



Departamento de
Telemática y Electrónica



Escuela Técnica Superior de Ingeniería
y Sistemas de Telecomunicación

Universidad Politécnica de Madrid

Electrónica 2

Programa de Actividades
del Bloque Temático 3

Bloque Temático 3. Sesión 1. Introducción a los circuitos secuenciales

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.P1	Introducción a los circuitos secuenciales. Circuitos combinacionales vs. secuenciales. Reloj, memoria de estado, circuitos secuenciales síncronos vs. asíncronos. Biestable. Circuito lógico. Tabla de transiciones. Análisis del funcionamiento.	Expositiva	25 min	S3_1 G1	No procede
BT3.P2	Flip-flop tipo D. Entradas asíncronas de reset y preset. Tabla de transiciones. Análisis del funcionamiento. Diagrama de estado de un flip-flop tipo D.	Expositiva	25 min	S3_1 G1	No procede
BT3.P3	Modelo de tiempos del flip-flop.	Expositiva	15 min	S3_1 G1	No procede
BT3.P4	Cronogramas ideales de circuitos con flip-flops y puertas lógicas.	Expositiva	35 min	S3_1 G1	No procede
BT3.AINP1	Realización en grupo de 3 ejercicios de análisis del funcionamiento de circuitos con flip-flops, completando cronogramas y realizando diagramas de estado.	Ejercicios Individual	60 min	NP PS3_2	Autocorrección
BT3.AINP2	Lectura sobre realización de entradas síncronas en flip-flops.	Lectura	40 min	NP PS3_2	No procede

Bloque Temático 3. Sesión 2. Cronogramas funcionales de circuitos con flip-flops

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.P5	Atención a dudas y consultas sobre realización de entradas síncronas en flip-flops.	Expositiva	20 min	S3_2 G2	No procede
BT3.ATGP1	Cronogramas funcionales de circuitos con flip-flops. Cronogramas con tiempos de circuitos con flip-flops. Obtención de diagramas de estado de circuitos con flip-flops. Diseño de flip-flops con entradas síncronas.	Aprendizaje colaborativo	80 min	S3_2 G2	Entregable
BT3.AINP3	Realización de la captura, simulación y configuración de la tarjeta de prototipado de dos ejercicios de circuitos combinacionales.	Ejercicios Individual	60 min	NP	Autocorrección
BT3.AINP4	Lectura sobre registros, arquitectura de circuitos secuenciales y estructura de los chips configurables.	Lectura	140 min	NP PS3_3	No procede

Bloque Temático 3. Sesión 3. Registros I

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.P6	Atención a dudas y consultas sobre registros, modelo general de la arquitectura de los circuitos secuenciales síncronos y modelo completo de un PLD sencillo (incluyendo lógica secuencial).	Expositiva	35 min	S3_3 G1	No procede
BT3.ATGP2	Realización de ejercicios sobre flip-flops: registros, identificación de los bloques básicos de un sistema secuencial (memoria, cálculo de las salidas y cálculo del estado futuro) y análisis de un circuito secuencial para obtener su diagrama de estados.	Aprendizaje colaborativo	65 min	S3_3 G1	Autocorrección

Bloque Temático 3. Sesión 4. Registros II

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.ATGP3	Realización de ejercicios sobre flip-flops: análisis de un circuito secuencial hasta la obtención del diagrama de estados, análisis de un circuito secuencial con retardos, diseño de un flip-flop con entradas síncronas, registro paralelo y registro de desplazamiento. En la parte final de la actividad se realizará un ejercicio entregable.	Aprendizaje colaborativo	100 min	S3_4 G2	Entregable
BT3.ATGNP1	Ejercicio guiado de captura, simulación y configuración de un circuito secuencial. Lectura sobre el diseño de pruebas de simulación de circuitos secuenciales. Realización de la captura, simulación y configuración de un circuito secuencial sencillo. Captura, simulación y configuración en la tarjeta de los ejercicios de diseño de flip-flops con entradas síncronas en clase.	Aprendizaje Cooperativo	180 min	NP	Autoevaluación

Bloque Temático 3. Sesión 5. Diseño de Autómatas

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.P7	Metodología de diseño de autómatas.	Expositiva	30 min	S3_6 G1	No procede
BT3.P8	Contador binario ascendente sin entradas síncronas: descripción funcional y cronograma. Diagrama de estados de un contador de 2 ó 3 bits. Diseño del autómata. Contador binario ascendente con habilitación de reloj: descripción funcional y cronograma. Diagrama de estados de un contador con habilitación de reloj de 2 ó 3 bits.	Expositiva	20 min	S3_6 G1	No procede
BT3.ATGP4	Ejercicios de diseño de autómatas de Moore y de Mealy a partir del diagrama de estados (3 ó 4 estados y una entrada síncrona). Realización del diagrama de estados de un contador descendente y de un contador decimal.	Aprendizaje Cooperativo	50 min	S3_7 G1	Autocorrección
BT3.AINP5	Ejercicios de aplicación de la metodología de diseño de autómatas de Moore para la implementación de contadores. Representación en un cronograma del funcionamiento de ambos contadores.	Ejercicios Individual	40 min	NP PS3_8	Autocorrección
BT3.AINP6	Lectura sobre funcionamiento y diseño de contadores. Ejercicios. 1 ^a Parte	Lectura	240 min	NP PS3_7	No procede

Bloque Temático 3. Sesión 6. Contadores I

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.P9A	Atención a dudas y preguntas sobre la 1 ^a parte de la lectura de contadores	Expositiva	30 min	S3_7 G1	No procede
BT3.ATGP5A	Ejercicios sobre contadores: diseño de contadores con estructuras regulares, análisis de cronogramas, contadores de módulo arbitrario y ampliación de contadores BCD.	Aprendizaje Cooperativo	70 min	S3_8 G1	Autocorrección
BT3.AINP6	Lectura sobre funcionamiento y diseño de contadores. Ejercicios. 2 ^a Parte	Lectura	200 min	NP PS3_7	No procede

Bloque Temático 3. Sesión 7. Contadores II

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.P9B	Atención a dudas y preguntas sobre la 2 ^a parte de la lectura de contadores	Expositiva	30 min	S3_7 G1	No procede
BT3.ATGP5B	Ejercicios sobre contadores: diseño de contadores con estructuras regulares, análisis de cronogramas, contadores de módulo arbitrario y ampliación de contadores BCD	Aprendizaje Cooperativo	70 min	S3_8 G1	Entregable
BT3.ATGNP2	Diseño y realización en la placa de prototipado de un divisor de frecuencia y de un contador de módulo programable (que utilice como clock enable la salida del divisor).	Aprendizaje Cooperativo	120 min	NP	Autocorrección
BT3.ATGNP3	Ejercicios de diseño de ampliación de contadores y de análisis del funcionamiento de contadores en cronogramas.	Aprendizaje Cooperativo	80 min	NP	Autocorrección

Bloque Temático 3. Sesión 8. secuenciales Ejercicios sobre flip-flops

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.ATGP6	Realización de ejercicios sobre flip-flops: 2 ejercicios con cronogramas de tiempos, 2 ejercicios de análisis completando cronogramas, 2 ejercicios de análisis extrayendo diagramas de estado, 4 ejercicios de diseño de entradas síncronas, 2 ejercicios con registros.	Aprendizaje Cooperativo	100 min	S3_5 G2	Entregable
BT3.ATGNP4	Preparación del examen en grupo. Presentación al grupo, para su resolución, de cualquier duda que haya surgido en el desarrollo de este tercer bloque temático.	Aprendizaje Cooperativo	60 min	S3_10 G1	No procede
BT3.AINP7	Preparación individual del examen: repaso del trabajo realizado y realización de ejercicios alternativos si es necesario.	Ejercicios Individual	60 min	NP	No procede

Bloque Temático 3. Sesión 9. Metodología completa de diseño de sistemas

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.P10	Resumen de la metodología completa de diseño de sistemas secuenciales a partir de una especificación funcional. Descripción de algunos ejemplos de desarrollo de diagramas de estado: Control simplificado de la puerta de un garaje, conformador de pulsos y detector de secuencias	Expositiva	50 min	S3_9 G1	No procede
BT3.ATGP7	Ejercicios de diseño de diagramas de estados.	Aprendizaje Cooperativo	50 min	S3_9 G1	Autocorrección
BT3.AINP8	Ejercicios sobre circuitos secuenciales: 4 ejercicios de diseño de flip-flops con entradas síncronas, 2 ejercicios de análisis completando cronogramas, 2 ejercicios de análisis extrayendo diagramas de estado, 2 ejercicios de análisis de funcionamiento de registros, 2 ejercicios de análisis de la arquitectura de circuitos secuenciales.	Ejercicios Individual	120 min	NP PS3_6	Autocorrección

Bloque Temático 3. Sesión 10. Repaso sobre contadores y diseño de autómatas

Código	Título	Modalidad	Duración	Sesión	Evaluación
BT3.ATGP8	Ejercicios de repaso sobre contadores y diseño de autómatas.	Aprendizaje Cooperativo	100 min	S3_10 G2	Autocorrección
BT3.AINP9	Ejercicios sencillos de diseño de diagramas de estado	Ejercicios Individual	40 min	NP	Autocorrección

La notación empleada es la siguiente:

BTn.Pm	Presentación del profesor nº m del Bloque Temático n
BTn.AINPm	Actividad Individual No Presencial nº m del Bloque Temático n
BTn.AIPm	Actividad Individual Presencial nº m del Bloque Temático n
BTn.ALPM	Actividad de Laboratorio Presencial nº m del Bloque Temático n
BTn.ATGNPm	Actividad de Trabajo en Grupo No Presencial nº m del Bloque Temático n
BTn.ATGPm	Actividad de Trabajo en Grupo Presencial nº m del Bloque Temático n
Sx_y	Sesión presencial y del bloque x
PSx_y	Actividad a realizar de manera previa a la sesión presencial y del bloque x
NP	Modalidad no presencial
G1	Grupo presencial completo
G2	Grupo presencial desdoblado
G3	Grupo presencial de laboratorio

Esta página se ha dejado intencionadamente en blanco



ETSiST-UPM



Electrónica 2

Bloque Temático 3

Introducción a los Sistemas Secuenciales.
Memoria de Estado. Sistemas
Secuenciales Síncronos y Asíncronos.
Señal de Tiempo. Biestables

Sistemas Secuenciales

LAS FUNCIONES CUYAS SALIDAS, EN UN INSTANTE DE TIEMPO, DEPENDE DEL VALOR DE LAS ENTRADAS EN OTROS INSTANTES DE TIEMPO, TIENEN QUE SER REALIZADAS POR SISTEMAS CON MEMORIA.

LA SUMA ACUMULADA DE NÚMEROS QUE SE SUCEDEN EN EL TIEMPO.

EL ALMACENAMIENTO DEL VALOR DE UN DATO.

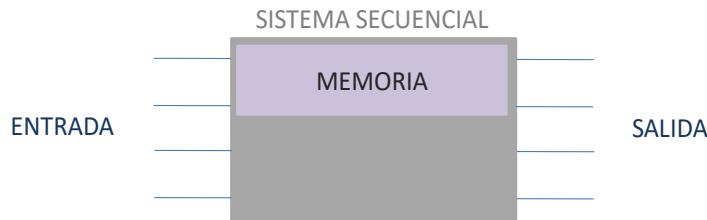
LOS SISTEMAS DIGITALES CON MEMORIA SE DENOMINAN SISTEMAS SECUENCIALES.



BT3_S1 BT3_P1_2

Sistemas Secuenciales

Los sistemas secuenciales tienen memoria



El valor de salida de un sistema secuencial depende del valor de la entrada y del valor que guarda la memoria del sistema

$$\text{SALIDA} = F(\text{ENTRADA}, \text{MEMORIA})$$

BT3_S1 BT3_P1_3

Sistemas Secuenciales



LA COMBINACIÓN DE SALIDA ASOCIADA A UNA DETERMINADA COMBINACIÓN DE ENTRADA ES SIEMPRE LA MISMA.

LA SALIDA PARA UNA MISMA COMBINACIÓN DE ENTRADA PUEDE SER DISTINTA EN DIFERENTES INSTANTES DE TIEMPO.

ENT	A	X	B	X	A	X	B
SAL	W	X	Z	X	W	X	Z

ENT	A	X	B	X	A	X	D
SAL	T	X	V	X	Y	X	V

BT3_S1 BT3_P1_4

Sistemas Secuenciales

SISTEMA SECUENCIAL

ENTRADA **SISTEMA SECUENCIAL** **SALIDA**

SALIDA = F (ENTRADA, MEMORIA)

ENTRADA	A	X	A	X	B	X	A
MEMORIA	T	X	V	X	Y	X	T
SALIDA	W	X	Z	X	Q	X	W

LA SALIDA DE UN SISTEMA SECUENCIAL ES SIEMPRE LA MISMA PARA UNA DETERMINADA COMBINACIÓN DE ENTRADA Y UN DETERMINADO VALOR ALMACENADO EN MEMORIA

LA MEMORIA DE UN SISTEMA SECUENCIAL PUEDE ALMACENAR UN CONJUNTO FINITO DE VALORES DIFERENTES. ESTE CONJUNTO DE VALORES SON LO QUE SE CONOCE COMO **ESTADOS** DEL SISTEMA.

EL **ESTADO** DE MEMORIA DEL SISTEMA, EN EL INSTANTE DE TIEMPO T, DEPENDE DE LOS VALORES DE LAS ENTRADAS EN INSTANTES ANTERIORES A T.

LA SALIDA DE UN SISTEMA SECUENCIAL ES FUNCIÓN DE LA ENTRADA Y DEL **ESTADO** DEL SISTEMA.

BT3_S1 BT3_P1_5

Sistemas Secuenciales

SISTEMA SECUENCIAL

MEMORIA

LA CANTIDAD (CAPACIDAD) DE MEMORIA DE UN SISTEMA SECUENCIAL SE MIDE EN BITS. EN SISTEMAS CON GRAN CAPACIDAD DE MEMORIA SE UTILIZAN FACTORES DE ESCALA:

- 2^{10} (1024) KILO
- 2^{20} (1.048.576) MEGA
- 2^{30} (1.073.741.824) GIGA
- 2^{40} (1.099.511.627.776) TERA

EN UN SISTEMA CON N BITS DE MEMORIA PUEDE HABER HASTA 2^N ESTADOS DISTINTOS

UN SISTEMA SECUENCIAL CON 4 BITS DE MEMORIA PUEDE TENER HASTA 16 ESTADOS

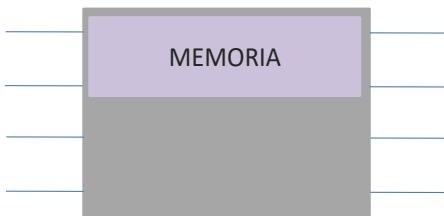
UN SISTEMA QUE NECESITE DISPONER DE M ESTADOS DE MEMORIA DEBE TENER UNA CAPACIDAD DE MEMORIA, N, TAL QUE $M \leq 2^N$

UN SISTEMA SECUENCIAL CON 12 ESTADOS DEBE TENER, AL MENOS, 4 BITS DE MEMORIA

BT3_S1 BT3_P1_6

Sistemas Secuenciales

SISTEMA SECUENCIAL



EL ESTADO DE LA MEMORIA DE UN SISTEMA SECUENCIAL EN UN INSTANTE DE TIEMPO, T, CODIFICA UN VALOR DERIVADO DE LA SUCESIÓN DE COMBINACIONES EN LA ENTRADA DEL SISTEMA EN INSTANTES DE TIEMPO ANTERIORES A T.

EN UN SISTEMA SECUENCIAL, CUYA ENTRADA ES UN NÚMERO DE 4 BITS, Y QUE DETECTA, ACTIVANDO SU SALIDA, QUE LA SUMA ACUMULADA POR LOS NÚMEROS QUE SE HAN SUCEDIDO EN SU ENTRADA ES MAYOR QUE 100, EL ESTADO DE MEMORIA EN EL INSTANTE T REPRESENTA LA SUMA ACUMULADA HASTA EL MOMENTO.

ENTRADA	6	X	5	X	3	X	2	X	X	4	X	5	X	2
ESTADO	0	X	6	X	11	X	14	X	X	93	X	97	X	0
SALIDA

EL ESTADO DEL SISTEMA CODIFICA UN NÚMERO EN BINARIO NATURAL ENTRE CERO Y CIEN, LUEGO DEBE DE DISPONER DE 7 BITS DE MEMORIA.

BT3_S1 BT3_P1_7

Sistemas Secuenciales

EN UN SISTEMA SECUENCIAL, CUYA ENTRADA ES UN NÚMERO DE 4 BITS, Y QUE INDICA, SI EL NÚMERO ACTUAL ES MAYOR (SALIDA 01), MENOR (SALIDA 10) O IGUAL (SALIDA 11) QUE EL ANTERIOR, EL ESTADO DE MEMORIA EN EL INSTANTE T DEBE REPRESENTAR EL NÚMERO PRESENTE EN LA ENTRADA EN EL INSTANTE T-1.

ENTRADA	6	X	5	X	5	X	7	X	14	X	7	X	7	
ESTADO	0	X	6	X	5	X	5	X	7	X	14	X	7	
SALIDA	01 (>)	X	10 (<)	X	11 (=)	X	01 (>)	X	01 (>)	X	10 (<)	X	11 (=)	

EL ESTADO DEL SISTEMA CODIFICA UN NÚMERO EN BINARIO NATURAL DE CUATRO BITS, LUEGO DEBE DE DISPONER DE 4 BITS DE MEMORIA.

EN UN SISTEMA SECUENCIAL, CUYA ENTRADA ES UN NÚMERO DE 4 BITS EN COMPLEMENTO A DOS, Y QUE INDICA EN SU SALIDA, CON UN UNO, SI EL SIGNO DE LOS DOS ÚLTIMOS NÚMEROS ES IGUAL, EL ESTADO DE MEMORIA EN EL INSTANTE T DEBE CODIFICAR EL SIGNO DEL NÚMERO PRESENTE EN LA ENTRADA EN EL INSTANTE T-1.

ENTRADA	-3	X	+5	X	+6	X	-7	X	-3	X	+2	X	-4	
ESTADO	0	X	1	X	0	X	0	X	1	X	1	X	0	
SALIDA

EL ESTADO DEL SISTEMA SÓLO REQUIERE DE 1 BIT DE MEMORIA PARA ALMACENAR EL SIGNO DEL PENÚLTIMO NÚMERO.

BT3_S1 BT3_P1_8

Sistemas Secuenciales

SISTEMA SECUENCIAL

EL FUNCIONAMIENTO DE LOS SISTEMA SECUENCIALES IMPlica LA EXISTENCIA DE UNA O VARIAS SEÑALES QUE DEFINEN EL TRANSCURSO DEL TIEMPO: ESTAS SEÑALES PERMITEN DISTINGUIR EL PASO DEL INSTANTE DE TIEMPO T AL INSTANTE DE TIEMPO $T+1$.

HAY DOS FORMAS DE DEFINIR EL TRANSCURSO DEL TIEMPO EN UN SISTEMA SECUENCIAL:

IMPLÍCITAMENTE: CUANDO EL SISTEMA TOMA COMO REFERENCIA PARA EL TRANSCURSO DEL INSTANTE T AL $T+1$ LA OCURRENCIA DE CAMBIOS EN LOS BITS DE ENTRADA DEL SISTEMA.

EXPLÍCITAMENTE: CUANDO EXISTEN SEÑALES CUYA FUNCIÓN ESPECÍFICA ES DEFINIR LA SUCESIÓN DE INSTANTES DE TIEMPO MEDIANTE UNA CONMUTACIÓN DE SU NIVEL LÓGICO.

BT3_S1 BT3_P1_9

Sistemas Secuenciales

SISTEMA SECUENCIAL ASÍNCRONO

LOS SISTEMAS EN QUE LA SUCESIÓN DE INSTANTES DE TIEMPO SE DEFINE POR CAMBIOS EN LA COMBINACIÓN DE ENTRADA SE CONOCEN COMO **SISTEMAS SECUENCIALES ASÍNCRONOS**

ENTRADA	P	X	Q	X	R	X	S	X	V
TIEMPO	T	X	T+1	X	T+2	X	T+3	X	T+4

SISTEMA SECUENCIAL SÍNCRONO

LOS SISTEMAS EN QUE LA SUCESIÓN DE INSTANTES DE TIEMPO VIENE DEFINIDA POR UNA SEÑAL DE TIEMPO (**SEÑAL DE RELOJ**), SE DENOMINAN **SISTEMAS SECUENCIALES SÍNCRONOS**

ENTRADA	P	X	Q	X	R	X	S	X	V
RELOJ	[Pulse]								
TIEMPO	T	X	T+1	X	T+2	X	T+3	X	T+4

BT3_S1 BT3_P1_10

Sistemas Secuenciales

SISTEMA SECUENCIAL SÍNCRONO

LA CARACTERÍSTICA MÁS IMPORTANTE DE UNA SEÑAL DE RELOJ ES LA FRECUENCIA.

CLK

T_{CLK}

$F_{CLK} = 1/T_{CLK}$

CADA PERÍODO DE LA SEÑAL DE RELOJ DEFINE UN NUEVO INTERVALO DE TIEMPO EN UN CIRCUITO SECUENCIAL SÍNCRONO.

CLK

T $T+1$ $T+2$ $T+3$

BT3_S1 BT3_P1_11

Sistemas Secuenciales

PARA REALIZAR UN SISTEMA SECUENCIAL ES NECESARIO UTILIZAR CÉLULAS CAPACES DE REALIZAR FUNCIONES LÓGICAS (COMO LAS QUE SE USAN EN LA REALIZACIÓN DE CIRCUITOS COMBINACIONALES) Y CÉLULAS CON MEMORIA.

SISTEMA COMBINACIONAL

REALIZACIÓN CON CÉLULAS QUE MATERIALIZAN FUNCIONES LÓGICAS

CON LAS CÉLULAS SE CONSTRUYEN LOS CIRCUITOS LÓGICOS QUE CALCULAN LAS COMBINACIONES DE SALIDA.

SISTEMA SECUENCIAL

REALIZACIÓN CON CÉLULAS QUE MATERIALIZAN FUNCIONES LÓGICAS Y CON CÉLULAS CON MEMORIA

CON LAS CÉLULAS CON MEMORIA SE MATERIALIZA LA MEMORIA DE ESTADO. CON LAS CÉLULAS QUE REALIZAN FUNCIONES LÓGICAS SE CONSTRUYEN CIRCUITOS LÓGICOS QUE CONTROLAN LOS CAMBIOS DE ESTADO DE LA MEMORIA Y CIRCUITOS LÓGICOS QUE CALCULAN EL VALOR DE SALIDA.

BT3_S1 BT3_P1_12

BIESTABLES

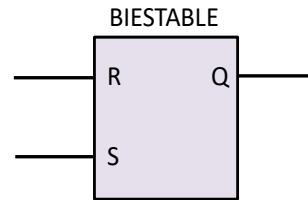
EN LOS SISTEMAS SECUENCIALES SE PUEDEN TRES TIPOS DE CÉLULAS CON MEMORIA: **LATCHES**, **FLIP-FLOPS** Y **MEMORIAS DIRECCIONABLES**.

LA MEMORIA DE ESTAS CÉLULAS SE MATERIALIZA MEDIANTE EL USO DE ESTRUCTURAS **BIESTABLES**.

UN **BIESTABLE** ES UN CIRCUITO CON UNA CAPACIDAD DE MEMORIA DE 1 BIT Y QUE DISPONE DE DOS ENTRADAS, **R** Y **S**, QUE PERMITEN CONTROLAR LAS OPERACIONES DE ALMACENAMIENTO Y MEMORIZACIÓN.

EN UN BIESTABLE EL TRANSCURSO DEL TIEMPO LO DETERMINAN LOS CAMBIOS EN LAS ENTRADAS DE **RESET** Y **SET**.

ES UNA CÉLULA ASÍNCRONA.



BT3_S1 BT3_P1_13

BIESTABLES

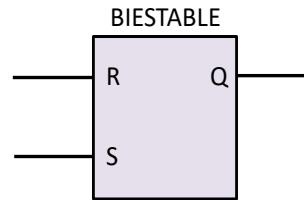
CUANDO SE QUIERE ALMACENAR UN DATO EN UN BIESTABLE, SUS ENTRADAS DEBEN TOMAR VALORES COMPLEMENTARIOS:

SI SE ACTIVA LA ENTRADA R (**RESET**), SE MEMORIZA (EN **Q**) UN CERO.

SI SE ACTIVA LA ENTRADA S (**SET**), SE MEMORIZA (EN **Q**) UN UNO.

MIENTRAS SE DESEE MANTENER MEMORIZADO UN DATO (EN **Q**), LAS DOS ENTRADAS DEBEN PERMANECER INACTIVAS.

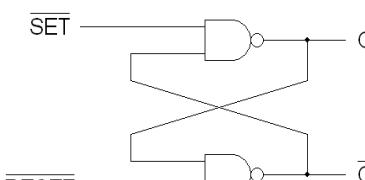
RESET	SET	ESTADO
ACTIVO	INACTIVO	0
INACTIVO	ACTIVO	1
INACTIVO	INACTIVO	MEMORIA

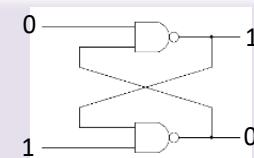
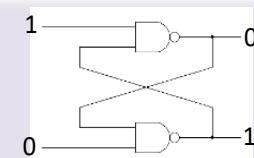
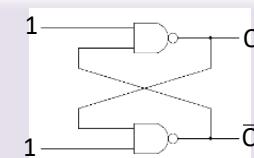


BT3_S1 BT3_P1_14

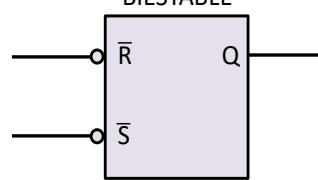
BIESTABLES

UN BIESTABLE PUEDE REALIZARSE CON UN CIRCUITO COMBINACIONAL REALIMENTADO.



$\overline{\text{RESET}}$	$\overline{\text{SET}}$	ESTADO
0	1	0
1	0	1
1	1	MEMORIA

BIESTABLE


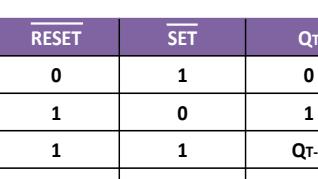
BT3_S1 BT3_P1_15

BIESTABLES

EL FUNCIONAMIENTO DE UN BIESTABLE PUEDE REPRESENTARSE EN UNA TABLA DE TRANSICIONES, QUE INDICA EL VALOR QUE TOMA EL ESTADO DE SALIDA Q EN EL INSTANTE DE TIEMPO T (Q_T), EN FUNCIÓN DEL VALOR DE LAS ENTRADAS EN ESE MISMO INSTANTE (T) Y DEL ESTADO EN $T-1$ (Q_{T-1}).

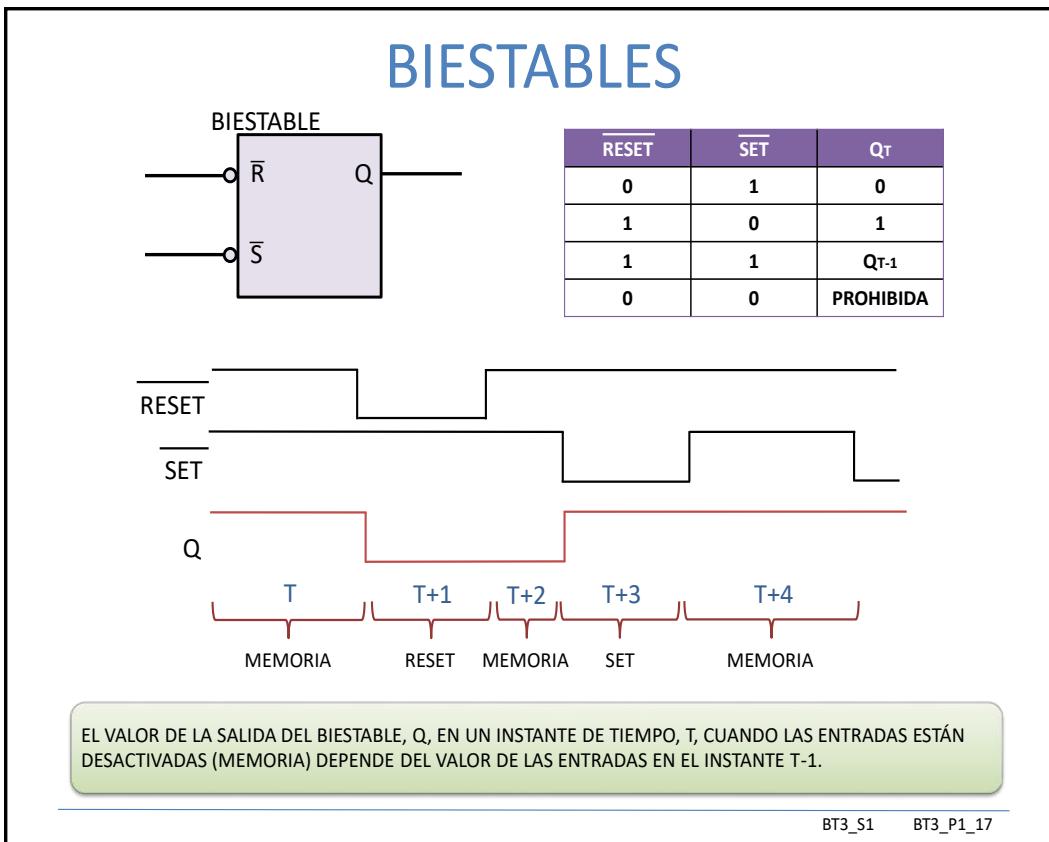
$\overline{\text{RESET}}$	$\overline{\text{SET}}$	Q_{T-1}	Q_T
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1
0	0	0	?
0	0	0	?

$\overline{\text{RESET}}$	$\overline{\text{SET}}$	Q_T
0	1	0
1	0	1
1	1	Q_{T-1}
0	0	PROHIBIDA

BIESTABLE


LA ACTIVACIÓN SIMULTÁNEA DE LAS DOS ENTRADAS ESTÁ PROHIBIDA: PUEDE PROVOCAR OSCILACIONES EN EL ESTADO DEL BIESTABLE SI EN EL INSTANTE ANTERIOR AMBAS ENTRADAS ESTABAN INACTIVAS (MEMORIA).

BT3_S1 BT3_P1_16



Esta página se ha dejado intencionadamente en blanco



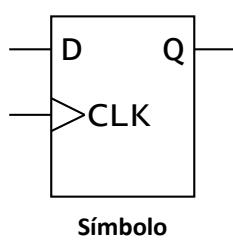
Electrónica 2

Bloque Temático 3

Flip-flops D. Entradas asíncronas reset y preset. Tabla de transiciones. Análisis del funcionamiento.

BT3_S1 BT3_P2_1

Flip-flop D

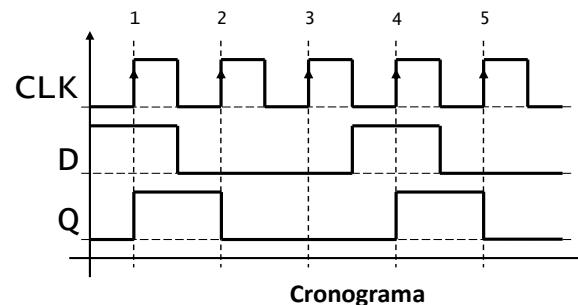


- Un flip-flop memoriza en su **salida Q** el valor presente en su **entrada síncrona D** cuando ocurre un **flanco activo** de la **señal de reloj CLK**:

$$Q(T+1) = D(T)$$

Entradas		Salida
D(T)	CLK	Q(T+1)
0	↑	0
1	↑	1

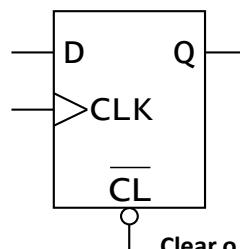
Tabla de transiciones



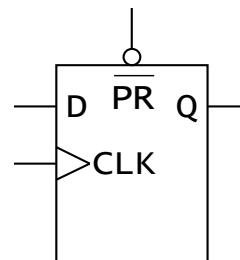
BT3_S1 BT3_P2_2

Flip-flop D con entradas asíncronas

- Las entradas asíncronas se utilizan para definir el estado de memoria **inicial** del flip-flop.
- Las entradas asíncronas:
 - Son prioritarias frente a las síncronas.
 - Suelen ser activas a nivel bajo.



Clear o Reset asíncrono

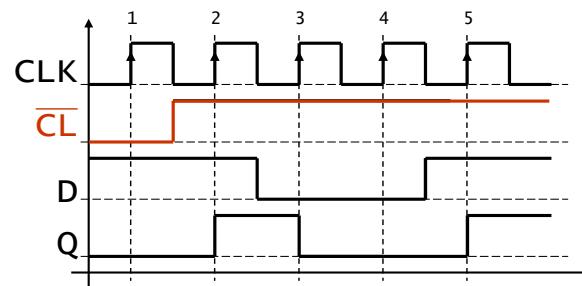
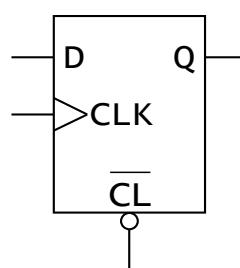


Preset asíncrono

BT3_S1 BT3_P2_3

Flip-flop D con entradas asíncronas

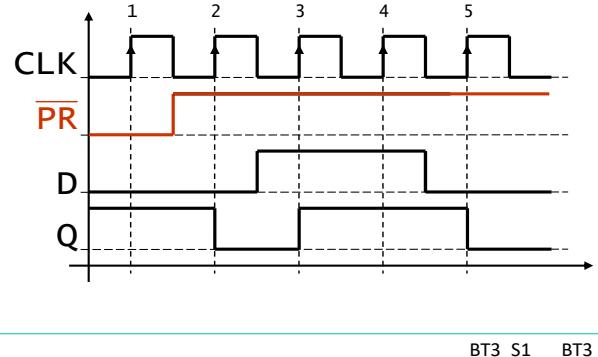
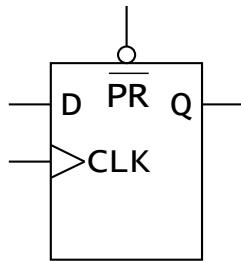
- La activación de la entrada asíncrona de **Reset** fuerza la salida del flip-flop (el estado de memoria) a **0**.
- Es activa por nivel.



BT3_S1 BT3_P2_4

Flip-flop D con entradas asíncronas

- La activación de la entrada asíncrona de **Preset** fuerza la salida del flip-flop (el estado de memoria) a **1**.
- Es activa por nivel.



Esta página se ha dejado intencionadamente en blanco



Electrónica II

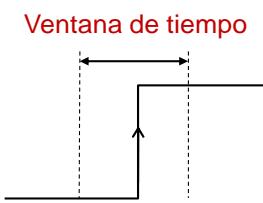
Bloque temático 3

Modelo de tiempos del flip-flop tipo D.

BT3_S1 BT3_P3_1

Modelo de tiempos del Flip-flop D

- Para que un flip-flop funcione correctamente, el nivel lógico en su entrada síncrona, D, debe ser estable en una *ventana de tiempo* definida en torno a la ocurrencia del flanco activo de reloj.

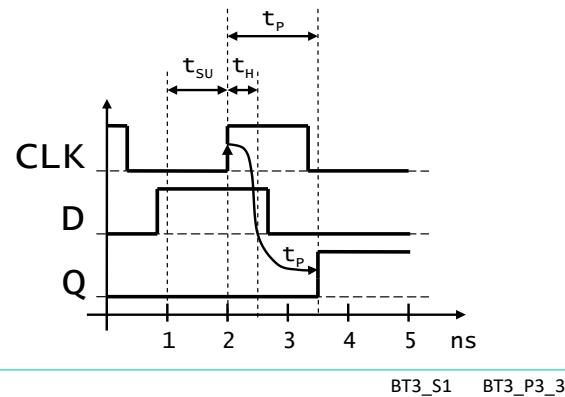
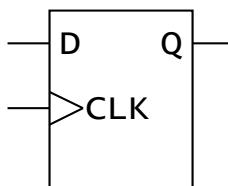


- La ventana de tiempo de estabilidad del dato de entrada se define mediante dos tiempos relativos a la ocurrencia del flanco activo: los tiempos de *set-up* y *hold*.

BT3_S1 BT3_P3_2

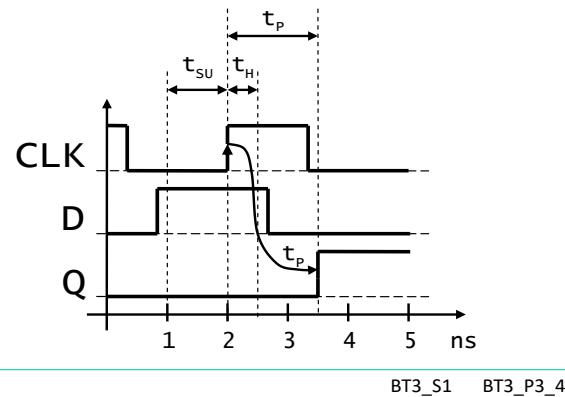
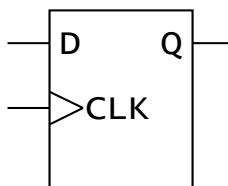
Modelo de tiempos del Flip-flop D

- Tiempo de SET-UP (tiempo de preparación) t_{SU} : Es el tiempo mínimo que el nivel lógico en la entrada síncrona D debe haber permanecido estable antes de la ocurrencia del flanco de reloj.



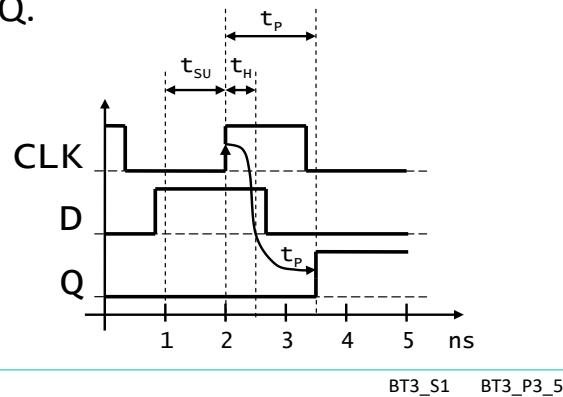
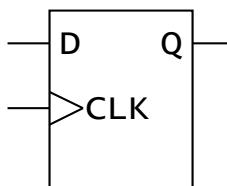
Modelo de tiempos del Flip-flop D

- Tiempo de HOLD (tiempo de persistencia) t_H : Es el tiempo mínimo que el dato debe mantenerse estable después de la ocurrencia del flanco activo de reloj.



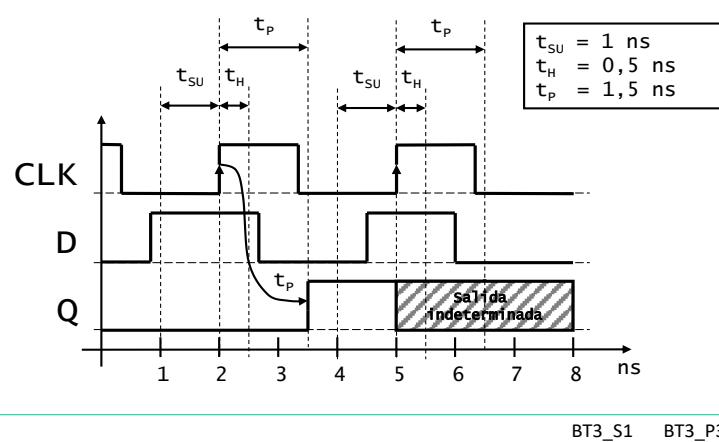
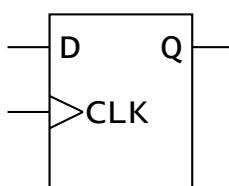
Modelo de tiempos del Flip-flop D

- El tiempo de propagación de un flip-flop relaciona la entrada de reloj y la salida del flip-flop.
- Tiempo de Propagación, t_p : Es el tiempo que transcurre entre la ocurrencia del flanco activo de reloj y la conmutación de la salida Q.



Modelo de tiempos del Flip-flop D

- Si el dato en la entrada D del flip-flop no es estable en el intervalo definido por t_{su} y t_h , el valor que toma la salida y el tiempo de propagación son impredecibles.



Esta página se ha dejado intencionadamente en blanco



ETSIST-UPM

Electrónica II

Bloque temático 3

Análisis de circuitos síncronos con flip-flops y puertas lógicas.

BT3_S1 BT3_P4_1

Estructura de los circuitos secuenciales síncronos

- Realizados con flip-flops tipo D activos con el mismo flanco de reloj.
- Disponen de una única señal de reloj conectada directamente a la entrada de reloj de todos los flip-flops.
- Cuentan con una señal de inicialización asíncrona conectada a la entrada asíncrona de Reset o Preset de todos los flip-flops del circuito.

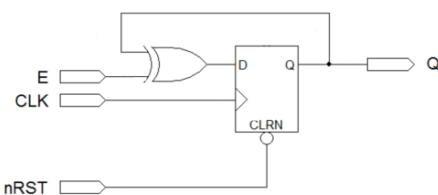
BT3_S1 BT3_P4_2

Metodología general de análisis

1. Obtener las expresiones de las **funciones lógicas** correspondientes a las **entradas síncronas** (estado de memoria futuro) de los flip-flops.
2. Obtener las expresiones de las **funciones lógicas** correspondientes a **las salidas**.
3. Representar el funcionamiento del circuito mediante **tablas de transiciones, cronogramas funcionales y diagramas de estado**.

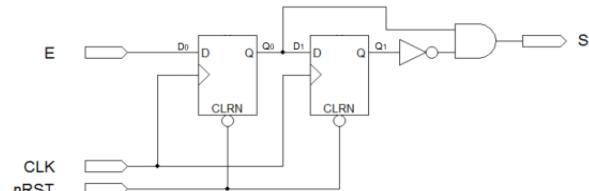
BT3_S1 BT3_P4_3

Obtención de las ecuaciones



$$D(T) = Q(T) \oplus E(T)$$

La Salida es directamente
el estado de memoria, $Q(T)$



$$D_0(T) = E(T)$$

$$D_1(T) = Q_0(T)$$

$$S(T) = Q_0(T) \cdot \overline{Q_1(T)}$$

BT3_S1 BT3_P4_4

Obtención de la Tabla de Transiciones

- ✓ La tabla de transiciones especifica el estado de memoria futuro (Q en $T+1$) en función del valor de las entradas y del estado de memoria actual (E y Q en T)

$$Q(T + 1) = F(Q(T), E(T))$$

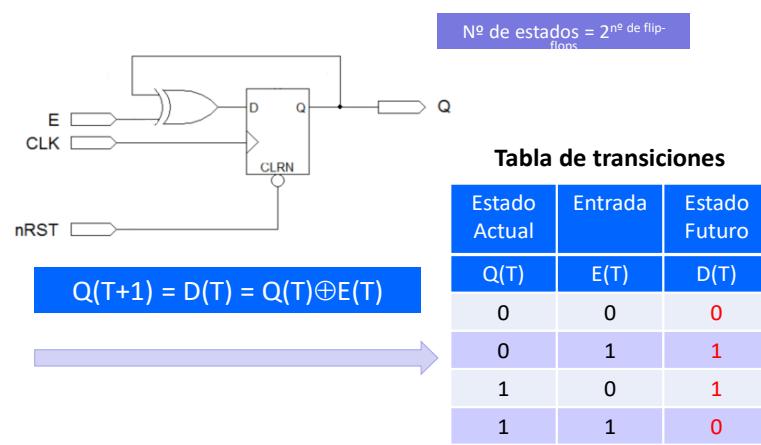
- ✓ El estado de memoria futuro es el valor que tienen las entradas de los flip-flops ($D(T)$) en cada flanco de reloj

$$Q(T + 1) = D(T) = F(Q(T), E(T))$$

BT3_S1 BT3_P4_5

Obtención de la Tabla de Transiciones

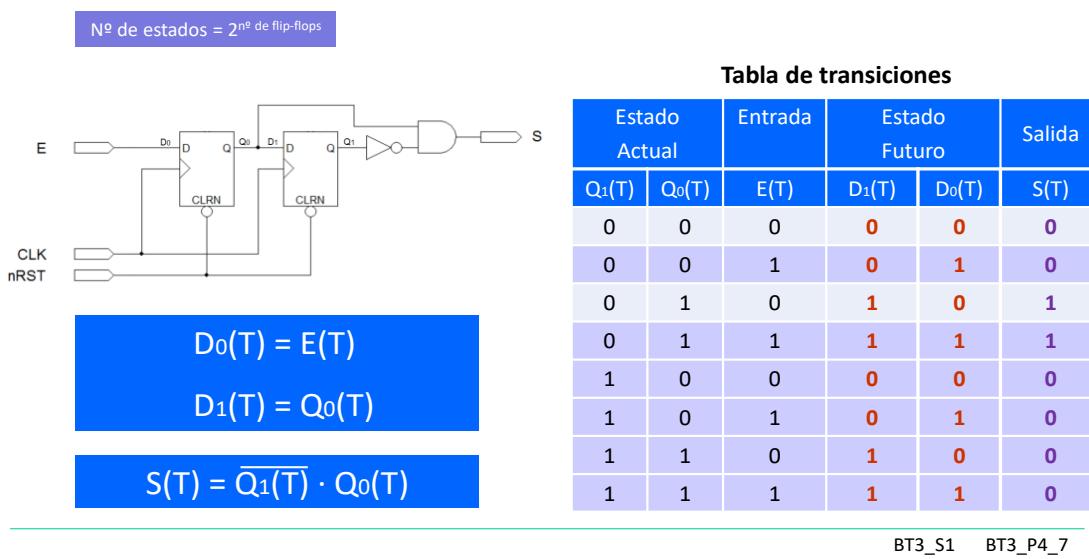
- ✓ Para realizar la tabla de transiciones hay que evaluar las ecuaciones de cambio de estado de memoria para **todos los valores de memoria y todas las combinaciones de entrada**.



BT3_S1 BT3_P4_6

Obtención de la Tabla de Transiciones

- ✓ Cuando la salida del circuito no es directamente la memoria de estado, suele añadirse también a la tabla de transiciones.



Obtención del Diagrama de Estados

1. Un **Diagrama de Estados** es la **representación gráfica** de la tabla de transiciones mediante un **grafo**.
2. Cada **estado de memoria** del circuito se representa como un **nodo** del grafo.
3. Los **cambios de estado** de memoria (las **transiciones**) son **arcos** del grafo que **se dirigen** desde **el estado de memoria actual hacia el estado de memoria futuro**.
4. Cada transición (arco) tiene asociada una combinación de valores de entrada.

BT3_S1 BT3_P4_8

Obtención del Diagrama de Estados

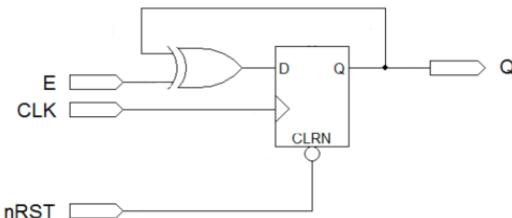
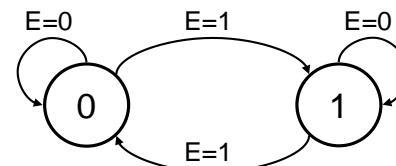


Tabla de transiciones

Estado Actual	Entrada	Estado Futuro
Q(T)	E(T)	D(T)
0	0	0
0	1	1
1	0	1
1	1	0

El número de nodos del diagrama es igual al número de estados de memoria del circuito: $2^{\text{nº de flip-flops}}$

De cada nodo sale un número de arcos igual al de combinaciones distintas de entrada: $2^{\text{nº de entradas}}$



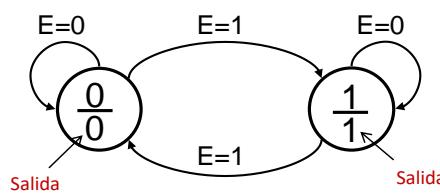
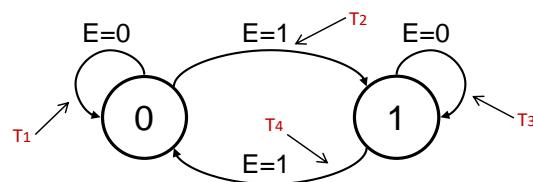
BT3_S1 BT3_P4_9

Obtención del Diagrama de Estados

Tabla de transiciones

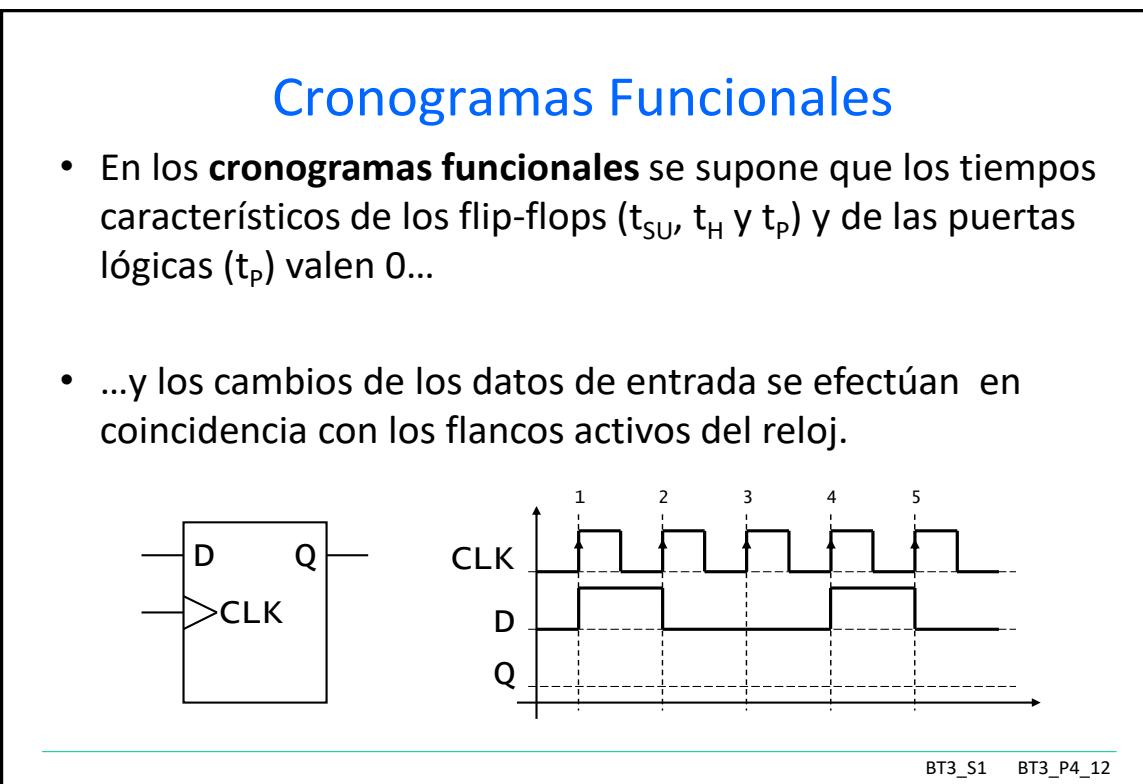
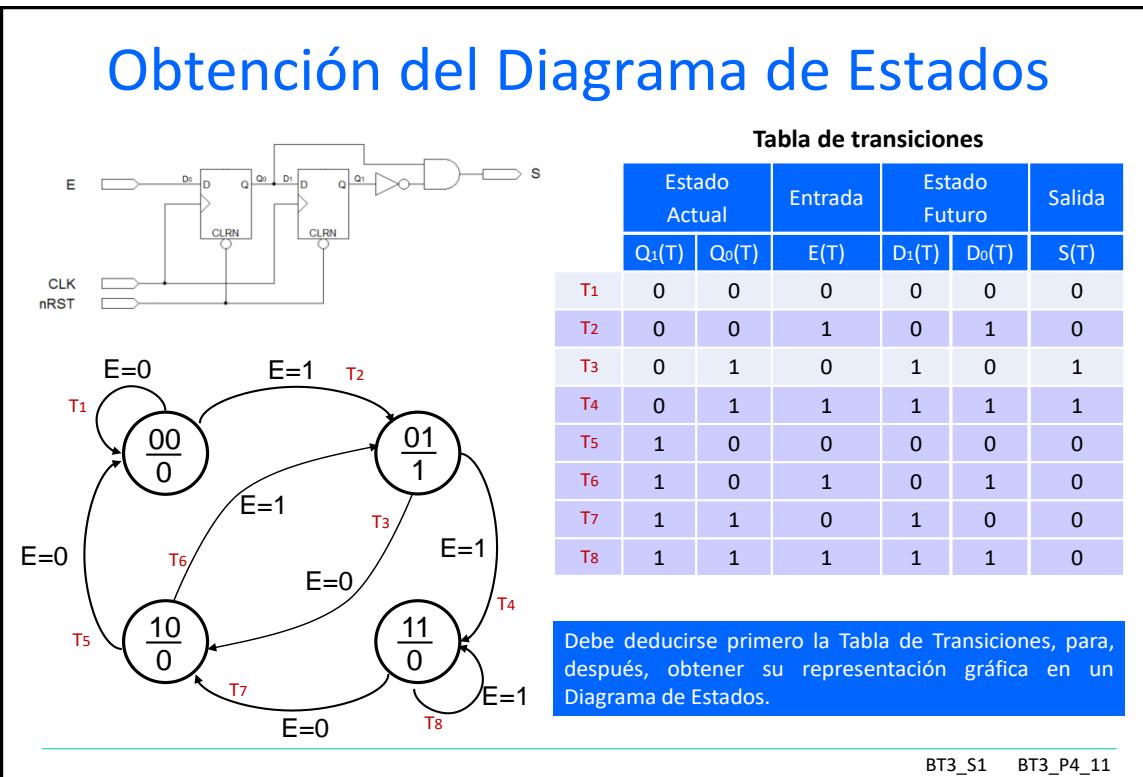
Estado Actual	Entrada	Estado Futuro
Q(T)	E(T)	D(T)
T ₁	0	0
T ₂	0	1
T ₃	1	0
T ₄	1	1

El número de arcos del diagrama de estados es igual al número de transiciones de la tabla.



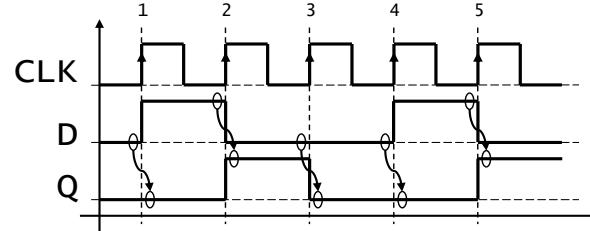
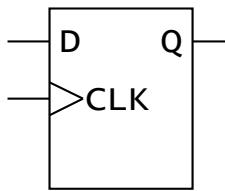
BT3_S1 BT3_P4_10

El valor de las salidas para cada estado de memoria se especifica en los nodos, ya que puede no coincidir con el estado de memoria.



Análisis mediante Cronogramas Funcionales

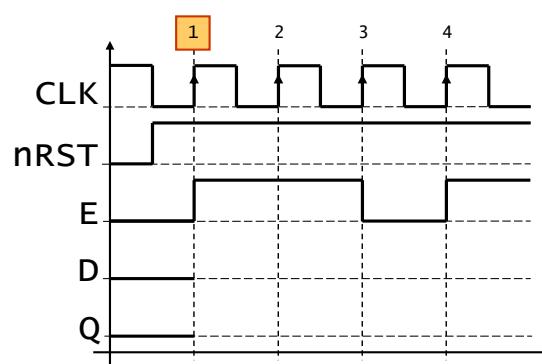
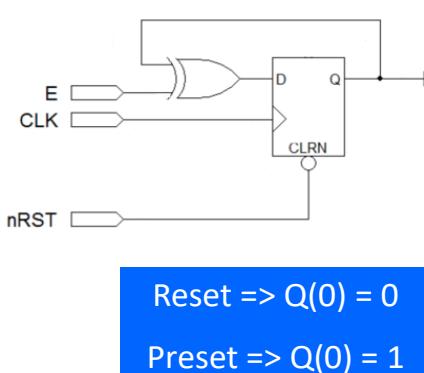
- El nivel que memorizan los flip-flops en cada flanco de reloj es el nivel presente en sus entradas justo antes del flanco.



BT3_S1 BT3_P4_13

Análisis mediante Cronogramas

- Hasta el primer flanco de reloj después de la desactivación de la señal de inicialización asíncrona, el valor de la memoria de estado, $Q(T=0)$ está determinado por dicha señal de inicialización.



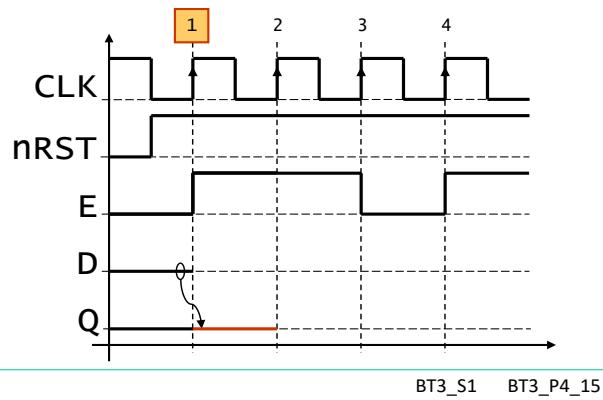
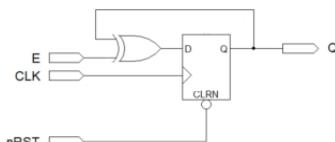
BT3_S1 BT3_P4_14

Metodología general de análisis

- ✓ Se calcula con las ecuaciones el valor de las entradas de los flip-flops en $T = 0$, antes de la ocurrencia del flanco para el paso a $T = 1$
- ✓ El valor calculado, $Q(1) = D(0)$, se apunta y se dibuja como estado de memoria en $T = 1$

$$Q(T+1) = D(T) = Q(T) \oplus E(T)$$

$$Q(1) = D(0) = 0 \oplus 0 = 0$$



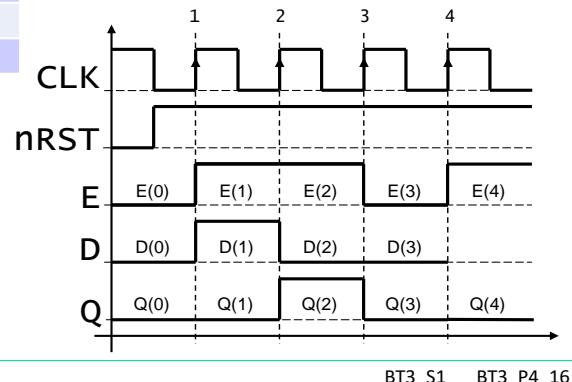
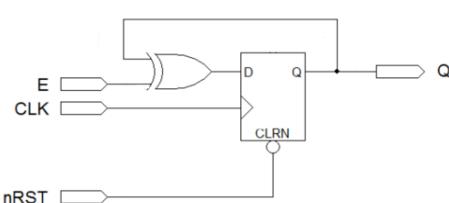
Metodología general de análisis

- ✓ Se repite el proceso para todos los flancos

	Entrada	Estado Actual	Estado Futuro
	E(T)	Q(T)	$Q(T+1) = D(T) = Q(T) \oplus E(T)$
T = 0	E(0) = 0	Q(0) = 0	$Q(1) = D(0) = 0 \oplus 0 = 0$
T = 1	E(1) = 1	Q(1) = 0	$Q(2) = D(1) = 1 \oplus 0 = 1$
T = 2	E(2) = 1	Q(2) = 1	$Q(3) = D(2) = 1 \oplus 1 = 0$
T = 3	E(3) = 0	Q(3) = 0	$Q(4) = D(3) = 0 \oplus 0 = 0$
			Q(4) = 0

El uso de una tabla auxiliar puede facilitar la realización del cronograma

Los valores de E(T) de la tabla se extraen de los estímulos del cronograma



BLOQUE TEMÁTICO 3**TÍTULO DE LA ACTIVIDAD:****Realización de ejercicios de análisis de funcionamiento de circuitos con flip-flops****CÓDIGO:
BT3_AINP1**

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Ejercicios Individual	TIPO:	No presencial	DURACIÓN:	60 minutos
------------	-----------------------	-------	---------------	-----------	------------

CALENDARIO:	Primera sesión. Previo a la segunda sesión.	REQUISITOS:	
-------------	--	-------------	--

CRITERIO DE ÉXITO:	
--------------------	--

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTEOVALUACIÓN: [entre 0 y 10 puntos]	puntos
------------------	---------	--	--------

1. Introducción

En esta actividad se van a realizar 3 ejercicios de análisis de circuitos con flip-flops completando cronogramas y realizando sus diagramas de estados. Antes de los ejercicios se incluye un ejemplo que repasa los procedimientos básicos de análisis expuestos en la última sesión presencial.

2. Ejemplo

En este ejemplo se va a completar el análisis del circuito de la figura 1, representando su funcionamiento en un cronograma y obteniendo su tabla de transiciones y diagrama de estados.

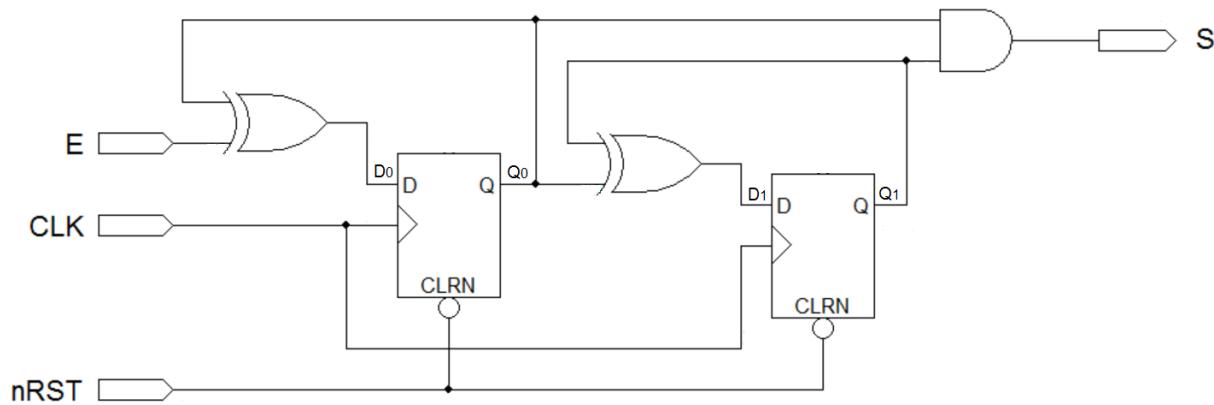


Figura 1

Obtención de las ecuaciones

En primer lugar hay que obtener las ecuaciones que determinan los cambios del estado de memoria (las transiciones) y las que definen las salidas del circuito. En el ejemplo, hay dos ecuaciones de estado (ya que el circuito cuenta con dos flip-flops) y una ecuación de salida; se deducen del esquema lógico del circuito:

$$D_0 = E \oplus Q_0$$

$$D_1 = Q_1 \oplus Q_0$$

$$S = Q_1 \cdot Q_0$$

Tabla de transiciones

La tabla de transiciones especifica el estado futuro de memoria, $Q(T+1) = D(T)$, que se alcanza en función del estado de memoria actual, $Q(T)$, y el valor actual de las entradas, $E(T)$. Se deduce evaluando las ecuaciones de estado y de salida para cada estado de memoria y combinación de entradas. El circuito tiene 2 flip-flops y, por tanto, cuatro estados ($2^2 = 4$); como tiene una sola entrada, la tabla de transiciones tendrá ocho filas, ya que para cada estado de memoria hay que evaluar dos combinaciones de entrada ($E = 0$ y $E = 1$)

Entrada	Estado Actual		Estado Futuro		Salida	
	E(T)	Q ₁ (T)	Q ₀ (T)	D ₁ (T)	D ₀ (T)	
0	0	0				
1	0	0				
0	0	1				
1	0	1				
0	1	0				
1	1	0				
0	1	1				
1	1	1				

$$D_0 = E \oplus Q_0$$

$$D_1 = Q_1 \oplus Q_0$$

$$S = Q_1 \cdot Q_0$$

Entrada	Estado Actual		Estado Futuro		Salida	
	E(T)	Q ₁ (T)	Q ₀ (T)	D ₁ (T)	D ₀ (T)	
0	0	0	0	0	0	0
1	0	0	0	0	1	0
0	0	1	1	1	1	0
1	0	1	1	1	0	0
0	1	0	1	0	0	0
1	1	0	1	1	1	0
0	1	1	0	1	0	1
1	1	1	0	0	0	1

Tabla de transiciones

Diagrama de estados

El diagrama de estados es una representación gráfica de la tabla de transiciones. Cada estado de memoria se representa mediante un círculo, etiquetado con el nombre del estado (al que se añade el valor de la salida en ese estado), y cada transición mediante una flecha, que parte de un estado (el estado actual de la transición representada) y se dirige a otro (el estado futuro de la transición) y tiene asociada la combinación de entrada que provoca la transición.

Entrada	Estado Actual		Estado Futuro		Salida	
	E(T)	Q ₁ (T)	Q ₀ (T)	D ₁ (T)	D ₀ (T)	
0	0	0	0	0	0	0
1	0	0	0	1	0	0
0	0	1	1	1	0	0
1	0	1	1	0	0	0
0	1	0	1	0	0	0
1	1	0	1	1	0	0
0	1	1	0	1	1	1
1	1	1	0	0	1	1

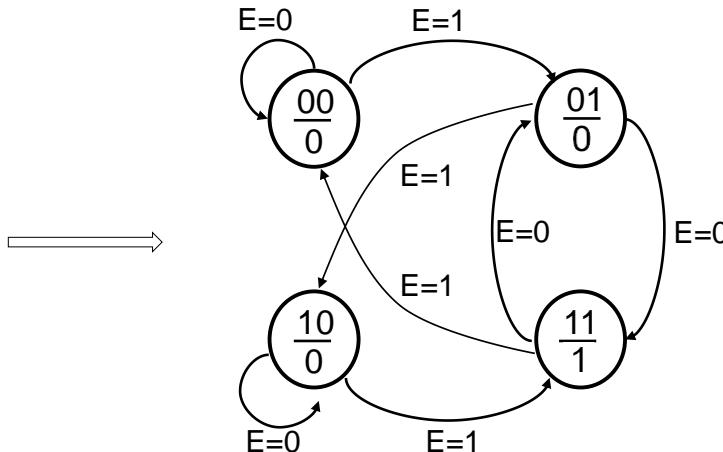


Diagrama de estados

Cronogramas funcionales

Para representar la respuesta del circuito a los estímulos especificados en un cronograma funcional es aconsejable ayudarse de una tabla para ir calculando las transiciones que se producen en cada flanco de reloj. Las columnas de la tabla son idénticas a las de la tabla de transiciones: entradas, estado actual, estado futuro y salidas; las filas representan instantes de tiempo:

Entrada	Estado Actual		Estado Futuro		Salida	
	E(T)	Q ₁ (T)	Q ₀ (T)	D ₁ (T)	D ₀ (T)	

Una vez dado el cronograma a completar (en el ejemplo que estamos realizando, el de la figura 2), debe procederse como se describe a continuación.

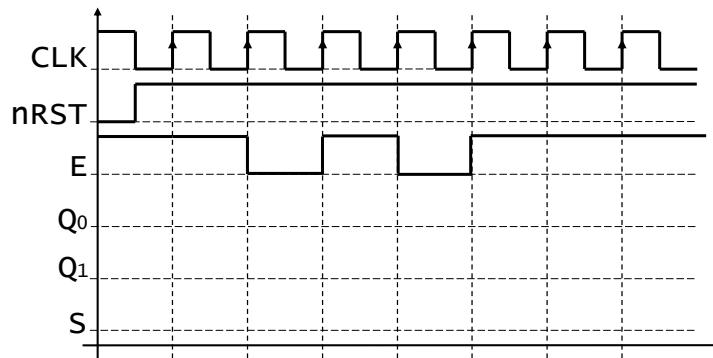
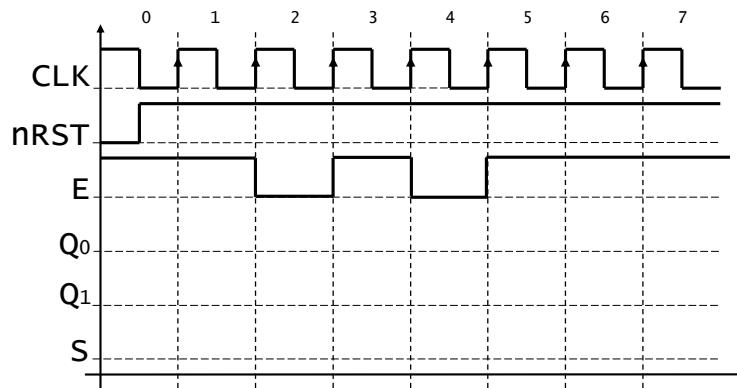


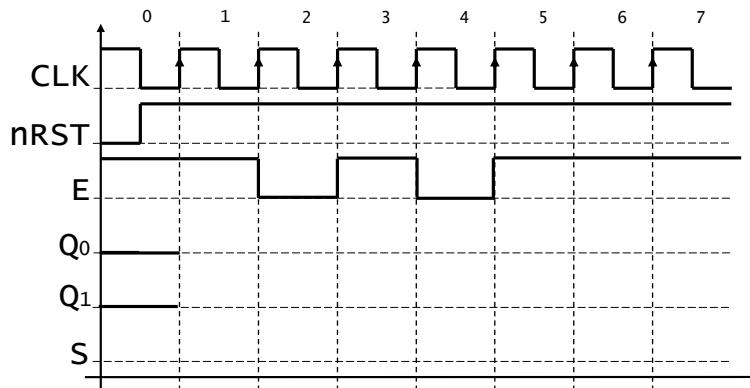
Figura 2

- En primer lugar deben numerarse los ciclos de reloj a partir del primer flanco activo en el que la señal de inicialización asíncrona esté **inactiva**. El número asociado a cada periodo de reloj indica el instante de tiempo que le corresponde.
 - La actividad de la entrada de inicialización asíncrona define el instante de tiempo inicial ($T = 0$), que se prolonga indefinidamente mientras dicha entrada permanece activa.



- Debe rellenarse el cronograma para $T = 0$ observando la conexión existente entre la entrada de inicialización y las entradas asíncronas de los flip-flops.

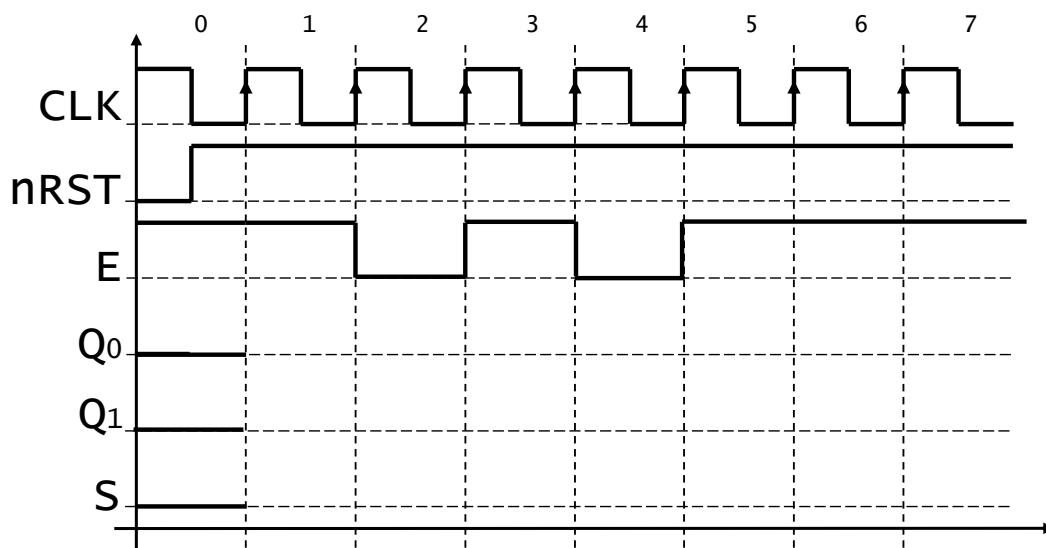
En el ejemplo, nRST está conectada a las entradas de reset:



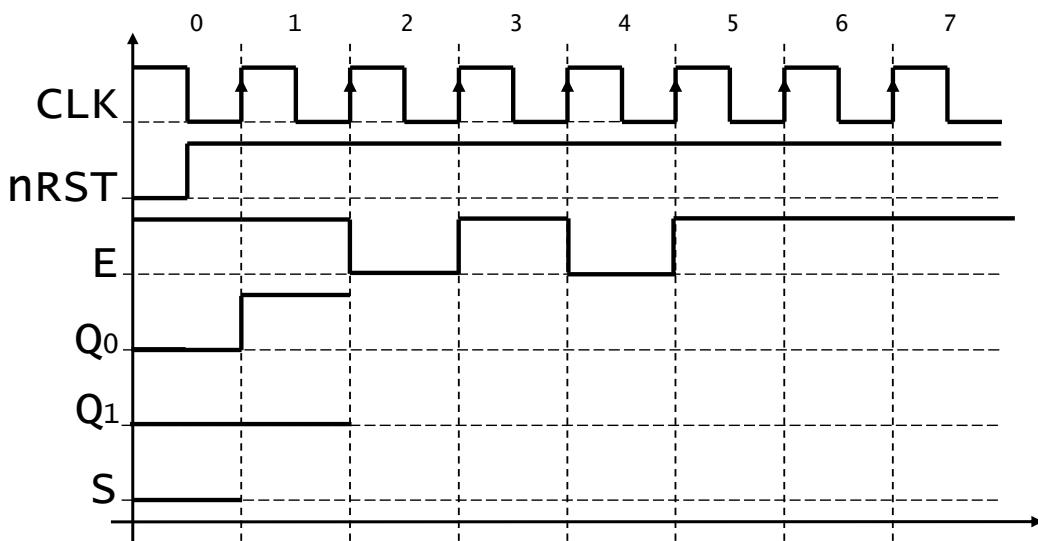
- III. Conocido el estado en $T = 0$, debe rellenarse la primera fila de la tabla auxiliar evaluando las ecuaciones extraídas del circuito:

Entrada	Estado Actual		Estado Futuro		Salida
	$E(T)$	$Q_1(T)$	$Q_0(T)$	$D_1(T)$	
	1	0	0	0	
$T = 0$					$S(T)$
					1
					0

El valor de salida calculado se representa para $T = 0$:



- IV. El estado futuro calculado en $T = 0$ ($D_1(0) = 0$ y $D_0(0) = 1$) es el que adopta el circuito tras la ocurrencia del flanco de reloj que da paso al instante $T = 1$ ($Q_1(1) = D_1(0)$ y $Q_0(1) = D_0(0)$); se dibuja por tanto el valor calculado para $T = 1$ en el cronograma:

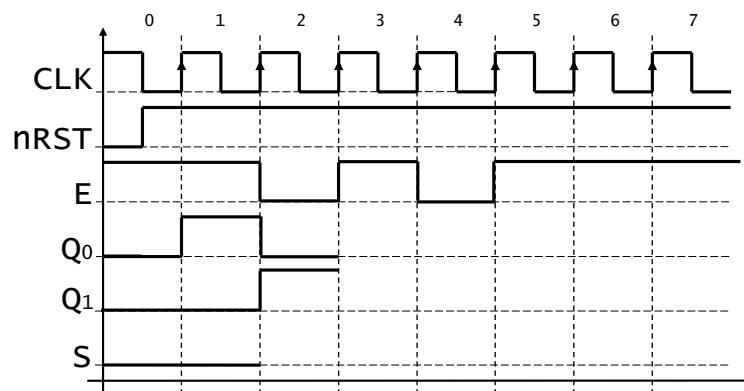


V. Y a partir de aquí se repite para cada ciclo de reloj la secuencia de operaciones descrita en III y IV. En el ejemplo:

- Conocido el estado en $T = 1$ se rellena la segunda fila de la tabla:

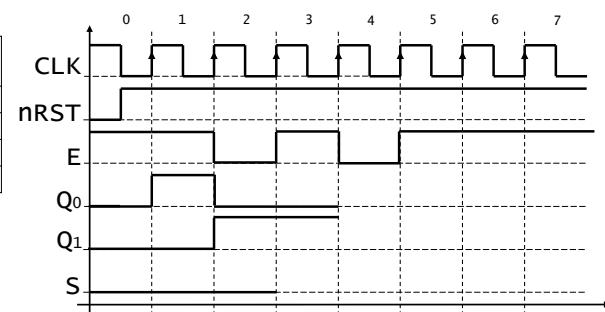
Entrada	Estado Actual		Estado Futuro		Salida	
	E(T)	Q ₁ (T)	Q ₀ (T)	D ₁ (T)	D ₀ (T)	
T = 0	1	0	0	0	1	0
T = 1	1	0	1	1	0	0

- Y se dibuja el cronograma (S(1), Q₁(2) y Q₁(2)):



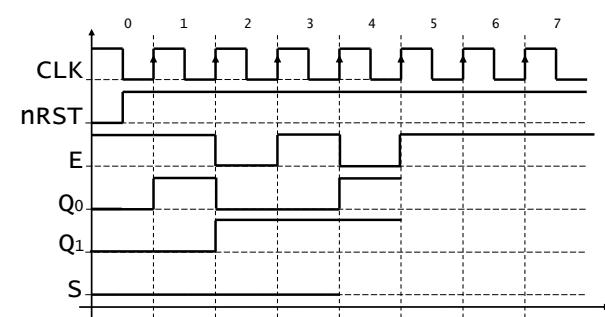
- En T = 2:

Entrada	Estado Actual		Estado Futuro		Salida	
	E(T)	Q ₁ (T)	Q ₀ (T)	D ₁ (T)	D ₀ (T)	
T = 0	1	0	0	0	1	0
T = 1	1	0	1	1	0	0
T = 2	0	1	0	1	0	0



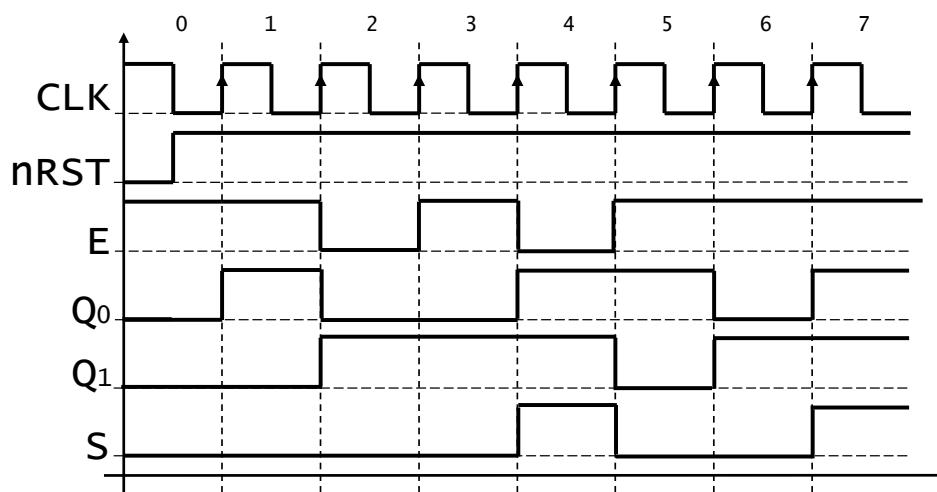
- En T = 3:

Entrada	Estado Actual		Estado Futuro		Salida	
	E(T)	Q ₁ (T)	Q ₀ (T)	D ₁ (T)	D ₀ (T)	
T = 0	1	0	0	0	1	0
T = 1	1	0	1	1	0	0
T = 2	0	1	0	1	0	0
T = 3	1	1	0	1	1	1



- El cronograma completo:

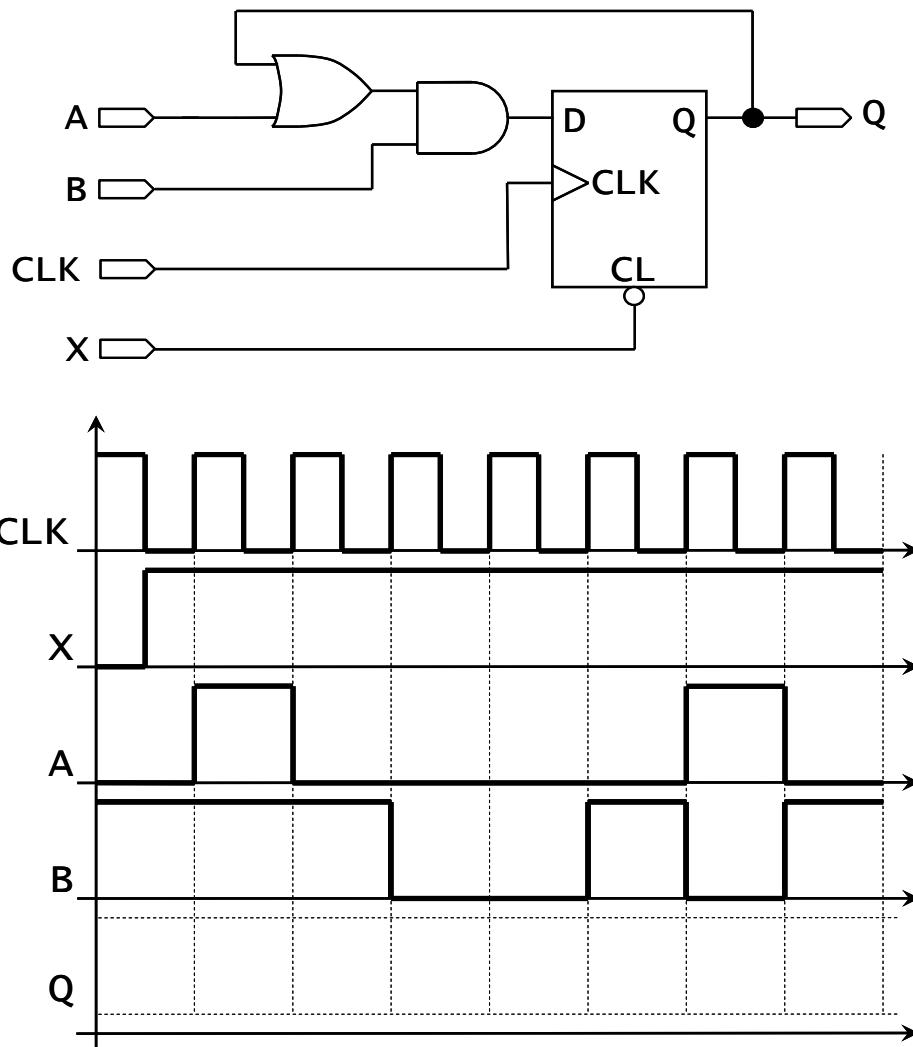
Entrada	Estado Actual		Estado Futuro		Salida	
	E(T)	Q ₁ (T)	Q ₀ (T)	D ₁ (T)	D ₀ (T)	
T = 0	1	0	0	0	1	0
T = 1	1	0	1	1	0	0
T = 2	0	1	0	1	0	0
T = 3	1	1	0	1	1	1
T = 4	0	1	1	0	1	0
T = 5	1	0	1	1	0	0
T = 6	1	1	0	1	1	0
T = 7	1	1	1	-	-	1



3. Ejercicios

Ejercicio 1

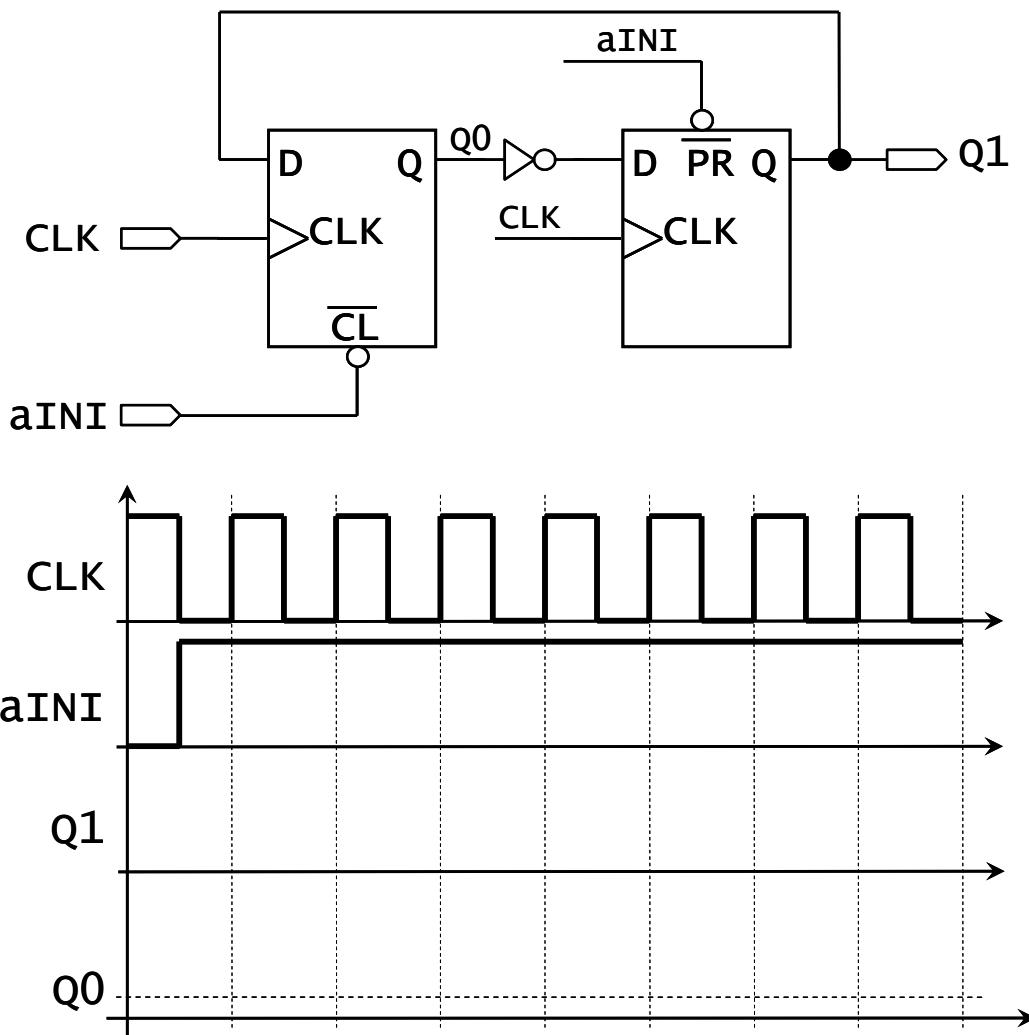
- a) Complete el cronograma adjunto para que represente el comportamiento del circuito de la figura. Considere que todos los elementos son ideales, esto es, que sus tiempos característicos valen cero.



- b) Represente el diagrama de estados del circuito anterior.

Ejercicio 2

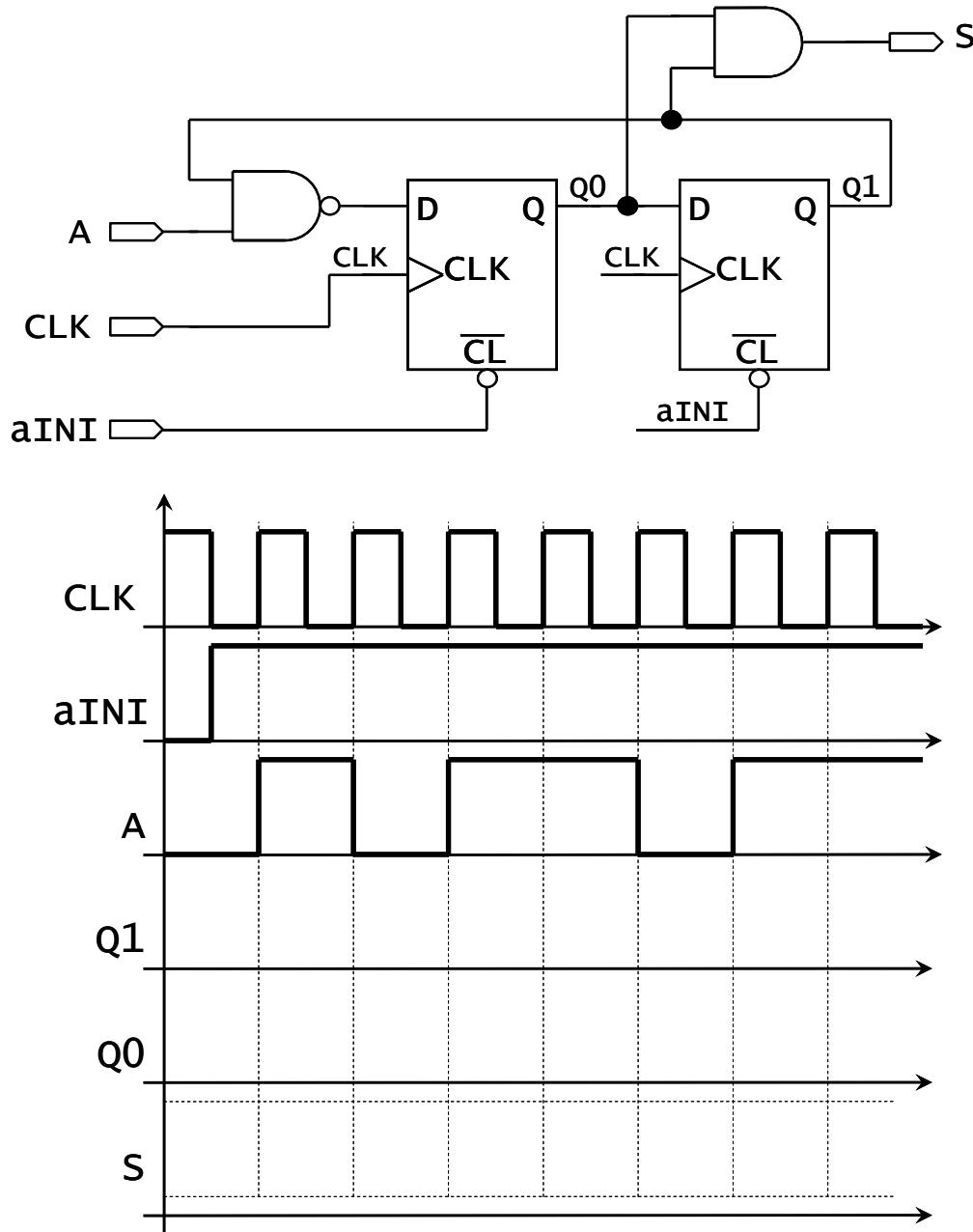
- a) Complete el cronograma adjunto para que represente el comportamiento del circuito de la figura. Considere que todos los elementos son ideales, esto es, que sus tiempos característicos valen cero.



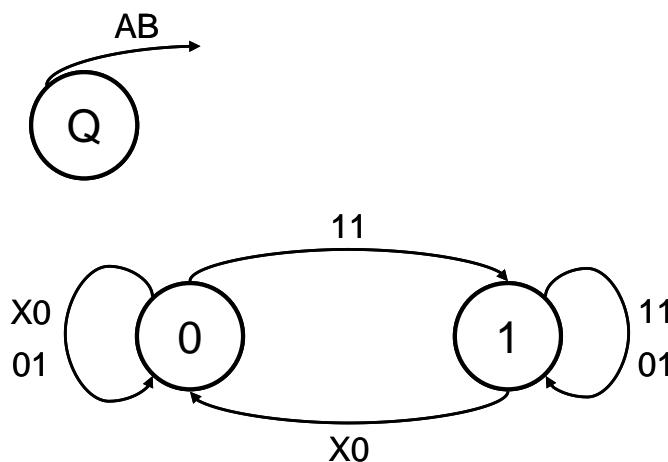
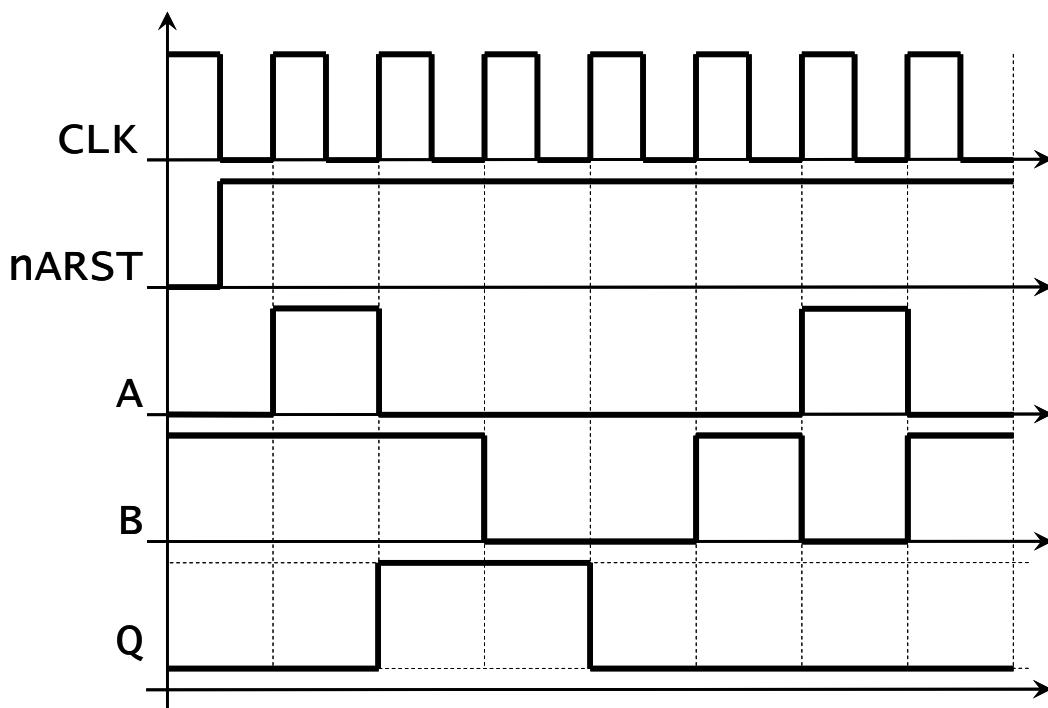
- b) Represente el diagrama de estados del circuito anterior.

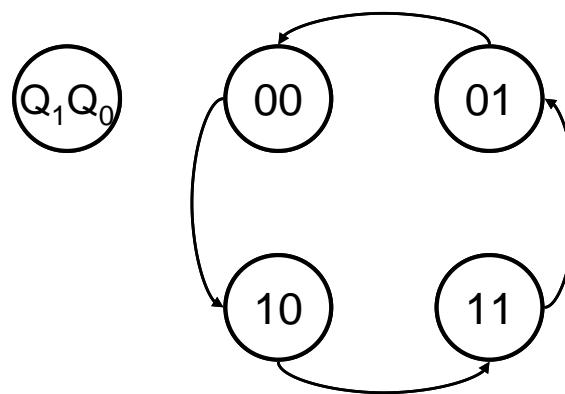
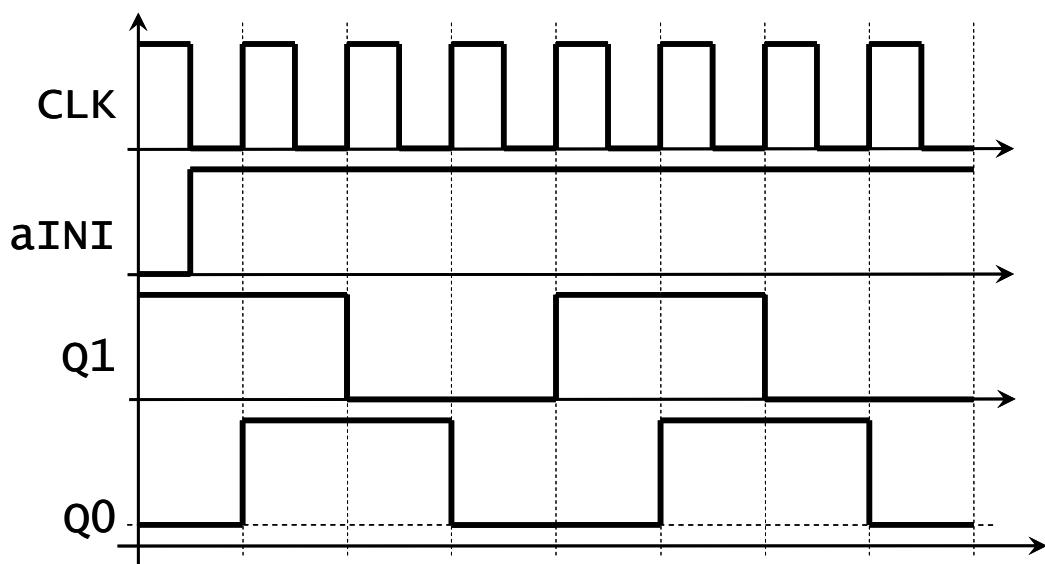
Ejercicio 3

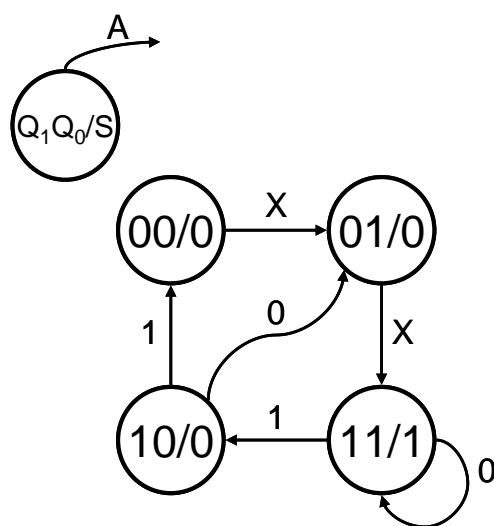
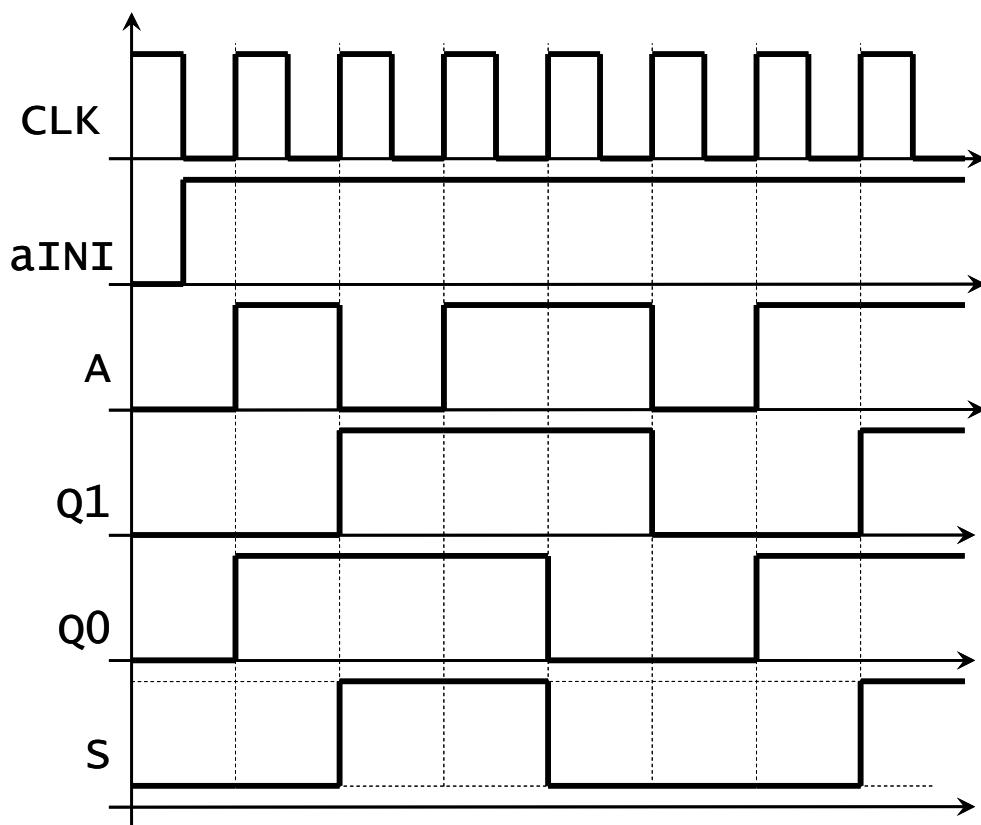
- a) Complete el cronograma adjunto para que represente el comportamiento del circuito de la figura. Considere que todos los elementos son ideales, esto es, que sus tiempos característicos valen cero.



- b) Represente el diagrama de estados del circuito anterior.

Soluciones**Ejercicio 1**

Ejercicio 2

Ejercicio 3

Esta página se ha dejado intencionadamente en blanco

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Lectura sobre la realización de entradas síncronas en flip-flops.

CÓDIGO:
BT3_AINP2

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Lectura y ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	40 minutos
------------	----------------------------------	-------	---------------	-----------	------------

CALENDARIO:	Primera sesión. Previo a la segunda sesión	REQUISITOS:	Metodología de síntesis de sistemas combinacionales. Flip-flop D
-------------	---	-------------	--

CRITERIO DE ÉXITO:	Se ha comprendido la metodología de realización de entradas síncronas en flip-flops.
--------------------	--

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTEOVALUACIÓN: [entre 0 y 10 puntos]	
------------------	---------	--	--

Flip-flop D con entradas síncronas

La memoria de los sistemas secuenciales síncronos cableados se construye con flip-flops tipo D. En algunos casos resulta suficiente utilizar como célula de memoria el flip-flop tipo D básico que ya conoce, pero en muchas ocasiones se puede simplificar considerablemente el diseño de un sistema si se emplean flip-flops que cuenten con **entradas síncronas** de control adicionales. Las entradas síncronas con más utilidad práctica son las de habilitación de reloj, de reset síncrono y de preset síncrono. El diseño de flip-flops con este tipo de entradas se puede realizar directamente, sin necesidad de recurrir a la realización de una tabla de verdad, empleando multiplexores, puertas AND y OR e inversores.

Entrada de habilitación de reloj

La entrada de habilitación de reloj –también se suele llamar, simplemente, *entrada de habilitación*, o por su nombre original en inglés: entrada de *clock enable*- sirve para controlar los instantes de tiempo (flancos activos) en que se permite al flip-flop memorizar un nuevo valor. Cuando está activa permite que el flip-flop *capture*, en los flancos activos de reloj, el bit presente en la entrada D; en los flancos de reloj en los que la habilitación está desactivada el flip-flop mantiene el valor memorizado. El funcionamiento de esta entrada se ilustra en el cronograma de la figura 1.

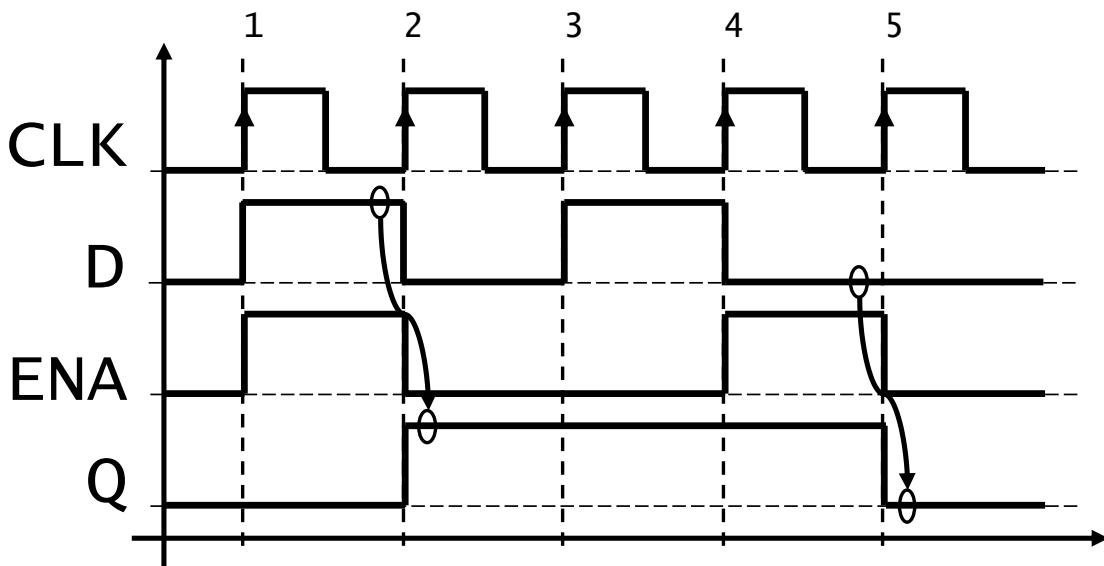


Figura 1.- Entrada de habilitación de reloj

En el cronograma puede observarse como, cuando la entrada de habilitación de reloj (**ENA**) está activa (en los flancos de reloj 2 y 5), el flip-flop tipo D captura el dato de entrada, mientras que en el resto de los flancos retiene el valor almacenado.

Tal y como muestra la figura 2a, la habilitación de reloj se materializa empleando un multiplexor de dos canales que permite controlar la conexión lógica de la entrada del flip-flop con el dato de entrada o el dato memorizado: cuando la entrada de habilitación está activa se selecciona en el multiplexor el dato de entrada (**D**); cuando no lo está se selecciona el dato almacenado (**Q**): este sencillo mecanismo permite controlar los flancos de reloj en que se escribe un dato en el flip-flop (**ENA** a nivel alto) y aquellos en que se mantiene el dato almacenado (**ENA** a nivel bajo). La única variante a esta realización de la habilitación de reloj consiste en el intercambio de las conexiones de entrada del multiplexor para cambiar el nivel de actividad de la entrada de habilitación de reloj (Figura 2b).

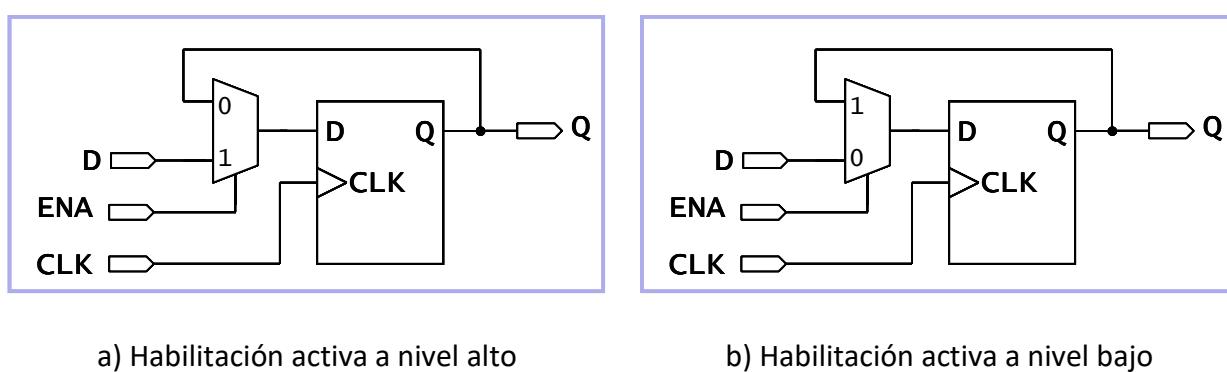


Figura 2.- Habilitación de reloj

Entradas de reset y preset síncrono

Una entrada de reset síncrono sirve para poner a 0 la memoria de un flip-flop síncronamente, en un flanco activo de reloj. Una entrada de preset síncrono funciona de manera semejante, pero pone a 1 la memoria de estado. La figura 3 muestra en un cronograma el funcionamiento de un flip-flop con una entrada de reset síncrono. En los flancos de reloj en que la entrada de reset está a nivel alto la salida (**Q**) se pone a 0.

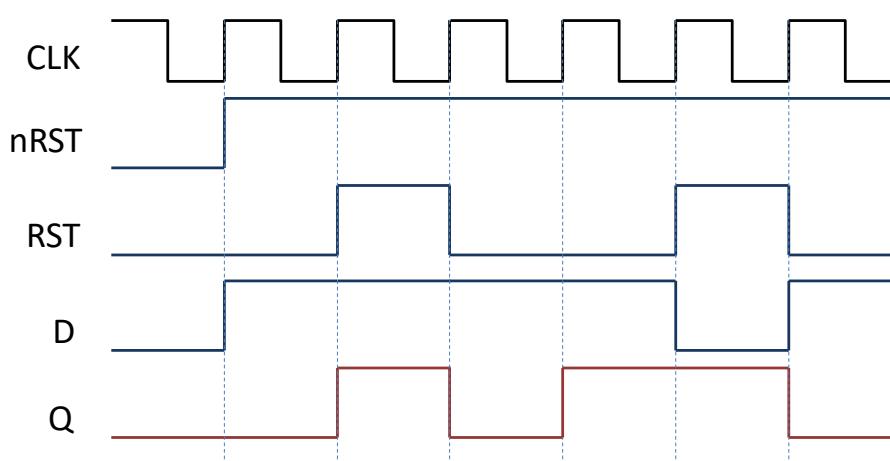


Figura 3.- Reset síncrono

La estructura lógica de un flip-flop con reset o preset síncrono se muestra en la figura 4a. Para la realización de un reset síncrono la lógica de realización se reduce a una puerta AND (figura 4b1), en

la que la entrada de reset, activa a nivel bajo, regula el *paso* del dato o la puesta a 0 de la salida: cuando está activa fuerza un cero ($\text{DATO}\cdot 0 = 0$), y cuando no lo está provoca que el nivel de salida de la puerta coincida con el que tiene el dato de entrada ($\text{DATO}\cdot 1 = \text{DATO}$). Si se desea que la entrada de reset sea activa a nivel alto basta con añadir un inversor (figura 4b2).

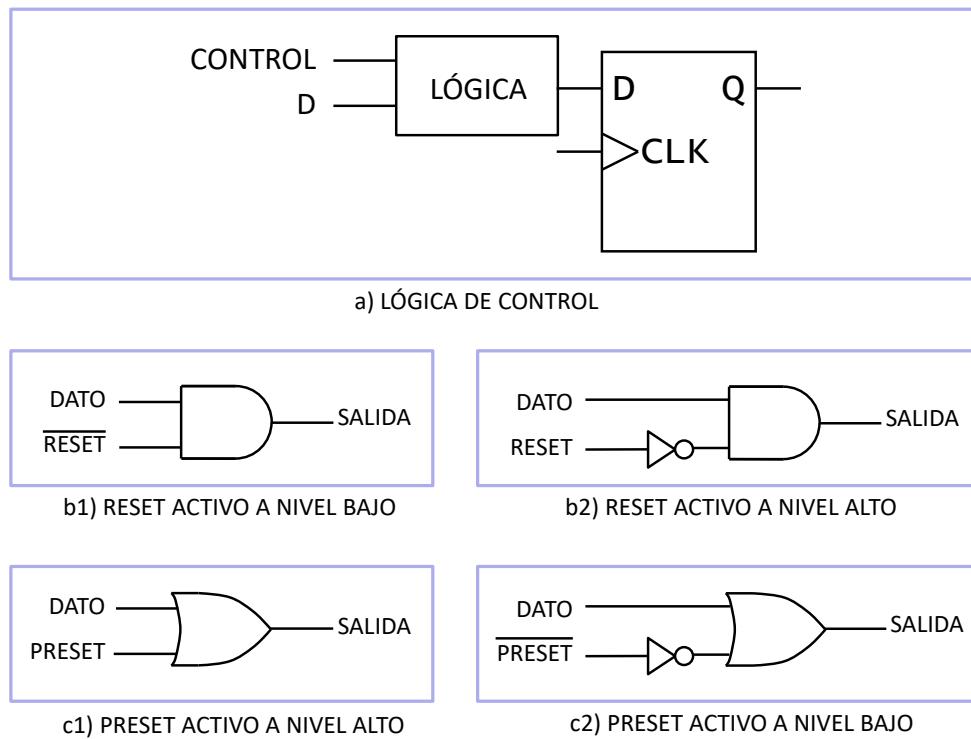


Figura 4.- Entradas de control síncronas

El circuito de la figura 5 muestra un flip-flop con una entrada de reset síncrono. Cuando la entrada **RST** se activa, a nivel alto, en la entrada D del flip-flop se fuerza un 0; cuando está inactiva (a nivel bajo), es el dato de entrada, **D**, el que define el nivel lógico en la entrada del flip-flop (en **D**).

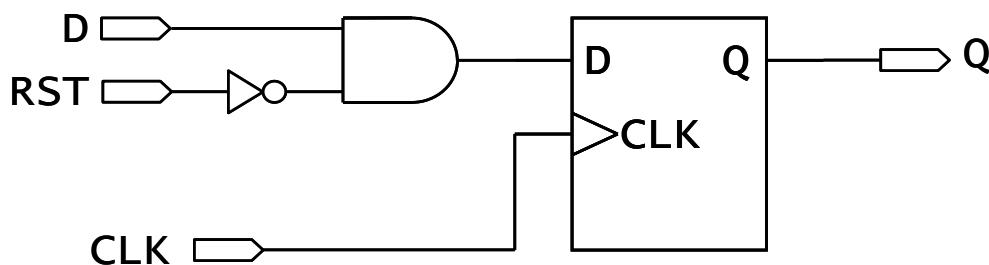


Figura 5.- Flip-flop con reset síncrono activo a nivel alto

La realización de un preset síncrono (figura 4c1) es análoga: en este caso se emplea una puerta OR, cuya salida se pone a 1 cuando se activa el preset ($1+\text{DATO} = 1$) y permite el *paso* del dato cuando

el preset está inactivo ($0+DATO = DATO$). También en este caso basta con utilizar un inversor para cambiar el nivel activo de la entrada (figura 4c2). El cronograma de la figura 6 representa el funcionamiento de un flip-flop con preset síncrono activo a nivel bajo.

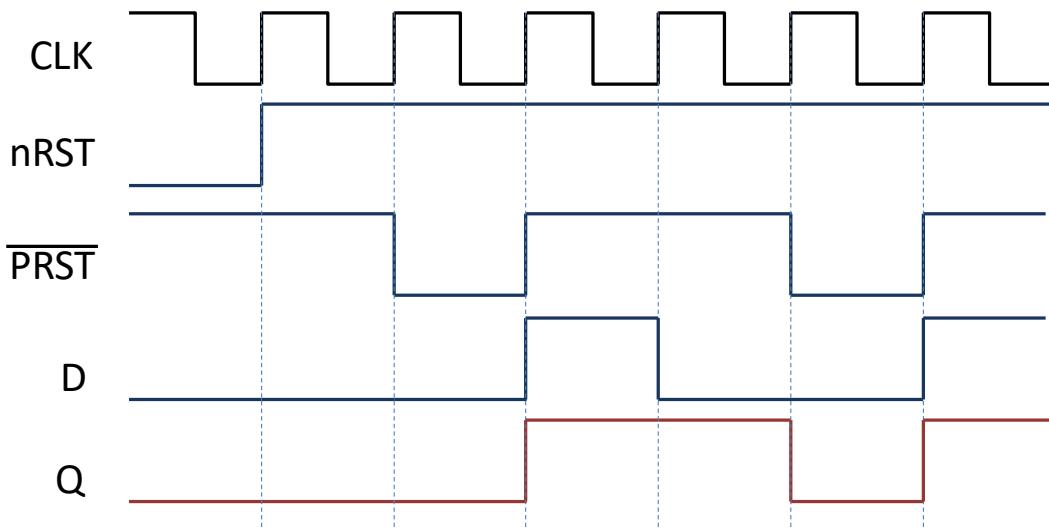


Figura 6.- Preset síncrono

En la figura 7 se muestra un flip-flop con entrada de preset síncrono: cuando la entrada **PRST** está activa, a nivel alto, fuerza un 1 en la entrada D del flip-flop; cuando está inactiva (a nivel bajo), en la entrada **D** del flip-flop hay el mismo nivel lógico que en la entrada **D** del circuito.

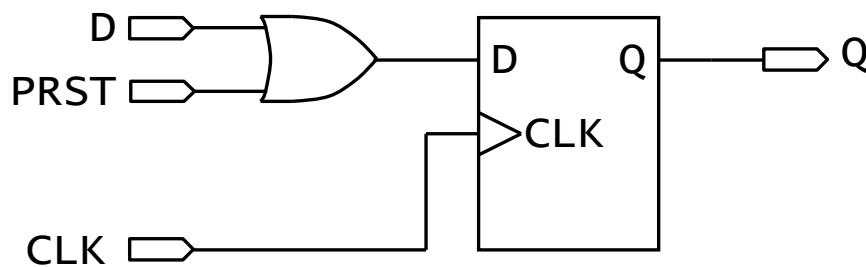


Figura 7.- Flip-flop con preset síncrono activo a nivel alto

Si se desea que la entrada de preset síncrono sea activa a nivel bajo, basta con añadir un inversor entre dicha entrada y la puerta OR, de manera análoga a la mostrada en el circuito de la figura 5.

Flip-flops con varias entradas de control síncronas

Un flip-flop puede contar con varias entradas síncronas de control. Este tipo de flip-flops se pueden diseñar directamente *mezclando* los circuitos que permiten realizar cada una de las funciones de control.

Flip-flops con entrada de habilitación y de control síncrona

Si se desea realizar un flip-flop con habilitación de reloj y una entrada de control síncrona, pueden emplearse dos estructuras diferentes (figura 8). Si la lógica de control síncrona (la puerta AND u OR) se conecta directamente a la entrada del flip-flop, el control ejercido por la entrada de reset o preset no requiere que la habilitación esté activa (Figura 8a). Si es la salida del multiplexor la que se conecta a la entrada del flip-flop, para que el reset o el preset tengan efecto es necesario que la entrada de habilitación esté activa (figura 8b).

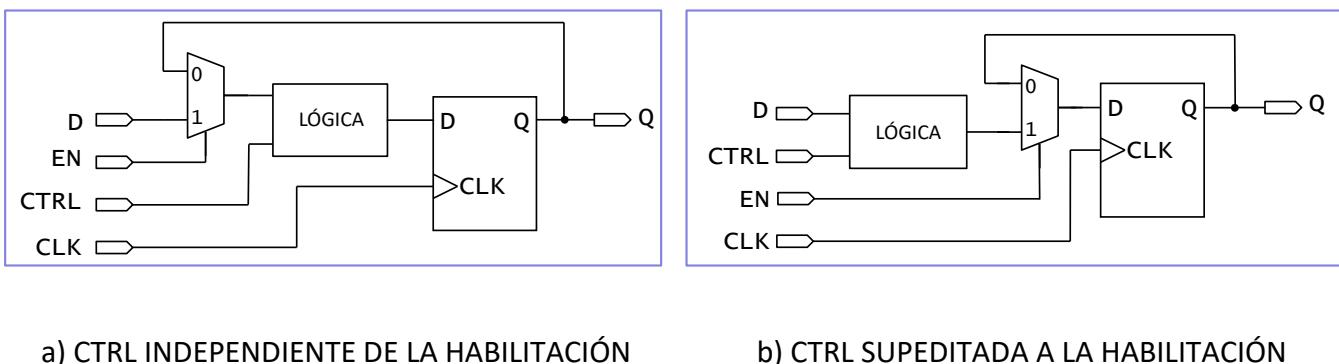


Figura 8.- Entrada de Habilitación y control

El circuito de la figura 9 muestra un flip-flop con habilitación de reloj, activa a nivel alto, y entrada de reset, independiente de la habilitación y activo a nivel bajo. Para que el flip-flop memorice el valor de la entrada D es necesario que la entrada de habilitación esté activa (a nivel alto) y la entrada de reset síncrono inactiva (a nivel alto); la actuación de la entrada de reset síncrono no está supeditada a la habilitación: si se activa (a nivel bajo), fuerza un 0 en la entrada D del flip-flop independientemente de que la entrada de habilitación esté activa o no.

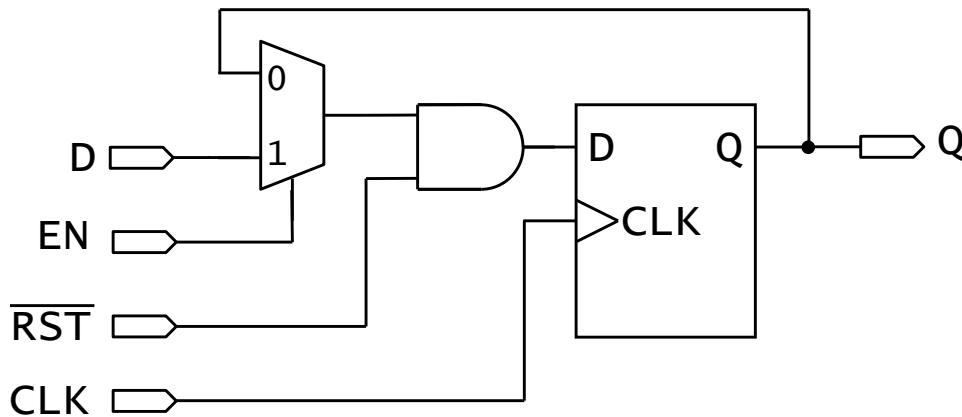


Figura 9.- Flip-flop con entrada de habilitación y entrada de reset síncrono activo a nivel bajo

Si se desea que la actuación del reset requiera de la activación de la entrada de habilitación de reloj, hay que cambiar la ubicación de la puerta AND del circuito de la figura 9, de modo que su salida sea una de las entradas del multiplexor (Figura 10).

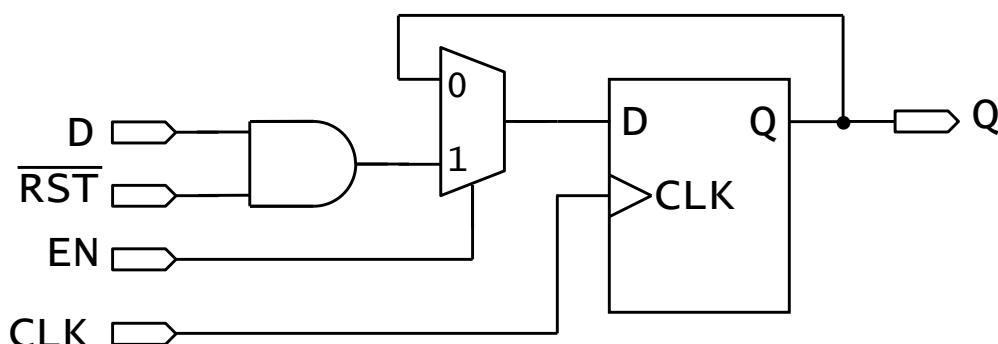
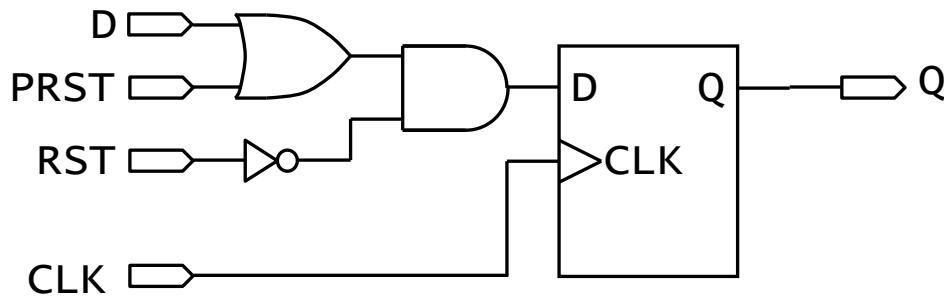


Figura 10.- Flip-flop con entrada de reset supeditada a la habilitación

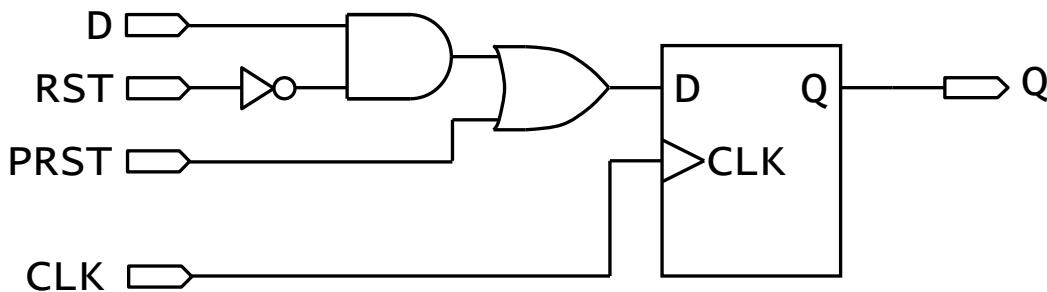
En este circuito la activación del reset síncrono no afecta al nivel lógico en la entrada del flip-flop si la entrada de habilitación está inactiva; este es el único detalle que diferencia su funcionamiento del circuito de la figura 9: el reset síncrono depende de la habilitación.

Flip-flops con entrada de reset y preset síncrono

También es posible que un flip-flop disponga simultáneamente de entradas de reset y preset síncrono. En este caso, una de las dos debe ser prioritaria sobre la otra, ya que en caso de activarse ambas, al ser contradictorio e incompatible su efecto sobre el estado de memoria del flip-flop (una ordena la puesta a 0 y la otra la puesta a 1), debe prevalecer una de las dos. En la figura 11 se muestran dos ejemplos que ilustran el mecanismo lógico para la realización de este tipo de circuitos.



(a) Flip-flop D con entradas de preset y reset síncronos. Reset más prioritario.



(b) Flip-flop D con entradas de preset y reset síncronos. Preset más prioritario.

Figura 11.- Flip-flop con entradas síncronas de reset y preset

En el flip-flop de la figura 11a, cuando el reset está activo (es activo a nivel alto) se fuerza un 0 lógico en la salida de la puerta AND y, por tanto, en la entrada D del flip-flop, y resulta indiferente, en este caso, que la entrada de preset síncrono esté, o no, activa. La activación del preset síncrono (es activo también a nivel alto) sólo fuerza un 1 lógico en la entrada D del flip-flop si el reset síncrono está inactivo (a nivel bajo). En consecuencia, en este circuito el reset es prioritario sobre el preset. Observe, por último, que para que el flip-flop capture el dato de la entrada D ambas entradas de control síncrono deben permanecer inactivas. En cambio, en el circuito de la figura 11b la entrada de preset es prioritaria sobre la de reset. La justificación de esta prioridad se va a omitir, ya que es análoga a la realizada para el caso de la figura 11a y puede ser deducida fácilmente por el lector.

El mecanismo que determina la prioridad de una u otra entrada resulta evidente: es más prioritaria aquella cuya lógica de realización (la puerta AND en el reset o la OR en el preset) controla directamente –está conectada a...– la entrada D del flip-flop.

Flip-flop con entrada de habilitación, reset y preset síncrono

Para terminar, un ejemplo que reúne las tres entradas de control y combina las técnicas de diseño explicadas hasta aquí: el circuito de la figura 12 muestra un flip-flop con habilitación de reloj activa a nivel alto, entrada de reset síncrono, activa a nivel bajo y supeditada a la habilitación de reloj, y entrada de preset síncrono, activa a nivel alto, independiente de la habilitación y prioritaria sobre la entrada de reset.

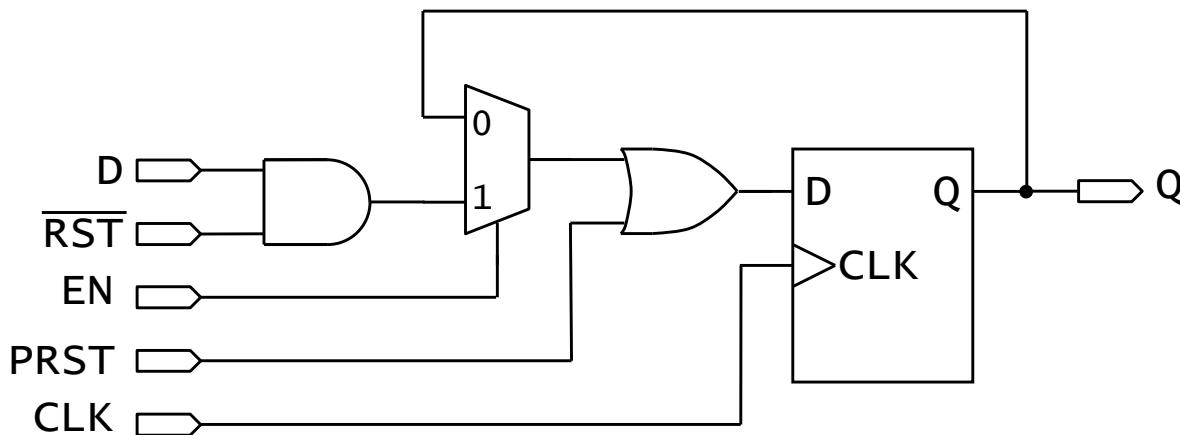


Figura 12.- Flip-flop con entrada de habilitación, reset y preset síncrono

Cuando un flip-flop dispone de las tres entradas de control síncrono (es un caso infrecuente), el número de variedades de diseño aumenta, pero la técnica de diseño sigue consistiendo en combinar y conectar la lógica que permite la realización individual de cada una de las entradas síncronas de control.

Representación del funcionamiento de flip-flops con entradas síncronas de control

El funcionamiento de un flip-flop con entradas síncronas de habilitación, reset o preset se representa con los mismos procedimientos que se utilizan para el resto de los circuitos secuenciales síncronos: tablas de transición, cronogramas y diagramas de estado.

El cronograma de la figura 13 representa, por ejemplo, el funcionamiento de un flip-flop con entrada de habilitación, activa a nivel alto, y entrada de reset síncrono, independiente de la habilitación y activa también a nivel alto. El flip-flop dispone, además, de una entrada de reset asíncrono activa a nivel bajo.

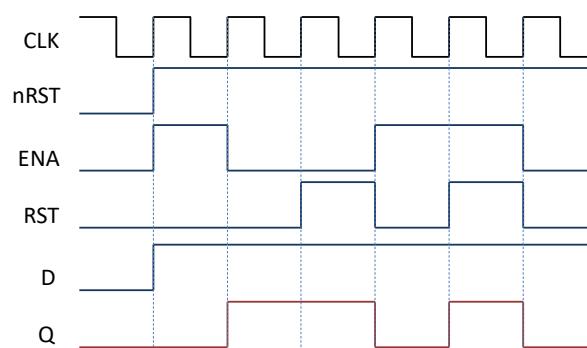


Figura 13.- Cronograma de flip-flop con entrada de habilitación y reset

El funcionamiento de este mismo flip-flop queda completamente definido en la tabla de transiciones de la figura 14.

	nRST	RST	ENA	D(T)	CLK	Q(T+1)
Reset Asíncrono	0	X	X	X	X	0
Reset Síncrono	1	1	X	X	↑	0
Captura de dato	1	0	1	0	↑	0
Captura de dato	1	0	1	1	↑	1
Retención	1	0	0	X	↑	Q(T)

Figura 14.- Tabla de transiciones de un flip-flop con entrada de habilitación y reset

Y la figura 15 muestra la representación de su diagrama de estados.

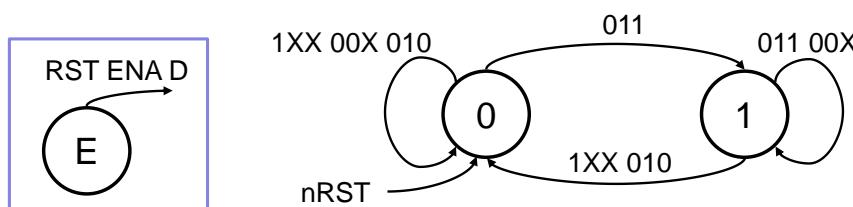


Figura 15.- Diagrama de estados de un flip-flop con entrada de habilitación y reset

Ejercicios

1.- Rellene la tabla de transiciones de un flip-flop con entrada de habilitación activa a nivel alto y entrada de reset asíncrono activo a nivel bajo.

nRST	ENA	D(T)	Q(T+1)

2.- Describa la funcionalidad e interfaz del flip-flop descrito por la siguiente tabla de transiciones.

Nota: Y es una entrada de reset o preset síncrono; Z es una entrada de control asíncrona.

Z	ENA	Y	D(T)	Q(T+1)
0	X	X	X	1
1	1	0	X	0
1	1	1	0	0
1	1	1	1	1
1	0	X	X	Q(T)

3.- Describa la funcionalidad e interfaz del flip-flop descrito por la siguiente tabla de transiciones.

Nota: X e Y son entradas de los siguientes tipos: de habilitación, reset o preset síncrono; Z es una entrada de control asíncrona.

Z	X	Y	D(T)	Q(T+1)
0	X	X	X	0
1	1	X	X	Q(T)
1	0	1	X	1
1	0	0	0	0
1	0	0	1	1

4.- Realice el circuito del ejercicio 2

5.- Realice el diagrama de estados del circuito del ejercicio 2.

6.- Realice el diagrama de estados de un flip-flop con entrada de reset síncrono activa a nivel bajo.

Soluciones

1.- Rellene la tabla de transiciones de un flip-flop con entrada de habilitación activa a nivel alto y entrada de reset asíncrono activo a nivel bajo.

nRST	ENA	D(T)	Q(T+1)
0	X	X	0
1	1	0	0
1	1	1	1
1	0	X	Q(T)

2.- Describa la funcionalidad e interfaz del flip-flop descrito por la siguiente tabla de transiciones.

Nota: Y es una entrada de reset o preset síncrono; Z es una entrada de control asíncrona.

Z	ENA	Y	D(T)	Q(T+1)
0	X	X	X	1
1	1	0	X	0
1	1	1	0	0
1	1	1	1	1
1	0	X	X	Q(T)

Flip-flop con entradas de habilitación de reloj (ENA) activa a nivel alto, reset síncrono (Y) activa a nivel bajo supeditado a la habilitación de reloj, y preset asíncrono (Z) activa a nivel bajo.

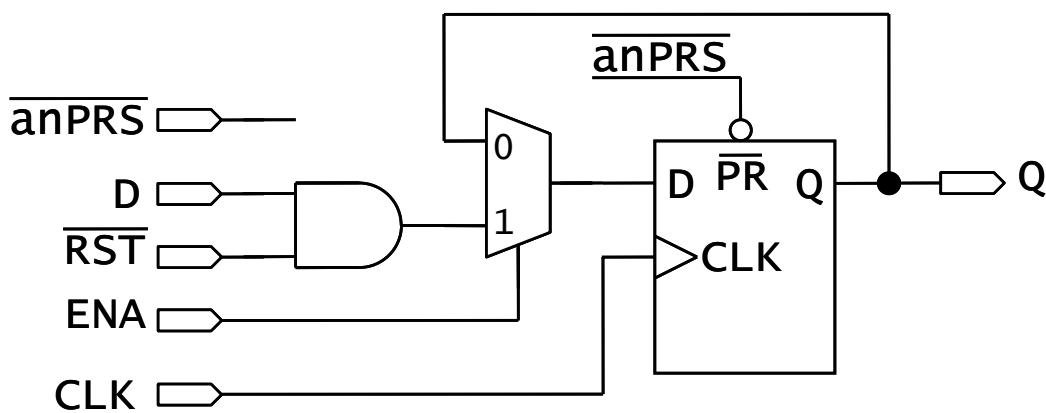
3.- Describa la funcionalidad e interfaz del flip-flop descrito por la siguiente tabla de transiciones.

Nota: X e Y son entradas de los siguientes tipos: de habilitación, reset o preset síncrono; Z es una entrada de control asíncrona.

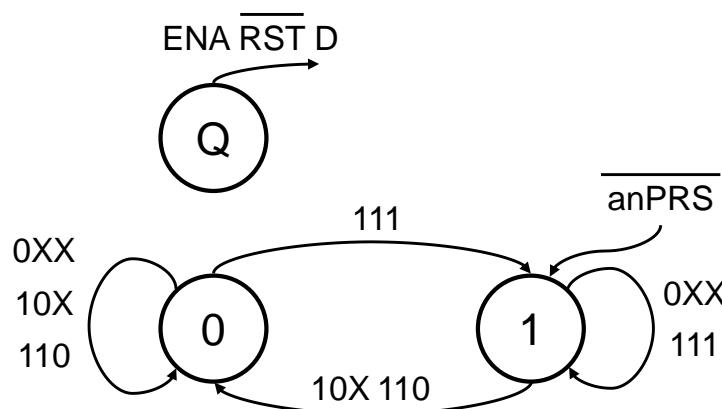
Z	X	Y	D(T)	Q(T+1)
0	X	X	X	0
1	1	X	X	Q(T)
1	0	1	X	1
1	0	0	0	0
1	0	0	1	1

Flip-flop con entradas de habilitación de reloj (X) activa a nivel bajo, preset síncrono (Y) activo a nivel alto supeditado a la habilitación de reloj, y reset asíncrono (Z) activa a nivel bajo.

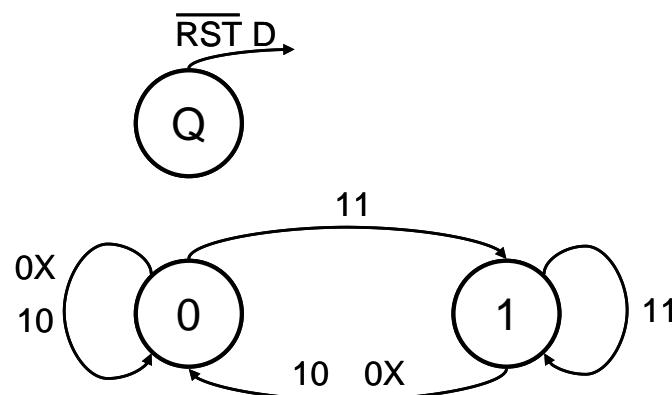
4.- Realice el circuito del ejercicio 2



5.- Realice el diagrama de estados del circuito del ejercicio 2.



6.- Realice el diagrama de estados de un flip-flop con entrada de reset síncrono activa a nivel bajo.



BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Ejercicios con flip-flops.

CÓDIGO:
BT3_ATGP1

GRUPO		FECHA:	
-------	--	--------	--

MODALIDAD:	Aprendizaje Cooperativo	TIPO:	Presencial	DURACIÓN:	80 minutos
------------	-------------------------	-------	------------	-----------	------------

CALENDARIO:	Segunda sesión	REQUISITOS:	
-------------	----------------	-------------	--

CRITERIO DE ÉXITO:	Realización correcta de los ejercicios.
--------------------	---

ROLES:	No hay distribución diferenciada de roles entre los miembros
--------	--

HABILIDADES SOCIALES:	Capacidad para expresar y sintetizar ideas, capacidad para escuchar, asimilar y organizar la información presentada por otra persona. Capacidad para razonar y analizar.
-----------------------	--

REFLEXIÓN DE GRUPO:	
---------------------	--

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	
------------------	------------	-------------------------------------	--

Descripción de la actividad

Esta actividad se compone de cuatro partes. Cada parte consta de 2 ejercicios de un mismo tipo. Los miembros del grupo se dividen en dos subgrupos. Cada subgrupo resolverá un ejercicio de cada parte. Una vez resueltos los problemas de la misma parte, los dos subgrupos del grupo comentarán los resultados. A continuación el profesor mostrará la solución de los dos ejercicios. Finalizada la actividad, uno de los miembros del grupo (el que el profesor indique) realizará una prueba breve acerca de los contenidos trabajados. La nota, que se aplicará a todos los miembros del grupo, servirá para calcular la calificación global de la actividad, junto con el entregable realizado por el grupo.

La distribución de tiempos para la realización de la actividad es la siguiente:

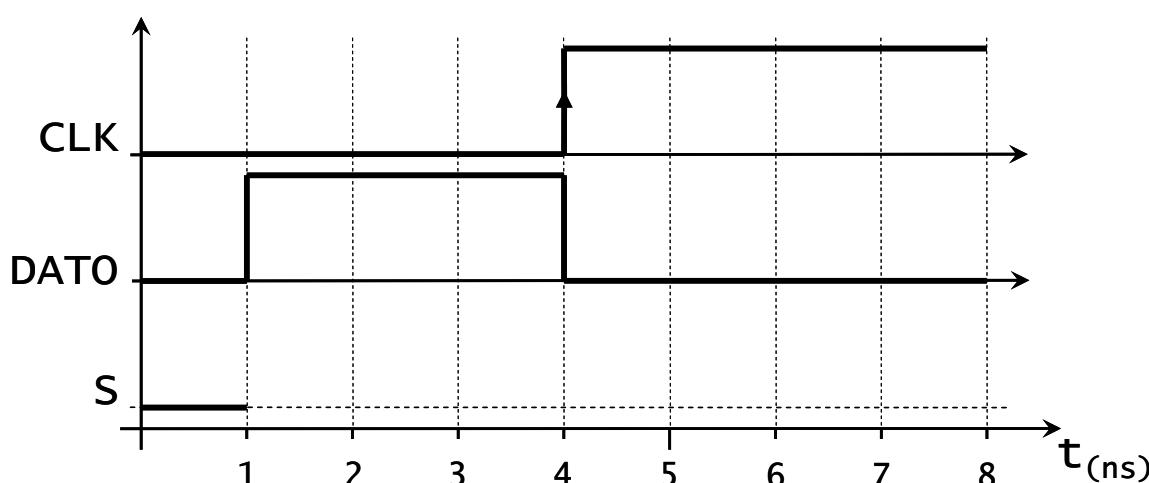
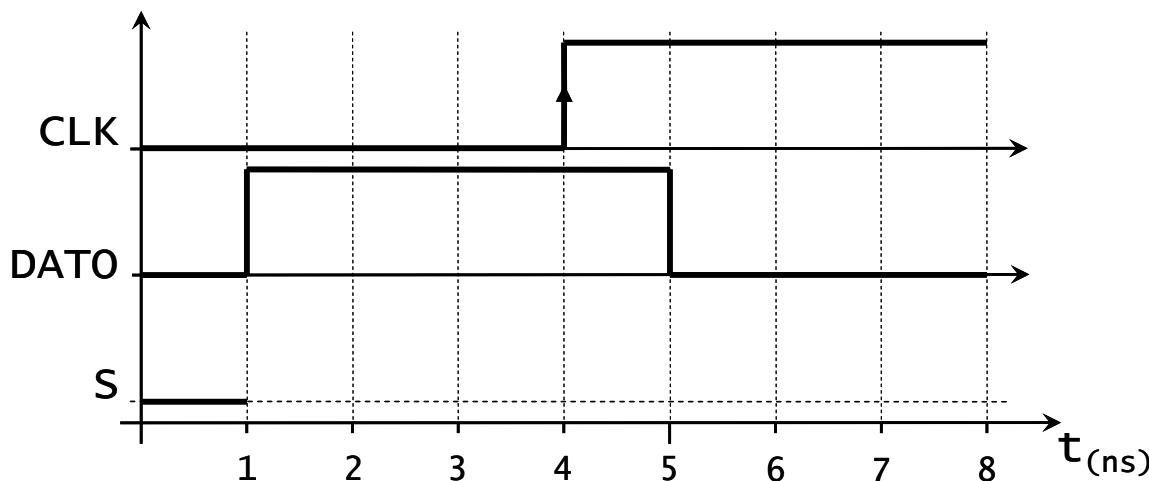
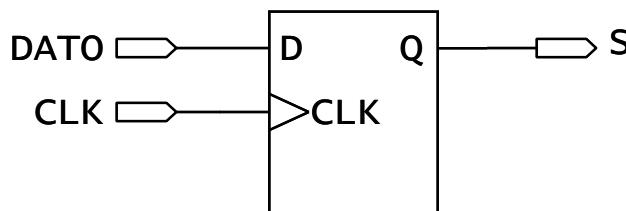
Parte	Realización	Discusión	Solución
I	6'	4'	3'
II	12'	8'	4'
III	12'	4'	3'
IV	8'	4'	2'

Ejercicios con Flip-flops

PARTE I. Ejercicios de cronogramas con tiempos de circuitos con flip-flops

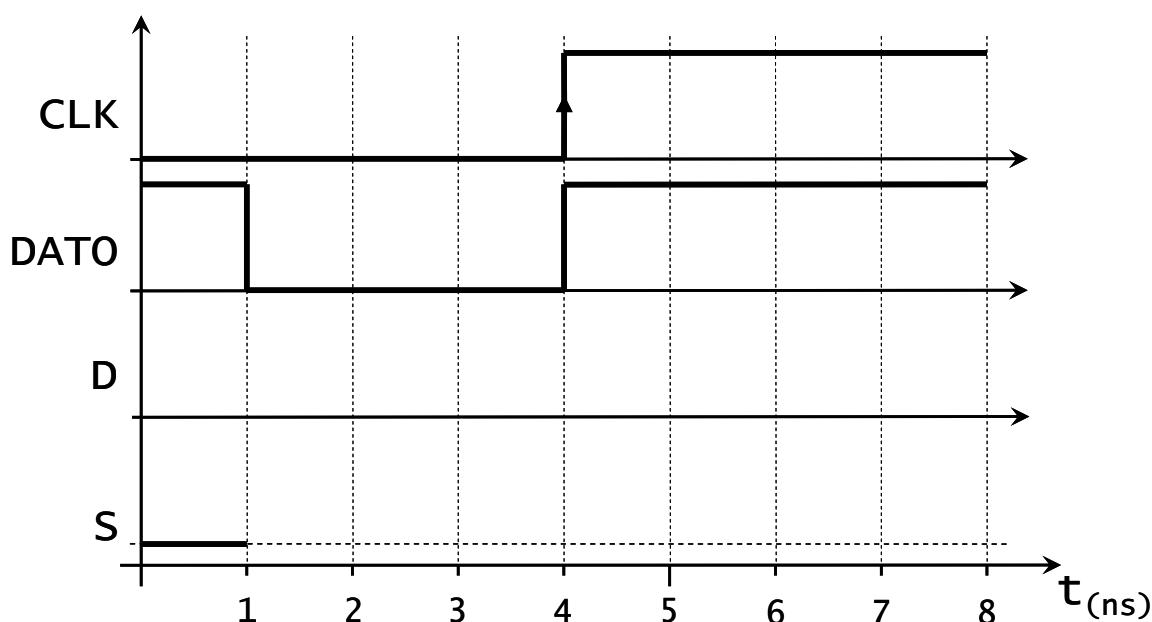
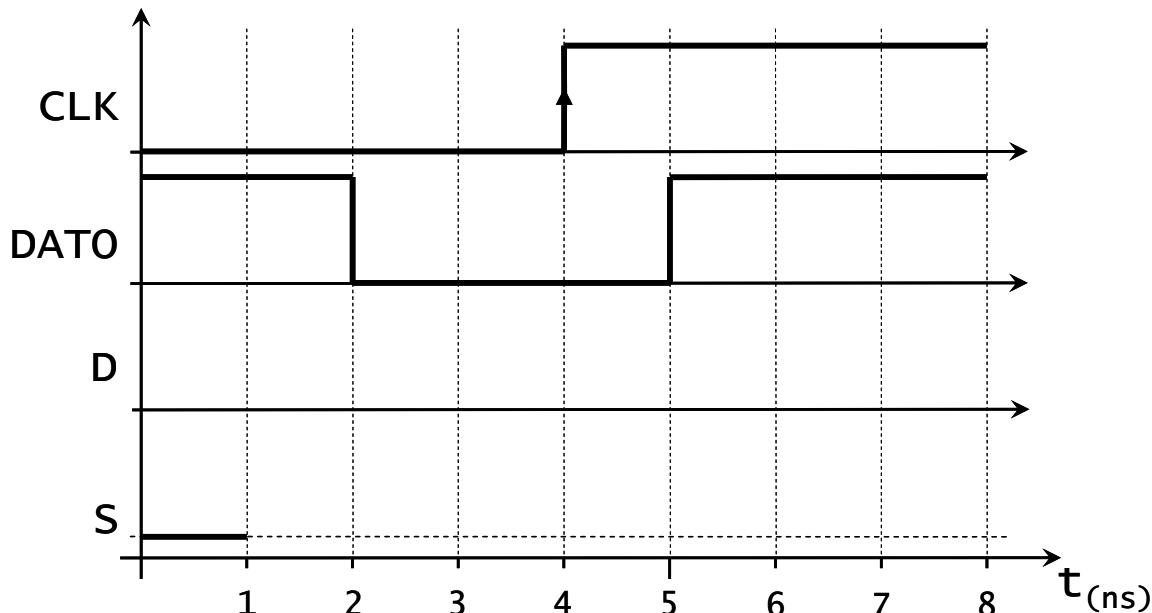
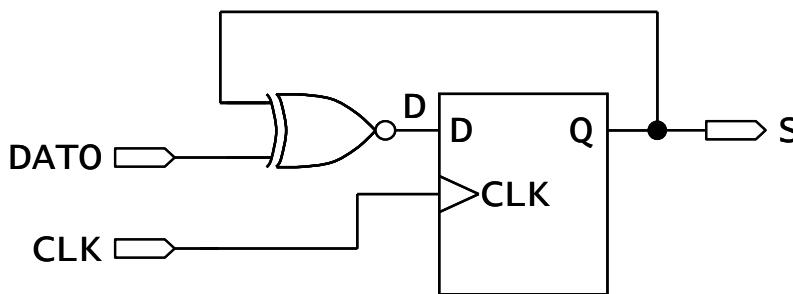
Ejercicio 1. Analice el circuito de la figura completando los siguientes cronogramas. Tenga en cuenta los tiempos característicos del flip-flop de la figura.

$$\begin{aligned}t_p &= 2 \text{ ns} \\t_{su} &= 1 \text{ ns} \\t_h &= 500 \text{ ps}\end{aligned}$$



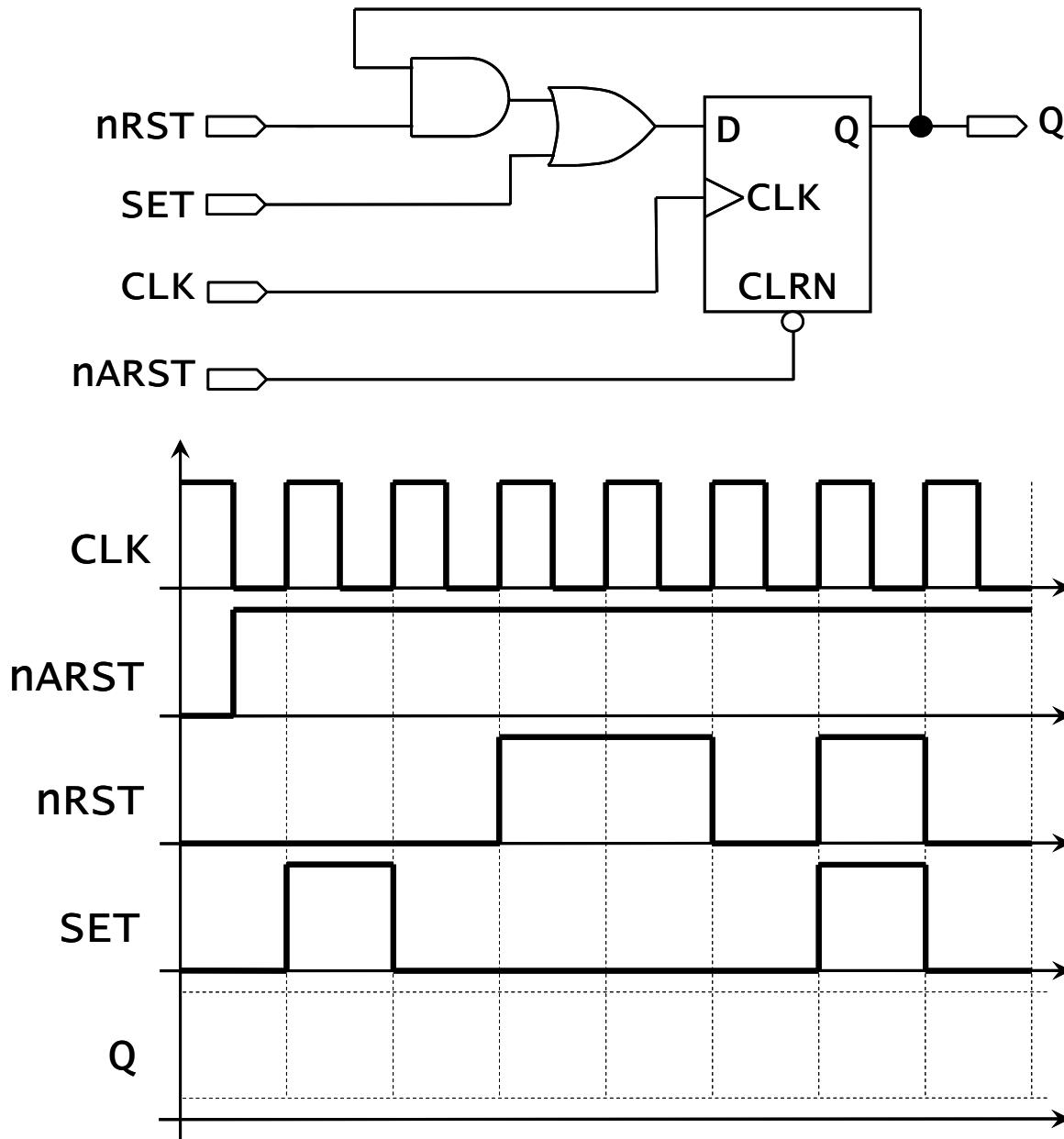
Ejercicio 2. Complete los siguientes cronogramas teniendo en cuenta los tiempos característicos del flip-flop de la figura.

$$\begin{aligned}t_p &= 2 \text{ ns} \\t_{su} &= 1 \text{ ns} \\t_h &= 500 \text{ ps} \\t_{pxor} &= 1.5 \text{ ns}\end{aligned}$$

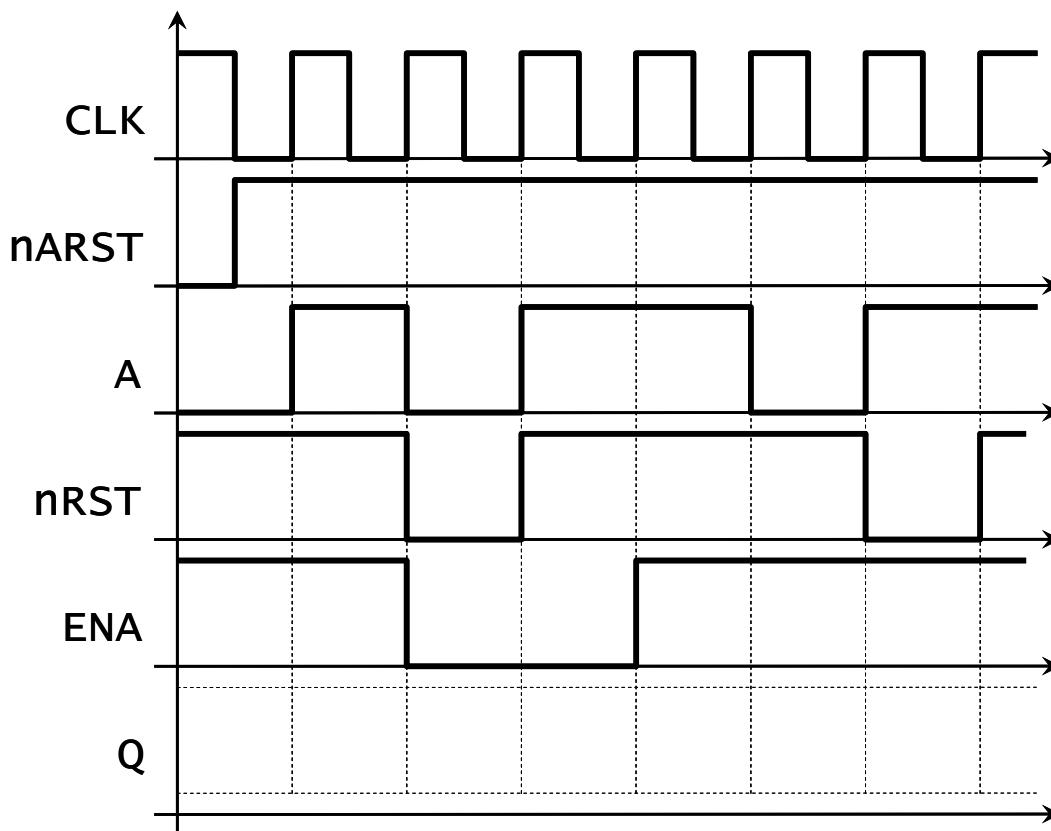
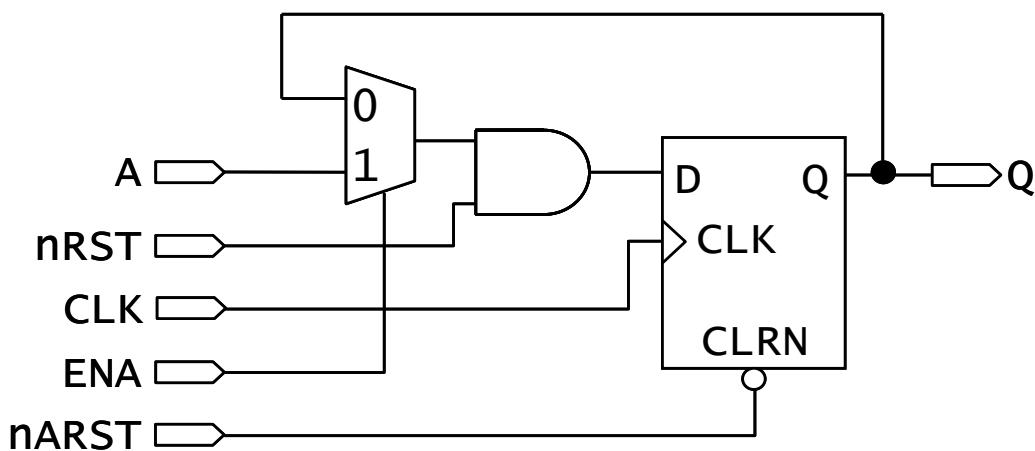


PARTE II. Ejercicios de cronogramas de circuitos con flip-flops y entradas síncronas

Ejercicio 1. Complete el cronograma correspondiente al circuito de la figura. Considere que los tiempos característicos son 0.

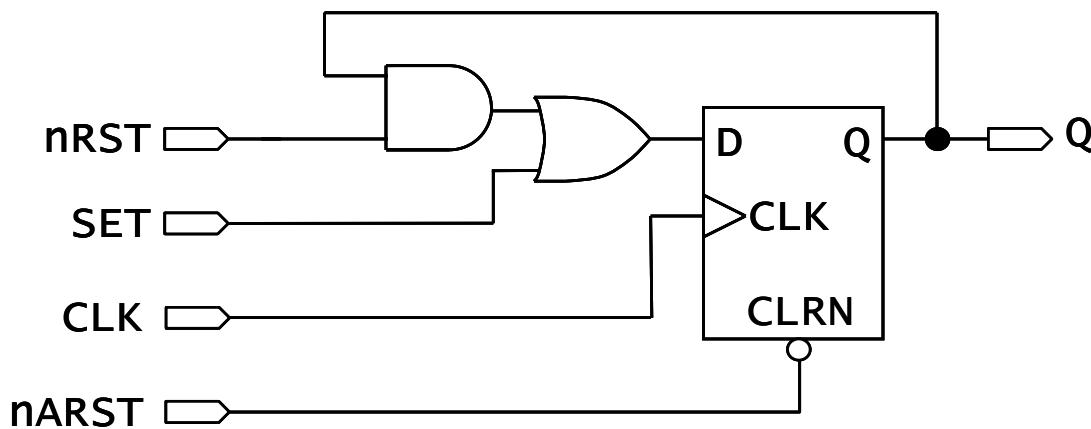


Ejercicio 2. Complete el cronograma correspondiente al circuito de la figura. Considere que los tiempos característicos son 0.

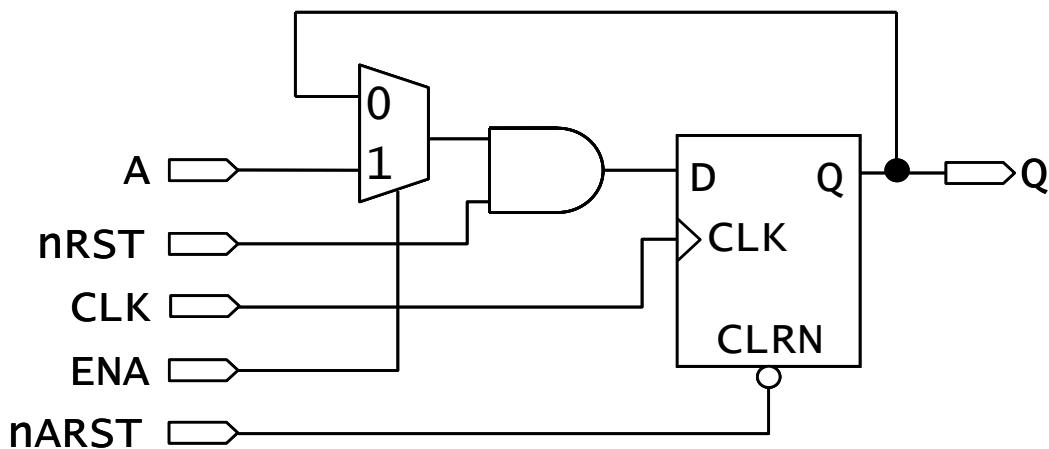


PARTE III. Ejercicios de obtención de diagramas de estados a partir de un circuito secuencial

Ejercicio 1. Realice el diagrama de estados del siguiente circuito.



Ejercicio 2. Realice el diagrama de estados del siguiente circuito.



PARTE IV. Ejercicios de diseño de flip-flops con entradas síncronas

Ejercicio 1. Diseñe el flip-flop que tiene la siguiente tabla de transición.

Entradas			Salida
PRST	ENA	A(T)	Q(T+1)
0	1	0	0
0	1	1	1
X	0	X	Q(T)
1	1	X	1

Ejercicio 2. Diseñe el flip-flop que tiene la siguiente tabla de transición.

Entradas			Salida
nRST	ENA	A(T)	Q(T+1)
0	X	X	0
1	0	X	Q(T)
1	1	0	0
1	1	1	1

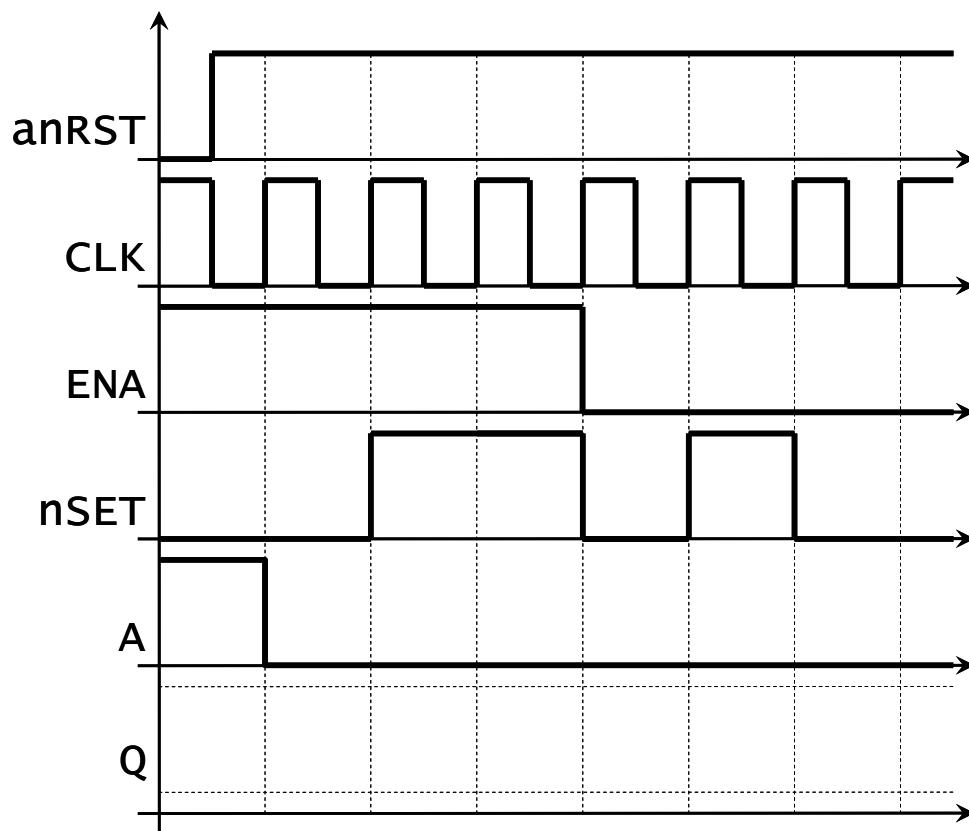
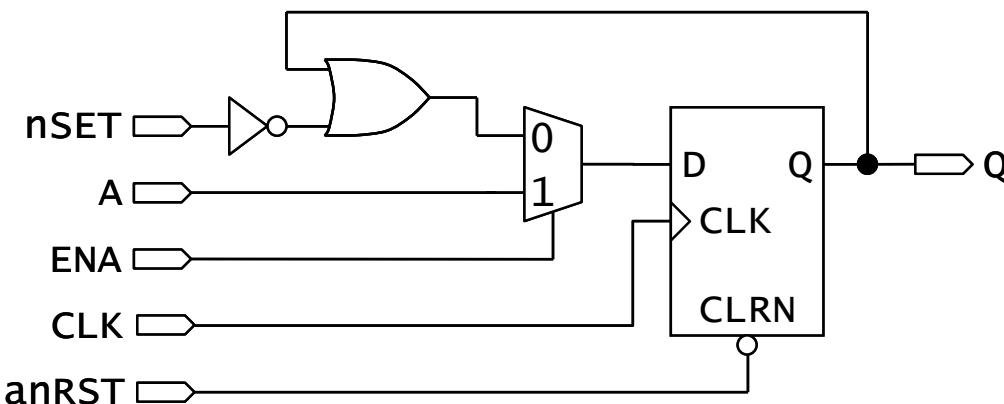
BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Test final sobre ejercicios con flip-flops.

CÓDIGO:
BT3_ATGP1

FECHA:			
NOMBRE:	APELLIDOS:		
GRUPO	CALIFICACIÓN: [entre 0 y 10 puntos]		puntos

- a) Analice el circuito de la figura rellenando el siguiente cronograma.



- b) Obtenga el diagrama de estados del circuito.

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Registros. Arquitecturas de sistemas secuenciales. Modelo de los chips configurables.

CÓDIGO:
BT3_AINP4

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Lectura Dirigida Individual	TIPO:	No presencial	DURACIÓN:	140 minutos
------------	-----------------------------	-------	---------------	-----------	-------------

CALENDARIO:	Segunda Sesión. Previo a la tercera sesión	REQUISITOS:	Flip-flops
-------------	---	-------------	------------

CRITERIO DE ÉXITO:	Compresión de los procedimientos y conceptos expuestos.
--------------------	---

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	No procede
------------------	---------	--	------------

PARTE I. Descripción de la actividad

Esta actividad persigue el propósito de que adquiera habilidades de análisis y diseño de registros, que conozca el modelo general de arquitectura de los circuitos secuenciales síncronos y, por último, que conozca el modelo completo de la estructura de las células integradas en los chips configurables para que pueda entender cómo se configuran para materializar circuitos lógicos secuenciales. La actividad se descompone en las siguientes tareas:

1. Lectura del texto que se incluye en este documento: El contenido del texto incluye todos los conocimientos que debe adquirir:
 - a. Registros: De la página 3 a la 12. Su lectura detallada debería ocupar unos 45 minutos. Al finalizar la lectura debe realizar (en unos 20 minutos, aproximadamente) los ejercicios que se le proponen.
 - b. Arquitectura de los circuitos secuenciales síncronos: De la página 17 a la 27. La lectura de esta parte del texto debería completarse en 40 minutos. Al finalizar la lectura debe realizar, también, (en uno 15 minutos, aproximadamente) los ejercicios que se le proponen.
 - c. Arquitectura de los chips configurables: De la página 30 a la 33. Esta lectura debe poder realizarla en 20 minutos.
2. Anotación de dudas y curiosidades: Debe anotar cualquier duda o curiosidad que le haya surgido en relación con los contenidos de la actividad. Durante la siguiente sesión presencial (la primera de la tercera semana del bloque temático 3), el profesor dedicará 50 minutos a resolver las cuestiones que se le planteen sobre estos temas.

PARTE II. Registros. Introducción

Los registros¹ son subsistemas cuya función consiste en el almacenamiento temporal de datos. El cronograma de la figura 1 muestra el modo de funcionamiento básico del tipo de registro de uso más común.

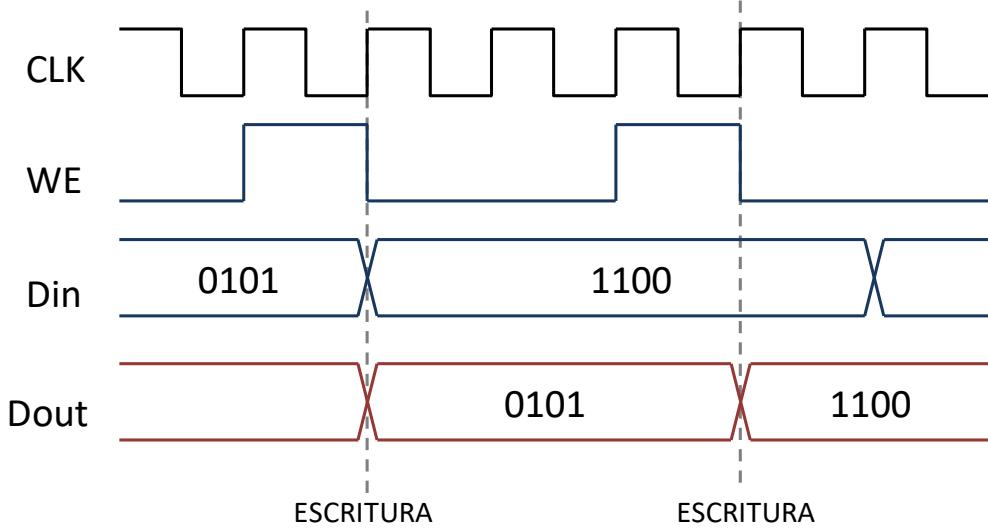


Figura 1.- Funcionamiento de un registro

Cuando se ordena una **escritura** en un registro, habilitándola mediante una entrada de control síncrona (en el cronograma, **WE**, activa a nivel alto), el dato que hay a la entrada del registro (**Din**) queda memorizado en el flanco activo de reloj y puede **leerse** en la salida (**Dout**) del registro. El dato almacenado persiste hasta la siguiente escritura del registro.

Características

Las características que distinguen a los registros de otros módulos de almacenamiento utilizados en la realización de sistemas digitales se enumeran a continuación.

- **Diseño lógico simple:** Los registros son subsistemas con una estructura simple y regular, por lo que su realización se reduce a la construcción de una célula básica, normalmente muy sencilla y de la que forma parte un **flip-flop**, cuyas réplicas (habrá tantas como bits memorice el registro) se conectan atendiendo a un patrón monótono que depende del **tipo de registro**.
- **Simplicidad y rapidez de las operaciones de almacenamiento:** En un registro se puede realizar una operación de almacenamiento síncrona (**escritura**) en cada ciclo de reloj, operación que puede controlarse mediante una interfaz compuesta únicamente por una entrada de habilitación de reloj – esto facilita enormemente la realización de los circuitos lógicos que manejan el registro.
- **Acceso directo al dato memorizado:** El dato que **guarda** un registro resulta accesible directamente, en cualquier instante de tiempo, en las salidas de los flip-flops con que está realizado. Esto quiere decir que, a diferencia de las memorias de acceso aleatorio, no resulta necesario realizar una

¹ Los registros a los que nos referimos aquí se conocen, dentro de una clasificación más amplia, como registros estáticos. Hay otro tipo de registros, los dinámicos, cuyo estudio escapa al alcance de este texto y, por tanto, todo lo que se expone en los siguientes apartados hace referencia única y exclusivamente a los registros estáticos.

operación de lectura para obtener el valor de los bits almacenados en un registro: la salida de un registro es el propio dato que almacena.

- Alto coste de recursos hardware: Los registros se realizan con **flip-flops**, que son las células con memoria cuya materialización supone una mayor complejidad hardware, lo que conlleva una pésima relación entre la capacidad de memoria del registro y el área del circuito que se precisa para realizarlo. No resultan adecuados, por tanto, para la construcción de sistemas con grandes capacidades de memoria.

Clasificación y caracterización de registros

Atendiendo a criterios funcionales, los registros se clasifican según el modo (en serie o en paralelo) en que los datos se escriben y leen en ellos. Debe entenderse que cuando se habla de escritura o lectura en paralelo se quiere decir que todos los bits del dato se escriben o leen en el mismo instante de tiempo (como en el cronograma de la figura 1), mientras que cuando se dice que la escritura o lectura es en serie, los bits del dato se escriben o leen, uno tras otro, en instantes de tiempo consecutivos, como en el cronograma de la figura 2.

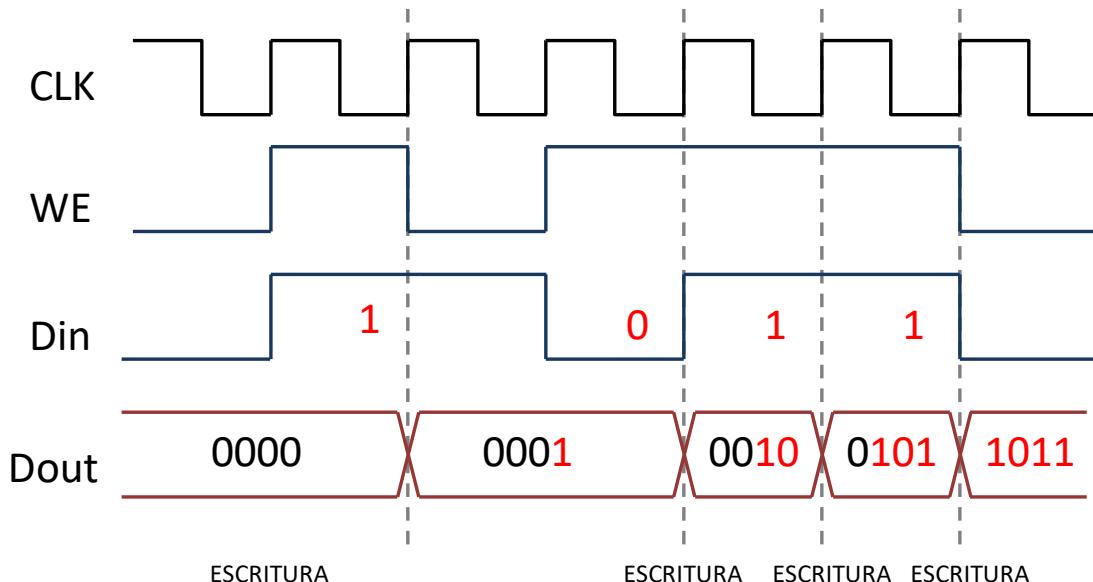


Figura 2.- Registro con escritura serie

En la figura 2 se detalla la escritura en un registro de un dato (1011), cuyos bits se ingresan en serie por la entrada (**Din**) cuando la escritura está habilitada (cuando **WE** está activa). Para escribir o leer un registro en serie se precisan tantos ciclos de reloj como bits de memoria tenga.

Atendiendo al modo de lectura y escritura de un registro, se puede hablar de:

- Registros con escritura y lectura en paralelo
- Registros con escritura y lectura en serie
- Registros con escritura en paralelo y lectura en serie
- Registros con escritura en serie y lectura en paralelo

La caracterización funcional de un registro debe incluir:

- El tipo de registro de acuerdo con la clasificación anterior
- Su capacidad de memoria
- El tipo de entradas síncronas de control, su nivel activo y su efecto
- El tipo de entradas asíncronas, su nivel activo y su efecto

De la interfaz de un registro forman parte, además de la entrada de reloj y la entrada de inicialización asíncrona, las entradas síncronas de control (si las hay), una entrada de datos, donde se ingresa la combinación de bits que se desea escribir en el registro, y una salida de datos en la que se puede leer el valor guardado en el registro.

Ejemplo: El registro cuyo funcionamiento describe el cronograma de la figura 2 tiene una interfaz de escritura en serie y de lectura en paralelo. Es un registro de 4 bits –porque su salida paralelo es de 4 bits- y tiene una entrada de habilitación (**WE**, Write enable) activa a nivel alto.

Ejemplo: El funcionamiento de los siguientes registros queda completamente determinado por las características que los definen:

- Registro con escritura y lectura en paralelo, de 8 bits, con habilitación de reloj, activa a nivel alto, y reset asíncrono, activo a nivel bajo, que inicializa el registro a 0.

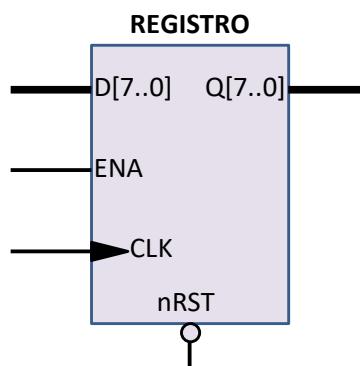


Figura 3.- Registro con escritura y lectura de datos en paralelo

- Registro con escritura en serie y lectura en paralelo, de 16 bits, con entrada de reset síncrono que pone el registro a 0 y entrada de reset asíncrono que inicializa el registro al valor 7FOCh.

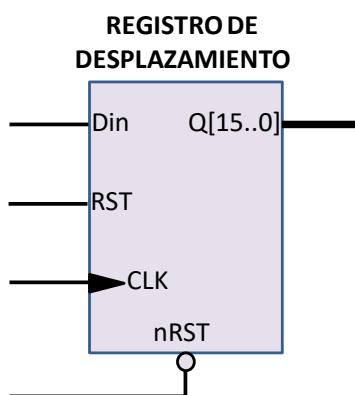


Figura 4.- Registro con escritura en serie y lectura en paralelo

PARTE III. Diseño de registros

Registros con escritura y lectura de datos en paralelo

De entre los cuatro tipos de registros, el más importante y de uso más común es el que dispone de una interfaz de escritura y lectura paralelo. De hecho, cuando se emplea el término **registro** sin especificar expresamente el modo de lectura o escritura (o sin que se pueda deducir otra cosa del contexto en que se emplee esta denominación), se sobreentiende que es una referencia a un registro de este tipo: con lectura y escritura en paralelo.

El cronograma de la figura 1 ilustra el funcionamiento de este tipo de registros. Su interpretación es simple: guarda el valor de un dato, que se le entrega en paralelo, cuando se ordena una escritura y lo mantiene en memoria hasta que se escribe un nuevo dato. Su interfaz de salida permite leer en paralelo todos los bits del dato almacenado.

La célula básica para el diseño de un registro como éste es un flip-flop con habilitación de reloj (Figura 5a). Si se desea realizar un registro de N bits, basta con disponer de N flip-flops que comparten la conexión de la señal de reloj y de habilitación, tal y como se muestra en el ejemplo de la figura 5b, que representa la estructura de un registro de 4 bits; en la figura 5c se muestra su símbolo.

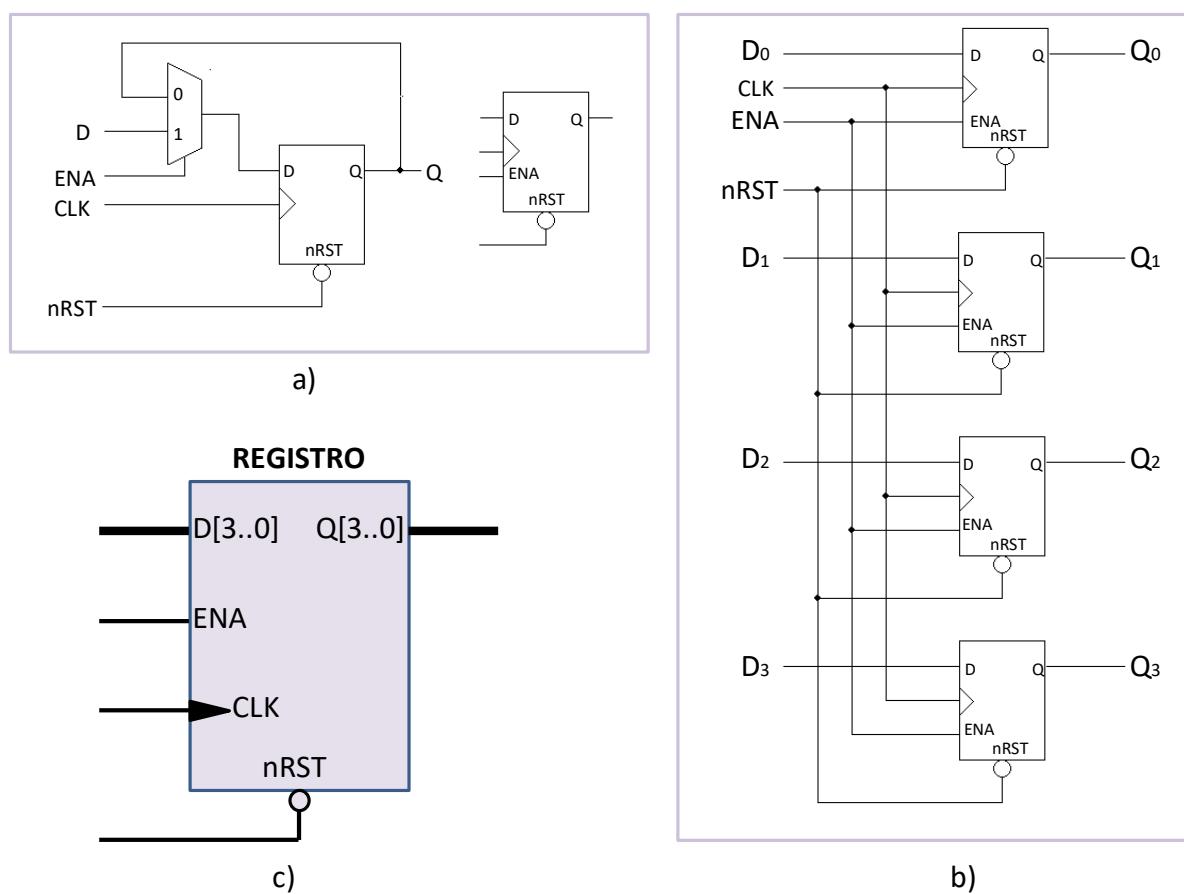


Figura 3.- Registro paralelo-paralelo

Realizando ligeras modificaciones sobre la célula básica, se pueden añadir fácilmente funciones síncronas adicionales. En la figura 4 se muestra la realización de un registro con reset síncrono.

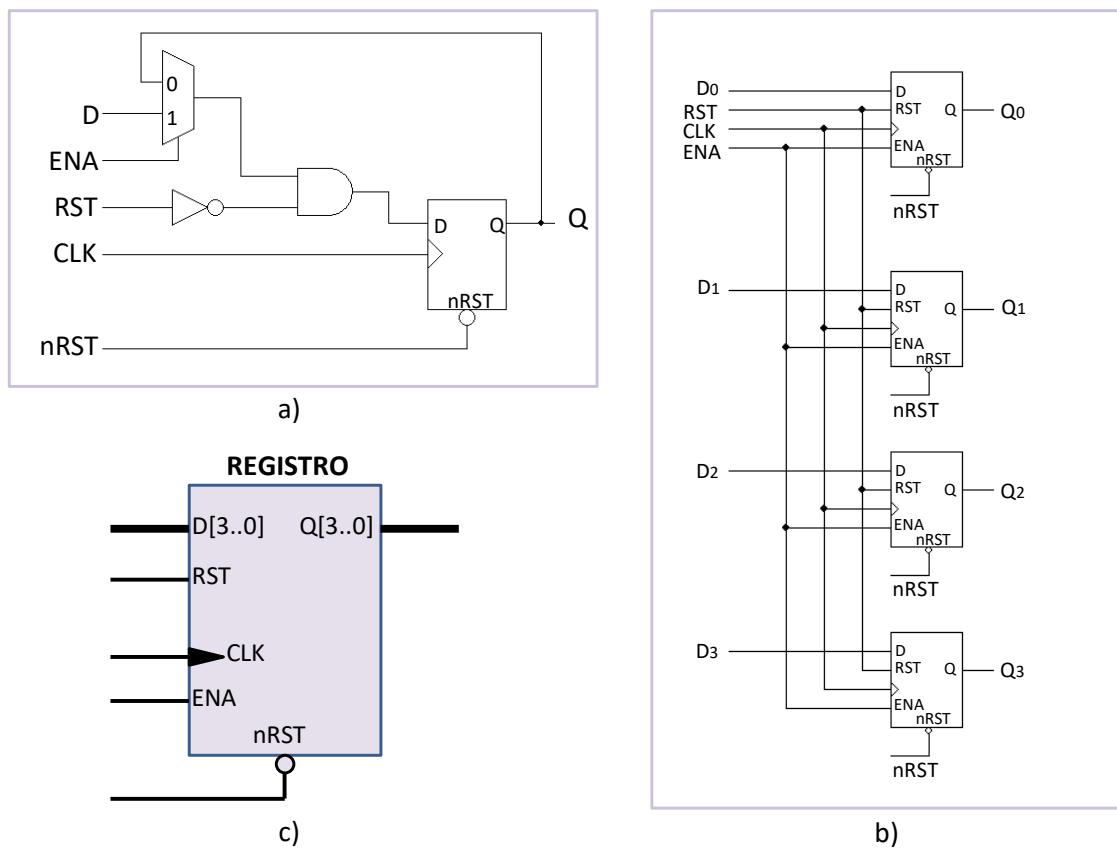


Figura 4.- Registro con reset síncrono

Un ejemplo singular, pero muy útil, de registro con interfaces paralelas es el que se muestra en la figura 5. Se trata de un registro sin entradas de control síncronas. Su única entrada es el dato que se escribe y su salida es, siempre, el dato de entrada en el instante de tiempo anterior, es decir: su entrada es el valor de un dato en el instante de tiempo **T** y su salida el valor de ese mismo dato en **T-1**.

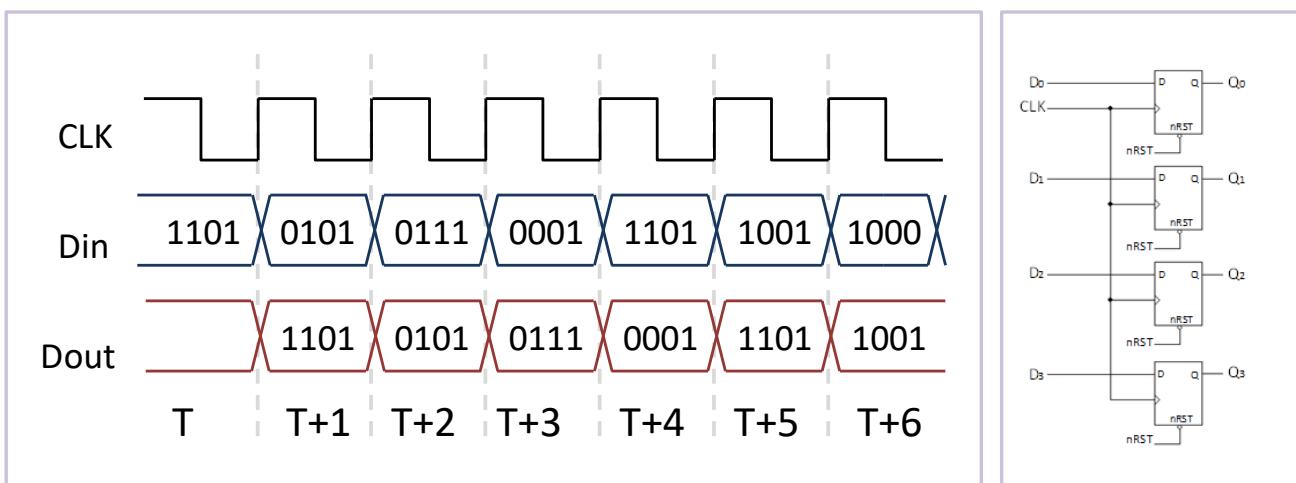


Figura 5.- Célula de retardo

Este registro funciona como una célula de retardo, la salida es el valor anterior al presente en la entrada. Si se encadenan en serie varios de estos registros, se realiza un sistema que dispone de un conjunto de **muestras** consecutivas de una señal de entrada (Figura 6). Este tipo de **muestreo** se utiliza en muchos algoritmos de procesamiento digital de señal.

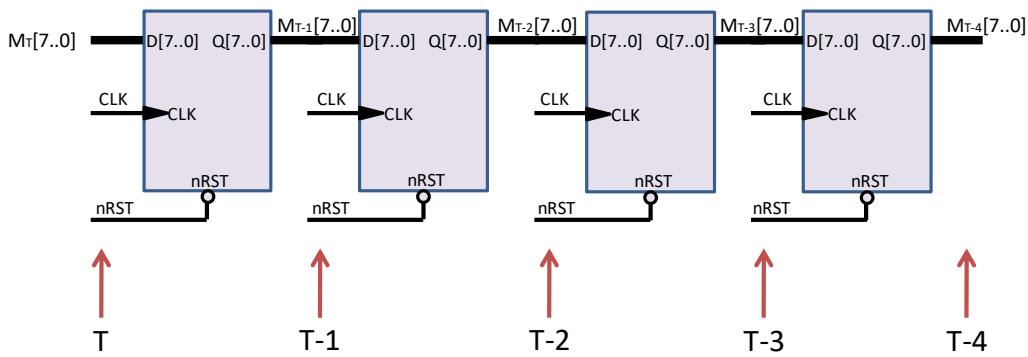


Figura 6.- Muestreo con registros

Registros de desplazamiento

Los otros tres tipos de registros comparten la denominación común de **registros de desplazamiento**. Este término, sin otra determinación, hace alusión a los registros con lectura y escritura serie. Para identificar los otros dos tipos de registros se añade la especificación de la operación que se realiza en paralelo: **registros de desplazamiento con lectura de datos en paralelo** se usa para identificar a los que tienen escritura serie y lectura paralelo, y **registros de desplazamiento con escritura de datos en paralelo** para aquellos en los que se escribe en paralelo y se lee en serie.

Los registros de desplazamiento con escritura de datos en serie tienen la estructura que se muestra en la figura 7a.

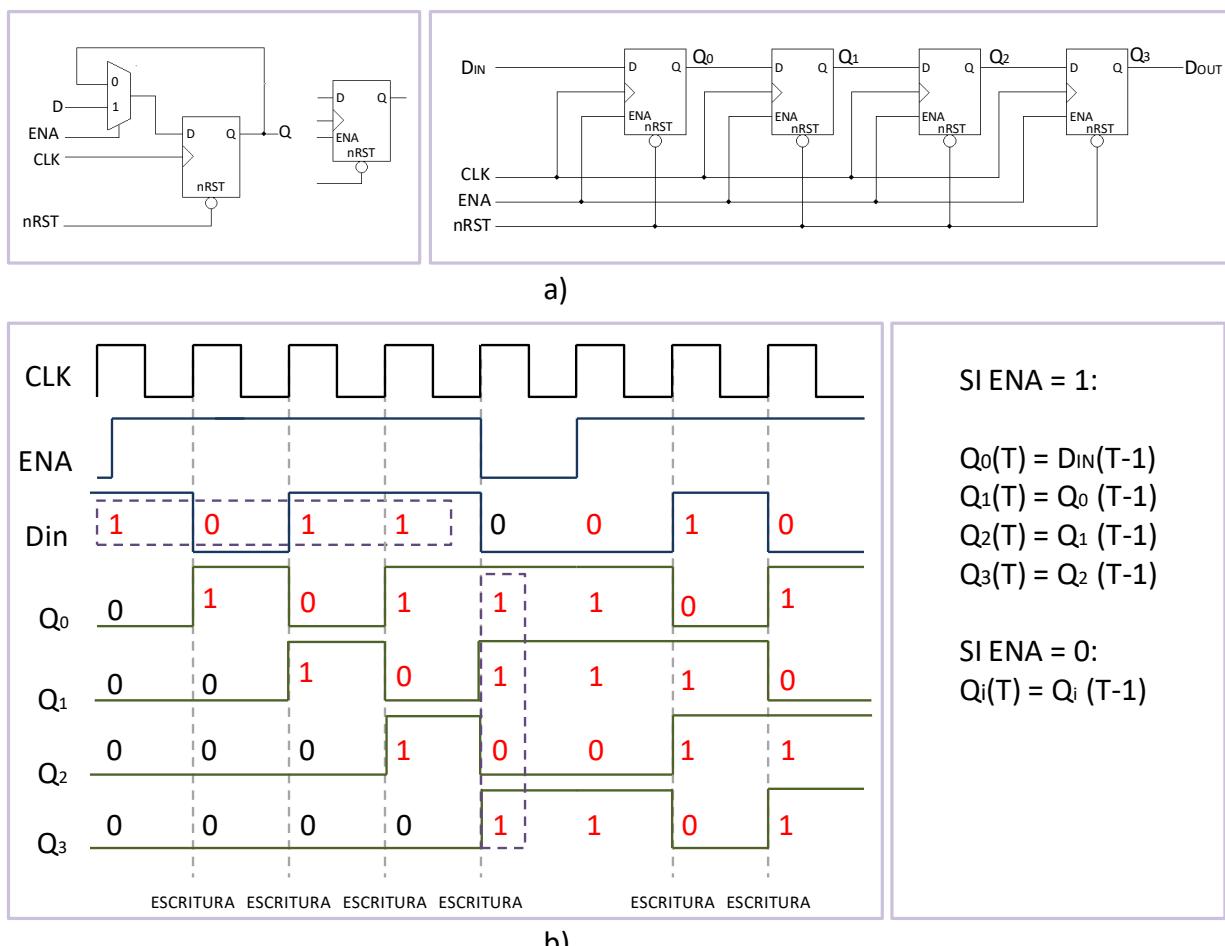


Figura 7.- Registro de desplazamiento

Tal y como muestra el cronograma de la figura 7 b, en cada flanco activo de reloj en que está habilitada la escritura, el primer flip-flop (el primero por la izquierda) **captura** el bit de entrada y los demás memorizan el bit almacenado, en el instante de tiempo anterior, en el flip-flop que tienen a su izquierda –aquel cuya salida, **Q**, está conectada a su entrada **D**.

Observando la combinación almacenada por el registro en un instante de tiempo, **T**, en el siguiente instante, **T+1**, la combinación memorizada es la anterior, pero desplazada una posición a la derecha, de modo que se pierde el bit de la derecha de la combinación en el instante **T** y se ingresa por la izquierda el bit de entrada del primer flip-flop (**Din** en el circuito de la figura 7); es a consecuencia de este efecto por lo que se los denomina registros de desplazamiento.

La escritura de un dato de N bits requiere disponer de un registro de desplazamiento con N flip-flops. El dato tarda en escribirse N ciclos de reloj si la habilitación de escritura está activa en todos ellos. En los registros de desplazamiento con lectura de datos en paralelo, el dato escrito puede leerse al cabo de esos N ciclos, tal y como se muestra en el ejemplo del cronograma de la figura 7b:

- El dato 1011 tarda en escribirse cuatro ciclos de reloj con la habilitación de escritura (**ENA**) activada.
- Al cabo de esos cuatro ciclos de reloj el dato está disponible en la salida: el primer bit en ser ingresado puede leerse en Q_3 , el último en Q_0 .

La principal utilidad de estos registros de desplazamiento con lectura paralelo es la conversión de un dato, que llega en serie por una línea -este es el modo en que se suele transmitir datos, en serie-, a formato paralelo, que es una disposición mucho más adecuada para procesarlo dentro de un sistema digital.

Si se desea utilizar un circuito como el de la figura 7a como un registro de desplazamiento con lectura serie, debe conectarse la salida del circuito a la del último flip-flop (el de la derecha). En este caso el dato escrito comienza a leerse en serie cuando hayan transcurrido tantos ciclos de reloj como flip-flops tenga el registro. Los bits se leen en el mismo orden en que se escriben, con un retardo que depende, como ya se ha dicho, del número de flip-flops del registro. El cronograma de la figura 8 ilustra el funcionamiento del circuito de la figura 7b como registro con escritura y lectura en serie.

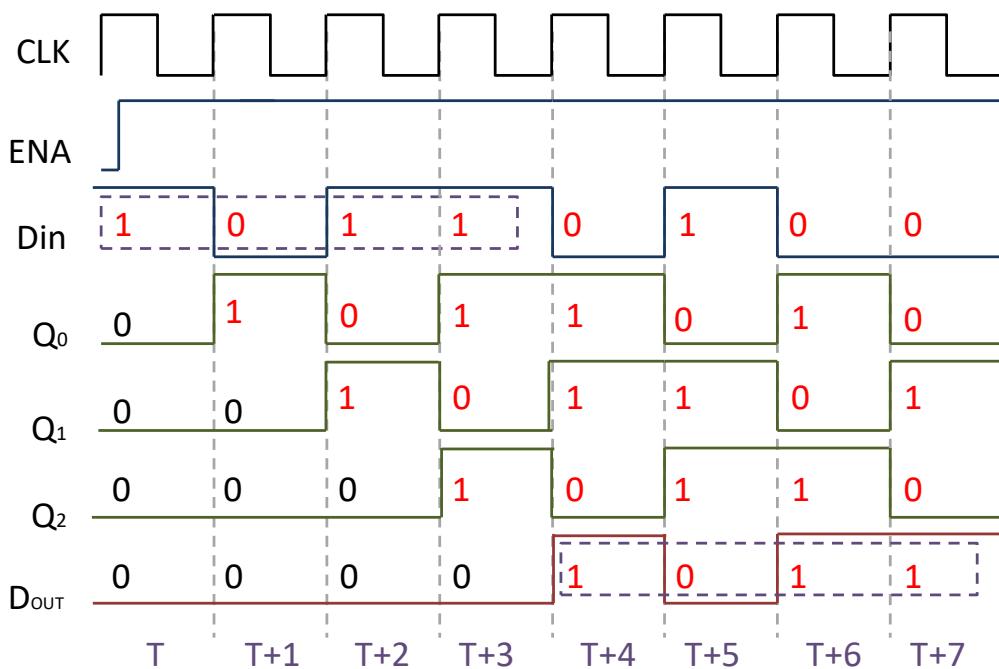


Figura 8.- Registro de desplazamiento con lectura serie

Los registros de desplazamiento con lectura serie sirven para retardar señales y para algunas aplicaciones específicas cuya descripción escapa al alcance de este texto. Si los flip-flops con que se construye carecen de entrada de habilitación de reloj, un registro de desplazamiento se convierte en un caso particular del circuito de la figura 6: un circuito que muestra el valor de un bit durante una serie de instantes consecutivos de tiempo –también se puede interpretar el circuito de la figura 6 como un conjunto de registros de desplazamiento en paralelo.

El tercer tipo de registro de desplazamiento es el que dispone de escritura de datos en paralelo. En la figura 9 se muestra la célula básica de este tipo de registros y su patrón de conexión para un ejemplo en el que se construye un registro de 4 bits.

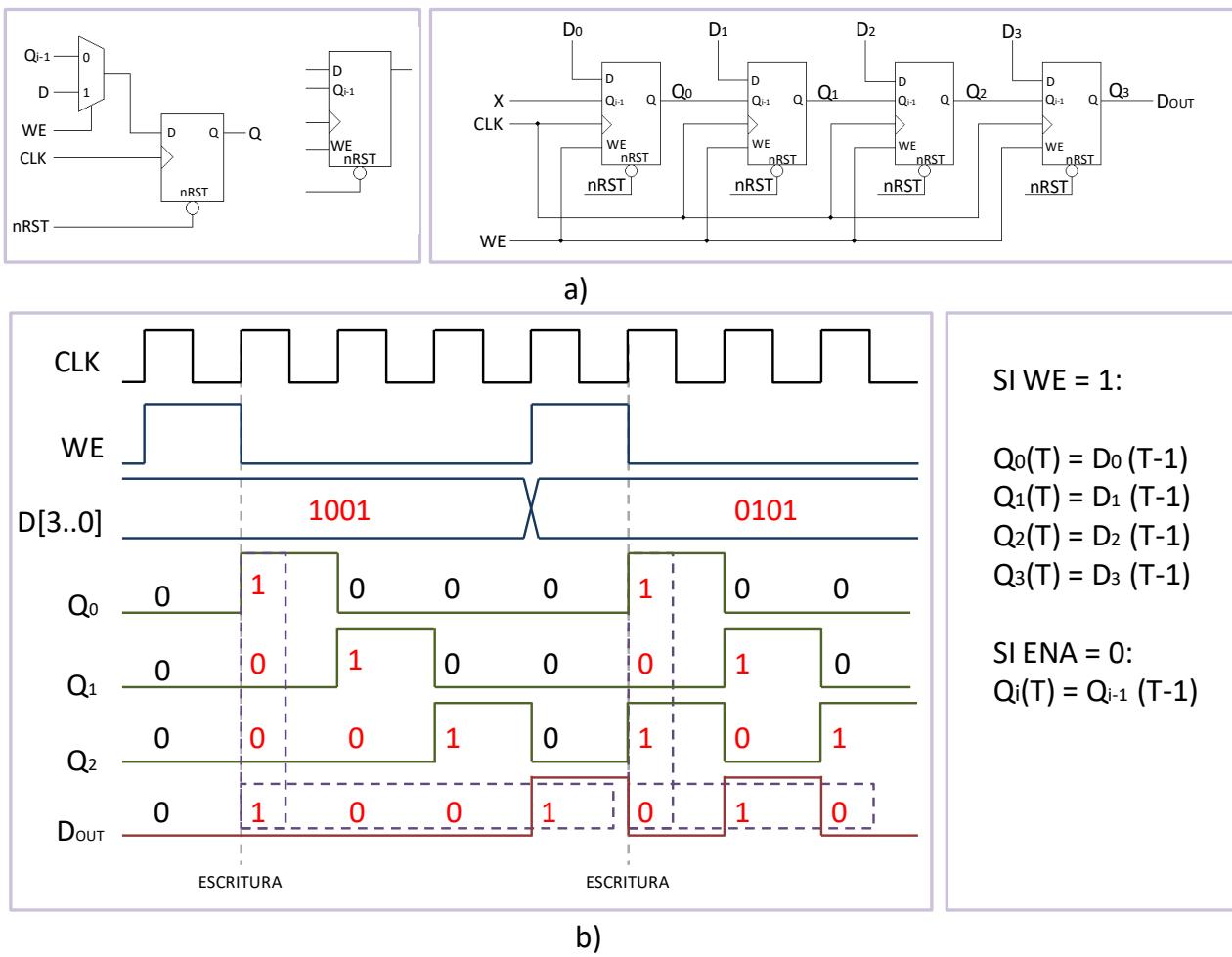


Figura 9.- Registro de desplazamiento con escritura de datos en paralelo

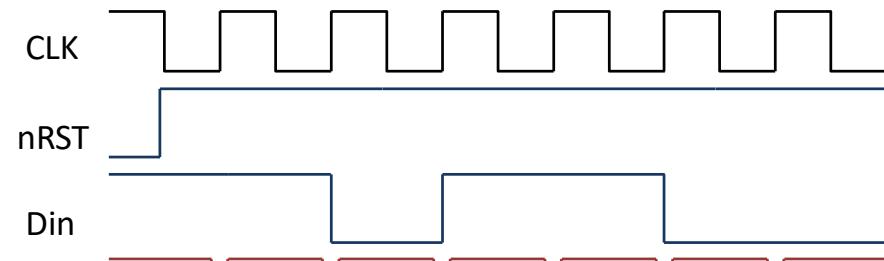
Cuando la entrada de habilitación de escritura (**WE**) está activa, los flip-flops memorizan el dato de entrada; en caso contrario, el dato se desplaza hacia la derecha, de modo que puede leerse, en serie, por la salida del último flip-flop durante los siguientes cuatro ciclos de reloj.

Este tipo de registro resulta útil para **serializar** datos en formato paralelo –para transmitirlos, por ejemplo, por una línea serie.

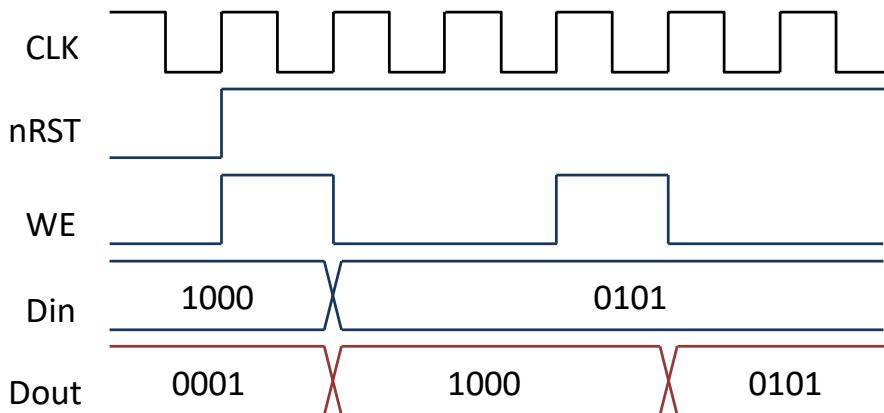
PARTE IV. Ejercicios sobre registros

Ejercicio 1.- Identifique, para cada uno de los siguientes tipos de registro, el cronograma que representa su funcionamiento (a, b, c, d, o ninguno).

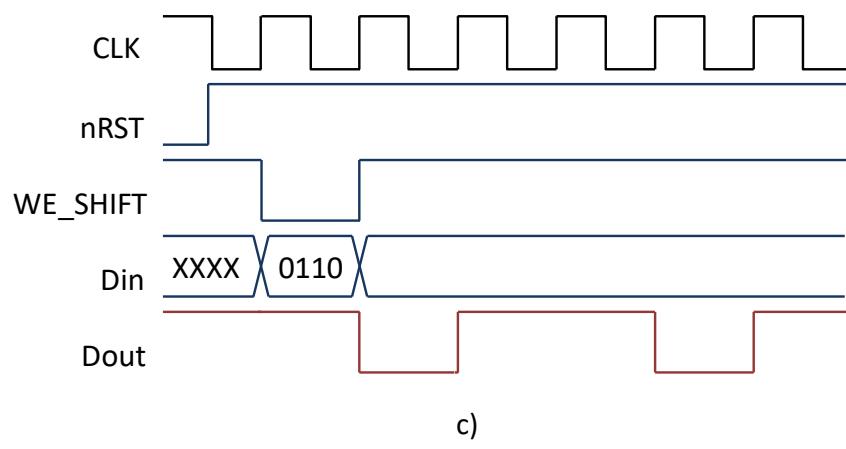
1. Registro de desplazamiento de 3 bits con lectura paralelo y con reset asíncrono, activo a nivel bajo, que inicializa el registro a 000.
2. Registro de desplazamiento de 2 bits y con reset asíncrono, activo a nivel bajo, que inicializa el registro a 0000.
3. Registro de desplazamiento de 4 bits con escritura paralelo, reset asíncrono activo a nivel bajo, que inicializa el registro a 0000, y entrada síncrona de control de carga y desplazamiento (a nivel bajo ordena la carga y a nivel alto el desplazamiento).
4. Registro de desplazamiento de 4 bits con escritura paralelo, reset asíncrono activo a nivel bajo, que inicializa el registro a 1111, y entrada síncrona de control de carga y desplazamiento (a nivel bajo ordena la carga y a nivel alto el desplazamiento).
5. Registro paralelo-paralelo de 4 bits, con reset asíncrono activo a nivel bajo que inicializa el registro a 0001, y entrada de habilitación de reloj activa a nivel alto.



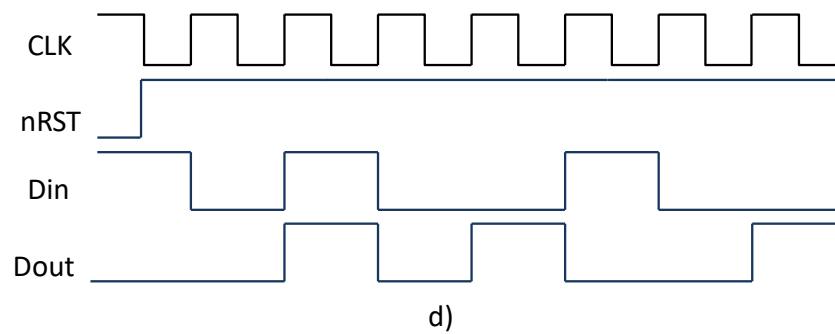
a)



b)



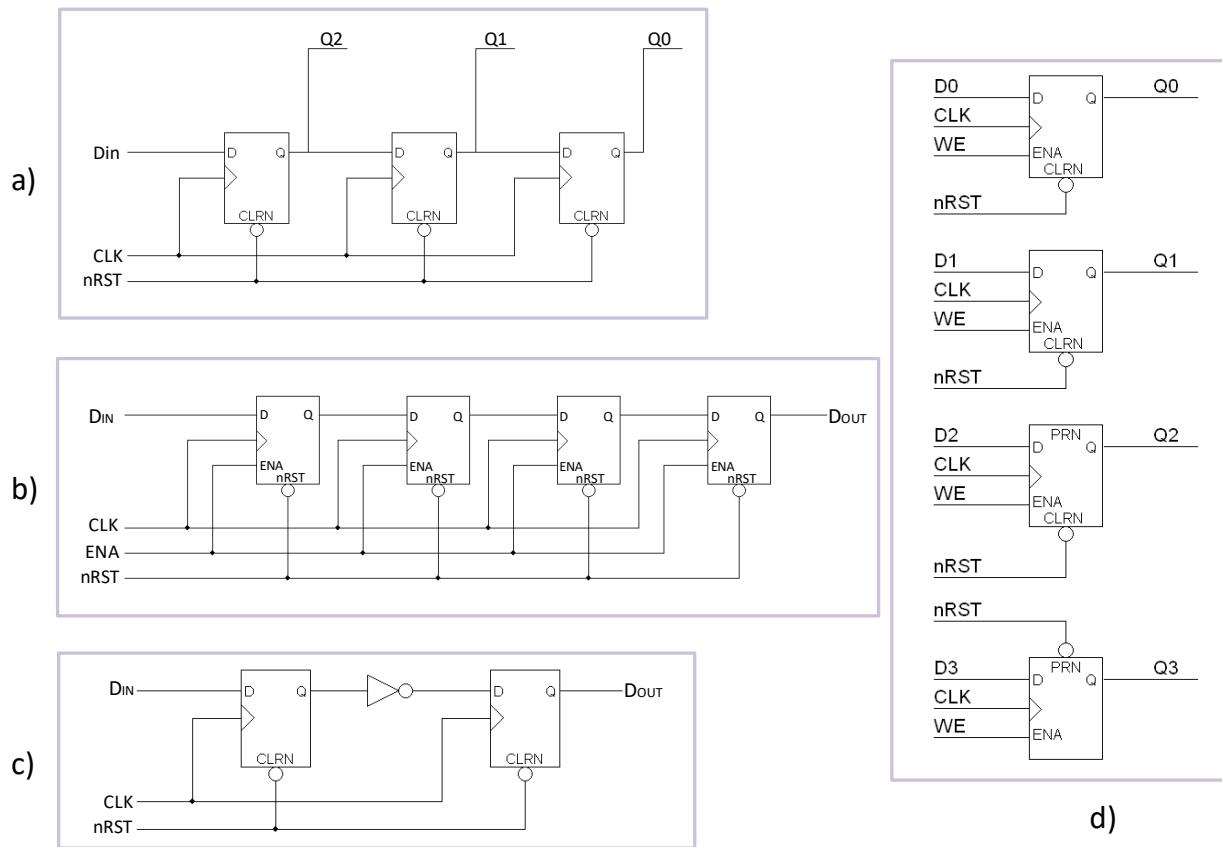
c)



d)

Ejercicio 2.- Identifique el circuito (a, b, c, o d) correspondiente a cada una de las siguientes definiciones.

- 1) Registro de desplazamiento de 4 bits con reset asíncrono activo a nivel bajo, que inicializa el registro a 0000, y entrada síncrona de habilitación de reloj activa a nivel alto.
- 2) Registro de desplazamiento de 3 bits con lectura paralelo y con reset asíncrono, activo a nivel bajo, que inicializa el registro a 000.
- 3) No es un registro.
- 4) Registro paralelo-paralelo de 4 bits, con reset asíncrono activo a nivel bajo que inicializa el registro a 1000, y entrada de habilitación de reloj activa a nivel alto.



Solución

Ejercicio 1

1. Registro de desplazamiento de 3 bits con lectura paralelo y con reset asíncrono, activo a nivel bajo, que inicializa el registro a 000. (a)
2. Registro de desplazamiento de 2 bits y con reset asíncrono, activo a nivel bajo, que inicializa el registro a 0000. (d)
3. Registro de desplazamiento de 4 bits con escritura paralelo, reset asíncrono activo a nivel bajo, que inicializa el registro a 0000, y entrada síncrona de control de carga y desplazamiento (a nivel bajo ordena la carga y a nivel alto el desplazamiento). (ninguno)
4. Registro de desplazamiento de 4 bits con escritura paralelo, reset asíncrono activo a nivel bajo, que inicializa el registro a 1111, y entrada síncrona de control de carga y desplazamiento (a nivel bajo ordena la carga y a nivel alto el desplazamiento). (c)
5. Registro paralelo-paralelo de 4 bits, con reset asíncrono activo a nivel bajo que inicializa el registro a 0001, y entrada de habilitación de reloj activa a nivel alto. (b)

Ejercicio 2

- 1) Registro de desplazamiento de 4 bits con reset asíncrono activo a nivel bajo, que inicializa el registro a 0000, y entrada síncrona de habilitación de reloj activa a nivel alto. (b)
- 2) Registro de desplazamiento de 3 bits con lectura paralelo y con reset asíncrono, activo a nivel bajo, que inicializa el registro a 000. (a)
- 3) No es un registro. (c)
- 4) Registro paralelo-paralelo de 4 bits, con reset asíncrono activo a nivel bajo que inicializa el registro a 1000, y entrada de habilitación de reloj activa a nivel alto. (d)

PARTE V. Arquitectura de circuitos secuenciales

Introducción

En los siguientes apartados se va a presentar el modelo de arquitectura genérico de los sistemas secuenciales síncronos, un diagrama de bloques que permite describir la estructura de cualquiera de estos sistemas. El modelo resulta útil para distinguir los distintos bloques que forman parte de un sistema secuencial síncrono y para definir, en base al análisis de la función e interfaz de dichos bloques, un método sistemático para el diseño de este tipo de sistemas.

El procedimiento que se va a seguir para obtener el modelo consiste en analizar una serie de circuitos que representan los distintos tipos de estructura que pueden tener los sistemas secuenciales para, a partir de ellos, obtener un modelo que resulte válido para todos. Durante este análisis, cuando resulte relevante, se destacarán aspectos estructurales que impactan en características funcionales de los sistemas.

El modelo que se va a obtener se basa en la distinción de dos funciones intrínsecas a cualquier sistema secuencial:

- el **control del estado de memoria**, realizado por recursos que determinan cómo evoluciona el estado del sistema en base a la secuenciación de los valores de las entradas a lo largo del tiempo y
- la **generación de las salidas** del sistema en base a las entradas y al estado de memoria del sistema.

Primero se obtendrá el modelo de arquitectura del **control de la memoria de estado**, después el modelo de **generación de las señales de salida** y, por último, se fusionarán ambos para obtener el modelo completo de un sistema secuencial.

Modelo de control de la memoria de estado

Como ya sabe, la característica esencial de los sistemas secuenciales es que disponen de memoria. La memoria de un sistema secuencial almacena, en un determinado instante de tiempo, una combinación de bits que depende de las combinaciones de entrada en instantes de tiempo anteriores.

Vamos a tomar como primer ejemplo de sistema secuencial un registro sin entradas de **control** síncronas (sus únicas entradas síncronas son el dato a registrar), como el de la figura 5 (que, por comodidad repetimos aquí).

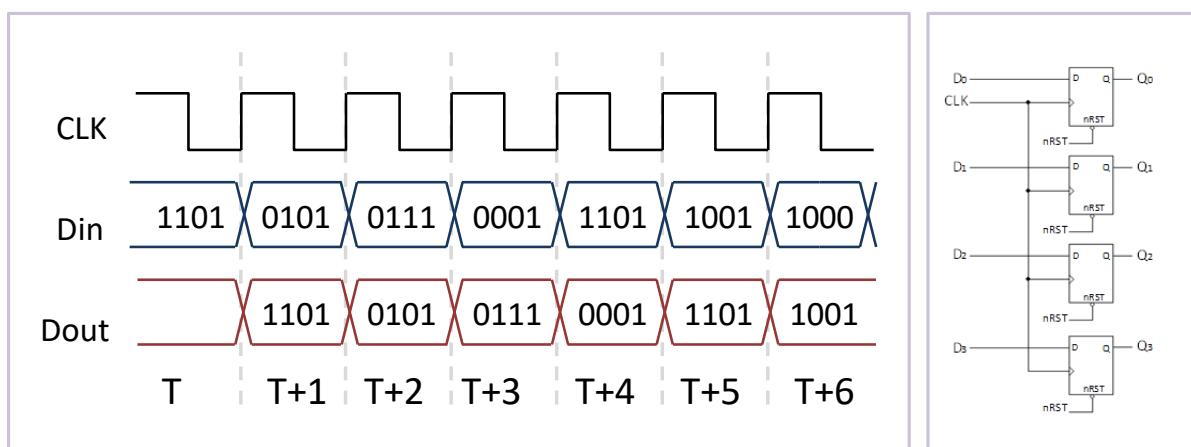


Figura 13

Cada célula de este registro está compuesta por un flip-flop cuya entrada es un bit del dato que se registra; por tanto, la ecuación del nodo de entrada, D_i , en cualquier instante de tiempo, T , de los flip-flop del registro es simplemente:

$$D_{i(\text{flip-flop})}(T) = D_i(T), \quad i = 1, 2, 3, 4$$

Cuando llega el flanco de reloj que marca el paso del instante T al $T+1$, el nivel lógico de D en el instante T es el que determina el estado (Q) del flip-flop en $T+1$. Basándonos en este modo de funcionamiento, que es generalizable a cualquier circuito secuencial, podemos decir que la combinación de bits en las entradas de datos (D_i) de los flip-flops es el **estado futuro** del circuito secuencial. El **estado actual** del circuito es el que memorizan los flip-flops (el que aparece codificado en sus salidas, Q_s) en ese instante.

En el caso del registro de la figura 13, puesto que las ecuaciones que definen el **estado futuro** (en $T+1$) sólo dependen del valor de las entradas en el instante actual (T), podemos deducir que el estado, en cualquier instante de tiempo, sólo depende de las entradas en el instante de tiempo anterior –una propiedad que ya conocíamos de este registro, pero que ahora hemos deducido analizando su estructura lógica.

Consideremos ahora el caso de un registro con habilitación de reloj como el de la figura 4 (que nuevamente replicamos en la figura 14). Tal y como se muestra en dicha figura, las ecuaciones de entrada de los flip-flops, en un instante de tiempo T , dependen en este caso, además del valor de las entradas en T , del propio estado del flip-flop (de su estado **actual**). Por tanto, el estado futuro (en $T+1$) depende del valor de las entradas y del estado en T . Como el estado en T habrá dependido a su vez de las entradas y el estado en $T-1$, esta estructura de las ecuaciones permite que la memoria de este circuito en cualquier instante de tiempo dependa de los valores de las entradas en un número indeterminado de instantes de tiempo anteriores –y no sólo del instante de tiempo anterior como en el caso del circuito de la figura 13.

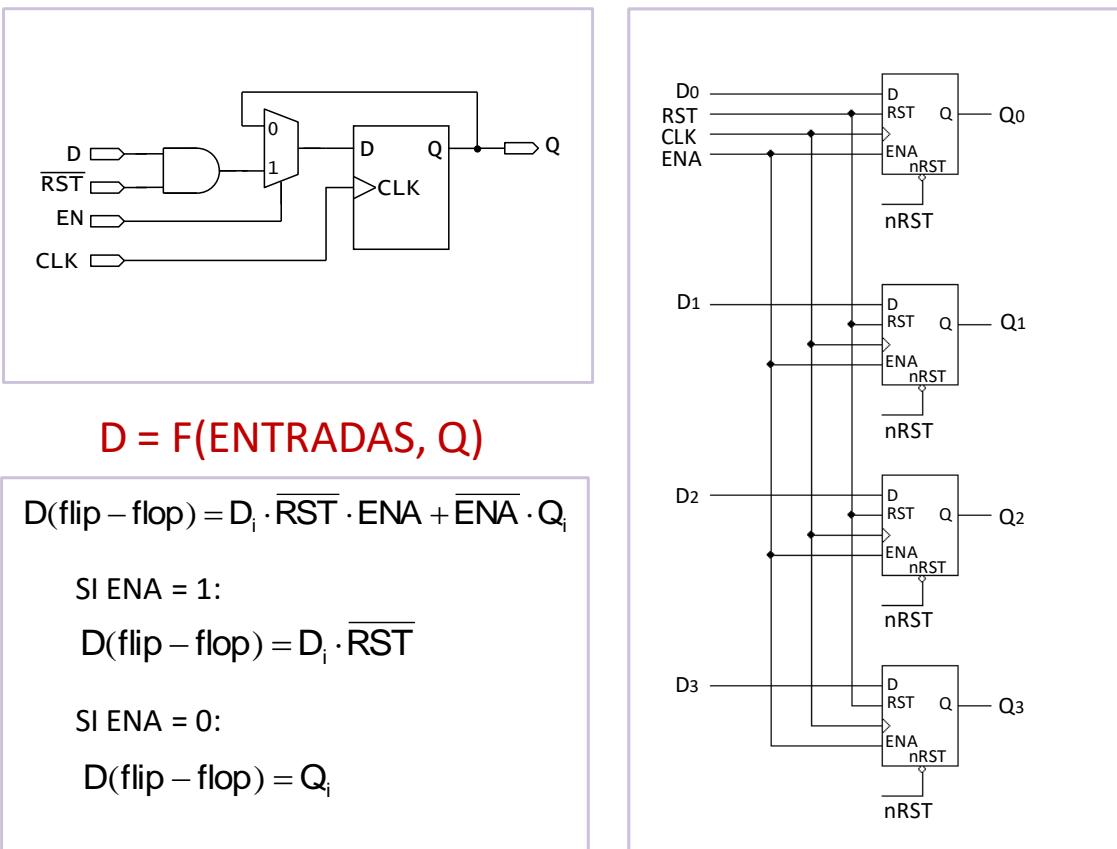


Figura 14

En resumen, del hecho de que el estado futuro (en **T+1**) del circuito de la figura 11 dependa del valor **actual** de las entradas y del estado, se deriva que el **estado actual** en cualquier instante de tiempo pueda depender del estado de las entradas en un número indeterminado de instantes anteriores. La indeterminación del número anterior de instantes de tiempo de que depende el estado de memoria se ilustra en el cronograma de la figura 15.

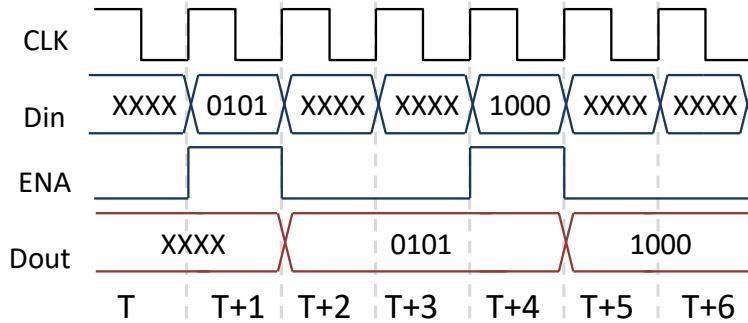


Figura 15.- Cronograma de un registro con habilitación de reloj

Si nos fijamos en el flanco de subida de la señal de reloj que marca el paso del instante **T+1** al **T+2**, podemos observar que el estado (D_{out}) en **T+2** depende exclusivamente del valor de las entradas en **T+1** (esto se debe a que en **T+1** la entrada de habilitación está activa y, tal y como se indica en la figura 14, en este caso se anula la dependencia del estado **futuro** respecto al estado **actual**): por este motivo podemos calcular el valor del estado en **T+2** considerando únicamente el valor de las entradas en **T+1**.

Si nos centramos, por ejemplo, en el análisis del estado en **T+3**, podemos observar que, en este caso, su valor (**0101**) depende de las entradas en **T+1** y **T+2**: en **T+1** porque es donde las entradas definen el valor memorizado y en **T+2** porque la inactividad de la habilitación mantiene el dato anterior. Por los mismos motivos en el instante **T+4** el estado depende del valor de las entradas en los tres instantes anteriores. Esta dependencia de la memoria se deriva también de la estructura de la ecuación del estado futuro: en los flancos de reloj en los que la señal de habilitación no está activa el estado futuro es el estado actual (figura 14).

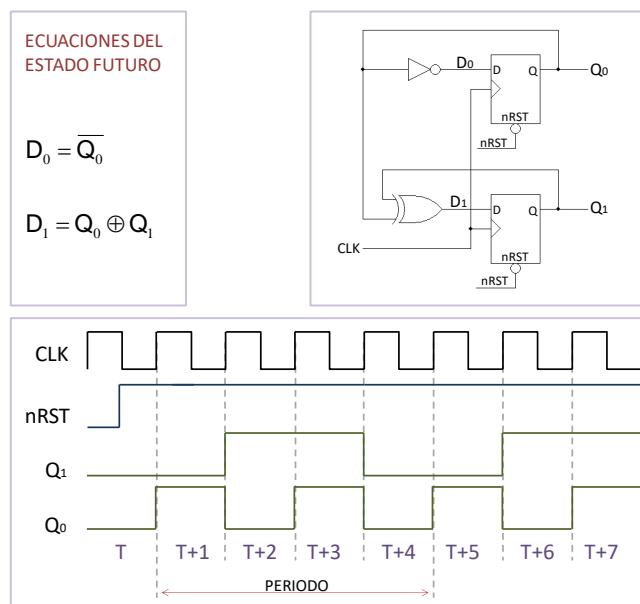


Figura 16.- Secuenciador

Antes de sacar conclusiones, vamos a analizar un tercer y último ejemplo (figura 16). Se trata de un circuito secuencial que genera una secuencia de combinaciones de manera cíclica: 00, 01, 10, 11. Este circuito

secuencial carece de entradas síncronas y su estado futuro depende, exclusivamente, del estado actual. Esta propiedad se refleja, lógicamente, en las ecuaciones de entrada de los flip-flops.

Este tipo de circuitos secuenciales, que se conocen como **secuenciadores**, sirven para generar secuencias de señales digitales periódicas, una función muy útil para generar señales de control para otros sistemas. En este tipo de circuitos la memoria actual depende únicamente del estado de memoria anterior y su estado representa, simplemente, el ciclo de la secuencia en que se encuentra el circuito (un estado que depende de los anteriormente *recorridos* dentro del periodo de repetición de la secuencia).

Los tres ejemplos que hemos analizado comprenden toda la casuística que se puede dar en la **estructura** de control de la memoria de un circuito secuencial. El diagrama de bloques de estas estructuras se representa en la figura 16.

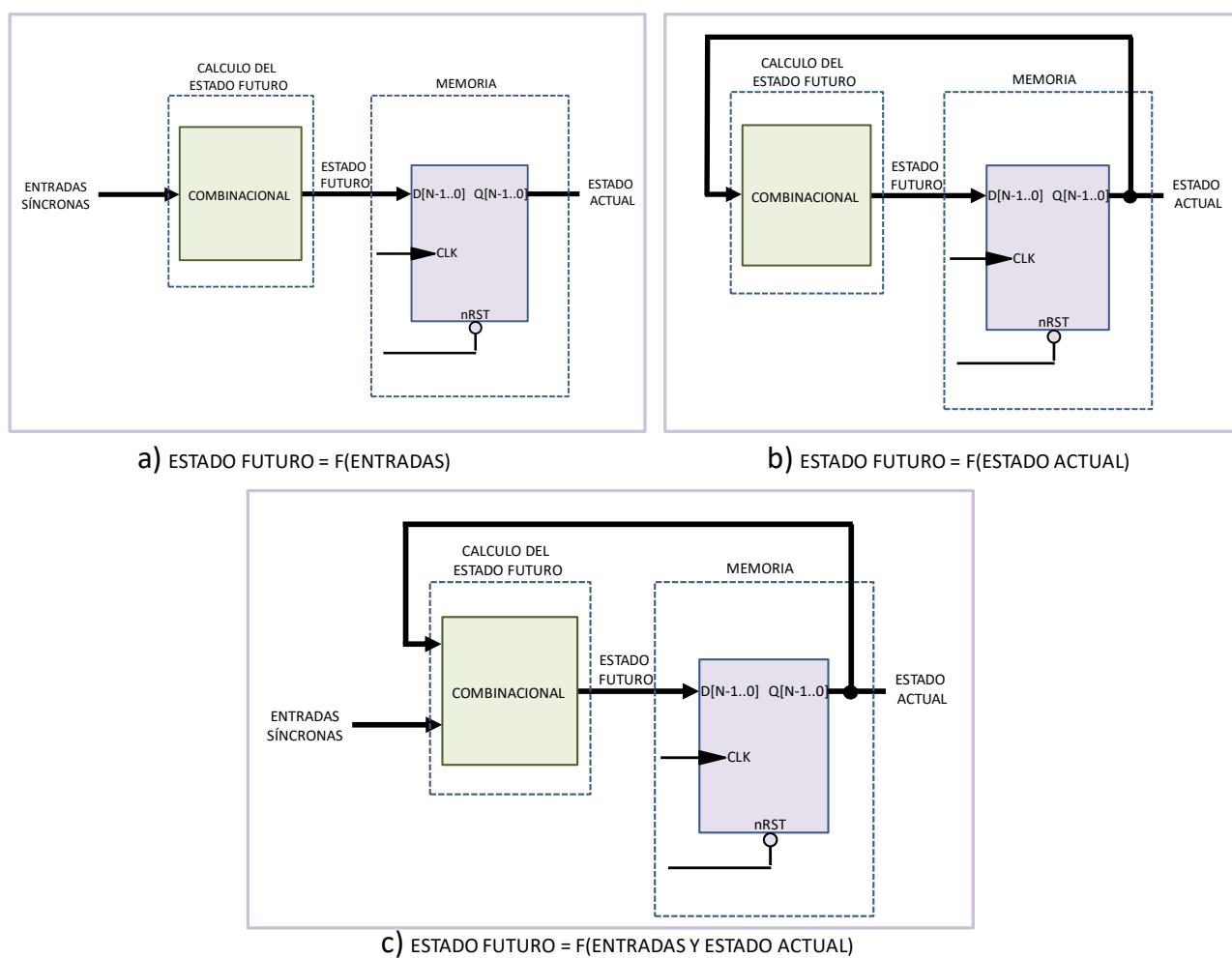


Figura 16.- Arquitectura del control del estado de memoria de un circuito secuencial

En la figura 16a se representa la arquitectura de los sistemas secuenciales en los que el estado de memoria futuro depende exclusivamente del valor de las entradas en el instante actual. Los únicos sistemas secuenciales que tienen esta arquitectura de control de memoria son los registros con escritura paralela y sin habilitación de reloj. La arquitectura de la figura 16b también es infrecuente: sólo se da en secuenciadores muy simples sin entradas de control síncronas. La arquitectura de la figura 16c, en cambio, es la que representa la estructura del control del estado de memoria en la inmensa mayoría de circuitos secuenciales. Este modelo de arquitectura engloba, además, a las otras dos, que no son más que casos

particulares de ésta, y es por tanto el **modelo de arquitectura de control de la memoria de estado de un circuito secuencial**.

En este modelo general se distinguen:

- La memoria de estado: Está compuesta por un conjunto de flip-flops conectados a la señal de reloj. La salida de estos flip-flops codifica el **estado actual** (en T) del circuito secuencial, mientras que la combinación en las entradas de los flip-flops, en el flanco activo de reloj que marca el paso de T a T+1, codifica el **estado futuro** (el que tomará el secuencial en T+1).

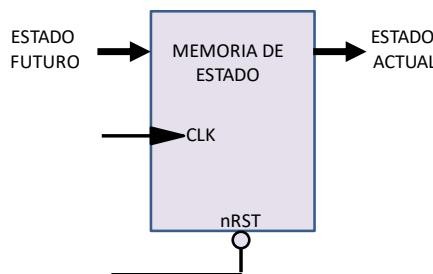


Figura 17.- Memoria de Estado

El número de flip-flops que componen la memoria de un sistema secuencial depende del número de estados diferentes que tenga y de su codificación. Por ejemplo, la memoria de un sistema con tres estados (A, B y C), codificados con dos bits ($A = 00$, $B = 01$, $C = 10$), puede realizarse con dos flip-flops; si la codificación se hiciera con tres bits ($A = 001$, $B = 010$, $C = 100$), se necesitarían tres flip-flops (Figura 18).

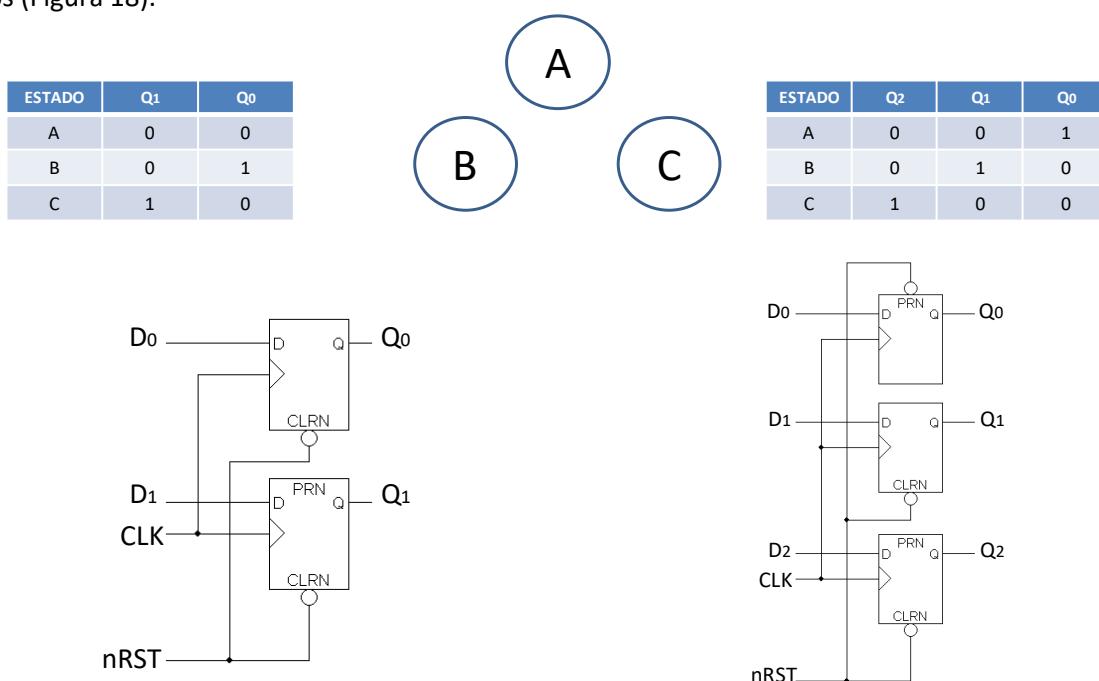
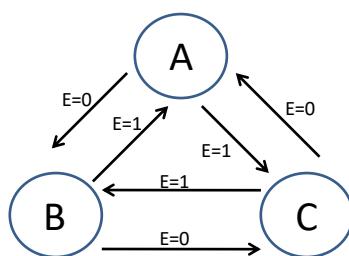
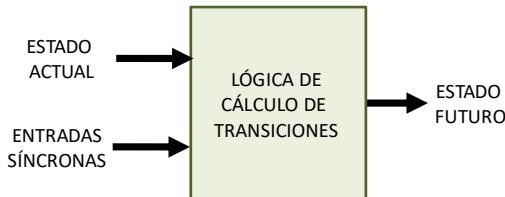


Figura 18

Como puede verse en la figura 18, todos los flip-flops de un sistema secuencial síncrono disponen de una señal de reloj común. La señal de inicialización asíncrona, también común, define el estado inicial del sistema (el estado de memoria en el primer instante de tiempo de funcionamiento del sistema: en los circuitos de la figura 18, la entrada nRST inicializa ambos circuitos en el estado A).

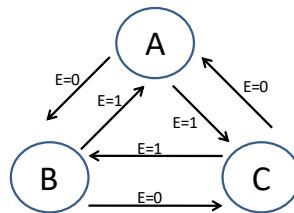
- La lógica combinacional de cálculo de transiciones entre estados: Este circuito calcula el estado futuro del circuito secuencial a partir de las entradas y del estado actual. Por este motivo se suele denominar **lógica de cálculo de transiciones**. Sus entradas son las entradas síncronas del circuito secuencial y las salidas de los flip-flops (el estado actual) y sus salidas son las entradas de los flip-flops (el estado futuro).



ESTADO ACTUAL	ENTRADA	ESTADO FUTURO
A	0	B
A	1	C
B	0	C
B	1	A
C	0	A
C	1	B

Figura 19.- Lógica de cálculo de transiciones

La tabla de verdad que define este circuito combinacional se deriva de las condiciones que determinan las transiciones en el diagrama de estados del sistema secuencial. Por ejemplo, del diagrama de estados de la figura 19 se obtiene la tabla que se muestra en la misma figura. Una vez asignada la codificación de estados de memoria, se puede construir la tabla de verdad y diseñar el circuito combinacional (Figura 20).



$$A = 00, B = 01, C = 10$$

ESTADO ACTUAL		ENTRADA	ESTADO FUTURO	
Q1	Q0	E	D1	D0
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1

$$D_1 = \bar{Q}_1 \cdot (Q_0 \oplus E)$$

$$D_0 = \bar{Q}_0 \cdot (\bar{Q}_1 \oplus E)$$

ESTADO ACTUAL			ENTRADA	ESTADO FUTURO		
Q2	Q1	Q0	E	D2	D1	D0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
1	0	0	0	0	0	1
1	0	0	1	0	1	0

$$A = 001, B = 010, C = 100$$

ESTADO ACTUAL			ENTRADA	ESTADO FUTURO		
Q2	Q1	Q0	E	D2	D1	D0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
1	0	0	0	0	0	1
1	0	0	1	0	1	0

$$D_2 = \bar{Q}_1 \cdot Q_0 \cdot E + Q_1 \cdot \bar{Q}_0 \cdot \bar{E}$$

$$D_1 = \bar{Q}_1 \cdot (Q_0 \oplus E)$$

$$D_0 = \bar{Q}_0 \cdot (\bar{Q}_1 \oplus E)$$

Figura 20.- Diseño de la lógica de cálculo de transiciones

En resumen: el bloque de control de la memoria de estado de un circuito secuencial síncrono está compuesto por un conjunto de flip-flops, cuya salida es el estado de memoria del circuito, y, salvo casos particulares (como el circuito de la figura 13), por un sistema combinacional que controla las transiciones, es decir, calcula el estado de memoria futuro, al que evoluciona el sistema secuencial en el siguiente flanko activo de reloj, en base al estado de memoria actual y al valor, también actual, de las entradas.

En la figura 21 se muestra la realización del control de memoria del sistema secuencial descrito por el diagrama de estados de la figura 20.

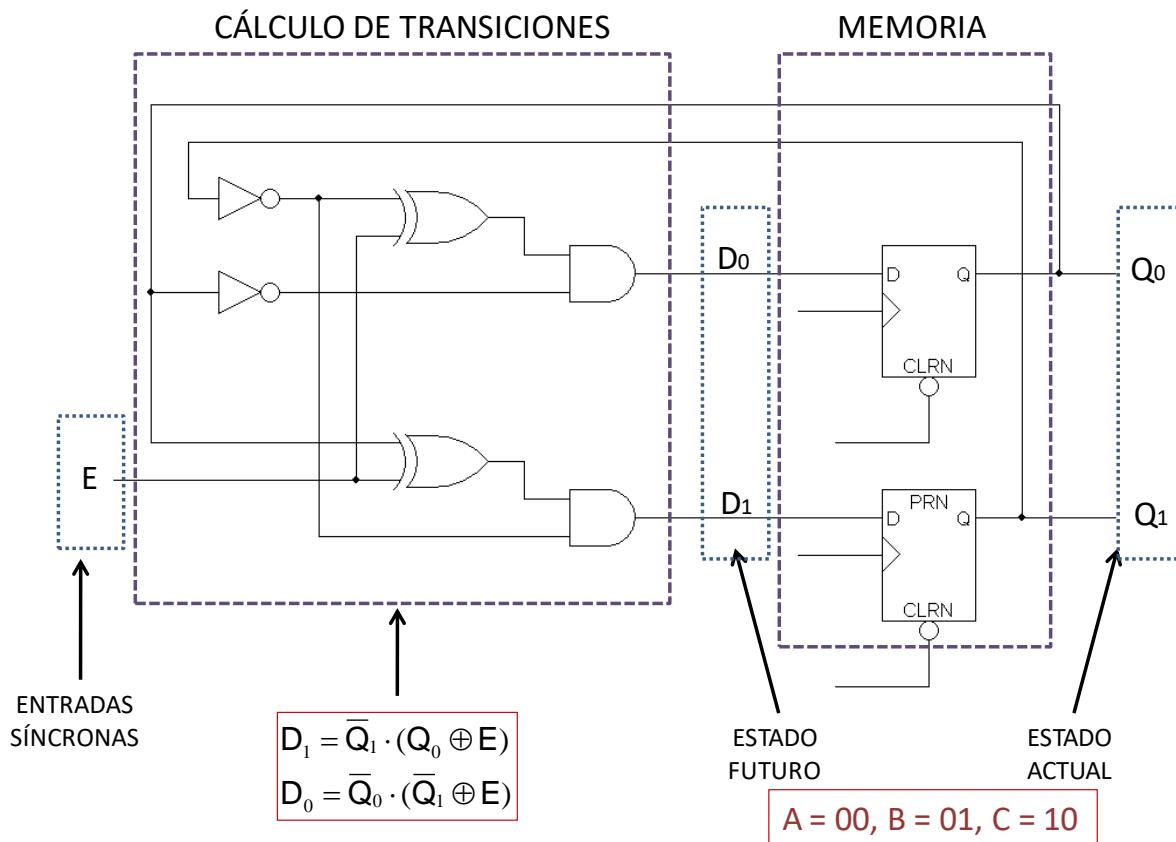


Figura 21.- Control del estado de memoria de un circuito secuencial

Arquitectura de salida

En todos los ejemplos de sistemas secuenciales que conoce hasta ahora, la salida del sistema es directamente el estado actual de memoria (las salidas de los flip-flops). Aunque esta característica es común a algunos sistemas de uso muy frecuente, como los registros o los contadores, en general la salida de un sistema secuencial se obtiene mediante el procesamiento combinacional de sus entradas sincrónicas y de la memoria de estado.

Por ejemplo, si un sistema tiene que indicar, en un instante de tiempo, T, el número de unos de un dato de 4 bits en el instante T-1, podría realizarse con un registro –que almacena en T el dato de entrada en T-1- cuya salida es la entrada de un circuito combinacional que calcula el número de unos de un dato de 4 bits (Figura 22).

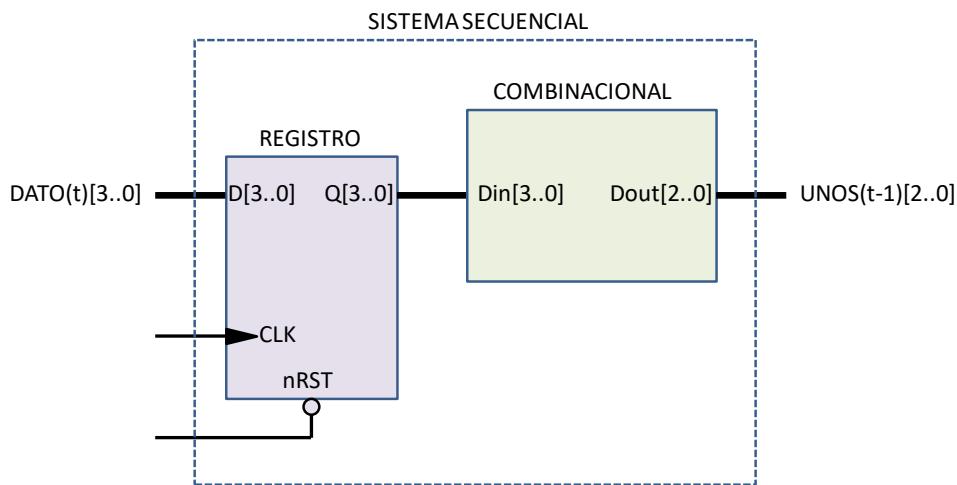


Figura 22.- Lógica combinacional de salidas que dependen del estado

En el ejemplo de la figura 22, aunque la salida no es directamente el estado actual de memoria, su valor depende única y exclusivamente de él.

Si, en cambio, el sistema secuencial tuviera que calcular el número de unos que totalizan los datos en el instante T y T-1, la salida podría calcularse tal y como se muestra en la figura 23. En este caso las salidas del sistema se obtienen a partir del estado actual y de las entradas síncronas.

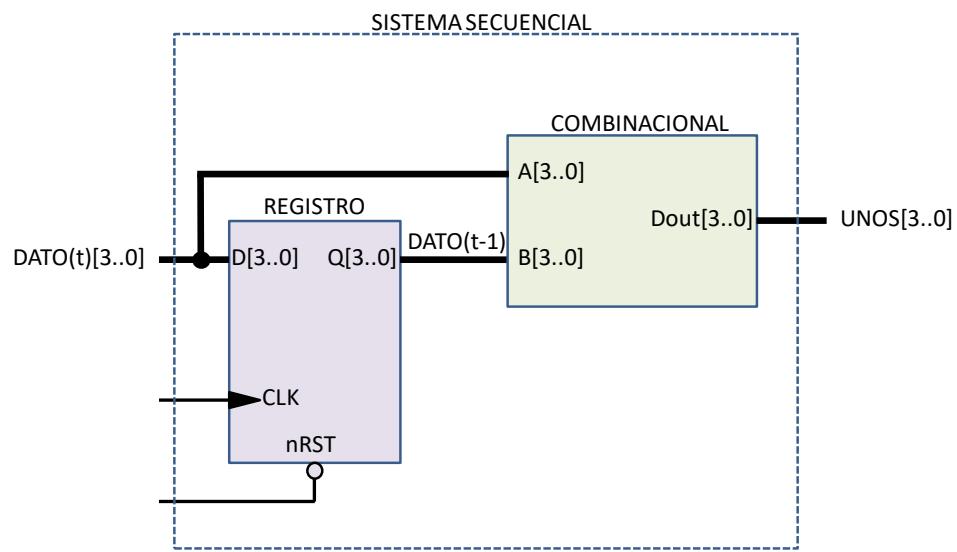


Figura 23.- Lógica combinacional de salidas que dependen del estado y las entradas

Los circuitos de las figuras 22 y 23 demuestran una propiedad de los sistemas secuenciales que ya conoce: las salidas para una determinada combinación de entradas pueden ser diferentes en distintos instantes de tiempos. En el circuito de la figura 22, porque dependen exclusivamente del estado de memoria y, en el de la figura 23, porque dependen de las entradas y del estado.

Observe, por otro lado, que el caso en que las salidas son el propio estado de memoria (como en los registros), es un caso particular de la arquitectura que ilustra el circuito de la figura 22, y que esta arquitectura, a su vez, es también un caso particular de la que representa el circuito de la figura 23.

Pero las características de los sistemas secuenciales que generan las salidas únicamente en función del estado de memoria y las de los que las generan a partir del estado y las entradas presentan diferencias que hacen conveniente distinguirlos: la arquitectura (estructura) de los primeros se conoce como **arquitectura de Moore** y la de los segundos como **arquitectura de Mealy**, y a los sistemas secuenciales síncronos cuya estructura se corresponde con cada una de ellas se los denomina, también, **autómatas de Moore** y **autómatas de Mealy**.

La representación de los valores de las salidas en los diagramas de estado varía en función del tipo de arquitectura (Moore o Mealy) que tenga.

- En los sistemas con arquitectura de Moore, la combinación de salida depende sólo del estado actual de memoria y, por tanto, puede representarse asociada a cada estado tal y como se muestra en el diagrama de la figura 24.

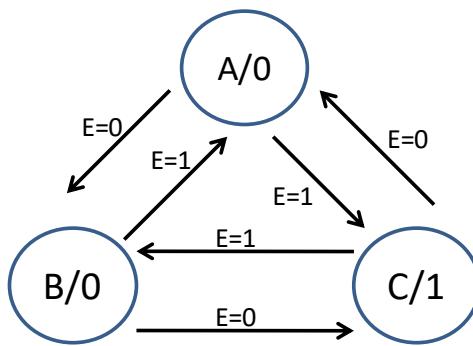


Figura 24.- Diagrama de estados de un autómata de Moore

En este diagrama de estados, cuyas transiciones corresponden al ejemplo utilizado anteriormente para ilustrar la arquitectura de control de la memoria, se indica en cada estado el valor que toma el único bit (en este caso) de salida: 1 en C, y 0 en A y B.

La lógica combinacional que genera las salidas en los autómatas de Moore se puede diseñar construyendo una tabla que relaciona cada estado con la combinación de salida asociada (Figura 25). Observe que, lógicamente, las variables de la ecuación de salida son bits de la memoria de estado.

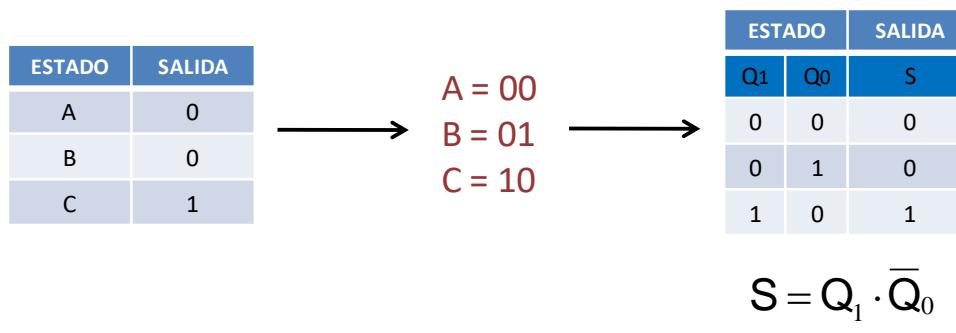


Figura 25.- Diseño de la lógica de salida en un autómata de Moore

- En los sistemas con arquitectura de Mealy, la salida depende del estado actual y de las entradas, por lo que para un determinado estado de memoria puede haber diferentes valores de salida y, por tanto, en los diagramas de estado no se puede asociar la salida al estado. Los diagramas de estado

de este tipo de arquitecturas indican los valores de salida en las transiciones, ya que cada transición corresponde a un estado y a una determinada combinación de entrada (Figura 26).

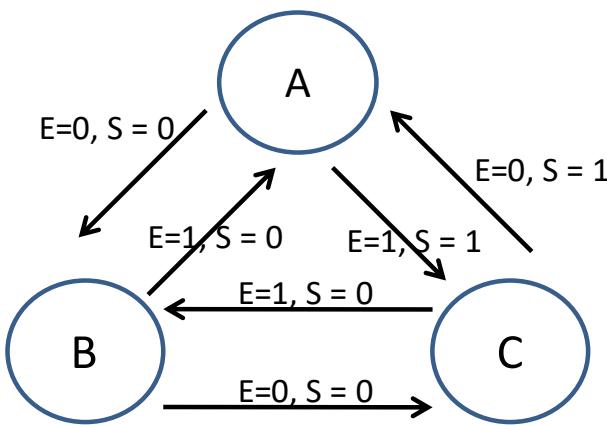


Figura 26.- Diagrama de estados de un autómata de Mealy

En este diagrama de estados, por ejemplo, la salida vale 1 cuando el estado actual de memoria es A y la entrada vale 1, o cuando el estado es C y la entrada vale 0; en el resto de casos la salida vale 0.

La lógica combinacional que genera las salidas en los autómatas de Mealy se puede diseñar construyendo una tabla que relaciona cada estado y combinación de las entradas con la combinación de salida asociada (Figura 27).

ESTADO	ENTRADA	SALIDA
A	0	0
A	1	1
B	0	0
B	1	0
C	0	1
C	1	0

→ A = 00
B = 01
C = 10 →

ESTADO	ENTRADA	SALIDA	
Q ₁	Q ₀	E	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0

$$S = \bar{Q}_1 \cdot \bar{Q}_0 \cdot E + Q_1 \cdot \bar{E}$$

Figura 27.- Diseño de la lógica de salida en un autómata de Moore

En este caso las variables de la ecuación extraída son la entrada síncrona (E) y un bit de la memoria de estado (Q_0).

Arquitectura completa de un circuito secuencial síncrono

Ya estamos en disposición de presentar la arquitectura completa de los circuitos secuenciales síncronos, combinando el modelo de control de los estados de memoria y el modelo de salida. Aunque sería posible reducirlos a uno solo, vamos a distinguir los dos tipos de arquitectura: la de Moore y la de Mealy.

El modelo de los circuitos secuenciales con arquitectura de Moore se muestra en la figura 28.

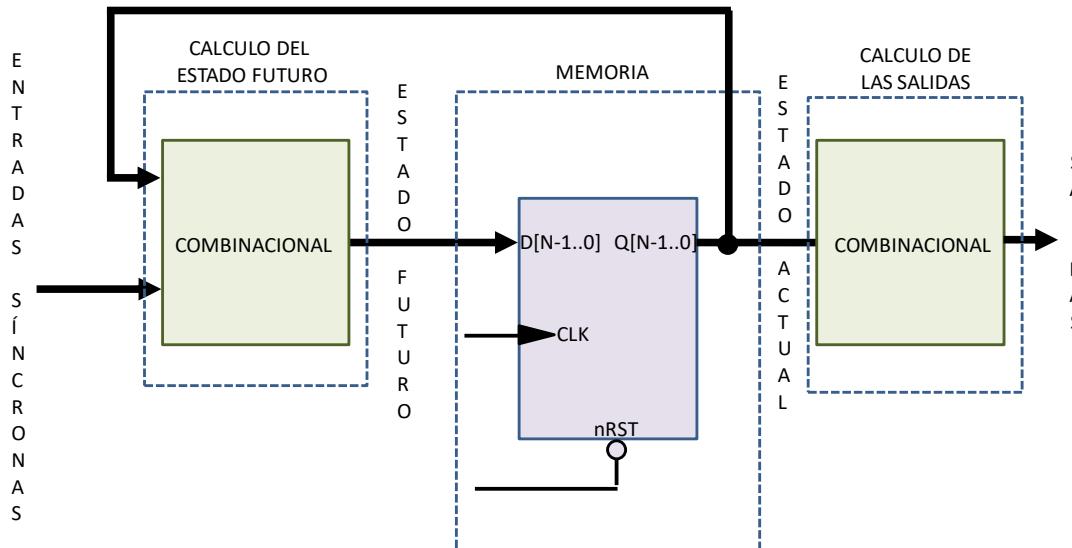


Figura 28.- Arquitectura de Moore

En la figura 29 se ilustra la estructura de las arquitecturas de Mealy.

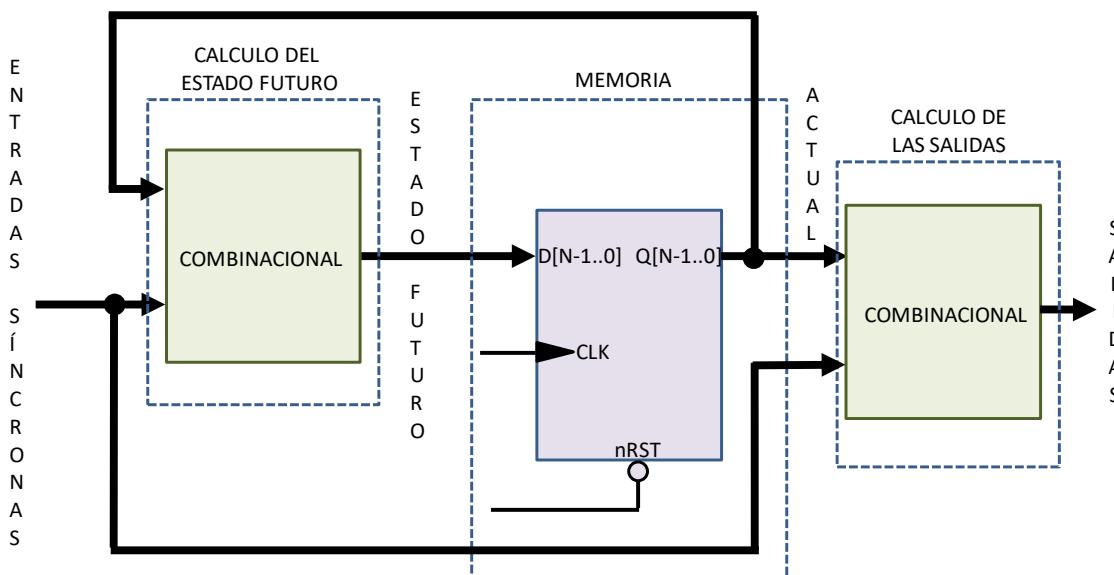
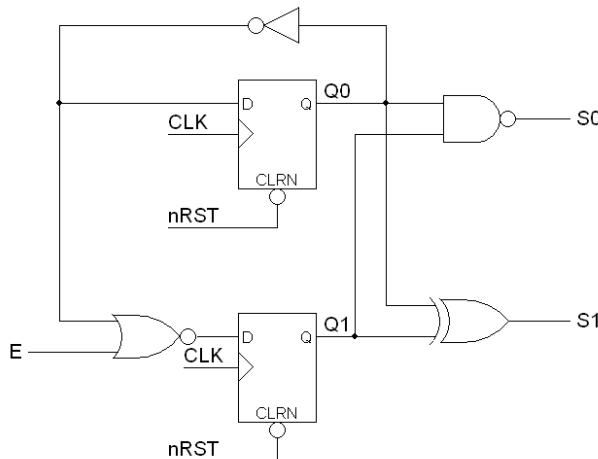


Figura 29.- Arquitectura de Mealy

Ambas arquitecturas comparten dos bloques comunes con idéntica interfaz y función, la memoria de estado y la lógica de cálculo de transiciones, y se diferencian únicamente por una característica del bloque combinacional que genera las salidas: en los autómatas de Moore la entrada de este bloque es el estado actual, mientras que en los de Mealy la entrada es el estado actual y las entradas síncronas del secuencial.

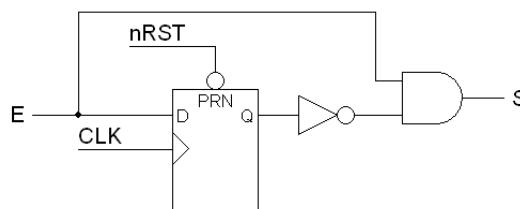
PARTE VI. Ejercicios de Arquitecturas de circuitos secuenciales síncronos

Ejercicio 1.- El circuito de la figura es la realización de un sistema secuencial síncrono.



- Obtenga las ecuaciones correspondientes a la lógica de cálculo de transiciones
 $D_1 =$
 $D_0 =$
- Obtenga las ecuaciones correspondientes a la lógica de cálculo de las salidas
 $S_1 =$
 $S_0 =$
- Deduzca el estado de inicialización asíncrona del circuito
 $Q_1 =$
 $Q_0 =$
- ¿Es un autómata de Mealy o de Moore? Razoné su respuesta

Ejercicio 2.- El circuito de la figura es la realización de un sistema secuencial síncrono.



- Obtenga las ecuaciones correspondientes a la lógica de cálculo de transiciones y de salida
 $D =$
 $S =$
- Deduzca el estado de inicialización asíncrona del circuito
 $Q =$
- ¿Es un autómata de Mealy o de Moore? Razoné su respuesta

Solución

Ejercicio 1

$$D_1 = \overline{\overline{Q}_0} + E$$

$$D_0 = \overline{Q_0}$$

$$S_1 = Q_0 \oplus Q_1$$

$$S_0 = \overline{Q_0 \cdot Q_1}$$

$$Q_1 = 0$$

$$Q_0 = 0$$

Es un autómata de Moore porque las salidas (S_1 y S_0) dependen únicamente del estado actual (Q_1 y Q_0).

Ejercicio 2

$$D = E$$

$$S = E \cdot \overline{Q}$$

$$Q = 1$$

Es un autómata de Mealy porque las salidas dependen del estado (Q) y de la entrada (E).

PARTE VII. Arquitectura de los chips configurables

En este curso se han estudiado ya las células configurables (PAL, LUT y generadores de funciones) que permiten materializar sistemas combinacionales. Para poder realizar sistemas secuenciales los chips configurables han de disponer de recursos adicionales a los que ya conoce –se le han ocultado hasta ahora para no complicar innecesariamente el modelo mínimo que necesitaba para entender cómo se podían materializar los circuitos combinacionales.

En esta parte del texto se van a presentar las estructuras completas que, realmente, integran los chips configurables. Estas estructuras disponen de una versatilidad que les permite realizar bloques combinacionales o secuenciales en función de las características del diseño que se desee materializar.

Para realizar un sistema secuencial en un chip configurable debe ser posible construir arquitecturas como las representadas en los diagramas de bloques de las figuras 28 y 29. Los bloques combinacionales de dicha figura pueden ser realizados con las células configurables que ya conoce, pero para materializar la memoria del sistema los chips se ha de disponer además de flip-flops. Y, además, debe ser posible disponer de recursos de interconexión que permitan componer la arquitectura conectando los distintos bloques que la forman.

Los bloques funcionales de los chips configurables, denominados **macrocélulas** (*macrocells*) por muchos fabricantes, tienen una estructura similar a la que se muestra en la figura 30.

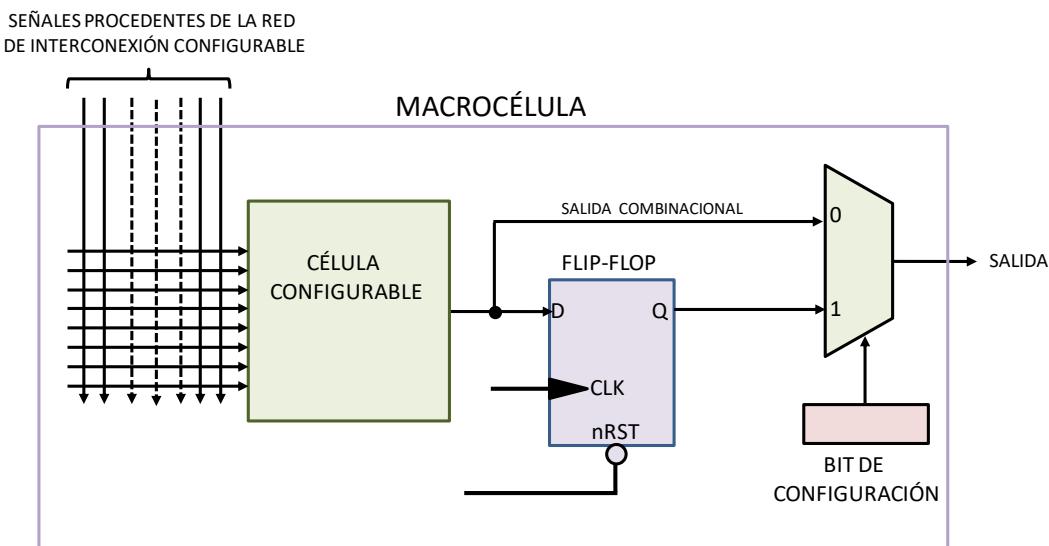


Figura 30.- Macrocelula de un chip configurable

Las macrocélulas disponen de un flip-flop conectado a la salida de una célula configurable (PAL o LUT). Para que una macrocélula pueda realizar funciones combinacionales y secuenciales, a salida del flip-flop y la salida de la célula configurable se conectan a un multiplexor direccionado por un bit de configuración, que es el que permite seleccionar la realización de una función secuencial o combinacional -si el bit de configuración selecciona la salida del flip-flop, la macrocélula realiza una función secuencial, en caso contrario materializa una función combinacional.

Las entradas de una macrocélula proceden de la red de interconexión configurable del chip. Por medio de esta red puede encaminarse hasta la entrada de la macrocélula cualquier señal ingresada a través de los pines del chip y, también, la salida de cualquiera de las macrocélulas del chip, lo que permite que la salida de

un flip-flop de una macrocélula pueda conectarse a la entrada de la célula combinacional configurable que maneja su entrada.

Por tanto:

- la memoria de estado de un circuito con la arquitectura de las figuras 25 ó 26 puede materializarse con los flip-flops de las macrocélulas,
- y la lógica combinacional para el cálculo de las transiciones puede realizarse con las células configurables que manejan las entradas de los flip-flops.

Las entradas de la lógica de cálculo de transiciones serán (como en el modelo de arquitectura):

- las entradas síncronas del secuencial (un conjunto de señales que llegan a través de la red de interconexión y que proceden de los pines o de otros bloques funcionales, combinacionales o secuenciales, materializados en el chip) y
- el estado actual (las salidas de los flip-flops que materializan la memoria de estado, que llegan también por medio de la red de conexión configurable).

En la figura 31 se muestra esta estructura y su concordancia con las necesidades del modelo general de arquitectura de control de la memoria de un sistema secuencial síncrono.

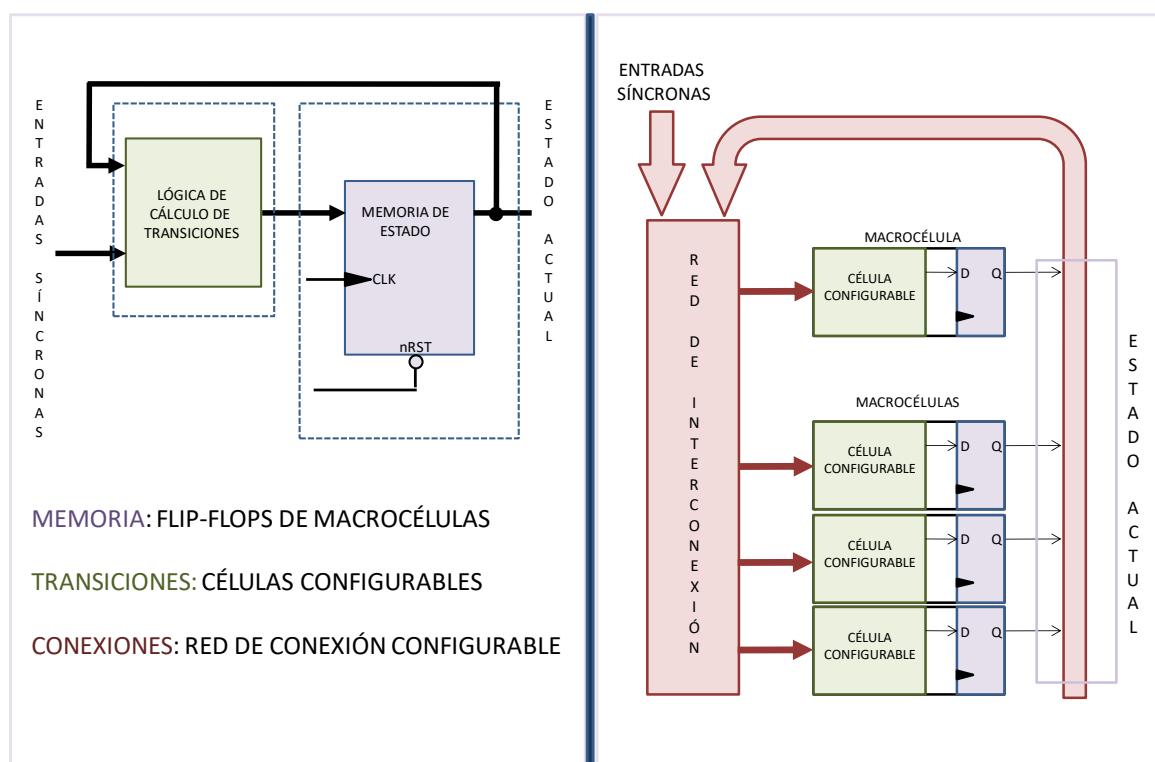


Figura 31.- Realización de la memoria de estado y la lógica de transiciones en un chip configurable

La lógica de salida del modelo de arquitectura se realiza de manera análoga, conectando las salidas de los flip-flops, por medio de la red de interconexión, a entradas de macrocélulas que se configuran con funciones combinacionales (Figura 32).

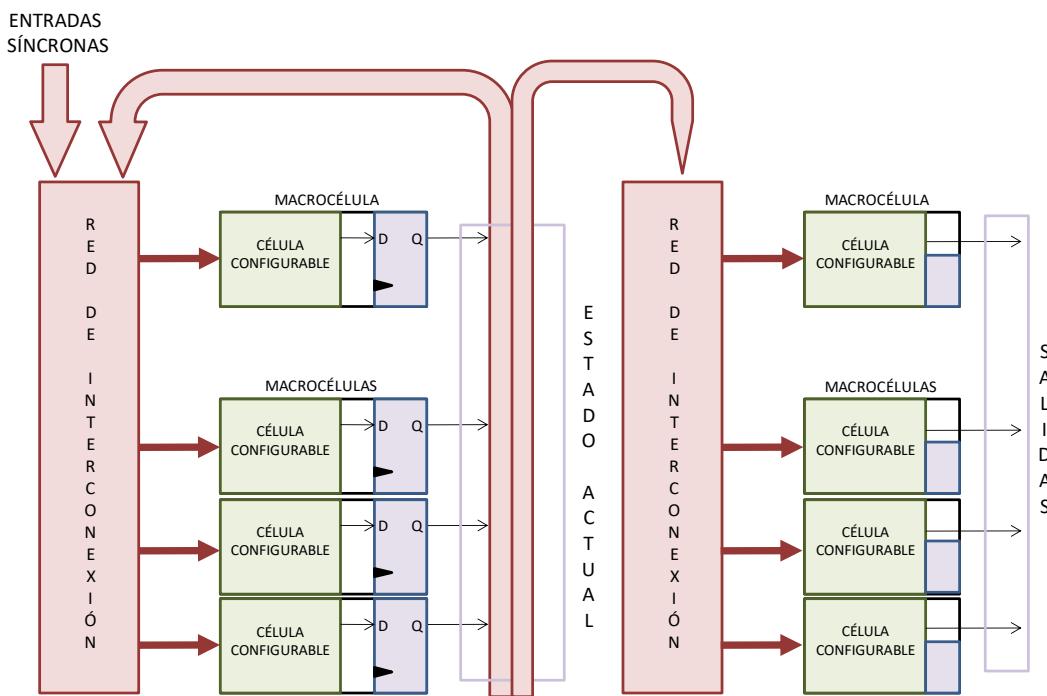


Figura 32.- Realización del modelo de arquitectura de Moore en un chip configurable

Hay que añadir, por último, que los chips configurables disponen de pines especiales para las señales de reloj e inicialización de los circuitos secuenciales. Se trata de pines **dedicados** que están conectados directamente a las entradas de reloj e inicialización asíncrona de todos los flip-flops del chip. Estos pines son los que deben ser utilizados para conectar el oscilador de reloj y la lógica de reset del sistema.

Ejemplo

La figura 33 muestra la representación, proporcionada por el fabricante, de la estructura lógica de las macrocélulas de la familia MAX 3000A de ALTERA.

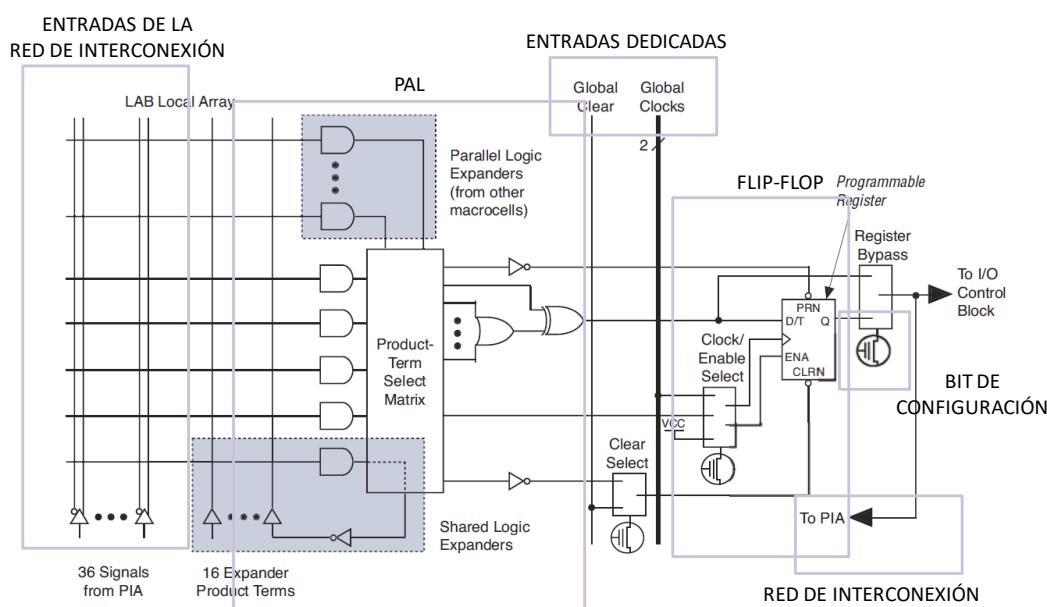


Figura 33.- Estructura de las macrocélulas de la familia MAX 3000A

En el dibujo se distingue:

- la célula configurable, una PAL con algunos *aditamentos* que potencian su capacidad de *encajar* funciones combinacionales,
- un flip-flop tipo D, cuya entrada síncrona está conectada a la salida de la célula configurable;
- las entradas asíncronas y el reloj, que disponen de conexión directa a las entradas **dedicadas** de reloj e inicialización (entradas *globales* de *Clear* y *Clock*), pero pueden ser manejadas también por señales procedentes de la red de interconexión configurables.

El tipo de salida de la macrocélula se configura mediante un bit que selecciona entre la salida de la PAL y del flip-flop: sea de un tipo u otro, puede conectarse a un pin configurable del chip (*I/O Control Block*) y a la red de interconexión configurable (*PIA, Programmable Interconnect Array*) para servir como entrada de otra macrocélula (o de ella misma). A cada macrocélula llegan 36 señales de entrada, por medio de la PIA, que pueden proceder de los pines y de las salidas de cualquier macrocélula del chip.

Esta página se ha dejado intencionadamente en blanco

BLOQUE TEMÁTICO 3**TÍTULO DE LA ACTIVIDAD:****Ejercicios sobre registros y arquitectura de los sistemas secuenciales.****CÓDIGO:****BT3.ATGP2**

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Ejercicios. Aprendizaje colaborativo	TIPO:	Presencial	DURACIÓN:	65 minutos
------------	--------------------------------------	-------	------------	-----------	------------

CALENDARIO:	Tercera Sesión.	REQUISITOS:	Sistemas secuenciales: flip-flops, registros. Arquitectura de los sistemas secuenciales.
-------------	-----------------	-------------	--

CRITERIO DE ÉXITO:	Todos los ejercicios bien hechos
--------------------	----------------------------------

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTEOVALUACIÓN: [entre 0 y 10 puntos]	En el cuestionario de moodle
------------------	---------	--	------------------------------

Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

Introducción.

Esta actividad se realizará en grupos G4.

Para los ejercicios 1 a 4:

- Cada miembro del grupo realizará uno de los ejercicios (10 minutos máx.).
- Cada miembro del grupo explicará a los demás un resumen del ejercicio que ha hecho y cómo lo ha resuelto (aprox. 2 minuto por ejercicio, 10 minutos máx. entre todos).
- El profesor explicará las soluciones de los cuatro ejercicios (aprox. 1 minuto por ejercicio, 5 minutos máx. entre todos).

Para los ejercicios 5 a 8:

- Los miembros del grupo realizarán de manera conjunta los cuatro ejercicios (10 minuto máx.)
- El profesor explicará las soluciones de los cuatro ejercicios (5 minutos máx. entre todos).

Para el ejercicio 9:

- Todos los miembros del grupo realizarán, de manera individual, el mismo ejercicio (15 minutos máx.).
- Posteriormente se hará una puesta en común en el grupo para comparar las soluciones de cada miembro del grupo (5 minutos máx.).
- El profesor explicará la solución del ejercicio (5 minutos máx.).

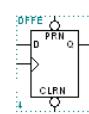
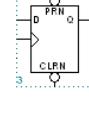
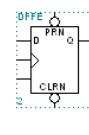
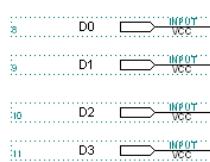
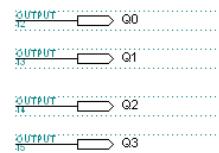
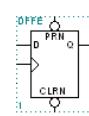
Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

Ejercicio 1: Registro paralelo con habilitación de escritura y reset síncrono

En este ejercicio se trata de diseñar un registro paralelo de 4 bits con habilitación de escritura activa a nivel bajo y reset síncrono activo alto. La activación de la habilitación de escritura no es necesaria para la operación de reset síncrono. El funcionamiento se resume en la siguiente tabla:

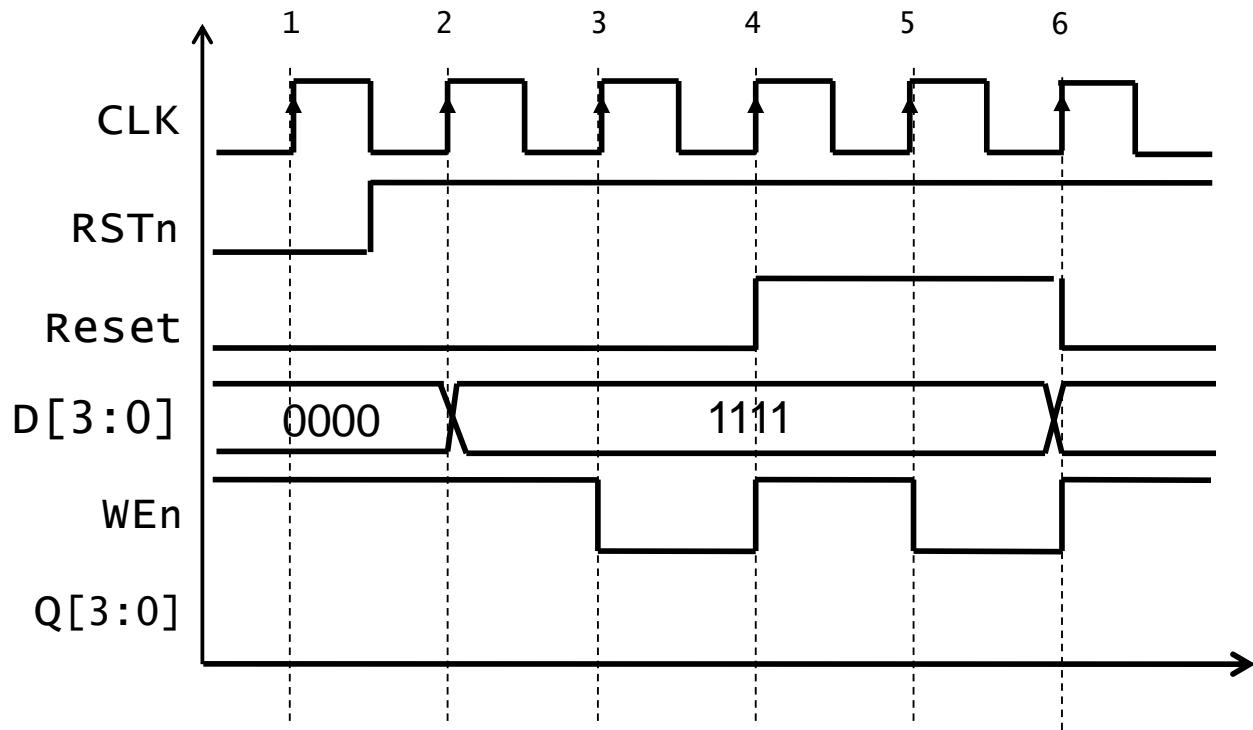
Entradas					Salida	Funcionamiento
RSTn	Reset	D _i	WEn	CLK	Q _i	
0	X	X	X	X	0	Reset asíncrono
1	1	X	X	↑	0	Reset síncrono
1	0	X	1	↑	q _i	Memoria
1	0	d _i	0	↑	d _i	Captura el dato

- A. Diseño. Diseñe el registro utilizando los flip-flops D de la figura y la lógica adicional que necesite.



Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

B. Cronograma. Complete el cronograma de la figura, relativo al circuito que acaba de diseñar.



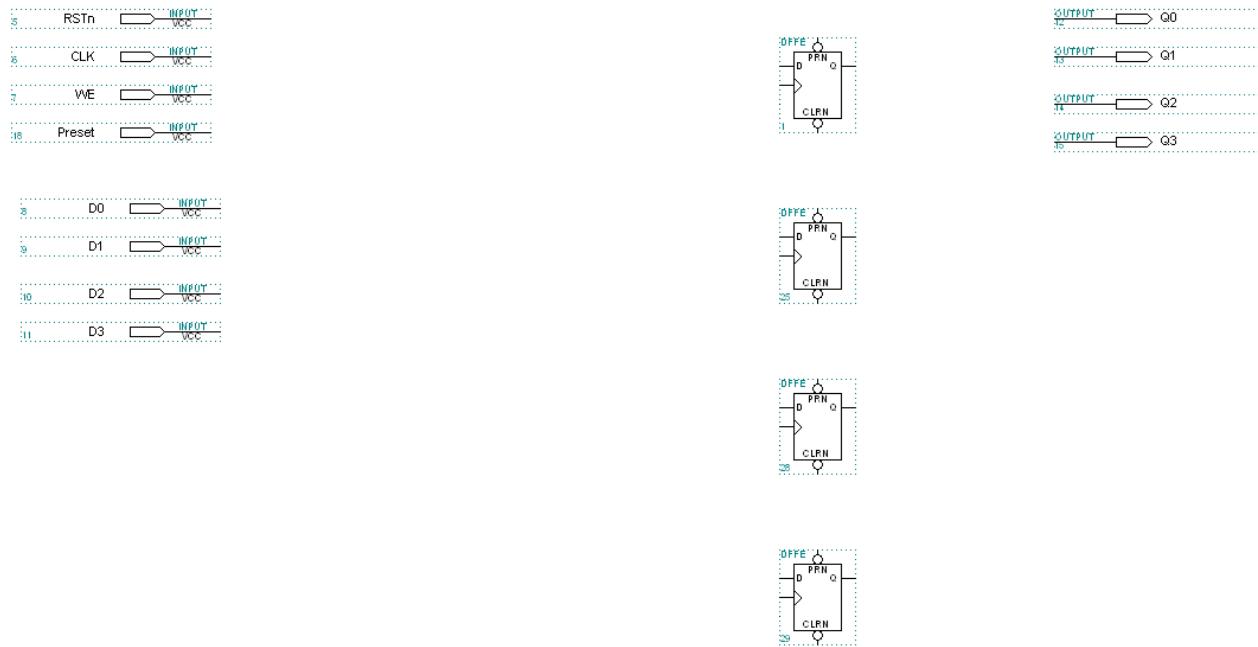
Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

Ejercicio 2: Registro paralelo con habilitación de escritura y preset síncrono.

En este ejercicio se trata de diseñar un registro paralelo de 4 bits con habilitación de escritura activa a nivel alto y preset síncrono. La habilitación de escritura debe estar activa para que se produzca el preset. El funcionamiento se resume en la siguiente tabla:

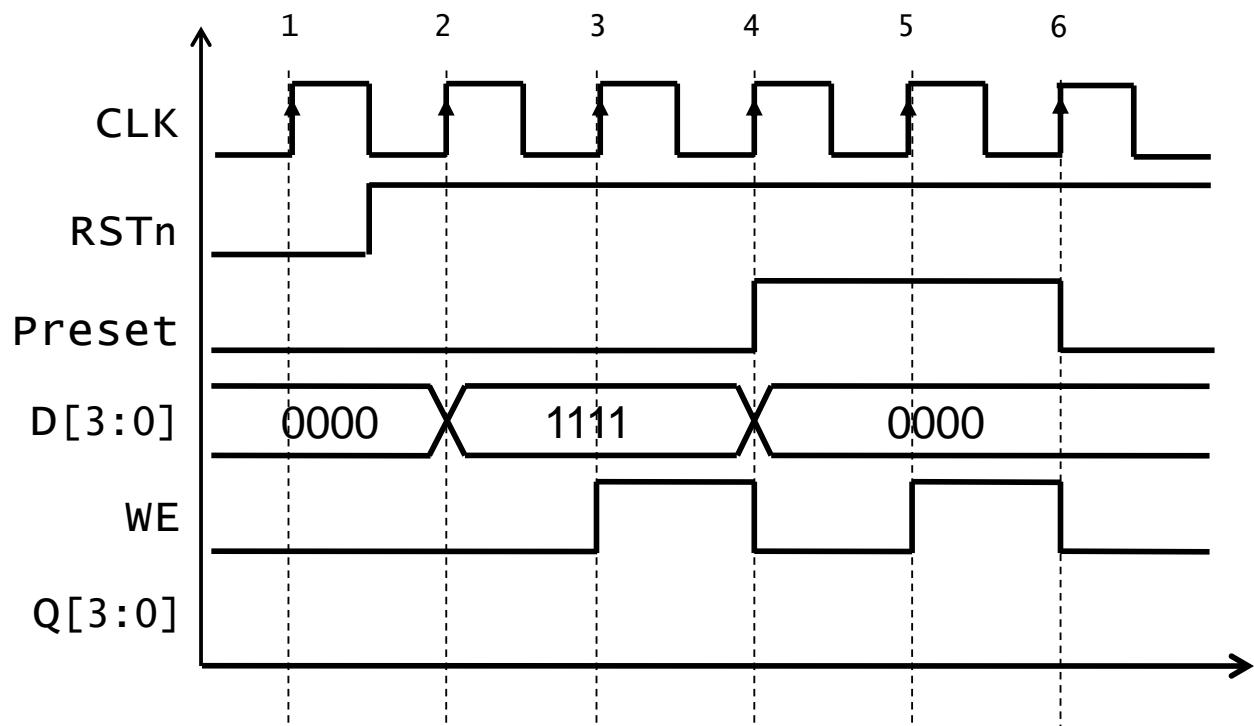
Entradas					Salida	Funcionamiento
RSTn	Preset	D _i	WE	CLK	Q _i	
0	X	X	X	X	0	Reset asíncrono
1	X	X	0	↑	q _i	Memoria
1	1	X	1	↑	1	Preset síncrono
1	0	d _i	1	↑	d _i	Captura el dato

- A. Diseño. Dibuje el registro utilizando los flip-flops D de la figura y la lógica adicional que necesite.



Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

B. Cronograma. Complete el cronograma de la figura, relativo al circuito que acaba de diseñar.



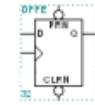
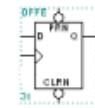
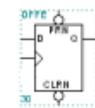
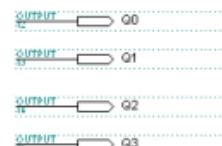
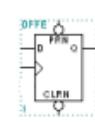
Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

Ejercicio 3: Registro de desplazamiento con entrada serie y salida paralelo.

En este ejercicio se trata de diseñar un registro de desplazamiento de 4 bits con entrada serie y salida paralelo. El registro tiene, además, habilitación de reloj activa a nivel alto. El funcionamiento se resume en la siguiente tabla:

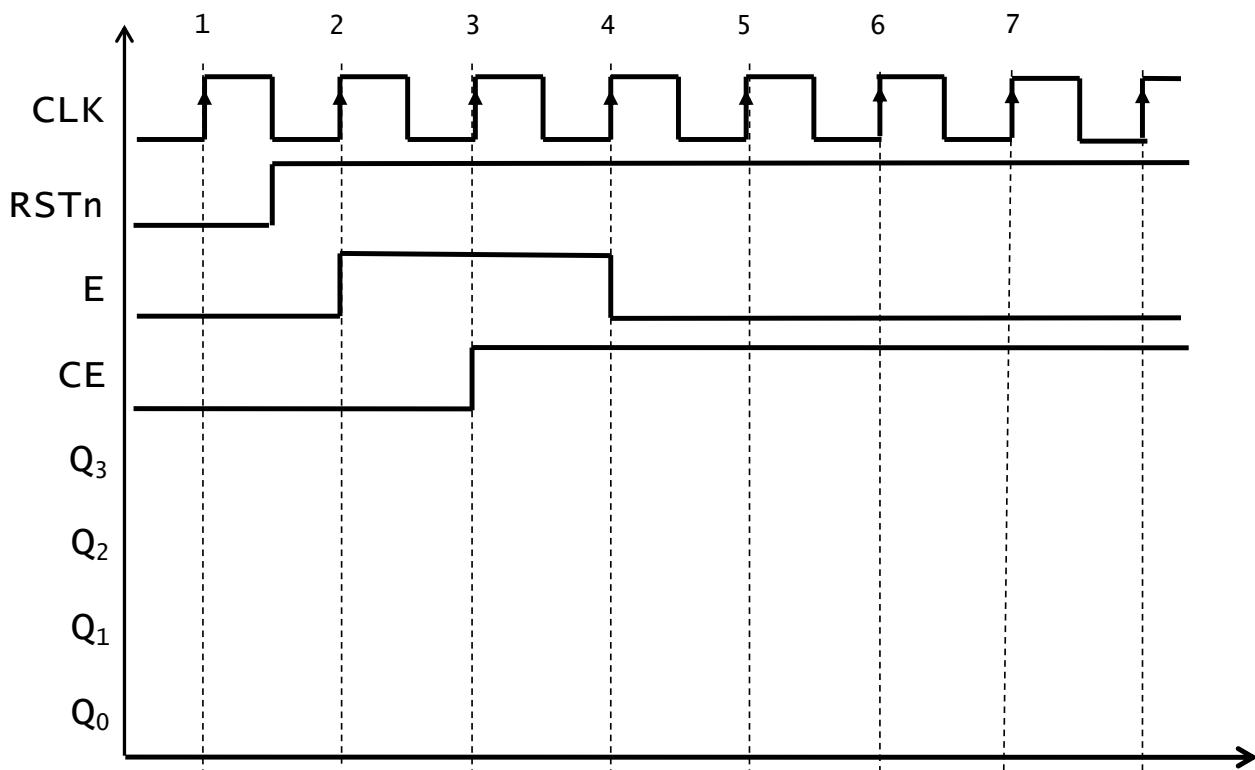
Entradas				Salida				Funcionamiento
RSTn	E	CE	CLK	Q ₃	Q ₂	Q ₁	Q ₀	
0	X	X	X	0	0	0	0	Reset asíncrono
1	X	0	↑	q ₃	q ₂	q ₁	q ₀	Memoria
1	e	1	↑	q ₂	q ₁	q ₀	e	Desplazamiento

A. Diseño. Dibuje el registro utilizando los flip-flops D de la figura.



Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

B. Cronograma. Complete el cronograma de la figura, relativo al circuito que acaba de diseñar.



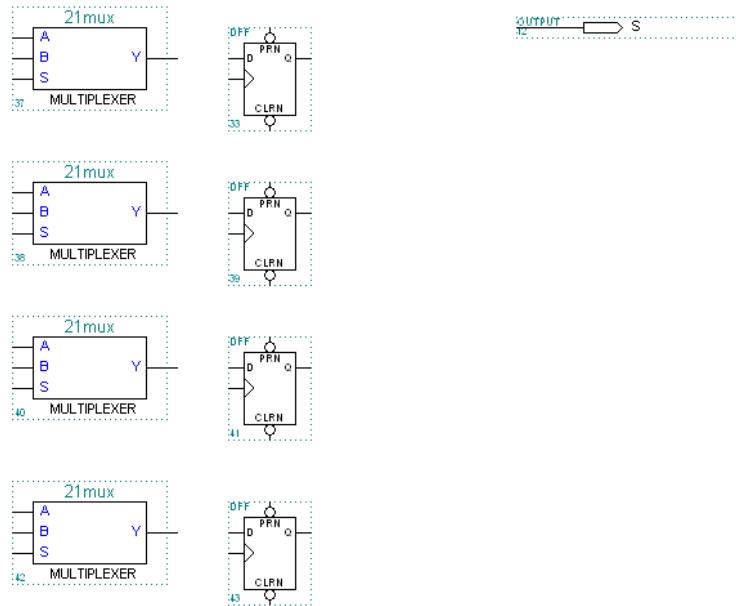
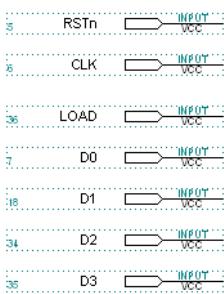
Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

Ejercicio 4: Registro de desplazamiento con entrada paralelo y salida serie.

En este ejercicio se trata de diseñar un registro de desplazamiento de 4 bits con entrada paralelo y salida serie. El funcionamiento se resume en la siguiente tabla:

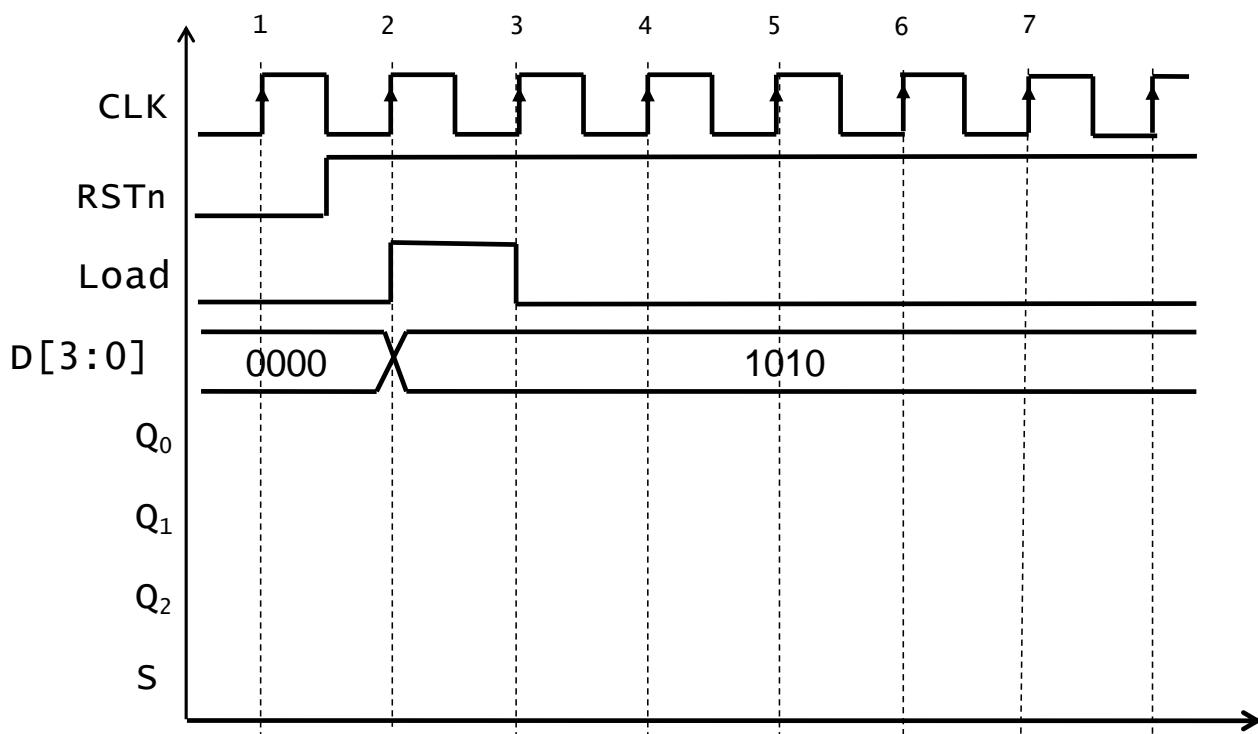
Entradas							Salida	Resto flip-flops			Funcionamiento
RSTn	D ₃	D ₂	D ₁	D ₀	Load	CLK	S	Q ₂	Q ₁	Q ₀	
0	X	X	X	X	X	X	0	0	0	0	Reset asíncrono
1	d ₃	d ₂	d ₁	d ₀	1	↑	d ₃	d ₂	d ₁	d ₀	Carga
1	X	X	X	X	0	↑	q ₂	q ₁	q ₀	0	Desplazamiento

- A. **Diseño.** Dibuje el registro utilizando los flip-flops D y los multiplexores de la figura (los multiplexores dejan pasar A con S = 1 y B con S = 0).



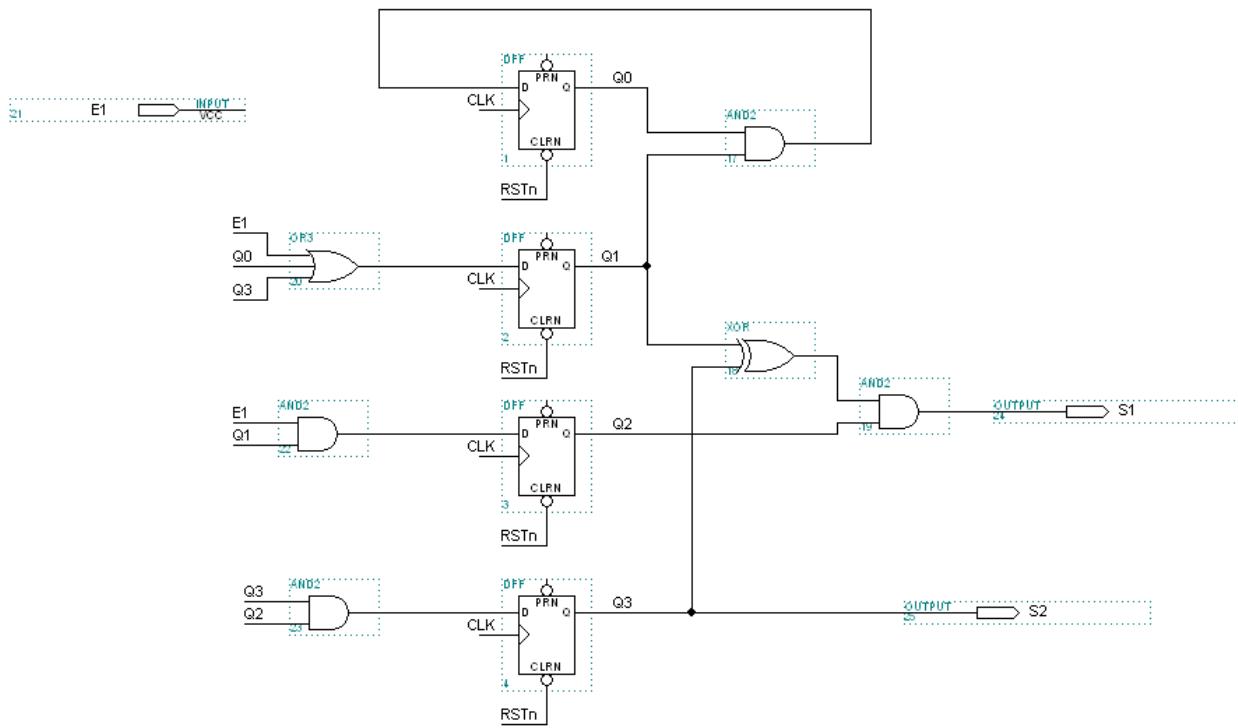
Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

B. Cronograma. Complete el cronograma de la figura, relativo al circuito que acaba de diseñar.

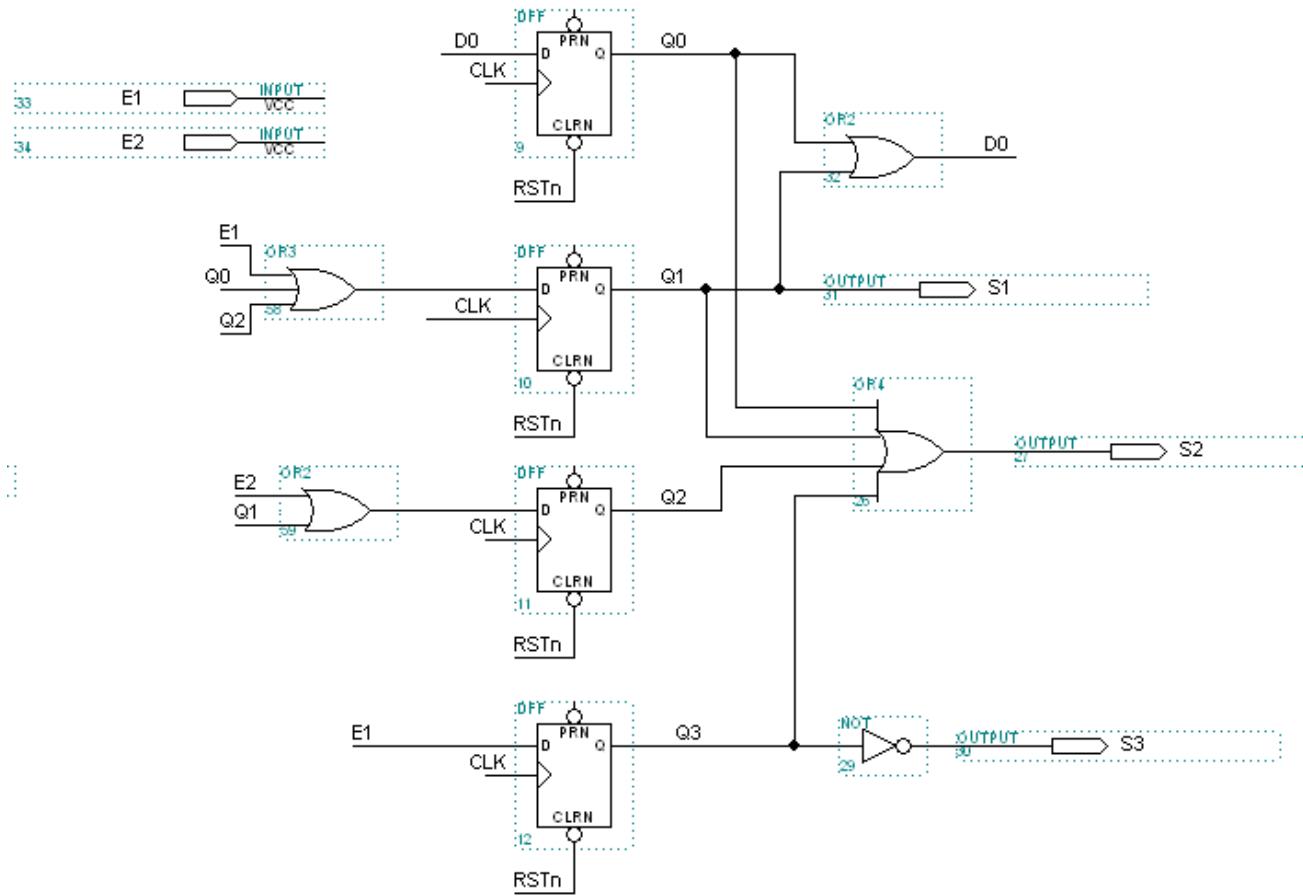


Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

Ejercicio 5: En el siguiente circuito identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.

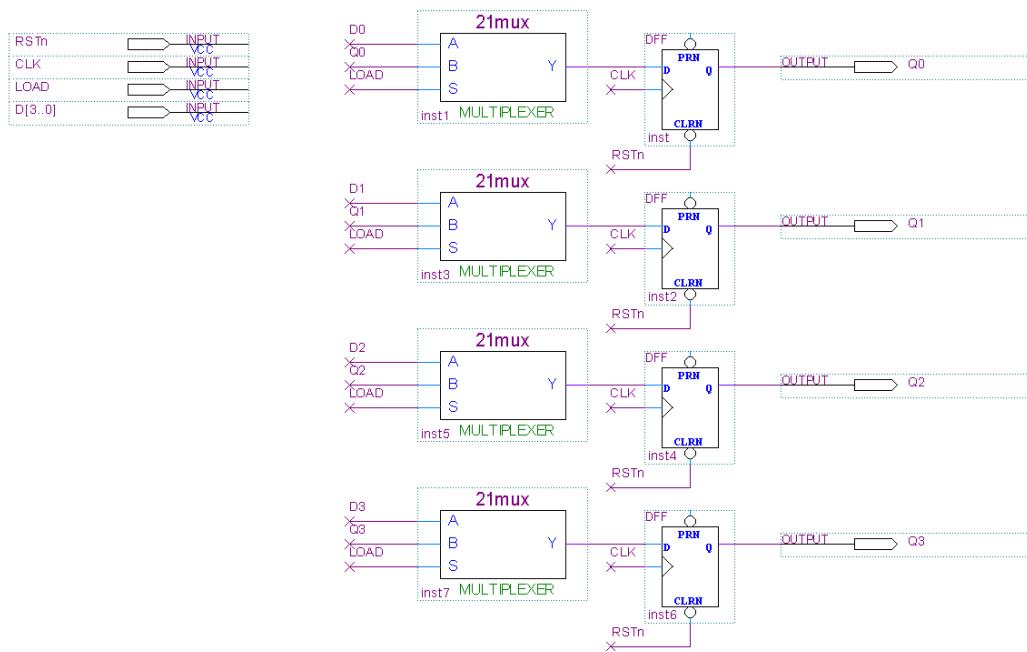


Ejercicio 6: En el siguiente circuito identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.

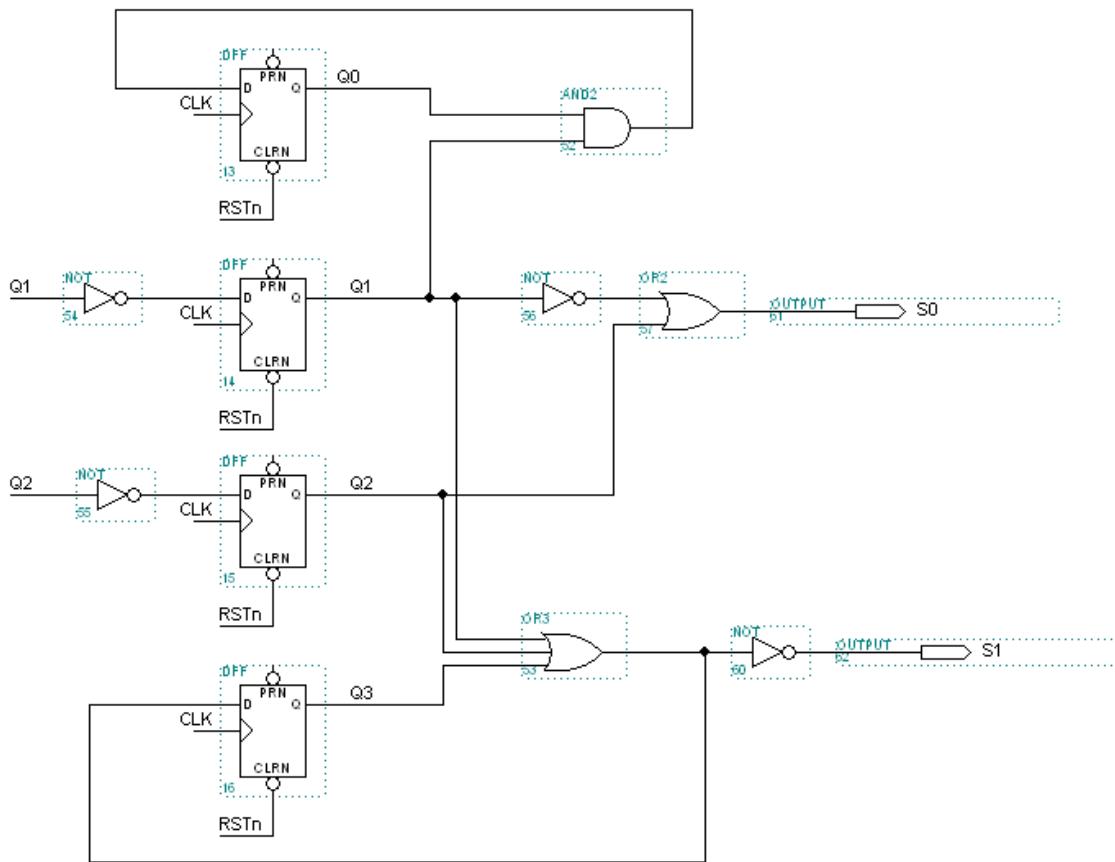


Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

Ejercicio 7: En el siguiente circuito identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.

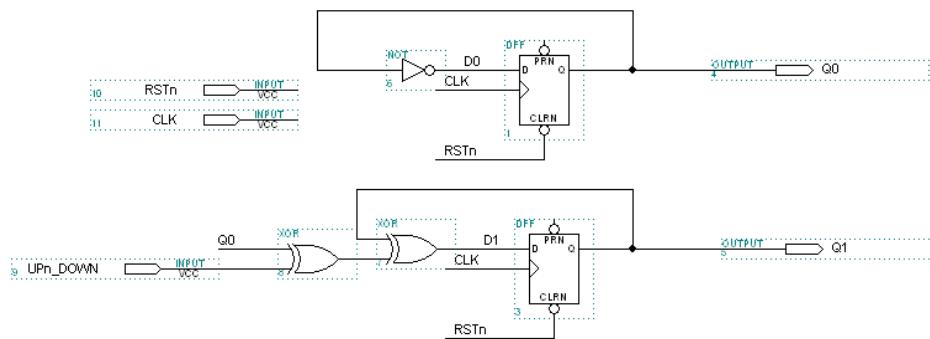


Ejercicio 8: En el siguiente circuito identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.



Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

Ejercicio 9: Conteste a los siguientes apartados, todos ellos relativos al circuito secuencial de la figura:



- A. **Estructura.** Identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.
- B. **Ecuaciones del estado futuro.** Obtenga las ecuaciones booleanas que ligan la entrada y el estado actual con el estado futuro. Complete la tabla de verdad que se adjunta.

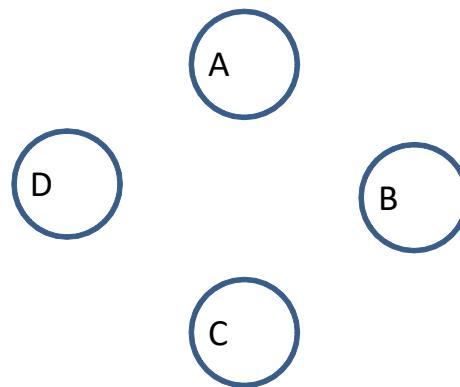
$$D0 =$$

$$D1 =$$

Entrada	Estado actual		Estado futuro		
	UPn_DOWN	Q1 _t	Q0 _t	D1=Q1 _{t+1}	D0=Q0 _{t+1}
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

- C. **Diagrama de estados.** Complete el diagrama de estados del circuito. Únicamente tiene que dibujar las transiciones entre estados para los diferentes valores de la entrada.

Estado	Código	
	Q1	Q0
A	0	0
B	0	1
C	1	0
D	1	1



Actividad BT3_ATGP2: Ejercicios sobre registros y arquitectura de los sistemas secuenciales.

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Ejercicios sobre sistemas secuenciales.

CÓDIGO:
BT3_ATGP3

GRUPO		FECHA:	
-------	--	--------	--

MODALIDAD:	Aprendizaje Cooperativo	TIPO:	Presencial	DURACIÓN:	100 minutos
------------	-------------------------	-------	------------	-----------	-------------

CALENDARIO:	Cuarta sesión	REQUISITOS:	
-------------	---------------	-------------	--

CRITERIO DE ÉXITO:	Realización correcta de los ejercicios.
--------------------	---

ROLES:	No hay distribución diferenciada de roles entre los miembros
--------	--

HABILIDADES SOCIALES:	Capacidad para expresar y sintetizar ideas, capacidad para escuchar, asimilar y organizar la información presentada por otra persona. Capacidad para razonar y analizar.
-----------------------	--

REFLEXIÓN DE GRUPO:	
---------------------	--

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	
------------------	------------	-------------------------------------	--

Actividad BT3_ATGP3: Ejercicios sobre sistemas secuenciales.

Introducción.

Esta actividad se realizará en la cuarta sesión presencial, en grupos G4. Se realizará un total de 7 ejercicios.

- Todos los miembros del grupo realizarán de manera individual el mismo ejercicio (8 minutos cada ejercicio, por término medio).
- Se realizará una puesta en común de las soluciones aportadas por cada miembro del grupo. Se constatarán las diferencias, si las hubiere, y se debatirá sobre las mismas (2 minutos máx.).

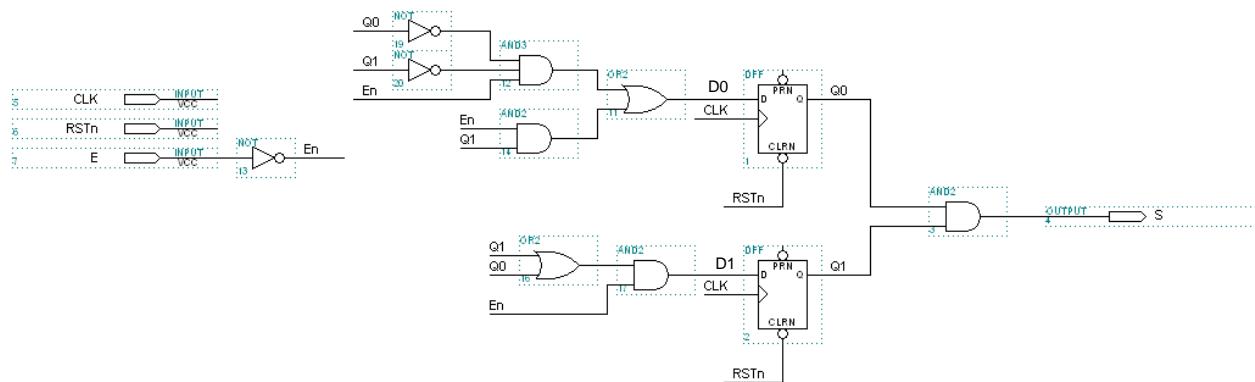
Para el ejercicio 7:

- Las mesas se colocarán en configuración de examen.
- Cada miembro del grupo realizará el ejercicio de forma individual (10 minutos máx.).
- El profesor recogerá un ejercicio de cada grupo G4 al azar.

Actividad BT3_ATGP3: Ejercicios sobre sistemas secuenciales.

Ejercicio 1: Cálculo de las salidas y el estado futuro.

Complete los siguientes apartados para el circuito de la figura:



- A. **Estructura.** Identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.
- B. **Salidas.** Obtenga la ecuación de la salida en función del estado actual. Complete la tabla de verdad que se adjunta.

$$S =$$

Estado	Código		Salida
	Q1	Q0	
A	0	0	
B	0	1	
C	1	0	
D	1	1	

- C. **Estado futuro.** Obtenga las ecuaciones del estado futuro en función del estado actual y de las entradas. Complete la tabla de verdad que se adjunta.

$$D0 =$$

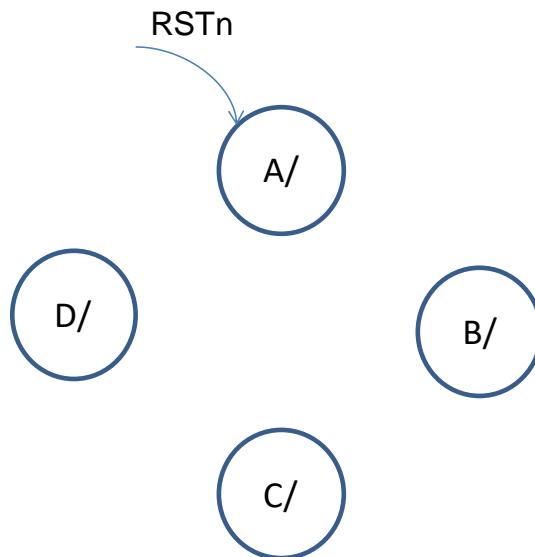
$$D1 =$$

Entrada	Estado actual		Estado futuro		
	E	Q1 _t	Q0 _t	D1=Q1 _{t+1}	D0=Q0 _{t+1}
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Actividad BT3_ATGP3: Ejercicios sobre sistemas secuenciales.

Ejercicio 2: Diagrama de estados.

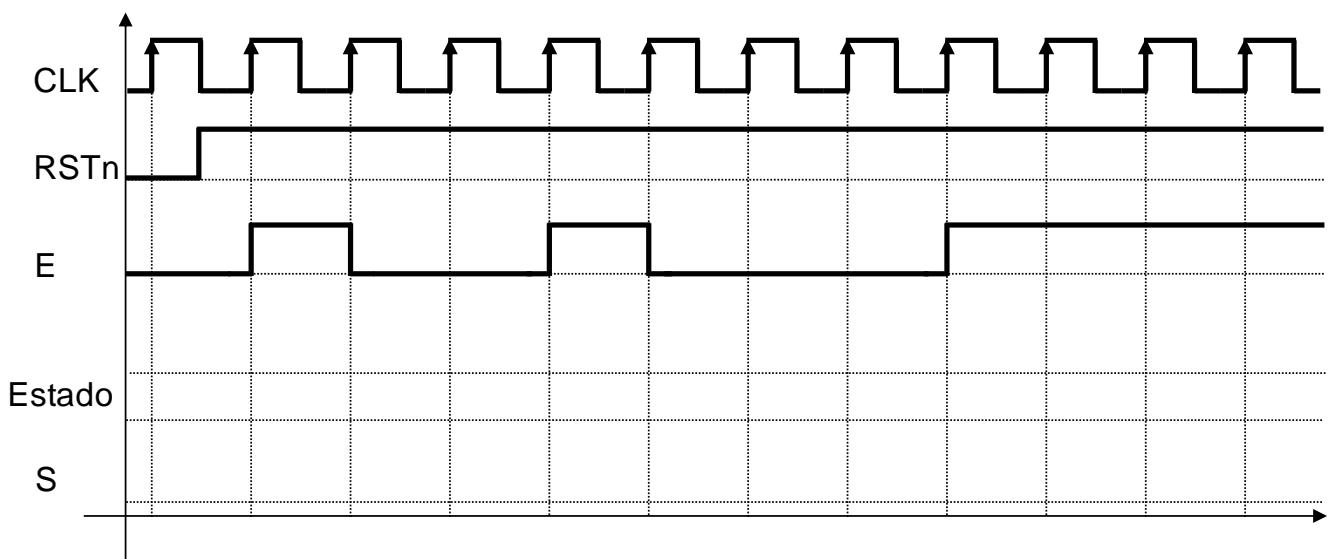
El diagrama de estados (incompleto) de la siguiente figura se corresponde con el circuito del ejercicio anterior. Complete el diagrama de estados especificando las salidas correspondientes a cada estado y las transiciones entre los estados correspondientes a cada entrada.



Ejercicio 3: Análisis del circuito.

El circuito de los ejercicios 1 y 2 detecta la aparición de una secuencia de ceros de una determinada longitud en la entrada “E” y activa su salida “S” cuando esta secuencia se produce.

Analice el circuito y complete el siguiente cronograma:

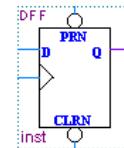
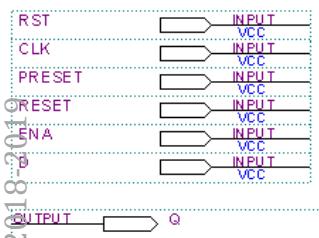


Actividad BT3_ATGP3: Ejercicios sobre sistemas secuenciales.

Ejercicio 4: Flip-flop con entradas síncronas.

Utilizando la lógica adicional que necesite, dote al flip-flop de la figura de:

- Reset asíncrono activo a nivel alto (RST).
- Habilitación de reloj activa a nivel alto (ENA).
- Reset síncrono activo alto, no sujeto a la habilitación de reloj (RESET).
- Preset síncrono activo alto, no sujeto a la habilitación de reloj, prioritario sobre el reset síncrono (PRESET).

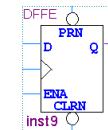
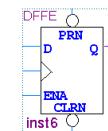
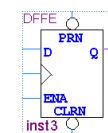
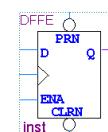
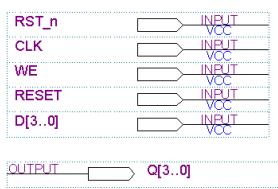


Ejercicio 5: Registro paralelo.

Utilizando flip-flops como los de la figura de la página siguiente y la lógica adicional que necesite, diseñe un registro paralelo de 4 bits con habilitación de escritura y reset síncrono.

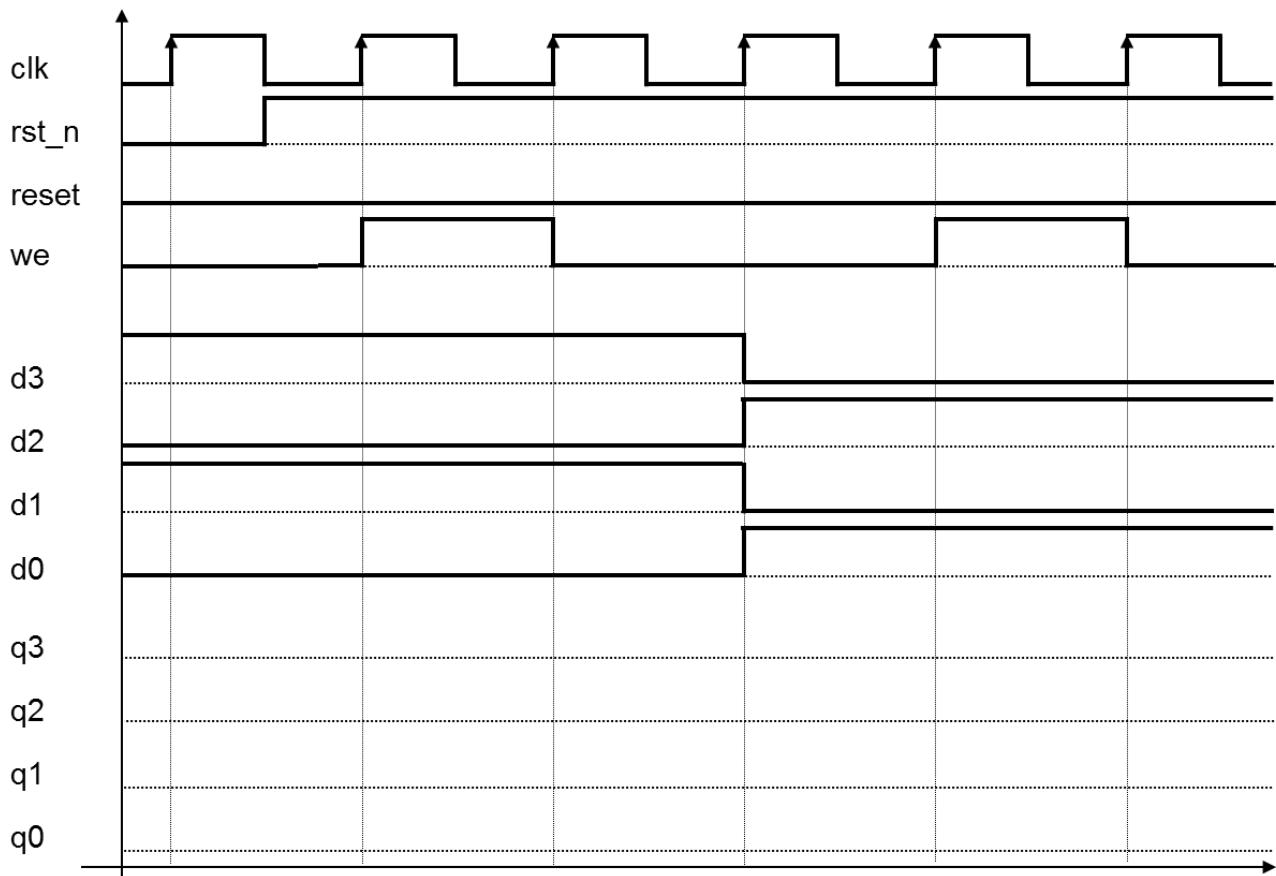
A. Diseño. Diseñe el registro. El funcionamiento debe ser como el que se resume en la siguiente tabla:

Entradas					Salida	Funcionamiento
RSTn	Reset	D _i	WE	CLK	Q _i	
0	X	X	X	X	0	Reset asíncrono
1	X	X	0	↑	q _i	Memoria
1	1	X	1	↑	0	Reset síncrono
1	0	d _i	1	↑	d _i	Captura el dato



Actividad BT3_ATGP3: Ejercicios sobre sistemas secuenciales.

B. Cronograma. Complete el siguiente cronograma:



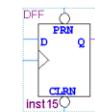
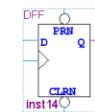
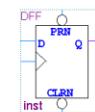
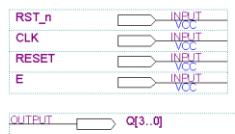
Actividad BT3_ATGP3: Ejercicios sobre sistemas secuenciales.

Ejercicio 6: Registro de desplazamiento.

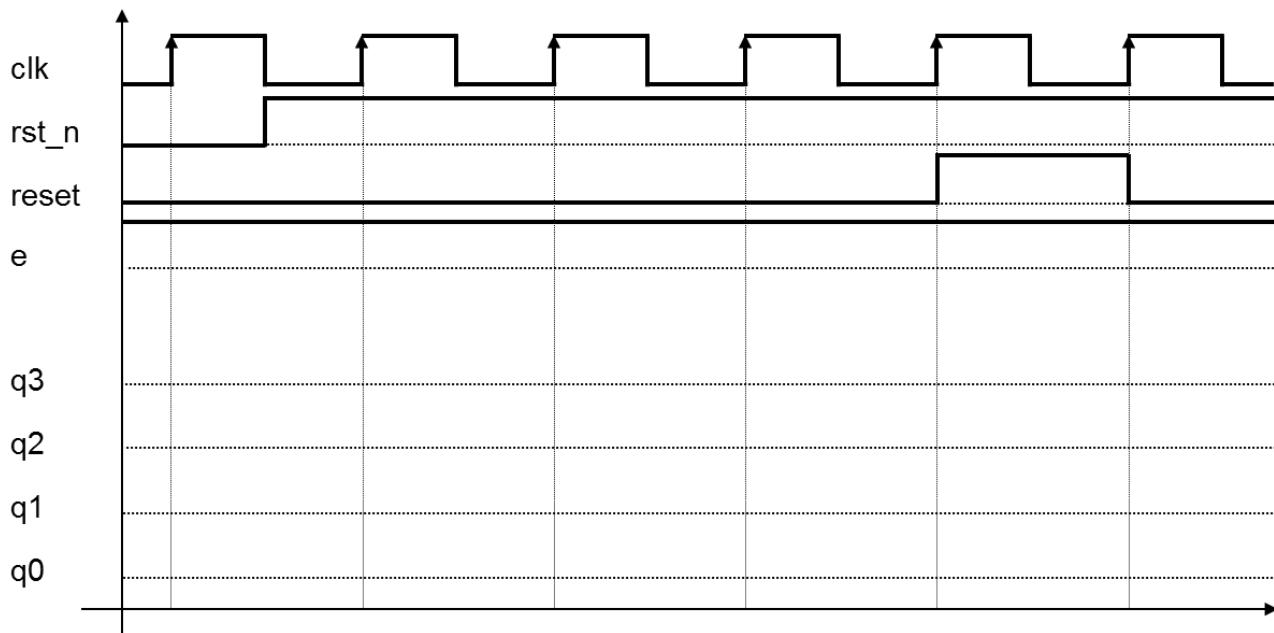
Utilizando flip-flops como los de la figura, diseñe un registro de desplazamiento de 4 bits con reset síncrono, entrada serie y salida paralelo.

A. Diseño. Diseñe el registro. El funcionamiento debe ser como el que se resume en la siguiente tabla:

Entradas				Salida				Funcionamiento
RSTn	E	Reset	CLK	Q ₃	Q ₂	Q ₁	Q ₀	
0	X	X	X	0	0	0	0	Reset asíncrono
1	X	1	↑	0	0	0	0	Reset síncrono
1	e	0	↑	q ₂	q ₁	q ₀	e	Desplazamiento



Cronograma. Complete el siguiente cronograma:



Actividad BT3_ATGP3: Ejercicios sobre sistemas secuenciales.

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Test final sobre ejercicios de sistemas secuenciales.

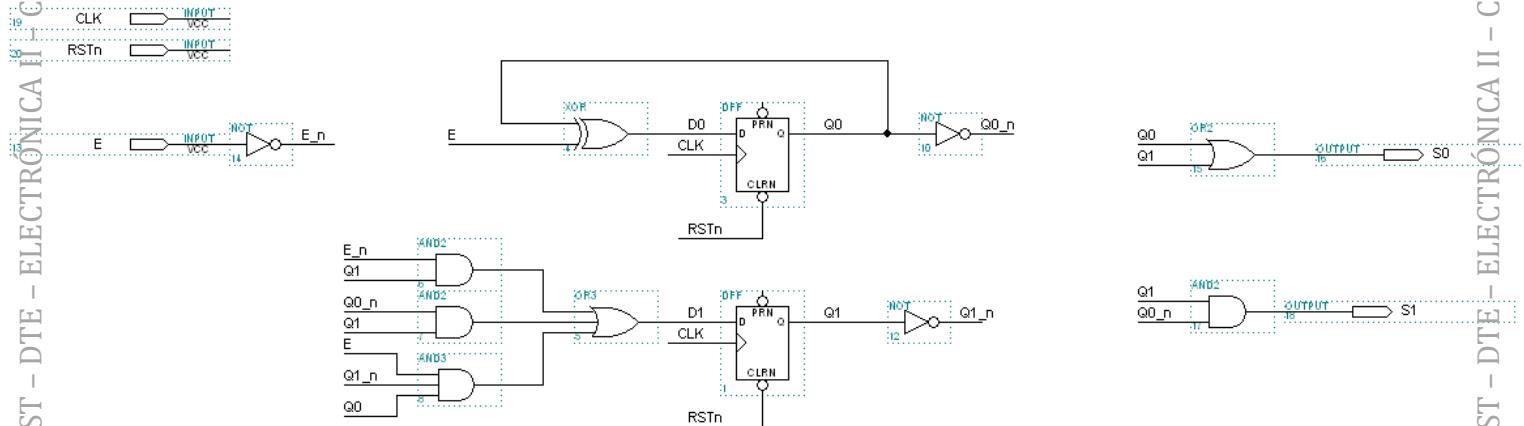
CÓDIGO:
BT3_ATGP3

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

GRUPO		CALIFICACIÓN: [entre 0 y 10 puntos]		puntos
-------	--	--	--	--------

Ejercicio 7: Complete los siguientes apartados relativos al circuito de la figura:



Estructura. Identifique, rodeando con un círculo sobre la figura, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.

A. Cálculo de las salidas y del estado futuro. Obtenga las ecuaciones que relacionan las salidas con el estado actual y el estado futuro con el estado actual y las entradas.

$$S0 =$$

$$S1 =$$

Estado	Código		Salidas	
	Q1	Q0	S1	S0
A	0	0		
B	0	1		
C	1	0		
D	1	1		

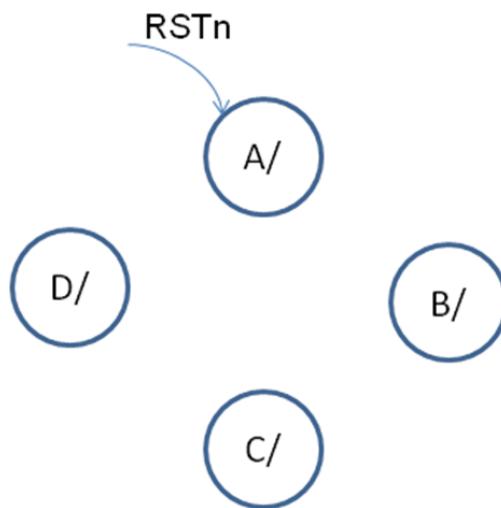
$$D0 =$$

$$D1 =$$

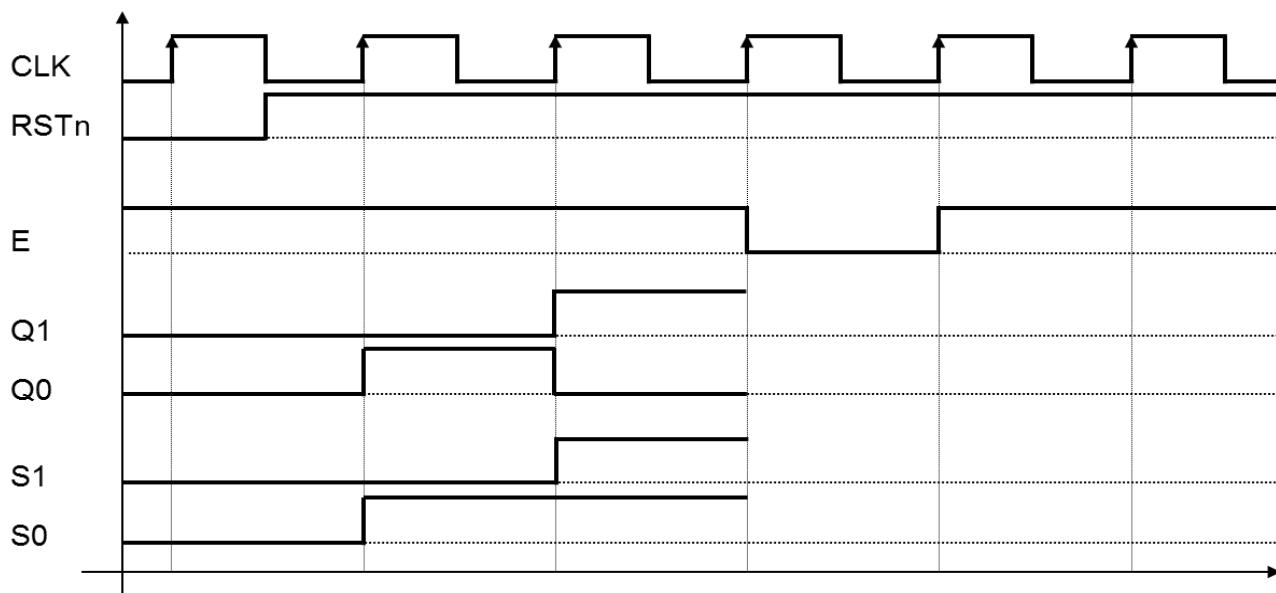
Actividad BT3_ATGP3: Ejercicios sobre sistemas secuenciales.

Entrada	Estado actual		Estado futuro	
	$Q1_t$	$Q0_t$	$D1=Q1_{t+1}$	$D0=Q0_{t+1}$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

- B. **Diagrama de estados.** Complete el diagrama de estados de la siguiente figura añadiendo las salidas correspondientes a cada estado y las transiciones entre estados en función del valor de la entrada.



- C. **Cronograma.** Complete el siguiente cronograma:





ETSiST-UPM

Electrónica 2

Bloque Temático 3

Metodología de diseño de autómatas

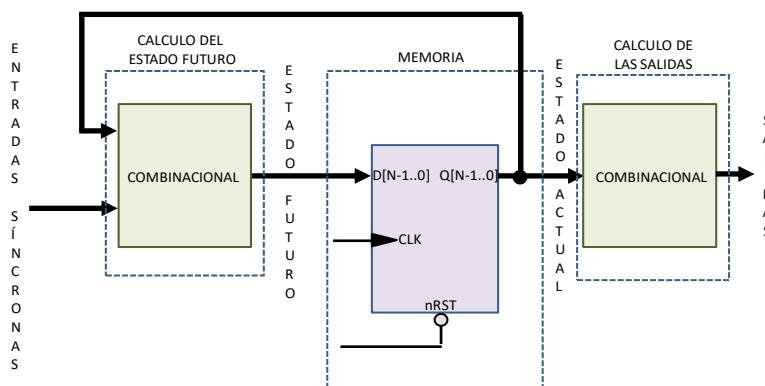
Diseño de Sistemas Secuenciales Síncronos

LA REALIZACIÓN DE UN CIRCUITO SECUENCIAL CONSISTE EN EL DISEÑO DE:

LA MEMORIA DE ESTADO

LA LÓGICA COMBINACIONAL DE CÁLCULO DEL ESTADO DE MEMORIA

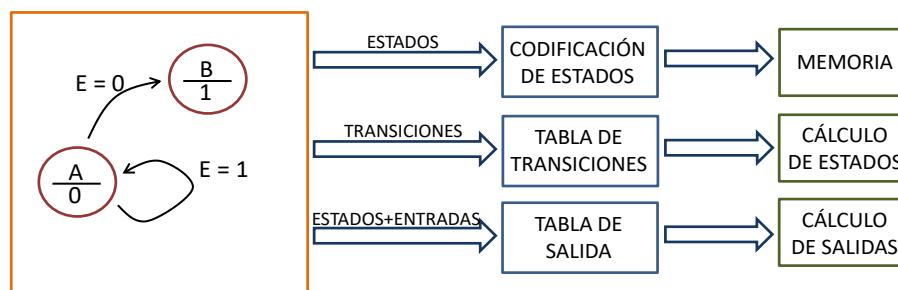
LA LÓGICA COMBINACIONAL DE CÁLCULO DE LAS SALIDAS



Diseño de Sistemas Secuenciales Síncronos

LA METODOLOGÍA SISTEMÁTICA DE DISEÑO DE CIRCUITOS SECUENCIALES UTILIZA LA REPRESENTACIÓN DEL FUNCIONAMIENTO DEL CIRCUITO MEDIANTE UN **DIAGRAMA DE ESTADOS** COMO PUNTO DE PARTIDA PARA EL DISEÑO DE LOS TRES BLOQUES DE LA ARQUITECTURA: **MEMORIA**, **LÓGICA DE CÁLCULO DEL ESTADO DE MEMORIA** Y **LÓGICA DE CÁLCULO DE LAS SALIDAS**

DIAGRAMA DE ESTADOS

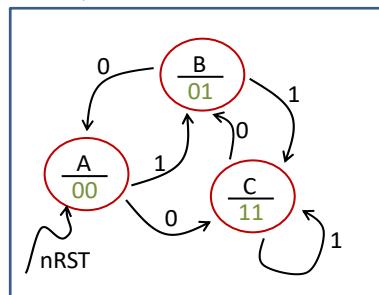


BT3_SS BT3_P7_3

Diseño de Sistemas Secuenciales Síncronos

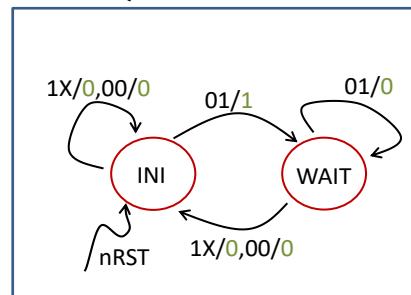
PARA DISEÑAR UN SISTEMA SECUENCIAL SE PARTE DE SU DIAGRAMA DE ESTADOS (DE MOORE O MEALY).

ARQUITECTURA DE MOORE



TRES ESTADOS: A, B Y C
UNA ENTRADA: E
DOS SALIDAS: S₁ Y S₀

ARQUITECTURA DE MEALY

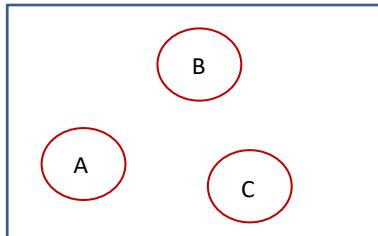


DOS ESTADOS: INI Y WAIT
DOS ENTRADAS: R Y P
UNA SALIDA: F

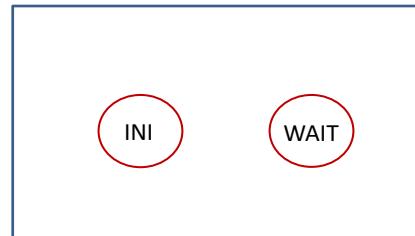
BT3_SS BT3_P7_4

Codificación de Estados (I)

EN PRIMER LUGAR HAY QUE CODIFICAR LOS ESTADOS CON EL **MENOR NÚMERO POSIBLE DE BITS** (SI NO SE ESPECIFICA OTRO CRITERIO)



ESTADO	Q ₁	Q ₀
A	0	0
B	0	1
C	1	0

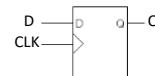
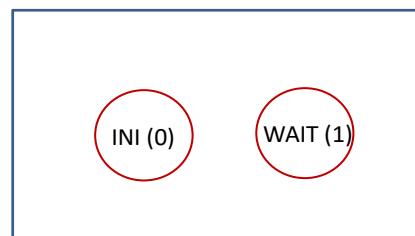
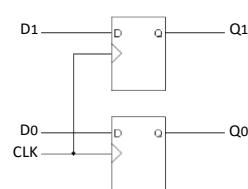
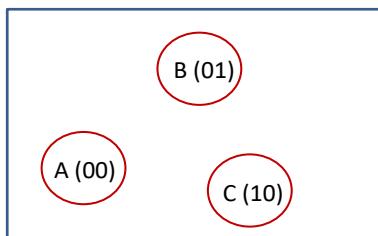


ESTADO	Q
INI	0
WAIT	1

BT3_SS BT3_P7_5

Codificación de Estados (II)

LA **MEMORIA DE ESTADO** DEL CIRCUITO TENDRÁ TANTOS FLIP-FLOPS COMO BITS HAYAN SIDO NECESARIOS PARA CODIFICAR LOS ESTADOS



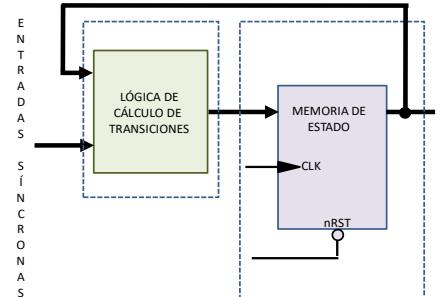
BT3_SS BT3_P7_6

Tabla de Transiciones (I)

A CONTINUACIÓN HAY QUE OBTENER LA TABLA DE TRANSICIONES, QUE ESPECIFICA EL ESTADO DE MEMORIA FUTURO DETERMINADO POR CADA COMBINACIÓN DE ENTRADAS PARA CADA ESTADO DE MEMORIA

TABLA DE TRANSICIONES

ESTADO ACTUAL	ENTRADA	ESTADO FUTURO
A	010	Z
...
...



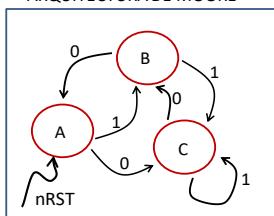
LA TABLA DE TRANSICIONES ES LA TABLA DE VERDAD DE LA LÓGICA DE CONTROL DE LA MEMORIA DE ESTADO: LAS ENTRADAS DE LA TABLA SON EL ESTADO ACTUAL DE MEMORIA (LAS SALIDAS DE LOS FLIP-FLOPS) Y LAS ENTRADAS SÍNCRONAS DEL CIRCUITO; LAS SALIDAS DE LA TABLA DE TRANSICIONES SON EL ESTADO FUTURO DE LA MEMORIA (LAS ENTRADAS DE LOS FLIP-FLOPS).

BT3_SS BT3_P7_7

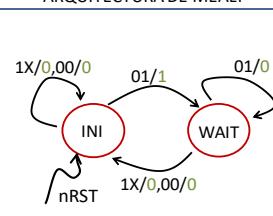
Tabla de Transiciones (II)

LA TABLA DE TRANSICIONES SE OBTIENE A PARTIR DE LAS TRANSICIONES ESPECIFICADAS EN EL DIAGRAMA DE ESTADOS DEL CIRCUITO. EN LA TABLA SIMBÓLICA DE TRANSICIONES LOS ESTADOS SE DENOTAN POR EL NOMBRE QUE TIENEN ASIGNADO EN EL DIAGRAMA DE ESTADOS

ARQUITECTURA DE MOORE



ARQUITECTURA DE MEALY



ESTADO ACTUAL	ENTRADA	ESTADO FUTURO
A	0	C
A	1	B
B	0	A
B	1	C
C	0	B
C	1	C

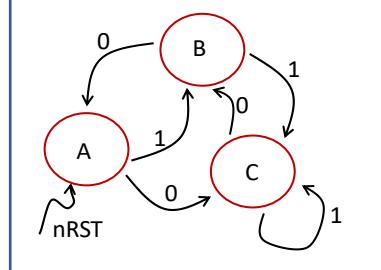
ESTADO ACTUAL	ENTRADA	ESTADO FUTURO
INI	00	INI
INI	01	WAIT
INI	1X	INI
WAIT	00	INI
WAIT	01	WAIT
WAIT	1X	INI

BT3_SS BT3_P7_8

Tabla de Transiciones (III)

LA TABLA DE TRANSICIONES CODIFICADA SE OBTIENE SUSTITUYENDO EL NOMBRE SIMBÓLICO DE LOS ESTADOS POR SU CÓDIGO.

ARQUITECTURA DE MOORE



CODIFICACIÓN DE ESTADOS

A: 00
B: 01
C: 10

TABLA SIMBÓLICA

ESTADO ACTUAL	ENTRADA	ESTADO FUTURO
A	0	C
A	1	B
B	0	A
B	1	C
C	0	B
C	1	C

TABLA CODIFICADA

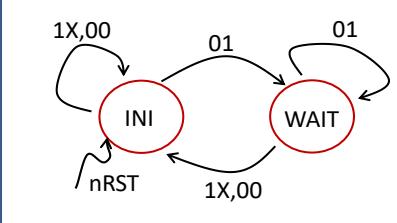
ESTADO ACTUAL	ENTRADA	ESTADO FUTURO		
Q1	Q0		D1	D0
0	0	E	1	0
0	0	0	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0

BT3_SS BT3_P7_9

Tabla de Transiciones (IV)

LA TABLA DE TRANSICIONES CODIFICADA SE OBTIENE SUSTITUYENDO EL NOMBRE SIMBÓLICO DE LOS ESTADOS POR SU CÓDIGO.

ARQUITECTURA DE MEALY



CODIFICACIÓN DE ESTADOS

INI: 0
WAIT: 1

TABLA SIMBÓLICA

ESTADO ACTUAL	ENTRADA	ESTADO FUTURO
INI	00	INI
INI	01	WAIT
INI	1X	INI
WAIT	00	INI
WAIT	01	WAIT
WAIT	1X	INI

TABLA CODIFICADA

ESTADO ACTUAL	R	P	D
0	0	0	0
0	0	1	1
0	1	X	0
1	0	0	0
1	0	1	1
1	1	X	0

BT3_SS BT3_P7_10

Tabla de Transiciones (V)

DE LA TABLA DE TRANSICIONES CODIFICADA SE OBTIENEN LAS FUNCIONES LÓGICAS QUE DEFINEN EL ESTADO FUTURO (LAS ENTRADAS DE LOS FLIP-FLOPS) A PARTIR DEL ESTADO ACTUAL (LAS SALIDAS DE LOS FLIP-FLOPS) Y DE LAS ENTRADAS SÍNCRONAS DEL CIRCUITO.

ESTADO ACTUAL		ENTRADA	ESTADO FUTURO	
Q1	Q0	E	D1	D0
0	0	0	1	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0

$$\begin{cases} D_1 = \overline{Q_1 + Q_0} \oplus E \\ D_0 = \overline{Q_0} \cdot (E \oplus Q_1) \end{cases}$$

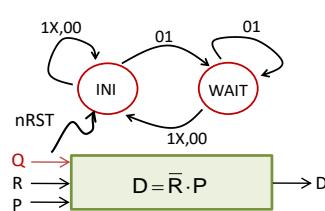
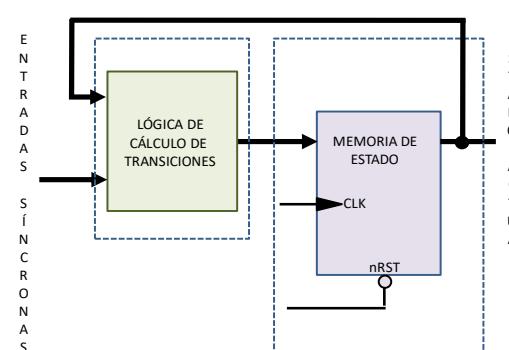
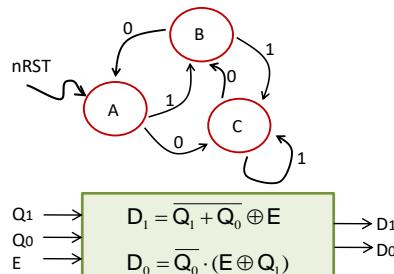
ESTADO ACTUAL		ENTRADAS	ESTADO FUTURO
Q		R P	D
0		0 0	0
0		0 1	1
0		1 X	0
1		0 0	0
1		0 1	1
1		1 X	0

$$D = \overline{R} \cdot P$$

BT3_SS BT3_P7_11

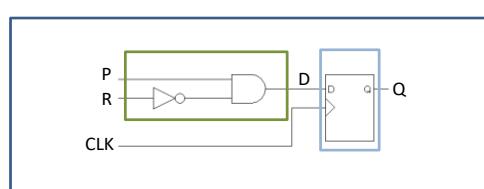
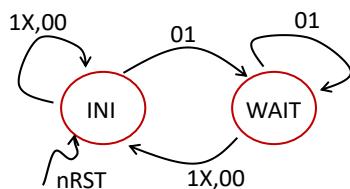
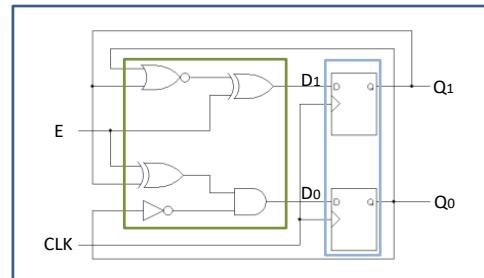
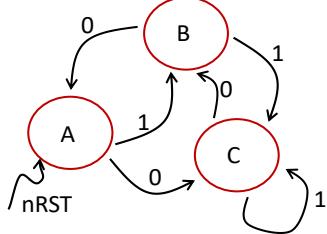
Lógica de Control de Transiciones (I)

LAS ECUACIONES EXTRAÍDAS DE LA TABLA DE TRANSICIONES CODIFICADA DEFINEN EL CIRCUITO COMBINACIONAL QUE CONTROLA LOS CAMBIOS DE ESTADO DE MEMORIA



BT3_SS BT3_P7_12

Lógica de Control de Transiciones (II)

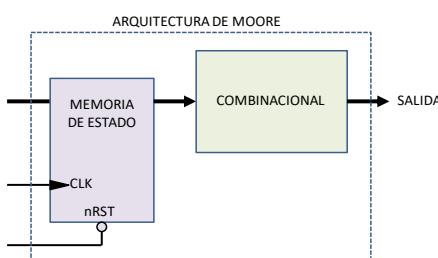


BT3_SS BT3_P7_13

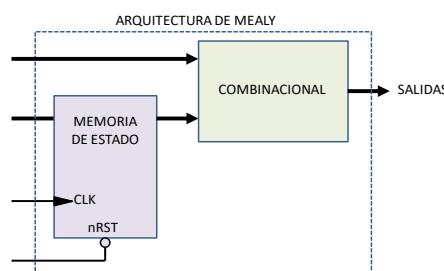
Tabla de Salida (I)

PARA TERMINAR EL DISEÑO DEL CIRCUITO SECUENCIAL HAY QUE OBTENER LA TABLA DE SALIDA.

EL PROCEDIMIENTO PARA LA OBTENCIÓN DE LA TABLA DE SALIDA DEPENDE DE SI LA ARQUITECTURA DEL CIRCUITO ES DE MOORE O DE MEALY



EN LOS AUTÓMATAS DE **MOORE** LAS SALIDAS DEPENDE SÓLO DEL ESTADO DE MEMORIA

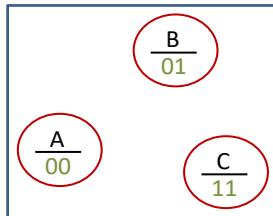


EN LOS AUTÓMATAS DE **MEALY** LAS SALIDAS DEPENDE DEL ESTADO DE MEMORIA Y DE LAS ENTRADAS

BT3_SS BT3_P7_14

Tabla de Salida (II)

EN LOS AUTÓMATAS DE MOORE LA TABLA DE SALIDA ESPECIFICA EL VALOR DE SALIDA ASOCIADO A CADA ESTADO

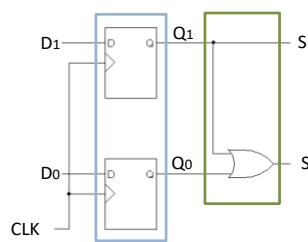
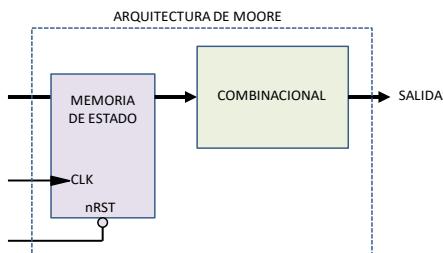


ESTADO	SALIDA
A	00
B	01
C	11

$$\begin{aligned} A &: 00 \\ B &: 01 \\ C &: 10 \end{aligned}$$

$S_1 = Q_i$
 $S_0 = Q_i + Q_0$

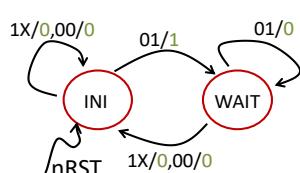
ESTADO	SALIDA
Q ₁	Q ₀
S ₁	S ₀
0	0
0	1
1	0
1	1



BT3_S5 BT3_P7_15

Tabla de Salida (III)

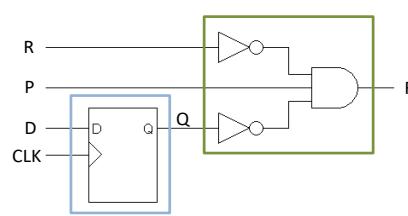
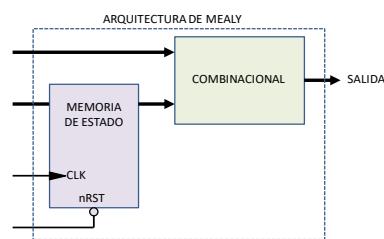
EN LOS AUTÓMATAS DE MEALY LA TABLA DE SALIDA ESPECIFICA EL VALOR DE SALIDA ASOCIADO A CADA COMBINACIÓN DE ESTADO ACTUAL Y ENTRADA



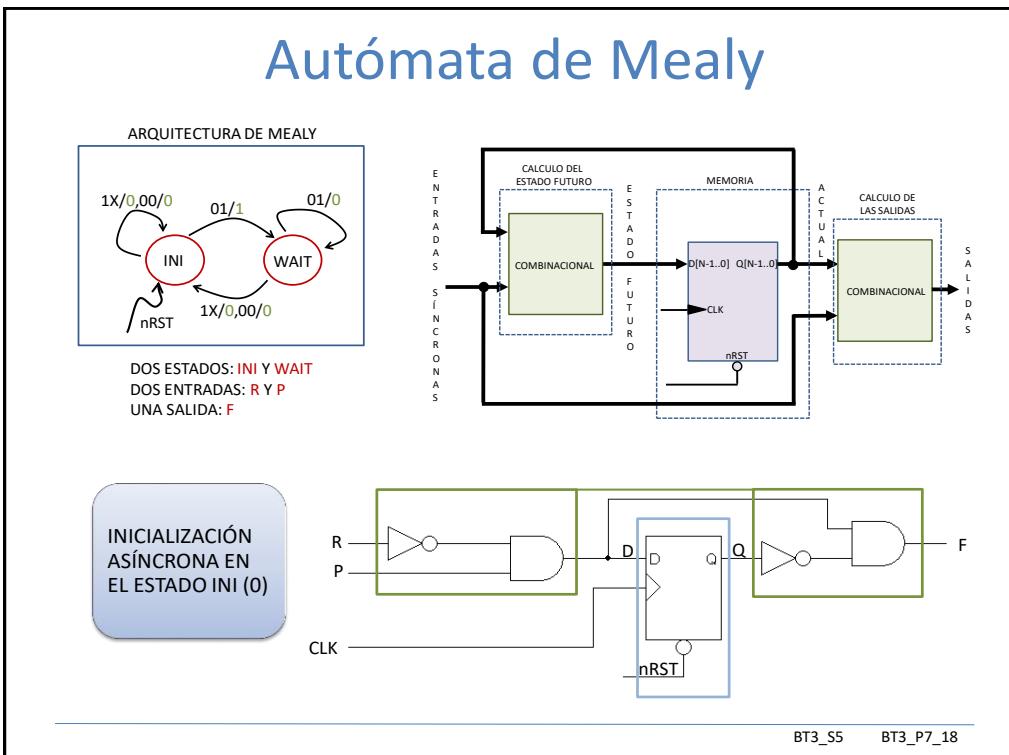
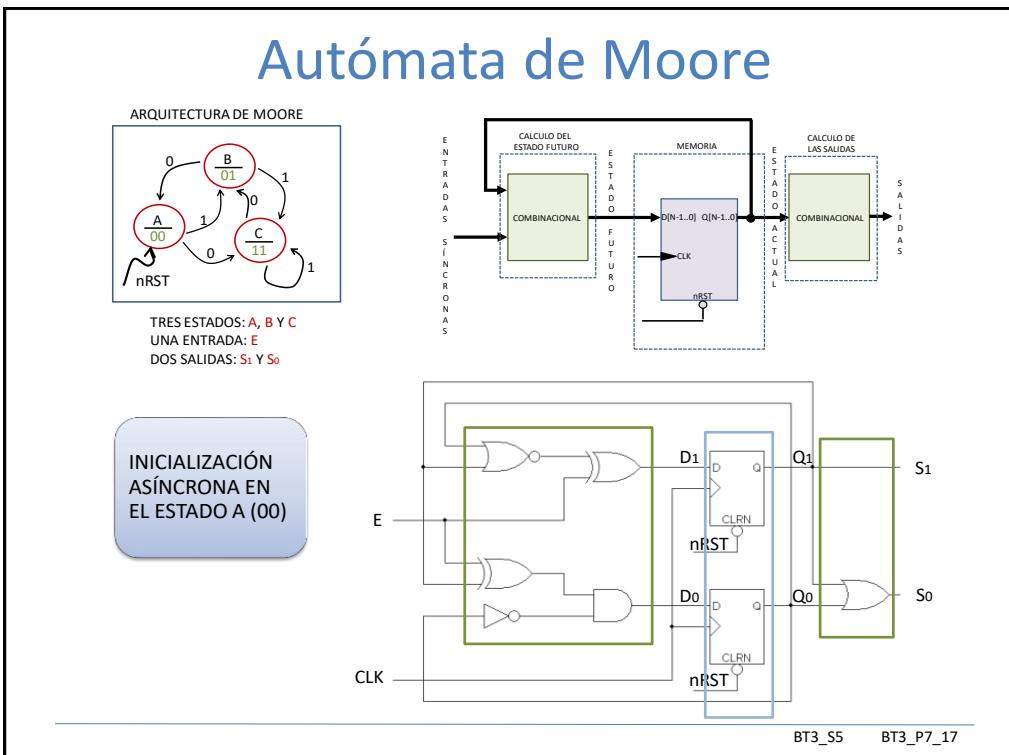
ESTADO	ENTRADA	SALIDA
INI	00	0
INI	01	1
INI	1X	0
WAIT	00	0
WAIT	01	0
WAIT	1X	0

ESTADO	R	P	F
0	0	0	0
0	0	1	1
0	1	X	0
1	0	0	0
1	0	1	0
1	1	X	0

$$F = \bar{Q} \cdot \bar{R} \cdot P$$



BT3_S5 BT3_P7_16



Esta página se ha dejado intencionadamente en blanco



ETSiST-UPM

Electrónica 2

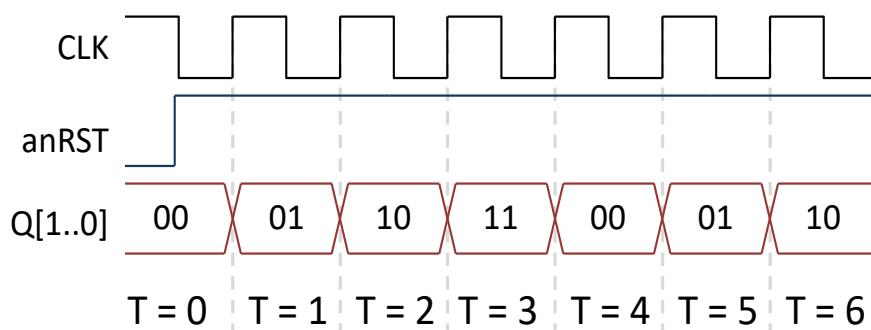
Bloque Temático 4

Ejemplo de autómatas: introducción a contadores

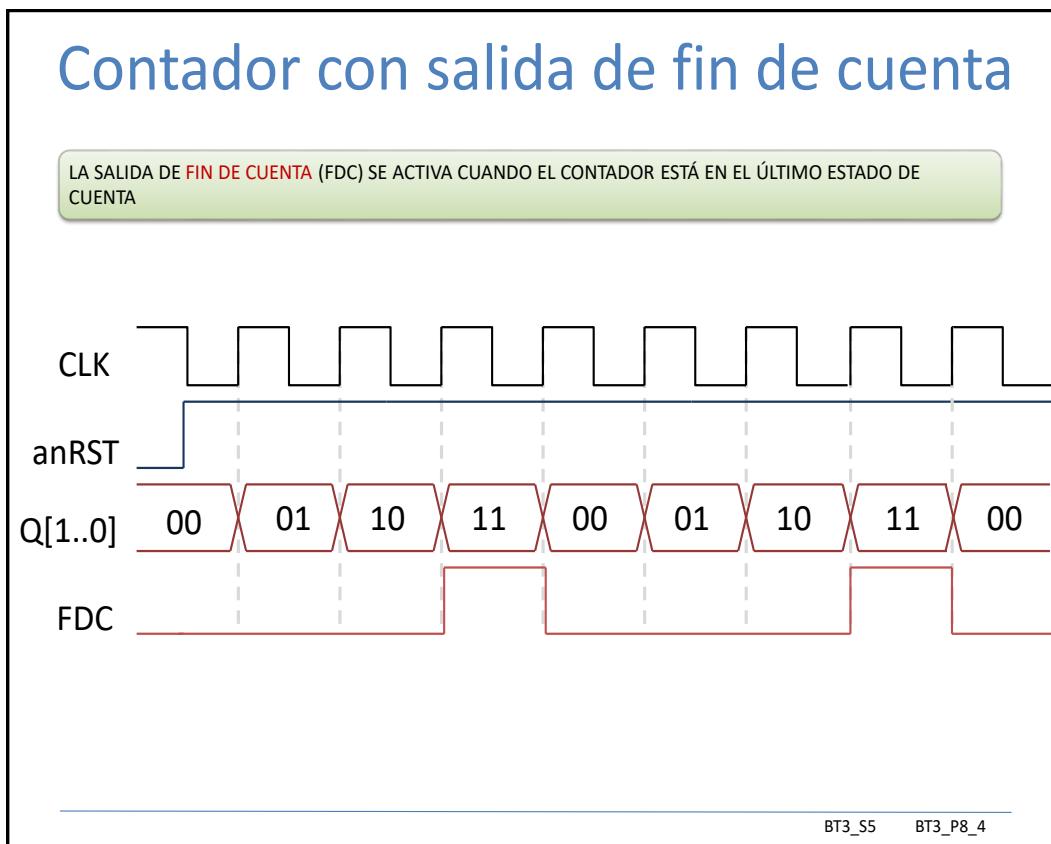
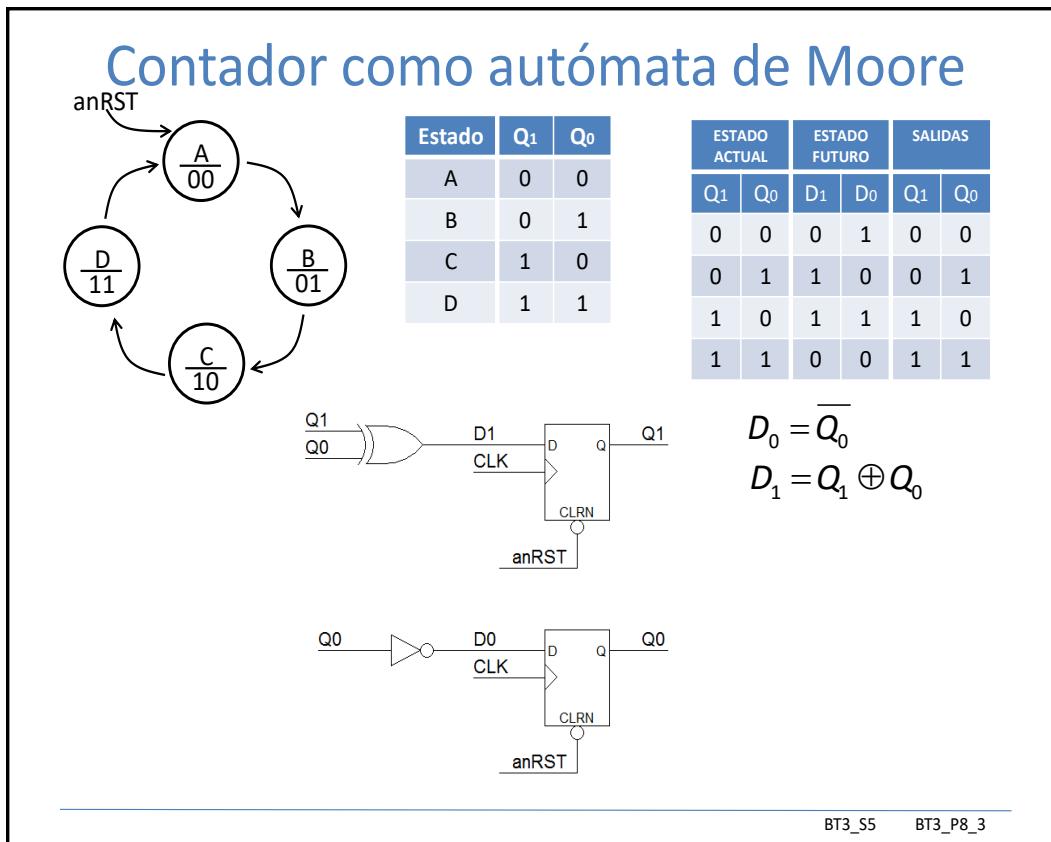
Contador binario ascendente

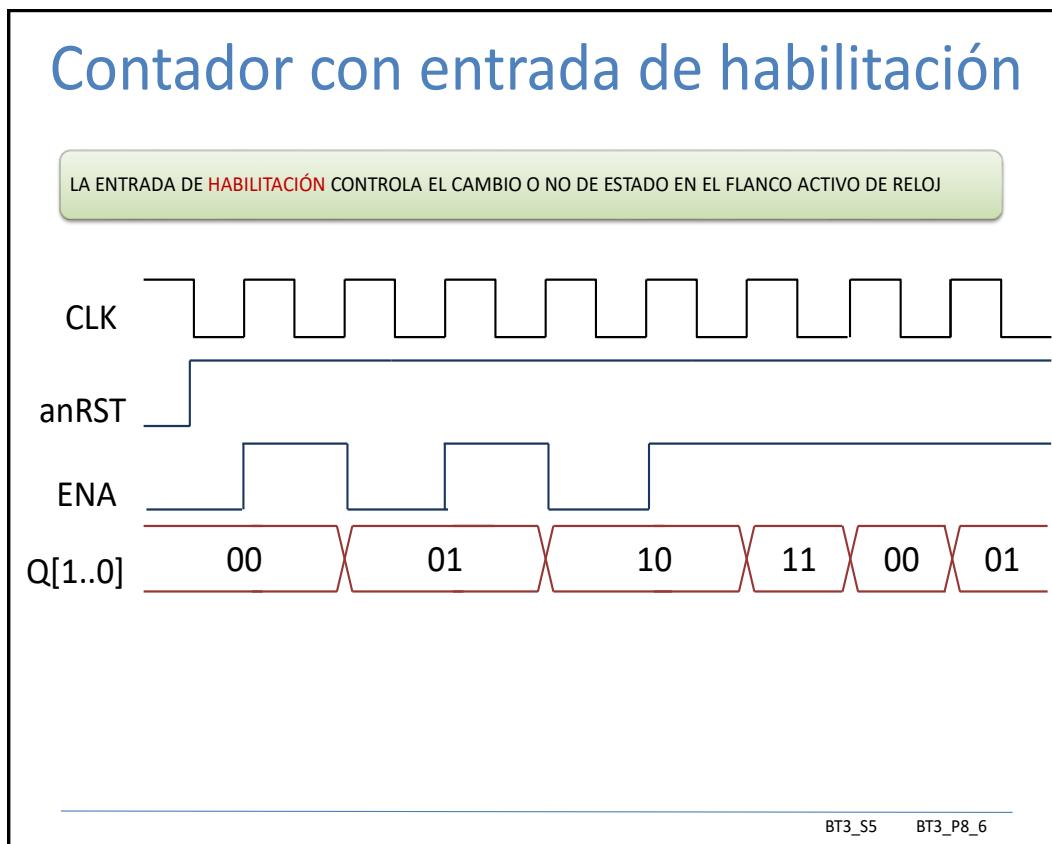
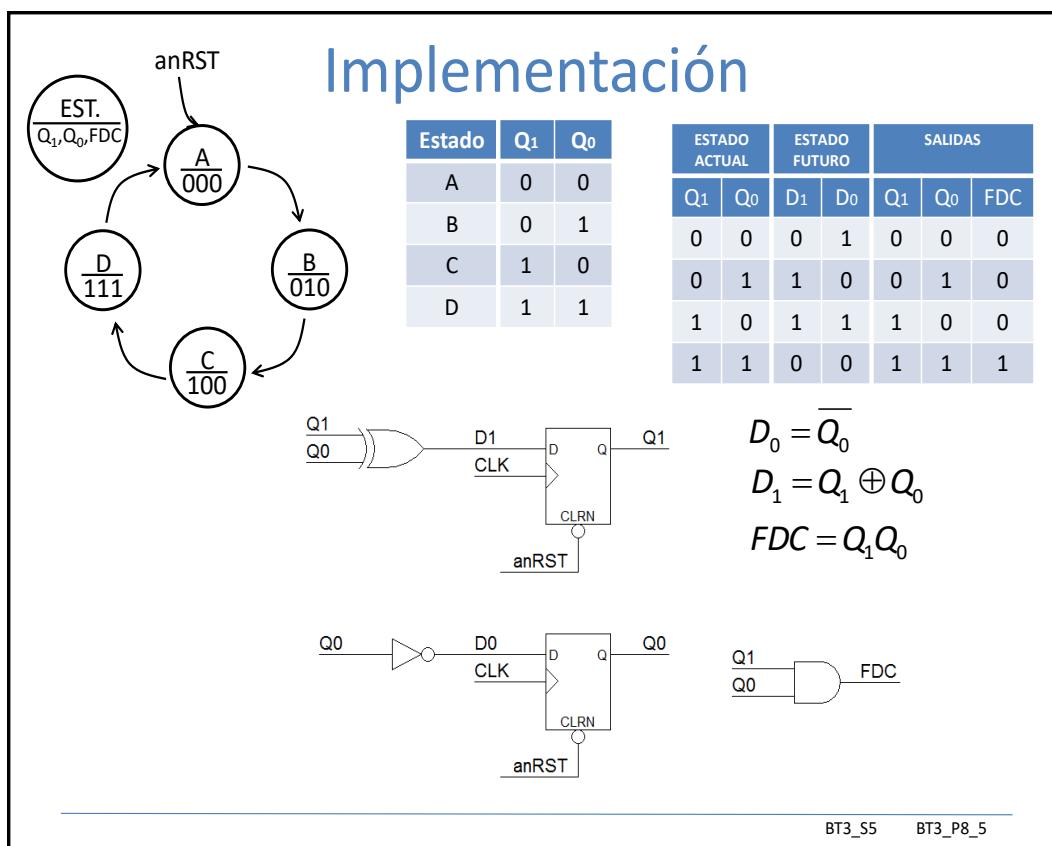
UN CONTADOR ES UN AUTÓMATA QUE RECORRE CÍCLICAMENTE UNA SECUENCIA DE ESTADOS. UN CONTADOR BINARIO ASCENDENTE ES AQUEL EN EL QUE LAS SALIDAS PARA LOS SUCESIVOS ESTADOS CORRESPONDEN CON UNA CUENTA ASCENDENTE EN BINARIO NATURAL.

EL NÚMERO DE ESTADOS DE LA SECUENCIA RECORRIDO POR EL CONTADOR ES EL **MÓDULO** DEL MISMO



BT3_SS5 BT3_P8_2





Implementación

The state transition diagram shows four states: A (00), B (01), C (10), and D (11). State A is the initial state. Transitions are as follows: A to B on 1, B to C on 1, C to D on 1, D to A on 1, A to C on 0, and C to B on 0. There is also a self-loop on state A for input 0. An asynchronous reset input *anRST* can return any state to 0.

Estado	<i>Q₁</i>	<i>Q₀</i>
A	0	0
B	0	1
C	1	0
D	1	1

ESTADO ACTUAL		ENTRADA		ESTADO FUTURO		SALIDAS	
<i>Q₁</i>	<i>Q₀</i>	ENA		<i>D₁</i>	<i>D₀</i>	<i>Q₁</i>	<i>Q₀</i>
0	0	0		0	0	0	0
0	0	1		0	1	0	0
0	1	0		0	1	0	1
0	1	1		1	0	0	1
1	0	0		1	0	1	0
1	0	1		1	1	1	0
1	1	0		0	1	1	1
1	1	1		1	0	1	1

The logic circuit diagram shows the implementation of the state transitions. It uses AND gates to generate the enable signals *ENAn* and *ENAn*, and an OR gate to generate *D1*. The output *D1* is fed into a D flip-flop with clock *CLK* and clear *CLRN* (active low) to update *Q1*. Similarly, the output *D0* is generated by an AND gate and fed into a D flip-flop with clock *CLK* and clear *CLRN* to update *Q0*. Asynchronous reset inputs *anRST* are connected to both flip-flops.

D₁ = $Q_1 \overline{Q_0} + Q_1 \overline{ENA} + \overline{Q_1} Q_0 ENA$
D₀ = $ENA \oplus Q_0$

BT3_S5 BT3_P8_7

Aplicación (I): Contador de pulsos

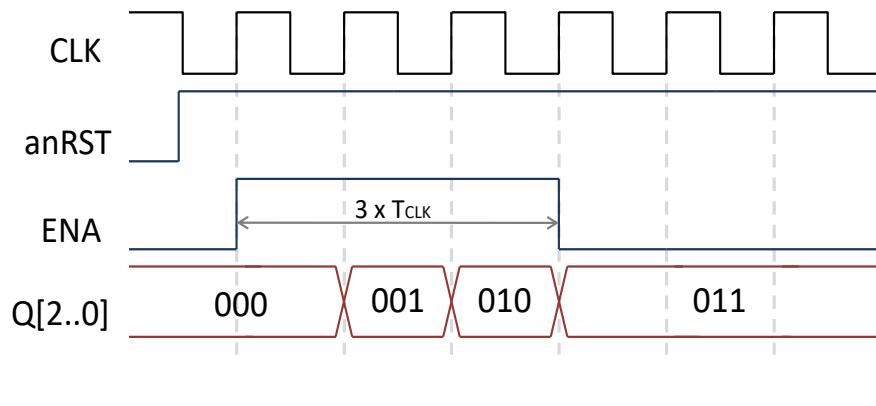
SI LA ENTRADA DE HABILITACIÓN, ENA, RECIBE PULSOS CON UNA DURACIÓN DE UN CICLO DE RELOJ, EL CONTADOR CUENTA EL NÚMERO DE PULSOS RECIBIDOS.

The timing diagram illustrates the operation of the counter. The *CLK* signal is a square wave. The *anRST* signal is asserted at the start of the first cycle and cleared during the second cycle. The *ENA* signal is asserted during the second and third cycles. The output *Q[2..0]* shows the binary count: 000, 001, and 010. Vertical dashed lines indicate the rising edges of *CLK* where the count changes.

BT3_S5 BT3_P8_8

Aplicación (II): Contador de ciclos de reloj

SI LA ENTRADA DE HABILITACIÓN, ENA, RECIBE UN ÚNICO PULSO DE DURACIÓN **N CICLOS DE RELOJ**, EL CONTADOR CUENTA LA DURACIÓN DEL PULSO EN CICLOS DE RELOJ.



BT3_SS BT3_P8_9

Esta página se ha dejado intencionadamente en blanco

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Ejercicios de diseño de autómatas.

CÓDIGO:
BT3_ATGP4

GRUPO		FECHA:	
-------	--	--------	--

MODALIDAD:	Aprendizaje Cooperativo	TIPO:	Presencial	DURACIÓN:	50 minutos
------------	-------------------------	-------	------------	-----------	------------

CALENDARIO:	Quinta Sesión	REQUISITOS:	
-------------	---------------	-------------	--

CRITERIO DE ÉXITO:	Realización correcta de los seis ejercicios que se proponen.
--------------------	--

ROLES:	No hay distribución diferenciada de roles entre los miembros
--------	--

HABILIDADES SOCIALES:	Capacidad para expresar y sintetizar ideas, capacidad para escuchar, asimilar y organizar la información presentada por otra persona. Capacidad para razonar y analizar.
-----------------------	--

REFLEXIÓN DE GRUPO:	
---------------------	--

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	
------------------	------------	-------------------------------------	--

Descripción de la actividad

Esta actividad se compone de tres partes. Cada parte consta de 2 ejercicios de un mismo tipo. Los miembros del grupo formarán dos subgrupos. Cada subgrupo resolverá un ejercicio de cada parte. Una vez resueltos los problemas de la misma parte, los dos subgrupos comentarán los resultados a fin de que todos los integrantes del grupo tengan un conocimiento pleno de ambos ejercicios.

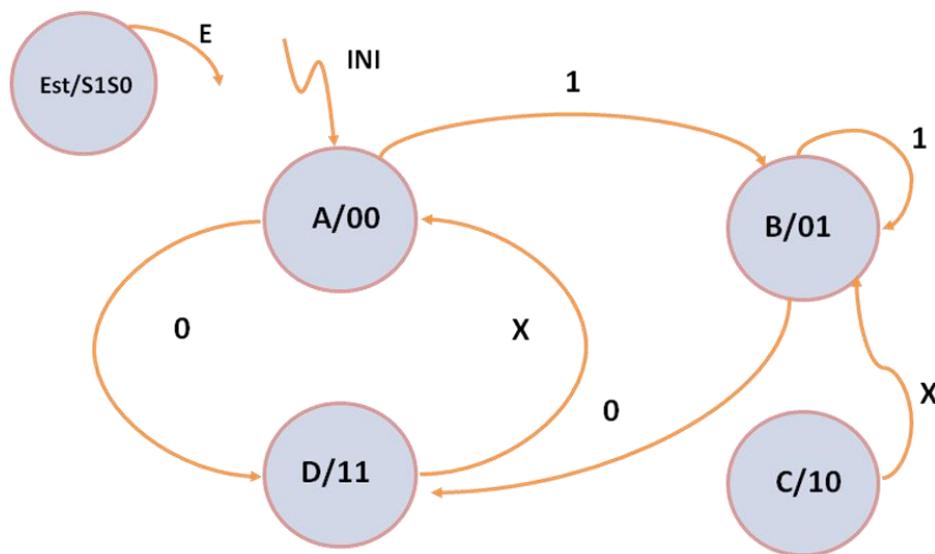
La distribución de tiempos para la realización de la actividad es la siguiente:

Parte	Realización	Discusión
I	10'	6'
II	10'	6'
III	10'	6'

Ejercicios de diseño de autómatas

PARTE I. Ejercicios de diseño de autómatas de Moore

Ejercicio 1. A partir del siguiente diagrama de estados



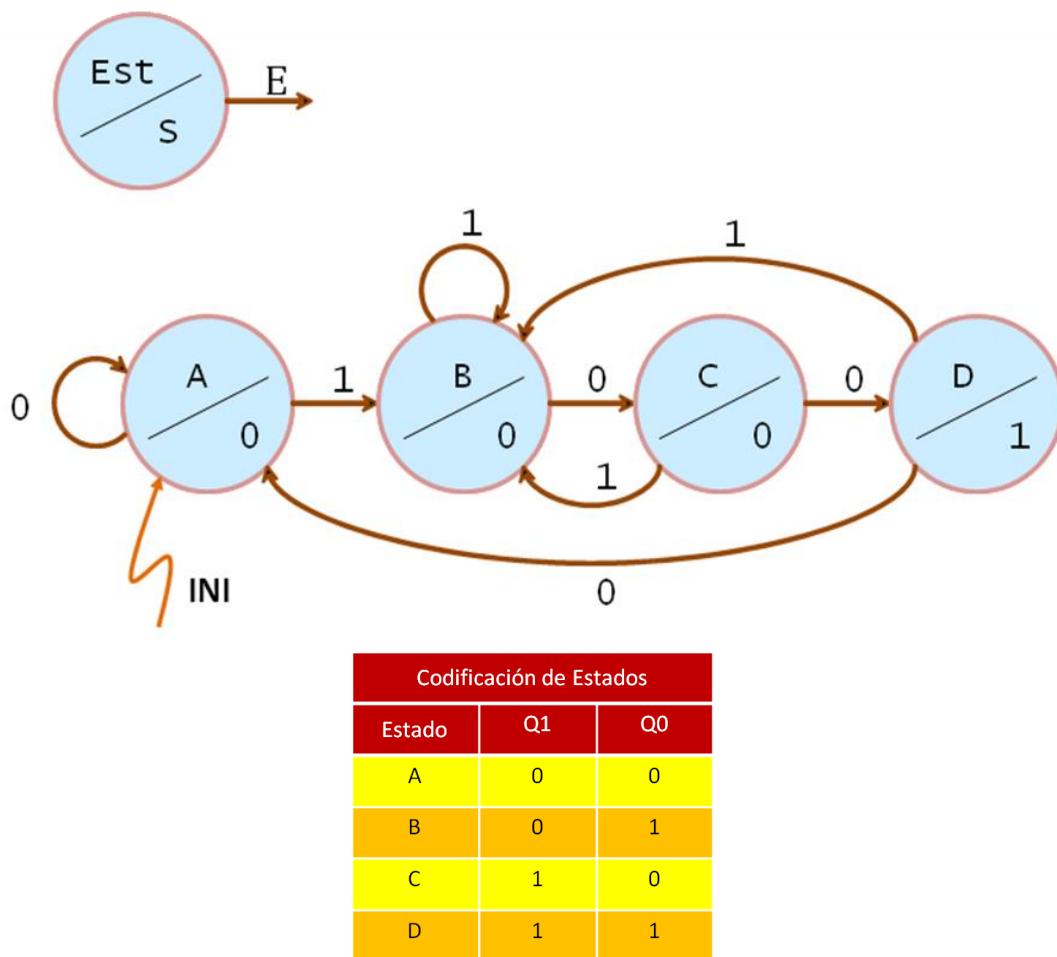
Codificación de Estados		
Estado	Q1	Q0
A	0	0
B	0	1
C	1	0
D	1	1

Se pide:

- a) Realizar las tablas simbólicas de transiciones y de salidas.

- b) A partir de la codificación de estados indicada en la tabla realizar las tablas codificadas de transiciones y de salidas.
- c) Expresar las funciones booleanas necesarias para generar el estado futuro y las salidas.
- d) Dibujar el circuito que implementa el diagrama de estados de partida.

Ejercicio 2. A partir del siguiente diagrama de estados



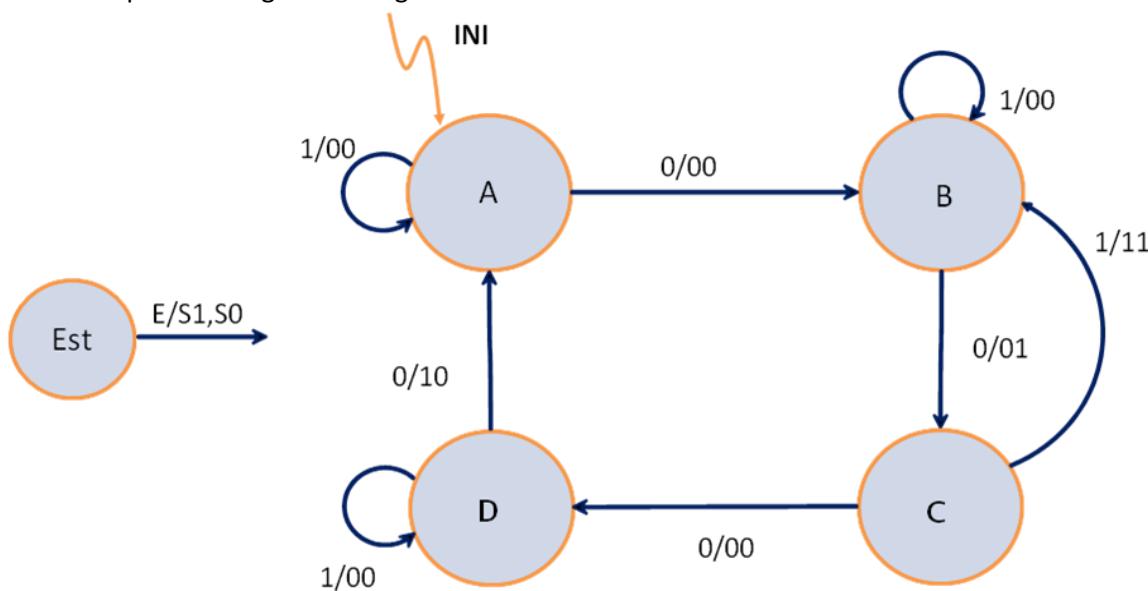
Se pide:

- a) Realizar las tablas simbólicas de transiciones y de salidas.

- b) A partir de la codificación de estados indicada en la tabla realizar las tablas codificadas de transiciones y de salidas.

- c) Expresar las funciones booleanas necesarias para generar el estado futuro y las salidas.

- d) Dibujar el circuito que implementa el diagrama de estados de partida.

PARTE II. Ejercicios de diseño de autómatas de Mealy**Ejercicio 1.** A partir del siguiente diagrama de estados

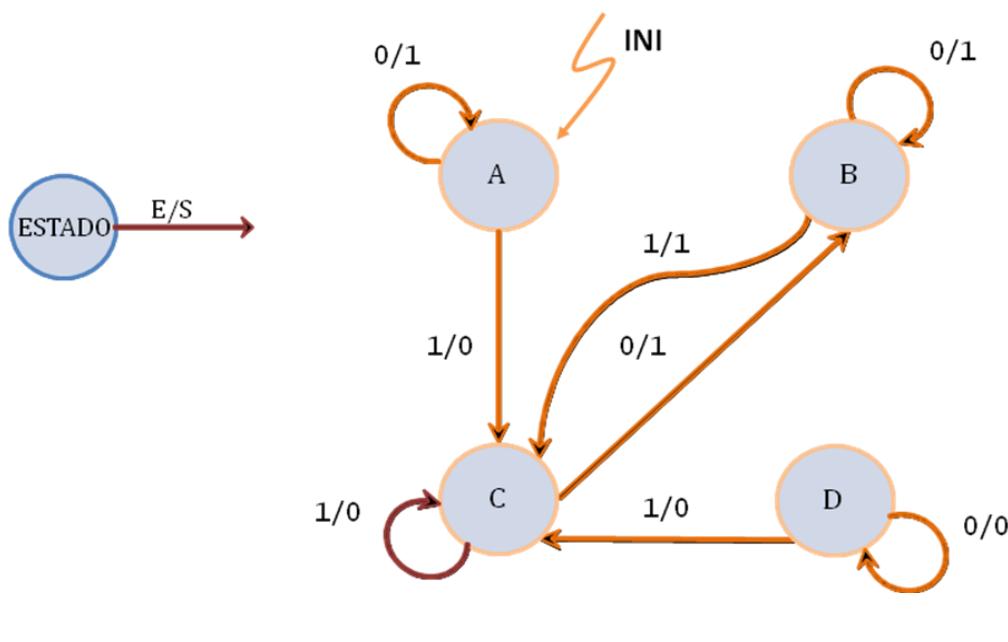
Codificación de Estados		
Estado	Q1	Q0
A	0	0
B	0	1
C	1	0
D	1	1

Se pide:

- a) Realizar las tablas simbólicas de transiciones y de salidas.

- b) A partir de la codificación de estados indicada en la tabla realizar las tablas codificadas de transiciones y de salidas.
- c) Expresar las funciones booleanas necesarias para generar el estado futuro y las salidas.
- d) Dibujar el circuito que implementa el diagrama de estados de partida.

Ejercicio 2. A partir del siguiente diagrama de estados



Se pide:

- a) Realizar las tablas simbólicas de transiciones y de salidas.

- b) A partir de la codificación de estados indicada en la tabla realizar las tablas codificadas de transiciones y de salidas.
- c) Expresar las funciones booleanas necesarias para generar el estado futuro y las salidas.
- d) Dibujar el circuito que implementa el diagrama de estados de partida.

PARTE III. Ejercicios de realización de diagramas de estado

Ejercicio 1. Realizar el diagrama de estados de un contador binario descendente de módulo ocho con entrada de habilitación de reloj activa a nivel alto.

Ejercicio 2. Realizar el diagrama de estados de un contador BCD de un dígito con una entrada que indica si la cuenta se realiza en sentido ascendente o descendente.

BLOQUE TEMÁTICO 3**TÍTULO DE LA ACTIVIDAD:****Realización de ejercicios aplicando la metodología de diseño de autómatas.****CÓDIGO:****BT3_AINP5**

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	40 minutos
------------	------------------------	-------	---------------	-----------	------------

CALENDARIO:	Quinta Sesión Previo a la Sexta Sesión	REQUISITOS:	Metodología de diseño de autómatas.
-------------	---	-------------	-------------------------------------

CRITERIO DE ÉXITO:	Realización correcta de los 2 ejercicios
--------------------	--

COMENTARIOS E INCIDENCIAS:

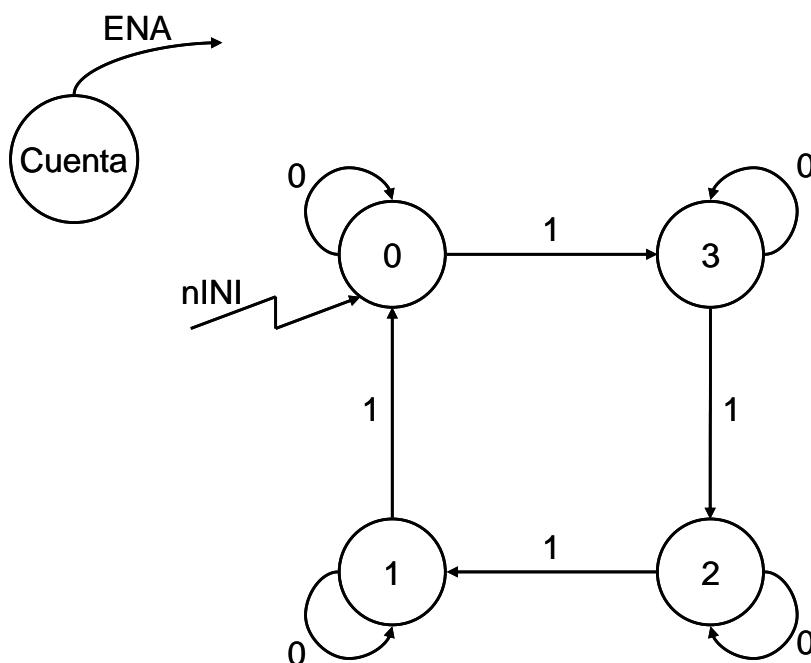
TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	Autocorrección
------------------	---------	--	----------------

REALIZACIÓN DE EJERCICIOS SOBRE LA METODOLOGÍA DE DISEÑO DE AUTÓMATAS

PARTE I. Enunciado de los ejercicios

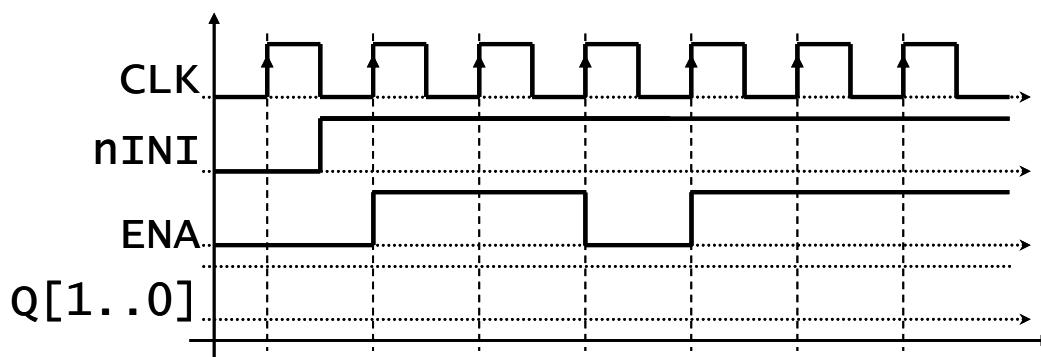
En lo que sigue se presentan algunos enunciados de ejercicios que deberán ser resueltos sin consultar la solución, tratando de poner en práctica los conocimientos sobre la metodología de diseño de autómatas expuesta en clase. Posteriormente se tendrá la oportunidad de contrastar la solución con la que se ofrece en la segunda parte de este documento y detectar posibles errores.

EJERCICIO 1) El diagrama de estados de la figura representa el funcionamiento de un contador descendente de 2 bits con entrada de habilitación de reloj activa a nivel alto.

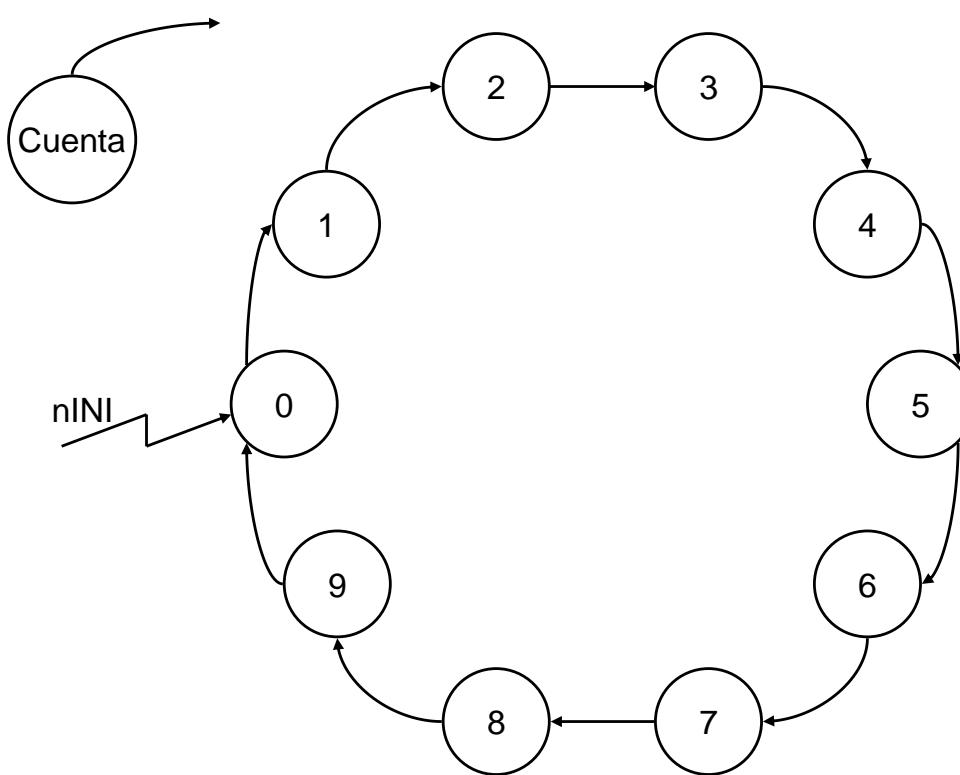


A partir de este diagrama de estados, aplique la metodología de diseño de circuitos secuenciales y diseñe el circuito correspondiente a dicho contador.

Represente en el siguiente cronograma el funcionamiento del contador descendente de 2 bits.

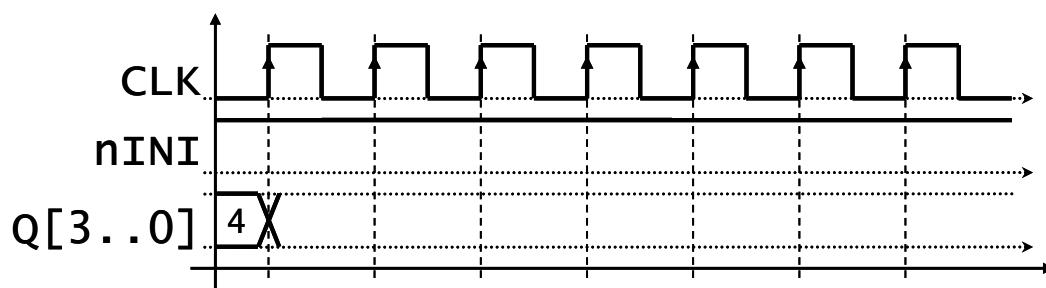


EJERCICIO 2) El diagrama de estados de la figura representa el funcionamiento de un contador decimal ascendente.



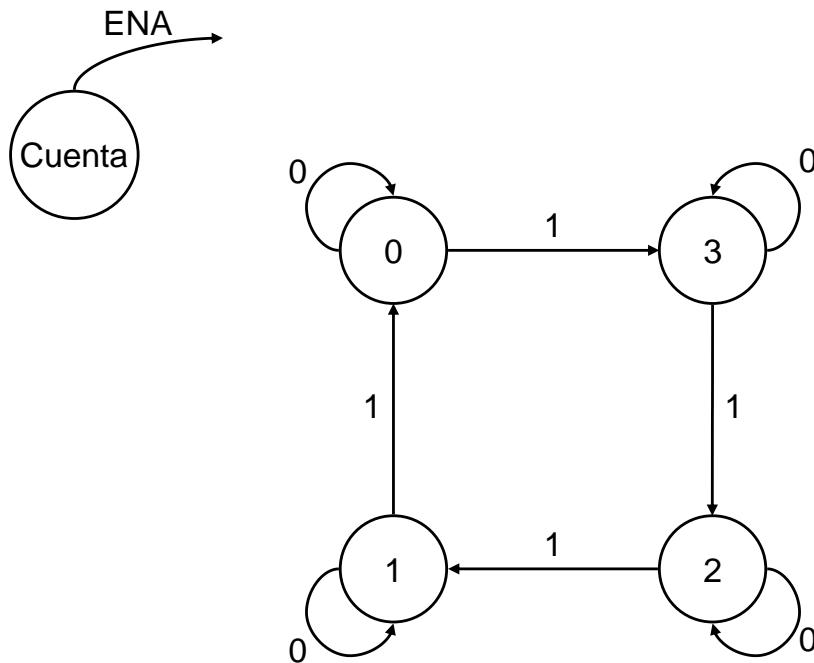
A partir de este diagrama de estados, aplique la metodología de diseño de circuitos secuenciales y diseñe el circuito correspondiente a dicho contador.

Represente en el siguiente cronograma el funcionamiento del contador decimal ascendente.



PARTE II. Soluciones de los ejercicios

EJERCICIO 1) El diagrama de estados de la figura representa el funcionamiento de un contador descendente de 2 bits con entrada de habilitación de reloj activa a nivel alto.



A partir de este diagrama de estados, aplique la metodología de diseño de circuitos secuenciales y diseñe el circuito correspondiente a dicho contador.

Codificación de los estados

ESTADO	Q1	Q0
0	0	0
1	0	1
2	1	0
3	1	1

Memoria de estado

El tener 4 estados son necesarios 2 flip-flops tipo D ($2 \text{ estados} \leq 2^2 \text{ flip-flops}$).

Tabla simbólica de transiciones

ESTADO ACTUAL	ENTRADA	ESTADO FUTURO
0	0	0
0	1	3
1	0	1
1	1	0
2	0	2
2	1	1
3	0	3
3	1	2

Tabla de transiciones codificada

ESTADO ACTUAL		ENTRADA	ESTADO FUTURO	
Q1	Q0	ENA	D1	D0
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	0

Funciones lógicas que definen el estado futuro

$$D1 = \overline{Q1} \overline{Q0} ENA + Q1 \overline{Q0} \overline{ENA} + Q1 Q0$$

$$D0 = Q0 \oplus ENA$$

Lógica de control de transiciones

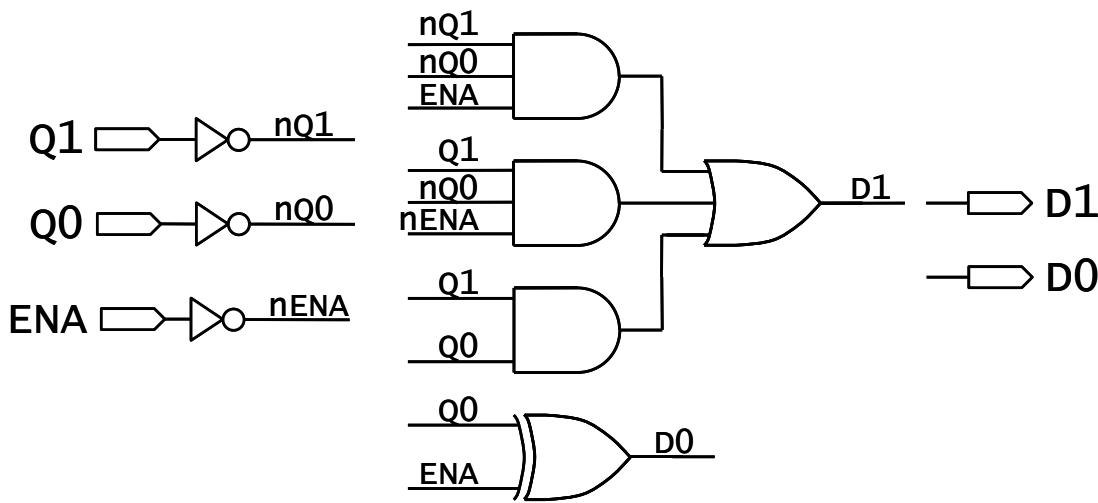


Tabla de salida

Este autómata es de Moore puesto que las salidas (cuenta) solo dependen del estado en el que se encuentra el autómata (estado actual)

ESTADO	SALIDA
0	00
1	01
2	10
3	11

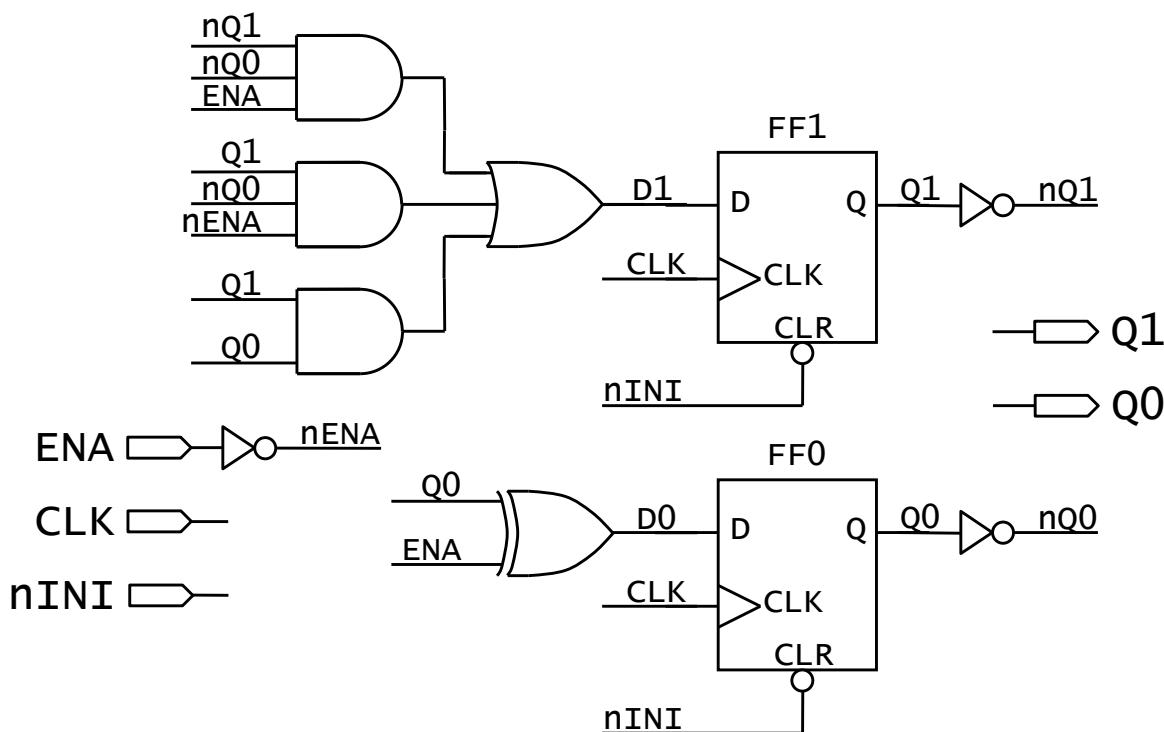
O lo que es lo mismo:

ESTADO ACTUAL		SALIDAS	
Q1	Q0	S1	S0
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

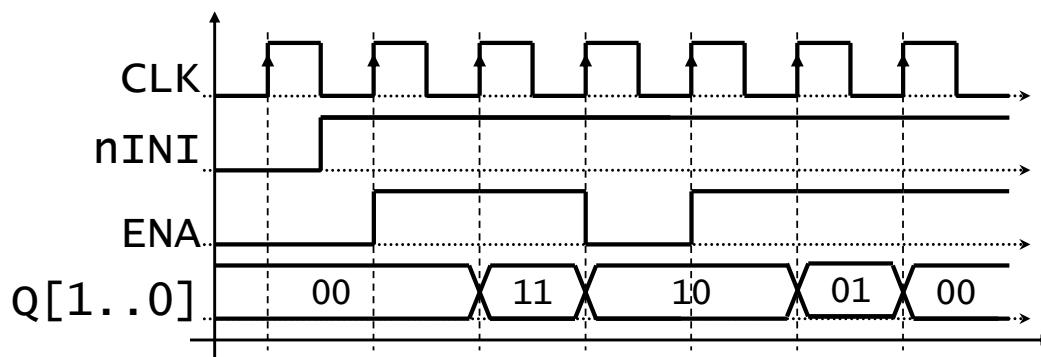
Logica de las salidas

$$S1 = Q1$$

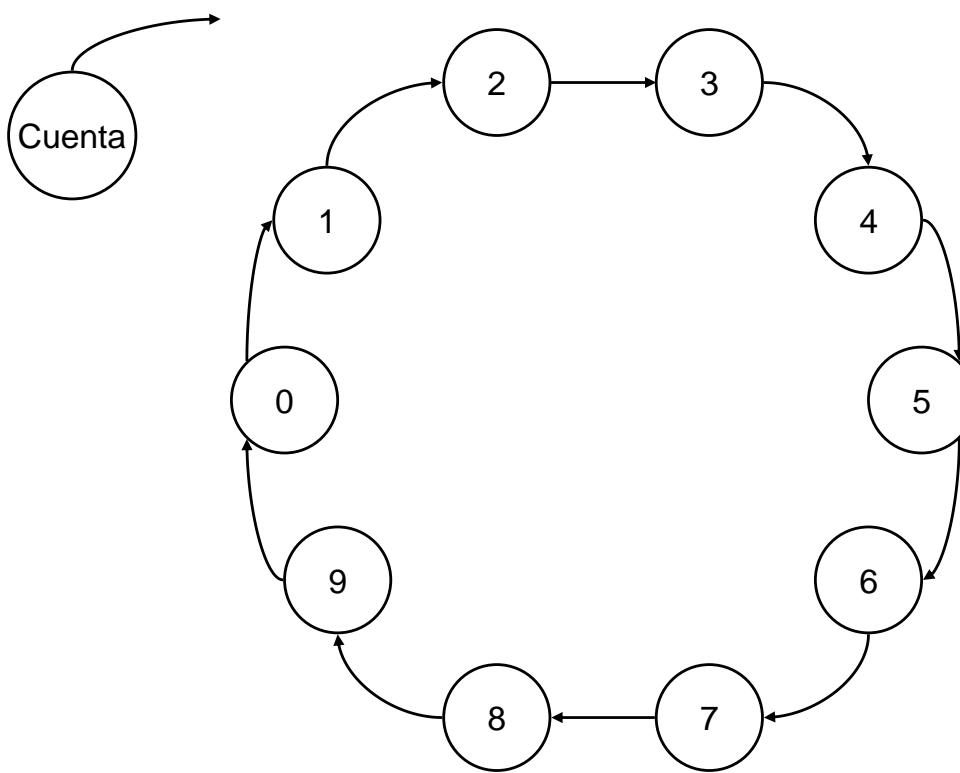
$$S0 = Q0$$

Círcuito resultante

Represente en el siguiente cronograma el funcionamiento del contador descendente de 2 bits.



EJERCICIO 2) El diagrama de estados de la figura representa el funcionamiento de un contador decimal ascendente.



A partir de este diagrama de estados, aplique la metodología de diseño de circuitos secuenciales y diseñe el circuito correspondiente a dicho contador.

Codificación de los estados

ESTADO	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Memoria de estado

El tener 10 estados son necesarios 4 flip-flops tipo D ($10 \text{ estados} \leq 2^4 \text{ flip-flops}$).

Tabla simbólica de transiciones

ESTADO ACTUAL	ESTADO FUTURO
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	0

Tabla de transiciones codificada

ESTADO ACTUAL				ESTADO FUTURO			
Q3	Q2	Q1	Q0	D3	D2	D1	D0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

Funciones lógicas que definen el estado futuro

$$D3 = \overline{Q3} Q2 Q1 Q0 + Q3 \overline{Q2} \overline{Q1} \overline{Q0}$$

$$D2 = \overline{Q3} (\overline{Q2} Q1 Q0 + Q2 \overline{Q1} Q0)$$

$$D1 = \overline{Q3} (Q1 \oplus Q0)$$

$$D0 = \overline{Q0}$$

Lógica de control de transiciones

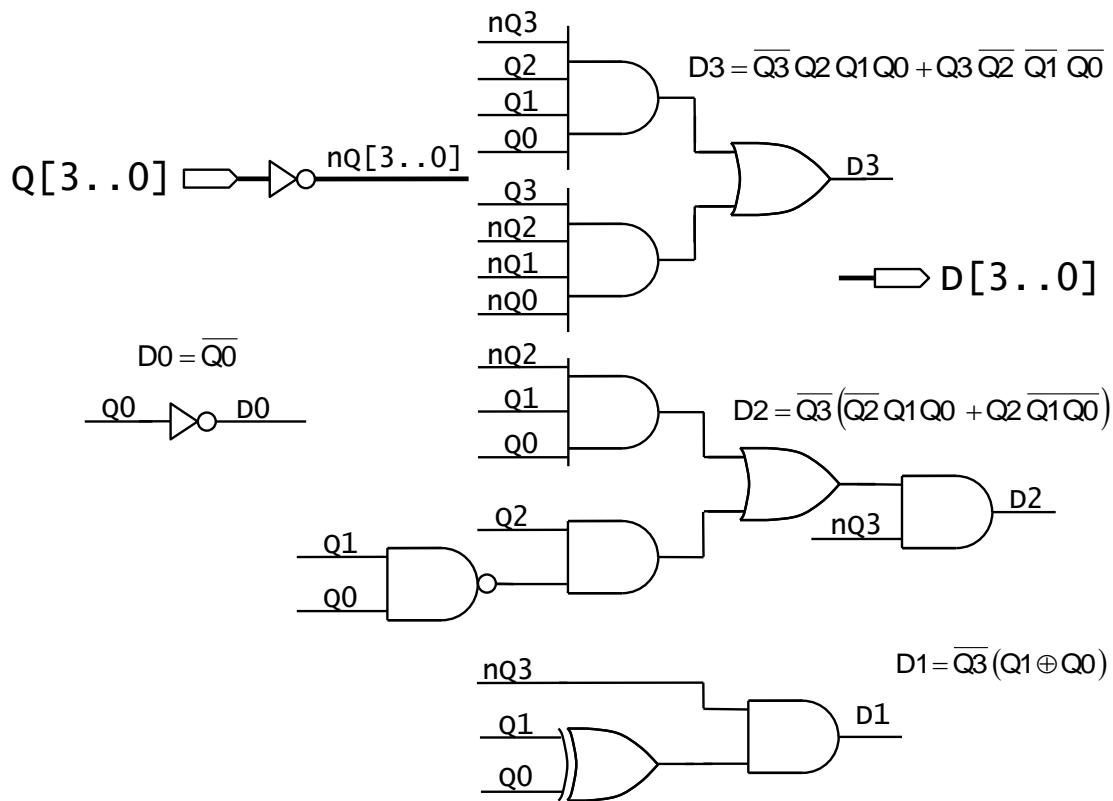


Tabla de salida

Este autómata es de Moore puesto que las salidas (cuenta) solo dependen del estado en el que se encuentra el autómata (estado actual).

ESTADO	SALIDA
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

O lo que es lo mismo:

ESTADO ACTUAL					SALIDAS			
Q3	Q2	Q1	Q0	S3	S2	S1	S0	
0	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	0	0	1	1	
0	1	0	0	0	1	0	0	
0	1	0	1	0	1	0	1	
0	1	1	0	0	1	1	0	
0	1	1	1	0	1	1	1	
1	0	0	0	1	0	0	0	
1	0	0	1	1	0	0	1	

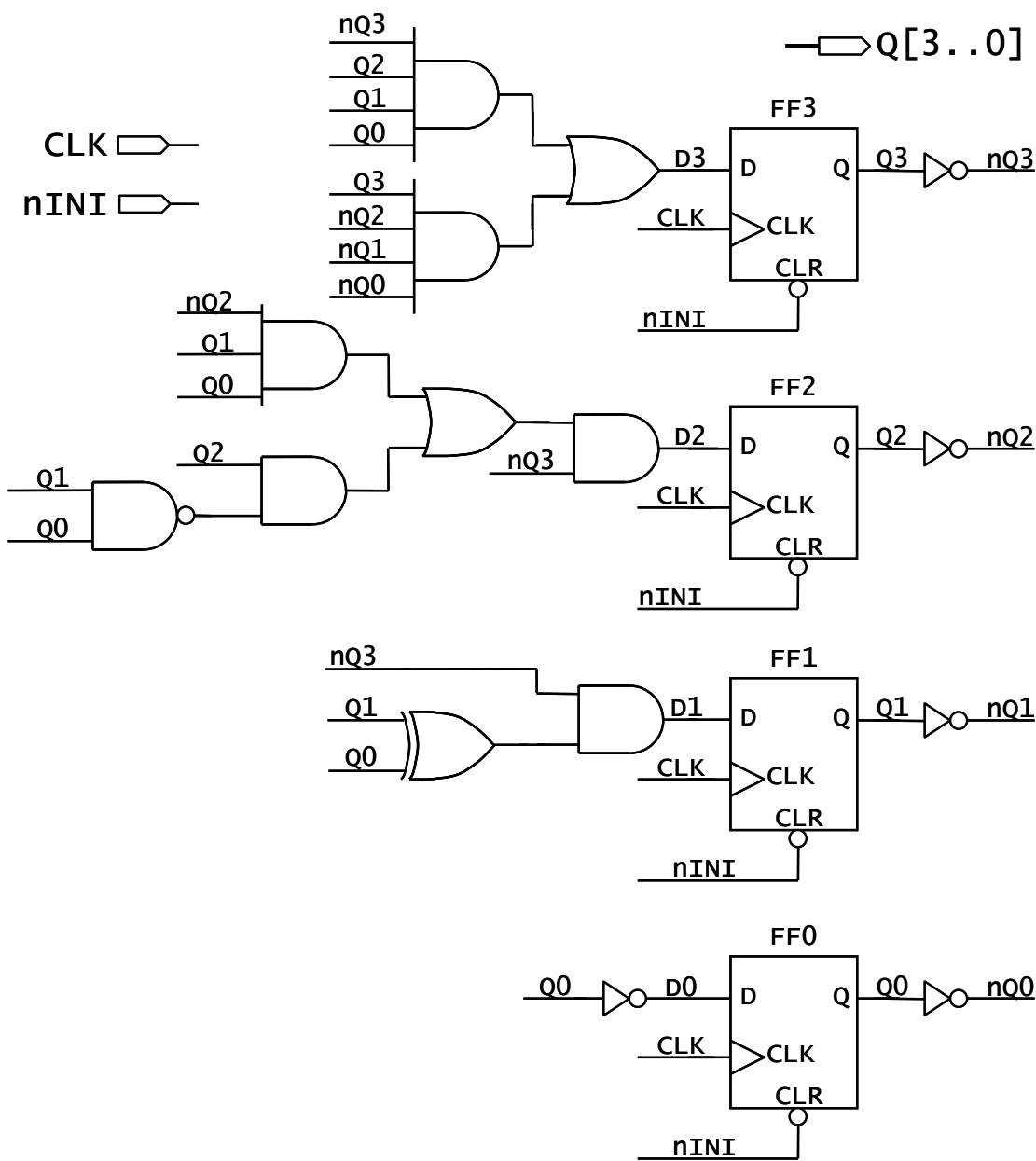
Lógica de las salidas

$$S3 = Q3$$

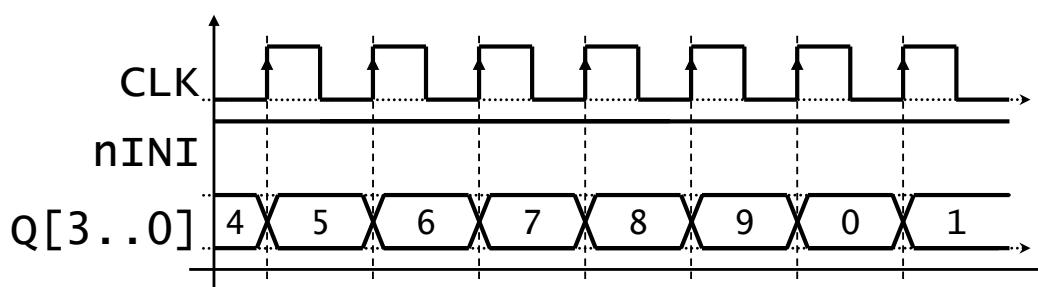
$$S2 = Q2$$

$$S1 = Q1$$

$$S0 = Q0$$

Círcuito resultante

Represente en el siguiente cronograma el funcionamiento del contador decimal ascendente.



BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Contadores

CÓDIGO:
BT3_AINP6

FECHA: _____

NOMBRE: _____ APELLIDOS: _____

MODALIDAD: Lectura Dirigida Individual TIPO: No presencial DURACIÓN: 440 minutos

CALENDARIO: Quinta y Sexta Sesión
Previo a la Sexta y Séptima Sesión REQUISITOS: Análisis y diseño de circuitos secuenciales síncronos

CRITERIO DE ÉXITO: Compresión de los procedimientos y conceptos expuestos.

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	_____ minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	No procede
------------------	---------------	--	------------

PARTE I. Descripción de la actividad

Esta actividad persigue el propósito de que adquiera habilidades de análisis y diseño de contadores. La actividad se descompone en las siguientes tareas:

1. Lectura del texto que se incluye en este documento: El contenido del texto incluye todos los conocimientos que debe adquirir. El texto lleva intercalados ejercicios, que debe realizar para comprobar si ha asimilado todos los conceptos y procedimientos expuestos.
2. Durante la actividad no presencial correspondiente a la quinta sesión deberán ser leídas las primeras 25 páginas. El resto, a partir del apartado “Diseño de contadores módulo M”, corresponderá a la actividad no presencial correspondiente a la sesión sexta.
3. Anotación de dudas y curiosidades: Debe anotar cualquier duda o curiosidad que le haya surgido en relación con los contenidos de la actividad. Durante la sexta y séptima sesiones presenciales el profesor dedicará 30 minutos aproximadamente a resolver las cuestiones que se le planteen.

PARTE II. Contadores

Los contadores, tal y como indica su nombre, son subsistemas secuenciales que sirven para contar. Y lo que cuentan, en concreto, es el transcurso de instantes de tiempo, de ciclos de reloj. En esta parte del documento se van a presentar las características y tipos de contadores, así como las metodologías que pueden utilizarse para diseñarlos. Primero se va a explicar el funcionamiento de un contador binario síncrono con una funcionalidad básica y, después, se pasará a estudiar contadores con funcionalidades más complejas y procedimientos para la realización de contadores modulares.

Contadores binarios

Los contadores binarios expresan la cuenta en código binario natural. En su forma más básica, la única entrada de un contador binario, además de la de inicialización asincrónica, es la entrada de reloj; la salida del contador indica el estado de cuenta. En la figura 1a se representa el funcionamiento de un contador binario de 3 bits; en la 1b se muestra su símbolo.

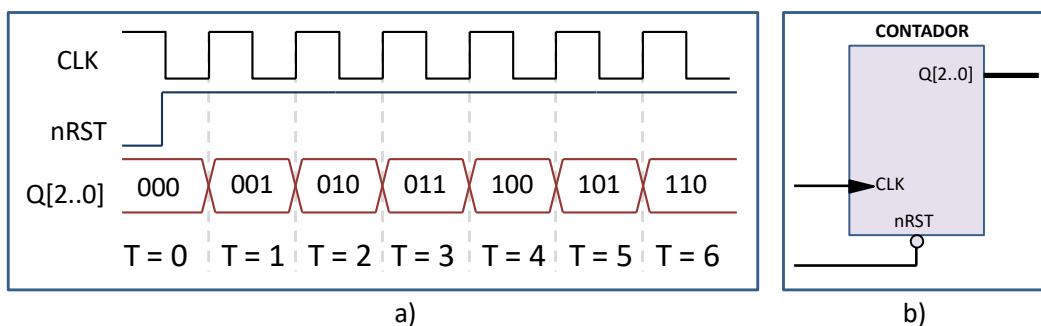


Figura 1.- Contador Binario de 3 bits

Como puede observarse en la figura 1a, el estado de cuenta se incrementa en cada flanco activo de reloj, de modo que la cuenta representa el número de instantes de tiempo (periodos de reloj) transcurridos desde el inicio de la misma.

Arquitectura de los contadores binarios

La estructura hardware de un contador binario como el de la figura 1 se muestra en la figura 2. El contador se construye como un circuito secuencial síncrono con arquitectura de Moore, donde la salida de cuenta es el estado de memoria de un registro, (la **cuenta actual**), y el estado de **cuenta futuro** (la entrada del registro) se calcula con un **incrementador** que suma 1 al estado de cuenta actual. Observe que este modelo básico de contador presenta la singularidad de carecer de entradas síncronas.

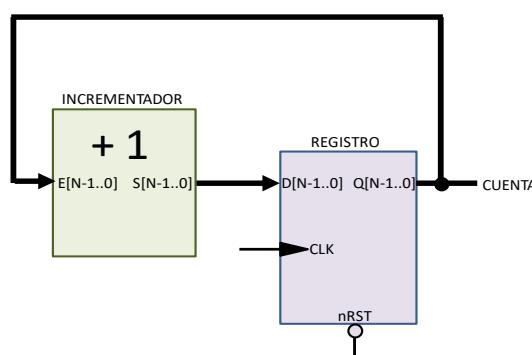


Figura 2.- Arquitectura de un contador binario básico

La arquitectura del contador binario pone de manifiesto que el número de flip-flops que es necesario emplear para construirlo es igual al número de bits de salida del contador.

Módulo de un contador binario

Un contador binario dispone de estados de cuenta que van desde 0 (todos los bits de cuenta a 0) hasta un valor máximo de cuenta. El valor máximo de cuenta de un contador con N bits de salida es el número más grande que puede expresarse en binario natural con N bits: $2^N - 1$ (todos los bits de cuenta a 1).

Ejemplos:

1. El valor máximo de cuenta de un contador binario de 4 bits es 15 (1111_b)
2. El valor máximo de cuenta de un contador binario de 8 bits es 255 (1111 1111_b)
3. El valor máximo de cuenta de un contador binario de 20 bits es 1 048 575 (FFFFFh)

Un contador binario de N bits dispone, por tanto, de 2^N estados de cuenta diferentes, desde 0 hasta $2^N - 1$; el número total de estados de cuenta de un contador se denomina **módulo** del contador.

Ejemplos:

1. El módulo de un contador binario de 4 bits es 16.
2. El módulo de un contador binario de 8 bits es 256.
3. El módulo de un contador binario de 20 bits es 1 048 576

Observando la arquitectura del contador en la figura 2, puede deducirse cuál es el estado de cuenta que sigue al de valor máximo: el valor máximo de salida, $2^N - 1$, es una combinación en que los N bits valen 1; tras sumar 1 a esta combinación, los N bits menos significativos del resultado son todos 0:

$$\begin{array}{r} \text{n bits} \\ \overbrace{111111\dots1111} \\ +1 \\ \hline \textbf{1} 000000\dots0000 \\ \text{n bits} \end{array}$$

En consecuencia, tras alcanzar el estado máximo de cuenta el circuito vuelve a cero y la secuencia de cuenta se repite cíclicamente con un periodo (en ciclos de reloj) igual al módulo del contador. En el cronograma de la figura 3 se muestra un ejemplo en el que puede observarse esta propiedad característica del modo de funcionamiento de los contadores.

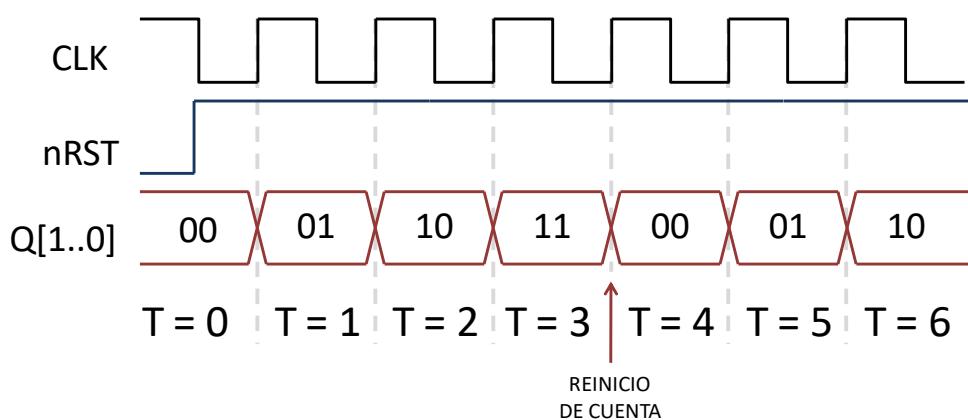


Figura 3.- Cuenta cíclica

Diseño de contadores binarios sin entradas síncronas

El diseño lógico de un contador binario como el de la figura 2 puede abordarse de tres maneras diferentes. Con todas ellas se alcanza el mismo resultado, pero, aún así, y porque cada una representa una estrategia diferente de diseño de la que se pueden extraer conclusiones provechosas, se van a explicar todas ellas.

Diseño del contador con flip-flops tipo T

La forma más simple de diseñar un contador binario se basa en el análisis de las propiedades de la secuencia de cuenta y el uso de flip-flops con entrada de *Toggle* (flip-flops tipo T).

Un flip-flop tipo T se realiza con uno de tipo D y una puerta XOR. Su estructura se muestra en la figura 4.

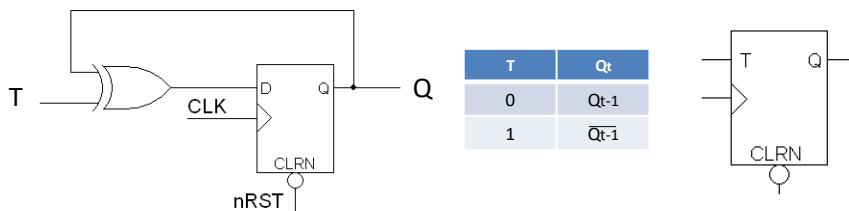


Figura 4.- Flip-flop tipo T

En los flancos activos de reloj en que la entrada T del flip-flop vale 0, el flip-flop mantiene el dato memorizado; en los que vale 1, invierte –conmuta, en inglés *toggle* significa conmutar- el bit memorizado.

En la secuencia de cuenta de un contador binario puede advertirse que existe, para cualquier pareja de estados de cuenta consecutivos (A y A+1), la siguiente relación:

- En el estado de cuenta sucesor (A+1), un bit de peso i conmuta su valor respecto al que tenía en el estado de cuenta antecesor (A), cuando todos los bits del antecesor (A) de peso menor que i valen 1; el bit de peso 0 conmuta en todos los ciclos de reloj.

Esta propiedad puede observarse en cualquier secuencia de cuenta binaria; por ejemplo, en la correspondiente a un contador de 4 bits:

Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	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
0	0	0	0

En base a esta propiedad de la secuencia de cuenta en binario natural es posible diseñar un contador de N bits de la siguiente manera:

- Utilizando un flip-flop tipo T por cada bit de cuenta (Figura 5)

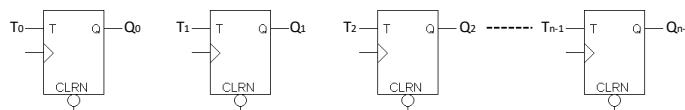


Figura 5.- Flip-flops tipo T

- Conectando, a cada entrada T_i , la AND lógica de todas las salidas de cuenta de peso menor que i, salvo la entrada T_0 que se conecta a 1 (Figura 6).

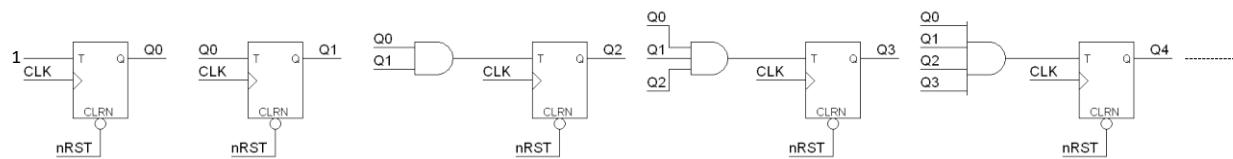


Figura 6.- Contador binario con flip-flops tipo T

Con esta conexión, un bit de cuenta de peso i comuta cuando todos los bits con un peso menor que el suyo valen 1.

El circuito de la figura 6, que representa ya la realización de un contador binario con flip-flops tipo T, puede, con una leve transformación, realizarse como un circuito regular realizado en base a una célula básica (Figura 7).

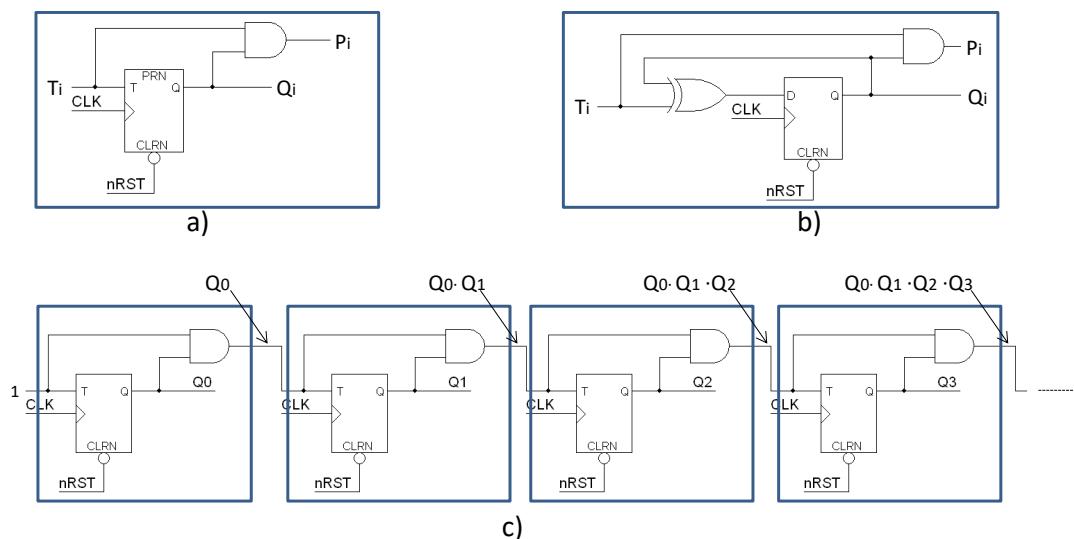


Figura 7.- Realización de un contador binario con una estructura regular

En este circuito cada célula añade, al producto lógico que controla su entrada T_i , el del bit de cuenta de la propia célula y propaga el resultado por su salida P_i para controlar la entrada T_{i+1} . La entrada T_0 del bit de menor peso está puesta a 1 para forzar la comutación de este bit en todos los ciclos de reloj.

El circuito de la figura 7c es la forma más usada en el diseño lógico de contadores binarios, pudiendo estar cada célula representada por su realización con flip-flops tipo T (Figura 7a) o con flip-flops tipo D (Figura 7b)

Diseño de la arquitectura del contador

Esta alternativa consiste en diseñar por separado los dos bloques de la arquitectura representada en la figura 2 (que por comodidad reproducimos aquí), el incrementador y el registro, y conectarlos tal y como se indica en dicha figura: la salida del registro, que es el estado actual de cuenta y, por tanto, la salida del contador, se conecta a la entrada de datos del incrementador, y la salida del incrementador, que es el estado futuro de cuenta, se conecta a la entrada de datos del registro.

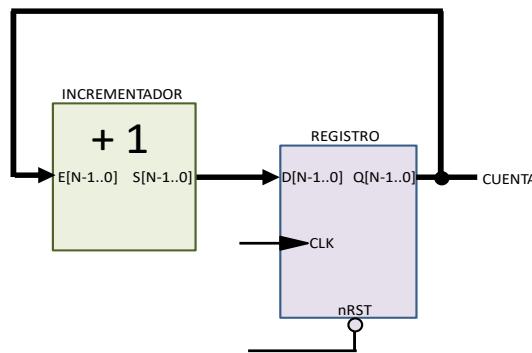


Figura 2 (repetida).- Arquitectura de un contador binario básico

Si en el circuito de la figura 7c se sustituye el flip-flop tipo T por su realización con un flip-flop tipo D y una puerta XOR, y se cambia la disposición espacial de las células (figura 8), puede constatarse que en dicho circuito se distinguen dos bloques: un conjunto de flip-flops tipo D y un circuito combinacional.

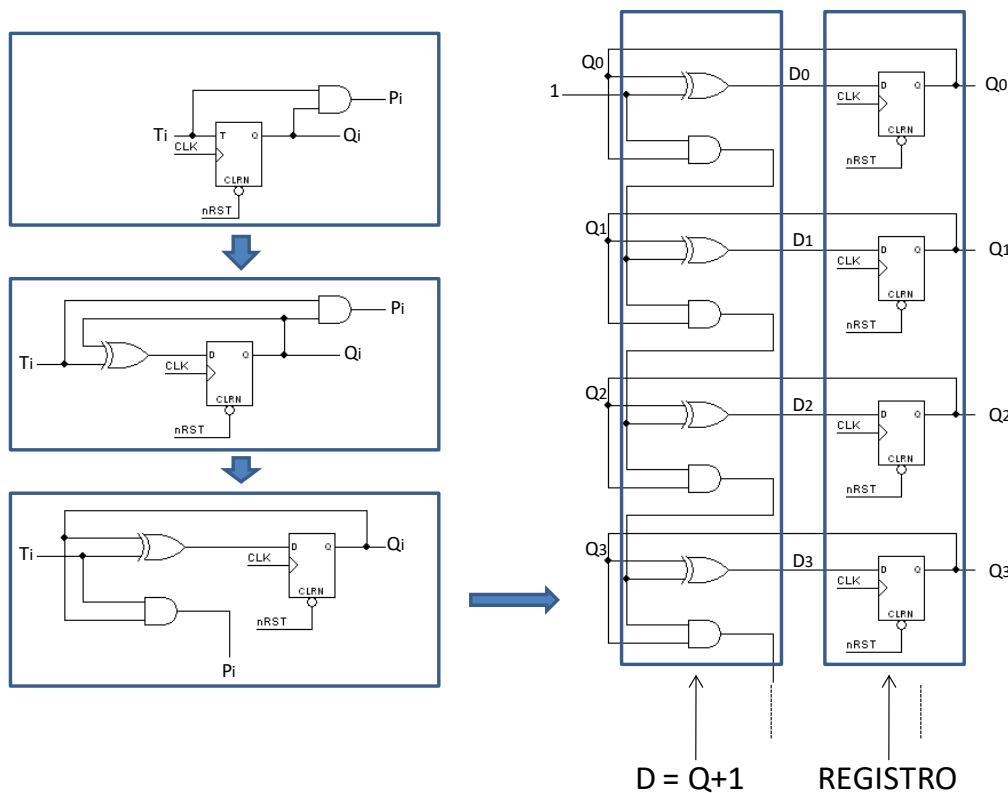


Figura 8.- Realización del contador con una arquitectura registro-incrementador

Es evidente que el conjunto de flip-flops remarcados en la figura 8 constituyen un registro paralelo-paralelo, sin habilitación, de tantos bits como tenga el contador y se puede hacer corresponder, por tanto, con el registro de la arquitectura del contador representada en la figura 2.

Las células que componen la lógica combinacional en el esquema de la figura 8 se han dispuesto de modo que se pueda apreciar la estructura regular del circuito. Cada puerta XOR y AND forman lo que se conoce como un **semisumador**: un circuito capaz de calcular la suma de dos bits y la generación de acarreo de dicha suma (figura 9).

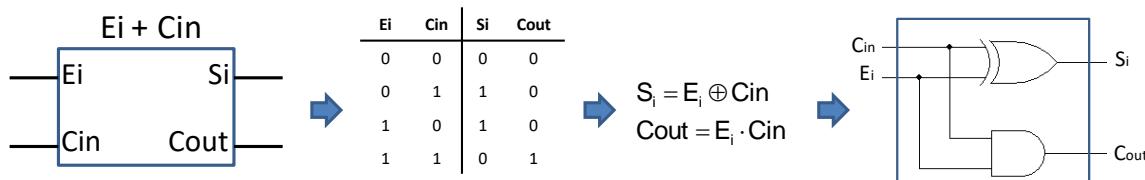


Figura 9.- Sumador de 2 bits

El semisumador de la figura 9 –se conoce con este nombre porque combinando dos puede realizarse un **sumador completo**– puede emplearse para realizar un circuito capaz de sumar un dato y un acarreo de entrada. En la figura 10 se muestra la estructura de este circuito.

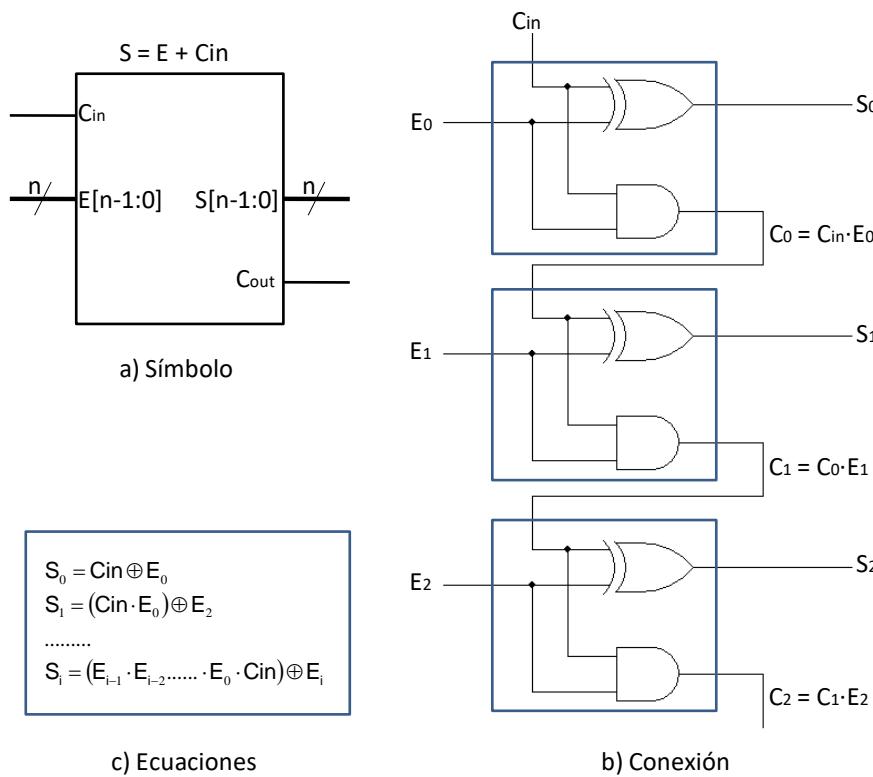


Figura 10.- Sumador de un dato y el acarreo

Como puede observarse en la figura 10b, cada bit de salida (S_i) se obtiene como la suma del bit homólogo de entrada (E_i) y el acarreo generado por el bit anterior (C_{i-1}). Si en la entrada de acarreo de este circuito, C_{in} , se fija un 1, el circuito se convierte en un incrementador ($S = E + 1$), y puede comprobar que es idéntico al circuito combinacional enmarcado en la figura 8, lo que nos permite deducir un método que permite diseñar un contador binario siguiendo el modelo de arquitectura registro + incrementador de la figura 2, con el que se obtiene exactamente la misma solución que siguiendo el método basado en el empleo de flip-flops tipo T.

El método consiste en:

- Realizar un registro paralelo-paralelo de tantos bits como tenga el contador.
- Realizar un incrementador con acarreo, utilizando circuitos semisumadores, y conectar su acarreo de entrada a 1.
- Conectar los dos bloques anteriores de acuerdo con la arquitectura de la figura 2.

En la figura 11 se muestra en detalle el diseño lógico de un contador de 6 bits realizado con esta metodología.

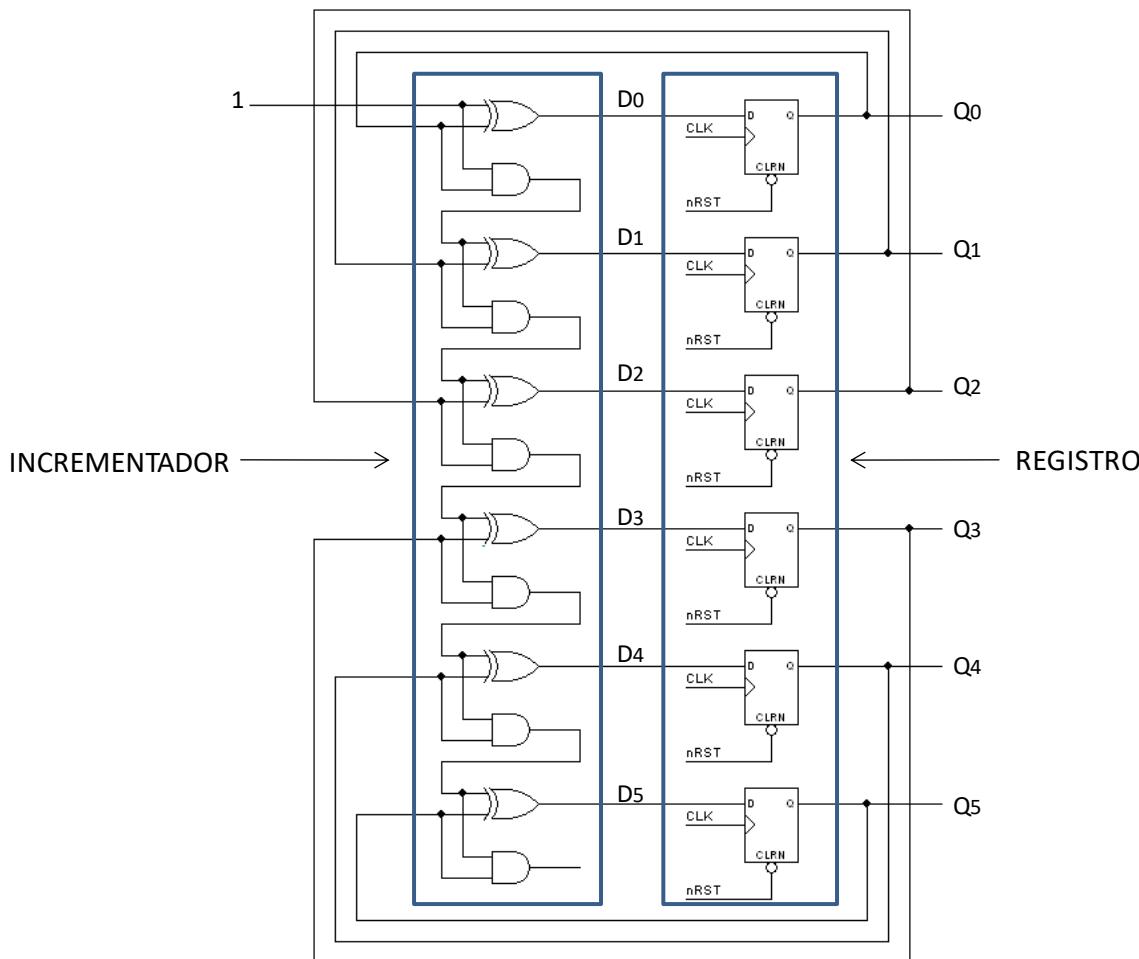
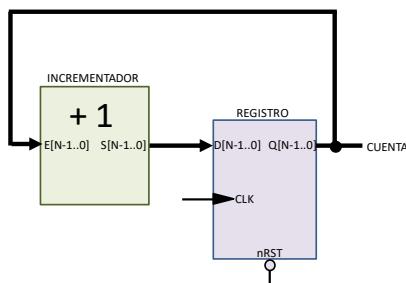


Figura 11.- Circuito lógico de un contador binario de 6 bits

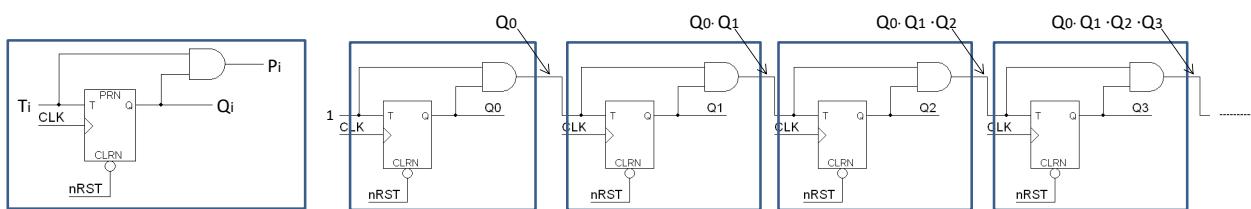
La estructura lógica de los circuitos de las figuras 11 y 7c es idéntica, sólo se diferencian en el modo de representación (con flip-flops tipo D o T). Por tanto, el resultado es equivalente, pero:

- La metodología de diseño basada en el uso de una célula básica construida con un flip-flop tipo T es más fácil de recordar y da lugar a una representación más simple –compare los esquemas de las figuras 7c y 11.
- El modelo de arquitectura de la figura 2 (registro + incrementador) expresa mucho mejor –más claramente- el funcionamiento de un contador que el circuito de la figura 7 y simplifica, además, considerablemente el análisis y diseño de contadores más complejos como los que estudiaremos en apartados posteriores de este texto.

Resulta conveniente, por tanto, interpretar que un contador binario tiene la estructura de bloques de la figura 2 y saber que la realización de esta arquitectura puede abordarse mediante el diseño de una célula básica en torno a flip-flops tipo T, siguiendo la estructura de conexión definida en la figura 7c. Esta doble concepción de la estructura del contador binario se utilizará como base de partida para explicar el funcionamiento y diseño de contadores más complejos y se resume en la figura 12.



a) Modelo de Arquitectura y funcionamiento



b) Modelo para el diseño lógico

Figura 12.- Modelo de diseño del contador binario

Diseño de contadores binarios a partir del diagrama de estados

El diseño de un contador a partir de un diagrama de estados es un método que apenas se utiliza. En primer lugar, porque no es posible aplicarlo cuando el contador tiene un módulo grande, ya que implicaría la realización de un diagrama de estados con cientos, miles o millones de estados de memoria distintos. Y, en segundo lugar, porque el uso de esta metodología termina consistiendo en el diseño de un incrementador y un registro (el diseño del contador en base a su arquitectura), ya que:

- La memoria de estado del circuito es un registro paralelo-paralelo de tantos bits como tenga el contador.
- No existe un bloque combinacional de cálculo de las salidas, porque la salida del contador es su estado de memoria.
- La tabla de verdad derivada de la tabla de transiciones define necesariamente la funcionalidad de un circuito incrementador: estado futuro = estado actual + 1.

En conclusión: termina por ser el mismo problema para el que ya se ha encontrado soluciones más eficientes (la realización con flip-flops tipo T o a partir del modelo de arquitectura) sin recurrir al uso de diagramas de estado.

A modo de ejemplo: en la figura 13 se muestra el diagrama de estados y la tabla de transiciones de un contador binario de módulo 4.

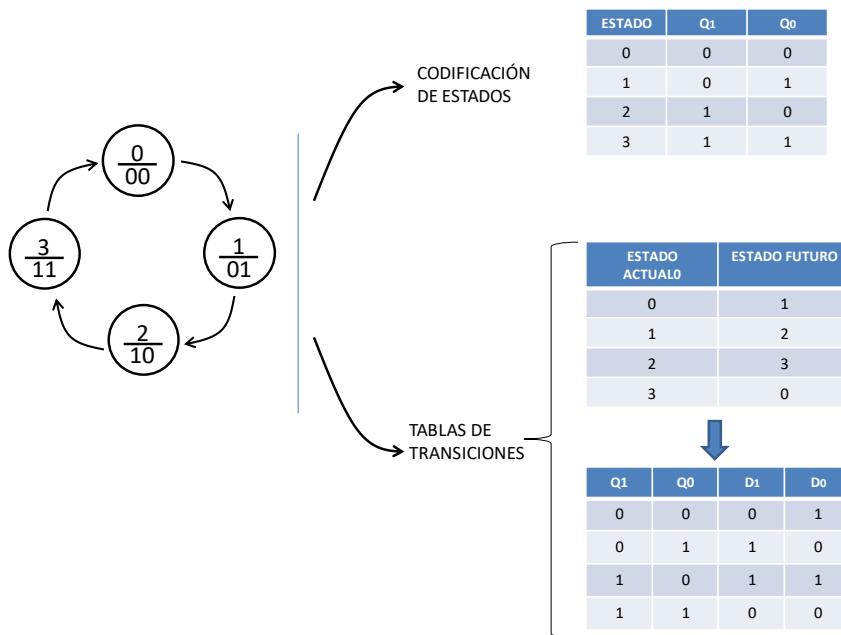


Figura 13.- Diagrama de estados de un contador binario de 2 bits

Codificando cada estado con el código binario de la cuenta que representa, la salida del circuito es la salida de los flip-flops de memoria, y el circuito carece de lógica combinacional de cálculo de las salidas. La lógica combinacional de control de la memoria de estado viene definida por la tabla de transiciones, y la relación entre las entradas de la tabla (el estado actual) y las salidas (el estado futuro) es la que corresponde a la función de un incrementador. Por tanto, el autómata tendría una arquitectura con dos bloques, como la que se muestra en la figura 14, equivalentes al registro y al incrementador de la arquitectura de la figura 12a.

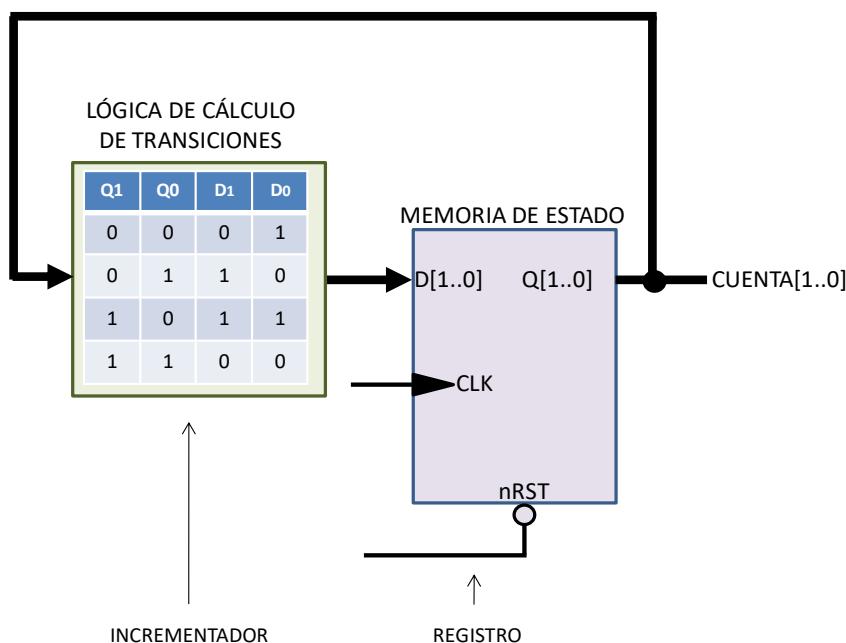


Figura 14.- Realización del contador a partir del diagrama de estados

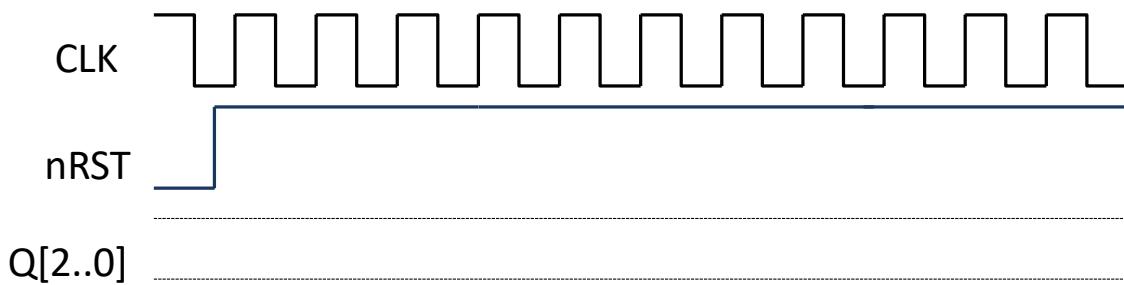
La conclusión final es que para el diseño de contadores no se recurre a la metodología de diseño de circuitos secuenciales síncronos a partir de diagramas de estado.

Ejercicios

1.- Diseñe un contador binario de 4 bits, con entrada de reset asíncrono activa a nivel bajo, empleando flip-flops tipo T. Detalle la realización de un flip-flop tipo T empleando un flip-flop tipo D y una puerta XOR.

2.- Diseñe un contador binario de 3 bits, con entrada de reset asíncrono activa a nivel bajo, empleando flip-flops tipo D.

3.- Rellene el cronograma adjunto para representar el funcionamiento del contador del ejercicio 2



Entradas síncronas de los contadores

La funcionalidad del contador binario que se ha presentado puede ampliarse añadiendo entradas síncronas de control. Un contador puede disponer de cualquiera de los siguientes tipos de entradas síncronas:

- Una entrada de **Habilitación de Reloj**: la entrada de habilitación permite controlar los instantes de tiempo (flancos de reloj) en los que el contador incrementa o mantiene su estado de cuenta.
- Una entrada de **Reset Síncrono**: sirve para reiniciar sincrónicamente la cuenta desde 0.
- Una entrada de **Inicialización Síncrona**: sirve para reiniciar sincrónicamente la cuenta desde un valor determinado (1, 35, etcétera).
- Una entrada de **Control del Sentido de Cuenta**: Es una entrada –denominada habitualmente *up/down*- disponible en contadores capaces de contar en sentido *ascendente* (0, 1, 2, etcétera) y *descendente* (7, 6, 5, etcétera). El nivel lógico de esta entrada determina un sentido de cuenta.

Y puede, también, disponer de una salida adicional a la de cuenta:

- Una salida de **Fin de Cuenta**: Es una salida que se activa cuando se alcanza el **último estado** de cuenta –en los contadores que cuentan en sentido ascendente, el último es el valor máximo de cuenta; en los que lo hacen en sentido descendente es el mínimo valor de cuenta.

A continuación se va a explicar cómo las entradas síncronas de control afectan al funcionamiento básico de un contador, su utilidad práctica, y cómo pueden materializarse modificando ligeramente la arquitectura básica representada en la figura 2. También se explicará la utilidad de la salida de fin de cuenta.

Contadores con habilitación de reloj

Los contadores sin entrada de habilitación, cuyo funcionamiento ya conoce, incrementan su cuenta en cada flanco activo de reloj y sirven, únicamente, para contar instantes de tiempo (ver figura 1). La salida de este tipo de contadores indica el tiempo transcurrido –y expresado en períodos de reloj- desde un determinado instante de tiempo. A partir del valor de su cuenta, y conocido el periodo de reloj, puede calcularse el tiempo efectivo transcurrido desde el comienzo de la cuenta:

$$\text{Tiempo} = \text{Cuenta} \times \text{Periodo (clk)}$$

Ejemplo: Si en un instante de tiempo, T, la salida de un contador sin habilitación de reloj toma el valor binario 101110 (46) y la frecuencia de reloj es de 1 MHz, el tiempo transcurrido desde el inicio de la cuenta es:

$$\text{Tiempo} = 46 \times 1 \mu\text{s} = 46 \mu\text{s}$$

Un contador con habilitación de reloj incrementa su cuenta únicamente cuando la entrada de habilitación está activa (Figura 15).

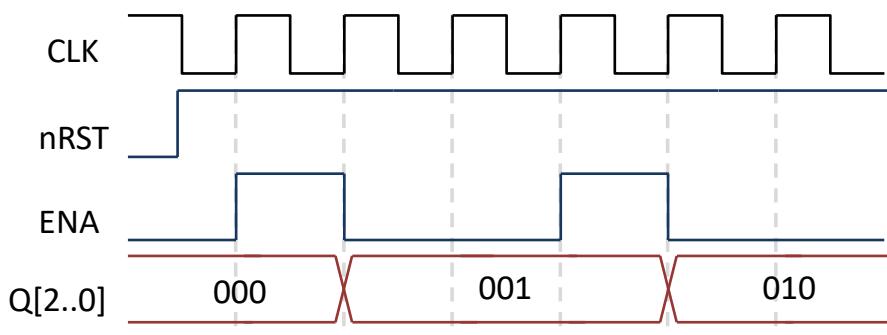


Figura 15.- Contador con habilitación de reloj

Este tipo de contadores llevan la cuenta del número de pulsos ocurridos en la entrada de habilitación, tal y como puede observarse en el cronograma de la figura 15, si la anchura de estos pulsos es de un periodo de reloj. En este caso y si los pulsos de habilitación son periódicos, también sirven para contar tiempo; si los pulsos de habilitación no son periódicos, pero duran un ciclo de reloj, cuentan eventos representados por pulsos de habilitación.

Ejemplo: Si la salida de un contador con entrada de habilitación vale, en un instante de tiempo T, 101110 (46) y su entrada de habilitación está conectada a una señal digital que tiene una frecuencia de 1 KHz, cuyos pulsos a nivel alto duran 1 periodo de reloj, el tiempo transcurrido desde el inicio de la cuenta es:

$$T = 46 \times 1 \text{ ms} = 46 \text{ ms}$$

Ejemplo: Si la entrada de habilitación de un contador está controlada por una señal que se activa, durante un ciclo de reloj, cada vez que una persona entra en un local, y el estado de cuenta es 101110 (46), éste es el número de personas que han entrado en el local desde el inicio de la cuenta.

Los contadores con entrada de habilitación también pueden utilizarse para calcular la duración del pulso de habilitación expresada en número de ciclos de reloj (figura 16).

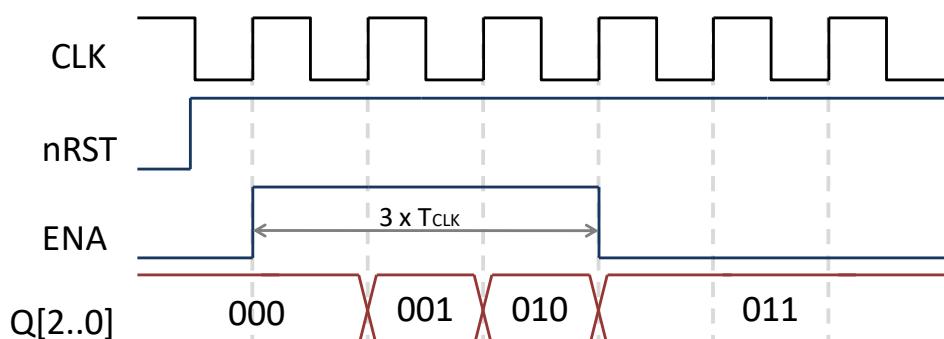


Figura 16.- Contador de anchura de pulso

En este cronograma la cuenta progresó únicamente cuando la entrada de habilitación estuvo activa y representó, cuando se desactivó la habilitación, la anchura del pulso expresada en base al periodo de reloj del contador.

Ejemplo: Si un contador inicializado a 0 presenta, tras una activación de la entrada de habilitación, una cuenta de 10111 (46) y la frecuencia del reloj del contador es de 100 MHz, la duración del pulso de habilitación es de:

$$\text{Duración del pulso} = 46 \times 10 \text{ ns} = 460 \text{ ns}$$

Diseño de contadores con entrada de habilitación

La estructura de un contador binario con habilitación de reloj se obtiene a partir de la representada en la figura 2 sustituyendo el incrementador por un circuito aritmético, que suma al estado actual de cuenta el valor presente en su entrada C_IN, que es la entrada de habilitación (Figura 17a): de este modo sólo se incrementa la cuenta en los flancos de reloj en los que el contador se encuentra habilitado (en los que la entrada C_IN vale 1), en caso contrario la memoria del circuito (el dato almacenado en el registro) no cambia (el estado de cuenta futuro es la suma del actual más 0) y se mantiene el estado de cuenta. En la figura 17b se muestra el símbolo de un contador con habilitación de reloj.

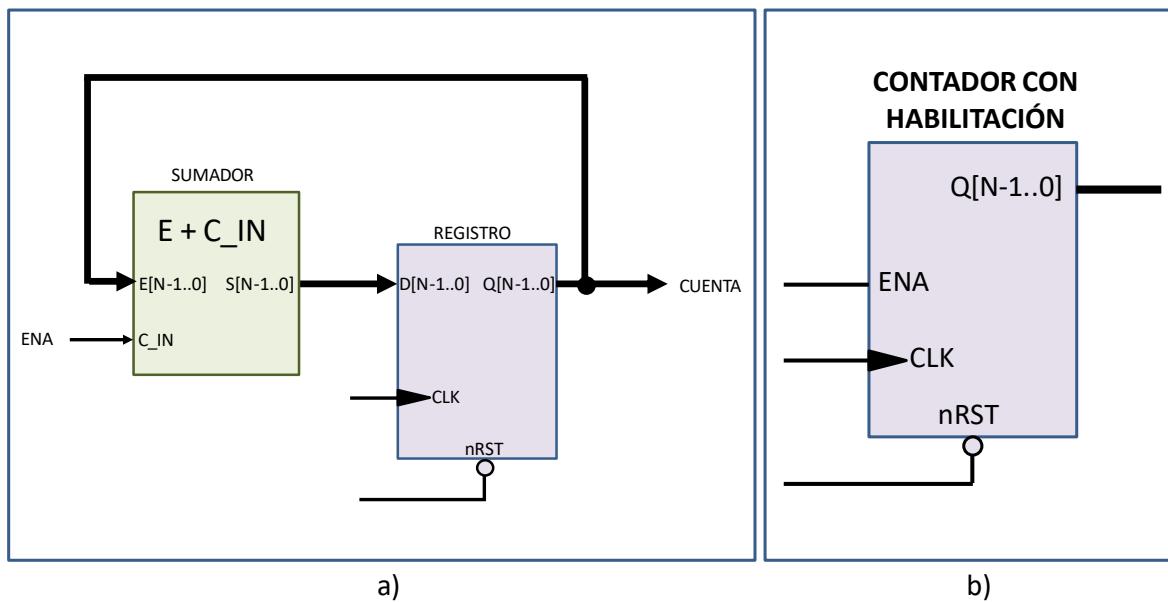


Figura 17.- Arquitectura de un contador con habilitación

Los contadores binarios sin entrada de habilitación, cuya metodología de diseño ya conoce, están realizados, en realidad, empleando los bloques que forman parte de la arquitectura de un contador con habilitación representada en la figura 17. Observando el diseño basado en el modelo de arquitectura esta relación resulta evidente: un contador sin habilitación es, en definitiva, un contador con la entrada de habilitación puesta a 1.

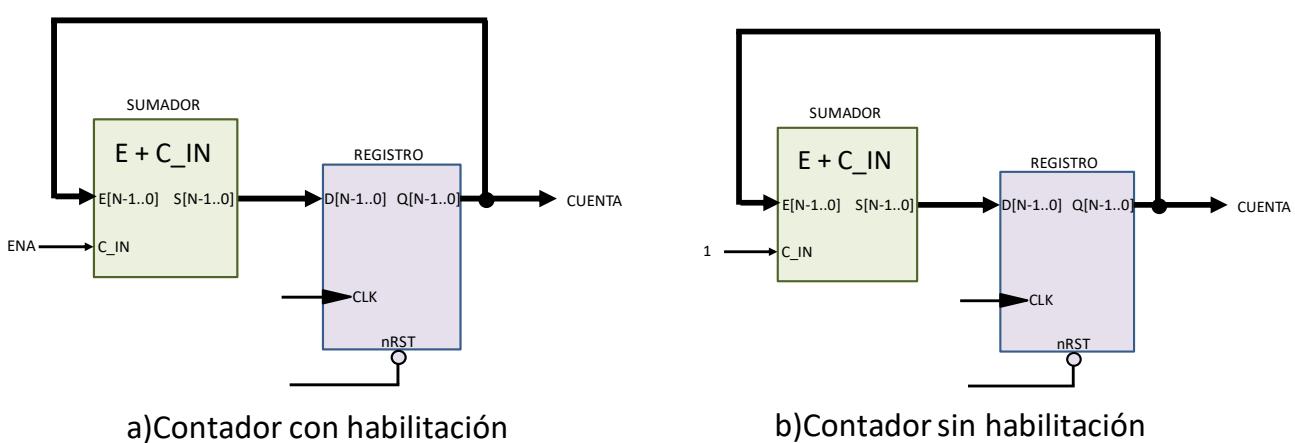


Figura 18.- Relación entre arquitecturas

En la figura 19 se muestra el circuito lógico correspondiente a la realización de ambos tipos de contadores. Un contador con habilitación se realiza, pues, aprovechando la funcionalidad completa del sumador: utilizando la entrada de acarreo como entrada de habilitación (Figura 19a).

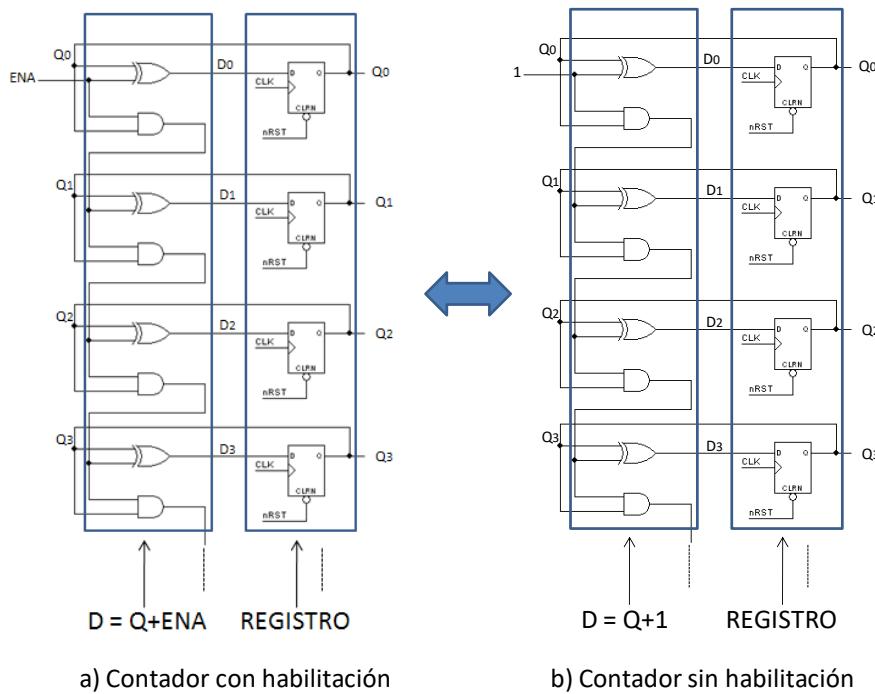


Figura 19.- realización de la arquitectura de contadores con y sin habilitación

Como ya sabe, el circuito de la figura 19b es una representación alternativa a la realización de un contador binario con flip-flops tipo T (Figura 20a). Para realizar, con este modelo, un contador con entrada de habilitación, bastará con conectar la entrada T del flip-flop de menor peso de cuenta a la habilitación (Figura 20b).

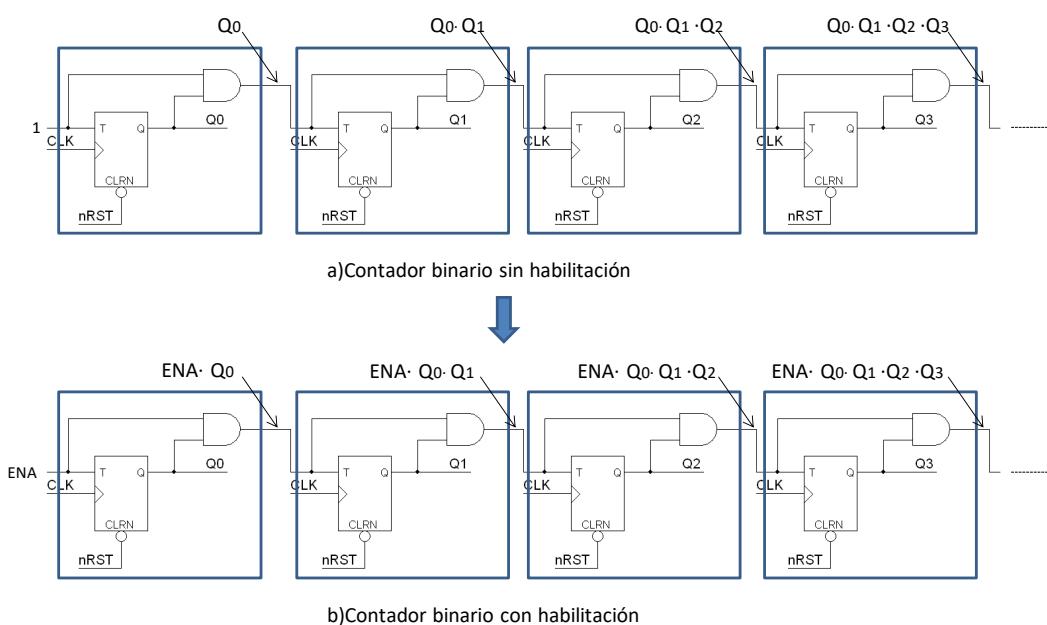


Figura 20.- Contador binario con habilitación realizado con flip-flops tipo T

En el circuito de la figura 20b:

- Cuando la entrada ENA vale 1 el circuito de la figura 20b funciona como el de la figura 20a: realiza la secuencia de cuenta en binario natural.
- En cambio, cuando ENA vale 0 la entrada T de todos los flip-flops del contador es también 0 y, por tanto, el contador mantiene el estado de cuenta.

En definitiva, el circuito de la figura 20b cuenta, cuando ENA vale 1, y mantiene el estado de cuenta cuando ENA vale 0, luego es un contador con entrada de habilitación, realizado con la misma célula básica y estructura de conexión utilizada para diseñar los contadores binarios sin entrada de habilitación.

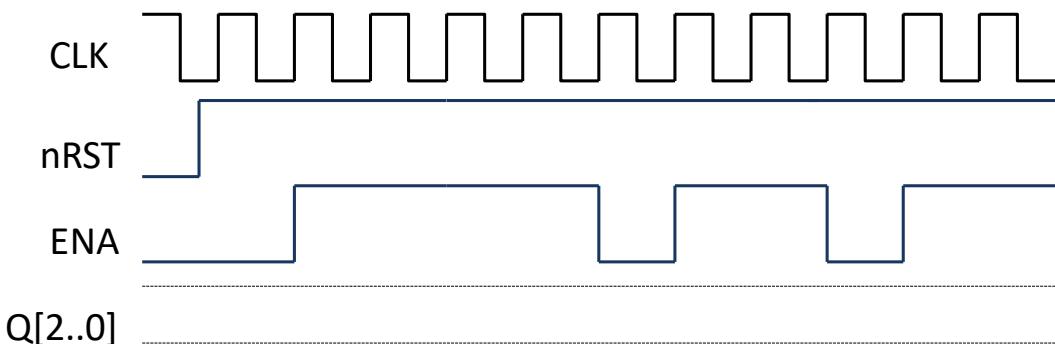
Ejercicios

4.- Diseñe un contador binario de 3 bits con entrada de habilitación (ENA) activa a nivel alto y con entrada de reset asíncrono activa a nivel bajo, empleando flip-flops tipo T.

5.- Indique la modificación que introduciría en el circuito diseñado en el ejercicio 4 para que la entrada de habilitación sea activa a nivel bajo.

6.- Indique la modificación que introduciría en el circuito diseñado en el ejercicio 4 para que se comporte como un contador binario sin entrada de habilitación.

7.- Rellene el cronograma adjunto para representar el funcionamiento del contador del ejercicio 4.



Contadores con salida de fin de cuenta

La salida de fin de cuenta se activa cuando el contador alcanza el último estado de cuenta. En la figura 21 se representa el funcionamiento de un contador binario de módulo 4 con este tipo de salida.

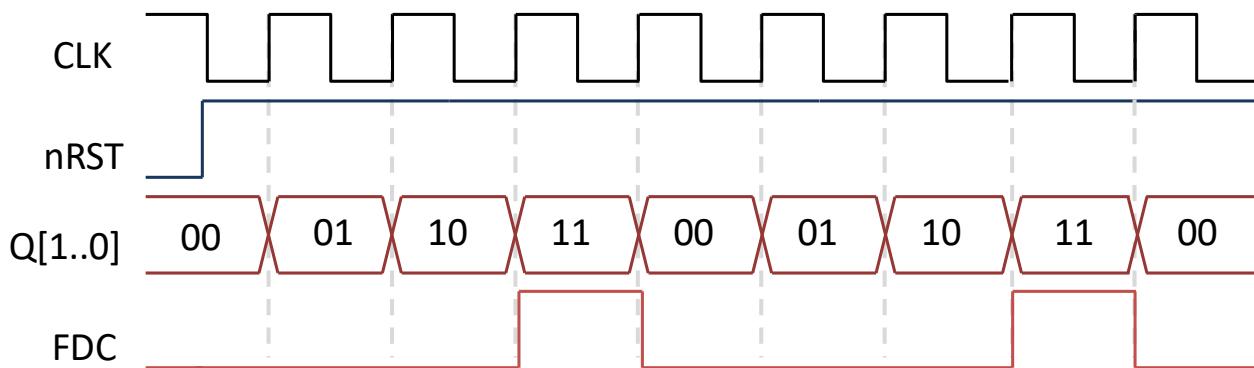


Figura 21.- Contador con salida de fin de cuenta

Como puede observarse, la salida de fin de cuenta (FDC) se activa cuando el estado de cuenta es 3, en cualquier otro estado la salida de fin de cuenta está inactiva y, como los contadores pasan por el último estado, o por cualquier otro, periódicamente –el de la figura 21 cada 4 ciclos de reloj-, es también una señal periódica. El periodo de la señal de fin de cuenta es:

$$T_{FDC} = \text{módulo del contador} \times T_{CLK}$$

Y su frecuencia, por tanto:

$$F_{FDC} = F_{CLK}/\text{módulo}$$

De esta relación se deriva una de las aplicaciones más importantes de los contadores: su uso como **divisores de frecuencia**. Para generar una señal con un periodo que sea múltiplo del de la señal de reloj –que es lo mismo que decir que su frecuencia sea un factor de la del reloj- debe utilizarse un contador con salida de fin de cuenta cuyo módulo sea el factor que relaciona el periodo de reloj y el periodo de la señal de salida.

Ejemplo:

Para generar una señal con un periodo de 1 segundo a partir de un reloj de 1 MHz, se debe realizar un contador de módulo 1 millón con salida de fin de cuenta; la salida de fin de cuenta tendrá una frecuencia de 1 Hz.

Diseño de contadores con salida de fin de cuenta

La salida de fin de cuenta en un contador binario básico depende únicamente del estado de cuenta -se genera decodificando el último estado de cuenta (Figura 22)- y da lugar a un circuito que encaja en una arquitectura de Moore.

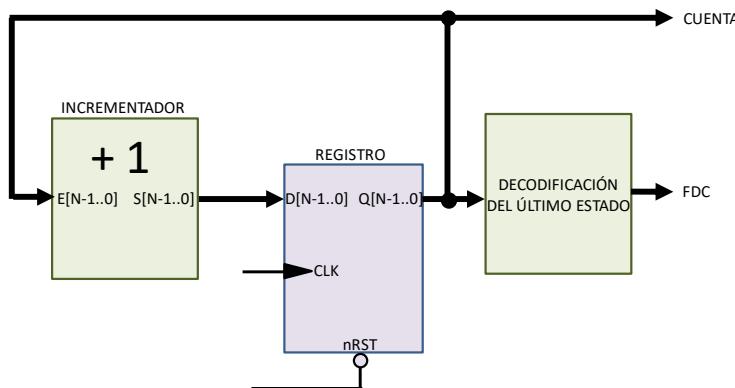


Figura 22.- Arquitectura de un contador con salida de fin de cuenta

La decodificación del estado de fin de cuenta en un contador binario es la AND lógica de todos los bits de cuenta. Observando la estructura de un contador binario básico, podemos comprobar que la salida de la puerta AND de la célula correspondiente al bit de cuenta de mayor peso es, directamente, la salida de fin de cuenta: vale 1 en el último estado de cuenta (11111...11). Por ejemplo, la figura 23 muestra el circuito correspondiente a un contador binario de 4 bits con salida de fin de cuenta.

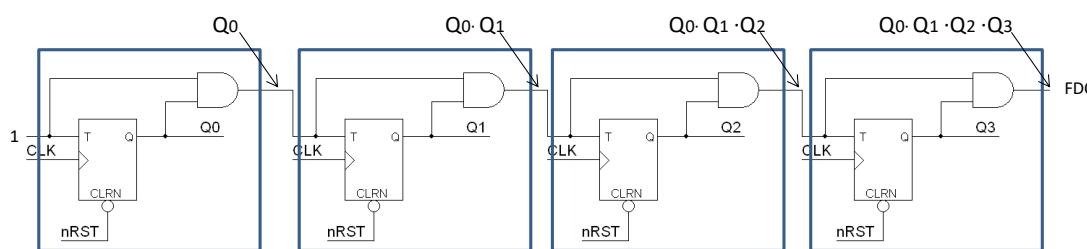


Figura 23.- Contador binario con salida de fin de cuenta

Cuando el contador tiene entrada de habilitación y salida de fin de cuenta, esta salida puede presentar dos modos de funcionamiento:

- Independiente de la entrada de habilitación: en este caso funciona de manera análoga a la descrita anteriormente, la salida se activa mientras el contador permanezca en el último estado de cuenta.
- Dependiente de la entrada de habilitación: la salida de fin de cuenta se activa únicamente cuando, estando el contador en su último estado de cuenta, la entrada de habilitación está activa.

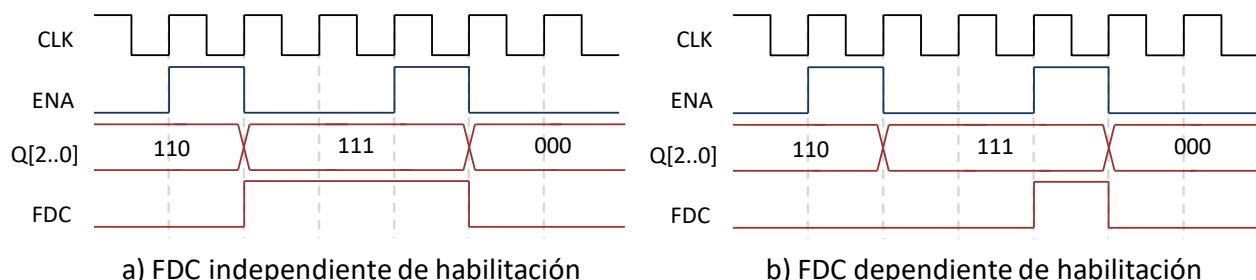


Figura 24.- Salida FDC en contadores con habilitación

La figura 24 ilustra la diferencia entre ambos modos de funcionamiento. En la figura 24a la salida de fin de cuenta depende únicamente del estado. La figura 25 muestra el diseño de una salida de fin de cuenta como

esta, utilizando como base un contador binario con entrada de habilitación; la salida se genera haciendo la AND lógica de todos los bits de cuenta.

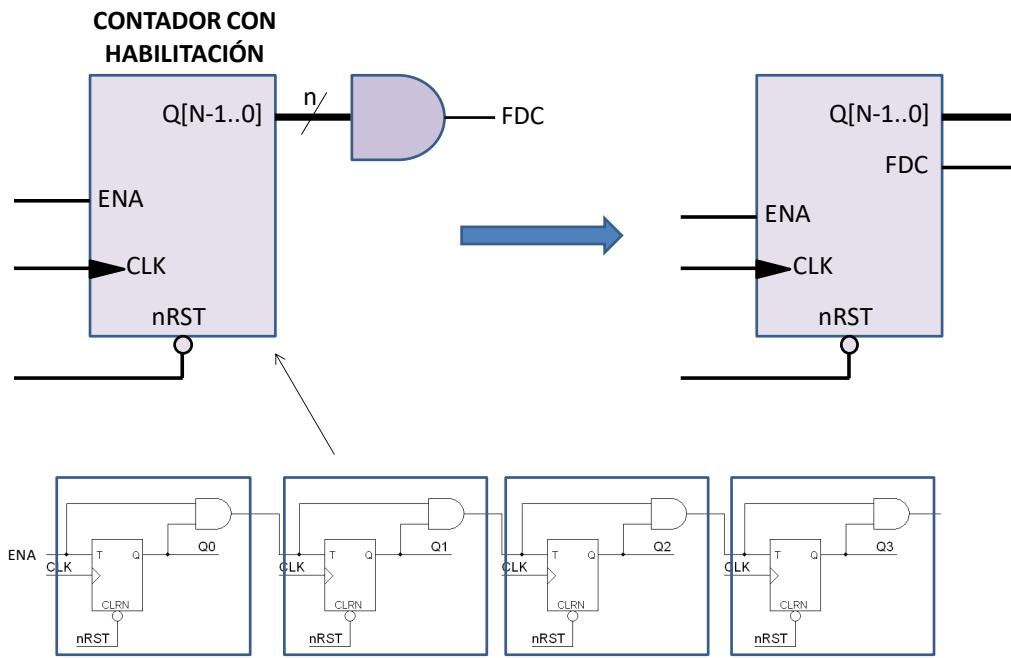


Figura 25.- Contador binario con salida FDC independiente de la habilitación

En la figura 24b, la salida de fin de cuenta depende del estado de cuenta y de la entrada de habilitación, por lo que se corresponde con la salida de un circuito con arquitectura de Mealy (Figura 26).

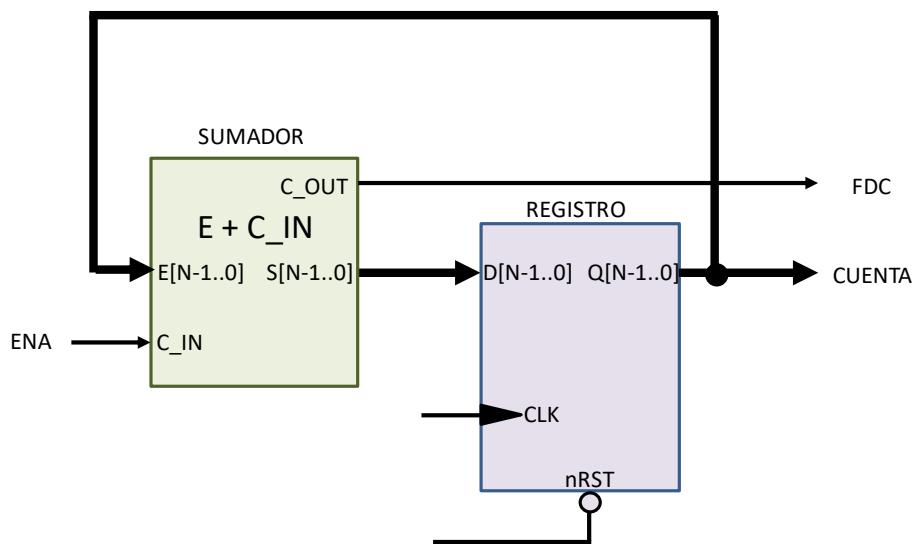


Figura 26.- Arquitectura de un contador con salida FDC dependiente de la habilitación

Observe que en el diagrama de bloques de la figura 26 el circuito sumador tiene una salida de acarreo, que sólo se activa cuando se sobrepasa la capacidad de representación de la suma, es decir, cuando todos los bits de la salida de cuenta están a 1 (en el último estado de cuenta) y el acarreo de entrada vale 1; el comportamiento de esta salida de acarreo se corresponde, por tanto, con el descrito para una salida de fin de cuenta dependiente de la habilitación, y puede obtenerse de la salida de la puerta AND de la célula

correspondiente al bit de mayor peso de un contador binario con salida de habilitación. En la figura 27 se muestra el circuito correspondiente a un contador de 4 bits.

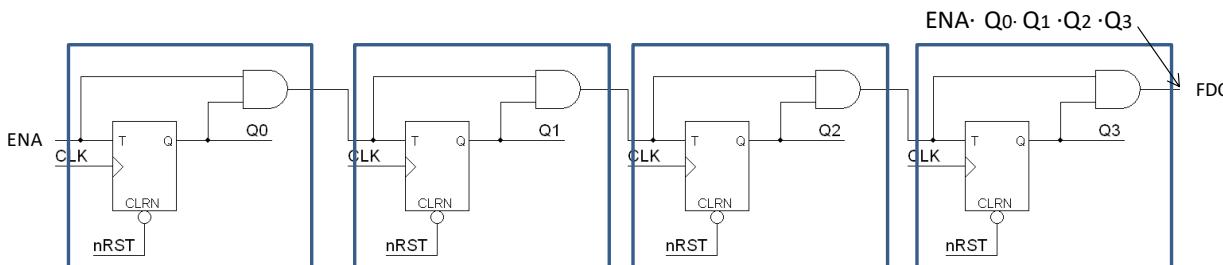


Figura 27.- Contador binario con salida de fin de cuenta dependiente de la habilitación

La ecuación de la salida FDC, $FDC = ENA \cdot Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0$ pone de manifiesto que sólo se activa cuando, estando el contador en el último estado de cuenta, la habilitación está activa.

Por la conexión que existe entre la entrada de habilitación de reloj y el acarreo de entrada del sumador, por una parte, y entre el acarreo de salida y la salida de fin de cuenta, por otra, es habitual que en los contadores con salida de fin de cuenta dependiente de la habilitación se denomine **entrada de acarreo** a la entrada de habilitación, y **salida de acarreo** a la salida de fin cuenta. Esta relación se ilustra sobre el esquema de la figura 28 que representa la versión *arquitectural* del circuito de la figura 27.

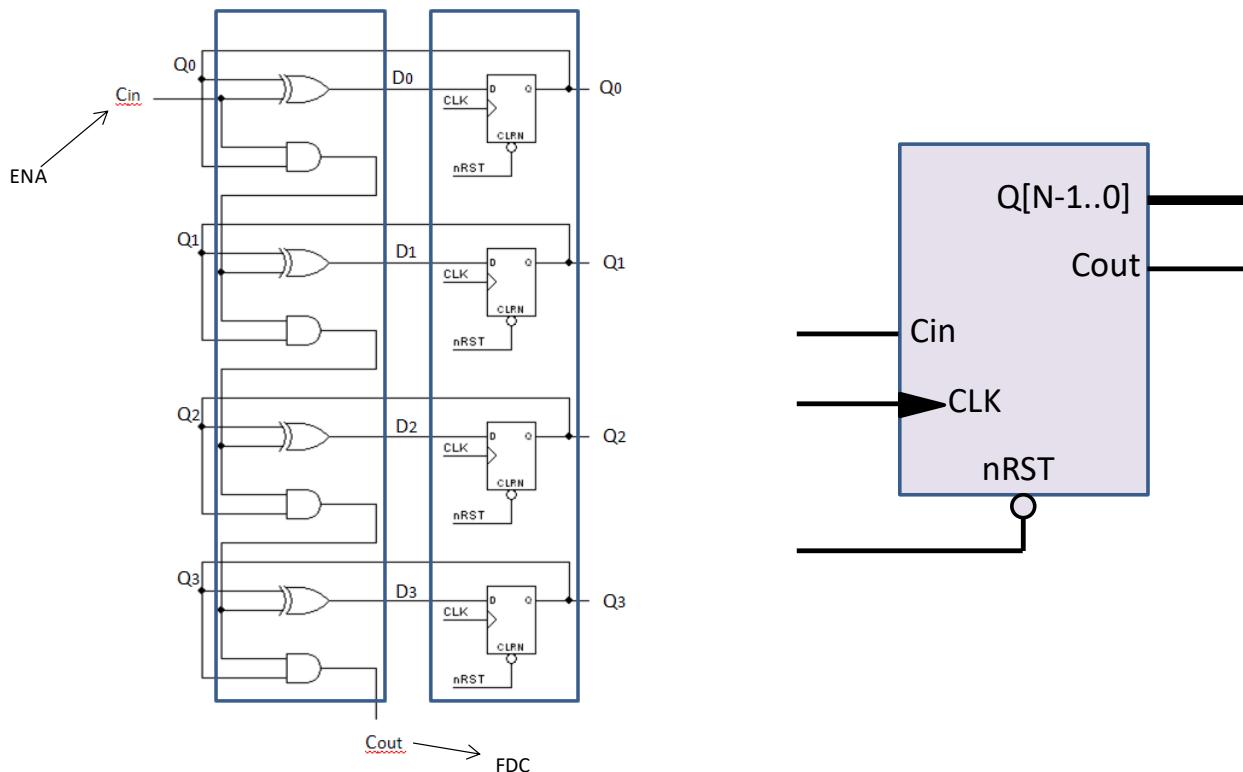


Figura 28.- Contador con entrada y salida de acarreo

En la figura 29 se muestran símbolos de los contadores con habilitación de reloj con salida de fin de cuenta independiente (figura 29a) y dependiente (figura 29b) de la habilitación, en los que se distingue la diferencia funcional que existe entre ambos por el nombre dado a la habilitación y fin de cuenta.

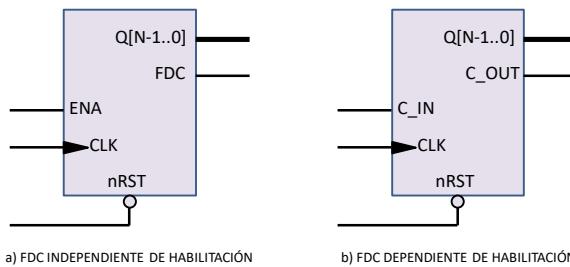


Figura 29.- Símbolos de contadores con habilitación y fin de cuenta

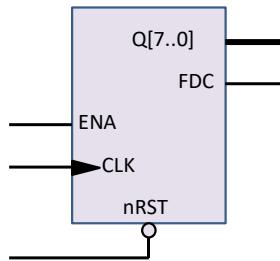
Ejercicios

8.- Diseñe un contador binario de 3 bits con entrada y salida de acarreo y entrada de reset asíncrono activo a nivel bajo, empleando flip-flops tipo D.

9.- Indique cómo modificaría el circuito anterior para construir un contador binario de 3 bits, sin entrada de habilitación y con salida de fin de cuenta.

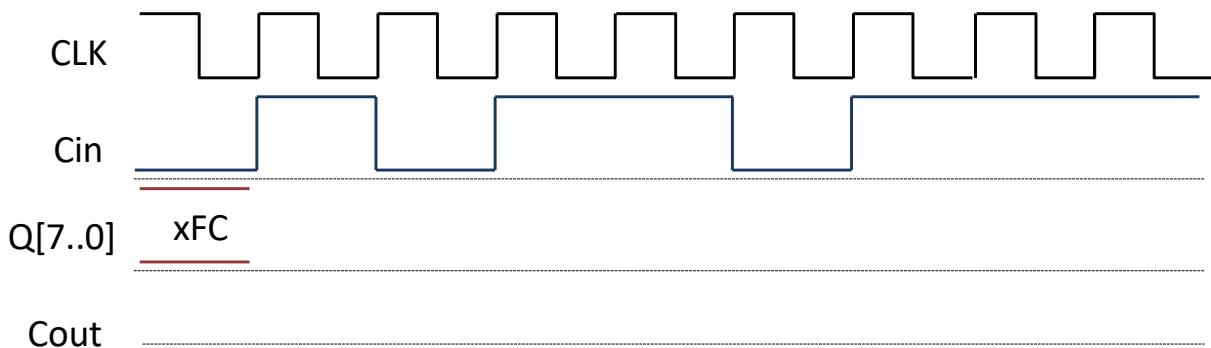
10.- Modifique el circuito realizado en el ejercicio 8 para realizar un contador de 3 bits con entrada de habilitación y salida de fin de cuenta independiente de la habilitación. Dibuje su símbolo.

11.- Realice un contador de 8 bits con entrada y salida de acarreo empleando un contador, con entrada de habilitación y salida de fin de cuenta independiente de la habilitación, y la lógica adicional que resulte necesaria.

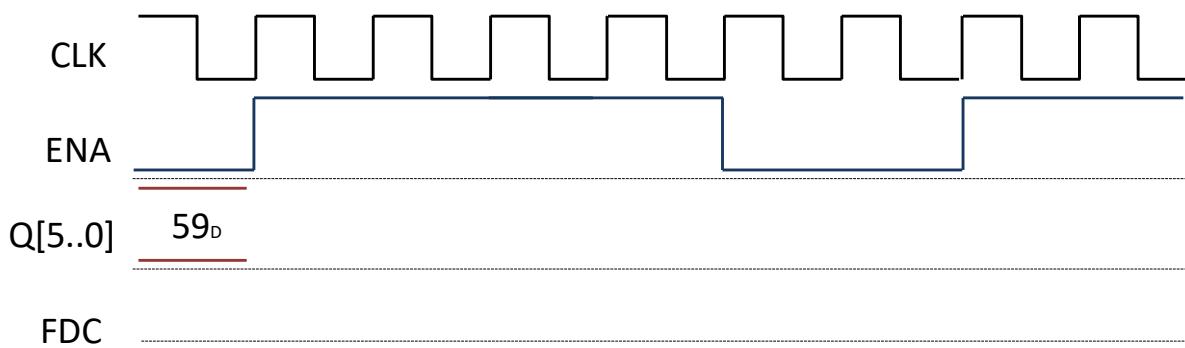


12.- Complete los cronogramas adjuntos sabiendo que representan el funcionamiento de:

- a) Un contador binario de 8 bits con entrada y salida de acarreo



- b) Un contador binario de 6 bits con entrada de habilitación y salida de fin de cuenta independiente de la habilitación



Contadores con entrada de reset o inicialización síncrona

Un contador puede disponer de una entrada de inicialización síncrona que sirva para llevarlo al estado inicial de cuenta en los flancos activos de reloj. Una entrada de reset síncrono *pone* el contador a cero; una entrada genérica de inicialización síncrona lo lleva a cualquier otro estado.

Para dotar a un contador con una entrada de reset síncrona basta con que el registro del estado cuente con un reset síncrono –si se desea que disponga de una entrada de inicialización síncrona en otro estado de cuenta, el registro debe estar realizado con una combinación de flip-flops con preset y reset síncrono conectados a la entrada de inicialización síncrona del registro. En la figura 30 se muestra la estructura de un contador con una entrada de reset síncrona activa a nivel alto y su símbolo.

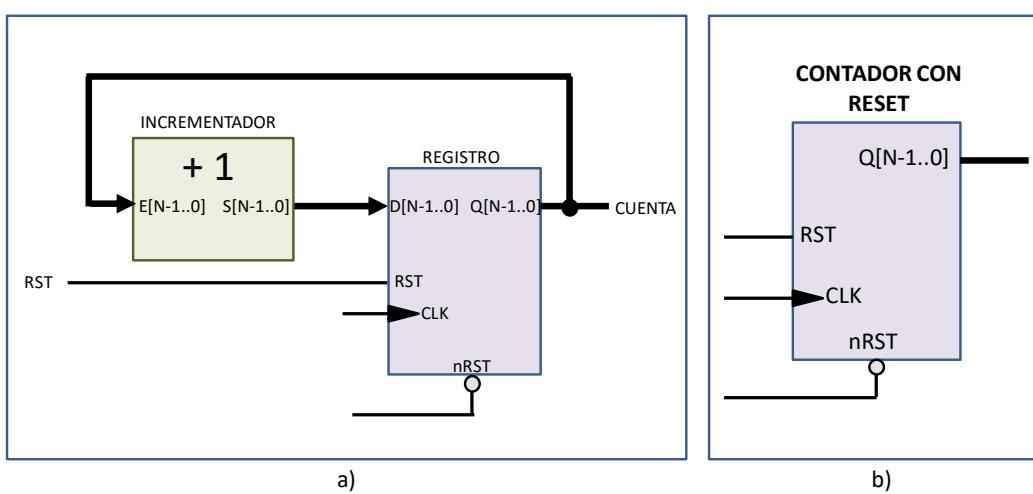


Figura 30.- Contador con reset síncrono

Un contador con reset síncrono altera su secuencia de cuenta cuando se activa la entrada de reset síncrono tal y como se muestra en la figura 31.

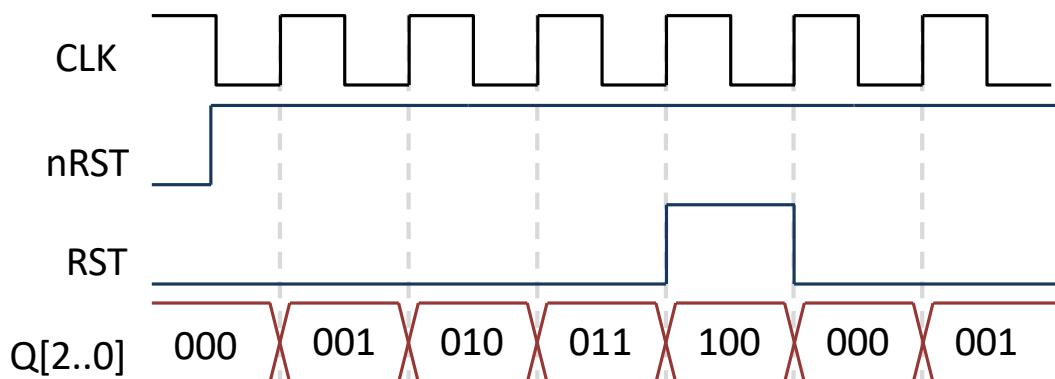


Figura 31.- Cronograma de funcionamiento de un contador con reset síncrono

Diseño de contadores con inicialización síncrona

La única modificación que hay que realizar sobre el circuito de un contador para que disponga de una entrada de inicialización síncrona consiste, precisamente, en utilizar flip-flops con una entrada de reset o preset. En la figura 32 se muestra la célula básica de los contadores que ya conoce, dotada de entrada de reset síncrono (figura 32a) o de preset síncrono (figura 32b).

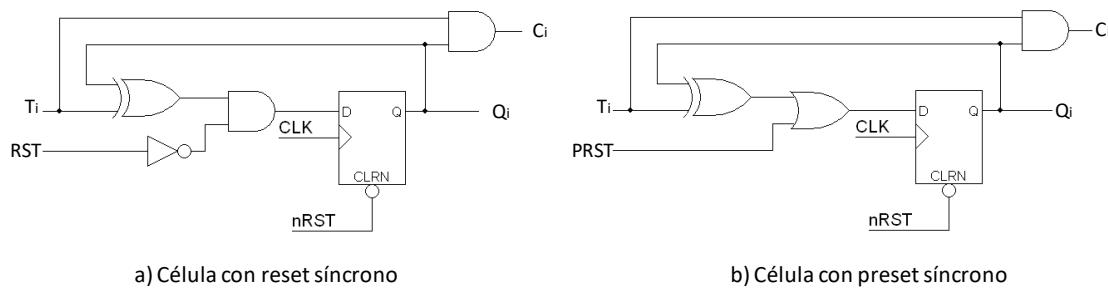


Figura 32.- Células básicas con entrada de inicialización síncrona

Las células a emplear para construir el contador dependen del estado de cuenta al que lleve la entrada de inicialización: por ejemplo, si es 0 todas las células del contador serían como la de la figura 32a, en cambio, si el estado de inicialización síncrona es el 1, la célula del bit de menor peso debe disponer de preset síncrono, como la célula de la figura 32b, y el resto de reset síncrono. La entrada de inicialización del contador debe conectarse a la entrada de inicialización síncrona (reset o preset) de todas las células.

Ejercicios

13.- Diseñe un contador de 3 bits con entrada y salida de acarreo, entrada de reset asíncrono activa a nivel bajo, y entrada reset síncrono activa a nivel alto e independiente de la habilitación (del acarreo de entrada).

14.- Modifique el circuito anterior para que la actividad del reset esté supeditada a la habilitación (al acarreo de entrada).

15.- Diseñe un contador binario de 3 bits, sin habilitación, y con una entrada de inicialización síncrona que inicialice el contador a 3 (011b).

Diseño de contadores módulo M

La arquitectura de contadores que se ha presentado hasta aquí sólo permite realizar contadores cuyo módulo sea potencia de 2 (contadores binarios) puesto que, debido al modo en que opera el circuito incrementador –o sumador en los contadores con habilitación-, cuentan desde 0 hasta el valor más alto que puede expresarse con el número de bits de salida del contador (N), y repiten esta secuencia de cuenta indefinidamente. Su módulo, que es igual al número de estados diferentes de cuenta, es, como ya sabe, 2^N .

Para realizar un contador con un módulo que no sea potencia de dos, hay que utilizar como base un contador binario con reset síncrono. La arquitectura para el diseño de estos contadores se representa en la figura 33.

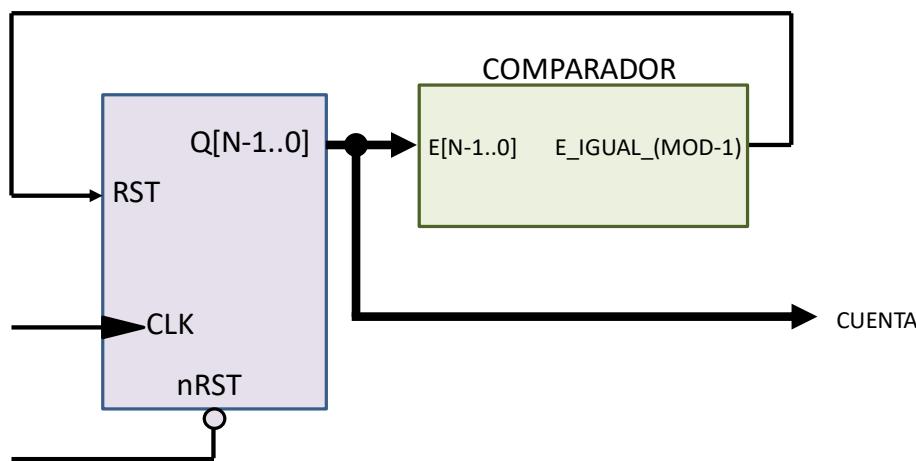


Figura 33.- Contador de módulo M

Para diseñar un contador de módulo M que no sea potencia de dos hay que:

- Realizar un contador binario de N bits con reset síncrono, tal que $2^N > P$. Normalmente se utiliza el contador con el menor número de bits que cumpla esta condición: por ejemplo, para realizar un contador de módulo 10 se utilizaría un contador binario de 4 bits ($2^4 = 16$), o para realizar un contador de módulo 300, un contador binario de 9 bits ($2^9 = 512$).

- Realizar un comparador¹, que detecte, activando su salida, el ciclo de reloj en que el contador binario alcanza el estado de cuenta M-1.
- Conectar la salida del comparador a la entrada de reset del contador. De este modo el estado de cuenta siguiente al M-1 (el estado en que se activa el reset síncrono) será el 0 y el contador resultante tendrá una secuencia de cuenta 0, 1, 2...M-1, 0, 1, 2...M-1, etcétera, que se repite indefinidamente y corresponde a un contador de módulo M.

Ejemplo:

En la figura 34a se muestra la estructura de un contador de módulo 6 realizado en base a un contador binario de 3 bits. En el cronograma de la figura 34c se representa su funcionamiento: cuando el contador alcanza el estado de cuenta 5 (el módulo menos 1) se activa la entrada de reset del contador binario y, en el siguiente flanco activo de reloj, el estado de cuenta pasa a ser el 0. La secuencia de cuenta resultante es 0, 1, 2, 3, 4, 5, 0, 1,... etcétera.

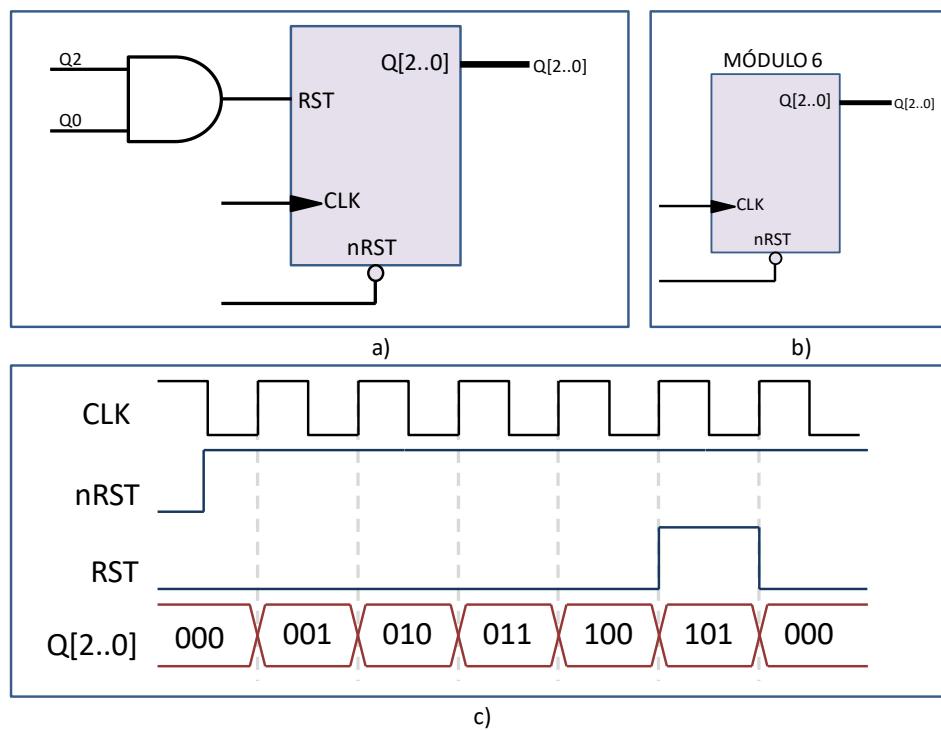


Figura 34.- Contador módulo 6

Observe que la salida del comparador que se conecta a la entrada de reset es una salida de fin de cuenta, ya que se activa en el último estado de cuenta del contador. Por tanto, si se precisa que el contador disponga de esa salida no es necesario añadir lógica adicional al contador, basta con utilizar la señal de salida del comparador.

Si se desea que el contador de módulo M tenga habilitación de reloj, el contador binario sobre el que se construye debe también tenerla y el reset debe activarse únicamente cuando, siendo el estado de cuenta el M-1, el contador esté habilitado, lo que obliga a que la condición de reset sea que el contador esté en el último estado de cuenta y la habilitación valga 1.

¹ Es un comparador con una constante, por lo que sólo tiene una entrada de datos. La función que realiza puede identificarse también como una decodificación y, por tanto, también se podría haber considerado al módulo como un decodificador.

En la figura 35 se muestra un ejemplo de diseño: el de un contador módulo 10 con entrada de habilitación.

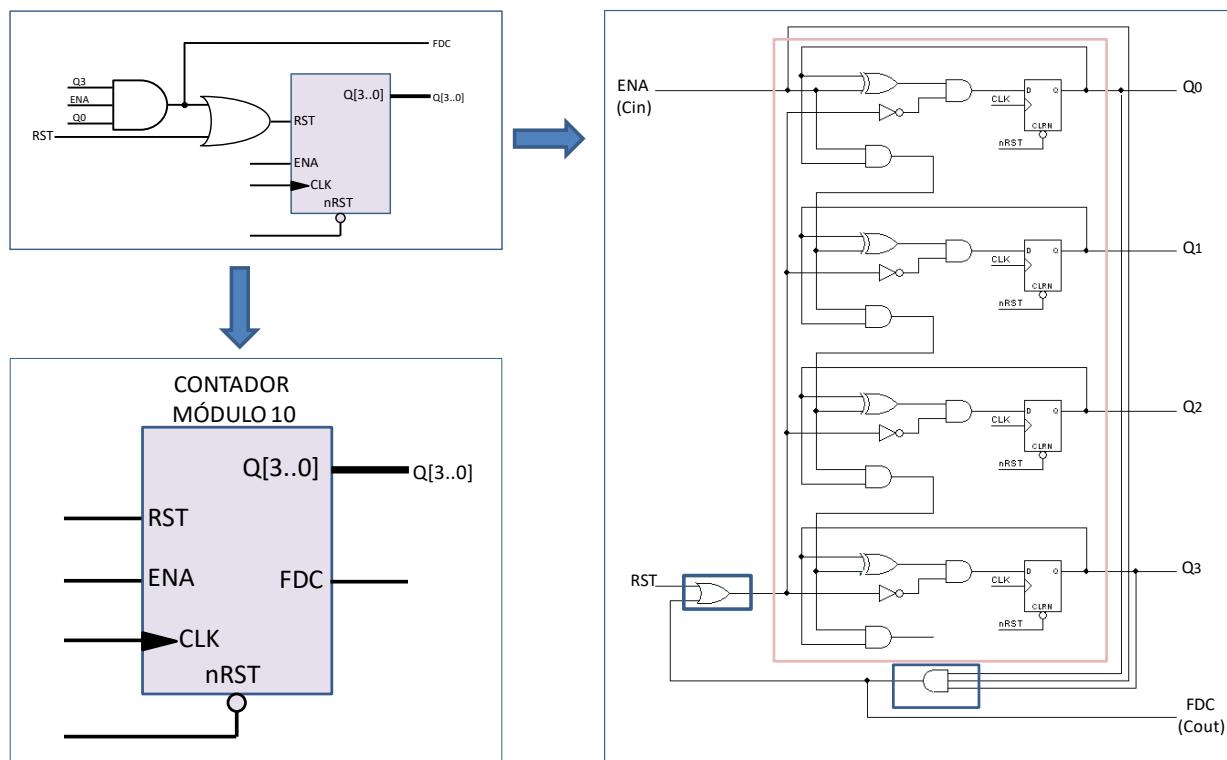


Figura 35.- Contador módulo 10

En este circuito pueden observarse varios detalles interesantes:

- La señal de fin de cuenta es dependiente de habilitación y es la señal que resetea síncronamente el contador binario de 4 bits para obtener una cuenta de módulo 10.
- Con una puerta OR se puede dotar al contador módulo 10 de una entrada de reset síncrono.
- La condición de detección del último estado se simplifica porque el contador nunca pasa por los estados 10, 11, 12, 13, 14 y 15.

Ejercicios

16.- Dibuje un diagrama de bloques como el de la figura 34a para definir el diseño de un contador binario de módulo 100, que cuente de 0 a 99.

17.- Indique las modificaciones que habría que introducir en el circuito del ejercicio anterior para que el contador de módulo 100 cuente desde 1 hasta 100.

18.- Modifique el contador del ejercicio 16 para que el contador resultante tenga habilitación de reloj y salida de fin de cuenta dependiente de la habilitación (acarreo de entrada y salida). Fíjese en el circuito de la figura 35.

Diseño de contadores de módulo programable

En ocasiones es necesario realizar contadores cuyo módulo pueda ser controlado por una entrada de control síncrono. La arquitectura e interfaz de este tipo de contadores se muestra en la figura 36.

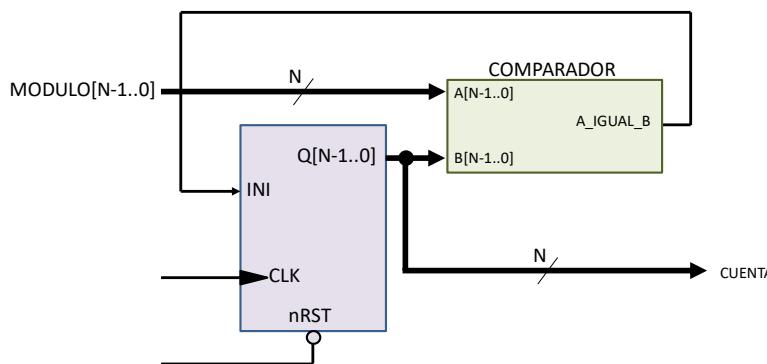


Figura 36.- Arquitectura de un contador con módulo programable

La arquitectura del contador de módulo programable incluye un comparador que activa su salida cuando el estado de cuenta es igual al valor ingresado en la entrada MODULO. Dicha salida, que está conectada a la entrada de inicialización del contador binario, reinicializa la cuenta de éste síncronamente: para que el módulo resultante sea el indicado en la entrada MODULO, es necesario que la inicialización lleve al contador al estado de cuenta 1 –si lo llevara a 0, el módulo sería MODULO + 1. En el cronograma de la figura 37 se muestra el esquema de un contador de módulo programable de 4 bits.

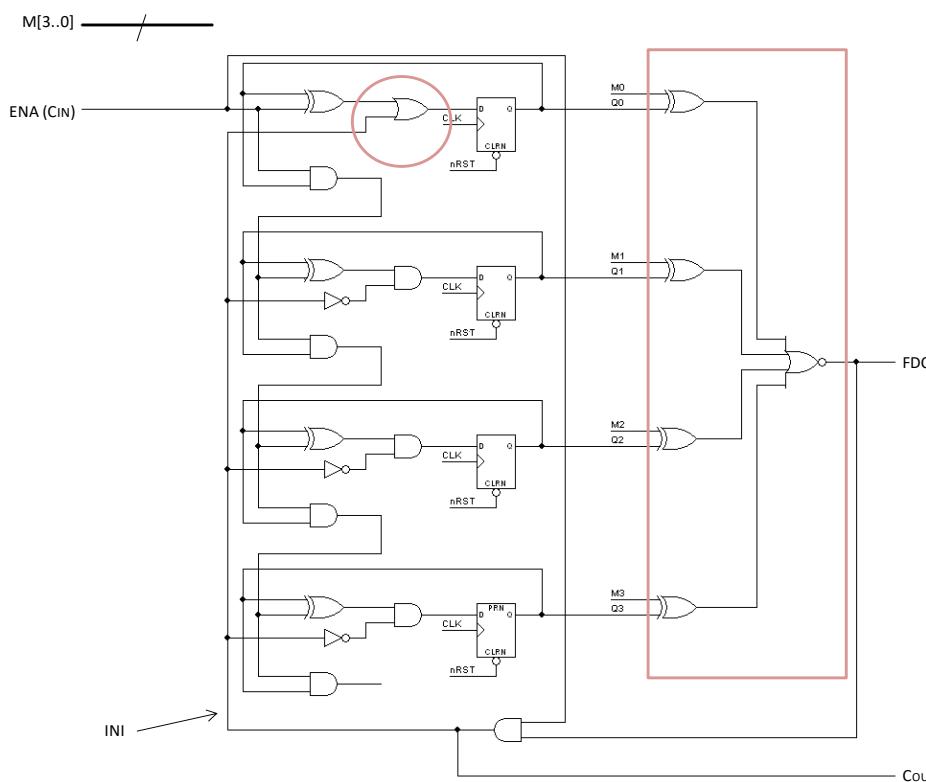


Figura 37.- Contador de módulo programable de 4 bits

En el esquema puede observarse que la célula asociada al bit de menor peso tiene, a diferencia del resto, preset síncrono, por lo que la señal INI fuerza, en los flancos activos de reloj, el estado de cuenta 0001. En el esquema están señaladas también las dos señales que pueden utilizarse para generar una salida de fin de

cuenta -que se activara cuando el estado de cuenta sea igual al valor ingresado en la entrada que programa el módulo, M- dependiente e independiente de la habilitación.

El cronograma de la figura 38 representa el funcionamiento del circuito cuando la entrada M vale 3.

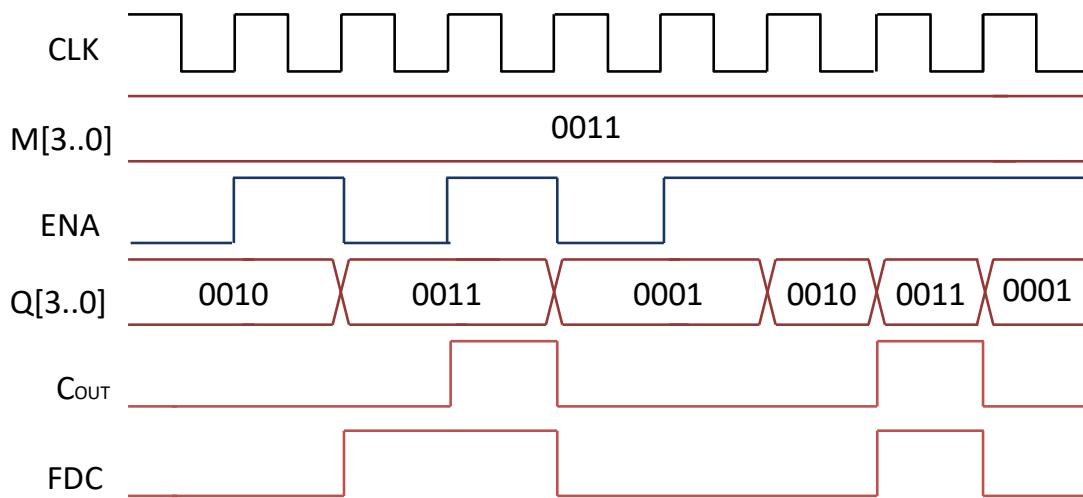


Figura 38.- Funcionamiento de un contador de módulo programable

Ejercicio

19.- Diseñe un comparador de igualdad de 8 bits. Tome como ejemplo el circuito de la figura 37.

20.- Utilizando el comparador de igualdad como bloque funcional y un contador con entrada de inicialización síncrona de 8 bits, construya un contador de módulo programable con salida de fin de cuenta. Indique el valor de inicialización de la entrada de control síncrona.

Contadores descendentes y contadores reversibles

Un contador puede realizar la cuenta en sentido descendente, de modo que a cada estado de cuenta le siga el estado cuyo número le precede en la serie de números naturales; por ejemplo, un contador de módulo 6 descendente tendría la siguiente sucesión de estados de cuenta: 5, 4, 3, 2, 1, 0, 5, 4... etcétera. Todo lo dicho en relación con el funcionamiento y estructura de los contadores binarios (ascendentes) y sus entradas de control síncronas nos permite deducir, por analogía, las propiedades, estructura y funcionamiento de los contadores descendentes:

- El estado de cuenta de un contador binario descendente se decremente en cada flanco activo de reloj y la secuencia de cuenta se repite periódicamente, con un periodo, en número de ciclos de reloj, que es igual al módulo (número de estados de cuenta) del contador; cuando se alcanza el último estado de cuenta, el 0 (todos los bits de cuenta a 0), la secuencia continúa decrementándose a partir del mayor estado de cuenta.
- Un contador descendente con habilitación de reloj sólo decrementa su cuenta cuando está habilitado.
- La salida de fin de cuenta de un contador descendente se activa cuando el contador pasa por el último estado de cuenta, el 0, y en el caso de que el contador disponga de entrada de habilitación, puede ser dependiente o independiente de la habilitación.
- Un contador descendente puede tener una entrada de reset o de inicialización síncrona cuya activación dé lugar a que el contador evolucione a un estado inicial de cuenta; normalmente una entrada de reset síncrona de un contador descendente lo lleva al primer estado de cuenta (todos los bits de cuenta a 1), y no a 0 como en el caso de los contadores ascendentes.
- En la arquitectura de un contador descendente, en lugar de calcularse el estado futuro de cuenta con un incrementador (o, si el contador tiene habilitación, un sumador del acarreo de entrada al estado de cuenta actual), se utiliza un decrementador (o un restador que sustrae del estado de cuenta actual el valor del acarreo de entrada, si el contador tiene habilitación). En la figura 39 se muestra la diferencia existente entre la estructura de un contador ascendente y descendente, con y sin habilitación de reloj.

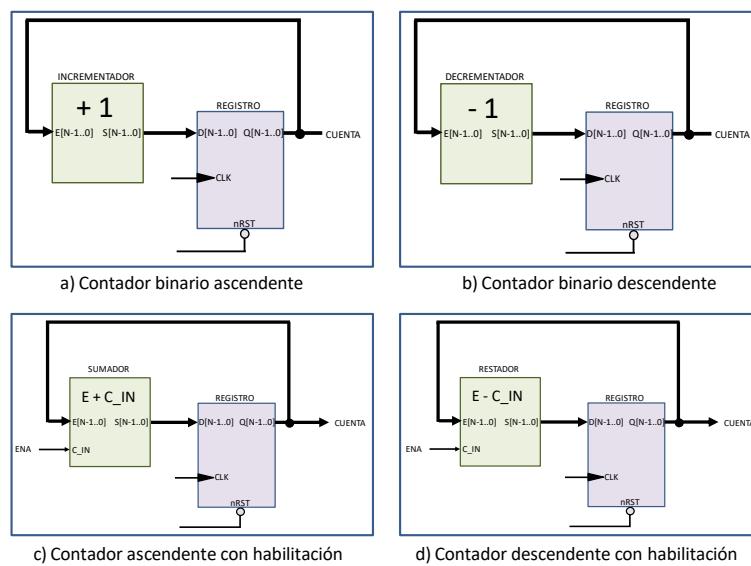


Figura 39.- Arquitectura de contadores ascendentes y descendentes

El decrementador de la figura 39b calcula el siguiente estado de cuenta restando 1 al estado de cuenta actual; operando en complemento a dos, la resta puede realizarse como la suma del estado actual y el complemento a 2 de 1 (1111...1111). Cuando se alcanza el último estado de cuenta de un contador descendente (el 0), el contador evoluciona al mayor estado de cuenta (todos los bits de salida a 1):

$$\begin{array}{r}
 \text{n bits} \\
 \overbrace{\quad\quad\quad\quad\quad\quad}^{\text{000000....0000}} \\
 +111111....1111 \\
 \hline
 \overbrace{\quad\quad\quad\quad\quad\quad}^{\text{111111....1111}}
 \end{array}$$

n bits

Y continúa decrementándose a partir de ese estado.

El restador de la figura 39d realiza la operación anterior cuando el acarreo es 1 y, cuando vale 0, resta al estado actual de cuenta el 0 (sumándole el complemento a 2 de 0 que también es cero), manteniéndose la cuenta; por tanto en este circuito la entrada de acarreo hace las veces de una entrada de habilitación.

Diseño de contadores descendentes

En la sucesión de estados de una cuenta descendente, un bit de cuenta, de peso i , comuta, al pasar de un estado (A) a su antecesor ($A-1$), cuando todos los bits de peso menor que i valen 0. El bit de menor peso comuta en todos los ciclos de reloj. Por ello, al igual que en el caso de los contadores ascendentes, los contadores descendentes pueden diseñarse en base a una célula básica. En la figura 40 se muestra la célula básica y la estructura de un contador binario descendente básico.

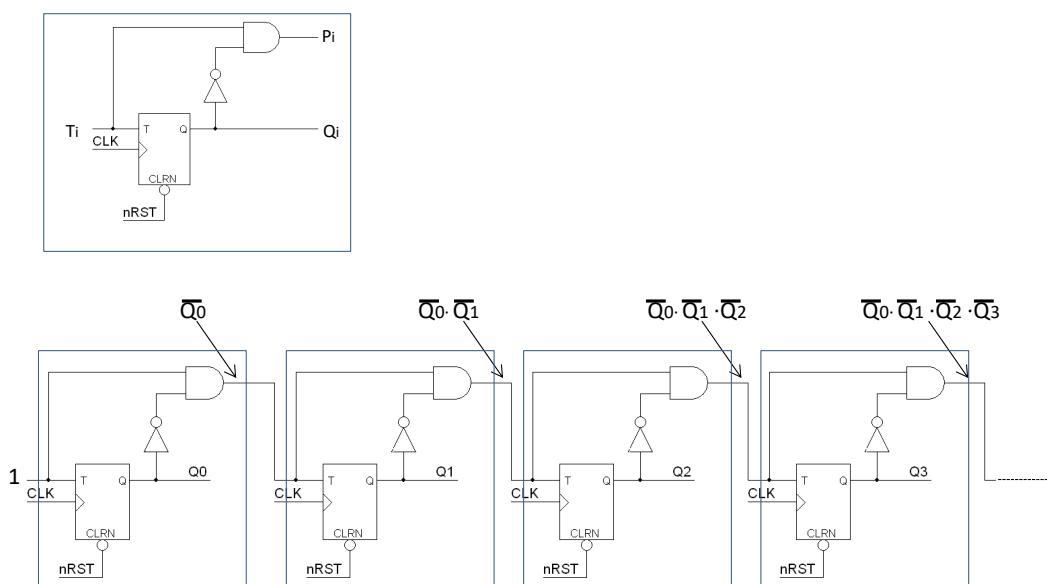


Figura 40.- Contador binario descendente

El flip-flop tipo T de cada célula (excepto el de la correspondiente al bit de menor peso) comuta únicamente cuando todos los bits de cuenta de peso menor que el de su célula están a 0. La puerta AND de cada célula añade un bit de cuenta negado al producto que se propaga para controlar las entradas T de los flip-flops. El funcionamiento es, en definitiva, análogo al de un contador ascendente.

Si las células se representan por su realización con flip-flops tipo D, se puede distinguir la arquitectura de la figura 39b, donde se representa el contador como la combinación de un registro y un decrementador,

realizado, en este caso, como un restador del acarreo de entrada –con el acarreo puesto a 1. La célula básica del restador de acarreo y la estructura de conexión para construirlo se muestra en la figura 41.

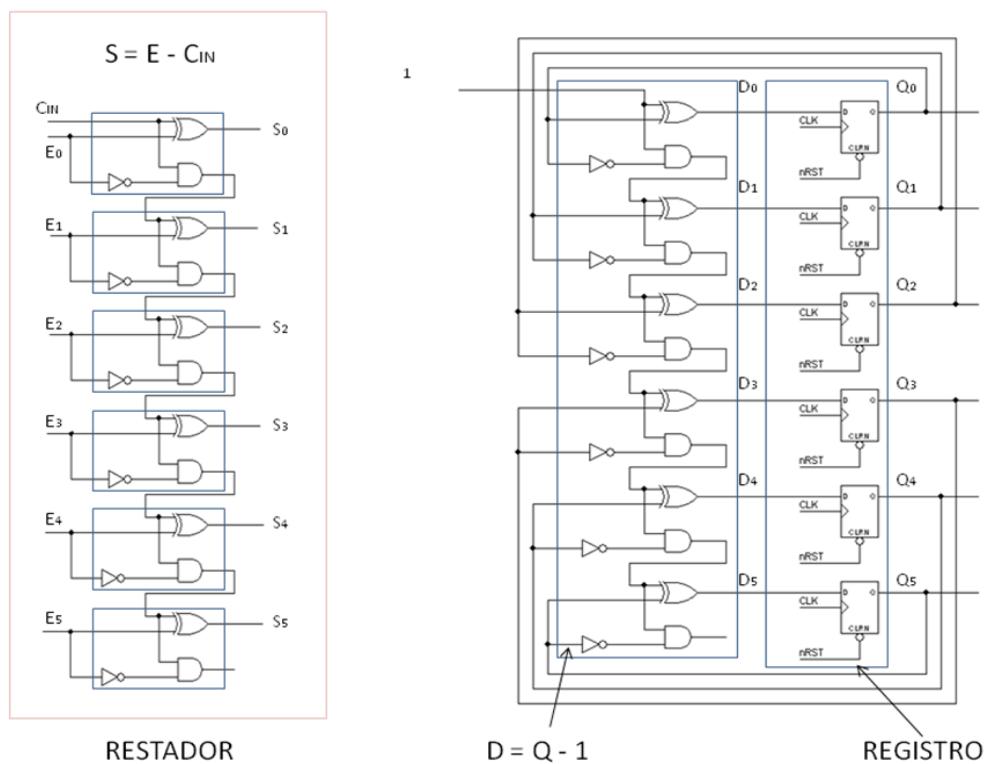


Figura 41.- Diseño de la arquitectura de un contador descendente

En la figura 41 se representa también la realización de un contador descendente de 6 bits empleando el restador, con la entrada de acarreo a 1, y un registro.

La realización de contadores descendentes con entradas síncronas y salida de fin de cuenta se realiza, a partir del circuito de la figura 40, repitiendo las modificaciones que se describieron en el caso de los contadores ascendentes.

Si se desea realizar un contador descendente con habilitación de reloj, basta con conectar a la entrada de la célula correspondiente al bit de menor peso de cuenta, en vez de un 1, como en la figura 40, la entrada de habilitación; si se desea que tenga salida de fin de cuenta dependiente de la habilitación, se obtiene de la salida de la puerta AND correspondiente al bit de mayor peso. En la figura 42 se muestra el esquema de un contador descendente con entrada de habilitación y salida de fin de cuenta.

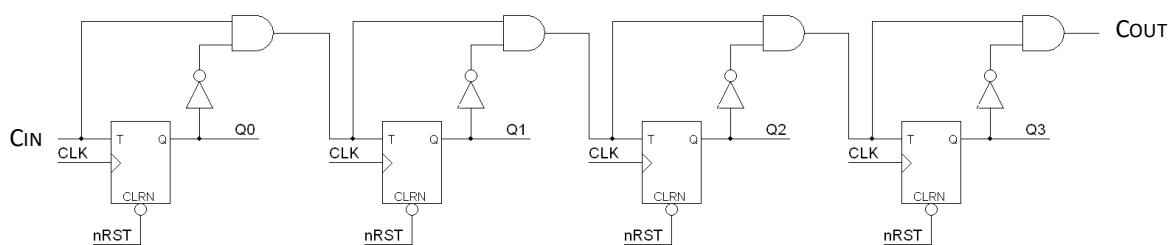


Figura 42.- Contador binario con salida FDC (C_{OUT}) dependiente de la habilitación (C_{IN})

Si se desea dotar al contador de una salida de fin de cuenta independiente de la habilitación, basta con generarla como la NOR lógica de todos los bits de cuenta; si se desea que tenga una entrada de inicialización síncrona, se modifican las células básicas para que dispongan de entrada de reset o preset síncrono, etcétera.

Ejercicio

21.- Diseñe, utilizando flip-flops tipo D y puertas lógicas, un contador descendente de 4 bits, de módulo programable con entrada de habilitación y salida de fin de cuenta.

Contadores reversibles

Hay pocas aplicaciones en las que resulte idóneo el uso de contadores descendentes –pocas en comparación con las que requieren el uso de un contador ascendente, o con aquellas en que resulta indiferente el uso de uno u otro tipo de contadores (en cuyo caso suele optarse por el uso de contadores ascendentes). Pero sí es más frecuente la necesidad de disponer de un contador que pueda contar en ambos sentidos, de un contador reversible. El sentido de cuenta de este tipo de contadores se controla mediante una entrada síncrona de control que suele denominarse entrada de *up/down*.

En la figura 43 se muestra un cronograma que representa el funcionamiento de un contador reversible de 3 bits.

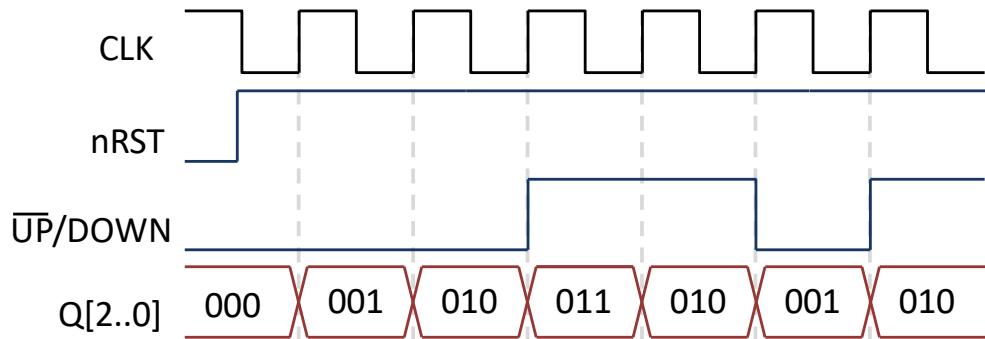


Figura 43.- Cronograma de funcionamiento de un contador reversible

En el cronograma puede observarse como la cuenta se incrementa en los flancos activos de reloj en que la entrada de control UP/DOWN vale 0, mientras que se decrementa en los que vale 1.

Este tipo de contadores suele utilizarse para contar y descontar la ocurrencia de eventos, pero no para medir tiempos.

Ejemplo: Un contador reversible puede utilizarse para contar el número de personas que hay dentro de un recinto, controlado por dos señales: una señal de habilitación que se activa durante un periodo de reloj cada vez que alguien entra o sale y otra que, simultáneamente con la señal anterior, indica si se trata de una entrada o salida (para utilizarla como entrada de control de *up/down*). Cada vez que alguien entra el contador incrementa la cuenta; cada vez que alguien sale la decrementa.

Un contador reversible puede tener las mismas entradas síncronas que los contadores ascendentes y descendentes (de habilitación y de reset o inicialización) y puede contar también con una salida de fin de cuenta; la salida de fin de cuenta de un contador reversible se activa en el último estado de cuenta, que varía en función del sentido de cuenta: un contador reversible que cuente entre 0 y 9 activa su salida de fin de cuenta en el estado 9 cuando cuenta en sentido ascendente y en 0 cuando cuenta en sentido descendente.

Diseño de contadores reversibles

En las arquitecturas de los contadores reversibles la lógica de cálculo del siguiente estado se realiza con un circuito aritmético capaz de sumar o restar en función de una señal de control (gobernada por la entrada de *up/down* del contador). El resto de entradas que puede tener son análogas a las ya conocidas para los diversos tipos de contadores ascendentes y descendentes, por lo que no nos vamos a detener a describirlas. A modo de ejemplo se muestra, en la figura 44, la célula básica para realizar un contador reversible y la realización de un contador reversible de 4 bits con entrada de habilitación y salida de fin de cuenta dependiente de la habilitación.

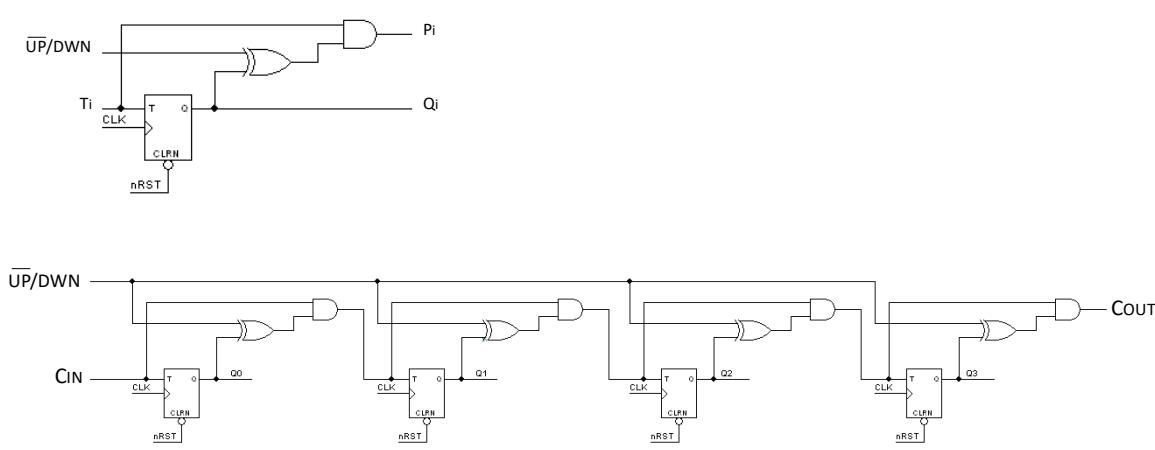


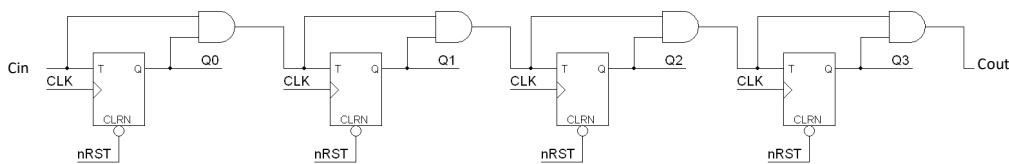
Figura 44.- Contador binario reversible de 4 bits con salida de fin de cuenta dependiente de la habilitación

Ejercicio

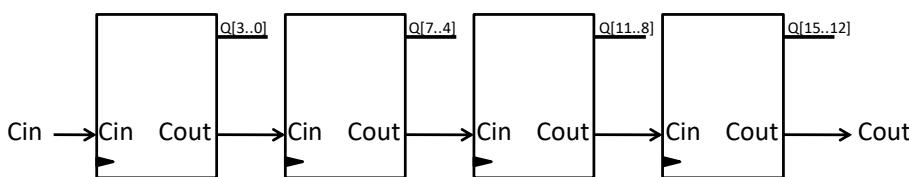
22.- Demuestre que la célula básica para la realización de un contador reversible equivale a la que se utiliza para construir contadores ascendentes, cuando la entrada de control UP/DWN vale 0, y a la que se emplea para construir contadores descendentes cuando la susodicha entrada vale 1.

Encadenamiento de contadores

Un grupo de módulos contadores con entrada y salida de acarreo pueden encadenarse para la realización de contadores con una mayor capacidad de cuenta. En la figura 45 se muestra el esquema para la realización de un contador binario de 16 bits empleando contadores binarios de 4 bits.



a) Contador binario de 4 bits con entrada y salida de acarreo



b) Contador binario de 16 bits con entrada y salida de acarreo

Figura 45.- Ampliación de contadores binarios

En este caso, considerando la estructura de cada módulo de 4 bits (figura 45a) resulta obvio que la estructura de la figura 45b implica el encadenamiento de 16 células y, por tanto, el resultado es un contador binario de 16 bits. Al margen de esta evidencia, el análisis del esquema puede interpretarse desde otra perspectiva: la cuenta de un módulo sólo puede incrementarse cuando todos los módulos que llevan la cuenta de bits de menor peso que él están en el último estado de cuenta y, están, además, habilitados, ya que sólo en estas condiciones su entrada de acarreo estará activa. Por ejemplo, el módulo que lleva la cuenta de los bits Q[11..8], en el contador de la figura 45b, sólo está habilitado ($Cin = 1$) cuando el módulo que tiene a su izquierda está en el último estado de cuenta (1111b) y habilitado, y si está habilitado, es porque el módulo que lleva la cuenta de los bits Q[3..0] está en el último estado de cuenta (1111) y habilitado; en resumen: la cuenta de los bits Q[11..8] sólo se incrementa –como debe ser– cuando los bits Q[7..0] valen todos 1 y la entrada de habilitación del contador de 16 bits (Cin) vale 1.

Ejercicios

23.- Basándose en el razonamiento realizado en el párrafo anterior, no debería tener ningún problema para contestar a la pregunta que se le va a formular a continuación:

- El esquema de conexión de la figura 45b **no** puede utilizarse si los módulos son contadores con salida de fin de cuenta independiente de la habilitación ¿Por qué?

24.- Realice un contador de 16 bits empleando módulos contadores de 4 bits con entrada de habilitación y salida de fin de cuenta independiente de la habilitación, y la lógica adicional necesaria que precise.

Ayuda: Con una puerta AND de dos entradas asociada a cada módulo puede corregir el defecto que tienen los contadores con salida FDC independiente de la habilitación a la hora de realizar conexiones como la de la figura 45b.

Contadores decimales

Los contadores decimales expresan el estado de cuenta en código BCD. En la figura 46 se muestra un ejemplo que permite comprobar la diferencia existente entre la secuencia de cuenta de un contador BCD y un contador binario.

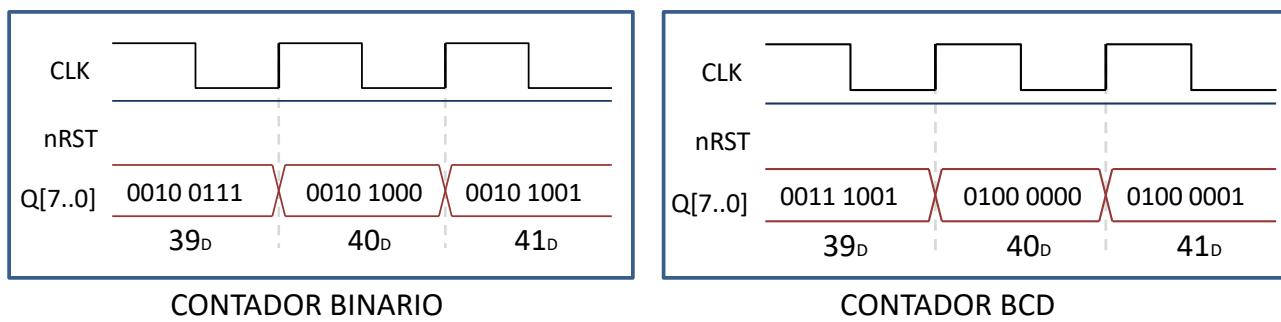


Figura 46.- Secuencia de cuenta BCD

El módulo de cuenta de un contador BCD depende del número de dígitos que tenga; por ejemplo:

- Un contador decimal de 1 dígito cuenta de 0 a 9 y tiene módulo 10
- Un contador decimal de 2 dígitos cuenta de 0 a 99 y tiene módulo 100
- Un contador decimal de 5 dígitos cuenta de 0 a 99.999 y tiene módulo 100.000

Por lo demás, los contadores decimales presentan la misma variedad de opciones que los binarios: pueden contar en sentido ascendente o descendente, o ser reversibles, y pueden tener entrada de habilitación y salida de fin de cuenta (dependiente o independiente de la habilitación) y entradas de inicialización síncrona.

Diseño de contadores decimales

Un contador decimal de un dígito se construye realizando un contador binario de módulo 10 que cuente desde 0 hasta 9. El contador de la figura 35 (que reproducimos aquí para su comodidad) es un contador decimal de un dígito, con entrada y salida de acarreo y entrada de reset síncrono activa a nivel alto.

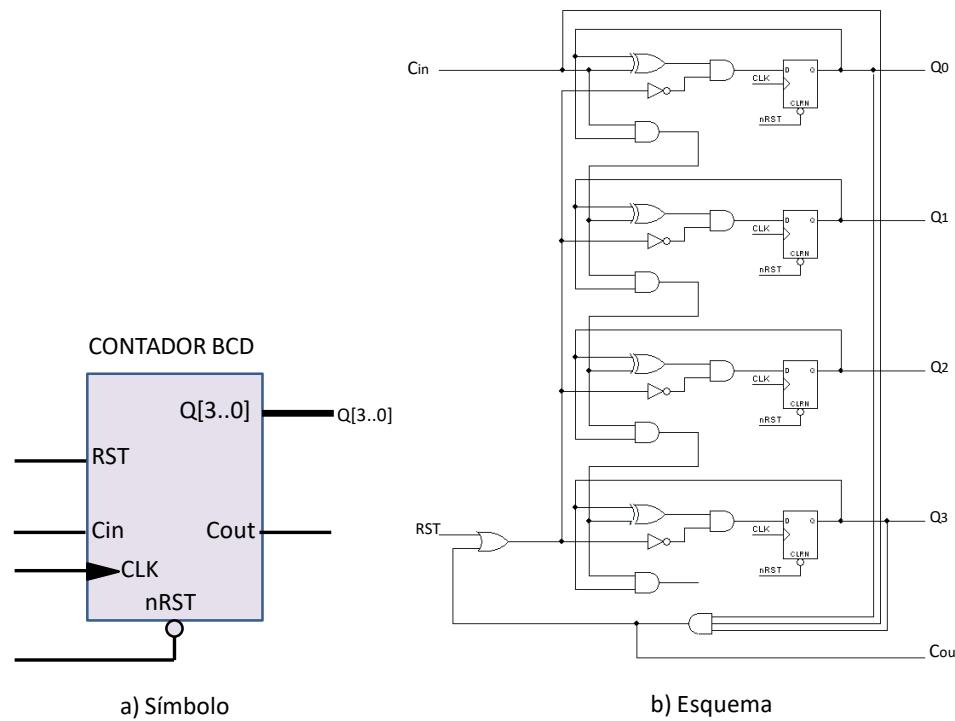


Figura 35 (repetida)

Para realizar un contador decimal de N dígitos hay que utilizar N contadores decimales de 1 dígito con entrada y salida de acarreo (como el de la figura 35, por ejemplo), y conectarlos en serie, de acuerdo con el esquema empleado para la realización de contadores binarios con módulos contadores de un menor número de bits (figura 45b). En la figura se muestra la estructura de conexión.

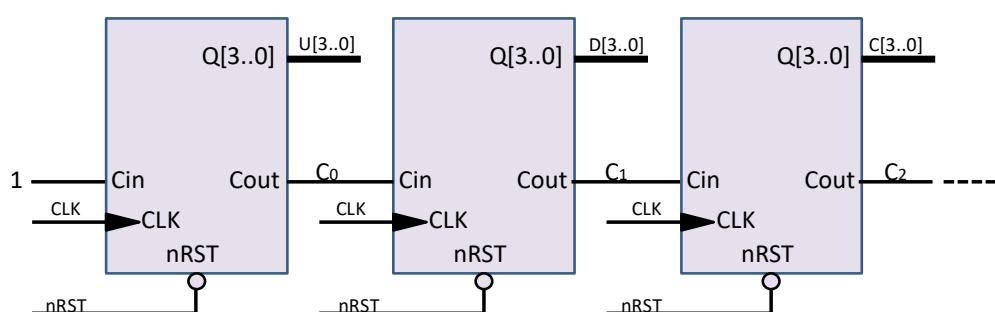


Figura 47.- Contador decimal de N dígitos

La conexión que se observa en este circuito se deriva de la regla que regula el incremento de un dígito de peso i en un cuenta decimal: se incrementa sólo cuando todos los dígitos de peso menor que él valen 9, salvo el dígito de menor peso (las unidades) que se incrementa siempre; por ejemplo:

- el dígito de las decenas se incrementa sólo cuando las unidades valen 9 (del 9 al 10, del 19 al 20, del 29 al 30, etcétera)

- el dígito de las centenas se incrementa sólo cuando las unidades y las decenas valen 9 (del 99 al 100, del 199 al 200, del 299 al 300, etcétera)
- el dígito de los millones se incrementa sólo cuando las unidades, decenas, centenas, millares, decenas de millar y centenas de millar valen 9 (de 999.999 a 1.000.000, de 1.999.999 a 2.000.000, etcétera)

En el circuito de la figura 47 el peso del dígito decimal correspondiente a cada módulo aumenta de izquierda a derecha (el de la izquierda cuenta las unidades, el siguiente a la derecha las decenas, el siguiente las centenas, etcétera). La estructura de conexión del circuito determina que para que un módulo incremente su cuenta, su entrada de acarreo debe valer 1 y, para que esto ocurra, todos los módulos que tiene a la izquierda (los que llevan la cuenta de los dígitos con menor peso que el suyo) deben estar en el último estado de cuenta (1001b) y habilitados. Esta condición se ilustra en la figura 48.

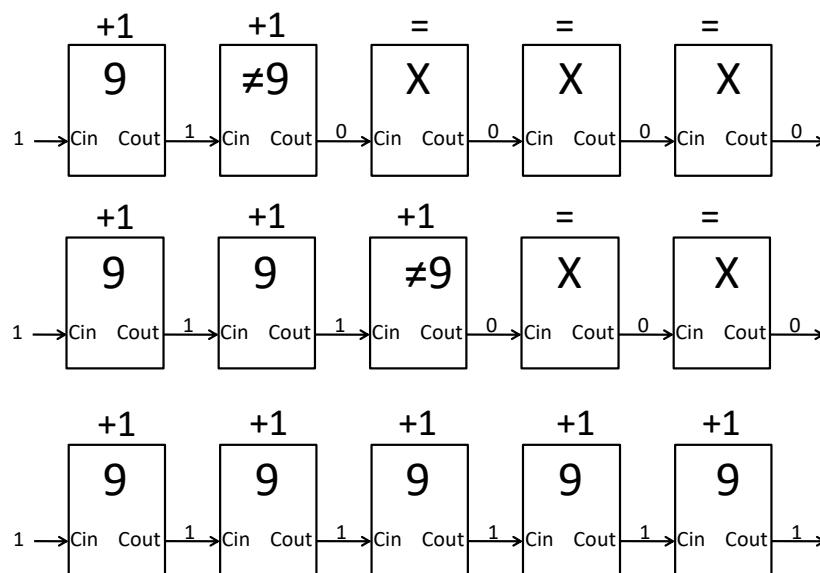


Figura 48.- Propagación del acarreo en un contador decimal

La entrada de acarreo del módulo correspondiente al bit de menor peso puede conectarse a una entrada que realizaría las veces de entrada de acarreo (de habilitación) del contador completo; en este caso la salida de acarreo del módulo correspondiente al dígito de mayor peso es el acarreo de salida del contador completo (se pone a 1 cuando todos los dígitos decimales son 9, 1001b) –en el contador de la figura 47 sería una salida de fin de cuenta de un contador decimal sin habilitación.

Ejercicio

25.- Diseñe un contador decimal de 3 dígitos con entrada y salida de acarreo y entrada de reset síncrono.

PARTE III. Especificación funcional de contadores

La especificación funcional completa de un contador debe incluir las siguientes características:

- El código utilizado para expresar la cuenta: Aunque hasta ahora sólo se han visto ejemplos de contadores que expresan el estado de cuenta en binario natural o BCD, un contador puede utilizar otro tipo de códigos: código Gray, *One hot*, etcétera. En la descripción de un contador resulta necesario indicar cuál es el código que emplea.
- El sentido de cuenta: Hay que especificar si se trata de un contador ascendente, descendente o reversible.
- El módulo del contador: el módulo de cuenta es una información que no puede deducirse directamente a partir del número de bits de salida: un contador binario de 4 bits, por ejemplo, puede tener cualquier módulo entre 9 (de 0 a 8) y 16 (de 0 a 15). Por otra parte, hay contadores *raro*s que cuentan entre estados diferentes del menor y mayor expresables con el número de bits de salida: por ejemplo, un contador binario de 3 bits que cuente en sentido ascendente entre el estado de cuenta 2 y el 6, u otro que cuente en sentido descendente entre 5 y 3.
- Las entradas síncronas de que dispone, su efecto y nivel de actividad: Deben enumerarse las entradas síncronas (de habilitación, reset o inicialización y de control *up/down*) de que disponga el contador, su nivel de actividad y cualquier relación adicional entre ellas, como por ejemplo, en un contador con habilitación y reset síncrono, si el reset está o no supeditado a la habilitación.
- La disponibilidad o no de salida de fin de cuenta y su modo de funcionamiento: El modo de funcionamiento debe indicar, en contadores que dispongan de habilitación, la dependencia, o no, de la habilitación.
- El nivel activo y el estado de cuenta determinado por la entrada de control asíncrono: Debe indicarse a qué estado de cuenta resulta ser inicializado el contador por la activación de la entrada de reset asíncrono.

La especificación del funcionamiento de un contador puede realizarse mediante una descripción verbal de las características señaladas, utilizando tablas de transiciones resumidas o, si se trata de un contador con una funcionalidad relativamente simple y pocos estados de cuenta, mediante un diagrama de estados. Cualquiera de estos tipos de descripciones funcionales se acompaña, en ocasiones y con la finalidad de facilitar la comprensión del funcionamiento del contador, de cronogramas que describen detalles de su funcionamiento.

Especificación verbal

Cuando se especifica verbalmente la funcionalidad de un contador hay algunas convenciones que deben tenerse en cuenta:

- Si el contador es binario y no se indica expresamente el módulo, se supone que es máximo: 2^N , siendo N el número de bits del contador. Por ejemplo, si se dice de un contador que es binario y de 5 bits, y no se indica su módulo, se supone que éste vale 32.

- Si el código de cuenta es BCD, el contador puede denominarse, indistintamente, contador BCD o decimal y el número de bits del contador puede expresarse en términos de dígitos BCD (cada dígito BCD son 4 bits que codifican una cifra decimal). Por ejemplo, un contador decimal de dos dígitos tiene 8 bits de salida, 4 de los bits codifican las unidades y 4 las decenas. Si no se especifica un módulo para este tipo de contadores, se supone también que es máximo: en un contador decimal de dos dígitos, por ejemplo, ha de suponerse que tiene módulo 100 (cuenta desde 0 a 99).
- Si no se indica explícitamente el sentido de cuenta se supone que es ascendente.
- Si se indica que un contador tiene acarreo de entrada y de salida, ha de interpretarse que el acarreo de entrada es la habilitación de cuenta y, el de salida, una salida de fin de cuenta dependiente de la habilitación; cuando se dice que el contador tiene entrada de habilitación y salida de fin de cuenta debe aclararse si esta última es dependiente o independiente de la habilitación.
- Las entradas de inicialización síncrona, en contadores con habilitación de reloj, pueden actuar de manera dependiente o independiente de la habilitación, la especificación debe aclararlo, pero es frecuente que se intente expresar que la inicialización es independiente de la habilitación denominando a ésta última entrada, entrada de *Count Enable* (de habilitación de cuenta), mientras que el nombre de entrada de habilitación (o habilitación de reloj) suele reservarse para cuando no hay entradas de inicialización síncronas, o cuando, habiéndolas, su actuación requiere de la activación de la habilitación.

Ejemplos:

- *Contador binario de 16 bits con entrada de reset síncrono, activa a nivel alto, y entrada de reset asíncrono activa a nivel bajo.*

Cabe suponer que es un contador ascendente, con módulo 2^{16} , y que tanto las entradas de inicialización síncrona como asíncrona llevan al contador al estado de cuenta 0.

- *Contador decimal descendente de 4 dígitos con entrada de habilitación, activa a nivel alto, entrada de inicialización asíncrona, activa a nivel bajo, que inicializa el contador en 9999, y salida de fin de cuenta independiente de la habilitación.*

Es un contador con 16 bits de salida (4 dígitos BCD), de módulo 10000 (cuenta, en sentido descendente, desde 9999 hasta 0).

- *Contador binario módulo 6 con entrada y salida de acarreo.*

Es un contador ascendente con salida de fin de cuenta (acarreo de salida) dependiente de la habilitación (acarreo de entrada).

- *Contador binario reversible de 4 bits, módulo 7, con entrada de habilitación, activa a nivel alto, reset síncrono, activo a nivel bajo, y supeditado a la habilitación y entrada de reset asíncrona*

Tablas de transición, diagramas de estado y cronogramas

Como la especificación verbal del funcionamiento puede resultar ambigua, debido a las convenciones y a la casuística que se puede dar en las entradas y salidas del contador, es habitual acompañar dicha descripción de tablas de transición, en las que se pueden tomar ciertas *licencias* formales para expresar más claramente el funcionamiento del circuito. Si el contador tiene pocos estados de cuenta, también se pueden usar

cronogramas y diagramas de estado. En la figura 49 se muestran tablas que especifican el funcionamiento de un contador reversible.

nRST	ENA	RST	UP/DOWN	Q[15..0]		ENA	UP/DOWN	Q[15..0]	FDC
0	X	X	X	Todos 0		1	0	Todos 1	1
1	0	X	X	Qt		1	1	Todos 0	1
1	1	0	0	Qt +1				RESTO	0
1	1	0	1	Qt -1					
1	1	1	X	Todos 0					

Figura 49.- Tablas para la descripción del funcionamiento de un contador reversible

De la tabla de la izquierda puede deducirse que el contador tiene 16 bits, es reversible, y su sentido de cuenta lo controla la entrada UP/DOWN; que tiene una entrada de habilitación, activa a nivel alto, y una entrada de reset síncrono, también activa a nivel alto y dependiente de la habilitación, y que dispone además de una entrada de reset asíncrono activo a nivel bajo. La tabla de la derecha describe el funcionamiento de la salida de fin de cuenta: es dependiente de la habilitación y se activa en el último estado de cuenta (ascendente o descendente).

En la figura 50 se muestra el diagrama de estados de un contador decimal con entrada de habilitación y salida de fin de cuenta –en el diagrama se omite, por simplicidad, la representación de la salida de cuenta.

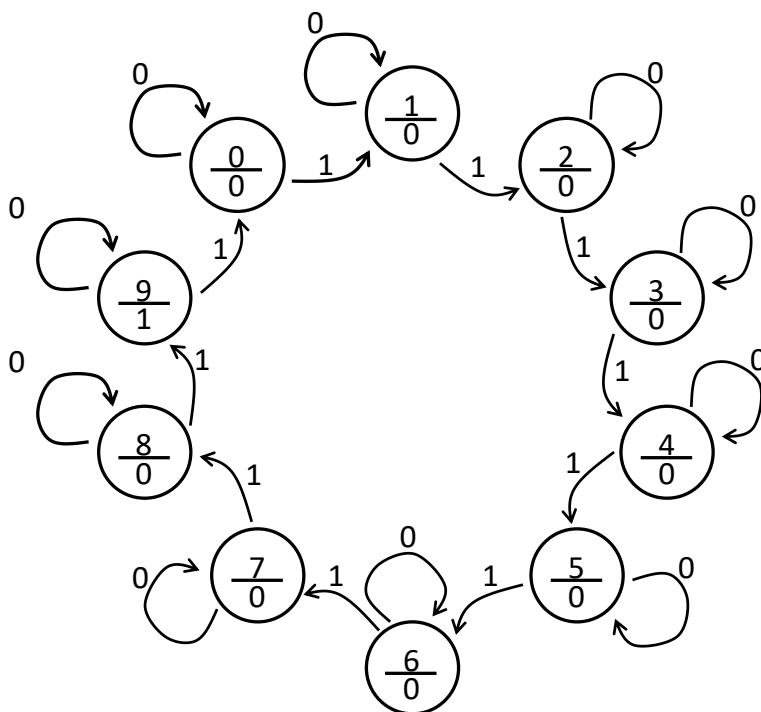


Figura 50.- Diagrama de estados de un contador

Los diagramas de estado sólo pueden emplearse para ilustrar el funcionamiento de contadores con un módulo pequeño. Incluso en estos casos, el diagrama puede tener cierta complejidad y resulta más difícil extraer la información sobre el funcionamiento del circuito que de las tablas de transiciones. Lo mismo ocurre con los cronogramas y, por ello, su uso como herramienta de especificación es menos frecuente.

Esta página se ha dejado intencionadamente en blanco

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Ejercicios sobre contadores

CÓDIGO:
BT3_ATGP5A

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Ejercicios. Aprendizaje colaborativo.	TIPO:	Presencial	DURACIÓN:	70 minutos
------------	---------------------------------------	-------	------------	-----------	------------

CALENDARIO:	Sexta Sesión	REQUISITOS:	Contadores.
-------------	--------------	-------------	-------------

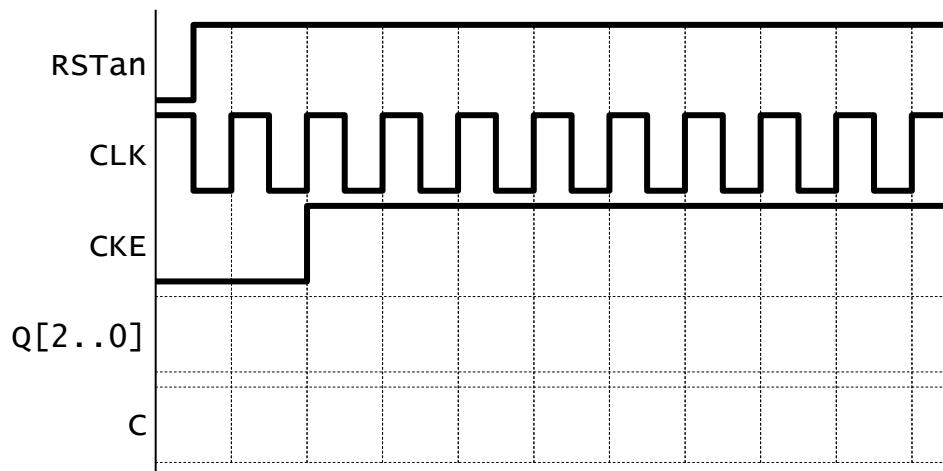
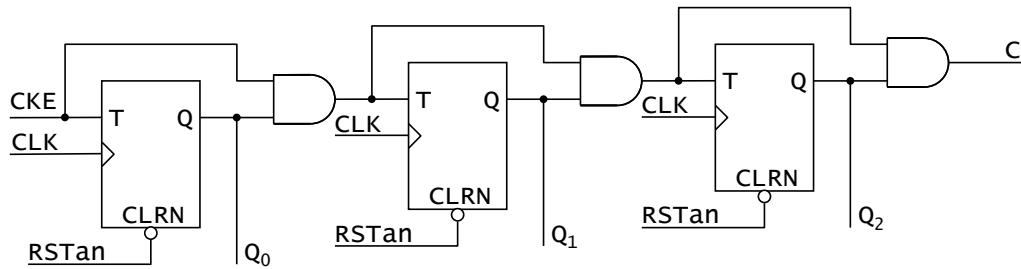
CRITERIO DE ÉXITO:	
--------------------	--

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTEOVALUACIÓN: [entre 0 y 10 puntos]	En el cuestionario de moodle
------------------	---------	--	------------------------------

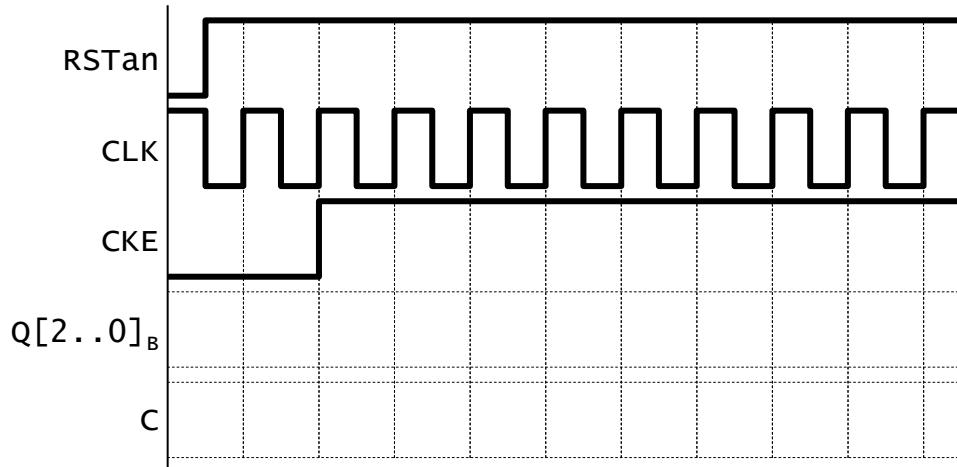
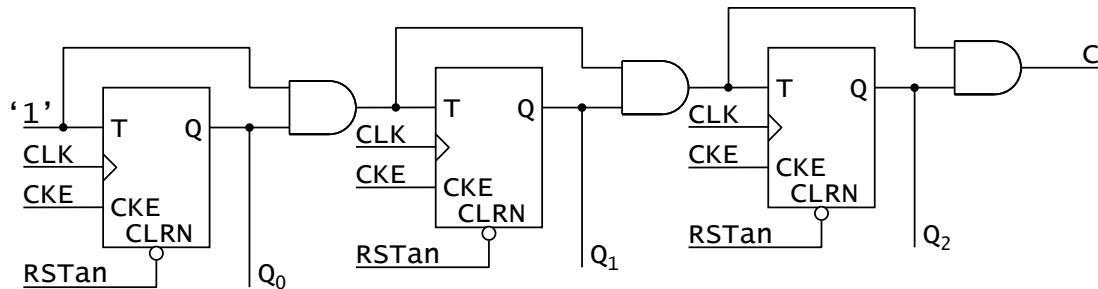
Actividad BT3_ATGP5A: Ejercicios sobre contadores.

Ejercicio 1: Complete el cronograma correspondiente al circuito de la figura y realice una descripción textual de su funcionalidad especificando el tipo de circuito y el comportamiento de sus entradas y salidas.



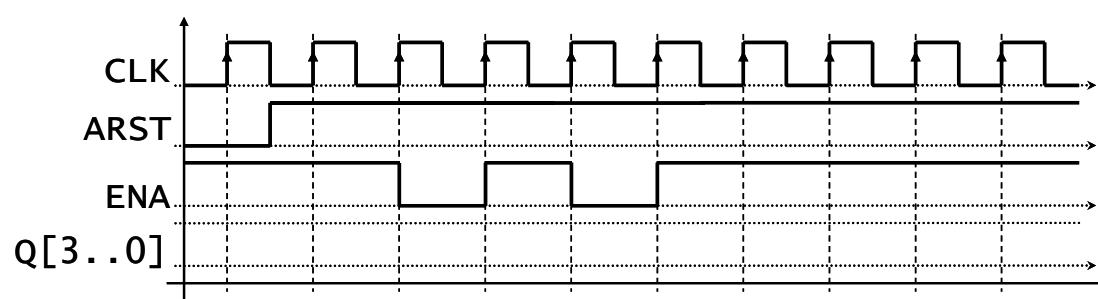
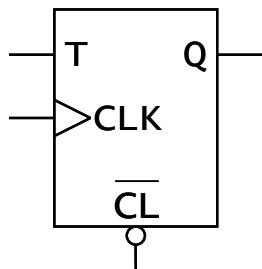
Actividad BT3_ATGP5A: Ejercicios sobre contadores.

Ejercicio 2: Complete el cronograma correspondiente al circuito de la figura y realice una descripción textual de su funcionalidad especificando el tipo de circuito y el comportamiento de sus entradas y salidas.



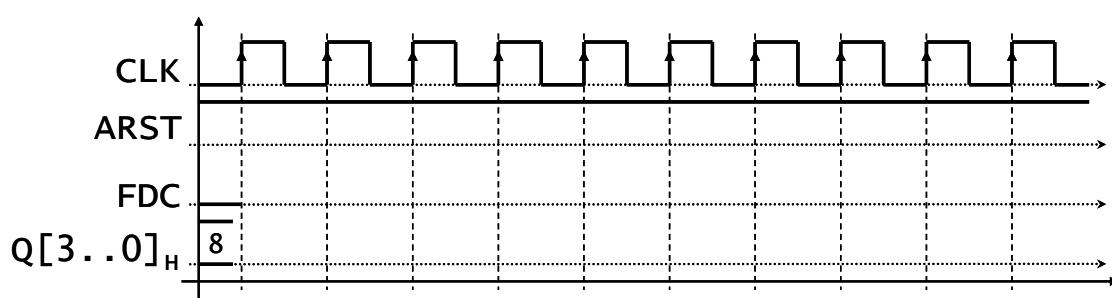
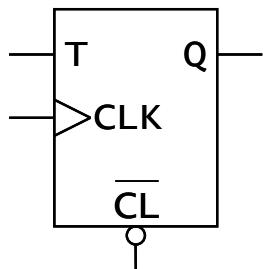
Actividad BT3_ATGP5A: Ejercicios sobre contadores.

Ejercicio 3: Utilizando flip-flops T como el de la figura, diseñe un contador binario ascendente de 4 bits con entrada de habilitación (ENA) activa a nivel alto y una entrada de reset asíncrono (aRST) activa a nivel bajo. Complete el cronograma de funcionamiento del contador.



Actividad BT3_ATGP5A: Ejercicios sobre contadores.

Ejercicio 4: Utilizando flip-flops T como el de la figura, diseñe un contador binario ascendente de 4 bits con una entrada de reset asíncrono (aRST) activa a nivel bajo y salida de fin de cuenta (FDC) activa a nivel alto. Complete el cronograma de funcionamiento del contador.

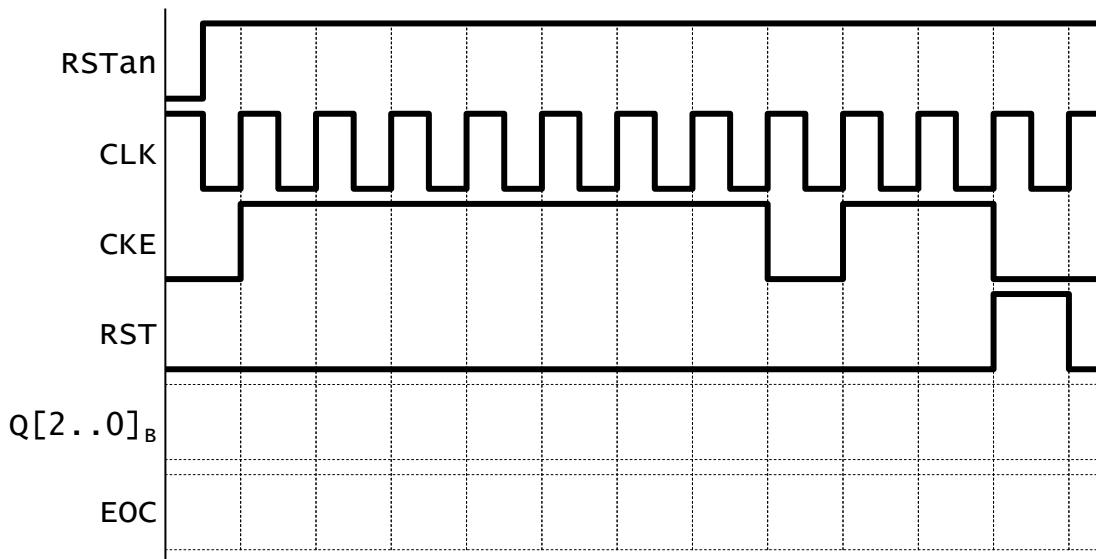


Actividad BT3_ATGP5A: Ejercicios sobre contadores.

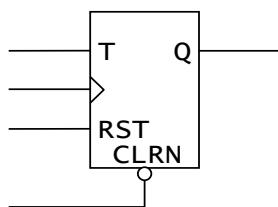
Ejercicio 5: Sea un contador de módulo máximo que dispone, entre otras entradas, de una entrada de inicialización asíncrona RSTan. El funcionamiento de este contador está determinado por las dos tablas dadas a continuación. Responda a los apartados A y B.

RST_{an}	RST	CKE	$Q[2..0]$	CKE	$Q[2..0]$	EOC
0	X	X	000	1	111	1
1	1	X	000	Resto		0
1	0	0	Q_t			
1	0	1	Q_t+1			

- A. Complete el cronograma de la figura correspondiente al contador citado.



- B. Utilizando *flip-flops* tipo T con reset síncrono RST y asíncrono CLRN como el de la figura y la lógica adicional necesaria realice el diseño del contador.



BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Ejercicios sobre contadores

CÓDIGO:
BT3_ATGP5B

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Ejercicios. Aprendizaje colaborativo.	TIPO:	Presencial	DURACIÓN:	70 minutos
------------	---------------------------------------	-------	------------	-----------	------------

CALENDARIO:	Séptima Sesión	REQUISITOS:	Contadores.
-------------	----------------	-------------	-------------

CRITERIO DE ÉXITO:	
--------------------	--

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTEOVALUACIÓN: [entre 0 y 10 puntos]	En el cuestionario de moodle
------------------	---------	--	------------------------------

Actividad BT3_ATGP5B: Ejercicios sobre contadores.

Introducción.

Esta actividad se realizará en la séptima sesión presencial, en grupos G4. Se realizará un total de 4 ejercicios.

Para el ejercicio 1:

- El apartado A lo realizarán todos los miembros del grupo de forma individual. Al terminar el apartado el profesor expondrá la solución (10 minutos máx.).
- Cada miembro del grupo realizará uno de los apartados: B, C, D y E. En el enunciado se indica qué ejercicio corresponde a cada miembro del grupo (4 minutos máx.).
- Cada miembro del grupo explicará a los demás el ejercicio que ha realizado y la solución que propone (4 minutos máx.).
- El profesor explicará la solución de los apartados B, C, D y E (2 minutos máx.).

Para el ejercicio 2:

- El ejercicio será realizado por todos los miembros del grupo de forma individual. Al terminar el ejercicio el profesor expondrá la solución (10 minutos máx.).

Para el ejercicio 3:

- Cada miembro del grupo realizará uno de los apartados: A, B, C y D. En el enunciado se indica qué ejercicio corresponde a cada miembro del grupo (10 minutos máx.).
- Cada miembro del grupo explicará a los demás el ejercicio que ha realizado y la solución que propone (4 minutos máx.).
- El profesor explicará la solución de los cuatro apartados (2 minutos máx.).

Para el ejercicio 4:

- El apartado A será realizado por una subgrupo y el apartado B por el otra. Al terminar los dos apartados el profesor expondrá la solución (8 minutos máx.).
- Cada miembro del grupo realizará uno de los apartados: C, D, E y F. En el enunciado se indica qué ejercicio corresponde a cada miembro del grupo (8 minutos máx.).
- Cada miembro del grupo explicará a los demás el ejercicio que ha realizado y la solución que propone (4 minutos máx.).
- El profesor explicará la solución de los apartados C, D, E y F (4 minutos máx.).

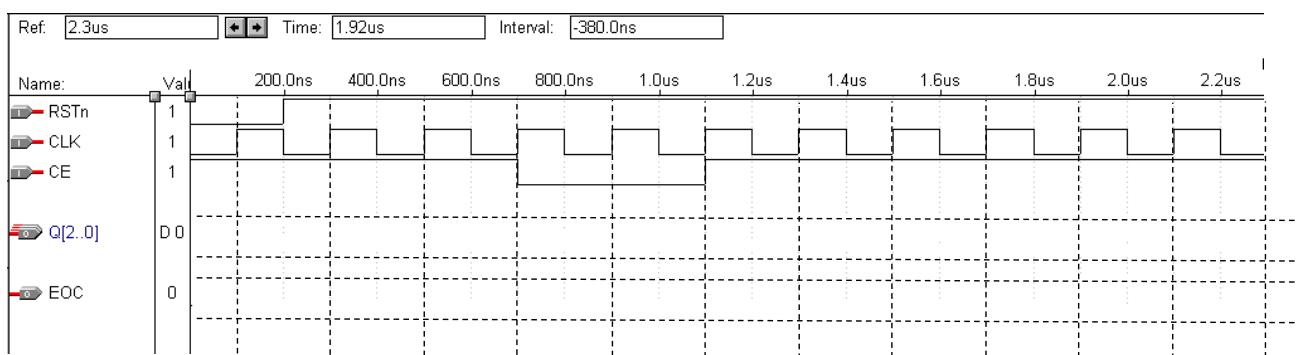
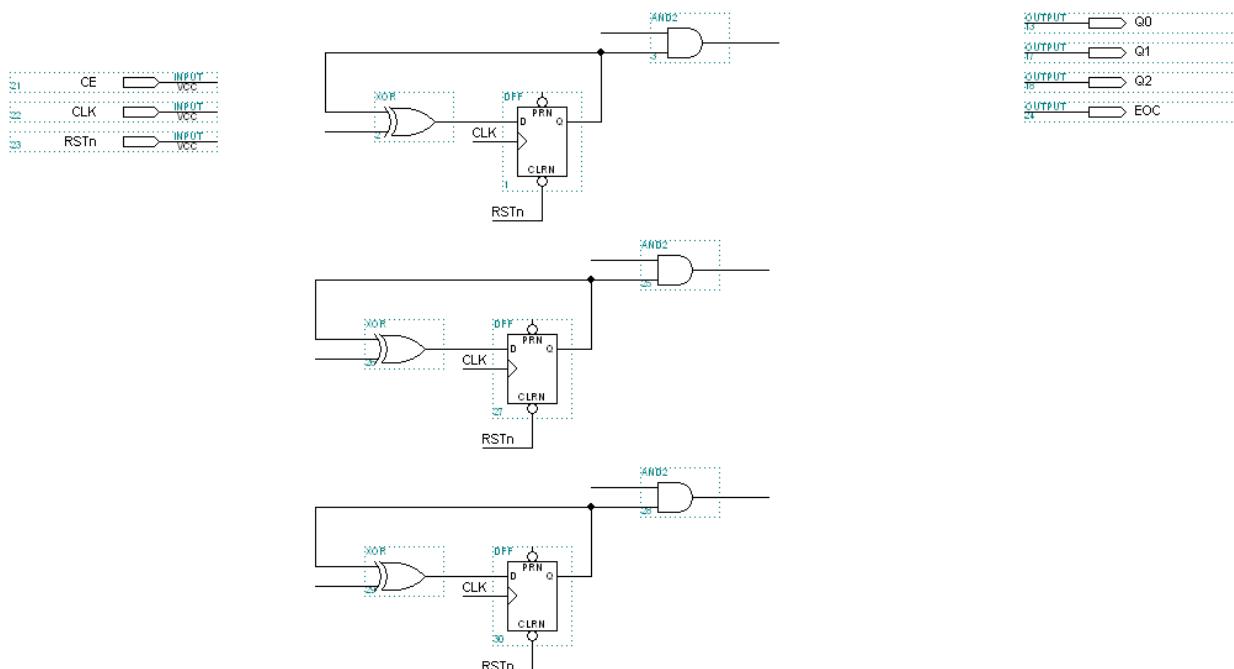
Para el ejercicio 5:

- Las mesas se colocarán en configuración de examen.
- Cada miembro del grupo realizará el ejercicio de forma individual (8 minutos máx.).
- El profesor recogerá un ejercicio de cada grupo G4 al azar.
- El profesor explicará la solución del ejercicio (2 minutos máx.).

Actividad BT3_ATGP5B: Ejercicios sobre contadores.

Ejercicio 1: Diseño de contadores a partir de estructuras regulares.

- A. Contador ascendente con entrada de habilitación (individual).** Complete el circuito de la figura para obtener un contador ascendente de 3 bits con entrada de habilitación de cuenta (CE) y acarreo de salida (EOC), ambas activas a nivel alto. Complete el cronograma de la figura relativo al mismo circuito.



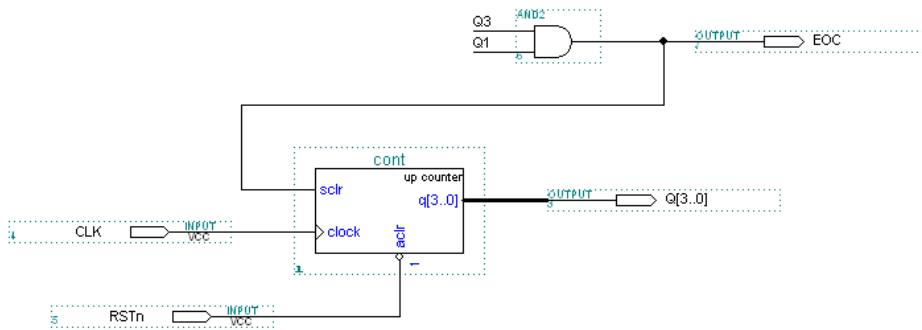
Actividad BT3_ATGP5B: Ejercicios sobre contadores.

- B. **Variante con entrada de reset síncrona (G4_A).** Diseñe una variante del contador del apartado A que tenga una entrada de reset síncrono activa a nivel bajo SRSTn. La entrada de reset síncrono no debe requerir la activación de la entrada de habilitación para su funcionamiento.
- C. **Variante con entrada de preset síncrona (G4_B).** Diseñe una variante del contador del apartado A que tenga una entrada de preset síncrono activa a nivel alto SSET. La entrada de preset síncrono no debe requerir la activación de la entrada de habilitación para su funcionamiento.
- D. **Variante con entrada de inicialización síncrona (G4_C).** Diseñe una variante del contador del apartado A que tenga una entrada de inicialización síncrona activa a nivel alto SSET que inicialice el contador a b"101". Esta entrada no debe requerir la activación de la entrada de habilitación para su funcionamiento.
- E. **Variante reversible (G4_D).** Diseñe una variante del contador del apartado A que tenga una entrada de sentido de la cuenta U_Dn, de modo que cuando $U_{Dn} = 1$ la cuenta es ascendente y descendente en caso contrario.

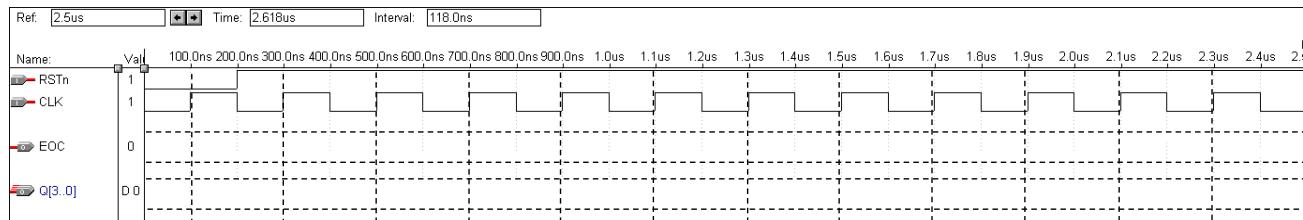
Diseñe aquí la variante que le corresponda:

Actividad BT3_ATGP5B: Ejercicios sobre contadores.

Ejercicio 2: Contadores de módulo arbitrario. Análisis de cronogramas (individual).
El circuito de la figura está basado en un contador binario ascendente de 4 bits:



A. Complete el siguiente cronograma, correspondiente al circuito de la figura.



B. Responda a las siguientes preguntas sobre el circuito:

- ¿cuál es el módulo del contador así diseñado?
- ¿cuál es el período de la señal EOC?
- ¿cuál es el cociente entre el período de la señal EOC y el período del reloj del sistema?

C. Con relación al contador binario “cont”, indique:

- ¿Cuál es su módulo?
- ¿Cuántos flip-flops tiene?

Actividad BT3_ATGP5B: Ejercicios sobre contadores.

Ejercicio 3: Diseño de contadores de módulo arbitrario.

Utilizando un contador binario ascendente con reset síncrono activo a nivel alto del número de bits que precise, diseñe:

- A. Un contador ascendente módulo 27 (G4_A).
- B. Un contador ascendente módulo 40 (G4_B).
- C. Un contador ascendente módulo 60 (G4_C).
- D. Un contador ascendente módulo 16 (G4_D).

Diseñe aquí el contador que le corresponda:

Actividad BT3_ATGP5B: Ejercicios sobre contadores.

Ejercicio 4: Ampliación de contadores BCD.

- A. A partir de un contador binario ascendente de 4 bits con entrada de habilitación de reloj (CE) y reset síncrono (SRST) diseñe un contador que cuente del 0 al 9 con salida de fin de cuenta (G4_A y G4_B).
- B. A partir de un contador binario descendente de 4 bits con entrada de habilitación de reloj (CE) y set síncrono a “9_H” (SSET) diseñe un contador que cuente del 9 al 0 con salida de fin de cuenta –en este caso el fin de cuenta es 0- (G4_C y G4_D).

Diseñe aquí el contador que le corresponda:

Utilice los contadores diseñados como bloques funcionales para realizar los siguientes diseños:

- C. Diseñe un contador BCD ascendente de 2 dígitos (G4_A)
- D. Diseñe un contador BCD ascendente de 3 dígitos (G4_B)
- E. Diseñe un contador BCD descendente de 2 dígitos (G4_C)
- F. Diseñe un contador BCD descendente de 3 dígitos (G4_D)

Diseñe aquí el contador que le corresponda:

Actividad BT3_ATGP5B: Ejercicios sobre contadores.

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Test final sobre ejercicios de contadores.

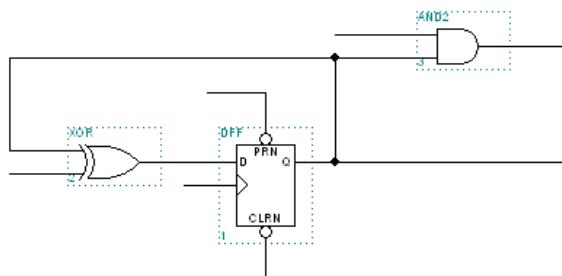
CÓDIGO:
BT3_ATGP5B

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

GRUPO		CALIFICACIÓN: [entre 0 y 10 puntos]	puntos
-------	--	--	--------

Ejercicio 5: Utilizando la estructura de la figura como punto de partida, modificándola si es necesario, diseñe un contador binario de 3 bits con *reset* asíncrono activo a nivel bajo *RSTn*, que cuente de forma ascendente y cíclica del 1 al 6.



Esta página se ha dejado intencionadamente en blanco

BLOQUE TEMÁTICO3

TÍTULO DE LA ACTIVIDAD:
Ejercicios sobre contadores.

CÓDIGO:
BT3_ATGNP3

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Ejercicios. Aprendizaje colaborativo.	TIPO:	No presencial	DURACIÓN:	80 minutos
------------	---------------------------------------	-------	---------------	-----------	------------

CALENDARIO:	Séptima Sesión. Antes de la Octava Sesión	REQUISITOS:	
-------------	--	-------------	--

CRITERIO DE ÉXITO:	Realizar correctamente tres de los cuatro primeros ejercicios más el 5 y el 6.
--------------------	--

COMENTARIOS E INCIDENCIAS:

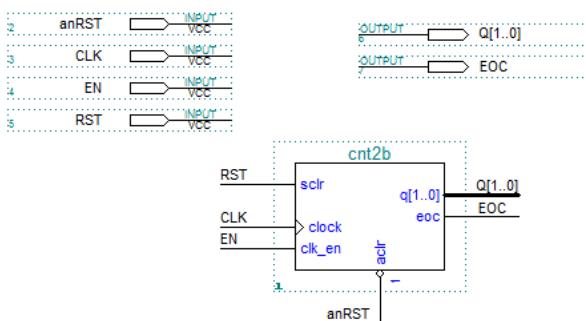
TIEMPO DEDICADO:	minutos	AUTEOVALUACIÓN: [entre 0 y 10 puntos]	puntos
------------------	---------	--	--------

En esta actividad deberá realizar en grupo los seis ejercicios sobre contadores que encontrará a continuación. Aunque puede elegir junto con sus compañeros el método de trabajo que considere oportuno, el método que se recomienda es que cada miembro del grupo trate por separado de resolver los ejercicios y después se realice una puesta en común, comparando las soluciones propuestas entre sí y con las proporcionadas al final de este documento y tratando de resolver en el seno del grupo cualquier duda que pueda existir. En caso de que haya alguna duda que no se pueda resolver se recomienda concertar una tutoría en grupo con su profesor.

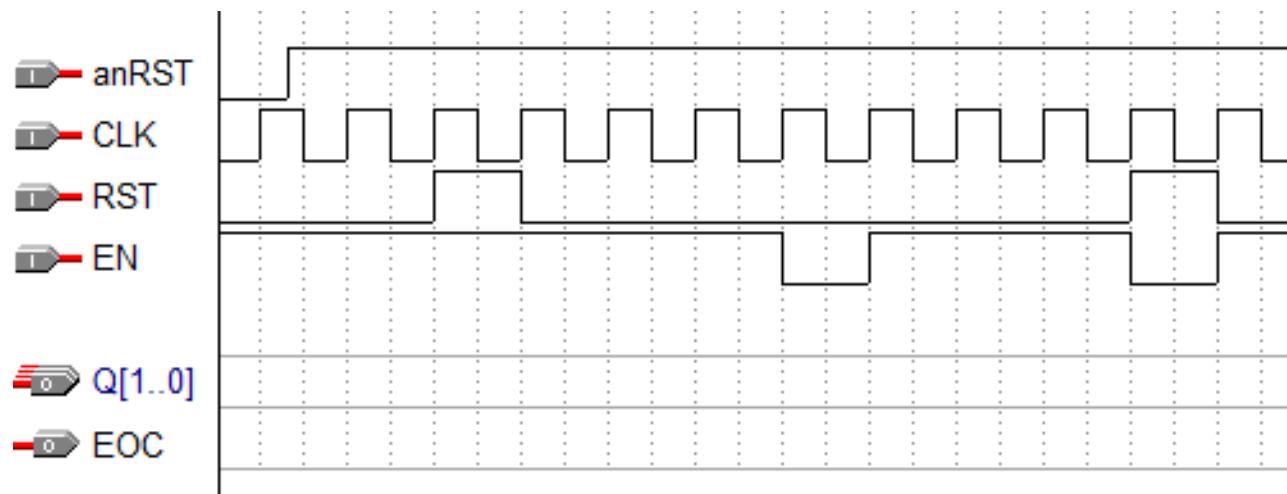
Ejercicio 1

Considere el circuito de la figura, en el que el contador *cnt2b* se comporta de acuerdo a la tabla adjunta:

aclr	clock	clk_en	sclr	q[1..0]	eoc
0	X	X	X	00	0
1	↑	0	X	q(t-1)	1 si q=11, 0 si no
1	↑	1	0	q(t-1)+1	
1	↑	1	1	00	0



Rellene el siguiente cronograma:

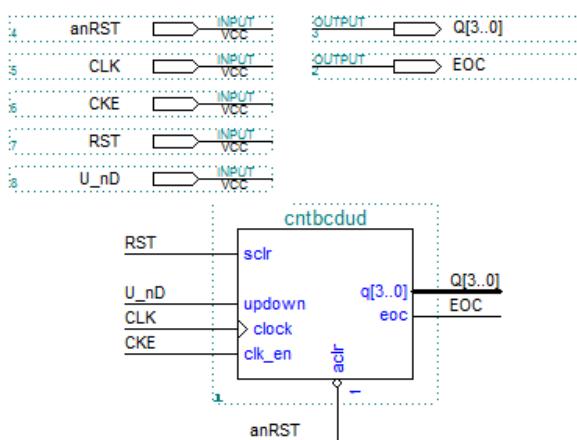


Ejercicio 2

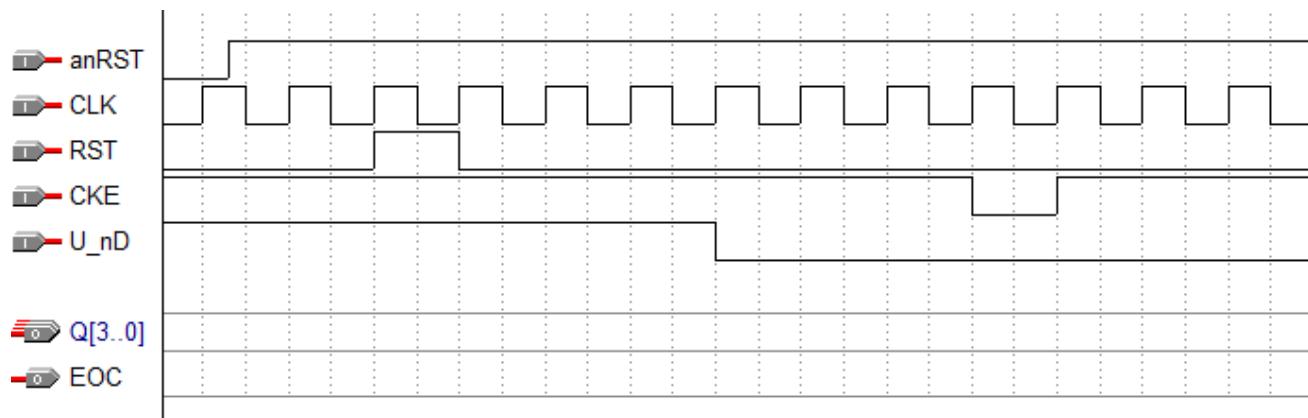
Considere el circuito de la figura, en el que el contador *cntbcdud* es un contador BCD reversible cuyo funcionamiento se describe en las tablas adjuntas:

aclr	clock	clk_en	sclr	updown	q[3..0]
0	X	X	X	X	0000
1	↑	0	X	X	q(t-1)
1	↑	1	1	X	0000
1	↑	1	0	1	q(t-1)+1
1	↑	1	0	0	q(t-1)-1

updown	q[3..0]	eoc
0	0000	1
1	1001	1
resto de combinaciones		0



Rellene el siguiente cronograma:

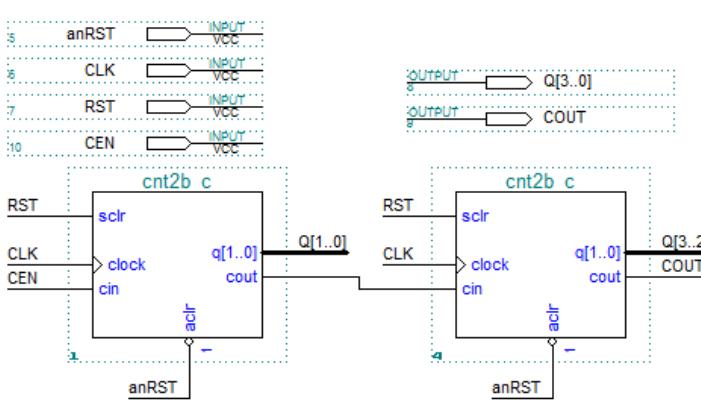


Ejercicio 3

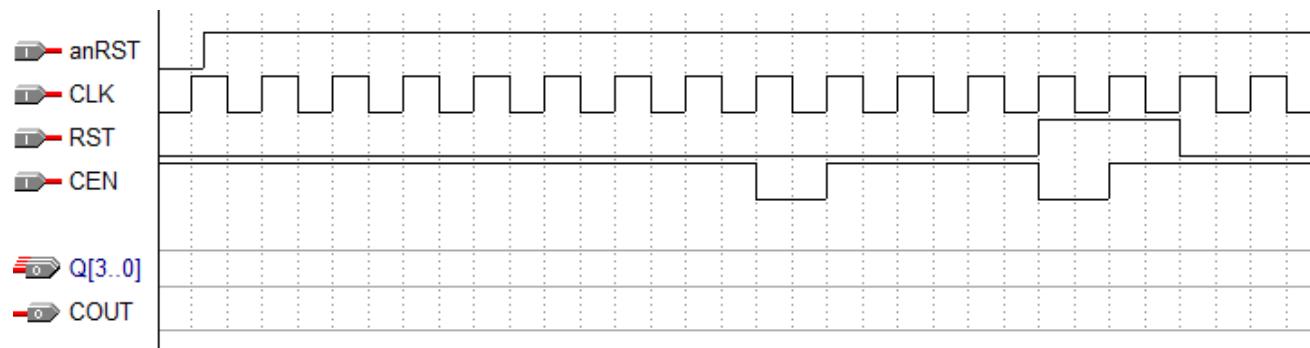
Considere el siguiente circuito, que consiste en un contador construido a partir de dos contadores binarios ascendentes con módulo 3 con entrada y salida de acarreo de cuenta, cuyo funcionamiento se especifica en las tablas adjuntas:

aclr	clock	sclr	cin	q[1..0]
0	X	X	X	00
1	↑	1	X	00
1	↑	0	0	q(t-1)
1	↑	0	1	q(t-1)+1

cin	q[1..0]	cout
1	10	1
resto de combinaciones		0



3.1.- Rellene el siguiente cronograma:

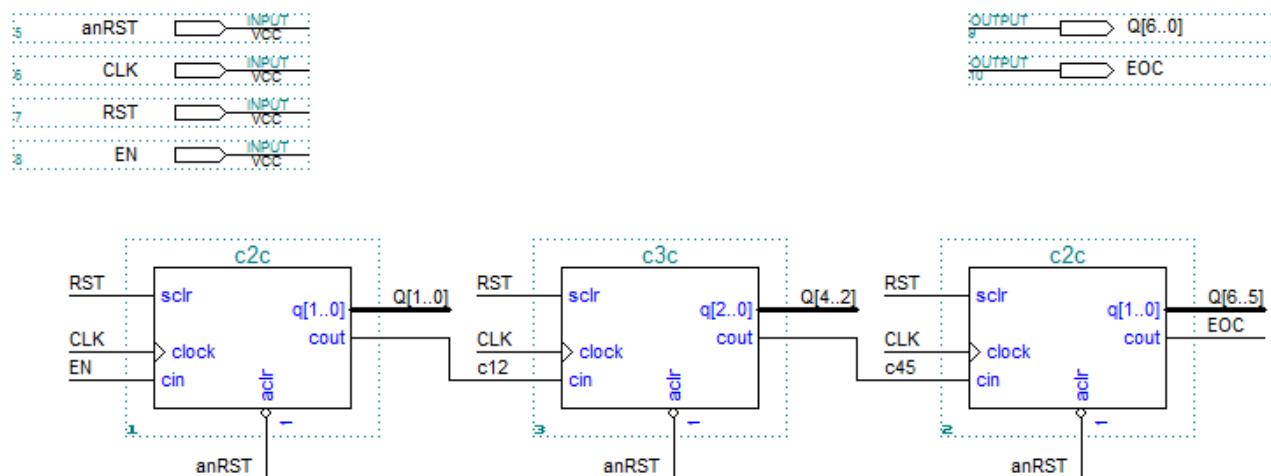


3.2.- ¿Cuál es el módulo del contador formado por los dos contadores *cnt2b_c*?

3.3.- El contador resultante, ¿es un contador binario? ¿Por qué?

Ejercicio 4

Considere el circuito que se muestra a continuación, en el que se construye un contador a partir de contadores de módulo más pequeño:

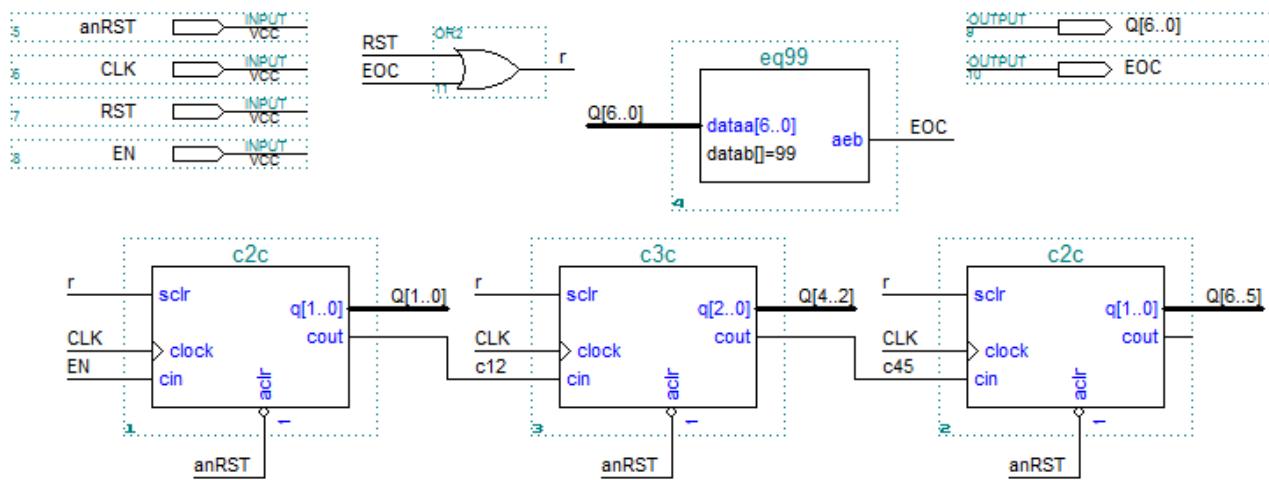


Los contadores *c2c* y *c3c* son contadores binarios ascendentes de módulo máximo de 2 y 3 bits, respectivamente, con entradas de entrada y salida de acarreo de cuenta. En ambos casos la salida *cout* se activa cuando el contador ha llegado a su último estado de cuenta y la entrada *cin* está activa. Los contadores tienen además una entrada de puesta a cero síncrona, *sclr*, y una de puesta a cero asíncrona, *aclr*. Las entradas síncronas son activas a nivel alto, mientras que la entrada asíncrona lo es a nivel bajo.

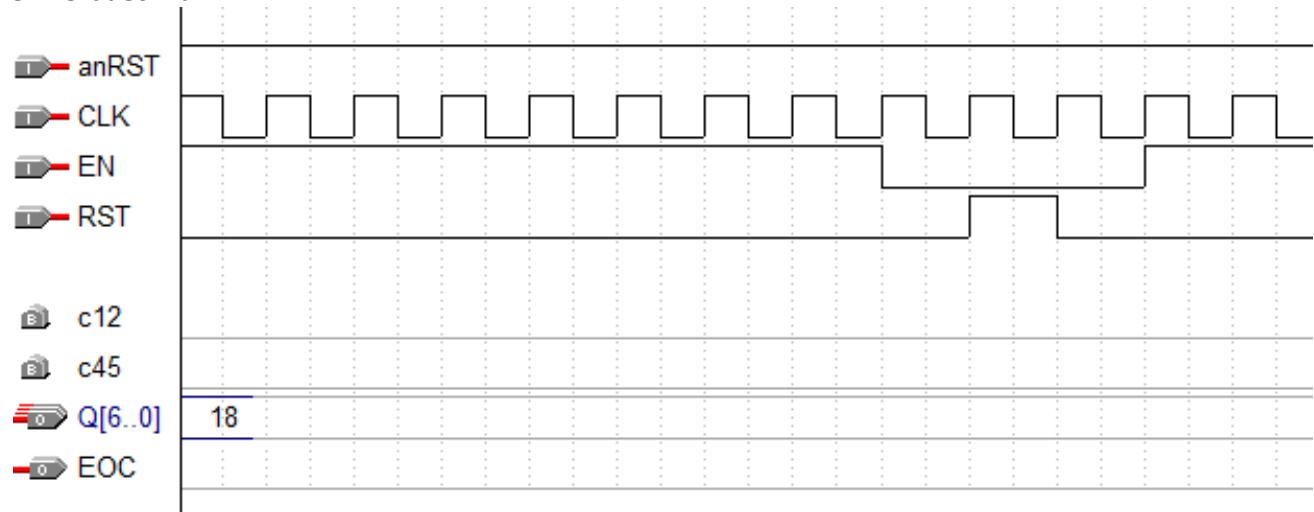
4.1.- ¿Cuál es el módulo del contador resultante?

4.2.- El contador resultante, ¿es un contador binario? ¿Por qué?

El circuito se modifica para alterar su módulo de cuenta, quedando como se indica a continuación:



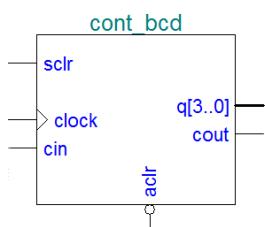
El módulo *eq99* es un comparador¹ que activa su salida *aeb* (a nivel alto) si su entrada *dataa* es igual a 99.

4.3.- ¿Cuál es ahora el módulo del contador?**4.4.- Rellene el siguiente cronograma para el circuito modificado, en el que el valor de la salida *Q* se indica en hexadecimal:****4.5.- ¿Existe alguna diferencia en la funcionalidad de la salida *EOC* en las dos versiones del contador?**

¹ En realidad el circuito *eq99* es un decodificador, pero en este caso es más apropiado para entender su función en el circuito verlo como un comparador con una constante.

Ejercicio 5

Se dispone de contadores BCD de un dígito cuyo símbolo y tablas de funcionamiento se muestran a continuación:



aclr	clock	sclr	cin	q[3..0]
0	X	X	X	0000
1	↑	1	X	0000
1	↑	0	0	q(t-1)
1	↑	0	1	q(t-1)+1

cin	q[3..0]	cout
1	1001	1
resto de combinaciones		0

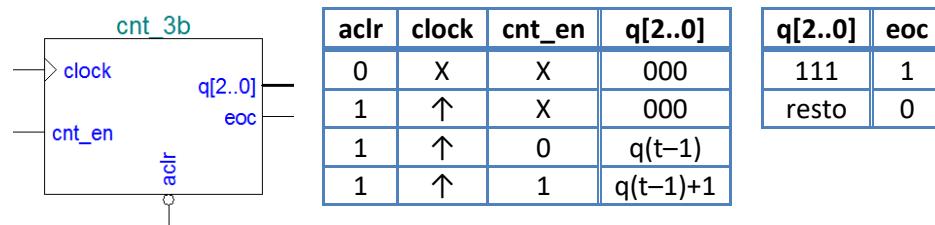
Diseñe, utilizando contadores como el descrito y la lógica adicional que necesite, un contador BCD módulo 50 con la siguiente interfaz:

- Entrada asíncrona de puesta a cero, *anRST*, activa a nivel bajo.
- Entrada de habilitación de cuenta, *CKE*, activa a nivel alto.
- Entrada síncrona de puesta a cero, *RST*, activa a nivel alto. Esta entrada necesita que *CKE* esté activa para actuar.
- Salida de cuenta, *Q* (con los bits que sean necesarios).
- Salida de fin de cuenta, *EOC*, activa a nivel alto. Esta salida debe activarse siempre que el contador esté en su último estado de cuenta, independientemente del estado de *CKE*.

Ejercicio 6

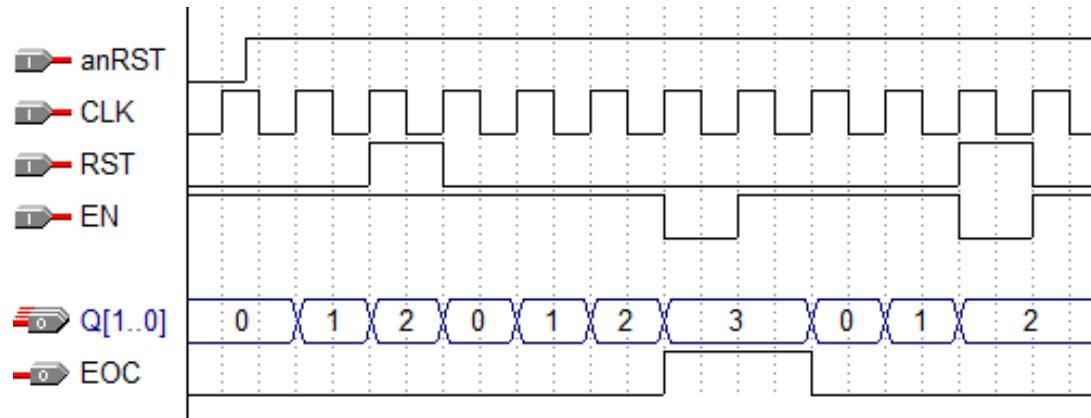
Diseñe un contador binario módulo 64 utilizando contadores binarios de tres bits como el que se muestra a continuación y la lógica adicional que necesite. El contador debe tener la siguiente interfaz:

- Entrada asíncrona de puesta a cero, *anRST*, activa a nivel bajo.
- Entrada de habilitación de cuenta, *CKE*, activa a nivel alto.
- Salida de cuenta, *Q* (con los bits que sean necesarios).
- Salida de fin de cuenta, *EOC*, activa a nivel alto. Esta salida debe activarse únicamente cuando el contador esté en su último estado de cuenta y la entrada *CKE* esté activa.

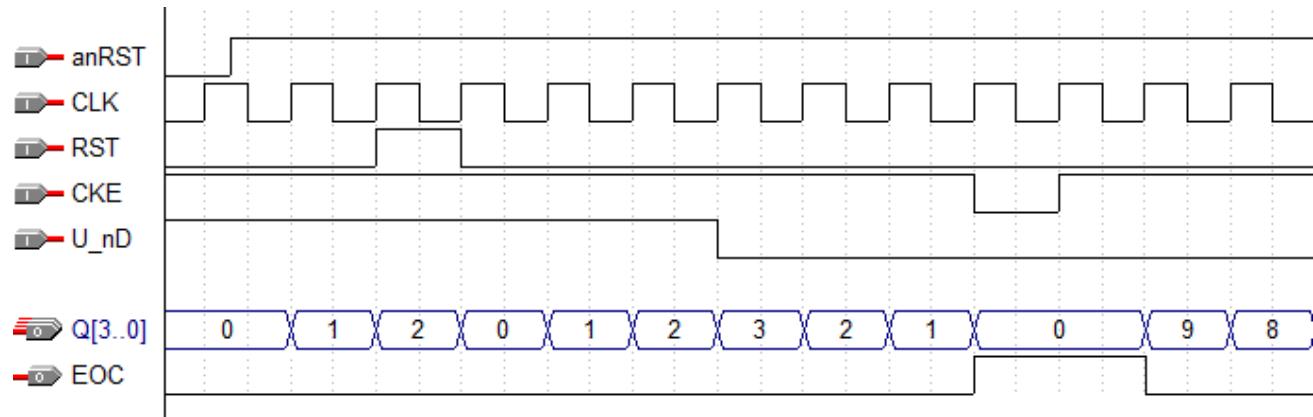


Soluciones

Ejercicio 1

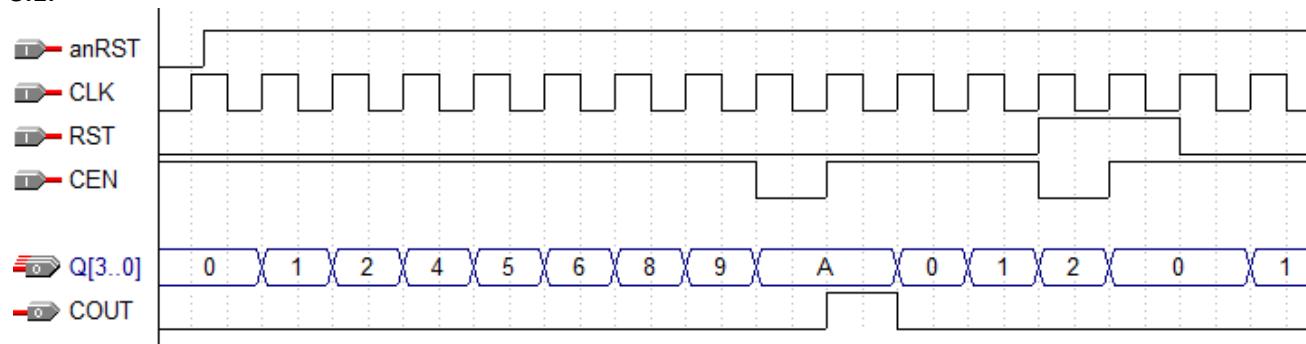


Ejercicio 2



Ejercicio 3

3.1.-



3.2.-

El módulo es $3 * 3 = 9$.

3.3.-

No es un contador binario, porque la secuencia de cuenta es 0000, 0001, 0010, 0100, 0101, 0110, 1000, 1001, 1010, 0000..., que no corresponde a una cuenta en binario natural.

Ejercicio 4

4.1.-

El módulo es $4 * 8 * 4 = 128$.

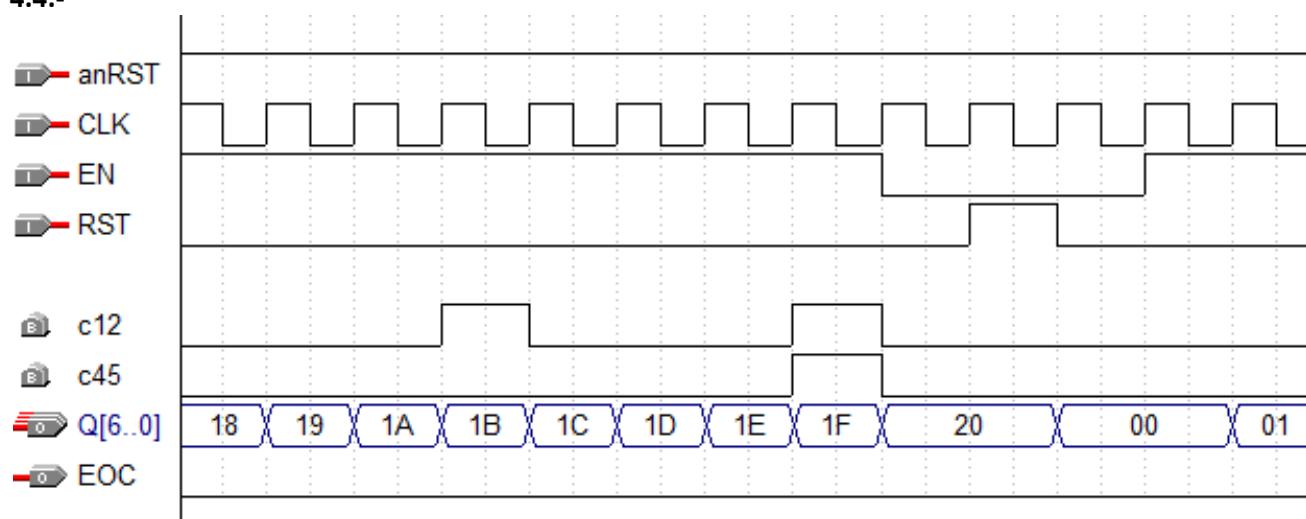
4.2.-

Sí, es un contador binario porque su secuencia de cuenta corresponde a la de una cuenta en binario natural.

4.3.-

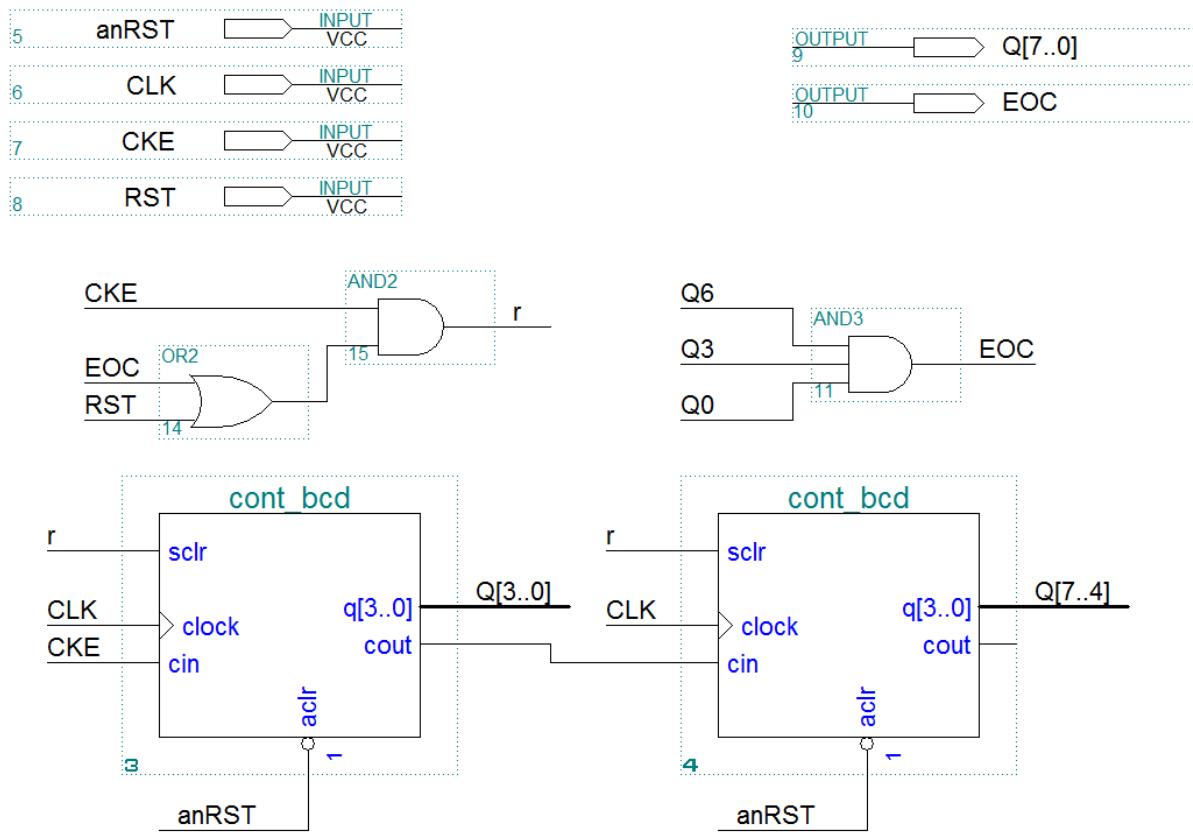
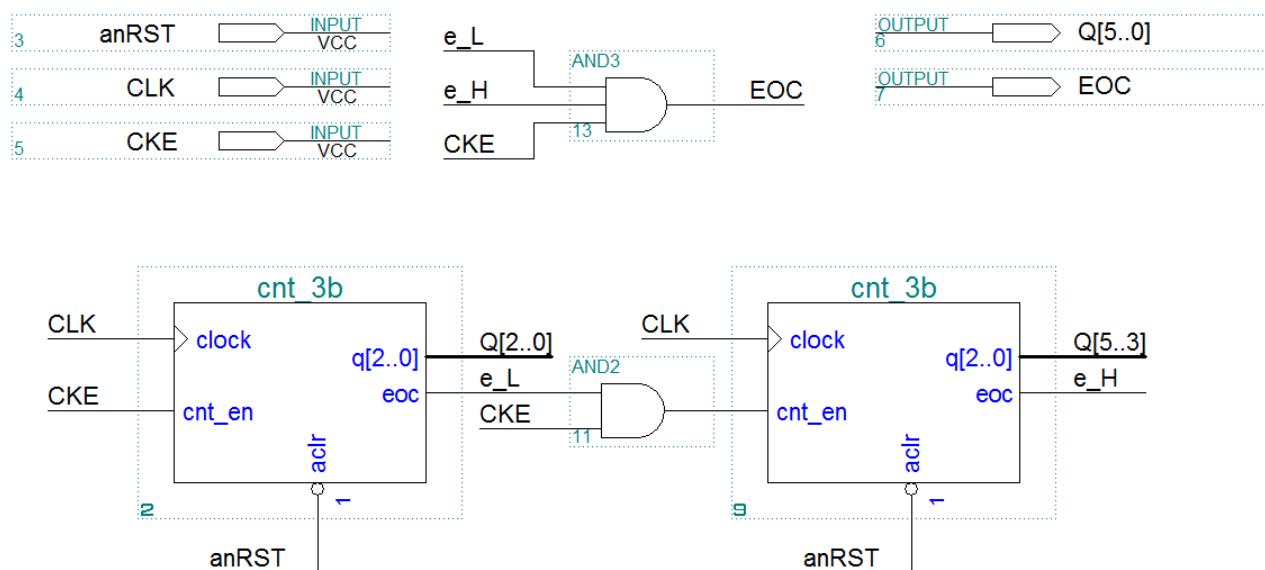
El módulo es ahora 100.

4.4.-



4.5.-

Sí, en el primer circuito la señal EOC se activa sólo si la entrada EN está activa, mientras que en la segunda versión se activa siempre que el contador esté en el último estado de cuenta, independientemente del valor de la entrada EN.

Ejercicio 5**Ejercicio 6**

Esta página se ha dejado intencionadamente en blanco

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Ejercicios sobre sistemas secuenciales.

CÓDIGO:
BT3_ATGP6

FECHA:	
--------	--

GRUPO:			
--------	--	--	--

MODALIDAD:	Ejercicios. Aprendizaje colaborativo.	TIPO:	Presencial	DURACIÓN:	100 minutos
------------	---------------------------------------	-------	------------	-----------	-------------

CALENDARIO:	Octava Sesión	REQUISITOS:	Sistemas secuenciales: flip-flops, registros. Arquitectura de los sistemas secuenciales.
-------------	---------------	-------------	--

CRITERIO DE ÉXITO:	Realizar correctamente cuatro ejercicios entre el 1 y el 5, más el ejercicio 6.
--------------------	---

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTEOVALUACIÓN: [entre 0 y 10 puntos]	En el cuestionario de moodle
------------------	---------	--	------------------------------

Introducción

Esta actividad se realizará en la novena sesión presencial, en grupos G4. Se realizarán en total 6 ejercicios.

Para los ejercicios 1 a 5:

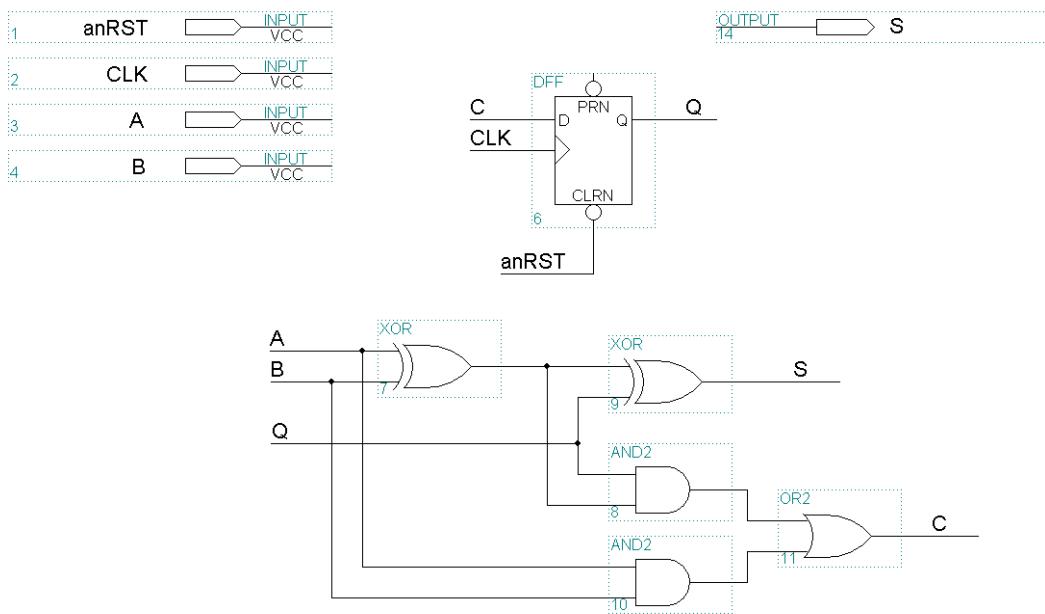
- Todos los miembros del grupo realizarán de manera individual el mismo ejercicio (10 minutos cada ejercicio, por término medio).
- Se realizará una puesta en común de las soluciones aportadas por cada miembro del grupo. Se constatarán las diferencias, si las hubiere, y se debatirá sobre las mismas (4 minutos como máximo).

Para el ejercicio 6:

- Las mesas se colocarán en configuración de examen.
- Cada miembro del grupo realizará el ejercicio de forma individual (25 minutos como máximo).
- El profesor recogerá un ejercicio de cada grupo G4 al azar.

Ejercicio 1

Complete los siguientes apartados para el circuito de la figura, que consiste en una máquina de estados de Mealy que suma dos números bit a bit, empezando por el dígito menos significativo:



- A. **Estructura.** Identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.
- B. **Salidas.** Obtenga la ecuación de la salida en función del estado actual y de las entradas. Complete la tabla de verdad que se adjunta.

$$S =$$

Estado	Código	Entradas		Salida
	Q	A	B	
X	0	0	0	
	0	0	1	
	0	1	0	
	0	1	1	
Y	1	0	0	
	1	0	1	
	1	1	0	
	1	1	1	

- C. Estado futuro.** Obtenga las ecuaciones del estado futuro en función del estado actual y de las entradas. Complete la tabla de verdad que se adjunta.

$C =$

Estado actual	Entradas		Estado futuro
Q_t	A	B	$C = Q_{t+1}$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

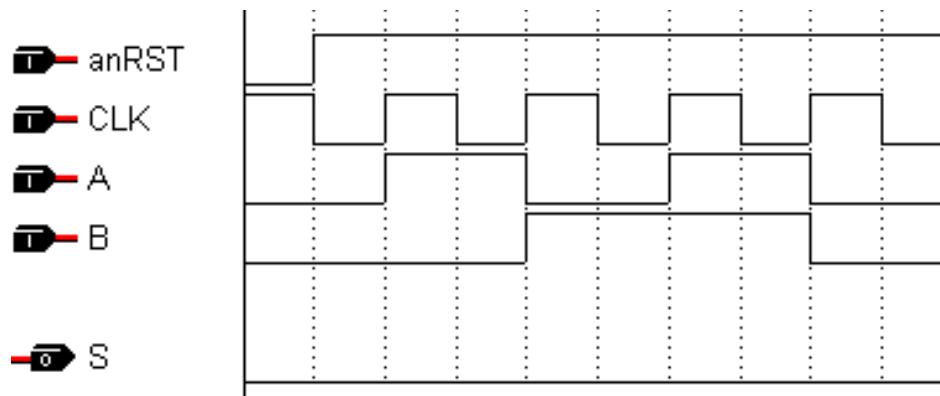
Ejercicio 2

El diagrama de estados (incompleto) de la siguiente figura se corresponde con el circuito del ejercicio anterior. Complete el diagrama de estados indicando para cada estado las transiciones y la salida para cada valor de las entradas.



Ejercicio 3

Analice el circuito presentado en el ejercicio 1 y complete el siguiente cronograma:



Ejercicio 4

Utilizando la lógica adicional que necesite, dote al flip-flop de la figura de:

- Reset asíncrono activo a nivel bajo (anRST).
- Habilitación de reloj activa a nivel bajo (nENA).
- Reset síncrono activo alto, sujeto a la habilitación de reloj (RESET).
- Preset síncrono activo alto, no sujeto a la habilitación de reloj, prioritario sobre el reset síncrono (PRESET).

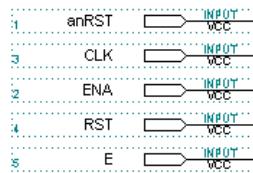


Ejercicio 5

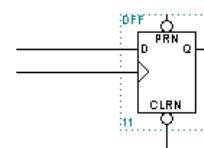
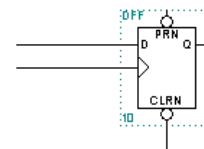
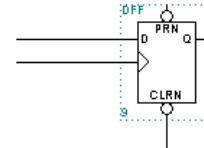
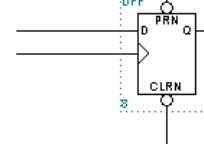
Utilizando flip-flops como los de la figura, diseñe un registro de desplazamiento de 4 bits con habilitación de reloj, reset síncrono, entrada serie y salida paralelo.

- A. Diseño.** Diseñe el registro. El funcionamiento debe ser como el que se resume en la siguiente tabla:

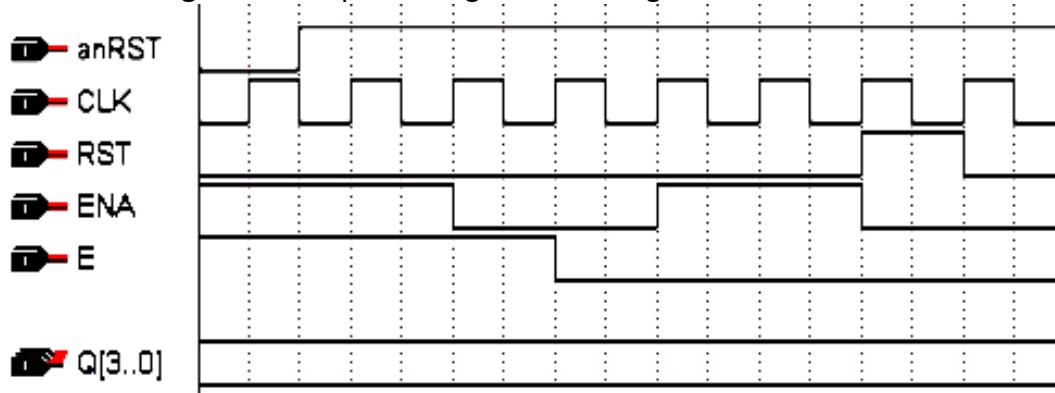
Entradas					Salida				Funcionamiento
anRST	E	ENA	RST	CLK	Q_3	Q_2	Q_1	Q_0	
0	X	X	X	X	0	0	0	0	Reset asíncrono
1	X	X	1	↑	0	0	0	0	Reset síncrono
1	X	0	0	↑	q_3	q_2	q_1	q_0	Memoria
1	e	1	0	↑	q_2	q_1	q_0	e	Desplazamiento



OUTPUT $\rightarrow Q[3..0]$



- B. Cronograma.** Complete el siguiente cronograma:



BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Ejercicios sobre sistemas secuenciales.

CÓDIGO:
BT3_ATGP6

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

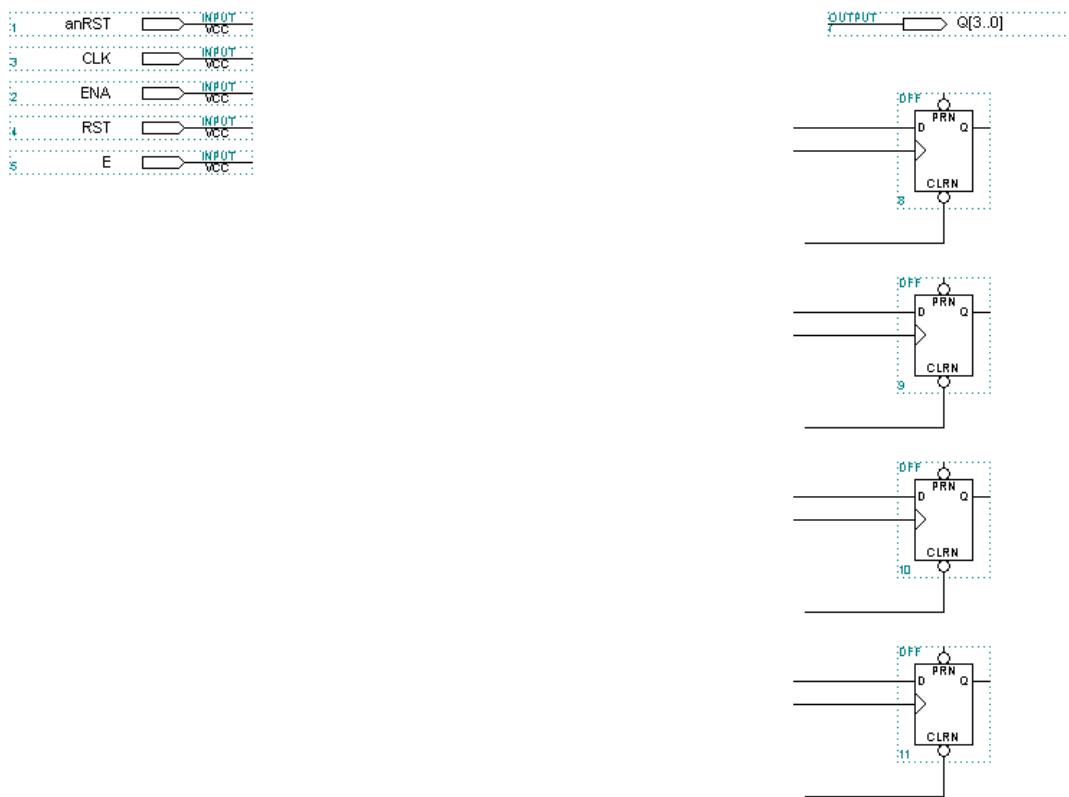
GRUPO		CALIFICACIÓN: [entre 0 y 10 puntos]		puntos
-------	--	--	--	--------

Ejercicio 6

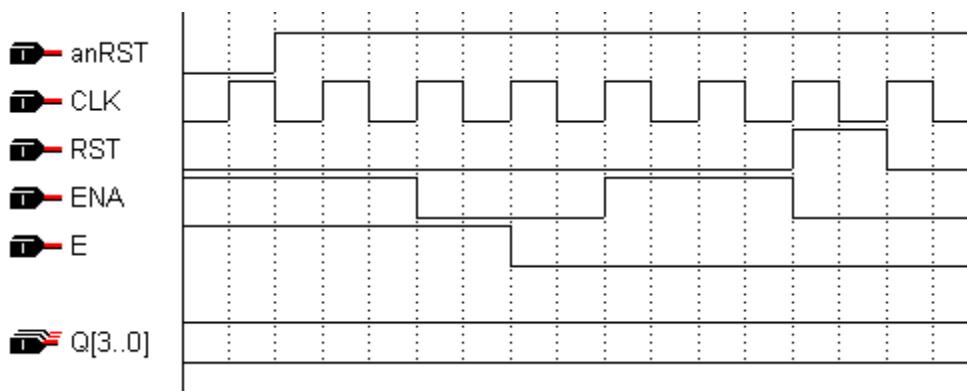
Utilizando flip-flops como los de la figura de la página siguiente, diseñe un registro de desplazamiento de 4 bits con habilitación de reloj, reset síncrono, entrada serie y salida paralelo.

- A. Diseño.** Diseñe el registro. El funcionamiento debe ser como el que se resume en la siguiente tabla:

anRST	Entradas					Salida				Funcionamiento
	E	ENA	RST	CLK		Q ₃	Q ₂	Q ₁	Q ₀	
0	X	X	X	X		0	0	0	0	Reset asíncrono
1	X	0	X	↑	q ₃	q ₂	q ₁	q ₀		Memoria
1	X	1	1	↑	0	0	0	0		Reset síncrono
1	e	1	0	↑	e	q ₃	q ₂	q ₁		Desplazamiento



B. Cronograma. Complete el siguiente cronograma:



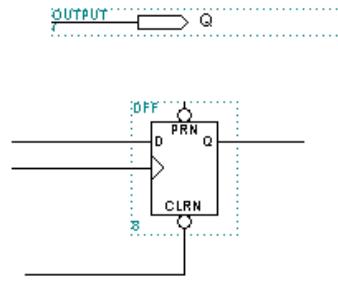
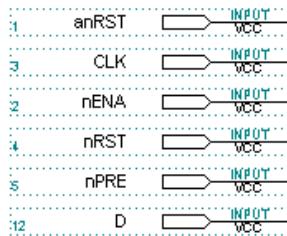
C. Diagrama de estados. Si se contempla el registro como un autómata,

- ¿Cuántos estados tendría?
- ¿Sería un autómata de Mealy o de Moore? ¿Por qué?

D. Partiendo de un flip-flop tipo D como el de la figura, añada la lógica combinacional necesaria para añadir las siguientes entradas:

- Entrada de habilitación activa a nivel bajo, **nENA**.
- Entrada de reset síncrono activa a nivel bajo, **nRST**.
- Entrada de preset síncrono activa a nivel bajo, **nPRE**.

Ambas entradas nRST y nPRE deben requerir que nENA esté activa para actuar, y nRST debe ser prioritaria frente a nPRE.



Esta página se ha dejado intencionadamente en blanco

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Preparación individual del examen.

CÓDIGO:
BT3_AINP7

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Repaso. Trabajo individual.	TIPO:	No presencial	DURACIÓN:	60 minutos
------------	-----------------------------	-------	---------------	-----------	------------

CALENDARIO:	Octava Sesión. Antes del examen del Bloque 3	REQUISITOS:	
-------------	---	-------------	--

CRITERIO DE ÉXITO:	
--------------------	--

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTEOVALUACIÓN: [entre 0 y 10 puntos]	puntos
------------------	---------	--	--------

Consejos para la preparación del examen

- Revise la lista de indicadores en la Guía de la Asignatura. Los indicadores proporcionan información detallada y concreta sobre lo que debe ser capaz de hacer tras realizar las actividades programadas del bloque temático.
- Revise todas las actividades realizadas hasta el momento. Si hay actividades que no ha realizado correctamente, revise el material disponible sobre la temática de esas actividades e intente realizarlas de nuevo.



ETSiST-UPM

Electrónica II

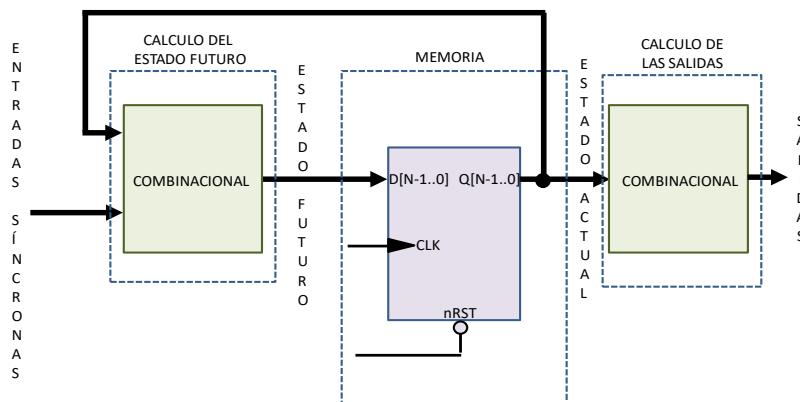
Bloque temático 3

Resumen de la metodología completa de diseño de sistemas secuenciales. Ejemplos de desarrollo de diagramas de estado

Resumen

LA REALIZACIÓN DE UN CIRCUITO SECUENCIAL CONSISTE EN EL DISEÑO DE:

- 1.- LA **MEMORIA DE ESTADO**
- 2.- LA LÓGICA COMBINACIONAL DE **CÁLCULO DEL ESTADO DE MEMORIA**
- 3.- LA LÓGICA COMBINACIONAL DE **CÁLCULO DE LAS SALIDAS**

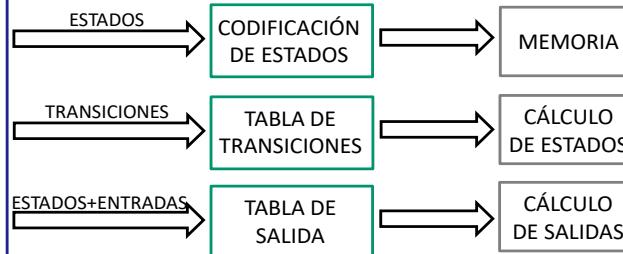
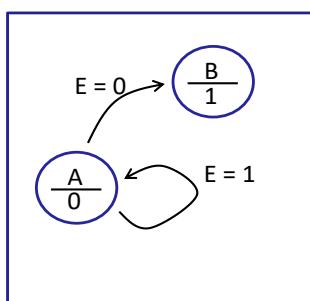


BT3_S9 BT3_P10_2

Resumen

LA METODOLOGÍA SISTEMÁTICA DE DISEÑO DE CIRCUITOS SECUENCIALES UTILIZA LA REPRESENTACIÓN DEL FUNCIONAMIENTO DEL CIRCUITO MEDIANTE UN DIAGRAMA DE ESTADOS COMO PUNTO DE PARTIDA PARA EL DISEÑO DE LOS TRES BLOQUES DE LA ARQUITECTURA: MEMORIA, LÓGICA DE CÁLCULO DEL ESTADO DE MEMORIA Y LÓGICA DE CÁLCULO DE LAS SALIDAS

DIAGRAMA DE ESTADOS



BT3_S9

BT3_P10_3

De especificación a diagrama de estados

- **Especificación:** descripción del conjunto de características y operaciones que definen el comportamiento de un autómata.
- Es la traducción de la información que especifica el funcionamiento del autómata al diagrama de estado. No hay una metodología formal.
- Es el paso previo a la metodología de diseño de circuitos secuenciales que ya conoce.

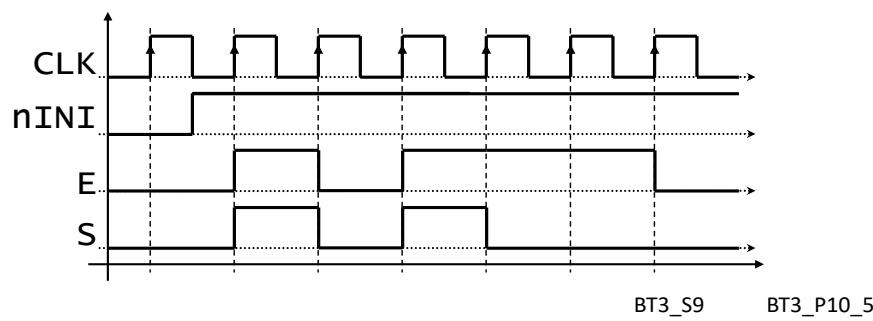
Solución: Entrenar para adquirir destreza y experiencia

BT3_S9

BT3_P10_4

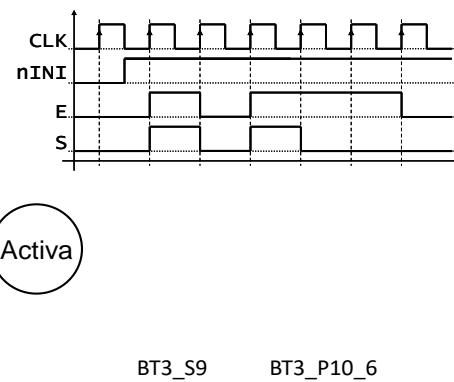
Ejemplo 1: Conformador de pulsos

- **Especificación:** circuito secuencial que hace que su salida valga ‘1’ durante un ciclo de reloj cuando se activa su entrada (independientemente del tiempo que esté activada).
- Cronograma adjunto a la especificación:



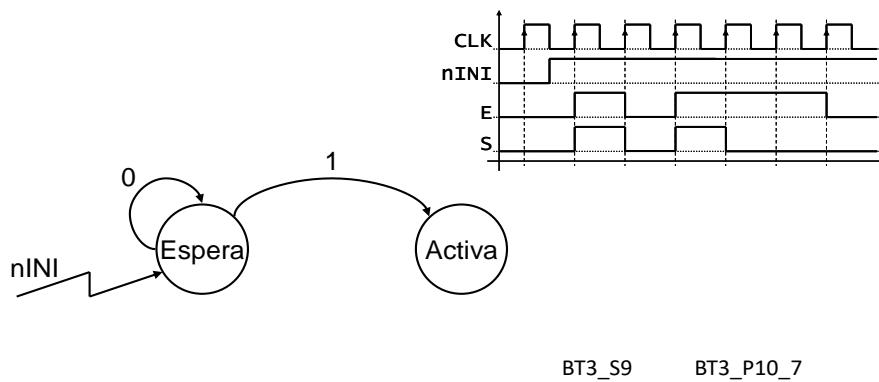
Ejemplo 1: Estados

- Previsión del número de estados.
 - Estado de espera a que se active la entrada.
 - Estado de entrada activada.



Ejemplo 1: Transiciones

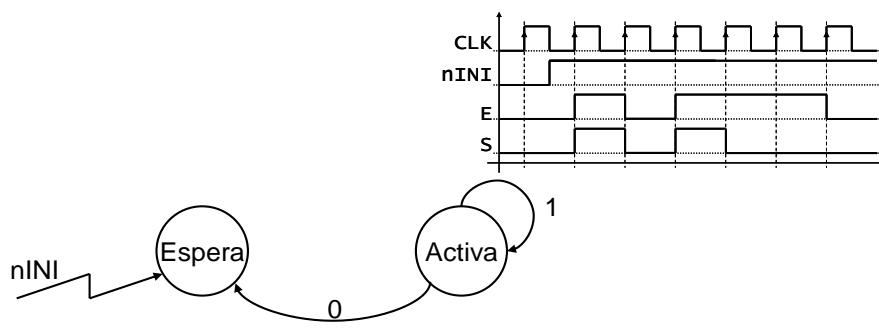
- Estado “Espera”
 - Si $E = '0'$, la entrada no se ha activado y, por tanto, el autómata seguirá en ese estado.
 - Si $E = '1'$, la entrada se ha activado y hay que generar el pulso en la salida.



BT3_S9 BT3_P10_7

Ejemplo 1: Transiciones

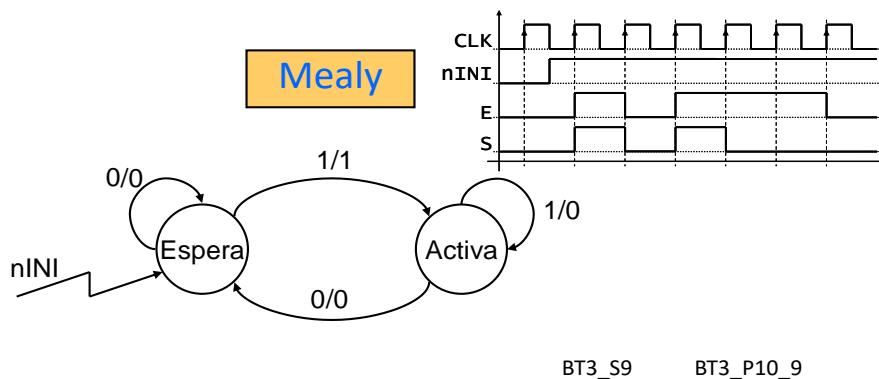
- Estado “Activa”
 - Si $E = '0'$, la entrada no se ha activado y, por tanto, el autómata vuelve a espera.
 - Si $E = '1'$, la entrada se ha activado y hay que permanecer en activa pero sin generar el pulso en la salida.



BT3_S9 BT3_P10_8

Ejemplo 1: Tipo de autómata. Salidas

- En este punto del análisis ya se puede identificar el tipo de autómata.
 - En el estado “Espera”, S es ‘0’ si E es ‘0’, y S es ‘1’ si E es ‘1’. Es decir, la salida depende del estado y de la entrada.
 - ¿Qué tipo de autómata es?



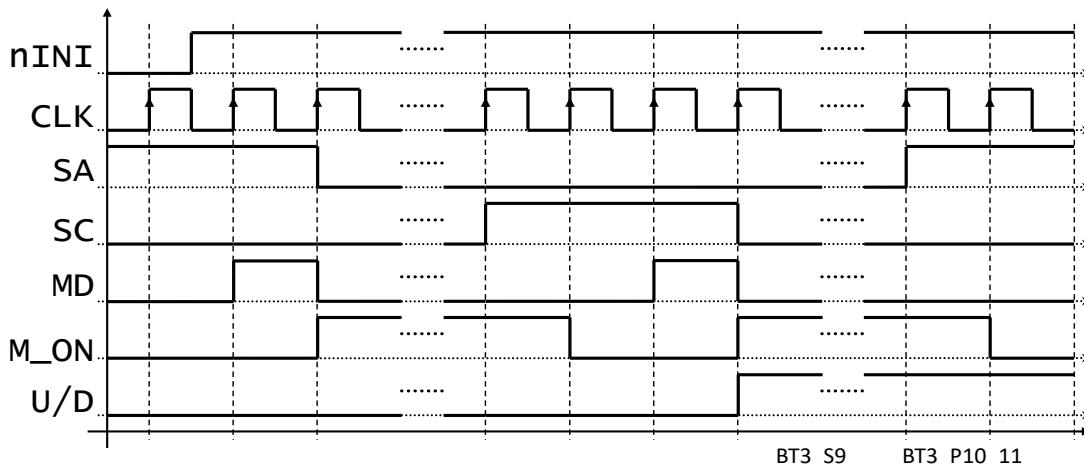
Ejemplo 2: Puerta de un garaje

- Especificación:** circuito secuencial que controla la apertura de una puerta de garaje. El sistema tiene dos sensores que indican si la puerta está totalmente abierta o totalmente cerrada, un motor que sube la puerta para abrirla y la baja para cerrarla y un mando a distancia para accionar la puerta.
- Entradas:**
 - Sensor de puerta abierta (SA): se activa a nivel alto cuando la puerta está totalmente abierta.
 - Sensor de puerta cerrada (SC): ídem para cuando esta cerrada.
 - Mando a distancia (MD): indica con un pulso de duración un ciclo de reloj que se abra la puerta (si estaba cerrada) o que se cierre (si estaba abierta).
- Salidas:**
 - Enciende motor (M_ON): pone el motor de la puerta en funcionamiento cuando vale ‘1’.
 - Dirección de giro del motor (U/D): vale ‘1’ cuando el motor gira para subir la puerta y ‘0’ cuando gira para bajarla.

BT3_S9 BT3_P10_10

Ejemplo 2: Puerta de un garaje

- **Especificación (Cont).**
- Estado inicial: puerta totalmente abierta y motor apagado.
- Cronograma adjunto a la especificación:



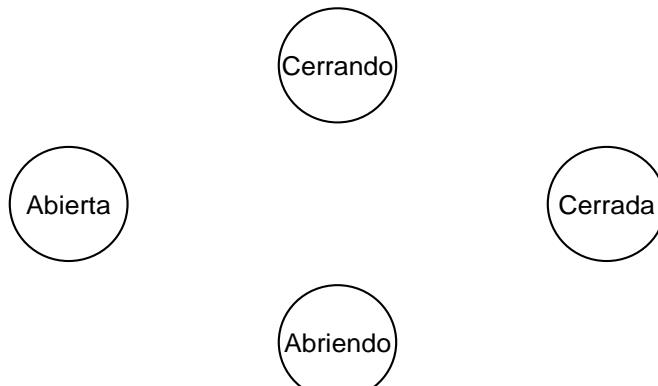
Ejemplo 2: Estados

- Previsión del número de estados.
 - Estado con la puerta totalmente abierta.
 - Estado de puerta cerrándose.
 - Estado con la puerta totalmente cerrada.
 - Estado de puerta abriéndose.

BT3_S9 BT3_P10_12

Ejemplo 2: Estados

- Previsión del número de estados.



BT3_S9 BT3_P10_13

Ejemplo 2: Transiciones

- Hay 3 entradas, por tanto, hay 8 posibles transiciones, aunque algunas no son posibles, en concreto, las transiciones que representa que la puerta está cerrada y abierta a la vez ($SA = SC = '1'$).
- Estado “Abierta”
 - El autómata permanecerá en este estado hasta que reciba la orden del mando a distancia (MD), independientemente del valor del resto de entradas. Es decir, cuando $MD = '1'$ irá al estado “Cerrando”.
- Estado “Cerrando”
 - Permanecerá en este estado hasta que la puerta esté totalmente cerrada, momento en el que irá al estado “Cerrada”.

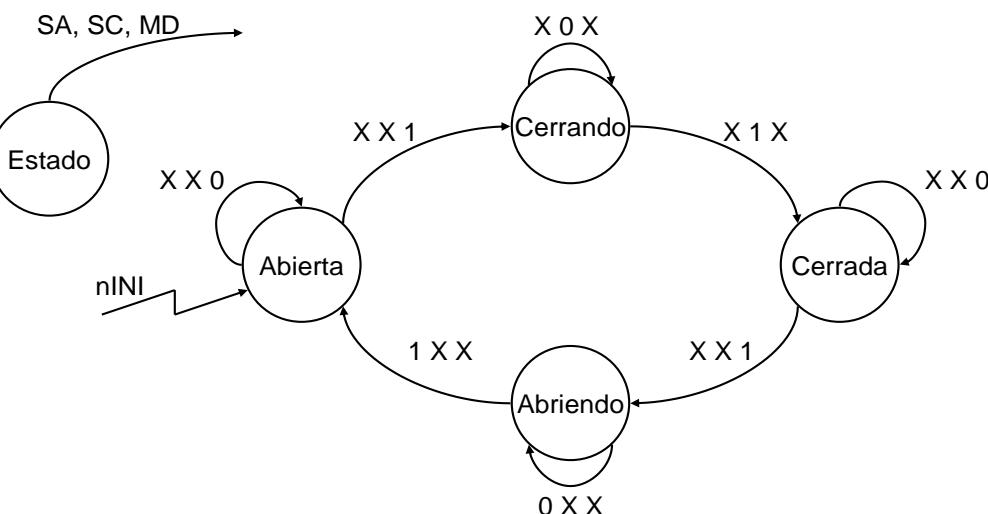
BT3_S9 BT3_P10_14

Ejemplo 2: Transiciones

- Estado “Cerrada”
 - En este estado permanecerá hasta que reciba la orden del mando a distancia (MD), independientemente del valor del resto de entradas. Es decir, cuando MD = ‘1’ ira al estado “Abriendo”.
- Estado “Abriendo”
 - Permanecerá en este estado hasta que la puerta esté totalmente Abierta, momento en el que irá al estado “Abierta”.

BT3_S9 BT3_P10_15

Ejemplo 2: Transiciones



BT3_S9 BT3_P10_16

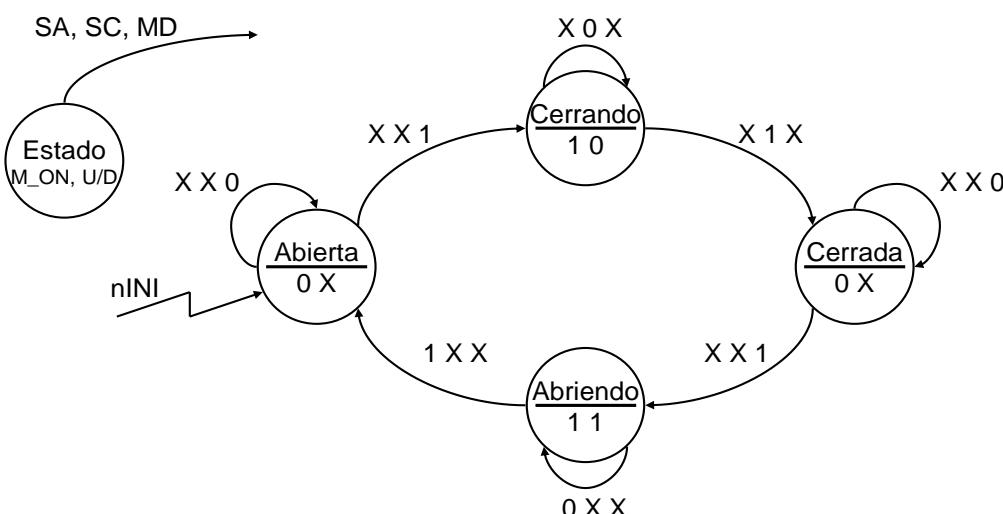
Ejemplo 2: Tipo de autómata. Salidas

- En el cronograma se observa que el motor y la dirección del motor se activan un ciclo después de activarse la señal de entrada. Esto nos induce a pensar que es un autómata de Moore.
- Salidas
 - Estado “Abierta”: M_ON = ‘0’ y U/D = ‘X’ (es indiferente pues el motor está parado).
 - Estado “Cerrando”: M_ON = ‘1’ y U/D = ‘0’.
 - Estado “Cerrada”: M_ON = ‘0’ y U/D = ‘X’ (es indiferente pues el motor está parado).
 - Estado “Abriendo”: M_ON = ‘1’ y U/D = ‘1’.

Las salidas dependen del estado. Se confirma el tipo de autómata

BT3_S9 BT3_P10_17

Ejemplo 2: Tipo de autómata. Salidas



BT3_S9

BT3_P10_18

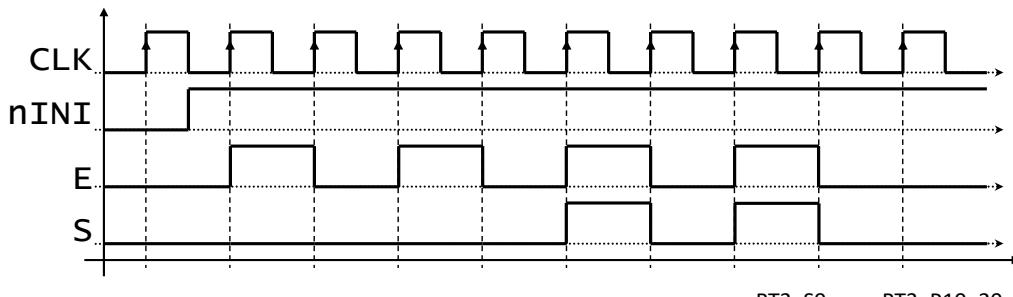
Ejemplo 3: Detector de secuencia

- **Especificación:** circuito secuencial que detecta la ocurrencia de una determinada secuencia de bits que llegan en serie por una entrada síncrona de datos. La secuencia detectar es ‘10101’. Cuando esto ocurra la salida se activa a nivel alto durante un ciclo de reloj.
- Entrada:
 - Entrada de datos (E) de un solo bit.
- Salidas:
 - Salida (S) de un solo bit activa a nivel alto durante un ciclo de reloj.

BT3_S9 BT3_P10_19

Ejemplo 3: Detector de secuencia

- **Especificación (Cont.)**
- Para elegir el estado inicial suponga que el circuito está recibiendo ceros todo el tiempo.
- Cronograma adjunto a la especificación:
 - Observe que parte de la secuencia se “reutiliza”.



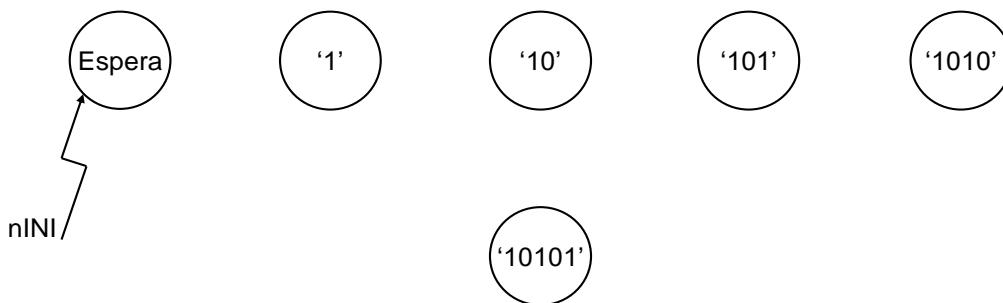
BT3_S9 BT3_P10_20

Ejemplo 3: Estados

- Previsión del número de estados.
 - Estado de espera a que por E entre un ‘1’ (“Espera”).
 - Estado “Ha llegado el primer 1” (‘1’).
 - Estado “Ha llegado un ‘0’ a continuación del ‘1’ (“10”).
 - Estado “Ha llegado un ‘1’ a continuación de la secuencia ‘10’ (‘101’).
 - Así sucesivamente hasta completar la secuencia ‘10101’.

BT3_S9 BT3_P10_21

Ejemplo 3: Estados



BT3_S9 BT3_P10_22

Ejemplo 3: Transiciones

- Estado “Espera”
 - Si $E = '0'$, el bit entrante no pertenece a la secuencia y, por tanto, el autómata seguirá en este estado.
 - Si $E = '1'$, el bit entrante si pertenece a la secuencia y el autómata pasa al estado ‘1’.
- Estado ‘1’
 - Si $E = '1'$, el bit entrante no pertenece a la secuencia y, por tanto, el autómata vuelve al estado de primer 1 de la secuencia.
 - Si $E = '0'$, el bit entrante si pertenece a la secuencia y el autómata pasa el siguiente estado ‘10’.

BT3_S9

BT3_P10_23

Ejemplo 3: Transiciones

- Estado “10”
 - Si $E = '0'$, el bit entrante no pertenece a la secuencia y, por tanto, el autómata vuelve al estado inicial “Espera”.
 - Si $E = '1'$, el bit entrante si pertenece a la secuencia y el autómata pasa al siguiente estado ‘101’.
- Estado ‘101’
 - Si $E = '1'$, el bit entrante no pertenece a la secuencia y, por tanto, el autómata evoluciona al estado ‘1’ (primer ‘1’ de la secuencia).
 - Si $E = '0'$, el bit entrante si pertenece a la secuencia y el autómata pasa al estado ‘1010’.

BT3_S9

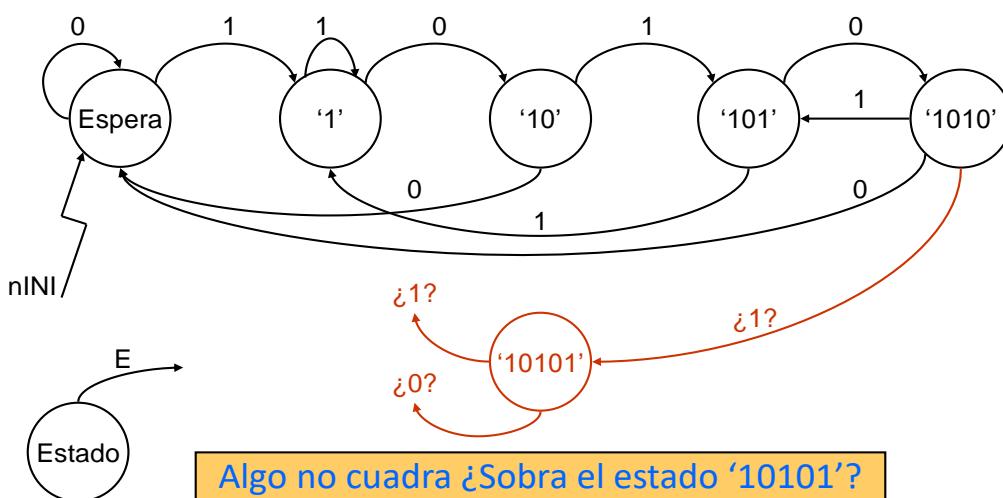
BT3_P10_24

Ejemplo 3: Transiciones

- Estado “1010”
 - Si $E = '0'$, el bit entrante no pertenece a la secuencia y, por tanto, el autómata vuelve al estado inicial “Espera”.
 - Si $E = '1'$, el bit entrante si pertenece a la secuencia y el autómata pasa al estado ‘10101’ puesto que ya había llegado anteriormente la secuencia ‘1010’.
- Estado ‘10101’
 - ¡Atención! En el estado 1010, al llegar por la entrada E un ‘1’ ya ha ocurrido la secuencia que se buscaba.
 - ¿Realmente hace falta este estado?

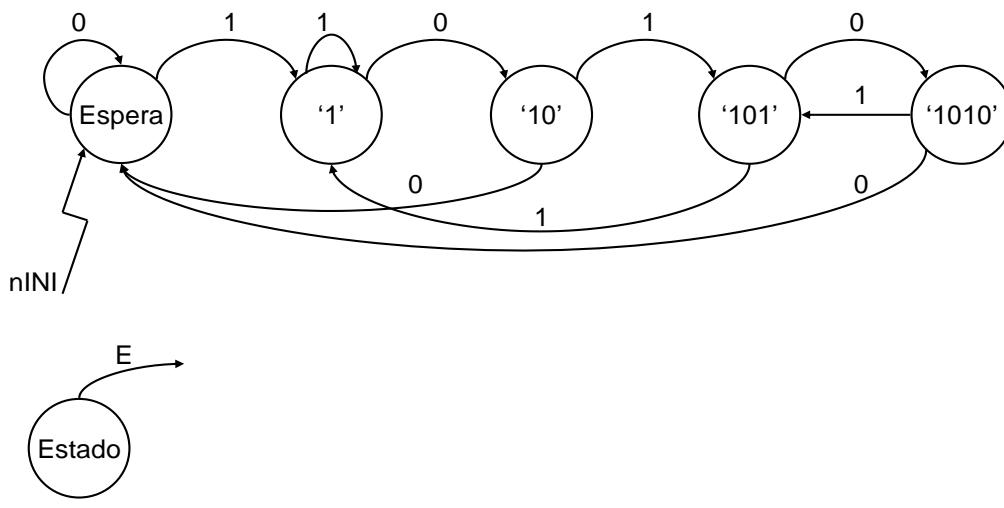
BT3_S9 BT3_P10_25

Ejemplo 3: Transiciones

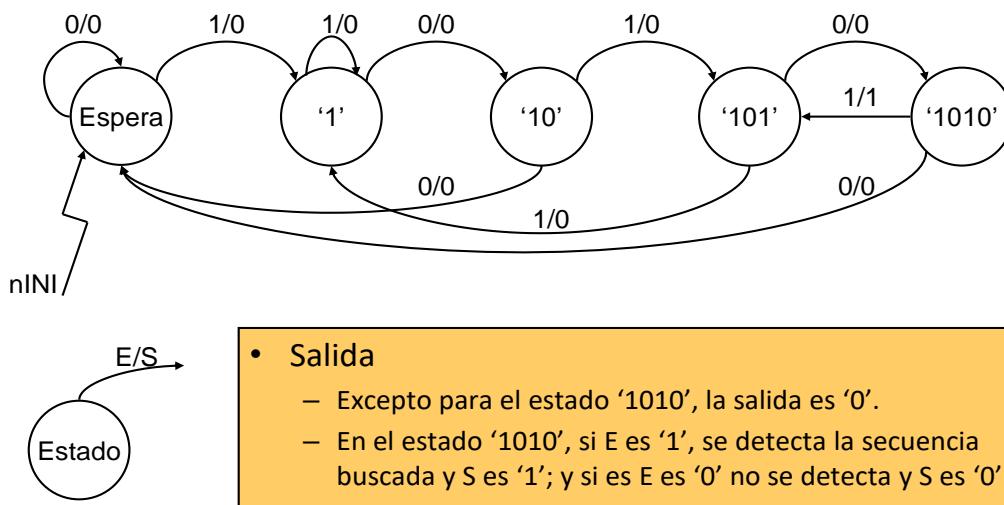


BT3_S9 BT3_P10_26

Ejemplo 3: Transiciones



Ejemplo 3: Tipo de autómata. Salidas



BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Ejercicios de diseño de diagramas de estados.

CÓDIGO:
BT3_ATGP7

GRUPO		FECHA:	
-------	--	--------	--

MODALIDAD:	Trabajo en grupo	TIPO:	Presencial	DURACIÓN:	50 minutos
------------	------------------	-------	------------	-----------	------------

CALENDARIO:	Novena Sesión	REQUISITOS:	
-------------	---------------	-------------	--

CRITERIO DE ÉXITO:	Realización correcta de los ejercicios.
--------------------	---

ROLES:	No hay distribución diferenciada de roles entre los miembros
--------	--

HABILIDADES SOCIALES:	Capacidad para expresar y sintetizar ideas, capacidad para escuchar, asimilar y organizar la información presentada por otra persona. Capacidad para razonar y analizar.
-----------------------	--

REFLEXIÓN DE GRUPO:	
---------------------	--

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	
------------------	------------	-------------------------------------	--

Descripción de la actividad

Esta actividad se compone de dos partes. Cada parte consta de 2 ejercicios de un mismo tipo. Los miembros del grupo formarán dos subgrupos. Cada subgrupo realizará uno de los ejemplos de cada parte. Una vez resuelto el primer ejercicio, los dos subgrupos pondrán en común los resultados. A continuación el profesor mostrará la solución de los dos ejercicios. Así se realizarán y discutirán todos los ejercicios de la actividad.

PARTE I. Diseño de contadores basados en diagramas de estados

Ejercicio 1. Obtenga el diagrama de estados de un autómata de Moore que modele un contador de módulo 4 que dispone de una entrada de control síncrona **E**. Cuando la entrada **E** = '0' el contador realiza la cuenta 0, 1, 2, 3, 0, ... y cuando **E** = '1' realiza la cuenta 0, 1, 3, 2, 0, ... El estado inicial es el correspondiente a la cuenta 0 y se consigue activando la señal de control asíncrona **nINI** (activa a nivel bajo).

Ejercicio 2. Obtenga el diagrama de estados de un contador que funcione de la siguiente manera:

- El contador realiza de forma cíclica una cuenta ascendente generando la siguiente secuencia de números: 1, 3, 6, 9 y 12.
- Dispone de una entrada de control síncrona de habilitación de la cuenta activa a nivel bajo (**nEN**).
- Dispone de una segunda señal de control síncrona que reinicia la cuenta llevando al contador al primer estado de cuenta. Esta señal de reset debe ser activa a nivel alto (**RST**). En caso de que el contador esté deshabilitado y se active la señal de reset el contador debe permanecer en el valor de cuenta en el que se encuentre.
- El estado inicial es el correspondiente a la cuenta 1 y se consigue activando la señal de control asíncrona **nINI** (activa a nivel bajo).

PARTE II. Diseño de detectores de secuencias basados en diagramas de estados

Ejercicio 1. Obtenga el diagrama de estados de un autómata de Mealy que pone a uno su salida si reconoce una secuencia de entrada que tenga, al menos, cuatro unos consecutivos. El circuito tiene una única síncrona, **X**, y una única salida, **Y**, activa a nivel alto. El circuito activa su salida **Y** si la señal de entrada **X** es igual a 1, al menos durante cuatro ciclos de reloj consecutivos. Cuando se detecte la condición mencionada, el circuito mantiene la salida a '1' hasta que se presente un '0', iniciándose de nuevo el proceso de detección. Indique cuál es el estado inicial del circuito en el que el autómata debería estar cuando se activa la señal de control asíncrona **nINI** (activa a nivel bajo).

Ejercicio 2. Obtenga el diagrama de estados de un autómata de Mealy que tiene dos entradas síncronas (**X** e **Y**) de un bit por las que se reciben bits en serie y una salida (**S**) de un bit. Esta salida se activará (a nivel alto) cada vez que el número de unos recibidos en dos ciclos de reloj consecutivos sea impar, permaneciendo a cero en caso contrario. Con la señal de control asíncrona **nINI** (activa a nivel bajo), seleccione como estado inicial aquel que corresponda a la recepción anterior de un número indefinido de ceros.

BLOQUE TEMÁTICO 3

TÍTULO DE LA ACTIVIDAD:
Ejercicios sobre sistemas secuenciales.

CÓDIGO:
BT3_AINP8

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Ejercicios individuales.	TIPO:	No presencial	DURACIÓN:	120 minutos
------------	--------------------------	-------	---------------	-----------	-------------

CALENDARIO:	Novena Sesión. Antes de la Décima Sesión	REQUISITOS:	Sistemas secuenciales: flip-flops, registros. Arquitectura de los sistemas secuenciales.
-------------	---	-------------	--

CRITERIO DE ÉXITO:	Resolución correcta de cinco de los seis ejercicios propuestos
--------------------	--

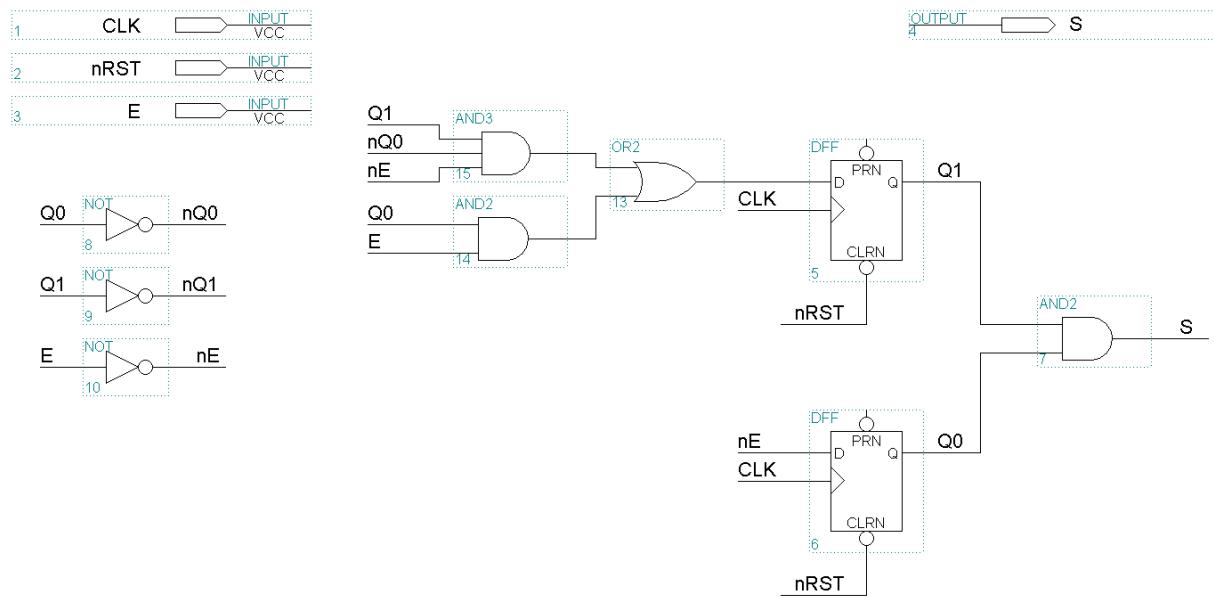
COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTEOVALUACIÓN: [entre 0 y 10 puntos]	
------------------	---------	--	--

Actividad BT3_AINP8: Ejercicios sobre sistemas secuenciales.

Ejercicio 1

Complete los siguientes apartados para el circuito de la figura, que detecta la aparición de una determinada secuencia de ceros y unos en la entrada “E” y activa su salida “S” cuando la secuencia se produce:



- A. **Estructura.** Identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.
- B. **Salidas.** Obtenga la ecuación de la salida en función del estado actual. Complete la tabla de verdad que se adjunta.

$S =$

Estado	Código		Salida S
	Q1	Q0	
A	0	0	
B	0	1	
C	1	0	
D	1	1	

Actividad BT3_AINP8: Ejercicios sobre sistemas secuenciales.

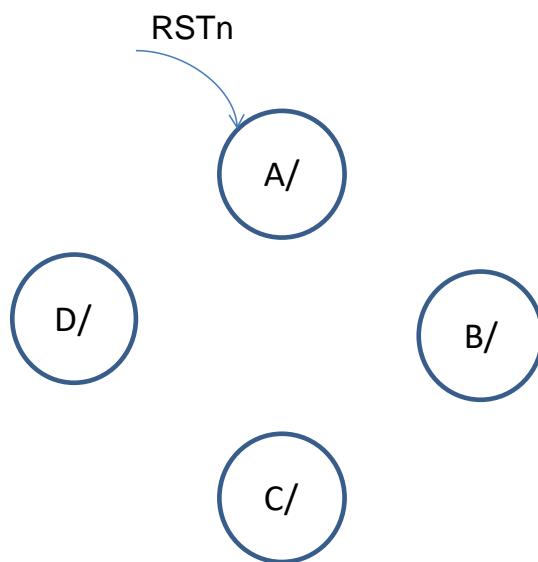
- C. Estado futuro.** Obtenga las ecuaciones del estado futuro en función del estado actual y de las entradas. Complete la tabla de verdad que se adjunta.

$$D0 =$$

$$D1 =$$

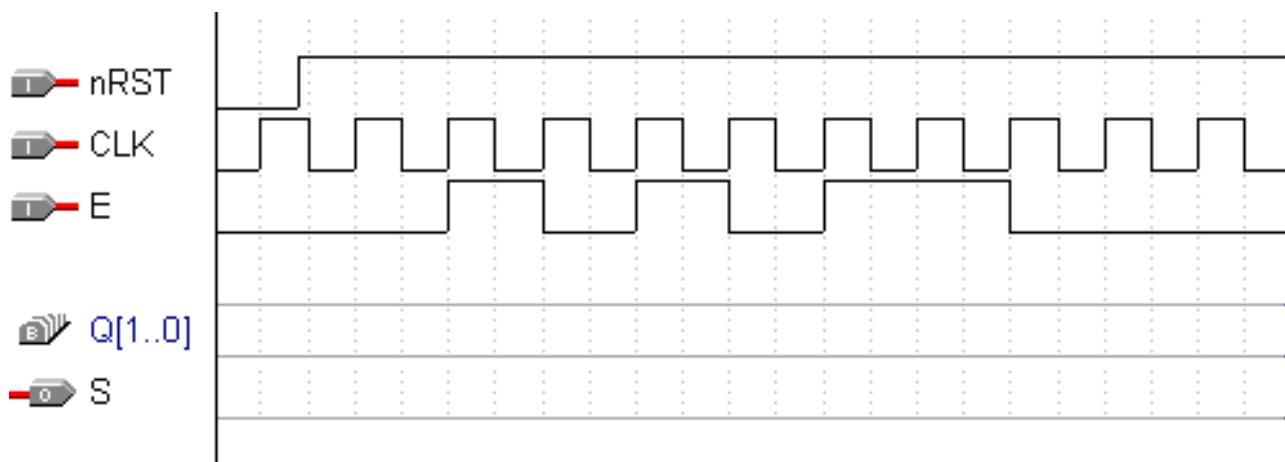
Entrada	Estado actual		Estado futuro	
	E	$Q1_t$	$Q0_t$	$D1 = Q1_{t+1}$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

- D. Diagrama de estados.** Complete el diagrama de estados especificando las salidas correspondientes a cada estado y las transiciones entre los estados.



Actividad BT3_AINP8: Ejercicios sobre sistemas secuenciales.

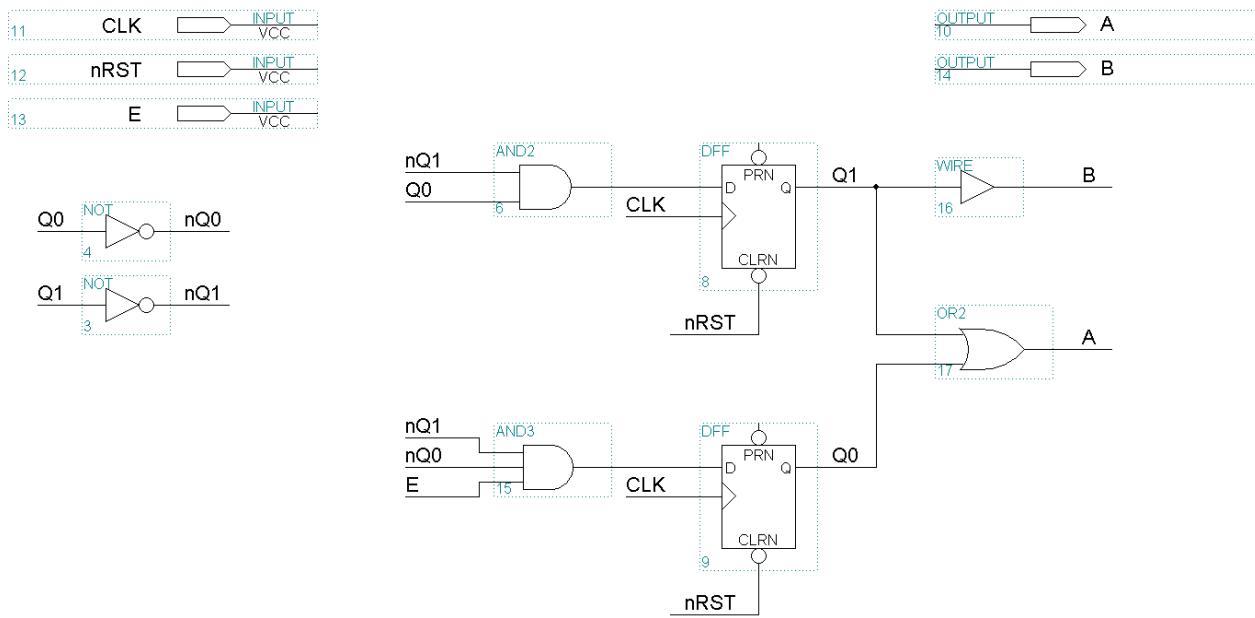
E. Análisis del circuito. Analice el circuito y complete el siguiente cronograma:



Actividad BT3_AINP8: Ejercicios sobre sistemas secuenciales.

Ejercicio 2

Complete los siguientes apartados para el circuito de la figura¹.



- A. **Estructura.** Identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.
- B. **Salidas.** Obtenga la ecuación de la salida en función del estado actual. Complete la tabla de verdad que se adjunta.

$$A =$$

$$B =$$

Estado	Código		Salida	
	Q1	Q0	A	B
A	0	0		
B	0	1		
C	1	0		
D	1	1		

¹ En caso de que no lo conozca, el componente WIRE de MAX+plus II reproduce en su salida el mismo valor lógico que tenga en su entrada.

Actividad BT3_AINP8: Ejercicios sobre sistemas secuenciales.

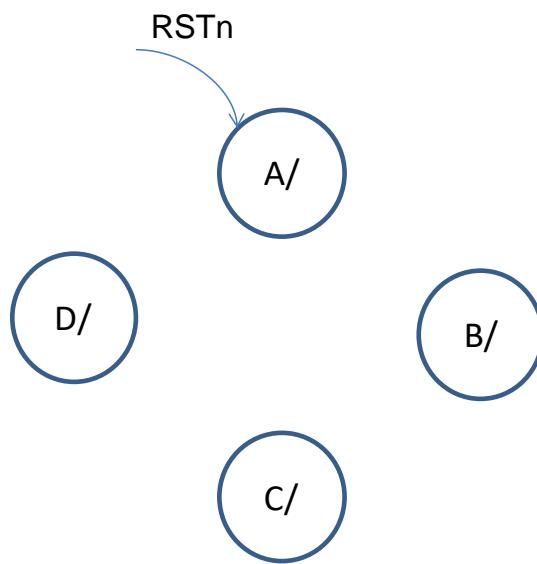
- C. Estado futuro.** Obtenga las ecuaciones del estado futuro en función del estado actual y de las entradas. Complete la tabla de verdad que se adjunta.

$$D0 =$$

$$D1 =$$

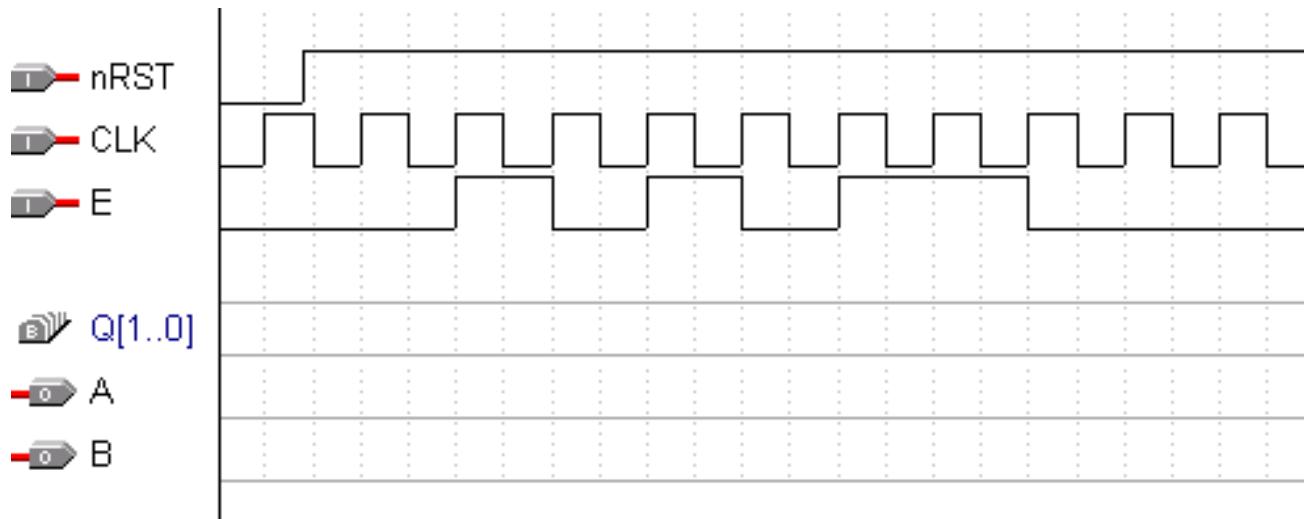
Entrada	Estado actual		Estado futuro	
	$Q1_t$	$Q0_t$	$D1 = Q1_{t+1}$	$D0 = Q0_{t+1}$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

- D. Diagrama de estados.** Complete el diagrama de estados especificando las salidas correspondientes a cada estado y las transiciones entre los estados.



Actividad BT3_AINP8: Ejercicios sobre sistemas secuenciales.

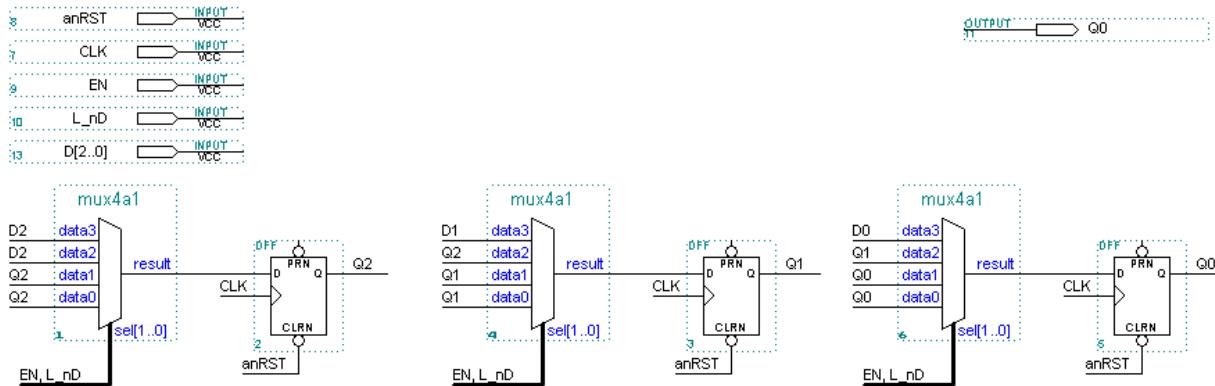
- E. Análisis del circuito.** El circuito es un secuenciador que produce un determinado patrón de valores en sus salidas A y B cada vez que se activa (a nivel alto) la entrada E. Analice el circuito y complete el siguiente cronograma:



Actividad BT3_AINP8: Ejercicios sobre sistemas secuenciales.

Ejercicio 3

El circuito de la figura representa un registro de desplazamiento de tres bits con habilitación de reloj y carga en paralelo. Los multiplexores de 4 a 1 que contiene funcionan como se indica en la tabla adjunta.

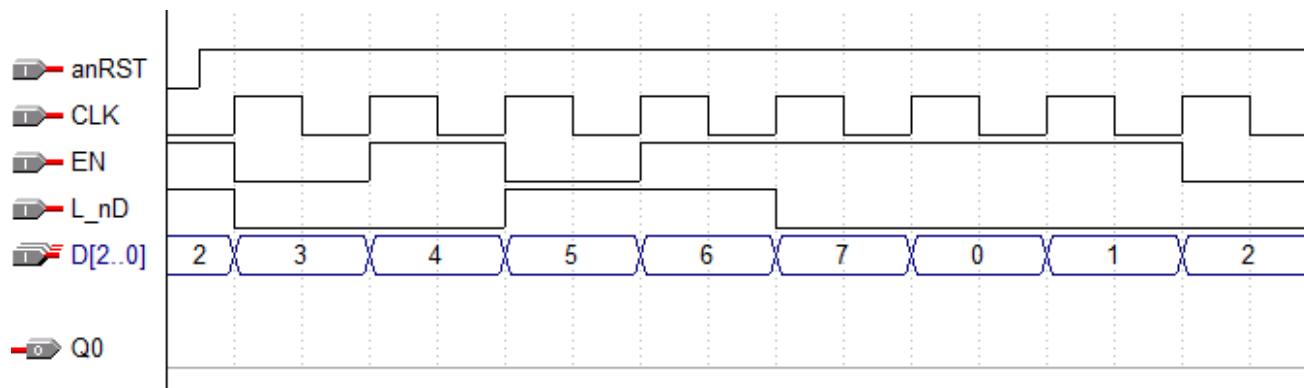


sel[1..0]	result
00	data0
01	data1
10	data2
11	data3

Complete la siguiente tabla, que describe el funcionamiento del registro.

Entradas						Salida	Funcionamiento
anRST	EN	L_nD	CLK	Q ₂	Q ₁	Q ₀	
0	X	X	X	0	0	0	Reset asíncrono
1			↑	Q _{2(t-1)}	Q _{1(t-1)}	Q _{0(t-1)}	Memoria
1			↑	D ₂	D ₁	D ₀	Carga
1			↑		Q _{2(t-1)}	Q _{1(t-1)}	Desplazamiento

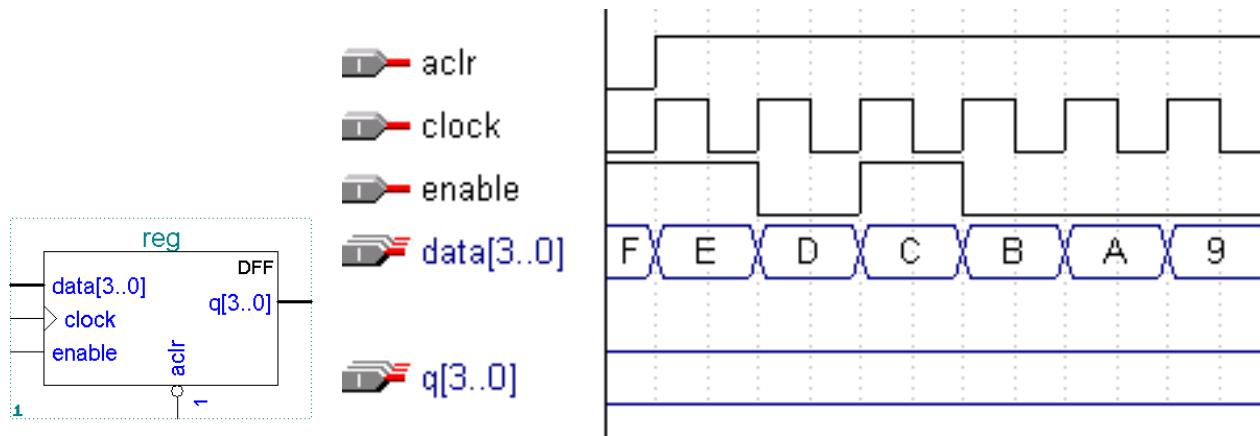
Rellene el siguiente cronograma:



Actividad BT3_AINP8: Ejercicios sobre sistemas secuenciales.

Ejercicio 4

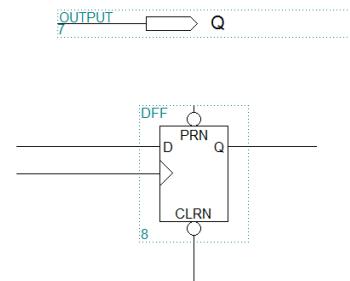
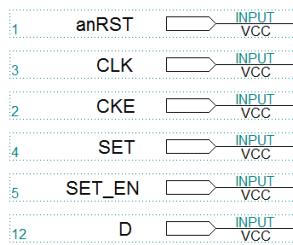
Complete el siguiente cronograma, correspondiente al registro paralelo de 4 bits con habilitación de reloj cuyo símbolo se muestra a continuación. Los valores de la entrada **data[3..0]** se representan en hexadecimal; utilice la misma base para representar los datos de la salida **q[3..0]**.



Ejercicio 5

Empleando la lógica adicional que necesite, dote al flip-flop tipo D de la figura de las siguientes entradas:

- Entrada de reset asíncrono activo a nivel bajo, **anRST**.
- Entrada de habilitación de reloj, **CKE**, activa a nivel alto.
- Entrada de preset síncrono, **SET_EN**, habilitada por CKE.
- Entrada de preset síncrono, **SET**, que no precise de CKE para actuar.

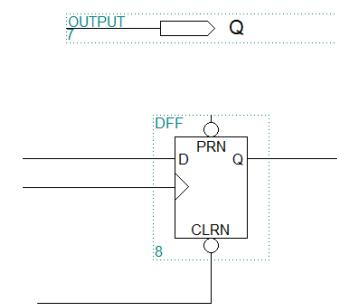
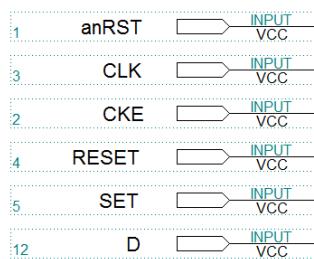


Actividad BT3_AINP8: Ejercicios sobre sistemas secuenciales.

Ejercicio 6

Empleando la lógica adicional que necesite, dote al flip-flop tipo D de la figura de las siguientes entradas:

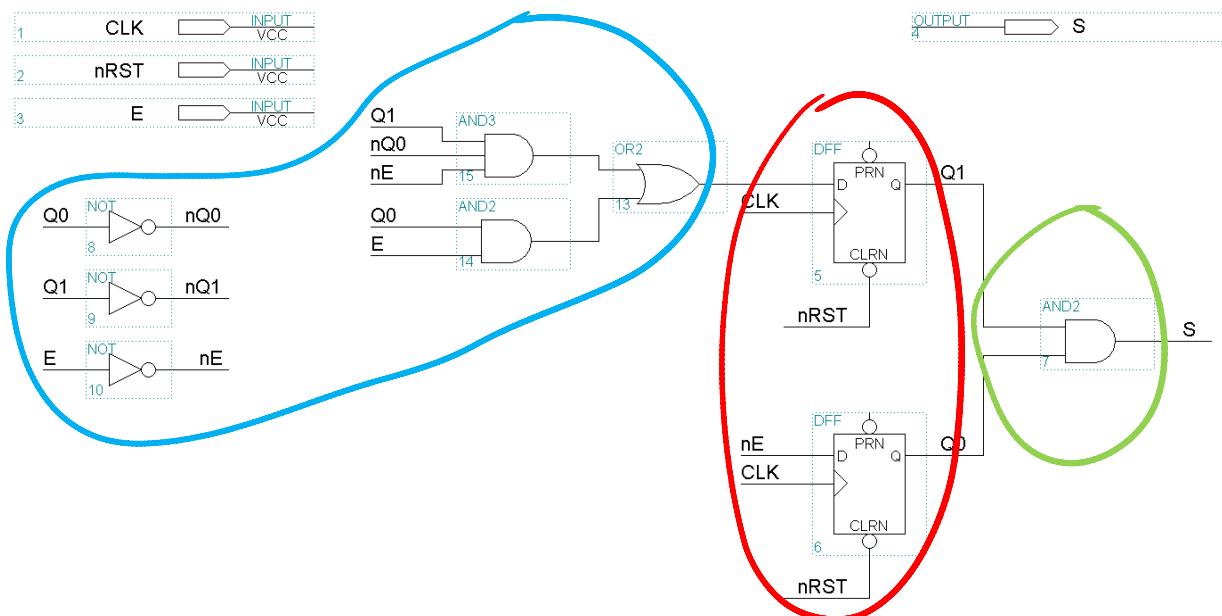
- Entrada de reset asíncrono activo a nivel bajo, **anRST**.
- Entrada de habilitación de reloj, **CKE**, activa a nivel alto.
- Entrada de reset síncrono, **RESET**, habilitada por CKE.
- Entrada de preset síncrono, **SET**, también habilitada por CKE y prioritaria sobre RESET.



Soluciones

Ejercicio 1

Complete los siguientes apartados para el circuito de la figura, que detecta la aparición de una determinada secuencia de ceros y unos en la entrada “E” y activa su salida “S” cuando la secuencia se produce:



- A. **Estructura.** Identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.
- B. **Salidas.** Obtenga la ecuación de la salida en función del estado actual. Complete la tabla de verdad que se adjunta.

$$S = \underline{Q_1} \cdot \underline{Q_0}$$

Estado	Código		Salida
	Q1	Q0	
A	0	0	0
B	0	1	0
C	1	0	1
D	1	1	0

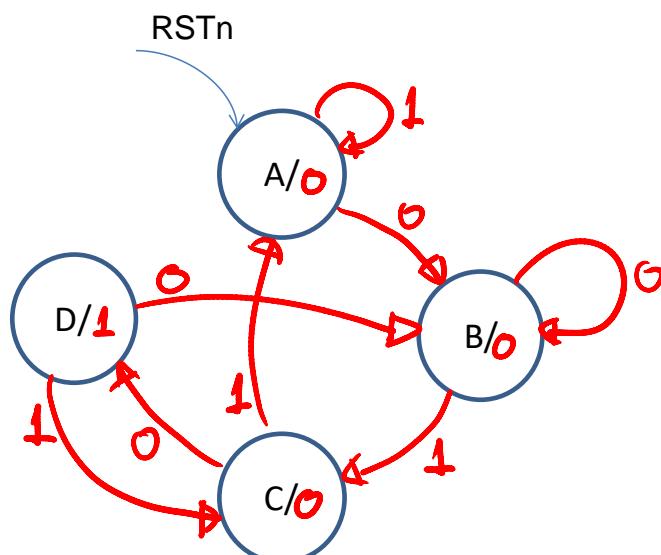
C. Estado futuro. Obtenga las ecuaciones del estado futuro en función del estado actual y de las entradas. Complete la tabla de verdad que se adjunta.

$$D_0 = \overline{E}$$

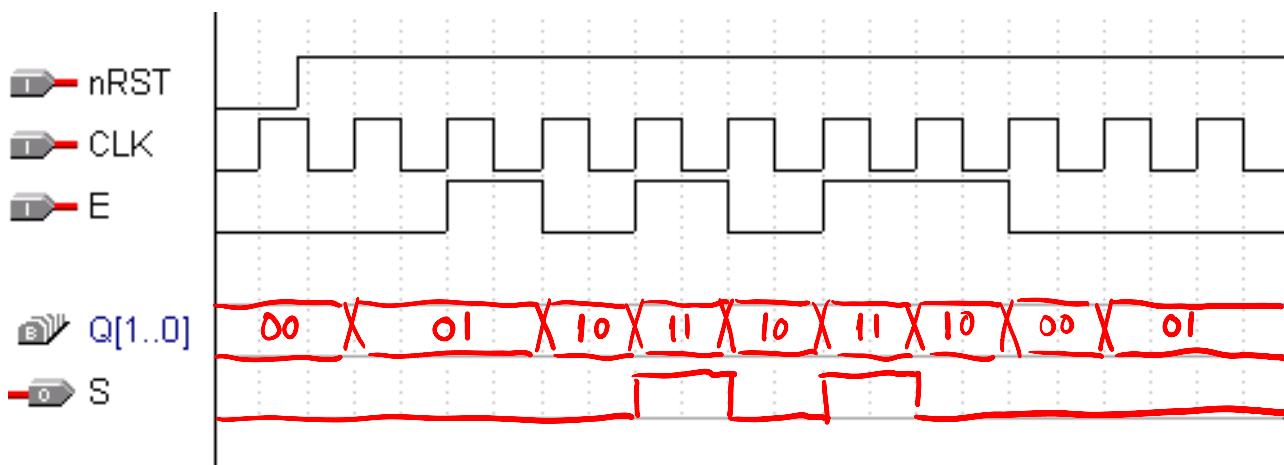
$$D_1 = Q_0 \cdot E + Q_1 \cdot \overline{Q_0} \cdot \overline{E}$$

Entrada	Estado actual		Estado futuro		
	E	Q _{1t}	Q _{0t}	D ₁ =Q _{1t+1}	
0	A	0	0	0	1
0	B	0	1	0	1
0	C	1	0	1	1
0	D	1	1	0	1
1	A	0	0	0	0
1	B	0	1	1	0
1	C	1	0	0	0
1	D	1	1	1	0

D. Diagrama de estados. Complete el diagrama de estados especificando las salidas correspondientes a cada estado y las transiciones entre los estados.

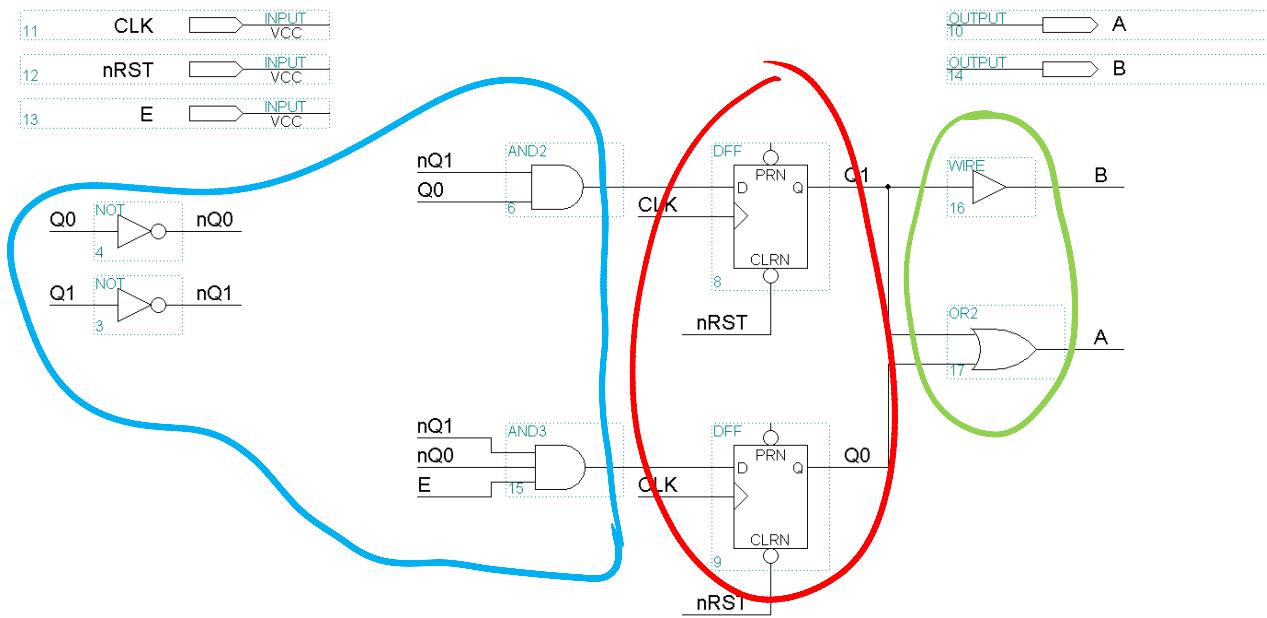


E. Análisis del circuito. Analice el circuito y complete el siguiente cronograma:



Ejercicio 2

Complete los siguientes apartados para el circuito de la figura².



- A. **Estructura.** Identifique, rodeando con un círculo, la memoria, la lógica para el cálculo de las salidas y la lógica para el cálculo del estado futuro.
- B. **Salidas.** Obtenga la ecuación de la salida en función del estado actual. Complete la tabla de verdad que se adjunta.

$$A = Q_0 + Q_1$$

$$B = Q_1$$

Estado	Código		Salida	
	Q1	Q0	A	B
A	0	0	0	0
B	0	1	1	0
C	1	0	1	1
D	1	1	1	1

² En caso de que no lo conozca, el componente WIRE de MAX+plus II reproduce en su salida el mismo valor lógico que tenga en su entrada.

C. Estado futuro. Obtenga las ecuaciones del estado futuro en función del estado actual y de las entradas. Complete la tabla de verdad que se adjunta.

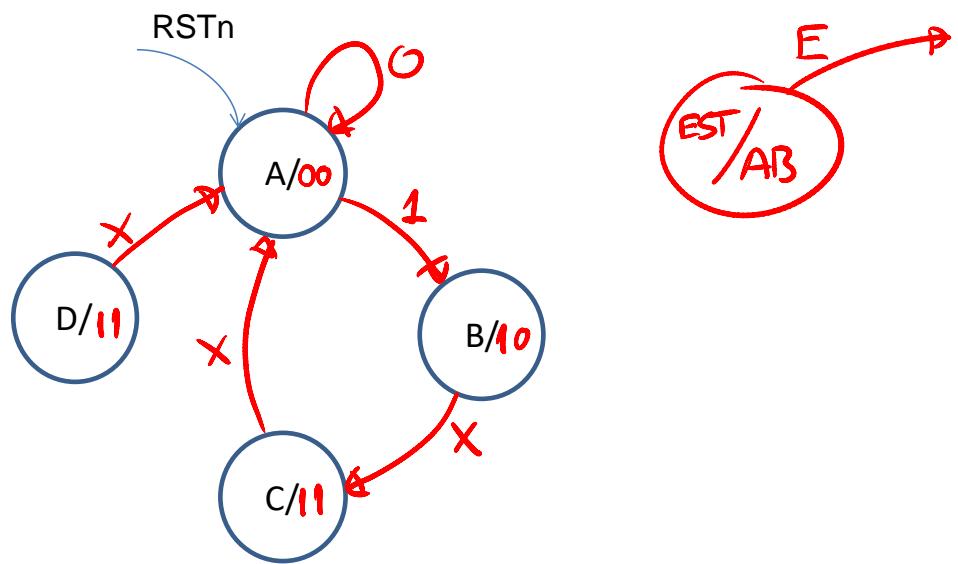
$$D_0 = E \cdot \bar{Q}_1 \cdot \bar{Q}_0$$

$$D_1 = \bar{Q}_1 \cdot Q_0$$

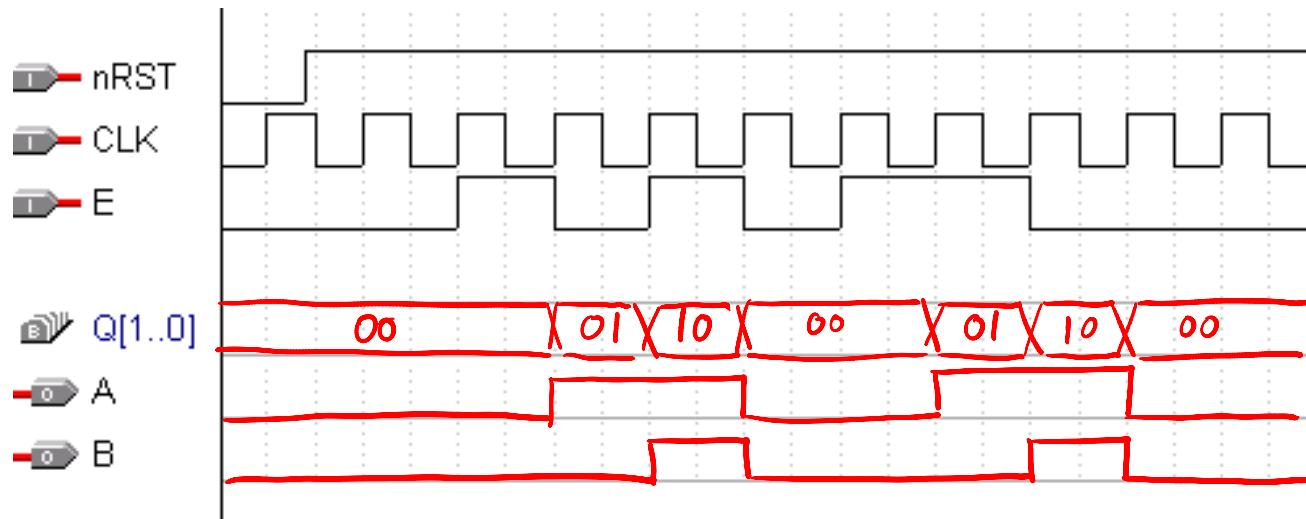
Entrada	Estado actual		Estado futuro		
	E	Q _{1t}	Q _{0t}	D ₁ =Q _{1t+1}	
0	A	0	0	0	0
0	B	0	1	1	0
0	C	1	0	0	0
0	D	1	1	0	0
1	A	0	0	0	1
1	B	0	1	1	0
1	C	1	0	0	0
1	D	1	1	0	0

A
C
A
D
A
C
A

D. Diagrama de estados. Complete el diagrama de estados especificando las salidas correspondientes a cada estado y las transiciones entre los estados.

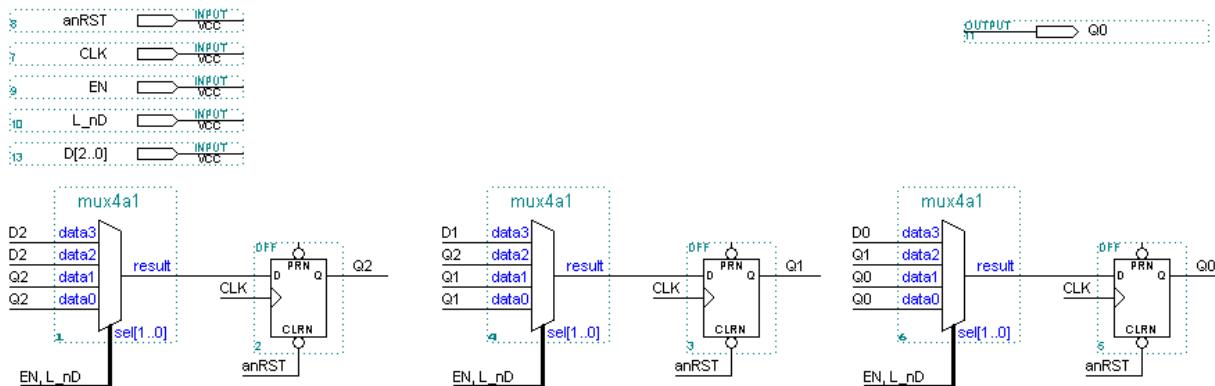


E. Análisis del circuito. El circuito es un secuenciador que produce un determinado patrón de valores en sus salidas A y B cada vez que se activa (a nivel alto) la entrada E. Analice el circuito y complete el siguiente cronograma:



Ejercicio 3

El circuito de la figura representa un registro de desplazamiento de tres bits con habilitación de reloj y carga en paralelo. Los multiplexores de 4 a 1 que contiene funcionan como se indica en la tabla adjunta.

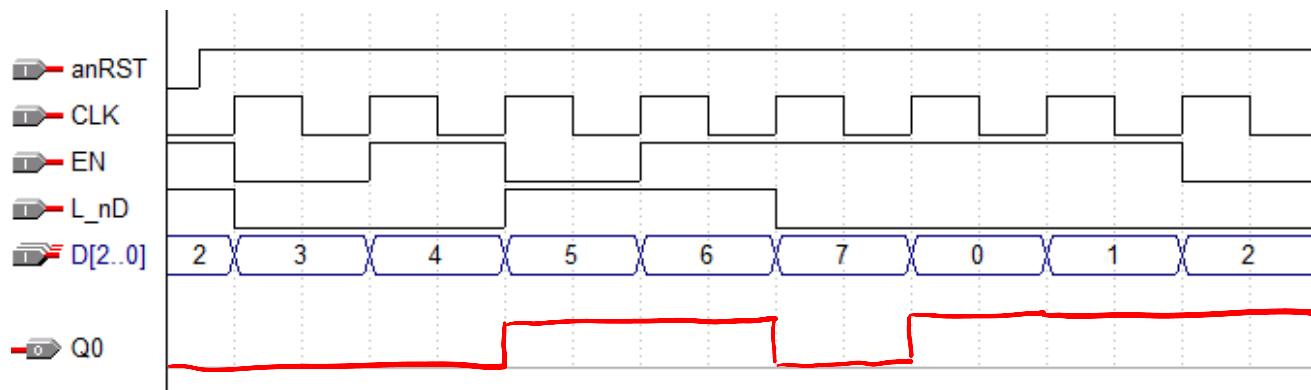


sel[1..0]	result
00	data0
01	data1
10	data2
11	data3

Complete la siguiente tabla, que describe el funcionamiento del registro.

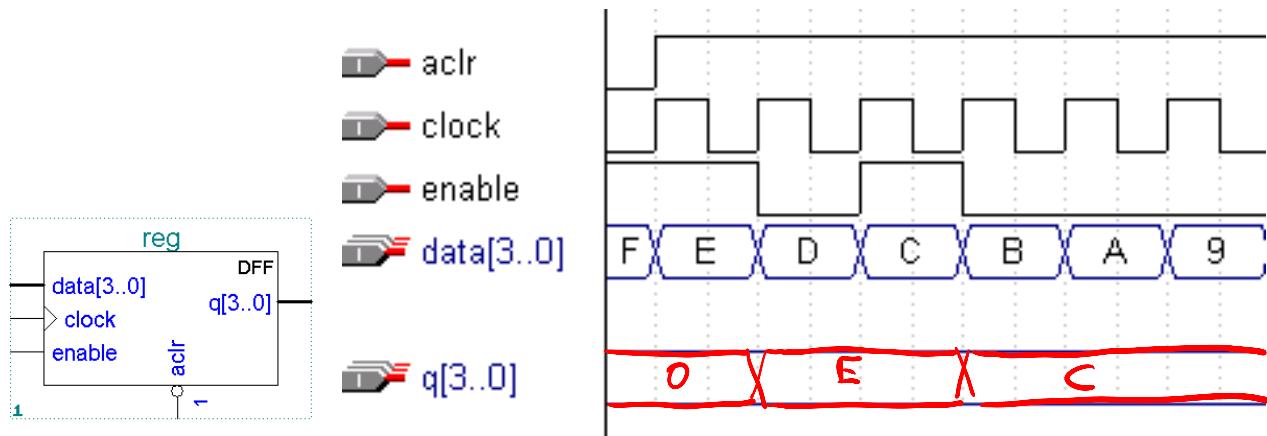
Entradas						Salida	Funcionamiento
anRST	EN	L_nD	CLK	Q ₂	Q ₁	Q ₀	
0	X	X	X	0	0	0	Reset asíncrono
1	0	X	↑	Q _{2(t-1)}	Q _{1(t-1)}	Q _{0(t-1)}	Memoria
1	1	1	↑	D ₂	D ₁	D ₀	Carga
1	1	0	↑	D ₂	Q _{2(t-1)}	Q _{1(t-1)}	Desplazamiento

Rellene el siguiente cronograma:



Ejercicio 4

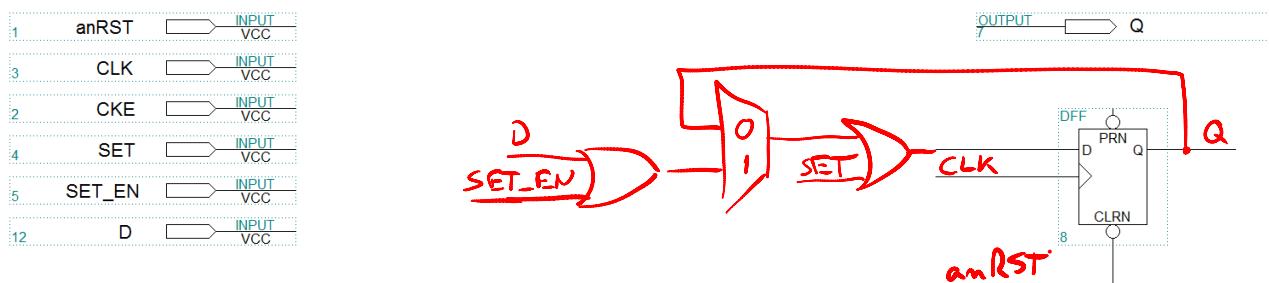
Complete el siguiente cronograma, correspondiente al registro paralelo de 4 bits con habilitación de reloj cuyo símbolo se muestra a continuación. Los valores de la entrada **data[3..0]** se representan en hexadecimal; utilice la misma base para representar los datos de la salida **q[3..0]**.



Ejercicio 5

Empleando la lógica adicional que necesite, dote al flip-flop tipo D de la figura de las siguientes entradas:

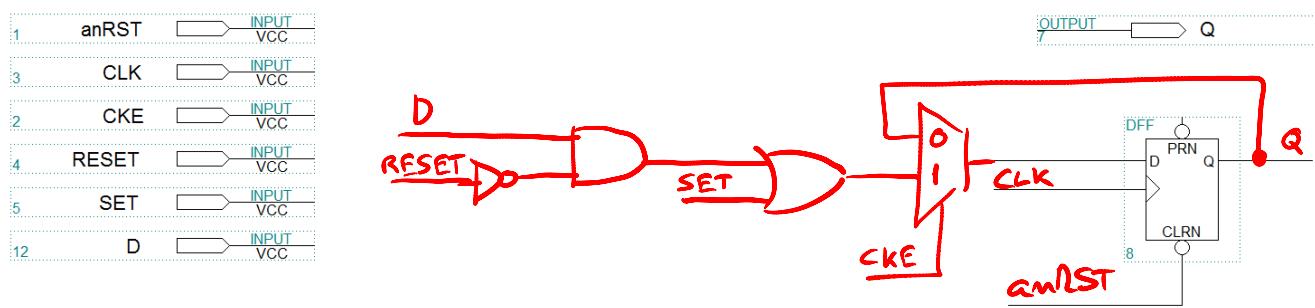
- Entrada de reset asíncrono activo a nivel bajo, **anRST**.
- Entrada de habilitación de reloj, **CKE**, activa a nivel alto.
- Entrada de preset síncrono, **SET_EN**, habilitada por CKE.
- Entrada de preset síncrono, **SET**, que no precise de CKE para actuar.



Ejercicio 6

Empleando la lógica adicional que necesite, dote al flip-flop tipo D de la figura de las siguientes entradas:

- Entrada de reset asíncrono activo a nivel bajo, **anRST**.
- Entrada de habilitación de reloj, **CKE**, activa a nivel alto.
- Entrada de reset síncrono, **RESET**, habilitada por CKE.
- Entrada de preset síncrono, **SET**, también habilitada por CKE y prioritaria sobre RESET.



Esta página se ha dejado intencionadamente en blanco

BLOQUE TEMÁTICO 3**TÍTULO DE LA ACTIVIDAD:****Ejercicios sobre contadores y diseño de autómatas.****CÓDIGO:****BT3_ATGP8**

FECHA:

NOMBRE:

APELLIDOS:

MODALIDAD:

Ejercicios. Aprendizaje colaborativo.

TIPO:

Presencial

DURACIÓN:

100 minutos

CALENDARIO:

Décima Sesión

REQUISITOS:

Contadores. Diseño de autómatas.

CRITERIO DE ÉXITO:

Realizar correctamente los ejercicios entregables

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:

minutos

AUTOEVALUACIÓN:
[entre 0 y 10 puntos]

En el cuestionario de moodle

Actividad BT3_ATGP8: Ejercicios sobre contadores y diseño de autómatas.

Introducción

Esta actividad se realizará en la décima sesión presencial, en grupos G4. Cada grupo deberá realizar los seis ejercicios de que consta la actividad.

Actividad BT3_ATGP8: Ejercicios sobre contadores y diseño de autómatas.

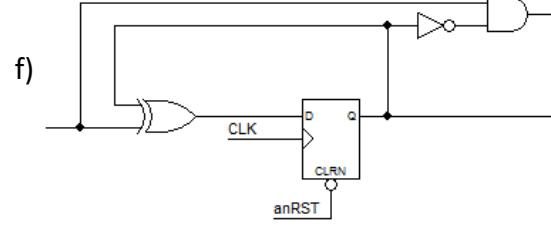
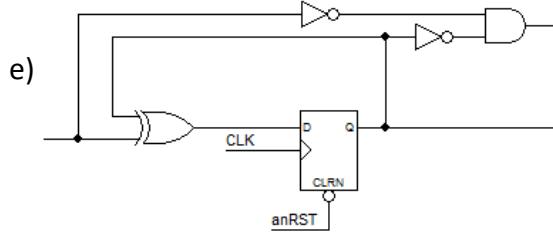
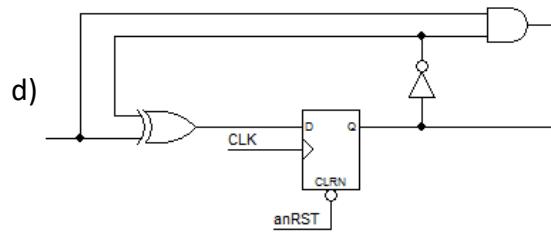
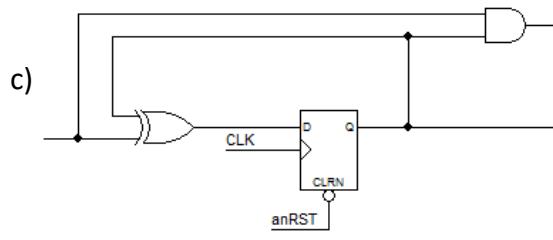
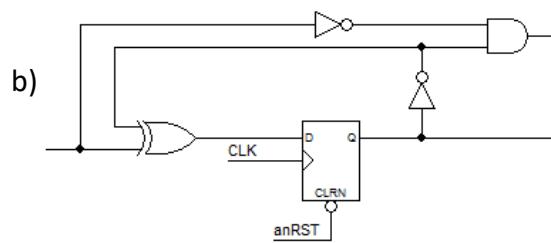
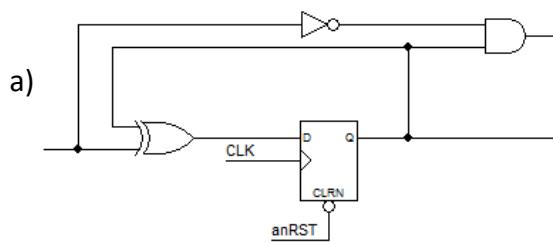
Ejercicio 1

Se desea construir un contador binario descendente de 3 bits con la siguiente interfaz:

- Entrada de reloj, **CLK**.
- Entrada de reset asíncrono, **anRST**, que lleva el contador al estado '000'.
- Salida de estado de cuenta, **Q[2..0]**.
- Salida de fin de cuenta, **FDC**.

1) Dibuje el símbolo del contador

2) Indique cuál de las siguientes células básicas debería utilizar para construir el contador:

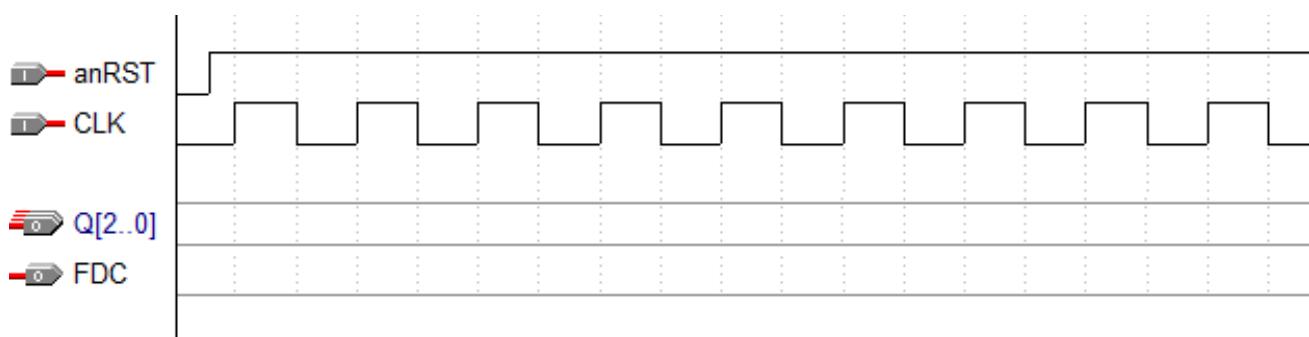


Respuesta:

Actividad BT3_ATGP8: Ejercicios sobre contadores y diseño de autómatas.

- 3) Diseñe el contador empleando células básicas como la que haya seleccionado en el apartado anterior y, si fuera necesario, la lógica adicional que necesite.

- 4) Analice el funcionamiento del contador, rellenando el siguiente cronograma:



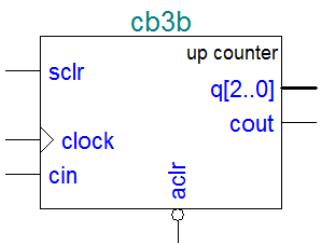
Actividad BT3_ATGP8: Ejercicios sobre contadores y diseño de autómatas.

- 5) Indique qué modificación habría que hacer al contador diseñado en el apartado 3) para añadir una entrada de habilitación de cuenta **EN**.
- 6) Indique qué modificación habría que hacer al contador diseñado en el apartado 3) para que fuese reversible; es decir, para añadirle una entrada que controle el sentido de cuenta (ascendente o descendente).

Actividad BT3_ATGP8: Ejercicios sobre contadores y diseño de autómatas.

Ejercicio 2

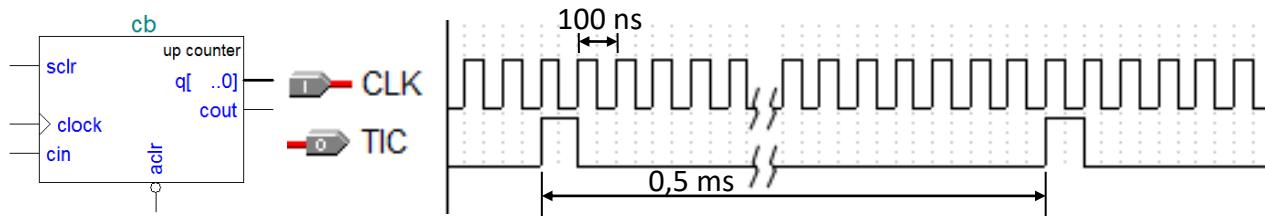
Construya un contador binario módulo 25 utilizando contadores binarios ascendentes de 3 bits con entrada y salida de acarreo y entrada de reset síncrono como el de la figura:



Actividad BT3_ATGP8: Ejercicios sobre contadores y diseño de autómatas.

Ejercicio 3

Se desea construir un divisor de frecuencia que genere en su salida **TIC** un pulso de un ciclo de reloj cada 0,5 ms a partir de un reloj de 10 MHz, tal como se indica en el cronograma adjunto, utilizando un contador binario con entrada y salida de acarreo de cuenta y entrada de reset síncrono como el mostrado a continuación.

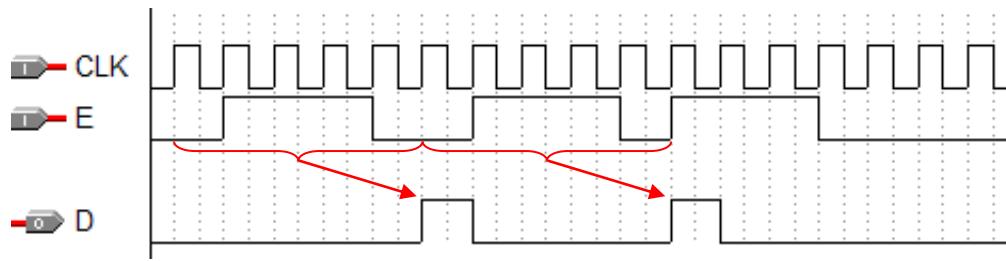


- 1) ¿Cuántos bits debe tener como mínimo el contador para que pueda utilizarse para construir un circuito como el descrito? Razone su respuesta.
- 2) Diseñe el divisor de frecuencia utilizando el contador (especificando el número de bits) y la lógica adicional que considere necesaria.

Actividad BT3_ATGP8: Ejercicios sobre contadores y diseño de autómatas.

Ejercicio 4

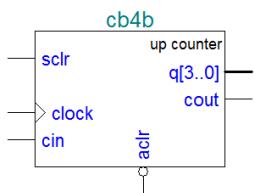
Diseñe el diagrama de estados de un autómata de Moore que active a nivel alto su salida **D** de un bit cuando se presente la secuencia de bits 01110 en su entrada **E**. No deben detectarse secuencias solapadas; es decir, si se presenta en la entrada la secuencia 011101110 la salida **D** debe activarse una única vez tras detectar la primera ocurrencia de la secuencia. El cronograma adjunto ilustra el funcionamiento del autómata.



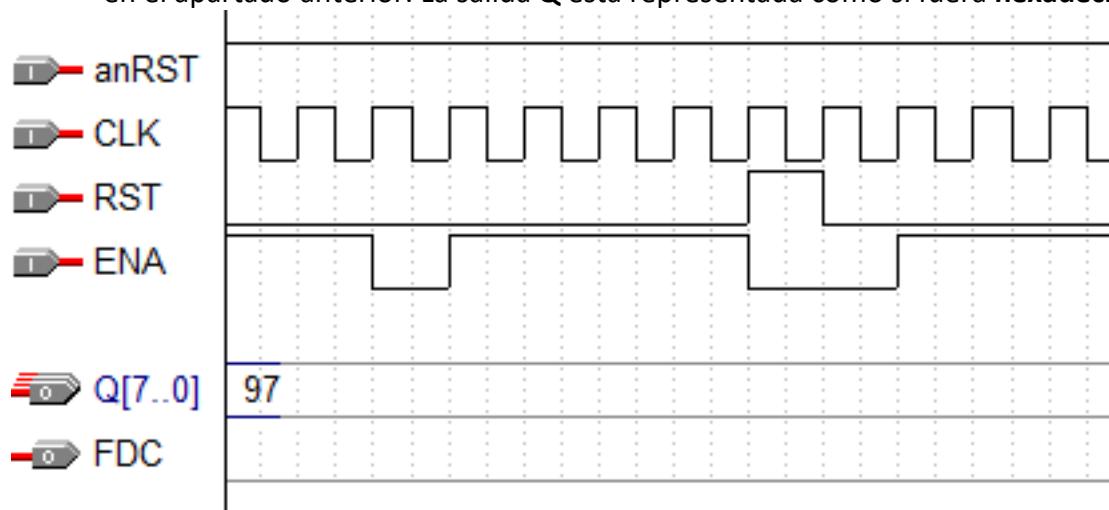
Actividad BT3_ATGP8: Ejercicios sobre contadores y diseño de autómatas.

Ejercicio 5

- 1) Diseñe un contador BCD de 2 dígitos con entradas de reset síncrono (**RST**) y asíncrono (**anRST**, activo a nivel bajo) y habilitación de cuenta (**ENA**, que no afecta al funcionamiento de **RST**) y salidas de estado de cuenta (**Q**, con el número de bits que sean necesarios) y de fin de cuenta (**FDC**, no afectada por **ENA**) utilizando contadores binarios ascendentes de 4 bits con entrada (**cin**) y salida (**cout**) de acarreo de cuenta y entrada de reset síncrono (**sclr**) y asíncrono (**aclr**, activo a nivel bajo) como el de la figura y la lógica adicional que necesite.



- 2) Rellene el siguiente cronograma, correspondiente al funcionamiento del contador descrito en el apartado anterior. La salida **Q** está representada como si fuera **hexadecimal**.

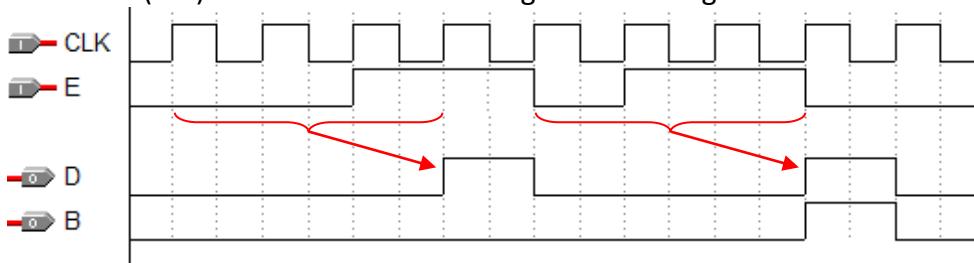


- 3) Indique qué modificación habría que hacer al contador diseñado para que la salida de fin de cuenta **FDC** sólo se active si también está activa la entrada de habilitación **ENA**.

Actividad BT3_ATGP8: Ejercicios sobre contadores y diseño de autómatas.

Ejercicio 6

Diseñe el diagrama de estados de un autómata de Moore que detecte la secuencia 0X1, donde “X” puede ser tanto 0 como 1. El autómata tiene una entrada síncrona de un bit, **E**, por la que recibe los datos, y dos salidas de un bit, **D** y **B**. La salida **D** se debe activar a nivel alto durante un ciclo de reloj cuando se detecte la secuencia buscada, y la salida **B** debe coincidir con el valor del bit central de la secuencia (“X”) cuando **D** se activa. El siguiente cronograma ilustra el funcionamiento del autómata.



BLOQUE TEMÁTICO 3**TÍTULO DE LA ACTIVIDAD:****Realización de ejercicios de diseño de diagrama de estados.****CÓDIGO:****BT3_AINP9**

FECHA:	
--------	--

NOMBRE:		APELLIDOS:	
---------	--	------------	--

MODALIDAD:	Ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	40 minutos
------------	------------------------	-------	---------------	-----------	------------

CALENDARIO:	Décima Sesión	REQUISITOS:	Metodología de diseño de autómatas.
-------------	---------------	-------------	-------------------------------------

CRITERIO DE ÉXITO:	Realización correcta de los 3 ejercicios
--------------------	--

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	Autocorrección
------------------	---------	--	----------------

REALIZACIÓN DE EJERCICIOS SOBRE LA METODOLOGÍA DE DISEÑO DE AUTÓMATAS

PARTE I. Enunciado de los ejercicios

En lo que sigue se presentan algunos enunciados de ejercicios que deberán ser resueltos sin consultar la solución, tratando de poner en práctica los conocimientos sobre la metodología de diseño de diagramas de estados expuesta en clase. Posteriormente se tendrá la oportunidad de contrastar la solución con la que se ofrece en la segunda parte de este documento y detectar posibles errores.

EJERCICIO 1) Obtenga el diagrama de estados de un autómata de Mealy que tiene una entrada (**X**) de un solo bit por la que se reciben bits en serie y una salida (**S**) también de un solo bit. Este autómata generará en su salida la secuencia “11” cada vez que se detecte en la entrada la secuencia “001”, permaneciendo **S** a cero en caso contrario. Indique el estado inicial al que se debe llevar el autómata al activar la señal asíncrona **nINI**.

EJERCICIO 2) Obtenga el diagrama de estados de un autómata de Moore que modele un detector de la secuencia 11011. Dicha secuencia se recibe a través de una entrada (In) por la que llegan los bits en serie, en sincronismo con el reloj. El detector debe poner su salida (Out) a nivel alto durante un pulso de reloj, cada vez que se detecte la secuencia. Indique el estado inicial al que se debe llevar el autómata al activar la señal asíncrona nINI.

Los bits que forman parte de una secuencia, pueden formar parte también de la siguiente, es decir, si se recibe la serie bits ...11011011..., la salida Out debería activarse dos veces, después del quinto y después del octavo bit.

EJERCICIO 3) Obtenga el diagrama de estados de un autómata de Moore que, en función de su entrada de control C (C es síncrona), realiza la cuenta “ $0 \rightarrow 1 \rightarrow 2 \rightarrow 0 \dots$ ” cuando $C = 0$ ó la cuenta “ $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \dots$ ” cuando $C = 1$. En cada flanco activo del reloj la salida del autómata (formada por los dos bits Q_1Q_0) sólo puede incrementarse en una unidad o decrementarse en dos unidades. Considere que el estado inicial al que se lleva el autómata al activar la señal asíncrona nINI es el de cuenta ‘0’.

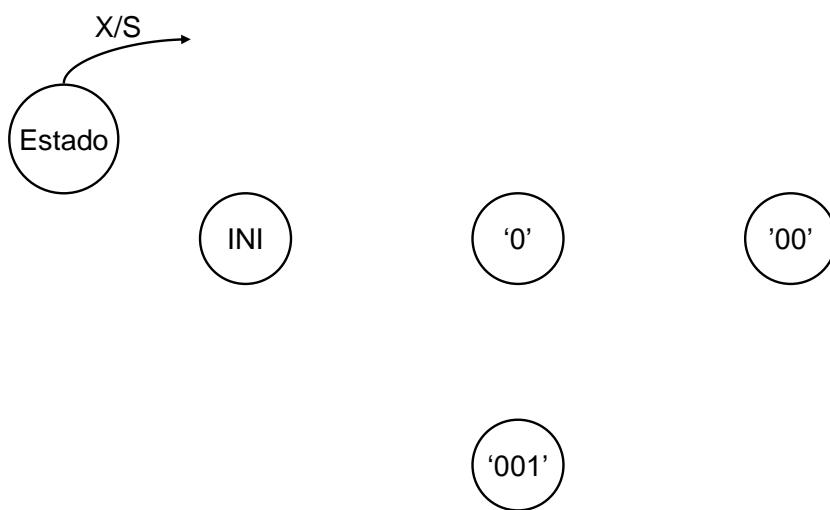
PARTE II. Soluciones de los ejercicios

EJERCICIO 1) Obtenga el diagrama de estados de un autómata de Mealy que tiene 1 entrada (**X**) de un solo bit por la que se reciben bits en serie y una salida (**S**) también de un solo bit. Este autómata generará en su salida la secuencia “11” cada vez que se detecte en la entrada la secuencia “001”, permaneciendo **S** a cero en caso contrario. Indique el estado inicial al que se debe llevar el autómata al activar la señal asíncrona nINI.

Previsión del número de estados

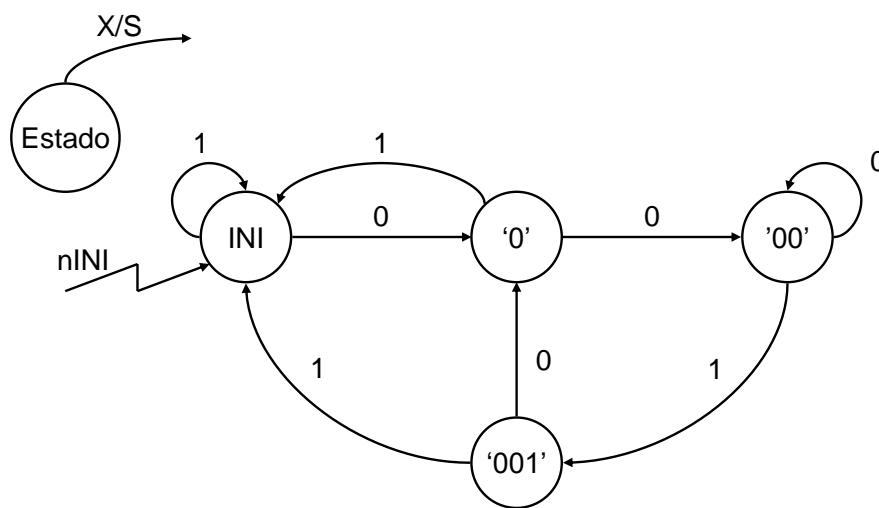
- Estado INI: espera a que por X entre el primer ‘0’.
- Estado ‘0’: Ha llegado el primer 0.
- Estado ‘00’: Ha llegado el segundo 0 consecutivo.
- Estado ‘001’: Ha llegado el 1 después de dos ceros consecutivos.

Nota: Hay que contemplar que, al diseñar el diagrama de estados del autómata, es posible que haga falta o que sobre algún estado.

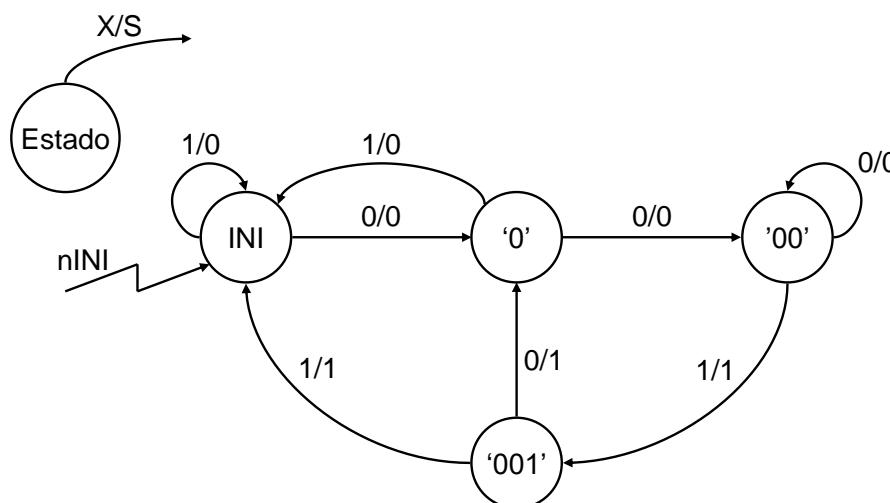


Transiciones

- Estado ‘INI’
 - Si $X = '1'$, el bit entrante no pertenece a la secuencia y, por tanto, el autómata seguirá en ese estado.
 - Si $X = '0'$, el bit entrante si pertenece a la secuencia y el autómata pasa al estado ‘0’.
- Estado ‘0’
 - Si $X = '1'$, el bit entrante no pertenece a la secuencia y, por tanto, el autómata vuelve al estado ‘INI’.
 - Si $X = '0'$, el bit entrante si pertenece a la secuencia y el autómata pasa al estado ‘00’.
- Estado ‘00’
 - Si $X = '1'$, el bit entrante si pertenece a la secuencia y, por tanto, el autómata evoluciona al estado ‘001’.
 - Si $X = '0'$, el bit entrante si pertenece a la secuencia pero como segundo cero consecutivo y el autómata permanece en el estado ‘00’.
- Estado ‘001’
 - Si $X = '1'$, el bit entrante no pertenece a la secuencia y, por tanto, el autómata evoluciona al estado ‘INI’.
 - Si $X = '0'$, el bit entrante si pertenece a la secuencia (es el primer bit de la secuencia) y el autómata pasa al estado ‘0’.

**Salida**

- Estados 'INI' y '0'
 - $S = '0'$, no ha ocurrido la secuencia buscada.
- Estado '00'
 - Si $X = '1'$, ha ocurrido la secuencia y $S = '1'$ durante un ciclo de reloj.
 - Si $X = '0'$, $S = '0'$, no ha ocurrido la secuencia buscada.
- Estado '001'
 - $S = '1'$ independientemente del valor de X pues la salida debe estar 2 ciclos de reloj consecutivos a nivel alto.

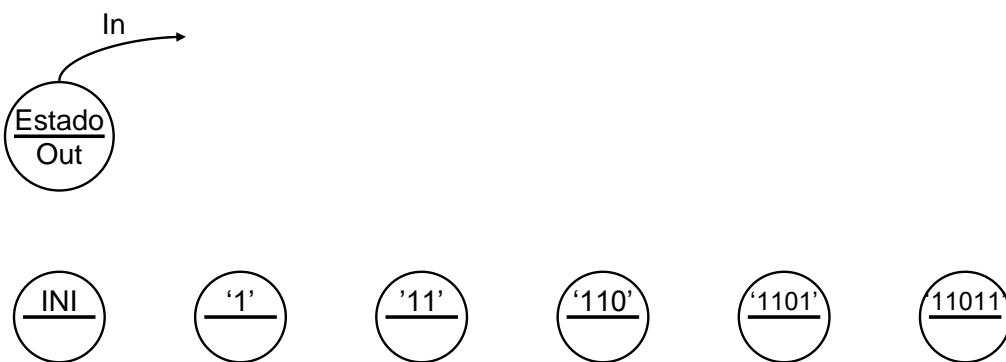


EJERCICIO 2) Obtenga el diagrama de estados de un autómata de Moore que modele un detector de la secuencia 11011. Dicha secuencia se recibe a través de una entrada (In) por la que llegan los bits en serie, en sincronismo con el reloj. El detector debe poner su salida (Out) a nivel alto durante un pulso de reloj, cada vez que se detecte la secuencia. Indique el estado inicial al que se debe llevar el autómata al activar la señal asíncrona nINI.

Los bits que forman parte de una secuencia, pueden formar parte también de la siguiente, es decir, si se recibe la serie bits ...11011011..., la salida Out debería activarse dos veces, después del quinto y después del octavo bit.

Previsión del número de estados

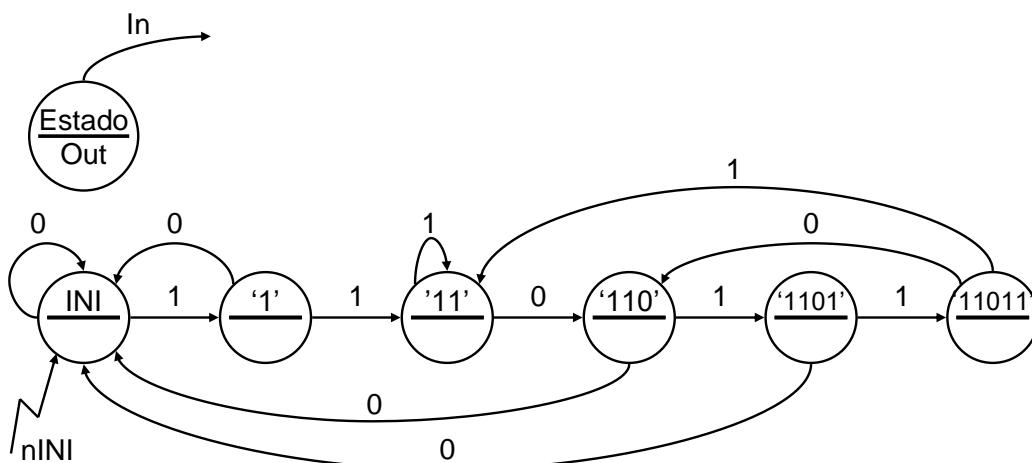
- Estado INI: espera a que por In entre el primer '1'.
- Estado '1': Ha llegado el primer '1'.
- Estado '11': Ha llegado el segundo '1' consecutivo.
- Estado '110': Ha llegado un '0' después de dos unos consecutivos.
- ...
- Estado '11011': Ha llegado la secuencia que se desea detectar.



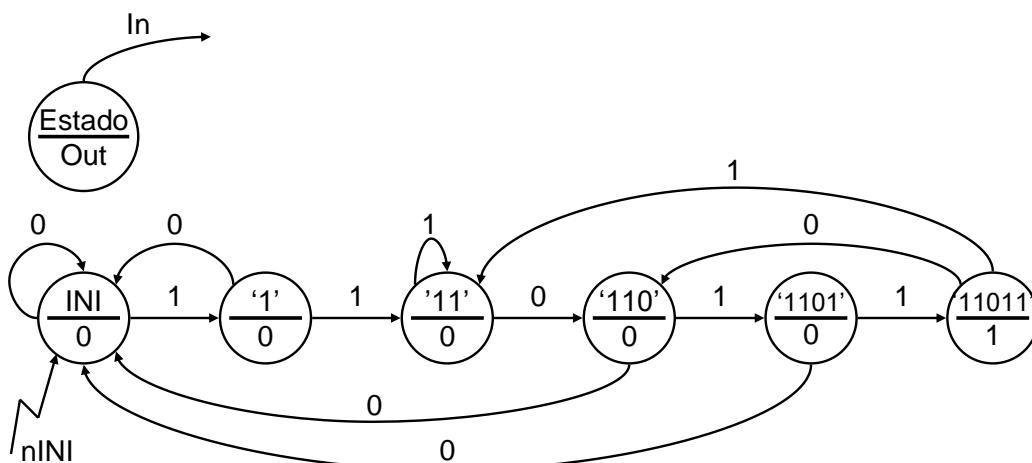
Transiciones

- Estado 'INI'
 - Si IN = '0', el bit entrante no pertenece a la secuencia y, por tanto, el autómata seguirá en ese estado.
 - Si IN = '1', el bit entrante si pertenece a la secuencia y el autómata pasa al estado '1'.
- Estado '1'
 - Si IN = '0', el bit entrante no pertenece a la secuencia y, por tanto, el autómata vuelve al estado 'INI'.
 - Si IN = '1', el bit entrante si pertenece a la secuencia y el autómata pasa al estado '11'.
- Estado '11'
 - Si IN = '0', el bit entrante si pertenece a la secuencia y, por tanto, el autómata evoluciona al estado '110'.
 - Si IN = '1', el bit entrante si pertenece a la secuencia pero como segundo uno consecutivo y, por tanto, permanece en el mismo estado ('11').
- Estado '110'
 - Si IN = '0', el bit entrante no pertenece a la secuencia y, por tanto, el autómata evoluciona al estado 'INI'.
 - Si IN = '1', el bit entrante si pertenece a la secuencia y el autómata pasa al estado '1101'.
- Estado '1101'

- Si $IN = '0'$, el bit entrante no pertenece a la secuencia y, por tanto, el autómata evoluciona al estado ‘INI’.
- Si $IN = '1'$, el bit entrante si pertenece a la secuencia y el autómata pasa al estado ‘11011’.
- Estado ‘11011’
 - Si $IN = '0'$, el bit entrante **pertenece a parte de la secuencia** (aprovechando los dos últimos unos llegados y el ‘0’ presente en la entrada IN se tiene la parte ‘110’ de la secuencia buscada) y, por tanto, el autómata evoluciona al estado ‘110’.
 - Si $IN = '1'$, el bit entrante **pertenece a parte de la secuencia** (aprovechando el ultimo ‘1’ llegado y el ‘1’ presente en la entrada In se tiene la parte ‘11’ de la secuencia buscada) y el autómata pasa al estado ‘11’.

**Salida**

- Estados ‘INI’, ‘1’, ‘11’, ‘110’ y ‘1101’
 - No ha ocurrido la secuencia buscada.
- Estado ‘11011’
 - Si ha ocurrido la secuencia y $S = '1'$ durante un ciclo de reloj.

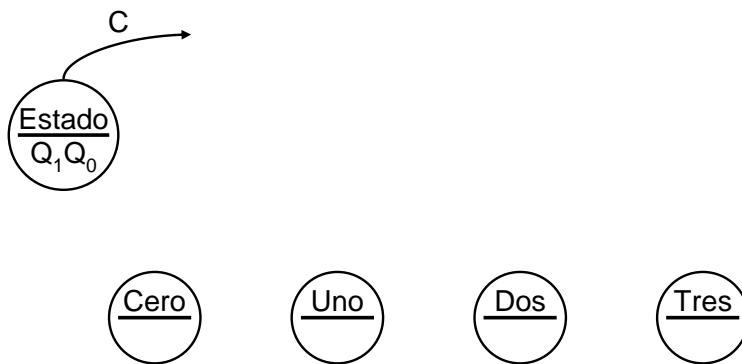


EJERCICIO 3) Obtenga el diagrama de estados de un autómata de Moore que, en función de su entrada de control C (C es síncrona), realiza la cuenta “ $0 \rightarrow 1 \rightarrow 2 \rightarrow 0 \dots$ ” cuando $C = 0$ ó la cuenta “ $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \dots$ ” cuando $C = 1$. En cada flanco activo del reloj la salida del autómata (formada por los dos bits Q_1Q_0) sólo puede incrementarse en una unidad o decrementarse en dos unidades. Considere que el estado inicial al que se lleva el autómata al activar la señal asíncrona nINI es el de cuenta ‘0’.

Previsión del número de estados

El autómata es un contador. En los contadores, cada cuenta tiene asociado un estado:

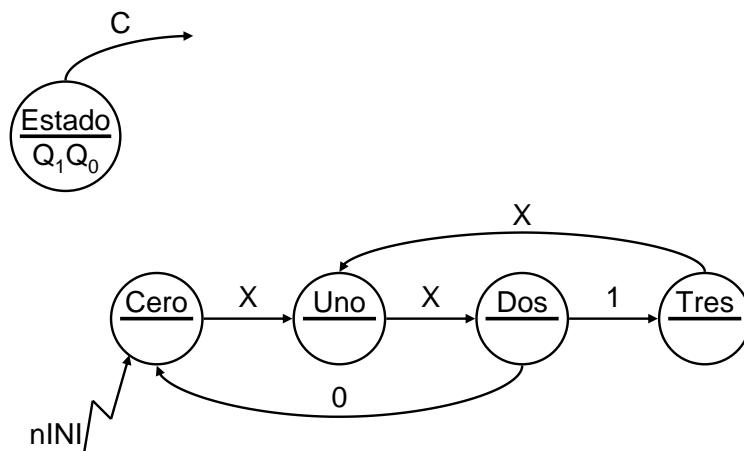
- Estado ‘Cero’: el valor de cuenta es 0.
- Estado ‘Uno’: el valor de cuenta es 1.
- Estado ‘Dos’: el valor de cuenta es 2.
- Estado ‘Tres’: el valor de cuenta es 3.



Transiciones

Las cuentas que debe realizar el autómata son: “ $0 \rightarrow 1 \rightarrow 2 \rightarrow 0 \dots$ ” cuando $C = 0$ ó la cuenta “ $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \dots$ ” cuando $C = 1$.

- Estado ‘Cero’
 - Si $C = '0'$, el siguiente estado es ‘Uno’ según está descrita la cuenta para $C = '0'$.
 - Si $C = '1'$, el siguiente estado es ‘Uno’ pues, aunque a la cuenta = 0 no pertenece al caso de $C = '1'$, el contador sólo puede incrementarse en una unidad (no puede decrementarse en 2 pues la cuenta quedaría en negativo).
- Estado ‘Uno’
 - Si $C = '0'$, el siguiente estado es ‘Dos’ según está descrita la cuenta para $C = '0'$.
 - Si $C = '1'$, el siguiente estado es ‘Dos’ según está descrita la cuenta para $C = '1'$.
- Estado ‘Dos’
 - Si $C = '0'$, el siguiente estado es ‘Cero’ según está descrita la cuenta para $C = '0'$.
 - Si $C = '1'$, el siguiente estado es ‘Tres’ según está descrita la cuenta para $C = '1'$.
- Estado ‘Tres’
 - Si $C = '0'$, el siguiente estado es ‘Uno’ pues, aunque la cuenta = 3 no pertenece al caso de $C = '0'$, el contador sólo puede decrementarse en dos unidades (no puede incrementarse en 1 pues se sobrepasaría la cuenta más alta, que es 3).
 - Si $C = '1'$, el siguiente estado es ‘Uno’ según está descrita la cuenta para $C = '1'$.

**Salida**

Al ser un autómata de Moore, la salida depende del estado y, en este caso, la salida es la propia cuenta

- Estado 'Cero': el valor de cuenta es 0.
- Estado 'Uno': el valor de cuenta es 1.
- Estado 'Dos': el valor de cuenta es 2.
- Estado 'Tres': el valor de cuenta es 3.

