

Estructura condicional

Organización de computadoras

Universidad Nacional de Quilmes

May 27, 2015

¿Que vimos?

1 Punto Flotante

- 1 Interpretar
- 2 Rango
- 3 Resolución

2 IEEE 754

- 1 Interpretar
- 2 Clases

EJERCICIO: Interpreta en IEEE las siguientes cadenas:

- 1 003F0010
- 2 F0C00200
- 3 FF800000
- 4 FF800010

Importante

- 1 Segundo Parcial 25 de Noviembre!!!!
- 2 Atención a la fecha que está cerca, no dejemos para lo último la resolución de las prácticas
- 3 Vienen temas que requieren de un tiempo para comprenderlo.

¡Desafío!

Ejecución condicional

Hacer un programa que, si el valor en R0 es igual al valor en R1, ponga en R2 un 1 ó 0 en caso contrario



??

Flags: motivación

¿Cómo determinar si dos números **son iguales**?

¿Cómo expresar **otras condiciones**?

¿Cómo ejecutar una instrucción **dependiendo de una condición**?

Flags: motivación

¿Cómo determinar si dos números son iguales?



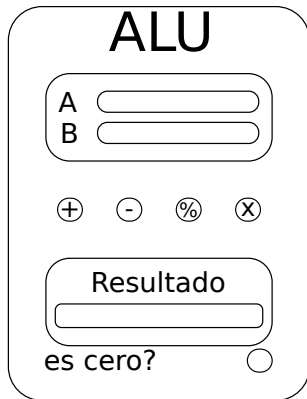
Mediante la resta: $A - B$ si $A = B$ entonces $C = 0$



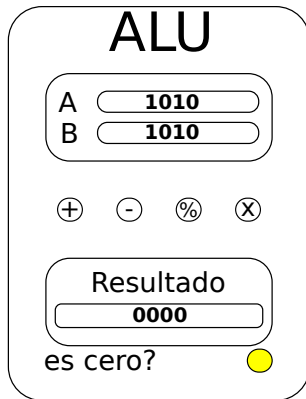
Se necesita una operación de resta (SUB) ✓

Se necesita observar el resultado de la ALU ✗

Flags: motivación



Flags: motivación



Flags: motivación

¿Cómo determinar si un número es negativo?

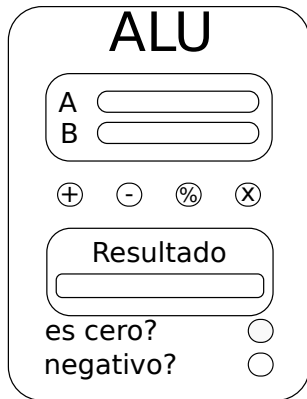


Mirando su primer bit (En ca2)

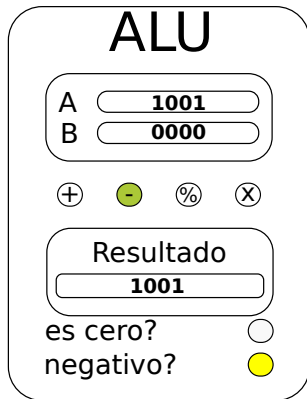


¿Cómo lo hago con lo que ya tengo?

Flags: motivación



Flags: motivación



Ejecución condicional

Ejercicio...

Hacer un programa que ponga en R2 el máximo
valor entre R0 y R1



??

Flags: motivación

¿Cómo determinar si un número es mayor a otro?



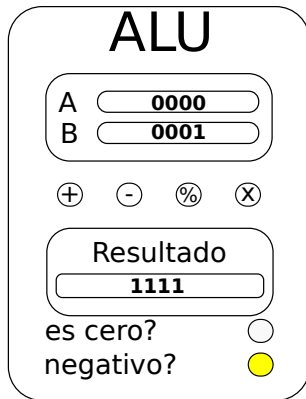
Mediante la resta:

$$A - \frac{B}{C} \quad \text{si } A > B \text{ entonces } C > 0$$



¿Cómo lo hago con lo que ya tengo?

Flags: motivación



Flags

Flags

- Son bits que se usan para caracterizar el resultado de la ALU
- La ALU los calcula cuando lleva a cabo una operación aritmética
- Cada uno indica una condición distinta
- La arquitectura provee instrucciones para **conocer su valor y actuar en consecuencia.**

Flags: Z, N, C y V

Son 4

- Z (Zero)
- N (Negative)
- C (Carry)
- V (Overflow)

Flag Z (Zero)

Z=1 cuando todos los bits del resultado son 0



En una resta:

$$\begin{array}{r} 111 \\ - 111 \\ \hline 000 \end{array} \quad Z=1$$

En una suma:

$$\begin{array}{r} 111 \\ + 001 \\ \hline 000 \end{array} \quad Z=1$$

Flag Z (Zero)

¿Cuanto vale Z?

$$\begin{array}{r} 101 \\ - 001 \\ \hline 100 \end{array} \quad Z=0$$

Flag N (Negative)

N=1 cuando el primer bit del resultado es 1



En una resta:
$$\begin{array}{r} 100 \\ - 001 \\ \hline 011 \end{array} \quad N=0$$

En una suma:
$$\begin{array}{r} 101 \\ + 001 \\ \hline 110 \end{array} \quad N=1$$

Flag N (Negative)

¿Cuanto vale N?

$$\begin{array}{r} 011 \\ + 011 \\ \hline 110 \end{array} \quad N=1$$

Flag C (Carry)

C=1 cuando luego al terminar la suma hay un acarreo (o al terminar la resta hay un préstamo)



En una resta:

$$\begin{array}{r} 100 \\ - 001 \\ \hline 011 \end{array} \quad C=0$$

En una suma:

$$\begin{array}{r} 111 \\ + 001 \\ \hline 000 \end{array} \quad C=1$$

Flag C (Carry)

¿Cuanto vale C?

$$\begin{array}{r} 011 \\ + 011 \\ \hline 110 \end{array} \quad C=0$$

$$\begin{array}{r} 011 \\ - 101 \\ \hline 110 \end{array} \quad C=1$$

Flag C (Carry)

¿Que significa C en *BSS()*?

$$\begin{array}{r} 011 \dots 3 \\ + 011 \dots 3 \\ \hline 110 \dots 6 \end{array} \quad C=0 \quad \checkmark$$

$$\begin{array}{r} 101 \dots 5 \\ + 101 \dots 5 \\ \hline 010 \dots ? \end{array}$$

C=1 ✗

$$\begin{array}{r} 011 \dots 3 \\ - 101 \dots 5 \\ \hline 110 \dots ? \end{array}$$

C=1 ✗

Flag C (Carry)

¿Que significa C en *BSS()*?



El resultado no se puede representar

$$\begin{array}{r}
 101 \dots 5 \\
 + 101 \dots 5 \\
 \hline
 \text{¿10?}
 \end{array}
 \quad
 \begin{array}{r}
 011 \dots 3 \\
 - 101 \dots 5 \\
 \hline
 \text{¿-2?}
 \end{array}$$



Fuera de rango de *BSS(3)*: $[0,7]$

Flag C (Carry)

¿Que significa C en **CA2()**?

$$\begin{array}{r}
 011 \dots 3 \\
 + 011 \dots 3 \\
 \hline
 110 \dots 2
 \end{array}
 \quad C=0 \quad \times
 \quad
 \begin{array}{r}
 011 \dots 3 \\
 + 101 \dots 3 \\
 \hline
 000 \dots 0
 \end{array}
 \quad C=1 \quad \checkmark$$

$$\begin{array}{r}
 011 \dots 3 \\
 - 101 \dots 3 \\
 \hline
 110 \dots 2
 \end{array}
 \quad C=1 \quad \times
 \quad
 \begin{array}{r}
 111 \dots 1 \\
 - 010 \dots 2 \\
 \hline
 101 \dots 3
 \end{array}
 \quad C=0 \quad \checkmark$$

Flag C (Carry)

¿Que significa C en `CA2()`?



¡Nada!

Flag V (Overflow)

V=1 cuando, en CA2() el resultado no se puede representar



$$(1) \quad + \frac{\text{positivo}}{\text{positivo}} \quad \times$$

$$(2) \quad + \frac{\text{negativo}}{\text{positivo}} \quad \times$$

$$(3) \quad - \frac{\text{negativo}}{\text{positivo}} \quad \times$$

$$(4) \quad - \frac{\text{positivo}}{\text{negativo}} \quad \times$$

Flag V en CA2()

Caso 1 + $\frac{\text{positivo}}{\text{positivo}}$
 negativo

$$\begin{array}{r}
 010 \dots 2 \\
 + 010 \dots 2 \\
 \hline
 100 \dots -4
 \end{array}
 , V=1 \quad \times$$

Flag V en CA2()

Caso 2 + $\frac{\text{negativo}}{\text{negativo}}$
 positivo

$$\begin{array}{r}
 100 \dots -4 \\
 + 100 \dots -4 \\
 \hline
 000 \dots 0
 \end{array}
 , V=1 \quad \times$$

Ejercicio

Calcular los flags

- 1 1010+1001
- 2 1111+1111
- 3 1000-0001
- 4 1111+0001

Hasta Aquí

Entonces

¿Cómo determinar si dos números **son iguales**? ✓

¿Cómo determinar si un número **es negativo**? ✓

¿Cómo ejecutar una instrucción **dependiendo de una condición**? ✗

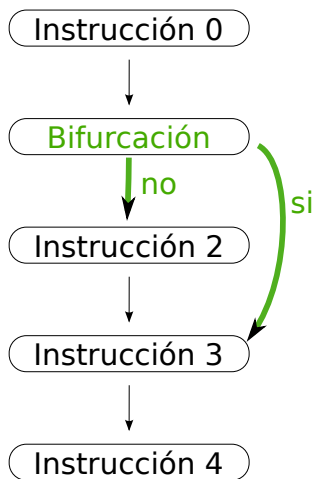
Salto: motivación

¿Cómo ejecutar una instrucción **dependiendo de una condición?**

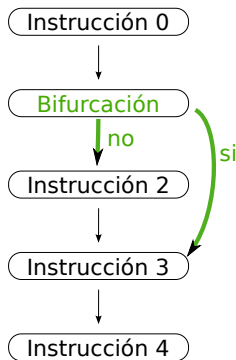


Se necesita una instrucción que observe los resultados de la ALU para **desviar** el programa

Bifurcación: idea



Bifurcación: idea



¿Cómo hacer para que la UC ejecute la instrucción 3 y no la instrucción 2?



¡Alterando el valor de PC!

Salto

Salto: Ejemplo

0000	Instrucción 1
0001	Instrucción 2
0002	Instrucción de salto a 0005
0003	Instrucción 4
0004	Instrucción 5
0005	Instrucción 6
0006	Instrucción 7

PC=0000

Salto: Ejemplo

0000	Instrucción 1
0001	Instrucción 2
0002	Instrucción de salto a 0005
0003	Instrucción 4
0004	Instrucción 5
0005	Instrucción 6
0006	Instrucción 7

PC=0001

IR = Instrucción 1

Salto: Ejemplo

0000	Instrucción 1
0001	Instrucción 2
0002	Instrucción de salto a 0005
0003	Instrucción 4
0004	Instrucción 5
0005	Instrucción 6
0006	Instrucción 7

PC=0002

IR = Instrucción 2

Salto: Ejemplo

0000	Instrucción 1
0001	Instrucción 2
0002	Instrucción de salto a 0005
0003	Instrucción 4
0004	Instrucción 5
0005	Instrucción 6
0006	Instrucción 7

PC=0003

IR = Instrucción de
salto

Salto: Ejemplo

0000	Instrucción 1
0001	Instrucción 2
0002	Instrucción de salto a 0005
0003	Instrucción 4
0004	Instrucción 5
0005	Instrucción 6
0006	Instrucción 7

PC=0005

IR = Instrucción de
salto

Ejecución

Salto: Ejemplo

0000	Instrucción 1
0001	Instrucción 2
0002	Instrucción de salto a 0005
0003	Instrucción 4
0004	Instrucción 5
0005	Instrucción 6
0006	Instrucción 7

PC=0005

IR = Instrucción 6

Ejecución

Absolutos vs relativos

Saltos absolutos y relativos

Salto Relativo

El nuevo valor de PC se expresa en términos de un desplazamiento con respecto a la **siguiente instrucción**

0000	Saltar una celda mas adelante
0001	Instrucción 2
0002	Instrucción 3
0003	Instrucción 4



$$PC \leftarrow PC + 1$$

Salto absolutos y relativos

Salto Absoluto

El nuevo valor de PC se expresa en términos de una dirección de memoria

0000	Saltar a la celda 0003
0001	Instrucción 2
0002	Instrucción 3
0003	Instrucción 4



PC ← 0003

Condiciona l vs Incondiciona l

Salto incondicionales

Salto incondicional

La actualización de PC se lleva a cabo siempre que se ejecute el salto

Salto incondicionales

Ejemplo

0000	SUB R0, R1
0001	<i>salto</i> 0x0002
0002	SUB R1, 0x0001
0003	
0004	ADD R0, 0x000A

Salto incondicionales

Ejemplo

0000	SUB R0, R1
0001	salto 0x0002 ← ¿Qué significa?
0002	SUB R1, 0x0001
0003	
0004	ADD R0, 0x000A

Salto condicional

Salto condicional

La actualización de PC se lleva a cabo si se cumple determinada condición sobre los flags

Salto condicionales

Ejemplo

0000	SUB R0, R1
0001	saltoSiEsCero 0x0002
0002	SUB R1, 0x0001
0003	
0004	ADD R0, 0x000A

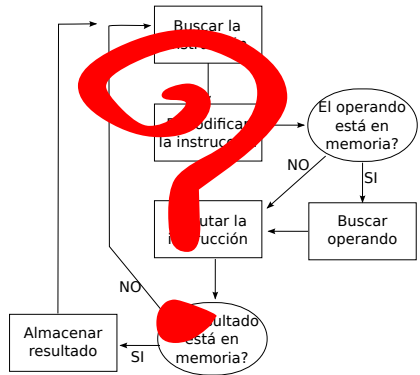
Salto condicionales

Ejemplo

0000	SUB R0, R1
0001	saltoSiEsCero 0x0002 ← ¿Qué significa?
0002	SUB R1, 0x0001
0003	
0004	ADD R0, 0x000A

Salto condicional

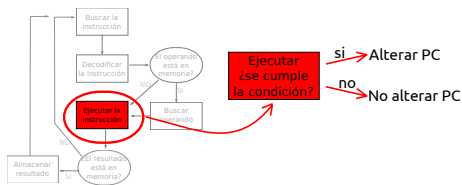
¿Si la condición no se cumple el salto no se ejecuta?



Saltos condicionales

¡No!

La instrucción de salto si se ejecuta, lo que no ocurre es la alteración de PC



Salto

s condicionales e incondicionales

Salto

s condicionales...¿relativos o absolutos?

Salto

s incondicionales...¿relativos o absolutos?

Salto condicionales e incondicionales ¿absolutos o relativos?

Salto condicional absoluto

0000	SUB R0, R1
0001	saltoSiEsCero 0x0002
0002	SUB R1, 0x0001
0003	
0004	ADD R0, 0x000A

Salto condicional relativo

0000	SUB R0, R1
0001	saltoSiEsCero 0x0002
0002	SUB R1, 0x0001
0003	
0004	ADD R0, 0x000A

Etiquetas

Etiqueta

Pseudo-instrucción que permite evitar el cálculo de la dirección (o desplazamiento del salto)

Etiquetas

Ejemplo de uso

```
MOV R0, 0x0001
```

```
SUB R1, 0x0001
```

```
salto-si-es-cero meDioCero
```

```
SUB R1 0x0001
```

```
meDioCero ADD R0, 0x000A
```

Arquitecturas Q

... La saga estaba incompleta

...

Arquitectura Q4

Arquitectura Q4

- Tiene 8 registros de uso general de 16 bits: R0..R7
- Tiene direcciones de 16 bits
- Los operandos pueden estar en registros, ser constantes o estar en direcciones de memoria
- permite 3 modos de direccionamiento:
 - modo registro: el valor buscado está en un registro
 - modo inmediato: el valor buscado está codificado dentro de la instrucción
 - modo directo: el valor buscado está contenido en una celda de memoria

(Idem Q3)

Arquitectura Q4: formato de instrucciones

- Instrucciones de 2 operandos
(MUL,MOV,ADD,SUB,CMP,DIV)

Cod_Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)
----------------	----------------------	---------------------	---------------------------	--------------------------

- Instrucciones con un operando Origen: CALL, JMP

Cod_Op (4b)	Relleno (000000)	Modo Origen (6b)	Operando Origen (16b)
----------------	---------------------	---------------------	--------------------------

- Instrucciones sin operandos: RET

Cod_Op (4b)	Relleno (00000000000000)
----------------	-----------------------------

- Saltos **condicionales** y relativos

Prefijo (1111)	Cod_Op (4)	Desplazamiento(8) (8b)
-------------------	---------------	---------------------------

Arquitectura Q4: Instrucciones Aritméticas

Cod_Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)
		Operación	CodOp	
		MUL	0000	
		MOV	0001	
		ADD	0010	
		SUB	0011	
		CMP	0110	
		DIV	0111	

Arquitectura Q4: Instrucción CMP

CMP: compare



Calcula los flags de destino-origen
pero no almacena el resultado

Arquitectura Q4: Instrucciones sin operandos

Tipo 5: Instrucciones sin operandos

Operación	CodOp	Efecto
RET	1100	$PC \leftarrow [SP+1]; SP \leftarrow SP + 1$

Cod.Op (4b)	Relleno (000000000000)
----------------	---------------------------

(Idem Q3)

Arquitectura Q4: Instrucciones con un operando Origen

Cod_Op (4b)	Relleno (000000)	Modo Origen (6b)	Operando Origen (16b)
----------------	---------------------	---------------------	--------------------------

Operación	CodOp	Efecto
JMP	1010	PC \leftarrow Origen
CALL	1011	[SP] \leftarrow PC; SP \leftarrow SP - 1; PC \leftarrow Origen

El operando Origen en el JMP es una dirección absoluta

Arquitectura Q4: Saltos condicionales (relativo) - 1 de 2

Prefijo (1111)	Cod_Op (4)	Desplazamiento(8) (8b)
-------------------	---------------	---------------------------

En todos los saltos, el efecto es:

$$PC \leftarrow PC + \text{desplazamiento}$$

Salto	Codop	Descripción	Condición
JE	0001	Igual / Cero	Z
JNE	1001	No igual	\overline{Z}
JLEU	0100	Menor o igual sin signo	$C^{\vee}Z$
JGU	1100	Mayor sin signo	$\overline{(C^{\vee}Z)}$
JCS	0101	Menor sin signo	C
JNEG	0110	Negativo	N

Arquitectura Q4: Saltos condicionales (relativo) - 2 de 2

Prefijo (1111)	Cod_Op (4)	Desplazamiento(8) (8b)
-------------------	---------------	---------------------------

Salto	Codop	Descripción	Condición
JVS	0111	Overflow	V
JLE	0010	Menor o igual con signo	$Z^V(N \oplus V)$
JG	1010	Mayor con signo	$\overline{(Z^V(N \oplus V))}$
JL	0011	Menor con signo	$N \oplus V$
JGE	1011	Mayor o igual con signo	$\overline{(N \oplus V)}$

Arquitectura Q4: Ejercicio

Ensamblar el siguiente programa, ubicándolo a partir de la celda 0FF0

```
MOV R0, R2  
SUB R0,R1  
JE afuera  
ADD [0001], 0x0005  
ADD R1,R2
```

afuera: MOV R2, 0x0005

Arquitectura Q4: Ejercicio

(1)

Hacer un programa que, si el valor en R0 es igual al valor en R1, ponga en R2 un 1 ó 0 en caso contrario

Arquitectura Q4: Ejercicio

(2)

Hacer un programa que, si el valor en R7 es negativo, le sume 1, o le reste 1 en caso contrario

Arquitectura Q4: Ejercicio

(3)

Hacer un programa que ponga en R2 el máximo valor entre R0 y R1. Considerar que los valores están en *BSS()*

Arquitectura Q4: Ejercicio

(4)

Hacer un programa que ponga en R2 el máximo valor entre R0 y R1. Considerar que los valores están en CA2()