ECSE 426
Microprocessor Systems

Zeljko Zilic

Room 536
McConnell Building
zeljko@ece.mcgill.ca
www.macs.ece.mcgill.ca/~zeljko

Microprocessors

- Enabling technology for general purpose computers and embedded systems
  - Really, lots&lots of things nowadays
- Foundation for software-intensive systems
- Data processor - arithmetic, logical, symbolic or application-specific operations
  - Architectural view: ALUs, registers, etc.
  - Circuit view: registers, interfaces, buses
  - Programmer's view: assembler instructions

Computer System Types

General Purpose Computers

PCs, Workstations, Mainframes, Supercomputers.

Embedded Systems (everything else)

Games, PDAs, Medical, Industrial, Aerospace, Military

Real-Time Systems

Computers and Applications

- Deciding factors:
  - Cost
  - Size
  - Power
  - Quantity

Type	Price ($)	Example application
Disposable computer	1	Greeting cards
Embedded computer	10	Watches, cars, appliances
Game computer	100	Home video games
Personal computer	1K	Desktop or portable computer
Server	10K	Network server
Collection of Workstations	100K	Departmental minisupercomputer
Mainframe	1M	Batch data processing in a bank
Supercomputer	10M	Long range weather prediction

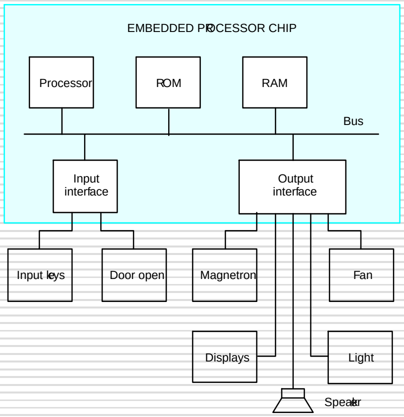
Embedded System Importance

- Ubiquitous processor-based control systems
- Development easier than with alternative technologies

- Makes products competitive: features AND price
- Enabling technology for many new products
- Likely source of jobs for ECE graduates

Embedded Systems

- They are just about everywhere
  - From toothbrush to space shuttle
- Incarnations of generic computer systems
  - Sometimes, specialized Input/Output

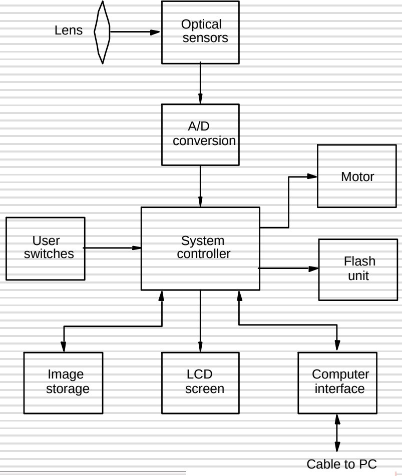


What is this diagram showing?

Microwave Oven

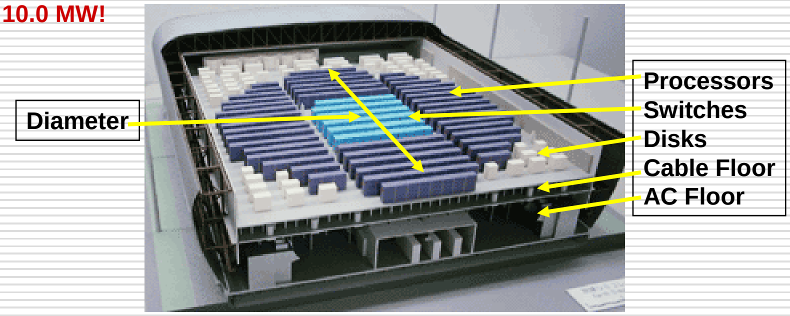
Another Example - Camera

- Computer system with:
  - Image control
  - Hardware (lenses, motors)
  - Interfaces
- Added sophistication to consumer electronics
  - Expandability (of functions)
  - Connectivity



Supercomputers: Earth Simulator

35.86 Tflop/s (#4), Footprint — 34,000 ft² (4 tennis courts x 3 floors)
10.0 MW!



Crossbar Interconnection Network
83000 Copper Cables
1800 Miles of Cable
http://www.es.jamstec.go.jp/esc/eng/index.html

⟷

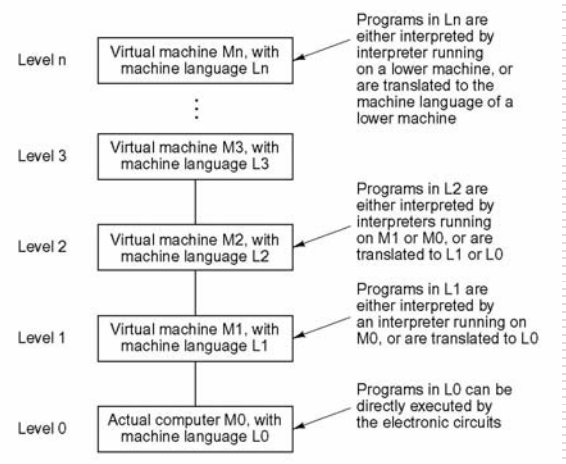
High Interprocessor Latency
(11 in = 1ns)

Views of Computer Systems

- Levels of abstraction
  - Logic Level - Circuits
    - Logic functions implemented by gates
  - Architectural Level - Microarchitecture
    - Operations performed by resources
  - Instruction Set Level - Instructions
    - Program execution
  - Operating System Level - Complete system
    - System operation

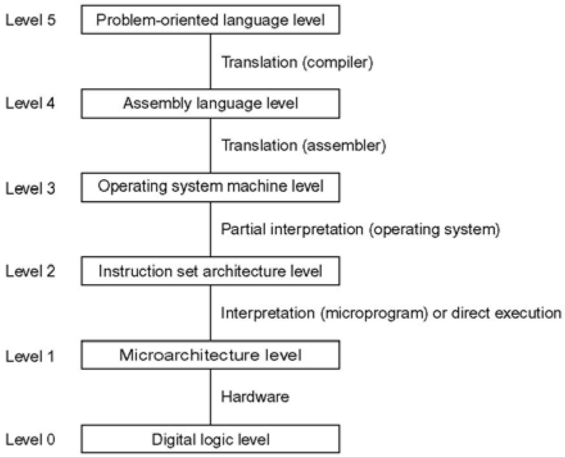
Layered Computer Architecture

- Concept necessary for complex systems
  - e.g., networking, large software systems etc.



Layered Computer Systems

- Includes hardware and software
  - User programs
  - Operating system
  - Instruction Set
  - Architecture
  - Hardware


Level 5 Problem-oriented language level

Translation (compiler)

Level 4 Assembly language level

Translation (assembler)

Level 3 Operating system machine level

Partial interpretation (operating system)

Level 2 Instruction set architecture level

Interpretation (microprogram) or direct execution

Level 1 Microarchitecture level

Hardware

Level 0 Digital logic level

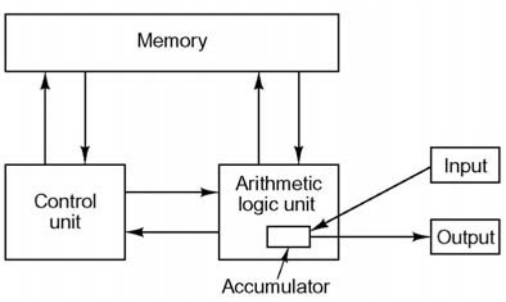
Computer History

- WWII effort
  - UK
  - (USA)
- Post-WWII
- Commercial development
  - IBM
  - DEC
  - Cray
  - Sun

Year	Name	Made by	Comments
1834	Analytical Engine	Babbage	First attempt to build a digital computer
1936	Z1	Zuse	First working relay calculating machine
1943	COLOSSUS	British gov't	First electronic computer
1944	Mark I	Aiken	First American general-purpose computer
1946	ENIAC I	Eckert/Mauchley	Modern computer history starts here
1949	EDSAC	Wilkes	First stored-program computer
1951	Whirlwind I	M.I.T.	First real-time computer
1952	IAS	Von Neumann	Most current machines use this design
1960	PDP-1	DEC	First minicomputer (50 sold)
1961	1401	IBM	Enormously popular small business machine
1962	7094	IBM	Dominated scientific computing in the early 1960s
1963	B5000	Burroughs	First machine designed for a high-level language
1964	360	IBM	First product line designed as a family
1964	6600	CDC	First scientific supercomputer
1965	PDP-8	DEC	First mass-market minicomputer (50,000 sold)
1970	PDP-11	DEC	Dominated minicomputers in the 1970s
1974	8080	Intel	First general-purpose 8-bit computer on a chip
1974	CRAY-1	Cray	First vector supercomputer
1978	VAX	DEC	First 32-bit superminicomputer
1981	IBM PC	IBM	Started the modern personal computer era
1985	MIPS	MIPS	First commercial RISC machine
1987	SPARC	Sun	First SPARC-based RISC workstation
1990	RS6000	IBM	First superscalar machine

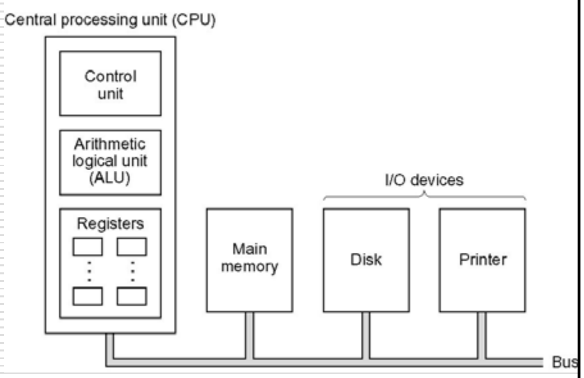
Von Neumann Machine

- Princeton IAS
  - 40-bit memory word
  - 20-bit instruction word
    - Loads “left” and “right” instruction at once



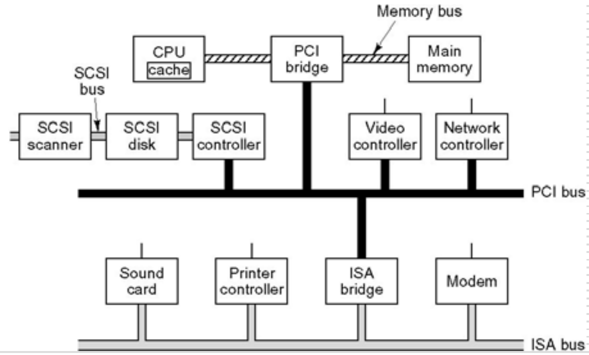
Computer Organization

- Processor
  - Microprocessor
- Memory
- Peripherals
- Common Bus



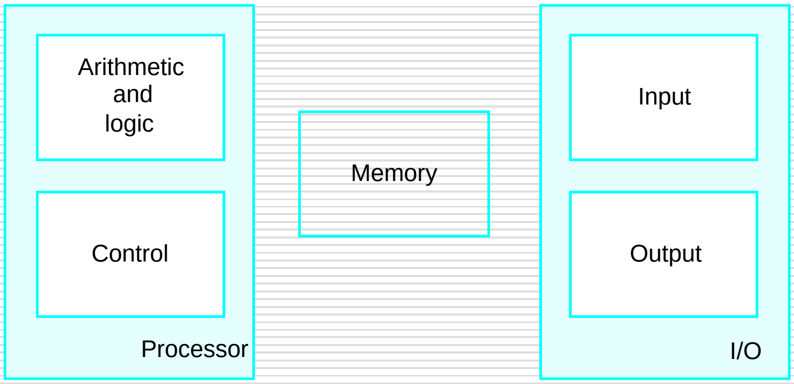
Example: PC Platform

- Driving modern technologies – economy of scale
- Hardware
    - Processors
    - Buses
    - Memory
    - Peripherals
- Software
    - OS's
    - Applications



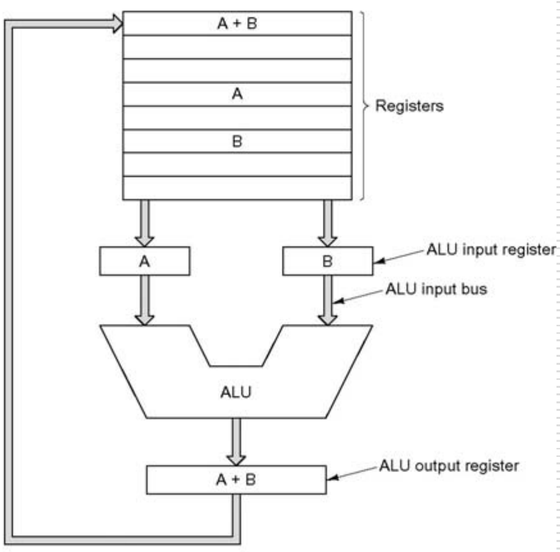
Basic Parts

- Processor, memory & IO



Common Processors

- Still Von Neumann architecture
  - Arithmetic-logic unit
  - Registers
  - Auxiliary registers

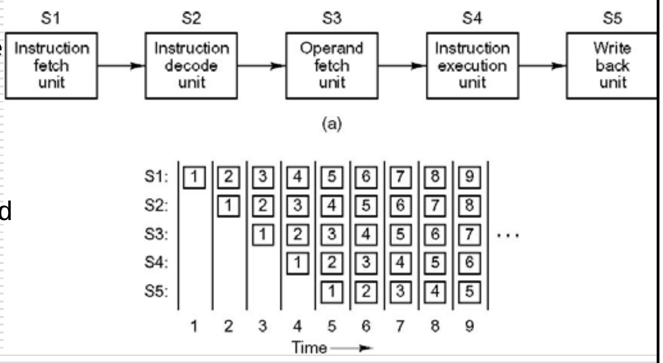


Processor Execution - Java code

```
public class Interp {
static int PC;                                    // program counter holds address of next instr
static int AC;                                    // the accumulator, a register for doing arithmetic
static int instr;                   // a holding register for the current instruction
static int instr_type;                            // the instruction type (opcode)
static int data_loc;                              // the address of the data, or −1 if none
static int data;                    // holds the current operand
static boolean run_bit = true;  // a bit that can be turned off to halt the ma
public static void interpret(int memory[ ], int starting_address) {
 PC = starting_address;
 while (runbit) {
    instr = memory[PC];                           // fetch next instruction into instr
    PC = PC + 1; /                   / increment program counter
    instr_type = get_instr_type(instr);           // determine instruction type
    data_loc = find_data(instr, instr_type);// locate data (−1 if none)
    if (data_loc >= 0)               // if data_loc is −1, there is no operand
          data = memory[data_loc];                // fetch the data
    execute(instr_type, data);       //execute instruction
    }
 }
private static int get_instr_type(int addr) { ... }
private static int find_data(int instr, int type) { ... }
private static void execute(int type, int data){ ... }
}
```

Basic Concepts - Pipelining

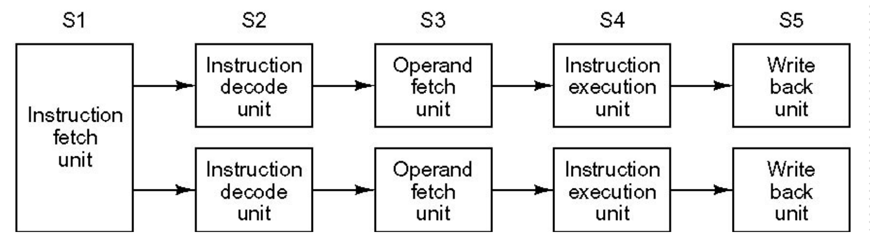
- Makes processor run at high clock rate
  - But might take more clock cycles
- Trick: overlap execution
  - Some overhead – first few instructions



S1:	1	2	3	4	5	6	7	8	9
S2:		1	2	3	4	5	6	7	8
S3:			1	2	3	4	5	6	7 …
S4:				1	2	3	4	5	6
S5:					1	2	3	4	5
	1	2	3	4	5	6	7	8	9

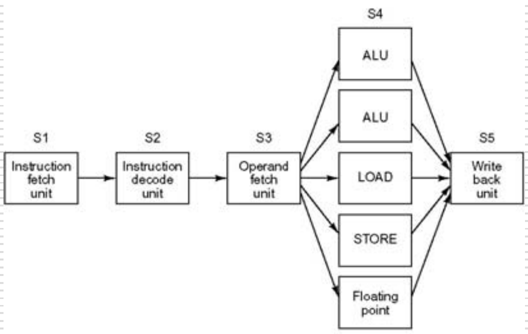
Other Speedups – Multiple Units

- Bottlenecks – execution in single pipeline units
  - ALU, especially floating point
- Resolution – provide multiple units



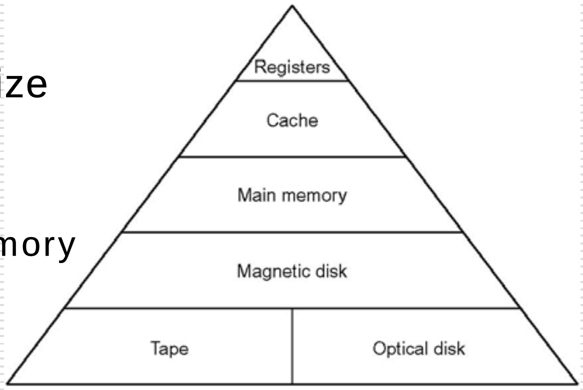
Superscalar Processors

- Common solution for modern processors
  - Multiple execution units

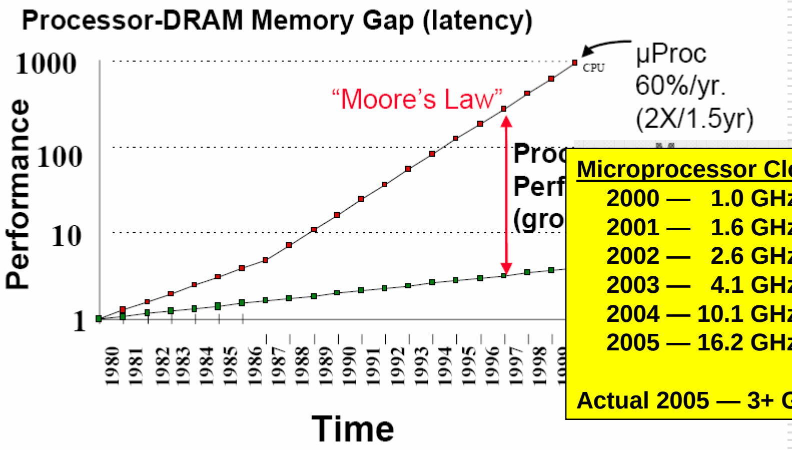


Memory

- Hierarchy of memory units
- Speed vs. size
- Solutions
  - Caching
  - Virtual memory



The Memory Latency Wall

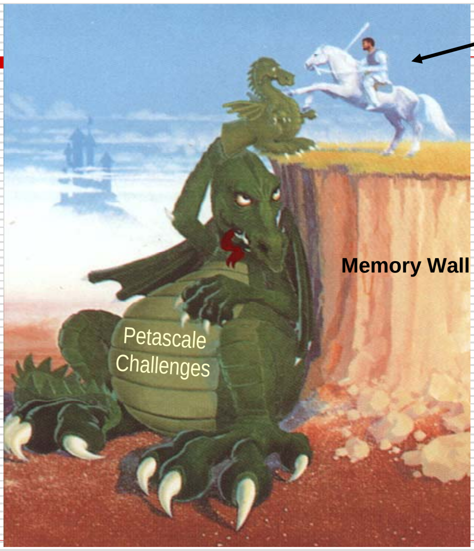


Microprocessor Clock

2000 —	1.0 GHz
2001 —	1.6 GHz
2002 —	2.6 GHz
2003 —	4.1 GHz
2004 —	10.1 GHz
2005 —	16.2 GHz

Actual 2005 — 3+ GHz

Challenges: Performance at Scale



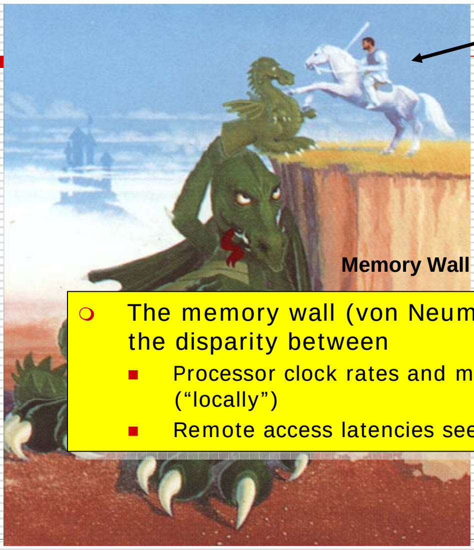
Advanced simulation and modeling apps

Conquering Terascale problems of today

Beware being eaten alive by the petascale problems of tomorrow.

Drawing by Thomas Zacharia (ORNL)

Performance at Scale



Advanced simulation and modeling apps

Conquering Terascale problems of today

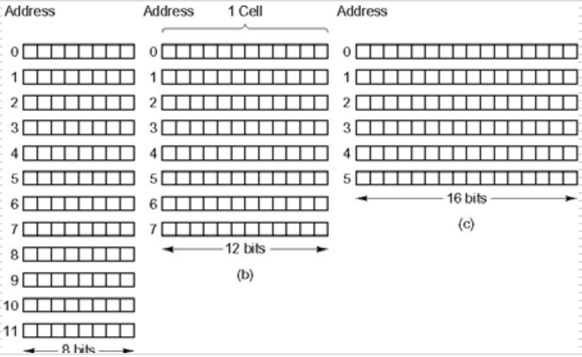
Memory Wall

- The memory wall (von Neumann bottleneck) — the disparity between
  - Processor clock rates and memory cycle times (“locally”)
  - Remote access latencies seen “system wide”

Drawing by Thomas Zacharia (ORNL)

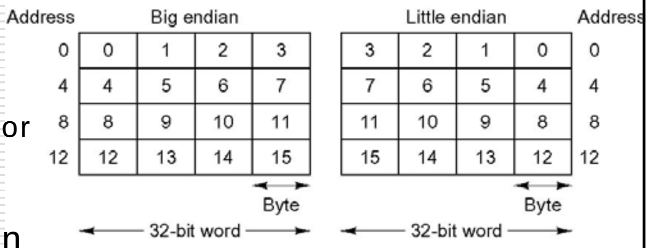
Memory Organization

- Computer **Word**
  - Basic unit of acces
- The same memory can be accessed in different ways



Little Endian vs. Big Endian

- Matter of preference
  - Significant implications for compatibility
- Some processors can have both

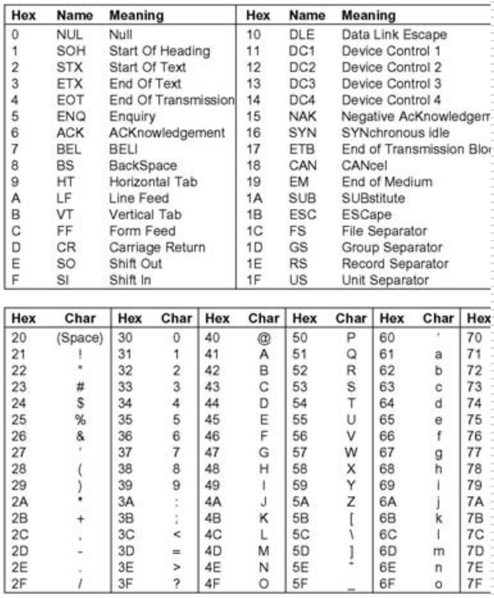


Address	Big endian			
0	0	1	2	3
4	4	5	6	7
8	8	9	10	11
12	12	13	14	15

Little endian				Address
3	2	1	0	0
7	6	5	4	4
11	10	9	8	8
15	14	13	12	12

Standardization –ASCII set

- Standardized way to use bits for encoding
  - Characters
  - Display
  - Communication
  - File

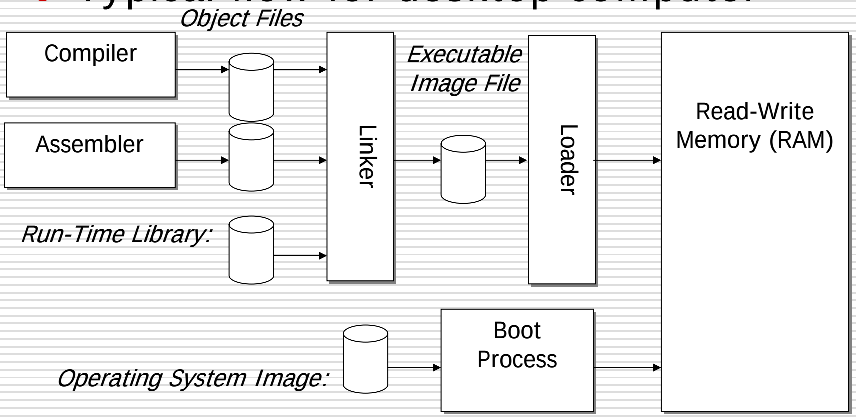


Hex	Name	Meaning	Hex	Name	Meaning
0	NUL	Null	10	DLE	Data Link Escape
1	SOH	Start Of Heading	11	DC1	Device Control 1
2	STX	Start Of Text	12	DC2	Device Control 2
3	ETX	End Of Text	13	DC3	Device Control 3
4	EOT	End Of Transmission	14	DC4	Device Control 4
5	ENQ	Enquiry	15	NAK	Negative AcKnowledgem
6	ACK	ACKnowledgement	16	SYN	SYNchronous idle
7	BEL	BELl	17	ETB	End of Transmission Blo
8	BS	BackSpace	18	CAN	CANcel
9	HT	Horizontal Tab	19	EM	End of Medium
A	LF	Line Feed	1A	SUB	SUBstitute
B	VT	Vertical Tab	1B	ESC	ESCape
C	FF	Form Feed	1C	FS	File Separator
D	CR	Carriage Return	1D	GS	Group Separator
E	SO	Shift Out	1E	RS	Record Separator
F	SI	Shift In	1F	US	Unit Separator

Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex
20	(Space)	30	0	40	@	50	P	60	`	70
21	!	31	1	41	A	51	Q	61	a	71
22	"	32	2	42	B	52	R	62	b	72
23	#	33	3	43	C	53	S	63	c	73
24	$	34	4	44	D	54	T	64	d	74
25	%	35	5	45	E	55	U	65	e	75
26	&	36	6	46	F	56	V	66	f	76
27	'	37	7	47	G	57	W	67	g	77
28	(	38	8	48	H	58	X	68	h	78
29	)	39	9	49	I	59	Y	69	i	79
2A	*	3A	:	4A	J	5A	Z	6A	j	7A
2B	+	3B	;	4B	K	5B	[	6B	k	7B
2C	,	3C	<	4C	L	5C	\	6C	l	7C
2D	-	3D	=	4D	M	5D	]	6D	m	7D
2E	.	3E	>	4E	N	5E	^	6E	n	7E
2F	/	3F	?	4F	O	5F	_	6F	o	7F

Software Build and Load

- Typical flow for desktop computer



Object Files

Compiler

Assembler

Run-Time Library:

Linker

Executable Image File

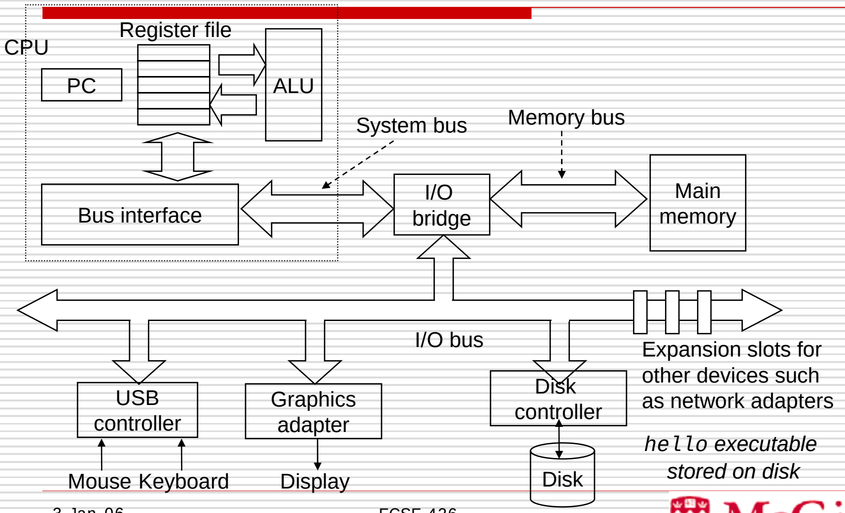
Loader

Read-Write Memory (RAM)

Operating System Image:

Boot Process

Example Program Creation & Run



CPU

Register file

PC

ALU

Bus interface

System bus

Memory bus

I/O bridge

Main memory

I/O bus

Expansion slots for other devices such as network adapters

USB controller

Graphics adapter

Disk controller

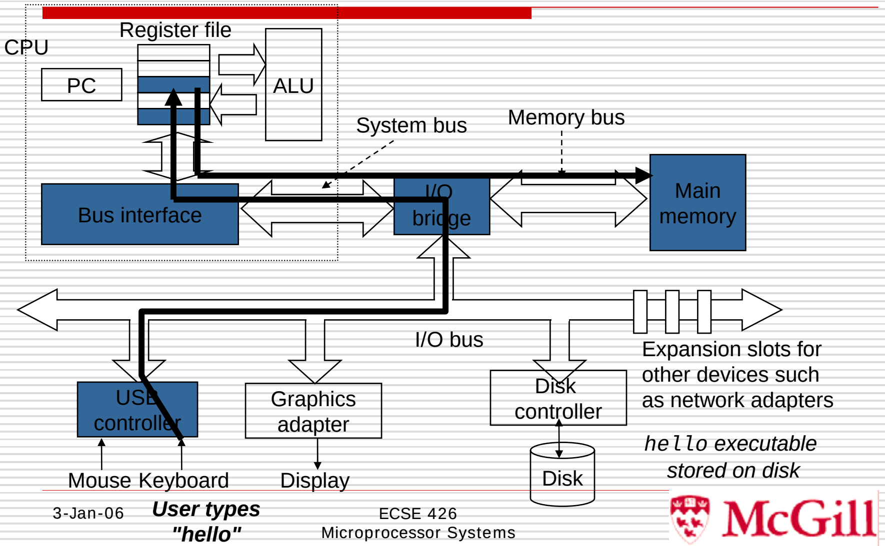
Mouse Keyboard

Display

Disk

`hello` executable stored on disk

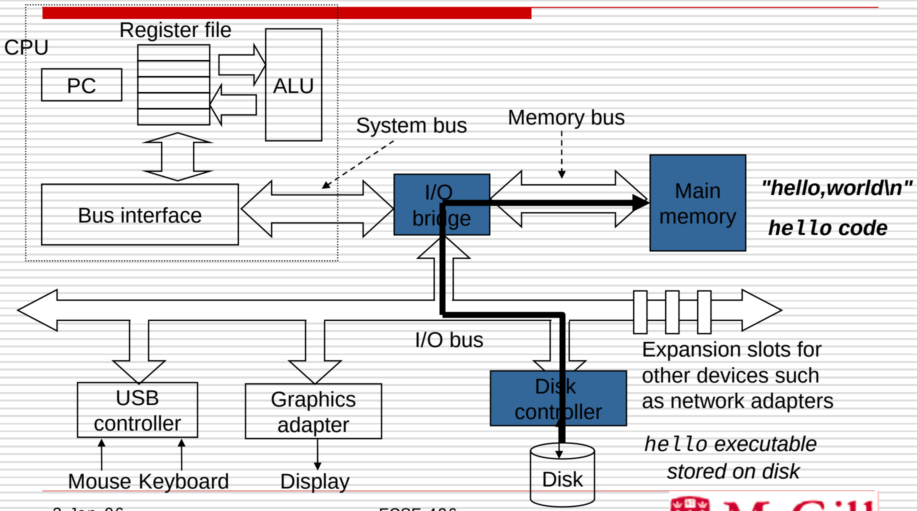
Reading “Hello” Command



hello executable stored on disk

User types "hello"

Loading “Hello” Program

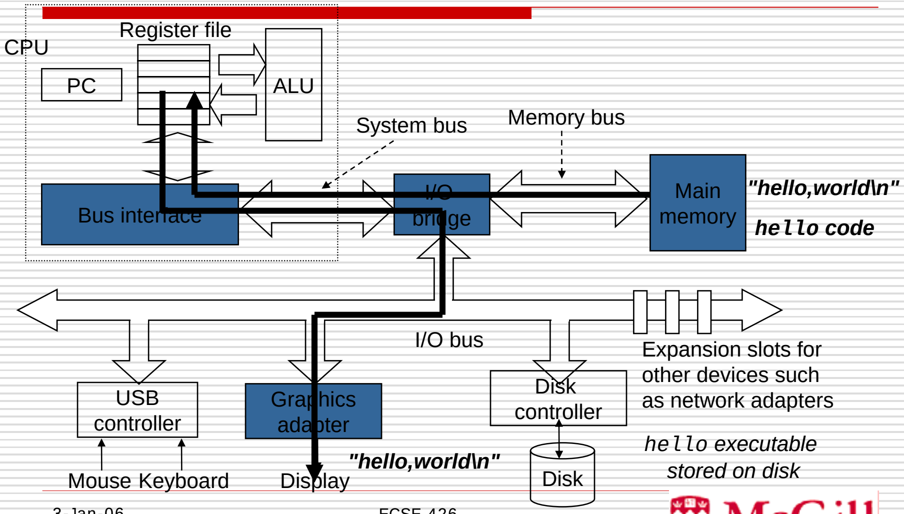


"hello,world\n"

hello code

hello executable stored on disk

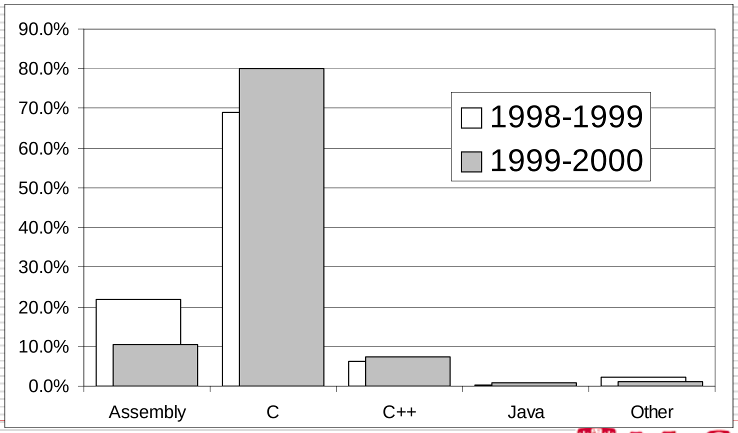
Finally -Program Running



CPU
Register file
PC
ALU
System bus
Memory bus
Bus interface
I/O bridge
Main memory
"hello,world\n"
`hello` code
I/O bus
Expansion slots for other devices such as network adapters
USB controller
Graphics adapter
Disk controller
`hello` executable stored on disk
Mouse Keyboard
Display
"hello,world\n"
Disk

Embedded Systems - Languages

- Recent Statistics



Course Overview

- Background
  - Computer Arch. Basics
- Microprocessors
  - Commercial: Pentium, Sparc; Potential: Java
- Embedded Processors
  - TI MSP430, ARM, PowerPC ▪ ▪
- Embedded System Design
  - HW and SW techniques
- Real Time Systems
  - Techniques and Tools

Course Objectives

- Understand microprocessor-based systems
- Get familiar with basic tools
- Skills in machine interfacing, assembler and embedded C programming
- Design a sizeable embedded system
  - Previous projects: Music player, file swapping system, PDAs (with handwriting recognition), wireless data collection systems
- Build teamwork skills

Reference Books

- Reference
  - A. Tanenbaum, *Structured Computer Organization*, 4th edition, Prentice-Hall, 1999.
  - C. Nagy, *Embedded Systems Design Using the TI MSP430 Series,* Elsevier Science, 2003
  - B. Shriver and B. Smith, *The Anatomy of a High-Performance Microprocessor - A Systems Perspective,* IEEE Computer Society Press, 1998.
  - C. Hamacher, Z. Vranesic and S. Zaky, *Computer Organization*, 5th edition, McGraw-Hill, 2002.

Academic Integrity

McGill University values academic integrity.

Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see http://www.mcgill.ca/integrity for more information).

Finally: Grading Scheme

- 12% participation - 4 quizzes
  - Schedule will be announced
- 38% laboratories
- 40% project

Assessment

Evaluations	Contribution to Final Grade		
Experiment 1	18	Demo	10
		Report	8
Experiment 2	15	Demo	10
		Lab Notes	5
Experiment 3	15	Demo	10
		Lab Notes	5
Project	40	Demo 1	13
		Final Demo	15
		Report	12
Quizzes	12		12