

CONTADOR ASINCRÓNICO MEDIANTE LA IMPLEMENTACIÓN DE FLIP FLOPS TIPO D (julio de 2020)

Bryan A. Carrillo, Samir B. Guzmán, Bryan S. Hernández
bacarrillo@espe.edu.ec sbguzman@espe.edu.ec bshernandez@espe.edu.ec

Resumen – El presente artículo tiene como objetivo principal la realización del diseño de un circuito contador síncrono de cuatro (4) bits que presente una secuencia de números dada. Este diseño permite aplicar los conocimientos adquiridos en cuanto al manejo de los circuitos FLIP FLOP tipo D de modo asíncrono, a la elaboración de diseños a través de la evaluación lógica de datos de entrada y también retomará el concepto del código BCD para la presentación de resultados a través de dos (2) Displays.

Será importante verificar en la simulación si el hecho de realizar una implementación síncrona para este contador reduce los tiempos de retardo en la propagación de cada FLIP FLOP en comparación con la implementación asíncrona del circuito anterior.

En la etapa de implementación del circuito obtenido en el desarrollo teórico se debe tener el conocimiento previo de los componentes a utilizar, así como de sus propiedades. También existió una etapa de pruebas en la cual se determine el éxito del diseño o la reevaluación del mismo.

Índice de Términos – FLIP-FLOP, asíncrono, código BCD, bits

I. INTRODUCCIÓN

En la actualidad los sistemas digitales son muy utilizados y variados para diferentes aplicaciones las cuales se ven reflejadas en su gran mayoría en la industria y en los equipos electrónicos. Por esto es necesario conocer a grandes rasgos sus aplicaciones y cómo funcionan los diferentes tipos de sistemas digitales, para así nosotros poder tener la capacidad de implementar y desarrollar estos sistemas sin ninguna clase de dificultad. En este artículo daremos a conocer una de las aplicaciones de los flip-flop la cual es contar. A continuación, mostraremos una configuración específica de un contador binario de 4 bits ascendente. Donde se mostrará el funcionamiento de dicho circuito el cual se encuentra constituido por el circuito integrado 74HC74. El diseño e implementación se encontrará respaldado por la simulación tanto en Proteus como en la plataforma Tinkercad del contador asíncrono de 4 bits que cuenta de 0-15.

II. OBJETIVOS

Objetivo general

Implementar y comprobar el funcionamiento de un circuito contador ascendente asíncrono, implementado con flip-flop tipo D.

Objetivos específicos

Investigar en funcionamiento de flip flops tipo D.

Comprobar el funcionamiento del diseño del contador mediante la simulación en proteus e implementada en Tinkercad (laboratorio virtual).

Convertir las salidas de cada flip flop de código BCD a decimal y mostrarlo en dos displays de 7 segmentos.

III. ESTADO DE ARTE

El circuito asíncrono realizado por los Ingenieros Rubén González; Marco Bardón y José Antonio Salcines; primero nos define que es un contador de la siguiente manera:

Un contador es un circuito en el que sus salidas siguen una secuencia fija que cuando acaba vuelve a empezar, o circuitos que reciben sus datos en forma serial ordenados en distintos intervalos de tiempo. Los pulsos de entrada pueden ser pulsos de reloj u originarse en una fuente externa y pueden ocurrir a intervalos de tiempo fijos o aleatorios. El número de salidas limita el máximo número que se puede contar. En nuestro estudio y para nuestro análisis se estudiará los contadores asíncronos que en el contexto de los autores nos explica que son: Las salidas de cada flip-flop sirven de entrada CP para disparar otro flip-flop. El primer biestable tendrá una entrada de tipo asíncrono, es decir que se acertará de forma aleatoria y cuando lo haga el circuito realizará una cuenta. El resto del tiempo, los flip-flops no cambiarán su estado presente. En su diseño ellos pusieron como ejemplo lo siguiente; un contador asíncrono modulo 10: y los pasos para resolver son los siguientes. Paso 1: elegir un contador ascendente de 4 bits (de 0 a 15); paso 2: Detectar el 10 (1010 en binario) con una compuerta NAND y como último paso Reset de todos los biestables cuando ocurra la detección.

(González, Bardón, & Salcines. (2019). Contadores. Manzanom.

<https://personales.unican.es/manzanom/Planantiguo/EDigitalI/CONTG5.pdf>)

El sitio web Wilaeba Electrónica nos dan muchos puntos a tomar en cuenta además de ejemplos para lograr trabajar con distintos flip-flops incluido con el flip-flop que se desea trabajar que es el 74hc74. Los puntos a tomar que nos otorga este sitio web para nuestros diseños son los siguientes puntos:

- Un contador asíncrono tiene como principal característica que cada flip flop que lo compone tiene diferente señal de reloj (clk).
- El temporizador 555 está configurado a una frecuencia de 1 Hz, es decir que el contador aumentará de valor cada segundo.
- Los integrados usados para los contadores con flip flops JK serán el cd4027 y 74hc112, y para los contadores con flip flops D serán el cd4013 y el 74hc74.

Los integrados cd40 son tecnología cmos, y los integrados 74hc son tecnología ttl. Wilaeba Electrónica. (2017, 24 septiembre).

Contador asíncrono de 4 bits ascendente.

<https://wilaebaelectronica.blogspot.com/2017/08/contador-asincrono-de-4-bits-ascendente.html>

IV. MARCO TEÓRICO

Flip-flops

El flip flop es el nombre común que se le da a los dispositivos de dos estados (biestables), que sirven como memoria básica para las operaciones de lógica secuencial [1]

Clasificación

- Asíncronico Sólo tienen entradas de control. El más empleado es el flip flop RS.
- Síncronico Además de las entradas de control necesita una entrada sincronismo o de reloj.

Características

- Asumen solamente uno de dos posibles estados de salida
- Tienen un par de salidas que son complemento una de la otra.
- Tienen una o más entradas que pueden causar que el estado del Flip-Flop cambie.

Contadores asincrónicos

Contador asincrónico. Cada salida del flip-flop sirve como señal de entrada CLK para el siguiente flip-flop, estos contadores no cambian de estado todos juntos por lo que se dice que no están en sincronía, solo el primer flip flop responde a los pulsos del reloj, luego para que al segundo flip-flop responda debe esperar que el primer flip-flop cambie de estado, y para que el tercer flip-flop se complemente debe esperar que el segundo flip-flop cambie de estado, y así sucesivamente con los demás flip-flop. [2]

Flip flop tipo D

Es uno de los FF más sencillos. Su función es dejar pasar lo que entra por D, a la salida Q, después de un pulso del reloj.

Entradas

Clear inicializa Q en cero sin importar entradas o reloj

Reset inicializa Q en 1 sin importar entradas o reloj

Ck entrada de reloj

V. LISTA DE COMPONENTES

- Simulador Proteus version 8.9.
- Laboratorio virtual Tinkercad.

VI. MAPA DE VARIABLES

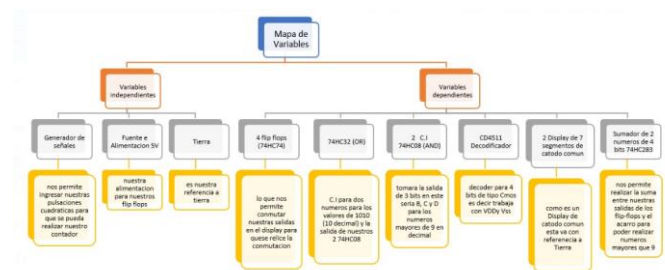


Fig. 1. Mapa de variables

VII. EXPLICACIÓN DEL CÓDIGO FUENTE

Contador en código binario.

En esta etapa es necesario indicar que se utilizará un generador de señal de reloj (CLK) para los FLIP FLOP (FF), de igual forma usaremos una frecuencia aproximada de 1 Hz dados los valores de las resistencias R_1 y R_2 (330 Ohmios Ω). Tomaremos en cuenta que:

- Un contador asíncrono tiene como principal característica que cada flip flop que lo compone tiene diferente señal de reloj (clk).
- El temporizador está configurado a una frecuencia de 1 Hz, es decir que el contador aumentará de valor cada segundo.
- Los integrados usados para los contadores con flip flops D serán el CD4013 y el 74hHC74. En nuestro caso usaremos el integrado 74HC74.

Para empezar nuestro análisis tendremos que plantear los estados, en este caso de 4 bits será de 0000 a 1111 es decir un conteo de de 0 a 15.

En esta parte vemos como se pasa de un estado al otro nuestra cuenta, lo que nos indica que las salidas de nuestros flip flops tendran una salida de 0 o 1.

Estado presente						Estado futuro					
Decimal	Q3	Q2	Q1	Q0		Decimal	Q3	Q2	Q1	Q0	
0	0	0	0	0		0	0	0	0	1	
1	0	0	0	0	1	1	0	0	1	0	
2	0	0	0	1	0	2	0	0	1	1	
3	0	0	1	1		3	0	1	0	0	
4	0	1	0	0		4	0	1	0	1	
5	0	1	0	1		5	0	1	1	0	
6	0	1	1	0		6	0	1	1	1	
7	0	1	1	1		7	1	0	0	0	
8	1	0	0	0		8	1	0	0	1	
9	1	0	0	1		9	1	0	1	0	
10	1	0	1	0		10	1	0	1	1	
11	1	0	1	1		11	1	1	0	0	
12	1	1	0	0		12	1	1	0	1	
13	1	1	0	1		13	1	1	1	0	
14	1	1	1	0		14	1	1	1	1	
15	1	1	1	0		15	0	0	0	0	

Fig. 2. Tabla de estados

Tabla de excitación del flip flop D

Ya que estamos usando un flip flop tipo D, tenemos la siguiente tabla, la cual nos muestra la respuesta a los cambios de estado que sufre el flip flop.

Presente	Futuro	Salida D
0	0	0
0	1	1
1	0	0
1	1	1

Fig. 3. Tabla de flip flop tipo D

Analizando el cambio de estado de cada columna tenemos el siguiente diagrama:

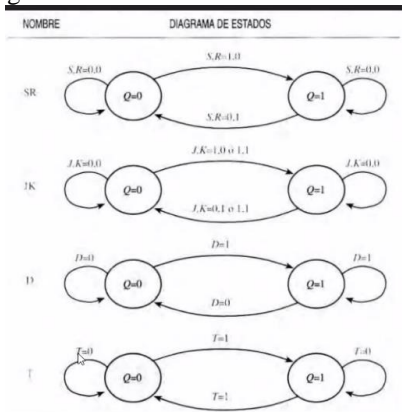


Fig. 4. Diagrama de estados

Una vez analizado los cambios de estado procedemos a implementar el circuito en el simulador proteus, para implementar nuestro circuito debemos tomar en cuenta lo siguiente:

Ahora, para implementar nuestro circuito debemos tomar en cuenta lo siguiente:

- Conectaremos la señal de reloj a nuestro primer flip flop, la salida de este significa el bit menos significativo de nuestro conteo.

- Cada entrada de reset y clear deberá estar conectada a Vcc, ya que se activan en bajo y nos las utilizaremos.

- Conectamos cada terminal D a Q' y también a los clock's. Con esto haremos que la salida anterior se duplique hacia la entrada del siguiente flip flop.

Implementación en proteus:

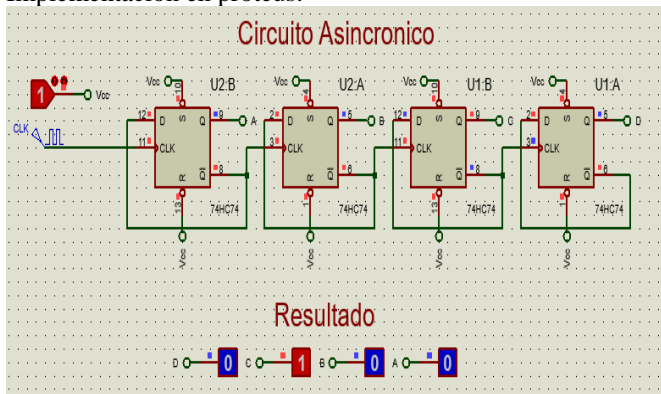


Fig. 5. Implementación en proteus

Ahora veremos la implementación en tinkercad

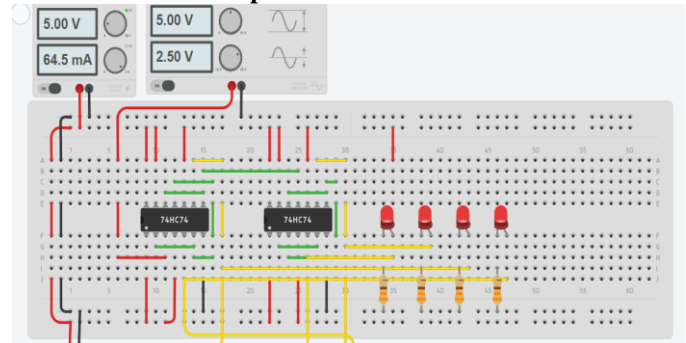


Fig. 6. Implementación en tinkercad

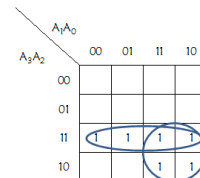
Conversión de código binario de cuatro (4) bits a BCD.

Al tener cuatro bits es posible manejar quince (15) combinaciones de entrada e igual número de combinaciones de salida. La tabla de valores para las posibles combinaciones es la siguiente:

Entradas (Código Binario)				Salidas (Código BCD)				Decimal
A ₃	A ₂	A ₁	A ₀	B ₄	B ₃	B ₂	B ₁	
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	0	2
0	0	1	1	0	0	1	1	3
0	1	0	0	0	1	0	0	4
0	1	0	1	0	1	0	1	5
0	1	1	0	0	1	1	0	6
0	1	1	1	0	1	1	1	7
1	0	0	0	1	0	0	0	8
1	0	0	1	1	0	0	1	9
1	0	1	0	1	0	1	0	10
1	0	1	1	1	0	1	1	11
1	1	0	0	1	1	0	0	12
1	1	0	1	1	1	0	1	13
1	1	1	0	1	1	1	0	14
1	1	1	1	1	1	1	1	15

Fig. 7. Tabla de conversión

Mediante estas tablas es posible observar que los valores de las salidas son iguales hasta el número nueve (9) decimal y de ahí en adelante el código de salida aparece incrementado en seis (6) decimal respecto al código de entrada. Dado lo anterior es posible concebir un sumador que reciba como entradas A los valores del código binario y como entradas B el número seis (6) en binario (0 1 1 0) pero solo cuando la salida B₄ tenga un valor de uno (1). Para obtener la expresión que brinde esta posibilidad se realizó el mapa de Karnaugh para la salida B₄, con el siguiente resultado:



Por lo tanto, se obtiene para B₄ la siguiente expresión:

$$B_4 = A_3A_2 + A_3A_1 = A_3(A_2 + A_1)$$

El resultado de esta implementación se llevará a las entradas B₃ y B₂ del sumador con el fin de obtener el resultado deseado.

Nota: Es necesario recordar que en la anterior descripción no tiene significado el hecho de no tener una secuencia consecutiva de valores. Así mismo en los valores de las entradas A serán ubicados los valores de salida de cada FF.

“Decodificación” y presentación del resultado.

Cada una de las salidas del sumador deben ser las entradas del decodificador/manejador que en este caso es el circuito 4511. A su vez este circuito arroja las salidas *a, b, c, d, e, f* y *g* a nivel alto que representan los siete (7) segmentos de un Display y específicamente uno de cátodo común. El Display utilizado es de la serie 5161 que tiene la siguiente distribución:

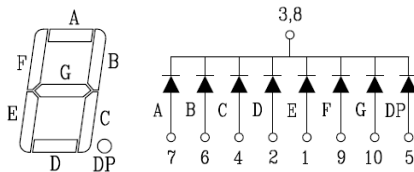


Fig. 8. Distribución de un display de 7 segmentos

Simulación.

Para llevar a cabo la simulación de la implementación del circuito se utilizó la herramienta proteus. Como se muestra a continuación:

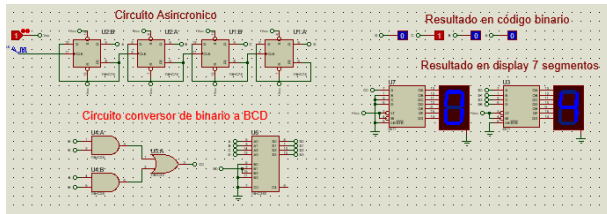


Fig. 9. Simulación en proteus

Para la demostración también se implementó en la plataforma Tinkercad como se muestra a continuación:

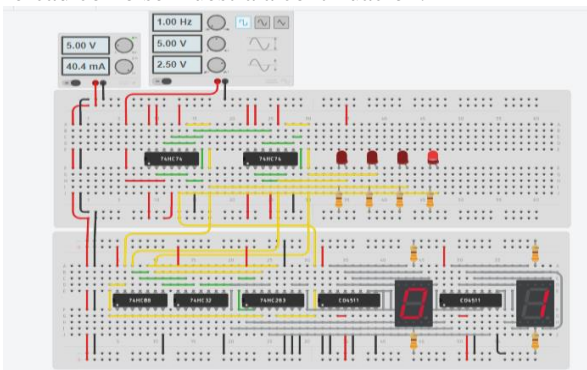


Fig. 10. Simulación en tinkercad

VIII. CONCLUSIONES

Teniendo en cuenta los resultados obtenidos teóricamente y en prácticamente podemos decir que una de la característica principal de los flip flop asíncronos, es que no comparten toda la misma entrada de reloj, algo muy importante ya que no cambian todos de estado al mismo tiempo.

En primera instancia se realizó el análisis de la secuencia de números asignada y se obtuvieron las expresiones necesarias para llevar a cabo la implementación del contador en código binario. Luego se implementó un circuito que ya era conocido como lo es el conversor de código binario a BCD y por último se realizó una actividad conocida como es la decodificación y posterior visualización en Displays de siete segmentos.

En la realización de la simulación se pudo ratificar que los tiempos de retardo en la propagación de datos entre un FF y otro que se presentaban en la implementación asíncrona se vieron disminuidos y que éstos no se presentaban en secuencias de 30ns por cada FF, sino que el retraso de 30ns ocurre una vez en cada cambio de estado.

IX. RECOMENDACIONES

Es importante conocer cuál es la lógica de funcionamiento de los flip flops tipo D para futuros diseños.

Se recomienda no mezclar integrados de tecnología TTL con tecnología Cmos ya que sus diseños admiten diferentes valores de voltajes y corriente, en este diseño de lo realizo debido a la escasez de modelos de integrados en la plataforma de Tinkercad.

Se recomienda tener conocimientos previos sobre contadores y sus tablas de verdad para futuros diseños.

Es preciso planificar un cronograma con diagramas de Grant en las diferentes aplicaciones que existen y para el desarrollo se recomienda el software Project.

X. CRONOGRAMA

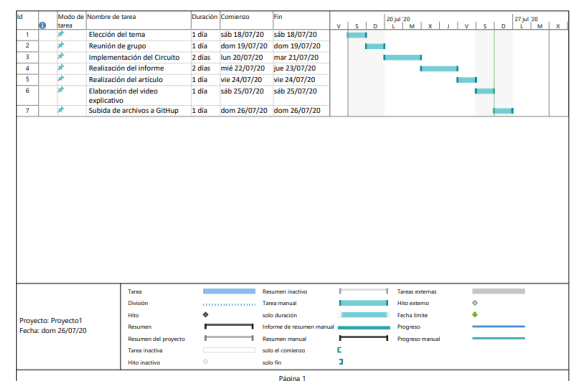


Fig. 11. Cronograma de actividades grupales

XI. ANEXOS

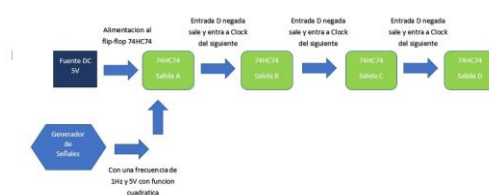


Fig. 12. Diagrama de bloques

XII. MANUAL DE USUARIO

Para poder usar el contador sincrónico se debe tener instalado si es posible la versión más actual de proteus.

Abriremos el archivo de la simulación

Nombre	Fecha de modificación	Tipo	Tamaño
Backup Of Investigacion.pdsbak	26/7/2020 5:02	Archivo PDSBAK	19 KB
Investigacion	26/7/2020 23:46	Proteus Project	19 KB
Investigacion.pdsprj.DESKTOP-K1SF59Q...	26/7/2020 23:46	Archivo WORKSP...	2 KB
Last Loaded Investigacion.pdsbak	26/7/2020 7:26	Archivo PDSBAK	19 KB

Fig. 13. Selección de archivo ejecutable

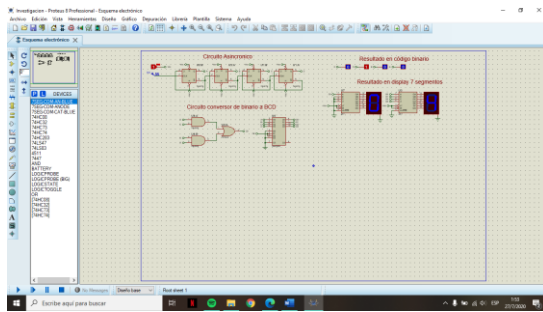


Fig. 14. Simulación en proteus

Nos presenta la interfaz de usuario

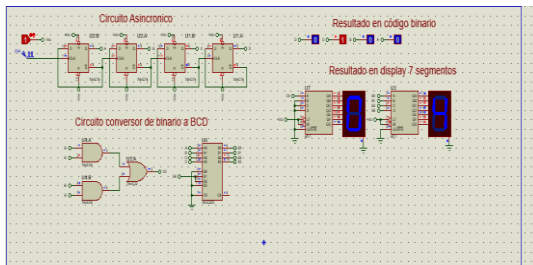


Fig. 15. Interfaz de usuario

Nota: tomar en cuenta que no se debe cambiar la frecuencia de la entrada de reloj. Ni cambiar el voltaje de entrada Vcc ya que podemos quemar los integrados.

Si queremos usar la implementación en tinkercad

procederemos a entrar al siguiente link:
<https://www.tinkercad.com/things/jxkNseHqJ4V-contador-asincronico/editel?sharecode=qnTRplA-IvCfeY6Auv5toNln4Gi4wd6hVyGVvV7Ki40>

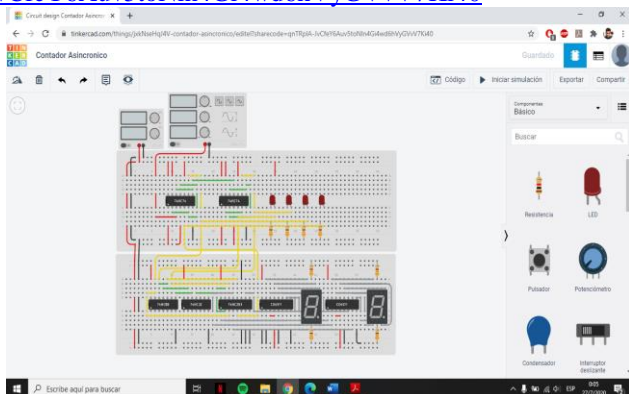


Fig. 16. Interfaz en tinkercad

Nos presenta la interfaz de usuario

Ahora procedemos a simular

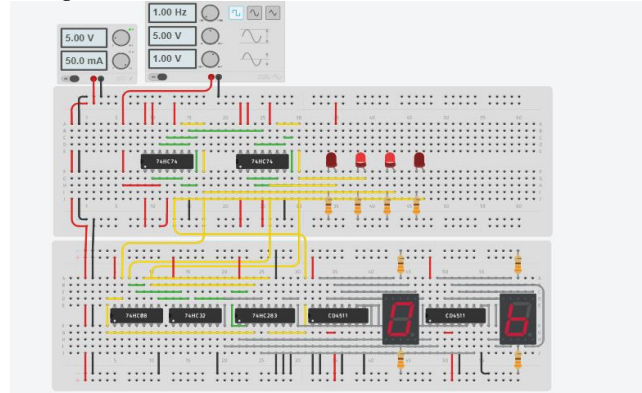


Fig. 17. Simulación en tinkercad

Nota: tomar en cuenta que es debido registrarse antes en la plataforma Tinkercad para poder tener acceso, no se debe cambiar la frecuencia de la entrada de reloj. Ni cambiar el voltaje de entrada Vcc ya que podemos quemar los integrados. Evitar mover las conexiones ya que dañaría el diseño.

XIII. HOJA TÉCNICA

Hojas técnicas de los circuitos integrados utilizados se adjunta en el Git Hub

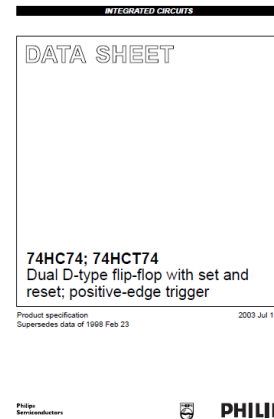


Fig. 18. Hojas técnicas

REFERENCIAS

- [1] R. Siliceo. (2018). *Algoritmo de las operaciones aritmeticas aplicadas a los codigos binarios, octal y hexadecimal con sus respectivas conversions* Ciudad de Mexico
- [2] T. Floyd, "Fundamentos de sistemas digitales", 6ta ed. Ed. Madrid: Pearson, 2006. 15–64.
- [3] D. Alulema, (2020) "Circuitos digitales" Quito, Ecuador
- [4] A. Ricoy. (2020,06). *App inventor en español* Available: <https://sites.google.com/site/appinventormegusta/primeros-pasos>