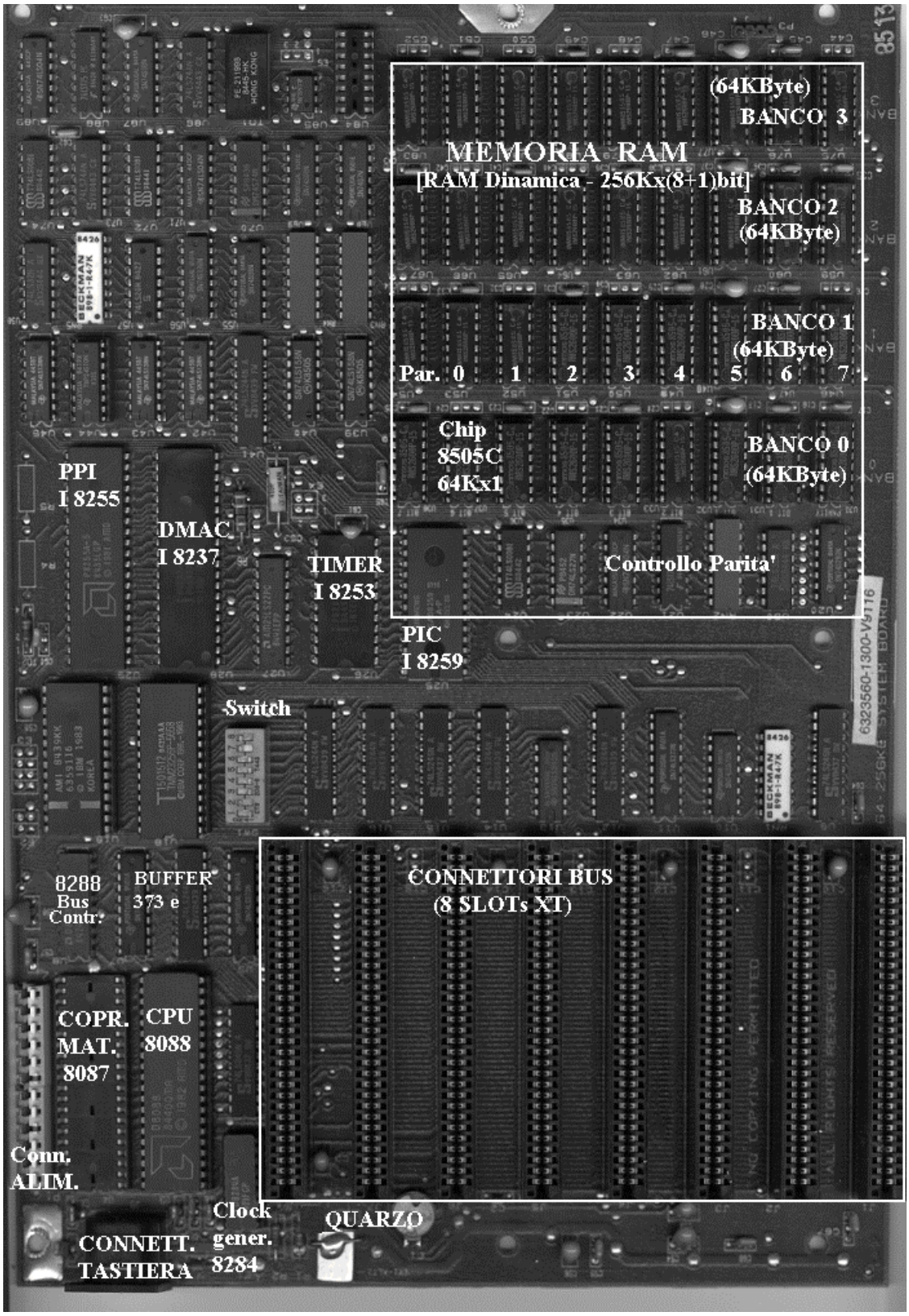
Enrico Tombelli

Docente presso
ITC "A. Volta" - Bagno a Ripoli - Firenze
(e.tombelli@libero.it)

DISPOSITIVI AUSILIARI

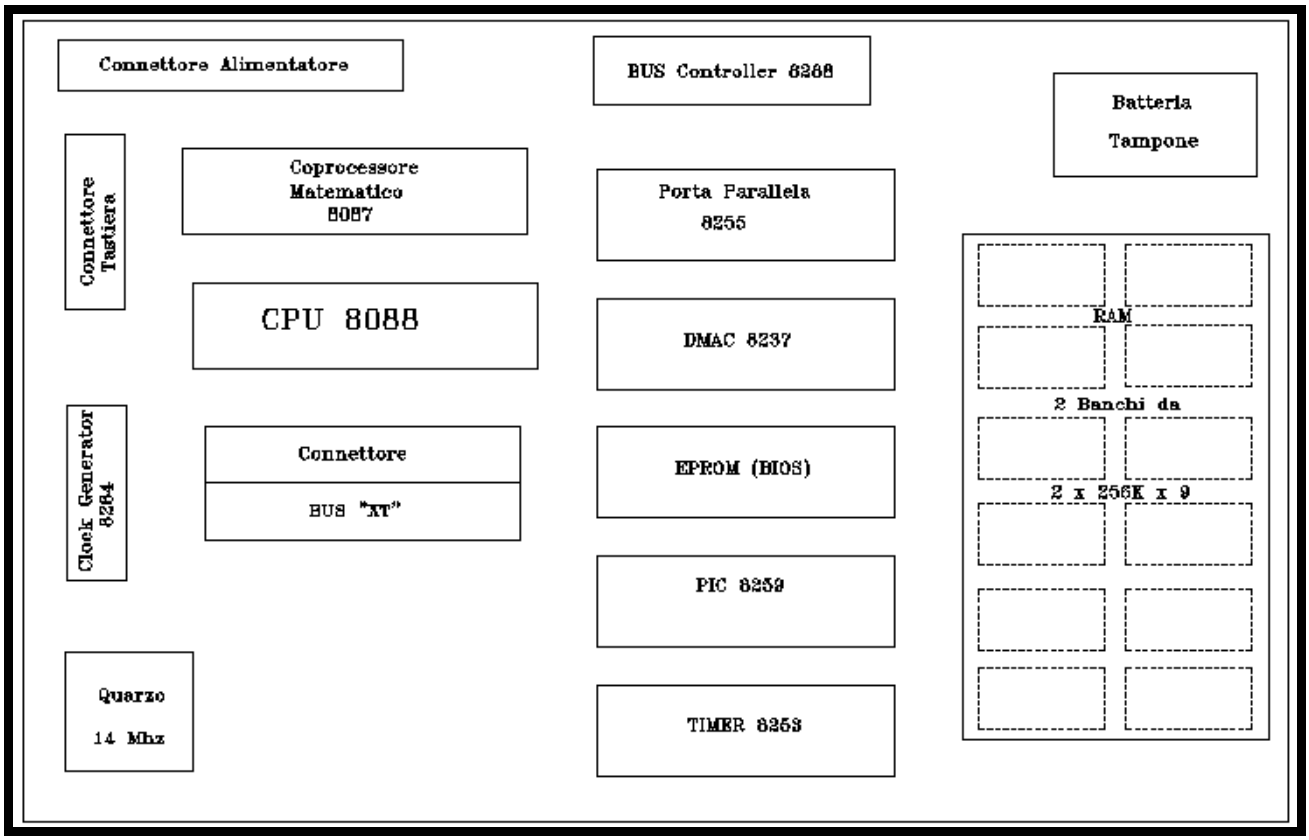
DISPOSITIVI AUSILIARI



MOTHERBOARD PC/XT IBM

Le funzioni realizzate dal microprocessore, tramite apposito programma, possono essere gestite in modo HW, affiancando alla CPU dispositivi complementari detti "ausiliari". L'utilizzo di questi dispositivi

aumenta notevolmente i costi del sistema, ma permette di velocizzare e in generale aumentare le prestazioni di tutta la logica.



Schema a blocchi della MOTHERBOARD del PC IBM

I dispositivi in oggetto sono:

- Interfaccia parallela programmabile (PPIC)
- DMAC (Controllore d'accesso diretto in memoria)
- Controllore di ritardo programmabile (PIT)
- Controllore dell'interrupt (PIC)
- Interfaccia seriale Programmabile (UART)

- CONTROLLORE PROGRAMMABILE D'INTERRUPT

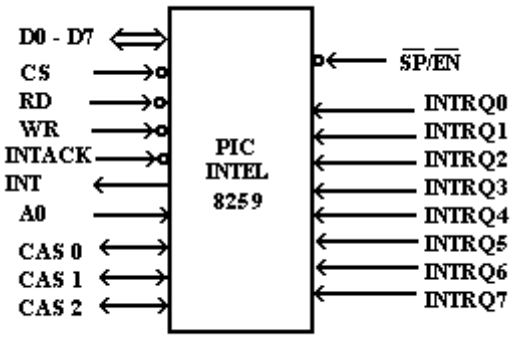
É un dispositivo che permette la gestione in modo HW e in modo programmabile, del controllo delle INTERRUZIONI con tutti i problemi connessi alla PRIORITÀ e al riconoscimento della periferica che ha richiesto l'operazione di I/O.

- PIC INTEL 8259

La sigla "PIC" sta ad indicare "Programmable Interrupt Controller" ovvero "Controllore Programmabile d'Interrupt ".

Il PIC INTEL 8259 é un dispositivo che dispone delle seguenti prestazioni:

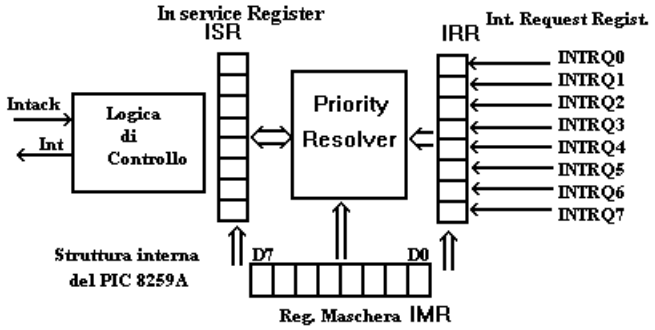
- Gestione dell'interrupt fino a 8 richieste in modo programmabile.
- Gestione della Priorità.



- Mascheratura dell'interrupt.
- Vettorizzazione automatica
- Tecnologia NMOS con 28 pin.
- Disposto in cascata permette la gestione fino a 64 periferiche

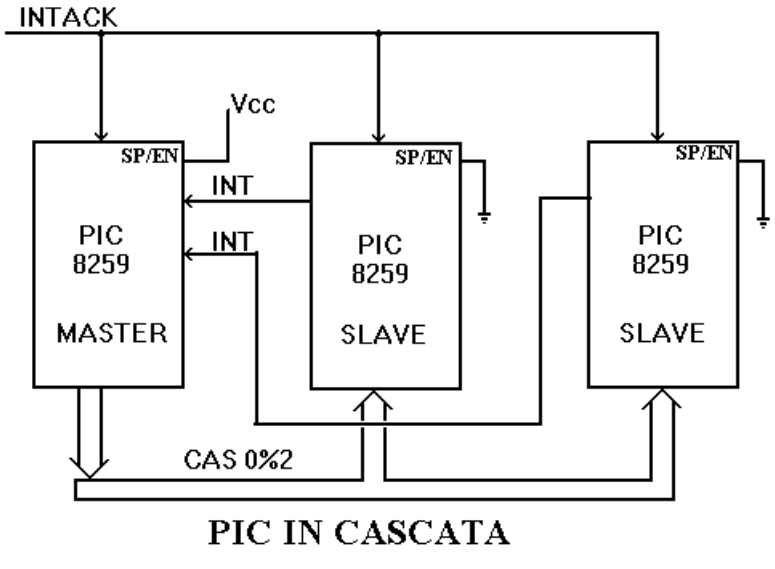
Quando una qualsiasi delle linee d'interrupt d'ingresso è attivata, scatta un sistema di riconoscimento della linea, e viene attivata la linea d'interrupt in uscita che é collegata a quella della CPU.

A parte i segnali comunemente usati per le varie operazioni di controllo (RD, WR, INT ecc.), ci sono altri segnali la cui funzione é meno ovvia:



SP/EN (Slave Progress / Enable Buffer) che stabilisce il modo master (1 ⇒ Collegato alla alimentazione Vcc) o se il PIC é slave (0 ⇒ Collegato a massa). In questo ultimo caso, la sua linea di INT, é collegata ad una linea di INTRQ d'ingresso ad un altro PIC che funge da Master (SP/EN=1). É possibile così gestire più PIC in cascata.

CAS 0%2 servono anch'essi per la gestione in cascata dei PIC. Collegandoli insieme parallelamente e in modo omologo permettono la comunicazione da un PIC all'altro, e possono essere sia ingressi sia uscite a seconda che il PIC sia rispettivamente Slave o Master.



PIC IN CASCATA

Tali linee servono al Master per informare il PIC slave interessato al servizio che il suo turno é stato accettato e che quindi é lui che deve porre sul bus dati il codice dell'interrupt da inviare alla CPU. Le tre linee "CAS 0%2" sono sufficienti ad identificare un PIC su 8.

Tutte le linee d'interrupt fanno capo ad un registro ad 8 bit, detto "Interrupt Request Register" ("IRR") che memorizza le varie richieste, in modo che non sia necessario far permanere attivo il segnale d'interrupt proveniente da ogni periferica (è sufficiente un impulso). Tale registro conserva quindi la situazione delle richieste d'interrupt ancora da servire.

In un altro registro a 8 bit detto "In Service Register" (ISR), é memorizzato il livello d'interrupt che é in servizio in quel momento. Si deve osservare che potendo esserci più linee d'interrupt in servizio contemporaneo ("fully nested mode") il registro ISR può contenere più flag a "1" contemporaneamente;

uno per ogni livello d'interrupt in servizio in quel momento. É ovvio che attualmente in esecuzione é quello a più alta priorità dato che ha bloccato tutti gli altri.

Nel registro di "IMR" ("Interrupt Mask Register") é memorizzata la mascheratura delle linee d'interrupt. Tale registro è caricato da programma con il carattere di mascheratura.

In base al contenuto di questi registri, una rete opportuna, detta "Priority Resolver", determina quale, fra le linee che hanno richiesto l'interrupt verrà eseguita successivamente.

Quando una linea di INTRQ viene attivata e se questa non é disabilitata, il PIC inoltra la richiesta d'interrupt alla CPU, la quale provvede ad effettuare le varie operazioni di rito, prima di cedere il controllo al servizio richiesto. Come già noto, l'accolta dell'interrupt da parte della CPU, viene manifestata all'esterno tramite la generazione del segnale di INTACK che arriva al chip 8289. In risposta esso ripone sul bus dati il codice relativo alla linea a più alta priorità che ha richiesto il servizio (Il PIC lavora in questo caso con modalità INTEL, ma é possibile selezionare anche la modalità 8085). Inoltre viene portato a "1" il relativo flag nel registro ISR per memorizzare che il livello d'interrupt in questione é attualmente in servizio. Tale flag viene resettato o previo comando di EOI al PIC da parte della CPU (apposita istruzione) o in modo automatico. La CPU interpreta il codice binario come numero dell'interrupt e salta alla routine del relativo servizio (oppure come indirizzo della routine nel caso di modalità 8085. In quest'ultimo caso vengono attivati 3 impulsi di INTACK per acquisire dal PIC un'istruzione di CALL o JUMP completa).

La componente d'interrupt deve essere precaricata nel PIC durante la fase di inizializzazione. Tale componente é costituita da una parte comune a tutte le linee d'interrupt, mentre la parte variante é diversa per ogni linea ed é stabilita direttamente dal PIC.

Questa operazione può essere effettuata tramite delle sequenze di 2 oppure 4 byte dette "Inizialisation Command Word" (ICW1 % ICW4), la prima delle quali avvia la sequenza di inizializzazione e ne definisce la lunghezza. Nelle prime due ICW risiede la componete comune del vettore d'interrupt.

Il PIC é visto dall'esterno tramite due soli indirizzi che differiscono tra loro solo per il valore di A0. Infatti, la linea A0 del bus indirizzi viene collegata direttamente al piedino omonimo del PIC, mentre le altre linee vanno decodificate e pilotano direttamente il CS del chip. Gli indirizzi del PIC sono uno pari e uno dispari.

L'indirizzo al quale inviare il byte di inizio sequenza é quello del PIC (ovvero quello che abilita il CS) con la linea A0=0 (Indirizzo Pari), e viene riconosciuta dal chip in quanto, fra i flag di informazione che la formano, il bit 4 viene posto a 1 (D4=1), le altre ICW invece vengono caricate all'indirizzo dispari (A0=1).

Oltre ai comandi d'inizializzazione, ce ne sono altri per la definizione del modo operativo ("Operation Command Words"). Anche questi comandi sono costituiti da una sequenza di massimo 3 byte, (OCW1 % OCW3), i primo dei quali é inviato all’indirizzo dispari (A0=1) viene caricato direttamente nel registro maschera (IMR). Gli altri byte della sequenza sono, invece, inviati all'indirizzo pari (A0=0).

- ICW (INIZIALISATION COMMAND WORD)

Per inizializzare il PIC si caricano le ICW in sequenza ponendo A_0=0 (come bit meno significativo dell’indirizzo del PIC) e fornendo il dato che rappresenta la prima ICW (1) in modo che D_4 sia "1". Il valore di A0 per il resto della sequenza (ICW2%ICW4) é 1. Per quanto riguarda gli altri bit relativi alla ICW1 e ICW2 si ha quanto riportato di seguito:

$ICW2_{(7, \dots ,0)}$ e $ICW1_{(7,6,5)}$ = A15, A14, ... A5 rappresentano l'indirizzo della routine relativa alla prima linea d'interrupt. Dato che l'indirizzo non é completo (mancano i bit relativi a A4%A0), si presuppone che essi siano variabili in funzione della linea d'interrupt interessata

(Esempio: spaziatura di 8)

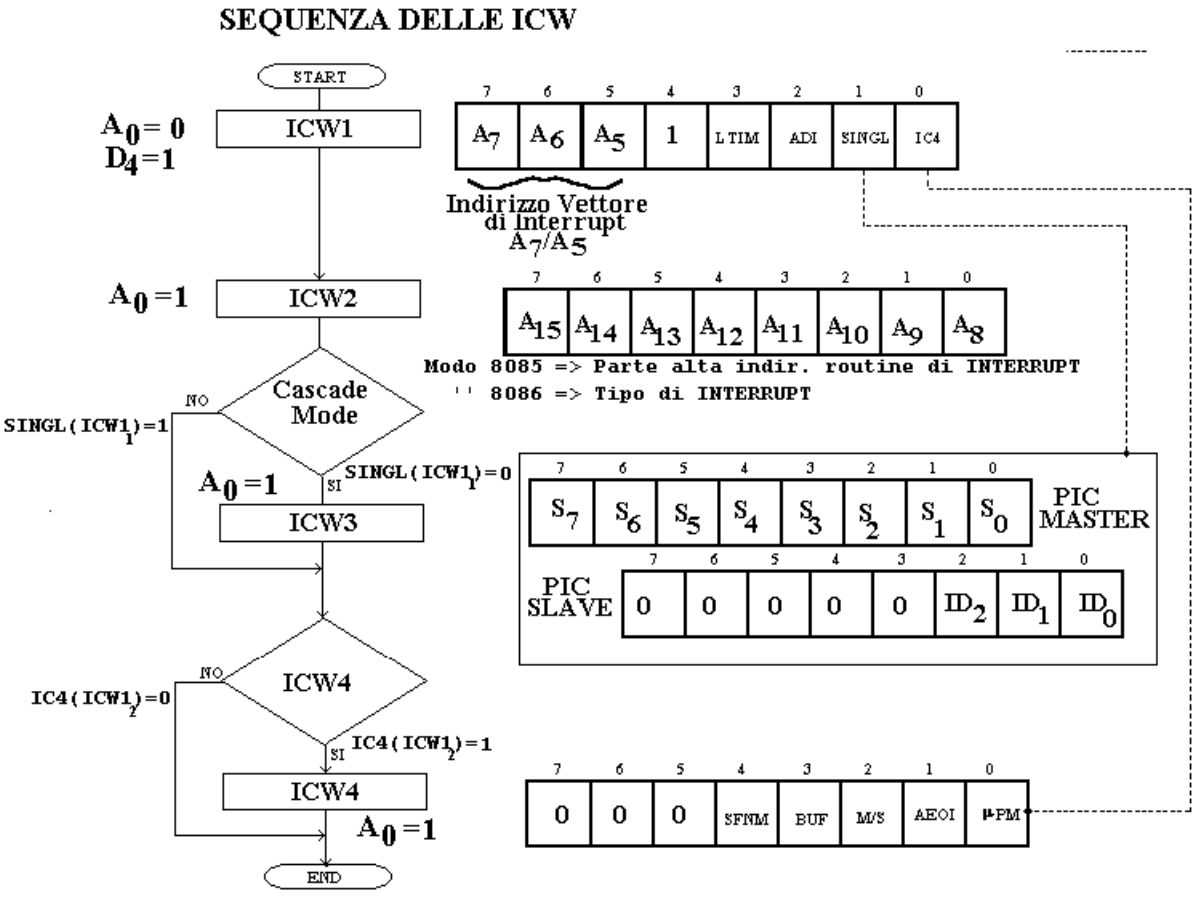
	A15	A5	A4	A0	HEX
linea 0⇒	1001 1100	100	0	00	9C80h
linea 1⇒	1001 1100	100	0	10	9C88h
linea 2⇒	1001 1100	100	1	00	9C90h
linea 3⇒	1001 1100	100	1	10	9C98h
linea 4⇒	1001 1100	101	0	00	9CA0h
linea 5⇒	1001 1100	101	0	10	9CA8h
linea 6⇒	1001 1100	101	1	00	9CB0h
linea 7⇒	1001 1100	101	1	10	9CB8h

(Esempio: spaziatura di 4)

	A15	A5	A4	A0	HEX
linea 0⇒	1001 1100	100	0	00	9C80h
linea 1⇒	1001 1100	100	0	01	9C84h
linea 2⇒	1001 1100	100	0	10	9C88h
linea 3⇒	1001 1100	100	0	11	9C8Ch
linea 4⇒	1001 1100	100	1	00	9C90h
linea 5⇒	1001 1100	100	1	01	9C94h
linea 6⇒	1001 1100	100	1	10	9C98h
linea 7⇒	1001 1100	100	1	11	9C9Ch

$ICW1_{(3)}$ = LTIM stabilisce se l'interrupt viene lanciato quando la linea d'interrupt viene portata a livello alto ("1") o se invece il PIC é sensibile al fronte di salita ("0"). Nel secondo caso é possibile richiedere l'interrupt semplicemente tramite impulso.

$ICW1_{(2)}$ = ADI (Address Interval) Definisce la spaziatura fra gli indirizzi delle routine d'interrupt. Infatti, essa può essere di 8 (ADI=0,allora, presupponendo A_{15},= ... =A_5=0 gli indirizzi possono essere 00h, 08h, 10h, 18h, 20h, 28h, 30h, 38h) oppure di 4 (ADI=1, allora, presupponendo A_{15},= ... =A_5=0, gli indirizzi possono essere 00h, 04h, 08h, 0Ch, 10h, 14h, 18h, 1Ch, 20h). Tali spaziature servono per permettere la scrittura di un numero sufficiente d'istruzioni per ogni routine. É ovvio che se la routine occupa più spazio di 4 o 8 celle, nel punto d'ingresso viene



messo un salto (JMP) od una chiamata ad un sottoprogramma (CALL) che svolge la funzione.

N.B. Nel caso che si lavori in modalità 8086 (vedi µMP[ICW4]=1), visto che i valori di address rappresentano il tipo d'interrupt, lo spazio fra un valore e il successivo é unitario, ovvero, stabilito il primo valore d'indirizzo (TIPO) gli altri sono consecutivi (es. presupponendo A_7,=A_6=A_5=0, si ha 00h, 01h, ...07h)

$ICW1_{(1)}$ = SINGL definisce se si lavora in modo singolo (1=SINGLE) o in "Cascata" (0=CASCADE), ovvero se il sistema é corredato di altri PIC collegati fra loro per disporre di più linee d'interrupt. Nel caso si lavori in "SINGLE" non viene richiesta la ICW3 la cui presenza serve proprio per definire l'identificatore del PIC slave.

$ICW1_{(0)}$ = IC4 Definisce la presenza (1) o no (0) della ICW4 (se no ci si ferma alla ICW3 nella sequenza). In mancanza della ICW4 si determina una situazione di default.

La ICW3 é presente solo nella modalità "CASCADE" (ovvero il PIC funziona in concerto con altri PIC in cascata). A seconda che il PIC interessato sia master o slave, la ICW3 é diversa (N.B. il fatto che un PIC sia master o slave é definito dal pin SP/EN): Nel PIC master il generico bit della ICW3 indica se la relativa linea d'interrupt é collegata ad uno slave o no; nel PIC slave invece contiene, (nei tre bit meno significativi $IC3_{2,1,0}$) il proprio codice di identificazione (valore da 0 % 7)

La ICW4 può essere omessa ponendo il LSB della ICW1=0 e in questo caso viene attivata la situazione di default. Se c'é, essa contiene le seguenti informazioni:

$ICW1_{(7,6,5)}$ = [0,0,0] fisse

$ICW1_{(4)}$ = SFNM (special fully nexted mode) modo completamente nidificato in modo speciale. La nidificazione si riferisce al fatto che ci possono essere più interrupt in servizio contemporaneamente, e ciò é la situazione ordinaria. Il SFNM é una estensione del modo nidificato che viene esteso anche ai PIC slave nel caso che il sistema funzioni con più PIC in cascata.

$ICW1_{(3)}$ = BUF (Buffered mode) se 1 si lavora in modo bufferato, ovvero i dati che vengono messi dal PIC sul bus vengono trattenuti finché non vengono letti da un altro dispositivo (es. la CPU). In tale modalità il PIC utilizza il pin SP/EN che ha la doppia funzionalità(che in questo caso funziona da enable) come uscita per comunicare all'esterno che il buffer con il dato da acquisire é pieno. Pertanto il pin SP/EN non può essere utilizzato per differenziare la modalità master da quella slave (funzione master/slave). Per questo é possibile definire il PIC master o slave (nella modalità Cascade) anche tramite i flag M/S sempre nella ICW3.

$ICW1_{(2)}$ = M/S (Master/slave) In caso "bufered mode" segnala se il PIC é master o slave. Se non siamo in modalità "buffered" tale flag non é significativo, ma viene utilizzato direttamente il pin SP/EN che in questo caso svolge la funzione SP.

$ICW1_{(1)}$ = AEOI Se a "1" é attiva la modalità di EOI ("end of interrupt") automatico nel sistema pienamente nidificato.

$ICW1_{(0)}$ = µPM Stabilisce se il microprocessore del sistema é 8080 (8085) oppure 8086. Questo flag influisce quindi sul modo col quale viene risposto all'INTACK. Nel caso dell'8085 il MP si aspetta un indirizzo; nel caso dell'8086 invece si aspetta il N. del tipo d'interrupt.

Sequenza di esempio (si suppone che al PIC siano associati gli indirizzi di I/O 20h e 21h):

N. ICW	Indirizzo	A_0	Valore Binario	Valore HEX	Significato
1	20h	0	xxx10x11	XXh	Si usa il modo di test dell'interrupt sensibile al fronte (LTIM=0), il PIC funziona in modo Singolo (SINGL=1) ed é necessaria la ICW4 (IC4=)
2	21	1	01101101	60h	La prima linea d'interrupt é associata al tipo 60h; le altre in successione (61h, 62, ecc)
3	21	1			NON USATA in quanto non siamo in modo "CASCADE"
4	21	1	10000x01	8Xh	Non viene utilizzata la nidificazione speciale (SFNM=0), modalità non bufferata (BUF=0 e M/S=X); microprocessore 8086

- OCW (OPERATION COMMAND WORD)

Le OCW sono tre e vengono utilizzate per il funzionamento a regime del PIC, ovvero per le operazioni di ordinaria amministrazione (Mascheratura, rotazione della priorità, comandi vari ecc.). Queste a differenza delle ICW non vengono necessariamente comunicate in sequenza. infatti si possono riconoscere da A0 e da D3 e D4[1]

La prima OCW ha indirizzo dispari (A0=1), mentre le altre due hanno indirizzo pari (A0=0).

La prima OCW é il registro maschera (0 ⇒ linea attiva).

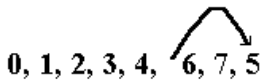
La OCW2 gestisce l'EOI e la priorità. Essa contiene i seguenti flag:

$OCW2_{(7,6,5)}$ = R, RR, EOI. Questi flag servono per comandare il PIC in EOI e per la gestione della priorità. Nella tabella sono elencati i codici da utilizzare per i vari comandi.

Tramite questa OCW é possibile stabilire il modo di gestione della priorità eventualmente ruotante per evitare che i livelli a priorità più bassa non vengano mai eseguiti. La priorità può essere gestita automaticamente dal PIC in modo RUOTANTE: le linee d'interrupt appena servite passa a livello di priorità più basso, mentre le altre shiftano di conseguenza a priorità di un livello più alto.

R	RL	EOI	Azione
0	0	1	Comando EOI non specifico. É utile nella nodalità pienamente nidificata perché il flag dell'ISR che viene resettato é quello relativo all'interrupt in servizio con priorità più alta perché é sicuramente quello attualmente in esecuzione..
0	1	1	Comando EOI specifico. Serve per definire un comando di EOI indirizzato ad una specifica linea d'interrupt. La linea interessata é quella identificata dai bit 0,1 e2 della OCW2 (L2,L1 e L0).
1	0	1	Rotazione su uno specifico comando di EOI:
1	0	0	Attivazione della rotazione su ricezione di un comando di EOI automatico
0	0	0	Disattivazione della rotazione su ricezione di un comando di EOI automatico
1	1	1	Rotazione su ricezione di uno specifico comando di EOI
1	1	0	Attivazione del comando di cambio priorità. Viene portata a livello più basso la linea specificata dai flag meno significativi della ICW2.
0	1	0	non usata

Esempio: é attualmente in servizio la linea 5 e la priorità é attualmente decrescente col numero di linea (la linea 0 é a priorità max). Appena servita, la linea 5 passa a livello priorità minimo e le altre linee scalano di un livello.



É possibile far svolgere la rotazione in due modi, ovvero in seguito ad un EOI specifico o automatico. É possibile anche fissare la priorità in modo programmato, con la combinazione "1 1 0". in questo caso viene portato a priorità piu bassa il livello d'interrupt che é specificato nei flag L2, L1, L3 della $OCW2_{(2,1,0)}$.

$OCW2_{(2,1,0)}$ = L2, L1, L0 identificano la linea d'interrupt interessata per l'indizzamento.

[1]Si deve notare chele ICW vengono date in sequenza e iniziano con A0=0 e D4=1, dopodiché si procede alla comunicazione delle ICW successive. Le OCW invece sono identificate da D4=0 a parte la prima (registro maschera, che comunque ha A0=1, a differenza delle altre che hanno A0=0. La OCW2 si differenzia dalla 3 per il flag 3 (OCW2=>D3=0; OCW3=>D3=1). Per questo non c'é possibilità di equivoco.

La OCW3 serve per inviare particolari comandi

$OCW3_{(6,5)}$ = ESMM e SMM attivano e disattivano la modalità "Special mask mode".

ESMM	SMM	Azione
1	0	Disattiva lo SMM
1	1	Attiva lo SMM

Il sistema di mascheratura ordinario é quello stabilito dal registro maschera IMR). É possibile però gestire la mascheratura delle linee d'interrupt in altro modo che permette di modificare ‘da programmà la struttura delle priorità delle linee d'interrupt. Con il sistema di SMM disabilitato il sistema di gestione della priorità non permette che possa essere interrotto in servizio d'interrupt da parte di un livello a più bassa priorità. Se il sistema SMM é invece attivo é possibile inibire una linea tramite la OCW1 e abilitare conporaneamente tutte le altre.

$OCW3_{(2)}$ = P (polling). Viene attivata la modalità "Polling" (se=1), ovvero il PIC non genera un segnale di INT su richiesta delle linee d'interrupt, ma testa su comando della CPU (tramite appunto tale flag) lo stato delle linee e poi la CPU legge dal PIC il codice (a 3 bit) della periferica a più alta priorità che ha richiesto il servizio.

$OCW3_{(1,0)}$ = (RR e RIS) Questi due flag servono per il comando di lettura dei registri ISR e IRR. Per leggere questi registri occorre inviare il comando e poi svolgere la lettura

RR	RIS	Azione
1	0	Lettura dell’ISR
1	1	lettura dell’IRR

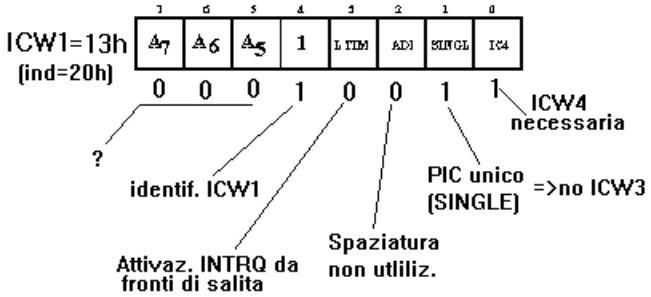
IDENTIFICAZIONE DELLE ICW e OCW

I/O CW	A0	D4	D3
ICW1	0	1	X
ICW2	1	X	X
ICW3	1	X	X
ICW4	1	X	X
OCW1	1	X	X
OCW2	0	0	0
OCW3	0	0	1

ESEMPIO D'INIZIALIZZAZIONE DEL PIC (vedere Tecnical Reference del PC/XT):

NB: si presuppone che il PIC abbia indirizzi 20h (A0=0) e 21h (A0=1).

1) CARICAMENTO DELLA ICW1: poniamo A0=0 e D4=1.



Si inserisce quindi nella ICW1, che ha indirizzo 20h, il valore 13h.

```
MOV AL , 13h
OUT 20,AL
```

2) CARICAMENTO DELLA ICW2: poniamo A0=1 e trasferiamo nella componente il valore 08h (tipo interrupt relativo alla prima linea)

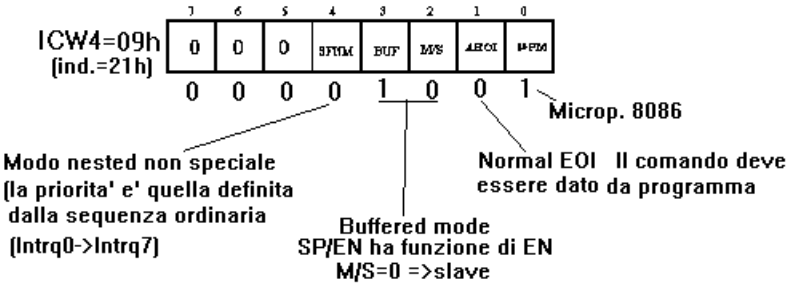

```
MOV AL , 08h
OUT 21h, AL
```

3) La ICW3 non é richiesta in quanto il PIC funziona in modo SINGLE

ICW3 = non utilizzata

4) CARICAMENTO DELLA ICW4. Essa é stata richiesta nel LSB della ICW1. A0=1 e il valore che viene caricato in essa é il 09h.

```
MOV AL,09h
OUT 21h,AL
```



- UTILIZZO DEL PIC 8259 INTEL IN UN COMPUTER IBM

In un personal IBM tipo XT, il PIC ha indirizzi corrispondenti a 20h per l'indirizzo pari e 21h per quello dispari.

L'inizializzazione del PIC é effettuata dal sistema operativo, ma é possibile modificare, p. es. la mascheratura delle linee di ingresso scrivendo il relativo byte-maschera nella prima OCW, ovvero all'indirizzo 21h.

LINEA	CODICE INT.	DISPOSITIVO
IRQ 0	08h	Timer (Canale 0)
IRQ 1	09h	Tastiera
IRQ 2	0Ah	Interfaccia Grafica a Colori (PIC Slave nel 286)
IRQ 3	0Bh	Seconda interf. seriale (COM2 e COM4)
IRQ 4	0Ch	Interfaccia Seriale (COM1 e COM3)
IRQ 5	0Dh	Non Usato
IRQ 6	0Eh	Dischi
IRQ 7	0Fh	Stampante

Nel personal IBM, inoltre le varie linee d'interrupt sono così utilizzate:

P.e. se si vuole disabilitare (1) o abilitare (0) la linea d'interrupt della stampante si deve inserire il valore opportuno nel bit 7 della porta 21h.

Per disabilitare la stampante e abilitare gli altri dispositivi:

```
MOV     AL,80h
OUT     21h,AL
```

NB: La routine di gestione dell'interrupt deve provvedere a chiudere l'interrupt ponendo il codice EOI (20h) nella OCW2.

```
MOV  AL,20h
OUT  20h,AL
```

Questa operazione si rende necessaria per resettare automaticamente il bit che contrassegna l'interrupt a più alta priorità (che ovviamente é attualmente in esecuzione) nel registro ISR ("In Service Register"), e poter quindi cedere il controllo dell'interrupt al processo precedentemente interrotto che é a priorità più bassa.

NB: Per differenziare l'accesso alla ICW1 da quello alla OCW2 (per le quali si pone per entrambi A0=0) si imposta il bit D4 diverso. Al contrario per differenziare la ICW2 dalla OCW1. In questo modo non si crea ambiguità nell'accesso.

ALTRI ESEMPI:

1) Lettura dell'ISR (il contenuto dell'ISR viene messo nel registro AL)

```
0100 B0 0B  MOV AL, 0Bh   (*)
0102 E6 20  OUT 20h, AL
0104 EB 00  JMP 106h      Attesa/ritardo
0106 E4 20  IN AL, 20h    ISR →AL
```

(*) 0000 1011 = 0Bh ⇒ OCW3

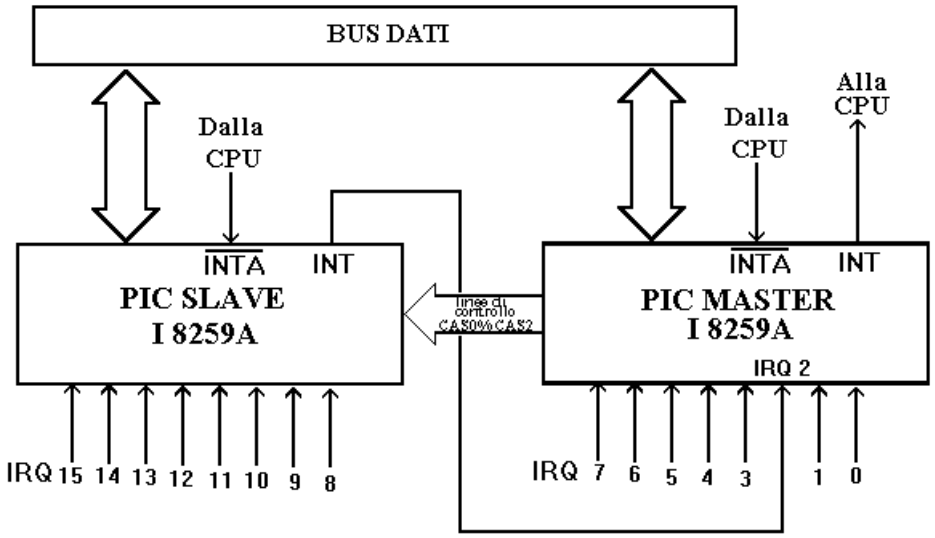
2) Lettura dell'IRR (il contenuto dell'IRR viene messo nel registro AL)

```
0100 B0 0A  MOV AL, 0Ah   (*)
0102 E6 20  OUT 20h, AL
0104 EB 00  JMP 106h      Attesa/ritardo
0106 E4 20  IN AL, 20h    IRR→ AL
```

(*) 0000 1010 = 0Ah ⇒ OCW3

NOTA:
Nei sistemi più evoluti (80286) ci sono due PIC in cascata uno dei quali é "master" e l'altro "slave". Infatti la linea di INT di quest'ultimo é collegata all'ingresso IRQ2 del Master. Gli indirizzi di cui sopra si riferiscono al master, mentre lo slave ha indirizzi A0h e A1h.
Per quanto riguarda la priorità siccome il PIC slave é collegato alla linea 2 del PIC master si ha che tutte le linee del PIC slave hanno priorità maggiore delle linee del master al di sopra del 2. Pertanto la sequenza delle priorità e:
0 (max), 1, (8, 9, 10, 11, 12, 13, 14, 15,) 3, 4, 5, 6, 7 (min).
PIC SLAVE

É necessario, alla fine della routine, inviare il comando di EOI a tutti i PIC collegati.



SCHEMA DI COLLEGAMENTO DEI DUE PIC IN CASCATA (SISTEMA I 80286)

I numeri associati ai tipi d'interrupt relativi al PIC slave sono riportati nella seguente tabella

LINEA PIC SLAVE	LINEA ASSOLUTA	NUMERO INTERR.	DISPOSITIVO
IRQ 0	IRQ 8	70h	Clock in tempo reale
IRQ 1	IRQ 9	71h	EX IRQ2
IRQ 2	IRQ 10	72h	
IRQ 3	IRQ 11	73h	
IRQ 4	IRQ 12	74h	
IRQ 5	IRQ 13	75h	COPROCESSORE MATEMATICO
IRQ 6	IRQ 14	76h	Controller disco fisso
IRQ 7	IRQ 15	77h	

- DMA (DIRECT MEMORY ACCESS)

Il DMA o accesso diretto in memoria é un tipo di operazione di I/O che, grazie ad un chip dedicato, permette di realizzare il trasferimento dei dati in modo diretto dalla periferica alla memoria, o viceversa, in modo estremamente veloce ed efficiente.

Ciò é dovuto al fatto che l'operazione é svolta in modo completamente HW dal chip già citato detto DMAC (DMA Controller).

L'operazione di I/O tramite DMA, viene svolta dal DMAC, che si sostituisce alla CPU, in modo tale che i dati, fluiscono dalla periferica alla memoria o viceversa senza passare attraverso la CPU o qualsiasi altro supporto intermedio. Il controllo dell'operazione é svolto completamente dal DMAC.

Il DMAC é quindi un dispositivo molto complesso e costoso, (più costoso della CPU, visto che ha una produzione più limitata), e viene utilizzato tutte le volte che si ha a che fare con periferiche molto veloci (CRT, Drive per dischi, Scanner, schede sonore multimediali, conversione A/D e D/A ecc.).

- DMAC

É un processore specializzato per il trasferimento dei dati ad alta velocità, grazie non solo al tipo di logica utilizzata, ma anche alla possibilità di trasferire interi blocchi di dati.

L'attivazione del DMAC comporta la piena disponibilità del bus, che pertanto deve essere lasciato libero dalla CPU.

Per questo , quando il DMAC é in azione, il microprocessore é sconnesso dal bus in quanto ha tutte le uscite in alta impedenza (la sconnessione dal bus é quindi solo a livello logico e non fisico), e si dice che la CPU si trova nello STATO DI "WAIT".

La tecnica utilizzata dal DMAC dipende dal tipo d'integrato, ma é comunque possibile descrivere il processo di DMA in modo generico.

Ciascuna periferica spedisce la richiesta di servizio di I/O generando un segnale d'interrupt, il quale viene intercettato dal DMAC. Esso, a differenza della CPU, ha un ingresso di INT per ogni periferica da gestire, ed ad ognuna di esse corrisponde una sezione del DMA indipendente dalle altre.

Generalmente il numero di collegamenti (CANALI) gestibili dal DMAC é 8 e ognuno di essi é costituito da:

- Un REGISTRO DI INDIRIZZO BASE (16 bit), che contiene l'indirizzo della cella di memoria a partire dalla quale é memorizzato il blocco di dati da trasferire (o la destinazione del blocco)
- Un REGISTRO CONTATORE (8/16 bit), che contiene inizialmente la lunghezza (in caratteri) del blocco di dati. Man mano che i caratteri del blocco sono trasferiti, il contatore viene decrementato.

P.e.

REG. IND. BASE	= 0561h
REG. CONTATORE	= 08h

Ciò vuol dire (p.e. nel caso di trasferimento dalla memoria alla periferica) che il blocco di memoria si trova all'indirizzo 0561h ed é lungo 8 caratteri.

Durante la fase di inizializzazione, il DMAC deve essere programmato, ovvero vanno inserite, nei suoi registri interni le informazioni necessarie a specificare le modalità con le quali deve avvenire il trasferimento dei dati (P. Es. devono essere riempiti i registri di indirizzo base e contatore, nonché tutti i parametri per il selezionamento della periferica.

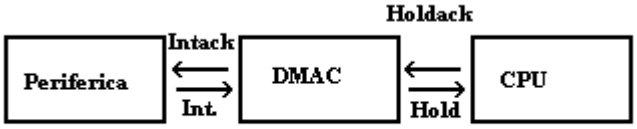
Quando al DMAC arriva il segnale di INT, inviato da una periferica che richiede una operazione di I/O, questo la inoltra a sua volta alla CPU per la richiesta del bus. Il segnale che arriva alla CPU, non é il segnale d'interrupt, ma quello di richiesta del bus, detto di "HOLD".

La CPU può non accogliere la richiesta, a seconda delle condizioni, ma se la accetta, pone tutte le sue uscite in ALTA IMPEDENZA e invia al DMAC il segnale di accettazione detto HOLDACK ("hold acknowledge"). Il microprocessore é come sconnesso dal bus (STATO DI WAIT) e il DMAC può procedere con l'operazione di I/O.

In risposta al segnale di HOLDACK proveniente dalla CPU, il DMAC invia alla periferica il segnale di riconoscimento dell'interrupt ("INTACK"), come se fosse lui stesso la CPU (La periferica ignora se la gestione avviene direttamente dal DMAC oppure se viene effettuata dalla CPU).

NOTA: Ogni canale del DMAC ha una priorità per la gestione della richiesta simultanea del servizio.

Nota la periferica, il DMAC, provvede al trasferimento del primo carattere prelevandolo dalla (o riponendolo alla) locazione di memoria puntata dal registro di indirizzo base, poi è decrementato il contatore e incrementato il registro con l'indirizzo base. Successivamente viene trasferito il secondo carattere e così via finché non é stato trasferito l'intero blocco di dati oppure la periferica ritira la richiesta di I/O.



Il trasferimento può essere interrotto nel caso in cui venga generata una richiesta di I/O da parte di una periferica a più alta priorità, per poi essere ripreso dopo aver concluso il servizio della periferica a priorità maggiore.

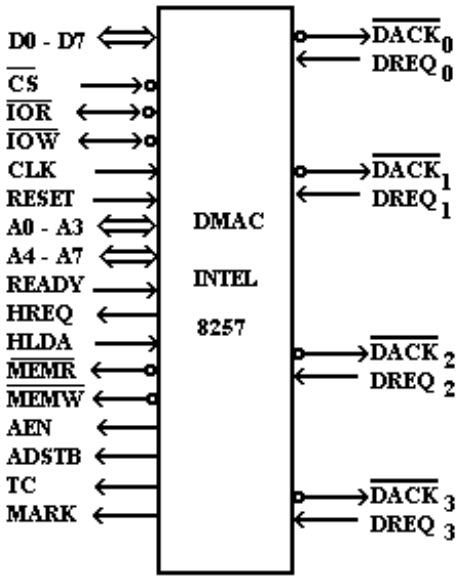
Conclusa l'operazione, il DMAC ritira la richiesta del bus, e il microprocessore può riprendere il programma interrotto.

Per velocizzare il più possibile il trasferimento, esiste la possibilità di caricamento dei valori iniziali su registri tampone, che vengono precaricati durante il trasferimento precedente.

- ESEMPIO DI DMAC: INTEL 8257

Permette di gestire 4 CANALI collegati ad altrettante periferiche, e può funzionare in due modi operativi:

- Byte by Byte. In questo modo, ad ogni richiesta della periferica viene trasferito solo un carattere per volta.
- Burst Mode. Viene trasferito un intero blocco di dati, con una stessa richiesta, senza restituire il controllo del bus al microprocessore.



L'8257, può effettuare anche delle operazioni di confronto, fra il carattere proveniente dalla periferica e un carattere campione disposto in una cella di memoria. In questo modo, si può controllare la fine dello scambio di dati, in funzione di un carattere di controllo che identifica la fine del blocco.

Ognuno dei 4 canali dispone dei due registri per l'indirizzo base ("DMA Address Register" 16 bit, che viene incrementato dopo ogni singolo trasferimento in modo da puntare sempre al prossimo carattere da trasferire) e il contatore del numero di caratteri scambiati ("Terminal Count Register" 2+14 bit, che viene decrementato dopo ogni singolo trasferimento in modo da mantenere il conteggio del numero di caratteri che devono essere ancora trasferiti).

I due bit più significativi del TCR, identificano il tipo di operazione da svolgere. Rimangono pertanto 14 bit per il conteggio dei caratteri da trasferire. In questo modo é possibile trasferire fino a 16K caratteri per volta.

L'handshacking con ogni periferica avviene tramite due segnali:

- DRQ DMA Request (Attivo alto).
- DACK DMA Acknowledge (Attivo Basso).

Il DRQ, é pilotato dalla periferica, e viene da essa attivato quando deve richiedere uno scambio di dati. In pratica sostituisce il segnale d'interrupt che dovrebbe arrivare alla CPU.

Trasferito un carattere, la periferica può disattivare la richiesta o no. Nel secondo caso, il DMAC effettua il trasferimento in Burst Mode, ovvero non rilascia il bus alla CPU fra un trasferimento e l'altro, in maniera tale da poter scambiare in modo veloce un intero blocco di caratteri.

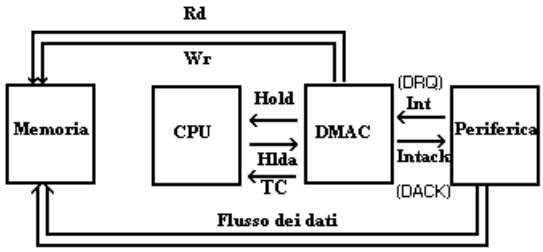
Il secondo segnale (DACK) è pilotato dal DMAC e funge da "STROBE", ovvero viene attivato dal DMAC per informare la periferica che il singolo carattere é stato trasferito, e che il DMA é pronto per il trasferimento del carattere successivo.

Dalla parte interna del sistema (DMAC/CPU) il DMAC dispone di 2 segnali più il segnale d'interrupt.

- HRQ Hold Request (Attivo Alto). Viene emesso dal DMAC e rappresenta il segnale di HOLD, ovvero la richiesta del bus.
- HLDA Hold Acknowledge (Attivo Alto). É generato dalla CPU e rappresenta l'accolta da parte di essa della richiesta del bus (HOLDACK).

Alla fine del trasferimento, ovvero quando il registro contatore é arrivato a zero, prima di rilasciare il bus alla CPU tramite la disattivazione del segnale di HLDA, viene attivato l'interrupt. Al momento del rilascio del bus alla CPU, la routine di gestione dell'interrupt, provvede a ricaricare i registri interni del DMAC con i nuovi valori iniziali per il prossimo trasferimento.

NOTA: É importante sottolineare il modo di collegamento fra la periferica e il sistema. Infatti la periferica é in questo caso collegata direttamente al Bus interno senza nessuna interposizione di buffer o altro tipo di interfaccia. Il DMAC, in questo modo, gestisce il trasferimento pilotando il bus indirizzi e il bus controlli, permettendo il trasferimento diretto del carattere dalla periferica alla memoria (o viceversa).



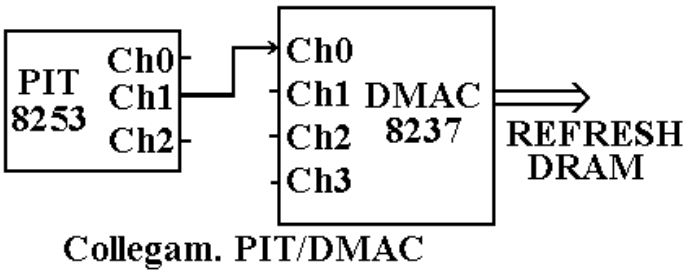
Questo sistema giustifica pienamente il nome del tipo di scheduling “Accesso Diretto in Memoria”.

La priorità di ogni canale é gestita direttamente dal DMAC. É possibile programmare il chip in modo da ottenere una PRIORITÀ RUOTANTE, ovvero assegnando la priorità minima al canale che é stato appena servito.

Tramite la tecnica di DMA é possibile effettuare trasferimenti di dati alla velocità di 1 MByte/sec.

- IL DMAC INTEL 8237 INSTALLATO SU IBM XT

Sui personal IBM sono istallati DMAC tipo Intel 8237, sostanzialmente simili ai DMAC Intel 8257. Gli indirizzi associati al DMAC sono tutti quelli che abilitano il relativo CS, ovvero vanno da 0000h a 000Fh.



Collegam. PIT/DMAC

NOTA: Nei sistemi più evoluti (80286) ci sono due DMAC che lavorano in parallelo. Gli indirizzi associati al secondo DMAC vanno da 00C0h a 00CFh.
Il canale "0" del DMAC é utilizzato per il REFRESH della memoria RAM dinamica, in combinazione con il canale "1" del Timer (Intel 8253 per l'XT o 8254 per i sistemi 286).

Un altro canale é utilizzato per la gestione dell'interfaccia con il Driver per i Floppy Disk.

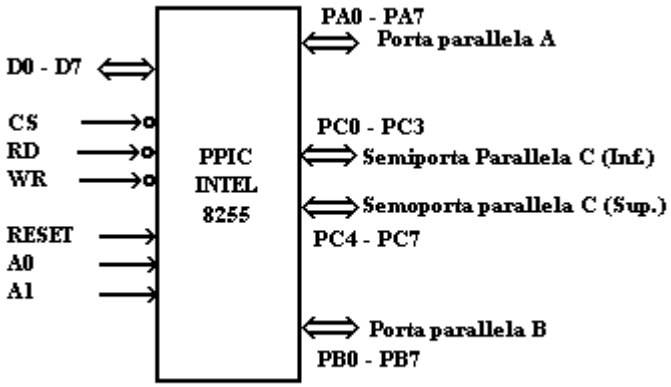
- INTERFACCIA PARALLELA

L' interfaccia parallela, permette il trasferimento di un carattere presente sul bus dati, alla periferica ad essa collegata, in modo parallelo, ovvero, tutti i bit del carattere vengono trasmessi contemporaneamente su 8 linee.

La CPU Z80 dispone, come interfaccia parallela del chip "PIO Z80", che ha come caratteristiche, la programmabilità e la compatibilità "TTL" (Livelli di tensione 0/0,8 V per il livello basso e 2,5/5 V per quello alto).

- PORTA PARALLELA INTEL 8255 (PPIC)

La porta parallela Intel 8255, ovvero la PPIC (Programmable Peripheral Interface Controller) é un chip di 40 pin, in logica TTL, capace di pilotare 3 porte di I/O ("A", "B" e "C") in modo parallelo. Dei 40 pin, 24 sono quelli relativi alle porte di I/O, le quali possono essere configurate in vari modi.



Mentre le porte "A" (PA0 % PA7) e "B" (PB0 % PB7) si considerano sempre ad 8 bit, la porta "C" può essere scomposta in due semiporte di 4 bit ciascuna (PC0 % PC3 :parte inferiore e PC4 % PC7

parte superiore). Quest'ultima suddivisione viene utilizzata perché la porta C, di solito, supporta i segnali di handshacking (rendendo così di fatto solo due le porte utilizzabili, ovvero "A" e "B"). Tanto é vero che le tre porte di I/O sono suddivise in due gruppi; quello della porta "A", costituito dalla porta "A" stessa e dalla semiporta "C" superiore e il gruppo "B", costituito dalla porta "B" e dalla semiporta "C" inferiore.

Alla PPIC sono associati 4 indirizzi consecutivi:

A1	A0	
0	0	Porta A
0	1	Porta B
1	0	Porta C
1	1	Registro di controllo

REGISTRO DI CONTROLLO (non sono ammesse operazioni di lettura su di esso, ma solo di scrittura). Il registro di controllo permette la programmazione dell'integrato

Il modo di funzionamento é stabilito configurando opportunamente il registro di controllo dell'interfaccia, il cui formato é il seguente:

B7 - Attivazione Modo ("0", "1" e "2") / porta C (1=Attivo / 0=Disattivo)
B6 - Attivazione Modo 2 (1=Attivo / 0=Disattivo).
B6=0 (Modo 2 disattivo): Sono disponibili tutti e due i gruppi di porte e per ognuno di essi si può definire un modo operativo indipendente
B6=1 (Modo 2 attivo):

Porte del gruppo "A"

B5 - 0=Modo 0, 1=Modo 1
B4 - Porta "A" 1=Input, 0=Output
B3 - Porta "C" sup. 1=Input, 0=Output

Porte del gruppo "B"

B2 - 0=Modo 0, 1=Modo 1
B1 - Porta "B" 1=Input, 0=Output
B0 - Porta "C" inf. 1=Input, 0=Output

		Gruppo porta A			Gruppo porta B		
		Porta A		Porta C Sup	Porta B		Porta C Inf.
PORTA C (0) 1=modo	MODO 2 Si/No	Modo Porta A	I/O Porta A	I/O Porta Cs	Modo Porta B	I/O Porta B	I/O Porta Ci
7	6	5	4	3	2	1	0

NB: 1=INPUT; 0=OUTPUT

L'interfaccia supporta 3 modi di funzionamento, ovvero il modo "0", "1" e "2".

Modi di Funzionamento:

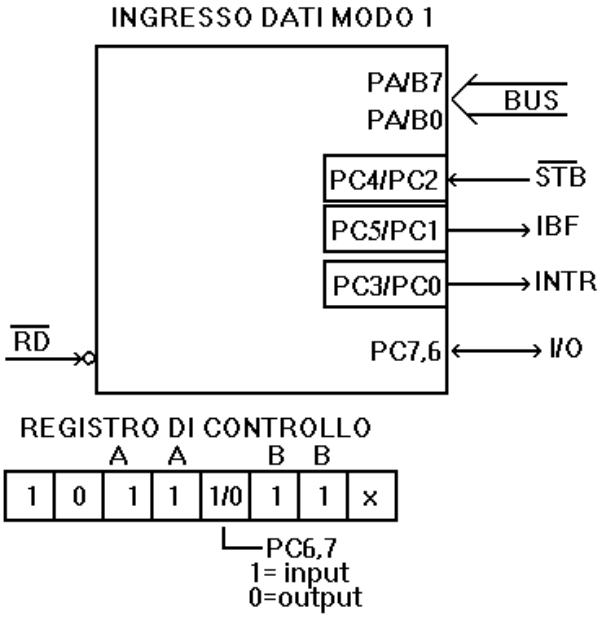
MODO 0: I/O Base (si può applicare al gruppo "A", al gruppo "B" e/o alle due semiporte "C" in modo indipendente).

Questo modo operativo prevede il normale funzionamento delle operazioni di I/O senza nessun segnale di handshacking. Qualsiasi porta può essere definita sia di ingresso che come uscita,

ma non contemporaneamente. Si hanno quindi 16 possibili configurazioni, una per ogni possibile configurazione delle 4 porte ("A", "B", "C superiore" e "C inferiore").

Il modo 0 viene attivato resettando il bit 2 (per la porta "B") e il bit 5 (per la porta "A") del registro di controllo. Per quanto riguarda la direzione dei flusso dati, questa dipende dai bit 4 (per la porta "A"), 3 (per la semi-porta "C" superiore), 1 (porta "B") e 0 (semi-porta "C" inferiore); bit=0 ⇒ Ingresso, bit=1 ⇒ uscita. I bit 7 e 6 sono posti a "1 0".

Es: 10^2 01 0 0 01 ⇒ porta "A" modo 0: ingresso, semiporta "C" superiore: uscita, porta "B" (modo 0): uscita e semiporta "C" inferiore: ingresso.



MODO 1: I/O Controllato da strobe (si può applicare al gruppo "A" e/o al gruppo "B" in modo indipendente).

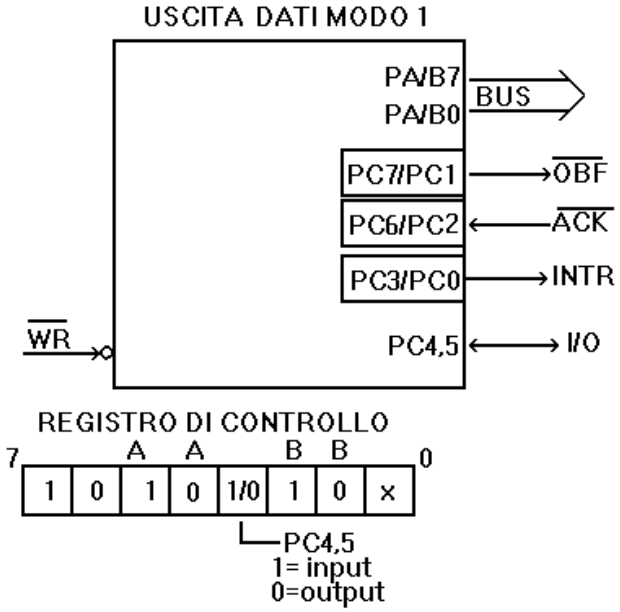
Questo modo si differenzia dal precedente per il fatto che le due semiporte "C" fungono da stato/controllo predefinito per i due gruppi, e sono gestite in modo automatico dal chip stesso. Le due porte possono configurarsi in ingresso o in uscita individualmente.

Bit della porta "C" per l'ingresso di un dato:

PC4/PC2 - "Strobe" (Ingresso di STB - attivo basso): Un impulso negativo in quest'ingresso produce la memorizzazione del dato all'ingresso della porta ("A" o "B") nel buffer.

PC5/PC1 - Buffer d'ingresso Pieno (Uscita IBF): Indica a livello alto che il dato é stato memorizzato nel buffer (risposta della porta al segnale di STB)

PC3/PC0 - INTR (uscita richiesta interrupt alla CPU): Viene generato un'interrupt (a 1) insieme al IBF quando é abilitato l'interrupt della porta.



Bit della porta "C" per l'uscita di un dato:

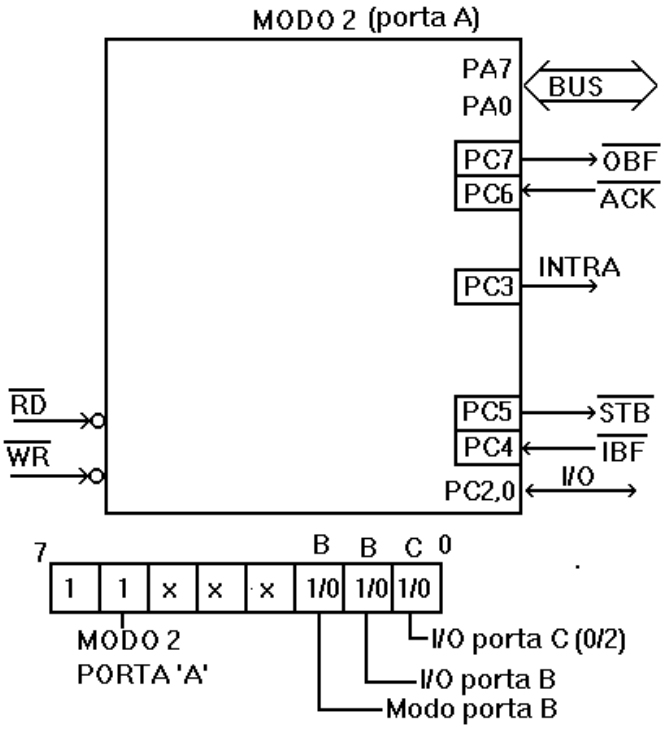
PC7/PC1 - Buffer uscita pieno (Uscita di OBF - attivo basso): A livello basso indica che il buffer della porta e pieno e che quindi é disponibile in uscita.

[2] MODO: no porta "C" e no modo "A"

PC6/PC2 - "Acknowledge" (Ingresso di ACK - attivo basso): Indica alla porta che il dato emesso é stato ricevuto quando é a livello basso.

PC3/PC0 - interrupt (uscita di INTR): a livello alto permette di lanciare un interupt alla CPU.

I due bit rimanenti della porta C (PC6 e PC7 per la configurazione di ingresso e PC4 e PC5 per la configurazione di uscita) possono essere configurati indipendentemente (I/O) tramite il bit 3 del registro di controllo.



MODO 2: Bus Bidirezionale (solo per la porta "A").
In questa configurazione si può operare in modo bidirezionale, ma questo modo é disponibile per la sola porta ("A"). 5 linee della porta "C" vengono utilizzate per l'handshacking. Tali linee sono:

- INT
- OBF e ACK per le operazioni di uscita.
- IBF e STB per le operazioni d'ingresso.

Le linee della porta B possono essere configurate con uno dei due modi ("0" o "1"), mentre le linee rimanenti della porta "C" (PC0, PC1 e PC2) possono essere configurate come ingresso o come uscita.

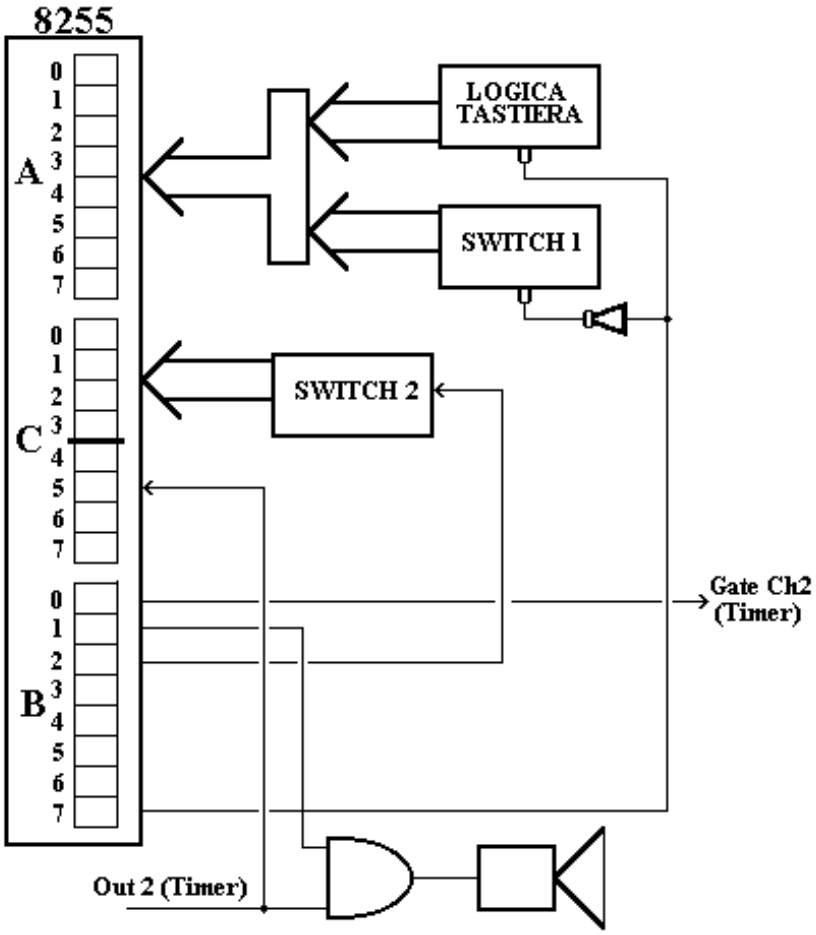
SET/RESET DI UN SIGOLO BIT

(solo per la porta "C"): La porta "C" ha la possibilità di essere pilotata bit a bit con una sola operazione di scrittura nel registro di controllo. Ciò é possibile portando a 0 il bit 7 del registro di controllo. In questa situazione tale registro accetta il seguente formato:
Bit 7 =0
Bit 0 = Valore da inserire nel singolo bit della porta "C"
Bit 1, bit 2 e bit 3 (MSB) = combinazione che forma il codice del bit da modificare nella porta "C" (Es bit 0= 1, bit 1 = 0, bit 2= 0 bit 3= 0 $\Rightarrow$ si modifica il bit 1 della porta "C" portandolo a 0). I bit 4, 5 e 6 sono ininfluenti
P.e. 0 000 011 1 per portare a 1 il bit 3 della porta "C"

La Funzione degli altri pin del chip (WR, RD, CS ecc.) é ovvia.

- UTILIZZO DELLA PORTA PARALLELA PPIC NEL PERSONAL IBM.

Nel PC IBM tipo XT (con 8088), l'interfaccia parallela sopra descritta é utilizzata in modo particolare per la GESTIONE DELLA TASTIERA, nonché per altri scopi, come la lettura degli interruttori di configurazione (SW1-1 % SW1-8 e SW2-1 % SW2-4), che servono per la configurazione del sistema (quantità di RAM, tipo display, n. unità dischi ecc.) e accessibili dall'esterno tramite commutazione manuale da parte dell'operatore. Inoltre, il bit 1 della porta "B" controlla il segnale dell'altoparlante proveniente dal canale 2 del timer (AND fra i due segnali).



La configurazione delle linee di indirizzo, associate alla interfaccia, ovvero quelle che attivano il relativo CS, sono tutte nulle escluso A6 e A5, pertanto il buffer della porta "A" é identificato dall'indirizzo 60h (A0=A1=0). Quello associato alla porta "B" (A0=1 e A1=0) é 61h, mentre quelli della porta "C" e del registro di controllo sono rispettivamente 62h e 63h.

Il sistema operativo inizializza il registro di controllo (Indirizzo 63h) con il valore 99h, corrispondente a:

b7=1 Modo Attivo
b6=0 Modo 2 non attivo

Gruppo porte "A"

b5=0 Modo 0
b4=1 Porta "A" Input
b3=1 Porta "C" Sup. Input

Gruppo porte "B"

b2=0 Modo 0
b1=0 Porta "B" Output
b0=1 porta "C" Inf. Input

La porta "A" (Indirizzo 60h) é quindi configurata come Input e contiene il codice di scansione della tastiera (se PB7=0) oppure la configurazione degli switch (se PB7=1).

La porta "B" (Indirizzo 61h) é configurata come output, e contiene i flag di abilitazione dei vari dispositivi (Altoparlante, RAM, motore cassetta ecc.).

Le due semiporte "C" (indirizzo 62h), sono configurate entrambe come input, e contengono lo stato delle periferiche.

Lo schema logico dei collegamenti della PPIC é integrato con la seguente tabella:

PORTA C SUPERIORE	BIT	INPUT
	4	DATA - IN (Registratore a cassette)
	5	OUT 2 (TIMER)
	6	ERRORE CONNNESSIONE ESPANSIONI DI MEMORIA
	7	ERRORE DI PARITÀ

PORTA B	BIT	OUTPUT
	3	ATTIVAZIONE MOTORE REGISTRATORE A CASSETTE
	4	ABILITAZIONE CHK CONTROLLO PARITÀ
	5	ABILITAZIONE CHK ERRORE ESPANSIONE MEMORIA
	6	ABILITAZIONE CLOCK TASTIERA

La logica della tastiera é collegata al chip 8259 (PIC), linea 1 (INTRQ1), pertanto, tutte le richieste d'interrupt da parte della porta parallela sfruttano questa linea d'interrupt. Ad essa il Sistema Operativo associa l'INT DI TIPO 09h (Pertanto, a questo tipo d'interrupt corrispondono tutte le operazioni di INGRESSO DA TASTIERA, ovvero ogni qualvolta viene premuto o rilasciato un tasto).

NOTA: Nei sistemi 286 (e successivi) al posto della porta 8255 c'è l'interfaccia parallela 8042, ma non cambiano sostanzialmente ne i collegamenti ne gli indirizzi.

Questo programma (beep_ppi.com), agendo sulla PPIC, genera un suono prodotto dall'altoparlante di frequenza e di durata fisse stabilite dai valori inseriti rispettivamente nei registri CX e DX.

Indirizzo	Codice macchina	CODICE ASSEMBLER		Commento
0100	50	PUSH	AX	Salva la situazione iniziale
0101	53	PUSH	BX	
0102	51	PUSH	CX	
0103	52	PUSH	DX	
0104	9C	PUSHF		
0105	BA0010	MOV	DX,1000	Numero periodi/ iterazioni
0108	E461	IN	AL,61	PONE linea altoparlante a "0"
010A	24FE	AND	AL,FE	
010C	0C02	OR	AL,02	
010E	E661	OUT	61,AL	
0110	B90010	MOV	CX,1000	ATTESA (1000)
0113	E2FE	LOOP	0113	pone linea altoparlante a "1"
0115	24FD	AND	AL,FD	
0117	E661	OUT	61,AL	
0119	B90030	MOV	CX,3000	ATTESA 3000
011C	E2FE	LOOP	011C	
011E	4A	DEC	DX	Controllo fine ciclo /uscita
011F	75EB	JNZ	010C	
0121	9D	POPF		ripristino situazione iniziale
0122	5A	POP	DX	
0123	59	POP	CX	
0124	5B	POP	BX	
0125	58	POP	AX	

- TIMER/CONTATORE PROGRAMMABILE

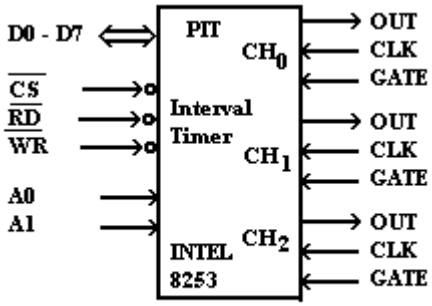
Questo dispositivo, detto anche "Generatore di Ritardo Programmabile", permette la gestione AUTOMATICA del CONTEGGIO DI EVENTI o il calcolo di intervalli di tempo (In pratica anche in questo caso funziona da conta-eventi dove ogni evento é un impulso di clock).

La programmazione consiste nell'inizializzazione di un registro interno che viene decrementato ad ogni evento (nel caso del Timer, ad ogni impulso di clock) a partire da un particolare istante di START.

Quando il contatore arriva a zero, il conteggio é finito e la CPU viene avvertita di questo fatto, o attivando un flag di stato che viene testato periodicamente (Polling), o generando un INTERRUPT.

Esempi di dispositivi di questo tipo sono: PIT INTEL 8253 e CTC Z80 (S.G.S.)

- PIT INTEL 8253



Il PIT ("Programmable Interval Timer") della Intel é composto da 3 canali indipendenti, capaci di svolgere operazioni di conteggio, tempificazione, divisione di frequenza ecc.

In effetti ha una logica più semplice di quella del CTC dello Z80, ma lo eguaglia nelle prestazioni, tanto é vero che dispone di 6 modi di funzionamento.

Infatti, ogni canale dispone di un registro a 16 bit, che può operare in singolo o in doppio byte per il conteggio in binario o in BCD. Il registro contatore di ogni canale può essere letto, anche durante il conteggio, dalla CPU.

Ogni canale dispone inoltre di un ingresso di "CLOCK" indipendente e di un ingresso di abilitazione ("GATE") che se disattivo inibisce l'ingresso del clock, nonché di un'uscita "OUT" che esplica lo stato del conteggio (è attivata una volta che il conteggio raggiunge un valore di set-point).

Il PIT dispone anche di un registro di controllo a 8 bit che predispone il chip secondo la programmazione selezionata.

Tale registro ha il seguente formato:

- b7, b6: Selezione canale (in fase di programmazione)

0 0 Canale 0
0 1 " 1
1 0 " 2

- b5, b4: 1 o 2 Byte per conteggio

0 0 Memorizzazione valore corrente
0 1 Lettura/Scrittura solo Byte alto del contatore
1 0 " / " " " basso " "
1 1 " / " sia Byte alto che basso in sequenza

- b3, b2, b1: Modo di funzionamento

0 0 0 Modo 0
0 0 1 " 1
0 1 0 " 2
0 1 1 " 3
1 0 0 " 4
1 0 1 " 5

- b0 Conteggio Binario o BCD (0=Binario)

L'accesso ad uno dei 3 registri contatori o al registro di comando viene selezionato tramite i piedini A0 e A1.

A1 A0

0	0	Indirizza il contatore del canale 0
0	1	" " " " " 1
1	0	" " " ' ' " 2
1	1	" il registro di controllo (solo scrittura)

Dato che i registri di controllo sono a 16 bit, la scrittura e la loro lettura nel caso di utilizzo a doppio Byte, avviene un byte per volta in sequenza.

MODI DI FUNZIONAMENTO:

Il modo di funzionamento del timer 8253, definisce come deve essere la forma d'onda d'uscita, che può essere un solo impulso (modi 4 e 5) oppure una sequenza ripetuta (modi 2 e 3). È possibile avere anche un solo livello alto in uscita alla fine del conteggio (modi 0 o 1).

Durante il conteggio è possibile leggere il valore contenuto nel registro contatore per vedere a che punto è arrivato il conteggio. Per questo è necessario scrivere prima nel registro di controllo. Il valore da inserire è:

Bit 6,7 numero del canale; Bit 4,5 a zero.

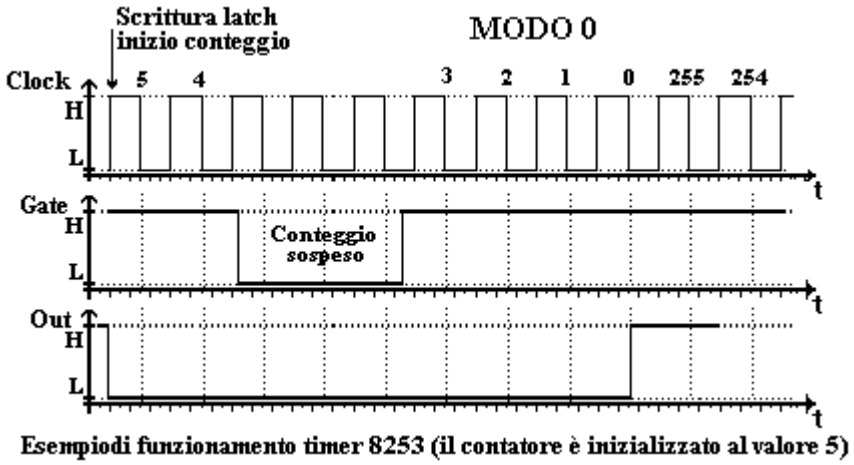
Esempio: per leggere il valore di conteggio del canale 0 del timer, si deve inserire 0000 0000 nel registro di controllo.

```
Indir.  Codice macchina  Istruzione assembler   Commento
0100    50               PUSH    AX            Salva i registri
0101    9C               PUSHF
0102    B000             MOV     AL,00         Programma il registro di controllo in modo
0104    E643             OUT     43,AL         da trasferire il valore di conteggio nel
                                                latch di impostazione
0106    E440             IN      AL,40         Legge il contenuto del registro
0108    88C3             MOV     BL,AL         contatore del canale 0
010A    E440             IN      AL,40
010C    88C7             MOV     BH,AL
010E    9D               POPF                  Ripristina i registri
010F    58               POP     AX
0110    CD20             INT     20
```

leggitim.com

MODO 0. Conto alla rovescia con inibizione facoltativa (uscita continua)

1. Appena il registro latch viene caricato, il suo contenuto è trasferito nel contatore.
2. Il segnale di OUT va basso, inizia il conteggio e il contatore si decrementa ad ogni impulso di CLOCK (sui fronti di discesa).
3. Se il segnale di GATE viene portato a livello basso il conteggio è temporaneamente

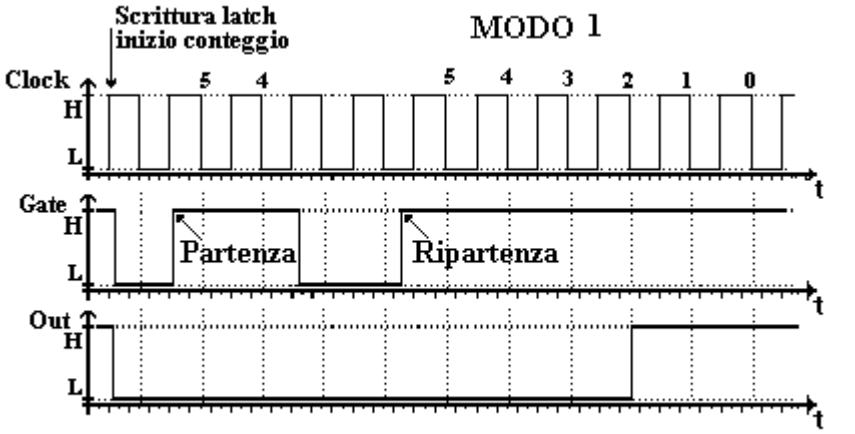


Esempiodi funzionamento timer 8253 (il contatore è inizializzato al valore 5)

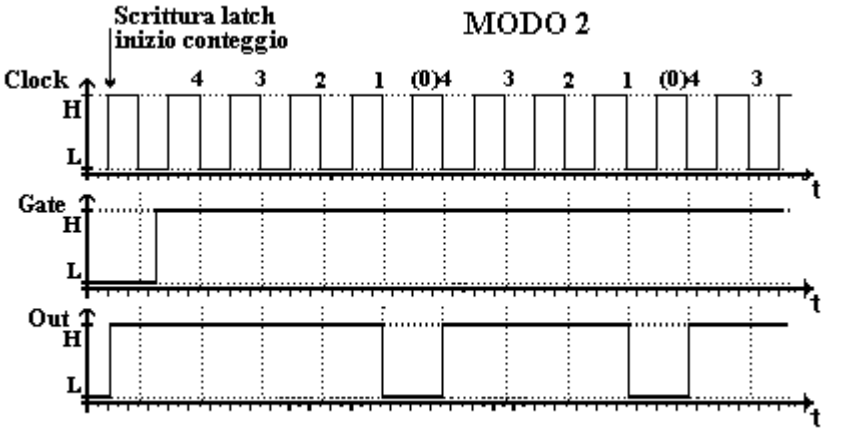
sospeso e riprenderà quando il GATE ritorna a livello alto.
4. Quando il contatore passa per lo zero (0000h o 00h) il segnale di OUT ritorna a livello alto e rimarrà in questo stato fino alla prossima definizione da parte dell'operatore.
5. Il conteggio continua oltre lo zero

MODO 1. Conteggio alla rovescia con ripartenza facoltativa (uscita continua)

1. Appena il registro latch viene caricato, il suo contenuto è trasferito nel contatore.
2. Il segnale di OUT va basso, ma il conteggio non inizia finché il segnale di GATE non è portato a livello alto.
3. Nel momento in cui il GATE va alto parte il conteggio e il contatore si decrementa ad ogni impulso di CLOCK (sui fronti di discesa).
4. Se il segnale di GATE viene riportato a livello basso il conteggio è temporaneamente sospeso e riprenderà quando il GATE ritorna a livello alto, ma il valore di partenza è quello originale definito dal contenuto del latch.
5. Quando il contatore arriva a zero (0000 o 00) il segnale di OUT ritorna a livello alto e rimarrà in questo stato fino alla prossima definizione da parte dell'operatore.
6. Il conteggio continua oltre lo zero



Esempiodi funzionamento timer 8253 (il contatore è inizializzato al valore 5)



Esempiodi funzionamento timer 8253 (il contatore è inizializzato al valore 4)

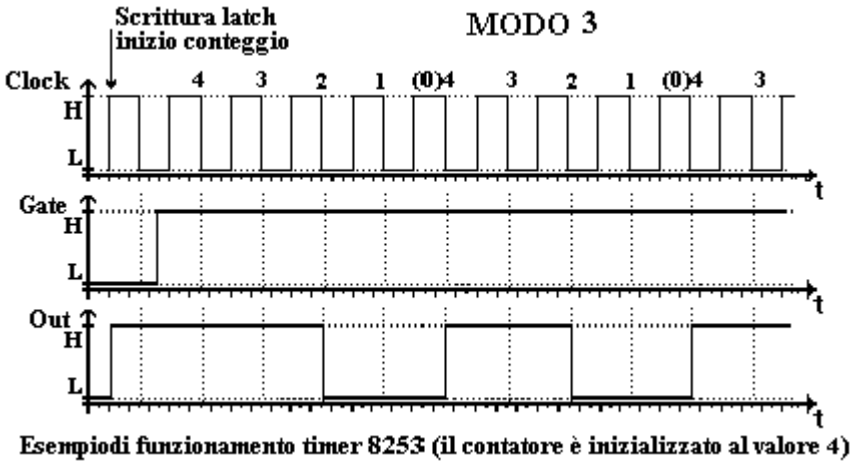
MODO 2. Generazione di un impulso d'uscita su "n" (uscita impulsiva)

1. Appena il registro latch viene caricato, il suo contenuto è trasferito nel contatore.
2. Il segnale di OUT va alto, inizia il conteggio e il contatore si decrementa ad ogni impulso di CLOCK (sui fronti di discesa).
3. Se il segnale di GATE viene portato a livello basso il conteggio è temporaneamente sospeso e riprenderà quando il GATE ritorna a livello alto.
4. Quando il contatore raggiunge "1" (0001 o 01) il segnale di OUT va basso per un periodo di CLOCK e ritornerà alto sul fronte di discesa dell'impulso successivo.
5. Intanto il registro contatore è reimpostato al valore iniziale predefinito ("n") disposto nel registro latch e l'operazione si ripete producendo un impulso negativo ogni "n" impulsi di CLOCK

MODO 3. Generazione di onda quadra (uscita ad impulsi)

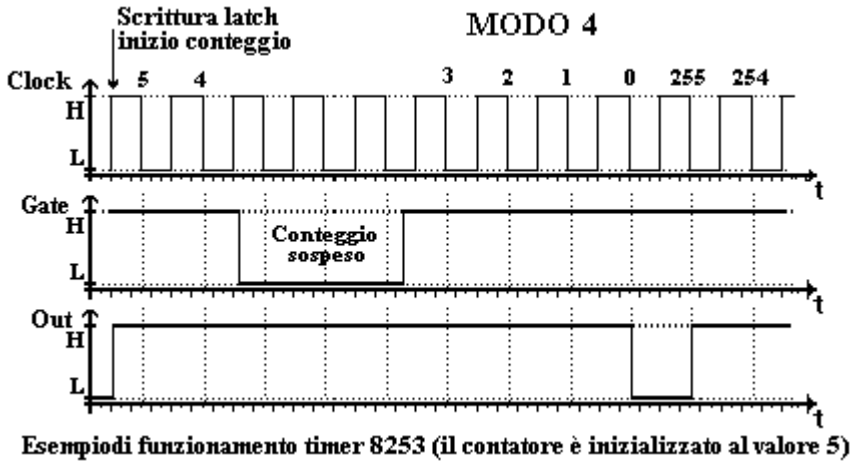
1. Appena il registro latch viene caricato, il suo contenuto è trasferito nel contatore.

2. Il segnale di OUT va alto, inizia il conteggio e il contatore si decrementa ad ogni impulso di CLOCK (sui fronti di discesa).
3. Se il segnale di GATE viene portato a livello basso il conteggio è temporaneamente sospeso e riprenderà quando il GATE ritorna a livello alto.
4. Quando il contatore raggiunge la metà del conteggio, (ovvero, in caso di caricamento nel lach di un valore dispari, la metà più uno) il segnale di OUT va basso per un periodo pari all'altra metà, ovvero al tempo impiegato al contatore per azzerarsi.
5. Tale situazione viene replicata con il progredire del conteggio, producendo un'onda quadra di frequenza pari a quella del CLOCK divisa pere il valore di inizio conteggio precaricato nel latch.



Esempiodi funzionamento timer 8253 (il contatore è inizializzato al valore 4)

MODO 4. Conteggio alla rovescia con inibizione facoltativa (uscita ad impulsi)



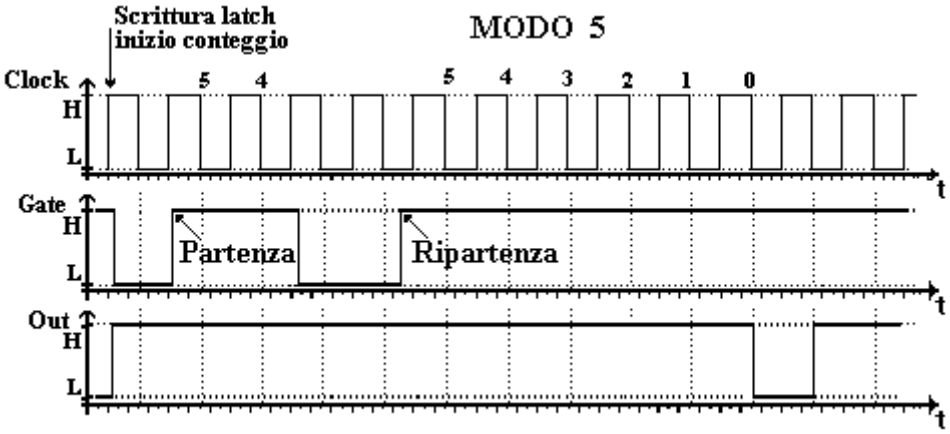
Esempiodi funzionamento timer 8253 (il contatore è inizializzato al valore 5)

1. Appena il registro latch viene caricato, il suo contenuto è trasferito nel contatore.
2. Il segnale di OUT va alto, inizia il conteggio e il contatore si decrementa ad ogni impulso di CLOCK (sui fronti di discesa).
3. Se il segnale di GATE viene portato a livello basso il conteggio è temporaneamente sospeso e riprenderà quando il GATE ritorna a livello alto.
4. Quando il contatore raggiunge lo zero (0000h o 00h) il segnale di OUT va basso per un periodo di CLOCK e ritornerà alto sul fronte di discesa dell'impulso successivo.
5. Il conteggio continua oltre lo zero, ma, a differenza del modo 2, l'impulso non viene più replicato.

MODO 5. Conteggio alla rovescia con ripartenza facoltativa (uscita ad impulsi)

1. Appena il registro latch viene caricato, il suo contenuto è trasferito nel contatore.
2. Il segnale di OUT va basso, ma il conteggio non inizia finché il segnale di GATE non è portato a livello alto.
3. Nel momento in cui il GATE va alto, parte il conteggio e il contatore si decrementa ad ogni impulso di CLOCK (sui fronti di discesa).

4. Se il segnale di GATE viene riportato a livello basso il conteggio è temporaneamente sospeso e riprenderà quando il GATE ritorna a livello alto, ma il valore di partenza è quello originale definito dal contenuto del latch.

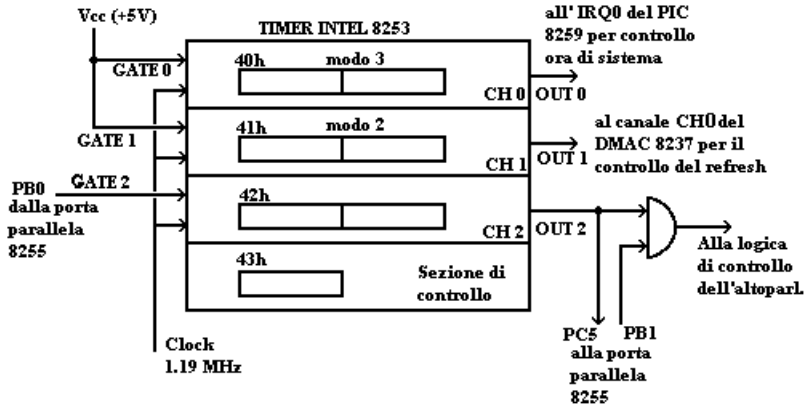


Esempiodi funzionamento timer 8253 (il contatore è inizializzato al valore 5)

5. Quando il contatore arriva a zero (0000h o 00h) il segnale di OUT ritorna a livello alto e rimarrà in questo stato fino alla prossima definizione da parte dell'operatore.
6. Il conteggio continua oltre lo zero

- IL PIT INTEL 8253 NEL PERSONAL IBM

NOTA : Gli ingressi di clock del timer sono collegati insieme e pilotati in parallelo dal clock di sistema decomposto a 1.19 MHz (periodo di 840 nSec).



Gli indirizzi associati ai registri interni del timer sono:

40h registro Contatore del canale 0
41h " " " " 1
42h " " " " 2
43h Registro di controllo

I tre canali del PIT vengono utilizzati nel seguente modo:

CANALE 0: Collegato alla linea INTRQ0 (INTRQ0=OUT0) del PIC. Inizializzando un conteggio con questo canale, viene generato un interrupt sulla linea 0, al quale é associato il tipo d'interrupt 08h. Il relativo ingresso GATE é sempre abilitato (GATE0=1)

Il sistema operativo lo inizializza con modo 3 al valore di conteggio max (0000h) generando quindi sull'uscita OUT0 un'onda quadra di 1,19 MHz/64K=18,2 Hz (T=55 msec). Qualora sia abilitata la linea d'interrupt 0 (INTRQ0), verrà generata una interruzione di tipo 08h ogni 55 msec. Questa interruzione é utilizzata dal sistema per tenere aggiornato in memoria l'orologio in tempo reale.

N.B. La memoria dell'orologio di sistema é un registro a 32 bit che viene incrementato ad ogni interruzione di tipo 08h (ovvero ogni 55 msec - 18.2 incrementi al sec.). Moltiplicando

55 msec x 2^{32} = circa 7,5 ANNI

otteniamo il tempo relativo al periodo previsto prima che si abbia una ripetizione di tempo.

Tale registro può essere impostato e letto utilizzando l'interrupt di tipo 1Ah del BIOS.

Nei PC più evoluti il registro é di 64 bit, e quindi può segnare date in un arco di tempo 2^{32} volte maggiore.

CANALE 1: L'uscita OUT1 é collegata al canale 0 (max priorità) del DMA, in modo che il segnale di TIME_OUT forzi il DMA ad effettuare una operazione di REFRESH in una zona di memoria. L'ingresso di GATE é sempre abilitato (GATE2=1). Esso é inizializzato dal sistema operativo col modo 2, al valore di conteggio di 18, producendo un impulso di uscita OUT1 ogni 18 impulsi di clock, ovvero ogni 15 microsec.
Quest'ultimo é quindi anche il periodo col quale viene effettuato il refresh della memoria.
L'operazione di refresh dura in totale 1.3 microsec., pertanto, per effettuare questa operazione, la CPU rimane inattiva (stato di wait) per circa il 9% del tempo (1.3/15).

```
0100   B0 74     MOV   AL , 74    Caricamento registro di controllo
0102   E6 43     OUT   43 , AL

0104   B0 XX     MOV   AL , XX    Impostazione parte BASSA
0106   E6 41     OUT   41 , AL    del Contatore

0108   B0 YY     MOV   AL , YY    Impostazione parte ALTA
010A   E6 41     MOV   41 , AL    del Contatore

010C   CD 20     INT   20         Ritorno al sistema operativo
```

Queste istruzioni permettono di programmare il canale "1" del timer in modo da modificare il tempo che intercorre fra un refresh e l'altro delle RAM dinamiche. Il tempo é stabilito dal valore **YY XX** posto nel registro contatore (veldram.com)

CANALE 2: Il canale 2 é collegato direttamente al circuito di potenza dell'altoparlante, e lo pilota generando su di esso varie forme d'onda di diversa frequenza. Il relativo ingresso di GATE é pilotato dal bit 1 della porta B dell'interfaccia parallela (PPIC 8255), che pertanto abilita o disabilita il canale. Può comunque essere utilizzato per scopi particolari.

Nei sistemi 286 il chip che funge da timer é l'8254 che é simile all'8253. Rimane pertanto invariato quanto detto finora.

Il programma seguente ("suono.com") genera un suono a onda quadra fino a che non si preme un tasto sulla tastiera. Viene programmato il canale 2 del timer.

Indirizzo	Codice macchina	CODICE ASSEMBLER	*COMMENTI*
0100	9C	PUSHF	*salva situazione iniziale*
0101	50	PUSH AX	
0102	53	PUSH BX	
0103	E4 61	IN AL,61	*attivazione timer*
0105	0C 03	OR AL,03	
0107	E6 61	OUT 61,AL	
0109	B0 B6	MOV AL,B6	*Impostazione timer (modo 3)*
010A	E6 43	OUT 43,AL	
010B	BB 00 00	MOV BX,0000	*Impostazione della frequenza*
010E	88 D8	MOV AL,BL	
0110	E6 42	OUT 42,AL	
0112	88 F8	MOV AL,BH	
0114	E6 42	OUT 42,AL	
0116	B4 00	MOV AH,00	*Impostaz. attesa tasto premuto*
0118	CD 16	INT 16	*(int bios tastiera)*
011A	E4 61	IN AL,61	*Disabilitazione timer*
011C	24 FC	AND AL,FC	
011E	E6 61	OUT 61,AL	
0120	5B	POP BX	*Ripristino situazione iniziale*
0121	58	POP AX	
0122	9D	POPF	
0133	CD 20	INT 20	

- TRASMISSIONE SERIALE

Le connessioni di tipo PARALLELO (una linea per ogni bit da trasferire), hanno grosse limitazioni, soprattutto in relazione alla distanza da coprire. Ciò é dovuto alla forte influenza che esse hanno ai disturbi esterni, e a quelli dovuti alla influenza mutua delle linee stesse. Il superamento di tali inconvenienti, comporterebbe consistenti schermature che renderebbero inaccessibili i costi. Per questo le linee parallele non vengono generalmente utilizzate per coprire distanze superiori ai 3 metri e solo in casi particolarissimi si arriva fino a 20 metri.

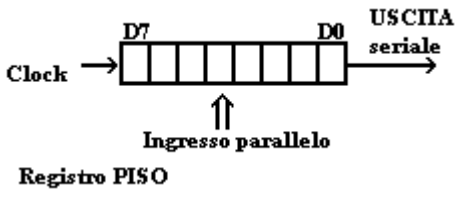
Per lunghe distanze, è molto più conveniente, l'utilizzo di sistemi di trasmissione di tipo SERIALE.

La TRASMISSIONE SERIALE consiste nel trasmettere le informazioni UN BIT ALLA VOLTA, in modo da utilizzare una linea a due fili elettrici (positivo e massa). Essa risulta molto meno sensibile ai disturbi soprattutto utilizzando particolari accorgimenti, come l'attorcigliamento detto “TWIST” dei due fili elettrici su se stessi. Si ha inoltre un notevole risparmio di materiale elettrico utilizzato e una minore complicazione circuitale.

Con questo sistema si riesce a coprire distanze fino a circa 20 Km (ma solo con velocità di trasmissione molto bassa).

- CONVERSIONE SERIALE/PARALLELO E VICEVERSA

Tale conversione viene realizzata tramite un registro a scorrimento di tipo PISO (nel caso di conversione Parallelo/Seriale) o SIPO (nel caso Seriale/Parallelo).

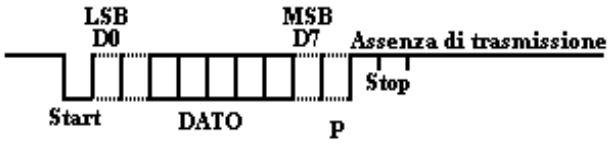


Allo scopo di rendere più sicura la trasmissione, ogni carattere è

inoltre corredato di alcuni bit di controllo secondo il seguente formato (detto ASINCRONO):

- 1 Bit di START (fisso a livello basso [0])
- 5/8 Bit di informazione formanti il carattere (D0 - D7). La trasmissione avviene inviando per primo il bit meno significativo. Per ultimo quello più significativo. Qualora il registro sia più lungo del dato ricevuto o da trasmettere, questo viene settato a destra nel senso che sono validi i bit meno significativi del registro.
- 1 Bit controllo di PARITÀ: É un bit ridondante che serve per verificare se la trasmissione é avvenuta correttamente, ovvero se il dato non si é degradato a causa dei disturbi sulla linea
- 1/2 Bit di STOP (fissi a livello alto [1])

In totale, il numero di bit per carattere inviati, va da 8 a 12.

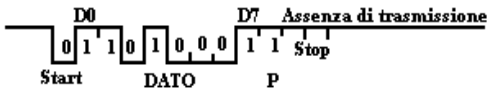


Sincronizzazione: l'utilizzo dei bit di start e di stop permette al ricevitore di agganciarsi al clock del trasmettitore in modo automatico. Per questo, la trasmissione é detta di tipo ASINCRONO.

- TRASMISSIONE SERIALE ASINCRONA

Il clock del ricevitore si predispone per la ricezione sul fronte di discesa del bit di start.

P.e. Si invia il byte 10001011b con codice ridondante a parità dispari (P=1).



Nella trasmissione seriale asincrona, il ricevitore si predispone per l'accettazione di un carattere rimanendo in attesa del fronte di discesa dell'impulso di start.

Infatti, in questo tipo di trasmissione, fra un dato e l'altro ci può essere un periodo di ASSENZA DI TRASMISSIONE che come minimo ha durata corrispondente ai bit di stop.

In assenza di trasmissione, il ricevitore si dispone in attesa del fronte di discesa dell'impulso di start, e questo periodo é di lunghezza arbitraria.

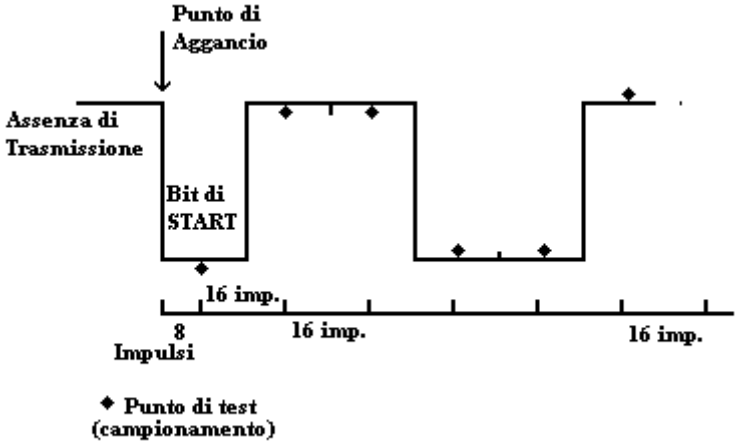
Il fronte di discesa dell'impulso di start definisce la "RIMESSA IN PASSO" (Sincronizzazione) del clock del ricevitore.

Affinché l'aggancio del clock del ricevitore al segnale, tempificato dal clock del trasmettitore, avvenga in maniera corretta, é necessario che le frequenze dei due segnali (Clock RX e segnale informativo) siano legate da opportune relazioni.

Generalmente la freq. del clock RX é un multiplo (Potenza del 2) della frequenza con la quale viene inviato il segnale informativo, definita dal clock del ricevitore.

P.e. freq. segnale: 9600 bit/sec (Baud-Rate)
" CK RX : 153.6 KHz (9600 x16)

La relazione fra le due frequenze é necessaria per effettuare correttamente il test di ogni bit del segnale. Questo deve avvenire sul punto equidistante dai due fronti successivi di ogni impulso cioè la zona centrale della forma d'onda, dove il segnale è più stabile.



Dato che il clock RX ha freq. 16 volte maggiore di quella del segnale, ovvero genera 16 impulsi fra un fronte e l'altro, l'aggancio corretto avviene contando 8 impulsi dal fronte di discesa dello start, e andando a testare il segnale ogni 16 impulsi successivi.

La garanzia di testare il bit in arrivo, nel punto di massima stabilità ovvero al centro della forma d'onda, é maggiore se la frequenza del Clock RX é alta. Per questo, il rapporto fra le frequenze è normalmente 1:64.
Dato che il formato del carattere é predefinito, il RX ha la possibilità di sapere quando é finita la trasmissione di un carattere e può così verificarne la congruenza. Il test dei bit del carattere di start, del bit di parità e successivamente i bit di stop procura una certa "garanzia" sull'affidabilità del dato. Qualora si siano verificate delle incongruenze, vuol dire che la lettura é avvenuta in modo sbagliato e che si é verificato un errore ("ERRORE DI QUADRO") eventualmente dovuto alla cattiva sincronizzazione.

Un altro possibile errore può essere dovuto al degradamento del dato durante la trasmissione a causa dei vari disturbi. Tale errore può comunque essere rivelato grazie al bit di parità ("ERRORE DI PARITÀ").

- CONTROLLO DI PARITÀ

E un codice ridondante (costituito da un solo bit) che viene aggiunto al carattere per la rivelazione degli errori di trasmissione.

La procedura di assegnazione del bit di parità consiste nel calcolo del numero di "1" presenti nel dato;

N. "1" PARI $\Rightarrow$ P=0
N. "1" DISPARI $\Rightarrow$ P=1

P.e.
Dato 1 0 0 0 1 0 1 1
D_7 D_0

Nel dato sono presenti 4 "1" $\Rightarrow$ P=0

NOTA: Si può osservare che il codice di parità non é altro che il risultato dell'operazione di XOR fra tutti i bit del dato.

P= 1 XOR 0 XOR 0 XOR 0 XOR 1 XOR 0 XOR 1 XOR 0 = 0

Il TX calcola la parità aggiungendola al dato e il RX fa altrettanto. Qualora il test abbia esito negativo si dice che si é verificato un ERRORE DI PARITÀ.

In alcuni casi è possibile adottare come risultato la negazione dell'operazione di XOR. Allora si dice che si è stabilito il controllo di parità DISPARI (il sistema ordinario è la parità PARI).

- TRASMISSIONE SERIALE SINCRONA

Si é visto che nella trasmissione di tipo asincrono é necessario trasmettere, oltre ai bit informativi, anche alcuni bit di controllo (Parità, start stop ecc.) e questo degrada le prestazioni della linea, in quanto diminuisce la velocità effettiva con cui si spediscono i dati.

P. Es. Con 9600 baud (Bit/sec) posso spedire caratteri alla velocità di:

9600 / 12 bit = 800 Car/sec
mentre in assenza dei caratteri di controllo:

9600 / 8 bit = 1200 Car/sec

I caratteri di controllo possono essere eliminati utilizzando un sistema di sincronismo DIRETTO, ovvero sincronizzando fra loro i clock di trasmissione e di ricezione ovvero utilizzando un tipo di trasmissione SINCRONO.

Questo sistema, può essere realizzato in due modi:

1) In modo molto banale, ovvero inviando al ricevitore il segnale di sincronismo su di una linea dedicata. Questo modo, pur apparendo semplice, in effetti può presentare alcuni problemi, per esempio dovuti al degradamento del segnale di sincronismo stesso, senza contare il fatto del maggiore costo della linea aggiuntiva.

2) É il sistema più utilizzato, e consiste nel riporre un messaggio di sincronismo direttamente nell'informazione da spedire. Se i clock da sincronizzare sono sufficientemente precisi, questo sistema é il migliore, in quanto permette di realizzare il sincronismo in modo semplice ed economico.

   P.e. Se la precisione del clock é dell'1% vuole dire che esso va fuori passo di un secondo ogni cento.

   Tale imprecisione provoca la precessione o la retrocessione del punto di test della forma d'onda, che col tempo si posizionerà su uno dei fronti o addirittura li scavalcherà trascurando un bit, con conseguente errore di lettura.

   Se il "tempo di bit" é di 100 msec il punto di test precederà o indietreggerà di 1 msec per ogni bit e dopo 50 bit si troverà sul fronte della forma d'onda.

   É necessario quindi rimettere in passo il clock ad intervalli regolari minori di 50 bit (Ovvero minori di 50 bit x 100 msec = 5 sec).

   Se ogni carattere é costituito da 8 bit, é necessario spedire un CARATTERE DI SINCRONISMO (SYNC codice ASCII = 16h) ogni 6 caratteri (50 bit /8) ovvero prima che vada fuori sincronismo.

   Con il secondo sistema, per inviare 6 caratteri occorre spedire effettivamente 56 bit (48 + 8 per SYNC), mentre con la trasmissione asincrona era necessario spedirne 72 (6 x 12 bit).

Dato che il clock é più preciso, si migliora notevolmente le prestazioni, rendendo la trasmissione SINCRONA più veloce di quell'asincrona.

- ESEMPIO DI REALIZZAZIONE DI TRASMISSIONE SERIALE ASINCRONA

Si suppone di dover realizzare un sistema di ricezione dati seriali da telescrivente.

Il bit 7 di una porta I/O di indirizzo 80h é collegato ad una telescrivente funzionante in modo seriale asincrono.

Il dato è messo in memoria all'indirizzo puntato da BP (Inizializzato al valore 60h).

La linea della TTY é normalmente alta (Assenza di comunicazione)

NOTA : la routine non effettua nessun controllo di Parità

```
            MOV     BP,60h

WAIT:       IN      AL [80h]        Attesa del
            RLA                     bit di
            JR      CL,WAIT         start

            CALL    DHALF           mezza Attesa

            MOV     DL,10000000b    Inizializzazione DL

LOOP:       CALL    DFULL           Attesa completa

            IN      AL,[80h]        Acquisizione
            RLA                     singolo bit
            RL      DL              in DL

            JNC     LOOP            fine dato ?

            MOV     AL,DL
            MOV     [BP],AL         Riposizione
            INC     BP              dato in
            JMP     WAIT            memoria

DHALF:      MOV     BL,8            routine
            JMP     DY16            di attesa

DFULL:      MOV     BL,16h

DY1:        MOV     CL,8Dh
DY16:       DEC     CL
            JNZ     DY1
            DEC     BL
            JNZ     DY16
```

RET

- UART / USART

La sigla USART sta ad indicare "Universal Syncronus/Asincronus Riceiver Transmitter", ovvero ricetrasmettitore universale sincrono o asincrono.

É un dispositivo che effettua in modo automatico tutte le funzioni legate alla trasmissione seriale (parallelizzazione, serializzazione, controllo errori ecc.).

In alcuni casi, la trasmissione può avvenire anche in modo sincrono, e in questo caso l'USART gestisce automaticamente il sincronismo in uno dei due modi visti precedentemente.

L'USART é un dispositivo programmabile, e può essere predisposto per l'attivazione del controllo di parità (sia pari che dispari), oppure nessuna. È possibile programmare la velocità di trasmissione (Baud-Rate), il numero di bit per carattere e il numero di bit di stop.

Ogni UART/USART é composto da 4 sezioni:

- Sezione di Trasmissione
- " " Ricezione
- Logica " Controllo
- Registro " Stato

- LOGICA DI CONTROLLO

É la sezione che permette la programmazione dell'USART, secondo i parametri:

- Parità (pari, dispari o nessuna)
- N. Bit Stop (1 o 2)
- Numero di bit per dato (da 5 a 8)
- Velocità di trasmissione (in modo quasi continuo da un valore minimo do 50 fino a 200.000 Baud)

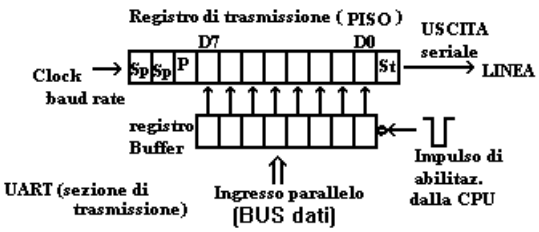
In alcuni dispositivi é possibile programmare il tipo di protocollo utilizzato relativamente al carattere di sincronismo inviato, nel caso di trasmissione sincrona.

NOTA: Fra i segnali di handshacking con la periferica, l'interfaccia dispone anche di un piedino per la comunicazione diretta del clock, che viene trasmesso tramite una linea aggiuntivo. In questo modo é possibile effettuare una trasmissione di tipo sincrono anche del primo tipo visto.

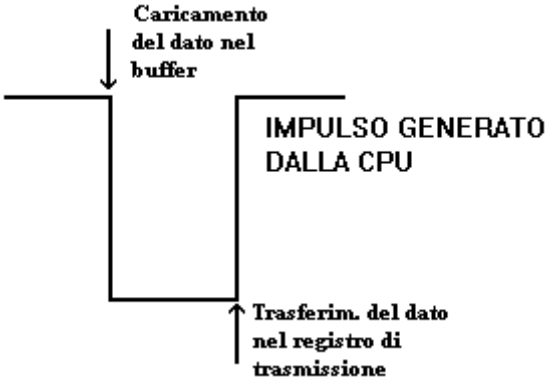
- SEZIONE DI TRASMISIONE

É costituita principalmente da due registri:

- Registro buffer che funge da interfaccia con la CPU e si affaccia sul bus dati.
- Registro di trasmissione, che é uno shift register. Esso provvede alla serializzazione del carattere.



Sp. = Bit di STOP
St. = " " START
P = " " Parità



La CPU indirizza il buffer e dispone il carattere (a 8 bit) da inviare su bus dati che viene presentato all'ingresso del buffer. In esso è scritto tramite l'impulso di caricamento (p.e. generato dal segnale di scrittura e abilitazione del microprocessore) nel suo fronte di discesa (Si presuppone che tale segnale sia attivo alto), mentre nel successivo fronte di salita il dato viene trasferito nel registro di trasmissione che lo serializza e lo spedisce sulla linea un bit per volta tramite scorrimento.

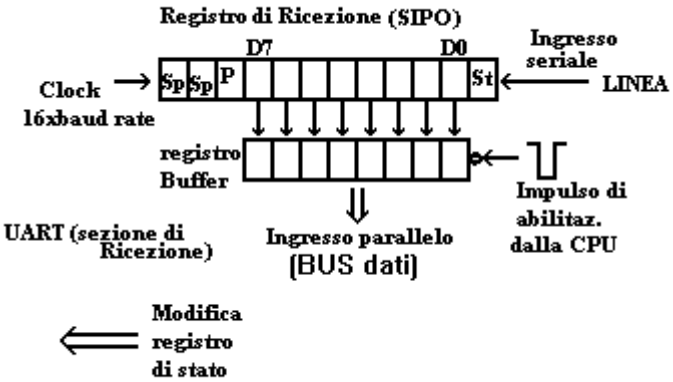
Prima di spedire il dato, questo é stato corredato dei bit di start e di stop, nonché del bit di parità, che vengono serializzati anch'essi e spediti sulla linea insieme al carattere.

Una volta che il buffer é stato vuotato e il registro di trasmissione ha inviato il carattere sulla linea, l'interfaccia attiva un bit nel registro di stato (DAC) per informare la CPU dell'avvenuta trasmissione, o può eventualmente generare automaticamente un INTERRUPT.

Dato che un carattere ASCII é generalmente costituito da 7 bit, il bit di parità é spesso rappresentato dall'MSB del carattere e non é necessario aggiungerlo automaticamente.

Quando il carattere é costituito da un numero di bit minore di 8, vengono caricati nel registro di trasmissione solo i bit meno significativi in modo da trasmettere solamente quelli prefissati.

- SEZIONE DI RICEZIONE



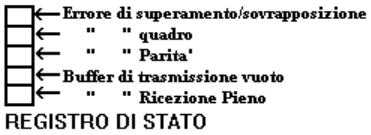
La sezione di ricezione é molto simile a quella di trasmissione. In assenza di trasmissione, il ricevitore rimane in attesa del fronte di discesa del bit di start. Il clock del RX si sincronizza sul fronte di discesa del segnale ricevuto e, a partire da esso conta 8 impulsi. Testa il bit di start (che deve essere trovato necessariamente a zero). Da questo momento in poi conta 16 impulsi e testa il bit successivo. Il procedimento continua per il numero di bit prefissato, caricando ogni bit testato nel registro di trasmissione.

Testato un numero di bit stabilito, il carattere, completo dei bit di controllo si trova nel registro di trasmissione e successivamente viene travasato nel registro buffer.

La presenza del dato nel registro buffer viene segnalata alla CPU (al solito) attivando un flag del registro di stato (DAC) o generando una interruzione.

Sp. = Bit di STOP
St. = " " START
P = " " PARITÀ

- REGISTRO DI STATO



REGISTRO DI STATO

Il registro di stato memorizza la condizione in cui si trova l'interfaccia

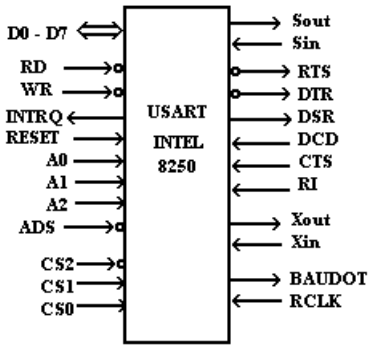
- SUPERAMENTO/SOVRAPPOSIZIONE. Il carattere appena ricevuto si é sovrapposto nel buffer di ricezione a quello precedente, prima che quest'ultimo sia stato memorizzato.
- ERRORE DI QUADRO. Non é stato trovato il bit di stop.
- PARITÀ. E stato riscontrato un errore di parità.
- Segnalazione di buffer VUOTO.
- Segnalazione di buffer PIENO.

- ESEMPI DI INTERFACCE SERIALI

I chip per l'interfacciamento seriale più utilizzati sono la SIO Z80 (Per la CPU Z80), USART 8251 Intel (per tutte le CPU Intel) e l'UART 8250

- UART INTEL 8250

É un integrato di 40 pin funzionante in logica TTL. Oltre alla logica ordinaria, include un generatore di Baudrate in grado di dividere per un valore variabile da 1 a 64K-1 il segnale di clock disposto fra i pin XOUT e XIN e genera un clock interno x16.



Dispone anche di tutte le funzioni di controllo del modem (DCD, DSR, DTR ecc).

É inoltre completamente programmabile per i parametri di controllo della linea seriale (N. bit dato, N. bit stop, parità, baud-rate ecc.) ed é capace di rivelare errori di tipo "Framming" o di parità.

Può gestire autonomamente delle interruzioni di vario tipo.

Tramite le linee di indirizzamento A0, A1 e A2 é possibile indirizzare i registri interni, che sono:

A2 A1 A0 Registro (Elenco registri dell'UART)

0 0 0 Registro BUFFER TRASMETTITORE, RICEVITORE e DIVISORE di baud-rate (LSByte) Ind. 3F8h.

Si può osservare che questo indirizzo identifica 3 registri contemporaneamente. In effetti i registri buffer per la trasmissione e la ricezione si differenziano per l'accesso, ovvero il primo é di scrittura (Output) e l'altro di lettura (Input). Per quanto riguarda la selezione fra il registro divisore di baud-rate (LSByte) e il buffer di trasmissione, che sono ambedue di output, questa avviene ponendo il bit7 del registro di controllo di linea a 0 per accedere al

buffer trasmettitore e a 1 per il registro divisore del baud-rate. Per accedere a questi registri bisogna quindi impostare prima il registro di controllo linea.

I due registri buffer vengono caricati con i dati (Caratteri) da trasferire.

Il registro divisore del baud-rate contiene il valore che va a dividere la frequenza del clock per ricavare il baud-rate. Infatti, Il segnale di clock, che viene inviato dall'esterno dall'ingresso di XIN e XOUT, può essere diviso per ottenere così la frequenza di trasmissione ovvero il baud-rate che viene ottenuto dalla seguente espressione:

$$\text{Baud-Rate} = \frac{\text{Freq. di Clock}}{16 \times \text{DIVISORE}}$$

Dato che il divisore é contenuto in due registri (MSByte e LSByte), é possibile dividere la frequenza del clock per multipli di 16 fino a 1M.

N.B. Il valore "16" deriva dalla scansione con la quale vengono testati e campionati i vari bit spediti o ricevuti. Infatti Il campionamento viene effettuato dopo 8 impulsi la prima volta (Bit di start) e successivamente ogni 16, dando luogo così al test di un bit ogni 16 impulsi.

0 0 1 Registro DIVISORE DI BAUD-RATE (MSByte) e registro di ABILITAZIONE DELLA INTERRUZIONE (Ind 3F9h).

Il registro divisore del baud-rate (LSByte) é già stato precedentemente trattato.

Per quanto riguarda il registro di abilitazione della interruzione, questo é identificato, oltre che dalla configurazione di A0, A1 e A2, dal bit7 del registro di controllo della linea. É un registro di uscita.

L'UART 8250 permette di gestire direttamente 4 tipi d'interrupt ad ognuno dei quali é associata una precisa priorità. In ordine di priorità, tali interrupt sono:

- Registro di stato della linea.
- Buffer ricevitore pieno (Dato ricevuto)
- Buffer trasmettitore vuoto.
- Stato del Modem.

Quando si verifica una delle condizioni sopra elencate, purché abilitata, il chip genera un segale di INTRQ sulla relativa uscita.

La funzione di questo registro é quella di mascherare o abilitare i quattro tipi d'interrupt secondo il formato:

b0 : Indica quando é a 1, la interruzione relativa alla condizione di Buffer ricevitore pieno (Dato ricevuto disponibile).

b1 : Indica quando é a 1, la interruzione relativa alla condizione di Buffer Trasmettitore vuoto (attesa di un altro dato da trasmettere)

b2 : Indica quando é a 1, la interruzione relativa alla condizione di stato della linea in ricezione.

b3 : Indica quando é a 1, la interruzione relativa alla condizione di stato del Modem.
b4 % b7 sono sempre a 0

0 1 0 Registro di IDENTIFICAZIONE DELLE INTERRUZIONI (Ind. 3FAh).

É un registro di lettura, la cui funzione é quella di gestire le 4 priorità di interruzione messe a disposizione dall'integrato. Il formato é il seguente:

b0 : Indica quando é a 0, la condizione di presenza di un interrupt pendente, ovvero che siamo ancora all'interno di una routine di servizio di un interrupt.

b1 : Questi due bit indicano l'interrupt pendente b2 : a priorità maggiore.
b3 % b7 sono sempre a 1

0 1 1Registro di CONTROLLO LINEA (Ind. 3FBh).

É un registro di scrittura e definisce i parametri relativi alla ricetrasmissione:

b1	b0	Definizione della lunghezza dato
0	0	5 Bit
0	1	6 Bit
1	0	7 Bit
1	1	8 Bit

b2 N. bit di stop. 0=1 bit di stop 1=2 bit di stop
b3 1=Abilitata la parità
b4 1=Parità pari (se abilitata)
b5 1=Impone a 0 il bit di parità (se abilitata)

b6 1= genera la condizione di BREAK, ovvero l'uscita seriale SOUT (trasmissione) é forzata a 0. L'uscita seriale é riabilitata riportando a 0 il b6. Serve per selezionare un terminale remoto. Infatti quando si attiva la linea possono essere generati in uscita tensioni spurie che potrebbero compromettere il collegamento. Questo bit non agisce sulla ricezione.

b7 É detto "DLAB", e serve per definire l'accesso al registro divisore (vedi sopra).

1 0 0 Registro di CONTROLLO DEL MODEM (Ind. 3FCh).

É indicato con la sigla "MCR", e controlla l'interfaccia con il modem.

b0 Controlla il DTR. In effetti effettua la negazione del valore del bit, infatti é forzato a 0 quando il bit assume il valore 1.

b1 Controlla la linea RTS e funziona in modo analogo al DTR.
b2 Controlla L'uscita ausiliaria "OUT1" che si comporta in modo analogo alle linee DTR e RTS.

NB: tale bit abilita l'UART a generare l'interrupt. Se é a 0, linterrupt é disabilitato
b3 Controlla L'uscita ausiliaria "OUT2" che si comporta in modo analogo alle linee DTR e RTS.
b4 Permette di utilizzare la retroazione locale della linea di Uscita (trasmissione) SOUT con quella di ingresso (Ricezione) SIN, in modo da ritornare in ingresso immediatamente il segnale appena trasmesso. viene utilizzato per scopi di test.
b5 % b7 permanentemente posti a 0.

1 0 1 Registro di STATO DELLA LINEA (Ind. 3FDh).

Esplicita lo stato della linea sia in trasmissione che in ricezione ed é un registro di lettura..

b0 É il valore del DSR.
b1 Segnala , quando é a 1, che si é verificato un errore di OVERRUN, ovvero di sovrapposizione dell'ultimo dato con quello precedente prima che questo fosse stato letto dalla CPU.
b2 Segnala, quando é a 1 che si é verificato un errore di parità.
b3 Segnala, quando é a 1 che si é riscontrato un errore di frame ("Framing Error").
b4 Segnala, quando é a 1 che é presente la condizione di break, ovvero che la linea di ricezione é rimasta a livello basso per un tempo maggiore di quello richiesto per la trasmissione di un carattere.
b5 Segnala, quando é a 1 che il buffer di trasmissione é vuoto. Viene generato l'interrupt, quando abilitato.
b6 Segnala quando é a 1 che il registro di shift del trasmettitore é vuoto.
b7 Permanentemente a 0.

1 1 0 Registro di STATO DEL MODEM (Ind. 3FEh).

É denominato MSR, e permette di acquisire lo stato corrente del modem. É un registro di lettura.

b0 Identifica la linea DCTS.
b1 Identifica la linea DSR.
b2 Individua la linea RI
b3 Identifica la linea DCD
b4 É il complemento della linea CTS.
b5 É il complemento della linea DSR
b6 É il complemento del segnale RI
b7 É il complemento del segnale DCD.

1 1 1 Non Usato

A parte i segnali di tipo ordinario, come i CS, WR, RD ecc, nonché dei segnali per il controllo del modem (DTR, DSR ecc), già trattati, il significato degli altri segnali é:

SOUT e SIN: Linee di uscita (Trasmissione) e ingresso (Ricezione) seriali rispettivamente.
ADS: (Address Strobe). É l'impulso per il caricamento del dato sul bus dati nel registro indirizzato dell'UART.
RCLK: Ad esso si applica il clock esterno che verrà poi diviso per il contenuto del registro divisore del baud-rate allo scopo di definire il baud-rate stesso.

OUT1 e OUT2: Sono uscite ausiliarie controllabili dal contenuto di due flag del registro di controllo della linea.
CSOUT: (Chip select out). Indica lo stato di selezione del chip stabilito dalla configurazione dei CS di ingresso.
DDIS: (Driver Disable). É a 1 tutte le volte che la CPU effettua una lettura di un registro interno dell'UART.
XIN e XOUT : Connettono il temporizzatore principale all'UART.

Per maggiori informazioni vedere "Sistemi ed Automazione" di A. Memo - vol 2 CEDAM pg.468.

- L'UART NEL PERSONAL COMPUTER IBM.

La scheda seriale del personal IBM si avvale dell'UART 8250 già visto sopra. Per la prima porta seriale (COM1), la linea di INTRQ del PIC interessata é la 4 (INTRQ4), e gli indirizzi dei registri interni sono i seguenti:

- Reg. Buffer (Trasmisione e ricezione) e registro divisore di baud-rate (LSByte). 3F8h
- Reg. Divisore di baud-rate (MSByte) e registro di abilitazione interruzione. 3F9h
- Reg. Identificazione delle interuzioni.3FAh
- Reg. controllo linea. 3FBh
- Reg. controllo Modem. 3FCh
- Reg. Stato Linea. 3FDh
- Reg. Stato Modem. 3FEh

La linea di INT della porta seriale 8250 é collegata direttamente alla linea INTRQ4 del PIC 8259 il quale fornisce alla CPU il codice di tipo 0Ch.

La porta seriale 8250 inoltre può essere gestita indirettamente dalla routine di servizio dell'interrupt 14h con BX=0 per la prima porta seriale (COM1) e BX=1 per la seconda porta seriale (COM2).

Per funzionare correttamente, la porta seriale deve essere inizializzata, ovvero devono essere predisposti opportunamente i registri interni, 5 dei quali poi non vengono più modificati.

L'inizializzazione del chip inizia con l'impostazione del valore del divisore del baud-rate, e per fare questo é necessario porre a 1 l'MSB (DLAB) del registro di controllo linea.

Es. Clock =1.8 MHz baud-rate=1200 Bit/sec.

DIVISORE=Clock/(16*Baud-rate)=1.8MHz/(16*1200)=96=60h

I due registri divisore di baud-rate devono essere caricati col valore 0060h.

```
MOV     DX,3FBh   Dispone l'MSB del registro di
MOV     AL,80h    controllo linea a 1 (DLAB).
OUT     DX,AL

MOV     DX,3F8h   Carica la parte bassa nel
MOV     AL,60h    registro divisore.
OUT     DX,AL

MOV     DX,3F9h   Carica la parte alta del
```

```
MOV     AL,0        registro divisore.
OUT     DX,AL
```

Deve essere successivamente inizializzato il registro di controllo di linea con i parametri opportuni.

Es. Linea a 1200 baud, con 7 bit di dato, 1 bit di stop, parità pari:

```
MOV     DX,3FBh
MOV     AL,1Ah
OUT     DX,AL
```

Deve essere successivamente inizializzato il registro di controllo del modem col valore 03h.

```
MOV   DX,3FCh
MOV   AL,03h
OUT   DX,AL
```

Nel caso si voglia gestire le interruzioni, é necessario allora predisporre anche il registro di abilitazione delle interruzioni. P. es. interruzioni tutte abilitate

```
MOV   DX,3F9h
MOV   AL,0Fh
OUT   DX,AL
```

Per trasmettere o ricevere un dato é necessario leggere o scrivere nei buffers di ricezione o trasmissione. É necessario conoscere lo stato dei registri allo scopo di non scrivere nel buffer quando é pieno o prelevare un dato quando é vuoto. Per evitare questo, quando non sono attivi gli interrupt, si può leggere il registro dello stato di linea prima di effettuare l'operazione (in particolare si testa il bit 5 e il bit 0).

Il buffer non può essere riempito finché il bit 5 del registro di stato di linea non é a 1. Il bit passerà automaticamente a 0 non appena si riempie il buffer.

```
        MOV  DX,3FDh     Attesa che il bit 5 del registro
LOOP:   IN   AL,DX       di stato linea passi a 1.
        TEST AL,20h
        JNZ  LOOP

        MOV  DX,3F8h     Carica il carattere contenuto nel
        IN   AH,DX       registro AL nel buffer di  trasmissione.
```

Il buffer non può essere letto finché il bit 0 del registro di stato di linea non é a 1. Il bit passerà automaticamente a 0 non appena si legge il buffer.

```
        MOV     DX,3FDh  Attesa che il bit 0 del registro
LOOP:   IN      AL,DX    di stato linea passi a 1.
        TEST    AL,01h
        JNZ     LOOP

        MOV     DX,3F8h  Carica il carattere appena ricevuto
        IN      AH,DX    nel registro AL.
```

- NOTE PER LE INTERFACCE SERIALI SUPPLEMENTARI

É possibile installare altre interfacce seriali che utilizzano i parametri della tabella:

Interfaccia	Indirizzo (HEX)	Indirizzo (BIN) A8, A7, ... , A0	Linea Interrupt	Cod. Int.
COM 1	3F8h	1 1 1 1 1 1 0 0 0	INTRQ4	0Ch
COM 2	2F8h	0 1	INTRQ3	0Bh
COM 3	3E8h	1 0	INTRQ4	0Ch
COM 4	2E8h	0 0	INTRQ3	0Bh

1 Dire quali sono i dispositivi ausiliari più comuni e quali sono le loro funzioni

DA

2 Spiegare il funzionamento del PIC e dell'interrupt in genere

DA

3 Come è organizzato logicamente un PIC (schema a blocchi)

DA

4 Con quale tipo di logica possono essere messi in "cascata" più PIC?

DA

5 Quali sono i principali registri interni del PIC ?

DA

6 Con quali criteri può essere programmato il PIC 8259?

DA

7 Attraverso quali elementi si può programmare il PIC 8259?

DA

8 Quali sono le periferiche servite dal PIC 8259 nel PC?

DA

9 In che cosa consiste il DMA e come viene realizzato

DA

10 Quali sono i vantaggi apportati dalla presenza di un DMAC e perché?

DA

11 Come è organizzato internamente un DMAC?

DA

12 Qual è la sequenza ovvero il protocollo col quale il DMAC si interfaccia con la CPU?

DA

13 Quali sono le caratteristiche del DMAC 8257?

DA

14 Come viene utilizzato il DMAC 8257 nel PC?

DA

15 Come è organizzata internamente la porta parallela PPIC 8255?

DA

16 Quanti e quali modi di funzionamento sono disponibili nella porta parallela PPIC 8255?

DA

17 Come viene utilizzata la porta parallela PPIC 8255 nel PC?

DA

18 Spiegare qual è il principio di funzionamento del PIT (interval timer)

DA

19 Qual è lo schema logico interno del PIT 8253?

DA

20 Come è possibile programmare il PIT 8253 e quanti sono i modi di funzionamento possibili?

DA

21 Come viene utilizzato il PIT 8253 nel PC?

DA

22 Spiegare da che cosa è caratterizzata la trasmissione seriale e quali altri tipi di collegamento sono disponibili.

DA

23 Spiegare quali sono i vantaggi della trasmissione seriale rispetto a quella parallela.

DA

24 Spiegare com'è possibile realizzare la trasmissione seriale

DA

25 Spiegare in che cosa consiste la trasmissione seriale ASINCRONA e qual è il formato

DA

26 Spiegare in che cosa consiste il bit di parità ed il relativo controllo. Spiegare cosa esso ha a che fare con la trasmissione seriale.

DA

27 Spiegare in che cosa consiste la trasmissione seriale SINCRONA e quali sono i vantaggi rispetto a quella asincrona

DA

28 Disegnare lo schema logico di un UART e le caratteristiche di funzionamento.

DA

29 Da quante e quali sezioni è costituito un UART?

DA

30 Spiegare il funzionamento della sezione di trasmissione di un UART

DA

31 Spiegare il funzionamento della sezione di ricezione di un UART

DA

32 Spiegare quali sono le informazioni memorizzate nel registro di stato di un UART

DA

33 Spiegare quali sono le caratteristiche di programmabilità di un UART ovvero quali sono i parametri che è possibile impostare dall'esterno.

DA

34 Descrivere quali sono i registri principali i un UART

DA

35 Come viene utilizzato l'UART nel PC?

DA