Operating Systems

BITS Pilani
K K Birla Goa Campus

Dr. Lucy J. Gudino
Dept. of CS and IS

Last Class

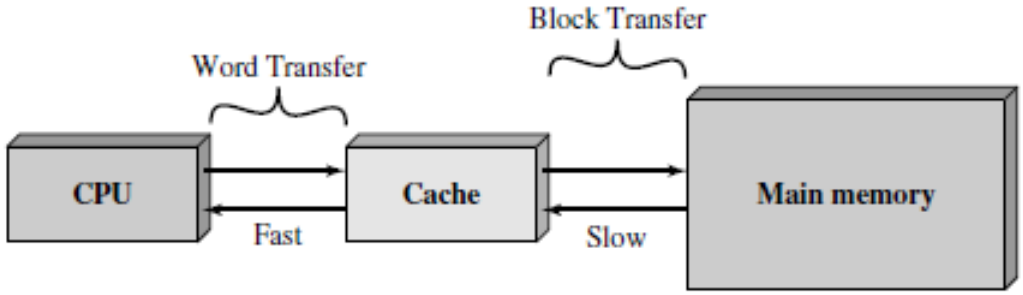
- Paging
  - Address Translation Scheme
  - Address translation Architecture : PD ➔ FD
- Hardware implementation of page table
  - Registers :
    - efficient paging – address translation
  - Page table in main memory
    - Page table base register(PTBR) and Page-table length register (PRLR)
    - Changing page tables requires changing only PTBR register, substantially reducing context-switch time.
    - Time consuming : 2 memory accesses

Contd…

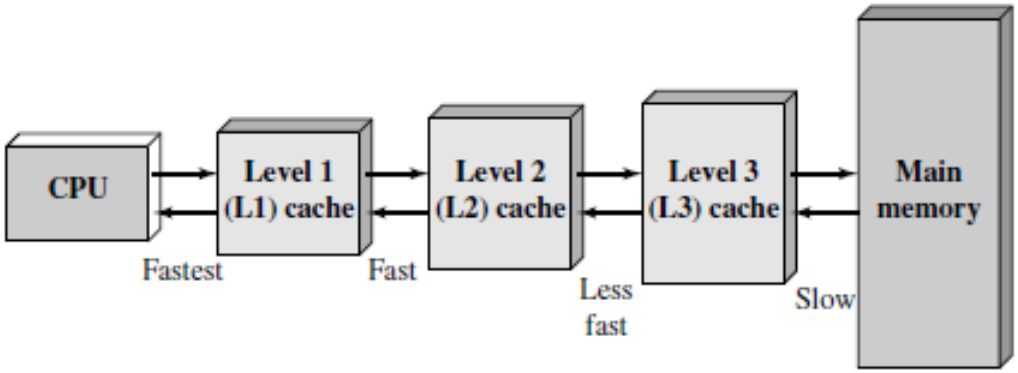
- Solution : use a special, small, fast lookup hardware cache, called a translation look-aside buffer (TLB)
- Associative memory

Cache

- Small amount of fast memory
- Sits between normal main memory and CPU
- May be located on CPU chip or module

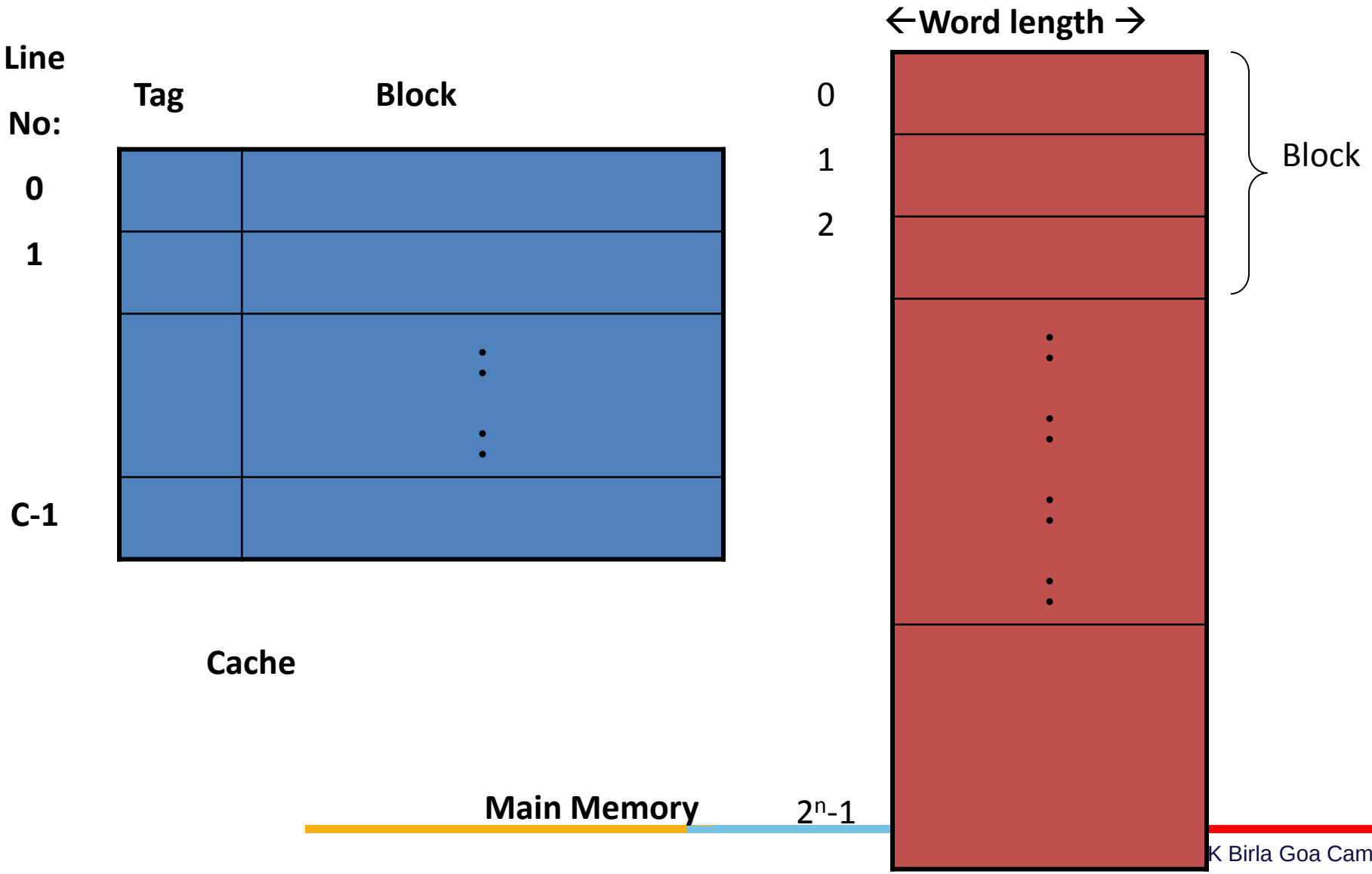


(a) Single cache



(b) Three-level cache organization

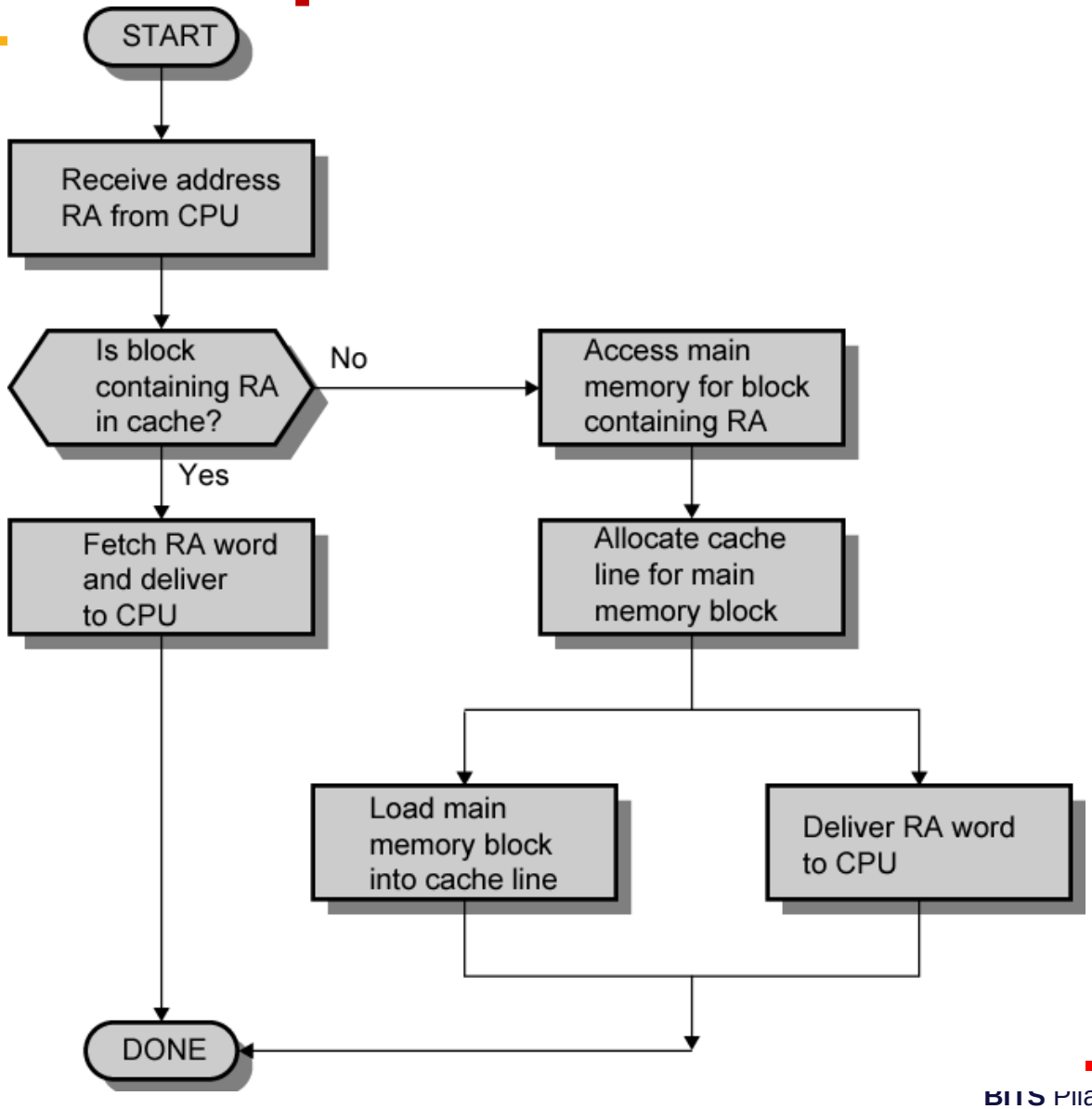
Cache/Main Memory Structure



Cache operation – overview

- CPU requests contents of memory location
- Check cache for this data
- If present, get from cache (fast)
- If not present, read required block from main memory to cache
- Then deliver from cache to CPU
- Cache includes **tags** to identify which block of main memory is in each cache slot

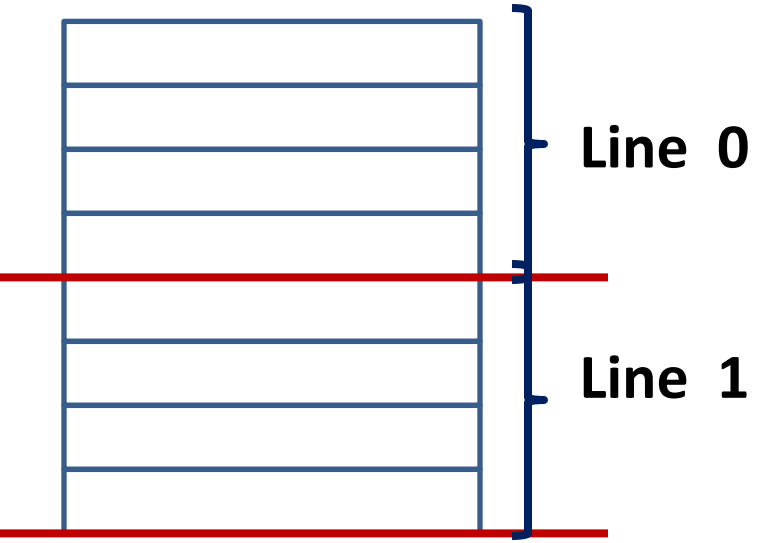
Cache Read Operation - Flowchart



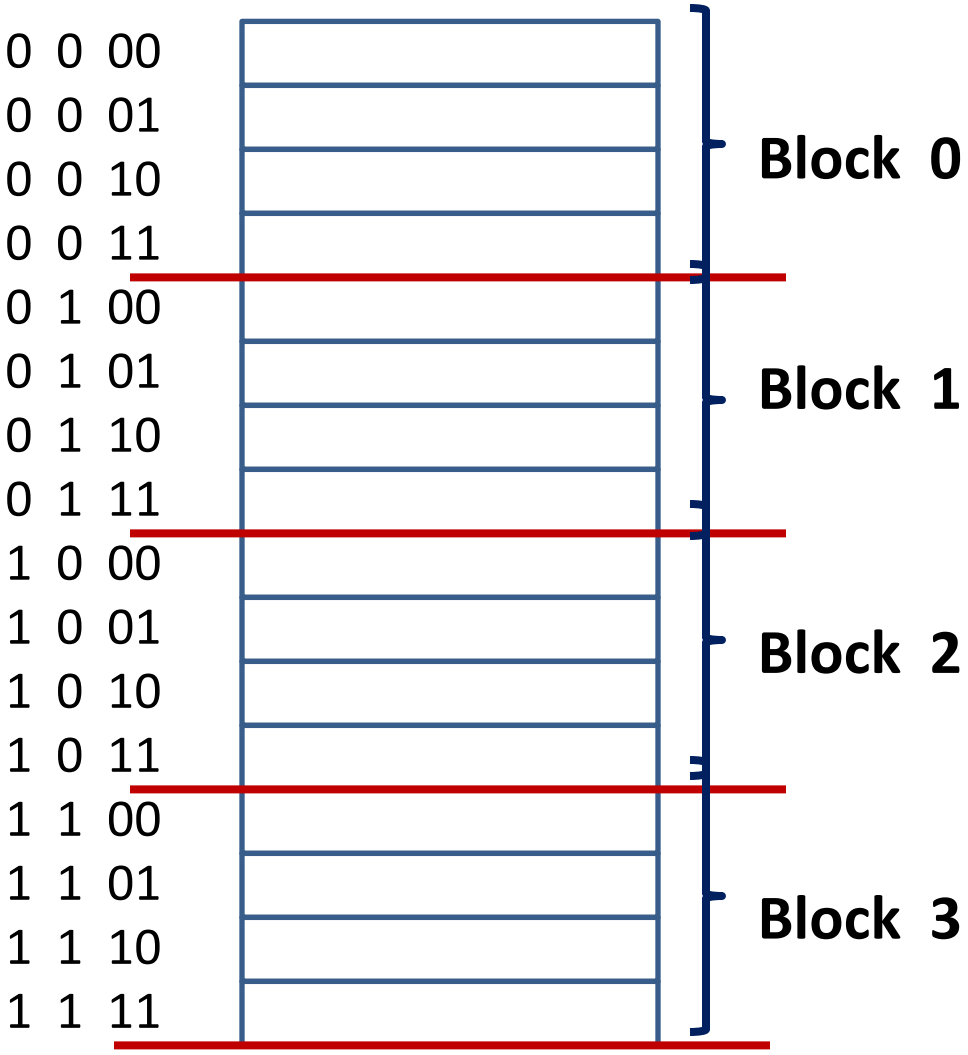
Example

- 16 Bytes main memory
  - How many address bits are required?
- Memory block size is 4 bytes
- Cache of 8 Byte
  - How many cache lines?
  - cache is 2 lines of 4 bytes each

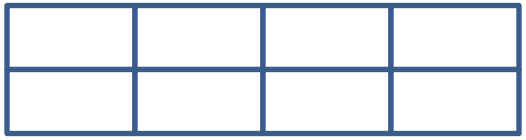
Cache Memory



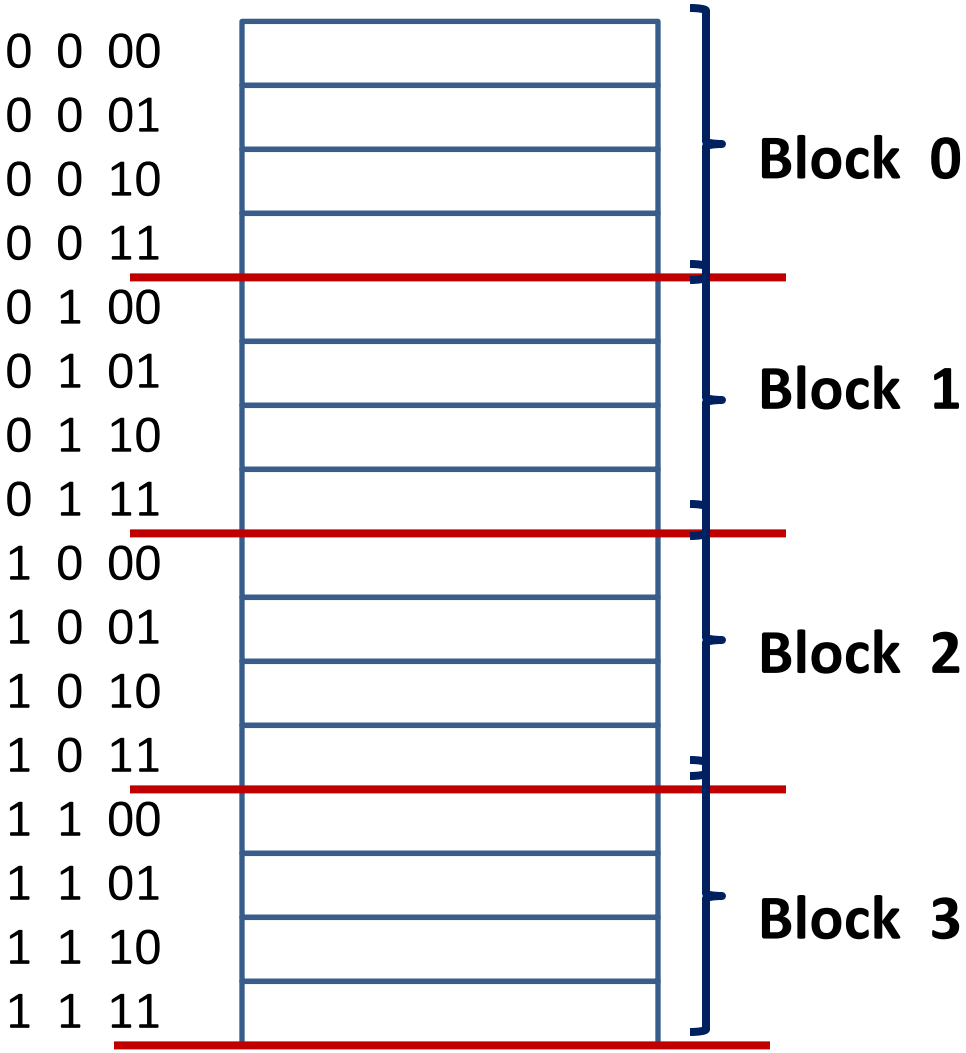
Main Memory



Cache Memory



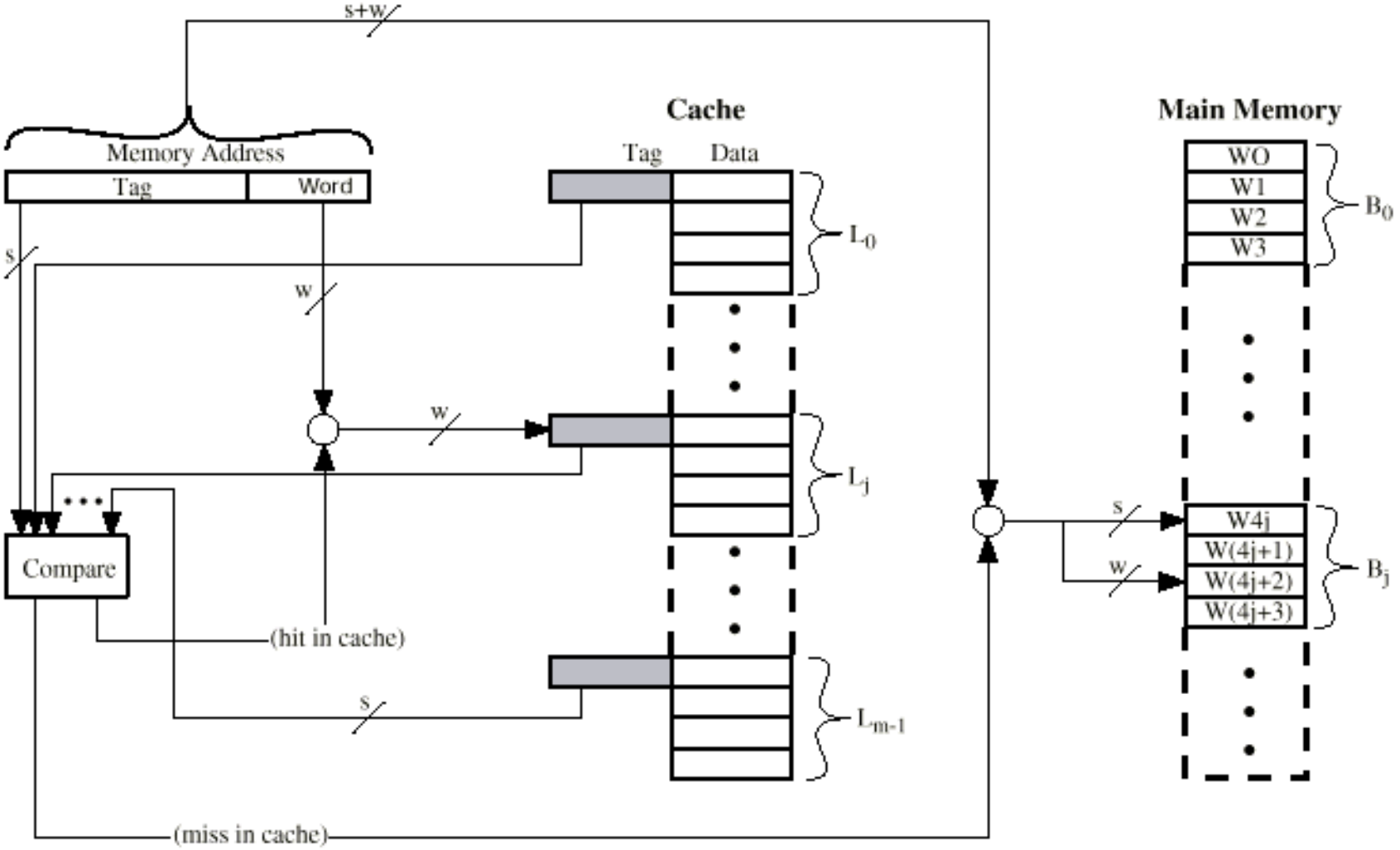
Main Memory



Associative Cache

- A main memory block can load into any line of cache
- Memory address is interpreted as tag and word
- Tag uniquely identifies block of memory
- Every line's tag is examined for a match
- Cache searching gets expensive

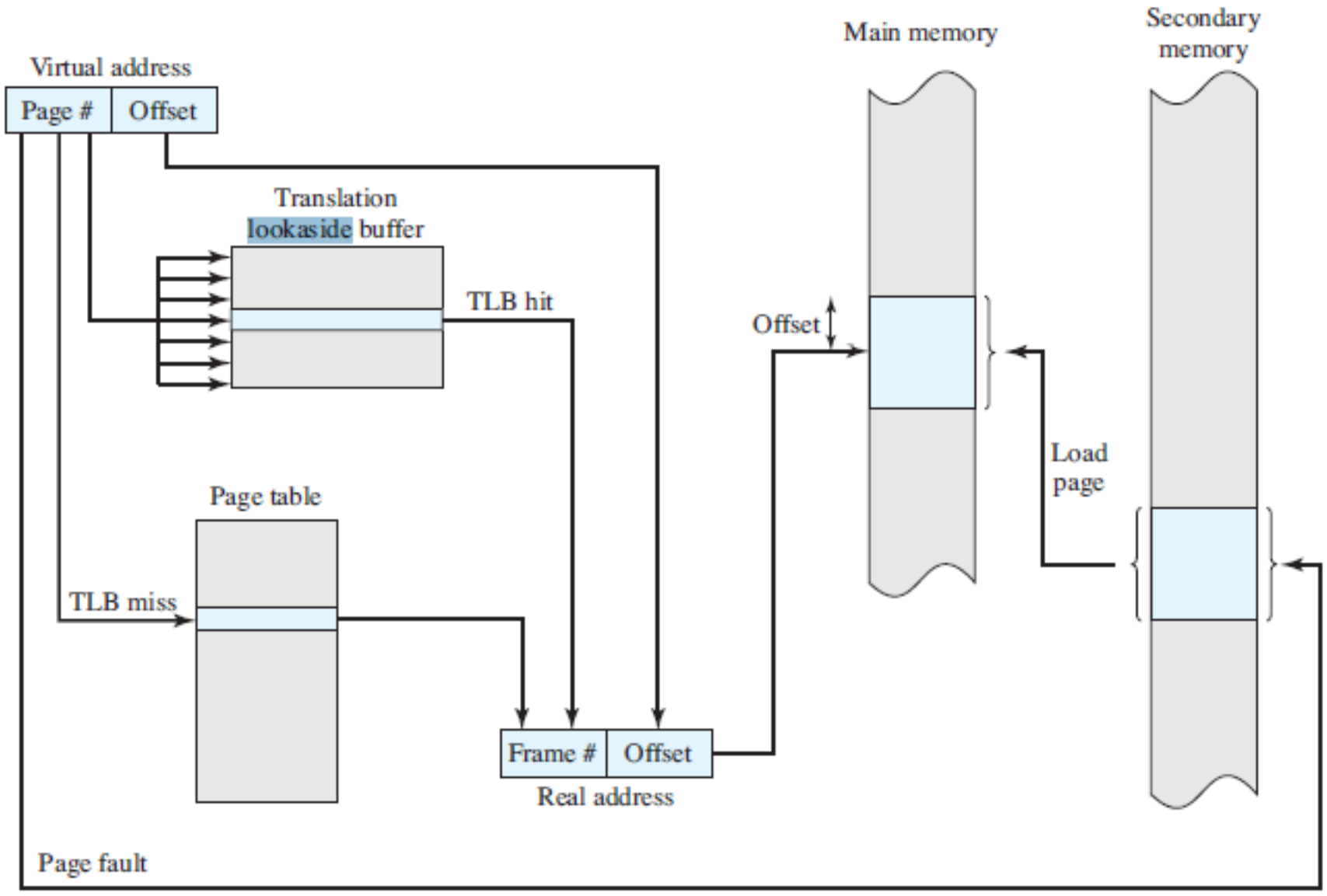
Fully Associative Cache Organization

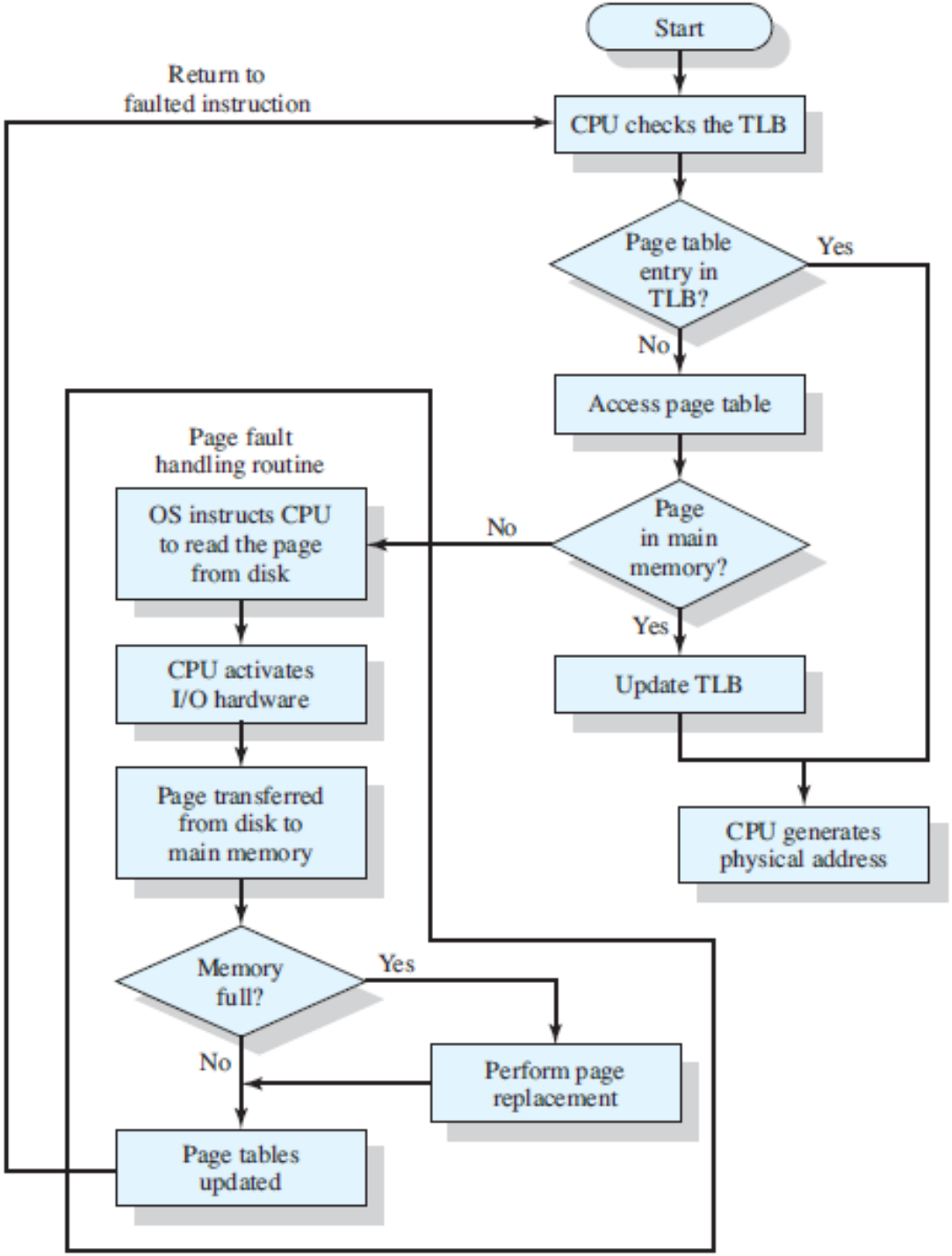


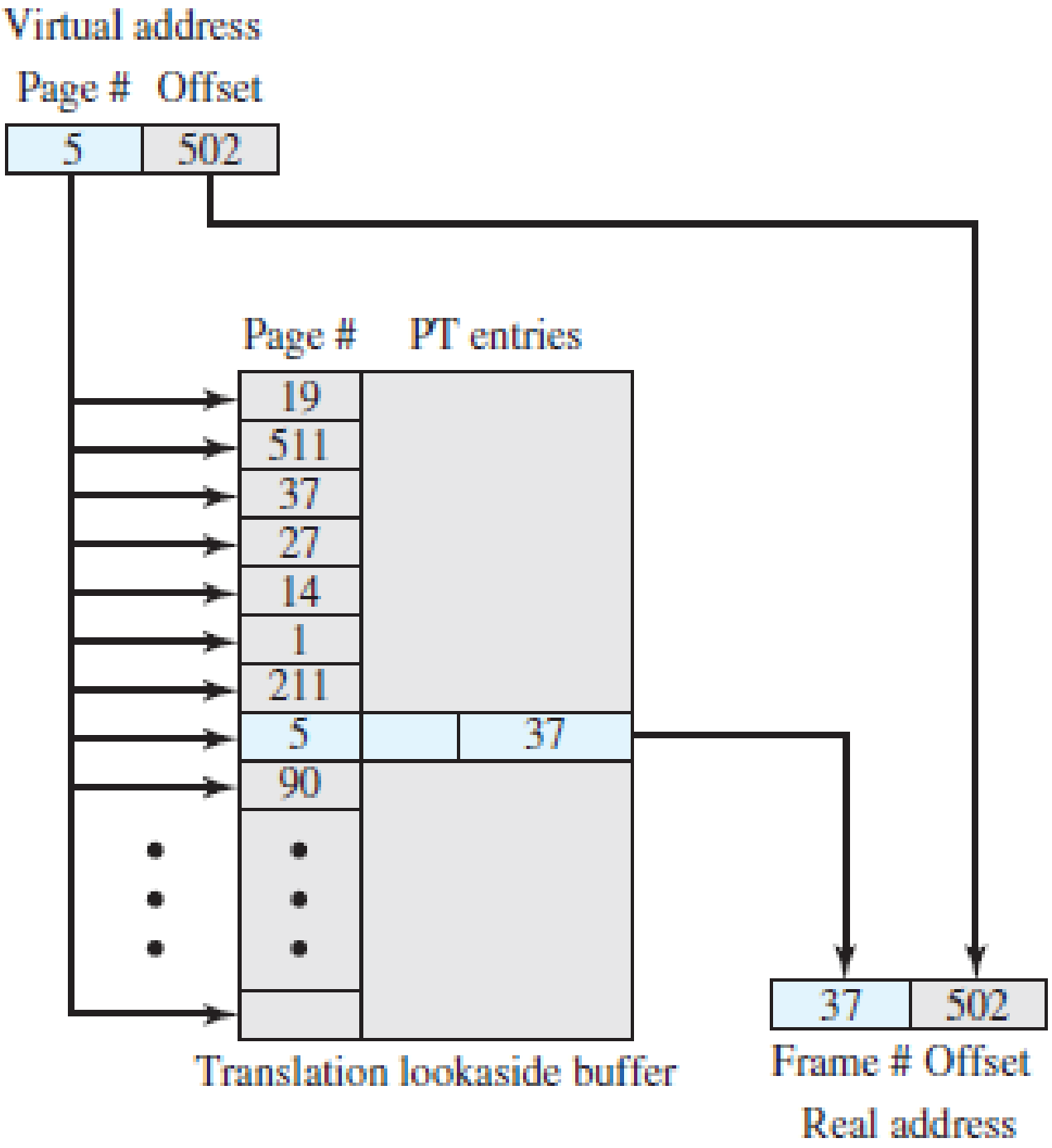
Translation Look Ahead Buffer (TLB)

- Uses a special fast-lookup hardware associative cache
- Each entry in the TLB consists of two parts: a key (or tag) and a value
- Contains those page table entries that have been most recently used (Principle of locality of reference)
- Faster but expensive
- The number of entries in a TLB is small, often numbering between 64 and 1,024.
- TLB Hit and Miss
- Replacement policy : FIFO, LRU, LFU and Random

Paging Hardware With TLB




Start
Return to faulted instruction
CPU checks the TLB
Page table entry in TLB?
Yes
No
Access page table
Page fault handling routine
OS instructs CPU to read the page from disk
No
Page in main memory?
Yes
CPU activates I/O hardware
Update TLB
Page transferred from disk to main memory
CPU generates physical address
Memory full?
Yes
No
Perform page replacement
Page tables updated


Virtual address
Page # Offset
5
502
Page # PT entries
19
511
37
27
14
1
211
5
37
90
37
502
Translation lookaside buffer
Frame # Offset
Real address

Contd…

- Entries can be wired down, meaning that they cannot be removed from the TLB
  - kernel code
- Address-Space IDentifiers (ASIDs) in each TLB entry
  - To uniquely identify each process
  - Provide address space protection

Example

- Hit ratio :The percentage of times that a particular page number is found in the TLB
- Hit ratio 80%
- Access time for TLB : 20 nano sec.
- Access time for main memory 100 nano seconds
- Access time without TLB 200 nano seconds
- Access time when TLB Hit : 120 nano seconds
- Access time when TLB miss 20+100+100 = 220 nano sec
- Effective access time = 0.80 x 120 + 0.20 x 220

  = 140 nanoseconds.