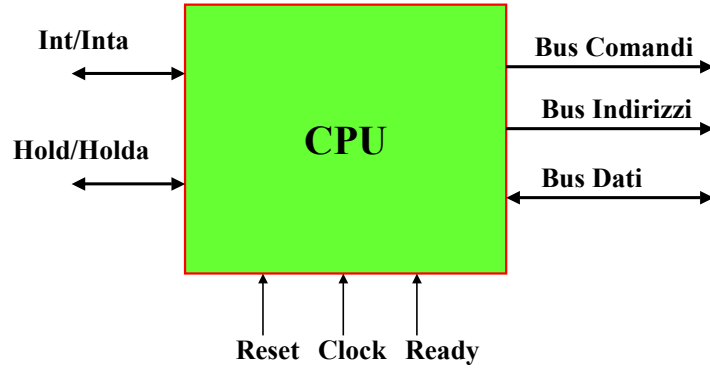


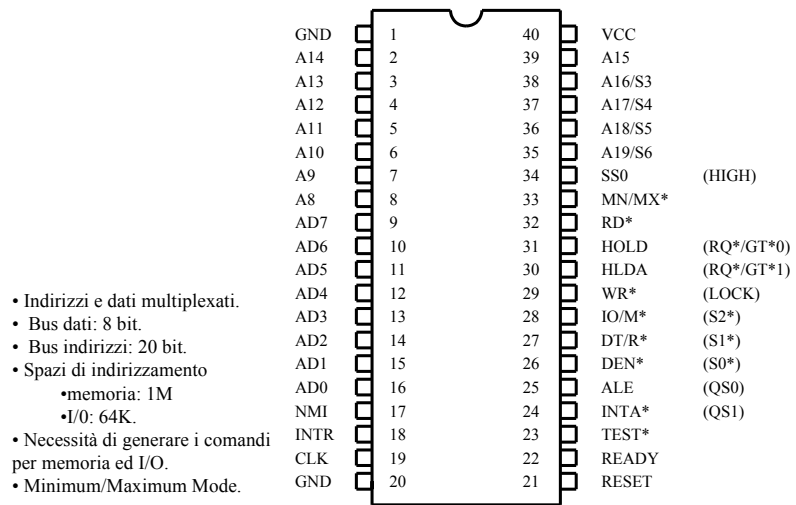
Pin-out "logico" di una CPU



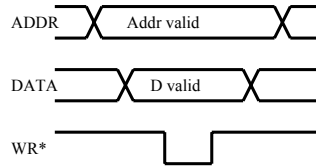
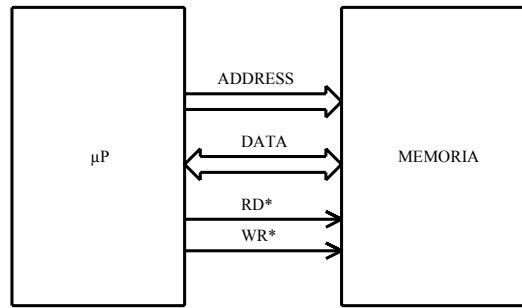
Architettura di un Sistema

- Il termine “architettura di un sistema basato su microprocessore” comprende le nozioni necessarie ad interfacciare con una CPU ed i suoi bus tutto l’hardware di cui il μP necessita per diventare un “calcolatore” (ossia un sistema a microprocessore).

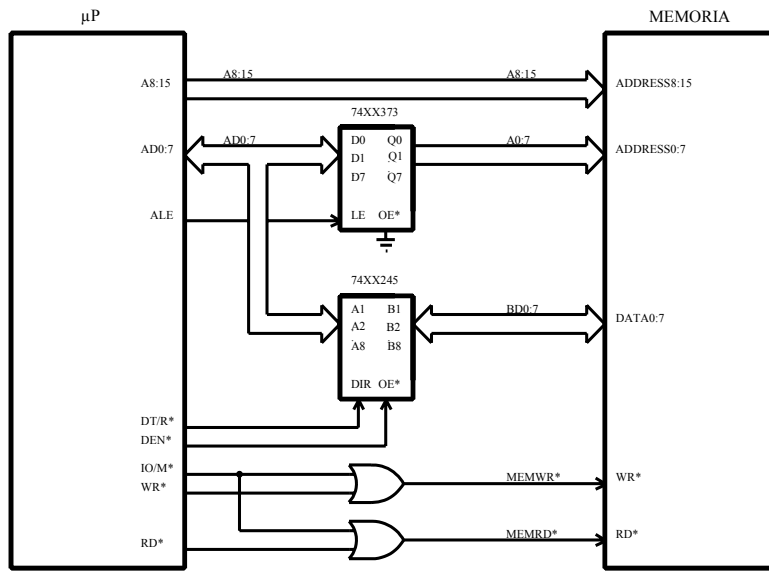
Pin-out μP 8088



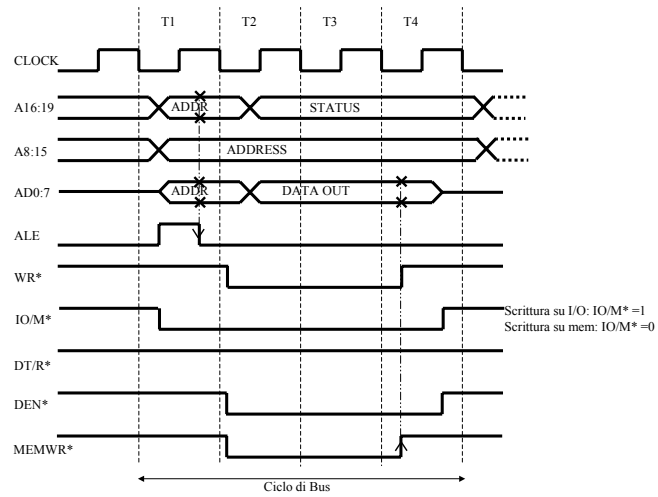
Interfacciamento μ P - memoria



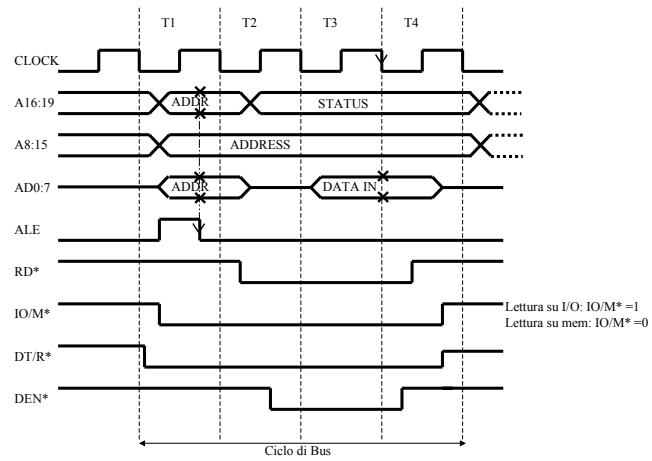
Interfacciamento μ P - memoria



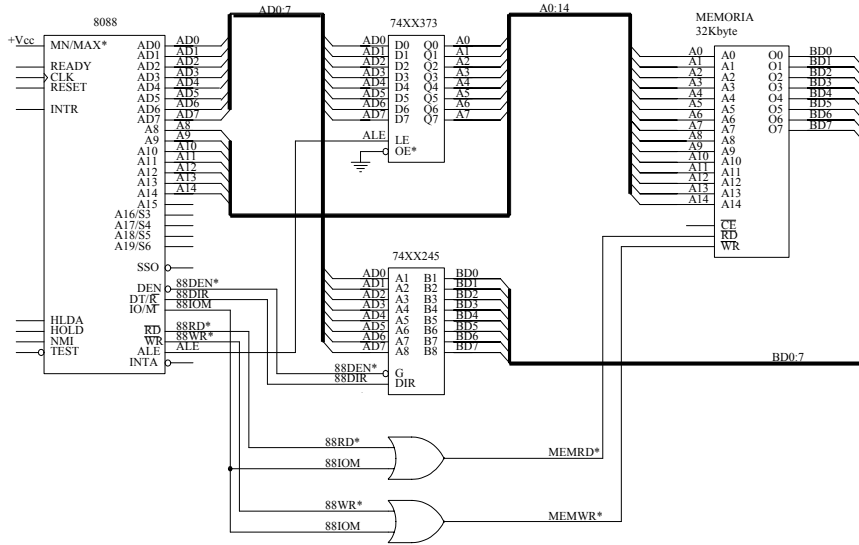
Ciclo di scrittura



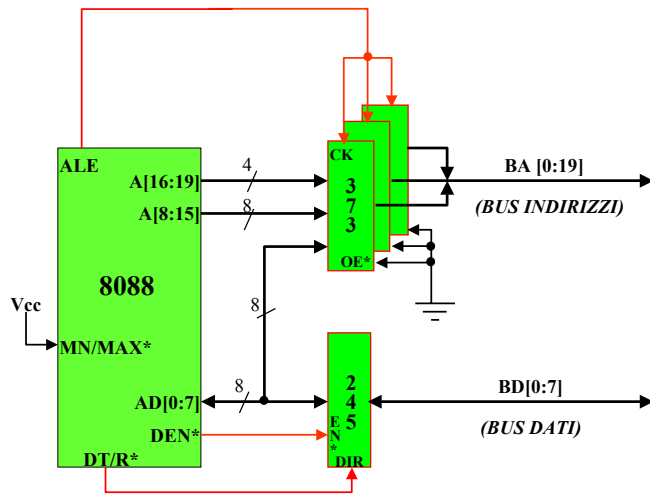
Ciclo di lettura



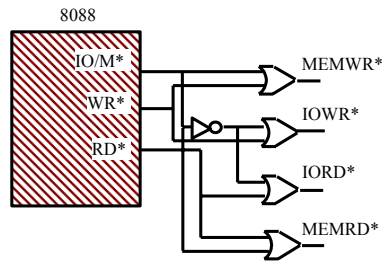
Interfacciamento μP - memoria



Bus demultiplexato



Segnali di lettura e scrittura



COMANDI PER I DISPOSITIVI
DI MEMORIA:

$$\text{MEMWR}^* = \text{IO/M}^* + \text{WR}^*$$

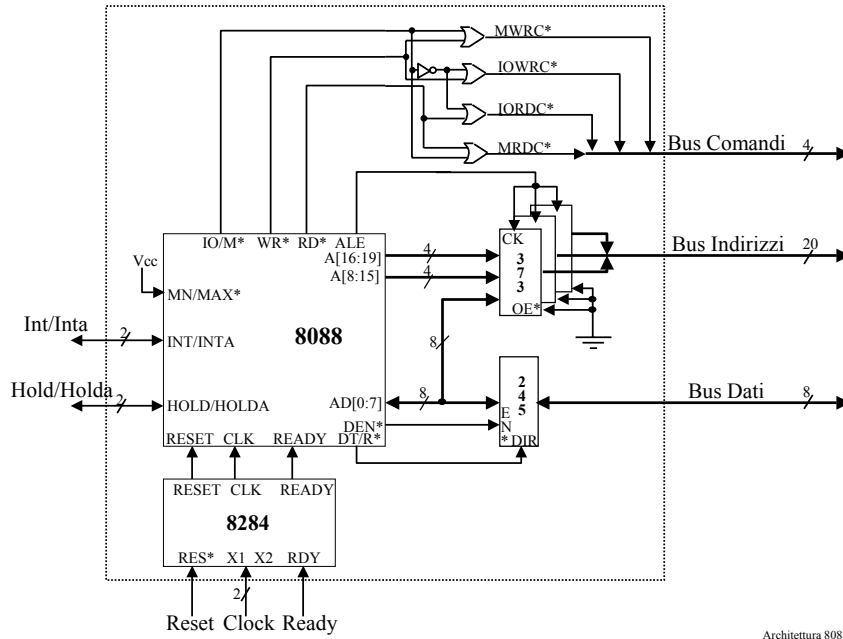
$$\text{MEMRD}^* = \text{IO/M}^* + \text{RD}^*$$

COMANDI PER I DISPOSITIVI
DI I/O:

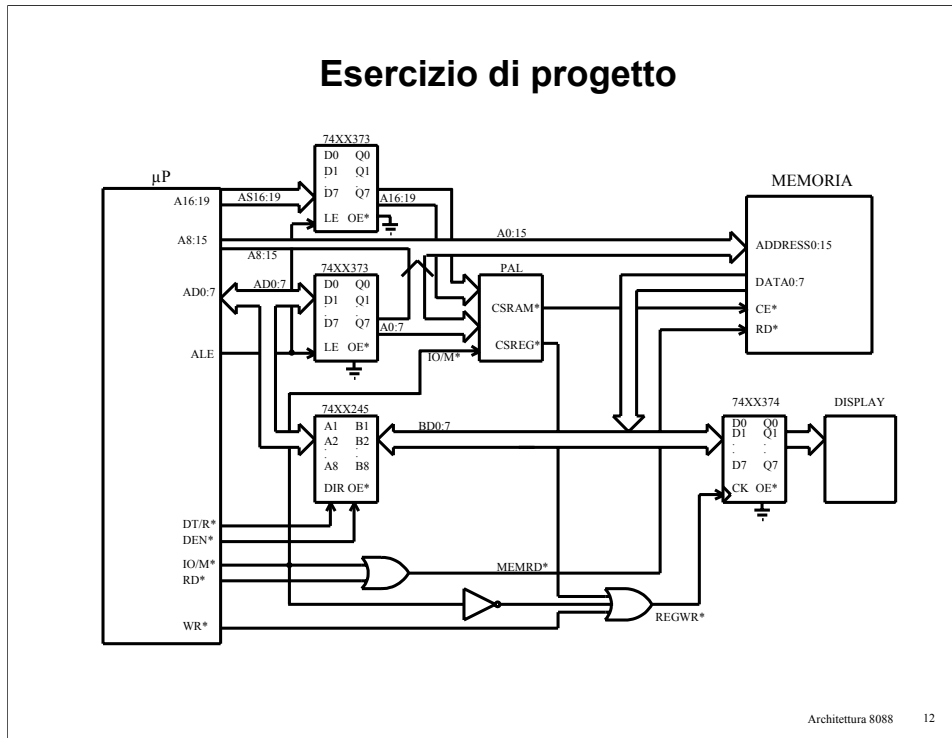
$$\text{IOWR}^* = \overline{\text{IO/M}^*} + \text{WR}^*$$

$$\text{IORD}^* = \overline{\text{IO/M}^*} + \text{RD}^*$$

Pin-out logico: sistema a μP basato su 8088



Esercizio di progetto



Esercizio di progetto di un sistema a microprocessore.

Si vuole interfacciare un registro 374 (a sua volta collegato a un display) al bus di un sistema a microprocessore basato su 8088 funzionante in minimum mode. Il sistema è inoltre dotato di un unico banco di memoria EPROM 27512 da 64 KB ($64KB=2^{16}$) da mappare nella parte alta dello spazio di memoria a partire dall'ultimo indirizzo. Progettare il sistema indicandone lo schema nell'ipotesi che il registro debba essere visibile alla porta 27h. Indicare le decodifiche complete e semplificate di entrambi i dispositivi allocati sul bus. Si scriva il codice assembler dell'iA16 che implementa un loop che ad ogni esecuzione incrementa di 1 il valore visualizzato dal display.

Decodifica della selezione del banco di memoria.

Gli indirizzi coinvolti sono da F0000 a FFFFF, in binario:

$\overline{1111} \ 0000 \ \overline{1111} \ \overline{1111}$ quindi le linee che non variano sono:

$$CS_RAM = BA_{19} \cdot BA_{18} \cdot BA_{17} \cdot BA_{16} \cdot \overline{IO/M}^*$$

Mentre la decodifica semplificata si può ridurre al solo $CSRAM = \overline{IO/M}^*$

Decodifica della selezione del registro.

L'indirizzo coinvolto è 27h = 0010 0111 perciò la decodifica completa è:

$$CS_REG = \overline{BA}_{19} \cdot \dots \cdot \overline{BA}_6 \cdot BA_5 \cdot \overline{BA}_4 \cdot \overline{BA}_3 \cdot BA_2 \cdot BA_1 \cdot BA_0 \cdot IO/M^*$$

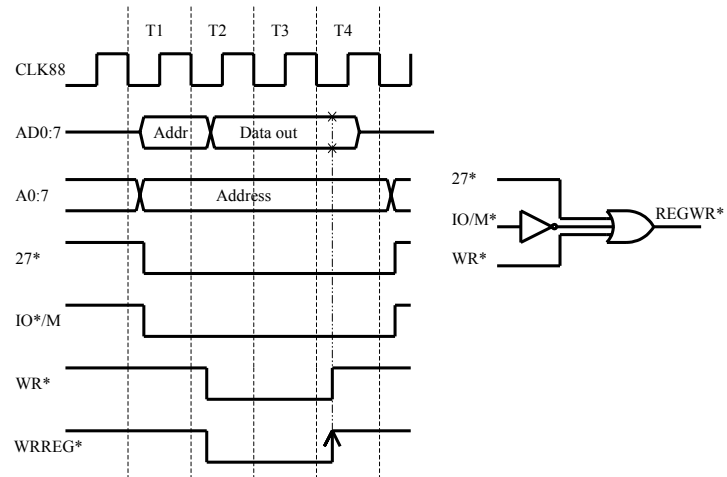
Ma dato che è l'unico dispositivo si può semplificare la decodifica con $CS_REG = IO/M^*$

Codice Assembler:

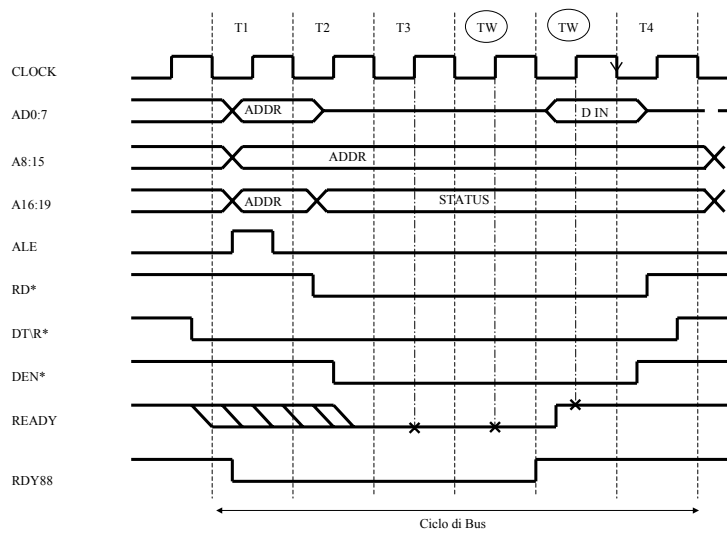
```

        JMP START ; prima istruzione eseguita all'accensione del μP
START:  MOV AL, 0 ; reset di AL
CICLO:  INC AL ; incremento di una unità di AL
        OUT 27h, AL ; copia sulla porta 27h il contenuto di AL
        JMP CICLO ; salta alla label CICLO
    
```

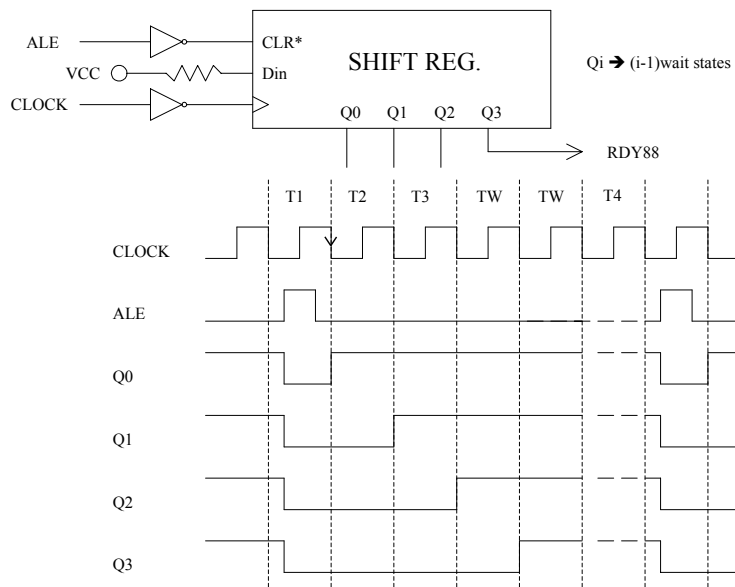
Esercizio: segnale di scrittura del registro



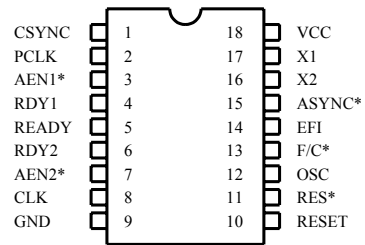
Ciclo di ready



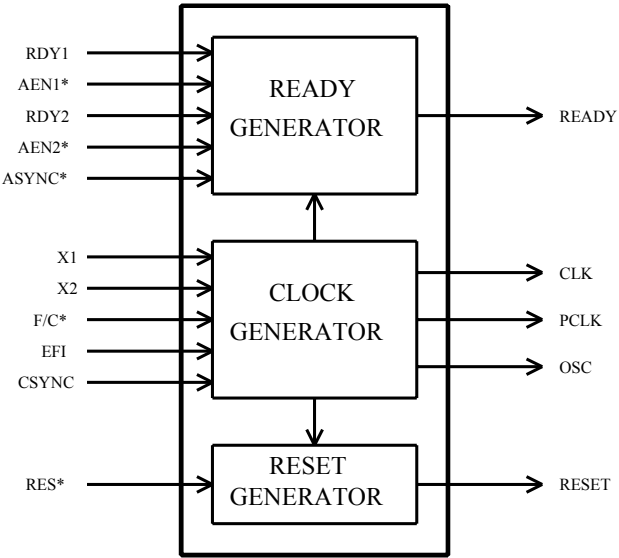
Esempio di generazione del segnale di ready

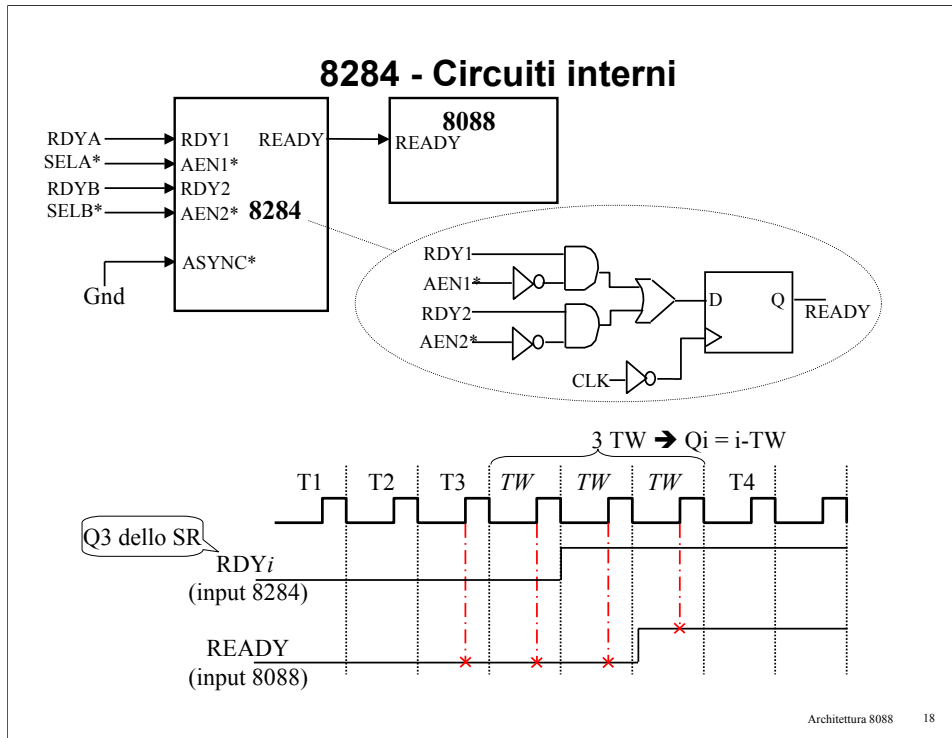


Dispositivo 8284



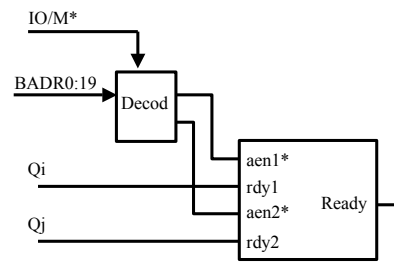
8284 - Schema a blocchi





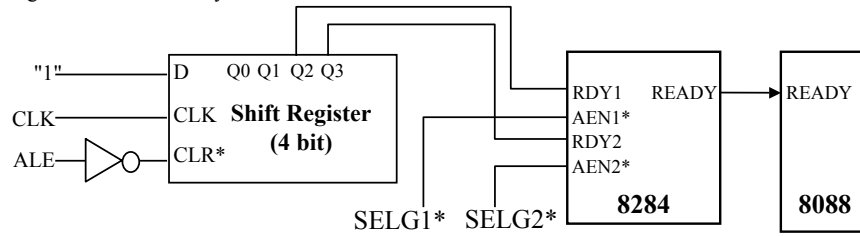
Collegando direttamente lo SR al READY dell'8088 si aveva $Q_i = (i-1)$ wait states
 Interponendo l'8284 tra lo SR e l'8088 si ottiene un "ritardo" di un periodo del segnale generato dallo SR e quindi $Q_i = i$ wait states.

8284 - Wait multipli



Alcuni esempi (1)

- Supponiamo di poter suddividere tutti i dispositivi presenti nel sistema in due gruppi, $G1$ e $G2$, che richiedono rispettivamente 2 e 3 stati di wait, e di disporre di due segnali, $SELG1^*$ e $SELG2^*$, attivi in corrispondenza dei cicli di bus relativi ai dispositivi dei due gruppi. Circuito per la generazione del Ready:



- Supponendo che i dispositivi appartenenti a $G1$ siano tutti i chip di memoria e quelli appartenenti a $G2$ tutti i dispositivi di I/O:

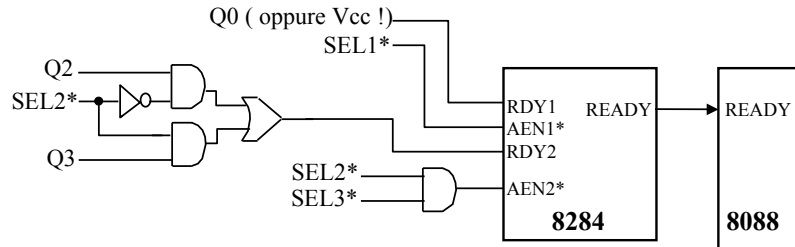
$$SELG1^* = IO/M^*, \quad SELG2^* = \overline{IO/M^*}$$

Alcuni esempi (2)

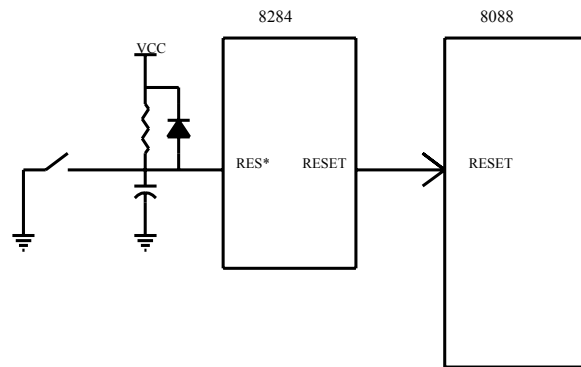
- Ipotizziamo invece che i dispositivi appartenenti a *G1* siano tutti i chip di memoria mappati ad indirizzi $\geq 512K$ (80000H) e quelli appartenenti a *G2* i chip di memoria mappati ad indirizzi $< 512K$ e tutti i dispositivi di I/O:

$$SELG1^* = IO/M^* + \overline{BA19}, \quad SELG2^* = \overline{(IO/M^*)} * BA19$$

- Supponiamo ora di poter suddividere i dispositivi in 3 gruppi, identificati da $SEL1^*$, $SEL2^*$ e $SEL3^*$, che richiedono rispettivamente 0, 2 e 3 stati di wait:



8284 - Segnale di Reset

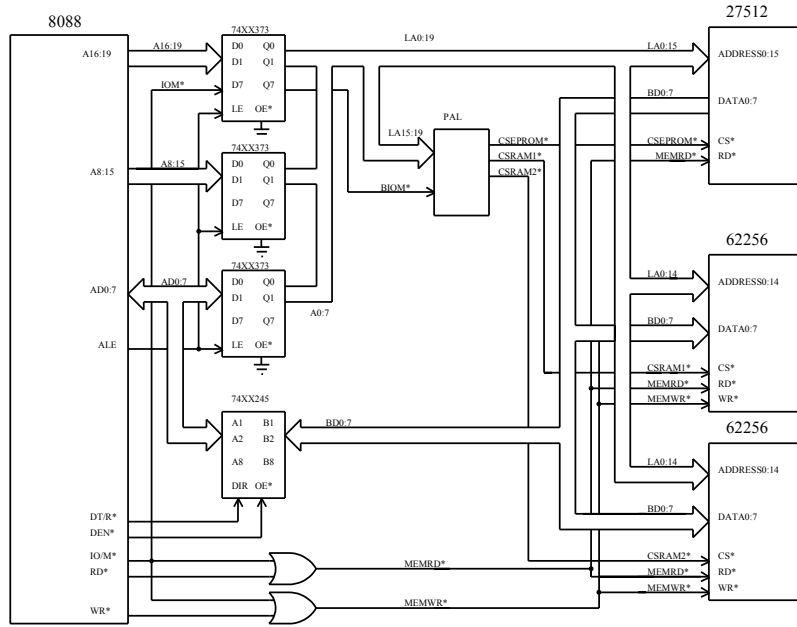


Esercizio

• Si consideri un sistema basato su 8088 dotato di 64K EPROM (1 chip) agli indirizzi alti (F0000H : FFFFFH), 128K RAM (4 chip) agli indirizzi bassi (0H : 1FFFFH) , dispositivi di I/O.

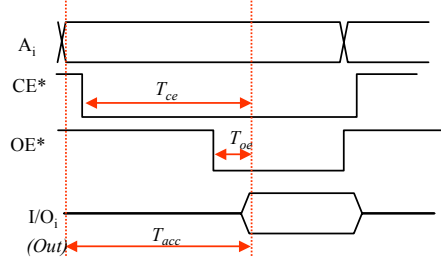
- 1) Si scrivano le espressioni dei CS* dei chip di memoria.
- 2) Si progetti il circuito per la generazione del READY nell'ipotesi che la EPROM richieda 1 wait-state, le RAM 0 wait-state ed i dispositivi di I/O 3 wait-state.

Esempio: circuiti integrati di memoria



Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (1)

- Forme d'onda e temporizzazioni fornite dal costruttore del chip di memoria:



Tipicamente



$$T_{acc} = T_{ce}$$

$$T_{acc} = T_{ce} \gg T_{oe} \quad (\text{es. } T_{acc} = T_{ce} \cong 2 \cdot T_{oe})$$

Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (2)

- Requisito per il corretto completamento di un ciclo di lettura:

T_{sample} : intervallo di tempo che intercorre fra l'inizio del ciclo di bus di lettura e l'istante in cui la CPU campiona i dati sul registro "MDR".

Poiché l'8088 campiona i dati sul fronte di discesa di T_3 (oppure, se il ciclo prevede stati di wait, sul fronte di discesa dell'ultimo TW), possiamo scrivere:

$$T_{sample} = (3+n) \cdot T_{CLCL}$$

con n =numero di stati di wait presenti nel ciclo.

$T_{data\ valid}$: intervallo di tempo che intercorre fra l'inizio del ciclo di bus di lettura e l'istante in cui i dati provenienti dalla memoria sono validi e possono quindi essere campionati dalla CPU.

Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (3)

Un ciclo di lettura viene eseguito correttamente se

$$T_{sample} \geq T_{data\ valid}$$

Conseguentemente, il calcolo degli stati di wait richiesti dal ciclo consiste nella determinazione del valore minimo di n che rende vera la relazione precedente:

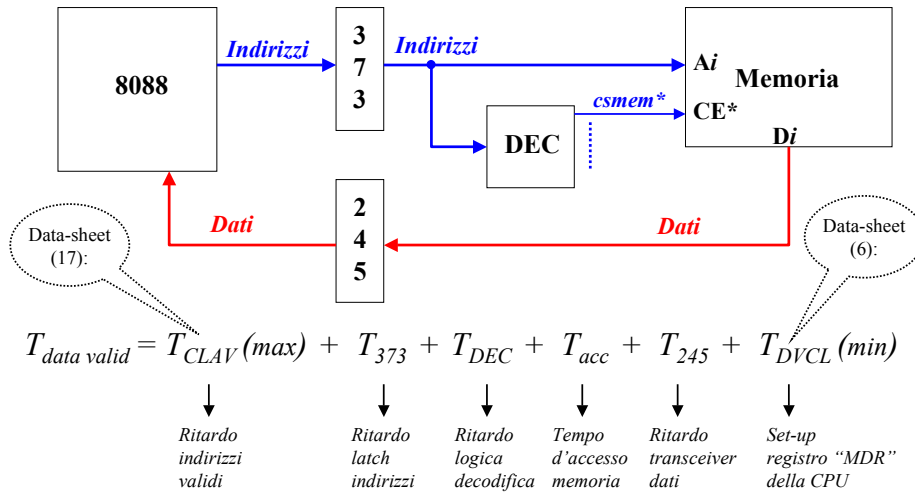
$$(3+n) \cdot T_{CLCL} \geq T_{data\ valid}$$

A tale scopo è necessario calcolare il valore di $T_{data\ valid}$.

Sarà considerato prima il ritardo associato al percorso degli indirizzi (Caso 1), poi quello associato al percorso dei comandi (Caso 2).

Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (4)

• Caso 1 (Percorso degli indirizzi, $T_{acc} = T_{ce}$)



Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (5)

• Esempi

A) Processore: 8088 8 MHz

$$T_{CLCL} = 125 \text{ ns}, T_{CLAV}(\text{max}) = 60 \text{ ns}, T_{DVCL}(\text{min}) = 20 \text{ ns}$$

Memoria:

$$T_{acc} = 45 \text{ ns} \text{ (data-sheet CY7C199-45 : } t_{AA}(\text{max}) = t_{ACE}(\text{max}) = 45 \text{ ns)}$$

Altri dispositivi presenti sul percorso degli indirizzi:

$$T_{373} = 18 \text{ ns} \quad (\text{data-sheet LS373: } t_{PHL}(\text{max}) = t_{PLH}(\text{max}) = 18 \text{ ns})$$

$$T_{245} \cong 10 \text{ ns} \quad (\text{data-sheet 74AC112245: } t_{PHL}(\text{max}) = t_{PLH}(\text{max}) = 9.5 \text{ ns})$$

$$T_{DEC} = 30 \text{ ns} \quad (\text{data-sheet PAL16L8 : } t_{pd}(\text{max}) = 30 \text{ ns})$$

$$(3+n) \cdot T_{CLCL} \geq T_{CLAV}(\text{max}) + T_{373} + T_{DEC} + T_{acc} + T_{245} + T_{DVCL}(\text{min})$$

$$(3+n) \cdot 125 \geq 60 + 18 + 30 + 45 + 10 + 20 = 183$$

↓
Soddisfatta per $n=0$ ($375 > 183$)

↓
0 wait-state

Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (6)

B) Processore: 8088 8 MHz

$$T_{CLCL} = 125 \text{ ns}, T_{CLAV}(\text{max}) = 60 \text{ ns}, T_{DVCL}(\text{min}) = 20 \text{ ns}$$

Memoria:

$$T_{acc} = 250 \text{ ns}$$

Altri dispositivi presenti sul percorso degli indirizzi:

$$T_{373} = 18 \text{ ns} \quad (\text{data-sheet LS373: } t_{PHL}(\text{max}) = t_{PLH}(\text{max}) = 18 \text{ ns})$$

$$T_{245} \cong 10 \text{ ns} \quad (\text{data-sheet 74AC112245: } t_{PHL}(\text{max}) = t_{PLH}(\text{max}) = 9.5 \text{ ns})$$

$$T_{DEC} = 30 \text{ ns} \quad (\text{data-sheet PAL16L8: } t_{pd}(\text{max}) = 30 \text{ ns})$$

$$(3+n) \cdot T_{CLCL} \geq T_{CLAV}(\text{max}) + T_{373} + T_{DEC} + T_{acc} + T_{245} + T_{DVCL}(\text{min})$$

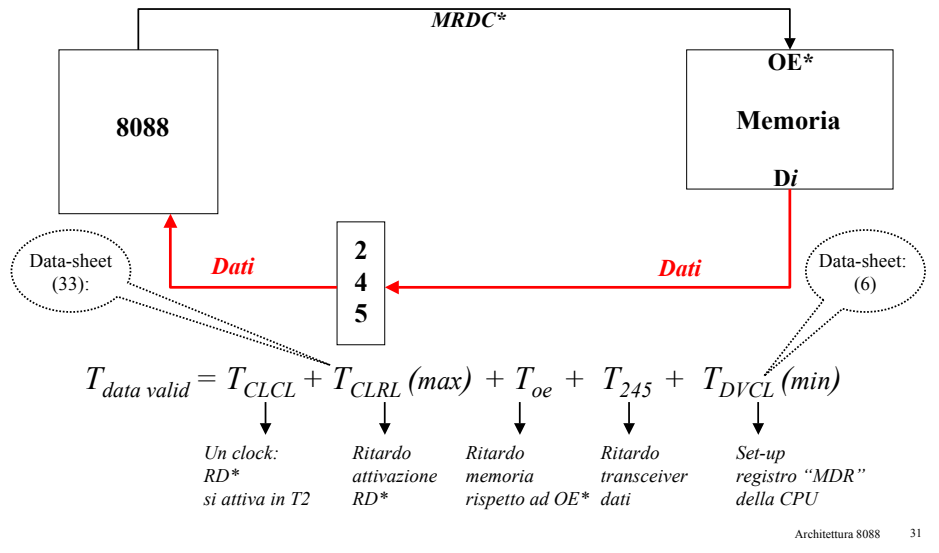
$$(3+n) \cdot 125 \geq 60 + 18 + 30 + 250 + 10 + 20 = 388$$

↓
Soddisfatta per $n=1$ ($500 > 388$)

↓
1 wait-state

Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (7)

- Caso 2 (Percorso dei comandi, T_{oe})



Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (8)

- Esempi

A) Processore: 8088 8 MHz

$$T_{CLCL} = 125 \text{ ns}, T_{CLRL}(\text{max}) = 100 \text{ ns}, T_{DVCL}(\text{min}) = 20 \text{ ns}$$

Memoria:

$$T_{oe} = 16 \text{ ns} \text{ (data-sheet CY7C199-45 : } t_{DOE}(\text{max}) = 16 \text{ ns)}$$

Altri dispositivi presenti sul percorso dei comandi:

$$T_{245} \cong 10 \text{ ns} \quad (\text{data-sheet 74AC112245: } t_{PHL}(\text{max}) = t_{PLH}(\text{max}) = 9.5 \text{ ns})$$

$$(3+n) \cdot T_{CLCL} \geq T_{CLCL} + T_{CLRL}(\text{max}) + T_{oe} + T_{245} + T_{DVCL}(\text{min})$$

$$(3+n) \cdot 125 \geq 125 + 100 + 16 + 10 + 20 = 271$$

↓
Soddisfatta per $n=0$ ($375 > 271$)

↓
0 wait-state

Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (9)

B) Processore: 8088 8 MHz

$$T_{CLCL} = 125 \text{ ns}, T_{CLRL}(\text{max}) = 100 \text{ ns}, T_{DVCL}(\text{min}) = 20 \text{ ns}$$

Memoria:

$$T_{oe} = 60 \text{ ns}$$

Altri dispositivi presenti sul percorso dei comandi:

$$T_{245} \cong 10 \text{ ns} \quad (\text{data-sheet } 74AC112245: t_{PHL}(\text{max}) = t_{PLH}(\text{max}) = 9.5 \text{ ns})$$

$$(3+n) \cdot T_{CLCL} \geq T_{CLCL} + T_{CLRL}(\text{max}) + T_{oe} + T_{245} + T_{DVCL}(\text{min})$$

$$(3+n) \cdot 125 \geq 125 + 100 + 60 + 10 + 20 = 315$$

↓
Soddisfatta per $n=0$ ($375 > 315$)

↓
0 wait-state

(se la memoria è quella dell'esempio B) visto precedentemente, il "caso peggiore" è quello associato al percorso degli indirizzi, quindi il ciclo richiede comunque 1 wait-state)

Calcolo degli stati di wait nel caso di cicli di lettura dalla memoria (10)

- In generale devono essere considerati entrambi i percorsi e la scelta del numero di stati di wait deve essere effettuata sulla base del caso peggiore.

Conseguentemente, il calcolo degli stati di wait richiesti da un ciclo di lettura può essere effettuato determinando il valore minimo di n che rende vera la seguente relazione:

$$(3+n) \cdot T_{CLCL} \geq \max \begin{cases} T_{CLAV}(\max) + T_{373} + T_{DEC} + T_{acc} + T_{245} + T_{DVCL}(\min) \\ T_{CLCL} + T_{CLRL}(\max) + T_{oe} + T_{245} + T_{DVCL}(\min) \end{cases}$$

Esercizio

• Si consideri un sistema basato su 8088 dotato di 256K EPROM agli indirizzi alti (C0000H : FFFFFH), 160K RAM agli indirizzi bassi (0H : 27FFFH), dispositivi di I/O. Per le memorie si considerino le seguenti temporizzazioni: EPROM: $T_{acc}=240\text{ ns}$, $T_{oe}=50\text{ ns}$; RAM: $T_{acc}=200\text{ ns}$, $T_{oe}=50\text{ ns}$.

- 1) Si scrivano le espressioni dei CS* dei chip di memoria.
- 2) Si determini il numero di stati di wait necessari per gli accessi in lettura ad EPROM e RAM.
- 3) Si progetti il circuito per la generazione del READY nell'ipotesi che, per le memorie, i cicli di scrittura richiedano lo stesso numero di stati di wait di quelli di lettura e che gli accessi ai dispositivi di I/O richiedano 3 stati di wait.

8088 - Forme d'onda cicli di bus (1)

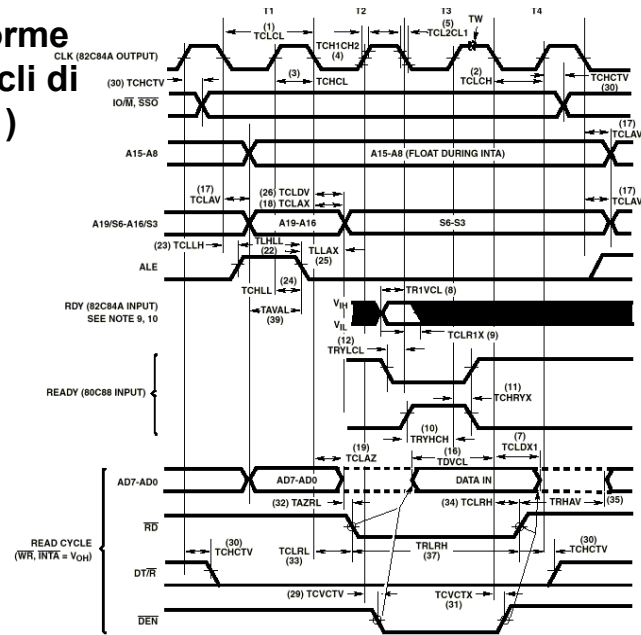


FIGURE 22. BUS TIMING - MINIMUM MODE SYSTEM

NOTES:

- 9. RDY is sampled near the end of T2, T3, T4 to determine if TW machine states are to be inserted.
- 10. Signals at 82C84A are shown for reference only.

8088 - Forme d'onda cicli di bus (2)

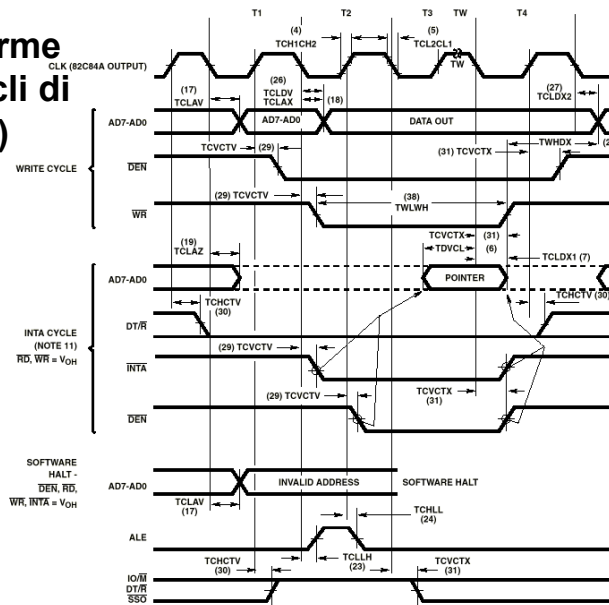


FIGURE 23. BUS TIMING - MINIMUM MODE SYSTEM (Continued)

NOTES:

- 11. Two INTA cycles run back-to-back. The 80C88 local ADDR/DATA bus is floating during both INTA cycles. Control signals are shown for the second INTA cycle.
- 12. Signals at 82C84A are shown for reference only.

Temporizzazioni 8088 (1)

MINIMUM COMPLEXITY SYSTEM

SYMBOL	PARAMETER	80C88		80C88-2		UNITS	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
TIMING REQUIREMENTS							
(1)	TCLCL	CLK Cycle Period	200	-	125	-	ns
(2)	TCLCH	CLK Low Time	118	-	68	-	ns
(3)	TCHCL	CLK High Time	69	-	44	-	ns
(4)	TCH1CH2	CLK Rise Time	-	10	-	10	ns
(5)	TCL2CL1	CLK Fall Time	-	10	-	10	ns
(6)	TDVCL	Data In Setup Time	30	-	20	-	ns
(7)	TCLDX1	Data In Hold Time	10	-	10	-	ns
(8)	TR1VCL	RDY Setup Time into 82C84A (Notes 6, 7)	35	-	35	-	ns
(9)	TCLR1X	RDY Hold Time into 82C84A (Notes 6, 7)	0	-	0	-	ns
(10)	TRYHCH	READY Setup Time into 80C88	118	-	68	-	ns
(11)	TCHRYX	READY Hold Time into 80C88	30	-	20	-	ns
(12)	TRYLCL	READY Inactive to CLK (Note 8)	-8	-	-8	-	ns
(13)	THVCH	HOLD Setup Time	35	-	20	-	ns
(14)	TINVCH	INTR, NMI, TEST Setup Time (Note 7)	30	-	15	-	ns
(15)	TILIH	Input Rise Time (Except CLK)	-	15	-	15	ns
(16)	TIHIL	Input Fall Time (Except CLK)	-	15	-	15	ns

Architettura 8088 38

Temporizzazioni 8088 (2)

MINIMUM COMPLEXITY SYSTEM

SYMBOL	PARAMETER	80C88		80C88-2		UNITS	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
TIMING RESPONSES							
(17)	TCLAV Address Valid Delay	10	110	10	60	ns	CL = 100pF
(18)	TCLAX Address Hold Time	10	-	10	-	ns	CL = 100pF
(19)	TCLAZ Address Float Delay	TCLAX	80	TCLAX	50	ns	CL = 100pF
(20)	TCHSZ Status Float Delay	-	80	-	50	ns	CL = 100pF
(21)	TCHSV Status Active Delay	10	110	10	60	ns	CL = 100pF
(22)	TLHLL ALE Width	TCLCH-20	-	TCLCH-10	-	ns	CL = 100pF
(23)	TCLLH ALE Active Delay	-	80	-	50	ns	CL = 100pF
(24)	TCHLL ALE Inactive Delay	-	85	-	55	ns	CL = 100pF

NOTES:

6. Signal at 82C84A shown for reference only.
7. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
8. Applies only to T2 state (8ns into T3).

Temporizzazioni 8088 (3)

MINIMUM COMPLEXITY SYSTEM

SYMBOL	PARAMETER	80C88		80C88-2		UNITS	TEST CONDITIONS
		MIN	MAX	MIN	MAX		
(25)	TLLAX Address Hold Time to ALE Inactive	TCHCL-10	-	TCHCL-10	-	ns	CL = 100pF
(26)	TCLDV Data Valid Delay	10	110	10	60	ns	CL = 100pF
(27)	TCLDX2 Data Hold Time	10	-	10	-	ns	CL = 100pF
(28)	TWHDX Data Hold Time After \overline{WR}	TCLCL-30	-	TCLCL-30	-	ns	CL = 100pF
(29)	TCVCTV Control Active Delay 1	10	110	10	70	ns	CL = 100pF
(30)	TCHCTV Control Active Delay 2	10	110	10	60	ns	CL = 100pF
(31)	TCVCTX Control Inactive Delay	10	110	10	70	ns	CL = 100pF
(32)	TAZRL Address Float to READ Active	0	-	0	-	ns	CL = 100pF
→ (33)	TCLRL \overline{RD} Active Delay	10	165	10	100	ns	CL = 100pF
(34)	TCLR \overline{H} \overline{RD} Inactive Delay	10	150	10	80	ns	CL = 100pF
(35)	TRHAV \overline{RD} Inactive to Next Address Active	TCLCL-45	-	TCLCL-40	-	ns	CL = 100pF
(36)	TCLHAV HLDA Valid Delay	10	160	10	100	ns	CL = 100pF
(37)	TRLRH \overline{RD} Width	2TCLCL-75	-	2TCLCL-50	-	ns	CL = 100pF
(38)	TWLWH \overline{WR} Width	2TCLCL-60	-	2TCLCL-40	-	ns	CL = 100pF
(39)	TAVAL Address Valid to ALE Low	TCLCH-60	-	TCLCH-40	-	ns	CL = 100pF
(40)	TOLOH Output Rise Time	-	15	-	15	ns	From 0.8V to 2.0V
(41)	TOHOL Output Fall Time	-	15	-	15	ns	From 2.0V to 0.8V

Architettura 8088 40

Schema base per il disegno dei progetti

