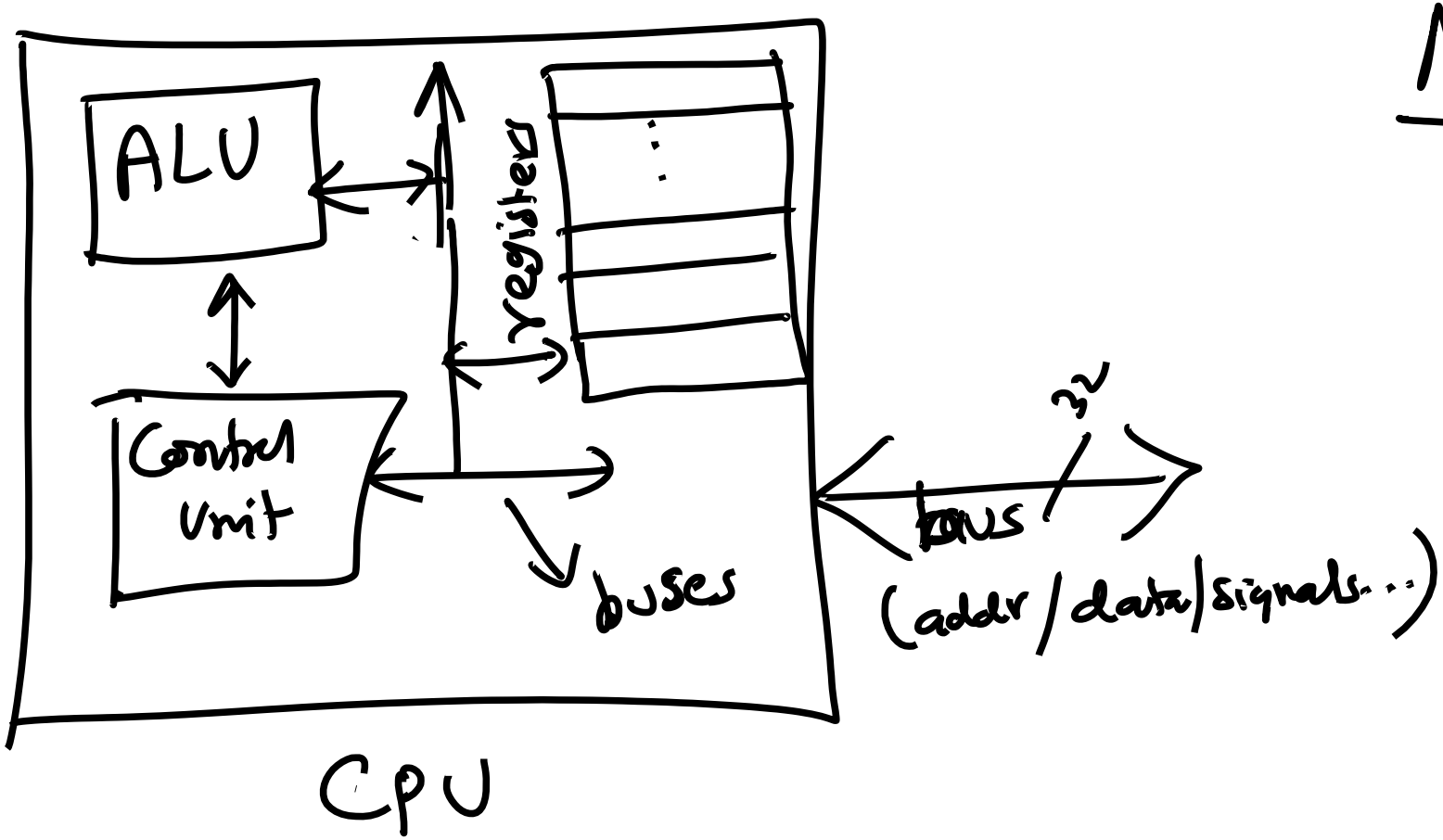
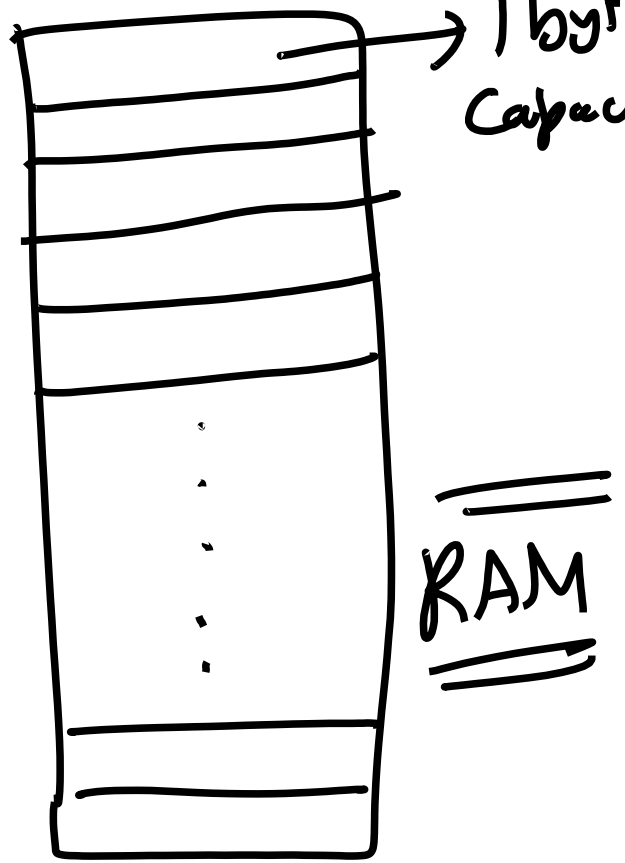
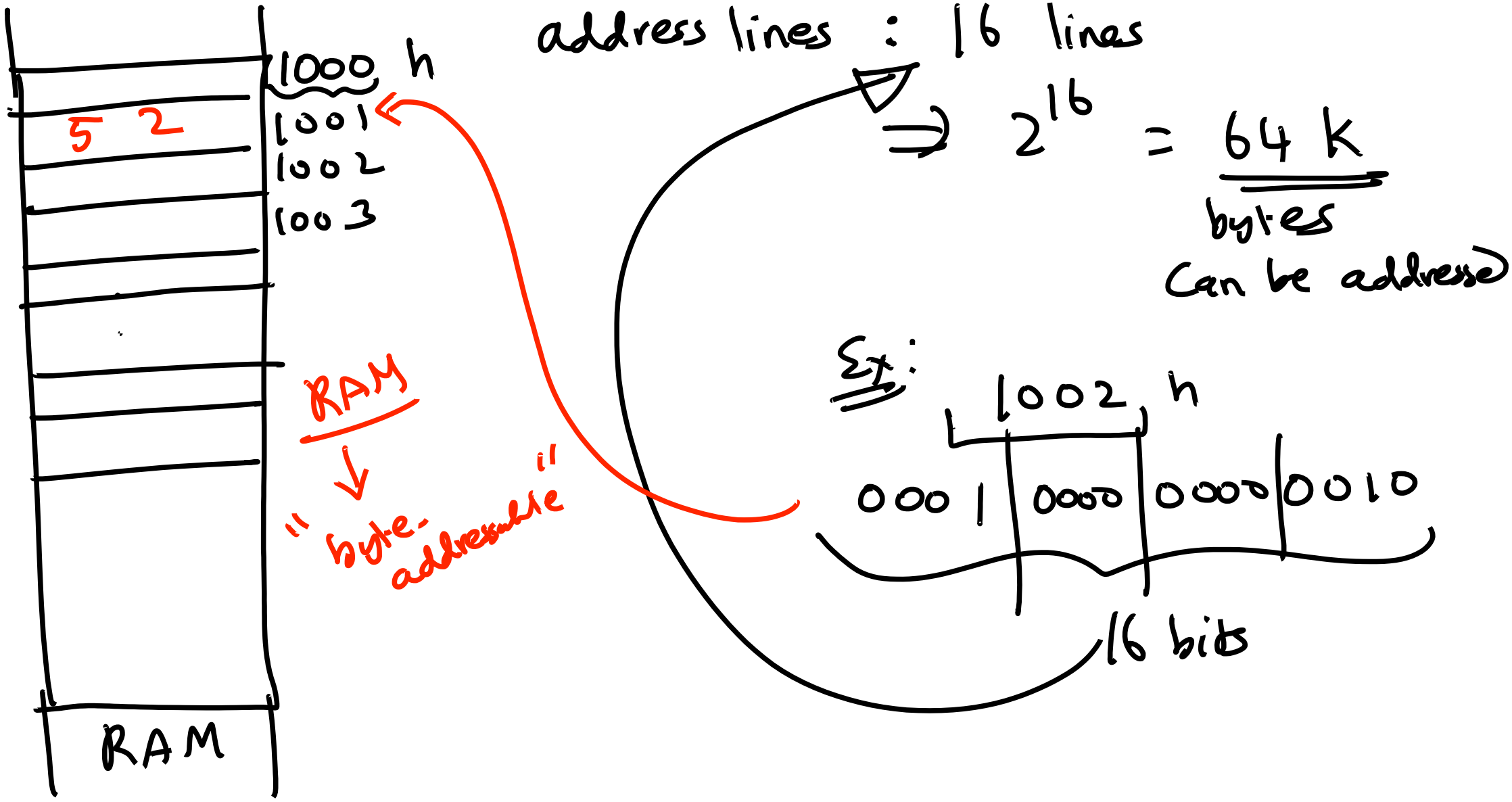
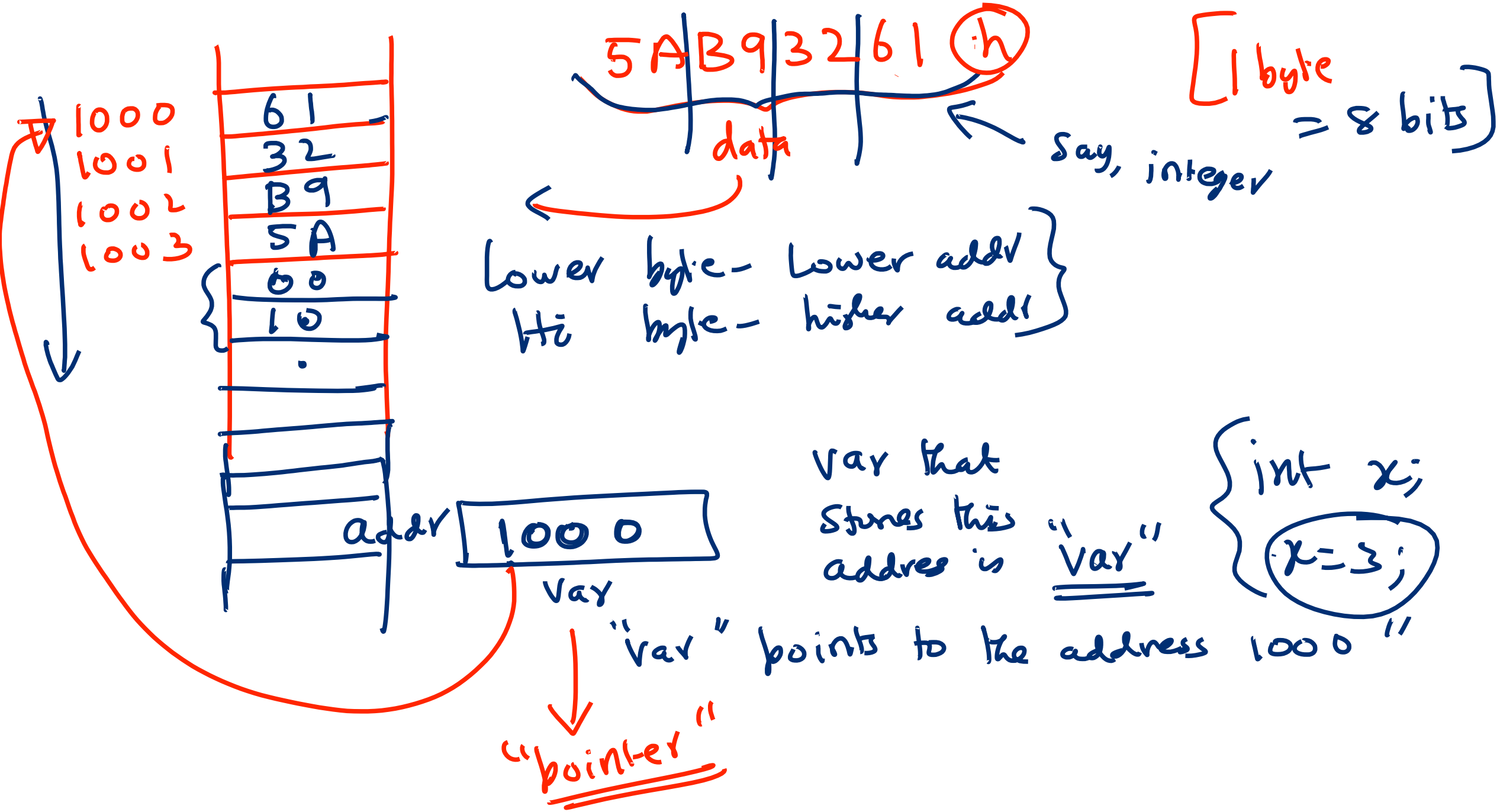
Nov 09, 2017


ALU
registers
Control Unit
buses
32
bus
(addr/data/signals...)
CPU


1 byte
Capacity
RAM


address lines : 16 lines
⇒ 2^{16} = 64 K bytes
Can be addressed
1000 h
1001
1002
1003
5 2
RAM
" byte-addressable "
RAM
Ex:
1002 h
0001 0000 0000 0010
16 bits


5A B9 32 61 h
data
Say, integer
[1 byte = 8 bits]
1000
1001
1002
1003
61
32
B9
5A
00
10
Lower byte - Lower addr
Hi byte - higher addr
addr
1000
Var
Var that stores this address is "Var"
{int x;
x=3;}
"Var" points to the address 1000
"pointer"

In C

- int pointer
- float ptr;
- char ptr.

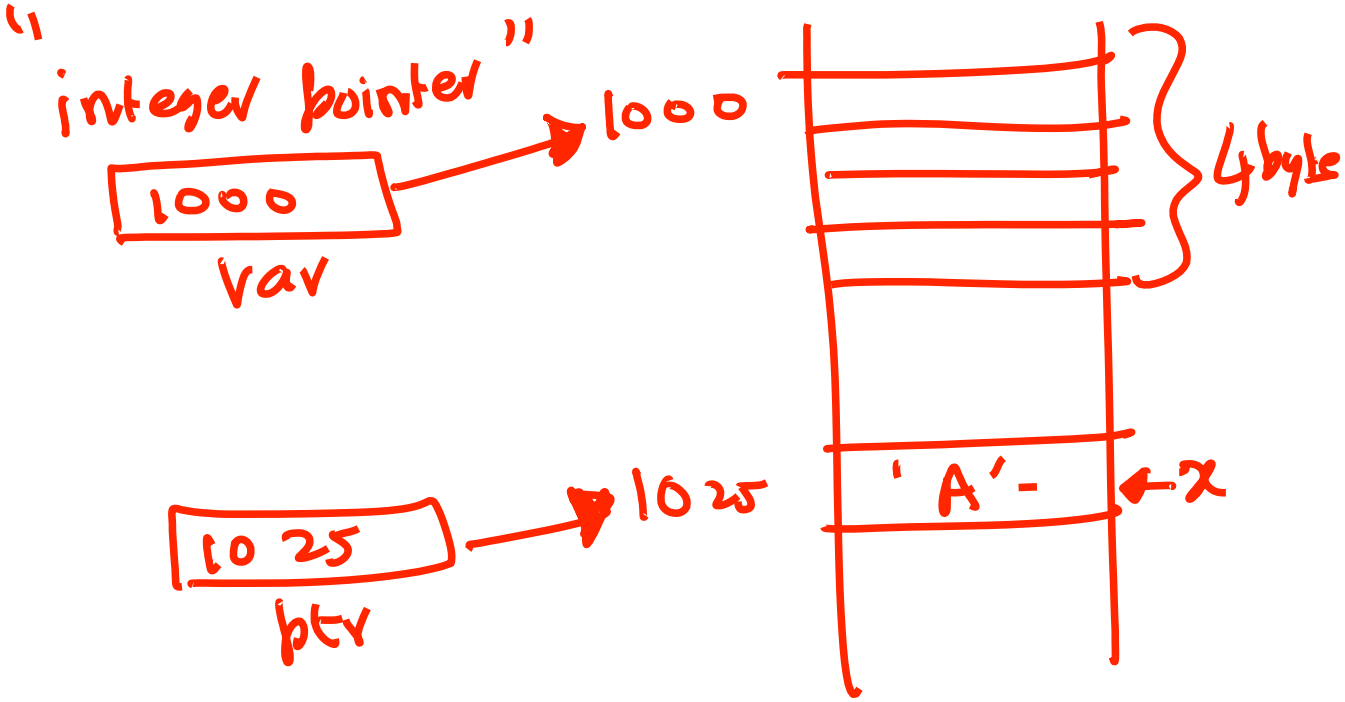
int

→ Char x;

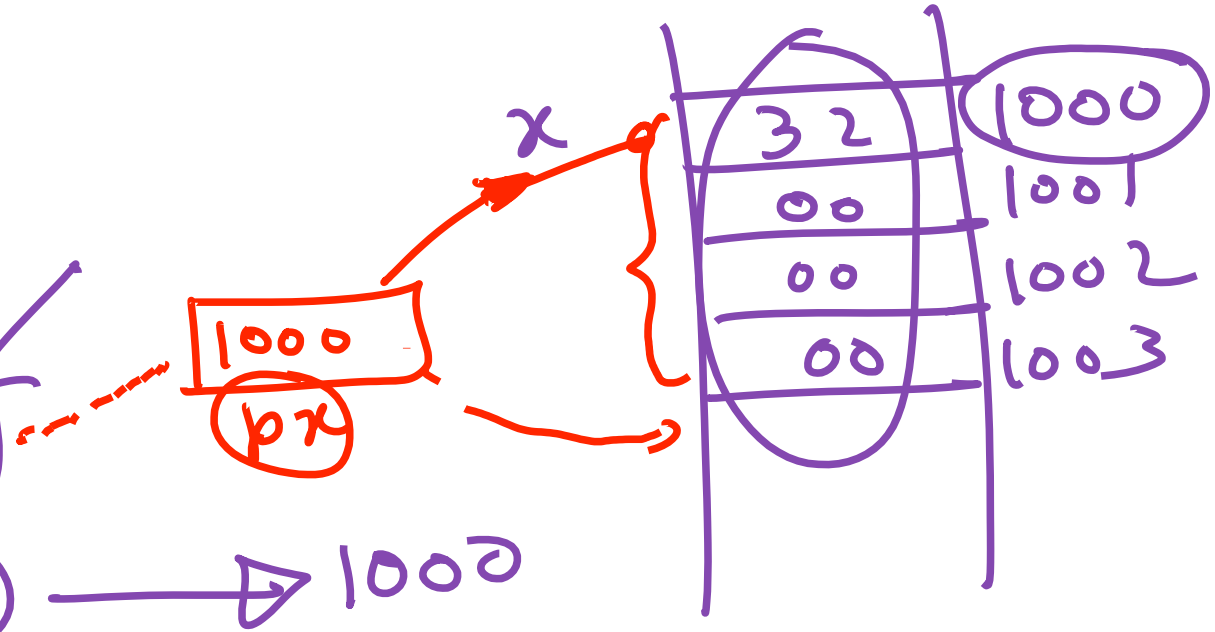
- data type
- int - 4 byte
- float - 4
- Char - 1

pointer declaration:

```
int *px;
float *qx;
char *cx;
→ int y;
```



```
int *px;
int x;
x = 32;
px = &x;
```



address info needs to be stored

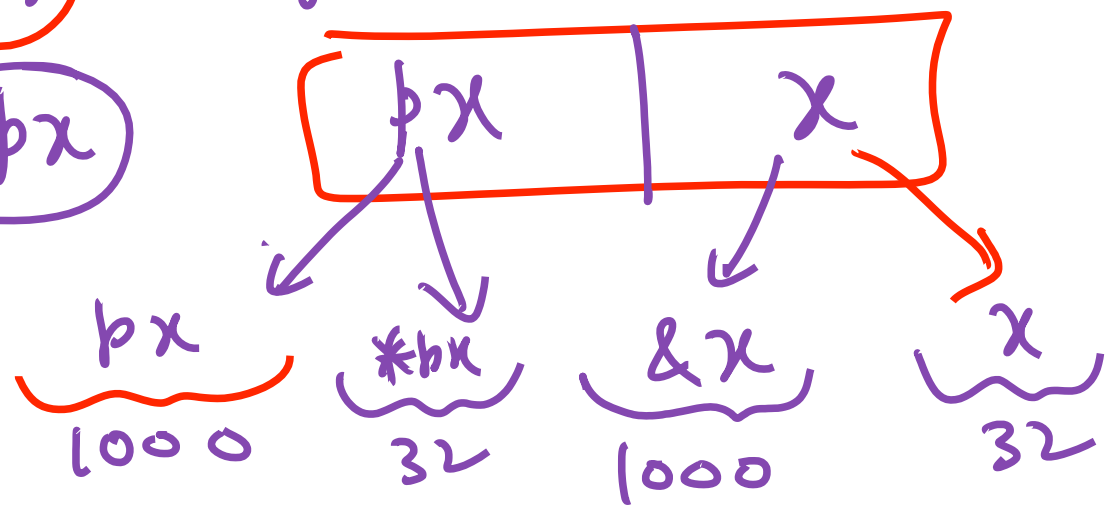
printf(px) → 1000

16 bit machine

32 bit #
+ 32 bit #

px → pointer to var. x

*px



ffooobe4 h

8 x 4 = 32 bits 4 bits

address

2^{32} = 4 Gigabytes

2^{10} = 1K

2^{20} = 1M

2^{30} = 1G

scanf(..... , &x);

x = 5