uc3m | Universidad Carlos III de Madrid

Grado en Ingeniería Informática
2018-2019

Apuntes

Tecnología de Computadores

Jorge Rodríguez Fraile[1]

[1] Universidad: 100405951@alumnos.uc3m.es | Personal: jrf1616@gmail.com

ÍNDICE GENERAL

Parte I

Teoría

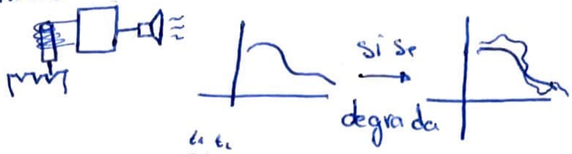
Introducción.

Digital = Lógica (0, 1)

No Analógico (Todos los valores existen)

Ejemplo: Sonido

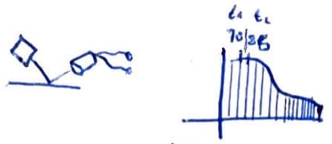
Vinilo



si se degrada

Aparece ruido con la degradación y campos magneticos.

DVD



t_1 t_2 70 85

Muchos pequeños que definen la onda

t_1 |1|0|1|1|0 70

t_2 |0|0|1|0|1| 85

si se dgrad

1 Por encima

0 Por debajo

Para evitar la perdida se duplia la f.

cap. persona   por si hay alguna   no poder

$20\,kHz + 10\% = 22\,kHz \cdot 2 = 44\,Hz$

Fourier (1960): Facilitó la forma de onda, como la suma de muchos senos y cosenos. (Como Taylor pero en ondas)

Transformada de Fourier (FFT) = Σ sen σ_i + cos σ

Sonido: .wav → $\boxed{FFT}$ → .mp3

Llamada: → $\boxed{\text{Sample}}$ → $\boxed{FFT}$ → ip envio → $\boxed{FFT^{-1}}$ → $\boxed{\text{Hold}}$ →

Con el algoritmo de Huffman $\boxed{\text{Zip}}$ nos permite en una emisora analogica guardar varias lógicas. (Busca coincidencias para reducir tamaño)

Lo analógico no es peor que lo lógico, juegan en terrenos distintos.

Tema 1.

Computador: Maquina que procesa información.

Sistemas de Numeración: Permite representar los números mediante dígitos.

Decimal	En base 10	0-9	
Binario	En base 2	0-1	
Octal	En base 8	0-7	
Hexadecimal	En base 16	0-F	

$172'8_{(10} = 1\cdot 10^2 + 7\cdot 10 + 2\cdot 10^0 + 8\cdot 10^{-1}$

Se leen digito a digito.

LSB → Least Significant Bit   CBA

MSB → Most Significant Bit   ABC   CBA

Sistema Binario.

$0_{(2} \to 0_{(10}$

$1_{(2} \to 1_{(10}$

$10_{(2} \to 2_{(10}$

000
001
010
011
100
101
110
111

$2^{10} \to 1024 \to$ 1 Kilo

$2^{20} \to$ 1 Mega

$2^{30} \to$ 1 Giga

$2^{40} \to$ 1 Tera

Multiplicar x2   0110 → 1100

Dividir /2   1010 → 0101

Decimal (10)	Binario (2)	Octal (8)	Hexadecimal (16)
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10

Conversiones entre sistemas de numeración.

$B_2 \to B_{10}$, $B_8 \to B_{10}$, $B_{16} \to B_{10}$

- Metodo del polinómio equivalente: Multiplicar cada digito por la base elevada a la posición que ocupa empezando en 0.

$$\underset{2\ 1\ 0\ -1}{abc,d}_{(2} = a \cdot 2^2 + b \cdot 2^1 + c \cdot 2^0 + d \cdot 2^{-1}{}_{(10}$$

$B_{10} \to B_2$, $B_{10} \to B_8$, $B_{10} \to B_{16}$

- Método de descomposición en pesos: Descomponer el numero decimal en suma de valores de potencias de la base

$$25_{(10} = 2^4 + 2^3 + 2^0_{(10} = 16 + 8 + 1_{(10} = 11001_{(2}$$

- Método de divisiones sucesivas por la base: Los cocientes se dividen por la base sucesivamente y los resto y ultimo cociente es el numero. Leido del final al comienzo.

$25_{10} = 11001_{(2}$

25 ⌊2 → 1, 12 ⌊2 → 0, 6 ⌊2 → 0, 3 ⌊2 → 1, 1

LSB ... MSB

$0'75_{(10} = 0'11_{(2}$ ← multiplicación sucesiva por la base (parte decimal)

$0'75 \cdot 2 = 1'5$

$0'5 \cdot 2 = 1'0$

$B_2 \leftrightarrow B_{16}$ de 4 en 4

$B_2 \leftrightarrow B_8$ de 3 en 3

$\underset{7}{\underline{111}}\ \underset{3}{\underline{011}}\ \underset{5}{\underline{101}}_{(2} = 735_{(8}$

Se agrupa de derecha a izq.

$\underset{2}{\underline{10}}\ \underset{E}{\underline{1110}}\ \underset{3}{\underline{0011}}_{(2} = 2E3_{(16}$

Al reves es poner digito a digito en binario

$11_{(8} = \underset{1}{\underline{1}}\ \underset{1}{\underline{001}} = 1001_{(2}$

T. Computadores T (2) 1/02/2019

Codigos Binarios:

- Binario natural : $1001_{BIN} = 1001_2$
- BCD: Se escribe digito a digito, 4bits, 0-9.

  $78_{(10} = \underbrace{0111}_{7}\ \underbrace{1000}_{8}\ _{BCD}$ No tiene porqué coincidir con el natural.

Cuando hacemos una lectura de datos, para evitar errores se utilizan codigos progresivos (todos son adyacentes, un unico cambio) y ciclicos (primero y ultimo progresivos)

1	0	0	100
1	0		100
1		1	101

- Jhonson: Progresivo y ciclico, agrupa 0. Con 3 bits solo hay 6 de las 2^3 combinaciones, por lo que es deficiente.

000	111
001	110
011	100

- Gray: Progresivo y cíclico, se forma haciendo espejo y 0 arriba 1 abajo. Con 3 bits se obtienen 2^3 combinaciones por lo que es eficiente.

1bit: 0, 1

2bits: 00, 01, 11, 10

3bits: 000, 001, 011, 010, 110, 111, 101, 100

Así sucesivamente.

Binario → Gray

$101_{(2} \rightarrow (1)(1+0)(0+1) = 111_{GRAY}$

$111_{(2} \rightarrow 100_{GRAY}$

Gray → Binario

$1011_{GRAY} \rightarrow (1)(1+0)(1+1)(0+1) = 1101_{(2}$

$abc_{GRAY} \rightarrow a\,d\,e$

$d = a+b \quad e = d+c$

Codigo detector de errores: Se envia junto al codigo para saber si es el correcto. Ej: Si el codigo tiene que ser par por el detector y no lo es, sé que no es el correcto.

DNI $\boxed{\text{Número}}\ \boxed{\text{Letra}}$ ← Detector

Código corrector de errores: Detecta y corrige el error. Al ser codigos suficiente- -mente largas al faltar o estar modificado algun valor el general se entiende.

Ejem: 1 Aunque no se entienda del todo
0 se distingue un codigo del otro.

Los más utilizados son los de paridad (por la sencillez) y CRC (por la eficiencia, el menor número de bits para detectar el mayor número de fallos), pero tambien está el de número de unos y el de número de transiciones (de 0 a 1 y de 1 a 0)

Algebra de Boole.

- Elementos: $\{0, 1\}$
- Operaciones: $\{+, \cdot, \bar{\ }\}$

Postulados:

Composición interna: $\forall a,b \in B \Rightarrow a+b \in B,\ a \cdot b \in B$

Elemento neutro: $a+0=a \quad a \cdot 1 = a$

Propiedad conmutativa: $a+b=b+a \quad a \cdot b = b \cdot a$

Propiedad distributiva: $a+b \cdot c = (a+b) \cdot (a+c) \quad a \cdot (b+c) = (a \cdot b)+(a \cdot c)$

Elemento inverso: $a+\bar{a}=1 \quad a \cdot \bar{a} = 0$

Propiedades fundamentales:

Dualidad: Toda ley tiene una dual, que se obtiene cambiando $0 \leftrightarrow 1$ y $+ \leftrightarrow \cdot$

Idempotencia: $a+a=a \quad a \cdot a = a$

$a+1=1 \quad a \cdot 0 = 0$

Involución: $\bar{\bar{a}} = a$

Absorción: $a+ab=a \quad a(a+b)=a$

Propiedad asociativa: $(a+b)+c = a+(b+c) \quad (a \cdot b) \cdot c = a \cdot (b \cdot c)$

Leyes de Morgan: $\overline{a+b} = \bar{a}\bar{b} \quad \overline{a \cdot b} = \bar{a}+\bar{b}$

Representación de funciones lógicas:

Expresión: $f(a,b) = a+b$ Tabla de verdad

a	b	f(a,b)
0	0	0
0	1	1
1	0	1
1	1	1

Función mintérmino (m_x)

1ª Forma Canónica

La posición de los unos, m_2, (suma de expresiones) que se multiplican.

0 negado/1 normal Ej: 010 $f(a,b,c) = \bar{a}b\bar{c} = m_2 \leftarrow$ 2 posición empezando a contar desde 0

Función maxtermino (M_x)

La posición de los ceros, M_2, sumandose las variables y 0 normal / 1 negado.

Ej: 010 $f(a,b,c) = a+\bar{b}+c = M_2 \leftarrow$ Contar desde 0 (Valor en binario del conjunto)

Primera forma canónica: Una función como suma de los mintérminos.

(más 1's que 0's) Ej: $f(a,b,c) = m_0+m_2+m_5 = \sum_{3}(0,2,5)$ ← nº de variables

Segunda forma canónica: Una función como el producto de los maxterminos.

(más 0's que 1's) Ej: $f(a,b,c) = M_1+M_3+M_4+M_6 = \prod_{3}(1,3,4,6)$ ← nº de variables

T. Computadores T (3) 1/02/2019

Puertas lógicas: Circuitos electrónicos que realizan las funciones básicas del Algebra de Boole.

- Identidad $z = a$

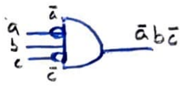
a	a
0	0
1	1

- Inversor o NOT $z = \bar{a}$

a	$\bar{a}$
0	1
1	0

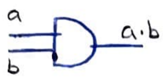
(Un circulo en una entra o salida indica negación) Tambien se puede usar inversor el

$z = \bar{a}b\bar{c}$



- AND $z = a \cdot b$ "y"

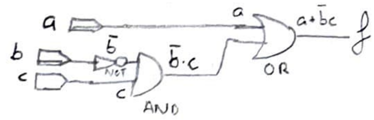
a	b	$a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1



- OR $z = a + b$ "o"

a	b	$a+b$
0	0	0
0	1	1
1	0	1
1	1	1

$f(a,b,c) = a + \bar{b}c$



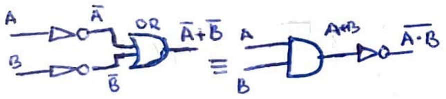
- NAND $z = \overline{a \cdot b} = \bar{a} + \bar{b}$

a	b	$\overline{a \cdot b}$
0	0	1
0	1	1
1	0	1
1	1	0

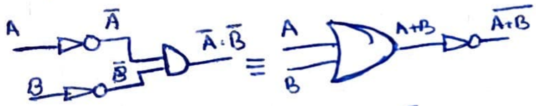
- NOR $z = \overline{a+b} = \bar{a} \cdot \bar{b}$

a	b	$\overline{a+b}$
0	0	1
0	1	0
1	0	0
1	1	0

Equivalente:



Equivalente:



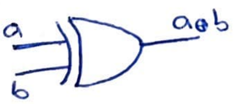
Inversor NAND

Inversor NOR:

- XOR $z = a \oplus b = \bar{a}b + a\bar{b} = (\bar{a}+\bar{b})(a+b)$

Paridad de 1's

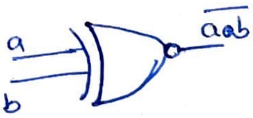
a	b		$a \oplus b$
0	0	par	0
0	1	imp	1
1	0	imp	1
1	1	par	0



XNOR $z = \overline{a \oplus b} = ab + \bar{a}\bar{b} = (\bar{a}+b)(a+\bar{b})$

Paridad de 1's

a	b		$\overline{a \oplus b}$
0	0	par	1
0	1	imp	0
1	0	imp	0
1	1		1



Mirar en sucio ejemplo frenado tren.

Metodo de los mapas de KARNAUGH:

Optimización basada en la adyacencia, hasta 6 variables manualmente.

Dos terminos son adyacentes si son identicos excepto por un termino, en uno negado y otro no.

$\bar{a}b\bar{c} = \bar{a}\bar{b}\bar{c}$ $\quad \bar{a}\bar{c}$ identico y $\bar{b} \neq b$

$f(a,b,c) = \sum_3 (0,1,2,3,7) = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}bc + abc$

$\bar{a}\bar{b} \cdot \cancel{(\bar{c}+c)}^{U} + \bar{a}b \cdot \cancel{(\bar{c}+c)}^{U} + bc \cdot \cancel{(\bar{a}+a)}^{U}$

$\bar{a} \cdot \cancel{(\bar{b}+b)}^{U} + bc$

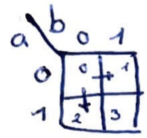
$\bar{a} + bc$

Mapas de KARNAUGH

Presenta la tabla de una forma que los terminos adyacentes son continuos.

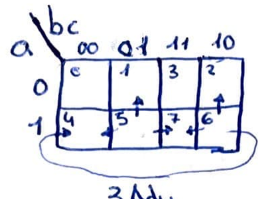
- Se numeran con GRAY
- En un mapa de n variable hay n casillas adyacentes.
- Una casilla por combinación

Dos variables:



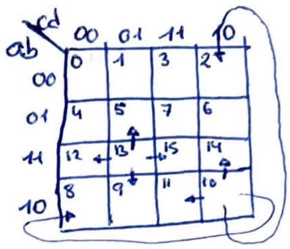
2 Ady.

Tres variables:



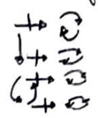
3 Ady

Cuatro variables:

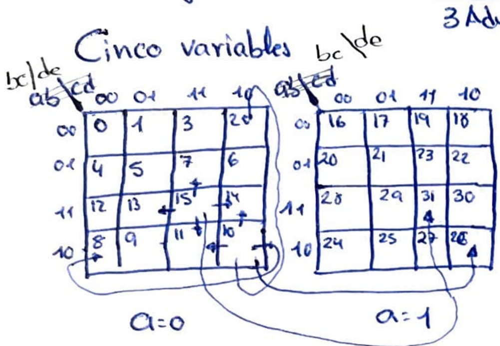


4 Ady

Cuidado gray



Cinco variables



a=0 a=1

5 Ady. 4 directos y el otro en la tabla opuesta

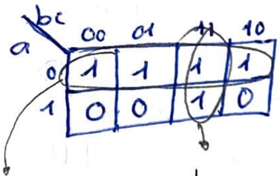
Representación de una funcion en el Mapa de KARNAUGH

Se marcan las casillas que corresponden a los minterminos "1" o los maxterminos "0" de la función

Los terminos independientes son comodines

$f(a,b,c) = \sum_3 (0,1,2,3,7)$

$= \prod_3 (4,5,6)$



1º Buscamos grupos de casillas adyacentes, ~~multiplos~~ potencias de 2: Empezando los menores

G. 2 casillas ady. elimina 1 variable

G. 4 casillas ady. elimina 2 variables

8 — 3

16 — 4

$f(a,b,c) = \bar{a} + bc$

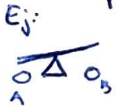
2º En los grupos, mirar que variable no varía, en el ejemplo no cambia la A, es 0 para todo el grupo horizontal, por lo que es $\bar{A}$ al ser cero y no variar. En la vertical 11 no varia BC

En los mapas de karnaugh las X son comodines, si falta un uno para hacer grupos o este queda suelto. (no solo grupo de X)

1	1
1	1
X	X
1	X

Las X representan imposibles, irreales, indeterminado...

Ej:



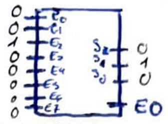
A	B	Pos.
0	0	A=B
0	1	A<B
1	0	A>B
1	1	X

En la funcion las X se ponen con $+\Delta_4 (9,13,15)$.

Ej: $f_1 = \sum_4 (1,7,11,13,14,15) + \Delta_4 (3,5,6,9,10,12)$

Tema 3: Circuitos combinacionales.

- **Codificadores:** Circuito combinacional que permite transformar un nivel activo en una de sus entradas en un valor codificado. M entras: N salidas



$E_2 \Rightarrow S = "010"\ (=2)$

Sin prioridad: Solo una entrada puede estar activa, si se activan varias la salida será errónea. COD 8:3

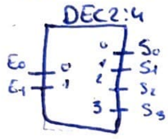
Enable Output: Sirve para diferenciar el caso de activarse E_0 y el de nada activo.

Con prioridad: Si se activan varias entradas a la vez, dan prioridad a una de ellas.

- Al bit más significativo: E_1, E_5 Se activa E_5
- Al bit menos significativo: E_1, E_5 Se activa E_1

Enable Input: Sirve para habilitar, 0 hace que no haya salidas y 1 el funcionamiento normal.

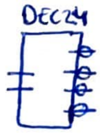
- **Decodificadores:** Circuito combinacional que transforma un valor codificado en la activación de la salida correspondiente a dicho valor. Inversa del codificador.



E_1	E_0	$S_3 S_2 S_1 S_0$
0	0	0001
0	1	0010
1	0	0100
1	1	1000

$S_0 = \overline{E_1}\,\overline{E_0}$
$S_1 = \overline{E_1} E_0$
$S_2 = E_1 \overline{E_0}$
$S_3 = E_1 E_0$

Funciones lógicas con mintérminos

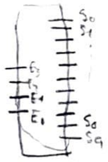


E_1	E_0	$S_3 S_2 S_1 S_0$
0	0	1110
0	1	1101
1	0	1011
1	1	0111

Implementación de funciones lógicas con descodificadores:

$f = \bar{a}b + ab = S_1 + S_3$

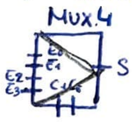
Decodificador BCD-decimal: Descodifica un valor en BCD a diez salidas 0-9



Decodificador BCD-7segmentos: 10 entradas de 16 posibles, para los números 0-9

- **Multiplexores:** Circuito que permite seleccionar una de las entradas y copiar su valor a la salida. La selección depende del valor de las entradas de control.

N entradas $= 2^n$ n entradas de control MUX4 → 2:4

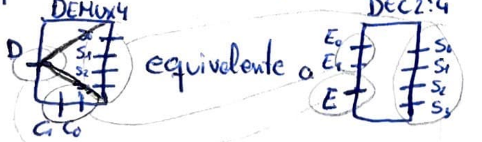


C_1	C_0	S
0	0	E_0
0	1	E_1
1	0	E_2
1	1	E_3

Si hay menos entradas de control que variables tiene la función, se agrupan en la tabla de verdad.

Ejem: Elegir dispositivo $E_0\ E_1$

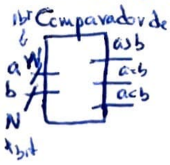
- **Demultiplexores:** Circuito que copia el valor de la entrada de datos en la salida indicada por el valor de las señales de control. Opuesto al multiplexor. Equivalente a descodificadores.



Ejem: Entrada de audio y elegir altavoces delanteros, traseros ...
S $S_0\ S_1\ S_2$

- **Comparadores:** Circuito que permite determinar si dos datos son iguales, mayor o menor el uno que el otro.

N nº de bits de los datos



a	b	a=b	a>b	a<b
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

1bit:
$f_{a=b} = \overline{a \oplus b}$
$f_{a>b} = a\bar{b}$
$f_{a<b} = \bar{a}b$

3bits:
$f_{a=b} = \underbrace{\overline{a_0 \oplus b_0}}_{a_0=b_0} \cdot \underbrace{\overline{a_1 \oplus b_1}}_{a_1=b_1} \cdot \underbrace{\overline{a_2 \oplus b_2}}_{a_2=b_2}$
$f_{a>b} = \underbrace{a_2\bar{b_2}}_{a_2>b_2} + \underbrace{(\overline{a_2 \oplus b_2})}_{a_2=b_2}\underbrace{a_1\bar{b_1}}_{a_1>b_1} + \ldots$

20/02/2019

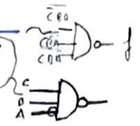
Tema 4

Pasar formulas a circuito de NAND

Mintérminos en Karnaugh (1)

$f_1 = \bar{C}BA + C\bar{B}\bar{A} + CB\bar{A}$

$f_1 = \overline{\overline{\bar{C}BA} \cdot \overline{C\bar{B}\bar{A}} \cdot \overline{CB\bar{A}}}$



$\boxed{\cdot\cdot} + \boxed{\cdot\cdot} + \boxed{\cdot\cdot}$ Inicial

$\Downarrow$

$\overline{\overline{\boxed{\cdot\cdot} + \boxed{\cdot\cdot} + \boxed{\cdot\cdot}}}$ Doble negación

$\Downarrow$

$\overline{\overline{\boxed{\cdot\cdot}} \cdot \overline{\boxed{\cdot\cdot}} \cdot \overline{\boxed{\cdot\cdot}}}$ De Morgan

Pasar formulas a circuito NOR

Maxtérminos en Karnaugh (0)

$f_1 = (\bar{C}+B+A) \cdot (C+\bar{B}+\bar{A}) \cdot (C+B+\bar{A})$

$f_1 = \overline{\overline{(\bar{C}+B+A)} + \overline{(C+\bar{B}+\bar{A})} + \overline{(C+B+\bar{A})}}$

$\boxed{++} \cdot \boxed{++} \cdot \boxed{++}$ Inicial

$\Downarrow$

$\overline{\overline{\boxed{++} \cdot \boxed{++} \cdot \boxed{++}}}$ Doble negación

$\Downarrow$

$\overline{\overline{\boxed{++}} + \overline{\boxed{++}} + \overline{\boxed{++}}}$ De Morgan.

Aritmetica Binaria.

Codificar signo: Segun el MSB, ocupa un bit, por lo que se pierde uno:

Positivo → Empieza por 0

Negativo → Positivo → CA_1 → Complemento a 2

$\underset{+3}{011} \xrightarrow[\text{XOR}]{\text{invertir}} 100 \xrightarrow{+1} \underset{-3}{101}$

Con n bits → 2^n números → $2^{n-1}-1 \sim 0 \sim -2^{n-1}$ (Por el 1 o 0 del signo)

Suma binaria.

← Añadimos 0 para que no salga 1 del ultimo MSB

```
  0 0 1 0 0 1
+ 0 0 1 1 0 1
-------------
  0|1 0 1 1 0
```

```
A=-9   1 0 1 1 1
     +
B=4    0 0 1 0 0
---------------
       1|1 0 1 1
       negativo
```

9 → 01001

CA_1 → 10110

CA_2 → 10111 ← -9

→ $\overset{CA_1}{00100}$ + 1 = $\overset{CA_2}{00101}$ → -5

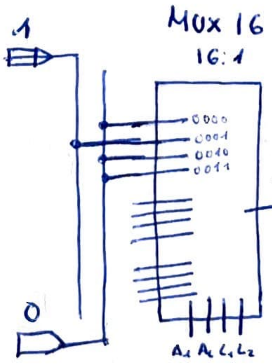
Multiplicación binaria.

```
  0|1 0 1 1   +11
    0|1 0 1   +5
-------------
+   0 1 0 1 1
        0
  0 1 0 1 1
-------------
  0|1 0 1 1 1   +55
```

Tabla de verdad en Multiplexor.

MUX 16

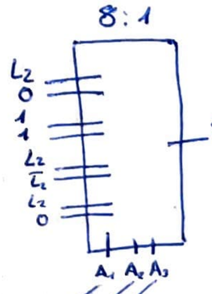
16:1



A_1	A_2	L_1	L_2	S_1
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0

MUX 8 para 3 entradas → Agrupar entradas en función de entradas; La salida es 1, 0 o suelta

8:1



A_1	A_2	L_1	L_2	S_1	Input
0	0	0	0	0	L_2
0	0	0	1	1	
0	0	1	0	0	0
0	0	1	1	0	
					1

16 entradas y tiene que haber 8

16/8 = 2 entradas por grupo

Representación de números reales.

- Punto fijo: La coma decimal se considera fija en un punto, ejemplo 20bits (1 signo) parte entera y 12 para la parte decimal.
- Punto flotante: El numero se descompone en mantisa $x, \boxed{}$ y exponente $\cdot 2^{E}$
- Standard IEEE 754: Precisión simple (32 bits)

  1 bit signo (0pos/1neg), 23 bits mantisa y 8 bits exponente (-127, 128)

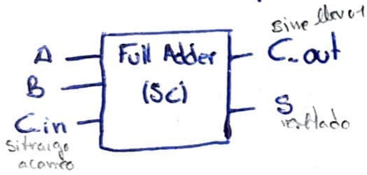
  $N = (-1)^{s} \cdot 2^{127+E} \cdot 1.M$

  $-7,625_{(10} = -111'101_{(2} = -1'11101 \cdot 10^{2}_{(2} = 1\ 10000001\ 11101000000000000000000$ (S, E, Mantisa)
- Standard IEEE 754: Precisión doble (64 bits)

  1bit signo, 52 bits mantisa y 11 bits exponente. 1023+E

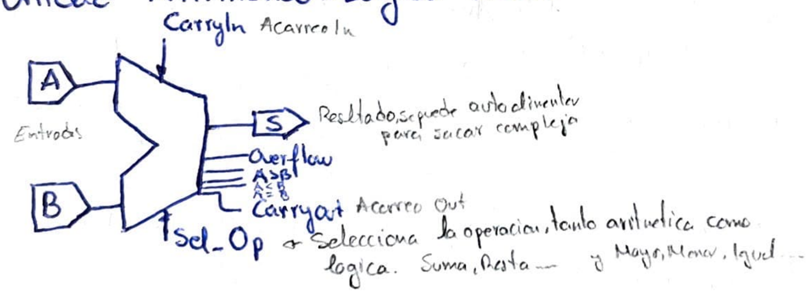
Circuitos combinacionales II

Sumador completo.



Semisumador (Sin traer acarreo)

Unidad Aritmetico-Logica (ALU)

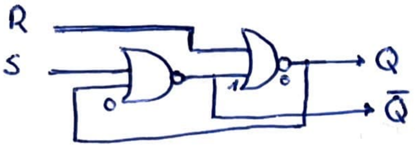


15/03/2019

Biestables.

Secuenciales: La salida no depende solo de las entradas, sino tambien de las salidas previas.

Los más sencillos:



R	S	Q_{T+1}
0	0	Q
0	1	1
1	0	0
1	1	Inestable No funciona

En el biestable JK

J	K	Q_{T+1}
0	0	Q
0	1	1
1	0	0
1	1	$\overline{Q}$

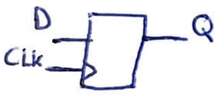
Segun el tiempo:

- Asíncronos: Sin tiempo (no depende de)
- Síncronos: Funcionan segun un reloj (CLK) ...

Nivel

Flancos:
- Subida
- Bajada

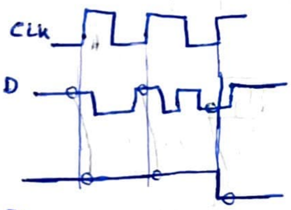
Biestable tipo D: $\boxed{Q_{T+1} = D}$



Para latch o buffer guarda con un relojato

	D	Q_{T+1}
Si	0	0
Si	1	1
No	X	Q

Almacena cuando no está subiendo.
Almacena lo anterior en 0 a empezar a subir en CLK



Da igual lo que pase entre subidas.

Biestables con entradas asíncronas.

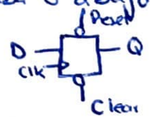
Ignora el reloj (CLK)

Clear: Inicialización a 0 asíncrona. normalmente al principio para limpiar lo anterior.

Preset: Inicialización a 1 asíncrona.

* Normalmente se activan con 0 (nivel bajo)

Son excepcionales sus activaciones, se dibujan arriba o abajo



* Un circuito es sincrono, si todos los biestables tienen el mismo reloj (CLK)

Biestable tipo T: $Q_{T+1} = T \oplus Q = \overline{T}Q + T\overline{Q}$

Para contadores

T	Q	Q_{T+1}
0	1	1
1	0	1
1	1	0
0	0	0

Memoriza con 0 en T el valor de Q y con 1 el inverso.

Tecnología de Computadores
T (6) 20/03/2019

Circuitos secuenciales síncronos.

Maquinas de estado finito (FSM)

Hay que marcar el origen
Solo se usa reset y preset en situaciones controlada

X Entradas   Z Estados   λ Funciones de salida.

Y Salidas   δ Funciones de estado

- Moore: Dependen unicamente del estado ← Las salidas

$Z = \delta(X, Z)$ $Y = \lambda(Z)$ No aparece el reloj.

(Estado 1 / Salida 1) → Entrada   Salida es la del origen.

- Mealy: Las salidas dependen de los estados y de las entradas.

$Z = \delta(X, Z)$ $Y = \lambda(X, Z)$ No aparece el reloj

(Estados 1) → Entrada 1 / Salida 1   Salida de la flecha

Pasos a seguir

1° Obtener el diagrama de estados. → Según enunciado

2° Codificación de estados. (no es unica)

	Q_1	Q_0
E_0	0	0
E_1	0	1

nº estados $\leq 2^{\text{nº biestables}}$

3° Obtener tablas de salidas y de transiciones de estados. Con las entradas y estados vamos completando Q_1^+, Q_0^+, T_1, T_0

4° Tabla inversa de biestables (o excitación) T_1 y T_0

5° Obtener funciones de salidas. Z

6° Obtener funciones de estado. T_0 T_1

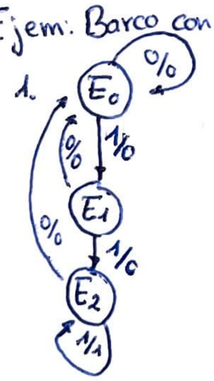
7° Implementación con biestables. Con las funciones

Para forzar que empiece en algun estado, forzamos con reset y preset el estado

	a preset	a reset
E_0	1	0
	Q_1	Q_0

Ejem: Barco con alarma a los 3 1's

1.



2.

	Q_1	Q_0
E_0	0	0
E_1	0	1
E_2	1	0

3 estados $\leq 2^2$

3. Entrada / Estados cod. / Los estados a los que lleva la entrada. / Con un biestable T si Q_1^+ es 0 y Q_1 a 0 → ¿T_1?

X	Q_1	Q_0	Q_1^+	Q_0^+	4. T_1	T_0	Z
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	X	X	X	X	X
1	0	0	0	1	0	1	0
1	0	1	1	0	1	1	0
1	1	0	1	0	0	0	1
1	1	1	X	X	X	X	X

Q	Q⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

5.

T_1

$x \backslash Q_1Q_0$	00	01	11	10
0	0	0	X	1
1	0	1	X	0

$T_1 = XQ_0 + \overline{X}Q_1$

$T_0 = X\overline{Q}_1 + Q_0$ ← otra tabla

6.

Z

$x \backslash Q_1Q_0$	00	01	11	10
0	0	0	X	0
1	0	0	X	1

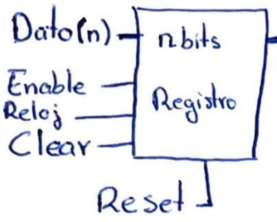
$Z = XQ_1$

7. Poner dos biestables con su reloj, en cada uno una T y la salida final Z

27/03/2019

Tema 7: Registros y Contadores.

Registro: Circuito digital con dos funciones básicas, almacenamiento de datos y movimiento de datos. Colección de 2 o más biestables tipo D (casi siempre) con entrada comun que almacenan una serie de bits.



Guarda datos (Dato(n)) cuando enable es 1, clear 0 (se activa con 1) y llega a un flanco de subida. Si enable es 0, lo mantiene.

SISO → Entrada y Salida en serie, entran y salen 1 dato por flanco ↑

SIPO → Entrada serie y salida paralelo, entran lento y salen todos a la vez.

PISO → Entrada paralelo y salida serie, entran en un relojazo y salen 1 a 1

PIPO → Entrada y salida en paralelo, entran y salen todos a la vez.

Registro universal de desplazamiento.

Q(n) → Dato salida. N bits en paralelo o El bit 0 en serie

CSDI → Dato carga serie. Desplazamiento a la izquierda. El ultimo que pasa es el izq

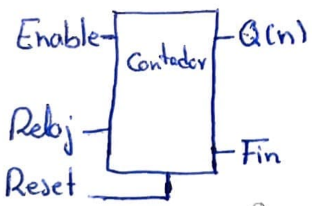
CSDD → Dato carga serie. Desplazamiento a la derecha. El ultimo para que pasa es el der

Modo → Modo de operación ← Izq / Derech / Paralelo

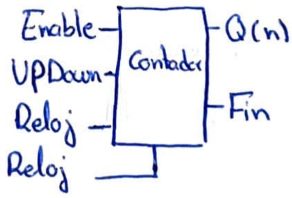
D(n) → Dato entrada N en paralelo o El bit 0 en serie.

Contador: Se caracterizan por modulo de X (desde 0 hasta X-1)
Síncronos

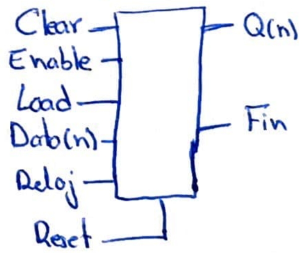
- básico



- Ascendente / Descendente



- Precarga / Clear sincrono



Si enable 1, suma 1 en cada flanco. (→ Q)

Si enable 0, mantiene el número.

Fin pasa un 1 cuando llega a X-1 (desde 0)

Si Up 0 1 2 3 4 | 0 1

Si Down 0 4 3 2 1 | 0 4

Cuando Load 1, se cambia el valor de Q(n) al de Dato(n) ~~4~~ 3 | 4 5 3

Ejemplo de Maquina de estados en sucio.

Tema 8: Memorias.

Memoria: Dispositivo de almacenamiento masivo de información. Componente fundamental de un sistema digital.

Tipos de memoria, segun la propiedad física:

Magnetica: Patrones sobre una superficie cubierta de material magnetizable. 

Opticas: Se graba con laser muescas minusculas, se leen con la reflexión de un laser. 

Semiconductor: Circuitos electricos.

- RAM (Random Access Memory): Lectura y escritura.
- ROM (Read Only Memory): Solo lectura.

Características de las memorias:

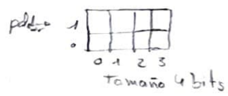
- Capacidad: Cantidad de información capaz de almacenar.

  Número de palabras: Normalmente potencias de 2. $2^{10} = 1k$ $2^{30} = 1G$ $2^{20} = 1M$ $2^{40} = 1T$

  Tamaño de palabra: Número de bits que se pueden aceeder de una vez.

  1 byte = 8 bits   Potencias de 2 normalmente

  Capacidad = nº x tamaño $16M \times 8 = 16 \cdot 2^{10} \cdot 8$



- Tipo de acceso:

  Secuencial: Sólo se puede acceder en un orden especifico.

  Aleatorio: Se puede acceder en cualquier orden.

- Permanencia:

  No volatil: Mantiene la información almacenada aunque se desconecte.

  Volatil: Se borra, si la fuente de alimentación se desconecta.

  Dinámica: Pierde información con el tiempo, aunque esté conectada. Necesita refresco periodico de la información.

- Coste/bit
- Consumo

No hay una mejor que la otra, depende el uso que le demos a la memoria.

Jerarquia de Memorias →

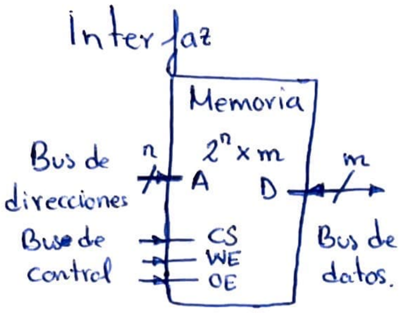
- Coste/bit (− arriba, + abajo) | + Velocidad de acceso − (+ arriba, − abajo)

Semi:
- Registro (Más caro, pero rápido)
- Memoria cache
- Memoria principal

Mag:
- Disco duro
- Almacenamiento masivo.

Mirar ciclo de lectura y escritura.

Interfaz



Bus de direcciones: Indica la posición a la que se desea acceder.
n bits entran, 2^n = nº de palabras

Bus de datos: Proporciona el dato.
m bits, el tamaño de la palabra.
Escritura entra, lectura sale o puede ser bidireccional.

Bus de control: Señales que controlan la operación de la memoria.
- CS (Chip Select) o CE (Chip Enable): Habilita el acceso. (bus de datos)
- R/W (Read/Write) o WE (Write Enable): Selecciona lectura o escritura.
- OE (Output Enable): Habilita la salida de datos. (bus de datos)
- RAS (Row Address Strobe) y CAS (Column Address Strobe): En memorias cor organización 3D

Memorias RAM:

Memorias de lectura y escritura

Dos tipos: Estatica SRAM: Se almacena en un biestable cada bit.
Dinámica DRAM: Cada bit se almacena en un condensador.

El más usado es DDR (Double Data Rate), organización 3D, accede a los datos en los dos flancos, ancho de 64bits y van evolucionando DDR, DDR2, DDR3, DDR4.

Memorias ROM:

Memoria de solo lectura. y no volatil.

Tipos: No programables: Contenidos fijados en la fabrica.
Programable: El usuario puede fijar los contenidos.
Borrables o reprogramables: Puede borrarse para almacenar otros contenidos.

Programación: Utiliza un mecanismo físico diferente que la lectura, es más lenta.

Expansión de memoria: (En los dibujos, he omitido el control)

Expansión del bus de datos. Para la misma entrada, pero todas las ~~entradas~~ salidas separadas 

Expansión del bus de direcciones. Las mismas salidas y entrada, pero se activa una para una parte y otra para otra (multiplexor) 

Expansion de bus de datos y direcciones (Las dos anteriores)

Mapa de Memoria

Grupos que dan una partes de salidas, pero todos la misma entrada, depende el multiplexor

Direccion → hex bin → A16 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 17 en bin, 5 en hexa

ROM (64k) 00000h 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
... 0 1
... 1 0
... 1 1
0FFFFh ← 1 menos que, 0 1 1 ...

RAM (32k) 10000h ← $64k = 2^6 \cdot 2^{10} = 2^{16}$ 1 ... 1 ... 1
1 0 .... 0 .. 0
17FFFh ← 1 men.

RAM (32k) 18000h ← $32k = 2^5 \cdot 2^{10} = 2^{15}$ 0 8 ... 0
0 1000 .. 0 0 Sumando el anterior
1FFFFh ← el maximo o sumar (08000h − 1) a 18000h 1 1 ... 1 .. 1

Hay tantos bits como el exponente de la sumas → $128k = 2^7 \cdot 2^{10} = 2^{17}$ 17 líneas de dirección

T : Introducción a los sistemas digitales:

Los sistemas digitales procesan información digital, de acuerdo con un algoritmo determinado.
Para diseñarlos de forma más abstracta, se utilizan los lenguajes de Descripción de Hardware. Como el VHDL *+nivel de abtracción que dibujar puertas.

Estructura:

Ruta de datos:

Conjunto de unidades funcionales que procesan datos.

- ALU: Realizan operaciones basicas. F → Selección de operación; SR → Información (acarreo, ...)
- Registros y memorias: Almacenan datos temporales. RFx → Registro Fuente x; RD → Registro Destino; OE → Salida al exterior. (Habilita); SE → Selección Fuente (Mux)
- Buses: Conectan los elementos
- Multiplexores: Selecciona los datos que se procesan en cada momento.

Unidad de control: Determina la correcta secuenciación y utilización de las operaciones sobre los datos.

- Maquinas de estado.
- Contadores
- Registros y memorias de datos de control.

Circuitos integrados:

- Componente discreto: Puertas logicas 74xx 54xx
- Circuitos integrados a medida: Para uno circuito especifico ASIC
- Circuitos programable: Circuito que solo utiliza parte de una generica. FPGA, PLD, CPLD
- Microprocesadores.

Tecnologias digitales.

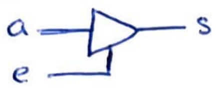
El nivel logico (1 o 0) se representan con un nivel de tensión, 1 (5V, 3V, 2.5V, etc). y 0 (0V) se detecta si está por encima de un nivel. Para evitar ruido mayor tension, pero mayor consumo.

Características de las tecnologias digitales.

- Magen de temperatura de operación.
- Tensión de alimentación.
- Margen de ruido.
- Retardo de conmutación: Idealmente instantaneo, pero al haber tantas limita la velocidad de operacion

- Consumo: Estatico, por estar alimentada, y dinamico, al conmutar 1→0 o 0→1. CMOS poco estatico. El consumo produce calor.

Buffer triestado:



e	a	s	
0	0	Z	> Alta impedancia.
0	1	Z	
1	0	0	
1	1	1	

Utiles para permitir varias conexiones a un mismo punto evitando corto circuitos.

Microprocesador: Orientado a realizar una amplia variedad de algoritmos.

- Unidad Central de Procesos ≡ CPU: Realiza las operaciones necesarias

  Ruta de datos general

  Unidad de control programable. Almacena en una memoria las palabras de control codificadas. En un ciclo de reloj se lee y descodifica. (Sale de RI, registro de instrucciones) La instrucción en cada momento la determina en contador (PC) ...counter a un decodec que envía las señales.

- Memoria, para almacenar información.
- Unidades de entrada/salida, para comunicarse.

Arquitectura Neuman: La memoria de datos y de programa es la misma, hay que tener cuidado no borrar el programa.

Arquitectura Harvard: Cada memoria por separado, una en la ruta de datos y otra en la unidad de control

Funcionamiento elemental:

Ciclo de instrucción: 2 fases que se pueden hacer en un solo ciclo de reloj dependiendo de la complejidad.

Busqueda de la instrucción: Cargar la instruccion en el IR

Ejecución de la instrucción: Descodifica y configura la ruta de datos para la operación.

Tipos de instrucciones:

Transferencia de datos, entre registros, registro a memoria,...

Aritmético-lógicas

Salto y bifurcación, cambios en la secuencia de ejecución.

Formato: Codigo de operación + Operandos variables ambas.

Modos de direccionamiento: Modos de indicar operandos.

Inmediato: Se indica directamente el valor.

Directo por registro: Indica un registro que lo contiene

Directo por memoria: Indica una posición de la memoria. para el operando.

Indirecto: Indica un registro, que contiene la dirección de memoria para el operando

Lenguaje ensamblador: Como los codigos de intrucción son poco manejables, se escriben mediante nombres simbolicos (R2 R1) los operandos y nemoticos (ADD MINUS) las intrucciones, y un programa ensamblador lo traduce al codigo correspondiente.