CHW 469 : Embedded Systems

- ~~Introduction to Embedded System.~~
- ~~Review of Electronics Part (KCL, KVL, Parallel/Series Resistance, DAC and ADC).~~
- Review for Digital Systems (Binary Number, Logic, MOS Implantation, Computer Architecture).
- Intro to Programming Concepts (Structure and Concurrent).

Assignment no. 1

In the design phase of the production line of fans. it required by the customer that fun to be controlled by three speeds (low, medium and high). your team leader asks you to identify the interface between the microcontroller and the fan motor. support your proposal by the circuit design.

Hint: motor driver, resistance ladder.

Binary System

- **Two discrete values:**
  - 1's and 0's
  - 1, TRUE, HIGH
  - 0, FALSE, LOW
- **1 and 0:** voltage levels, rotating gears, fluid levels, etc.
- Digital circuits use **voltage** levels to represent 1 and 0
- ***Bit*****:** *B*inary dig*it*
- *N*-bit binary number
  - How many values? **2^N** **-** Range: **$[0, 2^N - 1]$**
  - Example: 3-digit binary number:
    - **2^3 = 8 possible values - Range: [0, 7] = [000_2 to 111_2]**

8's column 4's column 2's column 1's column

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

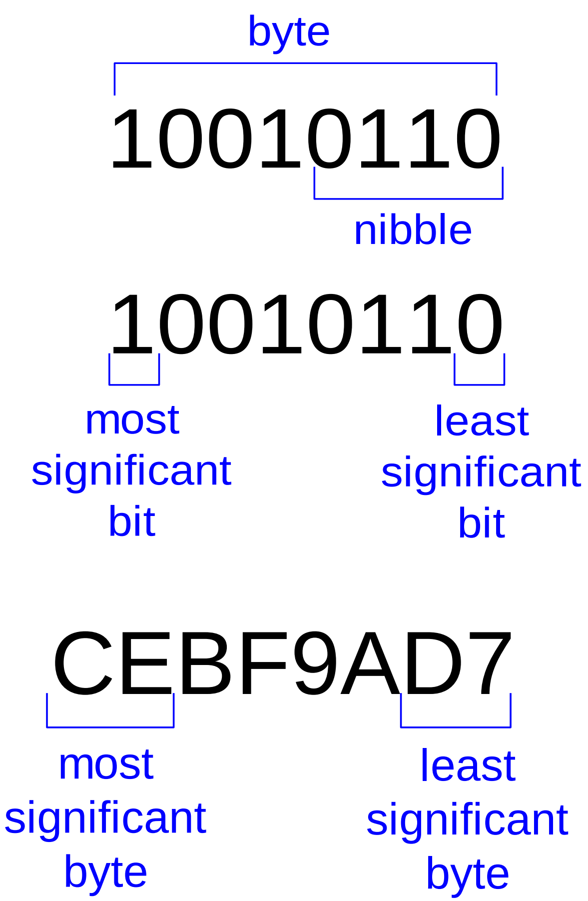
one eight — one four — no two — one one

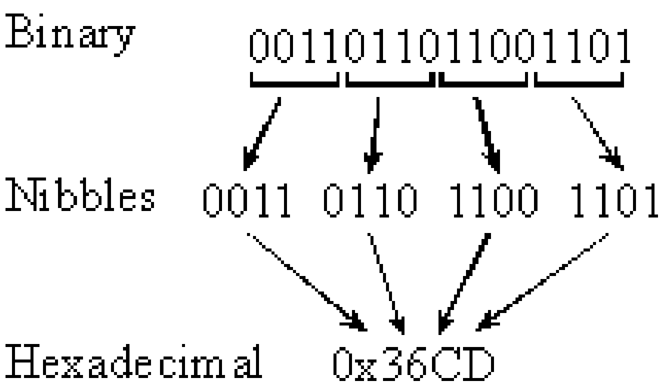
Hexadecimal Numbers

- Base 16
- Shorthand for binary

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

Bits, Bytes, Nibbles…





Boolean Algebra

- A & B = B & A … Commutative Law
- A | B = B | A … Commutative Law
- (A & B) & C = A & (B & C) … Associative Law
- (A | B) | C = A | (B | C) … Associative Law
- (A | B) & C = (A & C) | (B & C) … Distributive Law
- (A & B) | C = (A | C) & (B | C) … Distributive Law
- ~(A | B) = (~A) & (~B) … De Morgan’s Theorem
- ~(A & B) = (~A) | (~B) ... De Morgan’s Theorem

A & 0 = 0 … Identity of 0
A | 0 = A … Identity of 0
A & 1 = A … Identity of 1
A | 1 = 1 … Identity of 1

A | A = A … Property of OR
A | (~A) = 1 … Property of OR
A & A = A … Property of AND
A & (~A) = 0 … Property of AND
~(~A) = A … Inverse

Operators (Bitwise / logical)

A	B	C	A & (B | C)	A && (B || C)
01	01	11	01	True
00	01	11	00	False

Signed Binary Numbers

- Sign/Magnitude Numbers
  - 1 sign bit, N-1 magnitude bits
  - Positive number: sign bit = 0 , Negative number: sign bit = 1
  - Example, 4-bit sign/mag representations of ± 6:

    +6 = **0110**

    - 6 = **1110**
  - Range of an N-bit sign/magnitude number: **$[-(2^{N-1}-1), 2^{N-1}-1]$**
- Two's Complement Numbers
  - The most significant bit still indicates the sign (1 = negative, 0 = positive)
  - Range of an N-bit two's comp number: **[-(2N-1), 2N-1-1]**

1. 1001

2. + 1

$1010_2 = -6_{10}$

```
 111
   0110
+  1010
-------
  10000
```

Signed Binary Numbers (Cont…)

- Sign Extension
  - Sign bit copied to msb's
  - Number value is same
- **Example 1:**
  - 4-bit representation of 3 = 0011
  - 8-bit sign-extended value: 00000011
- **Example 2:**
  - 4-bit representation of -5 = 1011
  - 8-bit sign-extended value: 11111011

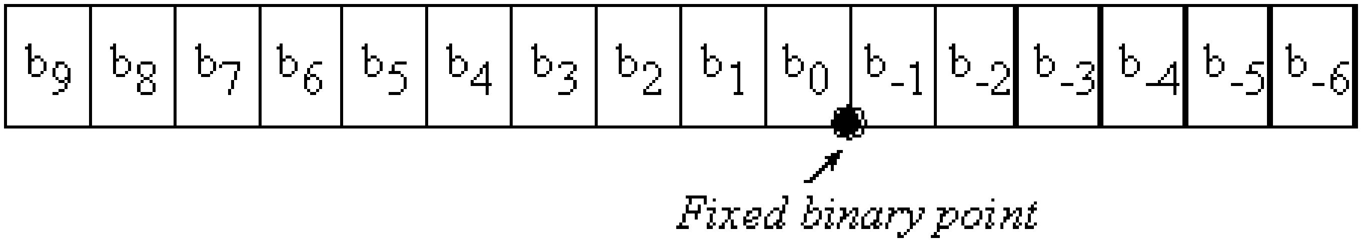
Signed Binary Numbers (Cont…)

Number System	Range
Unsigned	$[0, 2^N-1]$
Sign/Magnitude	$[-(2^{N-1}-1), 2^{N-1}-1]$
Two's Complement	$[-2^{N-1}, 2^{N-1}-1]$

For example, 4-bit representation:

	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Unsigned									0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Two's Complement	1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111								
Sign/Magnitude		1111	1110	1101	1100	1011	1010	1001	0000 1000	0001	0010	0011	0100	0101	0110	0111								

Fixed Point Binary Numbers



- **fixed-point numbers**: express values to the computer non-integer values.
- **A fixed-point number** contains two parts: variable integer, called *I*. fixed constant, called the resolution .
- The fixed constant will NOT be stored on the computer. The fixed constant is something we keep track of while designing the software operations. !!!
- The **precision** of a number system is the total number of distinguishable values that can be represented.
- The precision of a fixed-point number is the number of bits used to store the variable integer.

Binary fixed-point value = $I \cdot 2^n$

Fixed Point Binary Numbers

01101100

0110.1100

$2^2 + 2^1 + 2^{-1} + 2^{-2} = 6.75$

Addition

- Decimal

```
     11  ← carries
     3734
  +  5168
  -------
     8902
```

- Binary

```
     11  ← carries
     1011
  +  0011
  -------
     1110
```

- Digital systems operate on a **fixed number of bits**
- Overflow: when result is too big to fit in the available number of bits

```
    111
    1011
 +  0110
 -------
   10001
```

Overflow!

[Ariane 5.]

Multiplication

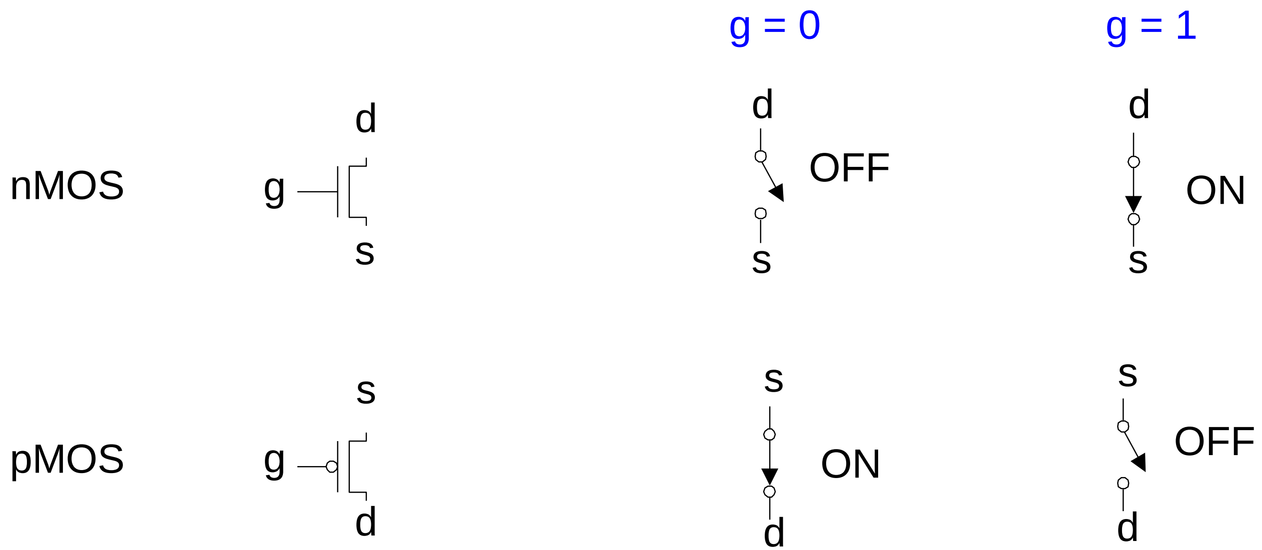
Decimal		Binary
230	multiplicand	0101
x 42	multiplier	x 0111
460	partial products	0101
+ 920		0101
9660		0101
		+ 0000
	result	0100011

230 x 42 = 9660

5 x 7 = 35

- Partial products formed by multiplying a single digit of the multiplier with multiplicand
- Shifted partial products summed to form result

Binary Information Implemented with MOS transistors

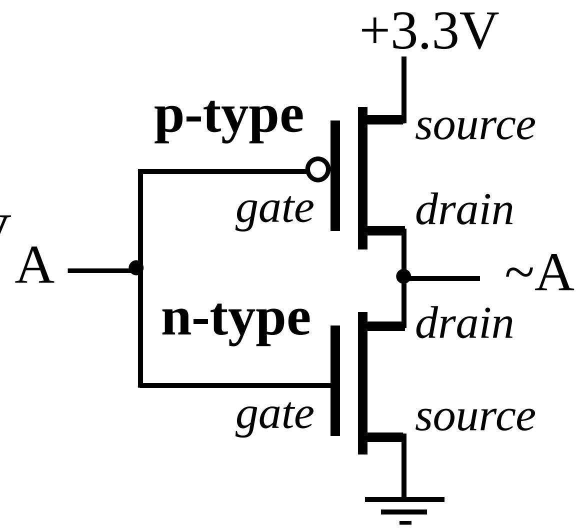


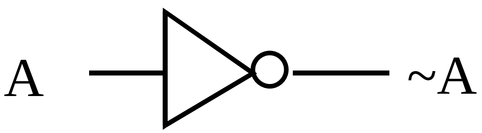
nMOS: pass good 0's, so connect source to GND

pMOS: pass good 1's, so connect source to V_{DD}

Binary Information Implemented with MOS transistors (Cont…)

A	p-type	n-type	~A
0 V	active	off	+3.3V
+3.3V	off	active	0V

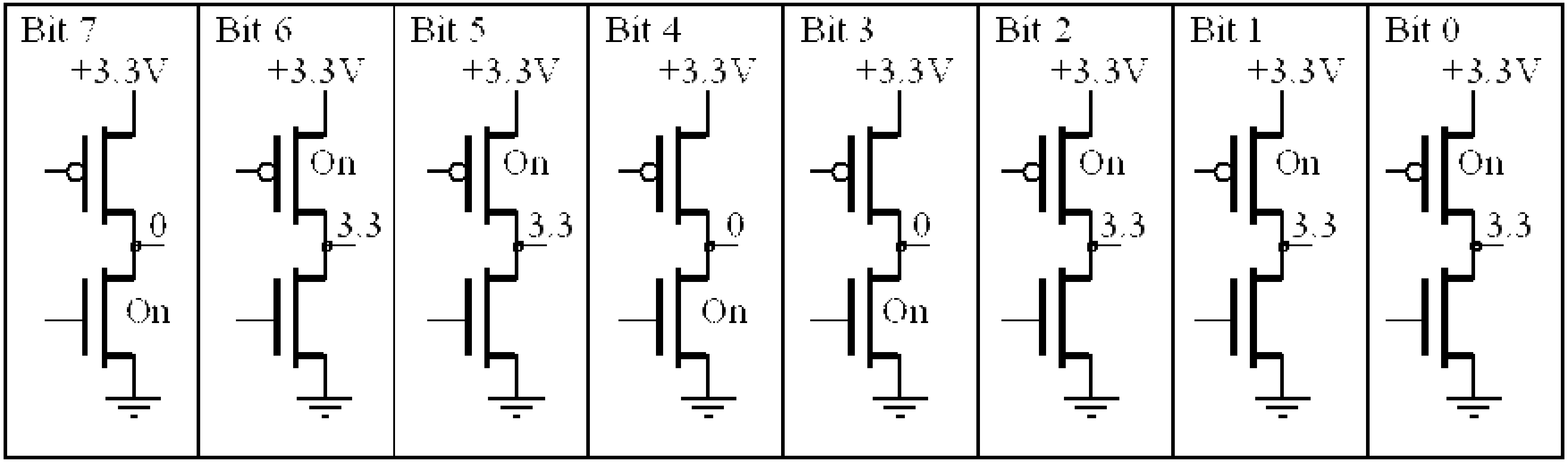




74HC04

A	~A
0	1
1	0

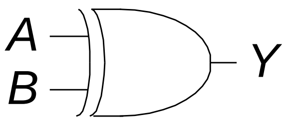
Binary Information Implemented with MOS transistors (Cont…)



A byte is comprised of 8 bits,
voltage 3.3V means true or 1 ; voltage of 0V means false or 0.
In this case representing the binary number 01100111.

Binary Information Implemented with MOS transistors (Cont…)

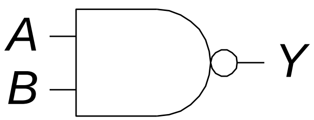
XOR



$Y = A \oplus B$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

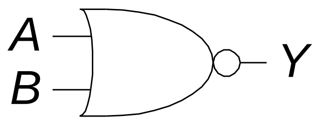
NAND



$Y = \overline{AB}$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

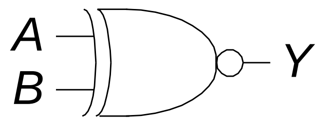
NOR



$Y = \overline{A + B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

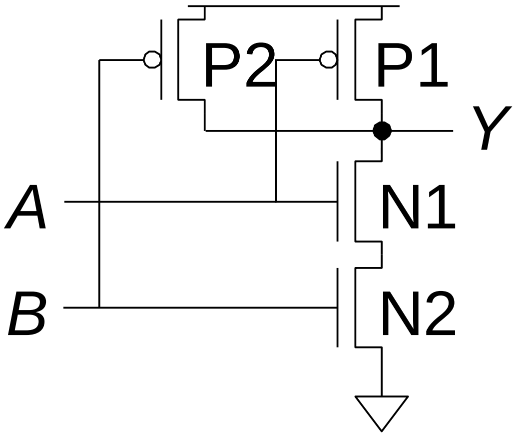
XNOR



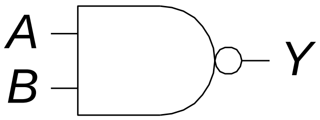
$Y = \overline{A \oplus B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Binary Information Implemented with MOS transistors (Cont…)



NAND



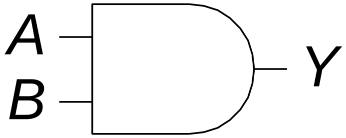
$Y = \overline{AB}$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

A	*B*	P1	P2	N1	N2	*Y*
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

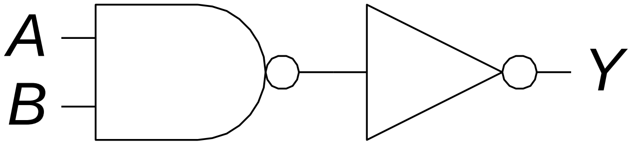
Binary Information Implemented with MOS transistors (Cont…)

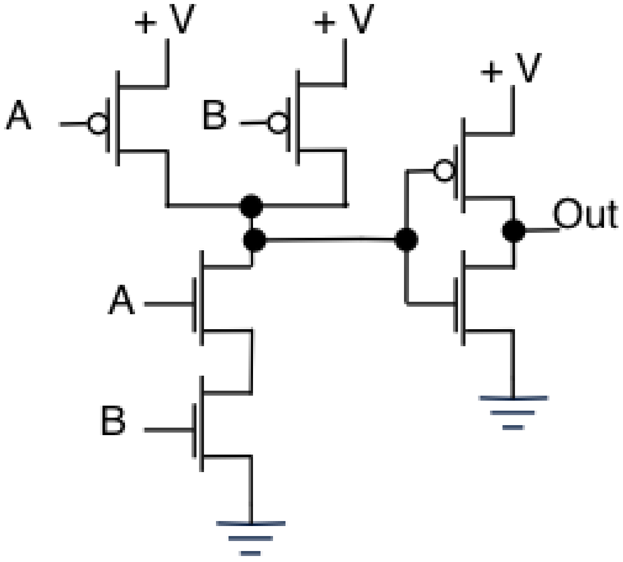
AND



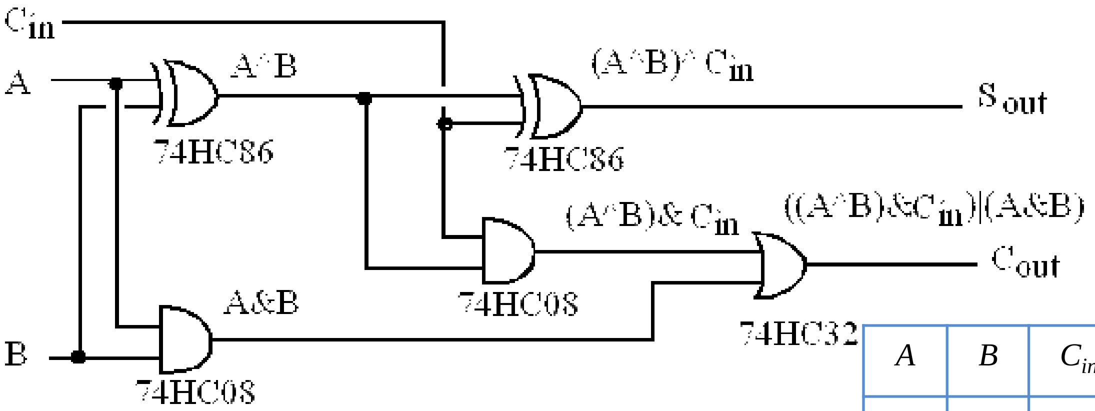
$Y = AB$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



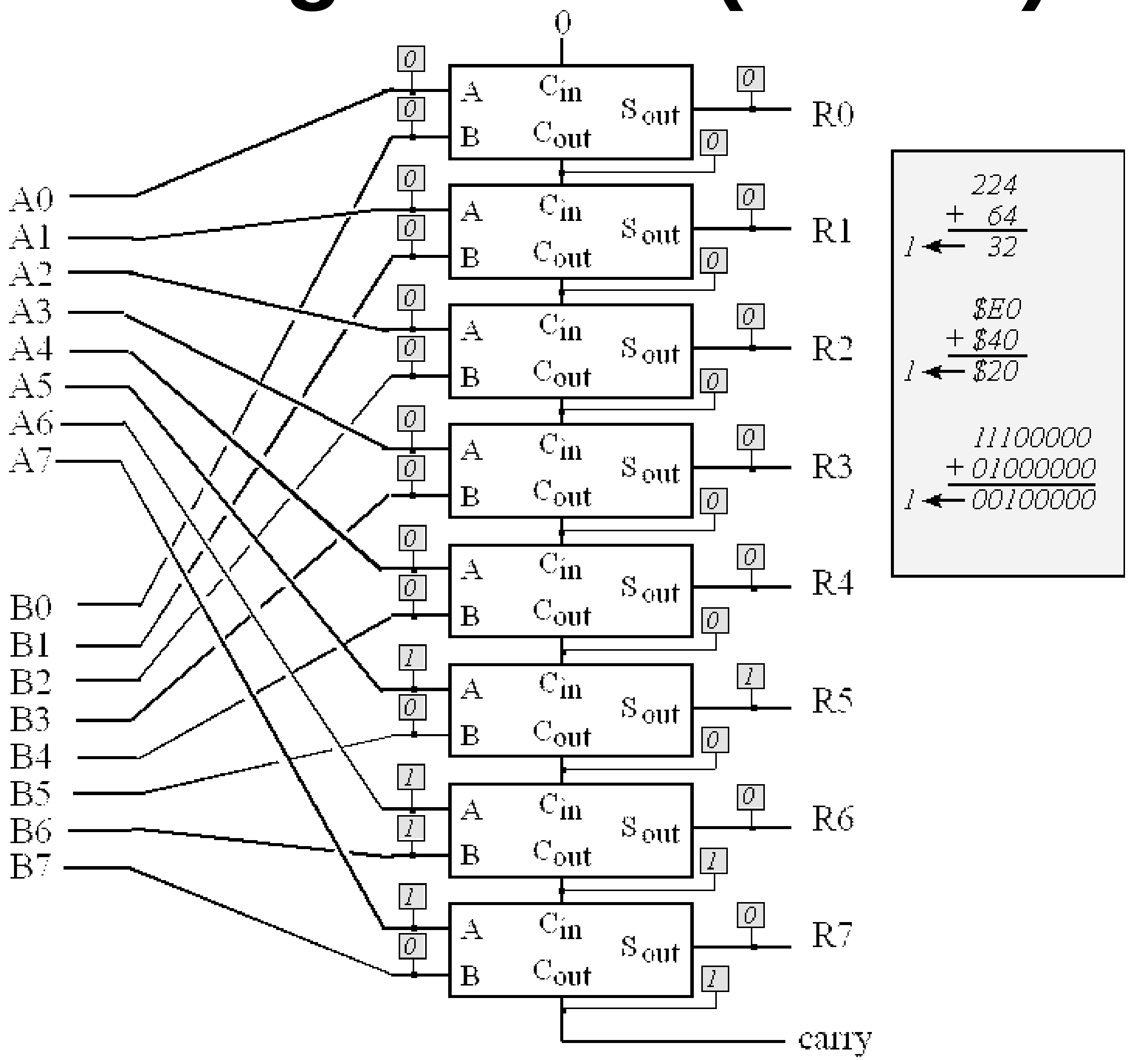


Building Blocks (Adder)



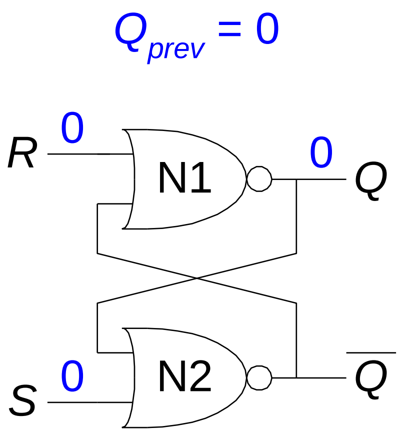
A	B	C_{in}	$A+B+C_{in}$	C_{out}	S_{out}
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	2	1	0
1	0	0	1	0	1
1	0	1	2	1	0
1	1	0	2	1	0
1	1	1	3	1	1

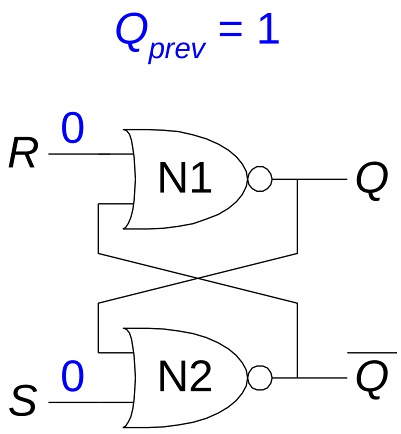
Building Blocks (Adder) cont…



Storage Elements

- Digital storage elements are essential components used to make registers and memory.
- The simplest storage element is the **set-reset latch**.
  - ***S* = 1, *R* = 0: *Set* the output**
  - ***S* = 0, *R* = 1: *Reset* the output**
  - ***S* = 0, *R* = 0:** then $Q = Q_{prev}$ **Memory!**
  - ***S* = 1, *R* = 1: Invalid State** $Q \neq \text{NOT } Q$

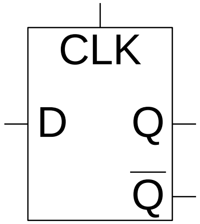




Storage Elements (Cont…)

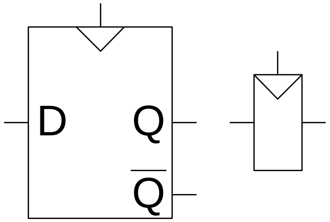
- D flip-flops are the basic building block of RAM and registers on the computer.
- D-Latch Two inputs: *CLK*, *D*
  - ***CLK*****:** controls *when* the output changes
  - ***D*** (the data input): controls *what* the output changes to
  - When ***CLK*** **= 1**, *D* passes through to *Q* (*transparent*)
  - When ***CLK*** **= 0**, *Q* holds its previous value (*opaque*)
  - Avoids invalid case when $Q \neq \text{NOT } Q$

D Latch Symbol



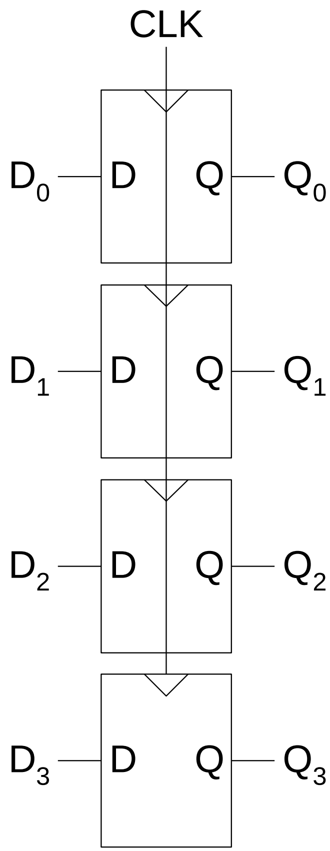
- D- flip flop Two inputs: *CLK*, *D*
  - *Samples D on rising edge of CLK*
  - *When* ***CLK rises from 0 to 1****, D passes through to Q*
  - *Otherwise, Q holds its previous value*
  - *Q changes only on rising edge of CLK, Called edge-triggered*

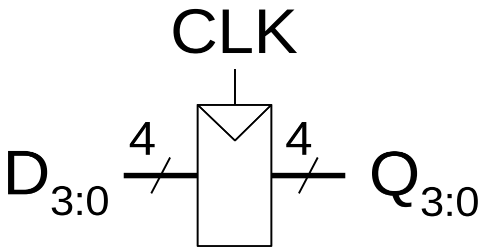
D Flip-Flop Symbols



Storage Elements (Cont…)

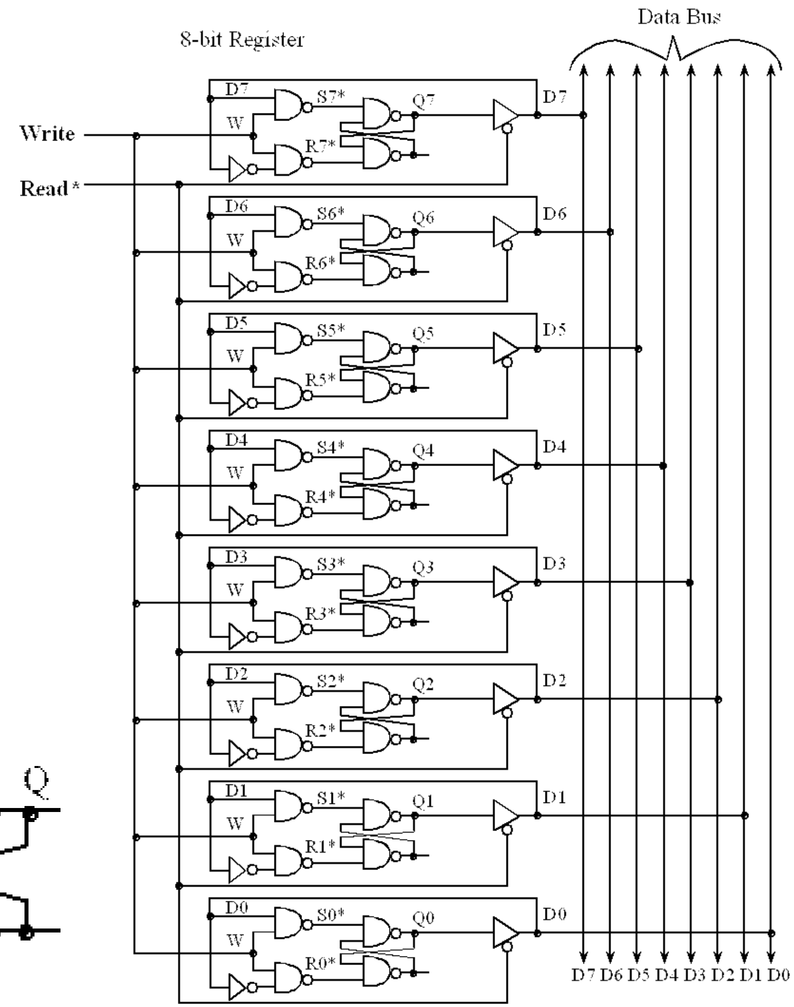
- D flip-flops are the basic building block of RAM and registers on the computer.



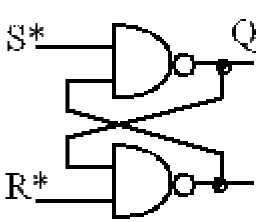


Storage Elements (Cont…)

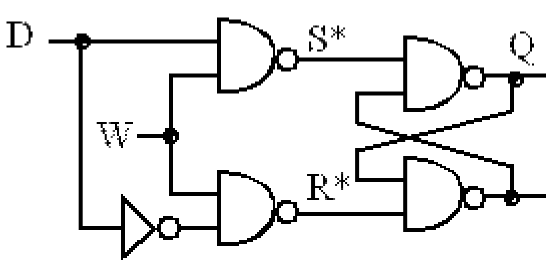
- **D flip-flop** is the basic building block of RAM and registers on the computer.
- This basic storage element is called a **register**, as shown in Figure. (assembly).
- A **bus** is a collection of wires used to pass data from one place to another.



Set-Reset Latch

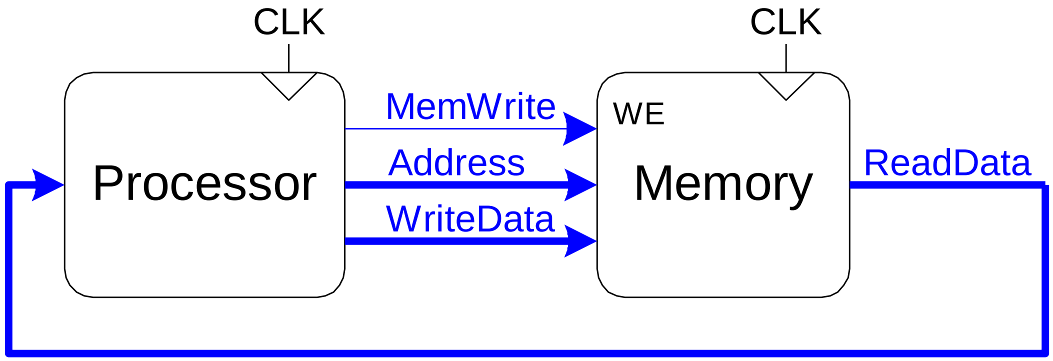


Gated D Latch



Digital Information stored in Memory

- Memory is a collection of hardware elements
- Each memory cell contains one byte of information, and each byte has a unique and sequential address. (byte-addressable)
- The address of a memory cell specifies its physical location.
- When we write to memory, we specify an address and 8, 16, or 32 bits of data, causing that information to be stored into the memory.
- When we read from memory we specify an address, causing 8, 16, or 32 bits of data to be retrieved from the memory.



Digital Information stored in Memory (Cont…)

- Software-Program is an ordered sequence of very specific instructions stored in memory, defining exactly what and when certain tasks are to be performed.
- The computer can store information in RAM by writing to it, or it can retrieve previously stored data by reading from it.
- Most microcontrollers have static RAM (SRAM) using six metal-oxide-semiconductor field-effect transistors to create each memory bit.
- Flash ROM is a popular type of EEPROM. Each flash bit requires only two MOSFET transistors. The input (gate) of one transistor is electrically isolated, charge is trapped on this input. The other transistor is used to read the bit by sensing whether or not the other transistor has trapped charge.
- Because flash is smaller than regular EEPROM, most microcontrollers have a large flash into which we store the software. For all the systems in this class, we will store instructions and constants in flash ROM and place variables and temporary data in static RAM.

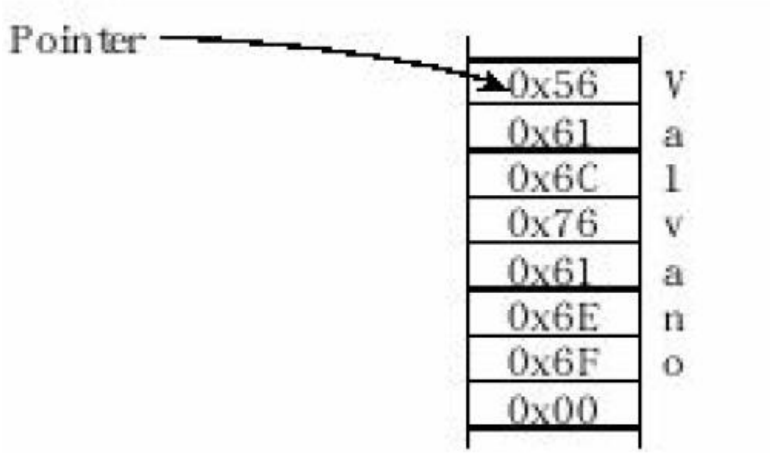
Digital Information stored in Memory (Cont…)

ROM	RAM
The information is programmed or burned into the device, and during normal operation it only allows read accesses.	The information is stored temporary, and during normal operation we can read from or write data into RAM.
nonvolatile, meaning the contents are not lost when power is removed.	volatile, meaning the contents are lost when power is removed.
ROM on the other hand is much denser than RAM.	Most microcontrollers have much more ROM than RAM.
It takes a comparatively long time to program or burn data into a ROM. (1ms)	Writing to RAM is about 100,000 times faster (on the order of 10 ns).

Character information

- American Standard **Code** for Information Interchange (ASCII) code is used to represent a character.
- Standard ASCII is actually only 7 bits, but is stored 8-bit bytes with the most significant bit equal to 0.
- For example, the capital ‘V’ is defined by the 8-bit binary 0101_0110 hexadecimal 0x56.
- In C, the char data type is used to represent characters.

- “Valvano” is encoded as these 8 bytes 0x56, 0x61, 0x6C, 0x76, 0x61, 0x6E, 0x6F, 0x00 (**NULL** character).



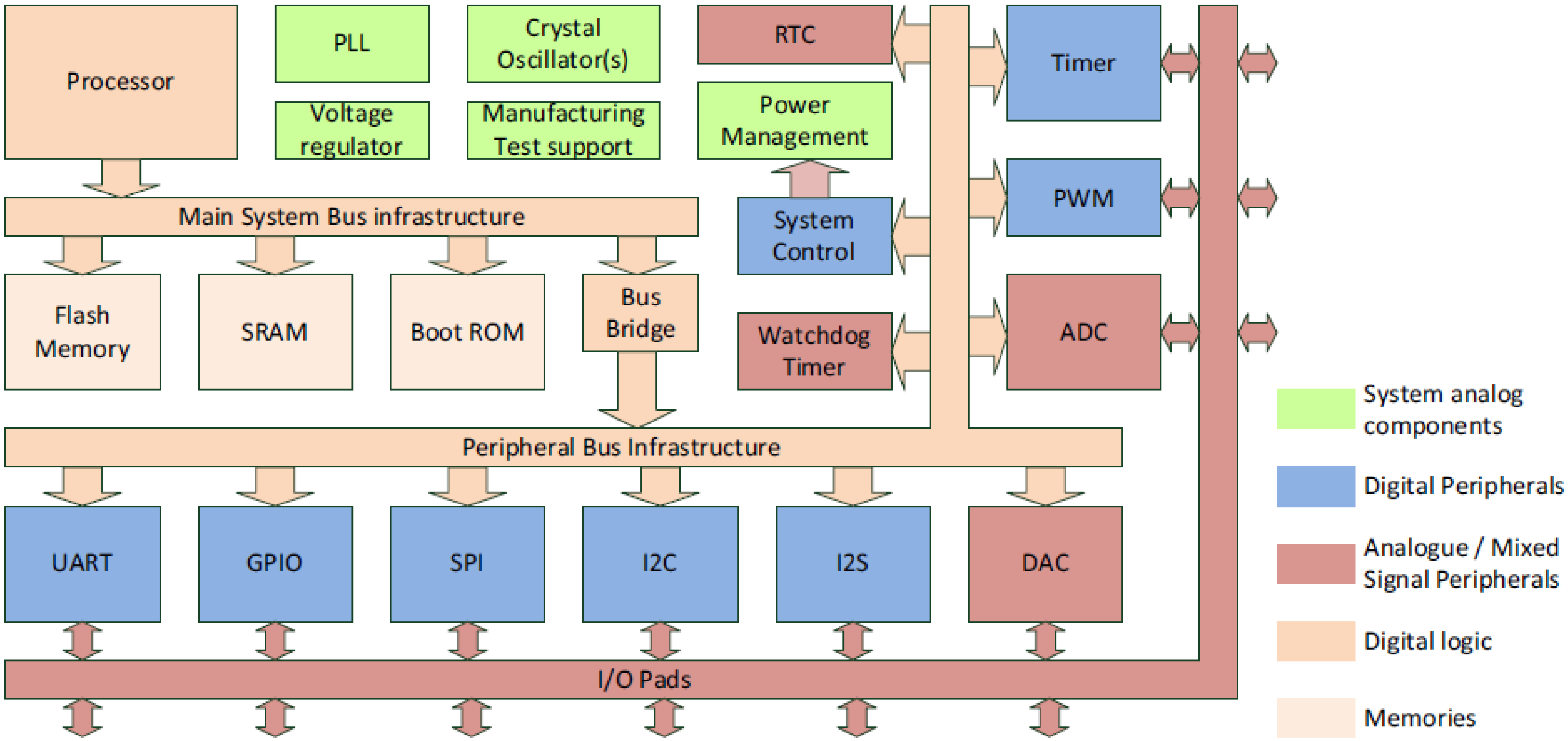
HW- Computer Architecture

- A computer combines a processor, random access memory (RAM), read only memory (ROM), and input/output (I/O) ports.
- Computers are electronic idiots. They can store a lot of data, execute programs quite quickly but they do exactly what we tell them to do. They don’t get bored doing the same tasks over and over again.
- A micro-computer is a small computer, where small refers to size.
- A very small micro-computer, called a microcontroller, contains all the components of a computer (processor, memory, I/O) on a single chip.
- A port is a physical connection between the computer and its outside world. Information enters via the input ports and exits via the output ports.
- A bus is a collection of wires used to pass information between modules.

HW- Computer Architecture (Cont…)

- An interface is defined as the collection of the I/O port, external electronics, physical devices, and the software, which combine to allow the computer to communicate with the external world.
- An example of an input interface is a switch, where the operator toggles the switch, and the software can recognize the switch position. An example of an output interface is a light-emitting diode (LED), where the software can turn the light on and off.
- In general, we can classify I/O interfaces into four categories
  - Parallel- binary data are available simultaneously on a group of lines.
  - Serial- binary data are available one bit at a time on a single line.
  - Analog- data are encoded as an electrical voltage, current, or power.
  - Time- data are encoded as a period, frequency, pulse width, or phase shift.

HW- Computer Architecture (Cont…)



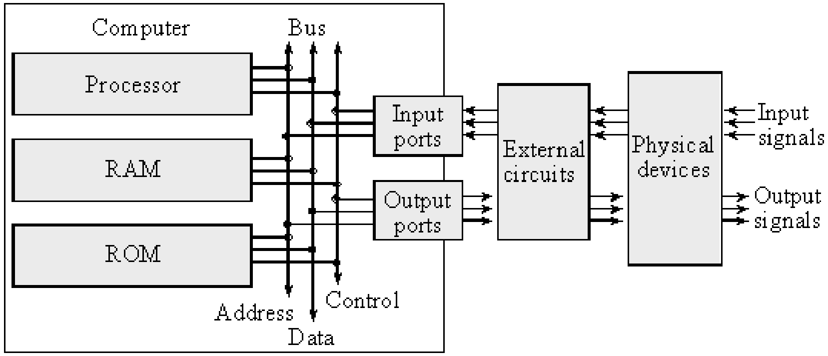
A simple microcontroller.

HW- Computer Architecture (Cont…)

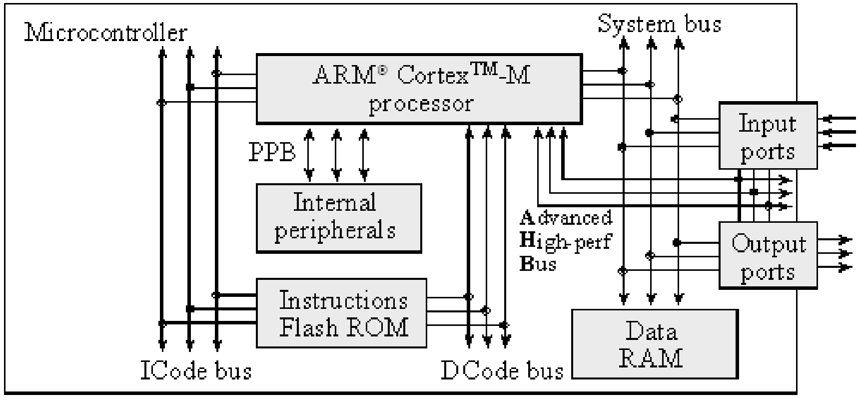
Item	Descriptions
ROM	Read Only Memory—Nonvolatile memory storage for program code.
Flash memory	A special type of ROM, which can be reprogrammed many times, typically for storing program code.
SRAM	Static Random Access Memory—for data storage (volatile)
PLL	Phase Lock Loop—a device to generate programmable clock frequency based on a reference clock.
RTC	Real Time Clock—a low power timer for counting seconds (typically runs on a low power oscillator), and in some cases also for minutes, hours and calendar functions.
GPIO	General Purpose Input/Output—a peripheral with parallel data interface to control external devices and to read back external signals status.
UART	Universal Asynchronous Receiver/Transmitter—a peripheral to handle data transfers in a simple serial data protocol.
I2C	Inter-Integrated Circuit—a peripheral to handle data transfers in a serial data protocol. Unlike UART, a clock signal is required and can provide higher data rate.
SPI	Serial Peripheral Interface—another serial communication interface for off-chip peripherals.
I2S	Inter-IC Sound—a serial data communication interface specifically for audio information.
PWM	Pulse Width Modulator—a peripheral to output waveform with programmable duty cycle.
ADC	Analog to Digital Converter—a peripheral to convert analog signal-level information into digital form.
DAC	Digital to Analog Converter—a peripheral to convert data values into analog signal level.
Watchdog timer	A programmable timer device for ensuring the processor is running program. When enabled, the program running needs to update the watchdog timer within a certain time gap. If the program crashed, the watchdog timed out and this can be used to trigger a reset or a critical interrupt event.

HW- Computer Architecture (Cont…)

- Von Neumann architecture

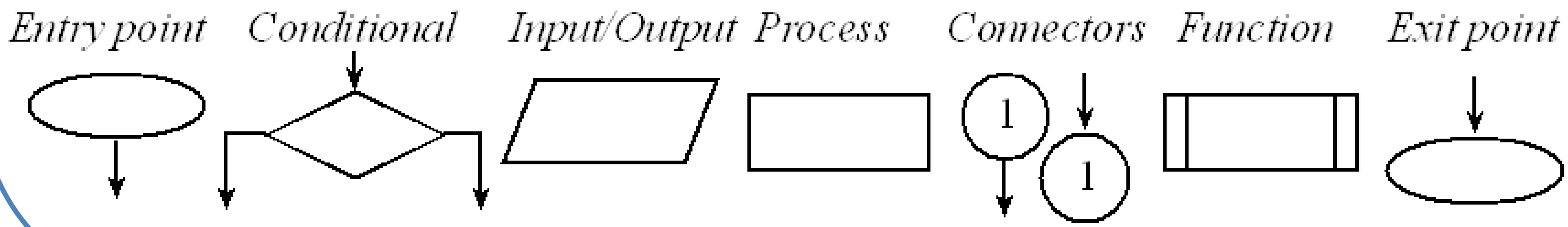


- Harvard architecture



SW- Structured Programming

- graphical tools are used to describe the organization of an embedded system such as: flowcharts, data flow graphs, and call graphs.
- Programs are written in a linear or sequential/one-dimensional fashion.
- Conditional branching and function calls create complex behaviors that are not easily observed in a linear fashion.
- Flowcharts are one way to describe software in a two-dimensional format, to visualize conditional branching and function calls.
- Flowcharts are very useful in the initial design stage and it used in the final documentation stage of a project.

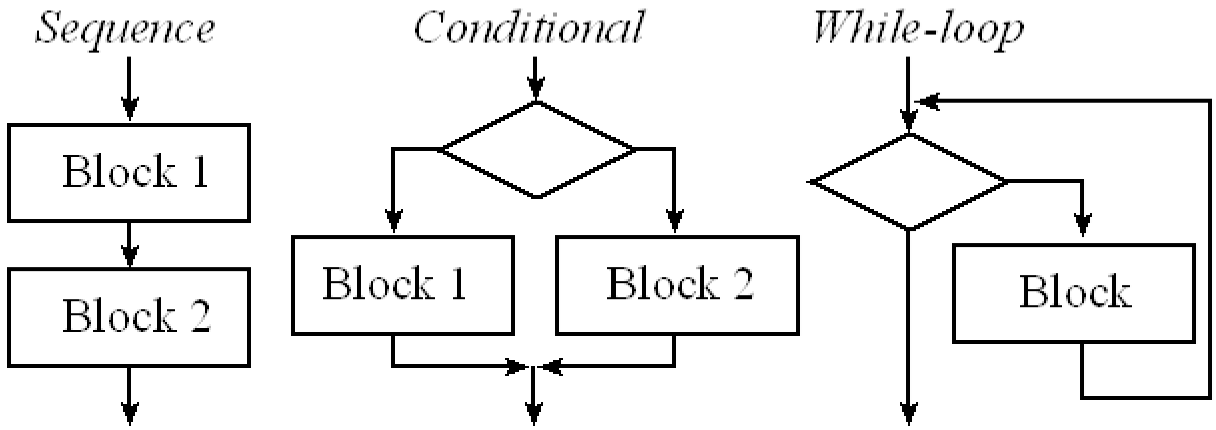


SW- Structured Programming

- The entry point is the starting point of the software/ function, or subroutine. The exit point returns the flow of control back to the place from which the function was called.
- The Rectangles specifies the process block. In a high-level flowchart, a process block might involve many operations.
- The parallelogram defines an input/output operation.
- The diamond-shaped defines a branch point or conditional block. Each arrow out of a condition block must be labeled with the condition causing flow to go in that direction. The condition for each arrow must be mutually exclusive.
- The Rectangle with double lines on the side specifies a call to a predefined function.
- Circles are used as connectors. Connectors with an arrow pointing into the circle are jumps or goto commands.

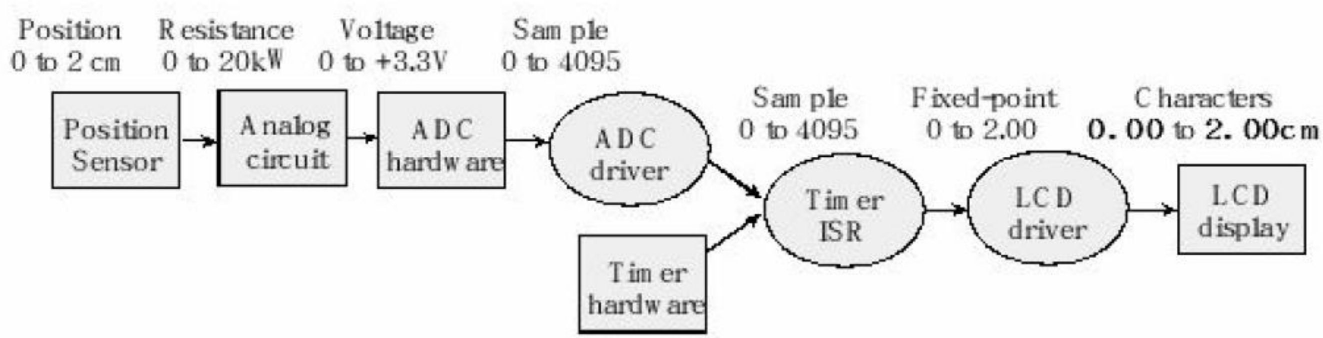
SW- Structured Programming

- Structured programs are built from three basic building blocks: the **sequence**, the **conditional**, and the **while-loop**.
- At the lowest level, the process block contains simple and well-defined commands.

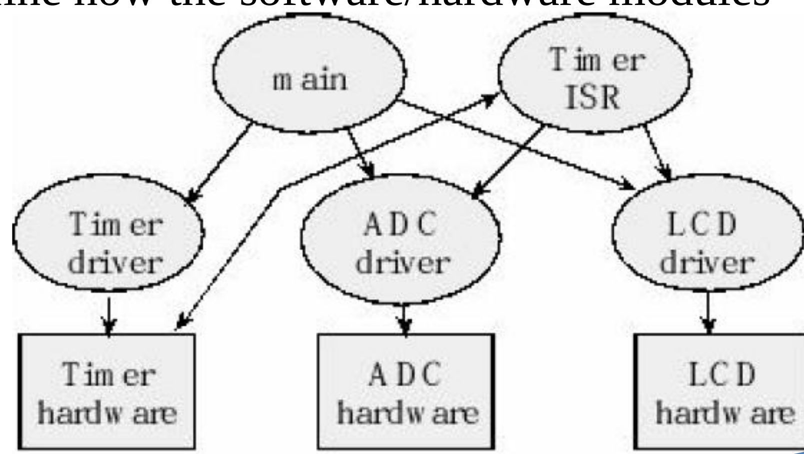


SW- Structured Programming

- Data flow graph is a block diagram of the system, showing the flow of information. Arrows point from source to destination.



- Call graph is a graphical way to define how the software/hardware modules interconnect.



SW- Parallel Programming

- Parallel programming to execute multiple threads at the same time. A computer with a multi-core processor can simultaneously execute a separate program in each of its cores.
- Fork and join are the fundamental building blocks of parallel programming.
- After a fork, two or more software threads run in parallel/ simultaneously on separate processors. Two or more simultaneous software threads can be combined into one using a join.
- Concurrent programming allows the computer to execute multiple threads, but only one at a time.
- Interrupts are one mechanism to implement concurrency on real-time systems. The foreground thread is defined as the execution of the main program, and the background threads are executions of the Interrupt.

CHW 469 : Embedded Systems

- Intro to Embedded System.
- Review of Electronics Part (KCL, KVL, Parallel/Series Resistance, DAC and ADC) .
- Review for Digital Systems (Binary Number, Logic, MOS Implantation, Computer Architecture).
- Intro to Programming Concepts (Structure and Concurrent).