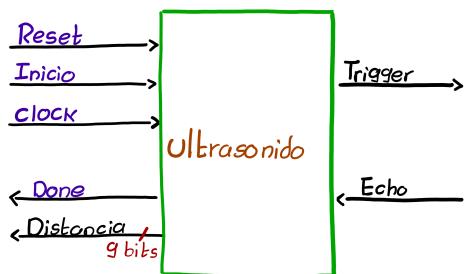
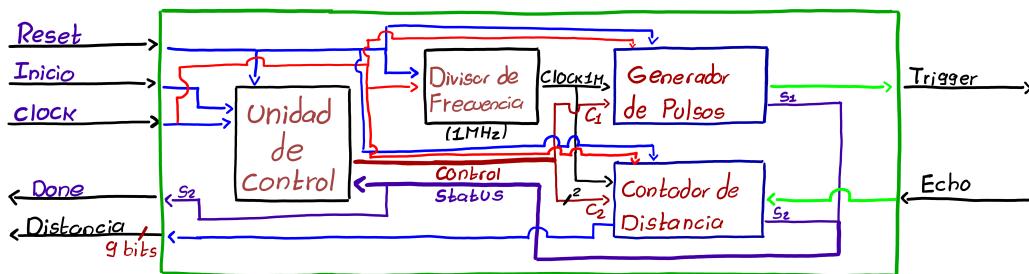


Proyecto Electronica Digital II - Ultrasonido

Modulo Basico (Sin Bus)

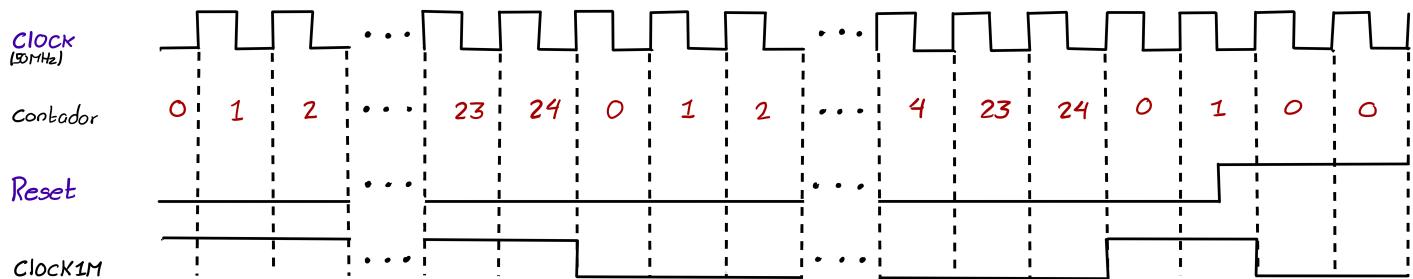


→ Diagrama Estructural

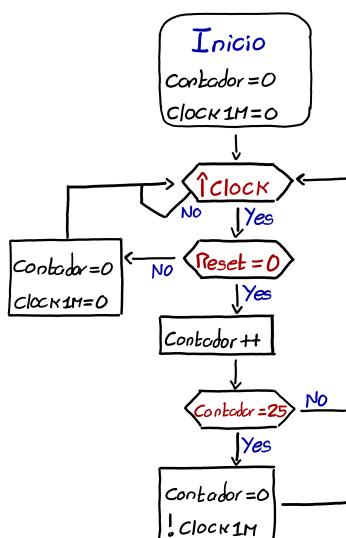


Divisor de Frecuencia

→ Diagrama de señales



→ Diagrama Funcional



Output Reg → Clock1M (1 bit)
Reg → Contador (5 bit)

Input → Clock
Input → Reset

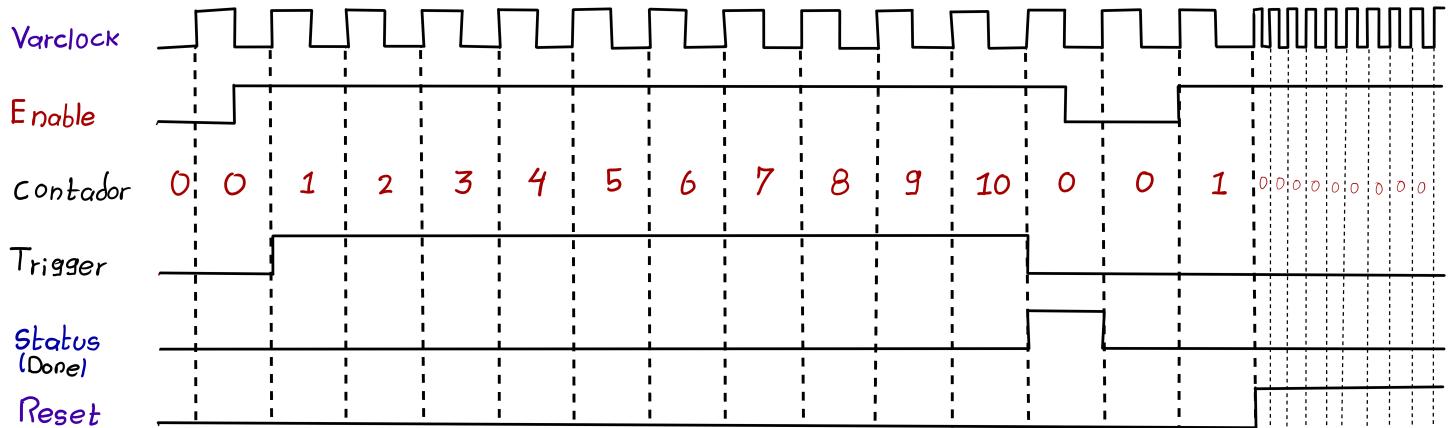
Suponiendo Clock = 50MHz → Contador = 25
el Contador deberá llegar a 25 dos Veces Para generar un Pulso con 50% de Ciclo Util, Por lo cual este Pulso se repetirá Periodicamente Y su frecuencia Será de $f = \frac{50\text{MHz}}{50} = 1\text{MHz}$

Se podría dejar este Parametro como una entrada Y asignar mas bits al contador haciendo un módulo General, Pero en este caso específico conocemos el clock y la frecuencia objetivo

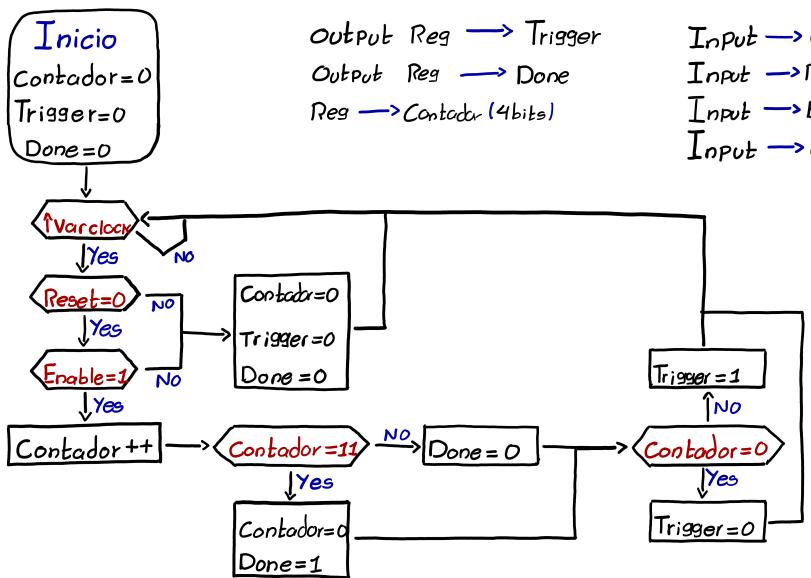
Proyecto Electronica Digital II - Ultrasonido

Generador de Pulso

→ Diagrama de Señales



→ Diagrama Funcional



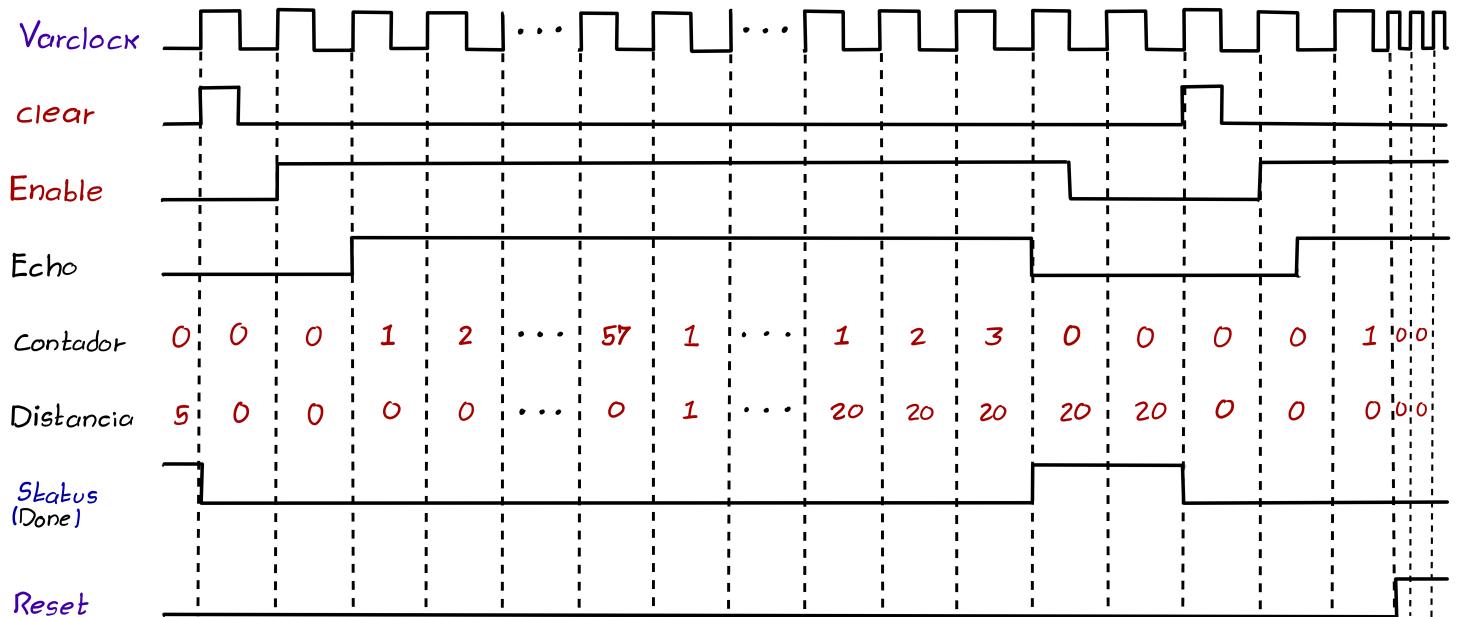
* debido a que Reset se comparte entre modulos este deshabilita el reloj artificial de 1MHz, imposibilitando que los registros se seteen en 0, debido a esto si Reset=1, Varclock combina del reloj de 1MHz al de 50MHz de la FPGA y de esta manera se pueden resetear los registros correctamente.

* Adicionalmente se usa logica bloquante para garantizar la existencia del 0 inicial con los incrementos al inicio, y que Trigger solo funcione entre 1 y 10 para que en el siguiente ciclo la señal de finalizado se active.

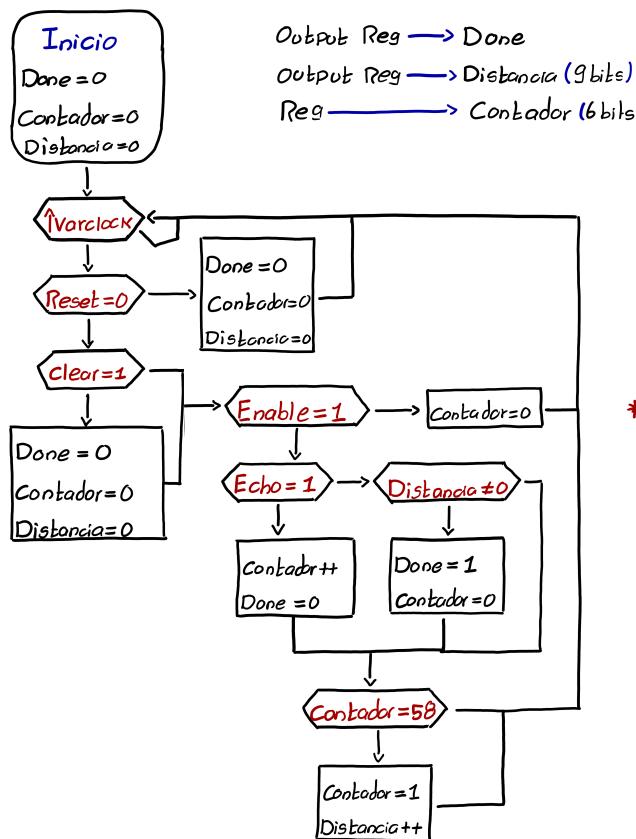
Proyecto Electronica Digital II - ultrasonido

Contador de distancia

→ Diagrama de Señales



→ Diagrama Funcional



Input → Clock
Input → ClockIM
Input → Enable
Input → Clear
Input → Reset
Input → Echo

* Por los motivos mencionados anteriormente se multiplexan los Reloj con el Reset.

* Para tomar una nueva medida es importante Borrar la anterior con clear y de esta manera se ahorra implementar un Registro Adicional de 9 bits.

* La señal cuenta desde el 1 hasta el 57 lo cual serian cuenta de 574S, por ende la distancia tiene una Resolucion de 0.968cm ≈ 1

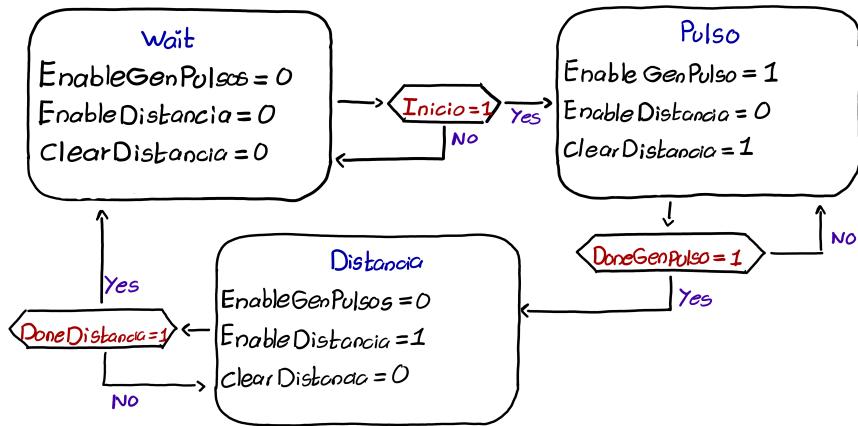
* La señal Done indica si el dato de distancia es valido y se puede utilizar para Conocer si El Proceso finalizo.

UPDATE: Experimentalmente con una Regla y un Osciloscopio Se comprobo que cada 504 Aproximadamente es 1cm.

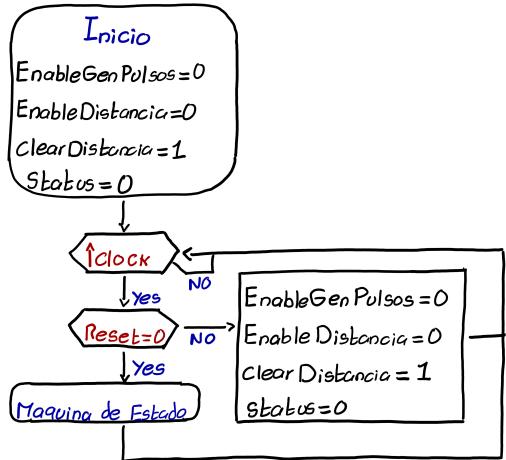
Proyecto Electronica Digital II - Ultrasonido

Unidad De Control

→ Maquina de Estados

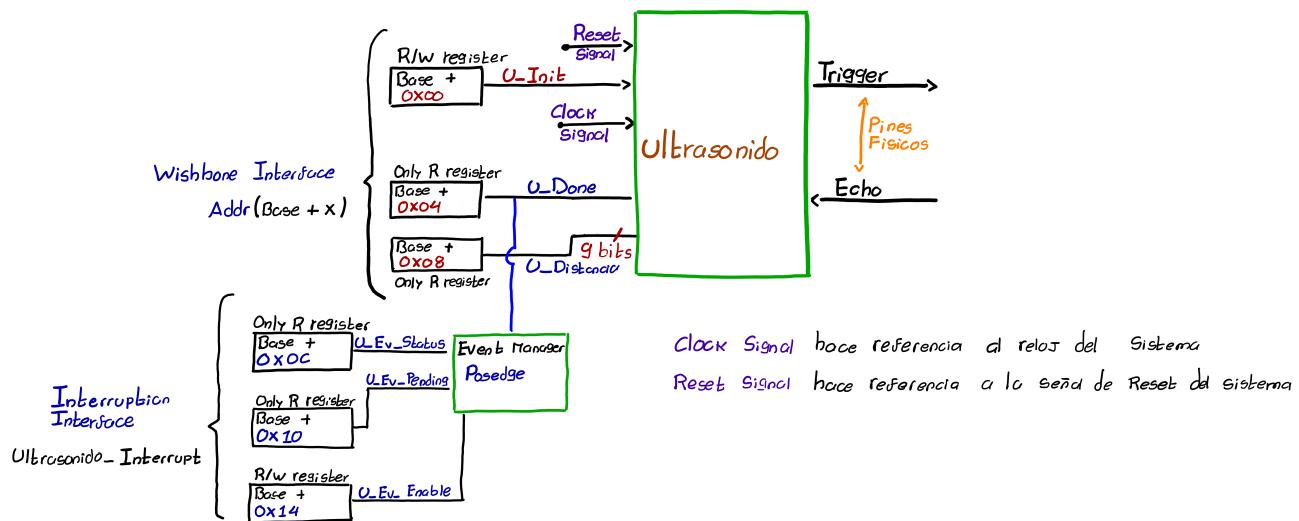


→ Diagrama Funcional



Proyecto Electronica Digital II - Ultrasonido

Modulo wishbone (Incluye Interrupciones)



Codigo C (Picov32 - Interrupciones)

→ debido a que posee Interrupciones se debe modificar **isr.c** agregando:

```
if (irqs & (1 << Ultrasonido_Interrupt)) → si la interrupcion coincide con el vector Ultrasonido_Interrupt
    Ultrasonido_isr();
    ↓
    Ejecuta la funcion
```

1	0	1	1	0	0
---	---	---	---	---	---

↓
Ultrasonido

→ si se activa la interrupcion del ultrasonido

→ Las funciones para hacer posible el funcionamiento son:

Static int Distancia; → Variable de memoria estatica para guardar el resultado de distancia.

```
Ultrasonido_init(){
    Ultrasonido_ev_Pending_Write(Ultrasonido_ev_Pending_Read());
    Ultrasonido_ev_enable_Write(Ultrasonido_ev_Done);
    ↓
    irq_Setmask(irq_Setmask() | (1 << Ultrasonido_interrupt));
    ↓
    habilita la interrupcion de
    Ultrasonido en el
    Vector de Interrupciones General
}
```

X	X	X	X	1
---	---	---	---	---

↓
Done

SubVector de Interrupciones (Ultrasonido)

Proyecto Electronica Digital II - Ultrasonido

Ultrasoundo_Inicio () {

Ultrasonido_Inicio_Write(1); → Activa la señal init actualizando el registro (Base + 0x00)
}

Ultrasonido_isr(){

```
int stat;
```

`Stat= Ultrasonido_ev_Pending_read();` → Lee el subvector de interrupciones y lo almacena

0x01

if(stat & Ultrasonido_ev_Done) { → Si la interrupcion del subVector es Por el evento Done

```
Ultrasonido_ev_Pending_write(Ultrasonido_ev_Done);
```

Ultrasonido - Inicio _write(0); → Descartivo la señal init para evitar multiples llamadas activando el registro (Base + 0x00)

Distancia = Ultrasonido - Distancia - Reed (); → Lee y Almacena el Valor de distancia Pues el dato es valido, Luego el

registro (Base + 0x08)

Ultrasonido_get_Distancia () {

} return Distancia; → Devuelve el valor del static Int