MikroKontroler

Eri Prasetyo Wibowo
Universitas Gunadarma

Why do we need to learn Microprocessors/controllers?

- The microprocessor is the core of computer systems.
- Nowadays many communication, digital entertainment, portable devices, are controlled by them.
- A designer should know what types of components he needs, ways to reduce production costs and product reliable.

Different aspects of a microprocessor/controller

- Hardware :Interface to the real world

- Software :order how to deal with inputs

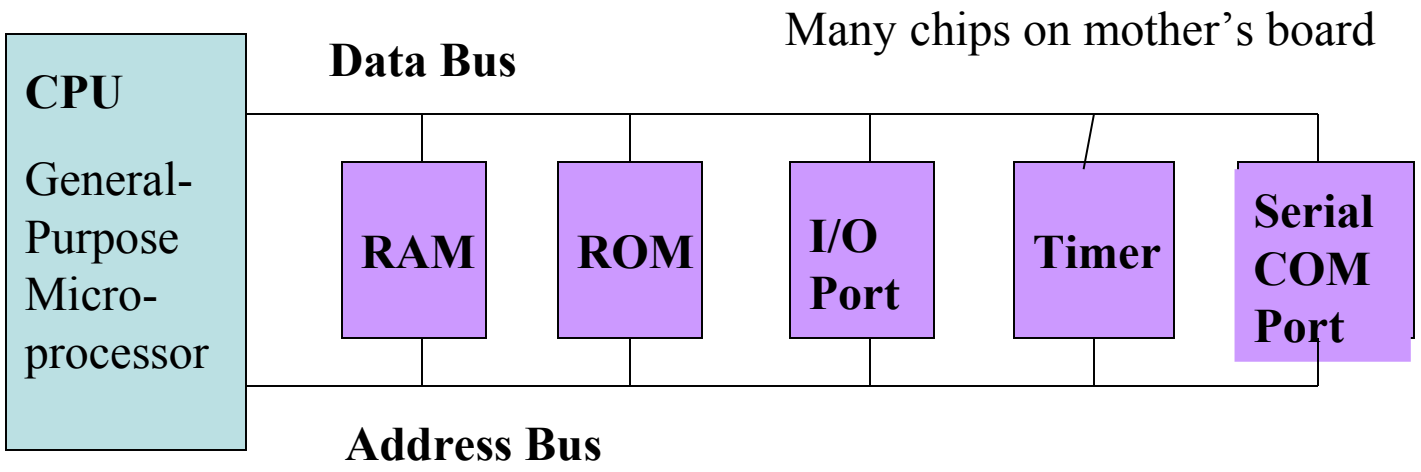
The necessary tools for a microprocessor/controller

- CPU: Central Processing Unit
- I/O: Input /Output
- Bus: Address bus & Data bus
- Memory: RAM & ROM
- Timer
- Interrupt
- Serial Port
- Parallel Port

Microprocessors:

General-purpose microprocessor

- CPU for Computers
- No RAM, ROM, I/O on CPU chip itself
- Example ： Intel’s x86, Motorola’s 680x0



General-Purpose Microprocessor System

Microprocessor vs. Microcontroller

Microprocessor

- CPU is stand-alone, RAM, ROM, I/O, timer are separate
- designer can decide on the amount of ROM, RAM and I/O ports.
- expansive
- versatility
- general-purpose

Microcontroller

- CPU, RAM, ROM, I/O and timer are all on a single chip
- fix amount of on-chip ROM, RAM, I/O ports
- for applications in which cost, power and space are critical
- single-purpose

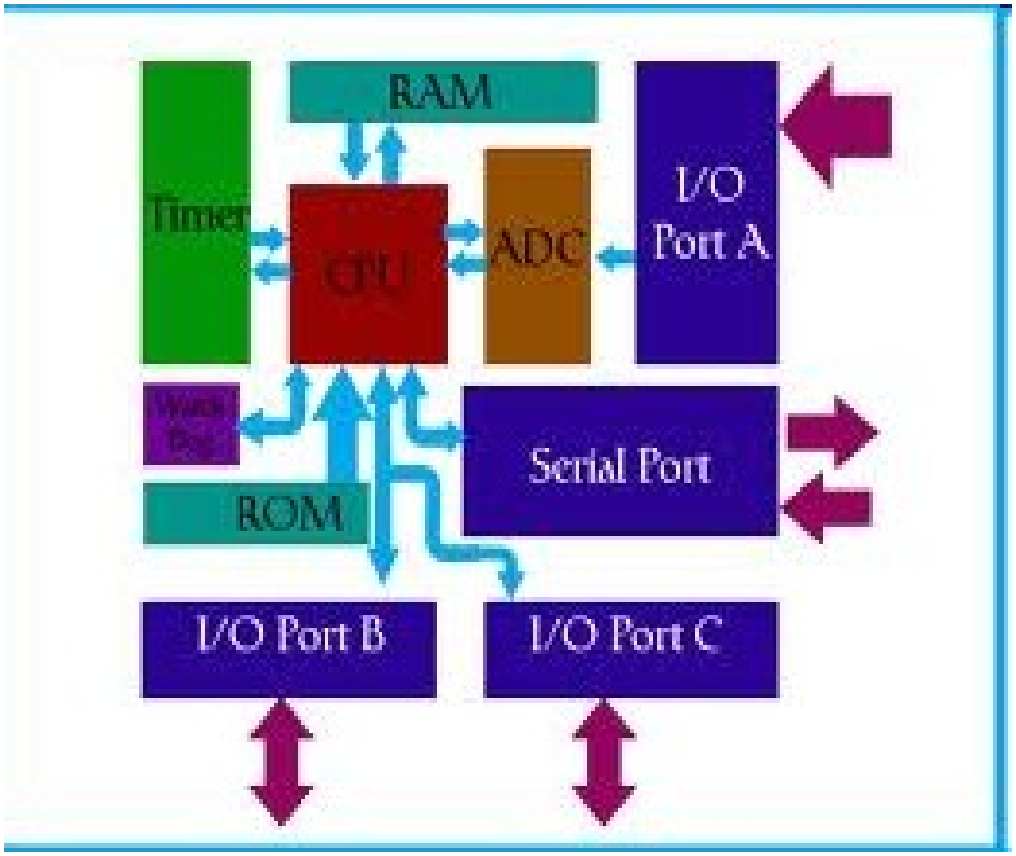
Three criteria in Choosing a Microcontroller

1. meeting the computing needs of the task efficiently and cost effectively
   - speed, the amount of ROM and RAM, the number of I/O ports and timers, size, packaging, power consumption
   - easy to upgrade
   - cost per unit
2. availability of software development tools
   - assemblers, debuggers, C compilers, emulator, simulator, technical support
3. wide availability and reliable sources of the microcontrollers.

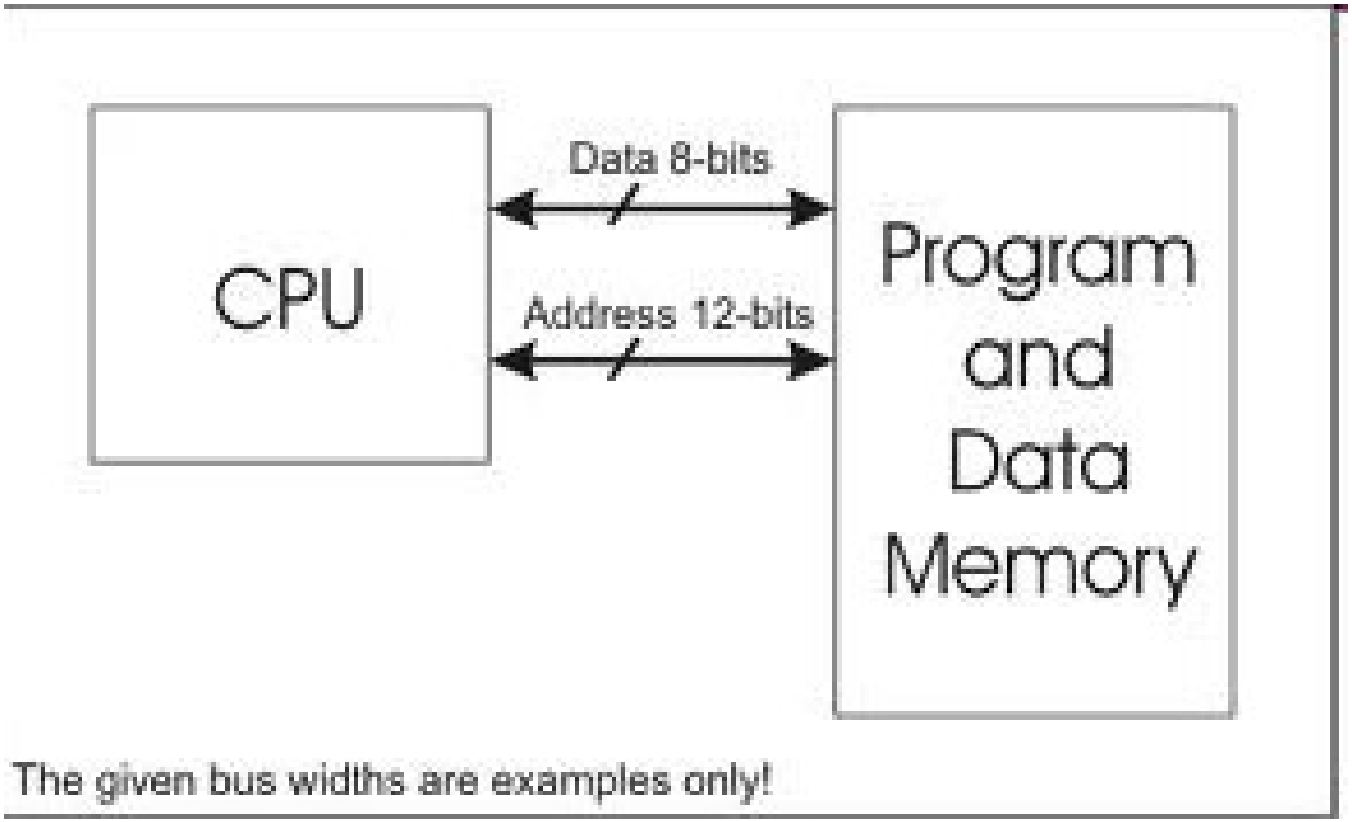
Apa Mikrokontroler ?

- Small-Computer
  - Microprocessor
    - The Brains
    - Arithmetic Logic Unit (ALU)
    - Control Unit
- Program/ Data Storage
- Peripherals (Input/Output)

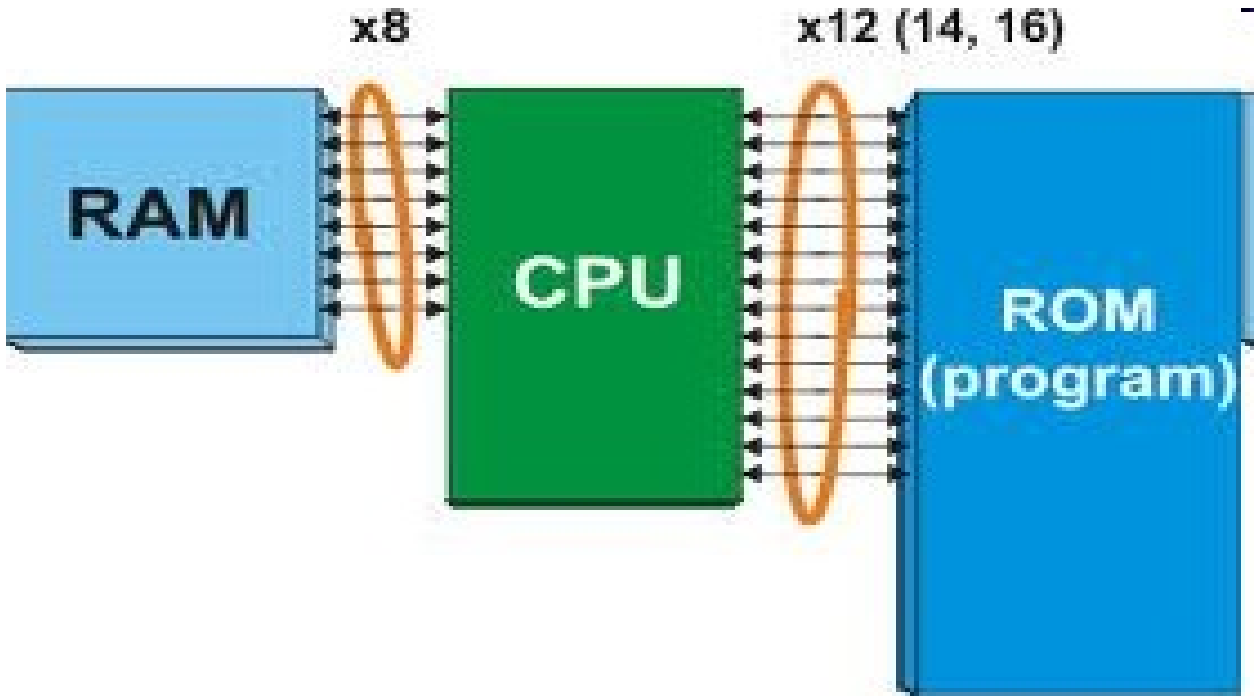
Low-Cost



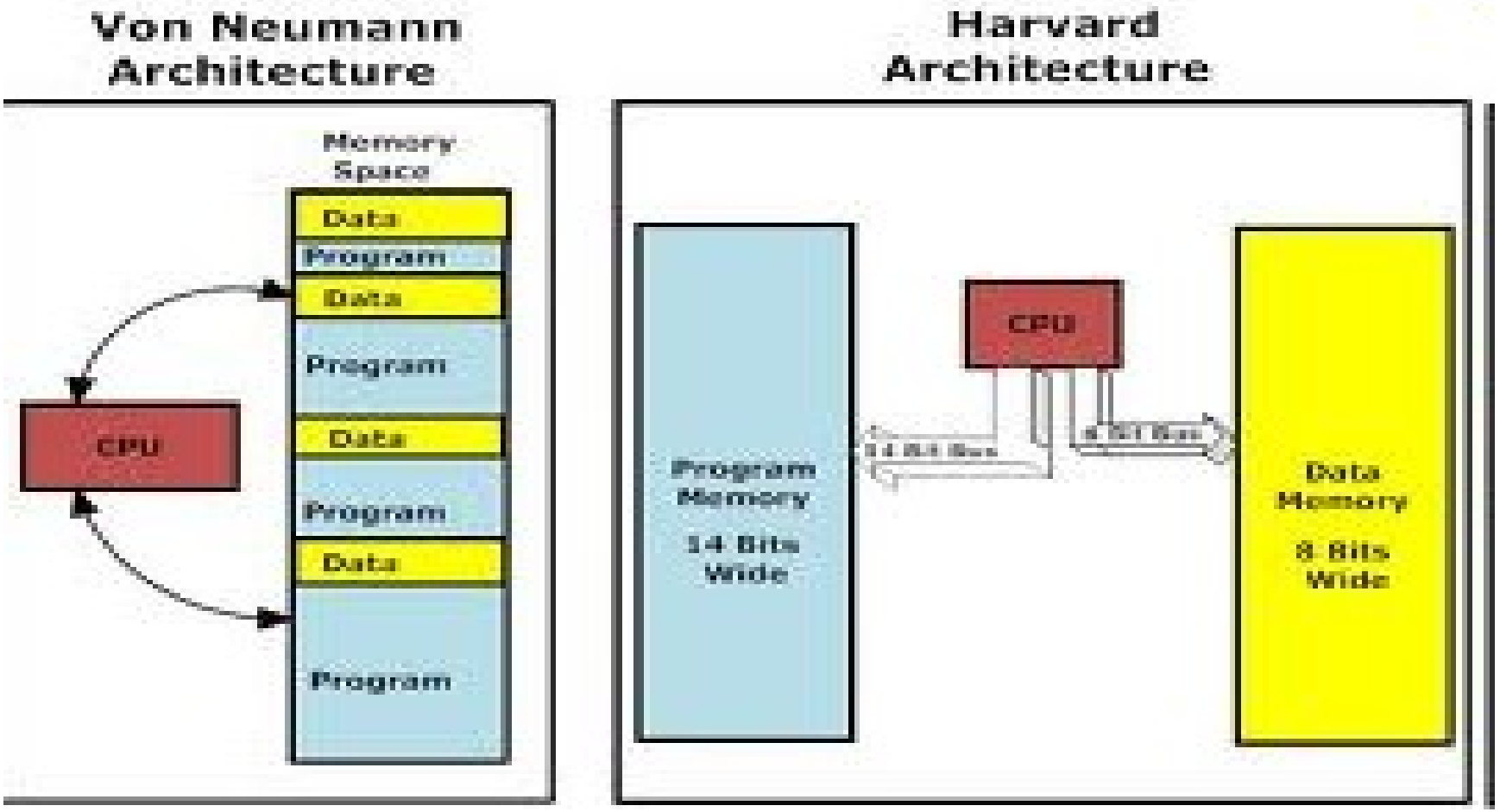
Arsitektur Von Neumann



Arsitektur Harvard

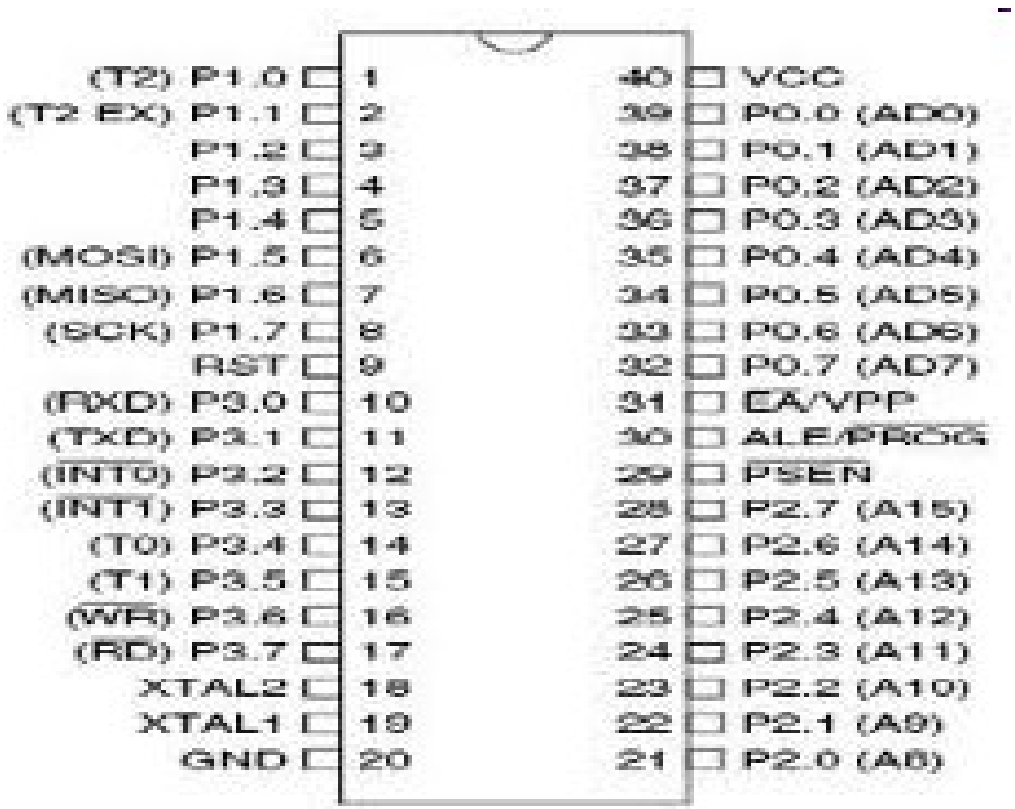


Perbedaan Keduanya



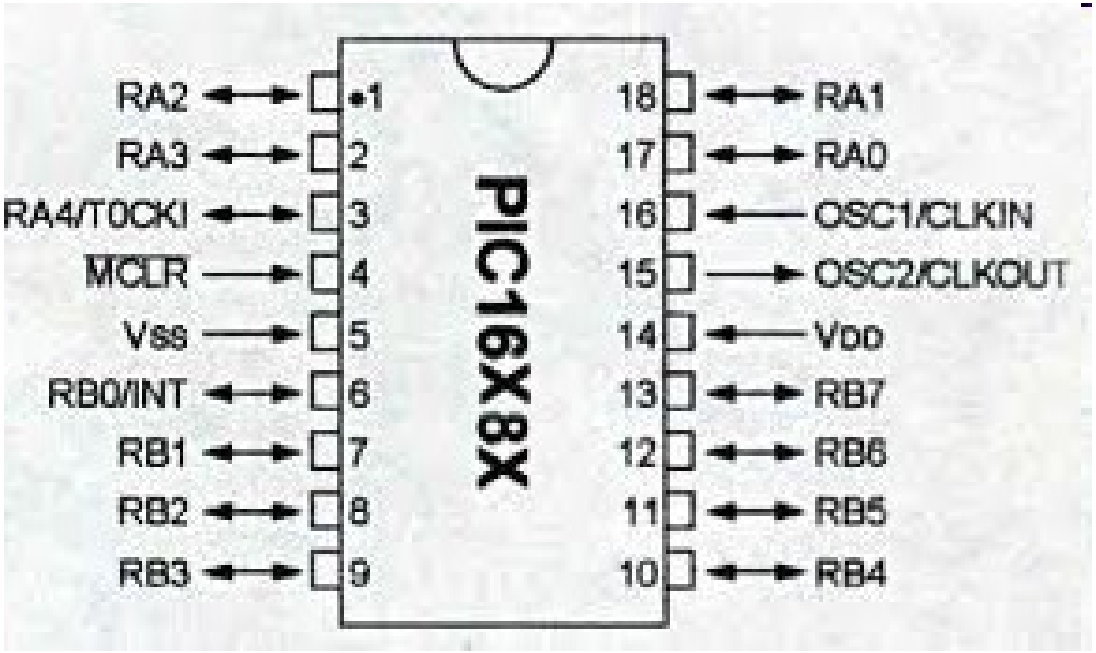
Arsitektur Mikrokontroler

- CISC, Complex Instructtion Set Computing yang lebih kaya intruksi tetapi fasilitas internal secukupnya (AT89 memiliki 255 intruksi)



Arsitektur Mikrokontroler

- RISC, Reduce instruction Set Computing yang lebih kaya fasilitas internalnya tetapi jumlah intruksi secukupnya (PIC16F hanya sekitar 30 intruksi)



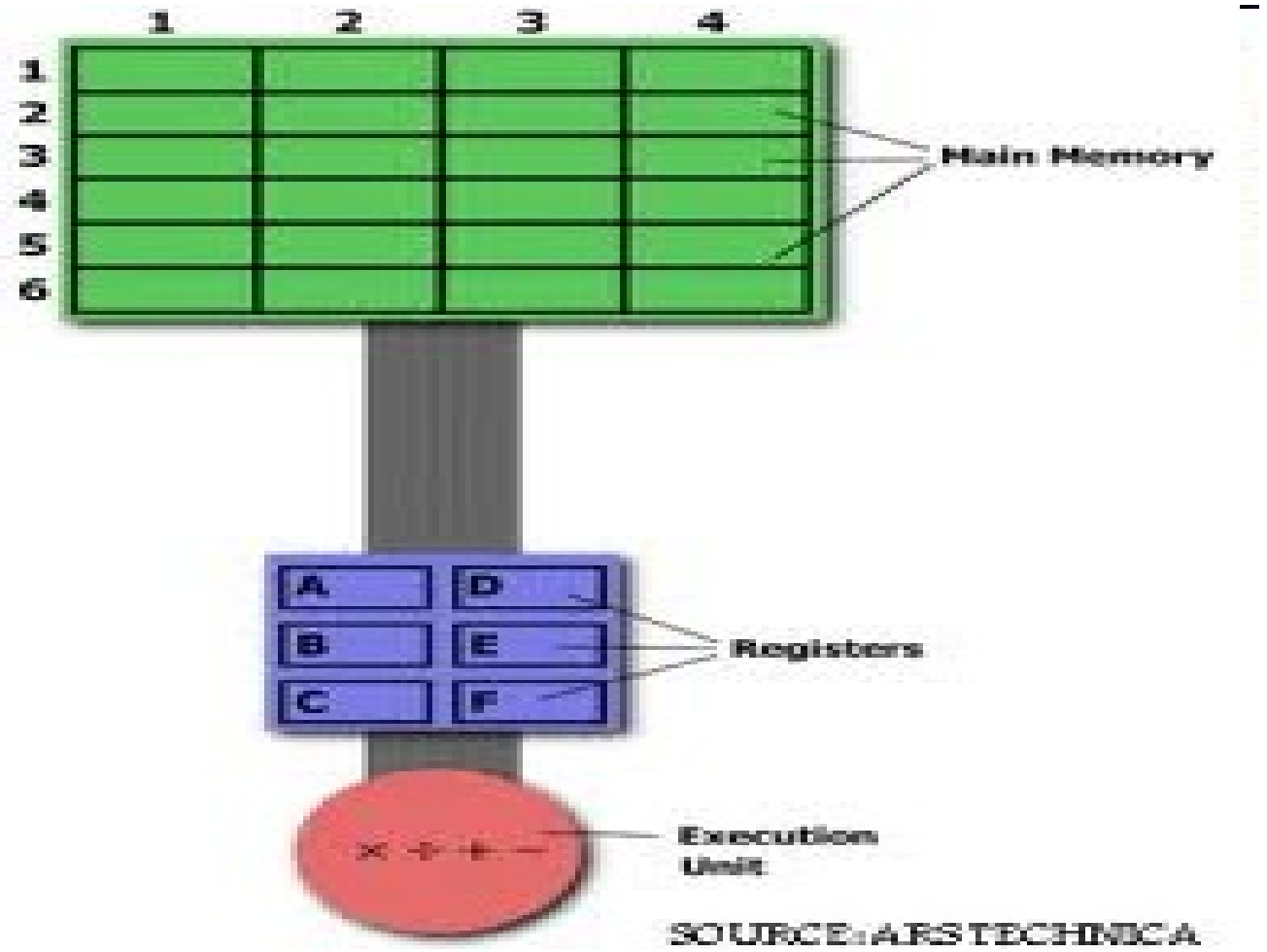
Perbandingan CISC dan RISC

- Pendekatan CISC
  - Melaksanakan perintah cukup dengan beberapa baris bahasa mesin sesedikit mungkin
  - Memori ke memori “ LOAD” dan “STORE “ saling kerjasama
  - Ukuran kode kecil , kecepatan rendah
  - Transistor digunakan untuk menyimpan intrusksi-intruksi kompleks

Perbandingan CISC dan RISC

- Pendekatan RISC
  - Penekanan pada perangkat lunak single clock hanya sejunlah kecil intruksi
  - Register ke register “LOAD” dan “STORE” adalah intruksi-intruksi terpisah
  - Ukuran kode besar , kecepatan relatif tinggi
  - Transistor banyak digunakan untuk register memori

Struktur Memori



- Memori terbagi menjadi beberapa lokasi (aris,kolom) dari 1,1 sd 6:4
- UNit eksekusi melakukan operasi komputasi
- Unit eksekusi beroprasi untuk data yang sudah disimpan di register (A,B,C,D,E,F)

Contoh kasus perkalian dua bilangan dalam memori

- CISC

  - MULT 2:3, 4:2,

    satu angka disimpan di 2,3 angka lain disimpan 4,2, hasil perkalian disimpan di 2,3 lagi.

- MULT dikenal dengan complex instruction, bekerja secara langsung melalui memori dan tidak memerlukan intruksi lain seperti fungsi baca atau simpan

Contoh kasus perkalian dua bilangan dalam memori

- RISC
  - LOAD A, 2:3 (memindah data ke memori)

    LOAD B, 4:2

    PROD A,B (operasi product atau kali)

    STORE 2:3, A (menyimpan data hasil ke memori)