Calcolatori Elettronici T

Ing. Informatica

Traccia soluzione

10 Febbraio 2021

Esercizio 1

L'esercizio prevede di accumulare modulo 2^{16}, mediante una rete logica RL da progettare, i byte *unsigned* letti da una porta in input con *handshake*.

Inoltre, è necessario mappare una porta in output, sempre con handshake, su bus dati diversi in base al resto della divisione per 4 del valore accumulato da RL. Il testo del problema indica di minimizzare il numero di istruzioni di branch nell'interrupt handler per gestire la collocazione dinamica della porta in output. A tal fine, nella traccia, verranno replicati i byte sui quattro bus con una strategia software in modo da inviare il byte alla porta in output indipendentemente dalla collocazione sul bus dati e senza dover leggere esplicitamente tale collocazione.

Infine, si ricorda che nella gestione degli interrupt generati dalle due porte con handshake, la gestione degli eventi originati dalla porta in input è prioritaria.

Dispositivi e segnali presenti nel sistema.

Dispositivi di memoria:

RAM_H	**mappata da 80000000h:FFFFFFFFh, 4 banchi da 512 MB**
RAM_L	**mappata da 60000000h:7FFFFFFFh, 4 banchi da 128 MB**
EPROM	**mappata da 00000000h:3FFFFFFFh, 4 banchi da 256 MB**

Altri dispositivi e/o segnali:

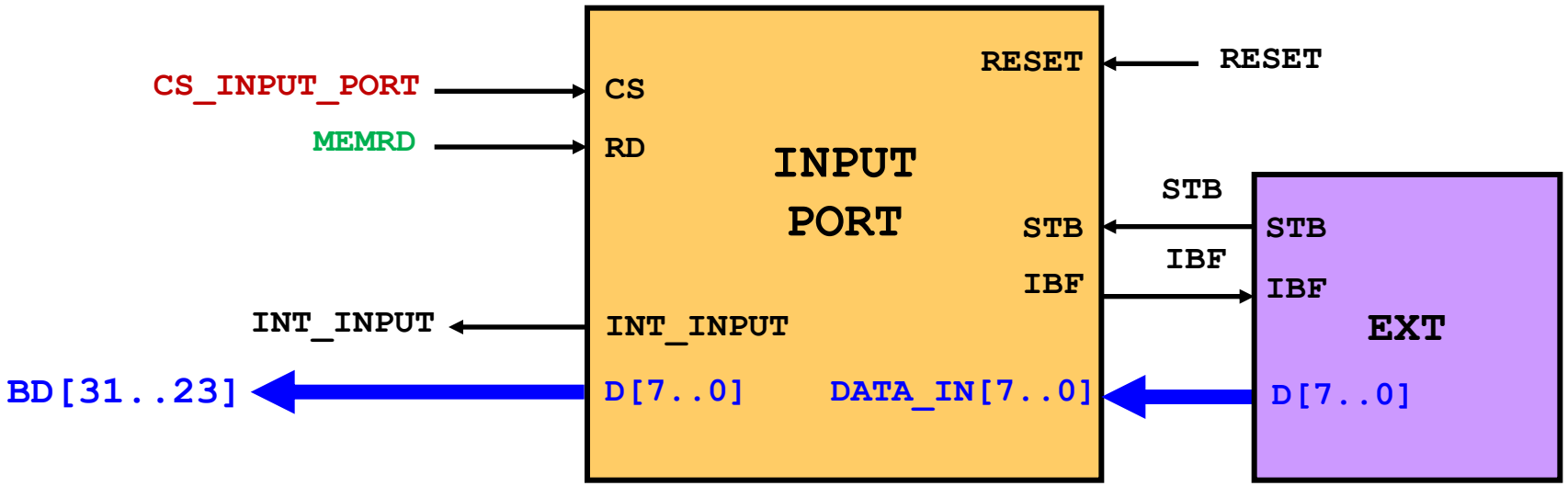
CS_INT_INPUT	**mappato a 40000000h**
CS_INT_OUTPUT	**mappato a 40000001h**
CS_READ_RL	**mappato a 40000002/3h**
CS_OUTPUT_PORT	**mappato a 40000004/5/6/7h** (dipende da resto divisione)
CS_FREEZE_INT	**mappato a 40000008h**
CS_INPUT_PORT	**mappato a 4000000Bh**

Segnali di decodifica di memorie, periferiche e segnali:

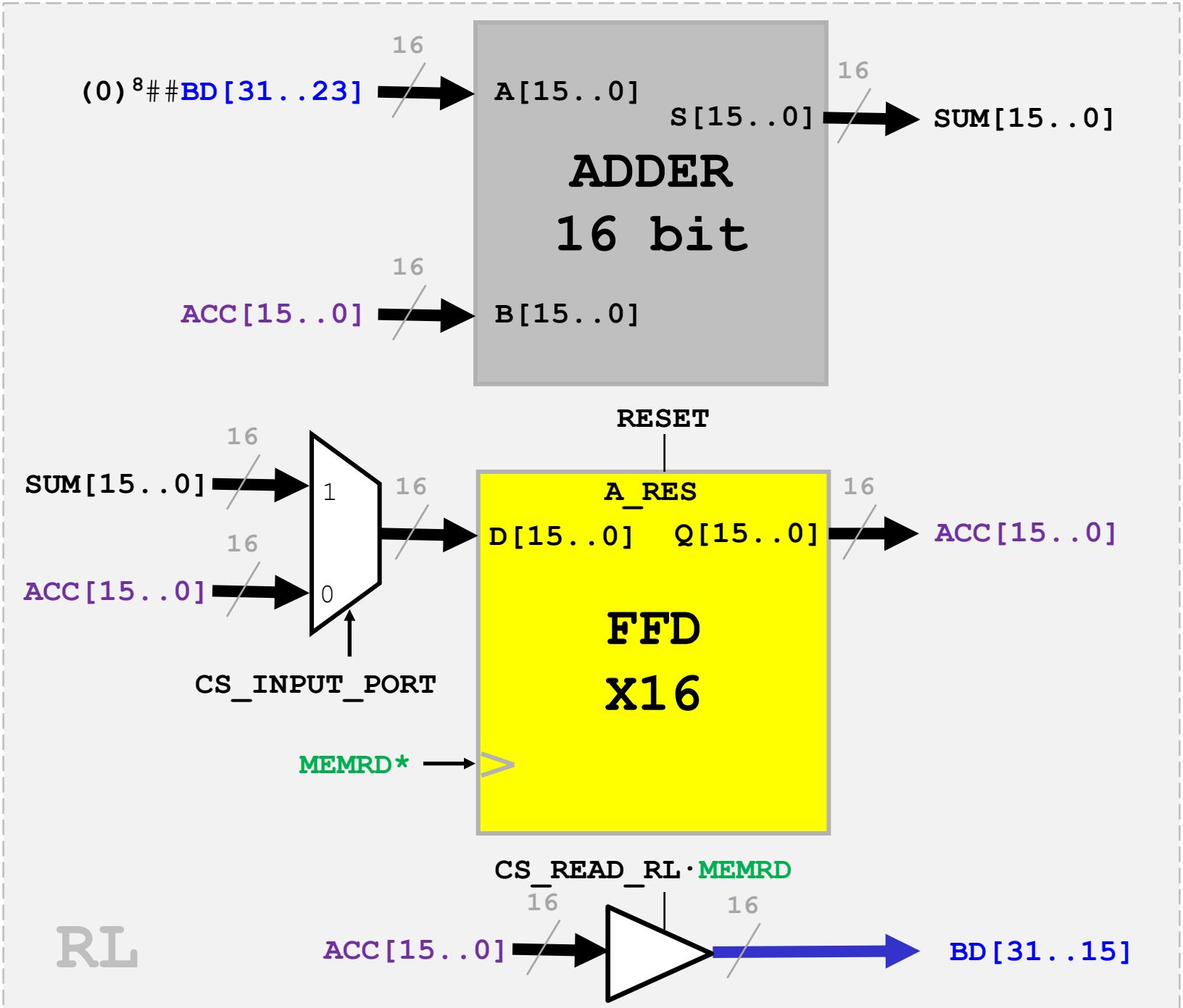
```
CS_RAM_H_0       = BA31·BE0
CS_RAM_H_1       = BA31·BE1
CS_RAM_H_2       = BA31·BE2
CS_RAM_H_3       = BA31·BE3

CS_RAM_L_0       = BA31*·BA30·BA29·BE0
CS_RAM_L_1       = BA31*·BA30·BA29·BE1
CS_RAM_L_2       = BA31*·BA30·BA29·BE2
CS_RAM_L_3       = BA31*·BA30·BA29·BE3

CS_INPUT_PORT    = BA31*·BA30·BA29*·BA3·BE3          mappato a 4000000Bh
CS_FREEZE_INT    = BA31*·BA30·BA29*·BA3·BE0          mappato a 40000008h
CS_OUTPUT_PORT   = BA31*·BA30·BA29*·BA3*·BA2         mappato a 40000004/5/6/7h(*)
CS_READ_RL       = BA31*·BA30·BA29*·BA3*·BA2*·BE3·BE2 mappato a 4000002/3h
CS_INT_OUTPUT    = BA31*·BA30·BA29*·BA3*·BA2*·BE1    mappato a 40000001h
CS_INT_INPUT     = BA31*·BA30·BA29*·BA3*·BA2*·BE0    mappato a 40000000h

CS_EPROM_0       = BA31*·BA30*·BE0
CS_EPROM_1       = BA31*·BA30*·BE1
CS_EPROM_2       = BA31*·BA30*·BE2
CS_EPROM_3       = BA31*·BA30*·BE3
```

(*) l'esatta posizione dipende da resto della divisione per 4 del valore accumulato da RL, come sarà mostrato nelle pagine successive

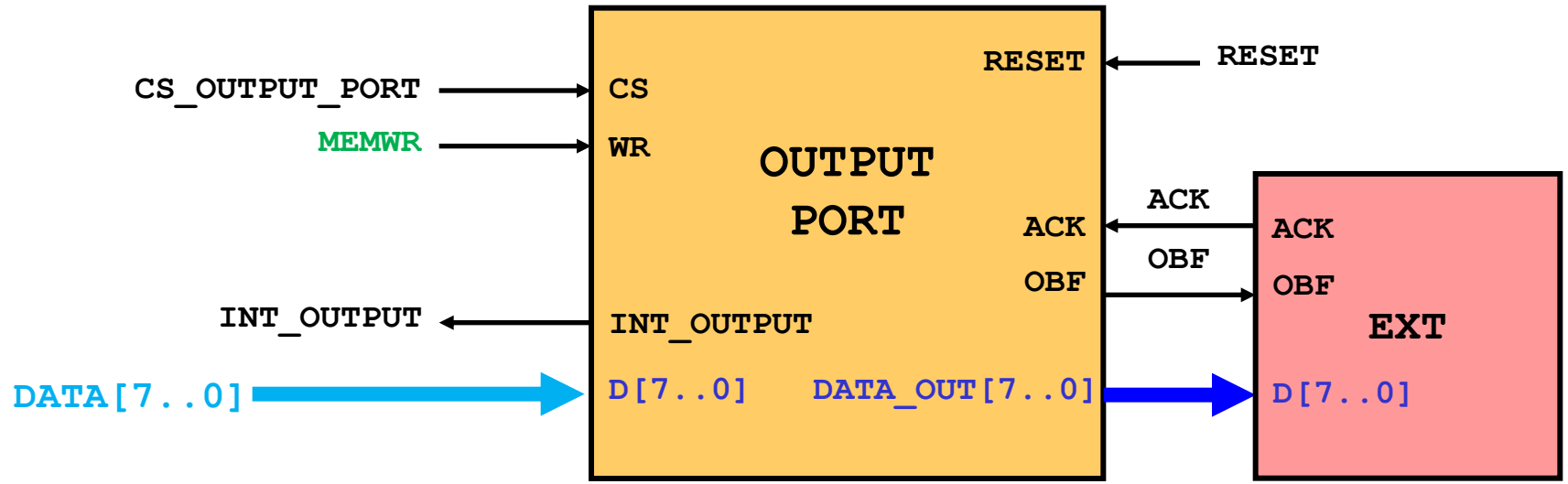
Nel sistema è presente una porta in input che comunica con l'esterno mediante il protocollo di handshake. La porta, come dalle specifiche del testo, è mappata sul bus dati **BD[31..23]**

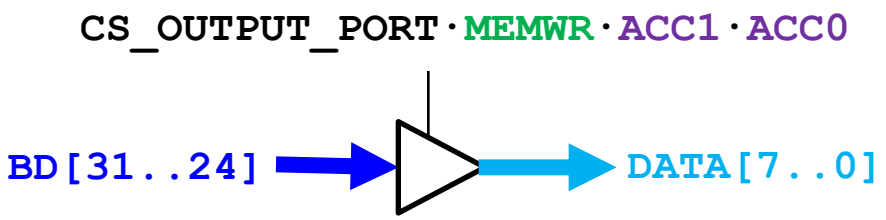


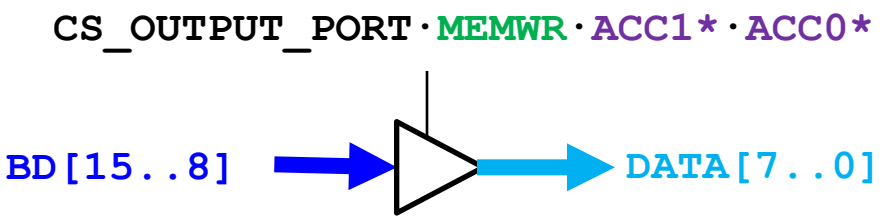
Nella pagina successiva è mostrata la rete logica **RL** che consente di effettuare l'accumulo dei dati, 8 bit di tipo *unsigned*, letti da **INPUT_PORT**. Inoltre, sono presenti le connessioni per leggere da **RL** il valore accumulato al suo interno attraverso i pin del bus dati **BD[31..15]**, come da specifiche del testo, mediante opportuni dispositivi tri-state.
All'avvio, la rete logica **RL** è inizializzata al valore (di accumulo) 0 mediante il segnale RESET.

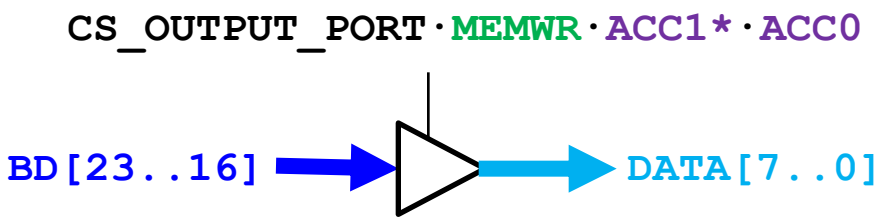

(0)8##BD[31..23]
16
A[15..0]
S[15..0]
16
SUM[15..0]
ADDER
16 bit
ACC[15..0]
16
B[15..0]
RESET
SUM[15..0]
16
1
16
A_RES
16
ACC[15..0]
16
D[15..0]
Q[15..0]
ACC[15..0]
0
FFD
X16
CS_INPUT_PORT
MEMRD*
CS_READ_RL·MEMRD
16
16
ACC[15..0]
BD[31..15]
RL

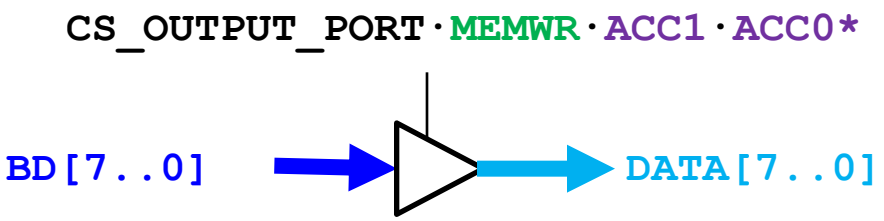
Nel sistema è presente anche una porta in output che comunica con l'esterno mediante il protocollo di handshake. La porta, come da specifiche del testo, è mappata su bus dati diversi in funzione del resto della divisione per 4 del valore accumulato da **RL**.



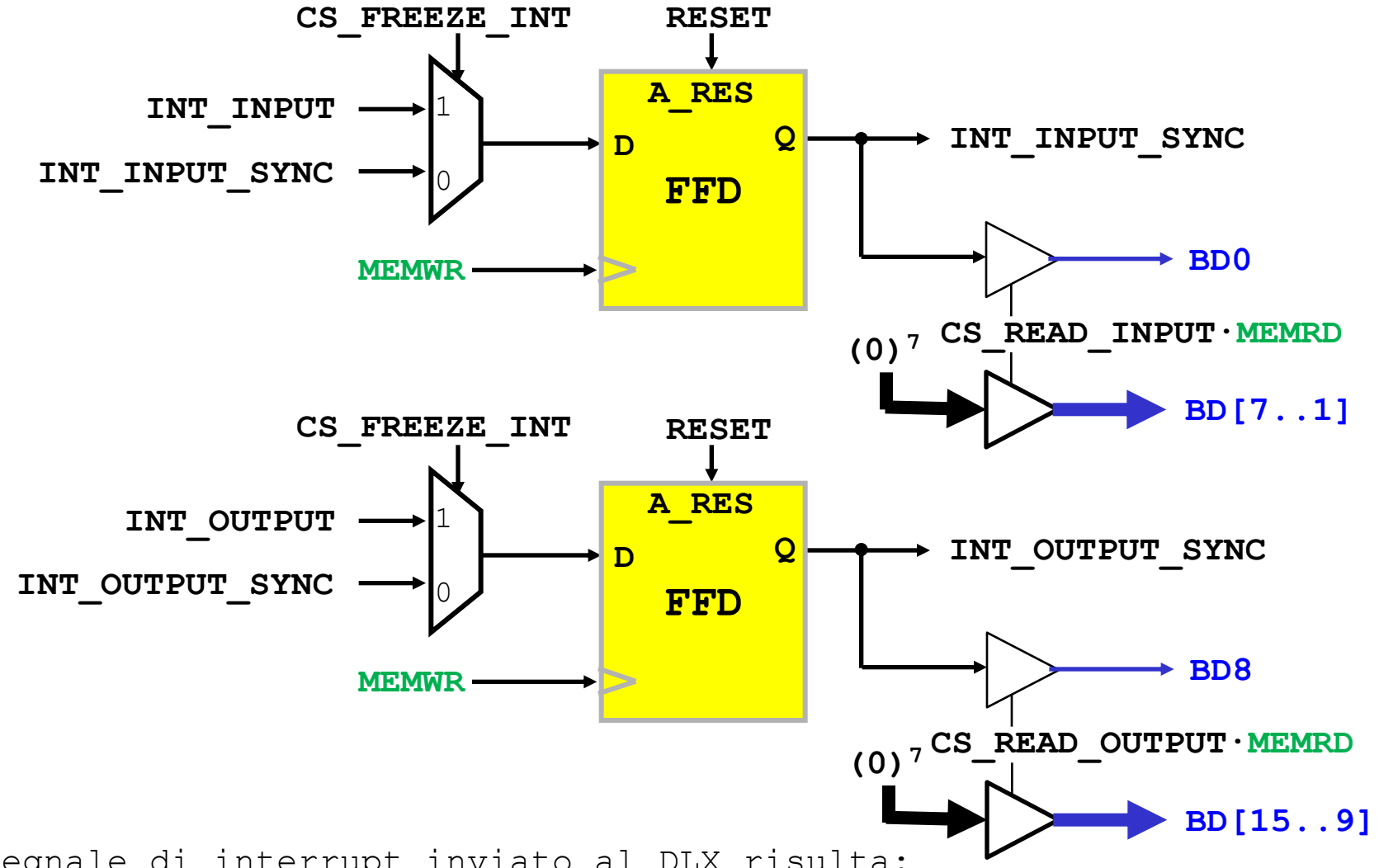








I segnali di interrupt generati dalle due porte saranno campionati mediante un comando software che asserisce **CS_FREEZE_INT**, e letti separatamente al fine di ridurre il numero di istruzioni di *branch* nell'handler.



Il segnale di interrupt inviato al DLX risulta:

INT (to DLX) = INT_INPUT + INT_OUTPUT

Per semplicità, nella traccia è omessa l'infrastruttura e relativo codice per gestire l'avvio del sistema. Il codice dell'*interrupt handler* che gestisce i trasferimenti dalle due porte risulta:

```
        00000000 LHI   R20,4000h       ; R20 = 40000000h
        00000004 SB    R0,R20(0008h)   ; dummy write a CS_FREEZE_INT
        00000008 LHI   R21,E000h       ; R21 = E0000000h
        0000000C LBU   R25,R20(0)      ; legge INT_INPUT_SYNC
        00000010 BEQ   R25,output:     ; se R25=0 salta a output:
        00000014 LBU   R25,R20(000Bh)  ; legge da INPUT_PORT
        00000018 SB    R25,R21(0123h)  ; scrive R25 in E0000123h
output: 0000001C LBU   R25,R20(0001h)  ; legge INT_OUTPUT_SYNC
        00000020 BEQ   R25,fine:       ; se R25=0 salta a fine:
        00000024 LBU   R25,R21(3211h)  ; R25 = 000000XX
        00000028 SLLI  R24,R25,8h      ; R24 = 0000XX00
        0000002C OR    R23,R24,R25     ; R23 = 0000XXXX
        00000030 SLLI  R24,R23,10h     ; R24 = XXXX0000
        00000034 OR    R25,R23,R24     ; R25 = XXXXXXXX
        00000038 SW    R25,R20(0004h)  ; scrive in OUTPUT_PORT una word
fine:   0000003C RFE                   ; ritorna da interrupt
```

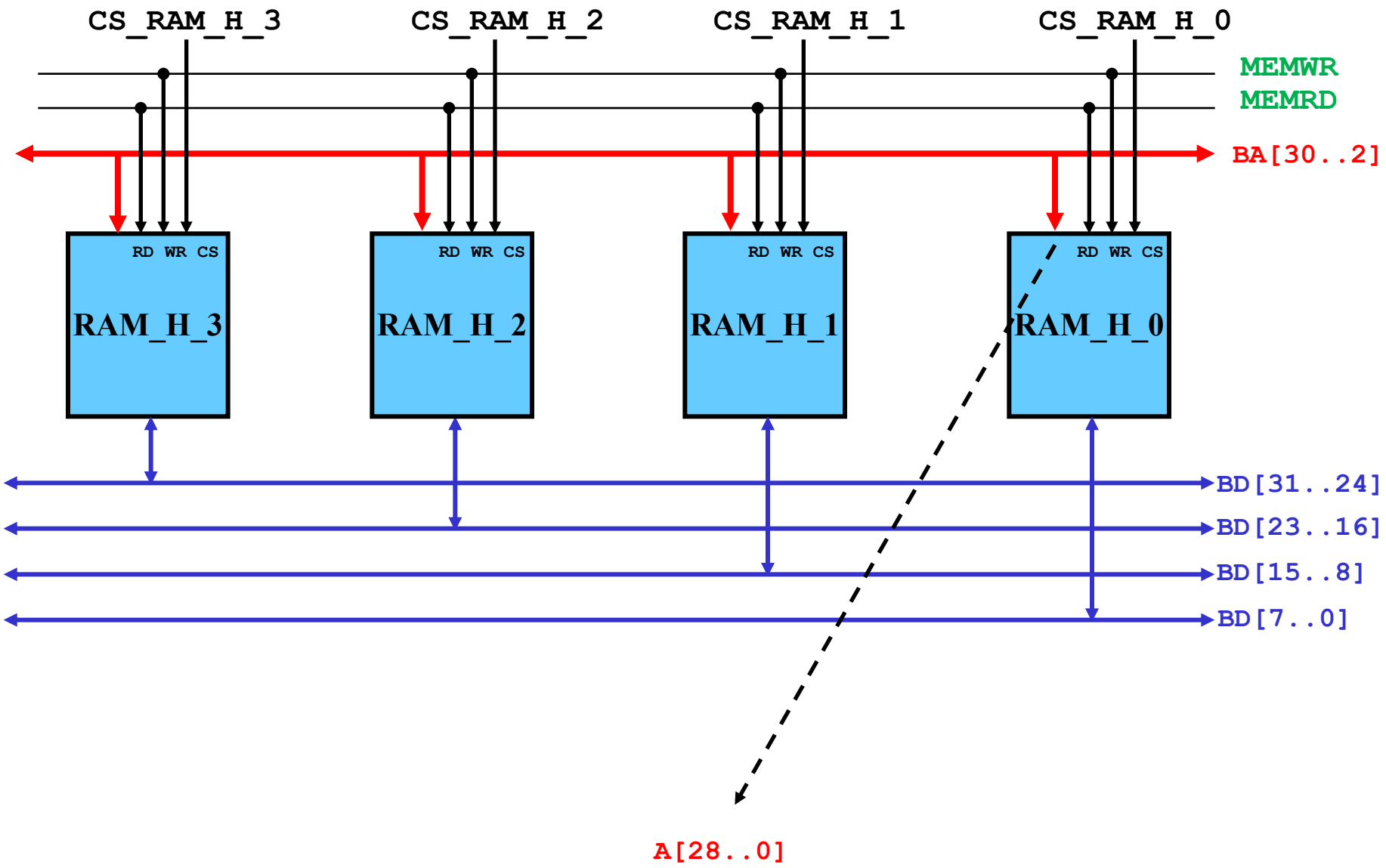
La label **output:** vale 8h mentre **fine:** vale 18h (24 decimale)

Nota: il salto a 00000010h, potrebbe avere come destinazione l'indirizzo 00000024h (vs. 0000001Ch) poiché, essendo in esecuzione l'handler, se **INT_INPUT_PORT_SYNC=0** sicuramente **INT_OUTPUT_PORT_SYNC=1** riducendo il numero di istruzioni in queste specifiche circostanze.

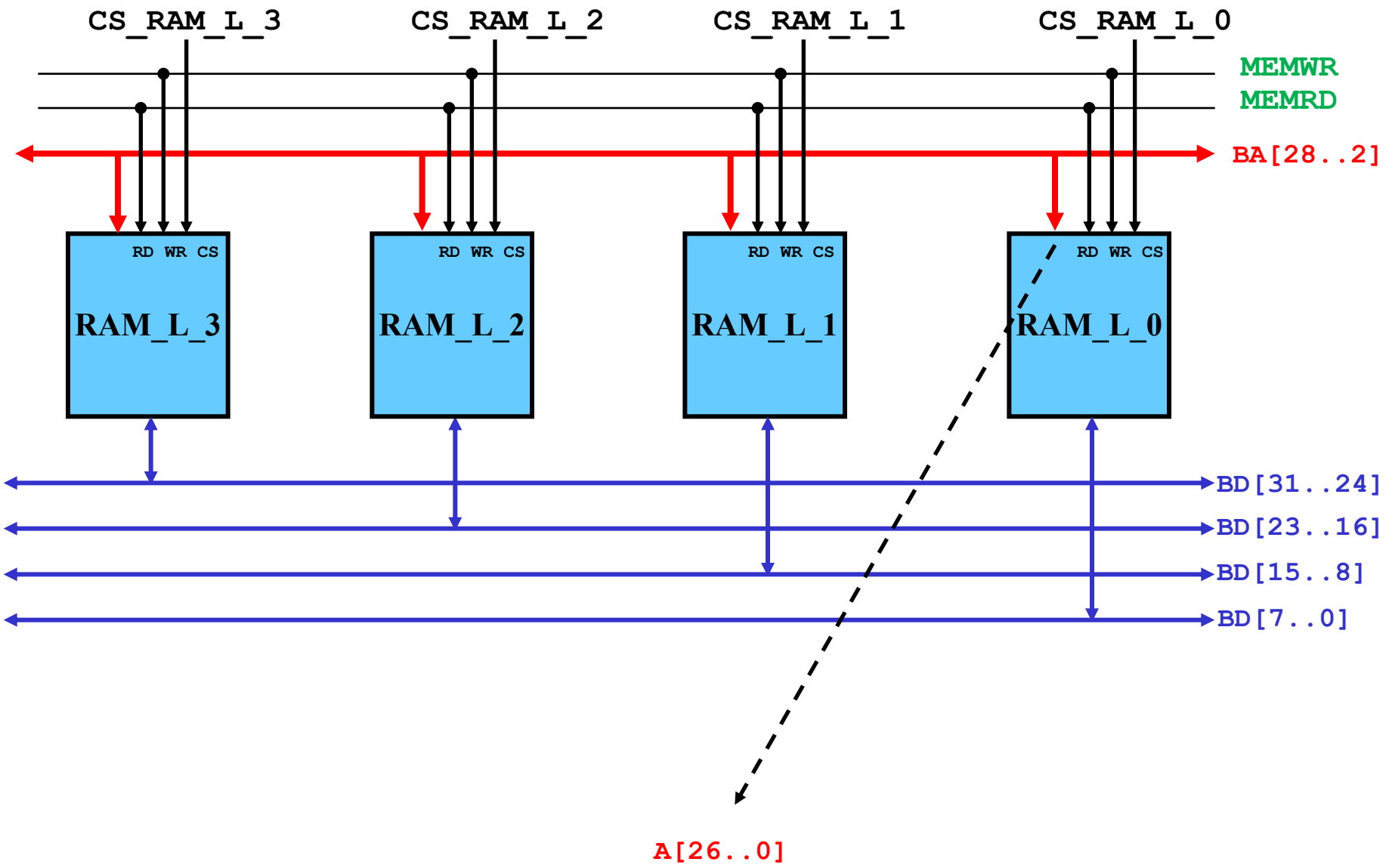
In seguito è riportato il codice per leggere da **RL** il valore a 16 bit accumulato al suo interno:

```
00001000 LHI   R7,4000h        ; R7 = 40000000h
00001004 LHU   R8,R7(0002h)    ; legge half-word da RL
```

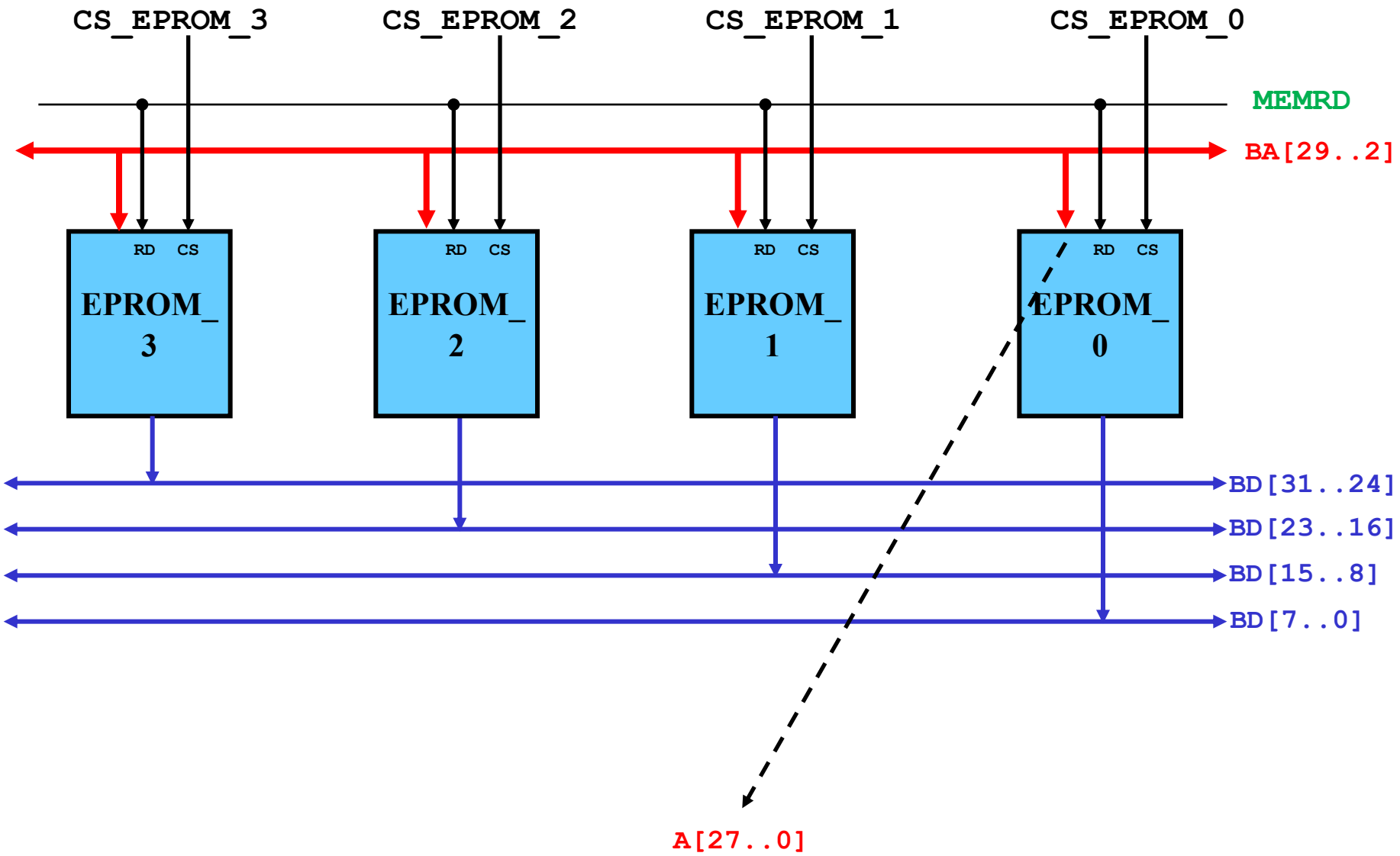
Interfacciamento RAM_H



Interfacciamento RAM_L



Interfacciamento EPROM



Esercizio 2

Rif. lucidi/lezioni.

Esercizio 3

Rif. lucidi/lezioni.