

RETI LOGICHE

SECONDA PROVA SCRITTA DELL'APPELLO DEL 14/04/97

Cognome:_____ Nome:_____

D1 Descrivere la tecnica per aggiungere ad un pacchetto dati di N bit i bit di controllo per ottenere un codice CRC

D2 Definire le funzioni di controllo di uno shifter a matrice per effettuare operazioni di rotazione e shift sia destra che a sinistra.

D3 Progettare un flip-flop D negative edge triggered

D4 Una rete sequenziale asincrona autosincronizzante è realizzata con una PLA ($t_{min}=6$ $t_{max}=24$ ns) un registro ($t_{setup}=5$ ns, $t_{cmin}=10$ nsec, $t_{cmax}=15$ ns) una porta OR ($t_{p1}= 7$ ns, $t_{p\epsilon}= 3$ ns): dimensionare il ritardo inerziale e stabilire la frequenza massima di funzionamento. Si supponga che le variabili di ingresso commutino una alla volta.

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 04-06-1997

STUDENTE (*): _____

MATRICOLA: _____

Progetto:

Sistema μ programmato PRSWITCH per la condivisione di una stampante fra 4 sistemi a microprocessore basati su PD32

Specifiche di interfacciamento:

- interfacce verso sistemi a μ P: 4 porte parallele a 8 bit gestite dai rispettivi micro PD32 su base interruzione (byte a byte)
- interfaccia stampante: porta parallela ad 8 bit dotata dei segnali di handshaking STROBE (PRSWITCH \rightarrow Stampante) e BUSY (PRSWITCH \leftarrow Stampante)
- protocollo di trasferimento dati verso la stampante: un byte può essere caricato nella porta d'uscita generando un impulso sul segnale STROBE solo dopo avere verificato che il segnale a livello BUSY sia disattivo (=0)

Specifiche funzionali:

- il controllore di PRSWITCH deve essere di tipo D-Mealy;
- per disaccoppiare la ricezione dei dati prodotti dai PD32 dalla trasmissione verso la stampante deve essere internamente prevista una RAM nella quale allocare 4 buffer tampone da 64Kbyte gestiti con disciplina FIFO;
- per evitare la miscelazione dei dati prodotti da sistemi diversi, la stampante deve essere alimentata dai dati estratti da un solo buffer FIFO alla volta; la trasmissione dei dati eventualmente presenti in un altro buffer potrà iniziare solo dopo che il buffer correntemente attivo rimanga vuoto per almeno 5 secondi;
- PRSWITCH deve gestire in modo concorrente la ricezione/memorizzazione dei dati provenienti dalle 4 porte di ingresso e la trasmissione dei dati del buffer attivo verso la porta stampante (è quindi possibile la "contemporanea" attività di tutte le 5 porte gestite da PRSWITCH).

Note/suggerimenti:

E' necessario tenere conto delle probabili situazioni di saturazione dei buffer. Si fa osservare che la gestione concorrente va intesa in senso logico: il SCO provvederà a sequenzializzare le varie operazioni di scrittura/lettura dei buffer e di gestione delle porte.

Si richiede:

1. lo schema a blocchi funzionali di PRSWITCH completo dei segnali di interfacciamento fra i blocchi ed il diagramma di timing del protocollo di handshaking sulle porte di ingresso e di uscita verso la stampante
2. di descrivere sinteticamente la tecnica adottata per la gestione concorrente dei processi di ricezione dati e stampa
3. lo schema elettrico di PRSWITCH ed il diagramma ASM del microprogramma di controllo

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 18-06-1997

STUDENTE: _____

MATRICOLA: _____

Progetto:

Sistema pipelined FASTDAS per l'acquisizione veloce di un pacchetto di dati analogici per PD32.

Specifiche funzionali:

- FASTDAS viene attivato da un segnale esterno START_ACQUISITION con periodo 50ms
- una volta attivato, FASTDAS deve campionare l'ingresso analogico ad una frequenza di 5MHz tramite un A/D converter con risoluzione migliore dello 0.1% acquisendo 4096 campioni;
- i dati acquisiti devono essere filtrati in tempo reale mediante media aritmetica su 16 campioni
- le medie via via ottenute devono essere trasferite in DMA nella memoria del PD32 in un buffer allocato a partire dall'indirizzo fisso FFFF0000 e con un formato long (32 bit)
- al termine dell'acquisizione FASTDAS genera un interrupt verso il PD32 per segnalare la disponibilità del buffer di dati filtrati

Specifiche dispositivi:

- i dati prodotti dall' A/DC sono espressi in complemento a 2; tempo di conversione fisso pari a 180ns. Il convertitore richiede il solo segnale di START_CONVERSION e non emette il segnale END_OF_CONVERSION.
- tempo di calcolo max. di un sommatore a 4 bit: 40ns.
- ritardi di tutti gli altri dispositivi SSI/MSI: 10ns.

Note/suggerimenti:

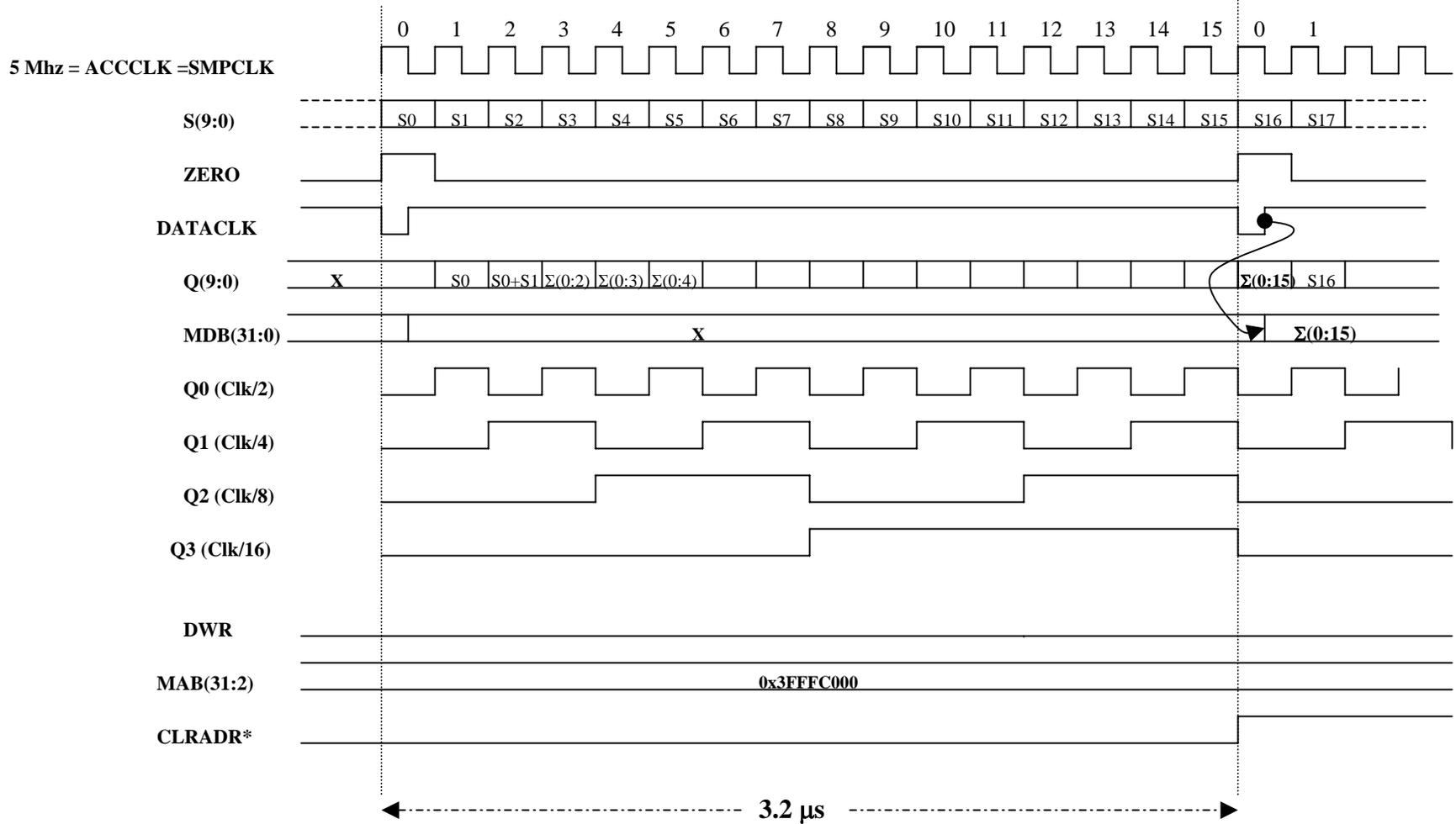
Si consiglia l'implementazione di un controllore cablato. Si assuma che il PD32 risponda alla richiesta dei bus di sistema entro 2 microsec. max. e che il ciclo di memory write sia non minore di 1.5 microsec. Dato che il processo di filtraggio dei campioni deve essere eseguito in tempo reale non si deve prevedere una memoria tampone per la memorizzazione intermedia dei campioni acquisiti.

Si richiedono:

1. lo schema a blocchi funzionali di FASTDAS completo dei segnali di interfacciamento fra i blocchi e del numero di bit necessari per i percorsi dati
2. il diagramma di timing dettagliato relativo al flusso dei dati all'interno di FASTDAS ed alla generazione dei segnali di gestione della memoria del PD32
3. lo schema elettrico di FASTDAS

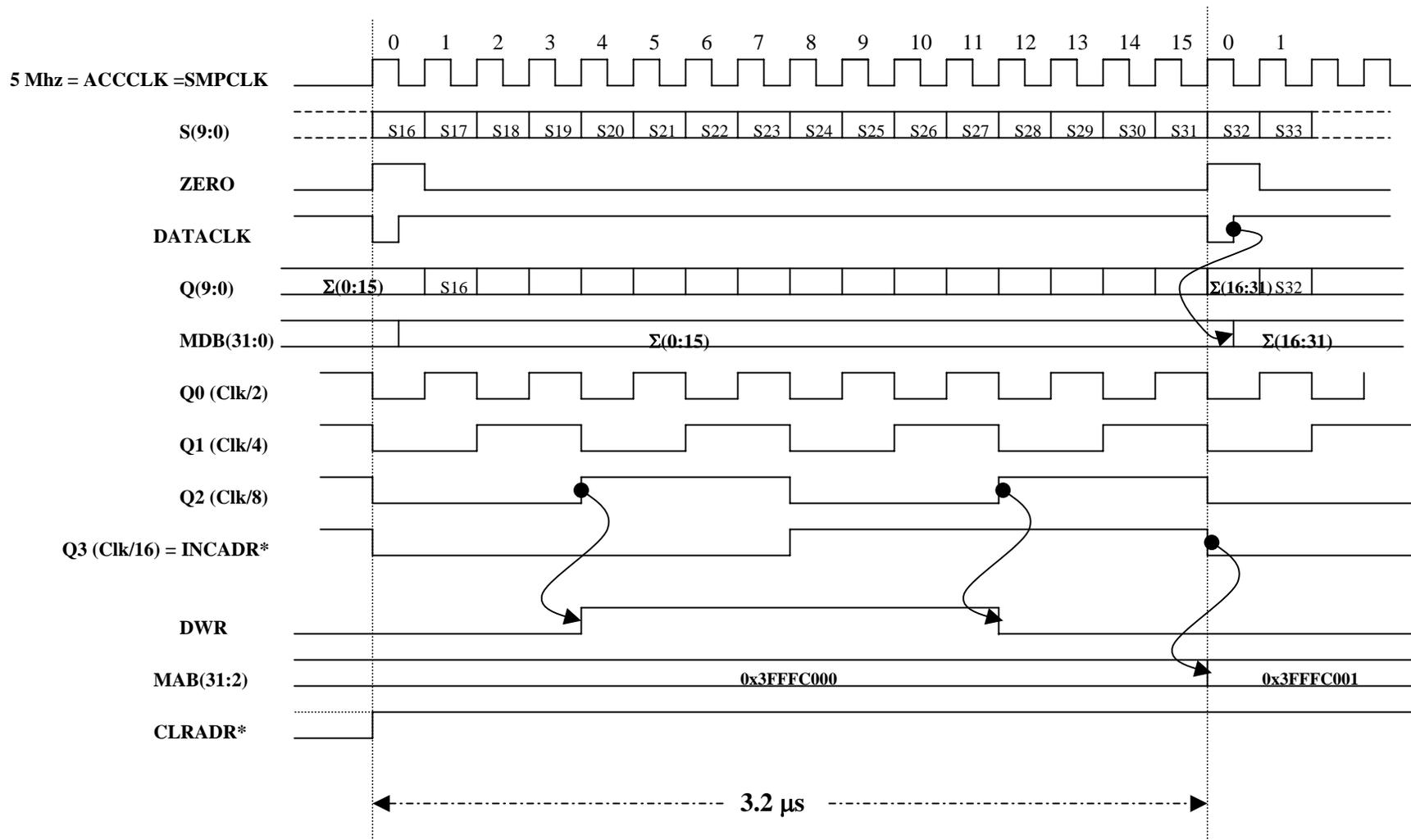
Timing interno FASTDAS

(inizio acquisizione)



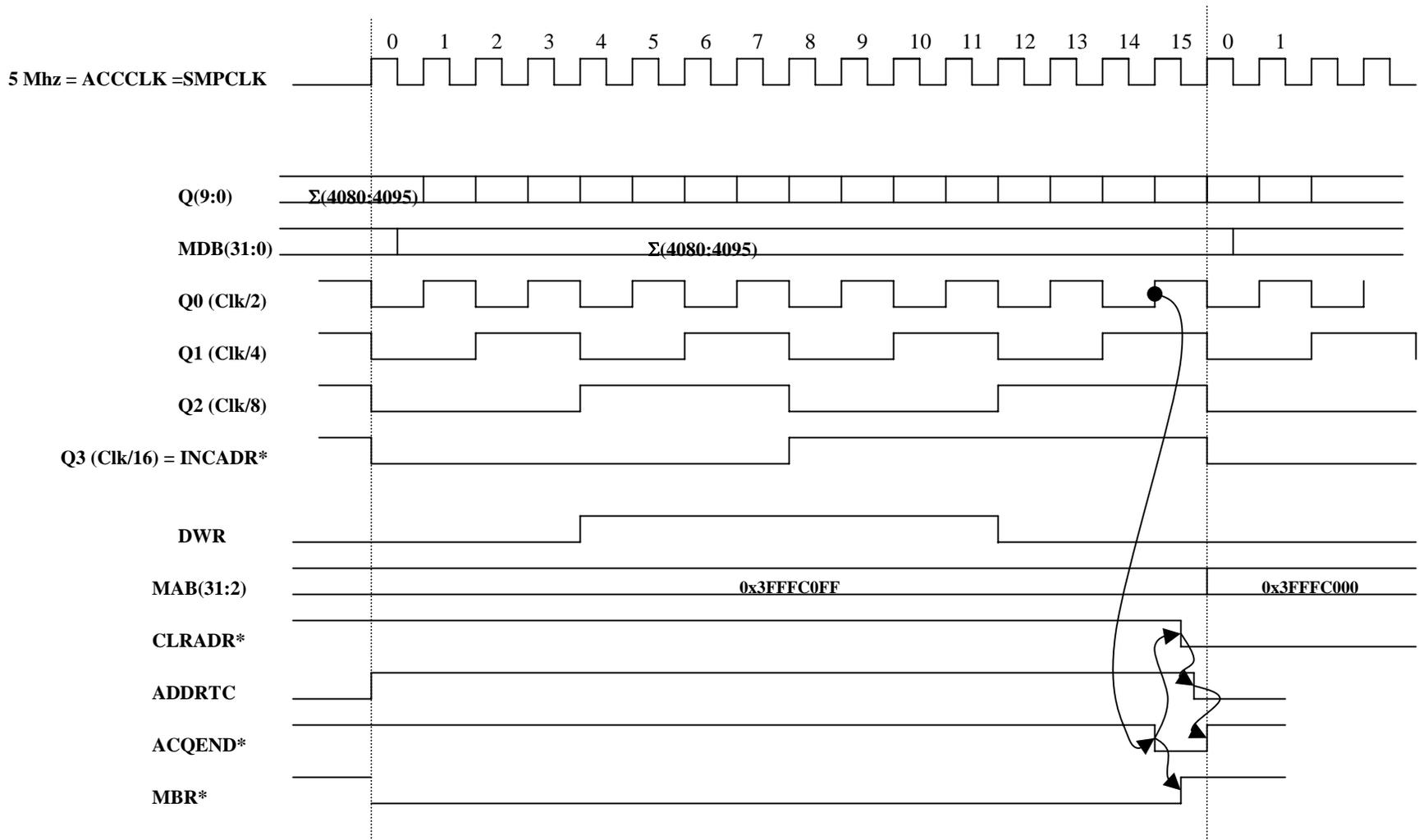
Timing interno FASTDAS

(fase intermedia)

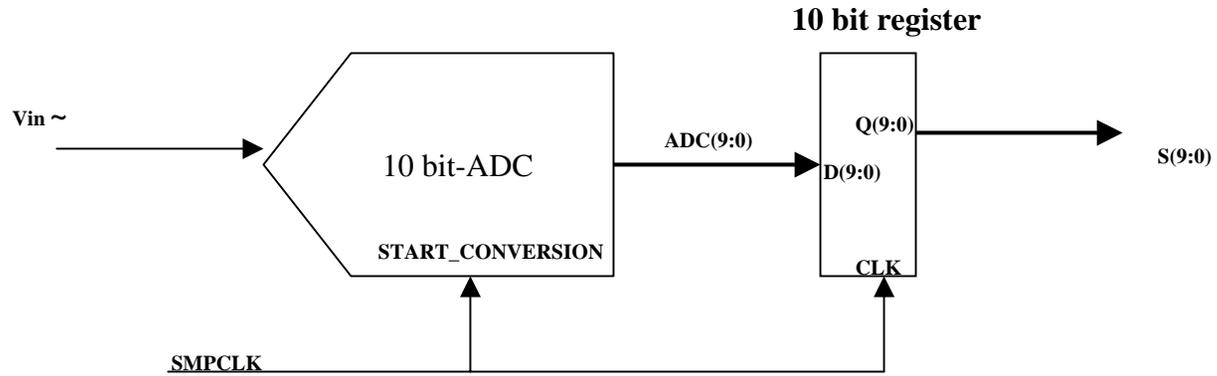


Timing interno FASTDAS

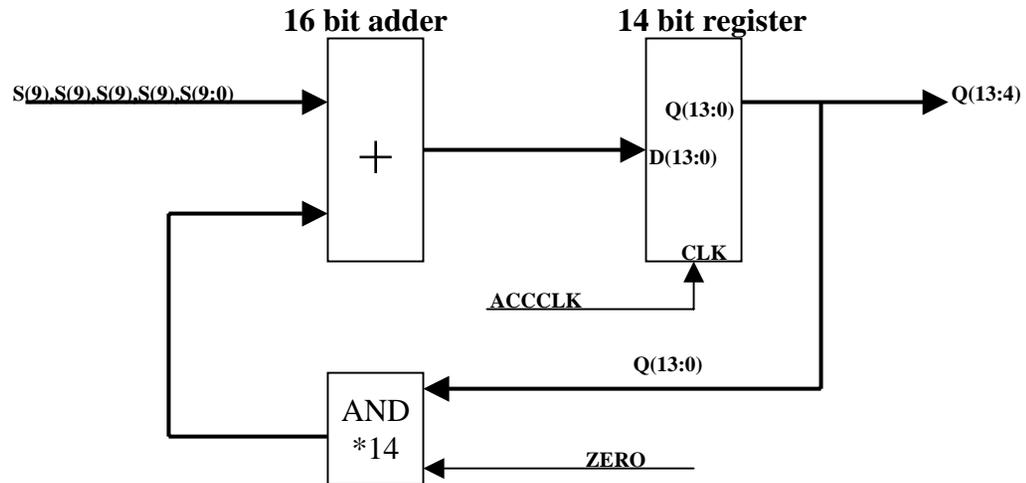
(fine acquisizione)



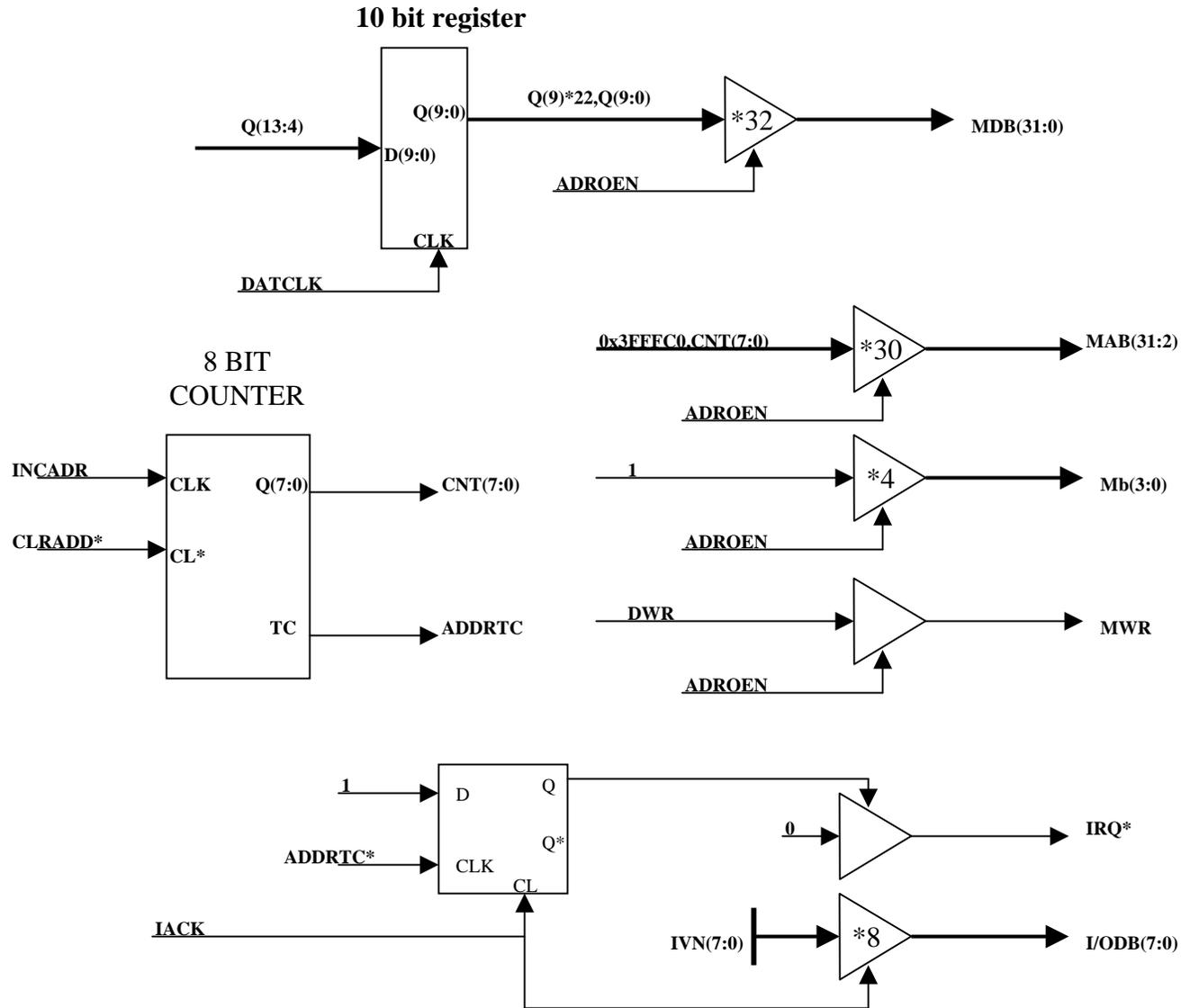
FASTDAS - CAMPIONAMENTO



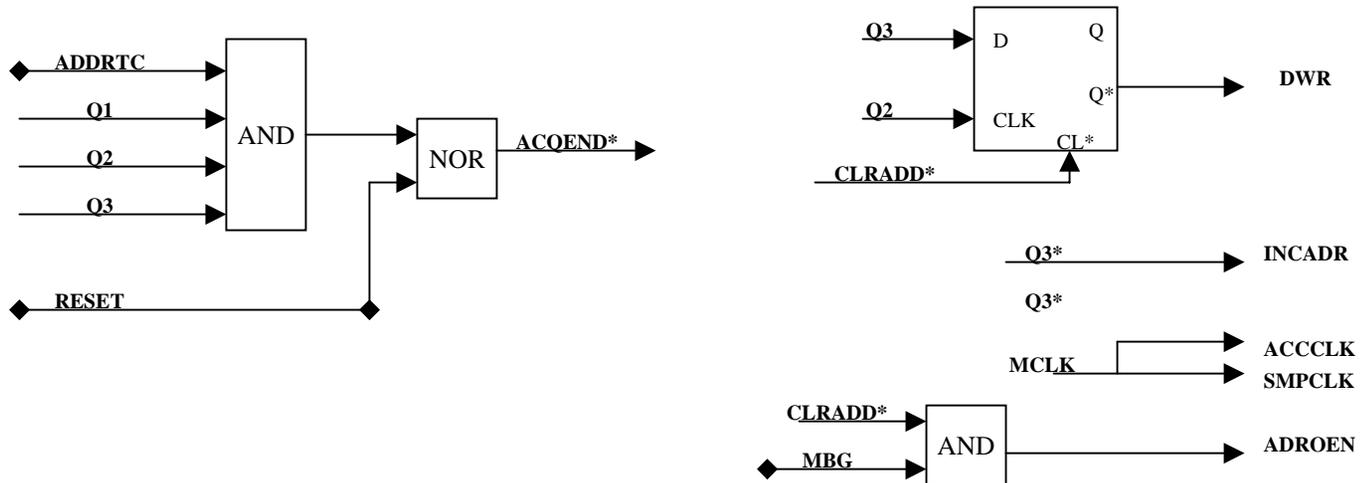
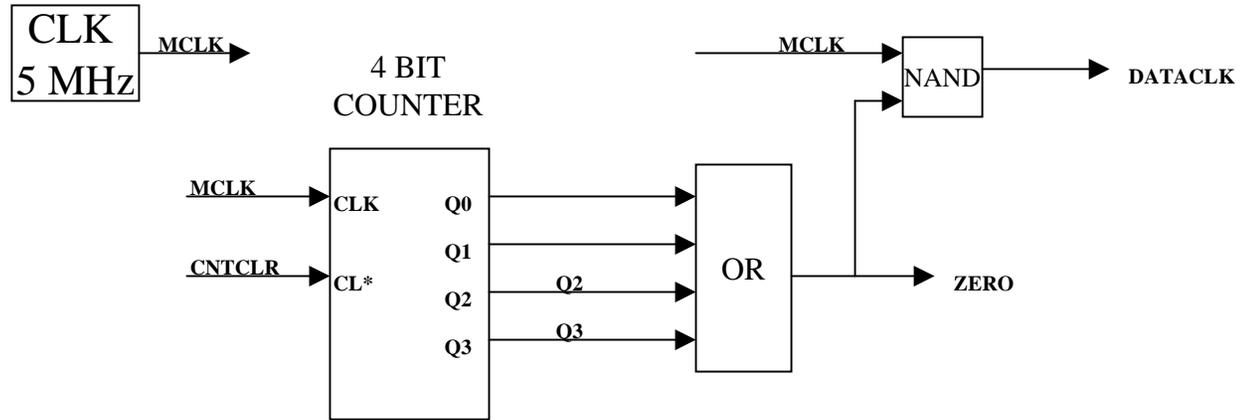
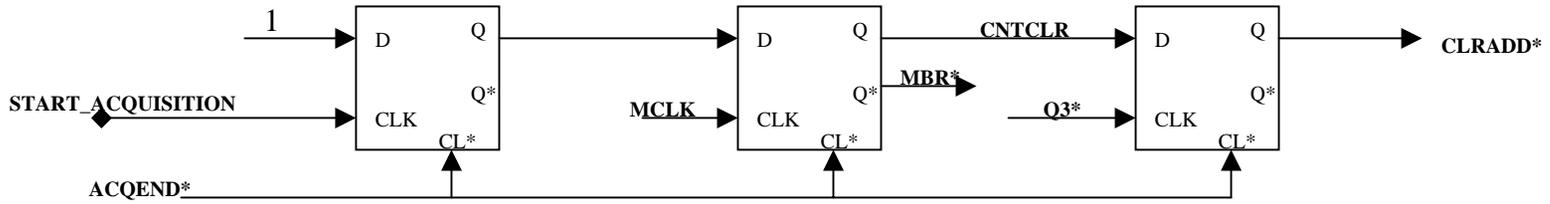
FASTDAS - MEDIA ARITMETICA



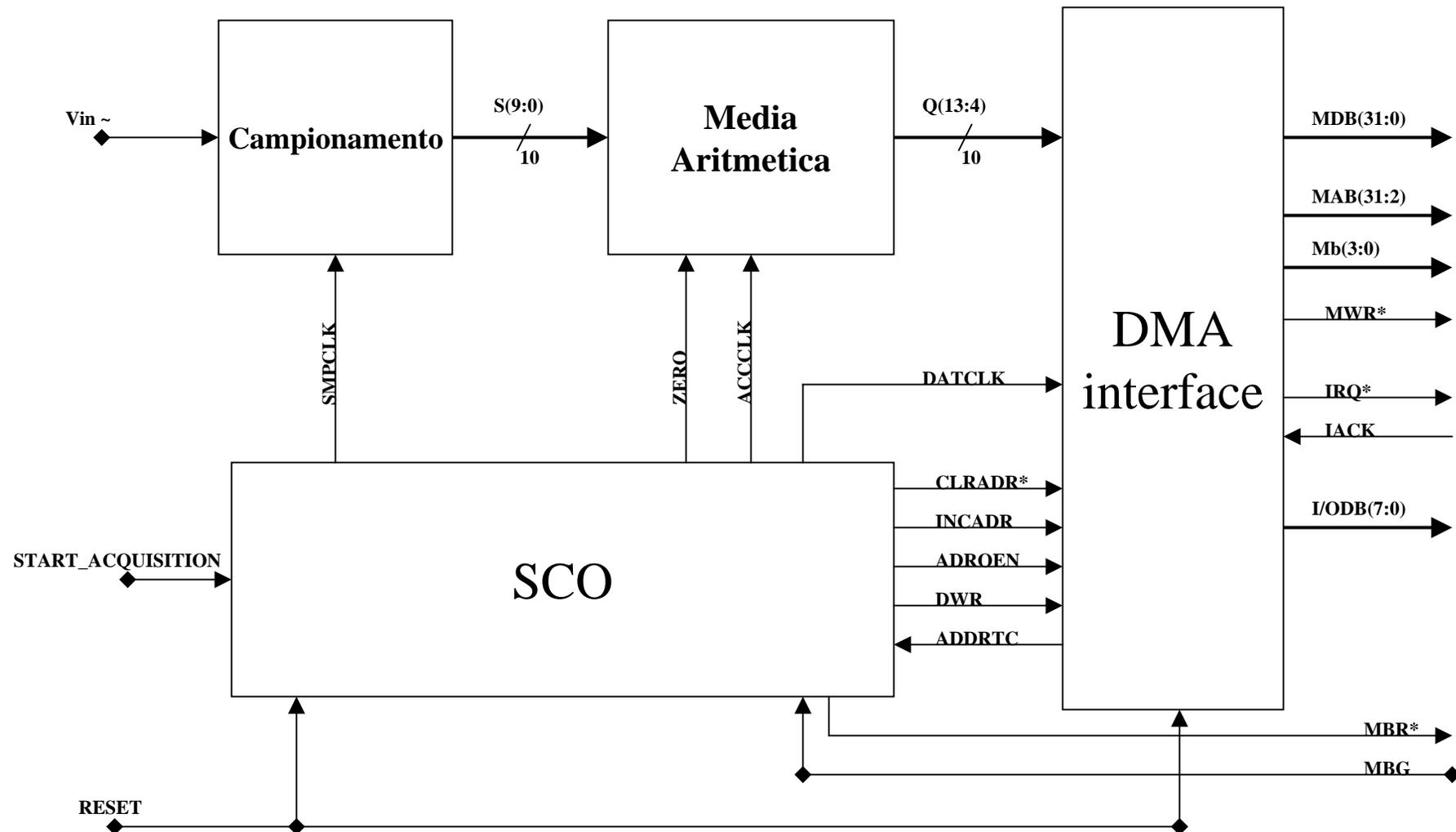
FASTDAS - DMA INTERFACE



FASTDAS - SCO



FASTDAS - Block Diagram



Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 02-07-1997

STUDENTE: _____

MATRICOLA: _____

Progetto:

VOXMSG, periferica PD32 per la generazione di messaggi vocali

Specifiche funzionali:

- VOXMSG contiene una memoria RAM da 64Kx8 bit mappata nello spazio di indirizzamento del PD32 a partire da FFF0000;
- il PD32 carica nella memoria RAM un messaggio vocale digitalizzato sotto forma di una sequenza di campioni S a 12 bit allocati in coppie di byte consecutivi nel formato 8 bit meno significativi, 4 bit più significativi:
Indirizzo N: [S₇S₆ ... S₀]
Indirizzo N+1: [0000S₁₁S₁₀S₉S₈]
e programma la lunghezza della sequenza ($\leq 32K$ campioni) in un registro di VOXMSG;
- una volta attivato sotto comando del PD32, VOXMSG riproduce il messaggio memorizzato mediante un convertitore D/A a 12 bit aggiornato ad una frequenza di 8KHz;
- al termine della riproduzione VOXMSG genera un interrupt verso il PD32.

Specifiche dispositivi:

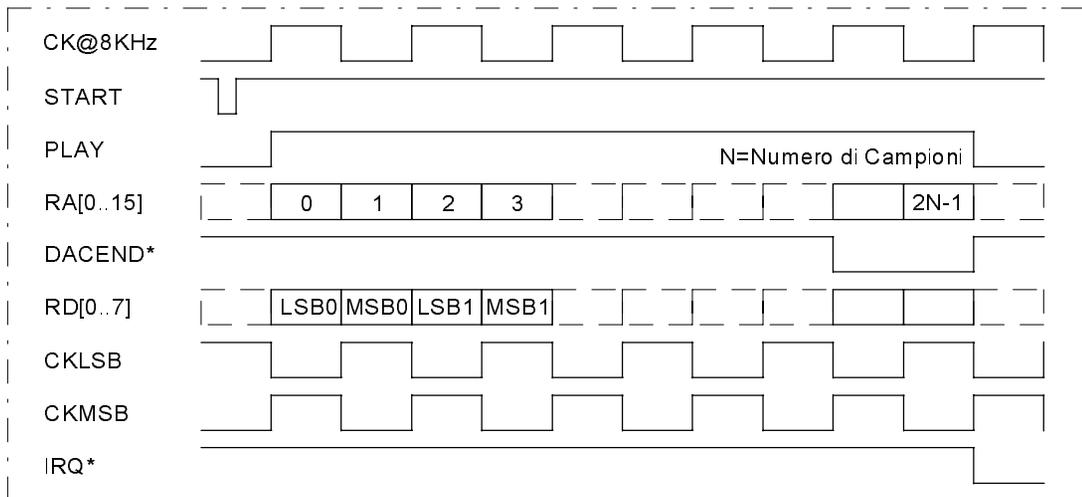
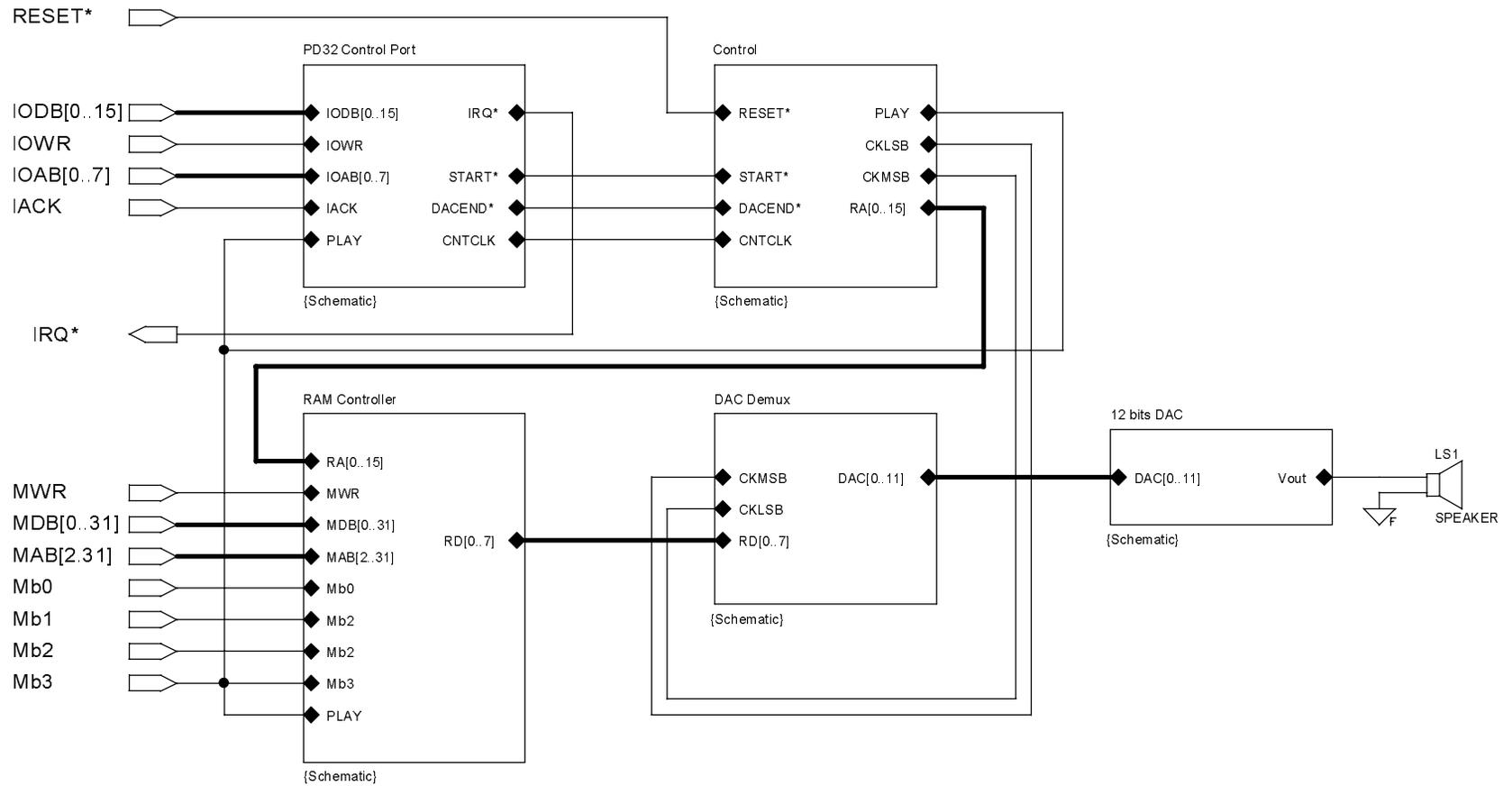
- il DAC a 12 bit non è dotato di registri dati interni

Note/suggerimenti:

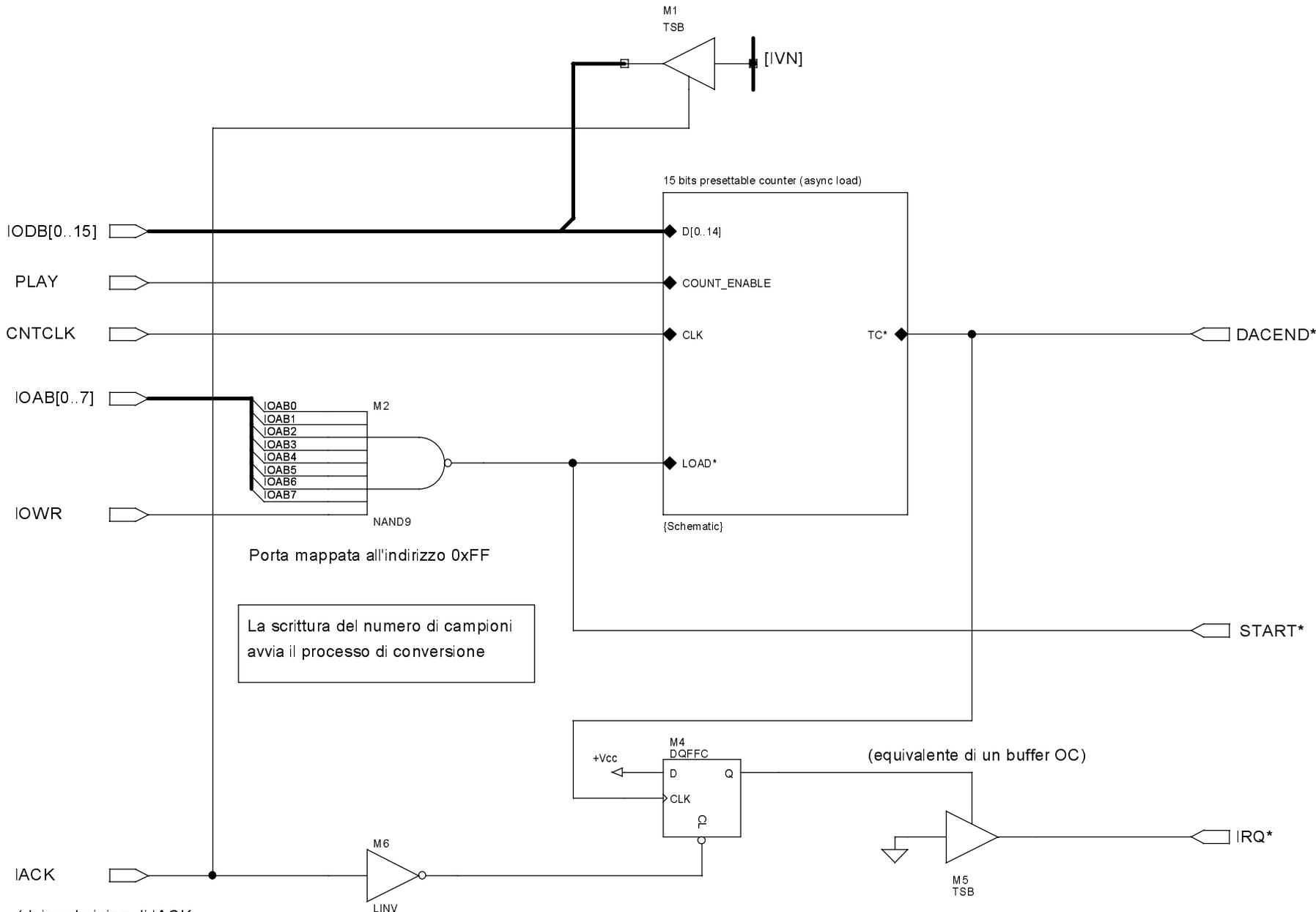
Per ottenere una corretta riproduzione del messaggio il DAC deve essere pilotato presentando contemporaneamente tutti i 12 bit di ingresso.

Si richiedono:

1. lo schema a blocchi funzionali di VOXMSG completo dei segnali di interfacciamento fra i blocchi
 2. il diagramma di timing dettagliato relativo al trasferimento della sequenza di campioni dalla RAM al DAC
 3. lo schema elettrico di VOXMSG
-



Title		
VOXMSG Block Diagram & Timing		
Size	Document Number	Rev
A4	1s970702	1
Date:	Tuesday, July 08, 1997	Sheet 1 of 4



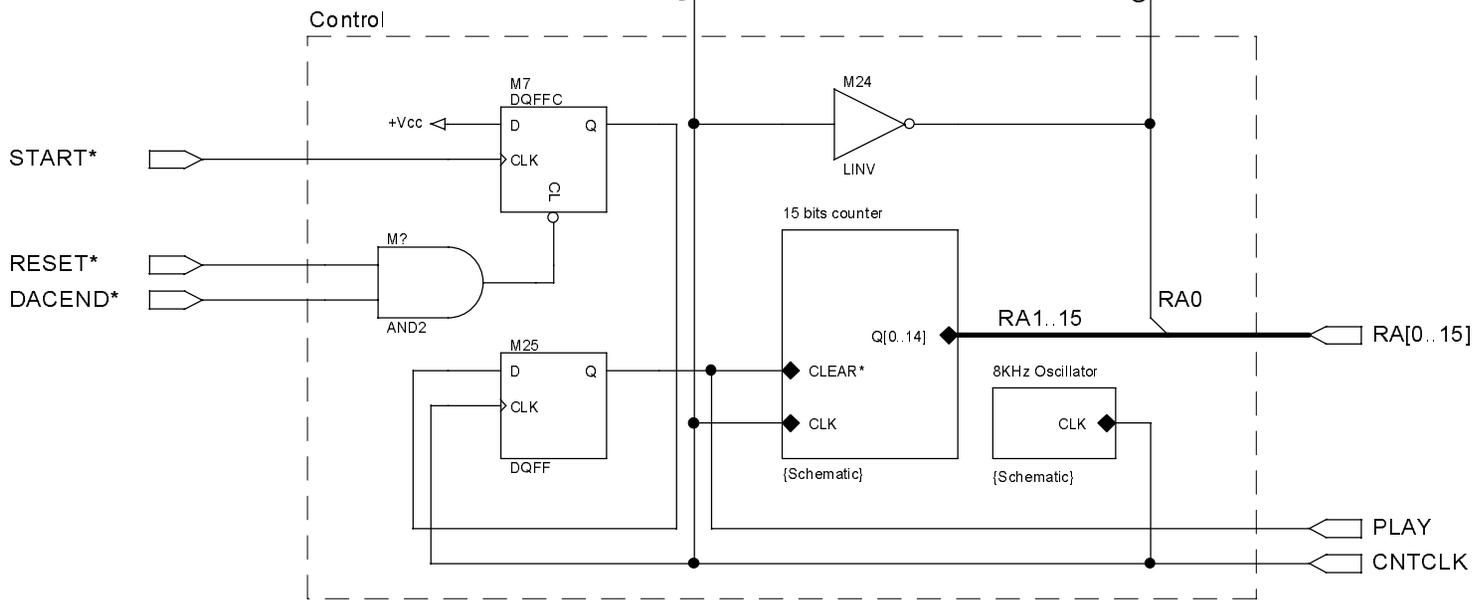
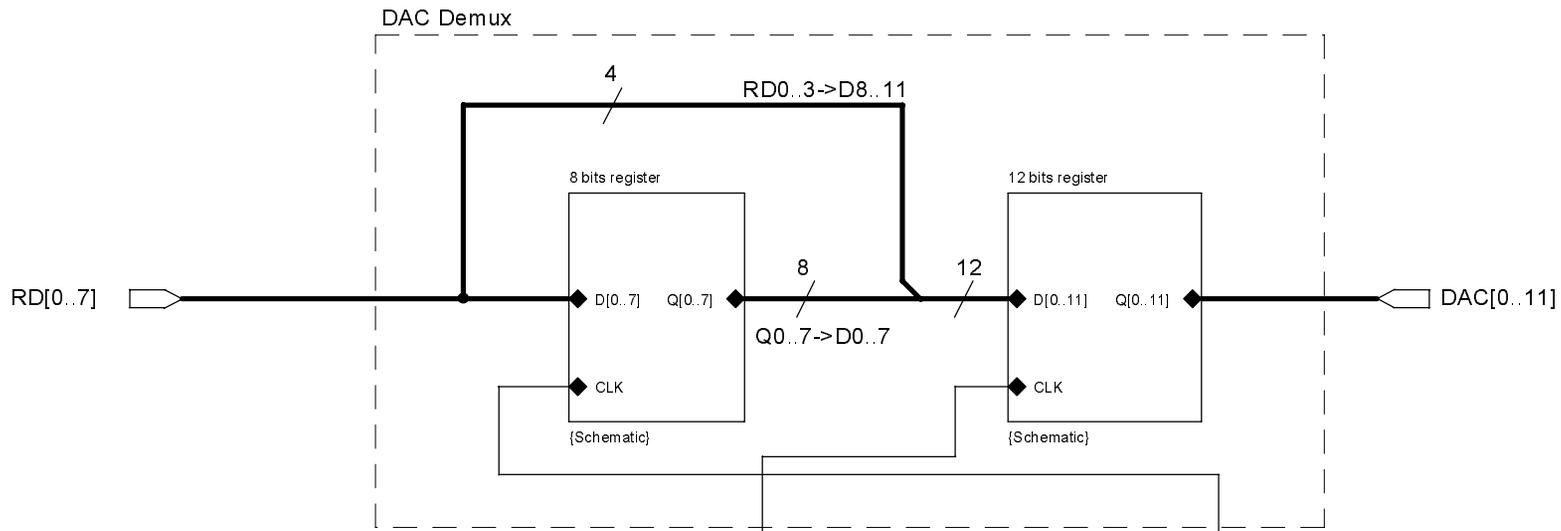
Porta mappata all'indirizzo 0xFF

La scrittura del numero di campioni avvia il processo di conversione

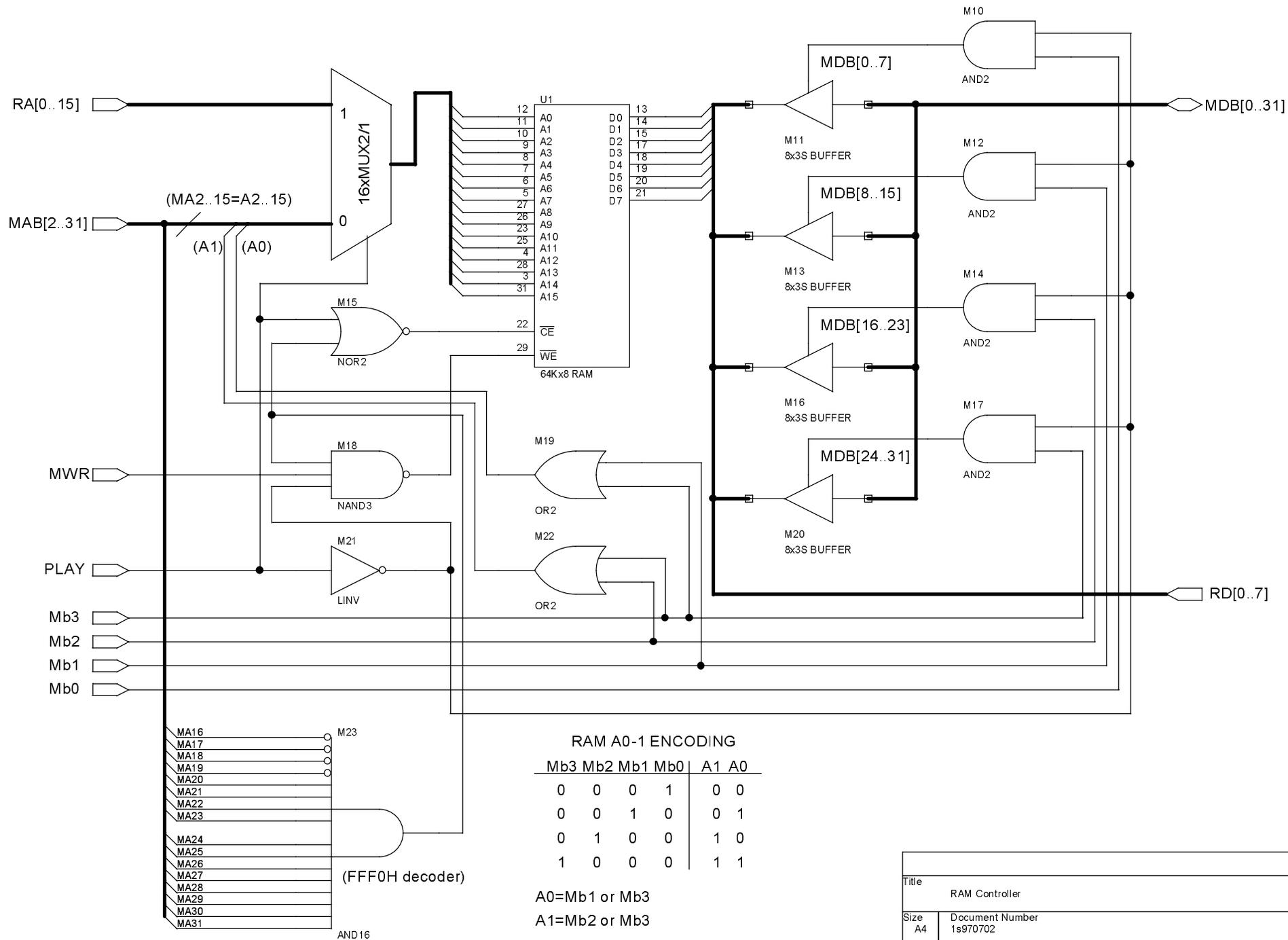
(daisy chaining di IACK implementato esternamente)

(equivalente di un buffer OC)

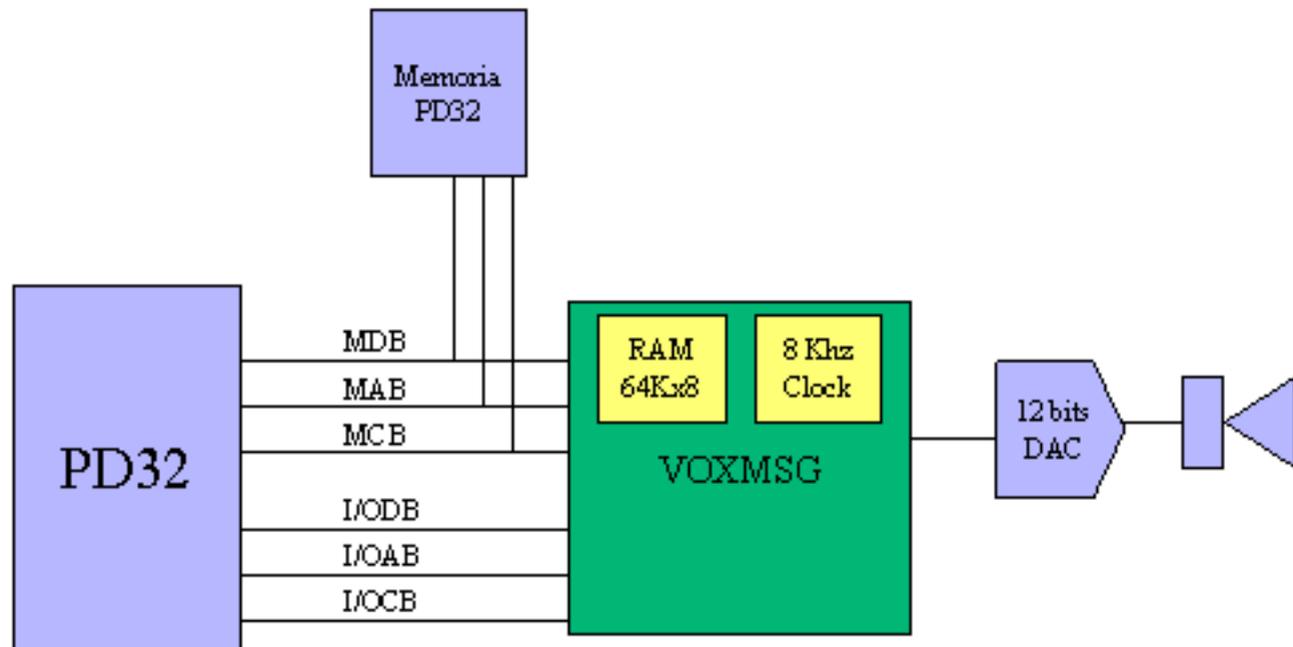
Title		
PD32 Control Port		
Size	Document Number	Rev
A4	1s970702	1
Date:	Tuesday, July 08, 1997	Sheet 2 of 4



Title		
Control & DAC Demux		
Size	Document Number	Rev
A4	1s970702	1
Date:	Tuesday, July 08, 1997	Sheet 3 of 4



Architettura progetto



Blocchi funzionali unita' VOXMSG

- **INTERFACCIA I/O BUS**

- PD32 => VOXMSG
 - Fornire il segnale di inizio riproduzione messaggio
 - Programmare la lunghezza della sequenza da riprodurre
- VOXMSG => PD32
 - Generazione di un interrupt alla fine della riproduzione

- **CONTROLLORE RAM INTERNA (64Kx8)**

- Gestione accessi PD32 in scrittura (memory mapped)
- Gestione accessi in lettura per riproduzione messaggio

- **INTERFACCIA con il DAC**

- Formattazione dei dati e presentazione ad 8 Khz

- **SCO**

- Gestione del Timing

Protocollo

PD32

VOXMSG

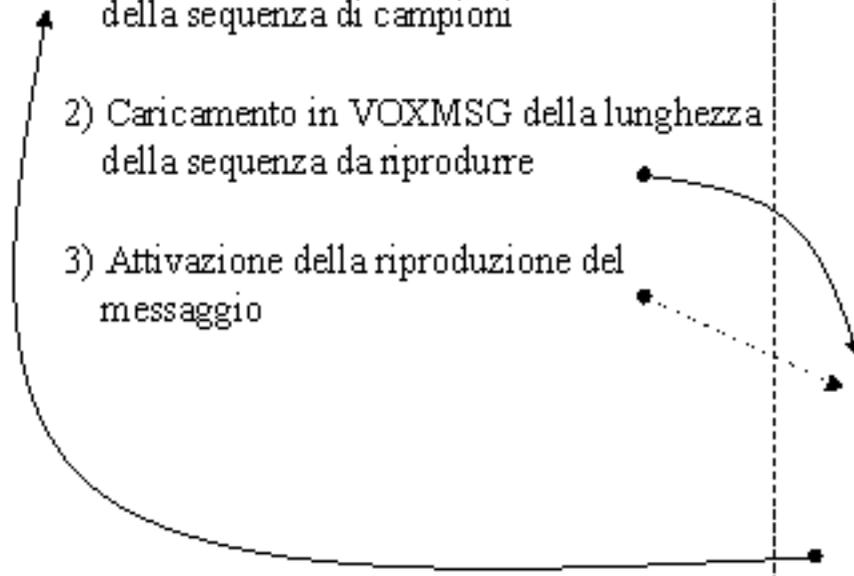
1) Caricamento nella RAM di VOXMSG della sequenza di campioni

2) Caricamento in VOXMSG della lunghezza della sequenza da riprodurre

3) Attivazione della riproduzione del messaggio

4) Preleva i dati dalla RAM e li presenta in ingresso la DAC in formato 12 bit alla frequenza di 8 KHz

5) Invia un Interrupt al PD32



Protocollo

PD32

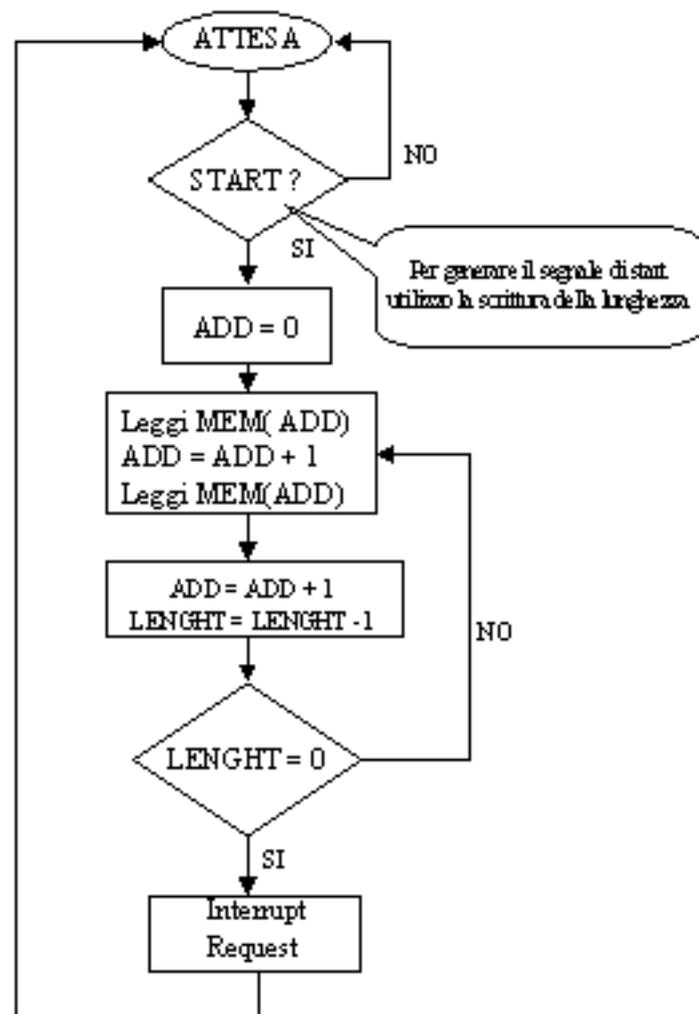
- 1) Caricamento nella RAM di VOXMSG della sequenza di campioni
- 2) Caricamento in VOXMSG della lunghezza della sequenza da riprodurre (*LENGHT*)
- 3) Attivazione della riproduzione del messaggio

Routine assembler

```
.....  
MOV 1 in registro R1  
MOV da memoria a registro R2 valore LENGHT  
hbell MOV da memoria a registro R3 R1-campione  
INC R1  
JUMP hbell se R1 < R2  
MOV R2 nel registro di VOXMSG  
.....
```

Protocollo

VOXMSG

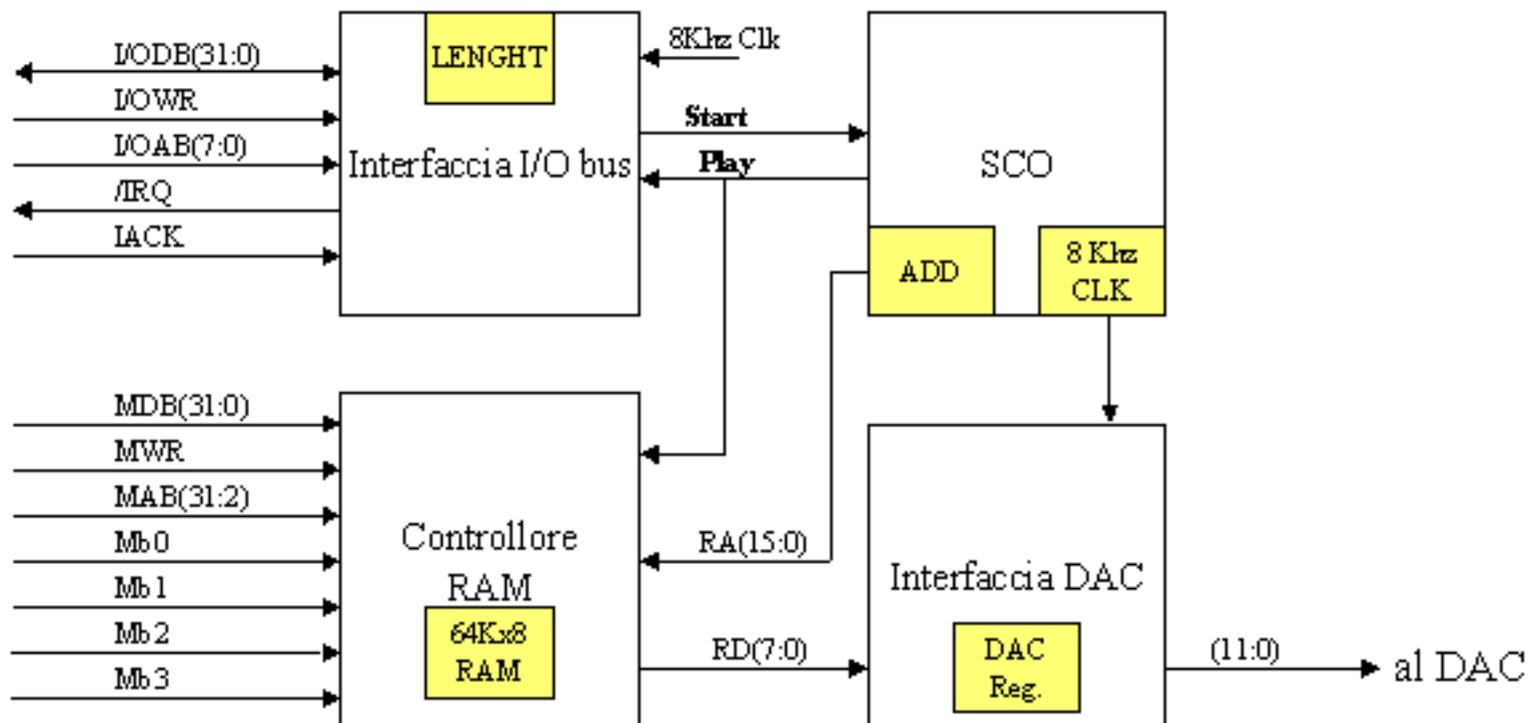


4) Preleva i dati dalla RAM e li presenta in ingresso la DAC in formato 12 bit alla frequenza di 8 KHz

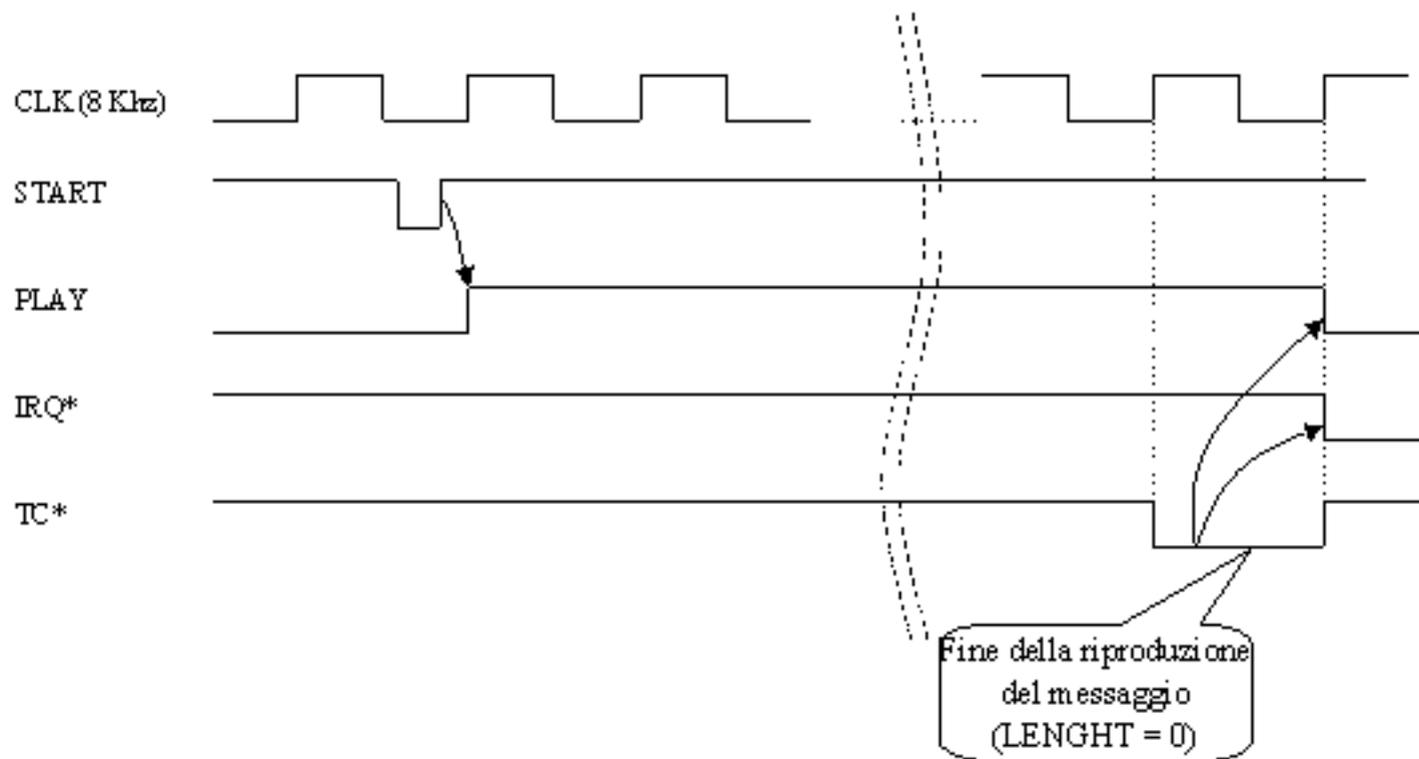
5) Invia un Interrupt al PD32

VOXMSG Schema a blocchi funzionale

- prima approssimazione -



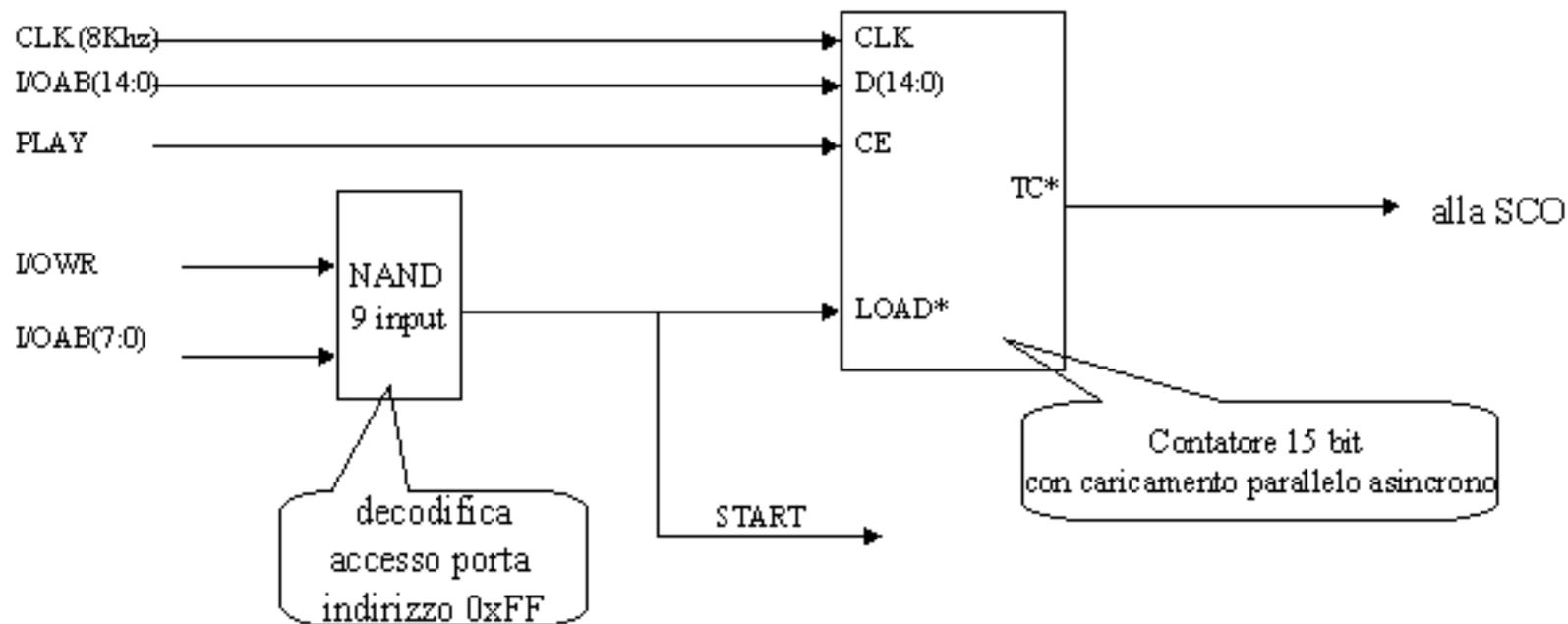
primo timing



INTERFACCIA I/O BUS

PD32 => VOXMSG

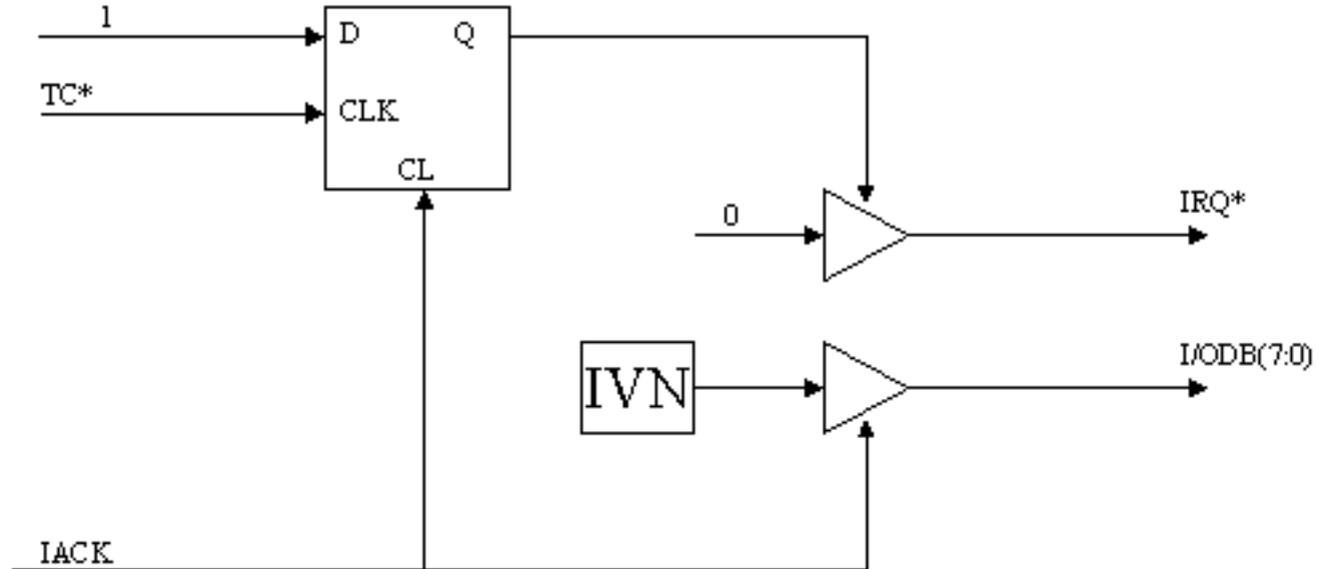
- Fornire il segnale di inizio riproduzione messaggio
- Programmare la lunghezza della sequenza da riprodurre



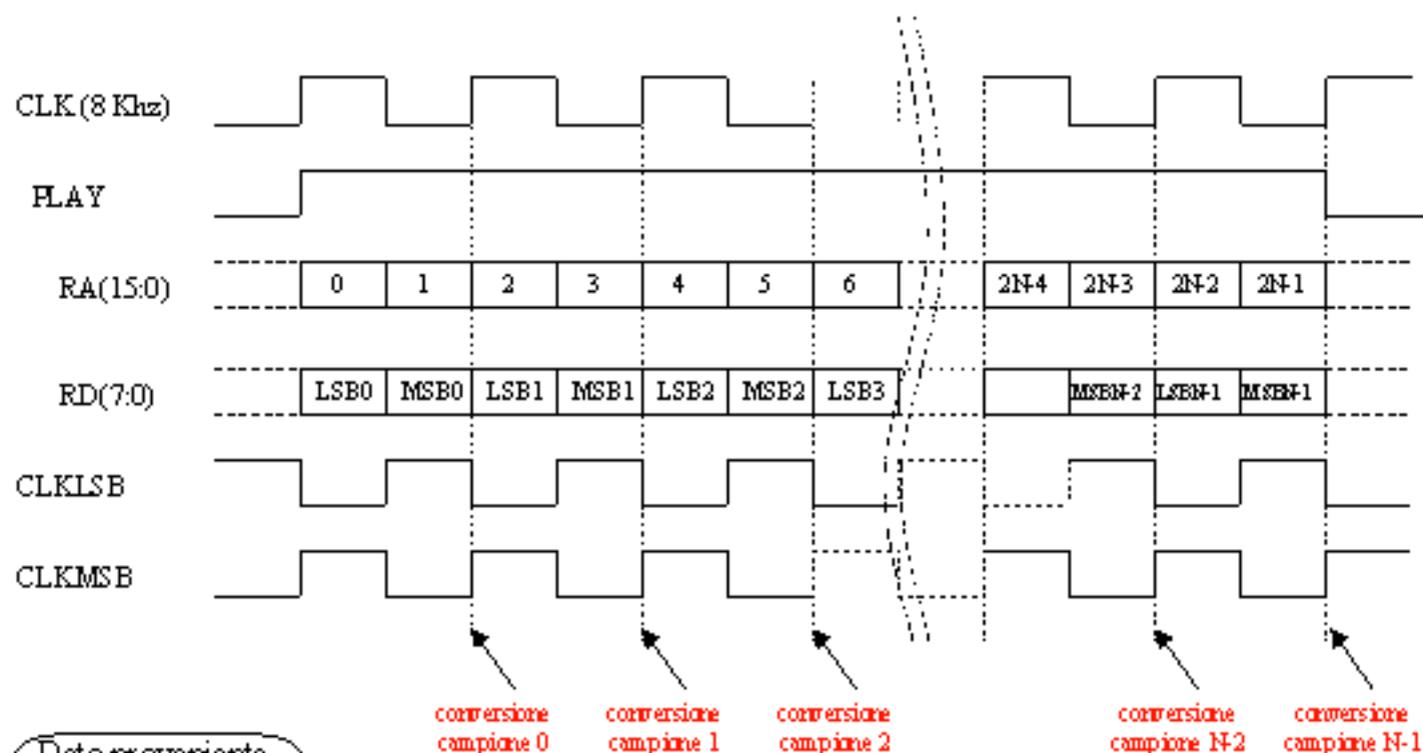
INTERFACCIA I/O BUS

VOXMSG => PD32

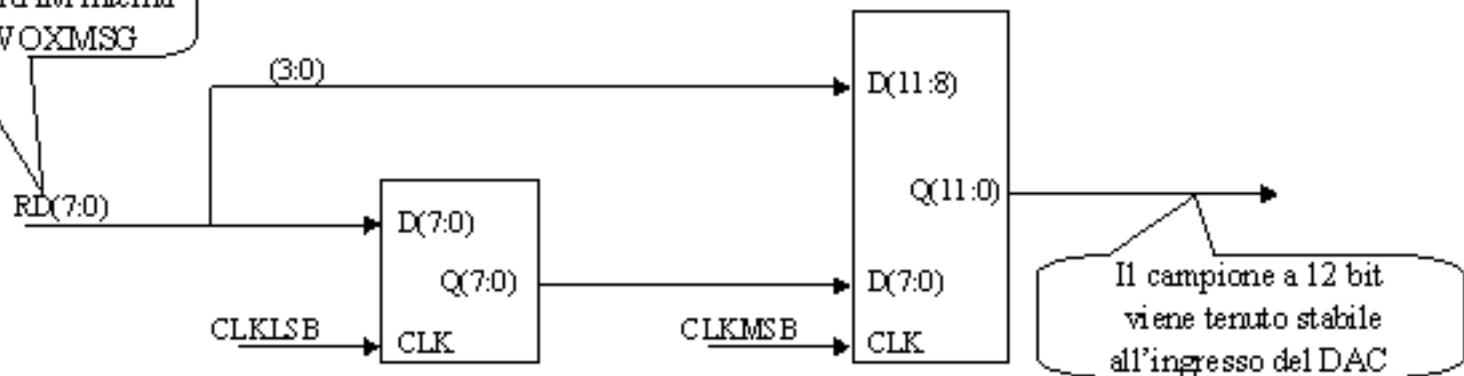
- Generazione di un interrupt alla fine della riproduzione



timing per interfaccia DAC



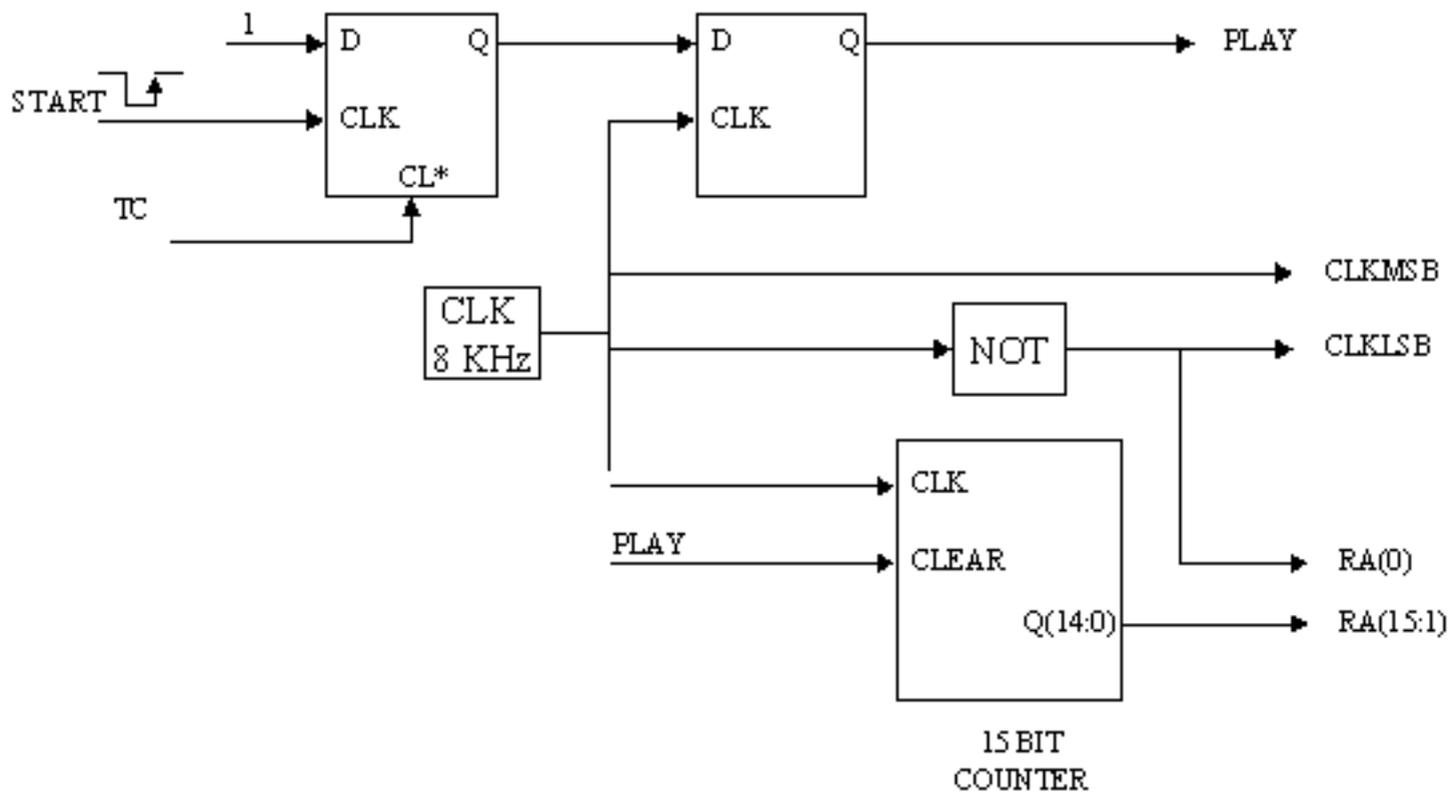
Dato proveniente
dalla RAM interna
al VOXMSG



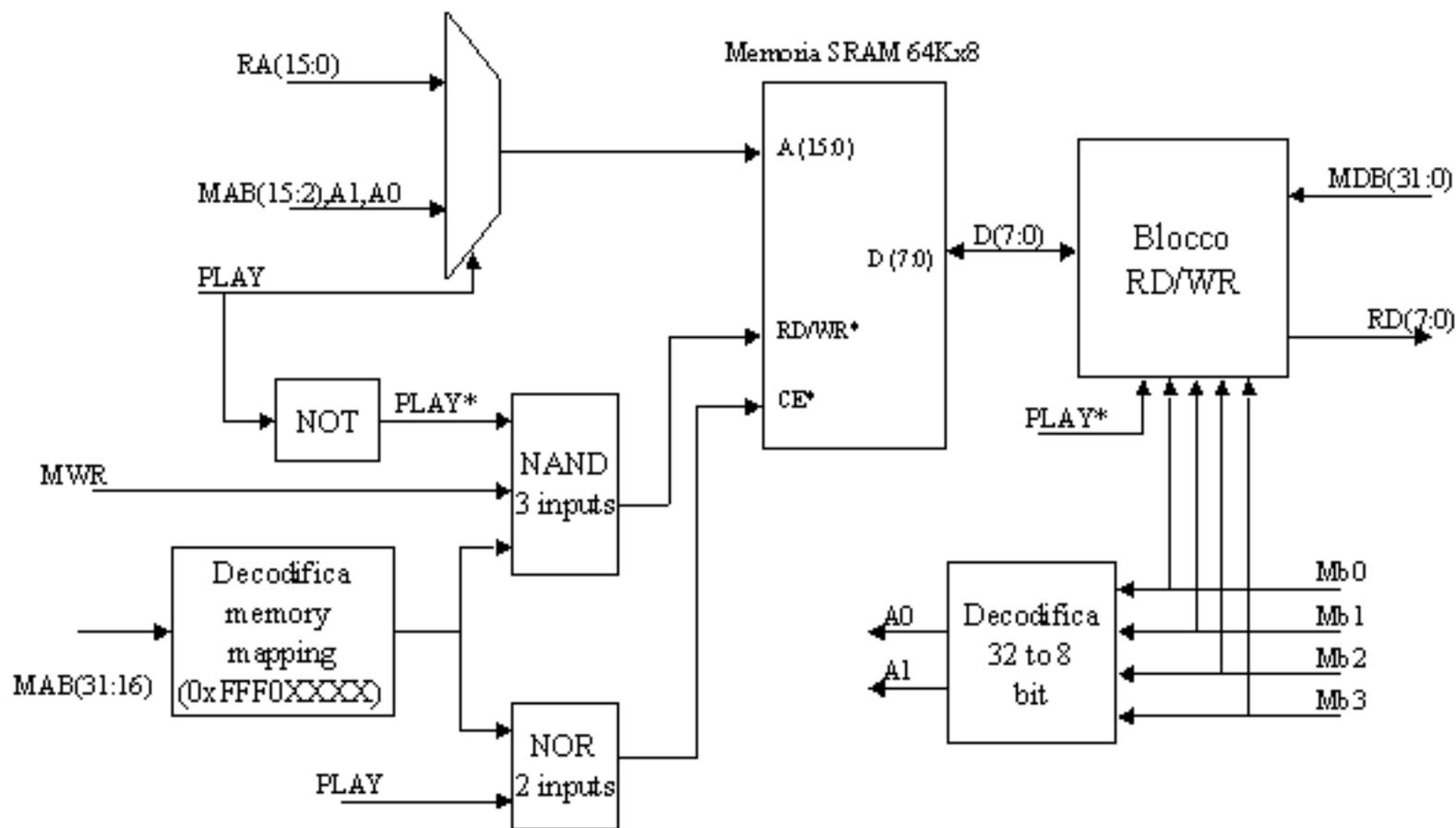
Il campione a 12 bit
viene tenuto stabile
all'ingresso del DAC

SCO

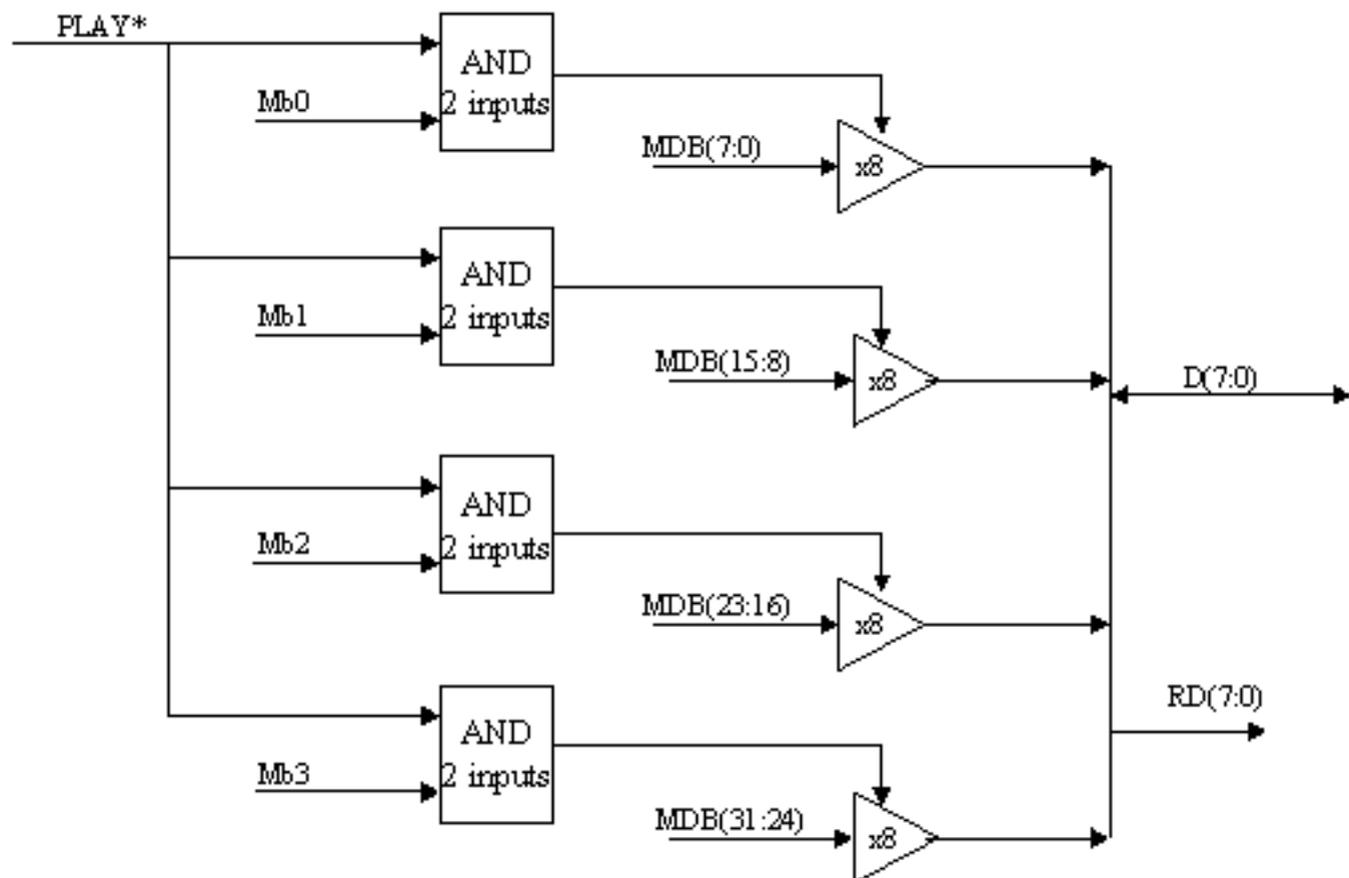
- Generazione dei clocks
- Gestione interazione segnali START/PLAY
- Generazione indirizzi RAM interna al VOXMSG



Controllore RAM

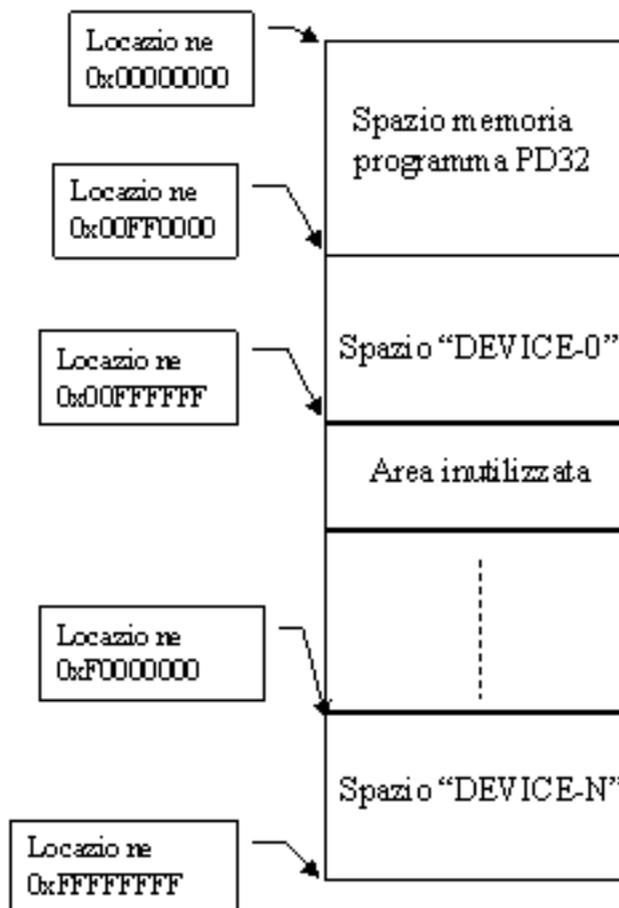


Blocco RD/WR

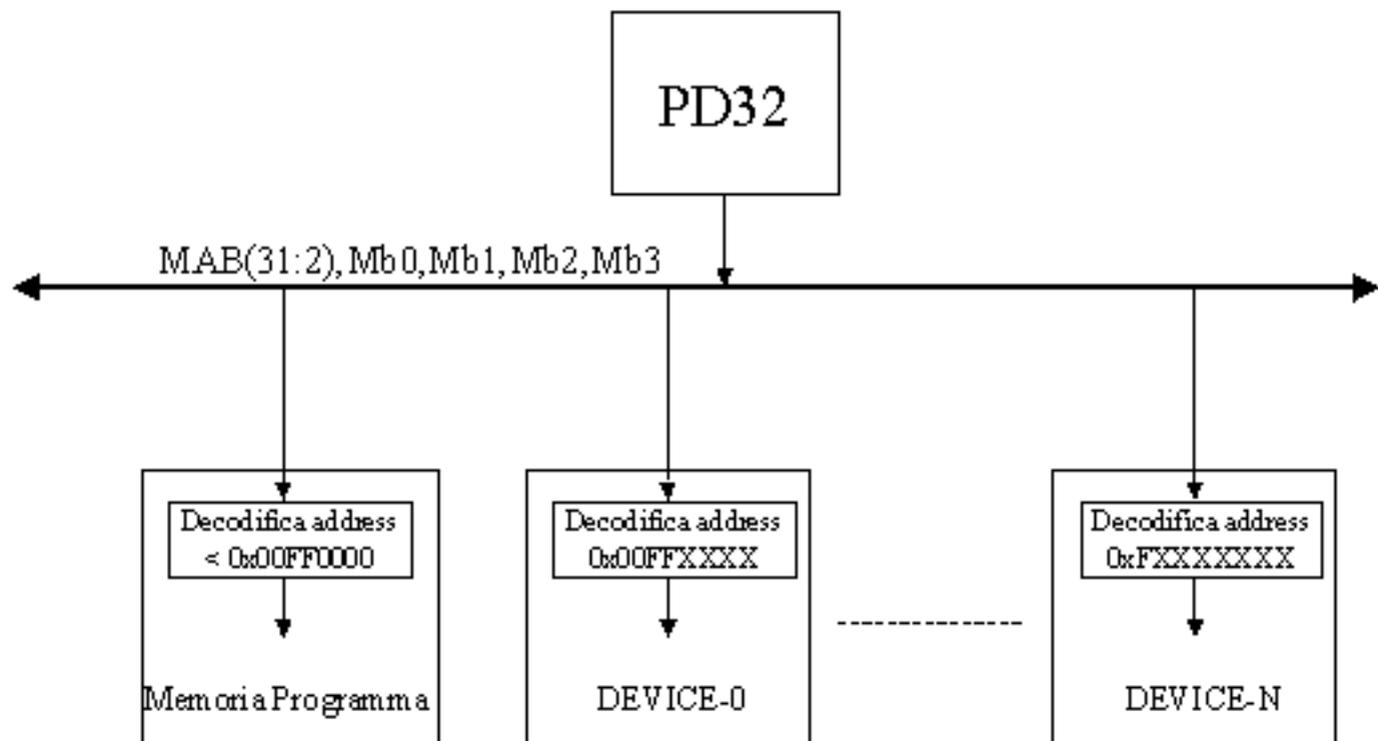


“Memory Mapping”

Spazio di indirizzamento PD32



Quando e' necessario si puo' utilizzare una parte dello spazio di indirizzamento del PD32 per “mappare”una periferica sul bus dati invece che su quello di I/O



Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 16-07-1997

STUDENTE: _____

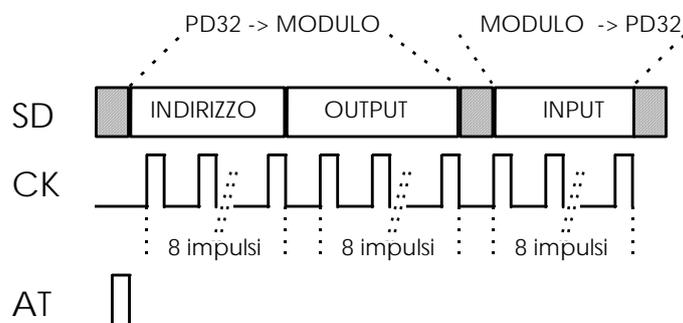
MATRICOLA: _____

Progetto:

Sistema FIELDMUX basato su PD32 per la gestione distribuita di I/O digitali

Specifiche funzionali:

- FIELDMUX è dedicato alla gestione di linee di I/O digitale distribuite in un edificio mediante 256 moduli periferici indirizzabili, ognuno dei quali gestisce 8 linee di input e 8 di output
- i moduli periferici sono controllati dal PD32 centrale mediante un bus con 3 linee di comunicazione SD (Serial Data), CK (Clock) e AT (Attention) completamente gestite via software dal microprocessore; il protocollo di colloquio con i moduli periferici si basa sulla trasmissione seriale di dati sulla linea SD sincronizzata dal clock CK
- il protocollo di colloquio è gestito dal PD32 come segue:
 1. il PD32 inizia la comunicazione con il generico modulo generando un impulso sulla linea AT
 2. il PD32 invia su SD l'indirizzo ad 8 bit del modulo da controllare
 3. il PD32 invia su SD gli 8 bit di output che il modulo indirizzato memorizzerà in un suo registro interno
 4. il modulo invia su SD gli 8 bit delle linee di input locali (in sincronismo con il clock CK generato dal PD32)



Specifiche dispositivi:

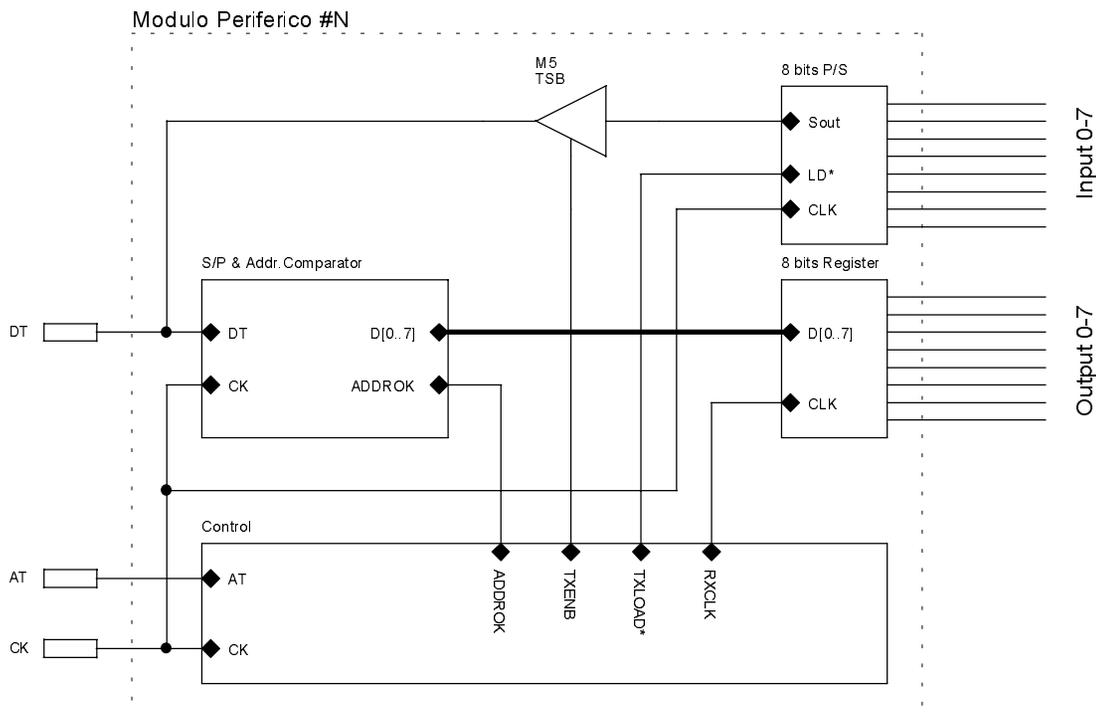
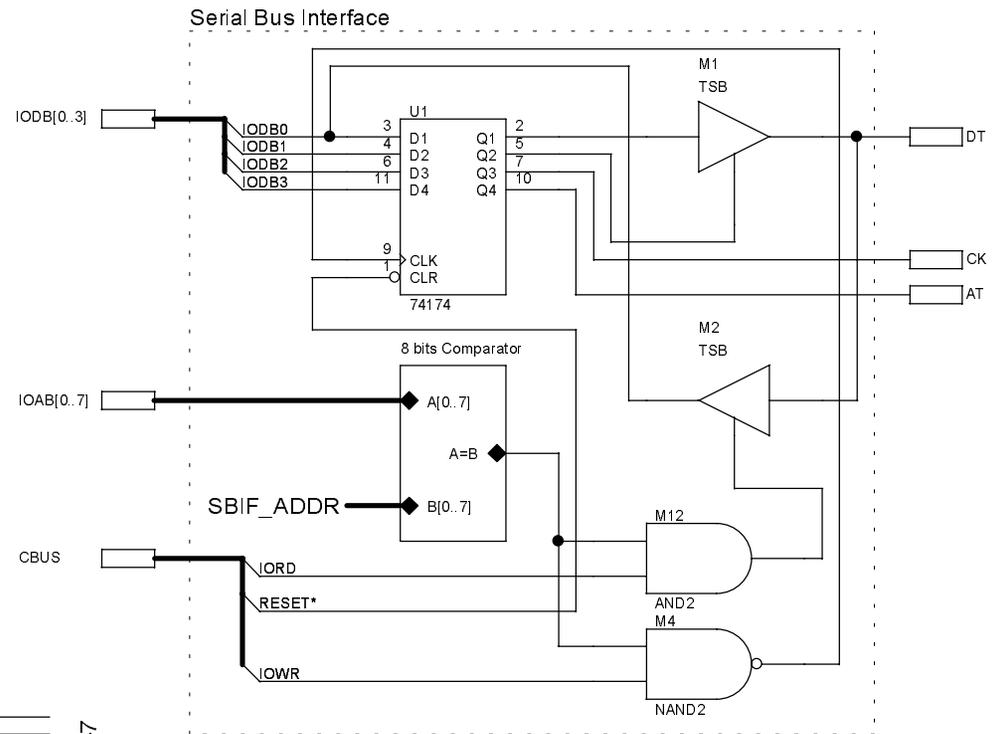
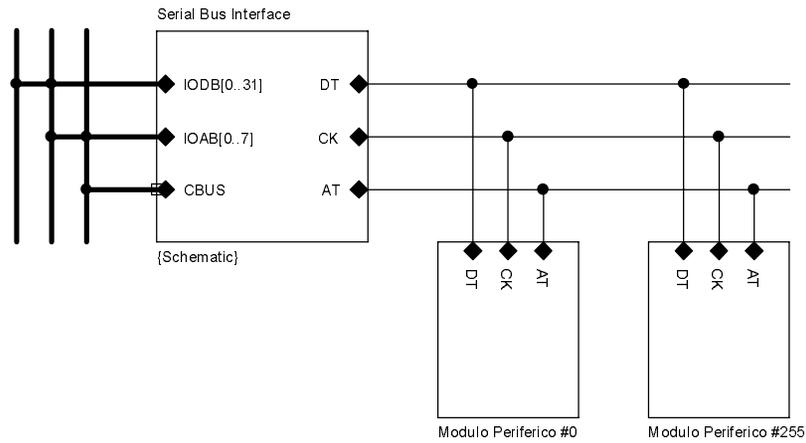
Il modulo periferico deve essere progettato con un controllore LLC (in logica cablata)

Note/suggerimenti:

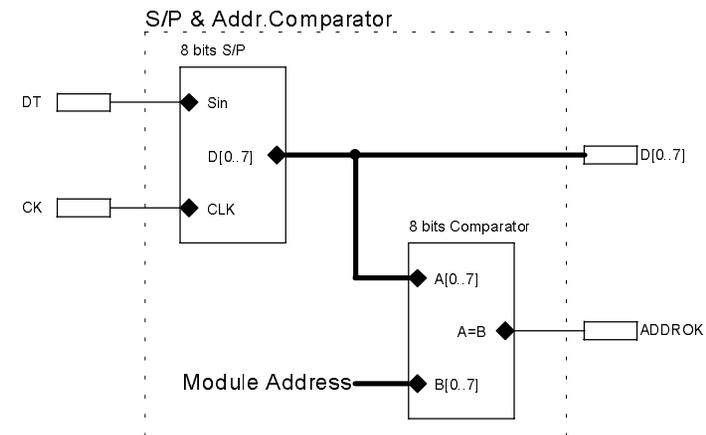
Ridurre al minimo indispensabile l'hardware dell'interfaccia PD32 con il bus seriale

Si richiedono:

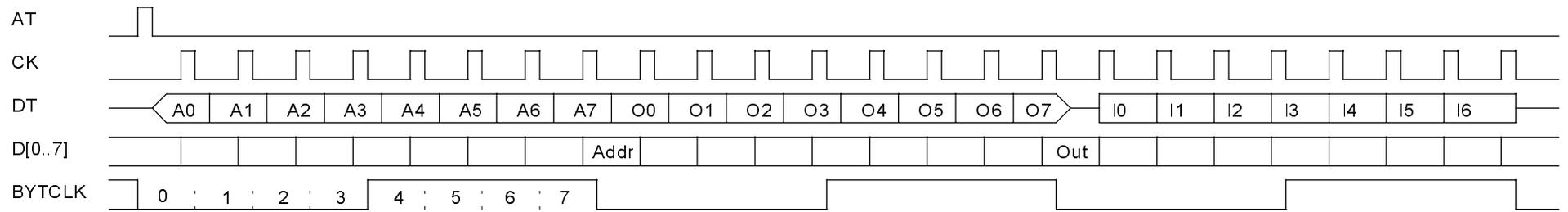
1. lo schema a blocchi funzionali del modulo di I/O periferico completo dei segnali di interfaccia
 2. il diagramma di timing relativo alla gestione della comunicazione con il PD32
 3. lo schema elettrico del modulo periferico e dell'interfaccia PD32 verso il bus di comunicazione
 4. la codifica in assembler PD32 della subroutine di gestione del protocollo specificato; la subroutine riceve in R0 l'indirizzo del modulo periferico, in R1 il byte di output e riporta in R0 il byte di input
-



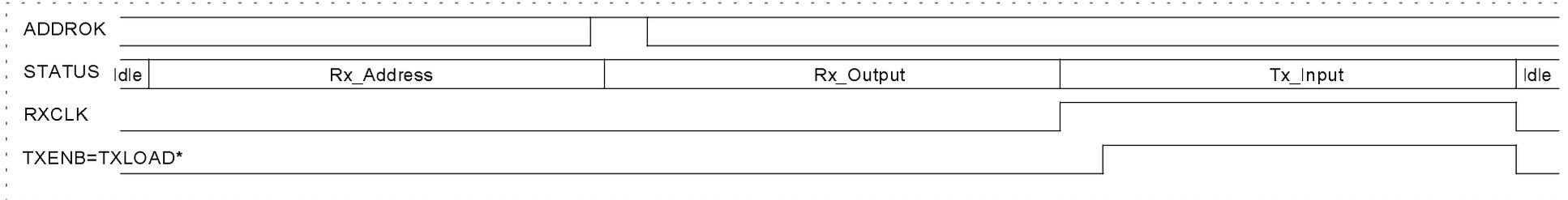
Input 0-7
Output 0-7



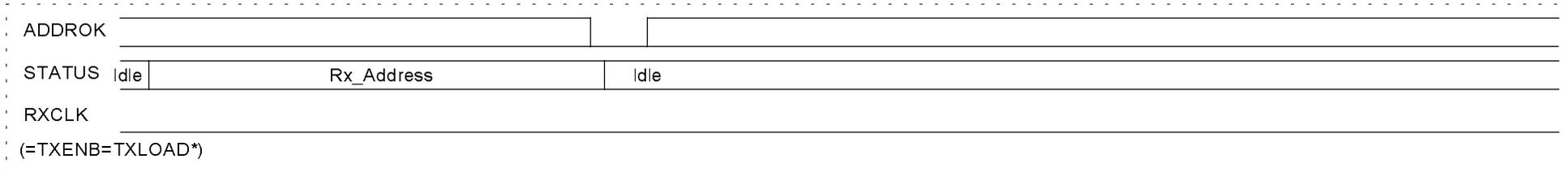
Title		
FIELDMUX: Block Diagram, S/P & Addr. Comparator, Serial Bus Interface		
Size	Document Number	Rev
A4	1s970716/A.PUGNALONI	1.1
Date:	Wednesday, July 23, 1997	Sheet 1 of 3



@ Addr=Module Address



@ Addr<>Module Address



Il modulo periferico è governato da una macchina LLC a 4 stati (Idle, Rx_Address, Rx_Output, Tx_Input)

La macchina è sincronizzata sul fronte di discesa di BYTCLK ed è inizializzata allo stato Rx_Address dal segnale AT

Risincronizza TXENB/TXLOAD* sul fronte di salita di CK successivo come richiesto dal timing
 TXENB/TXLOAD* tornano a 0 alla disattivazione di RXCLKEN (CL*=RXCLKEN)

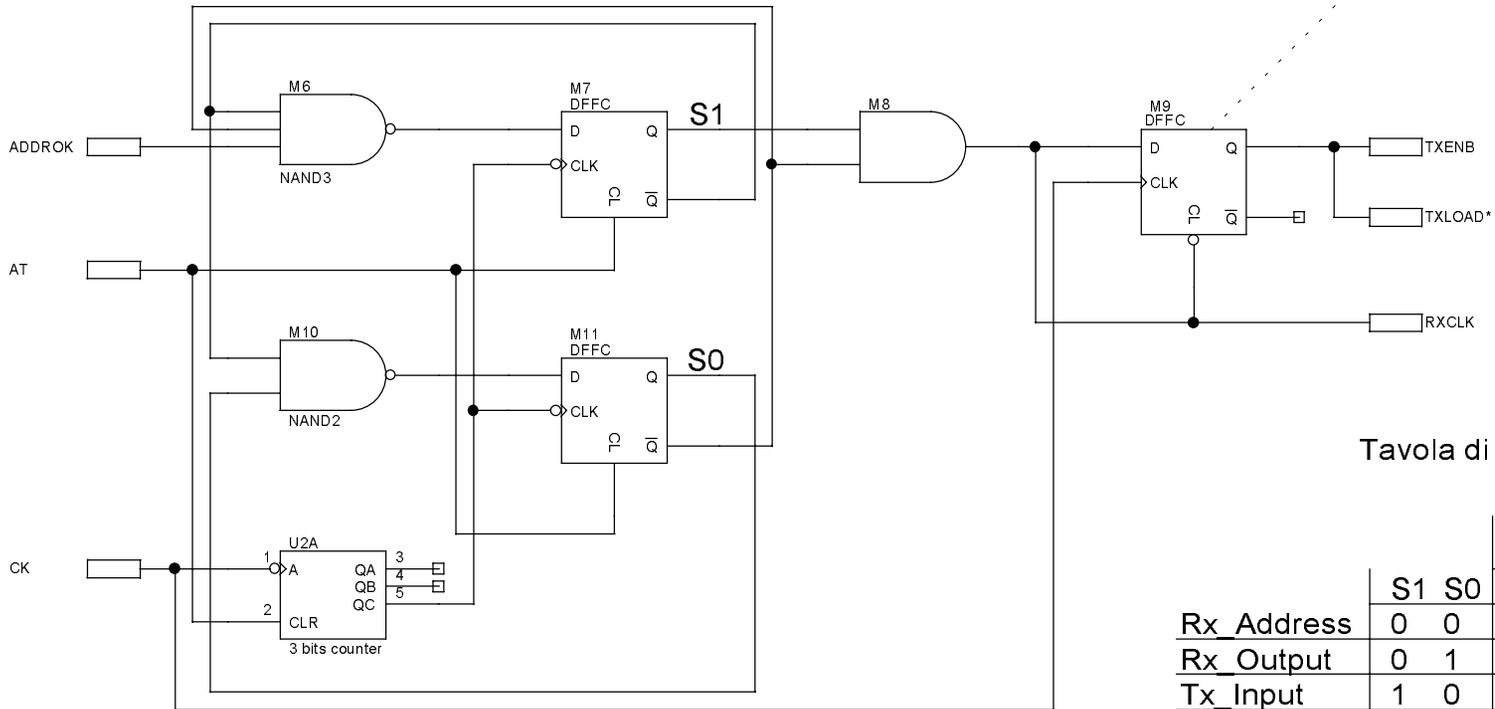


Tavola di flusso del controllore:

	S1 S0		ADDR0K		RXCLK
			0	1	
Rx_Address	0	0	11	01	0
Rx_Output	0	1	10	10	0
Tx_Input	1	0	11	11	1
Idle	1	1	11	11	0

$$RXCLK = S1 \ S0'$$

$$S1 = (S1' \ S0' \ ADDR0K)'$$

$$S0 = (S1' \ S0)'$$

Title		
FIELDMUX: Control Block		
Size	Document Number	Rev
A4	1s970716/A.PUGNALONI	1.1
Date:	Wednesday, July 23, 1997	Sheet 3 of 3

Driver di gestione dei dispositivi periferici FIELDMUX

```
SBIF_ADDR      EQU      xxxx          ; indirizzo assegnato alla
                                           ; Serial Bus Interface

;*****
;   Nome:          PollModule
;   Descrizione:   Questa subroutine interroga un modulo periferico
;   Parametri Input: R0: indirizzo del modulo
;                 R1: byte di output da trasmettere
;   Parametri Output: R0: byte di input ricevuto dal modulo
;   Registri usati: R3,R4,R5
;   Note:         TransmitR3 trasmette il byte passato in R3
;                 ReceiveR0 riceve il byte di input trasmesso dal modulo
;                 indirizzato
;*****

PollModule:     JSR      SendATN         ; genera ATtention
                MOVVB   R0,R3          ; trasmetti l'indirizzo del modulo
                JSR      TransmitR3
                MOVVB   R1,R3          ; trasmetti il byte di output
                JSR      TransmitR3
                JSR      ReceiveR0     ; ricevi il byte di input
                RET

;*****
;   Nome:          TransmitR3
;   Descrizione:   Questa subroutine trasmette un byte ricevuto in R3
;   Parametri Input: R3: byte da trasmettere
;   Parametri Output: -
;   Registri usati: R4: contatore di bit
;                 R5: registro di lavoro
;   Note:         il contenuto di R3 viene perso
;*****

TransmitR3:     MOVVB   #8,R4          ; inizializza il contatore di bit
TxLoop:        MOVVB   R3,R5          ; passa R3 in R5
                ANDB   #1,R5          ; estrai LSB di R5
                ORB    #2,R5          ; attiva Buffer 3S su linea DT
                OUTB   R5,SBIF_ADDR   ; Bit#n->DT
                JSR    Delay          ; attendi stabilizzazione dei segnali
                ORB    #4,R5          ; attiva stato CK=1
                OUTB   R5,SBIF_ADDR   ; Bit#n->DT 1->CK
                JSR    Delay          ; attendi stabilizzazione dei segnali
                XORB   #4,R5          ; attiva stato CK=0
                OUTB   R5,SBIF_ADDR   ; Bit#n->DT 0->CK
                JSR    Delay          ; attendi stabilizzazione dei segnali
                LSRB   1,R3           ; shift a destra di R3
                SUBB   #1,R4          ; decrementa R4
                JNZ    TxLoop         ; loop se R4>0
                XORB   R5,R5          ; azzera R5
                OUTB   R5,SBIF_ADDR   ; disattiva Buffer 3S su linea DT
                RET
```

```

;*****
;   Nome:                ReceiveR0
;   Descrizione:        Questa subroutine ricostruisce in R0 un byte ricevuto
;                       dalla linea DT
;   Parametri Input:    -
;   Parametri Output:   R0: byte ricevuto
;   Registri usati:    R4: contatore di bit
;                       R5: registro di lavoro
;   Note:               -
;*****

ReceiveR0:      MOVB #8,R4          ; inizializza il contatore di bit
RxLoop:        MOVB #4,R5          ; attiva stato CK=1
               OUTB R5,SBIF_ADDR ; 1->CK
               JSR Delay          ; attendi stabilizzazione dei segnali
               INB SBIF_ADDR,R5   ; DT->R5 LSB
               LSRB R5           ; R5 LSB->Carry
               RCLB R0           ; Carry->R0 LSB
               XORB R5,R5        ; attiva stato CK=0
               OUTB R5,SBIF_ADDR ; 0->CK
               JSR Delay          ; attendi stabilizzazione dei segnali
               SUBB #1,R4        ; decrementa R4
               JNZ RxLoop        ; loop se R4>0
               RET

;*****
;   Nome:                SendATN
;   Descrizione:        Questa subroutine ricostruisce genera l'impulso di
;                       Attention
;   Parametri Input:    -
;   Parametri Output:   -
;   Registri usati:    R5: registro di lavoro
;   Note:               -
;*****

SendATN:       MOVB #8,R5          ; attiva stato AT=1
               OUTB R5,SBIF_ADDR ; 1->AT
               XORB R5,R5        ; attiva stato AT=0
               OUTB R5,SBIF_ADDR ; 0->AT
               JSR Delay          ; attendi stabilizzazione dei segnali
               RET

;*****
;   Nome:                Delay
;   Descrizione:        Questa subroutine genera un ritardo temporale ciclando
;                       per un certo numero di iterazioni
;   Parametri Input:    -
;   Parametri Output:   -
;   Registri usati:    R4: contatore di iterazioni (salvato)
;   Note:               -
;*****

Cycles          EQU xxxx          ; costante a 16 bit

Delay:          PUSH R4            ; salva R4
DlyLoop:       MOVW #Cycles,R4    ; inizializza il contatore di loop
               SUBW #1,R4         ; decrementa R4
               JNZ DlyLoop        ; loop se R4>0
               POP R4             ; ripristina R4
               RET

```

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 17-09-1997

STUDENTE: _____

MATRICOLA: _____

Progetto:

Pannello grafico luminoso LEDPAN

Specifiche funzionali:

LEDPAN è un pannello luminoso per la visualizzazione di immagini animate controllato da un microprocessore PD32 e costituito da un certo numero di piastrelle elementari da 32x32 LED. Ogni piastrella da 1024 LED è dotata di una propria logica di controllo ed ha le seguenti caratteristiche:

- i LED sono organizzati in una matrice quadrata da 32 righe; la visualizzazione dell'immagine avviene mediante scansione sequenziale delle righe da 32 LED mantenendo accesa una sola riga alla volta per un periodo di 0.5ms ed ottenendo quindi il rinfresco dell'intera piastrella ogni 16ms;
- l'immagine visualizzata dalla piastrella è contenuta in una RAM locale da 32 parole di 32 bit scandita circolarmente dalla logica della piastrella la quale provvede inoltre alla generazione del segnale di accensione della riga estratta dalla memoria;
- il PD32 di controllo del LEDPAN accede alle RAM interne delle varie piastrelle tramite il memory data bus in modo da poter leggere/scrivere le relative immagini elementari SENZA INTERFERIRE CON IL PROCESSO DI SCANSIONE/VISUALIZZAZIONE; la mappatura di ogni piastrella nello spazio di indirizzamento del PD32 deve essere configurabile a passi di 32 LongWord (128 byte) nel range 80000000H-8000FF80H;
- l'accesso del PD32 alla RAM locale della generica piastrella deve essere sincronizzato con le letture effettuate dalla logica locale mediante tecniche hardware (segnale di WAIT del PD32) in modo completamente trasparente al programma software del PD32.

Specifiche dispositivi:

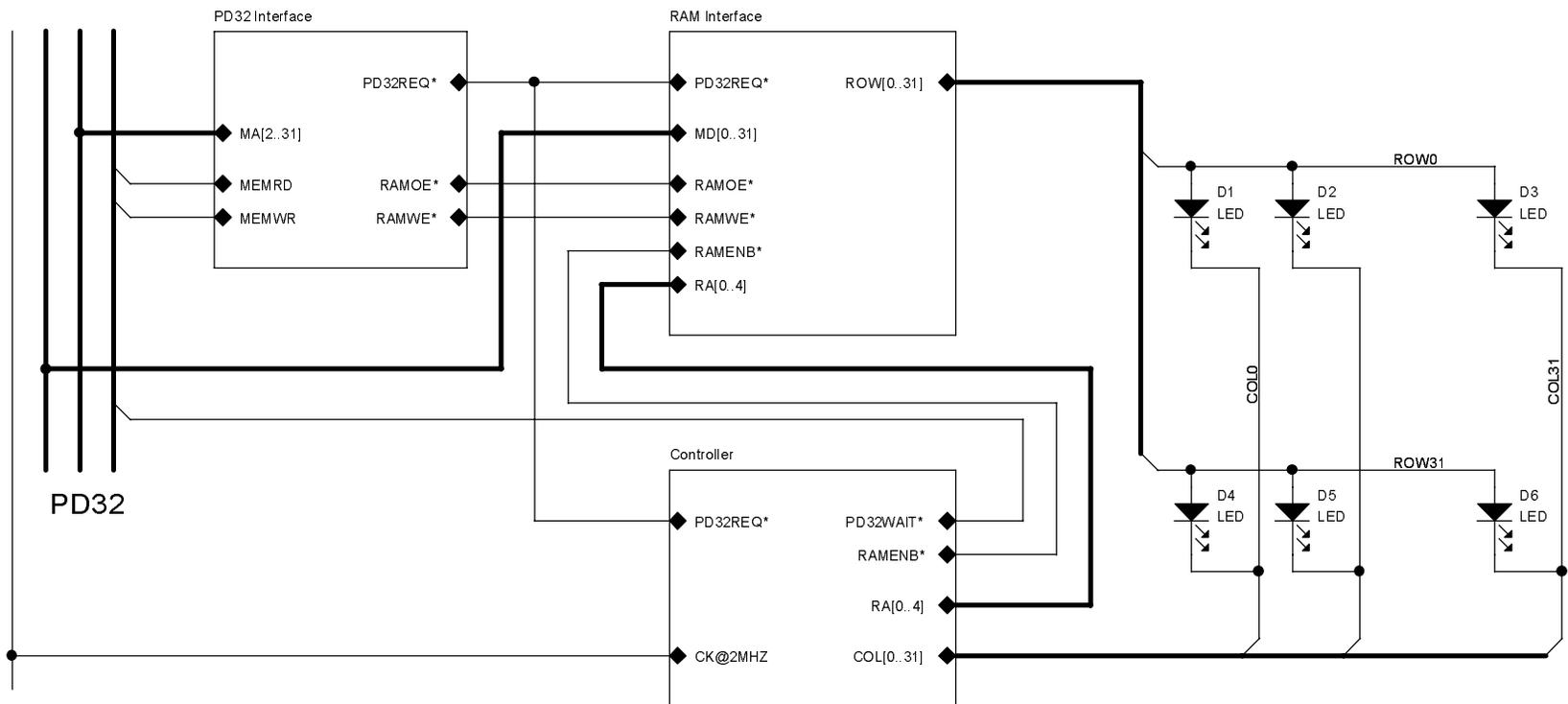
Tempo di accesso delle RAM locali = 200ns, tempi di propagazione di tutti gli altri dispositivi = 10ns, durata ciclo PD32 di R/W della memoria = 500ns

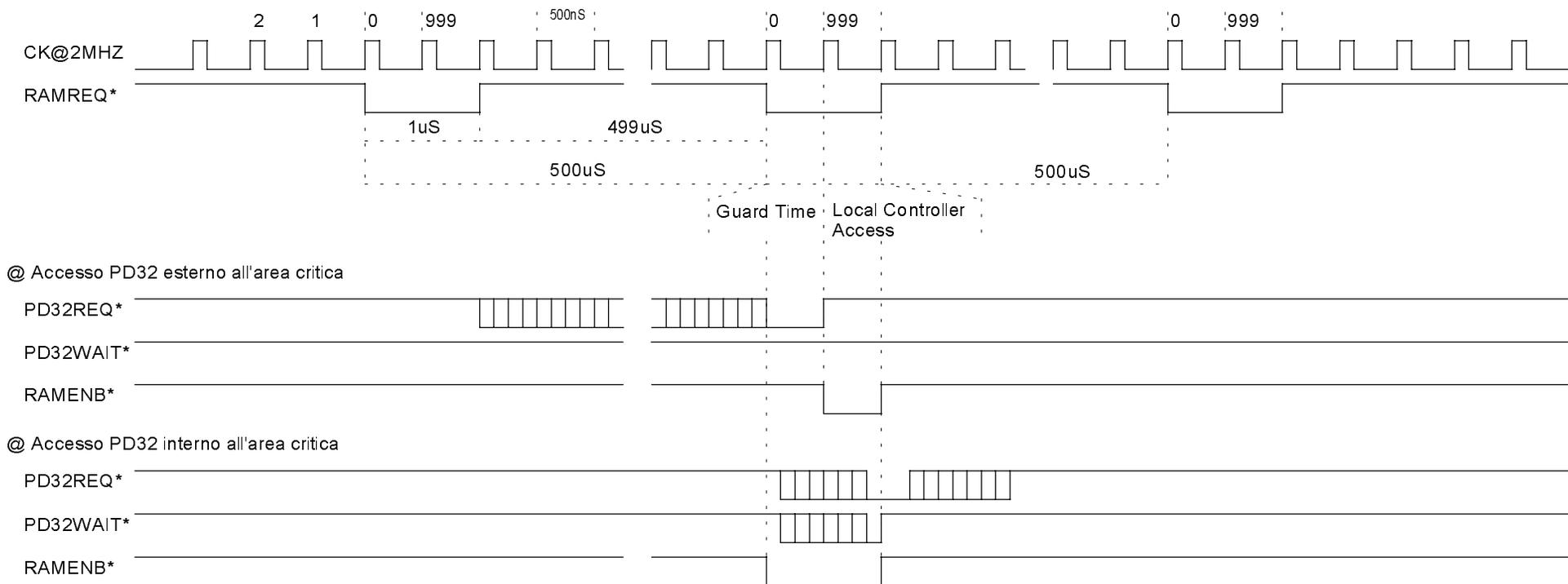
Note/suggerimenti:

Per condividere l'accesso alla RAM della generica piastrella si suddivide opportunamente il periodo di scansione disponibile (0.5ms) fra PD32 e logica locale in modo da non penalizzare eccessivamente il throughput di trasferimento dati PD32↔RAM locale.

Si richiedono:

1. lo schema a blocchi funzionali di LEDPAN e della generica piastrella con l'indicazione dei segnali necessari all'interfacciamento delle piastrelle con il PD32 di controllo
 2. il diagramma di temporizzazione relativo all'accesso condiviso alla RAM locale delle piastrelle che evidenzia la risoluzione di un conflitto di accesso fra PD32 e logica locale
 3. lo schema elettrico della generica piastrella
-





Risoluzione dei conflitti di accesso alla RAM locale della piastrina:

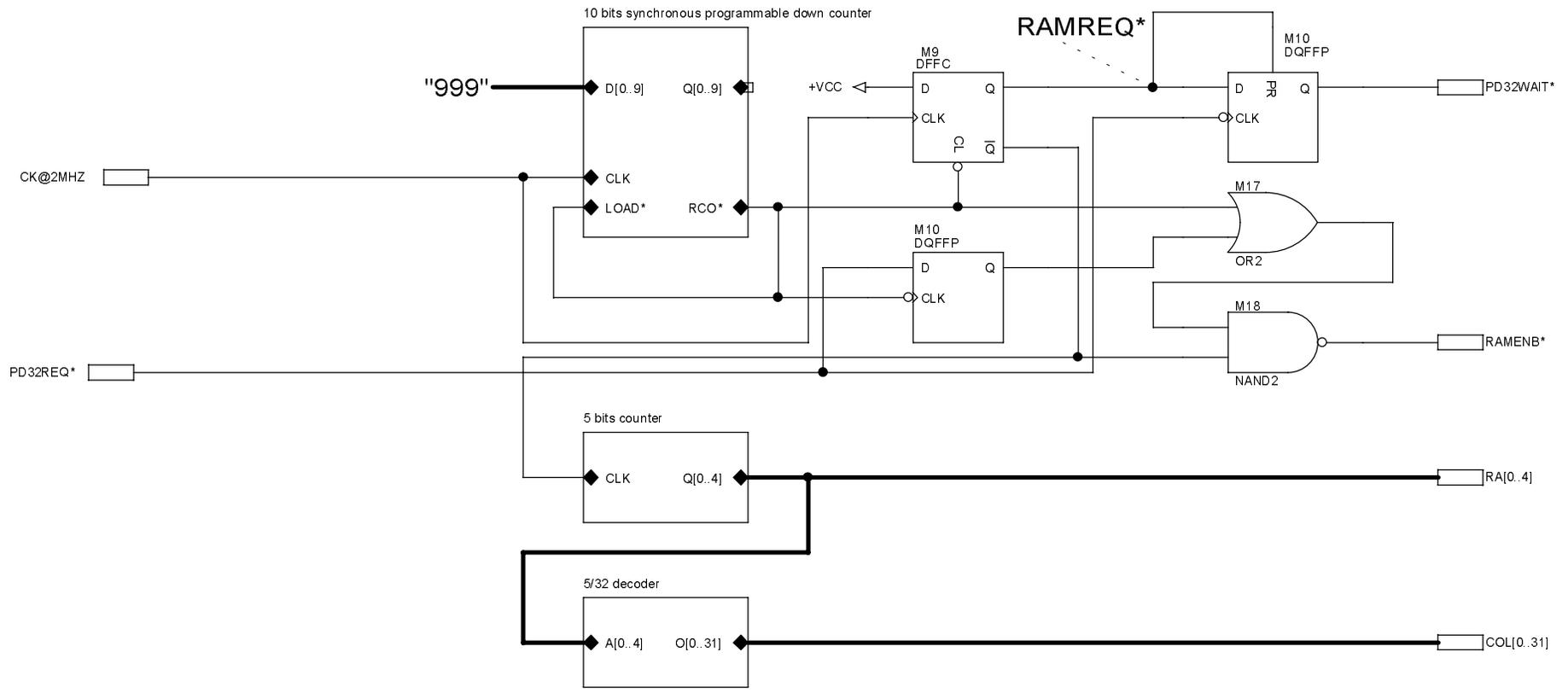
Il controllore della piastrina è sincronizzato da un clock di riferimento a 2MHz ($T=500\text{ns}$) il cui periodo è fissato in base al $\max(\text{ciclo di accesso del PD32}, \text{ciclo di accesso controllore locale})$; tenendo conto dei tempi specificati per i dispositivi critici ($T_a(\text{RAM})$, $T_{pd}(\text{dispositivi logici})$), tale valore può essere certamente assunto pari al tempo di ciclo del PD32.

2 cicli di clock vengono riservati per l'accesso del controllore della piastrina alla RAM locale e per introdurre un tempo di guardia pari alla massima durata del ciclo di accesso del PD32; tali cicli (RAMREQ*) vengono generati con la periodicità di 500µs richiesta dalle specifiche;

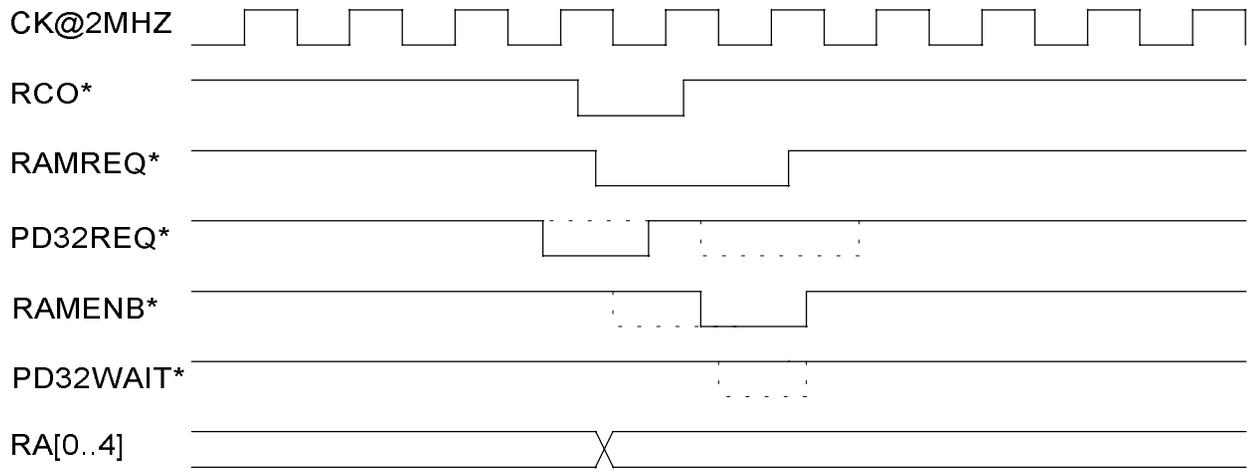
su 500µs di ciclo 499µs vengono riservati all'accesso esclusivo del PD32, 1µs al controllore locale (accesso+tempo di guardia); il controllore locale "consuma" lo 0.2% (1/500) del tempo disponibile

la linea di WAIT* del PD32 viene attivata solo se il ciclo di accesso alla RAM locale (in RD o WR) marcato dal segnale PD32REQ* inizia durante il periodo attivo del segnale RAMREQ*

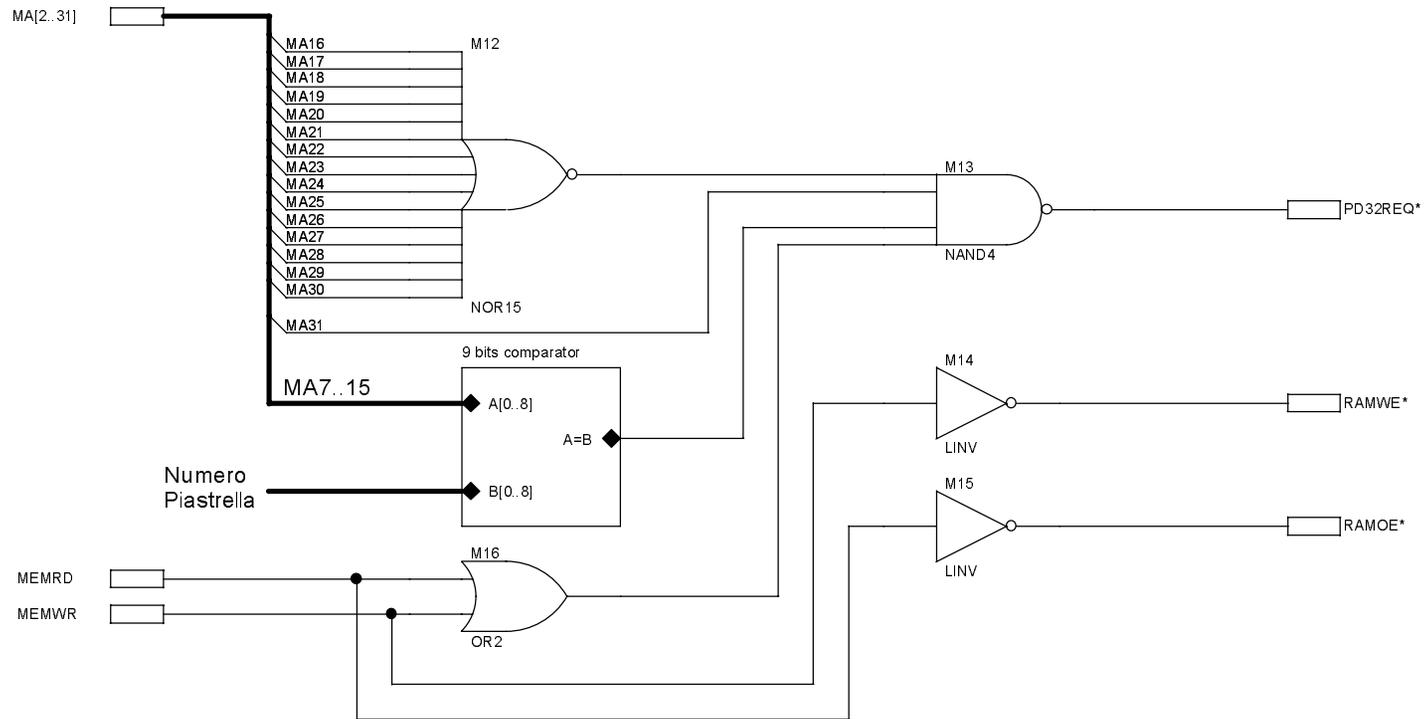
il segnale RAMENB* commuta i segnali di controllo della RAM verso il controllore locale; RAMENB* replica RAMREQ* se all'attivazione di RAMREQ* non vi è un ciclo di accesso PD32 attivo, altrimenti il segnale viene generato solo nella seconda metà del periodo di RAMREQ* quando il ciclo PD32 è certamente terminato



Timing di dettaglio:

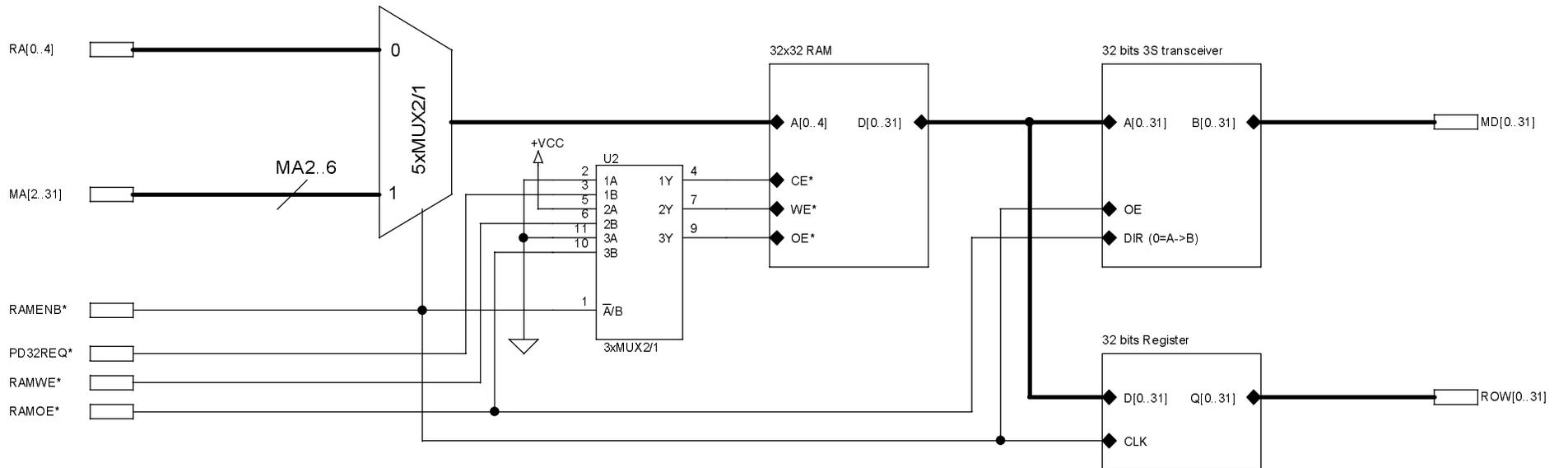


Title		
LEDPAN: Controller		
Size	Document Number	Rev
A4	1s970916/A.PUGNALONI	1
Date:	Wednesday, July 23, 1997	Sheet 3 of 5



Numero Piastrina	Indirizzo Base
0	8000000H
1	8000080H
2	8000100H
3	8000180H
511	800FF80H

Title		
LEDPAN: PD32 Interface		
Size	Document Number	Rev
A4	1s970916/A.PUGNALONI	1.1
Date:	Wednesday, July 23, 1997	Sheet 4 of 5



Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 14-01-1998

STUDENTE: _____

DOCENTE: _____

Progetto:

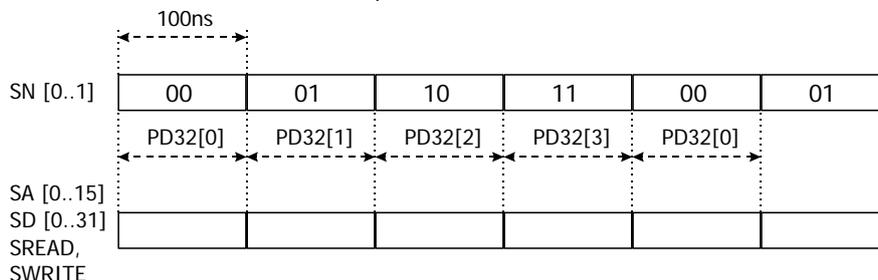
Memoria condivisa per sistema multiprocessore TDMMEM

Specifiche funzionali:

Un sistema multiprocessore costituito da 4 PD32 è dotato di un sottosistema di memoria condivisa TDMMEM da 64Kx32 bit per lo scambio di messaggi tramite mailbox (aree di memoria).

La memoria interna di TDMMEM è gestita da un controllore hardware che disciplina a divisione di tempo (TDM) gli accessi concorrenti dei 4 PD32 secondo le seguenti specifiche:

- il trasferimento dati tra i 4 PD32 e TDMMEM è basato su di un unico bus condiviso TDMBUS costituito dalle linee indirizzi SA[0..15], dati SD[0..31], SN[0..1], SREAD, SWRITE al quale sono attestati tutti i μ processori mediante opportune interfacce TDMIFC;
- l'accesso del generico PD32[0..3] alla memoria condivisa tramite il TDMBUS è disciplinato da TDMMEM mediante l'invio ciclico sulle linee SN[0..1] del numero del processore abilitato nella slot temporale corrente; la durata della singola slot temporale è di 100ns in modo che ogni PD32 abbia un'opportunità di accesso in ogni periodo di scansione di 400ns; la memoria condivisa di TDMMEM è mappata nell'area FFF00000-FFF3FFFF dello spazio di indirizzamento dei PD32.



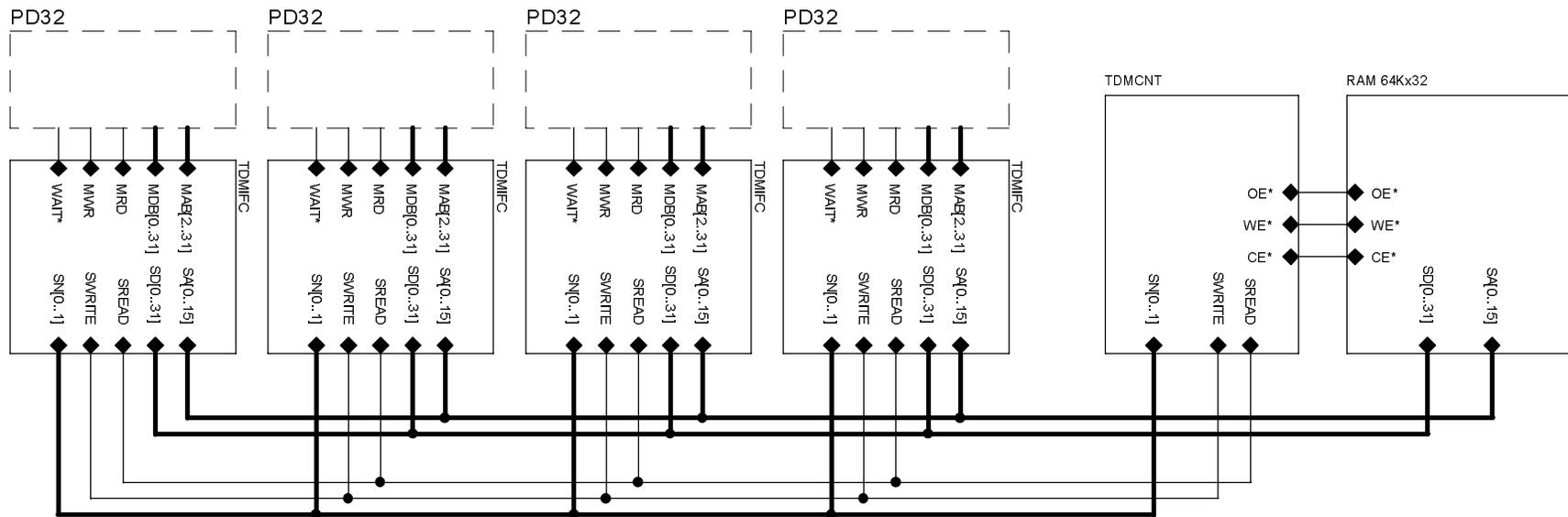
I PD32 hanno clock di macchina fra loro scorrelati ed a diverse frequenze; i cicli macchina di memory Read/Write hanno una durata minima di 300ns. Si presuppone che in una richiesta di scrittura sulla memoria condivisa il PD32 renda disponibile il dato contemporaneamente all'indirizzo.

Note/suggerimenti:

Per la sincronizzazione fra PD32 e TDMMEM si considerino attentamente i vincoli temporali specificati: occorrerà un semaforo HW che controlli la linea di wait del processore ed un registro tampone per la lettura da TDMMEM. Nel progetto di TDMMEM si tenga in debito conto il caso di cicli di scrittura adiacenti da parte di 2 o più PD32 consecutivi (segnale SWRITE).

Si richiedono:

1. lo schema a blocchi funzionali del sistema multiprocessore, di TDMMEM e dell'interfaccia TDMIFC;
2. il diagramma di timing che illustri la tecnica di sincronizzazione adottata nel trasferimento dei dati da/verso TDMMEM da parte del generico processore con i segnali TDMBUS e PD32 utilizzati;
3. lo schema elettrico di TDMMEM comprendente la RAM condivisa ed il controllore;
4. lo schema elettrico di TDMIFC.



Timing sincronizzazione PD32-TDMMEM:

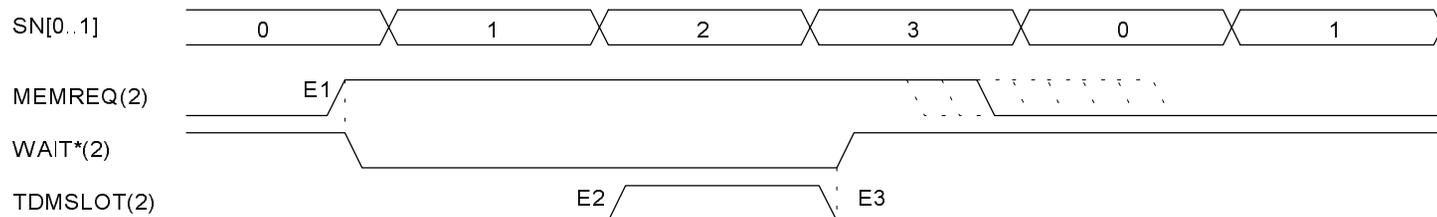
Esempio: accesso a TDMMEM da PD32(2): quando la TDMIFC rivela la richiesta di accesso alla RAM condivisa (MEMREQ=1)

il processore viene bloccato (WAIT*=0) in modo da estendere il ciclo macchina di memory read/write fino a quando

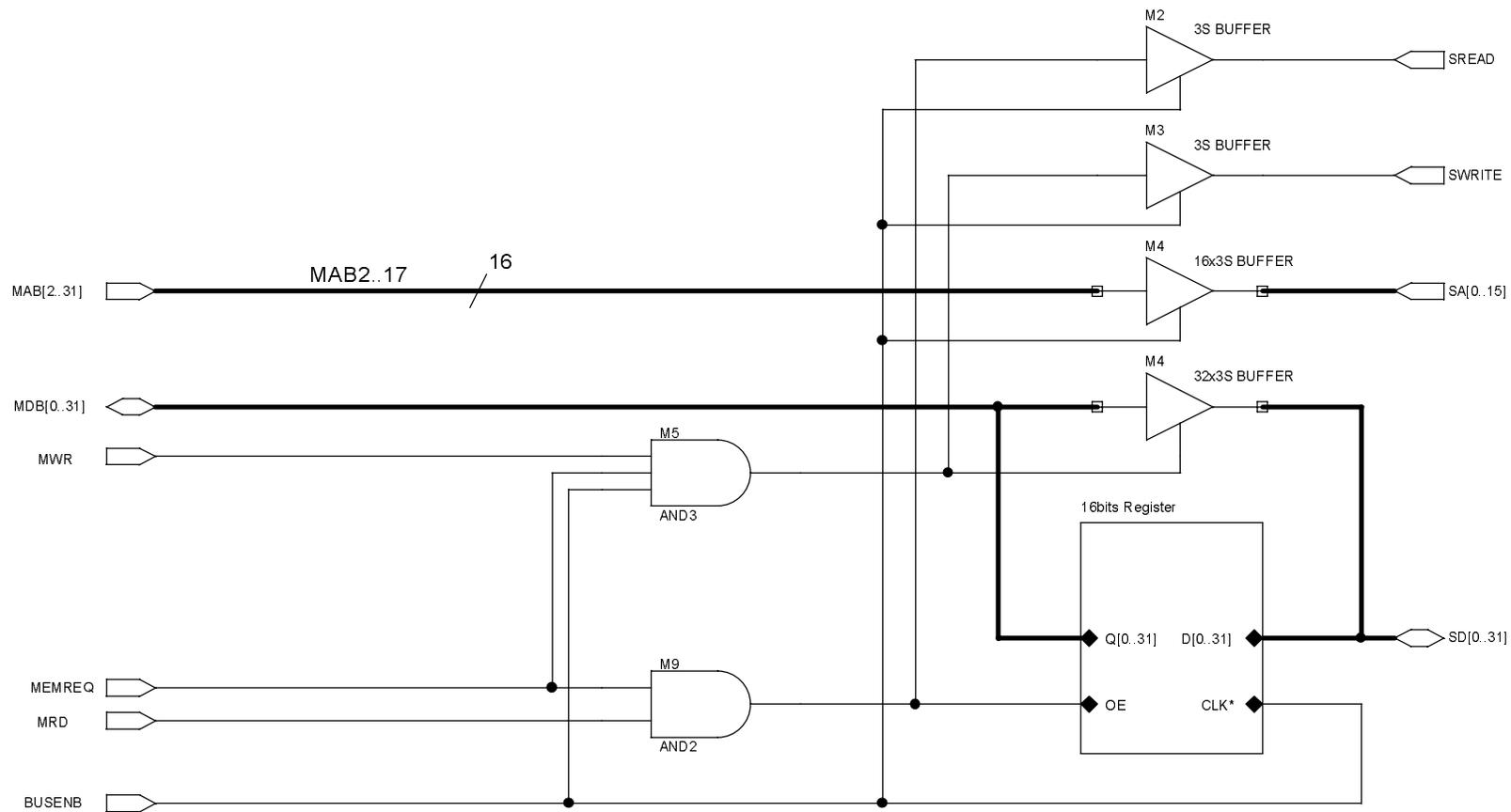
LA PROSSIMA SLOT assegnata al generico processore (ad.es. la slot 2) non è stata completamente utilizzata per eseguire l'accesso alla RAM (solo INSCRIVENDO un'intera slot di accesso nel ciclo macchina del PD32 è possibile garantire il corretto funzionamento del sistema indipendentemente dalle fasi relative e dai diversi periodi dei segnali SN[0..1] e MEMREQ

E' quindi necessario imporre il SEQUENZIAMENTO dei segnali MEMREQ, WAIT*, TDMSLOT per risolvere il problema della sincronizzazione.

Gli eventi da considerare sono quindi E1=attivazione di MEMREQ, E2=attivazione di TDMSLOT, E3=disattivazione di TDMSLOT nell'ordine indicato.



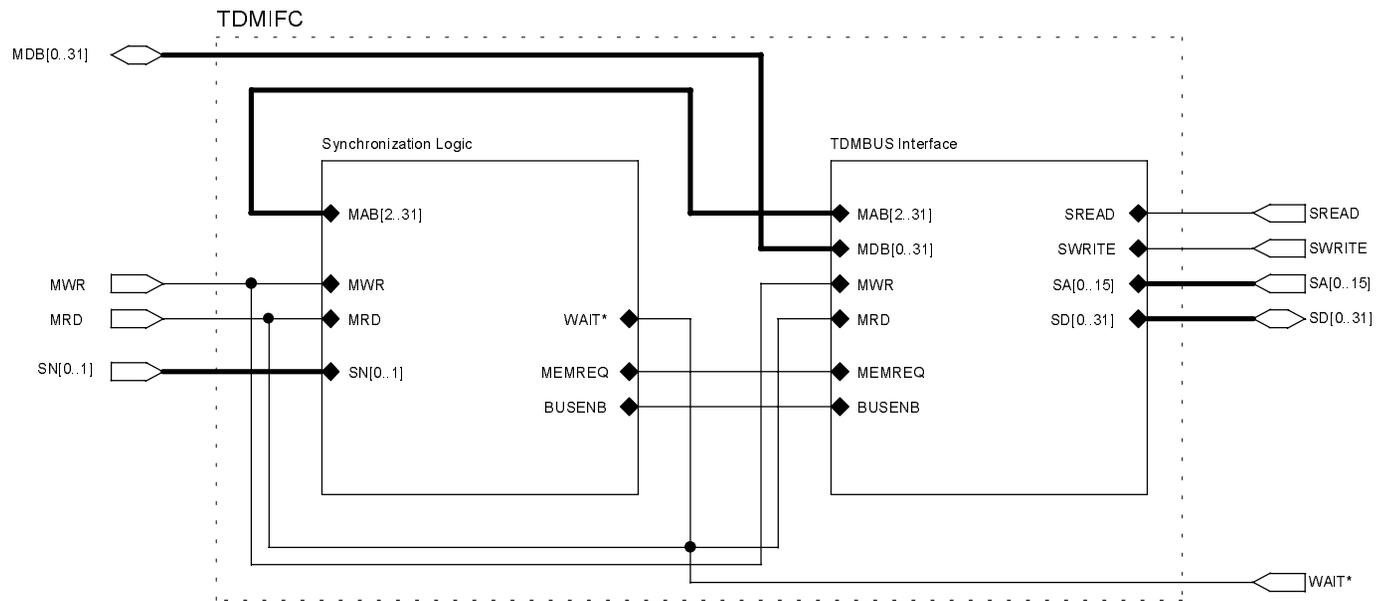
Title			TDMMEM: Block Diagram & Timing		
Size	Document Number	Date			Rev
A4	1s980114/A.PUGNALONI	Thursday, January 22, 1998			1
Date:	Wednesday, July 23, 1997	Sheet	1	of	5



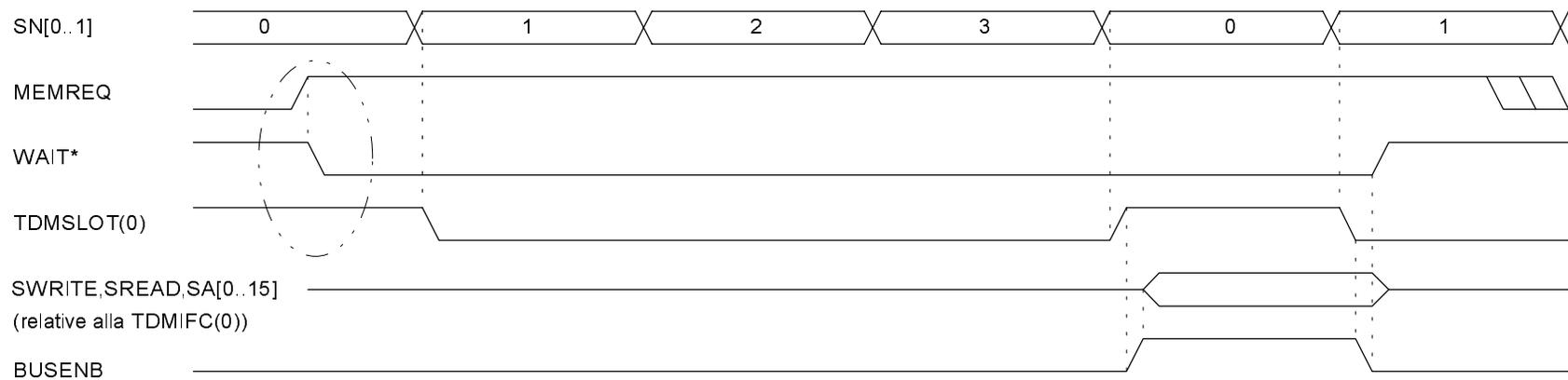
Il registro di appoggio è indispensabile per campionare il dato letto dalla RAM e mantenerlo stabile finchè il PD32 non termina il ciclo di memory READ (cfr. timing TDMIFC).

In scrittura è invece sufficiente un buffer 3S dato che il bus dati del processore viene "congelato" per tutta la durata del ciclo esteso tramite gli stati di WAIT.

Title			
TDMMEM: TDMIFC Bus Interface			
Size	Document Number	Thursday, January 22, 1998	Rev
A4	1s980114/A.PUGNALONI		1
Date:	Wednesday, July 23, 1997	Sheet	2 of 5



TIMING di riferimento TDMIFC:



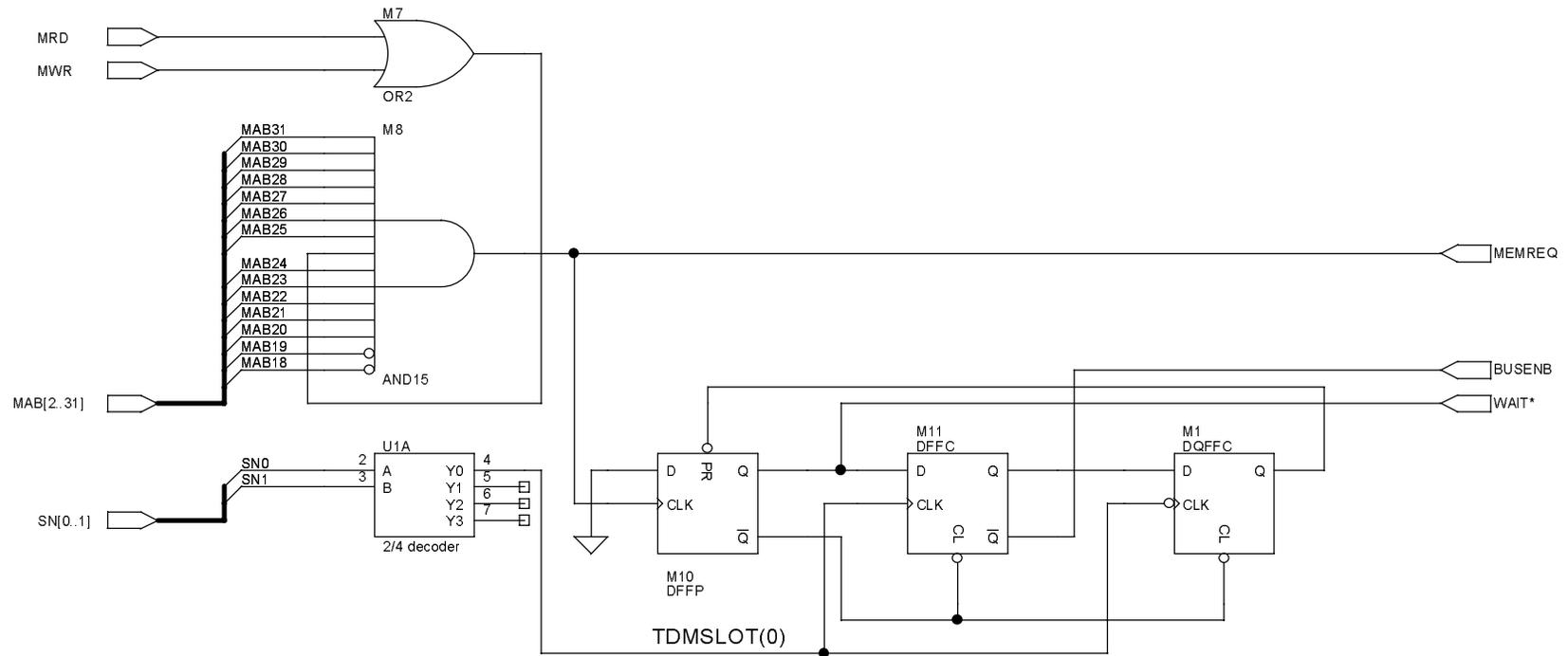
Il timing è riferito al funzionamento della TDMIFC(0) (PD32 #0).

Si noti che, nonostante il ciclo di MEMREQ inizi DURANTE una slot dedicata al PD32, il ciclo di accesso viene eseguito soltanto alla prossima TDMSLOT(0) quando il segnale BUSENB viene generato dalla logica di sincronizzazione.

Dalla temporizzazione risulta evidente la necessità di un registro di appoggio per i cicli di lettura dato che il segnale MEMREQ termina dopo un tempo ignoto (PD32 asincroni e con clock non specificato) dalla fine della TDMSLOT(0).

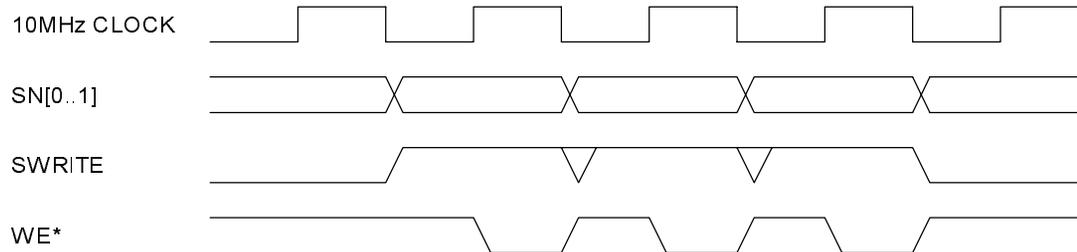
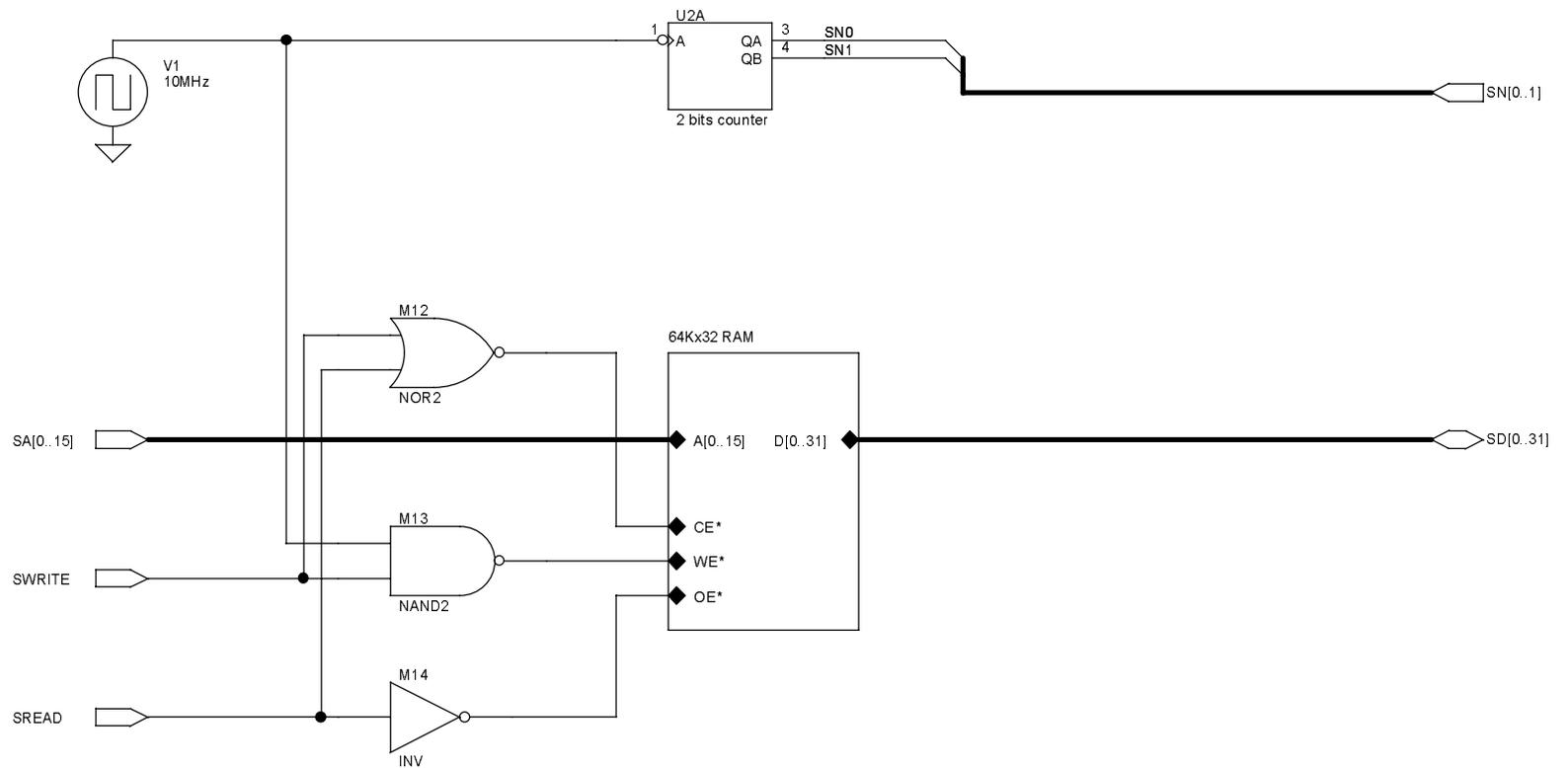
Il segnale WAIT* può essere disattivato solo dopo la fine di TDMSLOT(0).

Title			TDMMEM: TDMIFC Block Diagram & Timing		
Size	Document Number	Thursday, January 22, 1998			Rev
A4	1s980114/A.PUGNALONI				1
Date:	Wednesday, July 23, 1997	Sheet	3	of	5



(*): il decoder è cablato per generare la TDM SLOT(0); le altre interfacce (1..3) utilizzeranno le rispettive uscite Y1..Y3 dei propri decoder.

Title		
TDMMEM: TDMIFC Synchronization Logic		
Size	Document Number	Rev
A4	1s980114/A.PUGNALONI	1
Date:	Thursday, January 22, 1998	
Wednesday, July 23, 1997	Sheet	4 of 5



Per garantire la generazione di impulsi di WE* anche nel caso di cicli di scrittura consecutivi il segnale SWRITE deve essere impulsato, ad.es. utilizzando come abilitazione la fase alta del clock di scansione.

Title		
TDMMEM: TDMCNT & SHARED RAM		
Size	Document Number	Rev
A4	1s980114/A.PUGNALONI	1
Date:	Thursday, January 22, 1998	
Wednesday, July 23, 1997	Sheet	5 of 5

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 28-01-1998

STUDENTE: _____

DOCENTE: _____

Progetto:

Estrattore/formattatore di messaggi MSGFMT

Specifiche funzionali:

Una linea seriale sincrona basata sui segnali DATA, CLOCK trasporta trame di messaggi indirizzati [MSG] costituiti da nibble (parole da 4 bit) secondo il seguente formato:

.... [F][MSG][MSG] [MSG][F][MSG][MSG] [MSG];

il nibble [F]=1111 è un flag che marca l'inizio di ogni trama; per evitare falsi riconoscimenti dell'inizio trama i messaggi [MSG] sono codificati alla sorgente in modo da non presentare mai sequenze di 4 bit che imitino il flag 1111. I messaggi [MSG] sono costituiti da 8 nibble secondo il formato:

[A₁][A₂][A₃][M₁][M₂][M₃][M₄][M₅]

dove i 3 nibble A₁₋₃ rappresentano l'indirizzo del destinatario del messaggio.

Il formattatore di messaggi MSGFMT riceve dalla linea seriale le trame e deve effettuare le seguenti funzioni:

1. sincronizzarsi con le trame rivelando il passaggio del nibble di flag [F] sulla linea seriale;
2. riconoscere i messaggi con indirizzo [A₁][A₂][A₃] = [1010][0101][0110] (A56 Hex);
3. estrarre tali messaggi formattandoli in singole Longword da 32 bit ed inviarli su di una porta di uscita verso un microprocessore PD32.

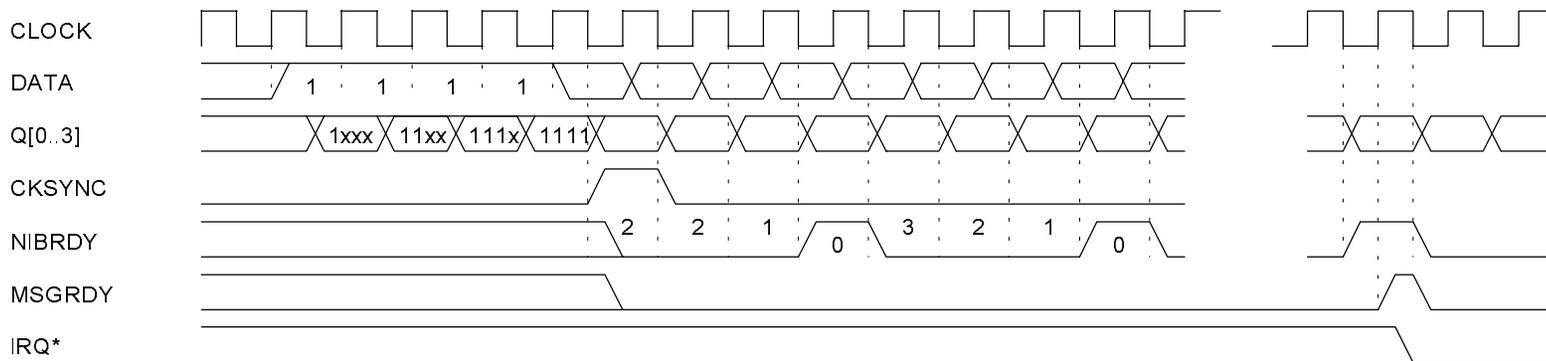
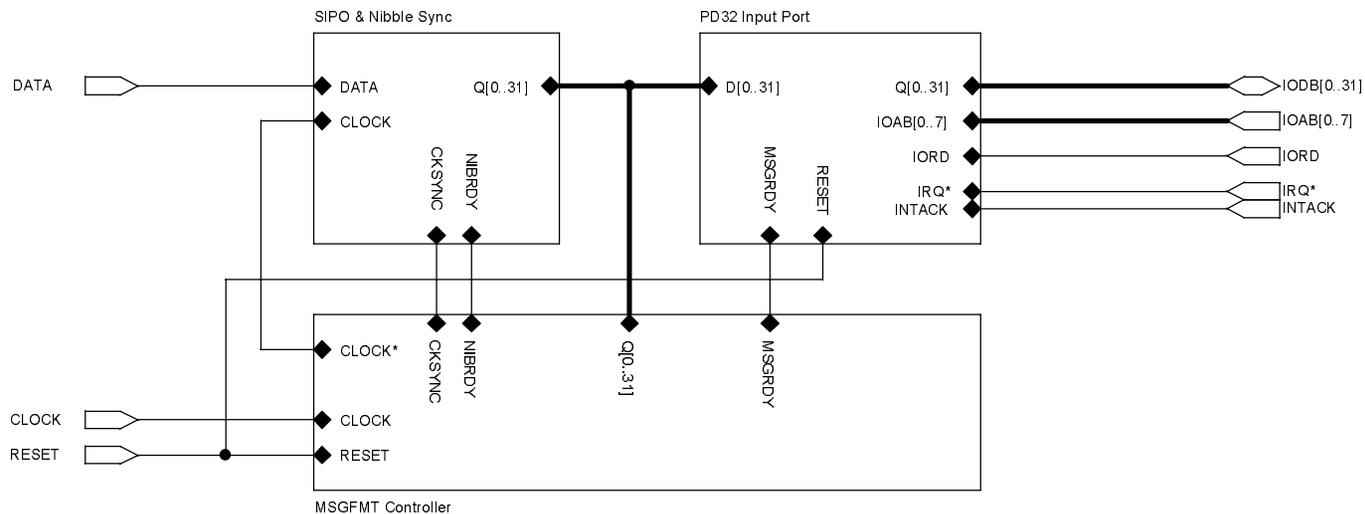
MSGFMT deve essere basato su di un controllore microprogrammato di tipo D-MEALY in grado di effettuare test su 4 (o più) variabili di ingresso contemporaneamente; in tal modo MSGFMT potrà direttamente riconoscere il valore di un nibble in un singolo stato macchina mediante un costruito ASM comprendente un test a 16 (o più) diramazioni.

Note/suggerimenti:

Si faccia attenzione ad effettuare la PRIMA sincronizzazione di trama analizzando il nibble in transito ad ogni ciclo di clock della linea seriale (la linea non è dotata di un segnale di sincronismo di nibble). Si supponga inoltre che il PD32 prelevi il messaggio dalla porta di uscita di MSGFMT prima che un successivo messaggio vi venga riscritto e quindi non sia necessario gestire un protocollo di handshaking bidirezionale tra MSGFMT e PD32.

Si richiedono:

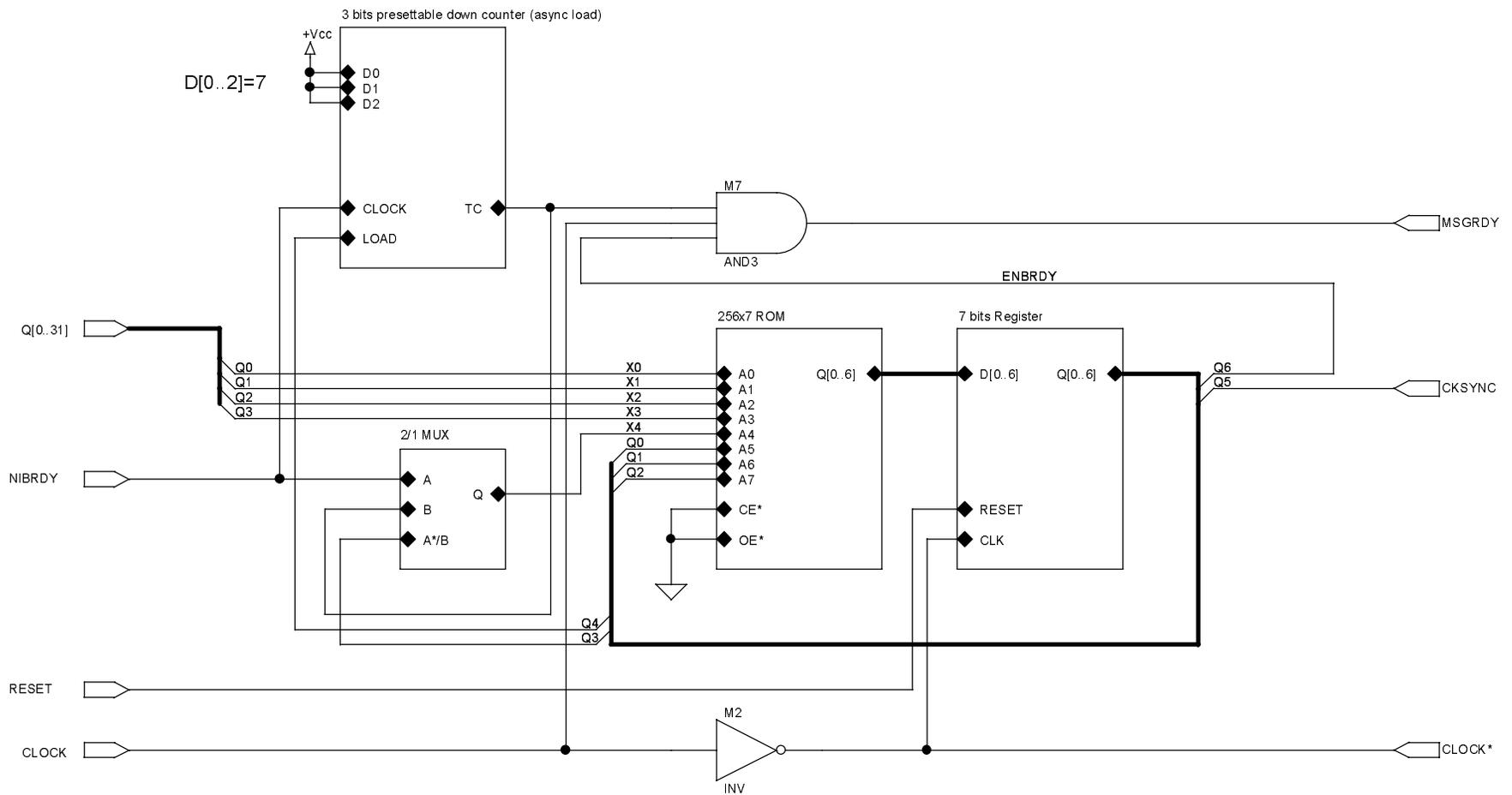
1. lo schema a blocchi di MSGFMT ed il diagramma di temporizzazione dell'interfaccia con la linea seriale di ingresso;
2. la struttura del controllore D-MEALY e l'organizzazione logica della ROM di microprogramma;
3. lo schema elettrico completo di MSGFMT;
4. il microprogramma di controllo di MSGFMT.



Timing sincronizzazione di messaggio:

Il controllore, una volta rivelata la configurazione 1111 sulle linee Q[0..3], genera il segnale CKSYNC per riallineare un contatore mod.4 che fornisce il sincronismo di nibble NIBRDY allineato sui fronti di campionamento di DATA; il campionamento di DATA avviene sui fronti di discesa del CLOCK; il controllore opera con lo stesso sincronismo (CLOCK*). Se l'indirizzo del messaggio è quello atteso, il controllore provvede alla generazione del segnale MSGRDY per caricare i 32 bit ricevuti in una porta d'ingresso del PD32. La disponibilità di un messaggio è segnalata al PD32 mediante l'attivazione della richiesta di interrupt IRQ*, automaticamente disattivata in seguito ad una lettura della porta di input.

Title			
MSGFMT: Block Diagram & Timing			
Size	Document Number	Date	
A4	1s980128/A.PUGNALONI	Tuesday, February 10, 1998	
Date:		Sheet	Rev
Wednesday, July 23, 1997		1 of 4	2



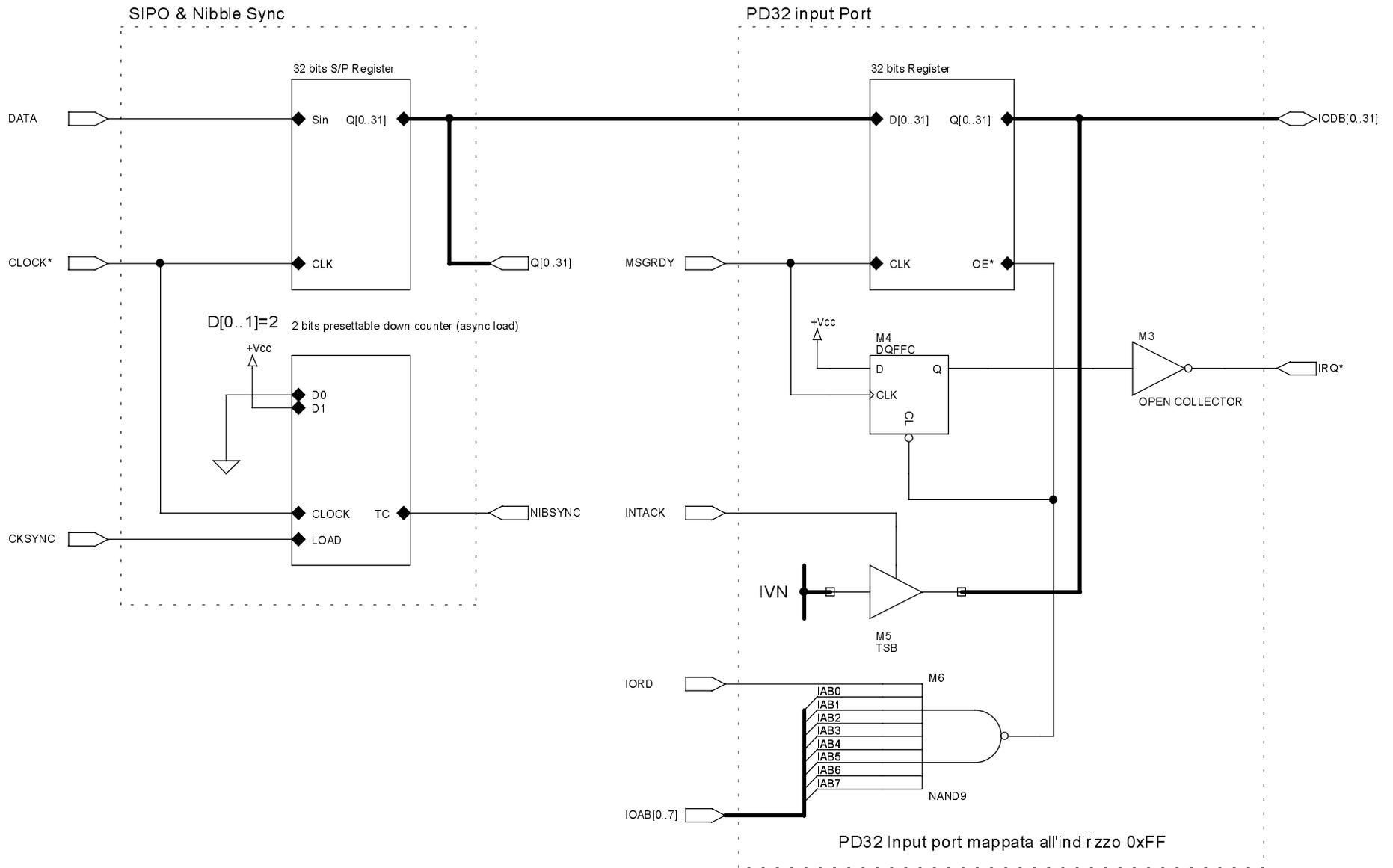
Formato ulstruzione: [SS=Q0Q1Q2][SEL=Q3][TASK=Q4Q5Q6]
 TASK Q4 = LOAD, TASK Q5 = CKSYNC, TASK Q6 = ENBRDY

Il controllore, tramite il campo SEL, può selezionare la 5a variabile di ingresso in modo da coprire le 6 variabili di condizione previste.

Ogni stato indirizza una pagina di 32 possibili ulstruzioni successive, una delle quali è selezionata dalle 5 variabili di condizione formate da Q0..3 e da X4 estratta in base al valore del bit SEL.

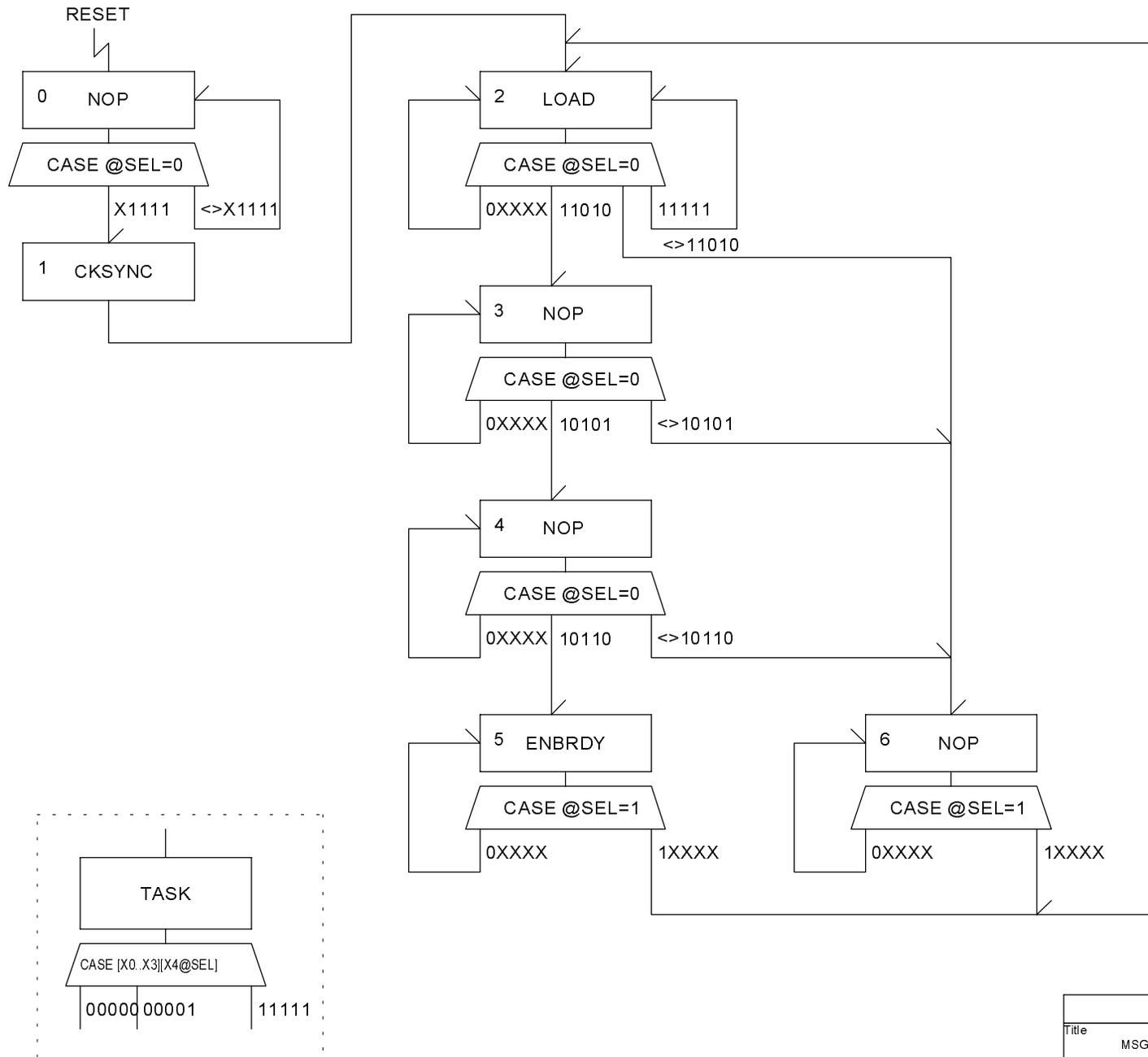
Il contatore presettabile a 3 bit genera il criterio di fine messaggio 8 nibble dopo l'inizio dell'indirizzo di un messaggio (cfr. microprogramma e timing di riferimento)

Title			
MSGFMT: Controller			
Size	Document Number	Date	
A4	1s980128/A.PUGNALONI	Thursday, February 05, 1998	
Date:	Wednesday, July 23, 1997	Sheet	2 of 4
			Rev 1



Lo schema non comprende l'eventuale logica di condizionamento del segnale INTACK (daisy-chain per la risoluzione delle priorità)

Title			
MSGFMT: SIPO&Nibble Sync, PD32 input port			
Size	Document Number	Thursday, February 05, 1998	
A4	1s980128/A.PUGNALONI	Rev 1	
Date:	Wednesday, July 23, 1997	Sheet	3 of 4



Commenti stati:

- 0: attesa Flag di sincronismo
- 1: Flag riconosciuto, reset generatore NIBSYNC
- 2: riconoscimento Flag o nibble "A"
- 3: riconoscimento nibble "5"
- 4: riconoscimento nibble "6"
- 5: attesa fine messaggio, abilitazione MSGRDY
- 6: attesa fine messaggio (indirizzo non riconosciuto)

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 11-02-1998

STUDENTE: _____

DOCENTE: _____

Progetto:

Interfaccia per la misura di periodo di rotazione TACHOM

Specifiche funzionali:

Un sistema a microprocessore PD32 è adibito al controllo di velocità di una turbina che può operare nel range di velocità da 1000 a 20000 giri/minuto; la turbina è dotata di un sensore magnetico che genera un impulso per giro sulla linea digitale PULSE. L'interfaccia di misura TACHOM del PD32 riceve tali impulsi ed effettua le seguenti pre-elaborazioni per ottenere il periodo di rotazione della turbina che il PD32 provvede a trasformare in velocità istantanea:

1. ricava il periodo di rotazione istantaneo della turbina T_{rot} misurando la distanza temporale fra i fronti successivi degli impulsi ricevuti con la precisione di 1 microsecondo; se la misura supera il valore massimo T_{rot} deve essere saturato a FF..F;
2. calcola la media mobile MA (Moving_Average) $MA[kT] = \frac{1}{4} \sum_{i=0,3} T_{rot}[(k-i)T]$ sugli ultimi 4 campioni della sequenza di misure T_{rot} ;
3. scrive ogni valore $MA[kT]$ così ottenuto in una variabile posta all'indirizzo 80000000H accedendo in DMA alla memoria del PD32;

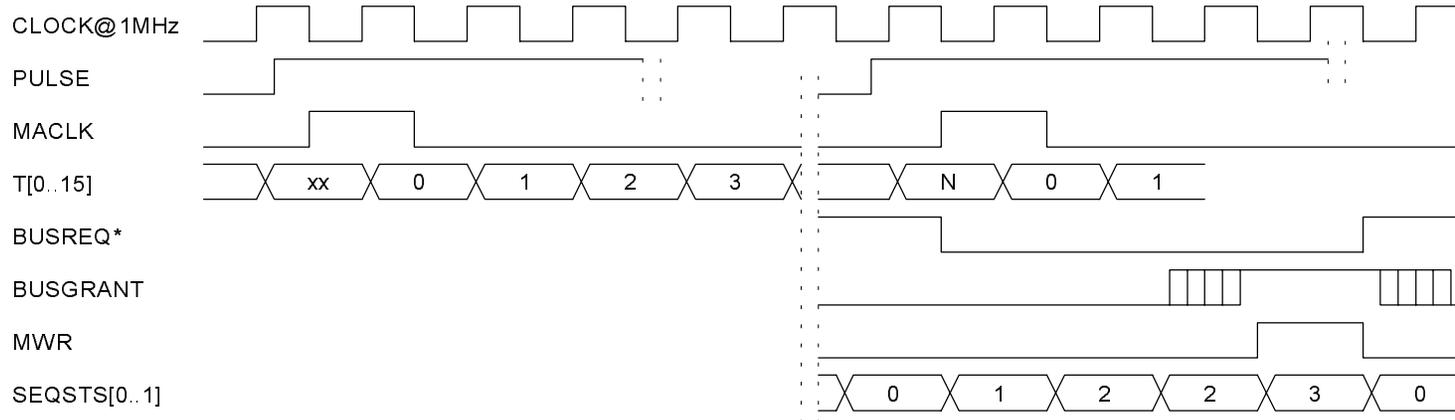
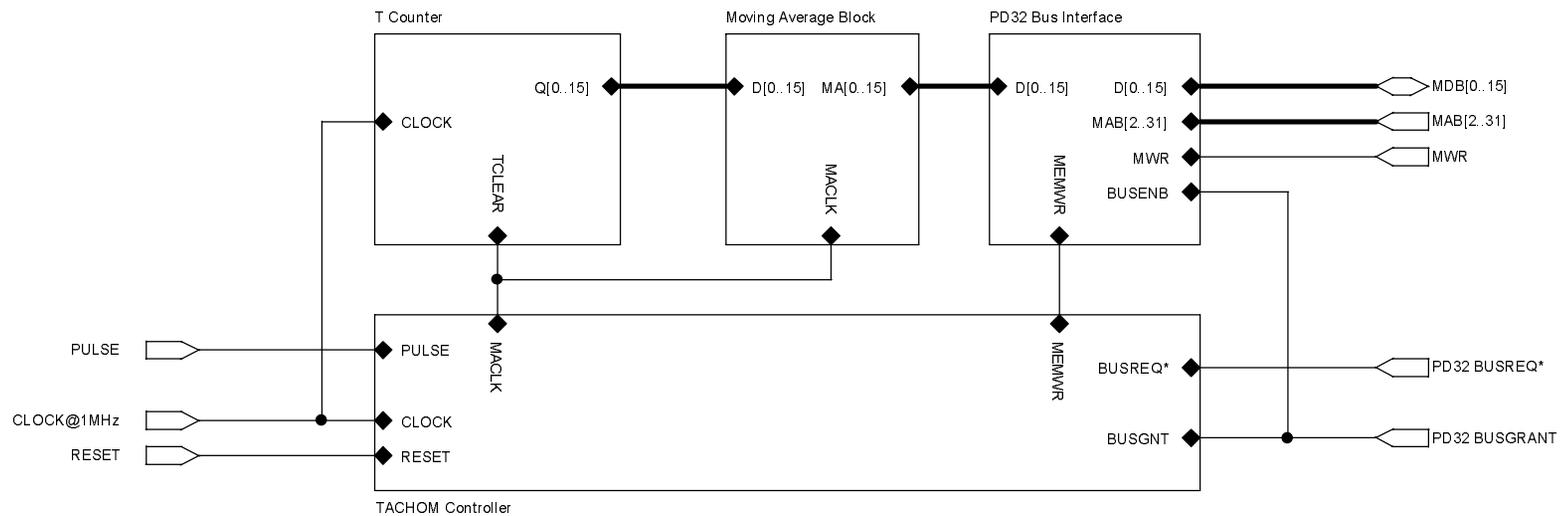
La memoria del PD32 può operare con cicli di accesso di durata minima di 500ns.

Note/suggerimenti:

Si ricorda che la funzione di media mobile su N campioni genera un risultato $MA[kT]$ ad ogni nuovo campione d'ingresso, a differenza della media semplice che fornisce un valore ogni N campioni d'ingresso. Si noti inoltre che per ottenere misure esatte è indispensabile misurare il periodo intercorrente fra i fronti (omologhi) degli impulsi in quanto la durata degli stessi è variabile in funzione della velocità di rotazione della turbina. Si consideri il funzionamento a regime di TACHOM, trascurando l'errore iniziale sulla media mobile. Si consiglia di implementare il DMA con un semplice sequenziatore.

Si richiedono:

1. il numero di medie $MA[kT]$ generate al secondo nel range operativo della turbina ed il numero di bit necessari per rappresentare il periodo T_{rot} e la media MA;
 2. lo schema a blocchi di TACHOM con l'indicazione dei necessari segnali di interfaccia verso il PD32;
 3. il diagramma di timing che specifichi il funzionamento di TACHOM;
 4. lo schema elettrico completo di TACHOM.
-



Calcolo del numero di bit necessari per esprimere il periodo Trot:

Range giri/sec: 16.6 (1000 giri/min) - 333.3 (20000 giri/min); Range Trot: 3 - 60 ms

Risoluzione in bit: $\text{INT_SUP}(\log_2(\text{Tmax} / 1\mu\text{s}) = 60000) = 16$

Timing sincronizzazione di TACHOM:

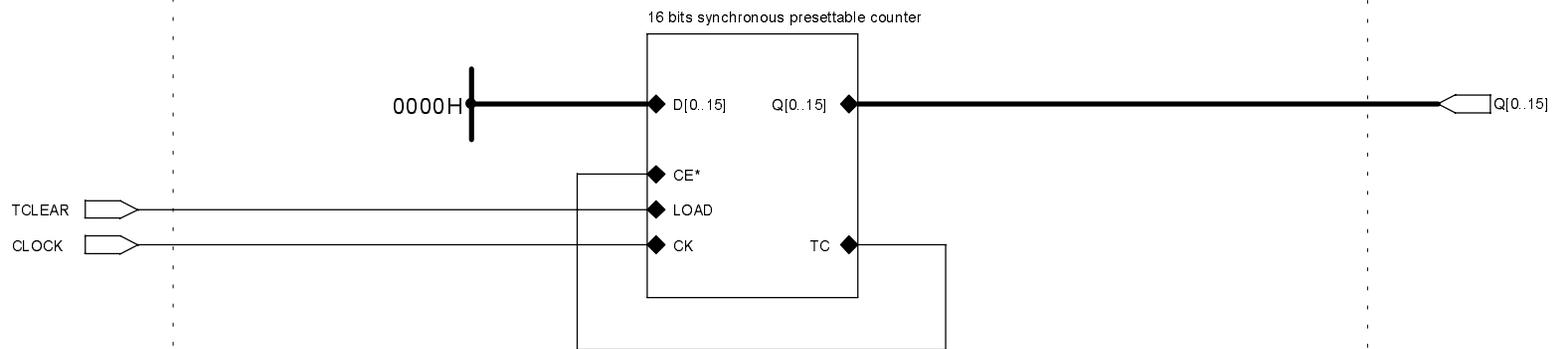
TACHOM è gestito da un sequenziatore a 4 stati codificati dai segnali SEQSTS[0..1];

il segnale PULSE è utilizzato per far partire il sequenziatore il quale genera il segnale LD/CE* che reinizializza il contatore a 16 bit utilizzato per misurare il periodo e trasferisce il periodo T[0..15] verso il blocco di calcolo della MA.

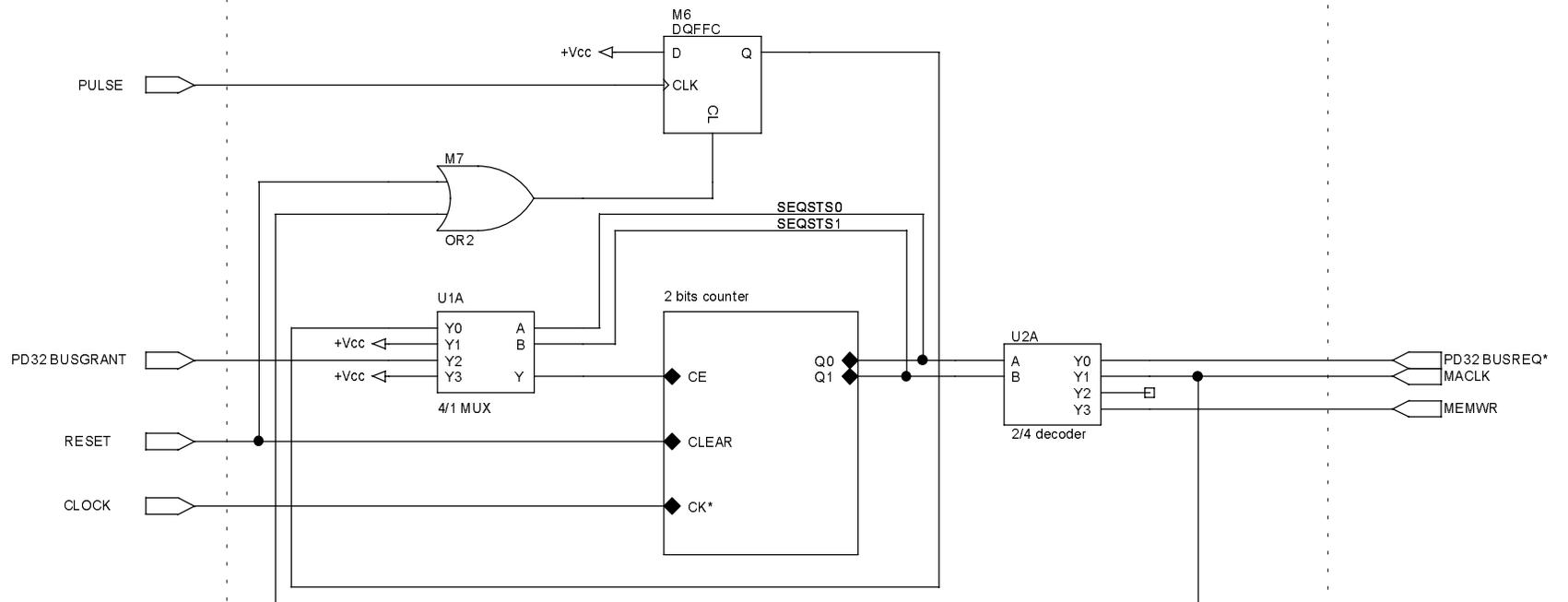
A partire dallo stato 1 del sequenziatore viene generata la richiesta di bus al PD32 e, una volta ricevuto il segnale di GRANT, lo stato 3 genera l'impulso di MWR verso la memoria.

Title			TACHOM: Block Diagram & Timing		
Size	Document Number	Date:			Rev
A4	1s980211/A.PUGNALONI	Wednesday, February 18, 1998			2
Date:		Wednesday, July 23, 1997	Sheet	1	of 3

T Counter



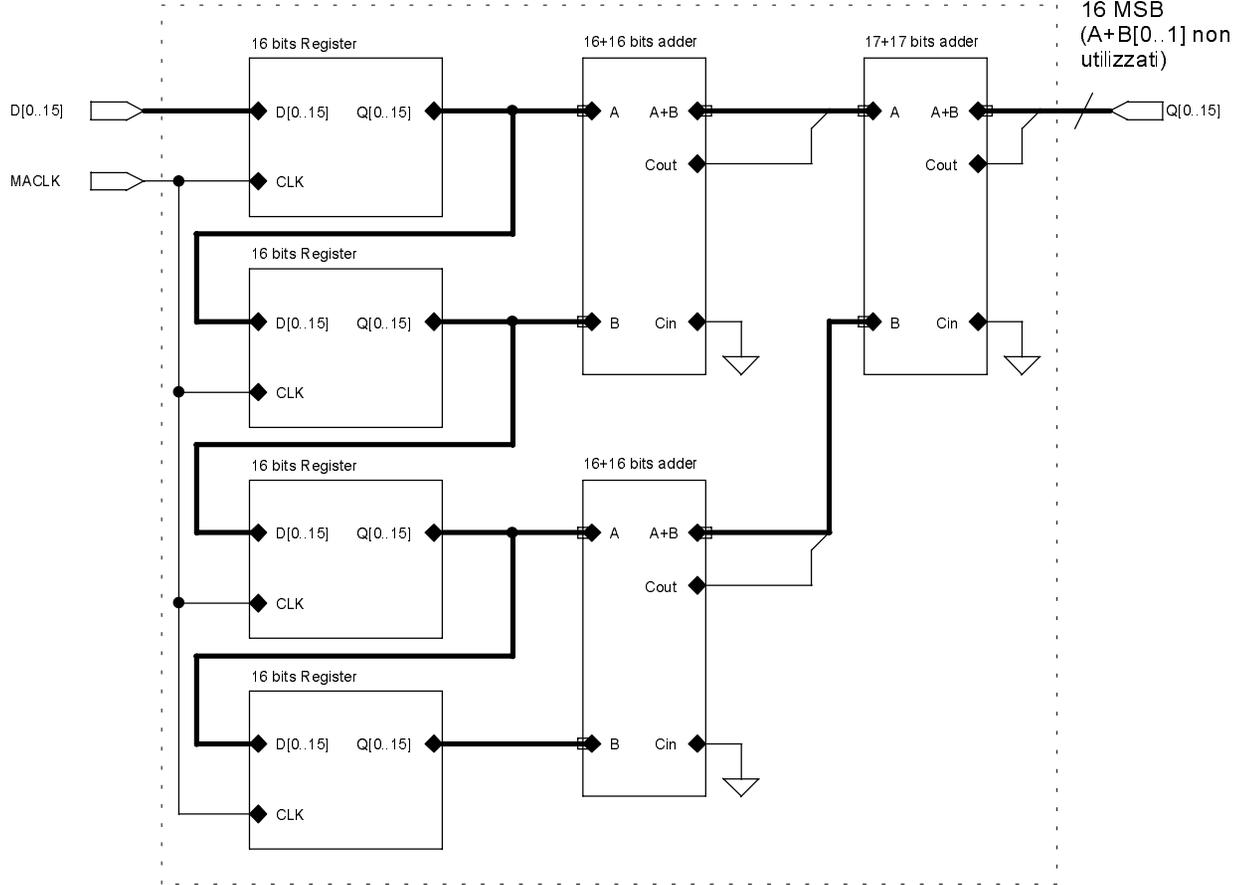
TACHOM Controller



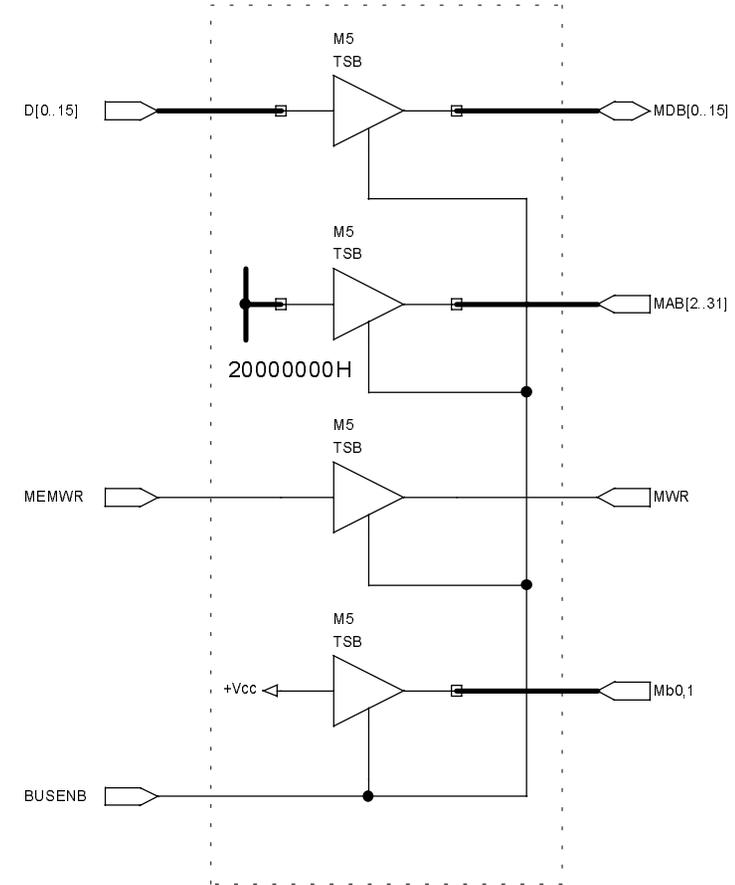
L'avanzamento di stato del sequenziatore è condizionato, stato a stato, alla variabile di ingresso estratta dal MUX 4/1 connesso al segnale Count Enable (CE) del contatore negative edge-triggered.

Title		
TACHOM: T Counter & Controller		
Size	Document Number	Rev
A4	1s980211/A.PUGNALONI	1
Date:	Wednesday, July 23, 1997	Sheet 2 of 3

Moving Average Block



PD32 Bus Interface



La MA è calcolata in parallelo mediante una linea di ritardo a 4 stadi (shift register a 16 bit).

Title			
TACHOM: Moving Average & PD32 Bus Interface			
Size	Document Number	Date	Rev
A4	1s980211/A.PUGNALONI	Wednesday, February 18, 1998	1
Date:	Wednesday, July 23, 1997	Sheet	3 of 3

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 16-04-1998

STUDENTE: _____

DOCENTE: _____

Progetto:

Pre-elaboratore di impulsi ECOSCAN

Specifiche funzionali:

ECOSCAN è una periferica PD32 dedicata all'acquisizione veloce ed al filtraggio di un burst di campioni ottenuti dal segnale prodotto da un sensore ecografico. Il software su PD32 comanda l'avvio dell'acquisizione generando un segnale di controllo verso la periferica ECOSCAN la quale processa il segnale analogico d'ingresso come segue:

1. campiona il segnale analogico ad una frequenza di 20MHz con una precisione di 0.1%;
2. calcola la media mobile MA (Moving_Average) $MA[kT] = 0.25 \times \sum_{i=0,3} S[(k-i)T]$ su 4 campioni della sequenza di misure $S[kT]$;
3. calcola il valore $\text{Log}_{MA}[kT] = \log_{10}(MA[kT])$ con una risoluzione di 16 bit mediante una ROM;
4. scrive ogni valore $\text{Log}_{MA}[kT]$ così ottenuto in una RAM locale da 64Kx16 bit accessibile dal PD32 a partire dall'indirizzo 80000000H;
5. una volta riempita la RAM locale blocca l'acquisizione e segnala al PD32 la disponibilità delle misure mediante interruzione.

Specifiche dispositivi:

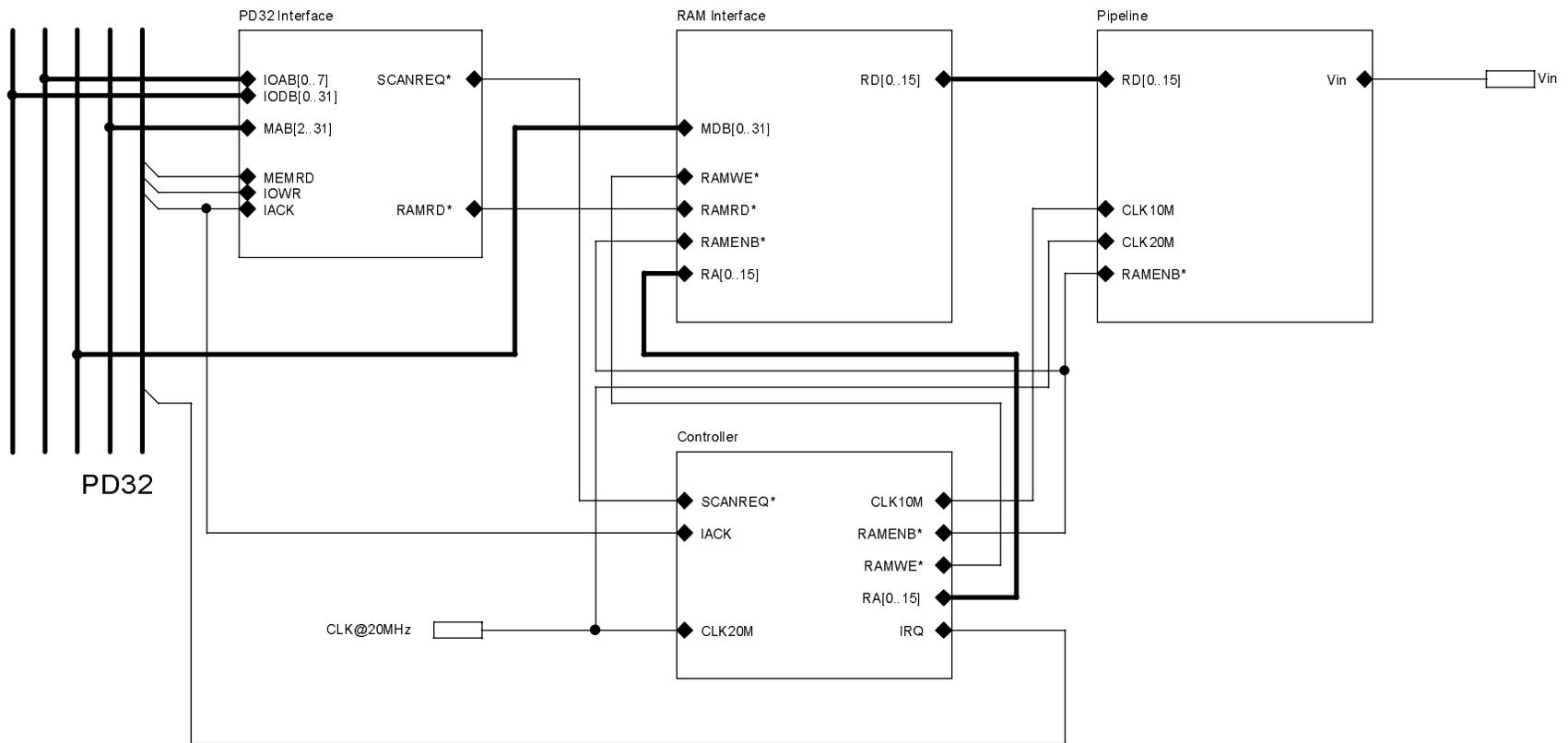
Il convertitore A/D Flash da utilizzare prevede l'unico segnale di controllo Encode e genera il campione digitale dopo un tempo massimo $T_{\text{ENCODE}}=40\text{ns}$. I sommatore da utilizzare per il calcolo delle MA sono moduli a 4 bit con $T_C=10\text{ns}$. La ROM contenente la tabella di conversione logaritmica ha un $T_A=70\text{ns}$.

Note/suggerimenti:

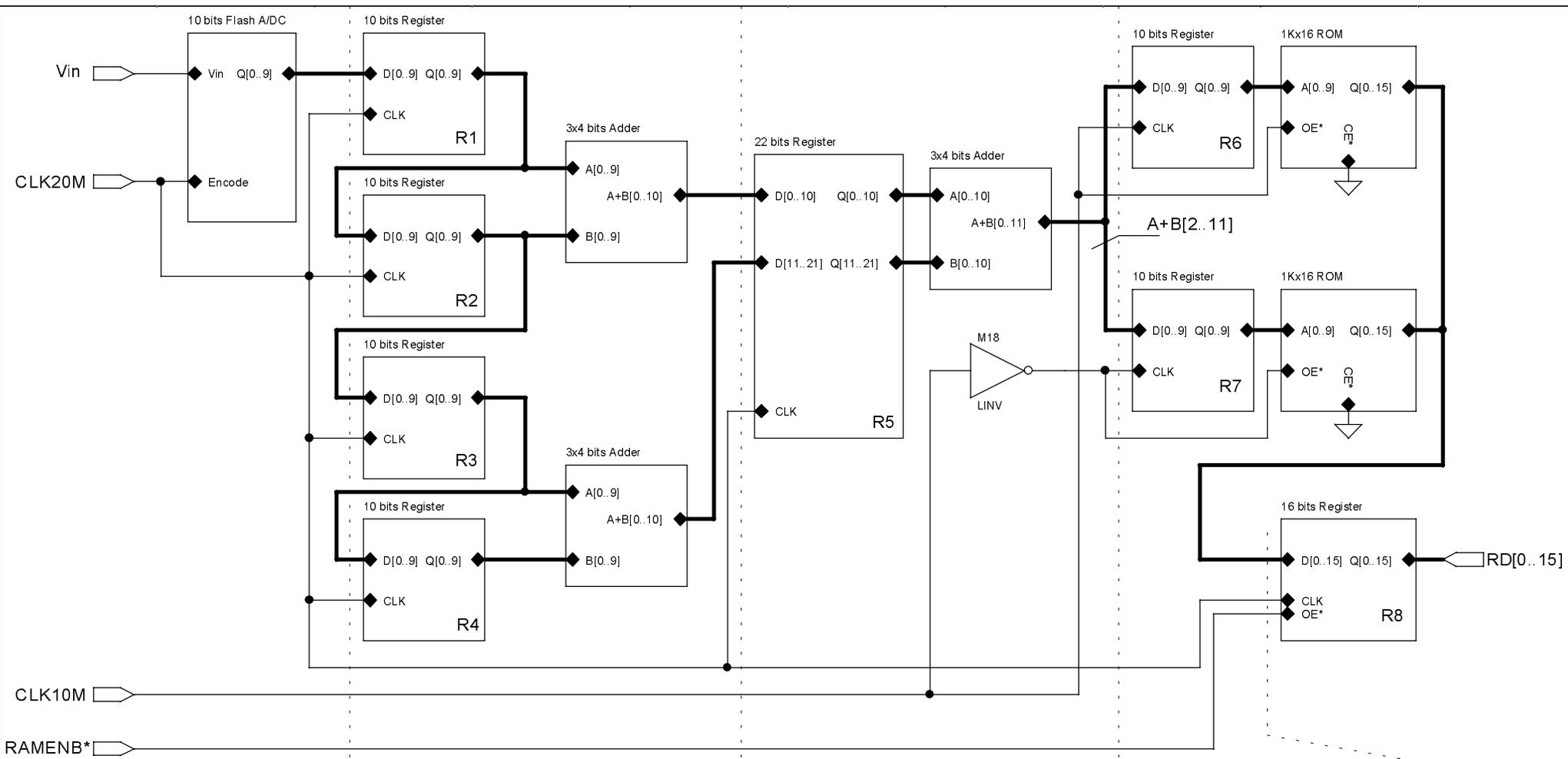
Si ricorda che la funzione di media mobile su N campioni genera un risultato $MA[kT]$ ad ogni nuovo campione d'ingresso, a differenza della media aritmetica semplice che fornisce un valore ogni N campioni d'ingresso. Si implementi la catena di elaborazione del segnale mediante una struttura pipelined tenendo in debito conto i tempi di calcolo specificati per i dispositivi fisici.

Si richiedono:

1. lo schema a blocchi di ECOSCAN con l'indicazione dei necessari segnali di interfaccia verso il PD32;
 2. l'architettura della pipeline, i tempi di calcolo massimi di ogni stadio ed il tempo di latenza;
 3. il diagramma dettagliato di timing che specifichi il funzionamento di ECOSCAN;
 4. lo schema elettrico completo di ECOSCAN.
-



Title		
ECOSCAN: Block Diagram		
Size	Document Number	Rev
A4	1s980416/A.PUGNALONI	1
Date:	Wednesday, July 23, 1997	Sheet 1 of 6



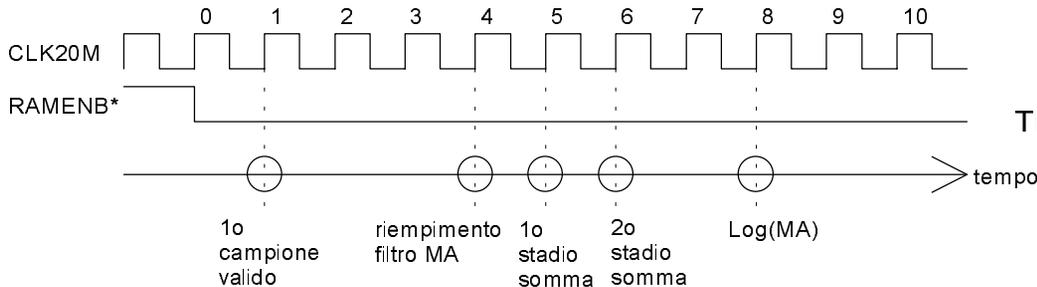
$T_{\text{encode}} + T_{\text{su,R}} = 45 \text{ ns}$

$T_{\text{pd,R}} + 3 \times T_{\text{pd,adder}} + T_{\text{su,R}} = 45 \text{ ns}$

$T_{\text{pd,R}} + 3 \times T_{\text{pd,adder}} + T_{\text{su,R}} = 45 \text{ ns}$

$(T_{\text{pd,inv}} + T_{\text{pd,R}} + T_{\text{a,rom}} + T_{\text{su,R}}) = 95 \text{ ns} (/2)$

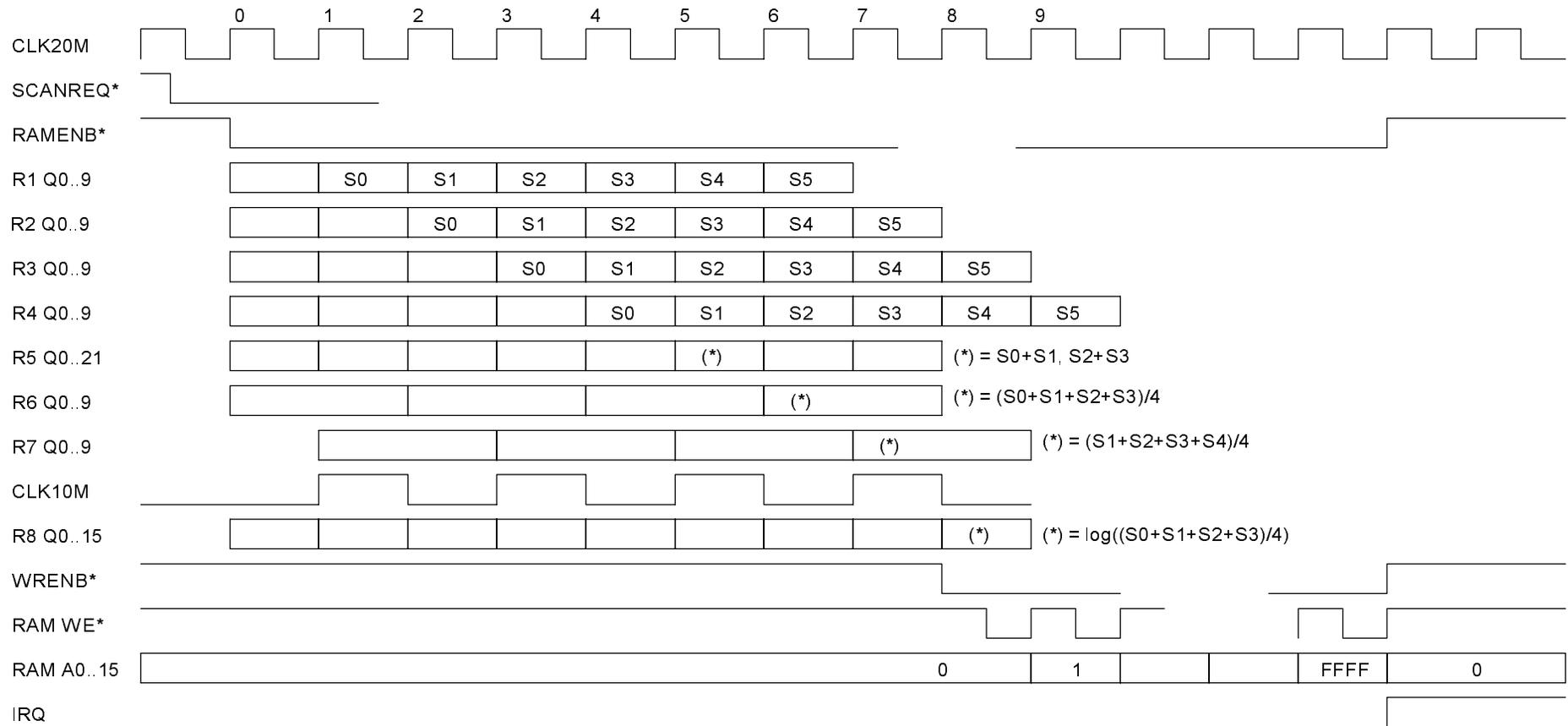
Tempi di elaborazione interstadio dal fronte di CLK20M/CLK10M



$T_{\text{latenza(Pipeline)}} = 8 T_{\text{clk}}$

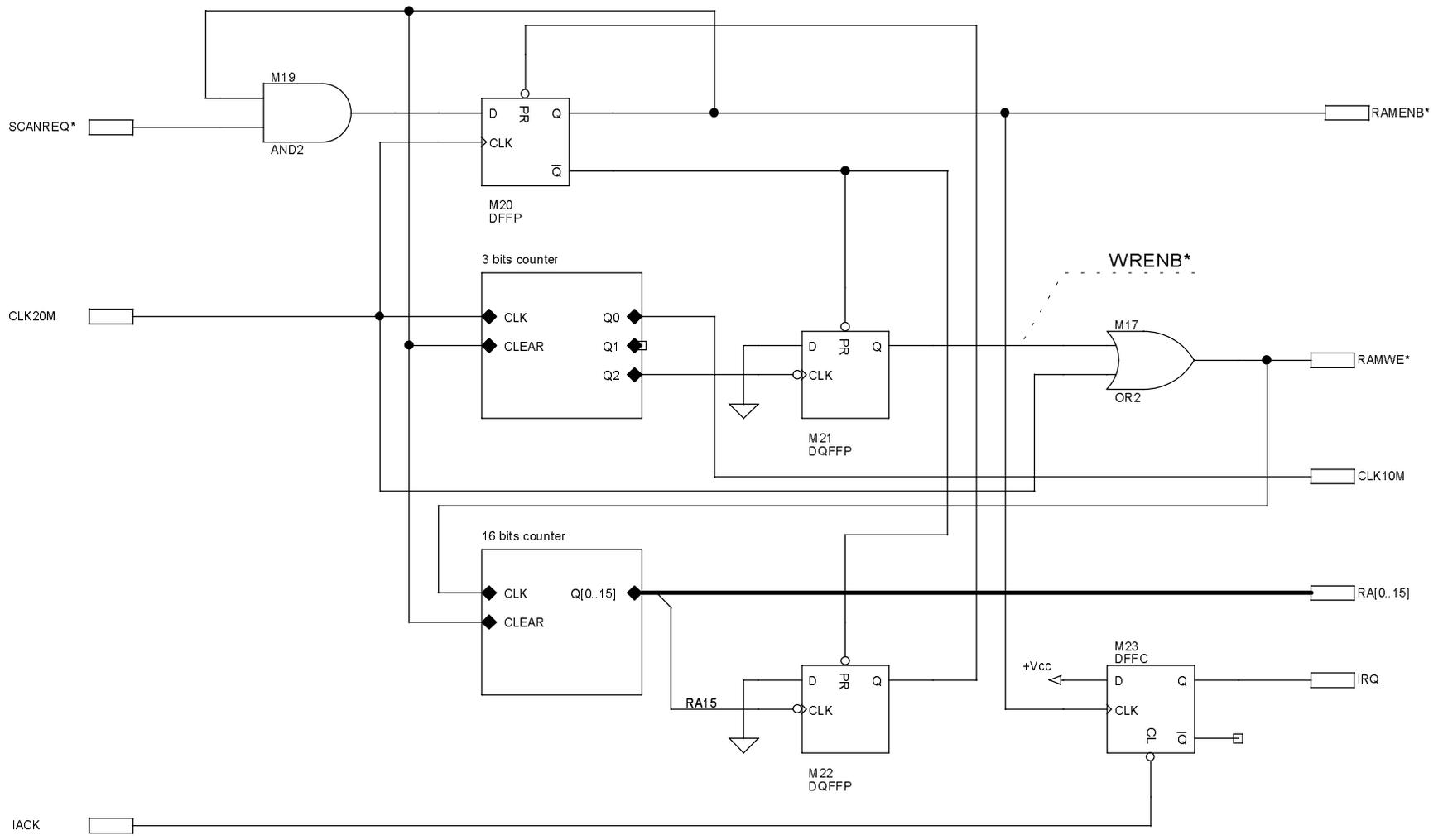
$T_{\text{su,R}} = 5 \text{ ns}$
 $T_{\text{pd,R}} = T_{\text{pd,inv}} = 10 \text{ ns}$
 $T_{\text{pd,adder}} = 10 \text{ ns}$
 $T_{\text{a,rom}} = 70 \text{ ns}$

Title		
ECOSCAN: Pipeline		
Size	Document Number	Rev
A4	1s980416/A.PUGNALONI	1
Date:	Wednesday, July 23, 1997	Sheet 2 of 6

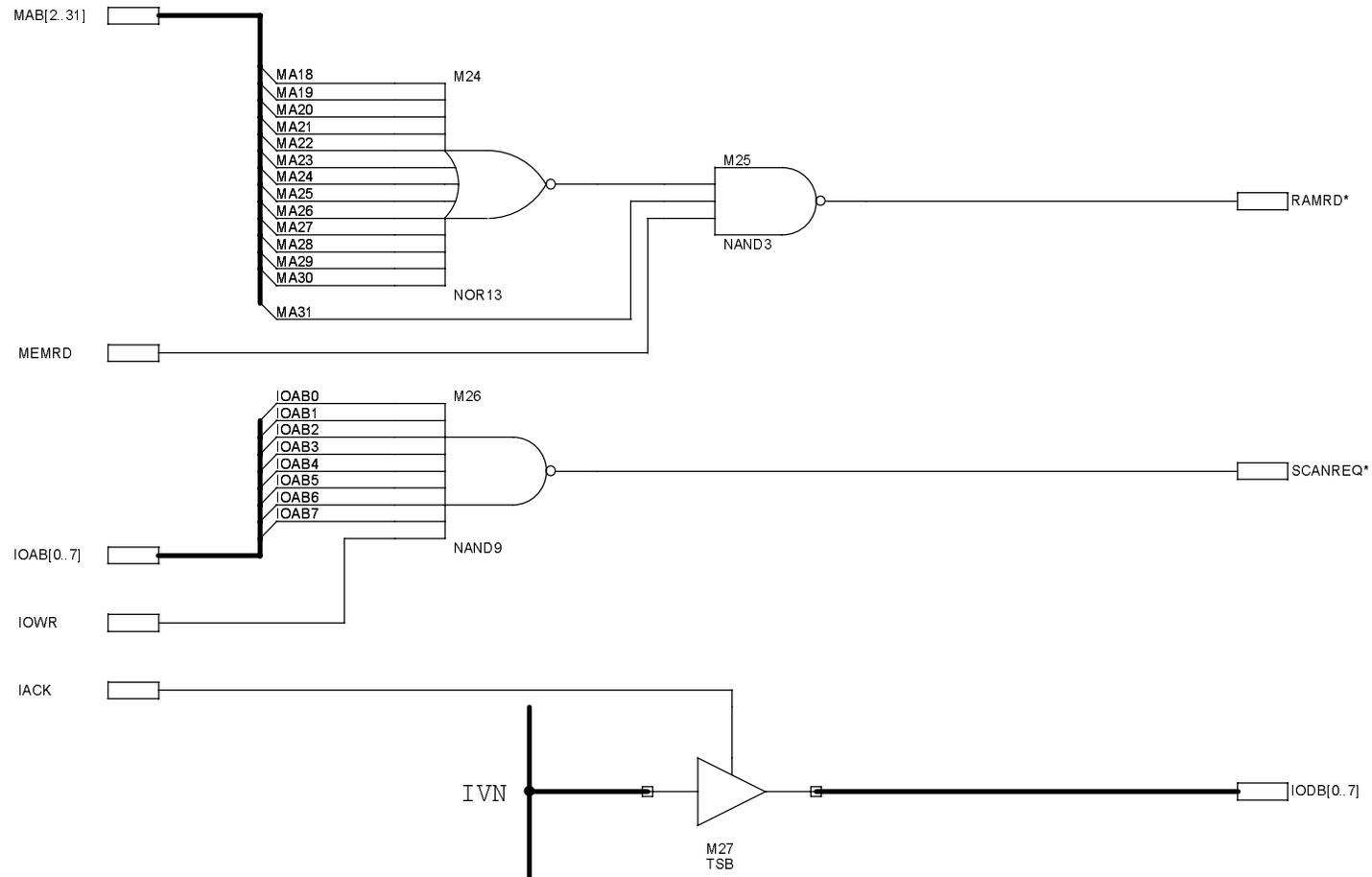


La pipeline produce il 1o dato da memorizzare dopo 8 periodi di clock dall'inizio acquisizione; il segnale WRENB* va attivo in questo istante per abilitare la scrittura in RAM.

La scrittura del 65536o valore provoca il blocco dell'acquisizione e la generazione dell'interrupt verso PD32.



Title		
ECOSCAN: Controller		
Size	Document Number	Rev
A4	1s980416/A.PUGNALONI	1
Date:	Wednesday, July 23, 1997	Sheet 4 of 6

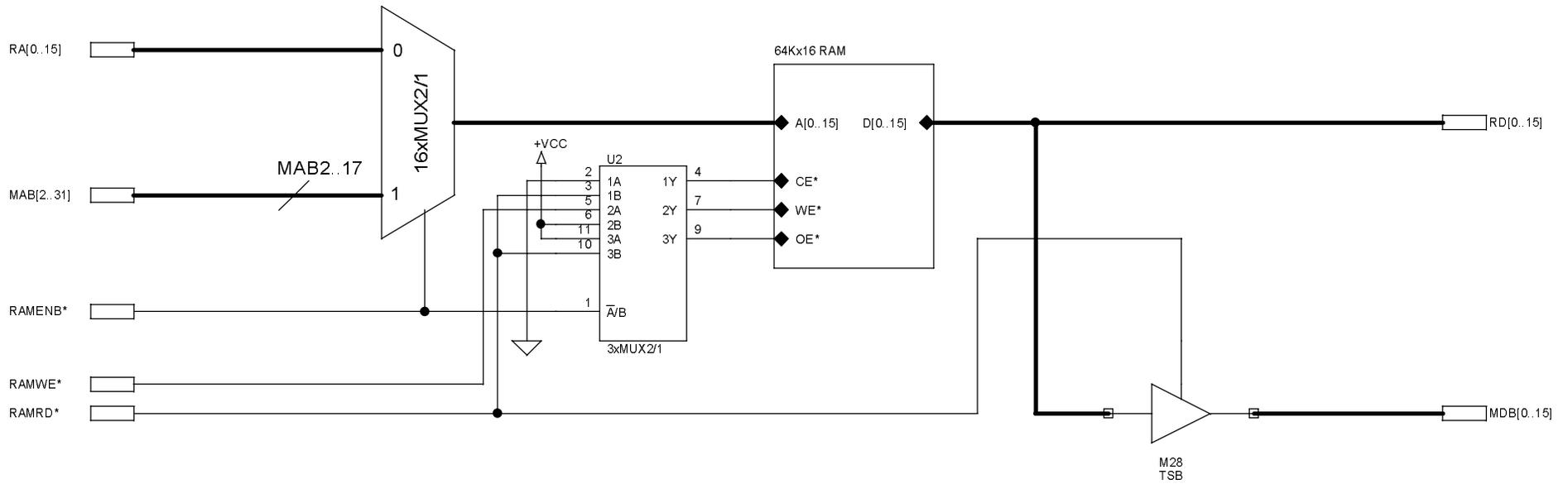


Si ipotizza che il segnale IACK, se necessario, venga generato da una logica esterna di risoluzione delle priorità tra le sorgenti di interruzione.

La RAM viene letta dal PD32 a longword nelle quali la word meno significativa è generata dalla RAM (MDB0..15) e la word più significativa non è utilizzata; l'area di memoria impegnata è quindi di 64K longword = 256KBytes (80000000-8003FFFF)

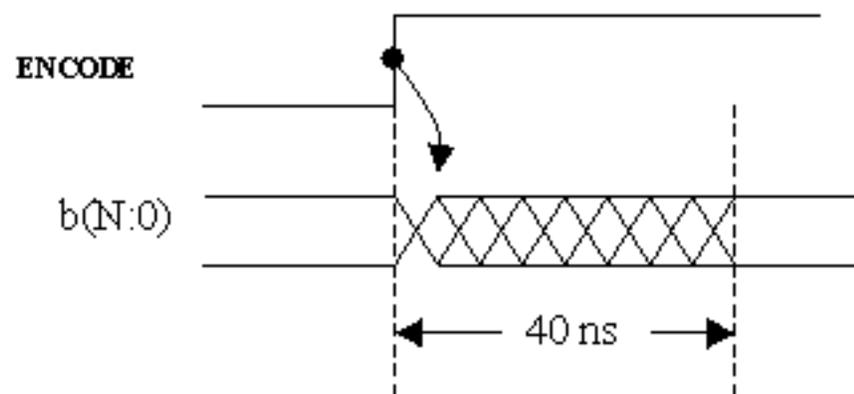
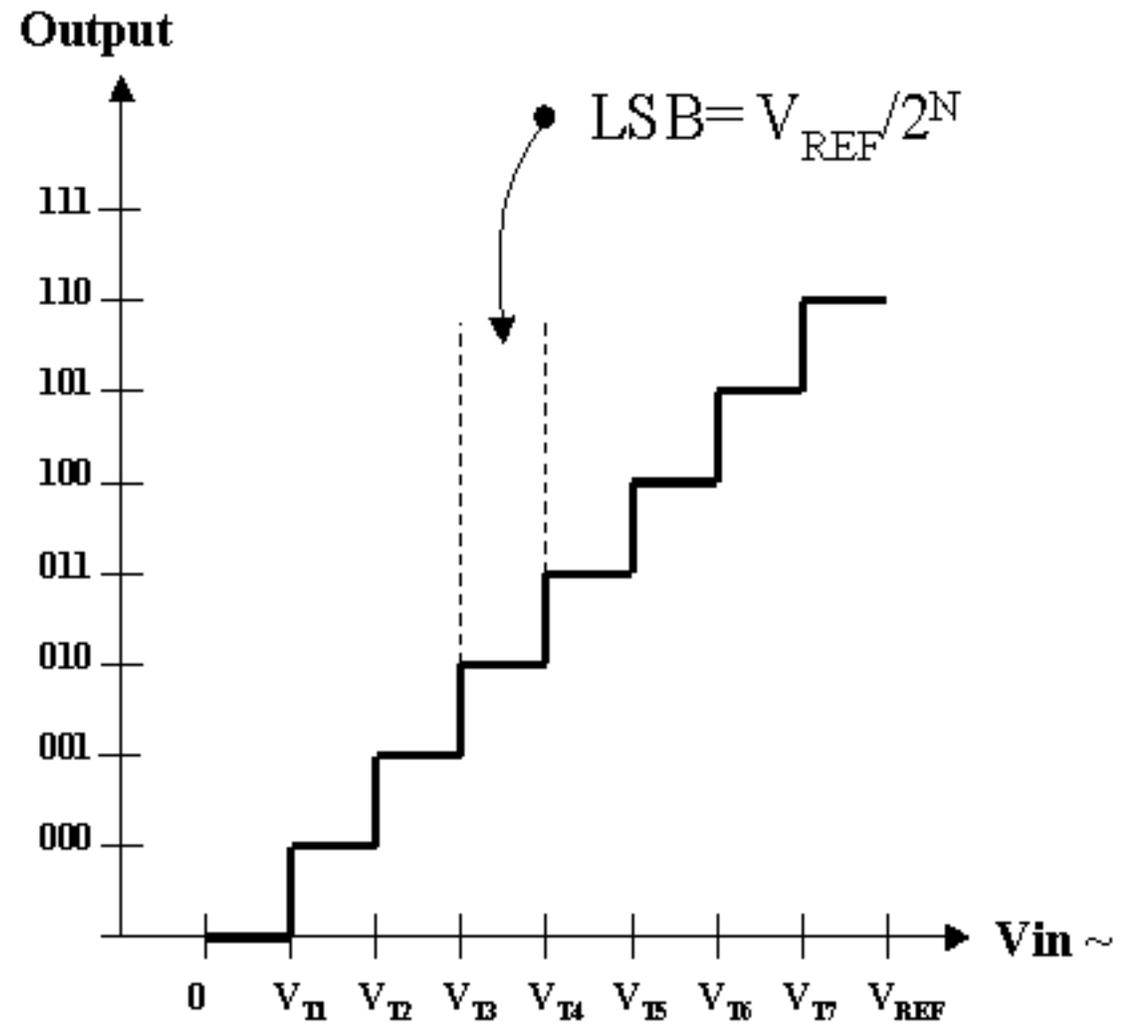
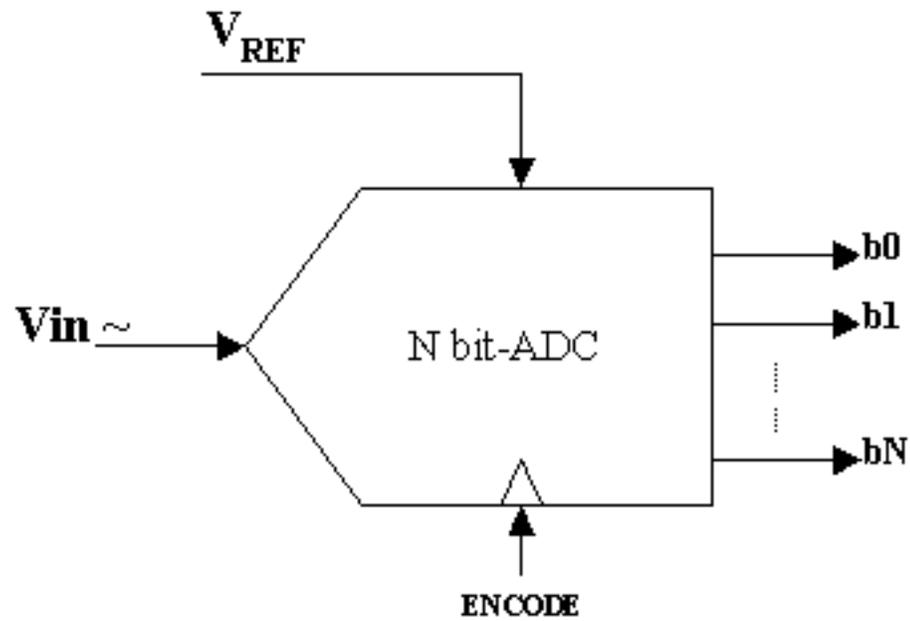
SCANREQ* viene generato eseguendo un'istruzione di OUT all'indirizzo FF

Title		
ECOSCAN: PD32 Interface		
Size	Document Number	Rev
A4	1s980416/A.PUGNALONI	1
Date:	Wednesday, July 23, 1997	Sheet 5 of 6



Title		
ECOSCAN: RAM Interface		
Size	Document Number	Rev
A4	1s980416/A.PUGNALONI	1
Date:	Wednesday, July 23, 1997	Sheet 6 of 6

Convertitore Analogico / Digitale



$LSB = \text{risoluzione del convertitore} < 0.1 \%$

$\text{risoluzione} < (V_{REF}/100) * 0.1 \quad \alpha \quad LSB < V_{REF}/1000$

$V_{REF}/2^N < V_{REF}/1000 \quad \alpha \quad 2^N > 1000 \quad \alpha \quad N \geq 10$

Media Mobile ECOSCAN

Media mobile su 4 campioni => $M(K) = \sum_{i=0,3} S(K+i)/4$

$$M(0) = [S(0) + S(1) + S(2) + S(3)] / 4$$

$$M(1) = [S(1) + S(2) + S(3) + S(4)] / 4$$

$$M(2) = [S(1) + S(2) + S(3) + S(4)] / 4$$

.....

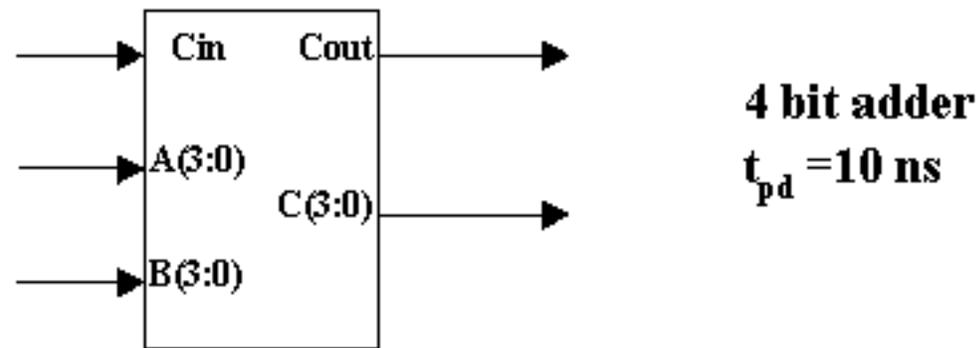
.....

$$M(65535) = [S(65535) + S(65536) + S(65537) + S(65538)] / 4$$

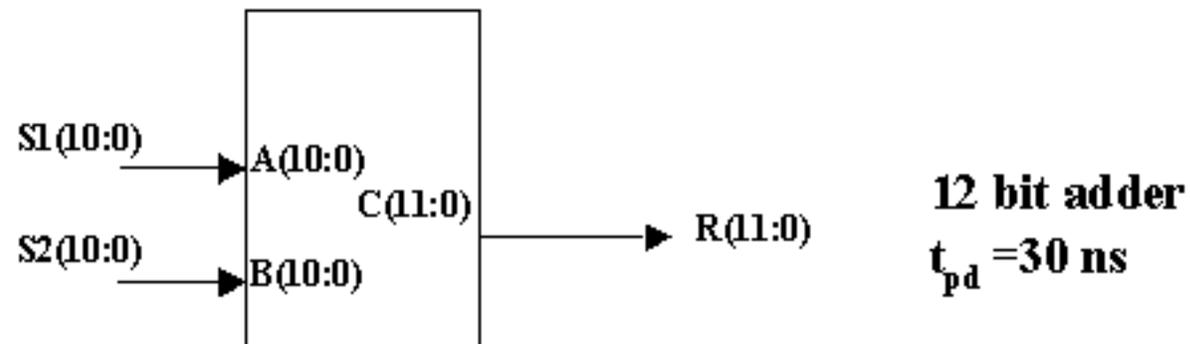
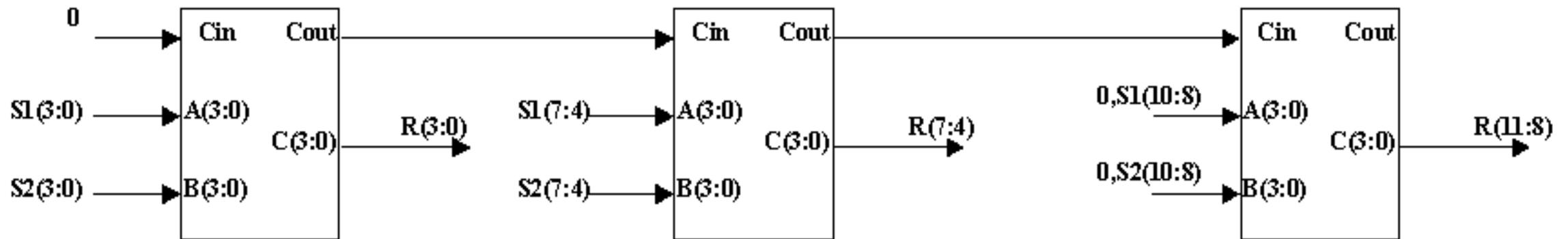
Valore max. possibile della somma campioni = $1023(10 \text{ bit}) * 4 = 4092$
rappresentabile con 12 bit (e' la dimensione dei sommatore)

Il risultato $M(i)$ e' ancora a 10 bit (i bit MSB della somma dei campioni)

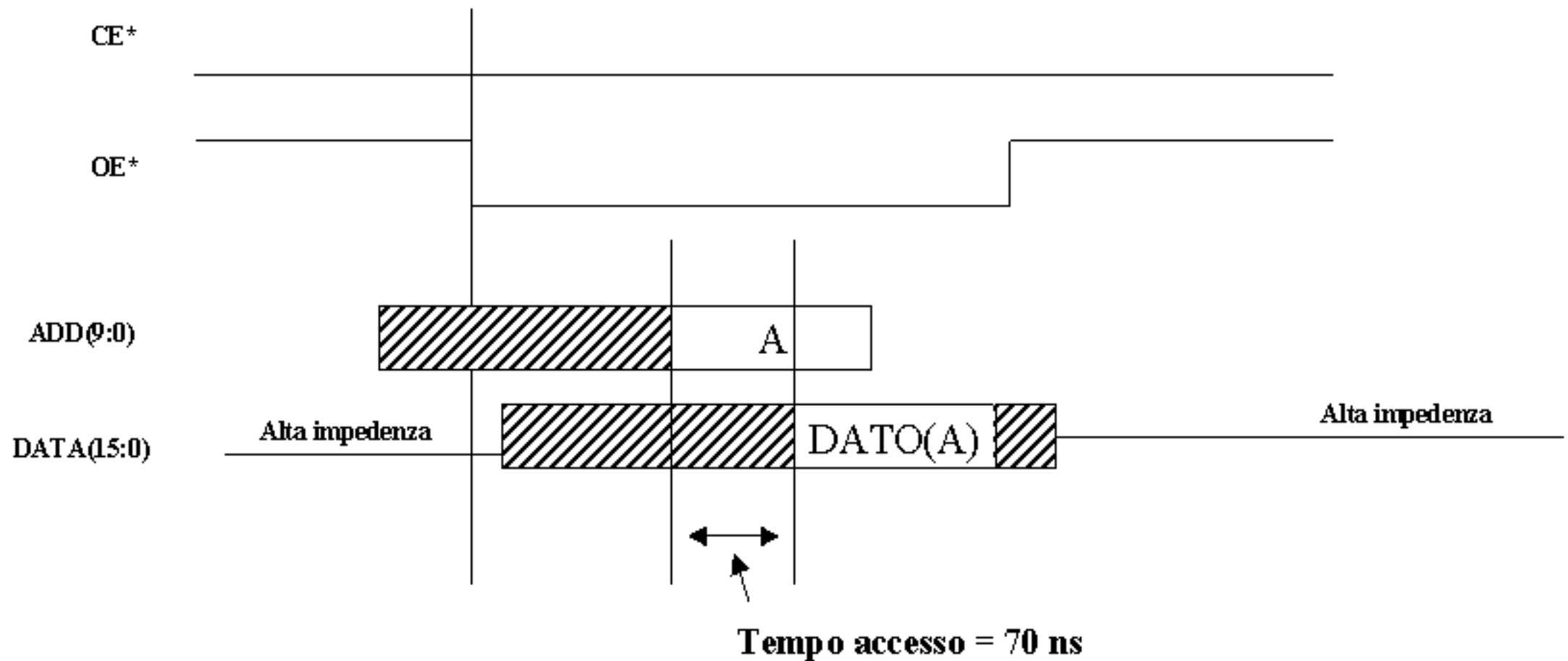
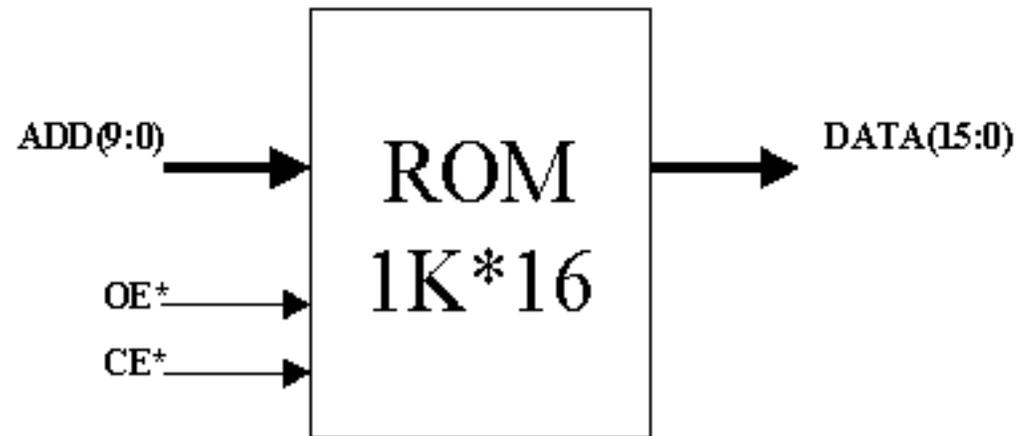
Sommatori 12 bit



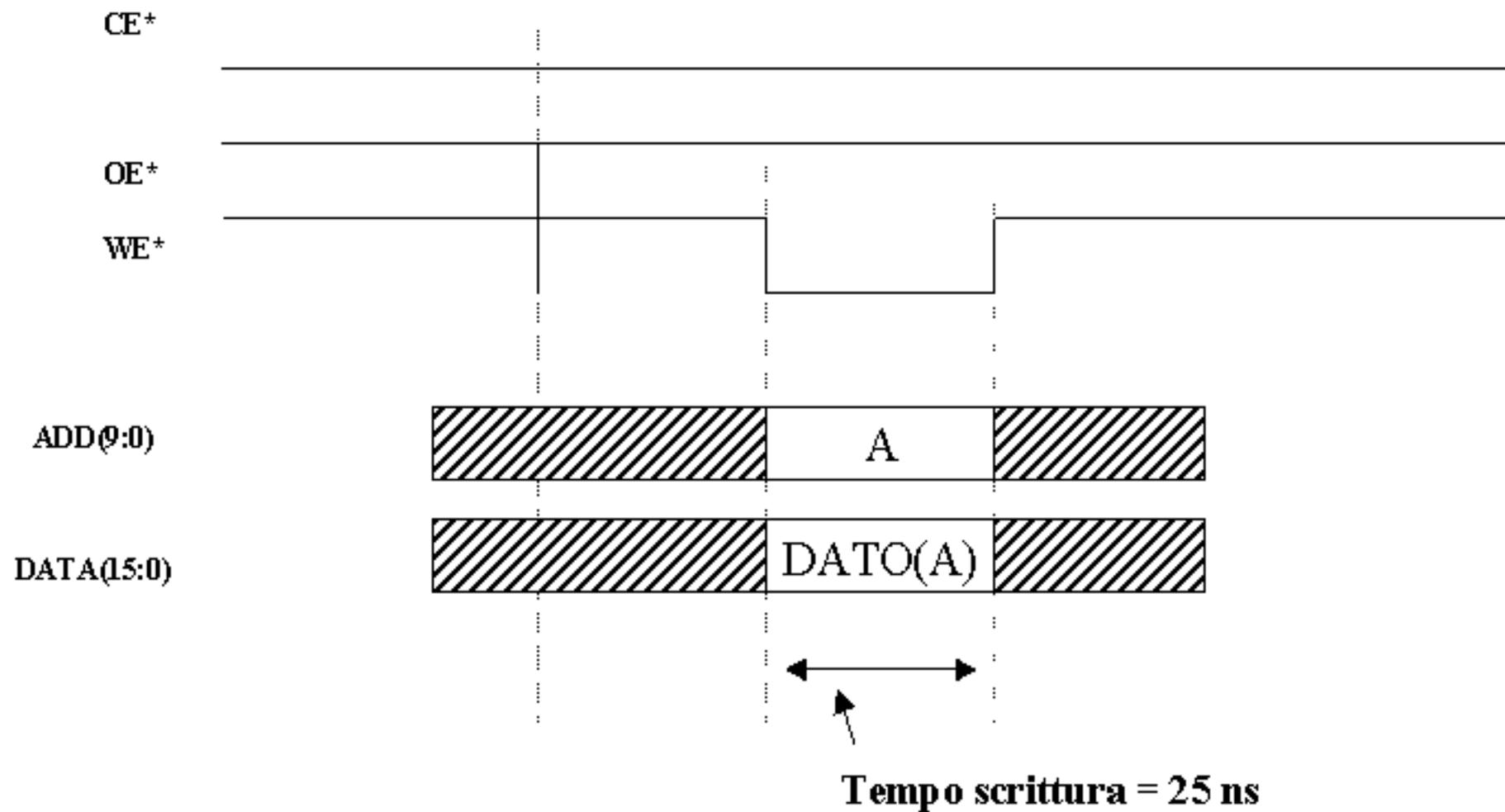
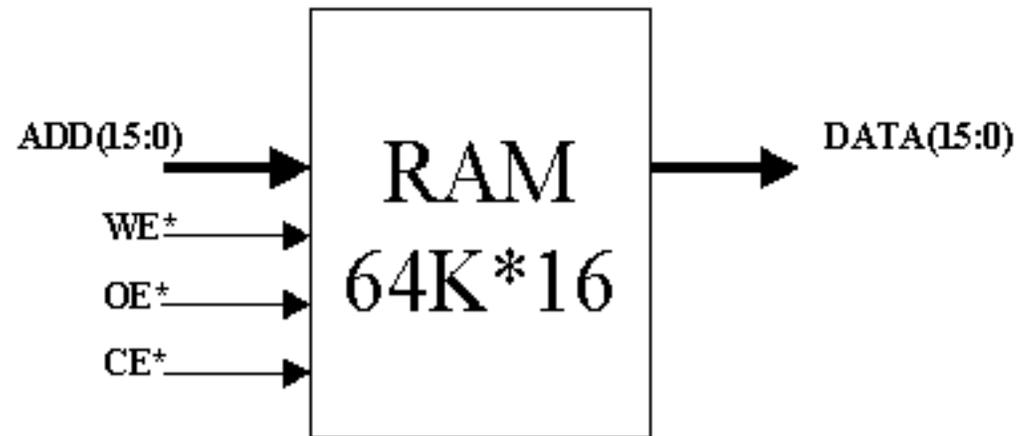
$$R(11:0) = S1(10:0) + S2(10:0)$$



ECOSCAN - Read Only Memory

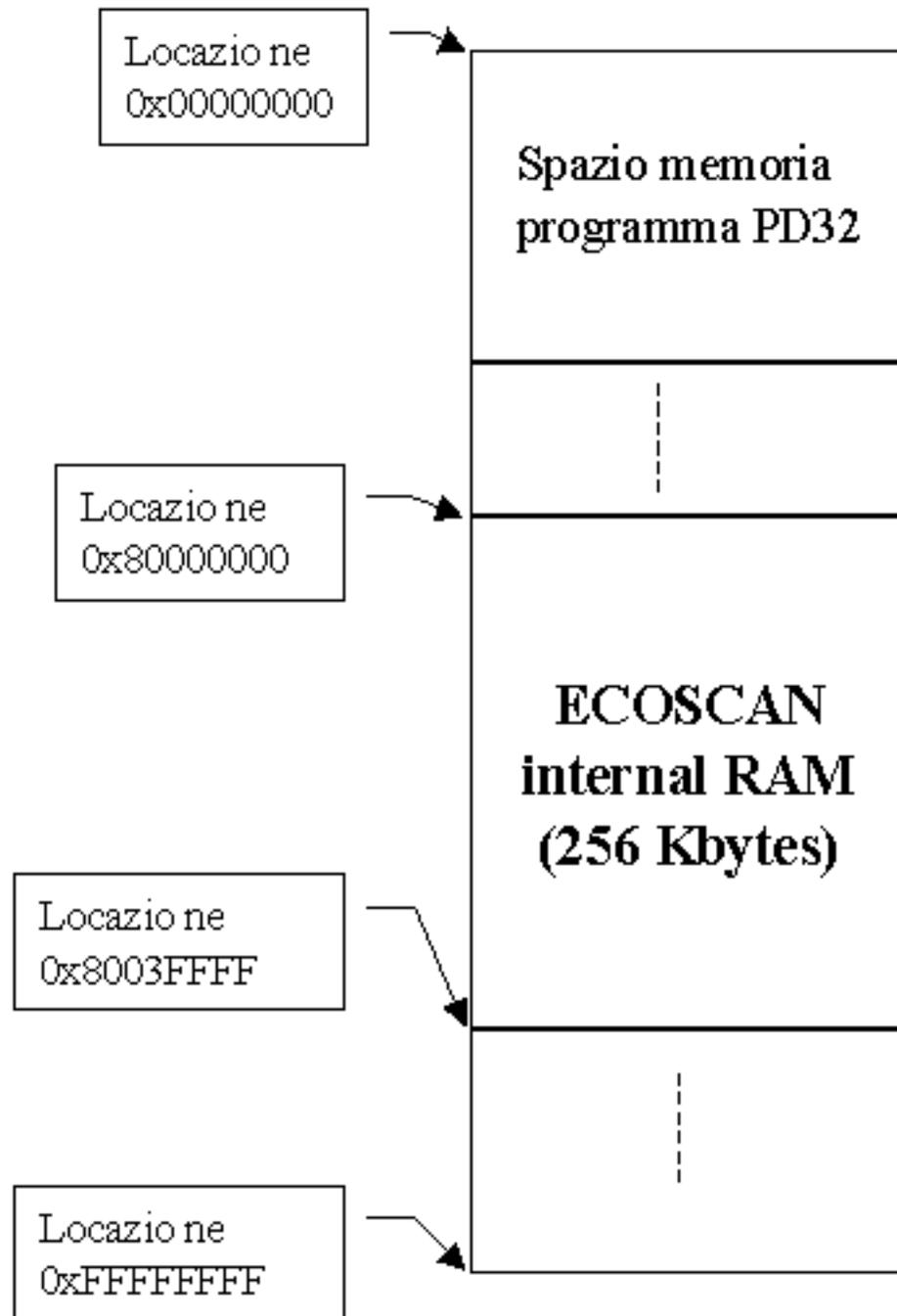


ECOSCAN - Random Access Memory



ECOSCAN Memory Mapping

Spazio di indirizzamento PD32 (byte)

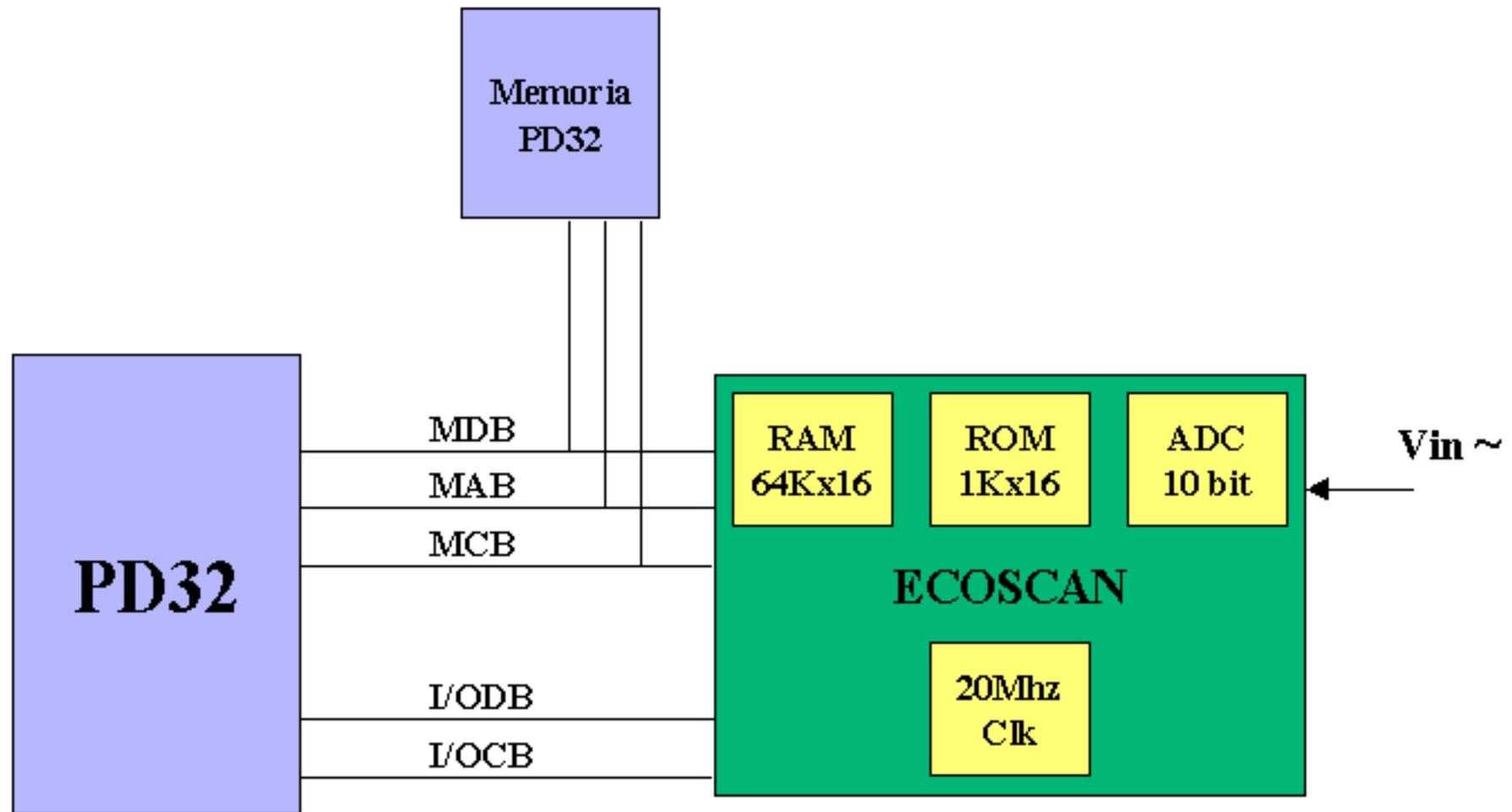


Affinche' il PD32 acceda in lettura la RAM interna ECOSCAN deve "tradurre" gli indirizzi del MAB PD32 in indirizzi validi per la RAM interna

MAB(31:2)	=>	ADD(15:0)
0x20000000		0x0000
0x20000001		0x0001
.....	
.....	
0x2000FFFF		0xFFFF

Poiche' l'accesso e' permesso solo a longword(32 bit) posso trascurare il valore dei segnali Mb(3:0)

Architettura progetto ECOSCAN



Blocchi funzionali unita' ECOSCAN

- **PIPELINE**

- conversione segnale analogico (20 Mhz)
- calcolo media mobile
- calcolo logaritmo

- **RAM INTERFACE**

- gestione segnali RAM interna (ADD,DATA,WE*,OE*,CS*)

- **PD32 INTERFACE**

- decodifica "memory mapping" in lettura (MAB,MWR)
- produzione segnale inizio acquisizione (SCANREQ*)
- generazione IVN durante la fase di richiesta interrupt

- **SCO**

- gestione timing acquisizione (SCANREQ*)
- generazione address e segnali di controllo per scritture valori calcolati
- gestione dei segnali di clock
- gestione Interrupt Request (IRQ*,LACK)

Protocollo ECOSCAN

PD32

ECOSCAN

1) Generazione via software del segnale di inizio acquisizione:

`OUTB S,0xFF`

2) Generazione segnale inizio acquisizione
`SCANREQ*`

3) Processamento campioni (uno ogni 50 ns):

- Conversione A/D
- Calcolo media mobile
- Calcolo logaritmo
- Scrittura in memoria

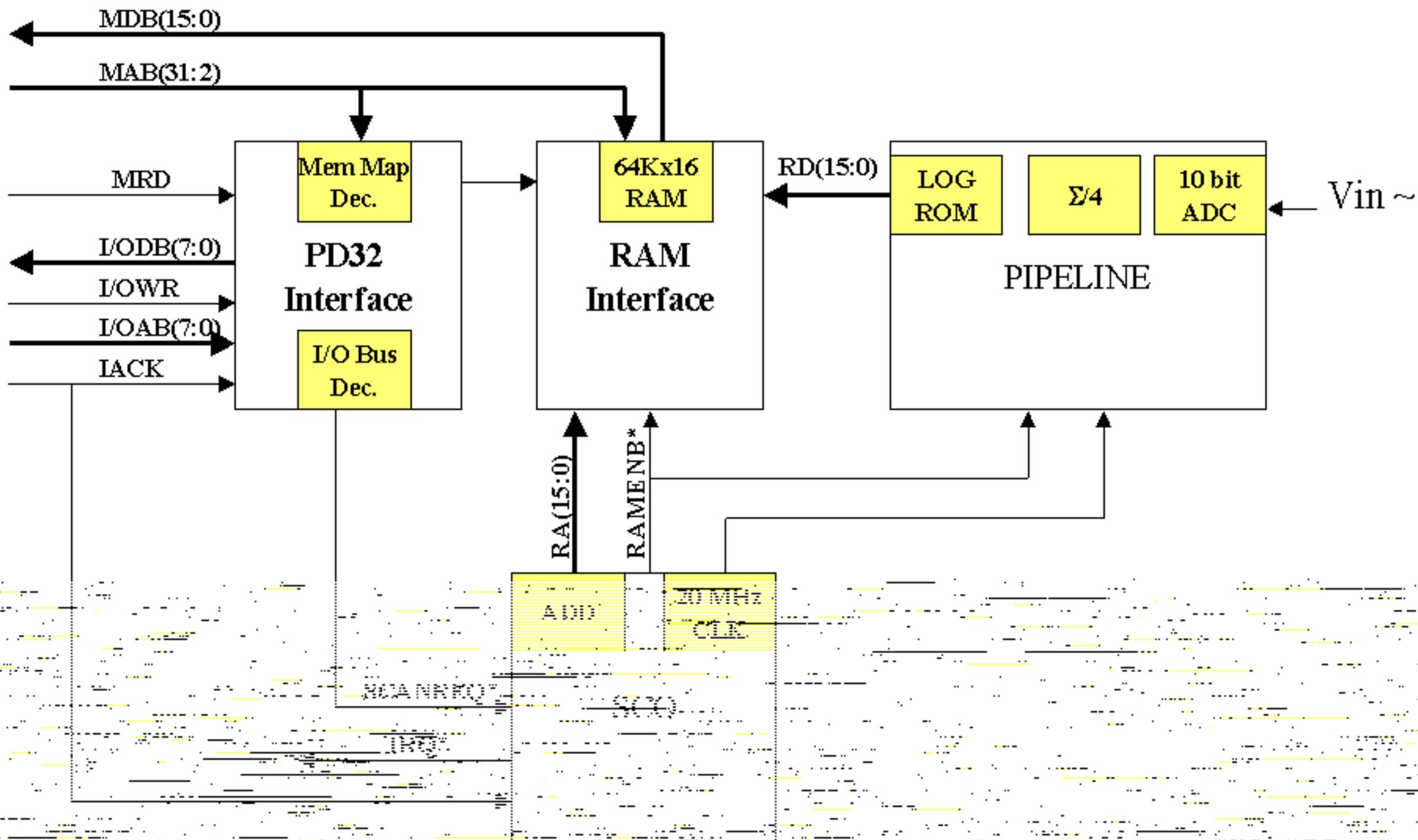
4) Invia un Interrupt al PD32 dopo 65536 scritture

5) Serve la richiesta di interruzione

6) Può accedere ai dati in memoria ECOSCAN

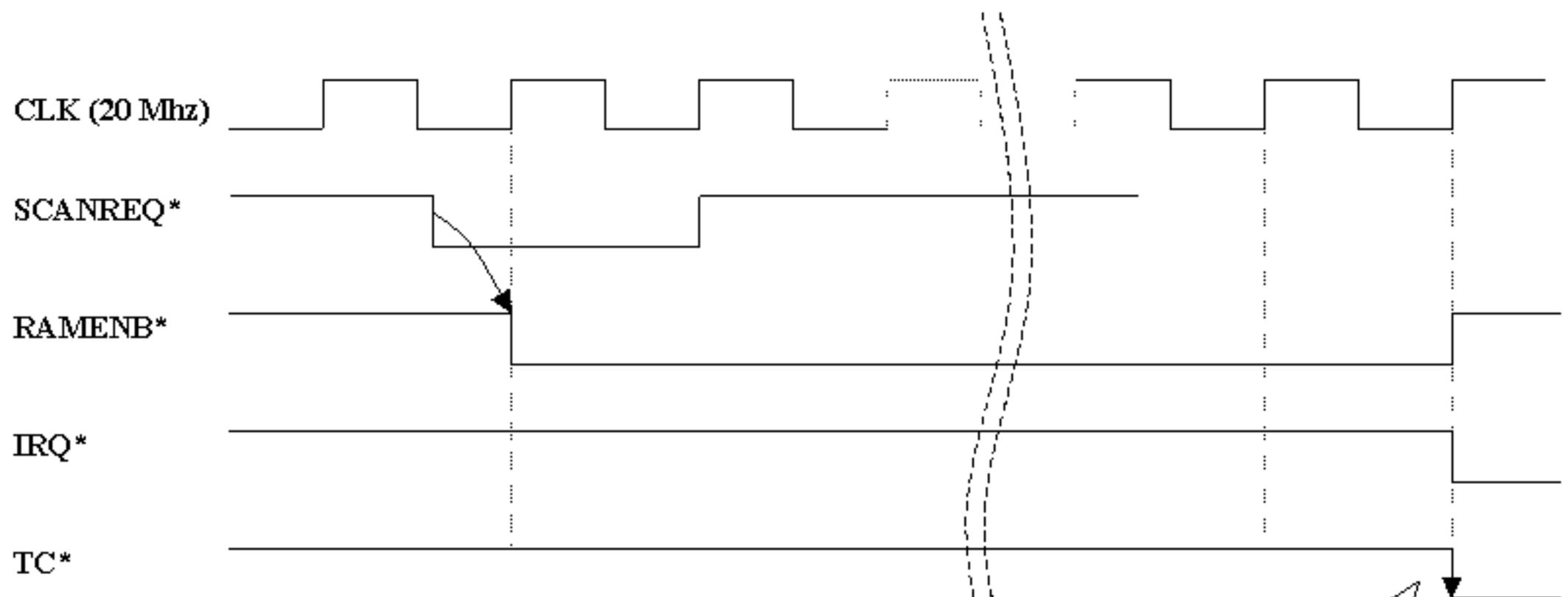
ECOSCAN Schema a blocchi funzionale

- prima approssimazione -

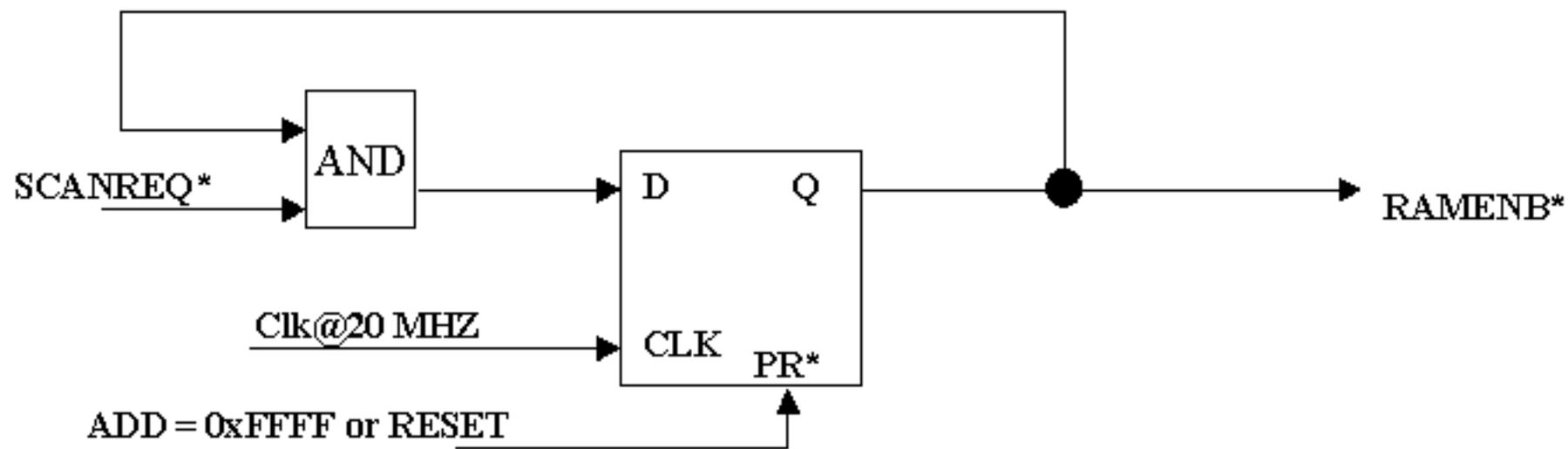


Il segnale RAMENB* sarà attivo (0) solo durante la fase 3 del protocollo
cioè quando la memoria interna è controllata dalla SCO di ECOSCAN

ECOSCAN - SCO (RAMENB* timing)

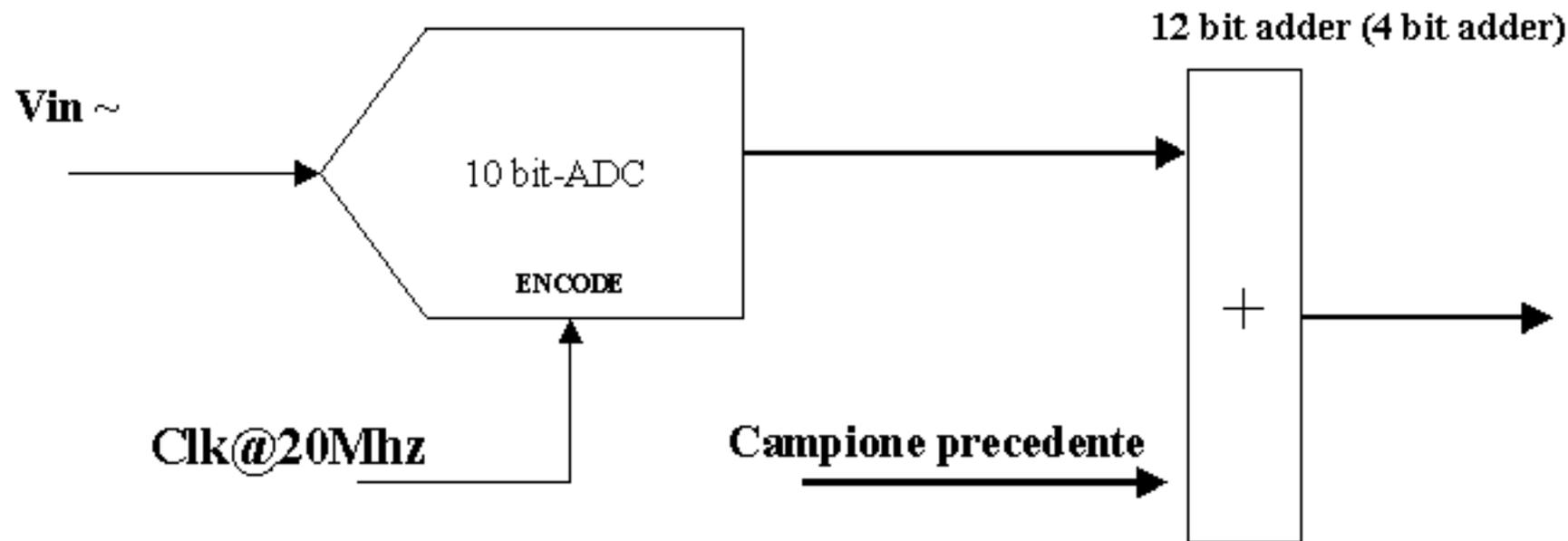


Scrittura ultima
locazione memoria interna
(ADD = 0xFFFF)

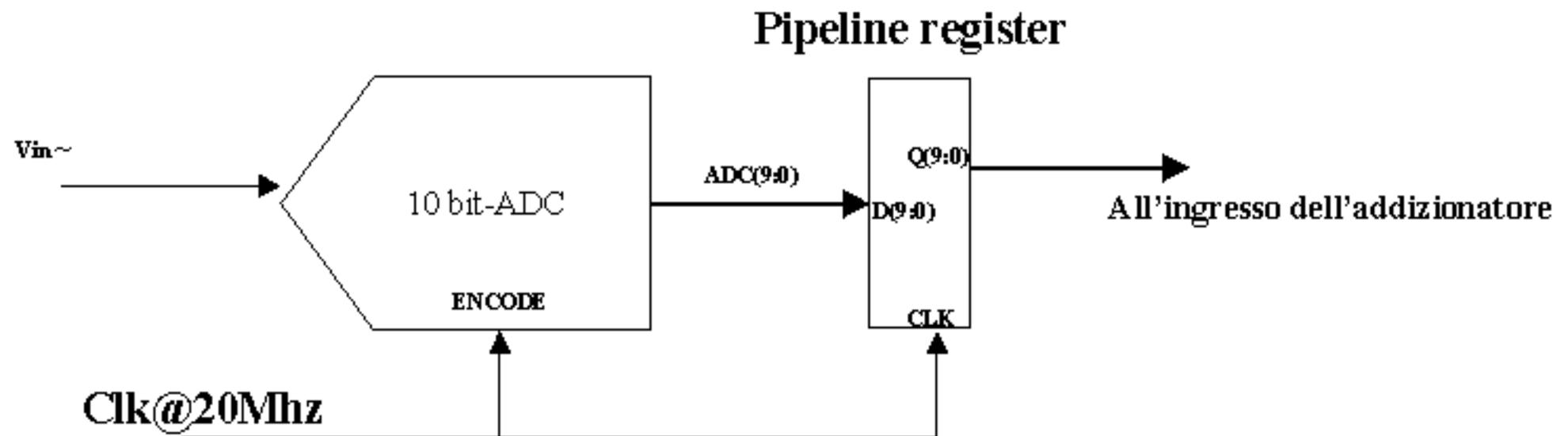


ECOSCAN - Pipeline

Conversione A/D



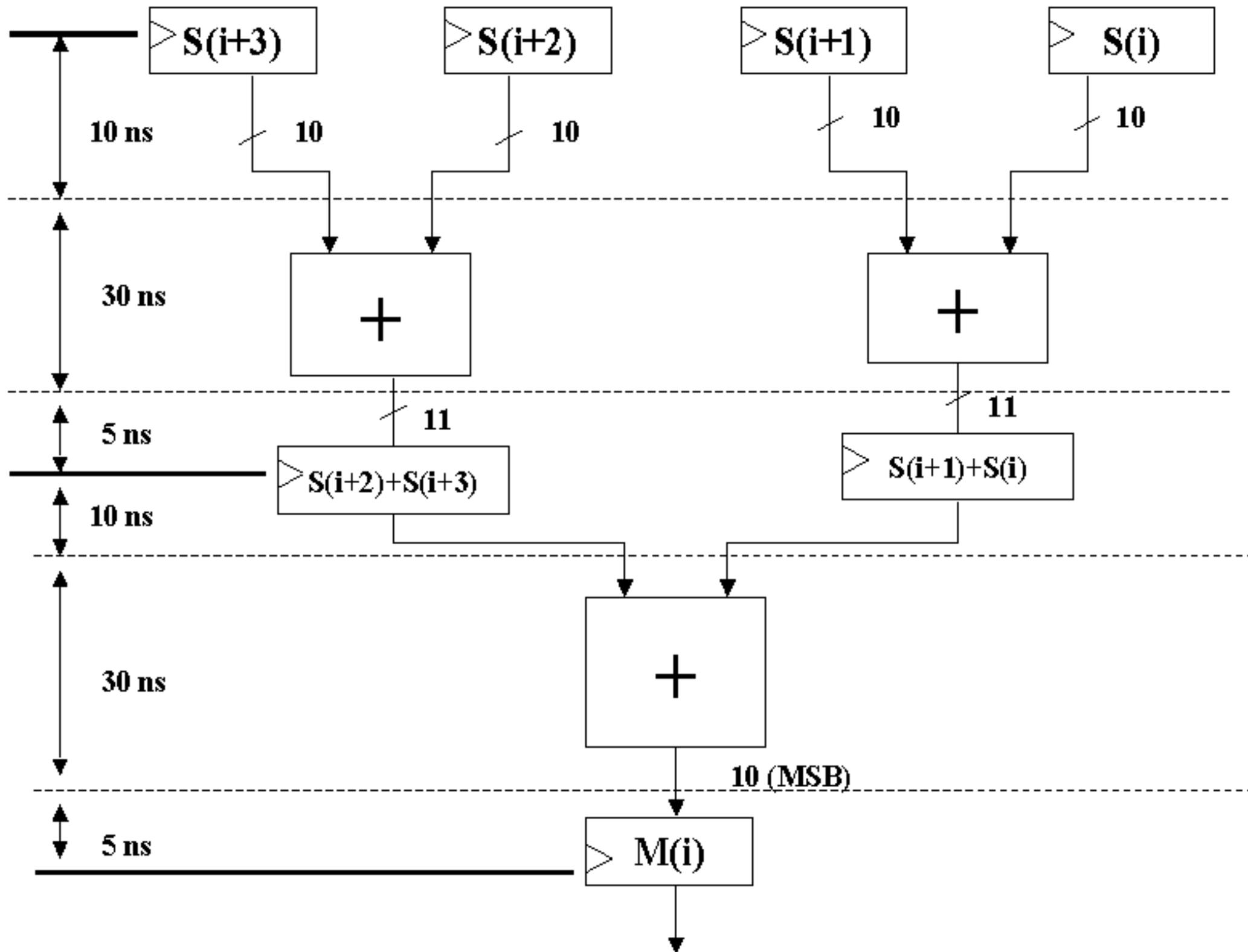
$$40 \text{ ns} + 30 \text{ ns} > 50 \text{ ns (20MHz)}$$



$$\underline{\text{Tempo pipe} = T_{\text{encode}} + T_{\text{su,R}} = 45 \text{ ns}}$$

ECOSCAN - Pipeline

Calcolo Media mobile

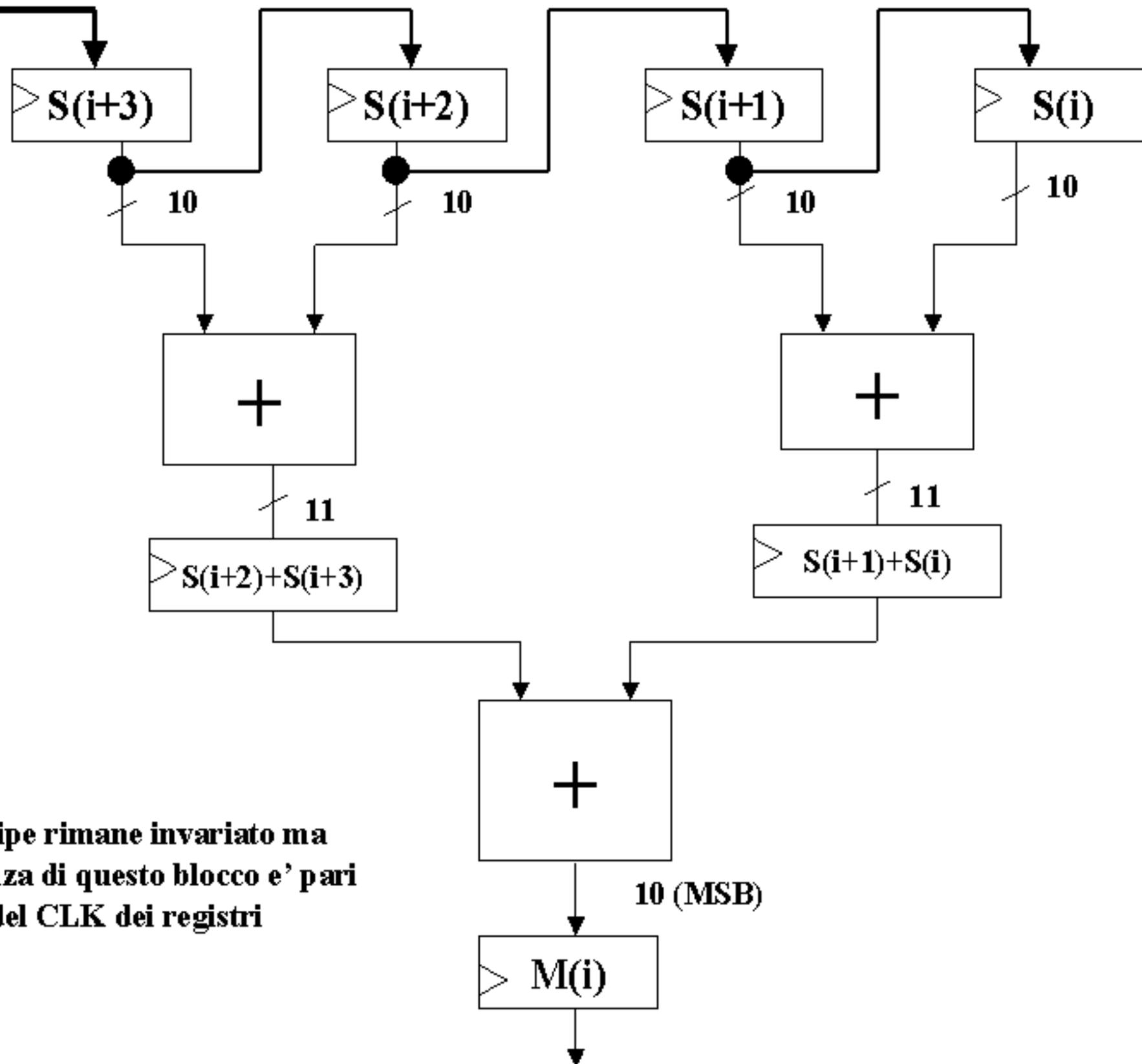


Per entrambe le pipe si ha $T_{\text{pipe}} = T_{\text{pd,R}} + T_{\text{pd,Adder}(12)} + T_{\text{su,R}} = 45\text{ ns}$

ECOSCAN - Pipeline

Calcolo Media mobile

Uscita ADC

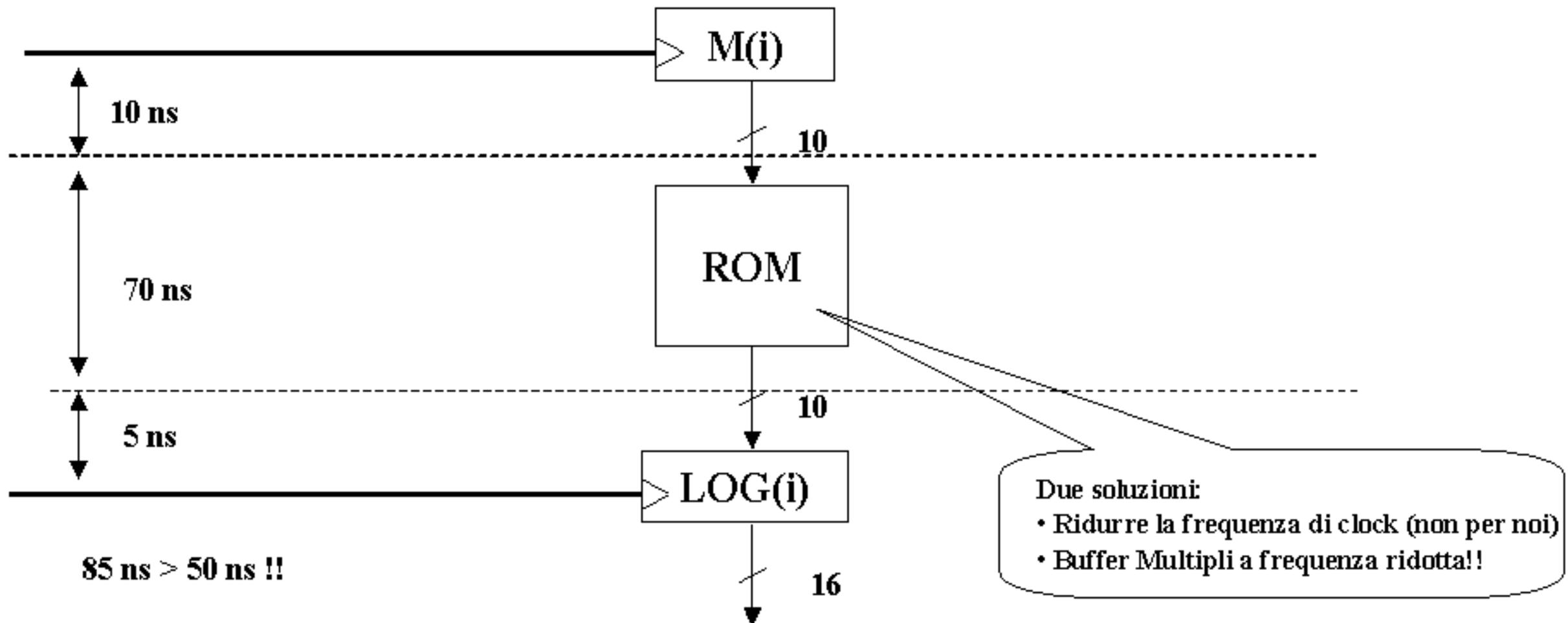


**Il tempo di pipe rimane invariato ma
il tempo di latenza di questo blocco e' pari
a 6 cicli del CLK dei registri**

Ingresso alla LOG-ROM

ECOSCAN - Pipeline

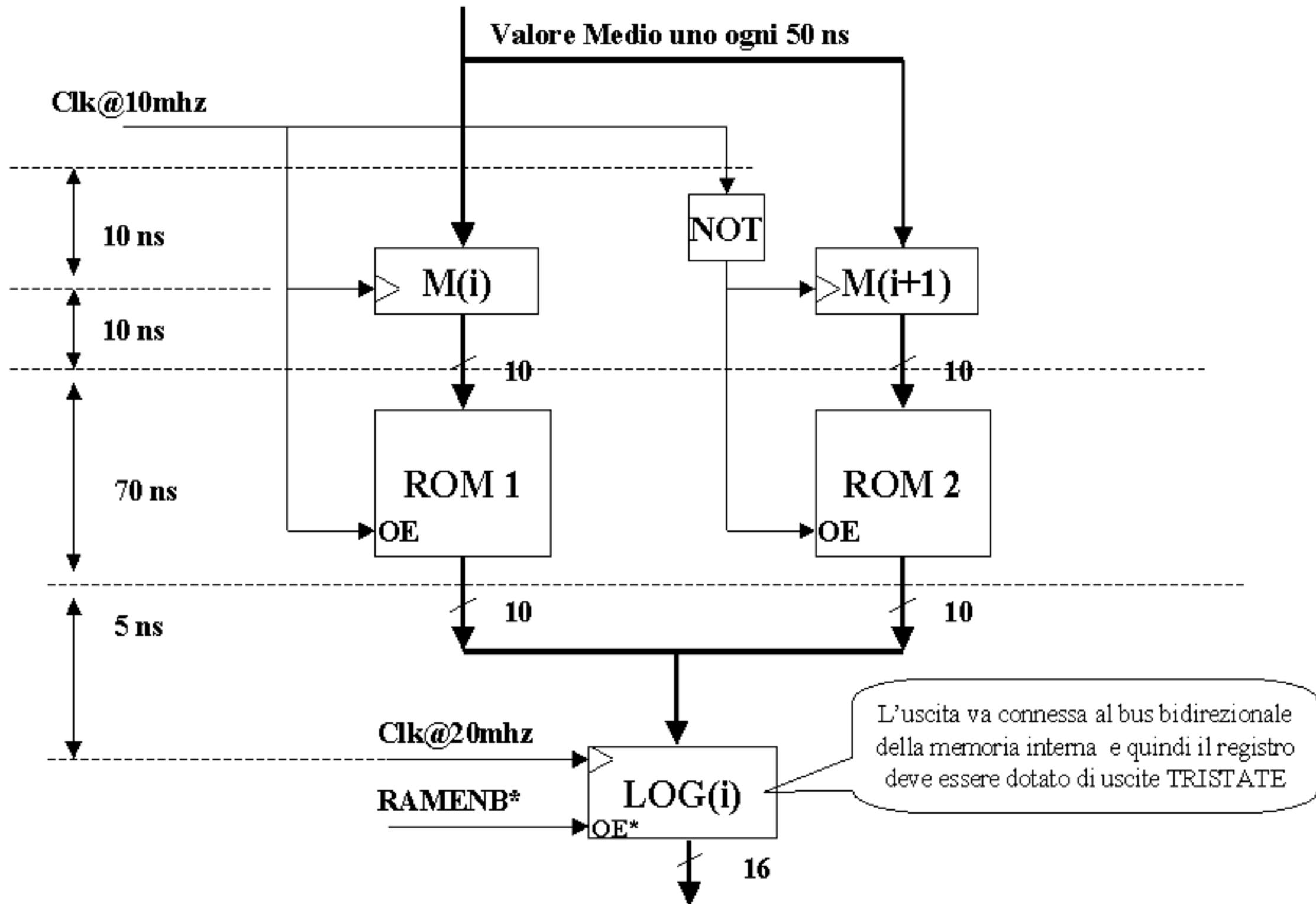
Calcolo Logaritmo



Poiche' il tempo di pipe e' $T_{clk} < t_{pipe} < 2 * T_{clk}$
introduciamo due buffer a frequenza $CLK/2 = 10Mhz$
all'uscita dell'ultimo addizionatore ed utilizziamo due ROM in parallelo

ECOSCAN - Pipeline

Calcolo Logaritmo



$$\underline{T_{pipe(max)} = T_{pd,NOT} + T_{pd,R} + T_{a,ROM} + T_{su,R} = 95 \text{ ns}}$$

La latenza e' pari a 2 cicli del CLK@20Mhz

ECOSCAN - RAM Interface

▪ Gestione segnali SRAM

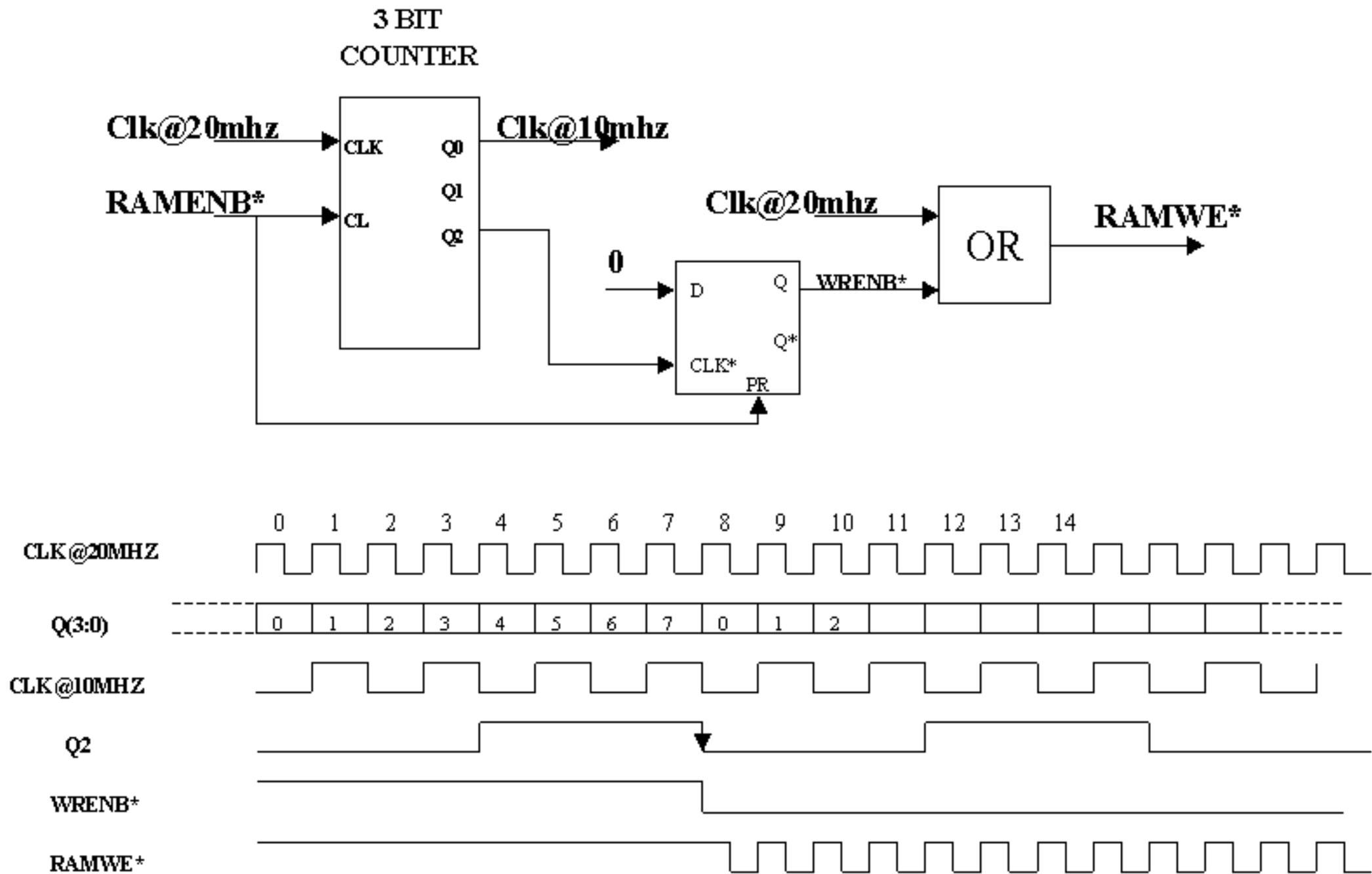
STATO	ADD	CE*	OE*	WE*
RAMENB* = 0	AD[15:0]	0	1	RAMWE*
RAMENB* = 1	MAB(17:0)	RAMRD*(0)	RAMRD*	1

- Segnale RAMRD* = Il PD32 ha indirizzato ECOSCAN
(MAB(31:18) = 0x2000 e MRD = 1)
Proviene dal blocco "PD32 Interface"

- Segnale RAMWE* = ha la stessa temporizzazione di un clk 20 Mhz
ma si deve attivare solo dopo che il tempo di latenza della PIPELINE è

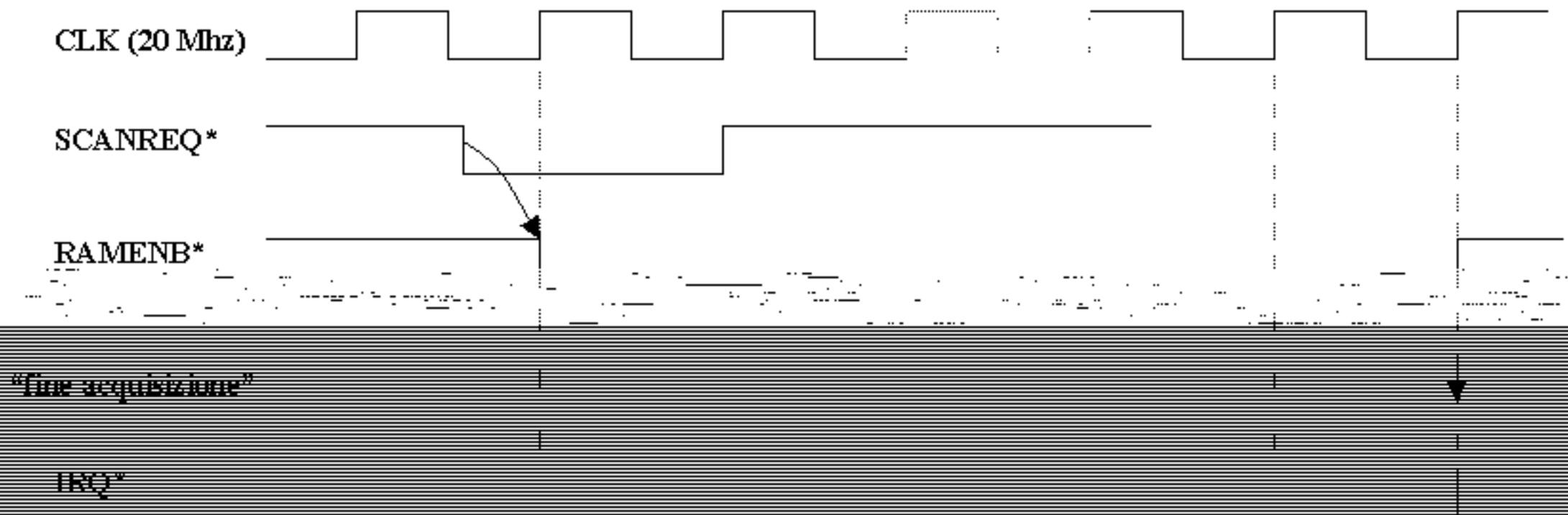
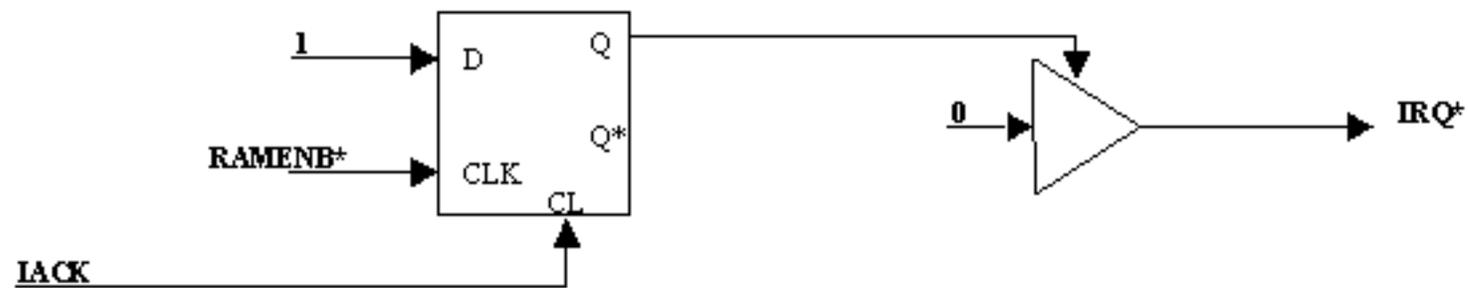
ECOSCAN - SCO

- Generazione Clk (CLK@10Mhz)
- Generazione segnale RAMWE* (Clk@20Mhz ma dopo 8 cicli inizio acquisizione)



ECOSCAN - SCO

- Generazione segnale IRQ* per richiesta interrupt a fine acquisizione



RETI LOGICHE

PRIMA PROVA SCRITTA DEL 08-06-1998

STUDENTE: _____

DOCENTE: _____

Progetto: Interfaccia video grafica GDCONT

Specifiche funzionali

GDCONT è una periferica PD32 dedicata al controllo di un display grafico in bianco e nero da 1024x1024 punti (pixel); lo stato di ogni pixel (acceso/spento) è rappresentato da un bit di una parola della RAM video da 32Kx32 bit contenuta in GDCONT ed accessibile in lettura/scrittura dal PD32 a partire dall'indirizzo di memoria F000000H. GDCONT genera i segnali di controllo del display PIXOUT, HSYNC, VSYNC come segue:

1. PIXOUT trasporta il flusso di 64Mpixel/sec generato serializzando le parole di 32 pixel ottenute dalla scansione ciclica della RAM interna;
2. alla fine di ogni riga di 1024 pixel deve essere generato un impulso su HSYNC, della durata di un periodo di riga, al termine del quale può iniziare la scansione della successiva riga;
3. alla fine di ogni quadro di 1024 righe deve essere generato un impulso su VSYNC, della durata di un periodo di riga, al termine del quale può iniziare la scansione del successivo quadro.

L'accesso del PD32 alla RAM video in concorrenza con GDCONT deve avvenire senza interrompere né differire in alcun modo la temporizzazione di scansione.

Specifiche dispositivi

Il PD32 opera con un clock asincrono rispetto a quello di GDCONT ed il ciclo di Memory R/W ha una durata di 400ns; la RAM video ha un tempo di accesso di 200ns.

Note/suggerimenti:

Per risolvere i conflitti di accesso alla RAM video condivisa si divida in due il periodo disponibile per la scansione di una parola riservandone metà al PD32 e metà a GDCONT; è necessaria una logica di sincronizzazione del PD32 (usare il segnale di controllo wait).

Si richiedono:

1. lo schema a blocchi di GDCONT con i necessari segnali di interfacciamento verso il PD32;
2. il diagramma di timing dettagliato che specifichi il funzionamento di GDCONT e la tecnica adottata per garantire l'accesso alla RAM video condivisa;
3. lo schema elettrico completo di GDCONT.

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 18-06-1998

STUDENTE: _____

DOCENTE: _____

Progetto:

Interfaccia di monitoraggio di valore di picco PEAKMON

Specifiche funzionali:

PEAKMON è un'interfaccia PD32 dedicata al monitoraggio del valore di picco raggiunto da una funzione numerica di un segnale analogico. PEAKMON deve eseguire le seguenti elaborazioni numeriche mediante una pipeline:

1. campionare il segnale analogico d'ingresso ad una frequenza di 33MHz con una precisione migliore di 0.025% per ottenere la sequenza di campioni $X[kT]$;
2. comprimere i campioni secondo una legge logaritmica ottenendo la sequenza $L[kT] = \log_{10}(X[kT])$ di valori a 14bit;
3. calcolare la funzione $F[kT] = 0.75 L[(k-2)T] + 0.50 L[(k-1)T] + L[kT]$;
4. se $F[kT] \leq F_{peak}$ il valore $F[kT]$ deve essere trascurato;
5. se $F[kT] > F_{peak}$ deve scrivere F_{peak} in una FIFO, generare un'interruzione verso il PD32 e porre $F_{peak} = F[kT]$;

Il PD32, sollecitato dalle interruzioni provenienti da PEAKMON, provvederà a leggere i(l) valori(e) F_{peak} accumulati(o) nella FIFO (durante il tempo di latenza dell'interruzione l'interfaccia può generare ulteriori valori di F_{peak} oltre a quello che ha inizialmente provocato l'interruzione). Il valore di picco corrente F_{peak} , mantenuto in un registro interno di PEAKMON, deve poter essere azzerato su comando del PD32.

Specifiche dispositivi:

Il convertitore A/D Flash da utilizzare prevede l'unico segnale di controllo Encode ed ha un tempo di conversione $T_{ENCODE} = 30ns$. La tabella di conversione logaritmica è contenuta in ROM con $T_A = 75ns$. I sommatore da utilizzare per il calcolo della $F[kT]$ sono moduli a 4 bit con $T_C = 5ns$. Tutti gli altri dispositivi logici hanno $T_{PD,MAX} = 5ns$. La FIFO è dotata di ingresso e uscita dati separati, dei segnali F_{read} , F_{write} , per leggere e scrivere nella fifo e dei segnali F_{piena} , F_{vuota} che segnalano lo stato della FIFO.

Note/suggerimenti:

Nel calcolo di $F[kT]$ si tenga conto che i fattori 0.50 e 0.75 sono ottenibili come potenze frazionarie o combinazione di potenze frazionarie di 2.

Si richiedono:

1. lo schema a blocchi di PEAKMON con l'indicazione dei necessari segnali di interfaccia verso il PD32;
2. l'architettura della pipeline, le dimensioni delle ROM di conversione logaritmica, i tempi di calcolo massimi di ogni stadio ed il tempo di latenza;
3. il diagramma dettagliato di timing che specifichi il funzionamento di PEAKMON;
4. lo schema elettrico completo di PEAKMON.

Reti Logiche

Seconda prova scritta del 18-6-98

Studente: _____ **Docente:** _____

1. Illustrare la metodologia per aggiungere ad un pacchetto dati di n bit il pacchetto di controllo per ottenere una parola di codice CRC. Il polinomio generatore è $G(x) = x^{16} + x^{12} + x^5 + 1$.
2. Progettare una rete con due ingressi x_1 e x_2 e due uscite y_1 e y_2 :
 $y_1 = x_1$ e $y_2 = x_2$ se $x_1 \geq x_2$, altrimenti $y_1 = x_2$ e $y_2 = x_1$
3. Descrivere la struttura di una rete sincrona con due ingressi a livello e due impulsivi.
4. Illustrare la temporizzazione di un sistema SCO-SCA di tipo DMealy-DMealy.
5. Una periferica del PD32 è una memoria FIFO con parole di 8 bit che alza il segnale di interruzione quando è vuota. Definire la routine di scrittura della FIFO. La FIFO è dotata di ingresso e uscita dati separati, dei segnali Fread, Fwrite, per leggere e scrivere nella fifo e dei segnali Fpiena, Fvuota che segnalano lo stato della FIFO.

RETI LOGICHE

PRIMA PROVA SCRITTA DEL 02-07-1998

STUDENTE: _____ DOCENTE: _____

Progetto: LANDMA, controllore DMA per trasferimento dati PD32 \Rightarrow Local Area Network

Specifiche funzionali

LANDMA è un coprocessore dedicato al trasferimento seriale unidirezionale di dati fra PD32 ed un controllore di rete LANCTL. LANDMA è alimentato dallo stesso clock del PD32 ospite e si interfaccia al LANCTL mediante i seguenti segnali sincroni con SCLOCK (sempre attivo):

Segnale	Da	A
TXDATA, TXRTS,	LANDMA	LANCTL
TXCTS, SCLOCK	LANCTL	LANDMA

Il protocollo di trasferimento dati è organizzato a blocchi di 2048 bit come segue:

1. quando PD32 vuole spedire un blocco dati, invia a LANDMA l'indirizzo iniziale di memoria;
2. quando è disponibile a ricevere, la destinazione (LANCTL) attiva il segnale TXCTS;
3. la sorgente (LANDMA) può quindi attivare il segnale TXRTS e trasmettere serialmente l'intero blocco di bit senza alcuna pausa; LANDMA provvede al prelievo delle longword dalla memoria del PD32 in DMA;
4. al termine della trasmissione del blocco LANDMA disattiva il segnale TXRTS e genera un'interruzione verso il PD32.

LANDMA è controllato da un SCO D-MEALY microprogrammato. Il trasferimento in DMA deve avvenire a blocchi (non in bus_stealing).

Specifiche dispositivi:

I cicli di memory Read/Write del PD32 hanno una durata di 3 stati macchina.

Si richiedono:

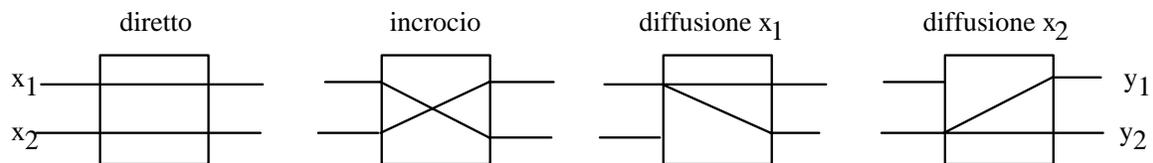
1. lo schema a blocchi di LANDMA con l'indicazione dei necessari segnali di interfaccia verso il PD32;
2. il diagramma dettagliato di timing che specifichi il funzionamento delle interfacce di LANDMA con LANCTL e PD32;
3. lo schema elettrico completo di LANDMA ed il microprogramma di controllo del SCO;
4. la massima velocità della linea seriale gestibile da LANDMA.

Reti Logiche

Seconda prova scritta del 2 luglio 1998

Studente: _____ Docente: _____

1. Progettare una rete n due ingressi x_1 e x_2 , due uscite y_1 e y_2 e due variabili di controllo c_1 e c_2 che permettono di avere un collegamento ingresso uscita secondo gli schemi di figura:



2. Descrivere la struttura di un addizionatore a propagazione rapida del riporto per dati di 32 bit utilizzando moduli di 4 bit.

3. Una rete asincrona a due ingressi x e y deve comportarsi nel modo seguente: sul fronte positivo di x (y) viene campionato e trasferito in uscita il valore di y (x). Si consiglia di sintetizzare la rete utilizzando il flip-flop T generalizzato.

4. Un microprogramma è costituito da 16 microistruzioni di cui quattro sono di branch a due vie. Le condizioni di test sono definite dall'AND di due delle quattro variabili di condizione. Realizzare la SCO con una PLA ed un registro di flip-flop T: dimensionare la PLA.

5. Data una CPU PD32 con ciclo macchina di 3 stati e clock a 10 Mhz, determinare la frequenza limite di trasferimento dati in DMA con modalità bus-stealing di una periferica che produce dati a 32 bit

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 14-07-1998

STUDENTE: _____

DOCENTE: _____

Progetto:

KBDCNT, controllore di tastiera alfanumerica

Specifiche funzionali:

Una tastiera alfanumerica per PC da 96 tasti è realizzata mediante una matrice di interruttori da 6 righe per 16 colonne; la pressione del generico tasto provoca la chiusura del circuito tra la riga e la colonna alle quali il corrispondente interruttore è connesso. KBDCNT esegue ciclicamente la scansione righe/colonne della matrice di interruttori trattando le possibili condizioni incontrate come segue:

1. se al termine di una scansione KBDCNT ha rivelato la chiusura di 2 o più interruttori (più tasti premuti contemporaneamente), la scansione viene considerata non valida e deve quindi essere ripetuta fino a quando non venga trovato un solo contatto chiuso;
2. se per 2 scansioni consecutive KBDCNT rivela la chiusura di un determinato contatto (e solo di quello) deve:
 - trasformare la coppia riga/colonna corrispondente al tasto premuto nel corrispondente codice ASCII ad 8 bit
 - trasferire il codice ASCII del tasto in un registro di uscita e generare un'interruzione verso il PD32 cui KBDCNT è interfacciato
 - attendere il servizio dell'interruzione e riprendere il ciclo di scansione

KBDCNT è sincronizzato da un clock di scansione a 10KHz fornito dall'esterno.

Specifiche dispositivi:

KBDCNT deve essere realizzato mediante un sequenziatore LLC realizzato in logica cablata (non microprogrammato).

Si richiedono:

1. lo schema a blocchi di KBDCNT con l'indicazione dei necessari segnali di interfaccia verso il PD32;
2. l'illustrazione della tecnica più semplice da adottare per implementare la funzione di trasformazione della coppia riga/colonna in codice ASCII;
3. il diagramma dettagliato di timing che illustri il processo di scansione e l'andamento dei segnali di controllo gestiti da KBDCNT;
4. lo schema elettrico completo di KBDCNT.

RETI LOGICHE

PRIMA PROVA SCRITTA DEL 17-09-1998

STUDENTE: _____

DOCENTE: _____

Progetto:

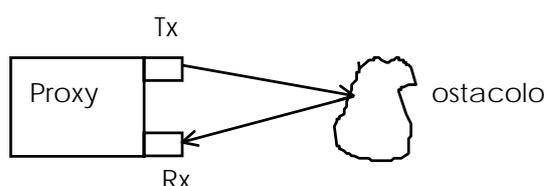
PROXY, misuratore di prossimità

Specifiche funzionali:

PROXY è un dispositivo di misura della distanza di un ostacolo che deve funzionare come segue:

1. PROXY genera sulla linea PULSE un impulso della durata di 50 μ S che attiva un trasmettitore ad ultrasuoni;
2. un rivelatore di ricezione rigenera il segnale d'eco ricevuto dall'ostacolo e produce un impulso verso PROXY sulla linea ECHO;
3. entro un tempo massimo di 10ms PROXY attende l'arrivo dell'impulso di ECHO misurandone il ritardo temporale rispetto all'emissione dell'impulso trasmesso: il ritardo, direttamente proporzionale alla distanza con l'ostacolo, deve essere misurato in modo da ottenere una risoluzione spaziale di 3 cm [si assuma una velocità di propagazione delle onde ultrasoniche di 300m/sec];
4. prima di generare un nuovo ciclo di misura PROXY esegue una pausa di 10ms per attendere lo smorzamento di echi multipli;
5. il flusso di 50 misure/sec così ottenuto deve essere filtrato con una media aritmetica su blocchi di 16 campioni in modo da ottenere 50/16 misure al secondo;
6. le misure filtrate devono essere convertite in 3 digit BCD che esprimono la distanza in centimetri e visualizzate mediante display decimali.

PROXY deve essere implementato in logica cablata (non microprogrammata).



Specifiche dispositivi:

Si assuma che i display decimali contengano internamente la logica di decodifica da codice BCD a codice di accensione dei segmenti del display.

Si richiedono:

1. la frequenza di misura del ritardo d'eco da utilizzare all'interno di PROXY per ottenere la risoluzione spaziale richiesta, il range di distanza misurabile e la relativa risoluzione in bit;
2. lo schema a blocchi di PROXY ed il diagramma dettagliato di timing che ne specifichi il funzionamento;
3. lo schema elettrico completo di PROXY.

RETI LOGICHE

PRIMA PROVA SCRITTA DEL 17-09-1998

Progetto:

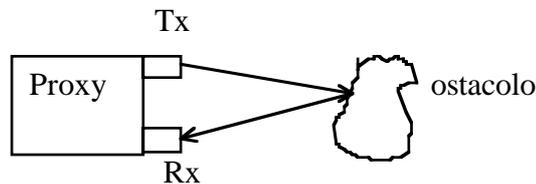
PROXY, misuratore di prossimità

Specifiche funzionali:

PROXY è un dispositivo di misura della distanza di un ostacolo che deve funzionare come segue:

1. PROXY genera sulla linea PULSE un impulso della durata di 50 μ S che attiva un trasmettitore ad ultrasuoni;
2. un rivelatore di ricezione rigenera il segnale d'eco ricevuto dall'ostacolo e produce un impulso verso PROXY sulla linea ECHO;
3. entro un tempo massimo di 10ms PROXY attende l'arrivo dell'impulso di ECHO misurandone il ritardo temporale rispetto all'emissione dell'impulso trasmesso: il ritardo, direttamente proporzionale al doppio della distanza con l'ostacolo, deve essere misurato in modo da ottenere una risoluzione spaziale della distanza dell'ostacolo di 1,5 cm [si assuma una velocità di propagazione delle onde ultrasoniche di 300m/sec];
4. prima di generare un nuovo ciclo di misura PROXY esegue una pausa di 10ms per attendere lo smorzamento di echi multipli;
5. il flusso di 50 misure/sec così ottenuto deve essere filtrato con una media aritmetica su blocchi di 16 campioni in modo da ottenere 50/16 misure al secondo;
6. le misure filtrate devono essere convertite in 4 digit BCD che esprimono la distanza in centimetri e visualizzate mediante display decimali.

PROXY deve essere implementato in logica cablata (non microprogrammata).



Specifiche dispositivi:

Si assuma che i display decimali contengano internamente la logica di decodifica da codice BCD a codice di accensione dei segmenti del display.

Si richiedono:

1. la frequenza di misura del ritardo d'eco da utilizzare all'interno di PROXY per ottenere la risoluzione spaziale richiesta, il range di distanza misurabile e la relativa risoluzione in bit;
2. lo schema a blocchi di PROXY ed il diagramma dettagliato di timing che ne specifichi il funzionamento;
3. lo schema elettrico completo di PROXY.

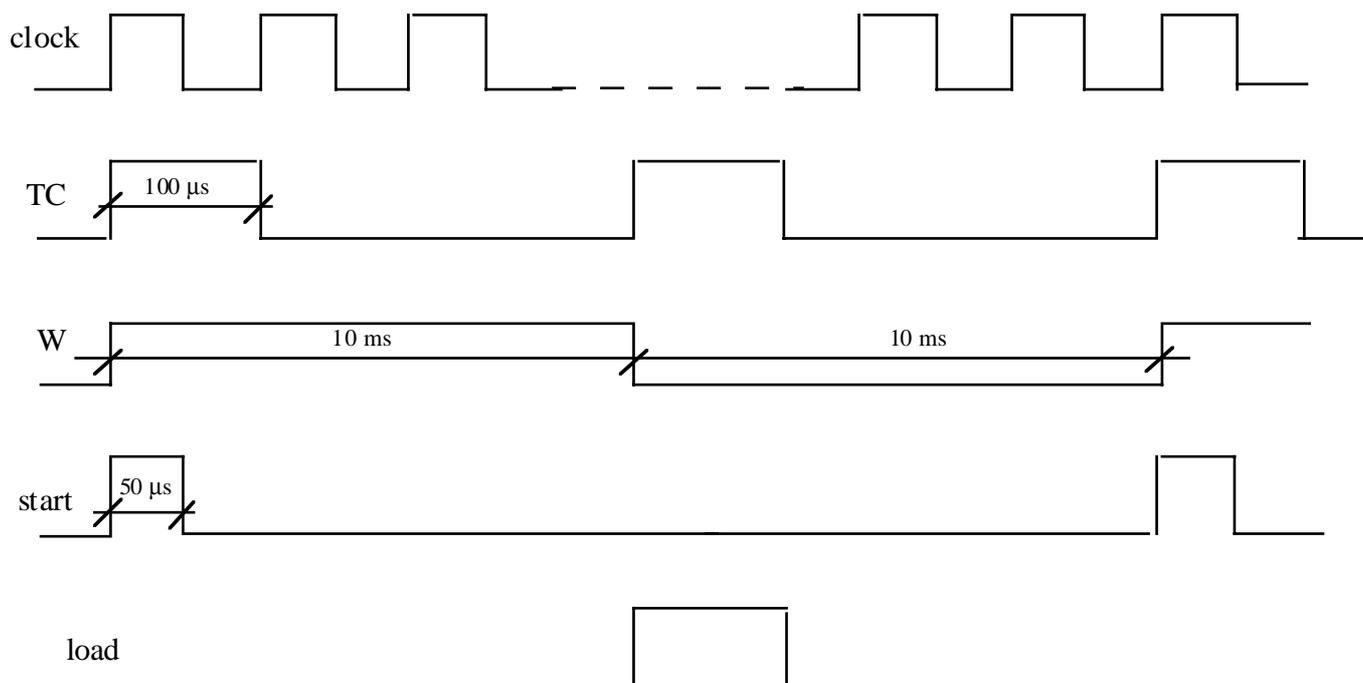
Una possibile soluzione

1. **Frequenza di campionamento:** dalle specifiche si evince che occorre una risoluzione di 3 cm sulla distanza doppia. Poiché si suppone la velocità di propagazione dell'impulso di 300 m/sec, il conteggio di un impulso comporta un periodo di $3 \cdot 10^{-2} / 3 \cdot 10^2 = 10^{-4}$ sec; pertanto la frequenza di campionamento dovrà essere di 10 KHz.
2. **Conteggio massimo da misurare:** si richiede di bloccare la misura a 10 msec, corrispondente ad una distanza dell'ostacolo di 150 cm, pertanto il contatore di misura dovrà avere una capacità di almeno

10KHz·10msec=100. Si utilizzerà un contatore a 7 bit provvisto di ingresso clock, ingresso CE (count enable) e ingresso CL (azzeramento).

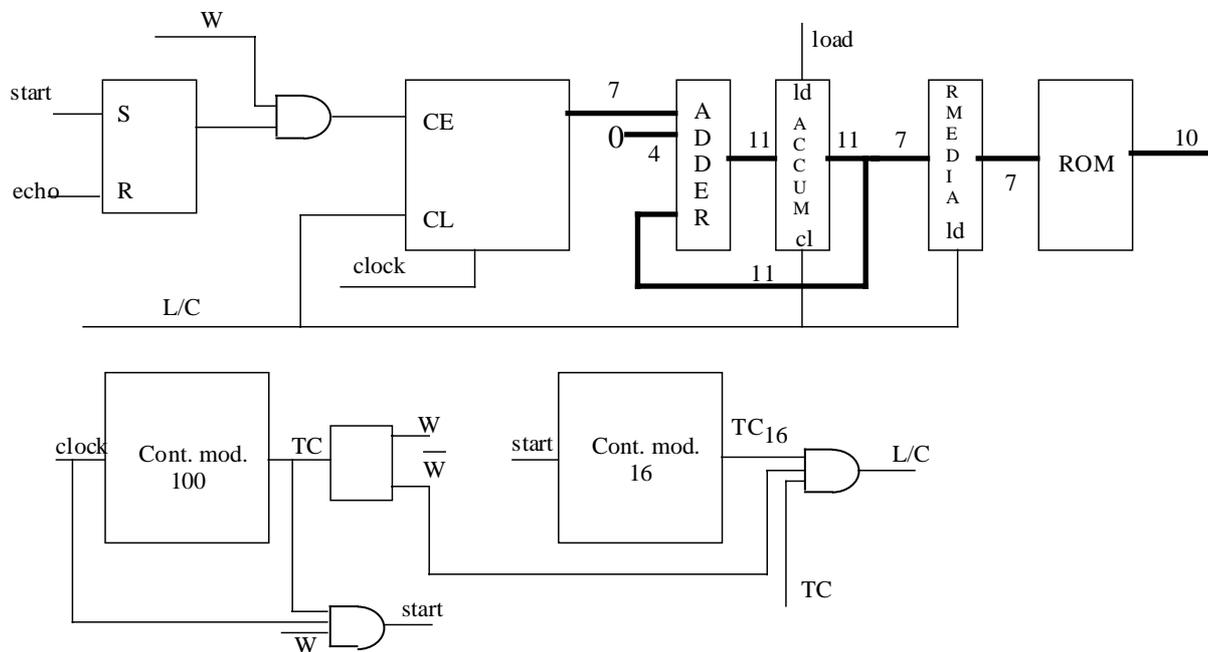
3. **Abilitazione al conteggio:** il contatore di misura, alimentato dal clock a 10 KHz dovrà essere abilitato per un periodo pari all'intervallo tra Start Pulse ed Echo Pulse, se quest'intervallo è inferiore a 10 msec
4. **Generazione dello start pulse:** questo impulso deve avere una durata di 50 microsecondi. Il periodo del clock è di 100 microsecondi. Occorre quindi un clock ad onda quadra e prelevare un impulso di clock ogni 20 millisecondi.
5. **Media sui sedici campioni:** il contenuto del contatore di misura andrà accumulato in un registro accumulatore sommando sedici campioni. Poiché ogni campione può essere di 7 bit occorrerà un addizionatore ed un registro di accumulo a 11 bit. La media si otterrà trasferendo in un registro di media i 7 bit più significativi.
6. **Conversione da media a distanza espressa in BCD:** la soluzione più semplice è quella di utilizzare una ROM pilotata dalla media e facendo corrispondere ad ogni valore della media una parola che rappresenti in BCD la distanza dell'ostacolo. La distanza massima da rappresentare è pari a 150 cm, la minima è invece pari a 1,5 cm. Il passo di incremento è di 1,5 cm per cui la cifra decimale vale 0 o 5 e quindi è necessario un solo bit per individuarla. La cifra delle centinaia è parimenti 0 o 1 e anche in questo caso è necessario un solo bit. In conclusione il numero di bit della parola di ROM dovrà essere $1+4+4+1=10$ ed il numero di parole è 128.

In base alle considerazioni precedenti si possono definire i segnali di temporizzazione necessari per il funzionamento del misuratore. Nella figura seguente sono riportate le forme d'onda che occorrono.



1. Il clock a 10 KHz pilota un contatore mod100 e TC è il terminal count relativo.
2. W è una finestra da 10 msec ottenuta tramite divisione per due del segnale TC
3. Start è ottenuto dall'AND di TC, clock e W
4. Il segnale load ottenuto dall'AND di W' e TC serve a memorizzare i campioni nel registro accumulatore.
5. Non riportato in figura è necessario un segnale analogo a load che chiameremo L/C che avrà cadenza 1/16 di load e che servirà a caricare sul fronte positivo il registro di media e sul livello ad azzerare contatore ed accumulatore.

In base alle forme d'onda definite, lo schema elettrico del misuratore è quello della figura seguente.



RETI LOGICHE
seconda prova del 17/09/1998

Candidato _____ Docente _____

- D1 Un codice di Hamming a distanza tre è formato da k bit di controllo e n bit dati: ricavare la relazione tra k ed n .
- D2 Realizzare un circuito combinatorio per la trasformazione di un numero decimale di tre cifre, codificato BCD, in binario puro utilizzando in modo ottimizzato la tecnica della moltiplicazione parallela (si faccia attenzione al fatto che i moltiplicatori sono delle costanti) .
- D3. Determinare la frequenza massima di funzionamento di un circuito asincrono autosincronizzante supponendo che le porte logiche elementari abbiano un tempo di propagazione minimo e massimo rispettivamente pari a 5 nsec e 10 nsec, e che i flip/flop usati abbiano un tempo di commutazione minimo e massimo pari a 10 nsec e 20 nsec.
- D4 Descrivere la struttura di una SCA di un processore con 8 registri interni in grado di eseguire operazioni a due operandi in un periodo di clock.
- D5 Progettare una memoria di lavoro di 2^{24} byte, connessa al data bus di un PD32, costituita da moduli di 2^{20} byte in modo da poter leggere/scrivere un byte, una parola, una doppia parola alla volta. Definire lo schema elettrico con tutti I segnali necessari.

RETI LOGICHE

PRIMA PROVA SCRITTA DEL 15-10-1998

STUDENTE: _____

DOCENTE: _____

Progetto: imp-dmac

Specifiche funzionali

Una periferica produce blocchi di dati di dimensione non superiori a 4k byte ad un data rate di 400 kB/s. I blocchi vengono prodotti in modo random: la periferica alza un flag quando un dato è disponibile sul suo registro di uscita.

Occorre realizzare una interfaccia verso un PD32 per trasferire i dati con tecnica DMA. Per ridurre il tempo di sospensione del processore e semplificare l'accesso in memoria del PD32 occorre prevedere un impacchettatore di byte in modo da richiedere i bus del PD32 quando è pronta una long word.

L'indirizzo di partenza della memoria in cui scrivere i dati è fisso e pari a 80000000H.

Il processore lavora con un clock di periodo 100 nsec ed il ciclo di memoria è di 3 periodi clock.

Si richiedono:

1. lo schema a blocchi di impdmac con l'indicazione dei necessari segnali di interfaccia verso il PD32 e la periferica;
2. lo schema elettrico di impdmac
3. il calcolo percentuale del rallentamento del PD32 durante la fase di trasferimento di un blocco dati.

RETI LOGICHE

SECONDA PROVA SCRITTA DELL'APPELLO 15-10-98

Studente: _____

Docente: _____

- D1. Descrivere le regole di rappresentazione dei numeri in virgola mobile secondo la normativa IEEE.
- D2. Mostrare che una rappresentazione minima di funzione di n variabili non può avere più di 2^{n-1} termini.
- D3. Descrivere il modello strutturale di principio delle reti impulsive con più variabili di tipo clock e le caratteristiche della memoria di stato.
- D4. Illustrare il protocollo di sincronizzazione per il trasferimento dati tra due unità microprogrammate. Si richiede lo schema logico d'interfaccia tra le due unità e i microprogrammi di gestione del protocollo.
- D5. Analizzare le due possibilità di scambio dati tra processore e periferica di tipo programmato: interruzione per ogni singolo dato o interruzione per inizio blocco. Definire il data-rate al di sopra del quale è più efficiente l'interruzione per inizio blocco, supponendo che il tempo impiegato dal processore per entrare nella (uscire dalla) routine di interruzione è di 10 microsec.

RETI LOGICHE

PRIMA PROVA SCRITTA DEL 14-01-1999

STUDENTE: _____

DOCENTE: _____

Progetto: BIFC, trasmettitore a burst di pacchetti dati

Specifiche funzionali:

BIFC è una periferica PD32 dedicata alla gestione dell'interfaccia con un modulatore a burst BMOD per trasmissioni seriali sincrone via satellite a 2Mbps; l'interfaccia sincrona con il modulatore è costituita dai segnali SD (Send_Data, BIFC→BMOD), ST (Send_Timing, BIFC←BMOD), RS (Request_to_Send, BIFC→BMOD), CS (Clear_to_Send, BIFC←BMOD); tutti i segnali di interfaccia sono sincronizzati sui fronti positivi del clock di riferimento ST. Il pacchetto dati da trasmettere, organizzato a word (16 bit), è allocato in memoria PD32 a partire dall'indirizzo fisso 80000000H; la prima word contiene la lunghezza Len (in word) del pacchetto. BIFC deve funzionare secondo i seguenti passi:

1. Nello stato di riposo, ogni 128 microS BIFC preleva in DMA la word Len; se Len=0 BIFC rilascia il bus e torna a riposo, altrimenti va al passo 2;
2. BIFC si predispone alla trasmissione attivando la linea RS e mettendosi in attesa del consenso alla trasmissione inviato da BMOD, il quale attiverà la linea CS dopo un certo numero di cicli di clock ST;
3. Una volta rilevata l'attivazione del segnale CS, a partire dal ciclo di clock ST successivo BIFC inizia la trasmissione a 2Mbps del pacchetto prelevando in DMA le relative word con accesso in bus stealing e serializzandole in sincronismo con il clock ST;
4. Durante la trasmissione dell'ultimo bit del pacchetto BIFC disattiva la linea RS per segnalare la fine del pacchetto dati ed attende che BMOD disattivi a sua volta la linea CS;
5. BIFC azzerla la word Len in memoria e torna al passo 1.

Specifiche dispositivi:

Il PD32 lavora con un clock a 10MHz e cicli di R/W della memoria della durata di 300ns (3 stati macchina).

Si richiedono:

1. il diagramma dettagliato di timing che specifichi il funzionamento dell'interfaccia sincrona fra BIFC e BMOD durante le fasi di avvio, esecuzione e terminazione della trasmissione a burst;
 2. lo schema a blocchi di BIFC ed il diagramma dettagliato di timing che ne specifichi il funzionamento;
 3. lo schema elettrico completo di BIFC ed il diagramma a stati od il microprogramma del SCO di BIFC;
 4. il calcolo del rallentamento percentuale del PD32 durante la trasmissione di un pacchetto.
-

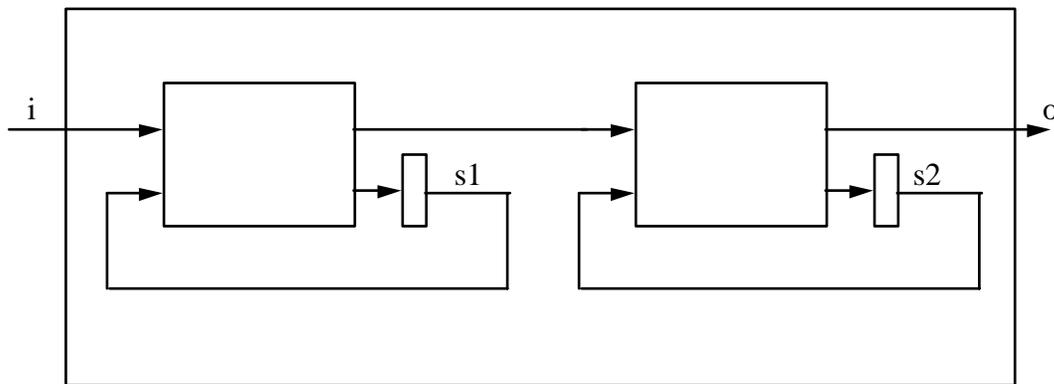
RETI LOGICHE

SECONDA PROVA SCRITTA DELL'APPELLO DEL 14-01-99

Studente: _____ Docente: _____

D1: Una struttura di calcolo è costituita da tre blocchi combinatori in cascata con tempi di calcolo pari a 20, 40, 20 nsec. Trasformarla in una struttura pipeline in modo da avere periodo di clock di 25 nsec; si hanno a disposizione registri con tempo di set-up di 2 nsec e tempo di commutazione di 3 nsec. Tracciare la temporizzazione relativa ai vari stadi.

D2 Si consideri la struttura in figura:



costituita da due macchine collegate in cascata: definire la macchina complessiva, in termini di I, S, O, δ, ω , in funzione dei parametri $I_1, S_1, O_1, \delta_1, \omega_1$ e $I_2, S_2, O_2, \delta_2, \omega_2$ delle due macchine componenti.

D3 Descrivere la procedura per trasformare una macchina sincrona impulsiva in una asincrona con ingressi impulsivi

D4 Descrivere la sequenza di microoperazioni della ipotetica istruzione macchina PD32:

MAXs D(R₁), R₂

La semantica è: metti in R₂ l'elemento massimo di un vettore di D dati di dimensione s (b, w, l) puntato da R₁. Il parametro D di dimensione longword è memorizzato in modo analogo allo spiazzamento.

D5 Sintetizzare il diagramma degli stati di una macchina che riconosca sequenze del tipo: 0(11)ⁿ0 non sovrapposte.

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 04-02-1999

STUDENTE: _____

DOCENTE: _____

Progetto: SHMARB, arbitro per memoria condivisa.

Specifiche funzionali:

SHMARB è un arbitro che disciplina su base ciclo macchina di memory R/W l'accesso alla memoria condivisa SHMEM di un sistema multiprocessore basato su 4 PD32 indipendenti e con clock fra loro scorrelati in frequenza e fase; SHMEM, di capacità pari a 16MB, è mappata nello spazio di indirizzamento dei PD32 a partire da F000000H. SHMARB, operante con un clock a frequenza molto maggiore di quelle dei PD32, deve concedere l'accesso alla SHMEM servendo i PD32 in base alla seguente disciplina

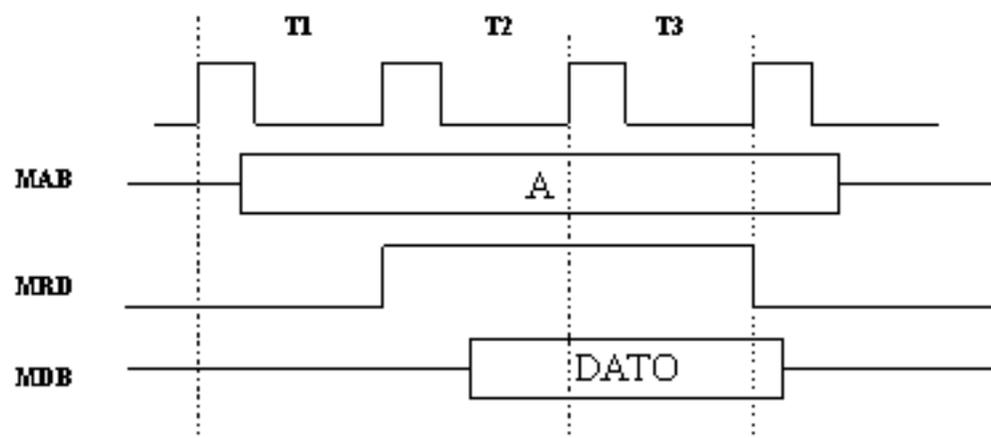
1. SHMARB contiene un contatore a 4 bit che avanza in sincronismo con il clock interno e genera un marker temporale di riferimento;
2. L'inizio di un ciclo di accesso alla SHMEM da parte del generico PD32 provoca la generazione di un segnale di richiesta verso la logica di controllo di SHMARB (*possono iniziare fino a 4 cicli di accesso concorrenti, non esistendo alcuna sincronizzazione fra i processori; il ciclo macchina del generico PD32 deve essere esteso fino a quando l'arbitro non conceda l'accesso alla SHMEM*) e la memorizzazione del marker temporale relativo all'istante di arrivo della richiesta;
3. SHMARB, non appena la SHMEM risulti libera, ottiene da un encoder combinatorio (da non progettare) l'indice del PD32 che deve essere servito, provvede ad abilitarne l'accesso alla SHMEM e riporta la relativa logica di richiesta in stato disattivo; l'encoder combinatorio ha come ingressi i 4 flag di richiesta ed i relativi marker temporali di arrivo delle richieste.

L'accesso alla SHMEM deve essere assicurato in tutte le modalità standard PD32 (byte, word, longword).

Si richiedono:

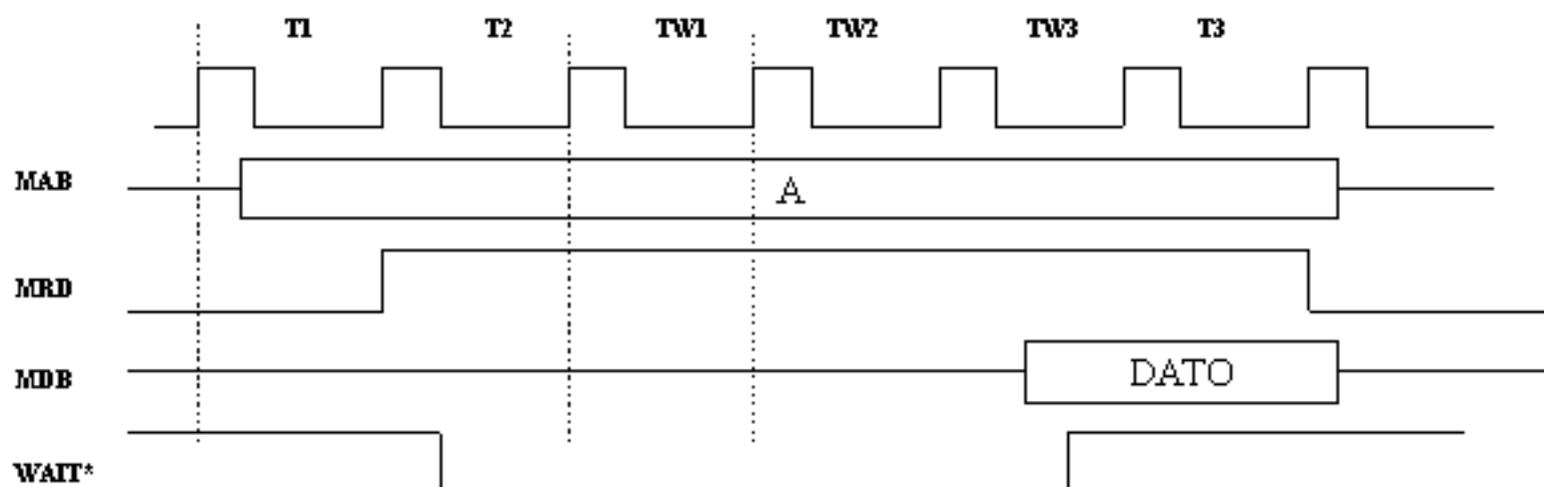
1. il diagramma dettagliato di timing che specifichi il funzionamento dell'arbitraggio fra cicli di accesso concorrenti alla SHMEM;
2. lo schema a blocchi di SHMARB con il dettaglio dei segnali di interfacciamento con i PD32 e SHMEM necessari;
3. lo schema elettrico completo di SHMARB ed il diagramma a stati od il microprogramma del controllore di arbitraggio;

SHMARB - Ciclo di Read PD32

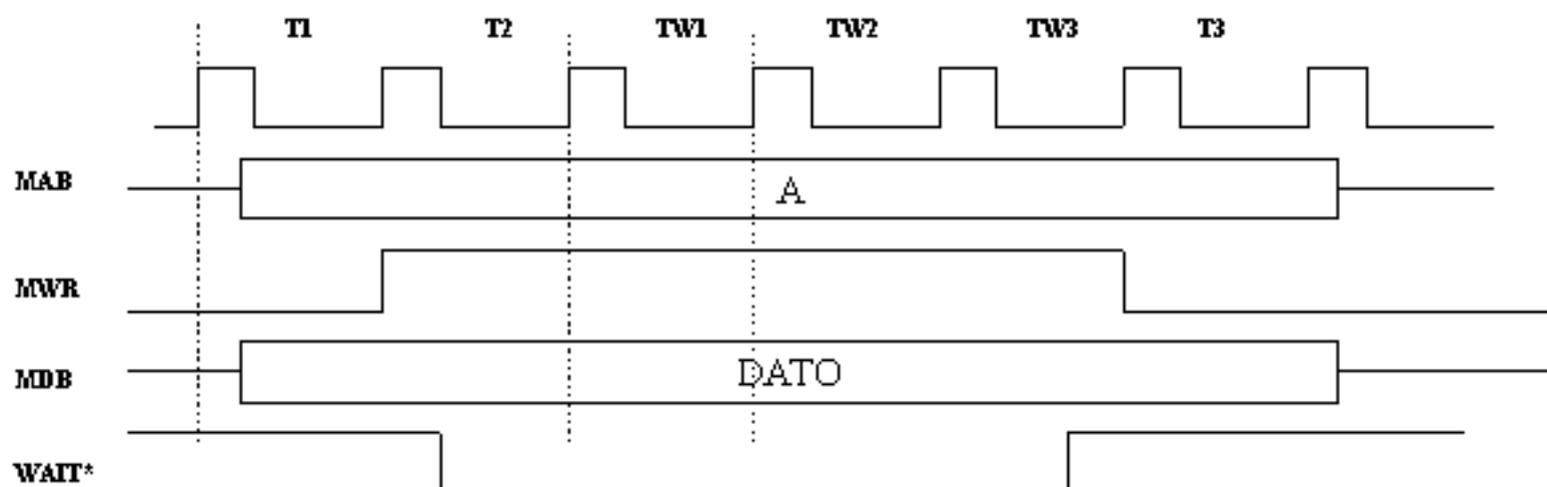
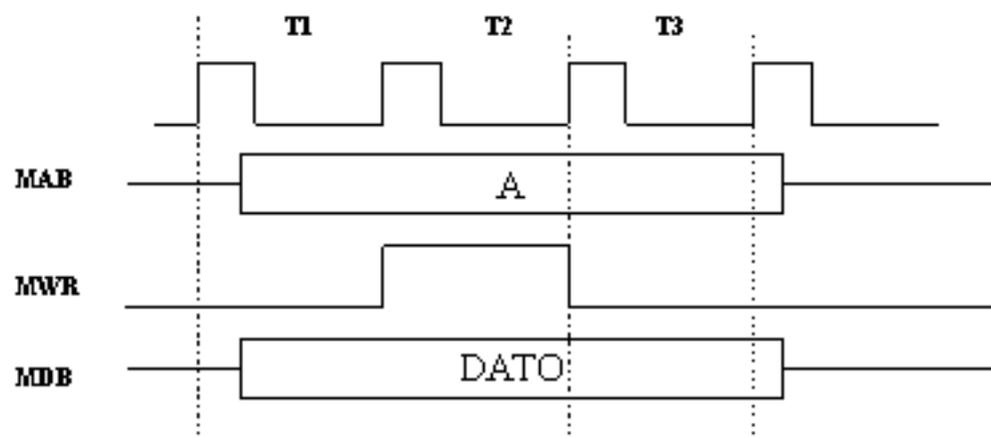


Utilizzo del segnale di WAIT*:

- Memorie o periferiche lente
- Memoria condivisa
- Controllore DMA



SHMARB - Ciclo di Write PD32



SECRET

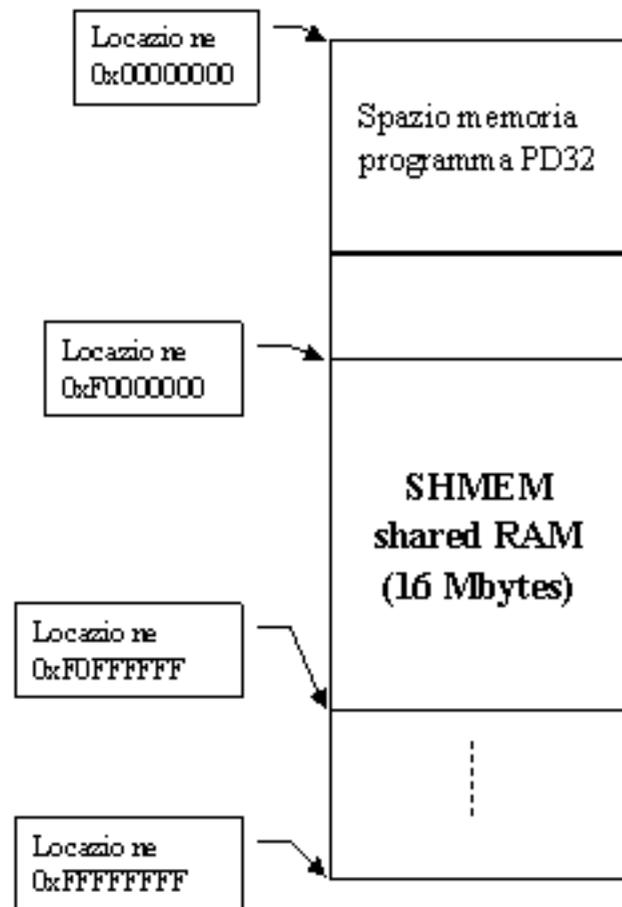
CLASSIFICATION	CONTROL	PROTECTION	REMARKS
SECRET	SECRET	SECRET	

CLASSIFIED BY: [redacted]
 AUTHORITY: [redacted]
 DATE: [redacted]

SECRET

SHMEM Memory Mapping

Spazio di indirizzamento PD32 (byte)



Affinche' il PD32 acceda in lettura la RAM interna SHMARB deve "tradurre" gli indirizzi del MAB PD32 in indirizzi validi per la SHMEM

MAB(31:2) => SHMEMA(21:0)

0x3C000000 0x000000

0x3C000001 0x000001

.....

.....

0x3C3FFFFFF 0x3FFFFFF

MAB(31:24) si usa per la decodifica

MAB(23:2) si usa per la generazione SHMEMA

inoltre per garantire tutti i tipi di accesso

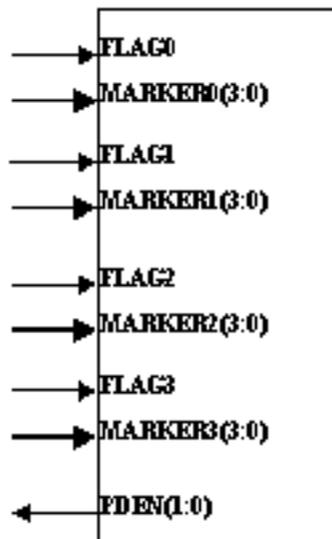
Mb0 => SHMEMCE0

Mb1 => SHMEMCE1

Mb2 => SHMEMCE2

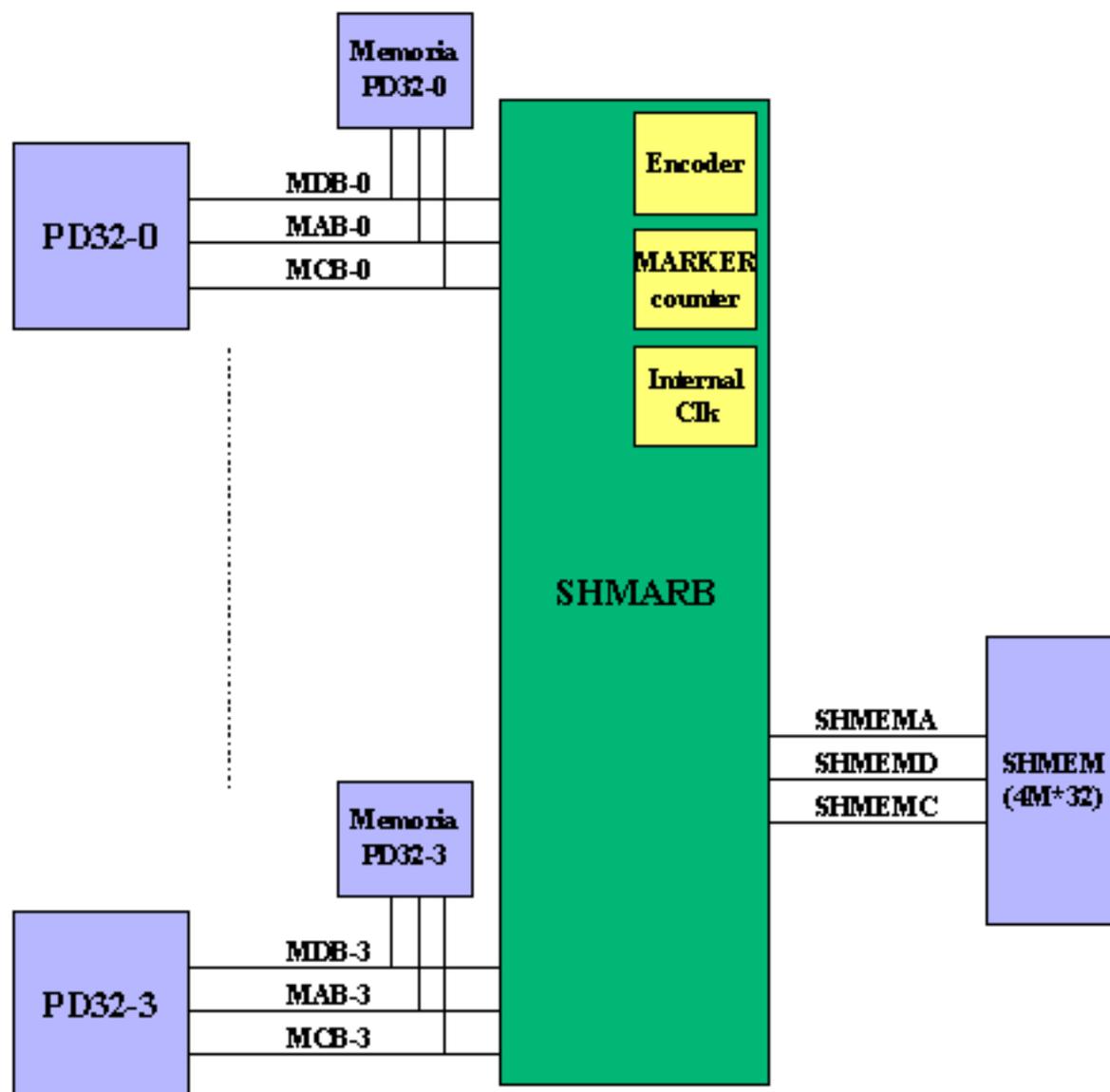
Mb3 => SHMEMCE3

SHMARB - Encoder



PDEN = i abilita l'accesso del PD32 i alla SHMEM

Architettura progetto SHMARB



Blocchi funzionali unita' SHMARB

- **SHMEM INTERFACE**

- gestione segnali SHMEM
 - SHMEMA
 - SHMEMD
 - SHMEMCE_i,SHMEMWE,SHMEMRD

- **PD32 DECODING**

- decodifica "memory mapping" MAB(31:24),MWR,MRD
- generazione flag richiesta accesso inizio (MEMREQ-i)

- **SCO**

- gestione timing flag (FLAG-i)
- generazione marker temporali
- gestione linee di WAIT*-i
- gestione abilitazione accesso SHMEM

Protocollo SHMARB

PD32

1) Inizio ciclo di accesso di uno piu' PD32

⋮

Inizio ciclo di accesso di uno piu' PD32

SHMARB

(al reset la SHMEM e' libera)

2) Memoria libera

3) Generazione flag richiesta di accesso (MEMREQ-i) e MARKER temporale (MARKER-i(3:0))

4) Lettura Encoder ed abilitazione di un PD32 all'accesso con eventuale gestione del WAIT per gli altri.

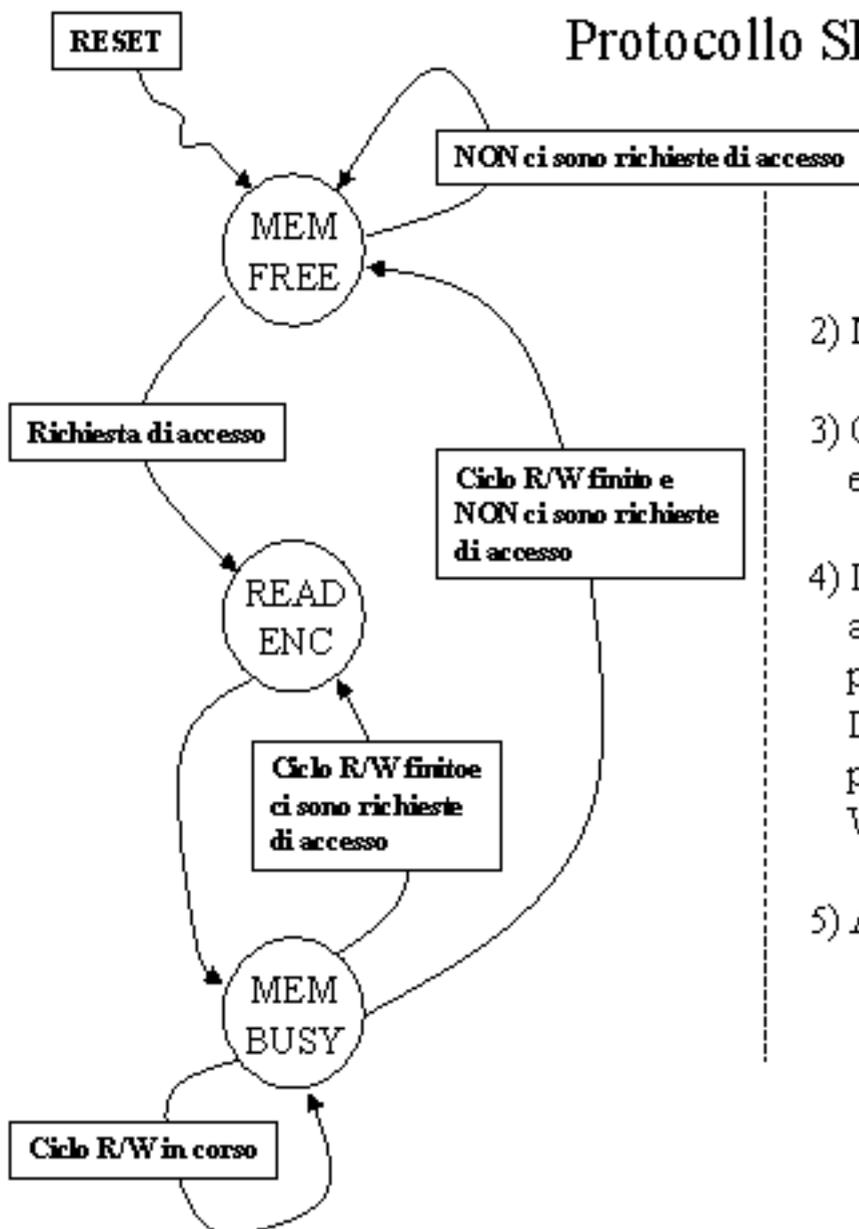
→ Durante l'accesso di un PD32 devo garantire il punto 3 per gli altri PD32 e conseguente generazione WAIT states

5) Alla fine del ciclo di accesso:

- Se non ho richieste attive (MEMREQi) torno al punto 2

- Se ho richieste attive tomo al punto 4

Protocollo SHMARB



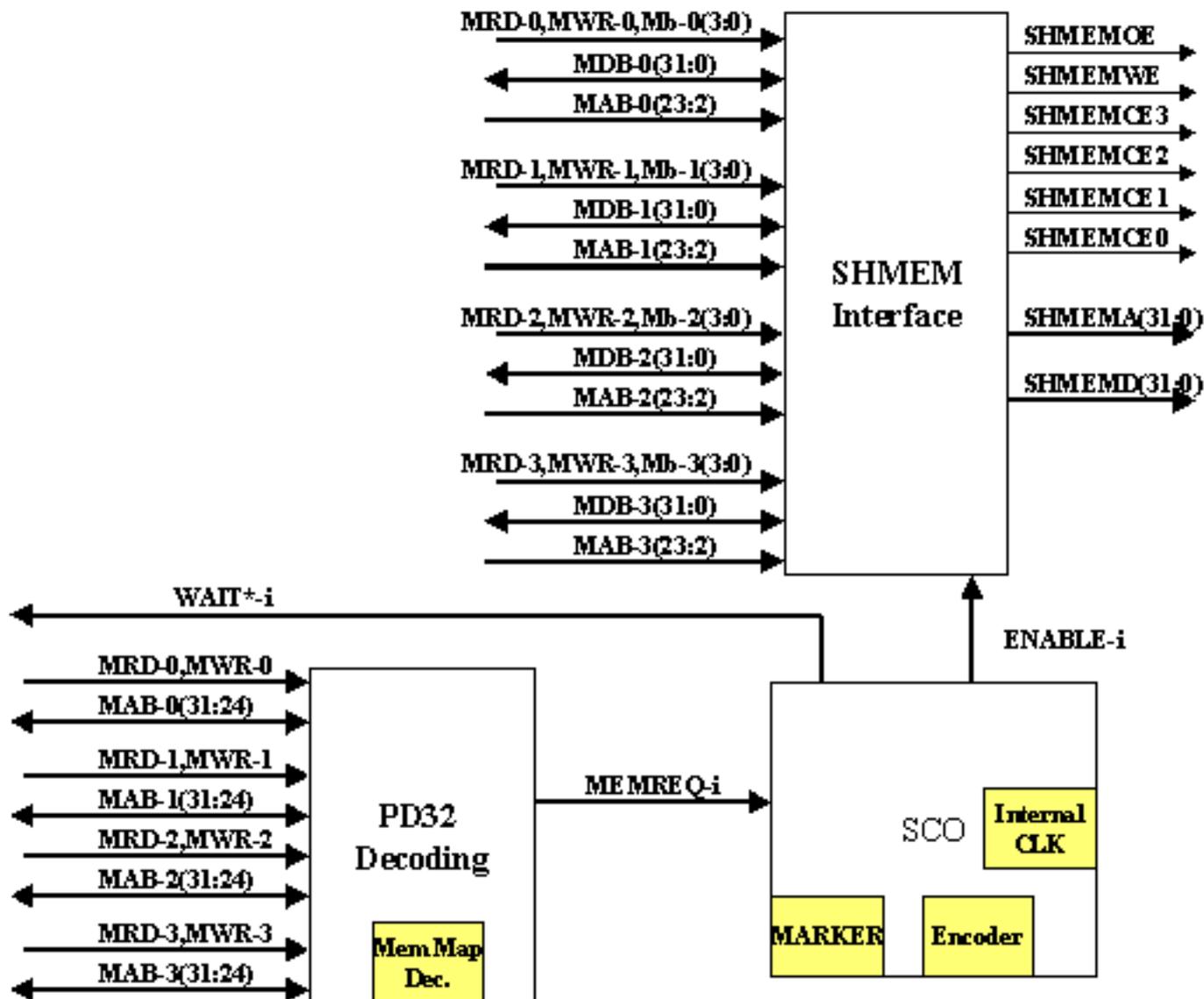
SHMARB

(al reset la SHMEM e' libera)

- 2) Memoria libera
- 3) Generazione flag richiesta di accesso (MEMREQ_i) e MARKER temporale (MARK_i(3:0))
- 4) Lettura Encoder ed abilitazione di un PD32 all'accesso con eventuale gestione del WAIT per gli altri.
Durante l'accesso di un PD32 devo garantire il punto 3 per gli altri PD32 e conseguente generazione WAIT states
- 5) Alla fine del ciclo di accesso:
 - Se non ho richieste attive (MEMREQ_i) torno al punto 2
 - Se ho richieste attive tomo al punto 4

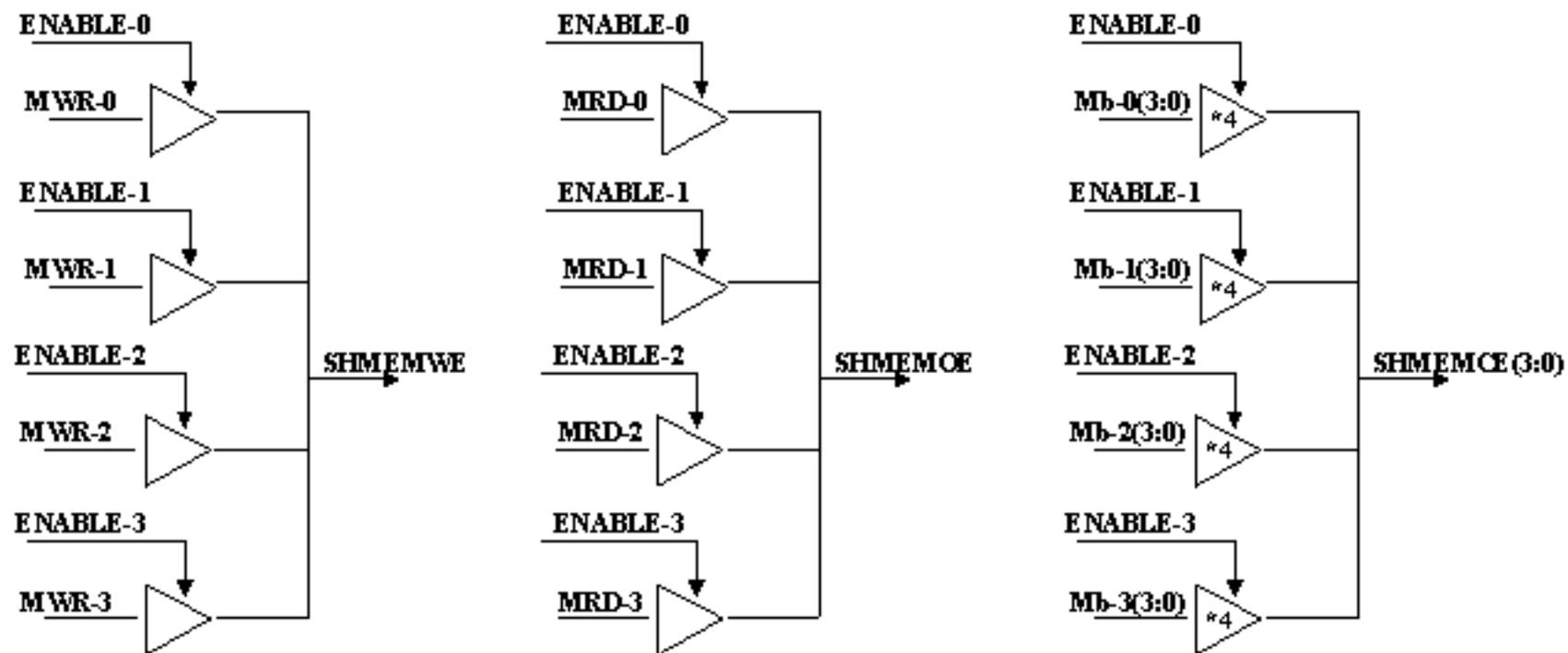
SHMARB Schema a blocchi funzionale

- prima approssimazione -



SHMARB - SHMEM INTERFACE

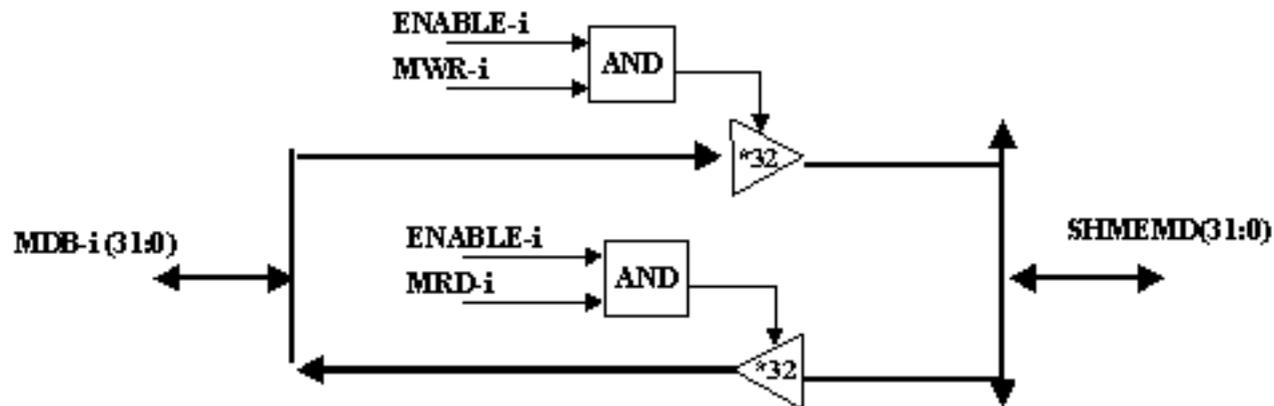
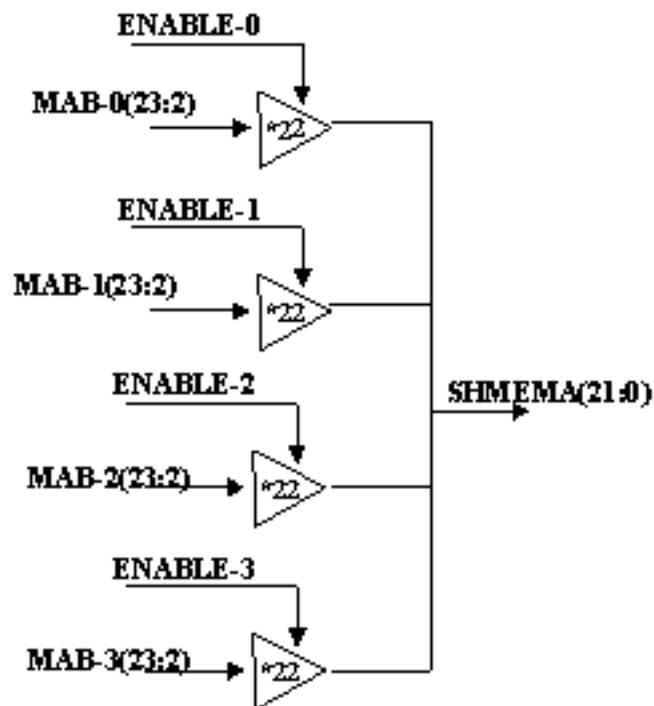
Segnali di controllo



SHMARB - SHMEM INTERFACE

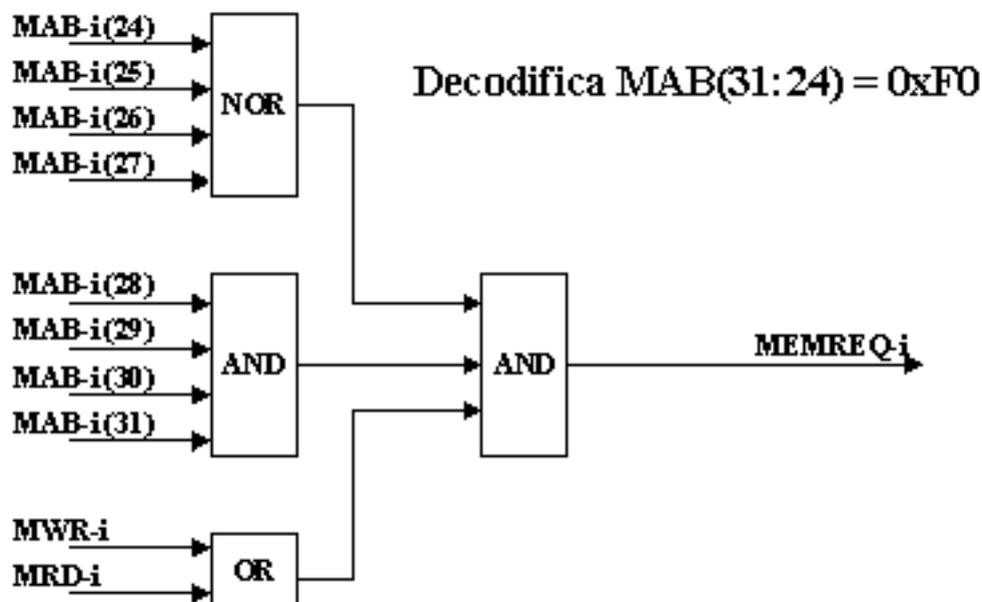
Address e Data Bus

Questa logica va ripetuta 4 volte



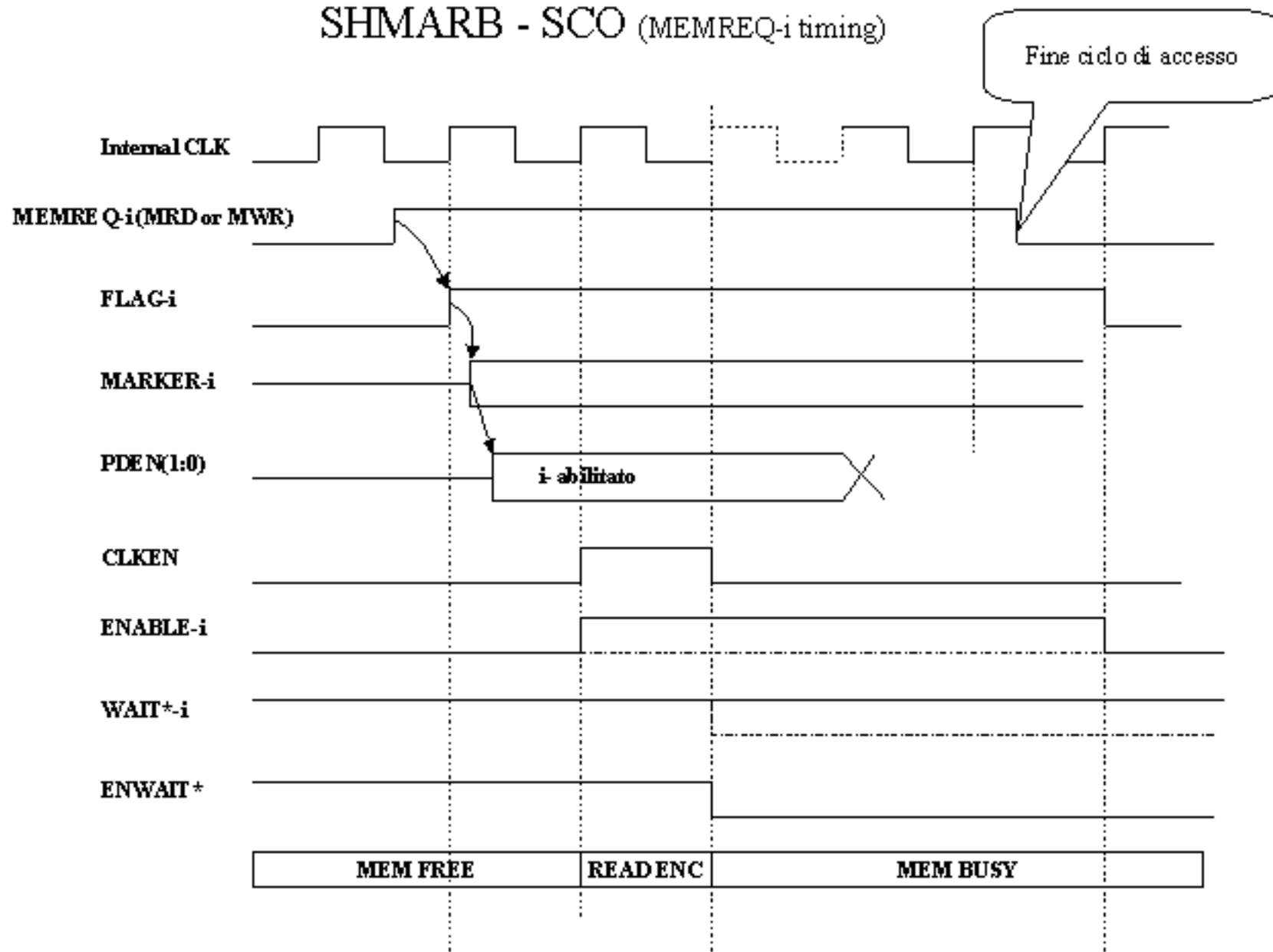
SHMARB - PD32 DECODING

Generazione MEMREQ-i



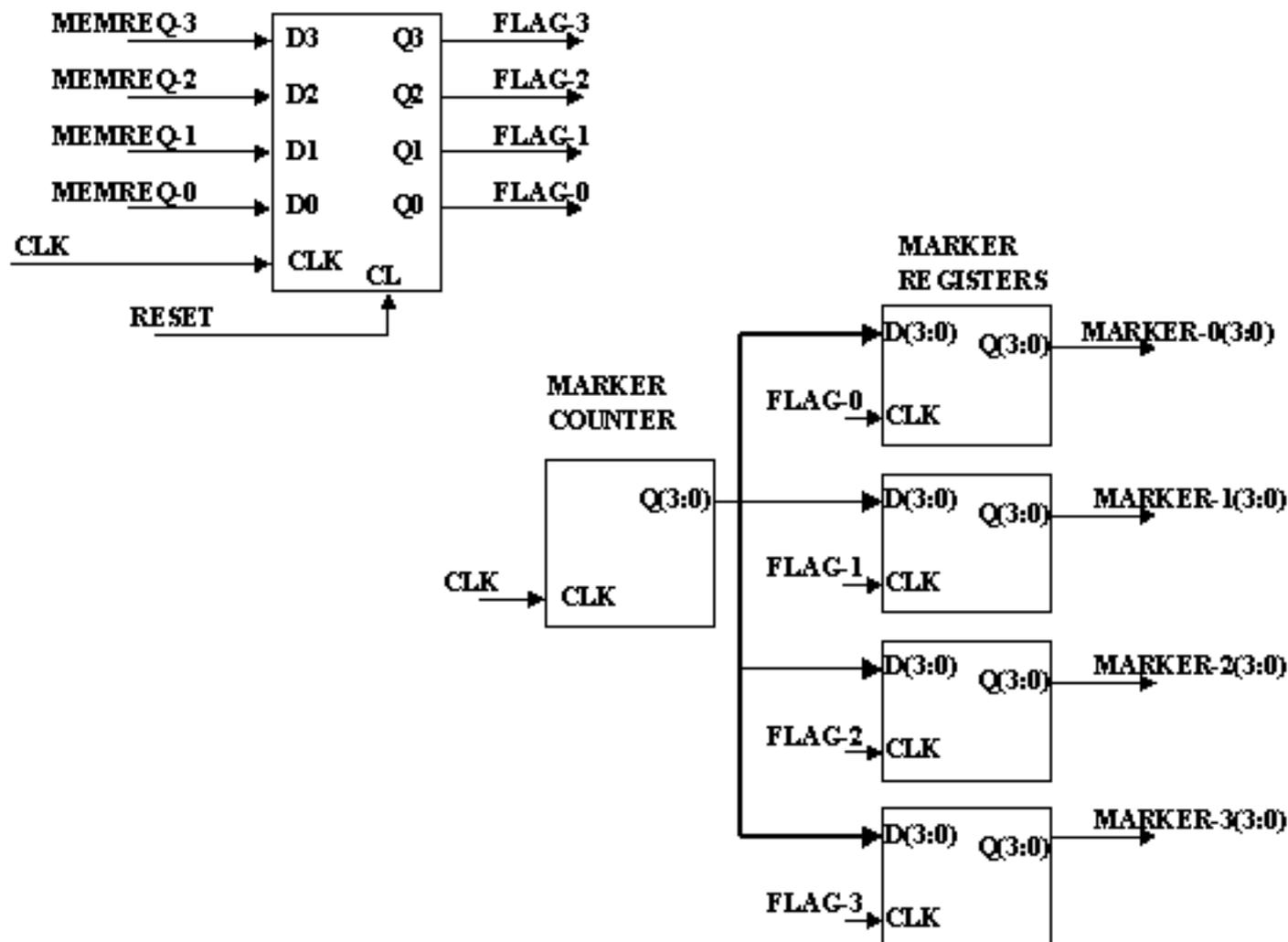
Questa logica e' ripetuta 4 volte per i che va (0..3)

SHMARB - SCO (MEMREQ-i timing)



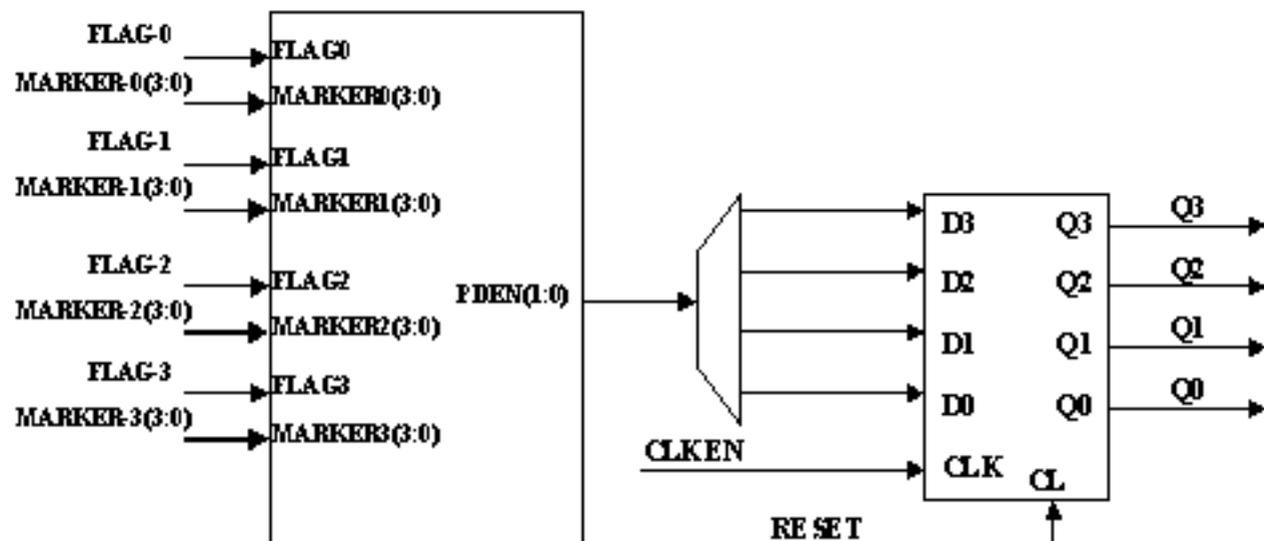
SHMARB - SCO

- gestione timing flag (FLAGi)
- generazione marker temporali

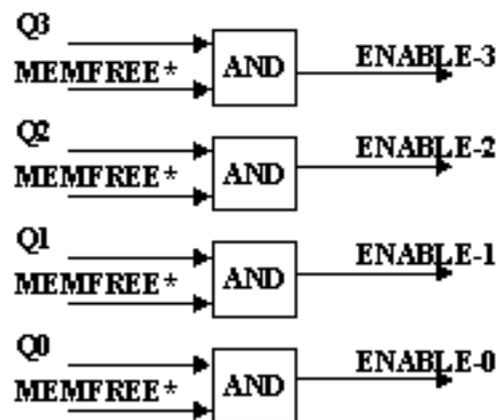


SHMARB - SCO

- gestione abilitazione accesso SHMEM (ENABLE-i)

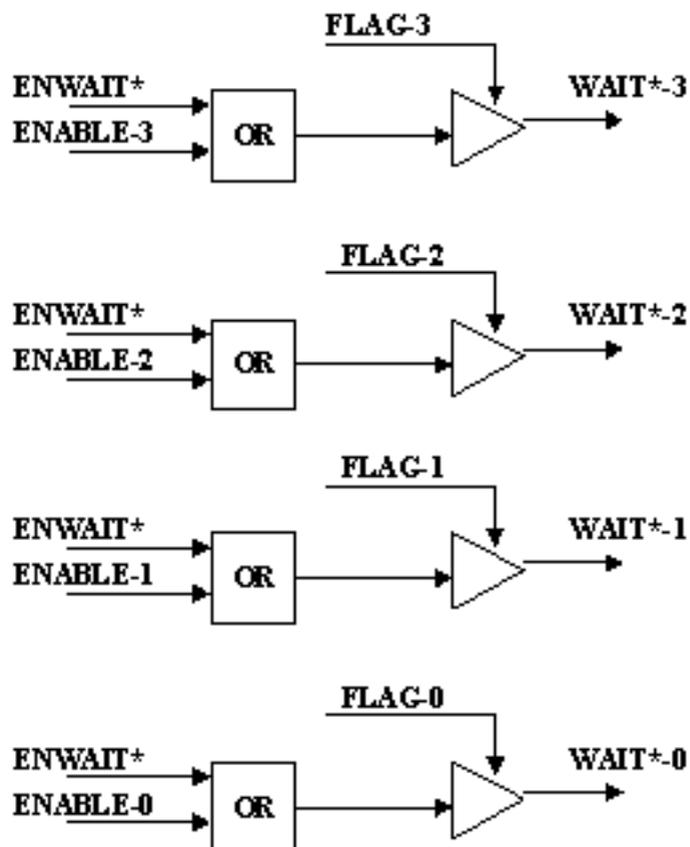


MEMFREE* e' generato dalla macchina a stati quando non ci sono trasferimenti in corso e lo uso per tenere i bus di SHMEM ad un valore inattivo



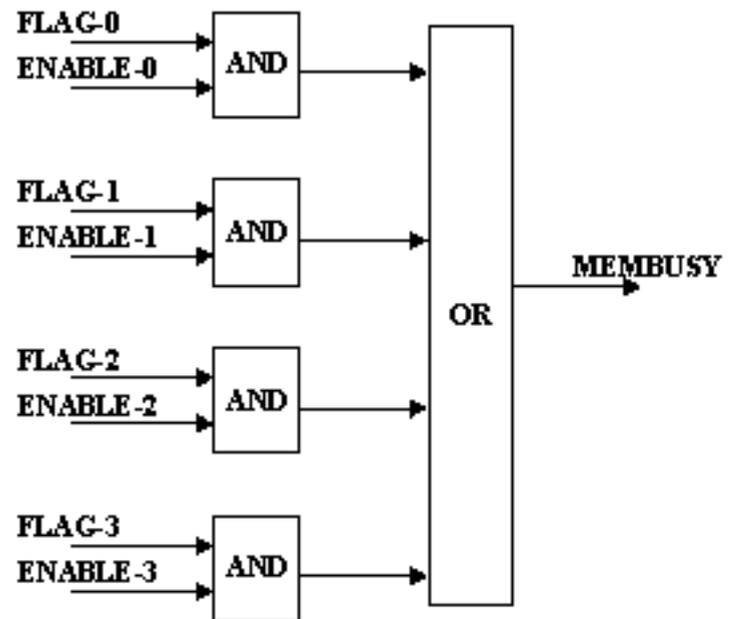
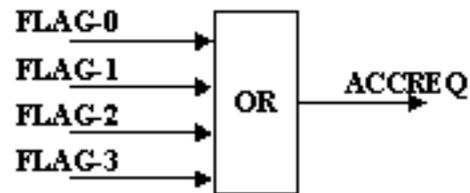
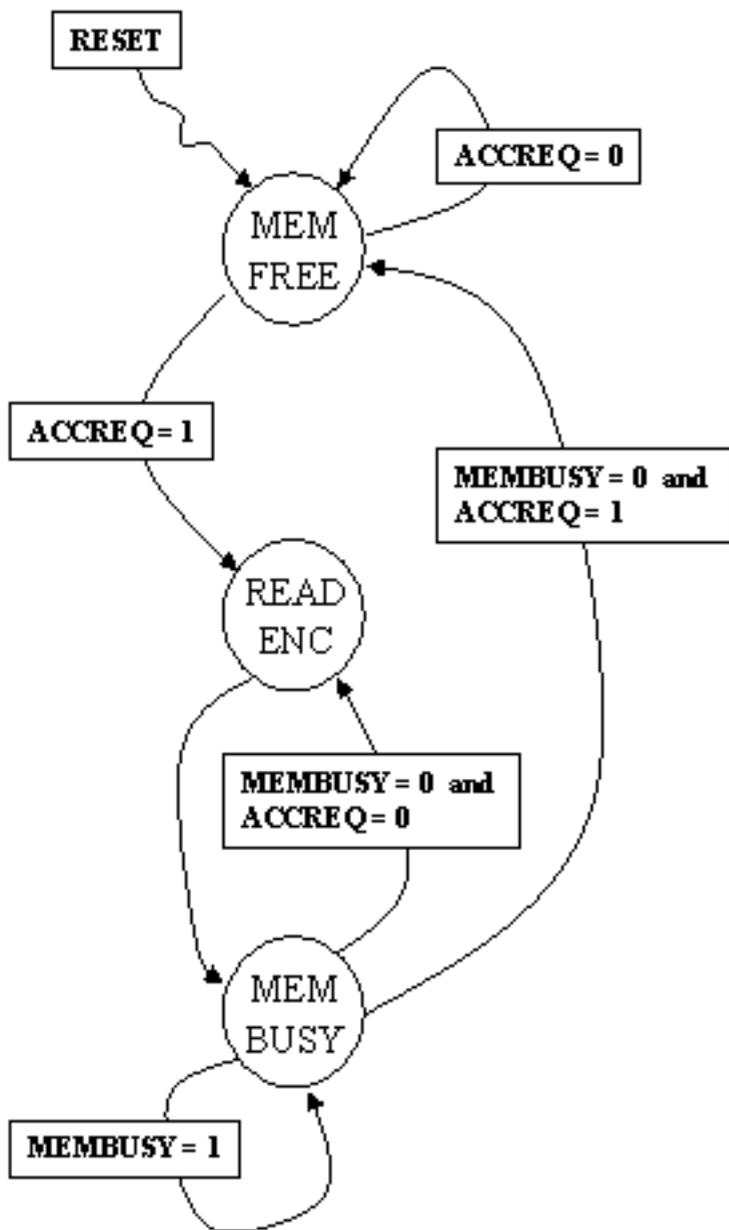
SHMARB - SCO

- gestione WAIT states



La SCO assicura bassa la linea di WAIT-i solo se il PD32-i ha richiesto l'accesso alla SHMEM e se la macchina a stati ne abilita uno diverso da i

SHMARB - SCO



SHMARB - SCO

Generazione segnali di controllo (CLKEN, ENWAIT, MEMFREE)

	CLKEN	ENWAIT*	MEMFREE*	SMREG(1)	SMREG(0)
MEM_FREE	0	1	0	0	0
READ_ENC	1	1	1	0	1
MEM_BUSY	0	0	1	1	0

$CLKEN \Leftarrow SMREG(0).Q$

$MEMFREE^* \Leftarrow SMREG(1).Q \text{ or } SMREG(0).q$

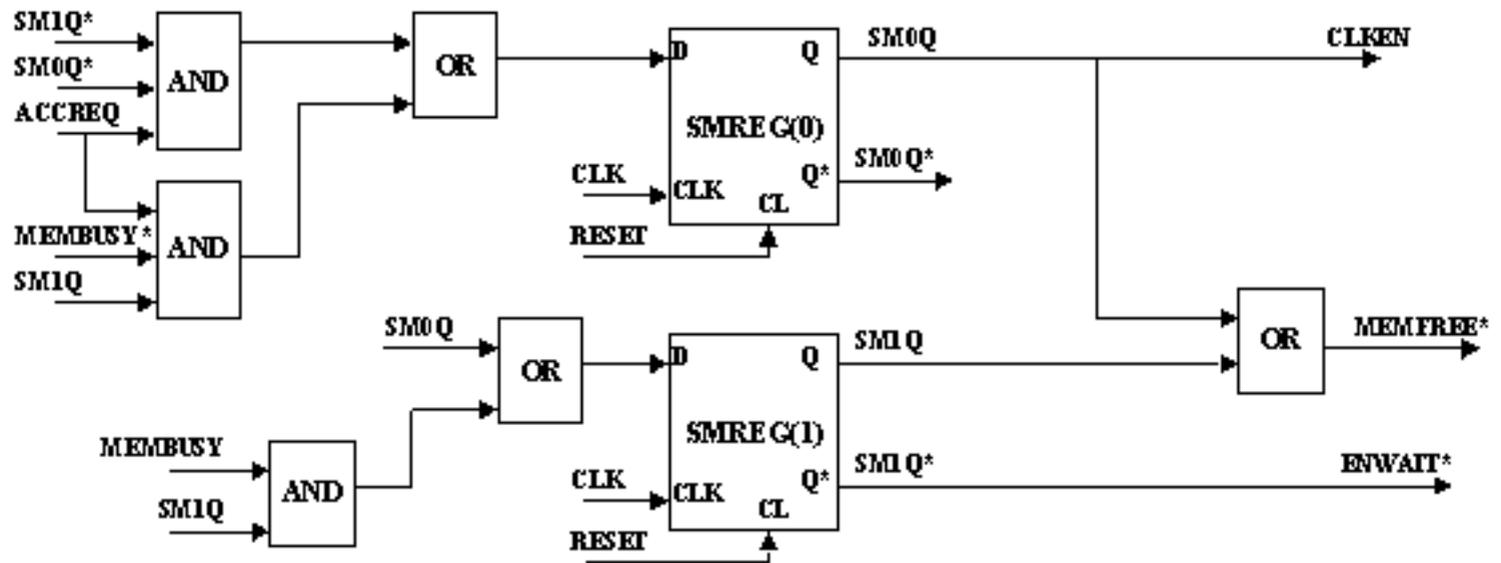
$ENWAIT^* \Leftarrow SMREG(1).Q^*$

$SMREG(0).D \Leftarrow \text{not}(SMREG(1).Q) \text{ and } \text{not}(SMREG(0).Q) \text{ and } ACCREQ \text{ or } SMREG(1).Q \text{ and } \text{not}(MEMBUSY) \text{ and } ACCREQ$

$SMREG(1).D \Leftarrow SMREG(0).Q \text{ or } SMREG(1).Q \text{ and } MEMBUSY$

SHMARB - SCO

Logica macchina a stati



RETI LOGICHE

SECONDA PROVA SCRITTA DELL'APPELLO DEL 4-02-99

Studente: _____ Docente: _____

1. Un dato contenuto in un registro a 32 bit è pari a: AC832000. Se il dato rappresenta un numero floating point, calcolare il suo valore in base 10 secondo la normativa IEEE.
2. Descrivere la struttura di un comparatore ad albero binario e sintetizzare la cella di un generico livello diverso dal primo. Si faccia attenzione agli ingressi della cella.
3. Descrivere la struttura di una rete asincrona autosincronizzante e definire la sua temporizzazione.
4. Descrivere la struttura e la temporizzazione di un sistema di controllo monosequenziale in cui il passaggio dal task T_i al task T_{i+1} è condizionato da una variabile x_k dipendente dal task T_i .
5. Un processore PD32 funziona con clock di 100 nsec e ciclo macchina di 3 periodi di clock. Determinare (in base al tipo di istruzione macchina corrente) il tempo massimo di risposta (emissione del segnale INTA) dall'istante di richiesta di una interruzione.

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 18-02-1999

STUDENTE: _____

DOCENTE: _____

Progetto: DELLINE, linea di ritardo digitale.

Specifiche funzionali:

DELLINE è una linea di ritardo digitale necessaria a compensare il ritardo di codifica audio/video di un codec per videoconferenza; DELLINE ha la funzione principale di ritardare un flusso seriale sincrono a 64Kbps, formato da campioni a 8 bit del segnale audio digitalizzato, di un multiplo intero programmabile di 125uS (durata di una slot a 8 bit). DELLINE riceve il flusso digitale di input tramite le 3 linee IDT (Dati seriali), ICK (Clock) e ISY (Sincronismo di campione: frequenza $1/8 f_{ck}$), genera il flusso di uscita sulle omologhe linee ODT, OCK e OSY ed opera come segue:

1. Per generare il ritardo ingresso-uscita impostato dal PD32 di controllo su di un registro di DELLINE, il sistema utilizza una RAM da 1KB gestita come buffer circolare impostando un offset fra i puntatori di scrittura e lettura corrispondente al ritardo desiderato; il ritardo ingresso-uscita è programmabile nel range 1-128mS a passi di 125uS;
2. il sistema parallelizza i campioni ad 8 bit via via ricevuti, li memorizza nella RAM e contemporaneamente estrae i campioni ritardati del valore impostato;
3. prima di essere serializzati sulle linee di uscita, il livello (valore) dei campioni estratti dalla RAM deve essere moltiplicato per un fattore 0,25 – 0,50 – 0,75 – 1,00 – 1,25 – 1,50 – 1,75 programmabile dal PD32 tramite un registro di controllo.

N.B. Si osservi che, nel transitorio iniziale pari al tempo di ritardo impostato, i dati in uscita dalla RAM non sono significativi.

Si richiedono:

1. il diagramma dettagliato di timing che specifichi il funzionamento della linea di ritardo digitale e delle interfacce seriali sincrone di I/O;
2. lo schema a blocchi di DELLINE con il dettaglio dei segnali di interfacciamento con il PD32 necessari;
3. lo schema elettrico completo di DELLINE.

RETI LOGICHE

SECONDA PROVA SCRITTA DELL'APPELLO DEL 18-02-99

Studente:_____ Docente:_____

- D1 Un codice ridondante ha distanza di Hamming pari a 7. Se si stabilisce di avere una capacità di correzione pari a 2, qual è la sua capacità di rivelazione? (Dare una risposta ragionata)=
- D2 Progettare una rete combinatoria che ha come ingresso un dato ad 8 bit (D_i) e come uscita un dato ad 8 bit (D_o): D_o coincide con D_i se quest'ultimo è compreso nell'intervallo 16-255, altrimenti se $D_i < 16$, $D_o = 16$; se $D_i > 240$, $D_o = 240$
- D3 Un serbatoio per liquido è fornito di due sensori di livello H_L e H_H ($H_L=1$ se il livello è minore di un valore minimo, $H_H=1$ se il livello è maggiore di un valore massimo). Realizzare il comando di una elettrovalvola in modo tale che essa si apra se il livello scende al disotto del minimo e si chiude se il livello supera il livello massimo.
- D4 Una rete asincrona autosincronizzante è realizzata con porte logiche che hanno $t_m=1$ nsec e $t_{Max}=3$ nsec, flip-flop con tempo di commutazione compreso tra 3 e 10 nsec, $t_{setup} = 2$ nsec, tempo di stabilizzazione dell'ingresso di 2 nsec. Calcolare la frequenza massima di funzionamento.
- D5 Specificare il protocollo di comunicazione tra una unità trasmittente TX ed una ricevente RX e definire i due microprogrammi di sincronizzazione.

Corso di RETI LOGICHE

PRIMA PROVA SCRITTA DEL 15-04-1999

STUDENTE: _____

DOCENTE: _____

Progetto:

PCMMON, Monitor di canali audio PCM.

Specifiche funzionali:

PCMMON è una periferica programmabile da un PD32 di controllo dedicata al monitoraggio selettivo di uno dei 32 canali audio digitalizzati trasmessi su di un flusso PCM (Pulse Coded Modulation); il flusso PCM è trasportato da una linea seriale sincrona a 2.048Mbps tramite i 3 segnali DT (Dati seriali), CK (Clock) e FS (Frame Sync, sincronismo di trama PCM). Il flusso PCM è organizzato in trame di 32 slot da 8 bit. Ogni slot rappresenta un campione digitalizzato del relativo canale audio; le trame sono trasmesse consecutivamente con periodo di ripetizione pari a 125 μ S (32x8/2048000).

PCMMON deve eseguire le seguenti funzioni:

1. Sincronizzarsi alla trama PCM mediante i segnali CK e FS;
2. Estrarre, in base al contenuto di un registro caricato dal PD32, una determinata slot (0..31) contenente un campione a 8 bit, convertirlo in parallelo ed regolarne il livello (valore) di un fattore 0,125 - 0,250 - 0,375 - 0,500 - 0,750 - 1.00 - 1,250 - 1,500 programmabile dal PD32 tramite un parametro che viene caricato in un registro di PCMMON; il valore ottenuto deve essere rappresentato con un formato numerico (in virgola fissa) tale da contenere l'intera dinamica del segnale senza introdurre alcun errore di troncamento o di overflow;
3. Il valore numerico ottenuto è inviato ad un convertitore D/A (da non progettare) che alimenta un amplificatore audio di monitoraggio.

Si richiedono:

1. il diagramma dettagliato di timing che specifichi il funzionamento dell'interfaccia PCMMON in relazione al timing dell'interfaccia seriale sincrona PCM;
2. la risoluzione necessaria per il convertitore d'uscita;
3. lo schema a blocchi di PCMMON con il dettaglio dei segnali di interfacciamento con il PD32 necessari;
4. lo schema elettrico completo di PCMMON.

Nota. Il convertitore D/A va visto come un blocco combinatorio.

RETI LOGICHE

PRIMA PROVA SCRITTA DEL 03-06-1999

STUDENTE: _____

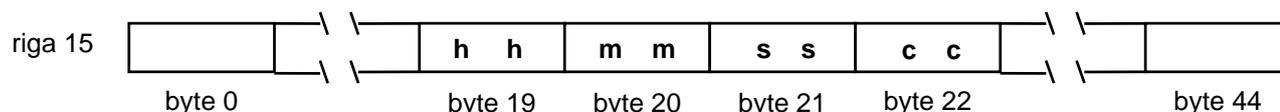
DOCENTE: _____

Progetto:

Sincronizzatore Televideo → Microprocessore: TLVSYNC

Specifiche funzionali:

Un sistema teletext utilizza le righe 8..15 (non visualizzate, perché comprese nell'intervallo di cancellazione verticale) di un segnale televisivo per diffondere informazioni testuali (televideo). Nella trama video organizzata su 625 righe consecutive per quadro e 25 quadri al secondo, ogni riga video di tipo teletext trasporta un pacchetto di 360 bits, corrispondenti a 45 bytes. Sui bytes 19, 20, 21 e 22 del pacchetto della riga 15 di ogni quadro viene trasmessa una informazione in BCD rappresentante ora (hh), minuto (mm), secondo (ss) e centesimo di secondo (cc) rispettivamente (codice temporale), come delineato in figura (il pacchetto inizia con il byte 0, il byte inizia con il bit più significativo).



Alla periferica TLVSYNC è affidato il compito di estrarre l'informazione di cui sopra dalla trama video e di comunicarla periodicamente ($T=1$ sec.) al processore, che la utilizzerà come riferimento su cui sincronizzare le proprie attività interne.

Oltre alla linea dati seriale SDA su cui vengono presentati i bit, si suppongano disponibili in ingresso a TLVSYNC una linea di clock SCK per la sincronizzazione dei bit, e una linea PSYNC per la sincronizzazione dei pacchetti, attiva per due periodi di SCK all'inizio del pacchetto della riga 8 e per un periodo di SCK all'inizio dei pacchetti delle righe $i=9..15$.

La periferica TLVSYNC effettua le seguenti attività:

1. si sincronizza con le trame video;
2. estrae il codice temporale per renderlo disponibile in un registro a larghezza 32 bits accessibile in lettura da PD32;
3. ad ogni nuovo secondo, lancia un interrupt al processore.

In risposta il PD32 preleva il codice temporale dalla periferica e aggiorna una locazione di memoria longword **TIMESTAMP**.

Si richiede:

- lo schema a blocchi funzionali di TLVSYNC ed il diagramma di temporizzazione dell'interfaccia con la linea seriale di ingresso;
- lo schema logico di TLVSYNC;
- la routine assembler PD32 di servizio della periferica.

Note/suggerimenti:

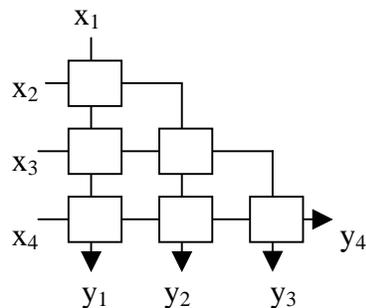
Nel codice temporale relativo ad un nuovo secondo il campo cc può non essere nullo.

RETI LOGICHE

SECONDA PROVA SCRITTA DELL'APPELLO DEL 3-6-99

Studente: _____ Docente: _____

D1 Si consideri la struttura iterativa di figura:



La generica cella ha due ingressi EST e NORD e due uscite SUD e OVEST; la cella invia a SUD il minore/uguale dei due ingressi e a OVEST il maggiore/uguale dei due ingressi: definire le due funzioni logiche della cella e descrivere verbalmente il comportamento ingresso-uscita dell'intera struttura: dato il vettore $\langle x_1, x_2, x_3, x_4 \rangle$, qual è il vettore $\langle y_1, y_2, y_3, y_4 \rangle$?

D2 Si considerino due stati S e T di due macchine parzialmente specificate e la seguente coppia di sequenze di uscita:

S: 0-110-11001--010-1

T: 01-0-011--0111-001

In base a questa coppia di sequenze è possibile stabilire una relazione tra i due stati?

D3 Una rete sequenziale sincrona è realizzata con una PLA ($\tau_{MAX}= 10$ nsec, $\tau_{min}=2$ nsec) e un registro con flip-flop D ($t_{setup}= 3$ nsec; $t_c= 5$ nsec; $t_{hold}=2$ nsec). Calcolare la frequenza massima di funzionamento.

D4 Definire la sequenza di microistruzioni per eseguire l'istruzione

MOVQuick byte, dest

di tipo immediato, dove il dato immediato è un byte contenuto nei bit 16-23 dell'istruzione (è una istruzione non implementata nel PD32).

D5 Specificare la sequenza completa di operazioni per effettuare il trasferimento di un blocco di dati da una periferica alla memoria di un PD32 in DMA.

RETI LOGICHE

SECONDA PROVA SCRITTA DELL'APPELLO DEL 17-6-99

Studente: _____ **Docente:** _____

- D1 Si consideri un codice autocorrettore di 12 bit a distanza di Hamming pari a 4; calcolare la probabilità di non rivelare errore in una parola di codice, supposto che la probabilità di errore in un bit è pari a 10^{-7} .
- D2 Progettare una macchina sequenziale che accetta in ingresso simboli 0 e 1 e dia una uscita 1 ogni volta che riconosce una sequenza del tipo $01(00)^n1$ con n intero e finito.
- D3 Una struttura di calcolo è costituita da tre blocchi combinatori in cascata con tempi di calcolo pari a 20, 40, 20 nsec. Trasformarla in una struttura pipeline in modo da avere tempo di ciclo pari a 25 nsec e tracciare la temporizzazione relativa ai vari stadi.
- D4 Un PD32 ha il ciclo di accesso alla memoria di 200nsec. Calcolare di quanto viene rallentato durante l'operazione di trasferimento dati in DMA da un disco che abbia una velocità di lettura/scrittura di 12 Mbit/sec. Si supponga di trasferire dati di 32 bit alla volta.
- D5 Descrivere la temporizzazione di un sistema SCA-SCO di tipo DMealy-Mealy.

RETI LOGICHE
PRIMA PROVA SCRITTA DELL'APPELLO DEL 30-06-99

Studente: _____ **Docente:** _____

Un microprocessore PD32 deve comandare una macchina utensile costituita da un trapano a colonna e da una slitta di lavoro su cui viene posizionato un pezzo da forare. La slitta di lavoro può spostarsi lungo due assi X-Y in modo da presentare sulla perpendicolare della testa del trapano il punto da forare; la testa del trapano, a sua volta, è in grado di ruotare lungo un asse orizzontale per cambiare la punta foratrice (si supponga che siano possibili quattro punte diverse). Un programma memorizzato nel microprocessore dovrà dare i comandi alla macchina utensile per eseguire sul pezzo una serie di fori di diametro diverso e in posizioni prestabilite. Le informazioni che riceve la macchina sono:

- posizione del foro sul piano X-Y
- diametro del foro

Un sensore posto sulla macchina utensile indica con un segnale 0/1 quando la punta del trapano ha trapassato il pezzo da forare.

La precisione di posizionamento del foro deve essere migliore di 0,01 mm. su una escursione massima di 300 mm. I motori di azionamento sono passo-passo: essi funzionano con comando ad impulsi e un impulso corrisponde ad uno spostamento di 0,005 mm della piastra in un senso o in senso opposto a seconda del valore di un segnale di controllo che indica il senso di rotazione. La selezione della punta viene effettuata in funzione del diametro del foro. Il clock con cui funziona la macchina utensile è dato.

La discesa e la risalita della punta avviene mediante un motore passo passo dello stesso tipo di quelli che comandano gli spostamenti della piastra. Al termine di ogni risalita della punta, la macchina richiede al PD32 le informazioni relative al foro successivo.

Al termine del programma di foratura il microprocessore manda un comando per riportare la slitta in posizione di riposo.

Si richiede:

- 1) l'interfaccia tra microprocessore e macchina utensile
- 2) la logica di comando dei motori passo-passo
- 3) la struttura del programma di controllo.

RETI LOGICHE

SECONDA PROVA SCRITTA DELL'APPELLO DEL 30-06-99

Studente: _____ Docente: _____

- D1: disegnare lo schema di una rete a PLA per 10 variabili di ingresso, realizzata con moduli PLA ad 8 ingressi e con uscita tristate.
- D2: definire le funzioni G^* e P^* (generazione e propagazione di gruppo) per un addizionatore a propagazione rapida del riporto e disegnare lo schema per un adder a 16 bit utilizzando blocchi di 4 bit.
- D3: illustrare i passi per la minimizzazione di macchine parzialmente specificate, partendo dalla copertura finale.
- D4: sintetizzare una macchina sequenziale che si comporti come un contatore decimale up/down con ingressi CE (count enable) U/D (up/down) ed uscita TC (terminal count).
- D5: scrivere una subroutine per trasferire il contenuto del registro R1 nel registro R2: nel trasferimento in R2 viene testato il bit 15 della long word e se questo è ad 1 viene scambiata la parola meno significativa con la più significativa.

RETI LOGICHE

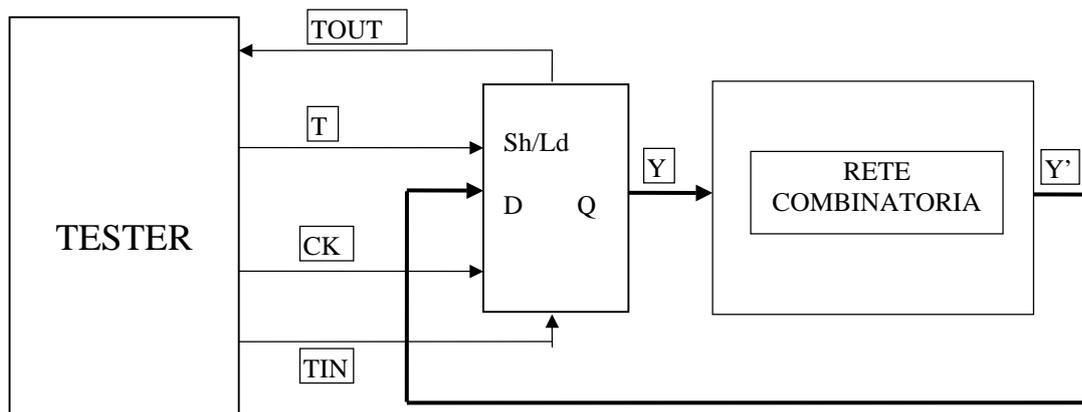
PRIMA PROVA SCRITTA DELL'APPELLO DEL 15-07-99

STUDENTE: _____

DOCENTE: _____

Specifiche funzionali:

In una linea di produzione di un circuito integrato LSI si vuole allestire una stazione di collaudo (TESTER) per individuare i chip guastati. Il circuito LSI può essere schematizzato (vedi figura) come una struttura sequenziale sincrona, predisposta al collaudo mediante l'impiego di un registro di stato a scorrimento, di lunghezza 128 bit.



I segnali di interfacciamento con il TESTER sono i seguenti:

- CK: è terminato sull'ingresso di clock del circuito sotto test;
- TIN, TOUT: ingresso e uscita seriali del registro di stato; vengono utilizzati solo in fase di test.
- T: se vale 1 provoca lo scorrimento del registro di stato; altrimenti il registro viene caricato in parallelo (nel funzionamento normale T sarà ovviamente fissato a 0).

Il collaudo consiste nella ripetizione di 64K cicli di test, il k-esimo dei quali è composto di 4 passi:

- 1 - il registro viene caricato serialmente attraverso TIN con una configurazione (vettore di test) Y_k ;
- 2 - il registro viene caricato in parallelo con il vettore Y_k' calcolato dalla rete combinatoria;
- 3 - il registro viene scaricato serialmente attraverso TOUT.
- 4 - il vettore Y_k' viene confrontato con un vettore E_k precalcolato: in caso di coincidenza tra le due sequenze si passa al ciclo k+1, altrimenti il collaudo verrà terminato.

Se tutte le 64K coppie di vettori (Y_k' , E_k) coincidono, il collaudo termina con successo. Il TESTER inizia il collaudo di un chip a seguito di una richiesta di un PD32. Al termine del collaudo il TESTER invia una interruzione comunicando l'esito del collaudo.

Tutti i vettori Y_k, Y_k', E_k sono a 128 bit. I vettori Y_k, E_k (precalcolati in fase di progetto del circuito LSI) sono memorizzati in una serie di chip ROM di dimensione 256K x 8 bit e tempo di accesso di 200 ns. Il circuito deve essere collaudato ad una frequenza di clock pari a 50 MHz

Si richiede:

- lo schema a blocchi funzionali del TESTER;
- il dimensionamento del banco di ROM e la organizzazione interna dei dati di test;
- il diagramma di temporizzazione dell'interfaccia con il circuito integrato dato;
- lo schema logico dettagliato dello SCA e dello SCO del TESTER;

Note/Suggerimenti:

Si noti che il passo 3 del ciclo di test k-esimo può essere sovrapposto temporalmente con il passo 1 del ciclo di test (k+1)-esimo.

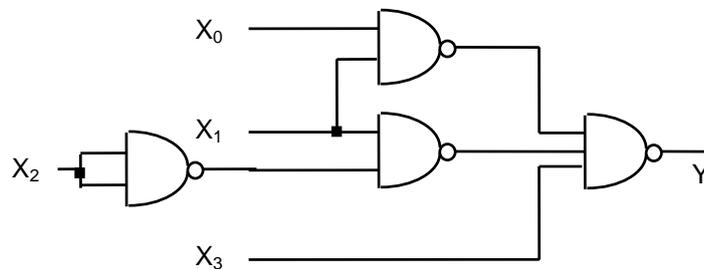
RETI LOGICHE

SECONDA PROVA SCRITTA DELL'APPELLO DEL 15-07-99

Studente: _____ Docente: _____

D1 - Rappresentare con la notazione in virgola mobile in base 2 i numeri 1999.25 e 1.75 e descrivere i passi per effettuarne la somma.

D2 - Trasformare la rete combinatoria in figura in una rete equivalente a tre livelli costituita esclusivamente da porte logiche di tipo NOR.



D3 - Dato un registro a scalamento a 8 bit, aggiungere una rete in modo che la lunghezza risulti programmabile da 1 a 8 bit, in funzione dello stato logico di 3 bit di controllo.

D4 - Dato lo SCA del PD32, definire un microprogramma per interpretare la fase di esecuzione dell'istruzione ipotetica XNORL R0,R1. Indicare su un diagramma temporale la dinamica dei bit di TASK e dei contenuti dei registri coinvolti.

D5 - Una periferica richiede al PD32 una sezione della sua memoria per appoggiare dati temporanei secondo la disciplina LIFO. Ciascun dato a 32 bit viene depositato e successivamente recuperato dalla periferica tramite interruzione al processore che provvede al servizio. Una coppia di FF con significato di stack vuoto e stack pieno vengono settati dal PD32 per prevenire eventuali richieste anomale da parte della periferica. Definire l'interfaccia HW di I/O della periferica ed i moduli del programma assembler PD32.

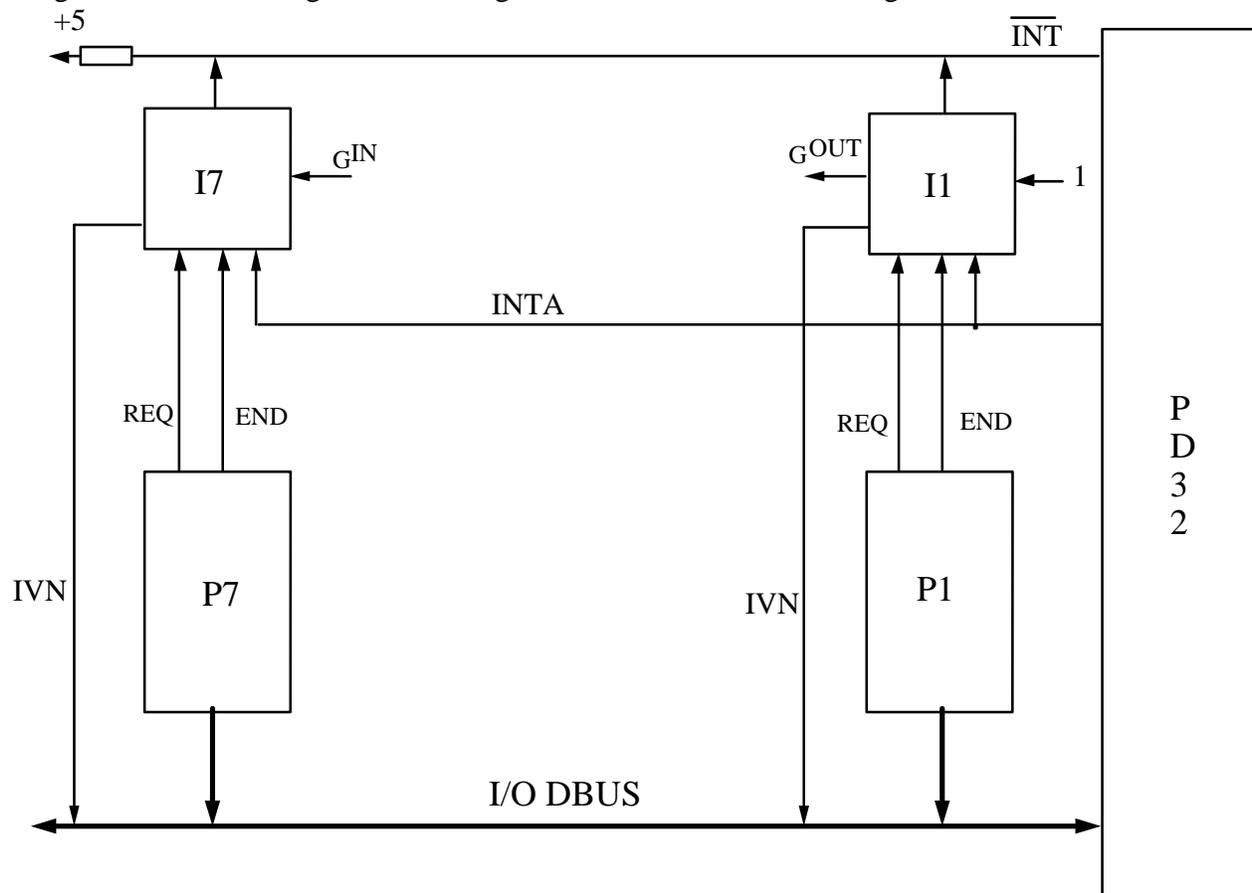
RETI LOGICHE

PRIMA PROVA SCRITTA DELL'APPELLO DEL 24-9-99

Studente: _____ Docente: _____

Un processore PD32 ha sette periferiche di ingresso gestite tramite interruzioni a priorità decrescente (Per.1 priorità massima; Per.7 priorità minima). Quando la periferica i -ma vuole mettersi in comunicazione col PD32, manda una richiesta di interruzione (INTREQ) alla propria interfaccia. L'interfaccia i -ma invia l'interruzione al processore se il suo G^{IN} è alto, nel qual caso mette a zero il suo G^{OUT} per inibire le periferiche di priorità più bassa. Se durante il servizio della Per. i una periferica h ($h < i$) chiede a sua volta il servizio, Per. i si mette in stand-by (attesa) e riprenderà il servizio al termine della comunicazione tra Per. h e PD32. Quando è terminato il servizio, Per. i avvisa la propria interfaccia con un segnale END. Lo schema di interconnessione è riportato in figura.

Progettare l'interfaccia generica e disegnare lo schema elettrico dettagliato.



Si noti che le interruzioni I2-I7 sono di tipo interrompibile; specificare come devono essere gli inizi delle routine di servizio di P2-P7 per il corretto funzionamento.

RETI LOGICHE

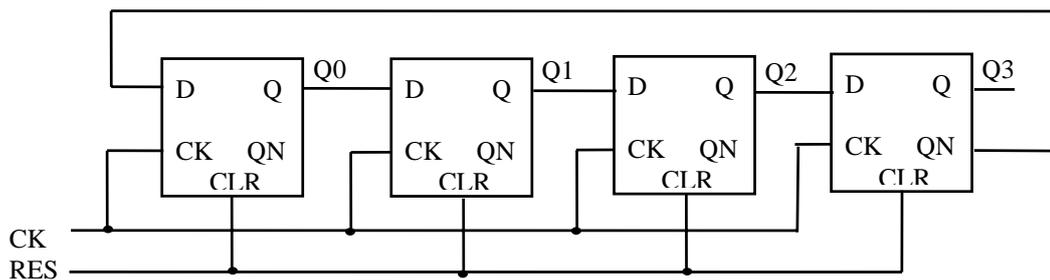
SECONDA PROVA SCRITTA DELL'APPELLO DEL 24-09-99

Studente: _____ Docente: _____

D1 - Data una rappresentazione numerica in virgola fissa a 24 bit con il formato II.DDDD (I: nibble parte intera; D: nibble parte frazionaria) con notazione in complemento a 2, definire un formato in virgola mobile che includa tutti i numeri rappresentabili con la notazione originale.

D2 - Data una ROM che funziona da moltiplicatore tra nibble (gruppi di 4 bit); progettare una rete combinatoria costituita di moduli ROM definiti come sopra e addizionatori per effettuare la moltiplicazione tra byte.

D3 - Analizzare il comportamento dinamico del sistema sequenziale sincrono autonomo riportato in figura.



Commentare le proprietà della sequenza degli stati scanditi a partire dall'inizializzazione e quindi generalizzarle per un anello a N flip-flop.

D4 - Un processore microprogrammato può eseguire 32 diversi microprogrammi, di cui uno (fetch) comune a tutti i cicli-istruzione. I microprogrammi hanno una lunghezza massima di 200 microistruzioni. Descrivere la struttura hardware dello SCO multimicroprogrammato di tipo D-mealy e l'allocazione dei microprogrammi nella ROM.

D5 - Nella memoria PD32 a partire dall'indirizzo SAMPLES sono allocati 256 bytes che rappresentano i valori dei campioni acquisiti tramite un convertitore analogico-digitale a 8 bit. Si richiede di scrivere una routine assembler PD32 che calcoli il valor medio dei campioni memorizzati e lo memorizzi in una locazione di indirizzo MEDIA