Assignment-5

1) A CPU generates 32-bit virtual address. The page size is 4 KB. The processor has a TLB which can hold of total of 128 page table entries and is 4 way-set associative. Find minimum size of TLB tag?

Soln:-

Size of Page = 4 KB = 2^12

Total no. of bits needed to address a page frame = 32-12 = 20

Definition:-
If there are 'n' cache lines in a set, the cache replacement is called n-way set associative.

Now,

∵ TLB is 4 way set associative & can hold total 128 (2^7) page table entries,

no. of sets in cache = $\frac{2^7}{(2^2)} = 2^5$. So 5 bits are needed to address a set, and

15 (20-5) bits needed for tag.

Any 15 bits.

2) A compiler uses 46-bit virtual address, 32-bit physical address, & a three-level paged page table organization. The page table base register stores the base address of first-level table (T1), which occupies exactly one page. Each entry of T1 stores the base address of a page of second level table (T2). Each entry of T2 stores base address of 3rd-level table (T3). Each entry of T3 stores a page table entry (PTE). The PTE

is 32 bits in size. The processor used in the computer has a 1 MB 16 way set associative virtually indexed physically tagged cache. The cache block size is 64 bytes. What is size of a page in KB in this computer?

Soln:-

Let page size is of 'x' bits.

Size of T1 = 2^x bytes (bcz T1 occupies one page exactly).

Now, no. of entries in T1 = $\frac{2^x}{4}$.

(bcz each page table entry is 32 bytes or 4 bytes in size)

No of entries in T1 = No of second level page tables.

(bcz each I-level page table entry stores the base address of page of II-level page table)

Total size of 2nd level page tables =

$$\left(\frac{2^x}{4}\right) \times (2^x)$$

Similarly, no. of entries in II-level page tables =

No. of III level page tables =

$$\left(\frac{2^x}{4}\right) \times \left(\frac{2^x}{4}\right).$$

Total size of third level page tables =

$$\left(\frac{2^x}{4}\right) \times \left(\frac{2^x}{4}\right) \times \left(2^x\right).$$

Similarly, total no. of entries (page) in all III-level page tables = $\left(\frac{2^x}{4}\right) \times \left(\frac{2^x}{4}\right) \times \left(\frac{2^x}{4}\right)$

$$= 2^{(3x-6)}.$$

Size of virtual memory (VM) $= 2^{46}$.

$\therefore$ No. of pages in VM $= \frac{2^{46}}{2^x} = 2^{(46-x)}$.

Total no. of the pages in III-level page tables = No. of pages in VM.

$\therefore \Rightarrow 2^{(3x-6)} = 2^{(46-x)}$

$3x - 6 = 46 - x$

$\boxed{x = 13}$.

$\Rightarrow$ Page size of is of 13 bits.

or Page size $= 2^{13}$ bytes $= 8$ KB.

Ans.

3.) Let the page fault service time be 10 ms in a computer with avg. memory access time being 20 ms. If one page fault is generated for every 10^6 memory accesses, what is effective access for the memory?

Soln:- Let P be the page fault rate

Effective memory access time = P * (page fault service time) + (1-P) * (memory access time)

$$\Rightarrow \left(\frac{1}{10^6}\right) \times 10 \times 10^6 \text{ ns} + \left(1 - \left(\frac{1}{10^6}\right)\right) \times 20 \text{ ns}$$

$$\approx 30 \text{ ns (approx.)}.$$

Ans.

4.) A process uses 36-bit Physical addresses and 32 bit virtual addresses, with a page frame size of 4 Kbytes. Each page table entry is of size 4 bytes. A three level page table is used for virtual to physical address translation, where the virtual address is used as follows.

- Bits 30-31 are used to index into the first level page table.
- Bits 21-29 are used to index into 2nd. level

- Bits 12-20 are used to index into 3rd level page table, and
- Bits 0-11 are used as offset within the page.

The no. of bits required for addressing the next level page table (or page frame) in the page table entry of first, second & 3rd level page tables are respectively.

Soln:-

Virtual address size = 32 bits.
Physical address size = 36 bits.
Physical memory size = 2^{36} bits.
Page frame size = 4K bytes = 2^{12} bytes.

No. of bits required to access physical memory frame = 36 - 12 = 24.

So, in 3rd level of page, 24 bits are required to access an entry.

Ans.

9 bits of virtual address are used to access 2nd level page table entry & size of pages in 2nd level is 4 bytes. So, size of 2nd level page table is $(2^9) * 4 = 2^{11}$ bytes.

It means there are $(2^{36} / 2^{11})$ possible locations to store this page table. $\hookrightarrow 2^{25}$

∴ 2nd page table requires 25 bits to address it.

Ans.

Similarly, the ~~page~~ first page needs 25 bits to address it.

Ans.

5.) Consider a paging hardware with a TLB. Assume that the entire page table and all the pages are in the physical memory. It takes 10 msec. to search the TLB & 80 msec. to access the physical memory. If the TLB hit ratio is 0.6, what is the effective memory access time (in msec)?

Soln: Effective memory access time =

TLB hit × (TLB access time + memory access time)
+ TLB miss (TLB access time + page table access time + memory access time)

= 0.6 (10 + 80) + 0.4 (10 + 80 + 80)

= 54 + 68

= 122 msec. Ans.

6) The memory access time is 1 nsec for a read operation with a hit in cache, 5 nsec for a read operation with a miss in cache, 2 nsec for a write operation with a hit in cache & 10 nsec for a write operation with a miss in cache. Execution of a sequence of instructions involves 100 instruction fetch operations, 60 memory operand read operations & 40 memory operand write operations. The cache hit-ratio is 0.9. What is avg MAT (in nsec) in executing the sequence of instructions?

Sol:-

Total no of instructions = 100 + 60 + 40 = 200

Time taken for 100 fetch operations (fetch = read)

$= 100 \times ((0.9 \times 1) + (0.1 \times 5))$

↳ time taken for read when there is cache hit

$= 140$ ns

Time taken for 60 read operations =

$60 * ((0.9 \times 1) + (0.1 \times 5))$

$= 84$ ns.

Time taken for 40 write operations

$= 40 \times ((0.9 \times 2) + (0.1 \times 10))$

$= 112$ ns.

So, the total time taken for 200 operations

$= 140 + 84 + 112 = 336$ nsec.

Avg. time taken = $336/200 = 1.68$ nsec Ans.

7.) Consider a system with a two-level paging scheme in which a regular MAT 150 nsec, and servicing a page fault take 8 msec. An avg. instructions take 100 nsec of CPU time, & 2 memory accesses. The TLB hit ratio is 90%, the page fault rate is one in every 10,000 instructions. What is the effective avg. instruction execution time?

Soln:-

Page fault rate is given 1 page / 10,000 instructions.

∵ There are 2 memory access / instruction, so we need double address translation time for avg. instruction execution time.

There are 2 page table accessed if TLB miss access occured.

⇒ TLB access assumed as 0.

∴ Avg. instruction exec. time = Avg. CPU exec. time + Avg. time for getting data.

⇒ (Avg. CPU exec. time + Avg. address translation time for each instru. + Avg. memory fetch time for each instruction + Avg. page fault time for each instruction.)

$$= 100 + 2 \times \left(0.9 \times (0) + 0.1 \times (2 \times 150) \right) + 2 \times 150 + \frac{1}{10,000} \times 8 \times 10^6$$

$$= 100 + 60 + 300 + 800 = 1260 \text{ ns}.$$ Ans.

8.) A processor uses 2-level page tables for virtual to physical address translation. Page tables for both levels are stored in main memory. Virtual & physical addresses are both 32 bits wide. The memory is byte addressable. For virtual to physical address translation, the 10 most significant bits of the virtual address are used as index into the 1st level page table while the next 10 bits are used as index into the 2nd level page table. The 12 least significant bits of the virtual address are used as offset within the page. Assume that the page table entries in both levels of page tables are 4 bytes wide. Further, the processor has a TLB, with a hit ratio of 96%. The TLB caches recently used virtual page no. and the corresponding physical page no.. The processor also has a physically addressed cache with a hit rate of 90%. Main memory access time is 10 nsec, cache access time is 1 nsec, & TLB access time is also 1 nsec. Assuming that no page faults occur, what is the avg. time taken to access a virtual address?

Sol:-

The possibilities are

TLB Hit * Cache Hit + TLB Hit * Cache Miss +
TLB Miss * Cache Hit + TLB miss *
Cache Miss

$= 0.96 * 0.9 * 2 + 0.96 * 0.1 * 12 +$

$0.04 * 0.9 * 22 + 0.04 * 0.1 * 32$

$= 3.8$ nsec.

≈ 4 nsec.

(When TLB miss occurs it takes 1 ns & for physical address it has to go through 2 level page tables which are in main memory & takes 2 memory access & that page is found in cache taking 1 nsec which gives total of 22 nsec)

∴ Avg. time taken to access a virtual add = 3.8 ≈ 4 nsec.

Ans.