KEY
Worksheet 0

Here is some assembly code:

```
load   8192 r1       # load value at memory location 8192 -> r1
add    r1, 1, r2     # put r1 + 1 -> r2
store r2, 8192       # stores r2 -> memory location 8192
```

Assume each instruction takes up 4 bytes in memory.

Assume the program counter (PC) is set to 4096 (4k) when running the first instruction of this sequence. The virtual address space of this process looks like this (not to scale):

```
0     --------------------------
       |                         |
       |                         |
       |                         |
       |                         |
4k     | load   8192, r1         |
       | add    r1, 1, r2        |
       | store r2, 8192          |
       |                         |
       |                         |
8k     | 100                     |
       |                         |
       |                         |
       |                         |
       |                         |
12K   --------------------------
```

VA	PA
4 KB	64+4 = 68 KB
8KB	64+8 = ~~74~~KB* 72KB
4KB+4	68KB+4
4KB+8	68KB+8
8KB	~~74 KB~~ 72KB

The total size of this virtual address space is 12 KB.

Assume this is a system with "base and bounds" registers used for memory relocation and protection.

In this example, assume that the process's address space is loaded into physical memory at physical address 64 KB (this is the base). The bounds is set to the size of the address space: 12 KB.

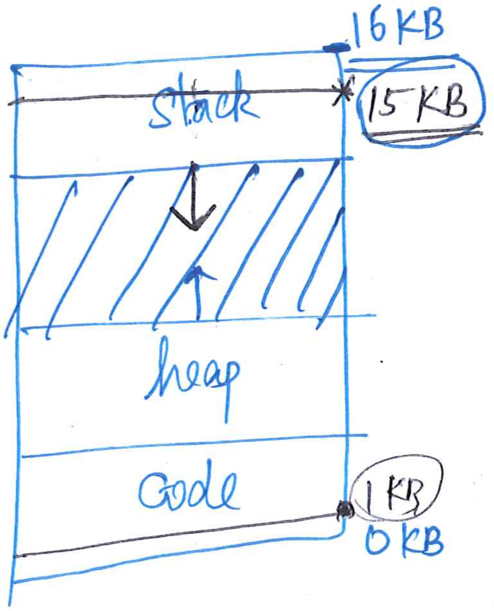
List all the physical memory locations that are referenced during the execution of this three-instruction sequence.

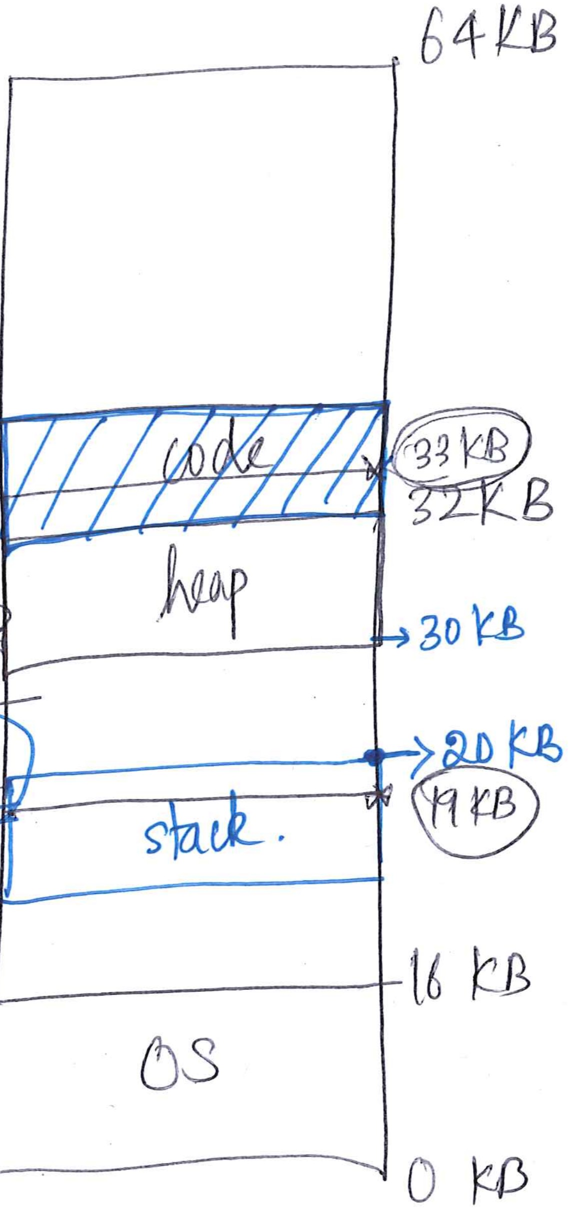
So, basically I just got 3/5 correct but still I would have got 2 points for this worksheet if it was graded! :)

* Maybe I should learn to add 2 number properly! ☺

Remember not to make similar mistakes on worksheets/exams. Good luck!

Segmentation





Seg	Base	Bounds	Grows +ve	Prot ~~positive~~
C	32 KB	2KB	1	RX
H	30 KB	2KB	1	RW
S	20KB	2KB	0	RW

15 KB = 0X 3C00

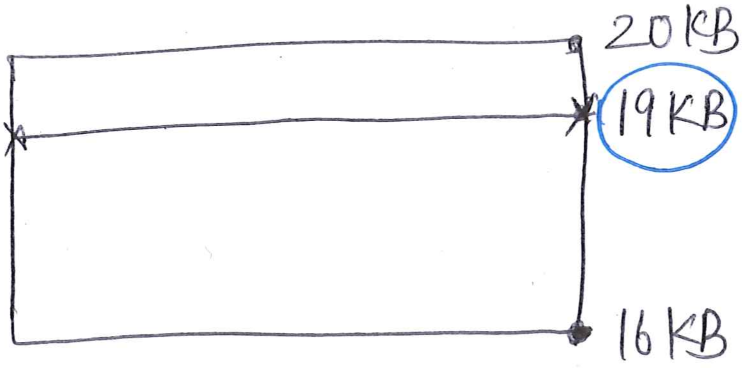
~~00~~ 11 | 1100 0000 0000

11 → stack

1100 0000 0000 → 3KB

base = 20 KB

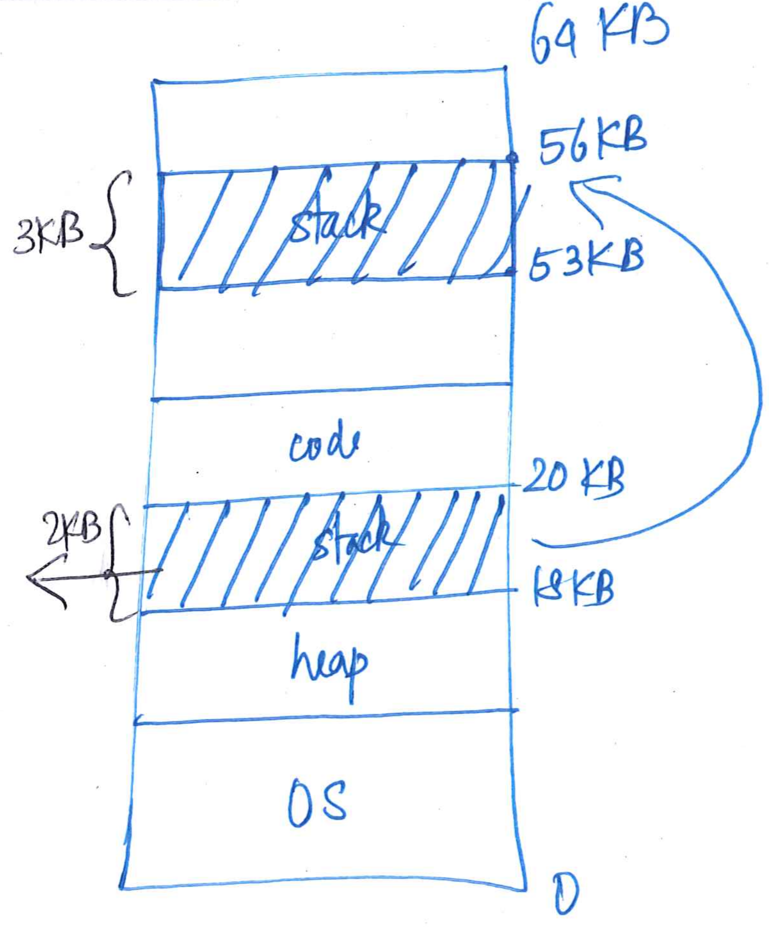
offset = 3 KB

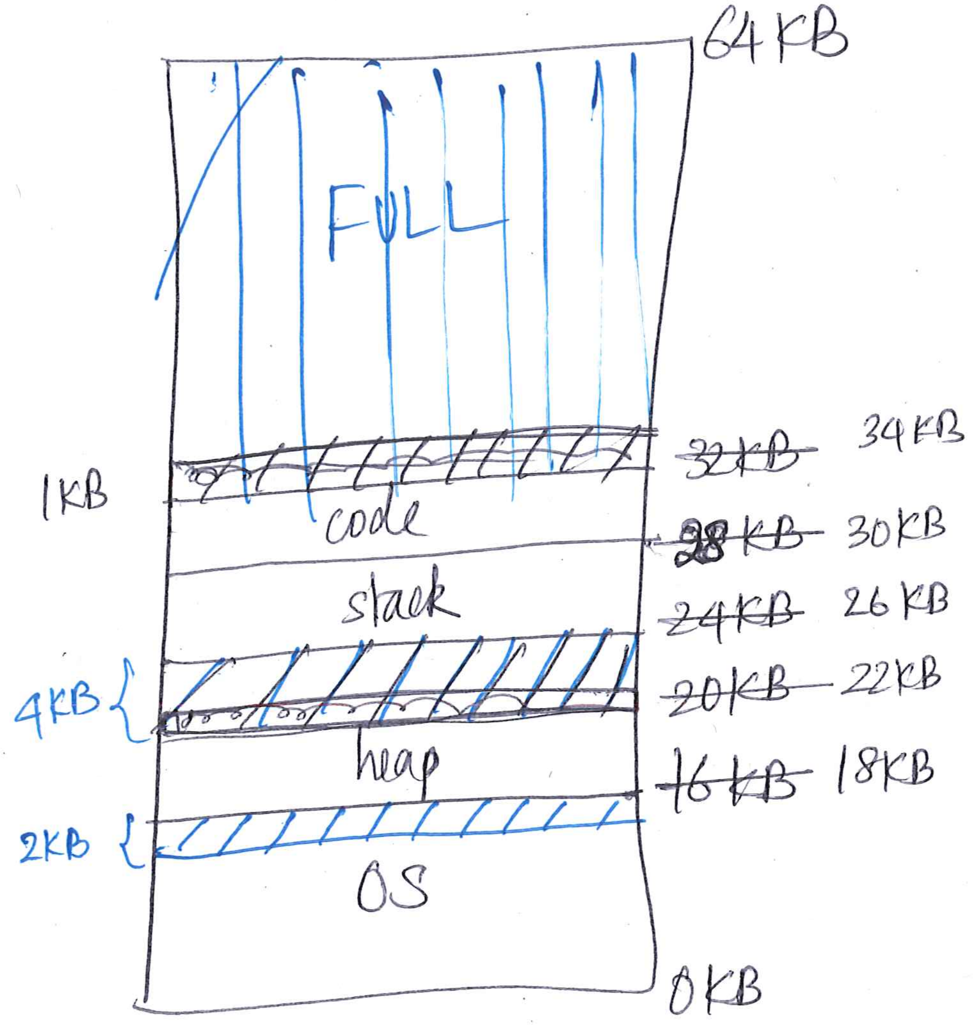


$$\underline{20\text{KB}} - \underbrace{4\text{KB}}_{\text{max. size of stack}} + \boxed{3\text{KB}}$$

$$= 20\text{ KB} - 1\text{KB} = \underline{19\text{ KB}}$$

Base	Bounds
~~20KB~~	~~2KB~~
56KB	3KB



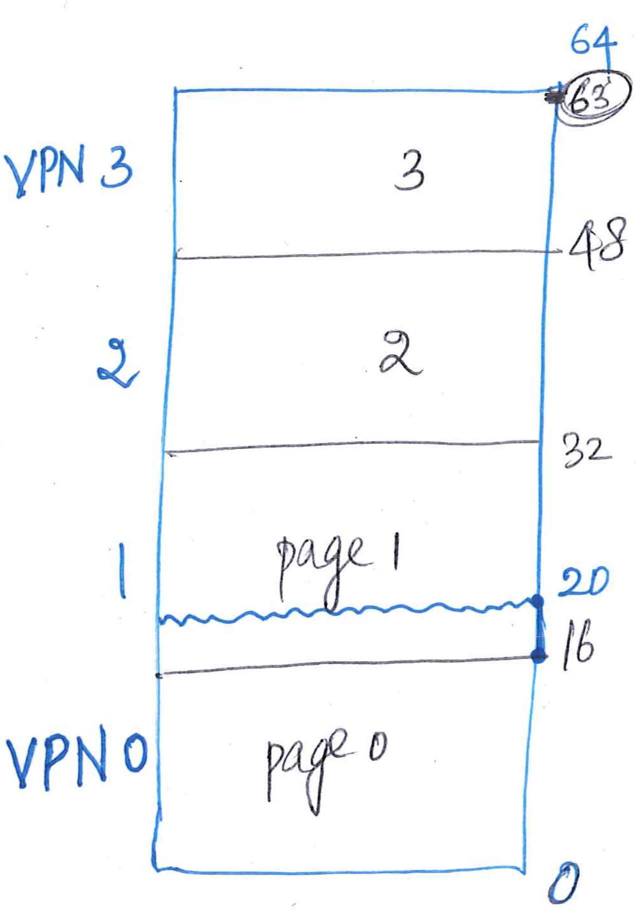


Code - 2 KB
heap - 3 KB
stack - 2KB

7KB

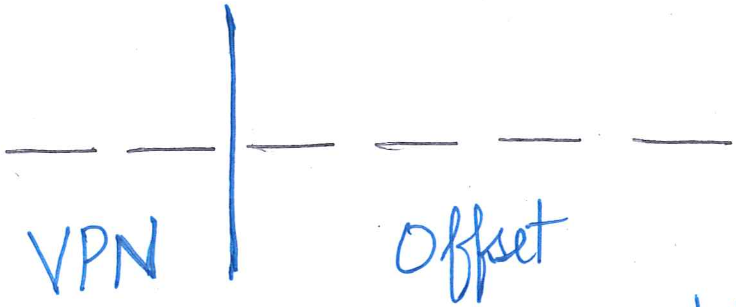
External Fragmentation.

Paging



Virtual Address Space.

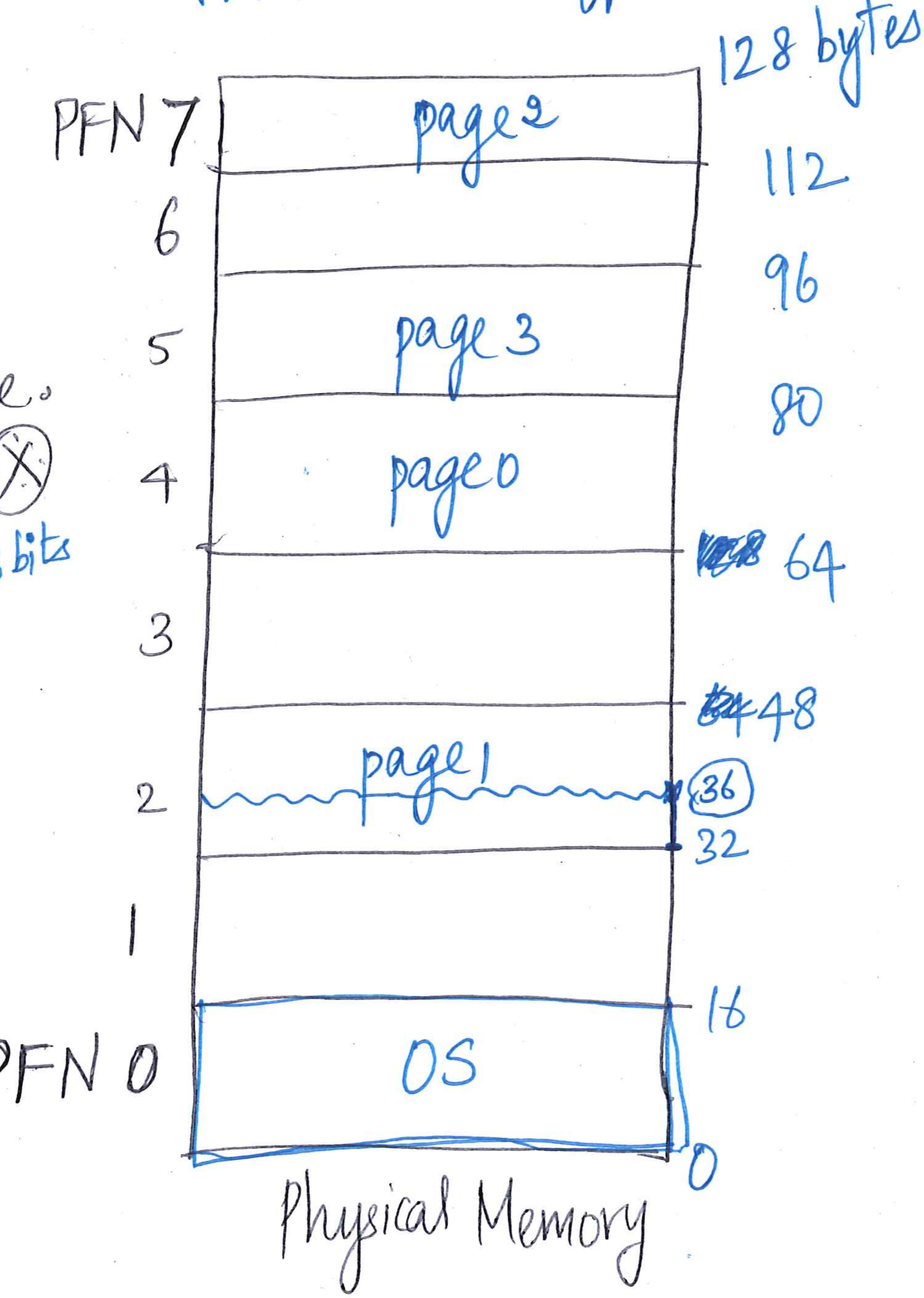
$2^{6} = 64$



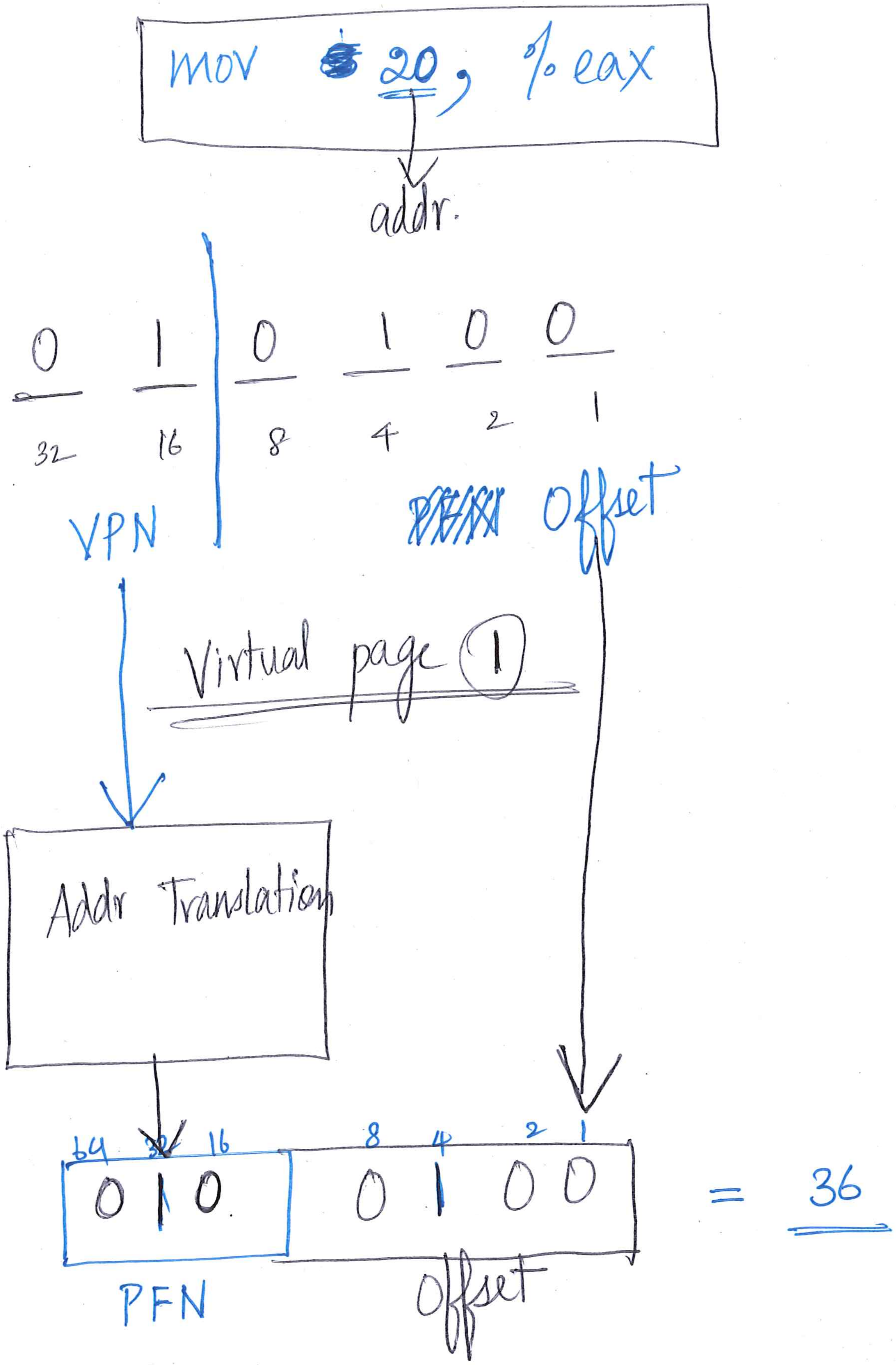
Page Table

VPN	PFN
0	~~2~~ 4
1	2
2	7
3	5

PFN → 3 bits



Physical Memory


mov 20, % eax
addr.
0 1 0 1 0 0
32 16 8 4 2 1
VPN
Offset
Virtual page 1
Addr Translation
64 32 16 8 4 2 1
0 1 0 0 1 0 0
= 36
PFN
Offset