Introducción al Diseño Lógico (E0301)

Ingeniería en Computación

Gerardo Sager

Clase 2 curso 2021

Objetivos

- *Temas que se desarrollan*:
  - Convertir entre sistemas numéricos
    - Decimal, binario, hexadecimal.
  - Ventajas del uso de hexadecimal
    - Conteo en hexadecimal.
  - Representación de números decimales usando código BCD.
  - Códigos Alfanuméricos: código ASCII.
  - Paridad y su utilización para la detección de errores.

Conversión Binaria a Decimal

- Convertir de binario a decimal sumando las posiciones que contienen 1 con su peso correspondiente:

$$11011_2$$

$$11011_2 \quad =1x2^4+1x2^3+0x2^2+1x2^1+1x2^0= =16+8+2+1 = \color{red}{27}$$

- Un ejemplo con más bits

$$10110101_2 =$$

- $10110101_2 = 1x2^7+1x2^5+1x2^4+1x2^2+1x2^0 = 128+32+16+4+1=\color{red}{181}$

Conversión Binaria a Decimal

- Los números binarios pueden convertirse con el método “double-dabble”

$$
\begin{array}{llllll}
\text{Given:} & 1 & 1 & 0 & 1 & 1_2 \\
\text{Results:} & 1 \times 2 = 2 & & & & \\
 & \quad + 1 & & & & \\
 & & 3 \times 2 = 6 & & & \\
 & & \quad + 0 & & & \\
 & & & 6 \times 2 = 12 & & \\
 & & & \quad +1 & & \\
 & & & & 13 \times 2 = 26 & \\
 & & & & \quad +1 & \\
 & & & & & 27_{10}
\end{array}
$$

El método double-dabble permite la conversión sin hacer la suma de grandes números

- Proceso inverso a la suma de potencias pesadas.
- Debo tener una tabla de potencias de 2 expresadas en decimal.
- Me fijo cual es el mayor número de la tabla, contenido en el número a convertir.
- resto del número a convertir el valor obtenido
- repito el proceso hasta llegar a 0
- establezco en 1 el bit correspondiente si usé el valor de la tabla y en 0 si no lo usé

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Ejemplo: 314_{10} = ?

314-256=58 (2^8)   58-32=26 (2^5)

26-16=10 (2^4) 10-8=2 (2^3)

2-2=0 (2^1)

$314_{10}=100111010_2$

División Repetida

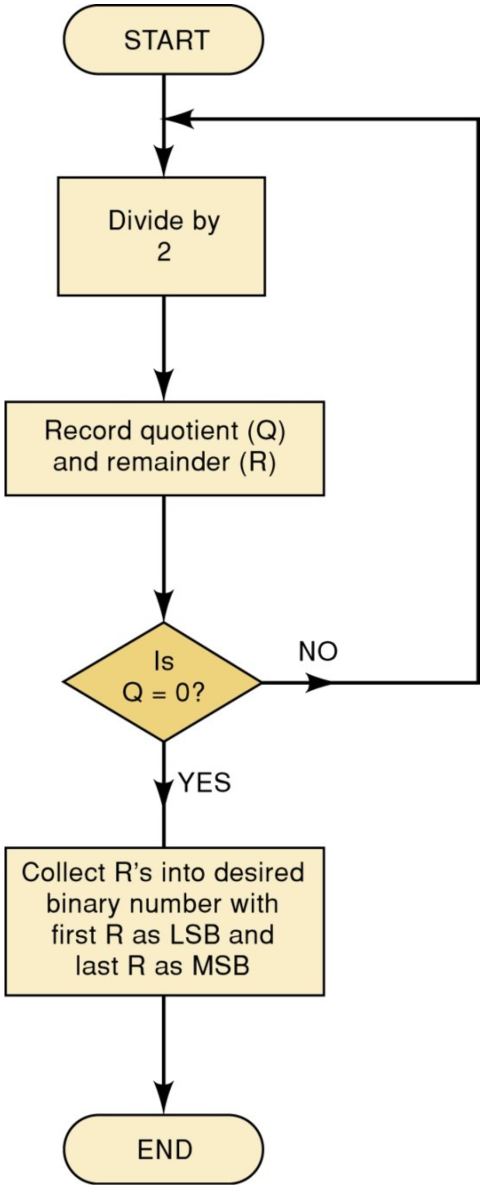
- Dividir el número decimal por 2.
- Escribir el resto de cada división y volver a dividir el cociente x 2 hasta que se obtenga un cociente = 0
- El primer resto es el LSB
- El último resto es el MSB

314/2	Q=157	R=0 (LSB)
157/2	Q=78	R=1
78/2	Q=39	R=0
39/2	Q=19	R=1
19/2	Q=9	R=1
9/2	Q=4	R=1
4/2	Q=2	R=0
2/2	Q=1	R=0
1/2	Q=0	R=1 (MSB)

$314_{10} = 100111010_2$

División Repetida

- Este diagrama de flujo describe el proceso a realizar para convertir de Decimal a Binario.
- Si en vez de dividir por 2 se divide por N, me permite convertir de decimal a base N



Sistema de Numeración Hexadecimal

- Hexadecimal permite un manejo conveniente de cadenas binarias largas, ya que 4 bits agrupados pueden representarse como un único dígito hexadecimal.(base 16)
- Los símbolos que representan los dígitos hexadecimales, son 0-9 y A-F
- La representación puede ser entera o fraccionaria.

16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}

Hexadecimal point

Relación entre números hexadecimales, decimales, y binarios.

Hexadecimal	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Sistema de Numeración Hexadecimal– Hexa a Decimal

- Convertir de hexa a decimal multiplicando cada dígito por su peso posicional.

$356_{16} = 3x256+5x16+6x1=$

- En el siguiente ejemplo, A se sustituye por 10 y F por 15 para realizar la suma pesada.

$2AF_{16} = 2x256+10x16+15$

Para practicar, verificar que $1BC2_{16}$ is igual a ….

- Se pueden usar varios métodos: convertir a binario y luego agrupar en dígitos hexadecimales.
- Usar el método de división repetida
  - Dividir el número decimal por 16
  - El primer resto es el LSB
  - El último es el MSB.
- Ejemplo:
- 100000_{10}

Sistema de Numeración Hexadecimal – Hexa a Binario

- Se convierte cada dígito Hexa directamente a binario y se reescribe.
- Si es necesrio se pueden agregar ceros a la izquierda para completar una palabra
- $19F2_{16}$ =0001 1001 1111 0010

Para practicar verificar que $BA6_{16} = 101110100110_2$

- Convertir de binario a Hexa, agrupando bits de a cuatro comenzando por el LSB.
  - Luego cada grupo se escribe como un único digito hexa equivalente

$$1\,1\,1\,0\,1\,0\,0\,1\,1\,0_2 = \underbrace{0\;0\;1\;1}_{3}\;\underbrace{1\;0\;1\;0}_{A}\;\underbrace{0\;1\;1\;0}_{6}$$

$$= 3A6_{16}$$

Verificar que 101011111 $_2$ = …

Codificación BCD

- La codificación BCD se usa ampliamente, sobre todo en equipos de bajo costo, para representar números decimales en forma binaria
  - Combina características de los sistemas decimal y binario.
    - Cada dígito decimal se convierte a su equivalente binario
- BCD **NO ES** un sistema numérico.
  - Es un número decimal, con cada dígito codificado como su equivalente en binario.
- Un número BCD no es lo mismo que un número binario.
  - La ventaja principal de BCD es que resulta fácil convertir de BCD a decimal y viceversa.

Codificación BCD

- Convertir el número 874_{10} a BCD:
  - Cada dígito se representa mediante 4 bits.
  - El ´numero representado mediante esos 4 bits no puede ser superior a 9

1	0	0	0	0	1	1	1	0	1	0	0

$$\underbrace{1\,0\,0\,0}_{\mathbf{8}}\;\underbrace{0\,1\,1\,1}_{\mathbf{7}}\;\underbrace{0\,1\,0\,0}_{\mathbf{4}}$$

- Para convertir de BCD a decimal se realiza el proceso inverso:
  - Se agrupan los bits de a 4.
  - Se determina el valor decimal de cada grupo de 4 bits, que deben estar comprendidos entre 0 y 9

- Comparación entre BCD y Binario.
  - Cual es el rango de valores que pueden representarse en BCD con 16 bits?
  - Cual es el rango de valores que pueden representarse en binario con 16 bits?
- Convertir 0110100000111001 a su equivalente decimal considerando primero que es BCD y luego que es binario.
  - Cuál de las dos conversiones es mas sencilla?

Byte, Nibble, Word

- La mayoría de las Microcomputadoras maneja y almacena datos binarios en grupos de ocho bits.
  - 8 bits = 1 **byte**.
    - Un byte puede representar distintos tipos de datos o información.
- Frecuentemente los números binarios se agrupan en grupos de cuatro bits.
  - Un grupo de cuatro bits suele llamarse **nibble** (del inglés nibble = mordisquito).
- **Word** o **palabra** es un grupo de bits que representa una cierta unidad de información.
  - El **tamaño de palabra (Word size)** puede ser definido de distintas maneras en distintos sistemas. Lo más comun es definirlo como la cantidad de bits en la palabra que utiliza el sistema digital para operar.
  - La longitud de palabra en una PC actual es 64 bits.
  - La longitud de palabra en un 80x86 es 16 bits.
  - La longitud de palabra en un sistema basado en ARM es 32 bits.

Códigos alfanuméricos

- Representan caracteres, dígitos, símbolos y valores que pueden interpretarse como comandos o funciones.
- Normalmente se encuentran en el teclado de una computadora.
- Hace falta definir al menos 26 minúsculas, 26 Mayúsculas, 10 dígitos, 7 signos de puntuación y entre 20 y 40 caracteres adicionales.
- En distintos idiomas, puede ser necesario agregar más letras o símbolos
- ASCII : Está definido con 128 valores que pueden representarse con 7 bits.
- ASCII Extendido: Posee características que se adaptan a distintos idiomas y consta de 256 valores que pueden representarse con 8 bits.

ASCII

Character	HEX	Decimal	Character	HEX	Decimal	Character	HEX	Decimal	Character	HEX	Decimal
NUL (null)	0	0	Space	20	32	@	40	64	.	60	96
Start Heading	1	1	!	21	33	A	41	65	a	61	97
Start Text	2	2	“	22	34	B	42	66	b	62	98
End Text	3	3	#	23	35	C	43	67	c	63	99
End Transmit.	4	4	$	24	36	D	44	68	d	64	100
Enquiry	5	5	%	25	37	E	45	69	e	65	101
Acknowlege	6	6	&	26	38	F	46	70	f	66	102
Bell	7	7	`	27	39	G	47	71	g	67	103
Backspace	8	8	(	28	40	H	48	72	h	68	104
Horiz. Tab	9	9	)	29	41	I	49	73	i	69	105
Line Feed	A	10	*	2A	42	J	4A	74	j	6A	106
Vert. Tab	B	11	+	2B	43	K	4B	75	k	6B	107
Form Feed	C	12	,	2C	44	L	4C	76	l	6C	108
Carriage Return	D	13	-	2D	45	M	4D	77	m	6D	109
Shift Out	E	14	.	2E	46	N	4E	78	n	6E	110

2-8 Alphanumeric Codes

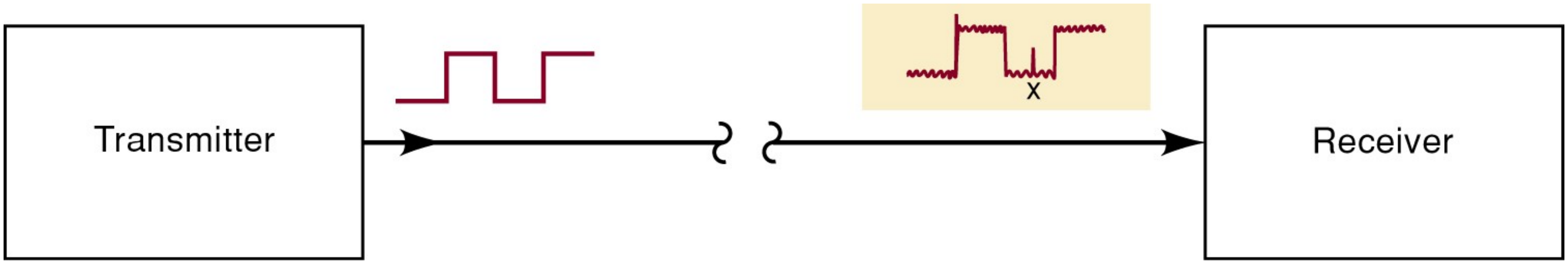
- Convertir la siguiente sentencia de C a codigo ASCII. Escribir en decimal, hexadecimal y binario

if (x>3)

Ascii	**i**	**f**		**(**	**x**	**>**	**3**	**)**
Hexade-cimal	***69***	***66***	***20***	***28***	***78***	***3E***	***33***	***29***
Decimal	***109***	***102***	***32***	***40***	***120***	***62***	***51***	***41***
Binario	***01101001***	***01100110***	***00100000***	***00101000***	***01111000***	***00111110***	***00110011***	***00101001***

2-9 Método de la Paridad para detectar errores

- El ruido eléctrico puede causar errores durante la transmisión.
  - Ruido? Fluctuaciones espurias y aleatorias en el voltaje o corriente. Aparecen en TODOS los sistemas electrónicos
  - NO puede eliminarse



- Muchos sistemas digitales emplean métodos para la detección e errores y algunas veces también su corrección.
- Uno de los sistemas de control de errores más simples y más utilizados es el método de PARIDAD.

2-9 Parity Method for Error Detection

- El método de paridad para la detección de errores requiere la adición de un bit extra al grupo de bits a transmitir.
  - Es llamado el bit de paridad.
  - Puede ser 0 o 1, dependiendo del número de 1s en el grupo de bits a transmitir..
- Hay dos métodos de paridad, PARIDAD PAR y PARIDAD IMPAR.
  - El transmisor y el receptor deben utilizar el mismo tipo de paridad.

Metodos de Paridad para detección de errores

- MÉTODO DE PARIDAD PAR:
- Se basa en completar el grupo de bits a transmitir con el bit de paridad de manera tal que la cantidad total de "1" que tenga la el conjunto sea PAR.
  - La cantidad de bits en 1 del (grupo de bits a transmitir + bit de paridad) debe ser par.
- Se cuenta la cantidad de "1" que tiene la palabra
- Si la cantidad es par, el bit de paridad se establece en 0
- Si la cantidad es impar el bit de paridad se establece en 1
- En ambos casos la cantidad TOTAL de bits en "1" debe ser par
- Ejemplo:

Grupo de bits a transmitir: 1000011

Bit de Paridad Par?

Metodos de Paridad para detección de errores

- MÉTODO DE PARIDAD IMPAR:
- Se basa en completar el grupo de bits a transmitir con el bit de paridad de manera tal que la cantidad total de "1" que tenga la el conjunto sea IMPAR.
  - La cantidad de bits en 1 del (grupo de bits a transmitir + bit de paridad) debe ser impar.
- Se cuenta la cantidad de "1" que tiene la palabra
- Si la cantidad es par, el bit de paridad se establece en 1
- Si la cantidad es impar el bit de paridad se establece en 0
- En ambos casos la cantidad TOTAL de bits en "1" debe ser IMPAR
- Ejemplo:

  Grupo de bits a transmitir: 1000011

  Bit de Paridad Impar?

Metodos de Paridad para detección de errores

- En ambos casos el bit de paridad se AGREGA al grupo original de bits a transmitir.
- El grupo de bits junto con la paridad, constituye la palabra de CÓDIGO.
- Por ejemplo si quiero transmitir palabras codificadas en ASCII con paridad PAR, la palabra de código tendrá 8 bits, 7 de los datos en ASCII y uno de paridad.

- Codificar HOLA MUNDO en ASCII
- Agregar los bits de paridad PAR (como MSB) correspondientes a cada símbolo ASCII.(PP)
- Repetir para paridad impar (PI)

H	O	L	A		M	U	N	D	O	
48	4F	4C	41	20	4D	55	4E	44	4F	ASCII
48	CF	CC	41	A0	4D	55	4E	44	CF	ASCII + PP
C8	4F	4C	C1	20	CD	D5	CE	C4	4F	ASCII + PI

FIN