Lecture No 6

Virtual to Real Mapping

Addressing & Memory Contd..)

Virtual to Real Mapping

- **A user programmer uses 32 bit virtual addresses.**
- **Depending on physical memory size available, these virtual addresses need to be mapped to real memory addresses.**
- **Each user has an ID, given to it by the system which acts as an overall base for that users address space.**
- **The user ID defines a base register, pointed to by the PSW, which defines the starting point of segment table belonging to this particular user**
- **Most significant or upper 12 bits of users 32 bit virtual address define the segment number. So addition of first 12 bits to 32 bit base address gives an entry in segment table that is contained in memory.**
- **This segment table entry contains a base address and a bound for the particular segment identified by the virtual address.**
- **Since 32 bit gives 4 GB of Virtual address space, Upper 12 bits give 4096 user segments of 1 MB each.**

Addressing & Memory Contd..)

Virtual to Real Mapping

- Each segment (represented by lower 20 bits of user virtual address) is further divided in pages.
- There are 256 pages (represented by upper 8 bits of these 20 bits) of size 4096 bytes each.

32 bit

Segment number (12 bits)	Page Number (8 Bits)	Byte off set in Page (12 Bits)

32 Bit User Virtual address

•Real memory is divided into *page frames* which are the same size as the virtual pages (4096 Bytes)

•When a page is needed during the running of a program, it is copied into a page frame in real memory.

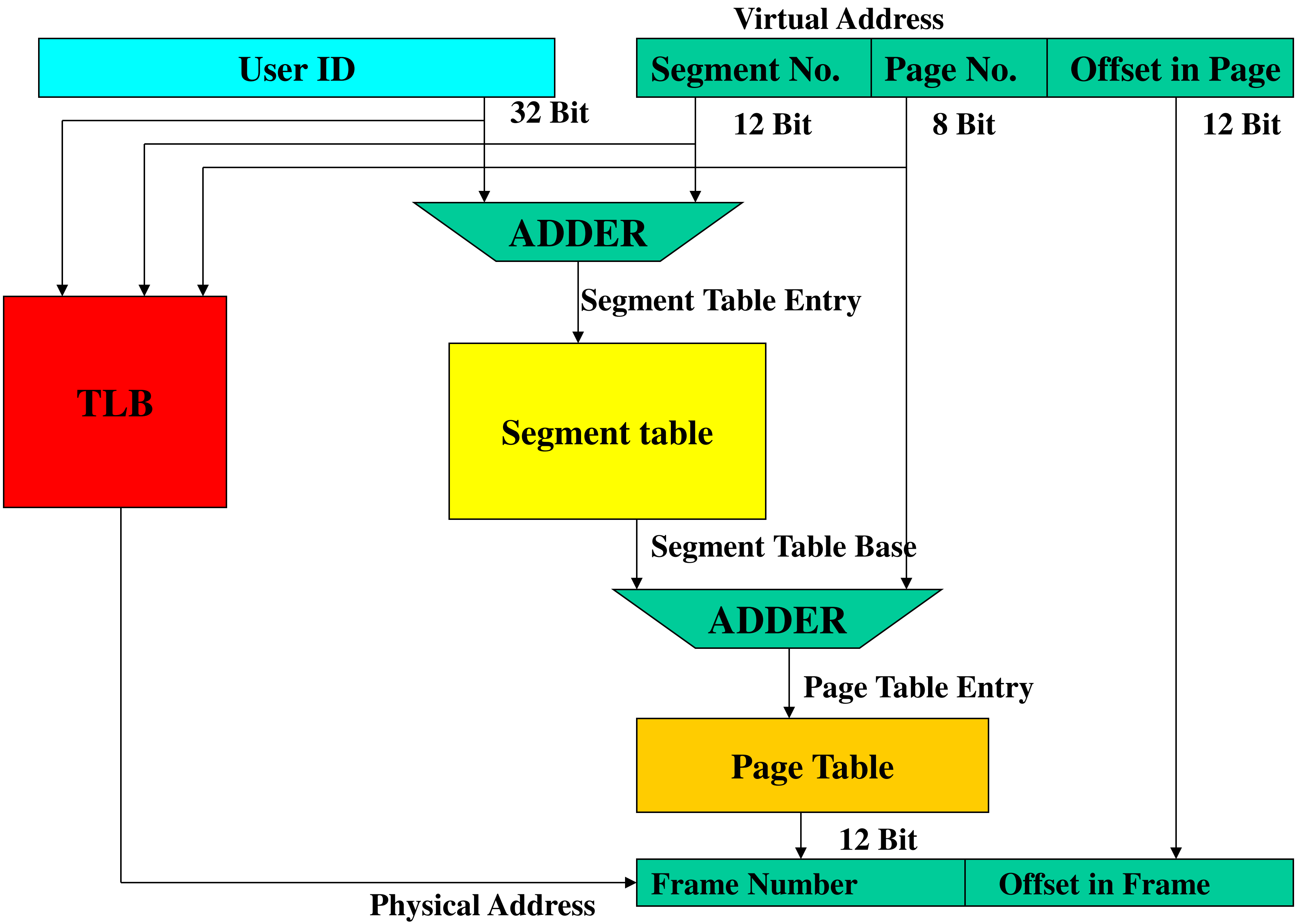
•The process of moving program pages to and from real memory is called *paging*.

Addressing & Memory Contd..)

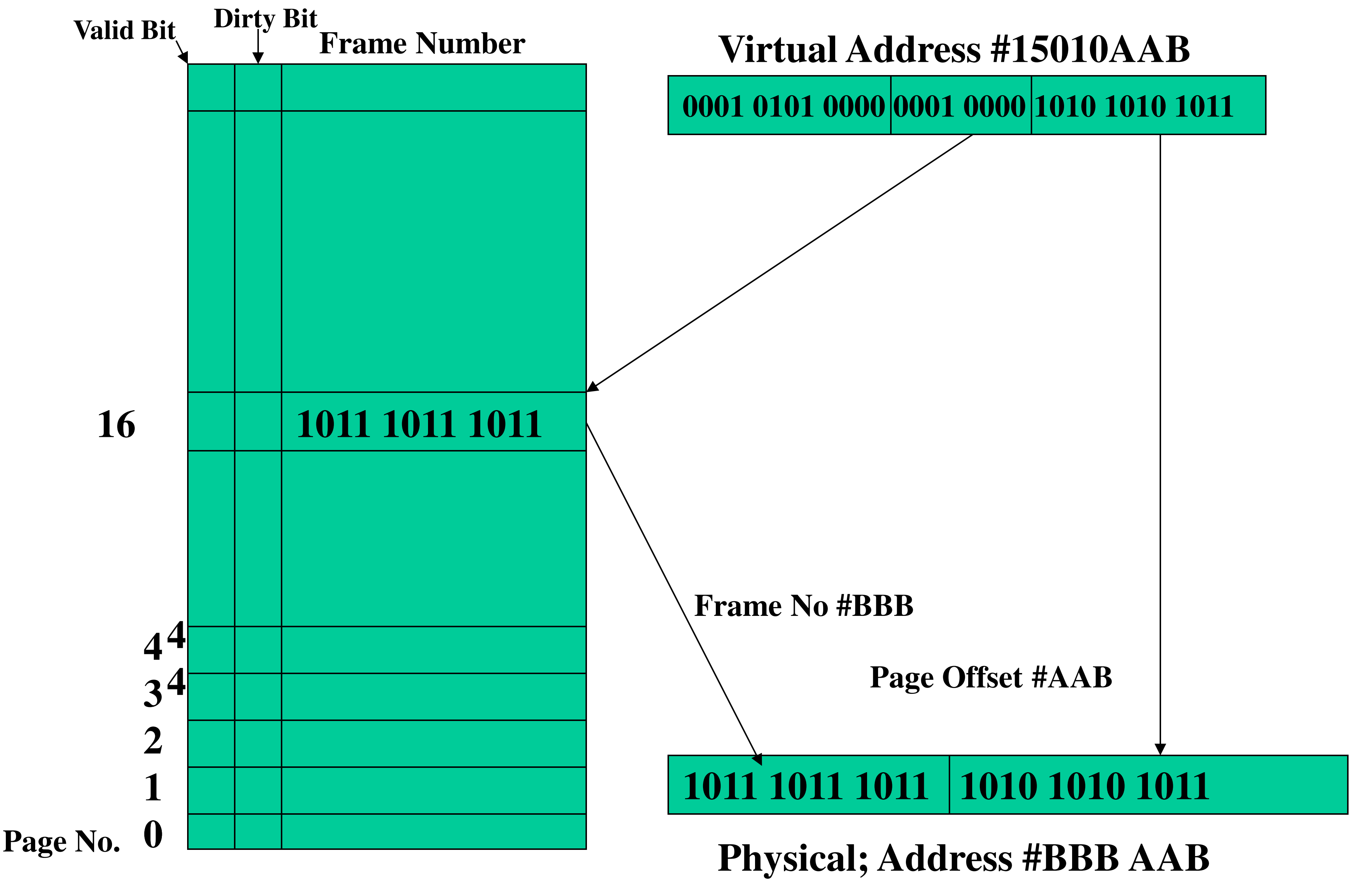
Virtual to Real Mapping

- **Any page can go into any page frame.**
- **The memory management process translates a 32-bit virtual address into a 24-bit physical address. (16 MB Real Memory)**
- **This is done with the aid of a *page table*.**
- **The segment table base plus the 8 bit page offset of a page with in a segment defines an entry into page table associated with that particular segment.**
- **Each entry contains a valid bit that indicates whether the page is currently in main memory, a dirty bit indicating whether the page has been modified, and a frame number pointing to a page frame in real memory.**
- **Since there are 4,096 4 K frames in a real memory of 16 M cells, the frame number in our example page table will be 12 bits.**
- **Since the pages and page frames are the same size, the offset from the virtual address can simply be copied into the offset part of the physical address.**

Addressing & Memory Contd..)



Addressing & Memory Contd..)



Addressing & Memory Contd..)

Virtual to Real Mapping

Example:

Segment table base (located in real memory) at #100000

User ID #000012

User segment table will start at #100012

User program specifies a virtual address #15010AAB

To get the segment table entry (Real Address) #100012 + #150 (Segment No.)

This entry will specify base address for page table #A00111

To get the page table entry (Real Address) #A00111+#10 (Page No.)

Frame no contained in this entry (and valid bit Set) #00000BBB

The real Address for virtual address #150 10 AAB #BBBAAB

Addressing & Memory Contd..)

Virtual to Real Mapping

- **The overall process of accessing User ID, Segment Table and Page table and doing appropriate calculation is a time consuming process (20-30 Cycles)**
- **A mechanism called Translation Lookaside Buffer (TLB) is used to cache the translations done earlier.**
- **The used ID, Virtual Segment information and Virtual Page information is used to access TLB and if entry is found there (Translation done earlier) the Physical address bits are available immediately (1 Cycle).**
- **If the desired page is not in memory, A *page fault* condition is generated and Operating system is interrupted to load the desired page from disc to any available frame in main memory.**
- **The page table entry is updated with address of the frame (Where page is loaded) and valid bit is set to indicate availability of page in main memory**
- **The dirty bit is set if any modifications have been done in the page so that page in back up can be updated when this entry is removed.**

Addressing & Memory Contd..)

Virtual to Real Mapping

- **The process of reading pages in from disk only when they are needed is called *demand paging*. Pages are not loaded into page frames until there is a demand for them.**
- **A program typically starts with none of its pages in real memory. The first reference causes a page fault. Soon, however, all the pages needed for a given part of the program are in memory. This set of pages is called the *working set.***
- **As long as the working set of a program is smaller than the available physical memory, the program runs nearly as fast as it would if it had free access to enough real memory for the entire program.**
- **If there is not enough real memory to hold the working set, page faults occur frequently and the CPU spends more time moving pages around than it does running the program.**
- **When a page fault occurs and there's no free page frame, the operating system must make room for the new page by replacing a page already in main memory.**

Addressing & Memory Contd..)

Virtual to Real Mapping

- **LRU (least recently used) or FIFO, (first-in first-out.) can be used as a replacement policy .**
- **The advantage of FIFO is that bookkeeping only has to happen when a new page is loaded, and not every time a page is referenced.**