

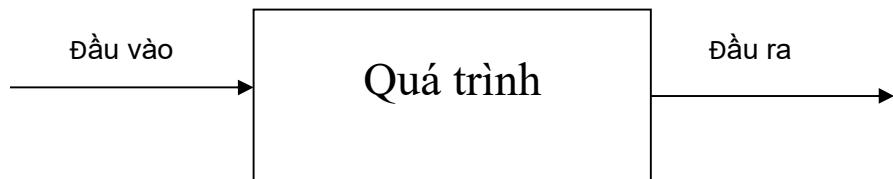
CHƯƠNG 1 GIỚI THIỆU CHUNG

I. Tổng quan về kỹ thuật vi xử lý

Vào năm 1971 hai công ty của Mỹ giới thiệu với thế giới sản phẩm vi xử lý. Các công ty non trẻ này là Intel và đối thủ cạnh tranh với nó là Texas Instruments. Vi xử lý (Microprocesor) và con đẻ của chúng là vi điều khiển (MicroController) đã thâm nhập vào từng quốc gia, từng sản phẩm và từng căn nhà trên thế giới. Hiện nay hầu như mọi người trên hành tinh đều sở hữu hoặc biết đồ vật hay sản phẩm sử dụng thiết bị này. Vậy vi xử lý là gì và nó làm việc như thế nào.

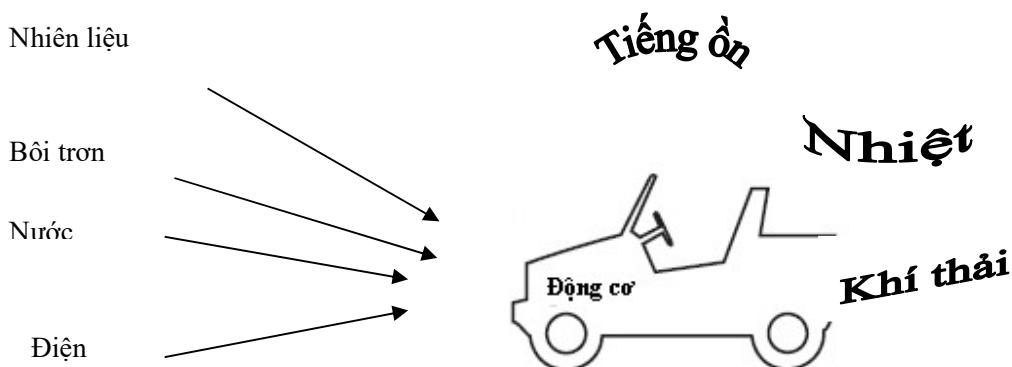
Để trả lời câu hỏi này ta xem xét đặc tính của một hệ thống.

Khái niệm hệ thống được sử dụng để mô tả một tổ chức hay một thiết bị bao gồm có ba phần: đầu vào, đầu ra và quá trình thực hiện một nhiệm vụ nào đó. Thông thường một hệ thống có nhiều đầu vào và nhiều đầu ra, trong một mức độ nào đó quá trình thực hiện nhiệm vụ sẽ thải ra một nhiệt lượng.



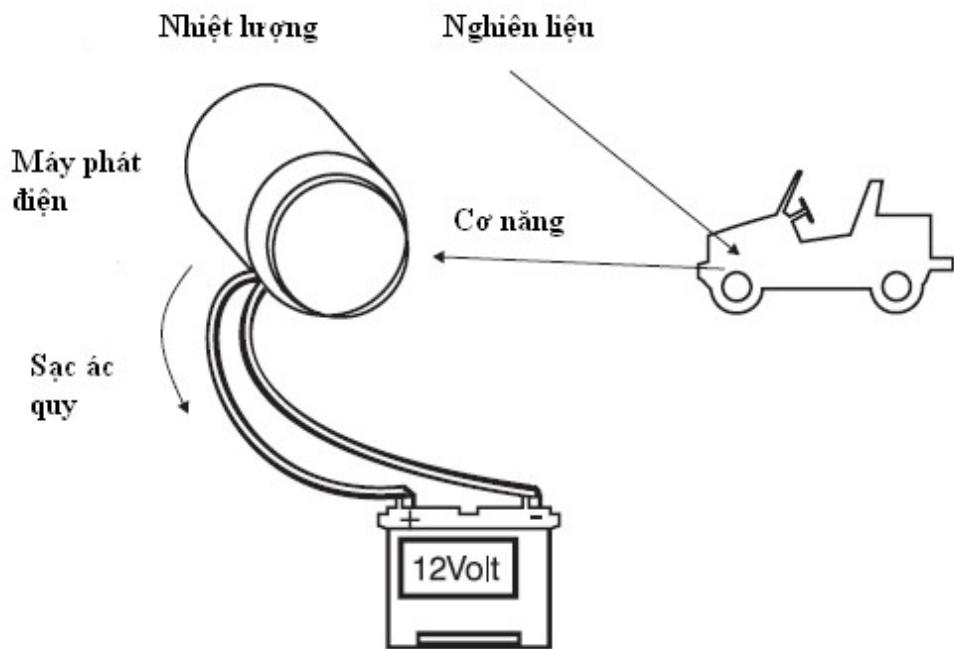
Hình 1.1 Mô hình của hệ thống

Lấy ví dụ một xe ô tô, đầu vào bao gồm nhiên liệu, nước làm mát và ác quy để khởi động xe và cấp điện cho các đèn báo và thiết bị điện. Ở đây hệ thống thực hiện quá trình chuyển hóa nhiên liệu thành năng lượng cơ để ô tô chạy trên đường.



Hình 1.2 Mô hình một xe ô tô

Đầu ra là đoạn đường dịch chuyển mong muốn và các hiệu ứng không mong muốn là khí thải, nhiệt lượng, hơi nước và tiếng ồn.



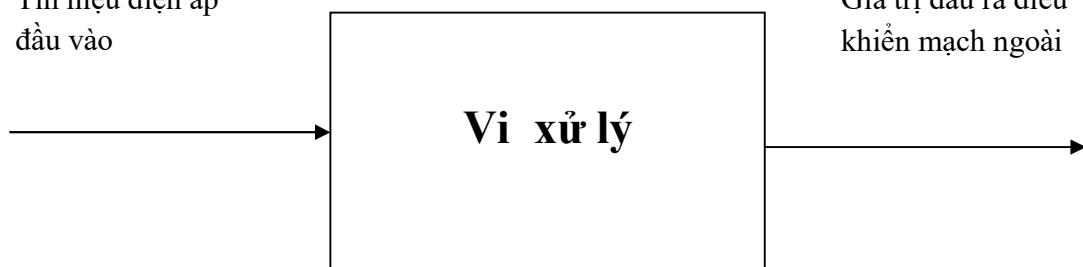
Hình 1.3. Một hệ thống con (nhúng) trên ô tô

Trong ô tô có thể có những hệ thống con bên trong, ví dụ như thiết bị nạp ác quy (hình 3.1), hệ thống con này cũng có đầy đủ đặc tính như một hệ thống (gồm đầu vào, đầu ra và quá trình thực hiện nhiệm vụ). Ta gọi hệ thống này được nhúng vào hệ lớn.

Một hệ thống vi xử lý.

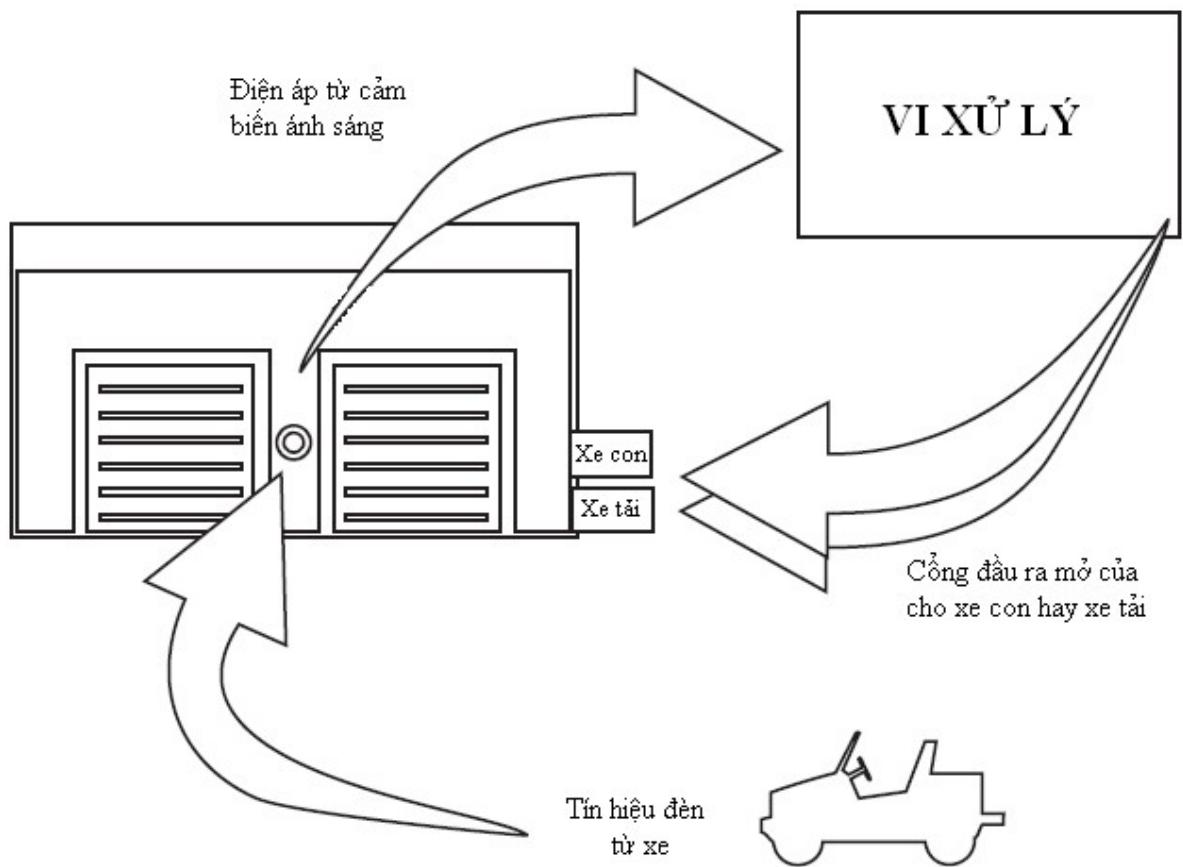
Cũng như một hệ thống, vi xử lý cũng có đầu vào, đầu ra và một quá trình xử lý bên trong. Tín hiệu điện áp đầu vào

Giá trị đầu ra điều khiển mạch ngoài



Hình 1.4 Hệ vi xử lý

Lấy ví dụ hệ thống điều khiển mở cửa tự động vào Gara ô tô, trong Gara có 2 chỗ để xe: một cho xe tải và 1 cho xe con. Trước cửa Gara bố trí đầu đo cảm nhận ánh sáng, mỗi loại xe khi vào Gara sẽ chớp sáng đèn theo kiểu riêng, vì xử lý sẽ tự động nhận dạng kiểu chớp sáng của đèn và tự động mở cửa cho từng loại xe vào.

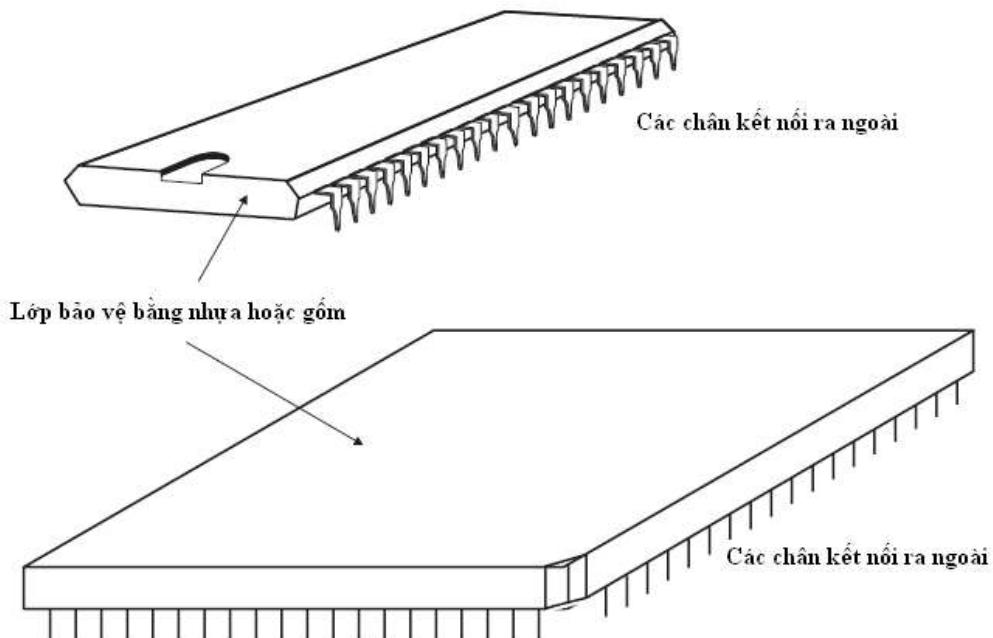


Hình 1.5. Vi xử lý điều khiển mở cửa Gara ô tô

Hình dạng thực thể của một vi xử lý.

Vi xử lý là một mạch điện tử rất bé, thông thường có kích thước theo 1 chiều khoảng 1.2cm đến vài cm. Được bảo vệ bằng tấm vỏ nhựa hoặc gỗ như hình vẽ.

Hình dáng kích thước và số lượng chân kết nối ra phụ thuộc vào nhu cầu sử dụng, số lượng chân vi xử lý có thể là từ 8 chân đến hàng trăm chân.



Hình 1.6 Hình dáng vật lý vi xử lý

II. Lịch sử phát triển của vi xử lý.

Tại Châu Á, vào năm 1969, một nhóm các kỹ sư Nhật Bản từ BUSICOM đã đến Mỹ để đặt hàng một vài vi mạch tích hợp cần thiết cho các máy tính được thiết kế theo dự án của họ. Yêu cầu này được gửi đến INTEL, và Marcian Hoff là người phụ trách của dự án đó. Là người có kinh nghiệm trong lĩnh vực điện tử máy tính, ông đề xuất một ý tưởng mới, thay vì ý tưởng thiết kế được đưa ra trước đó. Giải pháp này là thiết kế các vi mạch hoạt động đúng theo chương trình được lưu trữ trong chính bản thân nó. Điều này có nghĩa là cấu hình vi mạch sẽ đơn giản hơn, nhưng nó sẽ đòi hỏi nhiều không gian bộ nhớ hơn so với giải pháp được đề xuất bởi các kỹ sư Nhật Bản. Sau một thời gian, trong khi các kỹ sư Nhật Bản vẫn cố gắng tìm một giải pháp dễ dàng hơn, thì ý tưởng của Marcian đã được hiện thực và bộ vi xử lý đầu tiên đã ra đời. Một cộng sự lớn của Marcian, góp phần biến ý tưởng trở thành sản phẩm đó là Federico Faggin. Chín tháng sau khi thuê anh ta, Intel đã thành công trong việc phát triển một sản phẩm từ ý tưởng ban đầu đó. Năm 1971, Intel có được quyền bán mạch tích hợp này. Trước đó, Intel đã mua lại bản quyền từ BUSICOM, trở thành kho báu của Intel. Trong năm

đó, một bộ vi xử lý được gọi là 4004 xuất hiện trên thị trường. Đó là bộ vi xử lý 4-bit đầu tiên với tốc độ 108KHz, số lượng bóng bán dẫn là 2.300.

Không lâu sau đó, một công ty Mỹ là CTC yêu cầu Intel và Texas Instruments sản xuất một bộ vi xử lý 8-bit để sử dụng trong các thiết bị đầu cuối. Mặc dù CTC đã từ bỏ dự án này, nhưng Intel và Texas Instruments vẫn tiếp tục nghiên cứu bộ vi xử lý mới và vào tháng tư năm 1972, các bộ vi xử lý 8-bit đầu tiên được gọi là 8008 xuất hiện trên thị trường. Nó sở hữu 16kb bộ nhớ, với tốc độ 200KHz, số lượng bóng bán dẫn là 3.500. Bộ vi xử lý này là tiền thân của tất cả các bộ vi xử lý ngày nay. Intel tiếp tục phát triển nó và trong tháng 4 năm 1974 tiếp tục tung ra một bộ xử lý 8-bit được gọi là 8080. Nó sở hữu 64Kb bộ nhớ, với tốc độ 2MHz, và số lượng bóng bán dẫn là 6.000.

Một công ty khác của Mỹ có tên là Motorola, nhanh chóng nhận ra thị trường tiềm năng này, do đó họ tung ra bộ vi xử lý 8-bit 6800. Phụ trách chính của họ là Chuck Peddle. Ngoài bộ vi xử lý riêng của mình, Motorola là công ty đầu tiên sản xuất thiết bị ngoại vi khác như 6820 và 6850. Vào thời điểm đó, nhiều công ty nhận ra tầm quan trọng của bộ vi xử lý và bắt đầu phát triển nó. Chuck Peddle lúc này rời Motorola để làm việc cho MOS Technology và tiếp tục tập trung phát triển các bộ vi xử lý.

Tại triển lãm WESCON ở Mỹ vào năm 1975, một sự kiện quan trọng trong lịch sử của các bộ vi xử lý đã diễn ra. Tập đoàn MOS Technology công bố rằng vi xử lý 6501 và 6502 đã được bán ở mức \$25 một bộ, mọi khách hàng quan tâm có thể mua ngay lập tức. Thông tin này thậm chí khiến cho người ta nghĩ rằng đây là một kiểu lừa đảo, vì mức giá của các công ty cạnh tranh đã bán 8080 và 6800 là ở mức \$179 mỗi bộ. Vào ngày đầu tiên của triển lãm, để đáp trả lại với đối thủ cạnh tranh, cả Motorola và Intel cắt giảm giá của các bộ vi xử lý của họ đến \$69,95. Motorola cáo buộc MOS Technology và Chuck Peddle vi phạm bản quyền đã được bảo vệ cho bộ vi xử lý 6800. Do đó, MOS Technology ngừng việc sản xuất 6501, nhưng vẫn duy trì sản xuất 6502. Đó là các bộ vi xử lý 8-bit với 56 tập lệnh và khả năng truy cập trực tiếp 64Kb bộ nhớ. Do giá thấp, 6502 đã trở thành rất phổ biến, do đó nó đã được cài đặt vào máy tính như KIM-1, Apple I, Apple II, Atari, Commodore, Acorn, Oric, Galeb, Orao, Ultra và nhiều thiết bị khác. Rất nhanh sau đó, nhiều công ty đã bắt đầu sản xuất 6502 (Rockwell, Sznertek, GTE, NCR, Ricoh, Commodore, vượt qua cả MOS Technology). Trong năm thịnh vượng 1982, bộ vi xử lý này đã được bán với tốc độ khủng khiếp, 15 triệu bộ xử lý một năm!

Các công ty khác không muốn từ bỏ thị trường này. Frederico Faggin đã rời Intel và bắt đầu tại công ty Zilog Inc. Năm 1976 Zilog công bố Z80. Khi thiết kế bộ vi xử lý này Faggin đã có một quyết định rất quan trọng. Đó là nhận thấy 8080 đã rất phát triển và ông nhận ra rằng nhiều người sẽ vẫn trung thành với bộ xử lý đó vì chi phí cho việc thay thế, viết lại toàn bộ các chương trình sẽ rất lớn. Và ông quyết định rằng một bộ xử lý mới sẽ có thể tương thích với 8080, tức là nó có thể có thể thực hiện được tất cả các chương trình đã viết cho 8080. Bên cạnh đó, nhiều tính năng khác đã được thêm vào, để cho Z80 là bộ vi xử lý mạnh mẽ nhất tại thời điểm đó. Nó đã có thể truy cập trực tiếp đến 64Kb địa chỉ bộ nhớ, đã có 176 tập lệnh, một số lượng lớn thanh ghi, bộ nhớ RAM mới, một nguồn cung cấp điện duy nhất, tốc độ hoạt động lớn hơn ...v.v. Các bộ vi xử lý Z80 là một thành công lớn và tất cả mọi người đã thay thế 8080 thành Z80. Chắc chắn là Z80 là bộ vi xử lý 8-bit thương mại thành công nhất tại thời điểm đó. Bên cạnh Zilog, các nhà sản xuất mới khác như Mostek, NEC, SHARP và SGS xuất hiện sớm. Z80 là trung tâm của nhiều máy tính như: Spectrum, Partner, TRS703, Z-3 và Galaxy.

Năm 1976, Intel đã đưa ra một phiên bản nâng cấp của bộ vi xử lý 8-bit được gọi là 8085. Tuy nhiên, Z80 vẫn tốt hơn rất nhiều và Intel thất bại với bộ vi xử lý mới. Mặc dù một vài bộ vi xử lý khác tiếp tục xuất hiện trên thị trường (6809, 2650, SC / MP), nhưng Z80 đã chấm dứt mọi cơ hội. Không có những cải tiến lớn, các bộ vi xử lý khác đã không thể làm thay đổi lựa chọn của các nhà sản xuất, do đó cuối cùng 6502 và Z80 cùng với 6800 vẫn là đại diện chính của các bộ vi xử lý 8-bit thời điểm đó. Năm 1978 Một hợp đồng cung cấp sản phẩm quan trọng cho bộ phận máy tính cá nhân mới thành lập của IBM đã biến bộ vi xử lý 8088 trở thành bộ não của sản phẩm chủ đạo mới của IBM—máy tính IBM PC, số lượng bóng bán dẫn 29.000 với tốc độ xung nhịp 10Mhz. Từ năm 1982 đến 1989 Intel lần lượt cho ra đời vi xử lý 286 đến 486 DX với số lượng bóng bán dẫn lên đến 1.2 triệu với tốc độ 50Mhz.

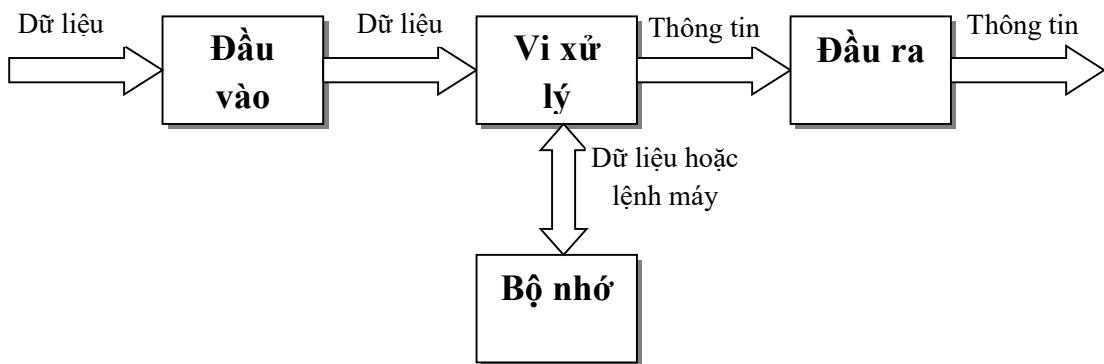
Năm 1993 xuất hiện bộ vi xử lý Pentium® cho phép các máy tính dễ dàng hơn trong việc tích hợp những dữ liệu ‘thế giới thực’ như giọng nói, âm thanh, ký tự viết tay và các ảnh đồ họa. Từ năm 1997 đến 2000 Intel giới thiệu vi xử lý Pentium II đến Pentium 4 với số lượng bóng bán dẫn đạt 42 triệu và tốc độ 2GHz, đến năm 2002 sử dụng Công nghệ Siêu phân luồng.

Từ năm 2005 xuất hiện vi xử lý đa nhân Core 2 Duo và Core Extreme có 291 triệu bóng bán dẫn. Năm 2011 Intel giới thiệu Chip Core i3,i5,i7 có tối đa 8 nhân với 995 triệu bóng bán dẫn.

Đồng thời với các Chip sử dụng cho máy tính và máy chủ hãng ARM giới thiệu dòng chip ARM dùng cho các thiết bị di động có đa nhân với tần số xung nhịp đến 1.5GHz

III. Cấu trúc hệ vi xử lý.

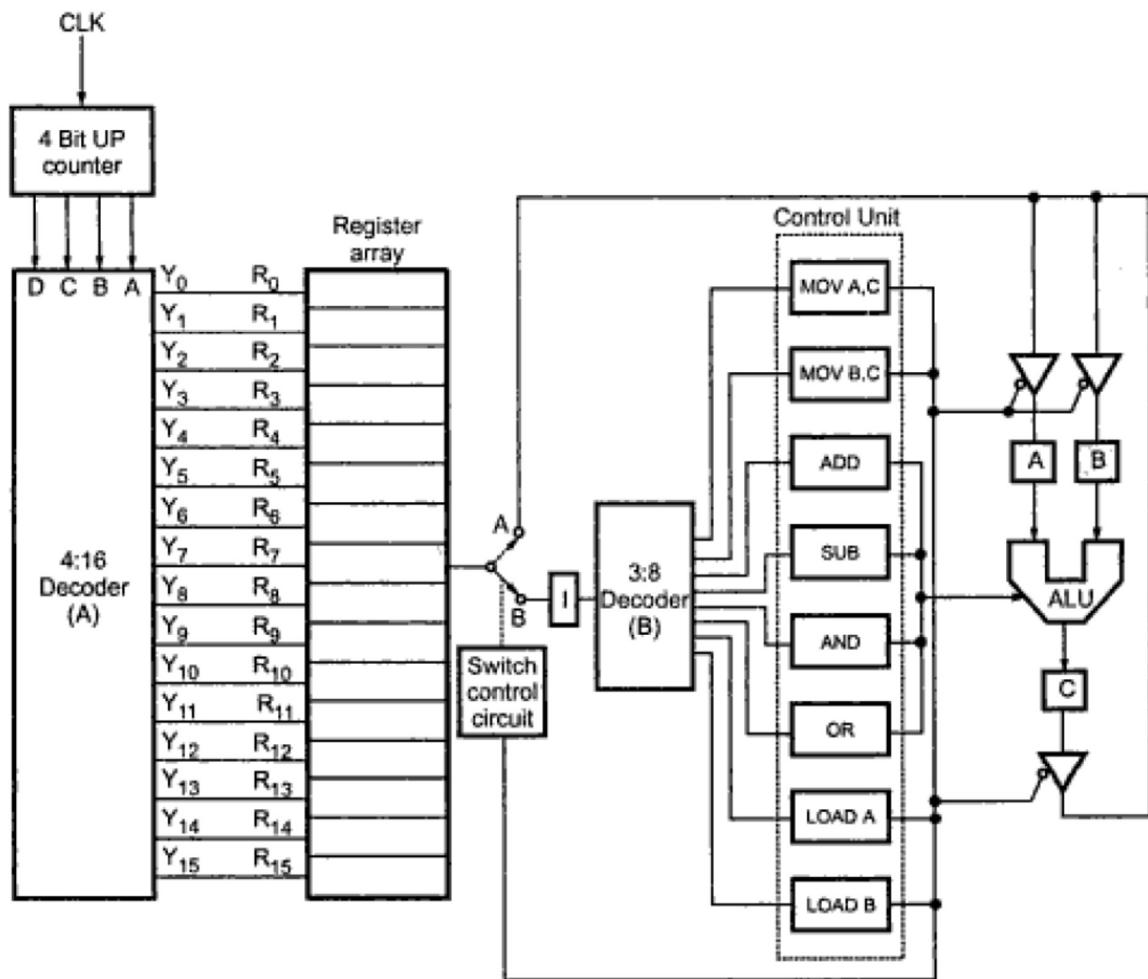
Một hệ thống gồm bộ xử lý trung tâm, đầu vào, đầu ra và bộ nhớ được thể hiện trong hình dưới. Bộ xử lý là một vi mạch tích hợp có kích thước rất bé, chính vì vậy được gọi là vi xử lý. Một hệ thống sử dụng và xử lý được gọi là hệ thống nền vi xử lý.



Hình 1.7 Mô hình hệ vi xử lý đơn giản nhất

Để xem xét sự hoạt động của các khối chức năng cơ bản nhất của vi xử lý, ta sử dụng một mô hình đơn giản nhất của một vi xử lý.

Mô hình đơn giản là một mạch số tuần tự ở hình dưới. nó gồm bộ đếm tăng 4 bit, bộ giải mã (decoder), thanh ghi (register), khâu chuyển mạch, khối điều khiển và khối logic số học (ALU).



Hình 1.8 Thực hiện câu lệnh bằng mạch tuần tự số

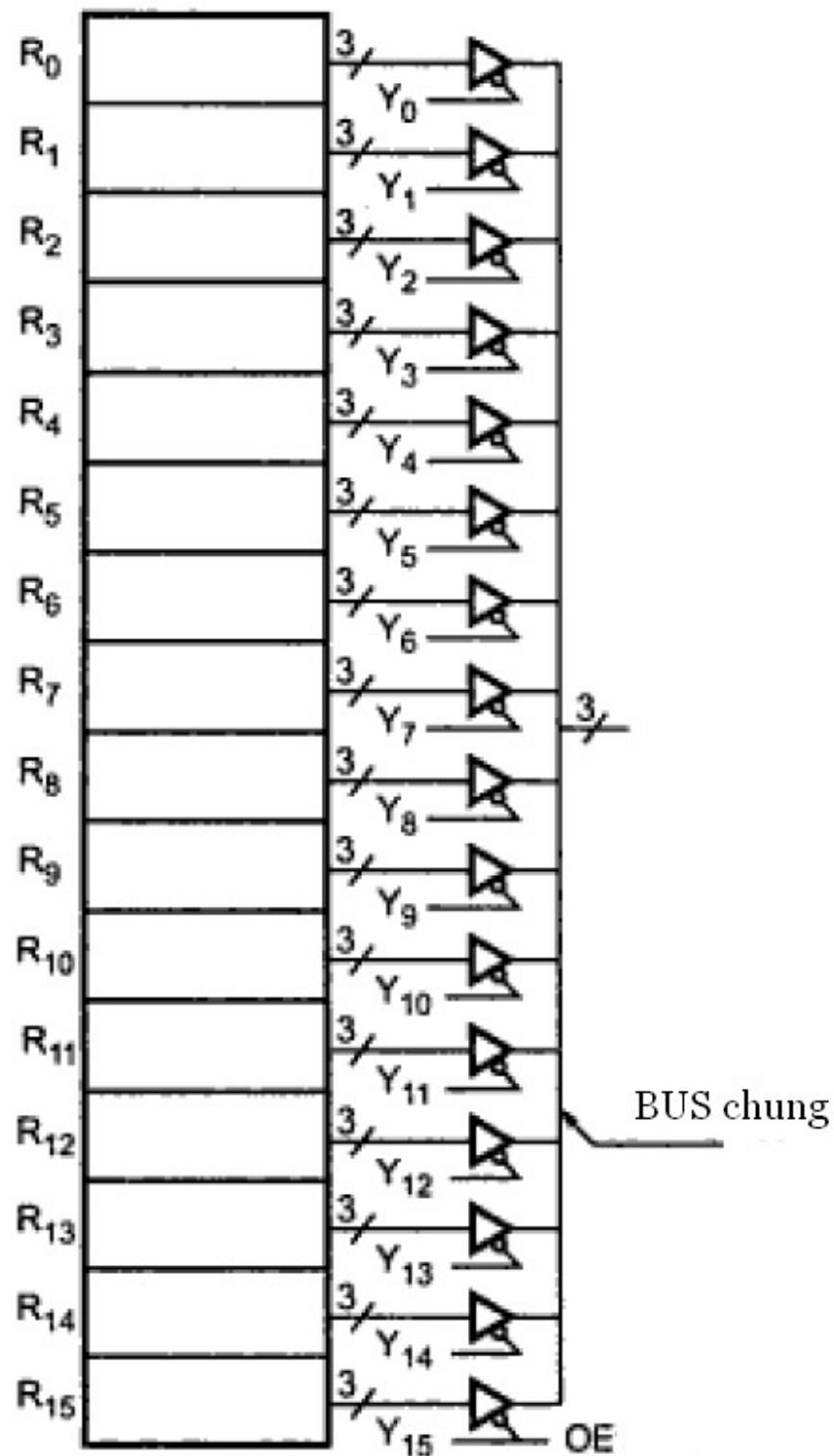
1.3.1 Bộ đếm

Đây là bộ đếm tăng 4 bit, giá trị đếm từ 0 đến 15 và luôn xuất phát từ 0. Bộ đếm được điều khiển bởi tín hiệu xung nhịp (clock) và đầu ra của bộ đếm đưa vào bộ giải mã A.

1.3.2 Bộ giải mã A

Đây là bộ giải mã 4:16. Bốn bit đầu ra của bộ đếm được sử dụng như là đầu vào của giải mã A. Trạng thái đỉnh cực của đầu ra phụ thuộc trạng thái đầu vào. Đầu ra của bộ giải mã A được sử dụng để lựa chọn một trong số các thanh ghi nằm trong mảng thanh ghi (register array).

1.3.3 Mảng thanh ghi



Hình 1.9 Mảng thanh ghi

Mảng thanh ghi bao gồm 16 thanh ghi 3 bit. Mỗi thanh ghi có thể chứa một dữ liệu 3 bit. Các đầu ra của tất cả các thanh ghi được nối vào đường Bus chung nhờ sự điều khiển của tín hiệu cho phép ra (output enable-OE) như hình dưới.

Đầu ra của bộ giải mã A được sử dụng để tính cực tín hiệu OE cho 16 thanh ghi (R0-R15). Tại một thời điểm chỉ có 1 đầu ra của bộ giải mã là tích cực, như vậy chỉ có dữ liệu từ 1 thanh ghi được đưa ra Bus chung.

1.3.4 Bus chung

Ba bit của Bus chung được nối vào thanh ghi A, thanh ghi B và đèn thanh ghi I thông qua chuyển mạch.

Nếu chuyển mạch ở vị trí A thì Bus chung được nối với thanh ghi A hoặc B, nếu không thì nó được nối với thanh ghi I. vị trí của chuyển mạch được điều khiển bởi mạch điều khiển chuyển mạch.

1.3.5 Thanh ghi B

Nó được sử dụng để lưu dữ liệu từ Bus chung.

1.3.6 Bộ giải mã B

Đây là bộ giải mã 3:8. Dữ liệu từ thanh ghi I được sử dụng như đầu vào của bộ giải mã. Nếu chuyển mạch ở trạng thái B, dữ liệu này sẽ được giải mã và được sử dụng để lựa chọn mạch điều khiển phù hợp trong khối điều khiển như bảng dưới.

Bảng 1.1. Bộ giải mã B.

D ₂	D ₁	D ₀	Mạch điều khiển được chọn
0	0	0	MOV A,C
0	0	1	MOV B,C
0	1	0	ADD
0	1	1	SUB
1	0	0	AND
1	0	1	OR
1	1	0	LOAD A
1	1	1	LOAD B

1.3.7 Khối điều khiển

Bao gồm 8 mạch điều khiển MOV A,C; MOV B,C; AOD; SUIS; AUD; DA; LOAD A và LOAD B. các mạch điều khiển này được sử dụng để tạo ra tín hiệu tác động vào khối logic số học ALU và tích cực đầu vào cho phép hoạt động của thanh ghi A và B. tại một thời điểm chỉ có 1 tín hiệu ra từ 1 mạch điều khiển được tích cực.....

MOV A,C. mạch điều khiển này tạo ra tín hiệu điều khiển cho phép tích cực cống vào của thanh ghi B và cống ra của thanh ghi C, như vậy dữ liệu từ thanh ghi C được copy vào thanh ghi B.

ADD. Mạch điều khiển này tạo ra tín hiệu điều khiển yêu cầu thực hiện lệnh cộng trong khối ALU với phép toán $A + B \rightarrow C$

SUB. Mạch điều khiển này tạo ra tín hiệu điều khiển yêu cầu thực hiện trừ trong khối ALU với phép toán $A - B \rightarrow C$

AND

Mạch điều khiển này được tạo ra tín hiệu điều khiển yêu cầu thực hiện lệnh logic AND trong khối ALU.

OR

Tương tự như trên nhưng thực hiện lệnh OR

LOAD A

Mạch điều khiển tạo tín hiệu điều khiển mở cổng đầu vào cho thanh ghi A, lúc này chuyển mạch chuyển về vị trí A.

LOAD B

Mạch điều khiển tạo tín hiệu điều khiển mở cổng đầu vào cho thanh ghi B, lúc này chuyển mạch chuyển về vị trí A.

1.3.8 Chuyển mạch điều khiển

Có nhiệm vụ điều khiển chuyển mạch ở vị trí A hay B.

1.3.9 ALU (arithumetric logic unit)

Khối logic số học lấy dữ liệu từ thanh ghi A và B. trong khối ALU sẽ thực hiện phép toán được lựa chọn bởi khối điều khiển, kết quả của quá trình này được đưa ra thanh ghi C.

IV. Quá trình hoạt động thực hiện xử lý

Ban đầu bộ đếm xuất phát từ giá trị 0. Lúc này bộ giải mã lựa chọn thanh ghi đầu tiên (Ro) từ mảng thanh ghi và dữ liệu trong thanh ghi lựa chọn (Ro) đưa ra Bus chung. Ban đầu chuyển mạch ở vị trí B, như vậy dữ liệu từ Bus chung được giải mã. Căn cứ vào dữ liệu, khối điều khiển lựa chọn lệnh thực hiện và khối ALU thực hiện lệnh đó.

Điểm quan trọng trong quá trình này là dữ liệu từ trong thanh ghi lựa chọn sẽ quyết định lệnh nào thực hiện. Dữ liệu quyết định lệnh thực hiện được gọi “mã lệnh”- operation code hoặc opcode. Mỗi một lệnh có 1 opcode riêng. Trong bảng dưới đây trình bày các opcode cho các lệnh khác nhau.

Lệnh tác động	Mã lệnh		
	D ₂	D ₁	D ₀
MOV A,C	0	0	0

MOV B,C	0	0	1
ADD	0	1	0
SUB	0	1	1
AND	1	0	0
OR	1	0	1
LOAD A	1	1	0
LOAD B	1	1	1

Bảng 1.2 Ví dụ về mã lệnh

Với các lệnh MOV A,C và MOV B,C: dữ liệu sẽ được chuyển từ thanh ghi C đến thanh ghi A và B tương ứng.

Trong lệnh ADD, SUD, AND và OR, dữ liệu sẽ được lấy từ thanh ghi A và B. nhưng với lệnh LOAD A và LOAD B dữ liệu từ Bus chung sẽ được nạp trực tiếp vào thanh ghi A và B tương ứng. với lệnh này chuyển mạch cần nằm ở vị trí A.

Sau mỗi lần thực hiện lệnh bộ đếm tăng lên 1 giá trị, do đó bộ giải mã lựa chọn thanh ghi tiếp theo. Tương tự như trên dữ liệu từ thanh ghi lựa chọn được đưa ra Bus chung và quá trình lặp lại. ví dụ sau khi thể hiện một quá trình thực hiện lệnh cùng 2 số:

Bước 1: Nạp số đầu tiên vào thanh ghi A

Bước 2: Nạp số đầu tiên vào thanh ghi B

Bước 3: cộng 2 số

Bước đầu tiên là nạp số thứ nhất vào thanh ghi A. để làm được điều này cần có mã lệnh (opcode) LOAD A trong thanh ghi số 0 (R_0) và số cần nạp trong thanh ghi số 1 (R_1).

Tương tự ở bước 2 cần có mã lệnh LOADB trong thanh ghi số 2 (R_2) và số thứ 2 trong thanh ghi số 3 (R_3). Cuối cùng, cần có mã lệnh AND trong thanh ghi số 4 (R_4).

Hình dưới thể hiện giá trị của các thanh ghi với số thứ nhất cần cộng là 0b100 và số thứ hai là 0b010.

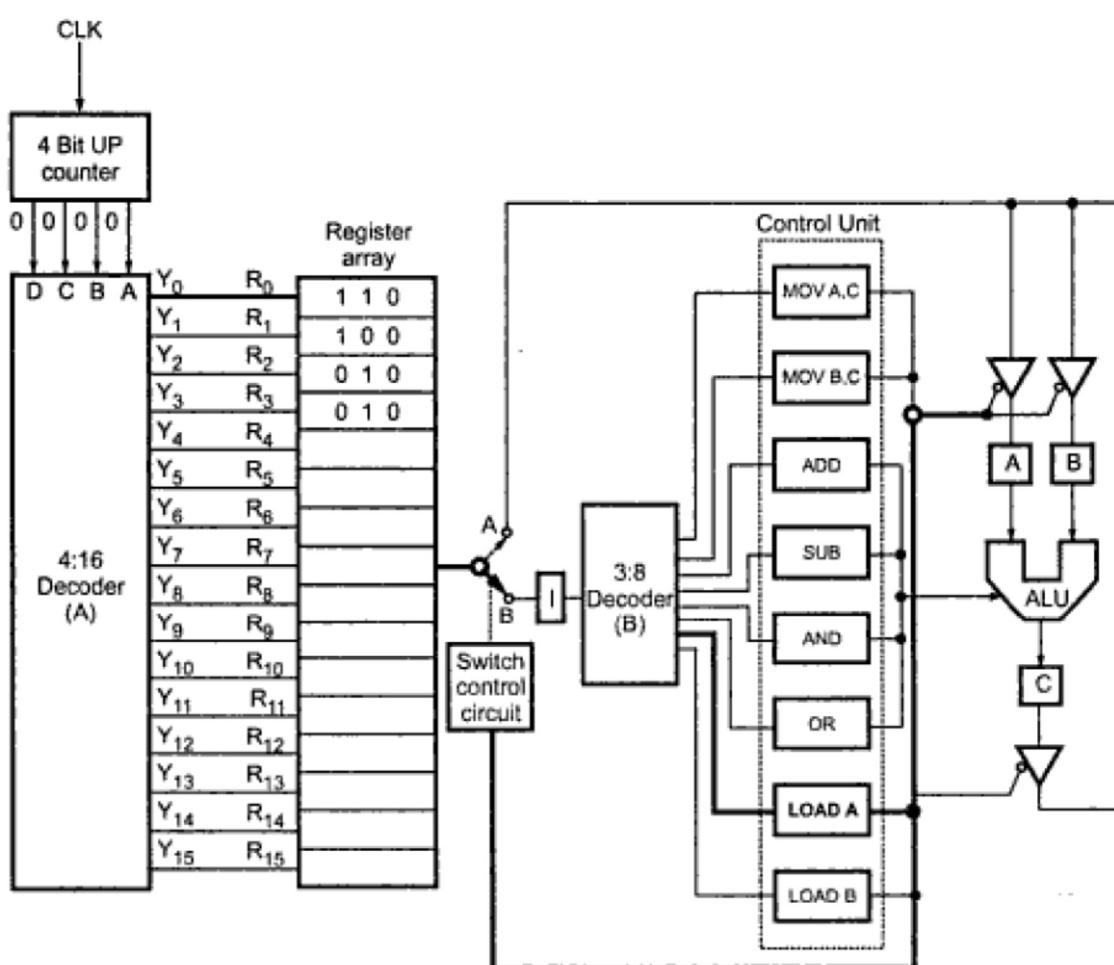
Quá trình thực hiện:

R_0	1	1	0
R_1	1	0	0
R_2	1	1	1
R_3	0	1	0
R_4	0	1	0
R_5	—		
R_6	—		

Bước 1: giá trị đầu ra của bộ đếm 0000

Khi khởi tạo, bộ đếm được xóa (reset) về giá trị 0. Bộ giải mã A sẽ lựa chọn thanh ghi 0 (R_0) do đầu vào của nó có giá trị 0. Khi thanh ghi R_0 được lựa chọn, dữ liệu từ thanh ghi 0, tức là giá trị 110 được đưa ra BUS chung. Bộ giải mã B sẽ sử dụng giá trị này để lựa chọn lệnh thực hiện. Với dữ liệu là 110 bộ giải mã B sẽ lựa chọn lệnh LOAD A. Lệnh LOAD A sẽ mở cổng đầu vào của thanh ghi A và chuyển công tắc từ vị trí B sang vị trí A. (xem hình dưới)

Hình 1.10

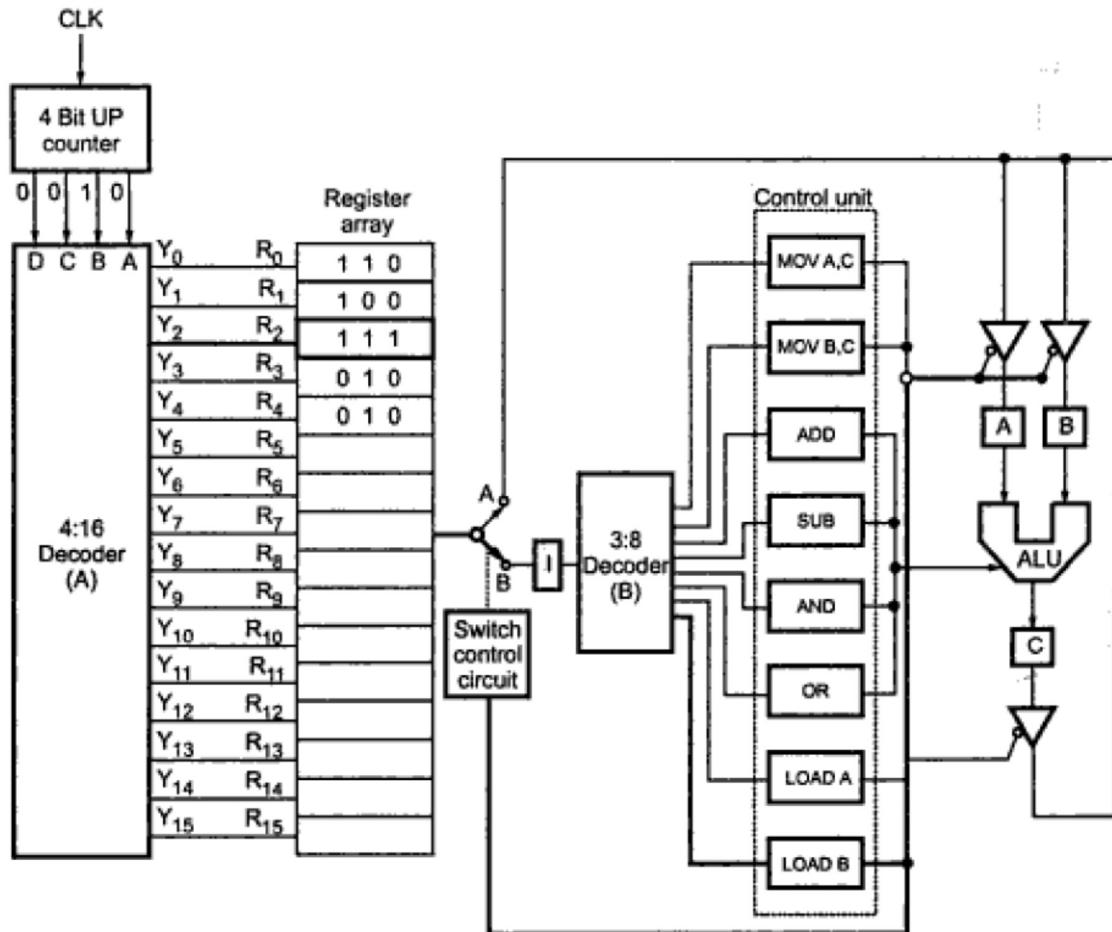


Hình 1.11 Thực hiện bước 1

Bước 2: giá trị đầu ra của bộ đếm là 0001

Bộ giải mã A sẽ lựa chọn thanh ghi 1 (R_1). Khi thanh ghi R_1 được lựa chọn, dữ liệu trong thanh ghi 1, tức là giá trị 100 được đưa vào BUS chung. Dữ liệu này được truyền trực tiếp vào thanh ghi A do khóa chuyển mạch nằm ở vị trí A và đầu vào của thanh ghi A đã được mở.

Sau khi chuyển dữ liệu, công tắc chuyển mạch chuyển từ vị trí A sang B.

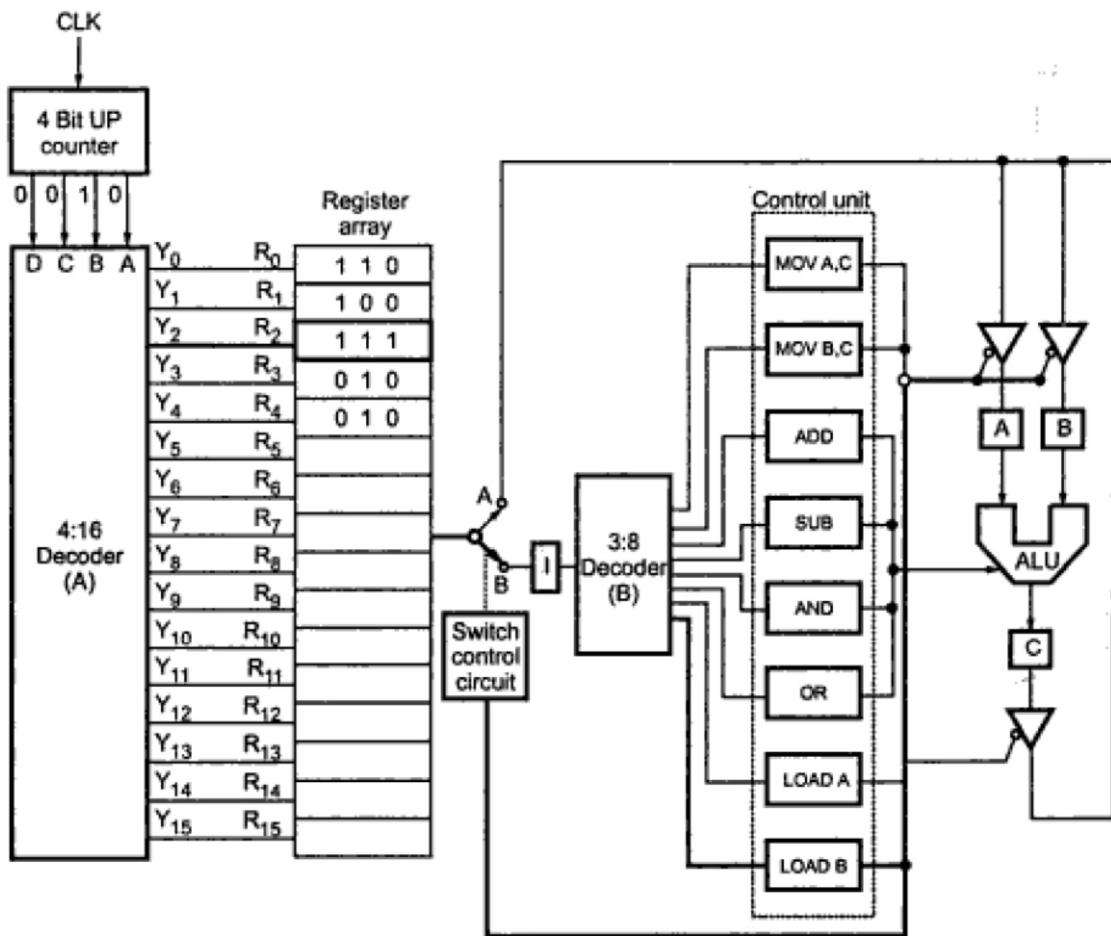


Hình 1.12 Thực hiện bước 2.

Bước 3: giá trị đầu ra bộ đếm là 0010

Bộ giải mã A sẽ lựa chọn thanh ghi 2 (R_2). Từ thanh ghi 2 (R_2) được lựa chọn, dữ liệu từ thanh ghi 2 tức là giá trị 111 được đưa vào BUS chung. Bộ giải mã B sẽ sử dụng giá trị này để lựa chọn lệnh thực hiện.

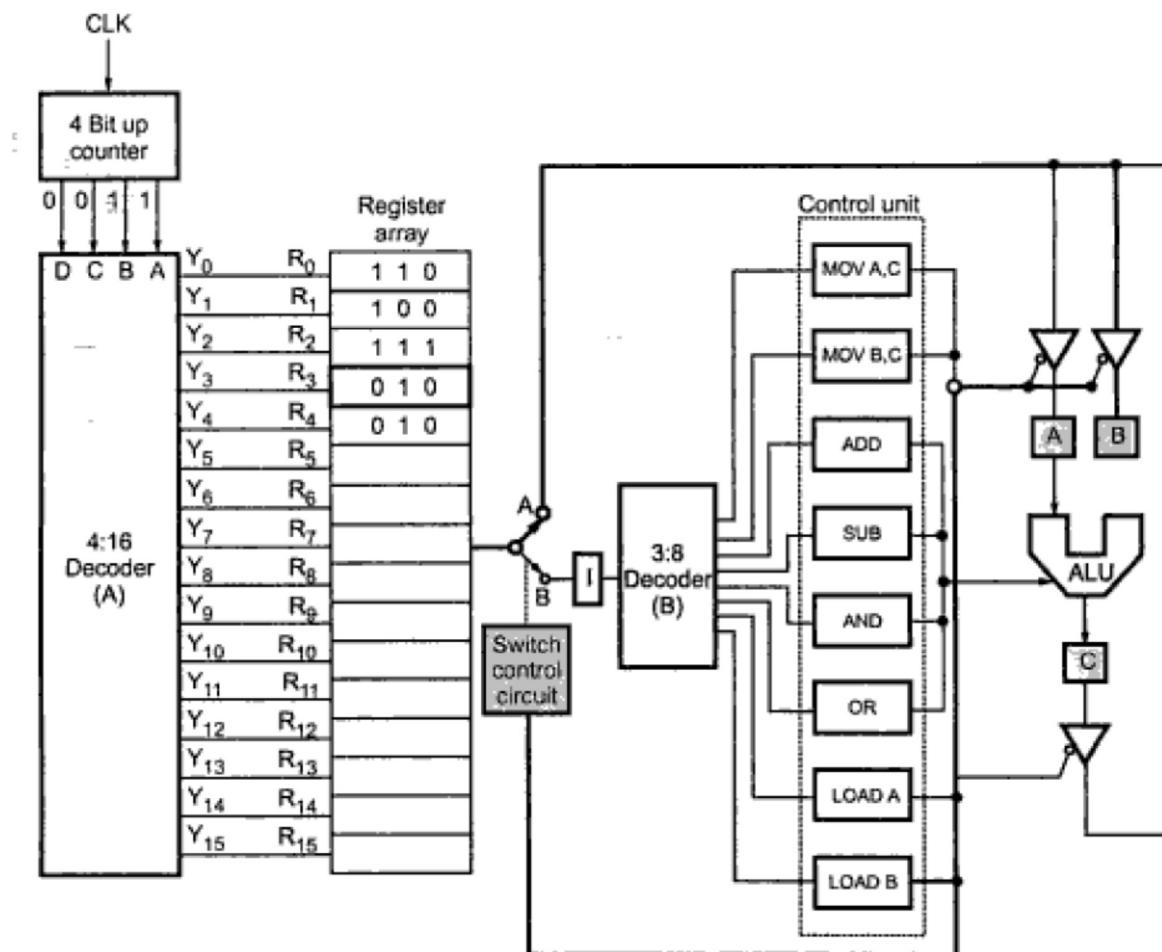
Với giá trị 111, bộ giải mã B sẽ lựa chọn lệnh LOAD B. Lệnh LOAD B sẽ mở cổng vào cho thanh ghi B và chuyển công tắc từ vị trí B về A.



Hình 1.13 Thực hiện bước 3

Bước 4: giá trị đầu ra bộ đếm là 0011

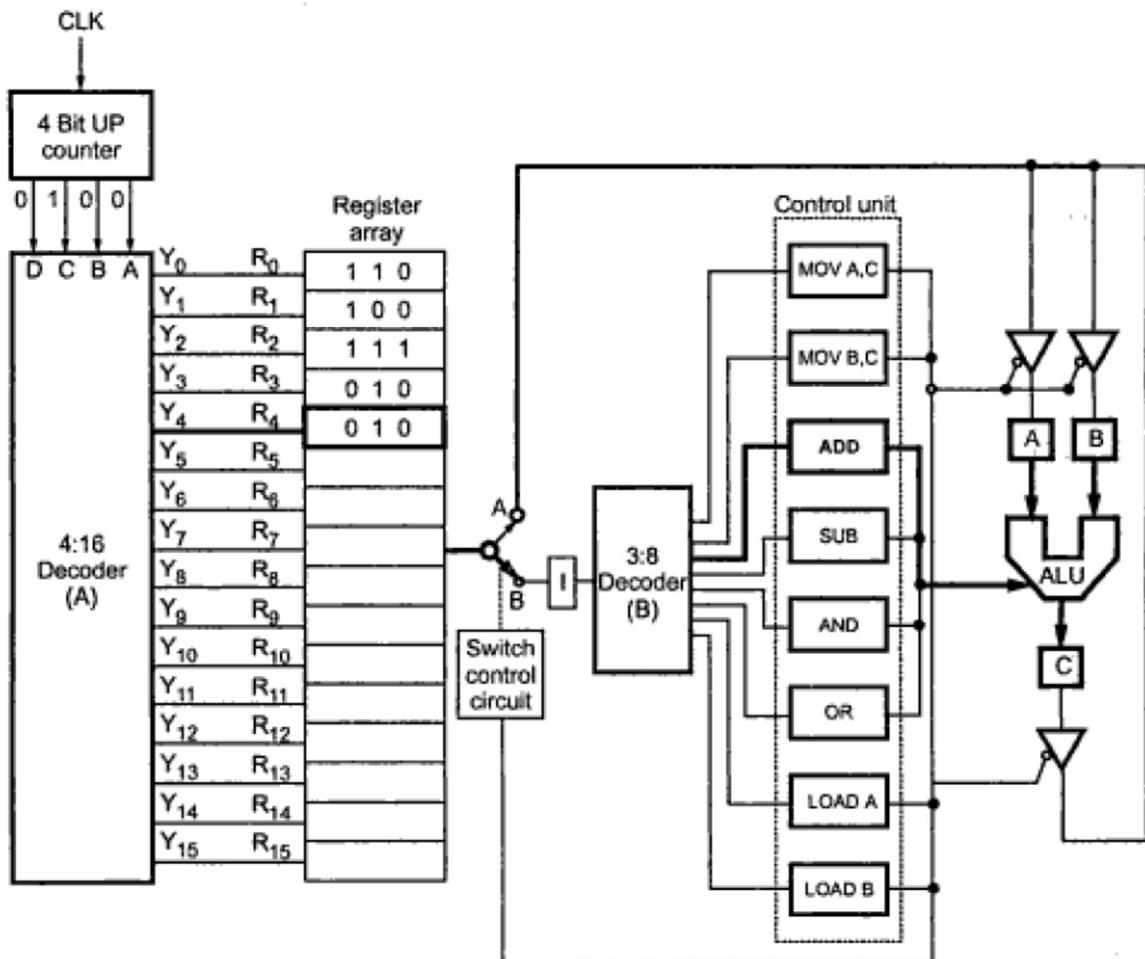
Bộ giải mã A sẽ lựa chọn thanh ghi 3 (R_3). Khi thanh ghi 3 được lựa chọn, dữ liệu từ thanh ghi 3, tức là giá trị 010 được đưa ra BUS chung. Giá trị này được truyền trực tiếp vào thanh ghi B, bởi vì công tắc lựa chọn đang ở vị trí A và công tắc đầu vào của thanh ghi B đã mở. sau khi chuyển dữ liệu, công tắc chuyển mạch chuyển từ vị trí A sang B.



Hình 1.14 Thực hiện bước 4

Bước 5: giá trị đầu ra bộ đếm là 0100

Bộ giải mã A sẽ lựa chọn thanh ghi 4 (R₄). Khi thanh ghi 4 được lựa chọn dữ liệu từ thanh ghi 4, tức là giá trị 010 được đưa ra BUS chung. Với giá trị 010, bộ giải mã B sẽ lựa chọn lệnh AOD. Lệnh AOD sẽ tạo tín hiệu để lựa chọn thao tác thực hiện lệnh cộng bên trong khối ALU. Khi đó, bộ ALU thực hiện cộng trong thanh ghi A với giá trị trong thanh ghi B và kết quả gửi tới đầu ra thanh ghi C, lúc này có giá trị là 110.



Hình 1.15 Thực hiện bước 5

Với ví dụ trên cho phép xem quá trình cộng 2 số thực hiện bằng 1 mạch tuần tự. sử dụng 1 mạch tuần tự ta có thể thực hiện các phép tính trừ, nhân, chia, phép logic AND, OR. Việc này có thể thực hiện bằng cách đặt vào mảng thanh ghi R_0, R_{15} các opcode và giá trị phù hợp.

Một điểm quan trọng khác là cần lưu ý là lệnh LOAD A và LOAD B cần 2 bước. Ở bước 1, đầu vào các thanh ghi nạp dữ liệu được mở và vị trí chuyển mạch thay đổi từ B sang A. Ở bước 2, giá trị cần nạp được chuyển vào thanh ghi tương ứng.

Giá trị này không phải mã lệnh (opcode) mà được gọi là số hạng một lệnh hàm chính bao gồm mã lệnh và số hạng (operand). Một lệnh hàm chính được gọi là câu lệnh (instruction).

Các ví dụ dưới đây thể hiện việc thực hiện các câu lệnh:

- a) Ví dụ 1: Lệnh trừ (SUB): lấy 7 trừ 2

Mảng thanh ghi			Tác động
1	1	0	Nạp thanh ghi A
1	1	1	Với giá trị 7 (111)
1	1	1	Nạp thanh ghi B
0	1	0	Với giá trị 2 (010)
0	1	1	Giá trị trong thanh ghi A trừ giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C

b) Ví dụ 2: Lệnh AND (3 and 5)

Mảng thanh ghi			Tác động
1	1	0	Nạp thanh ghi A
0	1	1	Với giá trị 3 (011)
1	1	1	Nạp thanh ghi B
1	0	1	Với giá trị 5 (101)
1	0	0	Giá trị trong thanh ghi A AND giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C

c) Ví dụ 3: Thực hiện phép toán: 5+1-3.

Mảng thanh ghi			Tác động
1	1	0	Nạp thanh ghi A
1	0	1	Với giá trị 5 (101)
1	1	1	Nạp thanh ghi B
0	0	1	Với giá trị 1 (001)
0	1	0	Giá trị trong thanh ghi A cộng giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C
0	0	0	Copy nội dung thanh ghi C vào thanh ghi A
1	1	1	Nạp vào thanh ghi B
0	1	1	Với giá trị 3 (011)
0	1	1	Giá trị trong thanh ghi A trừ giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C

d) Ví dụ 4: Thực hiện phép toán: 4 AND 3 OR 1

Mảng thanh ghi			Tác động
1	1	0	Nạp thanh ghi A

1	0	0	Với giá trị 4 (100)
1	1	1	Nạp thanh ghi B
0	1	1	Với giá trị 1 (011)
1	0	0	Giá trị trong thanh ghi A AND giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C
0	0	0	Copy nội dung thanh ghi C vào thanh ghi A
1	1	1	Nạp vào thanh ghi B
0	1	1	Với giá trị 3 (011)
1	0	1	Giá trị trong thanh ghi A OR giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C

Từ các ví dụ và mô tả trên, ta có thể nhận xét rằng thành phần cứng là cố định, nhưng khi thay đổi thứ tự các mã lệnh và số hạng (câu lệnh) cho ta các quá trình thực hiện khác nhau. Điều khiển quá trình thực hiện lệnh với một phần cứng không chỉ bằng cách thay đổi trình tự thực hiện câu lệnh là 1 nguyên tắc cơ bản khi sử dụng vi xử lý. Mô hình mô tả ở trên trên là 1 vi xử lý đơn giản nhất.

CHƯƠNG 2 CÂU TRÚC VÀ SỰ HOẠT ĐỘNG CỦA VI XỬ LÝ

I. Các thuật ngữ sử dụng trong vi xử lý

Trong mô hình đơn giản của bộ vi xử lý gồm có 2 khối: khối 1 vi xử lý và khối 2 là bộ nhớ.

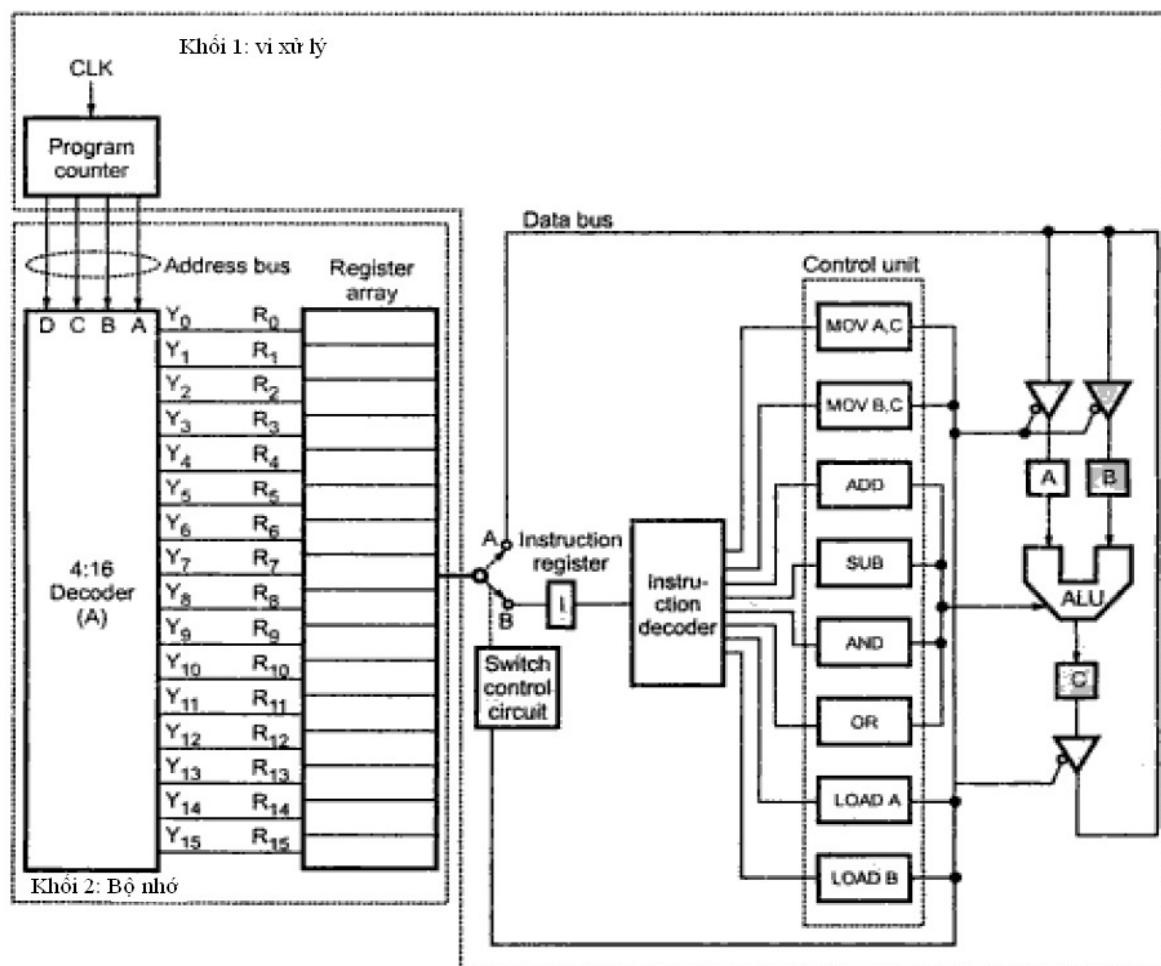
Vi xử lý: nó bao gồm bộ đếm, khúc chuyển mạch, bộ giải mã B, khôi điều khiển, ALU và các thanh ghi.

Bộ nhớ: nó bao gồm mảng thanh ghi và mạch giải mã A

Chương trình: trình tự thực hiện các câu lệnh được viết theo cú pháp xác định được gọi là chương trình.

Bộ đếm chương trình: bộ đếm sử dụng trong mô hình đơn giản được dùng để định vị câu lệnh trong 1 chuỗi lệnh (chương trình). Bộ đếm này được gọi bộ đếm chương trình.

Địa chỉ và BUS địa chỉ:



Hình 2.1 Địa chỉ và BUS địa chỉ

Dữ liệu và Bus dữ liệu:

Dữ liệu chứa trong thanh ghi có thể ở các dạng khác nhau như mã lệnh và số hạng. Dữ liệu này được chuyển từ thanh ghi đến thanh ghi , từ bộ nhớ đến thanh ghi thông qua 1 nhóm các dãy được gọi là bus dữ liệu. Bus dữ liệu kết nối với vi xử lý được gọi là bus dữ liệu bên trong (internal), trong khi bus dữ liệu bên ngoài được gọi là bus dữ liệu bên ngoài (external). Bus dữ liệu hình trên là Bus 3 bit. Kích thước của bus dữ liệu quyết định kích thước của số hạng. Nó cũng quyết định số lượng lớn nhất của các lệnh mà vi xử lý có thể thực hiện. Với bus dữ liệu của mô hình trên có 3 bit, như vậy có thể có $2^3 = 8$ lệnh khác nhau. Kích thước tiêu chuẩn của bus dữ liệu là bội số của 8, ví dụ 8, 16, 24, 32, 64... một vi xử lý với 8 bit dữ liệu có thể có tới 256(28) câu lệnh khác nhau.

Độ dài của từ: một nhóm các bit mà vi xử lý có thể quản lý và điều khiển được gọi là từ và vi xử lý được phân loại theo chiều dài của từ. Ví dụ vi xử lý có 8 bit từ gọi là vi xử lý 8 bit và vi xử lý 16 bit từ gọi là vi xử lý 16 bit.

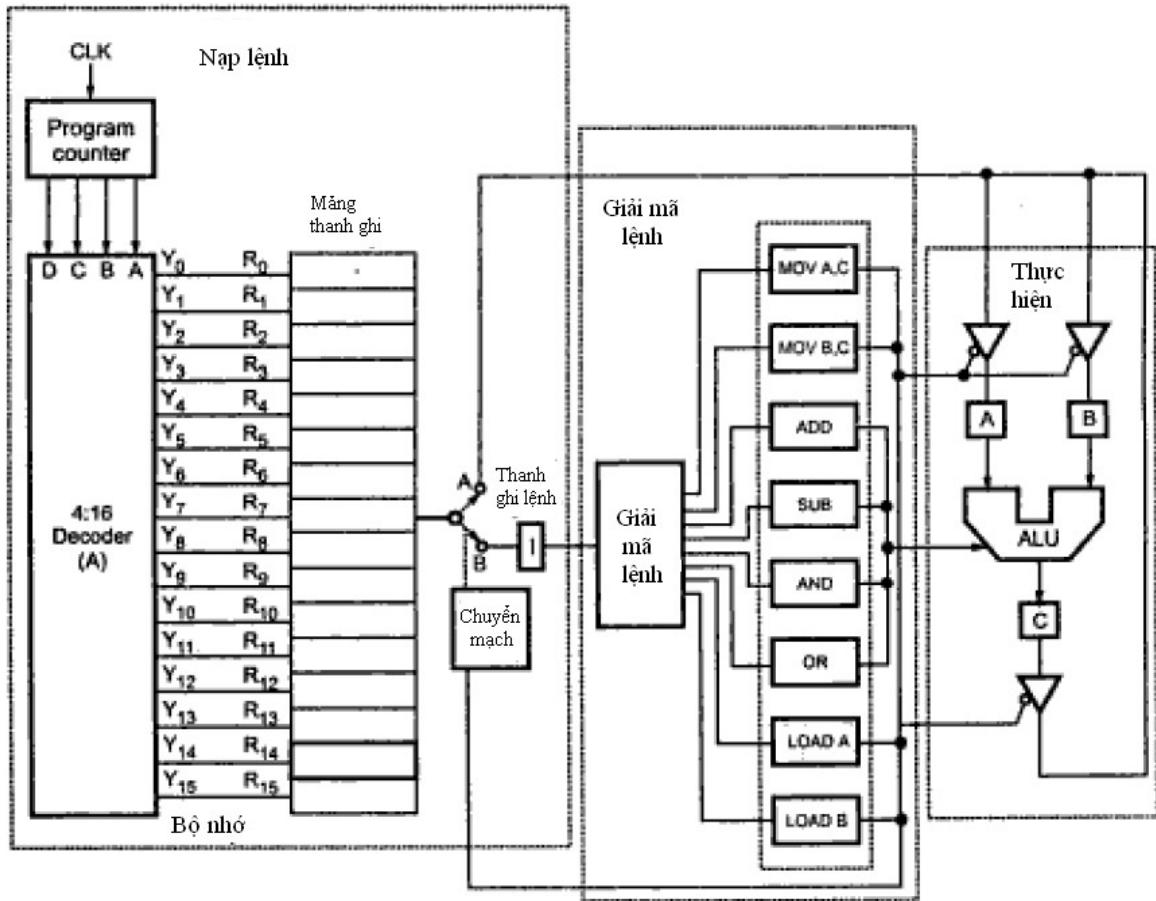
Thanh ghi câu lệnh: thanh ghi chứa các mã lệnh được lưu trữ tạm thời được gọi là thanh ghi câu lệnh (instruction register).

Giải mã câu lệnh: bộ giải mã (bộ giải mã và trong mô hình đơn giản) sẽ lấy mã lệnh từ thanh ghi câu lệnh và giải mã nó chế tạo ra tín hiệu điều khiển phù hợp được gọi là bộ giải mã bộ giải mã.

II. Các pha trong quá trình thực hiện

Ba khối trung bình dưới đây thể hiện 3 pha khác nhau trong quá trình thực hiện lệnh:

- 1) Nạp lệnh, 2) giải mã, 3) thực hiện lệnh



Hình 2.2 Quá trình nạp lệnh, giải mã và thực hiện lệnh.

- Nạp lệnh

Trong pha này, vi xử lý đặt giá trị của bộ đếm chương trình vào bus địa chỉ và lấy mã lệnh từ bộ nhớ được địa chỉ hóa. Sau đó, vi xử lý ghi mã lệnh và thanh ghi câu lệnh.

- Giải mã:

Trong pha này, mã lệnh từ thanh ghi câu lệnh được giải mã nhờ bộ giải mã lệnh chế tạo ra tín hiệu điều khiển phù hợp thực hiện lệnh.

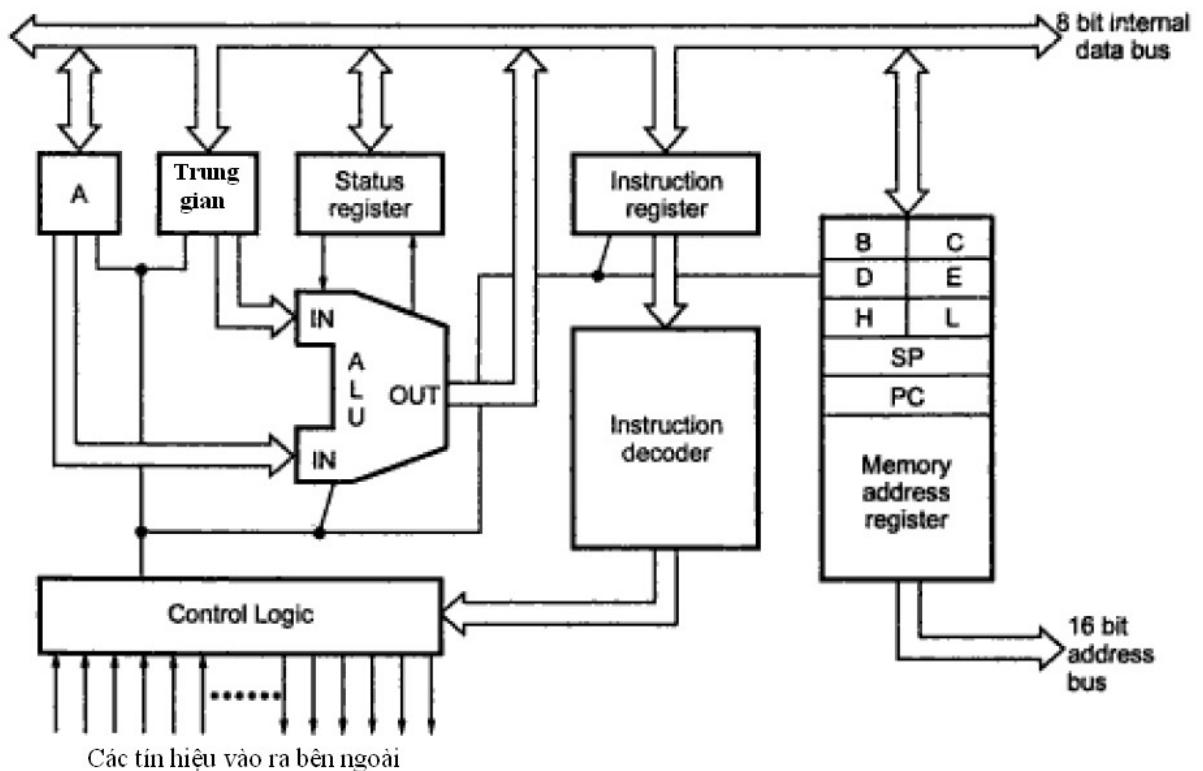
- Thực hiện:

Trong pha này, vi xử lý tạo ra tín hiệu điều khiển phù hợp để thực hiện lệnh.

III. Cấu trúc vi xử lý và hoạt động của nó

Ở phần trước ta đã xem xét 1 mô hình đơn giản của hệ thống vi xử lý. Ta đã xem xét các thành phần quan trọng của hệ thống vi xử lý như vi xử lý, bộ nhớ và đồng hồ xung nhịp trong khối vi xử lý ta đã xem xét bộ đếm chương trình giải mã lệnh, thanh ghi

lệnh, các thanh ghi khác, khối điều khiển logic và ALU. Trong phần này ta sẽ xem xét một sơ đồ khối và mô hình chương trình của một vi xử lý thực tế.



Hình 2.2 Sơ đồ khối 1 vi xử lý

Trong sơ đồ khối mô tả các hàm chức năng xử lý và quản lý dữ liệu. Nó cũng mô tả sự liên kết giữa các khối chức năng logic.

B ₇	B ₀ B ₇	B ₀
Accumulator (A)	Thanh ghi trạng thái	
Reg. B	Reg. C	
Reg. D	Reg. E	
Reg. H	Reg. L	
Con trỏ ngăn xếp		
Bộ đếm chương trình		
B ₁₅		B ₀

Hình 2.3 Mô hình chương trình

Trong số chữ khối bao gồm 3 khối logic cơ bản:

- ALU
- Các thanh ghi

Mô hình chương trình cho ta thấy quá trình hoạt động của chương trình. Sự khác nhau ở đây thể hiện ở điểm chỉ ở phần này người lập trình mới có thể thay đổi chương trình của vi xử

- Khối điều khiển.

Bus dữ liệu bên trong có vai trò kết nối giữa các logic này

ALU: một trong những khối quan trọng nhất của vi xử lý là khối logic số học (ALU). Nó chứa các phép toán số học và logic. Khối này có 2 đầu vào và 1 đầu ra. Bus dữ liệu bên trong của vi xử lý kết nối 2 đầu vào với khối ALU thông qua các thanh ghi trong giàn và thanh ghi dùng chung.

Đầu ra của ALU được kết nối với dữ liệu bên trong. Điều này cho phép gửi dữ liệu ra các ALU thông qua bus đến bất cứ thiết bị nào có kết nối với bus. Một thanh ghi đặc biệt (thanh ghi A) luôn được sử dụng khi thao tác ALU, được gọi là thanh ghi dùng chung.

Khối ALU có thể làm việc với 1 hoặc 2 đầu vào dữ liệu tùy thuộc vào loại lệnh sử dụng. Ví dụ lệnh cộng sử dụng 2 đầu vào, còn lệnh đảo thực hiện với 1 đầu vào.

Khối ALU với 1 vi xử lý thông dụng bao gồm các tính năng sau:

- Cộng
- Trừ
- AND
- OR
- XOR
- Đảo
- Dịch phải
- Dịch trái
- Tăng
- Giảm

Các thanh ghi

Các thanh ghi là một thành phần quan trọng trong các ở sơ chò khói vì phần mô hình chương trình của bất kỳ loại vi xử lý nào. Các thanh ghi cơ bản nhất của vi xử lý là thanh ghi dùng chung, bộ đếm chương trình, con trỏ ngăn xếp thanh ghi trạng thái, thanh ghi chức năng, thanh ghi địa chỉ vùng nhớ, thanh ghi câu lệnh và thanh ghi chờ lệnh tạm thời.

Thanh ghi dùng chung:

Thanh ghi dùng chung là thanh ghi có tần suất làm việc nhiều nhất của vi xử lý khi vi xử lý làm việc với cú pháp toán có 2 số lượng, thanh ghi dùng chung bao giờ cũng

chứa 1 số lượng. Số hạng khác có thể được đặt trong thanh ghi khác trong vùng bộ nhớ. Các kết quả của phép toán logic và số học được đặt trong thanh ghi dùng chung. Như vậy sau khi thực hiện xong phép toán, giá trị nguyên bản của thanh ghi dùng chung bị thay đổi do nó đã bị ghi chèn.

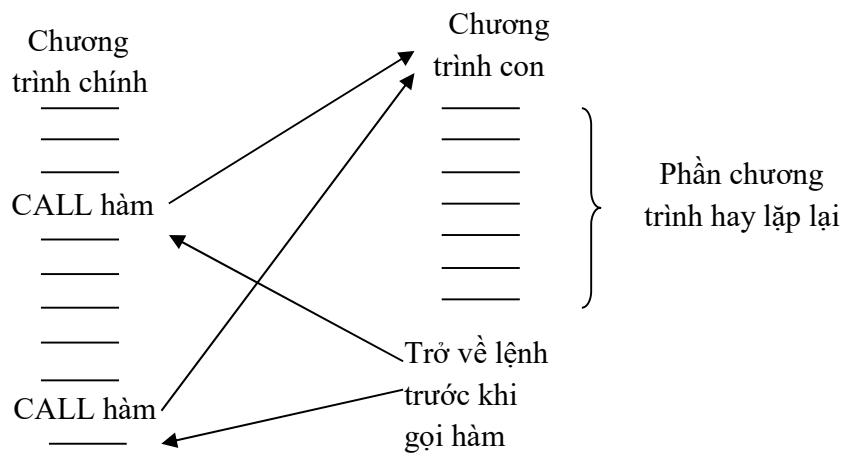
Thanh ghi dùng chung cũng được sử dụng để truyền dữ liệu 9 vừa 1 công vào ra và 1 bộ nhớ hoặc giữa 1 vùng này và 1 vùng nhớ khác.

Bộ đếm chương trình

Bộ đếm chương trình là 1 trong những thanh ghi quan trọng nhất của vi xử lý hoặc 1 cách đơn giản, 1 chương trình là tập hợp chuỗi lệnh liên tiếp đặt trong bộ nhớ. Các câu lệnh này điều hành vi xử lý cần thực hiện công việc gì và thế hiện như thế nào. Các chuỗi lệnh này được giám sát bởi bộ đếm chương trình. Nó quyết định lệnh nào được thực hiện trước tiên và lệnh tiếp theo là gì.

Bộ đếm chương trình xác định địa chỉ của vùng nhớ của lệnh kế tiếp đã được thực hiện. Chiều dài của từ bit bộ đếm chương trình sẽ quyết định dung lượng lớn nhất bộ nhớ chương trình. Ví dụ vi xử lý có 18 bit bộ đếm chương trình 16 bit có thể địa chỉ hóa được 2^{16} (64 kb) giá trị trong bộ nhớ. Trước khi vi xử lý bắt đầu thực hiện chương trình, bộ đếm chương trình cần phải nạp địa chỉ bộ nhớ quy định. Tại vùng nhớ này cần chứa mã lệnh của câu lệnh đầu tiên trong chương trình. Với đại đa số các vi xử lý vùng địa chỉ này là cố định, ví dụ là vùng nhỏ 0x0000 với bộ đếm chương trình 16 bit. Vùng địa chỉ cố định này được nạp vào bộ đếm chương trình khi vi xử lý bắt đầu chạy (Reset).

Mỗi 1 cách đơn giản, các câu lệnh cần phải được thực hiện để đạt được 1 mục đích nào đó. Tuy nhiên không phải lúc nào các câu lệnh được thực hiện tuần tự từ trên xuống dưới trong bộ nhớ chương trình. Tức là có thể có lệnh nhảy sang thực hiện các câu lệnh của 1 vùng nhớ khác không liên tiếp với vùng nhớ hiện tại. Trong trường hợp này là 1 chương trình con. Người sử dụng chương trình con khi đoạn lệnh này sử dụng nhiều lần khi thực hiện chương trình. Hình dưới đây thể hiện việc 1 chương trình chính và chương trình con hoạt động như thế nào.



Hình 2.4 Thực hiện 1 chương trình con

Thường thường, khi thực hiện xong 1 câu lệnh bộ đếm chương trình sẽ tự động tăng lên 1 giá trị để thực hiện lệnh tiếp theo. Tuy nhiên khi gọi 1 chương trình con sẽ thực hiện 1 lệnh cho phép nạp địa chỉ vùng nhớ có chứa chương trình con vào bộ đếm chương trình. Khi chương trình con được thực hiện bộ đếm chương trình sẽ chứa địa chỉ vùng nhớ có câu lệnh đầu tiên của chương trình con. Sau khi thực hiện xong chương trình con, bộ đếm chương trình sẽ được nạp địa chỉ vùng nhớ của câu lệnh tiếp theo mà trước khi chương trình con thực hiện.

Thanh ghi trạng thái:

Thanh ghi trạng thái được sử dụng để nạp các hình quả của trạng thái trong quá trình thực hiện lệnh. Các thanh ghi trạng thái còn được gọi là các thanh ghi cờ. Các cờ tương ứng với từng bit trong thanh ghi thể hiện trạng thái sau khi thực hiện phép tính trong khối ALU hoặc 1 khối phần cứng khác. Các bit cờ hay dùng nhất gồm:

- Cờ tràn :bit này được dựng lên khi cộng 2 số 8 bit có kết quả lớn hơn giá trị 0xFF.
- Zcvd : bit này được dựng lên khi nội dung thanh ghi kết quả bằng không, sau khi thực hiện lệnh trừ trường hợp lệnh giảm.
- Dấu âm :Nếu bit này có mức logic 1 thì số là số âm, nếu không là số dương.
- Parity :Kiểm tra số bit chẵn trong thanh ghi kết quả, nếu tổng số bit 1 là chẵn thì bit này set lên 1.

Con trả ngắn xép:

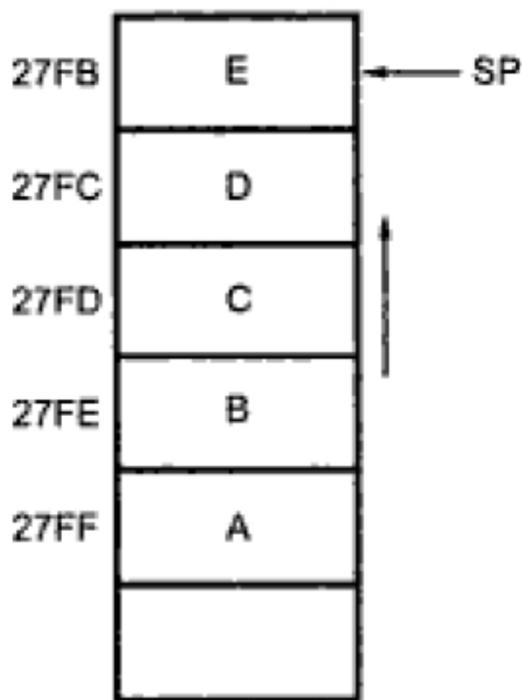
Thanh ghi này rất quan trọng khi sử dụng gọi 1 chương trình con. Khi 1 chương trình con được gọi, vi xử lý cần lưu trữ vị trí lúc gọi chương trình con để sau khi thực hiện

xong cần thực hiện lệnh tiếp theo sau khi chương trình con được gọi. Vị trí này được gọi là địa chỉ trả về được lưu trữ trong 1 vùng nhớ đặc biệt gọi là ngăn xếp. Trước khi trao quyền điều khiển cho chương trình con địa chỉ trả về cần được cất giữ vào trong ngăn xếp. Sau khi thực hiện xong chương trình con, địa chỉ trả về được lấy ra từ trong ngăn xếp và nạp vào bộ đếm chương trình. Ví dụ quá trình gọi chương trình con như sau:

Bộ nhớ chương trình	Lệnh	Bộ đếm chương trình
0x0087	MOV A,C	0x0087
0x0088	LCALL SUBROUTINE	Cất giữ giá trị 0x0088 vào stack
0x0089	MOV A,B	Nạp giá trị 0x1000 để thực hiện chương
.....		
SUBROUTINE: 0x1000	DEC A	trình con
0x1001	MOV A,C	Lấy giá trị 0x0088 từ
0x1002	RET	Ngăn xếp nạp giá trị
.....		Vào bộ đếm chương trình

Địa chỉ bộ nhớ của ngăn xếp được đặt ở thanh ghi đặc biệt gọi là con trỏ ngăn xếp. Giống như bộ đếm chương trình, con trỏ ngăn xếp được sắp xếp liên tiếp nhau trong vùng nhớ. Việc cất giữ trong ngăn xếp được thực hiện như sau:

Khi đưa dữ liệu mới vào sẽ đẩy dữ liệu cũ xuống dưới đáy, khi lấy ra ta sẽ lấy được dữ liệu mới đẩy vào. Như vậy hoạt động theo nguyên tắc vào cuối cùng thì ra đầu tiên:LIFO



Hình 2.5 Sự hoạt động của ngăn xếp

Các thanh ghi thông dụng:

Ngoài các thanh ghi như trên, trong vi xử lý sử dụng thêm một vài thanh ghi phục vụ các mục đích: khác như lưu các giá trị trung gian khi thực hiện lệnh. Tốc độ truy nhập các thanh ghi này có đặc điểm là nhanh hơn truy nhập bộ nhớ thông thường

- *Thanh ghi chứa địa chỉ bộ nhớ:*

Thanh ghi này chứa địa chỉ của vùng nhớ mà vi xử lý sử dụng. Tùy thuộc vào loại vi xử lý mà độ dài của thanh ghi này là 16 hay 32 bit. Đầu ra của thanh ghi địa chỉ bộ nhớ điều khiển bus 16 bit (32) địa chỉ. Nhiệm vụ của nó là lựa chọn vùng nhớ của vi xử lý sử dụng.

- *Thanh ghi lệnh:*

Thanh ghi lệnh chứa mã lệnh thực hiện. Giá trị được nạp vào thanh ghi ở chu kỳ đầu thực hiện lệnh. Nội dung của thanh ghi được sử dụng để điều khiển khối logic giải mã lệnh.

- *Thanh ghi dữ liệu tạm thời:*

Trong quá trình thực hiện lệnh tính toán, nhiều trường hợp cần lưu giá trị kết quả đạt được. Mà trong khối ALU không có khối lưu kết quả do đó kết quả có thể được lưu trong thanh ghi dùng chung hoặc thanh ghi dữ liệu tạm thời.

Khối điều khiển logic:

Khối điều khiển logic là 1 khối quan trọng trong vi xử lý. Nó điều hành sự phối hợp hoạt động của tất cả các khối khác trong vi xử lý, hay nói cách khác nó đồng bộ sự họa động của các khối chức năng khác nhau. Quá trình đồng bộ này hoạt động theo nhịp thời gian của xung đồng hồ vi xử lý.

Khối điều khiển logic nhận tín hiệu từ bộ giải mã lệnh, và nó sẽ tạo ra tín hiệu điều khiển cần thiết tương ứng với mã lệnh này. Ngoài ra nó còn có 1 vài chức năng đặc biệt khác như quản lý quá trình khôi định vị xử lý, quản lý ngắt. Ngắt là 1 lệnh đặc biệt đòi hỏi vi xử lý thực hiện 1 nhiệm vụ khi có yêu cầu từ thiết bị ngoài, lúc này vi xử lý sẽ tạm dừng thực hiện hiện tại chuyển sang thực hiện hoãn ngắt.

Bus dữ liệu bên trong:

Bus dữ liệu bên trong thực hiện việc kết nối các phần khác nhau của vi xử lý và cho phép chúng truyền dữ liệu với nhau. Dữ liệu được truyền thông qua bus dữ liệu bên trong nhờ khối điều khiển logic.

Bus dữ liệu bên trong của vi xử lý thông thường được nối với bus dữ liệu bên ngoài. Nhờ vậy mà vi xử lý có thể liên kết được với các bộ nhớ bên ngoài và với các thiết bị ngoại vi.

CHƯƠNG 3. VI XỬ LÝ 8085

I. Đặc tính của vi xử lý 8085

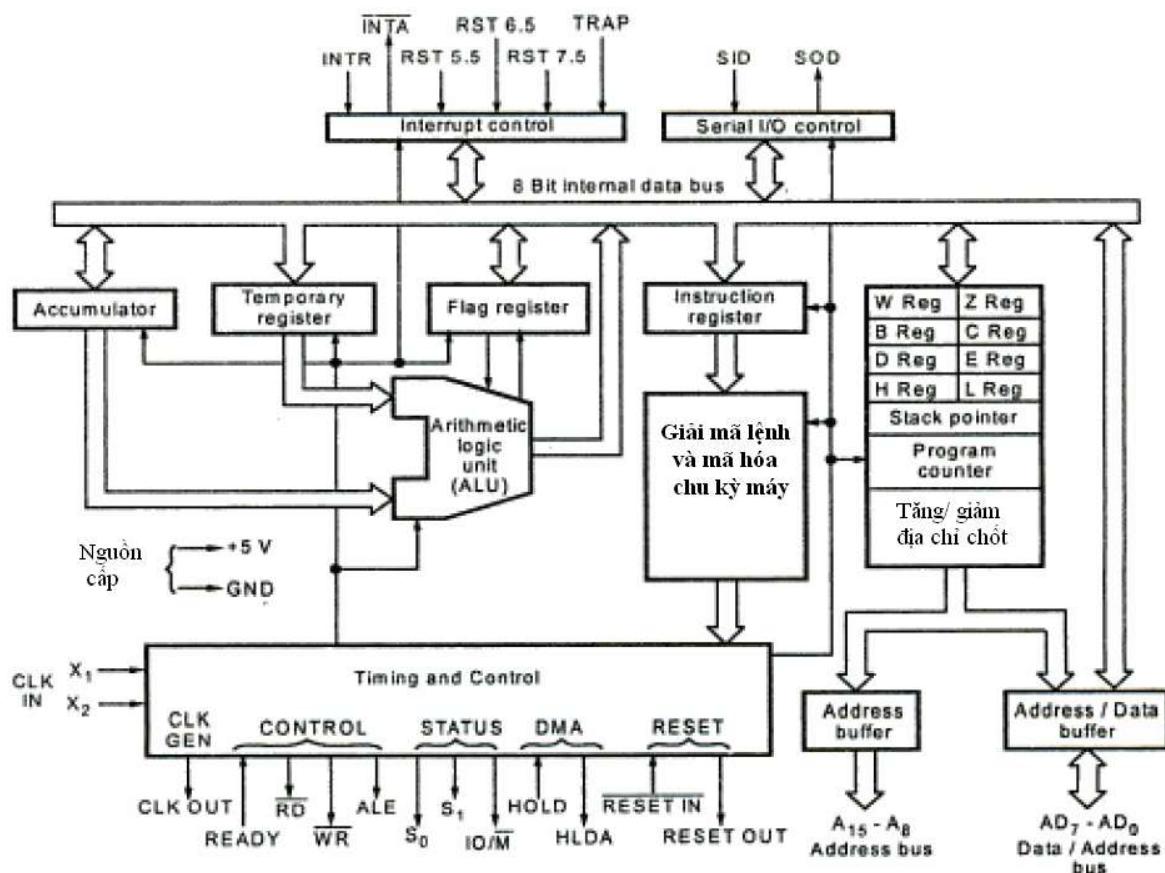
- Vi xử lý 8 bit
- Cấp nguồn đơn 5V
- Hoạt động cung xung nhịp tần số tối đa 5MHz, có bộ tạo dao động bên trong dùng LC, RC hoặc thạch anh.
- Có 16 bit địa chỉ, bộ nhớ như vậy có thể truy nhập bộ nhớ có dung lượng tối đa là (2^{18}) 64KB.
- Có 8 bit địa chỉ cổng vào ra, như vậy có thể điều khiển được (2^8) 256 cổng vào ra.
- Sử dụng bộ trộn hình giữa 8 bit thấp địa chỉ Bus (A₀, A₇) và bus dữ liệu (D₀-D₇) đưa ra 8 cổng ra. như vậy cần tách kênh các tín hiệu này ở phía ngoài vi xử lý .
- Sử dụng 72 câu lệnh với các chế độ địa chỉ hóa:
 - 1) trực tiếp
 - 2) thanh ghi
 - 3) gián tiếp
 - 4) trung gian
 - 5) implied
- Khối logic số học ALU có các chức năng:
 - ✓ bộ cộng nhị phân 8 bit báo hoặc không báo cờ tràn
 - ✓ bộ cộng nhị phân 16 bit
 - ✓ bộ cộng 2 digit BCD
 - ✓ bộ trừ 8 bit báo hoặc không báo cờ tràn
 - ✓ thực hiện phép logic 8 bit: AND, OR, NOR, NOT, và phép dịch bit trái, dịch bit phải.
- Các thanh ghi gồm thanh ghi 8 bit dùng chung, thanh ghi vò, thanh ghi lệnh 6 thanh ghi chức năng chung (B,C,D,E,H và L) và thanh ghi 16 bit (SP và PC).
- Có 5 ngắt cứng TRAT, RST7.5, RST6.5, RST5.5 và INTR.
- Có cổng vào ra nối tiếp cho phép truyền thông nối tiếp.
- Có các tín hiệu điều khiển (IO/M , \overline{RD} , \overline{WR}) để điều khiển quá trình đọc ghi dữ liệu.
- Cho phép sử dụng chế độ truy xuất bộ nhớ trực tiếp DMA.

- Có thể kết quả 3 chip vi xử lý khi sử dụng các chip điều khiển I/O như IC8155 hoặc IC8355 .

II. Cấu trúc của vi xử lý 8085

Xem hình vẽ dưới đây, các khối chức năng chính bao gồm:

- Các thanh ghi
- Khối logic số học ALU
- Bộ giải mã lệnh và mã hóa chu kỳ máy
- Bô đệm địa chỉ
- Bộ đệm địa chỉ / dữ liệu
- Bộ tăng/ giảm địa chỉ chốt
- Điều khiển ngắn
- Điều khiển cổng vào ra nối tiếp
- Mạch tạo dao động và điều khiển



Hình 3.1 Cấu trúc vi xử lý 8085

2.1 Cấu trúc của thanh ghi

Hình dưới thể hiện cấu trúc các thanh ghi của vi xử lý 8085. Các thanh ghi mà người sử dụng có thể thay đổi giá trị (lập trình) bao gồm 6 thanh ghi chức năng chung 8 bit (B, C, D, E, H, L), thanh ghi dùng chung A, thanh ghi cờ và 2 thanh ghi 16 bit (SP và PC). Các thanh ghi không thay đổi từ chương trình người sử dụng là thanh ghi tạm thời, thanh ghi w và z. chúng bị thay đổi nội dung bởi các lệnh bên trong của vi xử lý.

Thanh ghi tạm thời	
W Reg	Z Reg
A Reg	Flag Reg
B Reg	C Reg
D Reg	E Reg
H Reg	L Reg
Con trỏ ngăn xếp (SP)	
Bộ đếm chương trình (PC)	

Hình 3.2 Cấu trúc thanh ghi

2.1.1 Các thanh ghi chức năng chung:

Gồm 6 thanh ghi 8 bit B, C, D, E, H, L, có thể dùng như các thanh ghi đơn 8 bit hoặc ghép lại thành từng cặp thành thanh ghi 16 bit như BC, DE và HL. Khi ghép lại thành từng cặp thì byte cao được đặt trong thanh ghi đầu tiên (ví dụ thanh ghi B trong cặp BC), B là byte cao và C là byte thấp.

Thanh ghi HL có thể được sử dụng như 1 con trỏ dữ liệu trong con trỏ chương trình. Để ghi và đọc giá trị trong các thanh ghi này, sử dụng 1 lệnh bên trong mà không cần truy nhập Bus. Chính vì vậy, mà tốc độ truy nhập cao hơn so với truy nhập vào bộ nhớ thông thường.

2.1.2 Các thanh ghi tạm thời

a. Thanh ghi lưu dữ liệu tạm thời

Khối ALU có 2 đầu vào, 1 đầu vào ghi dữ liệu từ thanh ghi dùng chung A và 1 đầu vào từ thanh ghi lưu dữ liệu tạm thời. Người lập trình không thể trực tiếp truy xuất vào thanh ghi này. Vì xử lý sẽ tự động sử dụng thanh ghi này khi thực hiện các câu lệnh liên quan đến khối ALU.

Ví dụ: lệnh ADD B thực hiện phép toán như sau: lấy giá trị trong thanh ghi B cùng với giá trị trong thanh ghi A, kết quả ghi vào thanh ghi A. Phép cộng này được thực hiện trong khối ALU. Khối ALU sẽ lấy dữ liệu để cộng từ thanh ghi A và thanh ghi

lưu trữ dữ liệu tạm thời, như vậy trước khi khởi động ALU thực hiện lệnh thì vi xử lý sẽ chuyển nội dung trong thanh ghi B vào thanh ghi lưu trữ dữ liệu tạm thời.

b. Các thanh ghi w và z:

Các thanh ghi w và z là các thanh ghi tạm thời. Các thanh ghi này sử dụng để giữ dữ liệu 8 bit trong quá trình thực hiện 1 số lệnh.

Ví dụ: lệnh CALL (gọi 1 chương trình con) được dùng để chuyển quyền điều khiển chương trình đến 1 chương trình con. Lệnh này sẽ cắt giữ bộ đếm chương trình (PC) hiện tại vào trong ngăn xếp và nạp địa chỉ chương trình con vào bộ đếm chương trình (PC). Địa chỉ tạm thời này được chứa trong thanh ghi w và z và được đưa ra bus ở chu kỳ đầu thực hiện lệnh (Fetch).

Lệnh XCHG hoán đổi nội dung của các thanh ghi H và D với E cho nhau: khi thực hiện quá trình hoán đổi từ các thanh ghi w và z được sử dụng để lưu giá trị trung gian.

2.1.3 Các thanh ghi chức năng đặc biệt

a. Thanh ghi dùng chung A (accumulator): đây là thanh ghi 3 trạng thái 8 bit (tristate), nó tuy nhiên hay được dùng nhất trong các phép toán số học, logic, nạp và ghi dữ liệu, truy nhập các cổng vào/ ra. Tất cả các kết quả của phép toán số học và logic đều được nạp vào thanh ghi A, do tần suất thực hiện các lệnh này lớn nên thông thường thanh ghi A hay được sử dụng nhất.

b. Thanh ghi cờ:

Đây là thanh ghi 8 bit, trong đó có 5 bit được sử dụng để báo dấu hiệu để nhận biết của trạng thái sau quá trình thực hiện lệnh nào đó. Chúng bao gồm các cờ: s (cờ dấu), z (cờ zero), AC (cờ báo tràn phụ), P (cờ báo chẵn lẻ) và CY (cờ tràn).

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
s	z	x	AC	x	P	x	CY

Hình 3.3 Thanh ghi cờ

Cờ dấu s: sau khi thực hiện phép toán số học và logic nếu bit D₇ của kết quả có giá trị 1, thì cờ dấu s được đặt giá trị 1. Như vậy nếu byte nhận được có bit D₇ là 1 thì số đó là số âm, ngược lại là số dương.

Cờ zero z: cờ này được đặt lên 1 nếu kết quả thực hiện trong ALU có giá trị 0 và cờ này sẽ bị xóa (0) nếu kết quả $\neq 0$. Cờ zero cũng được đặt lên 1 khi sử dụng lệnh tăng hoặc giảm giá trị của 1 thanh ghi nào đó mà kết quả có giá trị 0.

Cờ tràn phụ AC: cờ này được đặt lên 1 trong trường hợp bị tràn bit số 3, tức là giá trị từ bit D₃ bị đẩy sang bit D₄. Cờ này được sử dụng khi thực hiện lệnh với dữ liệu BCD và người sử dụng không thay đổi được giá trị của cờ này.

Cờ chẵn lẻ P (Parity): bit kiểm tra chẵn lẻ dùng để xác định số lượng bit 1 trong thanh ghi dùng chung A là chẵn hay lẻ. sau phép toán số học hoặc logic nếu trong kết quả có số lượng bit 1 là chẵn thì cờ này được đặt lên 1, nếu là lẻ thì giá trị của nó sẽ là 0.

Cờ tràn CY: giá trị cờ này được đặt lên 1 nếu bị tràn bit 7.

Ví dụ phép tính có cờ tràn như sau:

$$\begin{array}{r} 10011011 \\ + 01110101 \\ \hline 1\ 00010000 \end{array} \quad \begin{array}{r} 10001001 \\ - 10101011 \\ \hline 1\ 11011110 \end{array}$$

c. Thanh ghi lệnh:

Khi thực hiện lệnh, đầu tiên mã lệnh sẽ được nạp vào vi xử lý từ bộ nhớ chương trình (vi xử lý sẽ đưa giá trị địa chỉ vào vùng nhớ ra bus địa chỉ, bộ nhớ chương trình sẽ đưa dữ liệu tại vùng nhớ chương trình vào bus dữ liệu). vi xử lý sẽ lưu mã lệnh vào thanh ghi lệnh. Mã lệnh này sau đó sẽ được giữ vào bộ giải mã lệnh để lựa chọn 1 trong số 256 giá trị tương ứng.

2.1.4 Các thanh ghi 16 bit:

a. Bộ đếm chương trình (PC): một chương trình là 1 chuỗi các câu lệnh. Nói cách khác, vi xử lý được nạp các lệnh này từ bộ nhớ chương trình và thực hiện chúng 1 cách lần lượt. bộ đếm chương trình là 1 thanh ghi chức năng đặc biệt cho phép điều hành thời điểm nạp lệnh tiếp theo để thực hiện. Bộ đếm chương trình hoạt động như 1 con trỏ chỉ đếm lệnh tiếp theo. Vi xử lý tăng giá trị của bộ đếm chương trình như thế nào thì phụ thuộc vào bản chất của lệnh, với lệnh có 1 byte thì nó sẽ tăng lên 1 giá trị sau khi thực hiện xong lệnh, với lệnh có 2 byte thì tăng 2 giá trị và lệnh 3 byte sẽ tăng 3 giá trị. như vậy con trỏ chương trình luôn chỉ đếm lệnh tiếp theo.

Trong trường hợp sử dụng lệnh nhảy JUMP và lệnh gọi chương trình con CALL thì địa chỉ khai báo trong lệnh JUMP và CALL sẽ được nạp vào bộ đếm chương trình. Vi xử lý sẽ nạp mã lệnh nằm ở địa chỉ được khai báo trong lệnh JUMP và CALL.

b. Con trỏ ngăn xếp (SP)

Ngăn xếp là 1 vùng nhớ dành riêng ở trong RAM, nơi mà lưu các thông tin tạm thời. một con trỏ ngăn xếp 16 bit được sử dụng để lưu giữ địa chỉ trả về trong quá trình gọi chương trình con.

2.2 Khối tính toán logic số học (ALU)

Khối ALU của vi xử lý 8085 làm việc với 8 bit dữ liệu. Các phép toán số học (như cộng và trừ) làm việc ở mức bit. Khối logic thực hiện các lệnh như đảo (NOT), AND, OR, X-OR hoặc lệnh dịch bit và xóa.

2.3 Bộ giải mã lệnh

Như đã nói ở trên, khi thực hiện lệnh đầu tiên vi xử lý nạp mã lệnh từ bộ nhớ vào thanh ghi lệnh. Sau đó sẽ giải mã lệnh đến bộ giải mã lệnh, bộ giải mã lệnh giải mã nó và tạo tín hiệu điều khiển với nhịp thời gian phù hợp để điều khiển thanh ghi, bộ đếm dữ liệu, ALU, tín hiệu ngoại vi phụ thuộc câu lệnh cụ thể.

2.4 Bộ đếm địa chỉ (address buffer)

Đây là bộ đếm điều khiển 1 chiều 8 bit. Nó được sử dụng để điều khiển Bus địa chỉ cao (A8 ÷ A5), đồng thời được sử dụng để chuyển về trạng thái trở kháng cao (high-z) Bus địa chỉ bit cao khi gặp các trường hợp như khởi động (reset), tạm dừng, dừng hay các dòng địa chỉ này không dùng.

2.5 Bộ đếm địa chỉ /dữ liệu

Đây là bộ đếm điều khiển 2 chiều 8 bit. Nó được dùng để điều khiển việc chọn kênh giữa kênh địa chỉ và kênh dữ liệu, tức là bus địa chỉ bit thấp (A7 ÷ A0) và Bus dữ liệu (D0 ÷ D7). Đồng thời được sử dụng để chuyển về trạng thái trở kháng cao bus chọn kênh địa chỉ / dữ liệu khi gặp các trường hợp như khởi động (reset), tạm dừng, dừng hay các đường dữ liệu này không dùng.

2.6 Chốt địa chỉ tăng hay giảm

Thanh ghi 16 bit này được sử dụng để tăng hay giảm nội dung của bộ đếm chương trình (PC) hoặc con trỏ ngăn xếp.

2.7 Điều khiển ngắt

Vi xử lý nạp lệnh, gửi mã và thực hiện lệnh 1 cách tuần tự, các câu lệnh được thực hiện liên tiếp nhau theo trình tự từ trên xuống dưới theo địa chỉ tăng dần của bộ nhớ. Tuy nhiên có những trường hợp đòi hỏi vi xử lý thực hiện nhiệm vụ đặc biệt theo 1 yêu cầu bất thường nào đó. Ví dụ, giả sử thấy vi xử lý đang điều khiển lái 1 xe ô tô trên đường, thông thường lái xe bằng cách tăng tốc, giảm tốc, đánh lái trái, phải, tuy

nhiên, đột nhiên trên đường xuất hiện chướng ngại vật, lúc này bộ phận điều khiển vẫn phải xử lý đặc biệt như phanh và tránh vật cản. như vậy có 1 đáp ứng đặc biệt cần thực hiện phù hợp với sự kiện xuất hiện khi 1 quá trình ngắn được thực hiện , chương trình đang chạy được tạm dừng chuyển sang chạy 1 hàm ngắn để thực hiện đáp ứng đặc biệt. sau khi hàm ngắn thực hiện , chương trình đang chạy được thực hiện tiếp từ điểm tạm dừng.

Điều khiển truyền thông nối tiếp:

Phục vụ việc truyền dữ liệu từ vi xử lý này đến vi xử lý khác hay tới máy tính thông thường sử dụng kênh truyền thông nối tiếp không đồng bộ UART. Lợi thế của kênh truyền thông này là sử dụng ít dây kết nối do đó giảm giám thành và khả năng chống nhiễu cao hơn kênh truyền song song, đặc biệt là truyền với khoảng cách lớn. Với kênh truyền nối tiếp, trên đường truyền sẽ gửi từng bit tại một thời điểm. Trong vi xử lý 8085, điều khiển cổng vào ra nối tiếp có 2 đường là dữ liệu đầu ra nối tiếp (SOD) và dữ liệu đầu vào nối tiếp (SID). Đường SOD được sử dụng để gửi dữ liệu nối tiếp, đường SID được sử dụng nhận dữ liệu nối tiếp.

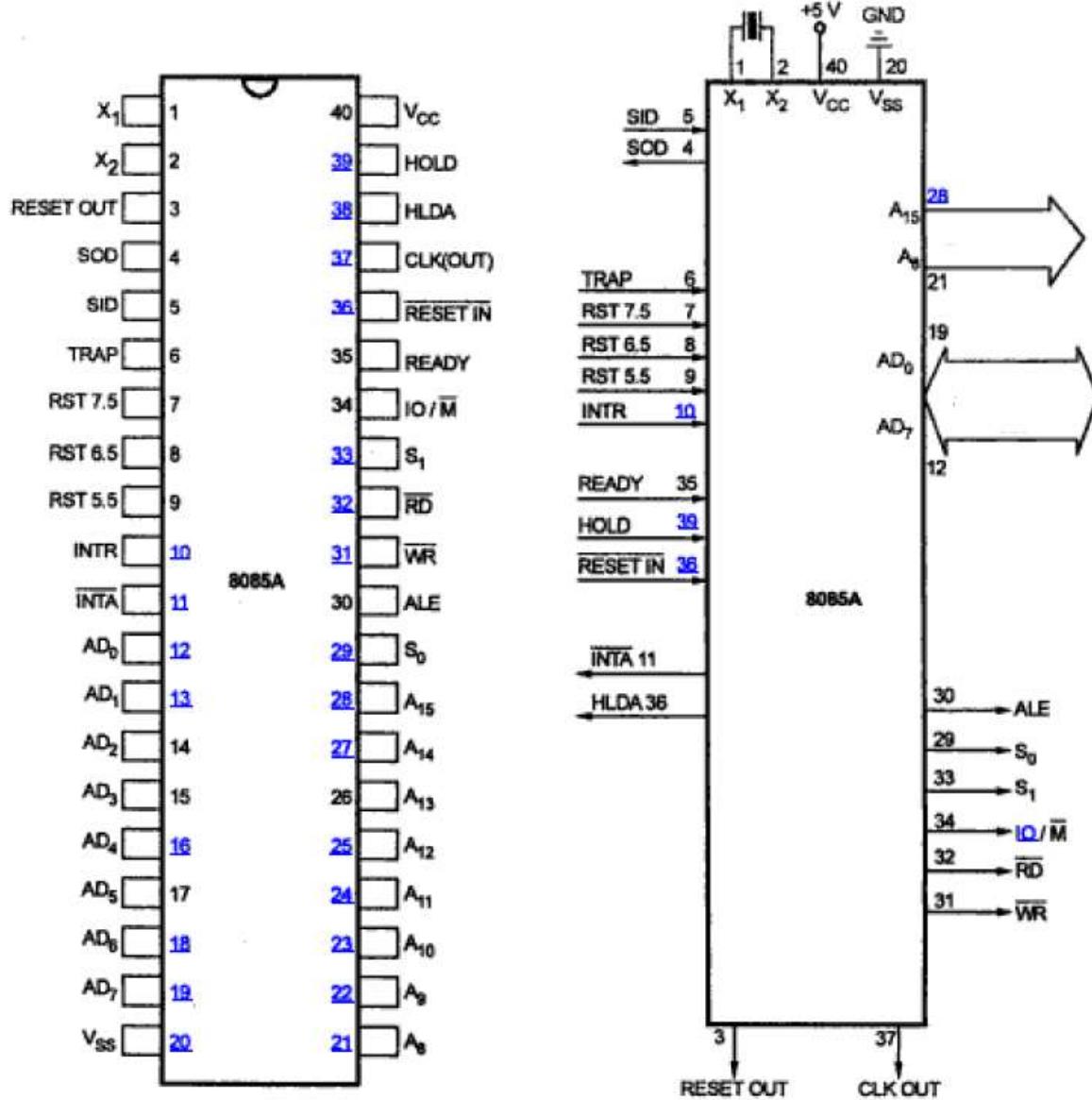
Mạch điều khiển và đồng bộ thời gian.

Mạch điều khiển trong vi xử lý 8085 có nhiệm vụ điều khiển toàn bộ các quá trình hoạt động của vi xử lý, nó gồm việc đồng bộ xung nhịp, nạp lệnh, giải mã lệnh, tạo các tín hiệu phù hợp để thực hiện lệnh. Mạch điều khiển còn tạo các tín hiệu cho các khối phần cứng kết nối các thiết bị ngoại vi.

III. Sơ đồ chức năng các chân vi xử lý.

Sơ đồ ghép nối và chức năng của các chân vi xử lý 8085 thể hiện ở hình dưới. Các tín hiệu vào ra vi xử lý có thể phân loại thành 7 nhóm như sau:

- Các chân tạo dao động và cấp nguồn.
- BUS dữ liệu và BUS địa chỉ.
- BUS điều khiển.
- Tín hiệu ngắn.
- Tín hiệu vào ra nối tiếp.
- Tín hiệu truy nhập bộ nhớ trực tiếp DMA.
- Tín hiệu Reset.



Hình 3.4 Sơ đồ chức năng chân

3.1 Các chân tạo dao động và cấp nguồn.

- Vcc: Chân cấp nguồn đơn cực +5V.
- Vss: Chân đất.
- X₁ và X₂ : mạch cộng hưởng như LC, RC, hoặc thạch anh được kết nối vào hai chân này. Mạch tạo dao động bên trong sau khi tạo ra dao động, sẽ chia tần số với hệ số 2. Ví dụ đèn hệ thống vi xử lý làm việc với tần số 3Mhz, mạch dao động cần tạo tần số 6Mhz.
- CLK OUT: Tín hiệu này được sử dụng như tín hiệu xung nhịp cho thiết bị khác. Tần số này bằng $\frac{1}{2}$ tần số được tạo ra từ mạch dao động.

3.2 BUS dữ liệu và BUS địa chỉ:

- AD₀ đến AD₇: Các chân của 8 tám bit dữ liệu (D₀ đến D₇) được sử dụng chung với tám bit địa chỉ thấp (A₀ đến A₇), và hoạt động theo nguyên tắc dồn kênh (Multiplex). Tại phần đầu của chu kỳ máy (T₁) 8 bit thấp của địa chỉ vùng nhớ hoặc địa chỉ vào ra được áp lên BUS. Tại các phần sau của chu kỳ máy (T₂ và T₃) đường BUS này được sử dụng cho 8 bit dữ liệu 2 chiều.
- A₈ đến A₁₅: Phần bit cao của 16 bit địa chỉ được áp vào các đường dây địa chỉ từ A₈ đến A₁₅. Các chân này được sử dụng chuyên cho đường BUS địa chỉ các bit cao (từ A₈ đến A₁₅).

3.3 Tín hiệu trạng thái và điều khiển:

- ALE (Address Latch Enable – Cho phép chốt địa chỉ): Như trên ta đã biết các chân AD₀ đến AD₇ được dùng chung cho BUS địa chỉ (8 bit thấp) và BUS dữ liệu, và chúng hoạt động theo chế độ dồn kênh (Multiplex). Vào thời điểm T₁ của chu kỳ máy, BUS địa chỉ sẽ được đưa ra các chân này và thời điểm T₂ và T₃, BUS dữ liệu sẽ được đưa ra. Như vậy BUS địa chỉ cần được chốt (giữ) lại sau thời điểm T₁. Việc chốt các địa chỉ bit thấp được thực hiện bởi tín hiệu “Cho phép chốt địa chỉ-ALE” và được đưa ra bên ngoài vi xử lý 8085
- RD và WR: các tín hiệu này được sử dụng để điều khiển chiều của dòng dữ liệu giữa vi xử lý và bộ nhớ hoặc thiết bị vào ra. Mức thấp của chân RD thể hiện rằng một dữ liệu cần được đọc từ vùng nhớ hoặc công vào ra lựa chọn thông qua BUS dữ liệu. Mức thấp của chân WR thể hiện rằng một dữ liệu cần phải ghi vào vùng nhớ hoặc công vào/ra lựa chọn thông qua BUS dữ liệu.
- IO/M, S₀ và S₁: IO/M thể hiện việc lựa chọn đang làm việc với cổng vào/ra hay bộ nhớ. S₀ và S₁ thể hiện dạng của chu kỳ máy hoạt động.
- READY: Được sử dụng để vi xử lý nhận biết được trạng thái sẵn sàng của các thiết bị ngoại vi mà nó kết nối. Nếu thiết bị ngoại vi chưa sẵn sàng thì vi xử lý cần phải đợi khi kết nối. Tín hiệu này được sử dụng để đồng bộ với các thiết bị ngoại vi hoạt động chậm hơn so với vi xử lý.

3.4 Tín hiệu ngắn.

Vi xử lý 8085 có 5 tín hiệu ngắn cứng: RST5.5, RST 6.5, RST7.5, TRAP và INTR. Vi xử lý sẽ xử lý các đòi hỏi ngắn này tại thời điểm cuối của lệnh đang thực hiện.

Tín hiệu INTA (Acknowledge Interrupt- Ngắn báo nhận) được sử dụng để thông báo vi điều khiển đã nhận được 1 tín hiệu ngắn INTR.

3.5 Tín hiệu vào ra nối tiếp.

- Tín hiệu SID (Dữ liệu vào nối tiếp): Tín hiệu đầu vào được sử dụng để nhận từng bit dữ liệu nối tiếp từ thiết bị ngoại vi
- Tín hiệu SOD (Dữ liệu ra nối tiếp): Tín hiệu đầu ra được sử dụng để truyền từ bit dữ liệu nối tiếp đến thiết bị ngoại vi.

3.6 Tín hiệu DMA:

- HOLD: Tín hiệu này thông báo có một trạm chủ khác đang đòi hỏi truy nhập (sử dụng) BUS địa chỉ, dữ liệu và điều khiển của vi xử lý.
- HLDA: Tín hiệu tích cực ở mức cao sử dụng để báo hiệu sự chấp nhận đòi hỏi HOLD

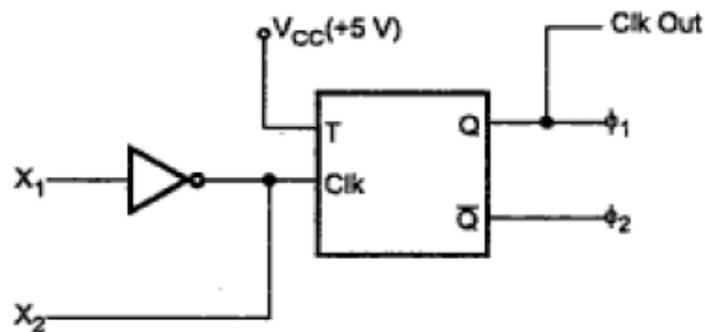
3.7 Tín hiệu Reset:

- SESET IN: Khi chân này ở mức thấp thì:
 - + Bộ đếm chương trình chuyển về địa chỉ 0 (0000H)
 - + Xóa trạng thái cho phép ngắt và Flip-flops HLDA.
 - + Chuyển BUS dữ liệu, địa chỉ và điều khiển về trạng thái trở kháng cao (High Z).Chú ý là chỉ thực hiện khi tín hiệu Reset ở trạng thái tích cực.
 - + Chuyển nội dung các thanh ghi bên trong của vi xử lý về giá trị ngầm định.Khi Reset, con trỏ chương trình chuyển về địa chỉ 0000 là do vi xử lý 8085 bắt đầu thực hiện lệnh đầu tiên từ địa chỉ 0000H. Để chức năng Reset được thực hiện đúng, tín hiệu Reset cần giữ ở mức thấp ít nhất trong 3 chu kỳ lệnh. Mạch Reset khi bật nguồn (Power-on) cần thiết để chắc chắn rằng khi vi xử lý khi bắt đầu hoạt động sẽ thực hiện lệnh tại địa chỉ 0000H.
- RESET OUT: Tín hiệu tích cực ở mức cao để thông báo là vi xử lý bắt đầu reset. Tín hiệu này được đồng bộ với đồng hồ xung nhịp vi xử lý và có thể sử dụng reset thiết bị khác có kết nối với vi xử lý.

IV. Các mạch điện kết nối vi xử lý

4.1 Mạch tạo dao động.

Vi xử lý 8085 có mạch tạo dao động bên trong, hình dưới đây hiện sơ đồ khói bộ tạo dao động. Mạch tạo dao động bên trong đòi hỏi một bộ cộng hưởng ví dụ như LC, RC hay thạch anh. Bộ Flip-flop bên trong sẽ chia tần số dao động với hệ số 2. Như vậy vi xử lý 8085 sẽ hoạt động với xung nhịp bằng $\frac{1}{2}$ tần số dao động mạch cộng hưởng.

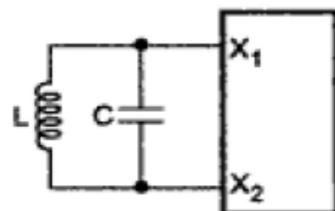


Hình 3.6 Mạch tạo dao động bên trong

Mạch cộng hưởng LC: Sử dụng mạch theo sơ đồ dưới đây, tần số dao động được tính theo công thức:

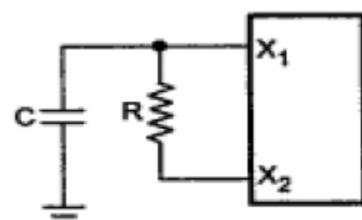
$$f_r = \frac{1}{2\pi\sqrt{L(C_{ext} + C_{int})}}$$

Trong đó C_{int} là điện dung bên trong, thông thường có giá trị là 15pF. Tần số dao động ra có sai lệch khoảng 10%. Để giảm thiểu sự sai lệch của tần số dao động ra, cần sử dụng tụ C_{ext} có dung lượng lớn gấp đôi so với C_{int} , tức là khoảng 30pF.



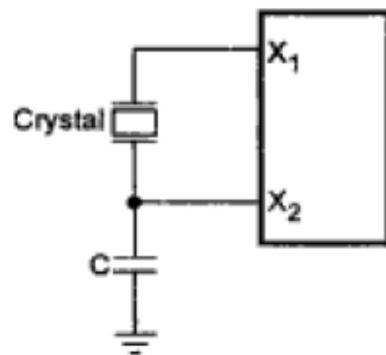
Hình 3.7 Mạch cộng hưởng LC

Mạch cộng hưởng RC: Hình dưới đây hiện mạch dao động sử dụng RC. Tần số dao động ra của mạch này không ổn định, nhưng mạch này có ưu điểm là giá thành thấp.



Hình 3.8 Mạch cộng hưởng RC

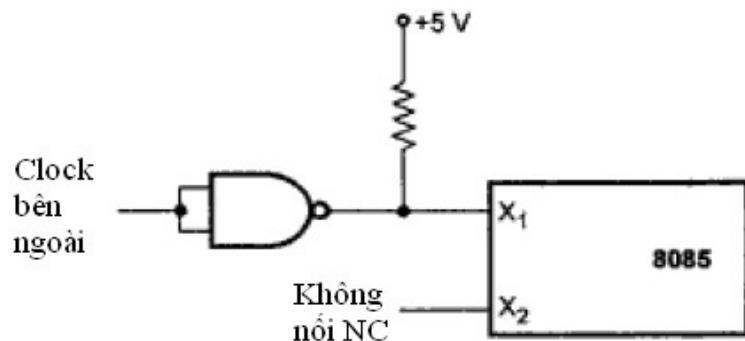
Mạch cộng hưởng thanh anh thể hiện ở hình dưới. Mạch này có độ ổn định tần số cao nhất. Sử dụng 1 tụ dung lượng 20pF nối đất để mạch dễ dao động khi mới bật điện.



Hình 3.9 Mạch cộng hưởng thạch anh

Mạch dao động ngoài.

Có thể sử dụng dao động từ bên ngoài cấp vào vi xử lý theo mạch như hình dưới. Ở đây xung nhịp bên ngoài được cấp cho chân X_1 và chân X_2 sẽ được để trống.

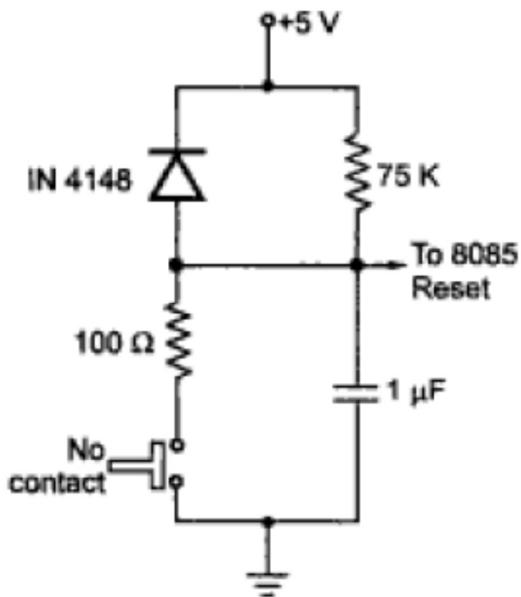


Hình 3.10 Mạch dao động ngoài

4.2 Mạch Reset.

Khi Reset, con trỏ chương trình chuyển về địa chỉ 0000H do khi bắt đầu hoạt động, vi xử lý 8085 thực hiện lệnh đầu tiên từ địa chỉ 0000H.

Khi bắt đầu bật nguồn reset cần được sử dụng để chắc chắn là lệnh đầu tiên được thực hiện từ địa chỉ 0000H. Hình dưới thể hiện mạch reset hoạt động khi bật nguồn, mạch này sử dụng linh kiện RC.



Hình 3.11 mạch Reset

Khi bật nguồn chân reset-in cần ở trạng thái mức thấp ít nhất trong khoảng thời gian 10 ms sau đó dùng điện áp lên Vcc. Trong mạch hình dưới, khi bật nguồn hoặc ấn nút, chân reset-in sẽ xuống mức thấp và từ từ tăng lên điện áp 5V đủ thời gian để vi xử lý reset. Diode được dùng để xả điện áp trên tụ khi nguồn bị cắt.

Sau khi reset, vi xử lý nạp địa chỉ 0000H vào bộ đếm chương trình PC và xóa cờ INTE. Cờ này có tác dụng cầm ngắt. Thông thường khi vi xử lý bắt đầu chỉ hoạt động cần khởi tạo các tham số của hệ thống, để tránh ảnh hưởng của ngắt trong quá trình này ta cần cầm các ngắt hoạt động. Sau khi khởi tạo tham số xong, nếu cần 1 ngắt nào đó hoạt động có thể sử dụng lệnh EI.

4.3 Cấu trúc và sự hoạt động của ngắt trong vi xử lý 8085

Các dạng ngắt:

Vi xử lý 8085 là hệ thống có ngắt đa mức, hỗ trợ 2 dạng ngắt là ngắt cứng và ngắt mềm.

Ngắt cứng (Hardware):

Một vài chân của vi xử lý 8085 cho phép thiết bị ngoại vi và tạm dừng sự hoạt động của chương trình chính để thực hiện 1 chương trình đặc biệt đáp ứng tác động của thiết bị ngoại vi. Khi 1 ngắt hoạt động, vi xử lý sẽ thực hiện nốt lệnh hiện tại, và chuyển sang thực hiện 1 hàm để phục vụ thiết bị ngoại vi. Khi thực hiện xong hàm này, vi xử lý lại trở về chương trình chính tại thời điểm vừa rời khỏi. Dạng ngắt này, khi các chân vi xử lý được dùng để nhận các đòi hỏi ngắt được gọi là ngắt cứng.

Ngắt mềm: với ngắt mềm, nguyên nhân xảy ra ngắt là do thực hiện 1 lệnh nào đó. Sau khi thực hiện các lệnh này vi xử lý sẽ thực hiện nốt lệnh đang hoạt động và chuyển sang gọi 1 hàm đặc biệt. Khi thực hiện xong hàm này, chương trình điều khiển lại chuyển về thực hiện chương trình chính.

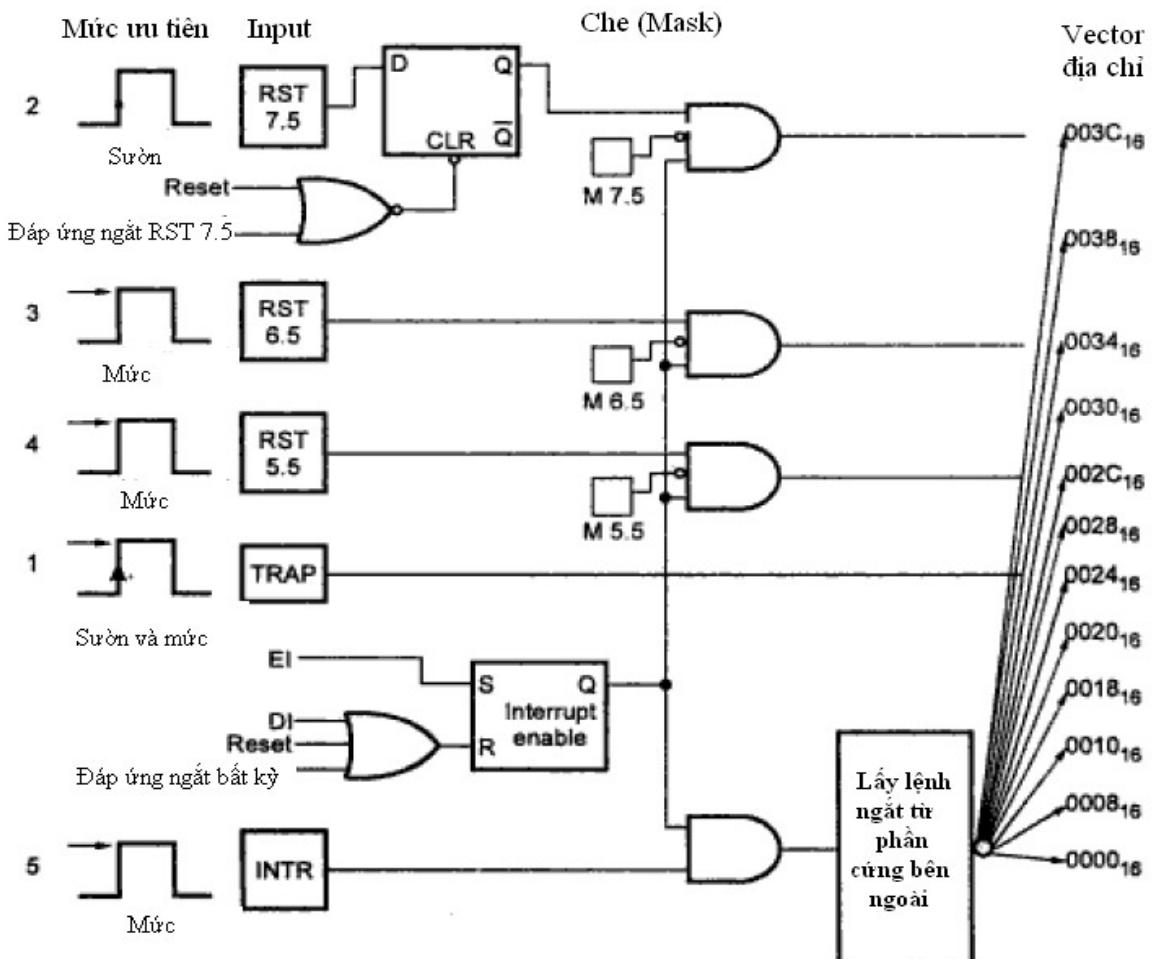
4.3.1 Tổng quan về cấu trúc ngắt:

Vi xử lý gồm có 2 dạng ngắt: ngắt cứng và ngắt mềm:

- **Ngắt cứng trong vi xử lý 8085:**

Vi xử lý 8085 có 5 ngắt cứng như sau: TRAP, RST7.5, RST6.5, RST5.5, và INTR.

Khi một chân ngắt bất kỳ (ngoại trừ INTR) tích cực, mạch điều khiển bên trong sẽ thực hiện lệnh gọi CALL đến vùng nhớ định trước. Vùng bộ nhớ này được gọi là bảng vecto ngắt và các ngắt này gọi là vecto ngắt.

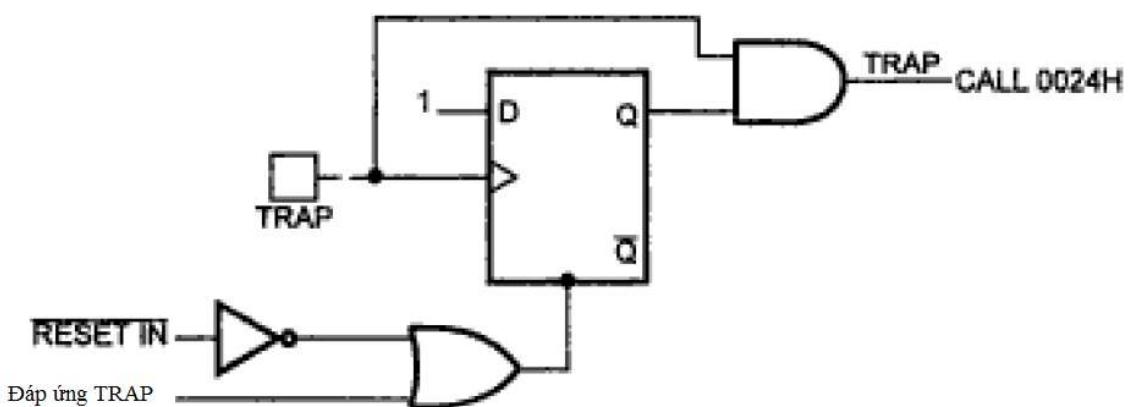


Hình 3.12 Cấu trúc ngắt vi xử lý 8085

Riêng ngắt INTR không là dạng vecto ngắt. thiết bị bên ngoài sẽ gửi địa chỉ cần gọi.

Với vi xử lý 8085, tất cả các ngắt (ngoại trừ TRAP) đều bị che (maskable). Tức là bình thường sẽ không hoạt động, khi cần hoạt động 1 ngắt nào đó, cần cho phép ngắt đó tích cực bằng lệnh từ vi xử lý. Như vậy từ vi xử lý có thể cấm ngắt hoặc ho phép ngắt hoạt động. riêng ngắt TRAP không bị che có nghĩa là nó luôn ở chế độ sẵn sang hoạt động. hình dưới mô tả cấu trúc của các ngắt vi xử lý 8085, trong hình này có thể thấy tác động ngắt có thể xảy ra khi thay đổi trạng thái, sùn xung hoặc cả hai của cổng vào ngắt tương ứng.

TRAP: như đã nói ở trên ngắt TRAP là ngắt không bị che. Nó hoạt động ở chế độ ưu tiên cao nhất. Dạng ngắt TRAP có thể là sùn xung và mức trạng thái. Điều này có nghĩa là để chế độ ngắt hoạt động, chân TRAP cần chuyển trạng thái từ mức thấp lên mức cao và giữ ở mức cao cho đến khi thực hiện đáp ứng ngắt. hoạt động theo nguyên tắc như vậy có phép loại trừ các yếu tố nhiễu.



Hình 3.13 Mạch điều khiển ngắt TRAP

Như trên hình 3.13, thể hiện việc điều khiển xung đầu vào của tín hiệu ngắt TRAP. Một sùn dương của tín hiệu TRAP sẽ set D-flip-flop. Thông qua cổng AND để kiểm tra trạng thái mức cao của đầu vào sẽ đưa ra mức 1 của cổng đầu ra “đòi hỏi ngắt lỗi gọi hàm ở địa chỉ 002HH”. Có 2 cách để xóa tín hiệu đầu ra:

- Tín hiệu reset của vi xử lý Reset-in
- Tín hiệu mức cao báo đã bắt đầu thực hiện đáp ứng ngắt.

Khi có 1 đòi hỏi ngắt TRAP, vi xử lý thực hiện nốt lệnh hiện tại, sau đó nó bắt giữ địa chỉ câu lệnh tiếp theo (chính là địa chỉ sẽ trả về khi thực hiện xong ngắt) vào trong ngăn xếp, tiếp đến sẽ nạp vào bộ nhớ chương trình địa chỉ 0024H. như vậy vi xử lý thực hiện hàm đáp ứng ngắt tại địa chỉ 002HH.

RST7.5.: ngắt RST7.5 là ngắt bị che, nó có mức độ ưu tiên thứ 2. Như hình vẽ trên, nó hoạt động khi có 1 sùm xung dương, lúc này sẽ set flip-flop D, việc xóa flip-flop D được thực hiện nhờ lệnh SIM của vi xử lý hoặc tín hiệu đã bắt đầu thực hiện đáp ứng ngắt.

Khi có tín hiệu sùm lên vào chân RST7.5 và bit che M7.5 ở mức 0 (tức là ngắt RST7.5 không bị che) thì vi xử lý sẽ thực hiện nốt lệnh hiện tại. sau đó sẽ cấp giữ địa chỉ tiếp theo và trong ngăn xếp và nạp con trỏ chương trình và vecto địa chỉ cố định là 003CH. Như vậy vi xử lý thực hiện hàm đáp ứng ngắt tại địa chỉ 003CH. Địa chỉ này gọi là địa chỉ thực hiện của ngắt RST7.5, RST6.5 và RST5.5. mức ngắt RST6.5 và RST5.5 là ngắt dạng mức. Các ngắt này có thể bị che khi sử dụng lệnh SIM. Ngắt RST6.5 có mức độ ưu tiên thứ ba và RST5.5 có mức độ thứ tư. Địa chỉ vecto tương ứng của RST6.5 và RST5.5 là 003HH và 002CH. Sauk khi có đòi hỏi ngắt, vi xử lý 8085 sẽ thực hiện nốt lệnh hiện tại cất giữ địa chỉ lệnh tiếp theo vào ngăn xếp và nạp con trỏ chương trình 1 địa chỉ của vecto ngắt tương ứng.

INTR. Ngắt này à ngắt bị che, nhưng không phải là ngắt vecto.nó có mức ưu tiên thấp nhất. khi chân INTR ở mức cao vi xử lý sẽ thực hiện 1 chuỗi sự kiện sau:

- Vi xử lý kiểm tra trạng thái cầu tín hiệu INTR trong quá trình thực hiện mỗi lệnh.
- Nếu chân tín hiệu INTR ở mức cao và ngắt được cho phép hoạt động , vi xử lý 8085 sẽ thực hiện nốt lệnh hiện tại và giữ tín hiệu đáp ứng ngắt ở mức tích cực thấp (INTA)
- Khi có tín hiệu INTA, mạch logic sẽ đưa lệnh OPCODE lên Bus dữ liệu. nếu đây là lệnh nhiều byte, các byte sẽ lần lượt được đưa lệnh Bus.
- Khi nhận được lệnh, vi xử lý lưu địa chỉ của lệnh kế tiếp vào trong stack và thực hiện lệnh nhận được.

Chú ý: về lý thuyết, mạch logic

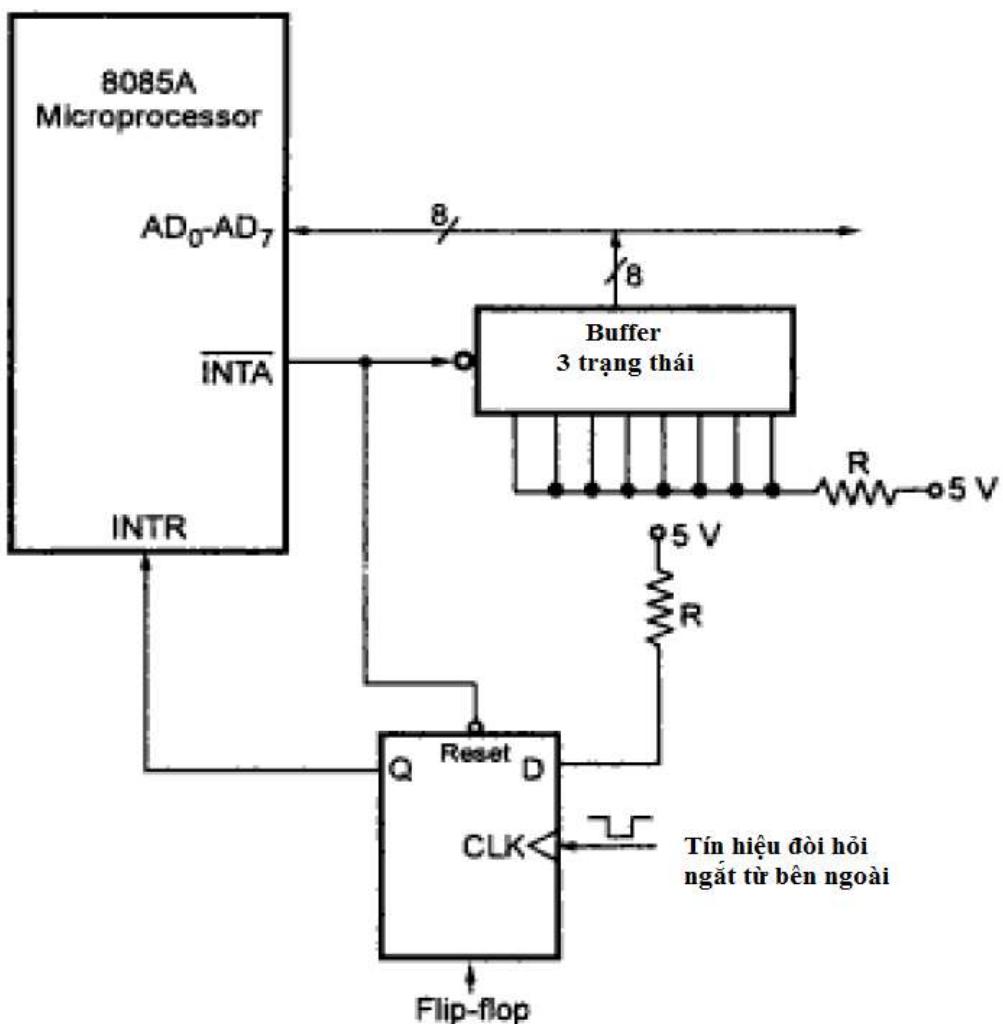
Có thể đặt bất kỳ lệnh nào lên bus dữ liệu. tuy nhiên chỉ có lệnh CALL và RST mới ghi nội dung của bộ đếm chương trình vào ngăn xếp, đồng thời nó sẽ chuyển chương trình điều khiển đếm địa chỉ vecto ngắt tương ứng với lệnh ngắt.

Đáp ứng với lệnh CALL: nêu thiết bị ngoại vi có yêu cầu thực hiện lệnh CALL thì vi xử lý sẽ tạo ra trình tự thực hiện như sau:

- Nó gửi 1 tín hiệu đáp ứng ngắt mức thấp lần thứ hai.

- Để trả lời đáp ứng này, mạch logic ngoài sẽ giữ địa chỉ byte thấp của lệnh gọi CALL.
 - Sau khi nhận được byte địa chỉ thấp, vi xử lý giữ tín hiệu đáp ứng ngắn mức thấp lần thứ ba.
 - Để trả lời đáp ứng này, mạch logic ngoài sẽ giữ địa chỉ byte cao của lệnh gọi CALL.
 - Sau khi nhận được 16 bit địa chỉ của lệnh CALL, vi xử lý 8085 cất giữ nội dung của bộ đếm chương trình vào ngăn xếp, và thực hiện hàm nằm ở địa chỉ mà thiết bị ngoại vi gửi.

Ví dụ, trong hình dưới là sơ đồ của mạch logic ngoài gửi lệnh RST7.



Hình 3.14 Sơ đồ logic điều khiển ngắt RST7

Mạch logic ngoài điều khiển buffer ba trạng thái (tri-state) với tín hiệu INTA từ vi xử lý. Tín hiệu này là tín hiệu cho phép đầu ra tích cực (output enable), đồng thời để

xóa D flip-flop. Trước tiên sẽ có 1 ngắn vẫn thực hiện ngắt tewf thiết bị bên ngoài (I/O device) thông qua flip-flop D để giữ tín hiệu INTR ở mức cao cho đến khi vi xử lý 8085 giữ tín hiệu đã đáp ứng ngắt. chân tín hiệu INTA đưa 1 tín hiệu cho phép truyền tín hiệu từ đầu vào ra đầu ra của buffer. Như hình trên, trên buffer sẽ có giá trị FCH tương ứng với lệnh RST7. Vì xử lý 8085 sẽ nhận lệnh này trong chu kỳ đáp ứng ngắt. sau khi nhận được lệnh, vi xử lý cắt giữ nội dung của bộ đếm chương trình PC vào ngắn xếp. sau đó, chương trình điều khiển sẽ chuyển về thực hiện ở địa chỉ 0038H (địa chỉ câu vecto ngắt RST7). Bảng dưới thể hiện các đặc tính của ngắt cùng vi xử lý 8085.

Dạng ngắt	Kiểu hoạt động	Mức độ ưu tiên	Che ngắt	Vecto địa chỉ
TRAP	Sùn và mức	1 (cao nhất)	Không	002HH
RST7.5	Sùn	2	Có	003CH
RST6.5	Mức	3	Có	003HH
RST5.5	Mức	4	Có	002CH
INTR	Mức	5 (thấp nhất)	Có	-

• Ngắt mềm của 8085:

Vì xử lý 8085 có 8 ngắt mềm từ RST0 đến RST7. Địa chỉ vector ngắt được tính như sau:

Vector address = số hiệu ngắt x 8

Ví dụ, địa chỉ vector ngắt của RST5 là $5 \times 8 = 40 = 28H$

Bảng dưới thể hiện địa chỉ vector ngắt mềm.

Lệnh	Mã Hex	Vector địa chỉ
RST 0	C7	0000H
RST 1	CF	0008H
RST 2	D7	0010H
RST 3	DF	0018H
RST 4	E7	0020H
RST 5	EF	0028H
RST 6	F7	0030H
RST 7	FF	0038H

4.3.2. Lệnh che và không che ngắt (masking/ unmasking of interrupt).

Việc che ngắt được cho phép (enable) hoặc cấm (disable) nhờ lệnh trong chương trình (lệnh EI, DI và SIM).

- EI (enable interrupt) – cho phép ngắt

Như hình 2.11 lệnh EI sẽ set flip-flop D để đưa mức cao vào các cổng AND của ngắt RST 7.5, RST 6.5, RST 5.5 và INTR. Cần nhớ rằng khi 1 ngắt bất kỳ có đáp ứng, thì tín hiệu cho phép ngắt sẽ bị xóa nhò tín hiệu reset flip-flop D. để cho phép ngắt trở lại cần thực hiện lại lệnh EI.

- DI (disable interrupt) - Cấm ngắt

Lệnh DI xóa trạng thái cho phép ngắt của flip-flop D ở trên hình 2.11, nó hiệu lực với các ngắt RST 7.5, RST 6.5, RST 5.5 và INTR.

- SIM (set interrupt mask) – thiết lập chế độ ngắt

Lệnh này dùng để thiết lập chế độ che ngắt và truyền ra cổng nối tiếp. nó truyền nội dung của thanh ghi dùng chung (ACC) tới mạch điều khiển ngắt và cổng vào/ra nối tiếp. như vậy cần nạp vào thanh ghi dùng chung ACC nội dung phù hợp trước khi thực hiện lệnh SIM.

4.5 BUS I/O, bộ nhớ và hệ thống

Chúng ta đã biết khối xử lý trung tâm (CPU), bộ nhớ và khối vào ra là các module phần mềm của vi xử lý. Khi làm việc chúng cần có mối liên hệ với nhau thông qua 1 kênh truyền dữ liệu. tập hợp các thành phần tạo mối liên kết này được gọi là cấu trúc kết nối bên trong. Việc thiết kế cấu trúc kết nối này phụ thuộc nội dung và phương thức trao đổi thông tin giữa các module. Các đường dây nối để truyền dữ liệu giữa các module được gọi là BUS. Đường BUS kết nối các thành phần chính bên trong vi xử lý (CPU, bộ nhớ, cổng vào/ra) được gọi là BUS hệ thống).

BUS hệ thống được chia thành 3 nhóm chức năng sau:

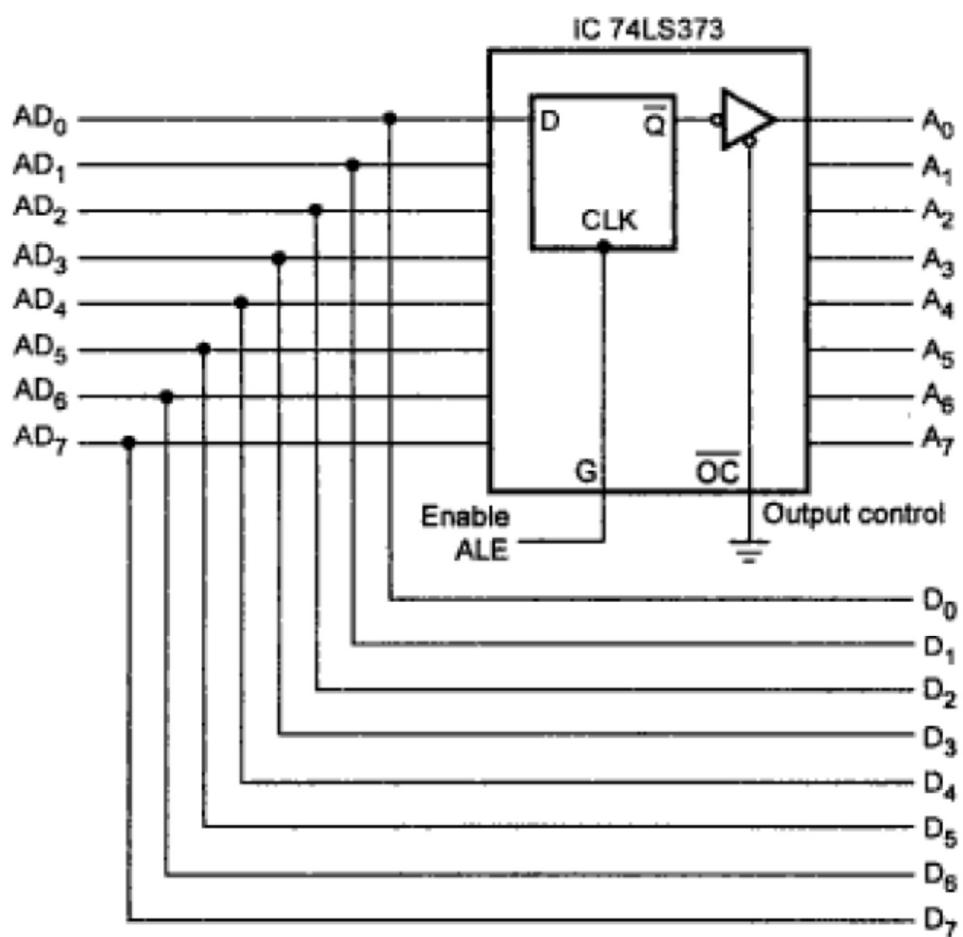
- BUS dữ liệu
- BUS địa chỉ
- BUS điều khiển

Vì xử lý 8085 có các đường BUS AD0 – AD15 là Bus địa chỉ và dữ liệu, các chân ID/M, RD và WR là đường BUS điều khiển.

4.5.1 BUS địa chỉ và BUS dữ liệu (tách kênh các đường AD7-AD0).

Như chúng ta đã biết các đường AD0 đến AD7 được dồn kênh (multiplexing) các địa chỉ byte thấp từ A0 đến A7 hoạt động chỉ trong thời gian T1 của chu kỳ máy để truy nhập vào vùng nhớ hoặc địa chỉ vào ra (I/O). Điều này có nghĩa là địa chỉ byte thấp cần được chốt (giữ lại) cho các thời điểm sau của chu kỳ máy. Việc chốt các dữ liệu của địa chỉ byte thấp được thực hiện nhờ tín hiệu ALE từ vi xử lý 8085.

Hình dưới thể hiện mạch phần cứng để chốt địa chỉ byte thấp sử dụng IC 7MLS373. Dữ liệu từ cổng input được truyền ra output khi chân clock ở mức cao. Tín hiệu clock này được điều khiển bởi tín hiệu ALE từ 8085.



Hình 3.15 Tách kênh BUS địa chỉ và dữ liệu

Tín hiệu ALE được tích cực chỉ trong thời gian T1, như vậy trong khoảng thời gian này tại cổng output sẽ có mức tín hiệu giống như input. Sau khoảng thời gian T1, tín hiệu ALE không tích cực, lúc này cổng đầu ra sẽ được chốt dữ liệu tại thời điểm tín hiệu ALE từ mức cao chuyển xuống mức thấp. tức là cổng đầu ra của 7HLS373 không

thay đổi trạng thái ngay cả khi đầu vào thay đổi. quá trình trên là quá trình tách lệnh địa chỉ (A0-A7) với kênh dữ liệu (D0-D7).

Các đường địa chỉ từ A8 – A5 không phải là bộ dồn kênh, do đó có thể đi trực tiếp đến chân địa chỉ của bộ nhớ hoặc cỗng vào ra.

4.5.2 Tín hiệu điều khiển

Vì xử lý 8085 có các chân RO và WR để khởi tạo chu kỳ đọc và ghi. Do các tín hiệu này được dùng chung cho việc ghi/ đọc bộ nhớ và thiết bị vào/ra bên ngoài, nên cần có mạch tách các tín hiệu ghi đọc cho bộ nhớ và thiết bị bên ngoài (I/O).

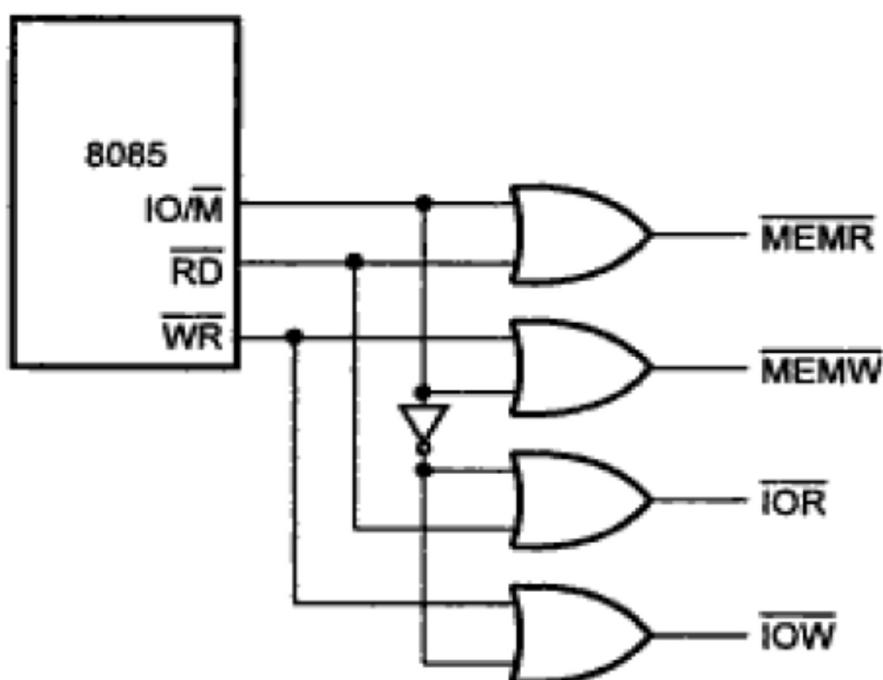
MEM R (Memory read)

MEM W (Memory write)

IOR (I/O read)

IOW (I/O write)

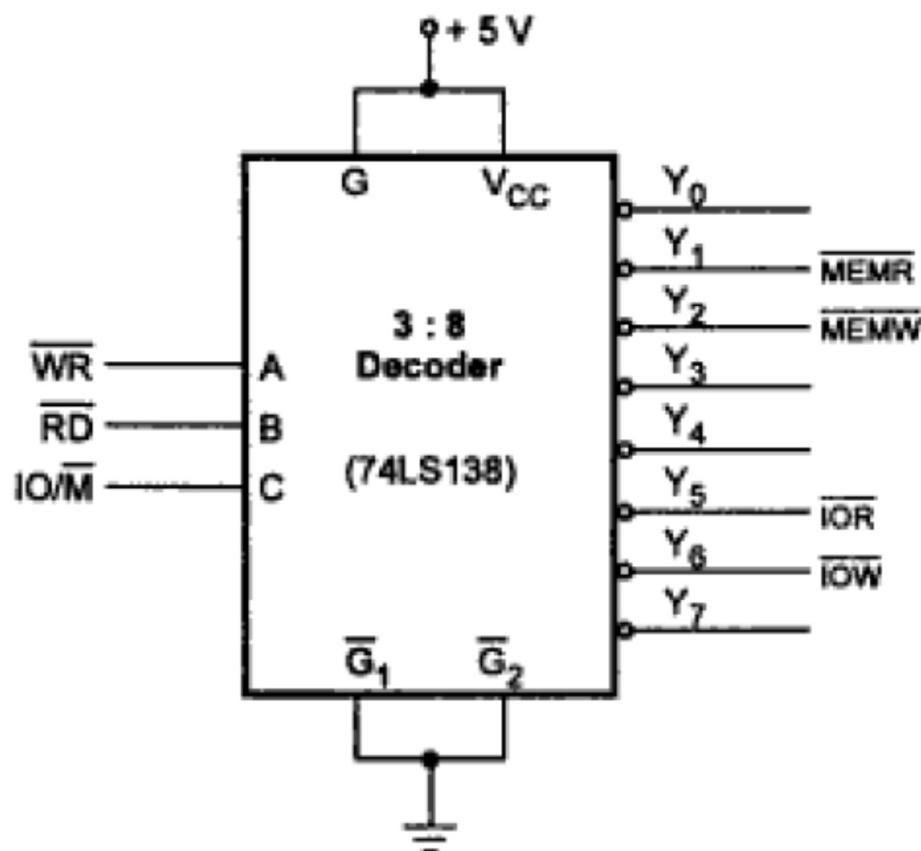
Có thể dùng mạch giải mã để tách các tín hiệu này.



Hình 3.16 Tách các tín hiệu điều khiển

IO/\bar{M}	\bar{RD}	\bar{WR}	\overline{MEMR} $\bar{RD}+IO/\bar{M}$	\overline{MEMR} $\bar{WR}+IO/\bar{M}$	\overline{IOR} $\bar{RD}+IO/\bar{M}$	\overline{IOW} $\bar{WR}+IO/\bar{M}$
0	0	0	Điều kiện này không bao giờ xảy ra do ghi và đọc không đồng thời			
0	0	1	0	1	1	1
0	1	0	1	0	1	1

0	1	1	1	1	1	1	1
1	0	0	Điều kiện này không bao giờ xảy ra do ghi và đọc không đồng thời				
1	0	1	1	1	0	1	
1	1	0	1	1	1	0	
1	1	1	1	1	1	1	

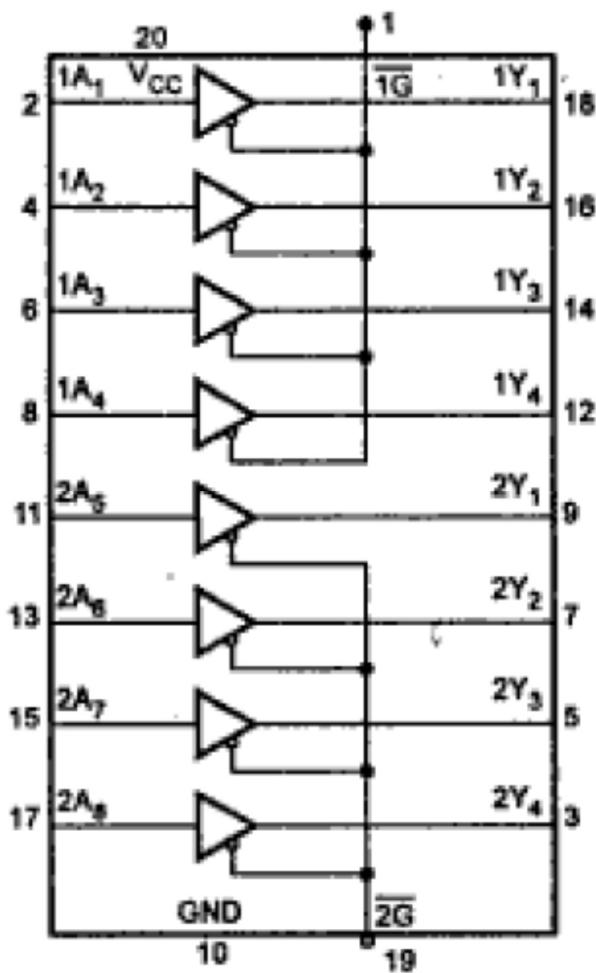


Hình 3.17 Sử dụng mạch giải mã tách các tín hiệu điều khiển

4.5.3 Điều khiển BUS (BUS driver)

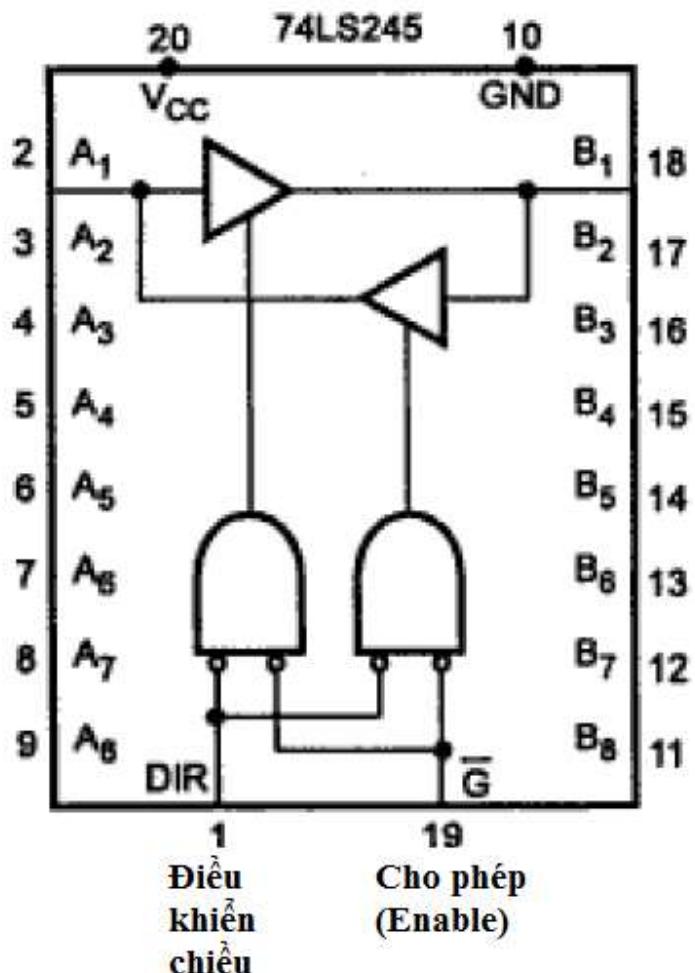
Thông thường BUS của vi xử lý 8085 có thể cấp nguồn ra với dòng H00WA và nhận dòng 2mA, tức là chỉ có thể điều khiển các cổng logic TTL. Nếu cần tăng dòng điều khiển (trường hợp truyền đi xa) thì cần thêm mạch điều khiển BUS (nâng công suất đầu ra).

- BUS 1 chiều: sử dụng IC 7HLS244 để driver BUS như hình dưới. mỗi cổng này có thể cung cấp dòng ra là 15 mA và dòng nhận là 24 mA. Các đường BUS được chia thành 2 nhóm để cho phép từng nhóm hoạt động sử dụng các chân điều khiển 1G và 2G.



Hình 3.17 Mạch điều khiển BUS 1 chiều

- BUS 2 chiều: lúc này sử dụng IC 7HLS245. Khác với IC 7425244 có thêm chân DIR quy định chiều hoạt động , khi DIR ở mức cao, dữ liệu dự trữ hướng A sang B và khi ở mức thấp dữ liệu đi từ B sang A. tương tự như vậy, dòng ra có thể tiếp đến 15 mA, còn dòng nhận là 24 mA.

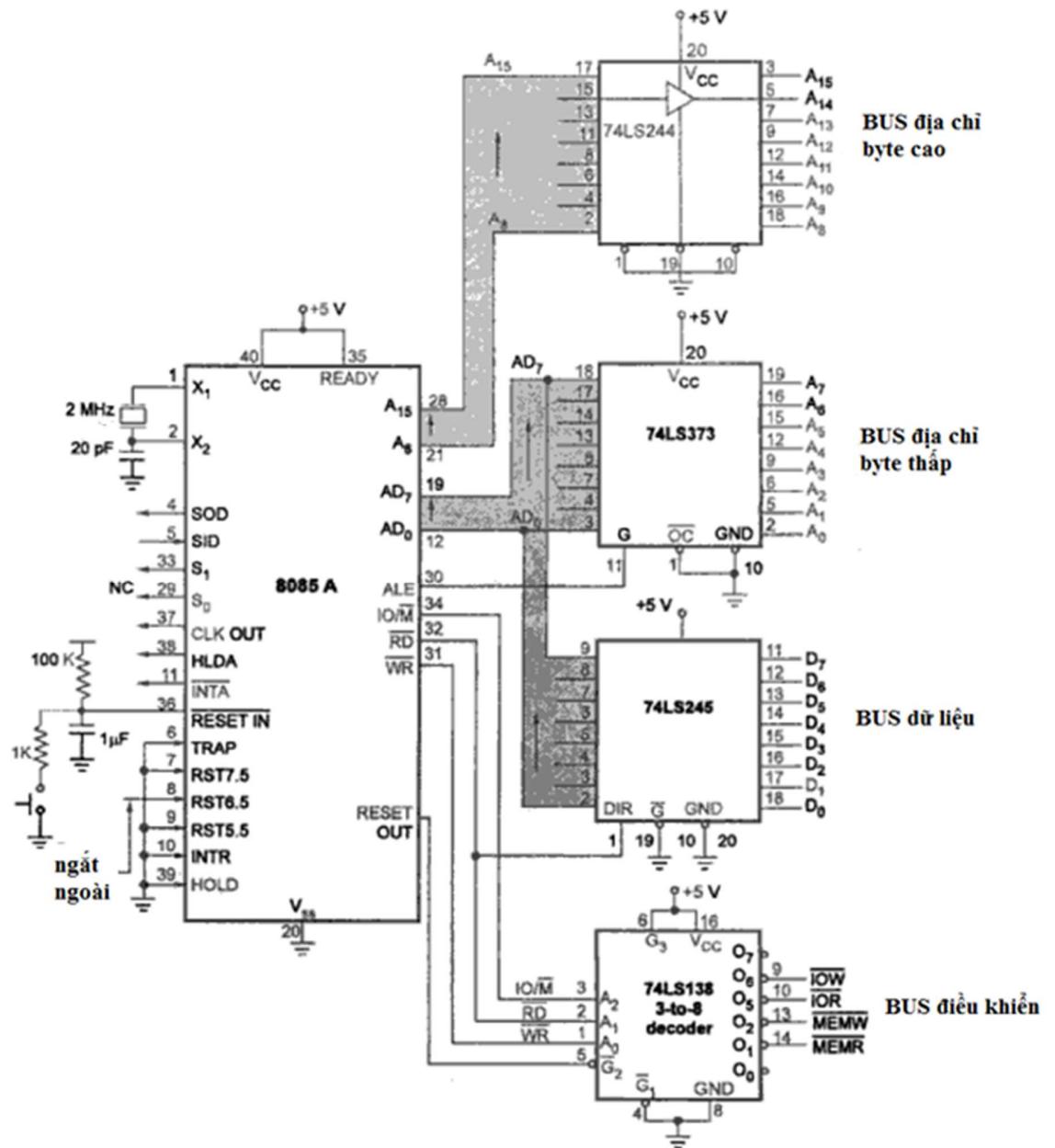


Hình 3.18 mạch điều khiển BUS 2 chiều

4.6 Sơ đồ ghép nối hệ thống.

Hình dưới là sơ đồ tách hình địa chỉ và các tín hiệu điều khiển, ngoài ra trong sơ đồ còn có mạch tạo dao động và reset. Đường tín hiệu ngắn không dùng do đó được nối đất, mục tiêu việc nối đất là chống nhiễu. đường điều khiển DMA không dùng, đường HOLD cũng được nối đất. Như ta đã biết tín hiệu READY dùng để đồng bộ với thiết bị ngoại vi hoạt động chậm hơn vi xử lý. Khi nó ở mức thấp, vi xử lý cần thực hiện chế độ đợi (wait state) và khi ở mức cao thì báo hiệu là bộ nhớ hoặc thiết bị ngoại vi sẵn sang truyền hoặc nhận dữ liệu. tín hiệu ALE để tách kênh địa chỉ byte thấp ở thời gian T1 của chu kỳ máy. Để điều khiển BUS 2 chiều sử dụng IC 7HLS2H5, chân RD của 8085 được nối với chân DIR của IC này. Như vậy khi RD ở mức thấp, chân DIR sẽ điều khiển BUS theo luồng đọc vào từ bộ nhớ trong thiết bị ngoại vi. Nó phù hợp với lệnh

đọc. Khi RD ở mức cao, DIR ở mức cao và điều khiển luồng ghi ra bộ nhớ hoặc thiết bị ngoại vi.

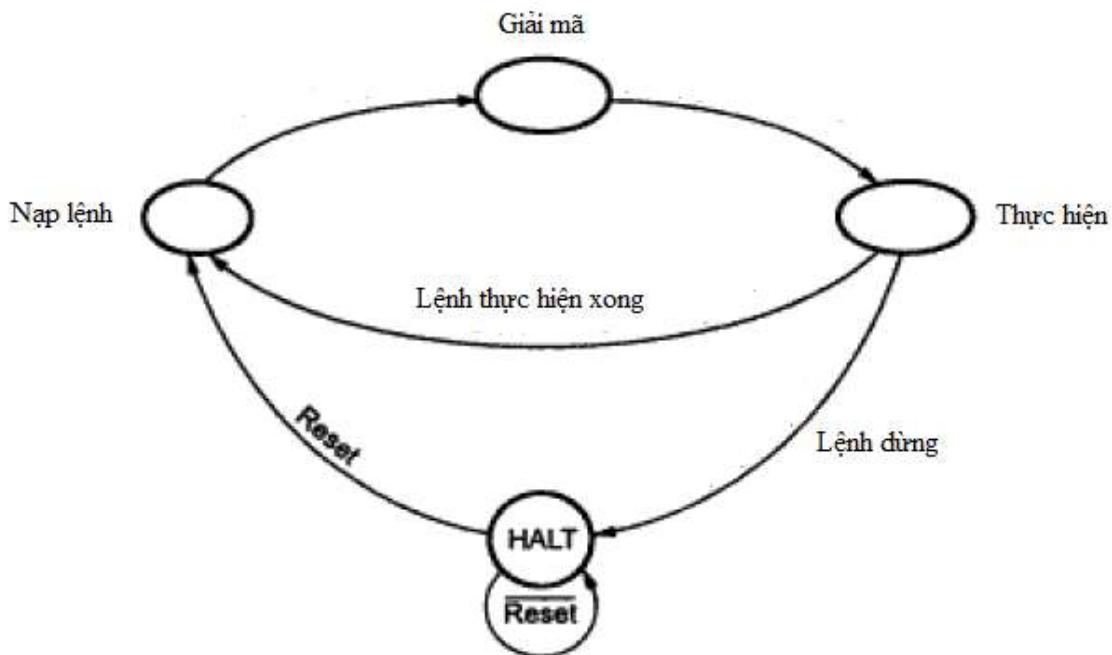


Hình 3.19 Sơ đồ ghép nối hệ thống

V.Chu kỳ lệnh

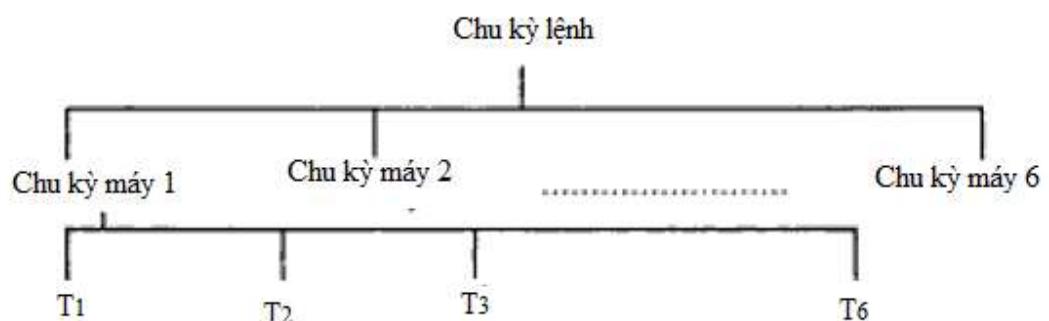
Khi thực hiện lệnh thông thường, vi xử lý tiến hành tuần tự các bước sau: nạp lệnh, giải mã lệnh và thực hiện, sau đó lặp lại cho đến khi thực hiện lệnh tạm dừng (Halt). Khi lệnh Halt được thực hiện vi xử lý chuyển vào trạng thái tạm dừng, để thoát khỏi trạng thái này cần tín hiệu reset (khởi động lại). Như vậy sau khi có tín hiệu reset vi

xử lý lại bắt đầu nạp lệnh., giải mã và thực hiện. quá trình này gọi là chu kỳ lệnh. Hình dưới thể hiện chu kỳ lệnh.



Hình 3.20 Chu kỳ lệnh.

Khi thực hiện chu kỳ lệnh, vi xử lý có thể thực hiện từ 1 đến 5 thao tác đọc hoặc ghi bộ nhớ trong thiết bị ngoại vi. Mỗi lần thao tác với bộ nhớ hay thiết bị ngoại vi cần 1 chu kỳ thời gian, gọi là chu kỳ máy. Hay nói một cách khác, để chuyển 1 byte dữ liệu vào/ra từ vi xử lý cần 1 chu kỳ máy. Mỗi 1 chu kỳ máy chiếm khoảng từ 3 đến 6 xung nhịp đồng hồ, được gọi là nhịp thời gian T (T-state). Như vậy ta thấy rằng, 1 chu kỳ lệnh chứa từ 1 đến 5 chu kỳ máy và 1 chu kỳ máy chứa từ 3 đến 6 nhịp thời gian T. Như hình dưới:



Hình 3.21 Quan hệ giữa chu kỳ lệnh, chu kỳ máy và trạng thái T

Với vi xử lý 8085 có 7 dạng khác nhau của chu kỳ máy. Các trạng thái của các tín hiệu IO/M, S₀, S₁ được thể hiện ở bảng dưới. Các tín hiệu này được tạo ra tại 1 thời điểm đầu của chu kỳ máy và không thay đổi trong quá trình của chu kỳ này.

Chu kỳ máy	Tín hiệu trạng thái			Tín hiệu điều khiển		
	IO/M	S ₁	S ₂	RD	WR	INTA
Nạp lệnh	0	1	1	0	1	1
Đọc bộ nhớ	0	1	0	0	1	1
Ghi bộ nhớ	0	0	1	1	0	1
Đọc I/O	1	1	0	0	1	1
Ghi I/O	1	0	1	1	0	1
Đáp ứng INTR	1	1	1	1	1	0
BUS nghỉ	0	0	0	1	1	1

V. Lệnh, thực hiện lệnh, thực hiện tuần tự và dòng dữ liệu.

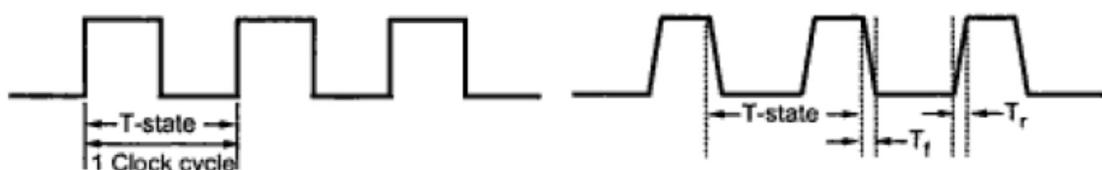
Để thực hiện lệnh, vi xử lý tiến hành tuần tự các chu kỳ máy. Trong phần này ta xem xét phương pháp mô tả tín hiệu, giảm chö thời gian và việc tiến hành tuần tự chu kỳ máy, đồng thời là dòng dữ liệu vào/ra vi xử lý.

5.1 Mô tả tín hiệu:

Để mô tả tín hiệu ta xem xét giản đồ thời gian của nó.

- Tín hiệu xung nhịp: clock signal

Vi xử lý 8085 chia xung đồng hồ dao động từ các chân x1 và x2 với hệ số 2. Tần số sau khi chia được gọi là tần số hoạt động này. Trong điều kiện lý tưởng, xung đồng hồ là xung vuông với độ trùm sùm xung lên và xuống bằng không. Nhưng trong thực tế độ trễ này luôn tồn tại như hình dưới.

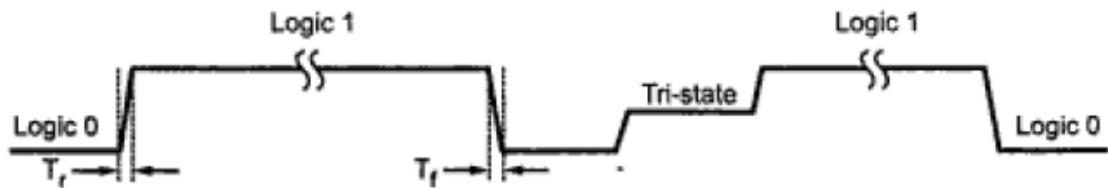


Hình 3.22 Tín hiệu xung nhịp

- Tín hiệu đơn

Tín hiệu đơn có thể có các mức logic 0, 1 hoặc trạng thái trở kháng cao (tri-state). Khi chuyển trạng thái cùng xuất hiện độ trễ thời gian chuyển mức ở sùm lên và sùm

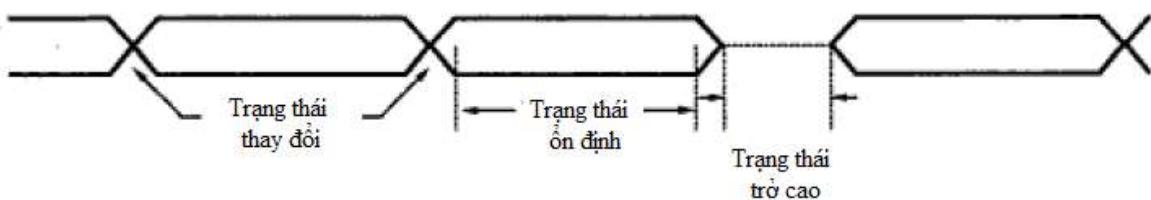
xuống.



Hình 3.23 Tín hiệu đơn.

- Nhóm tín hiệu

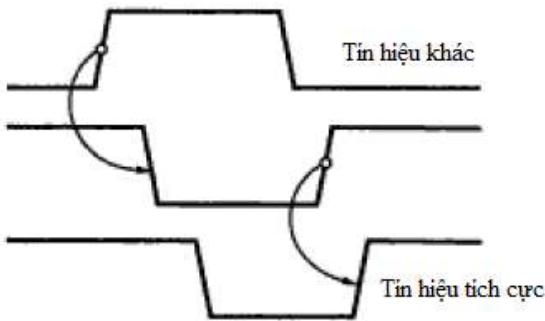
Nhóm tín hiệu, còn được gọi là Bus gồm Bus dữ liệu và Bus ddc. Giản đồ thời gian được nhóm lại thể hiện ở hình dưới. tín hiệu được ghép nhóm không thể hiện ở mức logic của từng tín hiệu, tuy nhiên thể hiện thời điểm các tín hiệu đó ở trạng thái ổn định và đang chuyển trạng thái (phản gạch chéo). Trạng thái trở kháng cao được thể hiện bằng đường nét đứt.



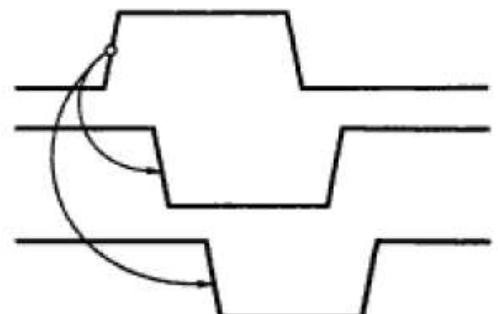
Hình 3.24 Nhóm tín hiệu

Trong hệ thống vi xử lý, tính tích cực của tín hiệu phụ thuộc vào sự thay đổi trạng thái của tín hiệu khác. Có 4 khả năng xảy ra như sau:

- Tính tích cực của 1 tín hiệu phụ thuộc sự thay đổi trạng thái của 1 tín hiệu khác.
- Tính tích cực của 1 tín hiệu phụ thuộc sự thay đổi của nhiều tín hiệu khác
- Tính tích cực của nhiều tín hiệu phụ thuộc sự thay đổi của 1 tín hiệu
- Tính tích cực của nhiều tín hiệu phụ thuộc sự thay đổi của nhiều tín hiệu .



Hình 3.25 a. Tính tích cực của 1 tín hiệu



Hình 3.25 b. Tính tích cực của nhiều

phụ thuộc trạng thái của 1 tín hiệu khác

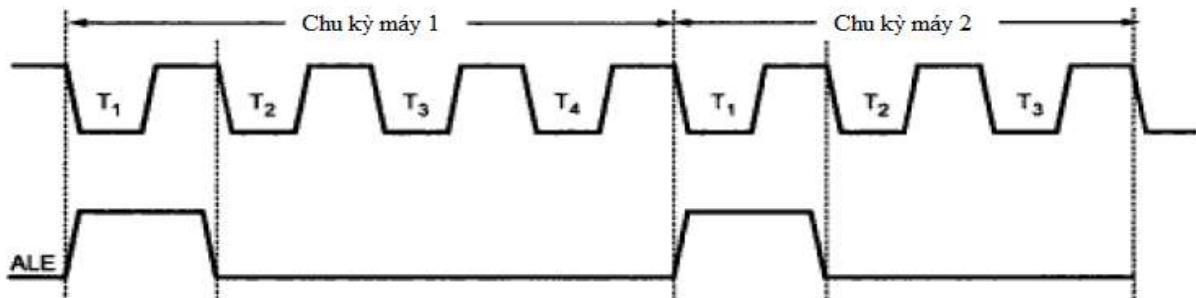


Hình 3.26 a. Tính tích cực của 1 tín hiệu phụ thuộc trạng thái của nhiều tín hiệu khác

5.2 Giản đồ thời gian của 1 số tín hiệu:

- Tín hiệu ALE

Tín hiệu này tích cực ở mức cao. nó được tích cực tại thời điểm đầu T1 của chu kỳ máy như hình dưới.

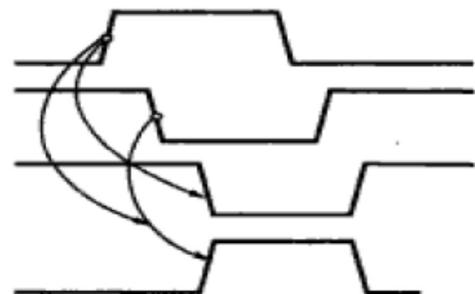


Hình 3.27 Tín hiệu ALE

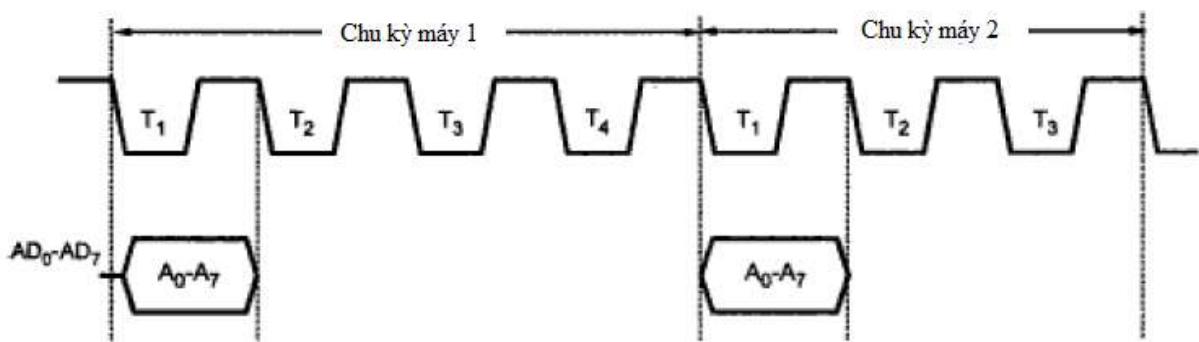
- Ao – A7 (địa chỉ byte thấp)

Địa chỉ byte thấp được đưa ra bộ tách kênh địa chỉ (dữ liệu AD0 – AD7) trong khoảng thời gian T1 như hình dưới.

tín hiệu phụ thuộc trạng thái của 1 tín hiệu khác



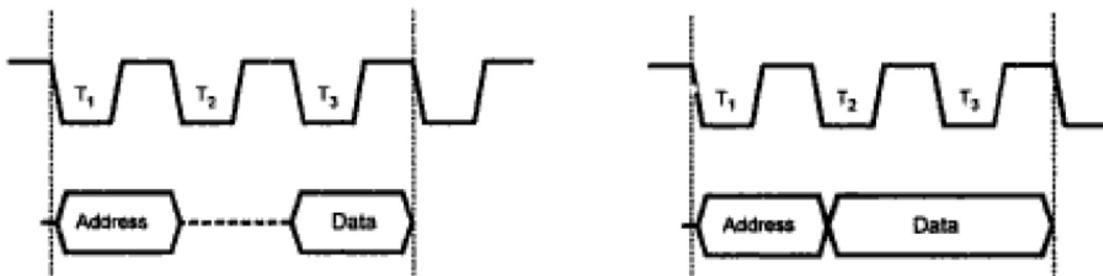
Hình 3.26b. Tính tích cực của nhiều tín hiệu phụ thuộc trạng thái của nhiều tín hiệu khác,



Hình 3.28 Tín hiệu BUS địa chỉ byte thấp

- Bus dữ liệu (data BUS)

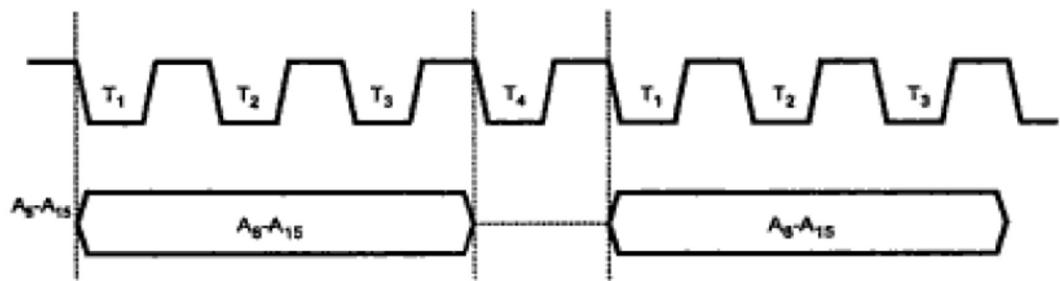
Dữ liệu từ bộ nhớ hoặc thiết bị ngoại vi đến vi xử lý được đưa ra trong khoảng thời gian T2 và T3. Một chú ý quan trọng là trong chu kỳ đọc, dữ liệu sẽ được đưa ra vào phần cuối của khoảng thời gian T2, còn trong chu kỳ ghi dữ liệu được đưa ra vào phần đầu của khoảng thời gian T2 như hình dưới.



Hình 3.29 BUS dữ liệu

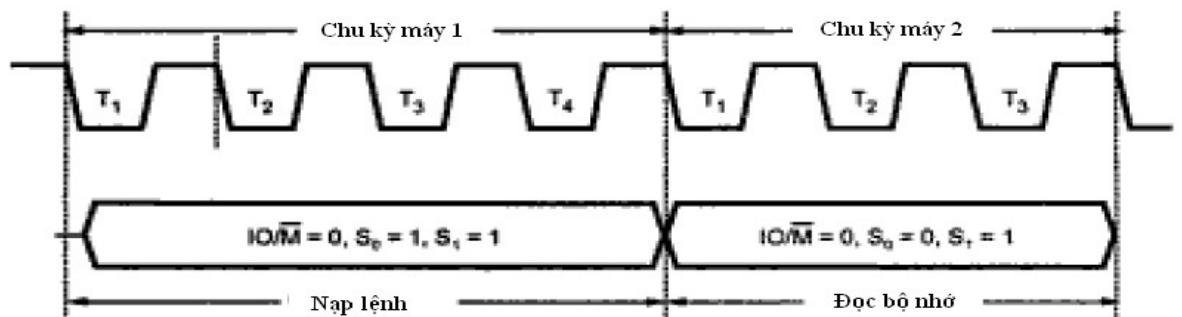
Để đọc dữ liệu từ bộ nhớ hoặc thiết bị ngoại vi, trước tiên cần lựa chọn bộ nhớ hoặc thiết bị ngoại vi, trước tiên cần lựa chọn bộ nhớ hoặc thiết bị ngoại vi. Sau khi được lựa chọn, thiết bị sẽ đưa dữ liệu từ vùng lựa chọn lên Bus dữ liệu. thao tác này cần 1 khoảng thời gian và được gọi là thời gian truy nhập (access time). Trong khi đó với chu kỳ ghi, dữ liệu đã sẵn sàng ở thanh ghi cẩu vi xử lý và việc đưa nó ra Bus dữ liệu có thời gian truy nhập bằng 0.

- Địa chỉ byte cao: các địa chỉ byte cao từ A8 – A15 được đưa ra trong khoảng thời gian từ T1 đến T3 của chu kỳ máy.



Hình 3.30 Địa chỉ byte cao

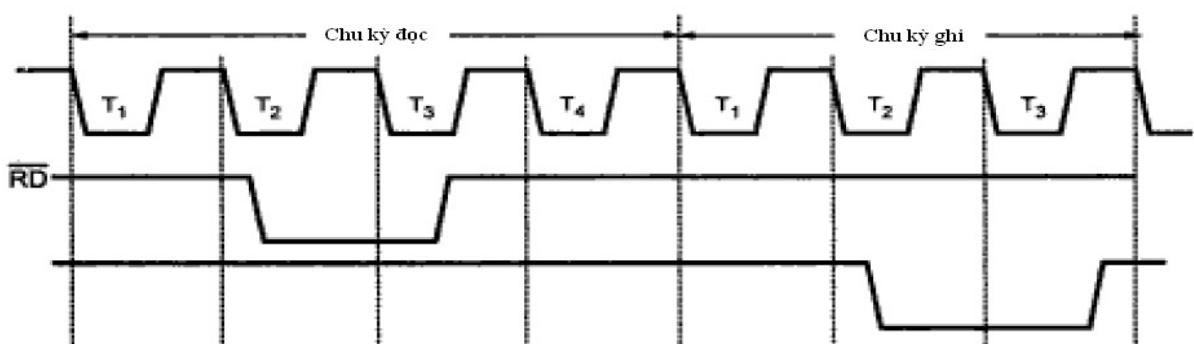
- IO/M, So, S1: các tín hiệu này được gọi là tín hiệu trạng thái. Nó quyết định dạng chu kỳ máy sẽ thực hiện. chúng được tích cực tại thời điểm T1 của chu kỳ máy và kéo dài đến khi kết thúc chu kỳ máy.



Hình 3.31 Tín hiệu trạng thái

- RD và WR: các tín hiệu này quyết định chiều của quá trình truyền dữ liệu. Khi RD tích cực, dữ liệu truyền từ bộ nhớ hoặc thiết bị ngoại vi tới vi xử lý, và khi WR tích cực, dữ liệu sẽ truyền từ vi xử lý đến bộ nhớ hoặc thiết bị ngoại vi. Lưu ý rằng hai tín hiệu này không bao giờ tích cực tại 1 thời điểm.

Giản đồ thời gian của các tín hiệu này thể hiện ở hình dưới.



Hình 3.32 Chu kỳ đọc ghi

5.2 Chu kỳ máy (machine cycles)

Vi xử lý 8085 có 7 dạng chu kỳ máy, chúng gồm:

- Nạp lệnh (opcode fetch)
- Đọc bộ nhớ (read memory)
- Ghi bộ nhớ (write memory)
- Đọc I/O (read I/O)
- Ghi I/O (write I/O)
- Đáp ứng ngắn
- Bus ở trạng thái không hoạt động (Bus idle)

Ta sẽ xem xét cụ thể từng dạng chu kỳ máy.

5.2.1 Chu kỳ nạp lệnh

Chu kỳ máy đầu tiên của quá trình thực hiện lệnh là nạp lệnh, trong chu kỳ này vi xử lý sẽ tìm đúng mã lệnh để thực hiện. Vi xử lý sẽ đưa nội dung câu bộ đếm chương trình lên Bus địa chỉ và thông qua quá trình đọc sẽ đọc mã lệnh của lệnh thực hiện. Trong hình 3.33 (a) thể hiện dòng dữ liệu từ bộ nhớ đến vi xử lý và hình 3.33 (b) là giản đồ thời gian của chu kỳ nạp lệnh.

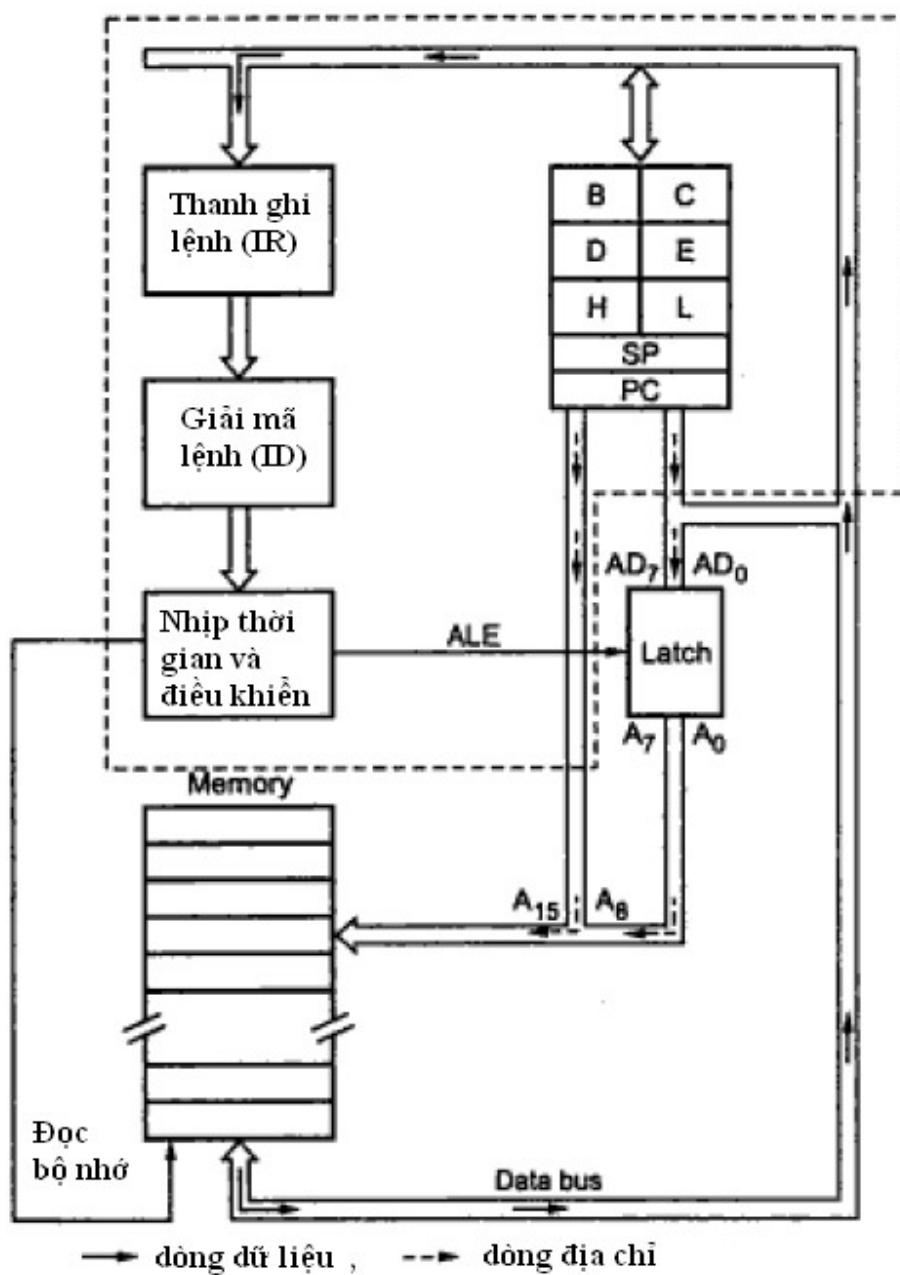
Chiều dài của chu kỳ này không cố định, nó nằm trong khoảng từ 4T đến 6T phụ thuộc vào loại câu lệnh. Quá trình nạp lệnh được thực hiện theo các bước sau:

Bước 1: (thời gian T1). Vào thời điểm T1, vi xử lý 8085 đưa nội dung của bộ đếm chương trình lên Bus địa chỉ. Các byte cao của bộ đếm chương trình PC được đưa ra địa chỉ từ A8 – A15. Tín hiệu ALE từ vi xử lý được đưa ra để chốt địa chỉ byte thấp.

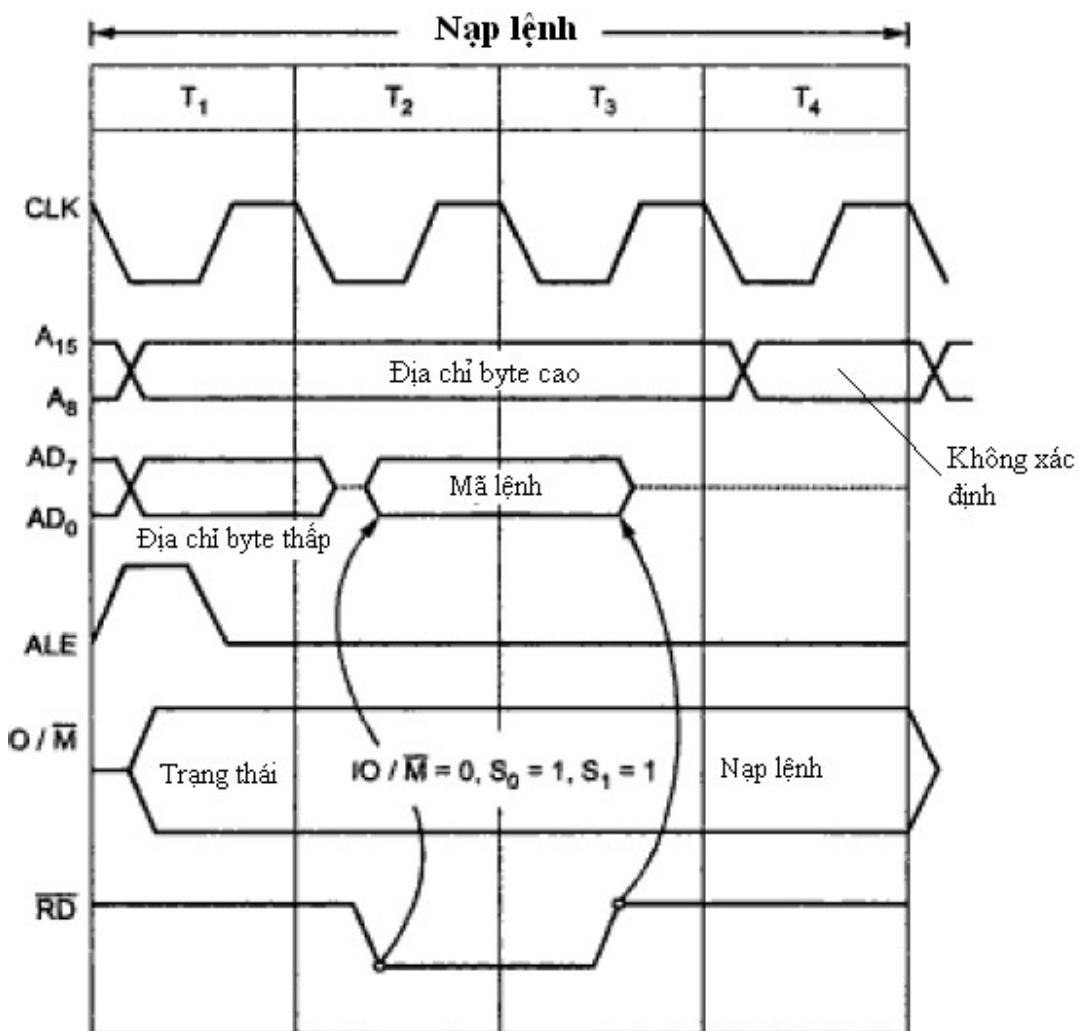
Tại thời điểm T1, vi xử lý đồng thời gửi tín hiệu trạng thái IO/M, S1 và S0. IO/M dùng để phân biệt việc chọn bộ nhớ hay thiết bị ngoại vi.

Trạng thái S1 để phân biệt đang đọc hay đang ghi. Phối hợp các trạng thái S0 và S1 cho ta biết hệ thống đang ở trạng thái nào: đọc, ghi, nạp lệnh hay ở chế độ dừng (Halt). Trong chu kỳ nạp lệnh thì IO/M = 0; S1=1; S0=1.

Bước 2 (thời điểm T2). Tại thời điểm T2, các địa chỉ byte được loại khỏi các chum Ado-AD7. Vi xử lý gửi tín hiệu RD có mức tích cực thấp để đọc dữ liệu từ bộ nhớ. Bộ nhớ sẽ đưa nội dung của vùng nhớ, tương ứng với địa chỉ áp vào từ chu kỳ trước, ra BUS dữ liệu (AD0-AD7).



Hình 3.33a Dòng dữ liệu từ bộ nhớ đến vi xử lý



Hình 3.33 b. Giản đồ thời gian của chu kỳ nạp lệnh

Bước 3 (thời điểm T3). Trong khoảng thời gian T3, vi xử lý nạp dữ liệu từ Bus dữ liệu vào thanh ghi lệnh và giữ tín hiệu RD ở mức cao để cấm sự truy nhập từ bộ nhớ.

Bước 4 (thời điểm T4). Tại thời điểm này, vi xử lý giải mã lệnh và trên cơ sở lệnh nhận được sẽ quyết định sử dụng khoảng thời gian T5 hay quay lại T1 tương ứng với lệnh kế tiếp.

Với lệnh 1 byte thực hiện với 8 bit dữ liệu thì quá trình dừng ở T4. Ví dụ các lệnh:

MOV A, B

ANA D

ADD B

INR L

DCR C

.....

Bước 5 (thời điểm T5 và T6). Trạng thái T5 và T6 được sử dụng trong các trường hợp đòi hỏi thêm thời gian thực hiện lệnh. Ví dụ lệnh 1 byte thực hiện với dữ liệu 16 bit sẽ sử dụng thêm trạng thái T5 và T6 như các lệnh DCX H, DCH L, SPHL, INXH...

5.2.2 Chu kỳ đọc bộ nhớ:

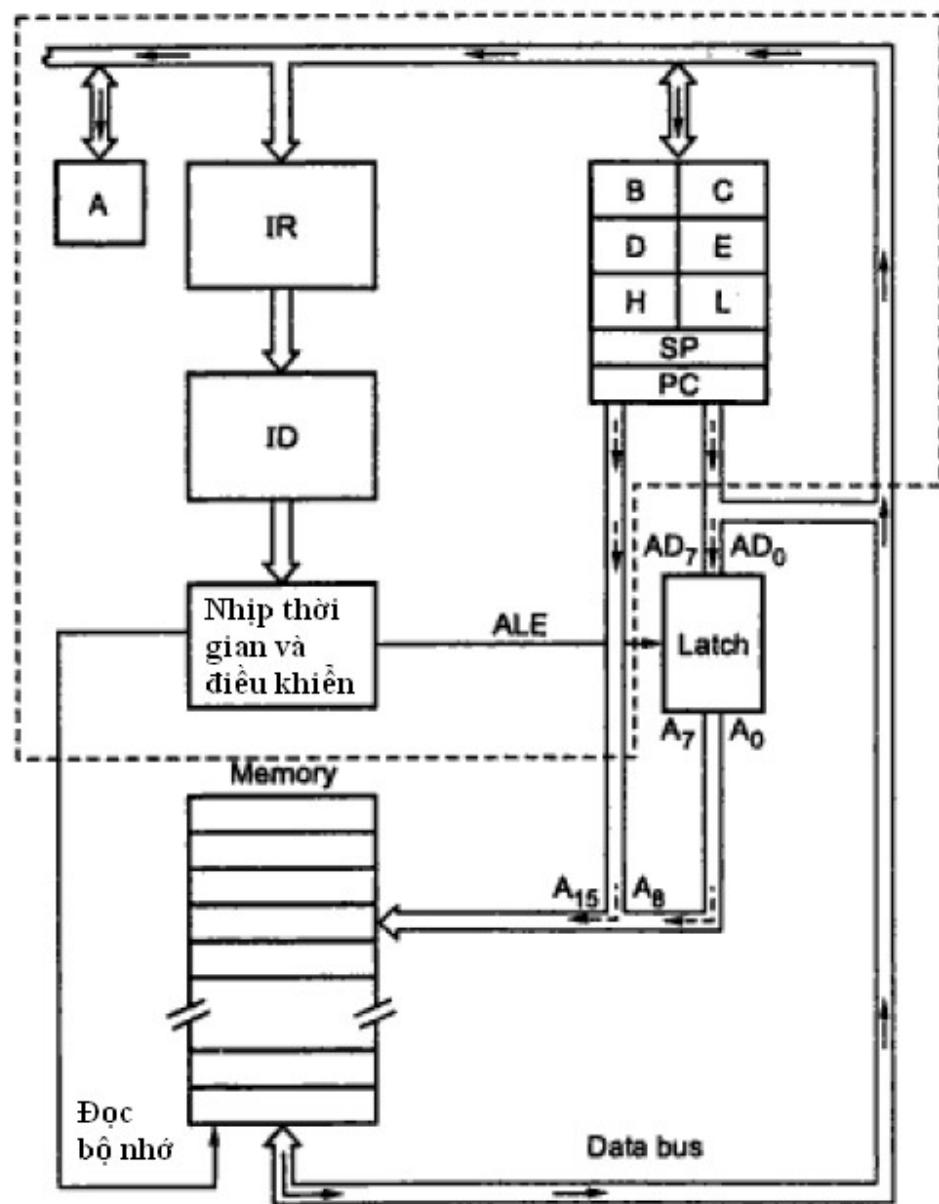
Vi xử lý thực hiện chu kỳ đọc bộ nhớ để đọc nội dung các RAM hoặc ROM. Chiều dài của chu kỳ máy là 3T ($T_1 \rightarrow T_3$). Trong chu kỳ máy này, vi xử lý lấy giá trị địa chỉ cần đọc từ con trỏ ngăn xếp từ vùng nhớ có địa chỉ cho trước đưa lên Bus địa chỉ. Thông qua quá trình đọc, đọc dữ liệu cặp thanh ghi dùng chung hoặc đếm bộ chương trình. Hình 3.34 (a) trình bày luồng thông tin từ bộ nhớ đến vi xử lý. Hình 3.34 (b) trình bày giản đồ thời gian của chu kỳ máy đọc bộ nhớ. Chu kỳ máy đọc bộ nhớ tương tự chu kỳ máy nạp lệnh. Chỉ khác nó chỉ sử dụng các trạng thái từ $T_1 \rightarrow T_3$ và giá trị tín hiệu trạng thái ($IO/M=0; S_1=1; S_0=0$) được sử dụng trong khoảng thời gian T_1 . Các bước tiến hành chu kỳ máy này như sau:

Bước 1 (thời điểm T_1). Trong khoảng thời gian T_1 , vi xử lý lấy giá trị địa chỉ cần đọc từ con trỏ ngăn xếp, cặp thanh ghi dùng chung hoặc bộ đếm chương trình đưa lên Bus địa chỉ. Đồng thời tích cực tín hiệu ALE để chốt địa chỉ byte thấp.

Trong khoảng thời gian T_1 , vi xử lý giữ các tín hiệu trạng thái $IO/M=0; S_1=1; S_0=0$ cho chu kỳ máy đọc bộ nhớ.

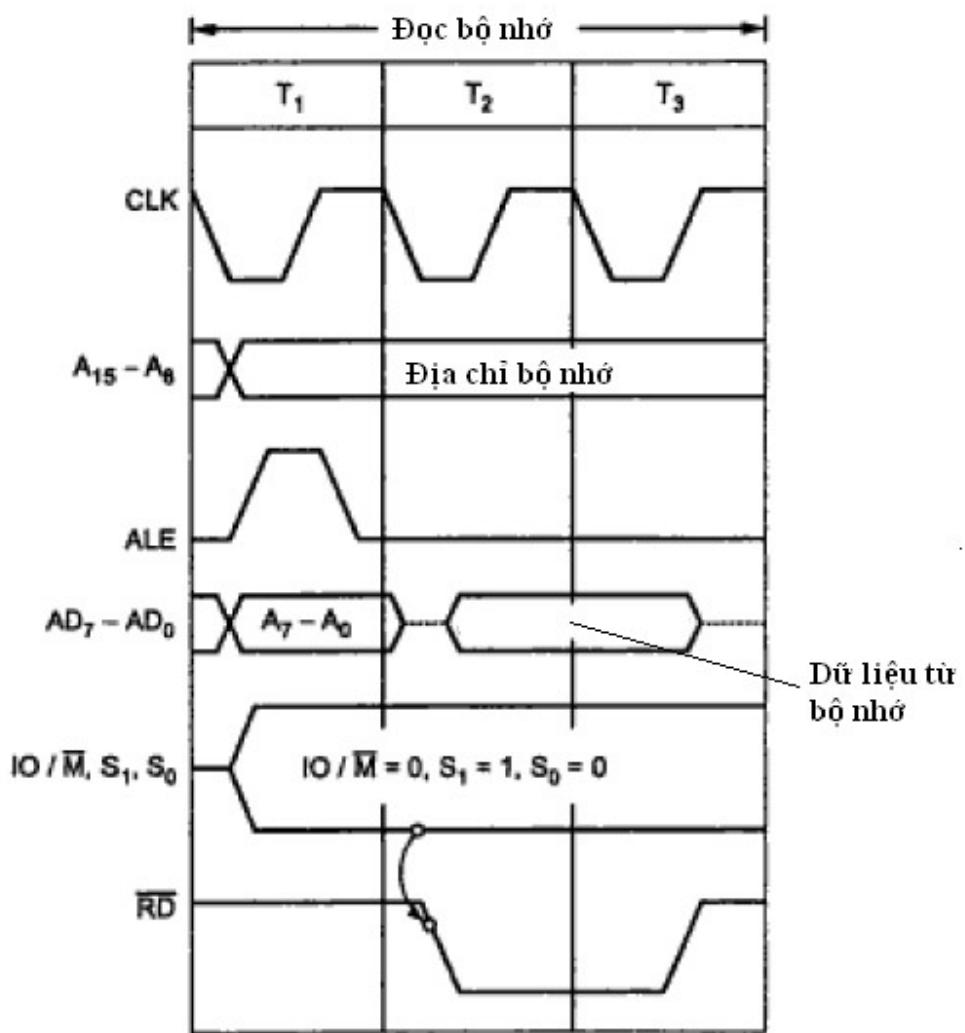
Bước 2 (thời điểm T_2). Trong khoảng thời gian T_2 , vi xử lý giữ tín hiệu RD mức thấp để cho phép truy nhập địa chỉ vùng nhớ. Bộ nhớ sẽ đưa nội dung của vùng nhớ có địa chỉ nhận được lên Bus dữ liệu (AD0-AD7).

Bước 3 (thời điểm T_3). Trong khoảng thời gian T_3 , vi xử lý nạp dữ liệu từ Bus dữ liệu vào các thanh ghi đặc biệt (F, A, B, C, D, E, H, L) và chuyển chân RD về mức cao để đa bộ nhớ về chế độ chờ (disable memory).



→ → dòng dữ liệu , - - - → dòng địa chỉ

Hình 3.24 Dòng dữ liệu từ bộ nhớ đến vi xử lý



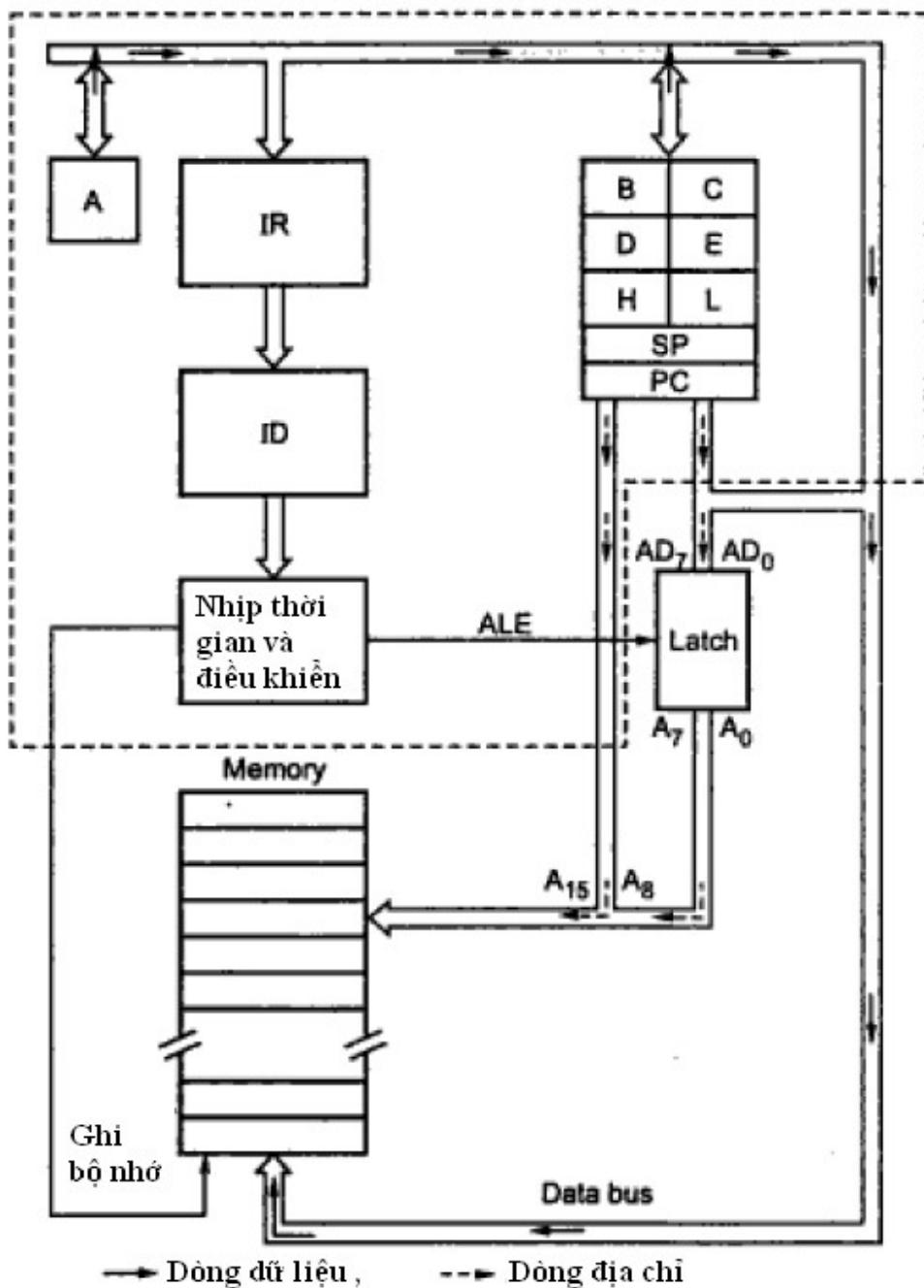
Hình 3.34 b. Chu kỳ đọc bộ nhớ

5.2.3 Chu kỳ máy ghi bộ nhớ:

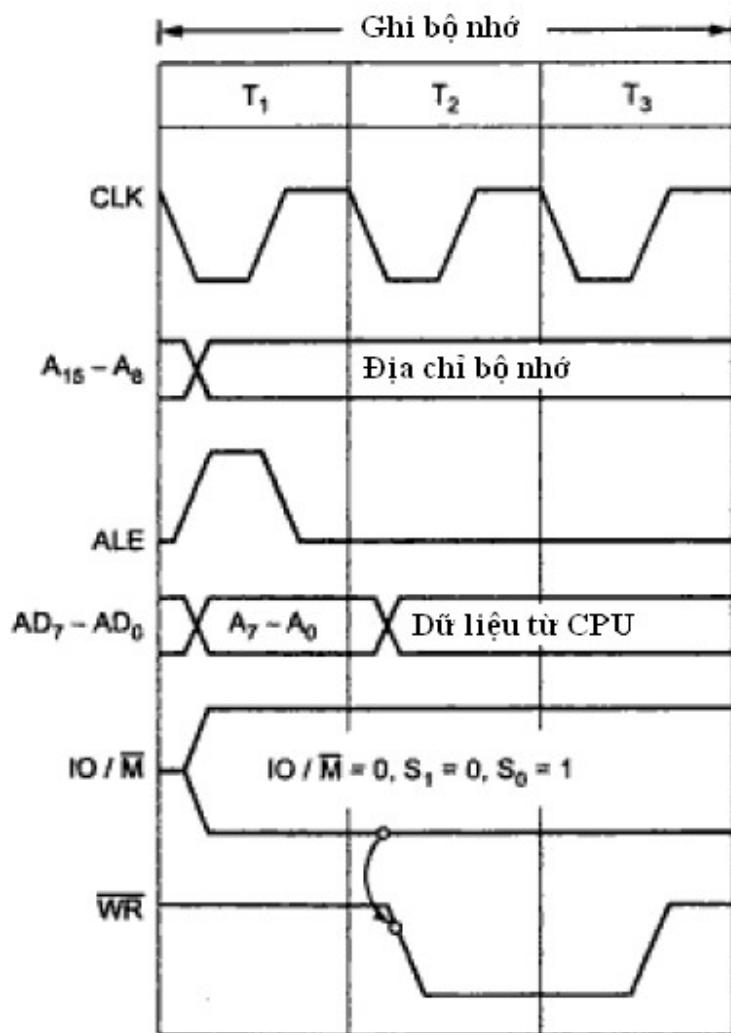
Vì xử lý thực hiện chu kỳ ghi bộ nhớ để lưu dữ liệu vào trong bộ nhớ dữ liệu hoặc bộ nhớ ngăn xếp. chiều dài của chu kỳ máy này là $3T$ ($T_1 \rightarrow T_3$). Trong chu kỳ máy này, vi xử lý lấy giá trị địa chỉ cần ghi từ con trỏ ngăn xếp hoặc cặp thanh ghi dùng chung đưa lên Bus địa chỉ. Thông qua ghi sẽ lưu dữ liệu vào vùng nhớ có địa chỉ cho trước. hình 2.35 thể hiện giản đồ thời gian của chu kỳ ghi bộ nhớ. Chu kỳ này tương tự chu kỳ đọc bộ nhớ chỉ khác thay bằng tín hiệu RD là tín hiệu WR sẽ xuống mức thấp ở thời điểm T_2 và T_3 . Tín hiệu trạng thái với chu kỳ ghi bộ nhớ là $IO/M=0$; $S_1=0$; $S_0=1$. Dưới đây là các bước ghi bộ nhớ:

Bước 1 (thời điểm T_1): ở thời điểm này, vi xử lý lấy giá trị địa chỉ cần ghi từ con trỏ ngăn xếp hoặc cặp thanh ghi dùng chung đặt lên Bus địa chỉ. Đồng thời tích cực tín hiệu

ALE để chốt địa chỉ byte thấp. trong thời điểm T1, vi xử lý giữ tín hiệu trạng thái là IO/M=0; S₁=0; S₀=1.



Hình 3.35 Dòng dữ liệu từ vi xử lý đến bộ nhớ



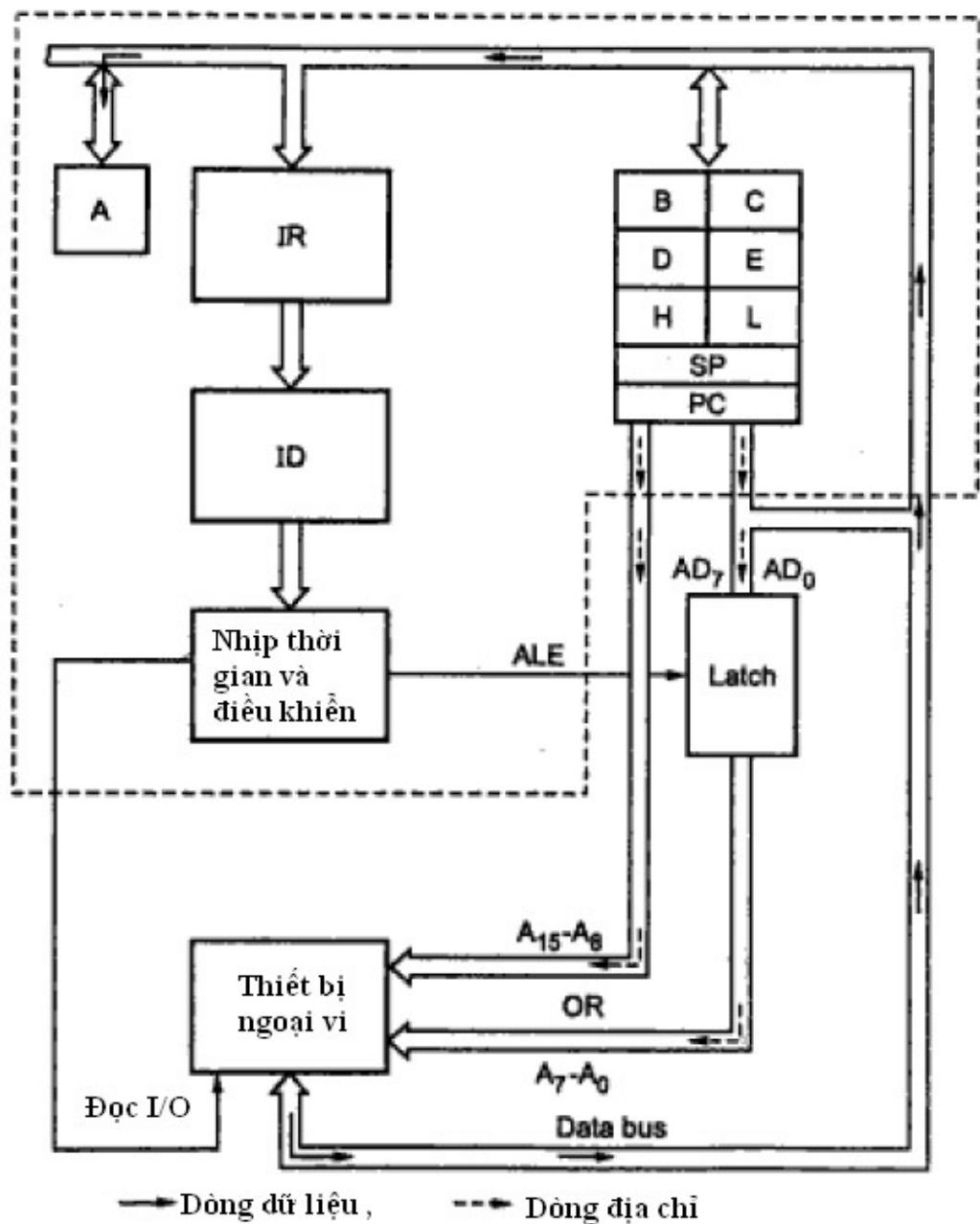
Hình 3.35 Chu kỳ ghi bộ nhớ

Bước 2 (thời điểm T2): ở thời điểm này vi xử lý đưa dữ liệu lên Bus dữ liệu và chuyển tín hiệu WR về mức thấp để ghi vào vùng nhớ định trước.

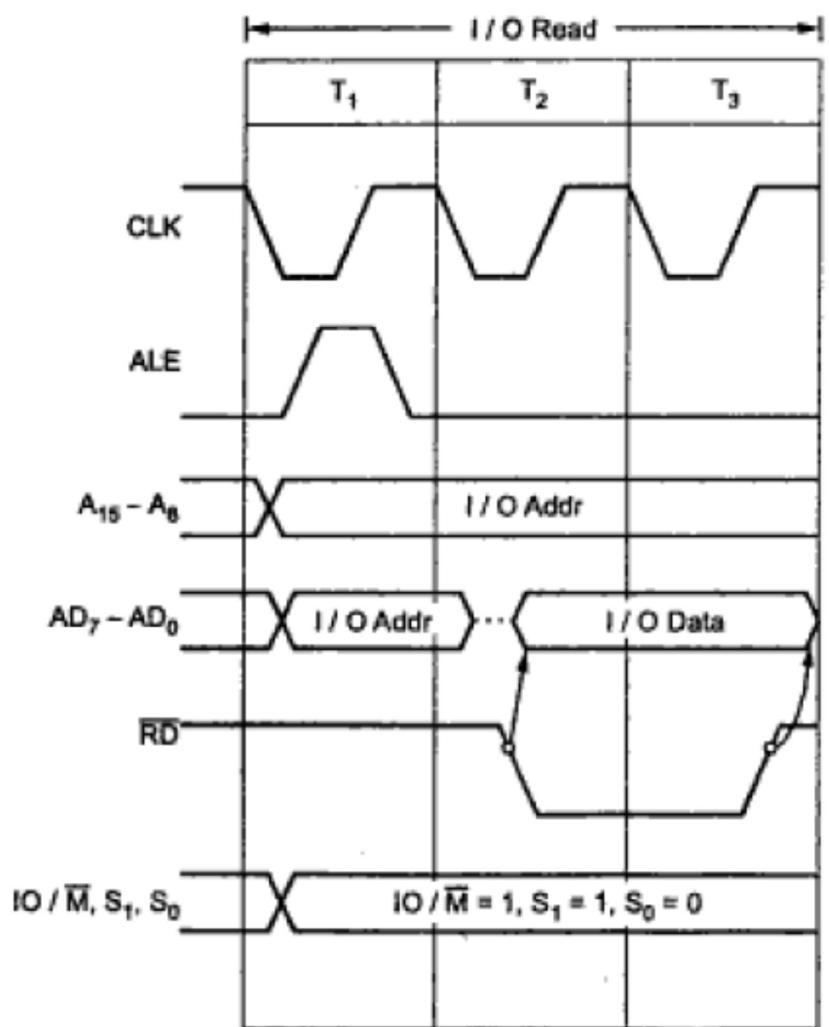
Bước 3 (thời điểm T3): trong khoảng thời gian này, tín hiệu WR chuyển lên mức cao để chuyển bộ nhớ về trạng thái chờ.

5.2.4 Chu kỳ máy ghi và đọc thiết bị ngoại vi (I/O read, I/O write).

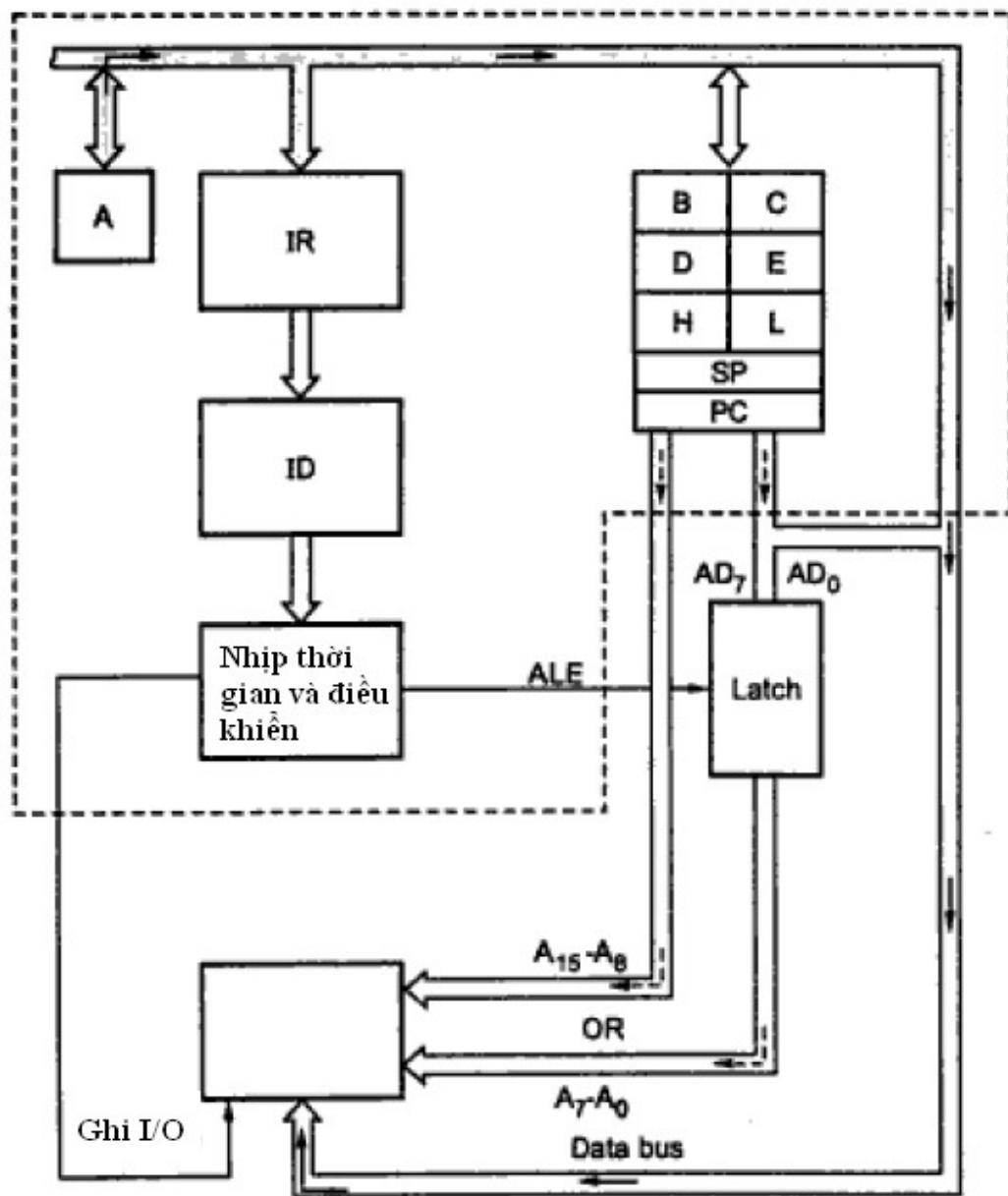
Chu kỳ đọc và ghi I/O tương tự như chu kỳ đọc và ghi bộ nhớ, ngoại trừ tín hiệu IO/M sẽ ở mức cao. Tín hiệu này ở mức cao thể hiện đang làm việc với thiết bị ngoại vi. hình 2.36 và 2.37 thể hiện giản đồ thời gian với chu kỳ đọc và ghi I/O.



Hình 3.36a Dòng dữ liệu từ thiết bị ngoại vi đến vi xử lý

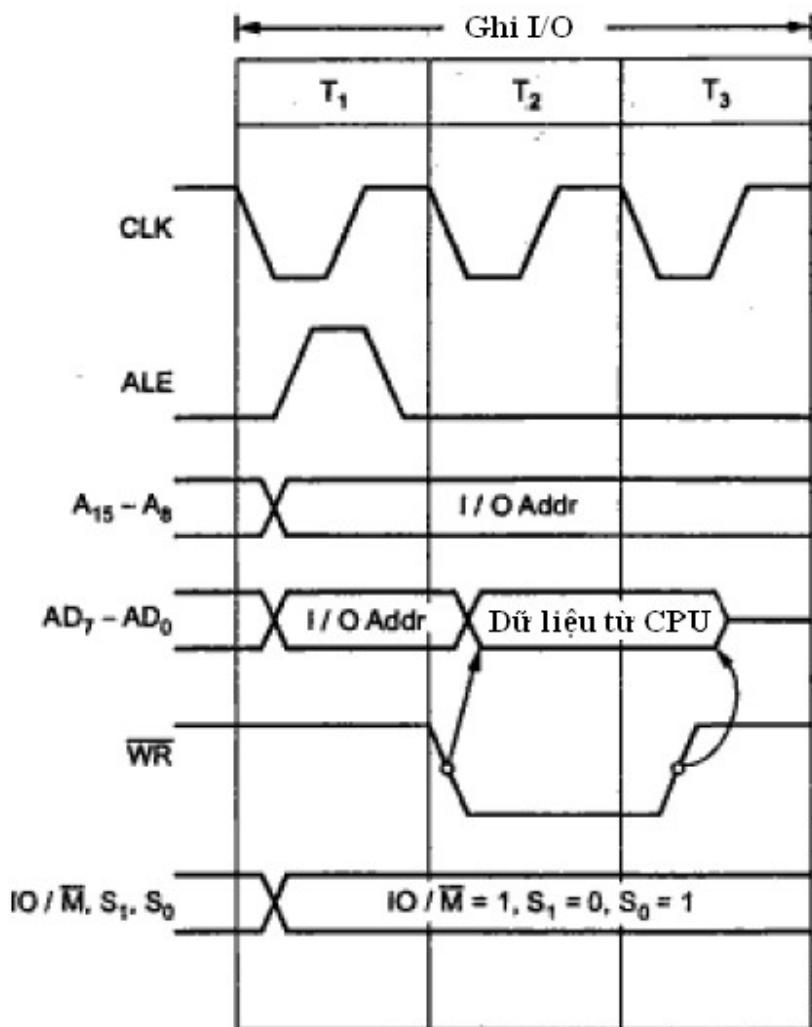


Hình 3.36 b Chu kỳ đọc thiết bị ngoại vi.



→ Dòng dữ liệu , → Dòng địa chỉ

Hình 3.37 a. Dòng dữ liệu từ thiết bị ngoại vi đến vi xử lý



Hình 3.37b Chu kỳ ghi ra thiết bị ngoại vi.

5.2.5 Chu kỳ đáp ứng ngắt

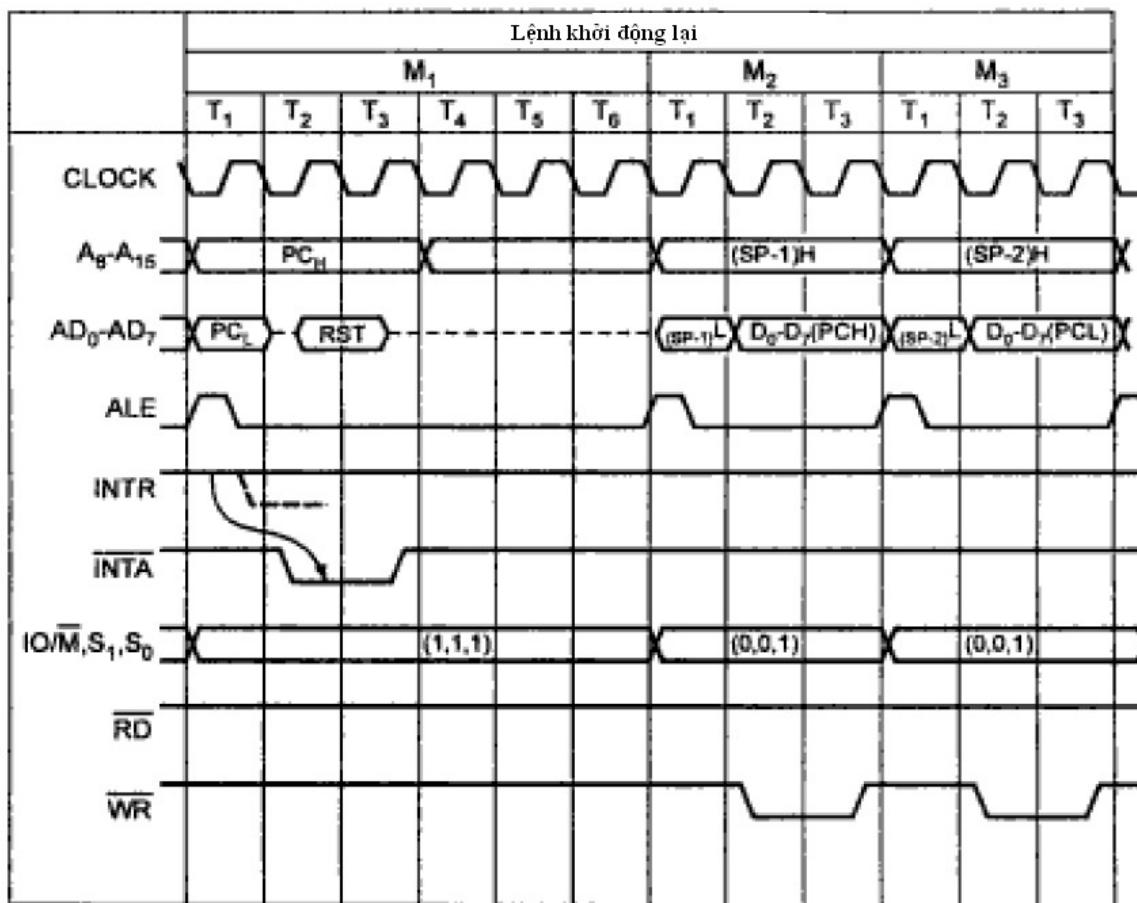
Để đáp ứng tín hiệu INTR, vi xử lý thực hiện chu kỳ đáp ứng ngắt để đọc lệnh máy từ thiết bị bên ngoài. Về lý thuyết, thiết bị bên ngoài có thể đưa bất kỳ lệnh máy nào lên trên Bus dữ liệu, tuy nhiên chỉ có lệnh RST và CALL mới lưu nội dung bộ đếm chương trình (địa chỉ trả về) trước khi chuyển quyền điều khiển cho chương trình van phục vụ ngắt.

Chu kỳ đáp ứng ngắt cho lệnh RST.

Hình 3.38 thể hiện giản đồ thời gian của chu kỳ máy đáp ứng ngắt và thực hiện lệnh RST. Chu kỳ máy đáp ứng ngắt tương tự như chu kỳ nạp lệnh gồm có 2 phần:

- + Tín hiệu INTA được tích cực tương tự như tín hiệu RD.

- + Tín hiệu trạng thái (IO/M, S0 và S1) có giá trị 111 thay cho trường hợp chu kỳ nạp lệnh là 011.



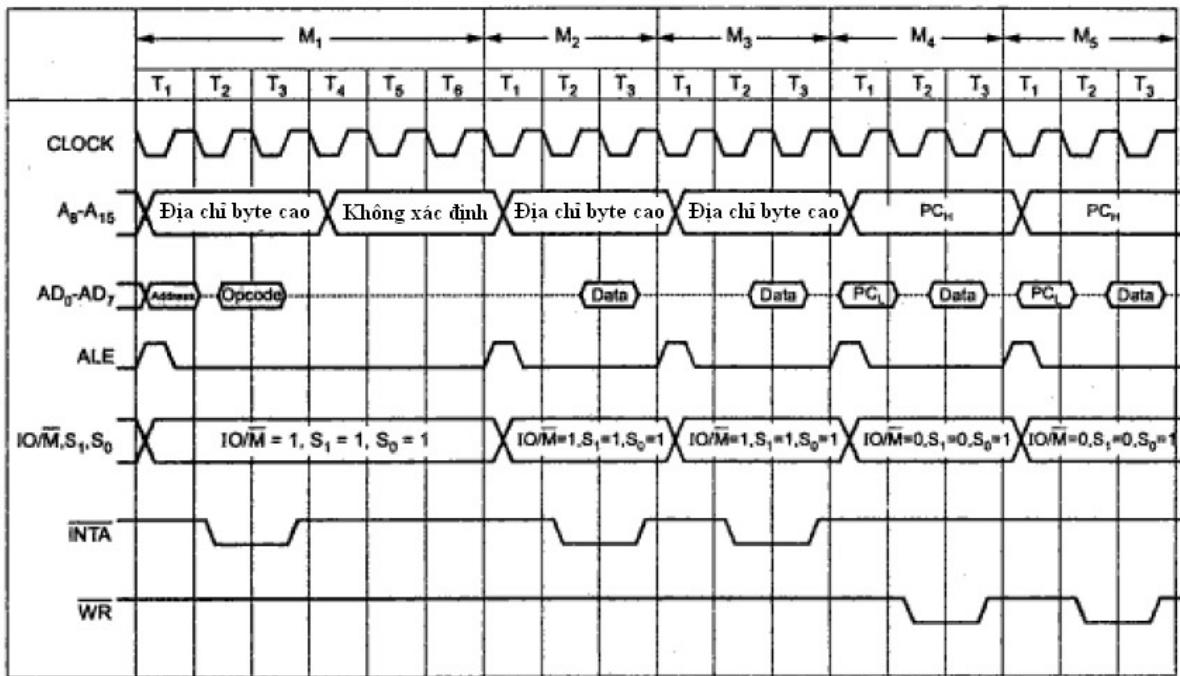
Hình 3.38 Lệnh khởi động lại

Trong chu kỳ máy đáp ứng ngắt (M1), lệnh RST được giải mã và thực hiện lệnh CALL 1 byte đến địa chỉ định trước. Ở chu kỳ máy M2 và M3 thực hiện chu kỳ ghi bộ nhớ với nội dung của bộ nhớ chương trình vào ngăn xếp, và bắt đầu chu kỳ lệnh mới.

Chu kỳ đáp ứng ngắt (M1), lệnh RST được giải mã và thực hiện lệnh CALL 1 byte đến địa chỉ định trước. Ở chu kỳ máy M2 và M3 thực hiện chu kỳ ghi bộ nhớ với nội dung của bộ nhớ chương trình vào ngăn xếp, và bắt đầu thực hiện chu kỳ lệnh mới.

Chu kỳ đáp ứng ngắt cho lệnh CALL

Trong hình 2.38 thể hiện chu kỳ máy đáp ứng ngắt và thực hiện lệnh CALL. Với lệnh CALL, việc nạp địa chỉ 2 byte được thực hiện trong chu kỳ máy M2 và M3. Chu kỳ máy M4 và M5 là chu kỳ ghi bộ nhớ với việc cắt giữ nội dung của bộ đếm chương trình vào ngăn xếp và bắt đầu thực hiện chu kỳ lệnh mới.

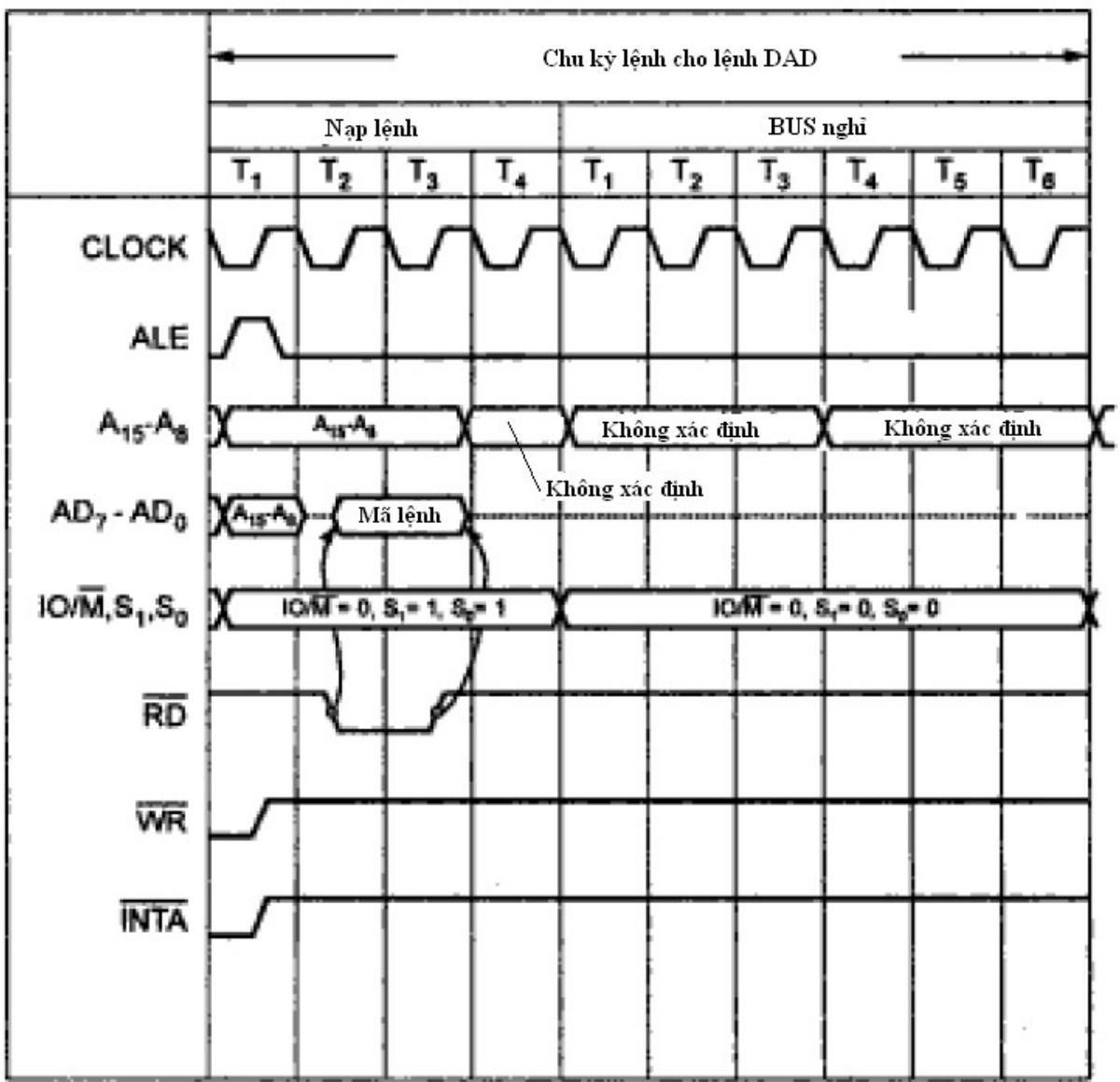


Hình 3.39 Giản đồ thời gian chu kỳ máy lệnh INTA và thực hiện lệnh CALL

*5.2.6 Chu kỳ Bus nghỉ

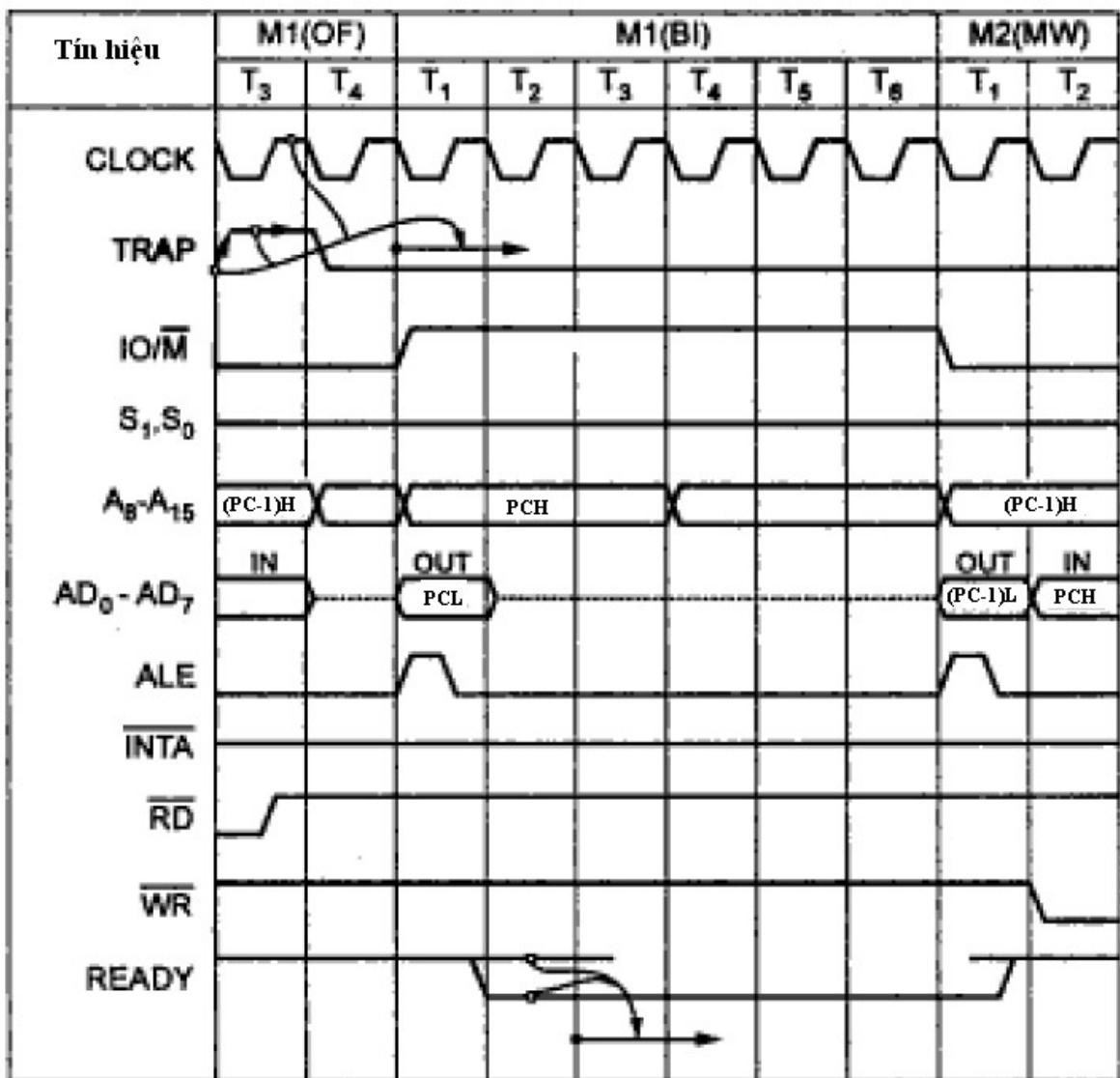
Có vài tình huống khi hoạt động vi xử lý không sử dụng lệnh đọc (read) và ghi (write). Các tình huống này là:

- Khi thực hiện lệnh DAD (lệnh này cộng nội dung của 2 thanh ghi và đưa vào cặp thanh ghi HL) sẽ sử dụng 10 khoảng thời gian T (T₁-T₁₀), điều này có nghĩa là sau khi thực hiện việc nạp lệnh, lệnh DAD sử dụng thêm 6 khoảng thời gian T nữa để cộng nội dung của 2 thanh ghi và đưa vào cặp thanh ghi HL. Khoảng thời gian thêm này sử dụng 2 chu kỳ máy mà không cần truy nhập bộ nhớ hay thiết bị ngoại vi. Các chu kỳ máy này gọi là Bus nghỉ (Bus Idle). Hình 2.39 thể hiện chu kỳ máy Bus nghỉ cho lệnh DAD. Ở chu kỳ máy này tương tự như chu kỳ máy đọc, chỉ khác các tín hiệu RD và ALE không được tích cực.



Hình 3.40 Giản đồ thời gian của lệnh DAD

- Khi thực hiện các công việc cẩu ngắt TRAT và RST, vi xử lý cũng thực hiện chu kỳ máy Idle. Hình 2.40 thể hiện chu kỳ máy Idle với lệnh TRAP. Để đáp ứng ngắt TRAP, vi xử lý sử dụng chu kỳ máy Idle để chạy câu lệnh nạp nội dung bộ đếm chương trình vào ngăn xếp và đưa giá trị 002H (vector địa chỉ của TRAP) vào bộ đếm chương trình.



Hình 3.41 Chu kỳ nghỉ khi thực hiện lệnh TRAP

Số lượng chu kỳ máy để nạp 1 lệnh phụ thuộc vào dạng của lệnh như sau:

Lệnh 1 byte, lệnh 2 byte, lệnh 3 byte.

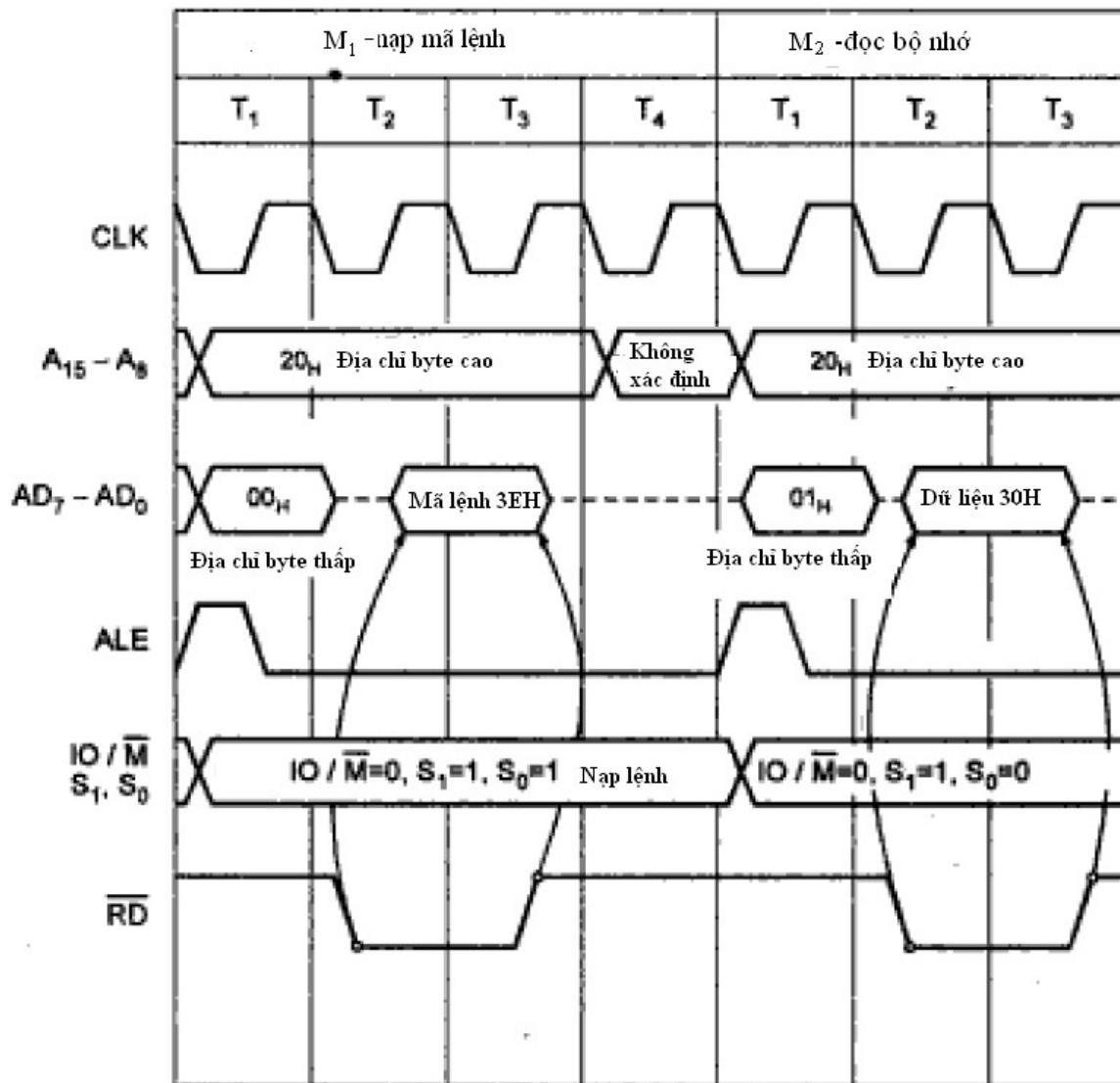
Với lệnh 1 byte không yêu cầu thêm chu kỳ máy. Với lệnh 2 byte cần bổ sung 1 chu kỳ máy đọc bộ nhớ và lệnh 3 byte cần bổ sung 2 chu kỳ máy đọc bộ nhớ.

Số lượng chu kỳ máy để thực hiện lệnh phụ thuộc vào loại lệnh. Tổng số chu kỳ máy nằm trong khoảng từ 1 đến 6.

Dưới đây đưa ra 1 vài ví dụ về giản đồ thời gian thực hiện các lệnh đặc trưng của vi xử lý 8085.

* Ví dụ 1: Lệnh MVI A, 30H. lệnh này nằm ở vùng nhớ 2000H. lệnh này là dạng lệnh 2 byte, byte thứ nhất là nạp mã lệnh MVI A và byte thứ hai là thao tác với các dữ liệu . vi

xử lý trước tiên cần đọc các byte này từ bộ nhớ, việc này cần ít nhất 2 chu kỳ máy. Chu kỳ máy thứ nhất là nạp lệnh và chu kỳ máy thứ hai là đọc bộ nhớ. Hình 3.42 thể hiện giản đồ thời gian của lệnh này.



Hình 3.42 Giản đồ thời gian thực hiện MVI A,30H

- chu kỳ máy thứ nhất là nạp mã lệnh. Trong khoảng thời gian T1 vi xử lý giữ địa chỉ bộ nhớ 16 bit (2000H) từ bộ đếm chương trình lên Bus địa chỉ. Như vậy giá trị 20H được đưa ra Bus A15-A8 và 00H đưa ra AD7-AD0, sau đó tăng bộ đếm chương trình lên giá trị 2001H đến vùng nhớ tiếp theo. Đồng thời giữ tín hiệu trạng thái (IO/M, S1, S0) giá trị 0+1 để thông báo vi xử lý đang ở chu kỳ máy nạp lệnh. Tại thời điểm T2, vi xử lý tích cực tín hiệu RD và đọc nội dung của vùng nhớ 2000H (có giá trị 3EH). Sau đó vi xử lý đưa lệnh vào thanh ghi lệnh và chuyển tín hiệu RD về mức cao (enable) ở thời

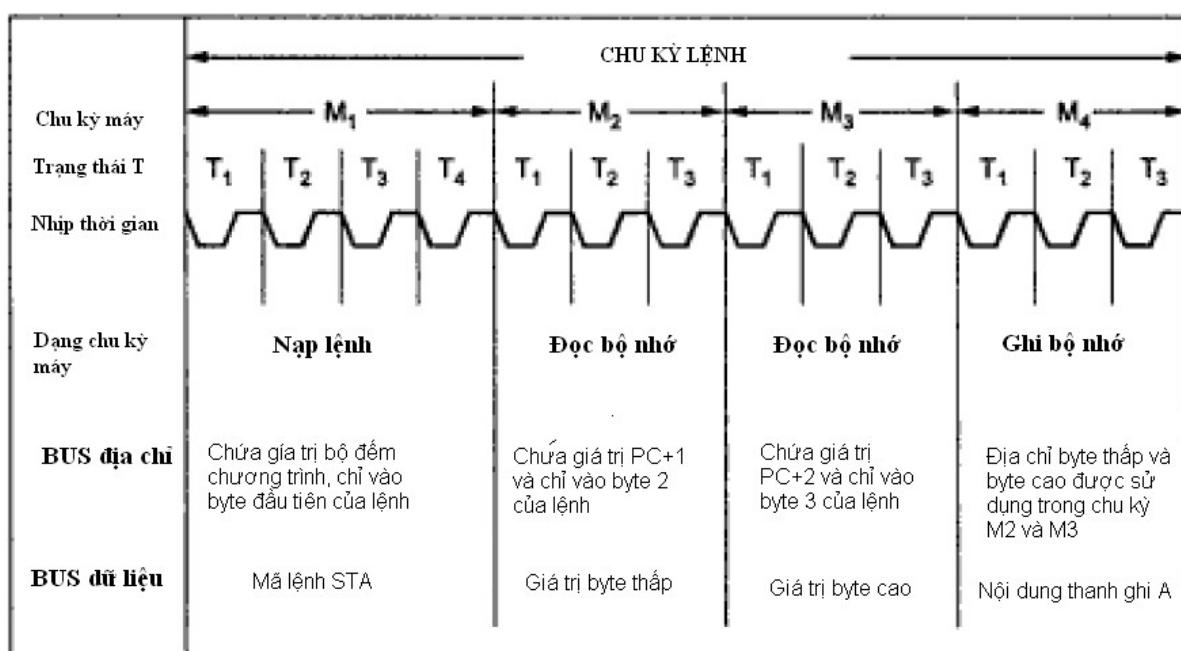
điểm T3. Vào thời điểm T4 , vi xử lý giải mã lệnh và căn cứ vào lệnh thấy rằng cần đọc bộ nhớ để lấy dữ liệu cho byte thứ 2 các lệnh này. Ở khoảng thời gian này (T4) không có việc truy nhập bộ nhớ (Bus Idle) nên các chum A15-A8 không xác định mức, còn các chân AD7-Ado ở trạng thái trở kháng cao.

- sau chu kỳ máy nạp lệnh, vi xử lý thực hiện chu kỳ đọc bộ nhớ vào thời điểm T1, vi xử lý gửi giá trị địa chỉ 2001H lên Bus địa chỉ và tăng bộ đếm chương trình lên vùng nhớ tiếp theo (2002H). vi xử lý tích cực các tín hiệu ALE và gửi tín hiệu trạng thái (IO/M, S1, S0) có giá trị 010 để báo hiệu vi xử lý đang ở chu kỳ đọc bộ nhớ. Vào thời điểm T2 và T3, vi xử lý tích cực tín hiệu RD và đọc 8 bit dữ liệu từ vùng nhớ 2001H (có giá trị 30H). sau đó vi xử lý giữ tín hiệu này vào thanh khi ghi dùng chung A.

* ví dụ 2: Biểu diễn chu kỳ máy và các trạng thái thời gian T khi thực hiện lệnh STA.addr → ghi dữ liệu từ thanh ghi A ra bộ nhớ. Như hình dưới biểu diễn 3 chu kỳ máy để nạp lệnh và 1 chu kỳ máy để giữ nội dung của thanh ghi A ra bộ nhớ.

1. nạp lệnh (thời gian 4T)
2. đọc bộ nhớ (thời gian 3T)
3. đọc bộ nhớ (thời gian 3T)
4. ghi bộ nhớ (thời gian 3T)

Như vậy với lệnh STA cần 13 chu kỳ xung nhịp (T)



Hình 3.43 Chu kỳ máy cho lệnh STA

Ví dụ 4. Biểu diễn chu kỳ máy và trạng thái thời gian T khi thực hiện lệnh LHLD addr. Copy dữ liệu 16 bit từ vùng nhớ vào thanh ghi L và H: $L \leftarrow (\text{addr})$; $H \leftarrow (\text{addr}+1)$. Trong hình 2.44 cần 3 chu kỳ máy để nạp lệnh và 2 chu kỳ máy để đọc dữ liệu 16 bit vào 2 cặp thanh ghi HL.

5.3 Nạp mã lệnh (thời gian 4T)

5.4 đọc bộ nhớ (thời gian 3T)

5.5 đọc bộ nhớ (thời gian 3T)

5.6 ghi bộ nhớ (thời gian 3T)

5.7 ghi bộ nhớ (thời gian 3T)

như vậy lệnh LHLD cần 16 chu kỳ xung nhịp.

5.3 Khái niệm về trạng thái đợi (wait state)

Trong 1 vài ứng dụng, tốc độ của bộ nhớ và thiết bị vào/ra không phù hợp (theo nhịp) được với nhịp thời gian của vi xử lý. Điều này cho nguyên nhân trước thời gian đọc/ghi dữ liệu. trong trường hợp này vi xử lý cần xác định sự sẵn sàng của thiết bị ngoại vi khi giữ dữ liệu cho nó. Nếu chân tín hiệu READY mức cao, báo thiết bị ngoại vi và đã sẵn sang nếu không vi xử lý cần chuyển vào chế độ đợi (wait state). Trong hình 2.45 trình bày giản đồ thời gian cho chu kỳ máy đọc bộ nhớ với chế độ chờ và không cần chờ. Chế độ chờ sẽ được tiếp tục trong khi tín hiệu READY vẫn ở mức thấp.

Sau chế độ chờ, vi xử lý sẽ tiếp tục trạng thái T3 của chu kỳ máy. Trong quá trình chờ nội dung của Bus địa chỉ, dữ liệu và Bus điều khiển được giữ nguyên.

Các câu hỏi ôn tập:

1. Mô tả cấu trúc của vi xử lý 8085
2. Vẽ giản đồ thời gian của tần số dao động ra và trạng thái thời gian T khi vi xử lý sử dụng thạch anh dao động 6.25 MHz; 6.144 MHz; 5 MHz; và 4 MHz.
3. Liệt kê các thanh ghi bên trong của 8085, độ lớn và chức năng của chúng
4. Mô tả cấu trúc của thanh ghi cờ trong vi xử lý 8085
5. Vẽ sơ đồ khôi chức năng vi xử lý 8085 và mô tả từng thành phần trong nó.
6. Mô tả các tín hiệu điều khiển mà vi xử lý 8085 sử dụng
7. Tại sao các chân AD0 – AD7 là dồn kênh?
8. Sử dụng tín hiệu ALE như thế nào?

9. Sử dụng các tín hiệu CLKout và reset out để làm gì?
10. Mô tả chức năng các chân sau: READY, ALE, IO/M, HOLD, RESET
11. Mô tả tín hiệu sử dụng DMA trong vi xử lý 8085
12. Vẽ và giải thích sơ đồ ghép với hệ thống của vi xử lý 8085
13. Thế nào là: chu kỳ lệnh, chu kỳ máy, và trạng thái thời gian T
14. Sử dụng 2 tín hiệu S0 và S1 để làm gì?
15. Mô tả giản đồ thời gian của tín hiệu đơn
16. Mô tả giản đồ thời gian của nhóm tín hiệu
17. Vào thời điểm thời gian nào của T thì tín hiệu ALE tích cực.
18. Mô tả Bus địa chỉ và dữ liệu hoạt động như thế nào trong chế độ tách kenh.
19. Vẽ và giải giản đồ thích thời gian của chu kỳ máy đọc bộ nhớ
20. Vẽ và giải giản đồ thích thời gian của chu kỳ máy ghi bộ nhớ
21. Vẽ và giải giản đồ thích thời gian của chu kỳ máy đọc/ghi I/O
22. Giải thiết chu kỳ đáp ứng ngắn với lệnh RST
23. Giải thiết chu kỳ đáp ứng ngắn với lệnh CALL
24. Giải thiết chu kỳ máy Bus nghỉ
25. Vẽ chu kỳ máy và trạng thái T khi thực hiện lệnh LDA
26. Giải thích ý nghĩa khi sử dụng tín hiệu READY
27. Vẽ và giải thích giản đồ thời gian của chu kỳ máy đọc bộ nhớ với chế độ đợi và không đợi.