



ni.com

## A. Convenciones del Manual.

---

Las siguientes convenciones parecen en este manual:



El símbolo » lo conduce a través de ítems de menús anidados y opciones de cajas de dialogo hacia una acción final. La secuencia **File** » **Page Setup** » **Options** le indica seleccionar el menú desplegable **File**, seleccionar el ítem **Page Setup** y seleccionar **Options** desde la ultima caja de dialogo.



Este icono denota una advertencia, la cual le alerta sobre información de **consejo**.



Este icono denota una **nota**, la cual le alerta sobre información importante.



Este icono denota prevención, la cual le advierte sobre precauciones a tomar con el fin de evitar lesiones, perdidas de datos o una caída del sistema.

### **bold**

Texto en negrita denota ítems que usted debe seleccionar o hacer Clic en el software, tales como ítems de menús u opciones de cajas de dialogo. El texto en negrita también denota nombres de parámetros, controles y botones en el panel frontal, cajas de dialogo, secciones de cajas de dialogo, nombres de menú y nombres de paletas.

### *italic*

Texto en itálica denota variables, énfasis, una referencia cruzada o una introducción a un concepto clave. El texto en itálica también denota texto que se ubica ara una palabra o valor que usted debe suministrar.

### *monospace*

Texto en esta fuente denota los mensajes y respuestas que la computadora automáticamente imprime en la pantalla. Esta fuente también enfatiza líneas de código que son diferentes de otros ejemplos.

# Introducción National Instruments

- Operaciones Directas en 40 Países
- Mas de 4,300 empleados; Operando en más de 40 países
- 800 Miembros del Programa de Alianza
- Corporativo en Austin, Texas
- Más de 5000 Universidades en 110 Países.
- Tecnología para todas las ciencias y disciplinas en ingeniería.



★ Oficinas de Ventas Directas  
● Distribuidores

ni.com

NATIONAL INSTRUMENTS

**National Instruments** es una empresa pionera y líder en la tecnología de la instrumentación virtual, un concepto revolucionario que ha cambiado la forma en que ingenieros y científicos abordan las aplicaciones de medición y automatización. Aprovechando el poder de la PC y sus tecnologías relacionadas, la instrumentación virtual aumenta la productividad y reduce los costos por medio de software de fácil integración (como el ambiente de desarrollo gráfico LabVIEW de NI) y hardware modular (como los módulos PXI para adquisición de datos, control de instrumentos y visión artificial).

Los clientes de NI incluyen ingenieros, científicos y profesionales técnicos en una amplia gama de industrias. Desde pruebas a grabadoras de DVD hasta investigación de medicamentos avanzados, nuestros clientes utilizan el software y hardware de NI para ofrecer un amplio conjunto de productos de manera más rápida y a un menor costo.

## Perfil

- Compañía Líder en mediciones basadas en Computadoras y Automatización
- Record a Largo Plazo de Crecimiento y Ganancias
- Registro de Ingresos: \$740M en el 2007
- Nombrada entre las mejores compañías para trabajar entre las mejores 100 de *Fortune* por noveno año consecutivo
- Registro a Largo Plazo de gran Crecimiento y Rentabilidad

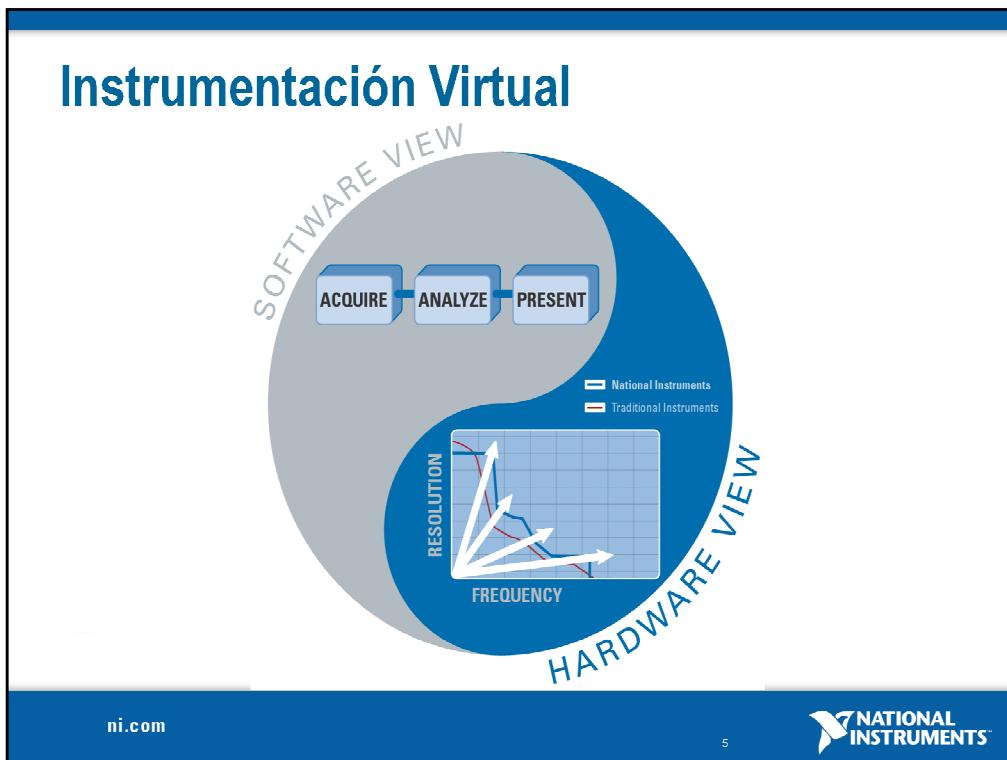


ni.com



**National Instruments** transforma la manera en la que ingenieros y científicos de todo el mundo diseñan, hacen prototipos, y desarrollan sistemas para pruebas, control y aplicaciones en sistemas embebidos. Usando el software de programación gráfica y los dispositivos modulares de NI, usuarios de más de 25,000 compañías simplifican anualmente el desarrollo, incrementando la productividad y reduciendo dramáticamente el tiempo de venta. Una prueba al asar son los sistemas de nueva generación para crear avances en dispositivos médicos, los clientes de NI continuamente desarrollan tecnologías que impactan a millones de personas.

Durante los pasados 20 años LabVIEW se ha ganado una fuerte reputación como una herramienta de software para crear soluciones en mediciones. Los usuarios de LabVIEW reducen el tiempo de desarrollo siendo esto una ventaja sobre los tradicionales programadores en texto. Esta ventaja puede ser usada para pre-construir aplicaciones de pruebas y de control. Esto permite a los usuarios reducir el tiempo de desarrollo arriba de 40% y crear fácilmente complejas aplicaciones de control y adquisición de datos las cuales, en el paso, requerían de extensivo conocimiento de diseño en software.



Desde su fundación en 1976, National Instruments ha revolucionado la forma en que trabajan los técnicos e investigadores de todo el mundo. Esta empresa desarrolla y fabrica una enorme variedad de productos de software y hardware integrados que, utilizados junto con la PC estándar, permiten sustituir o integrar instrumentos tradicionales para monitorear y controlar procesos. Como empresa líder en la producción de instrumentos de medición y automatización basados en PC, National Instruments propone soluciones innovadoras para ayudar a sus clientes a lograr sus objetivos de la forma más rápida y eficaz posible, manteniendo sus costos reducidos.

## La Revolución de la Instrumentación Virtual

La rápida adopción de las Computadoras en los últimos 20 años ha provocado una revolución en instrumentación para pruebas, mediciones y automatización. Un mayor resultado del concepto que han dejado las Computadoras, es el concepto de Instrumentación Virtual, el cual ofrece mucho beneficios a los Ingenieros y Científicos que requieren incrementar sus tiempos de productividad, exactitud y rendimiento.



Hay muchas plataformas donde podemos encontrar una Computadora que lleva a cabo mediciones y recolección de datos, actualiza el procesamiento de datos, manipula y provee la conexión a Red y ofrece reportes.

Hoy en día vemos ejemplos donde el poder de una Computadora es utilizado para desarrollar soluciones para múltiples aplicaciones de la Industria Petrolera.

## Instrumentación Virtual en Todos Lados

Más de 25,000 compañías, donde más del 90% de las compañías manufactureras corresponden a las 500 de *Fortune*

ni.com

NATIONAL INSTRUMENTS

National Instruments es una compañía muy afortunada en estar presente en mas de 25,000 compañías diferentes cada año, y apoyarles en seleccionar la solución optima para sus necesidades de pruebas y mediciones. NI continuamente monitorea los últimos retos y metas en las industrias, por lo que ha trabajado con mas del 90% de las 500 principales empresas de Manufactura de *Fortune*.

## Áreas de Aplicación en la Academia

- Medición e Instrumentación.
- Diseño de Circuitos.
- Diseño de Control y Simulación.
- Procesamiento de Señales e Imágenes.
- Comunicaciones y Radiofrecuencia.
- Sistemas Embebidos.




[ni.com](http://ni.com)

NATIONAL INSTRUMENTS

### • Medición e Instrumentación.

Vea como, desde temperatura hasta mediciones de señales dinámicas, NI ofrece una familia muy completa de dispositivos de Adquisición de Datos para Computadoras de Escritorio, Portátiles y aplicaciones de Red e Investigación.

### • Diseño de Circuitos.

Experiencia en una perfecta integración entre NI Multisim 10.1

®, NI LabVIEW, y NI ELVIS, para diseñar, simular, realizar prototipos y probar circuitos.

### • Diseño de Control y Simulación.

Investigación y enseñanza en los conceptos de diseño de control, incluyendo diseño, simulación de sistemas dinámicos, identificación de sistemas, e implementación de tiempo real con los productos de control de NI

### • Procesamiento de Señales e Imágenes.

Explore los conceptos de procesamiento de señales con diseño de filtros digitales interactivos a los prototipos prácticos de DSPs con una plataforma completa de NI.

### • Comunicaciones y Radiofrecuencia.

Combine Software gráfico e intuitivo con el hardware mas nuevo de mediciones de precisión para desarrollar una plataforma de alto rendimiento para aprendizaje práctico y pionero de investigación en las comunicaciones inalámbricas.

### • Sistemas Embebidos.

Use una colección completa de herramientas complementarias de NI para hacer sistemas embebidos con FPGAs, DSPs, MPUs o cualquier microprocesador económico y sencillo de 32-bit .

## Iniciativas Académicas

Facultades y Estudiantes en más de 5,000 universidades al rededor del mundo

ni.com

NATIONAL INSTRUMENTS

La Instrumentación Virtual ha sido adoptada por Universidades líderes en todo el mundo para educar a los futuros Ingenieros y Científicos. En laboratorios de enseñanza, la instrumentación virtual ha sido utilizada por la Ingeniería Eléctrica, Ingeniería Mecánica y Pruebas Físicas para mostrar los conceptos a los estudiantes.

En el campo de la Investigación, la Instrumentación Virtual ha sido utilizada en comunicaciones, diseño de control y simulación, procesamiento de señales, procesamiento de imágenes, y una gran variedad de otras áreas de investigación. Los estudiantes han aprendido a utilizar la instrumentación virtual y así dar un mayor valor a sus Currículos y al mismo tiempo divertirse.



## Recursos para Profesores e Investigadores

National Instruments está comprometido con la formación universitaria y con que los profesores estén siempre al corriente de las últimas tendencias tecnológicas. Hemos hecho hincapié en la formación universitaria con descuentos, participación, y con eventos pensados y organizados para la Universidad. Además, estamos orgullosos de haber inaugurado en los últimos dos años aulas de formación National Instruments en universidades. Estas aulas están destinadas a la formación continua de profesores y también empresas.

## Recursos para Estudiantes

National Instruments está comprometido con su aprendizaje, es por ello que dentro de los recursos que tiene para los estudiantes son: Usted podrá evaluar sus conocimientos en LabVIEW GRATIS realizando el **LabVIEW Fundamentals Exam for Academia** en línea

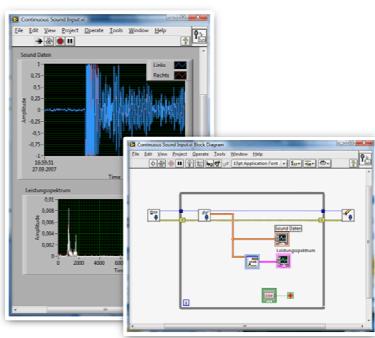
# Desde Preescolar hasta Grandes Proyectos de Ciencia

Preescolar | Primaria | Secundaria | Preparatoria | Universidad | Investigación

La Ciencia, Tecnología, Ingeniería y Matemáticas están donde quiera y a nuestro alrededor.



**LEGO MINDSTORMS NXT**  
*"El juguete más inteligente y divertido del año"*



**CERN Gran Colisionador de Hadrones**  
*"El instrumento más poderoso del mundo"*

ni.com

NATIONAL INSTRUMENTS

## MINDSTORMS NXT Works

Aprenda como programar un robot que haga todo lo que usted quiera, con un entorno nuevo de programación completamente grafico.

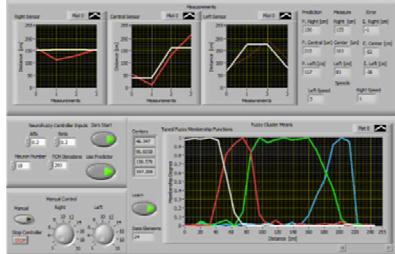
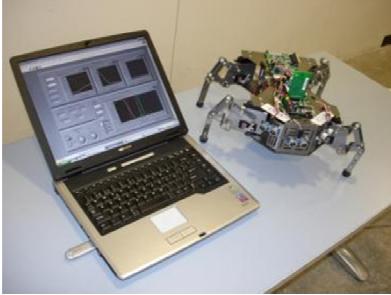
## CERN

Los Ingenieros en el CERN, eligieron a National Instruments para controlar el Gran Colisionador de Hadrones.

**Caso de Estudio: Diseño de Control y Simulación.**

**IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL INTELIGENTE EN LABVIEW**

- Desarrollar y validar un grupo de controladores inteligentes de lógica difusa y redes neuronales en LabVIEW
- Validación mediante un robot Cuadrúpedo

*"Gracias al uso de LabVIEW fuimos capaces de centrarnos en el desarrollo de nuestro proyecto... Nosotros no requerimos ser programadores especializados para desarrollar la caja de herramientas"*

ni.com

NATIONAL INSTRUMENTS

## Implementación de Algoritmos de Control Inteligente en LabVIEW

**Project Director:** Dr. Pedro Ponce

ITESM

**Software Implementado:** NI LabVIEW

### El Reto:

Validar de forma experimental el desempeño de algoritmos para controladores inteligentes industriales utilizando una plataforma de software con capacidad de interacción con hardware.

### La Solución:

Utilizar las capacidades de programación gráfica de LabVIEW y su facilidad de comunicación con hardware para implementar algoritmos de control inteligente y validarlos mediante su interacción con un robot cuadrúpedo.

## Simulador de los Lazos de Control de una Central Termoeléctrica Ciclo Combinado.



**Reto:**

- Sistema de control implantado en un equipo está diseñado para tratar de mantener bajo control un proceso específico, tomando diferentes variables del medio, como son temperatura, presión, nivel,

**Solución:**

- Simulación de los procesos de generación de energía eléctrica, variando los parámetros que intervienen en dichos procesos y logrando así ver la respuesta del sistema de control y los eventos que esto genere.

*"Este esfuerzo será recompensado en corto tiempo por medio de la tranquilidad y seguridad cuando se opera este equipo... y proveerá un gran beneficio a la compañía al reducir el riesgo de fallas humanas"*

Rafael Neri Fernández, Universidad Veracruzana

ni.com

NATIONAL INSTRUMENTS

## Simulador de los Lazos de Control de una Central Termoeléctrica Ciclo Combinado.

**Autores:** Rafael Neri Fernández, Certificado como Desarrollador Asociado de LabVIEW.

Universidad Veracruzana, México.

### Software Implementado:

- LabVIEW Professional Development System
- LabVIEW DSC Module
- LabVIEW Control and Simulation Module
- LabVIEW PID Toolkit
- LabVIEW Report Generation Toolkit.

### El reto:

Cada sistema de control implantado en un equipo está diseñado para tratar de mantener bajo control un proceso específico, tomando diferentes variables del medio, como son temperatura, presión, nivel, etc. Ya que con el tiempo, estos dispositivos suelen irse deteriorando, causando lectura errónea en los sensores, provocando una serie de fallas que repercuten en el proceso.

Por este motivo, es necesario llevar a cabo una simulación de los proceso de control aplicados en los diferentes sistemas de generación, con el fin poder prevenir posibles fallas. Además de facilitar la capacitación a los trabajadores acerca de los procesos llevados a cabo en la planta.

**La solución:**

Por medio de LabVIEW, se realizó la simulación de los procesos de generación de energía eléctrica, pudiendo variar los parámetros que intervienen dichos procesos y logrando así ver la respuesta del sistema de control y los eventos que esto genere.

**Caso de Estudio: Medición e Instrumentación**

## Monitoreo y Control por Internet de una Planta Pasteurizadora

- Sistema de control basado en computadoras
- Control total sobre E/S analógicas y digitales
- Control y Monitoreo en LAN e Internet

ni.com

NATIONAL INSTRUMENTS

## Monitoreo y Control por Internet de una Planta Pasteurizadora

Autores: Ing. Hugo H. Ordoñez R., M.C. José Rivera M.

Instituto Tecnológico de Chihuahua , DEPI.

### El reto:

Implementar un sistema de monitoreo y control remoto por red de área local (*LAN*) y extendida (*WAN*) que fuera capaz de generar gráficos en vivo, control, almacenamiento y la generación de reportes de las condiciones operativas del proceso, en un ambiente *Windows* en una PC de propósito general.

### La Solución:

Utilizar en *software* los componentes ActiveX *Components Works* Ver. 3.0 de *National Instruments* que incluyen perillas, gráficos, y protocolo de transferencia de comunicación DataSocket (*dstp:*) entre otras funciones y en *hardware* la tarjeta multifunciones Lab-PC+.

## Notas

---

## Notas

---

# Clase: LÓGICA DE PROGRAMACIÓN.

ni.com



18

## Temario de la clase

2. Lógica de Programación.
  1. Metodología para la solución de problemas por medio de computadoras.
  2. Ejercicio 1. Conversión de °C a °F Principios de Programación
  3. Ejercicio 2. Calculo del Área y Perímetro de un cuadrado.
  4. Ejercicio 3. Uso de Ciclo *While*
  5. Ejercicio 4. Clusters de Error y su manipulación
  6. Ejercicio 5. Máquina de estado Simple
  7. Propuesta de Proyecto Final: Juego del Simón

## Metodología para la solución de problemas por medio de computadoras.

1. Definición del problema.
2. Diseñar un algoritmo (Pseudocódigo) y/o Diagrama de Flujo
  - Identificar: Entradas, Salidas, requerimientos adicionales.
3. Implementar el Diseño.
4. Probar y Verificar la implementación.
5. Mantenimiento y Actualización de la implementación.

ni.com



19

**LabVIEW** es un lenguaje de programación que usted puede usar para resolver múltiples problemas de programación. Las habilidades para resolver problemas son esenciales en la creación de soluciones en LabVIEW. Los programadores emplean un método de desarrollo de software. Seguir un método ayuda a un programador a desarrollar código que posee un gran potencial para resolver exitosamente un problema dado cuando se compara con código escrito sin un plan previo. Un método también ayuda a realizar un código más entendible, escalable y modificable.

- **Definición del problema.**

En esta parte del método usted define cuál es su problema para que pueda abordarlo con todos los factores necesarios identificados. La forma como identifique inicialmente el problema le puede ahorrar tiempo mientras diseña e implementa una solución.

- **Diseñar un algoritmo (Pseudocódigo) y/o Diagrama de Flujo**

- **Identificar:** Entradas, Salidas, requerimientos adicionales.

Luego de determinar el alcance del problema, puede diseñar una solución analizando el problema. Parte de analizar la solución es identificar las entradas y salidas de software, así como cualquier requerimiento adicional. Después de definir las entradas y salidas, puede diseñar un algoritmo, un diagrama de flujo y/o diagrama de transición de estados como ayuda para llegar a una solución de software.

- **Implementar el Diseño.**

En la etapa de implementación, usted crea código para su algoritmo o diagrama de flujo. Cuando se escribe código en un lenguaje basado en texto el algoritmo se traduce elegantemente en cada línea de código, dependiendo del nivel de detalle mostrado en el algoritmo. Debido a que LabVIEW es un lenguaje gráfico de programación, el diagrama de flujo trabaja mejor de la misma forma.

- **Probar y Verificar la implementación.**

Probar y verificar es una parte importante del método de desarrollo de software. Asegúrese de probar su implementación con datos que sean tanto lógicos como ilógicos para la solución que creo. Al probar datos lógicos verifica que las entradas producen los resultados esperados. Al probar datos ilógicos, usted prueba para ver si el código posee control efectivo de errores.

- **Mantenimiento y Actualización de la implementación.**

El mantenimiento es el proceso continuo de resolver errores de programación adicionando cambios elaborados en paralelo a la solución original de un problema.

## Ejercicio 1. Conversión de °C a °F

Complete los pasos siguientes para crear un VI que tome un número representando **grados Celsius** y lo convierta a un número representando **grados Fahrenheit**.

### Definición del Problema.

La formula para convertir de °C a °F es:

$$^{\circ}\text{F} = ^{\circ}\text{C} * 1.8 + 32$$

### Objetivo.

Que el alumno aprenda a manejar herramientas básicas en LabVIEW, así como familiarizarse con el entorno de este.

### Escenario.

Complete los pasos siguientes para crear un VI que tome un número representando grados Celsius y lo convierta a un número representando grados Fahrenheit.

### Diseño

Este programa deberá contener las siguientes entradas y salidas:

Entradas	Salidas
°C (Grados Centígrados)	°F (Grados Fahrenheit)

## Reto.

Desarrollar aplicaciones similares, agregar algoritmos para llevar a cabo otro tipo de conversiones, por ejemplo: °F a °C, de °C a °K, etcétera.

## Implementación.

1. Prepare su Software.

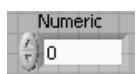
- Abra LabVIEW.
- Abra un VI en Blanco.

2. Diríjase el Panel Frontal.



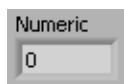
**Consejo** Para intercambiar la vista entre el Panel Frontal y el Diagrama de Bloques, puede presionar las teclas **Ctrl + E**

3. Coloque un Control Numérico en el Panel Frontal. Este se utilizará para introducir el valor en grados centígrados.



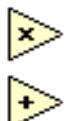
- En la ventana de Panel Frontal, de **Clic Derecho** sobre el área de trabajo, aparecerá la paleta de controles.
- Los Controles Numéricos se encuentran en la Paleta de **Controles** » **Numeric** » **Numeric Controls**
- Para colocarlo, ese se selecciona y se arrastra hacia el Panel Frontal.
- Una vez que el elemento se coloco en el Panel Frontal, podemos editar algunas de sus propiedades, tal como el nombre de la etiqueta. Esto se puede llevar a cabo dando **Doble Clic** sobre la etiqueta del Control Numérico, posteriormente ingrese el nombre con el que desee nombrar a su elemento. En este elemento ingresaremos el nombre **Grados Centígrados (°C)**.

4. Coloque un Indicador Numérico en el Panel Frontal.



- El **Indicador Numérico** se encuentra en la Paleta de **Controles** » **Numeric** » **Numeric Indicator**.
- El nombre de etiqueta para este elemento lo podemos cambiar a **Grados Fahrenheit (°F)**.

5. Agregar dos operaciones básicas: **Multiply** y **Add** en el Diagrama de Bloques.



- Cambiar la vista al Diagrama de Bloques.

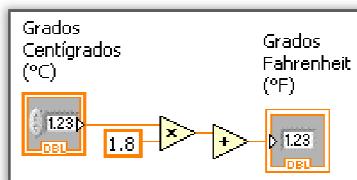


**Consejo** Para intercambiar la vista entre el Panel Frontal y el Diagrama de Bloques, puede presionar las teclas **Ctrl + E**

- La operación numérica **Multiply** y **Add** se localizan en la Paleta de Funciones » **Numeric**.

#### 6. Concluir el cableado de terminales.

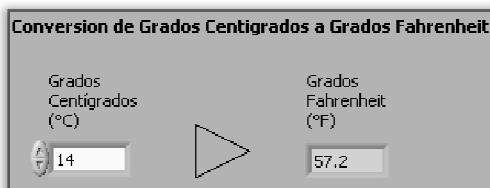
- Cablee de la terminal de salida del control **Grados Centígrados ( C )** a la terminal de entrada **x**, de la operación numérica **Multiply**.
- Colóquese sobre la terminal de entrada **y** de la operación numérica **Multiply**, de **Clic Derecho**, del menú desplegable seleccione **Create » Constant**. El valor de esta constante será de 1.8.
- Cablee la terminal de salida **x\*y**, de la operación **Multiply** a la terminal de entrada de la operación **Add**.
- Colóquese sobre la terminal de entrada **y** de la operación **Add**, de **Clic Derecho**, del menú desplegable seleccione **Create » Constant**. El valor de esta constante será de 32.
- Cablee la terminal de salida **x+y** a el indicador **Grados Fahrenheit ( F )**.



**Figura 2.1** Cableado de Terminales

## Prueba.

### 1. Muestre el Panel Frontal.



**Figura 2.2** Vista del Panel Frontal

### 2. Ejecute el VI.

### 3. Cierre el VI.

A partir de este punto el alumno podrá comenzar a complementar esta aplicación con mas innovaciones.

**Fin del Programa.**

## Ejercicio 2. Calculo del Área y Perímetro de un cuadrado.

Implementación de un algoritmo en **LabVIEW** para resolver el *Área y Perímetro de un Cuadrado* con del uso de Controles, Indicadores y Constantes en LabVIEW.

Cuando usted diseña el Panel Frontal, los controles e indicadores poseen terminales correspondientes en el Diagrama de Bloques, y se utilizan como las terminales interactivas de entradas y salidas, respectivamente. Los controles son perillas, botones de presión, diales y otros dispositivos de entrada. Los indicadores son gráficos, LEDs y otros tipos de despliegue.

## Definición del Problema.

El área de un cuadrado es el producto de uno de sus lados por si mismo.

$$\text{Área} = \text{lado} * \text{lado}$$

El perímetro de un cuadrado la suma de sus cuatro lados.

$$\text{Perímetro} = \text{lado} + \text{lado} + \text{lado} + \text{lado} = 4 * \text{lado}$$

## Objetivo.

Que el alumno identifique las diferencias entre los Controles e Indicadores en el Panel Frontal de LabVIEW. Así mismo que en el Diagrama de Bloques se conozca la paleta de funciones.

## Escenario.

Considere la implementación del algoritmo para el área del cuadrado:

$$\text{Área} = \text{lado} * \text{lado}$$

Y para el perímetro del cuadrado:

$$\text{Perímetro} = 4 * \text{lado}$$

## Diseño

Este programa deberá contener las siguientes entradas y salidas:

Entradas	Salidas
lado	Área
	Perímetro

## Reto.

Llevar a cabo el desarrollo de aplicaciones similares para obtener áreas y perímetros de otras figuras, así como otro tipo de análisis.

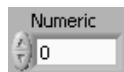
## Implementación.

1. Prepare su Software.
  - Abra LabVIEW.
  - Abra un VI en Blanco.
2. Abra el Panel Frontal.



**Consejo** Para intercambiar la vista entre el Panel Frontal y el Diagrama de Bloques, puede presionar las teclas **Ctrl + E**

3. Coloque un Control Numérico en el Panel Frontal

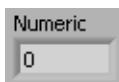


- En la ventana de Panel Frontal, de **Clic Derecho** sobre el área de trabajo, aparecerá la paleta de controles.
- Los Controles Numéricos se encuentran en la Paleta de **Controles** » **Numeric** » **Numeric Controls**
- Para colocarlo, ese se selecciona y se arrastra hacia el Panel Frontal.

- Una vez que el elemento se coloco en el Panel Frontal, podemos editar algunas de sus propiedades, tal como el nombre de la etiqueta.

Esto se puede hacer dando **Doble Clic** sobre la etiqueta del Control Numérico e ingresar el nombre que se deseé. Para este elemento ingresaremos el nombre **lado (cm)**.

#### 4. Coloque dos Indicadores Numéricos en el Panel Frontal.



- El **Indicador Numérico** se encuentra en la Paleta de **Controles » Numeric » Numeric Indicator**.



**Consejo** Para no tener que arrastrar desde la paleta de Controles el mismo elemento, basta con colocar uno solo, después sin dejar de presionar la tecla **Ctrl** y arrastrando el elemento deseado hacia cualquier parte del área de trabajo dentro el Panel Frontal, aparecerá una copia del elemento.

- El nombre de etiqueta de estos elementos lo podemos cambiar a **Área (cm<sup>2</sup>)** y **Perímetro (cm)**

#### 5. Agregar dos veces la operación básica: **Multiply** en el Diagrama de Bloques.

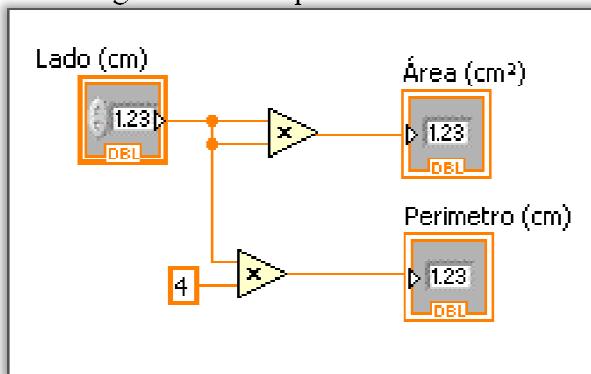


- Cambiar la vista al Diagrama de Bloques.
- La operación numérica **Multiply** se localiza en la Paleta de **Funciones » Numeric**

#### 6. Cablear las terminales de todos los elementos.

- Cablee de la terminal de salida del control **lado (cm)** a la terminal de entrada **x**, de la primera operación numérica **Multiply**.
- Cablee de la terminal de salida del control **lado (cm)** a la terminal de entrada **y**, de la primera operación numérica **Multiply**.
- Cablee la terminal de salida **x\*y**, de la operación numérica **Multiply** a el ícono del indicador **Área (cm<sup>2</sup>)**.
- Cablee de la terminal de salida del control **lado (cm)** a la terminal de entrada **x**, de la segunda operación numérica **Multiply**.

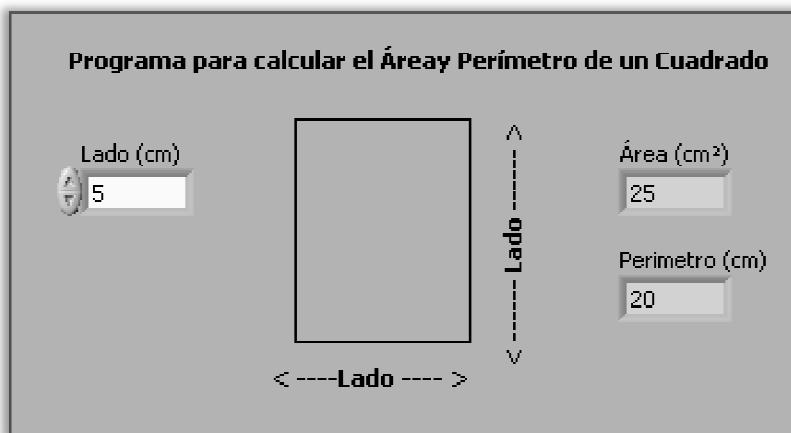
- Colóquese sobre la terminal de entrada y de la segunda operación numérica **Multiply**, de **Clic Derecho**, del menú desplegable seleccione **Create » Constant**. El valor de esta constante será **4**.
7. Complementar nuestro programa con Decoraciones, estas se encuentran en la Paleta de **Controles » Decorations**.
  8. Vista final del Diagrama de Bloques..



**Figura 2.3** Vista del Diagrama de Bloques

## Prueba.

1. Muestre el Panel Frontal.



**Figura 2.4** Vista del Panel Frontal

2. Ejecute el VI.
3. Cierre el VI.

A partir de este punto el alumno podrá comenzar a complementar esta aplicación con mas innovaciones.

**Fin del Programa.**

## Ejercicio 3. Uso de Ciclo While

Use un ciclo mientras (While) y un gráfico de forma de onda para construir un VI que demuestre temporizado de software.

The screenshot shows a LabVIEW front panel titled "Historial de Valores". On the left is a waveform plot with data points. The x-axis is labeled "Tiempo (Segundos)" and ranges from 0 to 20. The y-axis is labeled "Valor" and ranges from 0 to 1. The waveform oscillates between approximately 0.1 and 0.9. On the right side of the panel, there is a slider labeled "Retardo en Milisegundos" with a value of 0, and a red "STOP" button.

ni.com

NATIONAL INSTRUMENTS

### Definición del Problema.

La estructura del Ciclo *While* deja de ejecutar el sub diagrama solo si el valor en la condición terminal existe.

La gráfica de forma de onda es un indicador numérico especial que muestra una o más graficas (plots).

### Objetivo.

Que el alumno identifique las estructuras de los ciclos, conozca su forma de ejecutarse. También que aprenda a manejar temporizadores, gráficas de forma de ondas, y otros controles de LabVIEW.

### Escenario.

Cree un VI que genere un número aleatorio a una velocidad específica y muestre las lecturas en una grafica de forma de onda hasta que el usuario lo pare.

## Diseño

Este programa deberá contener las siguientes entradas y salidas:

Entradas	Salidas
Botón de parada (Stop)	Gráfica de forma de onda
Selector corredizo horizontal	

## Reto.

Implementar estructuras de ciclos en aplicaciones futuras, así como el uso de indicadores que ofrece LabVIEW.

## Implementación.

### 1. Prepare su Software.

- Abra LabVIEW.
- Abra un VI en Blanco.

### 2. Abra el Panel Frontal.

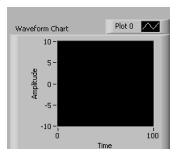
### 3. Coloque un selector corredizo horizontal en el panel frontal.

- El selector corredizo horizontal (Horizontal Pointer Slide) se encuentra en **Controls » Numeric Controls**
- Cambie el nombre de la etiqueta a “Retardo en Milisegundos”



### 4. Coloque un botón de parada (Stop) en el panel frontal.

- El botón de parada (Stop) se encuentra en **Controls » Buttons**.
- Para deshabilitar la vista de la etiqueta, de Clic Derecho sobre el botón y seleccione del menú desplegable **Visible Items » Label**



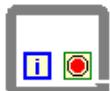
### 5. Coloque grafica de forma de onda (Waveform Chart) en el panel frontal.

- La grafica de forma de onda (Waveform Chart) se encuentra en **Controls » Graph Indicators**.
- Cambie el nombre de la etiqueta de la grafica de forma de onda (Waveform Chart) a “Historial de Valores”
- La leyenda de la grafica de forma de onda identifica la grafica como *Plot 0*. Utilice la herramienta de etiquetado para hacer tres clics en *Plot 0* en la leyenda de la grafica y escriba “**Valor**”
- Etiquete el eje X como “*Tiempo (Segundos)*” y el eje Y como “*Valor*”

-  6. Coloque un generador de números aleatorios (*Random Number*) en el Diagrama de Bloques.

- Cambie la vista al Diagrama de Bloques.
- El generador de números aleatorios se localiza en **Programming » Numeric**. El generador de números aleatorios (*Random Number*) genera como salida números entre 0 y 1.

7. Coloque un ciclo mientras (*While Loop*) en el Diagrama de Bloques.



- El Ciclo mientras (*While Loop*) se localiza en **Functions»Execution Control**.
- Haga clic y arrastre un rectángulo de selección alrededor de todos los elementos que se encuentran en el Diagrama de Bloques.
- Cablee los objetos del Diagrama de Bloques como se muestra en la figura a continuación.



**Figura 2.5** Vista del Panel Frontal

8. Coloque un VI Time Delay Express en el Diagrama de Bloques.

- Este VI se localiza en la paleta de **Functions » Programming » Timing**.
- A continuación aparecerá una ventana de dialogo e inserte 0.5. Esto asegurara que cada iteración ocurra cada medio segundo (500 ms).

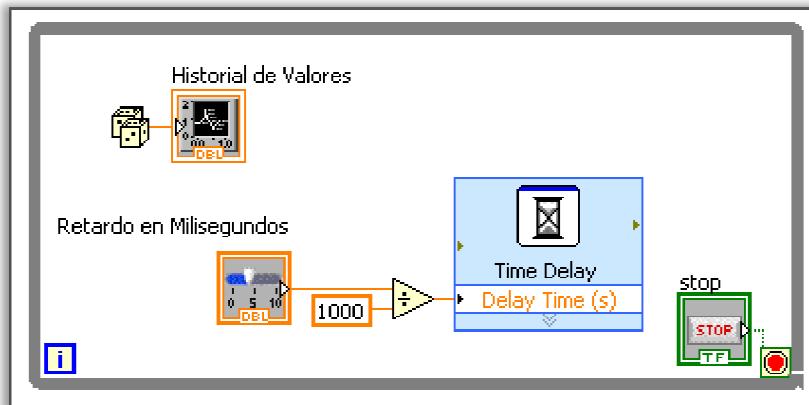


**Figura 2.6** Ventana de Diálogo del Time Delay

- Divida el retardo en milisegundos (millisecond delay) por 1000 para obtener tiempo en segundos.



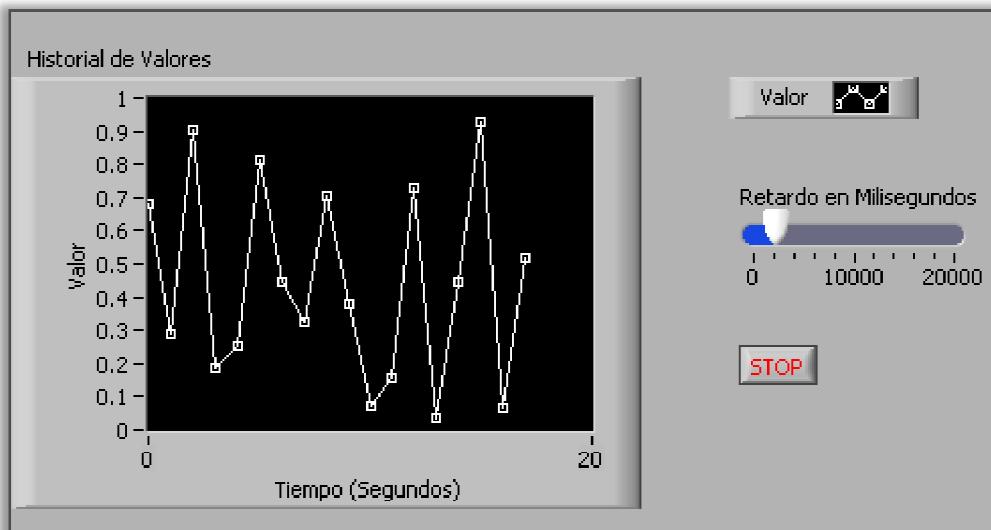
- Conecte la salida de la división a la entrada Delay Time (s) del VI Time Delay Express. Esto le permitirá ajustar la velocidad de la ejecución desde el selector corredizo en el panel frontal.
9. Guarde el VI



**Figura 2.7** Vista del Diagrama de Bloques

## Prueba.

1. Muestre el Panel Frontal.



**Figura 2.8** Vista del Panel Frontal

2. Ejecute el VI.
3. Cierre el VI.

Implementar otras configuraciones en la grafica de forma de onda (Waveform Chart).

**Fin del Programa.**

## Ejercicio 4. Clusters de Error y su manipulación

Crear un VI que calcule la raíz cuadrada de un número. Identificar si es mayor, igual o menor que cero. Generar un error cuando el numero sea menor que cero.

The screenshot shows a LabVIEW VI titled "Entrada para obtener Raiz Cuadrada". It has two main components: a "Raiz Cuadrada" output terminal and an "error in (no error)" cluster. The "error in (no error)" cluster contains three sub-components: "status", "code", and "source". Below this is another "error out" cluster with similar sub-components. The LabVIEW interface includes a "ni.com" watermark at the bottom left and the National Instruments logo at the bottom right.

### Definición del Problema.

Los Clusters agrupan componentes iguales o desiguales juntos. Los Componentes de los Clusters pueden ser de diferentes tipos de datos. Todos los elementos de un Cluster deben ser ya sea Indicadores o Controles. No se puede tener un Control de Cadena y un Indicador Booleano. Se puede considerar como un conjunto de alambres agrupados en un cable.

### Objetivo.

Que el alumno aprenda a manejar los Cluster de LabVIEW, identifique sus tipos y los implemente en su programación.

### Escenario.

Cree un VI que calcule la raíz cuadrada de un numero. Si el numero es mayor o igual a cero, el VI debe devolver el valor de la raíz cuadrada y no generar ningún error. Si el numero es menor que cero el programa debe devolver un resultado con valor de -9999.90 e insertar un error dentro del cluster para errores.

## Diseño

Este programa deberá contener las siguientes entradas y salidas:

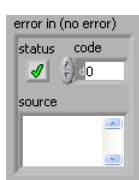
Entradas	Salidas
Dato numérico	Dato numérico

## Reto.

Implementar estructuras de ciclos en aplicaciones futuras, así como el uso de Clusters de Error de LabVIEW.

## Implementación.

1. Prepare su Software.
  - Abra LabVIEW.
  - Abra un VI en Blanco.
  
2. Abra el Panel Frontal.
  
3. Coloque un Control Numérico en el Panel Frontal.
  - El Control Numérico se encuentra en la paleta de **Controls > Numeric > Numeric Control**
  
4. Coloque un Indicador Numérico en el Panel Frontal.
  - El Control Numérico se encuentra en la paleta de **Controls > Numeric > Numeric Indicator**
  
5. Coloque un Error In 3D.ctl en el Panel Frontal.
  - El Error In 3D.ctl se encuentra en la paleta de **Controls > Array, Matrix & Cluster > Error In 3D.ctl**
  
6. Coloque un Error Out 3D.ctl en el Panel Frontal.
  - El Error Out 3D.ctl se encuentra en la paleta de **Controls > Array, Matrix & Cluster > Error In 3D.ct**



7. Construir el siguiente Diagrama de Bloques.

- Colocar una estructura Case en el Diagrama de Bloques. Se localiza en la paleta de **Functions»Structures»Case Structure**.

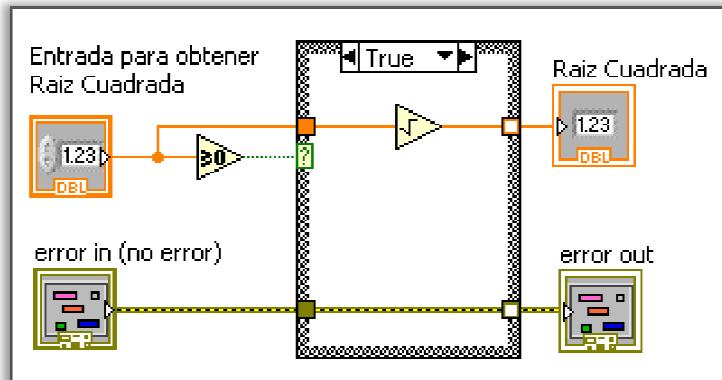


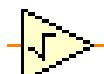
Figura 2.9 Vista del Diagrama de Bloques

- Coloque una **Greater or Equal to 0?** de la paleta **Functions » Comparision » Greater or Equal to 0?**



En el Caso *Verdadero* (True Case):

- Coloque una función **Root** de la paleta **Functions»Numeric**



En el Caso *Falso* (False Case):

- Coloque una **Constante Numérica**. Se localiza en la paleta de **Functions » Numeric » Numeric Constant** y asignar el valor de -9999.90.



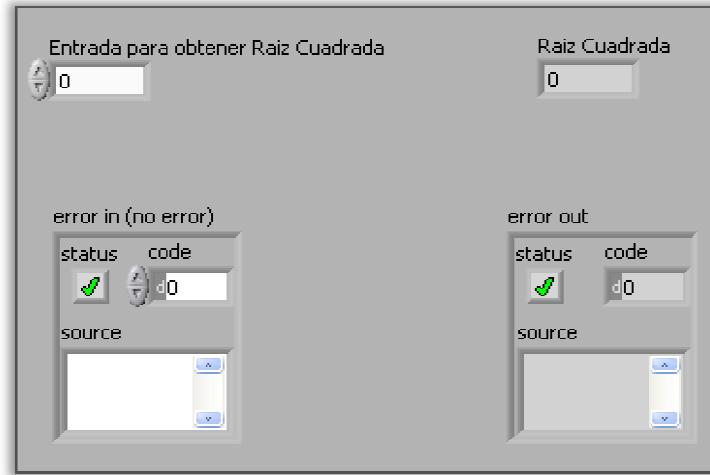
- Coloque una función **Bundle By Name**. Se localiza en la paleta de **Functions»Cluster, Class & Variant » Bundle By Name**. Conecte desde Error in hacia la terminal **Input Cluster** de Bundle by Name para hacer que el status se muestre. Cablee desde el indicador del Error Out hacia la salida **Output cluster** del Bundle By name.



- Redimensione el **Bundle By Name** para mostrar tres elementos (Status, Code y Source) y cree constantes para estos. Para crear la constante coloque en la terminal de entrada, de Clic Derecho con el mouse y seleccione Create Constant. Para Status deberá ser True, para Code, la constante será 5008 y para source

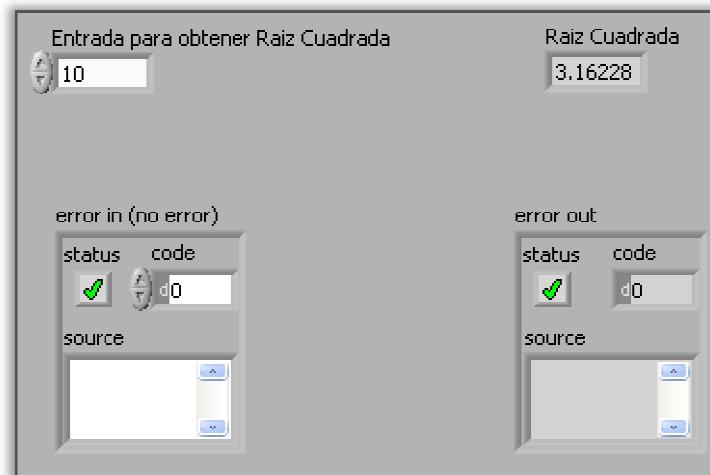
## Prueba.

1. Muestre el Panel Frontal.



**Figura 2.10** Vista del Panel Frontal sin ejecutar

2. Ejecute el VI.



**Figura 2.11** Vista del Panel Frontal al ejecutarse la primera vez

3. Cierre el VI.

**Fin del Programa.**

## Ejercicio 5. Máquina de estado Simple

Cree un VI utilizando la arquitectura del maquina de estado que simula una secuencia simple de prueba.

### Definición del Problema.

Una maquina de estados, la cual se define de forma mas precisa como una maquina finita de estados, consiste de un conjunto de estados y una función de transición que conduce hacia el siguiente caso.

### Objetivo.

Que el alumno aprenda a manejar máquinas de estado y las implemente en aplicaciones futuras.

### Escenario.

Crearemos un VI usando la arquitectura de Maquina de Estado. El VI tendrá un estado de ocio (idle), en donde espera por entradas. Cuando el usuario presiona un botón, el VI se ira al estado 1 (State 1). El estado 1 genera una caja de dialogo que le permite al usuario proceder a ir al estado 2 (State 2) o volver a empezar. Desde el estado 2 el usuario puede decidir cerrar el programa o iniciar nuevamente.

## Diseño

Este programa deberá contener las siguientes entradas y salidas:

Entradas	Salidas
Selección del Usuario	Mensajes

## Reto.

Desarrollar aplicaciones futuras utilizando las Maquinas de Estado que ofrece LabVIEW.

## Implementación.

1. Prepare su Software.

- Abra LabVIEW.
- En lugar de empezar desde cero, utilizaremos una Plantilla de VI para crear nuestra máquina de estado. Desde el menú **File**, en lugar de seleccionar **New VI**, seleccione **New**, y haga clic en **Start from template**. Busque en el Menú **VI»From Template»Frameworks»Dewsgin Patterns»Estándar State Machine** y seleccione **OK** para abrirlo. Esto abrirá una plantilla simple de una máquina de estado. Examine la plantilla, y guárdela en otro directorio antes de empezar a trabajar en ella.

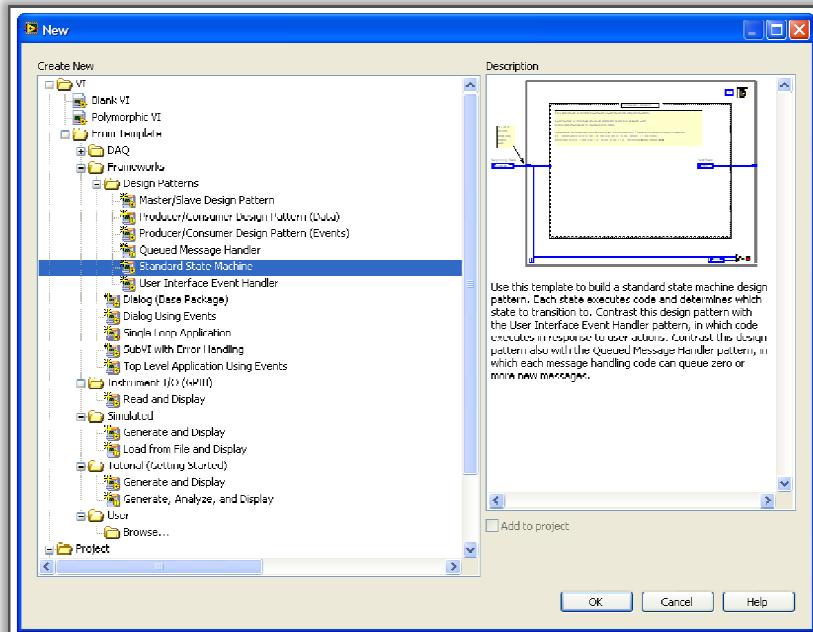
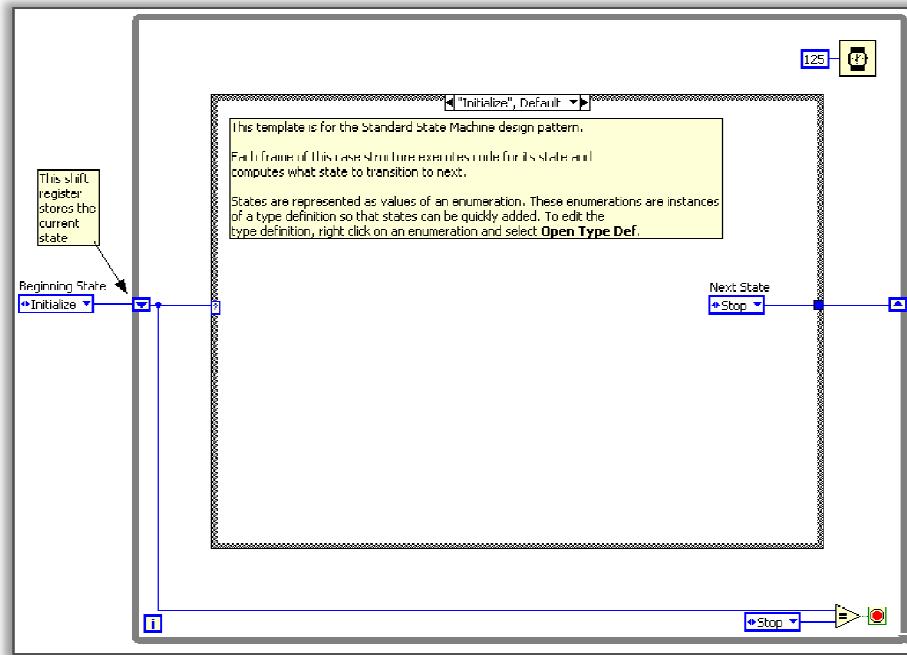


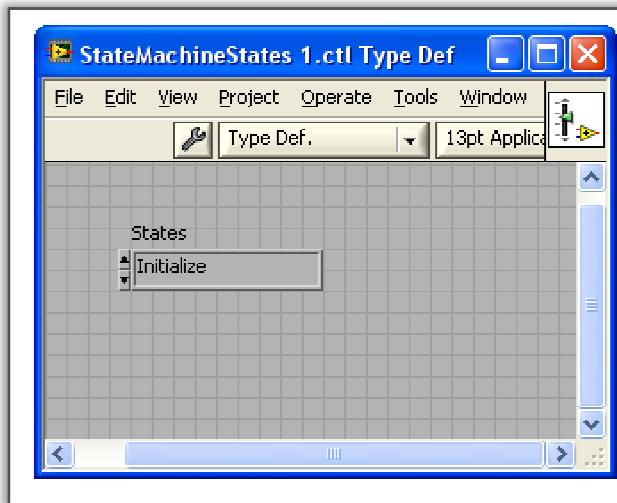
Figura 2.12 Plantilla de creación de LabVIEW

2. Abra el Diagrama de Bloques.



**Figura 2.13** Plantilla de Diagrama de Bloques de VI

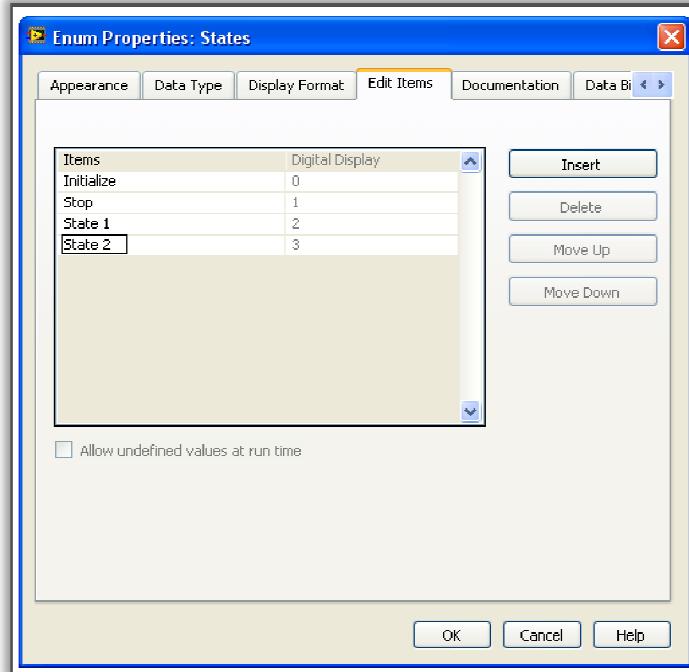
3. Haga clic derecho sobre la constante *enum* etiquetada **Next State** y seleccione **Open Type Def**. Posteriormente al seleccionar esto se abrirá un Panel Frontal:



**Figura 2.14** Plantilla de VI

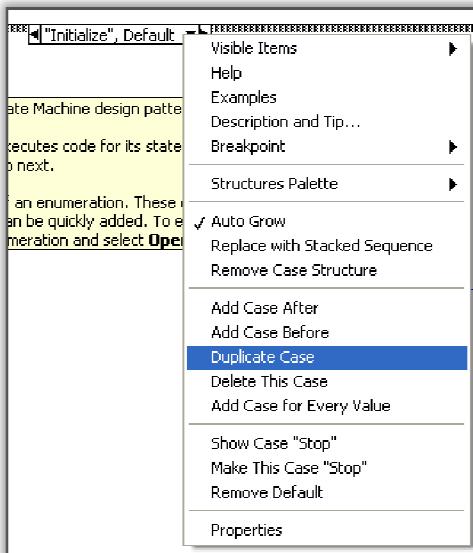
4. En el panel frontal de StateMachinesStates.ctl escriba Def VI, haga clic derecho en el control de estados enum y elija **Edit Items**.

5. Añada dos estados más. Nómbralos “State 1” y “State 2”



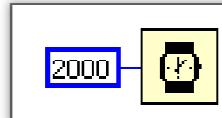
**Figura 2.15** Editar Elementos

6. Cierre State Machines.ctl y escriba Def Front panel y salve el control con el nombre por defecto cuando se le pregunte.
7. Haga clic derecho en la etiqueta del selector de casos (Case Selector Label) de la estructura de casos y elija Duplicate case. Haga esto una vez más de manera que hallan cuatro casos: Initialize, State 1, State 2, y Stop.



**Figura 2.16** Agregar Elementos

8. Cambie el valor conectado a la función Wait a 2000.

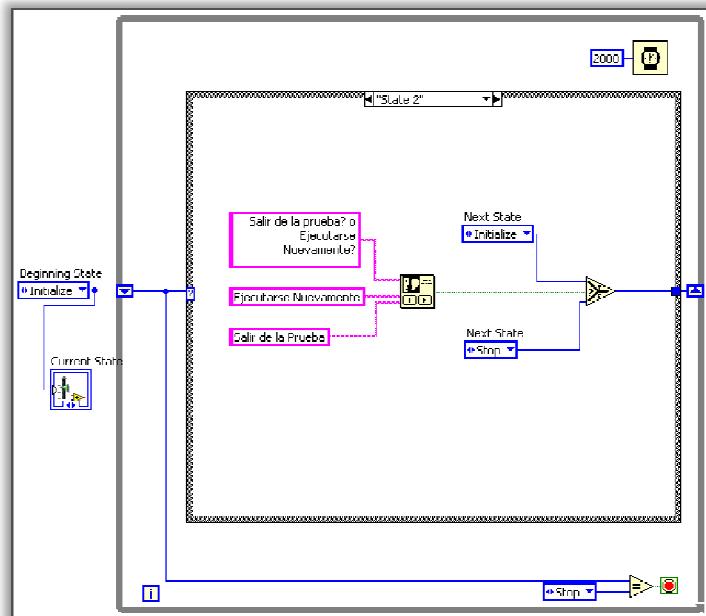


**Figura 2.17** Función Wait

9. Haga clic derecho en el registro de desplazamiento (shift register) en el costado izquierdo del ciclo mientras (while loop) y cree un indicador. Cambie el nombre de este indicador a “Current State”.
10. En “Initialize”, caso por defecto (Default case) coloque una function One Button Dialog y conecte una constante de cadena de caracteres a la entrada de **Message**. Escriba “Inicio de la prueba...” en la constante de cadena de caracteres.
11. Cambie la constante enum etiquetada Next State a “State 1”.
12. Cambiese al próximo estado en la estructura de casos (“State 1”) y cambia la constante enum etiquetada Next State a “State 2”.
13. Cambiese al próximo caso (“State2”) y añada el siguiente código:



- Coloque una función Select y conecte dos constantes enum  
*(Sugerencia: Copie las constantes enum desde uno de los casos previos)*
- Coloque un dialogo de dos botones (Two Button Dialog) y conecte las constantes como se muestra abajo.



**Figura 2.18** Función Wait

## Prueba.

1. Muestre el Panel Frontal.
2. Ejecute el VI.
  - ❑ Mensaje cuando se ejecuta por primera vez el VI



Figura 2.19 Primer mensaje

- ❑ Mensaje después de presionar el botón OK



Figura 2.20 Segundo mensaje

3. Cierre el VI.

**Fin del Programa.**

**Propuesta de Proyecto Final:  
Juego del Simón**

Vista de Distintos Programas del Simón en LabVIEW

ni.com

NATIONAL INSTRUMENTS

## Proyecto Final: Juego del Simón

### Definición del Problema.

El *juego del Simón* (Ver Figura 1), es un juego electrónico creado por Milton Bradley en 1978. Tuvo un gran éxito durante los 80. Tiene forma de disco, en una de sus caras se puede ver cuatro cuadrantes, cada uno con un color: verde, rojo, azul y amarillo en su versión original. Su nombre se debe por el conocido juego tradicional del mismo nombre: “Simón dice”.

El juego de forma aleatoria va iluminando los cuadrantes de colores, y a la vez que se ilumina cada cuadrante emite un sonido propio. Después de esperar, el usuario debe ir introduciendo la secuencia mostrada en el orden correcto, ayudándose de su memoria visual y sonora. Si lo consigue, éste responderá con una secuencia más larga, y así sucesivamente. Si falla, el usuario debe volver a empezar. Los distintos niveles de dificultad van aumentando la velocidad de la secuencia a repetir.

*Imagen del Juego del Simón, www.vivienlos80s.com*

## Objetivo.

Evaluar los conocimientos del alumno, adquiridos durante el curso de Lógica de Programación.

Este ejercicio se recomienda como proyecto final para el curso de Lógica de Programación, ya que comprende los tópicos vistos durante la clase.

## Escenario.

Construir un VI que simule el juego electrónico del Simón, por medio de la representación de sus 4 focos, así como la generación de la secuencia aleatoria para encender estos, y el funcionamiento del Simón.

A continuación se lista una gran variedad de elementos, con las que el alumno puede implementar en su diseño

En el **Panel Frontal**, se puede incluir:

- Led redondos (El uso de colores en estos)
- Botón de Stop y Botón de OK
- Botón para Iniciar el programa
- Decoraciones

En el **Diagrama de Bloques**, se puede implementar:

- Estructuras de Eventos:
  - While Loop
  - Flat Sequence
  - Case Structure
  - For Loop
- Tipos de datos: Booleanos
- Estructuras de arreglos
- Temporizadores

Y muchas cosas más que el alumno puede ir mejorando en base a la práctica y experiencia que vaya adquiriendo.

## Reto.

Que el alumno que anexe mejoras al juego del Simón y que implemente algoritmos para otros juegos de mesa en LabVIEW.

## Diseño

Se propone el desarrollo del siguiente pseudocódigo como base para el desarrollo del juego del simón.

```

Inicio
Mientras (BotonStop=False)
{
  Mientras (Secuencia ≠ “Terminar”)
  {
    CASO Secuencia HACER
      Secuencia = “Generar Secuencia”
      {
        x=random[1]
        num=x*3
        y=redondear(num)
        For i=0,3,i++
          Si y=i hacer
            Si
              {
                arreglo=[c,y]=true
              }
            Sino
              {
                arreglo=[c,y]=false
              }
            Secuencia=“Desplegar Secuencia”
          }
        Secuencia=“Desplegar Secuencia”
        {
          For i=0,c,i++
            For j=0,c,i++
              Imprimir arreglo [i,j]
            Secuencia=“Datos de Usuario”
          }
        Secuencia=“Datos Usuario”
        {
          For i=0,4,i++
            If Boolean1.ValueChange=True
              Arreglo[c,i]=True
            Si arreglo[sua]=arreglo hacer
              Si
                Secuencia=“Terminar”
              Sino
                Secuencia “Generar Usuario”
            }
          Secuencia=“Terminar”
          {
            Mensaje=“Juego Terminó”
          }
        }
      }
    Fin
}

```

## Implementación.

1. Prepare su Software.

### Software

- Abra LabVIEW.
- Abra un VI en Blanco.

2. Abra el Diagrama de Bloques.

3. Coloque una Estructura de Eventos *While Loop*.



- Consejo** El *While Loop* siempre se ejecuta al menos una vez.

El *While Loop* ejecuta el subdiagrama hasta que el terminal de condición, un terminal de entrada, reciba un valor Booleano específico. El comportamiento y apariencia por defecto del terminal condicional es **Stop if True**, mostrado a continuación.



Cuando un terminal condicional es **Stop if True**, el *While Loop* ejecuta su subdiagrama hasta que el terminal condicional reciba un valor TRUE.

El **Terminal de Iteración** (Un terminal de salida), mostrado a continuación, contiene el número de iteraciones realizadas.



4. Coloque un **Stop Button** en el Panel Frontal.



- El **Stop Button** se localiza en la paleta de **Controles » Boolean**.
- El botón de Stop aparecerá en su Diagrama de Bloques con forma de Icono. Para cambiar la vista puede dar Clic derecho sobre, y en el menú que se desplegará deshabilitar la opción **View as Icon**.

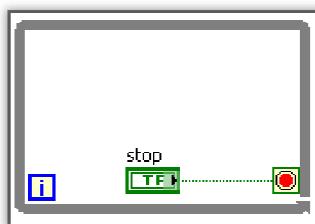
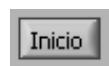


Figura 2.21 Diagrama de Bloques

5. Coloque un **OK Button** en el Panel Frontal.

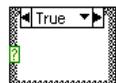


El *OK Button* se localiza en la paleta de **Controles » Boolean**.



- Debe colocarlo dentro del *While Loop*
- Al igual que con el Botón de *Stop*, podemos cambiar la vista de estos dando **Clic Derecho** sobre el Botón en el diagrama de Bloques. También estos botones se pueden personalizar en el Panel Frontal, y se puede habilitar / deshabilitar la vista de sus etiquetas.

6. Coloque la **Estructura Case** en el Diagrama de Bloques

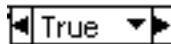


- La estructura Case se localiza se localiza en la paleta de **Funciones » Structures**.



**Consejo** La **Estructura Case** posee dos o mas subdiagramas o casos. Donde solo un diagrama es visible a la vez, y la estructura solo ejecuta un caso a la vez.

La etiqueta de selector de caso en la parte superior de la estructura Case, mostrada a continuación, contiene en medio el nombre del valor selector que corresponde a caso y flechas de incremento y decremento en cada lado.



Cablee un valor de entrada, o selector, al terminal selector, mostrado a continuación, para determinar el caso a ejecutar.



7. Cablee la terminal del **Ok Button** a la terminal **selector** de la Estructura case.

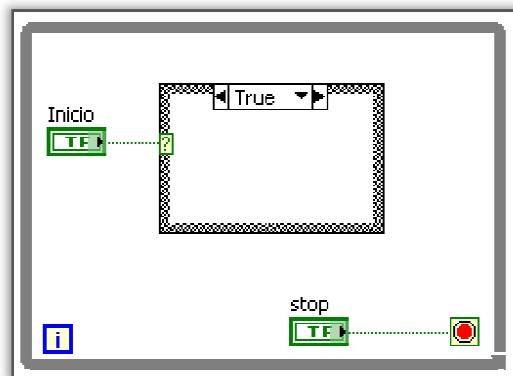
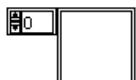


Figura 2.22 Diagrama de Bloques

8. Coloque una Estructura de Eventos *While Loop* dentro de la **Estructura Case**, en la selección **True**. La selección **False** no se utilizará.

9. Coloque un **Array Constant** dentro de la Estructura Case.



- El **Array Constant** se localiza en la paleta de **Funciones » Array**.

10. Dentro del *Array Constant* coloque un **False Constant**, que es una constante de tipo Falsa.



- Esta *False Constant*, se localiza en la paleta de **Funciones » Boolean**.

11. Despliegue el Array constant para mostrar dos elementos. Con esto representaremos un Arreglo de dos elementos con valores *False*.



- Se coloca el puntero del mouse sobre el lado izquierdo del Array Constant para delimitar el tamaño de este, se selecciona y el mouse se arrastra hasta elegir el tamaño adecuado.

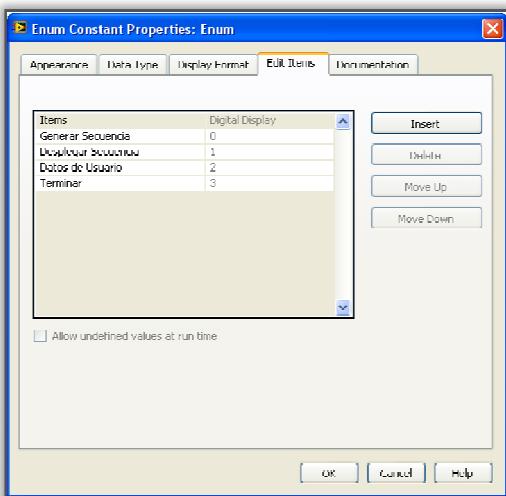
12. Coloque un **Enum Constant** dentro de la Estructura Case y personalícela.



- La **Enum Constant** se encuentra en la paleta de **Funciones » Numeric**.
- Para ingresar los datos que esta constante enumerada manejará, se debe ingresar en sus propiedades. Dando **Clic Derecho** sobre la constante y seleccionar **Properties**. Dar Clic a la pestaña de **Edit Items**, una vez en esta sección, se pueden agregar los elementos que la constante enumerada manejará.
- En este caso manejaremos cuatro etapas para que nuestro Simón se lleve a cabo:
  - Generar Secuencia
  - Desplegar Secuencia
  - Datos de Usuario
  - Terminar



La configuración deberá quedar de la siguiente manera:



**Figura 2.23** Diagrama de Bloques

13. Crear **Shift Registers** de los *elementos Enum Constant* y *Array Constant*, recien creados, en los bordes de la nueva estructura *While Loop*.



- Para crear un **Shift Register**, de da **Clic Derecho** sobre el borde de la estructura *While Loop*, y se selecciona la opción **Add Shift Register**. En este caso, agregamos dos de estas opciones.
- Cableamos las salidas del *Enum Constant* y *Array Constant*, a los Shift Register del Ciclo *While Loop*.



**Consejo** Un **Shift Register** contiene la información de transición de estado. Una vez que se conectan a un tipo de dato, su color se ve alterado, según el tipo de este.

14. Agregar una **Estructura Case** dentro de la Estructura *While Loop*.

15. Los casos de la Estructura Case dependerán del *Enum Constant*.

- Cablear el *Shift Register* del *Enum Constant* a el Case Selector de la Estructura *Case*.
- Dar **Clic Derecho** sobre el borde de la Estructura *Case*, del menú desplegable seleccionar la opción **Add Case For Every Value**.

16. Dentro del caso de Generar Secuencia, agregar la función **Wait Until Next ms Multiple**.



- Esta función se encuentra en la paleta de **Funciones » Timing**.
- Una vez que la insertamos, damos **Clic Derecho** sobre su punto de conexión izquierda, y seleccionamos **Create » Constant**. Ingresamos 300.



**Consejo** La función **Wait Until Next ms Multiple**, monitorea un contador en milisegundos y espera hasta que este alcance un múltiplo de la cantidad que usted especifica. Use esta función para sincronizar actividades. Coloque esta función en un ciclo para controlar su velocidad de ejecución.

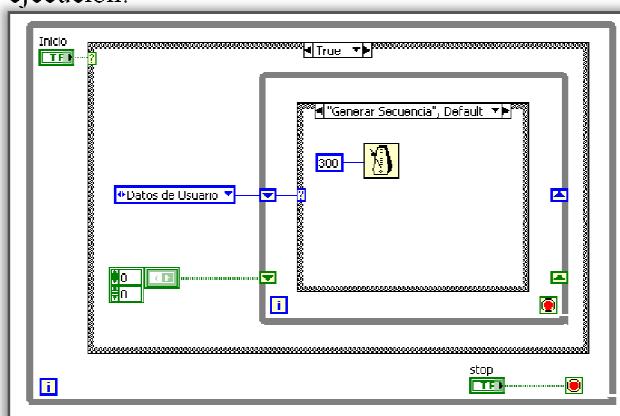


Figura 2.24 Diagrama de Bloques

17. Genere un numero aleatoriamente, y este valor será agregado a un arreglo.



- Seleccione la función **Random Number**. Esta se localiza en **Mathematics » Numeric » Random Number (0-1)**. Colóquela en el Diagrama de Bloques. Random Number (0-1) genera un numero aleatorio entre 0 y 1.



- Seleccione la función **Multiply** y colóquela en el diagrama de bloques. La función **Multiply** multiplica el numero aleatorio por y para producir un numero aleatorio entre 0 y Y.



- Haga Clic Derecho en la entrada **Y** de la función **Multiply**, seleccione **Create » Constant** desde el menú rápido, ingrese 3 y presione la tecla <Enter> para crear una constante numérica.



- Coloque la función **Round To Nearest** en el diagrama de Bloques. Esta función redondea el numero aleatorio al entero mas cercano.

- Coloque cuatro funciones **Equal?** en el diagrama de Bloques. Estas funciones permitirán comparar de 0 a 4 los valores que obtengas del conjunto de funciones anteriores.



- Haga Clic Derecho en la entrada **X** de la función **Equal?** , seleccione **Create » Constant** desde el menú rápido, ingrese 0 (Para el primer caso, hasta llegar a 4) y presione la tecla <Enter> para crear una constante numérica.



- Coloque la función **Build Array** en el diagrama de Bloques. Esta función nos permite concatenar múltiples arreglos, o agregar elementos a un arreglo de n dimensiones. A esta función cablearemos la salida de las 4 funciones **Equal?** colocadas anteriormente.

- Coloque la función **Insert To Array** en el diagrama de Bloques. Esta función permite insertar un elemento o un subarreglo en un arreglo de n dimensiones en el punto del índice que usted especifique. A esta función cablearemos la salida *Appended Array* de la función *Build array* a la entrada **New element/Subarray**. Y también cablearemos a la entrada **Array**, el Shift Register que habíamos colocado en el Borde del *Case Structure*, proveniente del *Arreglo Booleano* de dos dimensiones creado anteriormente. Finalmente la terminal de salida **Output Array** la cablearemos al **Shift Register** correspondiente.

- Colocar una **False Constant** dentro de la Estructura Case y conectar su terminal a la del **Stop if True** del Ciclo While externo.

- Colocar una copia del **Enum Constant** dentro de la Estructura Case y conectar su terminal a la del **Shift Register** que se encuentra en el Ciclo While.

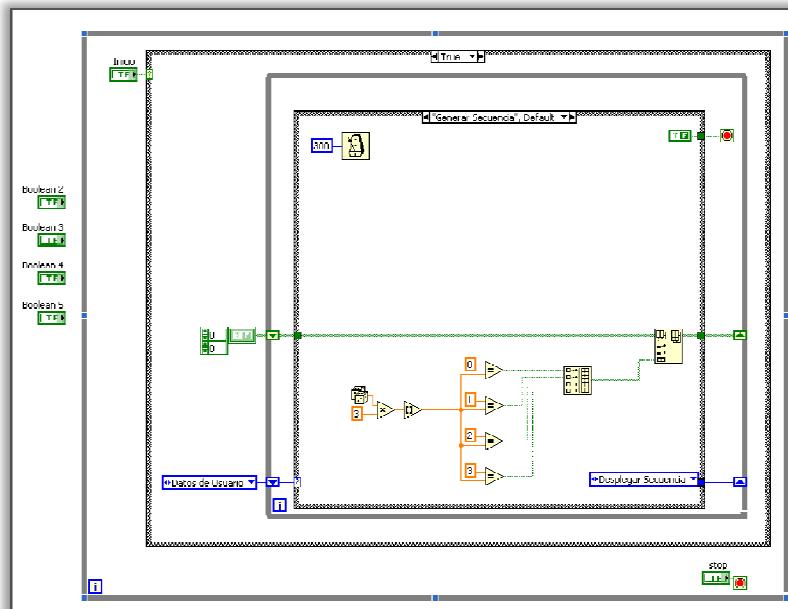


Figura 2.25 Diagrama de Bloques

18. Mostrar el contenido de todos los índices del Arreglo.



- Cambiar la Selección de la Estructura Case a "Desplegar Secuencia".
- Coloque una Estructura de Eventos **For Loop**. Esta estructura se localiza en la paleta de **Funciones > Structures**.

**Consejo**

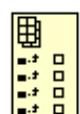
El *For Loop* se ejecuta un numero fijo de veces. El valor en el terminal de cuenta (Una terminal de entrada), mostrada a continuación. Indica cuantas veces se repite el subdiagrama.

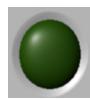


El terminal de iteración (Una terminal de salida), mostrada a continuación, contiene el numero de iteraciones realizadas.



- Dentro de la Estructura *For Loop*, se colocara una estructura **Flat Sequence**. Colocarse sobre el borde derecho de la estructura, dar Clic Derecho y seleccionar **Add Frame After**, ya que se trabajara con dos etapas.
- Agregar dos funciones **Wait Until Next ms Multiple**. Una dentro del primer frame del *Flat Sequence* (Con una constante de 700m), y la otra dentro del segundo (Con 400 ms).
- Colocar la función **Index Array** dentro del primer Frame del *Flat Sequence*. Colocar el puntero del mouse en la parte inferior de este, de tal manera que su tamaño se ajuste para manipular cuatro elementos.
- De la terminal del *Shift Register* que viene del Arreglo de dos dimensiones de tipo de datos booleanos, cablear a la entrada **Array** de la función **Index Array**. Y tambien cablear la terminal del Shift Register a la terminal a la terminal de salida de la Estructura case que va hacia el Shift Register de Salida.





- Coloque en su Panel Frontal los cuatro Leds que vaya a utilizar como Indicadores y Controles para simular el juego del Simón. Se recomienda utilizar **Round Leds**, y personalizar su color, para mejorar el desarrollo de su aplicación. Estos se encuentran en la paleta de **Controles** » **Boolean** » **Round Leds**.

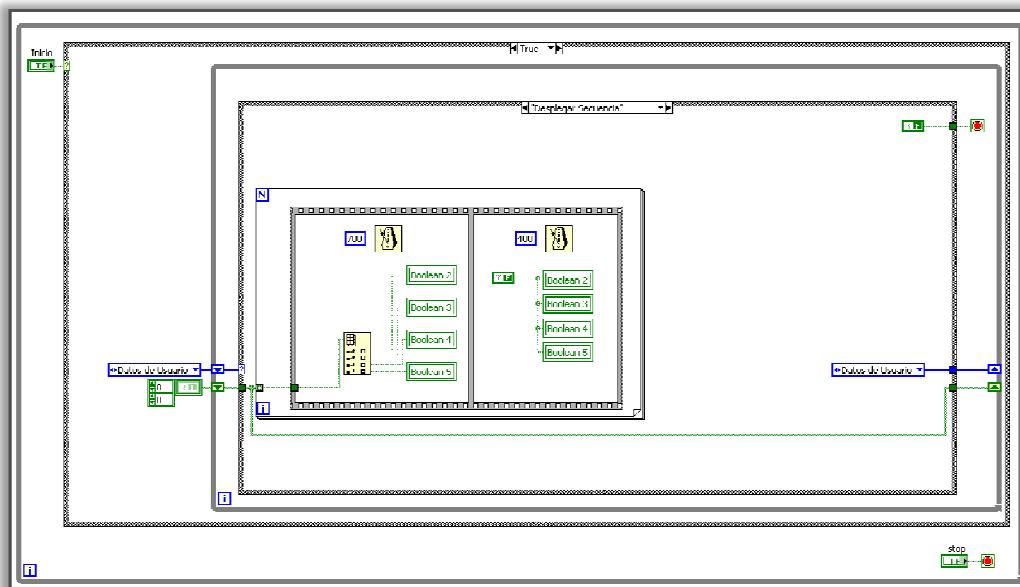
**Consejo**

Para personalizar los Round Leds, se puede dar Clic Derecho sobre el objeto, del menú desplegable seleccione **Properties** » pestaña de **Appearance** » **Colors**. Dar Clic Izquierdo sobre el cuadro de Color y elegir el adecuado para el desarrollo de su simulación en los estados **ON** y **OFF**.

Boolean 2



- En el Diagrama de Bloques, seleccionar los elementos booleanos correspondientes a cada *Round Led*, dar **Clic Derecho**, y seleccionar del menú desplegable **Create** » **Local Variable**, y colocar estas variables locales dentro del primer frame del Flat Sequence.
- Cablear cada variable local a las terminales de salida *element* del *Index Array*.
- Se hará una copia de las cuatro variables locales en el segundo frame del Flat Sequence. Esta se puede hacer haciendo presionar la tecla <Ctrl> y dar Clic Izquierdo sobre la variable local, sin soltarlos, arrastrar la variable local hasta posición deseada.
- Colocar un False Constant dentro del segundo frame del Flat Sequence y cablear a las 4 variables locales
- Colocar una False Constant dentro de la Estructura Case y conectar su terminal a la del Stop if True del Ciclo While externo.
- Colocar una copia del **Enum Constant** en la Estructura Case y conectar su terminal a la del Shift Register que se encuentra en el Ciclo While.



**Figura 2.26** Diagrama de Bloques

19. Una vez que se visualizo aleatoriamente la secuencia del Simón, el usuario deberá ingresar sus datos y estos se ingresaran a un arreglo para determinar si la secuencia es correcta o no.



- Colocar una estructura For Loop dentro la estructura Case.



- Agregar la función **Wait Until Next ms Multiple.**, dentro de la estructura For Loop . Dar un valor a una constante para esta función de 700ms.



- Colocar una **Event Structure** dentro de la estructura For Loop.



### Consejo

La Event Structure tiene uno o mas subdiagramas, o eventos, donde cada uno ejecuta cuando la estructura se ejecuta. La Event Structure espera hasta que el evento ocurre, luego ejecuta el caso apropiado de cada evento. De Clic Derecho sobre el borde de la estructura para agregar nuevos casos. En la terminal Time Out se especifica el numero de milisegundos que la estructura debe esperar cuando un evento ocurra. El numero por default es -1, y este indica que nunca haya tiempo de espera.

- Agregar Casos al Event Structure. Dar **Clic Derecho** sobre el borde » **Add Event Case**. Para los 4 Controles Booleanos elegir el evento **Value Change**, y presionar el signo <+> de **Insert**, en la ventana de *Edit Events*. Quedando de la siguiente manera:

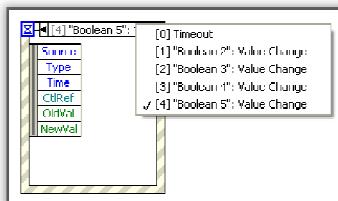
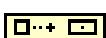


Figura 2.27 Event Structure



- Al seleccionar el caso Boolean 2: Value Change, mover hacia dentro del Event Structure en este caso, nuestro elemento Boolean 2, que teníamos fuera de toda de nuestra estructura. Hacer esto para cada caso.
- Colocar a la derecha del Event Structure una función **Build Array**, la cual adaptaremos para 4 elementos, y donde según el caso, conectaremos las terminales del event structure.

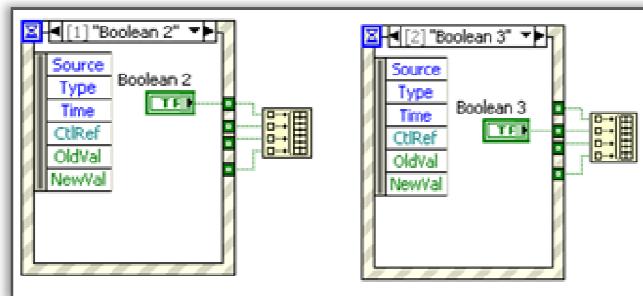
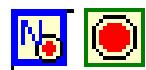


Figura 2.28 Event Structure



- Seleccionar el borde de la estructura **For Loop** y seleccionar **Conditional Terminal**. De esta manera nos aparece un **Stop If True** en el For Loop y en el Terminal Count, un signo pequeño de **Stop If True**.



- Colocar una función **Not Equal**. A esta cablearemos como terminales de entrada, un cable proveniente del Shift Register del Arreglo Doble y a la otra terminal, el proveniente de la terminal de salida del Build Array



- Colocar una función **Or Array Elements**, donde la terminal de entrada será la terminal de salida de la función Not Equal.



- Colocar una estructura Flat Sequence a la derecha de la función Or Array Elements.



- De la terminal de Salida del Or Array Elements, conectar un nodo en el borde de la Flat Sequence, para llevar a cabo un orden.



- Dentro de la Flat Sequence, colocar un **Wait Until Next ms Multiple** con una constante de 300ms

- Colocar un **False Constant**.



- Colocar cuatro variables locales, correspondientes a los cuatro leds booleanos, dentro de la estructura Flat Sequence.



- Cablear de la terminal de salida de la False Constant a los cuatro indicadores (representados como variables locales de los 4 leds).



- Cablear de la Salida del **Or Array Elements** al **Stop if True** y a un borde del lado derecho de la Estructura *For Loop*.



- Colocar un **False Constant** dentro de la Estructura *Case* y conectar su terminal a la del *Stop if True* de la estructura *While Loop* externo.



- Colocar dos copias del **Enum Constant** dentro de la Estructura *Case*. Configurar la primera para la secuencia “Terminar”, y la segunda para la secuencia “Generar Secuencia”.



- Colocar una función **Or Array Elements** a la derecha de la Estructura *For Loop*. Del nodo de conexión que se encuentra en el borde del For Loop (Proveniente del *Or Array Elements*), conectar a la entrada Boolean Array del *Or Array Elements*.



- Colocar una función **Select** a la derecha de la función *Or Array Elements*. De la salida Logical OR de la función *Or Array Elements* conectar a la entrada “*s*” de la función *Select*.

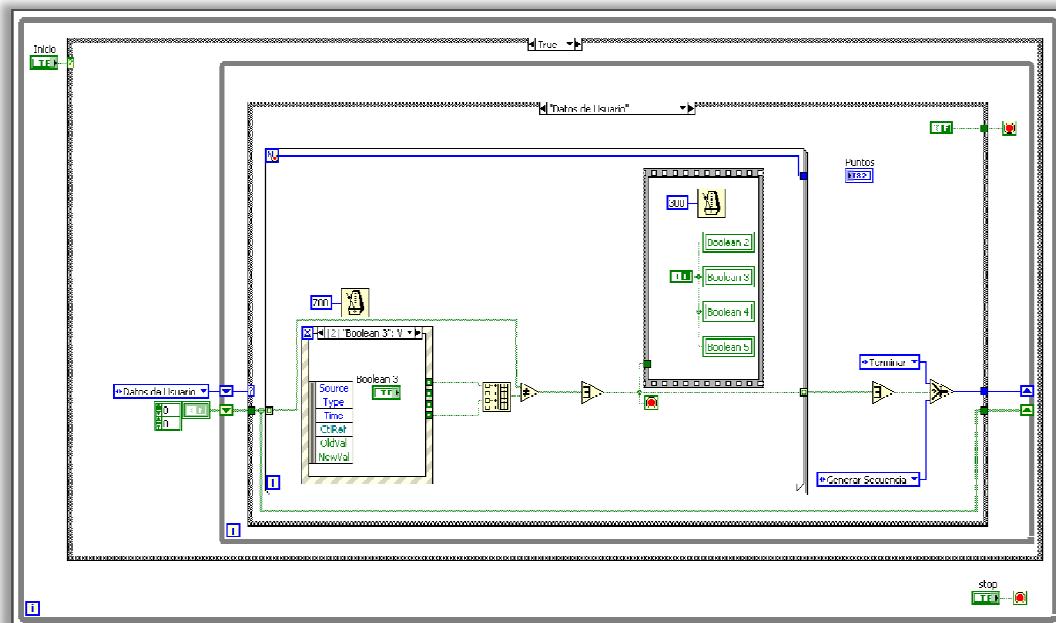


- Conectar el *Enum Constant* a la entrada “*T*” de la función *Select*. El segundo *Enum Constant* con valor de secuencia “Generar Secuencia”, conectarlo a la entrada “*F*” de la misma función.



- Conectar la salida “*s? t:f*” de la función *Select* al nodo del Shif Register que del inicio proviene del *Enum Constant*.

- ❑ Conectar de lado a lado los nodos correspondientes a la estructura de arreglo. Cabe de mencionar que al inicio se encontrara otro nodo que va a la estructura For Loop.
- ❑ Dirigirse al Panel Frontal y colocar un **Numeric Indicator**. Este se encuentra en la paleta de **Controles » Modern » Numeric Indicator**. Este indicador nos mostrará nuestro puntaje.
- ❑ Conectar la salida de la terminal Loop Count de la estructura For Loop interna, hacia la entrada del indicador. Cambiar la representación del tipo de datos del indicador a I32, y al realizar la conexión dar Clic derecho sobre el nodo que se crea automáticamente sobre el borde de la estructura For Loop y seleccionar **Disable Indexing**.

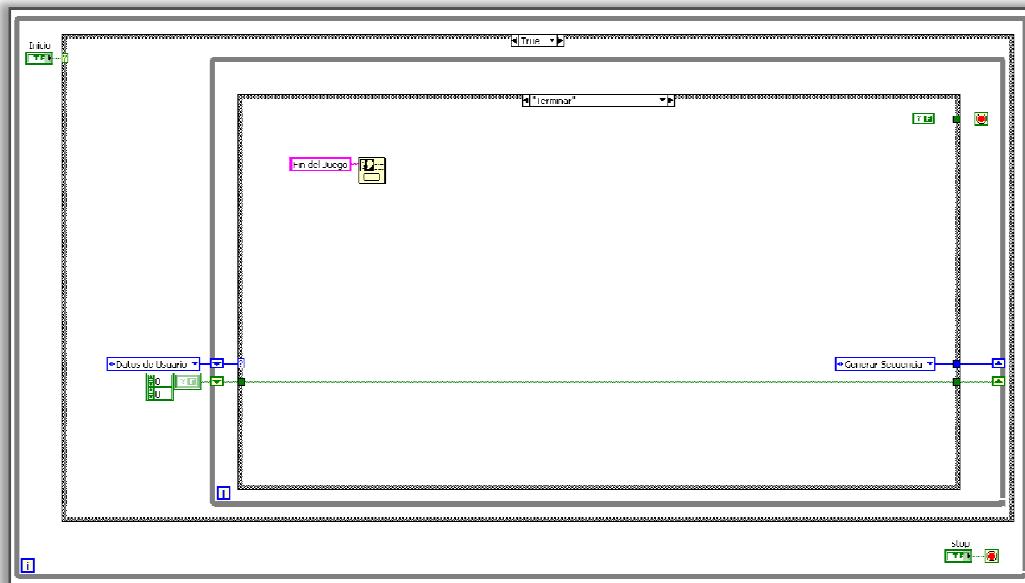


**Figura 2.29** Diagrama de Bloques

20. Finalizar las secuencias, cuando la secuencia sea “Terminar”

- ❑ Seleccionar la secuencia “Terminar” de la estructura Case.
- ❑ Colocar una False Constant dentro de la Estructura Case y conectar su terminal a la del Stop if True del Ciclo While externo.
- ❑ Colocar una copia del **Enum Constant** dentro de la Estructura Case y conectar su terminal a la del Shift Register que se encuentra en el Ciclo While.
- ❑ Cablear de lado a lado las terminales del Shift register de la estructura de arreglos.

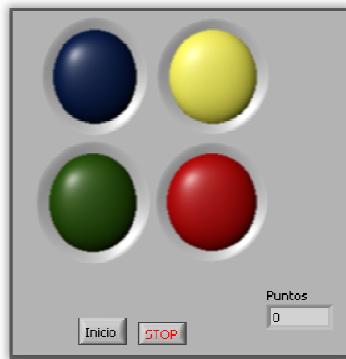
- ❑ Colocar la función **One Button Dialog** dentro de la estructura Case, del evento “Terminar”. Esta se encuentra en la paleta de **Funciones » Programming » Dialog & User Interface » One Button Dialog**.
- ❑ Colocarse sobre la terminal “**message**” , dar Clic Derecho y seleccionar **Create » Constant**, aquí ingresaremos el mensaje que nos indicara el Fin del Juego.



**Figura 2.30** Diagrama de Bloques

## Prueba.

1. Muestre el Panel Frontal.



**Figura 2.31** Panel Frontal Final

2. Ejecute el VI.
3. Cierre el VI.

A partir de este punto el alumno podrá comenzar a complementar el juego del Simón con mejoras.

**Fin del Programa.**

## Notas

---

## Notas

---

# Clase: Electrónica Analógica

ni.com



58

## Temario de la clase

### 3. Amplificadores operacionales

#### 3.1 Amplificadores de Potencia

- 3.1.1 Amplificadores de Potencia Clase A
- 3.1.2 Amplificadores de Potencia Clase B
- 3.1.3 Amplificadores de Potencia Clase AB
- 3.1.4 Otros tipos: Clase C y Clase D

#### 3.2 Circuitos de Compuertas Lógicas.

- 3.2.1 Circuito de compuerta NOT
- 3.2.2 Circuito de compuerta NAND
- 3.2.3 Circuito de compuerta AND
- 3.2.4 Circuito de compuerta OR
- 3.2.5 Circuito de compuerta NOR

#### 3.3 Sumadores

#### 3.4 Circuito Derivador e Integrador

- 3.4.1 Derivador
- 3.4.2 Integrador

#### 3.5 Convertidores Voltaje - Corriente y corriente a Voltaje

#### 3.6 Filtros Activos de Primer Orden

- 3.6.1 Filtros Pasa Bajas
- 3.6.2 Filtros Pasa Altas
- 3.6.3 Filtros Pasa Bandas

#### 3.7 Filtros de Orden Superior.

- 3.7.1 Filtro de Segundo Orden
- 3.7.2 Filtros de Orden Superior (Orden 6)

## 3.1 Amplificadores de Potencia

Amplificador Clase A

Amplificador Clase B

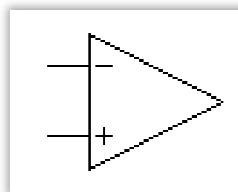
Amplificador Clase AB

[ni.com](http://ni.com)

### Amplificador

Un amplificador recibe una entrada de un elemento y proporciona una versión más grande de la señal a cierto dispositivo de salida.

La función de un amplificador es incrementar la magnitud de un circuito eléctrico, ya sea la intensidad de corriente, la tensión o la potencia de una señal.



**Figura 3.1-1**

Una manera de clasificar los amplificadores es en base a su clase. La clase de los amplificadores se clasifica de acuerdo con la forma de onda de corriente de colector que resulta cuando se aplica una señal de entrada.

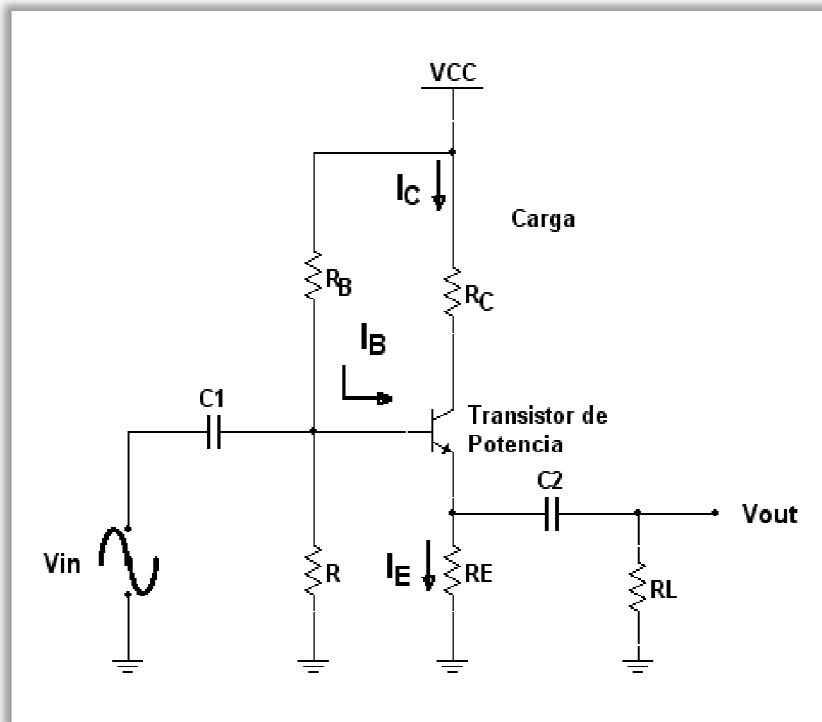
La eficiencia de potencia de un amplificador, definida como la relación de potencia de salida sobre la de entrada, mejora (se vuelve mayor) al ir de la clase A a la clase D.

### 3.1.1 Amplificadores de Potencia Clase A

Debido a su baja resistencia de salida, el seguidor de emisor es la clase más conocida.

Un amplificador de clase A funciona cuando la tensión de polarización y la amplitud máxima de la señal de entrada poseen tales valores que hacen que la corriente de salida circule durante todo el período de la señal de entrada.

A menudo consisten en un transistor de salida conectado al terminal positivo de la fuente de alimentación y un transistor de corriente constante conectado de la salida al negativo de la fuente de alimentación, como se muestra en la figura 1.2. Cuando no hay señal de entrada la corriente de polarización constante fluye directamente del positivo al negativo de la fuente de alimentación al negativo, consumiéndose potencia sin resultar útil.



**Figura 3.1-2** Amplificador de Clase A de gran señal, alimentado en Serie

En este amplificador la polarización de DC establecida por  $V_{CC}$  y fija la corriente de base de polarización en la ecuación (3.1-1)

$$I_B = \frac{V_{CC} - 0.7\text{ V}}{R_B} \quad \text{Ecuación (3.1-1)}$$

Donde la corriente del emisor es (Ecuación 3.1-2)

$$I_E = \frac{\beta + 1}{\beta} I_C \quad I_E = (\beta + 1) I_B \quad (\text{Ecuación 3.1-2})$$

Con el Voltaje Colector – Emisor (Ecuación 3.1-3)

$$V_E = V_{CC} - I_C R_C \quad (\text{Ecuación 3.1-3})$$

### Consideraciones de potencia

La potencia de un amplificador la suministra la fuente de alimentación. Cuando no existe una señal de entrada, el consumo de corriente de DC será la corriente de polarización de colector  $I_{CQ}$ . Entonces, la potencia que consume la fuente es (Ecuación 3.1-4)

$$P_1(DC) = V_{CC} I_{CQ} \quad (\text{Ecuación 3.1-4})$$

Esta ecuación representa la potencia de entrada proporcionada al amplificador de clase A alimentado en serie.

### Potencia de Salida

Mediante señales pico a pico la potencia de ac aplicada a la carga puede expresarse por la (Ecuación 3.1-5)

$$P_0(AC) = \frac{V_E^2 (pp)}{2 R_L} \quad (\text{Ecuación 3.1-5})$$

### Ecuaciones para cálculos de $R_A$ y $R_E$

$$V_B = 10v$$

$$V_B = \frac{R_B}{R_A + R_B} V_{CC}$$

$$R_B = 10k\Omega$$

$$R_A = \frac{R_B V_{CC}}{V_B} - R_B$$

$$R_A = \frac{R_B V_{CC}}{V_B} - R_B$$

$$R_A = R_B \left( \frac{V_{CC}}{V_B} - 1 \right) = 10000 \left( \frac{15}{10} - 1 \right) = 5k\Omega$$

$$V_E = R_E I_E \quad I_E = 1mA$$

$$R_E = \frac{V_E - 0.6}{1 \times 10^{-3}} = \frac{9.4}{1 \times 10^{-3}} = 9.4Khz$$

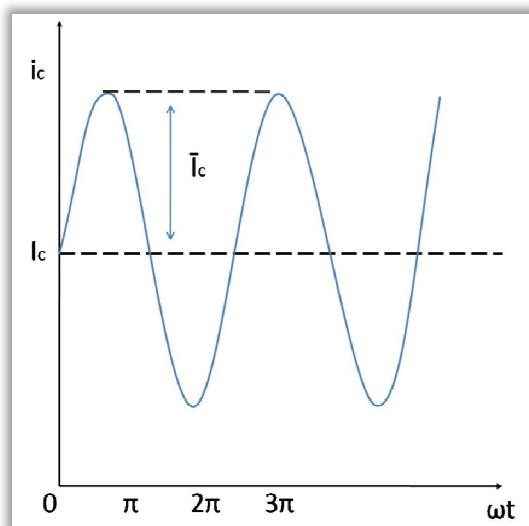
## Eficiencia

La eficiencia de un amplificador representa la cantidad de potencia de AC aplicada (transferida) desde la fuente de DC. Se calcula mediante la (Ecuación 3.1-6).

$$\% \eta = \frac{P_0(AC)}{P_1(DC)} \times 100\% \quad (\text{Ecuación 3.1-6})$$

La eficiencia máxima de un circuito de clase A será de solo 25% con una conexión de carga directa o con alimentación en serie y de 50% con una conexión de transformador a la carga.

En la Figura 1.2 se observa que la forma de onda esta polarizada por una corriente  $I_c$  mayor que la amplitud de la corriente de señal  $i_c$ . El transistor conduce durante todo el ciclo de la señal de entrada, esto es el ángulo de conducción es de  $360^\circ$ .



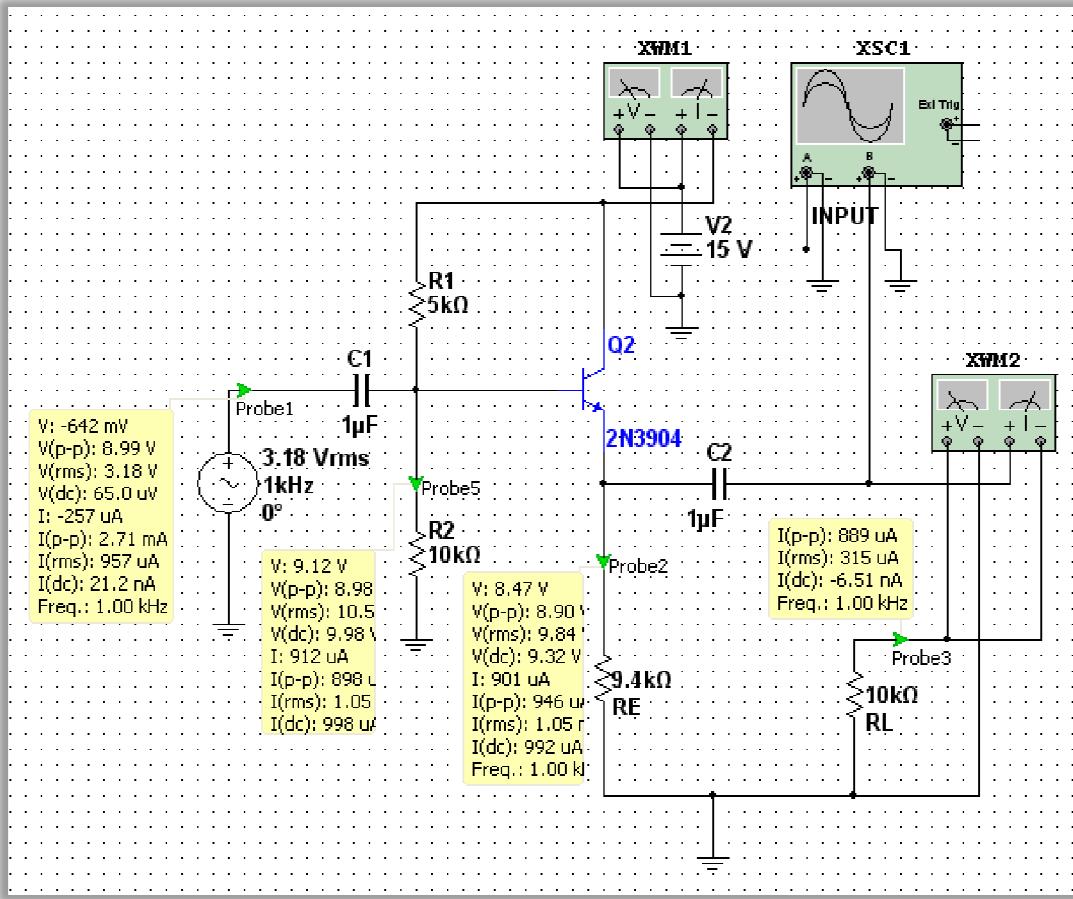
**Figura 3.1-3** Forma de onda de la corriente de colector de los transistores de la Clase A

## Simulación del circuito.

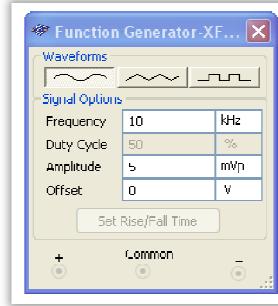
- Incorpore a NI Multisim 10.1 los siguientes elementos:

- 1 Capacitor de 10 uF
- 1 Capacitor de 1 uF
- 1 resistencia de 10 kΩ
- 1 resistencia de 5 kΩ
- 1 resistencia de 100 kΩ
- 1 resistencia de 1 MΩ
- 1 Transistor 2N3904
- 1 Fuente de 12v
- 1 Tierra

2. Conecta el circuito conforme a la Figura 3.1-4 y verifícalo
3. Configura el Generador de Funciones como se muestra en la Figura 3.1-5
4. Ejecute la simulación. Presione el botón RUN que aparece en la Figura 3.1-6



**Figura 3.1-4** Simulación del circuito de la Clase A

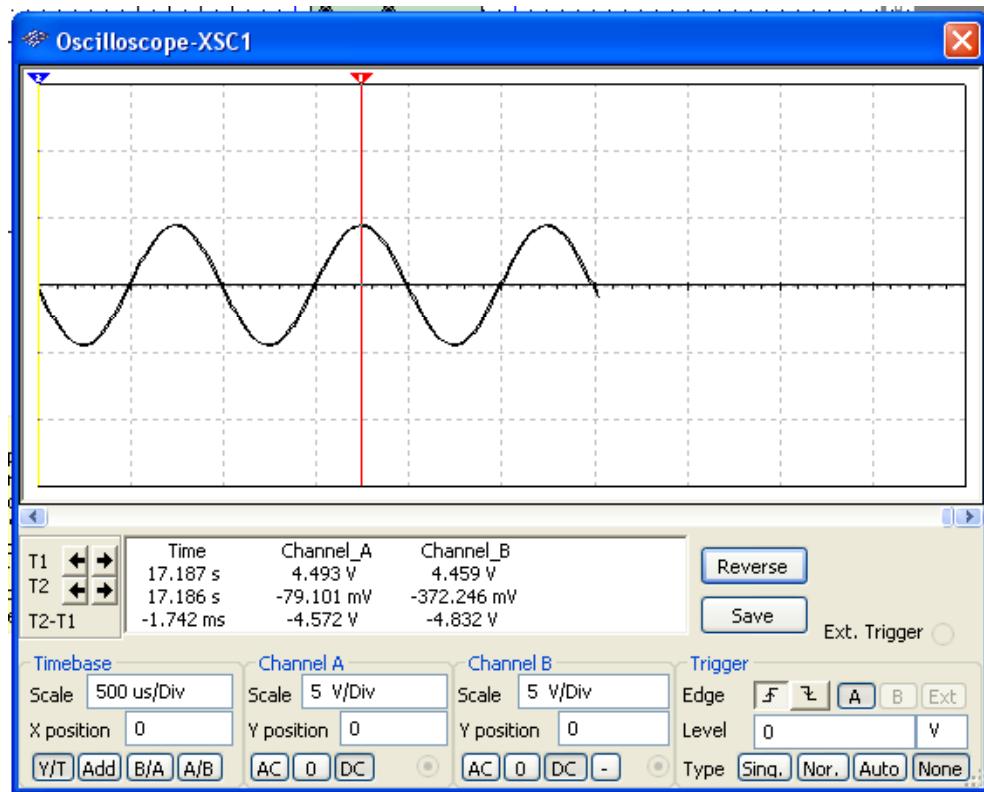


**Figura 3.1-5** Simulación del generador de funciones



**Figura 3.1-6 Inicio de la Simulación.**

5. De doble Clic sobre el Osciloscopio. Presione el botón de Reverse, para cambiar la tonalidad de la imagen a color Blanco de Fondo y poder apreciar mejor las graficas (Figura 3.1-7). En esta figura podemos ver que el canal dos nos proporciona  $4.459 \approx 4.46$  V.



**Figura 3.1-7 Simulación.**

6. A continuación se actualizara la salida del osciloscopio y se agregara el transient de corriente (Esto es para comparar la corriente de carga con la de emisor y ver como el capacitor bloquea la componente de DC y deja pasar la de AC).

- En el menú de **Simulate » Analyses » Transient Analysis** (Ver Figura 3.1-8)

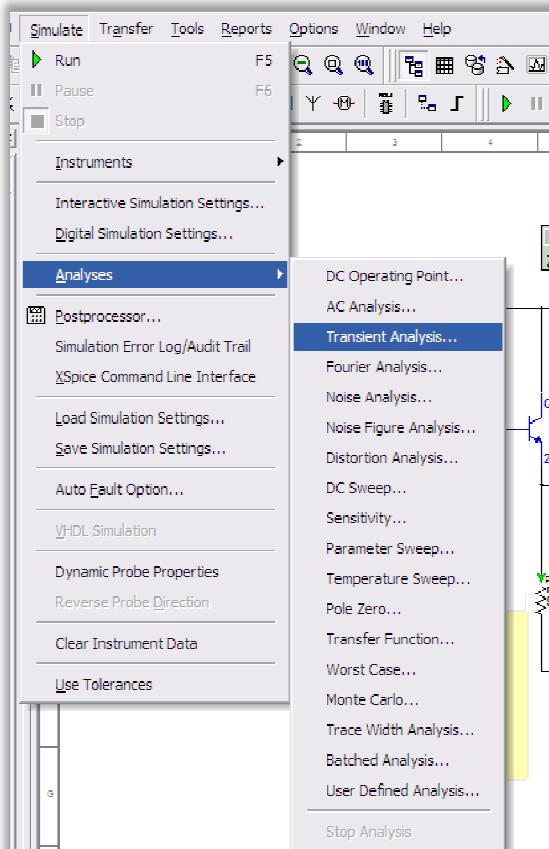
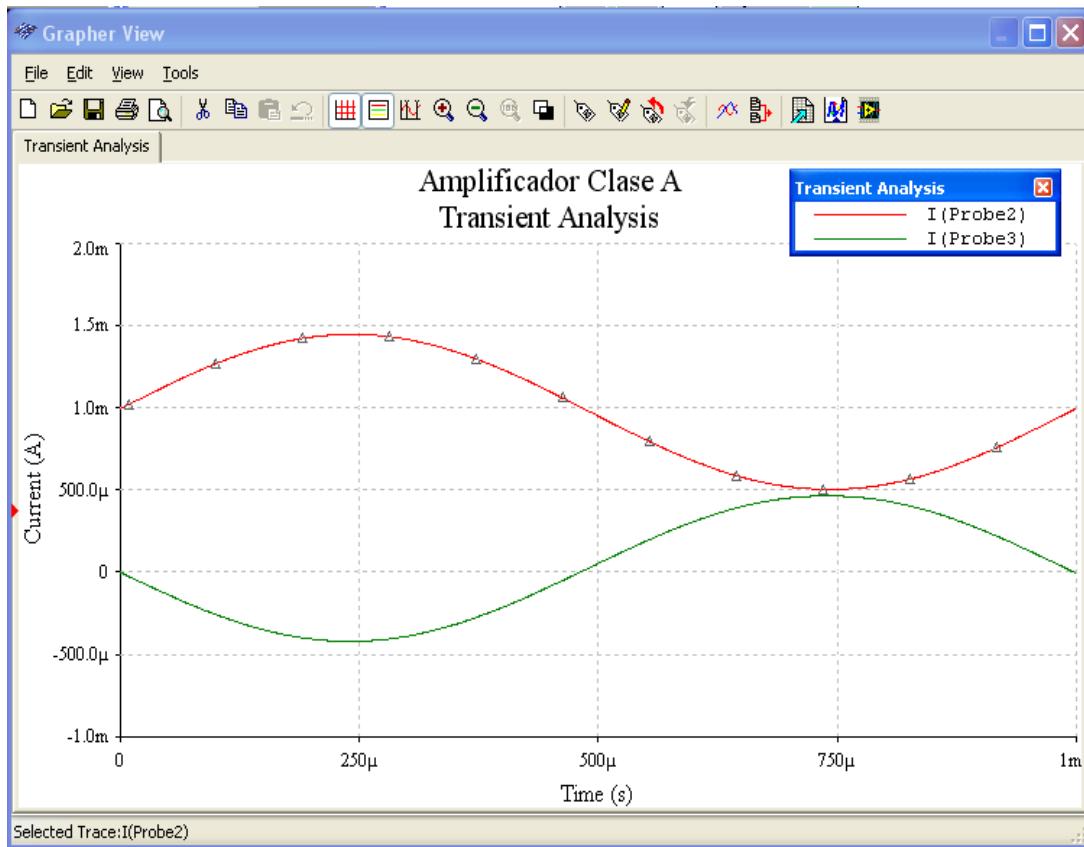


Figura 3.1-8

- Posterior a esto, de Clic en la pestaña de Output.
- En la sección de **Variables in circuit**, asegúrese de que este **seleccionado Static Probes**. Para este ejemplo agregara las variables: Probe 2 y Probe 3, que corresponden a las pruebas de Corriente en el Emisor y la Resistencia de Carga.
- Para ver la simulación, presione el botón de **Simulate**.
- En la Figura 3.1-9 se puede observar la simulación del Transient Analysis.
- De esta manera observamos como el capacitor bloquea la componente de DC y deja pasar la de AC.
- En la pestaña de View, podemos habilitar la vista de herramientas que pueden ser útiles para un mejor análisis.



**Figura 3.1-9** Transient Analysis

### Cálculos de Eficiencia.

- Basándose en las formulas de Potencia (Ecuación 3.1-4) y (Ecuación 3.1-5) realice los cálculos para calcular la eficiencia del amplificador

$$P_1(DC) = V_{cc} I_{cq}$$

$$P_1(DC) = (15V)(2mA) = 30\text{ mW}$$

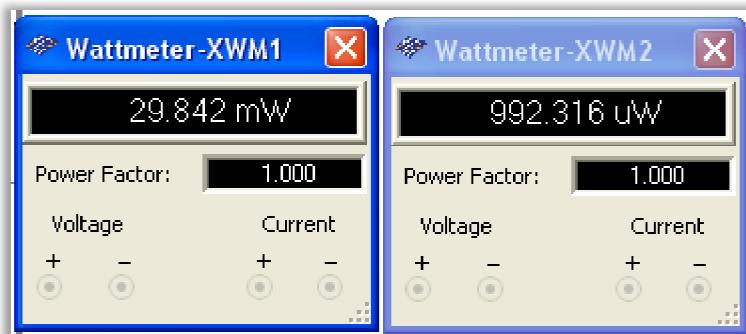
$$P_0(AC) = \frac{V_E^2 (pp)}{2 R_L}$$

$$P_0(AC) = \frac{(4.46)^2}{2(10\text{ k}\Omega)} = .994\text{ mW}$$

$$\% \eta = \frac{P_0(AC)}{P_1(DC)} \times 100\%$$

$$\% \eta = \frac{.994\text{ mW}}{30\text{ mW}} \times 100\% = 3.31\%$$

- Mediante la simulación observamos los valores de la Figura 3.1-10, los que a su vez son muy cercanos a los calculados.



**Figura 3.1-10**

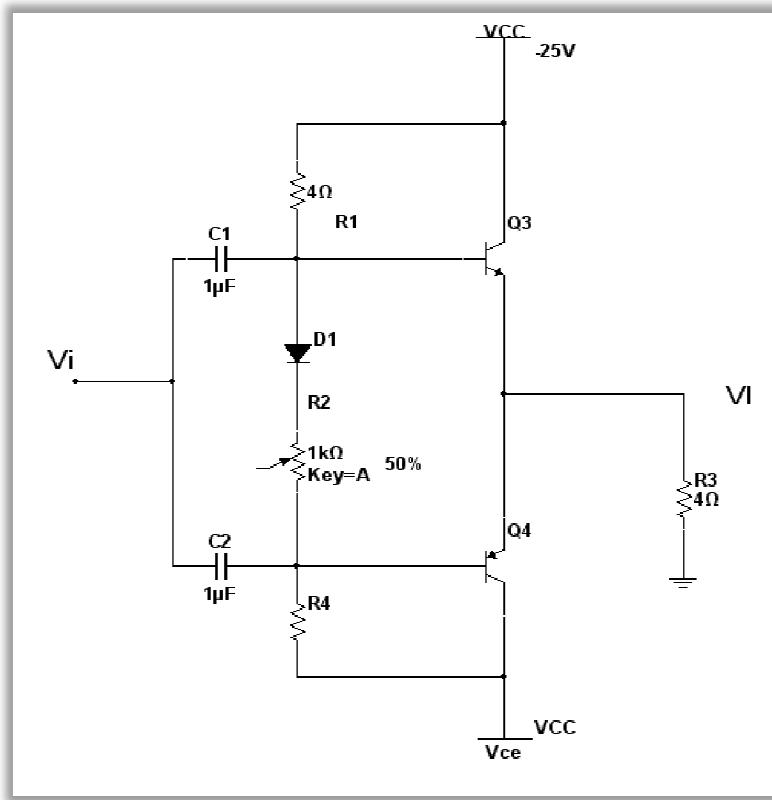
**Reto.**

Configure el Generador de Funciones con diversas frecuencias, y observe los cambios en el Osciloscopio.

### 3.1.2 Amplificadores de Potencia Clase B

Un amplificador de potencia funciona en clase B cuando la tensión de polarización y la amplitud máxima de la señal de entrada poseen valores tales que hacen que la corriente de salida circule durante un semiperíodo de la señal de entrada.

Estos se caracterizan por tener intensidad nula a través de sus transistores cuando no hay señal en la entrada del circuito. Ésta es la que polariza los transistores para que entren en zona de conducción, por lo que el consumo es menor que en la clase A, aunque la calidad es algo menor debido a la forma en que se transmite la onda. Se usa en sistemas telefónicos, transmisores de seguridad portátiles, y sistemas de aviso, aunque no en audio.



**Figura 3.1-11**

#### Potencia de Entrada (DC)

La potencia proporcionada a la carga por un amplificador se toma de la fuente de alimentación que proporciona la potencia de entrada de DC. La cantidad de esta potencia de entrada se calcula mediante

$$P_i (DC) = V_{cc} I_{dc} \quad (\text{Ecuación 3.1-7})$$

Donde  $I_{DC}$  es la corriente promedio o DC que se consume de las fuentes de alimentación. El valor de consumo promedio se puede expresar como

$$I_{DC} = \frac{2}{\pi} I(p) \quad (\text{Ecuación 3.1-8})$$

Donde  $I(p)$  es el valor pico de la forma de onda de la corriente de salida (Ecuación 3.1-8).

Al utilizar la ecuación 3.1-8 en la ecuación de potencia de entrada (Ecuación 3.1-7) obtenemos:

$$P_i(DC) = V_{CC} \left( \frac{2}{\pi} I(p) \right) \quad (\text{Ecuación 3.1-9})$$

### Potencia de Salida (DC)

La potencia aplicada a la carga se puede calcular mediante cualquiera de las distintas ecuaciones. Si se utiliza un medidor rms para medir el voltaje a través de la carga, la potencia de salida se puede calcular como

$$Po(AC) = V_L^2(rms) \quad (\text{Ecuación 3.1-10})$$

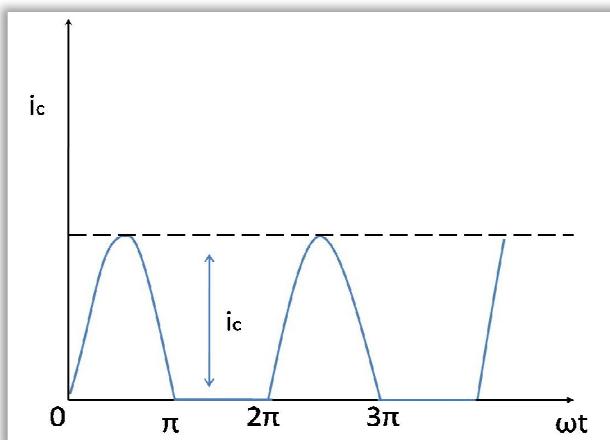
### Eficiencia

La eficiencia de un amplificador también puede calcularse por la ecuación:

$$\% \eta = \frac{P_0(AC)}{P_i(DC)} \times 100\% \quad (\text{Ecuación 3.1-11})$$

Sin potencia de polarización de dc y sin señal de entrada, puede mostrarse que proporciona un eficiencia máxima de 78.5%

En la Figura 3.1-12, se muestra como la forma de onda esta polarizada a cero corrientes de CD. Un transistor de clase B conduce durante solo la mitad del ciclo de la onda senoidal de entrada, resultando en un ángulo de conducción de 180 °.

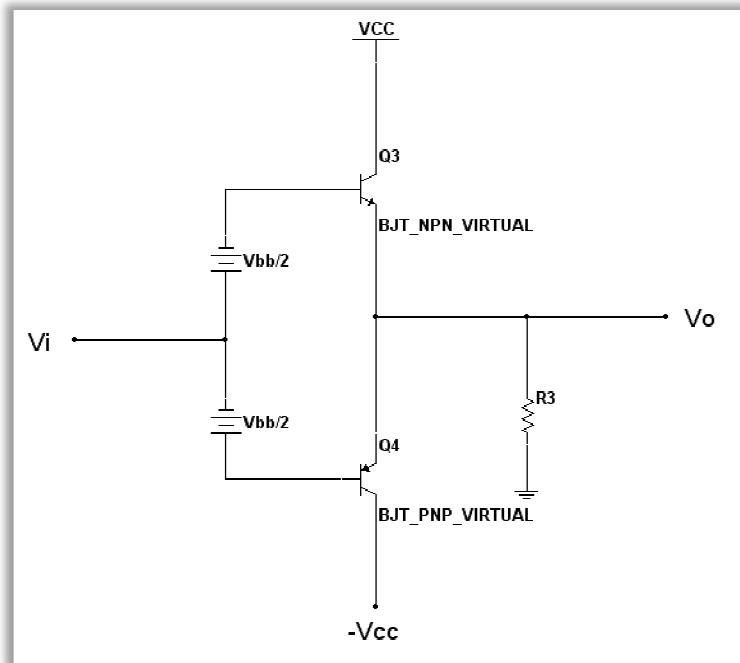


**Figura 3.1-12**

### 3.1.3 Amplificadores de Potencia Clase A B

Son una mezcla de los dos anteriores (A y B). Estos funcionan cuando la tensión de polarización y la amplitud máxima de la señal de entrada poseen valores tales que hacen que la corriente de salida circule durante menos de un período y más de un semiperíodo de la señal de entrada.

Los amplificadores de clase AB reciben una pequeña alimentación constante en su entrada, independiente de la existencia de señal. Es la clase más común en audio, al tener alto rendimiento y calidad. Estos amplificadores reciben su nombre porque con señales grandes se comportan como un clase B, pero con señales pequeñas se comportan como un clase A.



**Figura 3.1-13**

Tienen dos transistores de salida, como los de clase B, pero a diferencia de estos, tienen una pequeña corriente libre fluyendo entre los terminales de la fuente de alimentación, que sin embargo no es tan elevada como en los de clase A. Esta corriente libre corrige casi todas las no linealidades asociadas con la distorsión del filtro.

Cuando  $V_1$  se hace positivo en cierta cantidad, el voltaje en la base de Q3 aumenta en la misma cantidad y la salida se hace positiva a un valor casi igual

$$V_o = V_1 + \frac{V_{bb}}{2} - V_{ben} \quad (\text{Ecuación 3.1-12})$$

El Vo positivo hace que una corriente  $i_m$  circule por y entonces in debe aumentar, esto es

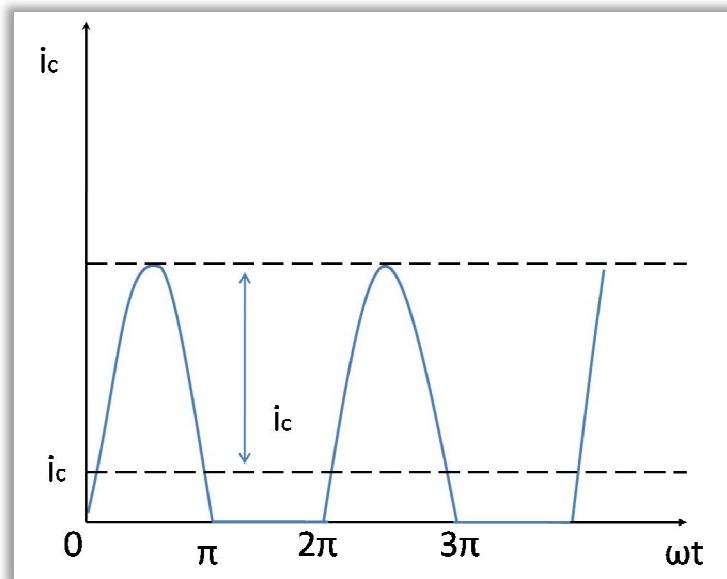
$$i_n = i_p + i_l \quad (\text{Ecuación 3.1-13})$$

Si suponemos que la fuente que alimenta V1 es ideal, entonces la resistencia de salida de la etapa se puede determinar:

$$R_{sal} = r_{eN} // r_{eP} \quad (\text{Ecuación 3.1-14})$$

Donde  $r_{eN}$  y  $r_{eP}$  son las resistencias de emisor de Q3 y Q4 a pequeña señal.

Esta clase implica la polarización del transistor a una corriente de CD diferente de cero, mucho más pequeña que la corriente de de la señal de la onda senoidal. Como resultado de esto, el transistor conduce durante un intervalo ligeramente mayor de medio ciclo. El ángulo de conducción es mayor a  $180^\circ$  pero mucho menor que  $360^\circ$ .



**Figura 3.1-14**

### Eficiencia

Dado que la clase AB se encuentra entre la A y la B en polarización y encuentra entre sus valores de eficiencia entre 25 % (O 50%) y 78.5%

## Simulación del circuito.

1. Incorpora a NI Multisim 10.1 los siguientes elementos:

- 2 Diodos virtuales
- 1 Fuente de CA 1mA
- 2 Transistores Virtuales
- 1 resistencia de  $100\Omega$
- 1 Fuente de 15v
- 8 Tierras
- 1 Generador de funciones
- Un osciloscopio (XSC1)

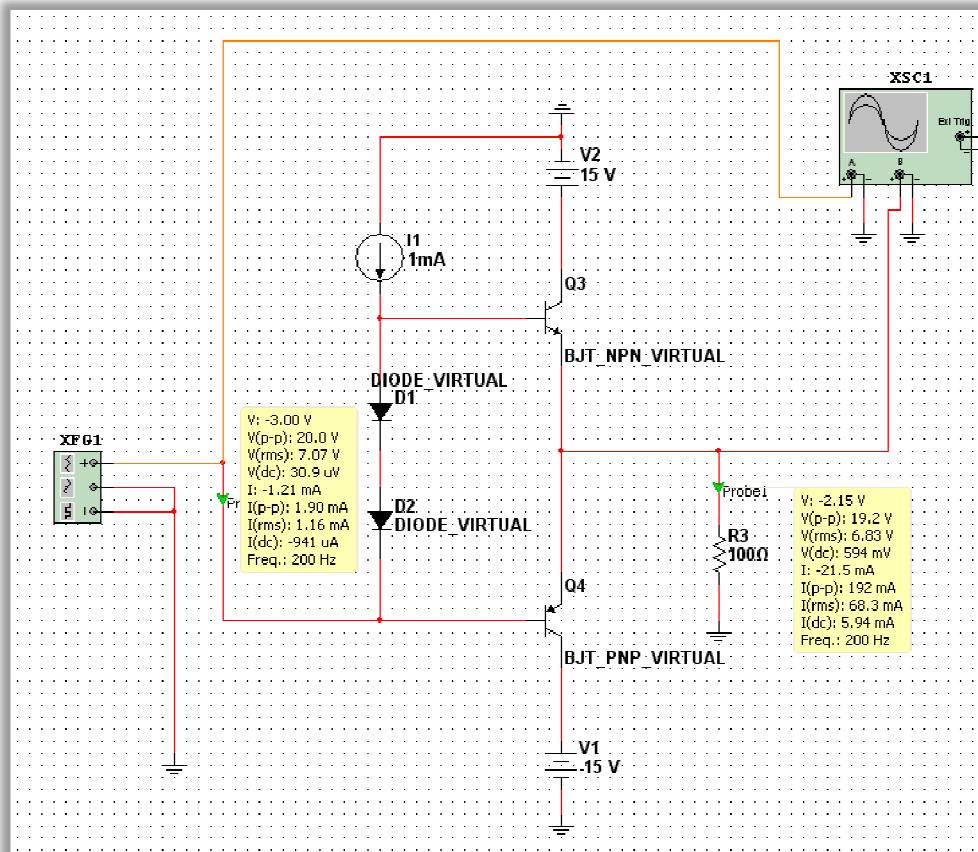
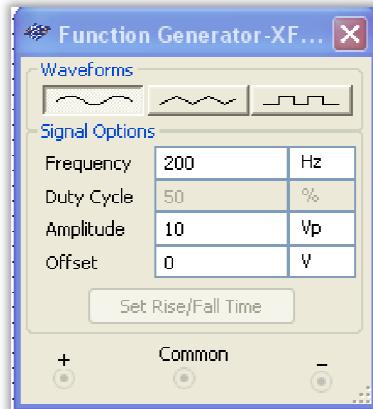


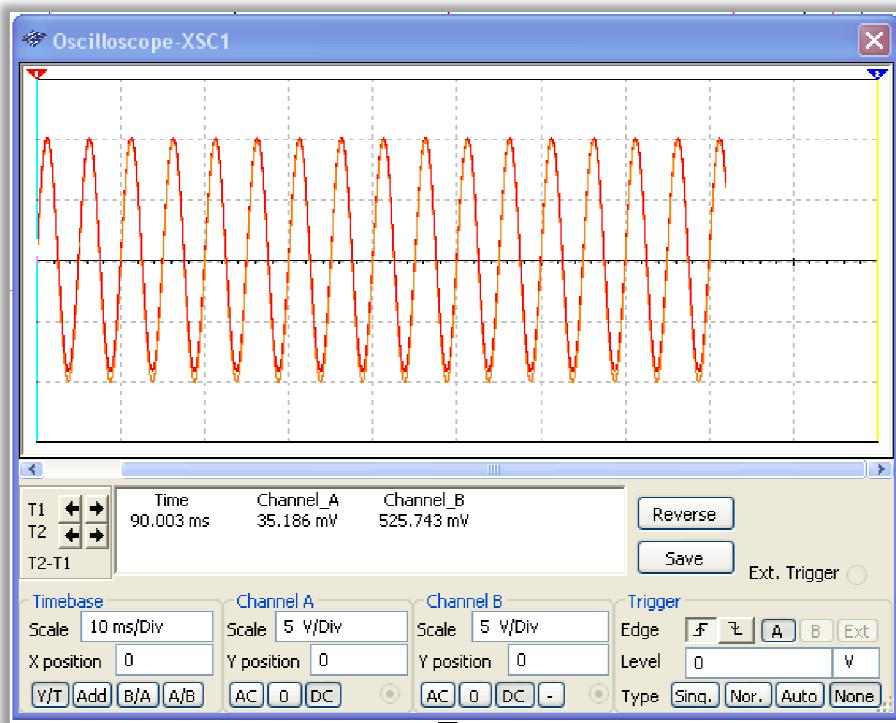
Figura 3.1-15 Circuito de Simulación clase AB

2. Conecta el circuito conforme a la Figura 3.1-13 y verifícalo
3. Configura el Generador de Funciones como se muestra en la Figura 3.1-16



**Figura 3.1-16** Configuración del Generador de Funciones, Clase AB

- Ejecute la simulación. Puede observar los resultados de la simulación se apeguen a la Figura 3.1-17



**Figura 3.1-17**

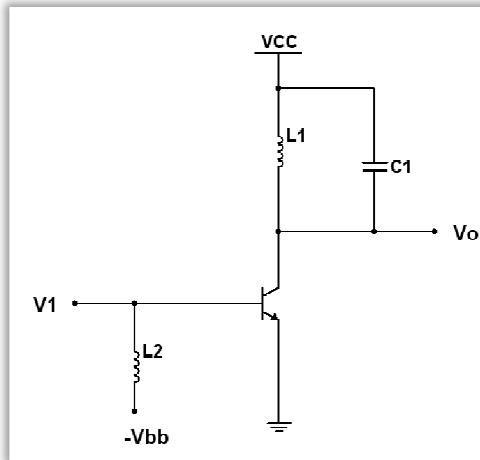
### Reto.

Configure el Generador de Funciones con diversas frecuencias, y observe los cambios en el Osciloscopio.

### 3.1.4 Otros tipos: Clase C y Clase D.

#### Clase C.

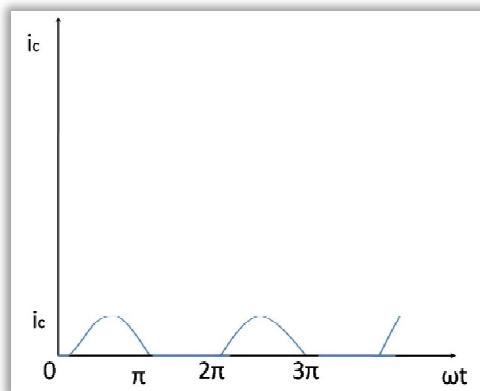
Un amplificador de potencia funciona en clase C cuando la tensión de polarización y la amplitud máxima de la señal de entrada poseen valores tales que hacen que la corriente de salida circule durante menos de un semiperíodo de la señal de entrada.



**Figura 3.1-18** Clase C

Los amplificadores de clase C son similares a los de clase B en que la etapa de salida tiene corriente de polarización cero. Sin embargo, tienen una región de corriente libre cero que es más del 50% del suministro total de voltaje. Las desventajas de los amplificadores de clase B son más evidentes en los amplificadores de clase C. Este tipo de amplificador no se usa en audio.

El transistor conduce durante un intervalo más corto que el de un semi ciclo, esto es el ángulo de conducción es menor que  $180^\circ$ . El resultado es la onda de corriente periódica de pulsos que se muestra en la siguiente figura.



**Figura 3.1-18** Forma de onda de la corriente de colector de los transistores de la Clase C

## Clase D

Los amplificadores de clase D tienen un elevado rendimiento energético, superior en algunos casos al 95%, lo que reduce el tamaño de los disipadores de calor necesarios, y por tanto el tamaño y peso general del circuito.

Los amplificadores de clase D se basan en la conmutación entre dos estados, con lo que los dispositivos de salida siempre se encuentran en zonas de corte o de saturación, casos en los que la potencia disipada en los mismos es prácticamente nula, salvo en los estados de transición, cuya duración debe ser minimizada a fin de maximizar el rendimiento.

Esta señal conmutada puede ser generada de diversas formas, aunque la más común es la modulación por ancho de pulso. Ésta debe ser filtrada posteriormente para recuperar la información de la señal, para lo que la frecuencia de conmutación debe ser superior al ancho de banda de la señal al menos 10 veces.

En la Tabla a continuación se muestra una comparación entre las clases A, AB, B, C y D.

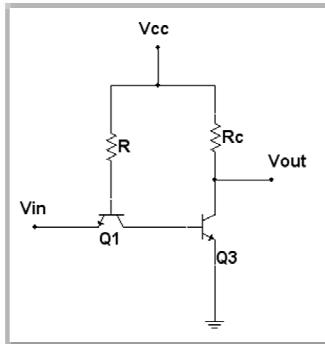
	Clase				
	A	AB	B	C	D
<b>Ciclo de operación</b>	360 °	180 ° a 360 °	180 °	Menor a 180 °	Operación de pulso
<b>Eficiencia de potencia</b>	25 % a 50%	Entre 25 % (50%) y 78.5%	78.5%		Por lo general cerca de 90%

\*La Clase C por lo general no se utiliza para proporcionar grandes cantidades de potencia, por lo que la eficiencia no se presenta aquí.

## 3.2 Circuitos de Compuertas Lógicas

Circuitos Lógicos de Transistor-Transistor, esto es una tecnología de construcción de circuitos electrónicos digitales.

En esta figura se muestra la forma conceptual de una compuerta TTL



ni.com

76

### Introducción

Los circuitos digitales emplean componentes encapsulados, los cuales pueden albergar compuertas lógicas o circuitos lógicos más complejos.

Estos componentes están estandarizados, para que haya una compatibilidad entre fabricantes, de forma que las características más importantes sean comunes.

De forma global los componentes lógicos se engloban dentro de una de las siguientes dos familias:

- TTL: diseñada para una alta velocidad.
- CMOS: diseñada para un bajo consumo.

### TTL (Lógica Transistor a Transistor)

#### Definición de TTL

Siglas en Ingles *Transistor-Transistor Logic*. Es una tecnología de construcción de circuitos electrónicos digitales, en los que los elementos de entrada de la red lógica son transistores, así como los elementos de salida del dispositivo.

## Características de los componentes TTL

- Su tensión de alimentación se encuentra entre los 4.75v y los 5.25V.
- Los niveles lógicos vienen definidos por el rango de tensión comprendida entre 0.2V y 0.8V para el estado L (bajo) y los 2.4V y Vcc para el estado H (alto).
- La velocidad de transmisión entre los estados lógicos es su mejor base, si bien esta característica le hace aumentar su consumo siendo su mayor enemigo. Motivo por el cual han aparecido diferentes versiones de TTL como FAST, LS, S, etcétera y últimamente los CMOS: HC, HCT y HCTLS. En algunos casos puede alcanzar poco más de los 250 MHz.
- Las señales de salida TTL se degradan rápidamente si no se transmiten a través de circuitos adicionales de transmisión (no pueden viajar más de 2 m por cable sin graves pérdidas).

## Construcción de la Tecnología TTL

La tecnología TTL se caracteriza por tener tres etapas:

- **Etapa de entrada por emisor.** Se utiliza un transistor multiemisor en lugar de la matriz de diodos de DTL.
- **Separador de fase.** Es un transistor conectado en emisor común que produce en su colector y emisor señales en contrafase.
- **Driver.** Está formada por varios transistores, separados en dos grupos. El primero va conectado al emisor del separador de fase y drenan la corriente para producir el nivel bajo a la salida. El segundo grupo va conectado al colector del divisor de fase y produce el nivel alto.

Esta configuración general varía ligeramente entre dispositivos de cada familia, principalmente la etapa de salida, que depende de si son búferes o no y si son de colector abierto, tres estados (ThreeState), etc.

## 3.2.1 Circuito de compuerta NOT

Simbología y simulación de la compuerta NOT en NI Multisim 10.1.

ni.com

NATIONAL INSTRUMENTS

### Funcionamiento

Este circuito siempre tiene una sola entrada y su nivel lógico de salida siempre es contrario al nivel lógico de esta entrada.

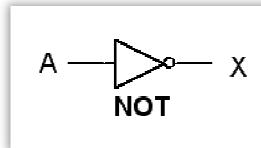


Figura 3.2-1

### Simbología

El símbolo de la compuerta NOT se muestra en la Figura 3.2-1, el cual se conoce comúnmente como INVERSOR (Inverter en inglés). Este circuito siempre tiene una sola entrada y su nivel lógico de salida siempre es contrario al nivel lógico de esta entrada; es decir:

$$X = A' = /A \quad (\text{Ecuación 3.2-1})$$

### Tabla de verdad

A continuación se muestra la tabla funcional para la compuerta NOT:

A	X
0	1
1	0

## Simulación del Circuito

1. Incorpore a NI Multisim 10.1 los siguientes elementos:

- 1 Switch para manipular la entrada A
- Un Diodo de propósito general 1N4001
- 4 Resistencias de 1k
- 3 Transistores 2N2222.
- Una fuente de Voltaje DC a 12 V
- Una alimentación de 5v
- 2 Tierras

2. Conecte el circuito conforme a la Figura 3.2-2

3. Inicie la simulación y compare sus resultados con los que se muestran en las Figuras 3.2-2 y 3.2-3.
4. En las Figuras 3.2-2 y 3.2-3 puede observar la simulación que se llevo a cabo de todo este circuito, y donde se utilizo un Multímetro para medir el voltaje a la salida. Vemos que en el primer caso, representado por la Figura 3.2-2, cuando tenemos un 0 lógico, a la salida obtendremos el 1 lógico, y en la Figura 3.2-3 observamos lo contrario.

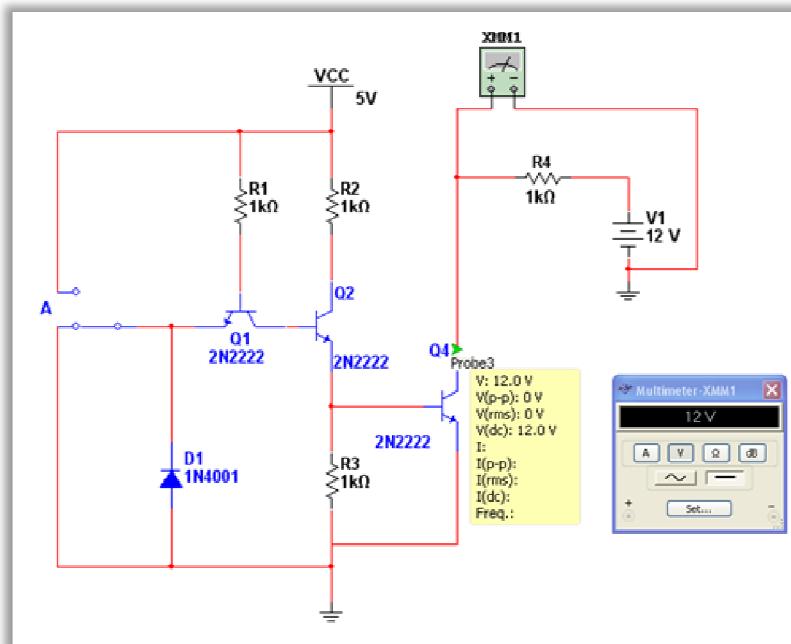


Figura 3.2-2

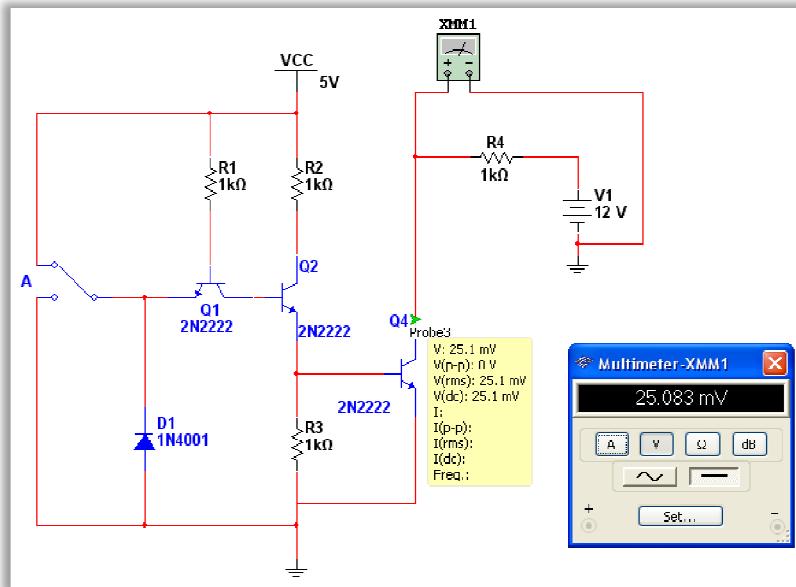


Figura 3.2-3

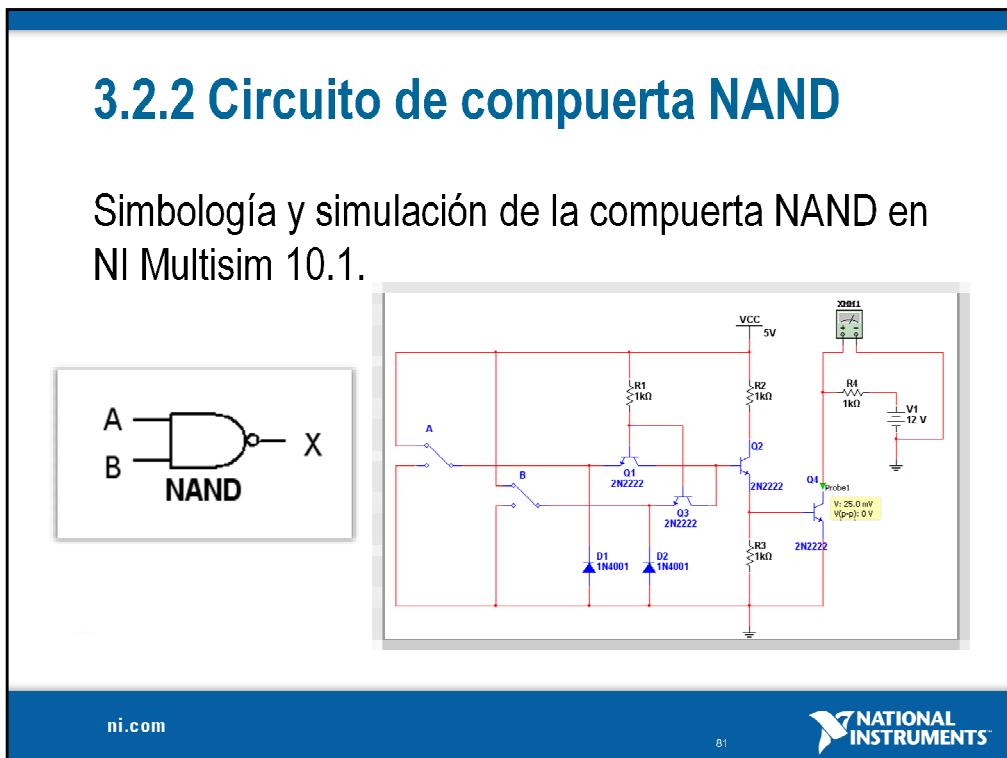
### Ejercicio (Opcional)

Llevar a la práctica la simulación antes mostrada mediante la prueba de los distintos valores de la tabla de verdad de esta compuerta lógica.

A	X
0	1
1	0

## 3.2.2 Circuito de compuerta NAND

Simbología y simulación de la compuerta NAND en NI Multisim 10.1.



### Funcionamiento

La compuerta NAND opera igual de la AND seguida de un INVERSOR, de manera que los circuitos de la figura son equivalentes.

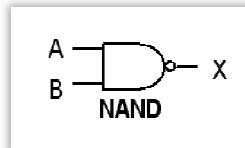


Figura 3.2-4

### Simbología

El símbolo correspondiente a una compuerta NAND de 2 entradas se muestra en la Figura 3.2-4. Es el mismo que el de la compuerta Y, excepto por el pequeño círculo en su salida. Una vez más, este círculo denota la operación de inversión. De este modo la compuerta NAND opera igual que la Y seguida de un inversor; es decir:

$$X = (A \cdot B)' = A' + B' \quad (\text{Ecuación 3.2-2})$$

### Tabla de verdad

La tabla de verdad de la compuerta NAND es:

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

## Simulación del Circuito

1. Incorpore a NI Multisim 10.1 los siguientes elementos:

- 2 Switch para manipular las entradas A y B
- 2 Diodos de propósito general 1N4001
- 4 Resistencias de 1k
- 4 Transistores 2N2222.
- Una fuente de Voltaje DC a 12 V
- Una alimentación de 5v
- 2 Tierras

2. Conecte el circuito conforme a la Figura 3.2-5

3. Inicie la simulación y compare sus resultados con los que se muestran en las Figuras 3.2-5, 3.2-6 y 3.2-7.

En las Figuras 3.2-5, 3.2-6 y 3.2-7 puede observar la simulación que se llevo a cabo de todo este circuito, y donde se utilizo un Multímetro para medir el voltaje a la salida. Vemos que en el primer caso, representado por la Figura 3.2-5, cuando tenemos en ambos switch un 1 lógico, a la salida obtendremos el 0 lógico, y en la Figura 3.2-7 otro ejemplo de respuesta.

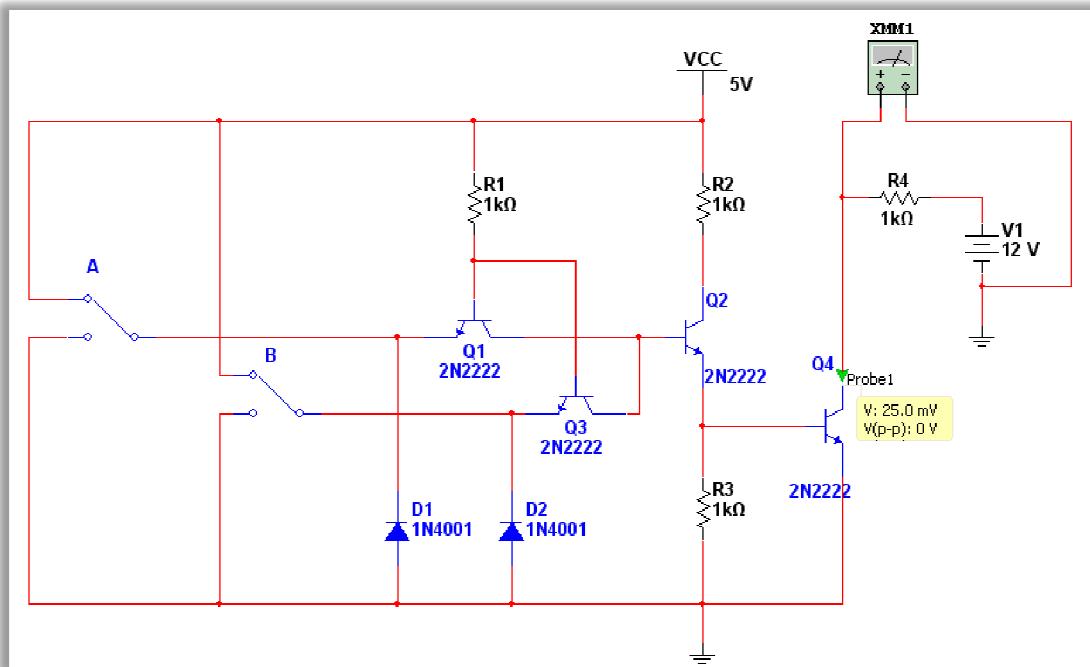


Figura 3.2-5



Figura 3.2-6

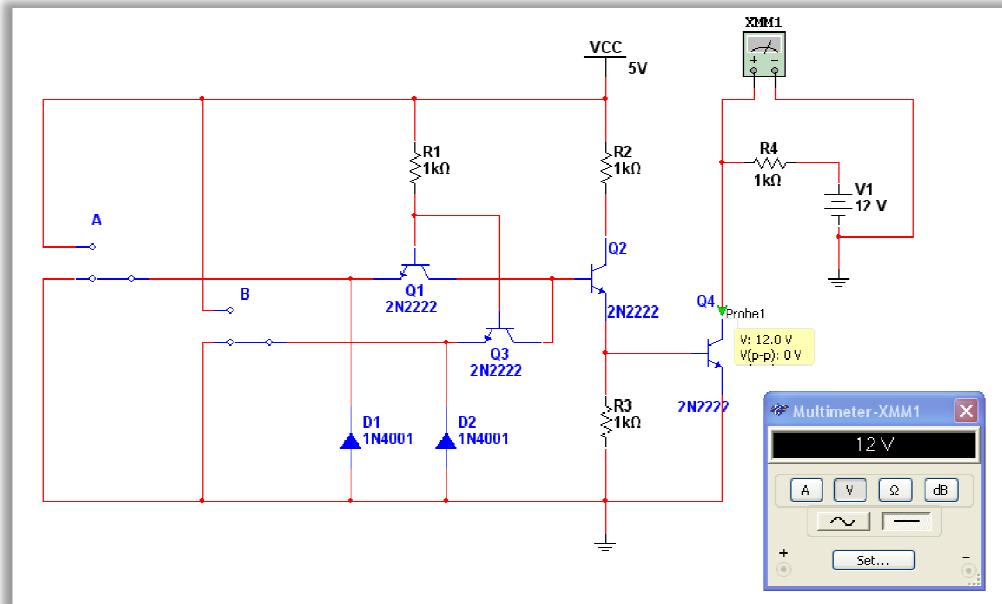


Figura 3.2-7

### Ejercicio (Opcional)

Llevar a la práctica la simulación antes mostrada mediante la prueba de los distintos valores de la tabla de verdad de esta compuerta lógica.

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

### 3.2.3 Circuito de compuerta AND

Simbología y simulación de la compuerta AND en NI Multisim 10.1.

The figure contains two parts. On the left is the standard logic symbol for a 2-input AND gate, with inputs A and B and output X. On the right is a detailed circuit diagram in NI Multisim. The circuit uses two NPN transistors (2N2222) as switches controlled by the inputs A and B. Diodes (1N4001) are used to ensure proper biasing. The circuit is powered by a 5V source (VCC) and a 10V source (V3). The output is measured across a 1kΩ resistor (R4). A Multimeter window shows the output voltage. The circuit is labeled with component values: R1-R6 (1kΩ), Q1-Q4 (2N2222), D1-D2 (1N4001), and a 10V power source V3.

ni.com

NATIONAL INSTRUMENTS

## Funcionamiento

La compuerta AND es un circuito que opera en forma tal que su salida es ALTA sólo cuando todas sus entradas son ALTAS. En todos los otros casos la salida de la compuerta AND es BAJA.

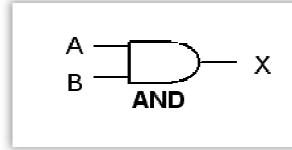


Figura 3.2-8

## Simbología

La simbología de la compuerta AND se muestra en la Figura 3.2-8, en este caso una compuerta AND de 2 entradas. La salida de la compuerta AND es igual al producto Y de las entradas lógicas; es decir:

$$X = A \cdot B \quad (\text{Ecuación 3.2-3})$$

## Tabla de verdad

La compuerta Y es un circuito que opera en forma tal que su salida es ALTA, sólo cuando todas sus entradas son ALTAS. En todos los otros casos su salida es BAJA. La tabla de verdad para la compuerta Y se muestra a continuación:

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

## Simulación del Circuito

1. Incorpore a NI Multisim 10.1 los siguientes elementos:

- 2 *Smith* para manipular las entradas A y B
- 2 Diodos de propósito general 1N4001
- 5 Resistencias de 1k
- 5 Transistores 2N2222.
- Una fuente de Voltaje DC a 12 V
- Una alimentación de 5v
- 2 Tierras

2. Conecte el circuito conforme a la Figura 3.2-9

3. Inicie la simulación y compare sus resultados con los que se muestran en las Figuras 3.2-9, 3.2-10 y 3.2-11.
4. En las Figuras 3.2-9, 3.2-10 y 3.2-11 puede observar la simulación que se llevo a cabo de todo este circuito, y donde se utilizo un Multímetro para medir el voltaje a la salida. Vemos que en el primer caso, representado por la Figura 3.2-9, cuando tenemos en ambos switch un 1 lógico, a la salida obtendremos el 1 lógico, y en la Figura 3.2-11 otro ejemplo de respuesta.

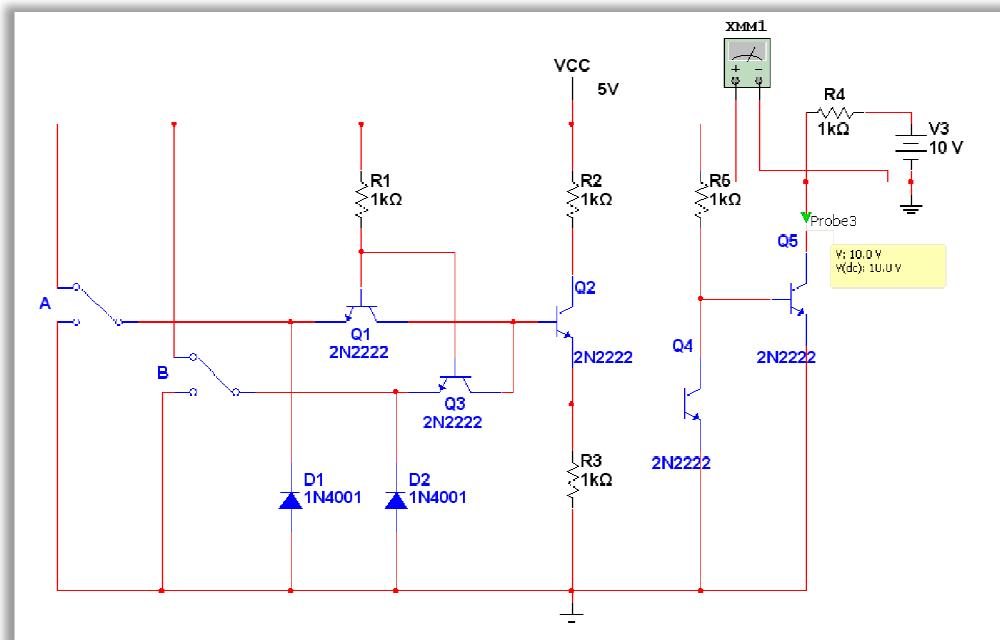


Figura 3.2-9

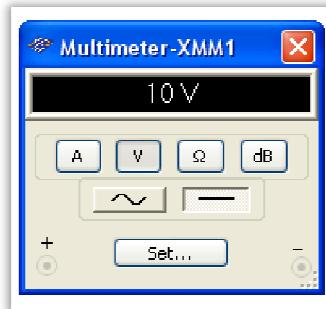


Figura 3.2-10

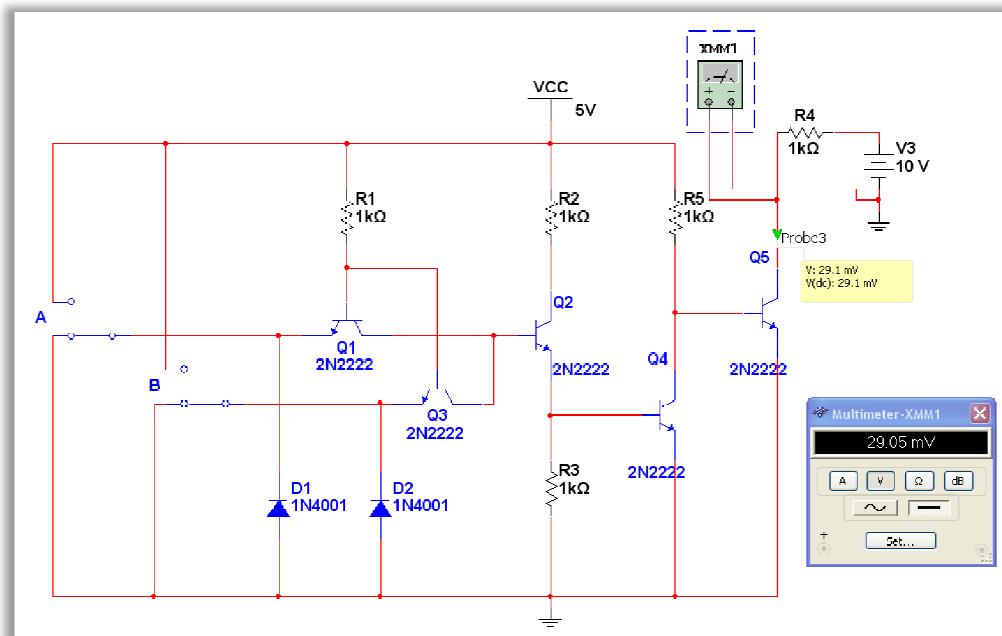


Figura 3.2-11

### Ejercicio (Opcional)

Llevar a la práctica la simulación antes mostrada mediante la prueba de los distintos valores de la tabla de verdad de esta compuerta lógica.

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

## 3.2.4 Circuito de compuerta OR

Simbología y simulación de la compuerta OR en NI Multisim 10.1.

The diagram illustrates the logic symbol for an OR gate and its corresponding circuit simulation in NI Multisim 10.1. The simulation shows a detailed circuit with transistors 2N2222, diodes IN1001, resistors R1-R6, and capacitors C1-C3. A multimeter window displays a reading of 15.669 mV.

### Funcionamiento

Este circuito siempre tiene una sola entrada y su nivel lógico de salida siempre es contrario al nivel lógico de esta entrada.

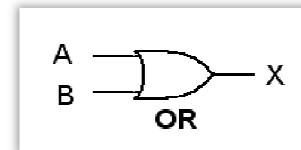


Figura 3.2-12

### Simbología

La compuerta OR es un circuito que tiene 2 o más entradas y cuya salida es igual a la suma OR de las entradas. Las entradas A y B son niveles de voltaje lógicos y la salida (o resultado) X es un nivel de voltaje lógico, cuyo valor es el resultado de la adición OR de A y B; esto es:

$$X = A + B \quad (\text{Ecuación 3.2-4})$$

### Tablas de verdad

La compuerta OR opera de tal forma que su salida es ALTA si las entradas A, B o ambas están en un nivel lógico 1. La salida de la compuerta OR será BAJA, si las entradas están en un nivel lógico 0. A continuación se presenta la tabla de verdad de la compuerta OR:

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

## Simulación del Circuito

Incorpore a NI Multisim 10.1 los siguientes elementos:

2 switch para manipular las entradas A y B

2 Diodos de propósito general 1N4001

5 Resistencias de 1k

5 Transistores 2N2222.

Una fuente de Voltaje DC a 12 V

Una alimentación de 5v

2 Tierras

Conecte el circuito conforme a la Figura 3.2-13

Inicie la simulación y compare sus resultados con los que se muestran en las Figuras 3.2-13, 3.2-14 y 3.2-15.

En las Figuras 3.2-13, 3.2-14 y 3.2-15 puede observar la simulación que se llevo a cabo de todo este circuito, y donde se utilizo un Multímetro para medir el voltaje a la salida. Vemos que en el primer caso, representado por la Figura 3.2-13, cuando tenemos en ambos switch un 0 lógico, a la salida obtendremos el 1 lógico, y en la Figura 3.2-15 otro ejemplo de respuesta..

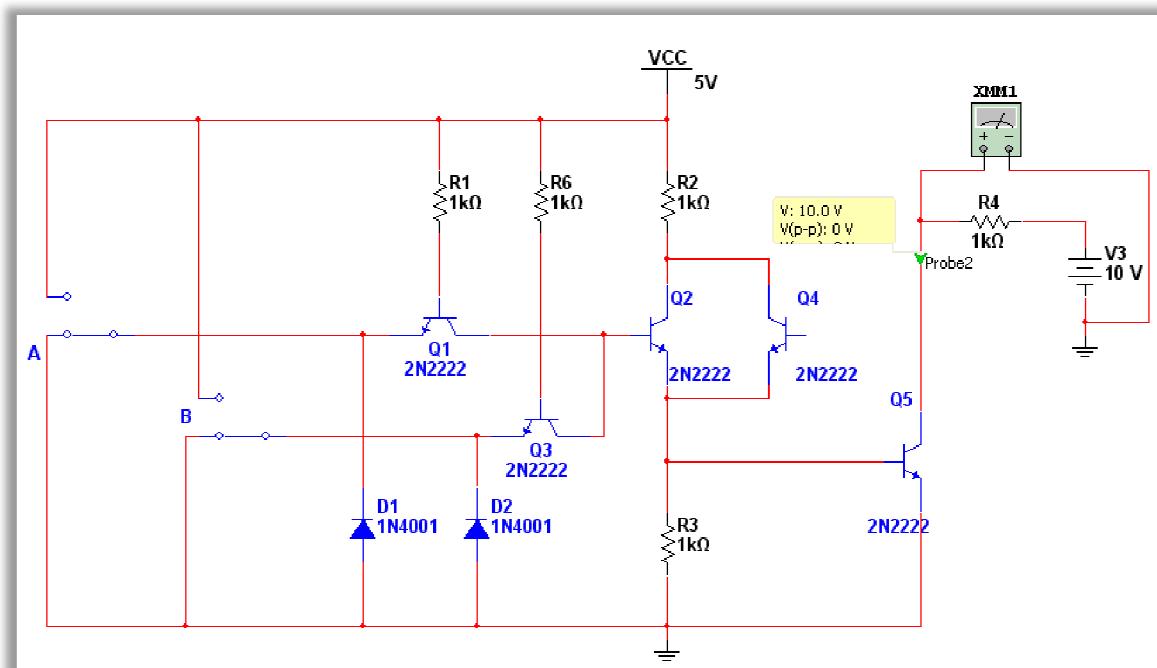


Figura 3.2-13

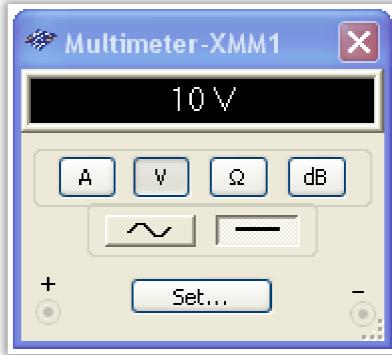


Figura 3.2-14

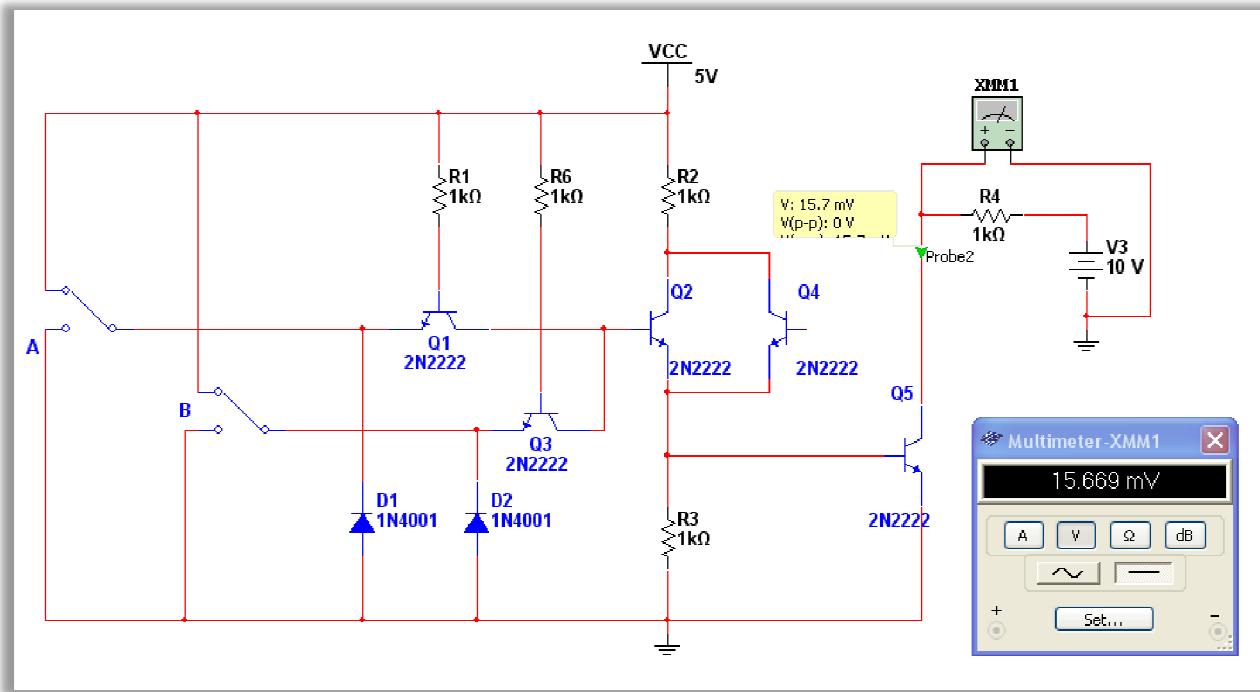


Figura 3.2-15

### Ejercicio (Opcional)

Llevar a la práctica la simulación antes mostrada mediante la prueba de los distintos valores de la tabla de verdad de esta compuerta lógica.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

## 3.2.5 Circuito de compuerta NOR

Simbología y simulación de la compuerta NOR en NI Multisim 10.1.

The figure displays the logic symbol for a NOR gate (A and B inputs, X output) and its corresponding circuit simulation in NI Multisim 10.1. The simulation shows a complex circuit using 2N2222 transistors and resistors, connected to a 5V power source and a 10V probe. The circuit is designed to implement the NOR function using a combination of OR and NOT operations.

### Funcionamiento

La compuerta NOR opera como una compuerta OR seguida de un INVERSOR, de manera que los circuitos de la figura son equivalentes.

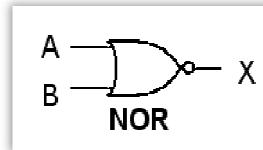


Figura 3.2-16

### Simbología

El símbolo correspondiente a una compuerta NOR de 2 entradas se muestra en la Figura 3.2-16. Es el mismo que el de la compuerta O, excepto por un pequeño círculo en su salida. Una vez más, este círculo denota la operación de inversión. De este modo la compuerta NOR opera igual que O seguida de un inversor; es decir:

$$X = (A + B)' = A' B' \quad (\text{Ecuación 3.2-5})$$

### Tabla de verdad

La tabla de verdad de la compuerta NOR es:

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

## Simulación del Circuito

Incorpore a NI Multisim 10.1 los siguientes elementos:

2 *Switch* para manipular las entradas A y B

2 Diodos de propósito general 1N4001

6 Resistencias de 1k

6 Transistores 2N2222.

Una fuente de Voltaje DC a 12 V

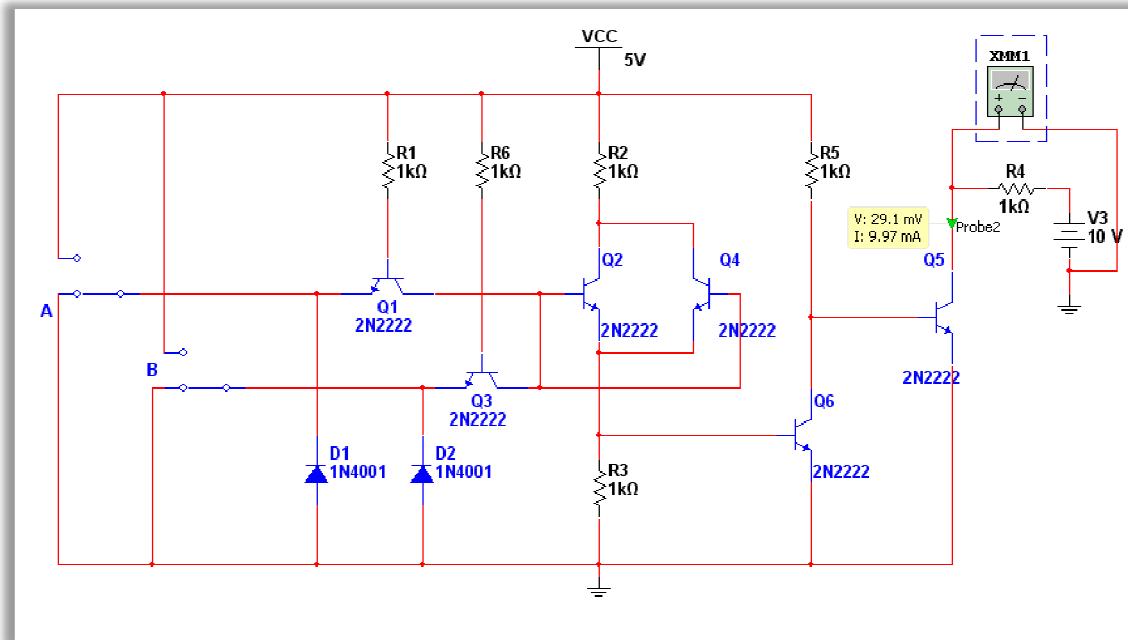
Una alimentación de 5v

2 Tierras

Conecte el circuito conforme a la Figura 3.2-17

Inicie la simulación y compare sus resultados con los que se muestran en las Figuras 3.2-17, 3.2-18 y 3.2-19.

En las Figuras 3.2-17, 3.2-18 y 3.2-19 puede observar la simulación que se llevo a cabo de todo este circuito, y donde se utilizo un Multímetro para medir el voltaje a la salida. Vemos que en el primer caso, representado por la Figura 3.2-17, cuando tenemos en ambos switch un 0 lógico, a la salida obtendremos el 0 lógico, y en la Figura 3.2-19 otro ejemplo de respuesta.



**Figura 3.2-17**

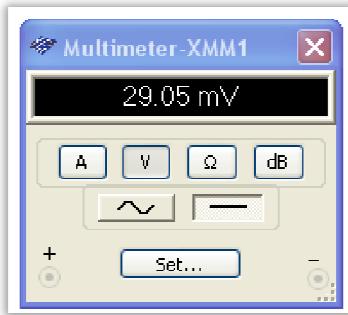


Figura 3.2-18

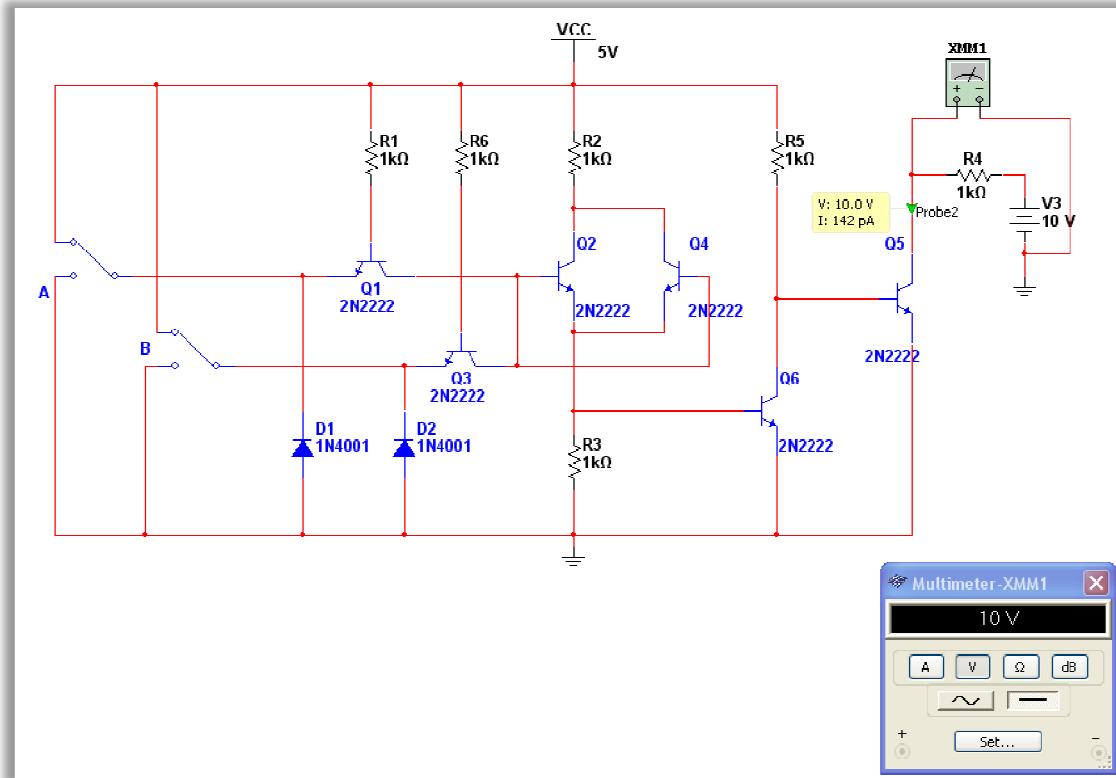


Figura 3.2-19

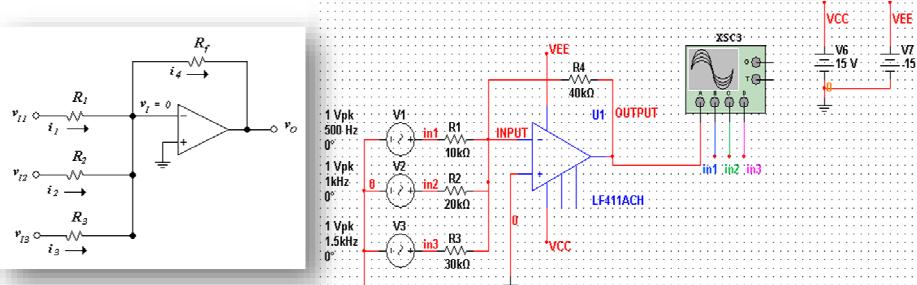
### Ejercicio (Opcional)

Llevar a la práctica la simulación antes mostrada mediante la prueba de los distintos valores de la tabla de verdad de esta compuerta lógica.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

### 3.3 Sumadores

Se analizara el amplificador sumador utilizando el principio de superposición.



ni.com

93

### Introducción

Los amplificadores operacionales tienen múltiples funcionalidades además de simplemente amplificar un cierto voltaje de entrada. Entre estas funcionalidades destacan realizar operaciones matemáticas (sumar, derivar, integrar), filtros, convertidores voltaje-corriente y corriente-voltaje, etc. Este tipo de funcionalidades las estaremos revisando en esta práctica y prácticas subsecuentes. Comenzaremos las aplicaciones del amplificador operacional con el tema de los sumadores.

## Amplificador Sumador

Analizaremos el circuito mostrado en la Figura 3.3-1. Utilizando el principio de superposición determinaremos la salida de voltaje provocada de manera individual por cada una de las entradas.

Algebraicamente sumaremos estos términos para determinar la salida total.

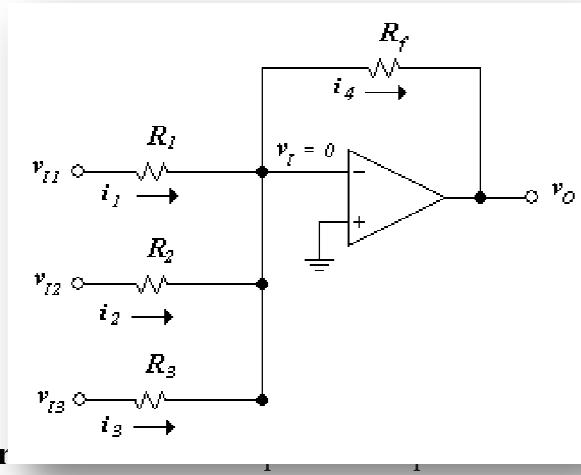


Figura 3.3-1 Circuito de un amplificador sumador.

Si seleccionamos a  $v_{I2} = v_{I3} = 0$ , la corriente  $i_1$  es:

$$i_1 = \frac{v_{I1}}{R_1} \quad (\text{Ecuación 3.3-1})$$

Debido a que  $v_{I2} = v_{I3} = 0$  y la terminal inversora es una tierra virtual, las corrientes  $i_2$  e  $i_3$  deben ser cero. La corriente  $i_1$  no fluye sobre  $R_2$  o  $R_3$ , pero la corriente debe fluir completamente sobre el resistor de retroalimentación  $R_F$ . El voltaje de salida provocado individualmente por  $v_{I1}$  es:

$$v_0 = -i_1 R_F = -\left(\frac{R_F}{R_1}\right) v_{I1} \quad (\text{Ecuación 3.3-2})$$

De manera similar, los voltajes de salida provocados por  $v_{I2}$  y  $v_{I3}$  actuando de manera individual son:

$$v_0(v_{I2}) = -i_2 R_F = -\left(\frac{R_F}{R_2}\right) v_{I2} \quad (\text{Ecuación 3.3-3})$$

$$v_0(v_{I3}) = -i_3 R_F = -\left(\frac{R_F}{R_3}\right) v_{I3} \quad (\text{Ecuación 3.3-4})$$

El voltaje de salida total es la suma algebraica de los voltajes de salida individuales.

$$v_0 = v_0(v_{I1}) + v_0(v_{I2}) + v_0(v_{I3}) \quad (\text{Ecuación 3.3-5})$$

El cual se convierte en:

$$v_0 = -\left(\frac{R_F}{R_1} v_{I1} + \frac{R_F}{R_2} v_{I2} + \frac{R_F}{R_3} v_{I3}\right) \quad (\text{Ecuación 3.3-6})$$

El voltaje de salida es la suma de los tres voltajes de entrada, con diferentes factores de ponderación. Este circuito es algunas veces llamado **el amplificador sumador inversor**. El número de terminales y resistores de entrada puede ser cambiado para agregar más o menos voltajes.

Un caso especial ocurre cuando las tres resistencias de entrada son iguales. Cuando  $R_1=R_2=R_3=R$ , entonces:

$$v_o = -\frac{R_F}{R}(v_{I1} + v_{I2} + v_{I3}) \quad (\text{Ecuación 3.3-7})$$

Esto significa que el voltaje de salida es la suma de los voltajes de entrada, con un factor de amplificación.

## Ejercicio

Considera un amplificador sumador ideal como el que se muestra en la FIGURA 1 .

### Datos:

$$R_1=10 \text{ k}\Omega$$

$$R_2=20 \text{ k}\Omega$$

$$R_3=30 \text{ k}\Omega$$

$$R_F=40 \text{ k}\Omega$$

Determina el voltaje de salida  $v_o$  si  $V_{II}=250 \sin(1000\pi t) \mu\text{V}$ ,  $v_{I2}=200 \sin(2000\pi t) \mu\text{V}$ ,  $v_{I3}=75 \sin(3000\pi t) \mu\text{V}$ .

## Simulación del Circuito

1. Incorpore a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH.
- Resistencias de 10k W, 20k W, 30k W y 10k W.
- Dos fuentes de voltaje de DC (15 V y -15 V).
- Tres tierras.
- Generador de funciones.
- Osciloscopio.

2. Conecta el circuito conforme a la Figura 3.3-2

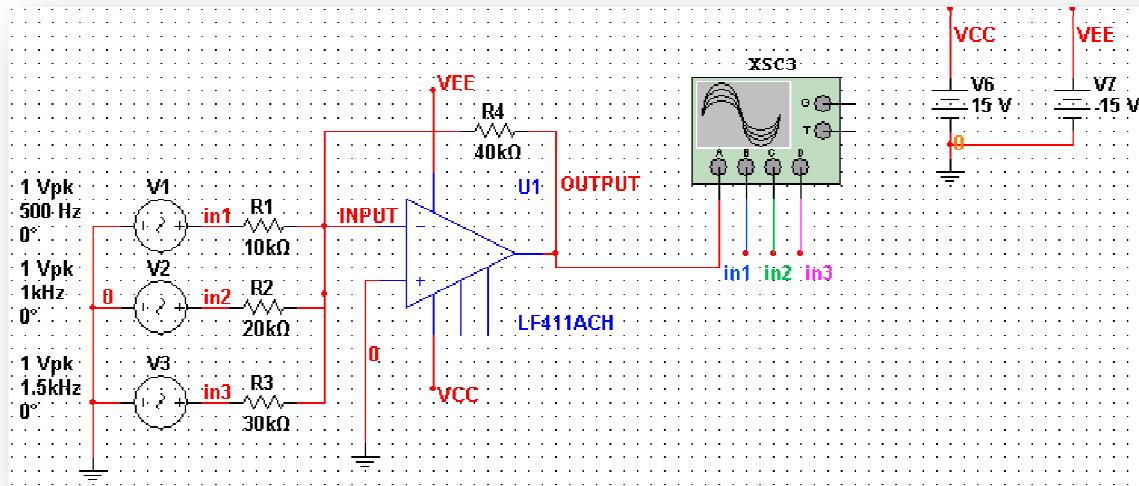


Figura 3.3-2

3. Configura cada una de las fuentes de alimentación de entrada como se muestra a continuación, toma en cuenta que para cada entrada corresponde una frecuencia diferente.

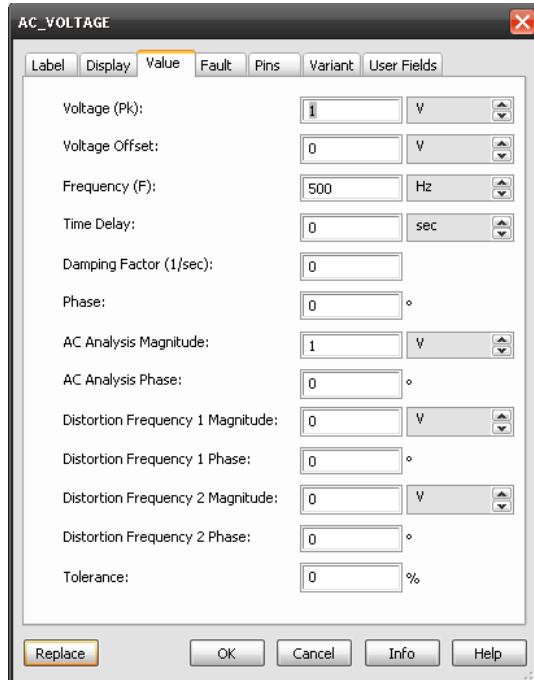
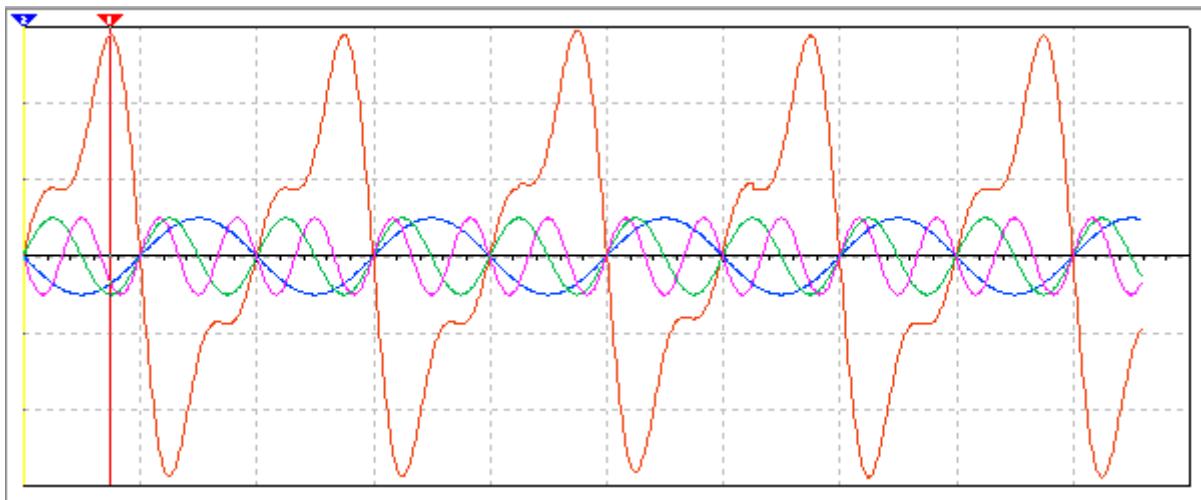


Figura 3.3-3

4. Corre la simulación y da doble clic al Osciloscopio. Los voltajes de entrada se muestran de acuerdo a los colores señalados en el esquemático.



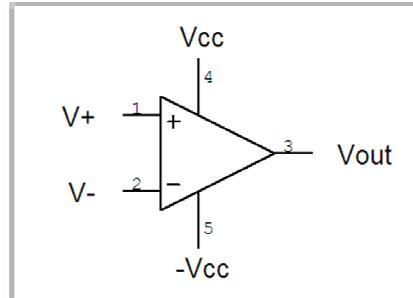
**Figura 3.3-4**

**Reto.**

Cambia los valores de los resistores y fuentes de voltaje en la entrada. Observa los cambios en la señal que representa la salida del sumador.

## 3.4 Circuito Derivador e Integrador

El símbolo del amplificador operacional



ni.com

 NATIONAL  
INSTRUMENTS™

98

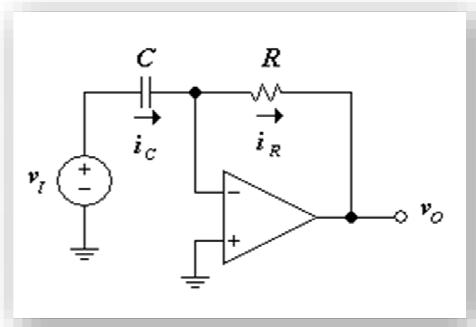
## Amplificadores Operacionales

Los amplificadores operacionales pueden conectarse en múltiples configuraciones para obtener distintas funcionalidades como sumar y restar, por ejemplo. En este caso, se va a revisar el uso de los amplificadores operacionales como integradores y derivadores, de tal manera que si ustedes conectan a la entrada de alguna de estas configuraciones una cierta función  $f(t)$ , podrán obtener tanto su integral.

$$\int f(t)dt \quad \text{como su derivada} \quad \frac{df(t)}{dt}$$

### 3.4.1 El Derivador

Es un circuito en el que la señal de salida es proporcional a la derivada en el tiempo de la señal de entrada.



ni.com

NATIONAL INSTRUMENTS

#### Definición del Problema

Si observamos la Figura 3.4-1 y consideramos amplificador operacional como ideal, podemos decir que la corriente que entra a la terminal inversora del amplificador operacional es cero, por lo cual podemos asegurar que  $i_c = i_R$ . Si expresamos dichos valores de corriente en términos de voltaje y despejamos el voltaje de salida obtenemos la siguiente ecuación:

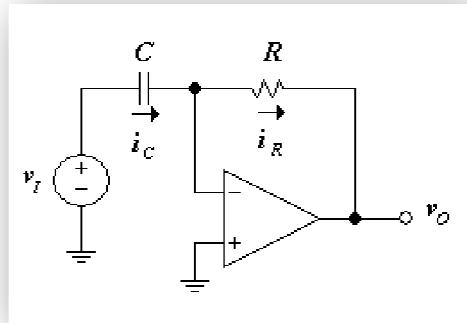
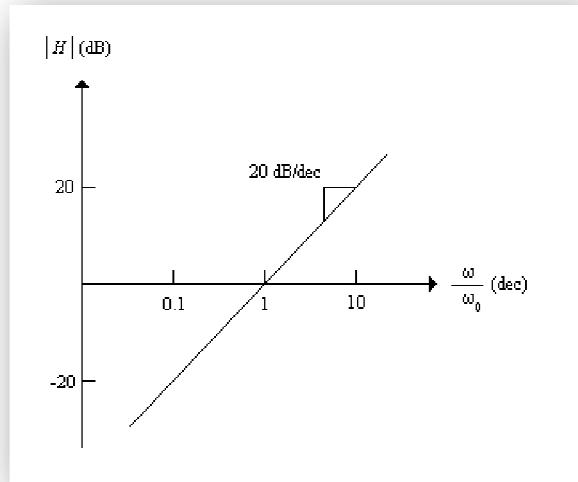
$$v_o(t) = -RC \frac{dv_I(t)}{dt} \quad (\text{Ecuación 3.4-1})$$

La función de transferencia de este circuito es una recta definida por la siguiente ecuación:

$$H(s) = -RCs \quad (\text{Ecuación 3.4-2})$$

Si sustituimos  $s$  por  $j\omega$  y buscamos el punto de intersección con el eje  $x$ , es decir, cuando la ganancia  $H(j\omega)$  es 1 (0 dB), entonces tenemos que la frecuencia de corte del derivador es:

$$\omega_0 = \frac{1}{RC} \quad (\text{Ecuación 3.4-3})$$

**Figura 3.4-1** Circuito Derivador**Figura 3.4-2**

Conociendo la respuesta en frecuencia del derivador, diseñemos uno con ganancia unitaria y frecuencia de corte de 1 kHz. Si hacemos  $C = 1\mu F$ , y sustituimos en la ecuación 1.2 obtenemos que:

$$R = \frac{1}{2\pi(1 \times 10^3)(1 \times 10^{-6})} = 159.15\Omega \quad (\text{Ecuación 3.4-4})$$

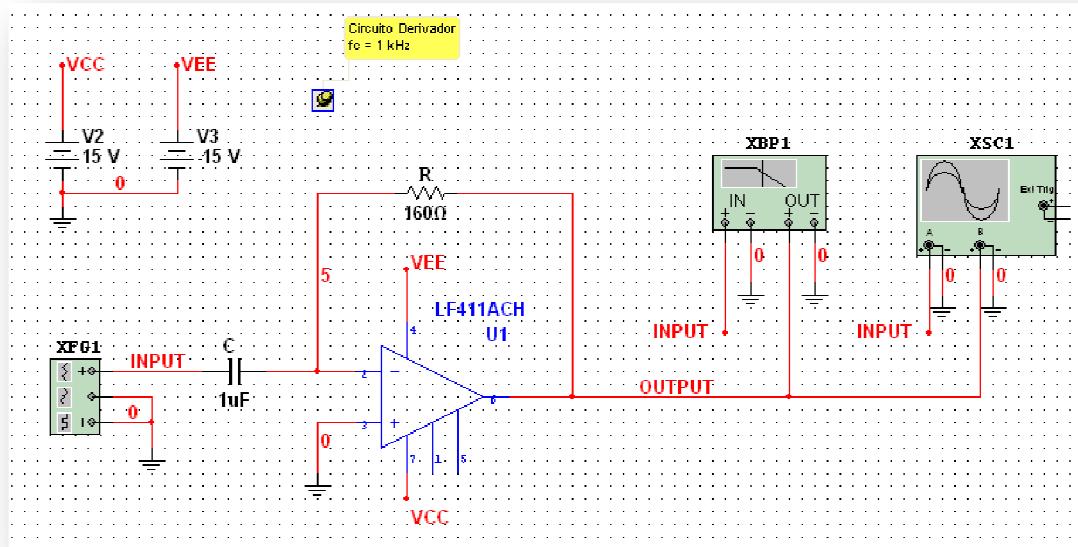
La resistencia comercial a 5% de tolerancia más cercana es 160  $\Omega$ .

## Simulación.

1. Incorpore a NI Multisim 10.1 los siguientes elementos:

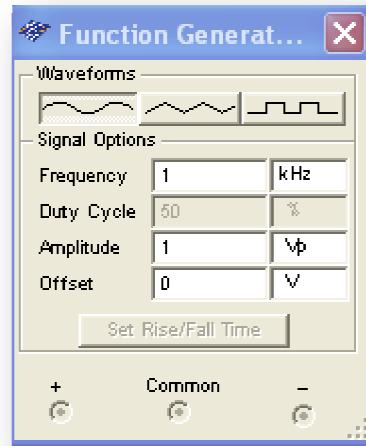
- Amplificador Operacional LF411ACH
- Resistencia de 160  $\Omega$
- Capacitor de 1 $\mu F$
- Dos fuentes de voltaje de DC (15 V y -15 V)
- Cuatro tierras
- Generador de funciones
- Graficador de Bode
- Osciloscopio

2. Conecta el circuito conforme a la Figura 3.1 y verifícalo con respecto al circuito mostrado en la Figura 3.3



**Figura 3.4-3** Circuito en NI Multisim 10.1 10.1

3. Configura el Generador de Funciones como se muestra en la Figura 1.3:



**Figura 3.4-4** Configuración del Generador de Funciones

4. Corre la simulación y da doble clic al Graficador de Bode. Si ajustas los rangos de los ejes y mueves el cursor a una frecuencia de 1 kHz, podrás verificar que el valor de la ganancia es prácticamente 0 dB como se muestra en la Fig. 3.4-5.

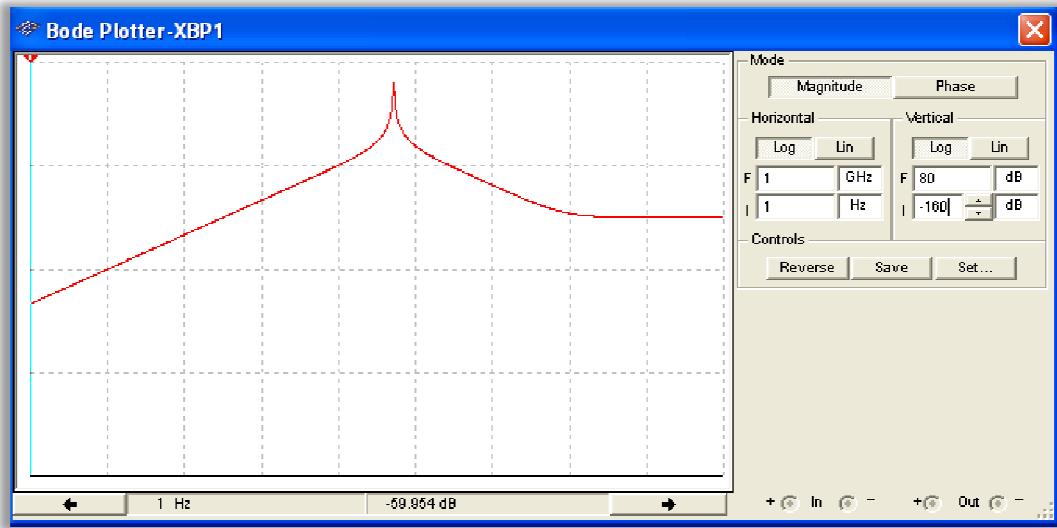


Figura 3.4-5 Grafica de Bode

5. Presiona el botón Phase para observar la respuesta de fase. De acuerdo a la ecuación 1.1 si aplicamos una señal de entrada  $\sin(t)$  esperamos como salida

$$-\frac{d\sin(t)}{dt} = -\cos(t) \quad (\text{Ecuación 3.4-5})$$

Eso significa que la salida será la misma senoidal desfasada  $-90^\circ$  multiplicada por la constante proporcional definida por R Y C. Este desfasamiento de  $-90^\circ$  lo podemos observar claramente en la respuesta de fase del circuito que se muestra en la Figura 3.4-6.

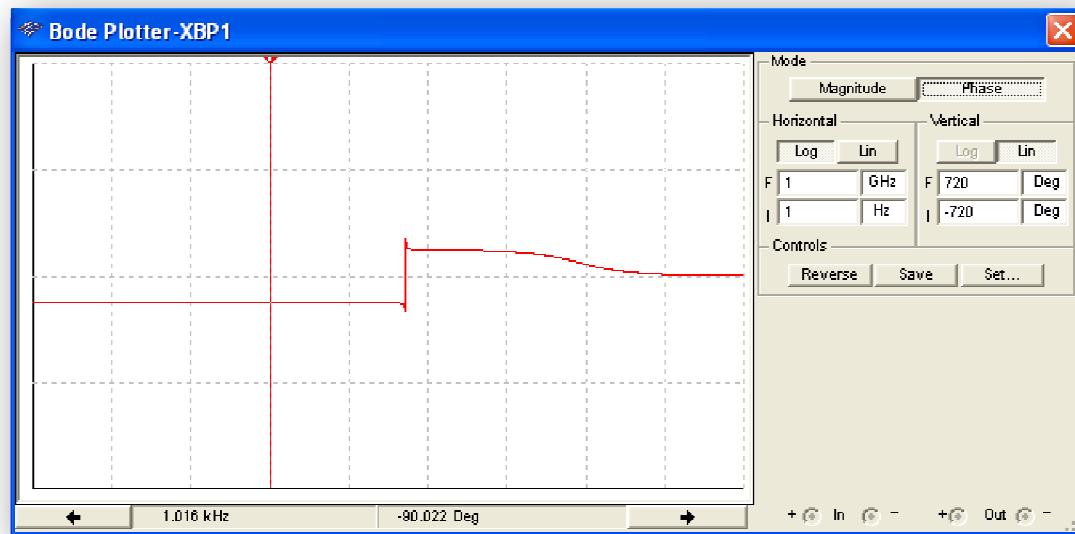


Figura 3.4-6

6. Cierra el Graficador de Bode y abre el Osciloscopio. Si los parámetros del Generador de Funciones son los que aparecen en la Figura 1.3, entonces deberás poder observar dos ondas senoidales desfasadas  $90^\circ$  una de otra como se puede observar en la Figura 3-7

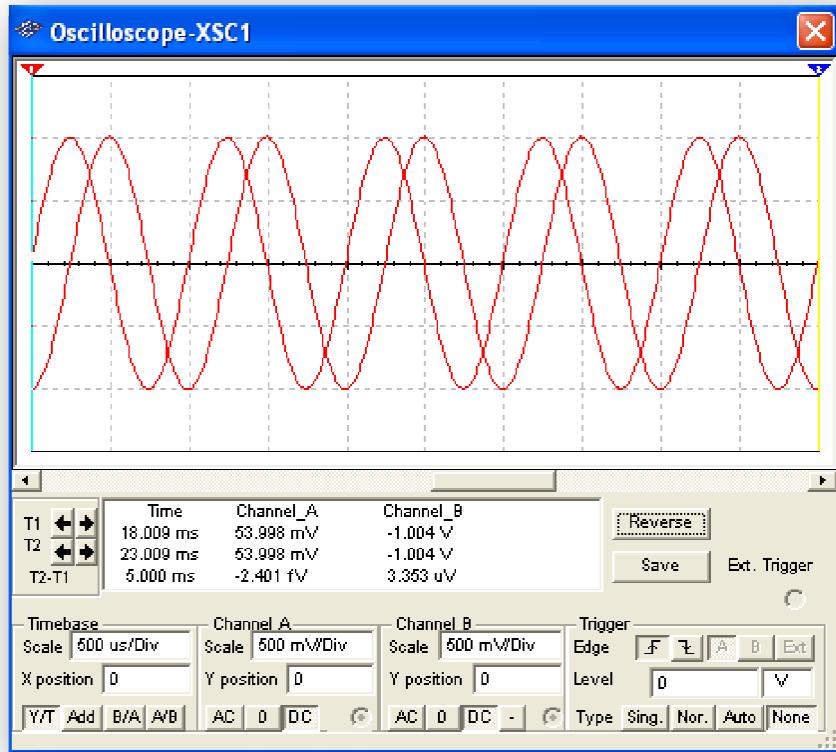
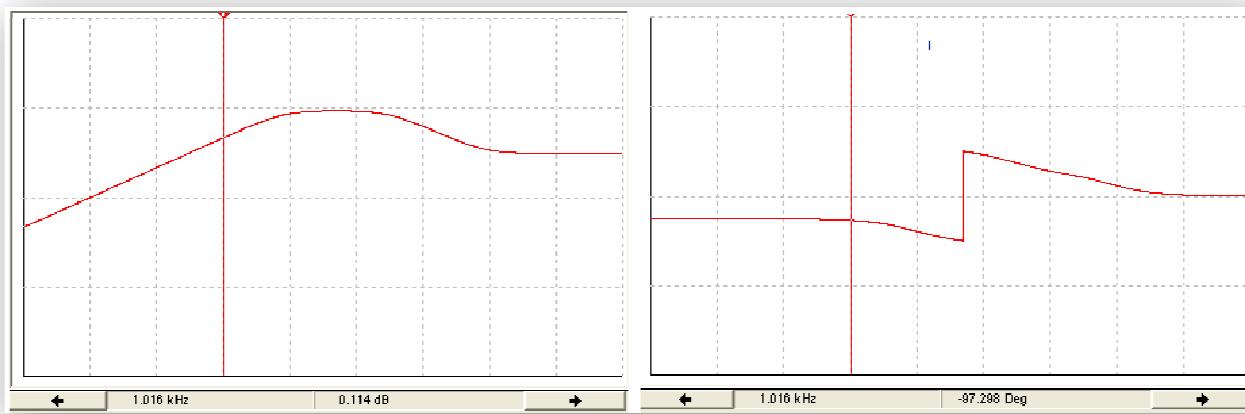
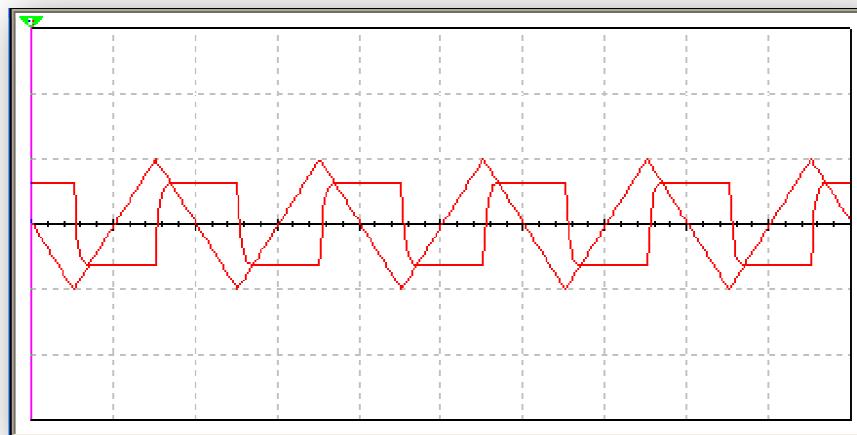


Figura 3.4-7

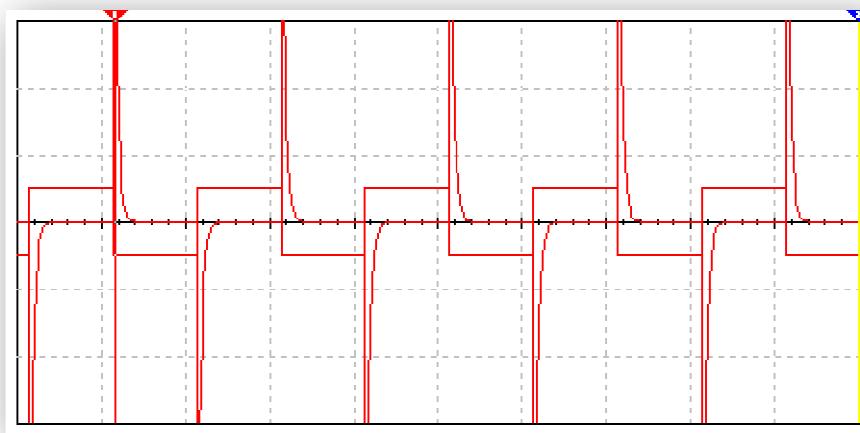
7. Haz pruebas variando las frecuencias por arriba y por debajo de la frecuencia de corte para que observes como la ganancia se ve seriamente afectada por la frecuencia.
8. Regresa a la frecuencia de corte. Cambia la señal de entrada por una triangular y posteriormente por una cuadrada. Para la derivada de una señal triangular esperamos una señal cuadrada y para la cuadrada esperamos un tren de impulsos. ¿Se obtienen las señales deseadas?
9. Las señales triangulares y cuadradas están conformadas por componentes frecuenciales por encima de la frecuencia de corte, fuera de la región lineal de la respuesta en frecuencia del circuito (Observar Figura 3-5). Si se agrega una resistencia en serie con el capacitor podemos suavizar esta curva y obtener los resultados deseados. Agrega una resistencia en serie con el capacitor y varía su valor hasta obtener curvas similares a las siguientes tanto para el Graficador de Bode como para el Osciloscopio.



**Figura 3.4-8** Bode: Magnitud y Fase



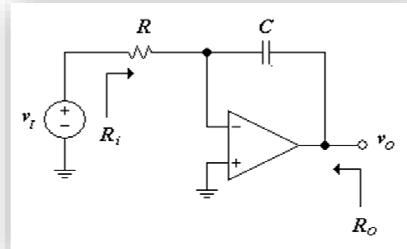
**Figura 3.4-9** Señal de entrada triangular



**Figura 3.4-10** Señal de entrada cuadrada

## 3.4.2 El Integrador

El circuito del integrador es muy similar al circuito del derivador, solamente se intercambian de posiciones la resistencia y el capacitor



ni.com

NATIONAL INSTRUMENTS

105

### Definición del Problema

La ecuación original es la misma que en el caso del derivador  $i_R = i_C$ . Al reemplazar las corrientes por sus expresiones en términos del voltaje de entrada y salida, se obtiene la siguiente expresión para el voltaje de salida.

$$v_o(t) = -\frac{1}{RC} \int_0^t v_i(\xi) d\xi + v_o(0) \quad (\text{Ecuación 3.4-6})$$

La frecuencia de corte  $\omega_0$  se calcula exactamente igual por lo que podemos utilizar los mismos valores para la resistencia y el capacitor.

La respuesta en frecuencia del circuito también es muy parecida a la del derivador, salvo porque esta tiene una pendiente negativa.

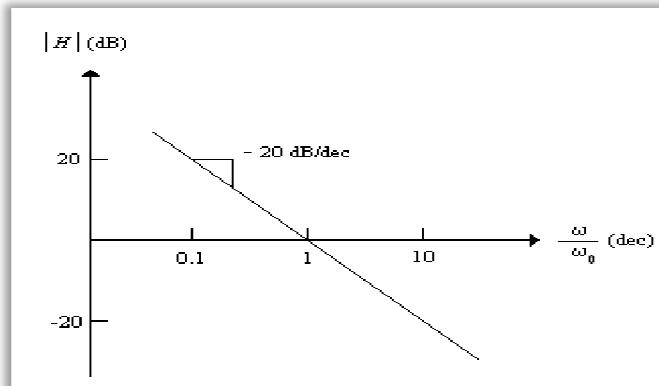


Figura 3.4-11

## Simulación del Circuito

1. Incorpora a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH
- Resistencia de 160 W
- Capacitor de 1 mF
- Dos fuentes de voltaje de DC (15 V y –15 V)
- Cuatro tierras
- Generador de funciones
- Graficador de Bode
- Osciloscopio

2. Conecta el circuito conforme a la Figura 3.4-11 y verifícalo con respecto al circuito mostrado en la Figura 3.4-12.

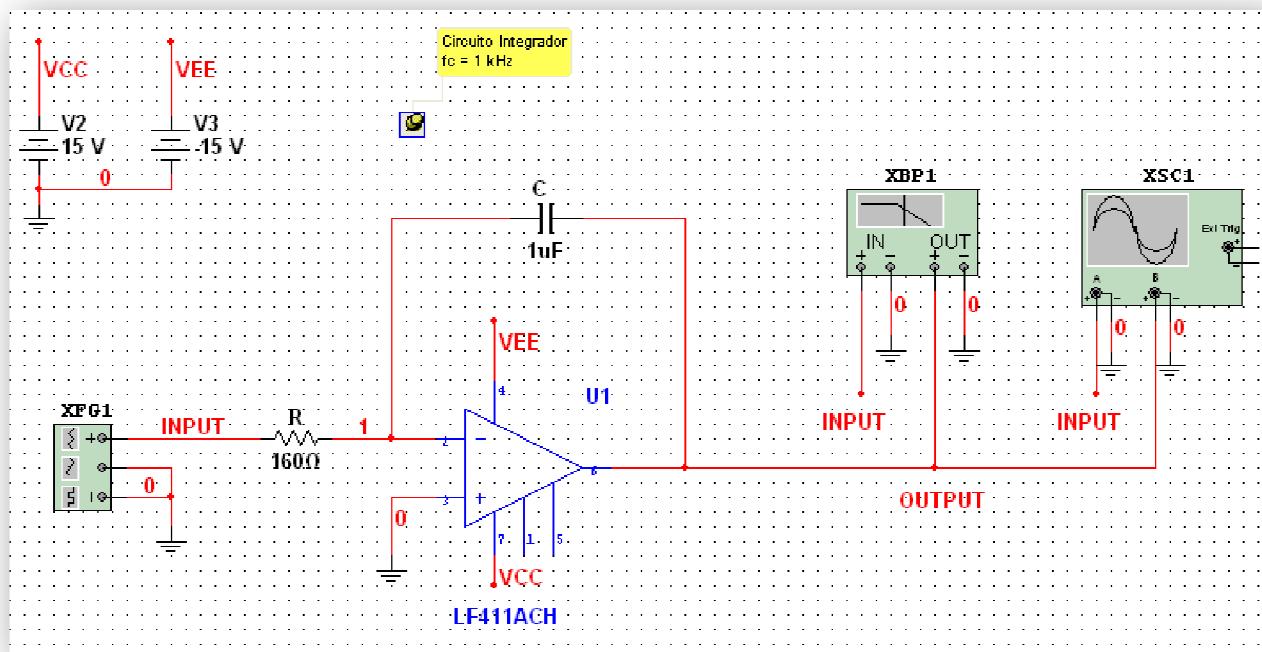
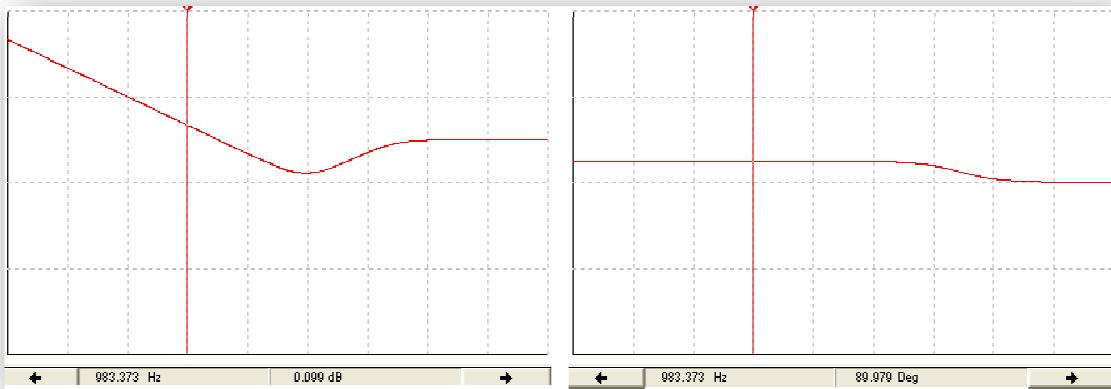


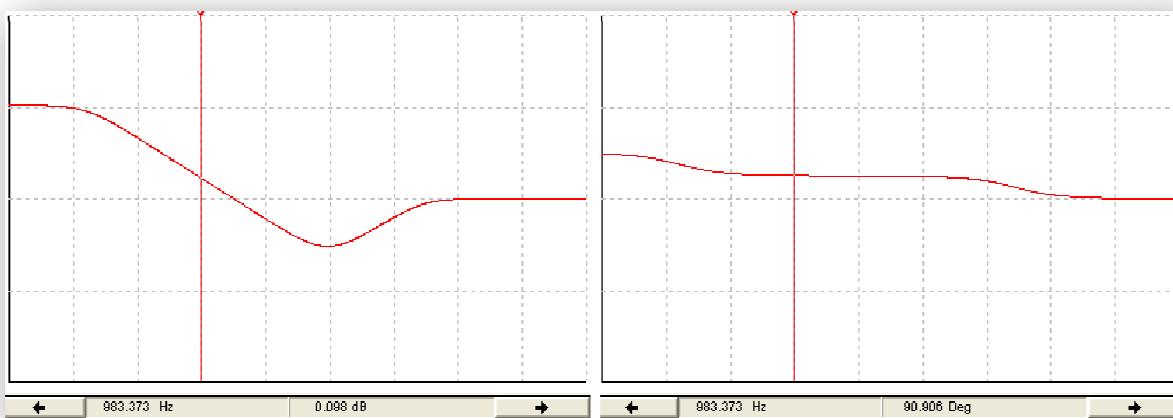
Figura 3.4-12

3. Abre el Graficador de Bode, observa las respuestas en magnitud y fase contrástalas con respecto al caso del derivador:



**Figura 3.4-13**

3. Ejecuta la simulación y abre el Osciloscopio. ¿Recibes los resultados esperados? En un circuito práctico el amplificador operacional se satura y esto se debe al error de offset de entrada. Una forma sencilla de corregir el error de offset es colocando una resistencia en paralelo con el capacitor. Esta no es la mejor solución pues limita el rango de frecuencia en el que trabaja el integrador. En aplicaciones reales no es necesario colocar esta resistencia, pues el integrador se utiliza normalmente dentro de un lazo cerrado (control PID) que evita que el amplificador se sature. Prueba con distintos valores de resistencia hasta que obtengas curvas similares a las siguientes:



**Figura 3.4-14** Bode: magnitud y fase

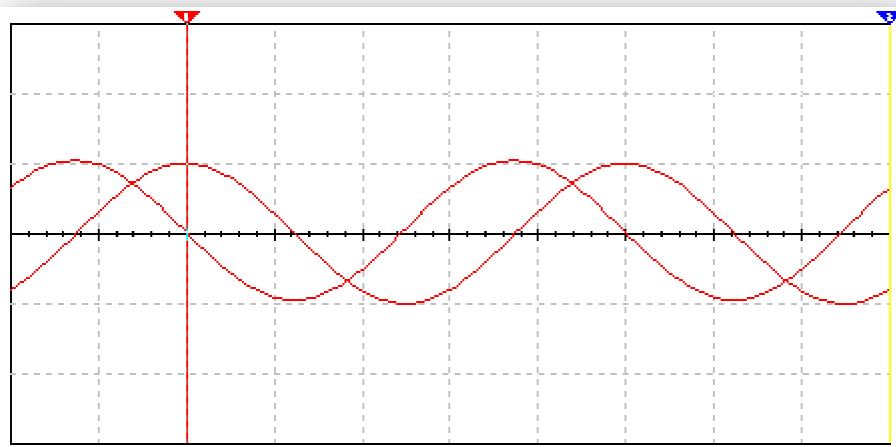


Figura 3.4-15

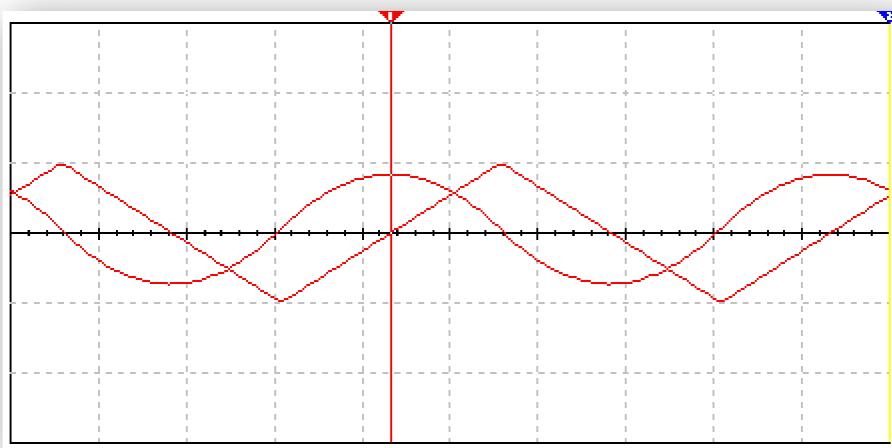


Figura 3.4-16

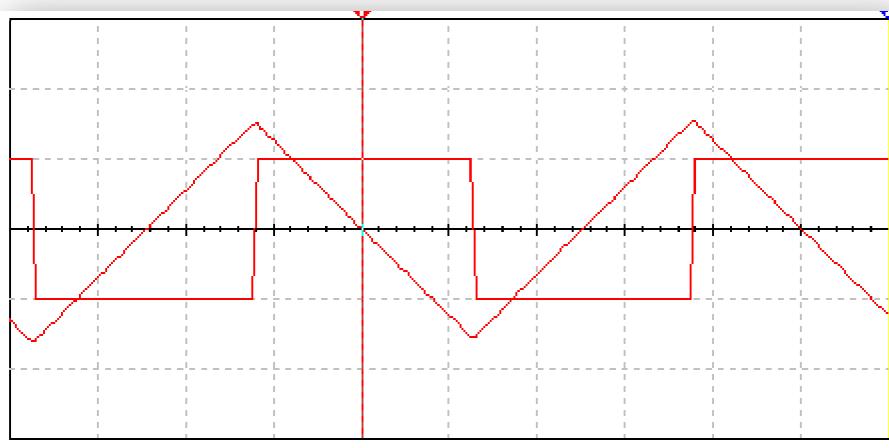


Figura 3.4-17

### 3. Ejercicio (Opcional)

Coloca el derivador en serie con el integrador para recuperar la señal original. Debes obtener resultados similares a los siguientes:

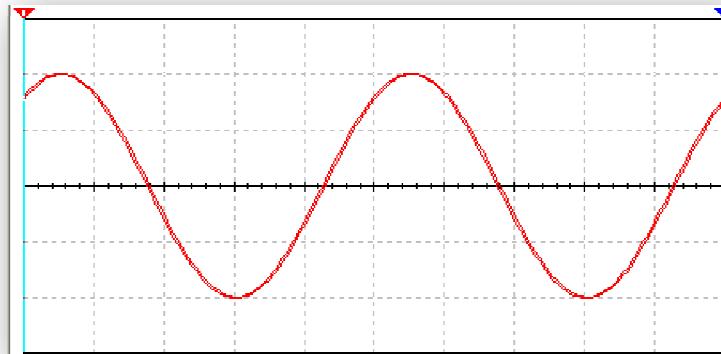


Figura 3.4-18

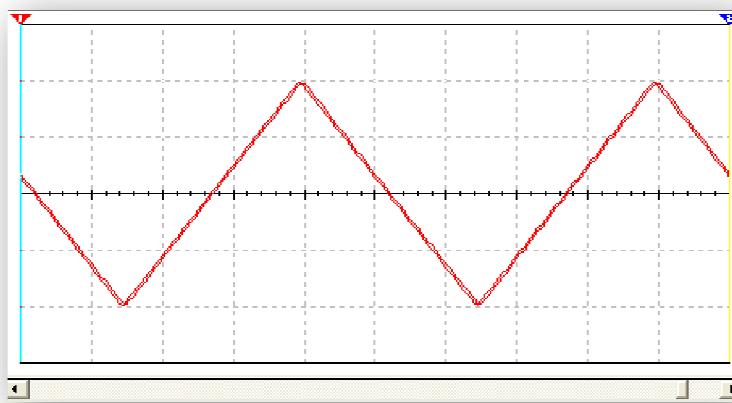


Figura 3.4-19

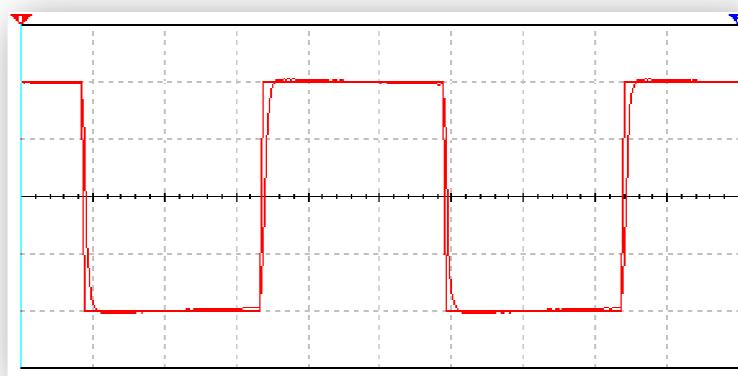


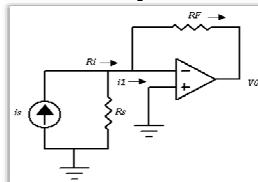
Figura 3.4-20

### Implementación del Circuito

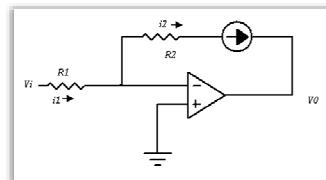
Utiliza la plataforma de ELVIS II para armar los circuitos derivador e integrador y verificar los resultados obtenidos durante la simulación.

### 3.5 Convertidores Voltaje - Corriente y Corriente a Voltaje

- Convertidor de corriente a voltaje.



- Convertidor de voltaje a corriente.



ni.com

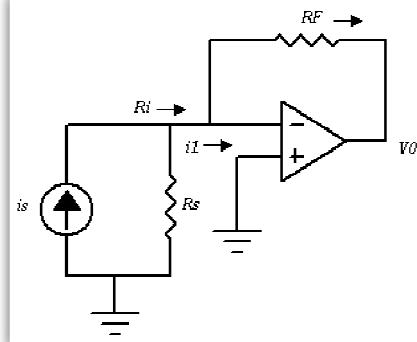
110



### 3.5.1 Convertidor de Corriente a Voltaje.

#### Introducción

Considera el circuito de la figura 3.5-1. La resistencia de entrada  $R_i$  en el nodo virtual a tierra:



**Figura 3.5-1**

$$R_i = \frac{V_1}{i_1} \approx 0 \quad (\text{Ecuación 3.5-1})$$

En algunos casos, podemos asumir que  $R_s \gg R_i$ ; por lo tanto, la corriente  $i_1$  es prácticamente igual a la corriente  $i_s$ . Entonces:

$$i_2 = i_1 = i_s \quad (\text{Ecuación 3.5-2})$$

$$v_o = -i_2 R_F = -i_s R_F \quad (\text{Ecuación 3.5-3})$$

El voltaje de salida es directamente proporcional a la señal de corriente, y la resistencia de retroalimentación  $R_F$  es la magnitud de la relación del voltaje de salida y la señal de corriente.

#### Simulación del Circuito

1. Incorpore a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH.
- Resistencia de 100 kW.
- Dos fuentes de voltaje de DC (15 V y -15 V).
- Cuatro tierras.
- Osciloscopio.
- Fuente de corriente.

2. Conecta el circuito conforme a la Figura 3.5-1 y verifícalo con respecto al circuito mostrado en la Figura 3.5-2

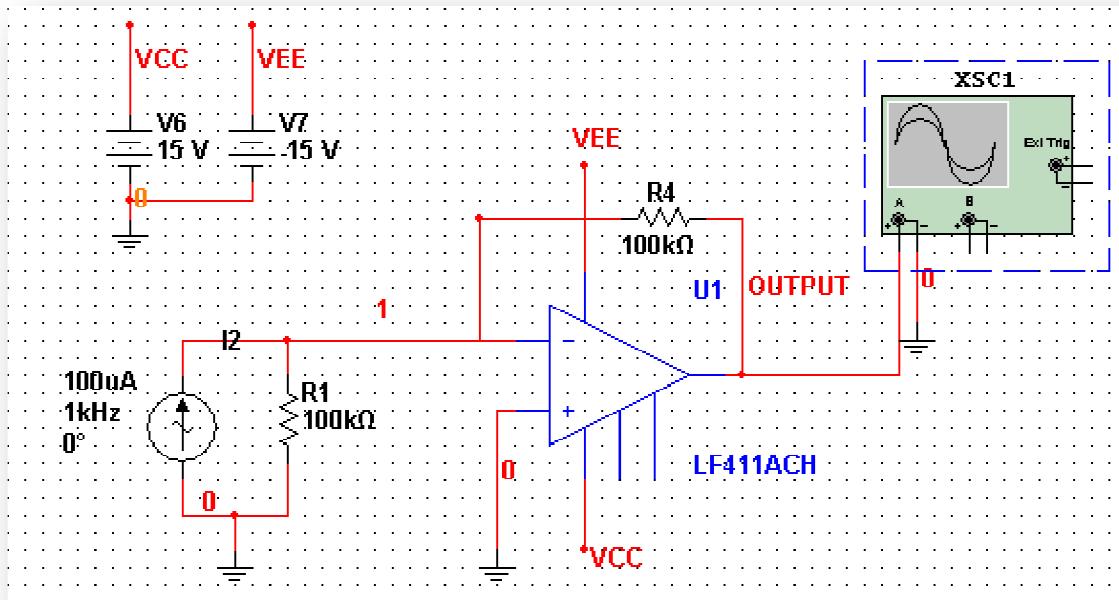


Figura 3.5-2

3. Configura la fuente de corriente de entrada como se indica a continuación:

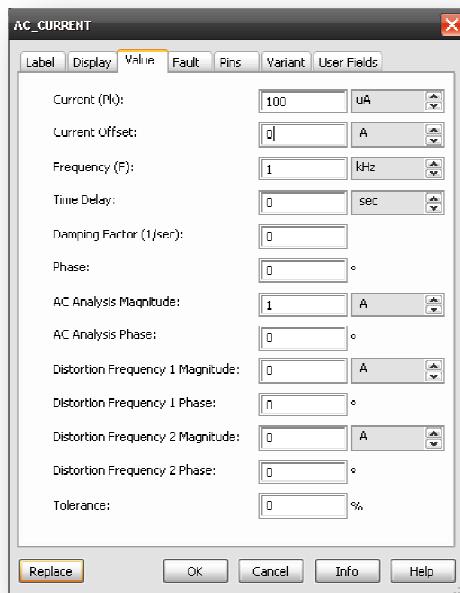
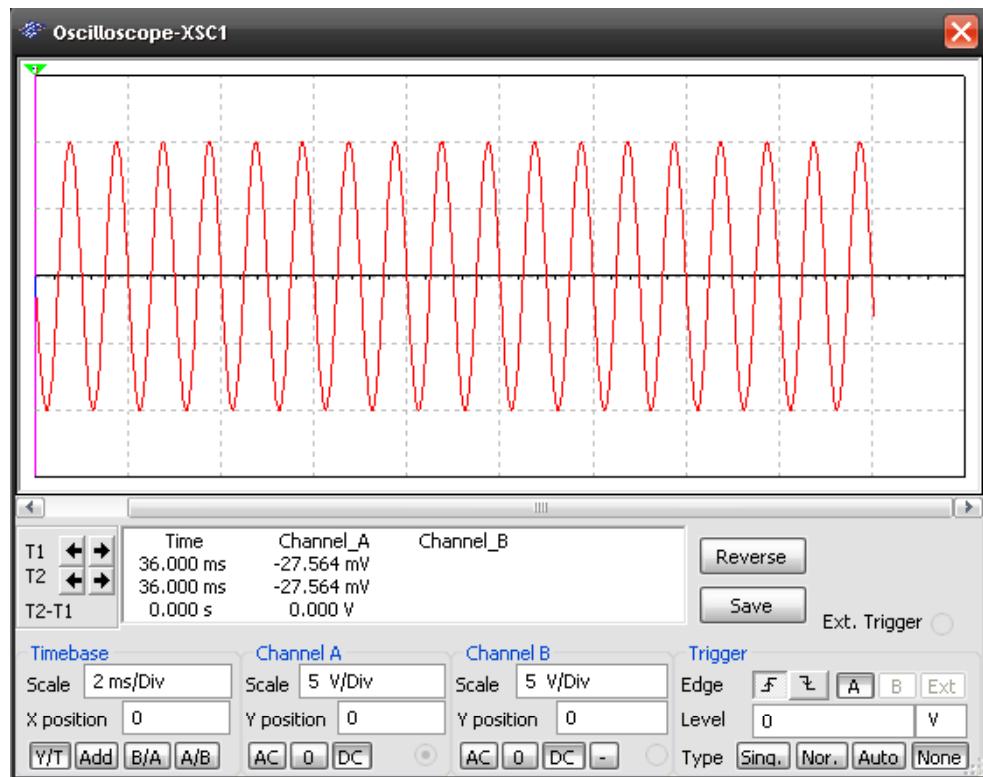


Figura 3.5-3

4. Ejecuta la simulación y abre el Osciloscopio. ¿Recibes los resultados esperados?  
 La figura 4 muestra el voltaje de salida el cual tiene una amplitud de 10V.



**Figura 3.5-4**

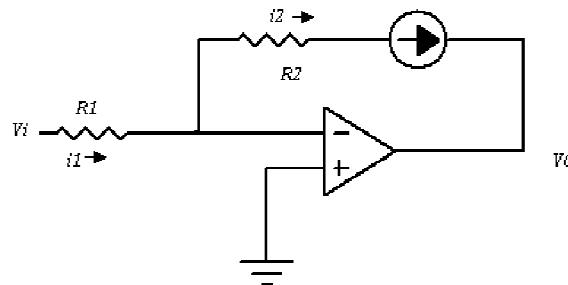
## Reto

Cambia los valores del resistor de retroalimentación. Observa los efectos que los valores de éste tienen en el voltaje de salida.

### 3.5.2 Convertidor de Voltaje a Corriente.

#### Introducción

El complemento del convertidor de corriente a voltaje es el convertidor de voltaje a corriente. Para esto podemos utilizar el amplificador operacional inversor mostrado en la siguiente figura:



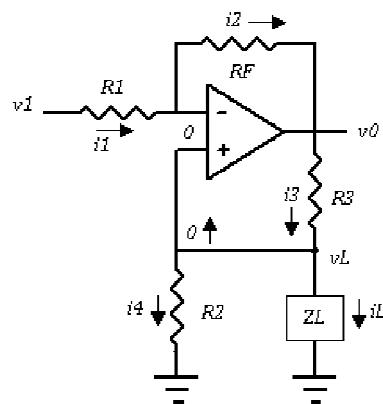
**Figura 3.5-5** Convertidor de Voltaje a Corriente Sencillo.

Para este circuito:

$$i_2 = i_1 = \frac{V_I}{R_1} \quad (\text{Ecuación 3.5-4})$$

Lo cual quiere decir que la corriente  $i_2$  es directamente proporcional al voltaje de entrada  $v_I$  y es independiente de la impedancia o resistencia de carga  $R_2$ . Sin embargo, la carga podría requerir de un potencial a tierra, tal que el circuito de la figura 3.5-5 no es práctico para tales aplicaciones.

Consideremos ahora el circuito de la figura 3.5-6. Es este caso una terminal de la carga tiene una impedancia  $Z_L$  aterrizada.



**Figura 3.5-6**

Partiendo del concepto de corto circuito virtual,  $v_1=v_2$ . Observe que  $v_1=v_2=v_L=i_L Z_L$ . Igualando las corrientes  $i_1$  e  $i_2$ , tenemos:

$$\frac{v_1 - i_L Z_L}{R_1} = \frac{i_L Z_L - v_0}{R_F} \quad (\text{Ecuación 3.5-5})$$

Al sumar las corrientes en la terminal no inversora:

$$\frac{v_0 - i_L Z_L}{R_3} = i_L + \frac{i_L Z_L}{R_2} \quad (\text{Ecuación 3.5-6})$$

Resolviendo para  $(v_0 - i_L Z_L)$  de la ecuación (2) y sustituyendo en la ecuación (3) se produce:

$$\frac{R_F}{R_1} \cdot \frac{(i_L Z_L - v_1)}{R_3} = i_L + \frac{i_L Z_L}{R_2} \quad (\text{Ecuación 3.5-7})$$

Factorizando:

$$i_L \left( \frac{R_F Z_L}{R_1 R_3} - 1 - \frac{Z_L}{R_2} \right) = v_1 \left( \frac{i_L}{R_1 R_3} \right) \quad (\text{Ecuación 3.5-8})$$

Con el objetivo de hacer a  $i_L$  más independiente de  $Z_L$ , podemos diseñar el circuito tal que el coeficiente de  $Z_L$  sea 0, ó:

$$\frac{R_F}{R_1 R_3} = \frac{1}{R_2} \quad (\text{Ecuación 3.5-9})$$

La ecuación 2.5 se reduce a:

$$i_L = -v_1 \left( \frac{R_F}{R_1 R_3} \right) = -\frac{v_1}{R_2} \quad (\text{Ecuación 3.5-10})$$

Lo cual significa que la corriente de carga es proporcional al voltaje de entrada y es independiente de la impedancia de carga  $Z_L$ , siempre y cuando el voltaje de salida permanezca entre los límites permitidos.

## Simulación del Circuito

1. Incorpora a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH.
- 2 Resistencias de 10 kW.
- 2 Resistencias de 1 kW.
- Resistencias de 100 W.
- Dos fuentes de voltaje de DC (10 V y –10 V).
- Fuente de voltaje AC a 5Vpp.
- Cuatro tierras.
- Osciloscopio.
- Fuente de corriente.

2. Conecta el circuito conforme a la Figura 3.5-6 y verifícalo con respecto al circuito mostrado en la Figura 3.5-7

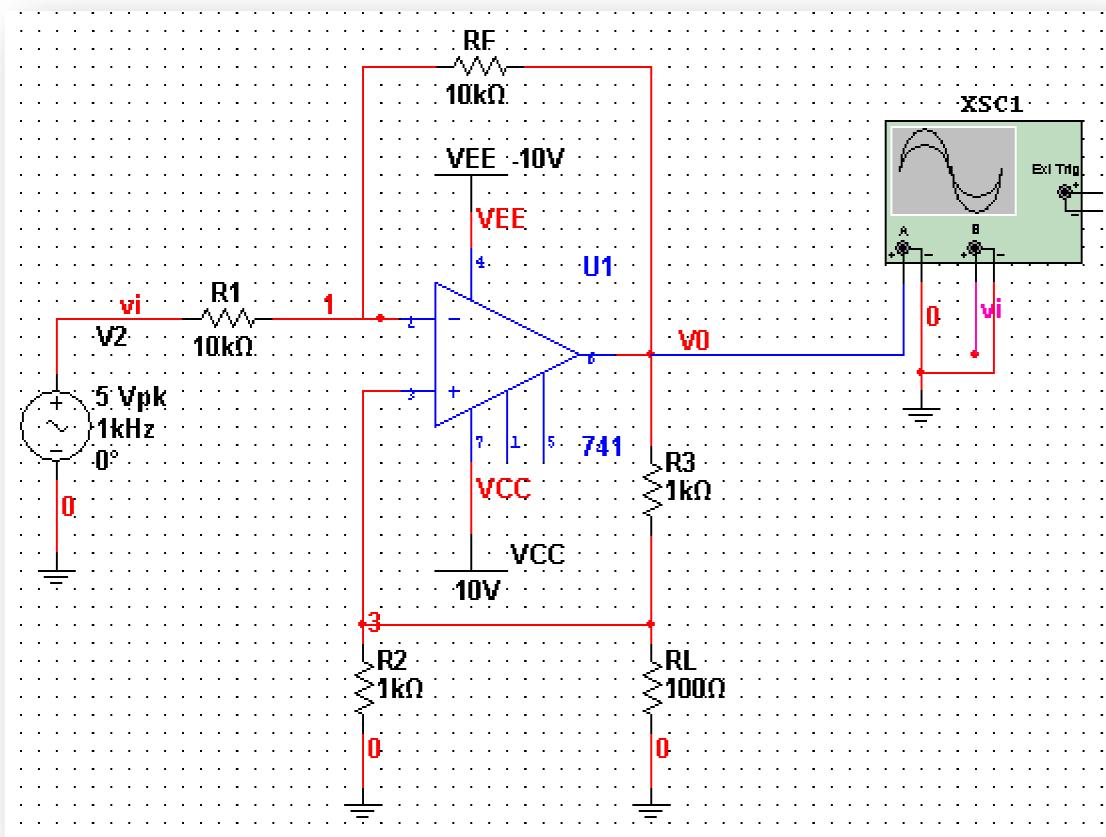
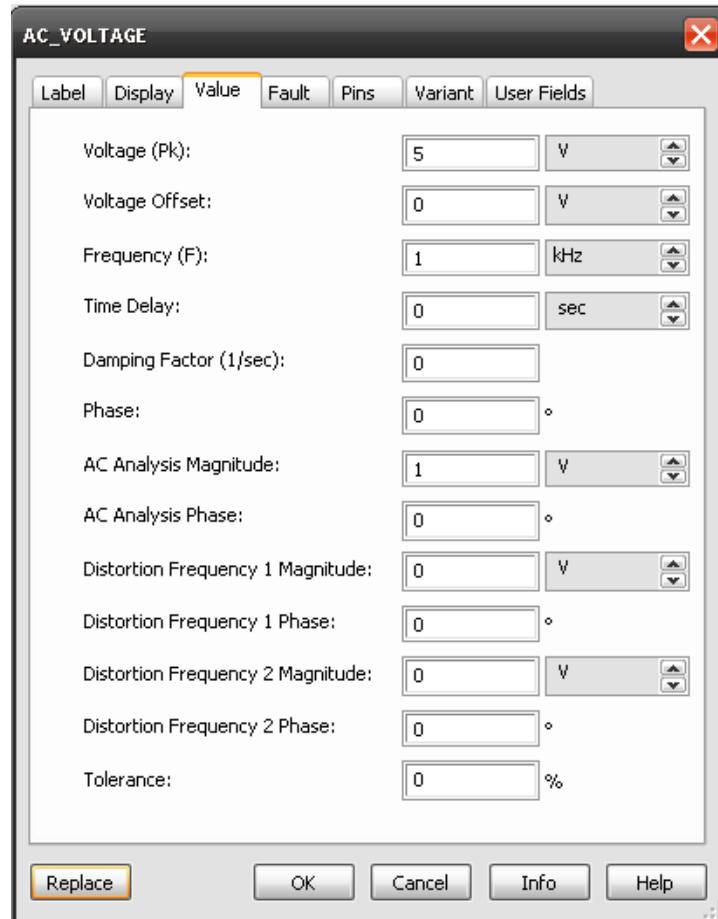


Figura 3.5-7

3. Configura la fuente de voltaje de AC como se muestra a continuación.



**Figura 3.5-8**

4. De doble clic en el osciloscopio y observe la salida del circuito con respecto a la entrada. ¿Son los resultados esperados?



Figura 3.5-9

5. Ahora, mide la corriente que se obtiene en la impedancia de carga. Coloca un multímetro en serie con la resistencia de carga. Observa la Figura 3.5-10

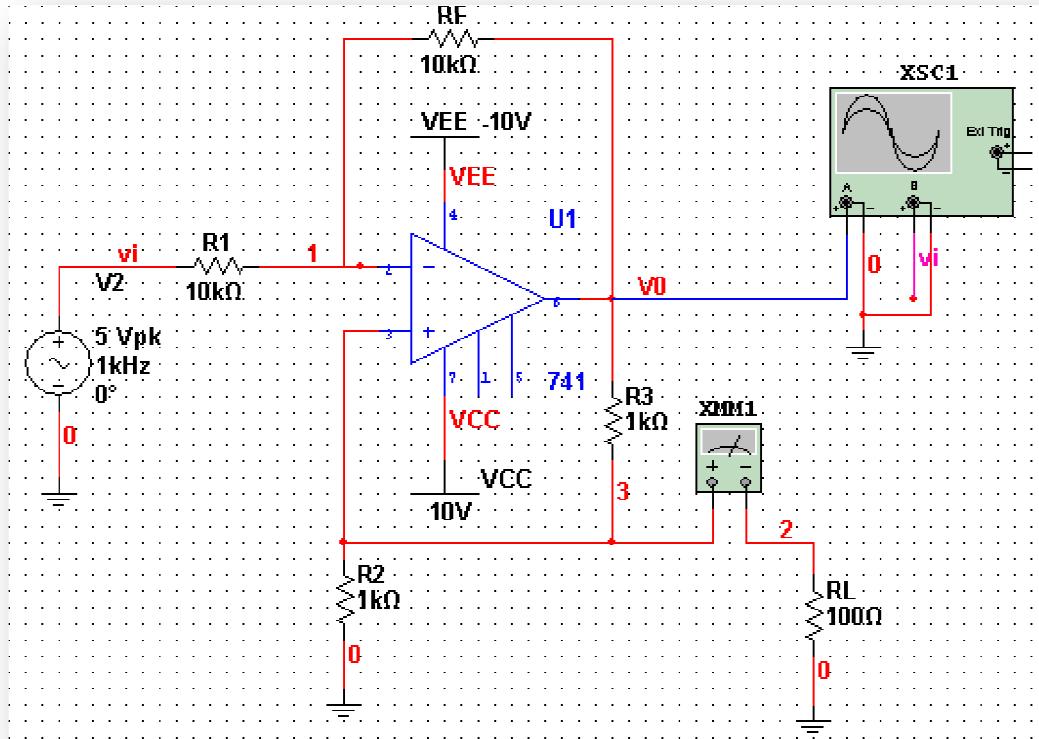


Figura 3.5-10

6. Da doble clic en el multímetro y observa la corriente que fluye a través de la resistencia de carga.



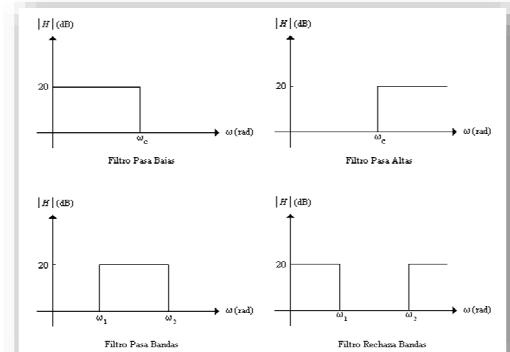
**Figura 3.5-11**

**Reto.**

1. Cambia los valores de voltaje de entrada así como su amplitud. Observa los resultados.

## 3.6 Filtros Activos de Primer Orden

Los filtros son sistemas que permiten o rechazan el paso a través de ellos de ciertas componentes frecuenciales de una cierta señal de entrada.



ni.com

120

### Introducción

Uno de las principales usos prácticos de los amplificadores operacionales es en la implementación de filtros. Aún cuando hoy en día los filtros digitales son una alternativa de filtrado sencilla de implementar, barata y muy eficiente, los filtros analógicos siguen estado vigentes, sobre todo en las etapas de conversión analógico-digital y digital-analógico, por ejemplo, los filtros anti-alias. Los filtros analógicos pueden ser pasivos (implementados en circuitos RLC con ganancia = 0) o activos (implementados con amplificadores operacionales con ganancia > 0). A lo largo de la práctica estaremos revisando distintos tipos de filtros activos de distintos órdenes.

Los filtros son sistemas que permiten o rechazan el paso a través de ellos de ciertas componentes frecuenciales de una cierta señal de entrada. Los filtros se diseñan a partir de ciertas formas ideales de su función de transferencia. De acuerdo a su forma y funcionalidad, los principales tipos de filtro son los siguientes (Figura 3.6-1):

- Filtros Pasa Bajas
- Filtros Pasa Altas
- Filtros Pasa Bandas
- Filtros Rechaza Bandas

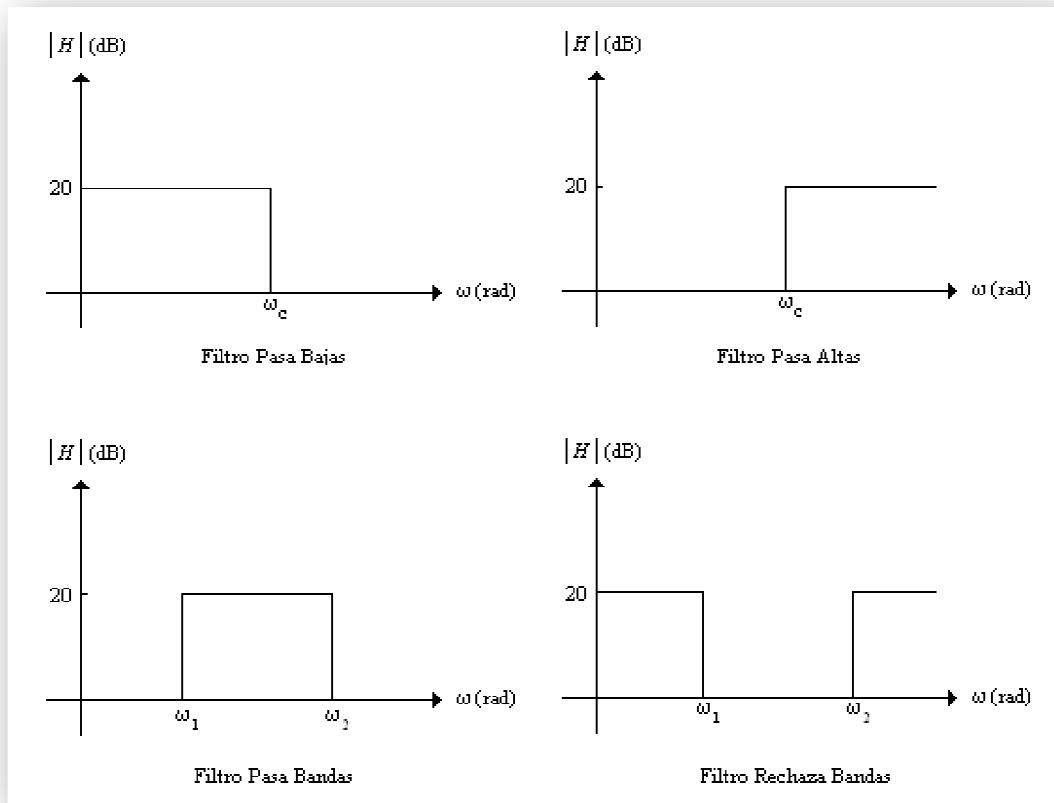


Figura 3.6-1

La Figura 3.6-1 muestra filtros ideales que sirven como modelo para diseñar filtros físicamente posibles de implementar. La forma de un filtro real no es tal cual un rectángulo, la ganancia se va atenuando en función de una cierta pendiente de atenuación (*Roll-off*) y la frecuencia de corte es la frecuencia a la que la señal ha perdido la mitad de su amplitud, comúnmente expresada como una caída de 3 dB en una escala logarítmica.

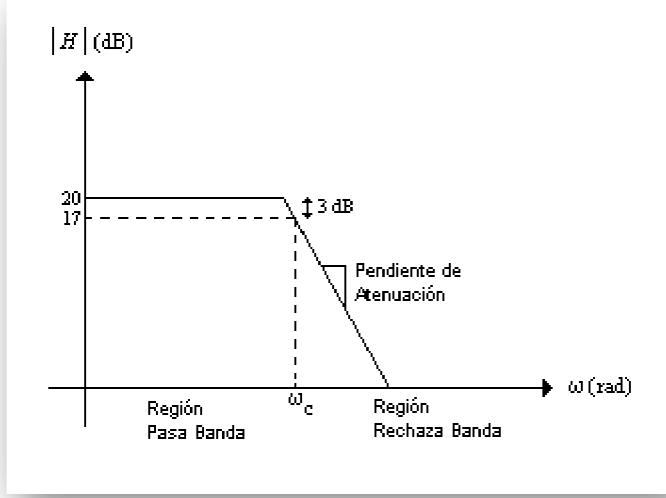
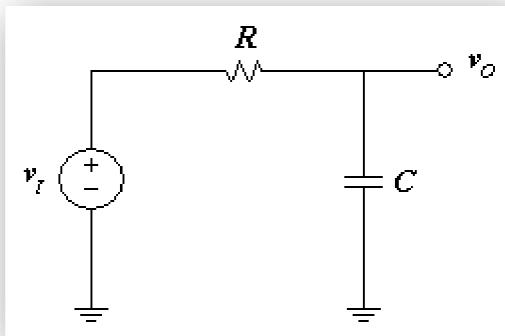


Figura 3.6-2

### 3.6.1 Filtros Pasa Bajas

Tal como se muestra en la Figura A de la Introducción, el Filtro Pasa Bajas permite el paso de las señales cuya frecuencia se encuentra por debajo de la frecuencia de corte (Región Pasa Banda). Para comprender un poco más el funcionamiento de un filtro pasa bajas revisemos brevemente el circuito mostrado en la Figura 3.6-3



**Figura 3.6-3**

El circuito mostrado en la Figura 3.6-3 es un filtro pasa bajas pasivo de primer orden. Si partimos del supuesto de que el capacitor a frecuencias muy altas ofrece muy baja impedancia y se comporta prácticamente como un cortocircuito y a frecuencias muy bajas ofrece muy alta impedancia y se comporta prácticamente como un circuito abierto, podemos ver que cuando pasa una señal de alta frecuencia se hará un cortocircuito a tierra y el voltaje de salida será cero, mientras que una señal de baja frecuencia pasará directamente a la salida pues el capacitor se comporta como circuito abierto. La impedancia del capacitor cambia de forma gradual formando una curva similar a la de la mostrada en la figura B de la parte introductoria. Sin embargo, puesto que se trata de un filtro pasivo, la ganancia del filtro es de 0 dB en la región pasa banda.

La frecuencia de corte se calcula a partir de la siguiente expresión:

$$\omega_0 = \frac{1}{RC} \quad (\text{Ecuación 3.6-1})$$

Para implementar un filtro pasa bajas activo, incorporemos el uso de amplificadores operacionales. En prácticas anteriores habíamos revisado los circuitos de los derivadores e integradores, así como del amplificador inversor. Combinemos ahora estos circuitos y revisemos el circuito de la Figura 3.6-4

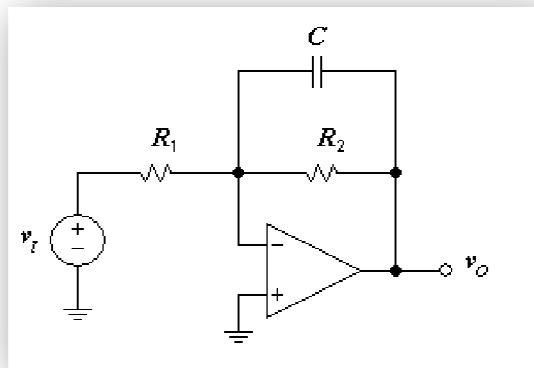


Figura 3.6-4

Si observamos este circuito podemos hacer un análisis parecido al que hicimos para el filtro pasivo. A frecuencias bajas, la impedancia del capacitor es mucho mayor que la ofrecida por , por lo que se puede despreciar su impedancia, dejando prácticamente el circuito de un amplificador inversor, cuya ganancia (dB) en la región pasa banda podemos calcular a partir de la siguiente expresión:

$$H_0 = -\frac{R_2}{R_1} \quad (\text{Ecuación 3.6-2})$$

En altas frecuencias la resistencia tiene una impedancia mucho mayor que la que ofrece el capacitor, por lo que se puede ignorar dejando el circuito como un integrador. La frecuencia de corte de este circuito se calcula como:

$$f_0 = \frac{1}{R_2 C} \quad (\text{Ecuación 3.6-3})$$

Realicemos el diseño de un filtro bajas con una frecuencia de corte de 1 kHz. Si fijamos el valor del capacitor en 100 nF y sustituimos los valores en la siguiente expresión obtenemos lo siguiente:

$$R = \frac{1}{2\pi(1 \times 10^3)(1 \times 10^{-7})} = 1591.5\Omega \quad (\text{Ecuación 3.6-4})$$

El valor más cercano de resistencia comercial con tolerancia del 5% es de 1.6 kΩ.

## Simulación del Circuito

1. Incorpora a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH
- Dos Resistencias de  $1.6\text{k}\Omega$
- Un Capacitor de  $100\text{nF}$
- Dos fuentes de voltaje de DC ( $15\text{V}$  y  $-15\text{V}$ )
- Generador de funciones
- Graficador de Bode
- Osciloscopio

2. Conecta el circuito conforme a la Figura 3.6-4 y verifícalo con respecto al circuito mostrado en la Figura 3.6-5

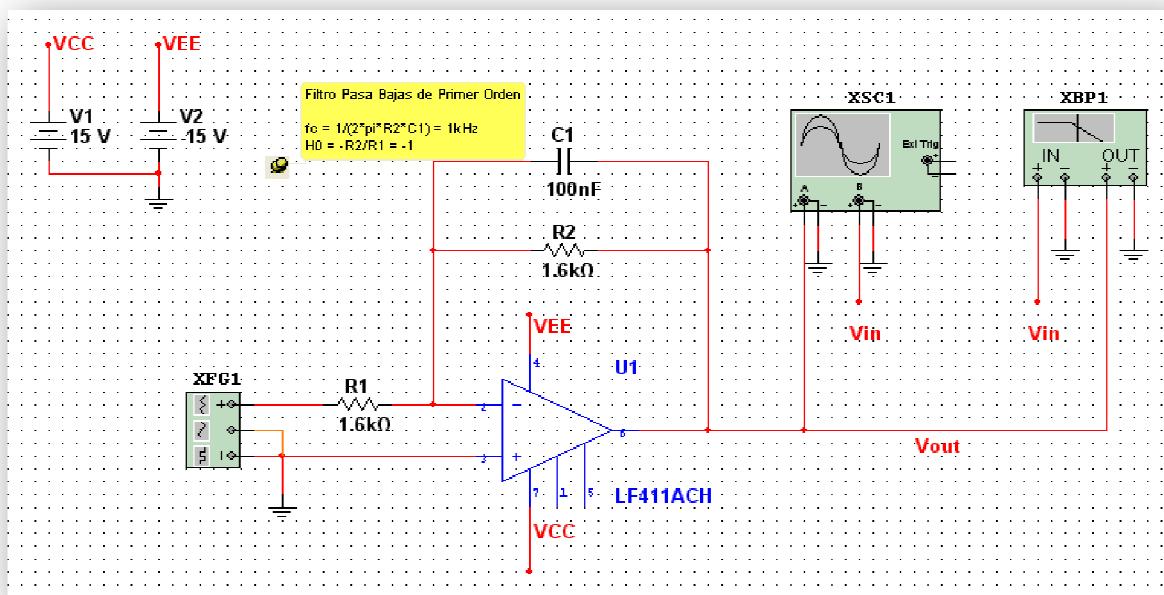


Figura 3.6-5

3. Configura el Generador de Funciones como se muestra en la Figura 3.6-6

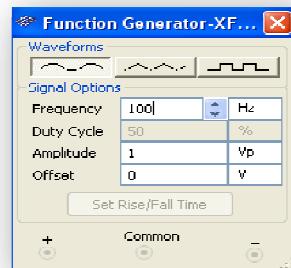
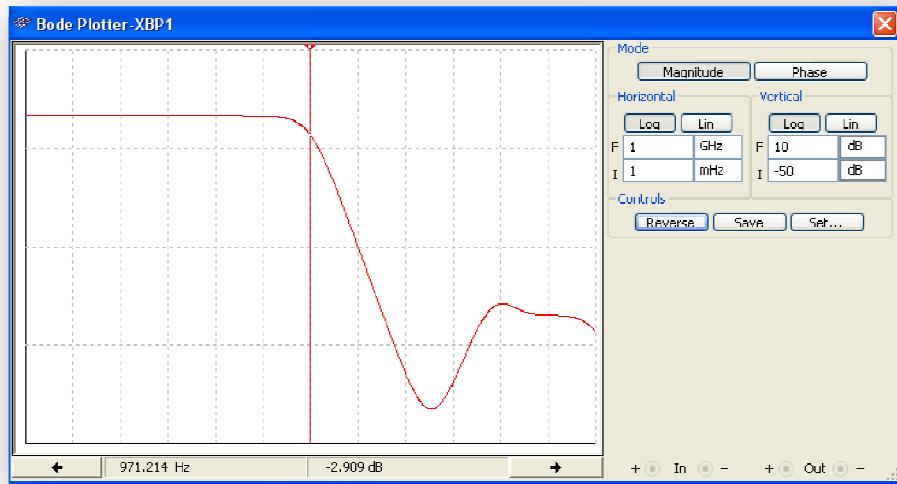


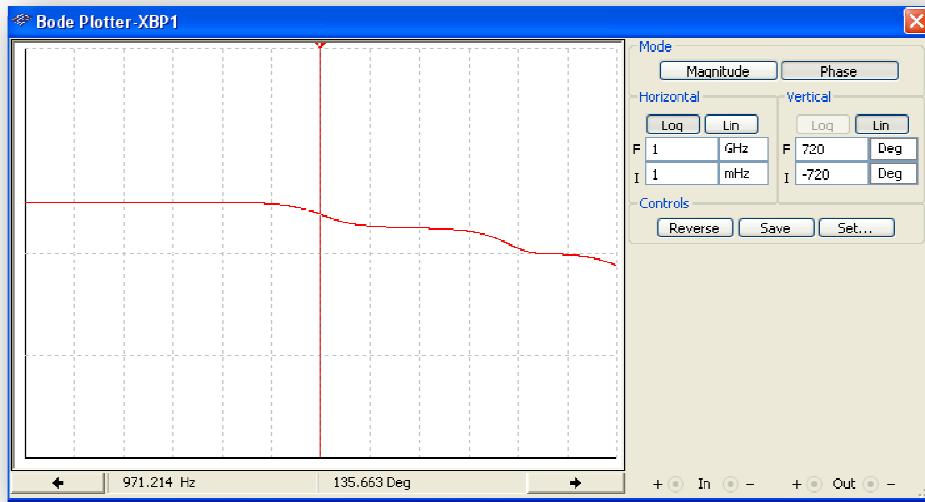
Figura 3.6-6

4. Corre la simulación y da doble clic al Graficador de Bode. Si ajustas los rangos de los ejes y mueves el cursor a una frecuencia de 1 kHz, podrás verificar que el valor de la ganancia es prácticamente de -3 dB como se muestra en la Figura 3.6-7



**Figura 3.6-7**

5. Presiona el botón Phase para observar la respuesta de fase. Se puede observar en la Figura 3.6-8 que la respuesta de fase va de  $180^\circ$  a  $90^\circ$  pasando justo a la mitad a la frecuencia de corte a 1 kHz.



**Figura 3.6-8**

6. Cierra el Graficador de Bode y abre el Osciloscopio. Con los parámetros del Generador de Funciones configurados como se muestra en la Figura 3.6-9 podrás ver como la señal pasa con una diferencia de fase de  $180^\circ$  sin perder casi amplitud.

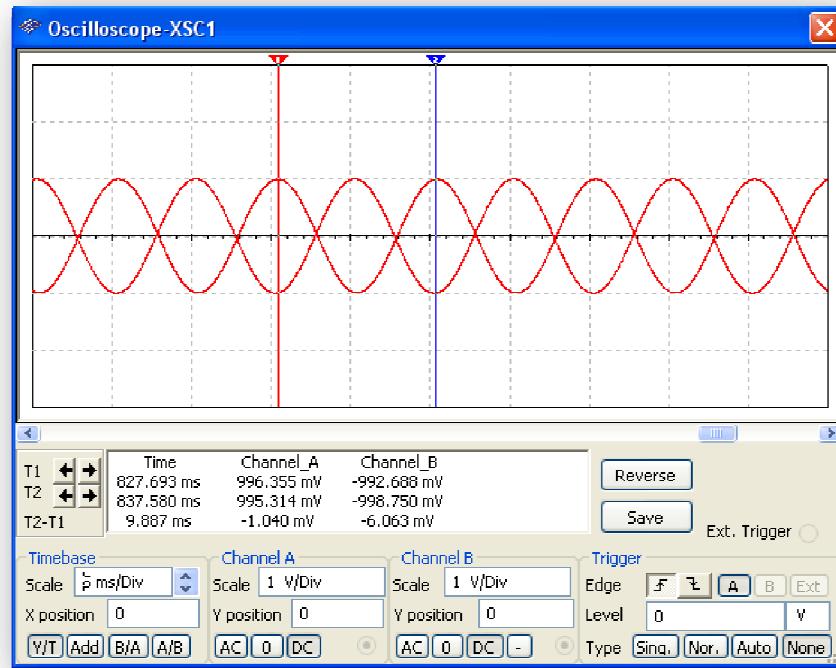


Figura 3.6-9

7. Cambia la frecuencia de la señal de entrada y prueba con los siguientes valores: 300, 500, 800, 1000, 3000, 5000, 10,000 y 15,000. Observa la variación en amplitud y fase y genera una tabla donde reportes los valores de amplitud.
8. Cambiar la resistencia por 800, 400 y 200  $\Omega$  (valores de resistencia ideales). Observa la respuesta en frecuencia con el graficador de Bode.

### 3.6.2 Filtros Pasa Altas

El Filtro Pasa Altas permite el paso de las señales cuya frecuencia se encuentra por encima de la frecuencia de corte  $\omega_c$ . Tal como se hizo para los filtros pasa bajas, comenzemos por explicar cómo trabaja un filtro pasa altas pasivo revisando el circuito en la Figura 3-6.10

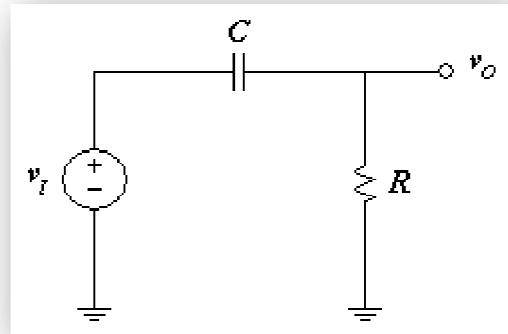


Figura 3-6.10

Cuando se hace pasar una señal de muy baja frecuencia por el circuito de la Figura 3-6.11, el capacitor se comporta como un circuito abierto y no permite el paso de la señal. En altas frecuencias, el capacitor se comporta como un cortocircuito permitiendo el paso de la señal. Para realizar el cálculo de la frecuencia de corte se utiliza exactamente la misma expresión utilizada en la (Ecuación 3.6-1).

El filtro pasa altas activo es ahora una mezcla entre el amplificador inversor y el derivador. Partiendo del filtro pasa bajas, solamente es necesario cambiar de lugar el capacitor y colocarlo en serie con la resistencia .

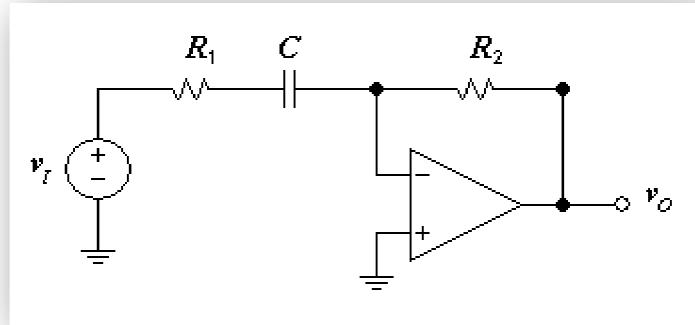


Figura 3-6.11

Siguiendo el análisis realizado para el filtro pasa bajas, el circuito se comporta como un derivador en bajas frecuencias y como un amplificador inversor en altas frecuencias.

La ganancia en la región pasa banda se calcula exactamente igual que en la (Ecuación 3.6-3) pero el cálculo de la frecuencia de corte se hace ahora con  $R_1$

$$f_1 = \frac{1}{R_1 C}$$

Podemos utilizar los mismos valores de componentes para la simulación.

## Simulación del Circuito

- Incorpora a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH
- Dos Resistencias de  $1.6\text{k}\Omega$
- Un Capacitor de  $100\text{nF}$
- Dos fuentes de voltaje de DC ( $15\text{V}$  y  $-15\text{V}$ )
- Generador de funciones
- Graficador de Bode
- Osciloscopio

- Conecta el circuito conforme a la Figura 3.6-11 y verifícalo con respecto al circuito mostrado en la Figura 3.6-12

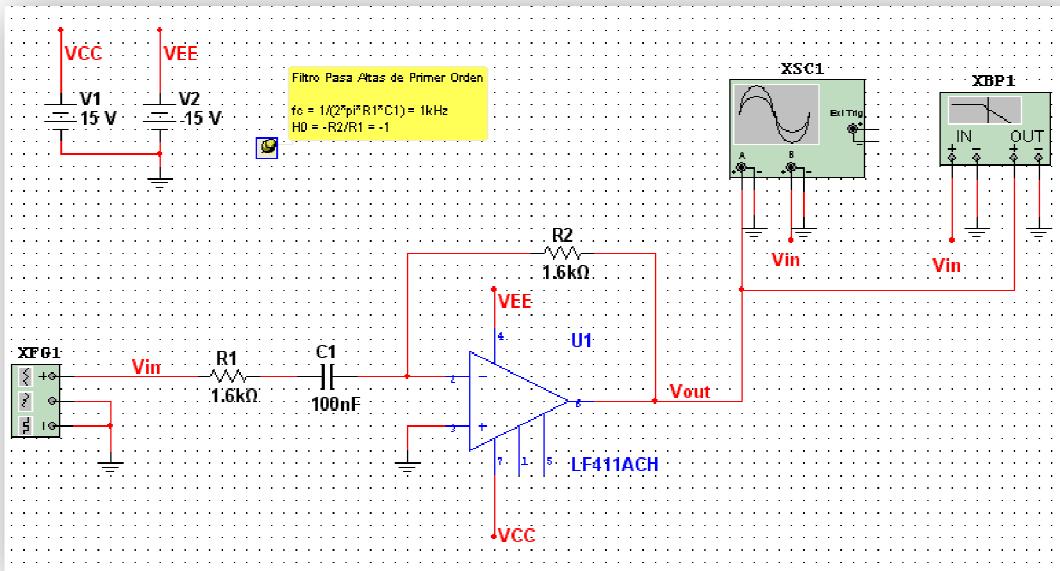


Figura 3.6-12

- Repite los pasos del 3 al 7 de la simulación del filtro pasa bajas.
- Cambia los valores de la resistencia por  $3.2$ ,  $6.4$  y  $1.28\text{k}\Omega$  (valores de resistencia ideales). Observa la respuesta en frecuencia con el graficador de Bode.

## Filtros Pasa Bandas

Si combinamos los circuitos del Filtro Pasa Bajas y del Filtro Pasa Altas revisados en las secciones anteriores podemos implementar un Filtro Pasa Bandas como el que se muestra en la Figura 3.6-13

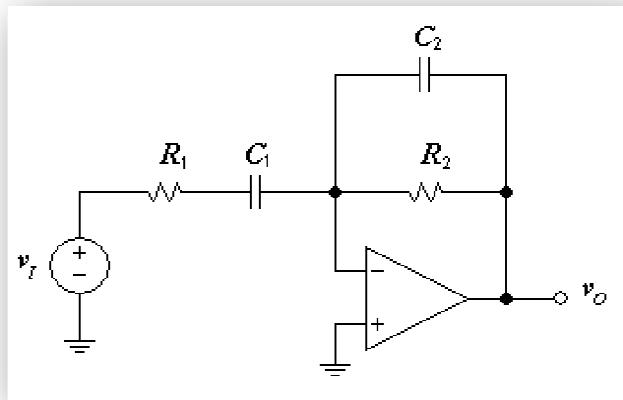


Figura 3.6-13

Este filtro tiene dos frecuencias de corte  $\omega_H$  (frecuencia de corte superior) y  $\omega_L$  (frecuencia de corte inferior). El cálculo de las frecuencias de corte es el siguiente:

$$f_L = \frac{1}{R_1 C_2} \quad f_H = \frac{1}{R_2 C_2} \quad (\text{Ecuación 3.6-5})$$

## Simulación del Circuito

1. Incorpora a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH
- Dos Resistencias de 16kΩ
- Un Capacitor de 100 nF
- Un Capacitor de 100 pF
- Dos fuentes de voltaje de DC (15 V y -15 V)
- Generador de funciones
- Graficador de Bode
- Osciloscopio

2. Conecta el circuito conforme a la Figura 3.6-14 y verifícalo con respecto al circuito mostrado en la Figura 3.6-15

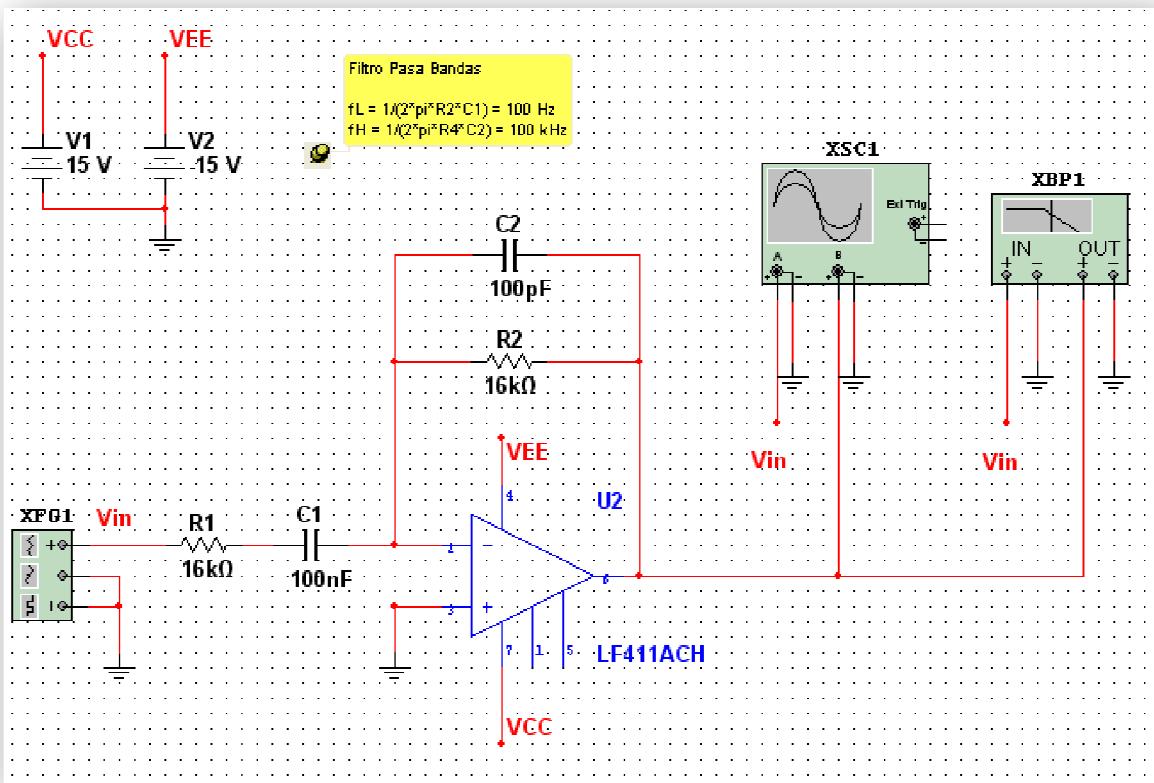
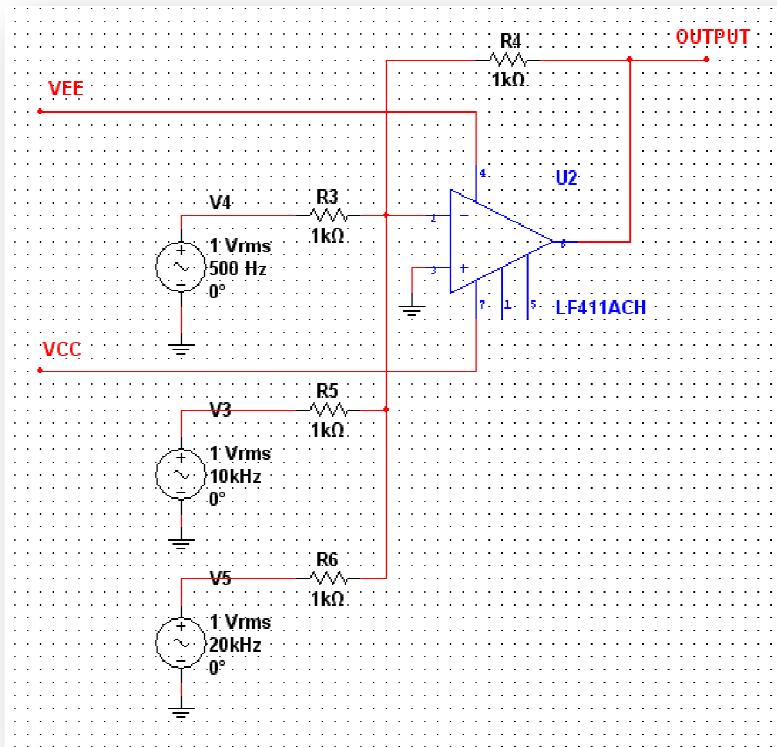


Figura 3.6-14

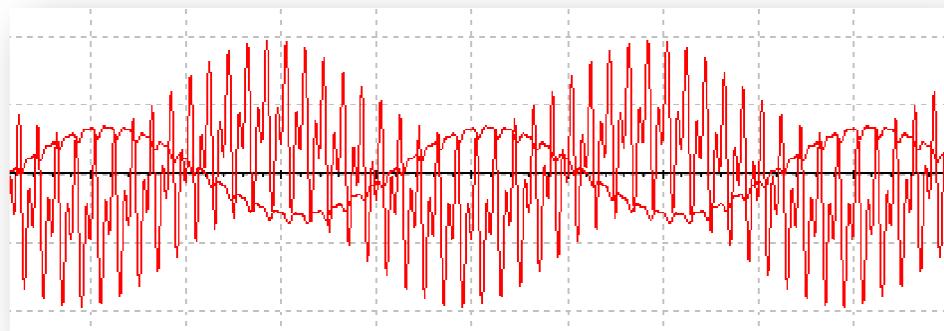
2. Repite los pasos del 3 al 7 de la simulación del filtro pasa bajas.

### Ejercicio (Opcional)

1. Implementa el siguiente circuito sumador en NI Multisim 10.1 (Figura 3.6-15)

**Figura 3.6-15**

2. Selecciona todo el circuito excepto las terminales de entrada y salida y presiona Ctrl+C para copiarlo. Ve a tu circuito del filtro pasa bajas, haz clic derecho y selecciona *Paste as Subcircuit*. Posteriormente utiliza este subcircuito para reemplazar el generador de funciones. Corrobora los siguientes resultados para el filtro pasa bajas:

**Figura 3.6-16**

3. Utiliza el subcircuito para probar tanto el filtro pasa altas como el filtro pasa bandas.

## Implementación del Circuito

Utiliza la plataforma de ELVIS II para armar los filtros pasa bajas, pasa altas y pasa bandas y verificar los resultados obtenidos durante la simulación.

## 3.7 Filtros de Orden Superior

Dentro de los mas destacados se encuentran los siguientes filtros de orden superior:

Butterworth, Chebyshev, Bessel, Elíptico.

The figure displays four subplots showing the Magnitude Response (dB) versus Frequency (Hz). Each plot has a y-axis ranging from -80.0 to 0.0 dB and an x-axis ranging from 0.0 to 800.0 Hz.

- Butterworth:** A smooth, monotonic roll-off starting at 0 dB at 0 Hz.
- Chebyshev:** Shows ripples in the passband (0 to 100 Hz) and a steeper roll-off than the Butterworth filter.
- Bessel:** Shows ripples in the passband (0 to 100 Hz) and a roll-off similar to the Butterworth filter.
- Elíptico:** Shows multiple ripples in the passband (0 to 100 Hz) and a very steep roll-off.

ni.com

NATIONAL INSTRUMENTS

## Introducción

Como hemos revisado en prácticas anteriores, podemos utilizar amplificadores operacionales para implementar filtros activos. Hasta el momento solo hemos implementado filtros activos de primer orden. En esta práctica implementaremos filtros de orden superior, comenzando por los filtros de segundo orden. Conforme vayamos incrementando el orden del filtro se irá incrementando la pendiente de atenuación del mismo, aunque implica el uso de más componentes activos.

Las distintas topologías de filtros de orden superior se implementan basadas en el concepto del factor de calidad Q. El factor Q describe que tan sub amortiguada es la respuesta de un oscilador. Pero, ¿por qué se habla de un filtro como un oscilador ó un sistema resonante? Pues justamente porque se diseñan a partir de capacitores e inductores formando circuitos resonantes RLC.

Entonces, ¿para qué sirve el factor Q en el diseño de un filtro, o de forma más genérica, en un oscilador? El factor Q nos sirve para saber que tan rápido se pierde la energía en el oscilador, que tan rápido muere dicha oscilación. La curva en la Figura 3.7-1 muestra gráficamente la descripción del factor de calidad:

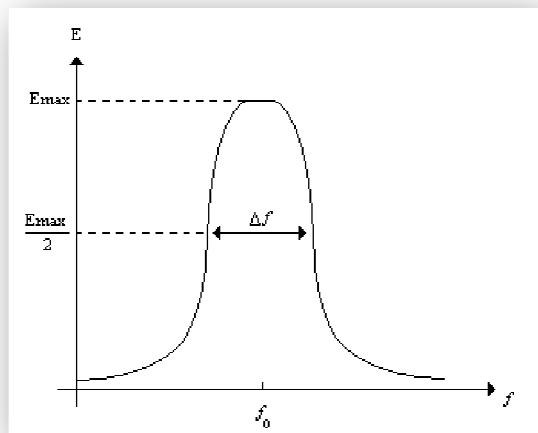


Figura 3.7-1

Mientras mayor sea el factor de calidad el ancho del oscilador será menor. Esto significa que si se desea que el oscilador retenga su energía, estableciendo un alto factor calidad  $Q$ , la frecuencia a la que se hace oscilar tiene que ser muy precisa.

En función al factor de calidad se pueden implementar múltiples topologías de filtros, dentro de las cuales destacan las que se muestran en la Figura 3.7-2 como unas de las más utilizadas.

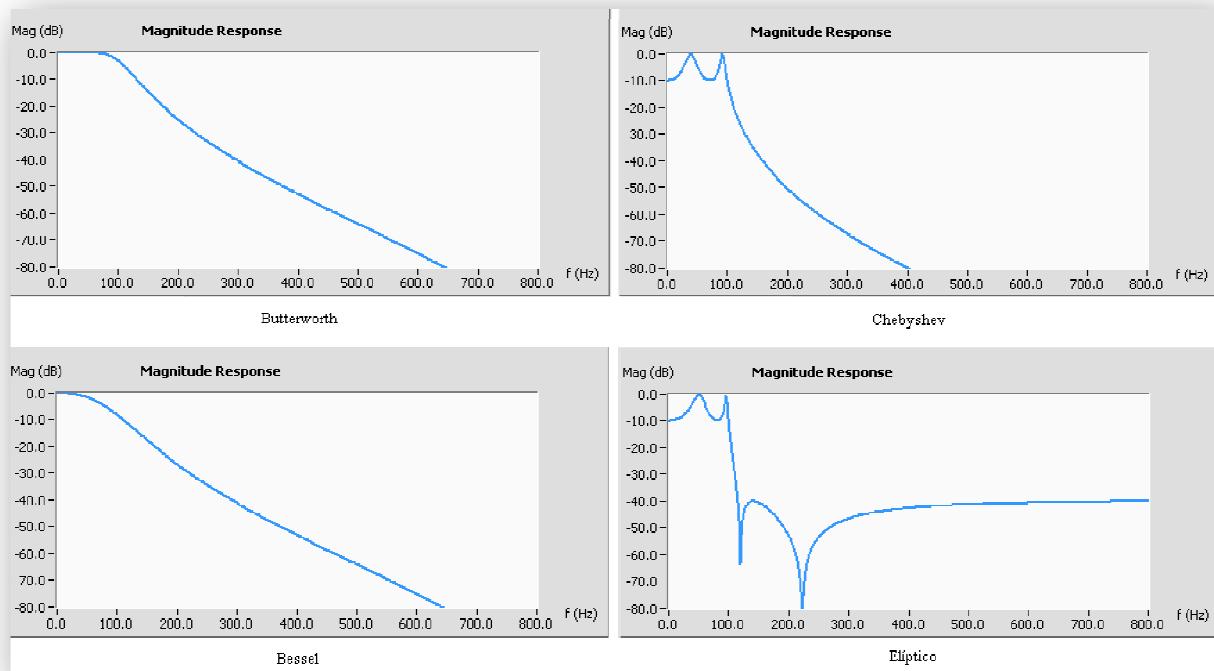


Figura 3.7-2

En esta práctica implementaremos filtros Chebyshev que se destacan por tener muy buena pendiente de atenuación. El problema de los filtros Chebyshev es que conforme se aumenta el orden, se incrementa el rizo en la región pasa banda. Si se desea tener una región pasa banda más estable se puede utilizar el Butterworth o el Bessel, aunque esto implica sacrificar pendiente de atenuación.

## Filtros de Segundo Orden

Una de las configuraciones más utilizadas en el diseño de filtros de segundo orden es la de los Filtros Sallen-Key de ganancia unitaria. Los filtros Sallen-Key normalizan la ganancia en la región pasa bandas para reducir el número de componentes y maximizar el ancho de banda del amplificador operacional.

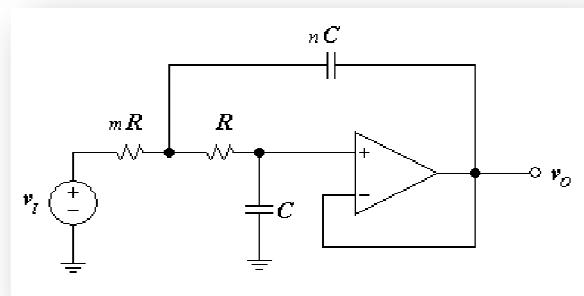


Figura 3.7-3

Esta configuración simplifica mucho el cálculo de los valores de los componentes del filtro. Las expresiones para realizar el cálculo de la ganancia, frecuencia de corte y factor de calidad son las siguientes.

$$H_{OLP} = 1V/V \quad \omega_0 = \frac{1}{\sqrt{mnRC}} \quad Q = \frac{\sqrt{mn}}{m+1} \quad (\text{Ecuación 3.7-1})$$

Implementemos ahora un filtro pasa bajas con frecuencia de corte de 995 Hz y factor de calidad Q de 8 (los valores de la frecuencia de corte y el factor de calidad tendrán sentido más adelante).

Hagamos  $n = 1000$  y sustituymos en la tercera de las ecuaciones de (Ecuación 3.7-1)

$$8 = \frac{\sqrt{1000m}}{m+1} \quad m^2 - 13.625m + 1 = 0$$

(Ecuación 3.7-2)

Despejando m obtenemos que una de las soluciones es 13.551.

Si hacemos  $C = 1 \text{ nF}$ , sustituimos todos los valores en la segunda de las ecuaciones de (Ecuación 3.7-1) y despejamos  $R$  obtenemos lo siguiente:

$$R = \frac{1}{2\pi\sqrt{(13.551)(1000)(995)(1 * 10^{-9})}} = 1.3374 \text{ k}\Omega \quad (\text{Ecuación 3.7-3})$$

Haciendo los cálculos de  $mR$  y  $nC$  y sustituyendo los valores de las resistencias por los valores comerciales a 1% de tolerancia obtenemos los siguientes valores de los componentes:

$$R = 1.37 \text{ k}\Omega$$

$$mR = 18.7 \text{ k}\Omega$$

$$C = 1 \text{ nF}$$

$$nC = 1 \mu\text{F}$$

## Simulación del Circuito

1. Incorpora a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH
- Resistencias de  $1.37 \text{ k}\Omega$  y  $18.7 \text{ k}\Omega$
- Capacitores de  $1 \text{ nF}$  y  $1\mu\text{F}$
- Dos fuentes de voltaje de DC ( $15 \text{ V}$  y  $-15 \text{ V}$ )
- Generador de funciones
- Graficador de Bode

2. Conecta los componentes conforme a la Figura 3.7-3 y verifícalos con respecto a la Figura 3.7-4

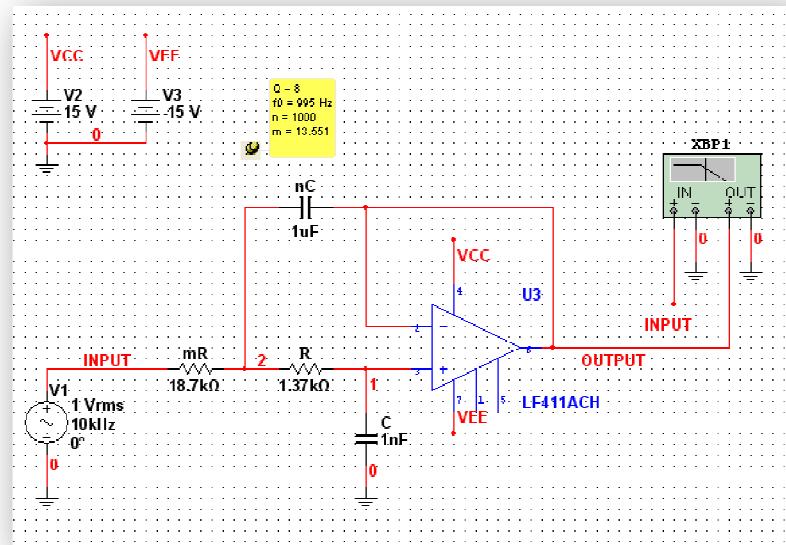


Figura 3.7-4

3. Corre la simulación y da doble clic al Graficador de Bode. Observa que el comportamiento es muy similar al Chebyshev mostrado en la Figura 3.7-2 de la parte introductoria. Si ajustas los rangos de los ejes y mueves el cursor a la frecuencia más cercana a 995 Hz observarás que se encuentra en el punto más alto, justamente antes de que comience a atenuarse la ganancia.

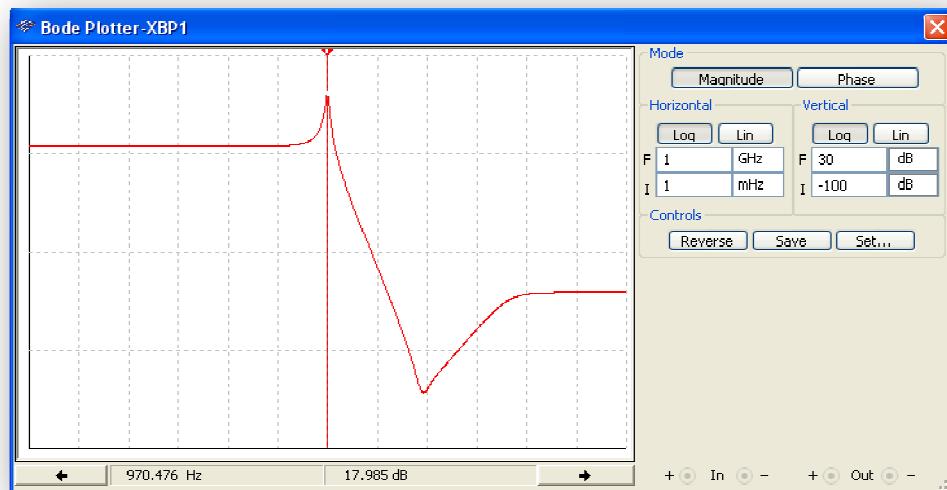


Figura 3.7-5

## Filtros de Orden Superior (Orden 6)

Para implementar filtros de orden superior puedes concatenar etapas de segundo orden Sallen-Key. Por ejemplo, para implementar un filtro de sexto orden puedes concatenar tres filtros Sallen-Key. Existen tablas que toman como variables la frecuencia de corte y el factor de calidad para implementar filtros de orden superior conforme a las distintas topologías (Butterworth, Chebyshev, Bessel, etc.). A continuación se muestra la tabla para los filtros Chebyshev:

Filtro Chebyshev Pasa Bajas con Rizo de 1.0 – dB.

<b>n</b>	<b>f<sub>01</sub></b>	<b>Q<sub>1</sub></b>	<b>f<sub>02</sub></b>	<b>Q<sub>2</sub></b>	<b>f<sub>03</sub></b>	<b>Q<sub>3</sub></b>	<b>f<sub>04</sub></b>	<b>Q<sub>4</sub></b>	<b>f<sub>05</sub></b>	<b>Q<sub>5</sub></b>	<b>Att(dB ) at 2f<sub>c</sub></b>
<b>2</b>	1.050	0.957									11.36
<b>3</b>	0.997	2.018	0.494								22.46
<b>4</b>	0.993	3.559	0.529	0.785							33.87
<b>5</b>	0.994	5.556	0.655	1.399	0.289						45.31
<b>6</b>	0.995	8.004	0.747	2.198	0.353	0.761					56.74
<b>7</b>	0.996	10.899	0.808	3.156	0.480	1.297	0.205				68.18
<b>8</b>	0.997	14.240	0.851	4.266	0.584	1.956	0.265	0.753			79.62
<b>9</b>	0.998	18.029	0.881	5.257	0.662	2.713	0.377	1.260	0.159		91.06
<b>10</b>	0.998	22.263	0.902	6.937	0.721	3.561	0.476	1.864	0.212	0.749	102.50

Tabla 3.7-1

Entonces, para el filtro que vamos a implementar tomaremos los valores correspondientes a  $n = 6$ . La frecuencia de corte de este filtro será de 1 kHz, por lo que nuestras etapas de segundo orden serán diseñadas en base a los siguientes valores:

$$\begin{array}{ll} f_{01} = 0.995f_c = 995 \text{ Hz} & Q_1 = 8.00 \\ f_{02} = 0.747f_c = 747 \text{ Hz} & Q_2 = 2.20 \\ f_{03} = 0.353f_c = 353 \text{ Hz} & Q_3 = 0.761 \end{array}$$

Cada una de las etapas consiste de un filtro pasa bajas con una determinada frecuencia de corte y un determinado factor de calidad. Al sumar las curvas de Bode queda una curva con una pendiente de atenuación de 146 dB por década.

Al revisar los valores nos damos cuenta que la primera etapa ya la tenemos implementada. Ahora solamente resta implementar las otras dos etapas.

## Etapa Dos

Para esta etapa tomaremos los valores de  $y$  y haremos  $n = 100$  y  $C = 100 \text{ nF}$ .

Sustituyendo nuevamente en las ecuaciones de (Ecuación 3.7-1) y buscando los valores de resistencia comercial al 1 % de tolerancia más cercanos obtenemos que:

$$m = 18.66$$

$$R = 51 \Omega$$

$$mR = 909 \Omega$$

$$C = 100 \text{ nF}$$

$$nC = 10 \mu\text{F}$$

## Simulación del Circuito

1. Incorpora a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH
- Resistencias de  $51 \Omega$  y  $909 \Omega$
- Capacitores de  $100 \text{ nF}$  y  $10 \mu\text{F}$
- Dos fuentes de voltaje de DC ( $15 \text{ V}$  y  $-15 \text{ V}$ )
- Generador de funciones
- Graficador de Bode

2. Conecta los componentes conforme a la Figura 3.7-3 y verifícalos con respecto a la Figura 2.1

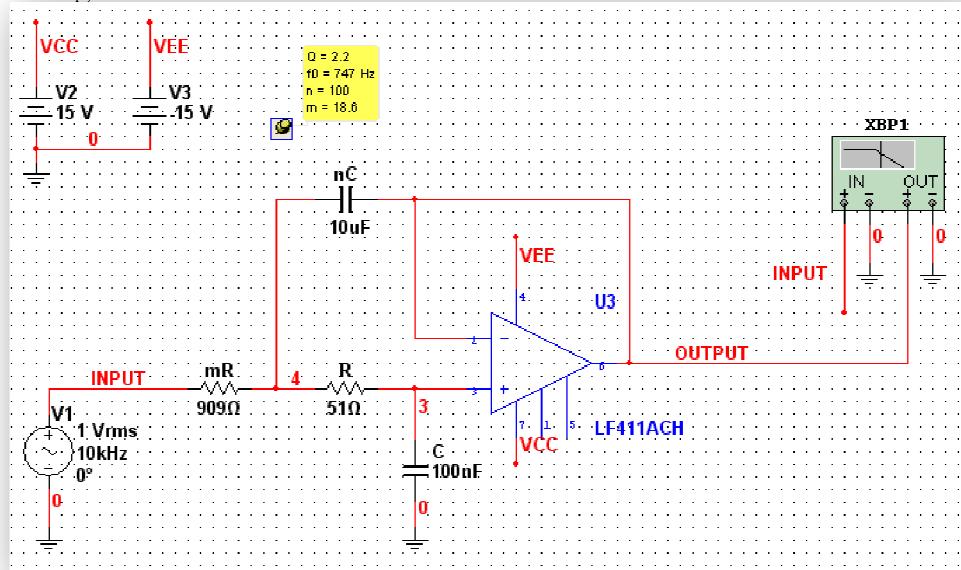


Figura 3.7-6

3. Corre la simulación y da doble clic al Graficador de Bode. Observa ahora que la frecuencia de corte está muy cercana a 747 Hz.

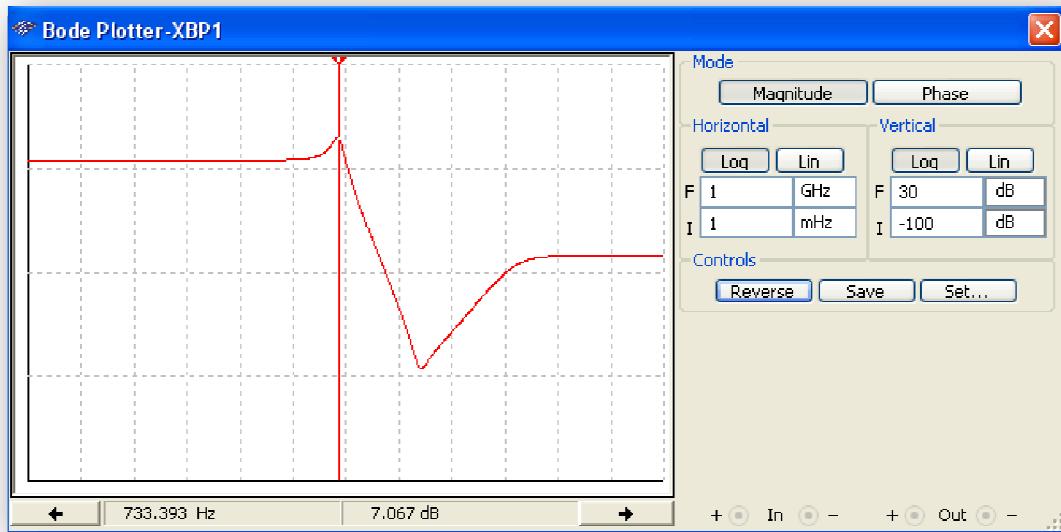


Figura 3.7-7

## Etapa Tres

Para esta etapa tomaremos los valores de  $\gamma$  y  $\beta$  y haremos  $n = 10$  y  $C = 100 \text{ nF}$ .

Sustituyendo nuevamente en las ecuaciones de (Ecuación 3.7-1) y buscando los valores de resistencia comercial al 1 % de tolerancia más cercanos obtenemos que:

$$m = 15.2$$

$$R = 365 \Omega$$

$$mR = 5.62 \text{ k}\Omega$$

$$C = 100 \text{ nF}$$

$$nC = 1 \mu\text{F}$$

## Simulación del Circuito

1. Incorpora a NI Multisim 10.1 los siguientes elementos:

- Amplificador Operacional LF411ACH
- Resistencias de  $365 \Omega$  y  $5.62 \text{ k}\Omega$
- Capacitores de  $100 \text{ nF}$  y  $1 \mu\text{F}$
- Dos fuentes de voltaje de DC ( $15 \text{ V}$  y  $-15 \text{ V}$ )
- Generador de funciones
- Graficador de Bode

2. Conecta los componentes conforme a la Figura 3.7-3 y verifícalos con respecto a la Figura 3.7-8

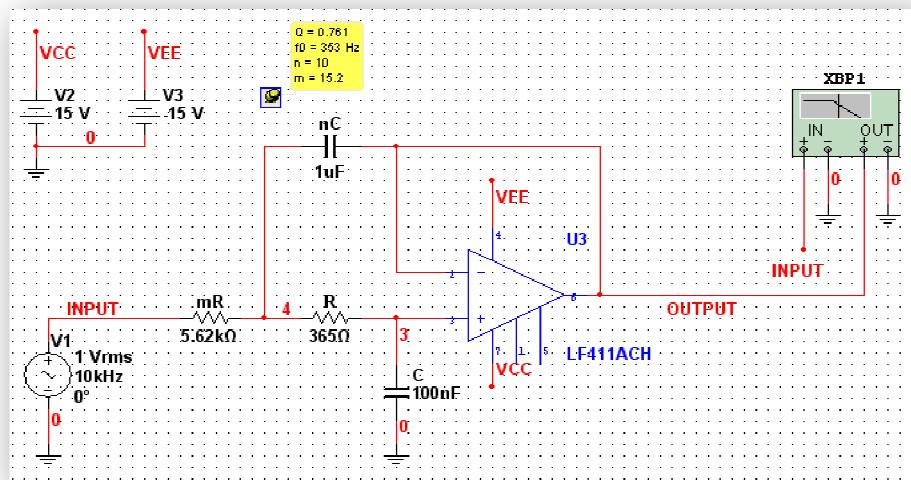


Figura 3.7-8

Corre la simulación y da doble clic al Graficador de Bode. Observa ahora que la frecuencia de corte está muy cercana a 747 Hz.

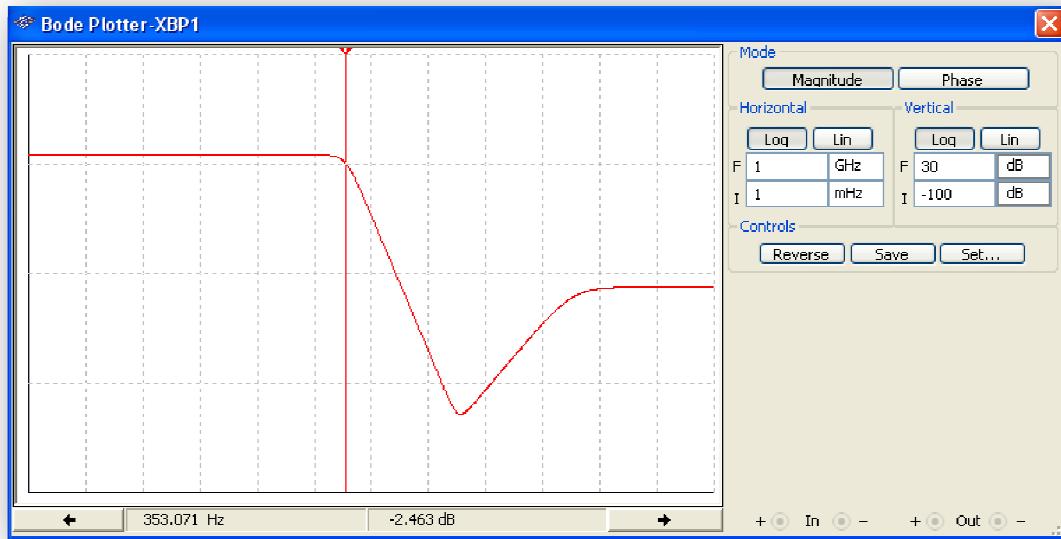


Figura 3.7-9

## Filtro Pasa Bajas Orden 6

Para implementar el filtro Pasa Bajas Orden 6 tenemos que concatenar las etapas anteriores en orden inverso. El circuito debe quedar como se muestra en la Figura 3.7-10

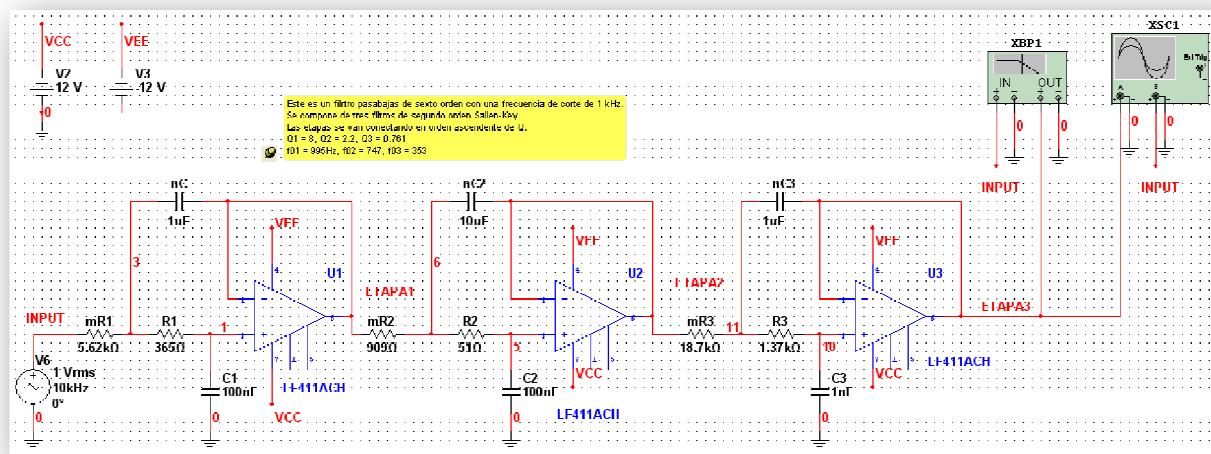
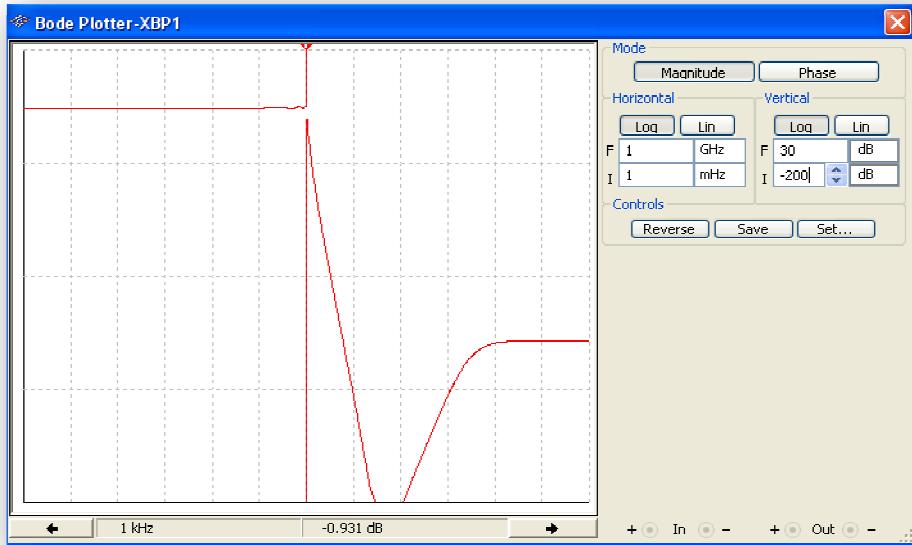


Figura 3.7-10

La curva del graficador de Bode se debe ver como en la Figura 3.7-11

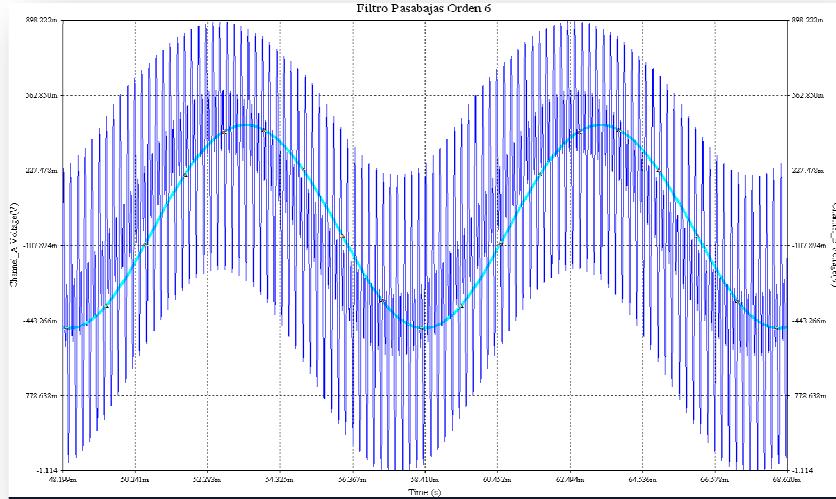


**Figura 3.7-11**

### Ejercicio (Opcional)

Implementar un sumador como subcircuito para probar el filtro tal como se hizo en la práctica de filtros de primer orden.

La señal de entrada es una suma de senoidales de amplitud 1 Vrms con frecuencias de 100 Hz, 5 kHz, 10 kHz.



**Figura 3.7-12**

### Reto

Implementar el Filtro Pasa Bajas de Orden 6 en la plataforma de ELVIS II y corroborar los resultados encontrados durante la simulación

## Notas

---

## Notas

---

# Clase: Electrónica Digital

ni.com



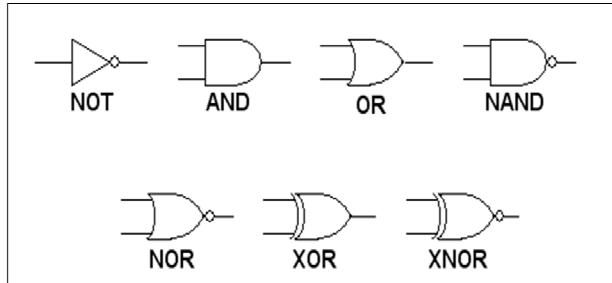
144

## Temario de la clase

- 4.1 Comprobación de las tablas de verdad.
- 4.2 Medio Sumador
- 4.3 Sumador completo
- 4.4 Medio Restador
- 4.5 Circuitos combinacionales.
- 4.6 Mapa de Karnaugh para 5 variables.
- 4.7 Ejercicio propuesto 1
- 4.8 Ejercicio propuesto 2
- 4.9 Proyecto Propuesto: Decodificador n a m.

## 4.1 Comprobación de las tablas de verdad.

Comprobación de las tablas de verdad de las compuertas lógicas: NOT, AND, OR, NAND, NOR, XOR, XNOR.



ni.com



145

### Definición del Problema.

Comprobar las tablas de verdad para las compuertas: NOT, AND, OR, NAND, NOR, XOR Y XNOR mediante un circuito básico en NI Multisim 10.1.

### Objetivo.

Que el alumno conozca el entorno de NI Multisim 10.1, el concepto de las Tablas de Verdad y aprenda el funcionamiento básico de las compuertas lógicas.

### Escenario.

Considere la implementación de las Tablas de Verdad de dos variables para las compuertas: AND, OR, NAND, NOR, XOR, y XNOR, y de una sola variable para la compuerta: NOT.

## Diseño

Para probar las compuertas lógicas, este ejercicio deberá apoyarse en las siguientes Tablas de Verdad:

a	NOT
0	1
1	0

**Tabla 4-1** Compuerta NOT

a	b	AND	OR	NAND	NOR	XOR	XNOR
0	0	0	0	1	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	1	0	0	0	1

**Tabla 4-2** Compuertas Lógicas

## Implementación.

El alumno verificará el comportamiento de la compuerta OR en NI Multisim 10.1.

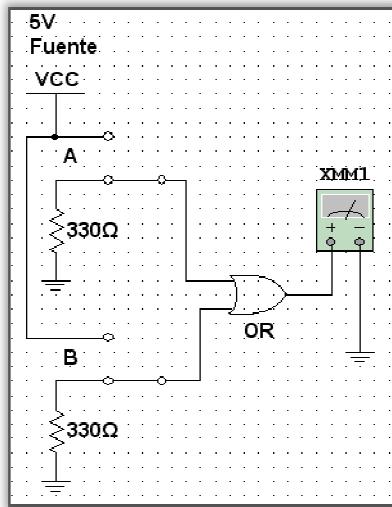
1. Abra NI Multisim 10.1

- De clic sobre la barra del Escritorio en el botón de Inicio.
- Diríjase a **Todos los Programas » National Instruments » Circuit Design Suite 10.1 » NI Multisim 10.1 10.1**

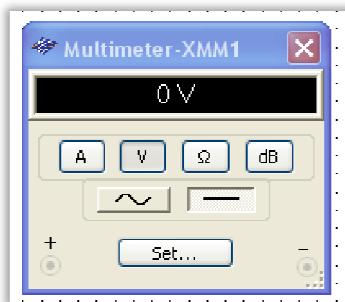
2. Incorpore a NI Multisim 10.1 los siguientes elementos:

- 1 compuerta lógica OR de dos entradas
- 1 Fuente de alimentación de 5V
- 3 Tierras
- 2 Resistencias de 330 Ω
- 1 Multímetro (Se encuentra en la Paleta de *Instruments*)
- 2 *Switch*

3. Conecte el circuito conforme a la Figura 4-1

**Figura 4-1** Conexión del circuito**Prueba.**

1. Ejecute la Simulación
  - De clic en el botón de **Run** localizado en la paleta de Simulación.
2. Compare el resultado obtenido con la Tabla de Verdad.
  - De doble clic sobre el Multímetro. Podrá visualizar el resultado en esta ventana de la Figura 4-2 y compararlo con la tabla de verdad, mientras realiza los cambios en los *Switch* de entrada.

**Figura 4-2** Ventana del Multímetro

**Fin del ejercicio.**

**Reto.**

Que el alumno desarrolle los pasos anteriores para probar cada una de las Compuertas Lógicas restantes, observar su comportamiento y comparar sus respuestas con las Tablas de Verdad.

## 4.2 Medio Sumador

Del *medio sumador* compruebe la tabla y las funciones correspondientes a este:

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = \bar{x}y + x\bar{y}$$

$$S = x \oplus y$$

$$C = x + y$$

ni.com

148



### Definición del Problema.

Sumador.

Entre las funciones básicas encontradas, la operación mas básica es sin duda la suma de dos dígitos binarios. Esta simple operación consiste en cuatro operaciones elementales:

$$0 + 0 = 0 \quad 0 + 1 = 1 \quad 1 + 0 = 1 \quad 1 + 1 = 10$$

El circuito combinacional que realiza la suma de dos bits se llama medio sumador.

### Objetivo.

Que el alumno conozca más sobre el entorno de NI Multisim 10.1, así como el concepto de Medio Sumador y lo ponga a prueba.

### Escenario.

Considere la Tabla de Verdad de dos variables del Medio Sumador para llevar a cabo este ejercicio (Tabla 4-3).

## Diseño

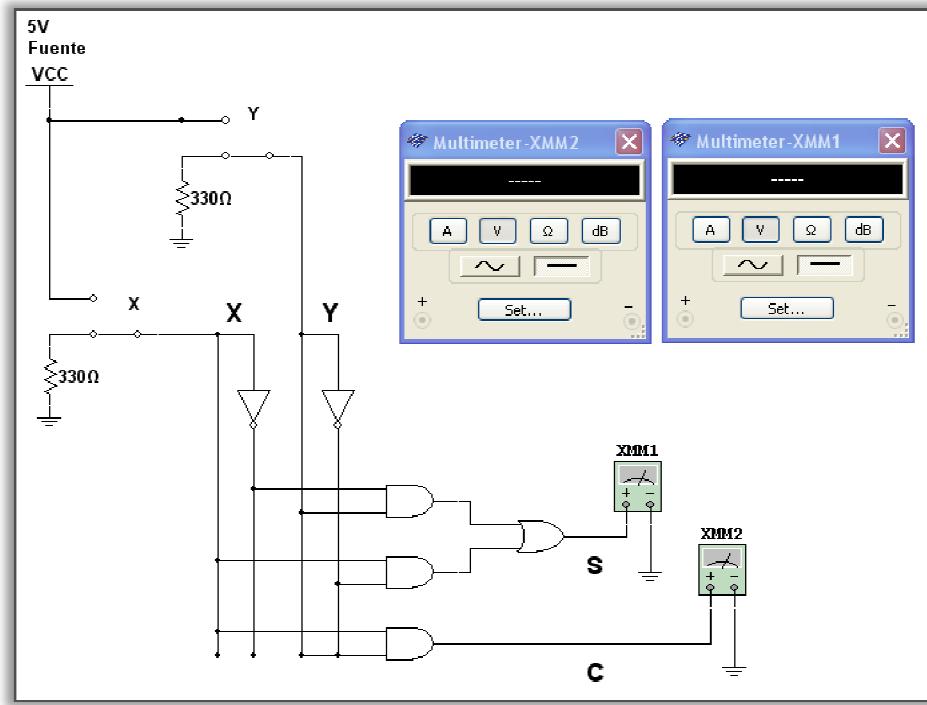
Para llevar a cabo las pruebas, puede apoyarse en la siguiente Tabla de Verdad (Tabla 4-3), la cual representa el Medio Sumador con dos variables de entrada.

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

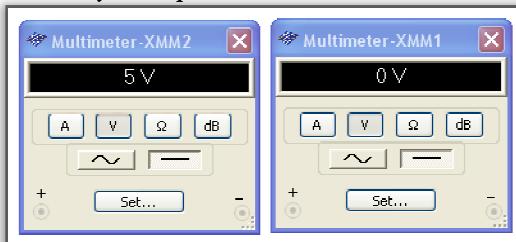
**Tabla 4-3** Tabla de verdad del Medio Sumador

## Implementación.

1. Abra NI Multisim 10.1 10.1
  - De clic sobre la barra del Escritorio en el botón de Inicio.
  - Diríjase a **Todos los Programas » National Instruments » Circuit Design Suite 10.1 » NI Multisim 10.1**
2. Incorpore a NI Multisim 10.1 los siguientes elementos:
  - 3 compuertas lógicas AND de dos entradas
  - 1 compuerta lógica OR de dos entradas
  - 2 compuertas lógicas NOT
  - 2 *Switch*
  - 1 Fuente de alimentación de 5V
  - 3 Tierras
  - 2 Resistencias de 330 Ω
  - 2 Multímetro (Se encuentra en la Paleta de *Instruments*)
3. Conecte el circuito conforme a la Figura 4-3

**Figura 4-3 Conexión del Medio Sumador****Prueba.**

1. Ejecute la Simulación
  - De clic en el botón de **Run** localizado en la paleta de Simulación.
2. Compare el resultado obtenido con la Tabla de Verdad.
  - De doble clic sobre los Multímetros. Ahora manipule los switches de entrada y compare los resultados con los de la Tabla de Verdad.

**Figura 4-4 Ventana de los Multímetros****Fin del ejercicio.****Reto.**

Que el alumno investigue y manipule otras formas de representar el medio sumador .

## 4.3 Sumador completo

### Sumador completo

x	y	z	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

ni.com

151



### Definición del Problema.

El circuito combinacional que realiza la suma de tres bits es un sumador completo. El nombre se deriva del hecho de que se utilizan dos medios sumadores para hacerlo. Este consiste en tres entradas y dos salidas.

Cada función de Boole de salida requiere un Mapa de Karnaugh único para su simplificación. Cada mapa tiene 8 cuadrados ya que cada salida es una función de las tres variables de entrada. Los siguientes mapas los utilizamos para simplificar las dos funciones de salida.

s

x	y	z	0	1
0	0	0	1	
0	0	1		1
0	1	0		1
0	1	1	1	

c

x	y	z	0	1
0	0	0		
0	0	1		1
0	1	0	1	
0	1	1	1	1

$$S = x'y'z + x'yz' + xy'z' + xyz$$

$$C = x'yz + xy'z + xyz' + xyz$$

$$C = xy + xz + yz$$

### Objetivo.

Que el alumno conozca más sobre el entorno de NI Multisim 10.1, así como el concepto del Sumador Completo y lo ponga a prueba.

## Escenario.

Considere la Tabla de Verdad de tres variables del Sumador Completo para llevar a cabo este ejercicio (Tabla 4-4).

## Diseño

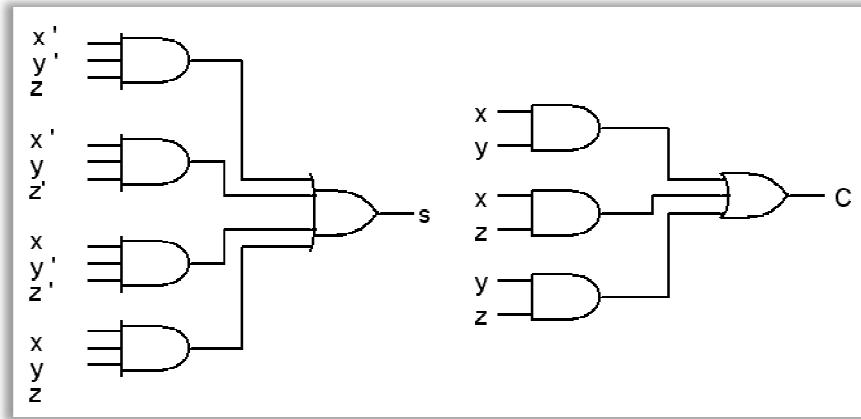
Para llevar a cabo las pruebas, puede apoyarse en la siguiente Tabla de Verdad (Tabla 4-4), la cual representa el Sumador Completo y sus tres variables de entrada.

<b>x</b>	<b>y</b>	<b>z</b>	<b>C</b>	<b>S</b>
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

**Tabla 4-4** Tabla de verdad del Sumador Completo

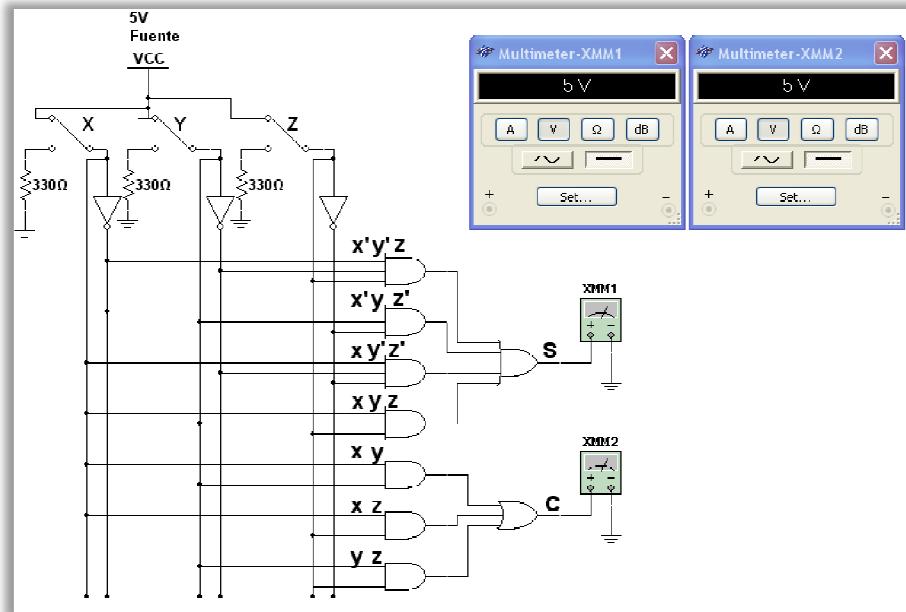
## Implementación.

1. Abra NI Multisim 10.1
2. Incorpore a NI Multisim 10.1 los siguientes elementos:
  - 3 compuertas lógicas AND de dos entradas
  - 1 compuerta lógica OR de dos entradas
  - 2 compuertas lógicas NOT
  - 2 *Switch*
  - 1 Fuente de alimentación de 5V
  - 3 Tierras
  - 2 Resistencias de 330 Ω
  - 2 Multímetro (Se encuentra en la Paleta de *Instruments*)
3. Conecte el circuito conforme a la Figura 4-3

**Figura 4-4** Configuración de un Sumador Completo**Prueba.**

- Ejecute la Simulación

□ De clic en el botón de **Run** localizado en la paleta de Simulación.

**Figura 4-5** Simulación de un Sumador Completo

- Compare el resultado obtenido con la Tabla de Verdad.

**Fin del ejercicio.**

**Reto.**

Implementar un sumador completo utilizando dos medios sumadores, fundamentar la solución con aritmética booleana.

## 4.4 Medio Restador

Del *medio restador* compruebe la tabla y las funciones correspondientes a este:

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$B = x'y$$

$$D = x'y + xy'$$

ni.com



154

### Definición del Problema.

Restador.

Es un circuito combinacional que resta dos bits y produce su diferencia. Tiene una salida que especifica si se ha prestado un 1.

La salida prestada B es 0 siempre y cuando  $X \geq Y$ . Será 1 para  $x=0, y=1$ . La salida D es el resultado de la operación aritmética  $2B+x-y$ .

Las funciones de Boole son:

$$B = x'y \quad D = x'y + xy' \quad (\text{Ecuacion 4.4-1})$$

### Objetivo.

Que el alumno conozca más sobre el entorno de NI Multisim 10.1, así como el concepto de Medio Restador y lo ponga a prueba.

### Escenario.

Considere la Tabla de Verdad de dos variables del Medio Restador para llevar a cabo este ejercicio (Tabla 4-5).

## Diseño

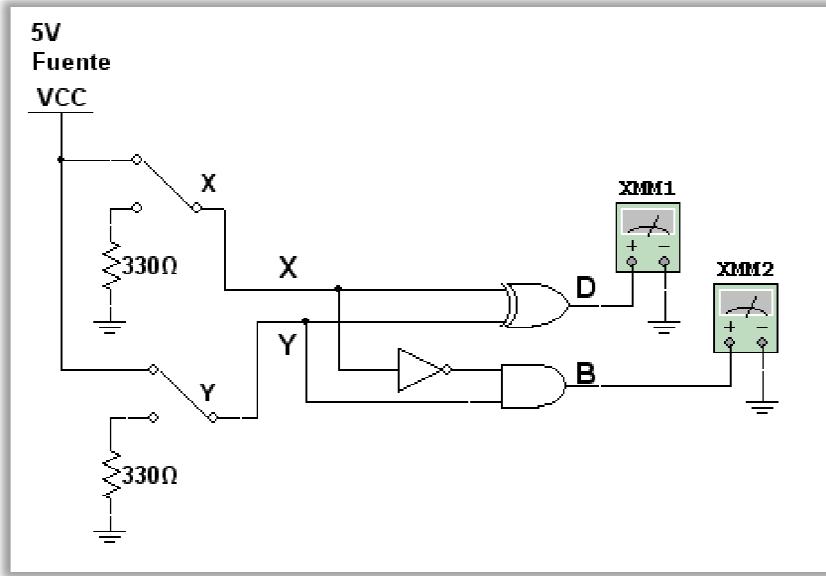
Para llevar a cabo las pruebas, puede apoyarse en la siguiente Tabla de Verdad (Tabla 4-5), la cual representa el Medio Restador con dos variables de entrada.

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

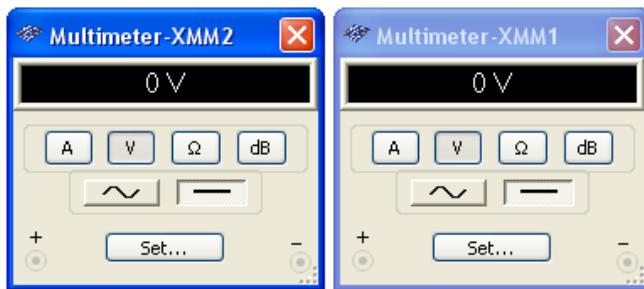
**Tabla 4-5** Tabla de verdad del Medio Restador

## Implementación.

1. Abra NI Multisim 10.1
  
2. Incorpore a NI Multisim 10.1 los siguientes elementos:
  - 1 compuerta lógica XOR de dos entradas
  - 1 compuerta lógica AND de dos entradas
  - 1 compuerta lógica NOT
  - 2 *Switch*
  - 1 Fuente de alimentación de 5V
  - 4 Tierras
  - 2 Resistencias de 330 Ω
  - 2 Multímetros (Se encuentra en la Paleta de *Instruments*)
  
3. Conecte el circuito conforme a la Figura 4-6

**Figura 4-6 Conexión del Medio Restador****Prueba.**

1. Ejecute la Simulación
  - De clic en el botón de **Run** localizado en la paleta de Simulación.
2. Compare el resultado obtenido con la Tabla de Verdad.
  - De doble clic sobre los Multímetros. Ahora manipule los switches de entrada y compare los resultados con los de la Tabla de Verdad.

**Figura 4-7 Ventana de Simulación para X y Y en alto.****Fin del ejercicio.****Reto.**

Que el alumno implemente y manipule la simulación de un restador completo.

## 4.5 Circuitos combinacionales.

- Obtener las funciones para reducir las funciones mediante mapas de Karnaugh a su mínima expresión.
- Construir los circuitos digitales.

a	b	c	d			
0	0	0	0	0	1	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	1	0	0
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	1	1	0
1	0	1	0	0	0	1
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

ni.com



157

### Definición del Problema.

Un Circuito Combinacional es un circuito donde su salida depende de la combinación de sus entradas. Los Circuitos Combinacionales están hechos a partir de las compuertas lógicas.

Para comprender mejor estos circuitos combinacionales se hace uso de Algebra Booleana, Mapas de Karnaugh y Tablas de Verdad.

### Objetivo.

Que el alumno conozca más sobre el entorno de NI Multisim 10.1, y manipule y aprenda más sobre Circuitos Combinacionales

### Escenario.

Considere la Tabla de Verdad de cuatro variables para llevar a cabo este ejercicio (Tabla 4-6).

## Diseño

Para llevar a cabo este ejercicio, puede apoyarse en la siguiente Tabla de Verdad (Tabla 4-6).

a	b	c	d			
0	0	0	0	0	1	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	1	0	0
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	1	1	0
1	0	1	0	0	0	1
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0

**Tabla 4-6** Tabla de verdad del Circuito Combinacional.

## Implementación.

- El alumno deberá obtener las funciones de la Tabla 5-6.

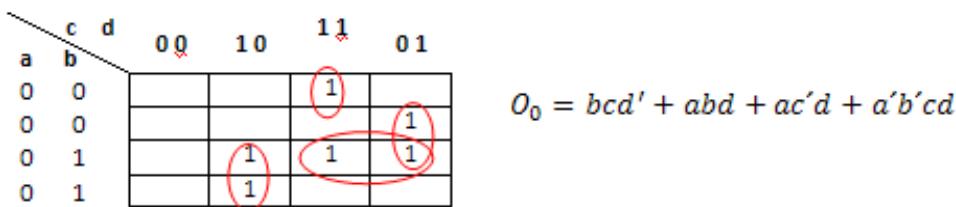
$$O_0 = a'b'cd + a'bcd' + ab'c'd + abc'd + abcd' + abcd$$

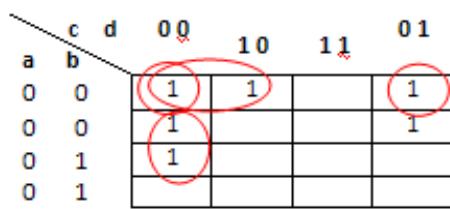
$$O_1 = a'b'c'd' + a'b'c'd + a'b'cd' + ab'c'd' + ab'c'd + abc'd'$$

$$O_2 = ab'cd + ab'cd' + ab'c'd' + ab'cd' + ab'cd$$

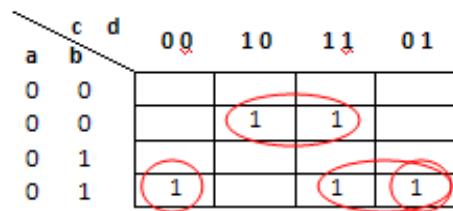
- Construir los mapas de Karnaugh en base a las ecuaciones anteriores.

### Mapa de Karnaugh de $O_0$



**Mapa de Karnaugh de  $O_1$** 

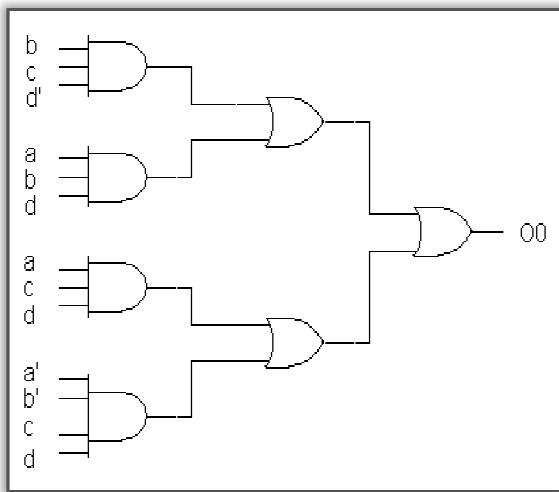
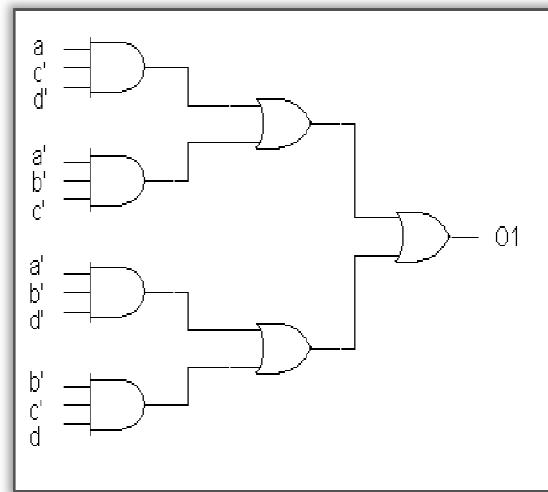
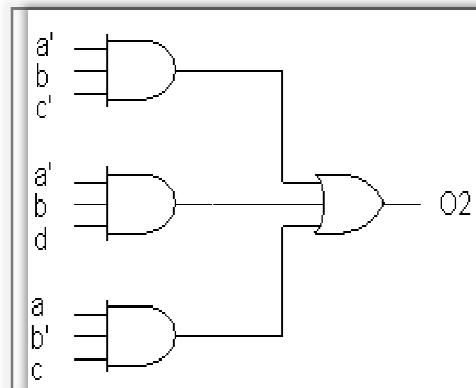
$$O_1 = ac'd' + a'b'c' + a'b'd' + b'c'd$$

**Mapa de Karnaugh de  $O_2$** 

$$O_2 = a'b'c' + a'bd + ab'c$$

3. En base a las funciones obtenidas, lleve a cabo los circuitos correspondientes en NI Multisim 10.1

Los diagramas de los circuitos que se obtuvieron en base a las funciones anteriores los puede observar en las Figuras 4-8, 4-9 y 4-10

**Figura 4-8 Circuito de O0****Figura 4-9 Circuito de O1****Figura 4-10 Circuito de O2**

## Prueba.

1. Configure los circuitos para poder llevar a cabo la simulación.
2. Ejecute la Simulación.
  - De clic en el botón de **Run** localizado en la paleta de Simulación.
3. Compare los resultados obtenidos con la Tabla de Verdad.

**Fin del ejercicio.**

## Reto.

Conecta los circuitos  $O_0$ ,  $O_1$ , y  $O_2$  a compuertas NAND.

**Nota:** Recuerde las leyes de Morgan,

$$(\overline{P + Q}) = \overline{P}\overline{Q} \quad \text{y} \quad \overline{PQ} = \overline{P} + \overline{Q} \quad (\text{Ecuacion 4.5-1})$$

Justifique su respuesta.

## 4.6 Mapa de Karnaugh para 5 variables.

a	b	c	d	e	
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	1	1
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	X
0	1	1	0	1	X
0	1	1	1	0	1
0	1	1	1	1	1

a	b	c	d	e	
1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	X
1	1	1	0	1	X
1	1	1	1	0	1
1	1	1	1	1	1

ni.com



161

### Definición del Problema.

El método de los Mapas de Karnaugh ofrece un procedimiento simple y directo para minimizar las funciones de Boole. Un Mapa de Karnaugh consiste en una serie de recuadros, donde cada uno representa una línea de la Tabla de Verdad. La Tabla de verdad de una función de N variables posee  $2^n$  filas, el Mapa de Karnaugh posee  $2^n$  cuadros.

### Objetivo.

Que el alumno explore el entorno de NI Multisim 10.1, el concepto de las Tablas de Verdad y Mapas de Karnaugh.

### Escenario.

Considere la Tabla de Verdad de cinco variables para llevar a cabo este ejercicio (Tabla 4-6). Obtenga los mapas de Karnaugh y las Tablas de Verdad.

## Diseño

Apóyese en la siguiente Tabla de Verdad para obtener los mapas

a	b	c	d	e			
0	0	0	0	0	1	1	
0	0	0	0	1	1	0	
0	0	0	1	0	0	0	
0	0	0	1	1	0	0	
0	0	1	0	0	1	1	
0	0	1	0	1	1	0	
0	0	1	1	0	1	0	
0	1	0	0	0	1	0	
0	1	0	0	1	0	1	
0	1	0	1	0	0	1	
0	1	1	0	0	0	X	
0	1	1	0	1	0	X	
0	1	1	1	0	1	1	
0	1	1	1	1	1	1	

a	b	c	d	e			
1	0	0	0	0	1	0	
1	0	0	0	1	1	1	
1	0	0	1	0	0	1	
1	0	0	1	1	1	0	
1	0	1	0	0	1	0	
1	0	1	0	1	1	1	
1	0	1	1	0	0	0	
1	0	1	1	1	1	0	
1	1	0	0	0	0	1	
1	1	0	0	1	0	0	
1	1	0	1	0	0	1	
1	1	0	1	1	0	1	
1	1	1	0	0	0	X	
1	1	1	0	1	0	X	
1	1	1	1	0	1	1	
1	1	1	1	1	1	1	

ve

ve

## Implementación.

- Construir los mapas de Karnaugh en base a las Tablas de Verdad anteriores.

Mapa de Karnaugh de  $Y_0$

Para  $a=0$

	d	e	00	10	11	01
b	c					
0	0	1	1			1
0	0		1			
0	1			1	1	
0	1		1	1		1

Para  $a=1$

	d	e	00	10	11	01
b	c					
0	0	1	1			
0	0			1		
0	0				1	1
0	1			1	1	
0	1		1		1	1

Mapa de Karnaugh de  $Y_1$

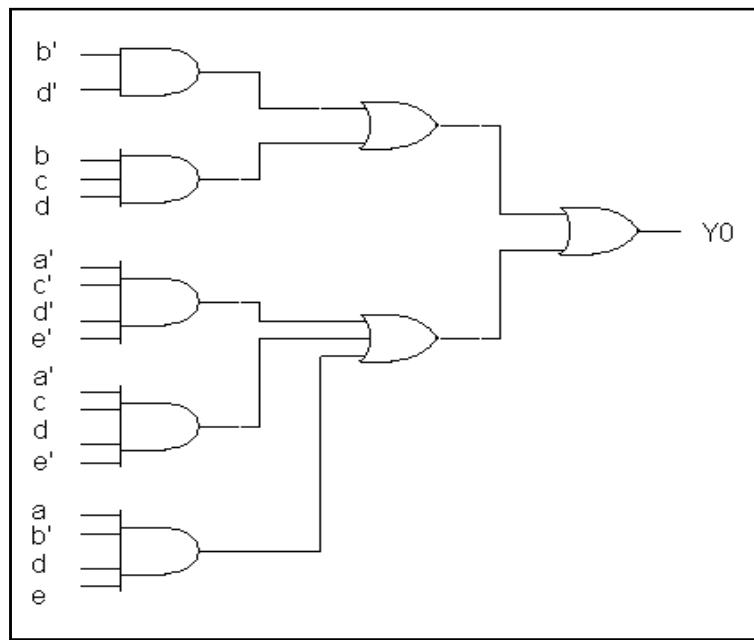
Para  $a=0$

	d	e	00	10	11	01
b	c					
0	0	1	1			
0	0			1	1	1
0	1		x	1	1	x
0	1		1		1	

Para  $a=1$

	d	e	00	10	11	01
b	c					
0	0			1		1
0	0		1	1	1	
0	1		x	1	1	x
0	1					1

$$Y_0 = b'd' + bcd + a'c'd'e' + a'cde' + ab'de$$

**Figura 4-11 Diagrama de  $Y_0$** **Reto 1.**

1. Que el alumno lleve a cabo los pasos anteriores de este ejercicio y complete los mapas de Karnaugh y el diagrama para  $Y_1$ .
2. Ejecute la Simulación
3. Compare el resultado obtenido con la Tabla de Verdad.

**Reto 2.**

4. Obtener la tabla de verdad, reducción mediante mapas de Karnaugh e implementar el circuito en **NI Multisim 10.1** para la salida  $Y_1$

**Tomando en cuenta:**

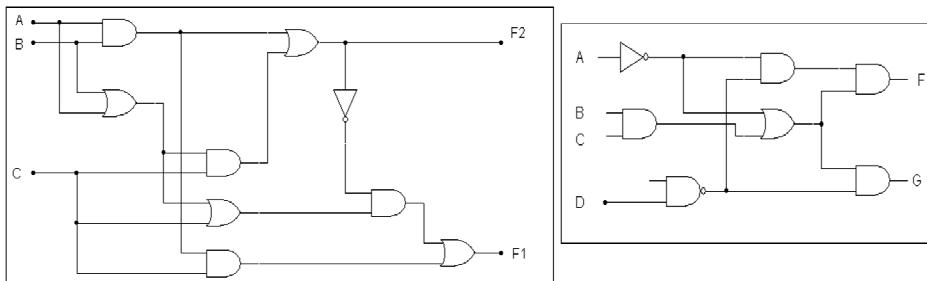
$$Y_1(a, b, c, d, e) = \sum(0, 4, 7, 9, 10, 11, 14, 15, 17, 18, 21, 24, 26, 27, 30, 31)$$

Donde los minterminos 12, 13, 28 y 29, no importa que valores tomen.

**Fin del ejercicio.**

## 4.7 Ejercicio propuesto 1

Obtenga las Tablas de verdad de los siguientes circuitos.



ni.com

 NATIONAL  
INSTRUMENTS™

164

### Definición del Problema.

Como ya se planteo anteriormente el concepto de una Tabla de Verdad., en esta ocasión se partirá de un Diagrama para obtener la Tabla de Verdad y los Mapas de Karnaugh de los dos Circuitos.

### Objetivo.

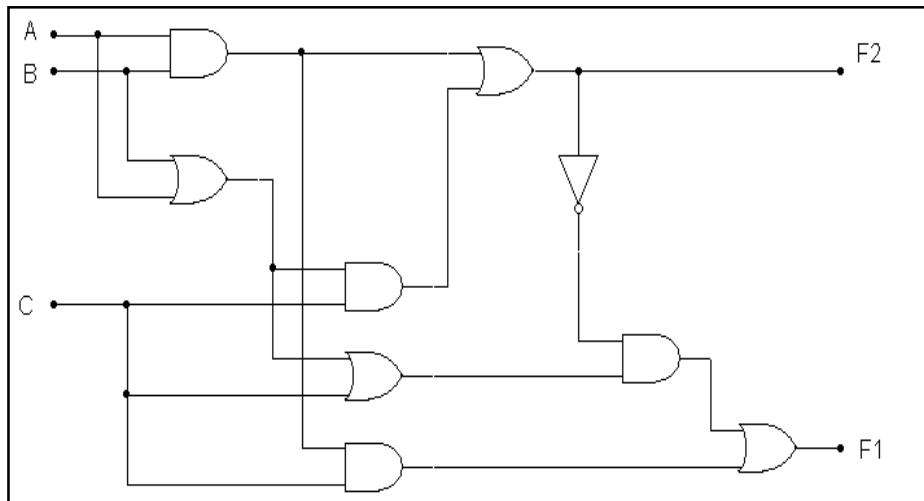
Que el alumno aprenda a plantear ecuaciones booleanas a partir de los diagramas de los circuitos para obtener las Tablas de Verdad.

### Escenario.

A partir de los dos circuitos dados, obtener las ecuaciones booleanas de ambos, establecer las Tablas de Verdad, y finalmente simular los diagramas para compararlos con estas.

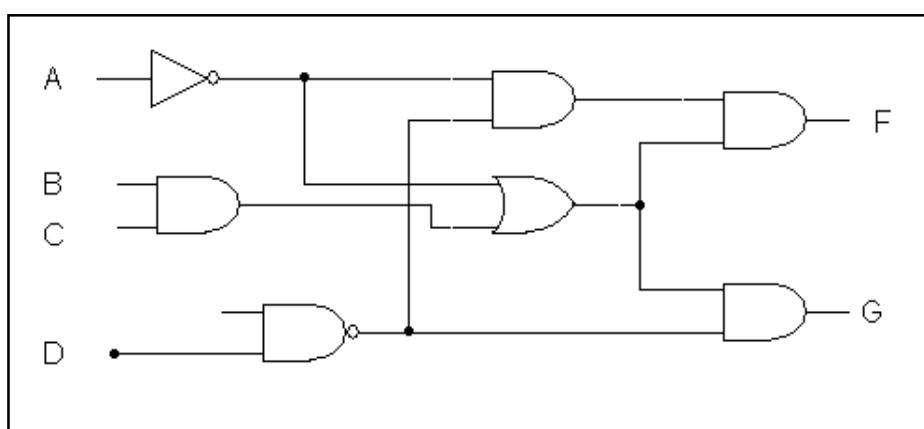
## Diseño

Construir las Tablas de Verdad a partir de los siguientes circuitos



A	B	C	F1	F2
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Figura 4-12 Diagrama del Circuito 1



A	B	C	D	F	G
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	0	
0	0	1	1	1	
0	1	0	0	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	1	1	
1	0	1	0	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	1	0	0	
1	1	1	1	1	

Figura 4-13 Diagrama del Circuito 2

## Prueba.

1. Configure los circuitos para poder llevar a cabo la simulación.
2. Ejecute la Simulación.
3. Compare los resultados obtenidos con la Tabla de Verdad.

**Fin del ejercicio.**

## Reto.

Diseñe un circuito combinacional con 4 entradas y 4 salidas. La salida debe ser el complemento a 2 de la entrada.

## 4.8 Ejercicio propuesto 2

### Comparador de magnitud

La comparación de dos números es una operación que determina si un número es mayor, menor o igual a otro.

Considere dos números A y B de 4 Bits, los cuales podemos representarlos como

$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0$$

ni.com



167

### Definición del Problema.

La comparación de dos números es una operación que determina si un número es mayor, menor o igual a otro.

### Objetivo.

Que el alumno conozca el entorno de NI Multisim 10.1, y el concepto de los Comparadores de Magnitud.

### Escenario.

Considere dos números A y B de 4 Bits, los cuales podemos representarlos como

$$A = A_3 A_2 A_1 A_0 \quad (\text{Ecuacion 4.8-1})$$

$$B = B_3 B_2 B_1 B_0 \quad (\text{Ecuacion 4.8-2})$$

## Diseño

A y B son iguales si todos sus bits son iguales uno con otro, es decir si  $A_0 = B_0$ ,  $A_1 = B_1$ ,  $A_n = B_n$ . Esta relación se puede expresar con:

$$X_i = A_i B_i + A_i' B_i' \quad (\text{Ecuacion 4.8-2})$$

Donde x=1, si y solo si el par de bits en posición son iguales para que la condición de igualdad exista. Todas las variables  $X_i$  deben ser igual a 1.

$$(A = B) = X_3 X_2 X_1 X_0 \quad (\text{Ecuacion 4.8-3})$$

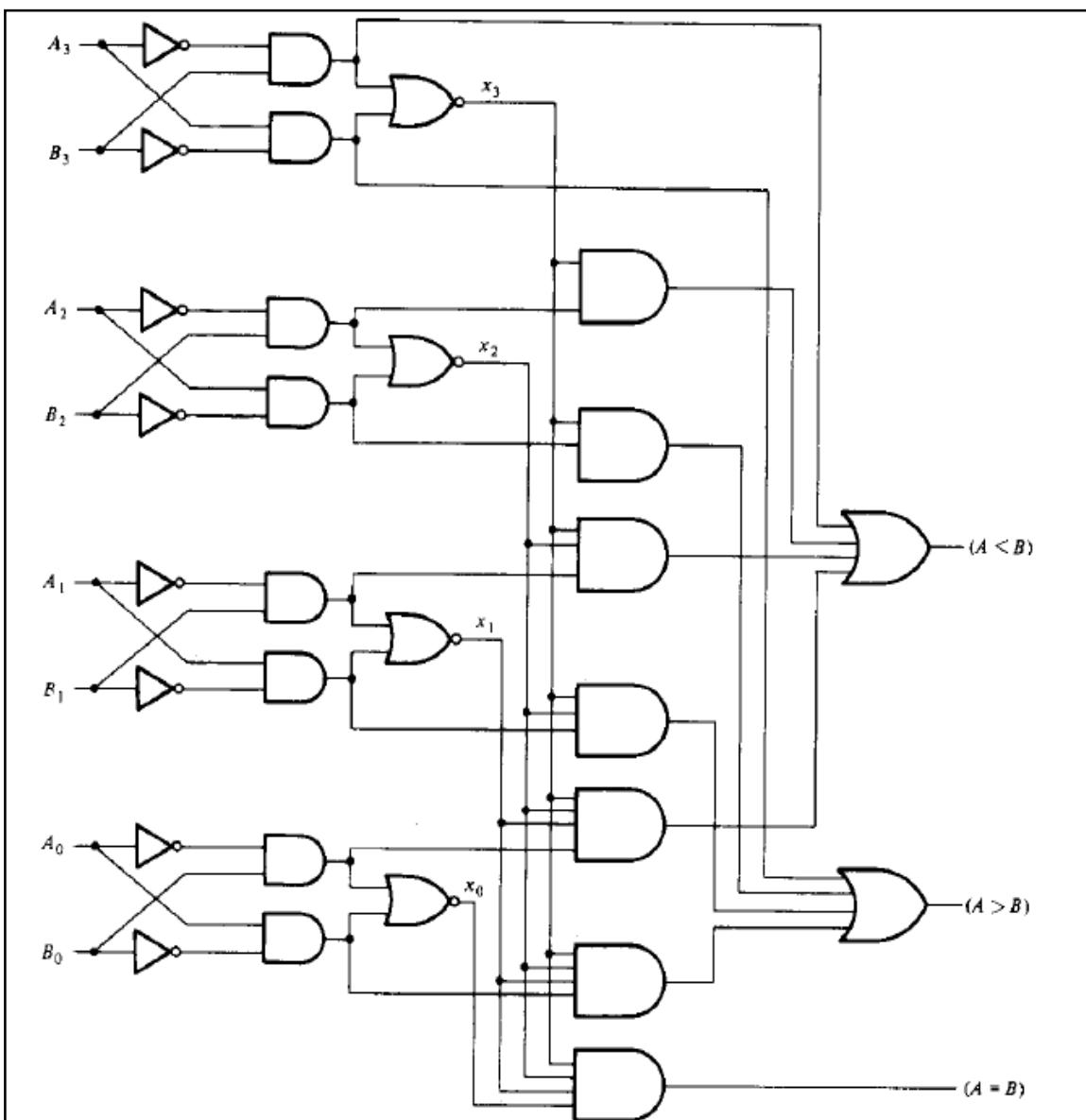
Para determinar si A es mayor o menor a B, se comparan las magnitudes relativas de los pares, comenzando por el más significativo. Si el primer par es igual, entonces el siguiente par, así hasta que se alcance una desigualdad. Si A es igual a 1 y B es igual a 0, podemos concluir que A es mayor que B, de lo contrario si A=0 y B=1 entonces sabemos que B es mayor que A.

Estas comparaciones secuenciales se pueden representar con las siguientes ecuaciones booleanas:

$$\begin{aligned} (A > B) = & A_n B_n' + X_n A_{n-1} B_{n-1}' + \dots + X_n X_{n-1} A_{n-2} B_{n-2}' \\ & + \dots + X_3 X_2 A_1 B_1' + X_3 X_2 X_1 A_0 B_0' \\ (A < B) = & A_n' B_n + X_n A_{n-1}' B_{n-1} + \dots + X_n X_{n-1} A_{n-2}' B_{n-2} \\ & + \dots + X_3 X_2 A_1' B_1 + X_3 X_2 X_1 A_0' B_0 \end{aligned} \quad (\text{Ecuacion 4.8-4})$$

## Implementación.

1. Abra NI Multisim 10.1
2. Mediante NI Multisim 10.1, implemente el circuito comparador para 8 bits. Como referencia utilice el comparador de 4 Bits (Figura 4-14). Haga el circuito físico.

**Figura 4-14** Comparador de 4 Bits**Prueba.**

1. Configure los circuitos para poder llevar a cabo la simulación.
2. Ejecute la Simulación.
3. Compare los resultados obtenidos con la Tabla de Verdad.

**Fin del ejercicio.**

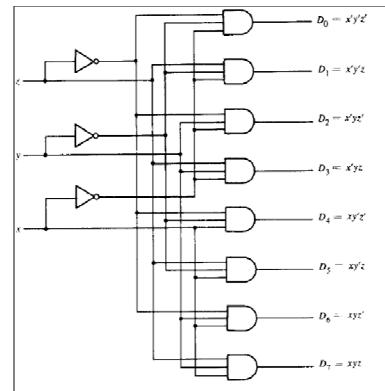
## 4.9 Proyecto Propuesto

Decodificador n a m.

n=bits de entrada

m=bits de salida

$$m = 2^n$$



ni.com

NATIONAL  
INSTRUMENTS™

170

### Definición del Problema.

Un código binario de n bits es capaz de representar hasta  $2^n$  elementos diferentes de información codificada. Un decodificador es un circuito combinacional que convierte la información binaria n líneas de entrada a un máximo de  $2^n$  líneas de salida. Si la información decodificada de n bits tiene combinaciones no utilizadas o no importa, la salida tendrá menos de  $2^n$  salidas.

### Objetivo.

Que el alumno conozca el entorno de NI Multisim 10.1 y aprenda sobre el concepto de Decodificador.

### Escenario.

De acuerdo a la tabla de verdad (Tabla 4-8) propuesta diseñe un circuito (La Figura 4-15 puede ser de apoyo) que la implemente. Haga el circuito en NI Multisim 10.1 y una vez probado impleméntelo físicamente.

## Diseño

Apóyese en la siguiente Tabla de Verdad para obtener los mapas

a	b	c	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Tabla 4-8 Decodificador n a m

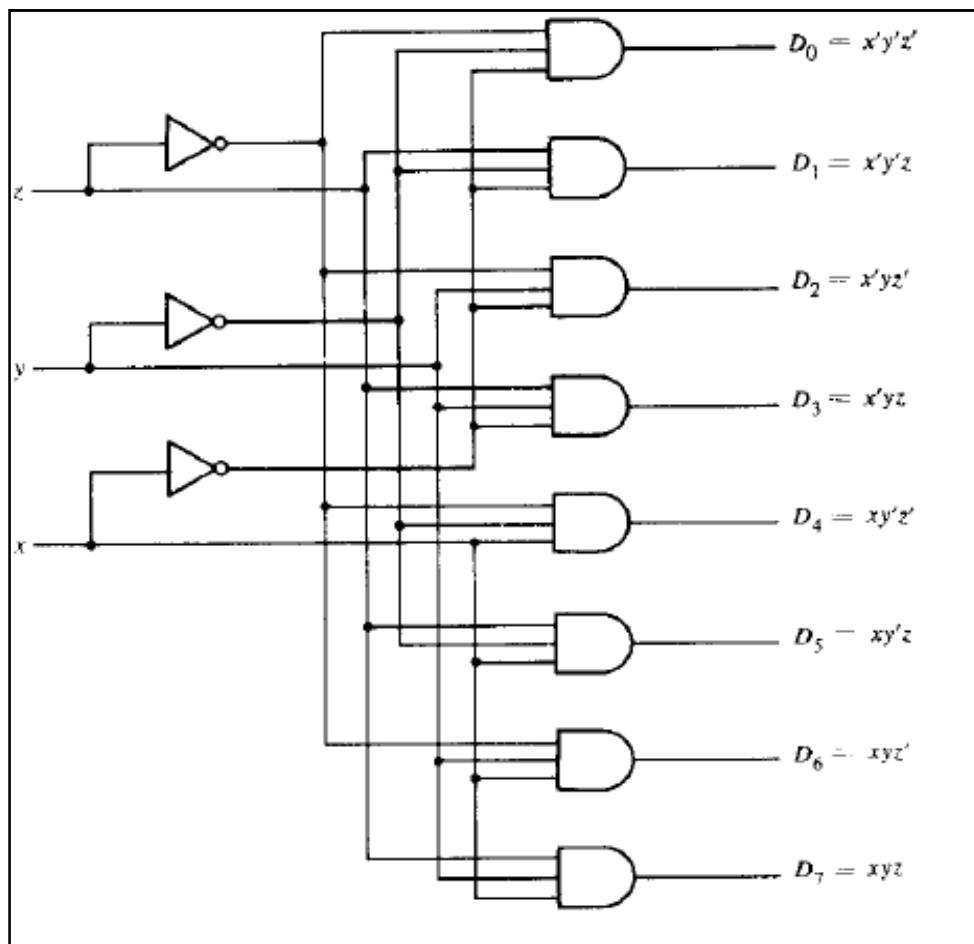


Figura 4-15 Decodificador n a m tomada del folleto de Digital Design de Morris Mano.

## Implementación y Reto.

Que el alumno sea capaz de realizar este proyecto propuesto al final del curso con las bases que se proporcionaron en ejercicios anteriores.

## Prueba.

1. Configure los circuitos para poder llevar a cabo la simulación.
2. Ejecute la Simulación.
3. Compare los resultados obtenidos con la Tabla de Verdad.

**Fin del ejercicio.**

## Notas

---

## Referencias

---

- **Guía de Aprendizaje “LabVIEW en 6 Horas”**, [www.ni.com/mexico](http://www.ni.com/mexico)
- *M. Morris Mano*, **Digital Design**, 2da ed., Prentice Hall, 1990, 516 p.
- *Robert L. Boylestad, Louis Nashelsky*, **Electrónica: Teoría de Circuitos y Dispositivos Electrónicos**, 8va ed., Pearson Educación, 2003, 1020 p.
- *Adel S. Sedra Et Al* , **Circuitos Micro electrónicos**, 4ta ed., Oxford University Press, 1998, 1283 p.
- *Neamen D.*, **Electronic Circuit Analysis and Design**, 2da Ed., Mc Graw Hill.