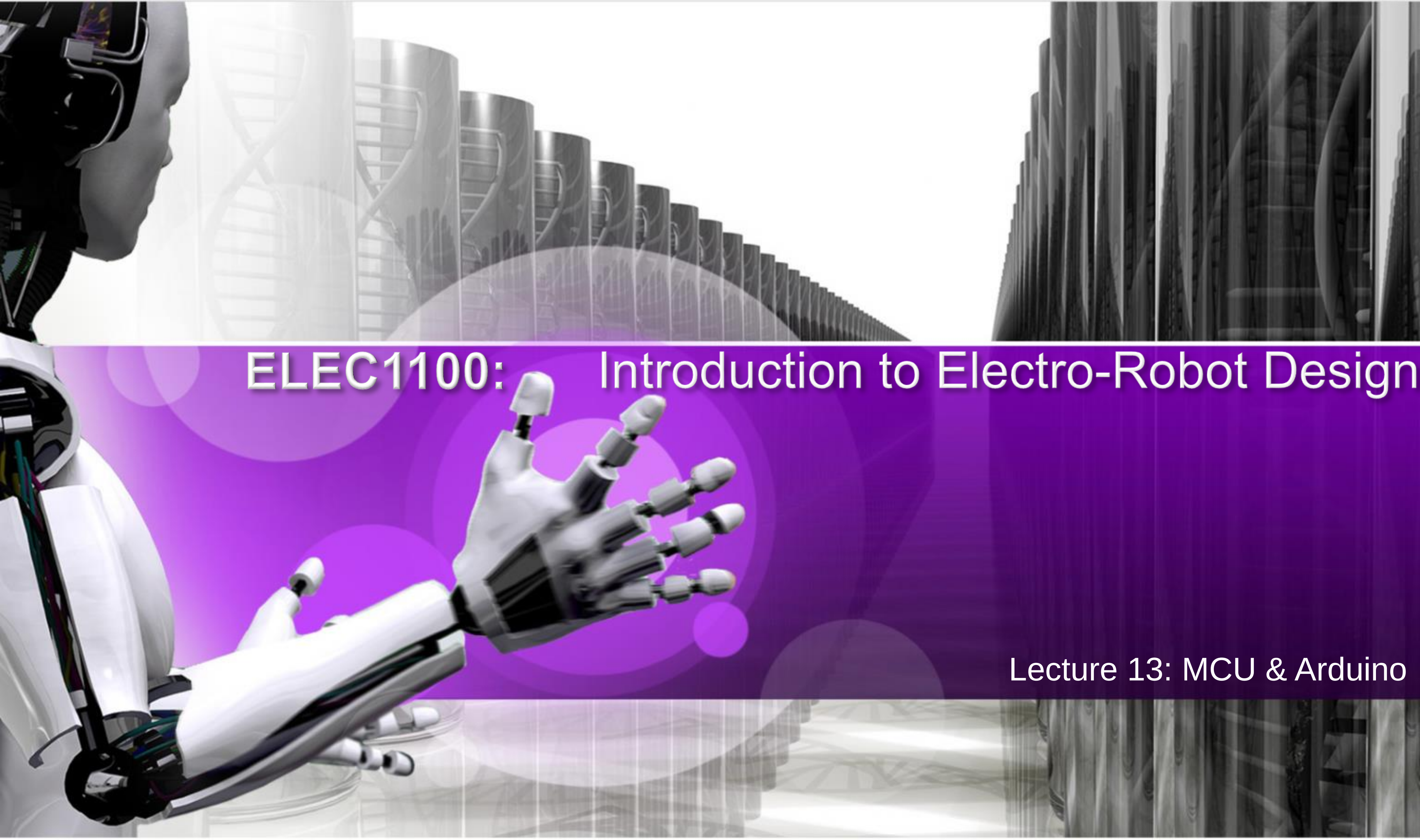
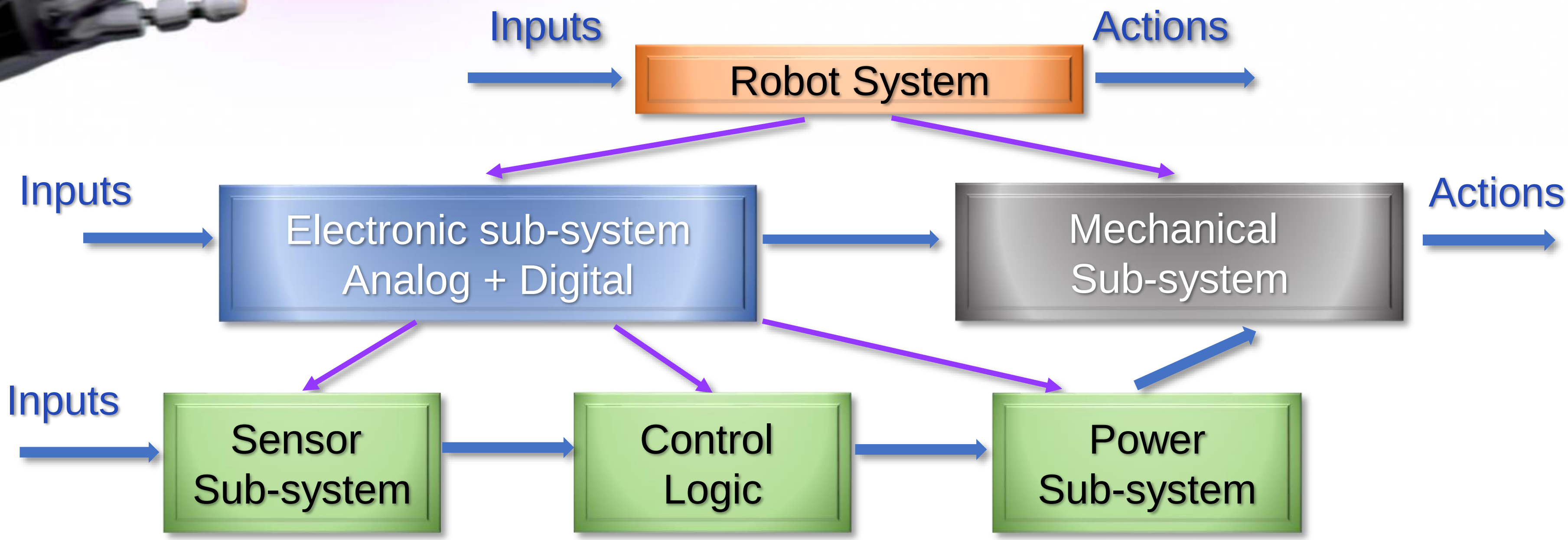


ELEC1100: Introduction to Electro-Robot Design

Lecture 13: MCU & Arduino

SONG Shenghui and MURCH Ross, Dept. of ECE, HKUST

ELEC1100 ROADMAP



Sensor Basics:

Wk6: Sensor Basic –
Sensor/Line/ADC

Combinational/Sequential Logic:

Wk7: Robot Brain: Logic Gate and
Logic Operation
Wk8: MCU & Arduino

Basic electronics:

Wk1: Basic Electronics -
Charge/Current/Voltage/Resistor
Wk2: Energy/Power and DC Sources

Motor Power Supply:

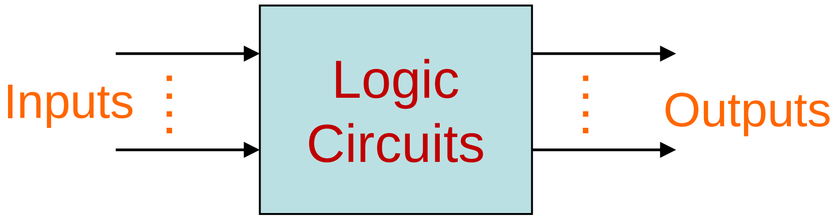
Wk3: Pulse Signal and PWM Control
Wk4: Transistor and H-Bridge

FROM LAST LECTURE

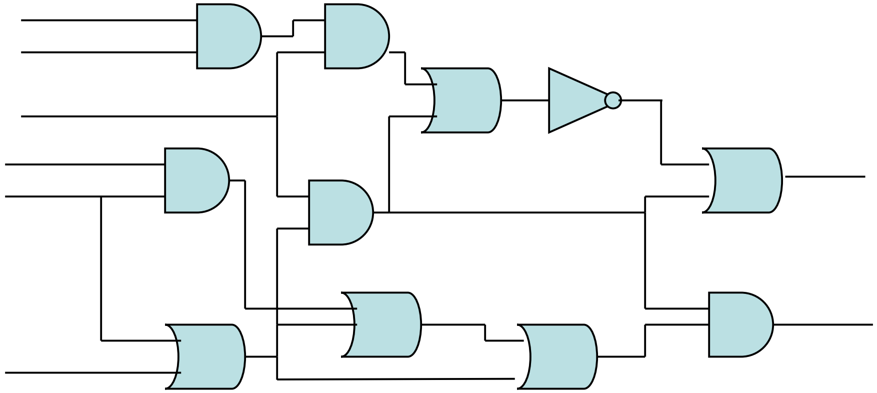
- ❖ Combinational circuit

➢ The outputs depend only on the current inputs of the circuits

➢ Output values are expressed by the truth table of the inputs



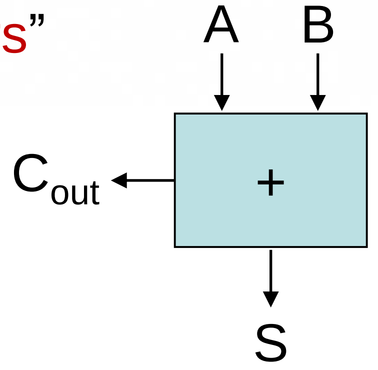
➢ Logic level of combinational circuit: maximum number of gates of all paths from inputs to outputs of the circuit



Logic level = 5

EXAMPLE: HALF ADDER

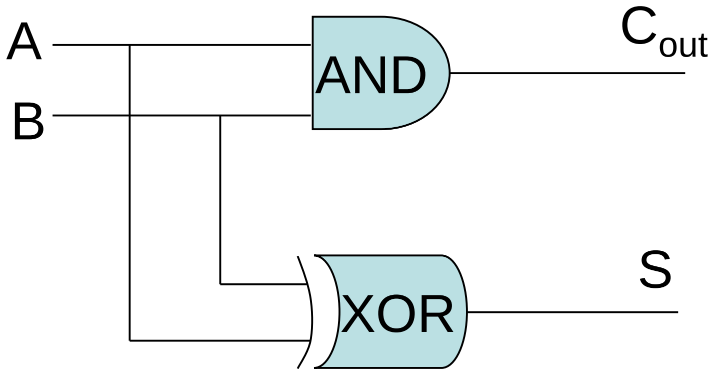
- The half adder is a circuit that adds two “1-bit numbers” and result of the addition is a “2-bit number”
  - 2 inputs – *A* and *B*
  - 2 outputs – Sum (S) and Carry (C_{out})



$$A+B=2^{1}\times C_{out}+2^{0}S$$
$$=2\times C_{out}+S$$

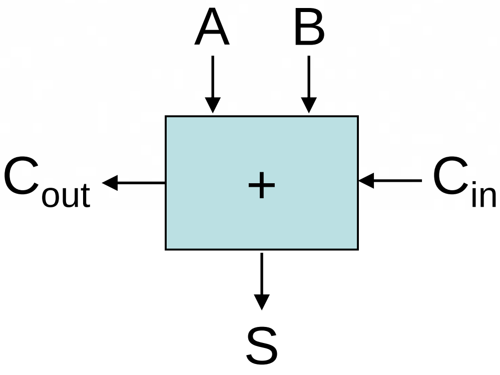
- Truth table

A	B	S	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



ONE-BIT FULL ADDER

- Full adder
  - 3 inputs – *A*, *B* and C_{in}
  - 2 outputs – Sum (S) and Carry (C_{out})



$$\begin{aligned} A+B+C_{in} &= 2^1\times C_{out}+2^0 S \\ &= 2\times C_{out}+S \end{aligned}$$

- Truth table

A	B	C_{in}	S	C_{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

- K-map

S	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}_{in}$	0	1	0	1
C_{in}	1	0	1	0

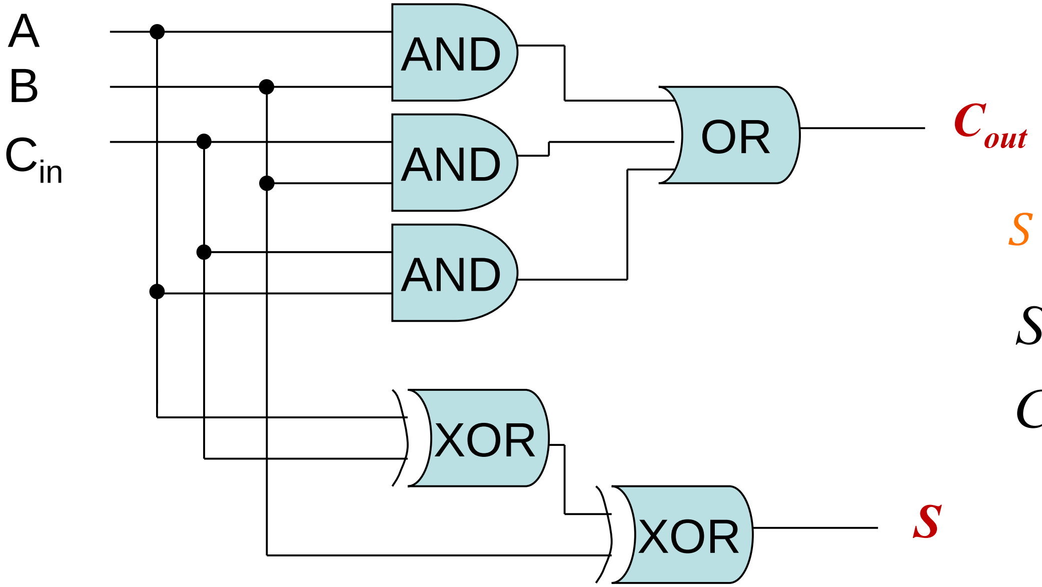
C_{out}	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}_{in}$	0	0	1	0
C_{in}	0	1	1	1

ONE-BIT FULL ADDER IMPLEMENTATION

S	$\overline{A}\,\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}_{in}$	0	1	0	1
C_{in}	1	0	1	0

C_{out}	$\overline{A}\,\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}_{in}$	0	0	1	0
C_{in}	0	1	1	1

- Logic gate implementation



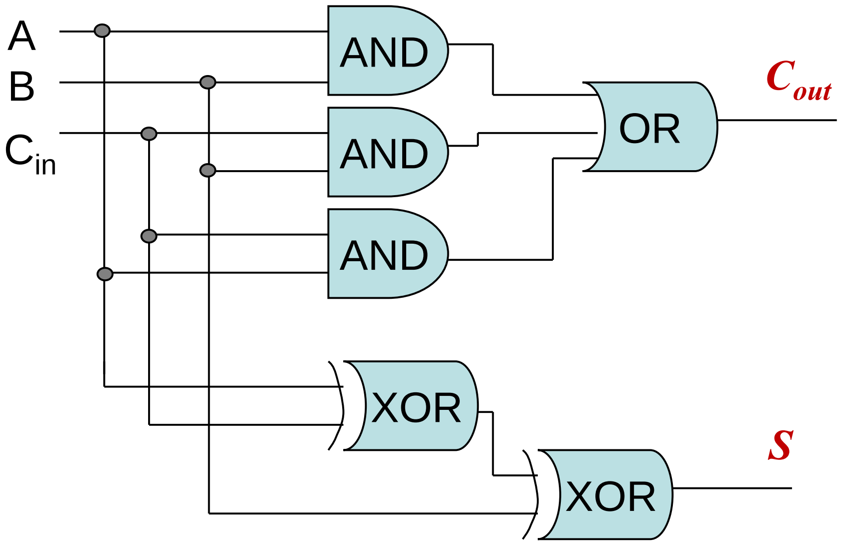
$$S = C_{in}\overline{A}\,\overline{B} + \overline{C_{in}}\,\overline{A}B + C_{in}AB + \overline{C_{in}}A\overline{B}$$

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = A \cdot B + A \cdot C_{in} + C_{in} \cdot B$$

COMBINATIONAL LOGIC

- Output of the logic only depends on the current inputs but not on the history of the inputs
- As soon as the inputs change their values, the output (in general) will be affected

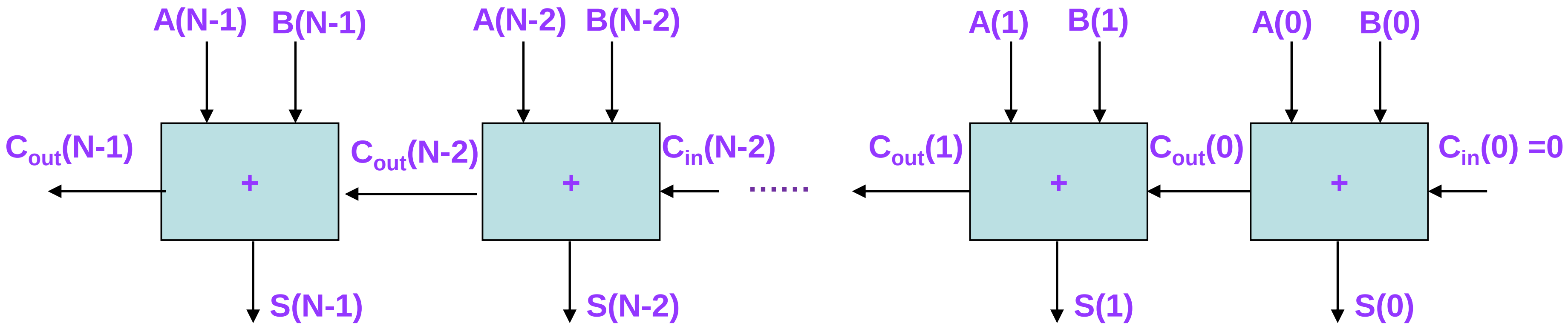


Take a full adder as an example

- if A = 1, B = 1, C_{in} = 1, then C_{out} = 1 and S = 1
- if A is changed to 0, then S is changed to 0

FROM 1-BIT ADDER TO N-BIT ADDER

- How to form N-bit adder ?



- Suppose we want to add m n-bit numbers (each X_i of the following equation is a N-bit number)

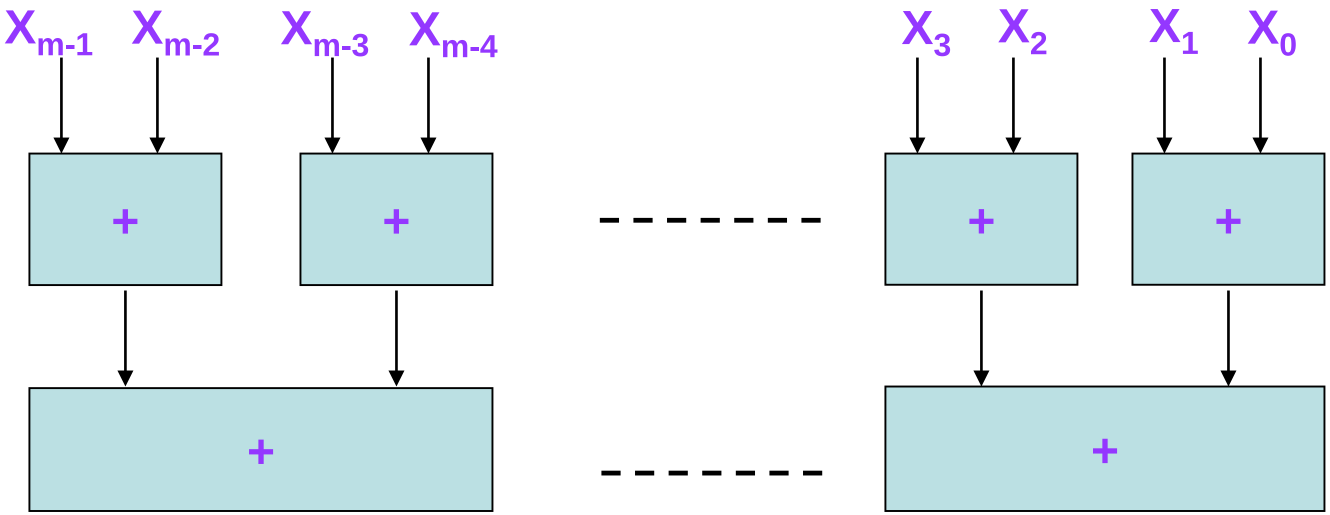
$$\text{Output} = X_{m-1} + X_{m-2} + \ldots + X_1 + X_0$$

- How would you find the output?

COMBINATIONAL LOGIC APPROACH

- Example: using divide and conquer

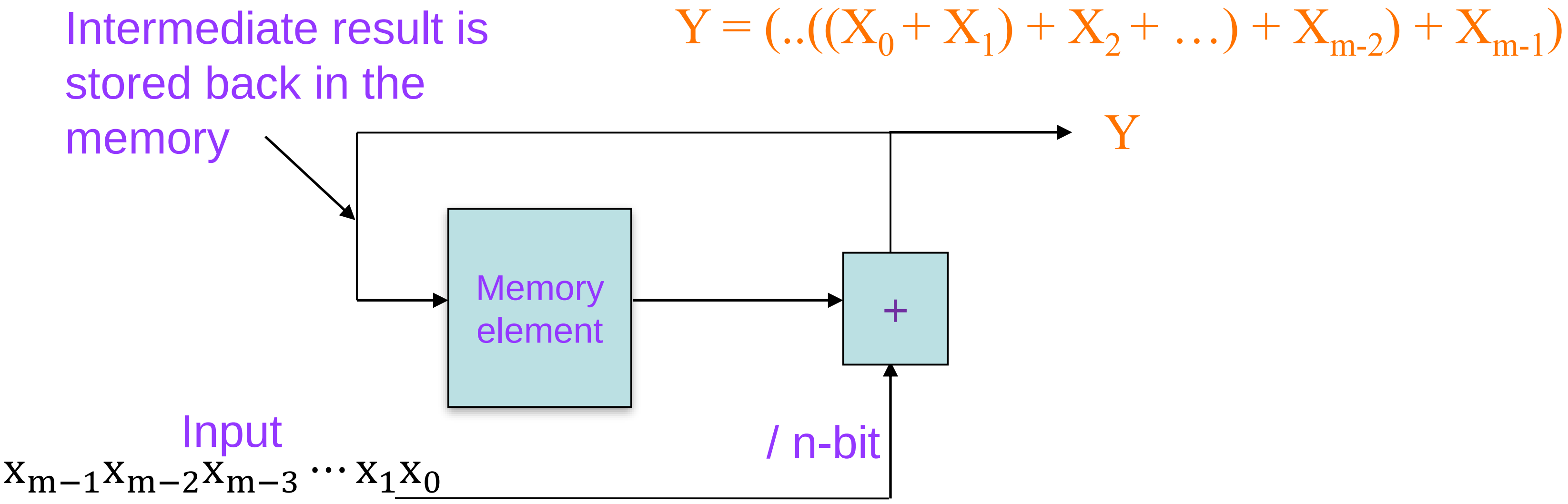
$$\text{Output} = (X_{m-1} + X_{m-2}) + \ldots + (X_1 + X_0)$$



- How many adders do you need?

USING SEQUENTIAL LOGIC

- We can also use sequential logic to feedback the output of the intermediate sum to the input of the adder



Intermediate result is stored back in the memory

$$Y = (..((X_0 + X_1) + X_2 + \ldots) + X_{m\text{-}2}) + X_{m\text{-}1})$$

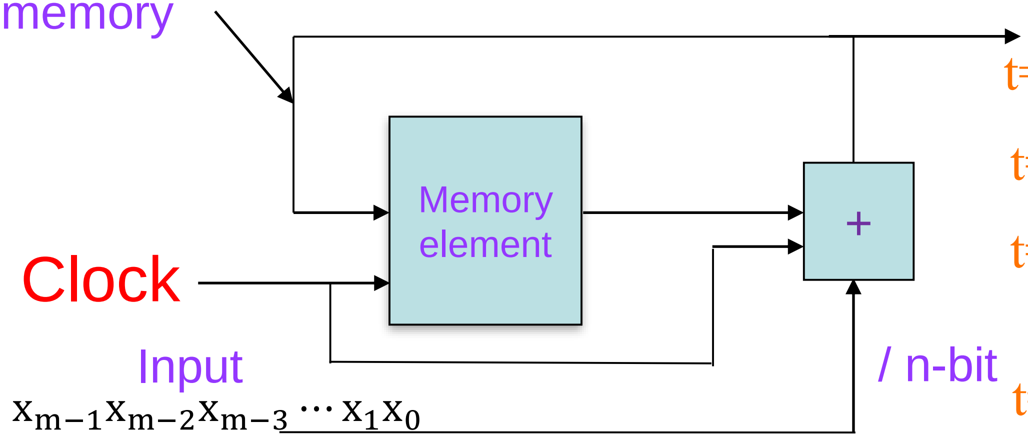
- What is missing in the scheme?
- When will we know it is ready to send the data in?

SEQUENTIAL LOGIC CIRCUIT

- We need a clock signal to synchronize the input and the memory output

Intermediate result is stored back in the memory

$Y(t)=Y(t-1)+X(t);\ Y(-1)=0$



$t=0$: $Y(0)=Y(-1)+X(0)=X(0)$

$t=1$: $Y(1)=Y(0)+X(1)=X(1)+X(0)$

$t=2$: $Y(2)=Y(1)+X(2)=X(2)+X(1)+X(0)$

⋮

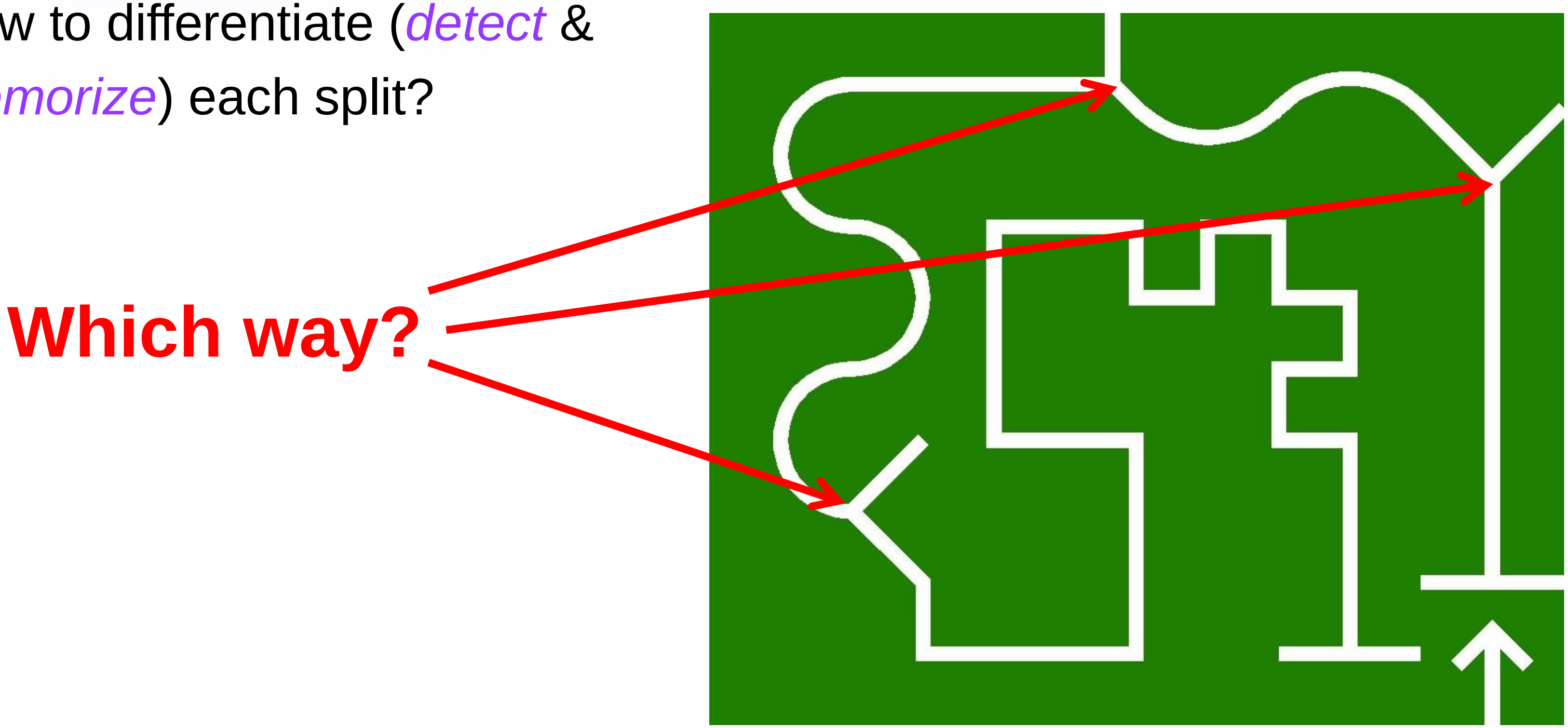
$t=m-1$: $Y(m-1)=Y(m-2)+X(m)$

$=X(m-1)+\ldots+X(1)+X(0)$

- What is the drawback compared with combinational logic?
- A key element, **memory**, is needed

SEQUENTIAL LOGIC IN YOUR PROJECT

- How to differentiate (*detect* & *memorize*) each split?

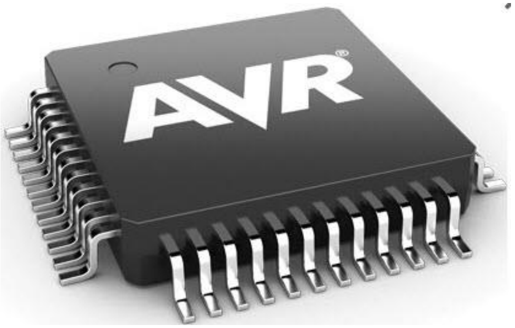


Which way?

- You will need to use a micro-controller for involving the ***clock*** signal and ***memory*** unit into the logic control system for your vehicle.

MICRO-CONTROLLER

- A micro-controller (MCU: micro-controller unit) is a small computer on a single integrated circuit.
  - It is a full computer on one chip but with resources far from a limited desktop personal computer.
  - The single chip contain CPU, NON-volatile memory (ROM), volatile memory (RAM), Timer & I/O port.
  - There is no micro-controller works alone in circuit, it must interface with other external device like sensors, motors……etc.



Atmel AVR Microcontroller

ARDUINO FAMILY

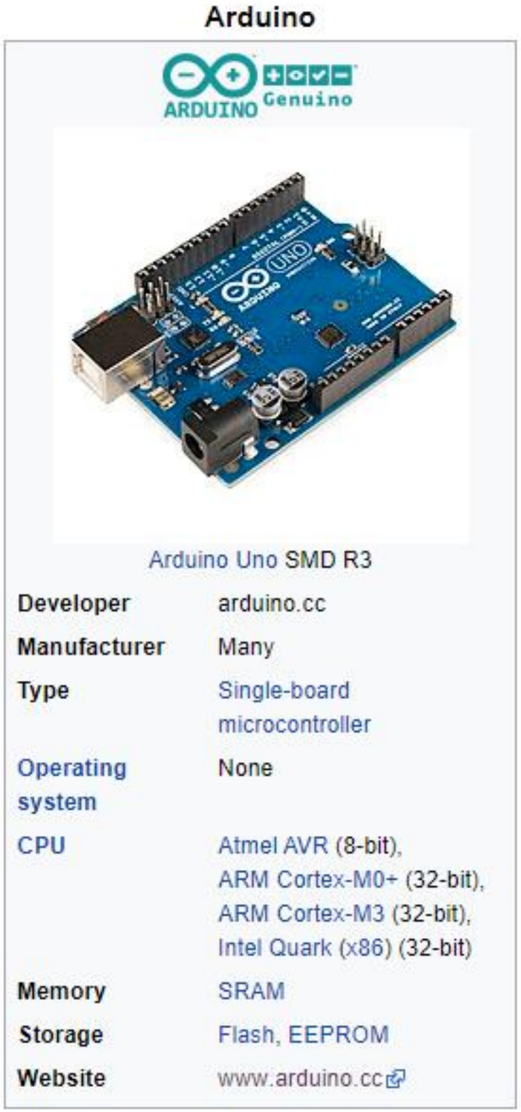
- **Arduino** is an open-source electronics platform that designs and manufactures single-board micro-controllers and micro-controller kits for building digital devices based on easy-to-use hardware and software.

- Hardware: Arduino Boards

  You can tell your board what to do by sending a set of instructions to the micro-controller on the board.

- Software: Arduino Integrated Development Environment (IDE)

  You can use Arduino programming language to write instructions and upload them to the Arduino board.



Arduino

Arduino Uno SMD R3

Developer	arduino.cc
Manufacturer	Many
Type	Single-board microcontroller
Operating system	None
CPU	Atmel AVR (8-bit), ARM Cortex-M0+ (32-bit), ARM Cortex-M3 (32-bit), Intel Quark (x86) (32-bit)
Memory	SRAM
Storage	Flash, EEPROM
Website	www.arduino.cc

SOME HISTORY

- The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in *Ivrea, Italy*, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators.

- The name **Arduino** comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after *Arduin of Ivrea*, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.

Arduin of Ivrea

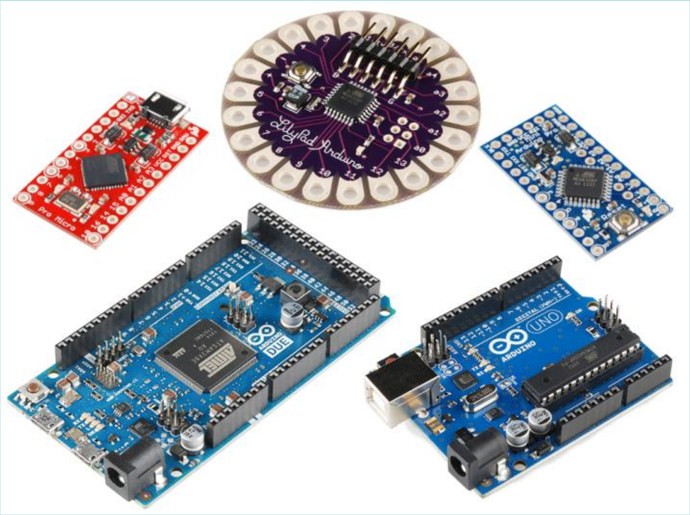
King of Italy

ARDUINO BOARDS

- Arduino board designs use a variety of micro-processors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) **pins** that may be interfaced to various expansion boards or breadboards and other circuits.

- The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers.

Different Types of Arduino Boards



ARDUINO SOFTWARE IDE

- The Arduino Software IDE is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java, also supports the languages C and C++.
- It includes a code editor, and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board.

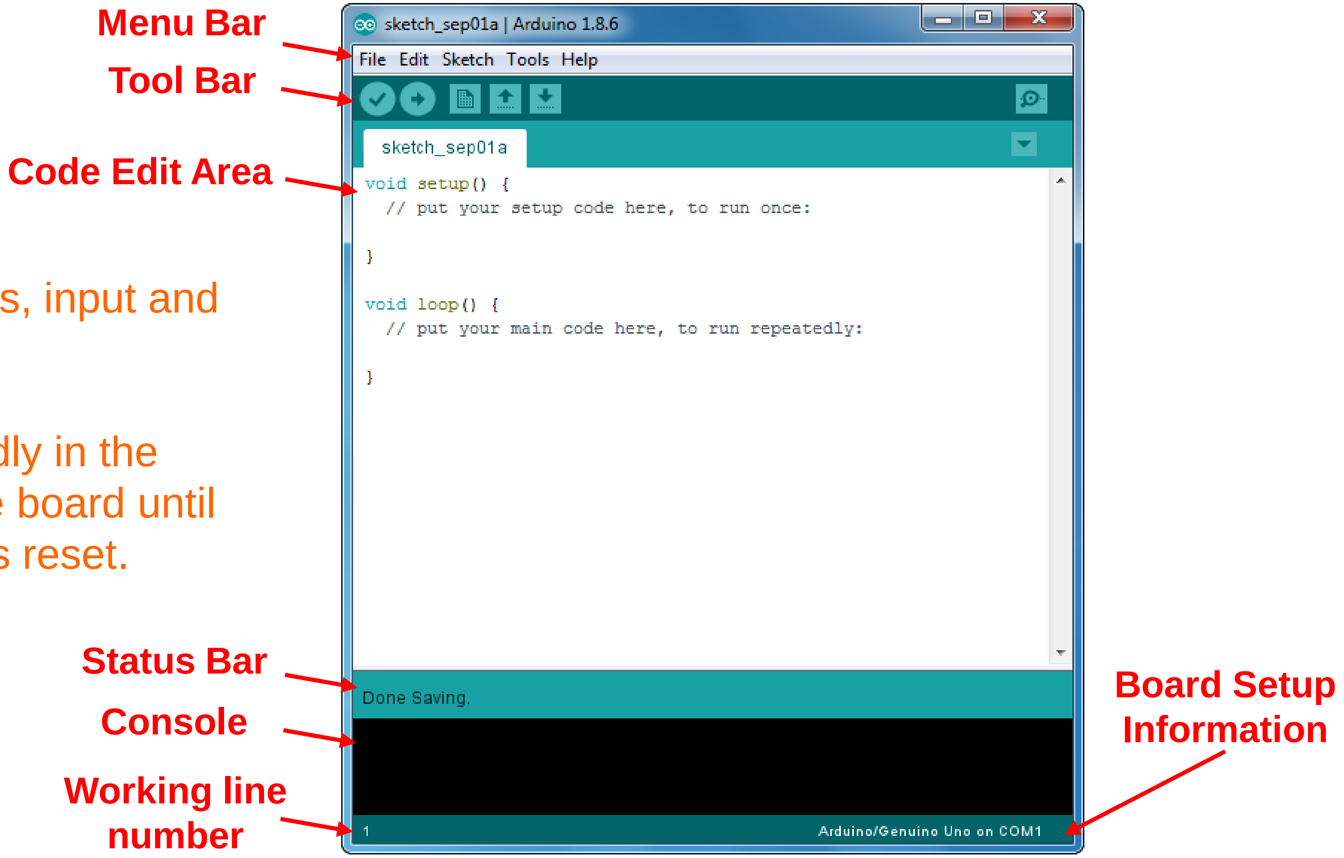
The Arduino software IDE on the desktop of your lab computer.

ARDUINO SKETCH

❖ A sketch is a program written with the Arduino IDE.

A minimal Arduino C/C++ program consists of only two functions:

- **setup( )**: to initialize variables, input and output pin modes.
- **loop( )**: is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.



Menu Bar

Tool Bar

Code Edit Area

Status Bar

Console

Working line number

Board Setup Information

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

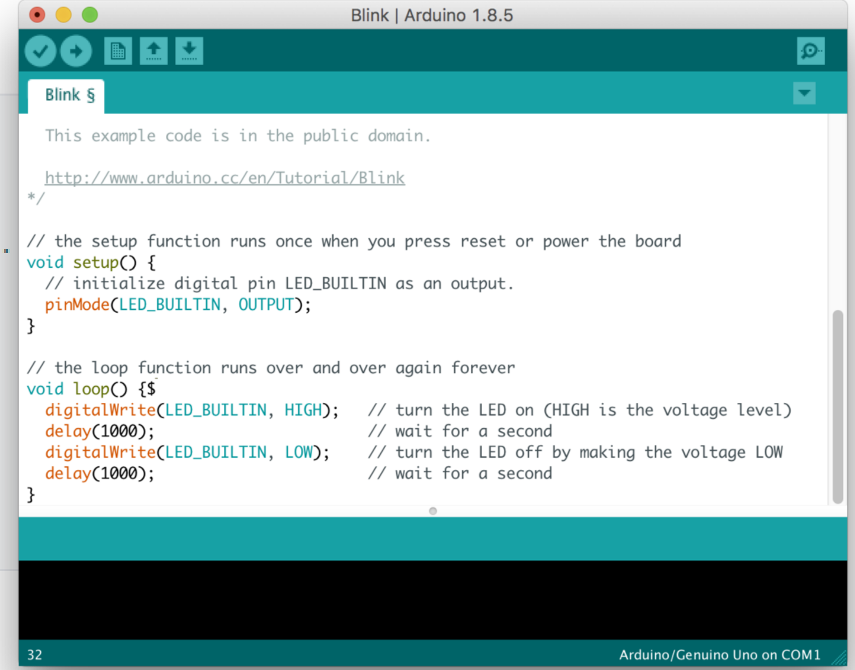
Done Saving.

```
1                                        Arduino/Genuino Uno on COM1
```

BLINK EXAMPLE

- Most Arduino Boards contain a LED and a current limiting resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions.

```
#define LED_PIN 13                  // Pin number attached to LED.

void setup() {
    pinMode(LED_PIN, OUTPUT);       // Configure pin 13 to be a digital output.
}

void loop() {
    digitalWrite(LED_PIN, HIGH);    // Turn on the LED.
    delay(1000);                    // Wait 1 second (1000 milliseconds).
    digitalWrite(LED_PIN, LOW);     // Turn off the LED.
    delay(1000);                    // Wait 1 second.
}
```

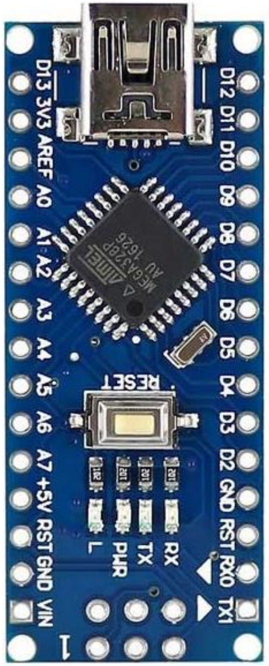


```
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

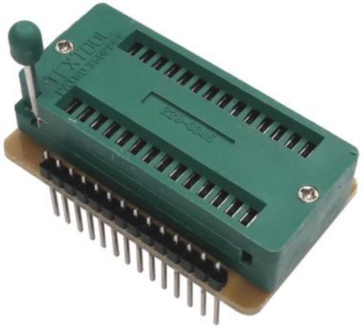
Screenshot of Arduino IDE showing *Blink* program

YOUR LAB TOOLS

❖ At physical lab:

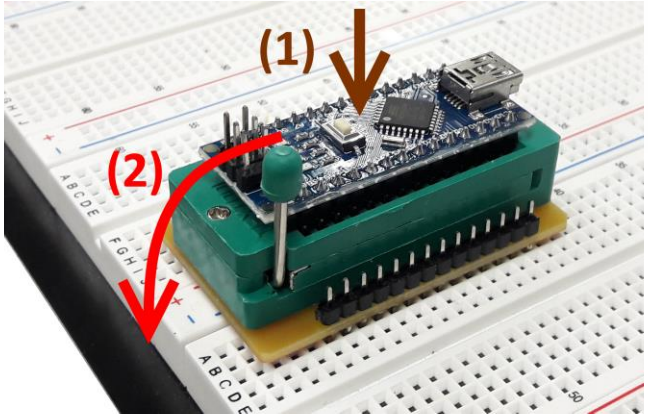

D13 3V3 AREF A0 A1 A2 A3 A4 A5 A6 A7 +5V RST GND VIN
D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
RESET
L PWR TX RX

Arduino compatible Nano-Board



IC Socket

Mini-B to Type-A USB cable



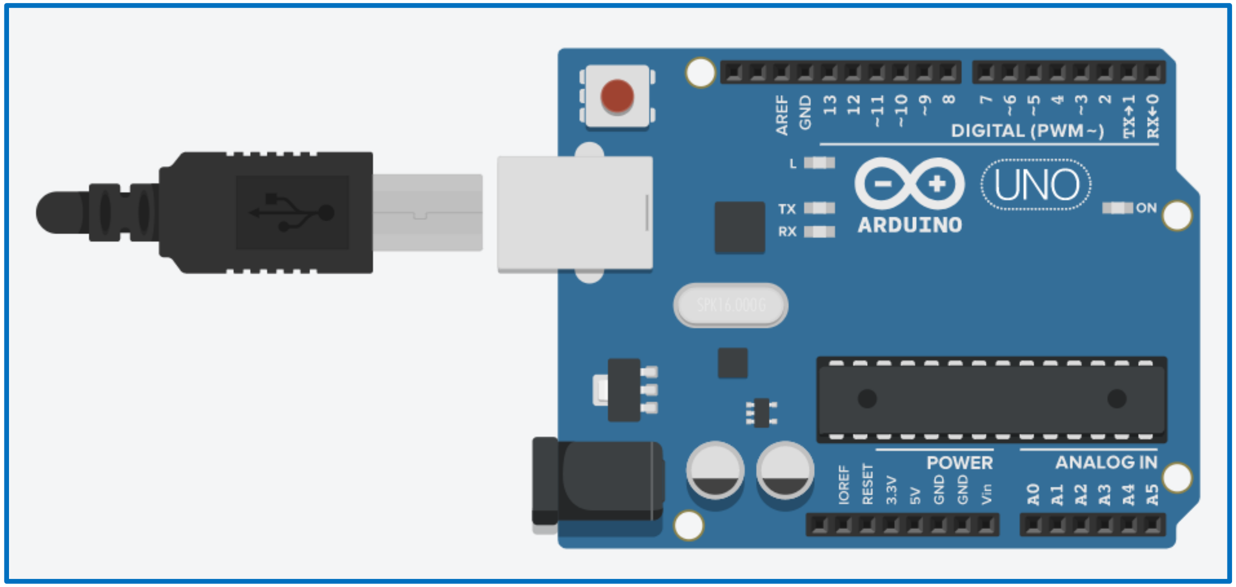
Pull down the handle to lock your Nano-board

Connect Nano-Board to computer’s USB port

ON-LINE SIMULATION

- In Tinkercad:

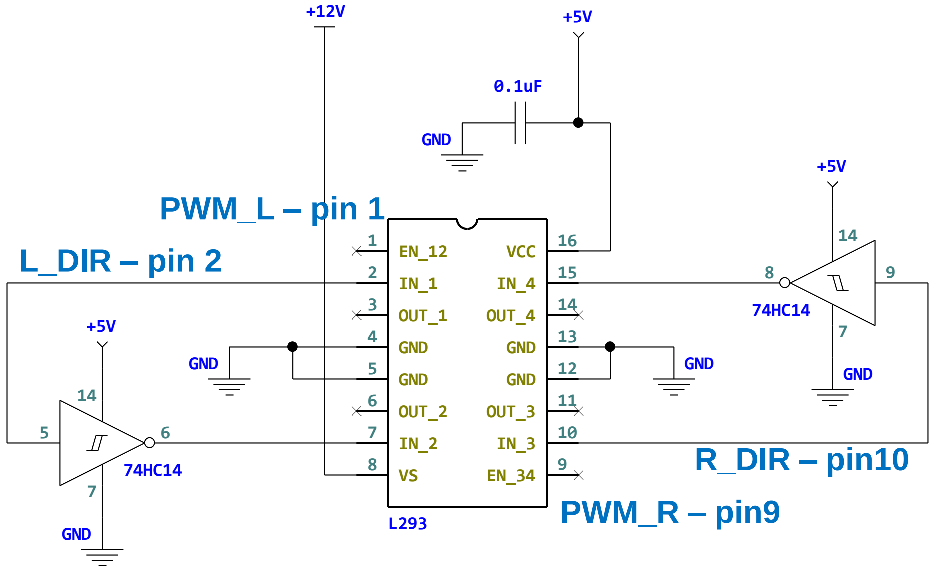
Arduino Uno-Board



- Arduino Uno-board fairly similar to Arduino Nano-board as both are built with a microcontroller Atmega328.
- Nano-board is preferred at physical lab because of small in size.

INPUT/OUTPUT SIGNALS

- ❖ Your lab#05:

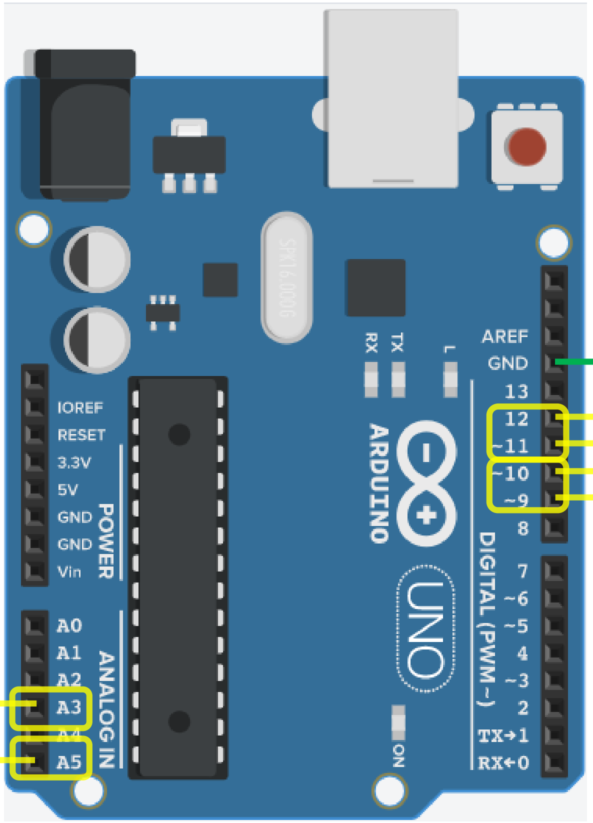


PWM_L – pin 1

L_DIR – pin 2

R_DIR – pin10

PWM_R – pin9



GND = 0V (connect to your breadboard GND)

D12 – R_DIR
D11 – L_DIR
D10 – PWM_L
D9 – PWM_R

OUTPUTs

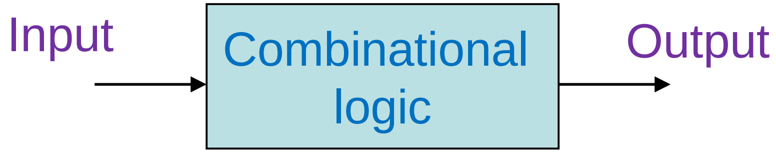
INPUTs

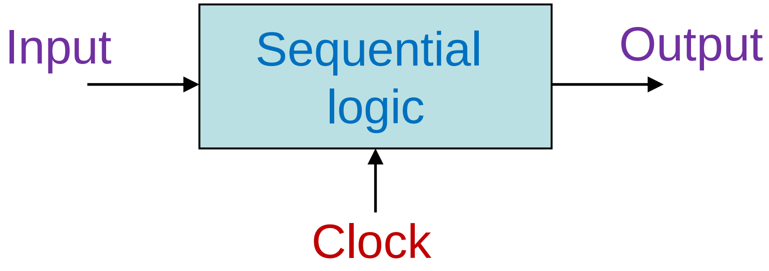
Sensor R – A3
Sensor L – A5

Connect your Uno-Board to the circuits from previous labs on your breadboard.

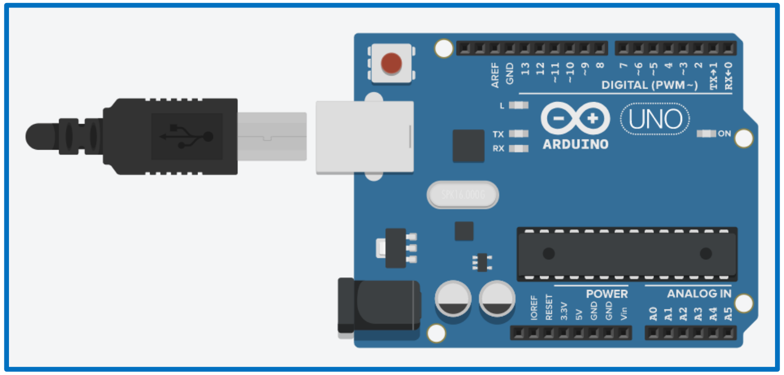
LECTURE SUMMARY

- Combinational logic: Output changes as soon as inputs change
- Sequential logic: Output may change only at a specific time, depending on the clock signal.





- Arduino: Your logic & memory unit.



NEXT LECTURE

- Arduino Code:
  - Introduction
  - Functions
  - Variables

QUESTIONS?