

VÀO/RA DỮ LIỆU

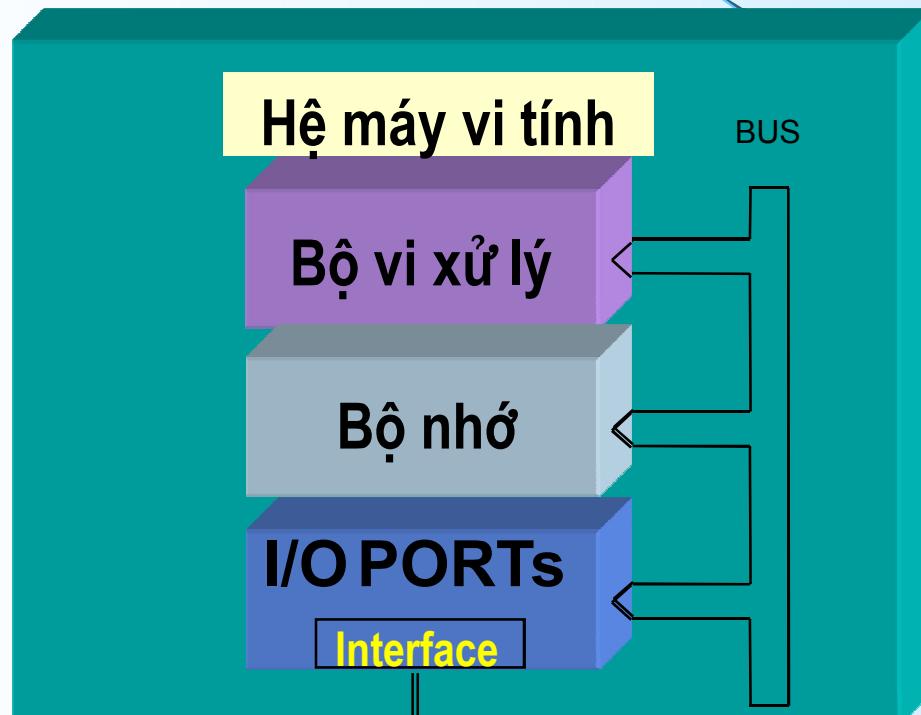
Nội dung:

Giới thiệu chung

1. Các lệnh vào/ra
2. Các cổng vào/ra
3. Vào/ra bằng cách thăm dò
4. Vào/ra bằng ngắt
5. Vào/ra bằng DMA
6. Vào/ra nối tiếp

GIỚI THIỆU CHUNG

Hệ thống vào/ra là nhằm đảm bảo sự liên hệ giữa hệ vi xử lý với các thiết bị ngoại vi.



Vào/Ra

CÁC LỆNH VÀO/RA

- Vào/ra riêng biệt:

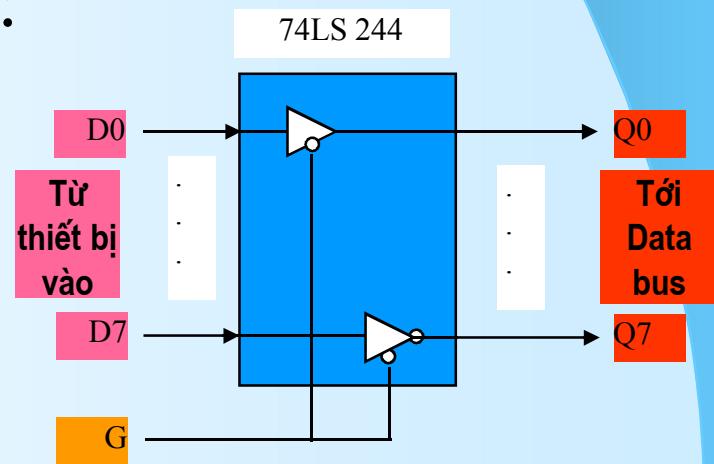
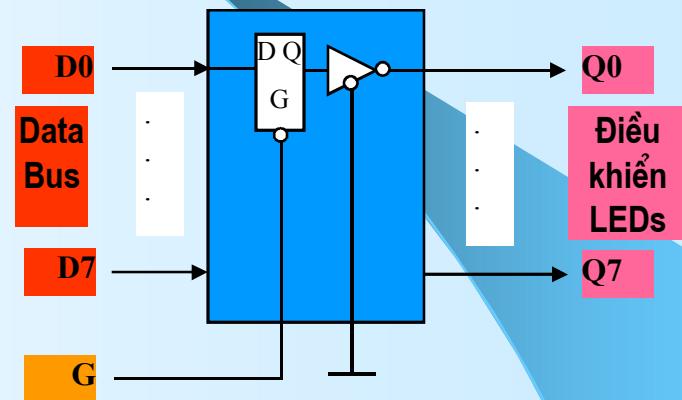
Địa chỉ Trực tiếp	Địa chỉ gián tiếp
IN AL,#port	IN AL,DX
IN AX, #port	IN AX, DX
OUT #port,AL	OUT DX,AL
OUT #port,AX	OUT DX,AX

- Vào/ra theo địa chỉ bộ nhớ:

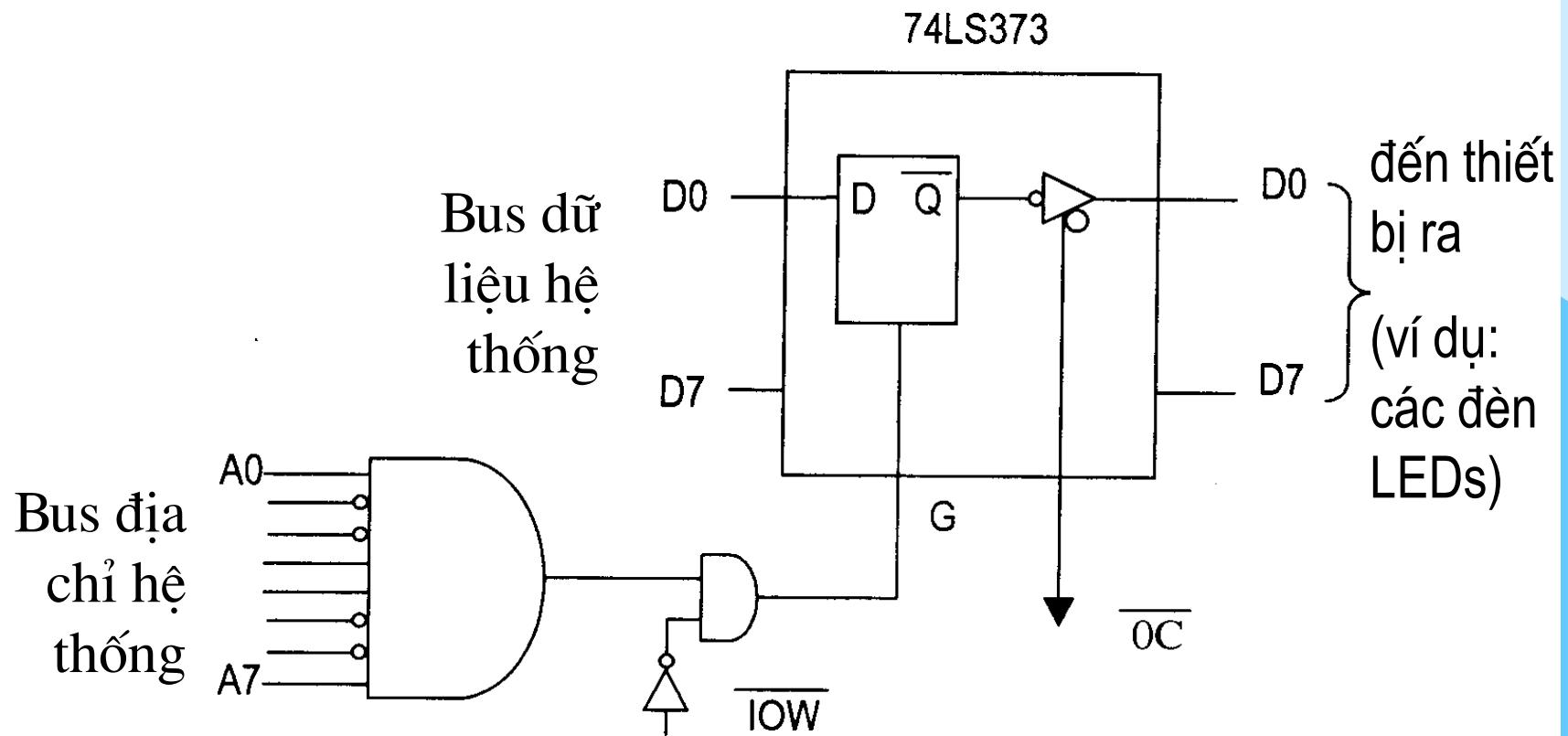
- **Vào:** MOV AL, mem
- **Ra:** MOV mem, AL

2. CÁC CỔNG VÀO/RA

- Cổng vào/ra đơn giản:
 - Cổng ra dùng 74LS373: (địa chỉ cổng 60h
Thiết kế mạch giải mã)
 - Cổng vào dùng 74LS244:
 - Địa chỉ 43h)

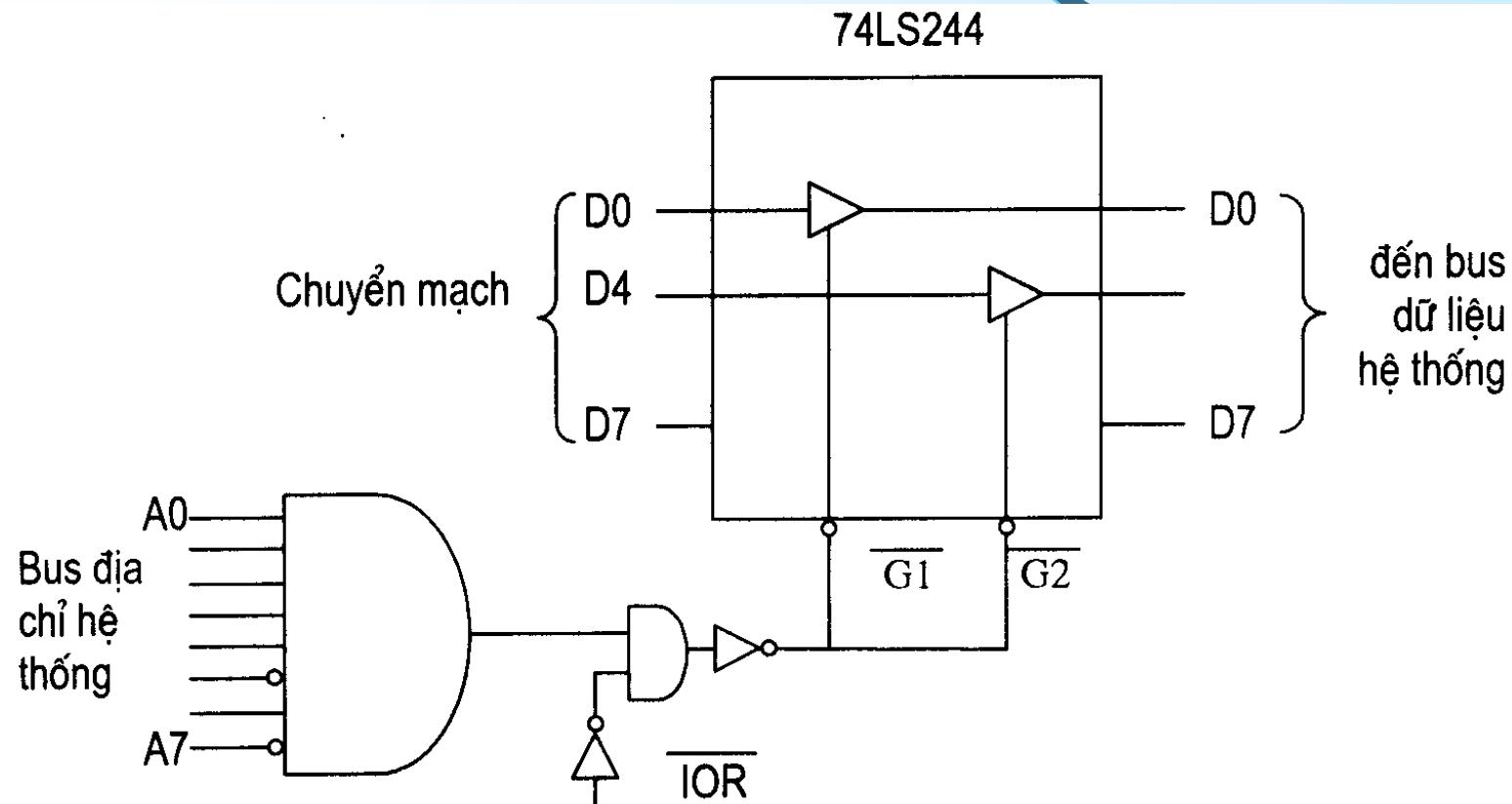


Ví dụ cổng ra



Đưa dữ liệu ra cổng 99h từ AL bằng lệnh: **OUT 99h ,AL**

Ví dụ cổng vào



Đọc dữ liệu vào AL từ cổng 57h bằng lệnh: **IN AL, 57h**

Quản lý quá trình Vào/Ra

Vào/ra bằng cách thăm dò

- Vào/ra dữ liệu được CPU thực hiện thông qua các cổng.
- CPU thăm dò trạng thái sẵn sàng của cổng thông qua các tín hiệu bắt tay (*Handshake Signals*).
- Các tín hiệu thăm dò lấy từ các mạch phối ghép được thiết kế.

VÀO/RA BẰNG NGẮT

- Ngắt: CPU tạm dừng công việc hiện tại để trao đổi dữ liệu.
- Được sử dụng để tận dụng khả năng xử lý của CPU cho nhiều công việc.
- Có hai loại ngắt: Ngắt cứng và ngắt mềm.

1. Ngắt cứng: Được phát sinh bởi các mạch của máy tính khi đáp lại 1 sự kiện nào đó. Nó được dùng để điều khiển các thiết bị quan trọng trên hệ máy như bàn phím, ổ đĩa, máy in..... Các ngắt này được linh kiện 8259A quản lý.

● **2. Ngắt mềm (Software Interrupt)**

Do chương trình yêu cầu để thực hiện 1 chương trình con nào đấy có sẵn trong ROM hoặc trong RAM. Ngắt mềm được gọi bởi lệnh INT.

Máy tính IBM PC XT có 8 ngắt cứng là:

- | | |
|------|-------------------|
| IRQ0 | Timer |
| IRQ1 | Keyboard |
| IRQ2 | Dùng trong máy AT |
| IRQ3 | Serial port 2 |
| IRQ4 | Serial port 1 |
| IRQ5 | Parallel port 2 |
| IRQ6 | Floppy disk |
| IRQ7 | Parallel port 1 |

Các máy PCs sử dụng Bus ISA có 15 ngắt cung:

IRQ	Address	Description	Detected	Handled By
0	CC00:0000	Timer Click	Yes	Unknown
1	0A2C:08D2	Keyboard	Yes	KEYB
2	F000:FF47	Second 8259A	Yes	BIOS
3	F000:FF47	COM2: COM4:	COM2:	BIOS
4	F000:FF47	COM1: COM3:	COM1:	BIOS
5	F000:FF47	LPT2:	No	BIOS
6	0956:009A	Floppy Disk	Yes	Default Handlers
7	0070:0465	LPT1:	Yes	System Area
8	0956:0035	Real-Time Clock	Yes	Default Handlers
9	F000:F7C9	Redirected IRQ2	Yes	BIOS
10	F000:FF47	(Reserved)		BIOS
11	F000:FF47	(Reserved)		BIOS
12	0956:00E2	(Reserved)	Not Detected	Default Handlers
13	F000:2930	Math Coprocessor	Yes	BIOS
14	0956:00F1	Fixed Disk	Yes	Default Handlers
15	F000:FF47	(Reserved)		BIOS

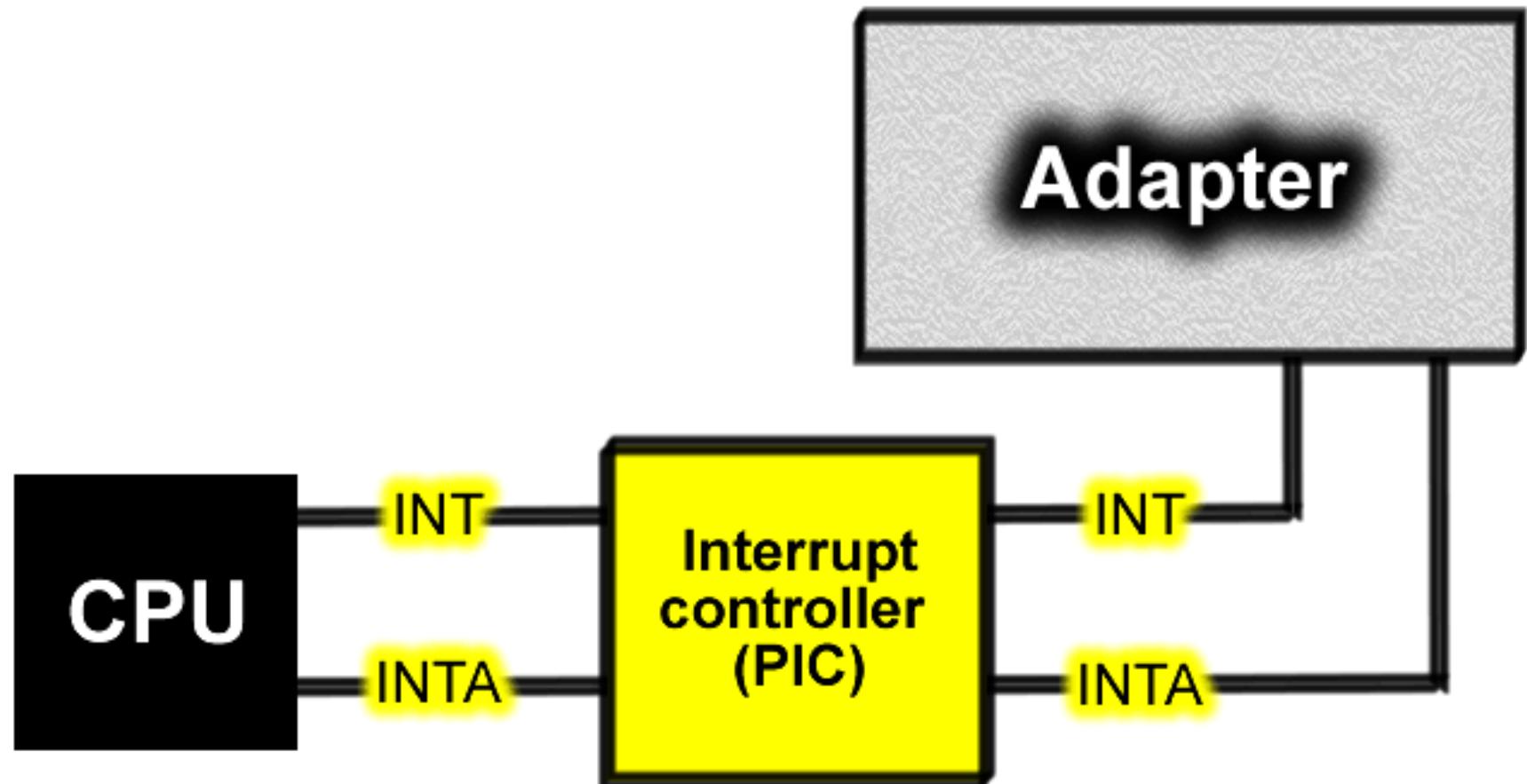
Shared IRQs

Trên các hệ Bus hiện đại như MCA, EISA, PCI có khả năng chia sẻ IRQs. Nghĩa là 2 ngoại vi có thể sử dụng một IRQ. Khi IRQ có mức tích cực, Chương trình điều khiển của 2 ngoại vi sẽ kiểm tra và nhận dạng ngoại vi nào trên Bus có yêu cầu ngắt.

-  (PCI) 11 Creative SB Audigy
-  (PCI) 11 D-Link DFE-530TX PCI Fast Ethernet Adapter (Rev A) #2
-  (PCI) 11 HPT370 UDMA/ATA 100 RAID Controller
-  (PCI) 11 OHCI Compliant IEEE 1394-værtscontroller
-  (PCI) 11 RADEON VE
-  (PCI) 11 VIA USB universel værtscontroller

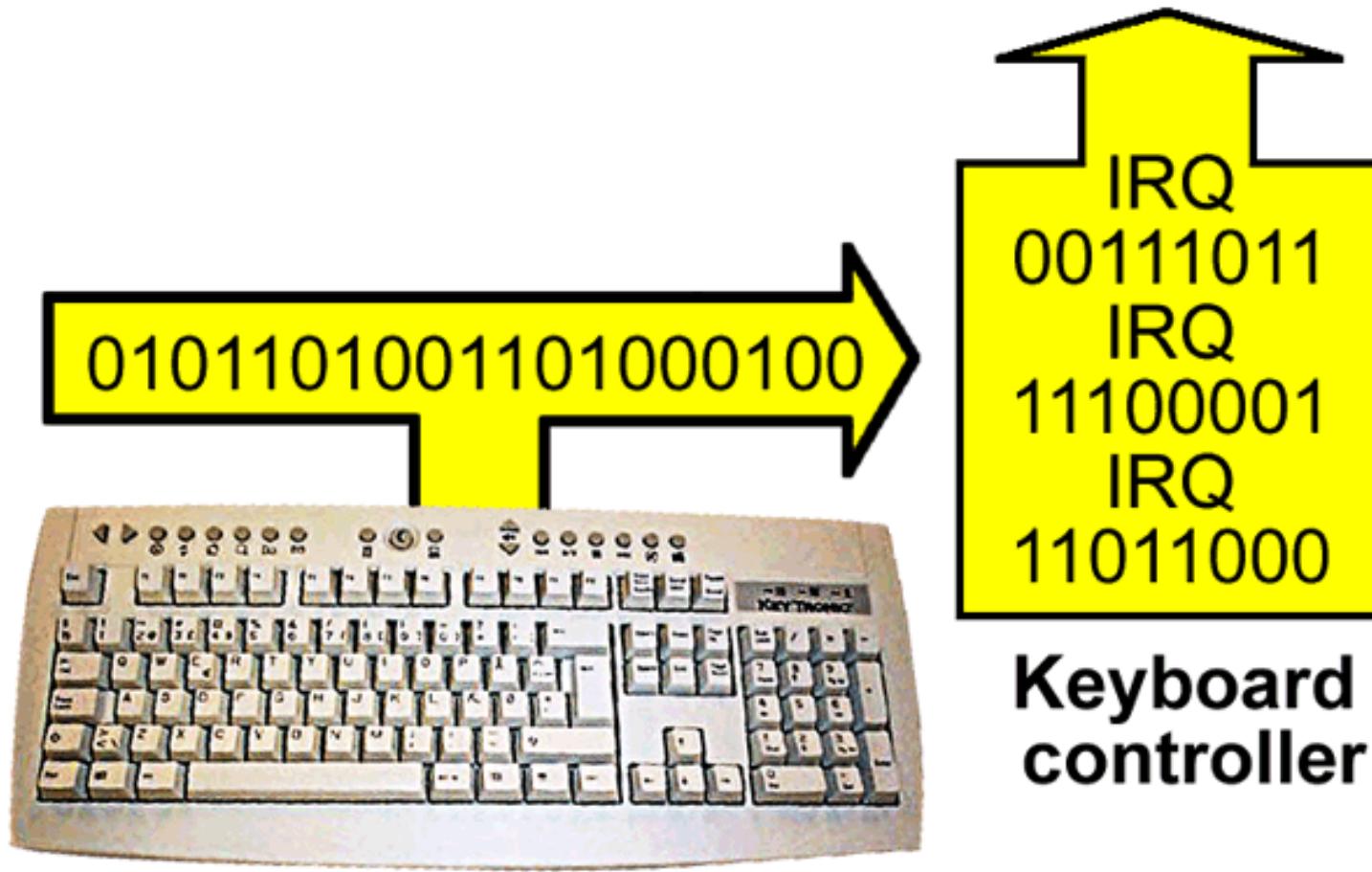
The following devices have been listed as located on the PCI bus, and they all use IRQ 11.

IRQ and conflicts on the ISA bus?



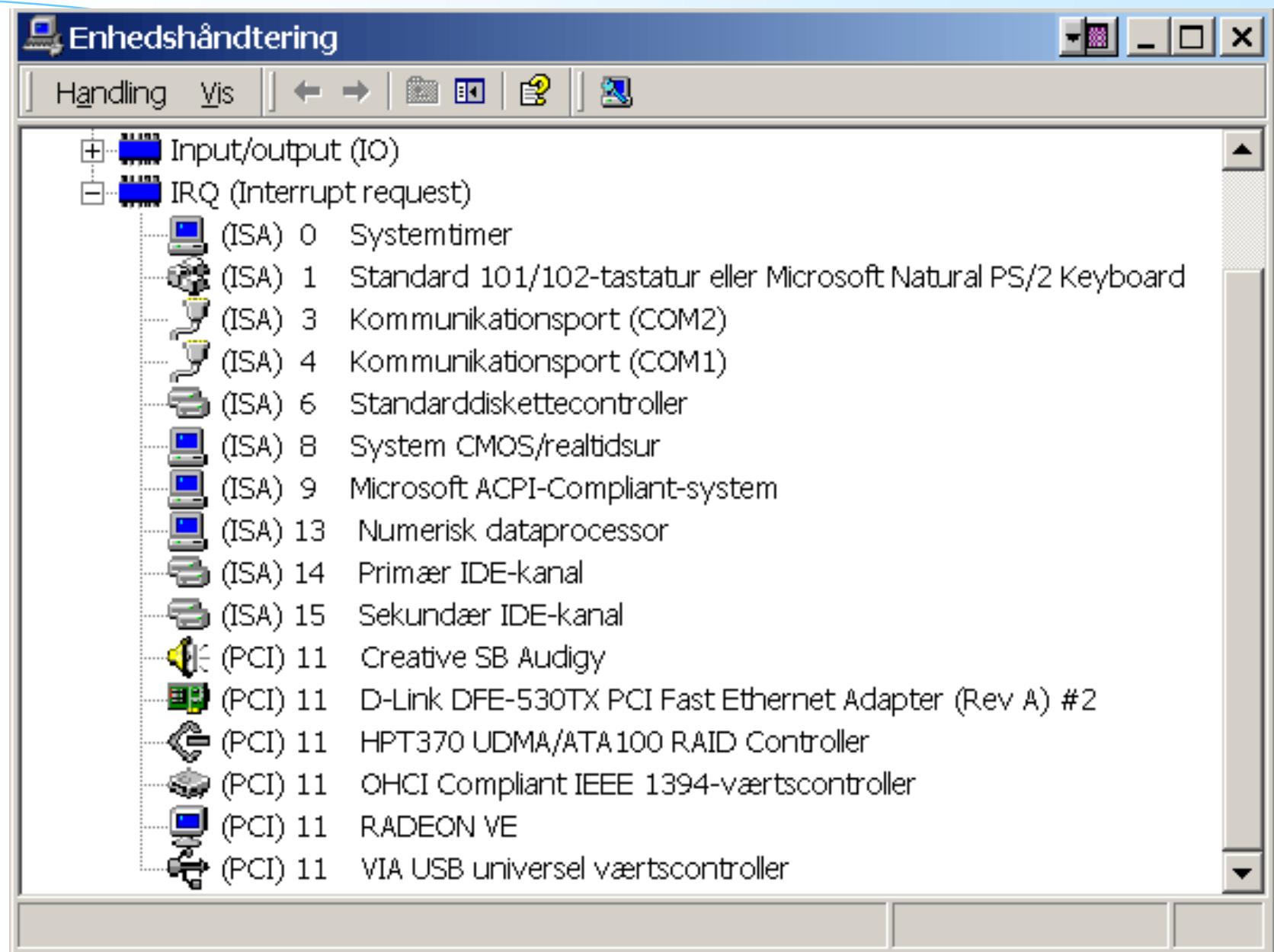
The interrupt controller is an important little controller. It communicates with the CPU using interrupt signals and regulates the traffic to and from the I/O devices.

I/O bus and IRQ controller



The keyboard controller delivers its data to the I/O bus when it gets permission from the PIC.

- Every time you press a key, an IRQ is sent from the *keyboard controller* to the CPU. First, the keyboard sends its data as a series of individual bits along the keyboard cable. The bits are received on the motherboard by the keyboard controller. On modern PC's this function is included in the Super I/O controller, but it used be on a separate 8042 chip. When the keyboard controller has collected one byte – that is one character – it sends an IRQ signal to the PIC
- The signal requests permission from the CPU to send the character on. The PIC notifies the keyboard controller when the way is clear to the CPU, and then the byte is moved onto the I/O bus. At the physical level, an IRQ is a printed track on the bus. The printed track connects to all the PCI slots on the motherboard, so no matter which slot the adapter is installed in it has contact with the IRQ system.



Windows reports on the distribution of IRQ's

Bảng vec-tơ ngắt và chương trình phục vụ ngắt

- ✓ *Bảng véc-tơ ngắt*: Có chiều dài 1 KB nằm trong bộ nhớ bắt đầu ở địa chỉ 0000:0000 đến địa chỉ 0000:3FF. Gồm 256 phần tử, mỗi phần tử gồm 4 byte. Phần tử thứ i của vector ngắt được dùng để chứa địa chỉ logic của chương trình con xử lý ngắt thứ i.
- ✓ *Chương trình con phục vụ ngắt* (ISR-Interrupt Service Routine): được thực hiện khi kích hoạt ngắt (bằng phần cứng hay mềm).
- Mỗi ngắt có một ISR tương ứng.
- Địa chỉ của ISR nằm trong bảng vectơ ngắt. Một địa chỉ gồm 4 byte (2 byte offset, 2 byte segment)
- Địa chỉ vật lý của ISR_{nn} = (số hiệu ngắt nn) x 4
- Máy tính PC có 256 ngắt (INT 00 - INT FF). Bảng vec-tơ ngắt chiếm 1KB, từ 00000 đến 003FFh.

Quá trình xử lý ngắn:

Ngăn xếp (Stack) và con trỏ ngăn xếp (Stack pointer).

Stack: là một vùng nhớ được máy tính dùng để chứa trạng thái của BVXL khi chương trình thực hiện các lệnh Push, POP và khi máy tính thực hiện các thao tác ngắn.

Khi chạy các file.com; Stack được tự động khởi tạo, cùng với CS, DS, ES ở trong cùng một đoạn và có kích thước quy định bởi chương trình.

Khi chạy các file.exe; Stack segment được khởi tạo và chiếm trọn 1 segment.

Stack pointer (SP): Là một thanh ghi 16bits luôn chỉ đến đỉnh Stack. Đỉnh Stack là ô nhớ nơi mà BVXL sẽ thực hiện cất giữ và hồi phục trạng thái của BVXL.

Xử lý ngắt

BVXL xử lý ngắt (mềm hoặc cứng) qua các bước sau:

1. Cắt thanh ghi cờ (FR) vào Stack và giảm SP đi 2
2. Đặt IF=0 và TF=0.
3. Cắt CS hiện hành vào stack và giảm SP đi 2.
4. Cắt IP hiện hành vào stack và giảm SP đi 2.
5. Nhân số hiệu INT với 4 để xác định địa chỉ vật lý của ngắt trong bảng vec tơ ngắt, qua đó có được CS và IP của chương trình con phục vụ ngắt ISR.
6. Với CS:IP mới, CPU bắt đầu nhận và thực hiện các lệnh của chương trình con phục vụ ngắt ISR.
7. Lệnh cuối cùng của ISR là lệnh IRET. Lệnh này thông báo để CPU tải lại giá trị IP, CS và FR từ stack và CPU có thể chạy tiếp chương trình tại nơi nó đã bị ngắt.

Ví dụ:

1. Sử dụng DEBUG để hiển thị nội dung của các ô nhớ 00000-0002FH.
2. Tìm CS:IP của INT 0h và INT 8h.

C>debug

```
-D 0000:0000-002F
0000:0000 E8 56 2B 02 56 07 70 00-C3 E2 00 F0 56 07 70 00...
0000:0010 56 07 70 00 54 FF 00 F0-47 FF 00 F0 47 FF 00 F0...
...
000:0020 A5 FE 00 F0 87 E9 00 F0-DD E6 00 F0 DD E6 00 F0...
```

INT 0h; CS:IP được đặt ở địa chỉ 0, 1, 2, 3. Do đó IP=56E8H và CS=022BH.

CS:IP của INT 8 bắt đầu từ địa chỉ 00020h vì $8 \times 4 = 32 = 20H$, nên chúng ta có IP = FEA5, CS = F000.

Mạch phối ghép vào/ra song song khả trình 8255A

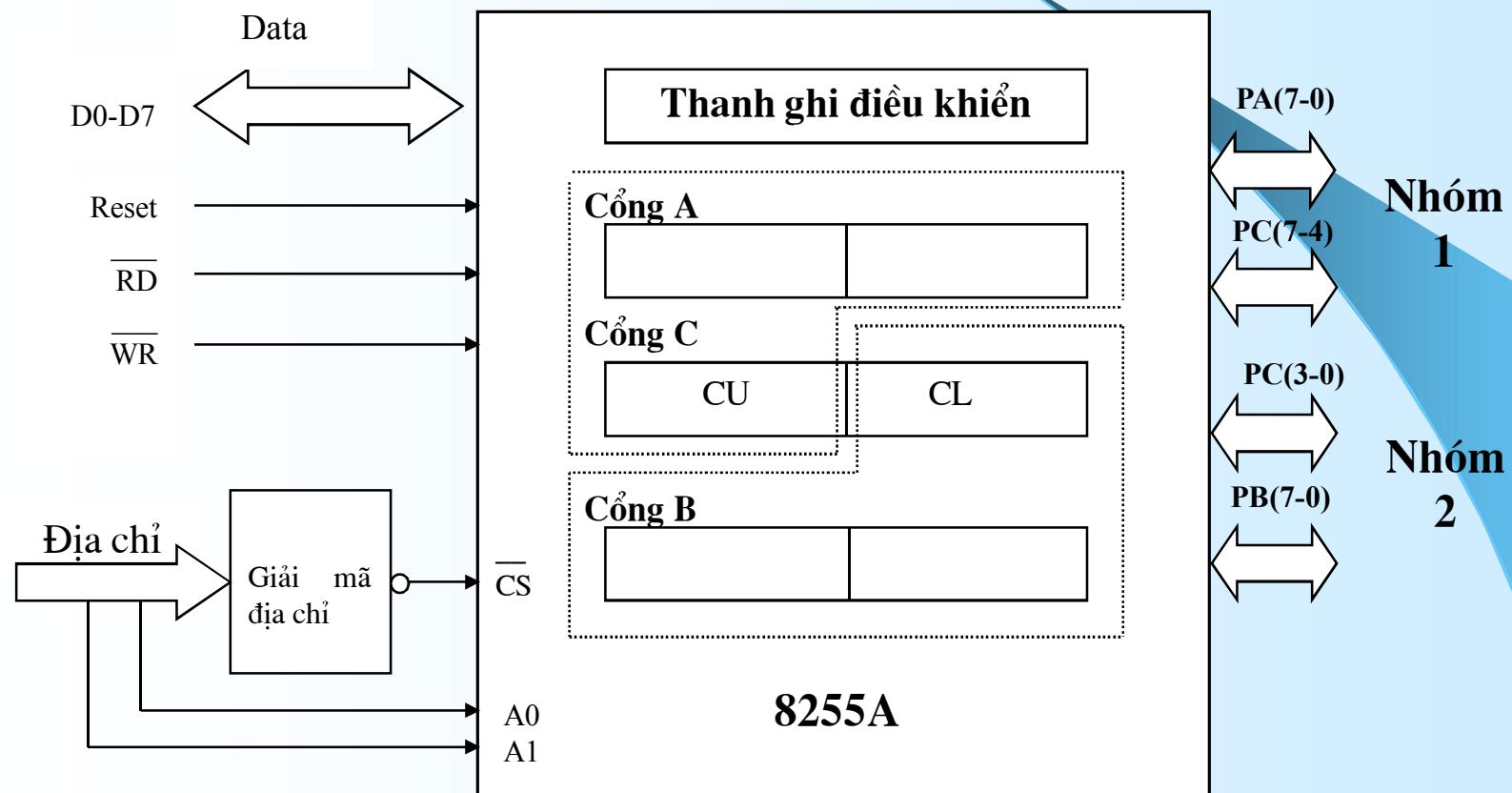
Chip PPI 8255

Chức năng: hỗ trợ lập trình
vào/ra song song đơn giản,
hoặc có bắt tay.

Thành phần: có 3 thanh ghi
vào ra 8 bit chia thành 2 nhóm
(A, B) và 1 thanh ghi điều
khiển (CW).

PA3	1	40	PA4
PA2	2	39	PA5
PA1	3	38	PA6
PA0	4	37	PA7
RD	5	36	WR
CS	6	35	RESET
GND	7	2	D0
A1	8	33	D1
A0	9	32	D2
PC7	10	31	D3
PC6	11	30	D4
PC5	12	29	D5
PC4	13	28	D6
PC0	14	27	D7
PC1	15	26	Vcc
PC2	16	25	PB7
PC3	17	24	PB6
PB0	18	23	PB5
PB1	19	22	PB4
PB2	20	21	PB3

Sơ đồ khối Intel8255

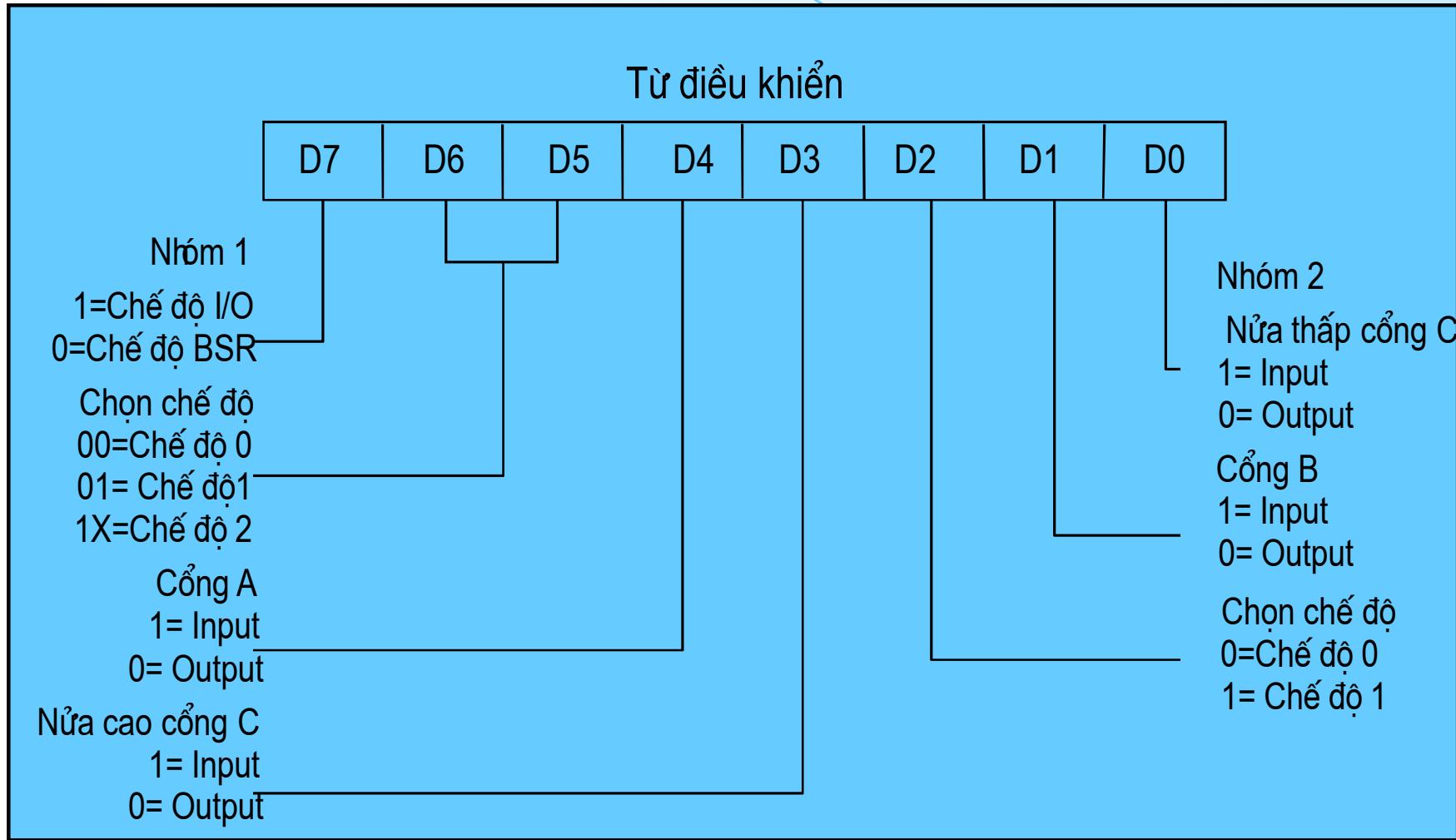


Sơ đồ cấu trúc vi mạch 8255 và ghép nối

Chọn cổng và chiều truyền dữ liệu của 8255

A1	A0	\overline{RD}	\overline{WR}	\overline{CS}	Transfer Description
0	0	0	1	0	Port A to Data Bus
0	1	0	1	0	Port B to Data Bus
1	0	0	1	0	Port C to Data Bus
0	0	1	0	0	Data Bus to Port A
0	1	1	0	0	Data Bus to Port B
1	0	1	0	0	Data Bus to Port C
1	1	1	0	0	Data Bus to Control Register if D7=1, if D7=0 Input from the Data bus is treated as a Set/Reset Instruction
X	X	X	X	1	D7-D0 go to High - Impedance State
1	1	0	1	0	Illegal Combination
X	X	1	1	0	D0-D7 go to High - Impedance State

Tùy điều khiển 8255



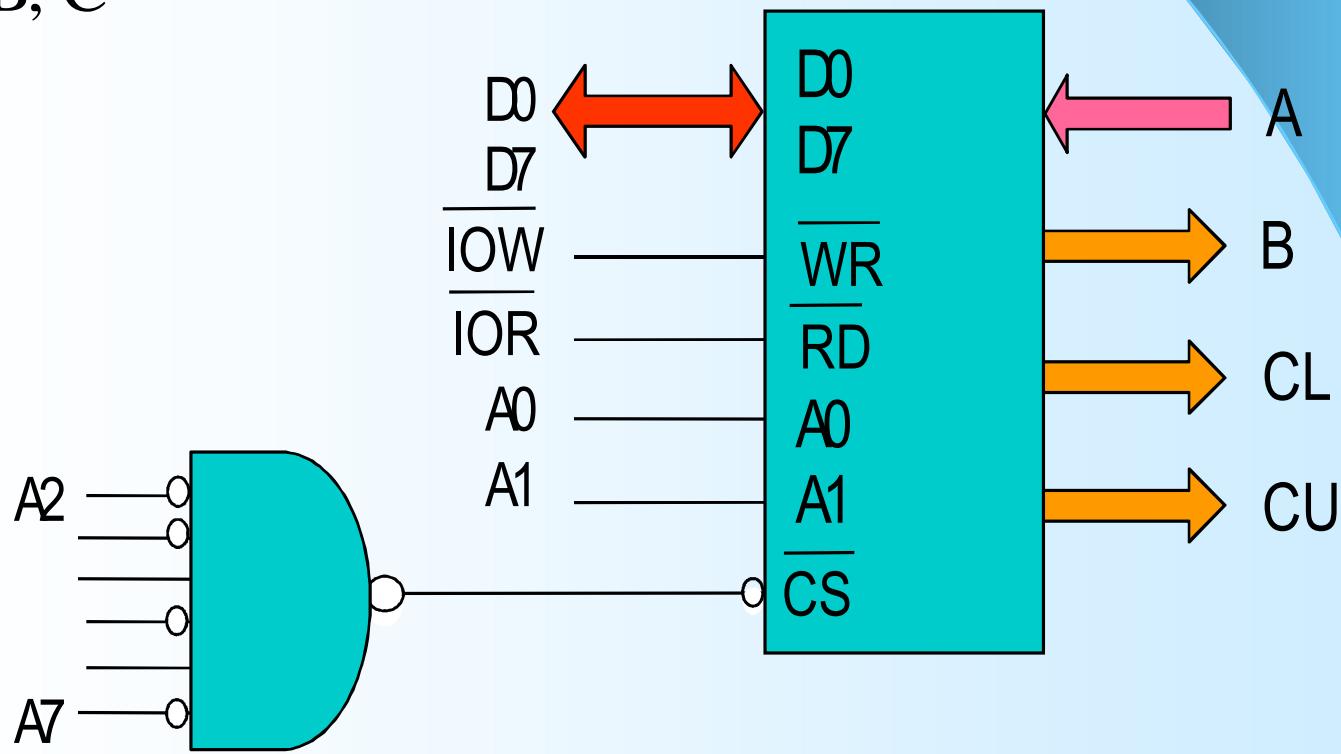
Các chế độ làm việc của 8255

- ✓ **Chế độ 0**: chế độ vào ra đơn giản. Ba cổng A, B, C đều có thể lập trình thành cổng vào hoặc ra (không thể điều khiển riêng cho từng bit).
- ✓ **Chế độ 1**: Cổng A và B có thể được dùng làm cổng vào hoặc ra với khả năng bắt tay, các tín hiệu bắt tay được truyền qua các bit của cổng C.
- ✓ **Chế độ 2**: Cổng A có thể được dùng làm cổng vào ra hai chiều với khả năng bắt tay nhờ các tín hiệu của cổng C. Cổng B có thể ở chế độ 0 hoặc 1.
- ✓ **Chế độ BSR (lập/xoá bit)**: Có thể lập trình cho từng bit trên cổng C.

Ví dụ về chế độ vào/ra đơn giản

Chip 8255 (hình vẽ) được định cấu hình như sau: Cổng A là cổng vào, cổng B là cổng ra và tất cả các bit của cổng C là ra.

- Tìm các địa chỉ cổng gán cho A, B, C và thanh ghi điều khiển.
- Hãy tìm byte (từ) điều khiển cho cấu hình trên.
- Hãy lập trình các cổng để dữ liệu vào cổng A được gửi tới cổng B, C



Giải:

a) Các địa chỉ cổng như sau:

<u>CS*</u>	<u>A1</u>	<u>A0</u>	<u>Địa chỉ</u>	<u>Cổng</u>
010100	0	0	50H	Cổng A
010100	0	1	51H	Cổng B
010100	1	0	52H	Cổng C
010100	1	1	53H	Thanh ghi điều khiển

b) Từ điều khiển là 90H hoặc 1001 0000

c) Chương trình thực hiện có thể như sau:

```
PORTA EQU 50H  
PORTB EQU 51H  
PORTC EQU 52H  
CNTLREG EQU 53H
```

...

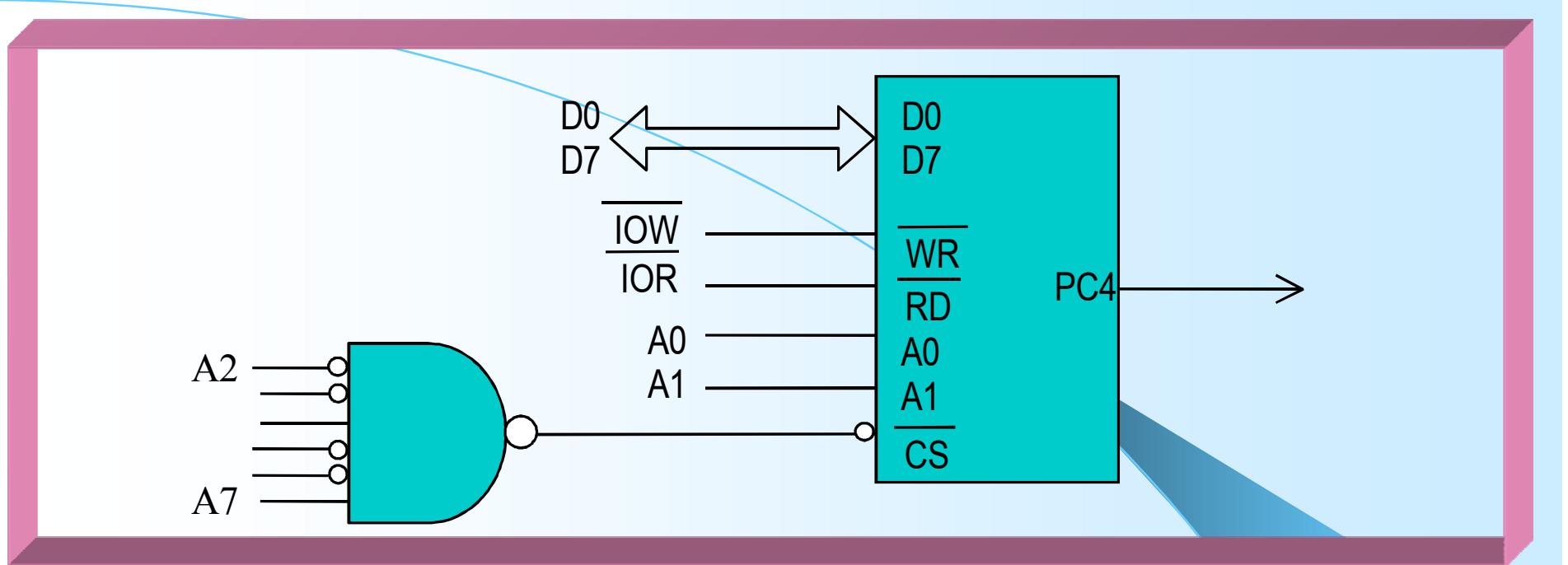
```
MOV AL, 90H ;byte điều khiển PA=vào, PB=ra, PC=ra  
OUT CNTLREG, AL ;gửi tới thanh ghi điều khiển  
IN AL, PORTA ;lấy dữ liệu từ PA  
OUT PORTB, AL ;gửi dữ liệu đến cả cổng PB  
OUT PORTC, AL ;và cổng PC
```

Ví dụ về chế độ BSR (lập / xoá bit)

- ✓ Từ điều khiển BSR

D7	D6	D5	D4	D3	D2	D1	D0
0	X	X	X	Bit Select		S/R	
Chế độ BSR	Không sử dụng Thờng Set=0		0	0	0	PC0	Set = 1
			0	0	1	PC1	Reset = 0
			0	1	0	PC2	
			0	1	1	PC3	
			1	0	0	PC4	
			1	0	1	PC5	
			1	1	0	PC6	
			1	1	1	PC7	

- ✓ Ví dụ: Lập trình để tạo xung 50ms với chu kỳ có hệ số lấp đầy là 50% trên chân PC4 của 8255 (hình vẽ).



Giải:

```

MOV AL, 00001001B ; nạp byte điều khiển (PC4=1)
OUT 93H, AL          ; đặt PC4 lên cao
CALL DELAY           ; thời gian cho phần cao xung
MOV AL, 00001000B ; nạp byte điều khiển (PC4=0)
OUT 93, AL           ; đặt PC4 về thấp
CALL DELAY           ; thời gian cho phần thấp xung

```

Tạo trẽ

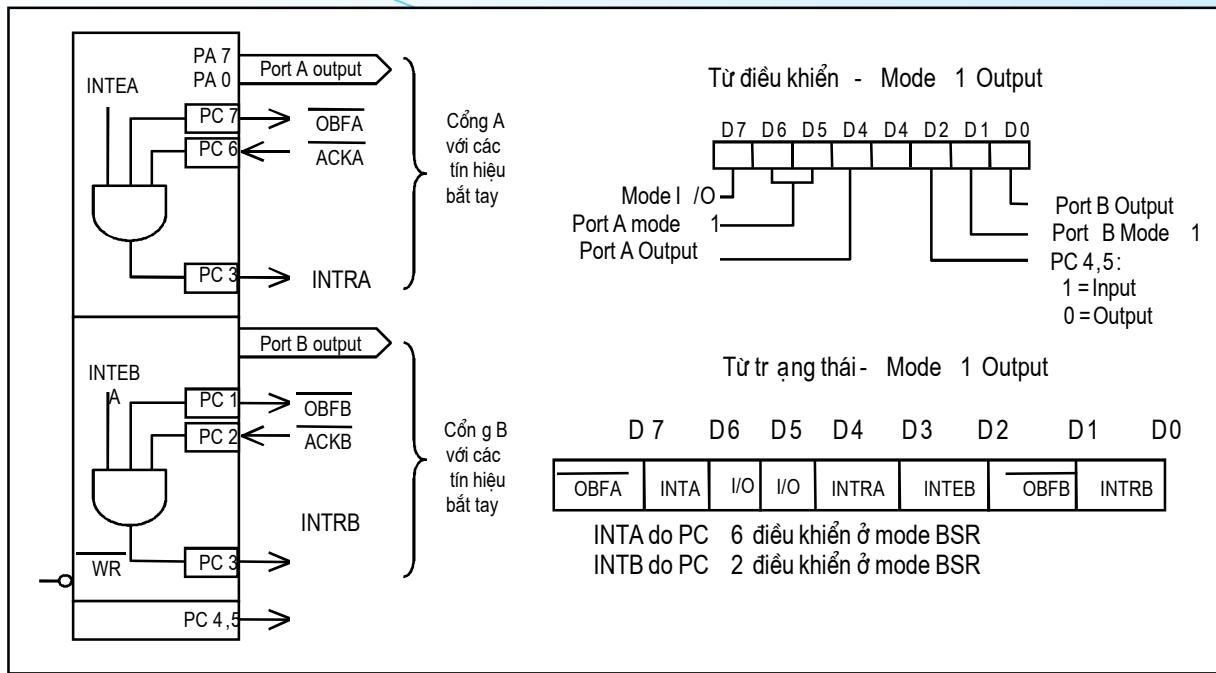
- Chương trình con tạo trẽ đơn giản:

```
        mov    CX, N  
AGAIN: loop   AGAIN
```

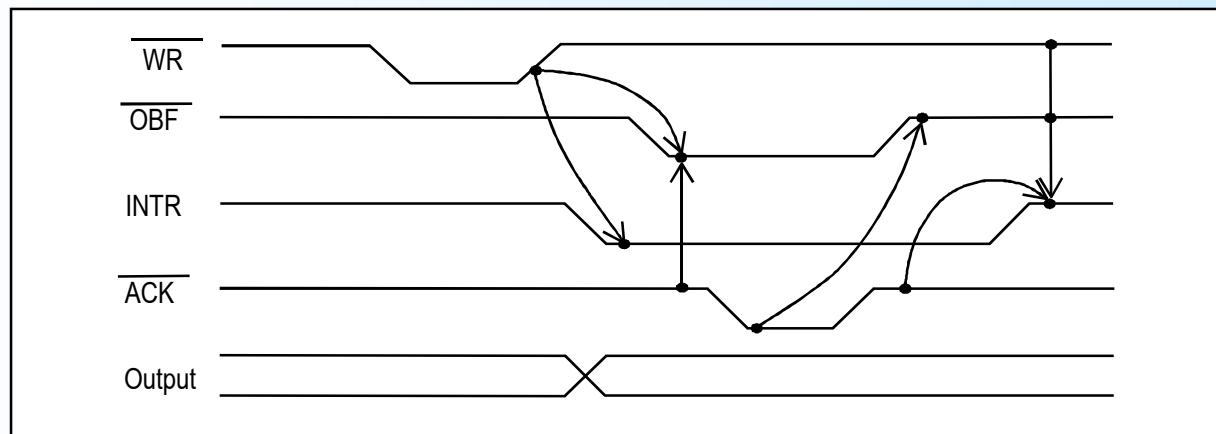
- Chương trình con tạo trẽ dài hơn:

```
        sub    CX, CX  
G7:    loop   G7  
        dec    BL  
        jnz    G7
```

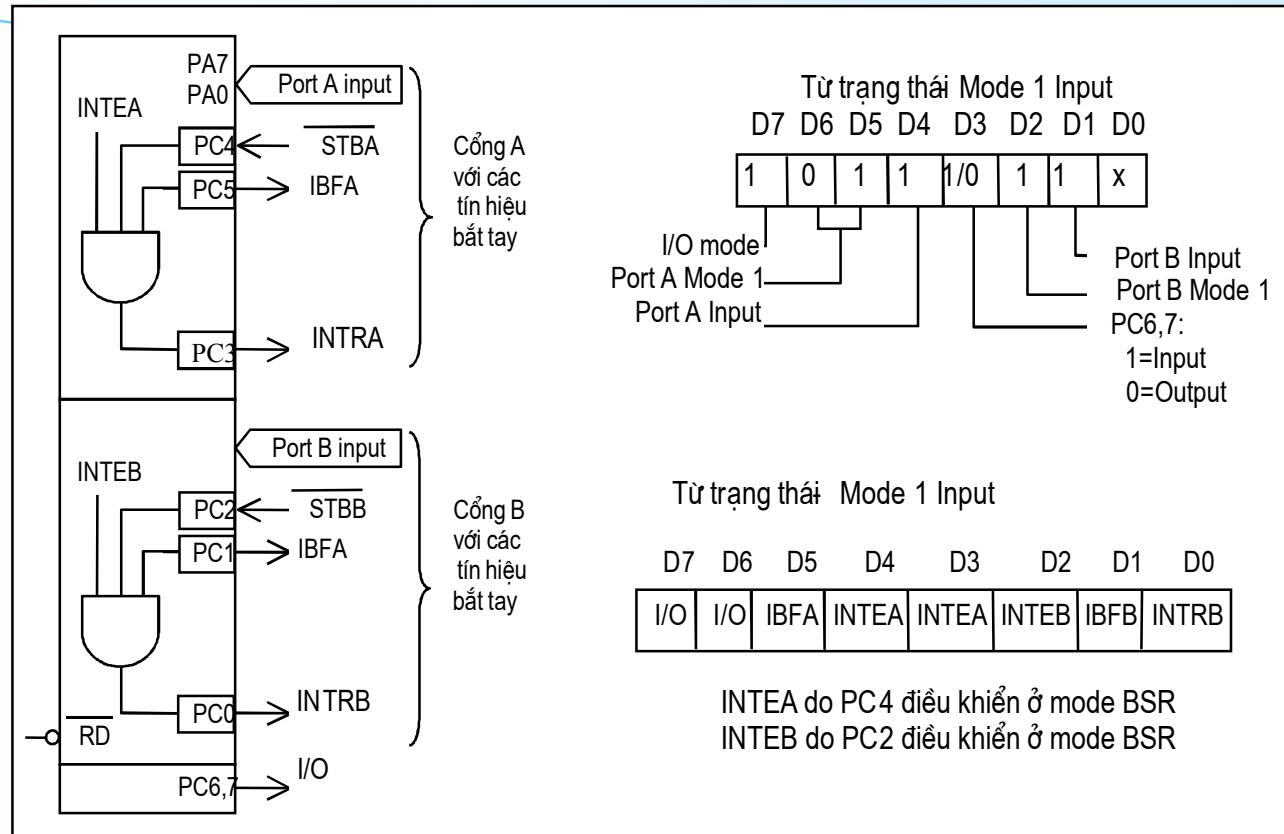
Chế độ 1: I/O với khả năng bắt tay



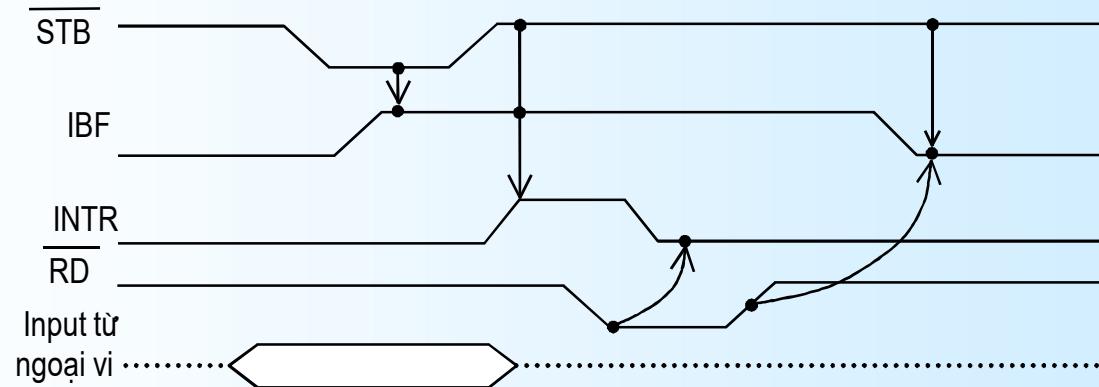
Sơ đồ đầu ra chế độ 1 của 8255 A



Biểu đồ thời gian lối ra chế độ 1

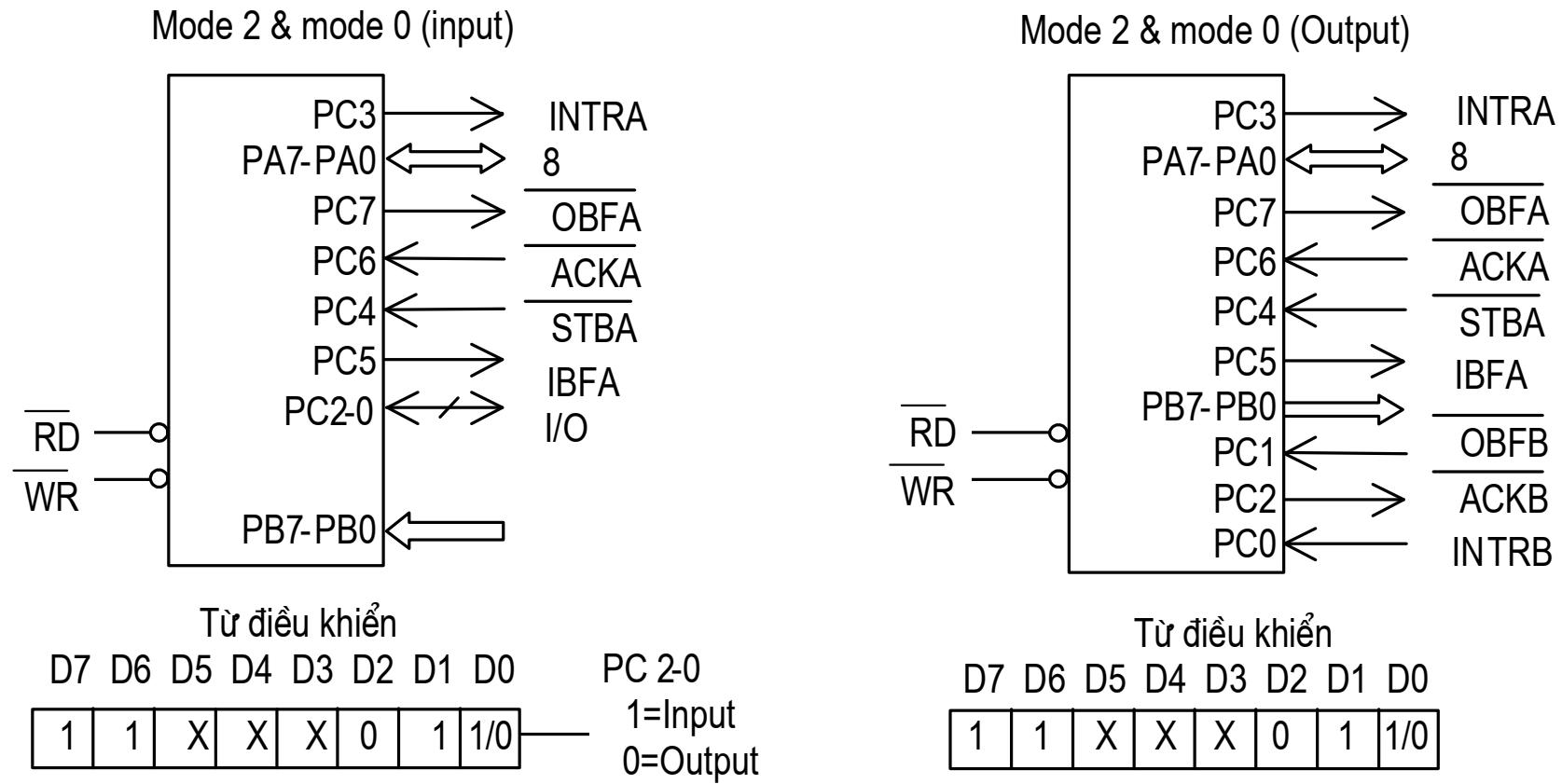


Sơ đồ đầu vào chế độ 1 của 8255A



Biểu đồ thời gian lối vào chế độ 1

Chế độ 2: I/O hai chiều với tín hiệu bắt tay



Sử dụng 8255 trong máy PC

Cổng A (vào)
Địa chỉ: 60H

PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7

Mã quét từ bàn
phím

Cổng B (ra)
Địa chỉ: 61H

PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7

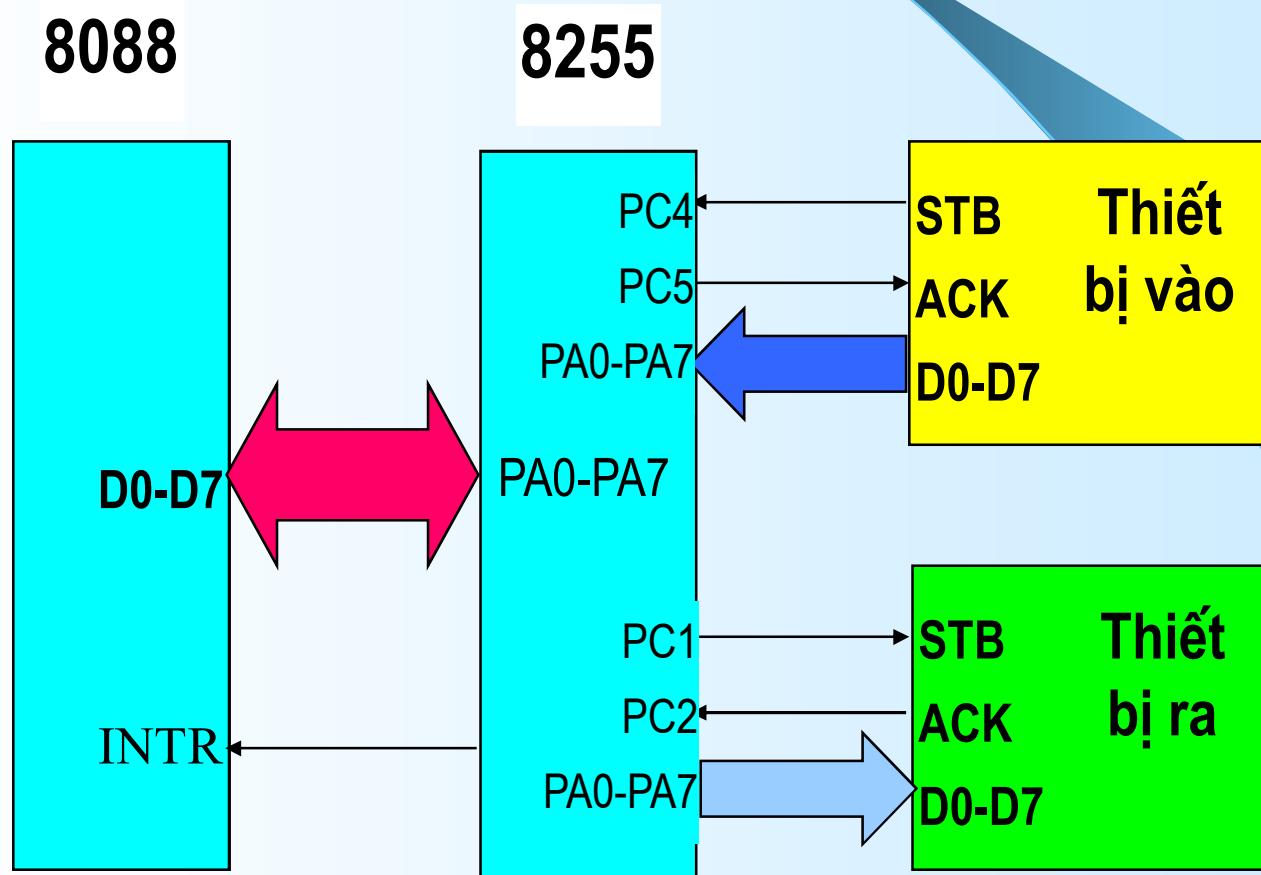
Mở loa Timer2
Dữ liệu cho loa
Không sử dụng
Không sử dụng
Cho phép kiểm tra chấn lě DRAM
Cho phép kiểm tra kênh I/O
Không sử dụng
Không sử dụng

Cổng C (vào)
Địa chỉ: 62H

PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7

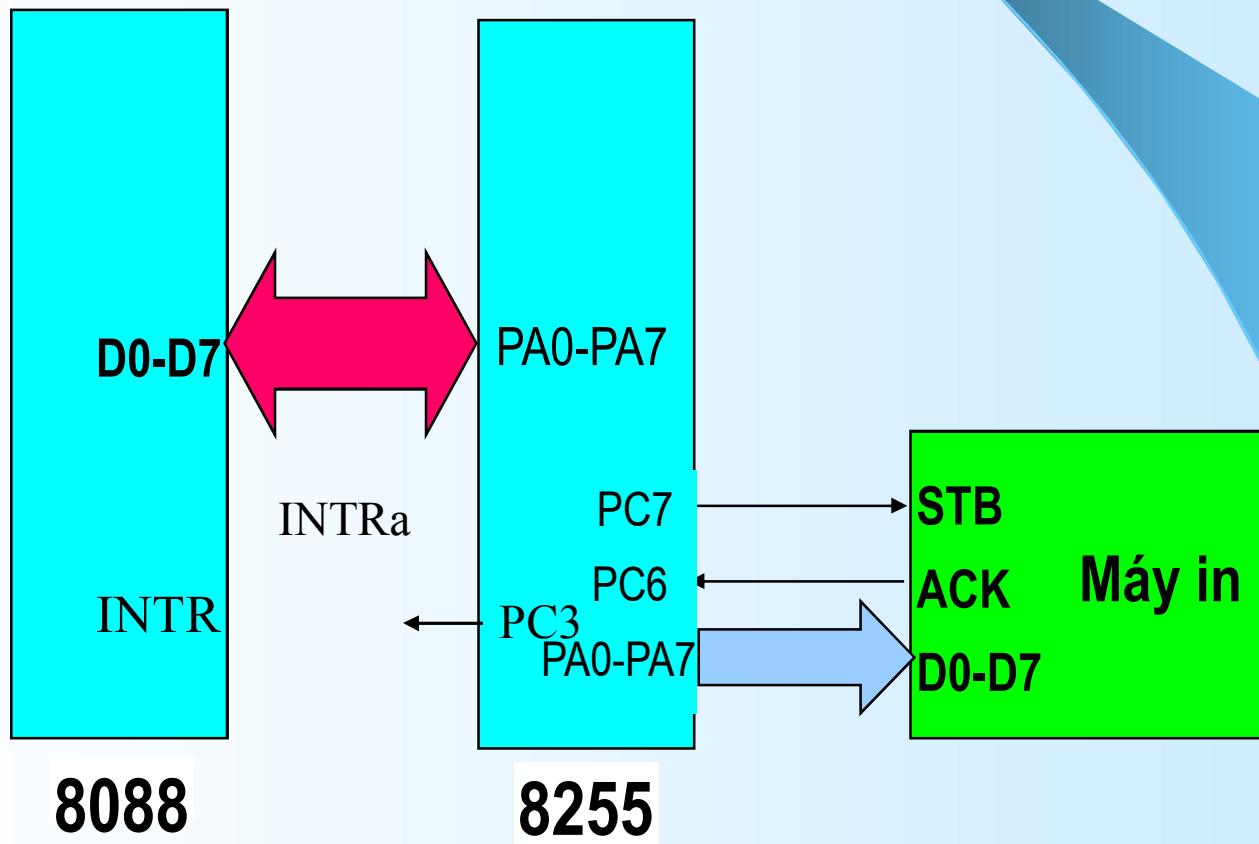
Không sử dụng
1: có đồng xử lý
0: có dĩa
Không sử dụng
Không sử dụng
Output kênh Timer2
Kiểm tra kênh I/O
Kiểm tra chấn lě DRAM

Dùng mạch 8255A để phôi ghép vào/ra ở chế độ thăm dò



Ví dụ vào ra bằng thăm dò

- Sơ đồ ghép nối máy in:



Chương trình in thông báo: “Chao ban. Ban co khoe khong?”

Đoạn dữ liệu:

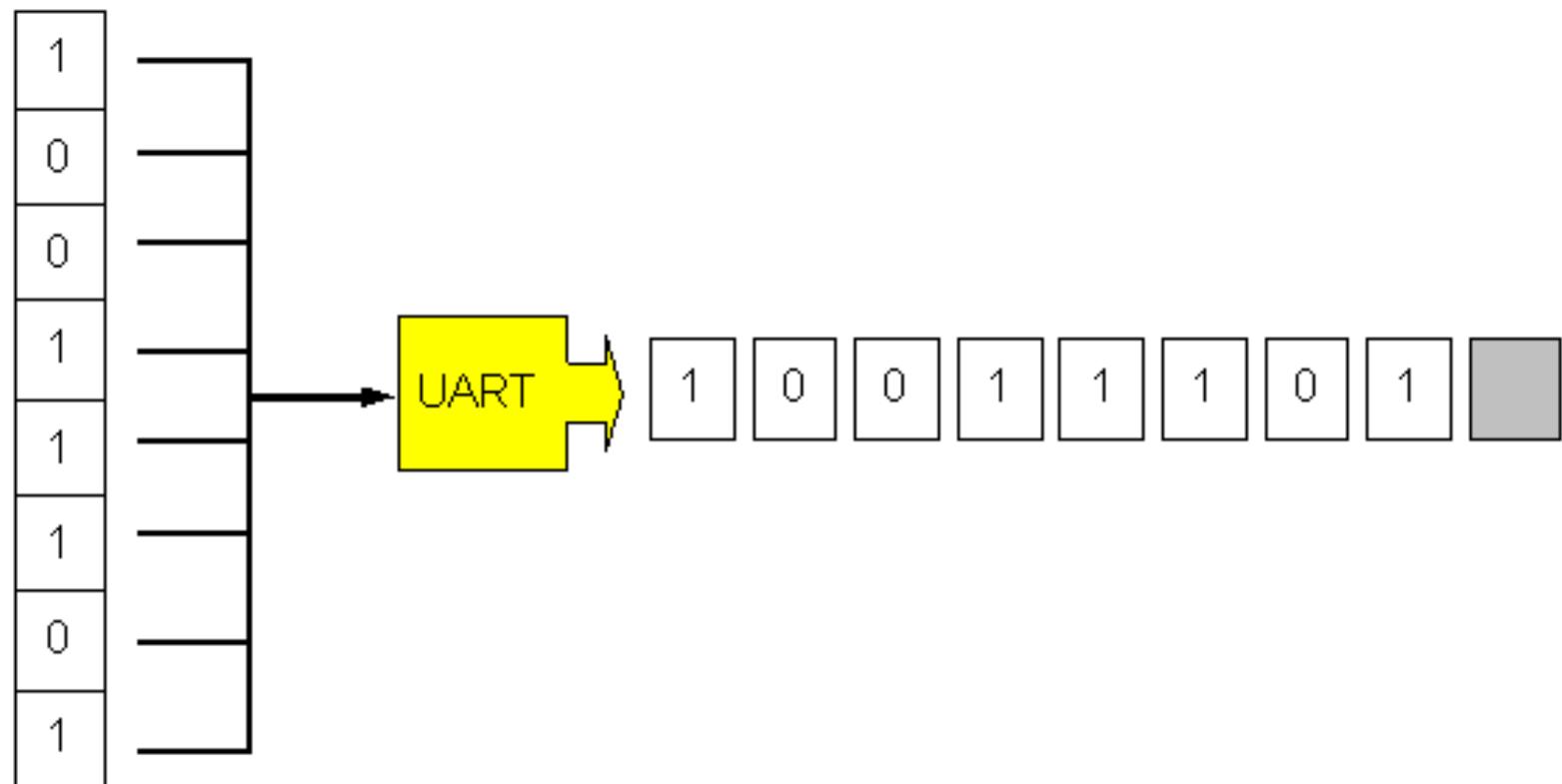
MY_DATA	DB	“Chao ban. Ban co khoe khong?”, "\$"
PA	EQU	300H ;dia chi cong PA
PB	EQU	301H ;dia chi cong PB
PC	EQU	302H ;dia chi cong PC
CWP	EQU	303H ;dia chi thanh ghi dieu khien
LF	EQU	0AH ;ky tu xuong dong
CR	EQU	0DH ;ky tu ve dau dong

Đoạn chương trình:

	MOV	AL,1010000B	;tu dieu khien PA ra, che do 1
	MOV	DX,CWP	;dia chi tu dieu khien
	OUT	DX,AL	;ghi tu dieu khien
	MOV	AL,00001101	;lap PC6=1 cho INTEa
	OUT	DX,AL	;dung che do BSR
	MOV	SI,OFFSET MY_DATA	;SI=dia chi chuoi du lieu
AGAIN:	MOV	AH,[SI]	;lay tung ky tu
	CMP	AH,"\$"	;ket thuc chuoi chua?
	JZ	OVER	;neu roi, thoat ra
	MOV	DX,PC	;dia chi cong PC
BACK:	IN	AL,DX	;doc byte trang thai cong PC
	AND	AL,08	;INTRa o muc cao khong?
	JZ	BACK	;neu khong tiep tuc hoi vong
	MOV	DX,PA	;neu co, chuyen dia chi cong
	MOV	AL,AH	;PA vao DX (DX=300h)
	OUT	DX,AL	;gui ky tu cua chuoi toi may in
	INC	SI	;tang con tro du lieu
	JMP	AGAIN	;tiep tuc thu hien
OVER:		;tro ve DOS

VÀO/RA NỐI TIẾP

- Vào/ra nối tiếp là gì? : Phương pháp truyền từng bit dữ liệu.



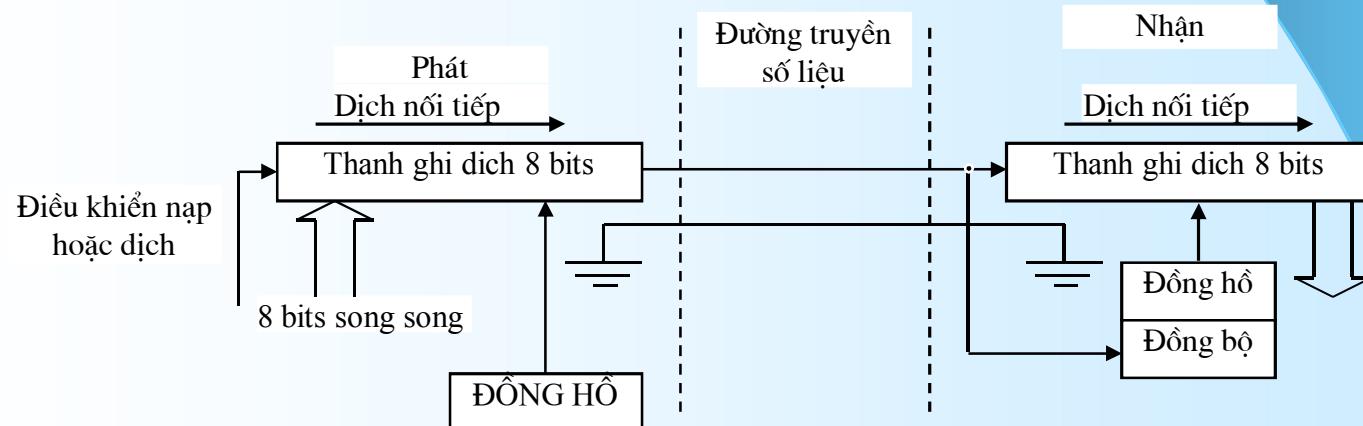
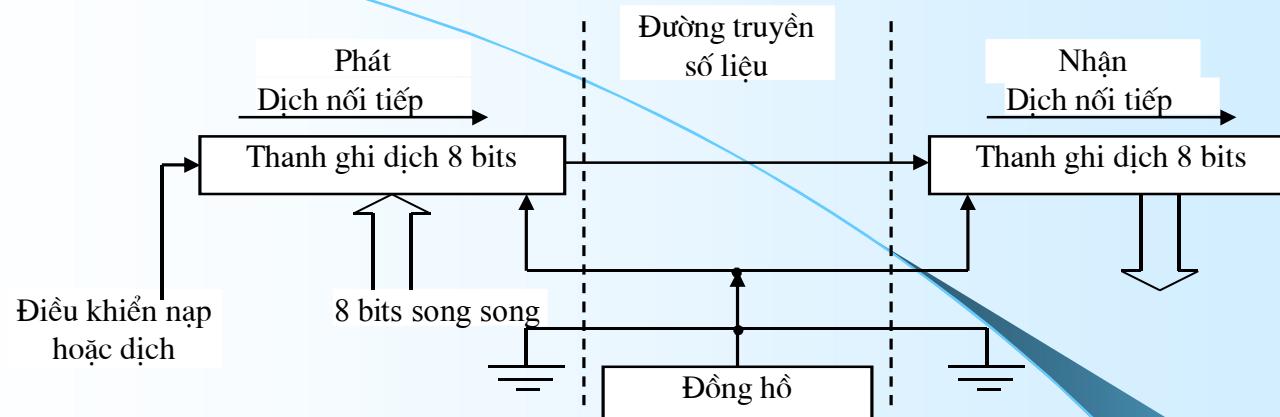
The serial transfer is limited to a speed of 115,200 bits per second. The cable can be up to 200 meter long. The serial ports can be used to connect:

- The mouse and digitizers
- Modems
- ISDN adapters
- Printers with serial interface
- Digital cameras
-

These units are connected to the serial ports using either DB9 or DB25 plugs.

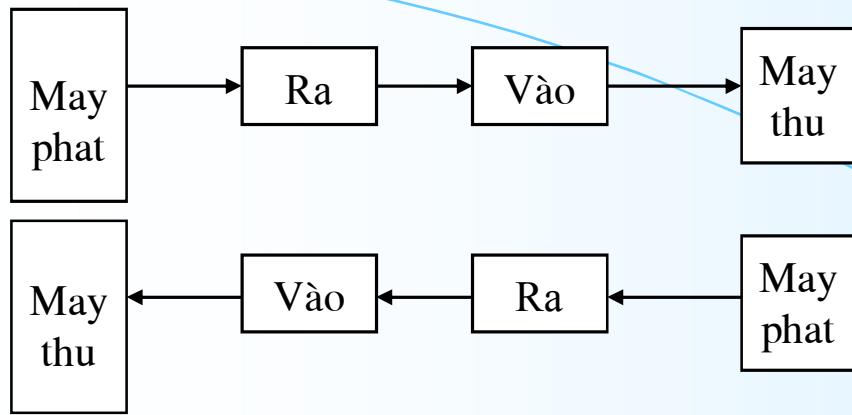
- Cách thực hiện:
 - Dùng lệnh phần mềm. Dùng mạch USART.
- Các giao thức truyền nối tiếp:
 - Truyền đồng bộ: mỗi lần chuyển một khối dữ liệu
 - In the synchronous transmission you need two separate cables. With every clock pulse (i.e. the positive going edge of the clock) one data bit is transferred.

- Truyền dữ liệu: mỗi lần chuyển một byte
 - Truyền dữ liệu và định khung dữ liệu: dữ liệu được “đóng khung” giữa bit khởi động và bit dừng. Ví dụ ký tự “A”
- Truyền dữ liệu, tín hiệu đồng bộ và số liệu được truyền trên cùng một cable.
- The clock has to be reconstructed from the mixed signal in the receiver: After a "1" start bit come 8 data bits and then a "0" stop bit (or two "0" stop bits) and so on.

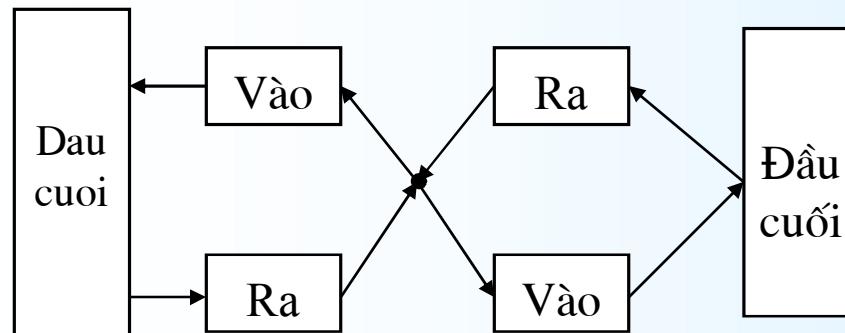


Truyền song công và bán song công

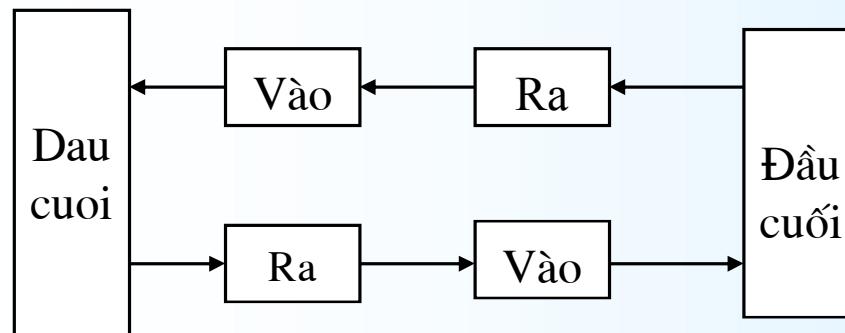
- *Truyền đơn công*: dữ liệu chỉ đi 1 chiều từ máy phát đến máy thu.
Ví dụ: truyền dữ liệu từ máy tính sang máy in
- *Truyền song công*: dữ liệu có thể vừa phát vừa thu.
 - *Truyền song công hoàn toàn*: tại mỗi thời điểm dữ liệu có thể truyền 2 chiều.
Ví dụ: truyền dữ liệu giữa máy tính với Modem.
 - *Truyền bán song công*: tại mỗi thời điểm dữ liệu chỉ truyền một chiều.



**Truyền
đơn
công**



**Truyền
bán
song
công**



**Truyền
song
công**

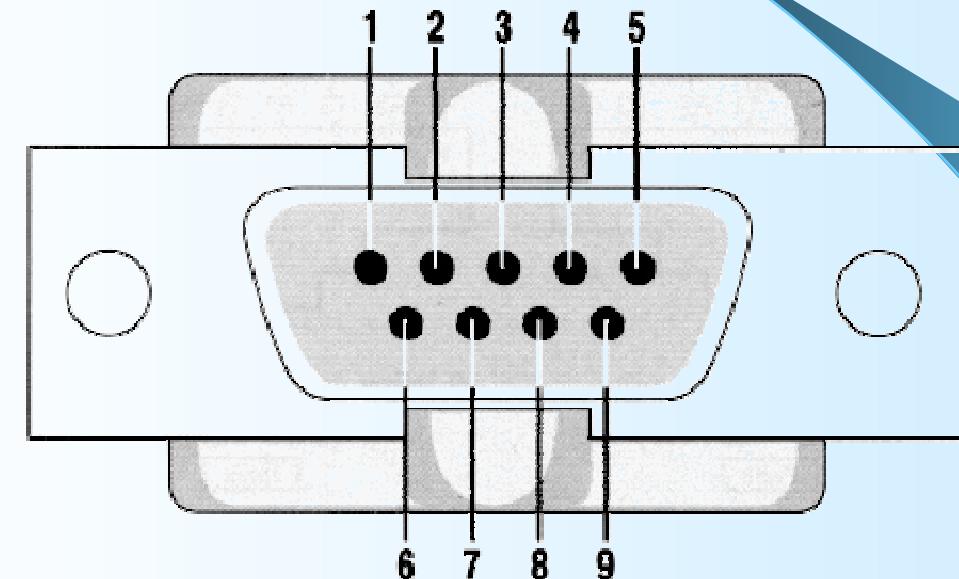
DTE và DCE

- **DTE (*data terminal equipment*)**-**Thiết bị đầu cuối dữ liệu:** chỉ máy tính hay thiết bị đầu cuối để thu/phát dữ liệu.
- **DCE (*data communication equipment*)**-**Thiết bị truyền dữ liệu:** để chỉ các thiết bị truyền thông, ví dụ như modem - chịu trách nhiệm truyền dữ liệu.

TRUYỀN THÔNG NỐI TIẾP KHÔNG ĐỒNG BỘ

- Sử dụng khi các máy thu, máy phát có tốc độ khác nhau.
- Nhược điểm: tốc độ chậm hơn truyền đồng bộ.
- Ưu điểm: truyền được giữa các máy có tốc độ khác nhau.
- Dùng chuẩn RS-232.
- Truyền từng Byte.
- Dữ liệu trước khi truyền phải định khung, tức đưa dữ liệu vào một khung bắt đầu bằng bit Start kết thúc bằng bit Stop.

Chuẩn truyền thông nối tiếp RS-232 (Giắc đực)



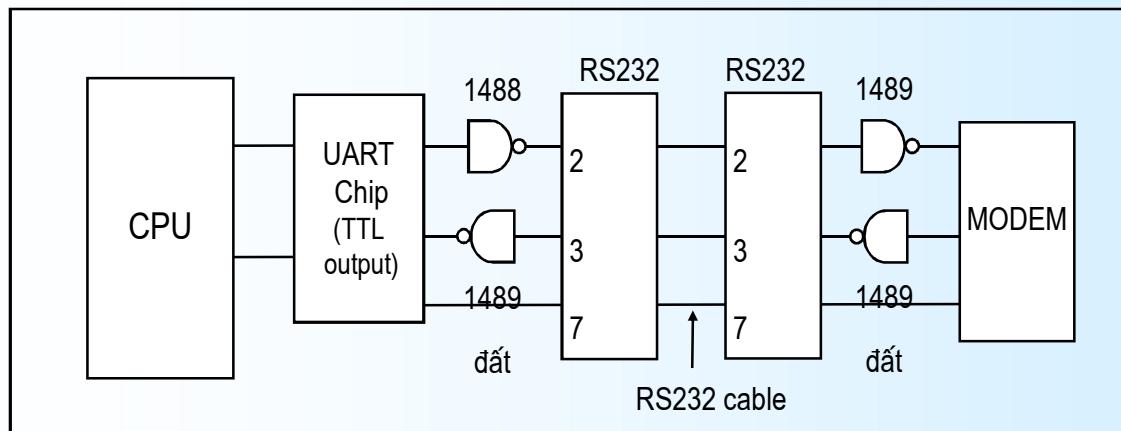
Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Ý nghĩa các tín hiệu

- TxD (*Transmitted Data*) - dữ liệu phát.
- RxD (*Received Data*) - dữ liệu thu.
- DTR (*data terminal ready*) - thiết bị đầu cuối sẵn sàng.
- DSR (*data set ready*) - dữ liệu sẵn sàng.
- RTS (*request to send*) - yêu cầu gửi dữ liệu.
- CTS (*clear to send*) - tín hiệu thông.
- DCD (*data carrier detect*) - dò sóng mang dữ liệu.
- RI (*ring indicator*) - chỉ thị chuông.

Chuyển đổi tín hiệu TTL sang chuẩn RS-232

- Mức TTL của máy tính là 0V("0") và +5V("1")
- Trong RS232, mức 1 tương ứng từ -3V đến -25V, còn mức 0 tương ứng từ +3V đến +25V, từ -3V đến +3V không xác định.
- Dùng 1488 và 1489 để chuyển đổi mức tín hiệu:

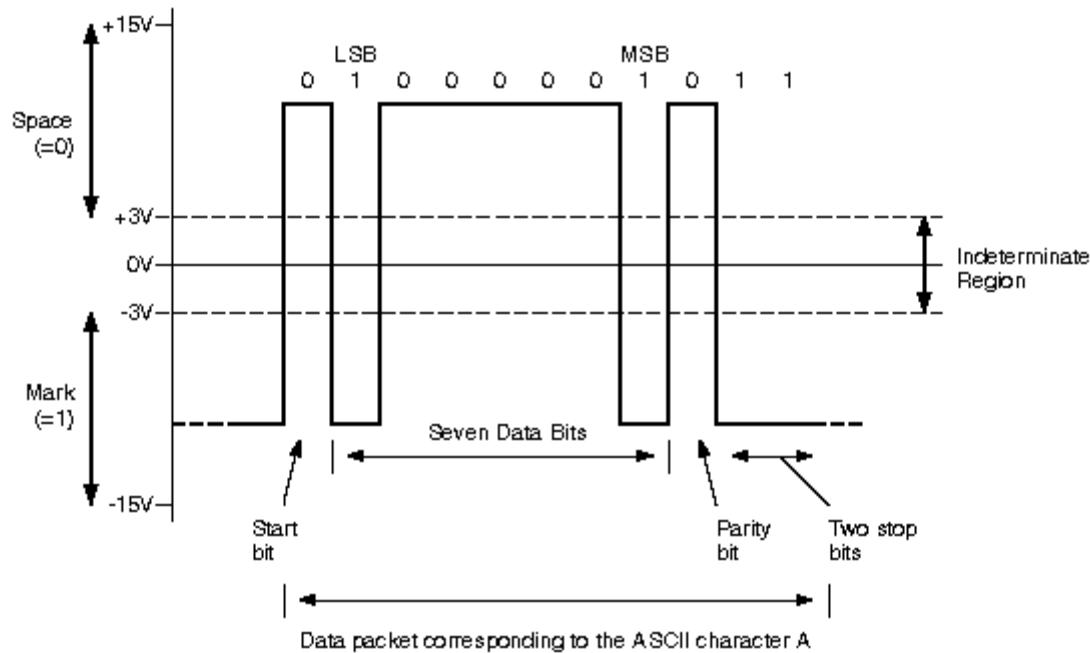


Hình 9-4. Sử dụng chip MC1488 và MC1489 nối UART với RS232

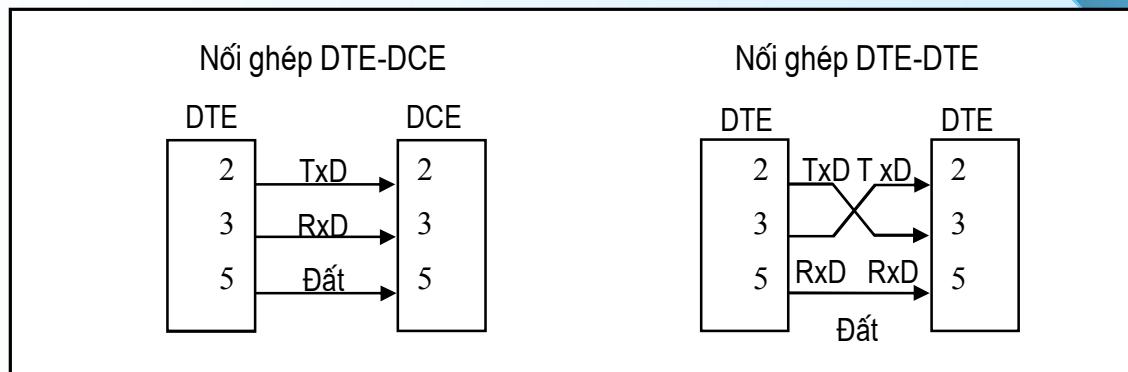
Tốc độ truyền dữ liệu.

- Tốc độ truyền dữ liệu thông tin nối tiếp được tính bằng bit/giây (*bps*).
- *Baud* là đơn vị đo cho modem và được định nghĩa là số lần thay đổi tín hiệu trong một giây.
- Đối với đường truyền thì tốc độ *baud* và *bps* là một.

Định khung dữ liệu

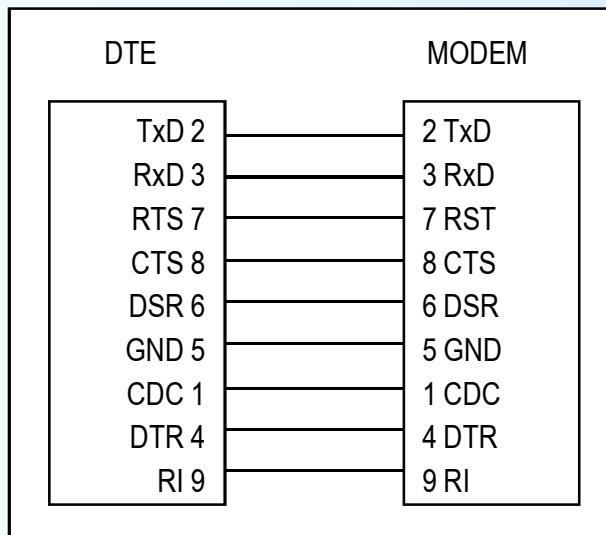


Nối ghép DTE-DCE, DTE-DTE (cable tối thiểu)



Hình 9-7. Nối ghép DTE-DCE và DTE-DTE

Nối ghép DTE-DCE, DTE-DTE (cable đầy đủ)



**Hình 9-8. Kết nối bắt tay
DTE - DCE**

TRUYỀN THÔNG NỐI TIẾP ĐỒNG BỘ

- Trong truyền dữ liệu nối tiếp đồng bộ, các bit khởi động và bít dừng được thay bằng các byte có mã đặc biệt,
- Thay cho việc truyền từng byte như trước đây thì phương pháp này thực hiện truyền hàng trăm byte một.
- Từng khối gồm nhiều byte được định khung giữa các mã điều khiển và sau đó mới được truyền đi.
- Có 2 giao thức đồng bộ chuẩn được sử dụng rộng rãi đó là: BISYNC và SDLC

Giao thức đồng bộ BYSYNC

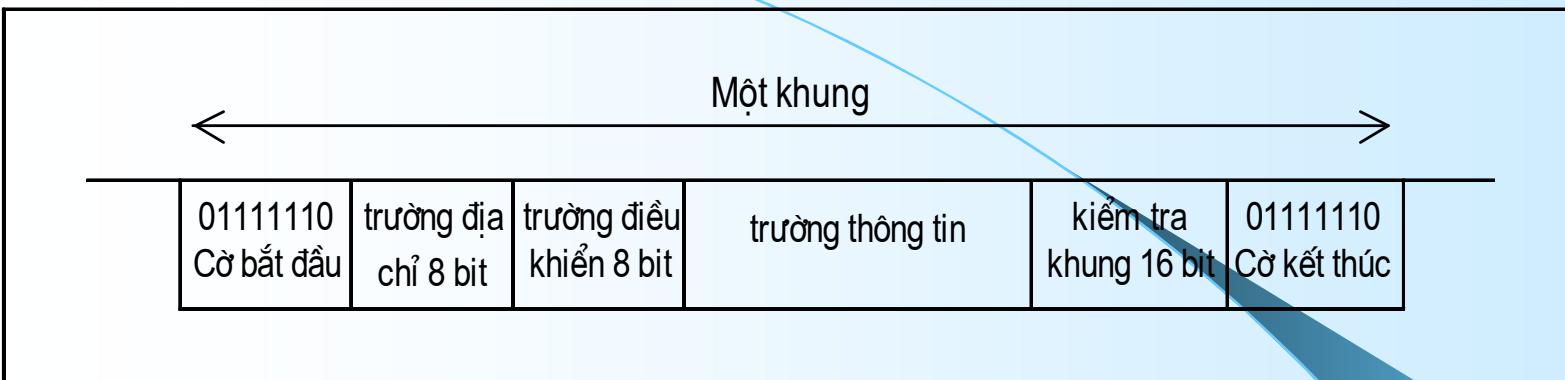


Hình 9-20. Định khung dữ liệu BISYNC

Bảng 9-9. Một số mã sử dụng giao thức BISYNC

Ký tự	Ý nghĩa	ASCII (H)
SOH	Start of header- khởi động đoạn đầu	01
STX	Start of text - bắt đầu văn bản	02
ETX	End of text - kết thúc văn bản	03
EOT	End of transmission - kết thúc truyền	04
ENQ	Inquiry - hỏi tin	05
ETB	End-of-transmission block kết thúc khối truyền	0F
DLE	Data link escape - thoát liên kết dữ liệu	10
SYNC	Sync - đồng bộ	16
ETB	End-of-transmission block khối kết thúc truyền	17
PAD	end-of-frame block khối kết thúc khung	FF

Giao thức đồng bộ SDLC

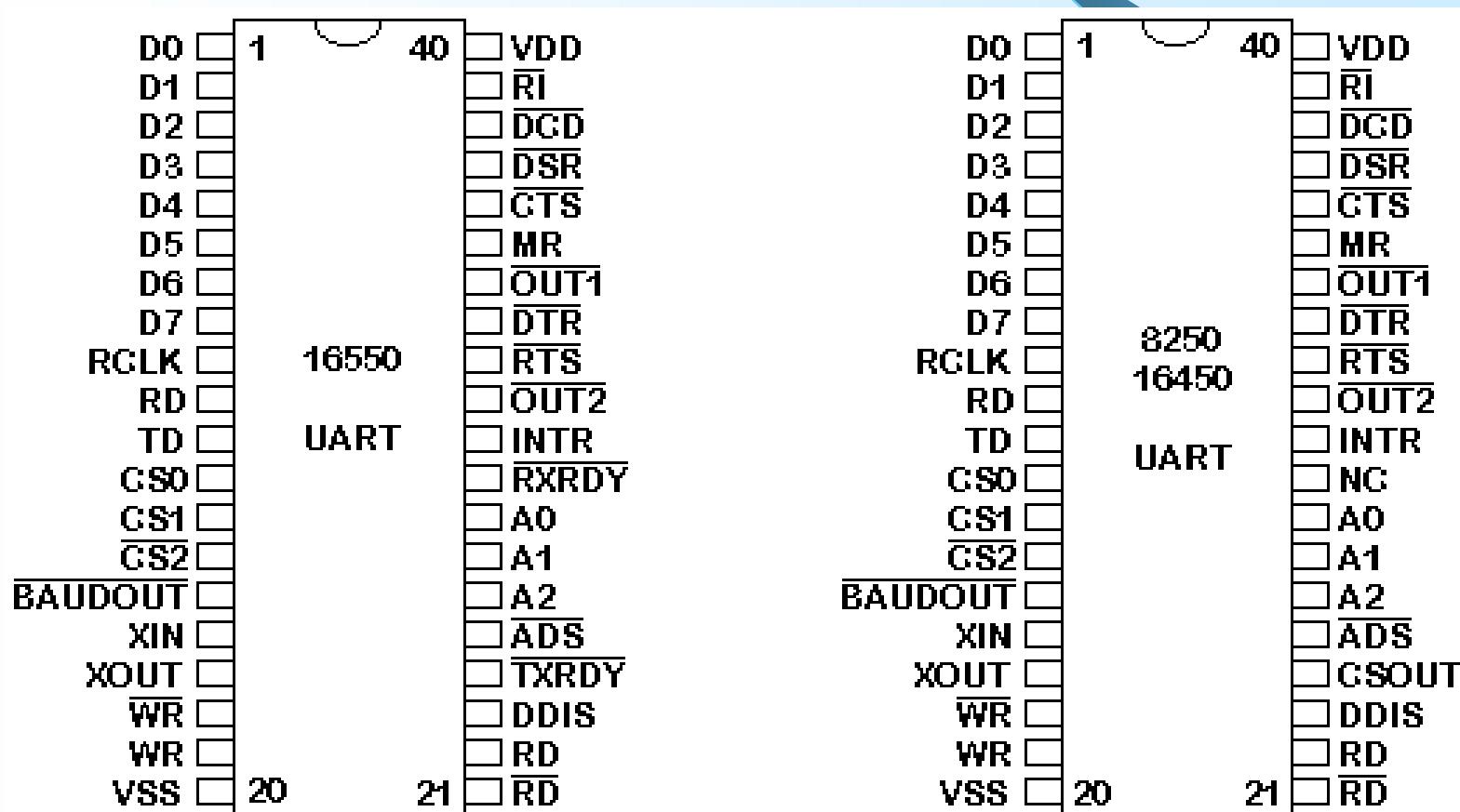


Hình 9-21. Định khung dữ liệu SDLC

Do IBM đề xuất và được sử dụng rộng rãi trong mạng máy tính như token ring của IBM. Giao thức chuyển byte 01111110 thay cho byte SYNC. Trong tài liệu của IBM, byte này được xem là byte cờ. Tiếp theo byte cờ là byte chứa địa chỉ đích trên mạng, nơi dữ liệu được gửi trên mạng máy tính. Tiếp theo là byte điều khiển, trong đó có thông tin về trình tự của dữ liệu. Tiếp theo là dữ liệu. Dữ liệu được sử dụng trong SDLC là bit, và có khoảng hàng ngàn bit. Đây là điểm khác với BISYNC vì ở BISYNC là hàng ngàn byte. Trường dữ liệu có thể lên tới 200.000 bit/khung. Tiếp theo trường dữ liệu là trường 16 bit có chức năng kiểm tra lỗi, và cuối cùng là byte cờ để chỉ thị kết thúc khung dữ liệu.

Vi mạch lập trình vào/ra nối tiếp 8250//16450

Vi mạch 8250 - Serial Communication Interfaces (SCI) là vi mạch 40 chân dạng DIP có chức năng gửi và nhận số liệu nối tiếp trên các máy tính họ 80x86.



CS0, CS1, $\overline{CS2}$ (chon chip)

Được dùng để kích hoạt 8250. Trong IBM PC CS0 và CS1 được thường xuyên nối lên mức cao.

Sin và Sout

Là chân số liệu nối tiếp, chính là chân RxD và TxD của RS 232 sau khi chuyển từ mức TTL sang mức điện áp của RS 232.

Xin và Xout

Xin và Xout được nối đến bộ dao động thạch anh. Máy tính IBM PC có tần số đặt vào Xin là 1,8432 MHz

MR

Là chân Reset của 8250, có mức tích cực cao dùng để xóa tất cả các thanh ghi trừ thanh ghi đệm đầu thu, lưu tín hiệu phát và chốt hệ số chia

RCLK

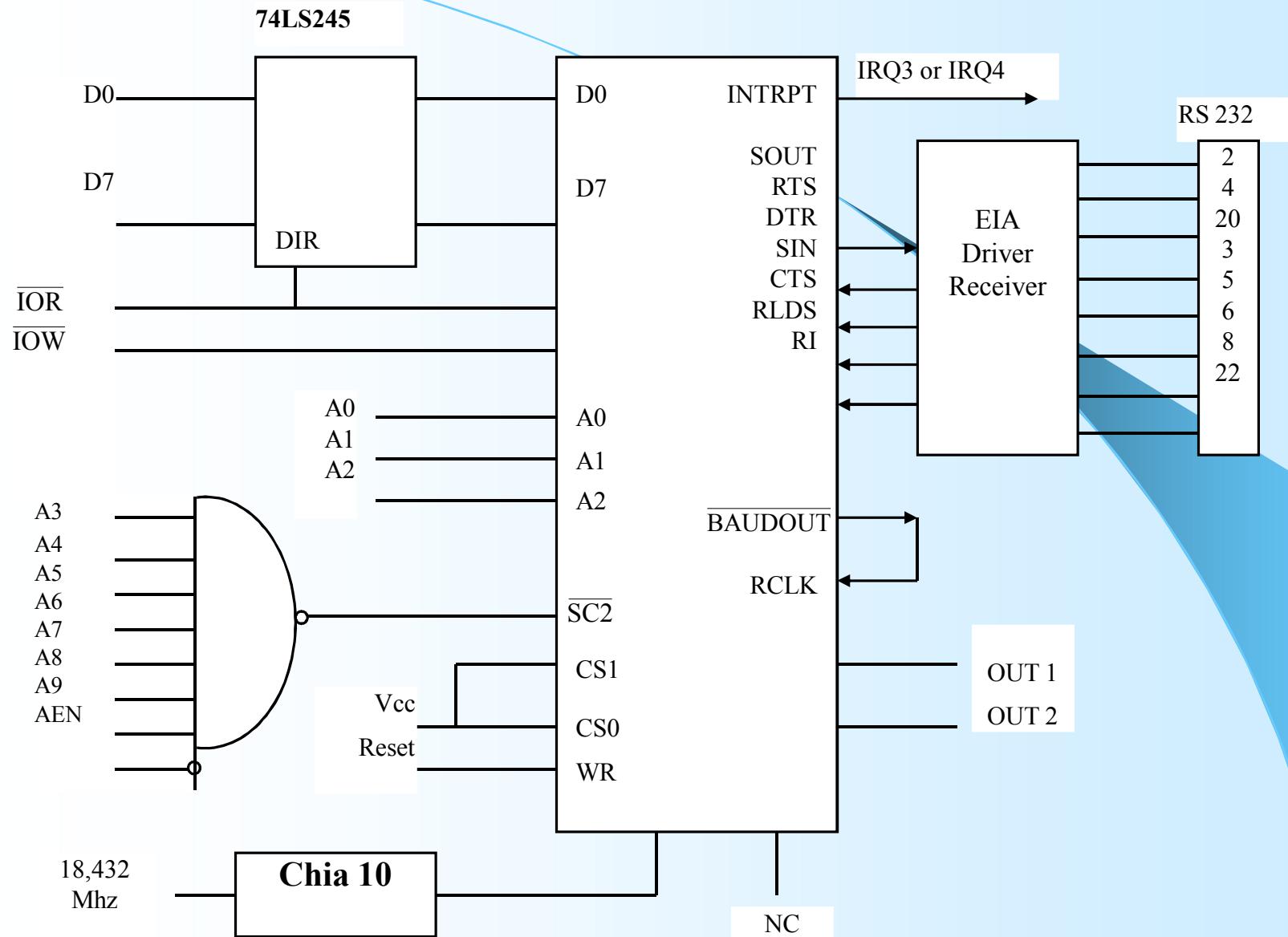
Đồng hồ máy thu là đầu vào có tần số bằng 16 lần đồng hồ tốc độ truyền của máy thu.

BAUDOUT

Là đầu ra có tần số bằng 16 lần tốc độ truyền của máy phát. Chân này được nối tới chân RCLK để cho bộ thu và phát làm việc ở cùng một tần số. Vì vậy tốc độ truyền và nhận luôn bằng 16 lần giá trị thanh ghi lưu số chia

D0 ÷ D7

BUS số liệu hai chiều nối với Data BUS của máy tính.



Ghép nối 8250 với máy tính IBM PC

A2 A1 A0

Để truy cập các thanh ghi của vi mạch điều khiển vào/ra nối tiếp 8250, máy tính sử dụng kết hợp với các chân địa chỉ A2, A1, A0 và bit D7 DLAB (*Divisor Latch Access Bit*) của thanh ghi điều khiển đường truyền (*Line Control Register – LCR* - Có địa chỉ 3FB H). Thường các tín hiệu CS1, CS0 được thường xuyên nối lên mức cao, còn được nối với đầu ra của mạch giải mã 74138. Các thanh ghi của 8250 được định nghĩa như sau:

Các thanh ghi của 8250

\overline{CS}	CS1	CS0	DLAB	A2	A1	A0	Hex	Chức năng
0	1	1	0	0	0	0	3F8	Thanh ghi đệm số liệu thu và lưu số liệu phát
0	1	1	0	0	0	1	3F9	Thanh ghi cho phép ngắt
0	1	1	x	0	1	0	3FA	Thanh ghi nhận biết ngắt (chỉ đọc)
0	1	1	x	0	1	1	3FB	Thanh ghi điều khiển đường truyền
0	1	1	x	1	0	0	3FC	Thanh ghi điều khiển Modem
0	1	1	x	1	0	1	3FD	Thanh ghi trạng thái
0	1	1	x	1	1	0	3FE	Thanh ghi trạng thái Modem
0	1	1	x	1	1	1	3FF	Thanh ghi xoá

Các thanh ghi của 8250

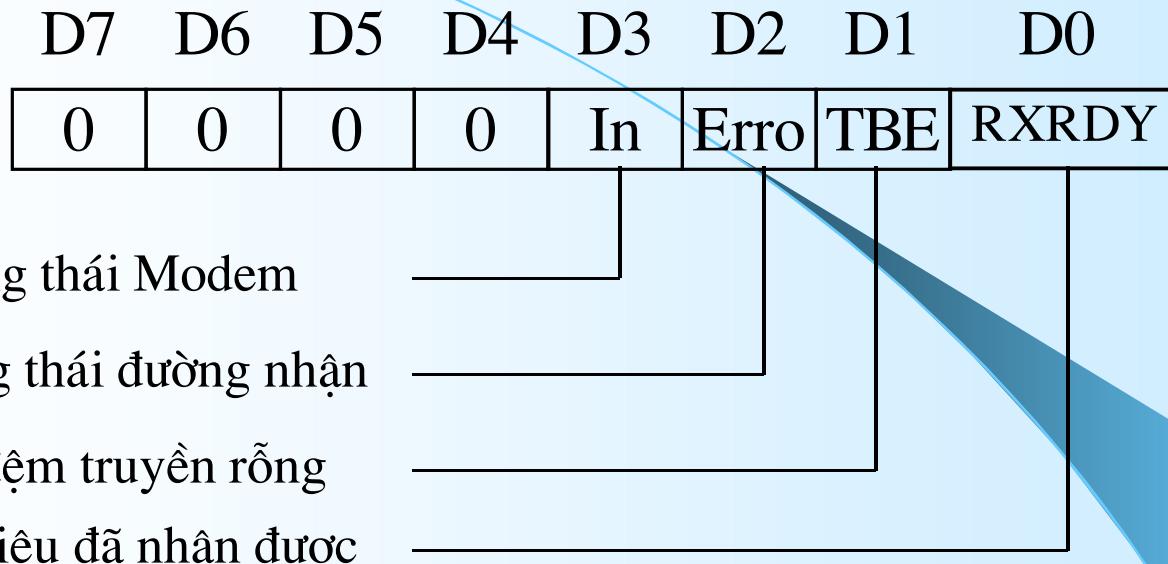
Thanh ghi lưu giá trị phát (3F8H và DLAB = 0)

Khi cần truyền một byte số liệu, BVXL ghi byte đó vào thanh ghi này. Sau đó 8250 định khung số liệu bằng cách bổ xung thêm bit Start, bit Stop để truyền nối tiếp qua chân S_{out} .

Thanh ghi đệm đầu thu (3F8h và DLAB = 0)

Sau khi 8250 nhận số liệu từ chân S_{in} nó sẽ tự bỏ các bit định khung (Start, Stop, Parity) để lọc lấy các bit số liệu. Các bit số liệu sau đó được chứa trong thanh ghi này.

Thanh ghi cho phép ngắt (3F9h, DLAB = 0)



Khi D0 =1 thông báo một byte số liệu đã được nhận qua chân S_{in}

Khi D1 = 1 thông báo bộ đếm phát rỗng, 8250 chờ nhận một byte mới.

Khi D2 = 1 thông báo có lỗi trong quá trình truyền số liệu. Lỗi này có thể là do định khung, lỗi chẵn lẻ, lỗi tràn hay lỗi bit stop.

Khi D3 =1 Thông báo trạng thái của modem đã được thay đổi

Thanh ghi nhận biết ngắt (3FAh)

Thanh ghi này cho phép xác định nguồn tín hiệu tạo INTR từ thanh ghi cho phép ngắt (3f9h)

D0 thông báo có ngắt đang đợi

D1 và D2 cho biết ngắt có mức ưu tiên cao nhất

D7-D3 luôn bằng 0

Để kiểm tra xem có ngắt nào đang đợi hay không máy tính sử dụng phương pháp hỏi vòng: Trước hết nó kiểm tra D0 xem có ngắt nào đang đợi hay không rồi tiếp tục kiểm tra D1, D2 để xác định ngắt.

D7 D6 D5 D4 D3 D2 D1 D0

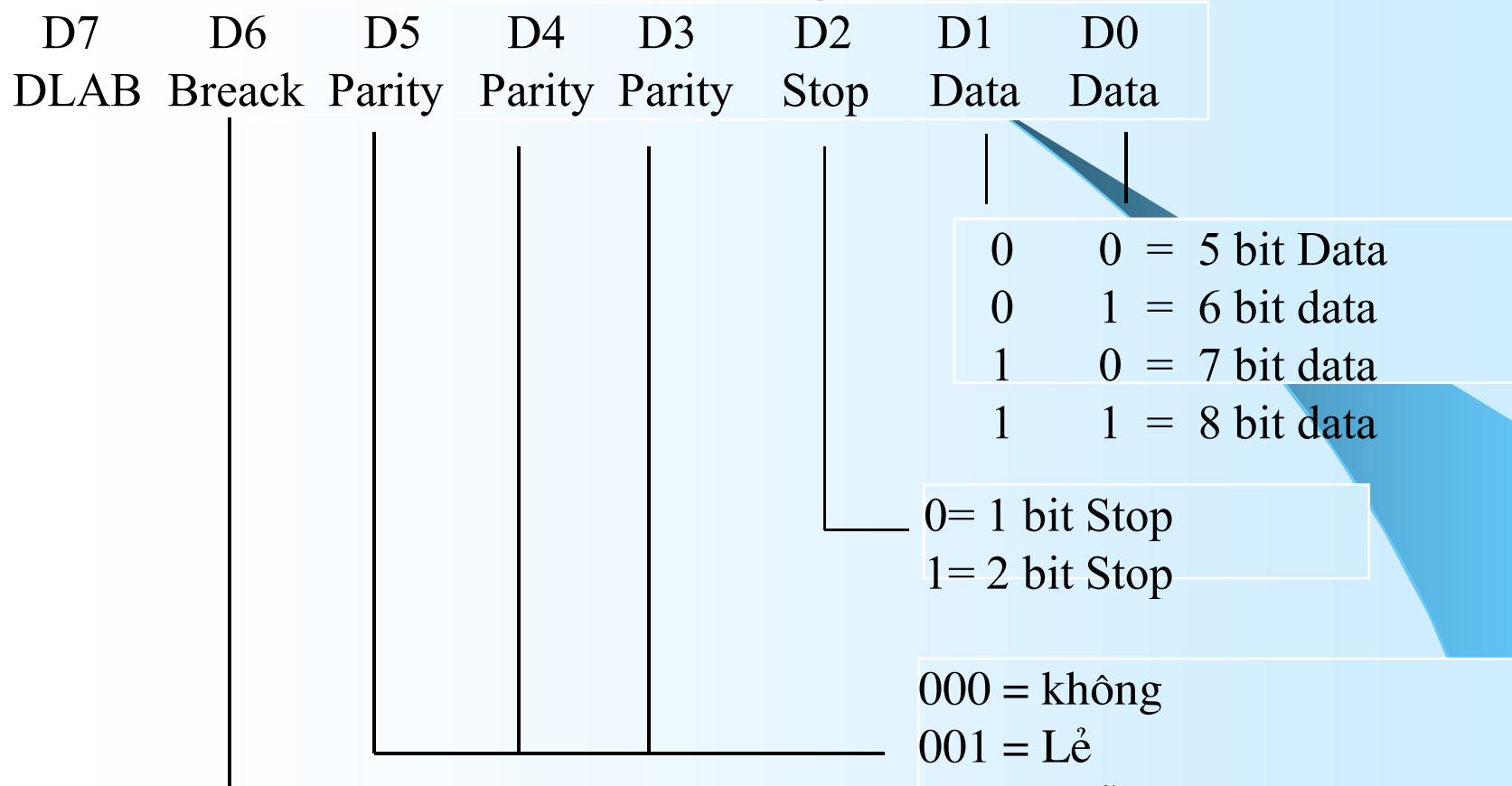
0	0	0	0	0			
---	---	---	---	---	--	--	--

			Mức ưu tiên	Ngắt
0	0	1	x	Không
1	1	0	0	Lỗi nối tiếp hóa hoặc Breack
1	0	0	1	Số liệu được nhận
0	1	0	2	Bộ đệm truyền rõng
0	0	0	3	Trạng thái Modem

Các mức ưu tiên được định nghĩa như sau:

D0	D1	D2	Loại INTR	Nguồn INTR	Mức ưu tiên
1	1	0	Trạng thái đường thu	Lỗi (tràn, định khung chẵn lẻ) hoặc stop	Cao nhất
1	0	0	Trạng thái Modem	Trạng thái Modem	Thứ 2
0	1	0	Trạng thái Modem	Trạng thái Modem	Thứ 3
0	0	0	Trạng thái Modem	CTS, DSR, RI, DDC	Thứ 4
x	x	1	Không có	Không có

Thanh ghi điều khiển đường truyền (3FBh)



Thanh ghi điều khiển Modem (3FCh)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	TEST	OUT2	OUT1	RTS	DTR

Kiểm tra cục bộ

Lối ra 2 do người dùng xác định

Lối ra 1 do người dùng xác định

= 1 xác nhận
(Data Terminal Ready- DTR = 0)

= 1 xác nhận
(Request To Send)

D0 = 1. DTR = 0

D1 = 1. RTS = 0

D2 và D3 dùng cho các chân OUT của 8250 và các hệ không tương thích RS 232

D4 cho phép kiểm tra 8250 theo phương pháp hỏi vòng. Khi D4 = 1, $S_{out} = 1$, S_{in} bị ngắt. Thanh ghi PISO nối tới thanh ghi SIPO. Các đầu vào điều khiển modem (DSR, CTS, RI, DCD) bị ngắt ở bên ngoài và bên trong được nối tới các tín hiệu điều khiển 5 Modem .

Thanh ghi trạng thái Modem (3FEh)

Được xem như trạng thái của RS 232. Bốn bit thấp có giá trị 1 khi có sự thay đổi trạng thái đầu vào so với lần đọc cuối cùng của BVXL của các tín hiệu CTS, DSR, RI, DCD. Bốn bit cao tương ứng với các tín hiệu RTS, DTR, OUT1 và OUT2.

D7	D6	D5	D4	D3	D2	D1	D0
DCD	RI	DSR	CTS	Delta DCD	Delta RI	Delta DSR	Delta CTS

Thanh ghi trạng thái đường truyền (3FDh)

Khi D2 của thanh ghi cho phép ngắn (3F9h) có mức cao, cho phép kiểm tra lỗi:

D7	D6	D5	D4	D3	D2	D1	D0
0	TEMPT	THRE	BI	PE	PE	OE	DR

= 1 :
Thanh ghi giữ giá trị phát (THR) và thanh ghi dịch phát rỗng. (Xóa khi TSR hoặc THR có số liệu)

= 1 : Số liệu sẵn sàng nhận (xóa khi đọc số liệu bộ đệm thu)

= 1 : Lỗi tràn (xóa khi CPU đọc bộ đệm thu)

= 1 : Lỗi chẵn lẻ (xóa khi CPU đọc thanh ghi trạng thái đường truyền)

= 1 : Lỗi định khung (xóa khi CPU đọc thanh ghi trạng thái đường truyền)

= 1 : Có ngắn (xóa khi CPU đọc thanh ghi trạng thái đường truyền)

= 1 : Thanh ghi lưu giá trị phát rỗng

Tốc độ truyền của 8250 được lập trình thông qua hai thanh ghi có địa chỉ 3F8h và 3F9h khi DLAB = 1. Tần số vào X_{in} được lấy từ đồng hồ chủ. Giá trị nạp vào hai thanh ghi này được tính như sau:

$$\text{Số chia} = \frac{X_{in}}{\text{Tốc độ truyền} * 16}$$

Ví dụ: Xác định số chia ghi vào các thanh ghi LSB và MSB để có được các tốc độ truyền 300 Baud với tần số đưa vào chân $X_{in} = 1,8432 \text{ Mhz}$:

$$384 = \frac{1,843,200}{300 * 16}$$

Lập trình :

Mov	al,80h	; đặt DLAB = 1
Mov	dx, 3fbh	; nạp địa chỉ thanh ghi điều khiển
đường truyền		
Out	dx, al	; đặt DLAB = 1
; gửi số chia		
mov	ax, 384	; tốc độ truyền là 300 baud
mov	dx, 3f8h	; địa chỉ phần thấp bộ chia
out	dx,al	; nạp phần thấp số chia
mov	al, ah	; chuyển phần cao số chia sang al
inc	dx	; địa chỉ phần cao số chia
out	dx, al	; nạp phần cao số chia

Sự ra đời của 16550

Vi mạch 8250/ 16450 chỉ chuyển một byte mỗi lần phát và thu số liệu vì bộ đệm phát và thu của nó chỉ có độ dài là 1 byte, do đó có tốc độ làm việc chậm. Để tăng cao tốc độ thu/phát National Semiconductor đưa ra chip 16550 à hoàn toàn tương thích với 8250 và 16450 nhưng có bộ đệm là 16 byte. Vì vậy CPU sẽ thực hiện ghi vào bộ đệm cả một block số liệu (16 byte) thay vì 1 byte như ở 8250/16450 trước đây. Và số liệu vẫn được 16550 thu/phát tuần tự từng byte một theo cách định khung cho trước

Bài tập:

1. Thực hiện nối hai máy tính qua cổng RS 232
2. Viết chương trình ASM thực hiện công việc sau:
 - a. Truyền số liệu qua hai máy dưới dạng đơn công: Khi trên máy phát gõ một dãy ký tự (kết thúc là phím Enter) thì dãy ký tự đó được truyền sang máy thu và được hiển thị trên màn hình của máy thu.
 - b. Truyền số liệu giữa hai máy dưới dạng song công: Cả hai máy cùng thực hiện nhận và phát số liệu: Mỗi lần truyền một dãy ký tự từ máy này sang máy kia. Chương trình sẽ kết thúc khi ấn phím ESC.