HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kỹ Thuật Vi Xử Lý

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: Khoa Học Máy Tính- Khoa CNTT1

Học kỳ/Năm biên soạn:2009

NỘI DUNG

TỔNG QUAN HỆ VI XỬ LÝ

Giảng viên: TS. Phạm Hoàng Duy
E-mail: phamhduy@gmail.com
Năm biên soạn: 2009

Nội dung

- Giới thiệu về hệ vi xử lý.
- Lịch sử phát triển và phân loại các bộ vi xử lý
- Các thành phần của hệ vi xử lý

Giới thiệu

- Máy tính
  - Lưu trữ, xử lý và liên lạc các thông tin dưới dạng số
  - Đơn vị đo thông tin: bit, byte, word, MB,GB
- Các bộ phận căn bản
  - Bộ xử lý trung tâm CPU
  - Bộ nhớ
  - Vào ra

Giới thiệu

- Phần cứng
  - CPU
  - Thanh ghi (register)
  - Buýt
  - RAM,ROM
  - Vào/Ra (Imput/Output)

Giới thiệu

- Phần mềm
  - Chương trình
  - Ngôn ngữ máy
  - Trình dịch
  - Ngôn ngữ lập trình

Phân loại máy tính

- Máy tính lớn (Mainframe)
  - Xử lý khối lượng lớn dữ liệu: thống kê, giao dịch tài chính
- Máy tính con (Minicomputer)
  - Phục vụ nhu cầu tính toán vừa

IBM Z800

Phân loại máy tính

- Máy ví tính (Microcomputer): phục vụ nhu cầu tính toán cá nhân

Phân loại máy tính

- CISC: Máy tính với tập lệnh phức tạp
  - Tập lệnh lớn, nhiều lệnh phức tạp (chu kỳ, định dạng lệnh)
  - Đơn giản hoá trình dịch
  - Chương trình nhỏ và nhanh hơn
  - Song song hoá phức tạp

Phân loại máy tính

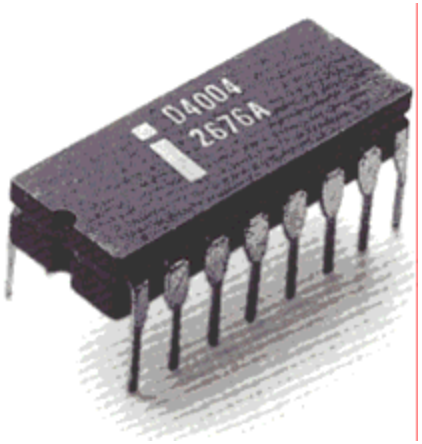
- RISC: Máy tính với tập lệnh rút gọn
  - Một lệnh cho 1 chu kỳ
  - Định dạng lệnh đơn giản (Độ dài lệnh cố định)
  - Chế độ địa chỉ đơn giản
  - Chú trọng các thao tác với thanh ghi
  - Song song hoá thuận tiện

Đánh giá hiệu năng

- Vi xử lý
  - MIPS (millions of instructions per second)
    - (Số lệnh/1 xung nhịp) * xung_nhịp
  - FLOPS (FLoating-point Operations Per Second)
- Hệ thống
  - Bài thử nghiệm (Benchmark)
  - Thao tác vào/ra
  - Tốc độ giao dịch
  - …

Vi xử lý Intel

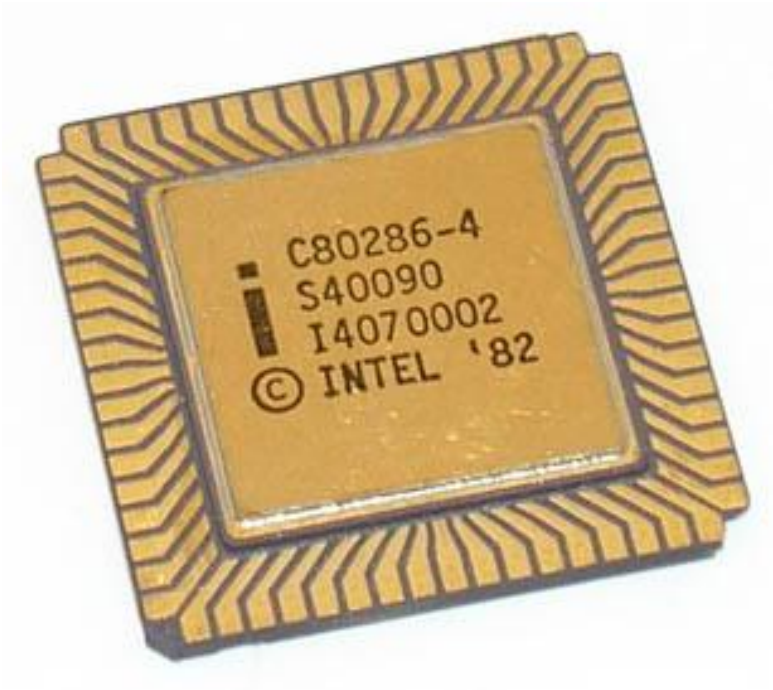
- **1970**
  - Intel 4004 : 4 bit, 2300 transitor, tốc độ 108KHz
  - Intel 8008: 8 bit, 3500 transistor 3500, tốc độ 200KHz
  - Intel 8080: 6000 transistor, tốc độ 2MHz
  - Intel 8086-8088 29,000 transistor, tốc độ 5MHz, 8MHz, 10MHz



Vi xử lý Intel

1980

- 1982: Intel 286 16 bit
  - 134,000 transitor, tốc độ 6MHz, 8MHz, 10MHz, 12.5MHz
- 1985: Intel386™, 32 bit
  - 275,000 transistors, tốc độ: 16MHz, 20MHz, 25MHz, 33MHz
- 1989: Intel486™ DX CPU, 32 bit đầy đủ
  - 1.2 tr transistors, tốc độ 25MHz, 33MHz, 50MHz
  - Tính hợp bộ xứ lý toán học

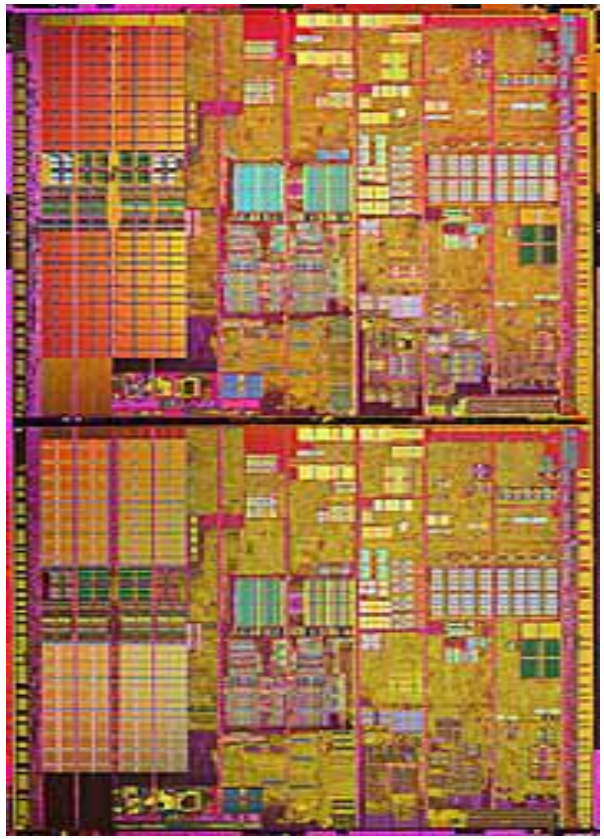


Vi xử lý Intel

- **1990**
  - 1993: Intel® Pentium® Processor
    - 3.1 tr. Transistor, 60MHz, 66MHz
    - 64 bít, hỗ trợ
    - Hỗ trợ xử lý hình ảnh, âm thanh
  - 1997: Pentium II Processor
    - 7.5 tr.Transistor, 200MHz, 233MHz, 266MHz, 300MHz
    - Tăng cường xử lý hình ảnh, âm thanh, video.
  - 1999: Pentium III Processor
    - 9.5 tr. transistors, 650MHz đến 1.2GHz,
    - Tích hợp SIMD hỗ trợ xử lý hình ảnh, âm thanh, 3D nâng cao

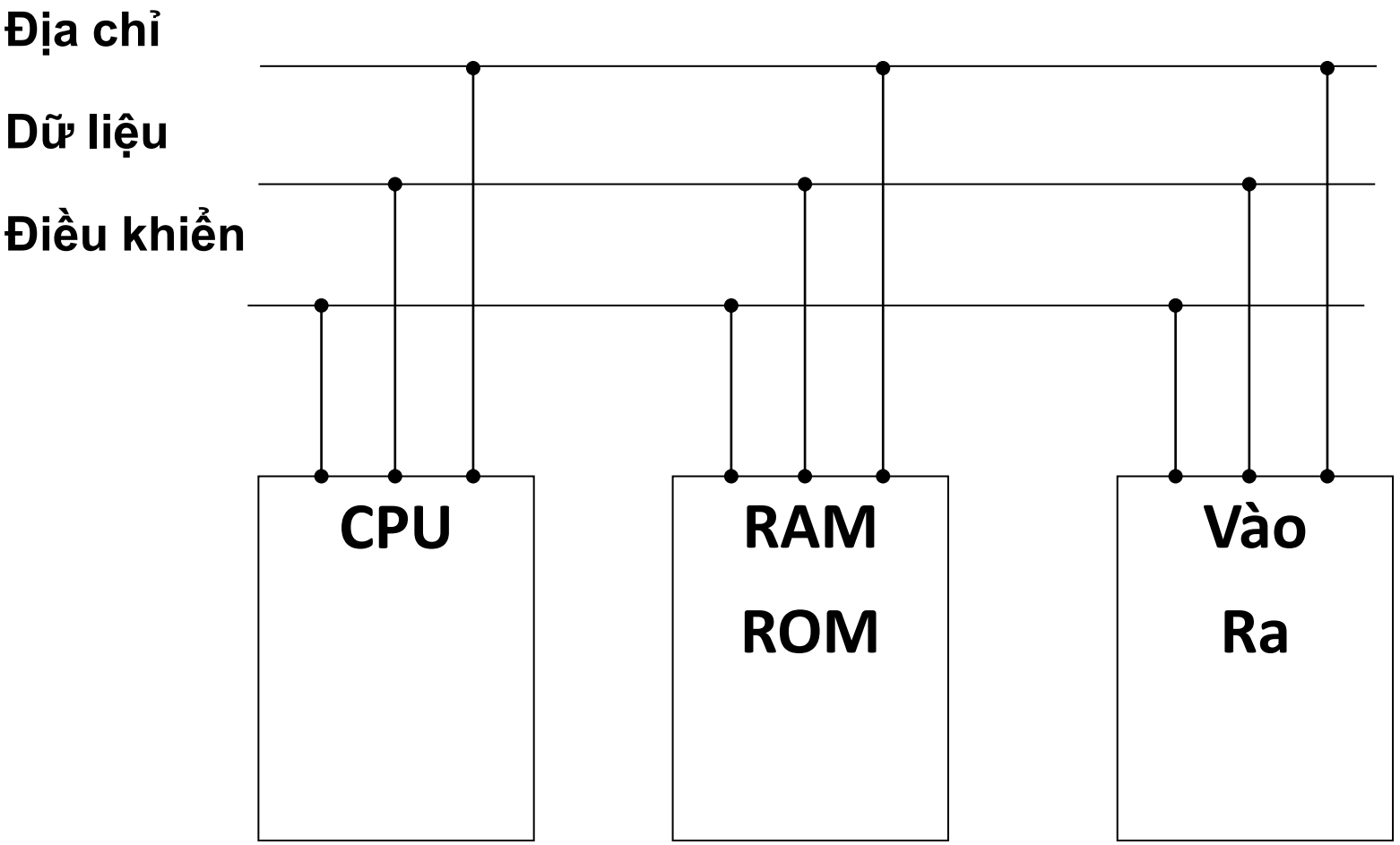
Vi xử lý Intel

- **2000**
  - 2000: Pentium 4 Processor
    - 42 tr. Transistors, 1.30, 1.40, 1.50, 1.70, 1.80 GHz
    - Hỗ trợ xử lý hình ảnh, âm thanh, đồ hoạ 3D thời gian thực
  - 2002: Intel Pentium 4 Processor with Hyper-Threading
  - 2005: Intel Pentium D hai nhân
  - 2006:
    - Intel Core 2 Duo
    - Intel Core 2 Quad: 4 nhân



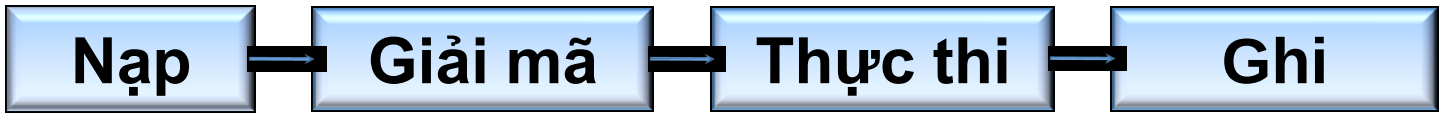
Pentium D 2 nhâm

Kiến trúc căn bản



Bộ xử lý trung tâm CPU

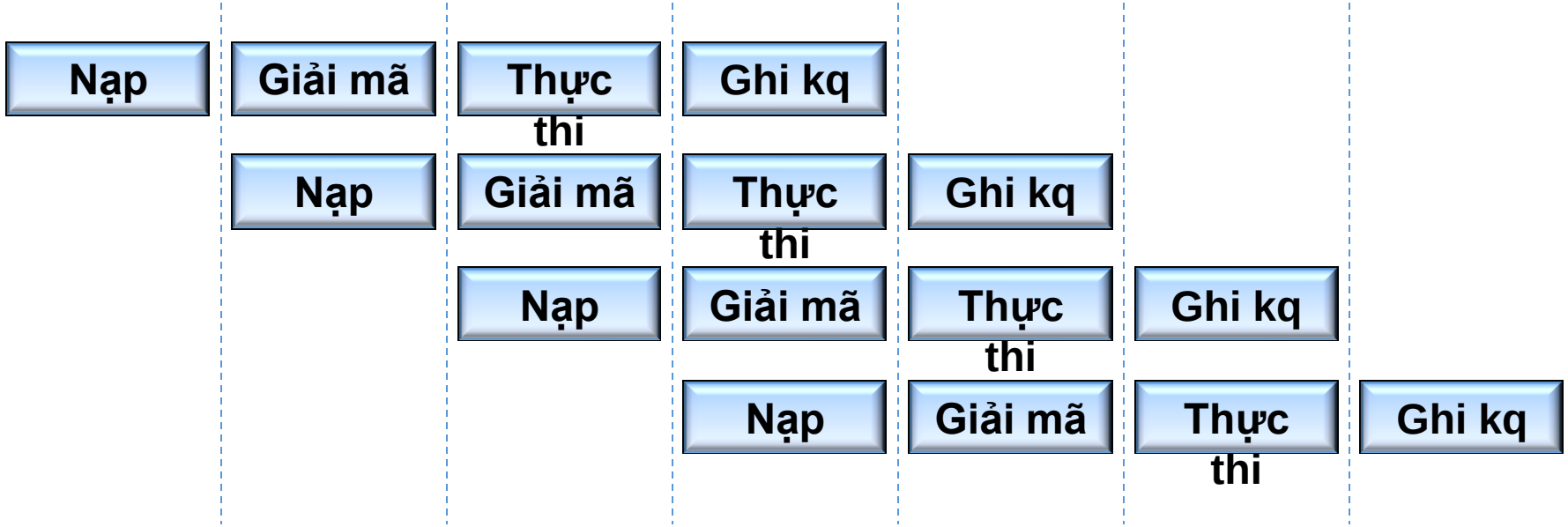
- Thực hiện các lệnh và các thao tác số học, lô-gíc với dữ liệu
- Xung nhịp (Clock)
- Quá trình thực hiện lệnh tiêu biểu



Phân luồng (pipeline)

- Việc thực hiện lệnh được chia nhỏ thành các giai đoạn
- Các giai đoạn được thực hiện kẽ nhau
  - Phân luồng lệnh
  - Phân luồng tính toán

Phân luồng lệnh



Bộ xử lý trung tâm CPU

- Các thanh ghi cơ bản
  - Thanh ghi lệnh
  - Đếm chương trình chứa địa chỉ của câu lệnh kế
  - Thanh ghi địa chỉ: chứa địa chỉ dữ liệu
  - Các thanh ghi đa năng: chứa dữ liệu hoặc kết quả xử lý

Bộ xử lý trung tâm CPU

- Đơn vị điều khiển
  - Đọc và giải mã các lệnh từ bộ nhớ chương trình
  - Sinh ra các tín hiệu điều khiển các bộ phận khác trong hệ VXL.
- Đơn vị xử lý số học và lôgíc
  - Thực hiện các thao tác xử lý dữ liệu

Hệ thống buýt

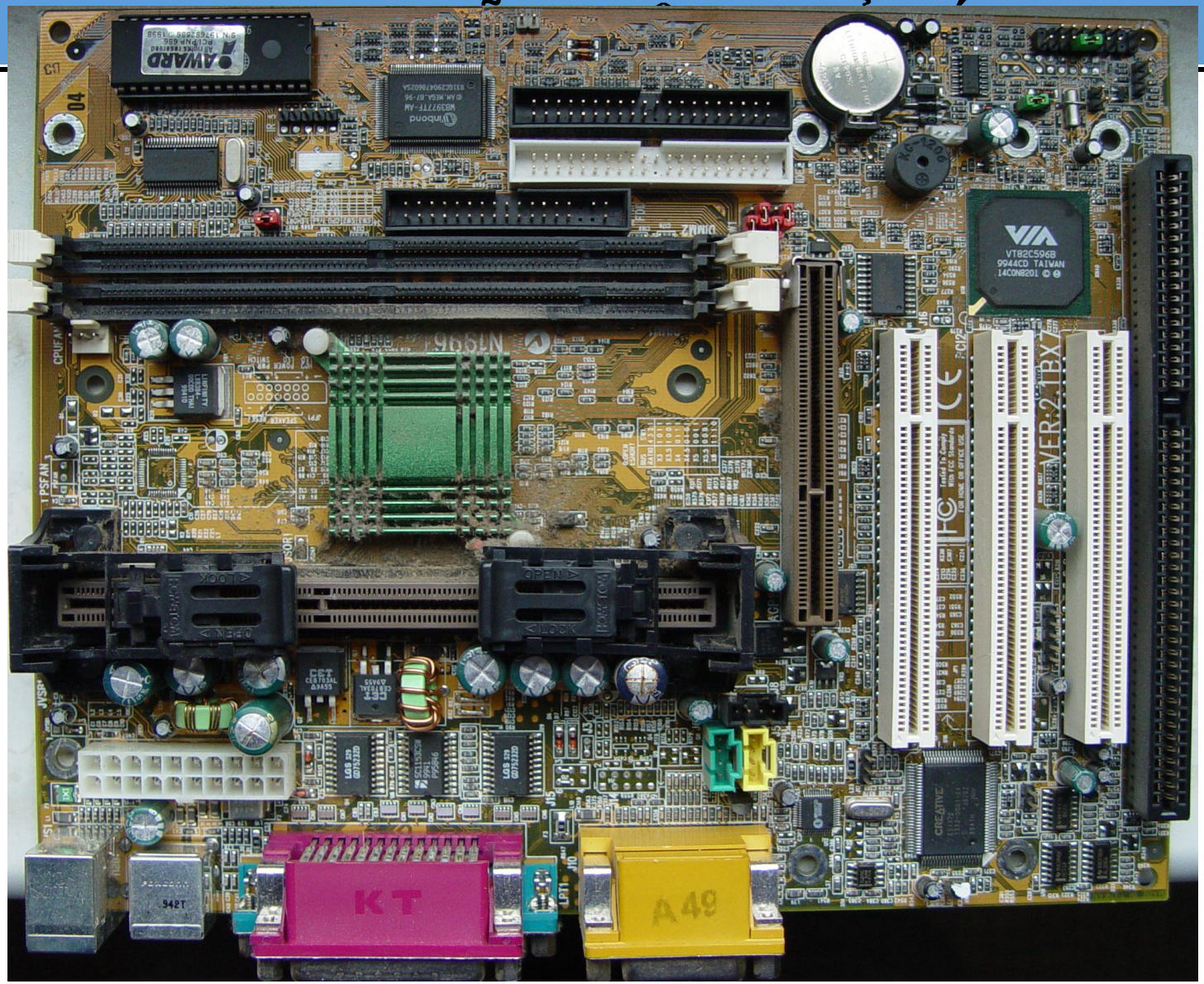
- Truyền thông tin giữa CPU và các bộ phận khác
  - Ghi: dữ liệu truyền từ CPU tới bộ nhớ/thiết bị vào ra
  - Đọc: dữ liệu truyền từ bộ nhớ/thiết bị vào ra tới CPU
- Các loại buýt
  - Buýt địa chỉ truyền thông tin từ CPU tới bộ nhớ/thiết bị vào ra
  - Buýt dữ liệu truyền dữ liệu theo 2 chiều
  - Buýt điều khiển chứa các tín hiệu đồng bộ hoạt động của các bộ phận trong hệ VXL

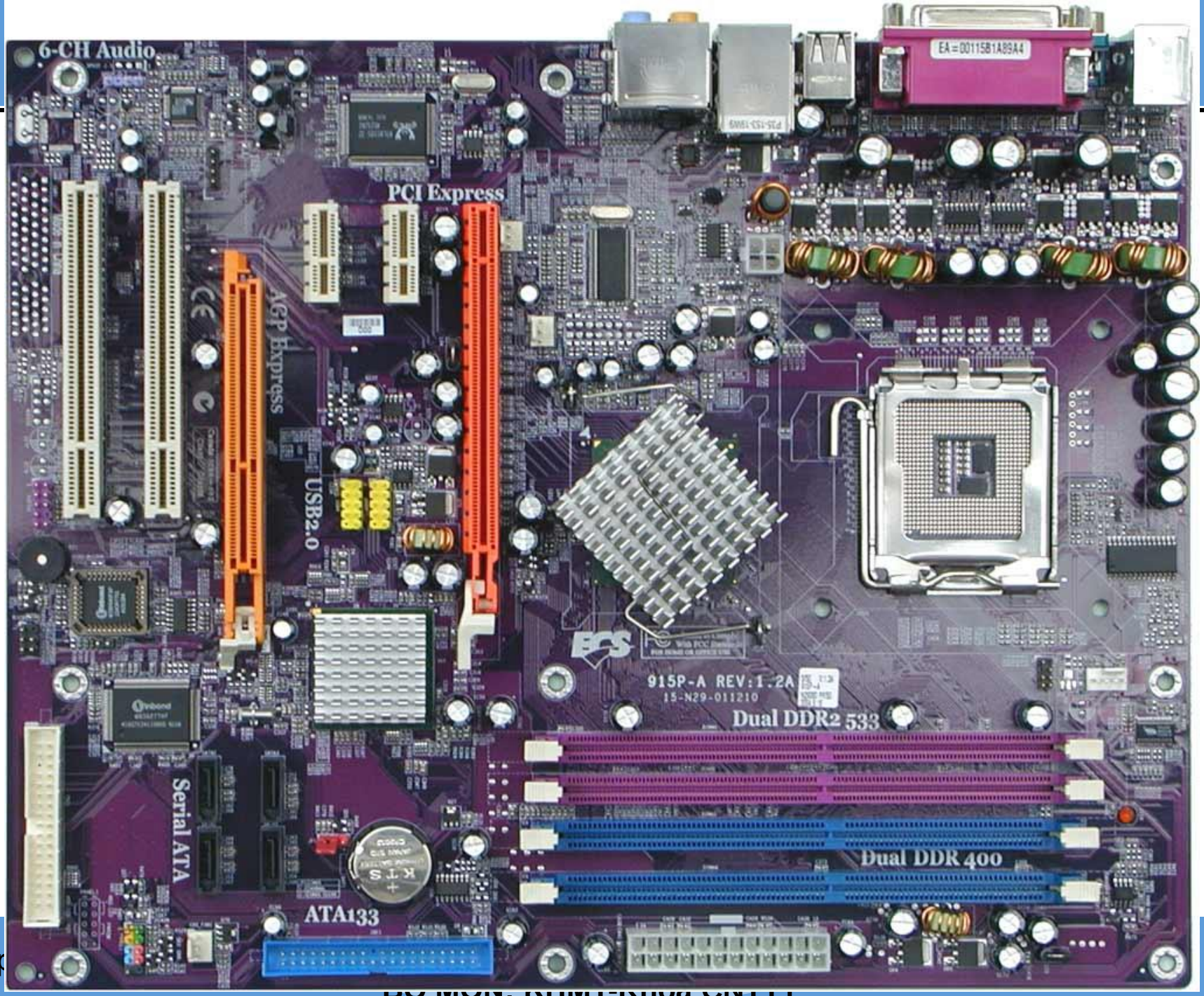
Hệ thống buýt

- Buýt địa chỉ truyền thông tin từ CPU tới bộ nhớ/thiết bị vào ra
  - Độ rộng buýt xác định kích cỡ bộ nhớ tối đa cho chương trình
- Buýt dữ liệu truyền dữ liệu theo 2 chiều
  - Độ rộn buýt xác định khối lượng dữ liệu tối đa cho 1 thao tác đọc/ghi
- Buýt điều khiển chứa các tín hiệu đồng bộ hoạt động của các bộ phận trong hệ VXL
  - Tín hiệu đồng hồ
  - Đọc/Ghi
  - Ngắt

Hệ thống buýt

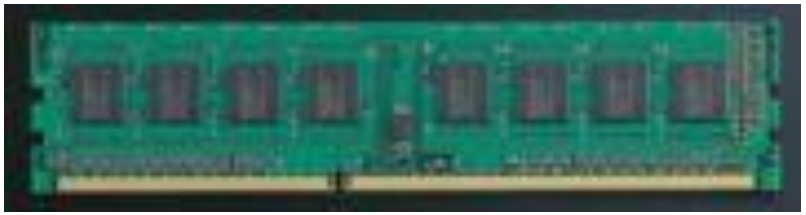
- **I**ndustry **S**tandard **A**rchitecture:8-16 bits,4-8MHz,16MB/s
- **E**xtended **I**ndustry **S**tandard **A**rchitecture : 32 bit, 8MHz, 32MB/s
- **A**ccelerated **G**raphics **P**ort: 32 bit, 66Mhz, 264MB/s(x1)
- **P**eripheral **C**omponent **I**nterconnect : 32 bit, 33MHz, 132MB/s
- PCI-Express: x1,1.25GHz,250MB/s (PCIe 1.x)


VIA
VT82C596B
9944CD TAIWAN
14C0N8201
VER:2.1
KT
A49


6-CH Audio
PCI Express
AGP Express
USB2.0
915P-A REV:1.2A
15-N29-011210
Dual DDR2 533
Dual DDR400
Serial ATA
ATA133

Bộ nhớ

- Lưu trữ các lệnh và dữ liệu
- RAM bộ nhớ truy cập ngẫu nhiên
  - RAM tĩnh
  - RAM động
- ROM bộ nhớ chỉ đọc
  - PROM: Programmable ROM
  - EFROM: Erasable PROM



Vào/Ra

- Truyền dữ liệu giữa CPU và các thiết bị ngoại vi thông qua cổng vào/ra

Modem cạc

Video cạc

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kỹ Thuật Vi Xử Lý

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: Khoa Học Máy Tính- Khoa CNTT1

Học kỳ/Năm biên soạn:2009

NỘI DUNG

BIỂU DIỄN DỮ LIỆU VÀ CÁC THAO TÁC SỐ HỌC

Giảng viên: TS. Phạm Hoàng Duy
E-mail: phamhduy@gmail.com
Năm biên soạn: 2009

Nội Dung

- Các hệ đếm và chuyển đổi
- Các thao tác số học và lô gíc

Các hệ đếm

- Hệ thập phân:
  - Định trị: 0,1,2,3…9
  - Cơ số : 10
  - $90=9*10^1+0*10^0$
- Hệ nhị phân:
  - Định trị: 0,1
  - Cơ số: 2
  - $10 = 1*2^1+0*2^0$
- Hệ thập lục phân:
  - Định trị: 0,1,2..9,A,B,…F
  - Cơ số: 16
  - $F0 = 15*16^1+0*16^0$

Chuyển đổi hệ 10 → hệ 2

- Số nguyên:
  - Chia 2 đến khi thương số = 0,
  - Đảo ngược số dư thu đc số hệ 2
  - 67 → ?

33	16	8	4	2	1	0
1	1	0	0	0	0	1

  - 1000011

Chuyển đổi hệ 10 → hệ 2

- Phân số:
  - Nhân 2 đến khi kết quả = 0 hoặc đạt độ chính xác cần thiết
  - Phần nguyên của kết quả chứa bít chuyển đổi
- 0.575 → ?

0.150	0.3	0.6	0.2	0.4	….
1	0	0	1	0	

- 10010

Các phép toán số học

❖ Cộng

3 + 5 = ???

3 → 0 1 1

5 → 1 0 1

1 | 0 0 0

Các phép toán số học

- **Trừ= Cộng với số bù 2**
  - **Số bù 2:**
    - **x+y=0 ↔ x,y là 2 số bù**
    - **-3: 011 → 100 + 1 → 101**

5	→	0	1	0	1
-3	→	1	1	0	1
		1	0	0	1	0

Các phép toán số học

- Nhân = Cộng & dịch trái

  5 = 1 0 1

  6 = 1 1 0

1 0 1 ×1 1 0						
	0			0	0	0
<<	1		1	0	1	
+			1	0	1	0
<<	1	1	0	1		
+		1	1	1	1	0

Các phép toán số học

- Chia= Dịch & Trừ
  - X/Y= T*Y+D
    1. A=X-Y
       - A>0: $T_i = 1$
       - A<0: $T_i = 0$ & A=A+Y
    2. Dịch trái A 1 bít
    3. Lặp bước 1 : A >Y & A≠0

	1010 ÷ 100							
-Y		1	1	0	0		T	
A		0	1	0	1	0		
-Y	–	1	1	0	0			
A	=	0	0	0	1			1
	< <	0	0	1	0			
-Y	–	1	1	0	0			
A	=	1	1	1	0		1	0
Y	+	0	1	0	0			
	=	0	0	1	0		1	0

Biểu diễn số thực

X= (Dấu)(Phần định trị)*(Cơ số) $^{\text{Số mũ}}$

Dấu	Số mũ	Phần định trị

- Cộng/Trừ = Qui đồng số mũ + cộng/trừ định trị
  - $X_1Y^m \pm X_2Y^n = (X_1' \pm X_2')Y^{max(m,n)}$
- Nhân/Chia = Cộng/Trừ số mũ + Nhân/Chia phần định trị
  - $X_1Y^m \ \times/\div \ X_2Y^n = X_1 \ \times/\div \ X_2Y^{m\pm n}$

Các phép toán logic

OR

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

AND

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

XOR

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kỹ Thuật Vi Xử Lý

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: Khoa Học Máy Tính- Khoa CNTT1

Học kỳ/Năm biên soạn:2009

NỘI DUNG

LẬP TRÌNH HỢP NGỮ VỚI 8088

Giảng viên: TS. Phạm Hoàng Duy
E-mail: phamhduy@gmail.com
Năm biên soạn: 2009

Nội dung

- Giới thiệu hợp ngữ
- Các câu lệnh căn bản 8088
- Các cấu trúc điều khiển

Câu hỏi

- Xây dựng lưu đồ
- Các câu lệnh xử lý dữ liệu
- Các cấu trúc điều khiển

Hợp ngữ

❖ Cú pháp câu lệnh

Tên	Mã lệnh	Toán hạng		Chú giải
Cộng:	ADD	AH	30H	AH=AH+30H

• Định nghĩa biến và hằng số

Tên	Độ dài	Giá trị	Chú giải
X	DB	1FH	Khởi tạo 1 byte
Y	DW	FFFFH	Khởi tạo 1 word
str	DB	‘string’	Chuỗi
M	DB	DUP(?)	Mảng
Hang	EQU	1	Hằng số

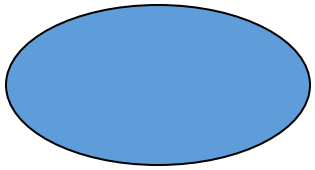
Định nghĩa các đoạn

- .Stack kích_cỡ
  - Định nghĩa đoạn ngăn xếp
- .Data
  - Định nghĩa đoạn dữ liệu
- .Code
  - Định nghĩa đoạn mã lệnh

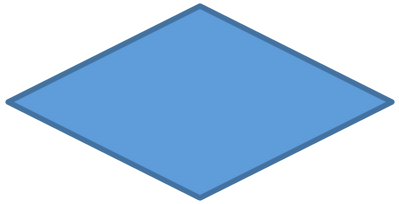
1. .Stack 100
2. **.Data**
3. ….
4. **.Code**
5. ….

Lưu đồ thuật toán

❖ Bắt đầu/Kết thúc



- **Điều kiện**



- **Thao tác**

Các thao tác số học

- Dịch trái
  - SHL Đích,CL
    - MOV CL,2
    - MOV AX,5
    - SHL AX,CL
- Dịch phải
  - SHR Đích,CL
- Tăng
  - INC DX; DX++
- Giảm
  - DEC DX;DX--

- A=A+B → ADD A, B
  - VD. 3+5
  - MOV AX,3
  - ADD AX,5
- A=A-B → SUB A, B
  - A=5-3
  - MOV BX,5
  - SUB BX,3

Các thao tác số học

- A=A*B →
  - MOV AX/AL,A;
  - MUL B

  B = 1 byte: AX = tích

  B = 2 byte: DXAX = tích

- A=A/B →
  - MOV [AX,DX],A
  - DIV B

  B = 1 byte: AL thương số, AH số dư

  B = 2 byte: AX thương số, DX số dư

Các thao tác lôgic

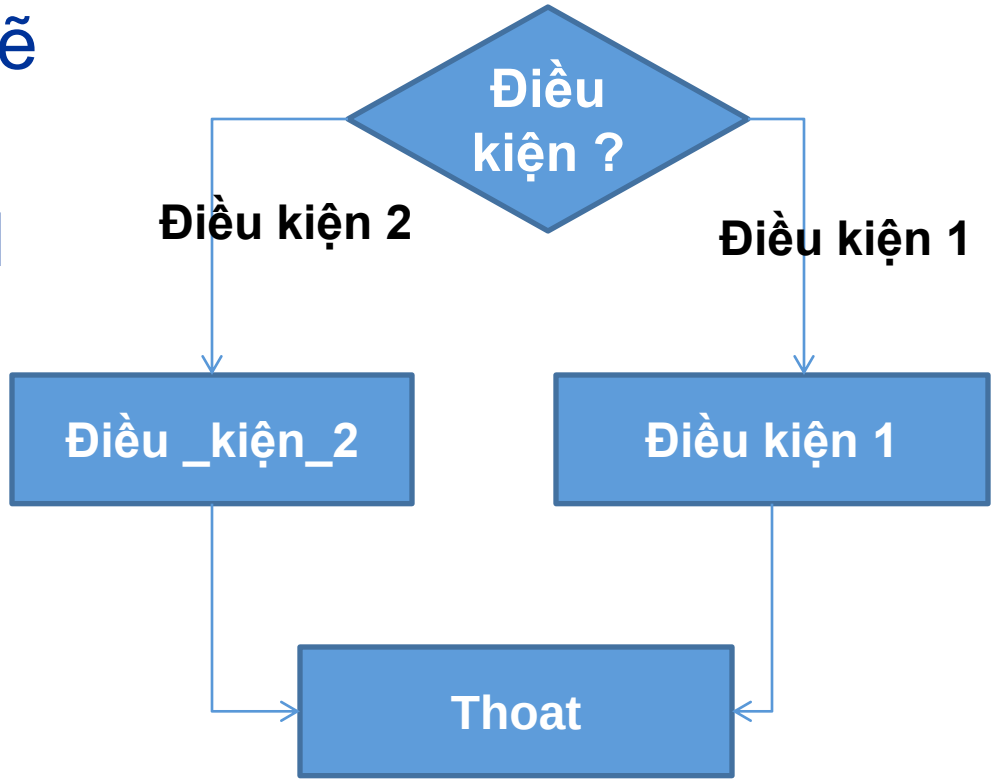
- A = A $\wedge$ B $\rightarrow$ AND A,B
- A = A $\vee$ B $\rightarrow$ OR A,B
- A = A xor B $\rightarrow$ XOR A,B
- A > B CMP A,B
  - A = B; ZF = 1
  - A > B; ZF = 0, CF = 0
  - A < B; ZF = 0,CF =1

Rẽ nhánh

- Ví dụ
  1. CMP AX,10; kiểm tra điều kiện
  2. JXX yyy; rẽ nhánh
- JMP XX; Jump
- JL XX; Jump if less - SF <> OF
- JG XX; Jump if greater - SF = OF
- JE XX; Jump if equal - ZF = 1
- JA XX; Jump if above - (CF = 0) and (ZF = 0)
- JB XX; Jump if below - CF = 1

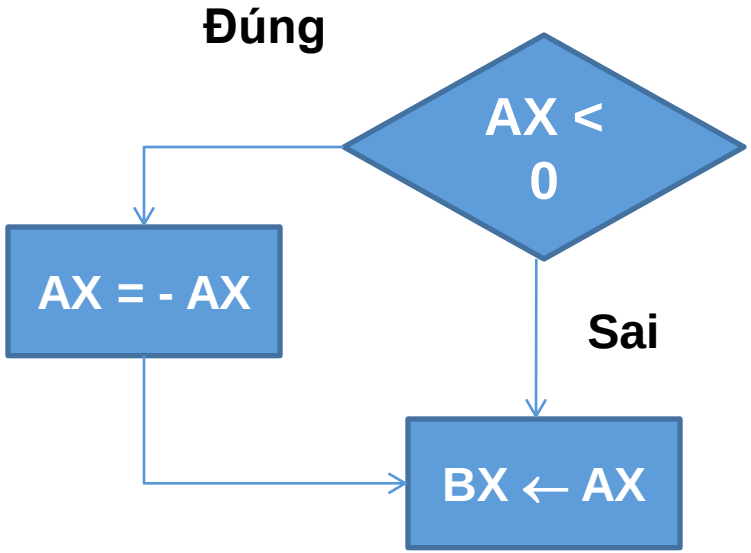
Cấu trúc IF …. THEN

1. CMP X,Y ; Điều kiện
2. Jmp Điều_kiện_2; Rẽ nhánh
3. …… ; Điều kiện 1
4. JMP Thoat
5. Điều_kiện_2:
6. …
7. Thoat:



Cấu trúc IF … THEN

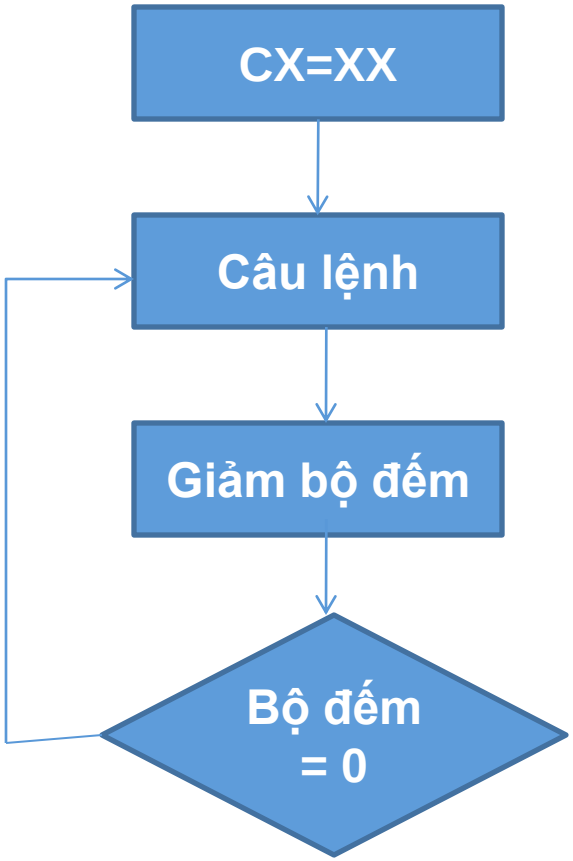
- IF điều kiện THEN thao tác
- Gán BX giá trị tuyệt đối AX
  1. CMP AX,0
  2. JNL GAN
  3. NEG AX
  4. GAN: MOV BX, AX



Cấu trúc lặp FOR

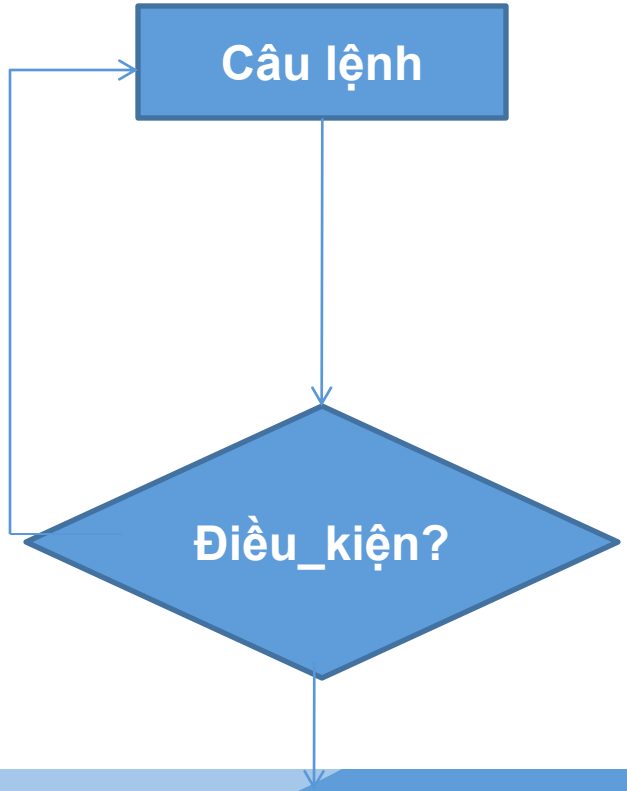
- Sử dụng lệnh LOOP
- Số lần lặp CX

1. MOV CX,10
2. MOV AH,2
3. MOV DL,'$'
4. Hien: INT 21H
5. LOOP Hien

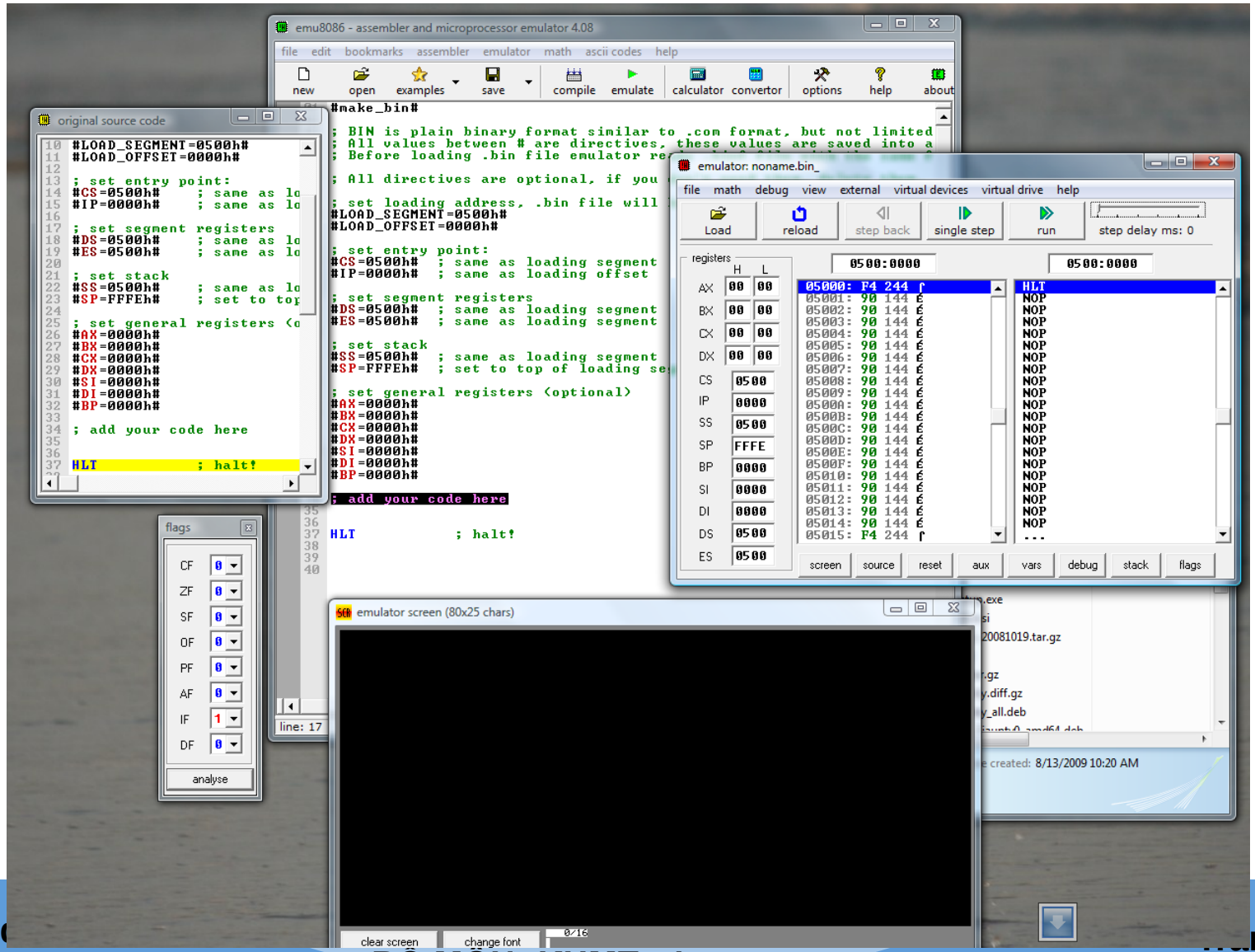


Cấu trúc lặp REPEAT UNTIL

1. …
2. Tiep:…
3. ….
4. CMP X,Y; điều kiện
5. JMP điều_kiện=sai;



Emu8086


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
original source code
#LOAD_SEGMENT=0500h#
#LOAD_OFFSET=0000h#
; set entry point:
#CS=0500h# ; same as lo
#IP=0000h# ; same as lo
; set segment registers
#DS=0500h# ; same as lo
#ES=0500h# ; same as lo
; set stack
#SS=0500h# ; same as lo
#SP=FFFEh# ; set to top
; set general registers (o
#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
#SI=0000h#
#DI=0000h#
#BP=0000h#
; add your code here
HLT ; halt!
; BIN is plain binary format similar to .com format, but not limited
; All values between # are directives, these values are saved into a
; Before loading .bin file emulator re
; All directives are optional, if you
; set loading address, .bin file will
#LOAD_SEGMENT=0500h#
#LOAD_OFFSET=0000h#
; set entry point:
#CS=0500h# ; same as loading segment
#IP=0000h# ; same as loading offset
; set segment registers
#DS=0500h# ; same as loading segment
#ES=0500h# ; same as loading segment
; set stack
#SS=0500h# ; same as loading segment
#SP=FFFEh# ; set to top of loading se
; set general registers (optional)
#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
#SI=0000h#
#DI=0000h#
#BP=0000h#
; add your code here
HLT ; halt!
line: 17
flags
CF 0
ZF 0
SF 0
OF 0
PF 0
AF 0
IF 1
DF 0
analyse
emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 00
BX 00 00
CX 00 00
DX 00 00
CS 0500
IP 0000
SS 0500
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0500
ES 0500
0500:0000
0500:0000
05000: F4 244 ſ
05001: 90 144 É
05002: 90 144 É
05003: 90 144 É
05004: 90 144 É
05005: 90 144 É
05006: 90 144 É
05007: 90 144 É
05008: 90 144 É
05009: 90 144 É
0500A: 90 144 É
0500B: 90 144 É
0500C: 90 144 É
0500D: 90 144 É
0500E: 90 144 É
0500F: 90 144 É
05010: 90 144 É
05011: 90 144 É
05012: 90 144 É
05013: 90 144 É
05014: 90 144 É
05015: F4 244 ſ
HLT
NOP
screen source reset aux vars debug stack flags
emulator screen (80x25 chars)
clear screen change font
20081019.tar.gz
.diff.gz
_all.deb
created: 8/13/2009 10:20 AM

Bài tập

- Kỹ thuật VXL, Văn Thế Minh
  - Ví dụ 1-11 (tr126)

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kỹ Thuật Vi Xử Lý

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: Khoa Học Máy Tính- Khoa CNTT1

Học kỳ/Năm biên soạn:2009

NỘI DUNG

Ghép nối 8088 với bộ nhớ

Giảng viên: TS. Phạm Hoàng Duy
E-mail: phamhduy@gmail.com
Năm biên soạn: 2009

Nội dung

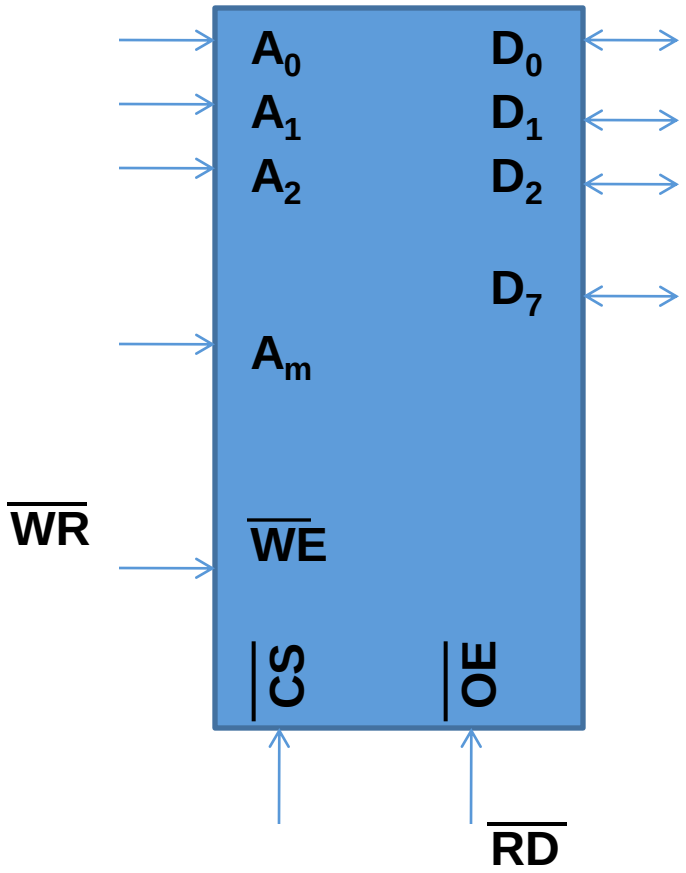
- Các loại bộ nhớ bán dẫn
- Giải mã địa chỉ
- Bộ nhớ kiểm tra chẵn lẻ

Bộ nhớ bán dẫn

- Bộ nhớ ROM
- Bộ nhớ EPROM
- Bộ nhớ RAM
  - RAM tĩnh
  - RAM động

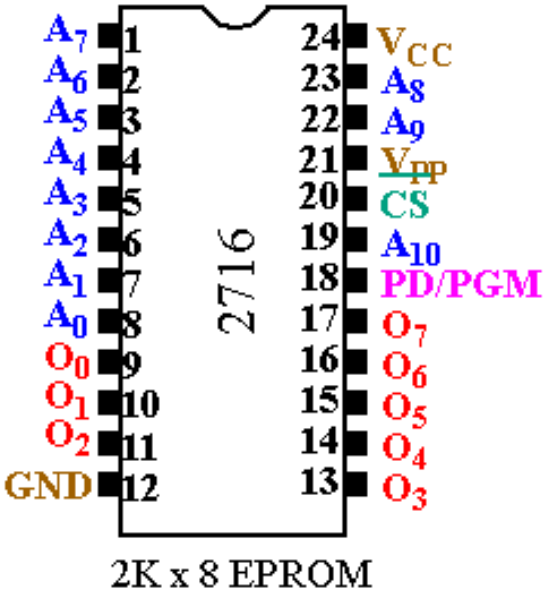
Sơ đồ chức năng của IC nhớ

- A_1-A_m: Địa chỉ
- D_0-D_7: Dữ liệu
- WE: Cho phép ghi
- OE: Cho phép ra
- CS: Kích hoạt

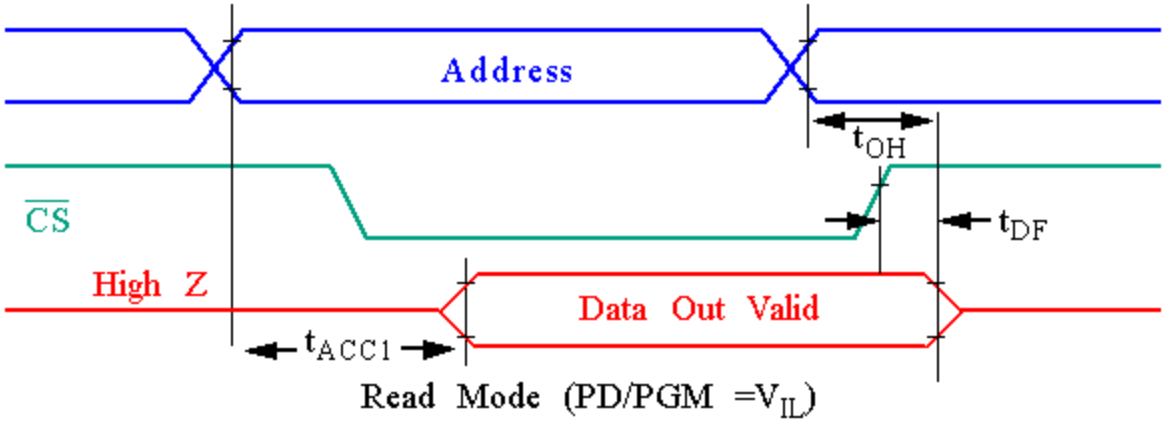


EFROM Intel 2176(2Kx8)

- A0-A10: Địa chỉ
- CS: chọn chíp(0-đọc,1-ghi)
- PD/PGM: Duy trì/Lập trình V_{pp} = 25V



2K x 8 EPROM

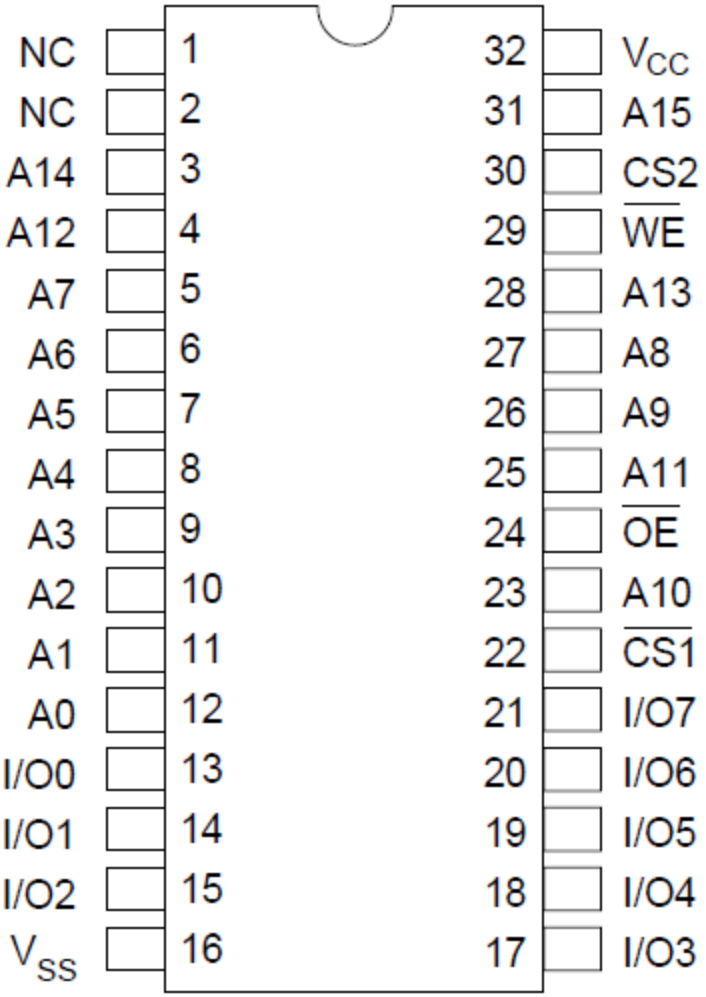


Read Mode (PD/PGM =V_{IL})

SRAM

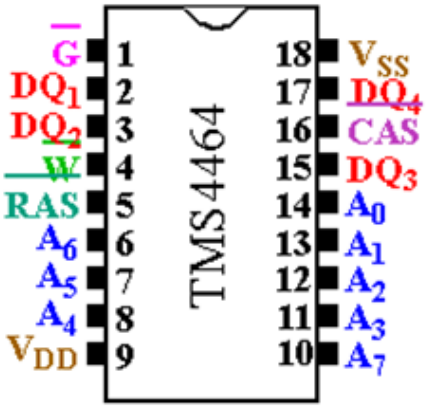
- Hitachi HM62864 - 64K×8
  - Tốc độ 50-85ns

Pin Name	Function
A0 to A15	Address
I/O0 to I/O7	Input/output
$\overline{CS1}$	Chip select 1
CS2	Chip select 2
$\overline{WE}$	Write enable
$\overline{OE}$	Output enable
NC	No connection
V_{cc}	Power supply
V_{ss}	Ground



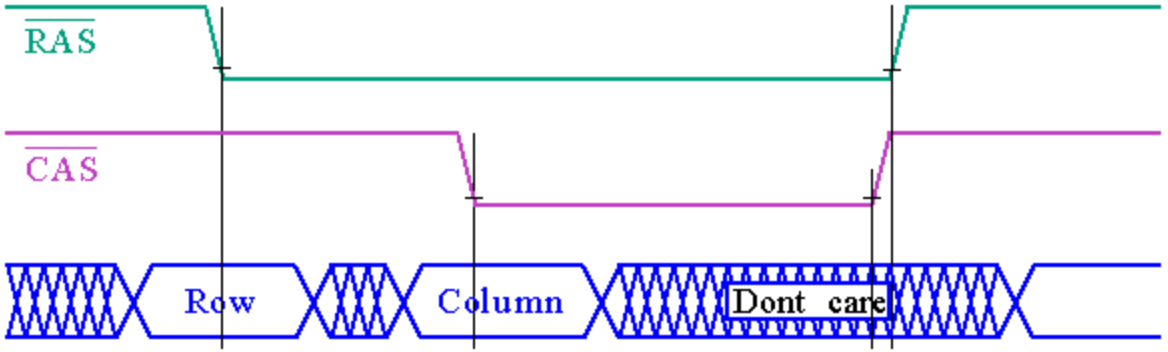
DRAM

- TMS 4464
  - 64K×4
- 64K = {RA_0 – RA_7}+{CA_0–CA_7}



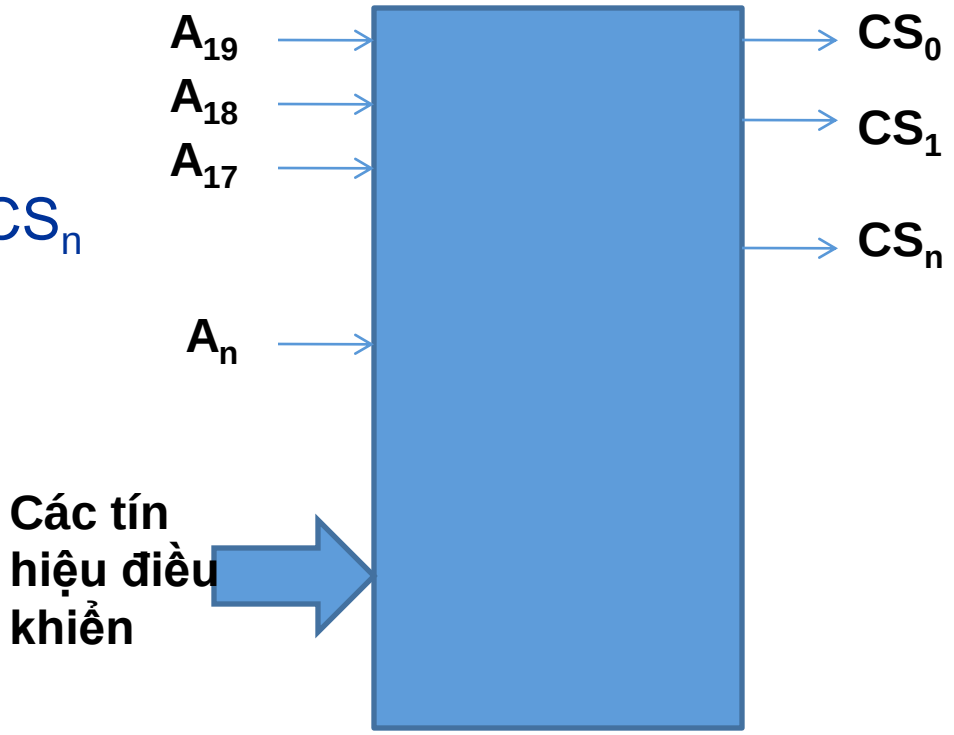
64K x 4 DRAM

Pin(s)	Function
A_0-A_7	Address
DQ_1-DQ_4	Data In/Data Out
RAS	Row Address Strobe
CAS	Column Address Strobe
G	Output Enable
W	Write Enable



Giải mã địa chỉ

- Ánh xạ các tín hiệu địa chỉ thành tín hiệu chọn (kính hoạt) chíp nhớ
  - $A_{19}A_{18}..A_n \rightarrow CS_0,CS_1,\ldots,CS_n$
- Giải mã đầy đủ
  - Sử dụng $A_{19}A_{18}..A_0$
- Giải mã rút gọn
  - Sử dụng $A_{19}A_{18}..A_n;n>0$



Các phương pháp thực hiện giải mã

- Mạch lôgic cơ bản AND/OR
- Mạch giải mã tích hợp
- Bộ nhớ ROM

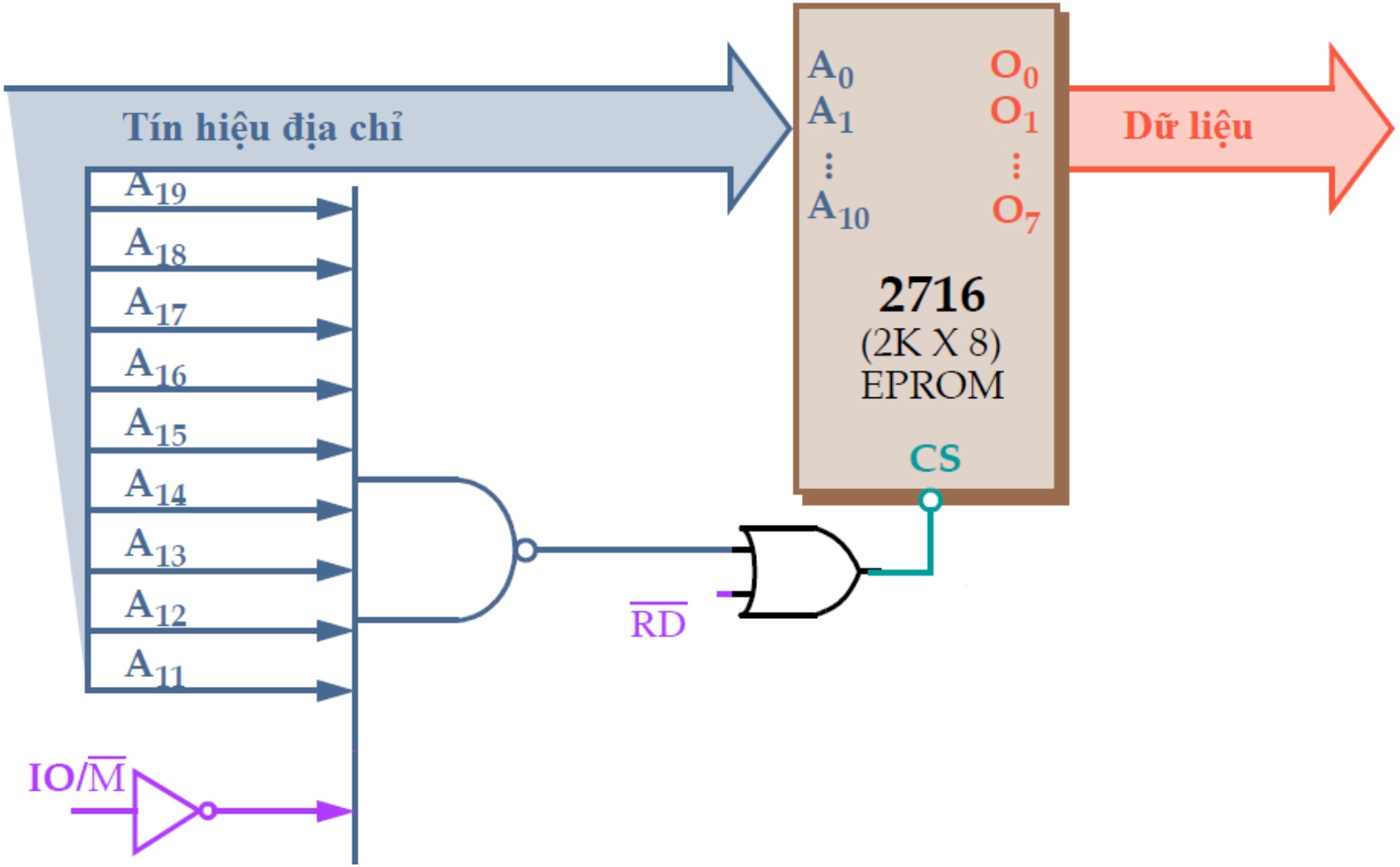
Mạch lôgic cơ bản AND/OR

- Chíp nhớ ROM 2K×8
  - Địa chỉ cấp: FF800–FFFFF
- Tín hiệu địa chỉ dùng kích hoạt chíp
  - $A_{19}…A_{16}A_{15}A_{12}A_{11}$
  - 1111 1111 1000 0000 0000 – 1111 1111 1111 1111 1111

NOT AND

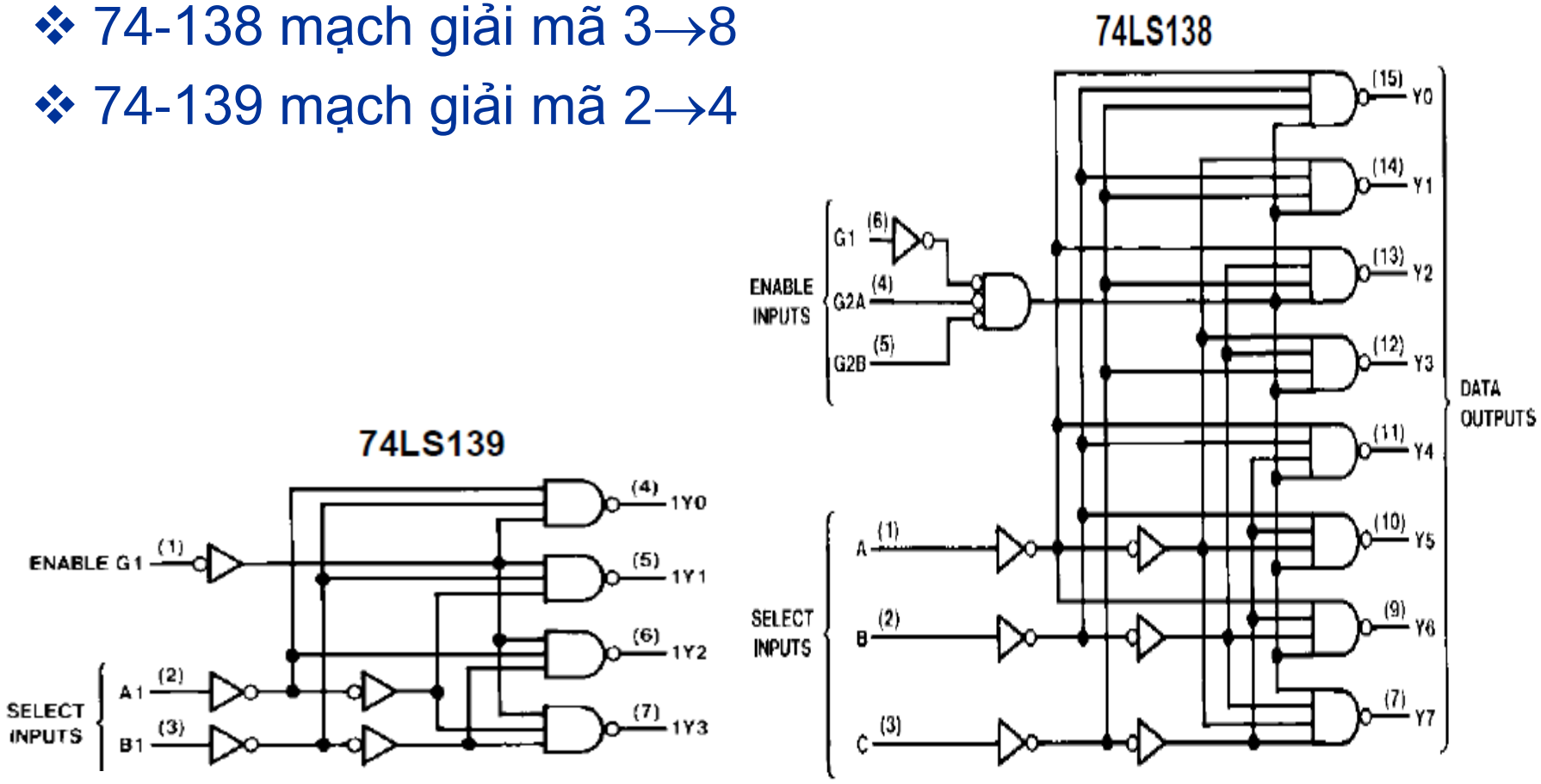
- CS = RD OR NOT($A_{19}…A_{16}A_{15}A_{12}A_{11}$)

Mạch lôgic cơ bản AND/OR

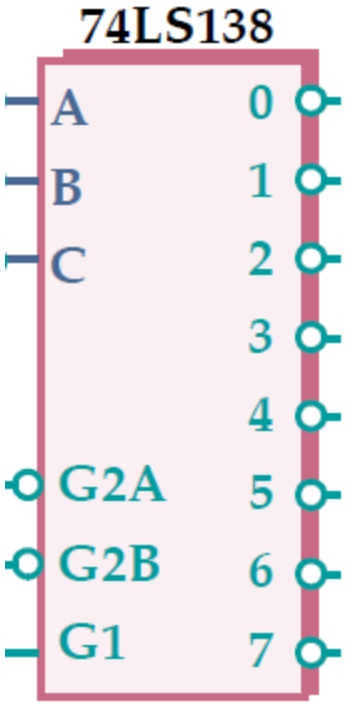


Mạch giải mã tích hợp

- 74-138 mạch giải mã 3→8
- 74-139 mạch giải mã 2→4



Bảng dữ liệu 74-138



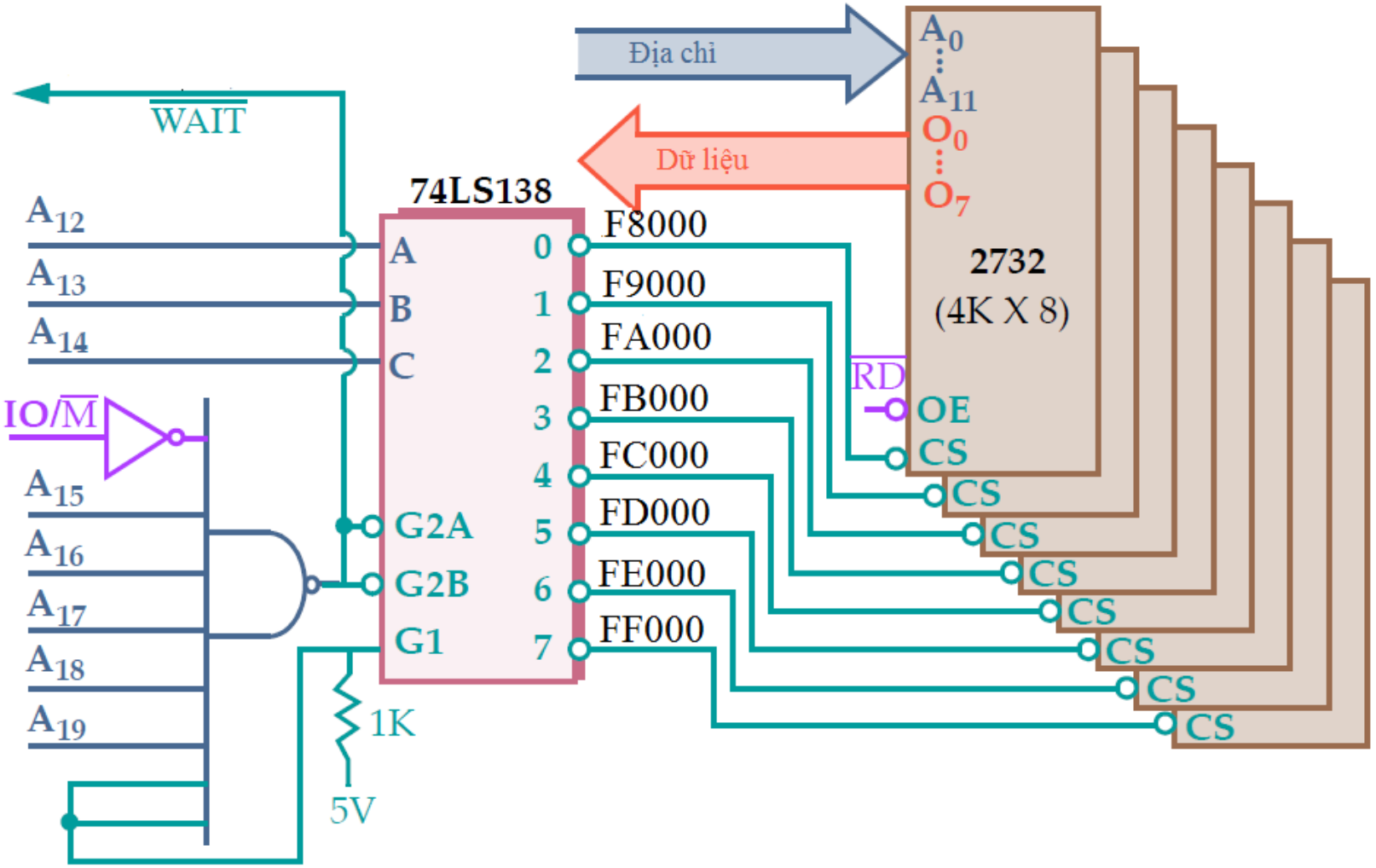
Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Mạch giải mã dùng 74-138

- Bộ giải mã không gian nhớ 32K×8
  - Địa chỉ F8000H-FFFFFH
  - A_0-A_{11}: Tín hiệu địa chỉ của chíp nhớ
  - A_{12}-A_{19}: Sinh ra tín hiệu kích hoạt chíp nhớ

	A_{19}-A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
F8	1111	1	0	0	0
F9	1111	1	0	0	1
FA	1111	1	0	1	0
FB	1111	1	0	1	1
FC	1111	1	1	0	0
FD	1111	1	1	0	1
FE	1111	1	1	1	0
FF	1111	1	1	1	1

Giải mã dùng 74-138

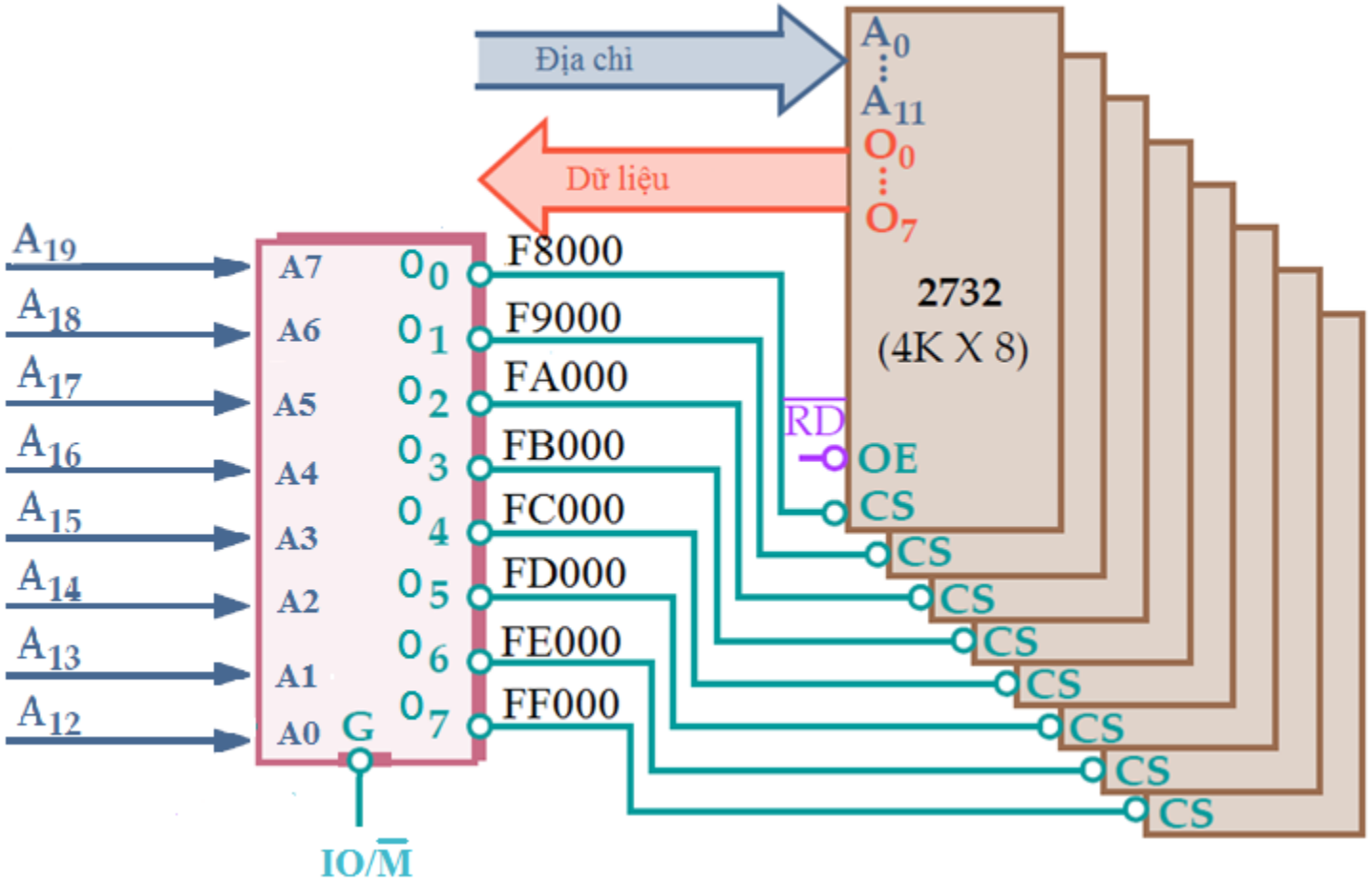


Giải mã dùng ROM

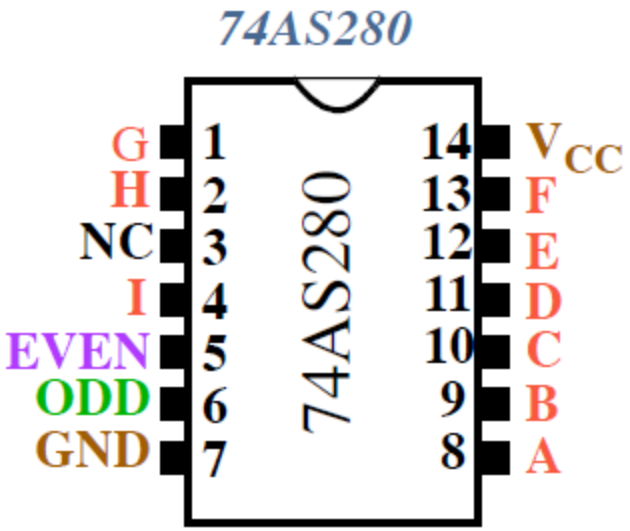
- Từ điều khiển kích hoạt chíp nhớ được lưu vào ROM địa chỉ 00-F7
  - $O_0=A_7*..A_3*\sim A_2*\sim A_1*\sim A_0$
  - $O_1=A_7*..A_3*\sim A_2*\sim A_1*\ A_0$
  - $O_2=A_7*..A_3*\sim A_2*\ A_1*\sim A_0$
  - $O_3=A_7*..A_3*\sim A_2*\ A_1*\ A_0$
  - $O_4=A_7*..A_3*\ A_2*\sim A_1*\sim A_0$
  - $O_5=A_7*..A_3*\ A_2*\sim A_1*\ A_0$
  - $O_6=A_7*..A_3*\ A_2*\ A_1*\sim A_0$
  - $O_7=A_7*..A_3*\ A_2*\ A_1*\ A_0$
- **Các từ khác trong ROM được đặt về FF**

G	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	$\overline{O_0}$	$\overline{O_1}$	$\overline{O_2}$	$\overline{O_3}$	$\overline{O_4}$	$\overline{O_5}$	$\overline{O_6}$	$\overline{O_7}$
0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1
0	1	1	1	1	1	0	0	1	1	0	1	1	1	1	1	1
0	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Giải mã dùng ROM



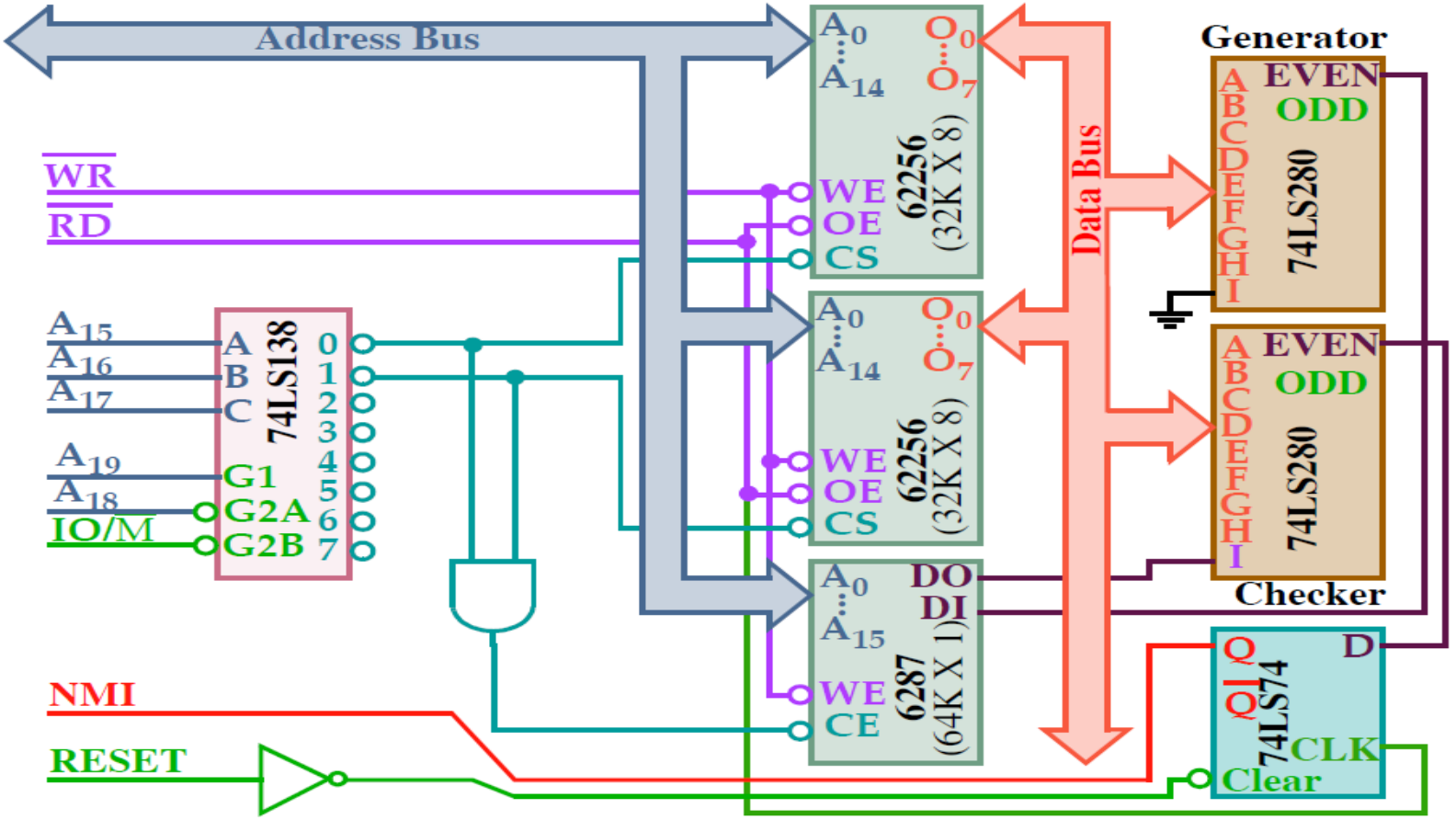
Vi mạch kiểm tra chẵn lẻ 74-280



Số các tín hiệu mức cao A-I	Đầu ra EVEN	Đầu ra ODD
0, 2, 4, 6, 8	H	L
1, 3, 5, 7, 9	L	H

- Tổng số bít 1 luôn chẵn (EVEN) hoặc lẻ (ODD)
  - **1010 0000 → Parity = 1 (lẻ)**
  - **1010 0000 → Parity = 0 (chẵn)**

Bộ nhớ kiểm tra lỗi chẵn lẻ



- Dữ liệu: 2 **đoạn 32K×8**
  - 62256×2
- **Bít chẵn lẻ 64K×1**
  - 6287

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kỹ Thuật Vi Xử Lý

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: Khoa Học Máy Tính- Khoa CNTT1

Học kỳ/Năm biên soạn:2009

NỘI DUNG

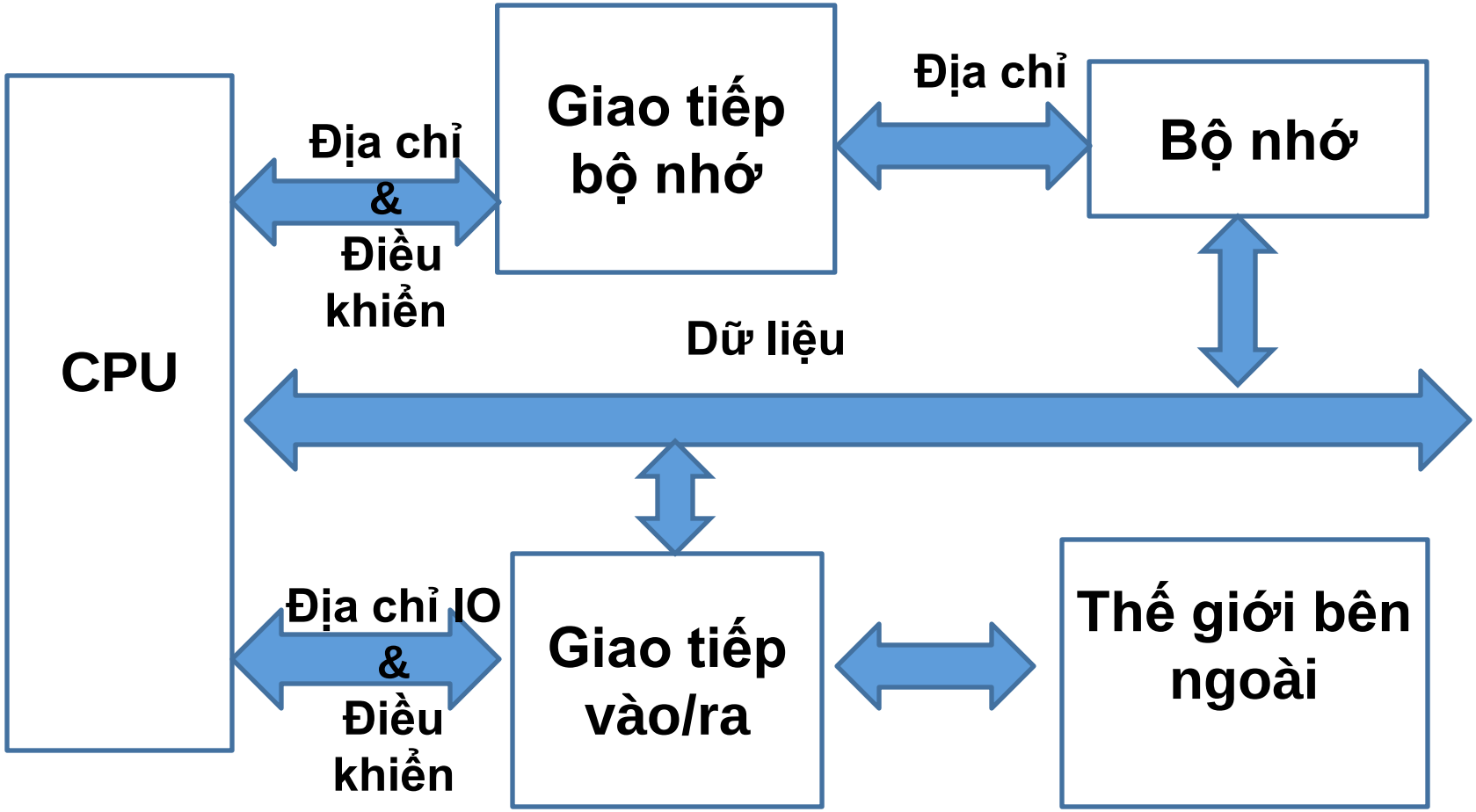
GHÉP NỐI 8088 VỚI THIẾT BỊ VÀO/RA

Giảng viên: TS. Phạm Hoàng Duy
E-mail: phamhduy@gmail.com
Năm biên soạn: 2009

Nội dung

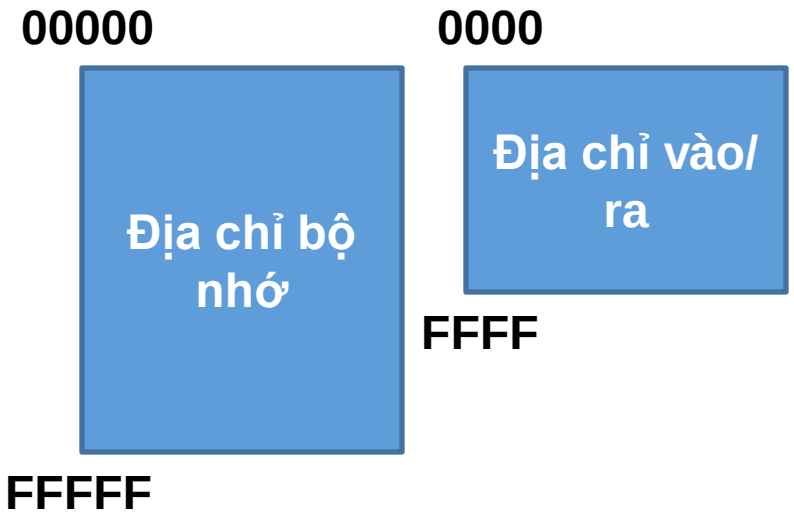
- Phân loại
- Giải mã địa chỉ
- Phương pháp lập trình vào ra

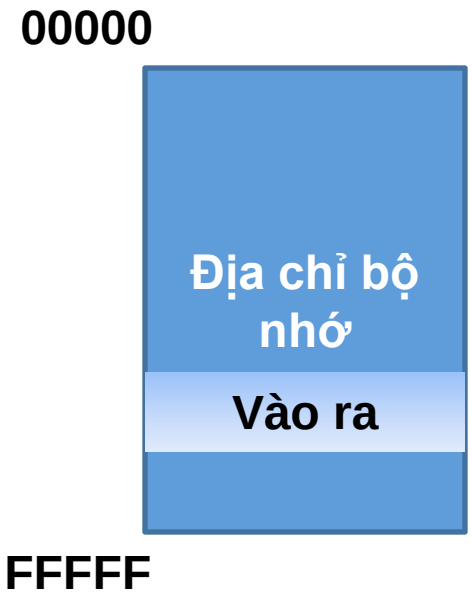
Ghép nối thiết bị vào ra



Phân loại thiết bị vào/ra

- Thiết bị vào/ra có không gian địa chỉ tách biệt
- Thiết bị vào/ra dùng chung không gian địa chỉ với bộ nhớ



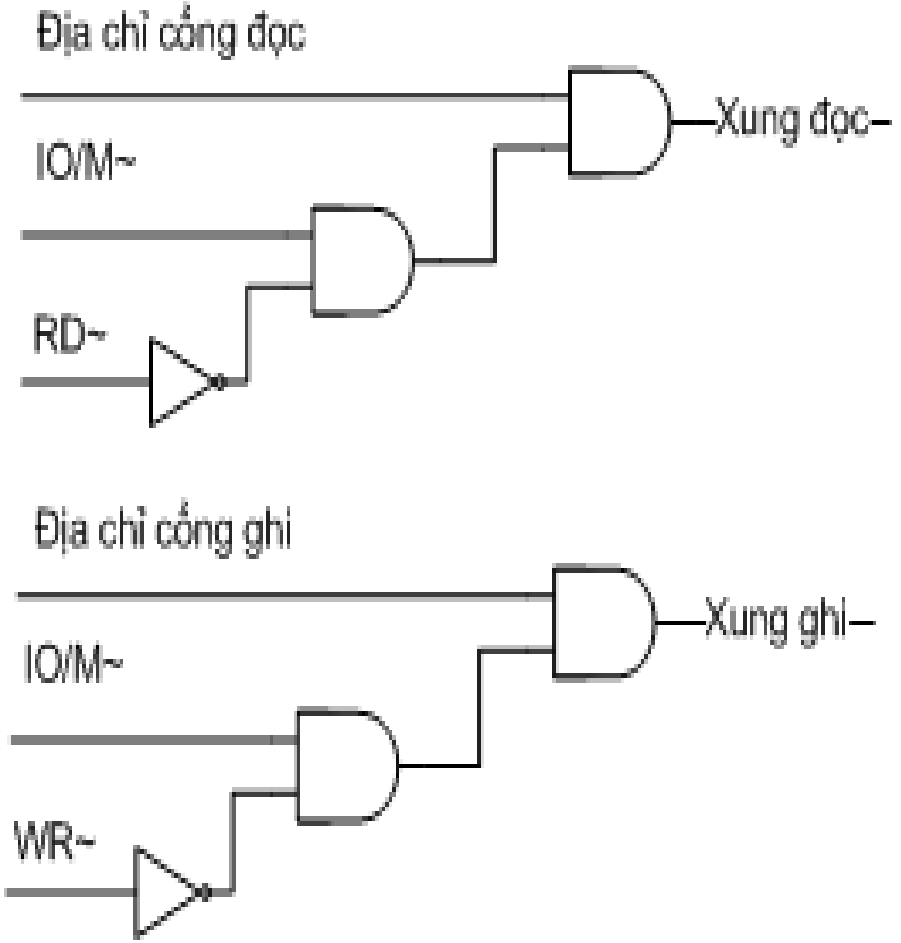


Phân loại thiết bị vào/ra

- Thao tác đọc/ghi dữ liệu
  - IN AX,[Địa chỉ cổng]
  - OUT [Địa chỉ cổng], AX
  - Địa chỉ cổng vào/ra
    - 0000-FFFF: Lưu trong DX
    - 00-FF: địa chỉ trực tiếp
- **Thao tác đọc/ghi dữ liệu**
  - **MOV** [địa chỉ cổng],AX
  - **Đọc: MOV AX,[Địa chỉ cổng]**
  - **Địa chỉ cổng vào/ra**
    - **00000-FFFFF**

Giải mã địa chỉ

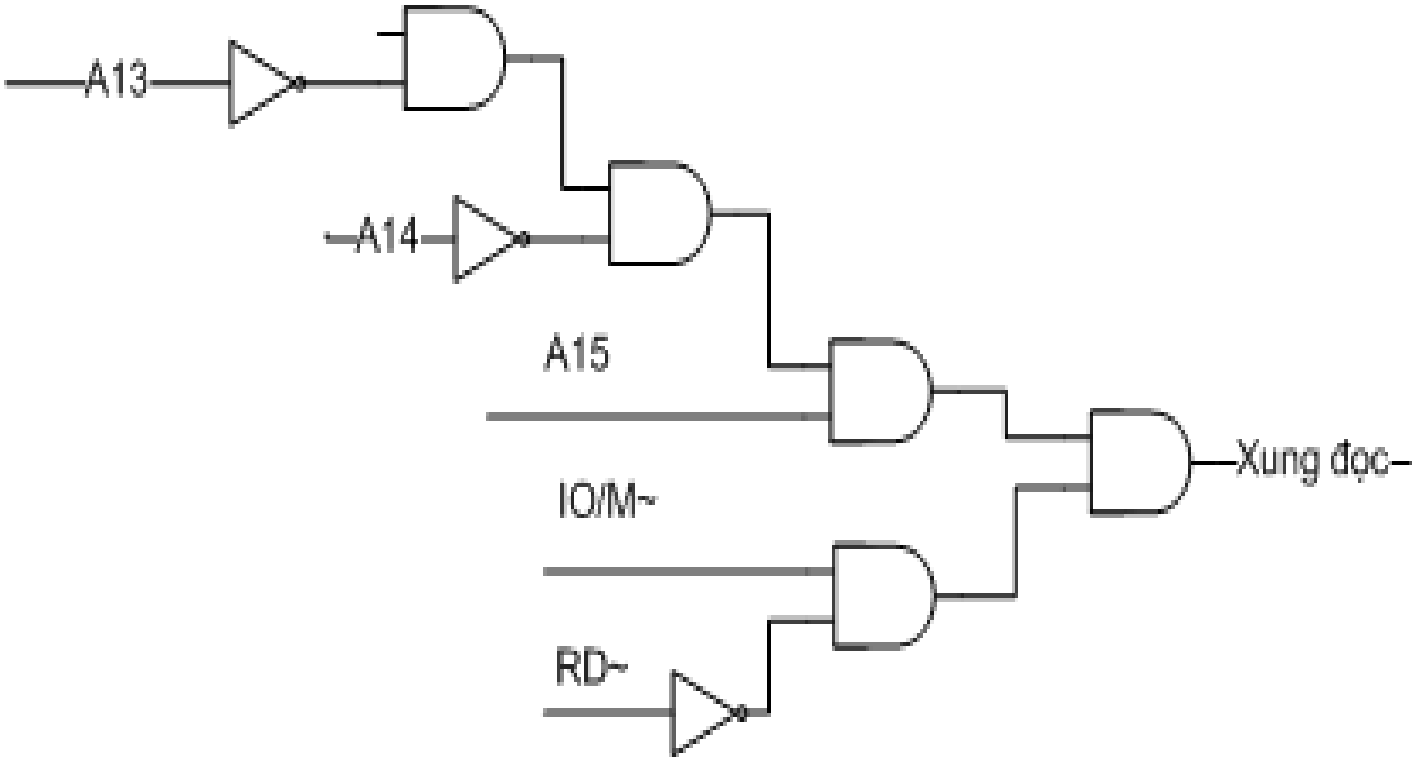
- Tổ hợp các tín hiệu địa chỉ và điều khiển thành xung đọc/ghi
  - Địa chỉ riêng
    - IO + RD~ + A_i…A_j = IN
    - IO + WR~ + A_i…A_j = OUT
  - Địa chỉ chung với bộ nhớ
    - M~ + RD~ + A_i…A_j = IN
    - M~ + WR~ + A_i…A_j = OUT



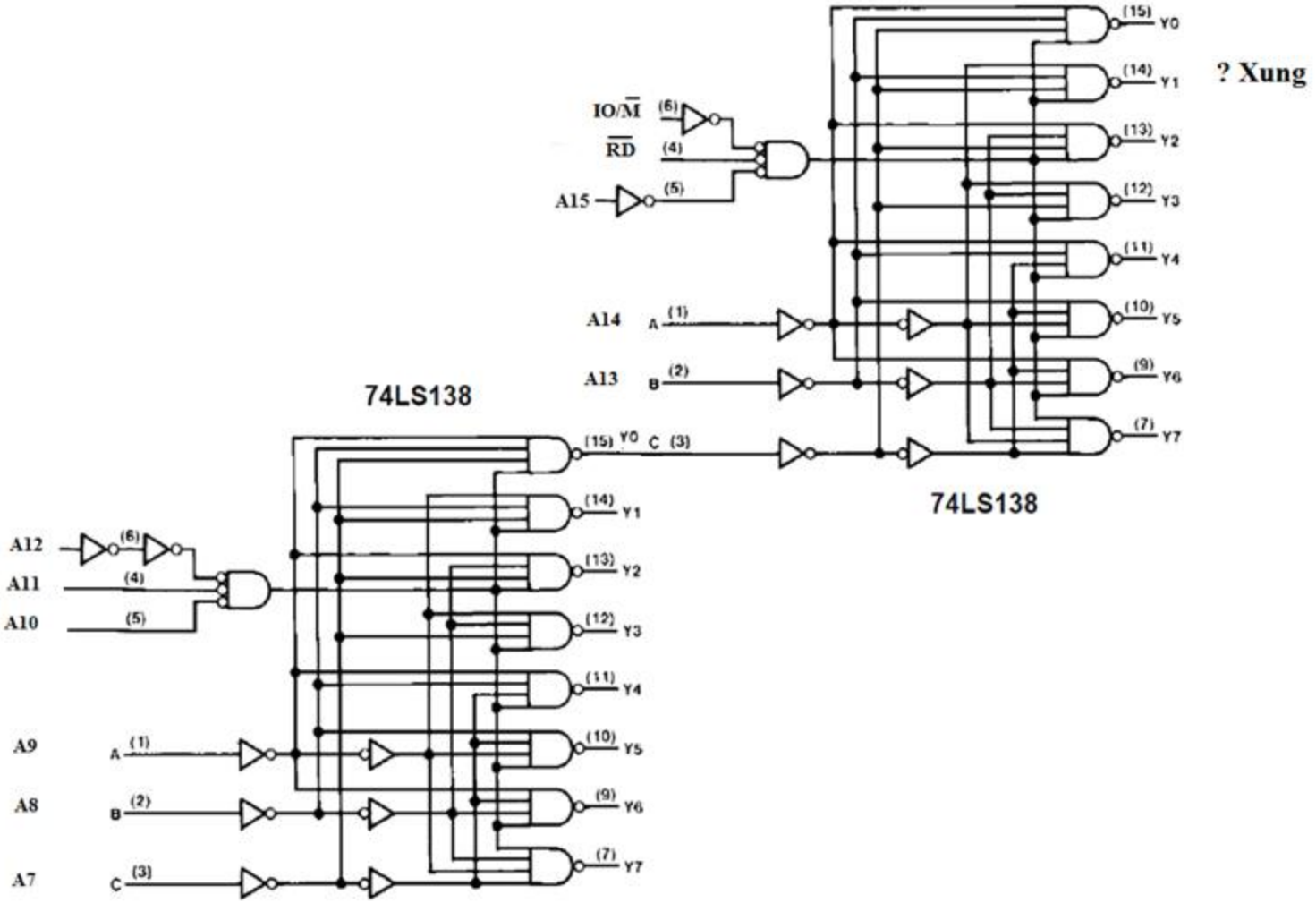
Bài tập

- Xây dựng mạch giải mã cho thiết bị đọc có địa chỉ cổng: 8000H

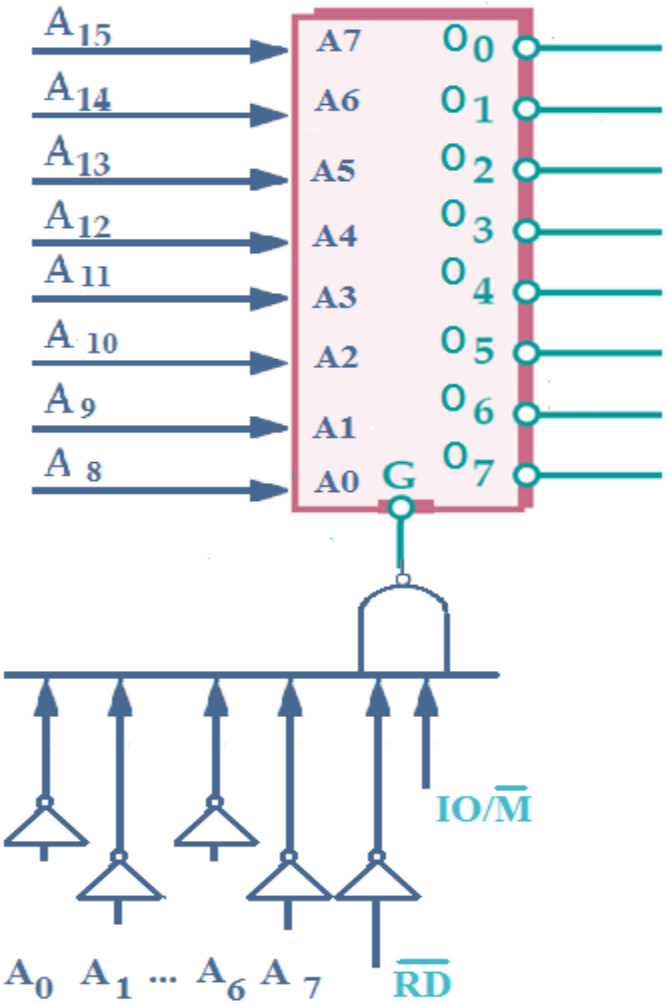
Ví dụ 1



Ví dụ 2


? Xung
74LS138
74LS138
IO/M
RD
A15
A14
A13
A12
A11
A10
A9
A8
A7
A
B
C
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7

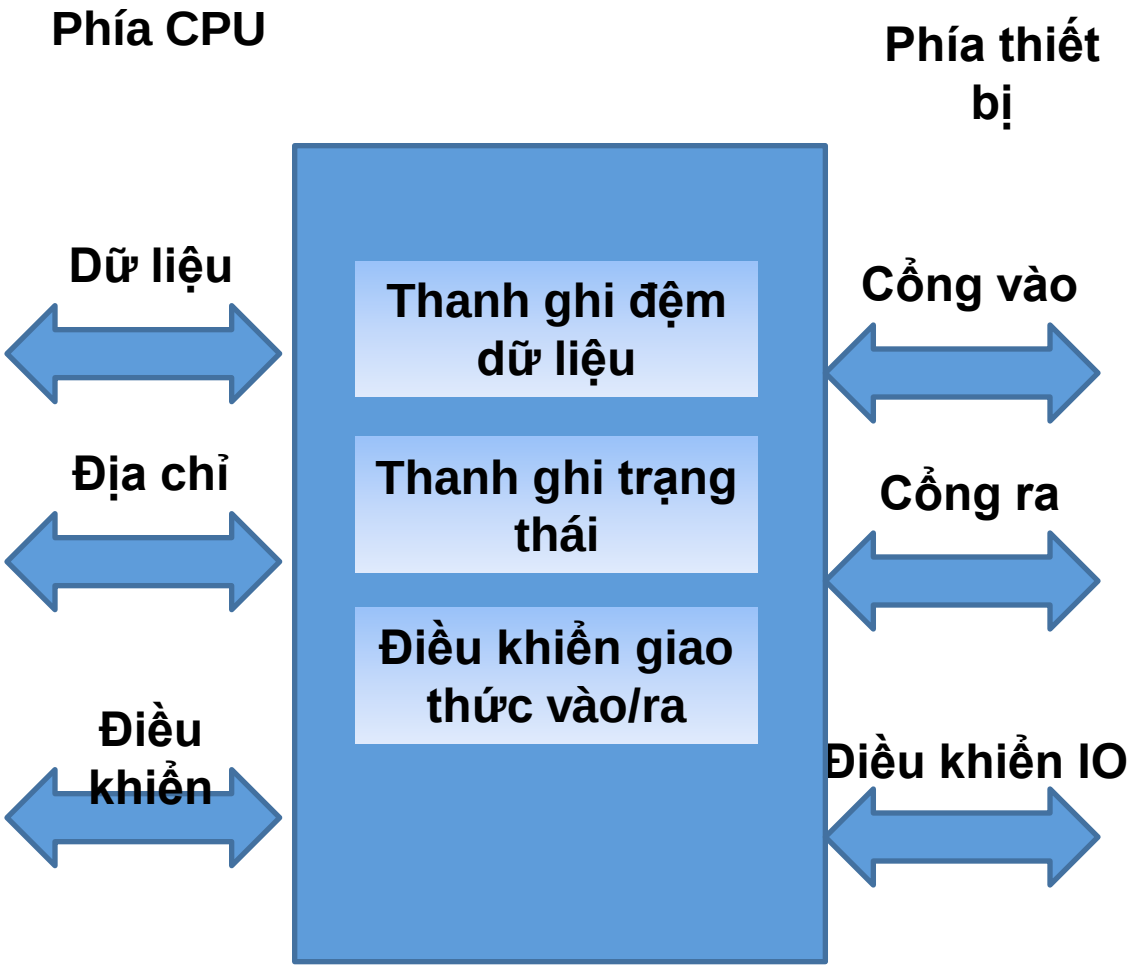
Ví dụ 3



G	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	$\overline{O_0}$	$\overline{O_1}$	$\overline{O_2}$	$\overline{O_3}$	$\overline{O_4}$	$\overline{O_5}$	$\overline{O_6}$	$\overline{O_7}$
0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	1	1	0	1	1	1	1	1	1
0	1	0	0	0	0	0	1	0	1	1	0	1	1	1	1	1
0	1	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	0	0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	0	0	0	1	0	1	1	1	1	1	1	0	1	1
0	1	0	0	0	0	1	1	0	1	1	1	1	1	1	0	1
0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0

Giao tiếp vào ra

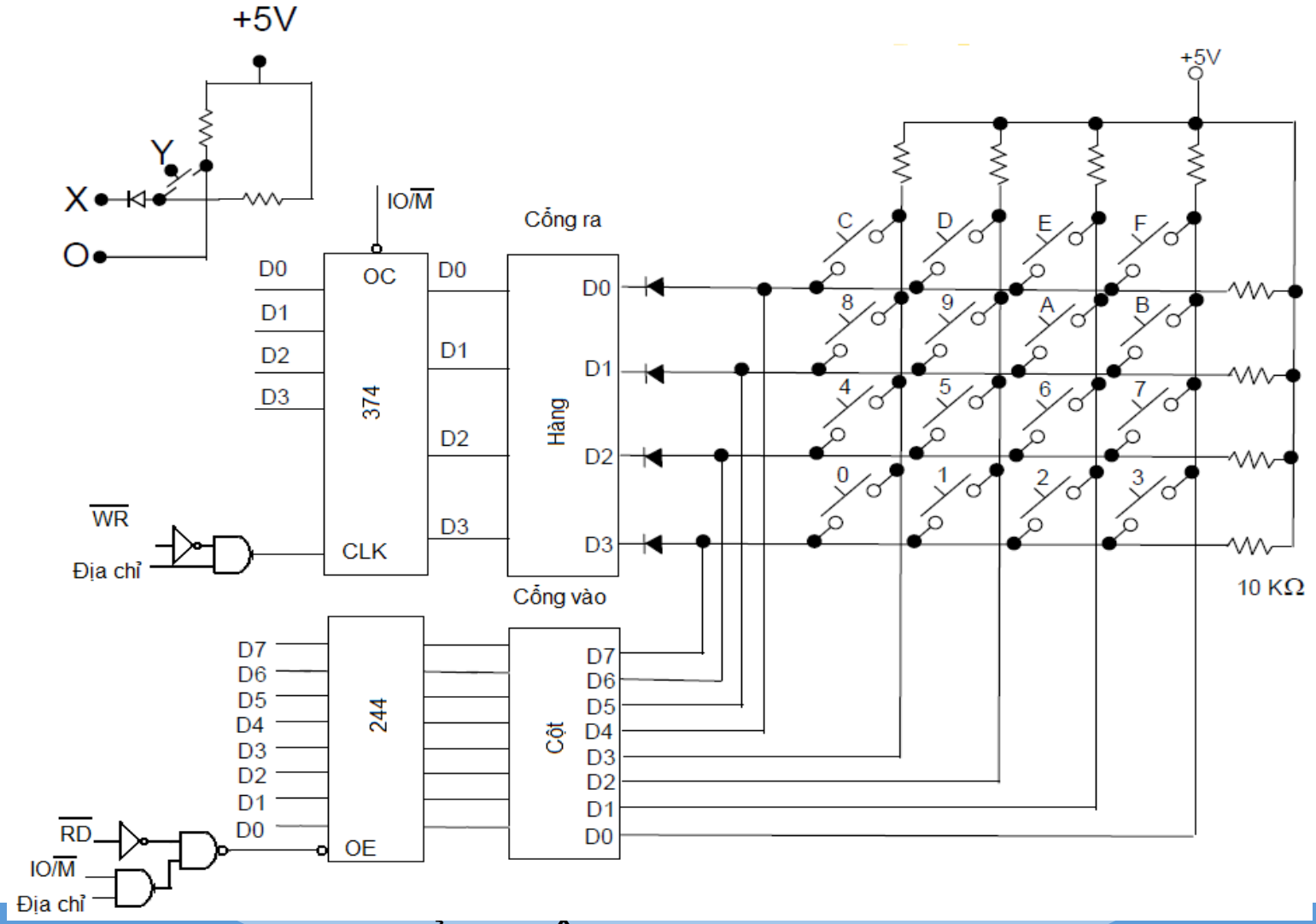
- Đệm dữ liệu
- Thực hiện giao thức điều khiển thiết bị
- Chuyển đổi định dạng dữ liệu
- Phát hiện và sửa lỗi



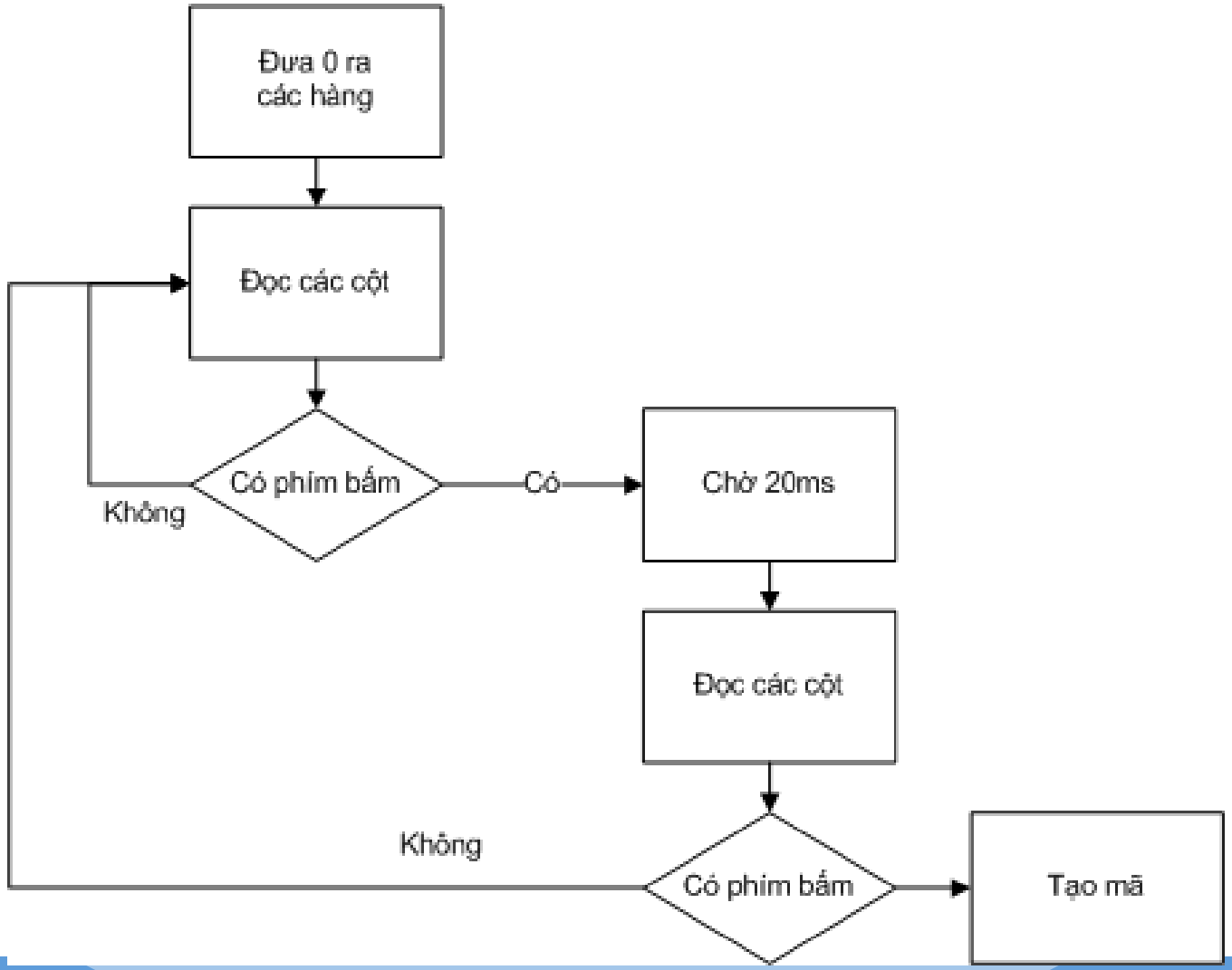
Phương pháp lập trình vào ra

- Vào ra lập trình
  - CPU thăm dò trạng thái thiết bị vào/ra
  - Thực hiện các thao tác đọc/ghi số liệu
- Vào ra sử dụng ngắt
  - Thiết bị vào ra thông báo cho CPU về tình trạng hoạt động
  - CPU thực hiện thao tác đọc/ghi số liệu
- Vào ra trực tiếp bộ nhớ
  - Yêu cầu phần cứng đặc biệt
  - CPU không phải thực hiện thao tác số liệu

Ví dụ ghép nối bàn phím



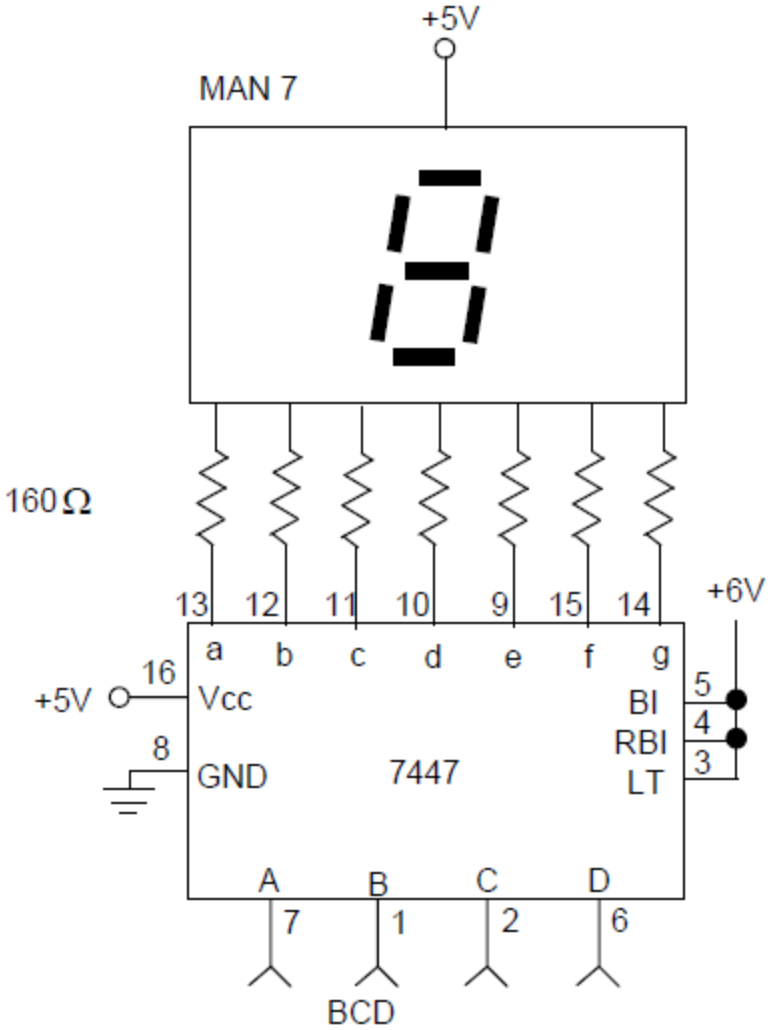
Chương trình đọc bàn phím



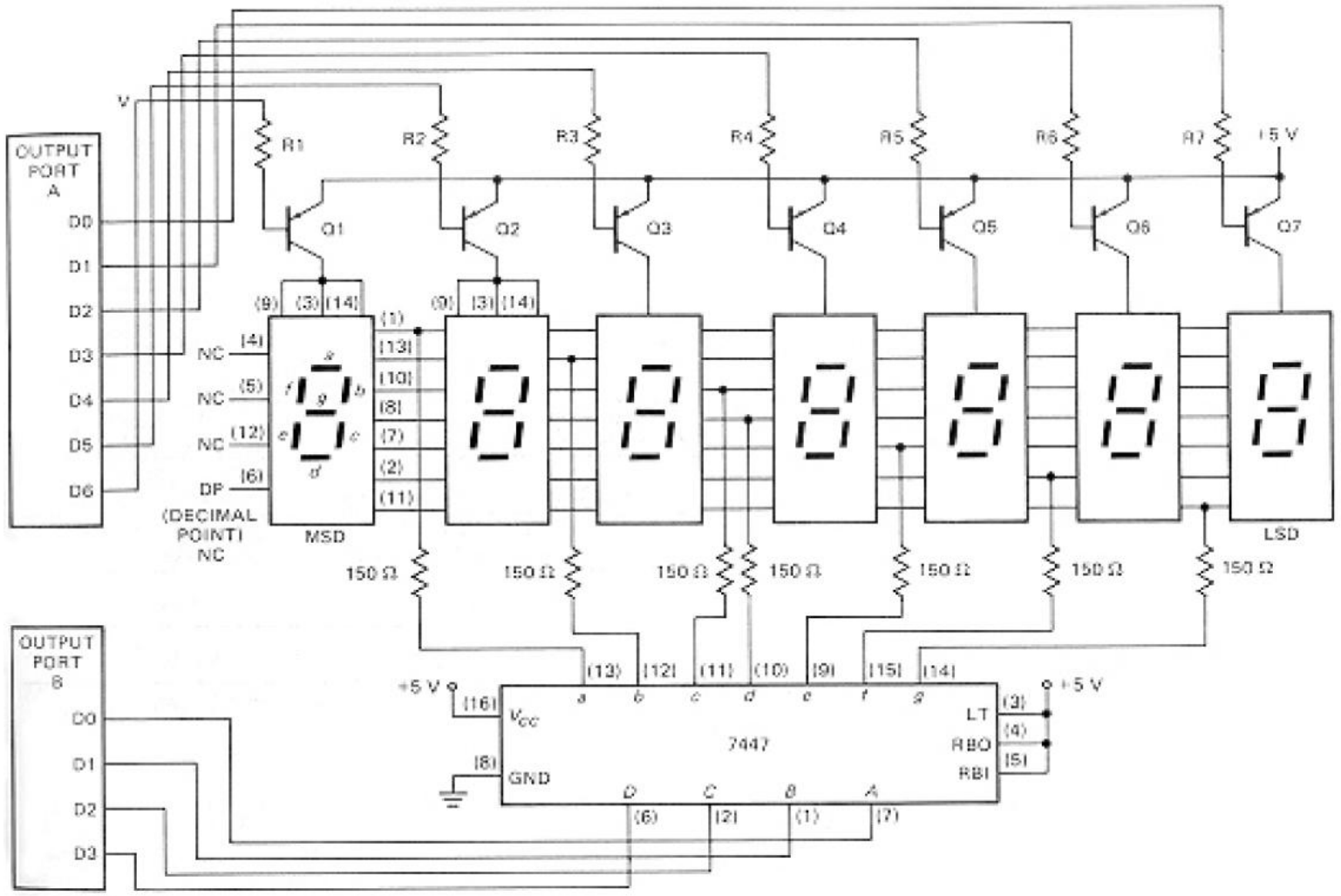
Chương trình đọc bàn phím

- Biết cổng ghi A
- Biết cổng đọc B
- Trễ ~4000 NOP
- Chương trình =??

Ghép nối hiển thị số

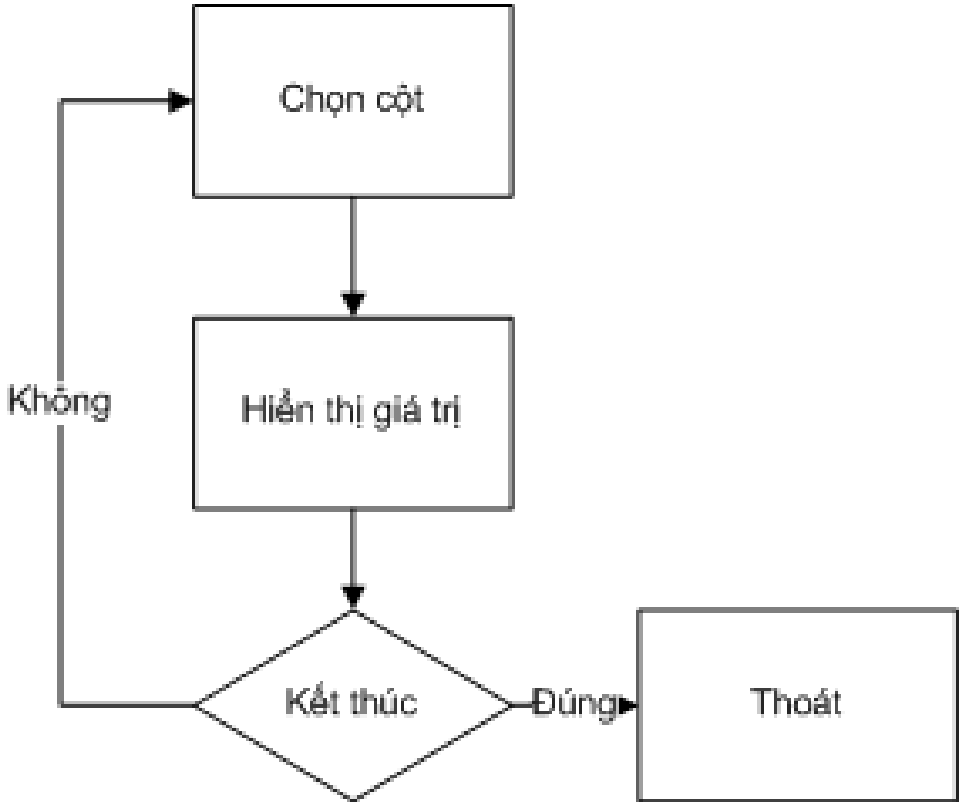


Ghép nối hiển thị số

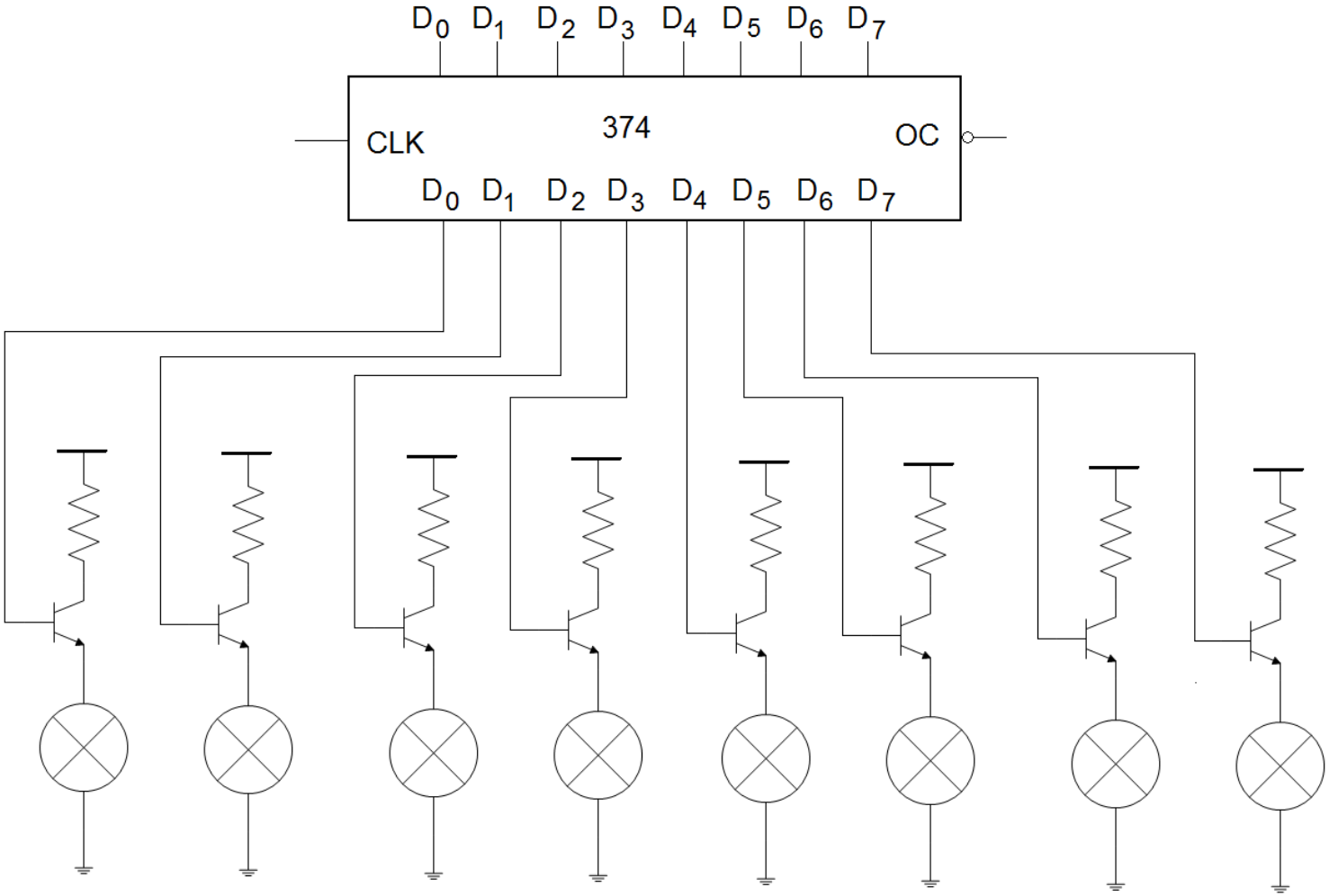


Ghép nối hiển thị số

- Cổng A: chọn số
- Cổng B: giá trị
- Chương trình!!!



Điều khiển đèn báo hiệu



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kỹ Thuật Vi Xử Lý

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: Khoa Học Máy Tính- Khoa CNTT1

Học kỳ/Năm biên soạn:2009

NỘI DUNG

GHÉP NỐI 8088 VỚI BỘ ĐIỀU KHIỂN NGẮT

Giảng viên: TS. Phạm Hoàng Duy
E-mail: phamhduy@gmail.com
Năm biên soạn: 2009

Nội dung

- Khái niệm ngắt
- Xử lý ngắt
- PIC 8259A

Ngắt

- Tạm dừng thao tác hiện thời của CPU để chuyển sang thao tác khác
  - Trao đổi dữ liệu với thiết bị ngoại vi
  - Báo lỗi
  - Phục vụ yêu cầu khẩn

Phân loại ngắt

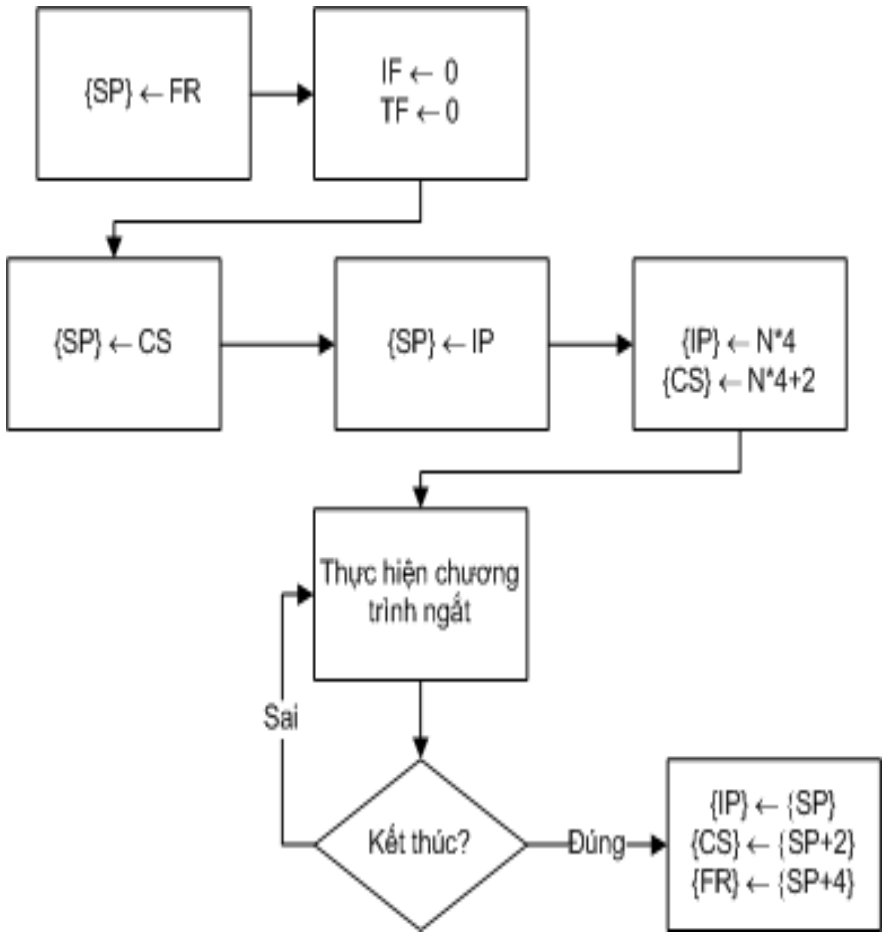
- Ngắt cứng: sinh ra do các tín hiệu INTR hay NMI
  - Ngắt che được: chịu tác động của cờ cho phép ngắt
  - Ngắt không che được
- Ngắt mềm: sinh ra do câu lệnh INT
- Ngắt tự động (ngoại lệ): sinh do thực hiện các lệnh của CPU như chia 0, đặt cờ ngắt, ..

Một số lệnh liên quan ngắt

- CLI: Xóa cờ ngắt
- STI: Đặt cờ ngắt
- INT XX: Gọi ngắt mềm số XX
- IRET: Câu lệnh trờ về khi kết thúc chương trình xử lý ngắt
- HLT: Treo CPU cho đến khi có ngắt hoặc khởi động lại

Quá trình xử lý ngắt

1. Lưu thanh ghi cờ
2. Cấm ngắt
3. Lưu đoạn lệnh
4. Lưu con trỏ lệnh
5. Nạp đoạn lệnh và con trỏ lệnh mới
6. Thực hiện chương trình ngắt
7. Khôi phục lại các thanh ghi trước khi ngắt



Xử lý yêu cầu ngắt

- Các ngắt cứng dùng để quản lý các thiết bị ngoại vi, đặc biệt hiệu quả đối với các thao tác vào/ra
- Tín hiệu ngắt không che được NMI dùng trong tình trạng khẩn cấp như lỗi phần cứng
- Tín hiệu ngắt thông thường INTR dùng để điều khiển thiết bị, CPU có thể chậm trễ khi xử lý tín hiệu này

Xử lý ngắt

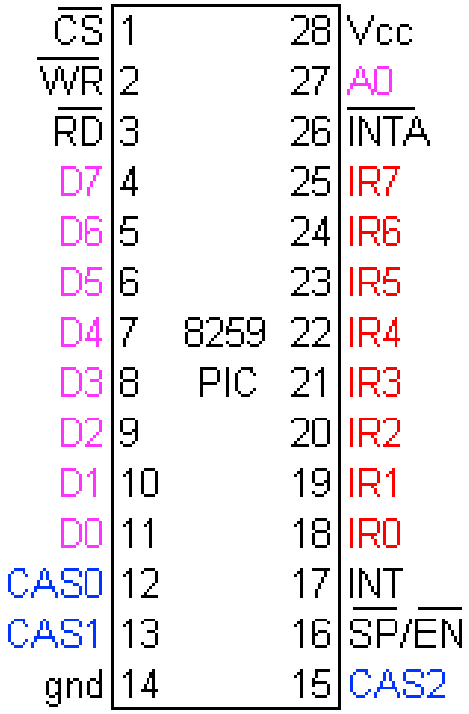
- Khi nhiều tín hiệu ngắt đồng thời xảy ra, tín hiệu ngắt nào có độ ưu tiên cao nhất sẽ được đưa tới CPU

Kiểu ngắt	Độ ưu tiên
Ngắt tự động	Cao nhất
Ngắt không che được NMI	
Ngắt che được INTR	
Ngắt chạy từng lệnh	Thấp nhất

Bảng véc tơ ngắt (PC BIOS)

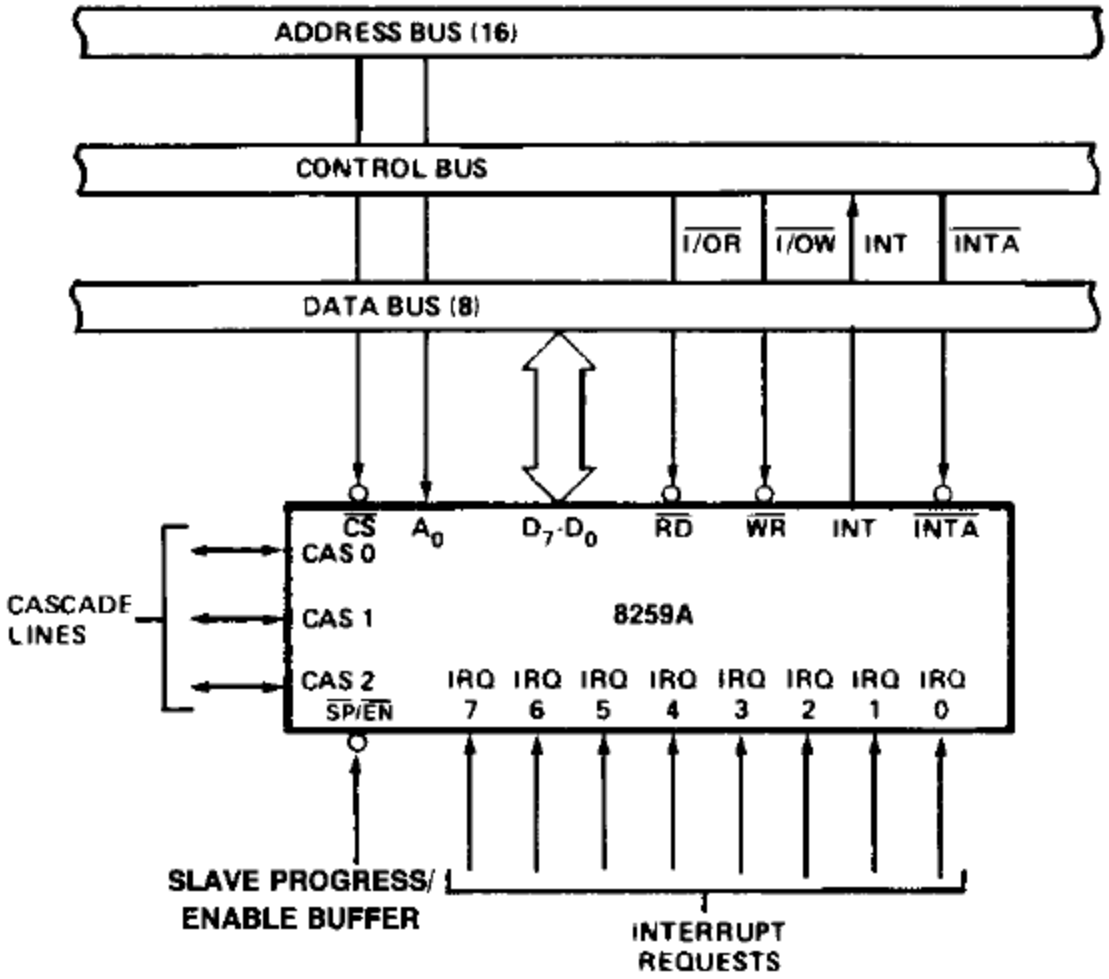
Số thứ tự	Chức năng
0H	Lỗi phép chia
1H	Chạy từng bước
2H	NMI
3H	Dừng (break point)
8H	Ngắt đồng hồ (thời gian)
10H	Ngắt dùng điều khiển màn hình
13H	Ngắt đọc ghi đĩa
16H	Ngắt điều khiển bàn phím
21H	Ngắt của DOS

Bộ điều khiển ngắt PIC-8259A

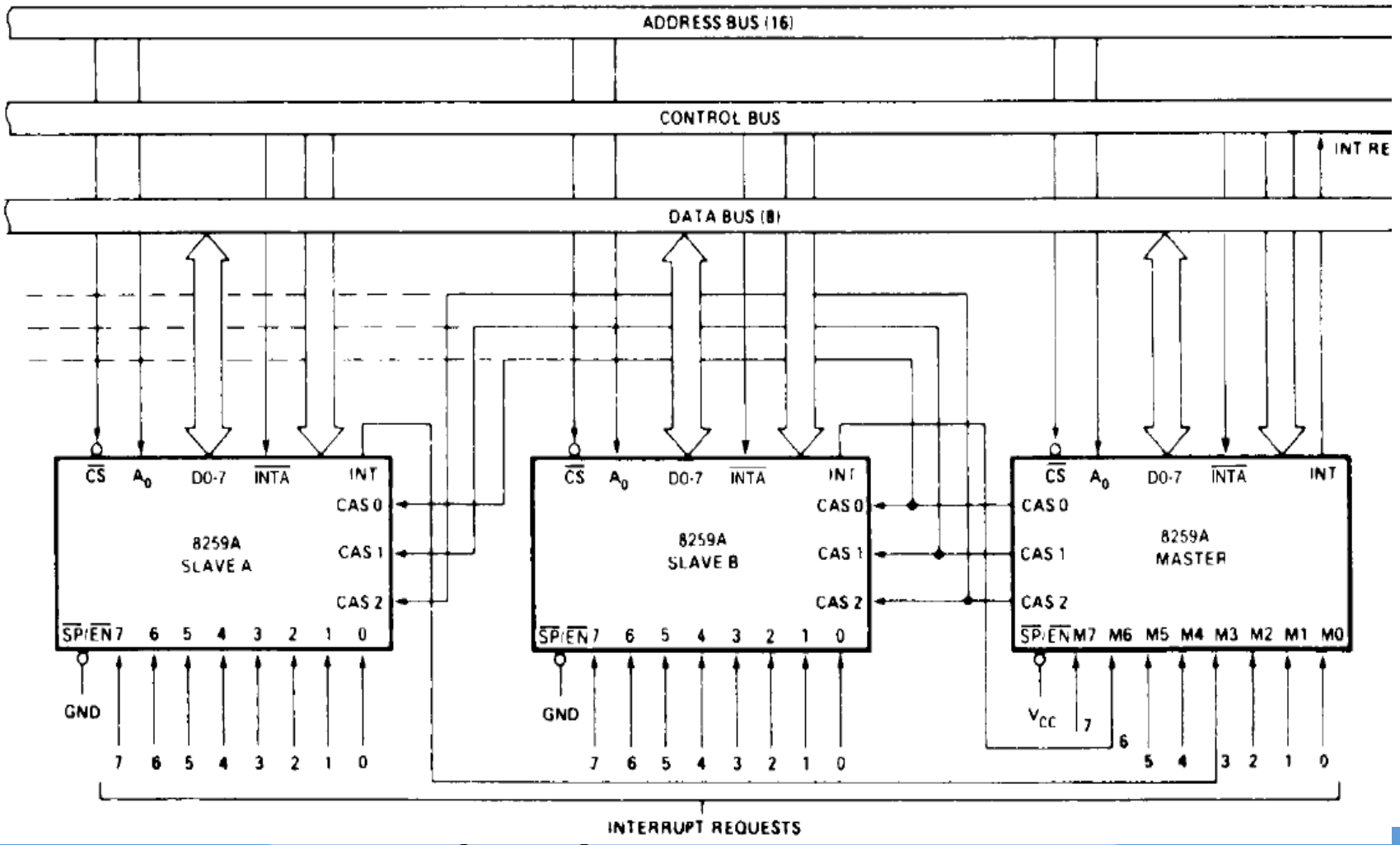


D0-D7	Dữ liệu
RD,WR	Đọc, Ghi (mức thấp)
A0	Địa chỉ thanh ghi
CS	Chọn chip
CAS0-2	Ghép tầng với PIC khác
SP	Xác định PIC chủ (master SP=1) thợ (slave SP=0)
EN	Mở đệm dữ liệu
INT	Yêu cầu ngắt
INTA	Chấp nhận ngắt

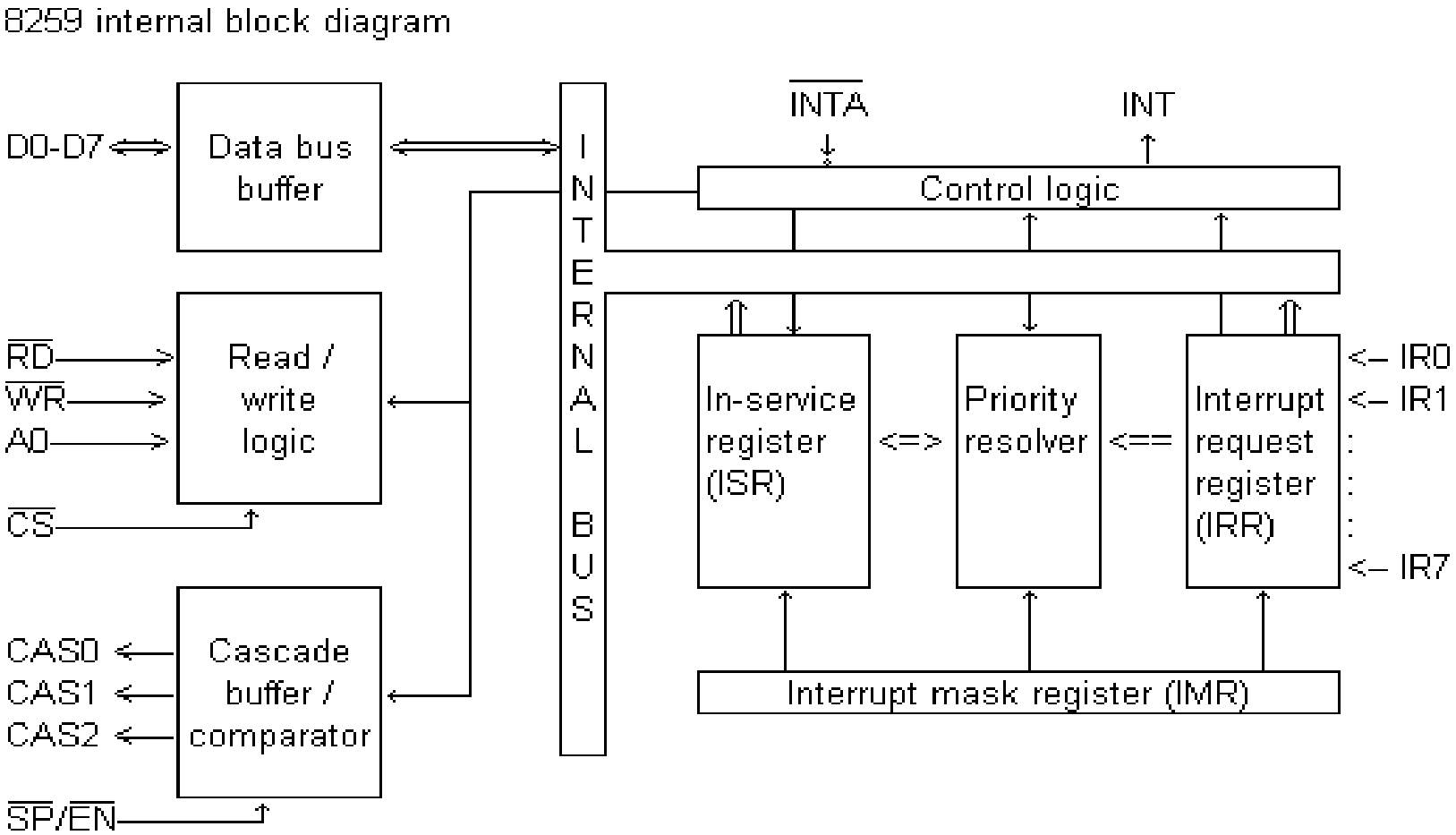
Ghép nối


ADDRESS BUS (16)
CONTROL BUS
I/OR
I/OW
INT
INTA
DATA BUS (8)
CS
A0
D7-D0
RD
WR
INT
INTA
CAS 0
CAS 1
CAS 2
SP/EN
8259A
CASCADE LINES
IRQ 7
IRQ 6
IRQ 5
IRQ 4
IRQ 3
IRQ 2
IRQ 1
IRQ 0
SLAVE PROGRESS/ ENABLE BUFFER
INTERRUPT REQUESTS

Ghép nối


ADDRESS BUS (16)
CONTROL BUS
INT RE
DATA BUS (8)
CS
A0
D0-7
INTA
INT
CAS 0
CAS 1
CAS 2
8259A
SLAVE A
SP/EN 7 6 5 4 3 2 1 0
GND
8259A
SLAVE B
8259A
MASTER
SP/EN M7 M6 M5 M4 M3 M2 M1 M0
VCC
INTERRUPT REQUESTS

Kiến trúc 8259A



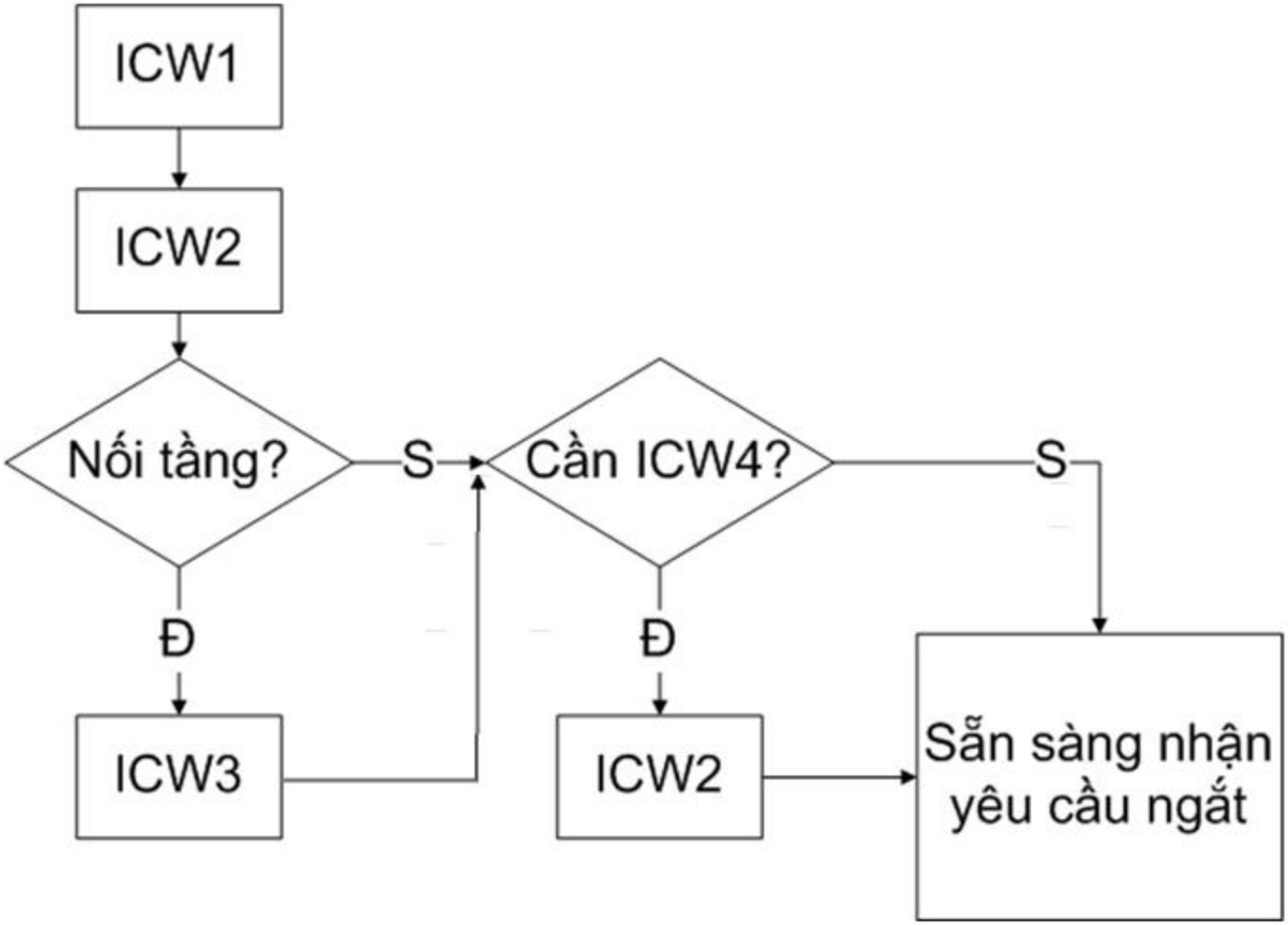
Kiến trúc 8259

- **Data bus buffer**: đệm dữ liệu (3 trạng thái)
- **R/W logic**: đọc/ghi các thông tin điều khiển và trạng thái
- **IMR**: ghi nhớ mặt nạ ngắt với các yêu cầu ngắt
- **IRR**: Lưu trạng thái hiện thời của các yêu cầu ngắt
- **Priority resolver**: xác định thứ tự ưu tiên của các yêu cầu ngắt
- **ISR**: lưu giữ các yêu cầu ngắt được phục vụ
- **Cascade buffer/comparator**: giao tiếp giữa PIC chủ/thợ

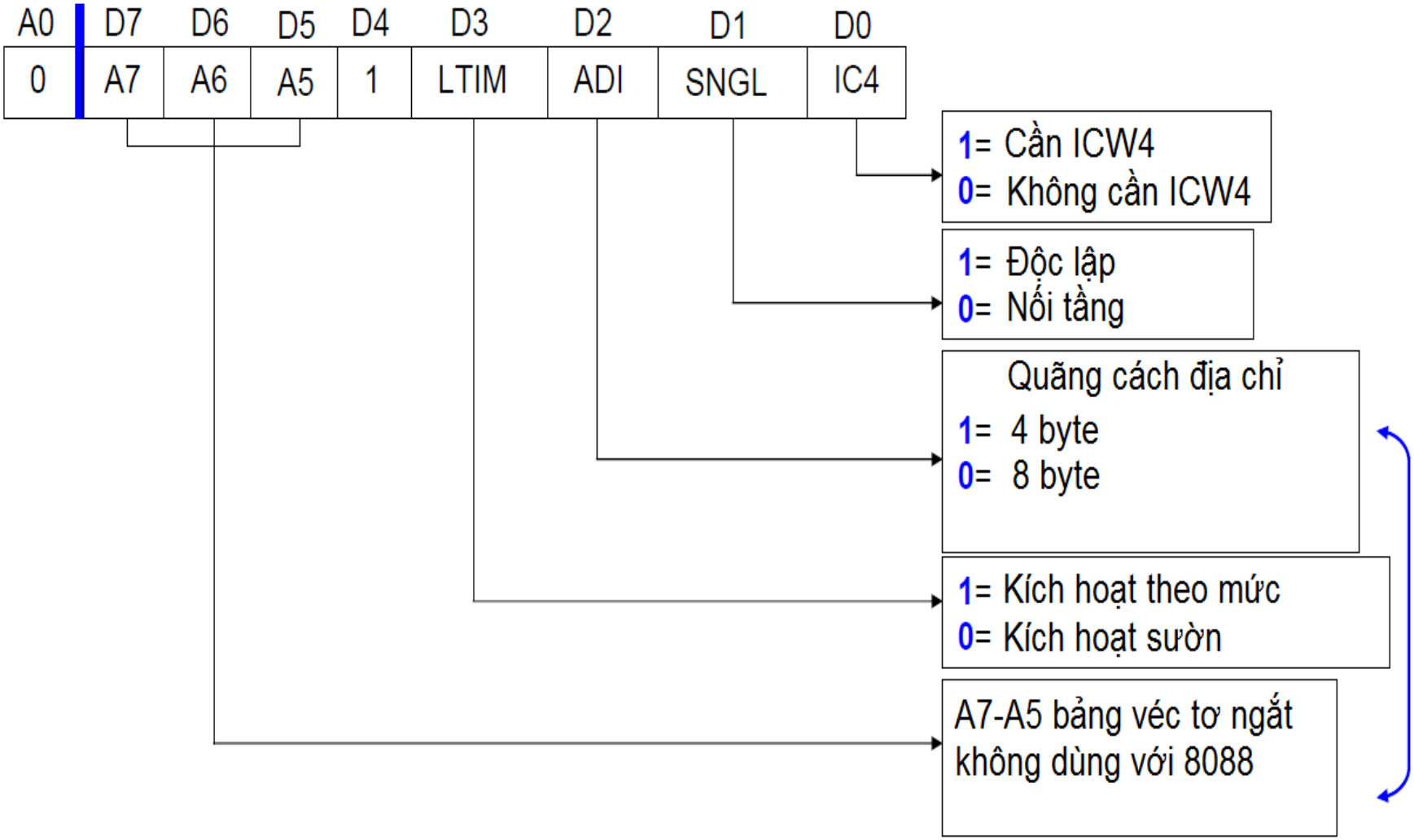
Lập trình PIC-8259A

- PIC được lập trình thông qua nạp các giá trị thích hợp cho 7 thanh ghi (ô nhớ trong) của 8259A:
  - 4 từ khởi tạo ICW
  - 3 từ điều khiển hoạt động OCW
- ICW xác lập chế độ hoạt động PIC-8259A
- OCW điều khiển 8259A hoạt động ở các chế độ khác nhau

Xác lập chế độ làm việc

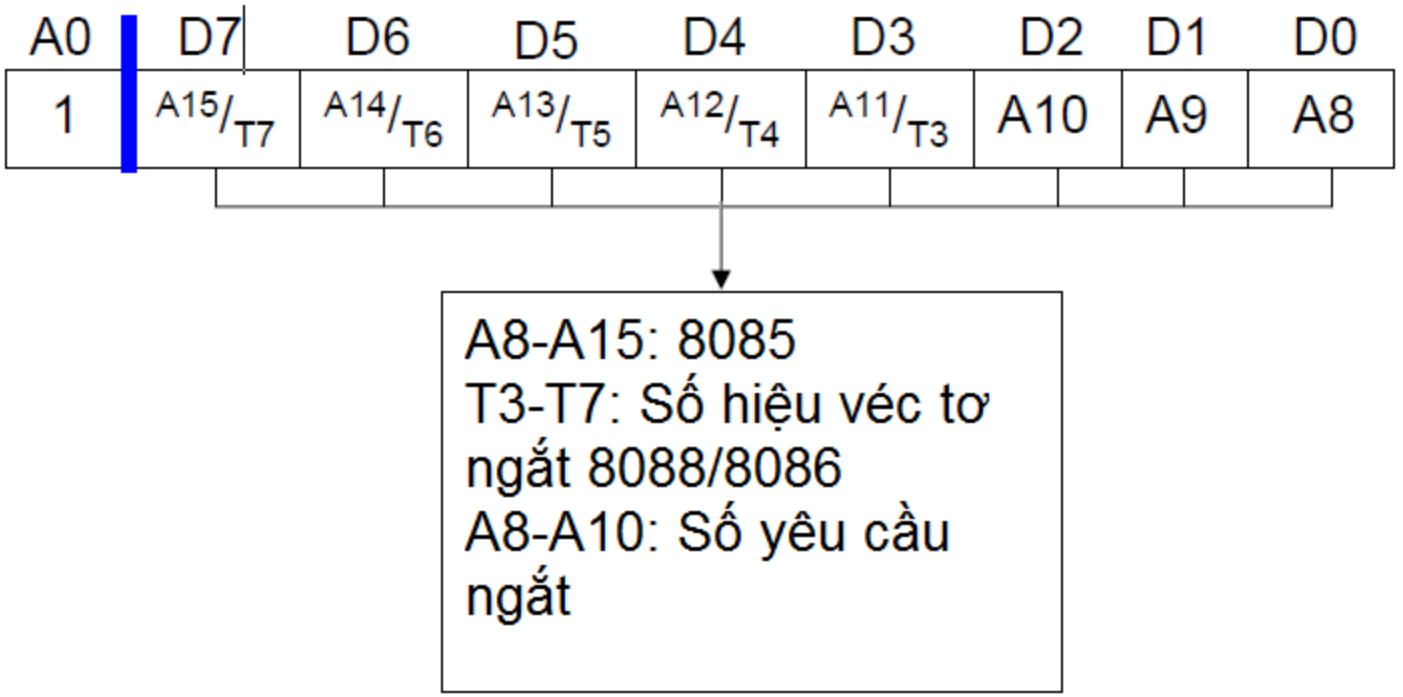


ICW1


A0 D7 D6 D5 D4 D3 D2 D1 D0
0 A7 A6 A5 1 LTIM ADI SNGL IC4
1= Cần ICW4
0= Không cần ICW4
1= Độc lập
0= Nối tầng
Quãng cách địa chỉ
1= 4 byte
0= 8 byte
1= Kích hoạt theo mức
0= Kích hoạt sườn
A7-A5 bảng véc tơ ngắt
không dùng với 8088

ICW2

❖ Xác định số hiệu ngắt



ICW2 với 8088/8086

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

ICW3

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0

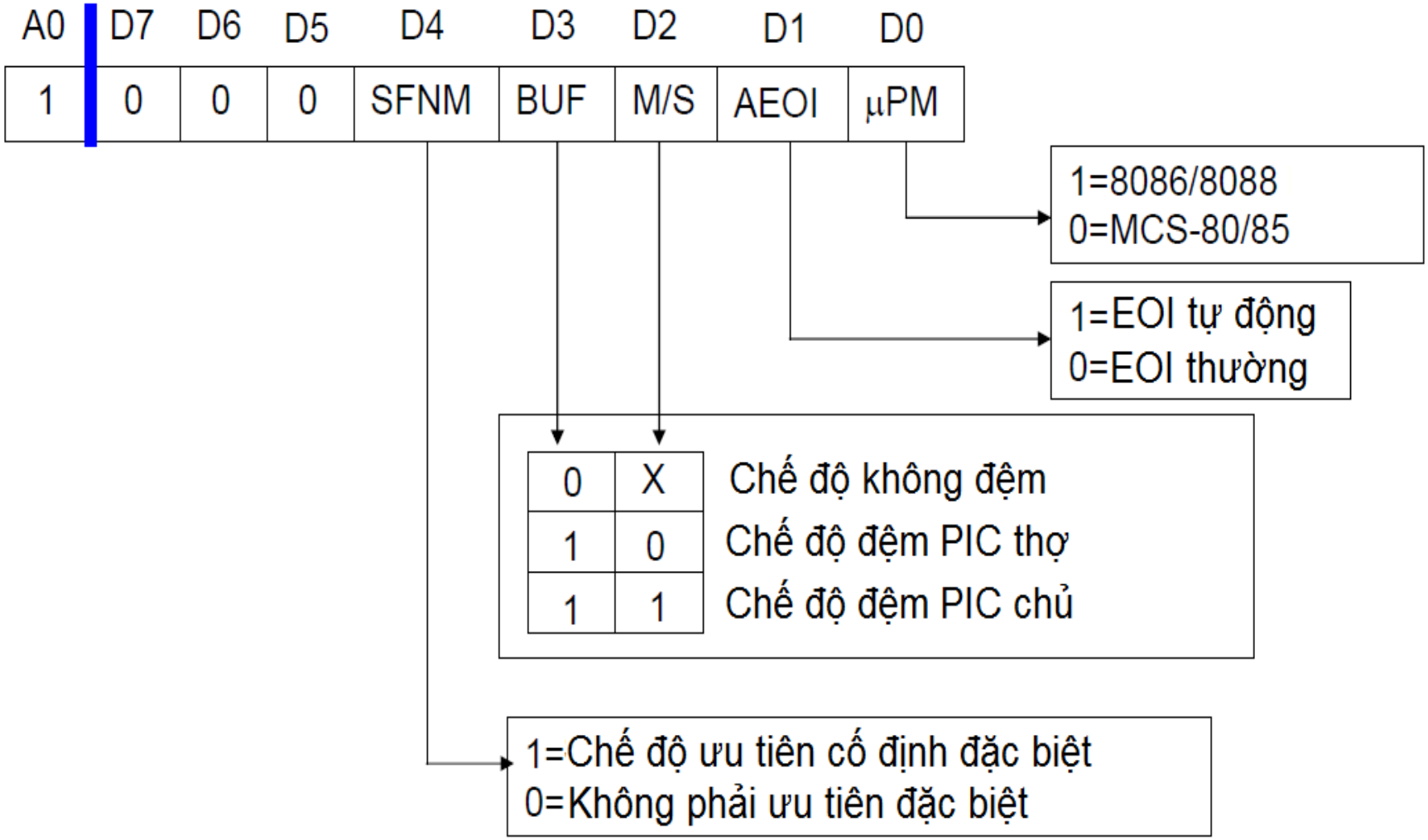
1= IR nối với PIC thợ
0= IR không nối với PIC thợ

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	ID2	ID1	ID0

Mã hóa số hiệu PIC thợ

0	1	2	3	4	5	6	7
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1

ICW4

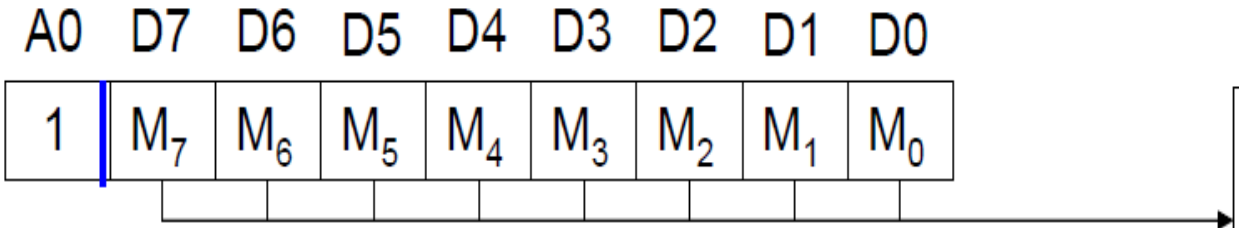

A0 D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 0 SFNM BUF M/S AEOI µPM
1=8086/8088
0=MCS-80/85
1=EOI tự động
0=EOI thường
0 X Chế độ không đệm
1 0 Chế độ đệm PIC thợ
1 1 Chế độ đệm PIC chủ
1=Chế độ ưu tiên cố định đặc biệt
0=Không phải ưu tiên đặc biệt

Ví dụ

- 8259 kết nối với 8088, hoạt động ở chế độ độc lập, yêu cầu ngắt kích hoạt bằng mức, có đệm dữ liệu, sử dụng chế độ ưu tiên bình thường. Xác định từ khởi tạo cho 8259?

Từ điều khiển hoạt động OCW

OCW1: Thiết lập và đọc trạng thái yêu cầu ngắt



OCW1

VD: Nếu chỉ dùng IR0 IR1: 11111100

OCW2

- Xác định cách PIC xử lý yêu cầu ngắt
  - Chế độ ưu tiên cố định:
    - IR0>…>IR7
  - Đổi mức ưu tiên tự động:
    - Quay vòng
  - Ưu tiên đích danh
    - Gán mức độ ưu tiên cho từng yêu cầu ngắt

Quay vòng ưu tiên

Trước khi quay

Giả sử IR4 có mức ưu tiên cao

Trạng thái ngắt (ISR)

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	1	0	0	0	0

Mức ưu tiên

7	6	5	4	3	2	1	0

Thấp nhất — Cao nhất

Sau khi quay

Trạng thái ngắt

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	0	0	0	0	0

Mức ưu tiên

2	1	0	7	6	5	4	3

Cao nhất — Thấp nhất

OCW2

OCW2: Xác định việc xử lý các yêu cầu ngắt của PIC

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	R	SL	EOI	0	0	L_2	L_1	L_0

IR | Mức ưu tiên

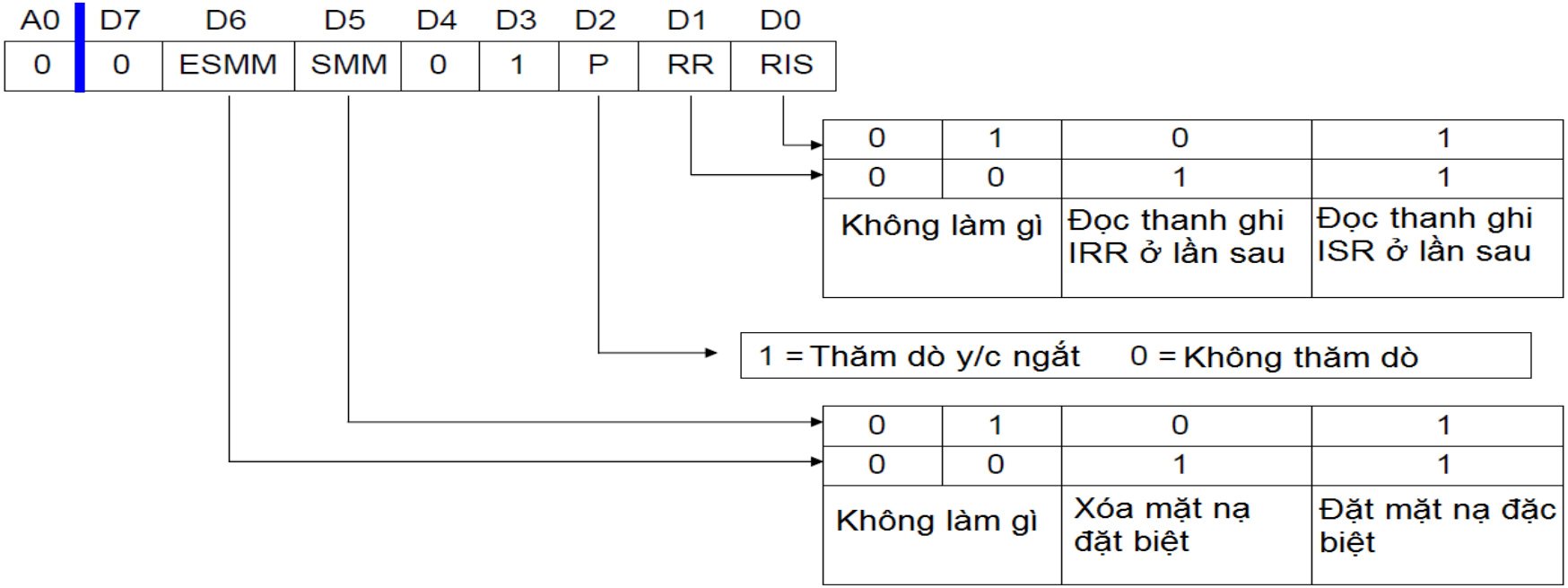
0	1	2	3	4	5	6	7
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1

0	0	1	EOI thường	Kết thúc ngắt
0	1	1	EOI đích danh (dùng với Lx)*	
1	0	1	Đổi mức ưu tiên với EOI thường	Đổi mức ưu tiên tự động
1	0	0	Lập chế độ đổi khi có EOI tự động	
0	0	0	Xóa chế độ đổi	
1	1	1	Đổi ưu tiên với EOI đích danh*	Specific Rotation
1	1	0	Đặt ưu tiên cho EOI*	
0	1	0	Không làm gì	

(* L0-L2) (Trường hợp còn lại : L0-L2=0)

OCW3

- Chọn các thanh ghi để đọc
- Thăm dò trạng thái yêu cầu ngắt
- Thao tác với thanh ghi mặt nạ



Thăm dò & IRR&ISR

D7	D6	D5	D4	D3	D2	D1	D0
1: có ngắt	X	x	X	x	Số hiệu yêu cầu ngắt		

IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
D7	D6	D5	D4	D3	D2	D1	D0

- 0 = Có yêu cầu ngắt
- 1 = Không có yêu cầu ngắt

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
D7	D6	D5	D4	D3	D2	D1	D0

- 0 = Yêu cầu ngắt IRi không được phục vụ
- 1 = Yêu cầu ngắt IRi đang được phục vụ

Ví dụ

- Cho phép ngắt tại các đầu vào IR012
- Đặt 8259 hoạt động ở chế độ ưu tiên quay vòng tự động
- Đặt yêu cầu ngắt IR 1 có độ ưu tiên cao nhất
- Đặt yêu cầu ngắt IR 2 có độ ưu tiên cao nhất
- Thực hiện kiểm tra yêu cầu ngắt bằng phần mềm (Đặt mặt nạ ngắt, Sử dụng lệnh Poll thăm dò, đọc thanh ghi yêu cầu ngắt)?

Trình tự sự kiện (8088)

- Các tín hiệu yêu cầu ngắt do thiết bị vào/ra gửi tới PIC làm cho các bít tương ứng trong IRR được bật lên
- PIC xem xét các yêu cầu ngắt và báo hiệu cho CPU khi cần (INTR)
- CPU xác nhận ngắt bằng cách đưa ra INTA
- Khi nhận được INTA, PIC xóa bít tương ứng trong IRR và bít ưu tiên cao nhất của ISR được bật
- CPU đưa ra INTA thứ 2, PIC đưa ra 1 byte dữ liệu về số hiệu ngắt
- Kết thúc chu kỳ ngắt. Nếu dùng AEOI thì bit ISR bị xóa vào cuối xung INTA thứ 2. Nếu không, bít ISR giữ nguyên cho đến khi có câu lệnh EOI

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kỹ Thuật Vi Xử Lý

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: Khoa Học Máy Tính- Khoa CNTT1

Học kỳ/Năm biên soạn:2009

NỘI DUNG

GHÉP NỐI 8088 VỚI BỘ ĐIỀU KHIỂN VÀO RA TRỰC TIẾP BỘ NHỚ

Giảng viên: TS. Phạm Hoàng Duy
E-mail: phamhduy@gmail.com
Năm biên soạn: 2009

NỘI DUNG

Ghép nối vào ra trực tiếp bộ nhớ

Nội dung

- Khái niệm DMA
- Bộ điều khiển DMA 8237

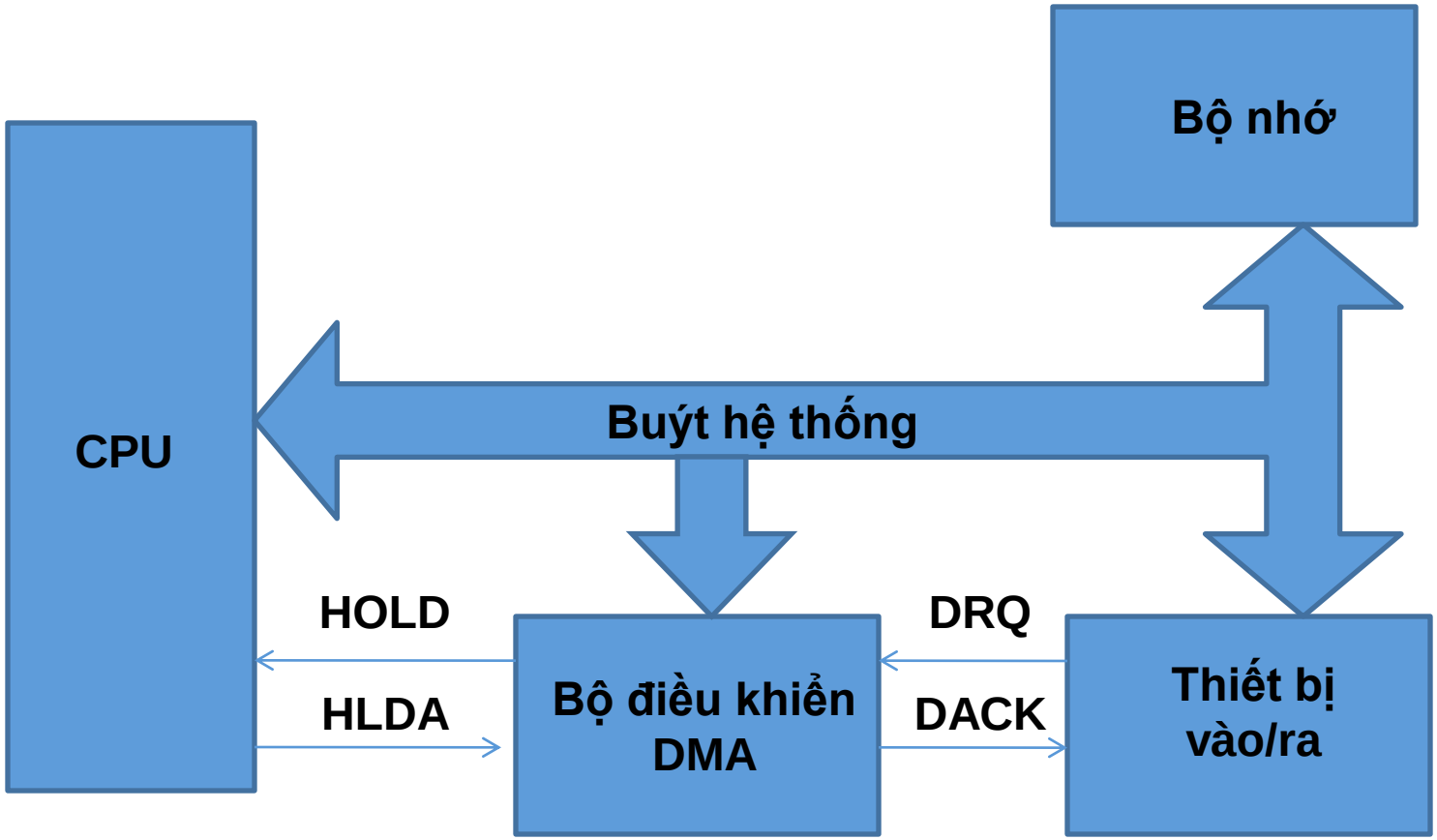
Truy nhập trực tiếp bộ nhớ - DMA

- Truy nhập trực tiếp bộ nhớ là quá trình các thiết bị vào/ra chiếm quyền điều khiển hệ thống buýt của CPU
- Truy nhập trực tiếp bộ nhớ thường dùng để truyền dữ liệu với tốc độ cao như ổ cứng, CDROM …
- Ý tưởng cơ bản của DMA là truyền dữ liệu theo từng khối trực tiếp giữa bộ nhớ và thiết bị ngoại vi mà không đi qua CPU
- Tốc độ truyền dữ liệu lệ thuộc vào tốc độ truy nhập của bộ nhớ và thiết bị

Truy nhập trực tiếp bộ nhớ - DMA

- Bình thường CPU toàn quyền kiểm soát buýt hệ thống. Trong quá trình DMA, các thiết bị lấy quyền điều khiển
- Các tín hiệu HOLD và HLDA được sử dụng để nhận và xác nhận yêu cầu treo CPU

Truy nhập trực tiếp bộ nhớ - DMA



Bộ điều khiển DMA – Intel 8237

- Hỗ trợ 4 kênh DMA độc lập
- Tự động khởi tạo độc lập cho tất cả các kênh
- Điều khiển cho phép hoặc cấm từng yêu cầu DMA riêng lẻ
- Truyền từ bộ nhớ tới bộ nhớ
- Khởi tạo các khối bộ nhớ
- Tự động tăng/giảm địa chỉ
- Tốc độ truyền dữ liệu tới 1.6MB/s với 8237A ở 5MHz

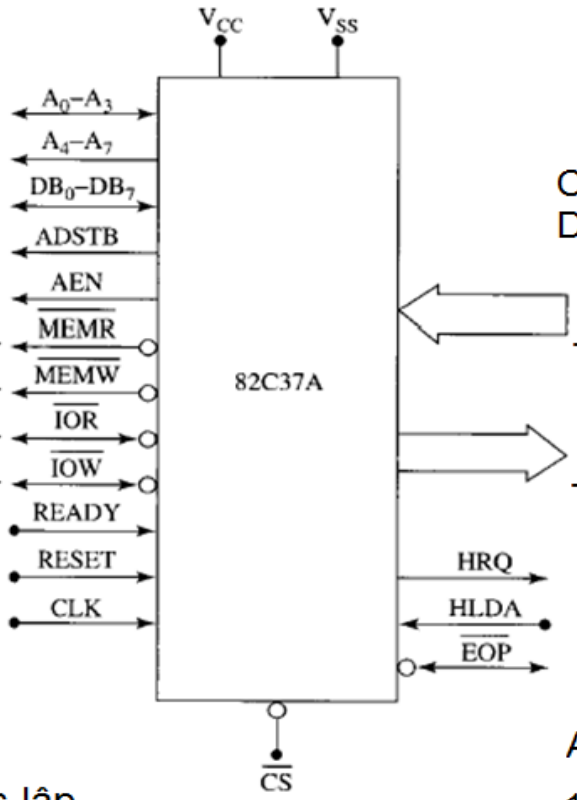
Các tín hiệu 8237

ADSTB: Tín hiệu địa chỉ (byte cao) sẵn sàng

AEN: Cho phép địa chỉ

Tín hiệu điều khiển bộ nhớ

Tín hiệu điều khiển thiết bị



Các tín hiệu kết nối DMA

$DREQ_0$–$DREQ_3$ Tín hiệu yêu cầu DMA

$DACK_0$–$DACK_3$ Tín hiệu xác nhận DMA

EOP: Kết thúc dịch vụ DMA

DB0-DB7 dùng để truyền dữ liệu hoặc lập trình 8237

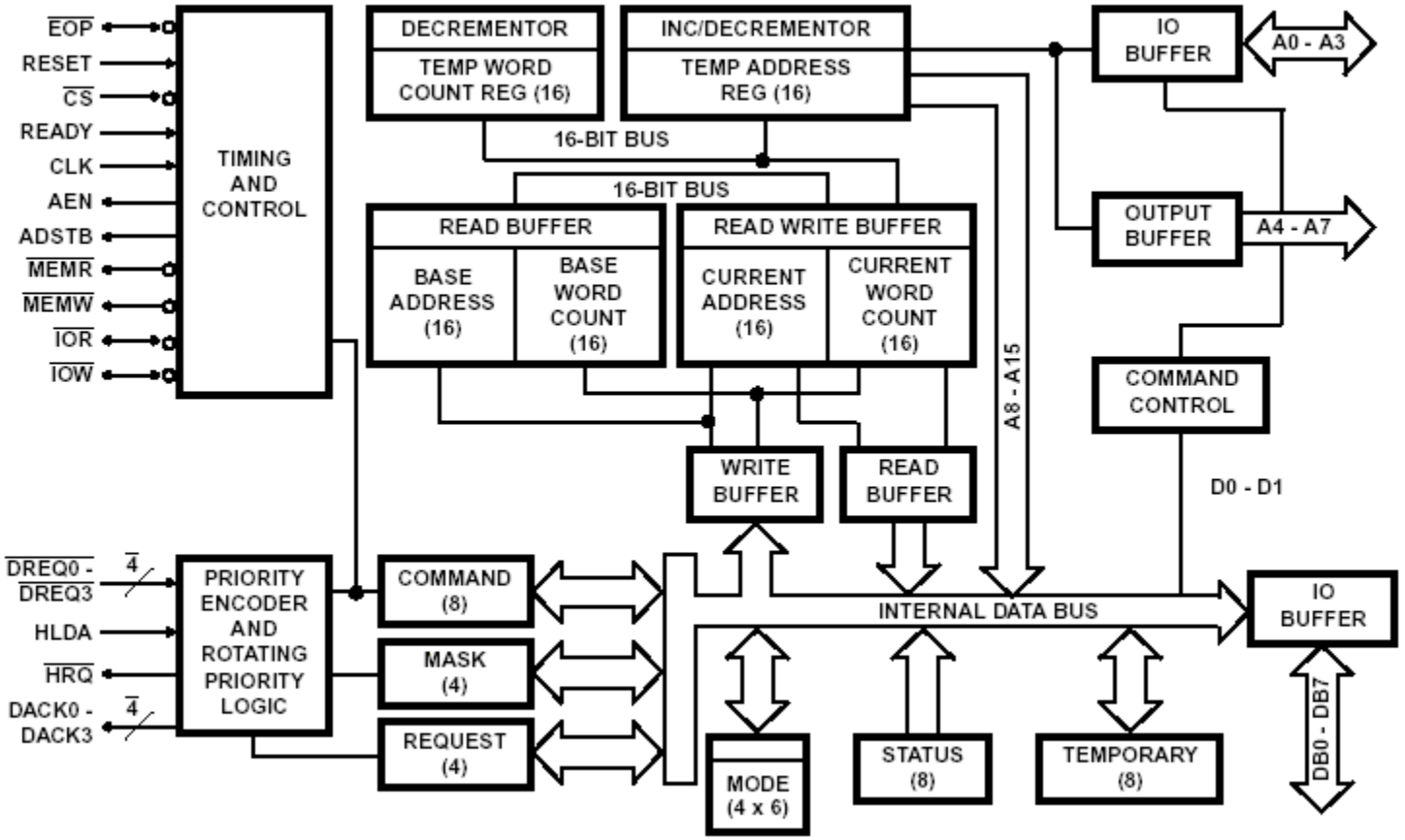
A0-A3 dùng để

1) Truy nhập các thanh ghi bên trong

2) Chứa 4 bít địa chỉ thấp trong khi truyền dữ liệu

Sơ đồ khối 8237

Block Diagram



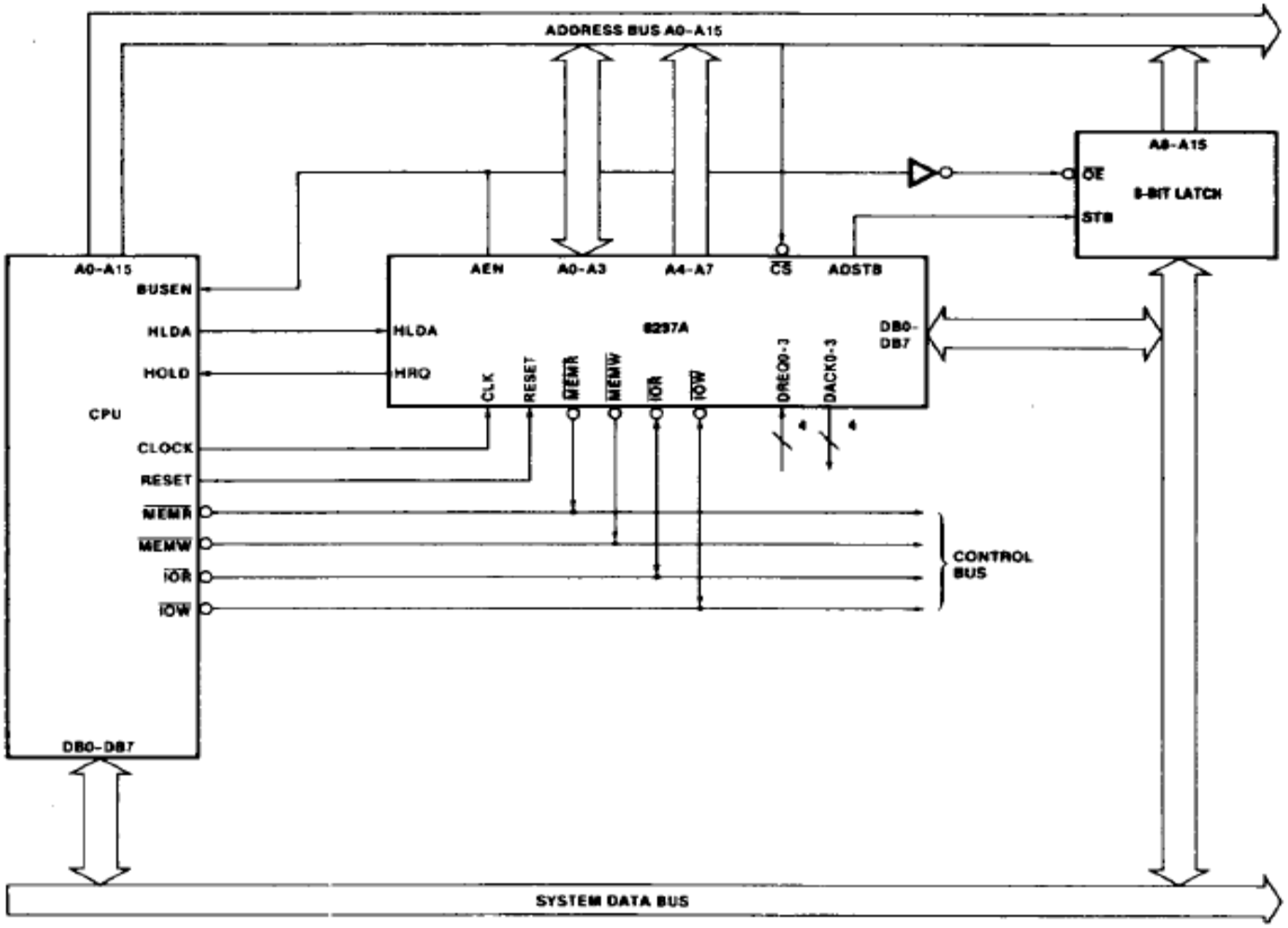
Sơ đồ khối 8237

- **Timing Control**
  - Sinh ra các tín hiệu định thời bên trong và tín hiệu điều khiển bên ngoài cho 8237
- **Program Command Control**
  - Giải mã các câu lệnh gửi tới 8237 trước khi phục vụ yêu cầu DMA
  - Giải mã từ điều khiển chế độ xác định kiểu DMA trong khi phục vụ yêu cầu DMA
- **Priority Encoder**
  - Giải quyết xung đột yêu cầu DMA đồng thời

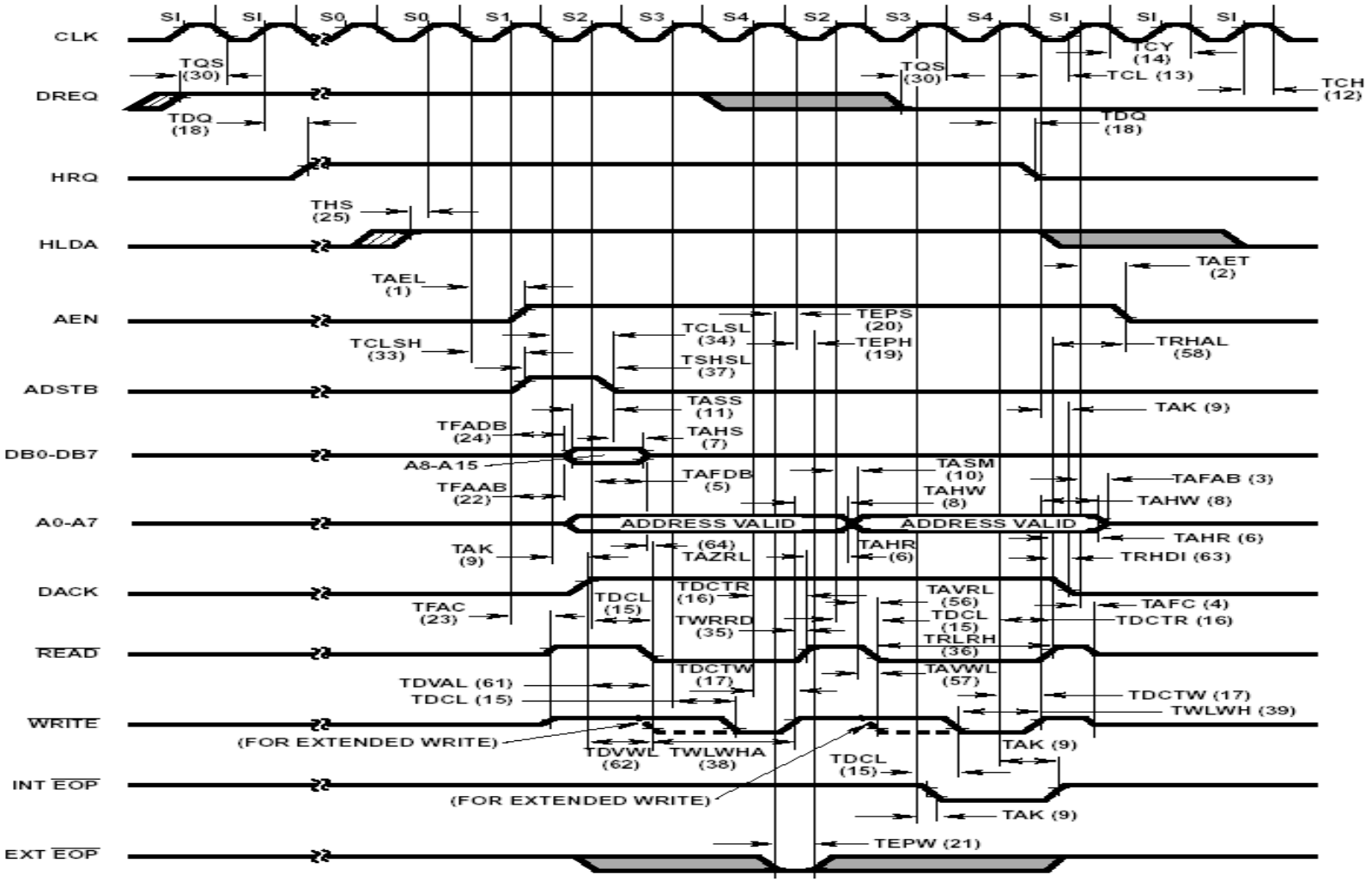
Sơ đồ khối 8237

Tên thanh ghi	Kích cỡ (bít)	Số lượng	A3..A0
Địa chỉ cơ sở	16	4	
Đếm từ cơ sở	16	4	
Địa chỉ hiện thời	16	4	
Đếm từ hiện thời	16	4	
Địa chỉ tạm	16	1	
Đếm từ tạm	16	1	
Trạng thái	8	1	
Lệnh	8	1	
Tạm	8	1	
Chế độ	6	4	
Mặt nạ	4	1	
Yêu cầu	4	1	

Ghép nối với CPU



Biểu đồ thời gian



Hoạt động của 8237

- Bao gồm 2 chu kỳ: rỗi và hoạt động
- Chu kỳ rỗi
  - Kiểm tra các tín hiệu DREQ xem có thiết bị nào yêu cầu DMA
  - Kiểm tra CS nếu CPU có yêu cầu đọc/ghi các thanh ghi bên trong
- Chu kỳ hoạt động
  - Diễn ra khi có yêu cầu DREQ từ thiết bị
  - Gửi tín hiệu HRQ tới CPU
  - Hoạt động ở 1 trong 4 chế độ

Các chế độ hoạt động

- Chế độ truyền đơn
- Chế độ truyền theo khối
- Chế độ truyền theo yêu cầu
- Chế độ xếp tầng

Chế độ truyền đơn

- Thiết bị được lập trình để chỉ thực hiện 1 thao tác truyền. Từ đếm giảm dần, địa chỉ giảm dần (hoặc tăng) sau mỗi thao tác truyền. Khi từ đếm giảm từ 0 sang FFFFH, quá trình truyền kế thúc
- DREQ phải giữ ở mức tích cực cho đến khi DACK được xác nhận. Nếu DREQ giữ ở mức tích cực trong suốt quá trình truyền đơn thì HRQ sẽ chuyển sang mức thụ động và giải phóng buýt cho hệ thống. Quá trình tiếp tục cho đến nhận được tín hiệu HLDA mới và, thao tác truyền được tiếp tục

Chế độ truyền theo khối

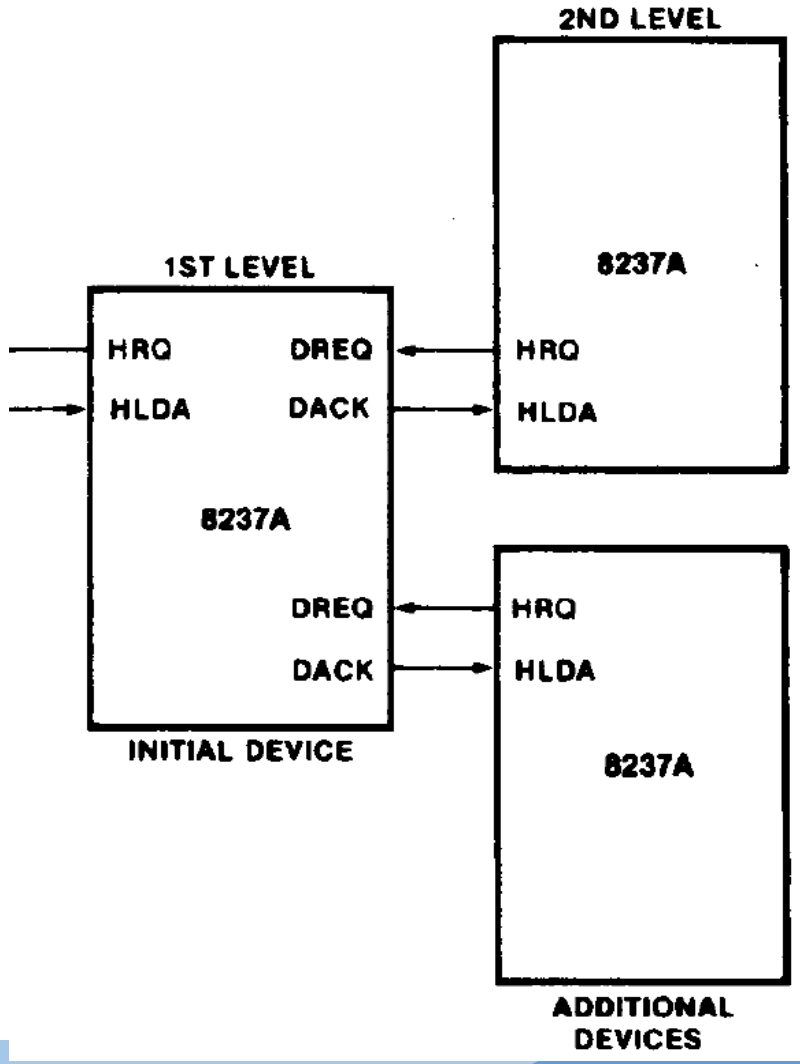
- Bộ điều khiển DMA được kích hoạt bởi DREQ và liên tục truyền trong quá trình phục vụ yêu cầu cho đến khi kết thúc do bộ đếm từ chuyển từ 0 về FFFFH hoặc do tín hiệu EOP từ bên ngoài.
- DREQ chỉ cần giữ tích cực cho đến khi nhận được DACK.

Chế độ truyền theo yêu cầu

- Thực hiện việc truyền liên tục cho đến khi bộ đếm chuyển sang FFFFH hoặc nhận được EOP hoặc DREQ chuyển sang thụ động.

Chế độ xếp tầng

- Dùng để mở rộng hệ thống



Các kiểu truyền dữ liệu

- Từ bộ nhớ tới bộ nhớ
- Tự động khởi tạo
- Ưu tiên

Từ bộ nhớ tới bộ nhớ

- Cho phép tiết kiệm thời gian truyền dữ liệu từ không gian nhớ này sang không gian nhớ khác
- Sử dụng 2 kênh của bộ điều khiển DMA
- Quá trình truyền được khởi xướng bằng cách đặt DREQ cho kênh 0. Sau khi nhận được HLDA, bộ điều khiển thực hiện việc truyền theo khối
  - Thanh ghi địa chỉ hiện thời trên kênh 0 gán vào địa chỉ bắt đầu của không gian nhớ cần đọc
  - Dữ liệu được đọc vào thanh ghi tạm
  - Kênh 1 truyền dữ liệu từ thanh ghi tạm vào bộ nhớ. Địa chỉ được xác định bằng thanh ghi địa chỉ hiện thời của kênh 1

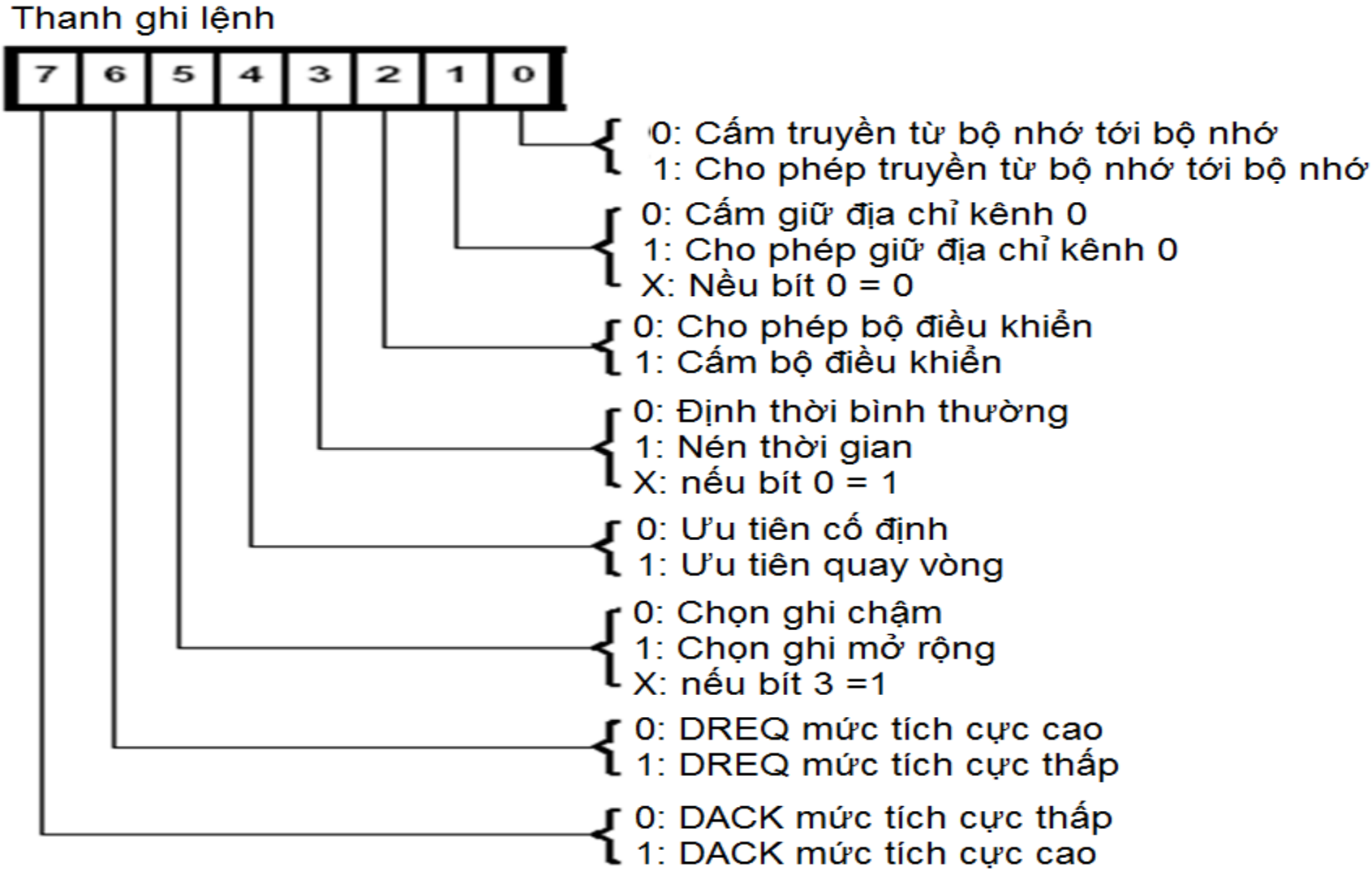
Tự động khởi tạo

- Trong quá trình xác lập, các giá trị của thanh ghi địa chỉ hiện thời và đếm từ hiện thời được khôi phục từ giá trị của thanh ghi địa chỉ cơ sở và đếm từ cơ sở của kênh khi có tín hiệu EOP

Truyền ưu tiên

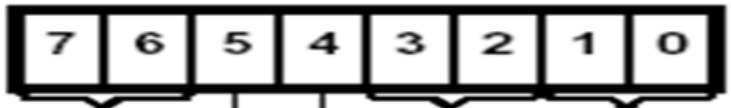
- Ưu tiên cố định
  - Kênh 0 > ..> Kênh 3
  - Khi có nhiều yêu cầu DMA, kênh nào có độ ưu tiên cao hơn được đáp ứng trước
- Ưu tiên quay vòng
  - Kênh nào được phục vụ thì sẽ chuyển xuống độ ưu tiên thấp nhất

Cấu trúc các thanh ghi



Cấu trúc các thanh ghi

Thanh ghi chế độ



00: Chọn kênh 0
01: Chọn kênh 1
10: Chọn kênh 2
11: Chọn kênh 3

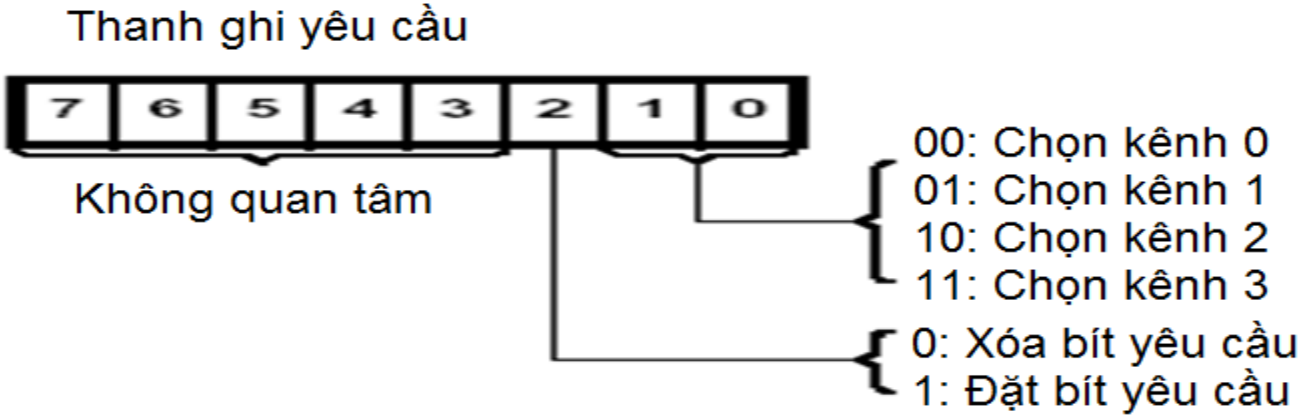
00: Kiểm tra truyền
01: Ghi
10: Đọc
11: Không dùng
XX: Nếu bít 6,7 =1,1

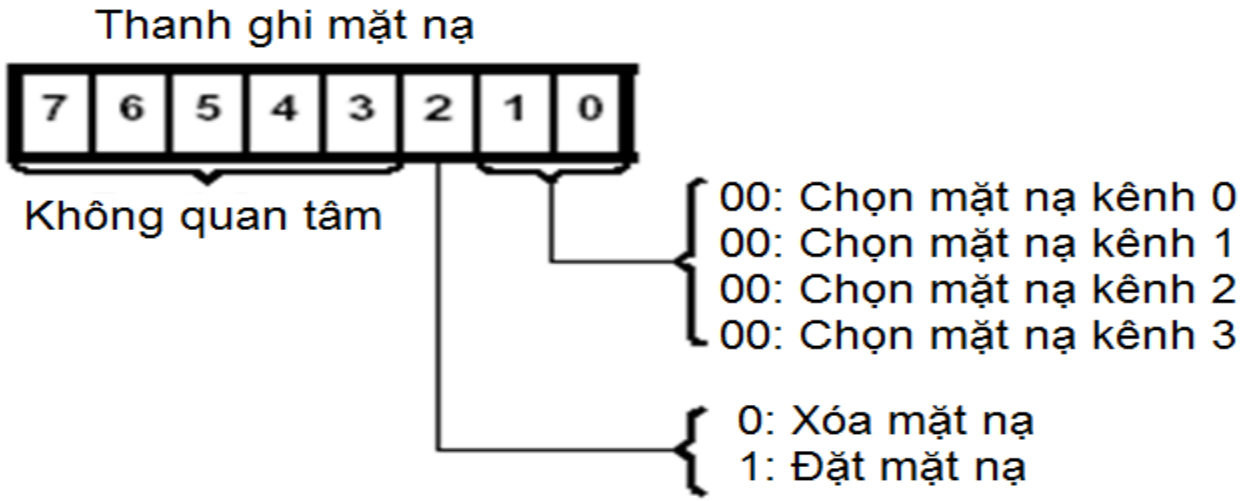
0: Cấm tự khởi tạo
1: Cho phép tự khởi tạo

0: Chọn tăng địa chỉ
1: Chọn giảm địa chỉ

00: Chọn chế độ truyền theo yêu cầu
01: Chọn chế độ truyền đơn
10: Chọn chế độ truyền theo khối
11: Chọn chế độ xếp tầng

Cấu trúc các thanh ghi





Các câu lệnh phần mềm

Tín hiệu						Thao tác
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Đọc thanh ghi trạng thái
1	0	0	0	1	0	Ghi thanh ghi lệnh
1	0	0	1	0	1	Không dùng
1	0	0	1	1	0	Ghi thanh ghi yêu cầu
1	0	1	0	0	1	Không dùng
1	0	1	0	1	0	Ghi thanh ghi mặt nạ
1	0	1	1	0	1	Không dùng
1	0	1	1	1	0	Ghi thanh ghi chế độ
1	1	0	0	0	1	Không dùng
1	1	0	0	1	0	Xóa con trỏ mạch lật
1	1	0	1	0	1	Đọc thanh ghi tạm
1	1	0	1	1	0	Khởi động lại
1	1	1	0	0	1	Không dùng
1	1	1	0	1	0	Xóa thanh ghi mặt nạ
1	1	1	1	0	1	Không dùng
1	1	1	1	1	0	Ghi tất các các bít mặt nạ

Mã lệnh thanh ghi đếm từ và địa chỉ

CHANNEL	REGISTER	OPERATION	SIGNALS							FIRST/LAST FLIP-FLOP STATE	DATA BUS DB0-DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	0	1	0	W0-W7
			0	1	0	0	0	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0-W7
			0	0	1	0	0	0	1	1	W8-W15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	1	1	0	W0-W7
			0	1	0	0	0	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0-W7
			0	0	1	0	0	1	1	1	W8-W15
2	Base and Current Address	Write	0	1	0	0	1	0	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	0	1	0	W0-W7
			0	1	0	0	1	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0-W7
			0	0	1	0	1	0	1	1	W8-W15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0-W7
			0	0	1	0	1	1	1	1	W8-W15

Lập trình 8237

- Xóa mạch lật
- Cấm kênh
- Đặt địa chỉ thấp (LSB), địa chỉ cao (MSB)
- Đặt từ đếm thấp, từ đếm cao
- Có thể đặt thêm chế độ hoạt động
- Kiểm tra trạng thái kết thúc

Ví dụ

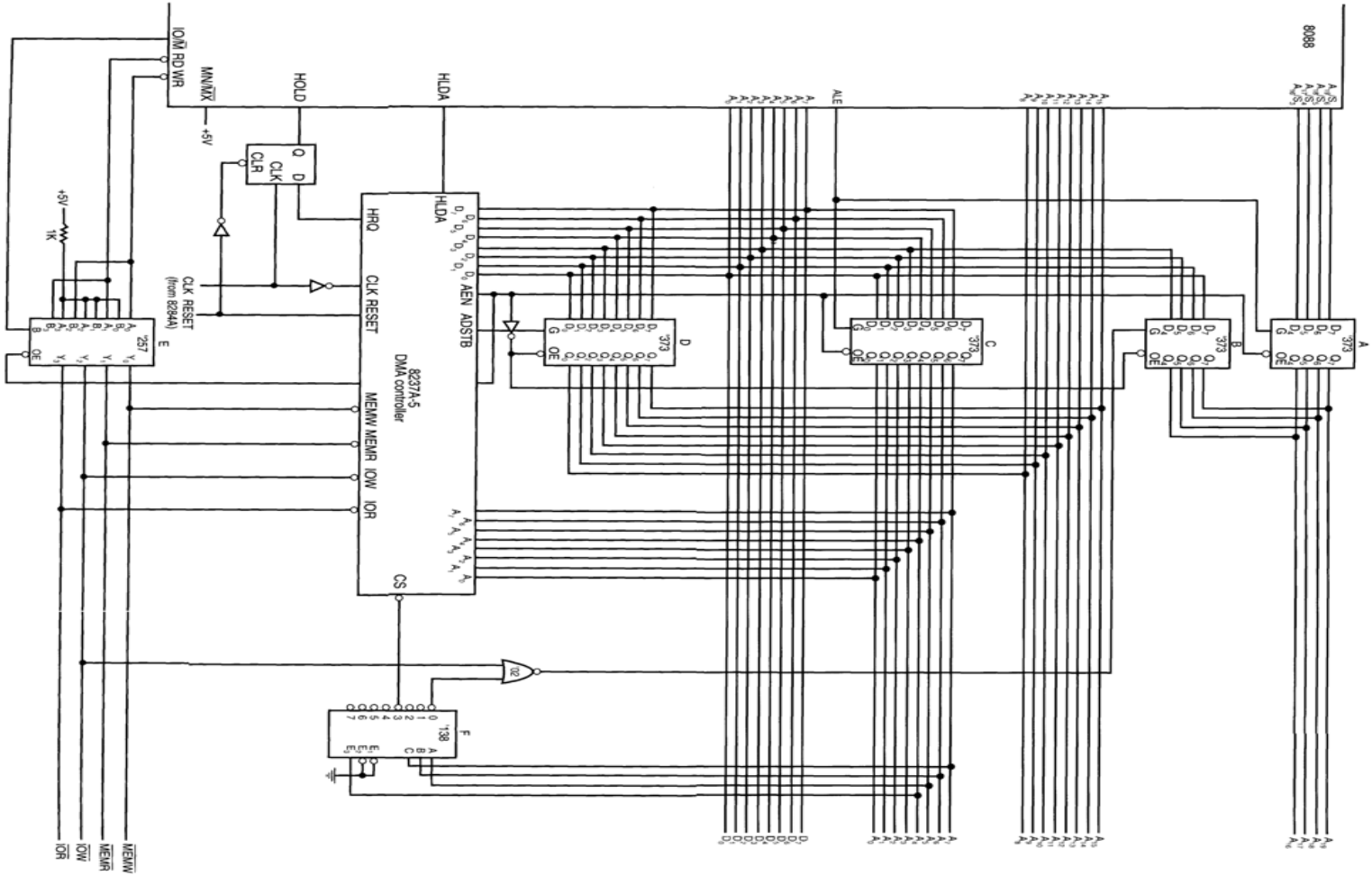


FIGURE 12–12 Complete 8088 minimum mode DMA system (pp. 476–477)

Ví dụ lập trình

EXAMPLE 12–1

```
                ;A procedure that transfers a block of data using th
                ;8237A DMA controller in Figure 12-12. This is a
                ;memory-to-memory block transfer.
                ;
                ;Calling parameters:
                ;   SI = source address
                ;   DI = destination address
                ;   CX = count
                ;   ES = segment of source and destination
;
= 0010          LATCHB  EQU   10H        ;latch B
= 007C          CLEAR_F EQU   7CH        ;F/L flip flop
= 0070          CH0_A   EQU   70H        ;channel 0 address
= 0072          CH1_A   EQU   72H        ;channel 1 address
= 0073          CH1_C   EQU   73H        ;channel 1 count
= 007B          MODE    EQU   7BH        ;mode
= 0078          CMMD    EQU   78H        ;command
= 007F          MASKS   EQU   7FH        ;masks
= 0079          REQ     EQU   79H        ;request register
= 0078          STATUS  EQU   78H        ;status register

0000            TRANS   PROC  FAR USES AX

0001  8C C0             MOV   AX,ES      ;program latch B
0003  8A C4             MOV   AL,AH
0005  C0 E8 04          SHR   AL,4
0008  E6 10             OUT   LATCHB,AL
000A  E6 7C             OUT   CLEAR_F,AL ;clear F/L flip-flop

000C  8C C0             MOV   AX,ES      ;program source address
000E  C1 E0 04          SHL   AX,4
0011  03 C6             ADD   AX,SI      ;form source offset
0013  E6 70             OUT   CH0_A,AL
0015  8A C4             MOV   AL,AH
0017  E6 70             OUT   CH0_A,AL

0019  8C C0             MOV   AX,ES      ;program destination address
001B  C1 E0 04          SHL   AX,4
001E  03 C7             ADD   AX,DI      ;form destination offset
0020  E6 72             OUT   CH1_A,AL
0022  8A C4             MOV   AL,AH
0024  E6 72             OUT   CH1_A,AL
0026  8B C1             MOV   AX,CX      ;program count
0028  48                DEC   AX         ;adjust count
0029  E6 73             OUT   CH1_C,AL
002B  8A C4             MOV   AL,AH
002D  E6 73             OUT   CH1_C,AL

002F  B0 88             MOV   AL,88H     ;program mode
0031  E6 7B             OUT   MODE,AL

0033  B0 85             MOV   AL,85H
0035  E6 7B             OUT   MODE,AL

0037  B0 01             MOV   AL,1       ;enable block transfer
0039  E6 78             OUT   CMMD,AL

003B  B0 0E             MOV   AL,0EH     ;unmask channel 0
003D  E6 7F             OUT   MASKS,AL

003F  B0 04             MOV   AL,4       ;start DMA transfer
0041  E6 79             OUT   REQ,AL

                        .REPEAT          ;wait until DMA complete
0043  E4 78             IN    AL,STATUS
                        .UNTIL AL &1

                        RET

                TRANS   ENDP
```

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kỹ Thuật Vi Xử Lý

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: Khoa Học Máy Tính- Khoa CNTT1

Học kỳ/Năm biên soạn:2009

NỘI DUNG

GHÉP NỐI TRUYỀN DỮ LIỆU SONG SONG

Giảng viên: TS. Phạm Hoàng Duy

E-mail: phamhduy@gmail.com

Năm biên soạn: 2009

Nội dung

- Giới thiệu
- 8255A- Giao tiếp song song

Truyền dữ liệu nối tiếp và song song

- Máy tính có thể kết nối với thiết bị theo kiểu song song hoặc nối tiếp
- Truyền dữ liệu song song cần tối thiểu 8 dây cho 8 bít dữ liệu trong 1 byte
  - Các dây khác dùng cho giao thức liên lạc như thăm dò trạng thái, xác nhận dữ liệu

Truyền dữ liệu nối tiếp và song song

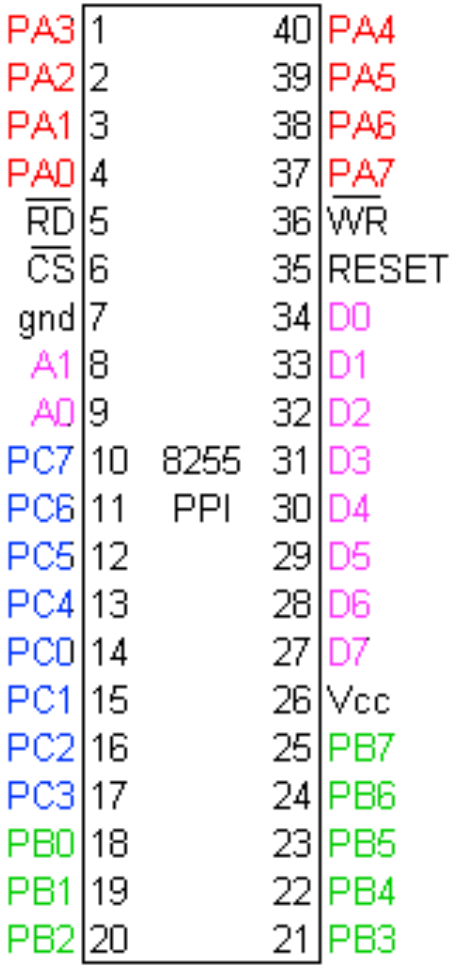
- Truyền nối tiếp thường truyền dữ liệu trên 1 dây, từng bít một. Giao tiếp vào ra sẽ chuyển đổi byte dữ liệu thành chuỗi bít
- Kết nối nối tiếp thường dùng để truyền dữ liệu đi xa. Chuẩn nối tiếp phổ biến là RS232
- Cổng song song thường nhanh hơn nối tiếp
- Ví dụ: Cổng máy in, cổng nối tiếp (chuột)

IC điều khiển song song lập trình được

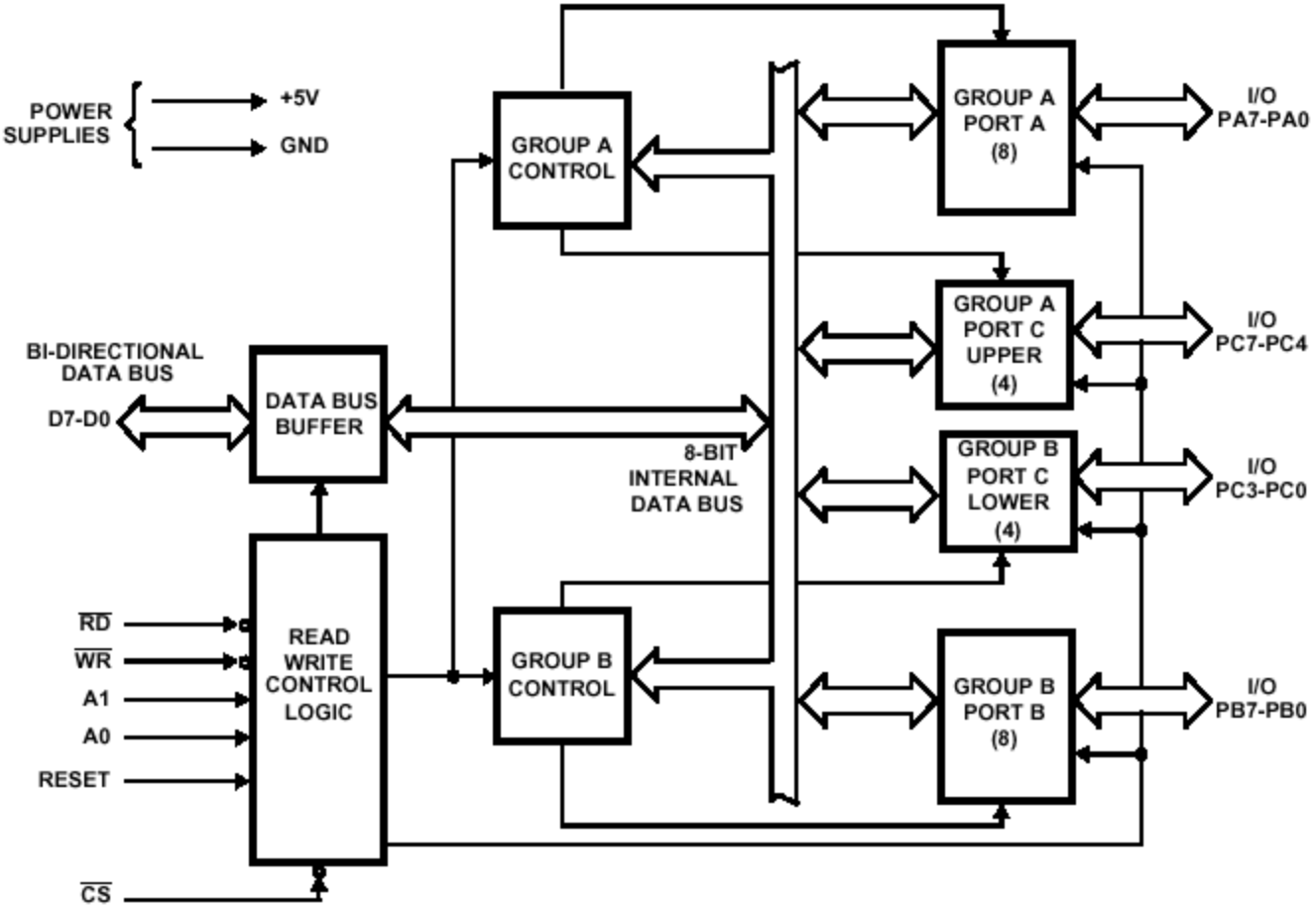
- 8255A là thiết bị giao tiếp ngoại vi lập trình được PPI dùng cho hệ thống máy tính Intel. Thiết bị có thể được lập trình mà không cần thiết bị logic ngoài để giao tiếp với thiết bị
- Đệm dữ liệu: Hỗ trợ bộ đệm 8 bít, 3 trạng thái
- Loogic điều khiển và đọc/ghi:
  - Quản lý truyền dữ liệu trong/ngoài PPI

Tín hiệu 8255A

Tín hiệu	Chức năng
CS	Chọn chíp (mức thấp)
RD	Đọc (mức thấp)
WR	Ghi (mức thấp)
A_0A_1	Chọn cổng
PA_7-PA_0	Cổng A
PB_7-PB_0	Cổng B
PC_7-PC_0	Cổng C
D_7-D_0	Dữ liệu

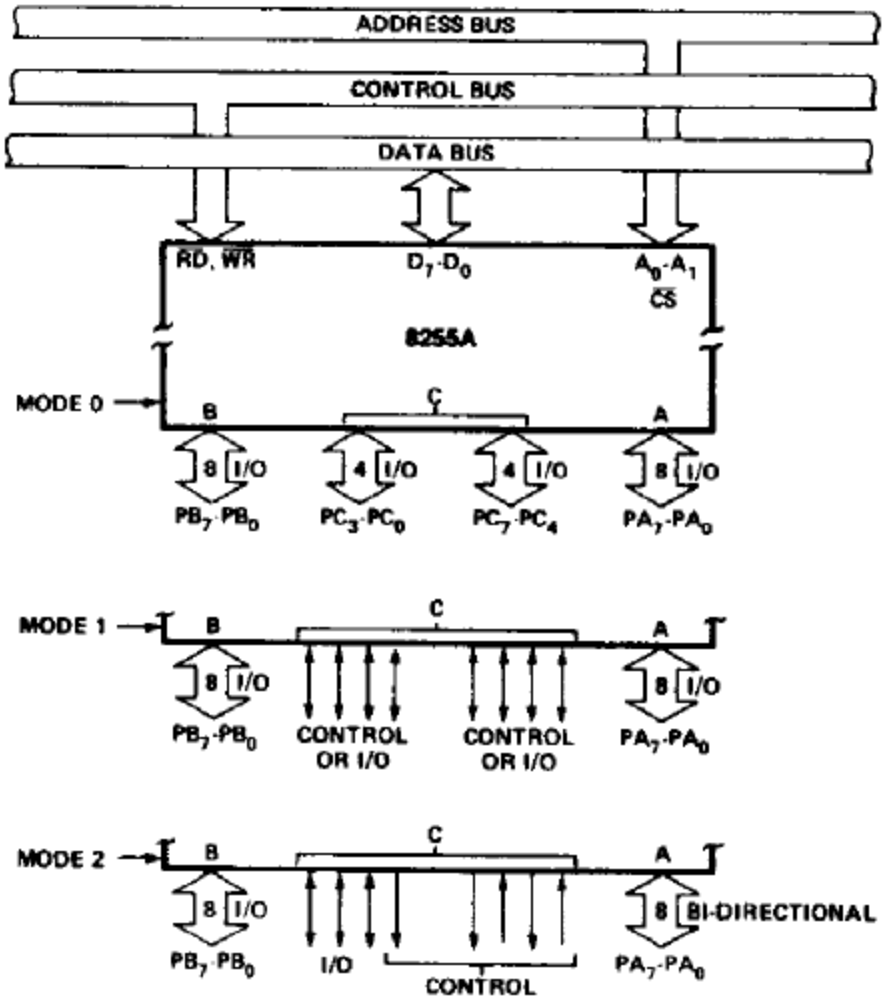


Sơ đồ chức năng



Chế độ làm việc

- Ba chế độ làm việc chọn bằng phần mềm
  - Chế độ 0: Vào/ra cơ sở
  - Chế độ 1: Vào/ra thăm dò
  - Chế độ 2: Vào/ra hai chiều
- Chế độ làm việc được chọn riêng rẽ trên cổng A, B
- Cổng C có thể tách làm 2 cổng riêng



Chế độ cơ sở

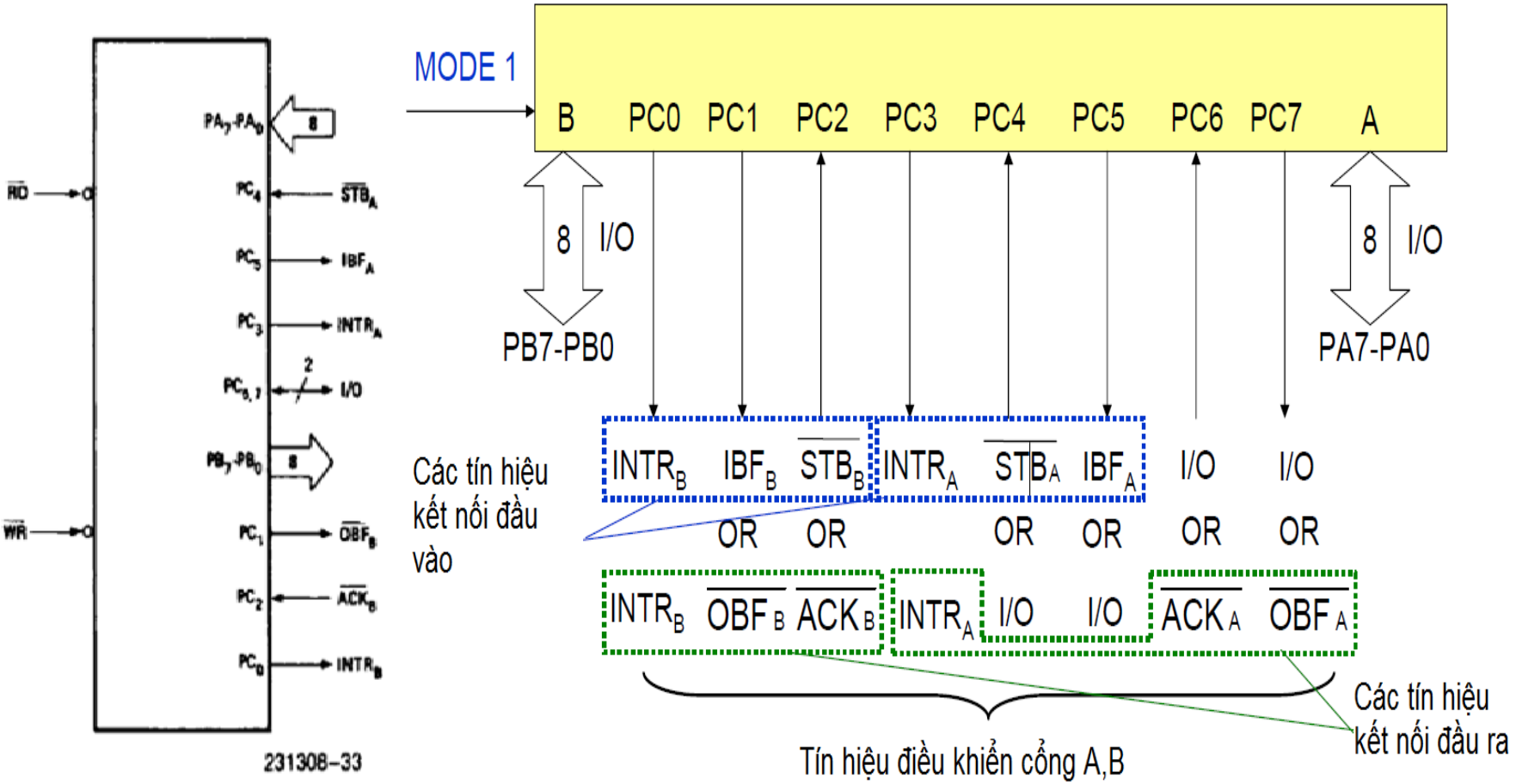
- Cung cấp thao tác vào/ra đơn giản cho từng cổng
  - Không có tín hiệu kết nối
- Có thể dùng 2 cổng 8 bít và 2 cổng 4 bít
- Bất kỳ cổng nào có thể dùng làm cổng vào/ra
- Các tín hiệu vào ra được chốt

Chế độ vào/ra thăm dò

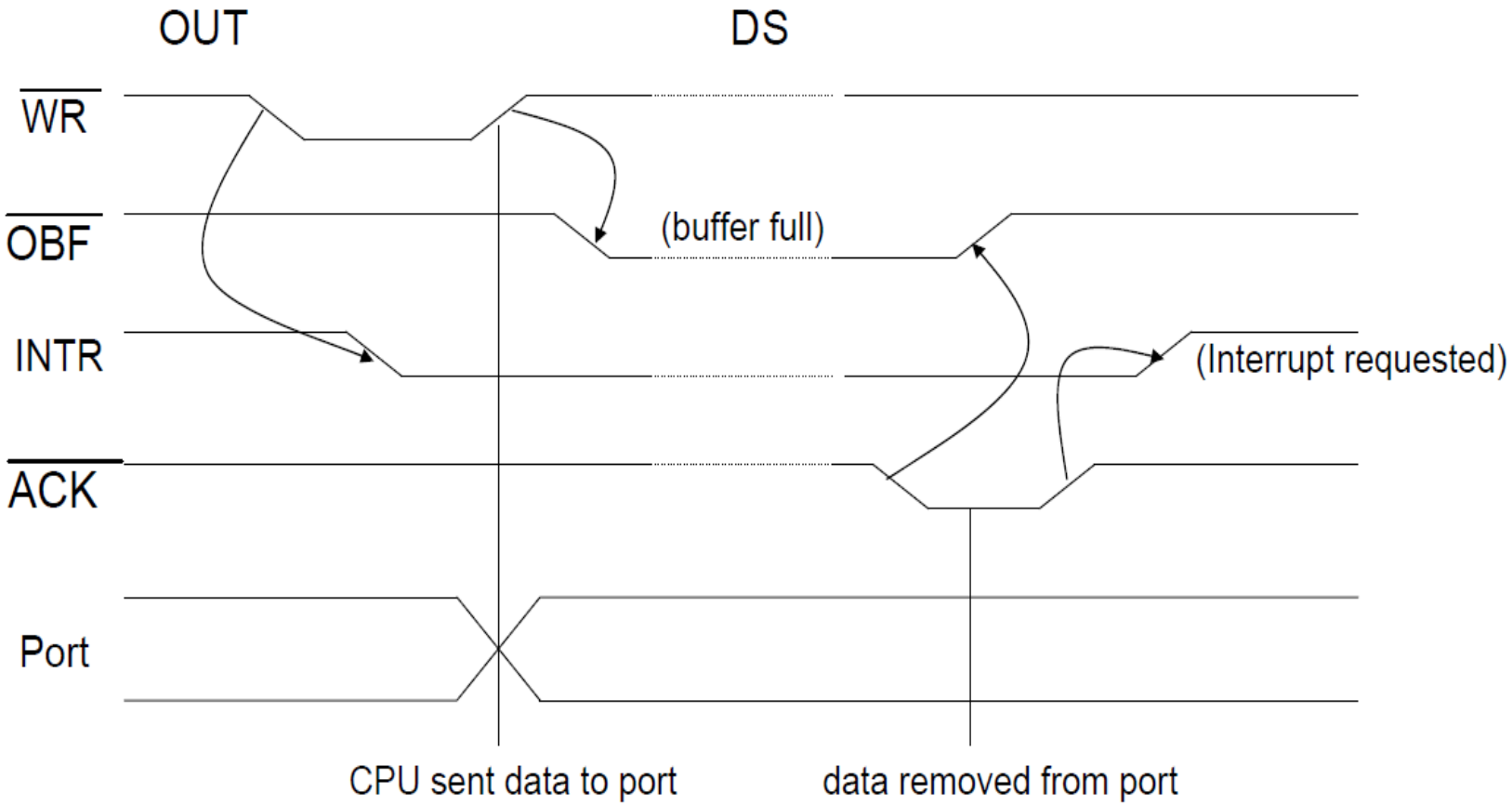
- Cung cấp tín hiệu kết nối qua cổng C
- 2 nhóm cổng A, B
- Mỗi nhóm bao gồm 8 bít dữ liệu và 4 bít điều khiển
- Các bít dữ liệu có thể vao/ra
- 4 bít điều khiển dùng để kiểm tra trạng thái dữ liệu

Đầu vào	Đầu ra
STB: Kiểm tra đầu vào (mức thấp) IBF: Dữ liệu sẵn sàng (Mức cao) INTR: Báo ngắt CPU (mức cao)	OBF: Dữ liệu ra sẵn sàng (mức thấp) ACK: Nhận xong dữ liệu (mức thấp) INTR: Báo ngắt CPU (Mức cao)

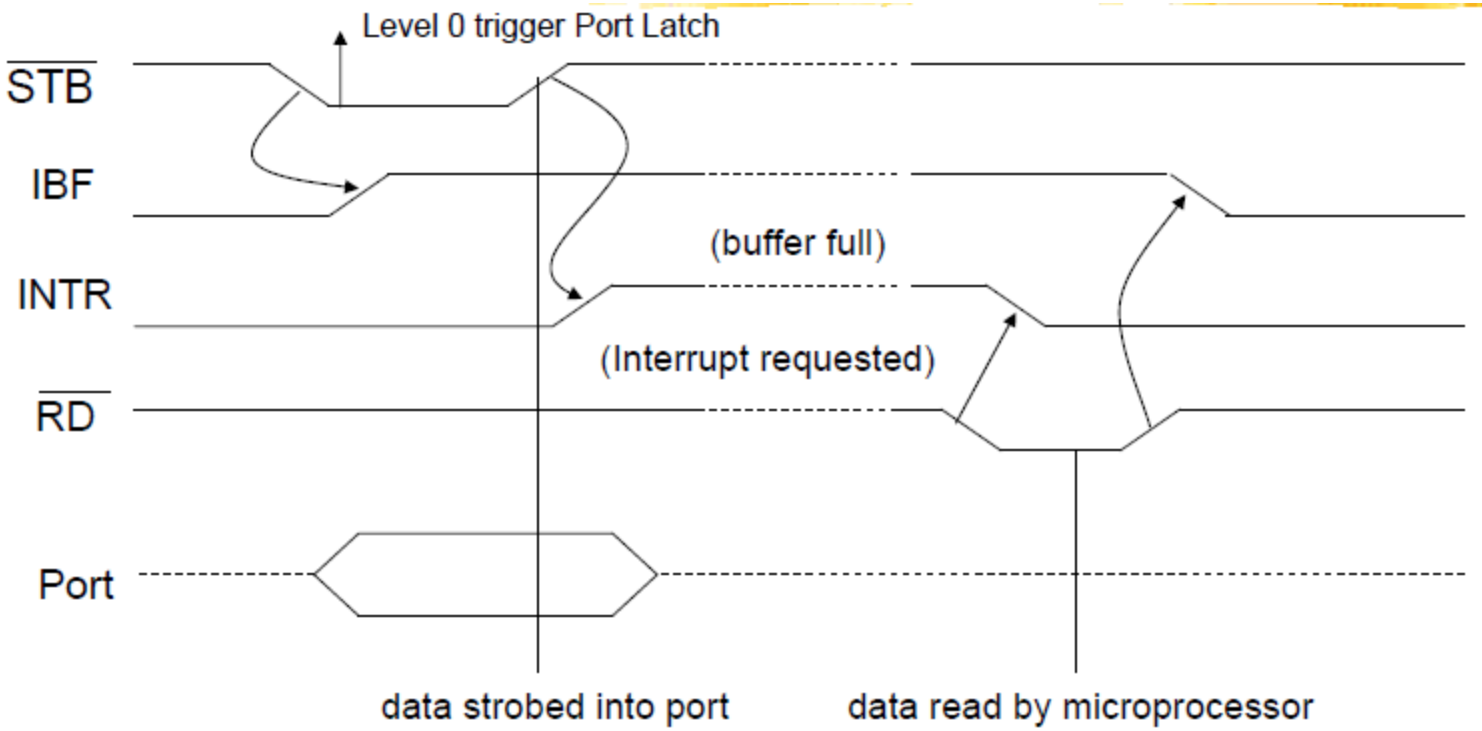
Cổng A, B kết hợp



Biểu đồ thời gian cổng ra



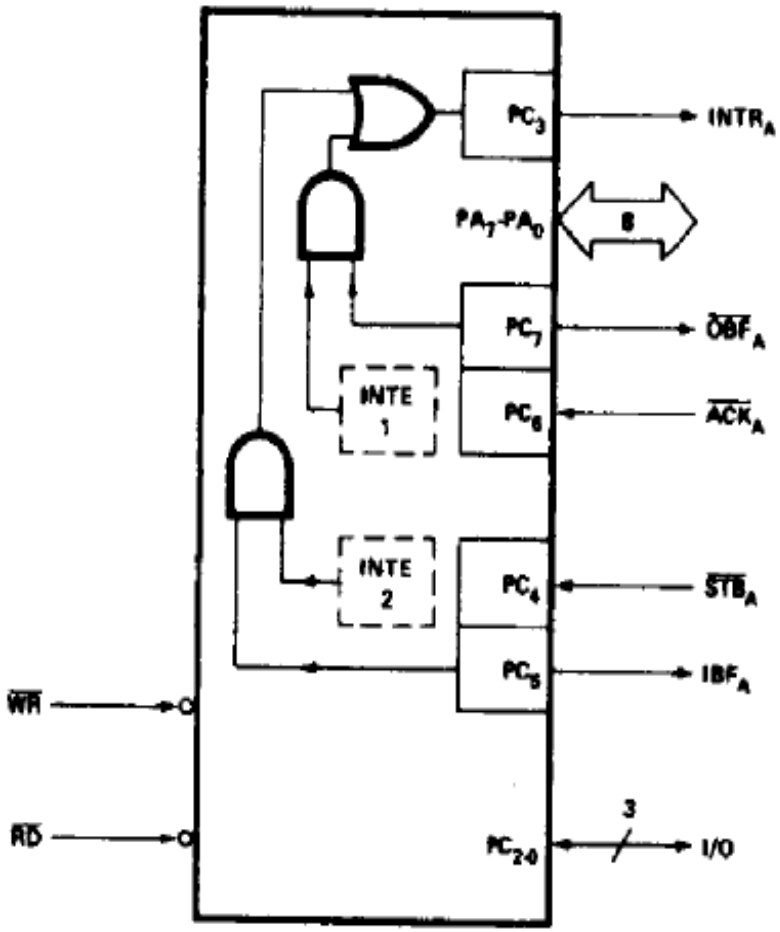
Biểu đồ thời gian cổng vào



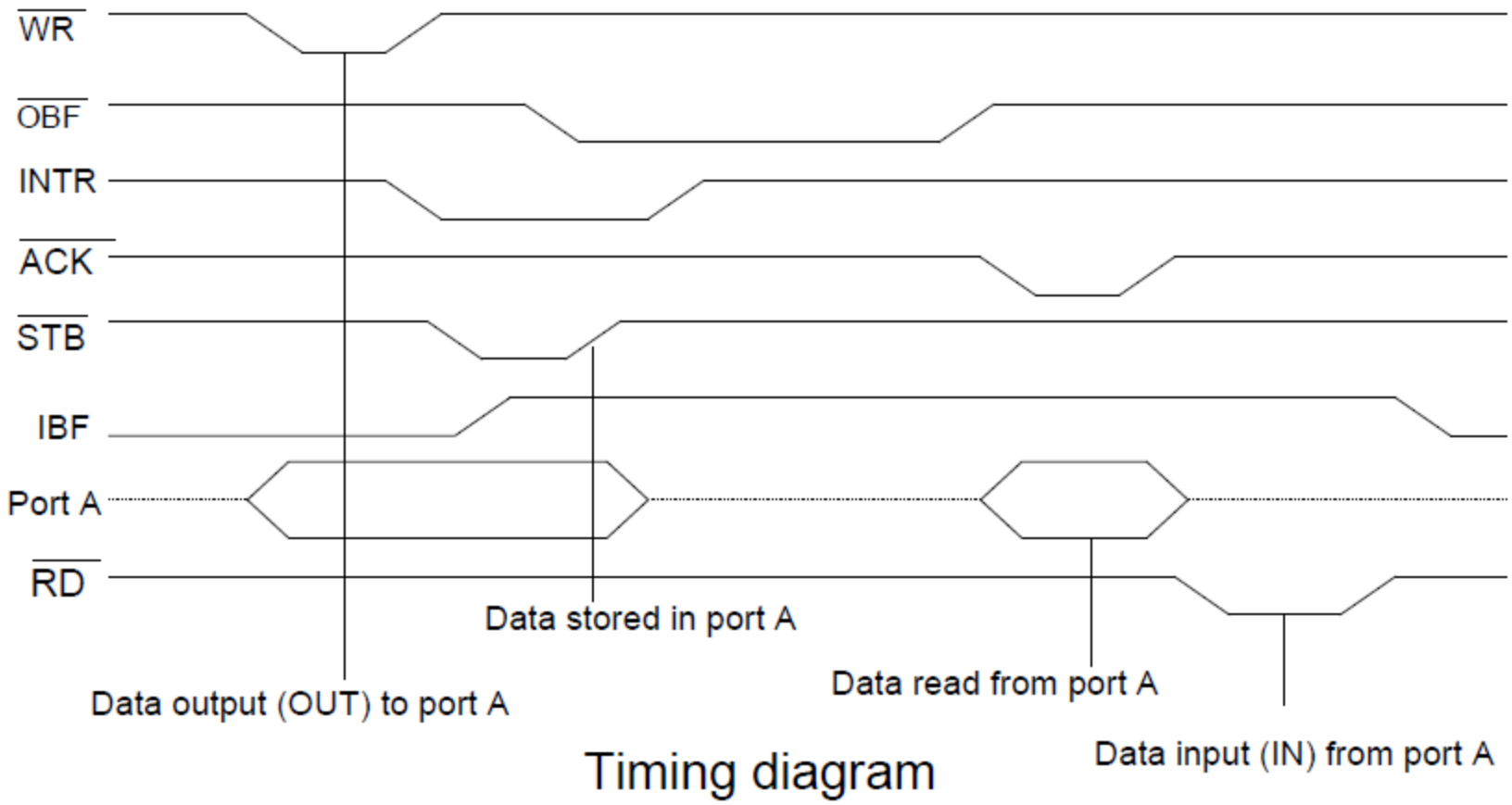
Timing diagram

Chế độ vào ra 2 chiều

- Cung cấp vào/ra 2 chiều cho kênh dữ liệu 8 bít
- Các tín hiệu kết nối dùng như trong chế độ 1.
- Chỉ dùng nhóm A



Biểu đồ thời gian



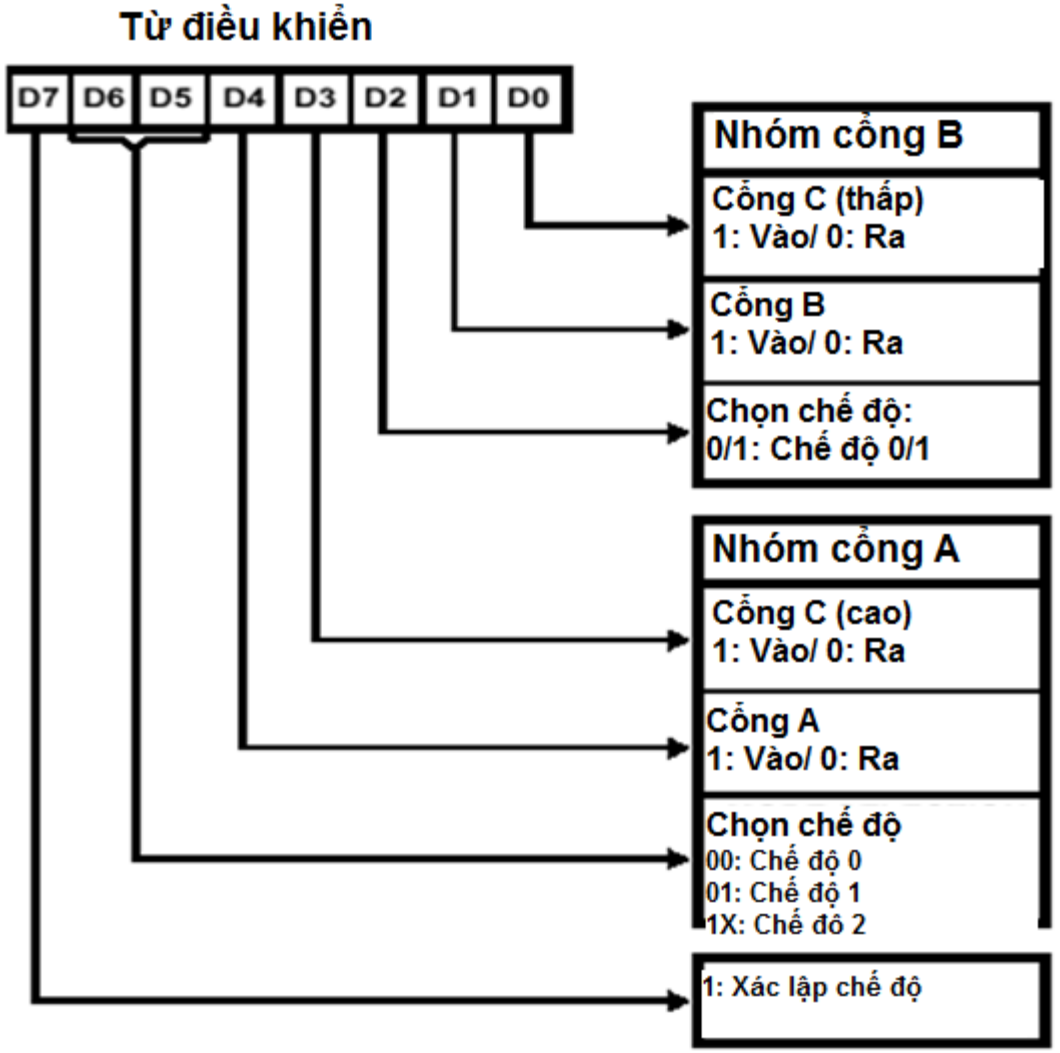
Timing diagram

Lập trình PPI

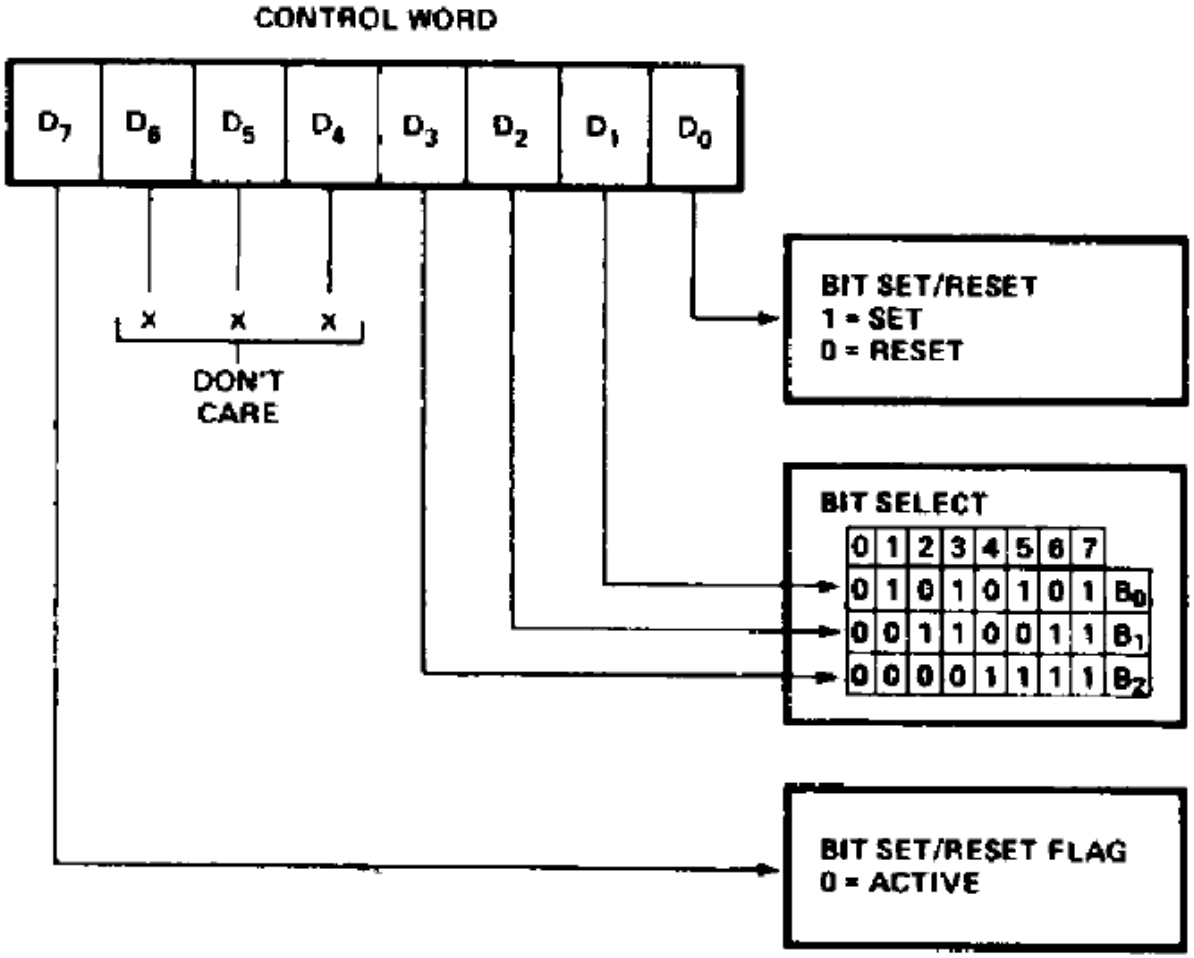
- Từ khóa điều khiển chế độ
- Từ khóa điều khiển trạng thái (cổng C)

Từ khóa điều khiển chế độ

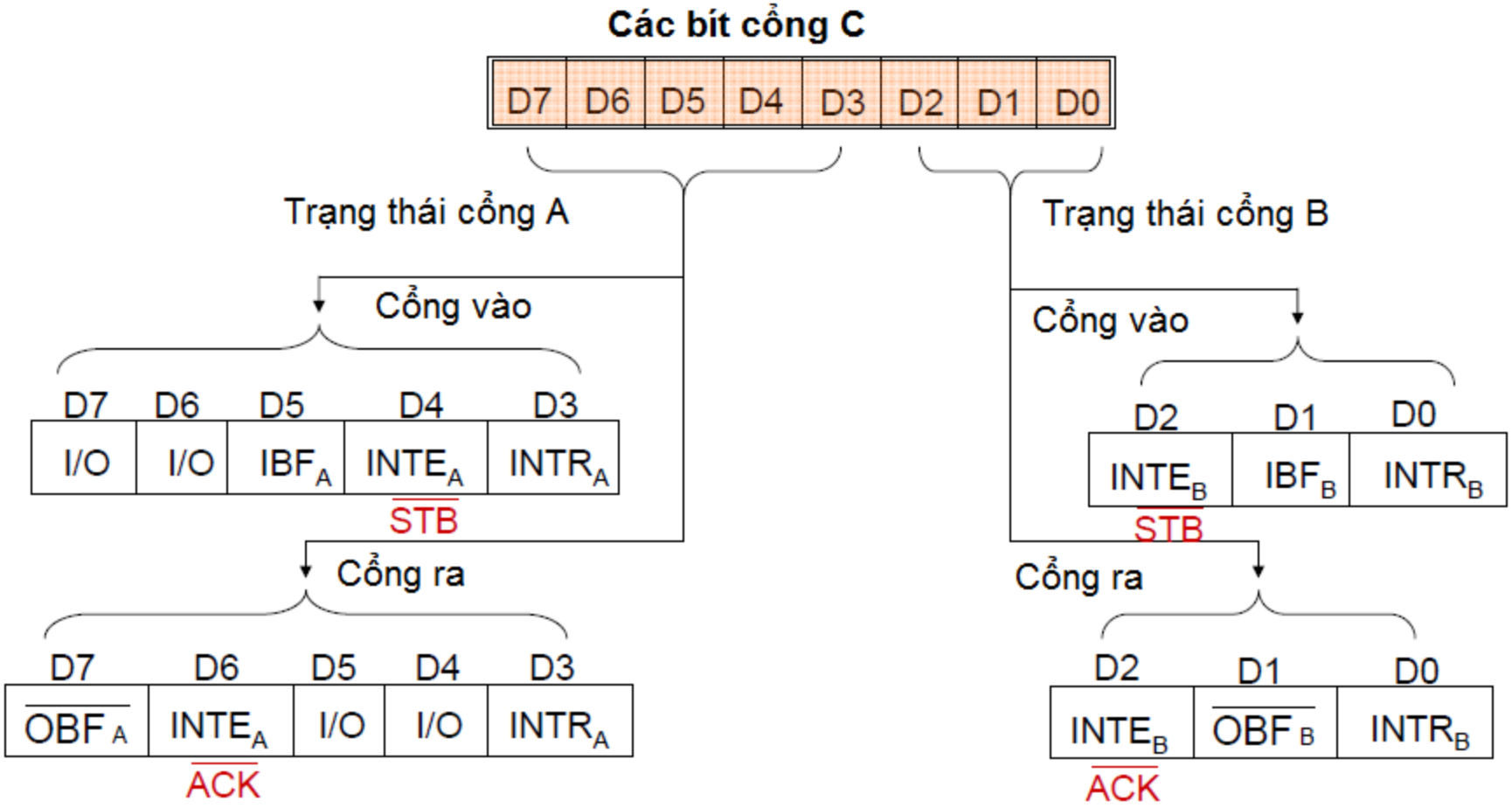
A1	A0	Chức năng
0	0	Cổng A
0	1	Cổng B
1	0	Cổng C
1	1	Điều khiển



Từ khóa điều khiển trạng thái cổng C



Từ điều khiển trạng thái



Một số ví dụ

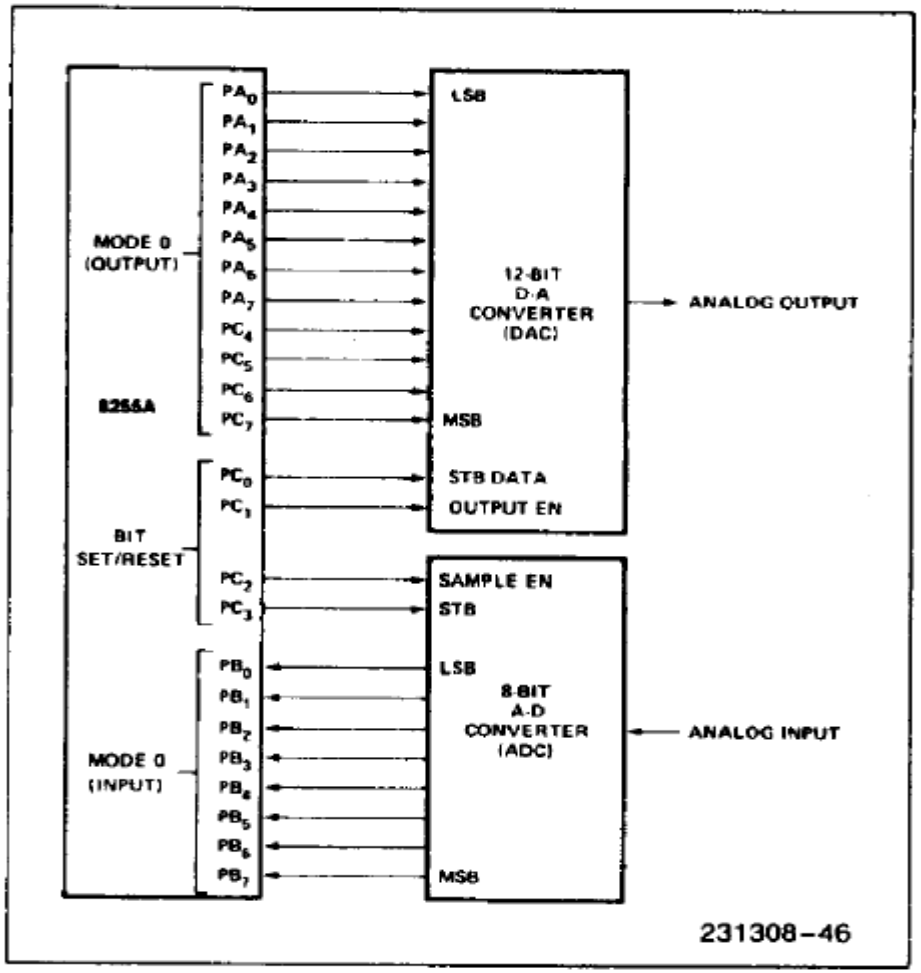


Figure 22. Digital to Analog, Analog to Digital

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

Kỹ Thuật Vi Xử Lý

Giảng viên: TS. Phạm Hoàng Duy

Điện thoại/E-mail: phamhduy@gmail.com

Bộ môn: Khoa Học Máy Tính- Khoa CNTT1

Học kỳ/Năm biên soạn:2009

NỘI DUNG

Ghép nối dữ liệu nối tiếp

Giảng viên: TS. Phạm Hoàng Duy
E-mail: phamhduy@gmail.com
Năm biên soạn: 2009

Nội dung

- Giới thiệu
- 8251

Truyền dữ liệu nối tiếp

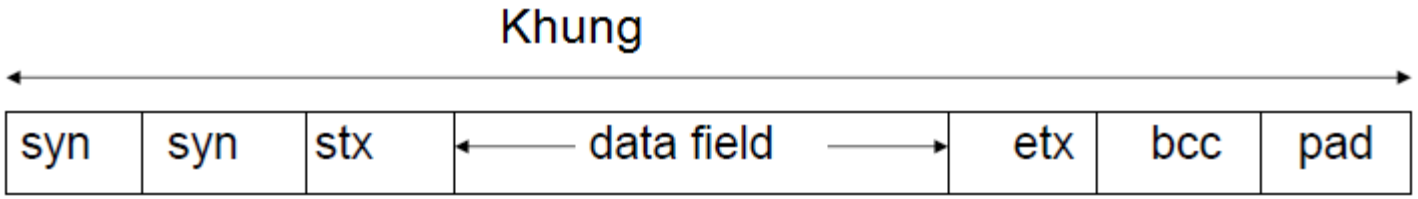
- Truyền nối tiếp thường truyền dữ liệu trên 1 dây, từng bít một. Giao tiếp vào ra sẽ chuyển đổi byte dữ liệu thành chuỗi bít
- Kết nối nối tiếp thường dùng để truyền dữ liệu đi xa. Chuẩn nối tiếp phổ biến là RS232

Truyền dữ liệu nối tiếp

- Truyền dữ liệu nối tiếp đồng bộ
- Truyền dữ liệu nối tiếp dị bộ

Truyền đồng bộ

- Dữ liệu được truyền thành các khối với khối bắt đầu và kết thúc. Các ký tự riêng lẻ trong khối không có các các bit khởi đầu và kết thúc.
- Ví dụ khung truyền



syn = Ký tự đồng bộ (ASCII 16)

stx = Bắt đầu đoạn text (ASCII 02)

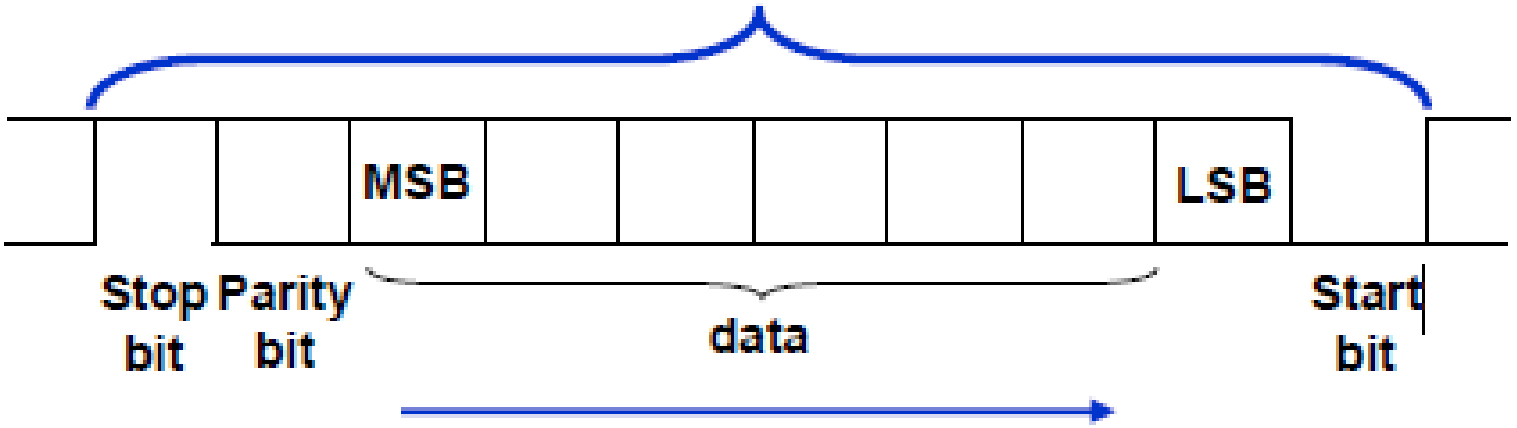
etx = Kết thúc đoạn text (ASCII 03)

bcc = Các ký tự kiểm tra lỗi

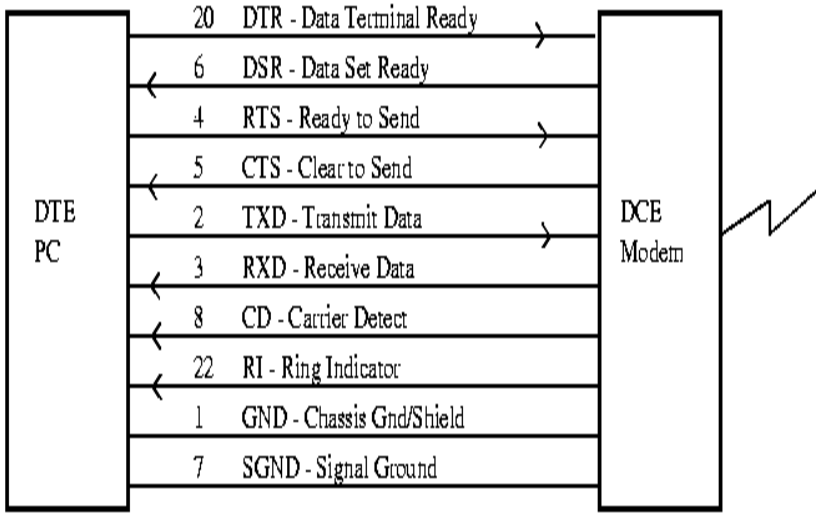
pad = Ký tự đệm (ASCII FFH)

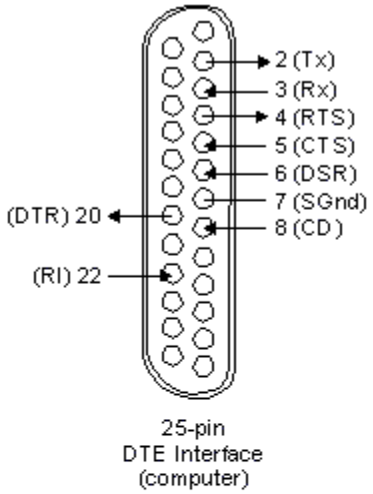
Truyền dữ liệu dị bộ

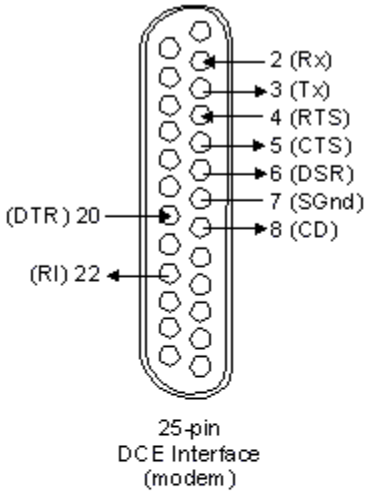
- Không có các bít đồng bộ khối. Các ký tự được nhận biết thông qua các bít khởi đầu (start) và kết thúc (stop).
- Các bít khởi đầu, kết thúc được chèn vào đầu và cuối ký tự
- Ví dụ đoạn dữ liệu

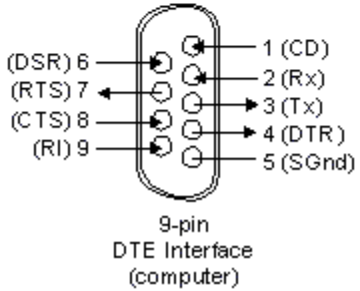


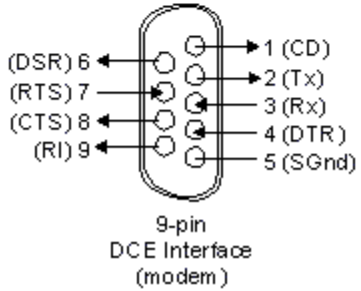
Tín hiệu kết nối RS232



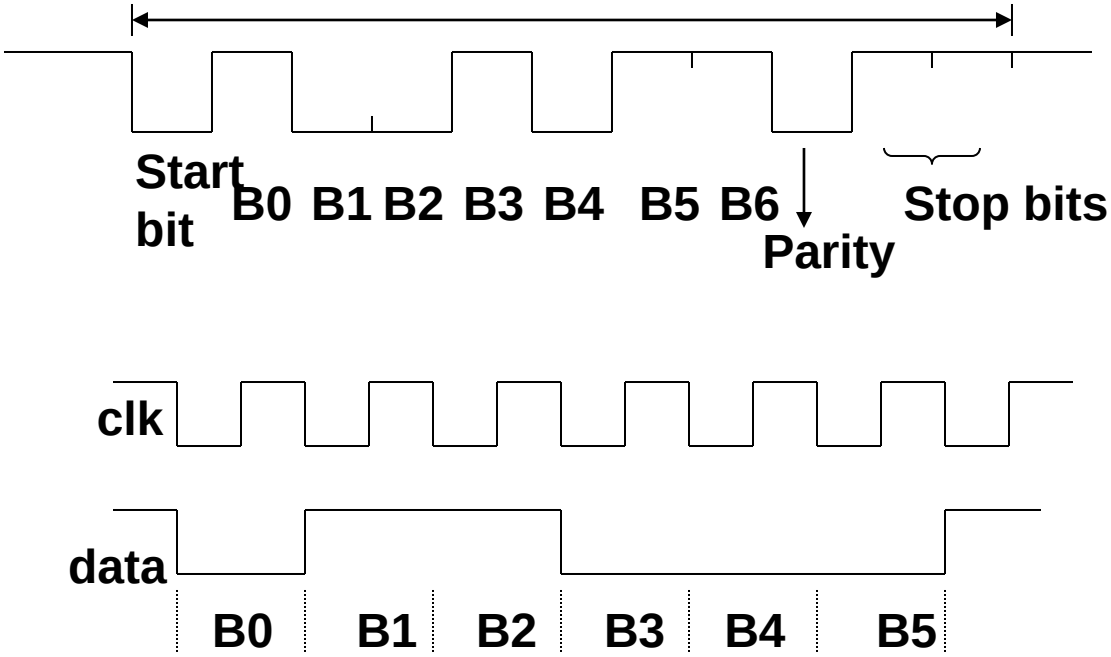








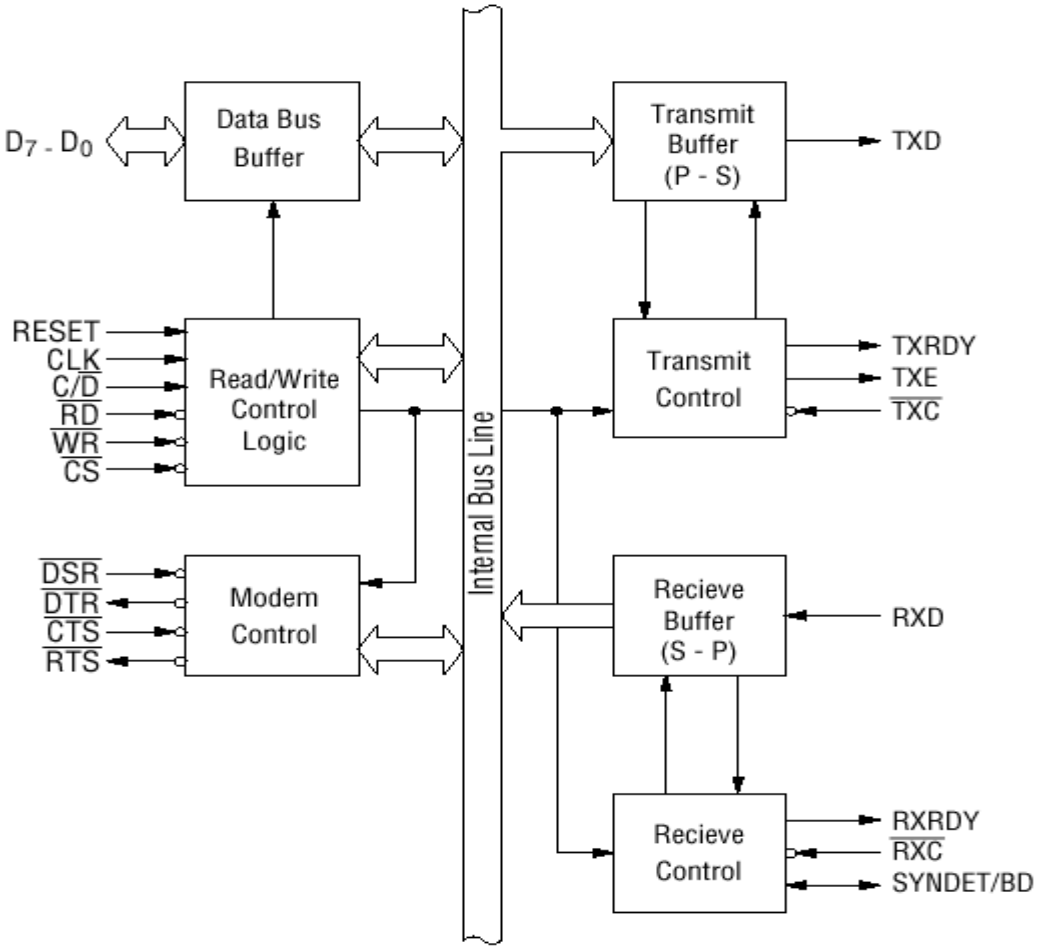
Truyền dữ liệu



Intel 8251

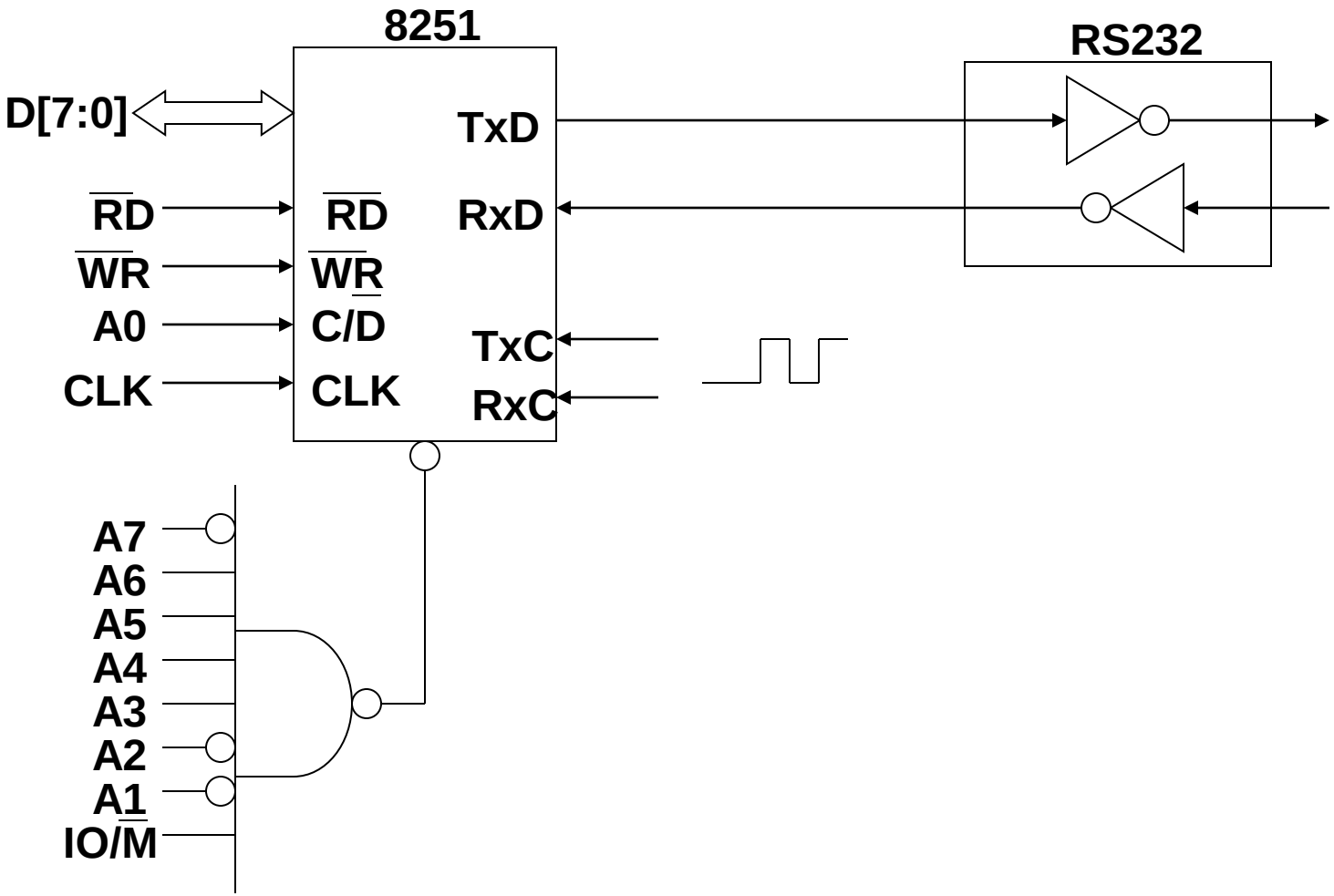
- Điều khiển truyền thông đồng bộ và dị bộ
- Đồng bộ hỗ trợ:
  - Định dạng 5-8 bít/ ký tự
  - Tự động chèn ký tự đồng bộ
  - Tốc độ 64K baud
- Dị bộ
  - Định dạng 5-8 bít/ ký tự
  - 1,2 bít stop
  - Tốc độ 19.2K baud

Intel 8251



- Các tín hiệu kết nối VXL
  - D0-D7
  - C/D~: Control/Data
  - RD~, WR~,CS~
- Tín hiệu đ/k modem
  - RS-232
- Tín hiệu đường truyền
  - TxD,RxD: Thu, phát
  - TxRDY,RxRDY: sẵn sàng
  - TxE:đệm rỗng
  - SynDET/BD: phát hiện đồng bộ/Gián đoạn

Ghép nối

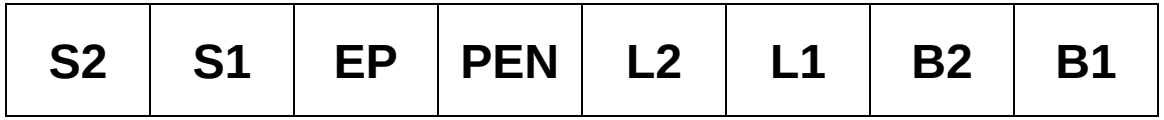

8251
RS232
D[7:0]
TxD
RD
RD
RxD
WR
WR
A0
C/D
TxC
CLK
CLK
RxC
A7
A6
A5
A4
A3
A2
A1
IO/M

Lập trình 8251

- Các thanh ghi chế độ
- Các thanh ghi trạng thái

C/D (A0)	RD	WR	Thanh ghi
0	0	1	Thanh ghi đệm thu (Đọc)
0	1	0	Thanh ghi đệm phát (Ghi)
1	0	1	Thanh ghi trạng thái (Đọc)
1	1	0	Thanh ghi điều khiển (Ghi)

Thanh ghi chế độ (Dị bộ)



Thanh ghi chế độ

Số bít Stop

00: XX
01: 1 bit
10: 1.5 bits
11: 2 bits

Parity
0: Lẻ
1: Chẵn

Dùng Parity
0: Không
1: Có

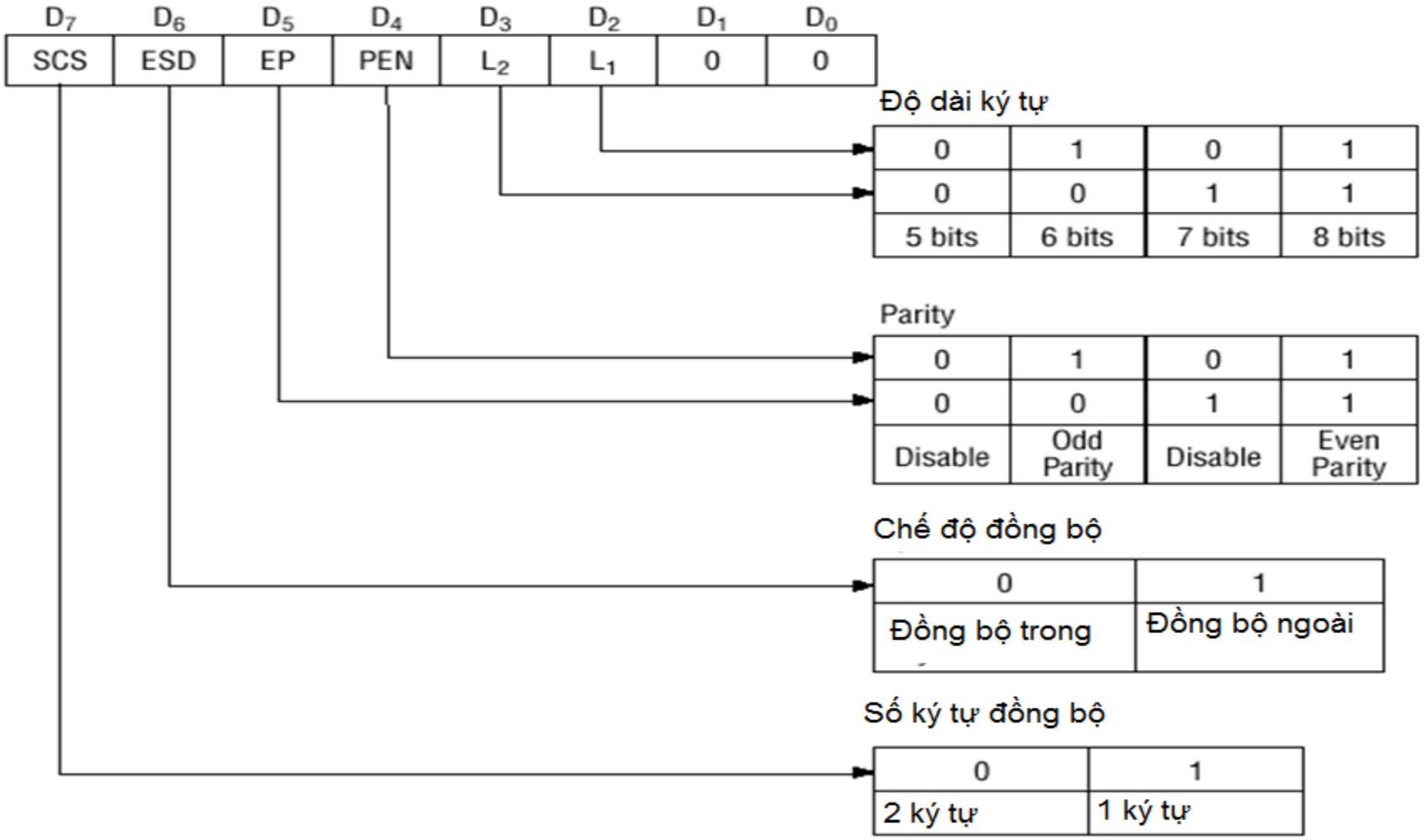
Độ dài ký tự

00: 5 bits
01: 6 bits
10: 7 bits
11: 8 bits

Baud Rate

00: Đồng bộ
01: x1
10: x16
11: x64

Thanh ghi chế độ (Đồng bộ)



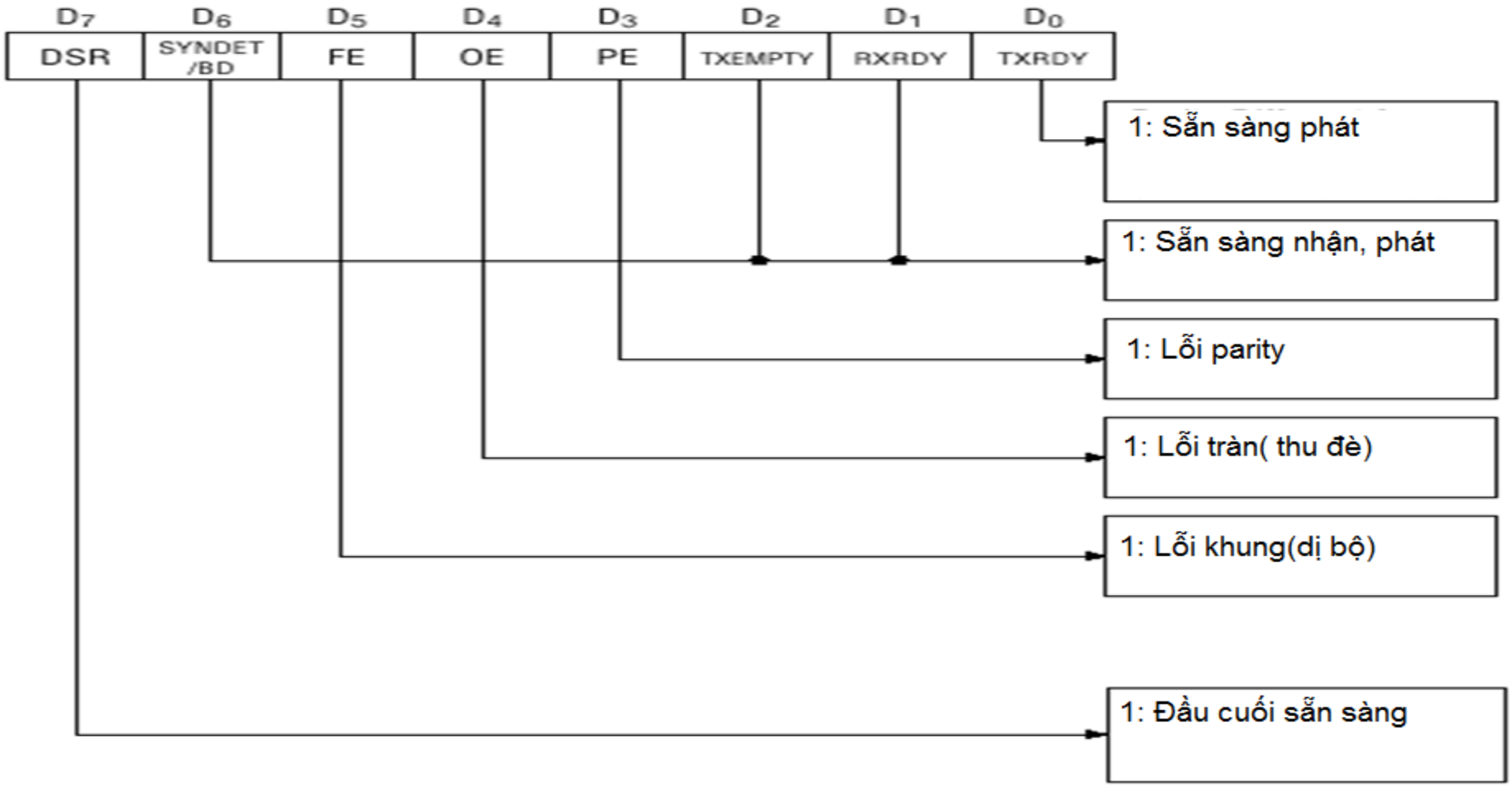
Thang ghi lệnh

EH	IR	RTS	ER	SBRK	RxE	DTR	TxE

Thanh ghi lệnh

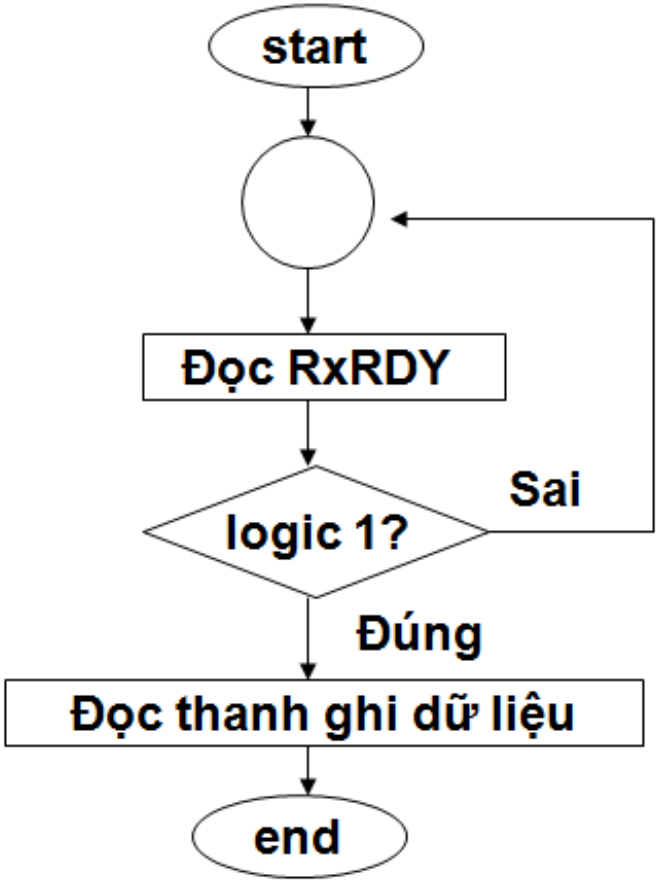
TxE: Cho phép truyền(=1)
DTR: Đầu cuối dữ liệu sẵn sàng(=1)
RxE: Cho phép nhận (=1)
SBPRK:Gửi ký tự gián đoạn(1)
ER: Xóa cờ(=1)
RTS: Yêu cầu gửi (=1)
IR: Khởi động lại(=1)
EH: Tìm ký tự đồng bộ(=1)

Thanh ghi trạng thái



Lưu đồ đọc ghi đơn giản

❑Đọc



❑ Ghi