

Maestría en Ingeniería Eléctrica especialización Telecomunicaciones

Diseño Digital

**Generador de Señal de Síntesis Digital Directa**

**DDS**

Tabla de contenido

[**Descripción** 3](#_Toc99640201)

[Creación de la Señal 3](#_Toc99640202)

[Almacenamiento de la señal en RAM 3](#_Toc99640203)

[Lectura de la señal 3](#_Toc99640204)

[Autómata Necesario 3](#_Toc99640205)

[**Diseño de Módulos con Verilog** 3](#_Toc99640206)

[Diseño Lector archivo .txt 3](#_Toc99640207)

[Diseño Memoria RAM 3](#_Toc99640208)

[Diseño Maquinas de Estados 3](#_Toc99640209)

[Diseño Top Module DDS “ARQUITECTURA” 3](#_Toc99640210)

[**Pruebas** 3](#_Toc99640211)

[Modulo RAM 3](#_Toc99640212)

[Modulo FSM 3](#_Toc99640213)

[Modulo TOP DDS GOLDEN 3](#_Toc99640214)

[Prueba Funcional en Hardware 3](#_Toc99640215)

[**Conclusiones** 3](#_Toc99640216)

[**Referencias** 4](#_Toc99640217)

# **Descripción**

Que es un DDS.

Para que se usa

Que parte del Sistema de Radio bases Actuales lo utilizan.

DDS aplicaciones y sistemas complejos que lo incluyen.

## Creación de la Señal

Para la creación de la señal que se va a utilizar en esta práctica necesitamos generar una señal senoidal de salida, sin embargo, necesitamos tener previamente esta señal almacenada de algún lado como lo puede ser la memoria RAM.

Entonces crearemos una señal senoidal con Matlab con las siguientes funciones.

fc = 40;

time = 0:1/fc:.24;

wc = 2\*pi\*time;

signal = sin(wc+time);

plot(time, signal);

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Ahora esta señal la encapsulamos en un array que pueda ser leído o guardado en un archivo de texto (.txt) esta señal guardada cada punto en cada ubicación de un array de x cantidad de saltos de línea, me convierte cada salto de línea en un byte que puede almacenar la memoria RAM y de ahí depende el tamaño de la memoria RAM a utilizar.

Caso en que solo se almacena la mitad de la señal senoidal

Caso en que solo se almacena ¼ de la señal senoidal

Caso en que se almacena toda la señal senoidal

## Almacenamiento de la señal en RAM

La memoria RAM de puerto simple utilizada en este proyecto necesita una lógica de estados para ser utilizada ya que como solo es una salida necesita tiempo para escribir o leer, vamos a ver mas adelante con el desarrollo del autómata como es que se declararon los estados de la maquina de estados.

Normalmente cuando se modela un subsistema de memoria el objeto en disco esta cargado en un array de memoria durante la inicialización.

## Lectura de la señal

Se puede implementar una FIFO, pero es complicar mucho si solo se necesita leer algo que ya esta en memoria, sin embargo, el fifo puede servir para variar la señal de salida de frecuencia, en caso de requerir un CHIRP frecuencia y el reloj o habilitador siempre incrementa o decrementa dependiendo la velocidad de ese reloj.

## Autómata Necesario

El Autómata se requiere para limitar la operación del diseño en solo algunos estados que pueden intercambiar entre ellos mediante sus entradas o salidas de cada estado.

Se necesitan 4 estados para que este modulo dds funcione en su primera fase de desarrollo, primero el estado IDLE que es el estado de reposo, el estado de lectura de memoria RAM, estado de captura de llenado de información en RAM, estado de salida de señal resultante por cada salida.

## **Verificacion**

Se pretende utilizar mediante se va diseñando el modulo la librería OVL junto con ciertas funciones **assert** que facilitan el desarrollo y la velocidad de detección +de fallas en un diseño.

# **Diseño de Módulos con Verilog**

## Diseño Lector archivo .txt

## Diseño Memoria RAM

## Diseño Maquinas de Estados

## Diseño Top Module DDS “ARQUITECTURA”

# **Pruebas**

## Modulo RAM

## Modulo FSM

## Modulo TOP DDS GOLDEN

## Prueba Funcional en Hardware

# **Conclusiones**

# **Referencias**

[1] Digital design with verilog HDL, Rajvir Singh, yatin trivedi

[2] Verilog digital syste design register transfer level, synthesis, testbench and verification, Zainalabedin Navabi.