

Repasemos

¿Que vimos la clase pasada?

Limitaciones de Q3

Limitaciones de Q3

Si el número que está en R0 es negativo, le sume 1, y si es positivo le reste 1

Flags

Flags

Z

Flags

Z

N

Flags

Z

N

C

Flags

Z

N

C

V

Flags

- Z

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

$$\begin{array}{r} 101 \\ - 101 \\ \hline 000 \end{array}$$

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

$$\begin{array}{r} 101 \\ - 101 \\ \hline 000 \end{array}$$

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

$$\begin{array}{r} - \quad 101 \\ \quad 101 \\ \hline \quad 000 \end{array} \qquad \begin{array}{r} + \quad 110 \\ \quad 011 \\ \hline \quad 001 \end{array}$$

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

$$\begin{array}{r} - \quad 101 \\ \quad 101 \\ \hline \quad 000 \end{array} \quad \begin{array}{r} + \quad 110 \\ \quad 011 \\ \hline \quad 001 \end{array}$$

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline 111 \end{array}$$

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline 111 \end{array}$$

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline 111 \end{array}$$

$$\begin{array}{r} - 100 \\ - 001 \\ \hline 011 \end{array}$$

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline 111 \end{array}$$

$$\begin{array}{r} - 100 \\ - 001 \\ \hline 011 \end{array}$$

Flags

- Z
 - Se enciende si todos los bits del resultado son cero

$$\begin{array}{r} - 101 \\ - 101 \\ \hline \underline{000} \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline \underline{001} \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline \underline{111} \end{array}$$

$$\begin{array}{r} - 100 \\ - 001 \\ \hline \underline{011} \end{array}$$

Flags

- N

Flags

- N
 - El primer bit del resultado vale 1

Flags

- N
 - El primer bit del resultado vale 1

$$\begin{array}{r} 101 \\ - 101 \\ \hline 000 \end{array}$$

Flags

- N
 - El primer bit del resultado vale 1

$$\begin{array}{r} 101 \\ - 101 \\ \hline 000 \end{array}$$

Flags

- N
 - El primer bit del resultado vale 1

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array} \quad \begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

Flags

- N
 - El primer bit del resultado vale 1

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

Flags

- N
 - El primer bit del resultado vale 1

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline 111 \end{array}$$

Flags

- N
 - El primer bit del resultado vale 1

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline 111 \end{array}$$

Flags

- N
 - El primer bit del resultado vale 1

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline 111 \end{array}$$

$$\begin{array}{r} - 100 \\ - 001 \\ \hline 011 \end{array}$$

Flags

- N
 - El primer bit del resultado vale 1

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline 111 \end{array}$$

$$\begin{array}{r} - 100 \\ - 001 \\ \hline 011 \end{array}$$

Flags

- N
 - El primer bit del resultado vale 1

$$\begin{array}{r} - 101 \\ - 101 \\ \hline \underline{000} \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline \underline{001} \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline \underline{111} \end{array}$$

$$\begin{array}{r} - 100 \\ - 001 \\ \hline \underline{011} \end{array}$$

Flags

- C

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

$$\begin{array}{r} 101 \\ - 101 \\ \hline 000 \end{array}$$

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

$$\begin{array}{r} 101 \\ - 101 \\ \hline 000 \end{array}$$

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

$$\begin{array}{r} - \quad 101 \\ \quad 101 \\ \hline \quad 000 \end{array} \qquad \begin{array}{r} + \quad 110 \\ \quad 011 \\ \hline \quad 001 \end{array}$$

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

$$\begin{array}{r} - 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + 110 \\ + 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + 101 \\ + 010 \\ \hline 111 \end{array}$$

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

$$\begin{array}{r} - \quad 101 \\ \quad 101 \\ \hline \quad 000 \end{array}$$

$$\begin{array}{r} + \quad 110 \\ \quad 011 \\ \hline \quad 001 \end{array}$$

$$\begin{array}{r} + \quad 101 \\ \quad 010 \\ \hline \quad 111 \end{array}$$

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

$$\begin{array}{r} - \quad 101 \\ - \quad 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + \quad 110 \\ + \quad 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + \quad 101 \\ + \quad 010 \\ \hline 111 \end{array}$$

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

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

$$\begin{array}{r} - \quad 101 \\ - \quad 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + \quad 110 \\ + \quad 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + \quad 101 \\ + \quad 010 \\ \hline 111 \end{array}$$

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

Flags

- C
 - El bit más significativo pidió uno o bien se llevó uno

$$\begin{array}{r} - \ 101 \\ - \ 101 \\ \hline \textcolor{red}{000} \end{array}$$

$$\begin{array}{r} + \ 110 \\ + \ 011 \\ \hline \textcolor{green}{000} \end{array}$$

$$\begin{array}{r} + \ 101 \\ + \ 010 \\ \hline \textcolor{red}{111} \end{array}$$

$$\begin{array}{r} - \ 100 \\ - \ 001 \\ \hline \textcolor{red}{011} \end{array}$$

Flags

- V

Flags

- V
 - Positivo + Positivo =

Flags

- V
 - Positivo + Positivo = Negativo

Flags

- V
 - Positivo + Positivo = Negativo
 - Negativo + Negativo =

Flags

- V

- Positivo + Positivo = Negativo
- Negativo + Negativo = Positivo

Flags

- V

- Positivo + Positivo = Negativo
- Negativo + Negativo = Positivo
- Negativo - Positivo =

Flags

- V

- Positivo + Positivo = Negativo
- Negativo + Negativo = Positivo
- Negativo - Positivo = Positivo

Flags

- V
 - Positivo + Positivo = Negativo
 - Negativo + Negativo = Positivo
 - Negativo - Positivo = Positivo
 - Positivo - Negativo =

Flags

- V
 - Positivo + Positivo = Negativo
 - Negativo + Negativo = Positivo
 - Negativo - Positivo = Positivo
 - Positivo - Negativo = Negativo

Flags

- V

Flags

- V
 - No se puede representar en CA2

Flags

- V
 - No se puede representar en CA2

$$\begin{array}{r} 101 \\ - 101 \\ \hline 000 \end{array}$$

Flags

- V
 - No se puede representar en CA2

$$\begin{array}{r} 101 \\ - 101 \\ \hline 000 \end{array}$$

Flags

- V
 - No se puede representar en CA2

$$\begin{array}{r} - \quad 101 \\ \quad 101 \\ \hline \quad 000 \end{array} \qquad \begin{array}{r} + \quad 110 \\ \quad 011 \\ \hline \quad 001 \end{array}$$

Flags

- V
 - No se puede representar en CA2

$$\begin{array}{r} 101 \\ - 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} 110 \\ + 011 \\ \hline 001 \end{array}$$

Flags

- V
 - No se puede representar en CA2

$$\begin{array}{r} - \quad 101 \\ \quad 101 \\ \hline \quad 000 \end{array}$$

$$\begin{array}{r} + \quad 110 \\ \quad 011 \\ \hline \quad 001 \end{array}$$

$$\begin{array}{r} + \quad 101 \\ \quad 010 \\ \hline \quad 111 \end{array}$$

Flags

- V
 - No se puede representar en CA2

$$\begin{array}{r} - \quad 101 \\ \quad 101 \\ \hline \quad 000 \end{array}$$

$$\begin{array}{r} + \quad 110 \\ \quad 011 \\ \hline \quad 001 \end{array}$$

$$\begin{array}{r} + \quad 101 \\ \quad 010 \\ \hline \quad 111 \end{array}$$

Flags

- V
 - No se puede representar en CA2

$$\begin{array}{r} - \quad 101 \\ \quad 101 \\ \hline \quad 000 \end{array}$$

$$\begin{array}{r} + \quad 110 \\ \quad 011 \\ \hline \quad 001 \end{array}$$

$$\begin{array}{r} + \quad 101 \\ \quad 010 \\ \hline \quad 111 \end{array}$$

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

Flags

- V
 - No se puede representar en CA2

$$\begin{array}{r} - \quad 101 \\ - \quad 101 \\ \hline 000 \end{array}$$

$$\begin{array}{r} + \quad 110 \\ + \quad 011 \\ \hline 001 \end{array}$$

$$\begin{array}{r} + \quad 101 \\ + \quad 010 \\ \hline 111 \end{array}$$

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

Flags

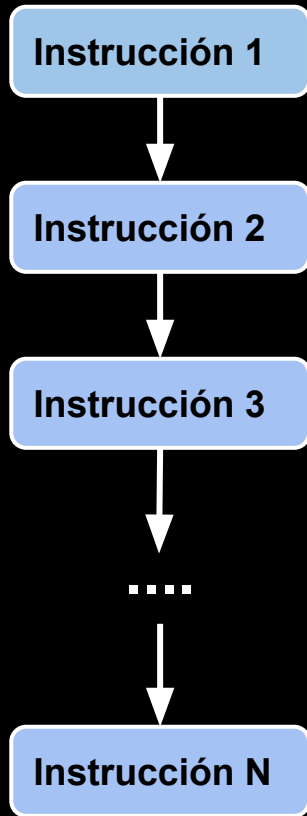
- V
 - No se puede representar en CA2

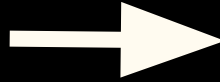
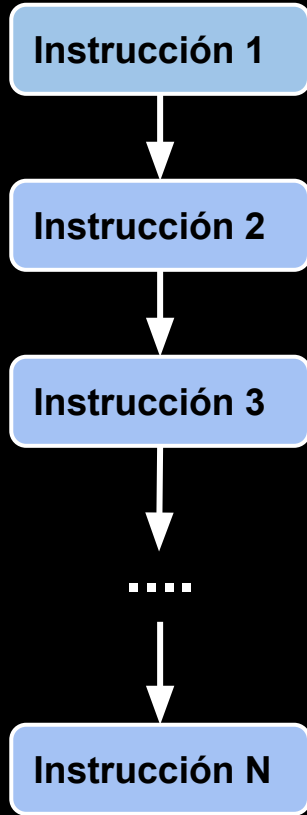
$$\begin{array}{r} - \quad 101 \\ - \quad 101 \\ \hline \quad \underline{000} \end{array}$$

$$\begin{array}{r} + \quad 110 \\ + \quad 011 \\ \hline \quad \underline{001} \end{array}$$

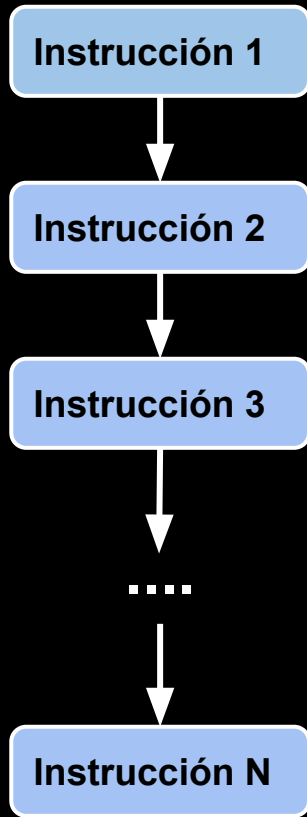
$$\begin{array}{r} + \quad 101 \\ + \quad 010 \\ \hline \quad \underline{111} \end{array}$$

$$\begin{array}{r} - \quad 100 \\ - \quad 001 \\ \hline \quad \underline{011} \end{array}$$



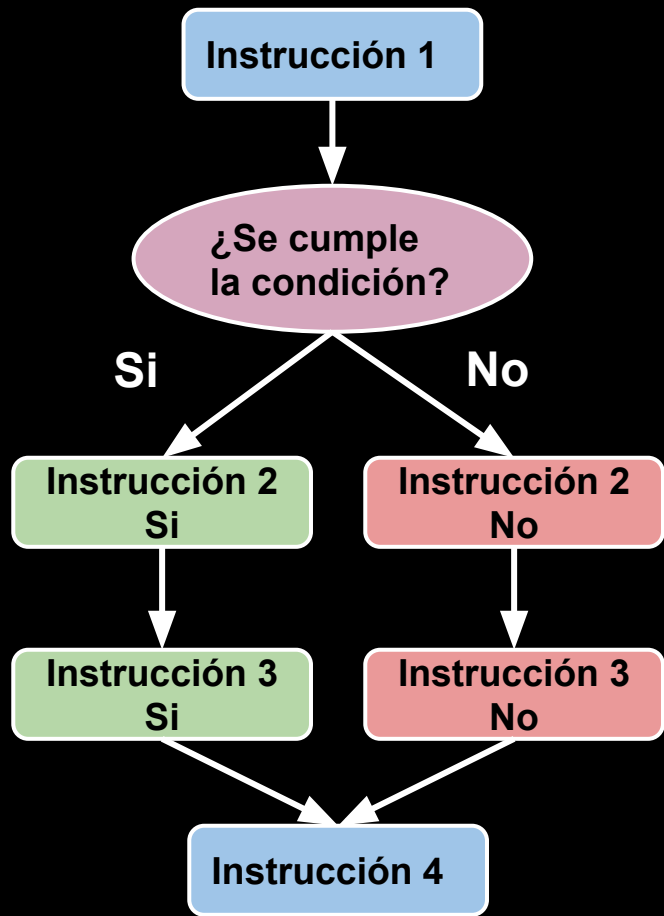


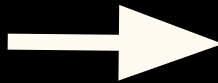
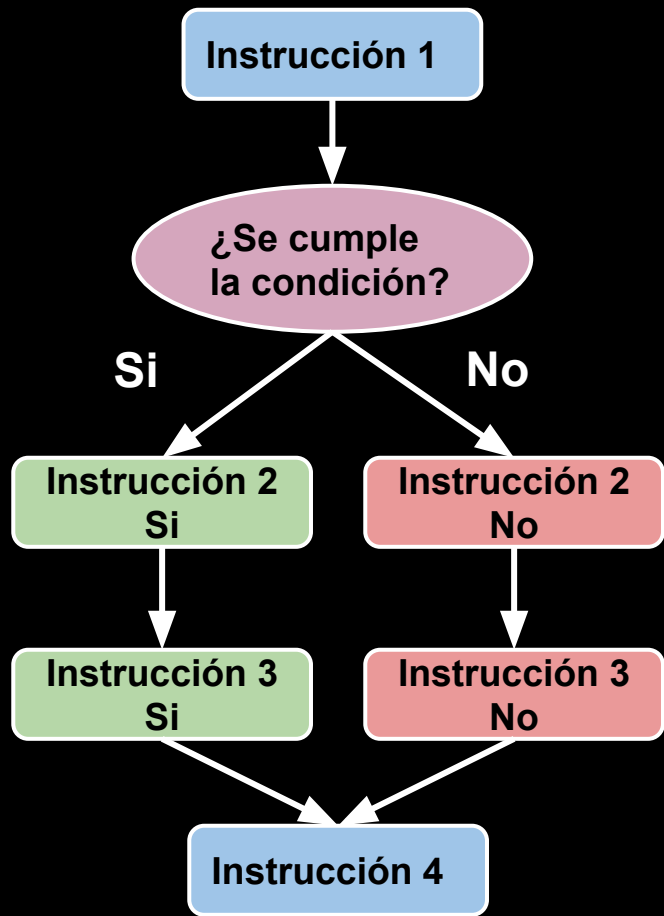
Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	XXXX
0x0002	XXXX
....
0x000F	XXXX



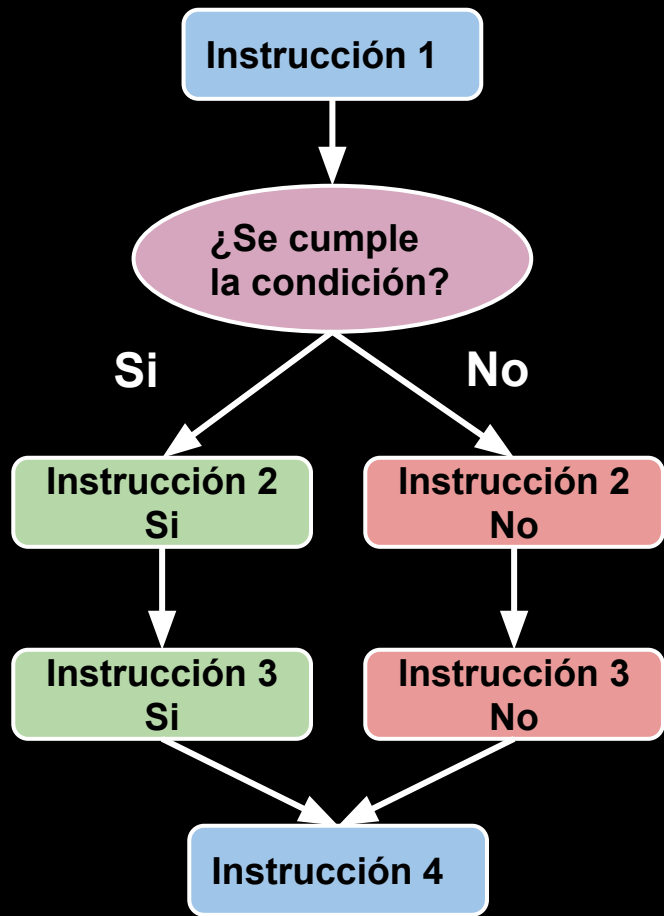
Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	XXXX
0x0002	XXXX
....
0x000F	XXXX

A vertical arrow points downwards along the left side of the table, indicating the sequence of memory addresses.

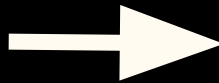
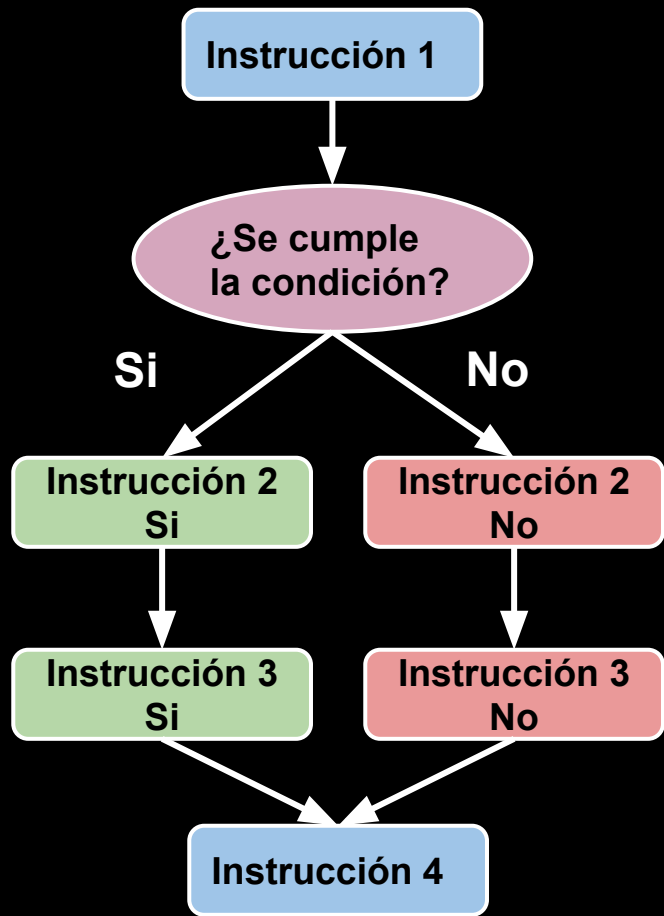




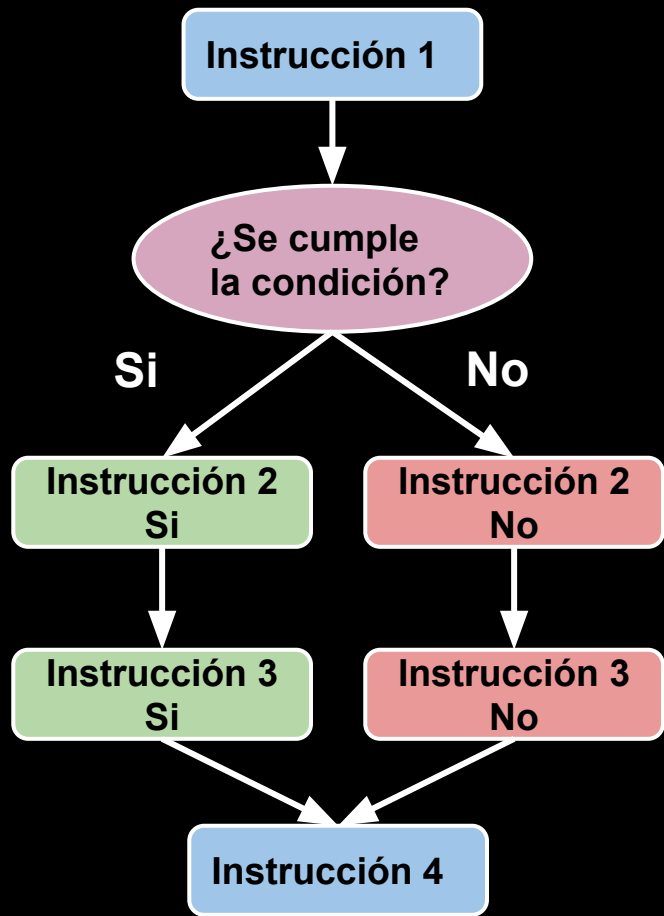
Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	XXXX
0x0002	XXXX
0x0003	XXXX
0x0004	XXXX
0x0005	XXXX
0x0006	XXXX
....



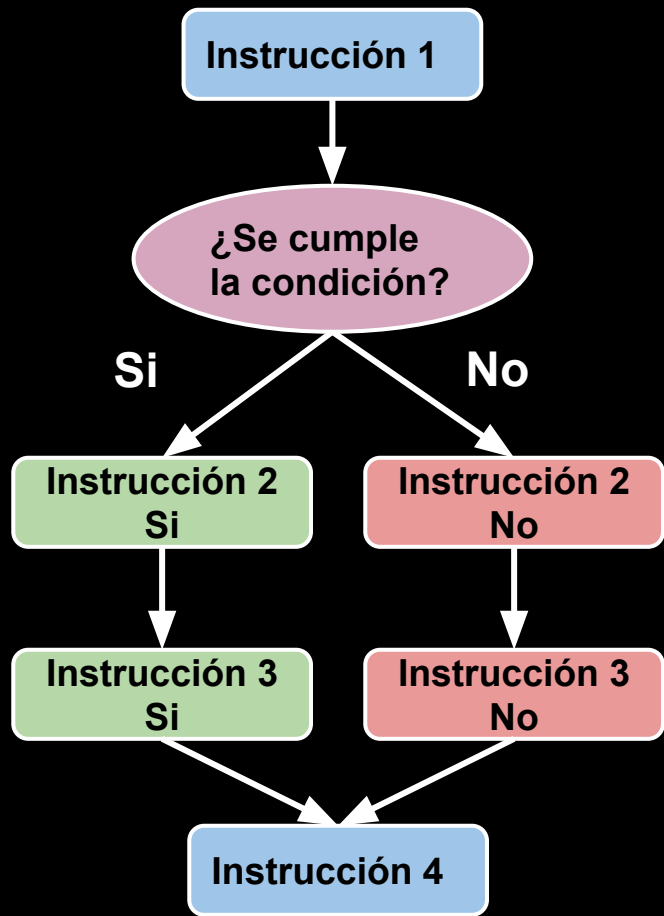
Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	XXXX
0x0002	XXXX
0x0003	XXXX
0x0004	XXXX
0x0005	XXXX
0x0006	XXXX
....



Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	XXXX
0x0002	XXXX
0x0003	XXXX
0x0004	XXXX
0x0005	XXXX
0x0006	XXXX
....



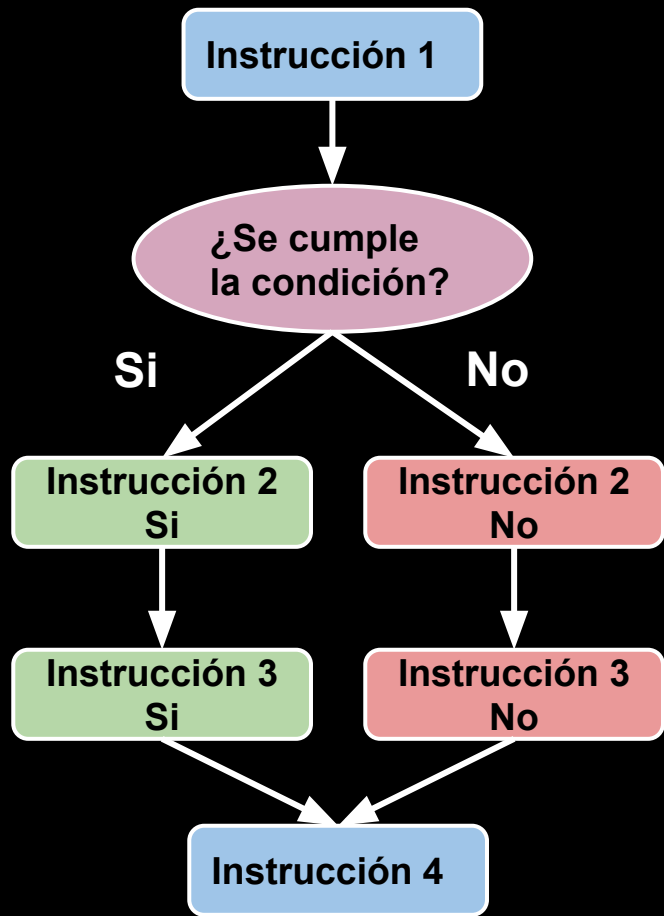
Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	XXXX
0x0002	XXXX
0x0003	XXXX
0x0004	XXXX
0x0005	XXXX
0x0006	XXXX
....



Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	XXXX
0x0002	XXXX
0x0003	XXXX
0x0004	XXXX
0x0005	XXXX
0x0006	XXXX
....

Diagram illustrating memory access:

- A red arrow points down to the memory address **0x0002**.
- A green arrow points down to the memory address **0x0003**.
- A green arrow points down to the memory address **0x0006**.




Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	XXXX
0x0002	XXXX
0x0003	XXXX
0x0004	XXXX
0x0005	XXXX
0x0006	XXXX
....

Execution paths indicated by arrows:

- Red arrow:** Starts at the top, points down to the first **0x0002** row, then curves left and points down to the **0x0003** row.
- Green arrow:** Starts at the top, points down to the second **0x0002** row, then curves left and points down to the **0x0004** row.

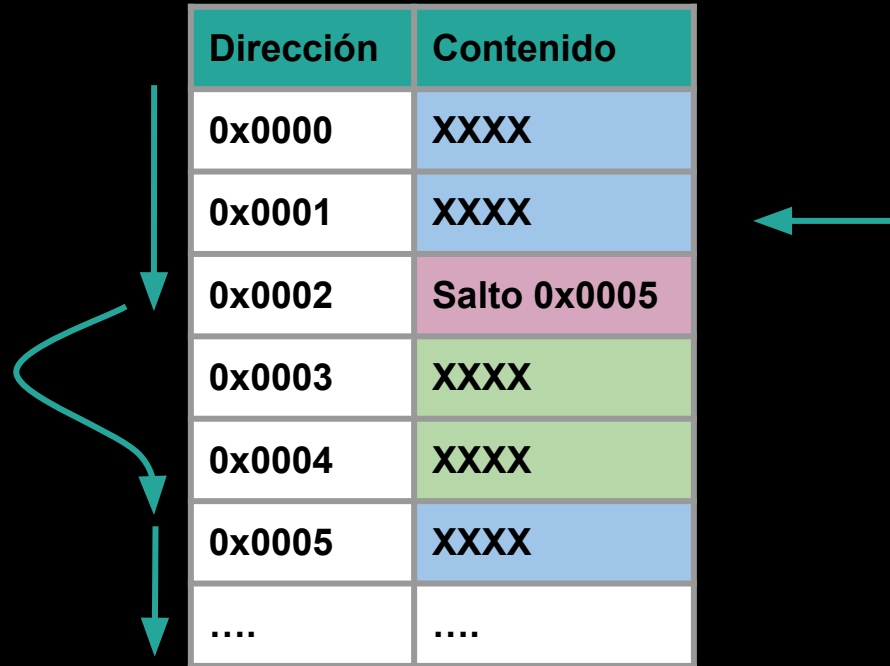
Salto



The diagram illustrates a jump instruction in memory. A vertical sequence of memory cells is shown, each with a 'Dirección' (Address) and a 'Contenido' (Content). The addresses range from 0x0000 to 0x0005, followed by an ellipsis. The contents are 'XXXX' for most addresses, except for 0x0002, which contains 'Salto 0x0005'. A red arrow points down from the top, and a curved red arrow points from the 'Salto 0x0005' entry to the start of the row for address 0x0005, demonstrating the jump.

Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	Salto 0x0005
0x0003	XXXX
0x0004	XXXX
0x0005	XXXX
....

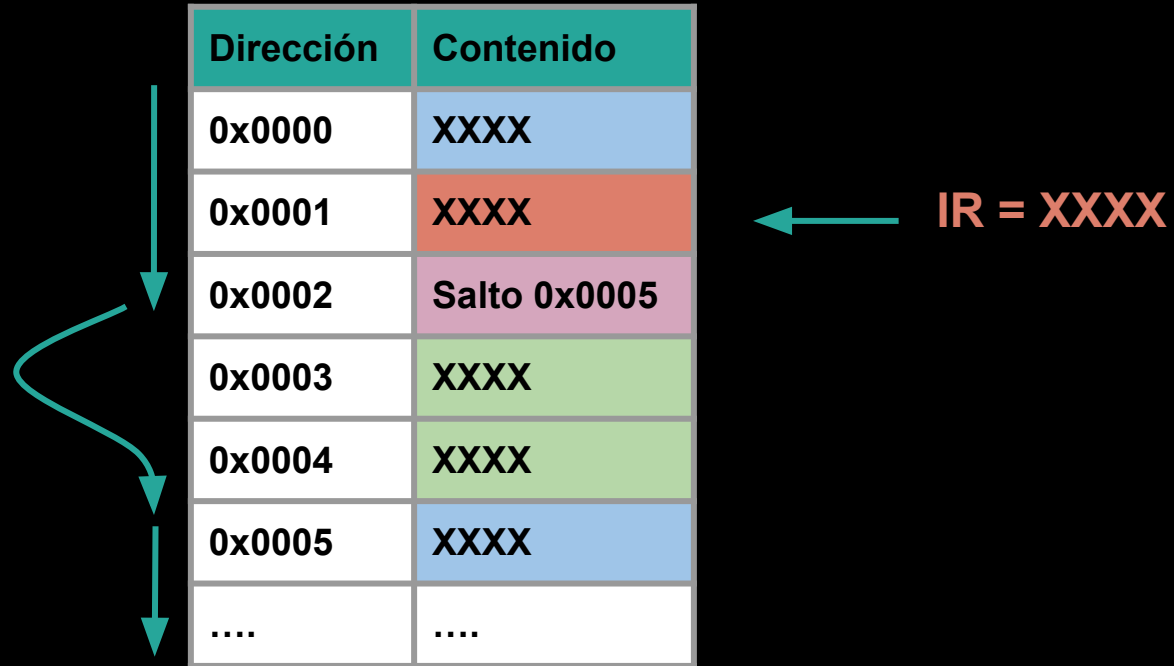
Salto



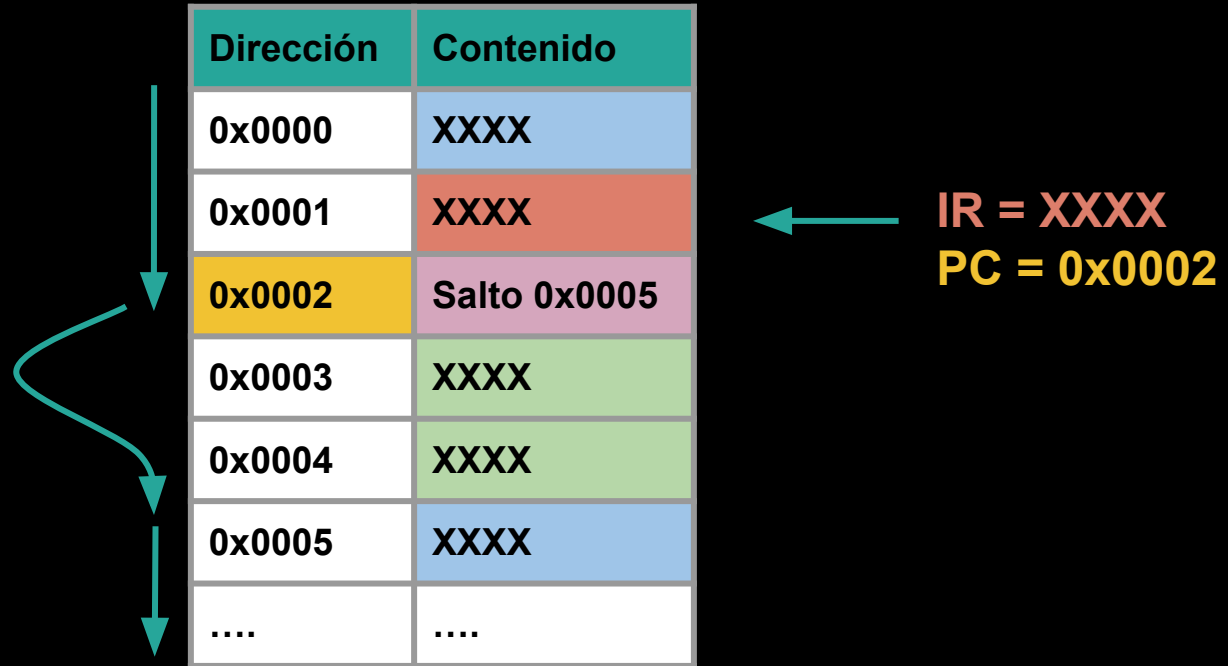
The diagram illustrates a jump instruction stored in memory. A table represents the memory layout with two columns: 'Dirección' (Address) and 'Contenido' (Content). The addresses range from 0x0000 to 0x0005, with an ellipsis for subsequent addresses. The contents are 'XXXX' for most addresses, except for 0x0002, which contains the instruction 'Salto 0x0005'. A red arrow points to the instruction at 0x0002, and a curved red arrow indicates the jump from 0x0002 to 0x0005. A horizontal red arrow points to the table from the right.

Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	Salto 0x0005
0x0003	XXXX
0x0004	XXXX
0x0005	XXXX
....

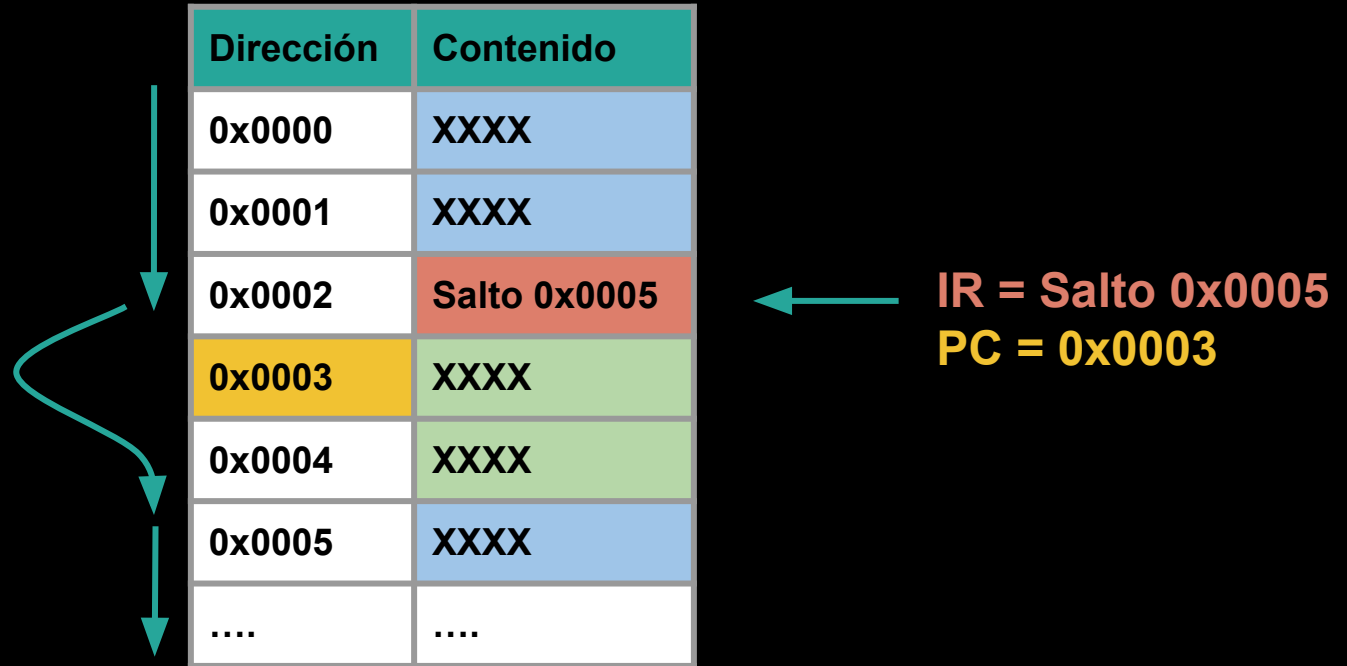
Salto



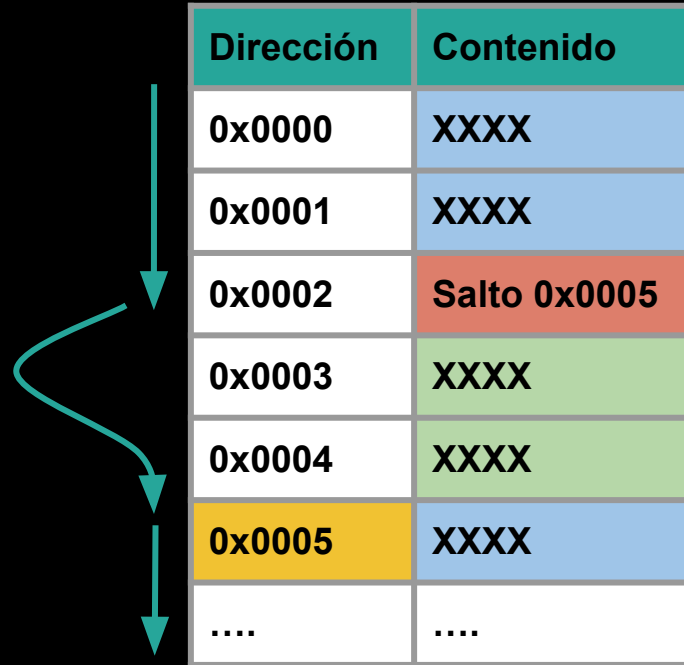
Salto



Salto



Salto

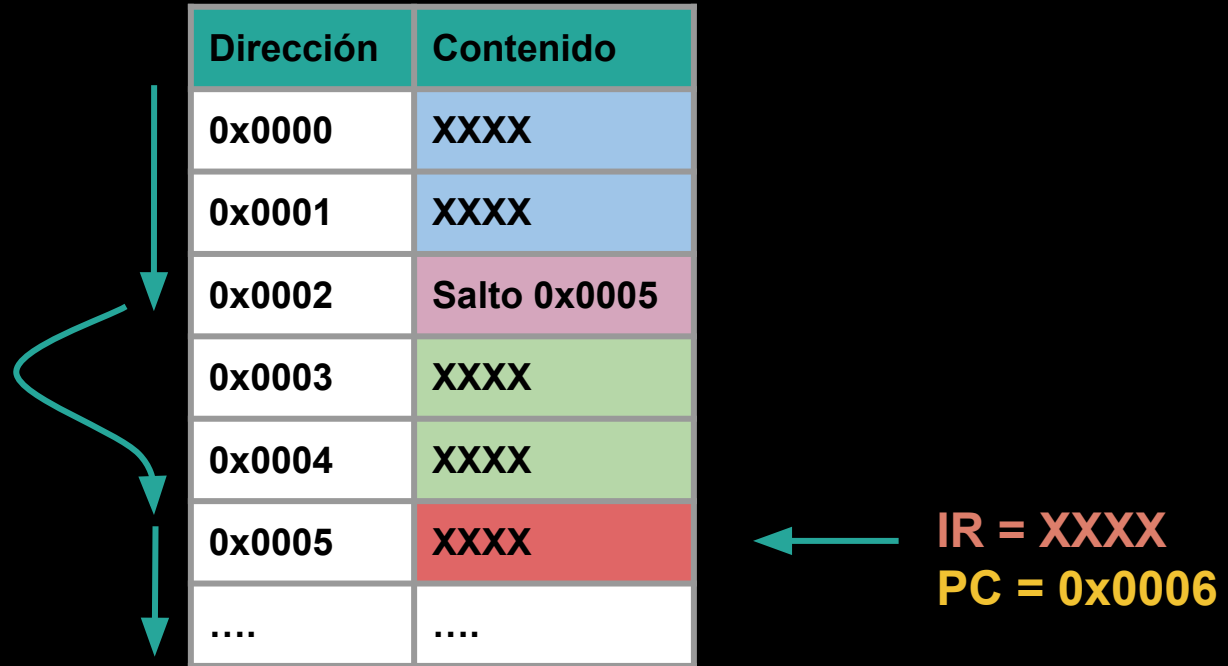


Dirección	Contenido
0x0000	XXXX
0x0001	XXXX
0x0002	Salto 0x0005
0x0003	XXXX
0x0004	XXXX
0x0005	XXXX
....

Durante la ejecución

← **IR = Salto 0x0005**
PC = 0x0005

Salto



Salto

- Salto absoluto

Salto

- Salto absoluto
 - Pone el valor del pc en una dirección determinada

Salto

- Salto absoluto
 - Pone el valor del pc en una dirección determinada
- Salto relativo

Salto

- Salto absoluto
 - Pone el valor del pc en una dirección determinada
- Salto relativo
 - Le suma al valor del PC un desplazamiento

Salto

- Salto condicional

Salto

- Salto condicional
 - Salta sólo si una condición es verdadera

Salto

- Salto condicional
 - Salta sólo si una condición es verdadera
- Salto incondicional

Salto

- Salto condicional
 - Salta sólo si una condición es verdadera
- Salto incondicional
 - Salta siempre

Salto

- Para saber si saltamos en un salto condicional usamos

Saltos

- Para saber si saltamos en un salto condicional usamos
 - Los flags (N, V, Z, C)

Saltos

- Para saber si saltamos en un salto condicional usamos
 - Los flags (N, V, Z, C)
- ¿Cómo los seteo?

Saltos

- Para saber si saltamos en un salto condicional usamos
 - Los flags (N, V, Z, C)
- ¿Cómo los seteo?
 - Usando CMP Destino, Origen

Saltos

- Para saber si saltamos en un salto condicional usamos
 - Los flags (N, V, Z, C)
- ¿Cómo los seteo?
 - Usando CMP Destino, Origen
 - Destino - Origen

Saltos

- Para saber si saltamos en un salto condicional usamos
 - Los flags (N, V, Z, C)
- ¿Cómo los seteo?
 - Usando CMP Destino, Origen
 - Destino - Origen
- ¿Qué Salto uso y cuándo?

Salto

- Para saber si saltamos en un salto condicional usamos
 - Los flags (N, V, Z, C)
- ¿Cómo los seteo?
 - Usando CMP Destino, Origen
 - Destino - Origen
- ¿Qué Salto uso y cuándo?

.....

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

Salto	Cod Op	Descripción
CMP	0110	Modifica los Flags según el resultado de Dest – Origen

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

Instrucción tipo 1

Salto	Cod Op	Descripción
CMP	0110	Modifica los Flags según el resultado de Dest – Origen

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

Instrucción tipo 1

Salto	Cod Op	Descripción
CMP	0110	Modifica los Flags según el resultado de Dest – Origen

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

Salto	Cod Op	Descripción
JMP	1010	PC <- Origen

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

Instrucción tipo 1

Salto	Cod Op	Descripción
CMP	0110	Modifica los Flags según el resultado de Dest – Origen

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

Instrucción tipo 2

Salto	Cod Op	Descripción
JMP	1010	PC <- Origen

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

Instrucción tipo 1

Salto	Cod Op	Descripción
CMP	0110	Modifica los Flags según el resultado de Dest – Origen

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

Instrucción tipo 2

Salto	Cod Op	Descripción
JMP	1010	PC <- Origen

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

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

Instrucción tipo 1

Salto	Cod Op	Descripción
CMP	0110	Modifica los Flags según el resultado de Dest – Origen

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

Instrucción tipo 2

Salto	Cod Op	Descripción
JMP	1010	PC <- Origen

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

Instrucción tipo 4

Salto	Cod Op	Descripción	Condición
JE	0001	Igual / Cero	Z
JNE	1001	No igual	$\neg Z$
JLE	0010	Menor o igual con signo	$Z + (N \oplus V)$
JG	1010	Mayor con signo	$\neg (Z + (N \oplus V))$
JL	0011	Menor con signo	$N \oplus V$
JGE	1011	Mayor o igual con Signo	$\neg (N \oplus V)$
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

Arquitecturas Q: Q4

Instrucciones:

Arquitecturas Q: Q4

Instrucciones:

MUL, MOV, ADD, SUB, DIV, CALL, RET

Arquitecturas Q: Q4

Instrucciones:

MUL, MOV, ADD, SUB, DIV, CALL, RET, CMP,
JMP, JE, JNE, JLE, JG, JL, JGE, JLEU, JGU, JCS,
JNEG, JVS.

Arquitecturas Q: Q4

Instrucciones:

MUL, MOV, ADD, SUB, DIV, CALL, RET, CMP, JMP, JE, JNE, JLE, JG, JL, JGE, JLEU, JGU, JCS, JNEG, JVS.

Operandos (Modos de direccionamiento):

Registro (modo registro)

Constante (modo inmediato)

Dirección memoria (modo directo)

Ejercicios

Ensamblar:

CMP R0, R3

JE **esCero**

SUB R0, 0x0001

JMP **guardar**

esCero: MOV R0, 0x0000

guardar: MOV [0x0003], R0

Ejercicios

En **R0**, **R1** y **R3** están guardadas las notas de los dos parciales y un tp de un alumno. El alumno promociona si el promedio de los tres es igual o mayor a 7. Hacer una rutina que guarde un **uno** en la celda **0x000A** si el alumno promociona o bien un **cero** si el alumno debe dar el final.