

# SUMADOR ARDUINO



Trabajo realizado por: Amanda Muñoz y Pablo Márquez.

Curso: 2ºBACH C

## Índice:

1. Código
2. Esquema
3. Explicación
4. Diferencias entre arduino y FPGA

## Código:

### Puerta AND:

```
/*
```

```
Boole
```

```
Función AND con 2 variables
```

```
*/
```

```
int var1 = 7; //Pin de entrada del pulsador 1
```

```
int var2 = 2; //Pin de entrada del pulsador 2
```

```
int led = 5; //Pin de salida para el led(rojo)
```

```
int estado1 = 0; //Para almacenar el estado de la variable1
```

```
int estado2 = 0; //Para almacenar el estado de la variable2
```

```
int resultado = 0; //Para almacenar el resultado
```

```
void setup() {
```

```
pinMode(var1, INPUT); //Iniciliza el pin de entrada 1 como salida
```

```
pinMode(var2, INPUT); //Iniciliza el pin de entrada 2 como salida
```

```
pinMode(led, OUTPUT); //Iniciliza el pin del led como salida
```

```
}
```

```
void loop(){
```

```

estado1 = digitalRead(var1); //Lee el estado del botón y lo almacena
estado2 = digitalRead(var2); //Lee el estado del botón y lo almacena
resultado = (estado1 && estado2); //Función AND con los dos estados
digitalWrite(led, resultado); //Escribimos el resultado en el led
}

```

#### Puertas AND y XOR juntas:

```
/*
```

Boole

Función AND con 2 variables

```
*/
```

```

int var1 = 7; //Pin de entrada del pulsador 1
int var2 = 2; //Pin de entrada del pulsador 2
int led1 = 5; //Pin de salida para el led(rojo)
int led2 = 13; //Pin de salida para el led(rojo)
int estado1 = 0; //Para almacenar el estado de la variable1
int estado2 = 0; //Para almacenar el estado de la variable2
int resultado1 = 0; //Para almacenar el resultado
int resultado2 = 0; //Para almacenar el resultado

```

```

void setup() {
    pinMode(var1, INPUT); //Iniciliza el pin de entrada 1 como salida
    pinMode(var2, INPUT); //Iniciliza el pin de entrada 2 como salida
    pinMode(led1, OUTPUT); //Iniciliza el pin del led como salida
    pinMode(led2, OUTPUT); //Iniciliza el pin del led como salida
}

```

```

void loop(){

    estado1 = !digitalRead(var1); //Lee el estado del botón y lo almacena
    estado2 = !digitalRead(var2); //Lee el estado del botón y lo almacena
    resultado1 = (estado1 && estado2); //Función AND con los dos estados
    resultado2 = (estado1 ^ estado2); //Función AND con los dos estados
    digitalWrite(led1, resultado1); //Escribimos el resultado en el led
    digitalWrite(led2, resultado2); //Escribimos el resultado en el led
}

```

## Esquema:

Objetivo: hacer un sumatorio.

Circuito:

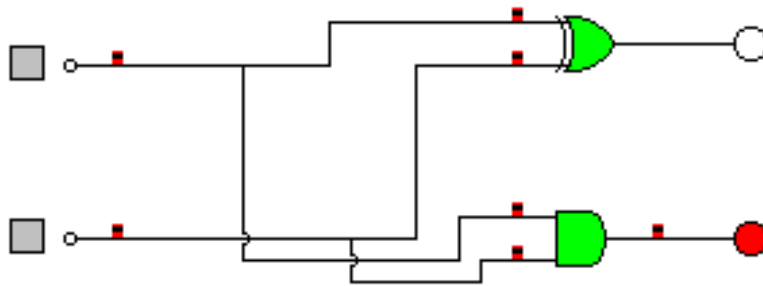


Tabla de verdad:

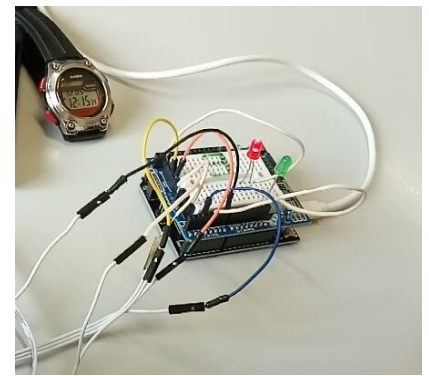
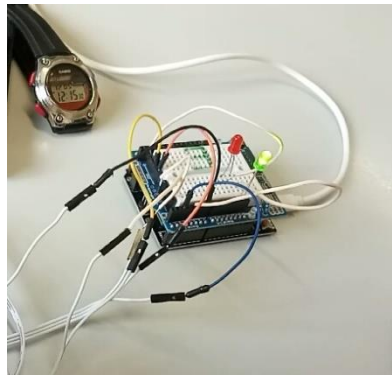
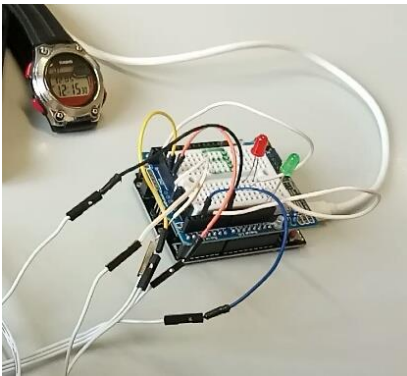
A	B	S1	S2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

## Explicación:

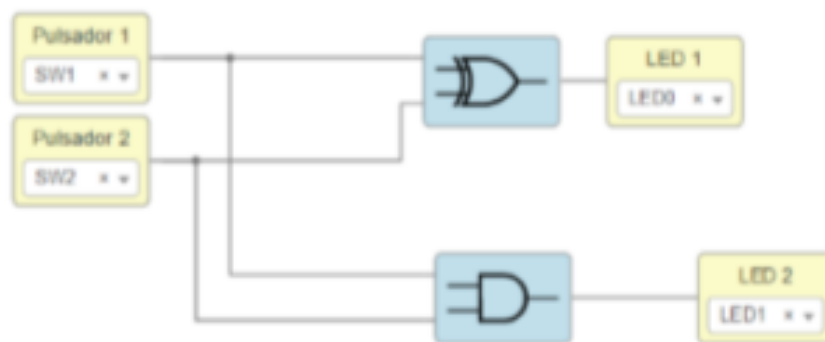
### Arduino:

Este circuito es un sumador, funciona de la siguiente manera: primero cumpliendo con la función de la puerta lógica AND, al dar a uno de los pulsadores no se enciende ningún LED, sin embargo, al pulsar ambos, uno de ellos sí que se enciende.

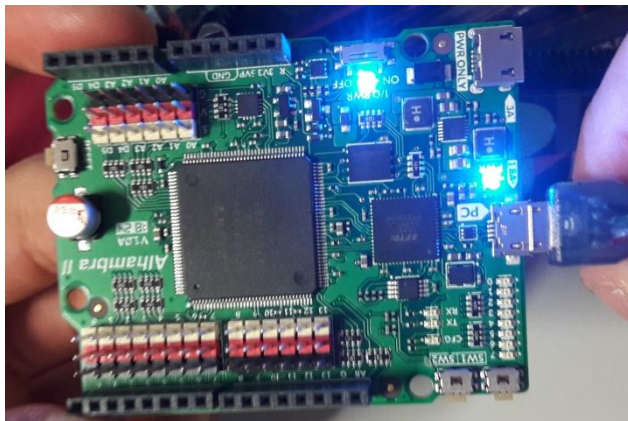
Al añadir la función de la puerta lógica XOR el circuito cambia por completo, ahora al dar solo a uno de los pulsadores uno de los dos LEDs siempre se enciende, pero, al dar a los dos pulsadores se enciende el otro LED



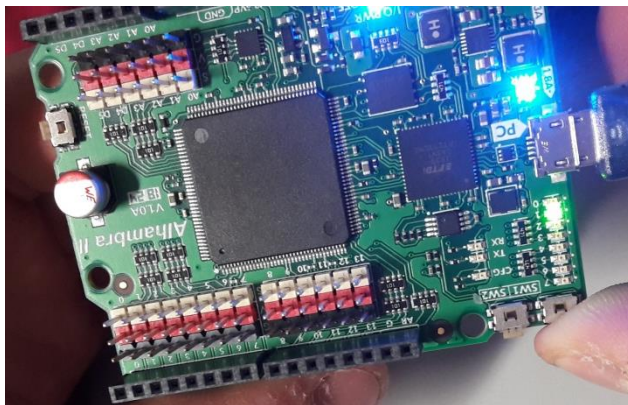
### FPGA:



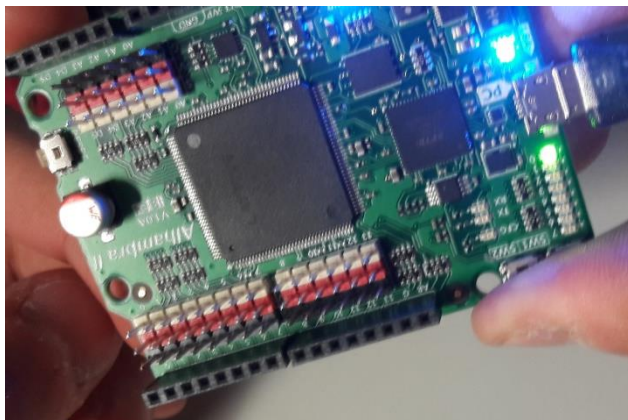
El funcionamiento de la FPGA es mas sencillo que el de Arduino y tiene una apariencia parecida a la del croclip:



Ningun pulsador---Ningún LED encendido



Un pulsador----Un LED encendidos



Dos pulsadores----Dos LEDs encendidos

## Diferencias entre FPGA y Arduino

- Las FPGA son placas, como Arduino, pero aún se trabaja a un nivel inferior. Si Raspberry Pi es un ordenador y lleva muchos procesos a la vez y luego tenemos el microcontrolador Arduino que va un poco más justo, la FPGA se olvida de intermediarios: todo lo que se hace es a nivel electrónico. Se simulan los circuitos electrónicos dentro de la placa.
- En la FPGA la microcomputadora no existe.
- El FPGA se diferencia del arduino en su precio , tamaño y utilidades por ejemplo:
  - Arduino
    - Proyectos informáticos.
    - Instalación interactiva.
    - Prototipos rápidos.
  - FPGA
    - Centros de datos.
    - Medicina.