



ESCUELA DE CIENCIAS BÁSICAS E INGENIERÍA

ASIGNATURA: Arquitectura de Computadores

CORPORACIÓN UNIVERSITARIA REMINGTON
DIRECCIÓN PEDAGÓGICA

Este material es propiedad de la Corporación Universitaria Remington (CUR), para los estudiantes de la CUR en todo el país.

2011

CRÉDITOS



El módulo de estudio de la asignatura Arquitectura de computadores es propiedad de la Corporación Universitaria Remington. Las imágenes fueron tomadas de diferentes fuentes que se relacionan en los derechos de autor y las citas en la bibliografía. El contenido del módulo está protegido por las leyes de derechos de autor que rigen al país.

Este material tiene fines educativos y no puede usarse con propósitos económicos o comerciales.

AUTOR

Álvaro de Jesús Laverde Quiróz

Tecnólogo en electrónica ITPB 1997, pregrado en electrónica (proyecto de grado USB) 2011, diplomado en diseño curricular y materiales de auto aprendizaje CUR, diplomado en web 2.0 UNAB. Investigador: Grupo GITECUR.

Disgráficas Ltda. soporte depto. Electrónico. Cede computo: Jefe de área electrónica. Cesde: Docente en electricidad.

Docente de la corporación Universitaria Remington, escuela de ciencias básicas.

alvaro.laverde@remington.edu.co

Nota: el autor certificó (de manera verbal o escrita) No haber incurrido en fraude científico, plagio o vicios de autoría; en caso contrario eximió de toda responsabilidad a la Corporación Universitaria Remington, y se declaró como el único responsable.

RESPONSABLES

Escuela de Ciencias Básicas e Ingeniería

Director Dr. Mauricio Sepúlveda

Director Pedagógico

Octavio Toro Chica

dirpedagogica.director@remington.edu.co

Coordinadora de Medios y Mediaciones

Angélica Ricaurte Avendaño

mediaciones.coordinador01@remington.edu.co

GRUPO DE APOYO

Personal de la Unidad de Medios y Mediaciones

EDICIÓN Y MONTAJE

Primera versión. Febrero de 2011.

Derechos Reservados

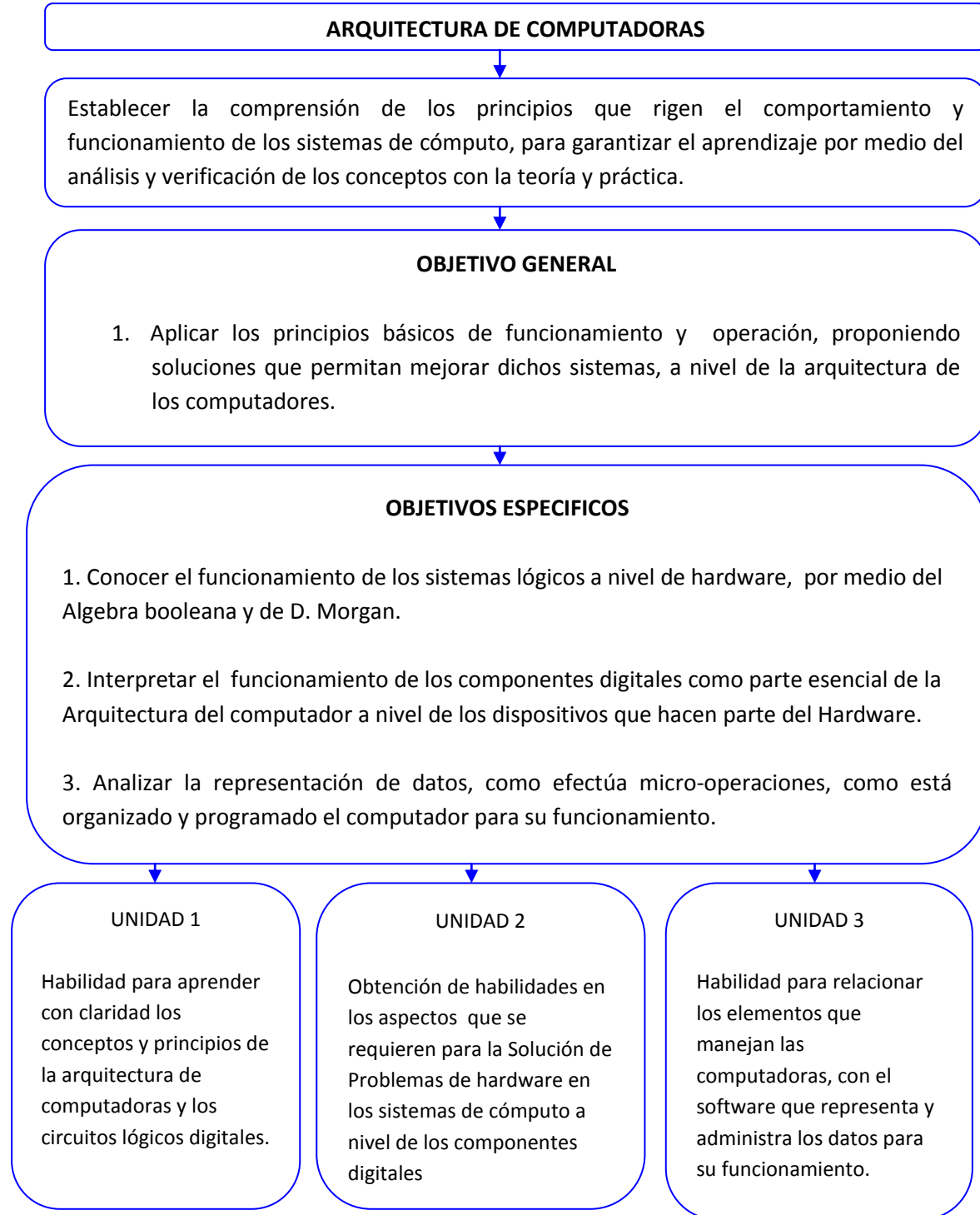


Esta obra es publicada bajo la licencia Creative Commons. Reconocimiento-No Comercial-Compartir Igual 2.5 Colombia.

TABLA DE CONTENIDO

1.	MAPA DE LA ASIGNATURA.....	7
2.	CIRCUITOS LOGICOS DIGITALES	8
2.1.	Compuertas digitales.....	9
2.2.	Algebra booleana	17
2.3.	Simplificación de expresiones Algebraicas.....	22
2.4.	Flip-Flops	28
3.	COMPONENTES DIGITALES.....	35
3.1.	Circuitos integrados	36
3.2.	Codificadores.....	41
3.3.	Multiplexores	50
3.4.	Demultiplexor.....	52
3.5.	Registros.....	54
3.6.	Registros de desplazamiento	56
3.7.	Unidad de memoria.....	58
4.	REPRESENTACION DE DATOS, TRANSFERENCIA DE REGISTROS Y MICROOPERACIONES, ORGANIZACIÓN BASICA Y PROGRAMACIÓN	61
4.1.	Representación de Datos	62
4.2.	Transferencia de registro y micro operaciones.....	64
4.3.	Organización básica del computador	67
4.4.	Programando el computador básico.....	73
4.5.	Pistas de Aprendizaje	79
4.6.	Glosario	80
4.7.	Bibliografía	81
4.8.	Fuentes digitales o electrónicas	81

1. MAPA DE LA ASIGNATURA



2. CIRCUITOS LOGICOS DIGITALES

Video Link introducción a las compuertas lógicas:

(<http://www.youtube.com/watch?v=1S3Y1Bz62Yw&feature=related>.Fecha:17072011))

Objetivo general

- ◆ Conocer el funcionamiento de los sistemas lógicos a nivel de hardware, por medio del Algebra booleana y de D. Morgan.

Objetivos específicos

- ◆ Analizar los circuitos lógicos, compuertas y componentes digitales.
- ◆ Efectuar representación de datos, intercambio de registros y micro-operaciones, la organización básica según códigos, instrucciones, parámetros, descripción y diseño usando el algebra booleana.
- ◆ Conocer las secuencias de diseño con la simplificación de expresiones algebraicas direccionamiento, control, programación básica y sus lenguajes.
- ◆ Comprender el funcionamiento lógico de los flip-flop y realizar implementaciones con dispositivos RS, Data, JK.

Prueba Inicial

- 1: Enuncie los principales sistemas numéricos que existen.
- 2: Que es un circuito digital.
- 3: Determine tres Familias de circuitos integrados.
- 4: Describa un sistema combi nacional.
- 5: Cuales son las características de un sistema digital secuencial.

2.1. Puertas digitales

Sistemas numéricos

Agrupación de símbolos y normas a través de los cuales se logra hacer acomodar los números válidos en el sistema.

$(b^n) * A$

Dónde:

b = valor de la base del sistema

n = número del dígito o posición del mismo

A = dígito.

Su representación puede ser:

$$N = S + A$$

$$S = A - N$$

$$A = S - N$$

N sistema de numeración (p.ej. decimal, binario, octal, hexadecimal etc.).

A reglas que nos indican qué números son válidos, y cuáles no lo son.

S Símbolos permitidos en el sistema. En el sistema:

Binario {0,1}.

Decimal {0,1...9}.

Octal {0,1...7}.

Hexadecimal {0,1...9, A, B, C, D, E, F}.

Conversión de sistemas numéricos:

Consiste en dar los valores numéricos equivalentes de un sistema en otro.

Para conocer el valor podemos utilizar:

a. De binario a octal: Separamos el numero binario en grupos de tres (3), de derecha a izquierda, luego damos el valor octal a cada grupo.

Ejemplo 1

Convertir 10010 de binario (b2) a octal (b8).

S/n: (010) (010) solo cuando sea necesario el último grupo de derecha a izquierda debe completarse con ceros para que cumpla el grupo de tres (3).

Luego para (010) (010)
2 2

Queda: El número 10010 en b2 = 22 en b8

b. De binario a hexadecimal: Separamos el numero binario en grupos de cuatro (4), de izquierda a derecha, luego damos el valor hexadecimal a cada grupo.

Ejemplo 2

Convertir 110010100 de binario (b2) a octal (b16).

S/n: (0001) (1001)(0100) cuando sea necesario el último grupo de derecha a izquierda debe completarse con ceros para que cumpla el grupo de cuatro (4).

Luego para (0001) (1001)(0100)
1 9 4

Queda: El número 110010100 en b2 = 194 en b16

c. De binario a decimal: Representamos cada valor binario uno (1) en su potencia equivalente de derecha a izquierda y sumamos las potencias.

Ejemplo 3:

Para el Valor binario 110010100 b2 convertirlo a decimal b10.

	256	128	64	32	16	8	4	2	1
S/n:	1	1	0	0	1	0	1	0	0

Luego sumamos $256+128+16+4=404$

Queda: 110010100 en b2 = 404 en b10

Compuertas digitales

Son arreglos de circuitos básicos a partir de representaciones seriales, paralelas y posibles combinaciones efectuadas con elementos que pueden poseer dos estados. Ej.: Cerrado ó uno (1), abierto o cero,

(0), Bajo o cero (0), alto o uno (1). Estos valores corresponden a los niveles de voltaje alto y bajo (1,0) de la alimentación o fuente del sistema.

Físicamente las compuertas lógicas están formadas por dispositivos transistores semiconductores, que son los encargados de comportarse como interruptor abierto o cerrado cuando recibe un estímulo de voltaje.

Compuertas lógicas digitales: Elementos electrónicos que representan físicamente un operador o expresión booleana según la lógica de conmutación. Son circuitos integrados de conmutación en un chip semiconductor.

Son: AND, OR, NAND, NOR, NOT, XOR.

Compuerta AND: Proviene de un circuito de conmutación con sus elementos en serie.

Es prioridad que todos los elementos estén cerrados para que pueda entregar salida en uno (1) lo demás es cero (0).

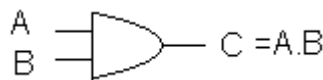


Fig.1 Compuerta AND.

Autor: Álvaro de J. Laverde. Q. (F.02-05-2011)

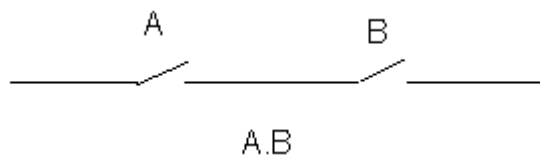


Fig.2 Representación AND con contactos

Autor: Álvaro de J. Laverde. Q. (F.02-05-2011)

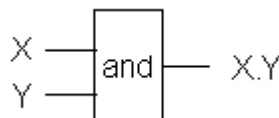


Fig.3 Símbolo AND

Autor: Álvaro de J. Laverde. Q. (F.02-05-2011)

Tabla de verdad: se usa para visualizar y presentar las funciones con compuertas poseen variables de entrada y salida. Las posibles combinaciones de valores binarios está dada por: 2^n con n igual al número de variables de entrada.

Posibles combinaciones de valores binarios 2^n con n=2.
Es igual a 4.

Tabla AND

Entradas		Salida
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

La expresión booleana o algebraica es. $A.B=F$ o $F= A.B$

Compuerta OR: Proviene de un circuito de conmutación con sus elementos en paralelo. Donde con un valor uno (1) en las variables de entrada la salida será uno (1) y es prioridad que cuando todos los elementos estén abiertos entrega cero (0) a la salida.

Representación simbólica:



Fig.4 Compuerta OR.
Autor: Álvaro de J. Laverde Q. (F: 02-05.2011)

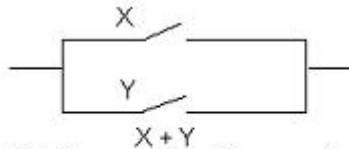


Fig.5 Representación OR con contactos
Autor: Álvaro de J. Laverde. Q. (F.02-05-2011)

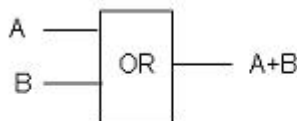


Fig.6 Símbolo OR
Autor: Álvaro de J. Laverde. Q. (F.02-05-2011)

Tabla OR

Entradas		Salida
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

La expresión booleana o algebraica es. $A+B=F$ o $F= A+B$

Compuerta NAND: Efectúa la operación de negado de producto lógico.

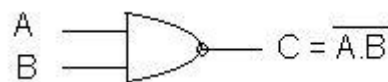


Fig.7: Compuerta NAND
Autor: Alvaro de J.Laverde Q. (F:02-05-2011)

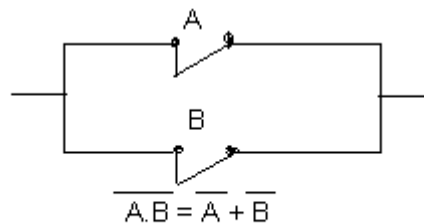


Fig.8: Representación NAND con contactos
Autor: Alvaro de J. Laverde Q. (F: 02-05-2011)

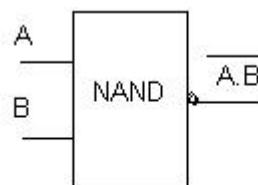


Fig.9: Simbolo NAND
Autor: Alvaro de J.Laverde Q. (F: 02-05-2011)

Tabla NAND

Entradas		Salida
A	B	(A.B)'
0	0	1
0	1	1
1	0	1
1	1	0

La expresión booleana o algebraica es. $(A.B)' = F$ ó $F = (A.B)'$

Compuerta NOR: Efectúa la operación de negado de suma lógica.

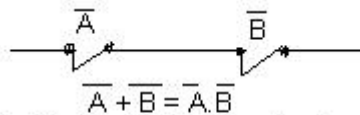


Fig.10: Circuito NOR con contactos
Autor: Alvaro de J.Laverde Q. (F:02-05-2011)

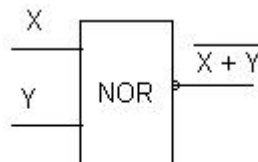


Fig.11: Símbolo NOR
Autor: Alvaro de J. Laverde Q. (F:02-05-2011)

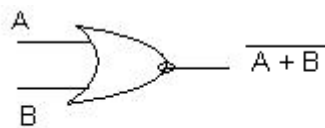


Fig.12: Compuerta lógica
Autor: Alvaro de J. Laverde Q. (F: 02-05-2011)

Tabla de verdad NOR

Entradas		Salida
A	B	$(A+B)'$
0	0	1
0	1	0
1	0	0
1	1	0

La expresión booleana o algebraica es. $(A+B)' = F$ ó $F = (A+B)'$

Compuerta NOT: Invierte o niega una variable lógica.

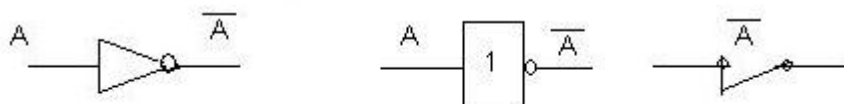


Fig.13: Compuerta NOT

Símbolo NOT

Circuito NOT

Autor: Alvaro de J. Laverde Q. (F: 02-05-2011)

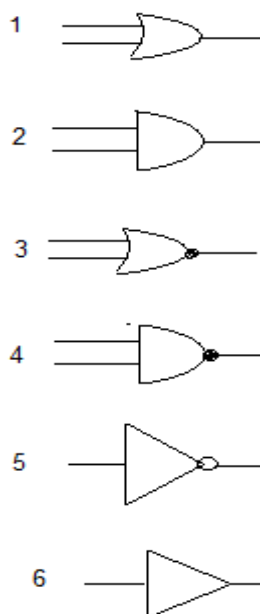
Tabla de verdad de compuerta NOT

Entrada	Salida
A	A'
0	1
1	0

$$F = A'$$

Ejemplo 4

Identifique las siguientes compuertas lógicas colocando el nombre al frente de cada uno de los siguientes símbolos.



Autor: Álvaro de J. Laverde Q.

Solución:

1= OR, 2 = AND, 3 = NOR, 4 = NAND, 5 = NOT, 6 = SEPARADOR YES.

Ejemplo 5

Diga a cual compuerta corresponde la siguiente tabla lógica.

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Solución:

Al observar las tablas correspondientes a los sistemas lógicos de las compuertas determinamos que es la tabla de la compuerta NAND ya que la salida es el inverso de la AND.

Significa que solo es cero cuando ambas entradas están en uno, y los demás valores son unos.

Ejercicios del tema 1: Compuertas digitales

- Para que se usen las compuertas digitales.
- Explique cómo se diferencian las compuertas digitales.
- Como se conocen las características básicas de las compuertas.
- Diseñe una tabla lógica que represente las compuertas.
- Diseñe un sistema que utilice las compuertas lógicas.

2.2. Algebra booleana

Es una representación simbólica o estructura algebraica que interrelaciona los operadores lógicos, el conjunto de operaciones complemento intersección y unión.

Por lo general para obtener la expresión algebraica booleana podemos partir de una tabla de lógica, un circuito lógico o un enunciado lógico.

$(A+B) \cdot C = F$ Esta es una expresión algebraica cualquiera.

Podemos observar el circuito equivalente.

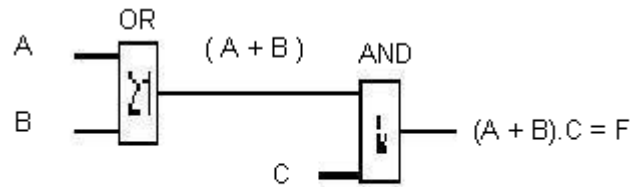


Fig.14: Representación $(A+B) \cdot C = F$
Autor: Alvaro de J. Laverde Q. (02-05-2011)

La tabla de verdad lógica, La cual tiene tres (3) variables: A, B, C, tiene 2 exp. n posibles combinaciones y contiene la expresión algebraica anterior.

A	B	C	(A+B)	.	C	F
0	0	0	0		0	0
0	0	1	0		1	0
0	1	0	1		0	0
0	1	1	1		1	1
1	0	0	1		0	0
1	0	1	1		1	1
1	1	0	1		0	0
1	1	1	1		1	1

Si vamos a pasar de una tabla a la expresión algebraica, debemos tomar las filas como operador and (.) y las columnas como operador (+), teniendo en cuenta que el cero (0) representa la variable complementada y el uno (1) representa la variable sin complemento, la salida corresponde a los requisitos del enunciado para la cual se pueden tomar los valores que están en uno (1) o viceversa como complemento.

Es recomendable tomar los valores de la salida que estén en menor cantidad para hacer más simplificado el proceso.

Ejemplo 1

Determine la expresión algebraica a partir de la salida de la siguiente tabla

W	X	Y	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Solución:

Tomamos los valores que en la salida estén en menor cantidad, para este caso los unos. Sacamos la variable para cada valor empezando de izquierda a derecha o viceversa y el primer uno de F, de arriba hacia abajo así:

F	Términos
0	
1	$W'X'Y$
0	
0	
1	$W'X Y$
0	
0	
1	$W X Y$

Luego debemos unir con el operador or (+) todos los términos $F=W'X'Y+W'XY+WXY$

El álgebra booleana utiliza éstos postulados o teoremas para simplificar o reducir expresiones de tal manera que sea más comprensible, al igual que los teoremas 7, 8 en honor al matemático que los descubrió llamado De Morgan.

Teoremas:

1: $A + A = A$

2: $A \cdot A = A$

3: $A + 0 = A$

4: $A \cdot 1 = A$

5: $A \cdot 0 = 0$

6: $A + 1 = 1$

7: $(A + B)' = A' \cdot B'$

8: $(A \cdot B)' = A' + B'$

9: $A + A \cdot B = A$

10: $A \cdot (A + B) = A$

11: $A + A'B = A + B$

12: $A' \cdot (A + B') = A'B'$

13: $AB + AB' = A$

14: $(A' + B') \cdot (A' + B) = A'$

15: $A + A' = 1$

16: $A \cdot A' = 0$

Su demostración se hace a partir de operaciones con las compuertas reemplazando las variables con cero(0) o uno (1) en las entradas si la variable vale 1 ó 0 se deja fija ya que será una variable que vale 1 ó 0 según el caso.

Demostración:

Para el caso del teorema 3.

Teorema $A+0 = A$

A puede valer 0 ó 1

Luego, reemplazando a A tenemos $0+0=0$, $1+0=1$ significa que A siempre va a tomar el valor con el cual se reemplace la variable A, por lo que A puede valer 0 ó 1.

Ejemplo 2

Aplicar los teoremas para resolver la función lógica de la siguiente tabla lógica:

X	Y	F
0	0	0
0	1	0
1	0	1
1	1	1

Solución:

Podemos iniciar con los ceros o con los unos porque está en la función de salida F en igual cantidad, para el primer valor de cero equivale a $X'Y'$ y para el segundo valor de 0 tenemos $X'Y$, además para el tercer valor de 1 es XY' y para el cuarto valor de 1 sería XY , entonces usando los dos valores que están 1 queda que la expresión es:

$$F = XY' + XY$$

Sacando factor común,

$$F = X(Y' + Y)$$

Aplicando el teorema 13

$$F = X$$

Ejemplo 3

Determine la salida de la siguiente expresión algebraica $F = X'Y'Z' + X'Y'Z + X'Y'Z$

Solución:

Primero extraemos el factor común,

$$X' Y' (Z' + Z + Z) = F$$

Luego aplicamos el teorema 1.

$$X' Y' (Z' + Z) = F$$

Usando el teorema 15 en el paréntesis,

$$X' Y' (1) = F$$

$$X' Y' = F$$

Ejemplo 4

Realizar la tabla para la expresión algebraica siguiente:

$$X = ABC' + A.B + ABC$$

Solución:

A	B	C	ABC'	+	AB	+	ABC	X
0	0	0	0		0		0	0
0	0	1	0		0		0	0
0	1	0	0		0		0	0
0	1	1	0		0		0	0
1	0	0	0		0		0	0
1	0	1	0		0		0	0
1	1	0	1		1		0	1
1	1	1	0		1		1	1

Podemos decir al comparar los valores de la tabla que $X = AB$ ya que son iguales

Para comprobar sacamos factor común en la expresión algebraica así:

$$X = AB (C' + 1 + C)$$

Aplico teorema 15, haciendo $C' + 1 = 1$ y lo reemplazo en la expresión, queda

$$X = AB (1 + C)$$

Aplico teorema 6 a $(1 + C) = 1$ y lo reemplazo quedando uno entre paréntesis que forma una and con AB

$$X = AB (1)$$

Siendo finalmente $X = AB$ está demostrado.

Ejercicios del tema 2: Algebra Booleana.

- a- Resolver usando teoremas $[(a+b)'' \cdot (c.d)'' + (a.b)' + (c+d)']' = f$
- b- Diseñe el circuito lógico para el caso anterior y su tabla lógica
- c- Simplifique $x = [(m+n) \cdot (m'+p) \cdot (n'+p')]'$

2.3. Simplificación de expresiones Algebraicas

Simplificación de expresiones algebraicas usando teoremas:

Consiste en aplicar a la expresión booleana o algebraica el caso más conveniente de los teoremas cuando la situación lo exija, para lograr reducir la operación sin cambiar la salida o respuesta. Por último dibujados el diagrama o circuito y lo comparamos con el de la expresión algebraica inicial.

Ejemplo 1

Simplificar la expresión: $A+B.A''+C.C+(A+B)'' = F$

S/n:

$$A+B \cdot A + C + A+B = FA+B. A+C=F$$

Circuito inicial para: $A+B.A''+C.C+(A+B)'' = F$

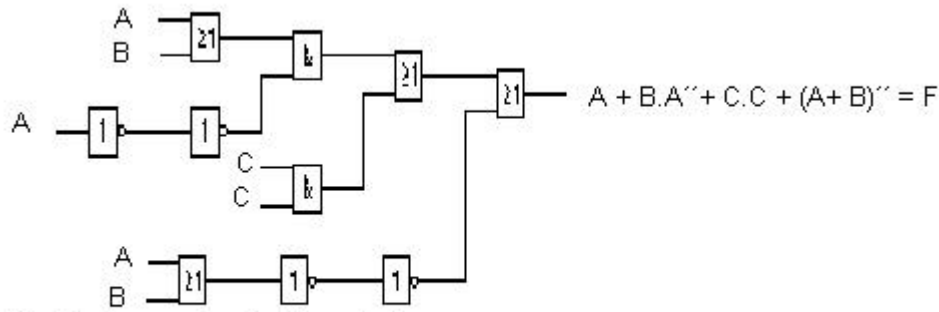


Fig.15: Representación Ejemplo 6
Autor: Alvaro de J. Laverde Q. (F: 03-05-2011)

Circuito Final para: $A+B. A+C=F$

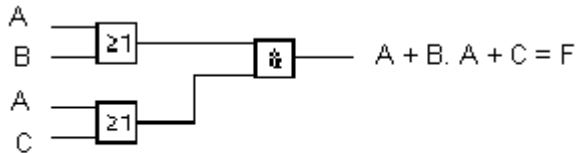


Fig.16: Representación $A+B.A+C= F$
Autor: Alvaro de J. Laverde Q. (F: 03-05-2011)

Observamos que ambos diagramas de circuito son iguales solo que el último es más reducido porque usa menos compuertas lógicas que el anterior.

Simplificación por medio de mapas de karnaugh:

El mapa de Karnaugh (también llamado como tabla de Karnaugh o diagrama de Veitch, abreviado como mapa K o mapa KV) Es un diagrama que se usa para la simplificación de funciones booleanas algebraicas, fue diseñado por Maurice Karnaugh en 1950 quien fue físico matemático de los laboratorios Bell.

Reglas para el uso de mapas de Karnaugh:

Los lazos de mini términos o maxi términos se hacen con base a la potencia del sistema numérico binario.

Los lazos diagonales no están permitidos, solo los horizontales y verticales adyacentes.

Las variables que cambien en un lazo se deben eliminar. Las variables que no cambien se deben representar en dicho lazo.

Realizar la menor cantidad de lazos o grupos con la mayor cantidad de maxi términos y mini términos.

Esta hecho de una serie de cuadros y posee 2 exp. N Filas y 2exp.N cuadros cada cuadrado tiene un 0 ó un 1segun el valor que tome la función en cada fila además los mapas K se pueden usar para funciones que tengan hasta 6 variables.

Mapa K de 2 variables:

Ejemplo 2

Simplificación de la expresión booleana de dos variables:

$$f = a'b + ab' + ab$$

Solución:

1- Represento el mapa. Al llenar la tabla, coloco un uno donde se intercepto el valor de la función. O sea el termino inicial $f = a'b + ab' + ab$, se ha marcado el cuadrado donde ubicó el 1.

	A	A'
B	1	1
B'	1	0

2- Se marcan los cuadros contiguos que manejen unos, las regiones marcadas son las simplificaciones.

	A	A'
B	1	1
B'	1	0

3- La función simplificada $f = b + a$ debido a que el 1 intercepto es común a ambos grupos.

Mapa K de 3 variables:

El mapa tiene 8 cuadrados que equivalen a 2^n , donde $n = 3$ variables A, B, C.

		BC			
		00	01	11	10
A	0	0 0	0 1	1 3	1 2
	1	1 4	1 5	1 7	1 6

En primera fila corresponde a $A = 0$

En la segunda fila corresponde a $A = 1$

En la primera columna corresponde a $BC = 00$ ($B=0$ y $C=0$)

En la segunda columna corresponde a $BC = 01$ ($B=0$ y $C=1$)

La tercera columna corresponde a $BC = 11$ ($B=1$ y $C=1$)

La cuarta columna corresponde a $BC = 10$ ($B=1$ y $C=0$)

Se colocaron "1" en los cuadrados que equivalen a los valores de la Función.

Ejemplo 3

Simplifique usando mapa K

$$F = A B C + A B C + A B C + A B C$$

Solución:

Se crea el mapa K de la fig.17 y se escogen los grupos. Vemos que surgen 3 grupos de dos unos.

		BC			
		00	01	11	10
A	0	1	1	1	0
	1	0	1	0	0

Fig.17: Mapa K
Autor: Alvaro de J. Laverde Q. (F:03-05-2011)

Hay un recuadro que es compartido por los tres grupos

La función simplificada es: $F = A B + A C + B C$

Con verde: AB

Con azul: AC

Con verde: BC

Mapa K de 4 variables:

Sea $f(A, B, C, D)$

El mapa de Karnaugh tendrá $2^4 = 16$ combinaciones posibles con cuatro variables como en la fig.18.

CD \ AB	AB			
	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Fig.18: Mapa K de cuatro Variables
Autor: Alvaro de J. Laverde Q. (F:03-05-2011)

Se utiliza de forma igual que con tres variables para ubicar los ceros y unos en los cuadrados.

Observe el orden de las variables A, B, C, D que van de más significativo a menos significativo. Al igual que antes para las columnas AB, las filas CD llevan el orden 00, 01, 11, 10 para que haya adyacencia.

Lógica combinatoria:

Son todos los sistemas de funciones lógicas con compuertas operando bajo una misma expresión algebraica o booleana, donde se puede emplear varios métodos de simplificación para reducir la cantidad de elementos combinatorial que tenga el sistema

Ejemplo 4

Resolver Fig.19.

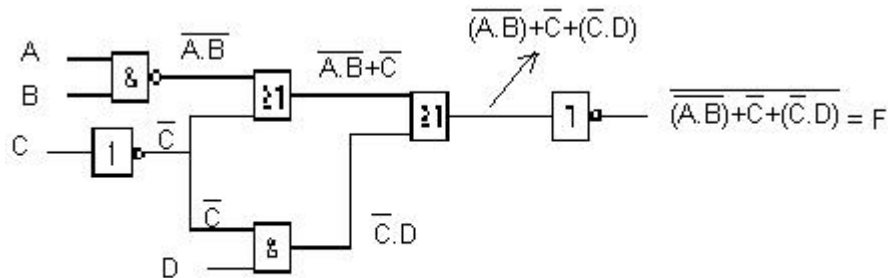


Fig.19 Ejemplo 9
Autor: Alvaro de J. Laverde Q. (F:03-05-2011)

La salida del sistema es:

$$((A.B)' + C' + (C'.D))' = F$$

Por tanto usando teoremas tenemos:

Primero resolvemos los complementos.

$$(A.B)'' + C'' + (C'.D)' = F$$

$$A.B + C + C''.D' = F$$

$$A.B + C + C.D' = F$$

$$A.B + C + D' = F$$

Entonces hacemos el diagrama final, representado en la fig.20

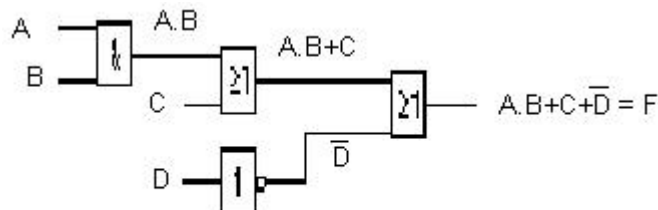


Fig.20: Diagrama final del ejemplo 9
Autor: Alvaro de J. Laverde. Q. (F: 03-05-2011)

Podemos ver que el sistema quedo más reducido de elementos lógicos, primero tenía seis compuertas y finalmente quedo de cuatro lo cual significa que en su proceso de simplificación redujo dos compuertas y su función es la misma.

Ejercicios del tema 3:

Simplificación por medio de teoremas y de mapas de Karnaugh.

- Simplifique mediante un mapa K: $f=w'x'y'+x'y+w'x$
- Evaluar la función lógica para el caso anterior si $w=0, y=1$.
- Realice el circuito lógico final del numeral b.

2.4. Flip-Flops

Son arreglos especiales con implementación de compuertas lógicas que permiten obtener funciones secuenciales de acuerdo al estado lógico de la salida con respecto de la entrada lógica, son RS, JK, D, y T.

La diferencia entre las clases de flip-flop está en el número de entradas y la forma en la cual las entradas modifiquen la salida binaria.

Flip-Flop RS:

Posee tres entradas, S= inicio, R = reinicio o borrado y C= entrada de pulso de reloj.

Una salida se llama Q, y la otra Q' complementada, la cual se marca con un círculo El flip-flop responde a un cambio positivo 0 a 1 generado por la señal de reloj.

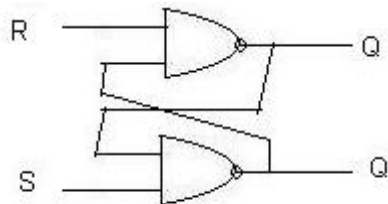


Fig.21: Flip-Flop RS con nand
Autor: Alavro de J.Laverde Q. (F:02-05-2011)

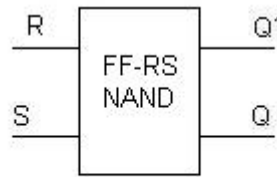


Fig.22: FF-RS Diagrama de bloque
Autor: Alavro de J.Laverde Q. (F:02-05-2011)

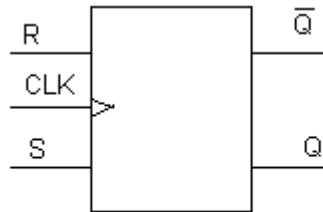


Fig.23: Simbolo grafico FF-RS
Autor: Alavro de J.Laverde Q. (F:02-05-2011)

Tabla Lógica RS

R	S	Q
0	0	q
0	1	1
1	0	0
1	1	x

R	S	Q (t+1)	
0	0	Q (t)	Sin cambio
0	1	0	Limpia en cero
1	0	1	Establece en uno
1	1	x	No determinado

Tabla de estado lógico

Operación:

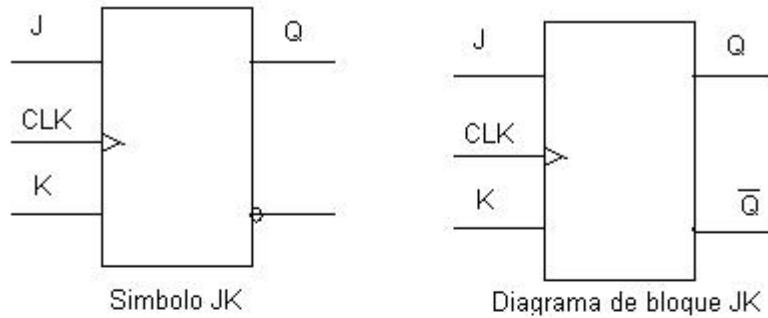
Si no hay señal de reloj la salida del circuito no cambia sin importar cuales sean los valores de R y S en la entrada.

Y solo hace cambio cuando el reloj pasa de 0 a 1, cuando S = 1 y R = 0 al C cambiar de 0 a 1, la salida Q se inicia en 1, cuando S = 0 y R = 1 al cambio de C, cambia de 0 a 1 y la salida Q se reinicia o borra en 0.

Al ser R y S iguales a 1 cuando cambia el reloj, la salida no se conoce.

Flip Flop JK:

Se comporta casi igual al anterior, la entrada J se usa para la entrada de puesta a uno y la entrada K para la puesta a cero, Cuando $Q=0$ cambia a $Q=1$ y viceversa.



J	K	Q
0	0	q
0	1	0
1	0	1
1	1	q'

Tabla Lógica

J	K	Q (t+1)	
0	0	Q (t)	Sin cambio
0	1	0	Limpia en cero
1	0	1	Establece en uno
1	1	Q'(t)	Complementado

Tabla de estado

Fig.24: Características FF-JK

Autor: Alvaro de J. Laverde Q. (F: 03-05-2011)

Este flip- flop es activado por nivel 0, 1 mediante $Q = J q^* + K^* q$ genera la siguiente configuración esquemática Fig.25

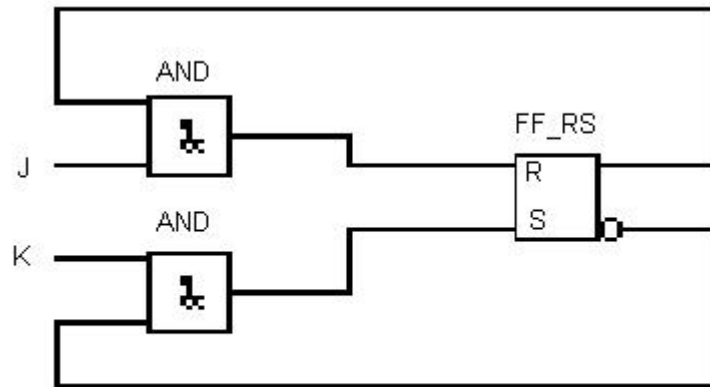


Fig.25: Esquema simplificado Jk
Autor: Alvaro de J. Laverde Q. (F: 03-05-2011)

Flip-Flop D (Datos):

Es una modificación del flip.flop RS colocando un inversor entre S y R, asignando el símbolo D a la única entrada, Cuando D = 1, la salida del flip-flop va al estado 1, y cuando D = 0, la salida del flip-flop está en el estado contrario.

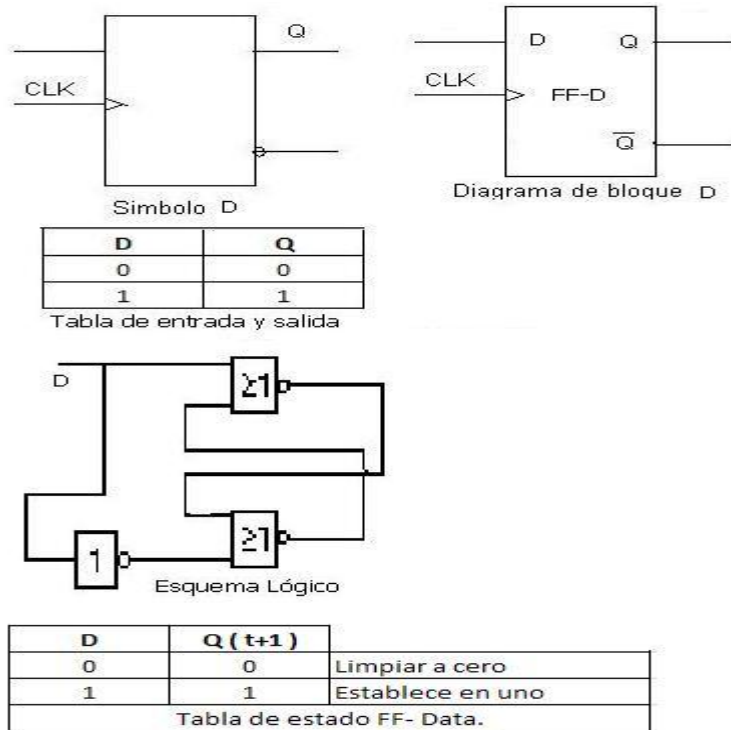


Fig.26. Características FF- Data
Autor: Alvaro de J. Laverde Q. (F: 03-05-2011)

Si no se determina como Flip-Flop D maestro-esclavo (FF-D-ME) llamado también Cerrojo Latch. La ecuación de funcionamiento es $Q = D$.

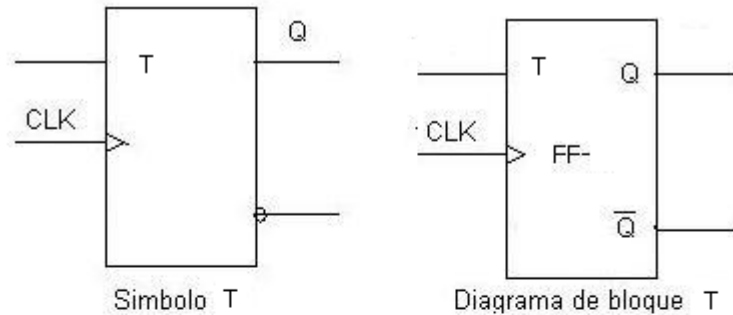
Flip-Flop T:

Surge a partir del tipo JK si J y K se unen forman una sola entrada denominada T.

El en flip-flop tipo T si $T = 0$ ($J = K = 0$) un cambio del pulso de reloj no modifica el estado del flip-flop.

Si $T = 1$ ($J = K = 1$) un pulso de reloj completa el estado del flip-flop también se denomina flip-flop activad por nivel y su ecuación es:

$$T = q \oplus Q$$



T	Q (t+1)	
0	Q(t)	Sin cambio
1	Q'(t)	Complementa
Tabla de estado FF- T		

Tabla de entra y salida

T	Q
0	q
1	q'

Fig.27: Características del FF-T
Autor: Alvaro de J.Laverde Q. (F: 03-05-2011)

Ejemplo 1

Implementar un contador de usando dos FF

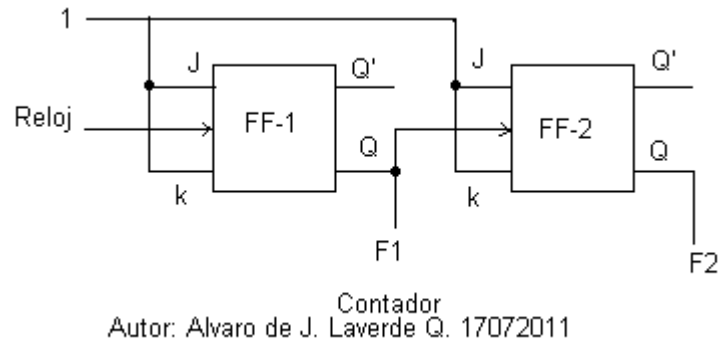
Solución:

Como punto de partida tenemos varios tipos de FF, de los cuales podemos usar el FF-JK tomando la salida q, dejando libre o sin uso la salida q'.

Unimos con un puente las entradas J y K en ambos FF luego sacamos como bit menos significativo.

La primera salida q del primer FF que será también la entrada de señal de reloj del segundo FF.

La señal de pulso de reloj del primer FF se debe implementar de forma independiente, pero la señal de reloj del segundo FF es el pulso de salida del primer FF, la unión común entre j y k de ambos FF se lleva un nivel o estado 1, F1 es el bit equivalente a las unidades y F2 es el bit equivalente a las decenas, observar la figura que resulta del proceso anterior



Ejemplo 2

“Demostrar que la salida de un flip-flop J-K, después de que ha ocurrido una transición, está dada por:

$$Q^{n+1} = JQ^n + K \cdot Q^n$$

En donde Q^n es la salida del flip-flop J-K antes de llevarse a cabo la transición.

Tomando en cuenta las propiedades del flip-flop J-K, se puede desarrollar la siguiente Tabla de Verdad:

J	K	Q^n	Q^{n+1}
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

De la Tabla de Verdad se puede obtener ahora Q^{n+1} en función de J, K y Q^n por medio de minterms:

$$Q^{n+1} = J \cdot K \cdot Q^n + J \cdot K \cdot Q^n + J \cdot K \cdot Q^n + JKQ^n$$

$$Q^{n+1} = (J + J) \cdot K \cdot Q^n + (K + K)JQ^n$$

$$Q^{n+1} = JQ^n + K \cdot Q^n$$

(Tomado de: <http://logica-digital.blogspot.com/2007/11/6-problemas-resueltos.html>, Fecha: 18072011).

Ejercicios del tema 4: Flip-Flop

1. Cuáles son los tipos de flip-flop
2. Diseñe un contador de cero hasta nueve implementado con
3. flip-flop.
4. 3-Cuales son las características de funcionamiento de los flip-flop.

Actividad de la unidad 1

Simular un sistema numérico contador digital de 0-99 en una aplicación de computadora con dispositivos flip flop.

3. COMPONENTES DIGITALES

Introducción a los componentes digitales en el siguiente link:

<http://www.monografias.com/trabajos7/codi/codi.shtml>

Objetivo general

- ◆ Interpretar el funcionamiento de los componentes digitales como parte esencial de la arquitectura del computador a nivel de los dispositivos que hacen parte del Hardware.

Objetivos específicos

- ◆ Analizar los tipos de circuitos integrados digitales.
- ◆ Conocer las partes y funcionamiento lógico del codificador.
- ◆ Conocer las partes y funcionamiento lógico del decodificador.
- ◆ Conocer las partes y funcionamiento lógico del multiplexor.
- ◆ Conocer las partes y funcionamiento lógico del demultiplexor.
- ◆ Conocer las partes y funcionamiento lógico de los registros.
- ◆ Analizar el funcionamiento de los registros de desplazamiento.
- ◆ Comprender el funcionamiento de las unidades de memoria.

Prueba Inicial

1. Mencione y explique los tipos de circuitos integrados que existen.
2. Que es un circuito digital codificador
3. Determine las aplicaciones de los elementos multiplexor y demultiplexor.
4. Describa un sistema de memoria.
5. Cuales son las características de un sistema de registro de desplazamiento.

3.1. Circuitos integrados

Aquellos componentes digitales que poseen unos números de funciones lógicas determinadas por el diseñador o fabricante con propósitos específicos de aplicación eléctrica, electrónica en el campo domestico ó industrial.

Dispositivos digitales: Son los elementos que hacen parte individual o integrado de un sistema de funciones y señales lógicas. Son básicamente: Circuitos integrados, Codificadores, Decodificadores Multiplexores, Registros, Registros de desplazamientos, Contadores binarios, Unidades de memoria.

Existen varias familias para la fabricación de circuitos integrados que son:

1. Familias MOS:

Se identifican por su bajo consumo de potencia, poseen alta impedancia, están fabricados con transistores MOSFET (FET de metal oxido semiconductor o transistor de efecto de campo) existe MOSFET de canal N y MOSFET de Canal P.

Básicamente tres familias:

CMOS: Metal Oxido Semiconductor Complementario.

NMOS: Metal Oxido Semiconductor Canal N

PMOS: Metal Oxido Semiconductor Canal P

2. Bipolares:

TTL: Lógica de Transistor -Transistor

ECL: Lógica de Emisores Acoplados implementada con dispositivos bipolares caracterizada por ser muy rápida.

EHL: Lógica de alto Umbral

IIL: Lógica de Inyección Integrada

RTL: Lógica de Resistencia Transistor

DTL: Lógica de Diodo Transistor

De acuerdo al nivel de integración en la cantidad de componentes los circuitos integrados se diferencian como:

GLSI: Giga grande o gran escala de integración más de 1 millón de transistores internos.

ULSI: Ultra grande más de 100.001 a 1.000.000 transistores

VLSI: Muy grande más de 10.001 a 100.000 transistores

LSI: Grande escala de integración de 1.001 a 10.000 transistores

MSI: Media escala de integración de 101 a 1.000 transistores

SSI: Pequeño nivel de integración de 10 a 100 transistores.

Los circuitos integrados están divididos en dos grandes grupos que son:

Circuitos integrados digitales (compuertas lógicas, microprocesadores, micro-controladores)

Circuitos integrados análogos (Osciladores, amplificadores, receptores).

Circuito integrados complejos que son diseñados para cumplir con una determinada función.

A nivel del tamaño, pueden llegar a ser microscópicos y de montaje rápido.

CIRCUITOS INTEGRADOS CMOS

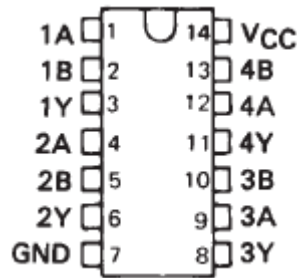
Se diferencian por su bajo consumo de potencia se fabrican con transistores MOSFET los que por su alta impedancia de entrada consume lo más mínimo.

En la tabla siguiente se pueden ver los rangos que manejan.

Familia	Potencia	Velocidad	Tensión
74HC00	10 mW	8-12 nS	3-15
4000	10 mW	20 – 300 nS	3-15
74C00	10 mW	20 – 300 nS	3-15

CIRCUITOS INTEGRADOS TTL

Usa elementos que son similares a los diodos, resistores, y transistores bipolares discretos, es la de mayor aplicación y posee subgrupos como: TTL estándar, TTL de potencia baja L, TTL de potencia baja LS, TTL S , TTL avanzada de potencia baja (ALS), TTL avanzada AS.



CI -TTL 74LS08 tomado de:

<http://proton.ucting.udg.mx/~zoad/programas/electfile/TexInst/Gates/74LS08.pdf>, fecha: 18-07-2011.

Su nivel de tensión de funcionamiento es a +5 Voltios para todos sus elementos.

Velocidad	Subgrupo
3nS	Schottky
4nS	Avanzada de potencia baja schottky
1.5nS	Avanzada
10nS	Estándar
33nS	Potencia Baja

Por su configuración interna, las salidas de los dispositivos TTL No deben unirse entre porque tienen que ser de colector abierto o de tres estados lógicos.

Nomenclatura externa de identificación en un circuito integrado.

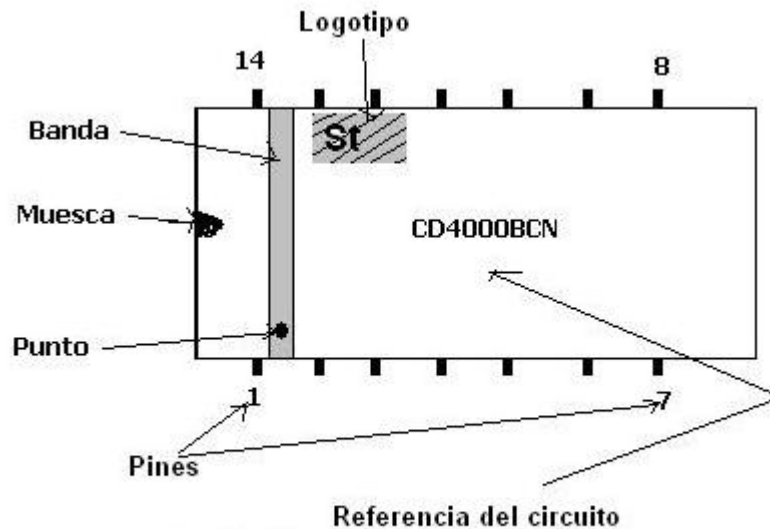
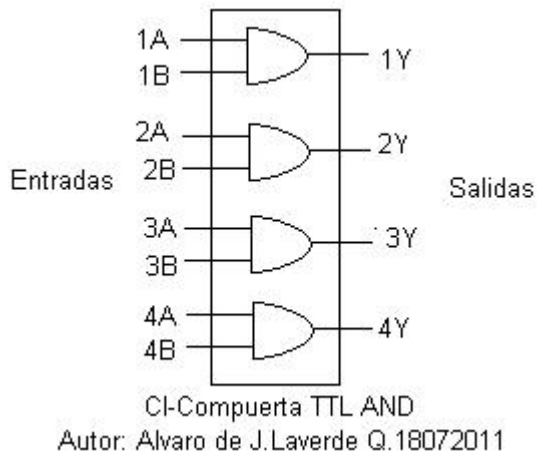


Fig.28: Nomenclatura externa del CI.
Autor: Alvaro de J.Laverde Q. (F: 03-05-2011)

Ejemplo 1

Implemente un circuito integrado con 4 compuertas AND diferenciando la configuración.

Solución:



Ejemplo2

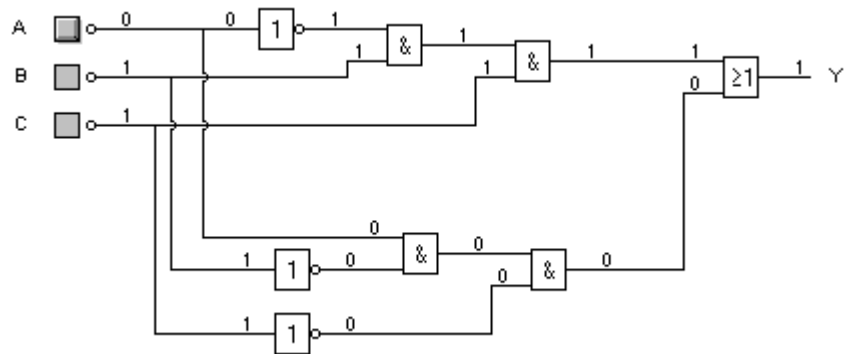
Diseñe un circuito lógico que cumpla para la siguiente tabla.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

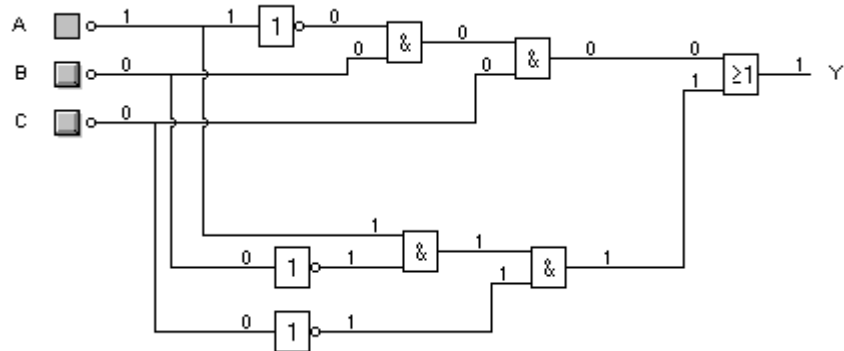
Solución:

$$Y = A'BC + AB'C'$$

Cuando las entradas son A = 0, B = 1, C = 1



Cuando las entradas son: A = 1, B = 0, C = 0



Ejercicio del tema 1: Circuitos integrados.

- Diseñe un circuito integrado con lógica AND.
- Cuáles son las especificaciones de los C.I. CMOS y TTL.
- Nombre 3 C.I. de CMOS y TTL.

3.2. Codificadores

Un codificador es un circuito diseñado para pasar información de un sistema digital a otro con clave diferente.

Codificador: Es un CI con 2^N entradas y N salidas, cuya función es entregar en la salida el binario correspondiente a la entrada activada.

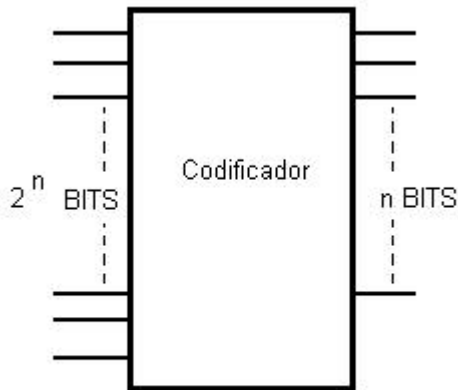


Fig.29: Codificador Binario
Autor: Alvaro de J. Laverde Q. (F:03-05-2011)

El más usado es el codificador de 7 segmentos para convertir entradas binarias codificadas en hexadecimal, a código decimal con 7 secuencias para representar caracteres alfanuméricos en una pantalla de diodos led.

TABLA LOGICA CODIFICADOR BCD															
ENTRADAS								SALIDAS							
1	2	3	4	5	6	7	8	9	0	Y4	Y3	Y2	Y1		
0	0	0	0	0	0	0	0	0	1	0	0	0	0		
0	0	0	0	0	0	0	0	1	0	0	0	1			
0	0	0	0	0	0	1	0	0		0	0	1	0		
0	0	0	0	0	1	0	0	0		0	0	1	1		
0	0	0	0	1	0	0	0	0		0	1	0	0		
0	0	0	1	0	0	0	0	0		0	1	0	1		
0	0	1	0	0	0	0	0	0		0	1	1	0		
0	0	1	0	0	0	0	0	0		1	0	0	0		
1	0	0	0	0	0	0	0	0		1	0	0	1		

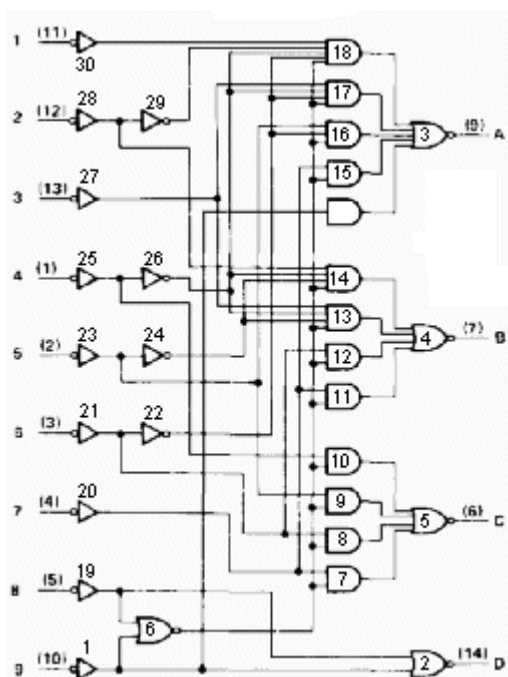


Diagrama lógico del codificador 74147 tomado del link: <http://ladelec.com/teoria/electronica-digital/183-codificadores.html>, Fecha: 17-07-2011.

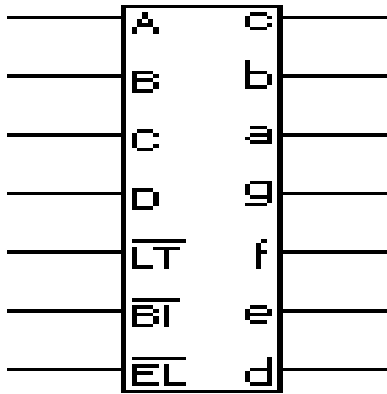
En la anterior tabla vemos que por cada combinación de ceros y unos en las entradas corresponde un valor binario el cual convertido a decimal equivale a los números decimales de 0-9.

La tabla lógica correspondiente al CI74147 es:

1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

Tabla de verdad del codificador 74147 tomada del link: <http://ladelec.com/teoria/electronica-digital/183-codificadores.html>, fecha: 17-07-2011

En la tabla de estados del codificador aparece representadas las entradas y salidas del codificador como nivel L= nivel bajo = 0 y H = nivel alto = 1, X representa los estados de transición que importan o no son tenidos en cuenta.



Codificador de 7 segmentos
Autor: Álvaro de J.Laverde Q.17052011

Si observamos el esquema logramos identificar que las entradas del codificador están marcadas con letras mayúsculas A, B, C, D, las salidas están identificadas con las letras minúsculas a, b, c, d, e, f, g y los demás son de control para garantizar su funcionamiento.

Inputs							Outputs							
No.	\overline{LT}	\overline{RBI}	D	C	B	A	$\overline{BI}/\overline{RBO}$	a	b	c	d	e	f	g
0	1	1	0	0	0	0	1	1	1	1	1	1	1	0
1	1	X	0	0	0	1	1	0	1	1	0	0	0	0
2	1	X	0	0	1	0	1	1	1	0	1	1	0	1
3	1	X	0	0	1	1	1	1	1	1	1	0	0	1
4	1	X	0	1	0	0	1	0	1	1	0	0	1	1
5	1	X	0	1	0	1	1	1	0	1	1	0	1	1
6	1	X	0	1	1	0	1	0	0	1	1	1	1	0
7	1	X	0	1	1	1	1	1	1	1	0	0	0	0
8	1	X	1	0	0	0	1	1	1	1	1	1	1	1
9	1	X	1	0	0	1	1	1	1	1	0	0	1	1
10	1	X	1	0	1	0	1	0	0	0	1	1	0	1
11	1	X	1	0	1	1	1	0	0	1	1	0	0	1
12	1	X	1	1	0	0	1	0	1	0	0	0	1	1
13	1	X	1	1	0	1	1	1	0	0	1	0	1	1
14	1	X	1	1	1	0	1	0	0	0	1	1	1	1
15	1	X	1	1	1	1	1	0	0	0	0	0	0	0
BI	X	X	X	X	X	X	0	0	0	0	0	0	0	0
RBI	1	0	0	0	0	0	0	0	0	0	0	0	0	0
LT	0	X	X	X	X	X	1	1	1	1	1	1	1	1

< I
< N
< V
< A
< L
< I
< D

Tabla codificador de 7 segmentos obtenida de:

<http://www.monografias.com/trabajos14/multiplexor/multiplexor.shtml>, fecha: 18-07-2011

La tabla anterior muestra todos los estados del dispositivo para comprender más profundamente su funcionamiento y operación, por ejemplo.

Para visualizar el número decimal 0 se requiere:

$(LT)' = 1$

$(RBI)' = 1$

Las entradas (D, C, B, A) = 0

$[(BI)' / (RBO)'] = 1$

Las salidas (a, b, c, d, e, f, g) = 1111110.

Ejemplo 1

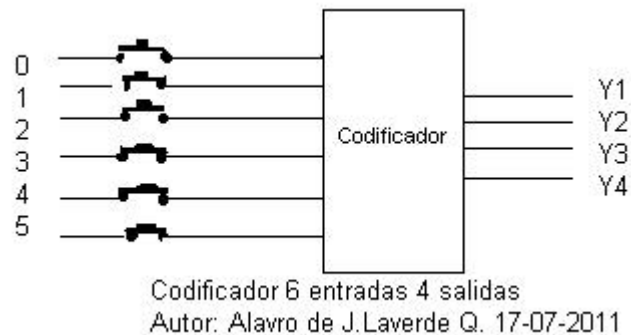
Implementar un codificador de 6 entradas y cuatro salidas

Solución:

La tabla lógica queda

F	E	D	C	B	A	Y4	Y3	Y2	Y1
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	1

La representación en bloques será.



Ejemplo 2

“Supongamos que queremos transmitir un código binario con cada una de las pulsaciones de un teclado numérico, como puede ser el de una calculadora, en éste existen diez dígitos y al menos seis caracteres especiales y, si consideramos sólo las diez cifras, esta condición la podemos satisfacer con cuatro bit.

Pero variemos el circuito de la calculadora para entender mejor el ejemplo. Modifiquemos el teclado de tal manera que al presionar una tecla se cierre el pulsador que conectará una línea de entrada.

En el interior del bloque podemos imaginar unos conductores cruzados que unen entradas y salidas entre sí. Veamos cómo han de conectarse a fin de que den los códigos deseados. Para representar los códigos de salida utilizaremos el denominado código BCD. La tabla de verdad que define este codificador es la siguiente:

Entradas	Salidas
1 2 3 4 5 6 7 8 9 0	Y4 Y3 Y2 Y1
0 0 0 0 0 0 0 0 0 1	0 0 0 0
0 0 0 0 0 0 0 0 1 0	0 0 0 1
0 0 0 0 0 0 0 1 0 0	0 0 1 0
0 0 0 0 0 0 1 0 0 0	0 0 1 1
0 0 0 0 0 1 0 0 0 0	0 1 0 0
0 0 0 0 1 0 0 0 0 0	0 1 0 1
0 0 0 1 0 0 0 0 0 0	0 1 1 0
0 0 1 0 0 0 0 0 0 0	0 1 1 1
0 1 0 0 0 0 0 0 0 0	1 0 0 0
1 0 0 0 0 0 0 0 0 0	1 0 0 1

Cuando alguna de las entradas está a 1, quiere decir que el pulsador correspondiente está accionado. Como suponemos que no hay más que un pulsador activado simultáneamente, entonces en cualquier línea de la tabla todas las entradas excepto una serán 0.

A partir de la tabla se deduce que la salida Y1 será 1 si lo es la entrada 9, ó la 7, ó la 5, ó la 3, ó la 1, de ahí que la ecuación booleana correspondiente a esta salida sea la suma de las entradas 1, 3, 5, 7 y 9. Si seguimos analizando la tabla obtendremos, de forma análoga, las ecuaciones que tienen que cumplir las salidas Y2, Y3 e Y4". Extraído de:

<http://www.angelfire.com/al2/Comunicaciones/Laboratorio/codifica.html>, Fecha. 18-07-2011.

Ejercicio del tema 2: Codificadores.

- Como se hace una codificación digital.
- Nombre dos tipos de codificadores
- Cuáles son las partes de un codificador.

Un decodificador es un circuito o dispositivo que retorna los datos o información de un segundo sistema digital al primer sistema que realizó la codificación.

Decodificador: Es un circuito combinacional, que transforma un código binario de entrada de N bits de entrada y M líneas de salida, N puede ser un entero y M es un entero menor o igual a 2^N , donde cada línea de salida será activa para una de las combinaciones posibles de entrada.

Se pueden encontrar como decodificador / demultiplexor porque un demultiplexor llega a funcionar como decodificador.

Entradas		Salidas			
B	A	F3	F2	F1	F0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Tabla de función lógica del decodificador 2x4

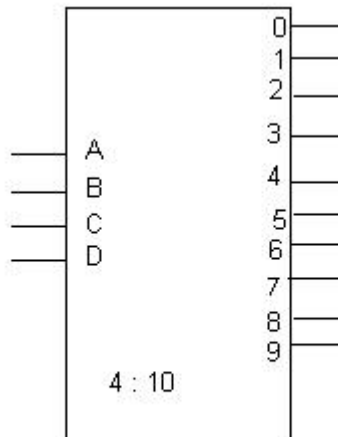
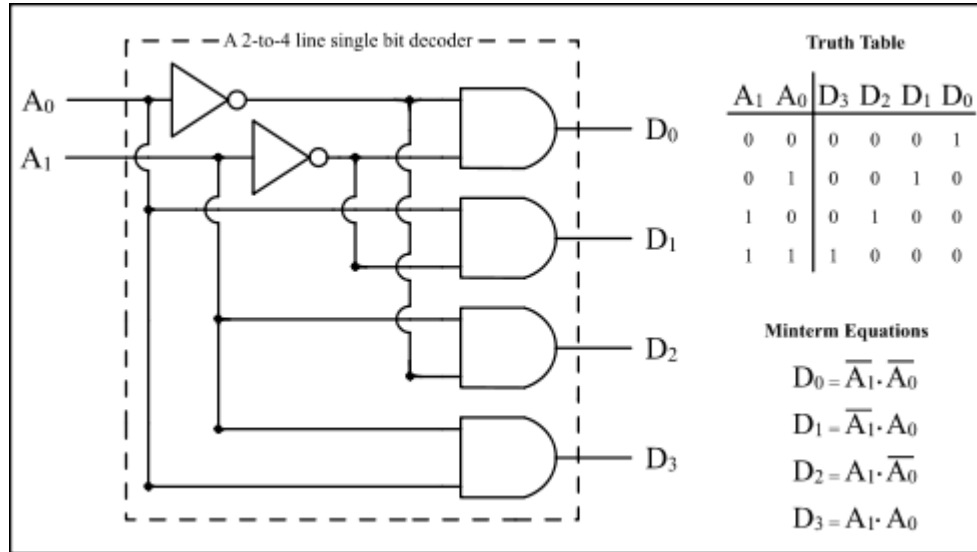


Fig.30 Decodificador 4:10

Autor: Alvaro de J.Laverde Q. (F: 03-05-2011)



Circuito lógico de decodificador tomado de la dir: <http://es.wikipedia.org/wiki/Decodificador>,
fecha: 17-07-2011

Ejemplo 1

Analizar la tabla lógica del siguiente decodificador:

ENTRADA			SALIDA			
S	R0	R1	Y1	Y2	Y3	Y4
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Si la entrada S es nivel bajo L el elemento decodificador estará funcionando, si está en nivel alto no funcionara porque está inactivo y las entradas R0, R1 no tendrán respuesta, la salida de este decodificador es activa en bajo y los estados X no importan.

Ejemplo 2

Con base en las 8 salidas de un decodificar 3 a 8 líneas construir la tabla lógica de operación si:

$$Y0 = C'B'A'$$

$$Y1 = C'B'A$$

$$Y2 = C'BA'$$

Y3 = C'BA
Y4 = CB'A'
Y5 = C B'A
Y6 = CBA'
Y7 = CBA.

Solución:

Partiendo de la información anterior concluimos que las variables de entrada al decodificador son A, B, C por lo tanto es una tabla de ocho filas por 3 columnas de entrada y 8 columnas de salida. Las entradas corresponden a todas las posibles combinaciones lógicas binarias 0 y 1 sin repetir valores, que forman el equivalente a las posiciones en la salida.

ENTRADAS			SALIDAS							
C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Tabla lógica del decodificador de 8 salidas

Ejercicio del tema 3: Decodificadores.

- Como se hace una decodificación digital.
- Nombre dos tipos de decodificadores digitales
- Cuáles son las partes de un decodificador.

3.3. Multiplexores

Multiplexor:

Se usa como componente electrónico digital que tiene varias entradas, y que puede enviar las señales de ellas por un medio, dividiendo este en múltiples canales, para realizar varias comunicaciones simultáneamente. Ampliación de este concepto en el link:

<http://www.angelfire.com/al2/Comunicaciones/Laboratorio/multiple.html>, fecha. 17-07-2011

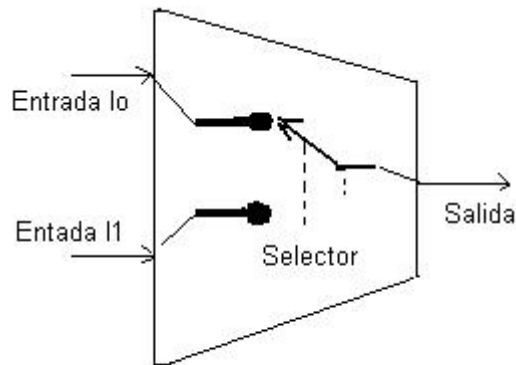
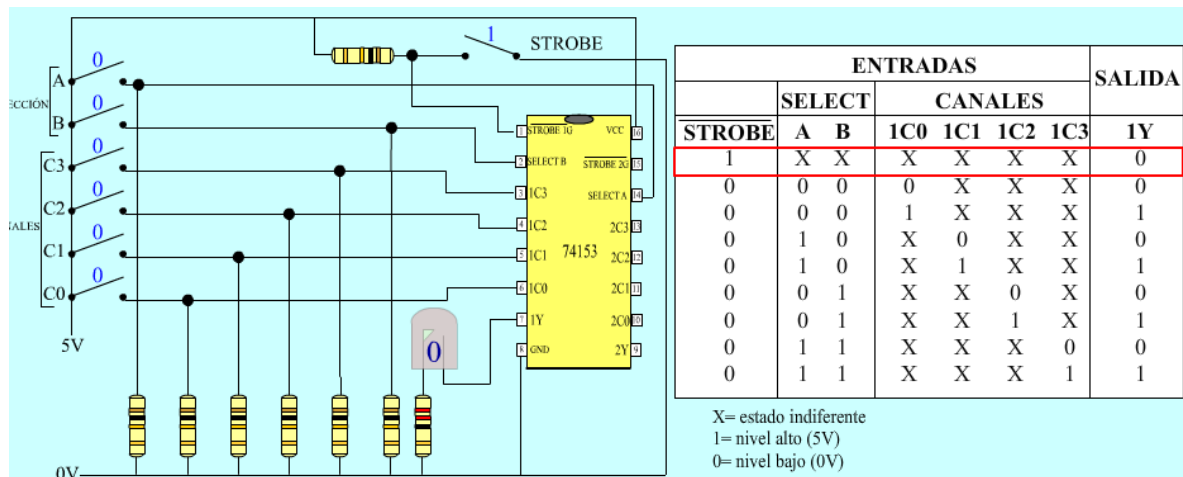


Fig. 31 Demultiplexor 2 a 1
Autor: Alvaro de J. Laverde Q. (F: 03-05-2011)



Funcionamiento de un multiplexor obtenido de:

http://meteo.ieec.uned.es/www_Usumeteog/comp_comb_multiplexores.html, fecha: 17- 07-2011.

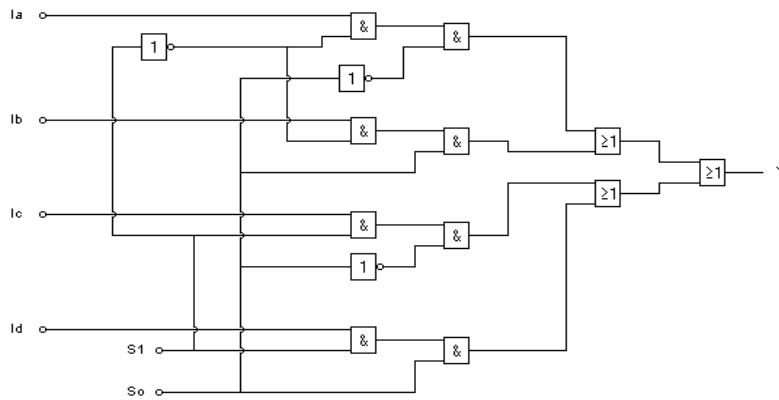
En el circuito el funcionamiento del multiplexor tiene un diodo LED conectado a la salida 1Y como dispositivo visualizador de la función, cuando la salida sea un estado 1 el diodo led estará encendido y cuando sea 0 el led estará apagado.

Ejemplo 1

Cuál es el diagrama del multiplexor descrito en la siguiente tabla:

S1	So	Y
0	0	Ia
0	1	Ib
1	0	Ic
1	1	Id

Solución:



Autor: Álvaro de J.Laverde Q.18072011

Para analizar el diagrama debemos tener en cuenta los valores de So y S1 para la salida. Se refiere a un multiplexor de 4 entradas que se transfieren en forma selectiva a la salida con referencia a las posibles combinaciones de las entradas de selección So, S1. Por ejemplo si las entradas de selección están en 0 y 0 la salida será la de señal (Ia) y en el caso de ser 1 y 1 la selección de la entrada entonces la salida será (Id).

Ejemplo 2 (Propuesto)

Analizar la tabla, el diagrama lógico y el símbolo lógico para la referencia de multiplexor número 74151.

Nota: Buscar en los manuales comerciales de los fabricantes en la web.

Cuestionario de repaso

- Que función desempeñan las entradas de selección de un multiplexor
- Cuántas entradas distintas pueden tenerse en un dispositivo multiplexor
- Cuáles son las condiciones de entrada de un multiplexor para que cada salida Y Tome el valor
- Lógico de la entrada que le pertenece.
Cuántas salidas y cuántas entradas pueden tener a lo máximo un multiplexor.

Ejercicio del tema 4: Multiplexores.

- Como se realiza un sistema multiplexor digital.
- Nombre dos tipos de multiplexores.
- Nombre varias aplicaciones con multiplexor.

3.4. Demultiplexor

Circuito que tiene una entrada de información de datos d y n entradas de control que sirven para seleccionar una de las 2^n salidas, por donde sale el dato que había en la entrada .
Colocando a las entradas de control la combinación binaria que corresponde a la selección de la salida requerida.

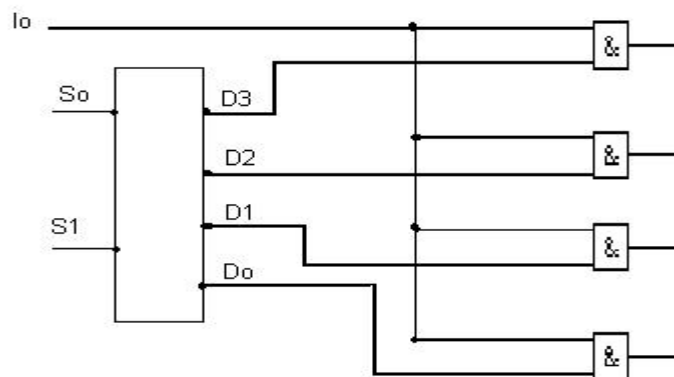


Fig.32: demultiplexor 1 a 4
Autor: Alvaro de J. Laverde Q.(F: 03-05-2011)

Son llamados dispositivos distribuidores de datos que realizan la operación inversa, coge una sola entrada y la entrega a varias salidas, tienen un código de entrada de selección escoge hacia qué salida se dirige la entrada de datos, significa que toma una base de entrada de datos y los coloca en uno de los canales de salida similar a un conmutador de varias posiciones de conexión.

C	B	A	strobe	d0	d1	d2	d3	d4	d5	d6	d7	y	w
x	x	x	1	x	x	x	x	x	x	x	x	0	1
0	0	0	0	0	x	x	x	x	x	x	x	0	1
0	0	0	0	1	x	x	x	x	x	x	x	1	0
0	0	1	0	x	0	x	x	x	x	x	x	0	1
0	0	1	0	x	1	x	x	x	x	x	x	1	0
0	1	0	0	x	x	0	x	x	x	x	x	0	1
0	1	0	0	x	x	1	x	x	x	x	x	1	0
0	1	1	0	x	x	x	0	x	x	x	x	0	1
0	1	1	0	x	x	x	1	x	x	x	x	1	0
1	0	0	0	x	x	x	x	0	x	x	x	0	1
1	0	0	0	x	x	x	x	1	x	x	x	1	0
1	0	1	0	x	x	x	x	x	0	x	x	0	1
1	0	1	0	x	x	x	x	x	1	x	x	1	0
1	1	0	0	x	x	x	x	x	x	0	x	0	1
1	1	0	0	x	x	x	x	x	x	1	x	1	0
1	1	1	0	x	x	x	x	x	x	x	0	0	1
1	1	1	0	x	x	x	x	x	x	x	1	1	0

Tabla demultiplexor de 8 entradas obtenido de:

<http://www.monografias.com/trabajos14/multiplexor/multiplexor.shtml> Fecha: 17-07-2011

Ejemplo 1 (Propuesto)

1-Establezca la diferencia entre MUX Y DEMUX por medio de símbolos lógicos.

Ejemplo 2

Cuando un demultiplexor tiene señal de salida 10000000, y en otro instante 00000010 determine los códigos de selección de entrada para S2, S1 y So.

Solución:

Como el dato de salida tiene 8 valores binarios y las entradas son S_0 , S_1 , S_2 se puede deducir que se trata de un DEMUX de 8 líneas o canales en el cual para la primera señal corresponde el código 111 y la segunda tiene el código 001.

Ejercicio del tema 5: Demultiplexor

1. Como funciona un Demultiplexor
2. Realice una aplicación que use Demultiplexor

3.5. Registros

Es un dispositivo electrónico semiconductor creado a partir de implementaciones con flip-flop los cuales permiten el manejo de datos usando circuitos combinatorios y secuenciales específicos. Las funciones de los registros son básicamente: Convertir los datos en arreglos serie y paralelo o al contrario paralelo a serie, recibir y transmitir datos de en paralelo y en serie, permitir el almacenamiento de información de registro llamado memoria.

Cada flip flop retiene un estado lógico 0 en la salida Q' , mientras realiza una transición pasando de estado 0 a estado 1 y lo contrario sucede en la salida Q , esto permite que esté un estado binario 0 en un instante y un estado binario 1 durante el cambio en el otro instante y a sucesivamente lo cual permite que se llegue a retener datos.

Contadores binarios:

Consiste en un circuito secuencial diseñado con compuertas lógicas y biestables, capaz de realizar el cómputo de los impulsos que recibe en la entrada. Sirve para retener datos o dividir una frecuencia, donde el cómputo se efectúa en código binario., pueden ser síncronos y asíncronos.

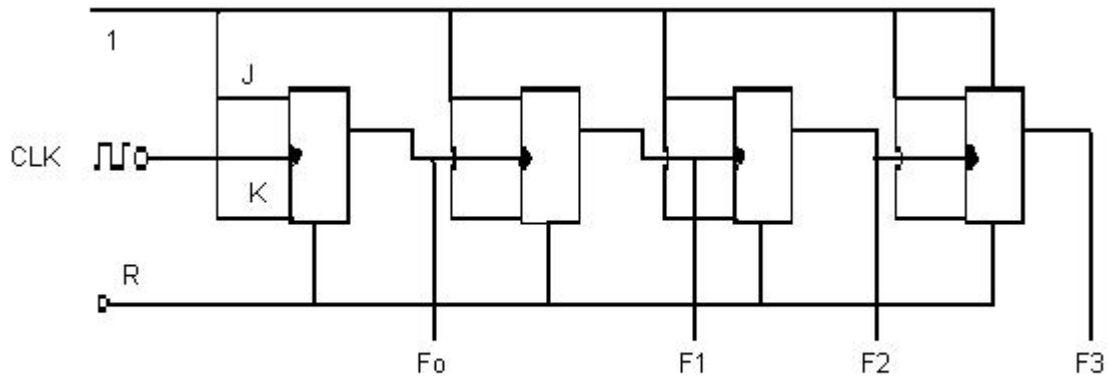
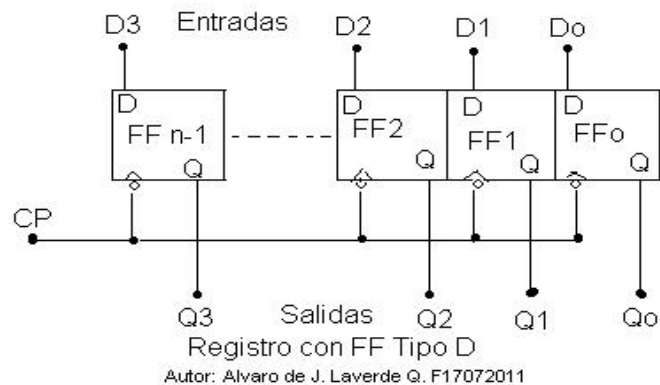


Fig.33: Contador BCD asincrónico
Autor: Alvaro de J. Laverde Q. (F: 03-05-2011)

Ejemplo 1

Al construir un registro con cuatro FF tipo D con una entrada común de pulsos de reloj llamada CP hace funcionar todos los flip-flop de manera que el dato presente este en las cuatro entradas y se pueda pasar al registro de 4 bit, para guardarlo por un tiempo, lo cual se lleva a cabo mediante un pulso de reloj.

Los datos pasan de la entrada de información D a la salida Q si el pulso de reloj CP está en 1, luego Si CP pasa a cero el dato que estaba fijo en la entrada de información antes de la transición, aparece en la salida Q, siendo los FF sensibles a la duración del pulso permitiéndole al registro habilitarse con CP =1



Autor: Alvaro de J. Laverde Q. F17072011

Ejemplo 2 (Propuesto)

Diseñar un sistema de transferencia de datos usando registros.

Ejercicio del tema 6: Registros

- Diga cómo funciona un registro.
- Nombre las clases de registros.
- Describa 3 aplicaciones con registros

3.6. Registros de desplazamiento

Son aquellos que tienen la facultad de pasar datos de un lugar a otro de su sistema de circuitos por medio de Flip-Flop en forma serial o paralela., logrando convertir de serie a paralelo y viceversa.

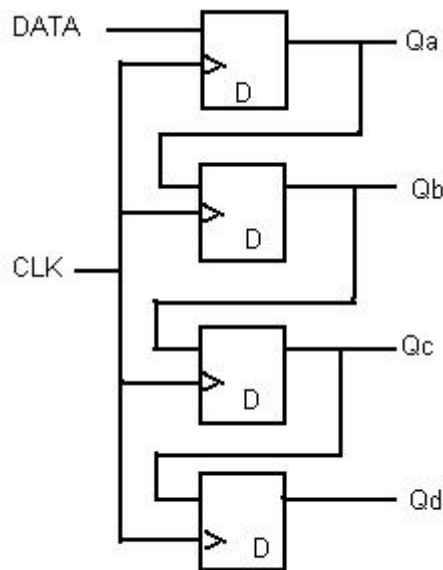


Fig.34: Registro D serie a paralelo
Autor: Alvaro de J. Laverde Q.(F :03-05-2011)

Registro de desplazamiento tipo data para pasar datos de forma serie a paralela.

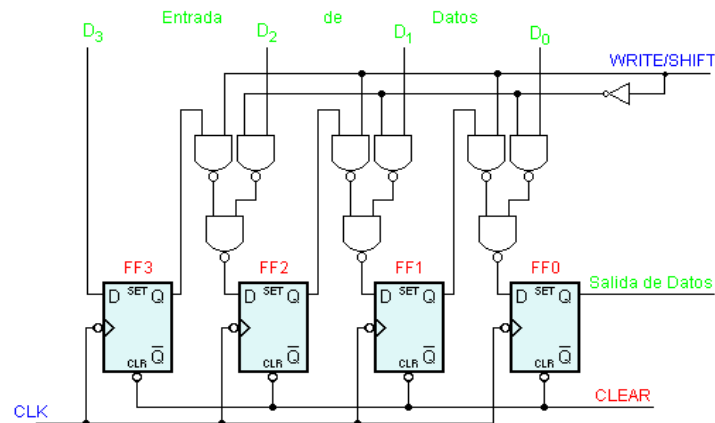
Los registros de desplazamiento también pueden ser de serie a serie, paralelo a paralelo.

Referencias adicionales en el link: <http://ladelec.com/teoria/electronica-digital/206-registros-de-desplazamiento.html>, fecha: 17-07-2011.

Ejemplo 1

“Registro entrada paralela y salida serial.

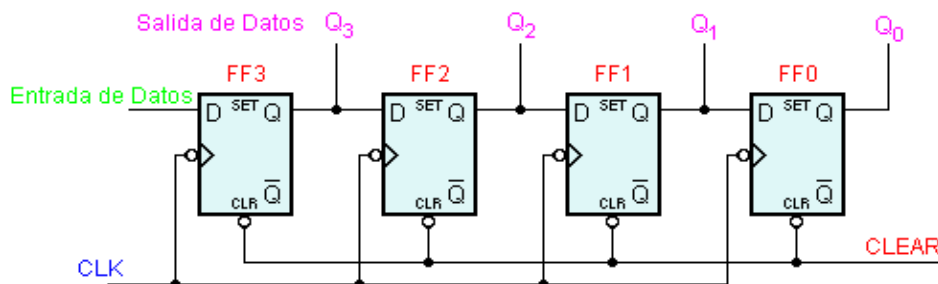
Se muestra un registro de desplazamiento con entrada serial y salida paralela. El circuito utiliza flip-flops tipo D y un arreglo de compuertas NAND para la entrada de datos al registro (escritura).



D3, D2, D1 y D0 son las entradas en paralelo, donde D3 es el bit más significativo y D0 el menos significativo. Para escribir los datos, la línea de control WRITE/SHIFT se coloca en BAJO (0 voltios) y la data se introduce con un pulso de reloj. La data se desplaza cuando la línea de control se coloca en ALTO (5 voltios). El registro realiza un desplazamiento hacia la derecha cuando se aplica el pulso de reloj, como se muestra en la animación a continuación”. Tomado del link: <http://medusa.unimet.edu.ve/sistemas/bpis03/registros.htm>, fecha 18-07-2011.

Ejemplo 2:

“Entrada serial y salida paralela.



Para este tipo de registro la data se introduce en forma serial, de la misma manera como se describió anteriormente. La diferencia está en la forma de extraerla. Una vez almacenada, cada bit aparece en su salida correspondiente, y todos los bits están disponibles simultáneamente. A continuación se muestra un registro de desplazamiento de 4 bits con esta configuración”.

Tomado del link:

<http://www.kumbaya.name/ci1210/leccion10%20registros%20y%20contadores/Registros.htm>,

Fecha: 18-07-2011

Ejercicio del tema 7: Registros de desplazamiento

- Diga cómo funciona un registro de desplazamiento.
- Nombre las clases de registros de desplazamiento.
- Describa 3 aplicaciones con registros de desplazamiento.

3.7. Unidad de memoria

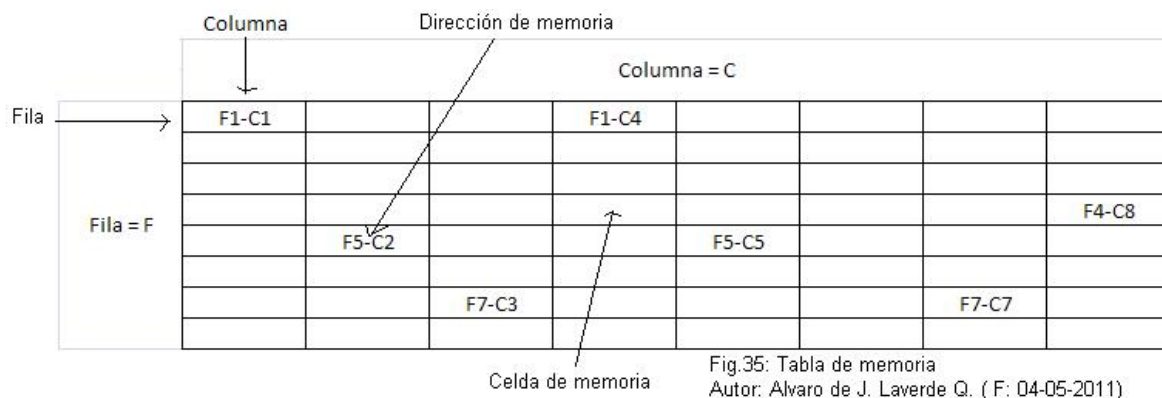
Memoria: Elemento o medio físico electrónico ó chip que es diseñado para almacenar datos en forma binaria durante un intervalo de tiempo, usando estructuras y arreglos de registros con flip-flop.

Celda de memoria:

Dispositivo o circuito electrónico que se usa para almacenar bits de datos en forma de 0 o 1, a través de una celda o flip- flop. Posee palabra de memoria, como un grupo de bits o celdas que representan instrucciones o datos de información de algún tipo.

Las memorias pueden estar diseñadas de flip-flops tipo data, cada espacio de memoria o celda posee una dirección hexadecimal que corresponde exactamente a la posición de la celda donde se encuentra o se va a almacenar un dato.

En la siguiente tabla se ilustra un ejemplo de funcionamiento de las memorias



Fundamentos de las memorias

Los elementos básicos de las memorias son:

- 1- El medio o soporte donde se almacenan los datos como unos o ceros.
- 2- El transductor que ubique en el soporte el dato requerido o que detecte su valor actual.
- 3- El medio de direccionamiento, que permita leer, fijar o grabar la información en lugar y tiempos Requeridos para lo cual hay tres condiciones:
 - a- Debe presentar dos estados estables, como una magnitud física.
 - b- Cambio de un estado a otro cuando se aplique una señal externa.
 - c- que se pueda saber el estado en el cual se encuentra en todo momento.

Los medios pueden ser discretos o sea un dispositivo físico individual que almacena cada bit, y continuos donde se almacenan unos bits a continuación de otros, en los medios discretos el acceso a los datos es más simple y rápido.

Según el tiempo que los datos permanecen grabados se puede clasificar la memoria:

- a- Intervalo de tiempo:
Permanecen los datos, hasta que se realice una operación de escritura y la memoria se determina como no volátil.
- b- Volátil:
La información desaparece cuando deja de suministrarse energía a la memoria.
- c- Con refresco:

Aun cuando la memoria este alimentada, los datos se van degradando presentando instantes en que no se pueden leer, para lo cual, deben refrescarse periódica y continuamente.

d- Lectura destructiva:

La lectura conlleva el borrado de la información e implica grabar el dato nuevamente

e- Solo lectura o permanente:

Permanece la información y no se puede borrar, algunas necesitan un proceso electrónico especial para ser borradas

Ejercicios propuestos

1- Describa las características de los tipos de memorias RAM del computador

2- Enuncie los tipos de memorias ROM

Ejercicio del tema 8: Unidad de memoria.

- a. Cuál es la importancia de las memorias.
- b. Describa el funcionamiento de una memoria
- c. Enuncie las características de almacenamiento de la memoria.

Actividad de la unidad 2

Explique el funcionamiento y operación de un sistema digital implementado con codificadores, registros, memoria.

4. REPRESENTACION DE DATOS, TRANSFERENCIA DE REGISTROS Y MICROOPERACIONES, ORGANIZACIÓN BASICA Y PROGRAMACIÓN

Algunos aspectos a tener en cuenta en los siguientes link:

(<http://www.modelo.edu.mx/univ/virtech/prograc/cbyn01.htm>)

(<http://www.youtube.com/watch?v=E6fvPix9gOQ&playnext=1&list=PLCABFB20F69E7F87>)

Objetivo general

Analizar la representación de datos, como efectúa micro-operaciones, como está organizado y programado el computador para su funcionamiento.

Objetivo específico

- ◆ Analizar los tipos de datos de uso frecuente en la computadora.
- ◆ Estudiar las funciones que ejecutan las instrucciones en el modo de operación de la computadora.
- ◆ Conocer el uso de la transferencia de registros.
- ◆ Diferenciar algunos conceptos en el proceso de programación básica del computador.

Prueba Inicial

- 1- Como se representan los datos
- 2- Que son Instrucciones y micro-operaciones
- 3- Determine las instrucciones básicas del computador.
- 4- Describa un sistema básico de programación
- 5- Cuales son las características de transferencia de datos

4.1. Representación de Datos

Tipos de datos.

Son bits (valores 0 y 1) en bajo nivel.

La unidad más pequeña que se puede direccionar de datos es un byte (8 bits).

La unidad de proceso por instrucción de código es una palabra de 32 o 64 bits.

Las instrucciones toman el nombre de palabra como un número binario, o como una palabra de 32 bits y puede representar valores enteros desde:

0 hasta $2^{32} - 1$ y desde -2^{31} hasta $2^{31} - 1$.

Primitivos

Los datos de un lenguaje de programación, originados por el lenguaje y con los que se puede hacer datos y estructuras de datos. Como son:

Char (carácter).

Int (entero).

String (cadena de caracteres).

Puntero (dirección de memoria).

Float (real (coma flotante)).

booleanos (lógico: Verdadero, Falso)

Simples:

Ordinal (Dato carácter, lógico, enumerado, sud rango, Dato entero,)

No ordinal (cronológico, puntero, real).

Compuestos:

Cadena de caracteres, estructura de datos, puntero, referencia, algebraicos, objeto, programación orientada a objeto, clase, función, tupla, vector.

Complementos

Son estructuras o formas que representan a algunos sistemas binarios de bajo nivel.

El complemento a dos de un número N dado por n dígitos, se deduce como:

$$C_2^N = 2^n - N$$

N = Numero

n = Numero de bits

Ejemplo:

$$N = 1000_2$$

$$N = 8_{10}$$

$$n = 4 \text{ bits}$$

$$\text{Aplicando } C_2^N = 2^n - N$$

$$C_2^N = 2^4 - 8$$

$$C_2^N = 16 - 8$$

$$C_2^N = 8 = 1000_2$$

$$\text{Si } N = 45$$

$$N = 101101_2$$

6 dígitos

Complemento a 2 de -45= 11010011 con 8 bit

El 19 en binario 00010011

En 16 bits = 111111111010011, 000000000010011

Técnica del complemento a 2 para números negativos

1. Se representa el número decimal dado como positivo.
2. Se representa en binario positivo
3. Se le saca el complemento 1 del número binario obtenido en el paso anterior cambiando los unos por ceros y viceversa.
4. Al complemento 1 se suma 1 y el resultado es el complemento 2.

Representación de punto fijo.

Se utiliza para los números enteros con signo o fracciones con signo, las cantidades se representan en binario con complemento a 1, 2 con 8,16 o 32 bits.

Representación de punto flotante.

Se utiliza para representar números no enteros, muy grandes o muy pequeños.

Un número en punto flotante se expresa:

$M.r^a$

M= Mantisa, un número de punto fijo

a= Exponente o característica, entero de punto fijo

r = La base, base 2.

La mantisa es la magnitud del número.

El exponente número de lugares a desplazar el punto decimal o binario.

Códigos de detección de error.

Dígito verificador

Reed-Solomon

Código Binario de Golay

Bit de paridad

Código Hamming

FEC (Forward Error Correction)

Ejercicios del tema 1: Representación de datos.

a-Que son complementos

b- Que significa punto fijo

c- Cual s el desempeño de la ALU.

d- Como funciona el bus.

4.2. Transferencia de registro y micro operaciones

Lenguaje de transferencia de registros.

La instrucción es una operación elemental efectuada en información guardada en varios registros llamados flip-flop.

La respuesta puede reemplazar la información binaria del registro o pasarla a otro. Pueden ser:

Contar, desplazar, cargar, borrar.

La organización interna del hardware de una computadora digital se define mejor al especificar:

- ◆ El conjunto de registros que contienen y sus funciones.
- ◆ La secuencia de micro operaciones que se realizan sobre la información binaria almacenada en los registros.
- ◆ El control que inicia la secuencia de micro operaciones.

Transferencia de registros.

Modo para representar con símbolos la secuencia de micro operaciones entre los registros de un módulo para diseñar sistemas digitales:

Registro de contador de programa.

Registro de instrucción

Registro de procesador.

Registro de dirección de memoria

“La transferencia de información de un registro a otro se representa con un símbolo usando un operador de sustitución $R1 \leftarrow R0$

Expresa, que el contenido de $R0$ se transfiere a $R1$.

Los registros de computadora están representados por letras mayúsculas (seguidas de números), para denotar la función del registro.

Por ejemplo:

- ◆ MAR (Memory address register) : registro que contiene una dirección para la unidad de memoria
- ◆ PC : contador de programa
- ◆ IR : registro de instrucción
- ◆ R1 : registro de procesador

Los flip-flops individuales en un registro de n bits están numerados en una secuencia de 0 hasta $n-1$, comenzando desde 0 en la posición de la extrema derecha y aumentando los números hacia la izquierda. La manera más común de representar un registro es mediante una caja rectangular con el nombre del registro dentro.

La transferencia de información de un registro a otro se representa en forma simbólica mediante un operador de sustitución. El enunciado $R2 \leftarrow R1$ denota una transferencia del contenido del

registro R1 al registro R2. Por definición, el contenido del registro fuente R1 no cambia después de la transferencia.

Normalmente queremos que la transferencia ocurra sólo bajo una condición de control predeterminada. Esto puede mostrarse mediante un enunciado si, entonces: Si ($P=1$) entonces ($R2 \leftarrow R1$), donde P es la señal de control que se genera en la sección de control.

En ocasiones es conveniente separar las variables de control de la operación de transferencia de registros al especificar una función de control. Una función de control es una variable booleana, que es igual a 0 o a 1. La función de control se incluye en el enunciado como sigue: $P : R2 \leftarrow R1$. La condición de control se termina con un signo de dos puntos. Una función de control es una variable booleana, que es igual a 0 a 1. La condición de control se termina con un signo de dos puntos. La operación de transferencia se ejecuta por el hardware sólo si $P=1$.

Se usan paréntesis para señalar una parte de un registro al especificar el rango de los bits o al otorgar un nombre simbólico a una parte de un registro. Se usan comas para separar dos o más operaciones que se ejecutan al mismo tiempo.” Tomado del link:

<http://sites.google.com/site/ycmario/home/cursos-fiis/st-314-arquitectura-del-computador/transferenciaderegistroymicrooperaciones>

Micro operaciones aritméticas, lógicas y de desplazamiento.

Corresponde a las operaciones básicas en binario efectuadas entre los registros del procesador como:

Suma, resta, multiplicación y división

(<http://www.monografias.com/trabajos14/sistemanumeracion/sistemanumeracion.shtml#op>)

Binarias.

Que requieren de algoritmos de bajo nivel en un lenguaje adecuado como:

Lenguaje de máquina o lenguaje ensamblador

(<http://homepage.mac.com/eravila/asmix862.html#II.2>) y junto con la unidad de desplazamiento lógico aritmético se encuentra el set de módulos dispuestos para tal fin dentro del sistema.

Ejercicios del Tema 2: Transferencia de registro y micro operaciones

- a- Explique el funcionamiento de los registros
- b- Como se realiza una micro-operación o micro-instrucción.

4.3. Organización básica del computador

Unidad de control

Unidad encargada de gestionar y operar las instrucciones de control que requiere el usuario programador alojados en la memoria ram, también se encarga de leer los datos que entran desde los elementos de entrada, hace que sean retenidos en la memoria, entregados a la unidad aritmética y lógica en el momento de ser requeridos por las operaciones de cálculo, logra que se puedan guardar y recuperar los datos y los valores almacenados en la memoria ram para en caso de ser requerido puedan llegar los datos a los periféricos de salida.

Generalmente la unidad de control emite y envía las órdenes a los dispositivos encargados de ejecutarlas en los sentidos requeridos cuando lee el programa almacenado y ejecutado en la memoria ram.

Posee las siguientes partes:

El contador que tiene la dirección de la instrucción que se está ejecutando en un momento determinado del programa.

El registro de control que lleva o tiene la instrucción siguiente del programa.

El registro de estado que es el acumulador

El secuenciador (o unidad lógica y de supervisión), que sincroniza los procesos durante la ejecución de la instrucción con la velocidad del reloj del sistema.

Unidad de procesador

Es la parte de la computadora llamada microprocesador capaz de realizar el procesamiento de toda la información, la cual está formada por la unidad de control, la unidad aritmética lógica, los registros de entrada y salida de datos, tiene también el reloj del sistema y adicionalmente en algunos casos integra cierta cantidad de memoria ram llamada ram caché de nivel 1.



Las características que los diferencian entre son:

La velocidad.

El consumo.

La cantidad de memoria interna.

La capacidad del bus de datos.

Las partes principales de un procesador son:

Unidad de control

Unidad aritmética y lógica

Los registros

La memoria

Unidad aritmética y lógica (ALU).

Es la unidad en la computadora encargada de realizar los cálculos, realizando las operaciones matemáticas como son la suma, resta, multiplicación, división y lógica de comparación requerida por el programa almacenado al ejecutarse en la memoria ram de la computadora.

Existe además la unidad de punto flotante que es la encargada de hacer los cálculos más complejos que la ALU no pueda llegar a efectuar.

Unidad de memoria (MMU).

Es un dispositivo de hardware electrónico dispuesto con circuitos integrados que sirve para manejar los accesos a la memoria a partir de la unidad de procesamiento central (CPU), convierte las direcciones lógicas en direcciones físicas, realiza la protección de la memoria, controla la memoria caché y tiene las entradas de la tabla de páginas para recuperar direcciones físicas que corresponden a direcciones lógicas.

El bus Principal

Es el encargado de llevar la información administrándola internamente entre todos los dispositivos de entrada-salida interconectado con la memoria ram.

Se subdivide en:

Bus de datos

Bus de direcciones

Bus de control.

Unidad de almacenamiento (U.HD)

Son aparatos electrónicos y/o eléctrico-mecánicos diseñados para almacenar información en forma permanente, constituidos por varios discos apilados de doble cara divididos en anillos concéntricos denominados cilindros y a la vez subdivido en sectores.

Los datos son grabados o alojados en predestinados sectores y pistas, cada sector soporta 512 Bit a 33Kb. Los datos son fijados por la CPU usando un brazo con varias cabezas, que lee o escribe sobre las pistas de los cilindros.

Tiene las siguientes partes:

Tarjeta Lógica de control

Carcasa

Brazo

Cabeza de lectura- escritura

Actuador

Discos

Sistema de entrada y salida (Y/O)

Conjunto de interfaces que usan los dispositivos funcionales de un sistema de procesamiento de datos para comunicarse con los demás medios de información, usando señales recibidas o enviadas.

Cuando la computadora recibe una señal de otra interfaz se le llama señal de entrada y cuando envía una señal a otro dispositivo de datos se le llama de entrada.

Algunos elementos de entrada son:

Mouse

Teclado

Scanner

Algunos de salida son:

Impresora

Monitor

Modem

Red

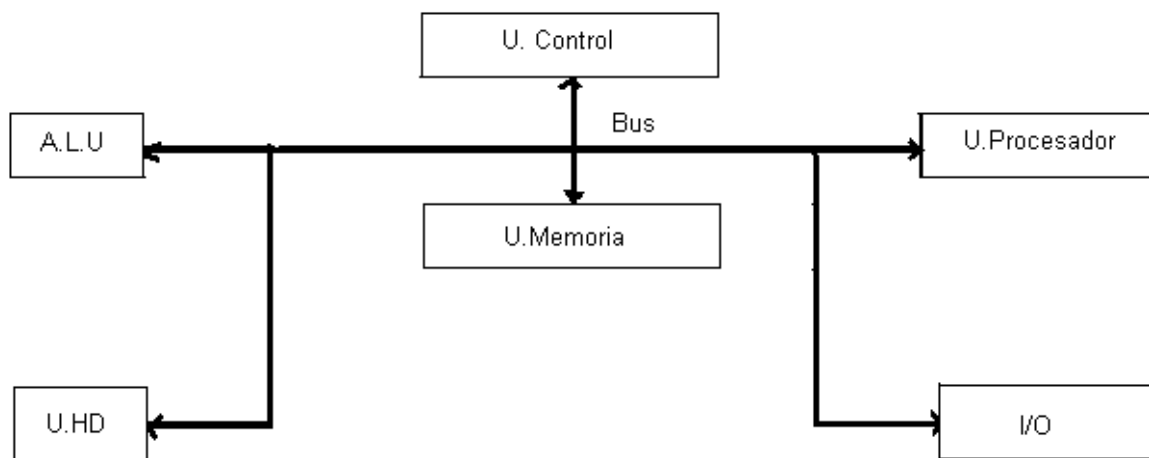


Diagrama de bloques de la computadora
Autor. Álvaro de J Laverde Q. 17052011

Códigos de instrucción.

Es un valor que representa una función que puede realizar el procesador del ordenador, un grupo de instrucciones es la suma de las operaciones elementales de un procesador como un factor determinante en su arquitectura para poder operar bajo ciertas condiciones de diseño del fabricante.

“En un sistema digital para propósitos especiales, la secuencia de micro-operaciones se fija y el sistema ejecuta la misma tarea específica repetidas veces. El usuario de una computadora puede controlar el proceso por medio de un programa. Los códigos de instrucción, conjuntamente con los datos, se almacenan en la memoria. El control interpreta entonces la instrucción y procede a ejecutarla emitiendo, una secuencia de funciones de control. La habilidad de almacenar y ejecutar instrucciones, el concepto de programa almacenado, es la propiedad más importante de una computadora para propósito general.

Un código de instrucción es un grupo de bits que le dice a la computadora como realizar una operación específica. La parte más básica de un código de instrucción es su parte operativa. El código de operación es un grupo de bits que define una operación. El número de bits requeridos para la parte de operación del código de instrucción es una función del número total de operaciones usadas. Debe de consistir de por lo menos n bits para 2^n operaciones dadas diferentes.

Formatos de códigos de instrucción.

Los bits de instrucción por lo general se dividen en grupos que subdividen la instrucción en partes. A cada grupo de le crea un nombre simbólico tal como parte del código de operación o parte de

una dirección. Las diferentes partes determinan diferentes funciones para la instrucción y cuando se muestran juntas constituyen el formato de código de instrucción.

En la siguiente figura, considérese, los tres formatos del código de instrucción en (a) consiste de un código de operación que implica un registro en la unidad procesadora. El formato de instrucción (b) tiene un código de operación seguido de un operando. Este se llama una instrucción de operando inmediato, porque el operando sigue inmediatamente después de la parte de código de operación de la instrucción. El formato de instrucción en (c) es similar al de (b) excepto que el operando debe extraerse de la memoria al lugar especificado por la parte de dirección de la instrucción.

(a) Implícito

(b) Operando Inmediato

Diseño de un computador sencillo.

Este sistema consiste básicamente de una unidad de memoria, siete registros y dos decodificadores. La unidad de memoria contiene 256 palabras de 8 bits con esto se puede demostrar las operaciones básicas que se encuentran en los computadores.

Una breve descripción de los registros que contiene un computador sencillo el cuál, estos se encargan del proceso de información se muestra en esta tabla.

Símbolo	Número de bits	Nombre del registro	Función
MAR	8	Registro de dirección de memoria	Almacena direcciones de memoria
MBR	8	Registro separador de memoria	Almacena contenidos de palabras de memoria
A	8	Registro A	Registro procesador
R	8	Registro R	Registro procesador
PC	8	Contador de programa	Almacena la dirección de instrucción
IR	8	Registro de Instrucción	Almacena códigos de operación corrientes
T	8	Contador de tiempo	Generador de secuencias

Si queremos leer una instrucción, el contenido que hay en PC se transfiere al MAR y así se inicia un ciclo de lectura de lectura de memoria, así pues el PC se incrementa a 1 y esto hace que almacene la siguiente dirección en la secuencia de instrucciones, después un código de operación leído de la memoria al MBR, se transfiere al IR. Si la parte de dirección de memoria de una instrucción se lee al MBR, esta dirección se transfiere al MAR para leer él operando. Entonces el MAR puede recibir direcciones del PC o del MBR.

Este computador está compuesto de 8 bits en el código de operación por lo que tenemos hasta 256 operaciones diferentes. Pero a manera de simplificar tenemos tres instrucciones para un computador sencillo.

Código de operación	Mnemónico	Descripción	Función
00000001	MOV R	Mover R a A	A R
00000010	LDI OPRD	Cargar OPRD a A	A OPRD
00000011	LDA ADRS	Cargar el operando especificado por ADRS a A	A M(ADRS)

La mnemotécnica asociada con cada instrucción puede usarse por los programadores para especificar las instrucciones con nombres simbólicos”. Tomado de la dirección:
<http://html.rincondelvago.com/transferencia-entre-registros.html>, 18-07-2011

Registros de computador.

Son soporte y ayuda a las operaciones efectuadas por unidad de control y la ALU, guardan información, temporalmente, para gestionar los datos y registros básicamente son: De propósito general, segmento de memoria y de instrucciones.

Para propósito general:

(AX) (DX) (CX) (BX)	Registros de Datos
(BP)	Registro Puntero Base
(SI)	Registro Índice Fuente
(DI)	Registro Índice Destino
(SP)	Registro Puntero de la Pila

Como segmento de Memoria:

(CS)	Registro Segmento de Código
(SS)	Registro Segmento de la Pila
(DS)	Registro Segmento de Datos
(ES)	Registro Segmento de Datos Extra

- (DS) Registro Segmento de Datos Extra
- (ES) Registro Segmento de Datos Extra

De instrucciones

- (FL) Registro de «Flags» o también denominado registro de estado
- (IP) Registro Puntero de Instrucción o Contador de Programa (PC)

Los cuatro más importantes son:

El Registro Puntero de Instrucciones.

El Registro Acumulador.

El registro de Estado.

El Registro Puntero de la Pila.

Las instrucciones de computador, Temporización y control, Ciclo de instrucción, instrucciones referidas a la memoria, entrada-salida e interrupciones. Pueden ser analizadas con referencia a un tipo de procesador, para conocer sus características específicas en el proceso de manejo, operación y establecer diferencias de capacidad, de funciones y desempeño a partir del 80386.

Ejercicios del tema 3: Organización básica del computador.

- a- Explique el proceso de temporización
- b- Como se identifica la interrupción de entrada/salida del computador.

4.4. Programando el computador básico

La programación del computador exige un lenguaje que puede ser conocido o desarrollado con base en conocimientos básicos del funcionamiento y operación del sistema de cómputo.

Para lo cual también se dispone del lenguaje de máquina, Lenguaje de ensamblador para realizar ciclos de programa, operaciones aritméticas, lógicas de la programación, rutinas, Subrutinas y programación de entrada y salida de periféricos.

Los programas realizados en lenguaje ensamblador están formados por una secuencia de enunciados origen, donde cada uno tiene una secuencia de caracteres ASCII que se autorizan con un retorno.

Grupo de instrucciones en lenguaje ensamblador para el procesador de la computadora:

Esta sección queda abierta para que se dirija al set de instrucciones del procesador que se requiera, para no limitarse, debido a la gran variedad que hoy existe para este estudio detallado,

una introducción previa se dispone del set de instrucciones para el procesador 8085A en el siguiente link:

(<http://www.webelectronica.com.ar/news12/nota09.htm>, Fecha 18-07-2011).

Ejemplo 1

El conjunto de instrucciones de un procesador:

Conjunto CISC (Conjunto de instrucciones complejas)

Conjunto RISC (Conjunto de instrucciones reducidas)

Ejemplo 2

Tipos de instrucciones y ejemplos

Transferencia de datos:

Copian datos de un origen a un destino, sin modificar el origen y normalmente sin afectar a los flags o indicadores de condición. Pueden transferir palabras, fracciones de palabras (bytes, media palabra) o bloques completos de n bytes o palabras.

Estas operaciones pueden ser:

Registro registro

Registro memoria

Memoria registro

Memoria memoria

Nemotécnicos más frecuentes:

move: copia el contenido de un registro(o memoria) a otro.

store: copia el contenido de un registro a memoria.

Load: copia el contenido de una posición de memoria a un registro.

- ◆ move block: copia un bloque de datos de una posición de memoria a otra.
- ◆ move múltiple: copia del origen en varias posiciones de memoria.
- ◆ exchange: intercambia el contenido de dos operando.
- ◆ clear: pone a 0 el destino. (todos los bits)
- ◆ set: pone a 1 el destino. (todos los bits)
- ◆ push: introduce un dato en la cabecera de la pila. (indicada por el SP)
- ◆ pop: saca un dato de la cabecera de la pila. (indicada por el SP)
- ◆ Instrucciones aritméticas: Son efectuadas por la ALU y suelen cambiar los flags o indicadores de condición.

Nemotécnicos más frecuentes:

- ◆ add: Suma.
- ◆ add with carry: Suma con acarreo.
- ◆ subtract: Resta.
- ◆ subtract with borrow: Resta teniendo en cuenta el adeudo anterior.
- ◆ increment: incrementa en 1 un valor.
- ◆ decrement: decremento en 1 un valor.
- ◆ multiply: multiplica.
- ◆ divide: divide.
- ◆ extend: aumenta el operando de tamaño.
- ◆ negate: cambia de signo.
- ◆ absolute: valor absoluto.

Pueden tener instrucciones para tratar con números en BCD e incluyen operaciones en coma flotante, lo cual se identifica con una 'f' antes del nombre del nemotécnico como por ejemplo: fabsolute

- ◆ Instrucciones de comparación: Suelen preceder a una instrucción de bifurcación condicional y modifican los flags. No hay que pensar que las instrucciones de salto condicional dependen de este repertorio, ya que lo único que hace el salto condicional es consultar los flags y salta si precede, pero no depende de ninguna instrucción de comparación. (de hecho cualquier operación aritmética realizada anteriormente a un salto condicional puede provocar que este "salte").

Nemotécnicos más frecuentes:

- ◆ compare: Resta los dos operando pero no almacena el resultado, solo modifica los flags.
- ◆ test: compara un cierto valor especificado con el 0.
- ◆ Instrucciones lógicas: Realizan operaciones booleanas "bit a bit" entre dos operando. Como las aritméticas también modifican los flags.

Nemotécnicos más frecuentes:

- ◆ and: el "y" lógico.
- ◆ or: el "o inclusivo" lógico.
- ◆ xor: el "o exclusivo" lógico.
- ◆ not: la negación lógica. (complemento a 1, no confundir con el cambio de signo "negate" que es el complemento a 2)

- ◆ Instrucciones de Desplazamiento: Pueden ser aritmético o lógico y pueden incluir o no rotaciones. Pueden ser de izquierda a derecha.

Nemotécnicos más frecuentes:

- ◆ shift: desplazamiento aritmético o lógico.
- ◆ rotate: rotación con o sin acarreo.
- ◆ Instrucciones de bits: Comprueban un bit del operando y su valor lo reflejan en el indicador de cero. Pueden poner un bit a 0 o complementarlo.

Nemotécnicos más frecuentes:

- ◆ Bit test: comprueba un bit.
- ◆ bit clear: comprueba un bit y lo pone a 0.
- ◆ bit set: comprueba un bit y lo pone a 1.
- ◆ Instrucciones de control: Permiten modificar la secuencia normal de ejecución de un programa, puede hacerse por salto condicional relativo o absoluto.

Se clasifican en cuatro grupos:

- > salto incondicional
- > salto condicional
- > Llamada a subrutinas
- > Gestión de las interrupciones

- ◆ saltos: Pueden ser condicionales o incondicionales, se suelen especificar como jump o brantch, y en el caso de los condicionales se suele llamar jcond o bcond donde cond es una o más letras que indican la condición que ha de cumplirse para que el salto se produzca.

-> Incondicional: salta sin comprobar ninguna condición.

Nemotécnicos más frecuentes: jump o brantch

-> Condicional: salta si la condición se cumple.

Nemotécnicos más frecuentes: jcond o bcond

- ◆ Llamadas a subrutinas: Invoca la ejecución de funciones anteriormente definidas.
Nemotécnicos más frecuentes: call (llamada) y ret (retorno)
- ◆ Gestión de interrupciones: Se usan para llamar a las rutinas de servicio de interrupción y esto se puede hacer por hardware o bien por software. Necesita una instrucción similar a return para retornar al contexto anterior pero restableciendo el estado de la máquina, para no afectar a la aplicación a la cual se interrumpió (iret).

- ◆ Instrucciones de E/S: Son instrucciones de transferencia salvo que el origen/destino de dicho flujo es un puerto de un dispositivo de E/S. Estas instrucciones pueden darse mediante dos alternativas:

->E/S "mapeada" en memoria: Los periféricos tienen direcciones asignadas de la MP por lo que no se necesitan instrucciones especiales y las operaciones se realizan con las ya vistas, como son: load, store y move.

->E/S independiente: Necesitan unas instrucciones especiales para indicarle al procesador que nos estamos refiriendo al mapa de direcciones de E/S, ya que este mapa y el mapa de memoria son disjuntos.

Nemotécnicos más frecuentes:

- ◆ input o read: Permite leer información de un puerto y trasladarla a memoria principal.
- ◆ output o write: Permite escribir información en un puerto de un dispositivo.
- ◆ test i/o: Lee información de control de un periférico.
- ◆ control i/o: Envía información de control hacia un periférico.

Instrucciones de control y misceláneas:

Nemotécnicos más frecuentes:

- ◆ halt: Detiene la ejecución del programa hasta que una interrupción arranca otro programa.
- ◆ wait: Sirve para detener la ejecución de un programa hasta que sucede un determinado evento que no es una interrupción (otra condición externa al primer programa).
- ◆ nop: No realiza ninguna operación, sirve para rellenar huecos en un programa o temporizar esperas.
- ◆ enable: Habilita las interrupciones.
- ◆ disable: Deshabilita las interrupciones.
- ◆ test and set: Se utiliza para la implementación de la exclusión mutua, esto es, que un procesador no pueda acceder a un determinado recurso que está siendo usado por otro procesador en ese mismo momento.

Se usan como semáforos, esto es, se declara una variable entera que tendrá el valor 0 si el recurso está libre y 1 si está siendo utilizado, de manera que si un procesador comprueba y el semáforo está en 1 tendrá que esperar hasta que este cambie a 0. (1 = s.rojo y 0 = s.verde)". Tomado de la dirección web: http://es.wikipedia.org/wiki/Conjunto_de_instrucciones, fecha 18-07-2017.

Ejercicios del tema 4: Programando el computador básico.

- a. Cuál es la diferencia entre lenguaje ensamblador y lenguaje de máquina.
- b. Explique el proceso de ciclo de programa
- c. Realice un programa en bajo nivel con binarios que efectúe la suma,
- d. resta y multiplicación de dos variables de valores conocidos

Actividad de la unidad 3

Diseñar un programa que permita aplicar los códigos de detección de errores

4.5. Pistas de Aprendizaje

Traer a la memoria: Para convertir sistemas de números debe usar el método rápido por agrupación de números binarios.

No olvide: Que las compuertas lógicas pueden ser de múltiples entradas y salidas.

Traer a la memoria: La tabla correspondiente a cada compuerta lógica resulta teniendo en cuenta su operador lógico único.

Tener en cuenta: Los teoremas de bool y Demorgan se usan para simplificar las expresiones algebraicas binarias.

Tenga Presente: Los circuitos secuenciales digitales tienen memoria.

Traer a la memoria: Los estados binarios 0 equivale a una amplitud de cero voltios y el estado 1 equivale a una amplitud de 5 voltios.

Traer a la memoria: Las instrucciones toman valores que equivalen a la función que debe realizar el microprocesador.

Tenga presente: La memoria es un dispositivo que almacena datos en forma binaria.

Traer a la memoria: El Comando contiene una instrucción que es ejecutada por el Procesador.

4.6. Glosario

Compuerta: Dispositivo semiconductor diseñado para realizar funciones binarias.

Lógico: Pasos consecuentes ordenados en función de una regla o ley establecida en los procesos informáticos.

Bajo nivel: Lenguaje de comunicación interpretativa entre el usuario y la máquina mediante símbolos y comandos dirigidos al hardware.

Digital: Sistema de operación mediante el uso de cantidades o valores que se originan al presionar un pulsador con los dedos básicamente 0 y 1.

Procesamiento: Conjunto de pasos necesarios que se requieren en la ejecución de un programa por parte de la unidad central de procesos.

Secuencial: Sucesión ordenada de instrucciones o interconexión de elementos para un determinado fin, en la realización de un programa o implementación de hardware lógico.

Registro: Dispositivo diseñado con flip-flop que es capaz de originar estados lógicos

Contador: Secuencia ordenada de números o situaciones lógicas en forma ascendente o descendente para un propósito digital.

Memoria: Dispositivo semiconductor hecho de flip-flop, que tiene la propiedad de almacenar datos en forma temporal o permanente.

Procesador: Microchip diseñado con transistores semiconductores, capaz de realizar operaciones basado en instrucciones.

Programa: Grupo secuencial de instrucciones para realizar operaciones en un procesador con algún objetivo predeterminado.

Dirección: Código Binario codificado para establecer un dato en una posición de memoria.

4.7. Bibliografía

- ◆ J.Tocci, Ronald, 1993, Sistemas digitales, quinta edición.
- ◆ Mano, M. Morris 1982, Logica Digital y Diseño de Computadores.
- ◆ Mcgraw-Hill, 1992, Electronica Practica, Primera edición en español.

4.8. Fuentes digitales o electrónicas

- ◆ <http://www.electronica2000.com/digital/codideco.htm>
- ◆ <http://www.monografias.com/trabajos7/regi/regi.shtml#conce>
- ◆ <http://members.fortunecity.es/deniro/unidad4.html>
- ◆ <http://www.fdi.ucm.es/profesor/jjruiz/WEB2/Temas/EC2.pdf>
- ◆ http://es.wikipedia.org/wiki/Modos_de_direccionamiento
- ◆ [http://msdn.microsoft.com/es-es/library/cc483387\(v=VS.71\).aspx](http://msdn.microsoft.com/es-es/library/cc483387(v=VS.71).aspx)
- ◆ <http://es.wikipedia.org/wiki/Perif%C3%A9rico>
- ◆ http://www.grupo-maser.com/PAG_Cursos/Step/step7/
- ◆ [Proyecto%20step7/paginas/contenido/automatas/A/a.5.htm](http://www.monografias.com/trabajos17/TiposTransmisi%C3%B3n_datos/TiposTransmisi%C3%B3n_datos.shtml#TiposTransmisi%C3%B3n_datos)
- ◆ http://www.monografias.com/trabajos17/TiposTransmisi%C3%B3n_datos/TiposTransmisi%C3%B3n_datos.shtml#TiposTransmisi%C3%B3n_datos
- ◆ http://www.monografias.com/trabajos/cpu/cpu.shtml#_Toc453341602
- ◆ http://es.wikipedia.org/wiki/Jerarqu%C3%ADa_de_memoria
- ◆ http://es.wikipedia.org/wiki/Tipo_primitivo
- ◆ <http://www.virtual.unal.edu.co/cursos/ingenieria/2000477/lecciones/010>
- ◆ [301.htm](#)

- ◆ <http://es.kioskea.net/contents/pc/processeur.php3>
- ◆ <http://www.monografias.com/trabajos17/arquitectura-computadoras/arquitectura-computadoras.shtml#regist>
- ◆ <http://proton.ucting.udg.mx/dpto/maestros/mateos/novedades/ensamblador/68HC11.html>
- ◆ <http://www.angelfire.com/al2/Comunicaciones/Laboratorio/codifica.html>