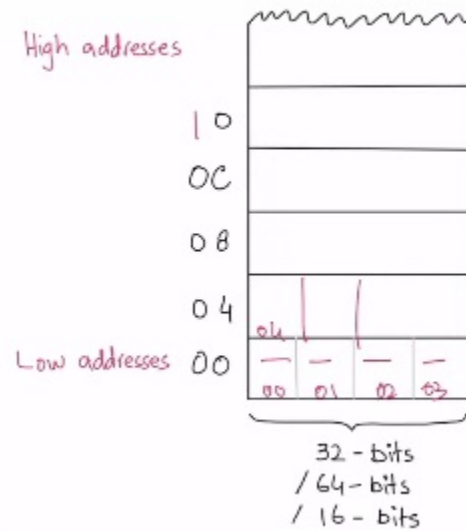
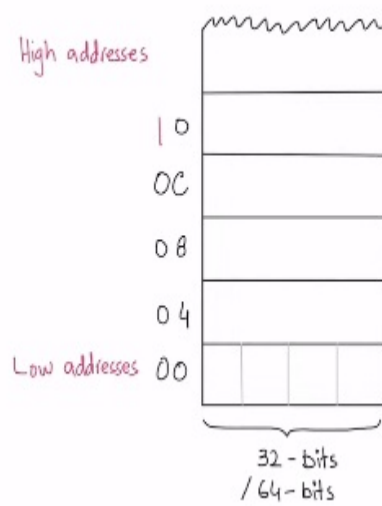


COAL

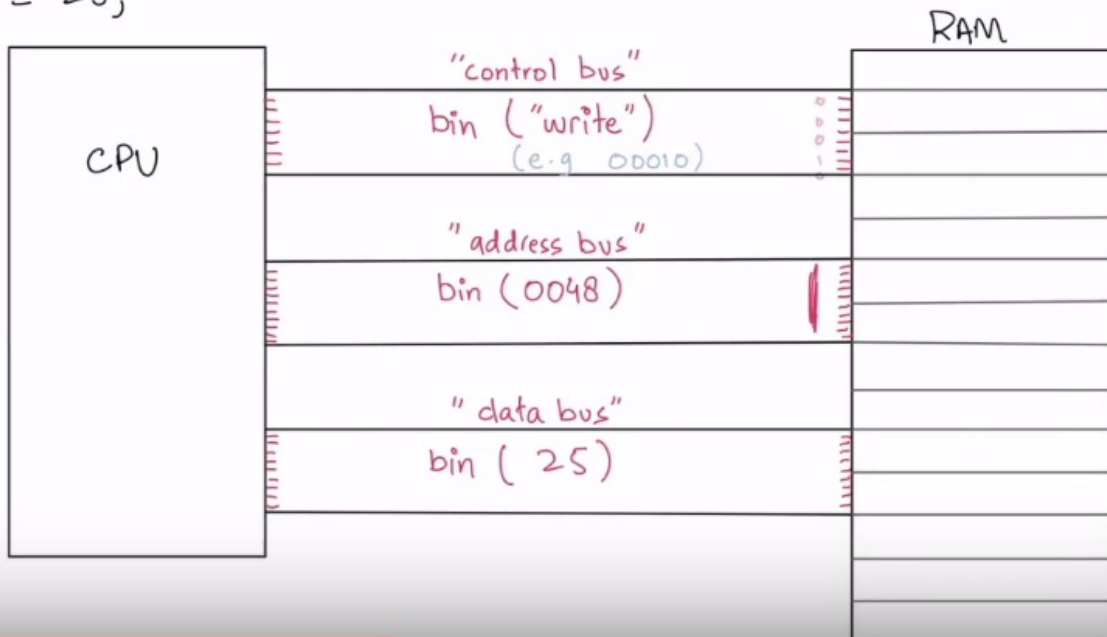
LECTURE 2

Memory Structure



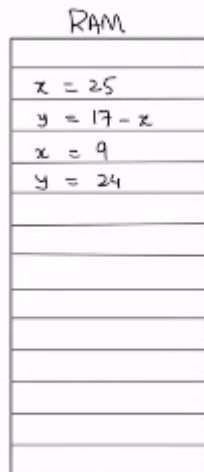
RAM →
LOW ADDRESSES (BELOW) TO HIGH
we split into 4 → 32 bits
8 → 64 bits
2 → 16 bits

$x = 25;$



CPU does **three** works

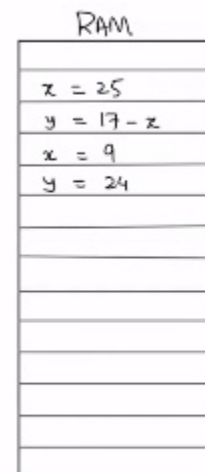
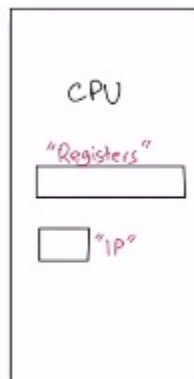
- first it tells what to read or write on ram (physical wire → contain 32 wires) specific pattern send to ram
- set of wires contain address (address where to write/from where to read)
- data (what to write/read)



x value will be 25 bcz we see it from low to high

here comes the prob CPU is very fast but everything else is slow

Where do the instructions come from?



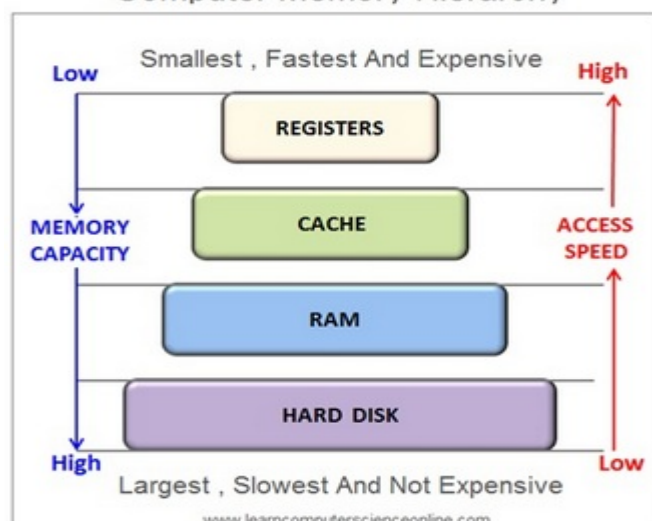
Two problems:

- ① RAM is *slowwww* !
- ② CPU needs to keep track of which instruction it is executing (plus some other stuff like errors)

CPU → working area → register(register **stores data and instructions for quick processing.**)

cpu goes to ram → fetch data from ram → decode → execute

Computer Memory Hierarchy



Instruction
 mov

Operands
 ax , 25

; move 25 to AX register

ax is the name of register
 → mov 25 to ax register

inc
 instruction

ax
 operand

; increment value in AX

Let's view some reference docs...

Intel's Architecture — "IBM PC"

8-bit	
16-bit	
32-bit	8086
64-bit	x86, "IA-32" architecture
	"amd64"

machine size depends on the size of accumulator register

In general, all are based on the
 "Von Neumann Architecture"

- Code and data both stored in memory
- General purpose
- Fetch, decode, execute

intel architecture manual → <https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html>

pg -55

memory.

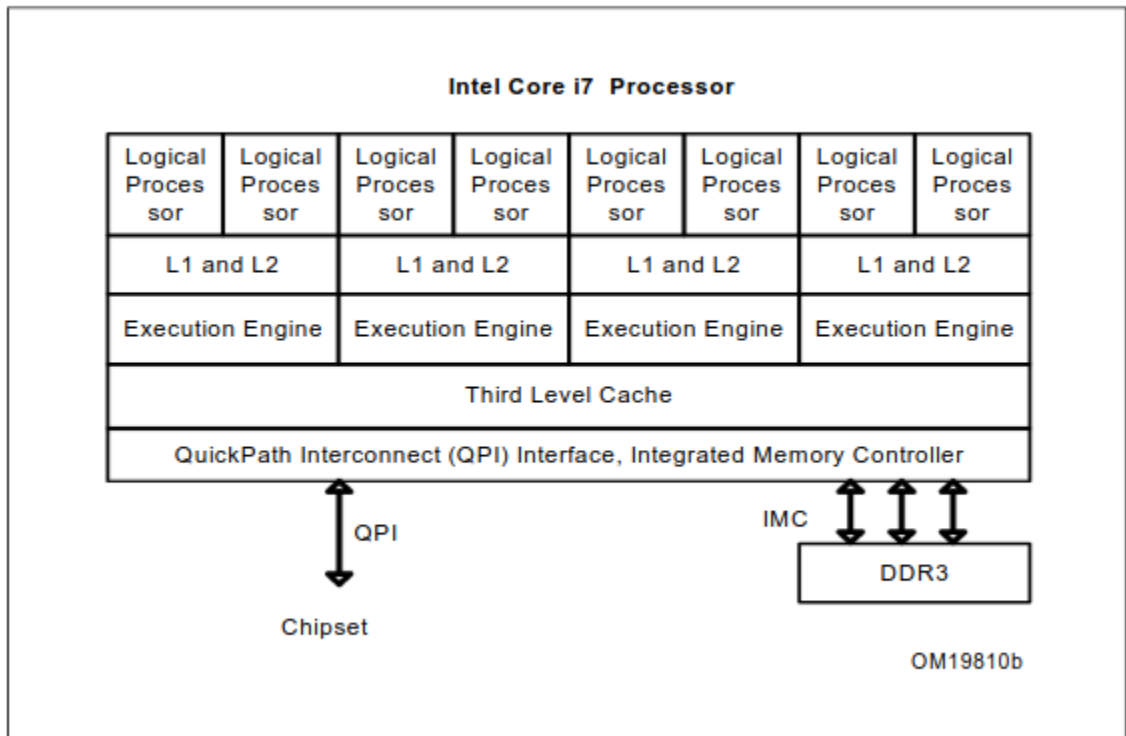


Figure 2-8. Intel® Core™ i7 Processor

DDR 3 → RAM
8 PROCESSORS
L1 AND L2 → CACHE

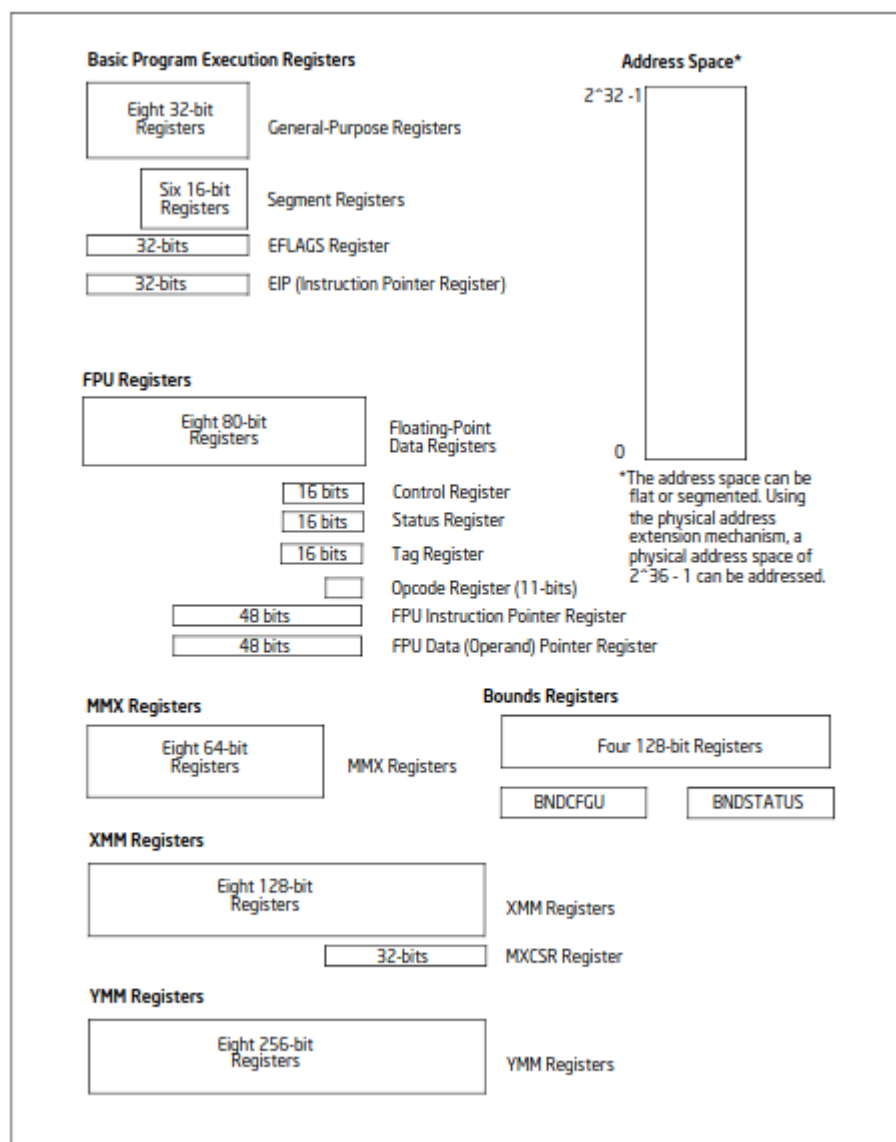


Figure 3-1. IA-32 Basic Execution Environment for Non-64-Bit Modes

