

Trabajo realizado por Andrés Pérez Guzmán y Javier Ayuso Vázquez.

Circuitos puerta XOR y AND

Pulsador 1	Pulsador 2	Led 1(Puerta XOR)	Led 2(Puerta AND)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

El circuito está compuesto por dos pulsadores, dos leds y un microprocesador o una FPGA , que harán la función de suma binaria con una puerta AND y XOR, en función del programa que vayas a utilizar , Arduino o Icestudio respectivamente.

- La puerta XOR representa una función de una desigualdad, es decir, la salida es 1 solo si las entradas son desiguales, 0+1 ó 1+0. Ecuación : $(A+B) \times (A+B)$
- La puerta AND representa una multiplicación por lo que solo funcionará si las dos entradas son 1. Ecuación: $A \times B$

El código de Arduino sería el siguiente:

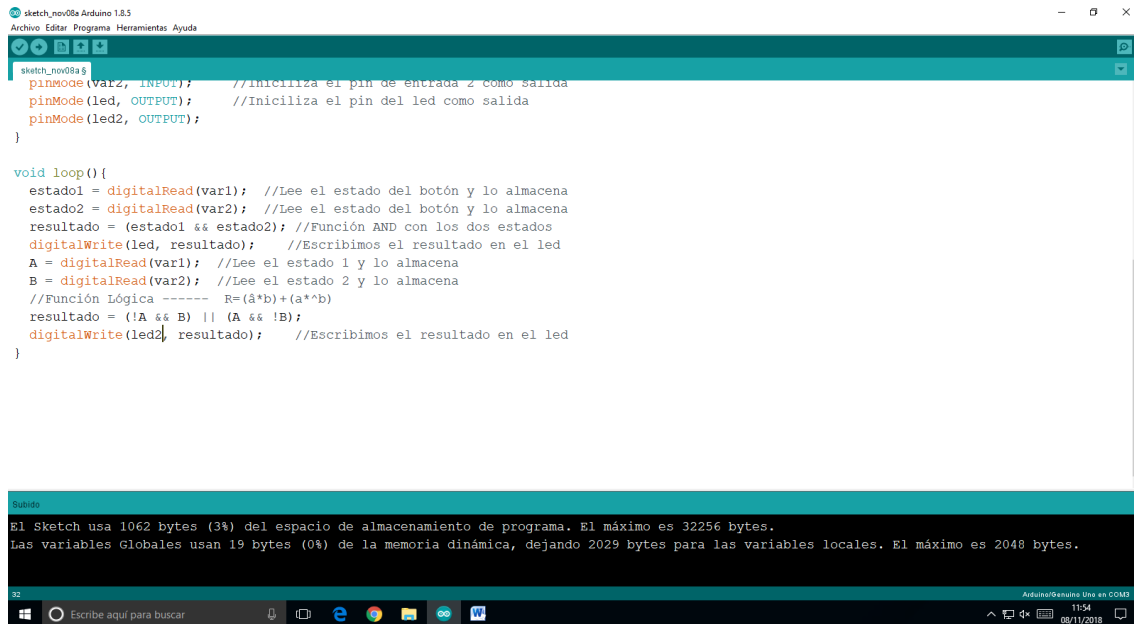
LED NORMALMENTE ENCENDIDO

```
sketch_nov08a Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
sketch_nov08a.g
/*
  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 = 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 resultado = 0; //Para almacenar el resultado
int led2 = 12;
int A = 0; //Para almacenar el estado de la variable1
int B = 0; //Para almacenar el estado de la variable2

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
  pinMode(led2, OUTPUT);
}

void loop() {
  Subido
  El Sketch usa 1062 bytes (3%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
  Las variables Globales usan 19 bytes (0%) de la memoria dinámica, dejando 2029 bytes para las variables locales. El máximo es 2048 bytes.
  30
  Escribe aquí para buscar
  Arduino/Genuino Uno en COM3
  11:53
  08/11/2018
```



```
sketch_nov09a.g
pinMode(var2, INPUT); //Iniciliza el pin de entrada 2 como salida
pinMode(led, OUTPUT); //Iniciliza el pin del led como salida
pinMode(led2, OUTPUT);
}

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
  A = digitalRead(var1); //Lee el estado 1 y lo almacena
  B = digitalRead(var2); //Lee el estado 2 y lo almacena
  //Función Lógica ----- R=(a*b)+(a^b)
  resultado = (!A && B) || (A && !B);
  digitalWrite(led2, resultado); //Escribimos el resultado en el led
}
```

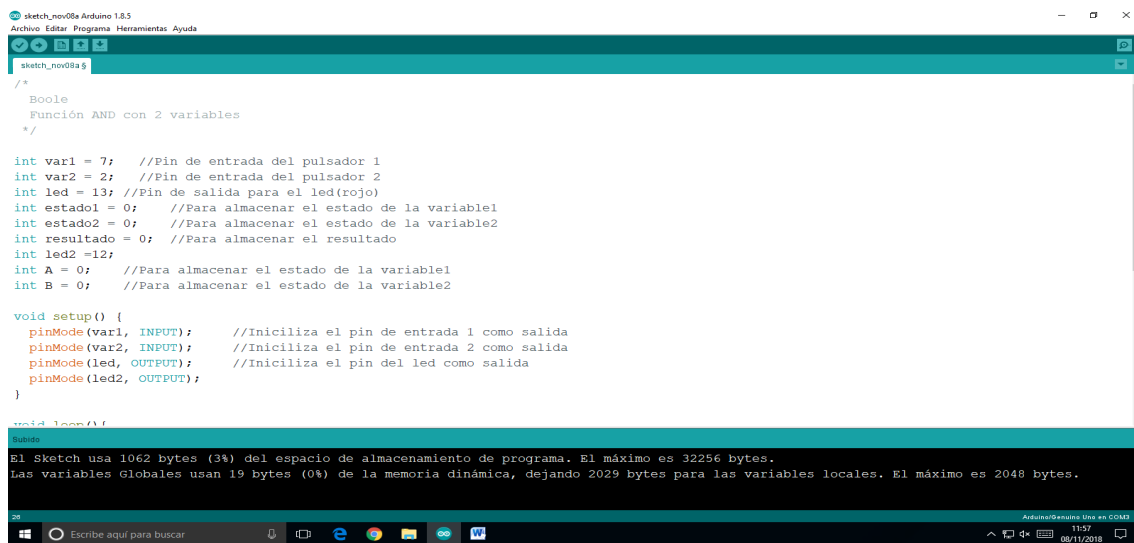
Subido

El Sketch usa 1062 bytes (3%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 19 bytes (0%) de la memoria dinámica, dejando 2029 bytes para las variables locales. El máximo es 2048 bytes.

Arduino/Genuino Uno en COM3

11:54 08/11/2018

LED NORMALMENTE APAGADO



```
sketch_nov09a.g
/*
  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 = 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 resultado = 0; //Para almacenar el resultado
int led2 = 12;
int A = 0; //Para almacenar el estado de la variable1
int B = 0; //Para almacenar el estado de la variable2

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
  pinMode(led2, OUTPUT);
}

void loop() {
}
```

Subido

El Sketch usa 1062 bytes (3%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 19 bytes (0%) de la memoria dinámica, dejando 2029 bytes para las variables locales. El máximo es 2048 bytes.

Arduino/Genuino Uno en COM3

11:57 08/11/2018

```
sketch_nov08a Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

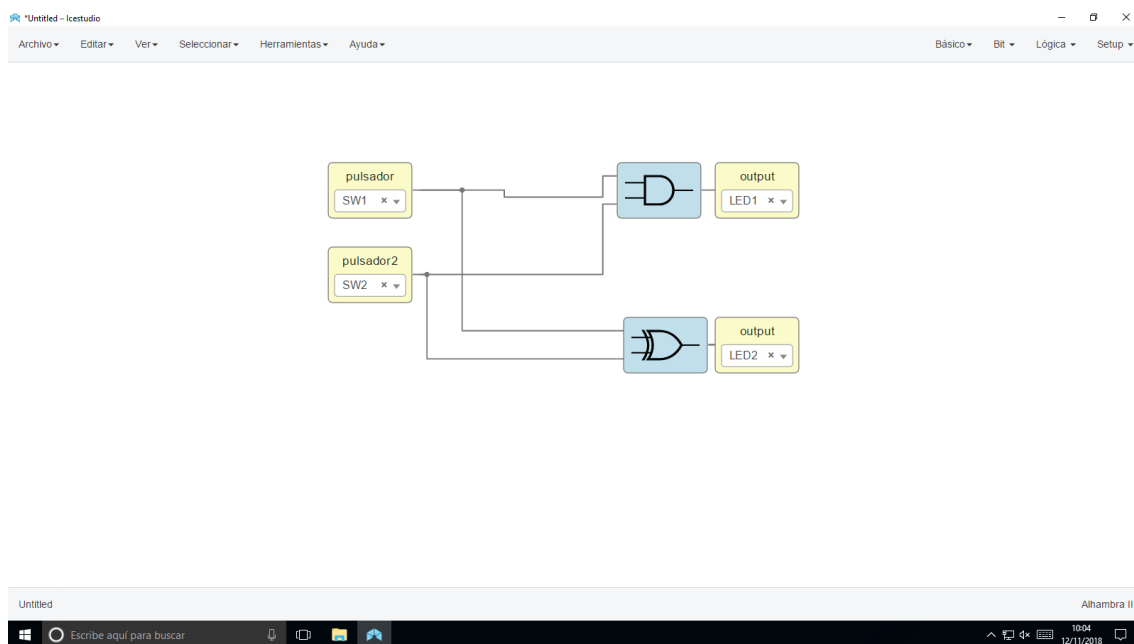
sketch_nov08a
pinMode(var2, INPUT); //Iniciliza el pin de entrada 2 como salida
pinMode(led, OUTPUT); //Iniciliza el pin del led como salida
pinMode(led2, OUTPUT);
}

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
  A = digitalRead(var1); //Lee el estado 1 y lo almacena
  B = digitalRead(var2); //Lee el estado 2 y lo almacena
  //Función Lógica ----- R=(a*b)+(a^b)
  resultado = (!A && B) || (A && !B);
  digitalWrite(led2, resultado); //Escribimos el resultado en el led
}

Sketch
El Sketch usa 1062 bytes (3%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 19 bytes (0%) de la memoria dinámica, dejando 2029 bytes para las variables locales. El máximo es 2048 bytes.

Arduino/Genuino Uno en COM3
11:59 08/11/2018
```

Mientras que desde Icestudio el código quedaría así:



Diferencias entre Microprocesador y La FPGA

- FPGA

Las FPGAs son un tipo de dispositivos electrónicos formados por bloques funcionales unidos a través de un array de conexiones programables. Este dispone de LEDs e interruptores entre otros incorporados de fábrica.

- Microprocesador

Mientras que un microprocesador también conocido como procesador, micro, chip o microchip, es un circuito lógico que responde y procesa las operaciones lógicas y aritméticas que hacen funcionar a nuestras computadoras. En definitiva, es su cerebro.

- Diferencias entre ellos

Aunque en primer momento parece que un procesador y un FPGA son dispositivos similares, porque ambos son capaces de realizar ciertas tareas, lo cierto es que al profundizar es casi más fácil encontrar diferencias que similitudes.

Para entrar en el tema, recordemos de forma muy resumida la forma de trabajar de un procesador. Un procesador contiene una serie de instrucciones que realizan operaciones sobre operadores binarios. Algunos procesadores tienen más instrucciones que otras y es uno de los factores que determinan su rendimiento.

Por otro lado, contiene una serie de registros, que contienen los datos de entrada y salida en las operaciones del procesador. Además, disponemos de memoria para almacenar información.

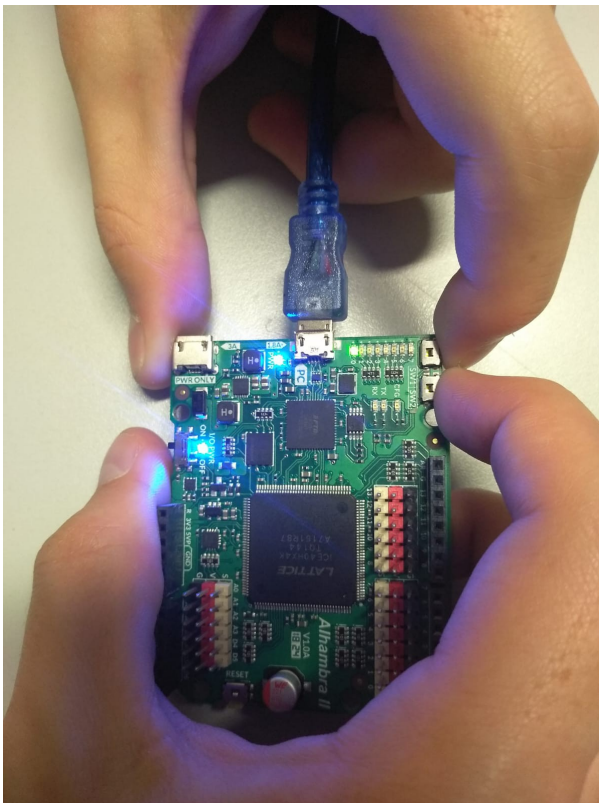
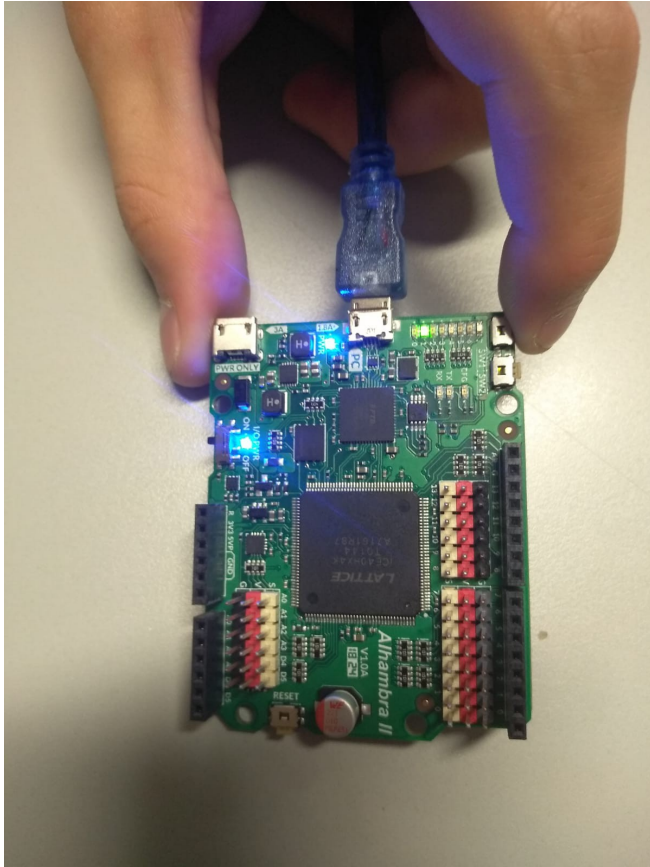
Sin embargo, al programar un FPGA lo que estamos haciendo es modificar una matriz de conexiones. Los bloques individuales están constituidos por elementos que les permiten adoptar distintas funciones de transferencia.

Como vemos, la diferencia sustancial. Un procesador tiene una estructura fija y modificamos su comportamiento a través del programa que realizamos, traducido en código máquina, y ejecutado de forma secuencial.

Sin embargo, en un FPGA variamos la estructura interna, sintetizando uno o varios circuitos electrónicos en su interior. Al “programar” el FPGA definimos los circuitos electrónicos que queremos que se configuren en su interior.

Diferencias gráficas entre ambos

- FPGA:



- Microprocesador:

