

PROCESADOR MONOCICLO

PROCESADOR MONOCICLO.

Autor: Jenny Lorena Arango Arias
Ingeniería de Sistemas
Facultad de Ingenierías
Universidad Tecnológica de Pereira
Jeulm333@utp.edu.co

Resumen— En el transcurso del curso realizamos un procesador monociclo, el cual se dividió en varias fases todo este proceso se tuvo en cuenta para que el procesador pudiera realizar instrucciones como lectura de instrucciones de memoria, la decodificación para determinar qué acción es necesaria, La ejecución de una instrucción puede exigir llevar a cabo alguna operación aritmética o lógica .

Abstract— In the course of the course we made a unicycle processor, which divided into several phases, this entire process was taken into account so that the processor could perform instructions such as reading memory instructions, decoding to determine what action is necessary, of an instruction may require carrying out some arithmetic or logical operation.

I. INTRODUCCIÓN

El procesador es el que se encarga de ejecutar las instrucciones específicas por el programa. Este procesador maneja varios datos, uno de ellos es el encargo de manipular, transformar los datos procedentes de la memoria, también tiene como labor el soportar un conjunto de operaciones precisas.

II. HISTORIA

SPARC (del inglés Scalable Processor Architecture). Un procesador con arquitectura escalable es aquel en el que se pueden definir versiones posteriores de la misma con mayor cantidad de características, siempre salvaguardando la compatibilidad con los programas de versiones anteriores. Tiene una arquitectura basada en un juego de instrucciones RISC (Reduced instruction set computer) desarrollado por la empresa Sun Microsystems y lanzado a mediados de 1987.

La implementación de la arquitectura original de 32 bits de Sparc fue inicialmente diseñada en estaciones de trabajo Sun-4 (son una serie de estaciones de trabajo y servidores Unix, lanzados en 1987), reemplazando los sistemas anteriores (Sun-3) basados en la familia de procesadores Motorola 68000.

Ha habido tres grandes versiones de la arquitectura Sparc. La primera de ellas (publicada en 1986) fue la versión V7 (32 bits). Sparc v8 (V8) (una versión superior) fue lanzada en 1990. La principal diferencia entre estas dos versiones fue la introducción de operaciones de producto y cociente de números enteros. La versión V8 deSPARC ha servido como base para Estándar IEEE 1754-1994, un estándar IEEE para una arquitectura de microprocesadores de 32 bits.

III. DESARROLLO

- a. Para la construcción de nuestro procesador se requieren varios pasos.

Se necesitara enviar el contenido del PC a la memoria, luego cargar la instrucción, leer varios registros dependiendo de las instrucciones dadas.

También se manejara un ciclo de instrucciones el cual lee varios registros, utilizando para ello los campos específicos seleccionados

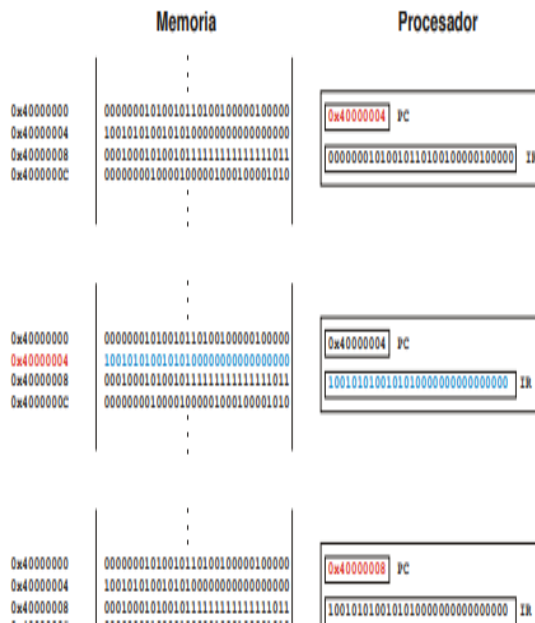


Figura 1. Ciclo de Instrucciones

Se sabe que este programa está formado por una secuencia de instrucciones, donde cada una cumple con una labor además los datos informan donde están.

Para nuestro procesador utilizaremos las siguientes instrucciones:

LOAD Y STORE: Estas instrucciones usan dos registros y una constante para calcular la dirección de memoria

ARITMETICOLOGIAS: Ejecutan operaciones aritméticas y lógicas donde se calcula si es una función de 2 operadores y se guarda en un registro el resultado.

READ/WRITE: Estas instrucciones leen y graban el contenido de los registros de control

Como la mayoría de los procesadores SPARC tiene instrucciones divididas

Las instrucciones son las siguientes:

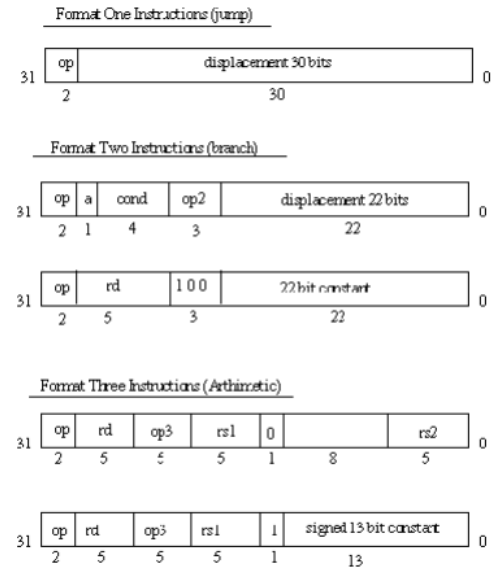


Figura 2. Formato de Instrucciones

El formato 1 se utiliza en las instrucciones de CALL, cuando esta instrucción se transfiere a la nueva ubicación dada por los 30 bits.

El formato dos se utiliza para instrucciones de Branch y Sethi.

Y el formato 3 utiliza instrucciones algebraicas o LOAD/STORE.

Estas instrucciones tienen un registro de destino, un registro fuente 1, un registro fuente 2 y un inmediato de 13 bits.

Instrucciones suma (Add):

opcode	op3	operation
ADD	000000	Add
ADDcc	010000	Add and modify cc
ADDX	001000	Add with Carry
ADDXcc	011000	Add with Carry and modify cc

Instrucciones suma (Add):

opcode	op3	operation
ADD	000000	Add
ADDcc	010000	Add and modify icc
ADDX	001000	Add with Carry
ADDXcc	011000	Add with Carry and modify icc

Instrucciones Lógicas:

opcode	op3	operation
AND	000001	And
ANDcc	010001	And and modify icc
ANDN	000101	And Not
ANDNcc	010101	And Not and modify icc
OR	000010	Inclusive Or
ORcc	010010	Inclusive Or and modify icc
ORN	000110	Inclusive Or Not
ORNcc	010110	Inclusive Or Not and modify icc
XOR	000011	Exclusive Or
XORcc	010011	Exclusive Or and modify icc
XNOR	000111	Exclusive Nor
XNORcc	010111	Exclusive Nor and modify icc

Instrucciones save and restore:

opcode	op3	operation
SAVE	111100	Save caller's window
RESTORE	111101	Restore caller's window

Figura3. Formatos de Instrucciones

Existen 4 grupos de registros diferentes:

Registros Globales: G0-G1
 Registros de Salida: O0-O7
 Registros Locales: L0-L7
 Registros de Entrada: I0-I7

- b. Para el desarrollo del procesador se creó los siguientes módulos:

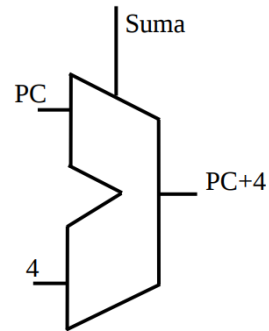
1. Sumador

Figura 4. Modulo Sumador

Entrada: Contiene una entrada que viene del PC de 32 bits y en ella se contará la instrucción actual que se ejecuta.

Salida: Contiene una salida de 32 bits y en ella se cuenta la instrucción del próximo programa.

Función: Su función es recibir el número actual y sumarle un 1.

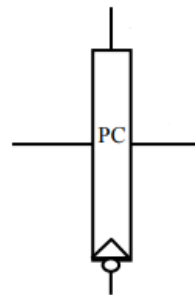
2. PC y NPC

Figura 5. Modulo Next Program Counter

Estos dos módulos son exactamente iguales tienen las mismas entradas y salidas por ende los definimos de la misma manera

Entrada: Contiene una entrada de 32bits y este indica la próxima instrucción que ejecutara. También tiene un reloj de 1 bit y un reset de un bit

Salida: Contiene un registro de salida de 32 bits que indica la siguiente instrucción que realizará el programa

Función: Se requiere un registro para almacenar la dirección de la instrucción que está en ejecución (PC), ya que este valor debe sostenerse para formar la

suma PC+1; y el nuevo contenido de PC, en caso de bifurcación o salto.

Cada instrucción está en una dirección de memoria dada y almacenada en la dirección en el registro PC, después de procesar una instrucción se pasa la información hasta la siguiente instrucción.

3. INSTRUCCIÓN MEMORY

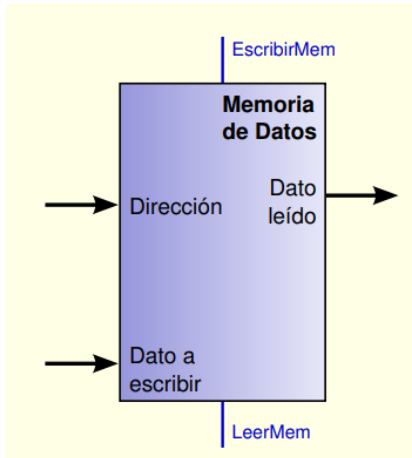


Figura 6. Módulo Instrucción de Memoria

Entrada: Contiene una entrada de 32 bits

Salida: Contiene una salida de 32 bits que llegará al RegisterFile

Función: Se requiere una memoria para almacenar las instrucciones que serán ejecutadas; esta memoria si se supone grabada previamente con un programa. El envía la instrucción al RegisterFile y la Unidad de Control

4. RF (REGISTER FILE)

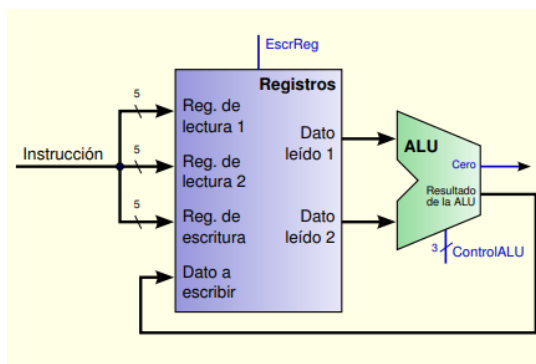


Figura 7. Módulo Register File

Entrada: Contiene las siguientes entradas:

- Registro fuente 1 de 5 bits
- Registro fuente 2 de 5 bits
- Registro de Destino de 5 bits
- Registro de 5 bits que contiene el resultado de la ALU.
- Reset

Salida: Contiene dos salidas cada una de 32 bits.

Función: Este módulo del procesador está formado por 40 registros de 32 bits cada uno.

Los archivos de registro pueden agruparse como bancos de registro. Algunos procesadores tienen varios bancos de registro.

5. ALU

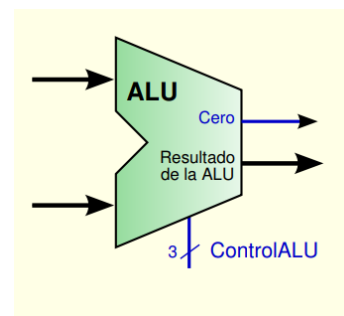


Figura 8. Módulo Alu

Entrada: Contiene dos registros fuentes cada uno de 32 bits

Salida: Contiene un resultado de 32 bits

Función: Se requiere una Unidad Aritmético Lógica que pueda sumar, restar y realizar el or de los operandos de entrada. Se empleará la ALU diseñada anteriormente, la siguiente tabla resume las funciones de la Unidad Aritmético Lógica.

6. UC

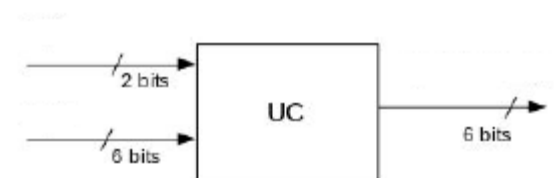


Figura 9. Módulo UC

Entrada: Contiene los siguientes registros:

- Entrada del op1 de 2 bits este registro me indica la operación que deseo realizar.
- Entrada del op3 de 6 bits me dice que operación aritmética puedo hacer

Salida: Tiene una salida de 6 bits que le y le envía a la ALU una señal que me dice que operación debe de realizar.

Función: Las salidas de la UC se encargan de controlar la actividad del resto del dispositivo. Se puede pensar en una UC como una máquina de estado finito.

La UC es la circuitería que controla el flujo de datos a través del procesador, y coordina procesador, que a su vez controla el resto de la computadora.

Algunos ejemplos de dispositivos que requieren una UC son las CPU y las GPU. La edad de la información moderna no sería posible sin diseños complejos de la UC.

Después de haber creado estos módulos se procedió a conectarlos todos los módulos y para ello se creó un módulo donde se creaban señales de cada uno de los mismos para así conectarlos.

Para la segunda parte se procede a la creación de los siguientes módulos que soporten las funciones del ICC

El ICC son bits que son modificados por las instrucciones aritméticas lógicas quienes tienen nombre terminan en la letra cc, son definidos así:

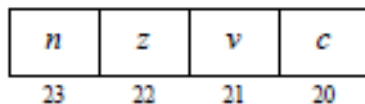


Figura 10. ICC

7. SEU

Entrada: Contiene una entrada del inmediato de 13 bits

Salida: Contiene una salida de 32 bits

Función: en este módulo se recibe un el inmediato de 13 bits para poder completarlo en una señal de 32 bits se le adiciona 19 bits y así la ALU lo pueda operar.

Si el bit más significativo tiene cero eso quiere decir que el número es negativo y se le debe de agregar 17 bit en caso de ser 1 el número es positivo y si se le agregan todos los números en 1

8. MUX

Función: en este módulo se recibe un el inmediato de 13 bits para poder completarlo en una señal de 32 bits se le adiciona 19 bits y así la ALU lo pueda operar.

9. PSR

Entrada: Contiene los siguientes registros:

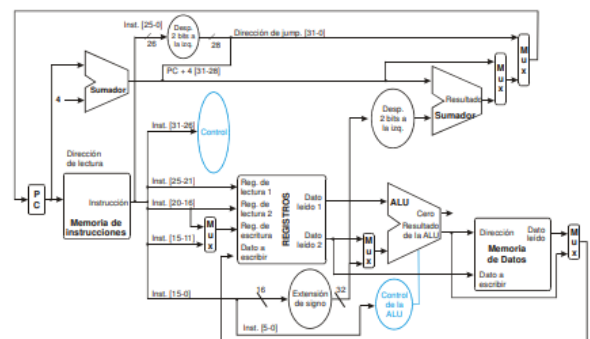
- NZVC de 4 bits
- Reset
- Clock

Salida: Contiene un carry que es el encargado de acarrear una operación realizada

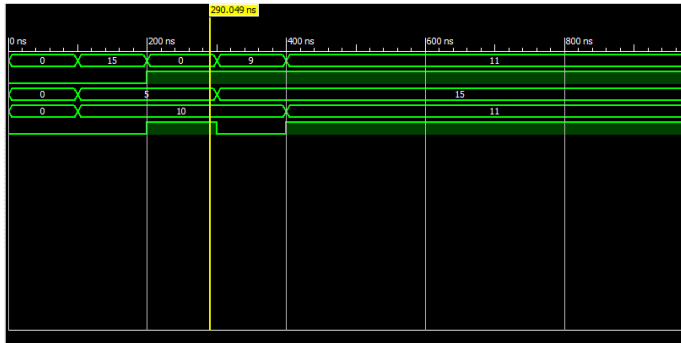
10. WINDOWS MANAGER

Función: Un gestor de ventanas o manejador de ventanas es un programa informático que controla la ubicación y apariencia de las ventanas bajo un sistema de ventanas en una interfaz gráfica de usuario.

Al realizar todos los módulos nos da como resultado el siguiente procesador



Cuando se dio los procesadores al simular el procesador dio varios resultados.



IV. CONCLUSIONES

- Aprendimos todos los elementos del procesador monociclo y cual es cada una de sus funciones.
- Se conocieron los elementos que se necesitaban para implementarlos.
- Se conocieron las instrucciones y cómo implementarlas.

V. RECOMENDACIONES

Para la programar en xilinx se recomienda que no se instale en Windows 10 porque el programa se ejecuta con errores lo más sano para hacer es realizarlo en una versión inferior o en una máquina virtual para que no haya problemas con el programa

VI. REFERENCIAS

Referencias de publicaciones internet:

- [1]. <https://es.wikipedia.org/wiki/Sumador>
- [2]. https://www.infor.uva.es/~bastida/OC/FormatosSPA_RC.pdf
- [3]. [https://es.wikipedia.org/wiki/MIPS_\(procesador\)](https://es.wikipedia.org/wiki/MIPS_(procesador))
- [4]. <http://www2.elo.utfsm.cl/~lsb/elo311/clases/c12.pdf>
- [5]. https://www.uv.es/serhocal/docs/7_diseno_pprocesador.pdf
- [6]. http://profesores.elo.utfsm.cl/~tarredondo/info/comp-architecture/paralelo2/C10_Disenio_Procesador_Monociclo.pdf
- [7]. <https://es.wikipedia.org/wiki/Sumador>

- [8]. <http://www.ensode.net/pdf-crack.jsf>
- [9]. https://es.wikipedia.org/wiki/Unidad_de_control

Referencias de libros:

- [10]. <https://www.gaisler.com/doc/sparcv8.pdf>