

S.O DE UN HORNO MICROONDAS

Génesis Sofía Méndez Castro

sofiamendezc@hotmail.com

Jose Andrey Sequeira Ruiz

andrey.js08@gmail.com

RESUMEN: El proyecto consiste en la implementación de la lógica combinacional de una máquina de estados finitos, lógica MSI y LSI a través de lenguaje Verilog, para desarrollar un sistema operativo de un horno microondas el cual se observa el funcionamiento a través de la FPGA Nexys 4.

PALABRAS CLAVE: FPGA, decodificador, VERILOG, MSI, LSI, máquina de estados, frecuencia de refresco.

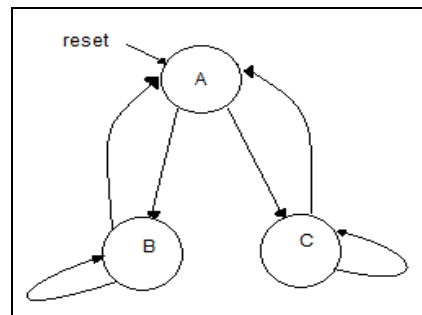


Figura 1. Esquema para guía.

1 MARCO TEÓRICO

Primeramente se definen los conceptos de circuitos de lógica MSI y LSI.

Los circuitos integrados de función fija se clasifican según su complejidad, para este proyecto se trabaja con lógica de integración a media escala o MSI por sus siglas en inglés. Dentro de esta clasificación entran los circuitos integrados (CI) que contienen entre 10 y 100 compuertas en un chip. Estos CI incluyen dentro de sus funciones lógicas codificadores, decodificadores, registros, contadores, entre otros [4].

La segunda clasificación a tomar en cuenta es la lógica de integración a gran escala o LSI, esta categoría incluye a los CI de 100 y 10000 puertas por chip, incluyendo también memorias [4].

2 RESULTADOS EXPERIMENTALES

Para el desarrollo del código en lenguaje verilog primeramente se trabaja con un esquema que permite ser utilizado de guía y poder distinguir los diferentes estados presentes además de los procesos o acciones que realizan cada uno de estos estados. El esquema utilizado se presenta en la figura 1.

El SO del microondas cuenta con un estado inicial o de espera que corresponde al momento donde se muestra el reloj con su respectiva hora, minutos y segundos en formato militar o de 24H, tal y como se muestra en la figura 2.

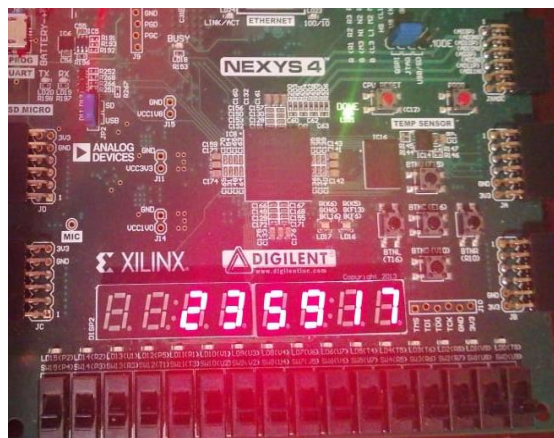


Figura 2. Reloj digital en placa Nexys 4.

Este estado inicial corresponde al estado A de la máquina de estados. Para el funcionamiento de este estado se crea un contador que atenúa entre 0 y 99999999, para cuando este llegue al valor máximo permita un aumento de segundos, así dependiendo del valor del segundo decide si aumentar los minutos y con los minutos decide si aumentar las horas.

El segundo estado corresponde al estado B el cual permite el ajuste de las horas y los minutos del reloj, para acceder a este estado es por medio del switch U9.

Para este estado se utilizan tres botones de la FPGA los cuales para la distribución de la funcionalidad se observa en la tabla 1. En este estado se aumenta un registro que siempre que se ingresa a este tiene valor de 0 como defecto, el aumento en uno de este registro depende de si se presiona o no el botón correspondiente para cada registro (minutos u horas).

Tabla 1. Distribución de botones para estado B.

Botón	Código de PIN	Función
BTNU	F15	Ajuste de Minutos
BTNL	T16	Ajuste de horas
BTND	V10	Cargar valores de ajuste

El último estado es el estado C, en este estado se puede realizar el proceso de calentamiento y posee dos maneras de escoger los tiempos de calentar, la forma de acceder a este estado es por medio del switch R7. Para la primera manera se trabaja con valores de tiempo ya predeterminados en los que los usuarios pueden escoger estos tiempos a través de los switches que posee la placa Nexys 4, el switch correspondiente para cada opción se muestra en la tabla 2 y estos valores se encuentran en el módulo llamado temporizador el cual la salida (tiempo escogido) se carga como entrada al módulo calentar. La segunda manera de escoger un tiempo de calentamiento es por medio de un botón, específicamente del botón BTNC que tiene el código de pin E16, este botón permite escoger un tiempo comprendido entre 0 y 20 segundos de calentamiento.

Tabla 2. Tiempos de calentamientos para el estado C.

Opción	Tiempo	Switch
Descongelar	20 s	P4
Palomitas	10 s	P3
Mantequilla	5 s	R3
Pizza	15 s	T1
Definido por usuario	[0-20] s	T3

Como último detalle del estado C, el botón para iniciar el calentamiento luego de escoger un tiempo es el botón BTNR que tiene como código de pin R10. Al terminar este tiempo de calentamiento se emite una luz de alarma durante aproximadamente 3 segundos, terminando el tiempo de alarma se coloca en 1 una

salida del módulo calentar (finish1) la cual está ligada al switch de reset a través de un operando or.

Al igual que con la salida del módulo calentar se encuentra ligado una salida del módulo del reloj la cual es 1 cuando el botón de cargar ajustes es presionado. En la figura 3 se muestra como se unen estas salida para formar el botón o estado de reset a través del operador or, donde si cualquiera de esos parámetros es 1 el wire reset es 1.

```
assign reset= S_reset | fin | finish1;
```

Figura 3. Unión para formar botón de reset.

Existe la posibilidad de regresar al estado de espera en cualquier momento que se suba el switch R5 el cual se habilita para la funcionalidad de reset.

El wire reset está conectado al módulo de registros y al módulo de la máquina de estados.

Además para facilitar el funcionamiento del sistema de reset se crea un módulo de registro, esto con la finalidad de que al subir algún switch de los estados este mismo pueda ser devuelto al valor de 0 y así en la salida del registro seguirá estando en valor 1, con esto al utilizar el reset no habrá algún switch que interfiera para volver al estado inicial. El módulo de registro corresponde al que se muestra en la figura 4.

```
module RegistrosEstados(
    input clk,
    input reset_E,
    input Est_B,
    input Est_D,
    output outB,
    output outD
);
    //////////////////////////////////////////////////EstadoB////////////////////////////////////
    reg state_B;
    always @(posedge clk) begin
        if(reset_E == 1)
            state_B <= 0;
        else if (Est_B == 1)
            state_B <= 1;
    end
    assign outB = state_B;
```

Figura 4. Módulo de registros.

El controlador del display 7 segmentos posee una frecuencia de refresco que va de la mano con la frecuencia de aumento del segundo, es decir se calcula de una forma que cuando el contador del segundo marque que se aumente el registro de segundos el tiempo de refresco ya pasó por todos los ánodos que deben encenderse (apagando y encendiendo el

siguiente), volviendo al ánodo de los segundos es decir el primer ánodo. Esto se puede observar en la figura 5.

El funcionamiento de este controlador consiste en un registro que se reinicia a 0 cada vez que llega a 6, esto porque cada uno de estos números representan el número del ánodo a encender y el dígito a mostrar en el ánodo. Por ejemplo el contador es 1 por lo que se encenderá el ánodo 1 y se mostrará las decenas de los segundos de igual manera si el contador es 5 se muestran las decenas de las horas en el ánodo 5.

Este contador no aumenta a la frecuencia del ciclo de reloj principal (100 MHz) si no a la frecuencia de refresco que se establece por lo ya mencionado en 12KHz. La frecuencia de los segundos se debe de tomar como referencia para la frecuencia de refresco debido a que estos son los que se mantienen en constante cambio.

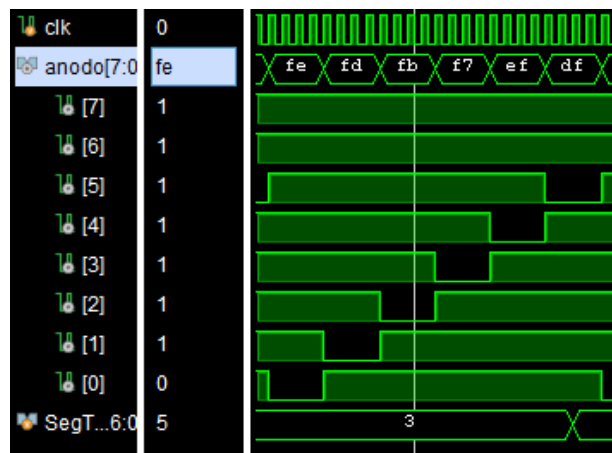


Figura 5. Señal de frecuencia de refresco para el control de 7 segmentos.

Cabe rescatar que la frecuencia de refresco representa un problema durante el desarrollo de este proyecto ya que por ejemplo existía momentos en los que los segundos era 2 s y pero no lo marcaba porque el ánodo que se encontraba encendido era el ánodo de los minutos, y la forma de solucionarlo es por medio del aprendizaje sobre el testbench el cual ayuda a la detección de problema y se soluciona de la manera ya mencionada.

Para la utilización de los botones se presenta una complicación que viene implicada al uso de componentes mecánicos. Este problema es el rebote que experimentan los resortes o contactos que posee estos componentes, problema que puede afectar la señal proveniente de presionar este botón. La señal de un botón mecánico con rebote se puede apreciar en la figura 6.

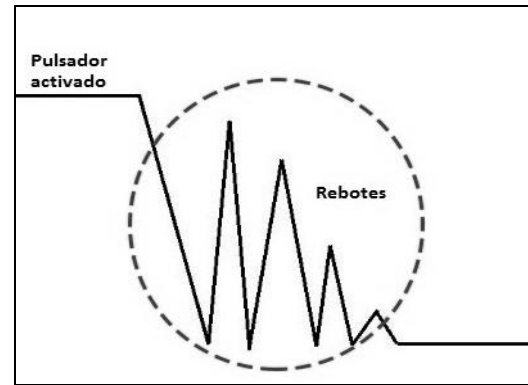


Figura 6. Señal con rebote.

Los rebotes que se presentan en la figura 6 pueden ser interpretados como 1's afectando los resultados esperados, en este caso el aumento de un registro en uno cada vez que el botón sea presionado el cual era incorrecto debido a se aumenta con valores de 3 o 4.

La solución a este problema es el uso de un circuito o sistema supresor de rebotes específicamente el que se muestra en la figura 7. Que básicamente funciona como un tipo de promediador que limpia la señal a través de flipflops que crea un delay y en la salida se obtiene una señal completamente limpia.

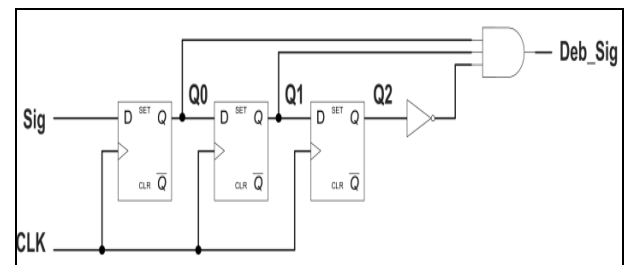


Figura 7. Circuito antirrebote.

Para observar el correcto funcionamiento de este sistema se hace uso de la herramienta testbench que facilita la plataforma vivado con el cual se realiza una simulación de una señal con rebote. El resultado se observa en la figura 8.

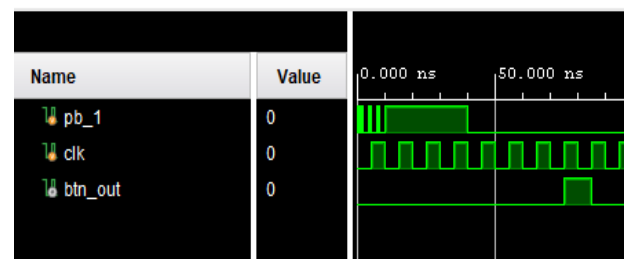


Figura 8. Señal resultante del sistema antirrebote.

El código que describe ese circuito se muestra en la figura 9, donde se observa que se realiza una operación AND con el ciclo de reloj lento que se crea esto con el fin de tener un pulso de duración de un ciclo de reloj principal, como se muestra en la figura 8 con la señal btn_out donde pb_1 es la señal que simula los rebotes..

```
module debounce_better_version(input pb_1,clk,output btn_out);

    wire slow_clk_en;
    wire Q1,Q2,Q2_bar,Q0;
    reg [3:0] numresta=0;

    reg [4:0] NumEscogido=0;
    reg estado;
    integer counter_v=0;

    localparam numero=8;

    clock_enable u1(clk,slow_clk_en);
    my_dff_en d0(clk,slow_clk_en,pb_1,Q0);
    my_dff_en d1(clk,slow_clk_en,Q0,Q1);
    my_dff_en d2(clk,slow_clk_en,Q1,Q2);

    assign Q2_bar = ~Q2;
    assign btn_out = Q1 & Q2_bar & slow_clk_en ; //s slow_clk
```

Figura 9. Código verilog de sistema antirrebote.

Por ultimo durante el desarrollo del proyecto se lleva un control del tiempo invertido para poder concluir este, se incluye tiempo de investigación y análisis de circunstancias que se presentaron. El tiempo invertido se aprecia en la tabla 3.

Tabla 3. Control de horas invertidas.

Día	Horas	Actividad
01/07/2020	7:00pm-12:00pm	Planificación, análisis del proyecto e inicio de código.
02/07/2020	12 md-6:30pm	Desarrollo del módulo reloj.
04/07/2020	7:30pm – 11:00pm	Análisis de controlador 7Seg y uso de botones.
06/07/2020	4:00 pm-11:00 pm	Análisis problema de botones, investigación del mismo.
10/07/2020	8:00 pm-11:30pm	Aprendizaje y uso del testbench con ejemplos.
13/07/2020	9:45 am-4:00 pm	Desarrollo y conclusión de módulos necesarios para estado B
14/07/2020	2:20 pm-8:00pm	Desarrollo de módulos necesarios para estado C

Día	Horas	Actividad
25/07/2020	9:00 am-4:00pm	Solución del problema con el controlador 7 segmentos
26/07/2020	12:00 md-4:00pm	Montaje final de los diferentes módulos y conclusión del proyecto.

En la tabla anterior se observa la cantidad de días y las horas invertidas por cada día y qué se realizó ese día.

3 ANÁLISIS DE RESULTADOS

Después de realizada la prueba a través de la Nexys 4 se logra observar un correcto funcionamiento de las diferentes opciones que posee el SO del microondas, además que los segundos aumentan a una frecuencia muy parecida a lo que lo hace realmente un reloj.

Con respecto al funcionamiento de los botones se logra suprimir el efecto de los problemas mecánicos que presentan este tipo de componentes, problemas como el rebote. Se suprime a través del módulo antirrebote mencionado en la sección anterior. Por lo que con esto se observa la importancia de agregar este sistema de antirrebote al trabajar con componentes mecánicos. Así el código o el circuito no detectará estos rebotes como un 1 y no afectará en el correcto funcionamiento.

La frecuencia de refresco del controlador del display 7 segmentos se logra adecuar de tal manera que se pueda mostrar los valores del reloj correctamente dependiendo de la frecuencia de aumento de los segundos por lo que la solución que se implementa funciona correctamente.

Las horas implementadas para la realización de este proyecto se considera que son muchas, pero la mayor parte del tiempo invertido es en la comprensión y aprendizaje más a fondo sobre lo que es el ciclo de reloj y utilizar el testbench, el cual se considera que es de suma importancia para realizar simulaciones y así tener una mayor comprensión de cómo se comporta el código ante el ciclo de reloj y además en no invertir tiempo en generar bitstream para probar el funcionamiento de algún módulo o algún cambio que se realice. Por lo que el uso de esta herramienta se considera importante para realizar el proyecto de una manera más eficiente y con menos tiempo invertido.

La mayor parte de la investigación y el motivo por el que se aprende a usar el testbench es principalmente para llegar a la solución de los dos problemas que se presentan, el rebote del botón y la frecuencia de refresco del decodificador.

Igualmente es de suma importancia realizar un esquema de ideas que permita tener un camino que seguir durante el desarrollo del proyecto.

4 CONCLUSIONES

- Para un mejor desarrollo del proyecto es necesario un buen entendimiento de los ciclos de reloj con que se trabajan
- Recordar que al trabajar con componentes mecánicos estos no son ideales por lo que necesitan de diferentes sistemas para un correcto funcionamiento.
- Una buena planificación a través de diagramas de flujos y de estados para abarcar un problema ayuda a solucionar este de una manera más eficiente y ordenada.
- Aprender el uso del testbench es importante cuando se trabaja con aplicaciones que dependen del ciclo de reloj.

5 REFERENCIAS

- [1] Dally William J, Curtis Harting R, Digital Design a Systems approach. Cambridge University press, 2012.
- [2] Nexys4 Vivado Tutorial. Xilinx. Online. 2013.
Disponble en:
https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2013x/Nexys4/Verilog/docs-pdf/Vivado_tutorial.pdf
- [3] Nexys 4 FPGA Board Reference Manual. [En linea].
Disponble en:
https://reference.digilentinc.com/_media/nexys/nexys4:nexys4_rm.pdf .
- [4] Floyd, Thomas L., Fundamentos de sistemas digitales, Prentice Hall, 7ª edición, Madrid, España, 2000.