METU Computer Engineering
METU Computer Engineering

Ceng 111 – Fall 2021
Week 4a

Digital Computation

Credit: Some slides are from the “Invitation to Computer Science” book by G. M. Schneider, J. L. Gersting and some from the “Digital Design” book by M. M. Mano and M. D. Ciletti.

Previously on CENG111!

METU Computer Engineering

The Control Unit

- Manages stored program execution
- Task
  1. Fetch from memory the next instruction to be executed
  2. Decode it: determine what is to be done
  3. Execute it: issue appropriate command to ALU, memory, and I/O controllers

Previously on CENG111!

Typical Machine Language Instruction Format

METU Computer Engineering

Previously on CENG111!

More on Instructions

Operation code	Address field 1	Address field 2	...

- LOAD X -> Load register R with the contents of memory cell X
- STORE X -> Store register R into memory cell X
- MOVE X, Y -> Copy the contents of X into Y
- ADD X -> Add contents of X to the contents of R
- ADD X,Y -> Add contents of X to the contents of Y, and put the result in register R
- COMPARE X, Y-> Set GT (greater than), EQ (equal) and LT (less than) condition codes
- JUMP X-> Jump unconditionally to the instruction in cell X
- JUMPGT X-> Jump, if GT=1, to the instruction in cell X
- HALT

Previously on CENG111!

Machine Language Instructions (continued)

- Types of machine instructions:
  - Data transfer
    - Move values to and from memory and registers
  - Arithmetic/logic
    - Perform ALU operations that produce numeric values
  - Compares
    - Set bits of compare register to hold result
  - Branches
    - Jump to a new memory address to continue processing

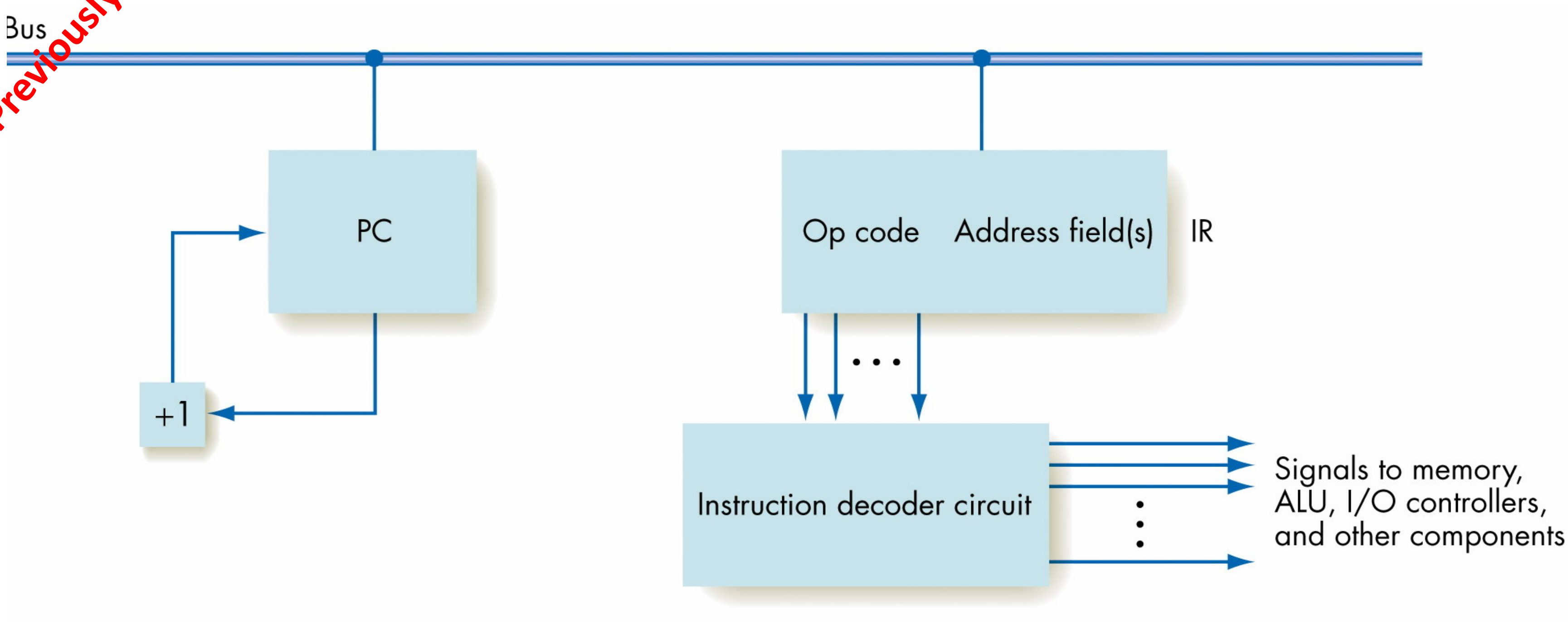
Previously on CENG111!

Address	Contents
100	Value of *a*
101	Value of *b*
102	Value of *c*

Algorithmic notation	Machine Language Instruction Sequences: Address	Contents	(Commentary)
		⋮	
1. Set *a* to the value $b + c$	50	LOAD 101	Put the value of *b* into register R.
	51	ADD 102	Add *c* to register R. It now holds $b + c$.
	52	STORE 100	Store the contents of register R into *a*.

Algorithmic notation	Address	Contents	(Commentary)
2. If $a > b$ then	50	COMPARE 100, 101	Compare *a* and *b* and set condition codes.
set *c* to the value *a*	51	JUMPGT 54	Go to location 54 if $a > b$.
Else	52	MOVE 101, 102	Get here if $a \le b$, so move *b* into *c*
set *c* to the value *b*	53	JUMP 55	and skip the next instruction.
	54	MOVE 100, 102	Move *a* into *c*.
	55	. . .	Next statement begins here.

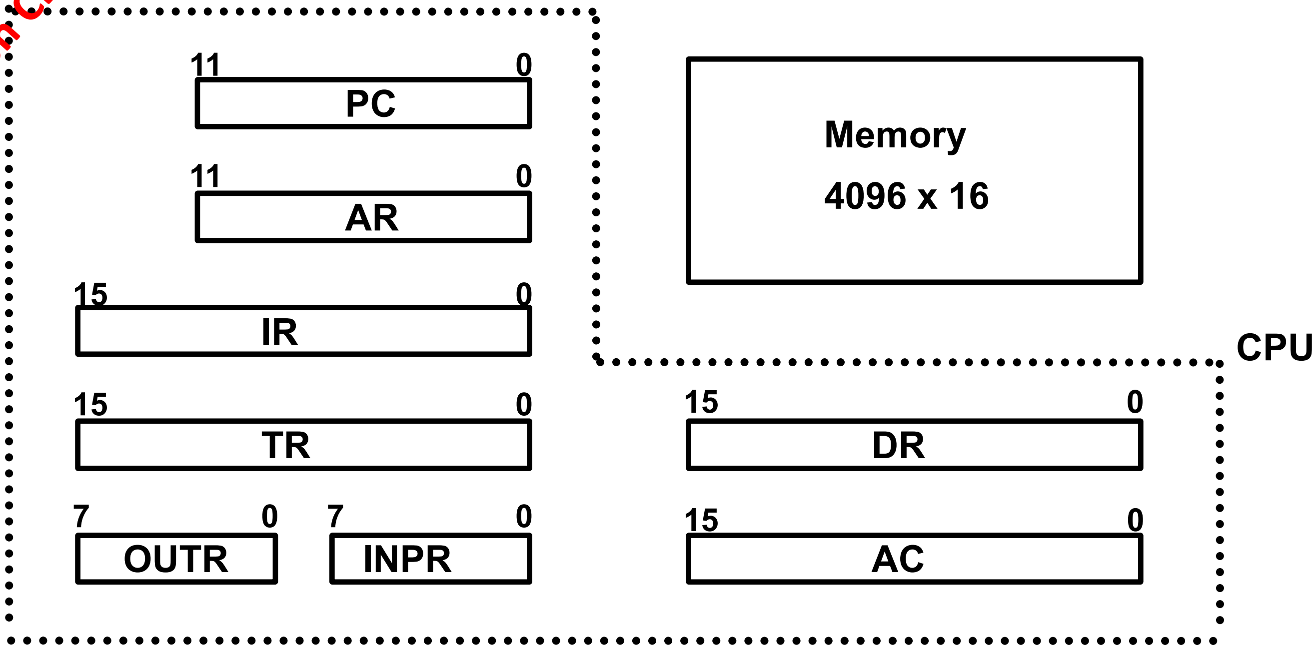
Previously on CENG111!



Organization of the Control Unit Registers and Circuits

Previously on CENG111!

Registers in a Basic Computer



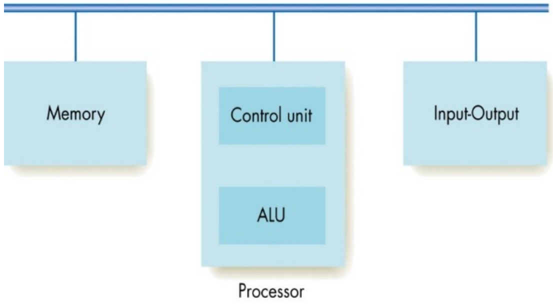
List of BC Registers

DR	16	Data Register	Holds memory operand
AR	12	Address Register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction Register	Holds instruction code
PC	12	Program Counter	Holds address of instruction
TR	16	Temporary Register	Holds temporary data
INPR	8	Input Register	Holds input character
OUTR	8	Output Register	Holds output character

METU Computer Engineering

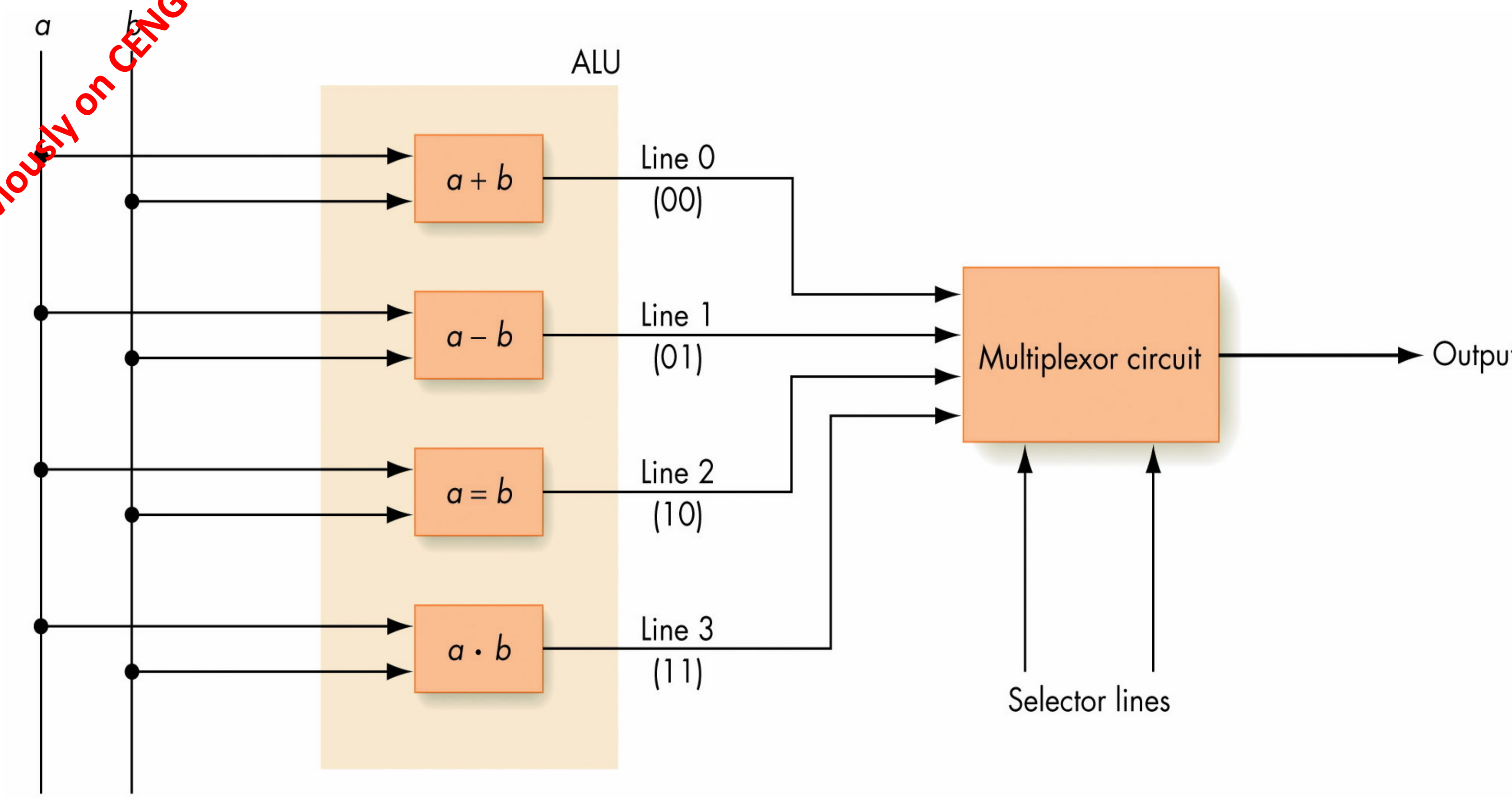
Previously on CENG111!

The Arithmetic/Logic Unit



- ***A**rithmetic and **L**ogic **U**nit*
  - *"Manufacturing" section of computer*
  - *Contains decision mechanisms and can make calculations+comparisons*
  - Actual computations are performed
- Primitive operation circuits
  - Arithmetic *[+, -, *, /]*
  - Comparison *[equality or CE, GT, LT, NEQ]*
  - Logic *[AND, OR, NOT, XOR]*
- Data inputs and results stored in registers
- Multiplexer selects desired output

Previously on CENG111!



Using a Multiplexor Circuit to Select the Proper ALU Result

(Not totally correct)

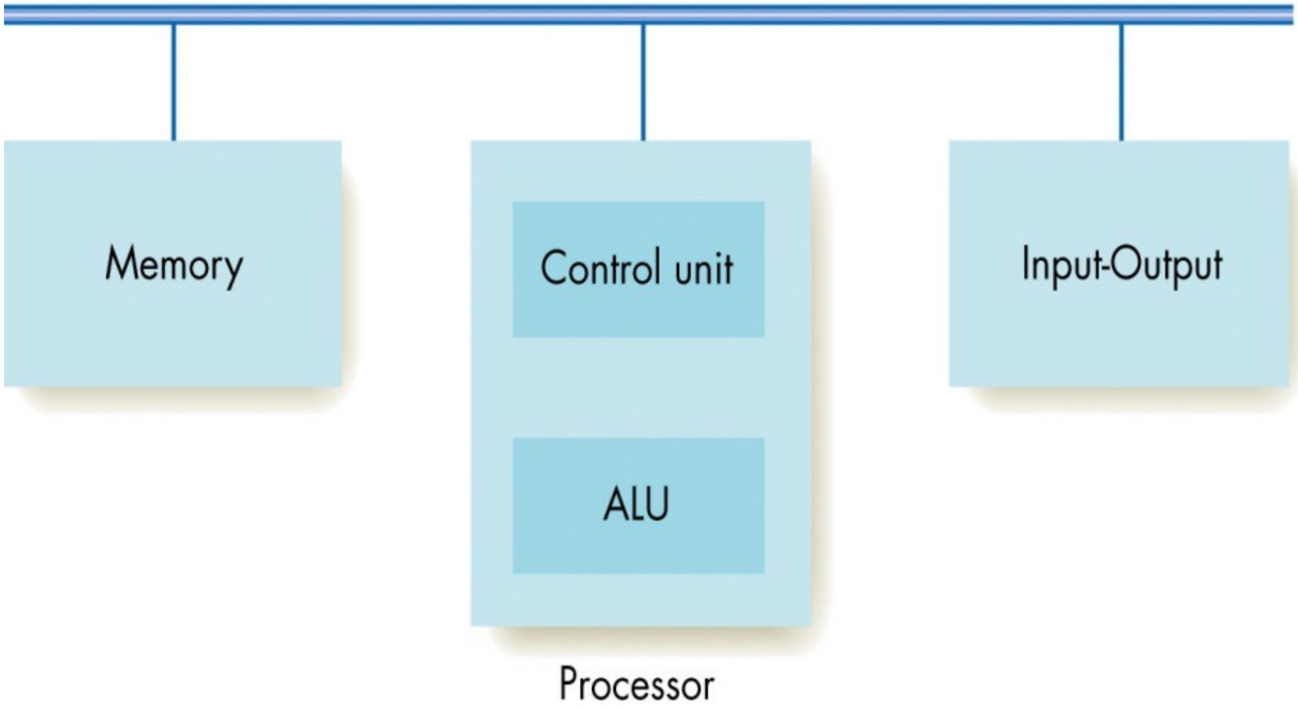
Today

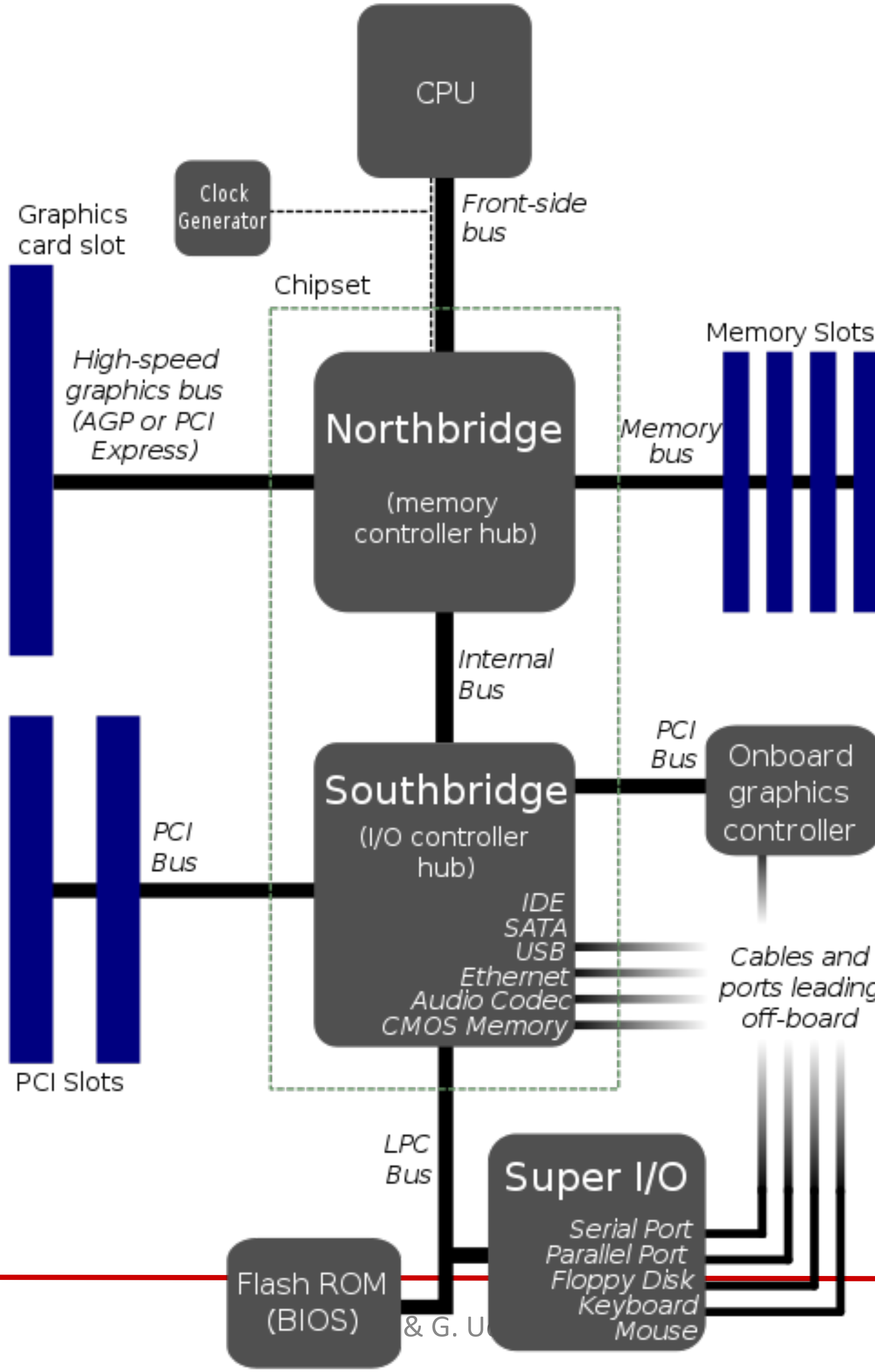
- Von Neumann Architecture
  - Pros and cons
  - The future
- More on memory
- Peripherals
- Interrupts
- Booting a computer

Administrative Notes

- THE1 announced!
  - Exam this Friday
- Tentative midterm date:
  - 22 December, Wednesday, 18:00

MORE ON THE VON NEUMANN ARCHITECTURE


Memory
Control unit
ALU
Processor
Input-Output

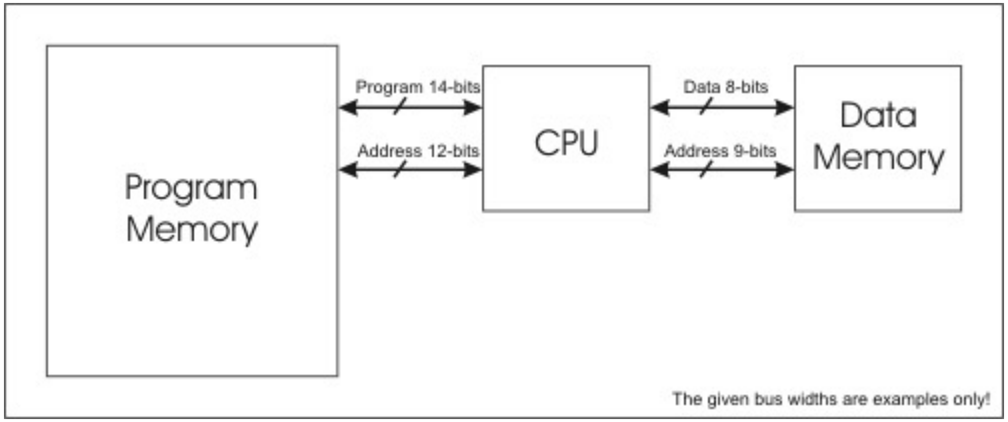

CPU
Clock Generator
Front-side bus
Graphics card slot
Chipset
High-speed graphics bus (AGP or PCI Express)
Northbridge
(memory controller hub)
Memory Slots
Memory bus
Internal Bus
PCI Bus
Onboard graphics controller
Southbridge
(I/O controller hub)
IDE
SATA
USB
Ethernet
Audio Codec
CMOS Memory
Cables and ports leading off-board
PCI Bus
PCI Slots
LPC Bus
Super I/O
Serial Port
Parallel Port
Floppy Disk
Keyboard
Mouse
Flash ROM (BIOS)

Von Neumann Architecture

- **Pros:**
  - Simplifies hardware (both circuit-design and layout)
  - Easier to generate re-locatable code, which makes multi-tasking easier to implement.
- **Cons:**
  - Instructions must be multiples of the data bus-width - can be inefficient.
  - Variable number of cycles required for instructions. For example, an instruction that requires data from memory must wait at least another cycle before it can complete, whereas some instructions execute much faster. This can be a problem for time-critical applications.

http://www.mhennessy.f9.co.uk/pic/architecture.htm

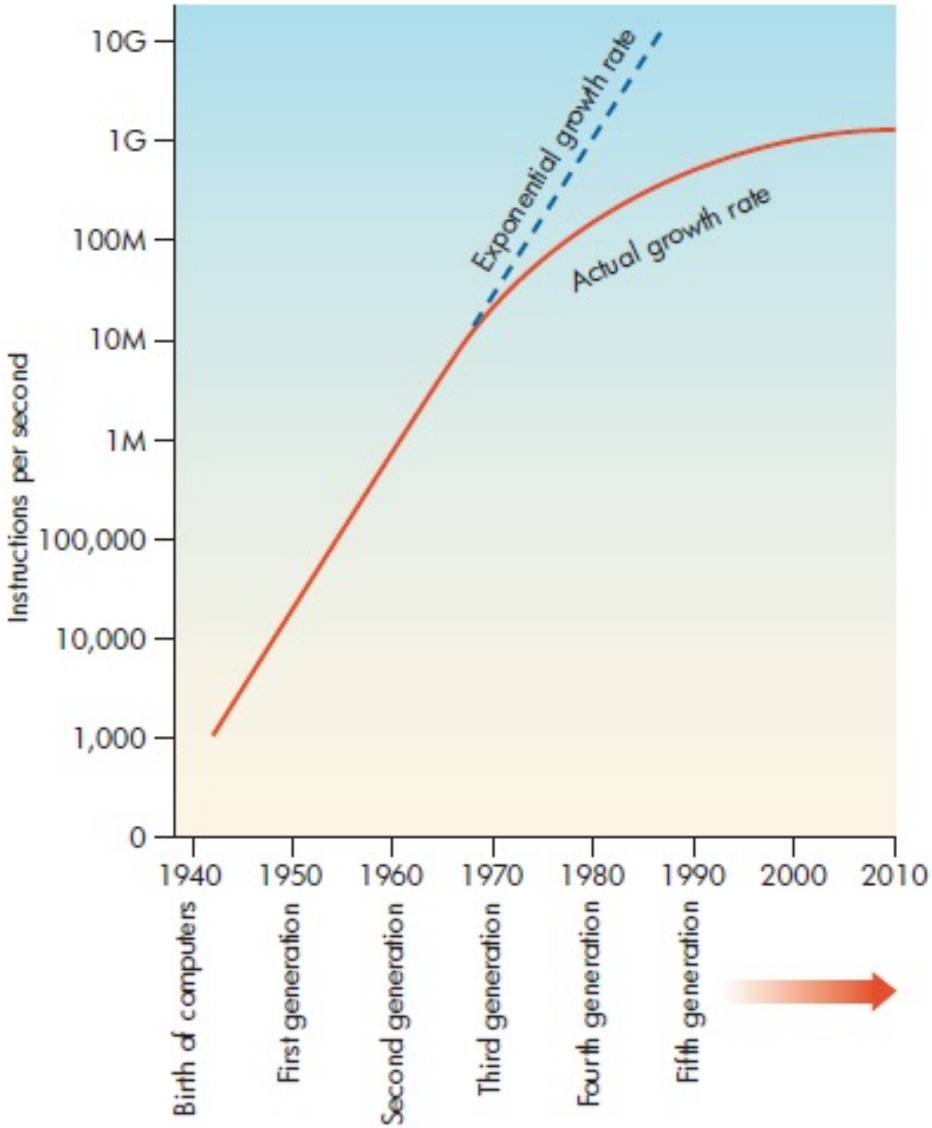
Harvard Architecture



http://www.mhennessy.f9.co.uk/pic/architecture.htm

- **Pros:**
  - Data and address busses can be of different widths. This means the program memory word can be wide enough to incorporate an instruction and a literal (fixed data) in a single instruction.
  - A built-in two-stage pipeline overlaps fetch and execution of instructions, meaning most instructions execute in a single clock cycle.
- **Cons:**
  - Slightly more confusing.
  - Hardware is more complicated.

The Future



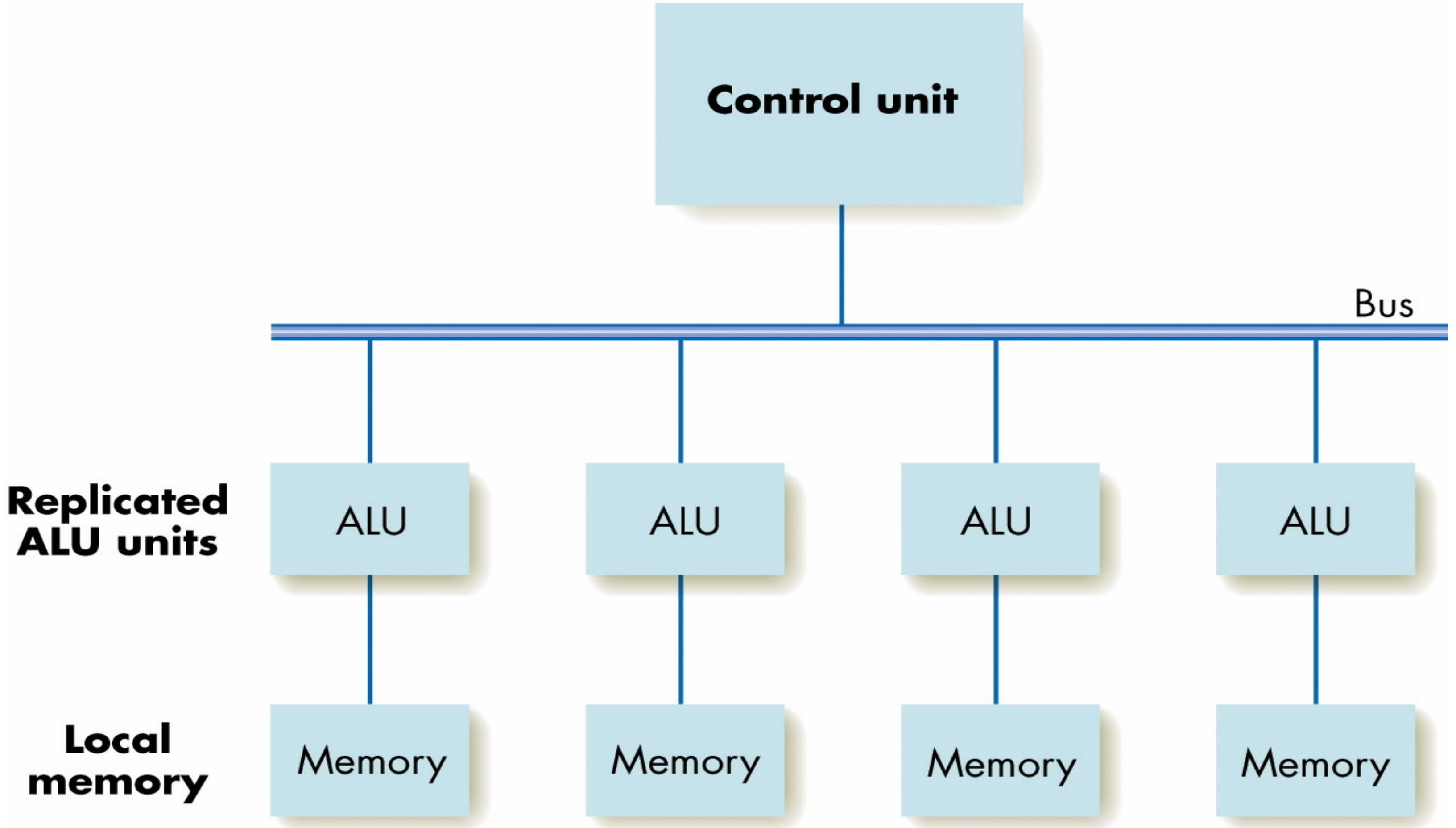
10G
1G
100M
10M
1M
100,000
10,000
1,000
0
Instructions per second
Exponential growth rate
Actual growth rate
1940
1950
1960
1970
1980
1990
2000
2010
Birth of computers
First generation
Second generation
Third generation
Fourth generation
Fifth generation

The Future

- Physical limitations on the speed of Von Neumann computers
- Non-Von Neumann architectures explored to bypass these limitations
- Parallel computing architectures can provide improvements:
    - multiple operations occur at the same time

The Future

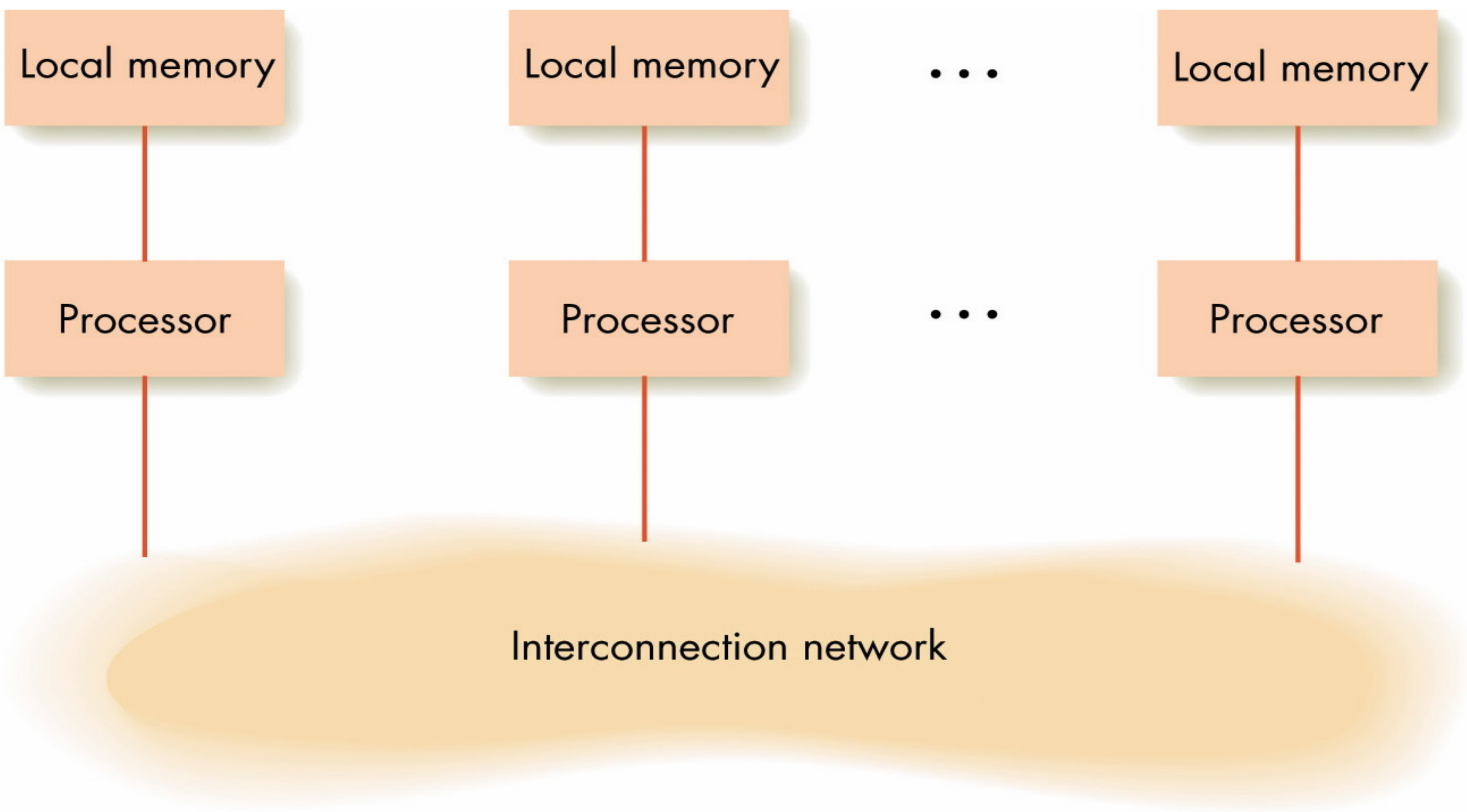
- SIMD architecture
  - Single instruction/Multiple data
  - Multiple processors running in parallel
  - All processors execute same operation at one time
  - Each processor operates on its own data
  - Suitable for “vector” operations



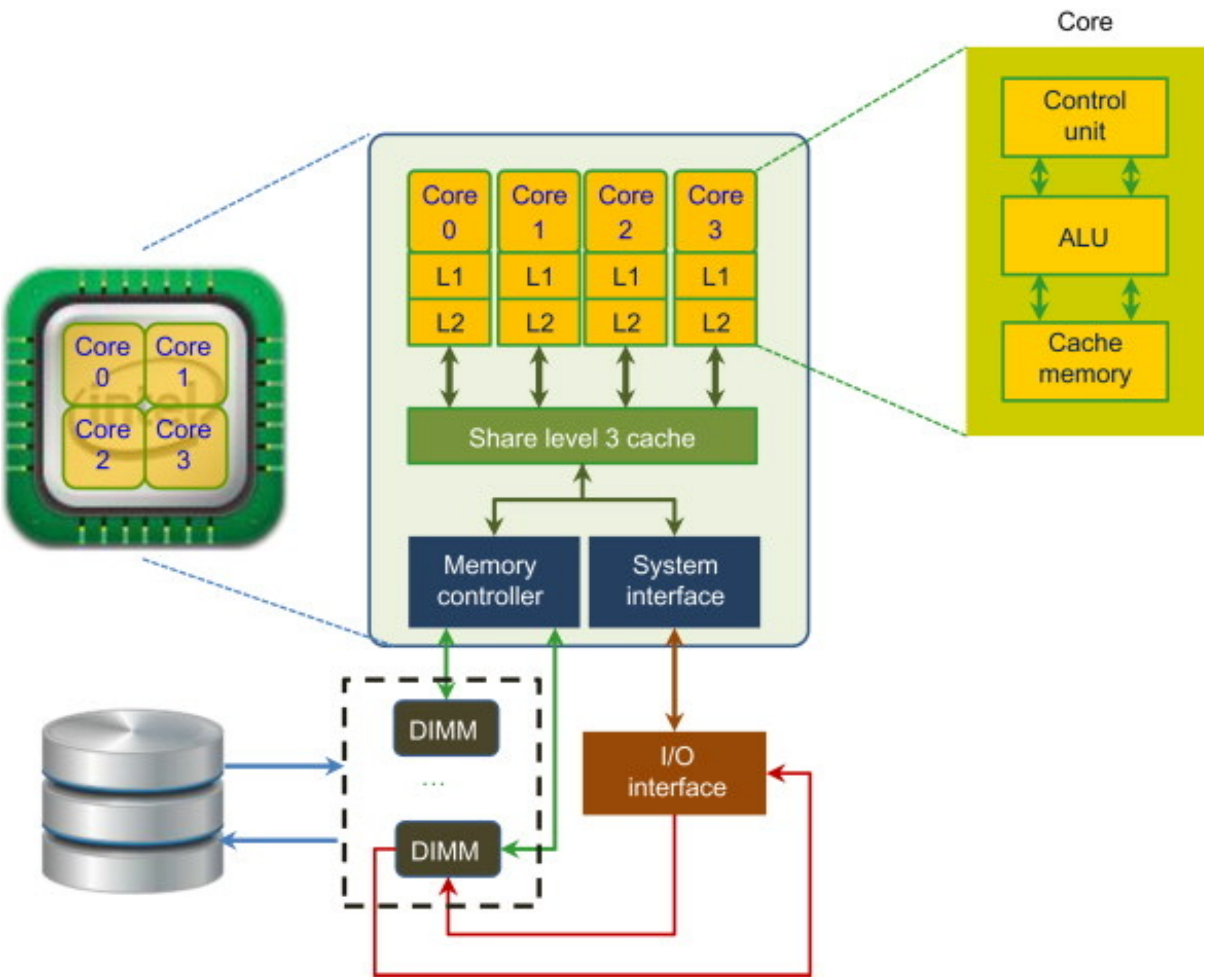
A SIMD Parallel Processing System

The Future

- MIMD architecture
  - Multiple instruction/Multiple data
  - Multiple processors running in parallel
  - Each processor performs its own operations on its own data
  - Processors communicate with each other



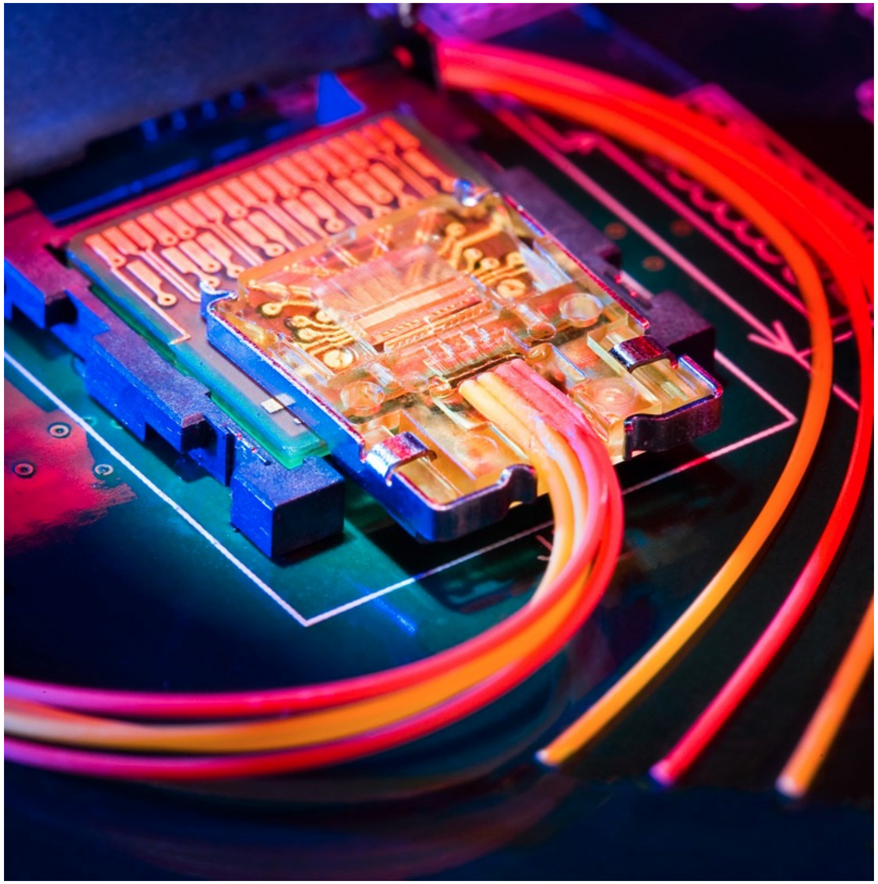
Model of MIMD Parallel Processing



https://www.sciencedirect.com/topics/computer-science/core-processor

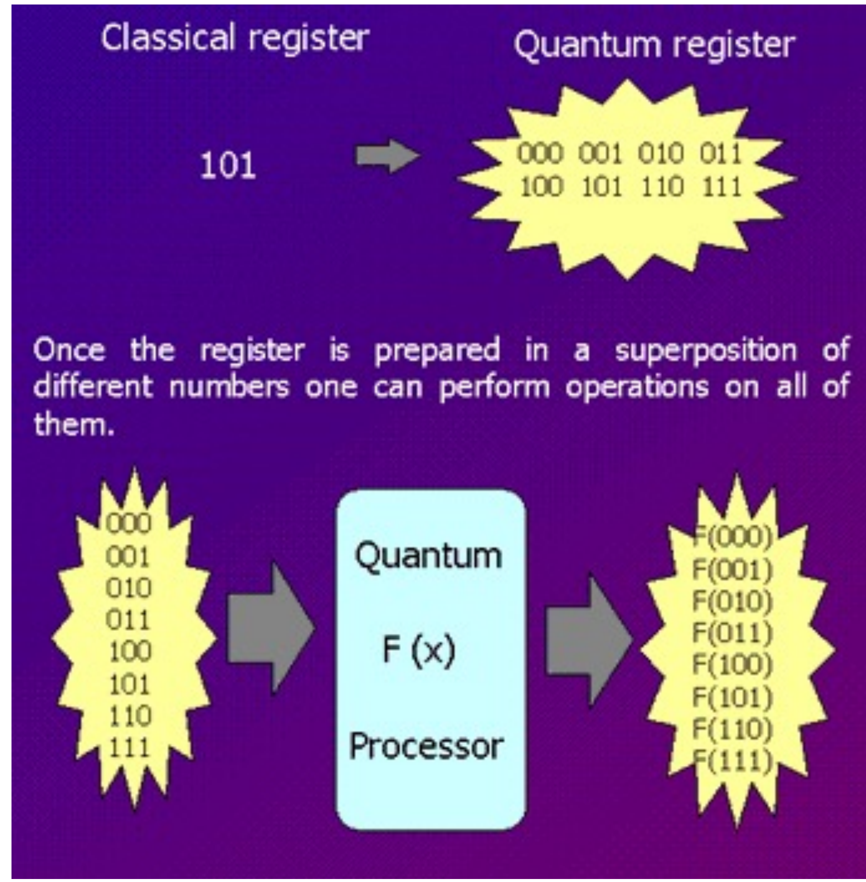
New Trend: Optical Computing

- Currently available for only data transfer.
- Work in progress towards "photonic logic" for designing circuits which use photons:
  - There are a lot of challenges and disadvantages



http://en.wikipedia.org/wiki/Optical_computing

New Trend: Quantum Computing



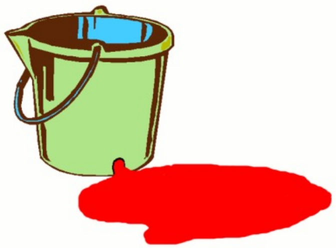
- There are several problems:
  - Decoherence: The more the qubits, the more their effect on the environment.

http://media.defenseindustrydaily.com/images/PUB_CQC_Cambridge_Quantum_Computing_Explained_lg.png

Read Especially This: http://www.cs.virginia.edu/~robins/The_Limits_of_Quantum_Computers.pdf

MORE ON MEMORY

Memory Types



- Dynamic Memory
  - The voltages stored in capacitors die away with time.
  - Solution?
    - Refresh the memory frequently; i.e., re-write the contents of the memory.
- Static Memory
  - A coupled transistor system stores the information.
  - One of the couples *triggers* the other.
- DRAM is cheaper and more widely used.

http://wiki.xtronics.com/index.php/How_Memory_Works

Memory Types

- Volatile Memory:
  - The stored values are lost when power is off.
  - DRAM, SRAM
- Non-volatile:
  - Read-only Memory (ROM), flash memory

http://wiki.xtronics.com/index.php/How_Memory_Works