon)
- Transputer (STMicroelectronics)
- VAX (Digital Equipment Corporation)
- x86 (IA-32, Pentium, Athlon) (AMD64, EM64T)

ISA yang diimplementasikan dalam bentuk perangkat lunak lalu dibuat perangkat kerasnya

- p-Code (UCSD p-System Version III on Western Digital Pascal Micro-Engine)
- Java virtual machine (ARM Jazelle, PicoJava)
- FORTH

ISA yang tidak pernah diimplementasikan dalam bentuk perangkat keras

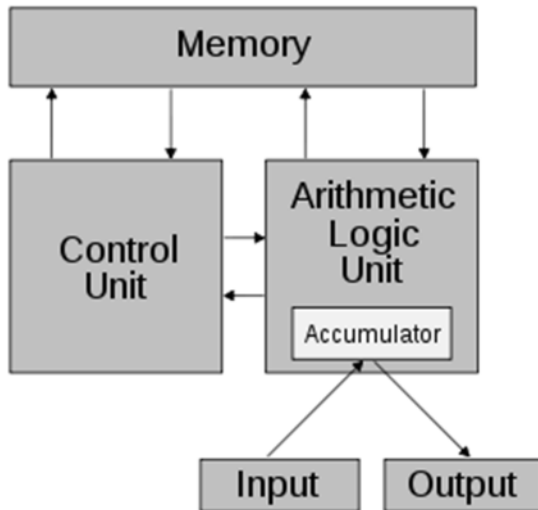
- SECD machine
- ALGOL Object Code

Arsitektur von Neumann

(Mesin Von Neumann) adalah arsitektur yang diciptakan oleh John von Neumann (1903-1957).

Arsitektur ini digunakan oleh hampir semua komputer saat ini.

- Arsitektur Von Neumann menggambarkan komputer dengan empat bagian utama: Unit Aritmatika dan Logis (ALU), unit kontrol, memori, dan alat masukan dan hasil (secara kolektif dinamakan I/O). Bagian ini dihubungkan oleh berkas kawat, "bus"



Arsitektur Von Neuman

Von Neumann diabadikan namanya sebagai arsitektur von Neumann, yaitu arsitektur komputer yang banyak digunakan di sebagian besar sistem komputer non paralel, karena dialah yang pertama kali mempublikasikan konsep tersebut. Meski konsep ini kemudian dikembangkan oleh J. Presper Eckert dan John William Mauchly dalam pengembangan komputer ENIAC, nama von Neumann lah yang lebih dikenal sebagai penemu arsitektur komputer tersebut.



John von Neumann pada tahun 1940-an.

John von Neumann (Neumann János) (lahir di Budapest, Hungaria, 28 Desember 1903 – meninggal di Washington DC, Amerika Serikat, 8 Februari 1957 pada umur 53 tahun) adalah seorang matematikawan dari Hungaria-Jerman yang memberikan kontribusi penting di bidang fisika kuantum, analisis fungsional, teori himpunan, ilmu komputer, ekonomi dan bidang lainnya yang berkaitan dengan matematika. Von

Neumann adalah pionir komputer digital modern dan penerapan teori operator di bidang mekanika kuantum.

CISC

Complex instruction-set computing atau **Complex Instruction-Set Computer (CISC;** "Kumpulan instruksi komputasi kompleks") 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. Karakteristik CISC dapat dikatakan bertolak-belakang dengan RISC. Sebelum proses RISC didesain untuk pertama kalinya, banyak arsitek komputer mencoba menjembatani celah semantik", yaitu bagaimana cara untuk membuat set-set instruksi untuk mempermudah pemrograman level tinggi dengan menyediakan instruksi "level tinggi" seperti pemanggilan procedure, proses pengulangan dan mode-mode pengalamatan kompleks sehingga struktur data dan akses array dapat dikombinasikan dengan sebuah instruksi. 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. Contohnya, arsitektur kompleks yang didesain dengan kurang baik (yang menggunakan kode-kode mikro untuk mengakses fungsi-fungsi hardware), akan berada pada situasi di mana akan lebih mudah untuk meningkatkan performansi dengan tidak menggunakan instruksi yang kompleks (seperti instruksi pemanggilan procedure), tetapi dengan menggunakan urutan instruksi yang sederhana.

Satu alasan mengenai hal ini adalah karena set-set instruksi level-tinggi, yang sering disandikan (untuk kode-kode yang kompleks), akan menjadi cukup sulit untuk diterjemahkan kembali dan dijalankan secara efektif dengan jumlah transistor yang terbatas. Oleh karena itu arsitektur - arsitektur ini memerlukan penanganan yang lebih terfokus pada desain prosesor. Pada saat itu di mana jumlah transistor cukup terbatas, mengakibatkan semakin sempitnya peluang ditemukannya cara-cara alternatif untuk optimisasi perkembangan prosesor. Oleh karena itulah, pemikiran untuk menggunakan desain RISC muncul pada pertengahan tahun 1970 (Pusat Penelitian Watson IBM 801 - IBMs)

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

Istilah RISC dan CISC saat ini kurang dikenal, setelah melihat perkembangan lebih lanjut dari desain dan implementasi baik CISC dan RISC. Implementasi CISC paralel untuk pertama

kalinya, seperti 486 dari Intel, AMD, Cyrix, dan IBM telah mendukung setiap instruksi yang digunakan oleh prosesor-prosesor sebelumnya, meskipun efisiensi tertingginya hanya saat digunakan pada subset x86 yang sederhana (mirip dengan set instruksi RISC, tetapi tanpa batasan penyimpanan/pengambilan data dari RISC). Prosesor-prosesor modern x86 juga telah menyangkikan dan membagi lebih banyak lagi instruksi-instruksi kompleks menjadi beberapa "operasi-mikro" internal yang lebih kecil sehingga dapat instruksi-instruksi tersebut dapat dilakukan secara paralel, sehingga mencapai performansi tinggi pada subset instruksi yang lebih besar

RISC

Reduced Instruction Set Computing (RISC) atau "Komputasi set instruksi yang disederhanakan" pertama kali digagas oleh John Cocke, peneliti dari IBM di Yorktown, New York pada tahun 1974 saat ia membuktikan bahwa sekitar 20% instruksi pada sebuah prosesor ternyata menangani sekitar 80% dari keseluruhan kerjanya. Komputer pertama yang menggunakan konsep RISC ini adalah IBM PC/XT pada era 1980-an. Istilah RISC sendiri pertama kali dipopulerkan oleh David Patterson, pengajar pada University of California di Berkely

Definisi

RISC, yang jika diterjemahkan berarti "Komputasi Kumpulan Instruksi yang Disederhanakan", merupakan sebuah arsitektur komputer atau arsitektur komputasi modern dengan instruksi-instruksi dan jenis eksekusi yang paling sederhana. Arsitektur ini digunakan pada komputer dengan kinerja tinggi, seperti komputer vektor. Selain digunakan dalam komputer vektor, desain ini juga diimplementasikan pada prosesor komputer lain, seperti pada beberapa mikroprosesor Intel 960, Itanium (IA64) dari Intel Corporation, Alpha AXP dari DEC, R4x00 dari MIPS Corporation, PowerPC dan Arsitektur POWER dari International Business Machine. Selain itu, RISC juga umum dipakai pada Advanced RISC Machine (ARM) dan StrongARM (termasuk di antaranya adalah Intel XScale), SPARC dan UltraSPARC dari Sun Microsystems, serta PA-RISC dari Hewlett-Packard.

Selain RISC, desain *Central Processing Unit* yang lain adalah CISC (*Complex Instruction Set Computing*), yang jika diterjemahkan ke dalam Bahasa Indonesia berarti Komputasi Kumpulan Instruksi yang kompleks atau rumit.

Blue Gene

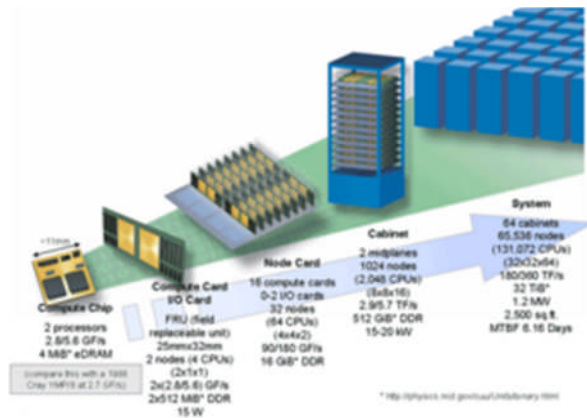


Blue Gene/L

Blue Gene adalah sebuah arsitektur komputer yang dirancang untuk menciptakan beberapa superkomputer generasi berikut, yang dirancang untuk mencapai kecepatan operasi petaflop (1 peta = 10 pangkat 15), dan pada 2005 telah mencapai kecepatan lebih dari 100 teraflop (1 tera = 10 pangkat 12). Blue Gene merupakan proyek antara Departemen Energi Amerika Serikat (yang membiayai proyek ini),

industri (terutama IBM), dan kalangan akademi. Ada lima proyek Blue Gene dalam pengembangan saat ini, di antaranya adalah Blue Gene/L, Blue Gene/C, dan Blue Gene/P.

Blue Gene/L



Arsitektur Blue Gene/L

Komputer pertama dalam seri Blue Gene. **Blue Gene/L** dikembangkan melalui sebuah "partnership" dengan Lawrence Livermore National Laboratory menghabiskan biaya AS\$100 juta dan direncanakan dapat mencapai

kecepatan ratusan TFLOPS, dengan kecepatan puncak teoritis 360 TFLOPS. Ini hampir sepuluh kali lebih cepat dari Earth Simulator, superkomputer tercepat di dunia sebelum Blue Gene. Pada Juni 2004, dua prototipe Blue Gene/L masuk dalam peringkat 500 besar superkomputer berada dalam posisi ke-4 dan ke-8.

Pada 29 September 2004 IBM mengumumkan bahwa sebuah prototipe Blue Gene/L di IBM Rochester (Minnesota) telah menyusul Earth Simulator NEC sebagai komputer tercepat di dunia, dengan kecepatan 36,01 TFLOPS, mengalahkan Earth Simulator yang memiliki kecepatan 35,86 TFLOPS. Mesin ini kemudian mencapai kecepatan 70,72 TFLOPS.

Pada 24 Maret 2005, Departemen Energi AS mengumumkan bahwa Blue Gene/L memecahkan rekor komputer tercepat mencapai 135,5 TFLOPS. Hal ini dimungkinkan karena menambah jumlah rak menjadi 32 dengan setiap rak berisi 1.024 node komputasi. Ini masih merupakan setengah dari konfigurasi final yang direncanakan mencapai 65.536 node.

Pada 27 Oktober, 2005, Lawrence Livermore National Laboratory dan IBM mengumumkan bahwa Blue Gene/L sekali lagi telah menciptakan rekor dengan mengalahkan rekornya sendiri setelah mencapai kecepatan 280.6 TFLOPS.