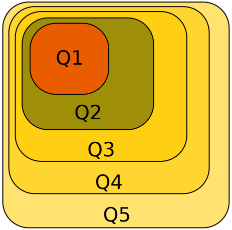
Operaciones lógicas + máscaras
Punto Fijo

Organización de computadoras

Universidad Nacional de Quilmes

21 de octubre de 2013

Repaso



1. Repeticiones
2. Arreglos
3. Modos de direccionamiento
    1. Indirecto
    2. Indirecto por registro

Repaso: Repeticiones

Example (¿Cómo encender el piloto del calefón?)

1. poner la perilla en posición piloto
2. acercar un fósforo mientras se presiona la perilla
3. mantener presionando aproximadamente 20 segundos
4. Si al liberar la perilla el piloto se apaga, volver al paso (1), sino seguir con el paso (5)
5. ...

Repaso: Repeticiones

Estructura general de la repetición

```
        Inicialización
arriba: Controlar condición de ciclo
        Si no se cumple: <salir>
        Cuerpo del ciclo
        Volver <arriba>
 salir: Finalizar programa
```

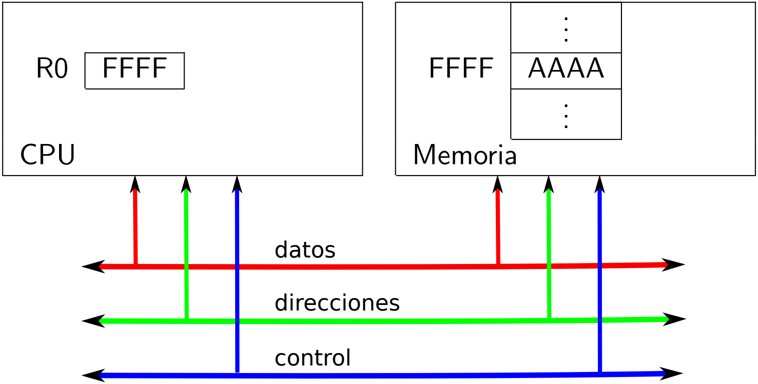
Repaso: Arreglos

⋮	⋮
000A	1er valor
000B	2do valor
000C	3er valor
000D	4to valor
000E	5to valor
000F	6to valor
⋮	⋮

Arreglo de valores
Posiciones de memoria consecutivas que contienen una colección de elementos. Cada elemento puede ocupar mas de una celda.

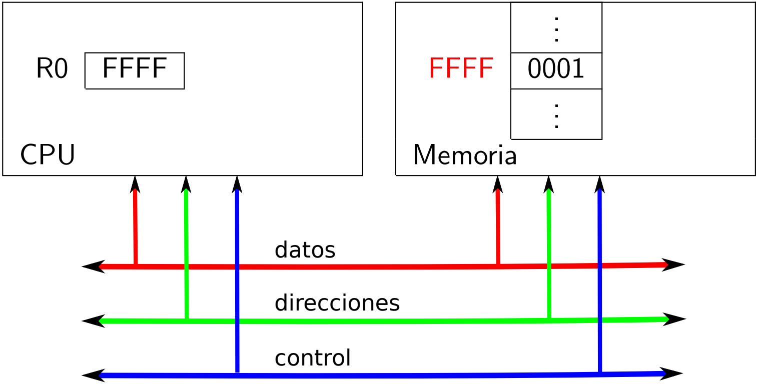
Repaso: Modo indirecto

MOV [R0],0x0001



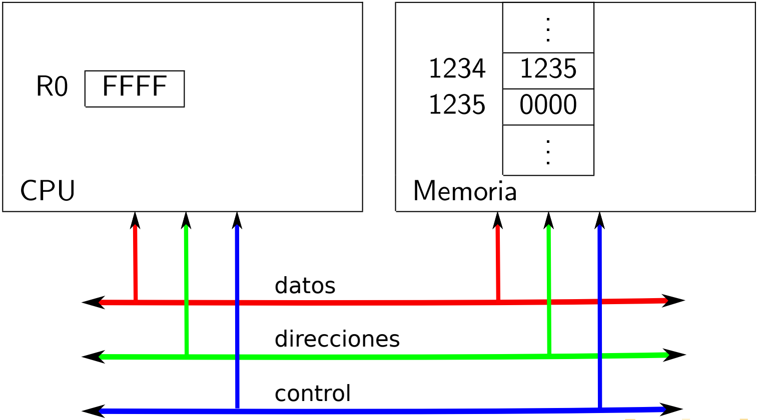
Repaso: Modo indirecto

MOV [R0],0x0001



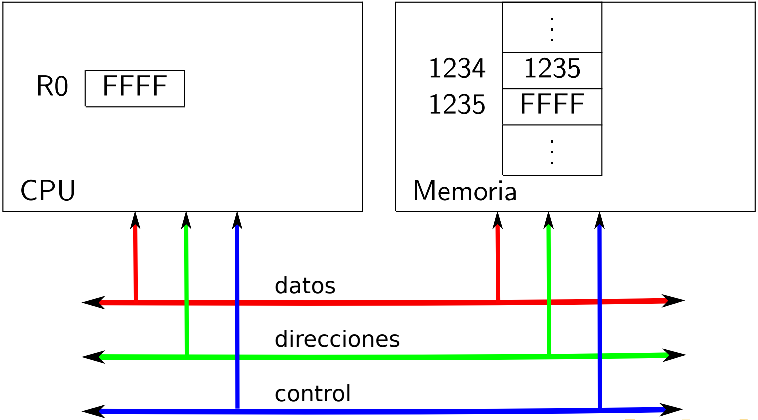
Repaso: Modo indirecto

MOV [[1234]],R0



Repaso: Modo indirecto

MOV [[1234]],R0



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

$$
\begin{array}{rr}
 & 1010 \\
\text{AND} & \underline{0101}
\end{array}
$$

Operaciones sobre cadenas

Operaciones sobre cadenas: AND bit a bit

$$
\begin{array}{rr}
 & 1010 \\
\text{AND} & 0101 \\
\hline
 & 0000
\end{array}
$$

Operaciones sobre cadenas

Operaciones sobre cadenas: OR bit a bit

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

Operaciones sobre cadenas

Operaciones sobre cadenas: OR bit a bit

$$\begin{array}{lr} & 1010 \\ \text{OR} & 0101 \\ \hline & 1111 \end{array}$$

Operaciones sobre cadenas

Operaciones sobre cadenas: NOT bit a bit

$$\begin{array}{r} 1010 \\ \text{NOT} \quad \overline{} \end{array}$$

Operaciones sobre cadenas

Operaciones sobre cadenas: NOT bit a bit

$$
\begin{array}{r}
\phantom{\text{NOT}}\ 1010 \\
\text{NOT}\quad\underline{} \\
0101
\end{array}
$$

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 una operación lógica para descubrir características sobre esa cadena

	0101	cadena
AND	1111	máscara
	0101	

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 AND 1 = x
- Si se quiere anular un bit: usar 0
  x AND 0 = 0

Máscaras

¿Para que usamos AND?

- Si se quiere preservar el bit: usar 1
  x AND 1 = x
- Si se quiere anular un bit: usar 0
  x AND 0 = 0

$$
\begin{array}{rc}
 & x_3 x_2 x_1 x_0 \\
\text{AND} & 0\ 0\ 0\ 1 \\
\hline
\end{array}
$$

Máscaras

¿Para que usamos AND?

- Si se quiere preservar el bit: usar 1
  x AND 1 = x
- Si se quiere anular un bit: usar 0
  x AND 0 = 0

$$
\begin{array}{rl}
 & x_3 x_2 x_1 x_0 \\
\text{AND} & 0\ 0\ 0\ 1 \\
\hline
 & 0\ 0\ 0\ x_0
\end{array}
$$

Máscaras

¿Para que usamos OR?

- Si se quiere preservar el bit: usar 0

  x OR 0 = x
- Si se quiere activar un bit: usar 1

  x OR 1 = 1

Máscaras

¿Para que usamos OR?

- Si se quiere preservar el bit: usar 0

  x OR 0 = x
- Si se quiere activar un bit: usar 1

  x OR 1 = 1

$$
\begin{array}{rc}
 & x_3 x_2 x_1 x_0 \\
\text{OR} & \underline{0\ 0\ 0\ 1}
\end{array}
$$

Máscaras

¿Para que usamos OR?

- Si se quiere preservar el bit: usar 0
  x OR 0 = x
- Si se quiere activar un bit: usar 1
  x OR 1 = 1

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

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 111111111 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

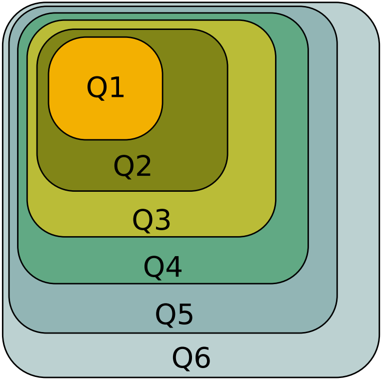
Permisos de acceso sobre archivos

¿Cómo saber si otro usuario del grupo puede escribirlo?

	?????????
AND	000010000
	0000?0000

Arquitectura **Q6**

Arquitectura **Q6**



Arquitectura Q6

- 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
  - modo indirecto: la dirección del valor buscado está contenido en una celda de memoria
  - modo registro indirecto: la dirección del valor buscado está contenido en un registro

Arquitectura **Q6**: 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 **Q6**: 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 **Q4**: 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 Q6: 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 **Q6**: Ejercicio

Completar la tabla de accesos

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

Arquitectura **Q6**: 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

Arquitectura **Q6**: 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.

Punto Fijo

Para hoy tenemos...

Desafío

Desafío

$\Downarrow$

Representar partes no enteras de una unidad (números fraccionarios)

Desafío

Representar partes no enteras de una unidad (números fraccionarios)

¿Cómo lo hacemos?

En el sistema decimal...

Se usa un caracter ","para separar las unidades de las fracciones

En el sistema decimal...

Se usa un caracter ”,”para separar las unidades de las fracciones

Si la cadena 16 vale: $10+6 = 1*10^1 + 6*10^0$

En el sistema decimal...

Se usa un caracter ","para separar las unidades de las fracciones

⇓

Si la cadena 16 vale: $10+6 = 1*10^1 + 6*10^0$

⇓

La cadena 1,6 vale: $1+0{,}6 = 1*10^0 + 6*10^{-1}$

La coma establece una posición para los pesos de los dígitos

Motivación

La coma establece una posición para los pesos de los dígitos

Motivación

La coma establece una posición para los pesos de los dígitos

Pesos					
10^1	10^0		10^{-1}	10^{-2}	Valor representado
1	6	,	0		16
	1	,	6		1,6
	0	,	1	6	0,16

Pesos fraccionarios: ¿Cómo se traslada a los binarios?

Pesos					
10^1	10^0		10^{-1}	10^{-2}	Valor representado
1	6	,	0		16
	1	,	6		1,6
	0	,	1	6	0,16

Pesos fraccionarios: ¿Cómo se traslada a los binarios?

Pesos					
10^1	10^0		10^{-1}	10^{-2}	Valor representado
1	6	,	0		16
	1	,	6		1,6
	0	,	1	6	0,16

⇓

Pesos					
2^1	2^0		2^{-1}	2^{-2}	Valor representado
1	1	,	0		$2^1 + 2^0 = 3$

Pesos fraccionarios: ¿Cómo se traslada a los binarios?

Pesos					
10^1	10^0		10^{-1}	10^{-2}	Valor representado
1	6	,	0		16
	1	,	6		1,6
	0	,	1	6	0,16

Pesos					
2^1	2^0		2^{-1}	2^{-2}	Valor representado
1	1	,	0		$2^1 + 2^0 = 3$
	1	,	1		$2^0 + 2^{-1} = 1 + 0,5 = 1,5$

Pesos fraccionarios: ¿Cómo se traslada a los binarios?

Pesos					
10^1	10^0		10^{-1}	10^{-2}	Valor representado
1	6	,	0		16
	1	,	6		1,6
	0	,	1	6	0,16

Pesos					
2^1	2^0		2^{-1}	2^{-2}	Valor representado
1	1	,	0		$2^1 + 2^0 = 3$
	1	,	1		$2^0 + 2^{-1} = 1 + 0,5 = 1,5$
	0	,	1	1	$2^{-1} + 2^{-2} = 0,5 + 0,25 = 0,75$

Pero... no podemos escribir la coma!

Sistema de Punto Fijo

Sistema de punto fijo

Sistema de numeración binario donde se establece una posición fija de la coma fraccionaria.

Sistema de Punto Fijo

Sistema de punto fijo
Sistema de numeración binario donde se establece una posición fija de la coma fraccionaria.

La coma se asume en un lugar fijo, y no se escribe

Para eso se acuerda un sistema de escritura (y lectura)

Sistema de Punto Fijo

Sistema de punto fijo
Sistema de numeración binario donde se establece una posición fija de la coma fraccionaria.

Suponer 2 bits enteros y 3 bits fraccionarios

Sistema de Punto Fijo

Sistema de punto fijo

Sistema de numeración binario donde se establece una posición fija de la coma fraccionaria.

Suponer 2 bits enteros y 3 bits fraccionarios

Los pesos son: $2^{1}2^{0}2^{-1}2^{-2}2^{-3}$

Sistema de Punto Fijo

Sistema de punto fijo

Sistema de numeración binario donde se establece una posición fija de la coma fraccionaria.

Suponer 2 bits enteros y 3 bits fraccionarios

Los pesos son: $2^1 2^0 2^{-1} 2^{-2} 2^{-3}$

$$\mathcal{I}(10100) = 2^1 + 2^{-1} = 2 + 0,5 = 2,5$$

Sistema de Punto Fijo

Ejercicio: interpretar

Parte entera		Parte Fraccionaria			
2^1	2^0	2^{-1}	2^{-2}	2^{-3}	Valor
1	0	0	0	0	?
0	1	0	0	0	?
0	0	0	0	1	?
0	1	0	0	1	?
0	0	0	1	0	?

Sistema de Punto Fijo

Notación

BSS(n,m) denota un sistema *Binario Sin Signo* con **n bits en total**, de los cuales m son fraccionarios

Parte entera		Parte Fraccionaria			
2^1	2^0	2^{-1}	2^{-2}	2^{-3}	Valor

Sistema de Punto Fijo

Notación

BSS(n,m) denota un sistema *Binario Sin Signo* con **n bits en total**, de los cuales m son fraccionarios

Parte entera		Parte Fraccionaria			
2^1	2^0	2^{-1}	2^{-2}	2^{-3}	Valor

BSS(5,3)

Comparación entre sistemas

Completar la siguiente tabla

Cadena	$BSS(2)$	$BSS(2,1)$
00		
01		
10		
11		

Comparación entre sistemas

Cadena	$BSS(2)$		$BSS(2,1)$	
00	0	0	0	0
01	$1 * 2^0$	1	$1 * 2^{-1}$	0,5
10	$1 * 2^1$	2	$1 * 2^0$	1
11	$1 * 2^1 + 1 * 2^0$	3	$1 * 2^0 + 1 * 2^{-1}$	1,5

Comparación entre sistemas

Completar la siguiente tabla

Cadena	$BSS(3)$	$BSS(3,1)$
000		
001		
010		
011		
100		
101		
110		
111		

Comparación entre sistemas

Cadena	$BSS(3)$		$BSS(3,1)$	
000	0	0	0	0
001	$1 * 2^0$	1	$1 * 2^{-1}$	0,5
010	$1 * 2^1$	2	$1 * 2^0$	1
011	$1 * 2^1 + 1 * 2^0$	3	$1 * 2^0 + 1 * 2^{-1}$	1,5
100	$1 * 2^2$	4	$1 * 2^1$	2
101	$1 * 2^2 + 1 * 2^0$	5	$1 * 2^1 + 1 * 2^{-1}$	2,5
110	$1 * 2^2 + 1 * 2^1$	6	$1 * 2^1 + 1 * 2^0$	3
111	$1 * 2^2 + 1 * 2^1 + 1 * 2^0$	7	$1 * 2^1 + 1 * 2^0 + 1 * 2^{-1}$	3,5

Interpretación en $BSS(n, m)$: dos mecanismos

Interpretación en $BSS(n, m)$: dos mecanismos

(A)

Sumar considerando los pesos fraccionarios

(B)

Interpretar el número como en $BSS()$ y dividir por 2^m

Interpretación en $BSS(n, m)$: dos mecanismos

(A)

Sumar considerando los pesos fraccionarios

(B)

Interpretar el número como en $BSS()$ y dividir por 2^m

$$\mathcal{I}_{bss(5,2)}(00101) = 2^0 + 2^{-2}$$

$$= 1,25$$

Interpretación en $BSS(n, m)$: dos mecanismos

(A)

Sumar considerando los pesos fraccionarios

$$\mathcal{I}_{bss(5,2)}(00101) = 2^0 + 2^{-2}$$

$$= 1,25$$

(B)

Interpretar el número como en $BSS()$ y dividir por 2^m

$$\mathcal{I}_{bss(5,2)}(00101) = \frac{\mathcal{I}_{bss(5)}(00101)}{4}$$

$$= \frac{5}{4} = 1,25$$

Rango

Rango

Intervalo de números representables

Rango

Rango

Intervalo de números representables

Calcular el rango del sistema $BSS(2,1)$

Rango

Rango

Intervalo de números representables

Calcular el rango del sistema $BSS(2,1)$

$\Downarrow$

Mínimo Interpretar la cadena que representa al mínimo:

$$\mathcal{I}_{bss(2,1)}(00) = 0$$

Rango

Rango
Intervalo de números representables

Calcular el rango del sistema $BSS(2,1)$

Mínimo Interpretar la cadena que representa al mínimo:

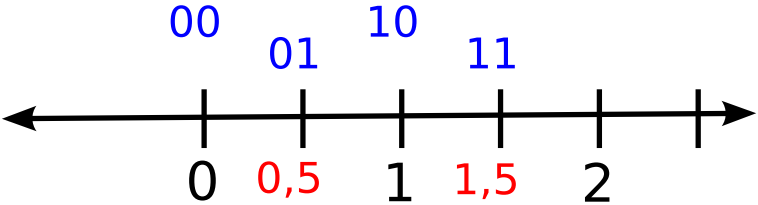
$$\mathcal{I}_{bss(2,1)}(00) = 0$$

Máximo Interpretar la cadena que representa al máximo:

$$\mathcal{I}_{bss(2,1)}(11) = 2^0 + 2^{-1} = 1 + 0,5 = 1,5$$

Rango

Gráficamente



Rango

Calcular el rango del sistema $BSS(2,0)$

Rango

Calcular el rango del sistema $BSS(2,0)$

Mínimo Interpretar la cadena que representa al mínimo:

$$\mathcal{I}_{bss(2,0)}(00) = 0$$

Rango

Calcular el rango del sistema $BSS(2,0)$

Mínimo Interpretar la cadena que representa al mínimo:

$$\mathcal{I}_{bss(2,0)}(00) = 0$$

Máximo Interpretar la cadena que representa al máximo:

Rango

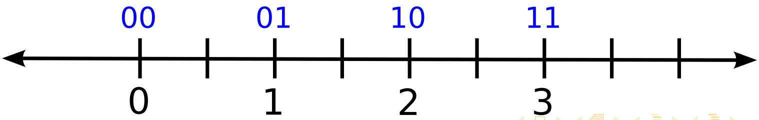
Calcular el rango del sistema $BSS(2,0)$

Mínimo Interpretar la cadena que representa al mínimo:

$$\mathcal{I}_{bss(2,0)}(00) = 0$$

Máximo Interpretar la cadena que representa al máximo:

$$\mathcal{I}_{bss(2,0)}(11) = 2^1 + 2^0 = 2 + 1 = 3$$



Rango

Calcular el rango del sistema $BSS(4,2)$

Rango

Calcular el rango del sistema $BSS(4, 2)$

$\Downarrow$

Mínimo Interpretar la cadena que representa al mínimo:

$$\mathcal{I}_{bss(4,2)}(0000) = 0$$

Rango

Calcular el rango del sistema $BSS(4, 2)$

Mínimo Interpretar la cadena que representa al mínimo:

$$\mathcal{I}_{bss(4,2)}(0000) = 0$$

Máximo Interpretar la cadena que representa al máximo:

$$\mathcal{I}_{bss(4,2)}(1111) = 2^1+2^0+2^{-1}+2^{-2} = 3+0,75 = 3,75$$

Rango

El rango del sistema $BSS(4,2)$ es $[0 : 3{,}75]$

Rango

El rango del sistema $BSS(4,2)$ es $[0 : 3{,}75]$

¿Esto implica que pueden representarse todos los números en ese intervalo?

Rango

El rango del sistema $BSS(4, 2)$ es $[0 : 3{,}75]$

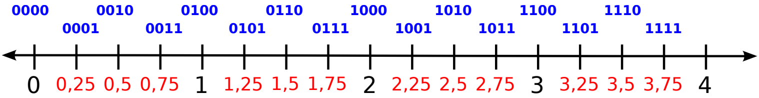
⇓

¿Esto implica que pueden representarse todos los números en ese intervalo?

⇓

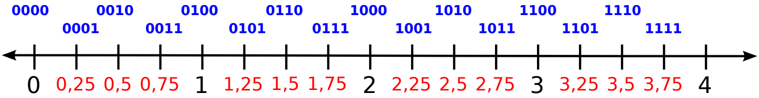
¡NO!

Rango

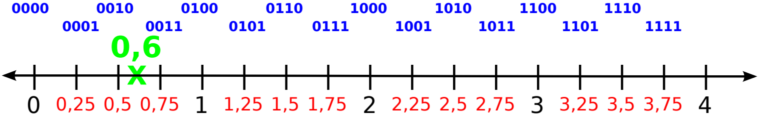


Por ejemplo: 0,6

Rango



Por ejemplo: 0,6



No es representable

Resolución

- En los sistemas enteros, los números representados van de uno en uno.
- En el sistema del ejemplo $BSS(2,1)$, ¿Que distancia tienen?

Resolución

- En los sistemas enteros, los números representados van de uno en uno.
- En el sistema del ejemplo $BSS(2,1)$, ¿Que distancia tienen?

cadena	número representado
00	0
01	0,5
10	1
11	1,5

Resolución

- En los sistemas enteros, los números representados van de uno en uno.
- En el sistema del ejemplo $BSS(2,1)$, ¿Que distancia tienen?

cadena	número representado
00	0
01	0,5
10	1
11	1,5

Los números van *saltando* de 0,5 en 0,5.

Resolución

- En los sistemas enteros, los números representados van de uno en uno.
- En el sistema del ejemplo $BSS(2,1)$, ¿Que distancia tienen?

cadena	número representado
00	0
01	0,5
10	1
11	1,5

Los números van *saltando* de 0,5 en 0,5.

Diremos entonces que la **resolución del sistema** es 0,5.

Resolución

distancia entre dos números representables consecutivos. Nos da una idea de precisión.

Resolución

Ejemplo

En el sistema $BSS(3,1)$, ¿Que distancia tienen?

Resolución

Ejemplo

En el sistema $BSS(3,1)$, ¿Que distancia tienen?

cadena	número representado
000	0
001	0,5
010	1
011	1,5
100	2
101	1,5
110	3
110	3,5

Resolución

Ejemplo

En el sistema $BSS(3,1)$, ¿Que distancia tienen?

cadena	número representado
000	0
001	0,5
010	1
011	1,5
100	2
101	1,5
110	3
110	3,5

Resolución: 0,5

Resolución

Ejemplo

En el sistema $BSS(3,2)$, ¿Que distancia tienen?

Resolución

Ejemplo

En el sistema $BSS(3,2)$, ¿Que distancia tienen?

cadena	número representado
000	0
001	0,25
010	0,5
011	0,75
100	1
101	1,25
110	1,5
110	1,75

Resolución

Ejemplo

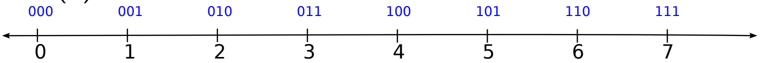
En el sistema $BSS(3,2)$, ¿Que distancia tienen?

cadena	número representado
000	0
001	0,25
010	0,5
011	0,75
100	1
101	1,25
110	1,5
110	1,75

Resolución: 0,25

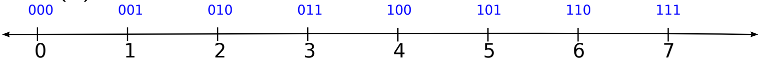
Resolución

- $BSS(3)$

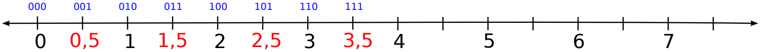


Resolución

- $BSS(3)$

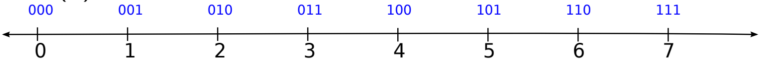


- $BSS(3,1)$

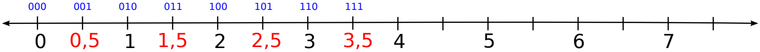


Resolución

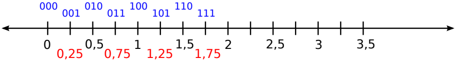
- $BSS(3)$



- $BSS(3,1)$



- $BSS(3,2)$



Ejemplos de interpretación BSS en 5 bits:

Ejemplo

Sistema $BSS(5,2)$

$00000 \rightarrow 0$
$00001 \rightarrow 2^{-2} = 0,25$
$00010 \rightarrow 2^{-1} = 0,5$
$00011 \rightarrow 2^{-1} + 2^{-2} = 0,75$
$00100 \rightarrow 2^{0} = 1$
$00101 \rightarrow 2^{0} + 2^{-2} = 1,25$
...
$01111 \rightarrow 2^{1} + 2^{0} + 2^{-1} + 2^{-2} = 3,75$
$10000 \rightarrow 2^{2} = 4$
$10001 \rightarrow 2^{2} + 2^{-2} = 4,25$
$10010 \rightarrow 2^{2} + 2^{-1} = 4,5$
$10011 \rightarrow 2^{2} + 2^{-1} + 2^{-2} = 4,75$
...
$11111 \rightarrow 2^{2} + 2^{1} + 2^{0} + 2^{-1} + 2^{-2} = 7,75$

Resolución

Calcular la resolución de los sistemas

- $BSS(8, 1)$
- $BSS(6, 4)$
- $BSS(16, 8)$

¿Cómo representar?

Representación en $BSS(n, m)$: dos métodos

(Separando partes)

La parte Entera: como en $BSS(n-m)$.
La parte Fraccionaria: con el método de las multiplicaciones sucesivas

(Corriendo la coma)

Correr el punto fraccionario para poder utilizar la representación en $BSS(n)$

Representación en $BSS(n, m)$: dos métodos

(Separando partes)

La parte Entera: como en $BSS(n - m)$.
La parte Fraccionaria: con el método de las multiplicaciones sucesivas

Representación en $BSS(n, m)$

(a) Parte Entera: ⇢ $BSS(n)$.

(b) Parte Fraccionaria:

1. Multiplicar la parte fraccionaria por 2
2. Reservar la parte entera del resultado
3. Si ya hicimos m+1 multiplicaciones, redondear, sino volver al paso 1
4. **Redondeo**: sumar **el bit obtenido en el último paso** a la cadena completa.

Representación en $BSS(n, m)$

(a) Parte Entera: ⇝ $BSS(n)$.

(b) Parte Fraccionaria:

1. Multiplicar la parte fraccionaria por 2
2. Reservar la parte entera del resultado
3. Si ya hicimos m+1 multiplicaciones, redondear, sino volver al paso 1
4. **Redondeo**: sumar **el bit obtenido en el último paso** a la cadena completa.

Ejemplo

Representar X= 3,14 en $BSS(7, 4)$

Representación en $BSS(n, m)$

(a) Parte Entera: ⇢ $BSS(n)$.

(b) Parte Fraccionaria:

1. Multiplicar la parte fraccionaria por 2
2. Reservar la parte entera del resultado
3. Si ya hicimos m+1 multiplicaciones, redondear, sino volver al paso 1
4. **Redondeo**: sumar **el bit obtenido en el último paso** a la cadena completa.

Ejemplo

Representar X= 3,14 en $BSS(7, 4)$

- Parte entera: $\mathcal{R}_{bss(3)}(3) = 011$

Representación en $BSS(n, m)$

(a) Parte Entera: ⇢ $BSS(n)$.

(b) Parte Fraccionaria:

1. Multiplicar la parte fraccionaria por 2
2. Reservar la parte entera del resultado
3. Si ya hicimos m+1 multiplicaciones, redondear, sino volver al paso 1
4. **Redondeo**: sumar **el bit obtenido en el último paso** a la cadena completa.

Ejemplo

Representar X= 3,14 en $BSS(7, 4)$

- Parte entera: $\mathcal{R}_{bss(3)}(3) = 011$
- Parte Fraccionaria:
  0,14*2 = 0,28
  0,28*2 = 0,56
  0,56*2 = 1,12
  0,12*2 = 0,24
  0,24*2 = 0,48

Representación en $BSS(n, m)$

(a) Parte Entera: ⇢ $BSS(n)$.

(b) Parte Fraccionaria:

1. Multiplicar la parte fraccionaria por 2
2. Reservar la parte entera del resultado
3. Si ya hicimos m+1 multiplicaciones, redondear, sino volver al paso 1
4. **Redondeo**: sumar **el bit obtenido en el último paso** a la cadena completa.

Ejemplo

Representar X= 3,14 en $BSS(7, 4)$

- Parte entera: $\mathcal{R}_{bss(3)}(3) = 011$
- Parte Fraccionaria:
  0,14*2 = 0,28
  0,28*2 = 0,56
  0,56*2 = 1,12
  0,12*2 = 0,24
  0,24*2 = 0,48

$$\text{Redondeo:} \quad \begin{array}{r} 0110010 \\ + \quad 0000000 \\ \hline 0110010 \end{array}$$

Representación en $BSS(n, m)$

Ejemplo

Representar 6,625 en $BSS(8, 4)$

- Parte entera: $\mathcal{R}_{bss(4)}(6) = 0110$

Representación en $BSS(n, m)$

Ejemplo

Representar 6,625 en $BSS(8, 4)$

- Parte entera: $\mathcal{R}_{bss(4)}(6) = 0110$
- Parte Fraccionaria:
  $0{,}625 * 2 = 1{,}250$
  $0{,}250 * 2 = 0{,}500$
  $0{,}500 * 2 = 1{,}000$
  $0{,}000 * 2 = 0{,}000$
  $0{,}000 * 2 = 0{,}000$
- Se componen las cadenas: 01101010

Representación en $BSS(n, m)$: dos métodos

(Corriendo la coma)

Correr el punto fraccionario para poder utilizar la representación en $BSS(n)$

Representación en $BSS(n, m)$: Corriendo la coma

Correr el punto fraccionario para poder utilizar la representación en $BSS(n)$

1. Multiplicar al número X que se quiere representar por 2^m
2. Redondear el número obtenido (**X'**)al entero más cercano (**X"**).
3. Representar **X"** en $BSS(n)$.

Representación en $BSS(n, m)$

1. Multiplicar al número X que se quiere representar por 2^m
2. Redondear el número obtenido (**X'**)al entero más cercano (**X"**).
3. Representar **X'** en $BSS(n)$.

Ejemplo

Representar X= 3,14 en $BSS(7, 4)$

Representación en $BSS(n, m)$

1. Multiplicar al número X que se quiere representar por 2^m
2. Redondear el número obtenido (**X'**)al entero más cercano (**X"**).
3. Representar **X'** en $BSS(n)$.

Ejemplo

Representar X= 3,14 en $BSS(7, 4)$

- $X * 2^4 = 50,24 = X'$

Representación en $BSS(n, m)$

1. Multiplicar al número X que se quiere representar por 2^m
2. Redondear el número obtenido (**X'**)al entero más cercano (**X"**).
3. Representar **X'** en $BSS(n)$.

Ejemplo

Representar X= 3,14 en $BSS(7, 4)$

- $X * 2^4 = 50,24 = X'$
- Redondeo: $X' \approx 50 = X''$
- $\mathcal{R}_{bss(7)}(50) = 0110010$

Representación en $BSS(n, m)$

1. Multiplicar al número X que se quiere representar por 2^m
2. Redondear el número obtenido (**X'**)al entero más cercano (**X"**).
3. Representar **X"** en $BSS(n)$.

Ejemplo

Representar X= 6,625 en $BSS(8, 4)$

Representación en $BSS(n, m)$

1. Multiplicar al número X que se quiere representar por 2^m
2. Redondear el número obtenido (**X'**)al entero más cercano (**X"**).
3. Representar **X"** en $BSS(n)$.

Ejemplo

Representar X= 6,625 en $BSS(8, 4)$

- $X * 2^4 = 106 = X'$

Representación en $BSS(n, m)$

1. Multiplicar al número X que se quiere representar por 2^m
2. Redondear el número obtenido (**X'**)al entero más cercano (**X"**).
3. Representar **X"** en $BSS(n)$.

Ejemplo

Representar X= 6,625 en $BSS(8, 4)$

- $X * 2^4 = 106 = X'$
- Redondeo: $X' \approx 106 = X''$
- $\mathcal{R}_{bss(8)}(106) = 01101010$

¿Como controlar el resultado obtenido?

¿Como controlar el resultado obtenido?

¡Interpretando!

Representación de 3,14 en $BSS(7,4)$ es 0110010

Representación de 3,14 en $BSS(7,4)$ es 0110010

$$\mathcal{I}_{bss(7,4)}(0110010) = 2^1 + 2^0 + 2^{-3} = 3,125$$

Representación de 3,14 en $BSS(7,4)$ es 0110010

$$\mathcal{I}_{bss(7,4)}(0110010) = 2^1 + 2^0 + 2^{-3} = 3,125$$ ✗

Representación de 3,14 en $BSS(7,4)$ es 0110010

$$\mathcal{I}_{bss(7,4)}(0110010) = 2^1 + 2^0 + 2^{-3} = 3,125$$ ✗

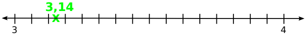
¿Porqué no obtuvimos 3,14?

Representación de 3,14 en $BSS(7,4)$ es 0110010

$$\mathcal{I}_{bss(7,4)}(0110010) = 2^1 + 2^0 + 2^{-3} = 3,125 \quad \times$$

¿Porqué no obtuvimos 3,14?

¡No es representable!



Aproximación

Si $3{,}14$ no es representable, se obtiene una aproximación

Aproximación

Si 3,14 no es representable, se obtiene una aproximación

$$3,125 \approx 3,14$$

Error por aproximación

Ejercicio: representar 0,4 en $BSS(2,1)$

Error por aproximación

Ejercicio: representar 0,4 en $BSS(2,1)$

- $\mathrm{X} = 0{,}4 \ \mathrm{m} = 1$
- $X' = 0,4x2^1 = 0,8$
- $\mathcal{R}_{bss(3)}(1) = 001$

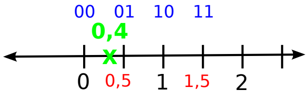
Error por aproximación

Ejercicio: representar 0,4 en $BSS(2,1)$

- X = 0,4 m = 1
- $X' = 0,4x2^1 = 0,8$
- $\mathcal{R}_{bss(3)}(1) = 001$

Comprobar el resultado

$$\mathcal{I}_{bss(2,1)}(001) = 2^{-1} = 0,5$$



¡~~H~~errar es humano!

Error Absoluto

Error Absoluto
Valor absoluto de la diferencia entre el número que se quiere representar y el número representado.

$$EA(X) = |X - E|$$

donde E es el valor de la Representación más próxima a X.

- En este caso $EA(0,4) = |0,4 - 0,5| = 0,1$.

Error Absoluto

- El error absoluto de un número representable es 0 ya que no hay diferencia entre el número que se quiere representar y su Representación.
- Notar que el Error Absoluto tiene como cota máxima la mitad de la resolución:

$$(\forall X \in rango) 0 \leq EA(X) \leq R/2$$

Error Relativo

Ejemplo

Representar el número 14,9 en $BSS(8,4)$:

1. $\mathrm{X} = 14{,}9 \;\; \mathrm{m} = 4$
2. $X' = 14,9x2^4 = 238,4$
3. $\mathcal{R}_{bss(8)}(238) = 11101110$
4. $\mathcal{I}_{bss(8,4)}(11101110) = 14,875$

$$EA(14,9) = |14,9–14,875| = 0,025$$

El error cometido al representar 14,9 es el mismo que al representar 3,9. Sin embargo el valor del error 0,025 es “más importante” al representar 3,9 que al representar 14,9.

Error Relativo

Error Relativo

$$ER(X) = |\frac{EA(X)}{X}| (\forall X \neq 0 \in rango)$$

$$ER(3,9) = |\frac{EA(3,9)}{3,9}| = |0,025/3,9| \approx 0,0064 = 0,64\,\%$$

$$ER(14,9) = |\frac{EA(14,9)}{14,9}| = |0,025/14,9| \approx 0,0016 = 0,16\,\%$$

El error relativo al representar 14,9 es menor que el error relativo al representar 3,9

Error Absoluto y Relativo

Calcular error absoluto en $BSS(8,4)$

- 1,1
- 2,125
- 3,099
- 4,75
- 19,99

Error Absoluto y Relativo

Calcular error relativo en $BSS(8,4)$

- 0,1
- 15,1

Error Relativo

- El error relativo más grande se produce al representar números muy cercanos a cero, para los cuales el sistema los representa como 0. Para estos números, EA(X) $=$ X, y ER(X) $= 1 = 100\,\%$.
- Los errores relativos más pequeños se producen en el extremo superior del rango.

1. Operaciones lógicas y máscaras
   - Máscaras
2. Punto Fijo
   - Resolución
   - Representación