Q1

Considering a computer system with a 10-bit logical address. Translate the logical address 1000011011 to the corresponding physical address for the following two cases:

a) Assuming paging is used, the page size is 128 bytes and the page table is as given in Figure Q4a.

b) Assuming segmentation is used, the maximum segment size that the system can support is 256 bytes, and the segment table is as given in Figure Q4b.

0	01011
1	00010
2	00010
3	10101
4	01001
5	01100
6	11010
7	00110

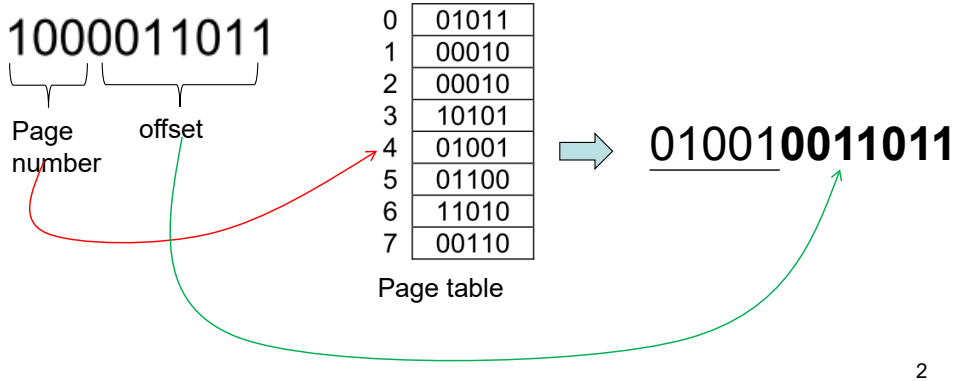
Figure Q4a

0	00010000	010000000000
1	00001000	100000000000
2	00100000	010011010000
3	00011000	100001100000

Figure Q4b

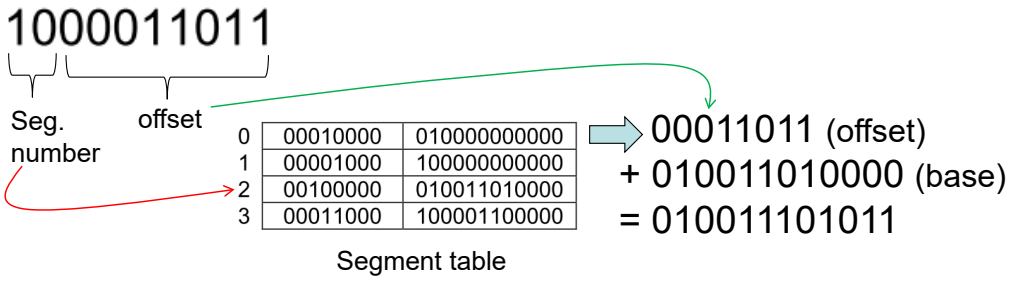
Q1 (a)

- Given page size is 128 bytes→ **7** bits are needed for page offset and **3** bits for page number.



Q1 (b)

- The maximum segment size is 256 bytes→ **8** bits are needed for the segment offset and **2** bits for segment number.



Q2

A paged memory system uses the page size of 1024 bytes. Size of a page table entry is 4 bytes and the logical address space is 2^{30} bytes.

a) What is the size of the page table if single level of paging is used?

b) What is the minimum number of levels of page tables needed in this system to ensure that the outmost page table will fit within a single page frame?

c) Draw an address translation diagram to show how logical address translation is performed.

Q2 (Answer)

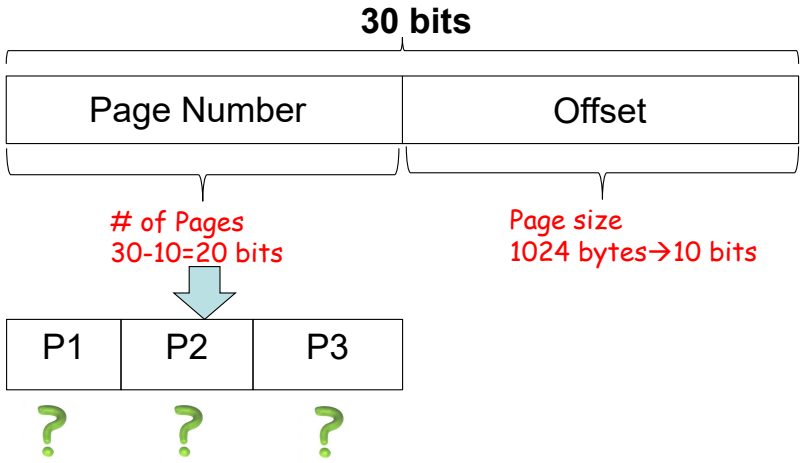
- a) size of page table: #pages × page_table_entry_size

  $=2^{30}/2^{10} \times 4 = 2^{22}$ = 4 megabytes
- b) If two-level paging is used, size of outer page table: $\mathbf{2^{22}}/2^{10} \times 4 = 2^{14}$ = 16k bytes > 1024 bytes.

  If three-level paging is used, size of outer page table: $\mathbf{2^{14}}/2^{10} \times 4 = 2^{6}$ bytes < 1024 bytes.

  → at least three levels.

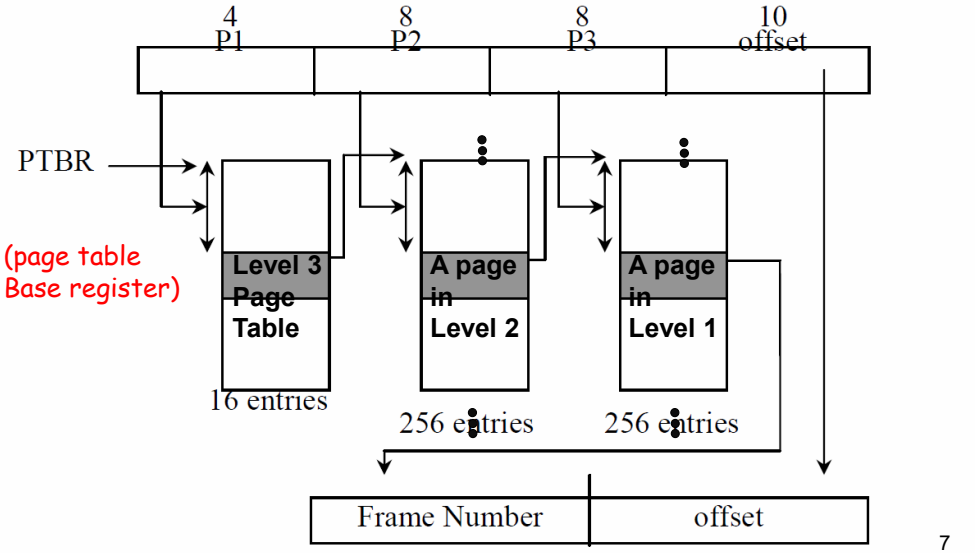
Q2 (c)



Hints:
P1: the index in the level 3 (outmost). Each entry in level 3 page table points to a page in Level 2.
P2: the index within a page in the level 2. Each entry in level 2 page table points to a page in Level 1.
P3: the index within a page in the level 1. Each entry in level 1 page table points to a memory frame.

Q2 (c)

c) Each page can contain $2^{10}/2^{2} = 2^{8}$ entries, the outmost page table has 2^{4} entries. The virtual address format is as shown in the following figure:



Q3

1. A computer has four page frames. The time of loading, time of last access, and the R bit for each page are as shown below (the times are in clock ticks):

PAGE	LOADED	LAST ACCESS.	R
0	126	279	0
1	230	260	0
2	120	272	1
3	160	280	1

Q3 (a)

a) Which page will FIFO replace?

PAGE	LOADED	LAST ACCESS.	R
0	126	279	0
1	230	260	0
2	120	272	1
3	160	280	1

Q3 (b)

b) Which page will second chance replace?

(different from FIFO)

PAGE	LOADED	LAST ACCESS.	R
0	126	279	0
1	230	260	0
2	120	272	1
3	160	280	1

Q3 (c)

c) Which page will LRU replace?

PAGE	LOADED	LAST ACCESS.	R
0	126	279	0
1	230	260	0
2	120	272	1
3	160	280	1

Q4

2. For each of the page replacement policies listed below, calculate the number of page faults encountered when referencing the following pages. Assume the availability of 4 empty page frames.

0 1 6 0 3 4 0 1 0 3 4 6 3 4

a) FIFO

b) CLOCK

c) LRU

Q4 (FIFO)

	0	1	6	0	3	4	0	1	0	3	4	6	3	4
F1	0	0	0	0	0	4	4	4	4	4	4	4	3	3
F2		1	1	1	1	1	0	0	0	0	0	0	0	4
F3			6	6	6	6	6	1	1	1	1	1	1	1
F4					3	3	3	3	3	3	3	6	6	6
						0	1	6				3	4	0
	P	P	P		P	P	P	P				P	P	P

There are 10 page faults.

Q4 (CLOCK)

Page Number

R Bit

Circular Queue

	0	1	6	0	3	4	0	1	0	3	4	6	3	4
F1	0,0	0,0	0,0	0,1	0,1	0,0	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,0
F2		1,0	1,0	1,0	1,0	4,0	4,0	4,0	4,0	4,0	4,1	4,0	4,0	4,1
F3			6,0	6,0	6,0	6,0	6,0	1,0	1,0	1,0	1,0	6,0	6,0	6,0
F4					3,0	3,0	3,0	3,0	3,0	3,1	3,1	3,0	3,1	3,1
						1		6				1		
	P	P	P		P	P		P				P		

There are 7 pages faults.

Clock hand in shaded entry.

Q4 (LRU)

	0	1	6	0	3	4	0	1	0	3	4	6	3	4
F1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F2		1	1	1	1	4	4	4	4	4	4	4	4	4
F3			6	6	6	6	6	1	1	1	1	6	6	6
F4					3	3	3	3	3	3	3	3	3	3
						1		6				1		
	P	P	P		P	P		P				P		

There are 7 page faults.