

Máscaras

Organización de computadoras

Universidad Nacional de Quilmes

June 9, 2020

Desafío: Conocer el valor de ciertos bits en una cadena

Pero antes... necesitamos
algunas herramientas previas

Operaciones sobre cadenas

Operaciones sobre cadenas: AND bit a bit

	1	0	1	0
AND	0	1	0	1
	<hr/>			

Operaciones sobre cadenas

Operaciones sobre cadenas: AND bit a bit

$$\begin{array}{r} \text{AND} \quad 1010 \\ \quad \quad 0101 \\ \hline \quad \quad 0000 \end{array}$$

Operaciones sobre cadenas

Operaciones sobre cadenas: OR bit a bit

$$\begin{array}{r} 1010 \\ \text{OR } 0101 \\ \hline \end{array}$$

Operaciones sobre cadenas

Operaciones sobre cadenas: OR bit a bit

$$\begin{array}{r} \text{OR} \quad 1010 \\ \quad 0101 \\ \hline \quad 1111 \end{array}$$

Operaciones sobre cadenas

Operaciones sobre cadenas: NOT bit a bit

1010

NOT _____

Operaciones sobre cadenas

Operaciones sobre cadenas: NOT bit a bit

$$\begin{array}{r} \text{NOT} \\ \hline 1010 \\ \hline 0101 \end{array}$$

Operaciones sobre cadenas

Ejercicios

1 NAND

$$\begin{array}{r} 1100 \\ 0100 \\ \hline ? \end{array}$$

2 NOR

$$\begin{array}{r} 1100 \\ 0100 \\ \hline ? \end{array}$$

3 XOR

$$\begin{array}{r} 1100 \\ 0100 \\ \hline ? \end{array}$$

Máscaras

Máscara

Cadena binaria que se aplica sobre otra mediante **determinada operación lógica** para **descubrir características sobre la segunda**

$$\begin{array}{r} \text{AND } \quad 0101 \text{ cadena} \\ \quad \quad 1111 \text{ máscara} \\ \hline \quad \quad 0101 \end{array}$$

Uso del AND

- Para **preservar** el bit original:
- Para **anular** un bit (dejarlo en 0):

Máscaras

¿Para que usamos AND?

- Si se quiere **preservar** el bit: usar **1**
 $x \text{ AND } 1 = x$
- Si se quiere **anular** un bit: usar **0**
 $x \text{ AND } 0 = 0$

Máscaras

¿Para que usamos AND?

- Si se quiere **preservar** el bit: usar **1**
 $x \text{ AND } 1 = x$
- Si se quiere **anular** un bit: usar **0**
 $x \text{ AND } 0 = 0$



$$\begin{array}{r} X_3 X_2 X_1 X_0 \\ \text{AND } \underline{0 \ 0 \ 0 \ 1} \end{array}$$

Máscaras

¿Para que usamos AND?

- Si se quiere **preservar** el bit: usar **1**
 $x \text{ AND } 1 = x$
- Si se quiere **anular** un bit: usar **0**
 $x \text{ AND } 0 = 0$



$$\begin{array}{r} \phantom{\text{AND}} \quad X_3 X_2 X_1 X_0 \\ \text{AND} \quad 0 \ 0 \ 0 \ 1 \\ \hline \phantom{\text{AND}} \quad 0 \ 0 \ 0 \ X_0 \end{array}$$

Máscaras

¿Para que usamos OR?

- Si se quiere **preservar** el bit: usar **0**
 $x \text{ OR } 0 = x$
- Si se quiere **activar** un bit: usar **1**
 $x \text{ OR } 1 = 1$

Máscaras

¿Para que usamos OR?

- Si se quiere **preservar** el bit: usar **0**
 $x \text{ OR } 0 = x$
- Si se quiere **activar** un bit: usar **1**
 $x \text{ OR } 1 = 1$



$$\begin{array}{r} X_3 X_2 X_1 X_0 \\ \text{OR} \quad 0 \quad 0 \quad 0 \quad 1 \\ \hline \end{array}$$

Máscaras

¿Para que usamos OR?

- Si se quiere **preservar** el bit: usar **0**
 $x \text{ OR } 0 = x$
- Si se quiere **activar** un bit: usar **1**
 $x \text{ OR } 1 = 1$



$$\begin{array}{r} x_3 x_2 x_1 x_0 \\ \text{OR} \quad 0 \quad 0 \quad 0 \quad 1 \\ \hline x_3 x_2 x_1 \quad 1 \end{array}$$

Máscaras: ejemplos de uso

Example

Determinar si la cadena en R0 es impar

Máscaras: ejemplos de uso

Example

Determinar si la cadena en R0 es impar

```
AND R0, 0x0001
```

```
JNE saltarAEsImpar
```

Máscaras: ejemplos de uso

Example

Copiar el byte mas significativo de la celda 0348 en el registro R1

Máscaras: ejemplos de uso

Example

Copiar el byte mas significativo de la celda 0348 en el registro R1

```
MOV R1, [0348]  
AND R1, 0xFF00
```

Máscaras: ejemplos de uso

Ejercicio (NO se entrega): Si la celda [CCCC] contiene un número par, sumar 30 al valor de R3. En caso contrario sumar 70 al valor de R4

Máscaras: ejemplos de uso

Ejercicio (NO se entrega): Si la celda [CCCC] contiene un número par, sumar 30 al valor de R3. En caso contrario sumar 70 al valor de R4

```
MOV R1, [CCCC]
AND R1, 0x0001
JNE noespar
ADD R3, 0x001E
JMP sigue
noespar:ADD R4, 0x0046
sigue:
```


Máscaras: ejemplos de uso

Si las celdas CCCC y CCCD contienen números impares, restarles 0x0001 a ambas

Máscaras: ejemplos de uso

Si las celdas CCCC y CCCD contienen números impares, restarles 0x0001 a ambas

Ayudita:

```
si ([CCCC] es impar)
  si ([CCCD] es impar)
    [CCCC] <-- [CCCC]-1
    [CCCD] <-- [CCCD]-1
  fin si
fin si
```

Máscaras: ejemplos de uso

Permisos de acceso sobre archivos

- Con 3 bits se indica:
 - 1 ¿puedo leer? (r)
 - 2 ¿puedo escribir? (w)
 - 3 ¿puedo ejecutar? (x)
- Con 3 cadenas se describen permisos de **usuario**, **grupo** y **otros**

Example

La cadena **1111111111** le da todos los permisos a todos

Máscaras: ejemplos de uso

Permisos de acceso sobre archivos

¿Cómo saber si otro usuario del grupo puede escribirlo?

Máscaras: ejemplos de uso

Permisos de acceso sobre archivos

¿Cómo saber si otro usuario del grupo puede escribirlo?



$$\begin{array}{r} \text{AND} \quad \text{????????} \\ \quad \quad \text{000010000} \\ \hline \quad \quad \text{0000?0000} \end{array}$$

Arquitectura Q5

Arquitectura Q5

Seguimos avanzando

Arquitectura Q5

- 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 5 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

Arquitectura Q5: formato de instrucciones

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

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 con un operando Destino: NOT**

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

- Instrucciones sin operandos: RET

Cod_Op (4b)	Relleno (000000000000)
----------------	---------------------------

- Saltos **condicionales** y relativos

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

Arquitectura Q5: 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	
		AND	0100	
		OR	0101	
		CMP	0110	
		DIV	0111	

Arquitectura Q5: Instrucciones con un operando Destino

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

Operación	CodOp	Efecto
NOT	1001	Dest \leftarrow NOT Dest

Arquitectura Q5: Ejercicio

Ensamblar las siguientes instrucciones

AND R0, R1

OR [0A0A], 0xFF00

NOT [F123]

Op.	CodOp	Formato de instrucción					
NOT	1001	Cod.Op (4b)	Modo Destino (6b)	Relleno (000000)	Operando Origen (16b)		
AND	0100	Cod.Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)	
OR	0101	Cod.Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)	

Arquitectura Q5: Ejercicio

Completar la tabla de accesos

Instrucción	B.Inst.	B.Op.	Alm.Op.
AND R0, R1			
OR [0A0A], 0xFF00			
NOT [F123]			

Arquitectura Q5: Ejercicio

En Linux, los permisos de acceso a los archivos se codifican:

- Con 3 bits los tipos de accesos: R (leer), W (escribir), X (ejecutar)
- Hay 3 categorías: permisos de **usuario**, **grupo** y **otros**

Por ejemplo:

- 111 000 000 da permisos al usuario para leer, escribir y ejecutar
- 111 100 000 da todos los permisos al usuario y el permiso de lectura a los otros miembros del grupo

Ejercicio: Escribir un programa que indique si otro usuario del grupo puede escribir un archivo cuyos permisos están en R0. Ponga un 1 si es posible, y 0 en caso contrario.