



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Práctica 1

*Implementación de compuertas básicas en la plataforma
Quartus II, en modo gráfico*

ALUMNOS

Ríos Lira, Gamaliel
Vélez Grande Cinthya

PROFESOR

Flores Olvera, Vicente

ASIGNATURA

Laboratorio de Diseño Digital Moderno

GRUPO

1

26 de febrero de 2023

Índice

1. Objetivo	2
2. Materiales	2
3. Introducción	3
4. Procedimientos	4
4.1. Previo	4
4.2. Parte A	8
4.2.1. Instrucciones	8
4.2.2. Análisis	8
4.2.3. Desarrollo en Quartus	8
4.3. Parte B	10
4.3.1. Instrucciones	10
4.3.2. Análisis	10
4.3.3. Desarrollo en Quartus	11
5. Conclusiones	13
6. Bibliografía	14

1. Objetivo

El alumno diseñara mediante bloques estructurados la implementación de compuertas lógicas utilizando el modo grafico de la plataforma *Quartus*, así como su respectiva simulación virtual.

2. Materiales

Equipo del laboratorio

- Computadora

Software

- *Quartus Prime Lite* 18 (o superior)

3. Introducción

La lógica binaria consiste en variables binarias y operaciones lógicas. Las variables se designan con letras del alfabeto, como A , B , C , x , y , z , etcétera, y cada variable tiene dos y sólo dos posibles valores: 1 y 0. De forma general, hay tres operaciones lógicas básicas, a través de las cuales se pueden conformar operaciones lógicas más complejas.

- **AND.** Se representa a través de un punto (\cdot) y su resultado es 1 si y sólo si ambas entradas son 1.
- **OR.** Se representa a través de un signo más ($+$) y su resultado es 0 si y sólo si ambas entradas son 0.
- **NOT.** Se representa como una testa (\bar{a}) y su resultado es el contrario de la entrada.

Para cada combinación de los valores de variables de entrada, la definición de la operación lógica especifica un valor de salida. Dichas definiciones se pueden presentar en forma compacta con **tablas de verdad**. Una tabla de verdad es una tabla de todas las posibles combinaciones de las variables, y muestra la relación entre los valores que las variables pueden adoptar y el resultado de la operación.

Durante el desarrollo de esta práctica se implementarán las compuertas lógicas básicas en la plataforma de desarrollo *Quartus II*. Una compuerta lógica no es más que un circuito electrónico que opera con una o más señales de entrada para producir una señal de salida. En cuanto a los sistemas digitales, las señales eléctricas (por lo general voltajes) existen con uno de dos valores reconocibles.

4. Procedimientos

4.1. Previo

Pregunta 1

¿Qué es un cronograma?

Se trata de la representación gráfica de un conjunto de señales de entrada y salida como funciones del tiempo. Las señales son representadas en el eje de las ordenadas, y el tiempo en el de las abscisas, reflejando mediante la gráfica la evolución o estado de la señal en cada instante de tiempo. Debido a que dichas señales pueden tomar únicamente valores de 0 y 1, su representación gráfica es una serie de pulsos cuadrados con diferentes longitudes. Las señales de entrada y salida se dibujan en el mismo diagrama para mostrar el comportamiento entrada-salida del sistema digital. Un cronograma muestra todos los posibles patrones de entrada-salida (tal como lo hace la tabla de verdad).

Pregunta 2

¿Qué son las señales tipo Toggle, Random y Pulse?

- **Toggle.** El término *toggle* hace referencia a que existe una conmutación; es decir, se trata de una señal que sólo puede tomar dos valores, cada uno de los cuales representa un estado lógico. La señal de tipo toggle se genera a partir de un dispositivo *flip-flop* (conformado por compuertas lógicas), el cual recibe una señal de reloj y una señal digital T ; esta última determina la forma que tendrá la señal *Toggle*, si $T = 0$, la señal *Toggle* mantiene su valor anterior hasta el siguiente ciclo de reloj; caso contrario, si $T = 1$, la señal invierte su valor.
- **Random.** Se trata de una señal que puede adquirir cualquier valor para cualquier tiempo dado; es decir, se genera a partir de valores aleatorios. No existe algún patrón que se repita en ella, de hecho, presenta muchas fluctuaciones y no se pueden modelar con alguna función matemática.
- **Pulse.** Se trata de una señal que toma un valor determinado (estado de encendido) durante un tiempo finito, denominado tiempo *on*, y toma un valor de 0 (estado de apagado) durante un tiempo *off*, también finito; la suma del tiempo *on* y el tiempo *off* es el periodo de la señal (denotado por T). El comportamiento anterior se repite a lo largo del tiempo para generar la señal; por lo que, ésta se encuentra conmutando entre su estado inicial y el complemento de este.

Pregunta 3

¿Para qué se utiliza la resistencia Pull-Up o Pull-Down?

Se trata de resistencias dispuestas en un circuito de forma especial con la finalidad de mantener un estado lógico de manera estable en la entrada del circuito; es decir, para evitar posibles perturbaciones derivadas de señales de ruido.

- **Resistencia Pull Down.** En esta configuración, la resistencia tendrá una caída de potencial de $0[V]$ de forma predeterminada (*LOW*), ya que un extremo de la resistencia se conecta directamente a Tierra, reflejándose en el otro extremo; mientras que tendrá una caída de potencial de V_{CC} en caso contrario (*HIGH*), si el extremo de la resistencia se conecta directamente a V_{CC} , disipando toda la tensión.
- **Resistencia Pull Up.** En esta configuración, la resistencia tendrá una caída de potencial de V_{CC} de forma predeterminada (*HIGH*); es decir, el voltaje V_{CC} conectado al otro extremo de la resistencia se ve reflejado; mientras que tendrá una caída de potencial de $0[V]$ en caso contrario (*LOW*); es decir, en caso de que la resistencia se conecte directamente a Tierra.



Figura 1: Resistencias *Pull Up* y *Pull Down*

Pregunta 4

¿Para un transistor a qué se le llama “Corte” y “Saturación” y cuál es su efecto sobre las señales lógicas?

Se trata de dos regiones de operación de los transistores.

- **Corte.** Un transistor está en corte cuando las uniones que lo conforman están polarizadas en sentido inverso, lo cual produce una corriente de colector despreciable y por ende, también la corriente a través de la terminal emisora es despreciable. En este caso, el transistor se comporta como un interruptor abierto.

- **Saturación.** Un transistor está en saturación cuando las uniones que lo conforman están polarizadas en directa, lo cual genera un crecimiento exponencial de la corriente que circula a través de su terminal colector, y en consecuencia, a través de la terminal emisora. En este punto el transistor actúa como un interruptor cerrado, permitiendo el paso de la corriente.

Abstrayendo el comportamiento descrito, se puede considerar un estado *on* cuando el transistor está en la región de saturación y un *off* cuando el transistor está en la región de corte.

Pregunta 5

Analiza y caracteriza el siguiente circuito a través de cualquier simulador e identifica el comportamiento que este tiene.

- *Identifica el voltaje resultante en cada uno de los transistores considerando que tanto la entrada A como B solamente pueden tener dos voltajes (ground y V_{CC}), toda esta información colócala dentro de una respectiva tabla y completa la misma.*

Para este caso, dentro del simulador se consideró un voltaje $V_{CC} = 5[V]$, con lo cual se obtuvieron los siguientes valores:

A	B	C	Q1	Q4	Q2, Q3
Ground	Ground	Ground	2.8	2.8	0
Ground	Ground	VCC	2.8	0	0
Ground	VCC	Ground	0	2.8	0
Ground	VCC	VCC	0	0	5
VCC	Ground	Ground	0	2.8	0
VCC	Ground	VCC	0	0	5
VCC	VCC	Ground	0	2.8	0
VCC	VCC	VCC	0	0	5

Tabla 1: Voltajes de la simulación (Pregunta 5)

La simulación se llevó a cabo en la plataforma *Qucs*, tal como se muestra a continuación:

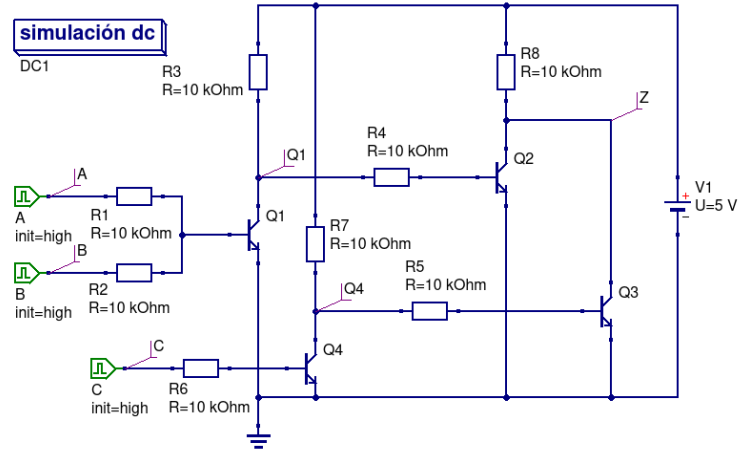


Figura 2: Circuito de simulación en *Qucs* (Pregunta 5)

La caracterización de la configuración de cada uno de los transistores es la siguiente:

- $Q_1: f_1(ABC) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C$
- $Q_4: f_4(ABC) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + ABC$
- $Q_2, Q_3: f_{2,3}(ABC) = \bar{A}BC + A\bar{B}C + ABC$

4.2. Parte A

4.2.1. Instrucciones

Construir de manera gráfica las siguientes compuertas lógicas y simularlas dentro de la plataforma Quartus.

- *AND*
- *OR*
- *NOT*
- *XOR* (y su forma expandida)

4.2.2. Análisis

Antes de comenzar a realizar esta sección, es importante tener muy presente el comportamiento de cada una de las compuertas con la finalidad de comprobar de forma satisfactoria el resultado de la actividad. La tabla de verdad de cada una de las compuertas se muestra a continuación:

Tabla 2: Tablas de verdad (Sección A)

<i>A</i>	<i>B</i>	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

(a) Compuerta AND

<i>A</i>	<i>B</i>	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(b) Compuerta OR

<i>A</i>	\bar{A}
0	1
1	0

(c) Compuerta NOT

<i>A</i>	<i>B</i>	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(d) Compuerta XOR

De igual forma, se pueden comprobar la siguiente equivalencia entre las operaciones lógicas XOR, AND y OR:

$$A \oplus B = \bar{A} \cdot B + A \cdot \bar{B} \quad (1)$$

4.2.3. Desarrollo en Quartus

Para este punto, lo primero que hicimos fue crear un proyecto con ayuda de la pestaña *File/New Project Wizard* y configurarlo. Posteriormente, se creó un archivo de tipo *Block Diagram/Schematic File*. Posteriormente, se agregaron todas las compuertas, tal como se muestra a continuación:

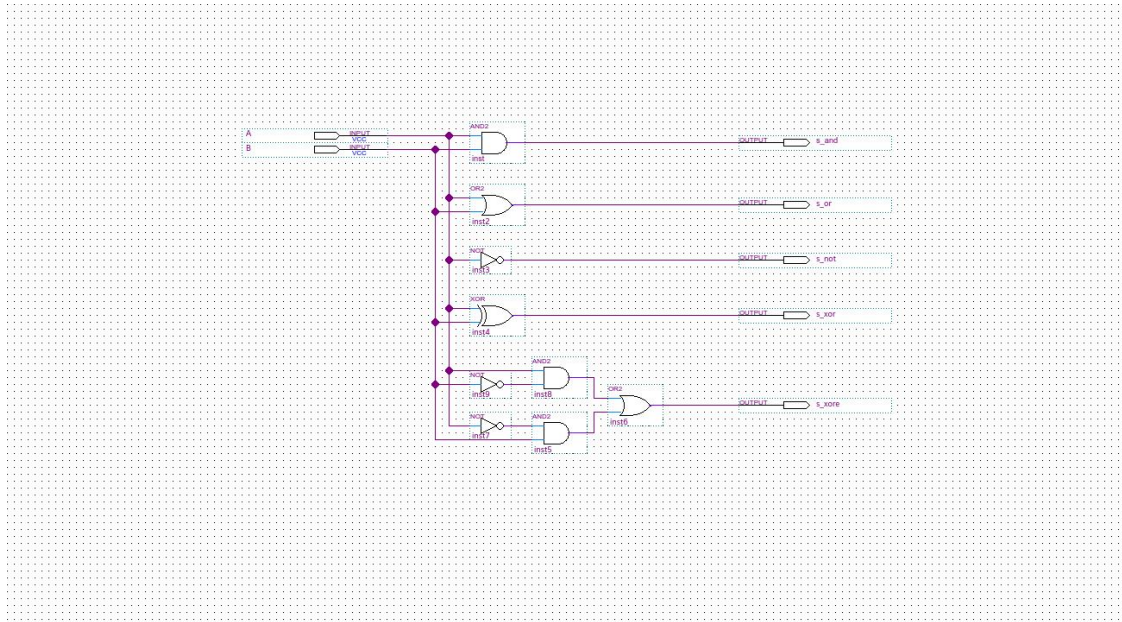


Figura 3: Esquema del funcionamiento de las cinco compuertas (Sección A)

Tal como se puede apreciar, se tuvieron que agregar algunos elementos adicionales. Por una parte, se tienen las señales de entrada:

- A
- B

Por otra parte, se tienen las salidas:

- *s_and*: La salida de la compuerta AND.
- *s_or*: La salida de la compuerta OR.
- *s_not*: La salida de la compuerta NOT (sólo toma como entrada el valor de A).
- *s_xor*: La salida de la compuerta XOR.
- *s_xore*: La salida de la compuerta XOR equivalente.

Con lo anterior, se compiló el proyecto. Es importante que el paso anterior se llevó a cabo con éxito para que funcione correctamente lo siguiente. Se creó un nuevo archivo del tipo *University Program VWF*, a través del cual se pudo realizar un **cronograma** con las señales de entrada y salida de simulación. Para llevar a cabo de forma correcta el cronograma, fue importante alternar de forma correcta los valores de A y B.

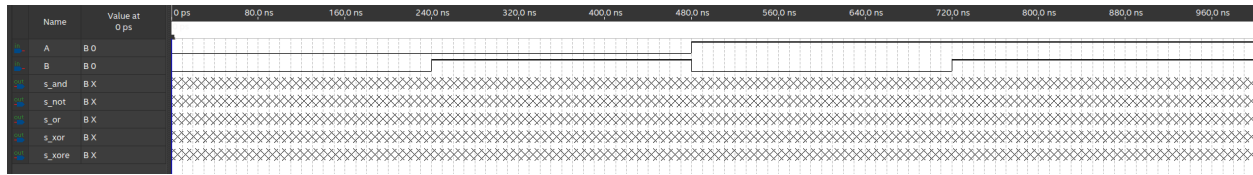


Figura 4: Alternación de los valores de A y B (Sección A)

Al ejecutar la *Simulación Funcional* se obtuvo lo siguiente:

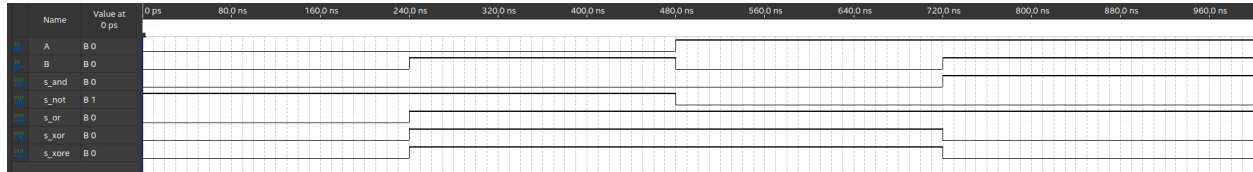


Figura 5: Cronograma del sistema (Sección A)

Se puede comprobar entonces, a través de la Tabla 2 que el cronograma de la Figura 4 tiene el comportamiento esperado, por lo tanto, el sistema fue construido de una forma adecuada.

4.3. Parte B

4.3.1. Instrucciones

Construir de manera gráfica las siguientes compuertas lógicas y simularlas dentro de la plataforma Quartus.

- *NAND (y su forma expandida)*
- *NOR (y su forma expandida)*
- *XNOR (y su forma expandida)*

4.3.2. Análisis

De igual forma que en la sección anterior, es importante tener presente el comportamiento de cada una de las compuertas con la finalidad de comprobar de forma satisfactoria el resultado de la actividad. La tabla de verdad de cada una de las compuertas se muestra a continuación:

Todas las compuertas anteriores tienen una equivalencia usando únicamente las compuertas AND, OR y NOT, tal como se muestra a continuación.

$$\overline{A \cdot B} = \bar{A} + \bar{B} \quad (2)$$

$$\overline{A + B} = \bar{A} \cdot \bar{B} \quad (3)$$

$$\overline{A \oplus B} = \bar{A} \cdot \bar{B} + A \cdot B \quad (4)$$

Tabla 3: Tablas de verdad (Sección B)

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

(a) Compuerta NAND

A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

(b) Compuerta NOR

A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

(c) Compuerta XNOR

4.3.3. Desarrollo en Quartus

Se comenzó esta actividad partiendo de lo que ya se había hecho anteriormente en la actividad A. El diagrama del sistema quedó de la siguiente forma:

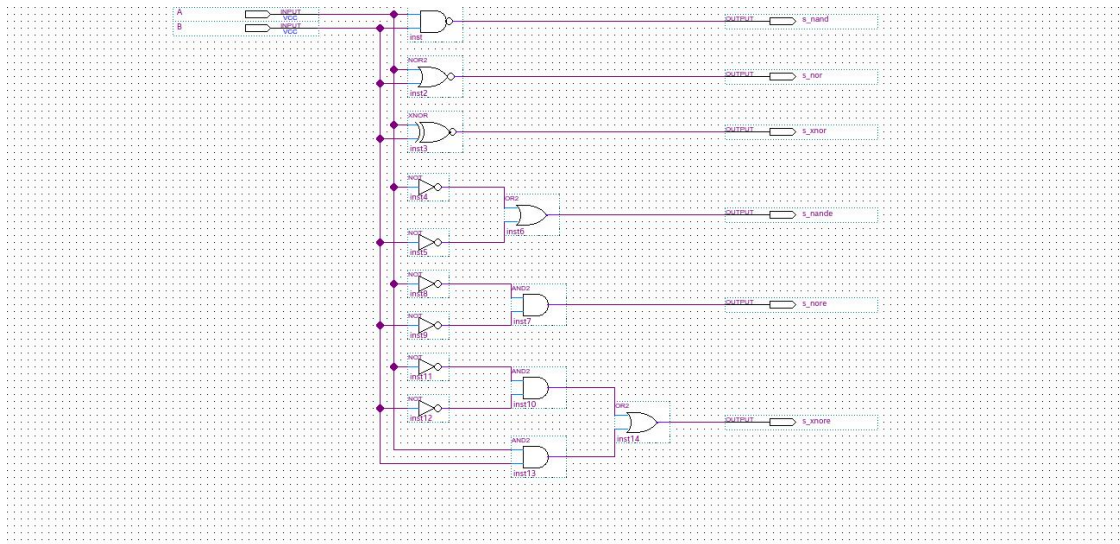


Figura 6: Esquema del funcionamiento de las seis compuertas (Sección B)

Posteriormente, se creó otro archivo con formato *University Program VWF*, a través del cual se pudo realizar un **cronograma** con las señales de entrada y salida de simulación. La configuración de las señales de entrada se muestra a continuación:

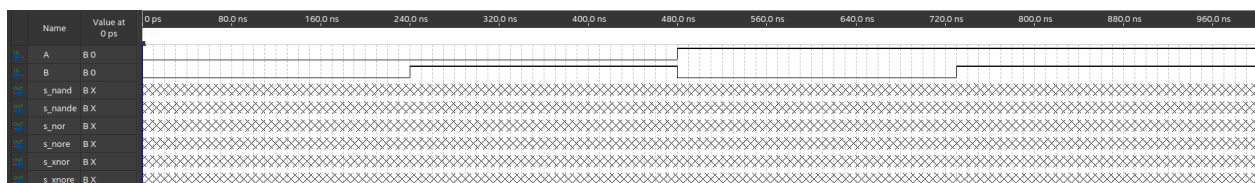


Figura 7: Alternación de los valores de A y B (Sección B)

Tal como se puede apreciar, se tienen las siguientes señales. Por un lado están las entradas:

- A
- B

y por otro lado se tienen las salidas:

- s_nand : La salida de la compuerta NAND.
- s_nor : La salida de la compuerta NOR.
- s_xnor : La salida de la compuerta XNOR.
- s_nande : La salida de la compuerta NAND equivalente.
- s_nore : La salida de la compuerta NOR equivalente.
- s_xnore : La salida de la compuerta XNOR equivalente.

Al ejecutar la *Simulación Funcional* se obtuvo lo siguiente:

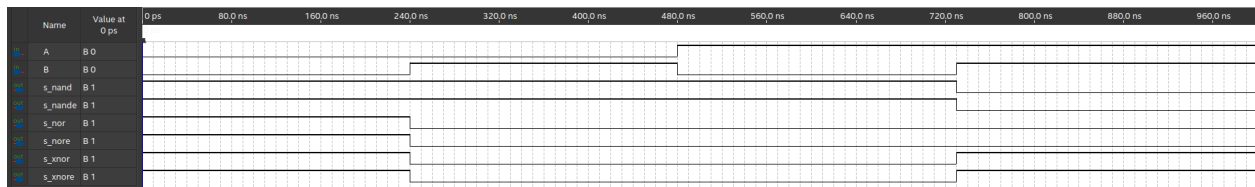


Figura 8: Cronograma del sistema (Sección B)

En la parte anterior, se pueden verificar dos cosas:

1. Que las compuertas extendidas tienen el mismo comportamiento que sus compuertas correspondientes.
2. Que el comportamiento de las compuertas en la Figura 8 es el mismo que el descrito por las tablas de verdad de la Tabla 3.

5. Conclusiones

Ríos Lira, Gamaliel. Con la realización de la práctica, se cumplió el objetivo planteado al inicio de la misma. Por un lado, nos ambientamos con la plataforma *Quartus II* y de forma general exploramos este entorno de desarrollo con la finalidad de distinguir sus funcionalidades. Y por otra parte, se lograron implementar las compuertas AND, OR, NOT, XOR, NAND, NOR y XNOR de forma exitosa. Además de lo anterior, se pudieron comprobar algunas formas equivalentes de las compuertas XOR, NAND, NOR y XNOR.

Vélez Grande, Cinthya. Mediante el desarrollo de la práctica se realizó un primer acercamiento al software de diseño *Quartus II*, con el cual se implementaron diversas compuertas lógicas de manera virtual, cumpliendo con el objetivo principal de la práctica. A partir de lo anterior, se reafirmaron los conocimientos vistos en teoría en torno al comportamiento de las compuertas básicas AND, OR y NOT; además se realizó el análisis de las compuertas complementarias NAND, NOR y de las compuertas XOR y XNOR; también se realizó el análisis de las compuertas equivalentes para estas últimas, formadas a partir de compuertas básicas, confirmando con los resultados obtenidos en la simulación su equivalencia.

6. Bibliografía

- [1] R. Boylestad y L. Nashelsky, *Electronic Devices and Circuit Theory*. Pearson Education Limited, 2013.
- [2] C. de Wikipedia, *Modulación por ancho de pulsos*, oct. de 2022. dirección: https://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_ancho_de_pulsos.
- [3] L. Del Valle Hernández, *Resistencia pull up y pull down*, mar. de 2021. dirección: <https://programarfacil.com/blog/arduino-blog/resistencia-pull-up-y-pull-down/>.
- [4] M. Ferdjallah, *Introduction to Digital Systems*. Hoboken, NJ, United States: Wiley, 2011.
- [5] M. Mano, *Digital Design*. Upper Saddle River, NJ, United States: Prentice Hall, 2002.
- [6] M. Olmo y Nave, *T Flip-Flops*. dirección: <http://hyperphysics.phy-astr.gsu.edu/hbasees/Electronic/Tflipflop.html>.