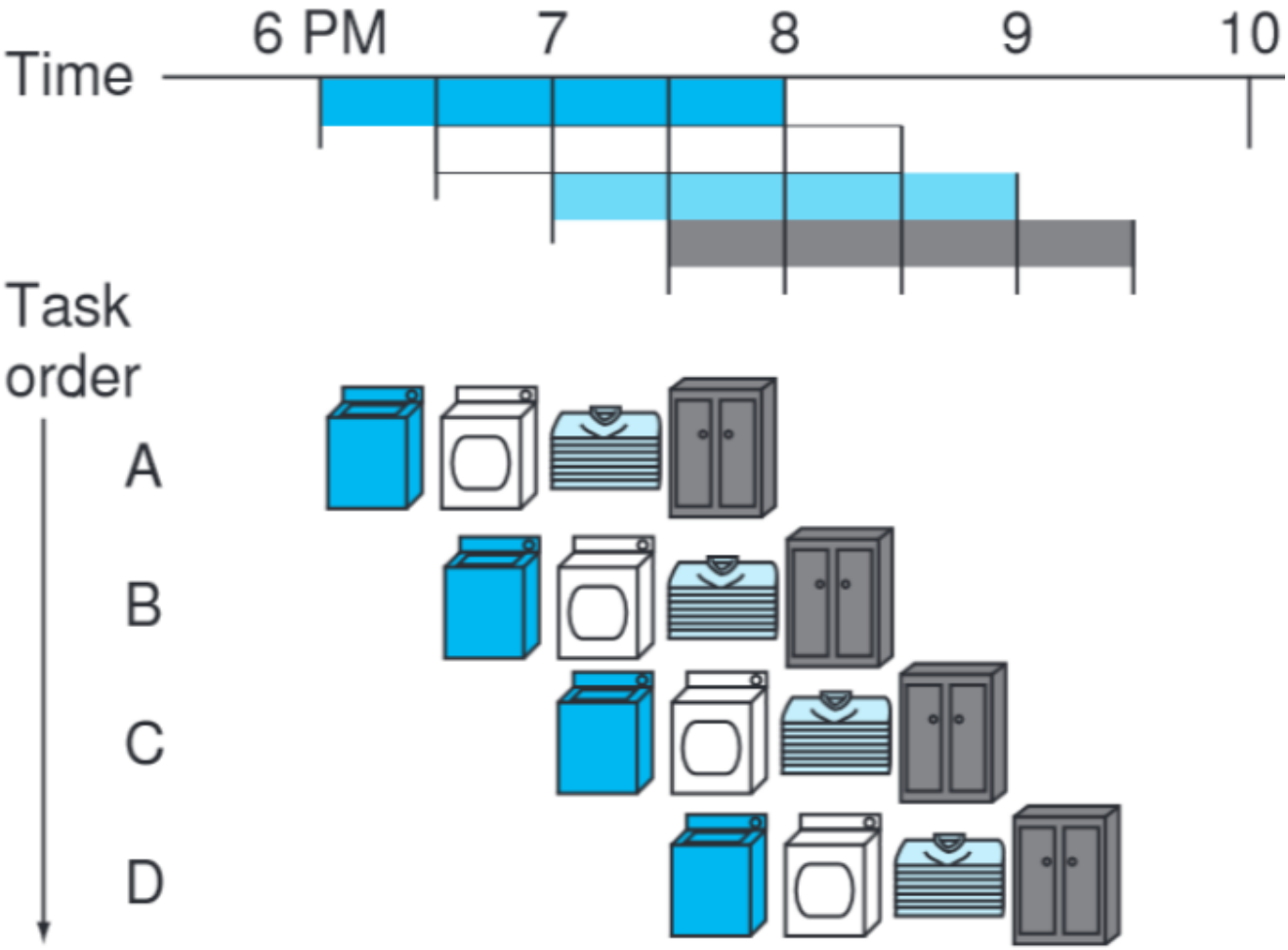
Tarea Corta 3 | B82957

Conceptos previos:

Recordemos que *pipelining* es escencialmente crear una "linea de montaje" para cada una corrida de una instrucción

Cada paso en el *pipeline* se llama **stage**.

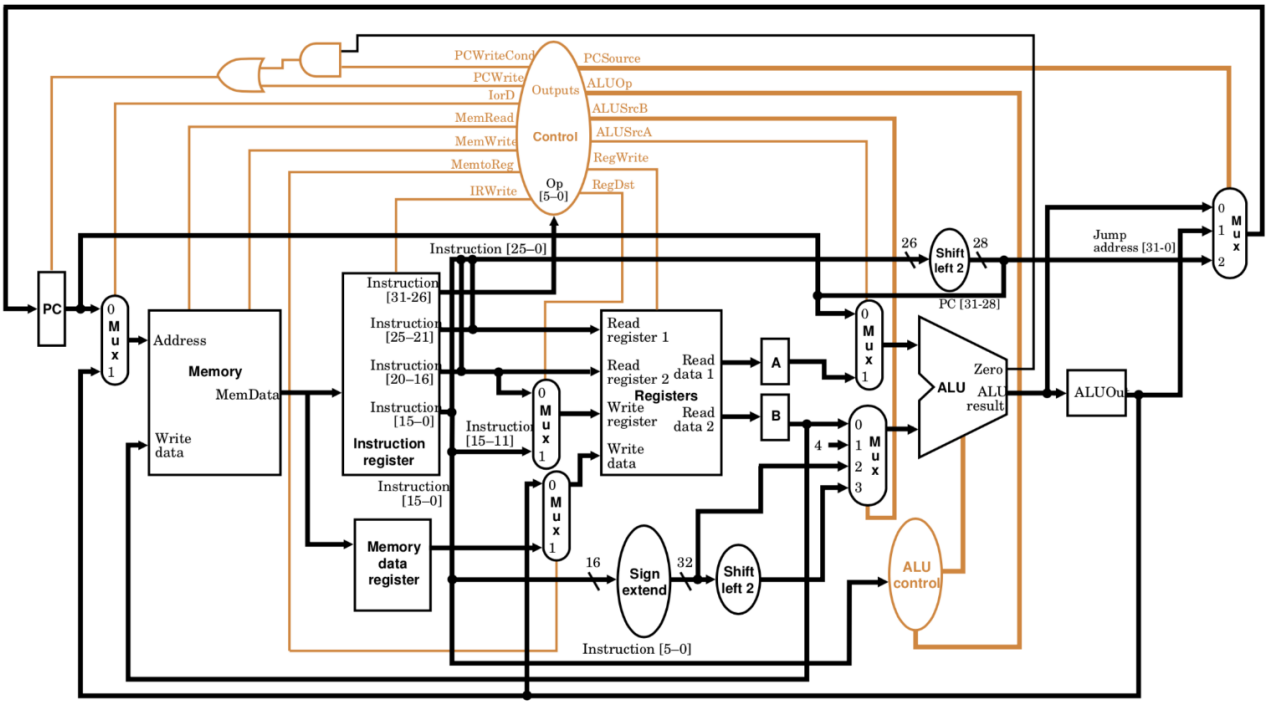


Recordemos el nombre de los stages:

- IF: Instruction Fetch
- ID: Instruction Decode
- EX: Execution or Address Calculation
- Mem: Data memory access
- WB: Write Back

Unidad de Control

MULTI-CYCLE DATAPATH AND CONTROL



Como se aprecia en la imagen, el objetivo de una unidad de control es manejar de manera paralela las operaciones que ocurren en cada **stage**.

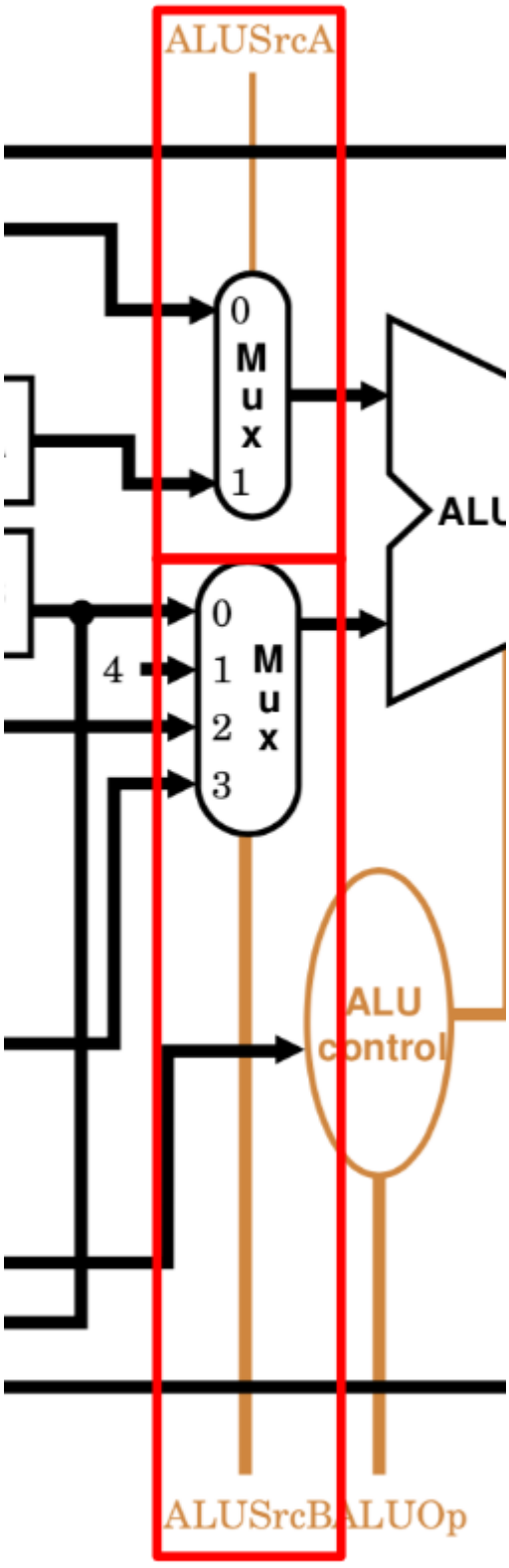
Las señales que se envien a la unidad de control dependen de la etapa en la que se encuentre una instrucción.

Para este diseño, se envian señales de 1 o 2 bits, dependiendo del tipo de señal que se envie.

Señales de 1 bit

- RegDst: Al registrarse esta señal, se obtiene el numero del registro destino para hacerle **write** del **rd field**
- MemToReg: Al registrarse esta señal, el valor de para el input del register file viene de **MDR**, si no, viene de **ALUout**.

- ALUSrcA y ALUSrcB: Señales para saber de donde cuál dónde obtener la información para la ALU:



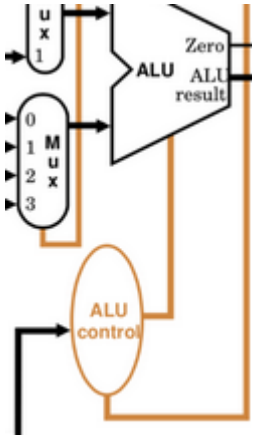
Señales de 2 bits

2-bit Signal	Value	Effect
ALUOp	00	The ALU performs an add operation.
	01	The ALU performs a subtract operation.
	10	The funct field of the instruction determines the operation.
ALUSrcB	00	The second input to ALU comes from the B register.
	01	The second input to ALU is 4.
	10	The second input to the ALU is the sign-extended, lower 16 bits of the Instruction Register (IR).
	11	The second input to the ALU is the sign-extended, lower 16 bits of the IR shifted left by 2 bits.
PCSource	00	Output of the ALU (PC+4) is sent to the PC for writing.
	01	The contents of ALUOut (the branch target address) are sent to the PC for writing.
	10	The jump target address (IR[25-0] shifted left 2 bits and concatenated with PC + 4[31-28]) is sent to the PC for writing.

Lo que más cabe destacar de acá es la señal de PCSource, como es una señal con multivalores escoje que salida de las ALU procesar para write.

ALU Control

La función principal de ALU Control es decodificar instrucciones para determinar que segmento estara activo en el datapath, además de decirle al ALU que operacion hacer.



Instruction Fetch / Instruction Decode

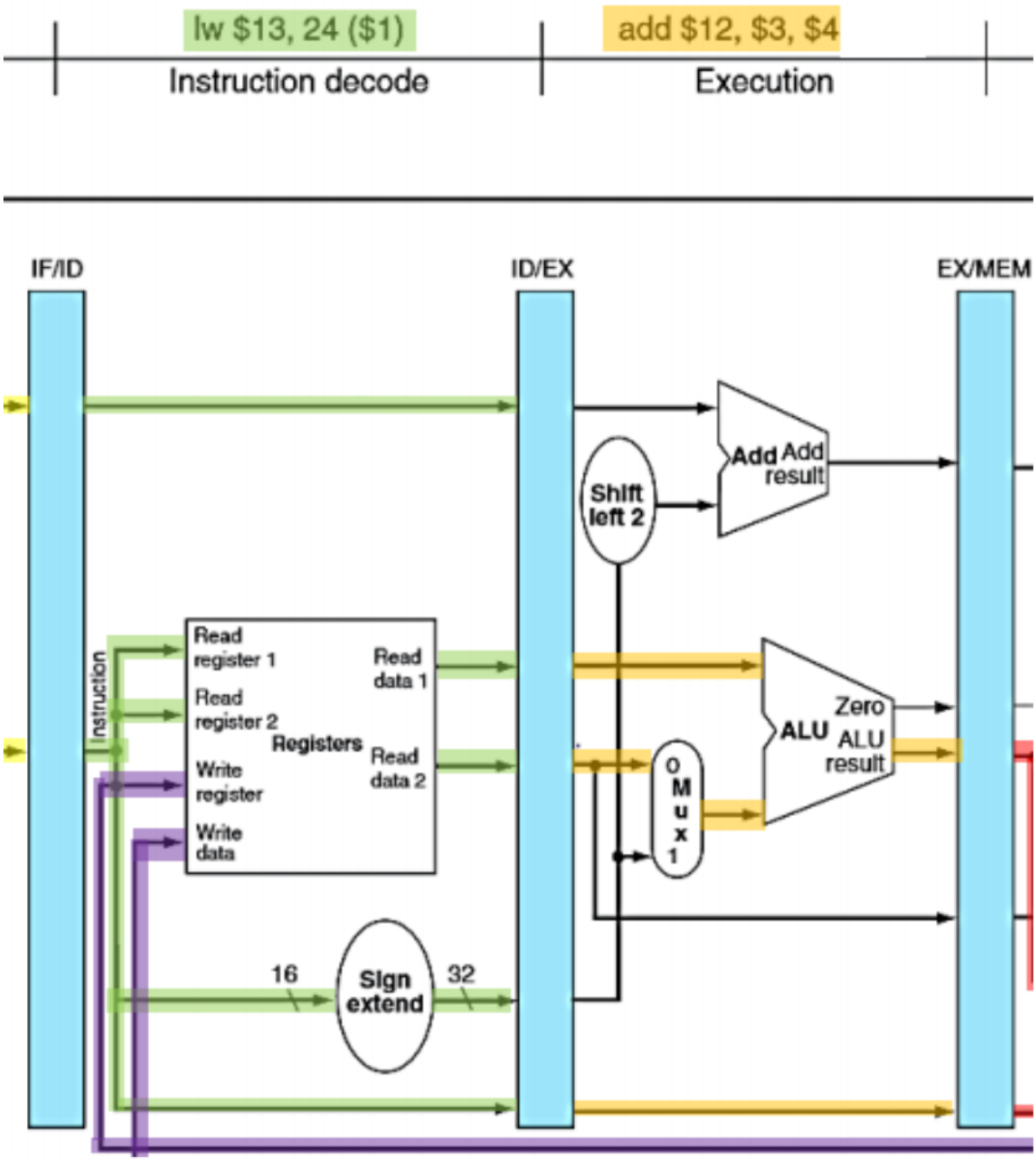
Ingresa la instruccion obtenida desde memoria usando el program counter y se guarda en el IF/ID Register.

Funcionalidad: Lee los registros rs y rt, y los guarda en el register de ID EX. Además, copia el valor del Program Counter + 4 bits desde el IF ID y lo envía al ID EX register.

Intruction Decode / Instruction Execute

Como entradas se tienen los registros rs y rt del register file de 32 bits cada uno. se guardan en el register de ID. Además se obtiene el offset o corrimiento PC+4 enviado desde el IF/ID register.

Funcionalidad:



Como se ve en la imagen, el ID/EX manda 2 salidas al ALU, los cuales son los el rs (1er registro del register file) y el campo inmediato de 32 bits para procesar un add, en el caso de la imagen.

Execution / Memory Access

Recibe el resultado de la operación que se proceso en el ALU del register file y del campo de 32 bits, y lo guarda en su registro.

Se obtiene el resultado de la operación del ALU además de enviar el resultado a la memoria de registros.

Memory Access / Write Back

Recibe un dato leído de memoria en su registro, y dependiendo del signal lo escribe en register file o si hay WB en memoria principal.