

## Diseño del CPU

### Componentes:

#### ***Nombre***

Program Counter.

#### ***Descripción***

Es un registro que almacena la dirección de la instrucción a ejecutar por el CPU.

#### ***Entradas***

##### ***Datapath***

Bit resultado de incremento para seleccionar la instrucción.

##### ***Control***

Reloj para controlar el funcionamiento del componente o un bit de Enable.

#### ***Salidas***

##### ***Datapath***

Dirección de la instrucción para ejecutar.

##### ***Control***

N/A

#### ***Función***

Almacenar la dirección de la instrucción a ejecutar por el CPU.

#### ***Elementos (Logisim)***

Puede representarse mediante un registro de 32 bits.

**Nombre**

Instruction Memory

**Descripción**

Es una memoria donde se almacenan las instrucciones para ser ejecutadas por el CPU.

**Entradas**

**Datapath**

Dirección la instrucción a ejecutar.

**Control**

Reloj para controlar el funcionamiento del componente o un bit de Enable.

**Salidas**

**Datapath**

Instrucción para ejecutarse.

**Control**

N/A

**Función**

Almacenar las instrucciones que serán ejecutadas por el CPU.

**Elementos (Logisim)**

Puede representarse mediante una memoria RAM, para esto sus entradas deben de inicializarse al comenzar la simulación.

**Nombre**

Register File

**Descripción**

Es un componente que almacena todos los registros de propósito general del procesador.

**Entradas**

**Datapath**

Cuatro entradas de 32 bits, las cuales se dividen en dos para la lectura de registros y dos para la escritura en los registros

**Control**

Reloj para control del funcionamiento del componente o un bit en Enable.

**Salidas**

**Datapath**

Datos resultados de la lectura de los registros.

**Control**

N/A

**Función**

Almacenar los registros de propósito general para que el CPU realice la lectura/escritura.

**Elementos (Logisim)**

Puede representarse mediante un conjunto de 32 registros de 16 bits con con multiplexores para el control de entrada y salida de los registros.

**Nombre**

Sign Extend

**Descripción**

Es un componente aritmético que permite aumentar la cantidad de bits en una hilera.

**Entradas**

**Datapath**

Dirección de 16 bits.

**Control**

Reloj para el control del funcionamiento del componente o Enable.

**Salidas**

**Datapath**

Dirección de 32 bits resultado de la extensión de los datos de entrada.

**Control**

N/A

**Función**

Incrementar la extensión de bits de los datos de entrada, en este caso de 16 a 32 bits.

**Elementos (Logisim)**

Se utilizará un bit extender proporcionado por logisim.

**Nombre**

Arithmetic Logic Unit

**Descripción**

Componente con varias entradas que determinan la operación aritmético o lógica a realizar entre dos operandos.

**Entradas**

**Datapath**

Tres entradas, dos para los operandos y una para la operación a realizar.

**Control**

Reloj para controlar el funcionamiento del componente o Enable.

**Salidas**

**Datapath**

Dos salidas, una para una bandera de cero y otra para el resultado de la operación realizada.

**Control**

N/A

**Función**

Realizar una operación lógica entre dos operandos.

**Elementos (Logisim)**

Las entradas pueden representarse mediante dos pines de 32 bits conectadas a componentes aritméticos como sumador, restador, divisor, etc. Para esto la operación a realizar debe de especificarse mediante otro bit que habilite el componente asociado a dicha operación.

**Nombre**

Data Memory

**Descripción**

Componente conformado por celdas donde se pueden almacenar los datos procesador por el CPU.

**Entradas**

**Datapath**

Una entrada de 32 bits correspondiente a la dirección donde se almacenará el dato.

Una entrada de 32 bits correspondiente al dato a almacenar,

**Control**

Reloj para controlar el funcionamiento del componente o un bit de Enable,

**Salidas**

**Datapath**

Una salida de 32 bits para el dato a leer en el Register File.

**Control**

N/A

**Función**

Almacenar los datos a procesar por el procesador, debe de inicializarse al iniciar la simulación.

**Elementos (Logisim)**

Puede representarse mediante una memoria RAM de 32 bits.

**Nombre**

Adders

**Descripción**

Componentes compuestos por dos entradas para determinar los operandos a sumar.

**Entradas**

**Datapath**

Dos entradas de 32 bits correspondientes a los operandos.

**Control**

Bits para el control del acarreo y reloj para el control del componente o bit de Enable.

**Salidas**

**Datapath**

Salida de 32 bits correspondiente al resultado de la suma entre ambos operandos.

**Control**

N/A

**Función**

Sumar dos operandos de 32 bits,

**Elementos (Logisim)**

Será representado mediante un sumador proporcionado por Logisim.

**Nombre**

Shift left by 2

**Descripción**

Componente que desplaza los bits de la entrada dos posiciones a la izquierda,

**Entradas**

**Datapath**

Operando de 32 bits para realizar la operación.

**Control**

Reloj para control del funcionamiento del componente y un bit para especificar la cantidad de bits para hacer “shift”.

**Salidas**

**Datapath**

Resultado de la operación,

**Control**

N/A

**Función**

Desplazar los bits de la entrada 2 espacios hacia la izquierda,

**Elementos (Logisim)**

Será representado mediante un desplazador proporcionado por Logisim.



**Nombre**

IF/ID

**Descripción**

Registro de pipeline de 32 bits para cargar la instrucción y decodificarla, almacena la información que puede utilizarse en la siguiente etapa del pipeline

**Entradas**

**Datapath**

Instrucción de 32 bits a procesar.

**Control**

Reloj para el control del funcionamiento del componente,

**Salidas**

**Datapath**

Instrucción de 32 bits a procesar, si es necesaria para el siguiente paso del pipeline,

**Control**

N/A

**Función**

Almacenar las instrucciones que pueden utilizarse en la siguiente etapa del pipeline.

**Elementos (Logisim)**

Puede representarse mediante un registro de 32 bits.

**Nombre**

ID/EX

**Descripción**

Registro de pipeline de 32 bits encargado de cargar las instrucciones recibidas del Register File mediante dos entradas.

**Entradas**

**Datapath**

Dos entradas de 32 bits correspondientes a las salidas del Register File.

**Control**

Reloj para el control del funcionamiento del componente.

**Salidas**

**Datapath**

Tiene una salida de datos la cual corresponde a los datos recibidos por el register file,

**Control**

Dirección de 32 bits para el Program Counter.

**Función**

Almacenar los datos correspondientes al Register File.

**Elementos (Logisim)**

Puede representarse mediante un registro de 32 bits,

**Nombre**

EX/MEM

**Descripción**

Registro que almacena los resultados de las operaciones del ALU para procesarse en la siguiente etapa del pipeline.

**Entradas**

**Datapath**

Entradas de 32 bits para resultados de la ALU

**Control**

Bit de bandera de cero y un reloj para el control del funcionamiento del componente.

**Salidas**

**Datapath**

Resultado de la ALU.

**Control**

N/A

**Función**

Almacenar los resultados del ALU para utilizarse en la siguiente etapa del Pipeline.

**Elementos (Logisim)**

Puede representarse mediante un registro de 32 bits proporcionado por Logisim.

**Nombre**

MEM/WB

**Descripción**

Registro de 32 bits que almacena los datos leídos de la memoria para utilizarse en la siguiente etapa del pipeline.

**Entradas**

**Datapath**

Dato de 32 bits leído de la memoria,

**Control**

Resultado de la ALU para el control del mutex de salida y reloj para control del funcionamiento del componente,

**Salidas**

**Datapath**

Dato de 32 bits leído de la memoria.

**Control**

Resultado de la ALU para el control del mutex de salida y determinar el dato a escribir en el Register File.

**Función**

Almacenar los datos leídos de la memoria.

**Elementos (Logisim)**

Puede representarse mediante un registro de 32 bits proporcionado por Logisim.