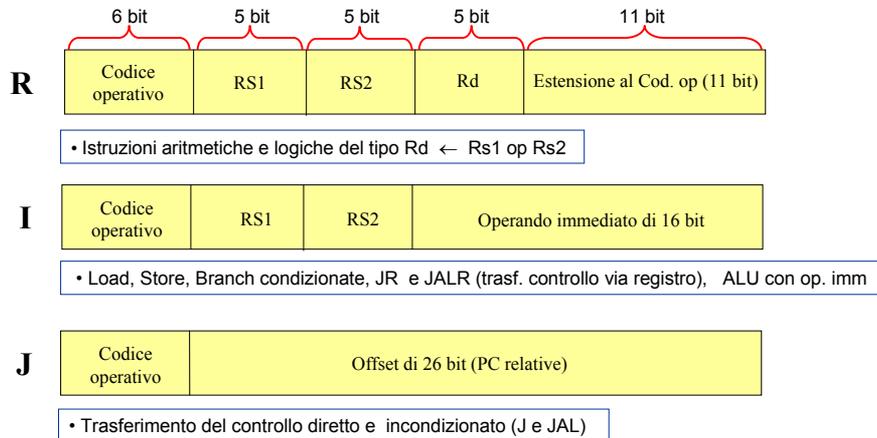


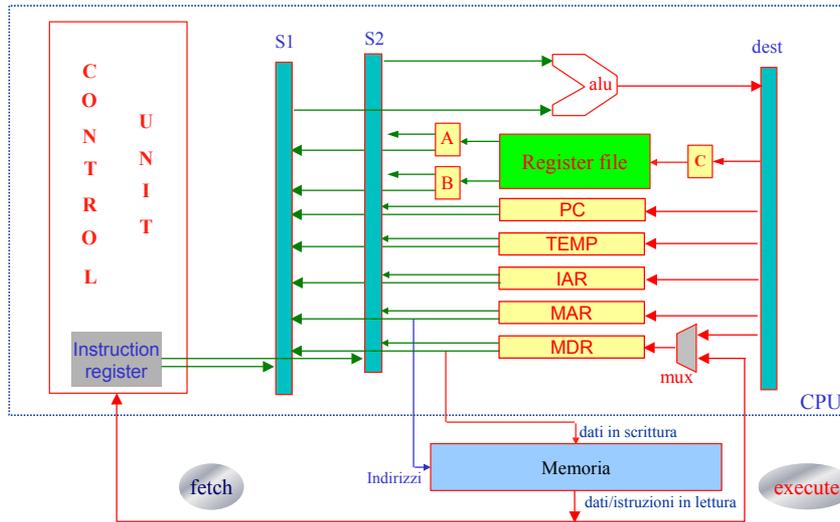
Formato tipico delle istruzioni nelle architetture R-R (Es. DLX)



Il set di istruzioni del DLX

- Le principali istruzioni aritmetiche e logiche
 - Istruzioni logiche anche con op. immediato: **AND, ANDI, OR, ORI, XOR, XORI**
 - Istruzioni aritmetiche: **ADD, ADDI, SUB, SUBI, MULT, DIV**
 - Istruzioni di traslazione logica (a destra anche aritmetica): **SLL(I), SRL(I), SRA(I)**
 - Istruzioni di SET CONDITION: **Scn, Scnl**, con **cn = EQ, NE, LT, GT, LE, GE**
- Le principali istruzioni di trasferimento dati
 - Load byte, Load Halfword, Load Word (**LB, LH, LW**)
 - Store byte, Store Halfword, Store Word (**SB, SH, SW**)
 - Load/Store Floating Point in singola o doppia precisione (**LF/SF e LD/SD**)
 - Copia un dato da un GPR a un FPR e viceversa (**MOVI2FP e MOVFP2I**)
- Le principali istruzioni di trasferimento del controllo
 - Istruzioni di salto condizionato (PC+4 relative): **BNEZ, BEQZ**
 - Istruzioni di salto incondizionato diretto e indiretto(PC+4 relative): **J, JR**
 - Istruzioni di chiamata a procedura (Jump and Link, l'indirizzo di ritorno viene automaticamente salvato in R31): **JAL, JALR**
 - Istruzione di ritorno dalla procedura di servizio delle interruzioni: **RFE**

Struttura del DLX (esecuzione sequenziale)



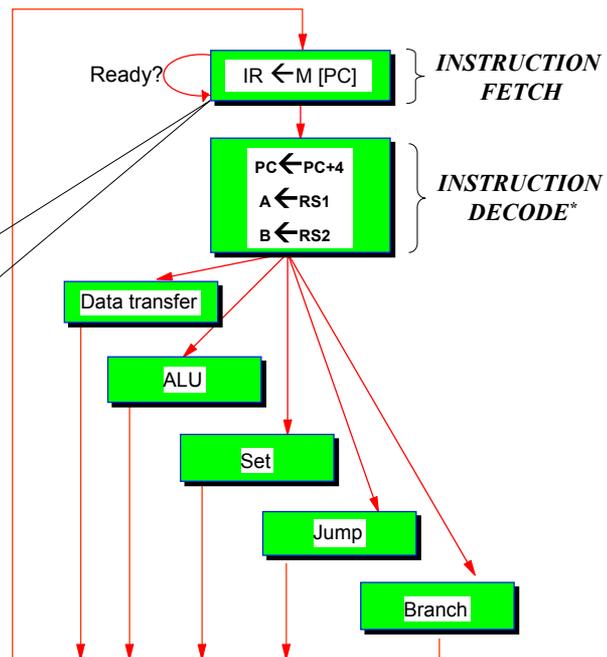
Parallelismo dell'architettura: 32 bit
(bus, alu e registri hanno parallelismo 32)

I segnali di controllo non sono indicati!

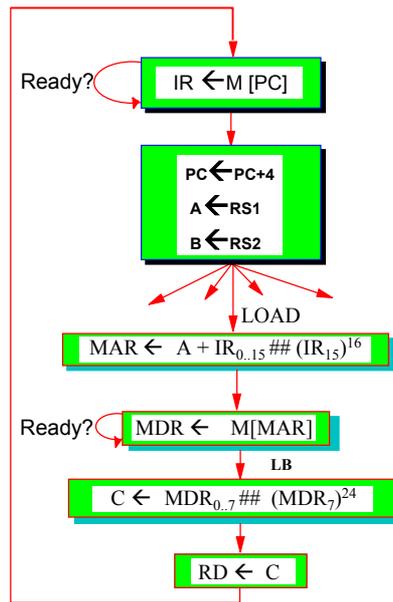
Il diagramma degli stati del controller (2)

Si modifica il DATAPATH in maniera da poter indirizzare la memoria da PC. Tutte le istruzioni impiegano un clock in meno per essere eseguite !

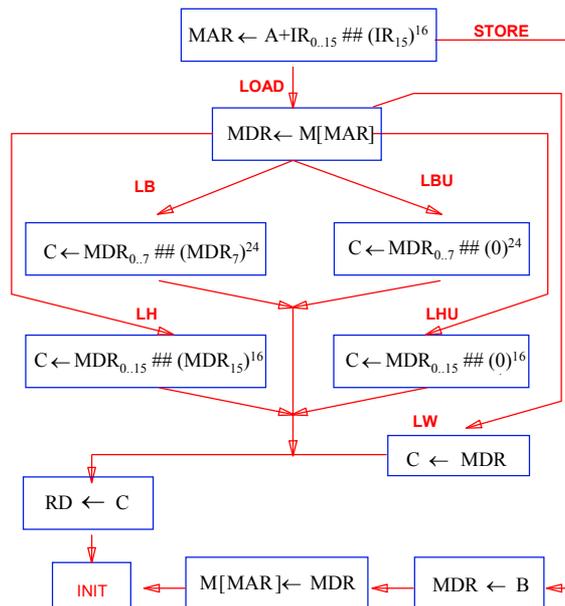
Quanti clock sono necessari per eseguire la fase di fetch?



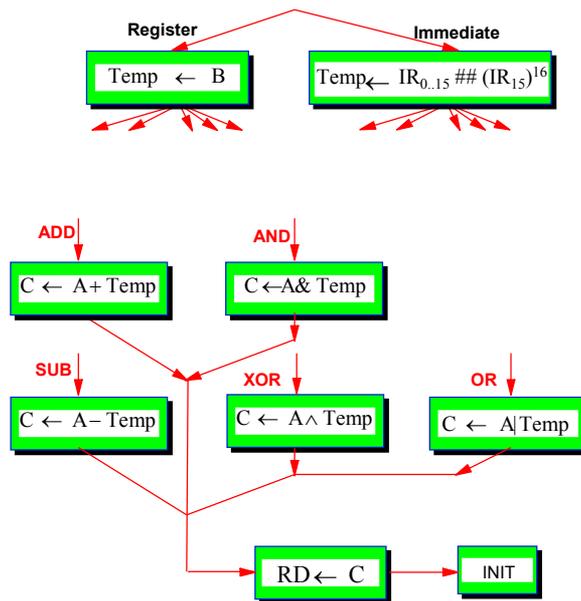
Controllo per l'istruzione LB (Load Byte)



Controllo per le istruzioni di Data Transfer



Controllo per le istruzioni ALU

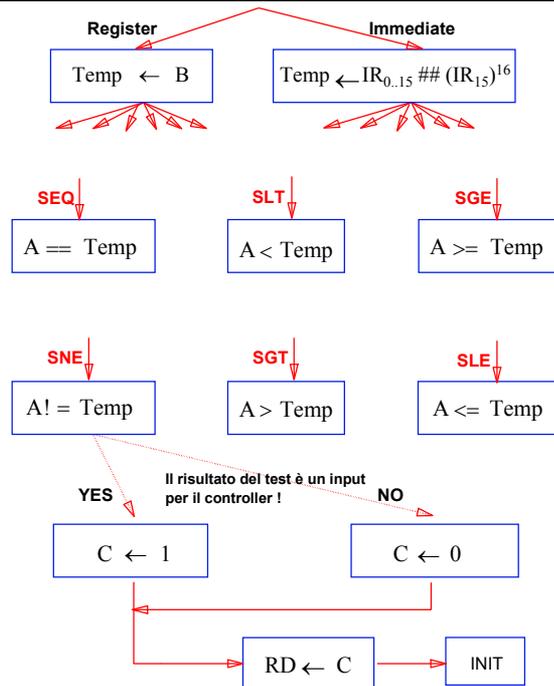


materiale utile per esercizi

7

Controllo per le istruzioni di SET (confronto)

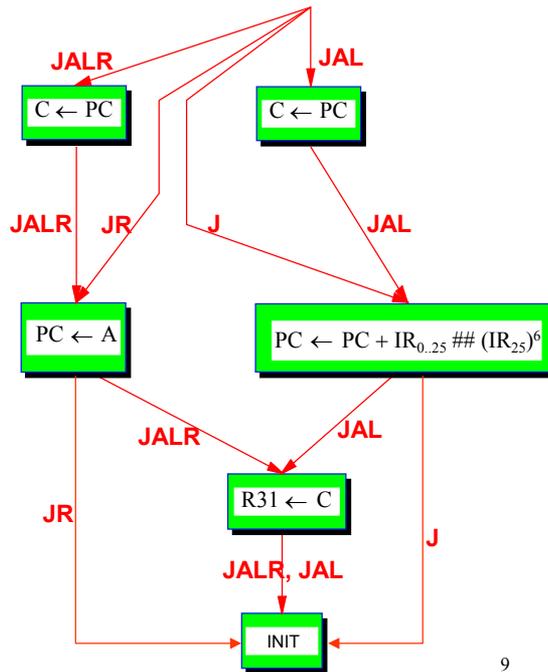
SET condition IN REGISTER:
Scnd Rd, Rs1, Rs2
 (dove *cond*: EQ, LT, NEQ, ...)



materiale utile per esercizi

8

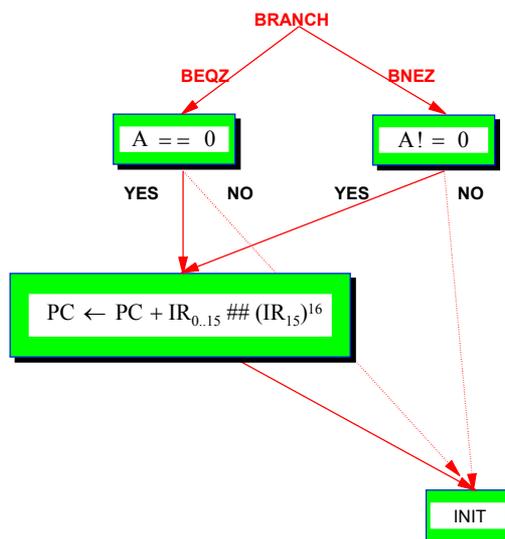
Controllo per le istruzioni di Salto



materiale utile per esercizi

9

Controllo per le istruzioni di Branch



materiale utile per esercizi

10

Bilanciamento di una Pipeline

- **Caso Ideale**

$$T_P = T_{pi} = \frac{T_A}{K} \quad \Rightarrow \quad \text{Pipeline perfettamente bilanciata} \quad \Rightarrow \quad \text{Speedup} = K$$

- **Caso Reale**

$$T_P = \max(T_{P1}, T_{P2}, \dots, T_{PK}) \quad \Rightarrow \quad \text{Bilanciamento imperfetto} \quad \Rightarrow \quad \text{Speedup} < K$$

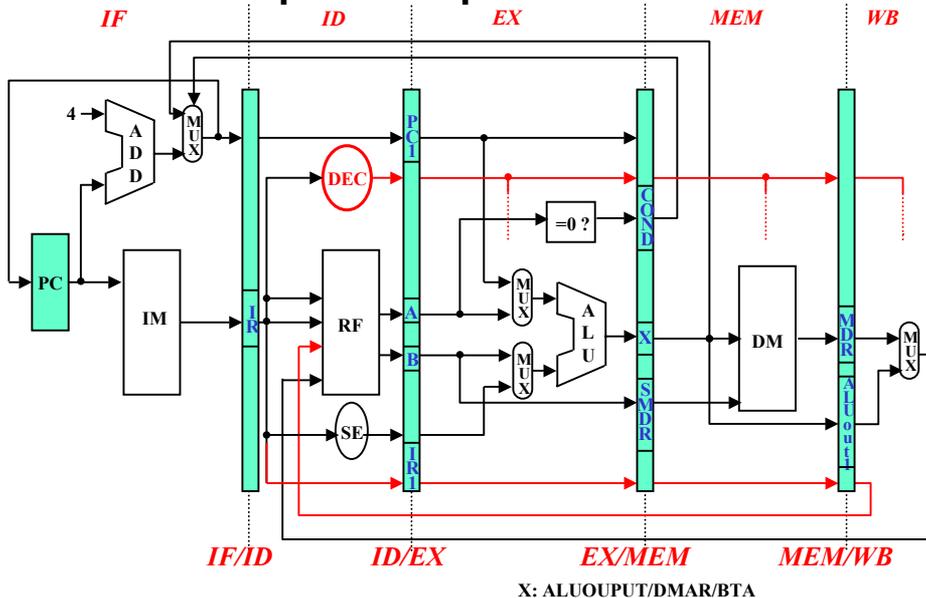
- **Un Esempio**

$T_A = 20 t$ (t : unità di tempo, es.: cicli di clock)

$T_{P1} = 5 t, T_{P2} = 5 t, T_{P3} = 6 t, T_{P4} = 4 t \quad \Rightarrow \quad T_P = 6 t$ (è il più lento)

$$\Rightarrow \text{Speedup}(p) = \frac{T_A}{T_P} = \frac{20 t}{6 t} \cong 3.33 (< 4)$$

Datapath in Pipeline del DLX



Esecuzione in pipeline di un'istruzione "ALU"

IF	$IR \leftarrow M[PC]; \quad PC \leftarrow PC+4$
ID	$A \leftarrow RS1; B \leftarrow RS2; PC1 \leftarrow PC;$ $IR1 \leftarrow IR$ (per gli stadi successivi)
EX	$ALUOUTPUT \leftarrow A \text{ op } B$ oppure $ALUOUTPUT \leftarrow A \text{ op } IR_{0..15} \text{ ## } (IR_{15})^{16}$
MEM	$ALUout1 \leftarrow ALUOUTPUT$ ("parcheggio" in attesa di WB)
WB	$RD \leftarrow ALUout1$

Esecuzione in pipeline di un'istruzione "MEM" (Load o Store)

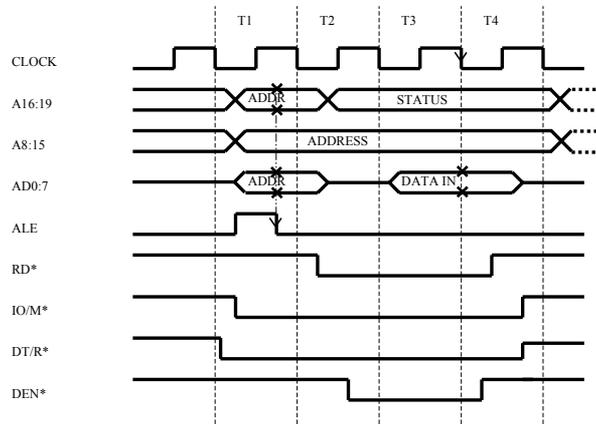
IF	$IR \leftarrow M[PC]; \quad PC \leftarrow PC+4$
ID	$A \leftarrow RS1; B \leftarrow RS2; PC1 \leftarrow PC; IR1 \leftarrow IR$
EX	$DMAR \leftarrow A \text{ op } IR_{0..15} \text{ ## } (IR_{15})^{16}$ $SMDR \leftarrow B$
MEM	$MDR \leftarrow M[DMAR]$ (LOAD) oppure $M[DMAR] \leftarrow SMDR$ (STORE)
WB	$RD \leftarrow MDR$ (LOAD)

Esecuzione in pipeline di un'istruzione "BRANCH"

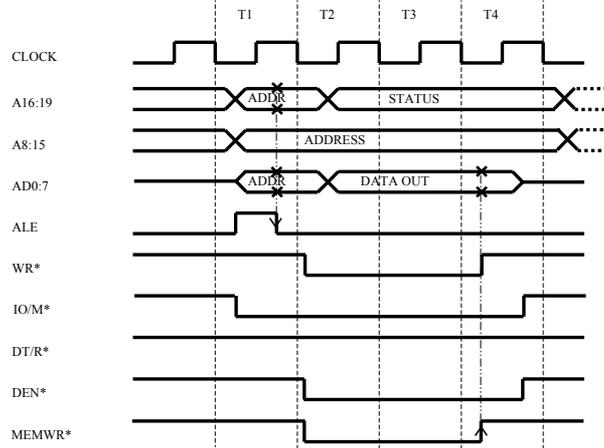
IF	$IR \leftarrow M[PC]; \quad PC \leftarrow PC+4$
ID	$A \leftarrow RS1; B \leftarrow RS2; PC1 \leftarrow PC; IR1 \leftarrow IR$
EX	$BTA \leftarrow PC1 + IR_{0..15} \text{ ## } (IR_{15})^{16}$ $Cond \leftarrow A \text{ op } 0$
MEM	if (Cond) $PC \leftarrow BTA$
WB	-----

BTA = BRANCH TARGET ADDRESS
indirizzo del salto (calcolato in ID)

8088 - Ciclo di lettura



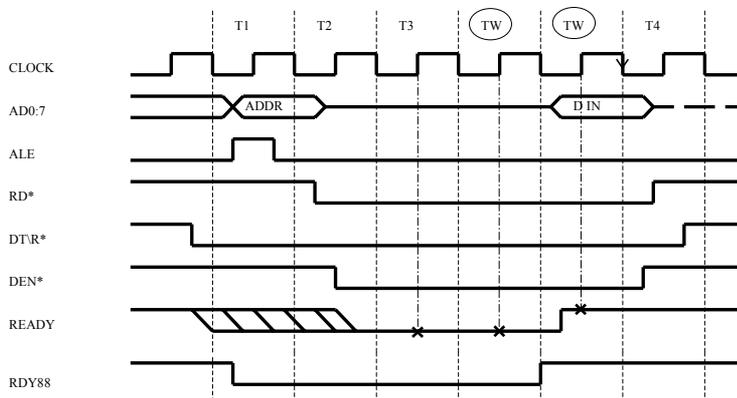
8088 - Ciclo di scrittura



materiale utile per esercizi

17

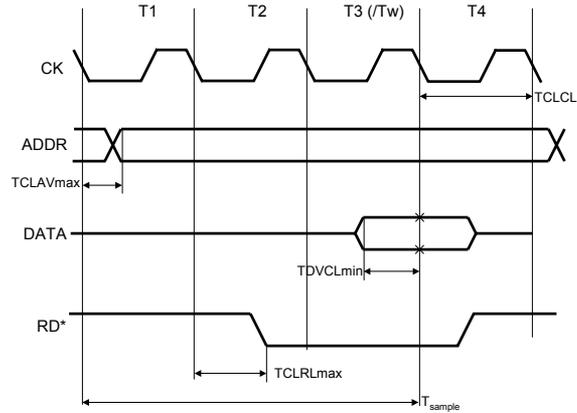
8088 - Ciclo di ready



materiale utile per esercizi

18

8088 Temporizzazioni Read cycle (minimum mode)



T_{sample} = tempo di campionamento su MDR del dato in lettura, misurato dall'inizio del ciclo.

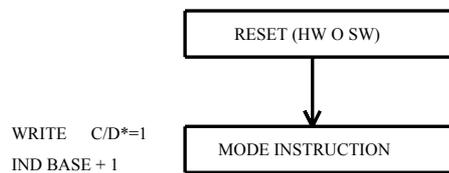
Fonte: Manuale Intel 8088 Pag. 2-78

timings [ns]	TCLCL	TCLA(max)	TCLR	TDVCL(min)	
@ 5 MHz	200	110	165	30	T ₃₇₃ = 18 ns
@ 8 MHz	125	60	100	20	T ₂₄₅ = 10 ns
					T _{dec PAL} = 30 ns

materiale utile per esercizi

19

8251 - Programmazione

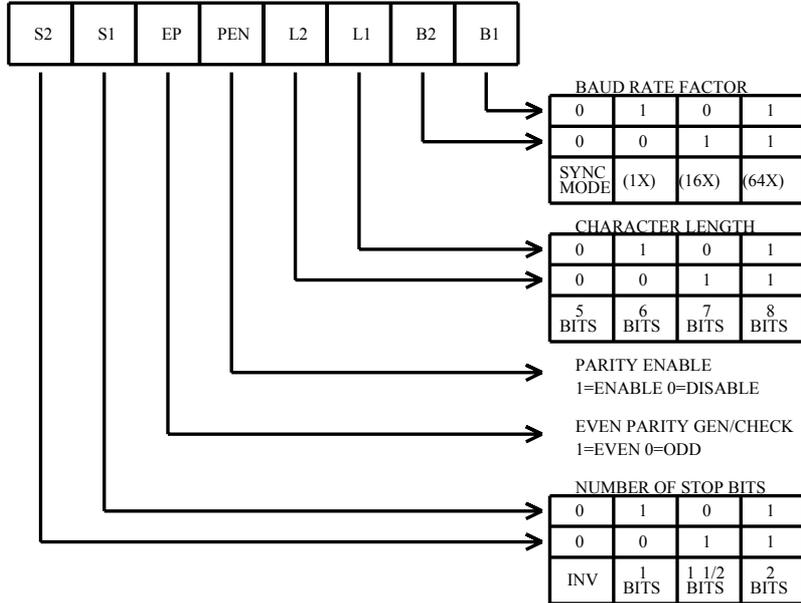


	C/D*	WRITE	READ
IND BASE + 0	0	DATA	DATA
IND BASE + 1	1	COMMAND INSTRUCTION	STATUS WORD

materiale utile per esercizi

20

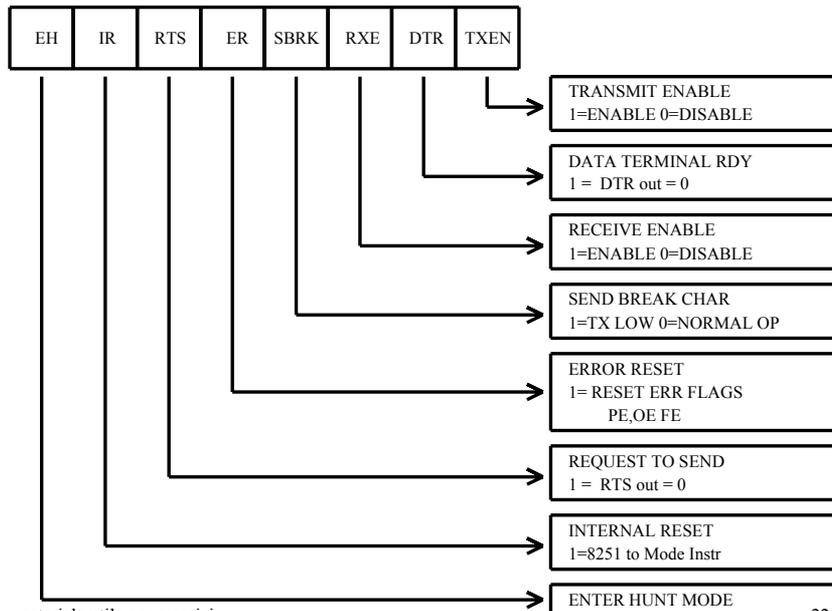
8251 - Mode Instruction



materiale utile per esercizi

21

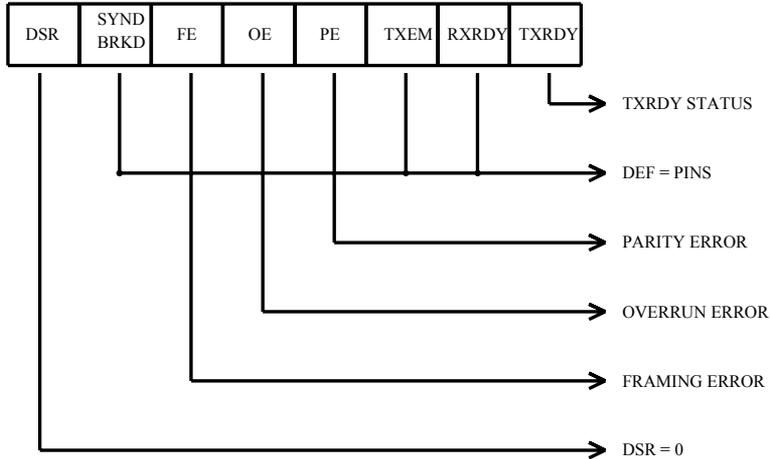
8251 - Command Instruction



materiale utile per esercizi

22

8251 - Status word



Inizializzazione COM1

Routine inizializzazione 8251

MODECW EQU FEh ;(2stopbit, parità pari, 8databit, /16)

CMDCW EQU 13h ; (asyn, noreset, RTS#=1, reseterror, normalop, Rxdisabled, DTR#=0, Txenabled)

```
ini_8251 PROC FAR
    PUSH AX
    MOV AL, MODECW
    OUT INDB51+1, AL ; INDB51 è l'ind. base dell'8251
    MOV AL, CMDCW
    OUT INDB51+1, AL
    POP AX
    RET
ini_8251 ENDP
```

INDB51 è l'indirizzo "base" dell'8251 nello spazio di I/O cioè il più basso degli indirizzi riservati a questa periferica

Letture e Scrittura su COM1 in modalità "polling"

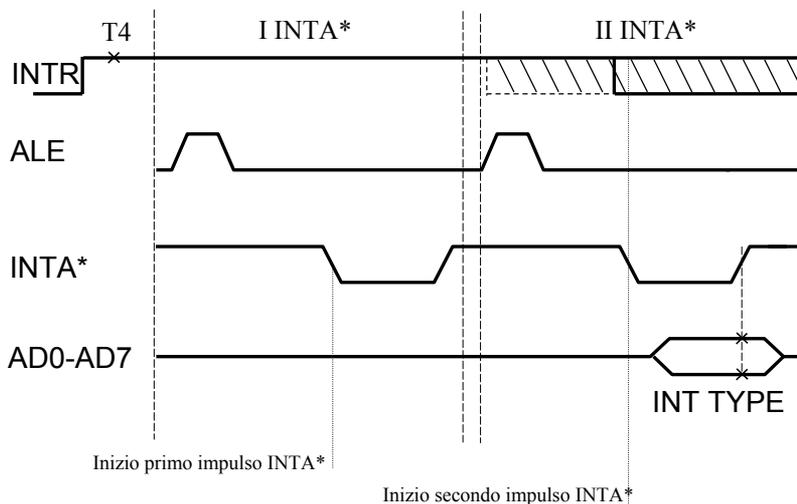
```

read_com1 PROC FAR
; la procedura legge un carattere da COM1
; e lo passa al chiamante tramite AL
; Attesa che il buffer di ricezione sia pieno: il bit 1
; di Status Word fornisce lo stato del buffer di
; ricezione ( 1=pieno, 0=vuoto )
wait_pieno: IN  AL, INDB51+1
            TEST AL, 02H
            JZ  wait_pieno
; lettura del carattere ricevuto
            IN  AL, INDB51
            RET
read_com1 ENDP

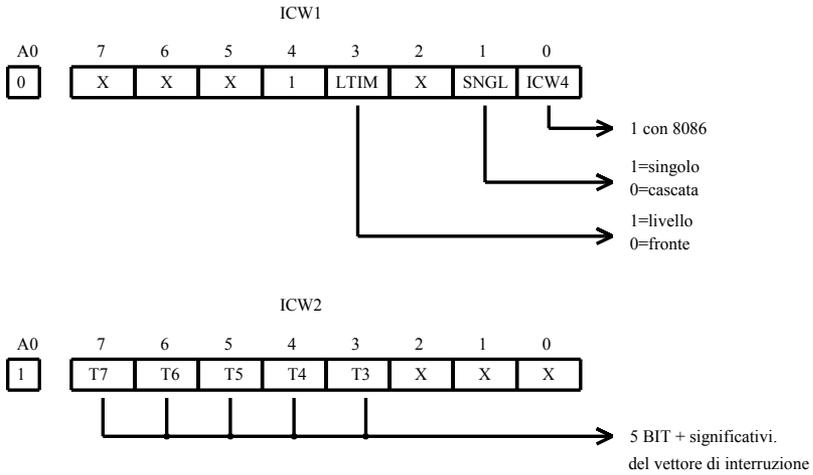
write_com1 PROC FAR
; la procedura scrive su COM1 il carattere passato
; dal chiamante tramite AL
            PUSH AX ; salva il dato sullo stack
; attesa che il buffer di trasmissione sia vuoto:
; il bit 0 di Status Word fornisce lo stato del buffer
; di trasmissione (1=vuoto, 0=pieno)
wait_vuoto: IN  AL, INDB51+1
            TEST AL, 01H
            JZ  wait_vuoto
; scrittura del carattere da trasmettere
            POP  AX ; preleva il dato dallo stack
            OUT INDB51, AL
            RET
write_com1 ENDP
    
```

INDB51 è l'indirizzo "base" dell'8251 nello spazio di I/O cioè il più basso degli indirizzi riservati a questa periferica

8088 - Ciclo di Interrupt Acknowledge



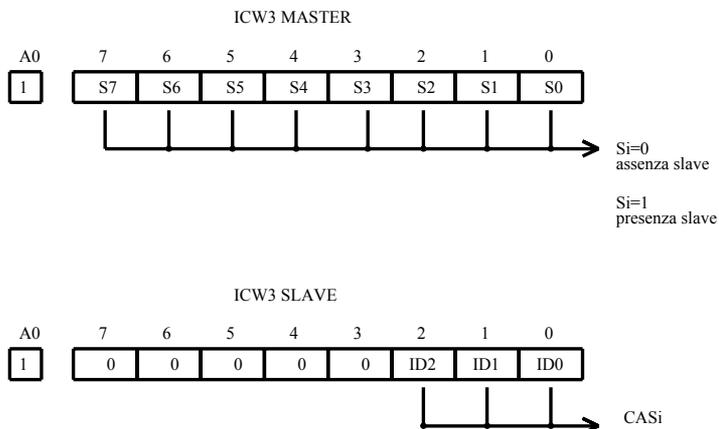
8259 - ICW1 e ICW2



materiale utile per esercizi

27

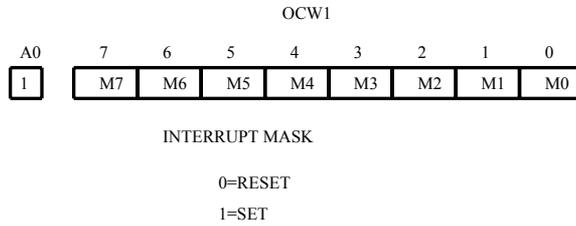
8259 - ICW3 master e slave



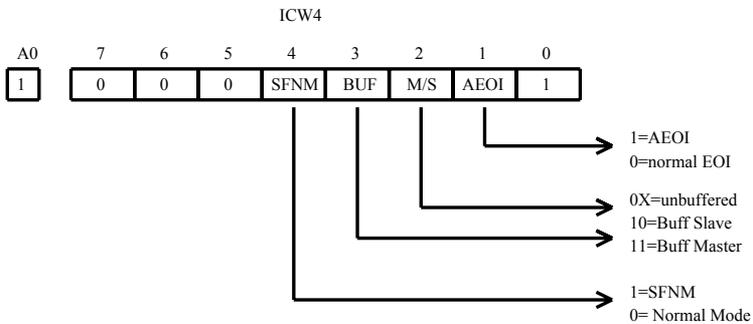
materiale utile per esercizi

28

8259 - OCW1



8259 - ICW4

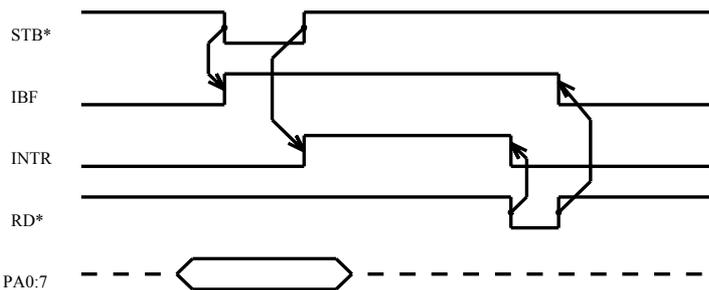
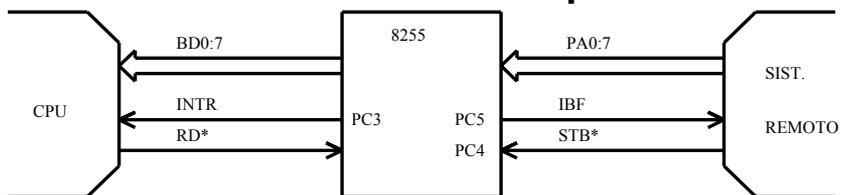


8259 – Routine di inizializzazione

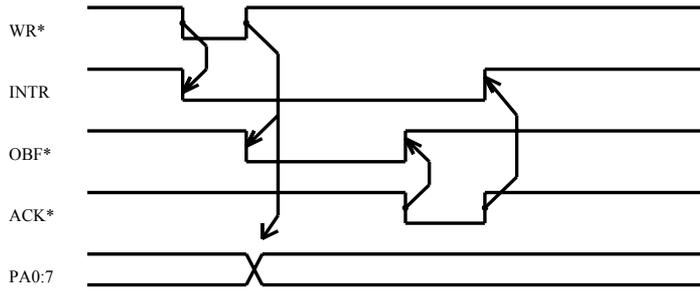
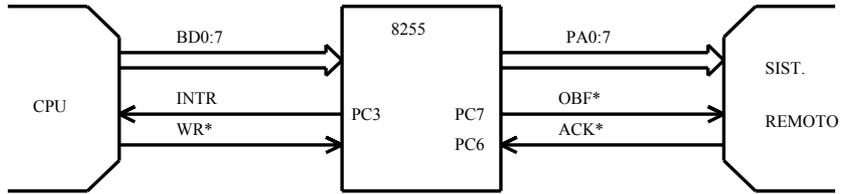
```
ICW1 EQU 13h      (fronte, singolo, +ICW4)
ICW2 EQU 80h      (int type base = 128 = 129°)
ICW4 EQU 03h      (f-nestd, unbuff., AEOL, 8088)

INI8259 PROC FAR
    PUSH AX
    MOV AL, ICW1
    OUT ADDR8259, AL ; ADDR8259 indir. base 8259
    MOV AL, ICW2
    OUT ADDR8259+1, AL
    MOV AL, ICW4
    OUT ADDR8259+1, AL
    POP AX
    RET
INI8259 ENDP
```

8255 - Modo 1 Input



8255 - Modo 1 Output



materiale utile per esercizi

33

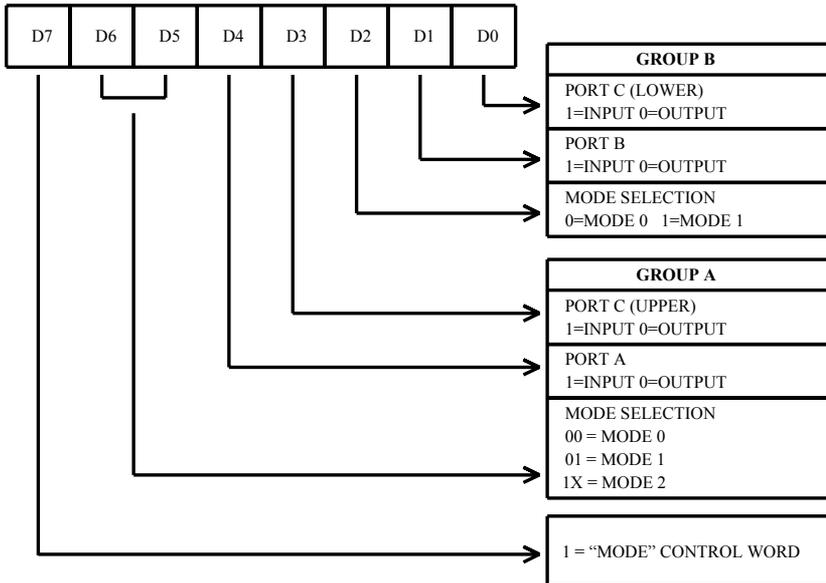
8255 - Programmazione

	A1	A0	READ	WRITE
IND BASE + 0	0	0	PORTA A	PORTA A
IND BASE + 1	0	1	PORTA B	PORTA B
IND BASE + 2	1	0	PORTA C	PORTA C
IND BASE + 3	1	1	—	CONTROL WORD

materiale utile per esercizi

34

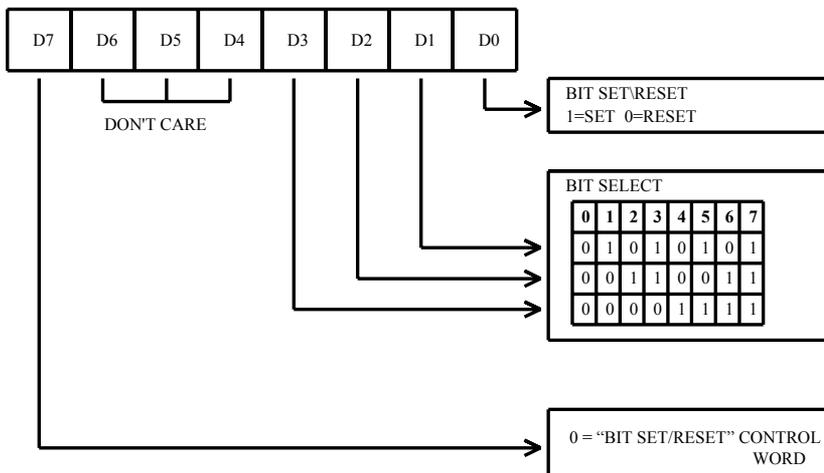
8255 - Mode Control Word



materiale utile per esercizi

35

8255 - Bit set/reset Control Word



materiale utile per esercizi

36

8255 – Routine di inizializzazione

CWMODE EQU B0h (porta A in mod01-input, don't care
su porte B e C)

```
INI8255 PROC FAR
    PUSH AX
    MOV AL, CWMODE
    OUT INDB55+3, AL ; INDB55 indir base 8255
    POP AX
    RET
INI8255 ENDP
```

8255 – Gestione a Polling e a Interrupt

```
poll_read_8255A PROC FAR
; la procedura legge un carattere dalla porta A
; in modo1-input dell'8255 interrogando il registro
; finchè non trova un carattere ricevuto
; il bit 3 della porta C contiene la copia SW del pin
; "INTRA" – La porta C si trova all'INDB55+2
; Attesa che il buffer di ricezione sia pieno...
wait_pieno: IN  AL, INDB55+2
            TEST AL, 04H
            JZ  wait_pieno
; lettura del carattere ricevuto
            IN  AL, INDB55
            RET
poll_read_8255A ENDP

int_read_8255A PROC FAR
; la procedura legge un carattere dalla porta A
; in modo1-input dell'8255 ed è eseguita su richiesta
; del PIC8259 a sua volta triggerato da "INTRA"
; dell'8255 stesso
; La procedura ritorna il carattere letto al chiamante
; tramite AL
            IN  AL, INDB55
            IRET
int_read_8255A ENDP
```