

El procesador

Datapath y control

Introducción

- El procesador o CPU (unidad central de procesamiento) sigue las instrucciones del programa al pie de la letra. Suma y compara números, ordena activarse a los dispositivos de I/O, etc.
- El procesador consta de dos componentes:
 - El datapath. Ejecuta operaciones aritméticas y lógicas.
 - El control. Ordena al datapath, memoria y dispositivos de I/O lo que hay que hacer de acuerdo al programa.

MIPS simplificado

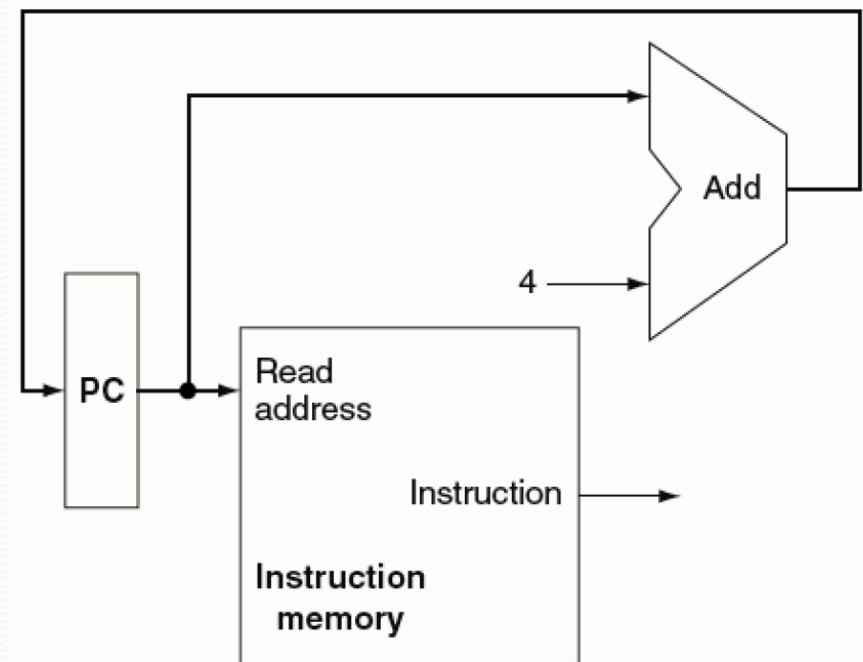
- Se examinará una implementación que incluye un subconjunto de instrucciones MIPS:
 1. Instrucciones de referencia a memoria: load word (lw) y store word (sw).
 2. Instrucciones aritmético-lógicas: suma (add), resta (sub), and, or y set on less than (slt).
 3. Instrucciones de brincos: brinca si igual (beq) y brinco incondicional (j).

Implementación

- La implementación de las distintas instrucciones tiene varias acciones en común.
- Los primeros dos pasos son iguales:
 1. Enviar el PC (contador de programa) a la memoria y sacar la siguiente instrucción (ciclo de fetch).
 2. Leer uno o dos registros del banco de registros.
- Lo siguiente depende de la clase de instrucción, pero es parecido sin importar el opcode exacto.

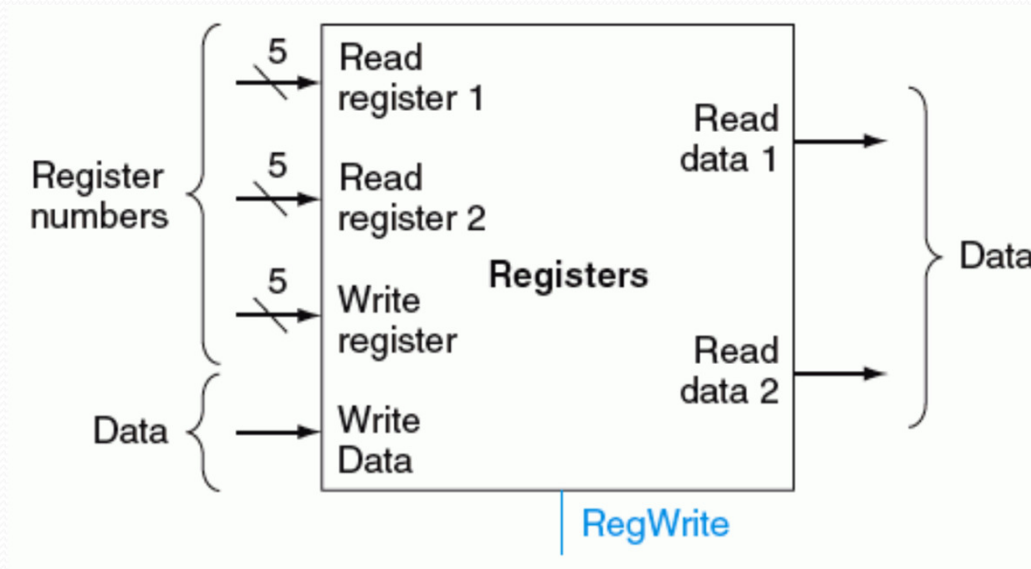
Ciclo de fetch

- Lee una instrucción e incrementa el PC.



Banco de registros

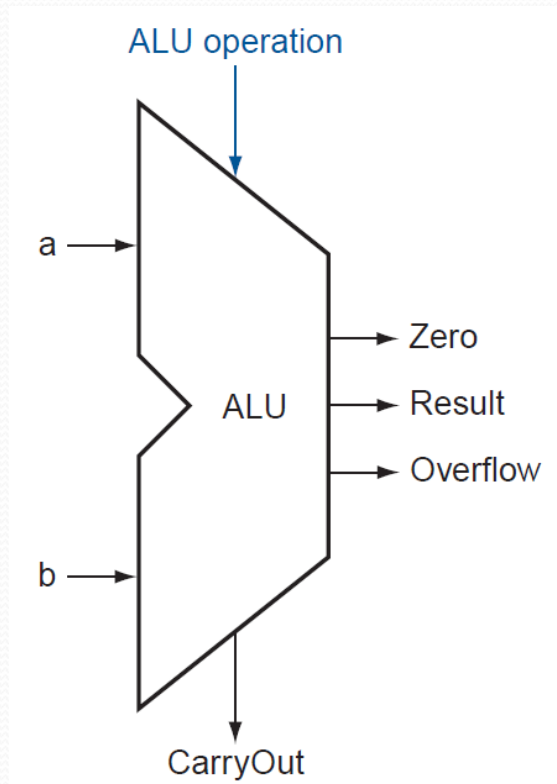
- Dos puertos de lectura y uno de escritura.



Implementación

- Todas las instrucciones, excepto el brinco incondicional (instrucción j), usan la ALU (unidad aritmético-lógica).
- Las instrucciones de referencia a memoria para calcular direcciones.
- Las instrucciones aritmético-lógicas para su operación.
- Los brincos para evaluar la condición.

ALU



ALU operation				Result
C_3	C_2	C_1	C_0	
0	0	0	0	AND
0	0	0	1	OR
0	0	1	0	suma
0	1	1	0	resta
0	1	1	1	set on less than
1	1	0	0	NOR

Implementación

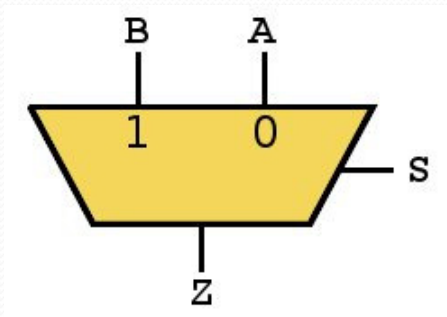
- Después de usar la ALU:
- Las instrucciones de referencia a memoria accesan la memoria para cargar o guardar un dato.
- Las instrucciones aritmético-lógicas guardan el dato de la ALU en un registro.
- Los brincos, dependiendo de la condición, cambian el contador de programa (PC) o lo incrementan en 4.

Implementación

- Se utilizan los siguientes componentes:
- Componentes principales: banco de registros, memoria de instrucciones, memoria de datos y ALU.
- Componentes extras: dos sumadores, dos unidades de corrimiento, cuatro multiplexores y un extensor de signo.

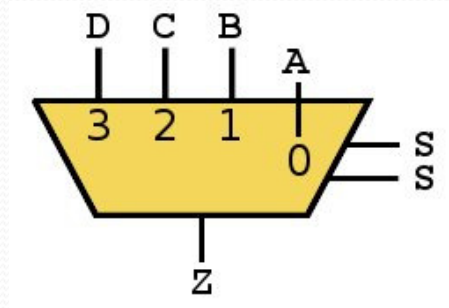
Multiplexor (mux)

- El multiplexor tiene 2^n entradas de datos, una salida y n bits de selección.
- Los bits de selección se usan para decidir cuál entrada pasa a la salida.
- Mux 2 a 1

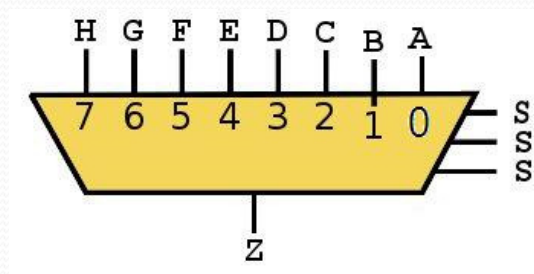


Multiplexor (mux)

- Mux 4 a 1.

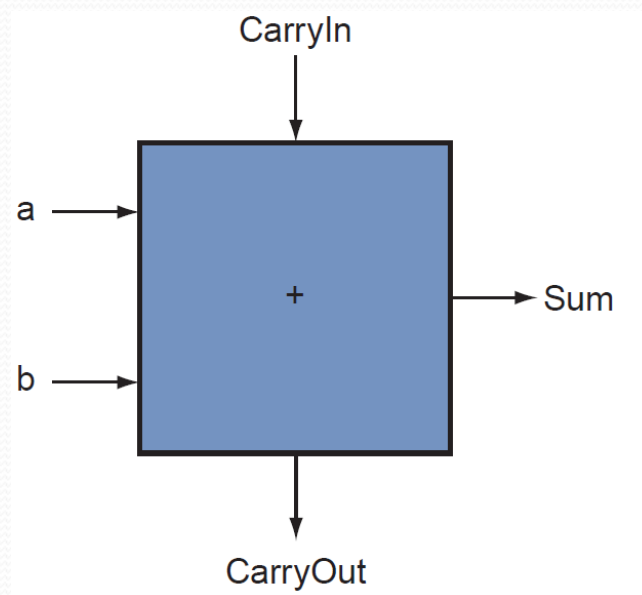


- Mux 8 a 1.



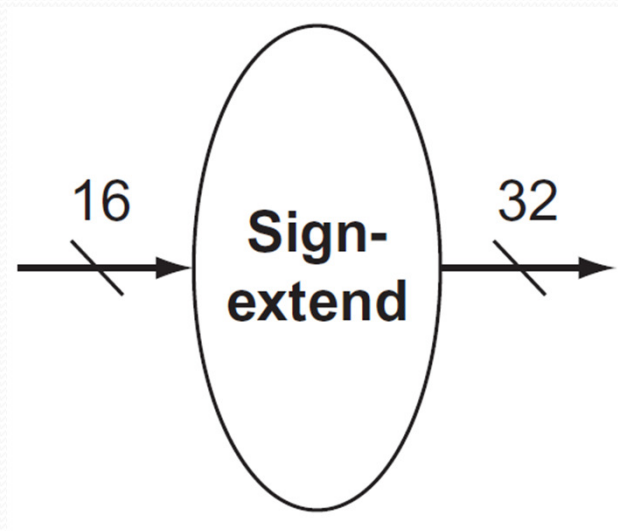
Full adder / sumador completo

- Entrada: 2 números de 32 bits y 1 bit de carry.
- Salida: 1 número de 32 bits y 1 bit de carry.



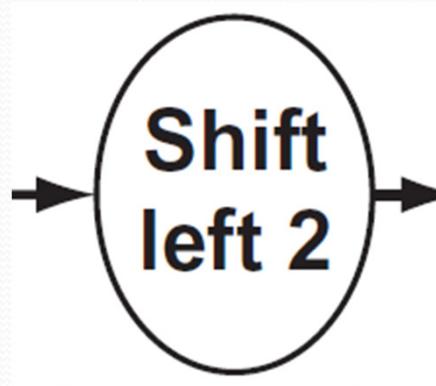
Extensor de signo

- Entrada: un número de 2 bytes.
- Salida: el mismo número extendido a 4 bytes.

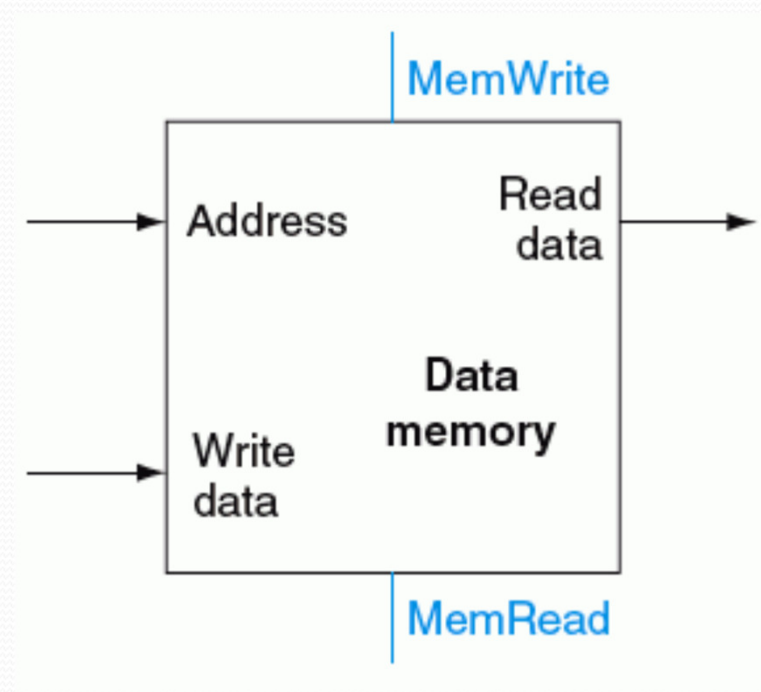


Unidad de shift

- Entrada: un número de 4 bytes.
- Salida: el mismo número recorrido 2 bits a la izquierda.



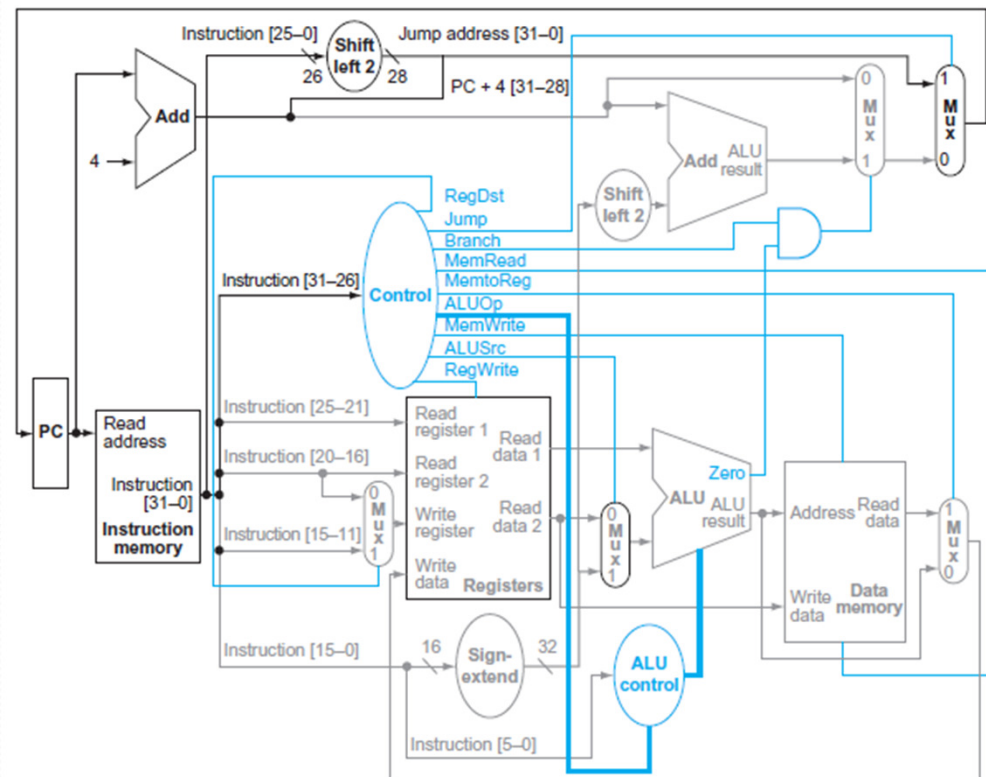
Memoria de datos



Señales de control

Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

Implementación: datapath y control



Fuente: COD 5, p. 271