

## **Diseño de Componentes de CPU con “Pipeline”**

Emmanuel Zúñiga Chaves – B98729

Escuela de Ciencias de la Computación, Universidad de Costa Rica

CI-0120: Arquitectura de Computadores

Francisco Arroyo

18 de junio, 2021

## **Diseño de Componentes de CPU con “Pipeline”**

### **Unidad de Control**

#### ***Entradas***

Una entrada de 32 bits correspondiente a la instrucción cargada desde la Instruction Memory.

Entrada del reloj para activar el funcionamiento del componente en el flanco positivo.

#### ***Salidas***

Señales de control de n bits correspondientes a las señales de activación de los demás componentes.

#### ***Función***

Se encarga de procesar la instrucción de 32 bits para generar señales de control para determinar cuáles componentes estarán activos.

#### ***Elementos del Datapath***

Se plantea utilizar registros de diferentes tamaños para almacenar las diferentes partes de la instrucción generadas tras la decodificación o bien, una memoria RAM para que sea más fácil comparar diferentes instrucciones almacenadas.

La ALU-Control forma parte de la datapath, no obstante, su diseño se tratará como otro componente.

#### ***Control***

Como la instrucción se divide en diferentes partes se utilizarán separadores para determinar a que registro se almacena cada parte de la instrucción.

Es necesario utilizar comparadores para controlar la activación de las señales de control, en caso de que una instrucción involucre Hazards.

### ***Suposiciones***

Como se mencionó anteriormente para el diseño se plantea utilizar  $n$  registros o  $n$  memorias RAM para almacenar las diferentes partes de las instrucciones decodificadas, es necesario utilizar comparadores para determinar cuáles señales se activan en caso de que haya instrucciones que compartan valores.

### **ALU-Control**

#### ***Entradas***

Entrada de 32 bits correspondiente a la instrucción a procesar.

Entrada del reloj para activar el funcionamiento del componente en el flanco positivo.

#### ***Salidas***

Salida de 4 bits correspondiente a las 16 posibles operaciones a realizar por la ALU.

#### ***Función del Componente***

Su función es de determinar la operación que la ALU tendrá que realizar a partir de la instrucción a procesar, dicha operación se enviará a la Unidad de Control para el procesamiento de las señales y a la ALU.

#### ***Elementos del Datapath***

Entrada de 32 bits para la instrucción a procesar.

Selector de bits con entrada para 32 bits y salida para 3 bits.

#### ***Control.***

Una constante de 3 bits para controlar que grupo de 4 bits de la entrada corresponde al Opcode.

### ***Suposiciones***

De momento se plantea utilizar un selector de bits para la implementación del componente, este tendría que recibir la instrucción, decodificarla seleccionando los bits del opcode para pasárselo a la Unidad de Control y a la ALU.

## **IF/ID**

### ***Entradas***

Entrada de 32 bits para la instrucción obtenida de la Instruction Memory.

Entrada de n bits para el resultado de la suma del Program Counter + 4.

Entrada del reloj para activar el funcionamiento del componente.

### ***Salidas***

Salida de 32 bits correspondiente a la instrucción a procesar.

Salida de n bits correspondiente al resultado de la suma del Program Counter + 4.

### ***Función del Componente***

Almacenar los datos necesarios generados en las etapas de Instruction Fetch e Instruction Decode para el correcto funcionamiento de las siguientes etapas del Pipelining.

### ***Elementos del Datapath***

Dos registros de 32 y n bits para la instrucción y el resultado del PC+ 4 respectivamente.

### ***Control.***

Entrada del reloj para controlar el funcionamiento del componente en el flanco positivo.

### ***Suposiciones***

Se plantea utilizar dos registros, ambos conectados a las dos entradas descritas anteriormente, de manera que al activar el reloj sea posible almacenar los valores.

## **ID/EX**

### ***Entradas***

Dos entradas de 32 bits para los datos obtenidos mediante la lectura de los registros.

Entrada de n bits para el resultado del  $PC + 4$ .

Entrada de 16 bits correspondiente a los bits [15 - 0] de la instrucción.

Entrada de 5 bits correspondiente a los bits [20 - 16] de la instrucción.

Entrada de 5 bits correspondiente a los bits [15 - 11] de la instrucción.

Tres entradas para las señales de la Unidad de Control, correspondientes a WB, M y EX.

Entrada del reloj para activar el funcionamiento del componente.

### ***Salidas***

Dos salidas de 32 bits correspondientes a los datos obtenidos mediante la lectura de los registros.

Salida de n bits para el resultado del  $PC + 4$ .

Salida de 16 bits correspondiente a los bits [15 - 0] de la instrucción.

Salida de 5 bits correspondiente a los bits [20 - 16] de la instrucción.

Salida de 5 bits correspondiente a los bits [15 - 11] de la instrucción.

Salida del reloj para activar el funcionamiento del componente.

Salidas de las señales WB, M y EX de la Unidad de Control.

### ***Función del Componente***

La función principal de este componente es almacenar las diferentes partes de la instrucción para su utilización en las siguientes etapas del pipeline, igualmente es necesario almacenar los datos leídos del Register File.

### ***Elementos del Datapath***

N registros donde se almacenan los datos obtenidos de las entradas del componente.

### ***Control.***

Es necesario utilizar separadores para controlar que partes de la instrucción a procesar se almacenarán en los diferentes registros.

El reloj controlará el funcionamiento del componente en el flanco positivo, habilitando la escritura en los registros.

### ***Suposiciones***

Como se mencionó anteriormente se plantea utilizar n registros para almacenar los datos de las n entradas del componente, en este caso los registros para los datos obtenidos del Register File serán de 32 bits. Para los demás se utilizarán registros de acorde a la extensión de la parte decodificada de la instrucción, mediante separadores se escogerá en cual registro se escribirán los datos.

## **EX/MEM**

### ***Entradas***

Entrada de 32 bits correspondiente al resultado de la suma de  $(PC + 4) + \text{Instruction}[15 - 0]$  (shift left).

Entrada de 32 bits para el resultado de la operación realizada por la ALU.

Entrada de 1 bit para la bandera de cero de la ALU.

Entrada de 32 bits para almacenar los datos leídos de Read data 2 del Register File.

Entrada para Instruction [15-11] o Instruction [20-16] seleccionada mediante un multiplexor.

Dos entradas correspondientes a las señales de la Unidad de Control, para WB y M.

Entrada del reloj para activar el funcionamiento del componente.

### ***Salidas***

Resultado de la suma entre  $(PC + 4) + Instruction[15 - 0]$ .

Resultado de la ALU, el cual se agrega al Data Memory como Address.

Dato del Read data 2, el cual se agrega como Write Data a la Data Memory.

Salida para Instruction [15-11] o Instruction [20-16] la cual conecta con el Write Data del Register File.

Zero flag el cual conecta con una puerta AND junto a la salida de M para un Branch.

Salidas de M para MemWrite y MemRead de Data Memory.

Salida para señal de WB.

### ***Función del Componente***

Se encarga “transferir” o almacenar los datos producidos en una etapa del Pipeline para utilizarse en las próximas etapas. En este caso almacena los datos producidos en la etapa de EX.

### ***Elementos del Datapath***

N entradas conectadas a N registros del tamaño de cada entrada para almacenar los datos.

### ***Control.***

Entrada de reloj para activar el almacenamiento de los datos en los registros en el flanco positivo.

### ***Suposiciones***

Como se mencionó anteriormente se plantea utilizar N registros correspondientes a las N entradas del registro EX/MEM, pues se supone que estos N registros serán los encargados de transferir los datos a la siguiente etapa del pipeline.

## **MEM/WB**

### ***Entradas***

Entrada de 32 bits correspondiente al Read Data de la Data Memory.

Entrada de 32 bits para almacenar el resultado de la operación realizada por la ALU o Address.

Entrada para Instruction [15-11] o Instruction [20-16] seleccionada mediante un multiplexor

Entrada para señal WB de la Unidad de Control.

### ***Salidas***

Salida de 32 bits para Read Data de la Data Memory, la cual es conectada a un multiplexor.

Salida de 32 bits correspondiente al resultado de la operación de la ALU o Address, la cual es conectada a un multiplexor.

Salida para Instruction [15-11] o Instruction [20-16] la cual es conectada como Write Data del Register File.

Señal de la Unidad de Control para MemtoReg del multiplexor.

Señal de la Unidad de Control para RegWrite del Register File.

### ***Función del Componente***

Se encarga de almacenar los datos producidos durante la etapa de MEM del pipeline, para que sean utilizados en la etapa WB.



### ***Elementos del Datapath***

N entradas conectadas a N registros del tamaño de cada entrada para almacenar los datos.

### ***Control.***

Entrada de reloj para activar el almacenamiento de los datos en los registros en el flanco positivo.

### ***Suposiciones***

Como se mencionó anteriormente se plantea utilizar N registros correspondientes a las N entradas del registro MEM/WB, pues se supone que estos N registros serán los encargados de transferir los datos a la siguiente etapa del pipeline.