CMPT 300 - Assignment 4 - Part I.

(1) We are given that the size of each page in memory is

$$4096 = 2^{12} \text{ bytes.}$$

Hence, we can infer that the offset in our logical address would be 12 bits. Also, we know that the computer provides its users a virtual-memory space of 2^{32} bytes. Therefore, the virtual address generated by the user is 32 bits in size. The least significant 12 bits of the address represent the offset inside the page where the actual word is located.

The most significant 20 bits of the address are mapped to a 6-bit physical address (page#) by the page table because our physical memory consists of

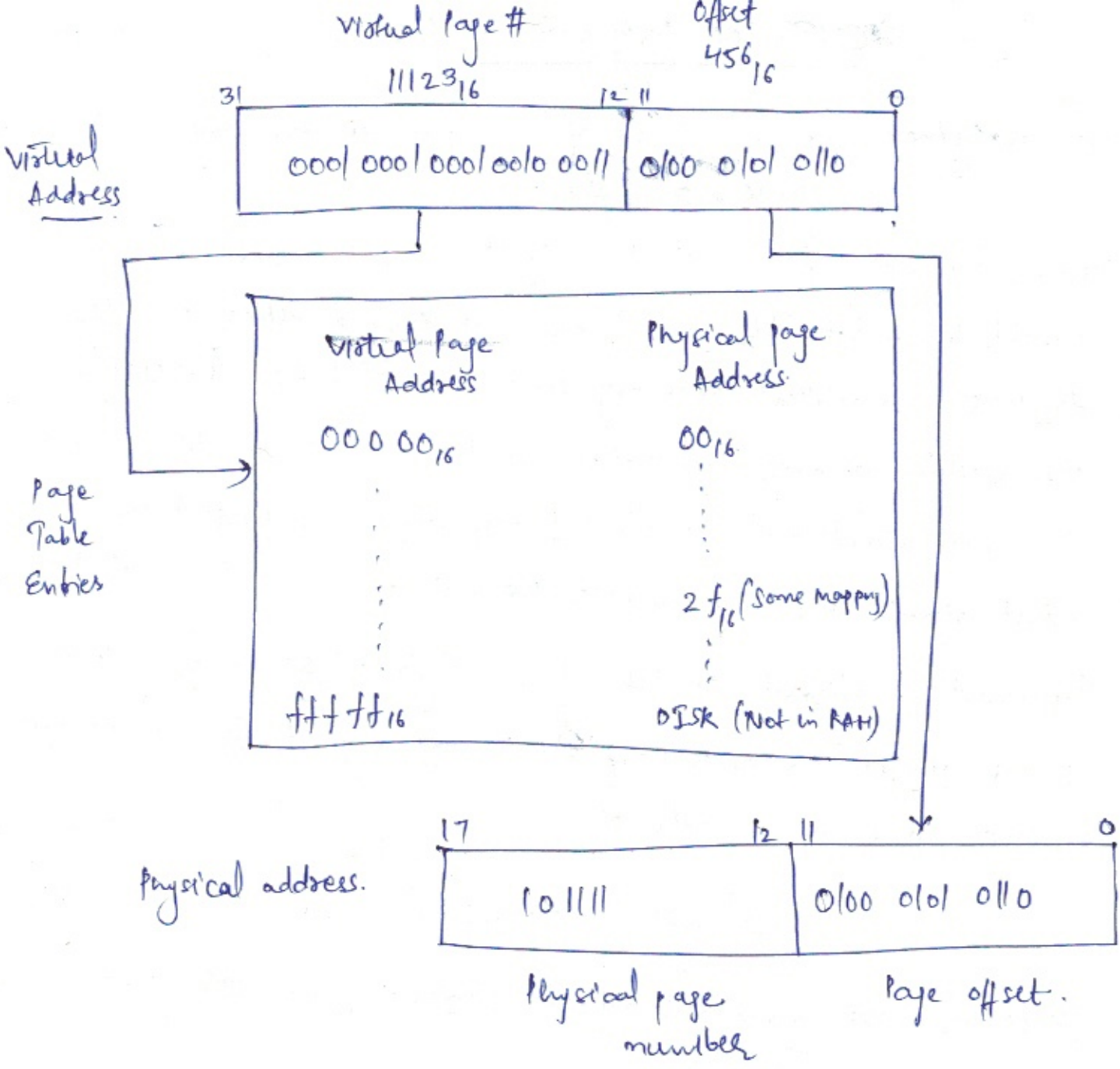
$$2^{18} \text{ bytes} = 64 \text{ pages}$$

Hence the user generated address 11123456_{16} can split into

Page#: 0001 0001 0001 0010 0011 | Offset: 0100 0101 0110

31 12 0

It is noteworthy that there are more pages in the user memory space than are available in the actual physical memory when requested are mapped to the disk.



Fig①: Address Translation of user generated address.

Hardware operations:-

As shown in the figure, if the user generated address is in a page that is already in RAM and is not out of bounds then the page table entry indicates it. In this case, the address translation takes place in hardware and the user generated

address is mapped to the actual physical memory address where the page is located. Otherwise, a hardware trap occurs and an exception is raised.

Software Operations:-

If the page is not in RAM, a page fault occurs and the following steps are taken:

(1) A page fault exception in OS page fault handler is generated.
(2) Control is given to the OS page fault handler.
(3) The OS saves the current process state and uses a page replacement algorithm to evict a page from memory.
(4) If the dirty bit is set, the old page is first written to the disk.
(5) The new page desired by the user is brought into the memory.
(6) Page table is uploaded.
(7) Control is given back to the instruction that can now access the mapping of the new page from the page table entry.

Therefore, address translation requires hardware support that will translate each virtual address by by adding it to the base value (kept in base register).

Also, hardware checks whether the address is valid with the help of bound register. If the page is out of bounds, an exception is raised and os exception handler runs in the privilidged mode.

(2). Memory access time = 100 nanoseconds.

Page fault service time = 8 milliseconds.
(if page was not modified or empty page available)

Page fault service time = 20 milliseconds
(if replaced page was modified).

Max effective access time = 200 nanoseconds

Probability that page to be replaced is modified = 70% = q.

Page fault frequency (p) = ?

Effective access time = p x (q x (page fault time when modified) + (1-q) x (when page available)

+ (1-p) memory access time

$= p \times 10^6 \times (0.70 \times 20 + 0.30 \times 8) + (1-p)\,100 = 200$ nanoseconds

$16.4 \times 10^6 \times p + (1-p) \times 100 = 200$

page-fault rate → $p = 0.000006098$

$= 0.000006\ \%$