

# Ejercicio motivador

El siguiente ejercicio es motivador,  
motívese:

Dada una cadena en R0, hacer la rutina  
quintoYSeptimo1 que devuelva un 1 en R1 si  
el quinto y séptimo bit de R0 son iguales a 1.  
En caso contrario devuelve un 0.

1011 0010 0100 1001 R1 = 0x0000

1011 0010 0101 1001 R1 = 0x0001



Y si en vez de 2 fuesen 7 bits  
los que quisiera observar?



# Lógica

0



Falso

1



Verdadero

# Lógica

Operaciones:

# Lógica

Operaciones:

- Se aplican sobre cadenas

# Lógica

Operaciones:

- Se aplican sobre cadenas
- Actúan bit a bit



# Lógica

Operaciones:

- AND:

# Lógica

Operaciones:

- AND: Realiza el “Y” lógico entre los bits de las dos cadenas

# Lógica

Operaciones:

- AND: Realiza el “Y” lógico entre los bits de las dos cadenas

| A | B | A AND B |
|---|---|---------|
| 0 | 0 | 0       |
| 0 | 1 | 0       |
| 1 | 0 | 0       |
| 1 | 1 | 1       |

# Lógica

Operaciones:

- AND: Realiza el “Y” lógico entre los bits de las dos cadenas

AND

11010010

01001110

---

# Lógica

Operaciones:

- AND: Realiza el “Y” lógico entre los bits de las dos cadenas

AND

11010010

01001110

---

01000010

# Lógica

Operaciones:

- OR:

# Lógica

Operaciones:

- OR: Realiza el “O” lógico entre los bits de las dos cadenas

# Lógica

Operaciones:

- OR: Realiza el “O” lógico entre los bits de las dos cadenas

| A | B | A OR B |
|---|---|--------|
| 0 | 0 | 0      |
| 0 | 1 | 1      |
| 1 | 0 | 1      |
| 1 | 1 | 1      |



# Lógica

Operaciones:

- OR: Realiza el “O” lógico entre los bits de las dos cadenas

OR

11010010

01001110

---

# Lógica

Operaciones:

- OR: Realiza el “O” lógico entre los bits de las dos cadenas

OR

11010010

01001110

---

11011110

# Lógica

Operaciones:

- NOT:

# Lógica

Operaciones:

- NOT: Realiza la negación de los bits de una cadena

# Lógica

Operaciones:

- NOT: Realiza la negación de los bits de una cadena

| A | NOT A |
|---|-------|
| 0 | 1     |
| 1 | 0     |

# Lógica

Operaciones:

- NOT: Realiza la negación de los bits de una cadena

NOT     11010010

# Lógica

Operaciones:

- NOT: Realiza la negación de los bits de una cadena

|     |          |
|-----|----------|
| NOT | 11010010 |
|     | <hr/>    |
|     | 00101101 |

# Lógica

Operaciones:

- XOR:



# Lógica

Operaciones:

- XOR: Realiza el “O exclusivo” lógico entre los bits de las dos cadenas

# Lógica

Operaciones:

- XOR: Realiza el “O exclusivo” lógico entre los bits de las dos cadenas

| A | B | A XOR B |
|---|---|---------|
| 0 | 0 | 0       |
| 0 | 1 | 1       |
| 1 | 0 | 1       |
| 1 | 1 | 0       |

# Lógica

Operaciones:

- XOR: Realiza el “O exclusivo” lógico entre los bits de las dos cadenas

|     |                   |
|-----|-------------------|
|     | 11010010          |
| XOR | 10110101          |
|     | <u>          </u> |

# Lógica

Operaciones:

- XOR: Realiza el “O exclusivo” lógico entre los bits de las dos cadenas

|     |          |
|-----|----------|
|     | 11010010 |
| XOR | 10110101 |
|     | <hr/>    |
|     | 01100111 |

# Mascaras



# Mascaras

- Son cadenas binarias que se combinan con otras mediante operaciones lógicas
- Sirven para analizar el contenido de las cadenas

# Observaciones

- Con AND:

Si quiero dejar pasar el bit de mi cadena, aplico un 1 en la máscara

Si quiero filtrar el bit de mi cadena, aplico un 0 en la máscara

# Observaciones

- Con AND:

Es decir: Si el bit me interesa, pongo un 1 y si no, un 0



# Mascaras

## Observaciones

- Con AND:

|     |          |
|-----|----------|
|     | 11010010 |
| AND | 10110101 |
|     | <hr/>    |
|     | 10010000 |

# Mascaras

## Observaciones

- Con AND:

|     |           |
|-----|-----------|
|     | XXXXXXXXX |
| AND | 10110101  |
|     | <hr/>     |
|     | X0XX0X0X  |

# Mascaras

## Observaciones

- Con OR:

Si quiero dejar pasar el bit de mi cadena, aplico un 0 en la máscara

Si quiero filtrar el bit de mi cadena, aplico un 1 en la máscara

# Observaciones

- Con OR:

Es decir: Si el bit me interesa, pongo un 0 y si no, un 1

# Mascaras

## Observaciones

- Con OR:

$$\begin{array}{r} \text{OR} \quad 11010010 \\ \quad 10000100 \\ \hline \quad 11010110 \end{array}$$

# Mascaras

## Observaciones

- Con OR:

OR

XXXXXXXXXX

10000100

---

1XXXX1XX

# Mascaras

- Ej: Resolver usando máscaras:

Si los bits 2 y 8 de una cadena en R0 son iguales a los de la cadena que está en R3, se debe poner un 1 en R2

0110 0101 1100 0100

0000 0111 0110 0100

# Mascaras

- Ejercicio:

Se sabe que en los primeros 4 bits de una cadena se guarda la edad de un chico. Hacer una rutina que dada una cadena en R4 devuelva la edad del chico en el registro R0.



# Mascaras

- Ejercicio:

```
obtenerEdad: AND R4, 0x000F
```

```
MOV R0, R4
```

```
RET
```

# Mascaras

- Ejercicio:

Se sabe que un menú se compone de dos elementos: Bebida(guardada en R0) y comida (guardada en R1).

El primer bit de un elemento determina si tiene descuento (Tiene = 1, No tiene = 0).

Hacer una rutina que determine si algún elemento tiene descuento, dejando un 1 en R2 en caso de tener y un 0 en caso contrario

# Mascaras

•Ejercicio:

tieneDescuento: MOV R2, R0

OR R2, R1

AND R2, 0x0001

RET

# Instrucciones en Q

## Formato:

|                   |                             |                         |                              |                             |
|-------------------|-----------------------------|-------------------------|------------------------------|-----------------------------|
| Cod Op<br>(4bits) | Modo<br>Destino (6<br>bits) | Modo origen<br>(6 bits) | <b>Destino<br/>(16 bits)</b> | <b>Origen<br/>(16 bits)</b> |
|-------------------|-----------------------------|-------------------------|------------------------------|-----------------------------|

| Operación | Código | Efecto  |
|-----------|--------|---|
| AND       | 0100   | $\text{Dest} \leftarrow \text{Dest} \wedge \text{Origen}$ |
| OR        | 0101   | $\text{Dest} \leftarrow \text{Dest} \vee \text{Origen}$   |

Y XOR?

**NO HAY**



# Instrucciones en Q

## Formato:

|                   |                          |                     |                              |
|-------------------|--------------------------|---------------------|------------------------------|
| Cod Op<br>(4bits) | Modo Destino<br>(6 bits) | Relleno<br>(000000) | <b>Destino<br/>(16 bits)</b> |
|-------------------|--------------------------|---------------------|------------------------------|

| Operación | Código | Efecto                        |
|-----------|--------|-------------------------------|
| NOT       | 1001   | Dest $\leftarrow$ $\neg$ Dest |

# Preguntas

