

# Digital Systems L2 – Computer Systems

Fang Yuan

School of Electronics Science and Engineering  
Nanjing University  
Nanjing 210046

2022



# Agenda

- Computer basis
  - Memory and filesystem
  - Linux
  - Device drivers
  - Computer interfaces
  - Embedded systems



# Outline

1

## Computer Basis

- History of Computer
- Digital Representations
- Characters and Their Representations
- Computer Systems
- Instruction Set
- Bus

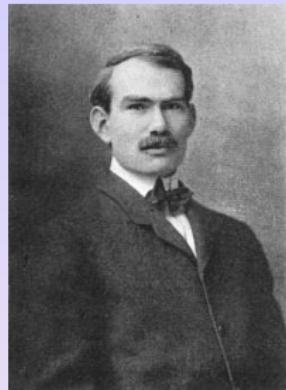


# Pre-historic Time – Vacuum Tube

John  
Fleming,  
diode  
inventor  
(1904)

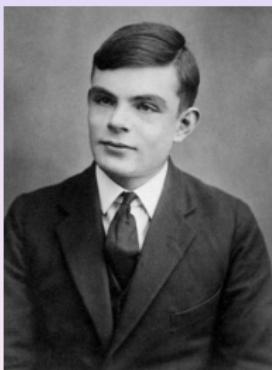


Lee  
DeForest,  
triode  
inventor  
(1906)

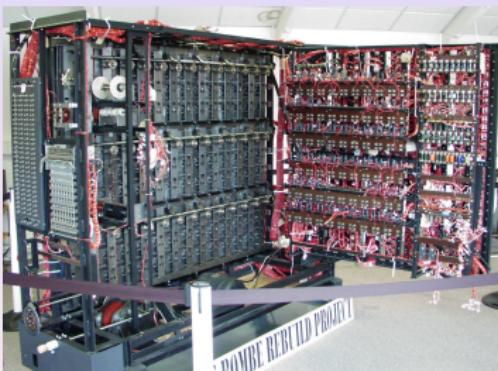


# Bombe

The bombe is an electro-mechanical device used by the British cryptologists to help decipher German Enigma-machine-encrypted secret messages during World War II.



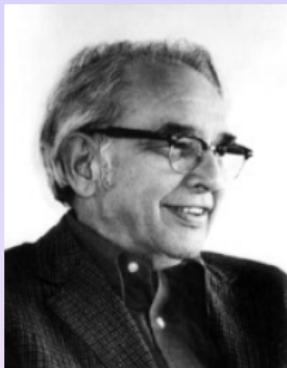
Alan Turing



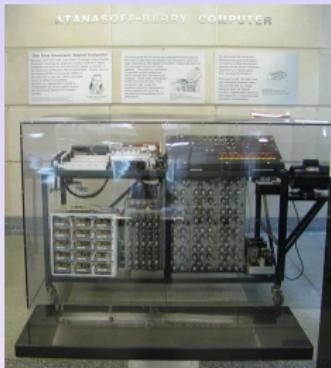
Bombe-rebuild(Bletchley Park)



# First Generation of Computers



John Atanasoff



ABC replica(Iowa  
State Uni.)

ABC (after the name of its creators, Atanasoff and Berry) was designed to solve systems of linear equations and was successfully tested in 1942. However...

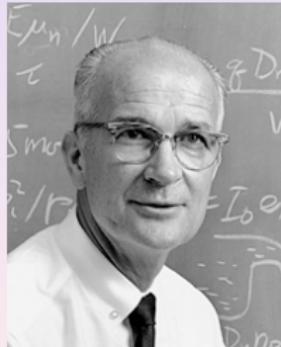


# Second Generation of Computers (Transistors)

- J.E. Lilienfeld patented a **field-effect transistor** in 1926.
- Point-Contact transistor**, invented by Bardeen, Shockley and Brattain in 1947 (Nobel Prize in physics, 1956).



John Bardeen



William Shockley

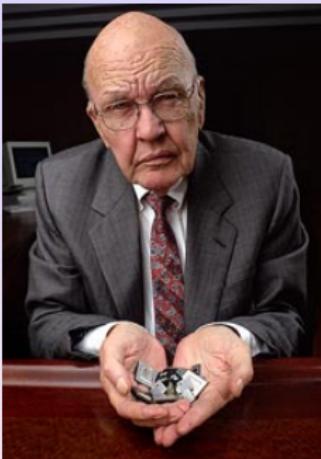


Walter Brattain



# Third Generation of Computers (IC)

1957, Kilby of TI, invented integrated circuits.



Jack Kilby (1923–2005)

2000, Jack Kilby won the Nobel Prize in physics for his invention.

# Third Generation of Computers (IC)

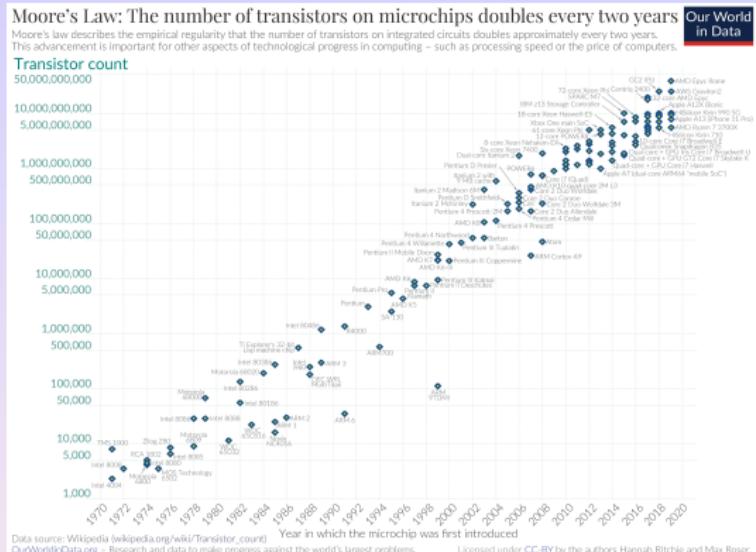
1958, Robert Noyce of Fairchild Semiconductor Inc. (he was also Intel Inc. co-founder) invented IC independently.



Robert Noyce (1927–1990) and Gordon Moore (1929–)



# Moore's Law



The number of Transistors in a dense IC doubles every year(Gorden Moore, 1965). This forecast was revised to doubling every two years in 1975.

# High Performance Computers



ORNL's Frontier, the first to break the Exaflop ceiling super computer (June, 2022). 8,730,112 cores, 21,100.00 kW. power efficiency rating of 52.23 Gigaflops/watt. <https://www.top500.org>



# Radix

A **true value** of radix  $r$  represented by:

$$\begin{aligned}
 v &= A_{n-1}r^{n-1} + A_{n-2}r^{n-2} + \dots + A_0r^0 + A_{-1}r^{-1} + \dots \\
 &= \sum_{i=-m}^{n-1} A_i r^i
 \end{aligned}$$

Table: 248.625 (Decimal) in different radices

Radix		Value
2	Binary	11 111 000.101B
8	Octet	370.50 (370.5Q)
10	Decimal	248.625D
16	Hexadecimal	F8.AH



# Binary Coding (Signed Int)

digits	original	1's complement	2's complement
-8	-	-	1000
-7	1111	1000	1001
:			
-1	1001	1110	1111
-0	1000	1111	0000
+0	0000	0000	0000
:			
+7	0111	0111	0111



# Binary Computations

## Conversion between radices

- Binary  $\Leftrightarrow$  Hex: a group of four bits (a nibble) can represent a value from 0 to 15, and vice versa.
- Binary  $\Leftrightarrow$  Decimal: division and modulus

## Binary Computations

- Binary arithmetic: addition/subtraction, multiplication, ...
- Boolean logic: AND, OR, Exclusive-OR, NOT, ...
- Shift and rotation



# Carry and Overflow

Signed int → 2's complement

Overflow, but no carry

$$\begin{array}{r} 99 \\ + 100 \\ \hline 199 \end{array} \rightarrow \begin{array}{r} 0110\ 0011 \\ + 0110\ 0100 \\ \hline 1100\ 0111 \end{array}$$

Carry, but no overflow

$$\begin{array}{r} -33 \\ + 100 \\ \hline 67 \end{array} \rightarrow \begin{array}{r} 1101\ 1111 \\ + 0110\ 0100 \\ \hline 1\ 0100\ 0011 \end{array}$$



# Floating Point

## User Defined Floating Point

exponential	mantissa
-------------	----------

4 bits (E, sign-mag.) 8 bits (M, 2's complement):

$$1011\ 01100000 = 2^{-3} \times 0.11 \quad (\text{normalized})$$

$$1010\ 00110000 = 2^{-2} \times 0.011$$

5 bits (E, 2's complement), 7 bits (1's complement):

$$11101\ 1001111 = -2^{-3} \times 0.11$$



# Floating Point

## IEEE-754 Single Precision Floating Point

s	e(8bit)	m(23bit)	Number
0	0	0	0
1	0	0	-0
s	0	$\neq 0$	$(-1)^s \times 0.m \times 2^{-126}$
s	$0 < e < 255$	m	$(-1)^s \times 1.m \times 2^{e-127}$
0	255	0	$+\infty$
1	255	0	$-\infty$
s	255	$\neq 0$	NaN(not a number)

41C40000

$$s = 0$$

$$e = 1000011b = 128 + 2 + 1 = 131$$

$$m = 1.10001...00b = 1.53125$$

$$\begin{aligned} x &= (-1)^0 \times 1.53125 \times 2^{(131-127)} = 1.53125 \times 2^4 \\ &= 24.5 \text{ (Decimal)} \end{aligned}$$



# ASCII

- ASCII encodes 128 specified characters into seven-bit integers based on the English alphabet.
- The original ASCII specification included 33 non-printing control codes which originated with Teletype machines, only a few of these are still in use, such as the carriage return, line feed and tab codes.
- DO NOT confuse digits and numbers when using ASCII.



# Chinese Characters

- National Standard (GB18030)
- Unicode

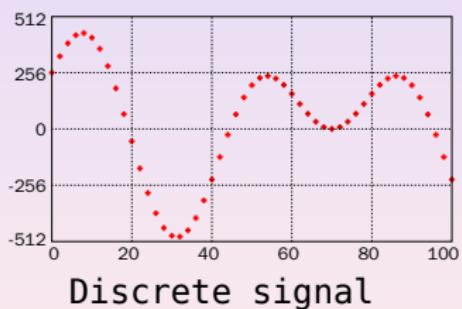
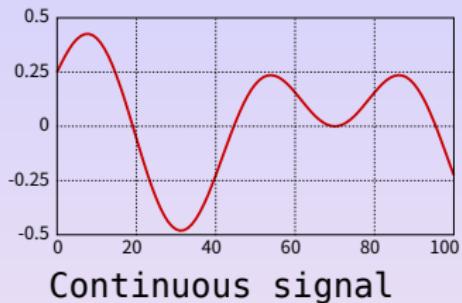
漢(Unicode: 6F22)    0110 1111 0010 0010  
(UTF-8: E6 BC A2)    11100110 10111100 10100010

↓                      ↓                      ↘

UTF-8 Chinese Character



# Digital and Analogue



n	x(n)
0	256
1	331
2	390
3	423
4	436
.	.
.	.
.	.
.	.



# Media Information

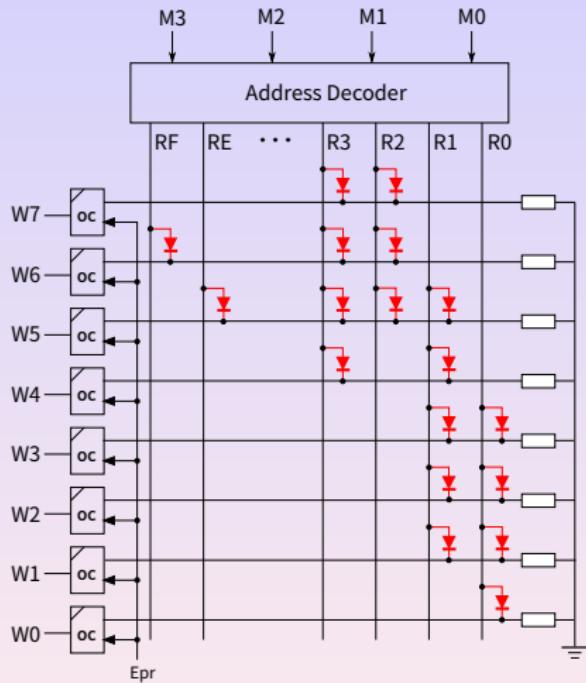
In computer, media information are often stored in files.

- Audios: wav, mp3, ogg, ...
- Images/Pictures: bmp, png, jpg, pdf, ...
- Videos: avi, mv, mpeg, ...

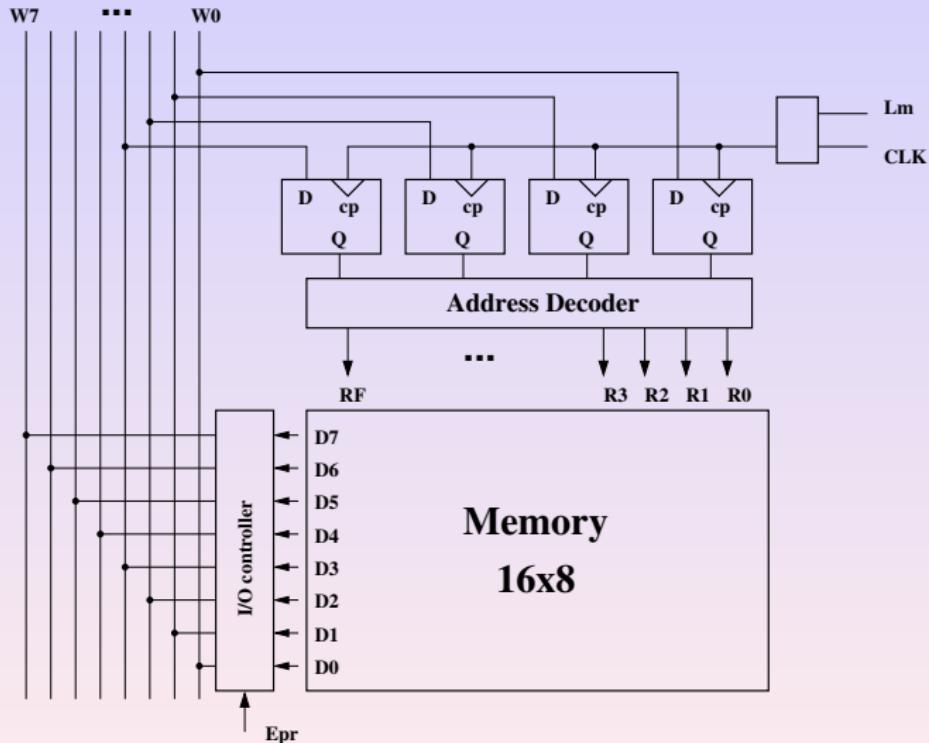
Media files are often compressed (*lossy* compression).



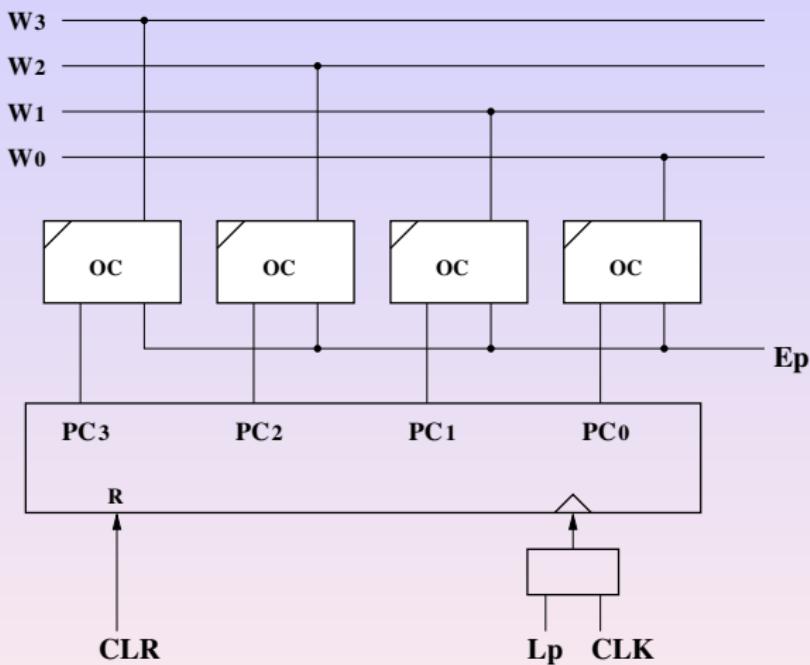
# How Program is Stored



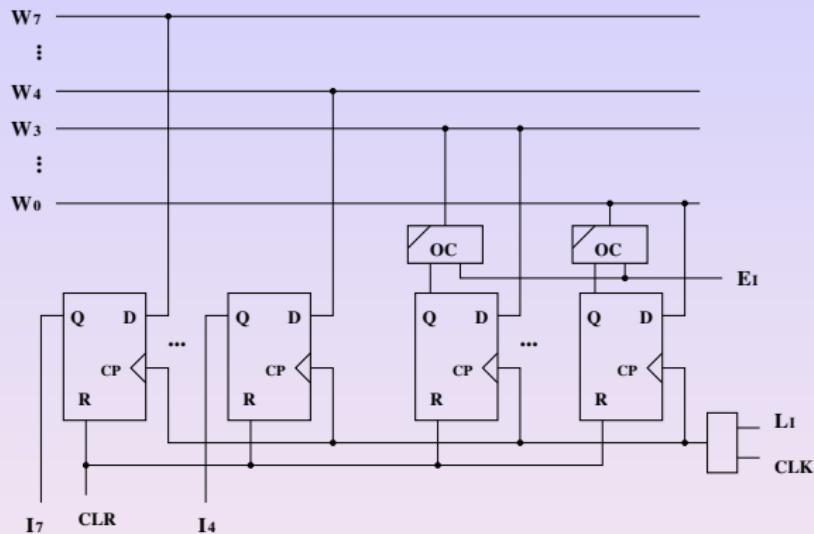
# Memory Array



# Program Counter(PC)



# Instruction Register



# Instruction set

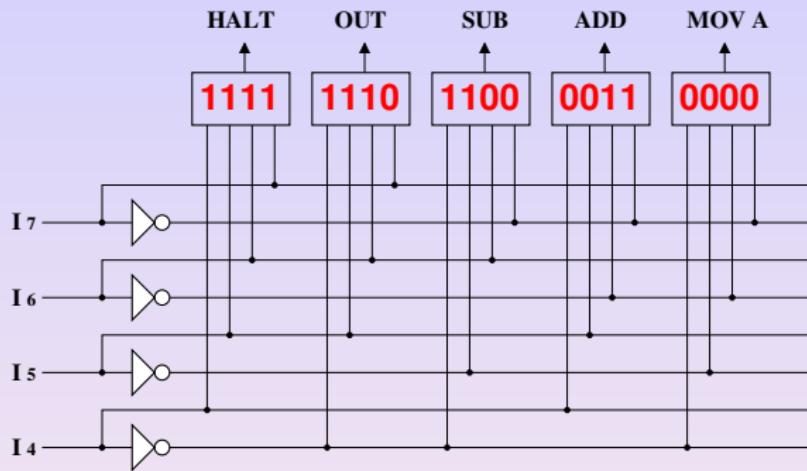
Instruction	Mnemonic	Machine code	Operation
Fetch	MOV A, (Rx)	0000	$A \leftarrow (Rx)$
Add	ADD A, (Rx)	0011	$A \leftarrow A + (Rx)$
Subtract	SUB A, (Rx)	0100	$A \leftarrow A - (Rx)$
Output	OUT (n), A	1110	$(n) \leftarrow A$
Halt	HALT	1111	CPU halt

Calculate  $32+64=?$

addr.	code	mnemonic	; comments
0:	00001111	MOV A, (RF)	; Load Rf to A
1:	00111110	ADD A, (RE)	; Load Re to B ; Add A and B
2:	11100000	OUT (0), A	; Output



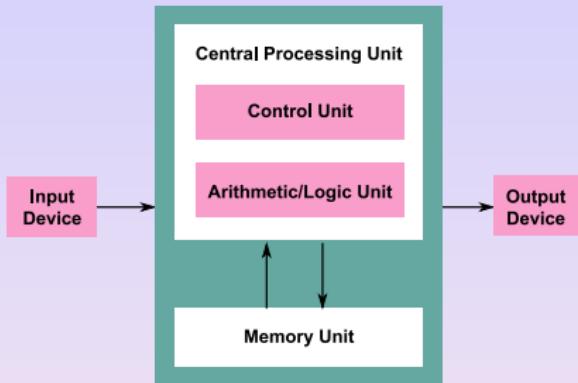
# Instruction Decoder



# Stored-Program Computer



John von  
Neumann



von Neumann Architecture

1945, John von Neumann described a design architecture for an electronic digital computer in his 101-page document. The architecture then is called **von Neumann architecture**, or **Princeton architecture**.



# Stored-Program Computer

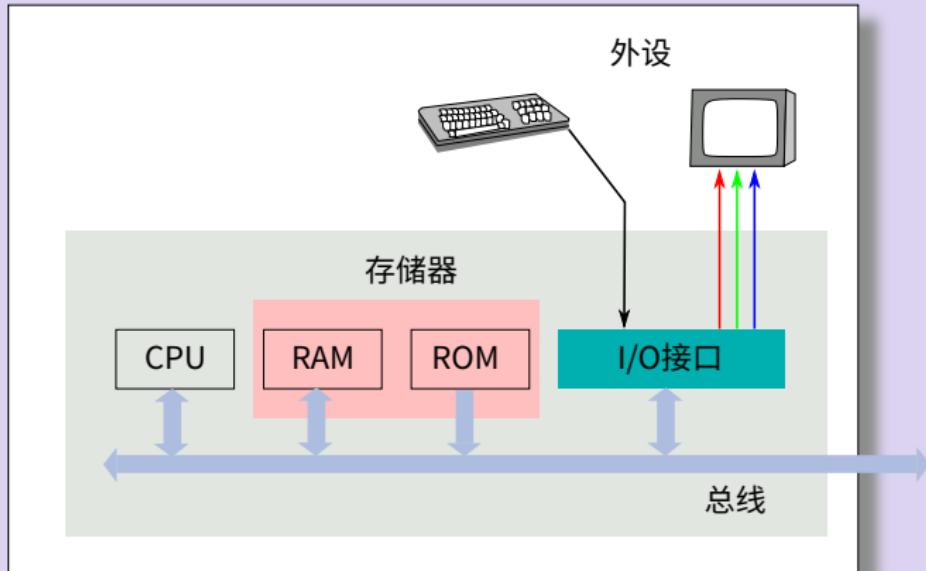
An electronic digital computer has following components:

- A processing unit that contains an arithmetic logical unit and processor registers
- A control unit that contains an Instruction register and program counter
- Memory that stores data and instructions
- External mass storage
- Input and output mechanisms



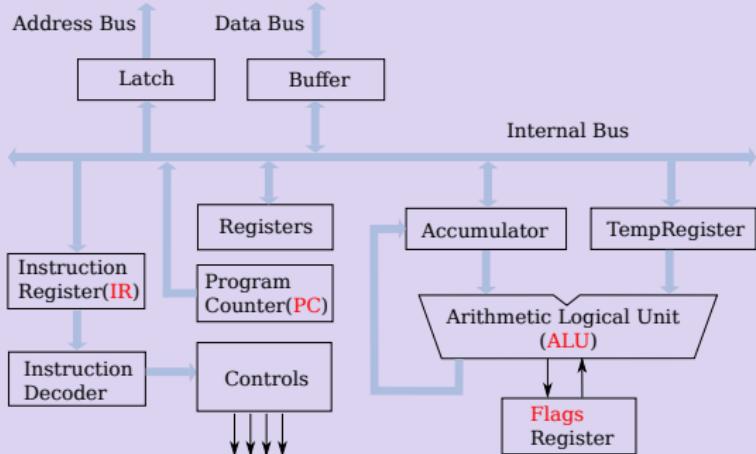
# Hardware

## Computer System Hardware



# CPU

## Simplified CPU Diagram



Traditionally, a CPU refers to its processing unit and control unit. Modern CPUs are microprocessors, including memory, peripheral interfaces and other components.



# CPU Operations

## Instruction Fetch

Retrieving an instruction from program memory. The instruction's location (address) in program memory is determined by a program counter (PC). The PC is then modified to the next instruction location.



# CPU Operations

## Instruction Decode

Instruction decoder converts the instruction into electrical signals that control other parts of the CPU.

The way in which the instruction is interpreted is defined by **Instruction Set Architecture (ISA)**.



# CPU Operations

## Execute

During each action in this step, various parts of the CPU are electrically connected so they can perform all or part of the desired operations. Normally the results are written to internal registers, including flags register.



# CPU Operations

## Write-back

After execute step, the result may be written to memory for later use.



# Instruction Set Architecture

- CPUs from different vendor have different ISAs.  
X86 (Intel and AMD), PowerPC (IBM alliance), Arm, MIPS,...
- CISC (Complex Instruction Set Computer) vs. RISC (Reduced Instruction Set Computer)

Intel Core i7		EM64T
AMD Ryzen 3000		AMD64
Rockchip RK3399	Cortex-A72/A53	ARMv8
TI AM335x	Cortex-A8	ARMv7
Apple A12	n.a	(ARMv8?)
TI C6000	C6X	C6X



# RISC vs. CISC

## Euclid's Algorithm of GCD

```
int gcd (int i, int j)
{
    while (i != j) {
        if (i > j) i -= j;
        else         j -= i;
    }
    return i;
}
```



## X86 Instructions of GCD

addr.	code	mnemonic	
100:	66 89 f8	MOV	EAX, EDI
103:	66 39 f0	CMP	EAX, ESI
106:	74 0a	JE	114
108:	7e 04	JLE	10f
10a:	66 29 f0	SUB	EAX, ESI
10d:	eb f6	JMP	103
10f:	66 29 c6	SUB	ESI, EAX
112:	eb f2	JMP	103
114:			



# RISC

## Arm Instructions of GCD

addr.	code	mnemonic	
100:	e1500001	CMP	R0 , R1
104:	c0400001	SUBGT	R0 , R0 , R1
108:	d0411000	SUBLE	R1 , R1 , R0
10c:	1affffffb	BNE	100

## RISC features:

- fixed-length instructions (32bit/64bit)
- fewer cycles per instruction(nearly 1)
- more registers (32 or more)
- simple addressing modes (orthogonality)



# RISC

## Arm Instructions of GCD

addr.	code	mnemonic	
100:	e1500001	CMP	R0 , R1
104:	c0400001	SUBGT	R0 , R0 , R1
108:	d0411000	SUBLE	R1 , R1 , R0
10c:	1affffffb	BNE	100

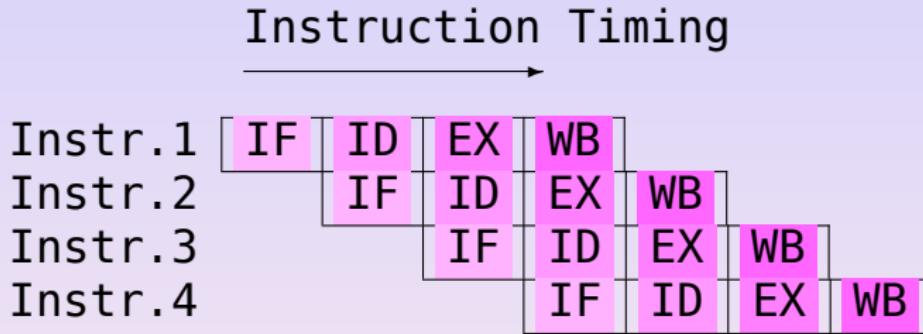
## RISC features:

- fixed-length instructions (32bit/64bit)
- fewer cycles per instruction(nearly 1)
- more registers (32 or more)
- simple addressing modes (orthogonality)



# Instruction Pipelining

Each action in the CPU is splitted by clock cycle.



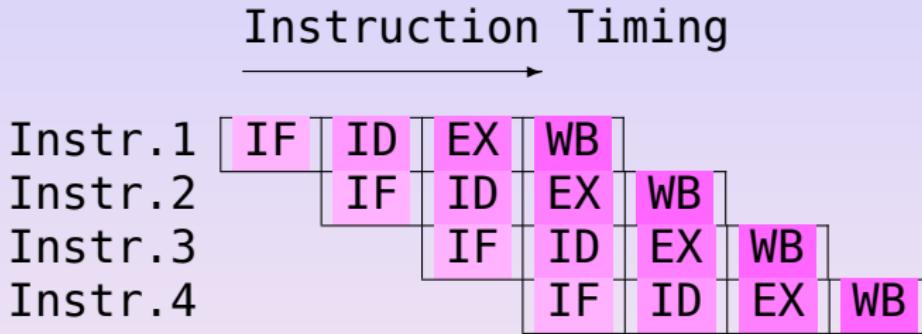
Conflicts and errors in pipelines:

- conflicts on resource: stalling – multiple pipelines
- dependencies on results: reordering
- branch instructions: guess and backtrack



# Instruction Pipelining

Each action in the CPU is splitted by clock cycle.



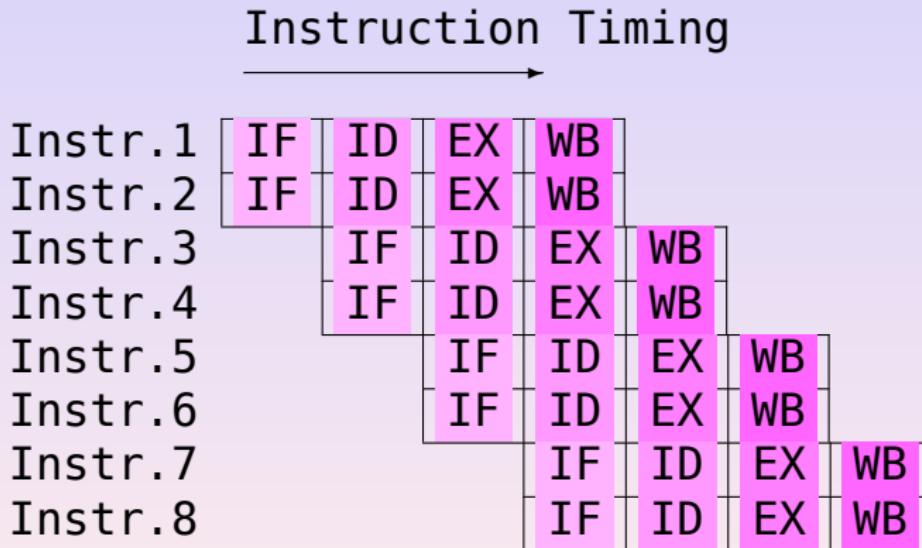
Conflicts and errors in pipelines:

- conflicts on resource: stalling – multiple pipelines
- dependencies on results: reordering
- branch instructions: guess and backtrack



# Superscalar Architecture

Many modern CPUs have more than one instruction pipelines.



# Memory

- Memory is used to store data or programs for temporary or permanent use.
- RAM – Random Access Memory
- ROM – Read-Only Memory



# Bus

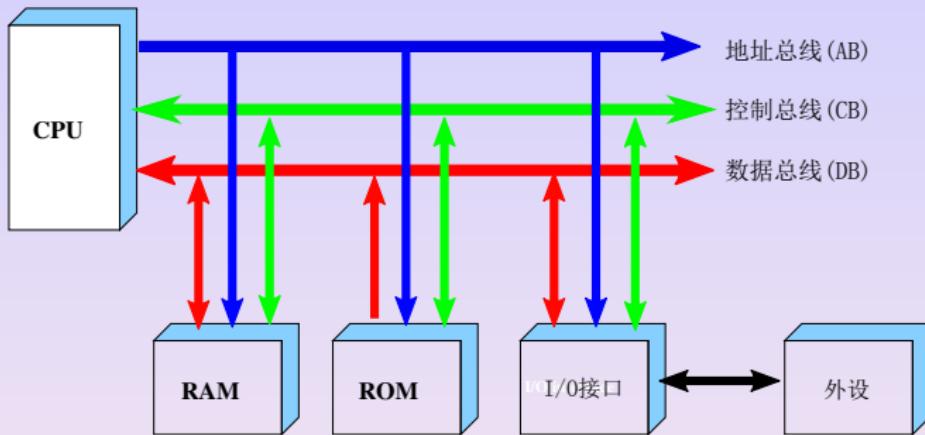
Bus is a communication system that transfers data between components inside (or eventually outside) a computer.

## Bus

- Data Bus: transferring data.
- Address Bus: used to specify a physical address of a memory.
- Control Bus: control the way how data are transferred.



# Typical Bus Structure

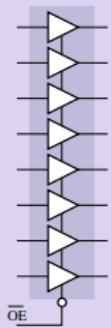
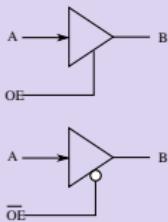


- parallel: ISA, PCI, IDE, ...
- serial: UART, I2C, USB, PCIe, SATA, ...

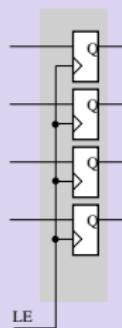
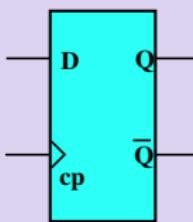


# Buffers and latches

## Tri-state and buffer



## Flip-flop and latch

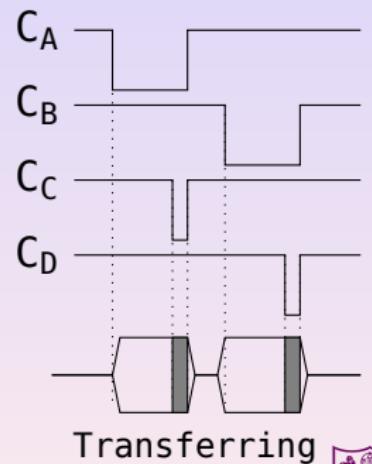
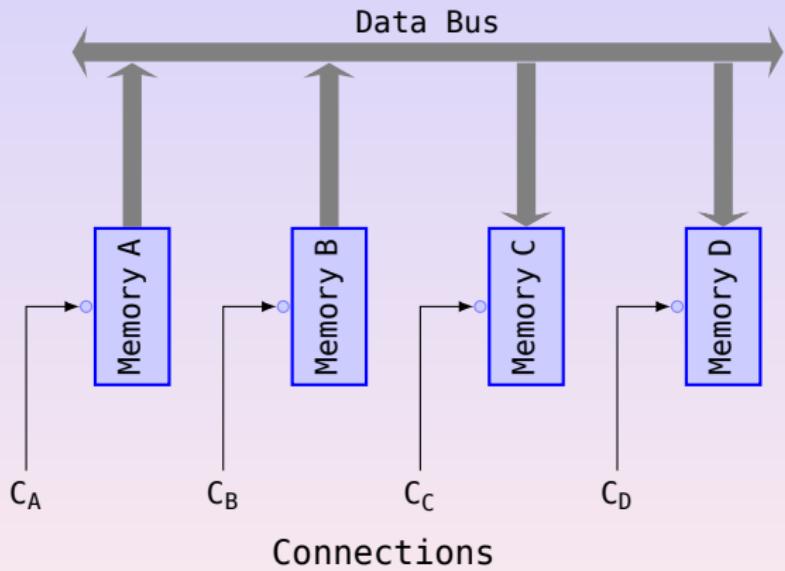


**Buffer** Either input and output have the same level, or output is in high impedance state. (**level controlled**)

**Latch** The output state (high level or low level) keeps unchanged until next clock pulse. (**edge triggered**)



# Data Transferring



# Specifications

## Processors

- Clockrate: MHz, GHz
- Performance: **MIPS** (Millions of Instructions Per Second), **MFLOPS** (Millions of Floating-point Operations Per Second)
- Word-length: 32bit, 64bit
- Multi-core
- On-chip memory (Cache)

## Systems

- Memory (MB, GB, TB)
- Devices
- Operating System/Software



# Specifications

## Processors

- Clockrate: MHz, GHz
- Performance: **MIPS** (Millions of Instructions Per Second), **MFLOPS** (Millions of Floating-point Operations Per Second)
- Word-length: 32bit, 64bit
- Multi-core
- On-chip memory (Cache)

## Systems

- Memory (MB, GB, TB)
- Devices
- Operating System/Software



# Summary

- Binary information representations
- Microcomputer systems
  - Architecture
  - Instruction Set
  - Memory
  - Bus
  - Performance

