


# RISC VS CISC

Arsitektur Komputer

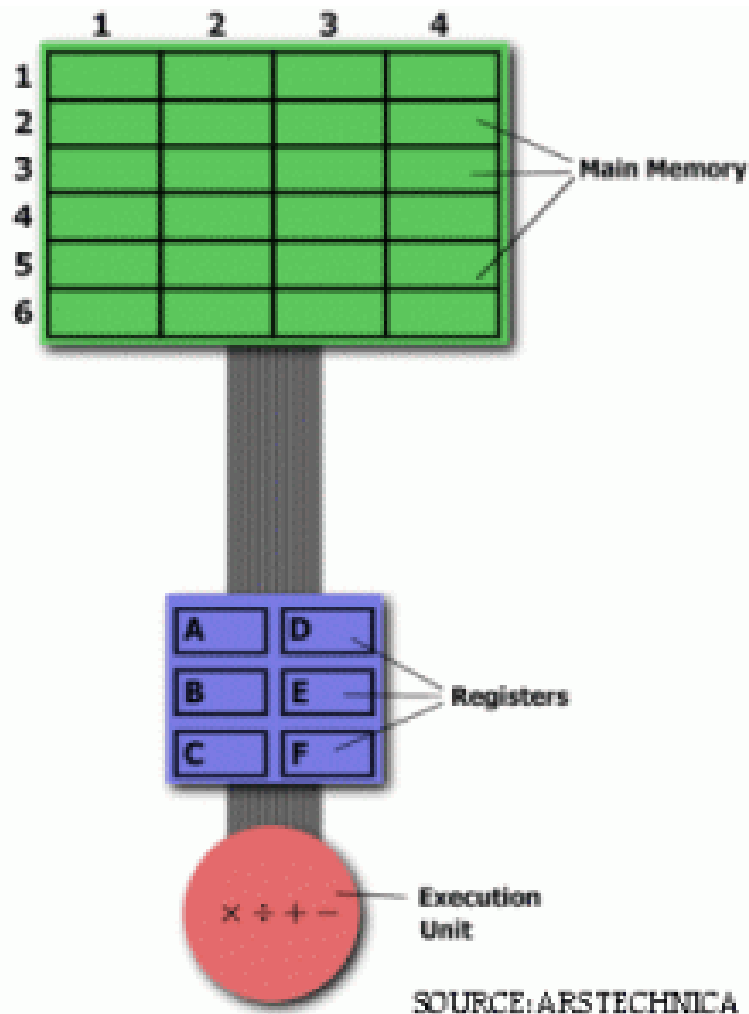
# Pendahuluan



- RISC singkatan dari Reduced Instruction Set Computer.
- Merupakan bagian dari arsitektur mikroprosesor, berbentuk kecil dan berfungsi untuk mengeset instruksi dalam komunikasi diantara arsitektur yang lainnya.
- 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 Barkeley RISC 1 dan 2 dibuat dengan konsep yang sama sehingga dikenal sebagai RISC

- 
- Cara sederhana untuk melihat kelebihan dan kelemahan dari arsitektur RISC (Reduced Instruction Set Computers) adalah dengan langsung membandingkannya dengan arsitektur pendahulunya yaitu CISC (Complex Instruction Set Computers).

# Perkalian Dua Bilangan dalam Memori




- Pada bagian kiri terlihat sebuah struktur memori (yang disederhanakan) suatu komputer secara umum.
- 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-data yang sudah disimpan ke dalam salah satu dari 6 register (A, B, C, D, E atau F).
- Misalnya, kita akan melakukan perkalian (product) dua angka, satu disimpan di lokasi 2:3 sedangkan lainnya di lokasi 5:2, kemudian hasil perkalian tersebut dikembalikan lagi ke lokasi 2:3.

# 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. 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 operan di unit eksekusi dan kemudian mengembalikan lagi hasilnya ke register yang benar. Jadi instruksi-nya cukup satu saja...

MULT 2:3, 5:2


- MULT dalam hal ini lebih dikenal sebagai “complex instruction”, atau instruksi yang kompleks. Bekerja secara langsung melalui memori komputer dan tidak memerlukan instruksi lain seperti fungsi baca maupun menyimpan.

- 
- Satu kelebihan dari sistem ini adalah kompailer hanya menerjemahkan instruksi-instruksi bahasa tingkat-tinggi ke dalam sebuah bahasa mesin. Karena panjang kode instruksi relatif pendek, hanya sedikit saja dari RAM yang digunakan untuk menyimpan instruksi-instruksi tersebut.


# 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 ke dalam register, "PROD", yang digunakan untuk melakukan operasi produk (perkalian) dua operan 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
```

- 
- Awalnya memang terlihat kurang efisien, hal ini dikarenakan semakin banyak baris instruksi, semakin banyak lokasi RAM yang dibutuhkan untuk menyimpan instruksi-instruksi tersebut. Kompailer juga harus melakukan konversi dari bahasa tingkat tinggi ke bentuk kode instruksi 4 baris tersebut.



- 
- Bagaimanapun juga, strategi pada RISC memberikan beberapa kelebihan. Karena masing-masing instruksi hanya membutuhkan satu siklus detak untuk eksekusi, maka seluruh program (yang sudah dijelaskan sebelumnya) dapat dikerjakan setara dengan kecepatan dari eksekusi instruksi "MULT".
  - Secara perangkat keras, prosesor RISC tidak terlalu banyak membutuhkan transistor dibandingkan dengan CISC, sehingga menyisakan ruangan untuk register-register serbaguna (general purpose registers). Selain itu, karena semua instruksi dikerjakan dalam waktu yang sama (yaitu satu detak), maka dimungkinkan untuk melakukan pipelining.

- 
- Memisahkan instruksi “LOAD” dan “STORE” sesungguhnya mengurangi kerja yang harus dilakukan oleh prosesor. Pada CISC, setelah instruksi “MULT” dieksekusi, prosesor akan secara otomatis menghapus isi register, jika ada operan yang dibutuhkan lagi untuk operasi berikutnya, maka prosesor harus menyimpan-ulang data tersebut dari memori ke register. Sedangkan pada RISC, operan tetap berada dalam register hingga ada data lain yang disimpan ke dalam register yang bersangkutan.

# RISC vs CISC

CISC	RISC
Penekanan pada perangkat keras	Penekanan pada perangkat lunak
Termasuk instruksi kompleks multi-clock	Single-clock, hanya sejumlah kecil instruksi
Memori-ke-memori: "LOAD" dan "STORE" saling bekerjasama	Register ke register: "LOAD" dan "STORE" adalah instruksi2 terpisah
Ukuran kode kecil, kecepatan rendah	Ukuran kode besar, kecepatan (relatif) tinggi
Transistor digunakan untuk menyimpan instruksi2 kompleks	Transistor banyak dipakai untuk register memori

# Persamaan Unjuk-kerja (Performance)

- Persamaan berikut biasa digunakan sebagai ukuran unjuk-kerja suatu komputer:

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$


- Pendekatan CISC bertujuan untuk meminimalkan jumlah instruksi per program, dengan cara mengorbankan kecepatan eksekusi sekian silus/detik. Sedangkan RISC bertolak belakang, tujuannya mengurangi jumlah siklus/detik setiap instruksi dibayar dengan bertambahnya jumlah instruksi per program.

# Penghadang jalan (Roadblocks) RISC

- Walaupun pemrosesan berbasis RISC memiliki beberapa kelebihan, dibutuhkan waktu kurang lebih 10 tahunan mendapatkan kedudukan di dunia komersil. Hal ini dikarenakan kurangnya dukungan perangkat lunak.
- Walaupun Apple's Power Macintosh menggunakan chip berbasis RISC dan Windows NT adalah kompatibel RISC, Windows 3.1 dan Windows 95 dirancang berdasarkan prosesor CISC. Banyak perusahaan segan untuk masuk ke dalam dunia teknologi RISC. Tanpa adanya ketertarikan komersil, pengembang prosesor RISC tidak akan mampu memproduksi chip RISC dalam jumlah besar sedemikian hingga harganya bisa kompetitif.

# Karakteristik RISC

- RISC mempunyai karakteristik :
  - One cycle execution time : satu putaran eksekusi → Processor 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 berlebihan dengan memory.

- 
- Rangkaian instruksi built-in pada processor yang terdiri dari perintah-perintah yang lebih ringkas dibandingkan dengan CISC.
  - RISC memiliki keunggulan dalam hal kecepatannya sehingga banyak digunakan untuk aplikasi-aplikasi yang memerlukan kalkulasi secara intensif.
  - Dewasa ini, RISC digunakan pada keluarga processor buatan Motorola (PowerPC) dan SUN Microsystems (Sparc, UltraSparc).
  - RISC dikembangkan melalui seorang penelitinya yang bernama John Cocke, beliau menyampaikan bahwa sebenarnya kekhasan dari komputer tidaklah menggunakan banyak instruksi, namun yang dimilikinya adalah instruksi yang kompleks yang dilakukan melalui rangkaian sirkuit.

# Tambahan ttg RISC vs CISC


- 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.

<u>Biner</u>	<u>Hexa</u>	<u>Mnemonic</u>
10110110	B6	LDAA ...
10010111	97	STAA ...
01001010	4A	DECA ...
10001010	8A	ORAA ...
00100110	26	BNE ...
00000001	01	NOP...
01111110	7E	JMP ...

Sebagian set instruksi 68HC11

- 
- Pada awalnya, instruksi yang tersedia amat sederhana dan sedikit. Kemudian desainer mikroprosesor berlomba-lomba untuk melengkapi set instruksi itu selengkap-lengkapannya. Jumlah instruksi itu berkembang seiring dengan perkembangan desain mikroprosesor yang semakin lengkap dengan mode pengalamatan yang bermacam-macam. Mikroprosesor lalu memiliki banyak instruksi manipulasi bit dan seterusnya dilengkapi dengan instruksi-instruksi aritmatik seperti penjumlahan, pengurangan, perkalian dan pembagian. Seperti contohnya 68HC11 banyak sekali memiliki set instruksi untuk percabangan seperti BNE, BLO, BLS, BMI, BRCLR, BRSET dan sebagainya.

- 
- Perancang mikroprosesor juga memperkaya ragam instruksi tersebut dengan membuat satu instruksi tunggal untuk program yang biasanya dijalankan dengan beberapa intruksi. Misalnya pada 80C51 untuk contoh program berikut ini.

```
LABEL ...
```

```
...
```

```
DEC R0
```

```
MOV A,R0
```

```
JNZ LABEL
```


Program 'decrement' 80C51


- Program ini adalah program pengulangan yang mengurangi isi register R0 sampai register R0 menjadi kosong (0). Intel menambah set instruksinya dengan membuat satu instruksi khusus untuk keperluan seperti ini :

```
LABEL ....
```

```
DJNZ R0,LABEL
```

Instruksi 'decrement jump not zero' 80C51

- 
- Kedua contoh program ini hasilnya tidak berbeda. Namun demikian, instruksi kompleks seperti DJNZ mempermudah pembuat program.
  - Set instruksi yang lengkap diharapkan akan semakin membuat pengguna mikroprosesor leluasa menulis program dalam bahasa assembler yang mendekati bahasa pemrograman level tinggi. Intel 80C51 yang dikembangkan dari basis prosesor 8048 dirilis pada tahun 1976 memiliki tidak kurang dari 111 instruksi. Tidak ketinggalan, 68HC11 dari Motorola yang populer di tahun 1984 dilengkapi dengan 145 instruksi. Karena banyak dan kompleksnya instruksi yang dimiliki 68HC11 dan 80C51, kedua contoh mikrokontroler ini disebut sebagai prosesor CISC.

- 
- Debat CISC versus RISC dimulai ketika pada tahun 1974 IBM mengembangkan prosesor 801 RISC. Argumen yang dipakai waktu itu adalah mengapa diperlukan instruksi yang kompleks. Sebab pada prinsipnya, instruksi yang kompleks bisa dikerjakan oleh instruksi-instruksi yang lebih sederhana dan kecil. Ketika itu penggunaan bahasa tingkat tinggi seperti Fortran dan kompilasi lain (compiler/interpreter) mulai berkembang. Apalagi saat ini compiler seperti C/C++ sudah lazim digunakan. Sehingga sebenarnya tidaklah diperlukan instruksi yang kompleks di tingkat prosesor. Kompilasi yang akan bekerja menterjemahkan program dari bahasa tingkat tinggi menjadi bahasa mesin.

# Contoh lain untuk memahami RISC vs CISC

- Untuk melihat bagaimana perbedaan instruksi RISC dan CISC, mari kita lihat bagaimana keduanya melakukan perkalian misalnya  $c = a \times b$ . Mikrokontroler 68HC11 melakukannya dengan program sebagai berikut :

```
LDAA #$5
```

```
LDAB #$10
```

```
MUL
```

```
Program 5x10 dengan 68HC11
```

- Cukup tiga baris saja dan setelah ini accumulator D pada 68HC11 akan berisi hasil perkalian dari accumulator A dan B, yakni  $5 \times 10 = 50$ .
- Program yang sama dengan PIC16CXX, adalah seperti berikut ini.

```
MOVLW 0x10 MOVWF Reg1
```

```
MOVLW 0x05
```

```
MOVWF Reg2
```

```
CLRWF
```

```
LOOP ADDWF Reg1,0
```

```
CFSZ Reg2,1
```

```
GOTO LOOP
```


```
...
```


```
...
```

```
Program 5x10 dengan PIC16CXX
```

- Prosesor PIC16CXX yang RISC ini, tidak memiliki instruksi perkalian yang khusus. Tetapi perkalian  $5 \times 10$  itu sama saja dengan penjumlahan nilai 10 sebanyak 5 kali. Kelihatannya membuat program assembly dengan prosesor RISC menjadi lebih kompleks dibandingkan dengan prosesor CISC. Tetapi perlu diingat, untuk membuat instruksi yang kompleks seperti instruksi MUL dan instruksi lain yang rumit pada prosesor CISC, diperlukan hardware yang kompleks juga. Dibutuhkan ribuan gerbang logik (logic gates) transistor untuk membuat prosesor yang demikian. Instruksi yang kompleks juga membutuhkan jumlah siklus mesin (machine cycle) yang lebih panjang untuk dapat menyelesaikan eksekusinya. Instruksi perkalian MUL pada 68HC11 memerlukan 10 siklus mesin dan instruksi pembagiannya memerlukan 41 siklus mesin.




- 
- Pendukung RISC berkesimpulan, bahwa prosesor yang tidak rumit akan semakin cepat dan handal. Hampir semua instruksi prosesor RISC adalah instruksi dasar (belum tentu sederhana), sehingga instruksi-instruksi ini umumnya hanya memerlukan 1 siklus mesin untuk menjalankannya. Kecuali instruksi percabangan yang membutuhkan 2 siklus mesin.
  - RISC biasanya dibuat dengan arsitektur Harvard, karena arsitektur ini yang memungkinkan untuk membuat eksekusi instruksi selesai dikerjakan dalam satu atau dua siklus mesin.

- 
- Sebagai perbandingan jumlah instruksi pada prosesor RISC
    - COP8 hanya dilengkapi dengan 58 instruksi dan PIC12/16CXX hanya memiliki 33 instruksi saja.
    - Untuk merealisasikan instruksi dasar yang jumlah tidak banyak ini, mikroprosesor RISC tidak memerlukan gerbang logik yang banyak. Karena itu dimensi dice IC dan konsumsi daya prosesor RISC umumnya lebih kecil dibanding prosesor CISC.
    - Bukan karena kebetulan, keluarga mikrokontroler PICXX banyak yang dirilis ke pasar dengan ukuran mini. Misalnya PIC12C508 adalah mikrokontroler DIP 8 pin.

# Hal lain...

- 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 menanam ribuan bahkan jutaan transistor di dalam satu dice. Berbagai 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.

- 
- 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.

# Kesimpulan

---

- Pentium Intel mampu mendominasi pasaran dan secara teknologi menggunakan rancangan CISC (complex instruction set computers) dalam arsitekturnya.
- PowerPC merupakan kelompok komputer yang menerapkan teknologi RISC (reduced instruction set computers).

# Generasi Pentium

- 8080, keluar tahun 1972 merupakan mikroprosesor pertama keluaran Intel dengan mesin 8 bit dan bus data ke memori juga 8 bit. Jumlah instruksinya 66 instruksi dengan kemampuan pengalamatan 16KB.
- 8086, dikenalkan tahun 1974 adalah mikroprosesor 16 bit dengan teknologi cache instruksi. Jumlah instruksi mencapai 111 dan kemampuan pengalamatan ke memori 64KB.
- 80286, keluar tahun 1982 merupakan pengembangan dari 8086, kemampuan pengalamatan mencapai 1MB dengan 133 instruksi.
- 80386, keluar tahun 1985 dengan mesin 32 bit. Sudah mendukung sistem multitasking. Dengan mesin 32 bitnya, produk ini mampu menjadi terunggul pada masa itu.
- 80486, dikenalkan tahun 1989. Kemajuannya pada teknologi cache memori dan pipelining instruksi. Sudah dilengkapi dengan math co-processor.

- Pentium, dikeluarkan tahun 1993, menggunakan teknologi superscalar sehingga memungkinkan eksekusi instruksi secara paralel.
- Pentium Pro, keluar tahun 1995. Kemajuannya pada peningkatan organisasi superscalar untuk proses paralel, ditemukan sistem prediksi cabang, analisa aliran data dan sistem cache memori yang makin canggih.
- Pentium II, keluar sekitar tahun 1997 dengan teknologi MMX sehingga mampu menangani kebutuhan multimedia. Mulai Pentium II telah menggunakan teknologi RISC.
- Pentium III, terdapat kemampuan instruksi floating point untuk menangani grafis 3D.
- Pentium IV, kemampuan floating point dan multimedia semakin canggih.
- Itanium, memiliki kemampuan 2 unit floating point, 4 unit integer, 3 unit pencabangan, internet streaming, 128 interger register.

# PowerPC



- Proyek sistem RISC diawali tahun 1975 oleh IBM pada komputer murni seri 801.
- Seri pertama ini hanyalah prototipe, seri komersialnya adalah PC RT yang dikenalkan tahun 1986.
- Tahun 1990 IBM mengeluarkan generasi berikutnya yaitu IBM RISC System/6000 yang merupakan mesin RISC superskalar workstation.
- Setelah ini arsitektur IBM lebih dikenal sebagai arsitektur POWER



# Power PC



- IBM menjalin kerja sama dengan Motorola menghasilkan mikroprosesor seri 68000 (68K)
- Apple menggunakan keping Motorola dalam Macintoshnya. Saat ini terdapat beberapa kelompok PowerPC


# Kelompok Power PC




- 601, adalah mesin 32 bit merupakan produksi massal arsitektur PowerPC untuk lebih dikenal masyarakat.
- 603, merupakan komputer desktop dan komputer portabel. Kelompok ini sama dengan seri 601 namun lebih murah untuk keperluan efisien.
- 604, seri komputer PowerPC untuk kegunaan komputer low-end server dan komputer desktop.
- 620, ditujukan untuk penggunaan high-end server. Mesin dengan arsitektur 64 bit.
- 740/750, seri dengan cache L2.
- G4, seperti seri 750 tetapi lebih cepat dan menggunakan 8 instruksi paralel

# From Wikipedia...

- Power Macintosh, atau Power Mac, adalah nama dari seri produk komputer pribadi Apple Macintosh yang menggunakan prosesor PowerPC.
- Apple Computer mengeluarkan Power Mac pertama kali pada tahun 1994, dimulai dengan jenis Power Mac 6100 yang memiliki kecepatan 60 dan 66 MHz. Model-model sebelumnya menggunakan seri prosesor Motorola 68000 ("68K").
- Bagian ROM dan Sistem operasi Mac OS di dalam Power Mac memiliki emulator yang memungkinkan sebagian besar program yang sebelumnya dibuat untuk komputer seri Motorola 68k dapat berjalan di mesin yang baru tanpa perubahan.
- Pada awalnya Power Mac dirancang Apple untuk dijual kepada pasaran tingkat tinggi sehingga mesin berbasis 68K masih diproduksi untuk memenuhi permintaan pasaran yang lainnya. Pada tahun 1996 Apple memberhentikan seri Macintosh LC 580 yang merupakan produksi terakhir dari seri 68K. Sejak saat itu semua komputer Macintosh menggunakan prosesor PowerPC.

- 
- Semua Power Mac sebelum tahun 1997 menggunakan seri prosesor PowerPC 60x. Pada tahun 1997 Power Macintosh generasi ketiga (G3) dilepas dengan menggunakan prosesor PowerPC 750. Mulai dari model ini, Apple tidak lagi menggunakan seri angka untuk memberi nama pada komputernya. Mereka memilih untuk menamai komputer mereka berdasarkan generasi prosesor yang digunakan (seperti G3, G4 dan G5). Model selanjutnya memiliki ciri khas untuk membedakan bagi tiap generasi prosesor yang keluar. Contohnya Power Mac G3 menggunakan kerangka berwarna biru-putih sedangkan Power Mac G4 menggunakan dua warna yang berbeda: hitam-putih dan perak. Power Mac G5 kemudian menggunakan bahan aluminium untuk melindungi perangkat keras di dalamnya. Metode identifikasi yang mirip juga dipakai untuk produk Apple lainnya seperti iMac, PowerBook dan iBook.

- 
- Pada saat ini Power Mac dipromosikan Apple untuk dijual ke pengguna tingkat atas seperti perusahaan besar dan universitas. Power Mac memiliki banyak keunggulan dibanding seri komputer buatan Apple lainnya karena mereka selalu melakukan eksperimen dengan teknologi terbaru menggunakan Power Mac sebelum diterapkan di seri yang lainnya. Oleh karena itu harga Power Mac selalu tergolong sangat mahal bagi kalangan awam.
  
  - Di bawah ini adalah beberapa seri Power Mac yang terakhir:
    - ▣ Power Macintosh G3
    - ▣ Power Macintosh G4
    - ▣ Power Mac G4 Cube
    - ▣ Power Macintosh G5
    - ▣ Tahun 2006 lini produk Power Mac dihentikan dan mulai digantikan Mac Pro.

# Mengapa harus switch ke Intel?

- Pertanyaannya mengapa PowerPC G5 tidak menjadi penerus chip yang dipergunakan Apple untuk masa depan, atau dengan release PowerPC G6? Masalahnya terletak pada dua hal.
  - ▣ Pertama IBM mulai melirik sektor game console sebagai market utama chip PowerPC (coba perhatikan semua chip game console, baik Nintendo GameCube, DreamCast, Microsoft XBOX, Sony PS3) semua berhasil diapproach IBM untuk mempergunakan chip PowerPC, dan market game console ini jauh lebih menarik dibandingkan market komputer desktop yang konsumen utamanya hanya Apple. Apple mengkonsumsi maksimal 5 juta chip per tahun sedangkan game console tersebut mengkonsumsi lebih dari 20 juta chip per tahun dan increasing.