Paper Name : Operating System
Prepared by : Debanjali Jana

Translation Lookaside Buffer (TLB) in Paging :

A Translation look aside buffer can be defined as a high-speed memory cache which can be used to reduce the time taken to access the page table again and again. It is a memory cache which is closer to the CPU and the time taken by CPU to access TLB is lesser then that taken to access main memory. TLB follows the concept of locality of reference which means that it contains only the entries of those many pages that are frequently accessed by the CPU. It is a solution that tries to reduce the effective access time. Being a hardware, the access time of TLB is very less as compared to the main memory.
Each entry in the TLB consists of two parts: a key (or tag) and a value. When the associative memory is presented with an item, the item is compared with all keys simultaneously. If the item is found, the corresponding value field is returned. The search is fast; the hardware, however, is expensive. Typically, the number of entries in a TLB is small, often numbering between 64 and 1,024. The TLB is used with page tables in the following way. The TLB contains only a few of the page-table entries. When a logical address is generated by the CPU, its page number is presented to the TLB. If the page number is found (**TLB hit**), its frame number is immediately available and is used to access memory. If the page number is not in the TLB (known as a **TLB miss)**. A memory reference to the page table must be made. When the frame number is obtained, we can use it to access memory (Figure 8.11). In addition, we add the page number and frame number to the TLB, so that they will be found quickly on the next reference. If the TLB is already full of entries, the operating system must select one for replacement. Replacement policies range from least recently used (LRU) to random.
Whenever context switching occurs, the entire content of TLB is flushed and deleted. TLB is then again updated with the currently running process.

When a new process gets scheduled-

- Initially, TLB is empty. So, TLB misses are frequent.
- With every access from the page table, TLB is updated.
- After some time, TLB hits increases and TLB misses reduces.

The percentage of times that a particular page number is found in the TLB is called the **hit ratio**. An 80-percent hit ratio, for example, means that we find the desired page number in the TLB 80 percent of the time. If it takes 20 nanoseconds to search the TLB and 100 nanoseconds to access memory, then a mapped-memory access takes 120 nanoseconds when the page number is in the TLB. If we fail to find the page number in the TLB (20 nanoseconds), then we must first access memory for the page table and frame number (100 nanoseconds) and then access the desired byte in memory (100 nanoseconds), for a total of 220 nanoseconds. To find the effective memory access time we weight the case by its probability:

effective access time = 0.80 x 120 + 0.20 x 220
= 140 nanoseconds
