Introduction to Microcontrollers

Boolean Algebra

- Boolean algebra is a mathematical system for the manipulation of variables that can have one of two values.
  - In formal logic, these values are “true” and “false.”
  - In digital systems, these values are “on” and “off,” 1 and 0, or “high” and “low.”
- Boolean expressions are created by performing operations on Boolean variables.
  - Common Boolean operators include AND, OR, and NOT.

Boolean Algebra

- A Boolean operator can be completely described using a truth table.
- The truth table for the Boolean operators AND and OR are shown at the right.
- The AND operator is also known as a Boolean product. The OR operator is the Boolean sum.

X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

Boolean Algebra

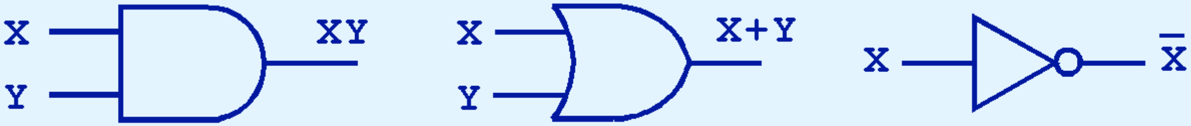
- The truth table for the Boolean NOT operator is shown at the right.
- The NOT operation is most often designated by an overbar. It is sometimes indicated by a prime mark (‘)

NOT x

x	$\bar{x}$
0	1
1	0

Logic Gates

- The three simplest gates are the AND, OR, and NOT gates.



X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT X

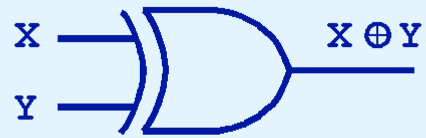
X	$\overline{X}$
0	1
1	0

Logic Gates

- Another very useful gate is the exclusive OR (XOR) gate.
- The output of the XOR operation is true only when the values of the inputs differ.

X XOR **Y**

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

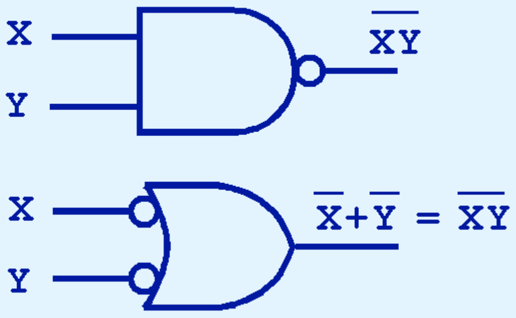


Logic Gates

- NAND and NOR are two very important gates. Their symbols and truth tables are shown at the right.

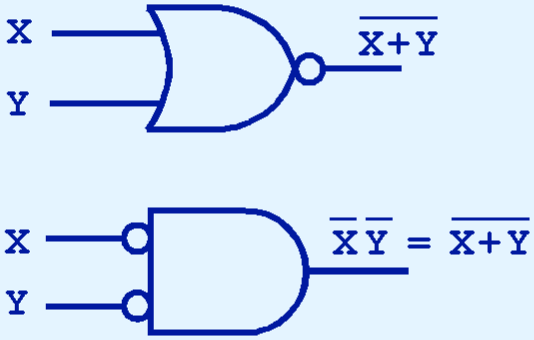
X NAND Y

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0



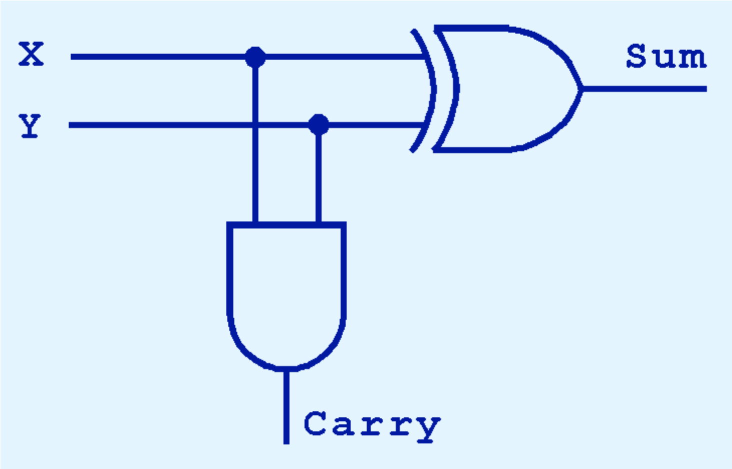
X NOR Y

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0



Combinational Circuits

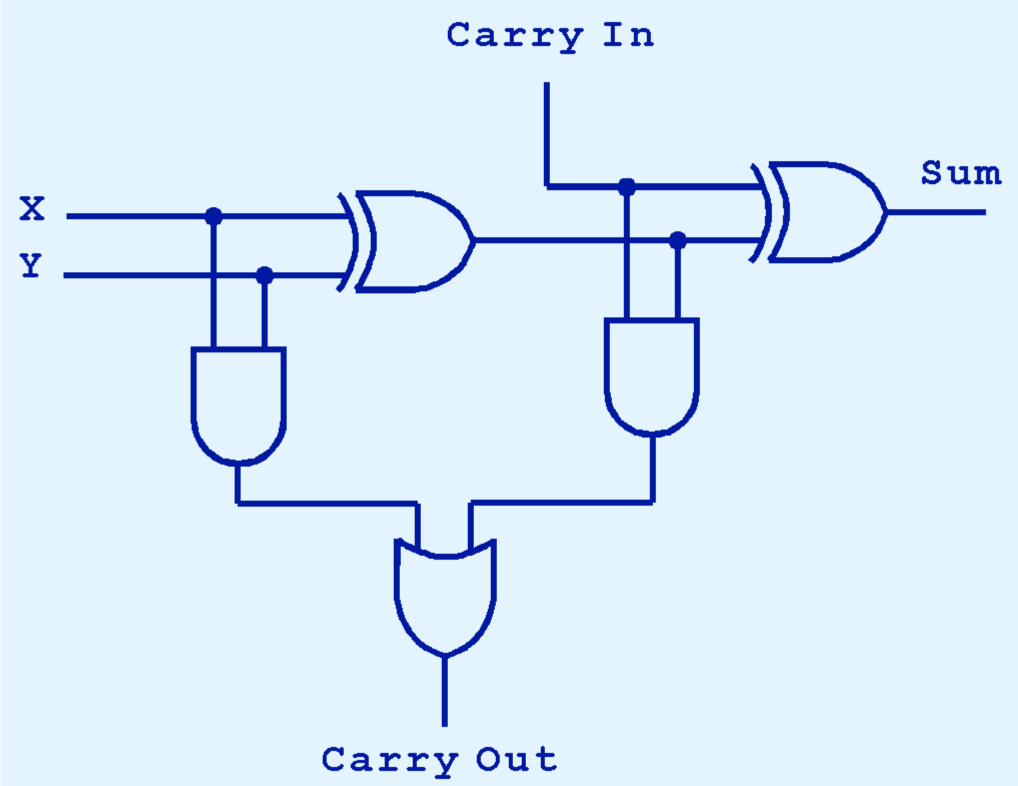
- Half adder



Inputs		Outputs	
X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Combinational Circuits

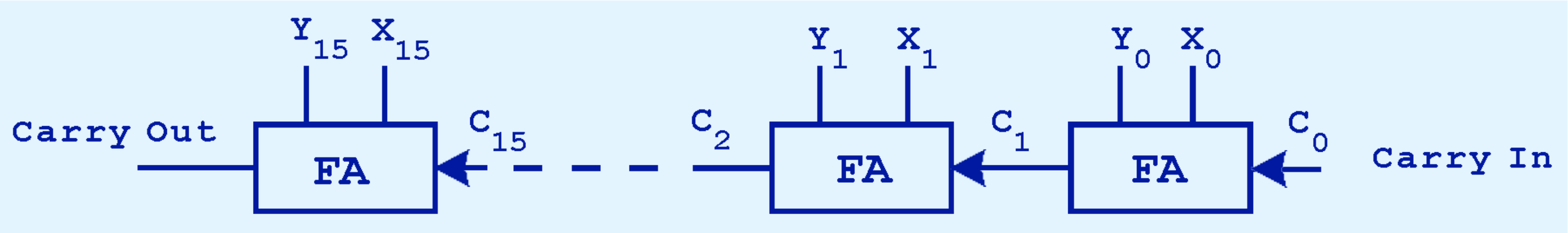
- Full adder



Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

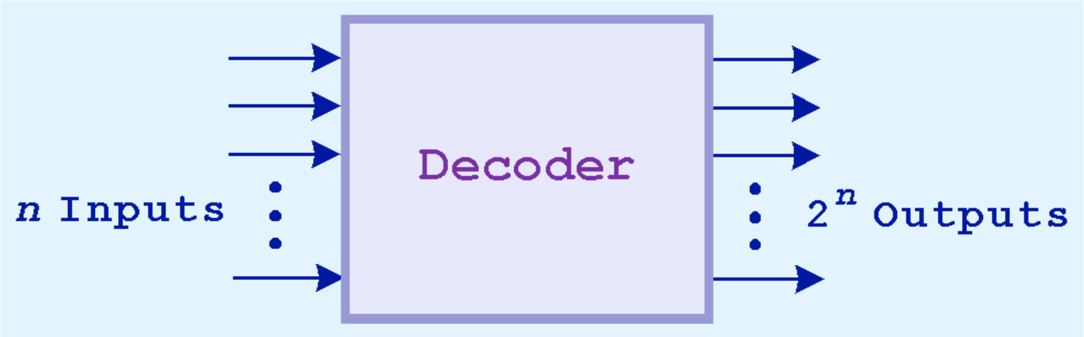
Combinational Circuits

- 16bits adder



Combinational Circuits

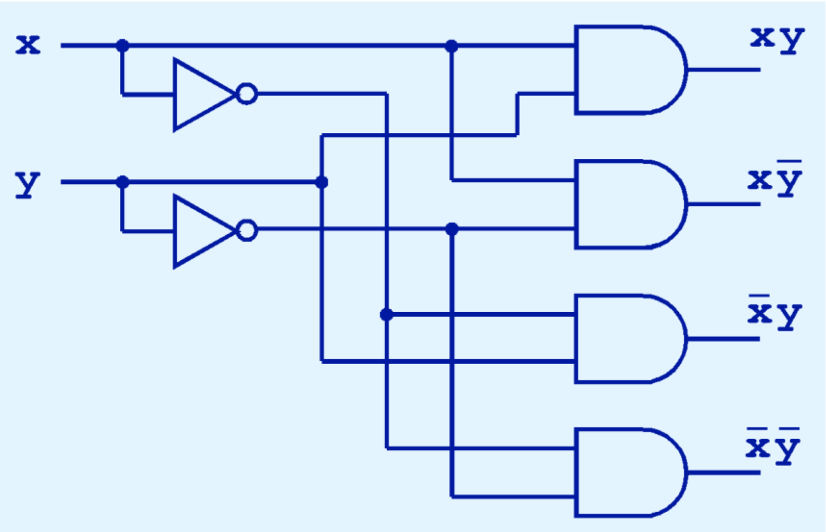
- Decoder
- Among other things, they are useful in selecting a memory location according a binary value placed on the address lines of a memory bus.
- Address decoders with *n* inputs can select any of 2^n locations.



Combinational Circuits

- 2-to-4 decoder

in1	in2	out1	out2	out3	out4
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

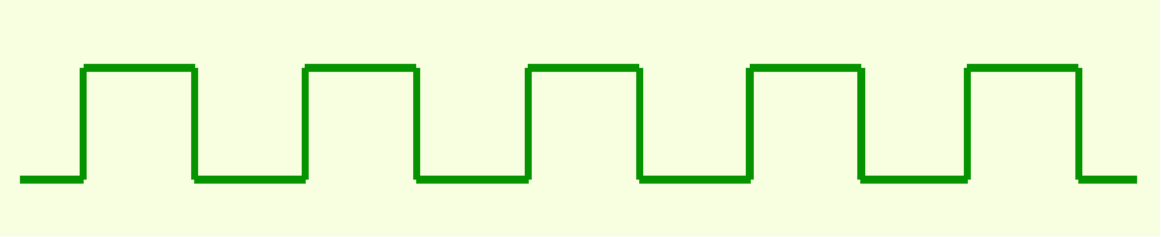


Sequential Circuits

- Combinational logic circuits are perfect for situations when we require the immediate application of a Boolean function to a set of inputs.
- There are other times, however, when we need a circuit to change its value with consideration to its current state as well as its inputs.
  - These circuits have to “remember” their current state.
- Sequential logic circuits provide this functionality for us.

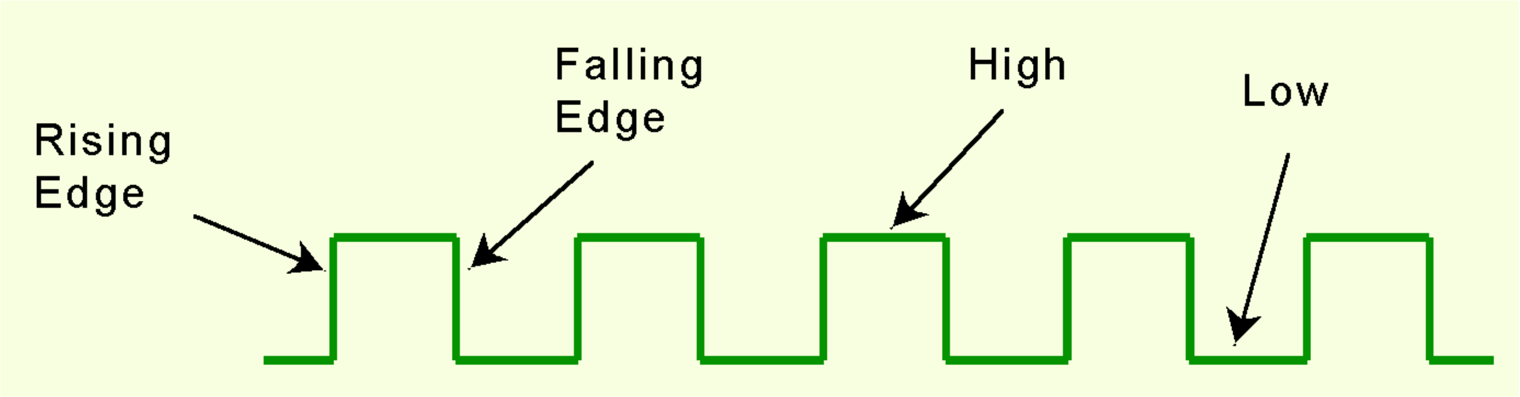
Sequential Circuits

- As the name implies, sequential logic circuits require a means by which events can be sequenced.
- State changes are controlled by clocks.
- A “clock” is a special circuit that sends electrical pulses through a circuit.
- Clocks produce electrical waveforms such as the one shown below.



Sequential Circuits

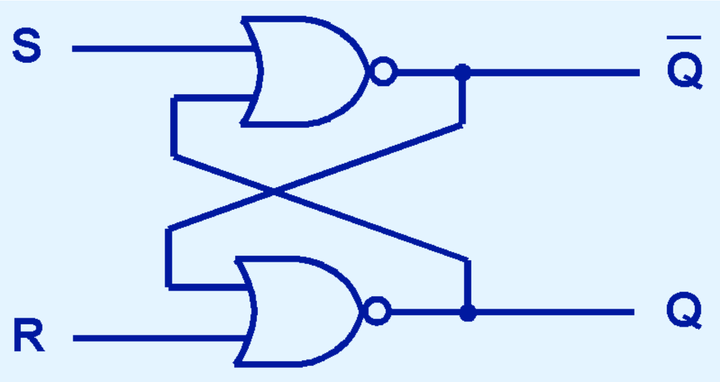
- State changes occur in sequential circuits only when the clock ticks.
- Circuits can change state on the rising edge, falling edge, or when the clock pulse reaches its highest voltage.

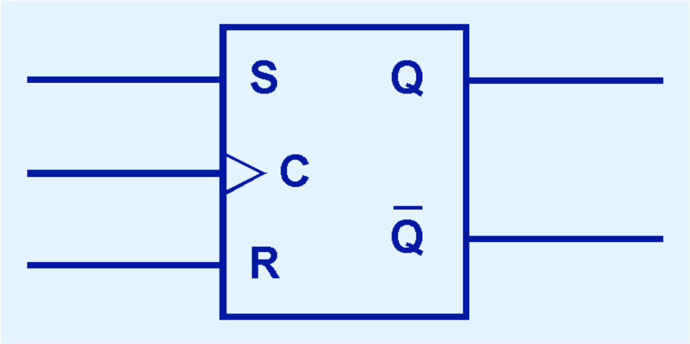


Sequential Circuits

- SR flip-flop
- Q(t) means the value of the output at time t. Q(t+1) is the value of Q after the next clock pulse.

S	R	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined





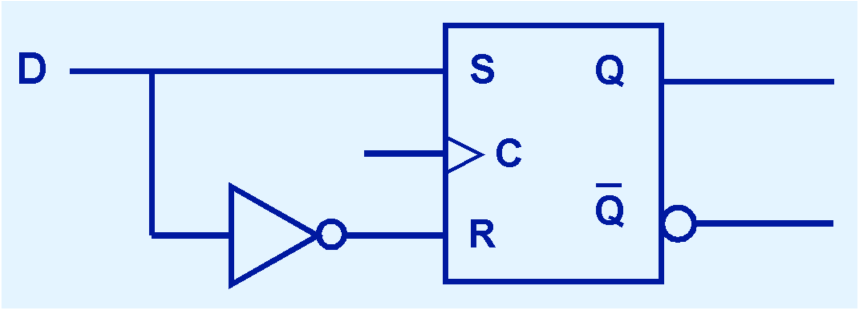
Sequential Circuits

- The SR flip-flop actually has three inputs: S, R, and its current output, Q.
- Thus, we can construct a truth table for this circuit, as shown at the right.
- Notice the two undefined values. When both S and R are 1, the SR flip-flop is unstable.

S	R	Present State Q(t)	Next State Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	undefined
1	1	1	undefined

Sequential Circuits

- D flip-flop
- You will notice that the output of the flip-flop remains the same during subsequent clock pulses. The output changes only when the value of D changes.
- The D flip-flop is the fundamental circuit of computer memory.



D	Q(t+1)
0	0
1	1

2진수와 16진수

- 10진수

$$1024 = 1\times10^3 + 0\times10^2 + 2\times10^1 + 4\times10^0$$
$$= 1000 + 0 + 20 + 4$$

$$9999 = 9\times10^3 + 9\times10^2 + 9\times10^1 + 9\times10^0$$
$$= 9000 + 900 + 90 + 9$$

2진수와 16진수

- 2진수(4자리)

$$1011_2 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0$$
$$= 8 + 0 + 2 + 1 = 11$$

$$1111_2 = 1\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0$$
$$= 8 + 4 + 2 + 1 = 15$$

2진수와 16진수

- 2진수(8자리)

$$10011011_2 = 1\times 2^7 + 0\times 2^6 + 0\times 2^5 + 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0$$

$$= 127 + 0 + 0 + 16 + 8 + 0 + 2 + 1 = 154$$

$$11111111_2 = 1\times 2^7 + 1\times 2^6 + 1\times 2^5 + 1\times 2^4 + 1\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0$$

$$= 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

2진수와 16진수

- 16진수(4자리)

$$1024_{16} = 1\times16^3 + 0\times16^2 + 2\times16^1 + 4\times16^0$$
$$= 4096 + 0 + 32 + 4 = 4132$$

10진수	2진수	16진수	10진수	2진수	16진수
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

2진수와 16진수

- 16진수(4자리)

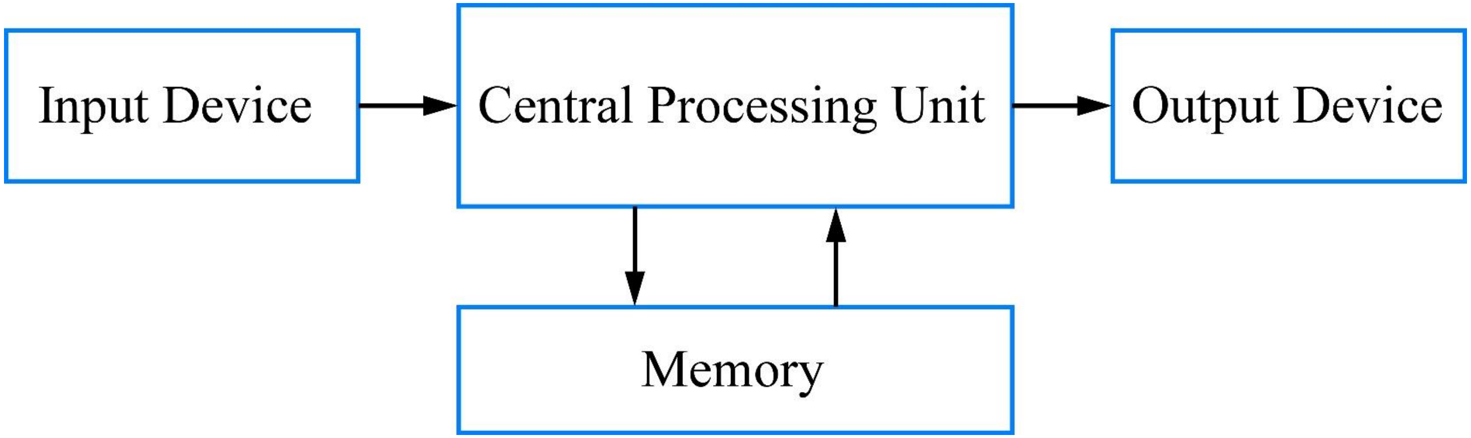
$$A0CD_{16} = 10\times16^3 + 0\times16^2 + 12\times16^1 + 13\times16^0$$
$$= 40960 + 0 + 192 + 13 = 41165$$

- 16진수를 이용한 2진수의 표기

$$10011011_2 = 9B_{16},\ 11111111_2 = FF_{16}$$

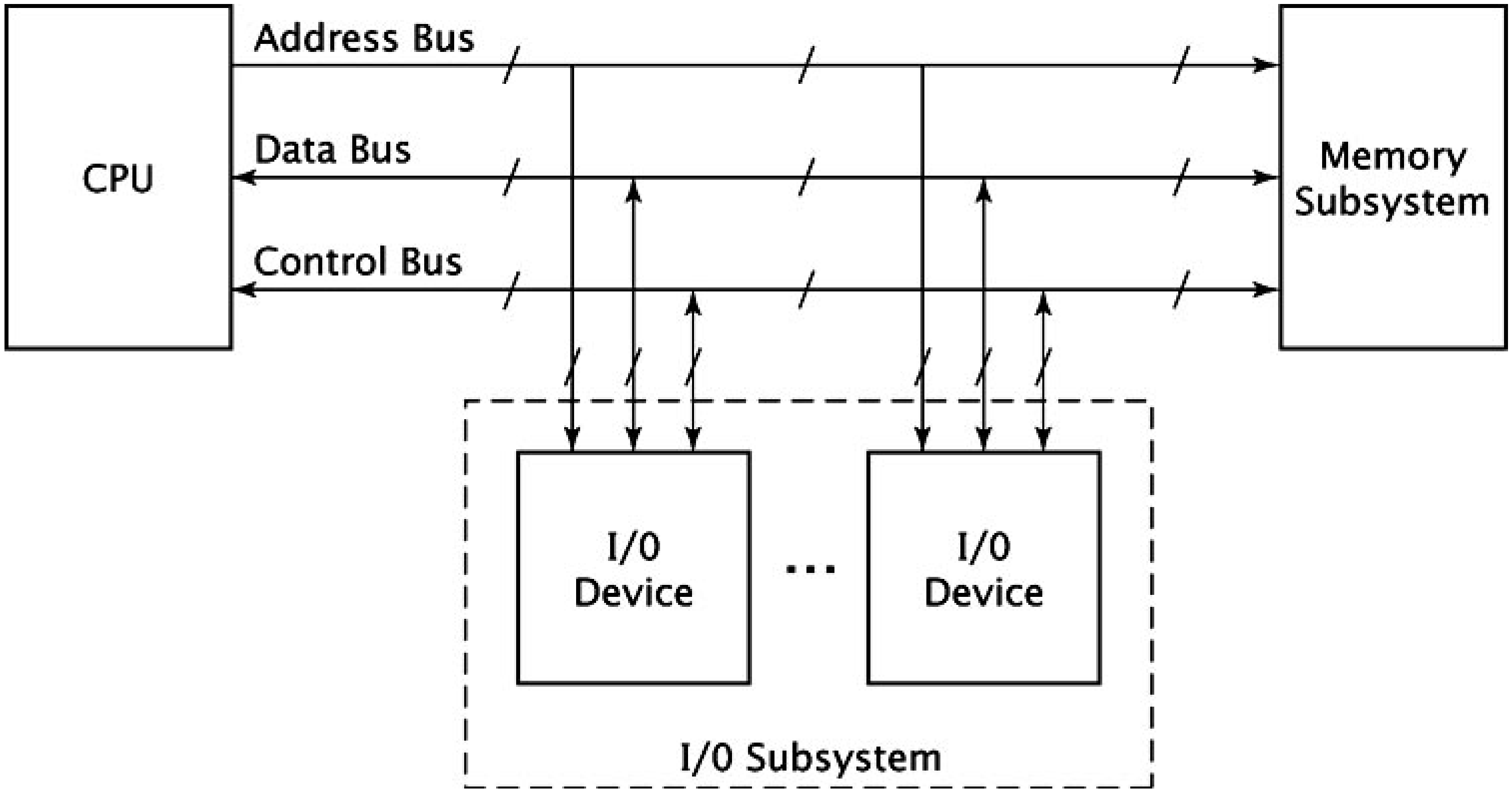
Stored Program Computer

- John von Neumann(1903~1957)



- Alan Turing(1912~1954)

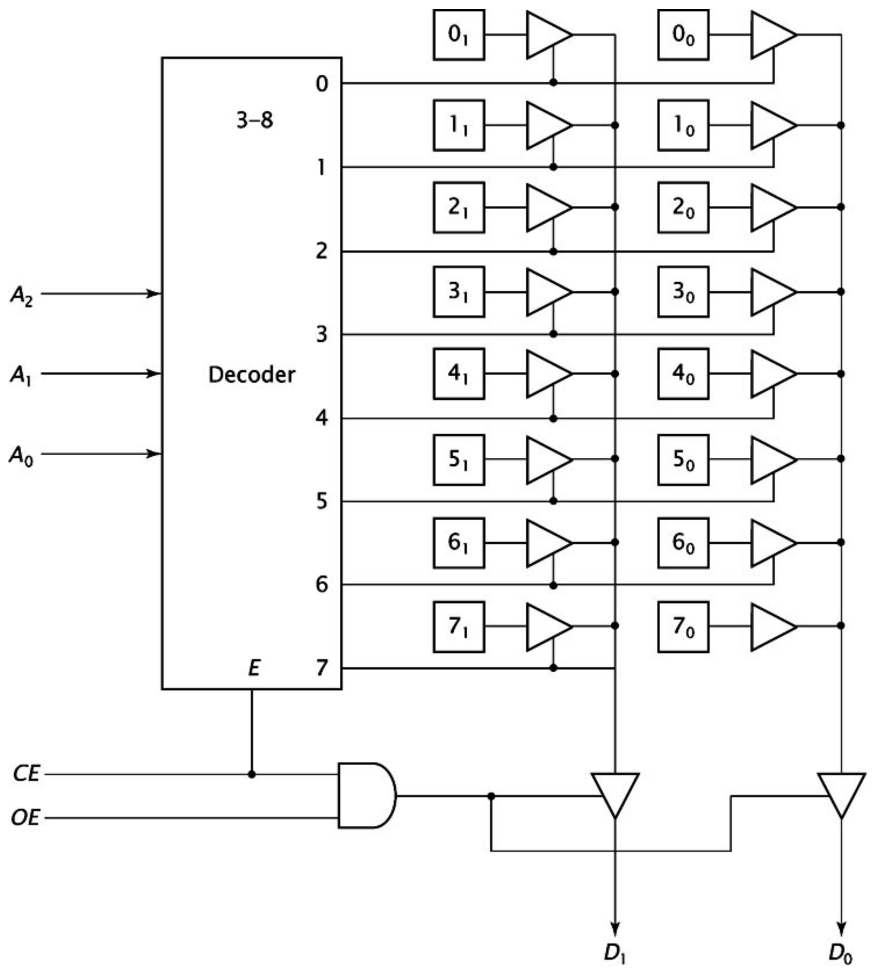
Basic Computer Organization

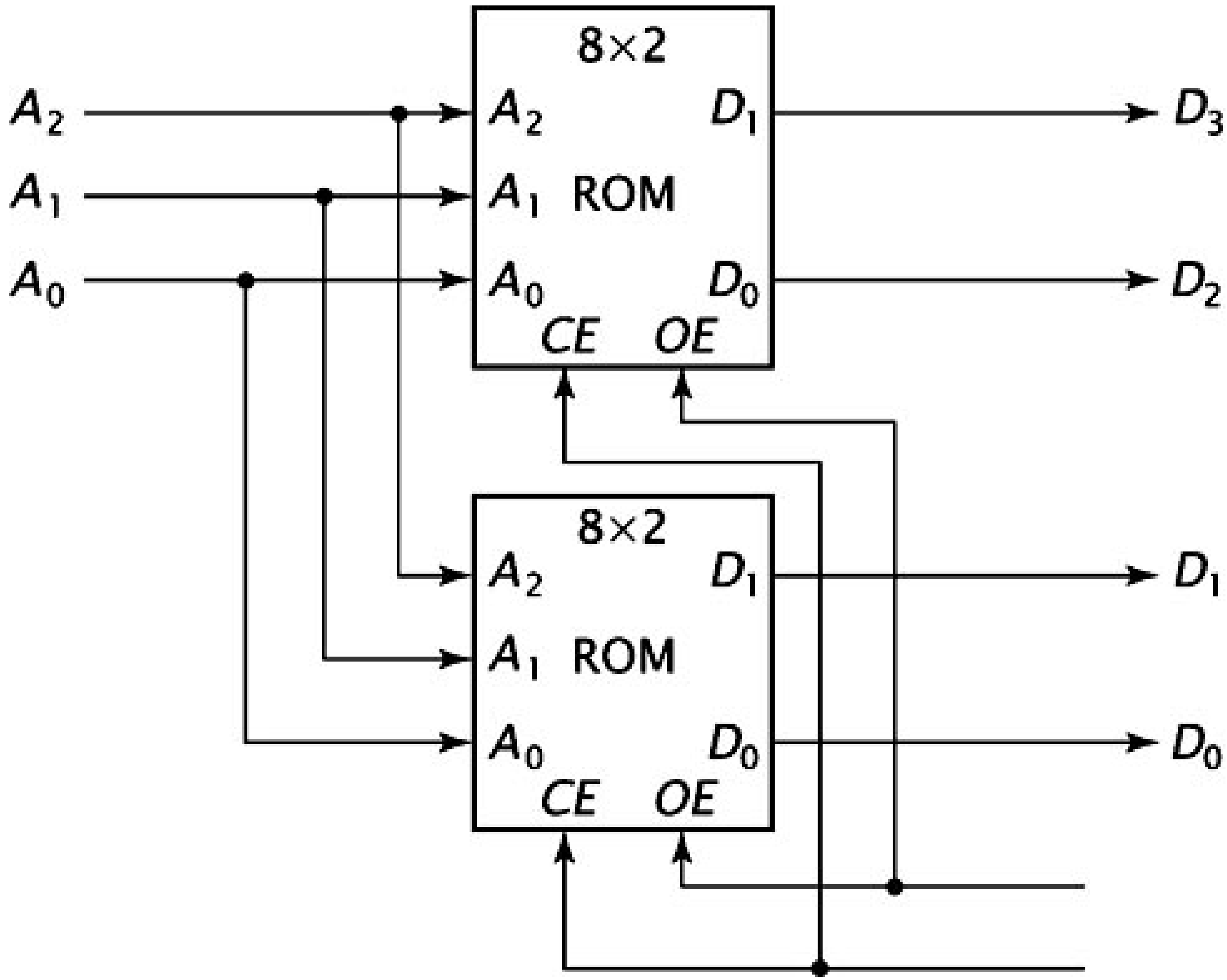


Types of Memory

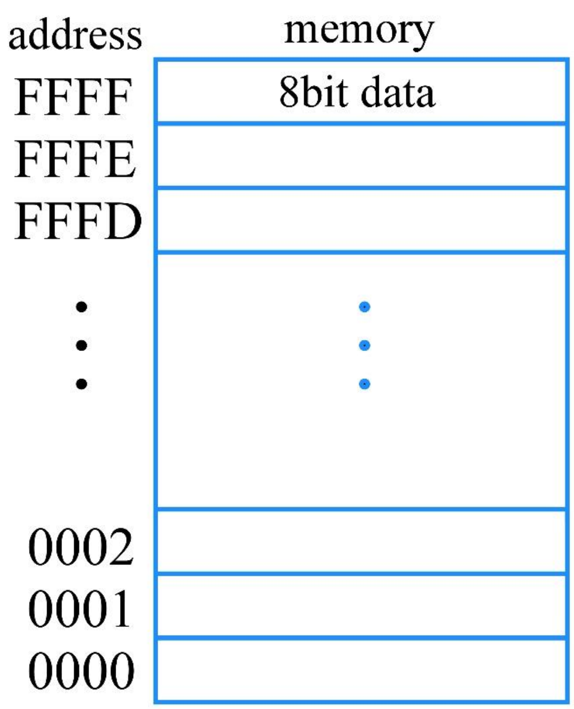
- Static RAM
- Dynamic RAM
- ROM
- PROM
- EPROM
- EEPROM

Memory Chip Organization

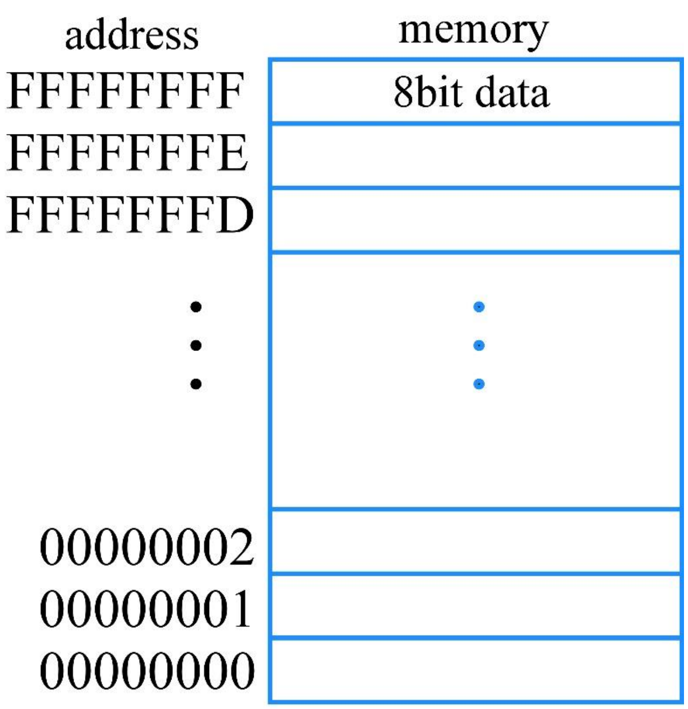



8×2
A2
A1
A0
A2
D1
A1 ROM
A0
D0
CE
OE
D3
D2
8×2
A2
D1
A1 ROM
A0
D0
CE
OE
D1
D0

Memory Map



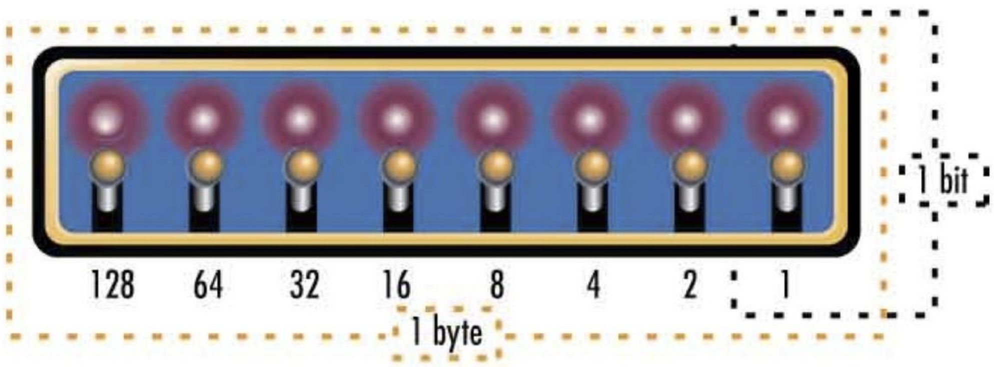
(a)



(b)

Bit and Byte

- Bit: From Binary digit
  - Smallest unit of information computer can process
  - Can have one of two values: 0 or 1
- Byte
  - Collection of 8 bits
  - Can represent 256 different messages (256 = 28)



Bits as Codes

- Codes represent each letter, digit, and special character
- ASCII: Most widely used
  - Each character is a unique 8-bit code
  - 256 unique codes for 26 letters, 10 digits, special characters
- Unicode: Supports more than 100,000 unique characters

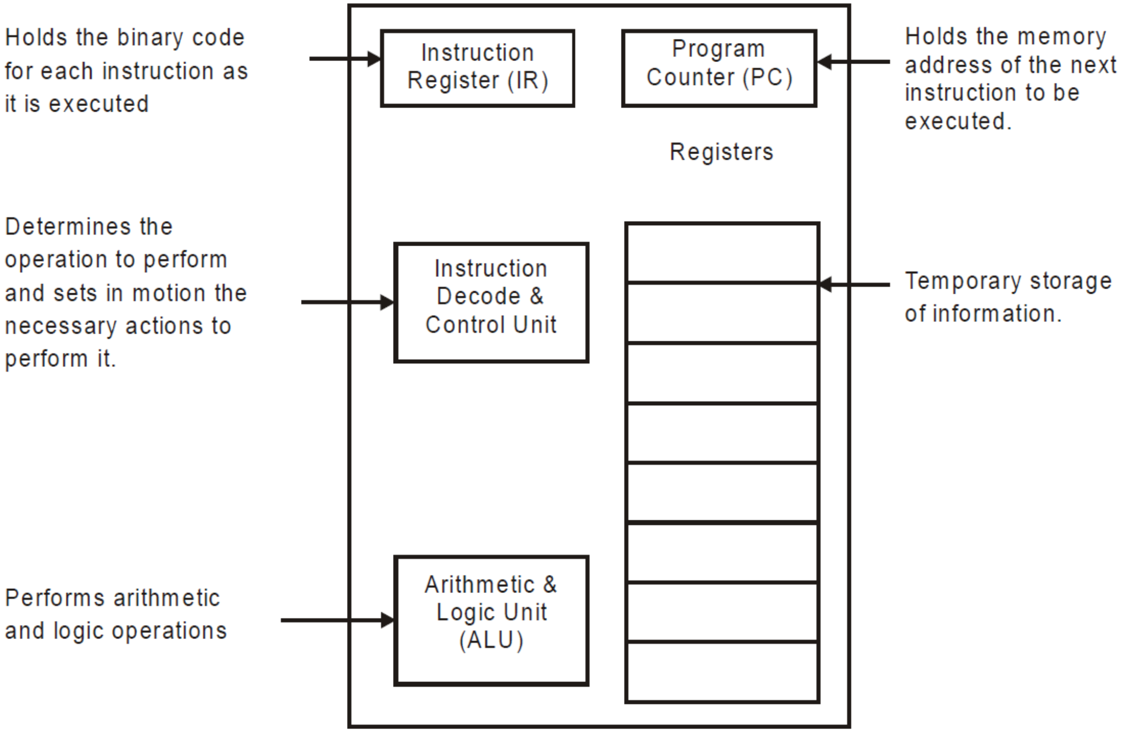
Character	ASCII binary code
A	0 1 0 0 0 0 0 1
B	0 1 0 0 0 0 1 0
C	0 1 0 0 0 0 1 1
D	0 1 0 0 0 1 0 0
E	0 1 0 0 0 1 0 1
F	0 1 0 0 0 1 1 0
G	0 1 0 0 0 1 1 1
H	0 1 0 0 1 0 0 0
I	0 1 0 0 1 0 0 1
J	0 1 0 0 1 0 1 0
K	0 1 0 0 1 0 1 1
L	0 1 0 0 1 1 0 0
M	0 1 0 0 1 1 0 1
N	0 1 0 0 1 1 1 0
O	0 1 0 0 1 1 1 1
P	0 1 0 1 0 0 0 0
Q	0 1 0 1 0 0 0 1
R	0 1 0 1 0 0 1 0
S	0 1 0 1 0 0 1 1
T	0 1 0 1 0 1 0 0
U	0 1 0 1 0 1 0 1
V	0 1 0 1 0 1 1 0
W	0 1 0 1 0 1 1 1
X	0 1 0 1 1 0 0 0
Y	0 1 0 1 1 0 0 1
Z	0 1 0 1 1 0 1 0
0	0 0 1 1 0 0 0 0
1	0 0 1 1 0 0 0 1
2	0 0 1 1 0 0 1 0
3	0 0 1 1 0 0 1 1
4	0 0 1 1 0 1 0 0
5	0 0 1 1 0 1 0 1
6	0 0 1 1 0 1 1 0
7	0 0 1 1 0 1 1 1
8	0 0 1 1 1 0 0 0
9	0 0 1 1 1 0 0 1

Embedded Controller

- Simply an embedded controller is a controller that is embedded in a greater system. One can define an embedded controller as a controller (or computer) that is embedded into some device for some purpose other than to provide general purpose computing.
- An embedded controller controls something (for example controlling a device such as a microwave oven, car braking system or a cruise missile).
- Microcontrollers and microprocessors are widely used in embedded systems. Though microcontrollers are preferred over microprocessors for embedded systems due to low power consumption.

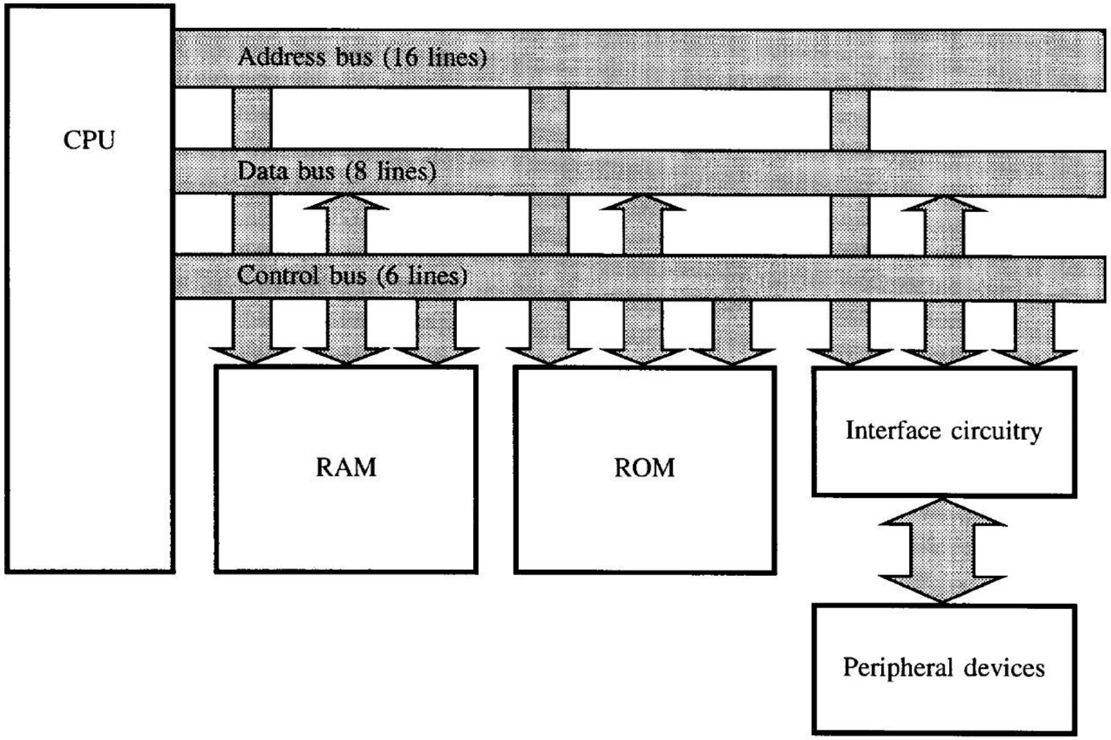
Microprocessor

- A CPU built into a single VLSI chip is called microprocessor. It contains arithmetic and logic unit (ALU), Instruction decode and control unit, Instruction register, Program counter (PC), clock circuit (internal or external), reset circuit (internal or external) and registers.



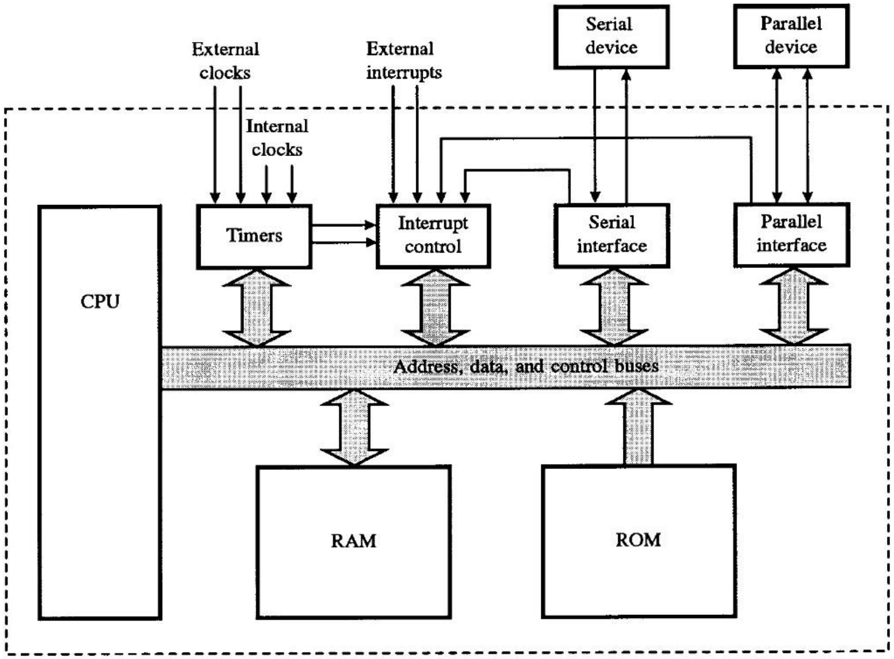
Microcomputer

- A digital computer having microprocessor as the CPU along with I/O devices and memory is known as microcomputer.



Microcontroller

- A microcontroller is a highly integrated chip, which includes on single chip, all or most of the parts needed for a controller. The microcontroller typically includes: CPU (Central Processing Unit), RAM (Random Access Memory), EPROM/PROM/ROM (Erasable Programmable Read Only Memory), I/O (input/output) – serial and parallel, timers, interrupt controller. For example, Intel 8051 is 8-bit microcontroller and Intel 8096 is 16-bit microcontroller.



Microprocessor and Microcontroller

- **Microprocessor** is a single chip CPU, **microcontroller** contains, a CPU and much of the remaining circuitry of a complete microcomputer system in a single chip.
- Microcontroller includes RAM, ROM, serial and parallel interface, timer, interrupt schedule circuitry (in addition to CPU) in a single chip.
  - RAM is smaller than that of even an ordinary microcomputer, but enough for its applications.
  - Interrupt system is an important feature, as microcontrollers have to respond to control oriented devices in real time. E.g., opening of microwave oven's door cause an interrupt to stop the operation. (Most microprocessors can also implement powerful interrupt schemes, but external components are usually needed).

Microprocessor and Microcontroller

- Microprocessors are most commonly used as the CPU in microcomputer systems. Microcontrollers are used in small, minimum component designs performing control-oriented activities.
- Microprocessor instruction sets are processing intensive, implying powerful addressing modes with instructions catering to large volumes of data. Their instructions operate on nibbles, bytes, etc. Microcontrollers have instruction sets catering to the control of inputs and outputs. Their instructions operate also on a single bit. E.g., a motor may be turned ON and OFF by a 1-bit output port.

Central Processing Unit(CPU)

- CPU is the brain of the computer system, administers all activity in the system and performs all operations on data. It continuously performs two operations: fetching and executing instructions. It understand and execute instructions based on a set of binary codes called the instruction set.

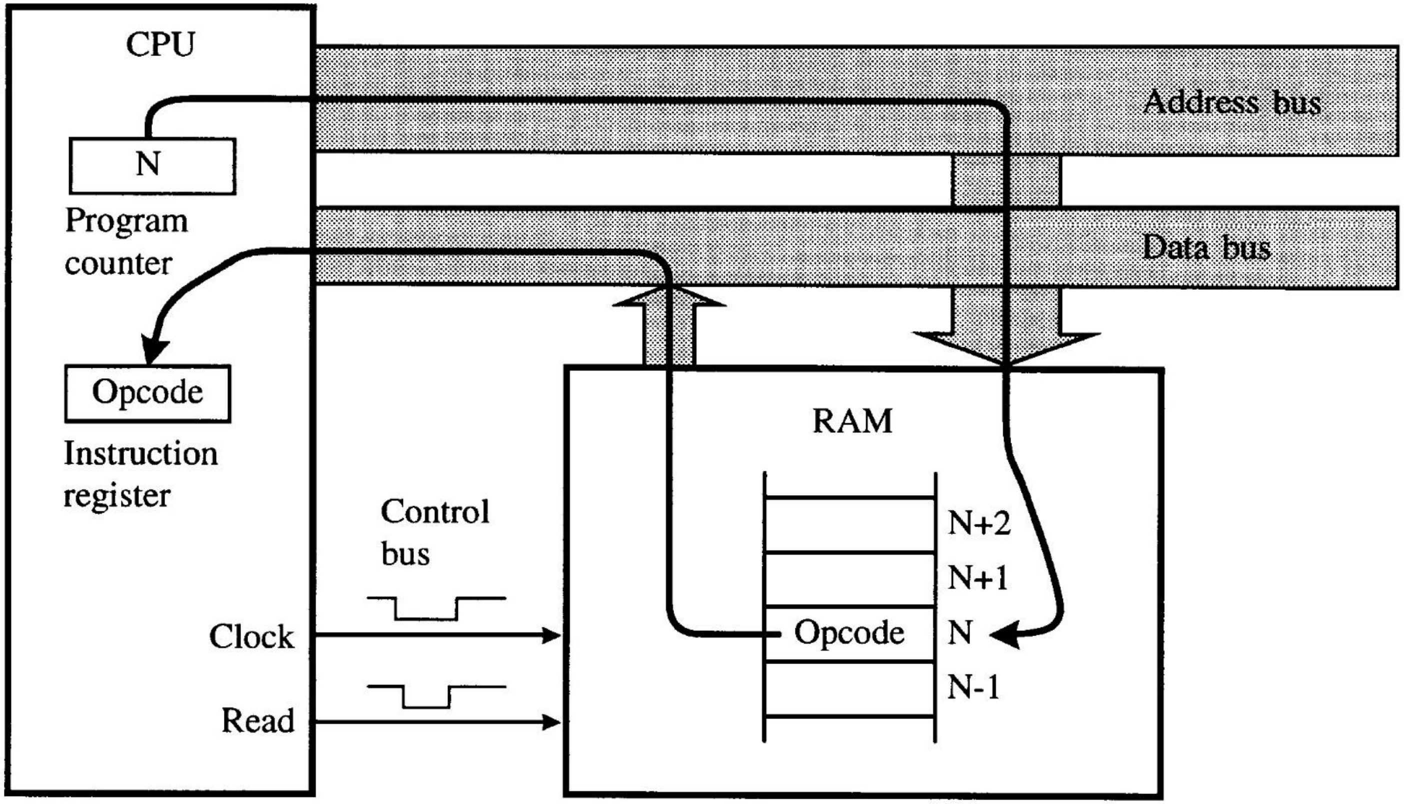
Machine Cycle

- To execute an instruction–the processor must:
  1. Fetch the instruction from memory
  2. Decode the instruction
  3. Execute the instruction
  4. Store the result back in the memory. These four steps refer to Machine Cycle.
- Generally one machine cycle = X clock cycles (“X” depends on the particular instruction being executed). Shorter the clock cycle, lesser the time it takes to complete one machine cycle, so instructions are executed faster. Hence, faster the processor.

Fetching and Executing an Instruction

- Fetching involves the following steps:
  1. Contents of PC are placed on address bus.
  2. READ signal is activated.
  3. Data (instruction opcode) are read from RAM and placed on data bus.
  4. Opcode is latched into the CPU's internal instruction register.
  5. PC is incremented to prepare for the next fetch from memory.
- While execution involves decoding the opcode and generating control signals to gate internal registers in and out of the ALU and to signal the ALU to perform the specified operation.

Bus activity for an opcode fetch cycle



Semiconductor Memory

- RAM: read/write memory, volatile (the contents are lost when power is removed)
- ROM: read-only memory, nonvolatile

The Buses: Address, Data, and Control

- A BUS is a collection of wires carrying information with a common purpose. For each read or write operation, the CPU specifies the location of the data or instruction by placing an address on the address bus, then activates a signal on the control bus indicating whether the operation is read or write.
  - READ OPERATIONS retrieve a byte of data from memory at the location specified and place it on the data bus. CPU reads the data and places it in one of its internal registers.
  - WRITE OPERATIONS put data from CPU on the data bus and store it in the location specified.

The Buses: Address, Data, and Control

- ADDRESS BUS carries the address of a specified location. For n address lines, 2^n locations can be accessed. E.g., A 16-bit address bus can access 2^{16} = 65,536 locations or 64K locations (2^{10}= 1024 = 1K, 2^6 = 64).
- DATA BUS carries information between the CPU and memory or between the CPU and I/O devices.
- CONTROL BUS carries control signals supplied by the CPU to synchronize the movement of information on the address and data bus.

8, 16, and 32-bit Microcontrollers

- When the ALU performs arithmetic and logical operations on a byte (8-bits) at an instruction, the microcontroller is an 8-bit microcontroller. The internal bus width of 8-bit microcontroller is of 8-bit. Examples of 8-bit microcontrollers are Intel 8051 family and Motorola MC68HC11 family.

8, 16, and 32-bit Microcontrollers

- When the ALU performs arithmetic and logical operations on a word (16-bits) at an instruction, the microcontroller is an 16-bit microcontroller. The internal bus width of 16-bit microcontroller is of 16-bit. Examples of 16-bit microcontrollers are Intel 8096 family and Motorola MC68HC12 and MC68332 families. The performance and computing capability of 16 bit microcontrollers are enhanced with greater precision as compared to the 8-bit microcontrollers.

8, 16, and 32-bit Microcontrollers

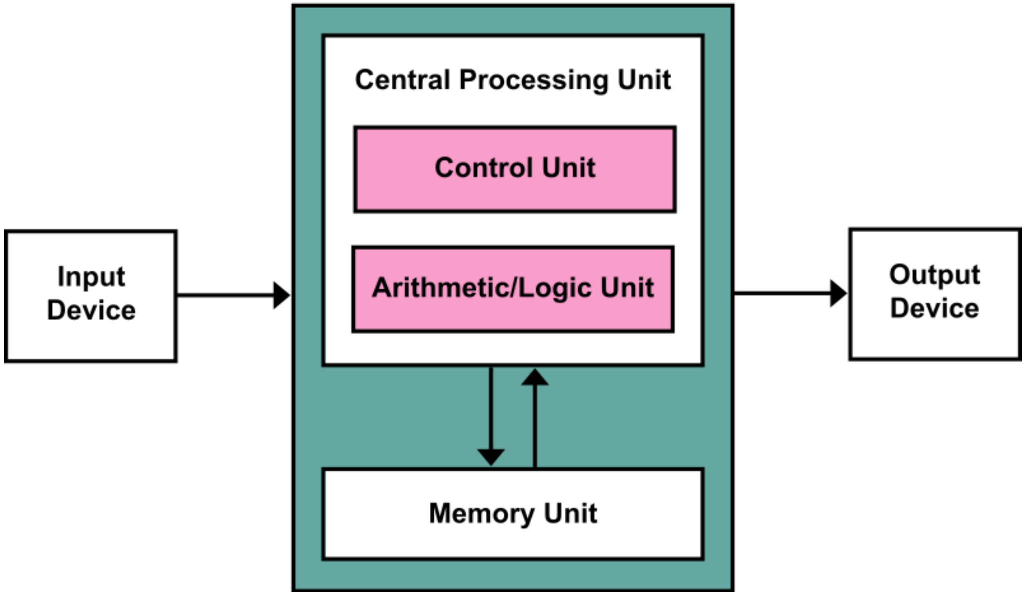
- When the ALU performs arithmetic and logical operations on a double word (32-bits) at an instruction, the microcontroller is an 32-bit microcontroller. The internal bus width of 32-bit microcontroller is of 32-bit. Examples of 32-bit microcontrollers are Intel 80960 family and Motorola M683xx and Intel/Atmel 251 family. The performance and computing capability of 32 bit microcontrollers are enhanced with greater precision as compared to the 16-bit microcontrollers.

Von-Neuman Architecture

- Microcontrollers based on the Von-Neuman architecture have a single “data” bus that is used to fetch both instructions and data. Program instructions and data are stored in a common main memory. When such a controller addresses main memory, it first fetches an instruction, and then it fetches the data to support the instruction. The two separate fetches slows up the controller’s operation. The Von-Neuman architecture’s main advantage is that it simplifies the microcontroller design because only one memory is accessed.

Von-Neuman Architecture

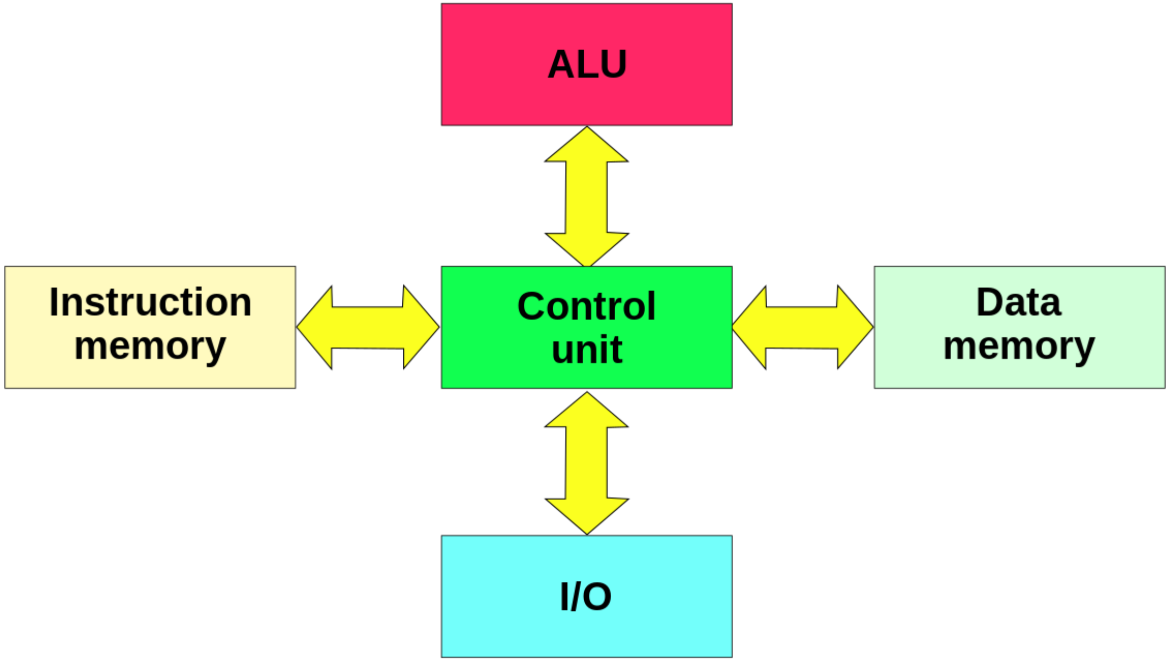
- In microcontrollers, the contents of RAM can be used for data storage and program instruction storage. For example, the Motorola 68HC11 microcontroller Von-Neuman architecture.



Harvard Architecture

- Microcontrollers based on the Harvard Architecture have separate data bus and an instruction bus. This allows execution to occur in parallel. As an instruction is being “pre-fetched”, the current instruction is executing on the data bus. Once the current instruction is complete, the next instruction is ready to go. This pre-fetch theoretically allows for much faster execution than Von-Neuman architecture, on the expense of complexity. The Harvard Architecture executes instructions in fewer instruction cycles than the Von-Neuman architecture. For example,the intel MCS-51 family of microcontrollers and PIC microcontrollers uses Harvard Architecture.

Harvard Architecture



CISC Architecture

- Almost all of today's microcontrollers are based on the CISC (Complex Instruction Set Computer) concept. When an microcontroller has an instruction set that supports many addressing modes for the arithmetic and logical instructions, data transfer and memory accesses instructions, the microcontroller is said to be of CISC architecture.
- The typical CISC microcontroller has well over 80 instructions, many of them very powerful and very specialized for specific control tasks. It is quite common for the instructions to all behave quite differently. Some might only operate on certain address spaces or registers, and others might only recognize certain addressing modes.

CISC Architecture

- The advantages of the CISC architecture are that many of the instructions are macrolike, allowing the programmer to use one instruction in place of many simpler instructions. An example of CISC architecture microcontroller is Intel 8096 family.

RISC Architecture

- The industry trend for microprocessor design is for Reduced Instruction Set Computers (RISC) designs. When a microcontroller has an instruction set that supports fewer addressing modes for the arithmetic and logical instructions and for data transfer instructions, the microcontroller is said to be of RISC architecture.

RISC Architecture

- The benefits of RISC design simplicity are a smaller chip, smaller pin count, and very low power consumption. Some of the typical features of a RISC processor-Harvard architecture are
  1. Allows simultaneous access of program and data.
  2. Overlapping of some operations for increased processing performance.
  3. Instruction pipelining increases execution speed.
  4. Orthogonal (symmetrical) instruction set for programming simplicity.
  5. Allows each instruction to operate on any register or use any addressing mode.

8-Bit Microcontroller Market Trend

MICROCHIP

Worldwide 8-Bit Microcontroller Market Share (Dollars)

No.	1991 Rank	1996 Rank	1998 Rank	2001 Rank	2005 Rank	2006-9 Rank	2010 Rank	2014 Rank
1	Motorola	Motorola	Motorola	Motorola	Motorola	Microchip	Renesas	Microchip
2	Intel	NEC	NEC	Hitachi	Renesas	NEC	Microchip	Renesas
3	Philips	Philips	ST-Micro	NEC	Microchip	ST-Micro	Atmel	NXP
4	Mitsubishi	Hitachi	Philips	Microchip	NEC	Freescale	ST-Micro	Atmel
5	NEC	Mitsubishi	Hitachi	ST-Micro	ST-Micro	Atmel	Samsung	ST-Micro
6	Hitachi	Toshiba	Mitsubishi	Philips	Atmel	Renesas	Freescale	Freescale
7	Toshiba	Matsushita	Microchip	Toshiba	Toshiba	NXP	NXP	Cypress
8	Siemens	SGS-Thomson	Toshiba	Atmel	Philips	Cypress	Cypress	Datang
9	TI	Intel	Siemens	Matsushita	Fujitsu	Sony	Panasonic	Si Labs
10	Matsushita	Microchip	TI	Sanyo	Infineon	Fujitsu	Fujitsu	Samsung
11	National	Siemens	Fujitsu	Samsung	Sanyo	Panasonic	Datang	CEC Huada
12	SGS-Thomson	Fujitsu	Sanyo	Mitsubishi	Samsung	Toshiba	NEC (1Q)	Holtek
13	Ricoh	TI	Matsushita	Infineon	Matsushita	Samsung	Sony	Spansion
14	MHS	Sony	Atmel	Sony	Sony	Datang	Toshiba	Tongfang
15	IIT	Zilog	Zilog	TI	Sunplus	Si Labs	Si Labs	SHIC
16	Sharp	Sharp	Sharp	Fujitsu	Micronas	Holtek	JSC	Panasonic
17	Fujitsu	Temic	Sony	Sunplus	Novatek	Infineon	Holtek	Sony
18	Oki	Sanyo	Intel	Zilog	Intel	Elan	Infineon	SH Fudan
19	Zilog	National	National	Novatek	Holtek	Winbond	Sonix	Infineon
20	Sony	Oki	LG Semi	Micronas	Winbond	Denso	Elan	Ixys
23	Microchip							

Based on dollar shipment volume 1991-2014, Source: Gartner and Microchip

8-Bit Microcontroller Market Trend

- 2015: NXP + Freescale = NXP
- 2016: Microchip + Atmel = Microchip

Leading MCU Suppliers ($M)

2016 Rank	Company	2015	2016	% Change	% Marketshare
1	NXP*	1,350	2,914	116%	19%
2	Renesas	2,560	2,458	-4%	16%
3	Microchip**	1,355	2,027	50%	14%
4	Samsung	2,170	1,866	-14%	12%
5	ST	1,514	1,573	4%	10%
6	Infineon	1,060	1,106	4%	7%
7	Texas Instruments	820	835	2%	6%
8	Cypress***	540	622	15%	4%

*Acquired Freescale in December 2015.

**Purchased Atmel in April 2016.

***Includes full year of sales from Spansion acquisition in March 2015.

Source: IC Insights, company reports