Logical Memory with 32 bytes = 2^5, each partition (offset)= page size = 8 bytes = 2^3

2 bits	3 bits
p	d

P + d = 5 bits

Page 0	Offset = 8 bytes	00
Page 1	8 bytes	01
Page 2	8 bytes	10
Page 3	8bytes	11

Single layer/one layer/one level/simple

Logical address = 5 bits

Limit register = 3 bits , 000, 001 .. 111

8 bytes, offset, page size

1 byte	000, address for offset
1 byte	001
1 byte	010
1 byte	011
1 byte	100
1 byte	101
1 byte	110
1 byte	111

Logical memory

Capacity = 32 bytes , 4 partitions = 2^2 , 2 bit are required to address the memory . if this memory , is a logical memory page 0, page 1, page 2 and page 3.

If the momory is physical memory = 32 bytes, (RAM, main memory) = frame 0, frame 1 , frame 2 and frame 3, each partition (offset) (frame size) = 8 bytes = 2^3

2 bits	3 bits
f	d

Physical memory address = 5 bits = f+d

Page size = frame size

8 bytes from the offset = 2^3 ===

3 bits are required to indentify the offset .

Storage

1 KB = 1024 B = 2^{10} B

1 MB = 1024 x 1024 = 1 KB x 1KB = 2^{10} x 2^{10} = 2^{20}

1 GB = 1024 x 1024 x 1024 = 1 KB x 1KB x 1KB = 2^{30}

16 locations = $2^{4,}$ the p is 4 bits

Offset = 4 bytes = $2^{2,}$ the d is 2 bits

Logical address = (p + d) bits

(4 + 2) bits = 6 bits

Total storage = 8 x 4 = 32 bytes

Each partition = 8 bytes

How many partitions to store 32 bytes ?

Total storage / offset (partition size) = number of partitions

32 / 8 = 4 partitions, how many bits to identify the memory ,

4 partitions = $2^{2,}$ 2 bits are required for addressing

Main memory

32 frames = 2^5

offset = 4 bytes = 2^2

physical address = f+d = 5 + 2 = 7 bits

Q1, a)

16 MB= storage = 16 x 1 MB = 16 X 1KB X 1 KB = 2^4 x 1024 X 1024

= 2^4 x 2^{10} X 2^{10} = $2^{4+10+10}$ = 2^{24} bytes

Partitions size = 64 KB = 64 x 1KB = 64 x 1024 = 2^6 x 2^{10} = 2^{16}

8 bits	16 bit
p	d (offset)

p + d = 24 bit

p = 24 – 16 = 8 bits

how many partitions ?

2^{24} / 2^{16} = $2^{(24-16)}$ = 2^8 = 256 partitions

Q1 , b

Offset = 64 KB = limit register

64 KB = 64 x 1 KB = 64 x 1024 = 2^6 x 2^{10} = 2^{16}

, 16 bits limit register bits

Q2

Consider a memory with a contiguous allocation scheme consisting of 4 partitions of 600K, 400K, 500K, and 300K (in order) respectively. Given a set of processes with 200K, 300K, 470K, and 500 K (in order):

left = 400 k k , left= k k , left = 200k k, left = k

200k ,		300k	
600k	400k	500k	300k

200K, 300K, 470K, and 500 K (in order):

Worst fit

470k, 500k are in the waiting

Q3

Consider a logical address space of sixty-four pages of 1024 words each mapped onto a physical memory of 32 frames. Frame size = page size/offset.
a. How many bits are there in the logical address?
b. How many bits are there in the physical address?

A) 64 pages = 2^6 , p = 6 bits

offset = 1024 = 2^{10} , d = 10 bits

logical address = p+d = 6 bits + 10 bits = 16 bits,

storage of logical memory = 64 x 1024 bytes , or 2^6 x 2^{10}

total logical memory capacity = 2^{16}

32 frames = 2^5 frames , f = 5 bits , frame size = page size = 1024 words = 2^{10}

physical address = f+ d = 5 + 10 = 15 bits

total physical memory (main memory) capacity = 2^{15}

main/primary/physical memory address = f+d =

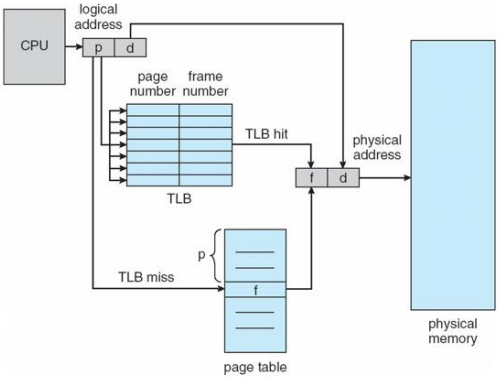
- Percentage of times that a particular page number is found in TLB – Hit ratio
- Example:

  Assume 80-percent hit ratio. It takes 20 nanoseconds to search the TLB and 100 nanoseconds to access memory. Calculate effective access time.

  Solution:

  Effective access time =

  0.80 *120 +0.20 * 220 = 140 nanoseconds

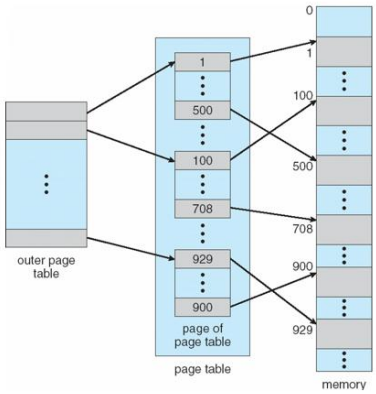


Assume the usage of Translation Look-aside Buffer (TLB) with 70 percentage hit ratio, and also assume that it takes 30 nanoseconds to search the TLB and 200 nanoseconds to access memory. Identify the effective memory access time.

EAT = 0.7 (30 nano + 200 nano) + 0.3 (30 nano+ 200 nano + 200 nano)

= 0.7 (230) + 0.3 (430) nanoseconds

= 161+ 129 = 290 nanoseconds



EAT for two level paging

EAT = 0.7 (30 nano + 200 nano) + 0.3 (30 nano+ 200 nano + 200 nano+ 200 nano)

= 0.7 (230) + 0.3 (630) nanoseconds

161+ 189 = 350 nanoseconds

Q5.

Simple paging system

EAT = p (10 ns + 180 ns) + (1 -p) (10 ns + 180 ns + 180 ns)

200 ns = p190ns + (1 – p) (370ns)

200 = 190p + 370 – 370p

200 – 370 = 190p -370p

-170 = -180p

170/180 = p

P = 0.944, 94.44 % hit ratio

Two level

EAT = p (10 ns + 180 ns) + (1 -p) (10 ns + 180 ns +180ns +180 ns)

6.

a) conventional single-level page table, simple paging system, one level page table system

given Logical address = 32 bit = p + d

Given Page size = 4 KB = 4 x 1KB = 2^2 x 2^{10} = 2^{12}

d = 12 bit

p = 32 bit – 12 bit = 20 bit

number of entries = number of pages = 2^{20}

20 bits	12 bits
p	d

b) An inverted page table

given physical memory = 512 MB = 512 x 1 MB = 2^9 x 2^{20} = 2^{29}

29 bits = i + d

frame size = page size = offset = 4 KB = d = 12 bit

17 bit	12
f	d

Total bit of frame = 29 -12 = 17 bit

Number of frames entry = 2^{17}.

Question 7,

Use calculator

a) 19366 = 19366 /1024= 18.91210937

page number = 18, offset = 0.91210937 x 1024= 934

b) 30000 = 29.296875

page number = 29, offset = 0. 296875x 1024= 304

c) 256 = 0.25

page number = 0, offset = 0. 25x 1024= 256

Page 0	Offset = 256, internal fragmentation = 1024 – 256 = 768

Question 8,

a) 219 + 430 = 649 physical address

b) 2300 + 10 = 2310

c) cannot fitted, error found

d) 1327 + 400 = 1727

e) cannot fitted, error found

Check the offset and length of limit register

Question 9

Consider a system that uses a two-level page table with page size of 4KB and 32-bit logical addresses. The first 8 bits of the address serve as the index into the first-level page table.
a. How many bits are needed to specify the second-level index?
b. How many entries are in a level-one page table?
c. How many entries are in a level-two page table?
d. How many pages are in the virtual address space?

P1+P2 +d = 32 bits

P_1	P_2	d
8 bits	12 bits	12 bits

Offset = 4 KB = 2^2 x 2^{10} B = 2^{12}

a) 32 – 12 = 20 bits for P1 + P2

P_2 = 32 – (12 – 8) = 12 bits

b) entries for P1 = 2^8

c) entries for P2 = 2^{12}

d) 2^{20} (Each 2^8 level-one page table entry access a level-two page table with 2^{12} entries, $2^8 \times 2^{12} = 2^{20}$)

process = 9 bytes, page size = 2 bytes

9/2 = 4.5

Page 4, offset = 1 , internal fragmentation = 2 -1 = 1 byte

Page 0	2 bytes	
Page 1	2 bytes	
Page 2	2 bytes	
Page 3	2 bytes	
Page 4	1 bytes , internal = 2 -1 = 1 byte	