Laboratorio de Estructura y Tecnología de Computadores

Julián García Donaire
Vicente González Ruiz

Depto de Arquitectura de Computadores y Electrónica

21 de abril de 1999

Índice General

1 Características y parámetros de los CIs **10**
- 1.1 Tensión de alimentación: +5V continuos . . . 10
- 1.2 Niveles lógicos . . . 10
  - 1.2.1 TTL . . . 11
  - 1.2.2 CMOS . . . 11
- 1.3 Inmunidad al ruido: margen de ruido . . . 11
- 1.4 Retardo de propagación . . . 12
- 1.5 Fan-out . . . 13
  - 1.5.1 TTL . . . 13
  - 1.5.2 CMOS . . . 14
- 1.6 Disipación de potencia . . . 14
  - 1.6.1 TTL . . . 14
  - 1.6.2 CMOS . . . 15
- 1.7 Producto velocidad-potencia . . . 15
- 1.8 Prácticas . . . 15
  - 1.8.1 Determinación del tiempo de propagación . . . 15

2 Biestables **17**
- 2.1 El cerrojo SR . . . 17
- 2.2 El cerrojo SR con entrada de habilitación . . . 17
- 2.3 El flip-flop SR . . . 19
- 2.4 El flip-flop JK disparado por flanco . . . 21
- 2.5 Características de operación . . . 21
  - 2.5.1 Retardo de propagación: t_{PLH} y t_{PHL} . . . 21
  - 2.5.2 Tiempo de set-up (establecimiento): t_S . . . 21
  - 2.5.3 Tiempo de hold (mantenimiento): t_H . . . 22
  - 2.5.4 Anchura de los impulsos: t_W . . . 22
  - 2.5.5 Frecuencia máxima de funcionamiento: f_{max} . . . 22
  - 2.5.6 Disipación de potencia: P . . . 22
- 2.6 Prácticas . . . 22
  - 2.6.1 El biestable SR con puertas NOR . . . 22
  - 2.6.2 El biestable SR con puertas NAND . . . 23
  - 2.6.3 El cerrojo D obtenido a partir del cerrojo SR . . . 24
  - 2.6.4 El cerrojo JK . . . 25
  - 2.6.5 El cerrojo D obtenido a partir del cerrojo JK . . . 26
  - 2.6.6 El cerrojo SR con entrada de habilitación . . . 27
  - 2.6.7 El cerrojo JK con entrada de habilitación . . . 29

2.6.8 El cerrojo SR con entradas de habilitación, PRESET y CLEAR . . . 30
2.6.9 El cerrojo JK con entradas de habilitación, PRESET y CLEAR . . . 30
2.6.10 El flip-flop SR disparado por flanco positivo . . . 34
2.6.11 El flip-flop SR disparado por flanco negativo . . . 35
2.6.12 El flip-flop JK disparado por flanco positivo . . . 37
2.6.13 El flip-flop SR maestro-esclavo . . . 37
2.6.14 El flip-flop JK maestro-esclavo . . . 39

3 Contadores **40**
3.1 Contador binario asíncrono ascendente . . . 40
3.2 Contador BCD asíncrono ascendente . . . 41
3.3 Contador binario síncrono ascendente . . . 42
3.4 Contador binario síncrono ascendente/descendente . . . 43
3.5 Contador BCD síncrono ascendente . . . 43
3.6 El contador en anillo . . . 45
3.7 El contador Johnson . . . 46
3.8 Prácticas . . . 46
3.8.1 Contador binario asíncrono ascendente . . . 46
3.8.2 Contador binario asíncrono descendente . . . 47
3.8.3 Contador asíncrono ascendente BCD . . . 49
3.8.4 Contador síncrono binario de 4 bits . . . 49
3.8.5 Contador síncrono ascendente/descendente . . . 51
3.8.6 Contador síncrono BCD ascendente . . . 53
3.8.7 El contador en anillo . . . 53
3.8.8 El contador Johnson . . . 53

4 Diseño de circuitos secuenciales **56**
4.1 Diseño de un contador BCD síncrono . . . 56
4.1.1 Diagrama de estados . . . 56
4.1.2 Tabla de transiciones . . . 57
4.1.3 Minimización de la lógica combinacional . . . 58
4.1.4 Implementación . . . 58
4.2 Diseño de un contador síncrono BCD ascendente/descendente . . . 58
4.2.1 Diagrama de estados . . . 58
4.2.2 Tabla de transiciones . . . 60
4.2.3 Minimización de la lógica combinacional . . . 60
4.2.4 Implementación . . . 60
4.3 Detector de sercuencias de tres unos consecutivos . . . 63
4.3.1 Digrama de estados . . . 63
4.3.2 Tabla de transiciones . . . 63
4.3.3 Minimización de la lógica combinacional . . . 64
4.3.4 Implementación . . . 64
4.4 Detector de sercuencias de tres unos consecutivos (Mealy) . . . 64
4.4.1 Diagrama de estados . . . 64
4.4.2 Tabla de transiciones . . . 66
4.4.3 Minimización de la lógica combinacional . . . 66
4.4.4 Implementación . . . 66
4.5 Prácticas . . . 68

4.5.1 Contador de estados impares 68
4.5.2 Generador de secuencia $3, 4, 5, 6, 7$ 68
4.5.3 Analizador de entradas 68
4.5.4 Detector de secuencias de ceros solapadas 68
4.5.5 Detector de secuencias de ceros no solapadas 69
4.5.6 Detector de secuencias de pares de bits alternos 69
4.5.7 Detector de eventos 69
4.5.8 Codificador unario 70
4.5.9 Decodificador unario 70

Índice de Figuras

1.1 Niveles lógicos. 11
1.2 Niveles lógicos TTL. 11
1.3 Niveles lógicos CMOS. 12
1.4 Retardos de propagación. 13
1.5 Tiempos de propagación asociados a un inversor. 15
1.6 Montaje del inversor realimentado. 16

2.1 Estructura interna de un cerrojo SR fabricado con puertas NOR. 17
2.2 Símbolo lógico de un cerrojo SR. 18
2.3 Estructura interna de un cerrojo SR (NOR) con entrada de habilitación. 18
2.4 Símbolo lógico de un cerrojo SR con entrada de habilitación. . . 18
2.5 Estructura de un flip-flop SR disparado por flanco (positivo). . . 19
2.6 Símbolo lógico de un flip-flip SR disparado por flanco positivo. . 19
2.7 Detector de transiciones positivas. 20
2.8 Estructura de un flip-flop JK disparado por flanco (positivo). . . 20
2.9 Símbolo lógico de un flip-flop JK disparado por flanco positivo. . 20
2.10 Tiempos de operación de un flip-flop D. 21
2.11 Montaje del cerrojo SR con puertas NOR y símbolo lógico. . . . 22
2.12 Cronograma del cerrojo RS usando puertas NOR. 23
2.13 Montaje del cerrojo RS con puertas NAND y símbolo lógico. . . 23
2.14 Cronograma del cerrojo SR usando puertas NAND. 24
2.15 Montaje del cerrojo D a partir de un SR y símbolo lógico. 25
2.16 Cronograma del cerrojo D usando SR. 25
2.17 Montaje del cerrojo JK y símbolo lógico. 25
2.18 Cronograma del cerrojo JK. 26
2.19 Montaje del cerrojo D a partir de un JK y símbolo lógico. 26
2.20 Cronograma del cerrojo D usando JK. 27
2.21 Montaje del cerrojo SR con entrada de habilitación y símbolo lógico. 27
2.22 Cronograma del cerrojo SR con entrada de habilitación. 28
2.23 Montaje del cerrojo JK con entrada de habilitación y símbolo lógico. 29
2.24 Cronograma del cerrojo JK con entrada de habilitación. 29
2.25 Montaje del cerrojo SR con entradas PRESET y CLEAR y símbolo lógico. 30
2.26 Cronograma del cerrojo SR con entradas habilitación, PRESET y CLEAR. 31

2.27 Montaje del cerrojo JK con entradas de habilitación, $\overline{\text{PRESET}}$ y $\overline{\text{CLEAR}}$ y símbolo lógico. 32
2.28 Cronograma del cerrojo JK con entrada de habilitación, $\overline{\text{PRESET}}$ y $\overline{\text{CLEAR}}$. 32
2.29 Montaje del flip-flop SR disparado por flanco positivo y su símbolo lógico. 35
2.30 Cronograma del flip-flop SR activo por flanco positivo. 35
2.31 Montaje del flip-flop SR disparado por flanco negativo y símbolo lógico. 36
2.32 Cronograma del flip-flop SR disparado por flanco negativo. . . . 36
2.33 Montaje del flip-flop JK disparado por flanco positivo y símbolo lógico. 37
2.34 Cronograma del flip-flop JK disparado por flanco positivo. 37
2.35 Montaje del flip-flop SR maestro-esclavo y símbolo lógico. 38
2.36 Cronograma del flip-flop SR maestro-esclavo. 38
2.37 Montaje del flip-flop JK maestro-esclavo y símbolo lógico. 39
2.38 Cronograma del flip-flop JK maestro-esclavo. 39

3.1 Esquema lógico de un contador binario asíncrono de tres bits. . . 40
3.2 Cronograma real de un contador binario asíncrono de tres bits. . 41
3.3 Esquema lógico de un contador BCD asíncrono. 41
3.4 Cronograma real de un contador BCD asíncrono. 41
3.5 Esquema de un contador binario síncrono de 4 bits. 42
3.6 Cronograma real de un contador binario síncrono de 4 bits. . . . 42
3.7 Esquema de un contador binario síncrono ascendente/descendente de 3 bits. 44
3.8 Esquema de un contador en anillo de 4 bits. 45
3.9 Esquema de un contador en Johnson de 4 bits. 46
3.10 Esquema lógico de un contador binario asíncrono ascendente de 4 bits. 47
3.11 Cronograma de un contador binario asíncrono ascendente de 4 bits. 47
3.12 Esquema lógico de un contador asíncrono binario descendente de 4 bits. 47
3.13 Cronograma de un contador asíncrono binario descendente de 4 bits. 48
3.14 Esquema lógico de un contador asíncrono BCD ascendente. . . . 49
3.15 Cronograma de un contador asíncrono BCD ascendente. 50
3.16 Cronograma de un contador síncrono ascendente de 4 bits. 51
3.17 Esquema lógico de un contador UP/DOWN síncrono de 3 bits. . 52
3.18 Cronograma de un contador UP/DOWN sícrono de 3 bits. 52
3.19 Cronograma de un contador síncrono BCD ascendente. 53
3.20 Esquema de un contador en anillo de 4 bits. 54
3.21 Cronograma de un contador en anillo de 4 bits. 54
3.22 Esquema de un contador Johnson de 4 bits. 55
3.23 Cronograma de un contador Johnson de 4 bits. 55

4.1 Esquema genérico de un circuito secuencial. 56
4.2 Diagrama de estados de un contador BCD. 57
4.3 Mapas de Karnaugh para la minimización de un contador BCD síncrono ascendente. 58

4.4 Estructura interna del contador síncrono BCD ascendente. 59
4.5 Diagrama de estados de un contador síncrono BCD ascendente/descendente. 59
4.6 Mapas de Karnaugh para la minimización de un contador BCD síncrono ascendente/descendente. 61
4.7 Estructura interna del contador síncrono BCD ascendente/descendente. 62
4.8 Diagrama de estados de Moore del detector. 63
4.9 Mapas de Karnaugh el detector de secuencias (Moore) 65
4.10 Esquema lógico del detector de secuencias (Moore). 65
4.11 Diagrama de estados de Mealy del detector. 66
4.12 Mapas de Karnaugh el detector de secuencias (Mealy) 67
4.13 Esquema lógico del detector de secuencias (Mealy). 68
4.14 Diagrama de flujo de un analizador de entradas. 69

Índice de Tablas

2.1 Tabla de verdad de un cerrojo SR. 18
2.2 Tabla de verdad de un cerrojo SR con entrada de habilitación. . 18
2.3 Tabla de verdad de un flip-flop SR disparado por flanco positivo. ** sifnifica: nunca 00. 19
2.4 Tabla de verdad de un flip-flop JK disparado por flanco positivo. 20
2.5 Tabla de verdad para el biestable SR con puertas NOR. 23
2.6 Tabla de verdad para el cerrojo SR con puertas NAND. 24
2.7 Tabla de verdad del cerrojo D usando SR. 24
2.8 Tabla de verdad del cerrojo JK. 26
2.9 Tabla de verdad del cerrojo D usando JK. 27
2.10 Tabla de verdad para el cerrojo SR con entrada de habilitación. . 28
2.11 Tabla de verdad del cerrojo JK con entrada de habilitación. . . . 30
2.12 Tabla de verdad del cerrojo SR con entradas de habilitación, PRESET y CLEAR. 31
2.13 Tabla de verdad del cerrojo JK con entradas de habilitación, $\overline{\text{PRESET}}$ y $\overline{\text{CLEAR}}$. 33
2.14 Tabla de verdad del flip-flop SR activo por flanco positivo. 34
2.15 Tabla de verdad del flip-flop SR disparado por flanco negativo. . 36
2.16 Tabla de verdad del flip-flop JK disparado por flanco positivo. . . 37
2.17 Tabla de verdad del flip-flop SR maestro-esclavo. 38
2.18 Tabla de verdad del flip-flop JK maestro-esclavo. 39

3.1 Tabla de estados de un contador BCD. 44
3.2 Tabla de estados de un contador en anillo de 4 bits. 45
3.3 Tabla de estados y lógica de decodificación de un contador Johnson de 4 bits. 46
3.4 Tabla de estados de un contador asíncrono binario ascendente de 4 bits. 48
3.5 Tabla de estados del contador asíncrono binario descendente de 4 bits. 49
3.6 Tabla de estados de un contador asíncrono BCD ascendente. . . . 50
3.7 Tabla de estados de un contador síncrono ascendente de 4 bits. . 50
3.8 Tabla de verdad para el contador UP/DOWN síncrono de 3 bits. 52
3.9 Tabla de estados de un contador síncrono BCD ascendente. . . . 53
3.10 Tabla de estados de un contador en anillo de 4 bits. 54
3.11 Tabla de estados de un contador Johnson de 4 bits. 55

4.1 Tabla de transiciones del contador síncrono BCD. 57

4.2 Tabla de verdad del flip-flop JK, en función del estado actual y el estado siguiente. 57
4.3 Tabla de transiciones del contador síncrono BCD ascendente/descendente. 60
4.4 Tabla de verdad del flip-flop T, en función del estado actual y el estado siguiente. 61
4.5 Tabla de transiciones del detector de secuencias (Moore). 63
4.6 Asignación de estados para el detector de secuencias (Moore). . . 64
4.7 Tabla de transiciones del detector de secuencias (Moore). 64
4.8 Tabla de transiciones del detector de secuencias (Mealy). 66
4.9 Asignación de estados para el detector de secuencias (Mealy). . . 66
4.10 Tabla de transiciones del detector de secuencias (Mealy). 67

Acrónimos

CC	Continous Current	corriente continua
CMOS	Complementary Metal-Oxide Semiconductor	semiconductor de metal-oxido complementario
DC	Direct Current	corriente directa
GND	GrouND	tierra
IC	Integrated Circuit	circuito integrado (CI)
TTL	Transistor-Transistor Logic	lógica transistor-transistor
Vcc	Voltage CC	Voltaje de corriente contínua
LSb	Low Significant bit	bit menos significativo
MSb	Most Significant bit	bit más significativo
LSB	Low Significant Byte	byte menos significativo
MSB	Most Significant Byte	byte más significativo
BCD	Binary Coded Decimal	

Capítulo 1

Características y parámetros de los CIs

En este capítulo vamos a estudiar las principales características operacionales y parámetros básicos de dos familias lógicas muy utilizadas: TTL y CMOS.

1.1 Tensión de alimentación: +5V continuos

El valor nominal de la tensión de alimentación continua DC para TTL y CMOS es típicamente +5V. Normalmente, las conexiones a la tensión de alimentación y a tierra (denotadas por Vcc y GND respectivamente) se omiten de los diagramas lógicos para simplificarlos. Esto no quiere decir que internamente, dichas señales lleguen a todas las puertas integradas en un CI.

1.2 Niveles lógicos

Los CIs digitales trabajan con dos márgenes de tensiones (suficientemente separadas) que codifican un dígito de código binario: 1 o 0. Estas tensiones reciben el nombre de niveles lógicos y como ya hemos dicho, tanto el nivel lógico alto (1) como el bajo (0) son un intervalo de tensiones[1].

La figura 1.1 ilustra el rango general de variación para los niveles alto y bajo. El límite $V_{H(max)}$ representa el máximo valor permitido para el nivel alto y el límite $V_{H(min)}$ representa el mínimo valor que puede tomar el nivel alto. Idénticas definiciones existen para el nivel mínimo ($V_{L(max)}$ y $V_{L(min)}$). El rango de tensiones entre $V_{L(max)}$ y $V_{H(min)}$ representa un intervalo de incertidumbre. Una tensión dentro de dicho rango podría ser interpretada por el elemento lógico que la utiliza, como un nivel alto o como un nivel bajo, pero nunca se podría determinar con seguridad cual será la interpretación a priori.

Por último diremos que los niveles lógicos son en general distintos sin estamos considerando la entrada o la salida del elemento lógico.

[1]La asignación es de esta forma si utilizamos el convenio establecido en la *lógica positiva*: el bit de código 1 representa el nivel más alto (high) y el bit de código 0 representa al nivel más bajo (low). El otro convenio alternativo se llama *lógica negativa*.

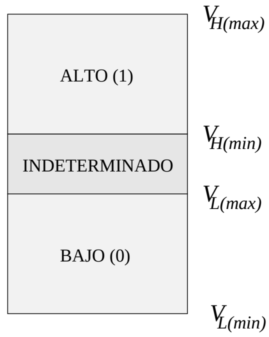


Figura 1.1: Niveles lógicos.

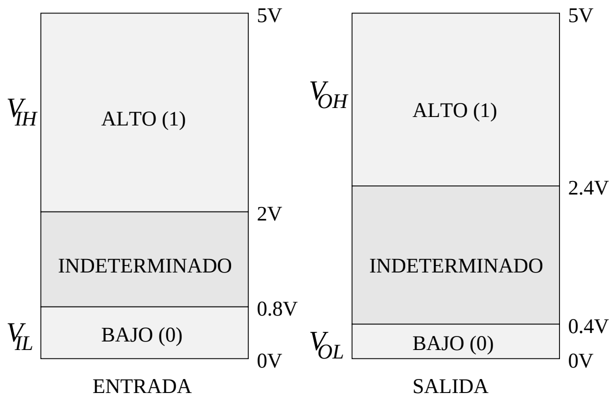


Figura 1.2: Niveles lógicos TTL.

1.2.1 TTL

En la entrada de los circuitos TTL, el nivel lógico bajo puede ser representado por cualquier tensión comprendida entre 0V y 0.8V, y el nivel lógico alto por cualquier tensión entre 2V y 5V. En rango de valores entre 0.8V y 2V es la región de funcionamiento impredecible.

En la salida, los rangos varían sensiblemente. Un 0 se representa por cualquier tensión entre 0V y 0.4V, y un 1 se representa mediante cualquier tensión entre 2.4V y 5V. La descripción gráfica de ésto se muestra en la figura 1.2.

1.2.2 CMOS

En la entrada, un 0 está representado por cualquier tensión entre 0V y 1.5V, y un 1 por cualquier entre 3.5V y 5V.

En la salida, un 0 es cualquier tensión entre 0V y 0.1 V y un 1 por cualquiera entre 4.9V y 5V. Ver la figura 1.3.

1.3 Inmunidad al ruido: margen de ruido

Las fluctuaciones de tensión en la línea de alimentación, las radiaciones electromagnéticas de alta frecuencia generadas por conductores adyacentes o cualquier

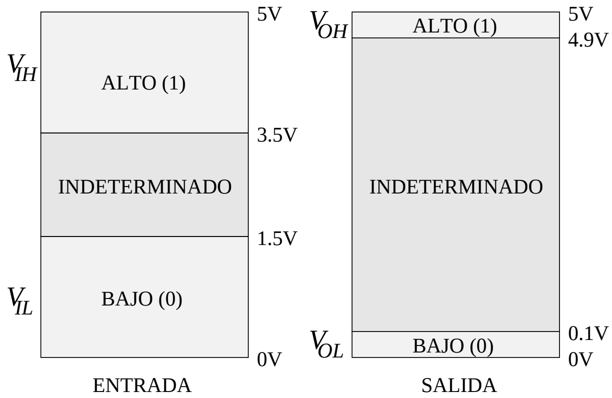


Figura 1.3: Niveles lógicos CMOS.

otra fuente de ruido externo (un rayo por ejemplo) puede modificar la tensión de una línea conductora dentro de un CI y por tanto, confundir el nivel lógico original.

Para no verse afectados adversamente por el ruido, los CI deben tener cierta inmunidad al ruido, que se define como la capacidad para tolerar ciertas variaciones de tensión no deseadas en sus entradas sin que cambie el estado de salida.

Por ejemplo, en el caso de la lógica TTL, supongamos que dos puertas se conectan de forma que una (A) proporciona una entrada a la otra (B). Si A está proporcionando un 1 a B con 2.4V ($V_{OH(min)}$) y por el ruido esta tensión baja hasta menos de 2V ($V_{IH(min)}$), el 1 podría ser interpretado como un 0. Por tanto, el margen de ruido para el nivel alto(V_{NH}) es de 0.4V. Es decir:

$$V_{NH} = V_{OH(min)} - V_{IH(min)}. \tag{1.1}$$

Idéntica situación ocurre para el nivel bajo. Si un 0 está siendo representado por 0.4V a la salida de A y debido al ruido esta tensión pasa a ser más de 0.8V, el 0 podría ser interpretado como un 1. Por tanto, el margen de ruido para el nivel bajo (V_{NL}) es de 0.4V. Así:

$$V_{NL} = V_{IL(max)} - V_{OL(max)}. \tag{1.2}$$

En el caso de CMOS, V_{NH} =1.4V y V_{NL} = 1.4, lo que indica que la familia CMOS es más inmune al ruido que la TTL.

1.4 Retardo de propagación

Cuando una señal digital pasa a través de un circuito lógico, siempre experimenta un retardo temporal llamado *tiempo de retardo de propagación.* Este tiempo es muy importante porque limita la frecuencia máxima a la que es posible trabajar.

Existen dos tiempos de retardo de propagación (ver figura 1.4):

1. t_{PLH} (tiempo de propagación de bajo a alto): es el tiempo entre un determinado punto del impulso de entrada y el correspondiente impulso de salida, cuando la salida cambia de 0 a 1.

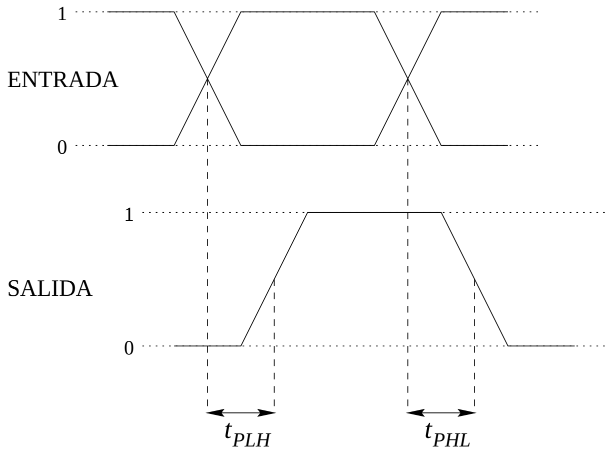


Figura 1.4: Retardos de propagación.

2. t_{PHL} (tiempo de propagación de alto a bajo): es el tiempo entre un determinado punto del impulso de entrada y el correspondiente impulso de salida, cuando la salida cambia de 1 a 0.

1.5 Fan-out

El *fan-out* de una puerta es el número máximo de entradas a puertas (de la misma familia que la puerta en cuestión) que es posible conectar. Si este número se supera, podemos salirnos de los niveles lógicos y por tanto, el circuito no funcionaría.

Para comprender el porqué del *fan-out*, es necesario conocer algo acerca de la estructura interna de las puertas y ésto, evidentemente va a depender de la familia lógica empleada. El *fan-out* se mide en *unidades de carga* (entradas a puertas de la misma familia).

1.5.1 TTL

En TTL, las puertas lógicas se comportan como una resistencia al paso de la corriente que siempre fluye desde la fuente de alimentación (Vcc) hacia la tierra (GND). Supongamos que conectamos dos puertas de forma que una (A) genera una entrada para otra (B) y llamemos R_A a la resistencia interna de la puerta A y a R_B a la resistencia interna de B. Si A proporciona un 1, la corriente fluirá desde A hasta B. Cierto potencial cae en R_A, por lo que el 1 nunca es Vcc. Si A ataca a i puertas B, todas las R_{B_i} están en paralelo, lo que genera una resistencia equivalente menor. Esto permite que pase más corriente a través de R_A y por lo tanto, caiga más tensión en dicha resistencia. En consecuencia, la tensión a la salida de A puede llegar a caer por debajo de 2.4V y en este punto dejaremos de proporcionar un 1.

Similar situación ocurre cuando A proporciona un 0. En este caso, muchas puertas B inyectan corriente a través de la salida de A que es ahora un sumidero de corriente (las puertas B son fuentes de corriente). Si el potencial que cae en

R_A es suficientemente alto como para superar los 0.4V, A dejará de proporcionar un 0.

Notar que para mantener el 1 lógico podemos permitirnos una caída de 2.6V (5V - 2.4V), mientras que para 0 el aumento no puede sobrepasar los 0.4V. Por tanto, en TTL, el establecimiento del 0 es el factor más crítico en la determinación del *fan-out*.

1.5.2 CMOS

En CMOS, las puertas lógicas se comportan como condensadores. Un condensador se asemeja a un aislante cuando la señal que se le aplica es constante y deja pasar la corriente cuando ésta es variable. Por tanto, cuando la frecuencia interna de las señales lógicas es baja, las resistencias internas de las puertas CMOS son muy altas y el *fan-out* sería muy alto, ya que apenas hay consumo de corriente y ésto provoca que no caiga tensión en las puertas. Sin embargo, cuando la frecuencia aumenta, la resistencia capacitiva disminuye, aumentando el consumo de corriente. Así las caídas de tensión aumentan y el *fan-out* disminuye por el mismo efecto que en TTL[2]. Resumiendo: en CMOS el *fan-out* depende de la frecuencia de operación.

1.6 Disipación de potencia

Puesto que circula corriente desde la fuente de alimentación hacia tierra, pasando por las resistencias internas de las puertas (ya sean de tipo resistivo o capacitivo), se consume energía eléctrica que se transforma en energía calorífica. En consecuencia, las puertas lógicas disipan calor y puede hablarse de potencia consumida que depende del voltaje de alimentación y de la cantidad de corriente consumida. La expresión genérica para la potencia es

$$P = V \times I \tag{1.3}$$

donde V es voltaje e I es intensidad. La potencia se mide en vatios (W).

1.6.1 TTL

En TTL, la corriente que circula cuando la salida es 1 (I_{CCH}) es mayor que la corriente que circula cuando la salida es 0 (I_{CCL}). Ya que las señales lógicas son variables en el tiempo, lo más común es suponer un ciclo de trabajo del 50% y por lo tanto se puede hablar de una disipación de potencia media

$$P_D = \mathrm{Vcc} \times I_{CC}, \tag{1.4}$$

donde I_{CC} es la intensidad media de corriente que pasa a través de la puerta, es decir

$$I_{CC} = \frac{I_{CCH} + I_{CCL}}{2}. \tag{1.5}$$

[2]Hay otra forma alternativa de explicar por que cuando aumenta la frecuencia, el *fan-out* disminuye: sabiendo que los condensadores se cargan describiendo una exponencial negativa con el tiempo (para una intensidad de corriente constante), si el *fan-out* aumenta, los condensadores conectados a una puerta tocan a menos corriente y por lo tanto, hay que esperar más a que se cargen.

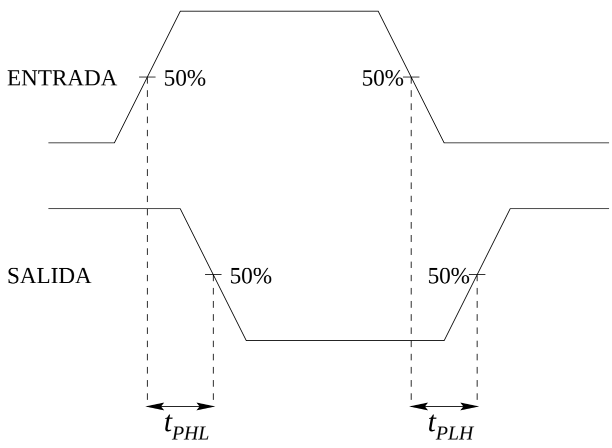


Figura 1.5: Tiempos de propagación asociados a un inversor.

1.6.2 CMOS

Mientras que en TTL el consumo de energía es constante en independiente de la frecuencia (manteniendo un ciclo de trabajo constante), para CMOS el consumo depende de dicha frecuencia en independientemente del ciclo de trabajo. Esto es consecuencia directa de que las resistencias internas de las puertas CMOS son de tipo capacitivo y su resistencia disminuye con la frecuencia de operación. Por tanto, para frecuencias de funcionamiento muy bajas, los dispositivos CMOS consumen muy poca potencia. Esto los hace ideales para relojes de pulsera, calculadoras, ..., elementos que deben ser alimentados por baterías pequeñas.

1.7 Producto velocidad-potencia

Esta medida es sólo útil en CMOS ya que es la única familia lógica (de las dos estudiadas) cuyo consumo depende de la velocidad (frecuencia) de operación. En TTL este producto es constante respecto de la velocidad. Si se indica es para poder comparar CMOS y TTL. El producto velocidad-potencia se mide típicamente en pico-Julios (pJ).

1.8 Prácticas

1.8.1 Determinación del tiempo de propagación

Objetivos

El objetivo de esta práctica es el estudio de una característica muy importante de todos los circuitos digitales: el tiempo de retardo de propagación. Este es el tiempo transcurrido desde la aplicación de un impulso o transición a la entrada, hasta la ocurrencia del impulso o transición asociado a la salida. El tiempo de retardo de propagación determina la máxima frecuencia a la que el circuito puede llegar a trabajar.

La figura 1.5 muestra *idealmente* lo que está ocurriendo a la entrada y a la salida de un inversor. Debido al tiempo de propagación, la transición a la salida

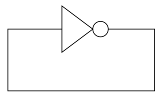


Figura 1.6: Montaje del inversor realimentado.

se produce un tiempo después que la transición de entrada. Yaque existen dos flancos por transición, se habla de tiempo de propagación para ir desde alto a bajo (t_{PHL}) y tiempo de propagación para ir desde bajo a alto (t_{PLH}). Para la mayoría de las puertas, ambos tiempos son iguales, aunque ésto no tiene por que ser siempre así. Típicamente, miramos cuando la señal tiene el 50% de su valor para realizar la medición de los tiempos.

Desarrollo

Para realizar la práctica necesitaremos un integrado con uno o más inversores (el 7404 que contiene 6 de ellos nos vendrá bien). Alimentaremos la pastilla y cortocircuitaremos la entrada y la salida del inversor al que vamos a medir el retardo de propagación (ver figura 1.6). Esta realimentación provoca que el inversor conmute lo más rápidamente posible, teniendo en cuenta el retardo que produce el cable que cortocircuita.

El circuito anterior conmuta a tanta velocidad que necesitamos un buen osciloscopio para ver la forma de onda, y de éstos no disponemos en el laboratorio. Para salir del paso, vamos a conectar secuencialmente tres inversores, de forma que funcionen como si de uno sólo se tratase. Esto se consigue conectando la salida del primero a la entrada del segundo, la salida del segundo a la entrada del tercero, y para cerrar el lazo, conectaremos la salida del tercero a la entrada del primero. Notar que el número de inversores que hay que conectar para que obtengamos una oscilación, debe ser un número impar, ¿ por qué ?.

Al tener 3 inversores funcionando en serie, la velocidad de conmutación se divide al menos entre tres (sería exactamente 3 si los cables no tuvieran asociado ningún retardo) y ya si es posible visualizar la señal generada entre cualquier punto de nuestro circuito y la tierra. Tu trabajo consiste en medir cuanto tiempo gasta un ciclo y determinar la frecuencia de la señal generada. Utiliza para este fin el osciloscopio (en su defecto, podríamos utilizar un frecuencímetro). Una vez calculada la frecuencia y tomando $t_{PHL} = t_{PLH}$, determina el tiempo de retardo de propagación. ¿ Por qué es un tiempo de propagación promedio ?. promedio encontrado, suponiendo siempre que los cables no introducen retardos. ¿ Para calcular este tiempo, es bueno aumentar el número de inversores utilizados ?. ¿ Por qué es importante que los cables sean lo más cortos posible ?. ¿ Qué ocurre si colocamos más de un cable en paralelo para hacer las conexiones, disminuyendo de esta forma la resistencia entre ellas ?.

Capítulo 2

Biestables

Los dispositivos biestables son circuitos digitales secuenciales (poseen realimentación) que tienen dos estados estables (perduran en el tiempo) llamados *Set* (activación) y *Reset* (desactivación). Los biestables se dividen en cerrojos (latches) y flip-flops. La forma más sencilla de diferenciarlos es teniendo en cuenta que los flip-flops son dispositivos síncronos (los cambios a la salida se producen sincrónicamente con una entrada obligada de reloj), mientras que los cerrojos (también llamados básculas) son asíncronos.

2.1 El cerrojo SR

Es un dispositivo digital secuencial asíncrono biestable que puede construirse por ejemplo con dos puertas NOR realimentadas, tal y como se muestra en la figura 2.1. La figura 2.2 muestra el símbolo lógico asociado y en la tabla 2.1 podemos ver su tabla de verdad. Notar que se trata de un dispositivo asíncrono.

2.2 El cerrojo SR con entrada de habilitación

Este dispositivo está a mitad de camino entre el cerrojo SR y el flip-flop SR (como veremos más adelante). Posee una entrada de habilitación (ENable) que permite operar al cerrojo. Las figuras 2.3, 2.4 y la tabla 2.2 presentan su estructura interna, el símbolo lógico asociado y su tabla de verdad.

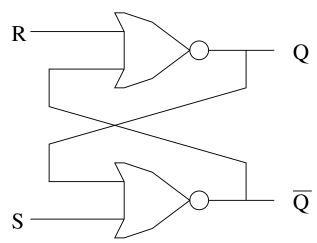


Figura 2.1: Estructura interna de un cerrojo SR fabricado con puertas NOR.

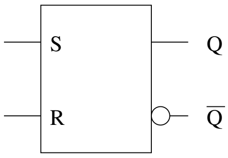


Figura 2.2: Símbolo lógico de un cerrojo SR.

S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

Tabla 2.1: Tabla de verdad de un cerrojo SR.

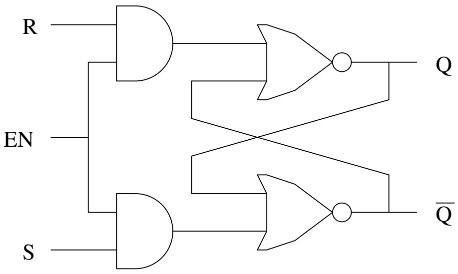


Figura 2.3: Estructura interna de un cerrojo SR (NOR) con entrada de habilitación.

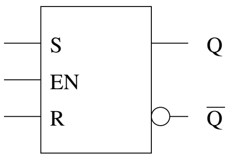


Figura 2.4: Símbolo lógico de un cerrojo SR con entrada de habilitación.

EN	S	R	Q	$\overline{Q}$
0	X	X	Q	$\overline{Q}$
1	0	0	Q	$\overline{Q}$
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0

Tabla 2.2: Tabla de verdad de un cerrojo SR con entrada de habilitación.

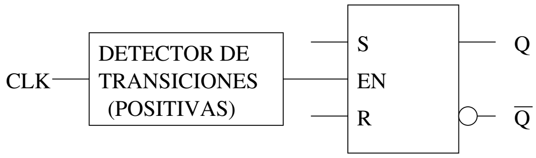


Figura 2.5: Estructura de un flip-flop SR disparado por flanco (positivo).

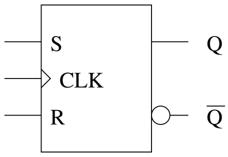


Figura 2.6: Símbolo lógico de un flip-flip SR disparado por flanco positivo.

2.3 El flip-flop SR

Un flip-flop es un dispositivo secuencial síncrono biestable. Existen varios tipos de flip-flops dependiendo de cuando leen las entradas y generan las salidas en función de las transiciones de reloj. En este tema sólo vamos a estudiar los flip-flops disparados por flanco por ser los más sencillos de construir y también los más utilizados.

Los flip-flops disparados por flanco actúan cuando se produce una transición de reloj y puesto que existen dos posibles transiciones, existen dos tipos de flip-flops: los disparados por flancos positivos (CLK: $0 \rightarrow 1$) y los disparados por flancos negativos (CLK: $1 \rightarrow 0$). Ambos tipos son equivalentes, pero no es corriente mezclarlos en la construcción de circuitos secuenciales.

La figura 2.5 muestra la estructura interna de un flip-flop SR disparado por flanco. Se compone de un detector de transiciones (positivas o negativas) y de un cerrojo SR con entrada de habilitación. La figura 2.6 y la tabla 2.3 muestran el símbolo lógico asociado y su tabla de verdad.

El detector de transiciones (en este caso positivas) puede construirse como se indica en la figura 2.7. Dicho circuito produce un pulso positivo (a 1) de una duración igual al tiempo de retardo de la puerta NOT. Este pulso de ínfima duración, generado inmediatamente después de la transición de reloj se utiliza para habilitar el cerrojo SR sólo durante ese instante de tiempo, instante en el que se tienen en cuenta las entradas (antes da igual el valor de éstas). Transcurrido un intervalo de tiempo igual al retardo de propagación del cerrojo, la salida es activará en consecuencia.

CLK	S	R	Q	$\overline{Q}$
$\uparrow$	0	0	Q	$\overline{Q}$
$\uparrow$	0	1	0	1
$\uparrow$	1	0	1	0
$\uparrow$	1	1	*	*

Tabla 2.3: Tabla de verdad de un flip-flop SR disparado por flanco positivo. ** sifnifica: nunca 00.

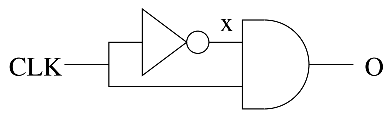


Figura 2.7: Detector de transiciones positivas.

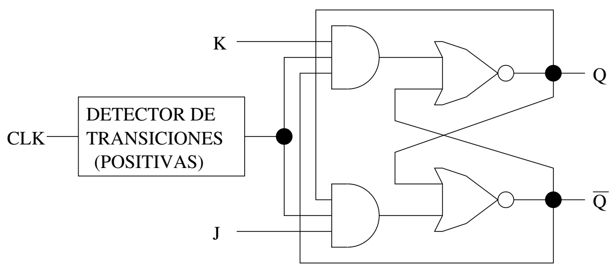


Figura 2.8: Estructura de un flip-flop JK disparado por flanco (positivo).

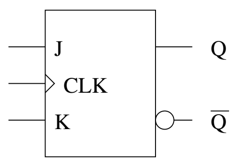


Figura 2.9: Símbolo lógico de un flip-flop JK disparado por flanco positivo.

CLK	J	K	Q	$\overline{\text{Q}}$
$\uparrow$	0	0	Q	$\overline{\text{Q}}$
$\uparrow$	0	1	0	1
$\uparrow$	1	0	1	0
$\uparrow$	1	1	$\overline{\text{Q}}$	Q

Tabla 2.4: Tabla de verdad de un flip-flop JK disparado por flanco positivo.

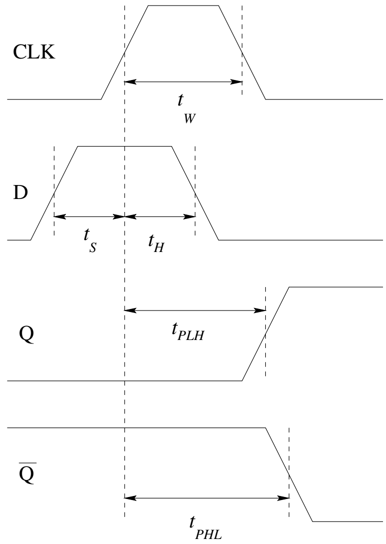


Figura 2.10: Tiempos de operación de un flip-flop D.

2.4 El flip-flop JK disparado por flanco

El biestable JK supera al SR en versatilidad, en concreto porque carece del estado indeseable para S=R=1. Se construye a partir de un SR que tiene las salidas Q y $\overline{Q}$ realimentadas a las entradas (ver figuras 2.8 y 2.9). Cuando $J = K = 1$, el flip-flop bascula, circunstancia muy útil en la construcción de contadores (ver tabla 2.4).

2.5 Características de operación

Los siguientes parámetros son proporcionados por el fabricante del dispositivo biestable, y son necesarios a la hora de diseñar circuitos secuenciales prácticos. La figura 2.10 muestra un ejemplo de situación de algunas de las características de operación para un flip-flop tipo D.

2.5.1 Retardo de propagación: t_{PLH} y t_{PHL}

Es el intervalo de tiempo que transcurre desde que ocurre una transición en la entrada, hasta que actúan las salidas. Así, t_{PLH} habla del tiempo necesario para que la salida cambie de 0 a 1 y viceversa para t_{PHL}.

2.5.2 Tiempo de set-up (establecimiento): t_S

Es el tiempo mínimo que deben permanecer constantes las entradas de datos (S,R,D,J y K) del flip-flop antes de que ocurra el flanco de reloj.

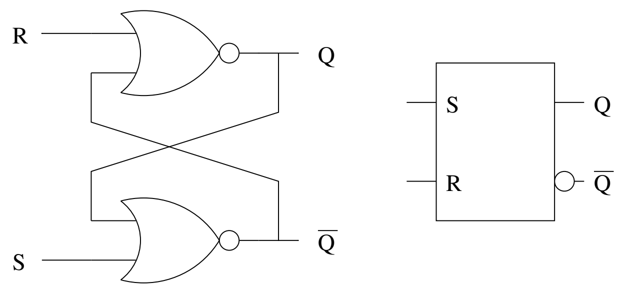


Figura 2.11: Montaje del cerrojo SR con puertas NOR y símbolo lógico.

2.5.3 Tiempo de hold (mantenimiento): t_H

Es el tiempo mínimo que deben permanecer constantes las entradas de datos (S,R,D,J y K) del flip-flop después de que ocurra el flanco de reloj.

2.5.4 Anchura de los impulsos: t_W

Es la duración mínima permitida para los impulsos de reloj, PRESET y CLEAR.

2.5.5 Frecuencia máxima de funcionamiento: f_{max}

Es la máxima frecuencia de reloj a la que se puede disparar el flip-flop de forma fiable.

2.5.6 Disipación de potencia: P

Potencia consumida en vatios por el flip-flop.

2.6 Prácticas

2.6.1 El biestable SR con puertas NOR

El diagrama de un cerrojo con puertas NOR junto con su símbolo lógico, se muestran en la figura 2.11. Los terminales de control de las puertas se denominan R y S (RESET y SET). Si S = R = 0, las puertas NOR se habilitan y con respecto a la otra entrada cada puerta se comporta como un inversor. Entonces, con S = R = 0 el cerrojo puede adoptar uno de sus dos estados posibles, que dependerá de su estado anterior. Así, cuando S = R = 0, los terminales de entrada R y S no influyen en el estado del cerrojo.

Supongamos ahora que el terminal de entrada R es igual a 1; entonces la puerta atacada por R queda inhabilitada y Q tomará el valor 0, mientras que $\overline{Q} = 1$. Es decir, cuando R = 1 se ha puesto en RESET el cerrojo. Si el cerrojo hubiera estado originalmente en RESET, con R = 1 no se produciría cambio de estado; sin embargo, si el estado hubiese sido el de SET, al hacer R = 1 se habría originado un cambio de estado del SET al RESET.

Con S = R = 0 es posible cualquier estado de la puerta. Si R cambia a 1, permanentemente o temporalmente, el cerrojo pasará al estado de RESET. De forma similar, partiendo de S = R = 0, si S cambiase a 1, permanentemente o

S	R	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Tabla 2.5: Tabla de verdad para el biestable SR con puertas NOR.

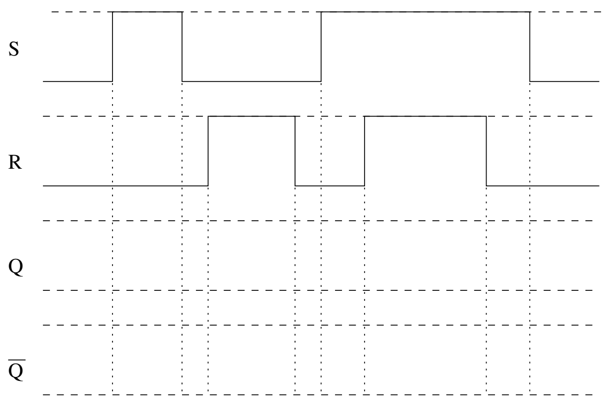


Figura 2.12: Cronograma del cerrojo RS usando puertas NOR.

temporalmente, el cerrojo iría al estado de SET o permanecería en ese estado si previamente estaba en él.

Para el desarrollo de esta práctica, necesitaremos un circuito integrado 7402 (cuádruple puerta NOR).

Una vez realizado el montajes de la figura 2.11 se completará el cronograma (figura 2.12) y la tabla de verdad 2.5.

2.6.2 El biestable SR con puertas NAND

Cualquier cosa que pueda hacerse con puertas NOR puede hacerse también con puertas NAND. La figura 2.13 muestra el diagrama lógico del biestable SR con puertas NAND.

El funcionamiento es similar al cerrojo con puertas NOR, a excepción de

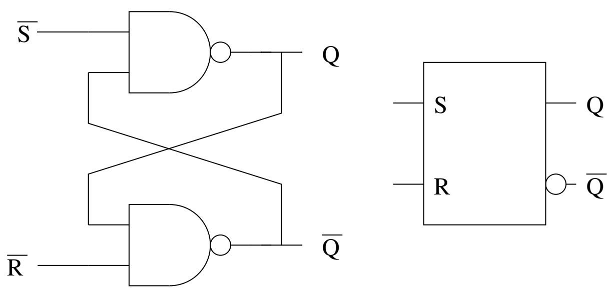


Figura 2.13: Montaje del cerrojo RS con puertas NAND y símbolo lógico.

$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Tabla 2.6: Tabla de verdad para el cerrojo SR con puertas NAND.

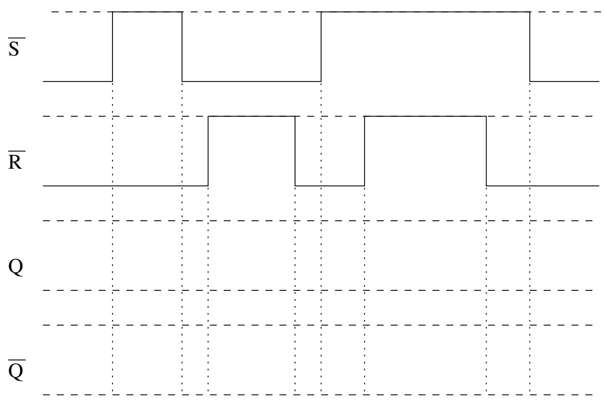


Figura 2.14: Cronograma del cerrojo SR usando puertas NAND.

que el nivel activo en las entradas es 0, y la condición no deseable se produce cuando S = R = 0, ya que da lugar a un estado siguiente indeterminado.

Para el desarrollo de esta práctica, necesitaremos un circuito integrado 7404 (cuádruple puerta NAND).

Una vez realizado el montaje de la figura 2.13, se completará el cronograma (figura 2.14) y la tabla de verdad 2.6.

2.6.3 El cerrojo D obtenido a partir del cerrojo SR

Para conseguir un cerrojo D a partir de un cerrojo SR, entre las entradas S y R del cerrojo se intercala un inversor, provocando que la entrada D sea igual a la entrada R, y la entrada S sea la negación de D.

Al tratarse de un cerrojo (sin entradas de habilitación o sincronismos), al valor contenido en D se va a propagar lo más rápido posible a la salida, así que la utilidad del cerrojo D es también muy reducida, si tenemos en cuenta que sólo sirve como elemento de retardo o como restaurador de señal (en casos de transmisiones de señales digitales a través de cables muy largos).

Para el desarrollo de esta práctica necesitaremos sólo un integrado 7402 para montar el cerrojo SR y un integrado 7404 para utilizar de él un inversor, que se

D	Q	$\overline{Q}$
0		
1		

Tabla 2.7: Tabla de verdad del cerrojo D usando SR.

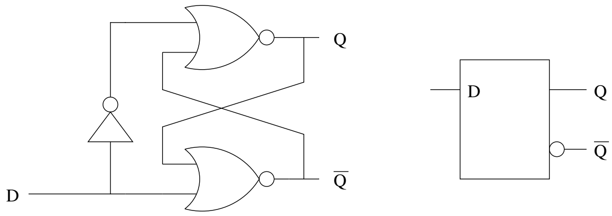


Figura 2.15: Montaje del cerrojo D a partir de un SR y símbolo lógico.

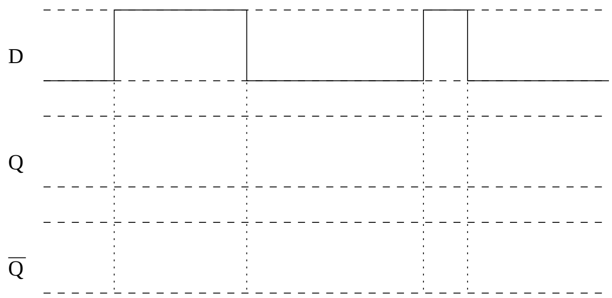


Figura 2.16: Cronograma del cerrojo D usando SR.

montará tal y como se indica en la figura 2.15, que muestra también el diagrama lógico asociado al cerrojo D.

Una vez realizado el montaje, completaremos el cronograma presentado en la figura 2.16 y deduciremos (rellenando) la tabla de verdad mostrada en la figura 2.7.

2.6.4 El cerrojo JK

El cerrojo JK es un elemento secuencial muy utilizado en el diseño de flip-flops JK, aunque por sí solo tiene poco interés desde el punto de vista de diseño de contadores y registros. El inconveniente del cerrojo JK (y que no tienen sus hermanos mayores los flip-flops) es que presenta un estado inestable o de constante transición, en el cual es imposible determinar el valor de su salida. Es trabajo del alumno encontrar y explicar dicho estado inestable.

Para el desarrollo de esta práctica necesitaremos sólo un integrado 7400 (con cuatro puertas NAND de dos entradas). Dichas puertas se montarán tal y como

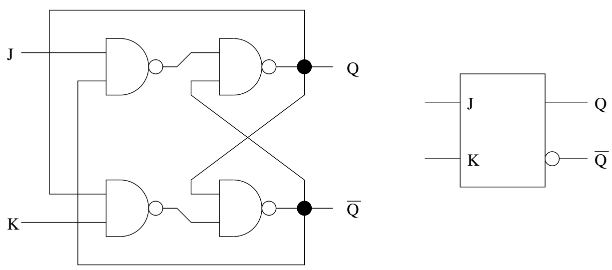


Figura 2.17: Montaje del cerrojo JK y símbolo lógico.

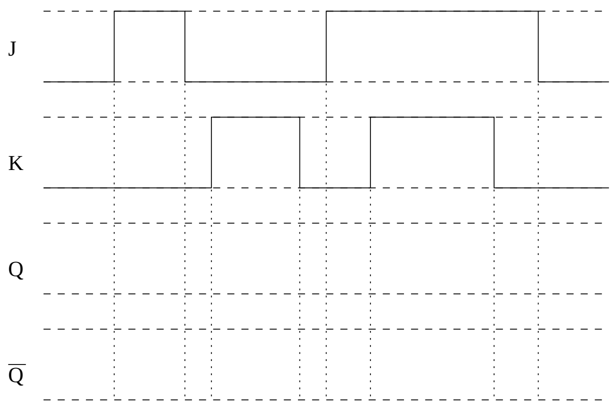


Figura 2.18: Cronograma del cerrojo JK.

J	K	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

Tabla 2.8: Tabla de verdad del cerrojo JK.

se describe en la figura 2.17, que muestra también el diagrama lógico asociado al cerrojo. Una vez realizado el montaje, completaremos el cronograma presentado en la figura 2.18 y deduciremos la tabla de verdad 2.8.

2.6.5 El cerrojo D obtenido a partir del cerrojo JK

El cerrojo D es el único cerrojo util' que es posible montar a partir de un cerrojo JK, gracias a que nunca ocurre que J = K = 1. Posee una única entrada llamada D. Para conseguir este comportamiento, entre las entradas J y K del cerrojo se intercala un inversor, provocando que la entrada D sea igual a la entrada J, y la entrada K sea la negación de D.

Para el desarrollo de esta práctica necesitaremos sólo un integrado 7400 para montar el cerrojo JK de la práctica anterior y un integrado 7404 para utilizar de él un inversor, que se montará tal y como se indica en la figura 2.19, que muestra también el diagrama lógico asociado al cerrojo D.

Una vez realizado el montaje, completaremos el cronograma presentado en la figura 2.20 y deduciremos (rellenando) la tabla de verdad 2.9.

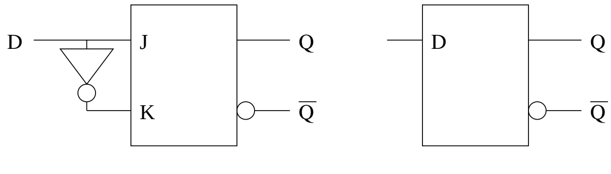


Figura 2.19: Montaje del cerrojo D a partir de un JK y símbolo lógico.

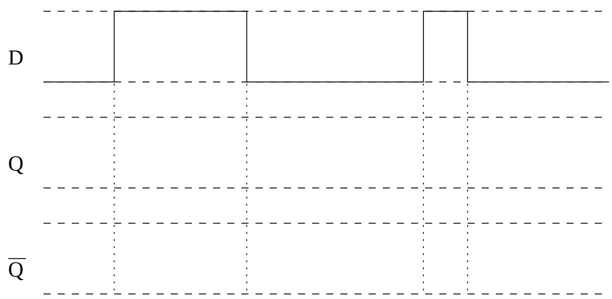


Figura 2.20: Cronograma del cerrojo D usando JK.

D	Q	$\overline{Q}$
0		
1		

Tabla 2.9: Tabla de verdad del cerrojo D usando JK.

2.6.6 El cerrojo SR con entrada de habilitación

Las entradas S y R se denominan con frecuencia entradas de datos, ya que la información presente en estos terminales determina lo que se almacena en el cerrojo. Muchas veces resulta de utilidad disponer de un mecanismo que nos permita conectar o aislar el cerrojo con una fuente de datos, tal y como se muestra en la figura 2.21. En este tipo de biestable tenemos tres entradas (S, R y la señal de habilitacion EN). Si EN = 0, entonces las dos puertas están inhabilitadas, y el estado del biestable no cambia. Si EN = 1, se habilitan las entradas del circuito, y según sea su valor, así será el estado siguiente del biestable.

Para el montaje de esta práctica, sólo necesitamos un circuito integrado 7404 (cuádruple puerta NAND). Para la entrada de habilitación, utilizaremos uno de los interruptores del entrenador.

Montar el circuito de la figura 2.21. Completar el cronograma de la figura 2.22. Completar la tabla de verdad del circuito.

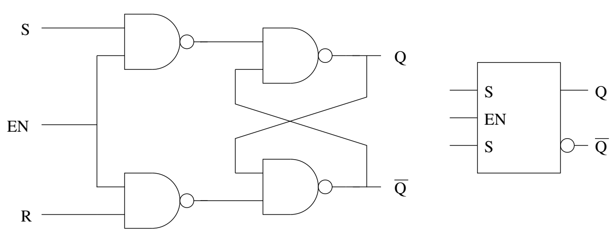


Figura 2.21: Montaje del cerrojo SR con entrada de habilitación y símbolo lógico.

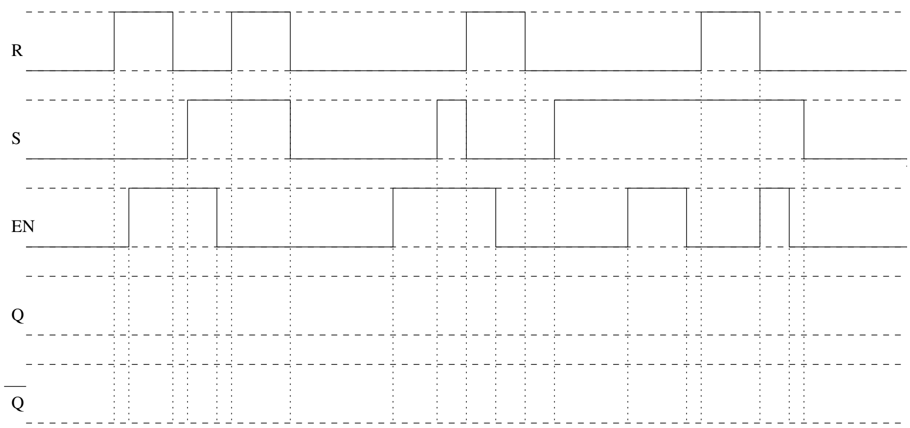


Figura 2.22: Cronograma del cerrojo SR con entrada de habilitación.

EN	S	R	Q	$\overline{Q}$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Tabla 2.10: Tabla de verdad para el cerrojo SR con entrada de habilitación.

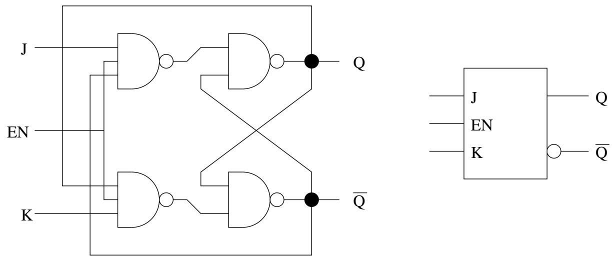


Figura 2.23: Montaje del cerrojo JK con entrada de habilitación y símbolo lógico.

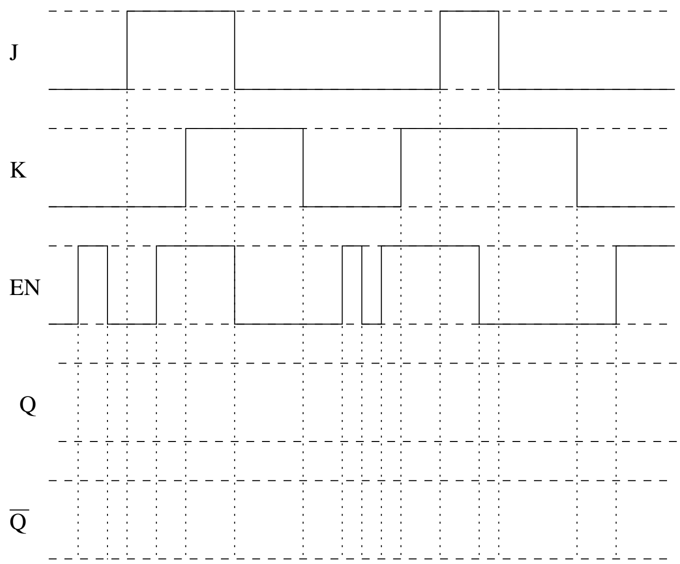


Figura 2.24: Cronograma del cerrojo JK con entrada de habilitación.

2.6.7 El cerrojo JK con entrada de habilitación

Vamos ahora a estudiar una versión nueva del cerrojo JK que posee una entrada de sincronización o de habilitación. Esta entrada se utiliza para sincronizar el cambio de estado en el cerrojo (de acuerdo con los valores presentados en las entradas J y K) con una señal externa que se introduce por su entrada EN (ENable).

El cerrojo JK con entrada de habilitación es muy parecido al cerrojo JK, salvo que las puertas de entrada (donde J y K atacan) son ahora de 3 entradas para permitir así el uso de la señal de habilitación. Esta señal es muy importante porque si controlamos la anchura de sus pulsos convenientemente, podemos construir un flip-flop disparado por flanco a partir de nuestro inutil' cerrojo JK. En una práctica posterior veremos como construirlo.

Para el desarrollo de esta práctica necesitaremos un integrado 7410 con NAND de 3 entradas y un 7400 con NAND de 2 entradas. El esquema propuesto se muestra en la figura 2.23, que muestra también el símbolo lógico asociado.

EN	J	K	Q	$\overline{Q}$
0	X	X		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Tabla 2.11: Tabla de verdad del cerrojo JK con entrada de habilitación.

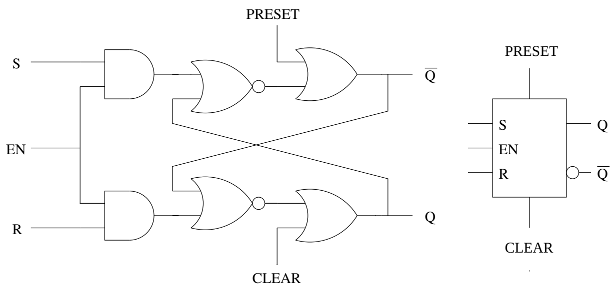


Figura 2.25: Montaje del cerrojo SR con entradas PRESET y CLEAR y símbolo lógico.

Una vez realizado el montaje, rellenaremos el cronograma presentado en la figura 2.24 y deduciremos la tabla 2.11.

2.6.8 El cerrojo SR con entradas de habilitación, PRESET y CLEAR

Este circuito (figura 2.25) tiene dos entradas asíncronas (PRESET y CLEAR) cuya función es colocar la salida del biestable a 1 ó 0 asíncronamente, es decir, independientemente del estado del reloj.

Para esta práctica necesitamos un circuito integrado 7408 (cuádruple puerta AND de dos entradas), un circuito integrado 7400 (cuádruple puerta NAND de dos entradas) y un circuito integrado 7402 (cuádruple puerta NOR de dos entradas).

Realizar el montaje del circuito de la figura 2.25. Comprobar y escribir su tabla de verdad (tabla 2.12). Completar el cronograma de la figura 2.26.

2.6.9 El cerrojo JK con entradas de habilitación, PRESET y CLEAR

El cerrojo JK, con entradas de habilitación, inicialización y borrado es muy similar al cerrojo JK con sólo habilitación. La única diferencia consiste en que a las puertas de salida entran sendas entradas asíncronas que fuerzan el estado de cerrojo, independientemente del valor de las entradas J, K y EN. Las entradas asíncronas de inicialización y borrado son muy útiles en la construcción de registros con estas propiedades.

Para el desarrollo de esta práctica necesitaremos dos integrados 7410 que contienen 3 NAND de 3 entradas cada uno. El esquema propuesto se muestra

PRESET	CLEAR	S	R	Q	$\overline{Q}$
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		

Tabla 2.12: Tabla de verdad del cerrojo SR con entradas de habilitación, PRESET y CLEAR.

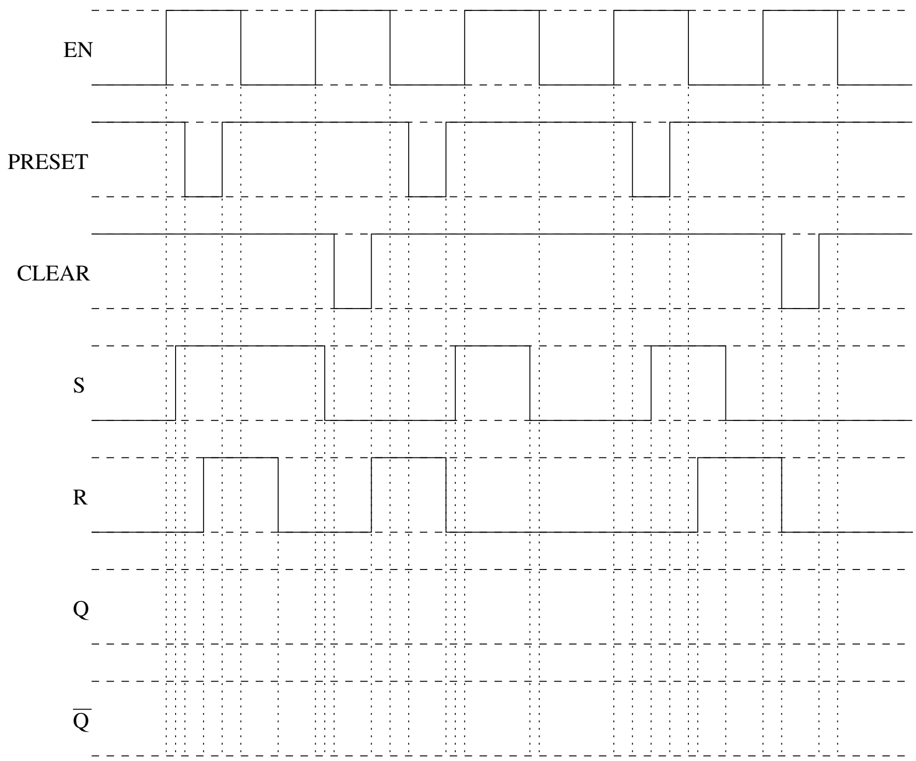


Figura 2.26: Cronograma del cerrojo SR con entradas habilitación, PRESET y CLEAR.

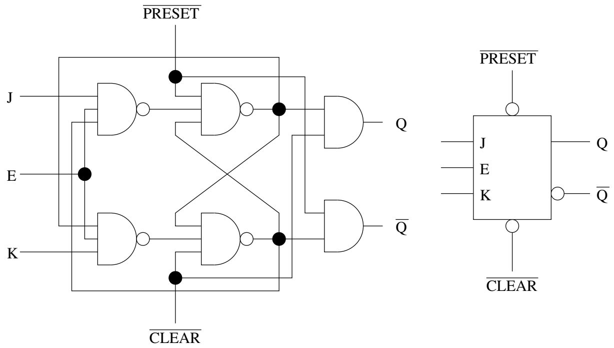


Figura 2.27: Montaje del cerrojo JK con entradas de habilitación, $\overline{\text{PRESET}}$ y $\overline{\text{CLEAR}}$ y símbolo lógico.

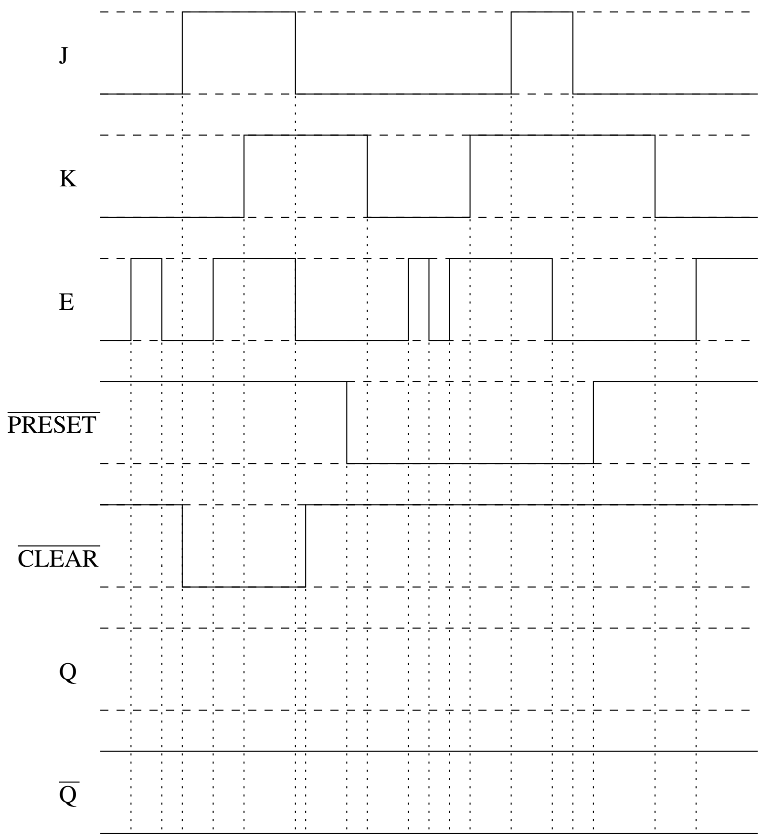


Figura 2.28: Cronograma del cerrojo JK con entrada de habilitación, $\overline{\text{PRESET}}$ y $\overline{\text{CLEAR}}$.

$\overline{\text{PRESET}}$	$\overline{\text{CLEAR}}$	E	J	K	Q	$\overline{\text{Q}}$
0	0	0	0	0		
0	0	0	0	1		
0	0	0	1	0		
0	0	0	1	1		
0	0	1	0	0		
0	0	1	0	1		
0	0	1	1	0		
0	0	1	1	1		
0	1	0	0	0		
0	1	0	0	1		
0	1	0	1	0		
0	1	0	1	1		
0	1	1	0	0		
0	1	1	0	1		
0	1	1	1	0		
0	1	1	1	1		
1	0	0	0	0		
1	0	0	0	1		
1	0	0	1	0		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	0	1		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	0		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		
1	1	1	1	0		
1	1	1	1	1		

Tabla 2.13: Tabla de verdad del cerrojo JK con entradas de habilitación, $\overline{\text{PRESET}}$ y $\overline{\text{CLEAR}}$.

S	R	CLK	Q	$\overline{\text{Q}}$
0	0	↑		
0	1	↑		
1	0	↑		
1	1	↑		

Tabla 2.14: Tabla de verdad del flip-flop SR activo por flanco positivo.

en la figura 2.27, que muestra también el símbolo lógico asociado.

Una vez realizado el montaje, rellenaremos el cronograma presentado en la figura 2.28 y deduciremos la tabla de verdad 2.13, suponiendo que la entrada EN es 1.

2.6.10 El flip-flop SR disparado por flanco positivo

Los flip-flops son dispositivos secuenciales síncronos (necesitan una señal de reloj para funcionar), que se utilizan para almacenar un bit de información. En concreto, los que son activos o disparados por flanco de reloj (existe otra variedad que son activos por nivel) son los más utilizados en la construcción de contadores y registros.

El flip-flop SR disparado por flanco es muy parecido al cerrojo SR con entrada de habilitación, visto en la practica 2.6.6. Como en ella se indicó, la entrada de habilitación del cerrojo puede utilizarse para sincronizar el funcionamiento del cerrojo con una señal de habilitación externa y éste es justamente el propósito de la señal de reloj. La idea es conectar a la entrada de habilitación un circuito capaz de proporcionar un pulso (en este caso positivo). Este circuito se llama detector de transiciones.

Ya que existen dos tipos de transiciones (positivas y negativas) existen dos tipos de flip-flops disparados por flanco: positivos y negativos. La diferencia entre ellos es mínima como veremos y depende de si el circuito que habilita el cerrojo detecta las transiciones positivas o negativas.

La construcción de un detector de transiciones es muy sencillo. Son necesarias un puerta NOT y una puerta AND de dos entradas. El inversor (además de invertir la señal de reloj) va a provocar un desfase entre las transiciones de la señal de reloj, debido al tiempo de retardo inherente a la puerta. Esta señal invertida entra a una entrada de la puerta AND y la otra se alimenta con la señal original, que no posee dicho desfase. Como consecuencia del desfase introducido, a la salida de la puerta AND vamos a registrar un pulso positivo de una duración igual al tiempo de retardo de la puerta NOT que es muy pequeño. En realidad podemos concluir que ya que un inversor es la puerta lógica más simple que es posible construir, ninguna otra puerta va a disfrutar de un tiempo de retardo menor y por lo tanto, la duración del pulso positivo nunca dará lugar a que una puerta NOT o de cualquier otro tipo que lo utiliza como señal de excitación pueda, debido a realimentaciones, realizar más de una transición. Esto es muy importante, ya que gracias a esta mínima (pero suficiente) duración de dicho pulso, el flip-flop va a estar atento a las entradas un periodo de tiempo muy corto. Como dicho pulso se produce muy poco tiempo después de la transición positiva de la señal de reloj, el cerrojo SR pasa a ser un flip-flop SR activo por flanco positivo.

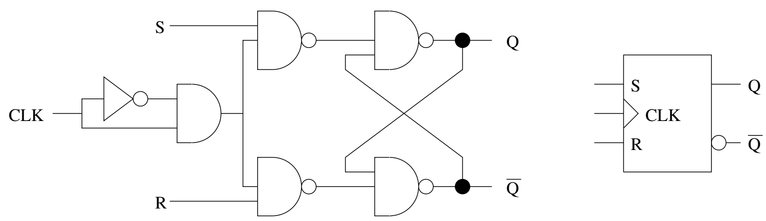


Figura 2.29: Montaje del flip-flop SR disparado por flanco positivo y su símbolo lógico.

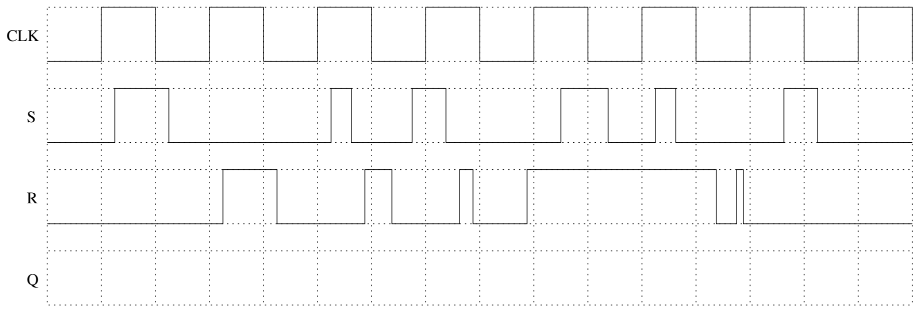


Figura 2.30: Cronograma del flip-flop SR activo por flanco positivo.

Para el desarrollo de esta práctica necesitaremos un integrado 7004 para el inversor, un 7008 con puertas AND y un 7000 para las 4 puertas NAND que forman el cerrojo SR. El montaje es mostrado en la figura 2.29, que muestra también el diagrama lógico asociado.

Una vez realizado el montaje, rellenaremos el cronograma presentado en la figura 2.30 y completaremos la tabla de verdad 2.14.

2.6.11 El flip-flop SR disparado por flanco negativo

Para construir un flip-flop disparado por flancos negativos sólo tenemos que invertir la señal de reloj antes de utilizarla en el circuito detector de flancos negativos. Por tanto, el circuito es idéntico al desarrollado para el flip-flop SR disparado por flanco positivo excepto porque hay un inversor a la entrada de la señal de reloj.

Como veremos, la utilización de flip-flops disparados por flanco positivos o negativos es sólo cuestión de gusto. Circuitos con idénticas características y prestaciones se pueden construir con ambos, pero una vez que decidimos que lógica utilizar, no es recomendable mezclar de los dos tipos.

Para el desarrollo de esta práctica necesitaremos un integrado 7004 para los inversores, un 7008 con puertas AND y un 7000 para las 4 puertas NAND que forman el cerrojo SR. El montaje es mostrado en la figura 2.31 (notar que es una simple ampliación del montaje realizado en la práctica anterior), que muestra también el diagrama lógico asociado.

Una vez realizado el montaje, rellenaremos el cronograma presentado en la figura 2.32 y completaremos la tabla de verdad 2.15.

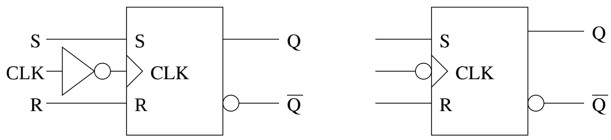


Figura 2.31: Montaje del flip-flop SR disparado por flanco negativo y símbolo lógico.

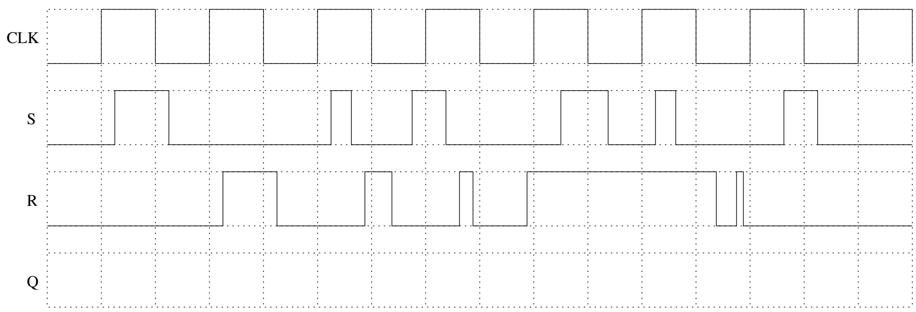


Figura 2.32: Cronograma del flip-flop SR disparado por flanco negativo.

S	R	CLK	Q	$\overline{Q}$
0	0	$\downarrow$		
0	1	$\downarrow$		
1	0	$\downarrow$		
1	1	$\downarrow$		

Tabla 2.15: Tabla de verdad del flip-flop SR disparado por flanco negativo.

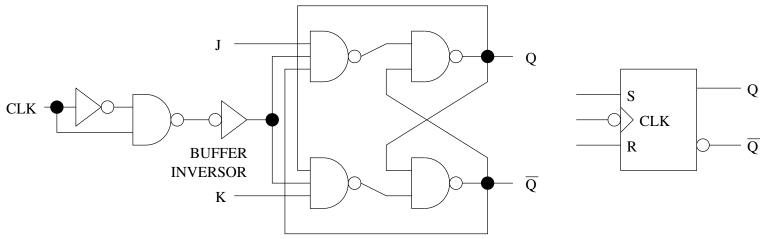


Figura 2.33: Montaje del flip-flop JK disparado por flanco positivo y símbolo lógico.

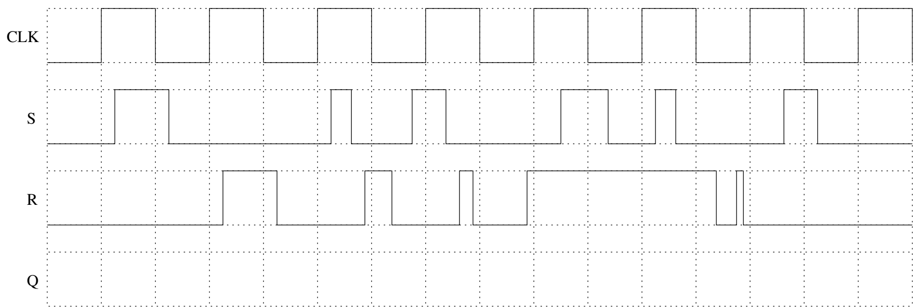


Figura 2.34: Cronograma del flip-flop JK disparado por flanco positivo.

2.6.12 El flip-flop JK disparado por flanco positivo

El flip-flop JK soluciona el problema del SR cuando ambas entradas son 1, mediante las realimentaciones ya conocidas. La figura 2.33 muestra el montaje de un flip-flop JK disparado por flanco positivo.

Para el desarrollo de esta práctica necesitaremos un integrado 7404 para el inversor, un 7400 con puertas NAND de dos entradas y un 7410 para las NAND de tres entradas y un 7406 con buffers inversores. El montaje es mostrado en la figura 2.33, que muestra también el diagrama lógico asociado.

Una vez realizado el montaje, rellenaremos el cronograma presentado en la figura 2.34 y completaremos la tabla de verdad 2.16.

2.6.13 El flip-flop SR maestro-esclavo

Los flip-flops maestro-esclavo (master-slave) han sido ampliamente utilizados hasta la aparición de los disparados por flanco, que poco a poco los van sustituyendo. La razón fundamental es que funcionan de forma idéntica y los

J	K	CLK	Q	$\overline{Q}$
0	0	↑		
0	1	↑		
1	0	↑		
1	1	↑		

Tabla 2.16: Tabla de verdad del flip-flop JK disparado por flanco positivo.

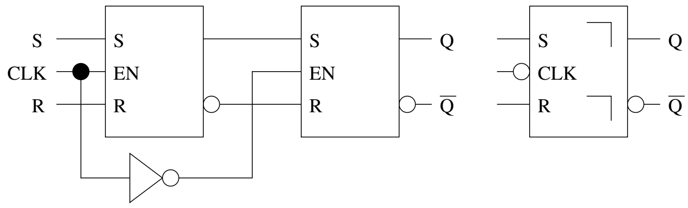


Figura 2.35: Montaje del flip-flop SR maestro-esclavo y símbolo lógico.

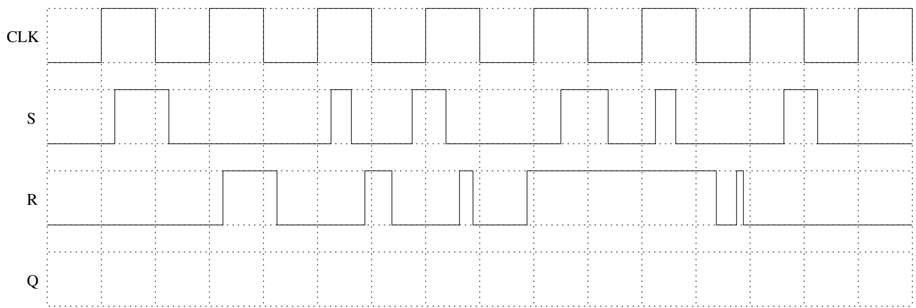


Figura 2.36: Cronograma del flip-flop SR maestro-esclavo.

disparados por flanco necesitan menos puertas lógicas.

La construcción de un flip-flop maestro-esclavo SR se realiza a partir de dos cerrojos SR con entrada de habilitación conectados en cascada, de forma que la señal de reloj entra al cerrojo maestro y la señal de reloj complementada entra al esclavo. La figura 2.35 muestra el montaje y también el símbolo lógico asociado.

Sólo el cerrojo maestro está habilitado cuando el reloj es 1. Durante todo ese intervalo de tiempo, sus salidas irán acorde con sus entradas. Si se produce una variación, la salida actuará en consecuencia. Cuando llega el flanco negativo de reloj, se habilita el cerrojo esclavo (y se deshabilita el maestro), que toma la salida del maestro (que ya no pueden variar porque se encuentra deshabilitado). Por tanto, en un tiempo igual al tiempo de propagación del cerrojo esclavo, justamente después del flanco negativo de reloj, la salida del cerrojo esclavo actúa en consecuencia.

Para montar el circuito necesitaremos los CIs utilizados para montar dos cerrojos SR (ver práctica 2.6.6) y un 7404 para el inversor. Una vez realizado el montaje, rellenaremos el cronograma presentado en la figura 2.36 y completaremos la tabla de verdad 2.17.

S	R	CLK	Q	$\overline{Q}$
0	0	$\downarrow$		
0	1	$\downarrow$		
1	0	$\downarrow$		
1	1	$\downarrow$		

Tabla 2.17: Tabla de verdad del flip-flop SR maestro-esclavo.

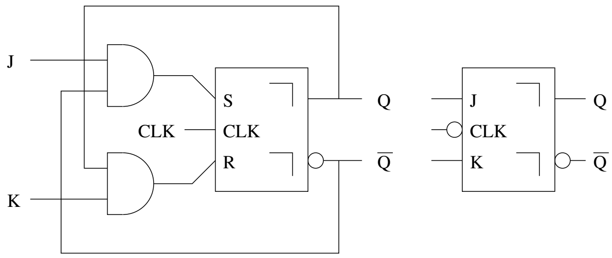


Figura 2.37: Montaje del flip-flop JK maestro-esclavo y símbolo lógico.

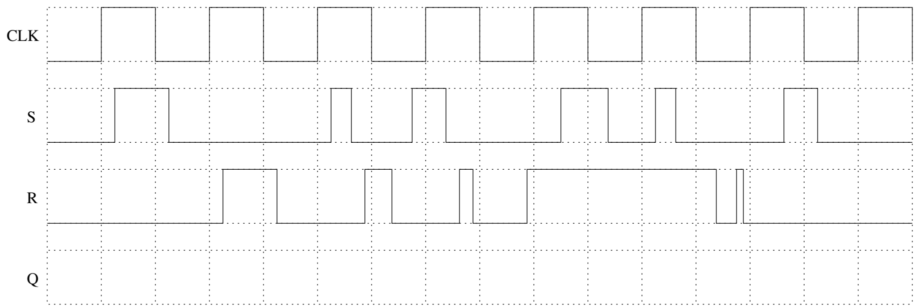


Figura 2.38: Cronograma del flip-flop JK maestro-esclavo.

2.6.14 El flip-flop JK maestro-esclavo

Una vez más, construiremos un flip-flop JK a partir de un SR (esta vez en su versión maestro-esclavo), realimentando las salidas hacia la entrada tal y como se muestra en la figura 2.37, que también muestra el símbolo lógico asociado.

Para montar el circuito necesitaremos sólo dos puertas AND (7808) respecto del montaje anterior. Una vez realizado el montaje, rellenaremos el cronograma presentado en la figura 2.38 y completaremos la tabla de verdad 2.18.

J	K	CLK	Q	$\overline{Q}$
0	0	$\downarrow$		
0	1	$\downarrow$		
1	0	$\downarrow$		
1	1	$\downarrow$		

Tabla 2.18: Tabla de verdad del flip-flop JK maestro-esclavo.

Capítulo 3

Contadores

Los flip-flops pueden utilizarse para construir circuitos secuenciales contadores. Existen una clasificación básica de los contadores, que los divide en asíncronos y síncronos, en función de si la señal de reloj dispara en paralelo a todos los flip-flops (síncrono) o no (asíncrono). Además, los contadores pueden clasificarse (1) atendiendo al tipo de secuencia que generan o (2) al número de estados por el que pasan (módulo).

3.1 Contador binario asíncrono ascendente

Por ser asíncrono, la señal de reloj (señal contada) sólo ataca al flip-flop que contiene el LSb. Por ser binario, la secuencia de cuenteo es $0, 1, \ldots, 2^n - 1, 0, 1, \ldots$, donde n es el número de flip-flops que componen el contador. El módulo de un contador binario de este tipo es 2^n (el máximo que se puede conseguir utilizando n flip-flops). Su construcción es muy sencilla si utilizamos flip-flops JK con J=K=1 – en cuyo caso el flip-flop se llama T –. Un contador binario de 1 bit es un flip-flop JK configurado de esta forma. Cada vez que llega un pulso de reloj, el flip–flop bascula, lo que genera sobre Q una secuencia de pulsos de frecuencia mitad que la frecuencia de reloj. Esta señal puede a su vez utilizarse para disparar otro flip-flop T y a la salida de éste tendremos una secuencia de pulsos de frecuencia (de nuevo) mitad. La figura 3.1 describe un contador asíncrono binario de 3 bits y la figura 3.2 muestra las señales obtenidas a las salidas de los flip-flops, junto con la señal de reloj contada.

Notar que, debido al tiempo de retardo asociado a cada flip-flop, el disparo de éstos nunca se produce al mismo tiempo. Dicha circunstancia limita la máxima frecuencia de reloj a la que pueden contar. Lo que está ocurriendo es que, dependiendo de la frecuencia de reloj y de los retardos de propagación de los

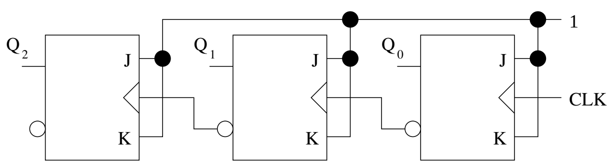


Figura 3.1: Esquema lógico de un contador binario asíncrono de tres bits.

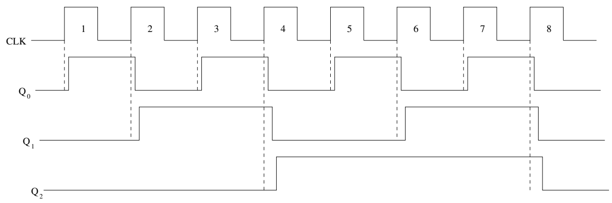


Figura 3.2: Cronograma real de un contador binario asíncrono de tres bits.

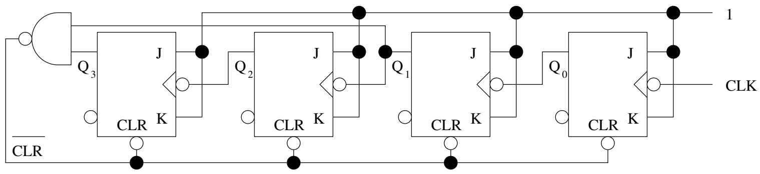


Figura 3.3: Esquema lógico de un contador BCD asíncrono.

flip-flops, puede ocurrir la circunstancia en la que, por ejemplo la condición $Q_0 = Q_1 = Q_2 = 0$ – por la que deben de pasar todos los contadores binarios – puede no llegar nunca a cumplirse. Se dice entonces que el contador es imposible de decodificar.

Generalizando, si $t_{p_{FF}}$ es el tiempo de retardo asociado al un flip-flop y n es el número de flip-flops empleados para construir el contador asíncrono binario, la máxima frecuencia de reloj a la que es posible operar es

$$f_{max} = \frac{1}{t_{p_{FF}} \times n}. \tag{3.1}$$

3.2 Contador BCD asíncrono ascendente

Un contador BCD es un contador de décadas que sigue concretamente los estados $0, 1, \ldots, 9, 0, 1, \ldots$. Su módulo es por tanto 10 y ya que $2^3 < 10 < 2^4$, necesitaremos 4 flip-flops para construirlo. Evidentemente estaremos desperdiciando $16 - 10 = 6$ estados – por eso se dice que la secuencia de cuenteo de un contador BCD es truncada –. La forma más sencilla de truncar en contador

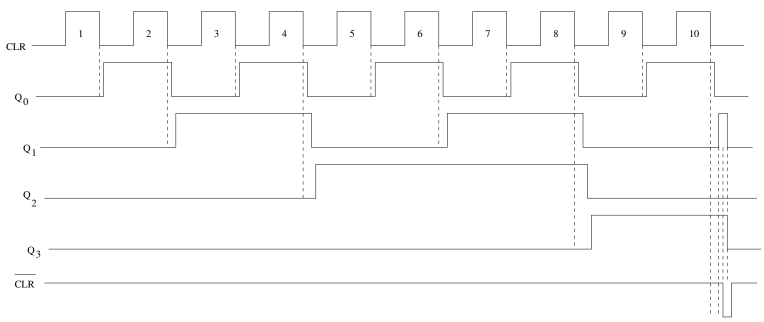


Figura 3.4: Cronograma real de un contador BCD asíncrono.

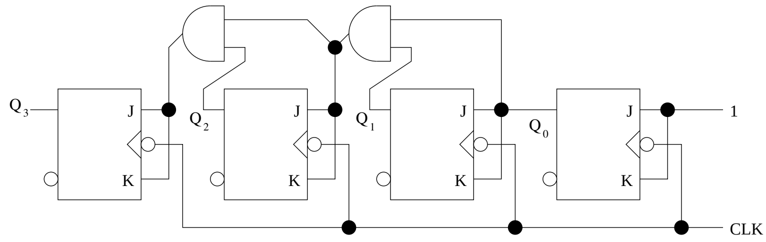


Figura 3.5: Esquema de un contador binario síncrono de 4 bits.

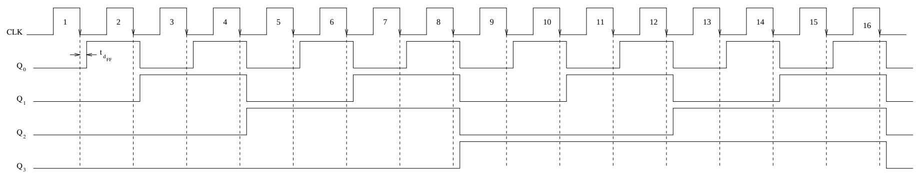


Figura 3.6: Cronograma real de un contador binario síncrono de 4 bits.

consiste en decodificar el estado de numeración más alto (el 9 en este caso) y utilizarlo para llevar el contador hasta el estado de numeración más bajo (el 0). Para dicho fin, pueden utilizarse las entradas asíncronas de CLEAR. La figura 3.3 muestra el esquema lógico de un contador de este tipo y la figura 3.4 presenta el diagrama de tiempos generado. Puede observarse que aparece un *glitch* en la señal de Q_1, inevitable y de duración igual al tiempo de propagación a través de la puerta NAND, más el tiempo que tarda el flip-flop primero en ponerse a cero.

3.3 Contador binario síncrono ascendente

En un contador síncrono, todos los flip-flops se disparan simultáneamente. Esto permite utilizar una frecuencia de reloj en general más elevada que en el caso de los contadores asíncronos, debido a que el mínimo periodo de reloj se limita al tiempo necesario para calcular el siguiente estado utilizando muy pocos niveles de puertas.

El flip-flop más conveniente para construir contadores síncronos sigue siendo el flip-flop JK con sus entradas cortocircuitadas (configurado como un flip-flop T). Un contador binario síncrono de 1 bit es idéntico a uno asíncrono, pero el de 2 bits es diferente. Llamemos Q0 a la salida del flip-flop que almacena el LSb y Q1 al que almacena el MSb. El contador síncrono se construye sabiendo que cuando Q0=1, Q1 debe bascular. Si tenemos un contador con 3 flip-flops, la basculación de Q2 debe producirse si Q0=Q1=1 y así sucesivamente.

La figura 3.5 presenta el esquema lógico de un contador binario síncrono ascendente de 4 bits y la figura 3.6 su cronograma. Hay un aspecto interesante de la implementación anterior que conviene resaltar y se refiere al tamaño de las puertas AND que aparecen asociadas a cada etapa del contador. El sistema empleado calcula la AND de todos los bits anteriores mediante una serie de puertas AND de dos entradas colocadas en cascada. La otra alternativa es utilizar puertas AND con un número creciente de entradas conforme vamos añadiendo flip-flops al contador. De esta forma, un contador de n bits necesitaría una puerta AND de dos entradas, otra de tres, ... y finalmente una puerta con $n-1$ entradas. Esto puede suponer un engorro en implementaciones que no dispongan de tal abanico de puertas y además, aunque a priori es más rápida la versión que utiliza sólo un nivel del puertas AND para calcular el estado siguiente, también suele ocurrir que una puerta AND de dos entradas es más rápida que otra con más entradas. Por estas razones, es más corriente encontrar la versión expuesta en la figura.

Para finalizar, calculemos cual es la máxima frecuencia de reloj a la que podemos trabajar. Concretamente, imaginemos que el contador está en el estado $0111 \cdots 1_2$ donde el LSb es el primer cero. Este provoca que a la salida de todas las puertas AND tengamos un cero. Con la siguiente transición negativa de reloj, Q0=1 y el estado del contador es $111 \cdots 1_2$. La primera puerta AND en generar un 1 a la salida es la que proporciona la entrada el FF2. A continuación, a la salida de la segunda puerta AND tendremos un 1 y así secuencialmente hasta llegar a la puerta AND que excita el flip-flop más significativo. Por tanto, si $t_{d_{gate}}$ es el tiempo de retardo asociado a una puerta AND, ya que existen $n-2$ puertas AND, el 1 en la salida de la puerta AND asociada al bit más significativo tarda en generarse $t_{d_{gate}} \times (n-2)$ unidades de tiempo. A este tiempo hay que sumar el tiempo que transcurrió desde que se produjo la transición negativa de reloj hasta que el FF0 basculó, es decir $t_{d_{FF}}$ unidades de tiempo. Por tanto, el mínimo periodo de reloj permitido es dicha suma y la frecuencia máxima a la que es posible contar es:

$$f_{max} = \frac{1}{t_{p_{FF}} + \times t_{p_{gate}} \times (n-2)}. \tag{3.2}$$

3.4 Contador binario síncrono ascendente/descendente

También llamado contador bidireccional, un contador ascendente/descendente es capaz de contar en dos sentidos dependiendo de una señal de control externa. En general se diseñan para que la secuencia de cuenteo pueda invertirse en cualquier momento. Por ejemplo:

$$\overbrace{0,1,2,3,4}^{\text{ascendente}},\ \underbrace{3,2,1,0}_{\text{descendente}},\ \overbrace{1,2,3,4,5,6}^{\text{ascendente}},\ \underbrace{5,4}_{\text{descendente}},\ \cdots$$

El diseño es bastante sencillo, si sabemos que para que un contador sea descendente, sólo hay que utilizar el valor de las $\overline{Q}$'s en lugar de el valor de las Q's para calcular el siguiente estado. La figura 3.7 muestra el esquema de un contador de este tipo de 3 bits.

3.5 Contador BCD síncrono ascendente

Al igual que en la versión asíncrona, utilizaremos 4 flip-flops para construir un contador truncado. El funcionamiento se entiende mejor si observamos la tabla 3.1 que muestra los estados alcanzados.

Analicemos por separado lo que le ocurre a cada biestable.

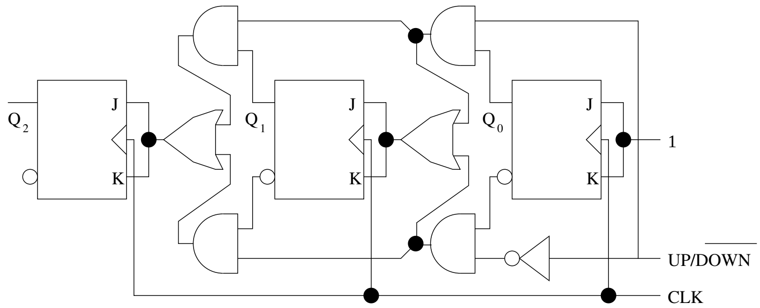


Figura 3.7: Esquema de un contador binario síncrono ascendente/descendente de 3 bits.

Pulso	Q_3	Q_2	Q_1	Q_0	Decodificador
0	0	0	0	0	$\overline{Q_3}\cdot\overline{Q_2}\cdot\overline{Q_1}\cdot\overline{Q_0}$
1	0	0	0	1	$\overline{Q_3}\cdot\overline{Q_2}\cdot\overline{Q_1}\cdot Q_3$
2	0	0	1	0	$\overline{Q_2}\cdot Q_1\cdot\overline{Q_0}$
3	0	0	1	1	$\overline{Q_2}\cdot Q_1\cdot Q_0$
4	0	1	0	0	$Q_2\cdot\overline{Q_1}\cdot\overline{Q_0}$
5	0	1	0	1	$Q_2\cdot\overline{Q_1}\cdot Q_0$
6	0	1	1	0	$Q_2\cdot Q_1\cdot\overline{Q_0}$
7	0	1	1	1	$Q_2\cdot Q_1\cdot Q_0$
8	1	0	0	0	$Q_3\cdot\overline{Q_1}\cdot\overline{Q_0}$
9	1	0	0	1	$Q_3\cdot\overline{Q_1}\cdot Q_0$

Tabla 3.1: Tabla de estados de un contador BCD.

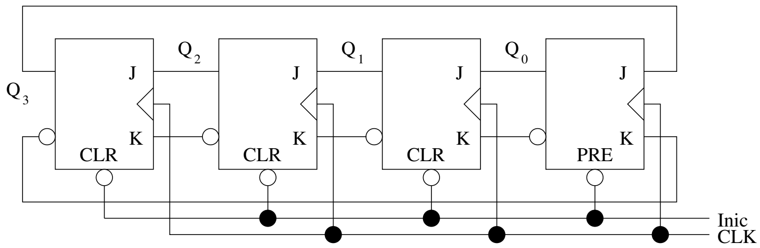


Figura 3.8: Esquema de un contador en anillo de 4 bits.

Pulso	Q_3	Q_2	Q_1	Q_0
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0

Tabla 3.2: Tabla de estados de un contador en anillo de 4 bits.

1. FF0 bascula en cada pulso de reloj, así que $J_0 = K_0 = 1$.
2. FF1 bascula si $Q_0 = 1$ y $Q_3 = 0$. Por tanto, $J_1 = K_1 = Q_0\overline{Q_3}$.
3. FF2 bascula si $Q_0 = Q1 = 1$. Así, $J_2 = K_2 = Q_0Q_1$.
4. FF3 bascula si $Q_0 = Q_1 = Q_2 = 1$ o si $Q_0 = Q_3 = 1$. En este caso, $J_3 = K_3 = Q_0Q_1Q_2 + Q_0Q_3$.

El esquema lógico, cronograma y tabla de estados de este contador se deja como práctica.

3.6 El contador en anillo

El contador en anillo es un registro de desplazamiento que tiene su salida conectada a la entrada. Normalmente se implementa con flip-flops con entradas de PRESET y CLEAR, conectados en cascada y disparados de forma síncrona. Por lo tanto, un contador en anillo es un contador síncrono. La figura 3.8 muestra un contador en anillo de 4 bits.

En contador en anillo funciona pasándose de flip-flop a flip-flop un único bit. Esto quiere decir que, en cualquier instante del proceso de cuenteo, sólo un flip-flop tiene su salida $Q = 1$. Esto provoca que el contador en anillo sea el contador más fácil de decodificar. De hecho, sabiendo que el flip-flop está a uno, conocemos en que estado se encuentra el contador. La tabla 3.2 presenta los estados por los que pasa nuestro contador de 4 bits, una vez inicializado.

El principal inconveniente del contador en anillo es que para procesar n estados necesita n flip-flops. Sin embargo, no necesita lógica añadida, ni para construir el siguiente estado, ni para decodificar el estado. Debido a ésto último, el contador en anillo es (junto con el contador Johnson) el más rápido que podemos construir, presentando una frecuencia máxima de funcionamiento

$$f_{max} = \frac{1}{t_{d_{FF}}}. \tag{3.3}$$

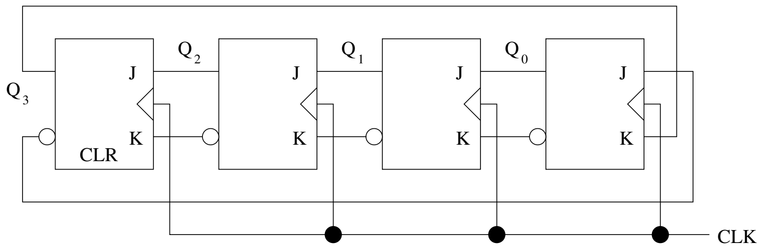


Figura 3.9: Esquema de un contador en Johnson de 4 bits.

Pulso	Q_3	Q_2	Q_1	Q_0	Decodificador
0	0	0	0	0	$\overline{Q_3 Q_0}$
1	0	0	0	1	$\overline{Q_1} Q_0$
2	0	0	1	1	$\overline{Q_2} Q_1$
3	0	1	1	1	$\overline{Q_3} Q_2$
4	1	1	1	1	$Q_3 Q_0$
5	1	1	1	0	$Q_1 \overline{Q_0}$
6	1	1	0	0	$Q_2 \overline{Q_1}$
7	1	0	0	0	$Q_3 \overline{Q_2}$

Tabla 3.3: Tabla de estados y lógica de decodificación de un contador Johnson de 4 bits.

3.7 El contador Johnson

El contador en anillo es elegantemente simple, pero utiliza los flip-flops antieconómicamente – recuérdese que con n biestables es posible codificar hasta 2^n estados –. El contador Johnson o *contador conmutado en cola* es una variación del contador en anillo que duplica el número de estados codificados, sin sacrificar su velocidad. Lo que si complica algo es la decodificación del estado.

La figura 3.9 presenta un contador Johnson de 4 bits. Como puede apreciarse, la diferencia con un contador en anillo es que ahora, en lugar de conectar Q_3 a J_0, y $\overline{Q_3}$ a K_0 conectamos $\overline{Q_3}$ a J_0 y Q_3 a K_0. Esto provoca que el biestable 3 cambie los ceros que le llegan por unos y viceversa. La tabla 3.3 presenta los estados alcanzados. Una agradable ventaja del contador Johnson respecto del contador en anillo es que no es necesario utilizar las entradas asíncronas para inicializar el contador – siempre y cuando, el estado inicial por defecto sea el 0000_2 –.

En consecuencia de lo expuesto, con n flip-flops, un contador Johnson es capaz de codificar $2n$ estados, y aunque la decodificación se complica, la velocidad de cuenteo es igual a la del contador en anillo.

3.8 Prácticas

3.8.1 Contador binario asíncrono ascendente

Montar el contador asíncrono mostrado en la figura 3.10. Rellenar el cronograma expuesto en la figura 3.11 y la tabla de estados 3.4. Indicar el módulo del

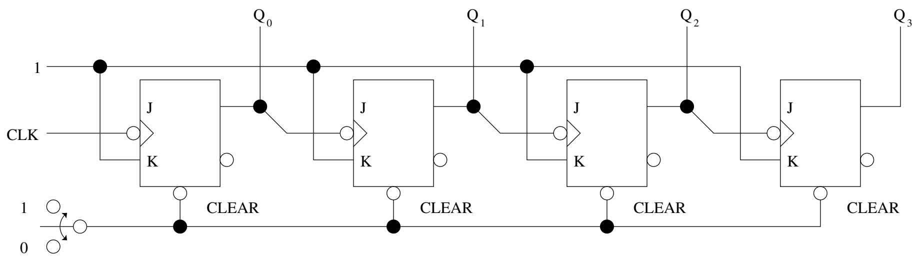


Figura 3.10: Esquema lógico de un contador binario asíncrono ascendente de 4 bits.

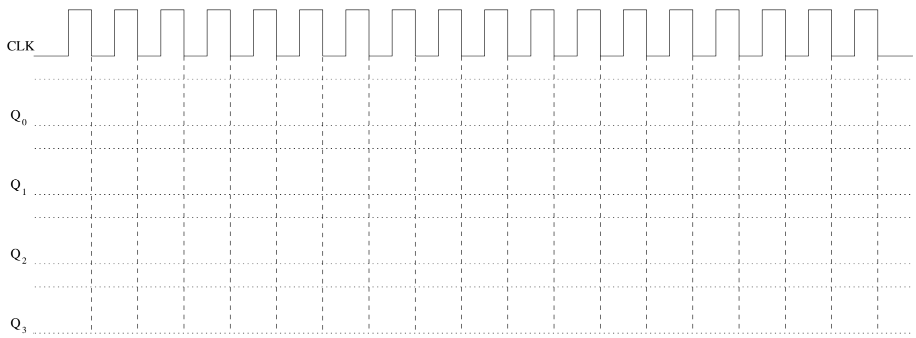


Figura 3.11: Cronograma de un contador binario asíncrono ascendente de 4 bits.

contador. Calcular f_{max} si $t_{d_{FF}} = 25$nsegs.

3.8.2 Contador binario asíncrono descendente

Con el objetivo de comprobar que un contador desdendente puede ser obtenido si utilizamos las $\overline{\mathrm{Q}}$'s en lugar de las Q's para calcular el estado siguiente, montar un contador binario asíncrono descendente de cuatro bits, a partir del montaje anterior. El esquema lógico se presenta en la figura 3.12. Rellenar el cronograma 3.13 y la tabla de estados 3.5. Calcula f_{max} utilizando los tiempos de retardo dados en la práctica anterior.

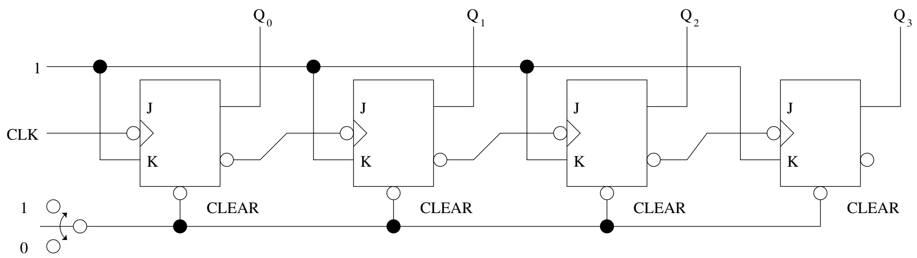


Figura 3.12: Esquema lógico de un contador asíncrono binario descendente de 4 bits.

Pulso	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Tabla 3.4: Tabla de estados de un contador asíncrono binario ascendente de 4 bits.

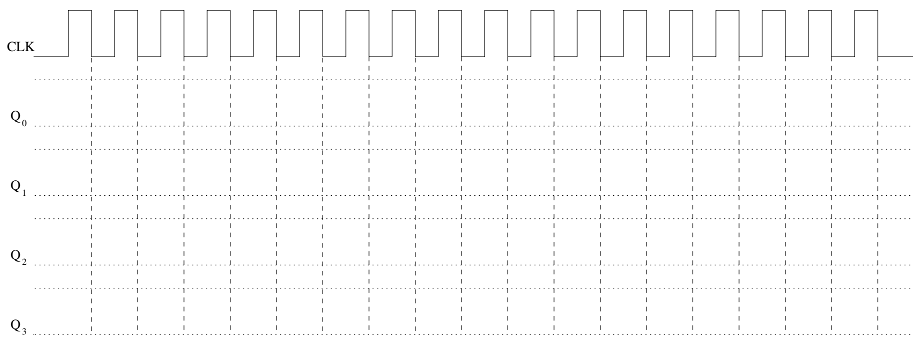


Figura 3.13: Cronograma de un contador asíncrono binario descendente de 4 bits.

Pulso	Q_3	Q_2	Q_1	Q_0
0	1	1	1	1
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Tabla 3.5: Tabla de estados del contador asíncrono binario descendente de 4 bits.

3.8.3 Contador asíncrono ascendente BCD

La figura 3.14 presenta el esquema lógico de un contador asíncrono BCD ascendente. Montarlo y rellenar el crongrama de la figura 3.15, junto con la tabla de estados 3.6. Teniendo en cuenta los tiempos dados para los biestables en prácticas anteriores y que $t_{d_{GATE}} = 10$nseg, calcular f_{max}.

3.8.4 Contador síncrono binario de 4 bits

Montar el circuito propuesto en la figura 3.5. Rellenar el cronograma de la figura 3.16 y la tabla de estados 3.7. Teniendo en cuenta los tiempos de retardo de los biestables y puertas indicados en prácticas anteriores, calcular f_{max}.

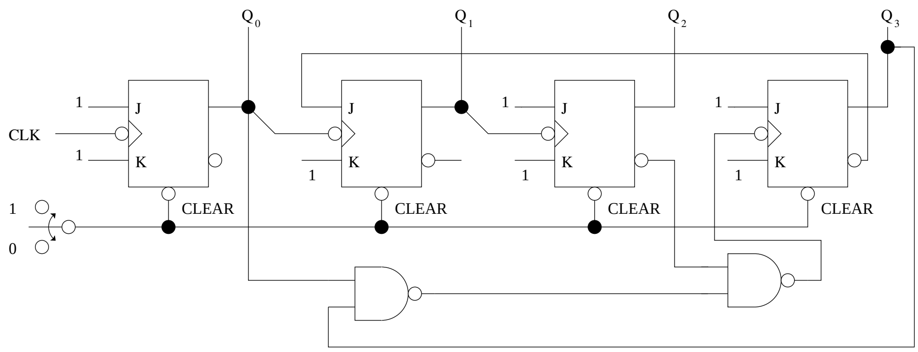


Figura 3.14: Esquema lógico de un contador asíncrono BCD ascendente.

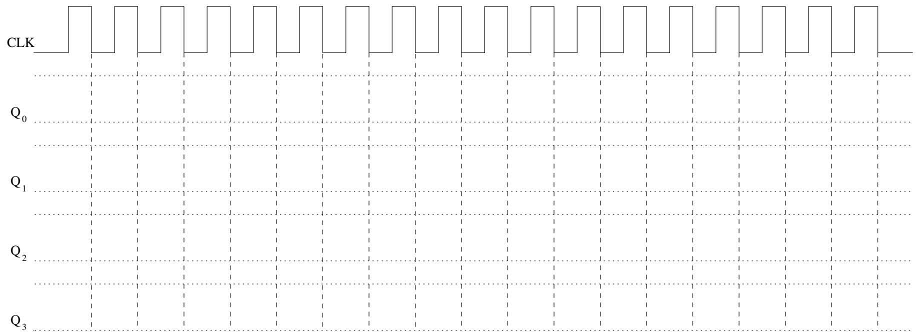


Figura 3.15: Cronograma de un contador asíncrono BCD ascendente.

Pulso	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1				
2				
3				
4				
5				
6				
7				
8				
9				

Tabla 3.6: Tabla de estados de un contador asíncrono BCD ascendente.

Pulso	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Tabla 3.7: Tabla de estados de un contador síncrono ascendente de 4 bits.

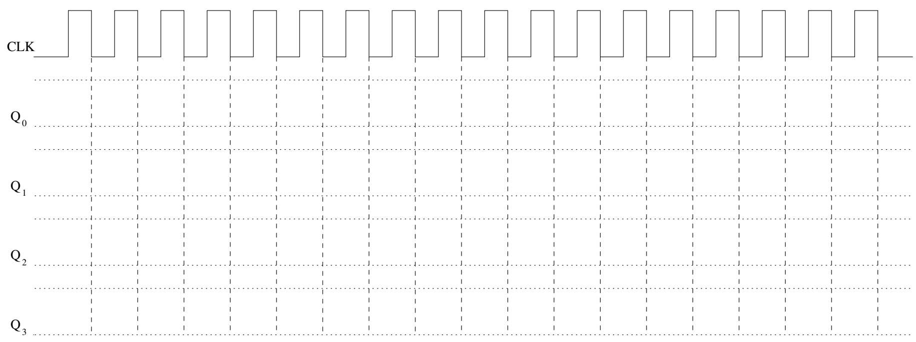


Figura 3.16: Cronograma de un contador síncrono ascendente de 4 bits.

3.8.5 Contador síncrono ascendente/descendente

Un contador ascendente/descendente (up/down) es aquel capaz de procesar en cualquier dirección a lo largo de una cierta secuencia. Un contador ascendente/descendente, algunas veces también denominado contador bidireccional, puede tener cualquier secuencia de estados especificada. Un contador binario de 3 bits que avanza en modo ascendente a través de la secuencia (0,1,2,3,4,5,6,7) y que luego pueda invertirse para recorrer la secuencia en sentido contrario (7,6,5,4,3,2,1,0) es un ejemplo de un modo de operación secuencial ascendente/descendente.

La figura 3.17 muestra un contador síncrono ascendente/descendente básico de 3 bits. El FF0 bascula con cada impulso de reloj. Luego las entradas J_0 y K_0 de FF0 son:

$$\mathrm{J}_0 = \mathrm{K}_0 = 1$$

Para la secuencia ascendente, Q_1 cambia de estado en el siguiente impulso de reloj cuando $\mathrm{Q}_0 = 1$. Para la secuencia descendente, Q_1 cambia en el siguiente impulso de reloj cuando $\mathrm{Q}_0 = 0$. Por lo tanto, las entradas J_1 y K_1 del FF1 tienen que ser igual a 1, para las condiciones expresadas en la siguiente ecuación:

$$\mathrm{J}_1 = \mathrm{K}_1 = (\mathrm{Q}_0 \cdot UP) + (\overline{\mathrm{Q}_0} \cdot DOWN)$$

Para la secuencia ascendente, Q_2 cambia de estado en el siguiente impulso de reloj cuando $\mathrm{Q}_0 = \mathrm{Q}_1 = 1$. Para la secuencia descendente, Q_2 cambia en el siguiente impulso de reloj cuando $\mathrm{Q}_0 = \mathrm{Q}_1 = 0$. Por lo tanto, las entradas J_2 y K_2 de FF2 tienen que ser igual a 1, para las condiciones expresadas en la siguiente ecuación:

$$\mathrm{J}_2 = \mathrm{K}_2 = (\mathrm{Q}_0 \cdot \mathrm{Q}_1 \cdot UP) + (\overline{\mathrm{Q}_0} \cdot \overline{\mathrm{Q}_1} \cdot DOWN)$$

Cada una de las condiciones para las entradas J y K de cada flip-flop produce una basculación en el punto apropiado de la secuencia del contador.

Realizar el montaje de la figura 3.17 y comprobar su funcionamiento. Completar el cronograma de la figura 3.18 y la tabla de estados 3.8.5 en concordancia con el cronograma realizado.

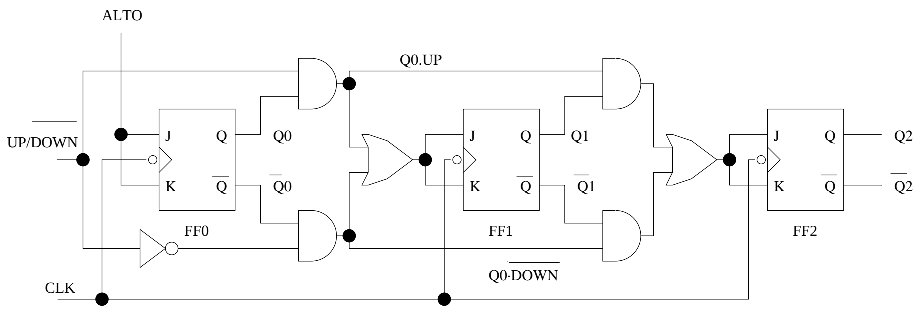


Figura 3.17: Esquema lógico de un contador UP/DOWN síncrono de 3 bits.

Pulso	UP/DOWN	Q_3	Q_2	Q_1
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				

Tabla 3.8: Tabla de verdad para el contador UP/DOWN síncrono de 3 bits.

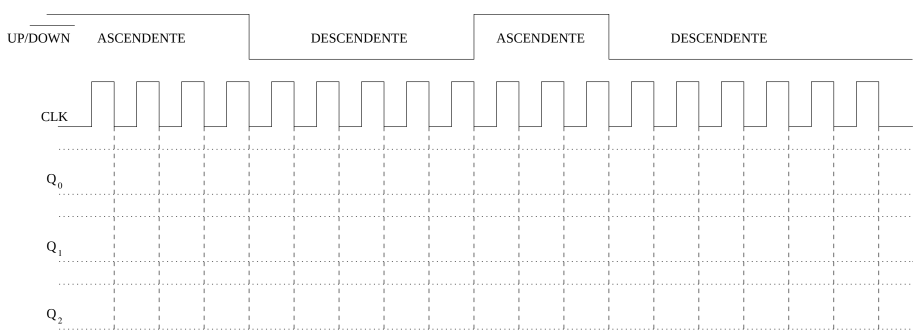


Figura 3.18: Cronograma de un contador UP/DOWN sícrono de 3 bits.

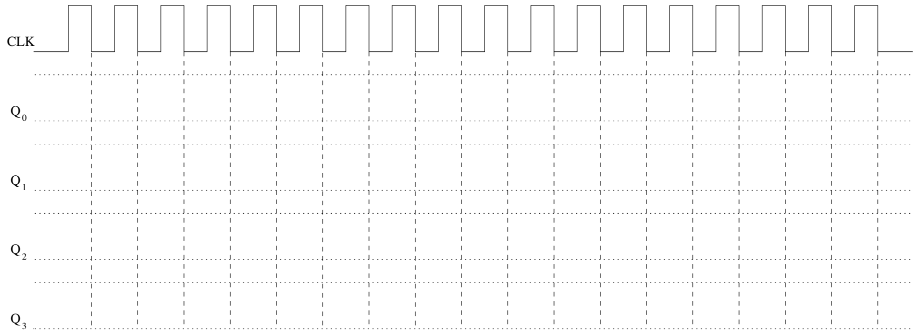


Figura 3.19: Cronograma de un contador síncrono BCD ascendente.

Pulso	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1				
2				
3				
4				
5				
6				
7				
8				
9				

Tabla 3.9: Tabla de estados de un contador síncrono BCD ascendente.

3.8.6 Contador síncrono BCD ascendente

Montar el circuito presentado en la sección 3.5. Dibujar su esquema lógico, rellenar la tabla de estados 3.9 y el cronograma de la figura 3.19. Calcular f_{max}.

3.8.7 El contador en anillo

El contador en anillo utiliza un flip-flop para cada estado de su secuencia. Tiene la ventaja de que no se requieren puertas de decodificación para realizar la conversión decimal, debido a que hay una salida para cada número decimal. La figura 3.20 muestra un contador en anillo de 4 bits utilizando flip-flops tipo D. Notar que hay que usar las entradas asíncronas PRESET y CLEAR para forzar el contador a un estado de la secuencia (el 0001_2 por ejemplo).

Realizar el montaje de la figura 3.20 e indicar el funcionamiento del contador. Completar la tabla de estados 3.8.7, hasta que se empiecen a repetir los estados. Completar el cronograma de la figura 3.21. Calcular f_{max}.

3.8.8 El contador Johnson

En un contador Johnson el complemento de la salida del último flip-flop se conecta a la entrada D del primer flip-flop (como es lógico se puede implementar

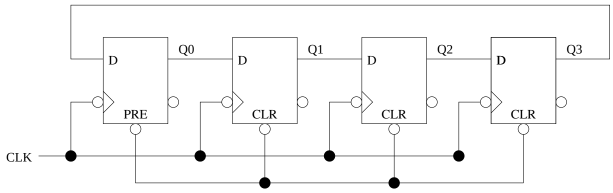


Figura 3.20: Esquema de un contador en anillo de 4 bits.

Pulso	Q_3	Q_2	Q_1	Q_0
1				
2				
3				
4				

Tabla 3.10: Tabla de estados de un contador en anillo de 4 bits.

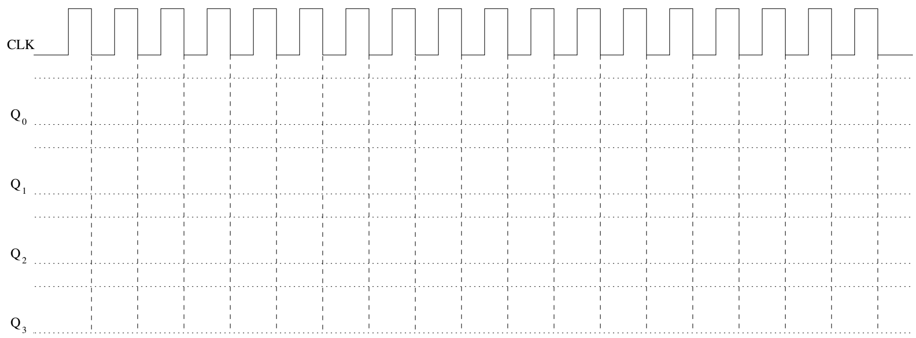


Figura 3.21: Cronograma de un contador en anillo de 4 bits.

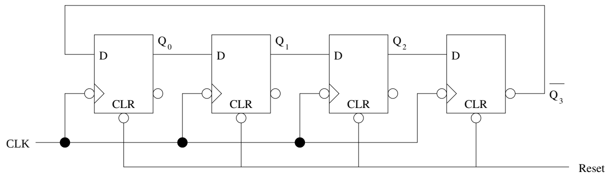


Figura 3.22: Esquema de un contador Johnson de 4 bits.

Pulso	Q_3	Q_2	Q_1	Q_0
0				
1				
2				
3				
4				
5				
6				
7				

Tabla 3.11: Tabla de estados de un contador Johnson de 4 bits.

con otro tipo de flip-flop), tal y como se muestra en la figura 3.22. Esta realimentación permite generar una secuencia de estados características. En general un contador Johnson generará un módulo de $2n$, siendo n el número de etapas del contador.

Realizar el montaje de la figura 3.22 y comprobar su funcionamiento. Completar la tabla de verdad 3.8.8. Completar el cronograma de la figura 3.23. Calcular f_{max}.

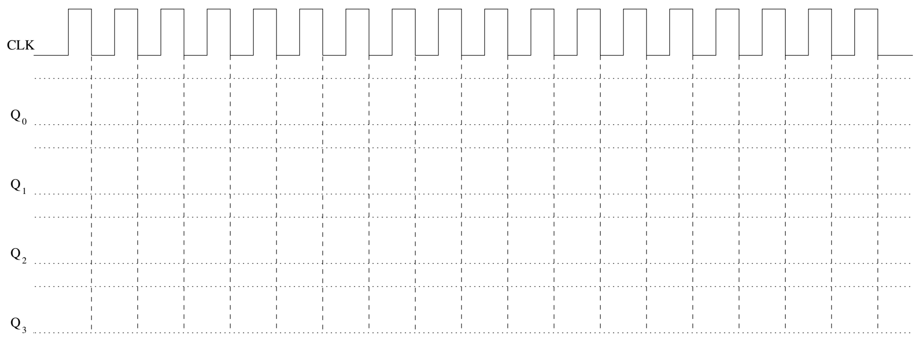


Figura 3.23: Cronograma de un contador Johnson de 4 bits.

Capítulo 4

Diseño de circuitos secuenciales

La metodología empleada para desarrollar los contadores síncronos expuestos en el tema anterior no es adecuado para el desarrollo de cualquier sistema de recuento. Existe una metodología llamada *Diseño de circuitos secuenciales síncronos* mucho más estudiada que, aunque es más laboriosa para esquemas sencillos, también es mucho más automática. Con ella se pueden generar contadores, generadores de secuencias, secuenciadores, registros, etc. Estudiaremos esta nueva forma de desarrollo mediante ejemplos.

4.1 Diseño de un contador BCD síncrono

Todo circuito secuencial – también llamado *máquina de estados* – está formado por una etapa de lógica combinacional y una etapa de memoria (típicamente construida con flip-flops), tal y como se muestra en la figura 4.1. Nuestro objetivo básicamente es encontrar la lógica combinacional, ya que la zona de memoria es prácticamente invariable – sólo hay que indicar cuantos biestables son necesarios –.

4.1.1 Diagrama de estados

Describe gráficamente el contador (en este caso), indicando cual es el estado siguiente en función del estado actual y de las entradas (que no existen para

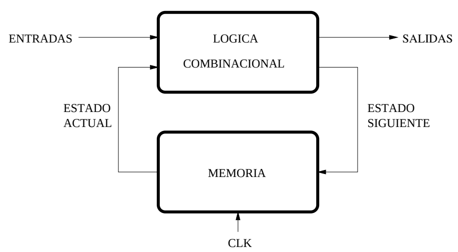


Figura 4.1: Esquema genérico de un circuito secuencial.

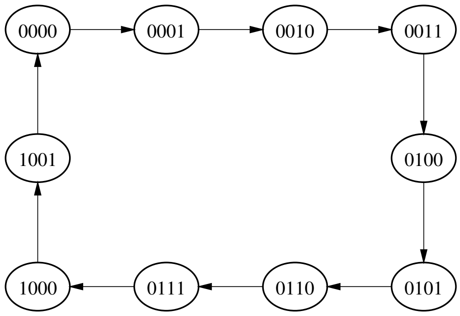


Figura 4.2: Diagrama de estados de un contador BCD.

Estado Actual					Estado Siguiente				Entradas a Biestables			
Pulso	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_3K_3	J_2K_2	J_1K_1	J_0K_0
0	0	0	0	0	0	0	0	1	0 X	0 X	0 X	1 X
1	0	0	0	1	0	0	1	0	0 X	0 X	1 X	X 1
2	0	0	1	0	0	0	1	1	0 X	0 X	X 0	1 X
3	0	0	1	1	0	1	0	0	0 X	1 X	X 1	X 1
4	0	1	0	0	0	1	0	1	0 X	X 0	0 X	1 X
5	0	1	0	1	0	1	1	0	0 X	X 0	1 X	X 1
6	0	1	1	0	0	1	1	1	0 X	X 0	X 0	1 X
7	0	1	1	1	1	0	0	0	X 1	X 1	X 1	X 1
8	1	0	0	0	1	0	0	1	X 0	0 X	0 X	1 X
9	1	0	0	1	0	0	0	0	X 1	0 X	0 X	X 1

Tabla 4.1: Tabla de transiciones del contador síncrono BCD.

este caso). La figura 4.2 muestra el diagrama de estados del contador. Notar que como no existen entradas, sólo hay una posible alternativa como estado siguiente dado un estado actual.

4.1.2 Tabla de transiciones

Indica para cada par estado actual/entrada, cual es el estado siguiente y cuales son las entradas a los flip-flops. Estas entradas están generadas por la parte combinacional de nuestra máquina de estados. La tabla 4.1 muestra la tabla de transiciones para el contador BCD. Las entradas de los biestables se calculan mirando la tabla de verdad del flip-flop JK escrita al revés (ver tabla 4.2).

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

Tabla 4.2: Tabla de verdad del flip-flop JK, en función del estado actual y el estado siguiente.

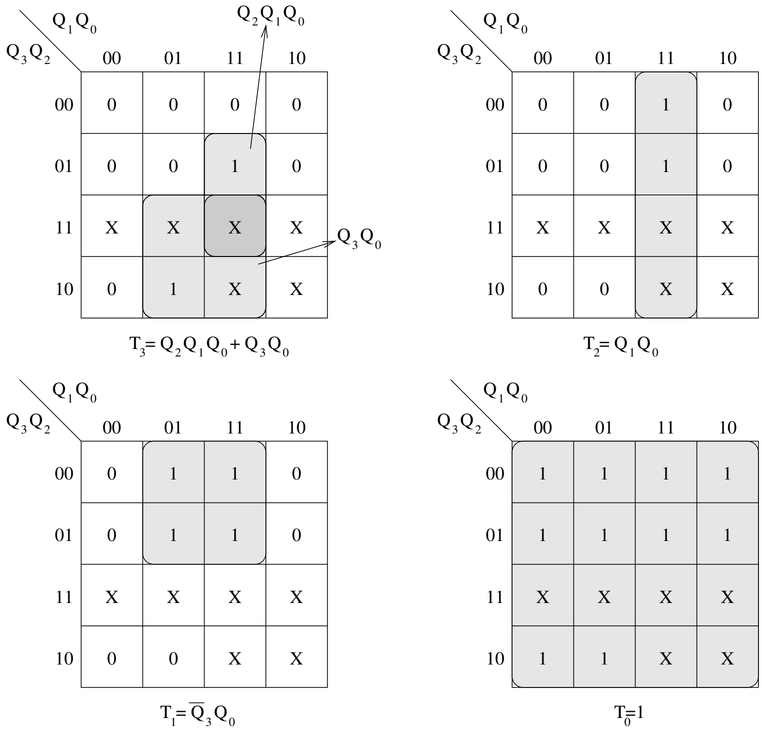


Figura 4.3: Mapas de Karnaugh para la minimización de un contador BCD síncrono ascendente.

4.1.3 Minimización de la lógica combinacional

Usaremos mapas de Karnaugh. Para este ejemplo en concreto, podemos hacer $T_3 = J_3 = K_3$, $T_2 = J_2 = K_2$, $T_1 = J_1 = K_1$ y $T_0 = J_0 = K_0$. Así, los mapas resultantes son los mostrados en la figura 4.3.

4.1.4 Implementación

La última fase del diseño consiste en dibujar, siguiendo la filosofía mostrada en la figura 4.1, la estructura interna del contador síncrono BCD ascendente. Ver figura 4.4.

4.2 Diseño de un contador síncrono BCD ascendente/descendente

4.2.1 Diagrama de estados

Determina el estado siguiente en función del estado actual y de la entrada A. Si $A = 1$, el contador cuenta ascendentemente y viceversa. Ver la figura 4.5.

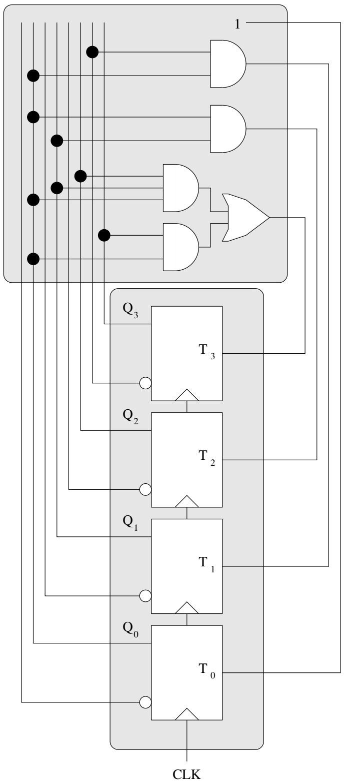


Figura 4.4: Estructura interna del contador síncrono BCD ascendente.

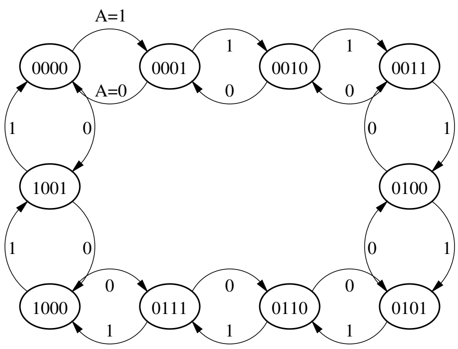


Figura 4.5: Diagrama de estados de un contador síncrono BCD ascendente/descendente.

Estado Actual					Estado Siguiente							
					$A=0$				$A=1$			
Pulso	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	1	0	0	1	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	1	0
2	0	0	1	0	0	0	0	1	0	0	1	1
3	0	0	1	1	0	0	1	0	0	1	0	0
4	0	1	0	0	0	0	1	1	0	1	0	1
5	0	1	0	1	0	1	0	0	0	1	1	0
6	0	1	1	0	0	1	0	1	0	1	1	1
7	0	1	1	1	0	1	1	0	1	0	0	0
8	1	0	0	0	0	1	1	1	1	0	0	1
9	1	0	0	1	1	0	0	0	0	0	0	0

Estado Actual					Entradas a Biestables							
					$A=0$				$A=1$			
Pulso	Q_3	Q_2	Q_1	Q_0	T_3	T_2	T_1	T_0	T_3	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1	0	0	0	1
1	0	0	0	1	0	0	0	1	0	0	1	1
2	0	0	1	0	0	0	1	1	0	0	0	1
3	0	0	1	1	0	0	0	1	0	1	1	1
4	0	1	0	0	0	1	0	1	0	0	0	1
5	0	1	0	1	0	0	1	1	0	0	1	1
6	0	1	1	0	0	0	0	1	0	0	0	1
7	0	1	1	1	0	0	1	1	1	1	1	1
8	1	0	0	0	1	1	1	1	0	0	0	1
9	1	0	0	1	0	0	0	1	1	0	0	1

Tabla 4.3: Tabla de transiciones del contador síncrono BCD ascendente/descendente.

4.2.2 Tabla de transiciones

Utilizaremos flip-flops JK con las entradas cortocircuitadas (flip-flops T). Ver la tabla de transiciones 4.3. Para rellenarla hemos utilizado la tabla de verdad 4.4 del flip-flop T en función del estado actual y del estado siguiente.

4.2.3 Minimización de la lógica combinacional

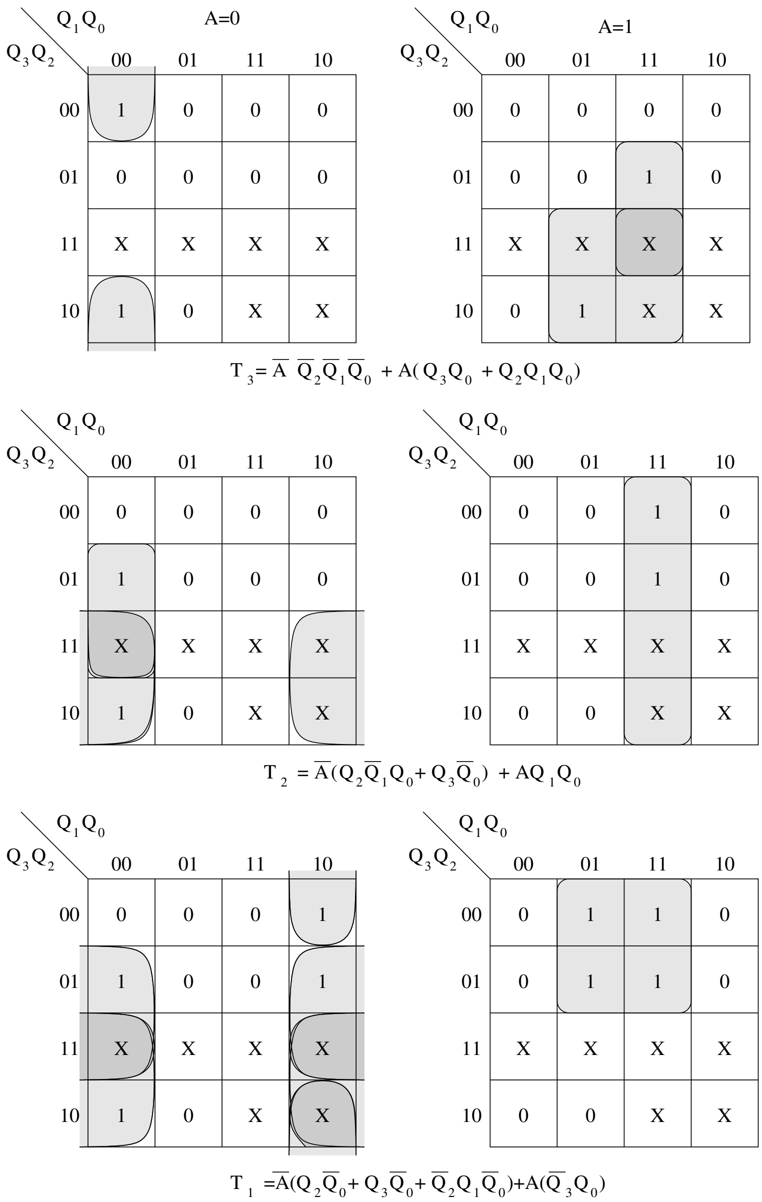
Utilizamos de nuevo los mapas de Karnaugh (ver figura 4.3).

4.2.4 Implementación

La última fase del diseño consiste en dibujar, siguiendo la filosofía mostrada en la figura 4.1, la estructura interna del contador síncrono BCD ascendente/descendente. Ver figura 4.7.

Q^n	Q^{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 4.4: Tabla de verdad del flip-flop T, en función del estado actual y el estado siguiente.



$$T_3 = \overline{A}\,\overline{Q}_2\overline{Q}_1\overline{Q}_0 + A(Q_3Q_0 + Q_2Q_1Q_0)$$

$$T_2 = \overline{A}(Q_2\overline{Q}_1Q_0 + Q_3\overline{Q}_0) + AQ_1Q_0$$

$$T_1 = \overline{A}(Q_2\overline{Q}_0 + Q_3\overline{Q}_0 + \overline{Q}_2Q_1\overline{Q}_0) + A(\overline{Q}_3Q_0)$$

Figura 4.6: Mapas de Karnaugh para la minimización de un contador BCD síncrono ascendente/descendente.

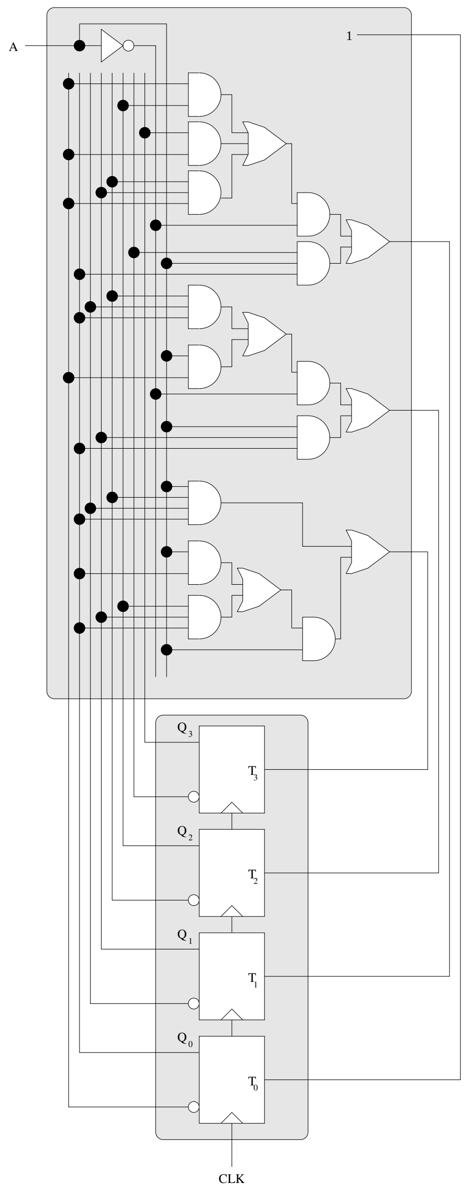


Figura 4.7: Estructura interna del contador síncrono BCD ascendente/descendente.

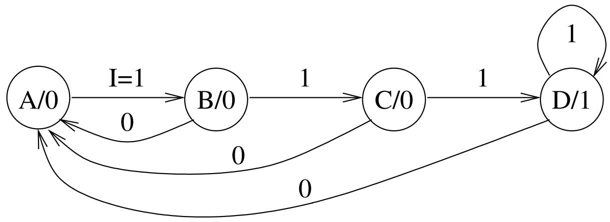


Figura 4.8: Diagrama de estados de un detector de tres unos consecutivos.

I	E.A.	E.S.
0	A	A
0	B	A
0	C	A
0	D	A
1	A	B
1	B	C
1	C	D
1	D	D

Tabla 4.5: Tabla de transiciones del detector de secuencias de tres unos consecutivos.

4.3 Detector de sercuencias de tres unos consecutivos

Vamos a diseñar un circuito secuencial síncrono capaz de detectar cuando le llegan al menos tres unos consecutivos por su entrada I. Siempre que esto ocurra, la salida O del circuito valdrá uno. Por ejemplo:

$$\begin{aligned} I &= \quad 0110111011111X_2 \\ O &= \quad ?0000001000111_2 \end{aligned}$$

donde, X significa irrelevante (da igual que sea 0 o 1) y ? significa desconocido (imposible de determinar el valor del bit).

4.3.1 Digrama de estados

Asociaremos la salida al estado. Debido a ésto, dentro de cada nodo del diagrama escribiremos un par (estado/salida). La figura 4.8 presenta un diagrama de estados adecuado a la especificación del problema.

4.3.2 Tabla de transiciones

Escribiremos la tabla de transiciones sin realizar la asignación de estados. La tabla 4.5 la presenta.

Haciendo la asignación de estados presentada en la tabla 4.6, obtenemos la tabla de transiciones 4.7. E.A.=Estado Actual, E.S.=Estado Siguiente y E.B.=Entradas Biestables.

Estado	Q_1Q_0
A	00
B	01
C	10
D	11

Tabla 4.6: Asignación de estados del detector de secuencias de 3 unos consecutivos.

I	E.A.	E.S.	E.B.	
	$Q_1^nQ_1^n$	$Q_1^{n+1}Q_0^{n+1}$	J_1K_1	J_0K_0
0	00	00	0X	0X
0	01	00	0X	X1
0	10	00	X1	0X
0	11	00	X1	X1
1	00	01	0X	1X
1	01	10	1X	X1
1	10	11	X0	1X
1	11	11	0X	0X

Tabla 4.7: Tabla de transiciones del detector de secuencias de 3 unos consecutivos.

4.3.3 Minimización de la lógica combinacional

Puesto que la salida sólo depende del estado actual, es fácil ver que $O = Q_1Q_0$. Las entradas a los biestables se minimizan utilizando mapas de Karnaugh tal y como se indica en la figura 4.9.

4.3.4 Implementación

Por último dibujamos la estructura lógica del detector siguiendo la filosofía de diseño de la máquina de estados genérica. La figura 4.10 la presenta.

4.4 Detector de sercuencias de tres unos consecutivos (Mealy)

En la versión anterior, la salida viene determinada por el estado actual. Cuando ésto ocurre, se dice que el circuito es de Moore. Sin embargo, es posible hacer que la salida dependa tanto del estado actual como de la entrada, consiguéndose en ocasiones un ahorro de circuitería. Si el diseño se realiza según esta segunda filosofía, diremos haber construido un circuito de Mealy.

4.4.1 Diagrama de estados

Ahora la salida no aparece dentro de los nodos del diagrama, sino en las aristas. Podemos hablar por tanto de pares (entrada/salida). La figura 4.11 presenta un diagrama de estados.

I \ Q_1Q_0	00	01	11	10
0	0	0	X	X
1	0	1	X	X

$J_1 = I\ Q_0$

I \ Q_1Q_0	00	01	11	10
0	X	X	1	1
1	X	X	0	0

$K_1 = \bar{I}$

I \ Q_1Q_0	00	01	11	10
0	0	X	X	0
1	1	X	X	1

$J_0 = I$

I \ Q_1Q_0	00	01	11	10
0	X	1	1	X
1	X	1	0	X

$K_0 = \bar{I} + \overline{Q_1} = \overline{IQ_1}$

Figura 4.9: Mapas de Karnaugh para minimizar el tector de secuencias de tres o más unos consecutivos.

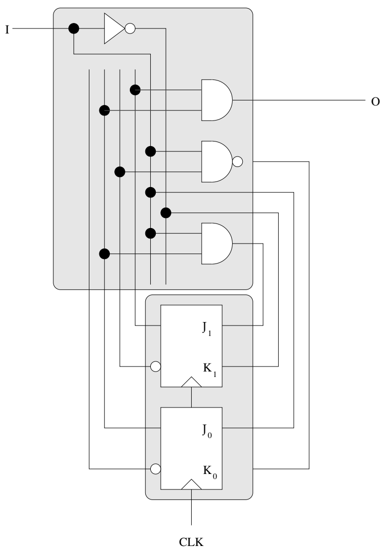


Figura 4.10: Esquema lógico del detector de secuencias de tres o más unos consecutivos.

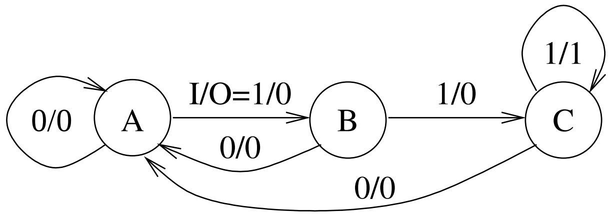


Figura 4.11: Diagrama de estados de un detector de tres unos consecutivos. Versión de Mealy.

I	E.A.	O	E.S.
0	A	0	A
0	B	0	A
0	C	0	A
1	A	0	B
1	B	0	C
1	C	1	C

Tabla 4.8: Tabla de transiciones del detector de secuencias de tres unos consecutivos. Versión de Mealy.

4.4.2 Tabla de transiciones

En la tabla de transiciones aparece ahora una columna nueva etiquetada por O y que identifica la salida del circuito. Ver tabla 4.5.

Haciendo la asignación de estados presentada en la tabla 4.9, obtenemos la tabla de transiciones 4.10.

4.4.3 Minimización de la lógica combinacional

Utilizamos los mapas de karnaugh expuestos en la figura 4.12.

4.4.4 Implementación

Por último dibujamos la estructura lógica del detector siguiendo la filosofía de diseño de la máquina de estados genérica. La figura 4.13 la presenta.

Estado	Q_1Q_0
A	00
B	01
C	10

Tabla 4.9: Asignación de estados del detector de secuencias de 3 unos consecutivos. Versión de Mealy.

I	E.A.	O	E.S.	E.B.	
	$Q_1^nQ_1^n$		$Q_1^{n+1}Q_0^{n+1}$	J_1K_1	J_0K_0
0	00	0	00	0X	0X
0	01	0	00	0X	X1
0	10	0	00	X1	0X
1	00	0	01	0X	1X
1	01	0	10	1X	X1
1	10	1	10	X0	0X

Tabla 4.10: Tabla de transiciones del detector de secuencias de 3 unos consecutivos. Versión de Mealy.

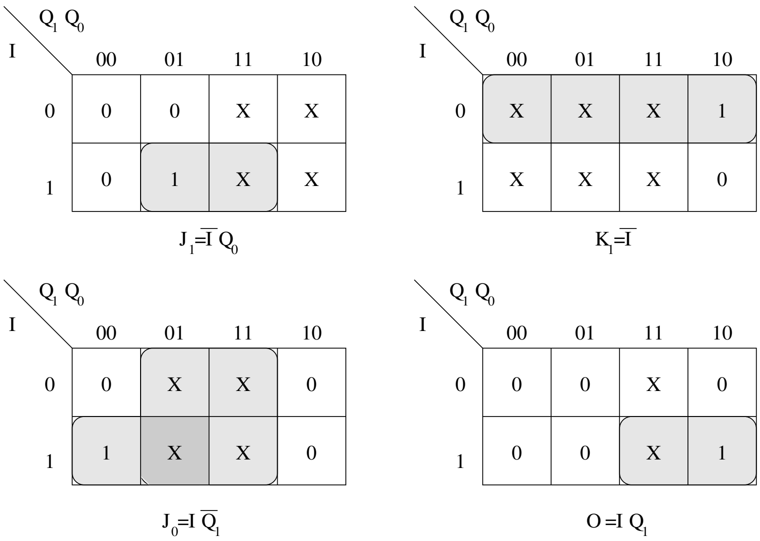


Figura 4.12: Mapas de Karnaugh para minimizar el tector de secuencias de tres o más unos consecutivos. Versión de Mealy.

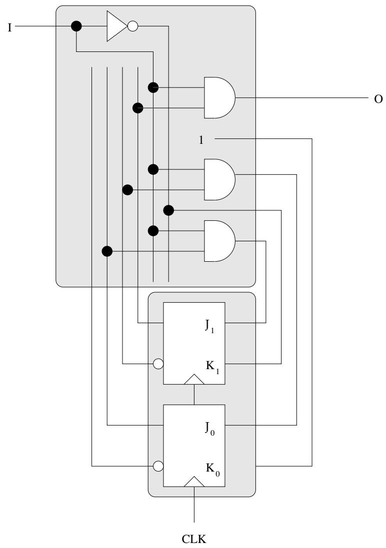


Figura 4.13: Esquema lógico del detector de secuencias de tres o más unos consecutivos. Versión de mealy.

4.5 Prácticas

4.5.1 Contador de estados impares

Diseñar e implementar físicamente un contador módulo 8 que sólo contemple estados impares (1,3,5, etc). Utilizar el tipo de biestable más conveniente. Calcular f_{max}.

4.5.2 Generador de secuencia $3, 4, 5, 6, 7$

Diseñar (diagrama de estados, tabla de transiciones y minimización por Karnaugh) e implementar físicamente un circuito digital secuencial que pase por los estados $011_2 \rightarrow 100_2 \rightarrow 101_2 \rightarrow 110_2 \rightarrow 111_2 \rightarrow 011_2$. Téngase en cuenta que si por alguna circunstancia el generador no está en uno de los estados previstos en el ciclo, es necesario llevarlo a cualquier de ellos. Utilizar los biestables que sean más convenientes.

4.5.3 Analizador de entradas

Diseñar (usando los métodos de Moore y de Mealy) e implementar físicamente un circuito digital secuencial que dadas tres entradas digitales X_0, X_1 y X_2, pase por los estados $A = 00$, $B = 01$, $C = 10$ y $D = 11$, siguiendo el diagrama de flujo propuesto en la figura 4.14. Utilizar tipo de biestable más conveniente.

4.5.4 Detector de secuencias de ceros solapadas

Diseñar (usando los métodos de Moore y de Mealy) e implementar físicamente un circuito secuencial digital que sea capaz de indicar cuando en su entrada tenemos tres ceros seguidos. Es válido solapar secuencias. Calcular f_{max}.

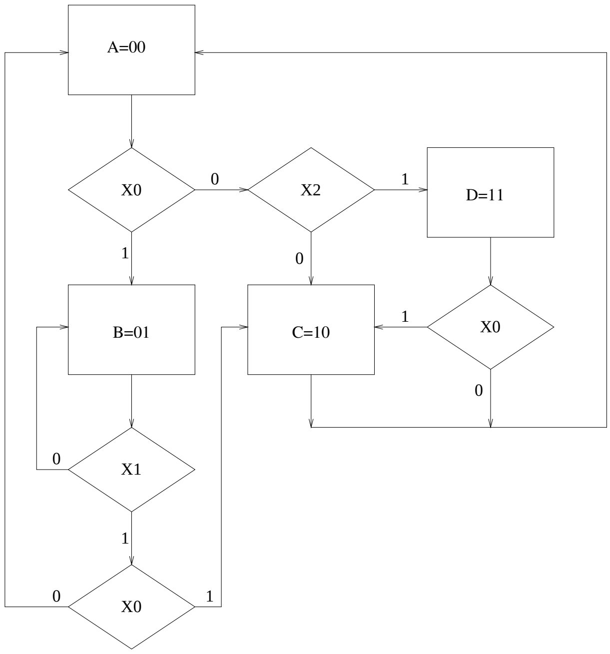


Figura 4.14: Diagrama de flujo de un analizador de entradas.

4.5.5 Detector de secuencias de ceros no solapadas

Diseñar (usando los métodos de Moore y de Mealy) e implementar físicamente un circuito secuencial digital que sea capaz de indicar cuando en su entrada tenemos tres ceros seguidos. NO es válido solapar secuencias. Calcular f_{max}.

4.5.6 Detector de secuencias de pares de bits alternos

Diseñar (usando los métodos de Moore y de Mealy) e implementar físicamente un circuito digital secuencial que tiene una entrada X y una salida Z, de forma que $Z = 1$ si en la entrada X aparecen dos bits idénticos consecutivos y si los dos bits precedentes también son iguales entre si, pero distintos al bit de entrada actual. Por ejemplo:

$$\begin{aligned} X = &\quad 00011001110011_2 \\ Z = &\quad ??001010100101_2 \end{aligned}$$

Utilizar el biestable que se crea más conveniente. ? significa irrelevante. Calcular f_{max}.

4.5.7 Detector de eventos

Este vez tienes que construir un circuito que va a funcionar en una central nuclear. En concreto, deberás averiguar cuando ocurren 5 eventos de forma consecutiva, y si ocurren, debes hacer sonar una alarma. Llamemos a los 5 eventos, E_0, E_1, E_2, E_3 y E_4. Ejemplos:

$E_0E_1E_2E_3E_4$	Suena la alarma.
$E_0E_2E_1E_3E_4$	No suena.
$E_1E_2E_3E_4E_0$	No suena.

Has de tener en cuenta que, siempre y cuando se rompa la secuencia porque ocurre un evento inesperado, debes reinicializar la cuenta. Realiza un deserrollo para las dos metodologías conocidas: Moore y Mealy. Tienes plena libertad para utilizar el número y tipo de flip-flops que consideres necesario. Calcular f_{max}.

4.5.8 Codificador unario

Un código unario es aquel que tiene tantos unos como indica el índice del símbolo a codificar y a continuación un 0. Seguidamente se muestran varios ejemplos.

Símbolo	Código
0	0
1	10
2	110
3	1110
4	11110
5	111110
6	1111110
7	11111110

Diseñar un codificador unario, que codifique símbolos con índices comprendidos entre 0 y 7. Calcular f_{max}.

4.5.9 Decodificador unario

Diseñar el decodificador unario que permita decodificar el código anterior. Calcular f_{max}.