TEC | Tecnológico de Costa Rica

CPI y Flujo de Datos Lección 7

CE-4301 ARQUITECTURA DE COMPUTADORES I

ÁREA DE INGENIERÍA EN COMPUTADORES

INSTITUTO TECNOLÓGICO DE COSTA RICA

TEC | Tecnológico de Costa Rica

Agenda

1. Rendimiento de un procesador
   - CPI
2. Flujo de datos - MIPS

TEC | Tecnológico de Costa Rica

Rendimiento de un CPU

En un CPU la métrica para medir el rendimiento es el tiempo de ejecución.

$$\text{CPU execution time for a program} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

o

$$\text{CPU execution time for a program} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

TEC | Tecnológico de Costa Rica

Ejemplo

Suponga que un programa corre en 10 segundos en una computadora A, la cuál tiene una frecuencia de reloj de 2 GHz. Usted como Ing. en computadores está diseñando una computadora B que sea capaz de correr el programa en 6 segundos. Usted notó que puede aumentar la frecuencia de reloj sustancialmente, pero con cambios en el hardware que hacen que la computadora B requiera 1.2 veces la cantidad de ciclos de reloj que la computadora A. ¿Qué frecuencia debe tener la computadora B para que el programa corra en 6 segundos?

TEC | Tecnológico de Costa Rica

Ejemplo

Suponga que un programa corre en 10 segundos en una computadora A, la cuál tiene una frecuencia de reloj de 2 GHz. Usted como Ing. en computadores está diseñando una computadora B que sea capaz de correr el programa en 6 segundos. Usted notó que puede aumentar la frecuencia de reloj sustancialmente, pero con cambios en el hardware que hacen que la computadora B requiera 1.2 veces la cantidad de ciclos de reloj que la computadora A. ¿Qué frecuencia debe tener la computadora B para que el programa corra en 6 segundos?

R/ B debe correr al doble de frencuencia que A.

TEC | Tecnológico de Costa Rica

Rendimiento de las instrucciones

La ecuación anterior no mostró cómo afecta el número de instrucciones el rendimiento

$$\text{CPU clock cycles} = \text{Instructions for a program} \times \text{Average clock cycles per instruction}$$

Ciclos por Instrucción: **CPI** * Varia según la instrucción

TEC | Tecnológico de Costa Rica

Ecuación clásica para rendimiento CPU

La ecuación clásica en términos de: número de instrucciones, CPI y tiempo de ciclo de reloj.

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

o

$$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$

TEC | Tecnológico de Costa Rica

Ejemplo

Suponga que tiene 2 implementaciones del mismo ISA. La computadora A tiene un tiempo de ciclo de 250 ps y un CPI de 2.0 para determinado programa. La computadora B tiene un tiempo de ciclo de 500 ps y un CPI de 1.2 para el mismo programa. ¿Cuál computadora es más rápida y por cuánto?

TEC | Tecnológico de Costa Rica

Ejemplo

Suponga que tiene 2 implementaciones del mismo ISA. La computadora A tiene un tiempo de ciclo de 250 ps y un CPI de 2.0 para determinado programa. La computadora B tiene un tiempo de ciclo de 500 ps y un CPI de 1.2 para el mismo programa. ¿Cuál computadora es más rápida y por cuánto?

R/ A es 1.2 veces más rápida que B.

Componentes básicos para medir rendimiento

TEC | Tecnológico de Costa Rica

Components of performance	Units of measure
CPU execution time for a program	Seconds for the program
Instruction count	Instructions executed for the program
Clock cycles per instruction (CPI)	Average number of clock cycles per instruction
Clock cycle time	Seconds per clock cycle

FIGURE 1.14 The basic components of performance and how each is measured.

TEC | Tecnológico de Costa Rica

Procesador

El rendimiento del procesador esta definido por 3 factores:

1. Número de instrucciones (Determinado por el compilador y el ISA)
2. Tiempo del ciclo de reloj (Determinado por la implementación -> microarquitectura)
3. CPI - Ciclos de reloj por instrucción (Determinado por la implementación -> microarquitectura)

TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

Implementación con un subconjunto de instrucciones MIPS:

1. Load word (lw), Store word (sw)
2. ADD, SUB, AND, OR
3. BEQ

TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

Los principios de diseño se ven reflejados en la implementación

1. Simplicity favors regularity
2. Smaller is faster
3. Good design demads compromise
4. Make the common case fast

TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

El inicio de todas las instrucciones es el mismo:

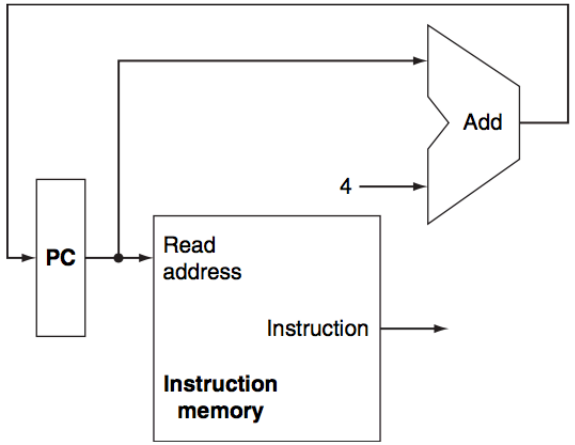
1. Se comienza por leer la instrucción de memoria.
2. Se prepara para leer la siguiente instrucción (PC + 4).

TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

El inicio de todas las instrucciones es el mismo:

1. Se comienza por leer la instrucción de memoria.
2. Se prepara para leer la siguiente instrucción (PC + 4).



TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

Los pasos siguientes dependen de cada tipo de instrucción:

Name	Fields						Comments
Field size	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	All MIPS instructions are 32 bits long
R-format	op	rs	rt	rd	shamt	funct	Arithmetic instruction format
I-format	op	rs	rt	address/immediate			Transfer, branch, imm. format
J-format	op	target address					Jump instruction format

FIGURE 2.20 MIPS instruction formats.

TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

Instrucciones **tipo R**:

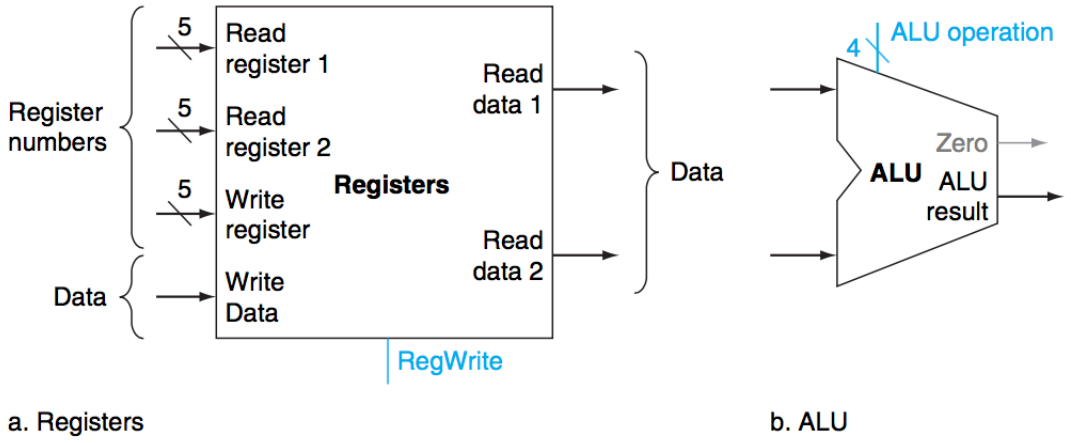
1. Leen dos operandos
2. Ejecutan una operación ALU sobre **registros**
3. Escribe el resultado en un **registro**

TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

Instrucciones **tipo R**:

1. Leen dos operandos
2. Ejecutan una operación ALU sobre **registros**
3. Escribe el resultado en un **registro**



TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

Instrucciones **load/store**:

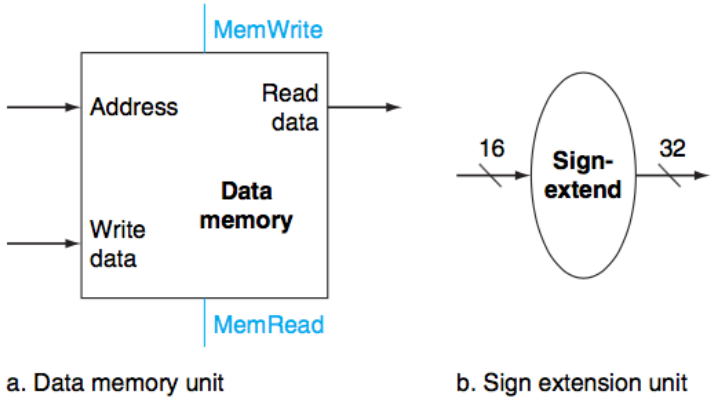
1. Indican una dirección de memoria: Base + Offset (signed 16 bits)
2. **lw** $t1,offset_value($t2) / **sw** $t1,offset_value ($t2)

TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

Instrucciones **load/store**:

1. Indican una dirección de memoria: Base + Offset (signed 16 bits)
2. **lw** $t1,offset_value($t2) / **sw** $t1,offset_value ($t2)



a. Data memory unit

b. Sign extension unit

TEC | Tecnológico de Costa Rica

Flujo de datos (MIPS)

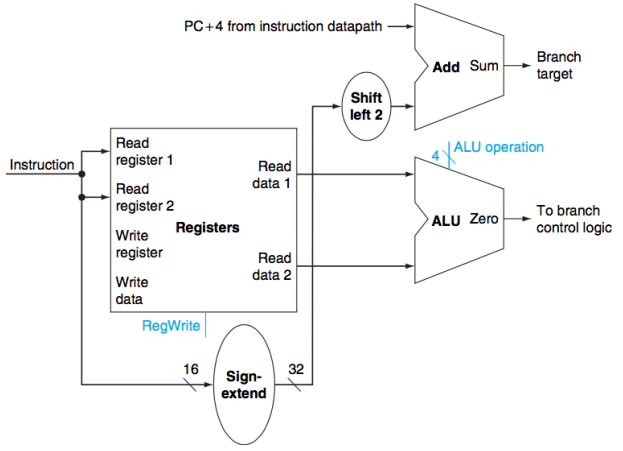
Instrucciones **BEQ**:

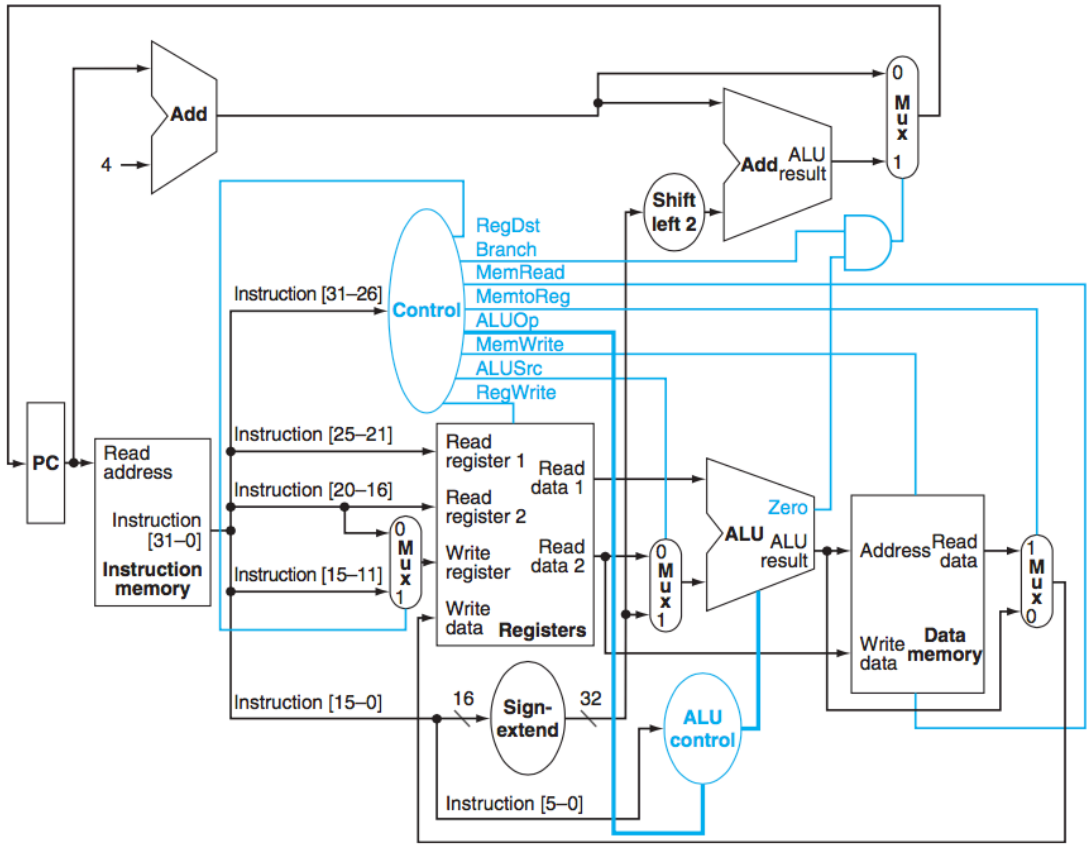
1. Dos operaciones: calcula la dirección en memoria y compara el contenido de dos registros.
2. La dirección de salto se calcula: (PC + 4) + (offset << 2) (signed 16 bits).
3. **beq** $t1,$t2,offset

Flujo de datos (MIPS)

Instrucciones **BEQ**:

1. Dos operaciones: calcula la dirección en memoria y compara el contenido de dos registros.
2. La dirección de salto se calcula: (PC + 4) + (offset << 2) (signed 16 bits).
3. **beq** $t1,$t2,offset




Add
4
Add
ALU result
0
M
u
x
1
Shift left 2
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
Sign-extend
32
ALU
Zero
ALU result
ALU control
Address
Read data
Write data
Data memory

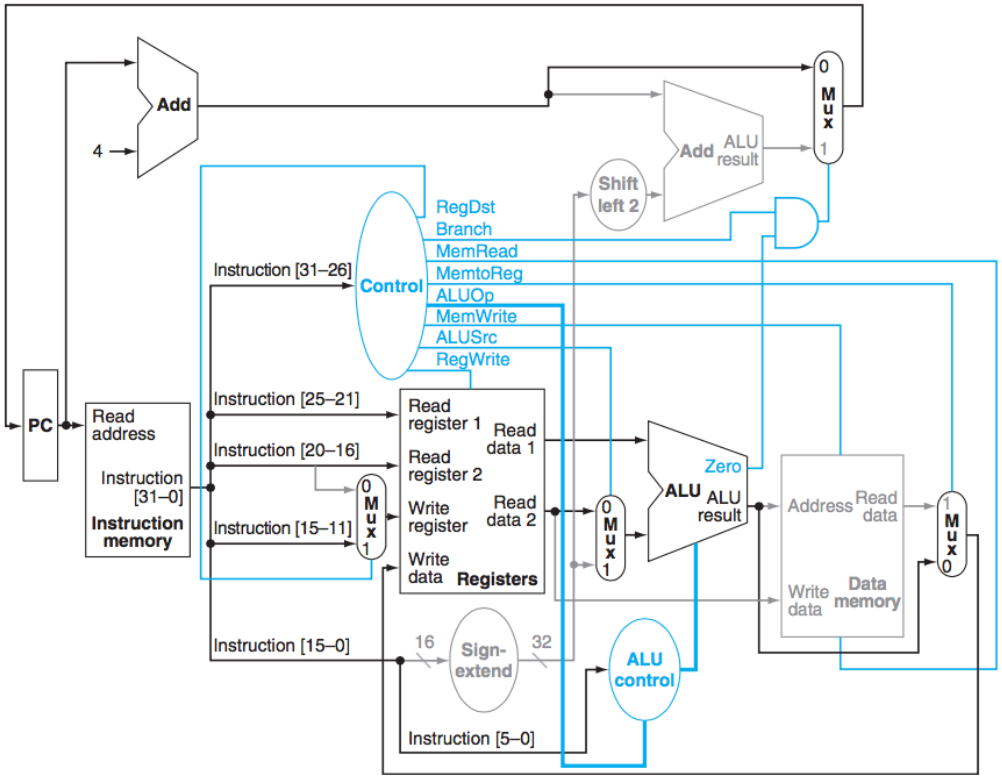


FIGURE 4.19 The datapath in operation for an R-type instruction, such as `add $t1,$t2,$t3`**.** The control lines, datapath units, and connections that are active are highlighted.

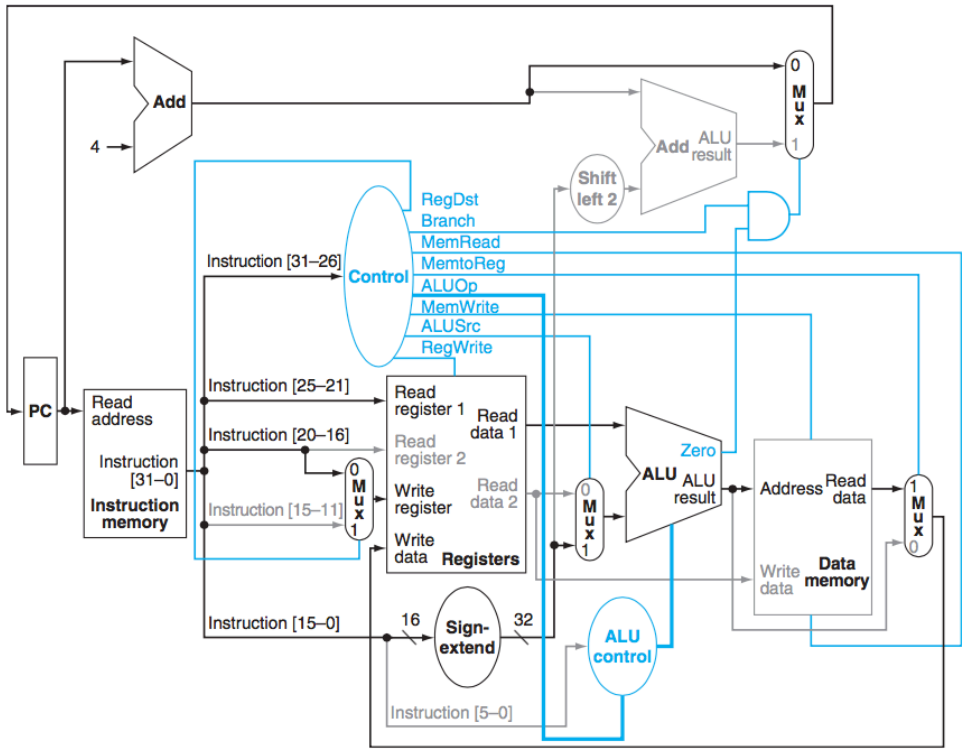


FIGURE 4.20 The datapath in operation for a load instruction. The control lines, datapath units, and connections that are active are highlighted. A store instruction would operate very similarly. The main difference would be that the memory control would indicate a write rather than a read, the second register value read would be used for the data to store, and the operation of writing the data memory value to the register file would not occur.

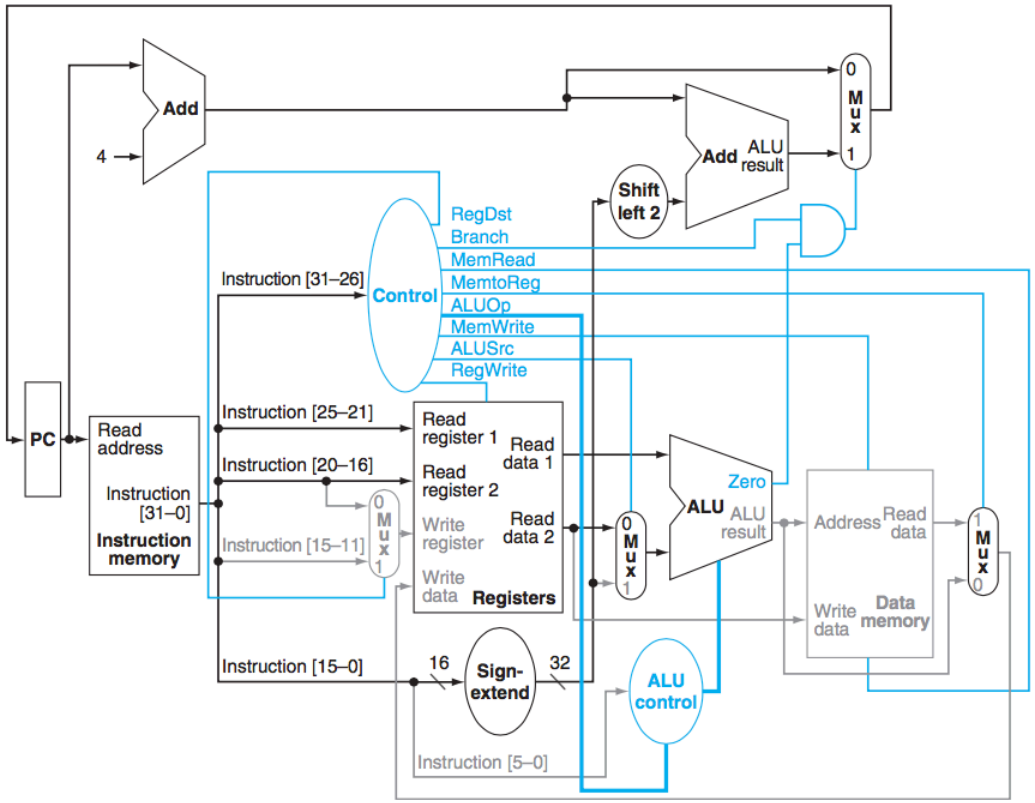


FIGURE 4.21 The datapath in operation for a branch-on-equal instruction. The control lines, datapath units, and connections that are active are highlighted. After using the register file and ALU to perform the compare, the Zero output is used to select the next program counter from between the two candidates.

TEC | Tecnológico de Costa Rica

¿Por qué ya no se usa implementación uniciclo?

TEC | Tecnológico de Costa Rica

¿Por qué ya no se usa implementación uniciclo?

1. CPI = 1
2. El tiempo de ciclo lo determina la instrucción más lenta
3. No se puede mejorar la implementación del caso más común

TEC | Tecnológico de Costa Rica

Referencias

David A. Patterson & John L. Hennessy (2004) 4th Ed.

Computer Organization and Design. The hardware / software interface.