# Assignment 9

**Problem 1:**

**a. How many bits are there in the logical address?**

To calculate the number of bits in the logical address, we need to consider both the page number and the offset:

1. Number of pages = 1024 = $2^{10}$, so we need 10 bits for the page number
2. Page size = 4 KB = 4096 bytes = $2^{12}$ bytes, so we need 12 bits for the offset

Total bits in logical address = bits for page number + bits for offset

= 10 + 12 = **22 bits**

**b. How many bits are there in the physical address?**

For the physical address, we need to consider the frame number and the offset:

1. Number of frames = 256 = $2^8$, so we need 8 bits for the frame number
2. The offset remains the same as in the logical address: 12 bits

Total bits in physical address = bits for frame number + bits for offset

= 8 + 12 = **20 bits**

**c. What is the maximum amount of physical memory in this system?**

To calculate the maximum amount of physical memory:

1. Number of frames = 256
2. Size of each frame = 4 KB = 4096 bytes

Maximum physical memory = Number of frames × Size of each frame

= 256 × 4096 bytes = 1,048,576 bytes = 1,024 KB = **1 MB**

Therefore, the maximum amount of physical memory in this system is 1 MB.

**Problem 2:**

Assuming a 2-KB page size (address 0.. 2047), we need to determine the page numbers and offsets for the given address references.To solve this, we'll use the following formula:

- Page number = Address / Page size
- Offset = Address % Page size

**a. 3085**

Page number = 3085 / 2048 = 1, Offset = 3085 % 2048 = 1037

**b. 42095**

Page number = 42095 / 2048 = 20, Offset = 42095 % 2048 = 1135

**c. 215201**

Page number = 215201 / 2048 = 105, Offset = 215201 % 2048 = 161

**d. 650000**

Page number = 650000 / 2048 = 317, Offset = 650000 % 2048 = 784

**e. 2000001**

Page number = 2000001 / 2048 = 976, Offset = 2000001 % 2048 = 1153

**f. 16479315**

Page number = 16479315 / 2048 = 8046, Offset = 16479315 % 2048 = 1107

**Problem 3:**

specifications:

- 21-bit virtual/logical address
- 16-bit physical address
- 2-KB page size

**a. Conventional, single-level page table**

1. Calculate the number of bits for page offset:2 KB = 2048 bytes = 2^11 bytesSo, we need 11 bits for the page offset
2. Calculate the number of bits for page number:Total bits in virtual address - Bits for page offset= 21 - 11 = 10 bits
3. Calculate the number of entries:Number of entries = 2^(number of bits for page number)= 2^10 = 1024 entries

Therefore, a conventional, single-level page table would have **1024 entries**.

**b. Inverted page table**

1. Calculate the number of bits for frame number:Total bits in physical address - Bits for page offset= 16 - 11 = 5 bits
2. Calculate the number of frames:Number of frames = 2^(number of bits for frame number)= 2^5 = 32 frames

Therefore, an inverted page table would have **32 entries**.

**Problem 4:**

**a. If a memory reference takes 50 nanoseconds, how long does a paged memory reference take?**

In a paging system with the page table stored in memory, each memory access requires two memory references:

1. One to access the page table
2. One to access the actual data

Therefore, the time for a paged memory reference is:Time = 2 × Memory reference time

= 2 × 50  = **100 ns**.

**b. If we add TLBs, and if 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes 2 nanoseconds if the entry is present.)**

To calculate the effective memory reference time with TLBs, we need to consider two scenarios:

1. TLB hit (75% of the time):Time = TLB access + Memory access= 2  + 50 = 52
2. TLB miss (25% of the time):Time = TLB access + Page table access + Memory access= 2  + 50  + 50 = 102

Now, we can calculate the weighted average:Effective time = (0.75 × 52) + (0.25 × 102)

$= 39 + 25.5 =$ **64.5 ns**