Procesador

MIPS

de ciclo único de reloj

Diseño de la ruta de datos y control

Diseño del nucleo del MIPS con las siguientes instrucciones:

- Referencia a memoria: LW , SW
- Aritmético-lógicas: ADD , SUB , AND , OR , SLT
- Saltos: BEQ , J

Pasos:

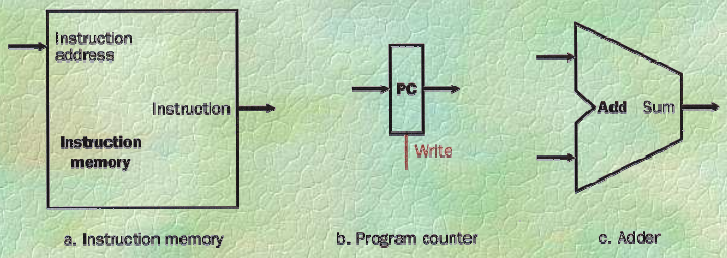
1) Enviar el PC a memoria para traer la instrucción
2) Leer uno o dos registros usando los campos de la instrucción

Los otros pasos difieren según el tipo de instrucción

Todos usan de alguna manera la ALU:

- Ref. a memoria: cálculo de la dirección efectiva
- Arit-lógicas: según el código de operación
- Saltos: para comparación (BEQ)

Componentes necesarios para la extracción de instrucciones



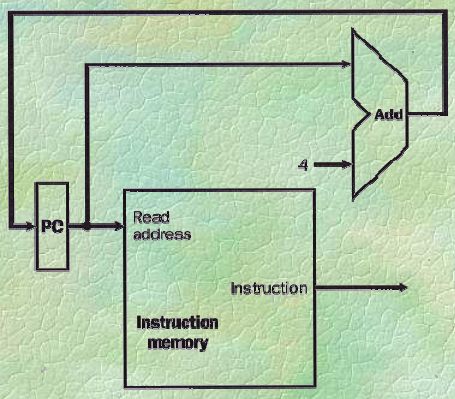
a. Instruction memory b. Program counter c. Adder

3

Subcircuito de extracción de instrucciones

Durante el ciclo actual se accede a la instrucción i en la dirección m.

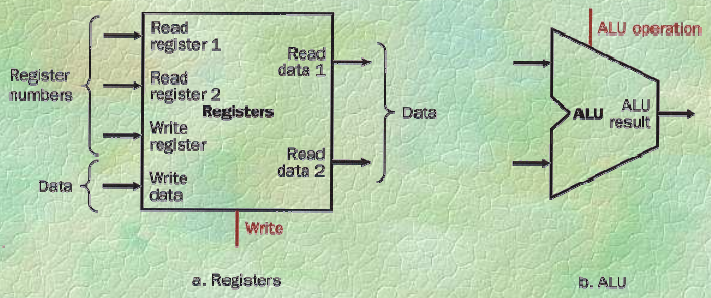
Al final del ciclo (flanco de bajada) PC va a apuntar a la instrucción i+1 en la dirección m+4



4

Componentes de las instrucciones tipo-R

Necesarios para las etapas de
- Extracción de operandos (ID)
- Ejecución (EX)
- Depósito de resultado (WB)

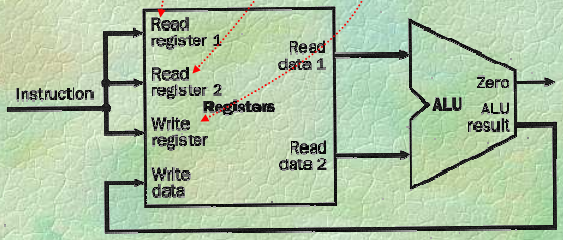


5

Subcircuito para las instrucciones tipo-R

La instrucción provee los 3 campos de dirección de registros:

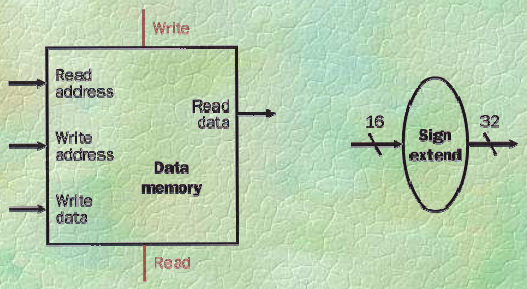
Rs, Rt y Rd



6

Componentes necesarios para las instrucciones tipo-I

En particular las instrucciones de acceso a memoria: LW y SW. Proveen un campo inmediato de 16 bits que debe ser convertido a 32 bits.



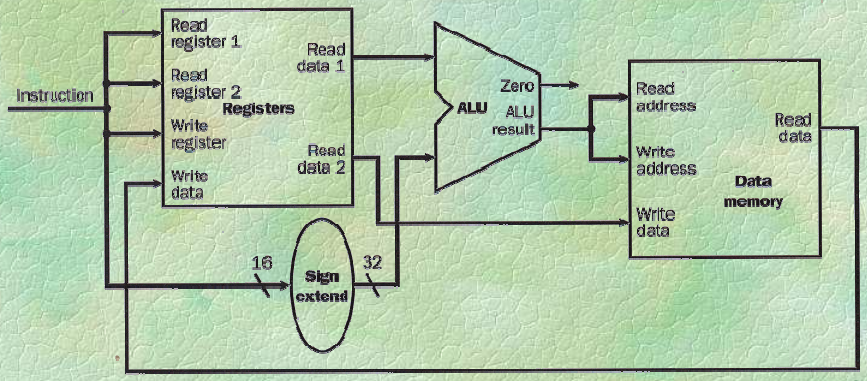
a. Data memory unit b. Sign-extension unit

7

Subcircuito para las instrucciones tipo-I

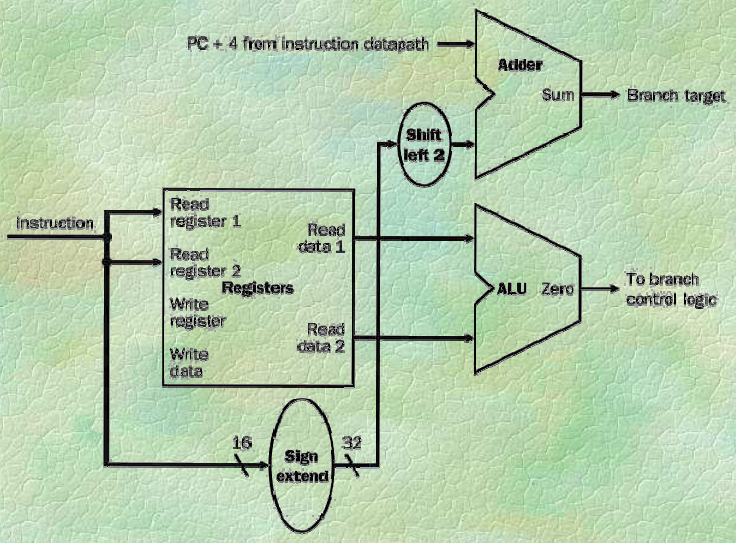
La ALU se utiliza para el cálculo de la dirección efectiva

Rs + sgn_ext (inmediato)



8

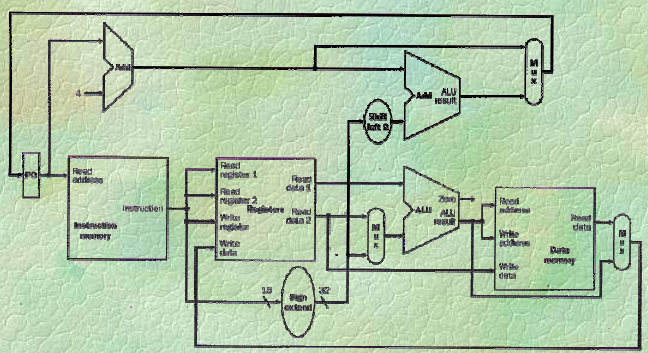
Subcircuito para la instrucción de salto: BEQ



9

Combinación de todos los circuitos

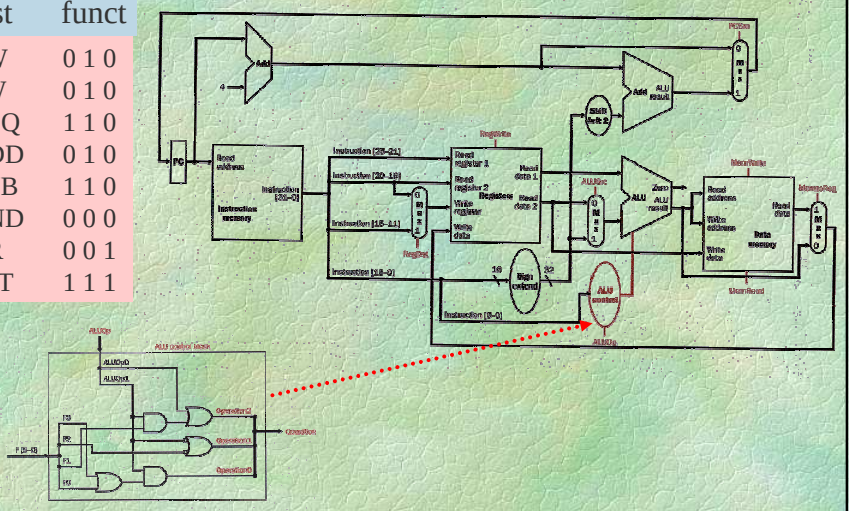
- Ningún elemento puede ser usado mas de una vez y cualquier elemento que se necesite usar mas de una vez será duplicado.
- La memoria de instrucciones está separada de la de datos.
- Algunos componentes pueden ser compartidos mediante la inclusión de MUX



10

Control de la ALU

Inst	funct
LW	0 1 0
SW	0 1 0
BEQ	1 1 0
ADD	0 1 0
SUB	1 1 0
AND	0 0 0
OR	0 0 1
SLT	1 1 1



11

Análisis del formato de instrucción

R	0	Rs	Rt	Rd	Shamt	Funct
	31-26	25-21	20-16	15-11	10-6	5-0
I	35 ó 43	Rs	Rt	A D D R E S S		
	31-26	25-21	20-16	15-0		
beq	4	Rs	Rt	A D D R E S S		
	31-26	25-21	20-16	15-0		

OBSERVACIONES

- El campo OP siempre esta en los bits 31 a 26 (OP[5-0]).
- Los 2 reg. a ser leidos siempre se referencian con Rs y Rt, en posiciones 25-21 y 20-16.
- El reg. base para las instrucciones LOAD y STORE siempre está en las posiciones 25-21.
- El campo ADDRESS siempre está en las posiciones 15-0.
- El registro destino siempre está en uno de dos lugares:
  - para LOAD siempre está en Rt (20-16).
  - para tipo-R siempre está en Rd (15-11).

Esto obliga a intercalar un multiplexor para seleccionar que campo de la instrucción es usado para indicar el número de registro a ser escrito

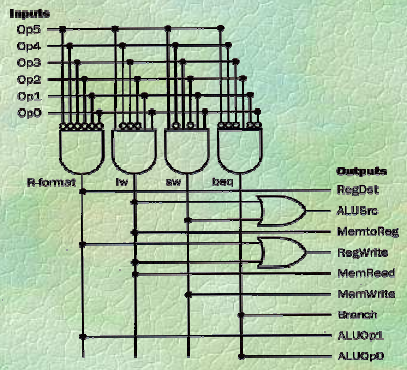
12

Función de las señales de control

señal	efecto a 0	efecto a 1
MemRead	nada	El contenido de la memoria en la dirección dada es copiado en la salida de datos
MemWrite	nada	El contenido de la memoria en la dirección dada es escrito por el dato de entrada
ALUSrc	El segundo operando de la ALU viene del segundo registro del banco de registros	El segundo operando de la ALU viene de los 16 bits bajos de la instrucción extendidos en signo
RegDst	El número de registro de destino proviene del campo Rt	El número de registro de destino proviene del campo Rd
RegWrite	nada	Habilitación de escritura del registro de escritura
PCSrc	El PC es reemplazado por la salida del sumador que computa PC+4	El PC es reemplazado por la salida del sumador que computa la dirección destino
MemtoReg	EL valor a guardar en el registro de destino proviene de la ALU	EL valor a guardar en el registro de destino proviene de la memoria de datos

13

instrucción	RegDst	ALU Src	Memto Reg	Reg Write	Mem Read	Mem Write	Branch	ALU OP1	ALU OP0
R-type	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

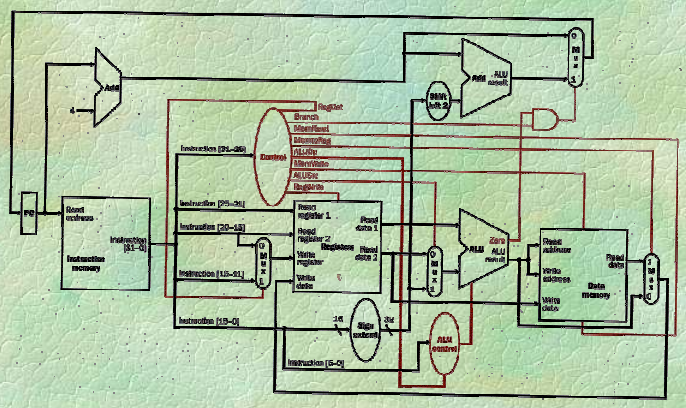


14

Unidad de control General

Las señales generadas por la unidad de control actúan sobre:

- Multiplexores
- habilitación de escritura del banco de reg. y memoria de datos
- Operación de la ALU



15

Análisis de rendimiento de la implementación

Considérese la tabla de tiempos siguiente (en nanosegundos):

	IF	ID	EX	MEM	WB	Total
Lw	10	5	7	10	5	37
Sw	10	5	7	10	--	32
Add	10	5	7	--	5	27
Sub	10	5	7	--	5	27
And	10	5	7	--	5	27
Or	10	5	7	--	5	27
Slt	10	5	7	--	5	27
Beq	10	5	7	--	--	22
J	10	5	7	--	--	22

16

Análisis de rendimiento de la implementación

Observación: Por ser de ciclo único, todas las instrucciones tardan lo mismo: **37 ns.**

CPI estático:

(37+37+37+37+37+37+37+37+37) / 9 = **37 ns.**

CPI dinámico:

(3 * 37ns + 1000 * [5 * 37ns]) / 5003 = **37ns**

- Se calcula respecto de código
- En el ejemplo se ejecutan 5003 inst.
- El ejemplo itera 1000 veces

```
        ADDI    $2, $0, 1
        ADDI    $5, $0, 5
        ADDI    $10, $0, 4004
Ciclo:  LW      $1, base_B($2)
        MUL     $1, $1, $5
        SW      $1, base_A($2)
        ADDI    $2, $2, 4
        BNE     $2, $10, ciclo
```

Inconvenientes de la implementación de 1 ciclo

- Se debe respetar el tiempo de la instrucción que mas tarde para fijar el ciclo de reloj
- Desperdicio de tiempo para instrucciones que tarden menos
- En implementaciones con cientos de instrucciones: demoras grandes
- Duplicación de componentes de igual o similar funcionalidad
- Imposibilidad de reusar componentes inactivos en un ciclo

18