

Organización de Computadoras

SEMANA 9

UNIVERSIDAD NACIONAL DE QUILMES

¿ Qué vimos?

- Limitaciones de Q3
- Flags:
 - ¿Qué?
 - ¿Cómo?
 - ¿Para qué?
- Saltos:
 - ¿Qué?
 - Absolutos vs relativos
 - Condicionales vs incondicionales

Hoy!

- Mascaras
- Repeticiones controladas
- Arreglos
- Modo indirecto
- Q5

Operaciones Lógicas

- Se aplican sobre cadenas
- Actúan Bit a Bit
- Ejemplos:
 - AND
 - OR
 - NOT
 - XOR

Operaciones Lógicas

AND

- Realiza el “Y” lógico entre los bits

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

and

$$\begin{array}{r} 1010101010 \\ 1100101100 \\ \hline 1000101000 \end{array}$$

Operaciones Lógicas

OR

- Realiza el “O” lógico entre los bits

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

or

```
1010101010  
1100101100  
            
1110101110
```

Operaciones Lógicas

NOT

- Realiza la negación de los bits

A	NOT A
0	1
1	0

not 1010101010
0101010101

Operaciones Lógicas

XOR

- Realiza el “O exclusivo” lógico entre los bits

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{array}{r} \text{xor} \quad 1010101010 \\ \quad 1100101100 \\ \hline 0110000110 \end{array}$$

Mascaras

- Cadenas binarias que se combinan con otras mediante operaciones lógicas
- Sirven para analizar características de las cadenas

Mascaras

- Con AND:
 - Si queremos preservar el bit original: 1
 - Si queremos dejar un bit en 0: 0
- Con OR:
 - Si nos interesa el bit original: 0
 - Si queremos que quede un 1: 1

Mascaras

Ejemplos

- ¿La cadena en R0 es impar?

```
AND R0, 0x0001  
JNE TerminabaEnCero
```

- Copiar los 3 bits mas significativos de R0 a R1

```
MOV R1, 0xE000  
AND R1, R0
```

Mascaras

Ejemplos

- Permisos de archivos en Linux:
 - 3 bits: leer (r) , escribir (w), ejecutar (x)
 - Usuario, Grupo y Otros
 - 111 111 111 le da permisos a todos para hacer cualquier cosa
 - ¿Cómo saber si otro usuario del grupo del archivo puede escribirlo?

AND 000 010 000

Repeticiones controladas

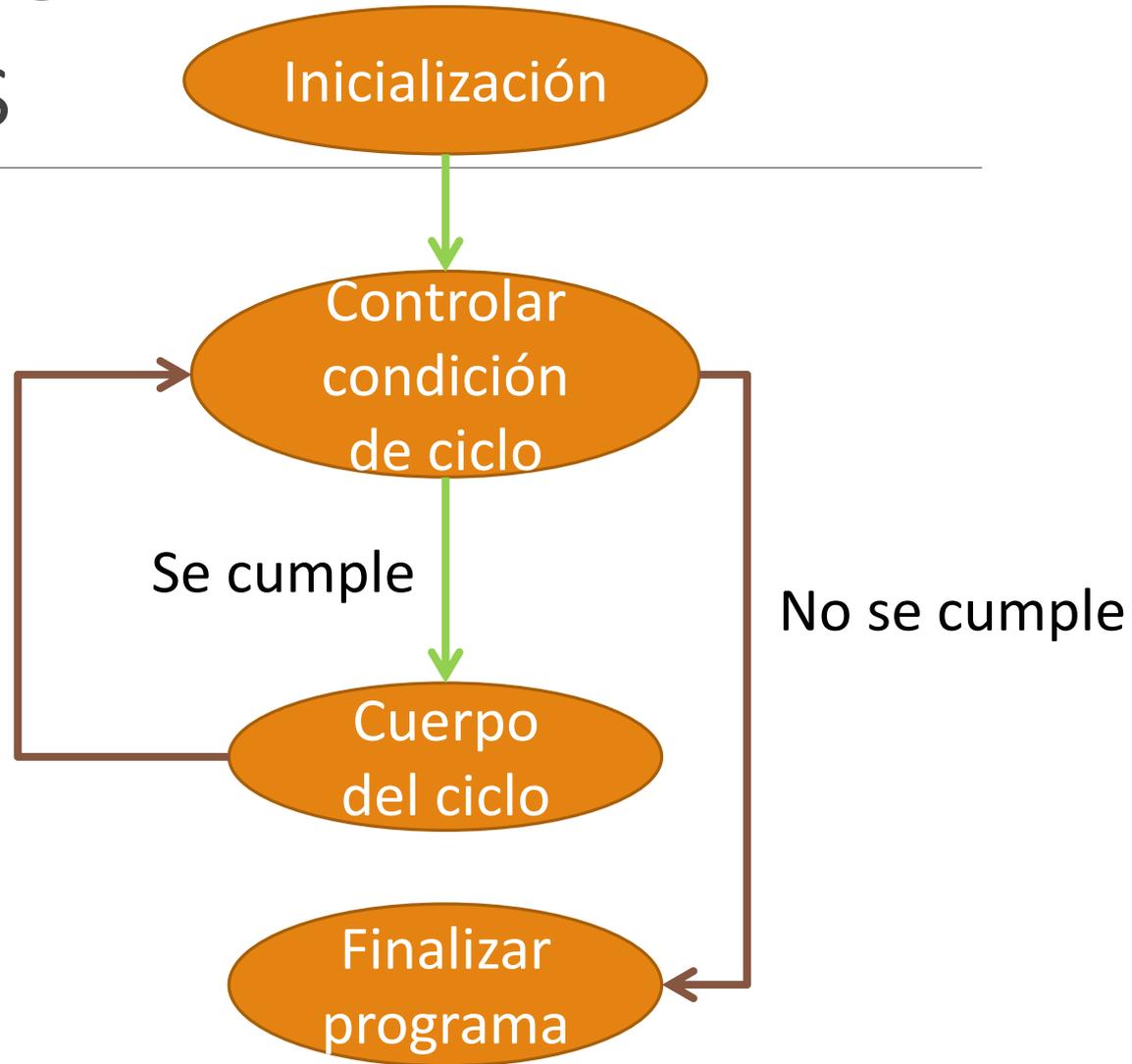
- Los saltos condicionales nos permiten flujos alternativos de ejecución
- Además podemos usarlos para repetir un bloque de código varias veces
- Ejemplo, A está en R0, y B en R1. Hacer $A * B$ sin usar MUL

tot=0

Repetir B veces:

tot = tot + A

Repeticiones controladas



Repeticiones controladas

Ejercicios

- Calcular $A * B$ sin usar MUL
- Calcular N a la M . N en $R0$ y M en $R1$.

Arreglos

- Posiciones de memoria consecutivas
- Colección de datos
- Los elementos no necesariamente ocupan una celda
- El tamaño del arreglo es la cantidad de elementos

Arreglos

Ejemplo

Posición	Contenido
0xF000	0x1020
0xF001	0xDCE3
0xF002	0xE451
0xF003	0x29C8

Un arreglo de 4 números de 16 bits

Un arreglo de 8 números de 8 bits

Un arreglo de 2 números de 32 bits

Arreglos

- Cómo saber donde termina el arreglo?
 - Nos dicen el tamaño
 - Nos dicen que termina con algún valor especial (i.e: 0)

Arreglos

Ejercicio

- A partir de 0x000F hay un arreglo de números. El último de ellos es 0. Hacer un programa que sume todos esos números en R3

Modos de direccionamiento

Indirectos

- Especificamos la celda o el registro en el que se encuentra la dirección del operando

Modo	Código
Inmediato	000000
Registro	100RRR
Directo	001000
Indirecto Registro	
Indirecto	

Modos de direccionamiento

Indirecto Registro

- Se denota $[R_i]$
- Semántica: El valor que está contenido en la celda cuya dirección está en R_i

- $R5 = AA00$

Dirección	Contenido
AA00	AA02
AA01	5647
AA02	4FFA

¿Qué valor queda en R0 si hacemos `MOV R0, [R5]` ?

Modos de direccionamiento

Indirecto

- Se denota `[[Cte]]`
- Semántica: El valor que está contenido en la celda cuya dirección está en contenida en la celda *Cte*

Dirección	Contenido
B100	B102
B101	5647
B102	EDAB

¿Qué valor queda en R0 si hacemos `MOV R0, [[B100]]`?

Modos de direccionamiento

Indirecto

- Requiere 2 accesos a memoria para obtener el operando
- Para guardarlo se necesita 1 solo, porque ya queda calculada la dirección de la celda

Modos de direccionamiento

Ejercicio

Dirección	Contenido
0x0000	0x0007
0x0001	0x0000
0x0002	0x2345
0x0003	0xFEDE
0x0004	0x0002
0x0005	0x0006
0x0006	0x0003
0x0007	0x0001

- R1 = 0x0005
- R2 = 0x0003

- ¿Qué valor queda en R0?
 - MOV R0, [[0x0006]]
 - MOV R0, [[0x0001]]
 - MOV R0, [[0x0004]]
 - MOV R0, [R1]
 - MOV R0, [R2]

Modos de direccionamiento

Ejercicio

- A partir de 0x000F un arreglo de números. El último de ellos es 0. Hacer un programa que sume todos esos números en R3



Q5

Modos de direccionamiento

- Mismos modo de direccionamiento

Modo	Código
Inmediato	000000
Registro	100RRR
Directo	001000
Indirecto	011000
Registro indirecto	110RRR

Q5

Formato de instrucción

- Formato de Instrucción: Tipo 1

Cod Op (4bits)	Modo Destino (6 bits)	Modo origen (6 bits)	Destino (16 bits)	Origen (16 bits)
-------------------	--------------------------	-------------------------	----------------------	---------------------

Operación	Código	Efecto
MUL	0000	Dest \leftarrow Dest * Origen
MOV	0001	Dest \leftarrow Origen
ADD	0010	Dest \leftarrow Dest + Origen
SUB	0011	Dest \leftarrow Dest - Origen
DIV	0111	Dest \leftarrow Dest % Origen
CMP	0110	Modifica los Flags según el resultado de Dest - Origen

Q5

Formato de instrucción

- Formato de Instrucción: Tipo 1

Cod Op (4bits)	Modo Destino (6 bits)	Modo origen (6 bits)	Destino (16 bits)	Origen (16 bits)
--------------------------	---------------------------------	--------------------------------	-----------------------------	----------------------------

Operación	Código	Efecto
AND	0100	Dest \leftarrow Dest \wedge Origen
OR	0101	Dest \leftarrow Dest \vee Origen

Q5

Formato de instrucción

- Formato de Instrucción: Tipo 2 (**1 operando origen**)

Cod Op (4bits)	Relleno (000000)	Modo origen (6 bits)	Origen (16 bits)
-------------------	---------------------	-------------------------	---------------------

Operación	Código	Efecto
CALL	1011	$[SP] \leftarrow PC; SP \leftarrow SP-1;$ $PC \leftarrow \text{Origen}$
JMP	1010	$PC \leftarrow \text{Origen}$

Q5

Formato de instrucción

- Formato de Instrucción: Tipo 2 (**1 operando destino**)

Cod Op (4bits)	Modo destino (6 bits)	Relleno (000000)	destino (16 bits)
-------------------	--------------------------	---------------------	----------------------

Operación	Código	Efecto
NOT	1001	DEST ← not DEST

Q5

Formato de instrucción

- Formato de Instrucción: Tipo 3 (sin operandos)

Cod Op (4bits)	Relleno (000000000000)
-------------------	---------------------------

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

Q5

Formato de instrucción

- Formato de Instrucción: Tipo 4

Prefijo (1111)	Cod Op (4 bits)	Desplazamiento (8 bits)
---------------------------------	----------------------------------	--

Q5

Formato de instrucción – Tipo 4

Salto	Codop	Descripción	Condición
JE	0001	Igual / Cero	Z
JNE	1001	No igual	$\neg Z$
JLE	0010	Menor o igual con signo	$Z + (N \vee V)$
JG	1010	Mayor con signo	$\neg(Z + (N \vee V))$
JL	0011	Menor con signo	$N \vee V$
JGE	1011	Mayor o igual con Signo	$\neg (N \vee V)$
JLEU	0100	Menor o igual sin signo	$C + Z$
JGU	1100	Mayor sin signo	$\neg(C + Z)$
JCS	0101	Menor sin signo	C
JNEG	0110	Negativo	N
JVS	0111	Overflow	V

Q5

Ejercicios

- A partir de la celda B0B0 hay un arreglo con las temperaturas de una cierta localidad, y que finaliza con el primer valor 0. Calcular el promedio. Considerar que 0 no es una temperatura en la localidad.
- Ensamblar:
 - MOV R0, [R1]
 - ADD R0, [[0xF0CA]]
 - SUB [0x1111],][0x1111]]

Q5

Ejercicios

- Completar la cantidad de accesos a memoria en la siguiente tabla:

Instrucción	FI	FO	ST
MOV R0, [R1]			
ADD R0, [[0xF0CA]]			
SUB [0x1111],][0x1111]]			
MUL [[0x0010]], [R5]			
DIV [R1], [[0x43AE]]			
MOV [[0xFEDE]], R1			

Q5

Ejercicios

- Hacer un programa que arme un arreglo a partir de la celda 0099 solo con las posiciones pares de otro arreglo que comienza en 1099 y que finaliza con el valor 0

¿Que pasó hoy?!

- Mascaras
- Repeticiones controladas
- Arreglos
- Modo indirecto
- Q5