

Virtual Memory

- La dimensione della Mem. fisica \leq address space size
- Un programma di grosse dimensioni, o più programmi in esecuzione concorrente, non possono essere contemporaneamente in memoria

Utilizzare il secondary storage (e.g., magnetic disk) per contenere le porzioni di programma non in esecuzione

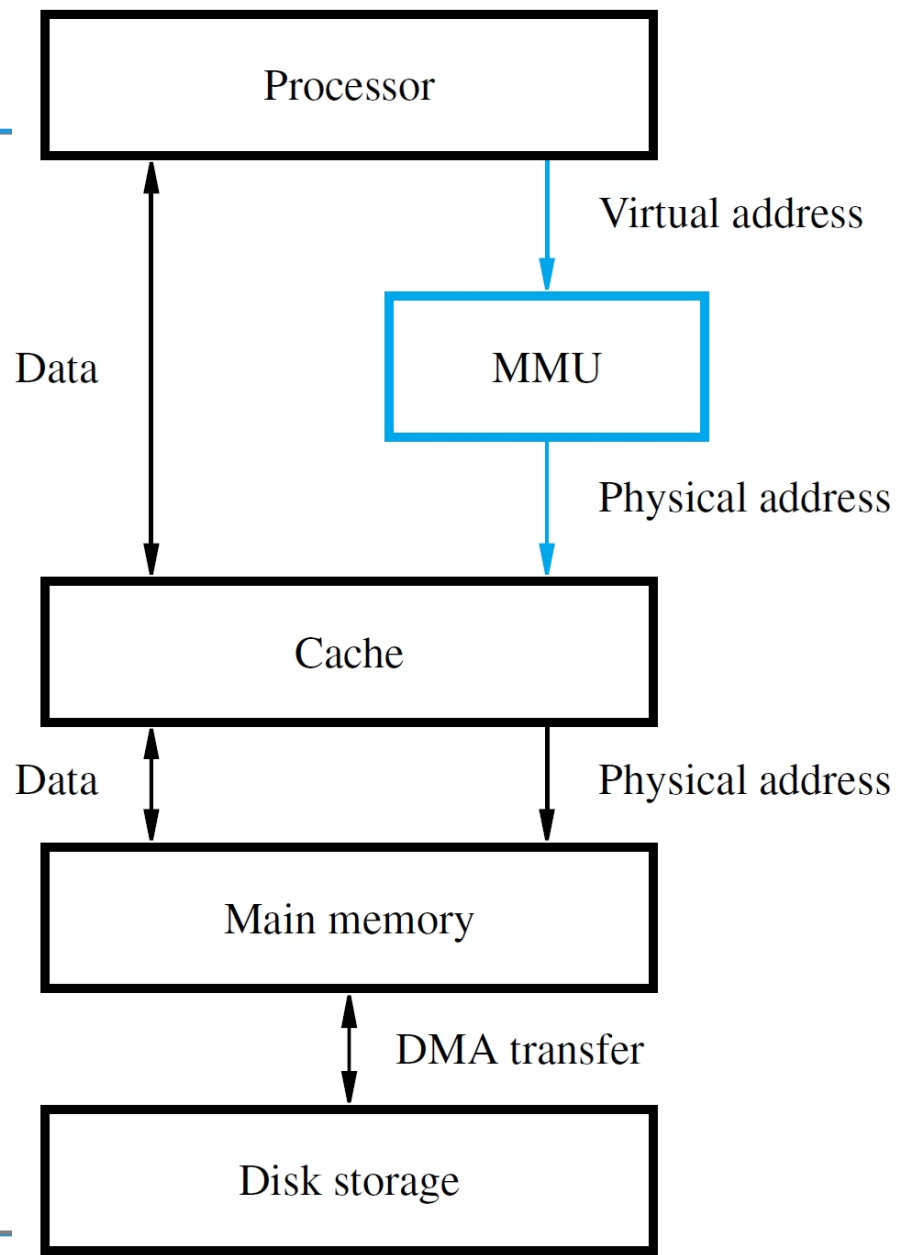
- Quando necessarie, le porzioni necessarie saranno *automaticamente* caricate in memoria per rimpiazzare quelle non più utili
- Il tutto deve essere trasparente al programmatore:
- **La memoria virtuale** deve nascondere la gestione di questi meccanismi

Virtual Memory

- I programmi saranno scritti assumendo che tutto lo spazio logico sia disponibile
- Il Processore genera indirizzi *virtuali* o *logici*
- Questi indirizzi devono essere tradotti in *physical address*
- Se l'indirizzo fa riferimento a contenuti già in memoria, si procede con un normale accesso alla memoria
- Quando l'indirizzo fa riferimento a contenuti non in memoria, bisogna intraprendere le azioni necessarie al caricamento dei dati

Memory Management Unit

- Il meccanismo della memoria virtuale è gestito da una *memory management unit (MMU)*
- Preserva il mapping tra virtual→physical address
- Se l'indirizzo fisico generato non è nella RAM, la MMU richiede l'intervento del SO
 - Trasferimento del blocco di dati attraverso l'uso di DMA
 - Aggiornamento dei dati di mapping della MMU

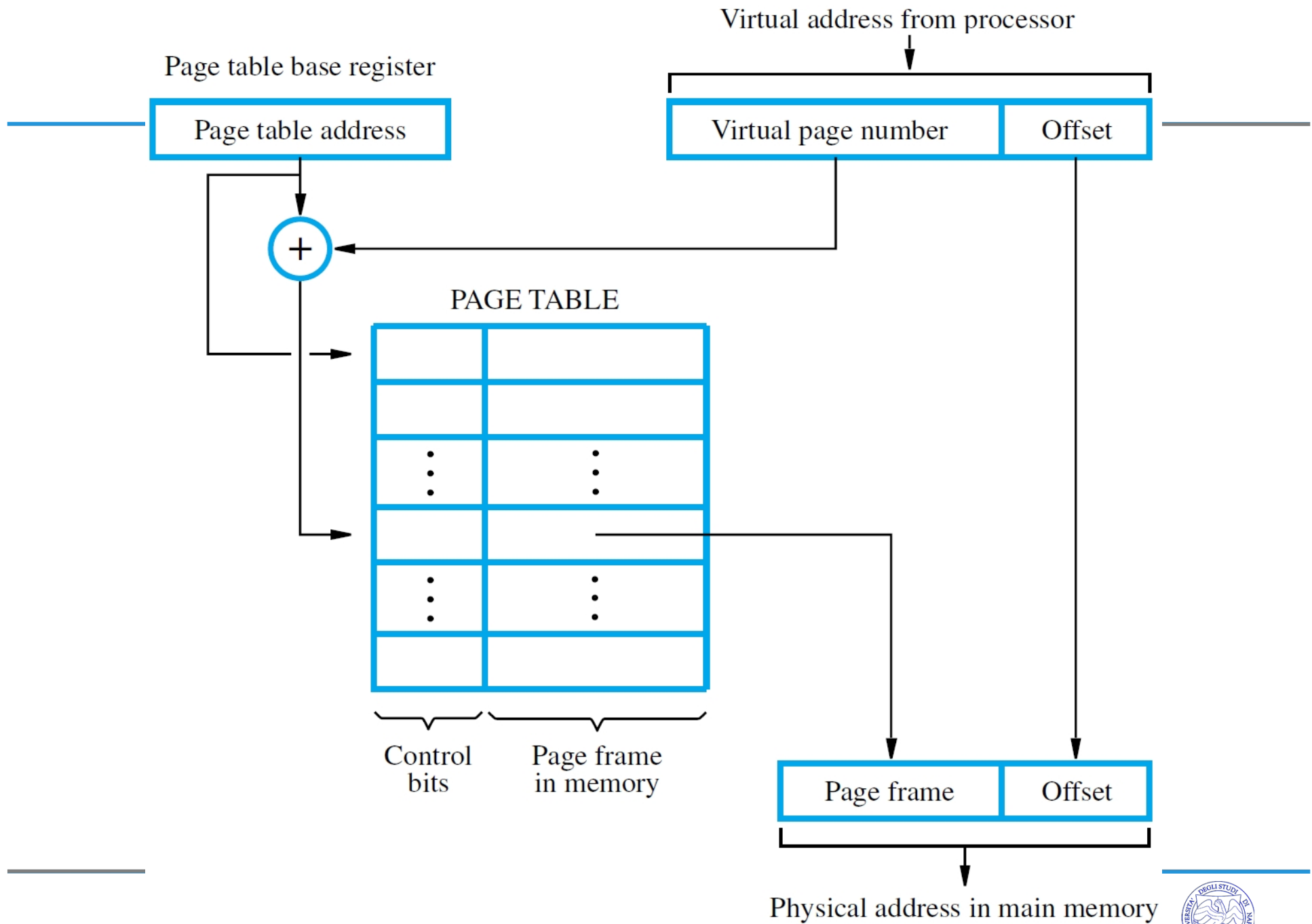


Address Translation

- Uso di unità a dimensione fissa *pages* (2K-16K bytes)
 - Dimensioni maggiori rispetto ai blocchi della cache per compensare la lentezza dei dischi
- Per la traduzione dell'indirizzo:
 - Dividere l'indirizzo in due parti
 - Bit bassi di *offset* per identificare la word nella pagina
 - Bit alti forniscono il numero della pagina: *virtual page number (VPN)*
- Traduzione: preserviamo l'offset bits, sostituire i VPN bits con il numero di *frame*
- *Page table* (conservata in RAM) fornisce le informazioni necessarie alla traduzione

Page Table

- MMU deve sapere dove trovare la page table
 - Conservare l'indirizzo nella *Page table base register*
- Sommando il VPN al base register otteniamo l'indirizzo della entry della pagina
- Se la pagina è in memoria, la Page Table fornisce il numero di frame occupato
- Altrimenti, la tabella può indicare la disk location
- Aggiungere dei bit di controllo
 - Bit di pagina non valida, necessità di copy-back
 - Bit per permessi di read/write
 - Bit per permessi di execute

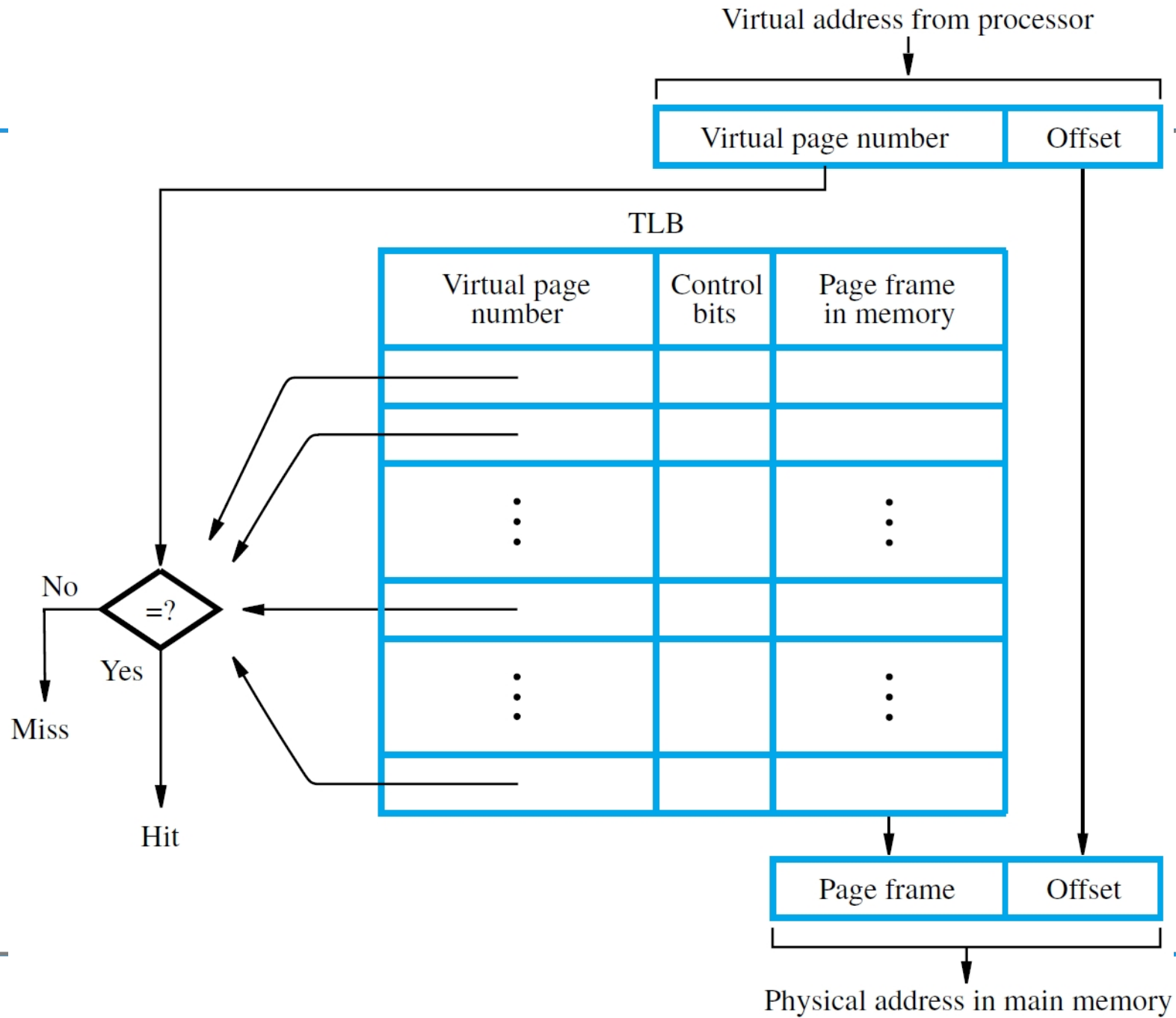


Translation Lookaside Buffer

- MMU interroga la page table ad ogni indirizzo logico generato
- Per grandi memorie fisiche la MMU non può conservare l'intera page table

Translation lookaside buffer (TLB) è un buffer interno alla MMU che conserva una porzione della page table che include le pagine accedute più recentemente

- Ricerche associative nella TLB per identificare dei match
 - In caso di miss accedere ed aggiornare alla TLB intera



Page Faults

- *page fault*: l'indirizzo non corrisponde ad una entry
- MMU solleva un'interruzione per far caricare la pagina -> Sistema Operativo
- Il SO seleziona una posizione da liberare (se necessario) mediante LRU, e, se necessario, effettua il copy-back della pagina
- Il caricamento della pagina può richiedere un tempo lungo, essendo coinvolto il disco => schedulare un altro programma da mandare in esecuzione
- Il programma sospeso può riprendere dopo il caricamento o al prossimo turno