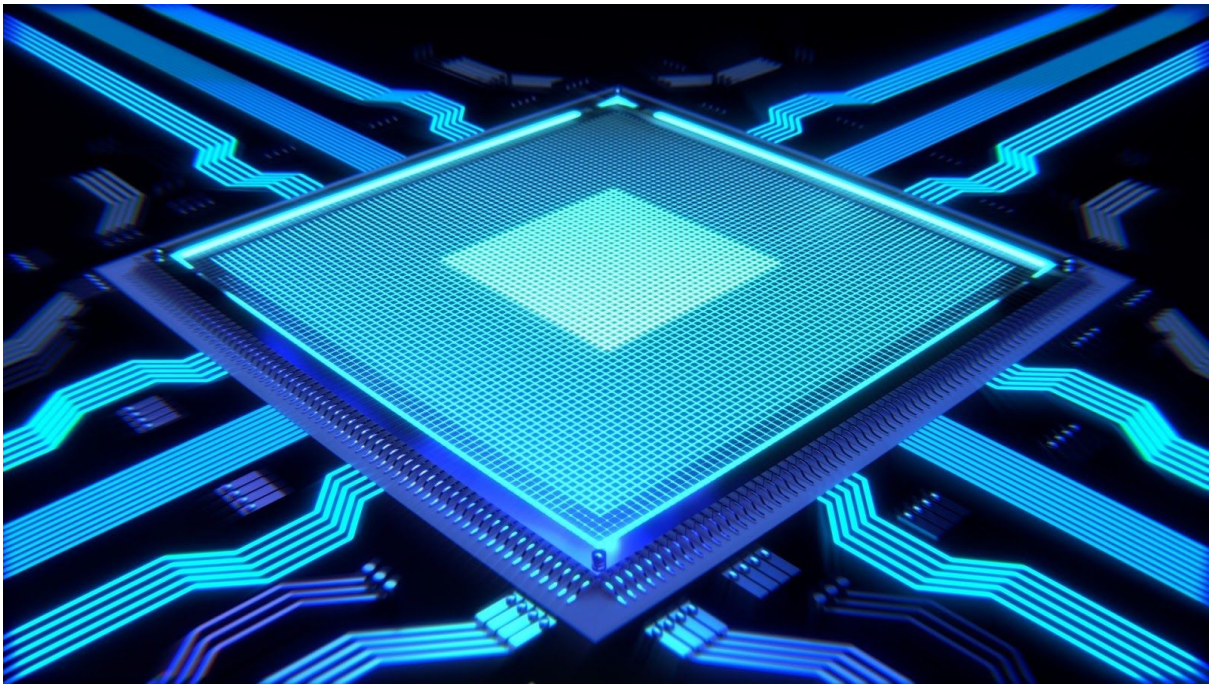
Informe

Implementación de la CPU

Diseño de Procesadores



Diego Cruz Rodríguez
Universidad de La Laguna
Ingeniería Informática
3º Curso, 2º Semestre
11/06/2020

Índice

Modificaciones realizadas en el proyecto 1

Diagrama de Camino de datos 2

Codificación de las instrucciones 2

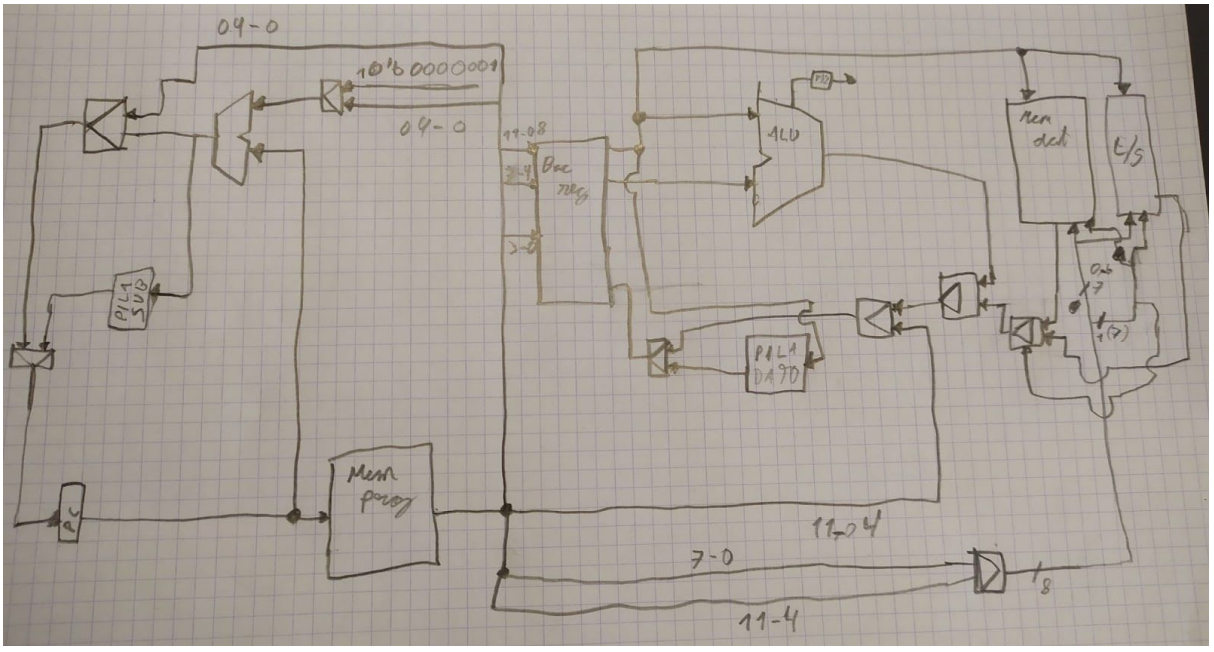
Implementación del programa simple (leds) 4

Referencias 4

Modificaciones realizadas en el proyecto

En el proyecto de la CPU he añadido una pila de datos, una pila de subrutinas.y una memoria de datos, con esta memoria de datos he realizado el direccionamiento a la entrada-salida de forma que cuando la dirección que se pone en las instrucciones de memoria empiezan con un 1 se trata de la entrada salida.

Diagrama de Camino de datos



Codificación de las instrucciones

El repertorio está basado en tres esquemas de instrucciones.

- Primer esquema: estas instrucciones cuentan con tres registros implementados de la siguiente forma. Los bits 0-3 se emplean para la codificación del registro resultado de la operación, los bits 11-8 se emplean para el registro del primer operando y los bits 7-4 se emplean para la codificación del registro del segundo operando por lo tanto esta instrucción cuenta con los bit 15-12 para el opcode.

- Segundo esquema: estas instrucciones cuentan con un inmediato/dirección y un registro. dentro de este esquema contamos con dos implementaciones. Una para cuando el registro es el destino y otra para cuando el registro es el origen de la operación. En ambas codificaciones se emplea bit 15-12 para el opcode.
    - Cuando el registro es el origen se emplean los bits 11-8 se emplean para el registro y del 7-0 para el inmediato/dirección.
    - Cuando el registro es el destino se emplean los bits 3-0 se emplean para el registro y del 11-4 para el inmediato/dirección.

- Tercer esquema: estas instrucciones cuentan con 2 campo uno primero para el opcode del bit 15-10 y un segundo campo del bit 9-0 este campos se suele usar como dirección de programa.

Bajo estos tres esquemas se han codificado el siguiente juego de instrucciones .

- Movimiento del valor de un registro a otro.
    - Para ello se emplea el opcode 0000, seguido de 4 bits del registro a mover, 4 bits que no se usan y los 4 bits del registro destino.

- Negación lógica de un registro
    - Para ello se emplea el opcode 0001, seguido de 4 bits del registro a negar, 4 bits que no se usan y los 4 bits del registro destino.

- Sumas de dos registros
    - Para ello se emplea el opcode 0010, seguido de 4 bits del registro con el primer operando, 4 bits con el segundo operando y los 4 bits del registro destino.

- Resta de dos registros
    - Para ello se emplea el opcode 0011, seguido de 4 bits del registro con el primer operando, 4 bits con el segundo operando y los 4 bits del registro destino.

- Operación & lógica entre dos registros
    - Para ello se emplea el opcode 0100, seguido de 4 bits del registro con el primer operando, 4 bits con el segundo operando y los 4 bits del registro destino.

- Operación | lógica entre dos registros
    - Para ello se emplea el opcode 0101, seguido de 4 bits del registro con el primer operando, 4 bits con el segundo operando y los 4 bits del registro destino.

- Negación aritmética del registro 1
    - Para ello se emplea el opcode 0110, seguido de 4 bits del registro con el primer operando, 4 bits con el segundo operando y los 4 bits del registro destino.

- Negación aritmética del registro 2
    - Para ello se emplea el opcode 0111, seguido de 4 bits del registro con el primer operando, 4 bits con el segundo operando y los 4 bits del registro destino.

- Carga de inmediato
    - Para ello se emplea el opcode 1000, seguido del inmediato de 8 bits y en los últimos 4 bits el registro destino de la carga de ese inmediato.

- Push pila de datos
    - Para ello se emplea el opcode 1001, seguido 4 bits para el registro a guardar.

- Pop pila de datos
    - Para ello se emplea el opcode 1010, seguido de 8 bit de relleno y los últimos 4 bits para el registro en el que se guarda el dato.

- Guardar datos en memoria o enviar dato a entrada-salida
    - Para ello se emplea el opcode 1011, seguido de 4 bits del registros a guardar y la dirección en la que se va a guardar codificada en 8 bits(el primer bit indica si es memoria o entrada-salida).

- Leer datos en memoria o leer dato de entrada-salida
    - Para ello se emplea el opcode 1011, la dirección en la que se va a guardar codificada en 8 bits(el primer bit indica si es memoria o entrada-salida) y seguido de 4 bits del registros a destino.

- Salto incondicional
    - Para ello se emplea el opcode 111100, 10 bits para dirección.

- Salto JZ
    - Para ello se emplea el opcode 111101, 10 bits para dirección.

- Salto JNZ
    - Para ello se emplea el opcode 111110, 10 bits para dirección.

- Salto relativo
    - Para ello se emplea el opcode 111111, 10 bits para valor a sumar al pc.

- Salto a subrutina
    - Para ello se emplea el opcode 111011, 10 bits paradirecion.

- Vuelta de subrutina
    - Para ello se emplea el opcode 111010.

Para interrupciones tenemos tres bits en los cuales se activar cada interrupción, activar hará que empiece una subrutina codificadas en las últimas partes del código del programa. Cuando se activa la interrupción 1 se ejecuta el salto del última dirección del programa.
Cuando se activa la interrupción 2 se ejecuta el salto de la penúltima dirección del programa. tenemos hasta 7 interrupciones.

Dentro de la carpeta del código en el proyecto, hay un archivo de código de programa que contiene ejemplos de la codificación de las instrucciones.

Implementación del programa simple (leds)

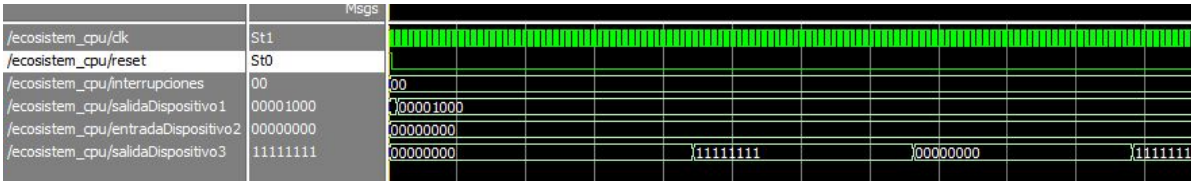
Para hacer la implementación del programa simple, se codificó el timer y posteriormente se codificó un ecosistema para su ejecución de quartus.
En este ecosistema y programa nos encontramos que la cpu interactúa con el timer con el dispositivo 1 y el timer usa la interrupción 001, el dispositivo 2 lo usa el programa para leer las entradas de los botones de usuario. Y en el dispositivo 3 la cpu envía las señales a los leds.

Si se implementara en una placa los 4 botones de la placa se usaría uno para hacer reset del programa, otro para cambiar el modo en el que parpadea los leds y los dos siguientes harían que disminuir y aumentar la velocidad.

El código del programa lo encontraremos en la carpeta del proyecto.
En el código del programa lo encontramos dividido en dos en primer lugar nos encontramos con la subrutina de timer. En ella se comprueba el modo en el que se encuentra actualmente el programa. Y el estado de la secuencia en la que se encuentra de ese modo, y por ultimo actualiza la secuencia de los leds. En el programa principal lo que hace es una pequeña máquina de estados que cuando se deja de pulsar un botón actualiza en consecuencia el modo del programa o aumenta/disminuye la secuencia de los leds.

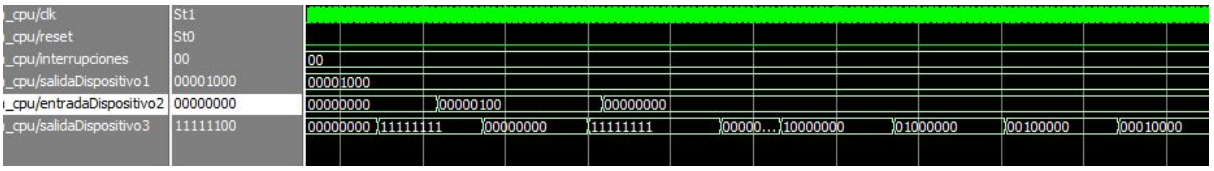
El programa tiene 5 modos, el primero por defecto hace parpadear todas la luces, en segundo lugar desplazamiento hacia la derecha, en tercer lugar incremental hacia la derecha,en cuarto pasao desplazamiento hacia la izquierda y por último acumulativo hacia la izquierda.

Para empezar la ejecución del programa en modelsim asignamos el reloj a la señal clk del ecosistem_cpu, ponemos a 1 la señal reset, y asignamos a la señal de interrupciones 00, y al dispositivo de entradaDispositivo2 00000000. Esto nos permitirá empezar la ejecución del programa tras el primer ciclo de reloj podemos bajar la señal de reset y empezará la ejecución del programa.

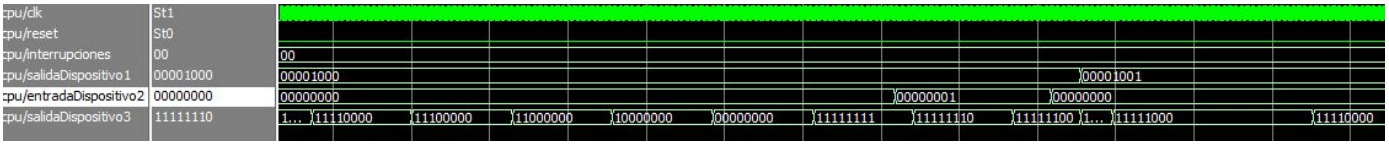


Tras empezar la ejecución del programa podemos observar como en la salida dispositivo 1 se pone a 1000 indicando al timer que empiece a funcionar. A partir de este momento empieza la ejecución del programa y podemos apreciar como cada vez que el timer causa la interrupción se cambia el estado de las luces al estar en el programa por defecto de estar todas encendidas a estar todas apagadas.

A partir de aquí podemos interactuar con la ejecución enviando la secuencia 100, 000 para cambiar el modo de parpadeo de las luces.



Para aumentar o disminuir la velocidad de las luces (por defecto viene en la secuencia mas rapida). Para disminuir la velocidad de las luces enviaremos el código 001 para indicar que pulsamos el último botón y 000 para indicar que soltamos dicho botón así se duplica el tiempo antes de se produzca un cambio de estado.



Para volver a aumentar la velocidad enviaremos la secuencia 010 indicando que hemos pulsado el penúltimo botón y 000 para indicar que lo soltamos.

Las velocidades en las que cambia el programa por defecto son 64 ciclos y partir de aquí se van duplicando siendo la segunda 128, tercera 256, hasta llegar a el octavo que son 8192 ciclos. Cuando intentamos disminuir la velocidad en el octavo vuelve al primero y cuando estamos en el primero y intentamos aumentar la velocidad pasa al octavo.

Referencias

- https://pixabay.com/es/