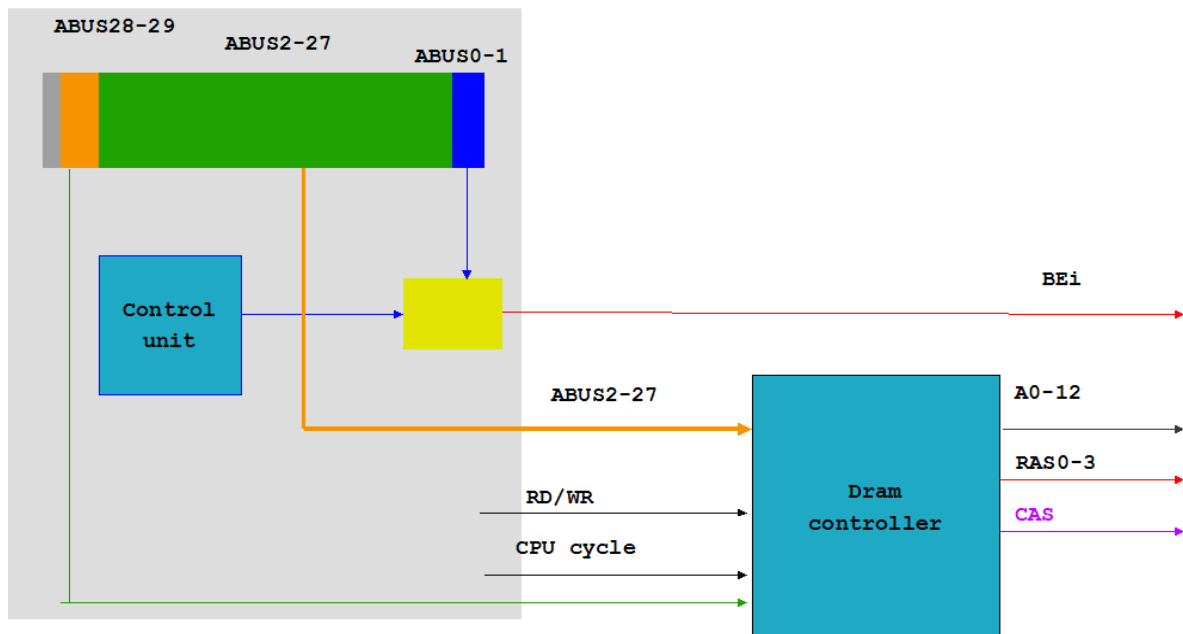


## LEZIONE 15/10/08 (slide 20-28)

# Schema di memoria dram



Prof. Marco Mezzalama 08/09

Il DBus ha un parallelismo di 32 bit, ne consegue che la ram è organizzata in 4 byte.

Essendo il bus da 4 byte, l'architettura logica sarà di 4 banchi, 256MB l'uno (ogni banco porrà sul DBus un byte), 4 chip da 64MB l'uno per banco. Ogni chip ha 8 bit di uscita, quindi ne avrò 4 attivi per ottenere il parallelismo a 32 bit. Vi sono 4 segnali di RAS, ognuno dei quali andrà in un banco; il segnale *CPU cycle* è legato all'*address strobe* ed indica l'inizio di un ciclo di memoria.

BEi sono i segnali di abilitazione per i 4 byte, uno per ogni byte che si vuole attivare.

Se il parallelismo di ogni chip fosse 1bit sarebbero necessari 32 chip per avere i 4byte su ogni banco, i segnali BEi andrebbero a blocchi di 8 chip.

### Rilevazione e controllo degli errori (slide 25-28)

Nelle memorie ram vi è una certa probabilità che si verifichino degli errori nella fasi di lettura o scrittura dei dati: vi possono essere di tipo errori soft (non permanenti e ripristinabili, come ad esempio le radiazioni: nel caso di accessi alla memoria dopo il termine di tali radiazioni non dovrebbero più verificarsi errori) ed errori di tipo hard (errori non ripristinabili, come ad esempio la rottura di una pista oppure di uno stadio di uscita).

Vi sono due modelli per provare a rilevare e correggere gli errori:

- Rilevazione di parità, consiste nell'aggiungere un bit di parità per ogni byte (oppure parola) presente in memoria; vi è la possibilità che gli errori non vengano rilevati nel caso in cui essi siano a multipli di due sulla stessa quantità protetta da un singolo bit; tale metodo risulta essere debole poiché identifica un errore ma non permette né di localizzarlo né di correggerlo, inoltre visto il crescente aumento delle dimensioni delle memorie ram vi è una crescente probabilità di errori multipli, non è un buon metodo per i sistemi che richiedono un MTBF elevato.
- Codici di correzione e rilevazione degli errori sul parallelismo della parola (la quale può essere anche da 16 e 32 bit), sono detti ECC (Error Correcting Code), rilevano e correggono

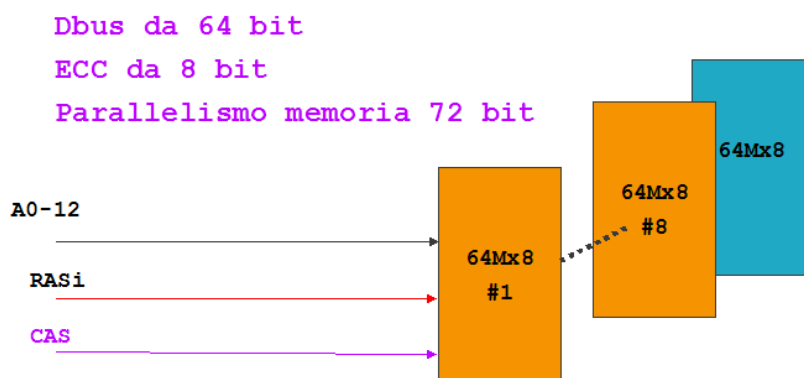
gli errori singoli, rilevano quelli doppi ed alcuni multipli (però per quanto riguarda i doppi non devono essere relativi allo stesso byte).

Nel caso avvenga un errore viene generato un interrupt non mascherabile, il quale blocca la sequenza delle istruzioni in esecuzione in quel momento.

Le unità per il calcolo della parità, sia quelle relative alla fase di lettura sia di scrittura, sono integrate dentro la CPU.

## memoria con ECC

(banco da 512MB con bit di Ecc)



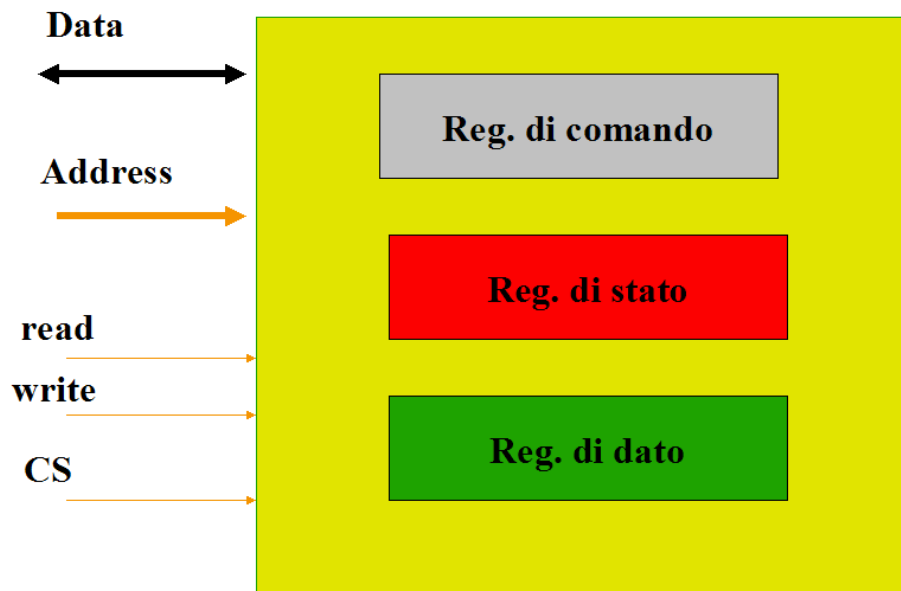
Prof. Marco Mezzalama 08/09

I codici ECC derivano dai codici di Hamming: nel caso di memorie ECC dovrà essere presente un ulteriore chip da 8 bit di parallelismo (supposto che il Dbus sia da 64bit), come sarà necessaria della circuiteria aggiuntiva, nel caso in cui vi siano stati degli errori nei dati, per riscrivere in memoria il valore corretto e quindi per trasmetterlo sul data bus; in generale tutto questo comporta una perdita nelle prestazioni di lettura e scrittura alla memoria.

### **Interfaccia tra dispositivi di I/O e processore (slide sottosistema I/O 1-16)**

La comunicazione tra processore e periferico avviene tramite una interfaccia, la quale si occupa di comunicare con il bus di sistema, virtualizzare la periferica retrostante ed occuparsi della temporizzazione fra l'hardware del sistema e quello della periferia stessa. Tutti i modelli di interfacce possono essere ricondotti ad un unico modello concettuale che prevede la presenza di tre registri principali:

# Il modello



Prof. Marco Mezzalama 08/09

- registro di comando: utilizzato per programmare l'interfaccia con apposite combinazioni di bit i quali specificano il tipo di operazione che la periferica deve intraprendere;
- registro di stato: è memorizzato lo stato dell'interfaccia (e di conseguenza quello della periferica);
- registro di dato: contiene i dati da trasferire (verso la periferica oppure verso il sistema).

L'interfaccia, quindi, è connessa al bus di sistema per poter comunicare con il processore, nello specifico sarà connessa sia all'address bus che al data bus (in particolar modo questo ultimo sarà un bus apposito ben diverso da quello utilizzato per accedere alla memoria RAM, il quale ha un parallelismo tale da ottimizzare il riempimento della memoria cache). Dal punto di vista del programmatore assembler (e del processore) l'interfaccia viene vista come un dispositivo contenente registri su cui andare a leggere o scrivere. La gestione di tali registri può avvenire secondo le modalità di gestione tradizionali della memoria (stesso spazio di indirizzamento, medesimi segnali di controllo e stesse istruzioni) oppure può avvenire tramite uno spazio di indirizzamento riservato, istruzioni speciali e appositi segnali di controllo. Nel primo caso si parla di memory mapped, nel secondo caso di isolated I/O.

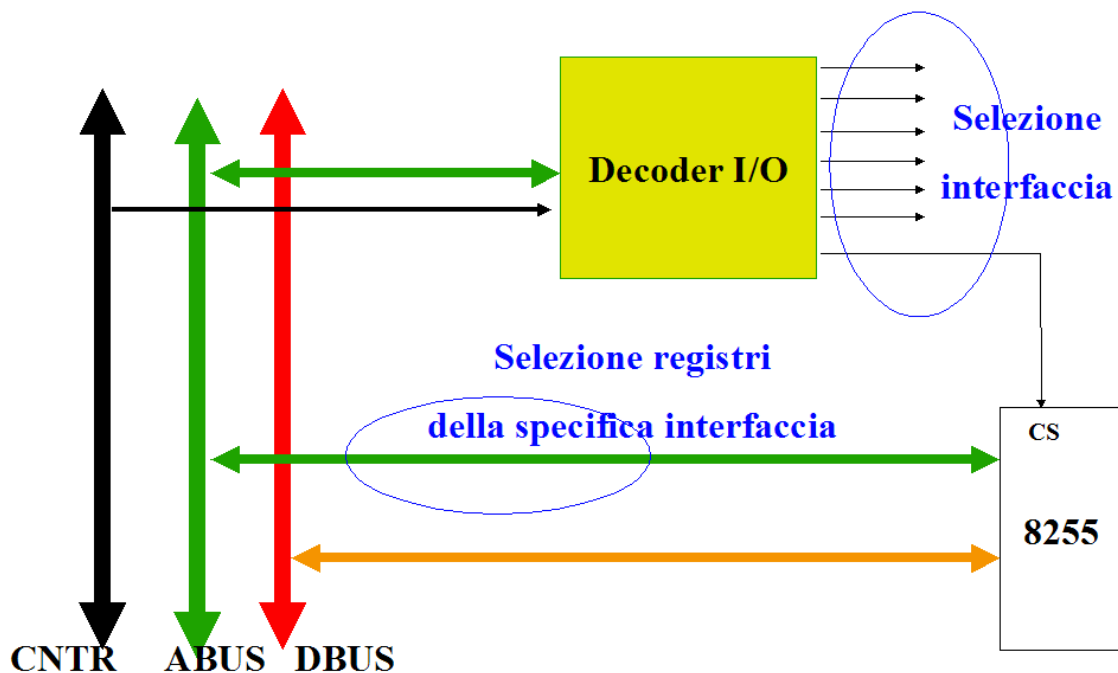
La comunicazione interfaccia-processore avviene tramite un decoder. Tramite tale decoder, è infatti possibile selezionare l'interfaccia con cui comunicare (nell'es. interfaccia parallela 8255).

La connessione dell'interfaccia al decoder è illustrata nella slide seguente.

Il decoder è costituito da 3 bit di input (A,B,C) e da 8 bit di output (O0-O7). I segnali C1, C2, C3 servono per l'abilitazione del decoder.

Nell'esempio è collegato un 8255 (interfaccia di una connessione parallela) costituito da 3 registri di dato e 1 registro di controllo. L'8255 è dotato di 2 pin (A1 e A2) per indirizzare i 4 registri, il pin CS (*chip selected*) attiva il chip e i pin RD e WR per leggere e scrivere nei registri.

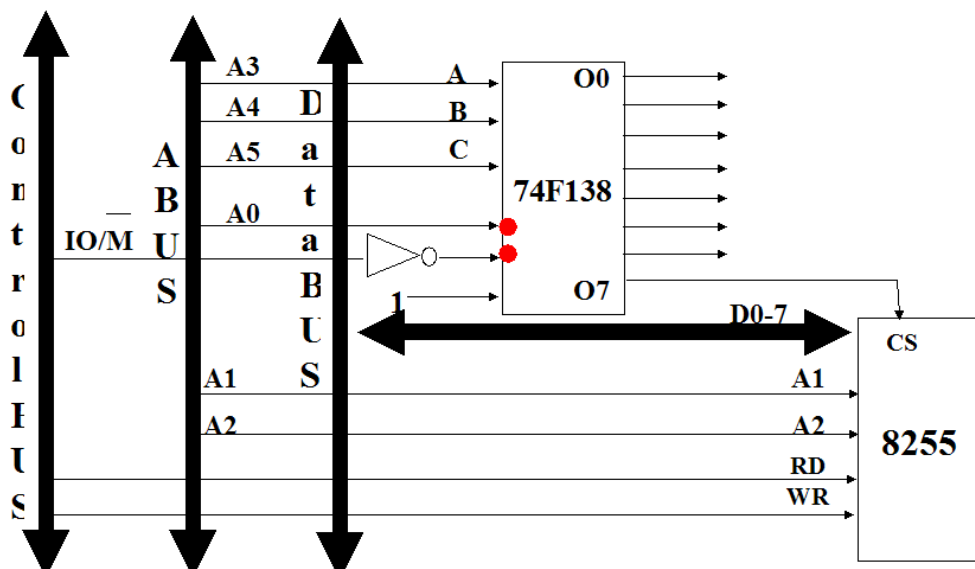
# Connessione I/O



Prof. Marco Mezzalama 08/09

Viene selezionata prima l'interfaccia e poi i registri specifici, quando viene creata l'architettura bisogna tenerne conto per l'indirizzamento (e per il suo dimensionamento); selezionare l'interfaccia corrisponde alla selezione del banco, poi dall' address bus viene ricavato qual è il registro di interesse.

# Connessione Isolated I/O



Prof. Marco Mezzalama 08/09

Quindi ricevuto il segnale CS, l'8255 viene attivato, in base ai valori di A1 A2 si accede al registro desiderato e in base ai segnali di RD e WR viene stabilita l'operazione da attuare sul registro selezionato.

### **Gestione di I/O: Interrupt (slide 1-6)**

Vi sono in generale 3 modalità per gestire le richieste di servizio delle periferiche:

- polling;
- interrupt
- DMA (solo per i periferici a blocchi, ad esempio la memoria ram oppure l'hard disk).

Nei periferici a blocchi oltre il tempo di accesso devo considerare anche il tempo di trasferimento, in quelli a carattere è rilevante solo il primo.

Vi sono diverse modalità di gestione degli interrupt inviati dalle interfacce (essi non sono inviati dalle periferiche, l'interrupt è un segnale architetturale interno e l'architettura è nascosta alla periferica dall'interfaccia). L'interesse verrà posto nel caso della modalità di gestione centralizzata degli interrupt, la quale risulta essere la più efficiente: vi è un dispositivo (l'interrupt controller) a cui arrivato tutti i segnali di interrupt dalle varie periferiche e si occuperà di gestire la priorità, l'abilitazione ed la comunicazione con la CPU.

