SISTEMA DE NUMERACIÓN

CONVERSIÓN DECIMAL A BINARIO (PARTE DECIMAL)

$(0,6875)_{10} \rightarrow$ BASE 2 $\qquad (0,6875)_{10} = (0,1101)_2$

$0,6875 \cdot 2 = 1 + 0,375$
$0,375 \cdot 2 = 0 + 0,75$
$0,75 \cdot 2 = 1 + 0,5$
$0,5 \cdot 2 = 1 + 0$

EQUIVALENCIA DE BINARIA A HEXADECIMAL

HEXA	BINARY	HEXAD	BINARY	HEXAD	BINARY	HEXAD	BINARY
0	0000	4	0100	8	1000	c	1100
1	0001	5	0101	9	1001	d	1101
2	0010	6	0110	a	1010	e	1110
3	0011	7	0111	b	1011	f	1111

CALCULAR EL COMPLEMENTO A 2 DE UN NÚMERO

- AGREGAR 0 HASTA COMPLETAR EL REGISTRO
- NEGAR BIT A BIT TODOS LOS DIGITOS Y SUMAR 1

- HEXADECIMAL A BINARIO (32 BITS)

0XABCDEF00 = 1010 1011 1100 1101 1110 1111 0000 0000

- BINARIO A HEXADECIMAL

$\underbrace{1110}_{E},\underbrace{0111}_{7},\underbrace{1000}_{8},\underbrace{0011}_{3} = E783$ (SI NO HAY EXACT. 4 AGREGAR 0 A IZQ)

- DECIMAL A BINARIO CON SIGNO NEG = 1, POS = 0

$(-76)_{10}$
- PASAMOS |-76| A BINARIO = 1001100
- RELLENAR CON 0 A LA IZQ. HASTA COMPL. EL REGISTRO 01001100
- HACER COMPLEMENTO a DOS 10110100 (NEGAR UAO Y SUM 1)

- BINARIO EN COMPLEMENTO 2 A DECIMAL
  - POSITIVO → NO HACEMOS NADA, LO EXPRESAMOS NORMAL A DECIMAL
  - NEGATIVO → 10010110 HACER COMP. A 2 01101010
    PASAR EL RES A DECIMAL Y AGREGAR EL SIGNO MENOS $(-106)_{10}$

EL RANGO DE NÚMEROS QUE ADMITE UN REGISTRO → -2^{n-1} A $2^{n-1}-1$

SUMA BINARIA

$$\begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array} \qquad \begin{array}{r} 1 \\ +1 \\ \hline 11 \end{array}$$

1 ACARREO

RESTA BINARIA

HACEMOS EL COMPLEMENTO A2 DEL SEGUNDO OPERANDO Y SE LO SUMAMOS AL PRIMERO

FLOATING POINT

REPRESENTACIÓN PARA NÚMEROS NO ENTEROS (COMO UNA ESPECIÉ DE NOT. CIENT.)

$\pm 1,xxxx_2 \times 2^{yyyy}$ $(-1\underbrace{,0000110}_{7},0101)_2 \equiv -1.00001100101 \times 2^7$

STANDARD IEEE 754 FLOATING POINT FORMAT

|S| exponent |Fraction|

$x = (-1)^S \times (1 + \text{Fraction}) \times 2^{(\text{exp} - \text{bias})}$

SING BIAS → 127

DOUBLE BIAS → 1023

DECIMAL A IEEE754

1. ENCONTRAR EL BIT SIGNO
2. PASAR EL NÚMERO A BINARIO Y NORMALIZAR
3. SUMAR EL BIAS AL EXPONENTE Y CONVERT. A BINARIO
4. ENCONTRAR LA PARTE FRACCIONARIA
5. CONFORMAR EL NÚMERO

EJEMPLO

263,3 → IEEE754

(1) BIT SIGNO = 0 (POS)

(2) $263,3 = 100000111,01001 \times 2^0$

$263,3 \text{(NORMAL)} = 1,00000111 0\ldots \times$

(3) $127 + 8 = 135 = 10000111$

(4) LA PARTE FRACCIONARIA SON 23 BITS Q' SIGUEN DESP. DE LA (,) EN EL NORMAL

00000111010011001100110011

(5) BIT SIGNO - EXPONTE _ PARTE FRACCIONARIA

*cortar en 23 bits

0 10000111 00000111010011001100110011001100110

IEEE 754 A DECIMAL

1. DIVIDIR LOS BITS EN 3 PARTES RESPETANDO EL FORMATO
2. ENCONTRAR EL BIT DE SIGNO
3. ENCONTRAR EL EXPONENTE Y -BIAS
4. DESNORMALIZAR EL NÚMERO
5. PASAR A DECIMAL

EJEMPLO

01000011100000111010011001100110

(1) 0 (signo) 10000111 (exponente) 00000111010011001100110 (fraction)

(2) BIT SIGNO = 0 => POSITIVO

(3) $10000111 = (135)_{10} \Rightarrow 135 - 127 = 8$

(4) Agregar el uno que se elimina al normalizar y desplazar "," en el sentido

$1,00000111010011001100110 \times 2^8$

$100000111,0100110011001100110$ (*)

(5) (*) = 263,2998770

INFINITE AND NaNs

INFINITE → EXPONENT = 11,..,1 FRACTION = 00,..,0

NaN (RES. INDEF. ej 0/0) → EXP = 11,..,1 FRACTION ≠ 00,..,0

ALGEBRA DE BOOLE

ELEMENTOS BÁSICOS

- SEA B UN CONJUNTOS DE ELEMENTOS
- $B=\{a,b,c,\dots\}$ Y SU CONJUNTO $\{0,1\}$
- SEA ϕ UN CONJUNTO DE OPS EN B CON RESULTADOS EN $\{0,1\}$
- $\phi=\{+,\cdot,'\}$

$\cdot$ = AND = — F_1 $F_1=X\cdot Y$

$+$ = OR = — F_2 $F_2=X+Y$

$'$ NOT = — F_3 $F_3=X'$

X	Y	F_1	F_2	F_3
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

BÚFER = X — F $F=X$

NAND = — F_4 $F_4=(X\cdot Y)'$

NOR = — F_5 $F_5=(X+Y)'$

X	Y	F_4	F_5
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

OR EXCL = XOR — $F_6=X\cdot Y'+X'\cdot Y$ $=X\oplus Y$

NOR EXCLU = — F_7 $F_7=XY+X'Y'$ $=(X\oplus Y)'$

X	Y	F_6	F_7
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

AXIOMAS

- $a\in B$ y $b\in B \Rightarrow a+b\in B$, $a\cdot b\in B$
- $\exists\, 0\in B \mid a+0=a$
- $\exists\, 1\in B \mid a\cdot 1=a$
- $a+b=b+a$, $a\cdot b=b\cdot a$
- $a+(b\cdot c)=(a+b)\cdot(a+c)$
- $a\cdot(b+c)=(a\cdot b)+(a\cdot c)$
- $a+a'=1$, $a\cdot a'=0$

TEOREMAS

- $a\cdot a=a$ - $a\cdot 1=a$
- $a+a=a$ - $a\cdot a'=0$
- $a\cdot 0=0$
- $a+1=1$

$+\rightarrow$ OR
$\cdot\rightarrow$ AND
$'\rightarrow$ NOT

- $a+a_1=1$, $a\cdot a_1=0$; $a+a_2=1$, $a\cdot a_2=0$ $\Big\}\ a_1=a_2$
- $a+(a\cdot b)=a$
- $a\cdot(a+b)=a$
- $(a+b)'=a'\cdot b'$
- $(a\cdot b)'=a'+b'$

$1\rightarrow$ TRUE
$0\rightarrow$ FALSE

MAPAS DE KARNAUGH

SOLO SE PUEDE USAR HASTA 4 VARIABLES

CON 2 VARIABLES

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

A\B	0	1
0	1	0
1	0	1

CON 3 VARIABLES

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
S	0	1	1	0	1	0	0	0

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

CON 4 VARIABLES

AB\CD	00	01	11	10
00				
01				
11				
10				

HACER LOS GLOBOS

SOLO DE 2, 4, 6, 8, 16 - □ ▭ TENER EN CUENTA QUE SE DOBLAN

NOMBRAR LOS GLOBOS Y DESPUES DE OBT. LO SUMAMOS $BC'D'+A'B'C'D+\dots$

AB\CD	00	01	11	10
00		1		
01	1			
11	1		1	1
10			1	1

$BC'D'$

SI LAS COORD. NO SE REPITEN NO LAS ESCRIBIMOS SI SE REPITEN LAS ESCRIBIMOS Y SI SON 0 LAS ESCRIBIMOS NEGADAS LAS COORDENADAS QUE COMPARTEN

SI ESTA SOLO VEMOS LAS COORDENADAS 0001

0 → NOT, 1 → TRUE

DON'T CARE

PONER X EN EL MAPA DONDE SABEMOS DONDE NUNCA VA A OCURRIR ESE CASO LA PUEDO ELEGIR COMO 1 Y 0, ENTONCES A VECES LA ELIJO COMO 1 PARA HACER GLOBOS Y SEGUIR SIMPLIFICANDO

ACLARACIÓN PARA HACER EL MAPA CON ALG.

SI TENEMOS $XYZ+\bar{X}Y+X\bar{Y}\bar{Z}$ LO HACEMOS ASÍ

	YZ	$Y\bar{Z}$	$\bar{Y}\bar{Z}$	$\bar{Y}Z$
X	1	1	0	0
$\bar{X}$	1	1	0	0

LO EN ESTE NO IMPORTA EL VALOR DE Z, ES COMO TENER $\bar{X}YZ+\bar{X}Y\bar{Z}$

FUNCIÓN OR EXCLUSIVO

(XOR) $X \oplus Y = XY' + X'Y$. ES 1 SOLO CUANDO EXACTAMENTE ALGUNO ES 1

NOR EXCLUSIVO (EQUIVALENCIA

$(X \oplus Y)' = X \cdot Y + X'Y'$. EL COMPLEMENTO DEL XOR. ES 1 SI AMBAS SON 1

GENERADOR DE PARIDAD

HACEMOS LA TABLA DE VERDAD. SI ES PAR (2 UNOS, 4 UNOS) PONEMOS 0 SI ES IMPAR PONEMOS 1. SUELE ESTAR RELACIONADO CON EL XOR

$F = A \oplus B \oplus C$ (IMPAR) $F = (A \oplus B \oplus C)'$ (PAR)

FORMAS CANONICAS

MINITERMINOS

- ESCRIBIMOS LA NEGACIÓN SI ES 0
- ESCRIBIMOS SIN NEGAR SI ES 1
- ES LA SUMA DE PRODUCTOS
- ESCRIBIMOS DONDE EN LA TABLA DE 1

MAXITERMINOS

- ESCRIBIMOS LA NEGACIÓN SI ES 1
- ESCRIBIMOS SIN NEGAR SI ES 0
- ES LA SUMA DE CADA COSA MULTIPLICADA
- ESCRIBIMOS DONDE LA TABLA ES 0

a	b	c	F	
0	0	0	0	m_0
0	0	1	0	m_1
0	1	0	0	m_2
0	1	1	1	m_3
1	0	0	0	m_4
1	0	1	1	m_5
1	1	0	1	m_6
1	1	1	1	m_7

$\bar{a}bc + a\bar{b}c + ab\bar{c} + abc$ → MINITERMINOS

$$\sum(m_3, m_5, m_6, m_7) = \sum_m(3,5,6,7)$$

$(a+b+c) \cdot (a+b+\bar{c}) \cdot (a+\bar{b}+c) \cdot (\bar{a}+b+c)$ → MAXITERMINOS

$$\prod(m_0, m_1, m_2, m_4) = \prod_m(0,1,2,4)$$

PARA ENCONTRAR LA TABLA DE VERDAD DE UNA FUNC. REEMPLAZAMOS EN CADA COMBINACIÓN (ALGEBRA A TABLA DE VERDAD)

CIRCUITO A TABLA DE VERDAD → REEMPLAZAMOS CADA COMBINACIÓN POR LO Q' DA EL CIRCUITO

CIRCUITO A ALGEBRA → VAMOS ESCRIBIENDO LO Q' VEMOS (HACER POR PARTES)

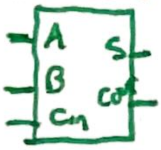
TABLA. V. A ALGEBRA → DONDE HAY 1 EN LA TABLA ESCRIBIMOS LA COMBINACIÓN

Escalas de Integración Media

ES COMO HACEMOS CON FUNCIONES CUANDO UNA PARTE DEL CÓDIGO SE REPITE

SUMADORES

HALF ADDER



LE PASAMOS LO QUE VAMOS A SUMAR (a+b+CARRY)

EL Cin ES LO QUE ACARREAMOS ANTERIORMENTE

DEVOLVEMOS LA SUMA Y EL CARRY.

$S = X \oplus Y$

$C = XY$

Decodificadores

PASAR UN NÚMERO BINARIO A UNA SOLA SALIDA (SE ACTIVA UNA SOLA COSA)

ES UN CIRCUITO COMBINACIONAL QUE CONVIERTE INFORMACIÓN BINARIA DE "N" ENTRADAS CODIFICADAS A $\boxed{2^N}$ SALIDAS ÚNICAS. (SÓLO UNA ESTÁ ACTIVA)

DECODIFICADOR ACTIVO POR BAJO → SALIDA ACTIVA = "0" (TODAS 1 MENOS UNA)

DECODIFICADOR ACTIVO POR ALTO → SALIDA ACTIVA = "1" (TODAS 0 MENOS UNA)

Multiplexores

MUCHAS ENTRADAS Y LA DIRIGE A $\boxed{1}$ SOLA SALIDA

ES UN CIRCUITO COMBINACIONAL QUE SELECCIONA INFORMACIÓN BINARIA DE MUCHAS ENTRADAS Y LA DIRIGE A UNA ÚNICA SALIDA

$Y = A\overline{S} + BS$

SI TIENE 2^m ENTRADAS NECESITA m SEÑALES DE SELECCIÓN

TABLA DE VERDAD DECODIFICADOR

A_3	A_2	A_1	A_0	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0
0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	0	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1	1	1	1	1
0	[illegible]	1	1	0	1	1	1	1	1	1	1
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

DIRECCIONAMIENTO Y LÓGICA DE DECODIFICACIÓN DE MEMORIAS

MAPA DE MEMORIA

CAPACIDAD TOTAL = $2^{n} \times 2^{m}$ bits

2^{n} → CANTIDAD DE WORDS

2^{m} → bits DE CADA WORDS

EJ → 2K WORDS × 8 bits (2K WORDS DE 8 bits CADA UNA)

AUMENTAR ANCHO DE LA WORD → CONEXIÓN EN PARALELO (//)

EJ → SI TENGO 2K WORDS × 8 bits Y NECESITO 2K WORDS × 16 bits ⇒ 2 //

AUMENTAR CAPACIDAD DE DIRECCIONAMIENTO → CONEXIÓN EN SERIE

EJ → SI TENGO α = 2K × 8 bits Y NECESITO 4K × 8 bits ⇒ 2 αS NECESITO

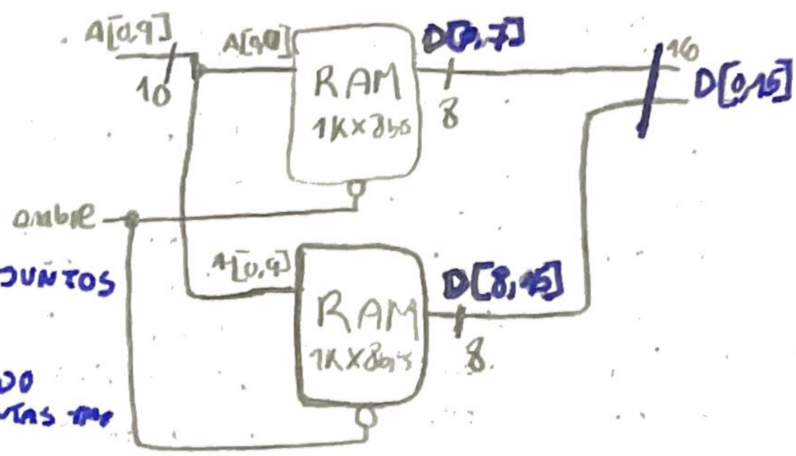
AUMENTAR ANCHO Y CAPACIDAD →

EJ → TENGO α = 2K × 8 bits Y NECESITO 4K × 16 bits ⇒ 2 α

CONSTRUIR SISTEMA DE MEMORIA RAM EN PARALELO

USANDO EL SIGUIENTE EJEMPLO: TENGO CHIPS DE 1K × 8 bits Y NECESITO DE 1K × 16 bits

NECESITO ESTO ENTONCES VEMOS QUE TENGO QUE USAR 2 CONEXIONES EN PARALELO

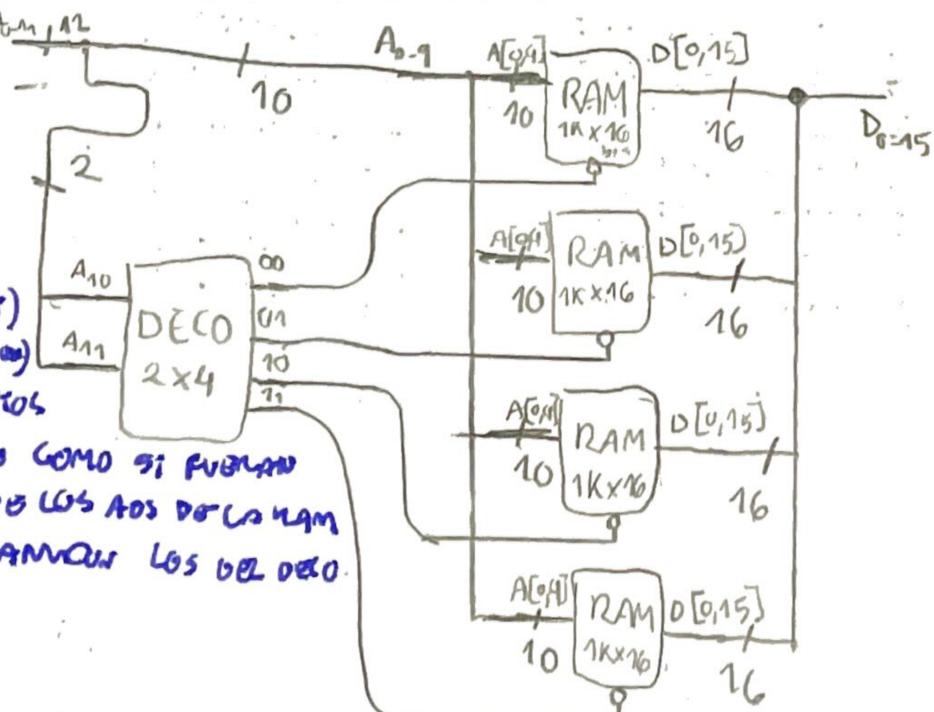


- PRESTAR ATENCIÓN A D (DATA)
- CONECTAR TODOS LOS ENABLES (CHIP SELECT) JUNTOS
- CONECTAR TODOS LOS ADDRES JUNTOS
- CONECTAR LOS DATA JUNTOS ESPECIFICANDO HASTA DONDE VA CADA UNO Y CUANDO LOS JUNTAS

CONSTRUIR SISTEMA DE MEMORIA RAM EN SERIE

USANDO EL SGTE EJEMPLO: TENGO CHIPS DE 1K × 16 Y NECESITO 4K × 16

NECESITO ESTO ENTONCES VEMOS QUE NECESITO USAR 4 CONEXIONES EN PARALELO



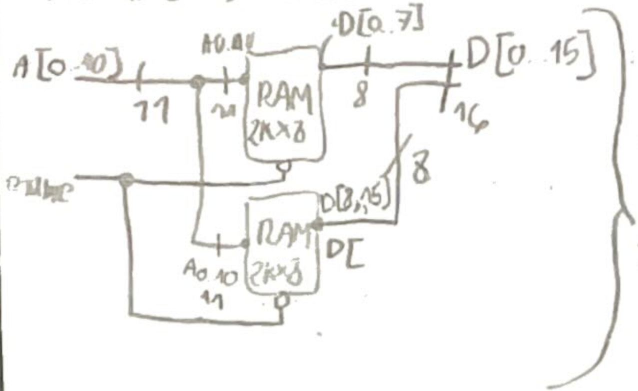
- CONECTAR TODOS LOS DA JUNTOS
- CONECTAR LOS ENABLE (LOS SELECT) AL DECODIFICADOR, (PONER LOS MAS ALTOS)
- CONECTAR TODOS LOS ADDRES JUNTOS
- CONECTAR LAS ENTRADAS DEL DECO COMO SI FUERAN ADDRES CON LOS NUM. SIGUIENTES DE LOS ADS DE LA RAM
- CONECTAR LAS ADDRES DE LA RAM CON LOS DEL DECO

CONSTRUIR SISTEMA DE MEMORIA RAM EN PARALELO + EN SERIE

POR EJEMPLO SI TENGO 2KX8BITS Y NECESITO 4KX16 ENTONCES VIMOS QUE NECESITAMOS 4

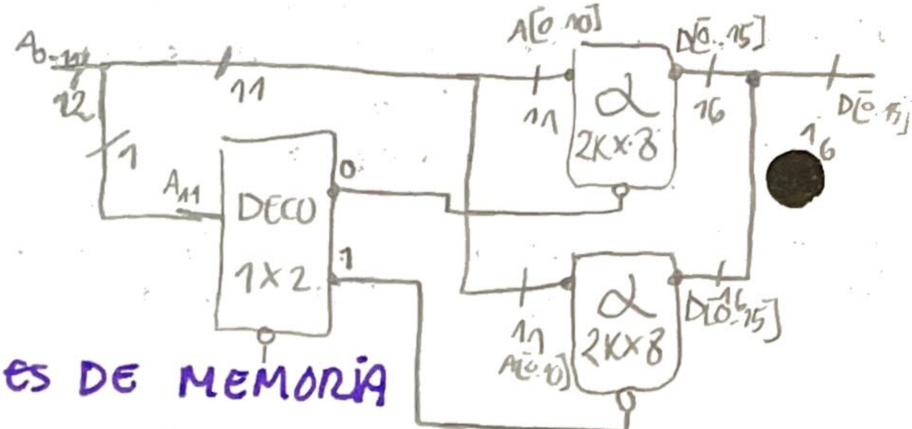
PRIMERO CREAMOS UN BLOQUE GENERAL AL QUE LLAMARÉ "α"

UTILIZAREMOS 2 CHIPS DE 2KX8BITS CONECTADOS EN PARALELO PARA OBTENER 2KX16 bits



BLOQUE α 2KX16BITS

· AHORA (YA CREADO MI BLOQUE α) VOY A HACER MI CONEXIÓN EN SERIE UTILIZANDO EL BLOQUE α DE 2K X 16 BITS Y COMO NECESITO 4KX16 VOY A USAR 2 BLOQUES α

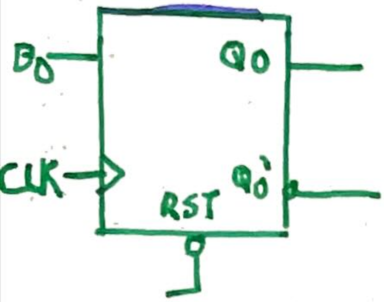


MAPA DE DIRECCIONES DE MEMORIA

00000 - 0FFFF	SEGMENTO 0
10000 - 1FFFF	SEGMENTO 1
20000 - 2FFFF	SEGMENTO 2
30000 - 3FFFF	SEGMENTO 3
40000 - 4FFFF	SEGMENTO 4
50000 - 5FFFF	SEGMENTO 5
60000 - 6FFFF	SEGMENTO 6
70000 - 7FFFF	SEGMENTO 7
80000 - 8FFFF	SEGMENTO 8
90000 - 9FFFF	SEGMENTO 9
A0000 - AFFFF	SEGMENTO A
B0000 - BFFFF	SEGMENTO B
C0000 - CFFFF	SEGMENTO C
D0000 - DFFFF	SEGMENTO D
E0000 - EFFFF	SEGMENTO E
F0000 - FFFFF	SEGMENTO F

CIRCUITOS SECUENCIALES

FLIP FLOP D

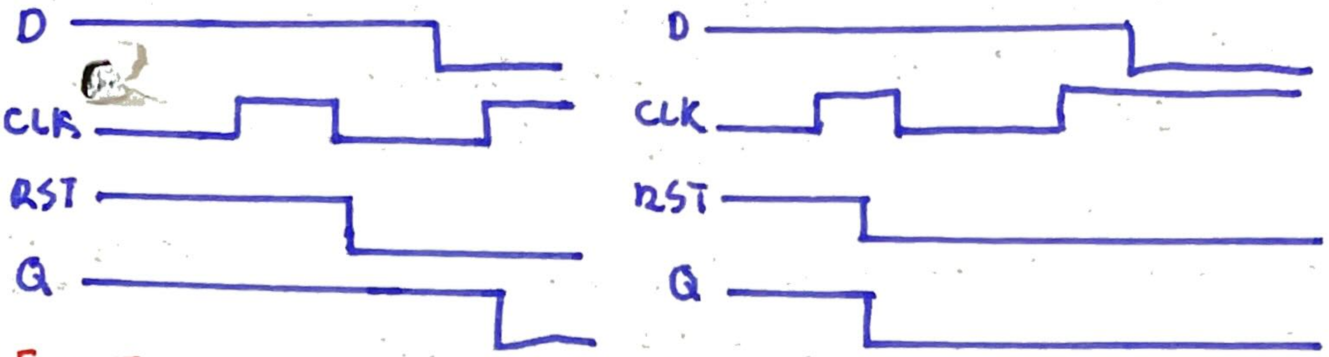


R	C	D	Q	Q'
0	x	x	0	1
1	↑	0	0	1
1	↑	1		

EL FLIP FLOP MIRA SU ENTRADAS Y PONE ALGO QUE AL FLIP FLOP SE LE ANTOJE, SOLAMENTE CUANDO OCURREN DETERMINADAS CONDICIONES EN UNA ENTRADA DE CONTROL O ENTRADA DE CLOCK

EL CLOCK ES UNA ENTRADA PARTICULAR DEL FLIP FLOP YA QUE ES AQUELLA QUE EL FF VA A TENER QUE MIRAR PARA DETERMINAR EN QUE MOMENTO DEL TIEMPO VA A MIRAR SUS ENTRADAS PARA PROCESARLA.

CON RESET SINCRONO — CON RESET ASINCRONO

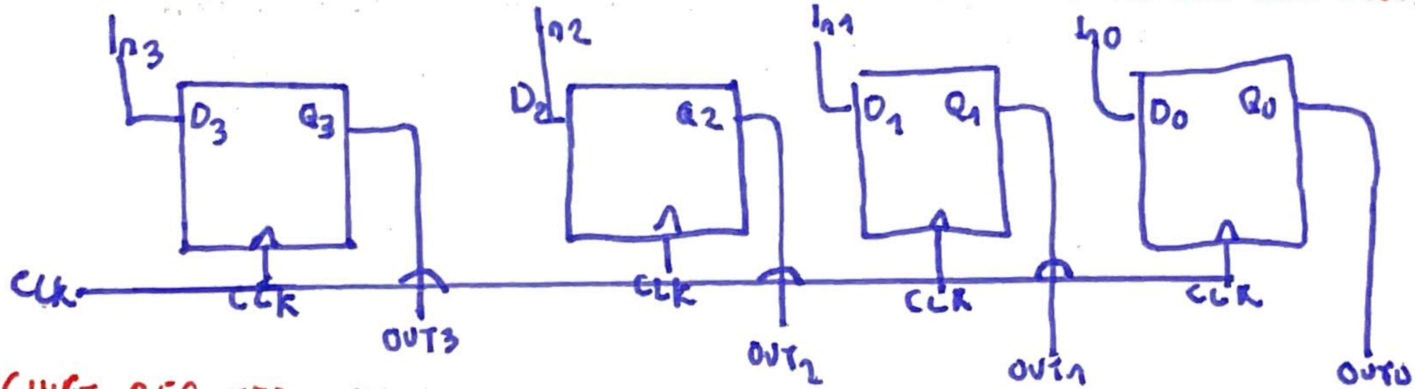


CADA FLIP FLOP PUEDE ALMACENAR SOLO 1 BITS.

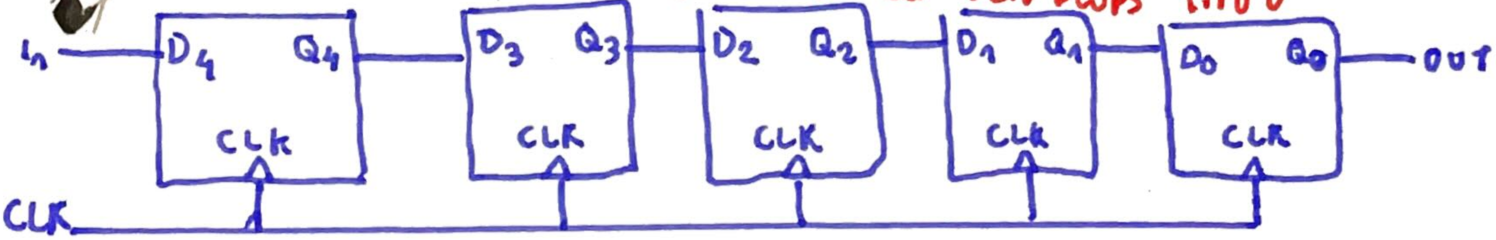
FLIP FLOP CON FLANCO CRECIENTE →

SOLO MIRA LOS INSTANTES DE TIEMPO ↑

REGISTRO DE ENTRADA Y SALIDA EN PARALELO DE 4 BITS CON FLIP-FLOPS TIPO D

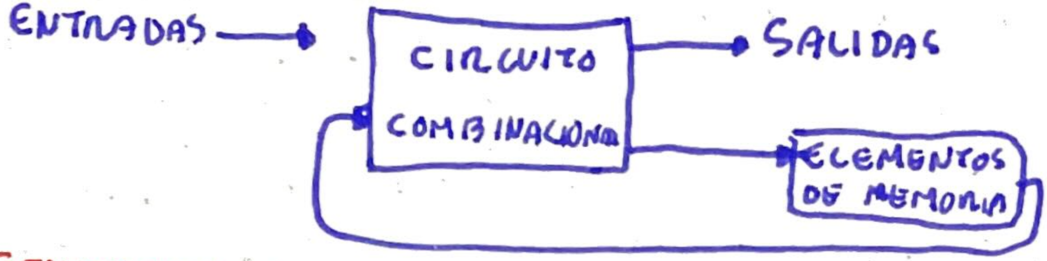


SHIFT REGISTER UNIDIRECCIONAL DE 5 BITS CON FLIP FLOPS TIPO D



RECORDAR QUE EN PARALELO DE LA SALIDA AL OUT MUCHOS OUT.
EN EL DE SERIE ES COMO UNA CADENA 1 IN 1 OUT.

CIRCUITOS SECUENCIALES CON RELOJ



- LA INFORMACIÓN ALMACENADA EN LOS ELEMENTOS DE MEMORIA EN CUALQUIER MOMENTO DADO DEFINE EL ESTADO DEL CIRCUITO SECUENCIAL EN ESE MOMENTO

ECUACIONES DE ESTADO

EL COMPORTAMIENTO DE LOS CIRCUITOS SECUENCIALES CON RELOJ SE DESCRIBEN ALGEBRAICAMENTE CON ECUACIONES DE ESTADO, QUE ESPECIFICAN EL SIGUIENTE ESTADO EN FUNCIÓN DEL ESTADO ACTUAL Y LAS ENTRADAS

$$A(t+1) = A(t)z(t) + B(t)z(t) \qquad A(t), B(t), .. = Q_k \dots Q_0$$

$$B(t+1) = A'(t)z(t) \qquad A(t+1), B(t+1) = D_k \dots D_0$$

TABLA DE ESTADOS

- ESTADO ACTUAL → ESTADOS DE LOS FLIP-FLOPS A Y B (EN CUALQUIER INSTANTE DADO)
- ENTRADA → VALOR DE z PARA CADA POSIBLE ESTADO ACTUAL
- SIGUIENTE ESTADO → ESTADOS DE LOS FLIP-FLOPS, UN CICLO DE RELOJ DESP $(t+1)$
- SALIDA → VALOR DE Y EN EL TIEMPO T PARA CADA ESTADO ACTUAL Y CONDICIÓN DE ENTRADA