



POST AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY



CHAPTER 2

Hardware of Embedded Systems and 8051

Faculty:

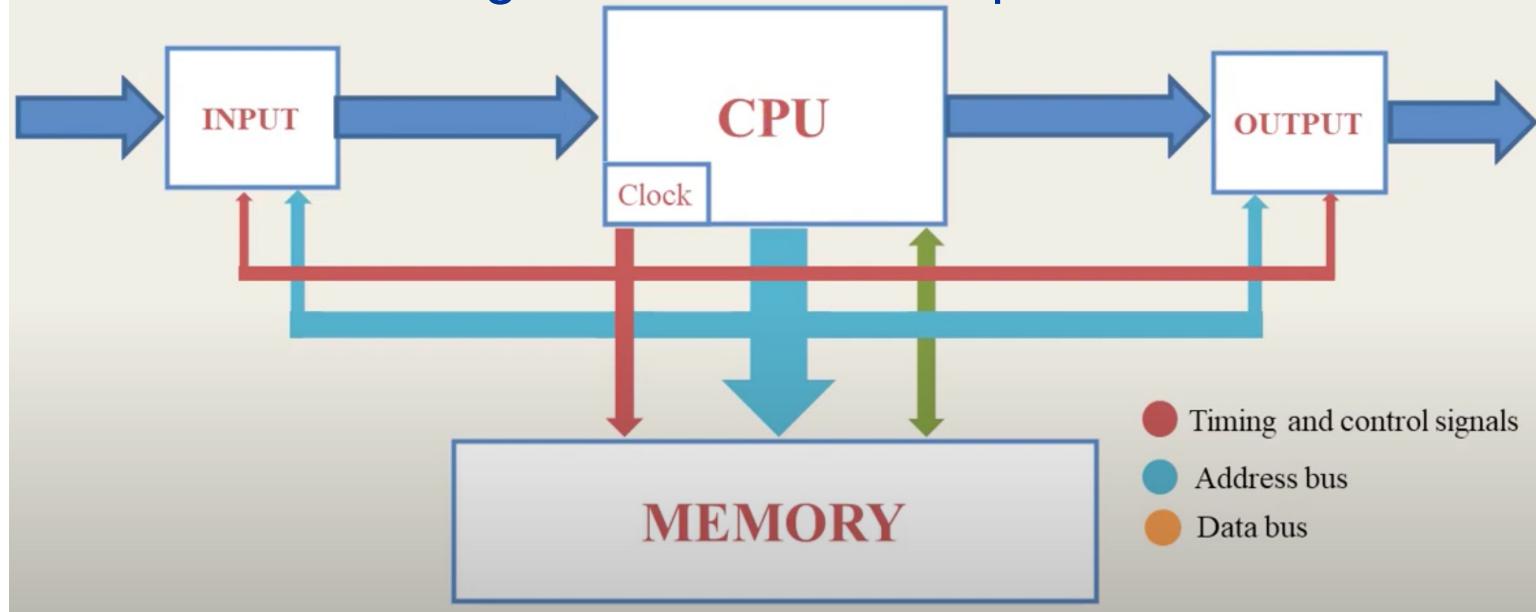
Computer science – IT1

CONTENT

1. Central processing unit - CPU
2. Memory and memory design
3. Pairing with peripheral devices
4. 8051 microprocessor: Architecture, memory and practical interface

❖ Hardware Overview of Embedded Systems

- CPU (Central Processing Unit): Executes instructions and processes data
- Memory: Program and data memory
- Peripheral: Input / output devices
- Bus: transmits signals between components



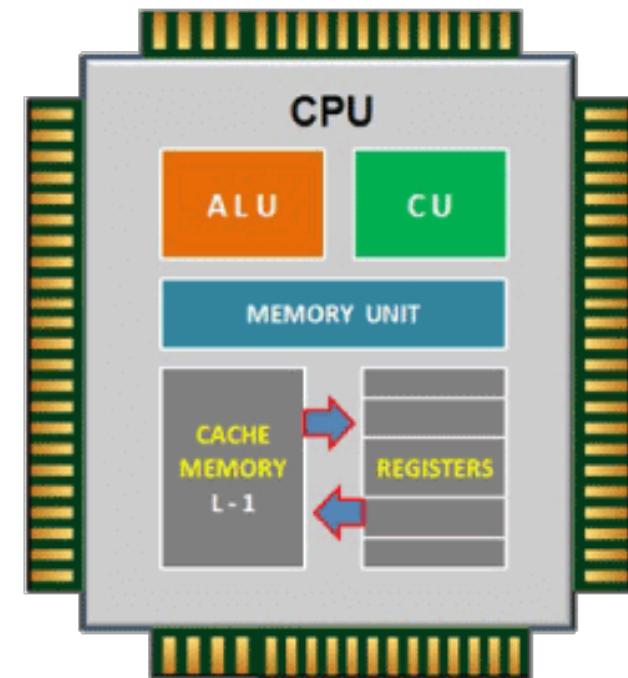
Chapter 2: Hardware components

2.1: CPU

Central Processing Unit

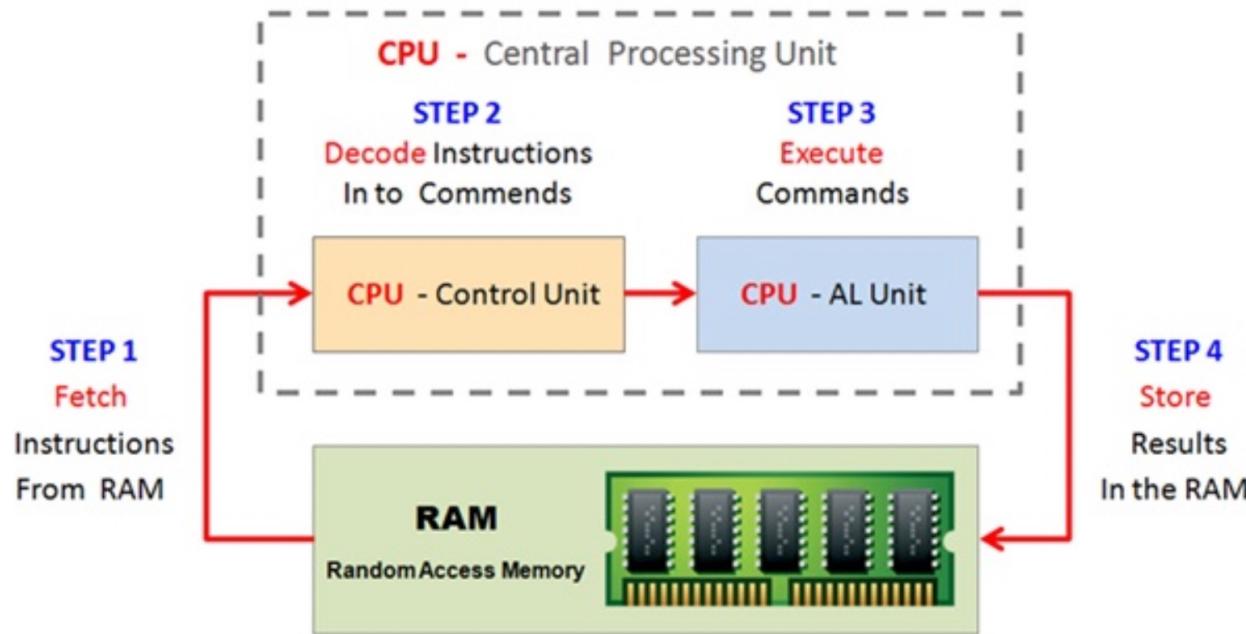
❖ Basic CPU block diagram includes:

- ALU: Computational and logic unit
- CU: Control unit
- Memory Unit: Some memory is built into the CPU to support storage and fast calculations. Includes registers and cache.



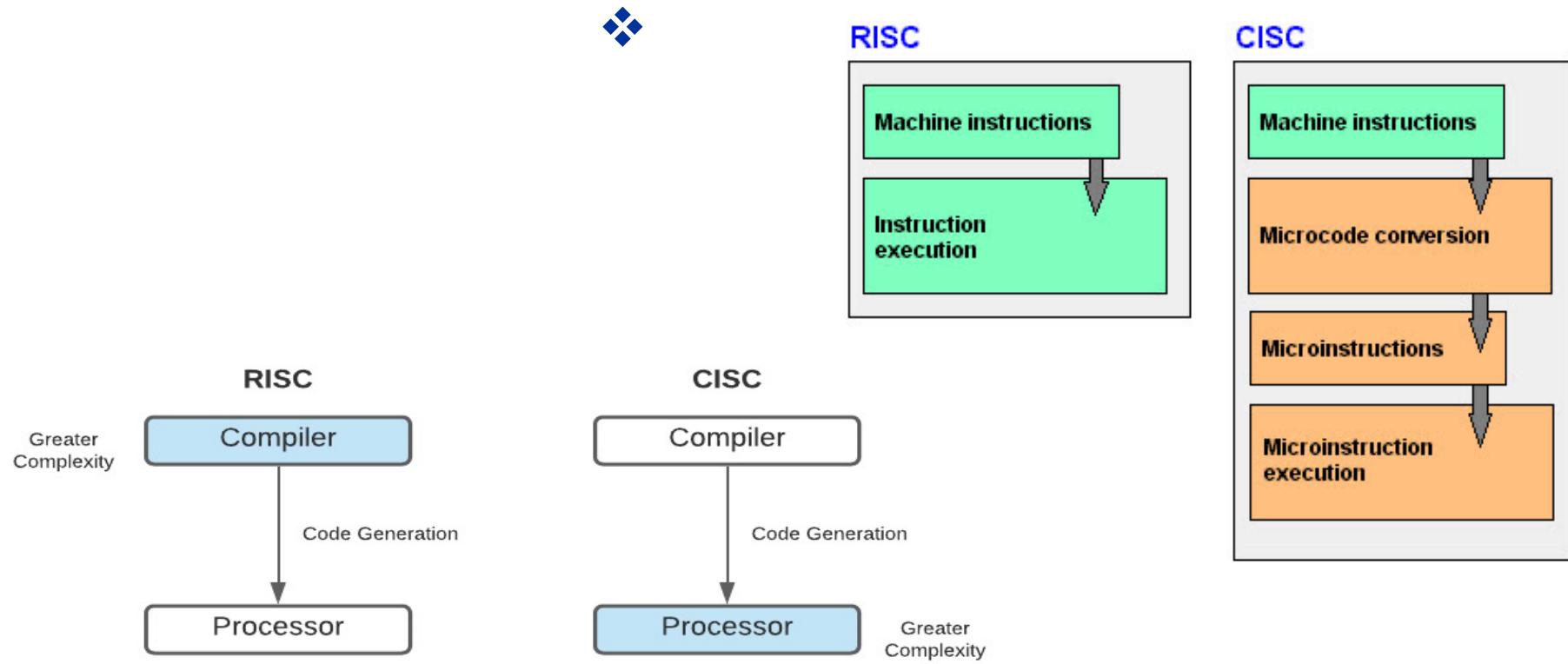
❖ Process of executing a command:

1. Load instructions from memory into CU
2. CU Decodes the command
3. The ALU executes the command
4. The results are saved in memory



❖ ISA (Instruction Set Architecture):

1. CISC (Complex Instruction Set Computing)
2. RISC (Reduced Instruction Set Computing) - ARM

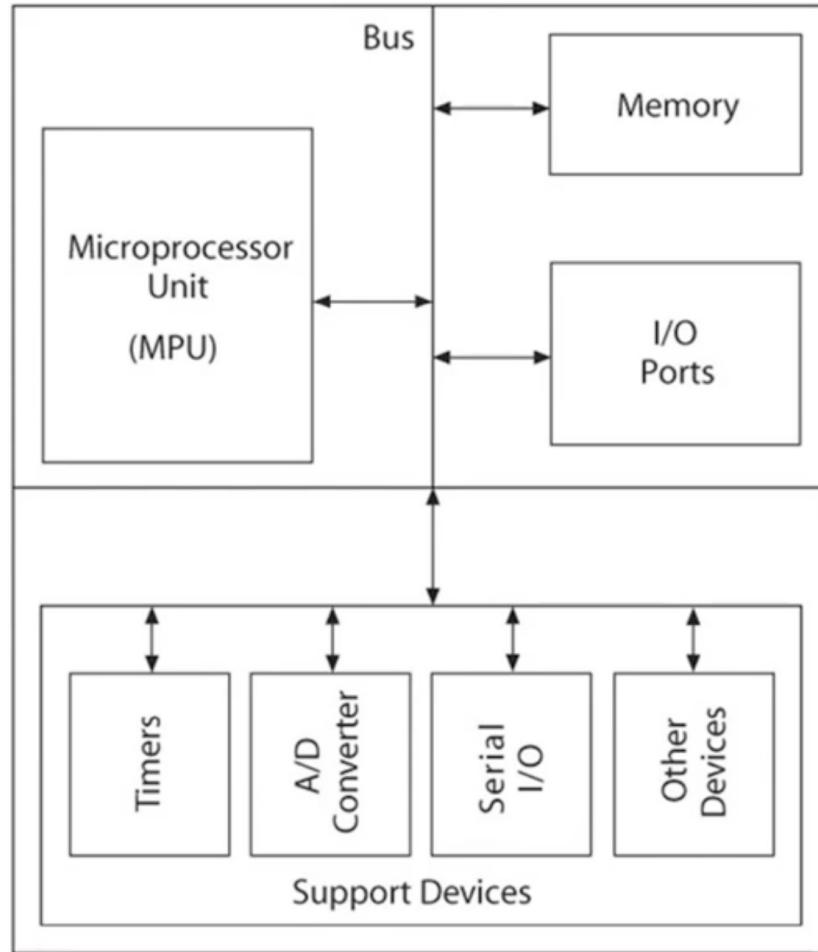


❖ Microcontroller is considered as a fully integrated circuit with components:

- Processor
- Memory
- Peripherals (GPIO, Timer, SPI..)

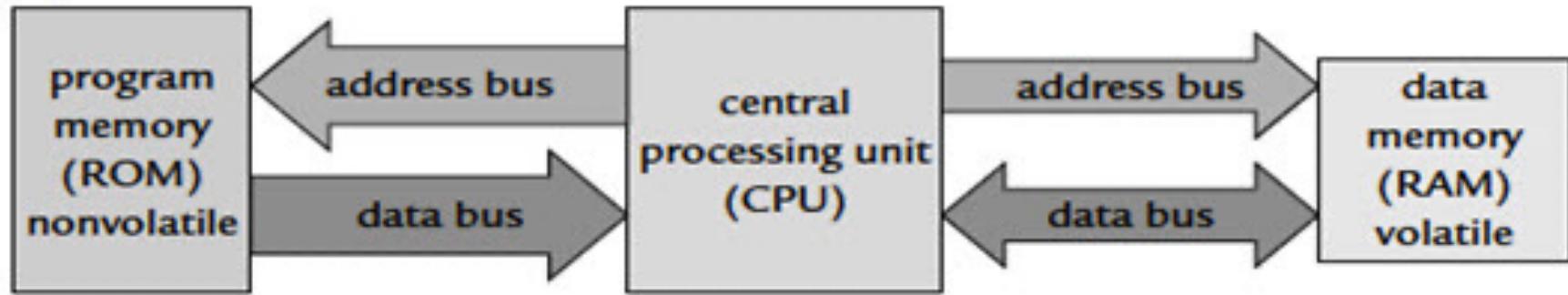
❖ For example:

- Processor: Core i3, i5, i7...
- Microcontroller: PIC, AVR...

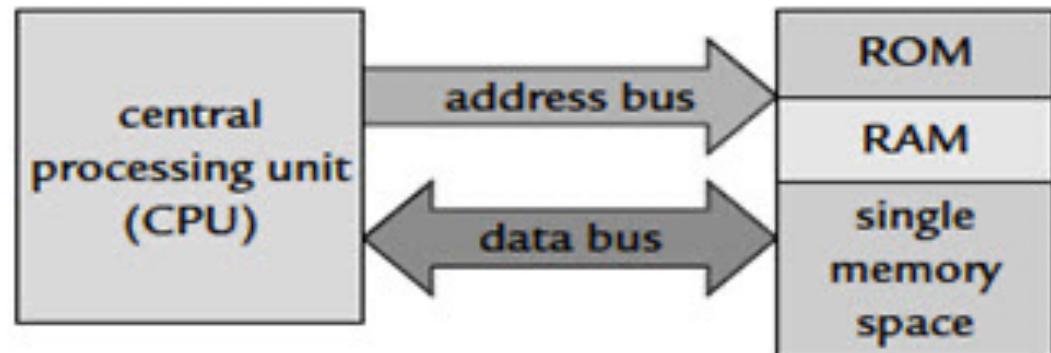


Von-Neumann & Harvard

(a) Harvard architecture



(b) von Neumann architecture



Chapter 2: Hardware components

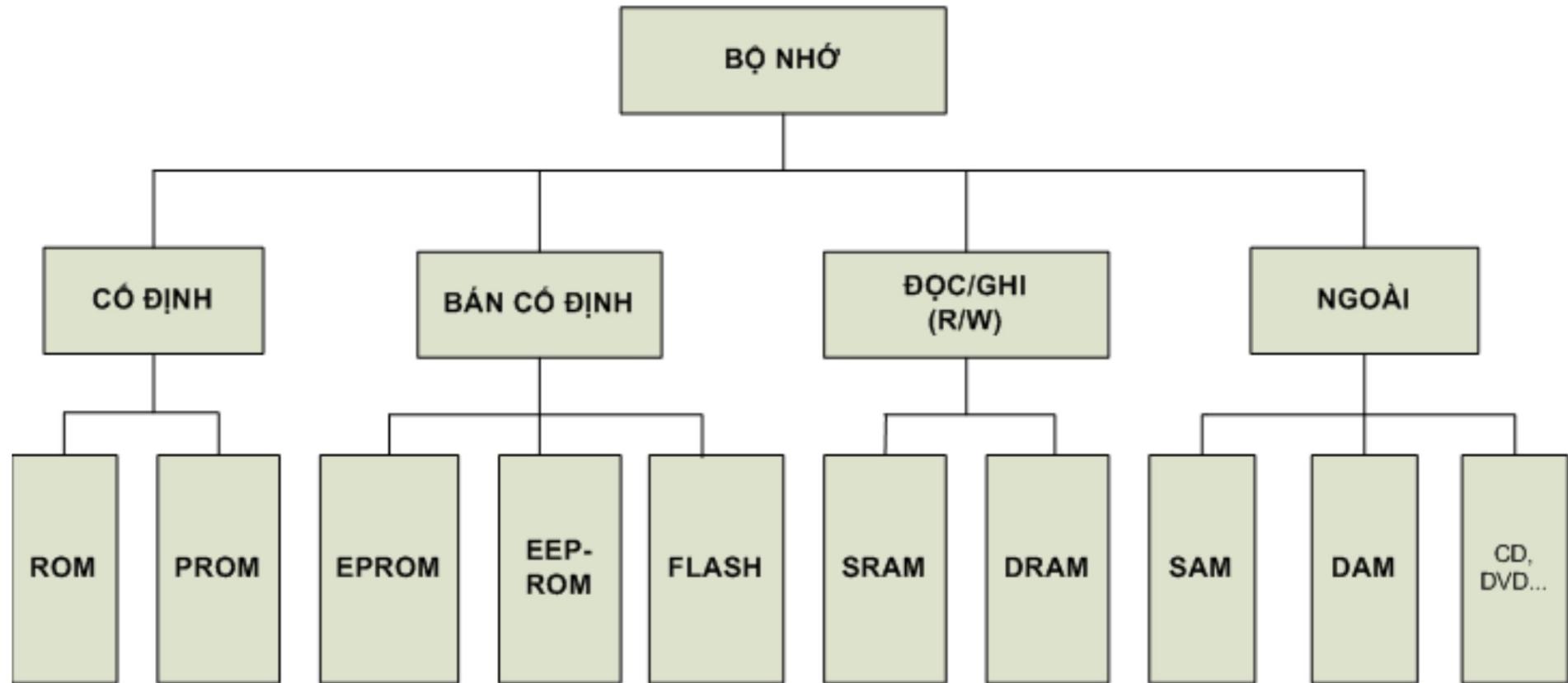
2.2: Memory and memory design

Memory and memory design

- ❖ Memory is a means of storing information including programs and data.
- ❖ Some basic information about computer memory.

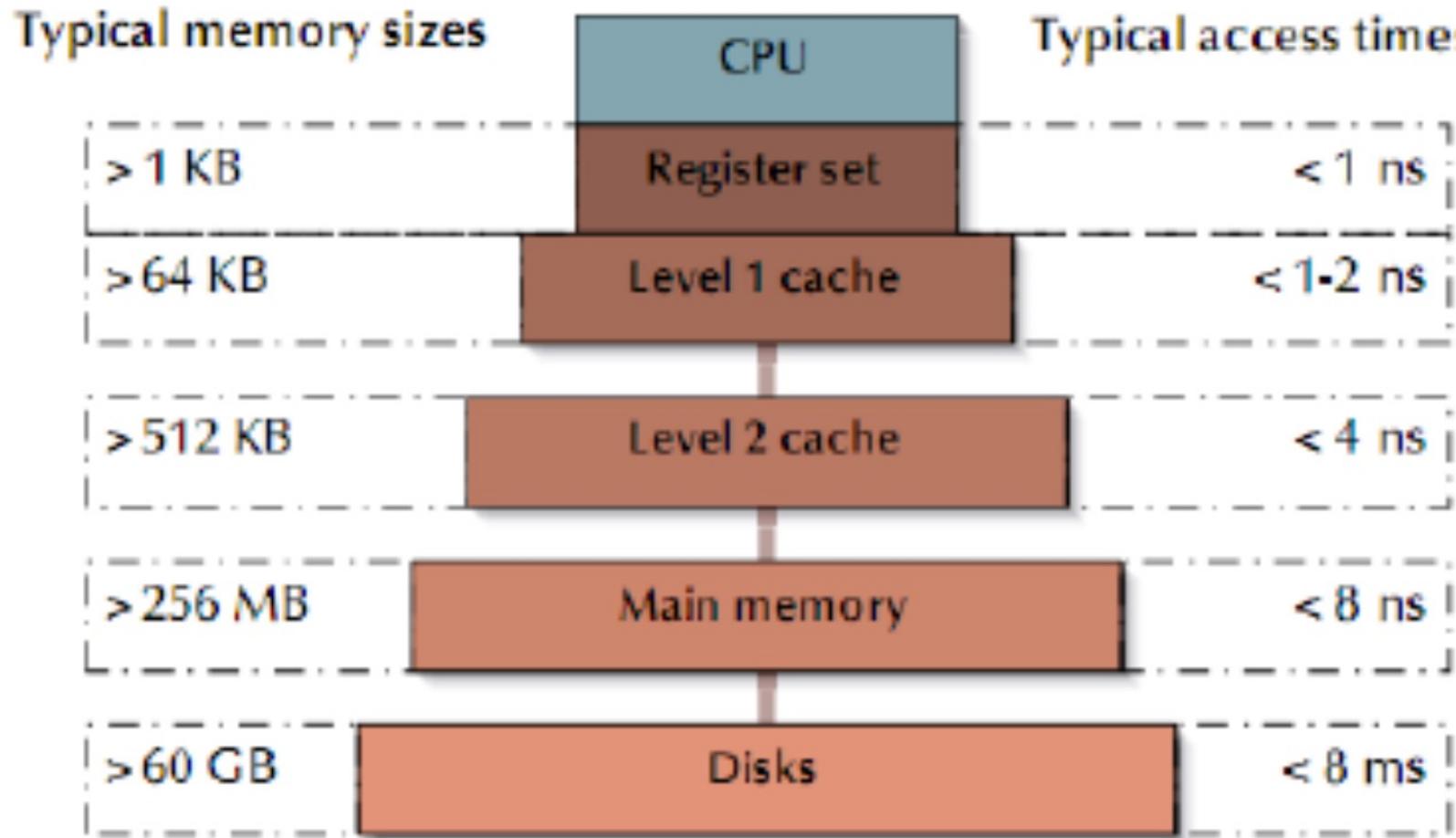
Memory Type	Access Time	Cost /MB	Typical Amount Used	Typical Cost
Registers	1ns		1KB	
Cache	5-20 ns		1MB	
Main memory	60-80ns		MB	
Disk memory	10 ms		GB	

❖ Classification:



Memory and memory design

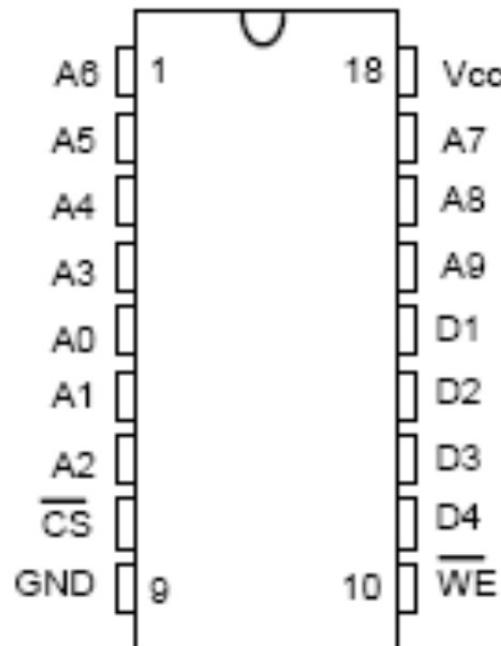
❖ Memory hierarchy model:



- ❖ A memory chip is a specific microchip, arranged with basic pins.
- ❖ The pins of a typical memory chip include address bus inputs, data inputs, chip select control pins, write/read and power pins.
- ❖ For example: a 1Kx4 static RAM (1024 “memory words”, each word is 4 bits long

A0 ÷ A9	Các chân địa chỉ
D1 ÷ D4	Các chân dữ liệu
CS	Chân chọn chip
WE	Điều khiển Ghi/Đọc
Vcc	Chân nguồn nuôi +5V
GND	Chân nối đất

Hình III.8 Sơ đồ nối chân một vi mạch nhớ
RAM 1Kx4



- ❖ Designing memory from available memory chips has the following steps:

1. Determine the number of memory chips to create the required memory capacity according to the formula:

$$M = Q/D, \text{ whereas}$$

Q is the capacity of memory.

D is the capacity of each chip

M is the number of memory chips needed.

2. Determine the number of base address wires (ie the number of low address wires connected directly to the memory chip or interconnect chip): according to the following expression:

$$2^m = D, \text{ whereas}$$

m is the wire number of the base address.

D is the capacity of each chip

3. Determine the number of address lines to create chip select according to the following expression:

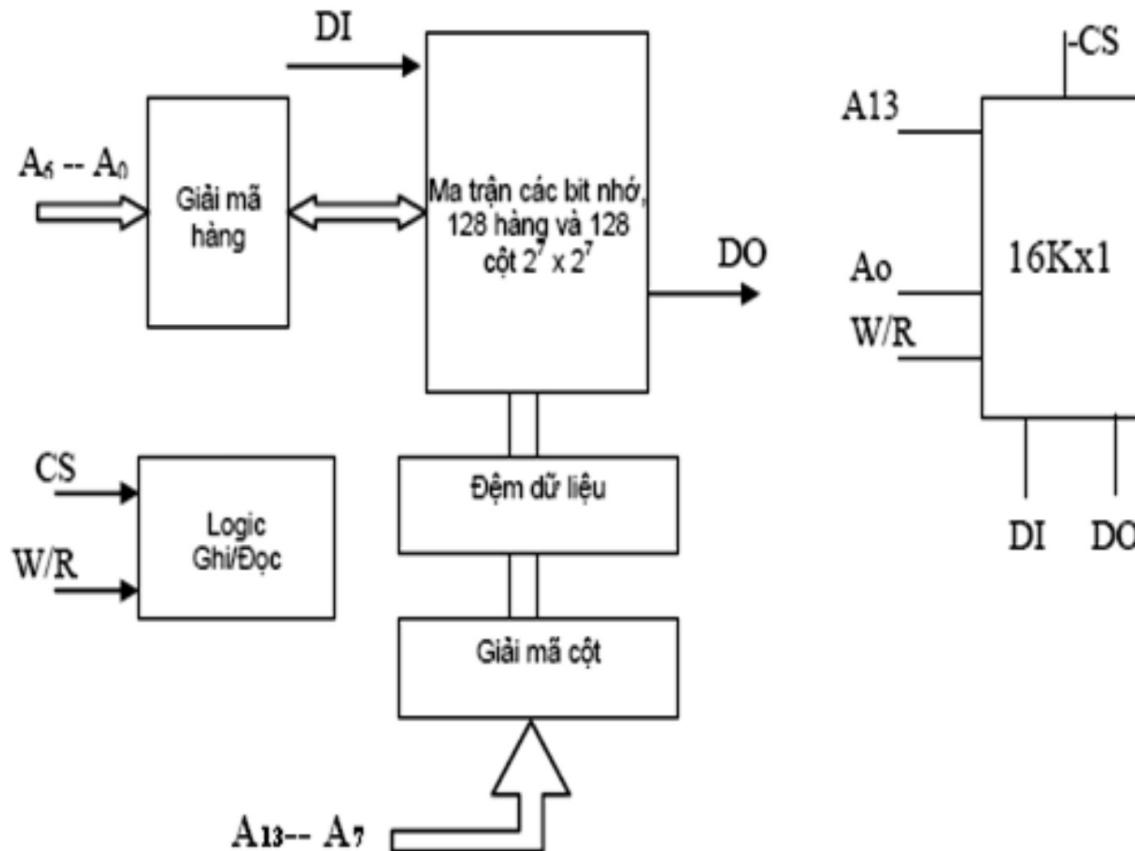
$$2^i = M, \text{ whereas}$$

D is the capacity of each chip

i is the number of wires needed to decode the signals chip select signal (C_{Si})

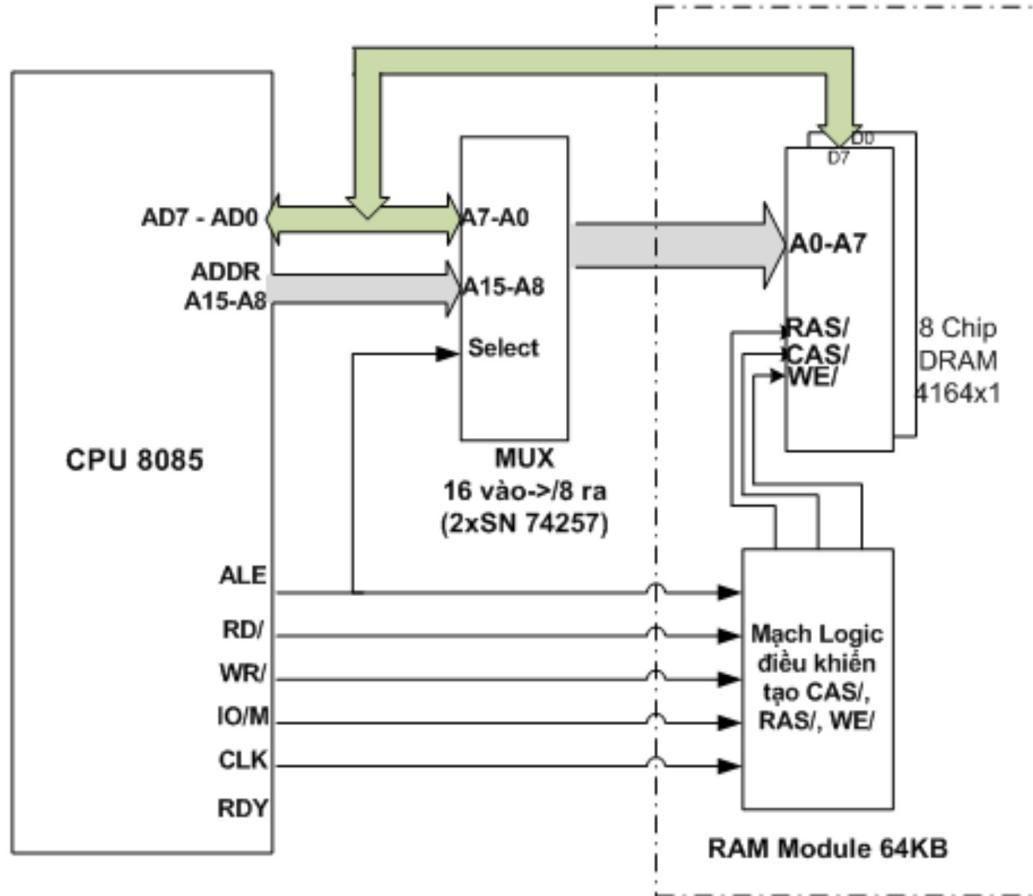
❖ Design Static RAM

Build a 16Kbyte memory based on 16Kx1bit SRAM chips. The 16Kbyte SRAM memory tape is built on the basis of 8 16K x 1bit SRAM chips, to obtain a memory cell with a length of 8 bits.



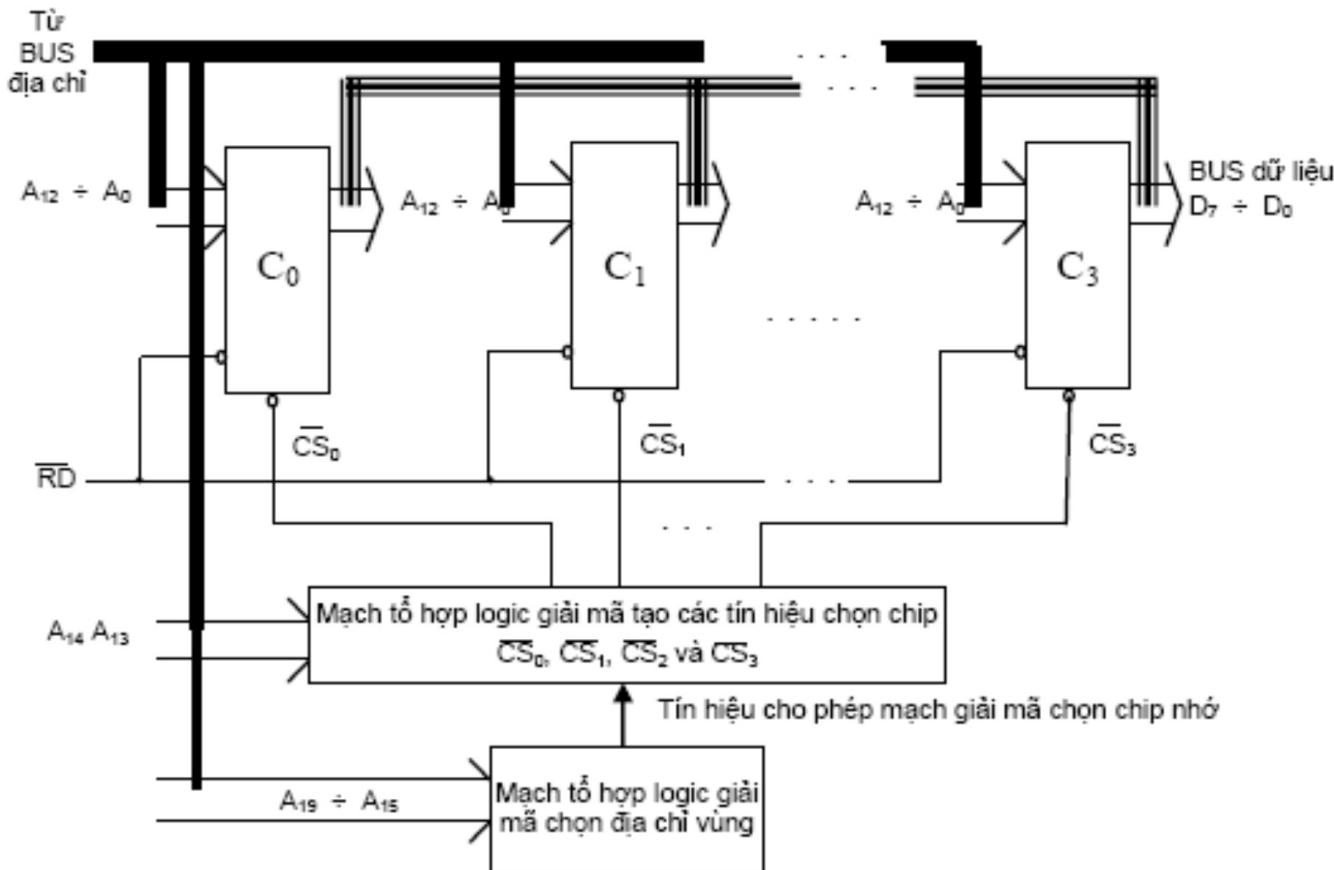
❖ Design DRAM

DRAM MK 4164 is 64K bit DRAM in 1 Chip. Assuming we will design RAM for the 8085 CPU with a maximum RAM of 64 KB, 8 Chips will be needed.



❖ ROM/EPROM

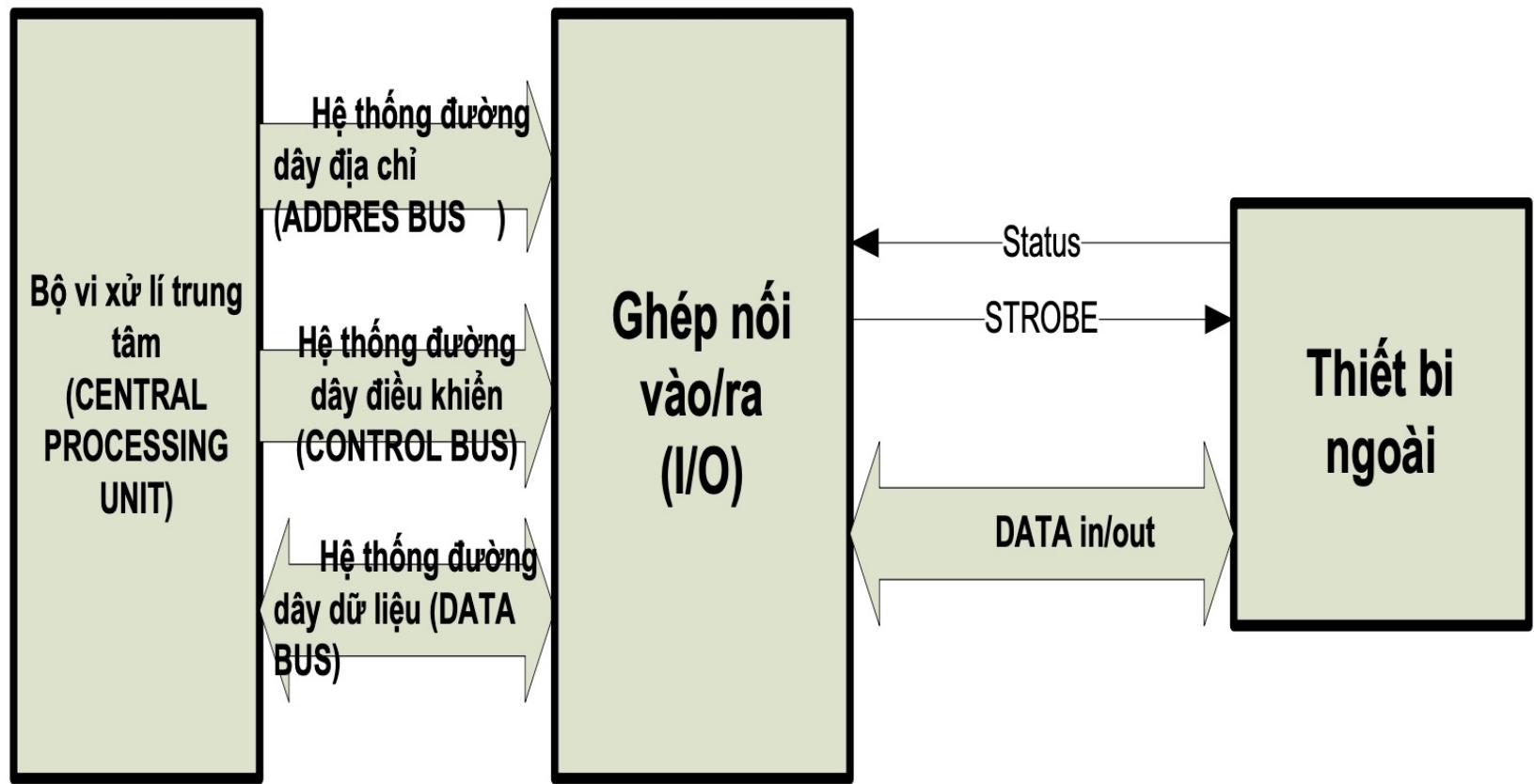
Build a ROM module with a capacity of 32KB, using Chip 2764 8K x 8 bits, the first address is 20000hex. The application program loads into this module.



Chapter 2: Hardware components

2.3: Pair with peripheral devices

Model of coupling technique

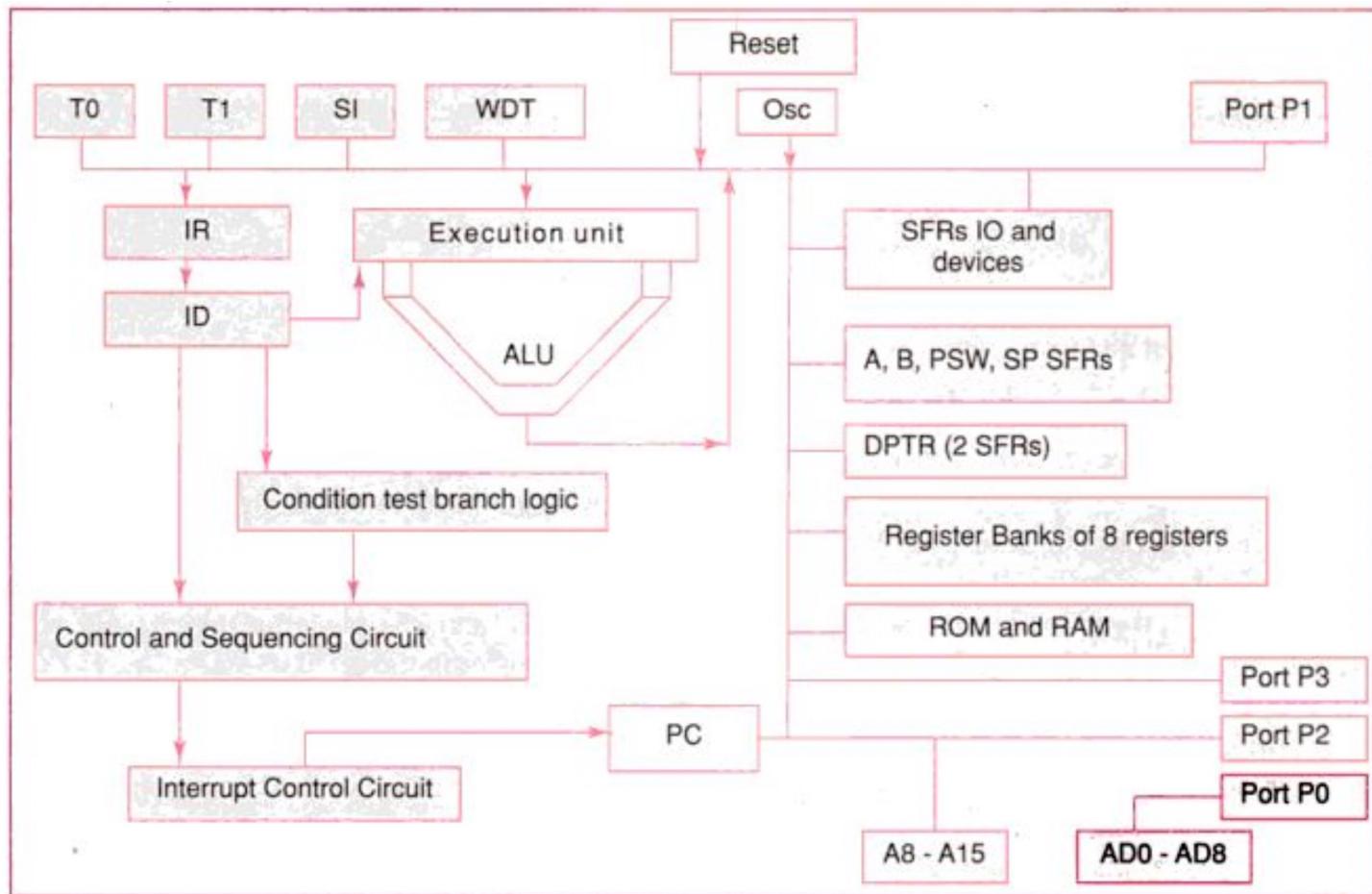


Chapter 2: Hardware components

2.4: 8051 microprocessor: Architecture, memory and practical interface

8051 microcontroller architecture

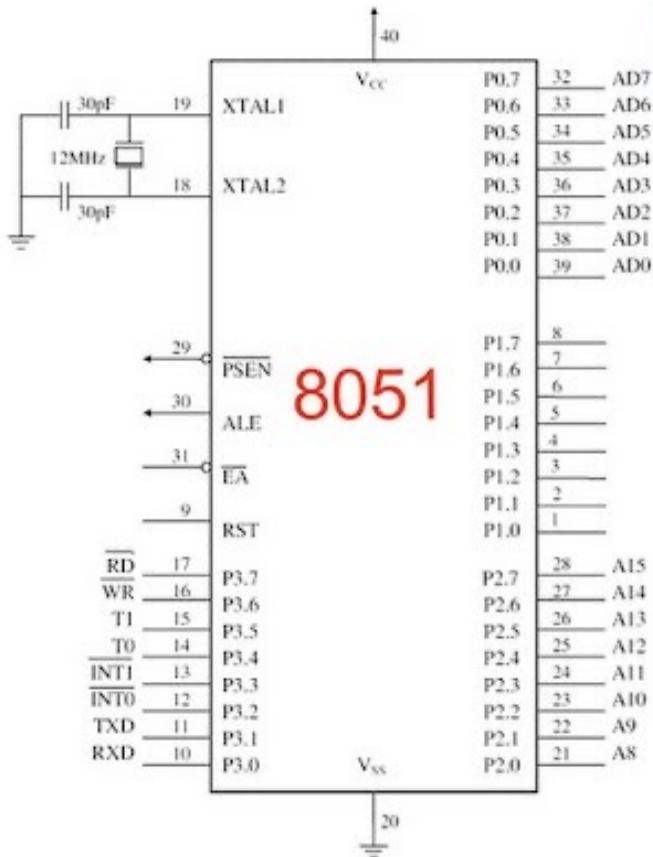
- ❖ Basic 8051 architecture: microprocessor, registers, memory (in Harvard architecture) and ports, counters/timers, I/O, and interrupt handlers.



8051 microcontroller architecture

❖ Characteristics of 8051

- 12 MHz clock. Processor instruction cycle time 1 μ s.
- ALU-8bit
- Harvard Memory Architecture
- Internal 8-bit data bus and Internal 16-bit address bus
- CISC instruction set architecture

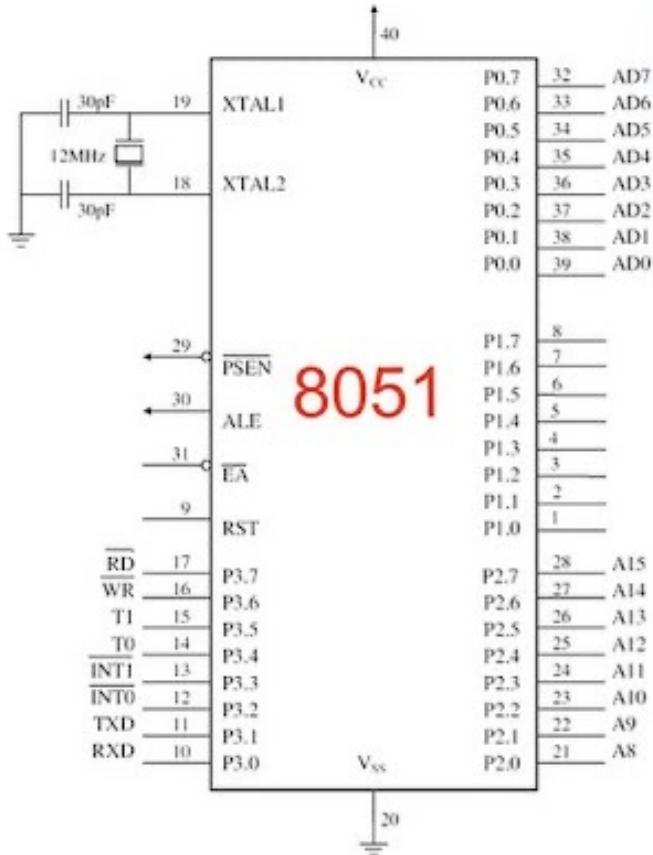


8051 microcontroller architecture

❖ Characteristics of 8051

■ Special Registers (SFR)

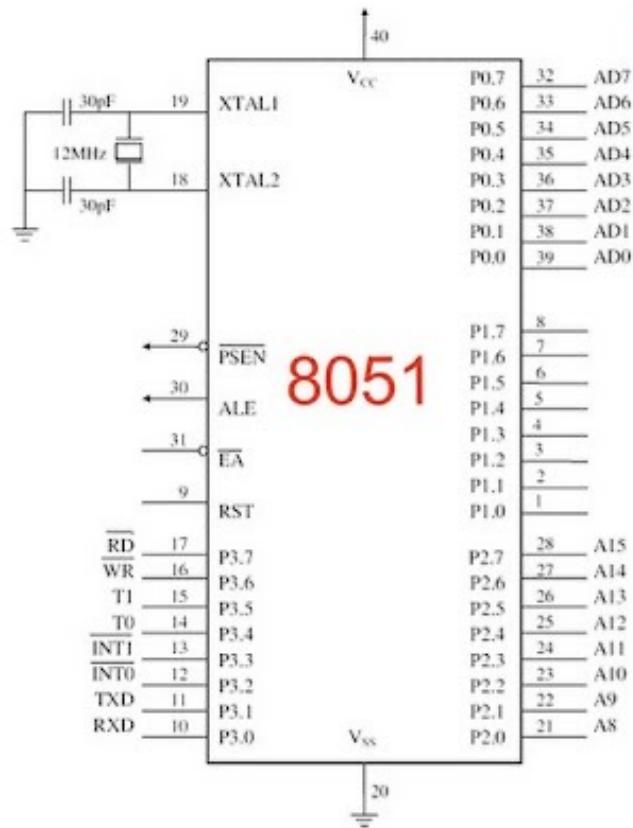
- PSW (status word)
- A (cumulative)
- Register B, SP (stack pointer)
- Registers for I/O, timers, ports, and interrupt handlers
- 16-bit program counter (PC) with factory default value of 0x0000.
- 8-bit stack pointer (SP) with an initial default value of 0x07



8051 microcontroller architecture

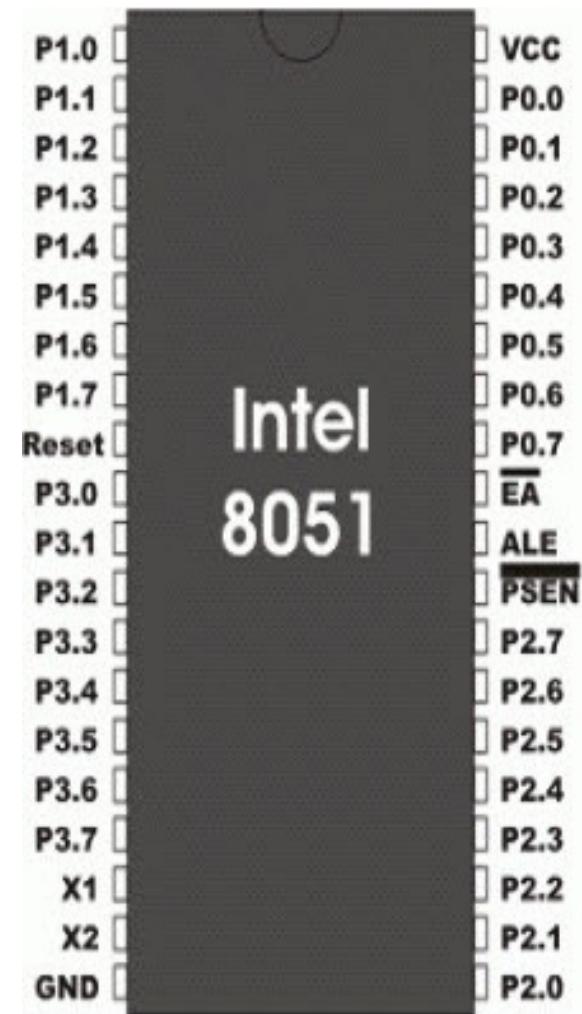
❖ Characteristics of the simple 8051:

- Does not handle floating point,
- No Cache,
- There is no MMU memory management unit,
- There is no atomic operations unit.
- There is no CPU Pipeline engineering
- Do not process commands in parallel.



❖ Features of 8051:

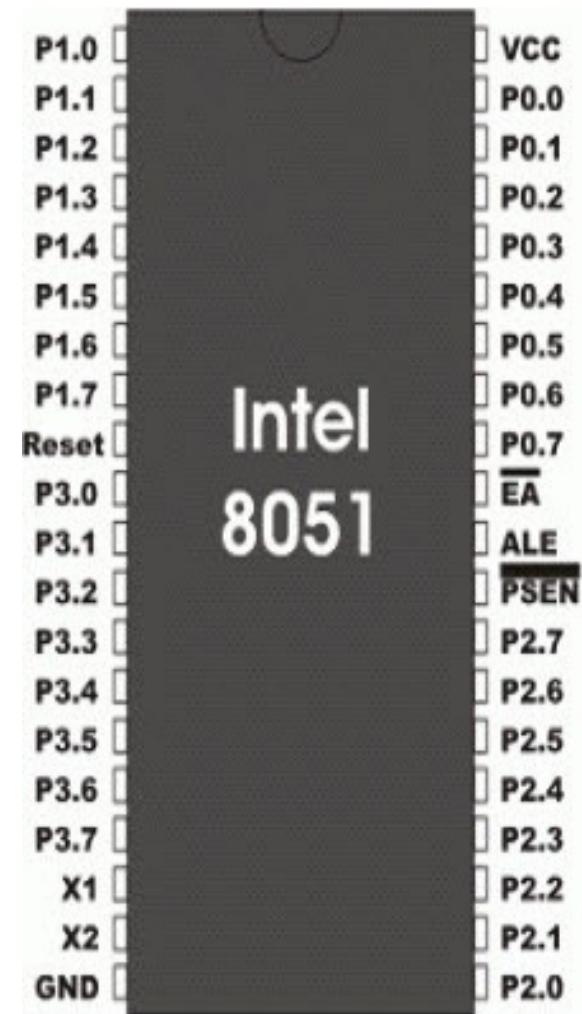
- RAM size 128bytes
- The 32 bytes of RAM are used as four register sets (register-set/bank). Each bank has 8 registers.
- Stack memory/External data memory can be added up to 64 kB.



8051 microcontroller architecture

❖ Some improved versions of the 8051

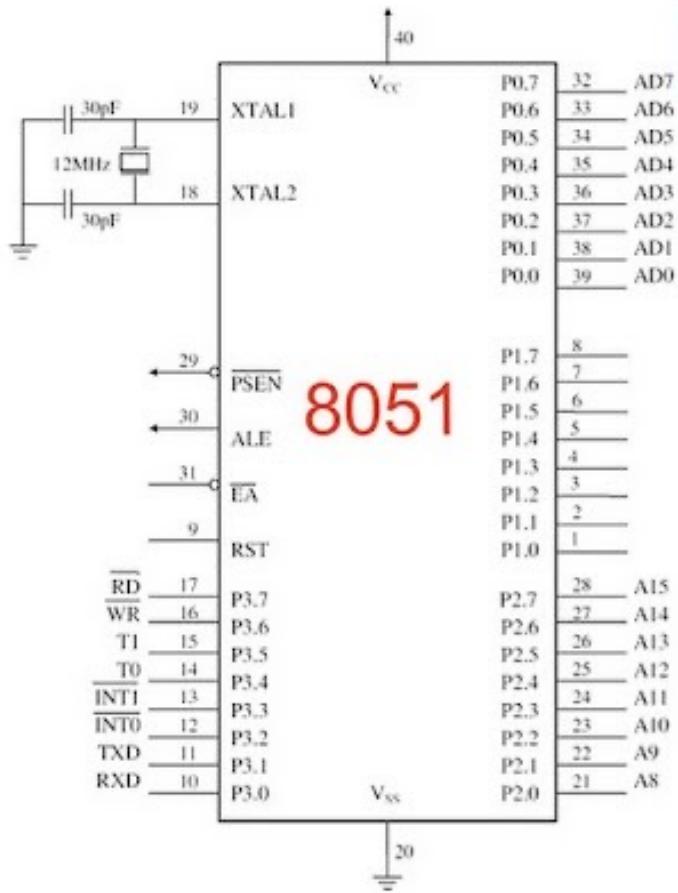
- Version 8351 has on-chip ROM
- Version 8751 uses EPROM
- Version 8951 uses on-chip EEPROM or 4 kB flash memory.
- In the extended 8051 versions the address space is expanded up to 16 MB.



8051 microcontroller architecture

❖ Features of 8051:

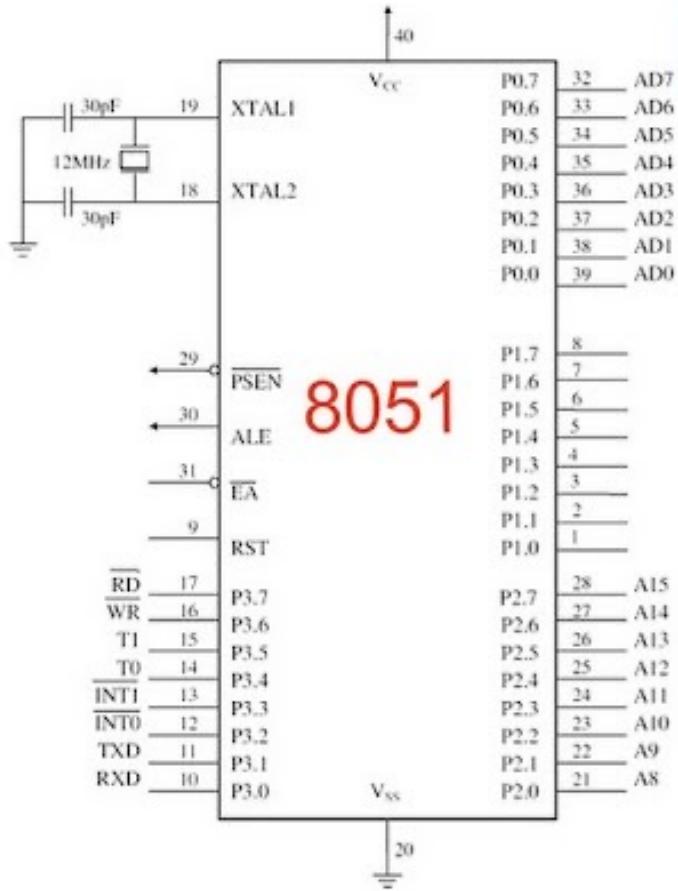
- Two interrupt pins 12 and 13 INT0 and INT1.
- Four ports of 8 bits each (P0, P1, P2, P3)
- Two timers T0, T1.
- Serial Interface (SI) – programmable for three full-duplex UART modes for I/O



8051 microcontroller architecture

❖ Features of 8051:

- In some versions – there is a DMA controller, a pulse width modulator
- In some versions of the 8051, modems, watchdog timers, and ADCs were also integrated.
- For example, the Siemens SAB 80535-N supports an ADC with a programmable reference voltage.

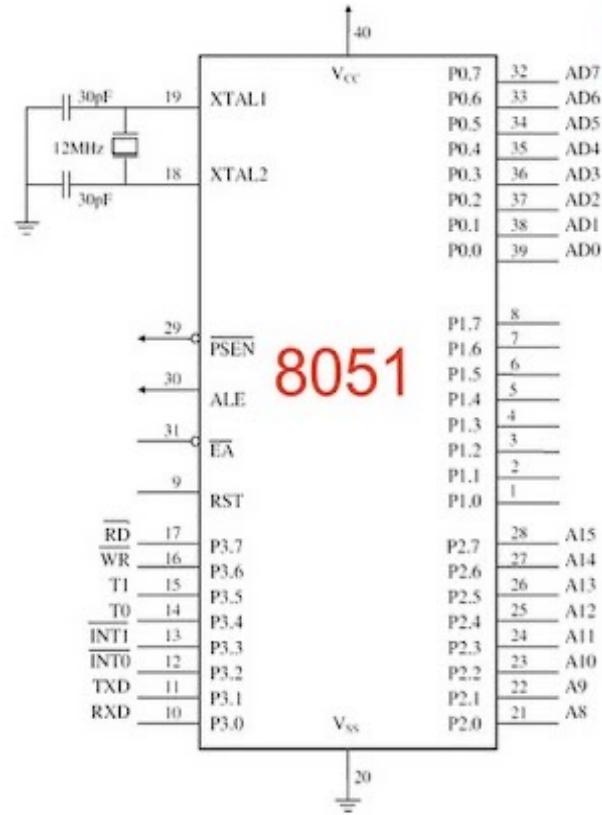


8051 microcontroller architecture

8051 INSTRUCTION SET ARCHITECTURE

❖ Data transfer commands are:

- Move bytes between register A and another register
- Move bytes from one register/from internal RAM to another register
- Indirect move:
 - MOVC indirect
 - MOVX indirect
- Instant migration:
 - MOV immediate DPTR
- Push or Pop direct

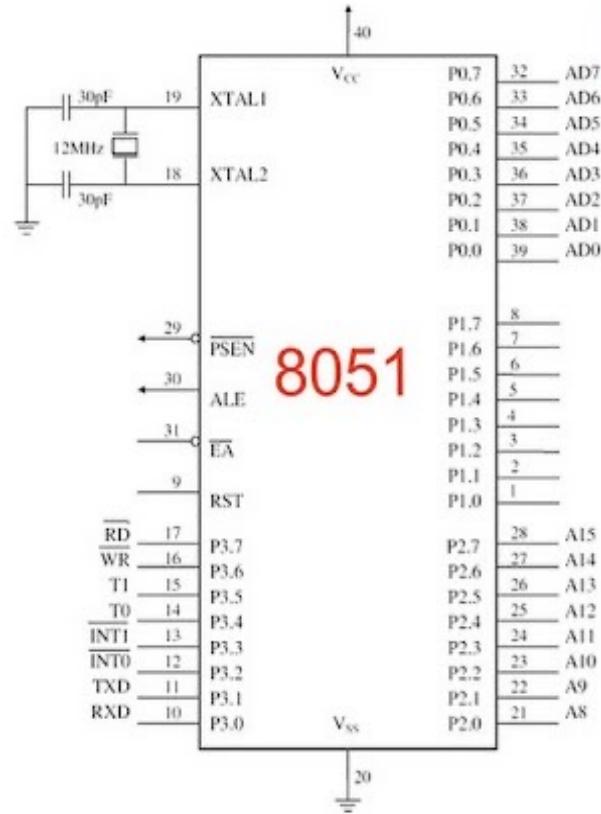


8051 microcontroller architecture

8051 INSTRUCTION SET ARCHITECTURE

❖ Manipulates bits, bytes, and logic instructions

- Bit manipulation:
 - Set value, add value with commands like AND, OR or MOV (Clr, Set, Mov...)
- Logical command:
 - Logical instructions AND, OR, XOR
- Byte operations:
 - Delete, add or swap commands and rotation commands (CLR, RL, RC...)

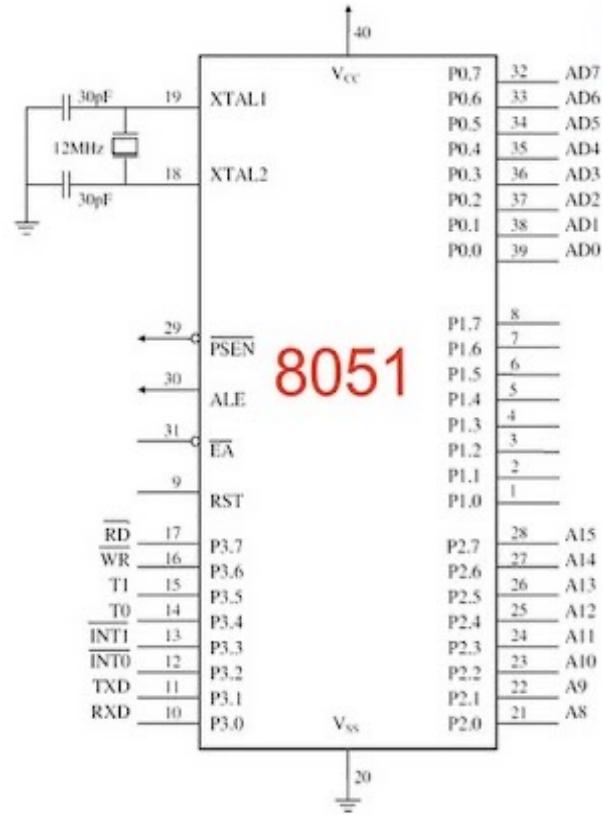


8051 microcontroller architecture

8051 INSTRUCTION SET ARCHITECTURE

❖ Math commands:

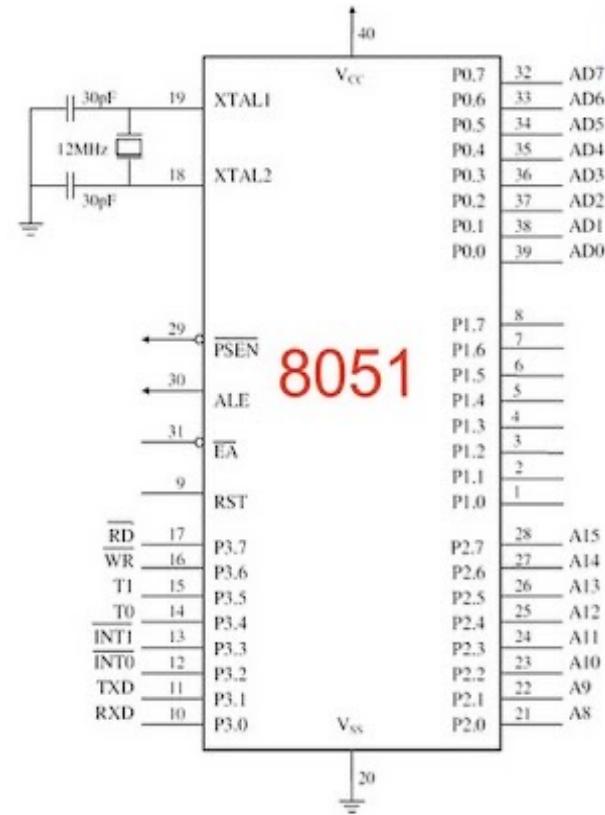
- These are 8-bit instructions
 - Add (Add), subtract (Sub), multiply (Mul), divide (div)
- Increase or decrease orders:
 - Inc (Increment), Dec (Decrement)



8051 INSTRUCTION SET ARCHITECTURE

❖ Program flow control commands:

- Branching
- Conditional jump command
- Comparison command
- Subprogram call
- NOP command
- Delay command
- Break command:
 - Flow control interrupt mask bits, priority bits
 - RETI



8051 I/O Port

❖ 8051 has 4 input and output ports: P0, P1, P2, P3:

- It is an 8 bit input and output port
- The addresses are 0x80, 0x90, 0xA0, 0xB0 respectively
 - Each bit of port P0 is denoted P0.0 to P0.7 corresponding to addresses 0x80 to 0x87.
- Similar to P1, there are 8 ports P1.0 to P1.7 corresponding to addresses 0x90 to 0x97. P2.0 to P2.7 are addresses 0xA0 to 0xA7, P3.0 to P3.7 are addresses 0xB0 to 0xB7.
- For example:
 - MOV 0xA0, #0xF ; means assigning value to port P2 bits is 0000 1111₂
 - INC 0xA0 ; P2 = 0000 1111₂ + 1 = 0001 0000₂

8051 I/O Port

❖ P0 và P1 ports

P0

P0.0 P0.1 P0.2
P0.3 P0.4 P0.5
P0.6 P0.7
Also as
AD0- AD7

P1

P1.0 P1.1 P1.2
P1.3 P1.4 P1.5
P1.6 P1.7
Also P1.6 as
I²C clock, P1.7
as I²C serial
data, and P1.0
and P1.1 for T2
(8052)

- The pins of P0 are both the low 8 bits of the address bus and the data bus

8051 I/O Port

❖ P2 và P3 ports

P2

P2.0 P2.1 P2.2
P2.3 P2.4 P2.5
P2.6 P2.7
Also as
A8- A15

P3

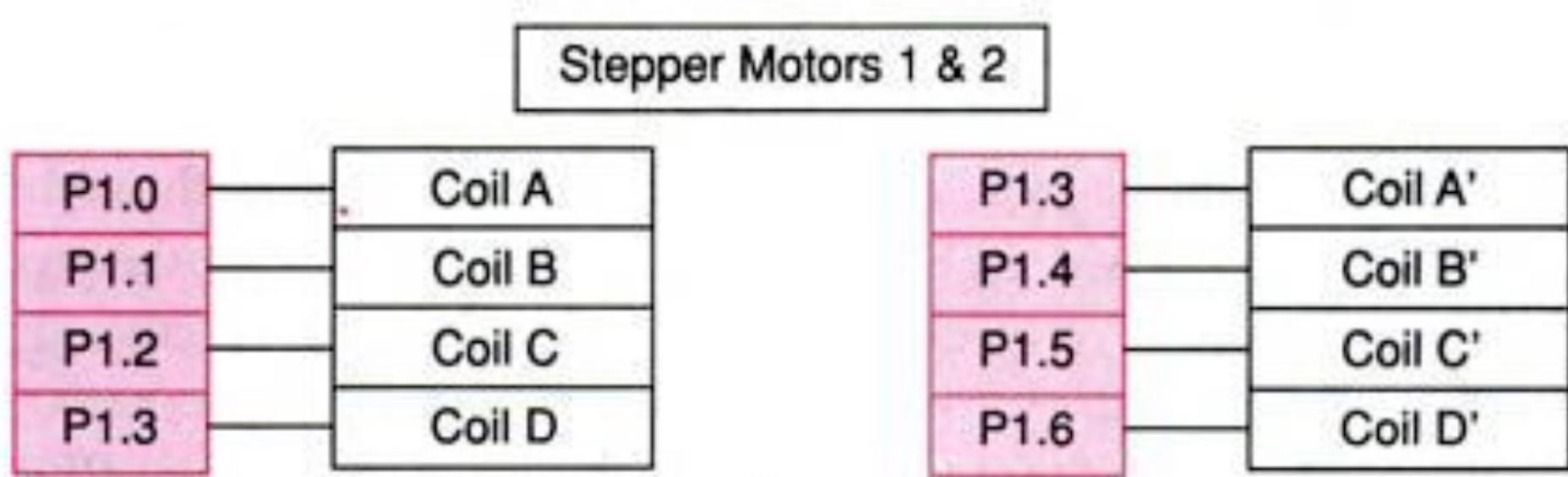
P3.0 P3.1 P3.2
P3.3 P3.4 P3.5
P3.6 P3.7
Also
RxD/SyncData,
TxD/SyncClk,
INT0/GT0,
INT1/GT1, T0,
T1, WR, RD

- The pins of P2 are the high 8 bits of the address bus

Input and output circuit of 8051

8051 I/O Circuit

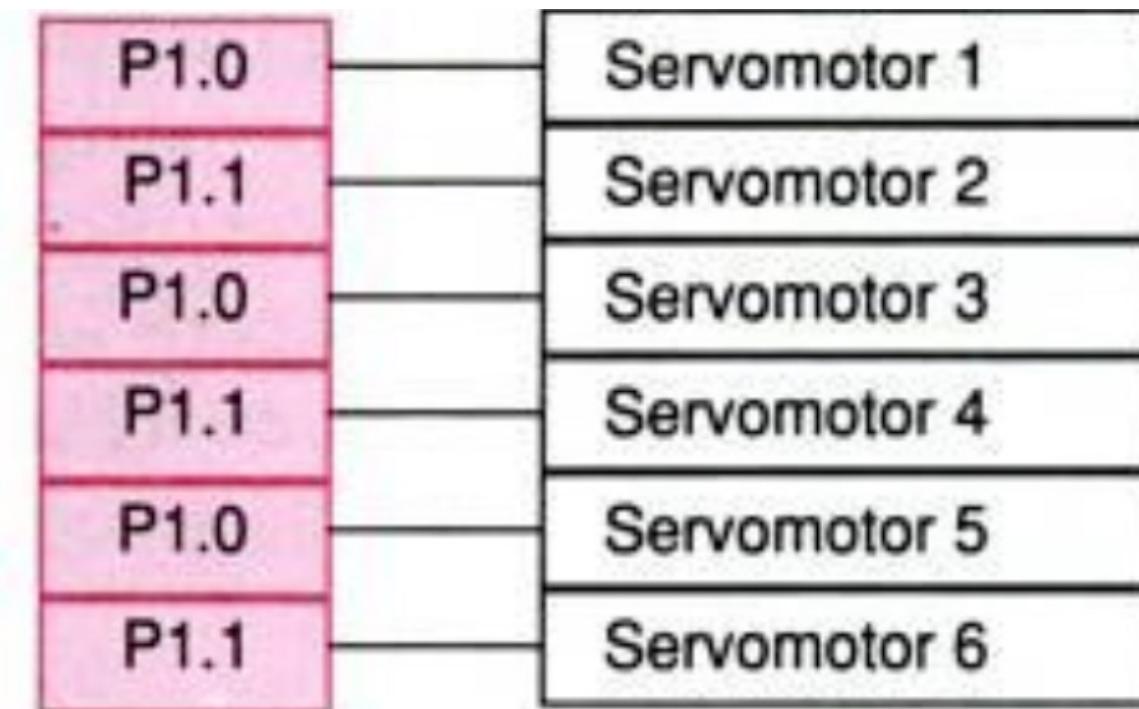
- The circuit of input and output ports controls stepper motors used in printers



Input and output circuit of 8051

8051 I/O Circuit

- ❖ Circuit of input and output ports for six servo motors in the robot



Byte programming using 8051 ports

- ❖ Byte addresses of I/O ports P0, P1, P2 and P3 of the 8051 used to access and execute read or write or other commands
- ❖ The direct 8-bit address of each address is given in the instructions
 - For example: MOV 0xA0, #0x0F ; means assigning values to the bits of port P2 is 0000 11112
- ❖ The addresses of the bytes at P0, P1, P2 and P3 are 0x80, 0x90, 0xA0 and 0xB0 respectively.

Byte programming using 8051 ports

- ❖ For example:
 - MOV 0xA0, #0x0F ;
 - INC 0xA0 ;
- ❖ That means assigning values to the bits of port P2 is 0000 11112
- ❖ Then port P2 will be increased by 1 value
- ❖ $P2 = 0000\ 11112 + 1 = 0001\ 00002$

Byte programming using 8051 ports

- ❖ Ports P0, P1, P2 and P3 each have 8 bits
- ❖ Each bit has an address to access and perform reading or writing using bit manipulation instructions.
- ❖ Address is the bit address of the pins. Each bit address is specified in the corresponding instruction.
 - Bit addresses P0.0 to P0.7 correspond to 0x80 to 0x87.
 - Bit addresses P1.0 to P1.7 correspond to 0x80 to 0x97.
 - Bit addresses P2.0 to P2.7 correspond to 0xA0 to 0xA7.
 - Bit addresses P3.0 to P3.7 correspond to 0xB0 to 0xB7.

Byte programming using 8051 ports

❖ For example:

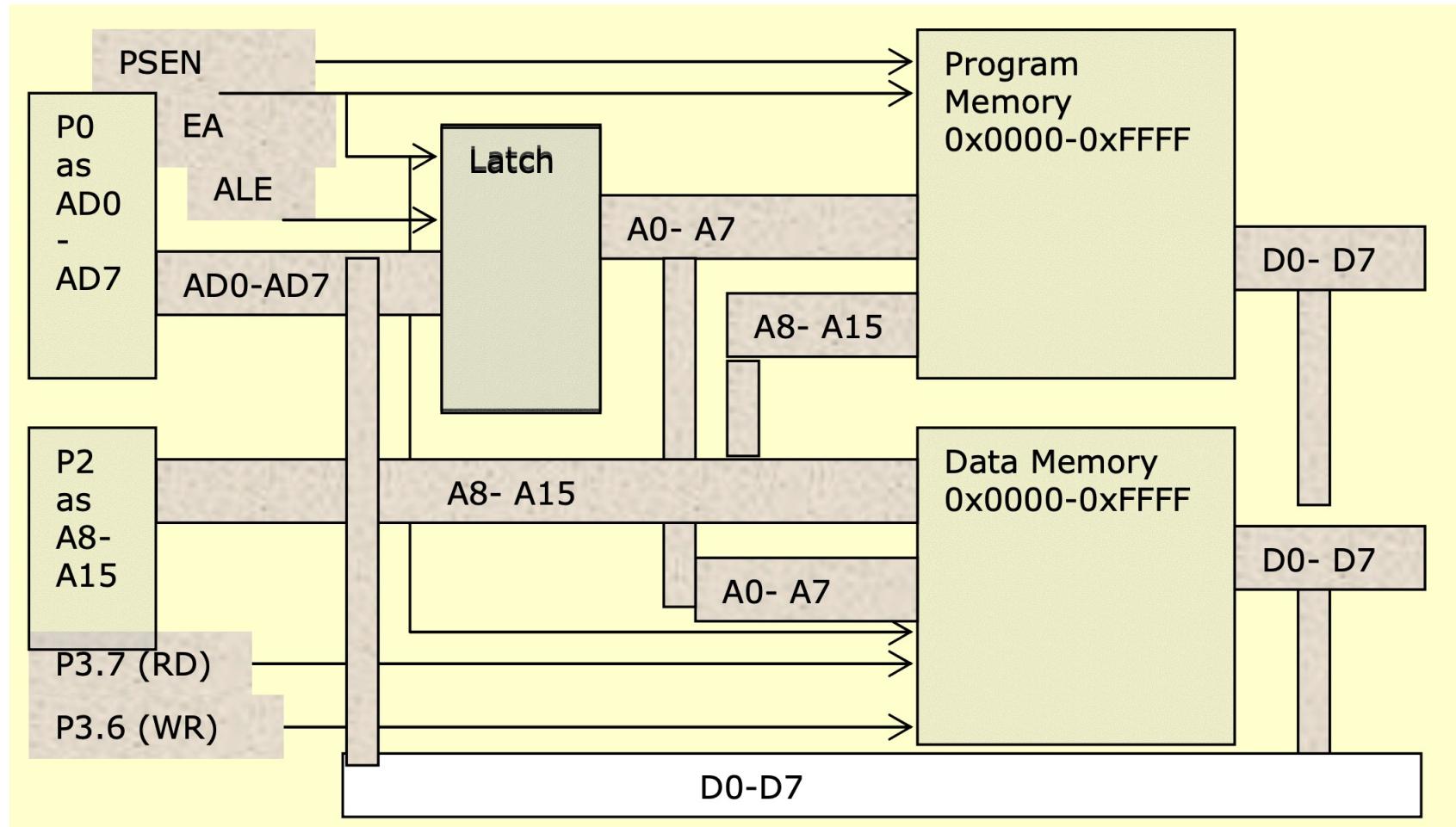
- CPL 0x90 ; Port P1 is assigned the value 0
- CLR 0x80 ; P0.0 has a value of 0.
- SETB 0x80; P0.0 has a value of 1.
- CLRB 0x80; Assign P0.0 to a value of 1.

Mapping in and out of memory

- ❖ The memory and ports in the 8051 are both assigned addresses, each port has its own address range in the data memory address space.
- ❖ The communication circuitry is identically designed for the memory to connect to the external and programmable peripheral (PPI) ports.

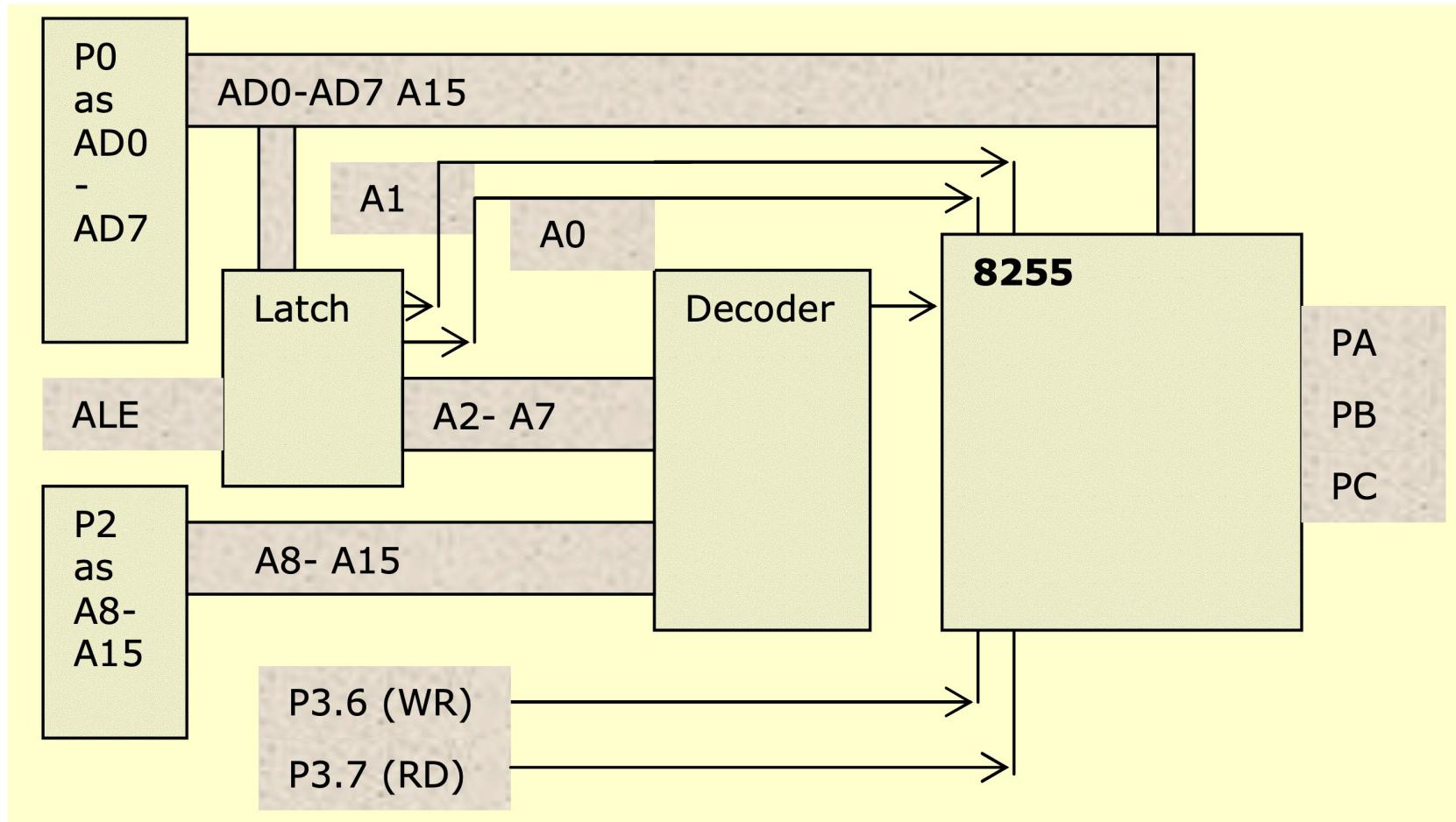
Pair with external storage

Connection to external program and data memory circuits



Pair with external storage

The interface uses the PPI external port of the 8051



Port P0 in extension mode and ALE signal

- ❖ The AD0-AD7 pins of port P0 are multiplexed signals of the low address bits A0-A7 of the address bus and bits D0-D7 of the data bus.

- ❖ Time division multiplexing of signals A0-A7 and D0-D7 based on the address latch enable (ALE) processor signal

Harvard memory architecture

- ❖ There are two memories — program memory and data memory.
- ❖ Two control signals — PSEN and RD to control reading from program memory or data memory.
- ❖ ALE control signal to control the use of AD0-AD7 as address or data at a given instance

EA control signal

- ❖ Is a signal that when enabled – the processor always accesses external memory addresses instead of internal memory or register addresses

External address RAM, Register, program memory when EA control signal is not active

- ❖ VXL always accesses external memory whether EA is active or not.
- ❖ RAM addresses and registers are between 0x00 and 0xFF just like external data memory addresses are between 0x0000 and 0xFFFF.
- ❖ Internal program memory addresses 0x0000 to 0xFFF (in the case of 4 kB internal ROM) are the same as external program memory addresses 0x0000 and 0xFFFF.

Counter/timer devices

- ❖ The original 8051 family had two timers T0 and T1
- ❖ The 8052 family (extension of 8051) has three timers T0, T1 and T2
- ❖ **Counting/timing device used as a timer**
- ❖ A timing device when counting inputs are provided by a clock.
- ❖ Clock pulses are given at specified intervals: acting as a timer.
- ❖ **Counting/timing device used as counter**
- ❖ A counting device when the inputs are for counting. The counter is provided with input to count from the external input pin.

- ❖ Timer T0 and T1
- ❖ TMOD special register
- ❖ TCON special register on four bits
- ❖ Four external P3 pins for external control and external counting input
- ❖ Modes 0, 1, 2, 3 of T0
- ❖ Modes 0, 1 and 2 of T1

Timing and counting devices External controls to activate or deactivate running

- ❖ When timing or counting devices are externally controlled by gate input, when GT0 or GT1 is externally triggered, the device can operate otherwise it will shut down in gate input mode.
- ❖ The GT0 or GT1 signal is given at P3.2 and P3.3.

Counter/timing device External counting input in counter mode

- ❖ T0 count T0 is given input for counting from the external input T0 pin at P3.4.
- ❖ T1 counts when T1 is given input to count from external input pin T1 at P3.5.

Two special registers TH1-TL1

- ❖ To access the number or time of the higher 8 bits and lower 8 bits of device T1
- ❖ Two 16-bit storage registers of the T1 device.

Two special registers TH0-TL0

- ❖ To access the number or time of the higher 8 bits and lower 8 bits of device T0
- ❖ Two registers store 16 bits of device T0.

TMOD special register

- ❖ Controls T1 and T0 modes using 4 bits per mode, programming count/timing of T1 and T0.
- ❖ One bit in each function controls whether the external port input controls or not.
- ❖ A bit that controls the function for which counter or timer mode is used.
- ❖ Two bits control the function mode of the timer/counter such as mode 0 or 1 or 2 or some other action

Special register TCON Control and status bits T1 and T0

- ❖ The four bits 4 above program the counting/timing device modes T1 and T0.
- ❖ TCON.7 and TCON.5 show the timer/counter overflow status for T1 and T0, respectively.
- ❖ TCON.6 and TCON.4 control the start and stop of the timer/counter
- ❖ The lower bits of TCON are for interrupt control for INT0 and INT1

Timer/Counter T0

- ❖ 8 register bits TMOD (lower 4 bits), TCON (bits 5 and 4), TL0 (count/time bit), TH0 (count/time bit)
- ❖ Counter with input at P3.4 when bit 2 TMOD = 1, timer with internal clock timing input when bit 2 TMOD = 0
- ❖ When mode is set = 0, 8-bit Timer/Counter mode and TH0 are used and TL0 is used to pre-scale (divide) the input by 32
- ❖ When mode is set = 1, 16-bit timer/counter mode with TH0-TL0 is used for timing or counting

Timer/Counter T1

- ❖ 8 bits used Register TMOD (upper 4 bits), TCON (bits 7 and 6), TL1 (count/time bit), TH1 (count/time bit)
- ❖ Counter with input at P3.5 when bit 6 TMOD = 1, timer with internal clock timing input when bit 6 TMOD = 0
- ❖ When mode is set = 0, 8-bit Timer/Counter mode and TH1 are used and TL1 is used to pre-scale (divide) the input by 32
- ❖ When mode is set = 1, timer/set mode đếm 16 bit với TH1-TL1 được sử dụng để định thời hoặc đếm

Timer/Counter T1

- ❖ When mode is set = 2, the 8-bit Timer/Counter TH1 is used and TL1 is used to automatically reload TH1 after timeout using the preset value at TL1
- ❖ When mode is set = 3, T1 stops because TH0 is now active instead of T1.

- ❖ Serial Interface function (Serial Interface)
- ❖ Half-duplex synchronous serial mode 0
- ❖ Full duplex asynchronous UART mode 1, 2 or 3
- ❖ SBUF
- ❖ SCON

Serial Interface SI

Programming for:

- SI serial interface
- full-duplex asynchronous UART mode

Two special 8-bit registers

- ❖ SBUF (8 bit serial receive or transmit bit register depending on whether the command is using SBUF as source or destination)
- ❖ SCON (8-serial mode cum control bit register) and SFR PCON.7 bit

SBUF

- ❖ Unique register address for the transmitted and received byte buffer when serial output or input is sent.
 - 0x99 address of SI buffer.
 - The register holds the 8-bit SI line as it is written.
- ❖ For example:
 - MOV 0x99, instruction A writes A to the transfer buffer from register A
 - MOV R1, read command 0x99 Register R1 from receive buffer

SCON

- ❖ Register to control the SI interface.
- ❖ The three bits above program modes such as 0 or 1 or 2 or 3.
 - Mode 0 is full duplex synchronization.
 - Mode 1 or 2 or 3 is full duplex asynchronous mode.
 - Bit SCON.4 enables or disables the SI receiver function.
 - The two bits SCON.3 and SCON.2 specify the 8th bit transmitted and the 8th bit received when the mode is 2 or 3. One bit SCON.1 enables or disables the SI transmitter interrupts (TI) when complete the transmission.
 - Bit SCON.0 enables or disables SI receiver interrupts (RI) on completion of transmission.

Mode function 0 – Input or output

- ❖ Depends on the instruction using SBUF as source or destination
 - Synchronous serial mode data and clock input
 - Synchronous serial mode data and clock output
- ❖ Mode 0 when SCON bits 7 and 6 (mode bits) are 00