

CHƯƠNG 1

GIỚI THIỆU HỆ VI XỬ LÝ-VI ĐIỀU KHIỂN

1.1 Giới thiệu chung

- Thiết kế số sử dụng nhiều vi mạch SSI cổng logic, vi mạch MSI giải mã, MUX, DEMUX..., FF và thanh ghi...→phần cứng phức tạp ,không thay đổi được chức năng đã thiết kế
- Năm 1971 Intel đưa ra chip vi xử lý(VXL, Microprocessor,CPU) 4 bit đầu tiên→tích hợp phần cứng gọn hơn và có thể thay đổi chức năng thiết kế bằng cách lập trình.
- Họ VXL 8 bit chuẩn đầu tiên của Intel là 8080, Motorola 6800 ,Zilog Z80...
- Họ VXL phát triển lên 16 bit,32 bit,64 bit...
- Các vi mạch VXL mạnh về tính toán,xử lý dung lượng dữ liệu lớn
- Để thiết kế các hệ thống,mạch điện nhỏ chuyên một chức năng hoặc vài chức năng như đo&điều khiển nhiệt độ,độ ẩm,điều khiển tốc độ động cơ,hiển thị,thu nhận data....,cần phần cứng tích hợp cao,dễ giao tiếp và dễ lập trình theo hướng điều khiển→họ vi điều khiển(VĐK , Microcontroller,MCU)

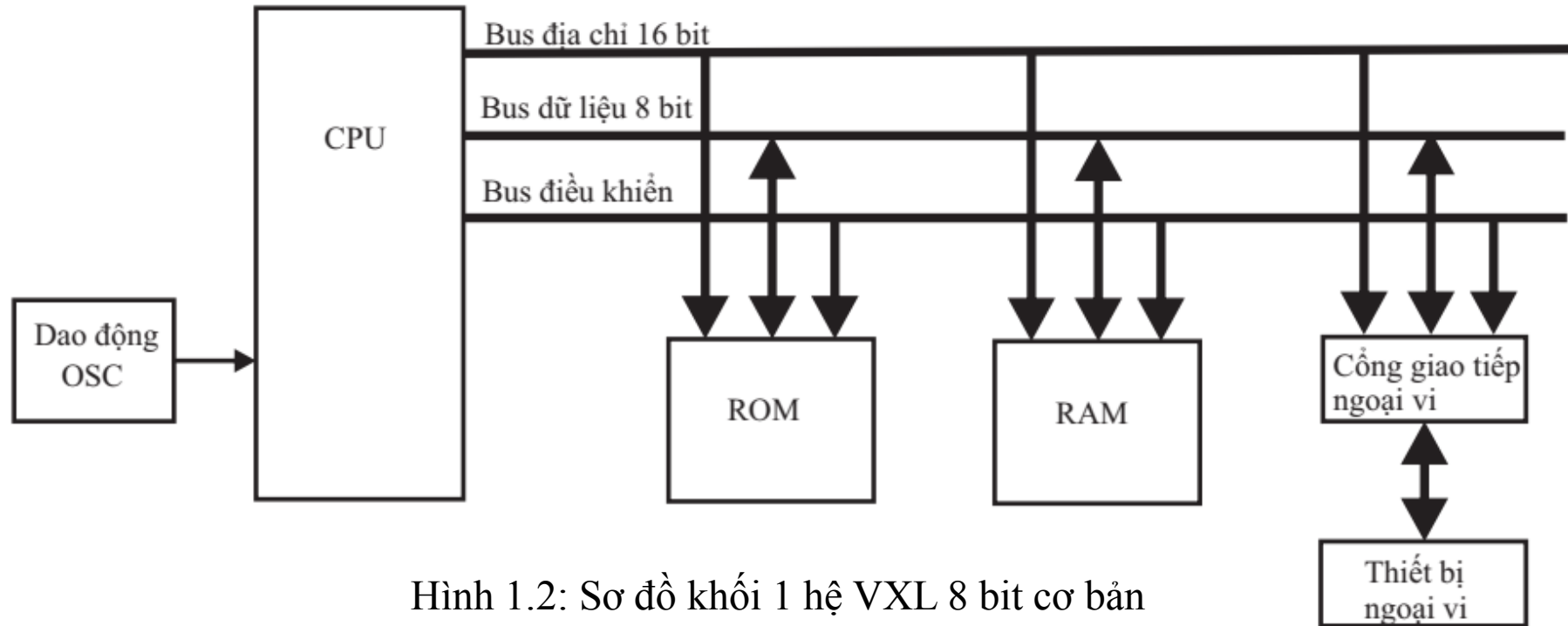
1.1 Giới thiệu chung

- Năm 1976 Intel giới thiệu họ MCU8748
- Năm 1980 Intel giới thiệu họ MCU8051, VĐK 8 bit chuẩn
- Các họ VĐK PIC 16FXX, AVR...chuyên một chức năng đơn giản gọi là RISC(Reduced Instructions Set Computer)
- Họ VĐK đa năng 8,16,32,64 bit ARM
- Thiết kế điện tử hiện nay= VXL/VĐK + lập trình
- Họ AVR là hệ VĐK RISC do hãng Atmel chế tạo từ năm 1996, trong đó tiêu biểu thông dụng nhất là dòng ATmega 8 bit.
- Trong giáo trình sẽ nghiên cứu AVR ATmeg324P tiêu biểu cho họ AVR 8 bit. Trên cơ sở đó, SV có thể tự nghiên cứu và áp dụng họ AVR và các hệ VXL/VĐK khác.
- Các ví dụ về mã lệnh, tập lệnh cụ thể đều dựa trên tập lệnh AVR ATmega324P



Hình 1.1: VĐK ATmega324P dạng PDIP

1.2 Sơ đồ khối hệ VXL cơ bản



Hình 1.2: Sơ đồ khối 1 hệ VXL 8 bit cơ bản

1. Bộ xử lý trung tâm CPU(Central Processing Unit)
2. Bộ nhớ ROM(Read Only Memory) và RAM(Random Access Memory)
3. Hệ thống bus: địa chỉ(Adress), dữ liệu(Data),điều khiển(Control)
4. Mạch giao tiếp ngoại vi (Interface Circuitry)
5. Thiết bị ngoại vi(Peripheral device)
6. Mạch dao động

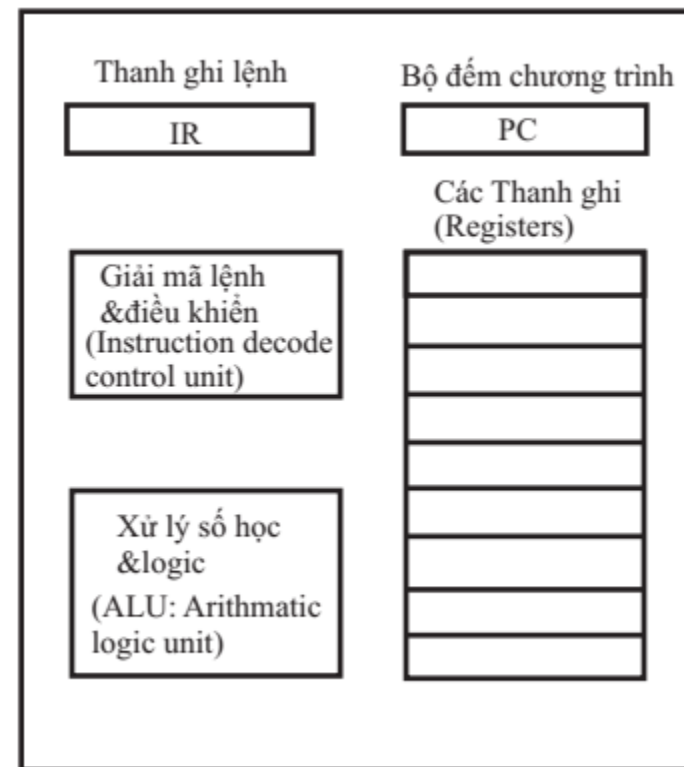
1.2 Sơ đồ khối hệ VXL cơ bản

1.2.1 Bộ xử lý trung tâm CPU

- Là bộ não kiểm soát toàn bộ hoạt động của hệ thống
- Nhận lệnh là các mã nhị phân từ bộ nhớ, thực thi lệnh và lưu kết quả vào bộ nhớ/ngoại vi hoặc các thanh ghi trong CPU

Cấu hình cơ bản của CPU gồm:

1. Bộ đếm chương trình PC(Program Counter): giữ địa chỉ bộ nhớ chứa mã lệnh
 2. Thanh ghi lệnh IR(Instruction Register): chứa mã lệnh cần giải mã để thực thi
 3. Bộ giải mã lệnh và điều khiển ID&CU(Instruction decode and Control Unit): giải mã lệnh và tạo tín hiệu điều khiển các khối khác thực thi lệnh
 4. Bộ số học logic ALU(Arithmetic and Logic Unit): thực thi lệnh, các phép toán số học và logic
 5. Các thanh ghi(Registers): chứa tạm thời các dữ liệu, quản lý các chức năng khác của hệ thống
- Các khối trên được kết nối với nhau qua hệ thống bus dây nối bên trong CPU



Hình 1.3: Cấu trúc cơ bản CPU

1.2 Sơ đồ khối hệ VXL cơ bản

1.2.2 Bộ nhớ ROM và RAM

- Chứa mã lệnh(code) chương trình và dữ liệu(data)
- ROM là bộ nhớ chỉ đọc không ghi /xóa được, không mất dữ liệu khi ngừng cấp nguồn
- Thông thường chỉ dùng để chứa chương trình cố định như khởi động, quản lý hệ thống.
- Các loại ROM:
 - ☐ - Mask ROM(ROM mặt nạ): được lập trình cố định từ hãng sản xuất, sử dụng khi chương trình ổn định không cần điều chỉnh, sản xuất hàng loạt, giá thành rẻ.
 - ☐ - PROM(Programmable ROM): là ROM lập trình bằng điện chỉ 1 lần(OTP: One Time Programed)
 - ☐ - EPROM(Electrical Programmable ROM): là ROM lập trình bằng điện, xóa bằng tia UV, có thể nạp/xóa nhiều lần
 - ☐ - EEPROM(Electrically Erasable Programmable ROM): là loại ROM có thể nạp/xóa bằng điện nhiều lần

1.2 Sơ đồ khối hệ VXL cơ bản

1.2.2 Bộ nhớ ROM và RAM

- RAM là bộ nhớ có thể đọc /ghi được, mất dữ liệu khi ngừng cấp nguồn
- Thời gian truy xuất RAM nhanh hơn so với ROM gấp nhiều lần
- Thông thường dùng RAM để chứa dữ liệu hoặc chương trình ứng dụng nạp từ bên ngoài

Các loại RAM:

- RAM tĩnh(Static RAM): dung lượng thấp, giải mã địa chỉ đơn giản
- RAM động(Dynamic RAM): dung lượng cao, thời gian truy xuất ngắn hơn so với RAM tĩnh, giải mã địa chỉ phức tạp, cần chu kỳ làm tươi bộ nhớ.
- CPU truy xuất bộ nhớ thông qua hệ thống bus dây ngoài kết nối với các chân địa chỉ, dữ liệu và điều khiển của bộ nhớ.

1.2 Sơ đồ khối hệ VXL cơ bản

1.2.3 Hệ thống bus

- Bus là tập hợp các đường dây nối cùng chức năng, kết nối các khối bên ngoài với CPU
- Bus địa chỉ kết nối các ngõ địa chỉ của bộ nhớ và ngoại vi với các ngõ địa chỉ của CPU
 - - CPU nhận diện truy xuất bộ nhớ hoặc ngoại vi thông qua địa chỉ
 - - Truyền data là địa chỉ trên bus địa chỉ một chiều từ CPU
 - - CPU 8 bit thường có bus địa chỉ 16 bit → khả năng truy xuất $2^{16} = 65536$ địa chỉ
- Bus dữ liệu kết nối các ngõ data của bộ nhớ và ngoại vi với các ngõ data của CPU
 - - Truyền data trên bus dữ liệu là hai chiều: data từ CPU → ngoài là ghi(WR), data từ ngoài → CPU là đọc(RD)
 - - Các ngõ data có thể là ngõ vào hoặc ngõ ra(I/O: input/output), bus data kết nối chung → tình trạng các ngõ ra nối chung(wiredAND) gây ngắn mạch! → sử dụng mạch ngõ ra 3 trạng thái(tristate), tích cực khi CPU truy xuất đến(thông qua địa chỉ) và hở mạch(hi-Z) khi CPU không truy xuất đến
- Bus điều khiển gồm các đường tín hiệu điều khiển hoạt động độc lập
 - - Các tín hiệu cơ bản như reset(RST), đọc(/RD), ghi(/WR), ngắt(/INT), xung nhịp(CK)...

1.2 Sơ đồ khối hệ VXL cơ bản

1.2.4 Mạch giao tiếp ngoại vi

- Còn gọi là cổng ngoại vi (I/O port) kết nối CPU với thiết bị ngoại vi
 - - Cổng ngoại vi có thể là mạch chốt D, mạch đệm 3 trạng thái, mạch chuyển đổi song song/nối tiếp...
 - - Ngõ ra cổng ngoại vi giao tiếp bus data với CPU phải có dạng 3 trạng thái (CPU nhìn ngõ ra cổng ngoại vi như là ngõ vào)
 - - Cổng ngoại vi phải có địa chỉ truy xuất như bộ nhớ

1.2.5 Thiết bị ngoại vi

- Giao tiếp hệ VXL với thế giới bên ngoài thông qua cổng ngoại vi
 - - Các thiết bị ngoại vi cơ bản cho PC: màn hình, bàn phím, chuột, ổ đĩa, máy in...
 - - Hệ thống điện tử còn có các thiết bị ngoại vi như cảm biến, nút nhấn, LED, LCD, rơ le, động cơ, loa, ADC, DAC...

1.2.6 Mạch dao động: có thể bên ngoài đưa vào CPU hoặc nằm trong CPU

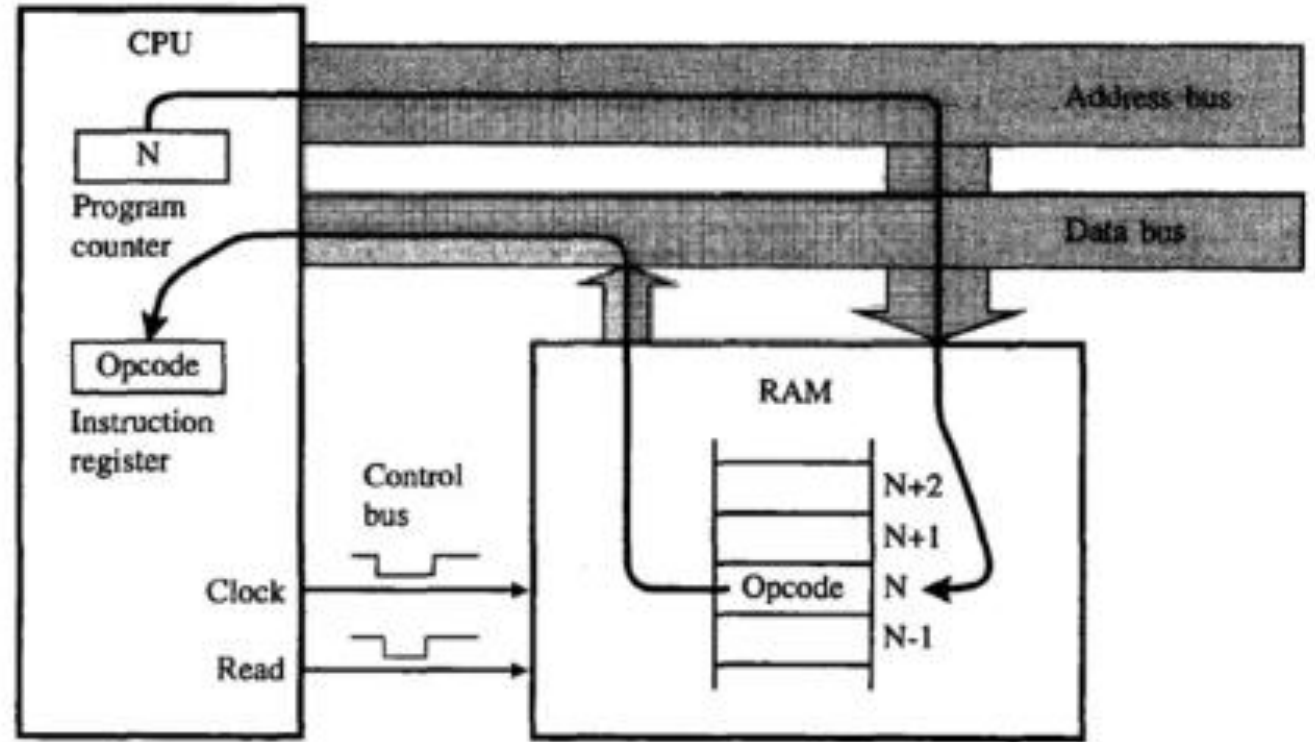
- - Tạo xung nhịp CK đồng bộ các hoạt động của CPU và các khối khác
- - Tạo xung định thì, đếm sự kiện...

1.3 Chu kỳ thực hiện lệnh(chu kỳ bus)

Chu kỳ thực hiện lệnh chia làm 2 chu kỳ: nhận lệnh(fetching) và thực thi lệnh(execution)

1.3.1 Chu kỳ nhận lệnh(fetching)

- Xuất nội dung PC ra bus địa chỉ.
Nội dung PC=địa chỉ ô nhớ cần truy xuất mã lệnh(opcode)
- Xuất tín hiệu đọc kích hoạt bộ nhớ
 - Bộ nhớ được phép xuất data=mã lệnh(opcode) trong ô nhớ có địa chỉ tương ứng lên bus data
- Mã lệnh được đọc về cất trong thanh ghi lệnh IR
- Nội dung PC tăng đến địa chỉ lệnh kế tiếp(giá trị tăng tùy thuộc vào độ dài mã lệnh hiện hành)



Hình 1.4: Hoạt động chu kỳ nhận lệnh

1.3 Chu kỳ thực hiện lệnh(chu kỳ bus)

1.3.2 Chu kỳ thực thi lệnh(execution)

- Bộ giải mã và điều khiển giải mã mã lệnh trong IR
 - - Xuất tín hiệu điều khiển đến ALU và các thanh ghi có liên quan thực hiện lệnh
 - - Kết quả thực hiện lệnh có thể trả về các thanh ghi hoặc bộ nhớ data/ngoại vi
- Trường hợp kết quả trả về bộ nhớ/ngoại vi:
 - Địa chỉ bộ nhớ/ngoại vi chứa trong thanh ghi quy định được xuất ra bus địa chỉ
 - Nếu là lệnh ghi,data được xuất ra bus data cùng tín hiệu ghi,bộ nhớ/ngoại vi có địa chỉ tương ứng được phép nhận data
 - Nếu là lệnh đọc,tín hiệu đọc được xuất,bộ nhớ/ngoại vi có địa chỉ tương ứng được phép xuất data lên bus data,CPU đọc data về cất trong thanh ghi chỉ định.

➤ Lưu ý về PC hiện hành(PCHH)

Trong chu kỳ thực thi lệnh PC đã tăng đến địa chỉ lệnh kế tiếp

→PCHH=địa chỉ lệnh kế tiếp

1.3 Chu kỳ thực hiện lệnh(chu kỳ bus)

1.3.2 Chu kỳ thực thi lệnh(execution)

Ví dụ 1.1: Giả sử có các nội dung của các thanh ghi: PC=0100H,R20=23H,R16=15H với R20 và R16 là tên các thanh ghi trong CPU.Trong bộ nhớ chứa chương trình tại địa chỉ 0100H và 0101H lần lượt chứa giá trị 40H và 0FH là mã lệnh ADD R20,R16 (cộng nội dung thanh ghi R20 và R16 trả kết quả về thanh ghi R20).Hãy mô tả chi tiết chu kỳ thực hiện lệnh trên.

Giải:

I. Chu kỳ nhận lệnh: xem hình 1.5a,b

1. PC=0100H được xuất ra bus địa chỉ

2. Xuất tín hiệu đọc bộ nhớ /RD ra bus điều khiển

3. Ô nhớ địa chỉ 0100H được truy xuất nội dung=40H(opcode byte thấp) lên bus data

4. Data 40H được đọc về cát trong IR : IR=0040H

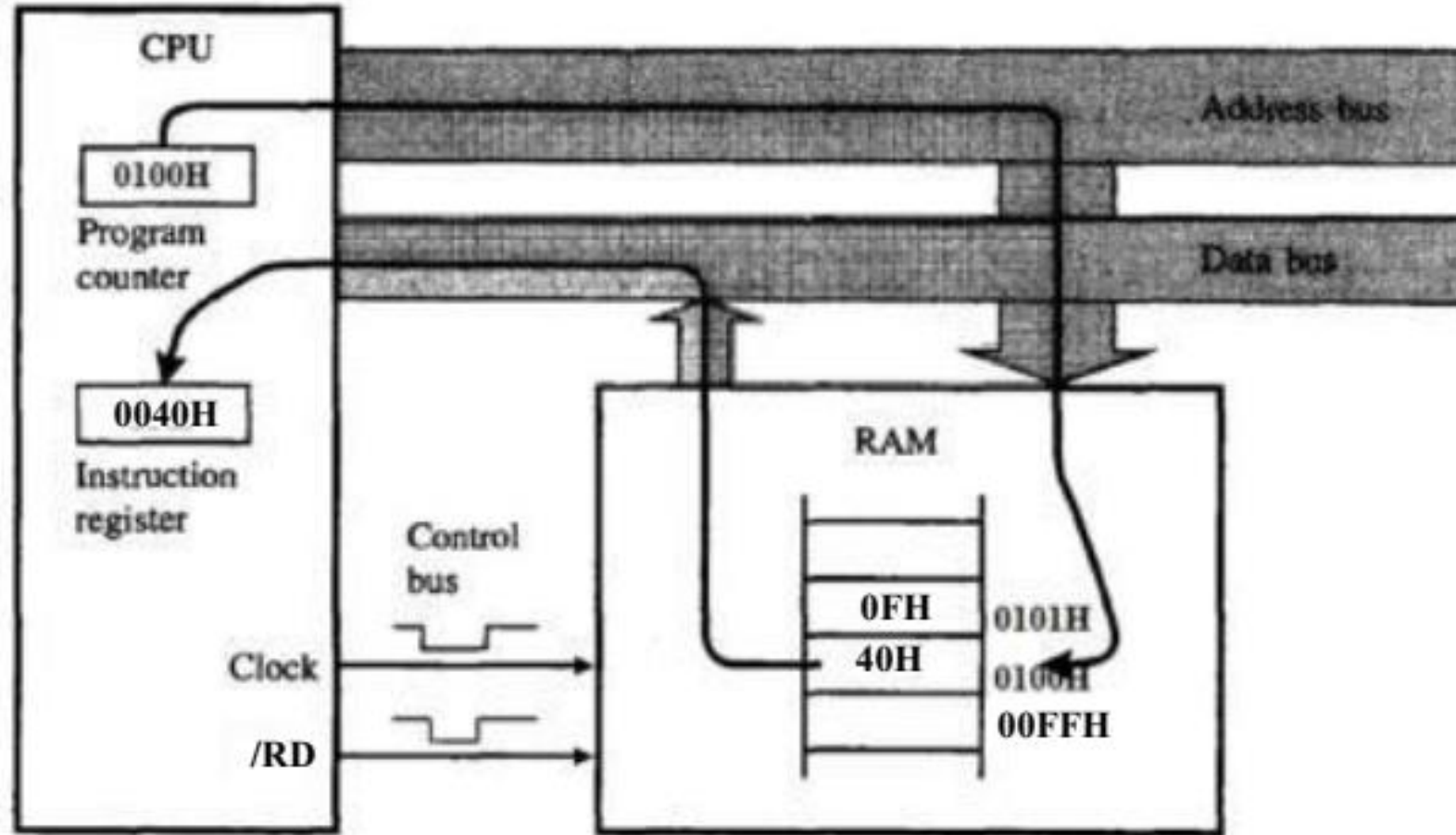
5. Lệnh này dài 2 byte nên PC tăng thêm 1→PC=0101H,CPU tiếp tục thực hiện như từ bước 1 đến bước 4 đọc nội dung ô nhớ 0101H là 0FH(opcode byte cao) về cát trong IR.

Cuối cùng IR=0F40H chính là mã lệnh ADD R20,R16.

6. PC tăng thêm 1 trở vào địa chỉ ô nhớ chứa lệnh kế tiếp→PCHH=0102H

1.3 Chu kỳ thực hiện lệnh(chu kỳ bus)

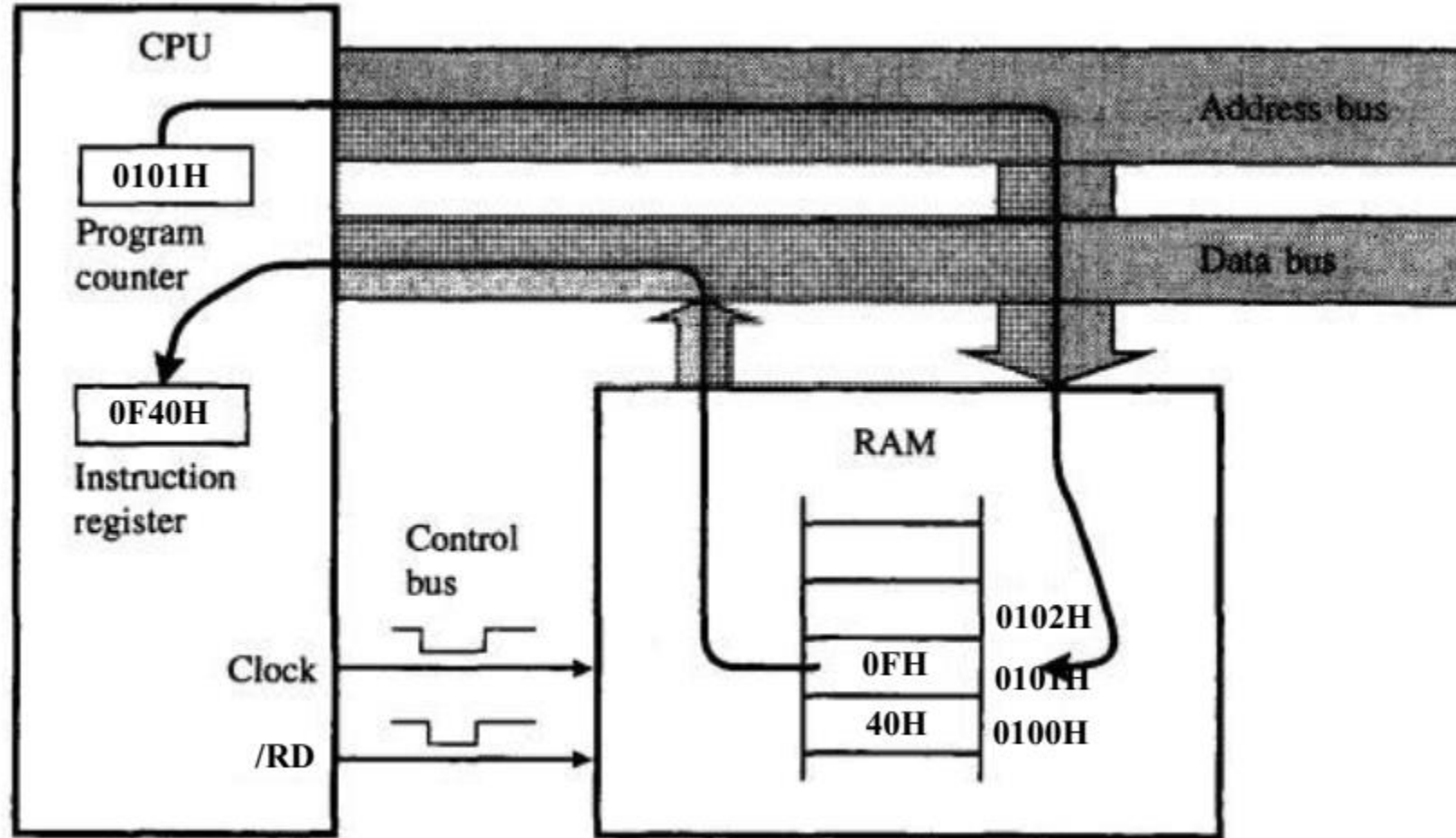
1.3.2 Chu kỳ thực thi lệnh(execution)



Hình 1.5(a): Chu kỳ nhận lệnh byte thấp

1.3 Chu kỳ thực hiện lệnh(chu kỳ bus)

1.3.2 Chu kỳ thực thi lệnh(execution)



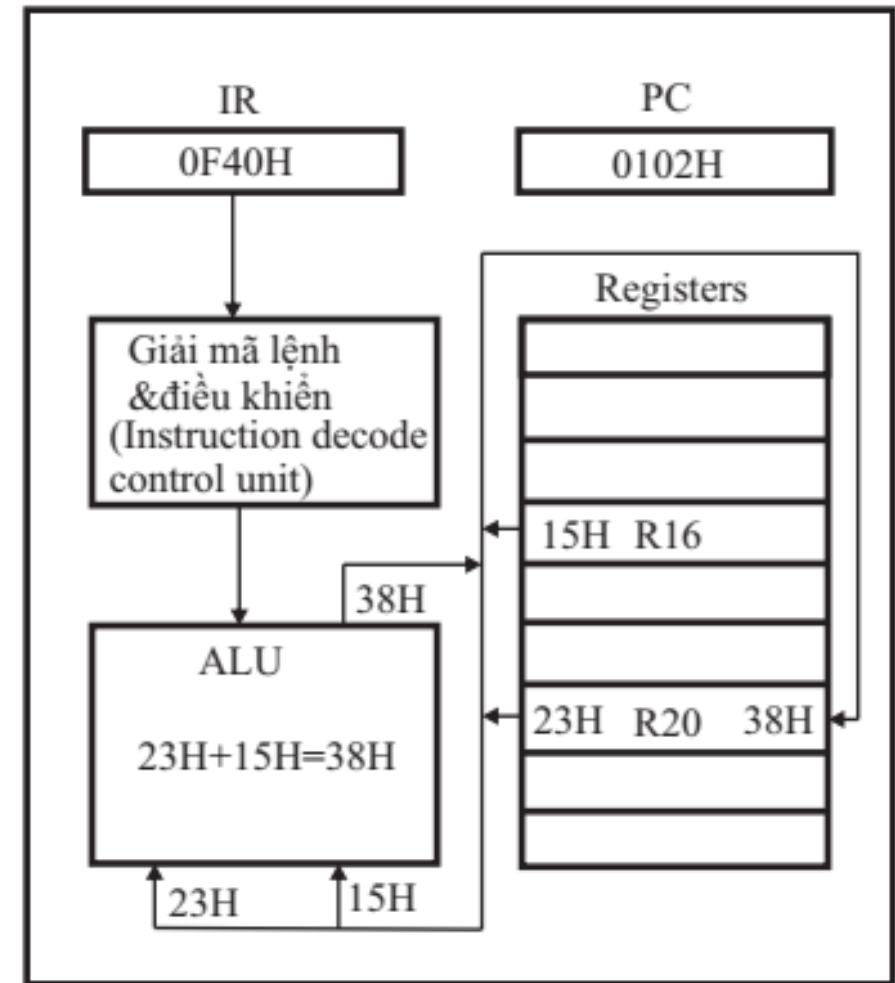
Hình 1.5(b): Chu kỳ nhận lệnh byte cao

1.3 Chu kỳ thực hiện lệnh(chu kỳ bus)

1.3.2 Chu kỳ thực thi lệnh(execution)

2. Chu kỳ thực thi lệnh

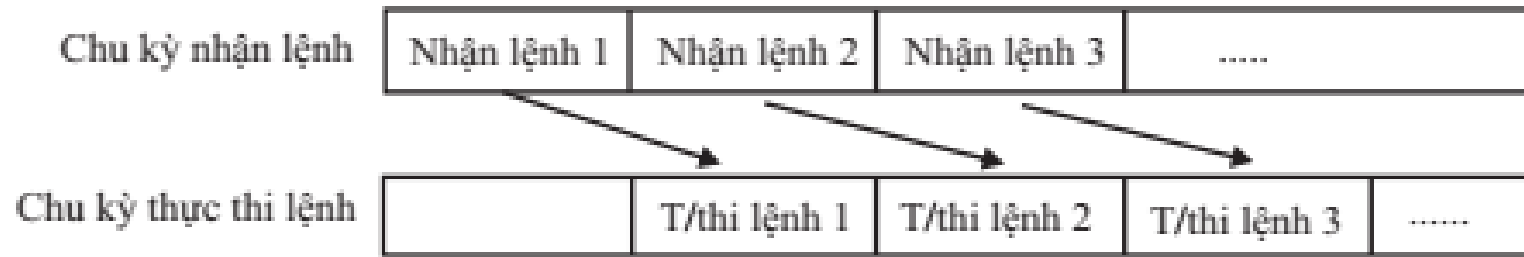
- ID&CU nhận mã lệnh 0F40H từ IR
giải mã $\rightarrow R20 + R16 \rightarrow R20$
- ALU nhận data từ $R20 = 23H$ và $R16 = 15H$
cộng lại cho kết quả $= 38H$
- ALU trả kết quả $38H$ về cất trong $R20$



Hình 1.5(c): Chu kỳ thực thi lệnh

1.3 Chu kỳ thực hiện lệnh(chu kỳ bus)

1.3.3 Kỹ thuật đường ống(pipeline)



Hình 1.6: Chu kỳ thực thi lệnh sử dụng kỹ thuật đường ống

- Để tăng tốc độ làm việc của CPU,sử dụng kỹ thuật đường ống:
 - Trừ chu kỳ nhận lệnh đầu tiên,từ chu kỳ nhận lệnh thứ 2,CPU đồng thời thực hiện chu kỳ thực thi lệnh 1,chu kỳ nhận lệnh thứ 3 đồng thời chu kỳ thực thi lệnh thứ 2....
 - Nếu xem thời gian thực hiện chu kỳ nhận lệnh và thực thi lệnh bằng nhau,tốc độ làm việc của CPU được tăng gấp đôi.

1.4 Chu kỳ dao động-Chu kỳ máy

1.4.1 Chu kỳ dao động

- Chu kỳ dao động còn gọi là chu kỳ dao động chủ, thường được ký hiệu là F_{osc} , là chu kỳ xung nhịp hay xung CK đồng bộ các khối trong CPU
- Chu kỳ dao động có thể bằng chu kỳ xung ngõ ra từ mạch dao động nội hay dao động thạch anh ngoài, hoặc bằng chu kỳ xung ngõ ra mạch dao động được chia với hệ số đặt trước.

Ví dụ AVR Atmega324P có mạch dao động thạch anh ngoài tần số 8Mhz, qua mạch chia hệ số 8 (lập trình được) cho chu kỳ dao động $F_{osc} = 8/8 = 1\text{Mhz}$.

1.4.2 Chu kỳ máy (MC: machine cycle)

- Là thời gian ngắn nhất thực hiện xong một lệnh
- Chu kỳ máy được tính theo chu kỳ xung CK.

Ví dụ: MCU8051 có $1\text{MC} = 12$ chu kỳ xung CK, MCU PIC8F452 có $1\text{MC} = 4$ CK

- MC càng ngắn tốc độ CPU càng cao, các RISC CPU thường có $1\text{MC} = 1$ chu kỳ xung CK

Ví dụ : Họ AVR có $1\text{MC} = 1$ CK

- Biết MC \rightarrow thời gian thực để thực hiện lệnh

1.5 Tập lệnh-Chương trình

1.5.1 Tập lệnh(Instruction sets)

- CPU chỉ hiểu và thực hiện mã lệnh dưới dạng mã nhị phân hay mã máy(machine language, opcode)
 - Theo ví dụ 1.1, mã lệnh 0F40H=0000 1111 0100 0000 là lệnh “cộng nội dung thanh ghi R20 và R16 trả kết quả về R20”
- Để dễ dàng cho người lập trình,nhà sản xuất dùng từ gợi nhớ (mnemonics)thay thế cho mã lệnh
 - Theo ví dụ 1.1, mã lệnh 0000 1111 0100 0000 tương đương lệnh ADD R20,R16
- Tập lệnh là tập hợp các mã lệnh dưới dạng từ gợi nhớ và mã máy tương ứng mà CPU có thể thực hiện.
 - - Mỗi CPU có một tập lệnh tương ứng(do nhà sản xuất quy định)

1.5 Tập lệnh-Chương trình

1.5.2 Chương trình(Program)

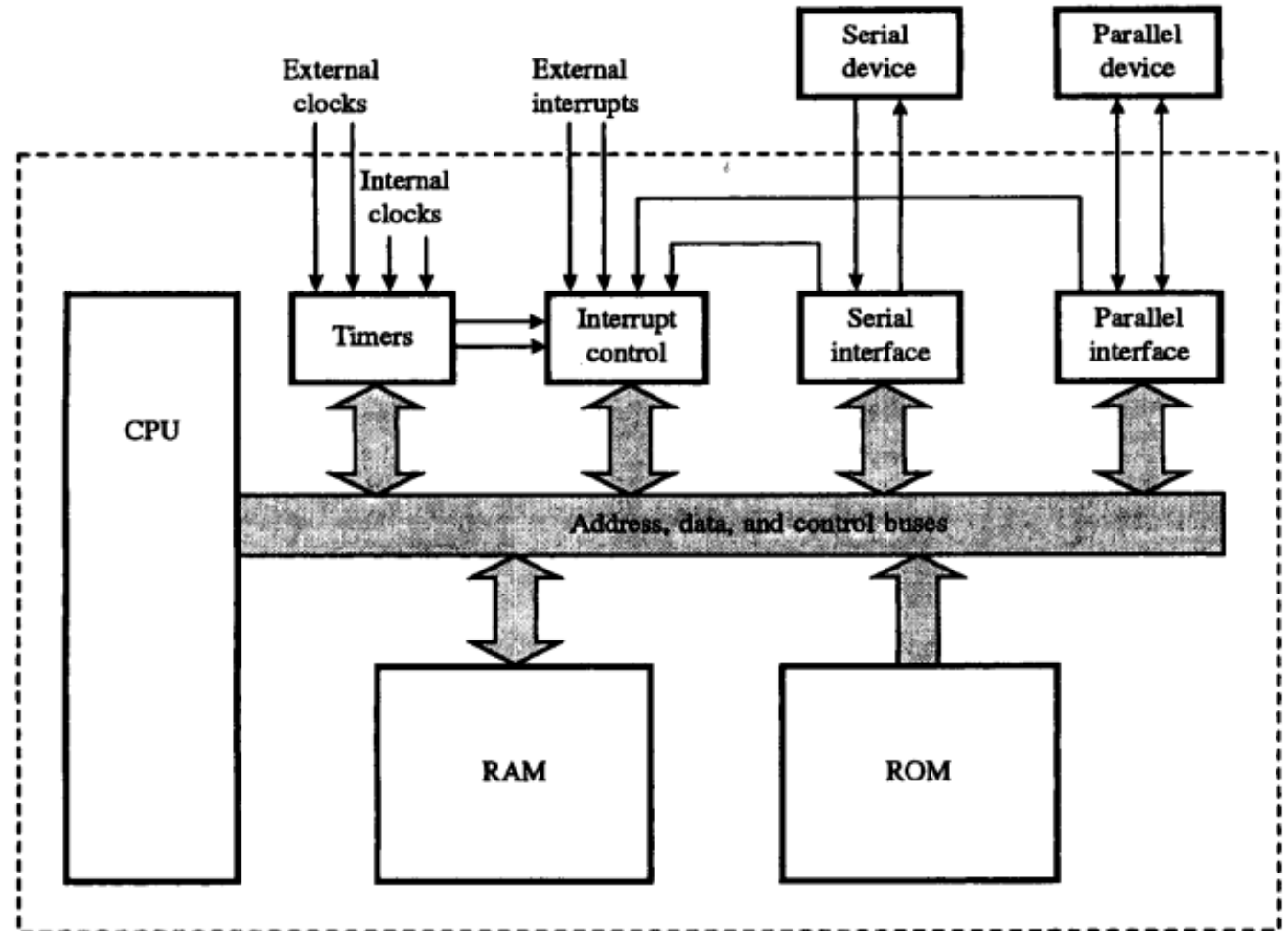
- Tập hợp các lệnh để CPU thực hiện một hoặc nhiều công việc hoàn thành chức năng của hệ thống gọi là chương trình
- Chương trình được viết bằng văn bản gồm các ký tự, ký hiệu theo quy định cú pháp(syntax) gọi là ngôn ngữ lập trình
- Ngôn ngữ lập trình : Hợp ngữ(Assembly) hay C

Ví dụ: Lập trình AVR có hợp ngữ AVR2 và C GCC trong Atmel Studio 7

- Trình biên dịch: chuyển chương trình nguồn dưới dạng văn bản sang mã máy thường dưới dạng file.bin hoặc file.hex CPU mới hiểu và thực hiện được.
- Sơ đồ khối hình 1.1 gọi là phần cứng(hardware): cố định
- Chương trình gọi là phần mềm(software): có thể điều chỉnh dễ dàng
- Các chương trình điều khiển hệ thống hoặc ứng dụng cơ bản cố định thường được biên dịch thành file.hex và nạp trực tiếp(bằng bộ nạp chương trình) vào bộ nhớ chương trình của MCU(ROM)thường được gọi là phần dẻo(firmware)

1.5 Từ VXL đến VĐK

- Chip VXL thường được hiểu là CPU
- Phân biệt VXL và VĐK qua: kiến trúc phần cứng, ứng dụng, tập lệnh



Hình 1.7: Sơ đồ khối cơ bản 1 chip VĐK(MCU)

1.5 Từ VXL đến VĐK

- **Kiến trúc phần cứng VĐK**

- VXL chỉ có CPU
- VĐK có tích hợp trên chip(onchip): CPU,ROM,RAM,Timer,Điều khiển ngắt,cổng ngoại vi song song,nối tiếp....

- **Ứng dụng**

- VXL thường sử dụng trong các hệ thống tính toán,mạnh về tính toán,xử lý dung lượng lớn data
- VĐK thường sử dụng trong các thiết bị điện tử,thiết kế phần cứng gọn,mạnh về giao tiếp và điều khiển

- **Tập lệnh**

- Tập lệnh VXL mạnh về tính toán trên byte,word,double words,định vị địa chỉ truy xuất bộ nhớ đa dạng, xử lý chuyển dời chuỗi,mảng data lớn,nhiều kiểu lệnh
- Tập lệnh VĐK nghiêng về điều khiển và giao tiếp ngoại vi,mạch về xử lý bit,mã lệnh đơn giản,dễ tính thời gian thực,số lượng lệnh ít

Hiện nay việc phân biệt VXL hay VĐK rất khó,tùy theo ứng dụng của chip được chế tạo!

1.5 Từ VXL đến VĐK

Các khái niệm mới

- Chương trình VĐK thường chỉ thực hiện một công việc nên dung lượng bé hơn chương trình VXL→biên dịch trực tiếp sang mã máy và nạp vào ROM,còn RAM dùng để lưu data tạm thời
- Mã chương trình được chứa trong ROM gọi là “phần dẻo”(firmware)
- Hiện nay họ VĐK đa năng tốc độ cao được tích hợp đầy đủ phần cứng như ADC,DAC,cổng I2C,SPI, UART,USB,wifi...được thiết kế trong các kit,module hoàn chỉnh phần cứng như Arduino,Tiva-C...Người thiết kế chủ yếu chỉ thiết kế phần mềm và “nhúng” vào phần cứng tích hợp sẵn→hệ thống nhúng(Embedded Systems)