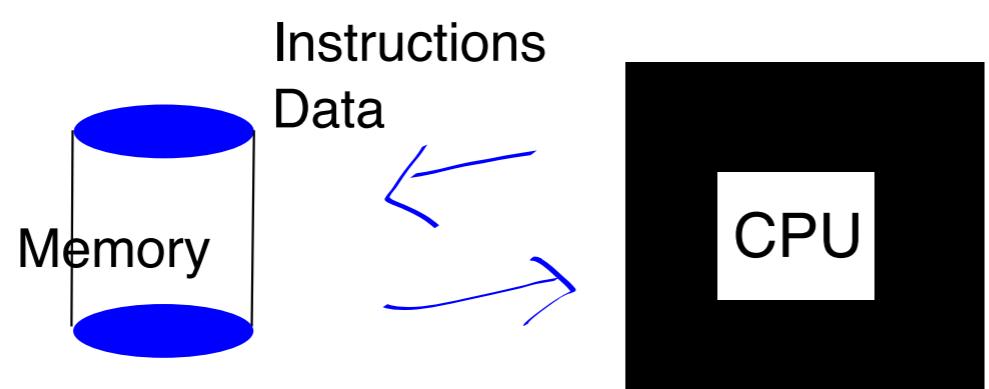


FIT 1047

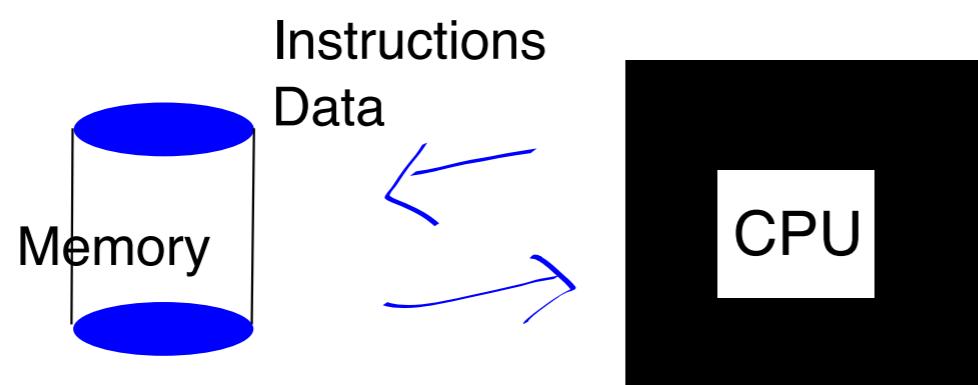
Introduction to computer systems, networks and security

Overview

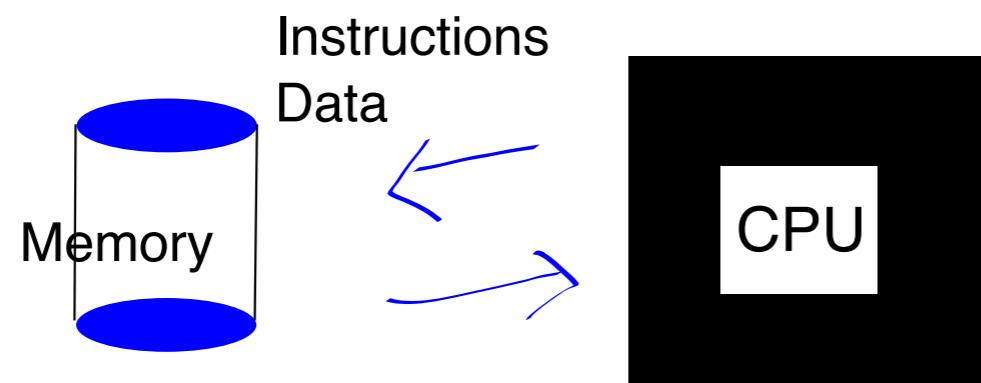
- Look at CPUs in context
- Components on a PC's motherboard



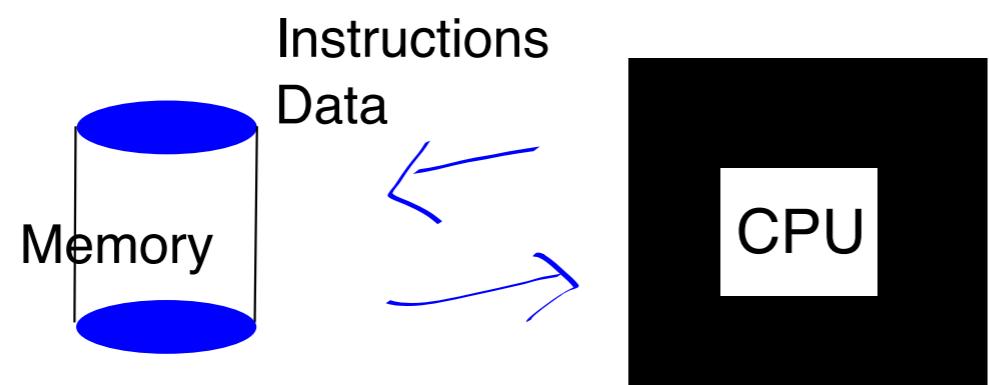
32-bit CPU means a **word** is
32 bit long:
* Registers are 32 bit
* Load 32 bit in one step
* Use up to 32 bit for addresses
* Address up to 2^{32} locations
(for byte-addressing this is 4 GB)

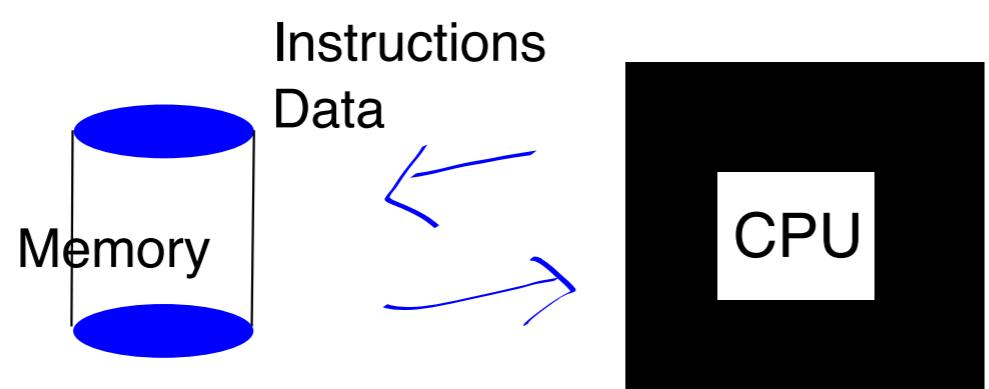


Note that our *toy* CPU MARIE has 16-bit words.
But, it only uses 12 bits for addressing. It can address 2^{12} locations with 16 bit in each.
Thus, it can address $2^{12} \times 2$ bytes.
This is 8192 bytes.

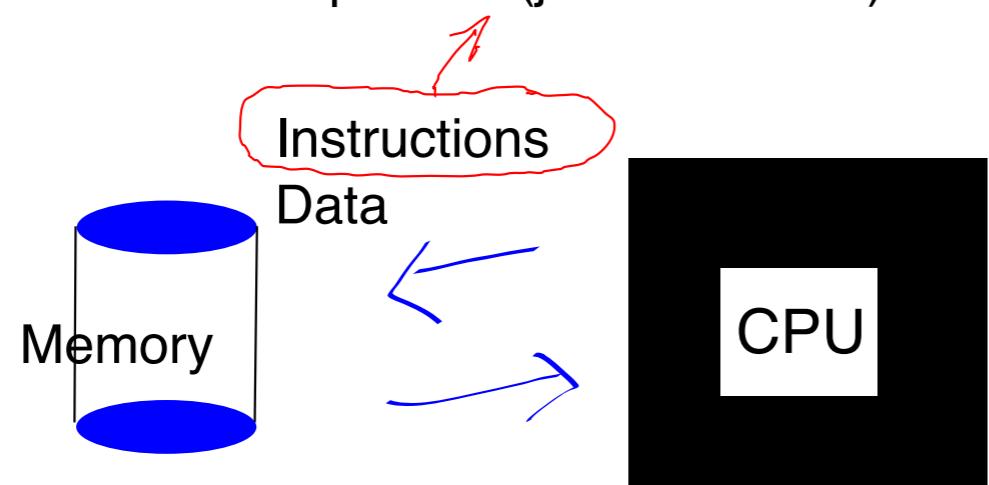


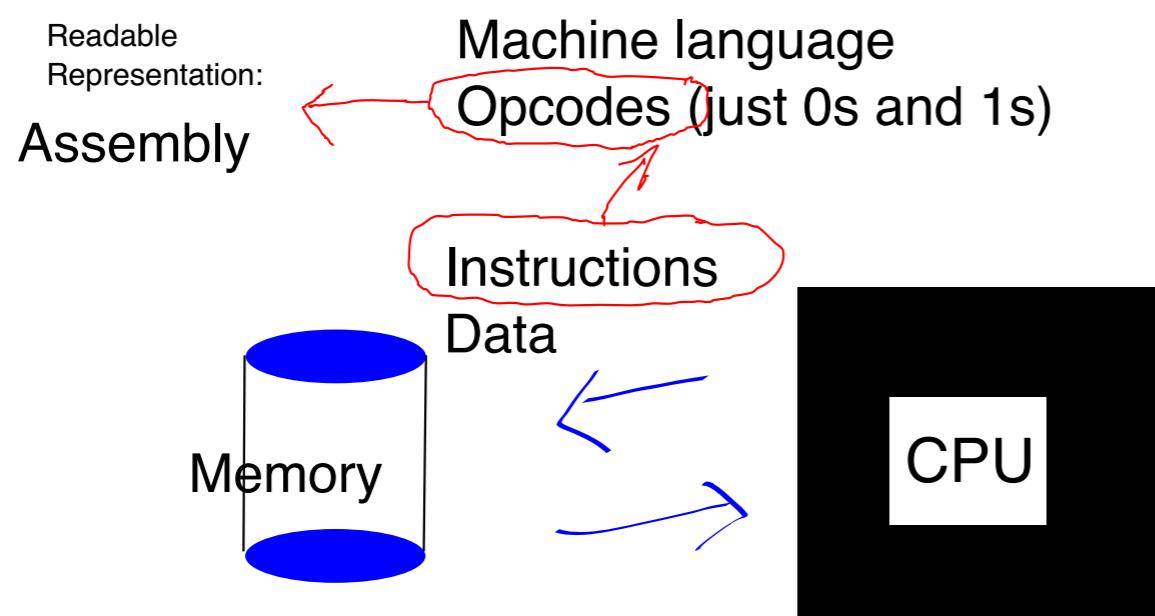
Many current CPUs use a
64 bit architecture.



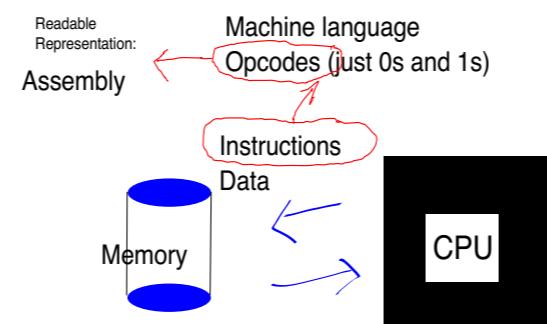


Machine language
Opcodes (just 0s and 1s)





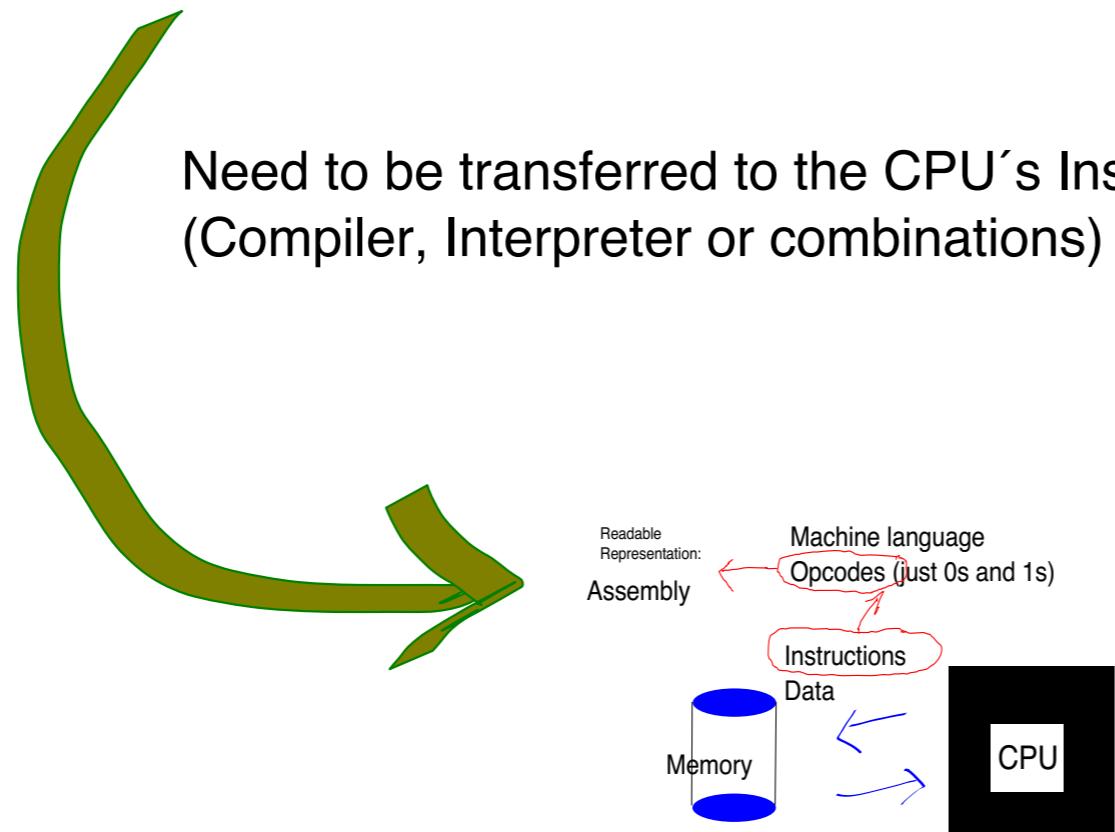
Higher Programming languages:
Python, C, C++, Java, Javascript, etc.



Higher Programming languages:

Python, C, C++, Java, Javascript, etc.

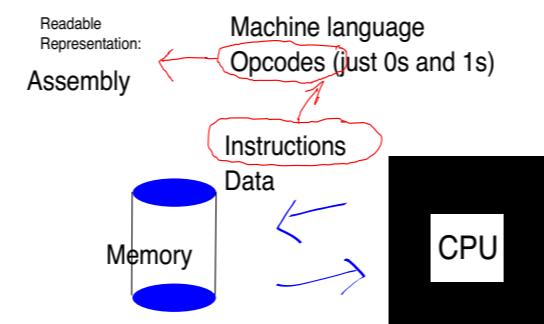
Need to be transferred to the CPU's Instruction Set
(Compiler, Interpreter or combinations)



Higher Programming languages:

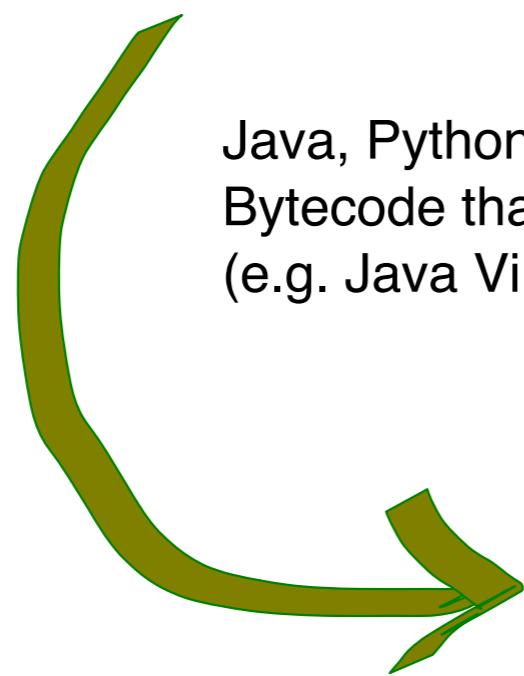
Python, C, C++, Java, Javascript, etc.

Executable Code only runs on the target architecture
with the right instruction set

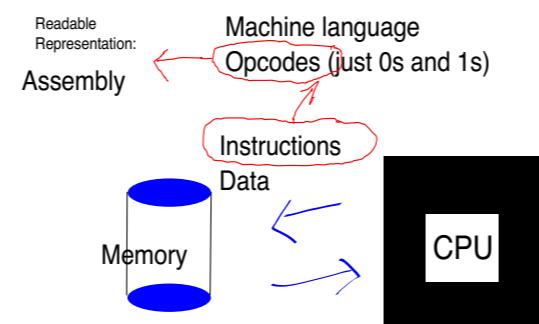


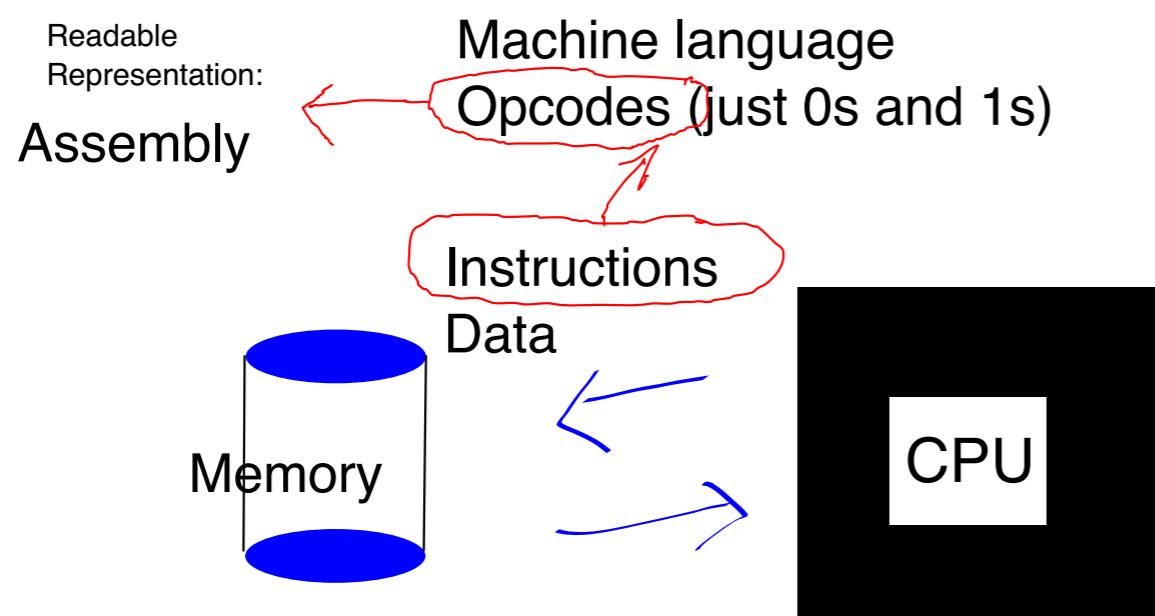
Higher Programming languages:

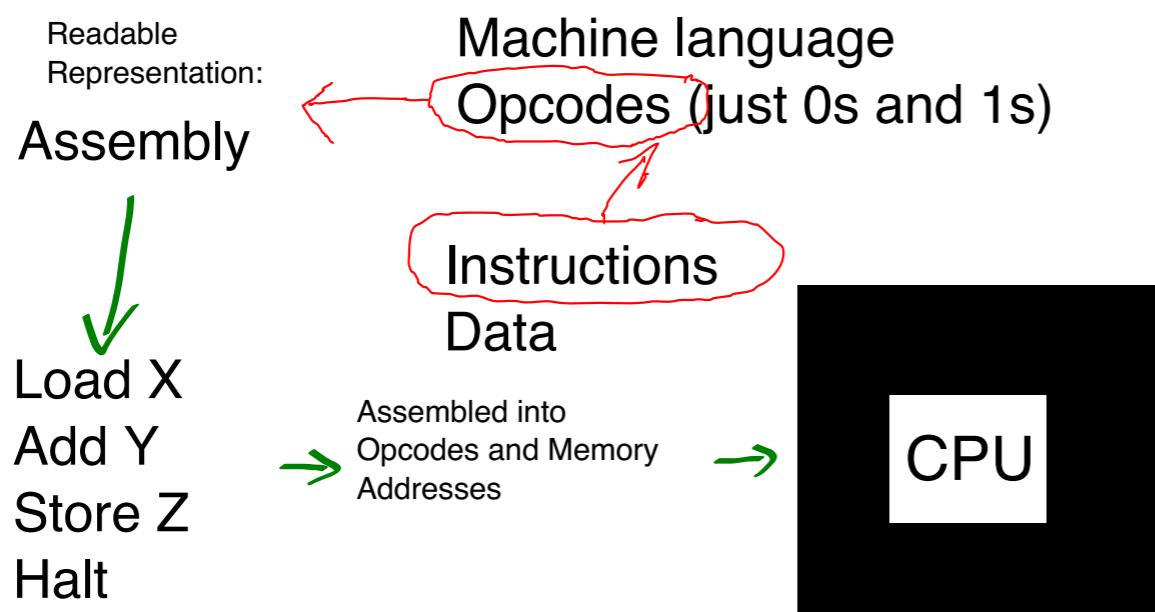
Python, C, C++, Java, Javascript, etc.



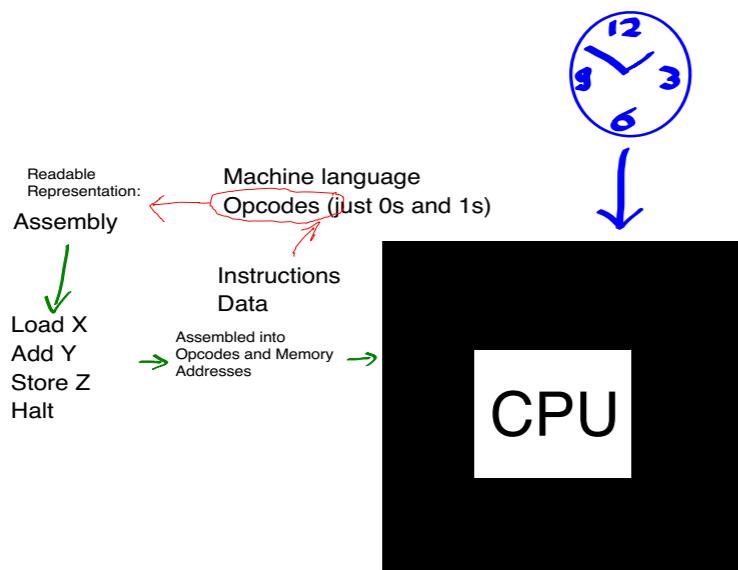
Java, Python can be compiled to platform-independent
Bytecode that needs another software to run
(e.g. Java Virtual Machine JVM)



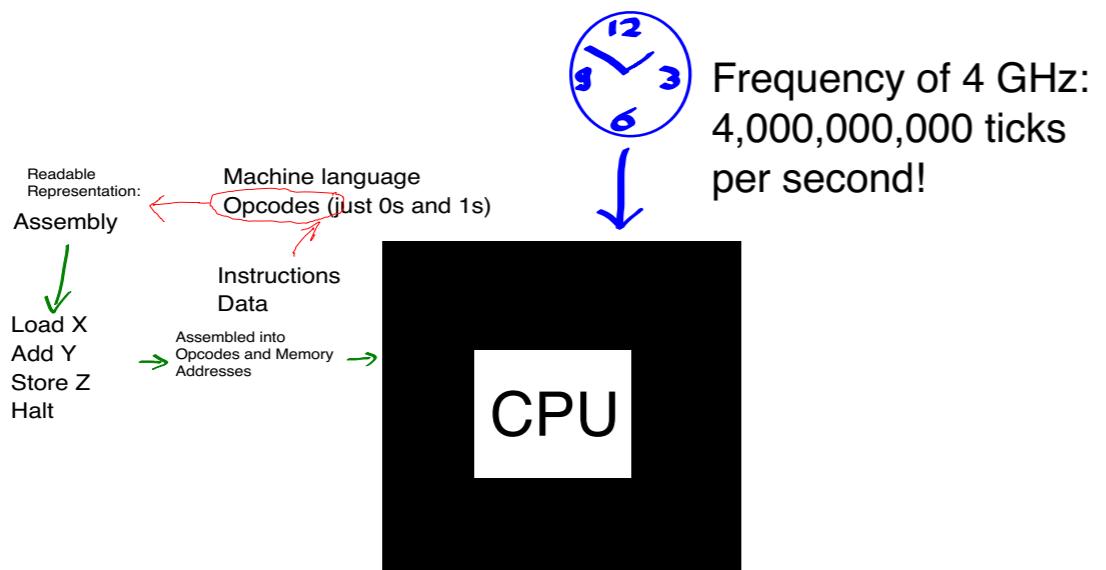




How are Instructions implemented in the CPU?

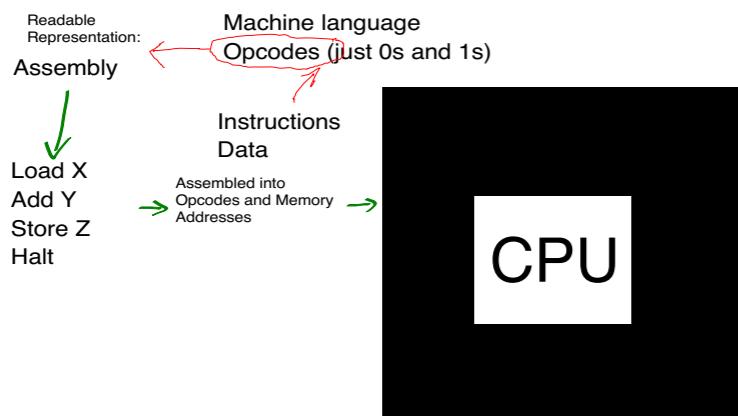


How are Instructions implemented in the CPU?



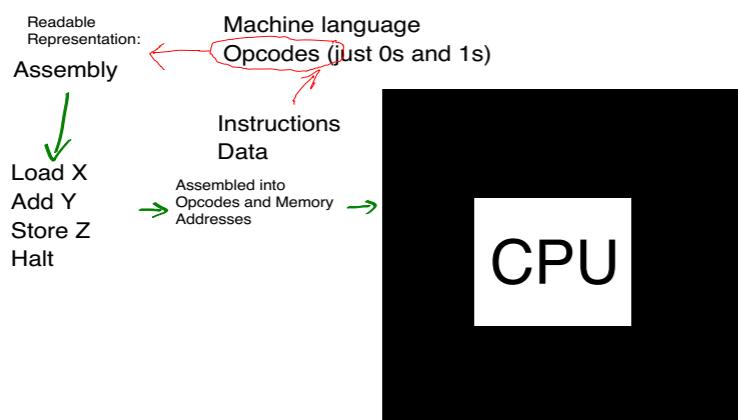
How are Instructions implemented in the CPU?

* One instruction needs several
Clock cycles

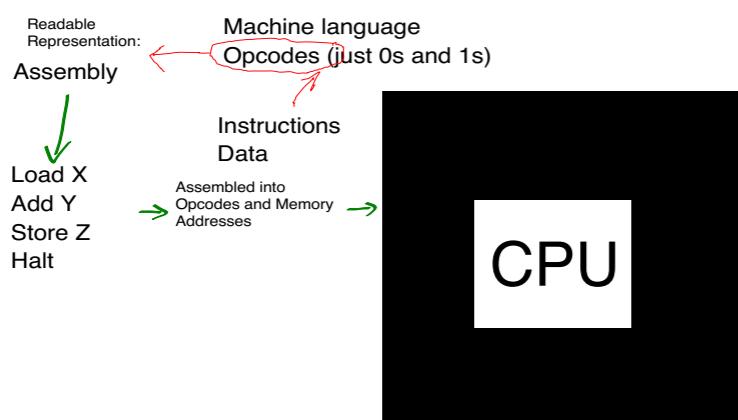


How are Instructions implemented in the CPU?

- * One instruction needs several Clock cycles
- * CPU works mainly on data in registers

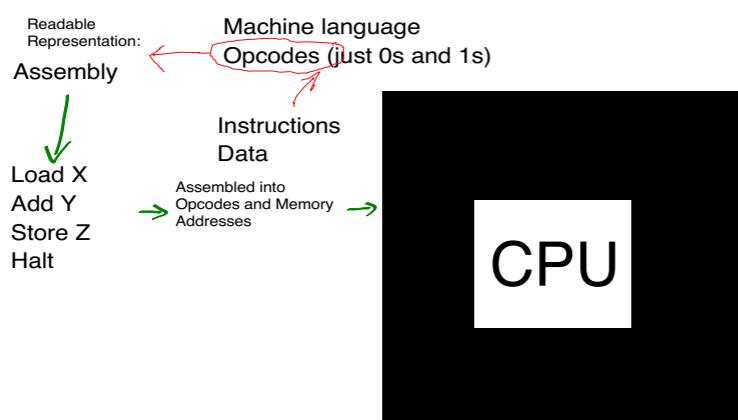


How are Instructions implemented in the CPU?



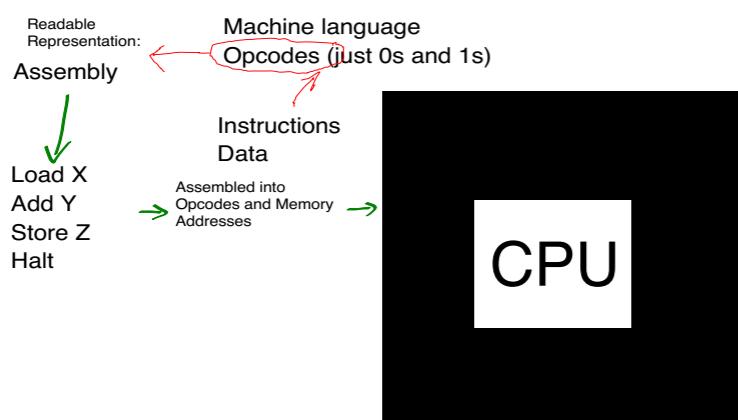
- * One instruction needs several Clock cycles
- * CPU works mainly on data in registers
- * Register Transfer Language (RTL) is used to define what needs to be done

How are Instructions implemented in the CPU?



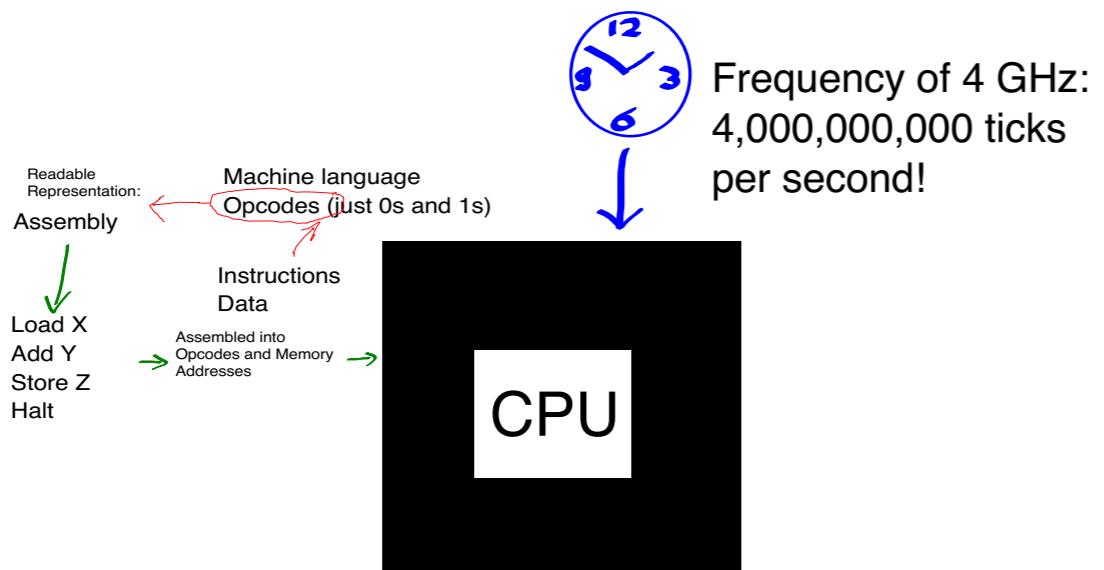
- * One instruction needs several Clock cycles
- * CPU works mainly on data in registers
- * Register Transfer Language (RTL) is used to define what needs to be done
- * Fetch-decode-execute cycle

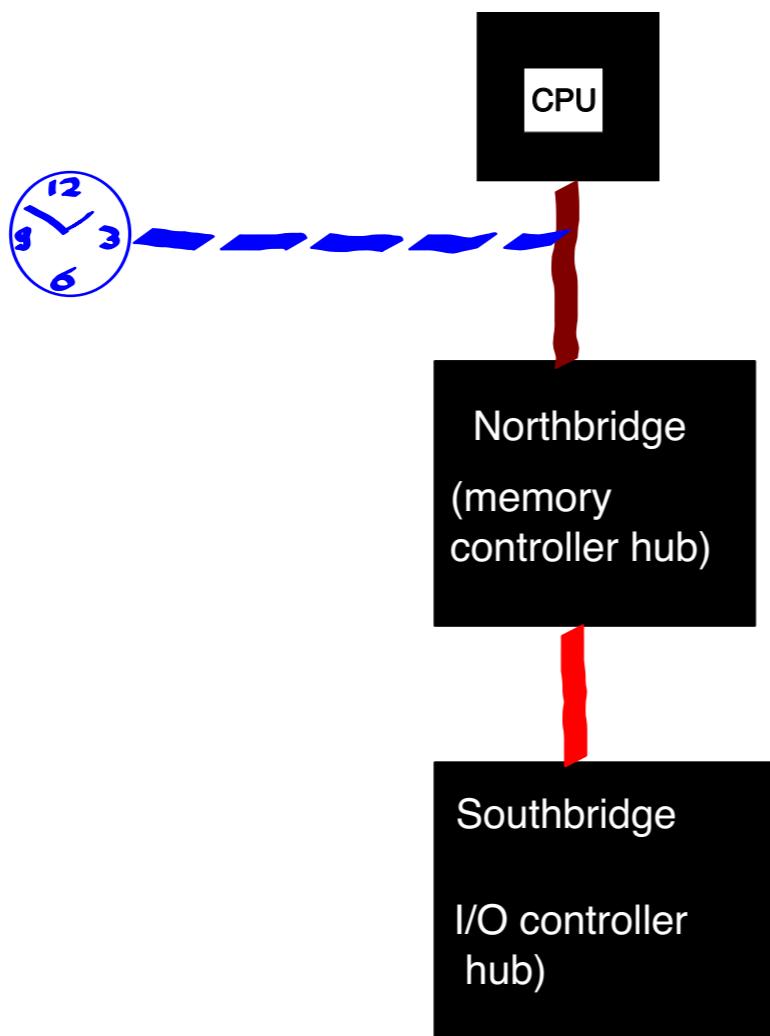
How are Instructions implemented in the CPU?

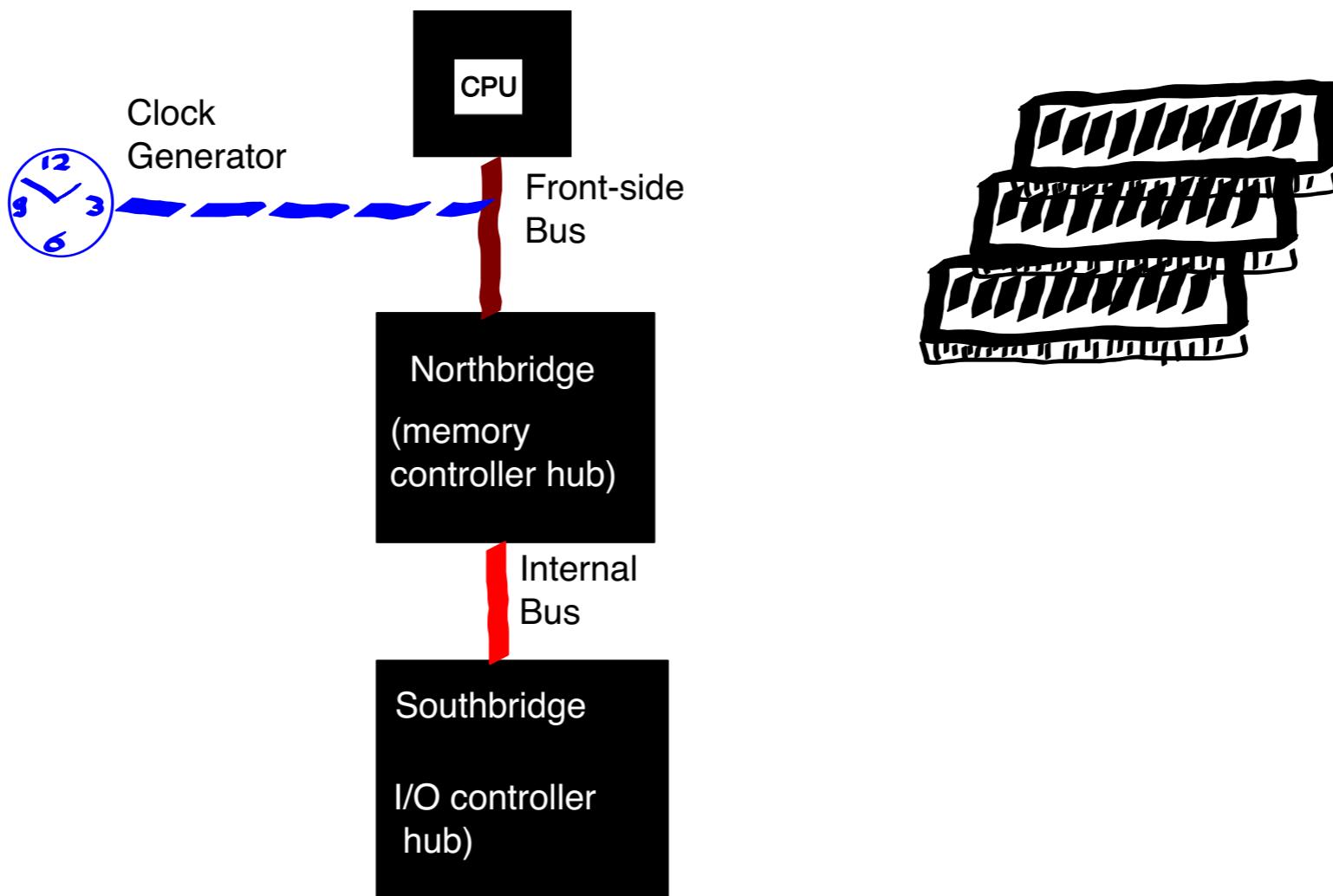


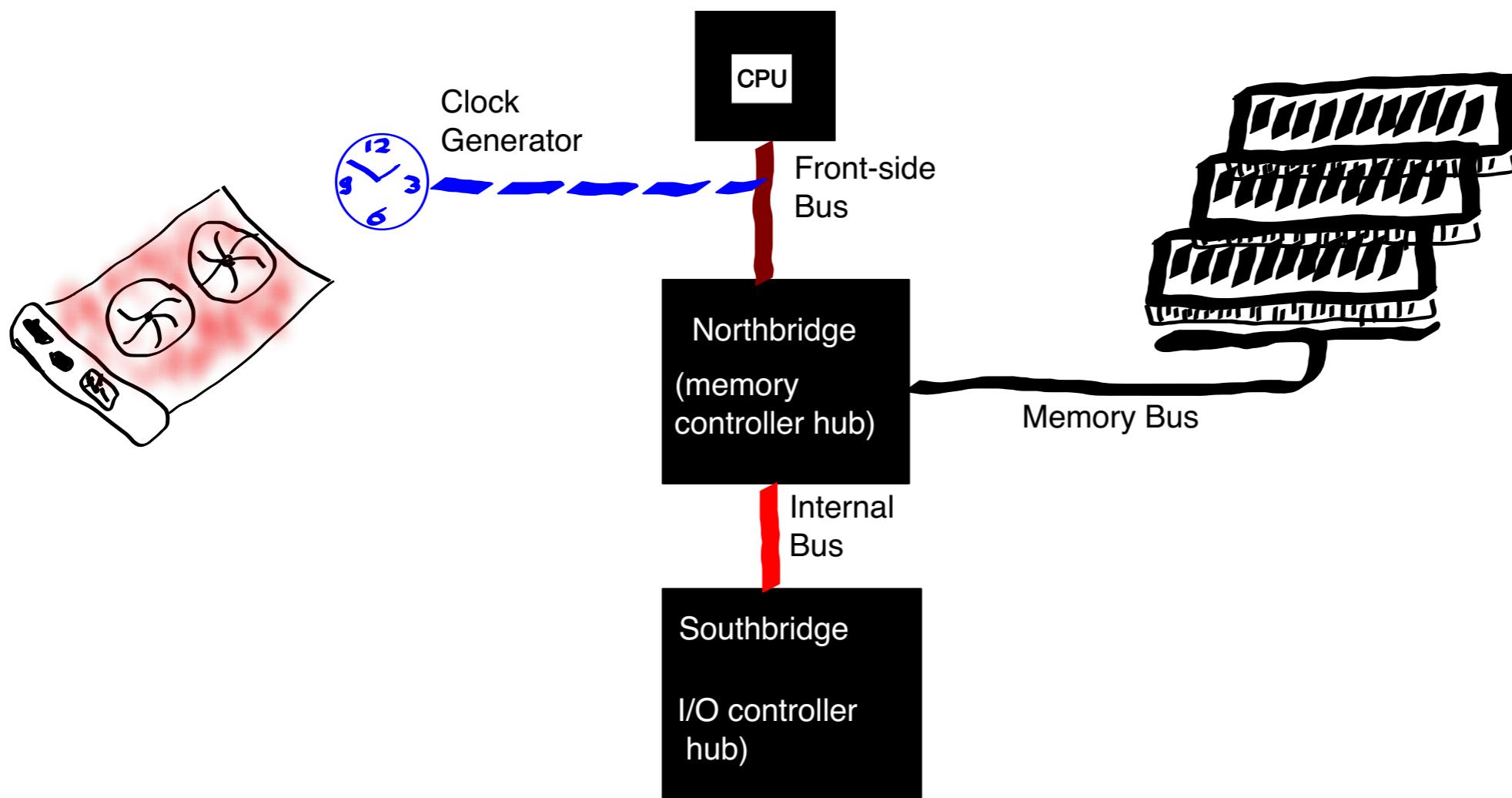
- * One instruction needs several Clock cycles
- * CPU works mainly on data in registers
- * Register Transfer Language (RTL) is used to define what needs to be done
- * Fetch-decode-execute cycle
- * CPU contains lots of circuits doing arithmetics (adders, etc.)

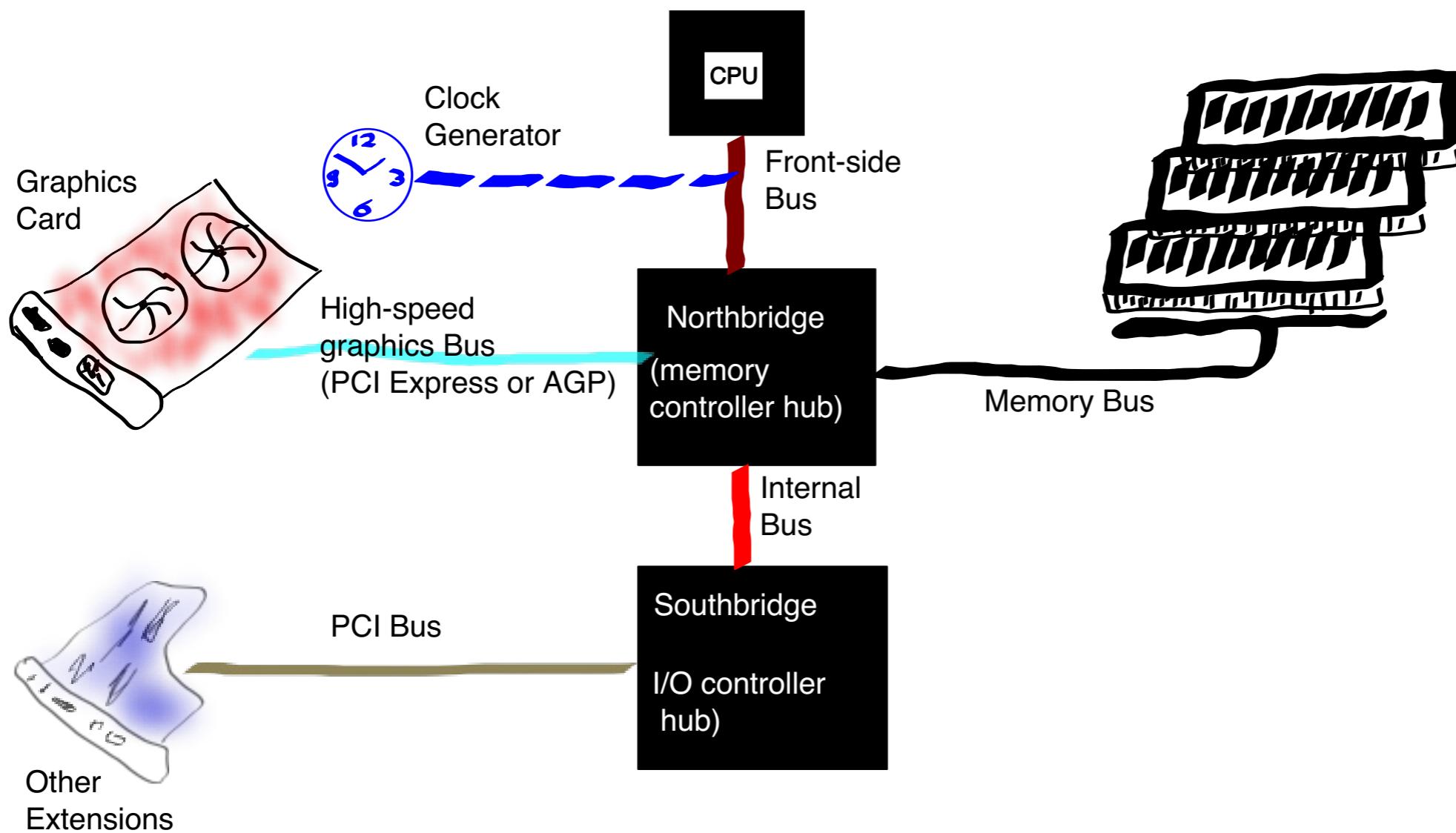
How are Instructions implemented in the CPU?

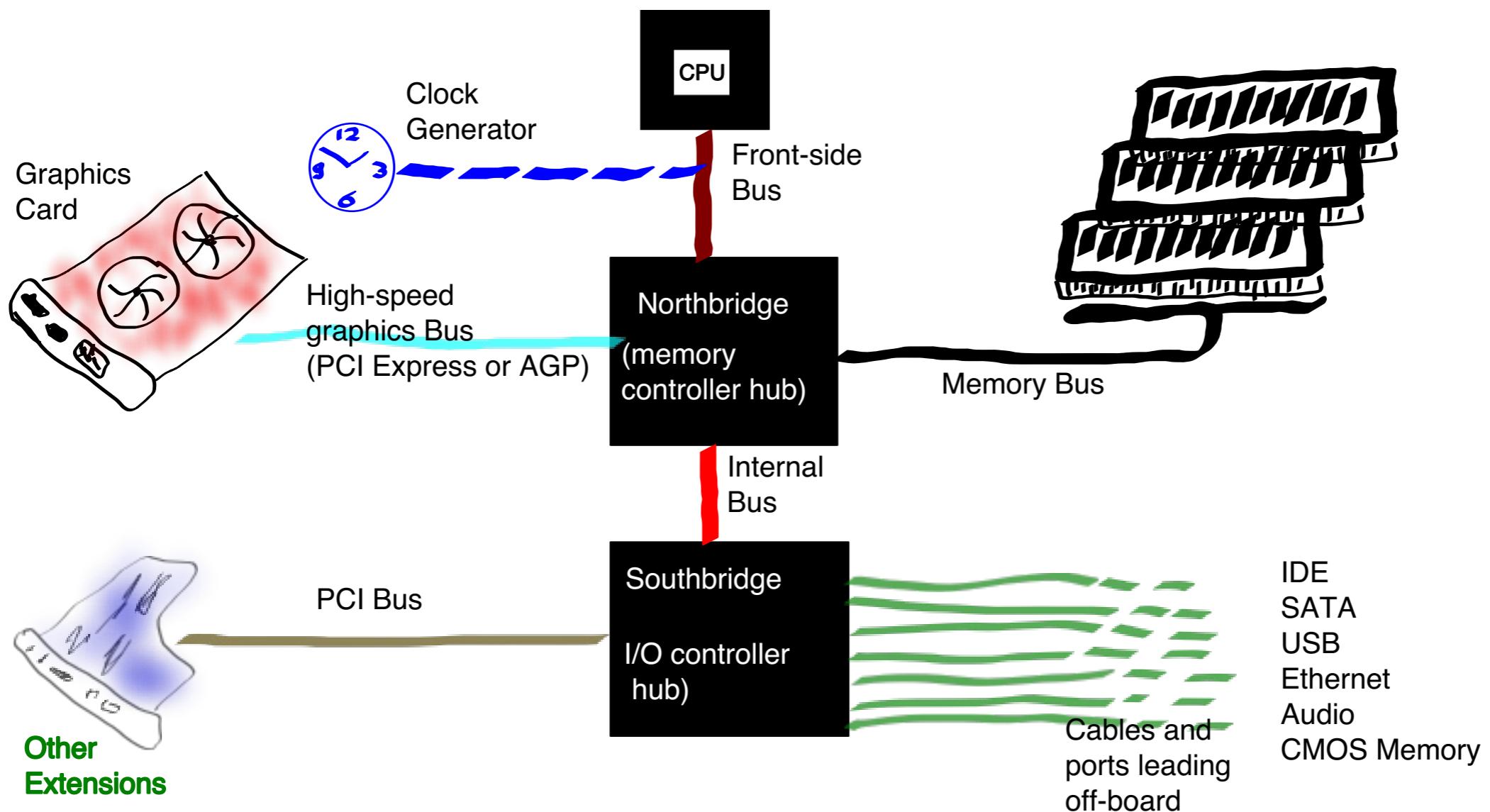


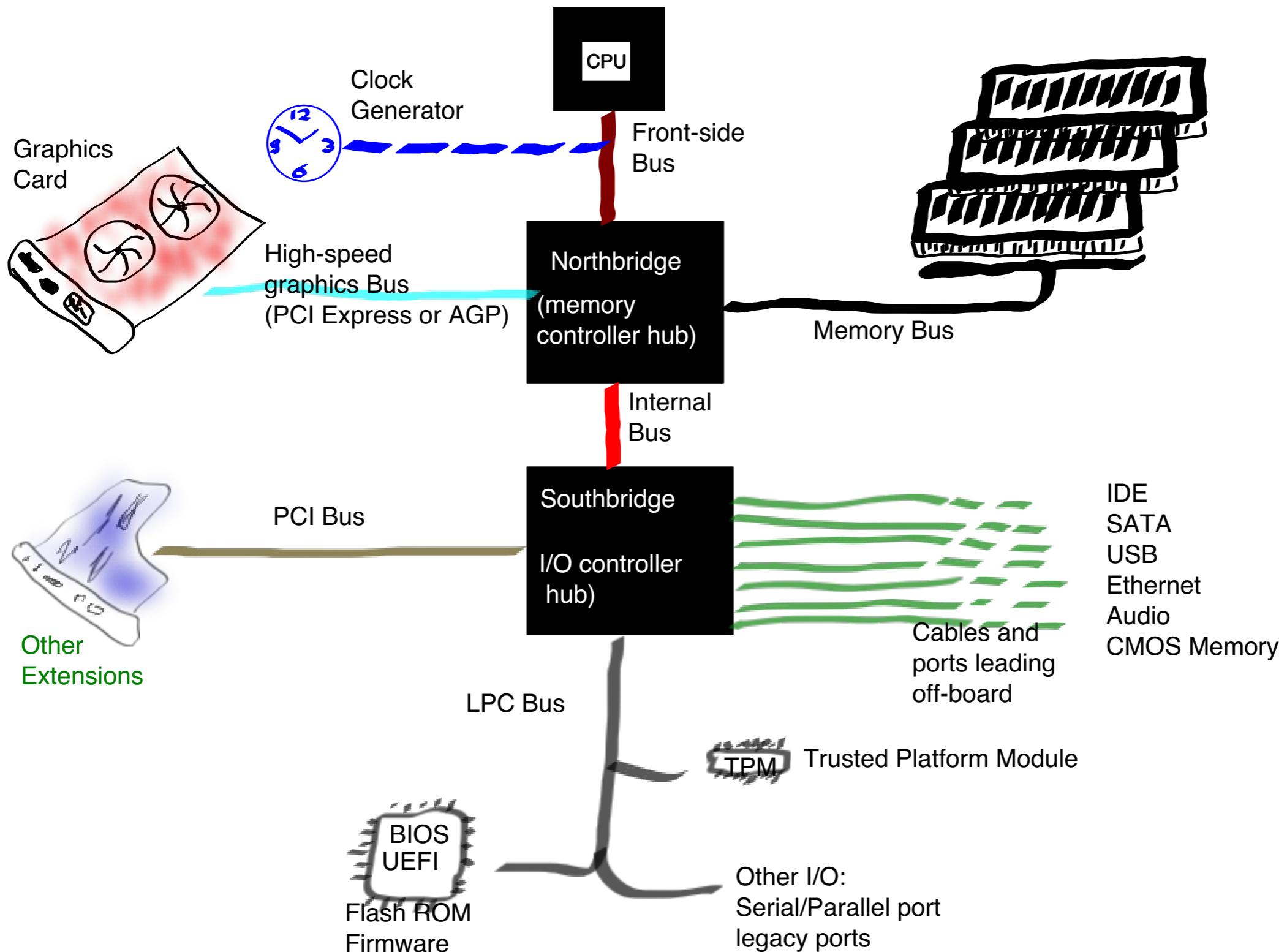


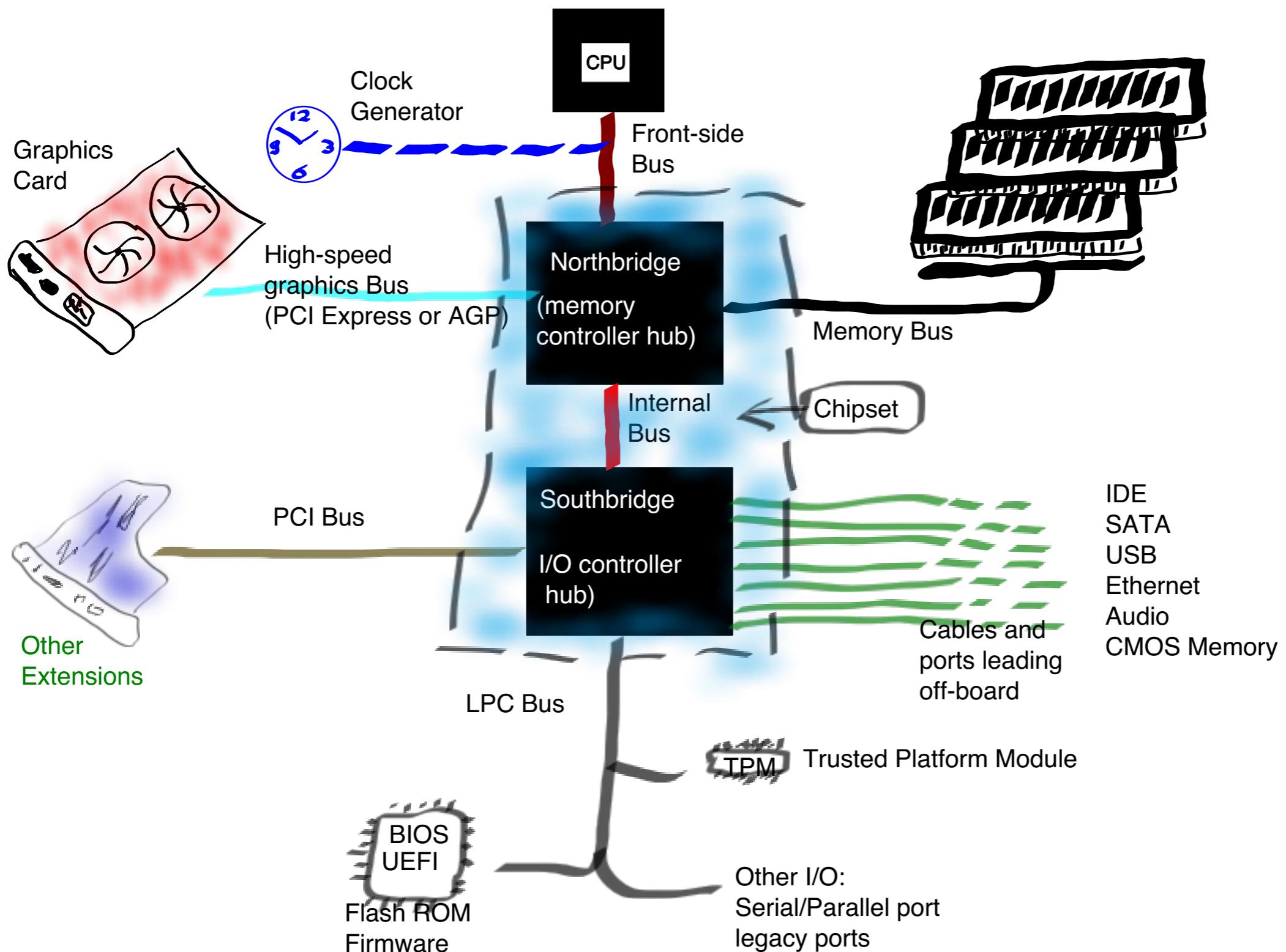












Motherboards

<http://www.computerhope.com/jargon/n/northbri.htm>
(contains a description and image of an ASUS
motherboard from around 2005)

Modern PCs

- Northbridge mostly integrated into CPU
 - faster, more control by CPU manufacturer
 - often contains GPU as well
- Rest done by Southbridge

Motherboards

<https://www.ifixit.com/Teardown/Retina+MacBook+2016+T>
(teardown of 2016 Apple MacBook)

System-on-Chip (SoC)

Integrate multiple components on a single chip:

- Processor
- Memory controller
- GPU
- RAM
- I/O interfaces
- Network

System-on-Chip (SoC)

Integrate multiple components on a single chip:

- low power consumption
- small form factor
- simplified motherboard layout

Used for smartphones, tablets, Raspberry Pi, etc.

Raspberry Pi

Complete computer based on SoC for around \$30

https://upload.wikimedia.org/wikipedia/commons/b/b4/Raspberry_Pi_B_rev_2_silicones.jpg

(the second chip in the picture is an Ethernet network adapter)

Next lecture

- Booting a PC
- BIOS and UEFI