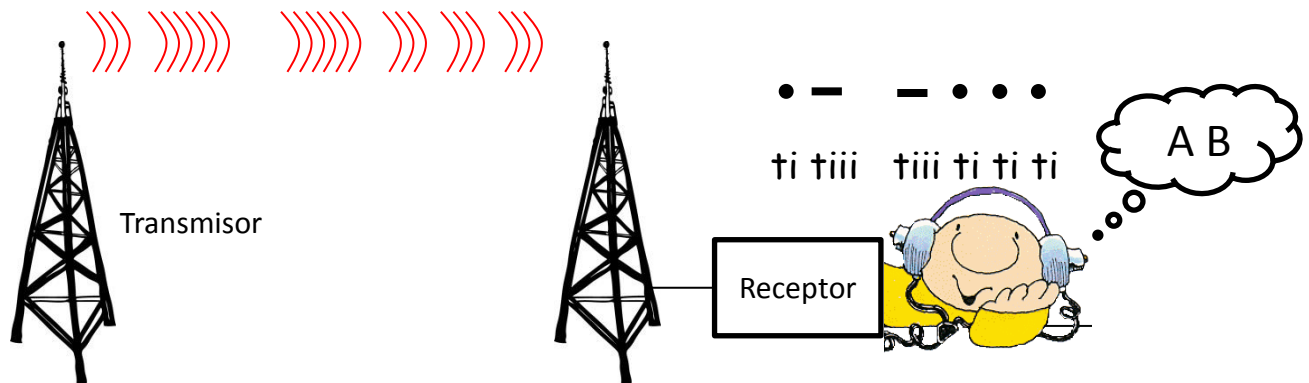


Circuitos Electrónicos (CELT)

Descripción del proyecto

Curso 2018-2019

Decodificador de código Morse



Álvaro de Guzmán Fernández
Miguel Ángel Sánchez

ÍNDICE GENERAL

1. INTRODUCCIÓN.....	4
Calendario de la asignatura.....	5
2. DESCRIPCIÓN GENERAL.....	6
2.1 Objetivo del proyecto.....	7
2.2 Nomenclatura.....	7
2.3 Esquema general: circuito analógico.....	7
2.4 Esquema general: circuito digital.....	8
3. REALIZACIÓN DEL PROTOTIPO.....	9
3.1 Descomposición en módulos analógicos y digitales.....	9
3.2 Forma de trabajar en la asignatura.....	9
3.3 Equipo necesario.....	9
3.4 Montaje.....	10
3.5 Descripciones en VHDL.....	10
3.6 Obtención de la señal de pulsos.....	13
4. FUNCIONAMIENTO DETALLADO.....	14
4.1 SUBSISTEMA ANALÓGICO.....	14
4.1.1 Descripción detallada del circuito analógico.....	14
4.1.2 Acondicionador de señal.....	15
4.1.3 Amplificador de potencia para auriculares.....	16
4.1.4 Filtro paso banda.....	16
4.1.5 Rectificador.....	17
4.1.6 Filtro paso bajo.....	17
4.1.7 Comparador.....	18
4.2 SUBSISTEMA DIGITAL.....	18
4.2.1 Estructura general del subsistema digital.....	19
4.2.2 Estructura del receptor.....	21
4.2.3 Representación binaria del morse y caracteres asociados en los displays.....	22
4.2.4 Funcionamiento del receptor.....	23
4.2.5 Divisor del reloj.....	24
4.2.6 Descripción del módulo de visualización.....	25
4.2.6.1 Visualización: Decodificador de Morse a 7 segmentos.....	25
4.2.6.2 Visualización: Multiplexor.....	25
4.2.6.3 Visualización: Refresco.....	26
4.2.6.4 Visualización: Registro de desplazamiento.....	26
4.2.6.5 Visualización: Cableado final y pruebas.....	27
4.2.7 Detector de flanco.....	28
4.2.8 Automata de medida de duración de 0s y 1s.....	31
4.2.9 Comparadores.....	32
4.2.10 Automata de control.....	33
4.2.11 Cableado del circuito completo y fichero de asociaciones.....	35
4.3 MONTAJE DEL PROTOTIPO.....	37
4.3.1 Montaje del circuito por etapas.....	37
4.3.2 Medidas que deben realizarse sobre el circuito.....	37
4.3.3 Realización de diagramas de Bode.....	38
4.3.4 Utilización del osciloscopio.....	39
5. ESPECIFICACIONES DEL SISTEMA.....	39
6. DESARROLLO RECOMENDADO POR SESIONES.....	40
ENTREGAS.....	44
7. MEMORIA FINAL PARA EXAMEN ORAL.....	49
8. MEJORAS.....	50
8.1 Introducción de un separador entre palabras (dificultad baja).....	50
8.2 Incorporación de una señal asíncrona de reset (dificultad baja).....	50
8.3 Detección del final del mensaje (dificultad media).....	50
8.4 Detección avanzada del final del mensaje (dificultad alta).....	51
8.5 Uso de esquemas circuitales alternativos a los propuestos.....	51
8.6 Simulación con PSPICE ó LTSPICE (dificultad alta).....	51
8.7 Esquema alternativo para la detección de código morse transmitido manualmente (dificultad alta).....	52
Referencias.....	53

ÍNDICE DE FIGURAS

Figura 1. Código Morse.....	6
Figura 2. Izquierda: pantalla multifuncional de un avión donde se muestra el identificador del ILS sintonizado (AVS aeropuerto de Asturias). Derecha: carta de navegación aérea donde se muestran diversas radioayudas con sus identificadores Morse.	6
Figura 3. Señal morse suministrada para su decodificación.	7
Figura 4. Ejemplo de señal transmitida.....	7
Figura 5. Funcionamiento general del circuito analógico	8
Figura 6. Funcionamiento general del circuito digital	8
Figura 7 Esquema con la placa de inserción y la conexión de las bornas de alimentación.....	11
Figura 8 Condensadores de desacoplo y LEDs para la detección rápida de cortocircuitos.	11
Figura 9 Salida analógica del entrenador donde se obtiene la señal de pulsos.	13
Figura 10 Izquierda: entradas y salidas empleadas por el hardware gen_senal.vhd para generar la señal analógica. Derecha: disposición de cables para generar la señal analógica de salida del entrenador ENT2004CF.....	13
Figura 11 Esquema detallado del circuito analógico y de las formas de onda en cada punto del circuito.....	14
Figura 12 Filtro paso alto, atenuador y separador del acondicionador de señal.	15
Figura 13 Amplificador de potencia basado en el integrado LM386 y patillaje del jack	16
Figura 14 Filtro paso banda de realimentación múltiple de adaptación al canal.	16
Figura 15 Rectificador construido con un diodo.....	17
Figura 16 Filtro paso bajo Sallen-Key.....	17
Figura 17 El comparador LM311 y esquema del proceso de comparación. ATENCIÓN: las señales de umbral/salida (negra/roja) NO están representadas en la misma escala que la señal de salida (verde).	18
Figura 18 Esquema de bloques del subsistema digital completo.....	19
Figura 19 Estructura del receptor con sus bloques fundamentales	21
Figura 20 Representación en los displays de todos los caracteres.....	22
Figura 21 Diagrama de bloques del módulo de visualización y cronograma de su funcionamiento.	25
Figura 22 Diagrama de estados del detector de flanco. Se comporta como un comparador con histéresis	29
Figura 23 Muestreo constante de la señal en el tiempo. a) Aparecen glitches, pero la suma acumulada no alcanza el umbral y la salida (VALOR) se mantiene a 0 ignorando el glitch. b) En un flanco con rebotes, solamente cuando la suma alcanza el umbral es cuando la salida (VALOR) toma el valor 1 eliminando los rebotes.	29
Figura 24 Diagrama de estados del autómata de medida de la duración de 0s y 1s	31
Figura 25 Diagrama de estados del autómata de control.....	33
Figura 26 Esquema del circuito de medida de los filtros. Se muestra un ejemplo real de cómo calcular el módulo y la fase de la función de transferencia.	38

1. INTRODUCCIÓN

El objetivo de esta asignatura consiste en que el alumno amplíe y consolide de una manera práctica los conocimientos adquiridos en las asignaturas de segundo curso de Electrónica Analógica y Digital. Para ello se utilizará un método **PBL** ("**Project based learning**", **aprendizaje basado en proyectos**) que consistirá en el desarrollo de un prototipo a partir de unas especificaciones iniciales.

Esta tarea se lleva a cabo mediante **evaluación continua** y comprende varias fases:

- Por un lado, se impartirán unas clases introductorias donde se suministra la información básica sobre el proyecto a realizar.
- Existirán unas sesiones prácticas en un aula docente con medios para el desarrollo del proyecto propuesto. Dicho proyecto será realizado por parejas de alumnos.
- Además, deberá seguir las instrucciones aquí incluidas, que implicarán diversas fases de diseño, análisis, montaje y medida de los circuitos o subsistemas propuestos. Igualmente se hará especial énfasis en que los alumnos adquieran una visión práctica de los problemas con los que se encuentra el diseño de circuitos analógicos y digitales en las implementaciones de prototipos reales de laboratorio.
- También existirán exposiciones cortas diarias en el aula docente de la asignatura durante el desarrollo de las sesiones de prácticas

El resultado del trabajo realizado **deberá quedar reflejado en tres documentos entregables en las sesiones 5, 8 y 11 del curso, junto con una memoria escrita final** (para entregar los días 10, 11 y 12 de diciembre de 2018) que contengan los detalles del proceso, así como los resultados obtenidos y todas aquellas cuestiones específicas que se indiquen en el enunciado.

El proyecto propuesto contiene las **especificaciones mínimas** que deben cumplir los circuitos realizados para obtener un máximo de 8 puntos. Adicionalmente, se presentarán sugerencias de **mejoras opcionales** que permitirán alcanzar notas mayores, dejando a los alumnos la libertad de añadir otras mejoras y montajes alternativos (ver Apartado 8). Podrá encontrar éste y otros documentos relacionados, así como información actualizada sobre la asignatura, en la página MOODLE de la asignatura

Durante este curso deberá tener en cuenta las siguientes consideraciones:

1. **Cada pareja deberá disponer de un cuaderno de prácticas**, donde dibujará los esquemas de los circuitos, detallará los cálculos y justificaciones realizados, tomará nota de todas las medidas efectuadas en el Laboratorio y, en general, recogerá cuantas observaciones e incidencias tengan lugar durante el desarrollo del proyecto. Este cuaderno **será de gran utilidad a la hora de escribir los documentos entregables y la memoria**.
2. La evaluación continua se realizará considerando la evolución semanal de los alumnos, los tres documentos entregables, un examen oral donde se evaluarán: a) el funcionamiento del prototipo, b) la memoria, c) los conocimientos sobre el diseño y d) el uso de los equipos de laboratorio. Por último habrá también una prueba escrita. Tenga en cuenta que aunque el proyecto se realiza por parejas, **la nota será individual para cada alumno**.
3. Mediante esta evaluación, la nota máxima que podrá obtener será 8.0 puntos sobre 10. Para obtener notas más altas será imprescindible realizar mejoras (apartado 8).
4. **Para aprobar la asignatura es imprescindible que el prototipo funcione correctamente y que se demuestren los conocimientos necesarios sobre el funcionamiento del mismo.**

Para cualquier consulta, puede dirigirse al coordinador de la asignatura Miguel Ángel Sánchez, despacho B-107, (sanchez@die.upm.es).

Calendario de la asignatura:

Se muestra a continuación el calendario de la asignatura. Cada pareja tiene 12 sesiones de laboratorio. Existe la posibilidad de reservar sesiones adicionales mediante el portal de la asignatura. Las sesiones que aparecen como TL son turnos libres. Durante la primera semana estará abierto el periodo de elección de turno en la asignatura y deberá hacerse a través del portal de Moodle. **Es obligatoria la asistencia a una de las dos clases introductorias que se impartirán los días 6 y 10 de Septiembre.**

CALENDARIO CELT 2018-2019

	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
	3	4	5	6	7
	Periodo de elección de turno				
				Clase introductoria	
SEP	10 LT1 Clase introductoria	11 MT1	12 XT1	13 JT1	14 VC1
	17 JT2 (de 10 a 13) LT2	18 MT2	19 XT2	20 JT3	21 VC2
	24	25 MT3	26 XT3	27 JT4	28 VC3
	1 XT4 (de 10 a 13) LT3	2 MT4	3 XT5 Entrega 1	4 JT5 Entrega 1	5 VC4
OCT	8 LT4	9 Entrega 1 VC5 (Clase de viernes)	10 XT6	11 JT6	12
	15 LT5 Entrega 1	16 MT5 Entrega 1	17 XT7	18 JT7	19 VC6
	22 LT6	23 MT6	24 VC7 (Clase de viernes)	25 JT8 Entrega 2	26 VC8 Entrega 2
	29 LT7	30 MT7	31 XT8 Entrega 2	1	2
	5 LT8 Entrega 2	6 MT8 Entrega 2	7 VC9 (Clase de viernes)	8 JT9	9
NOV	12 LT9	13 MT9	14 XT9	15 JT10	16 VC10
	19 LT10	20 MT10	21 XT10	22 JT11 Entrega 3	23 VC11 Entrega 3
	26 LT11 Entrega 3	27 MT11 Entrega 3	28 XT11 Entrega 3	29 JT12	30 VC12
	3 LT12 Examen escrito	4 MT12	5 XT12	6	7
DIC	10 TL Entrega memorias LT, JT, VC	11 TL Entrega memorias MT	12 TL Entrega memorias XT	13 TL	14 TL
	17	18	19	20	21
	EXÁMENES ORALES				

2. DESCRIPCIÓN GENERAL

El Código Morse es un código de transmisión en el que cada letra del alfabeto y cada cifra se codifica mediante una secuencia de símbolos de diferente duración separados por pausas que se conocen como puntos (duración corta) y rayas (duración larga). Entre cada dos letras o números se transmite una pausa más larga. Este código fue inventado en 1830 para la comunicación entre dos estaciones terrestres conectadas por un cable telegráfico. Cuando posteriormente se inventó la telegrafía sin hilos, comenzó a emplearse igualmente transmitiéndose en este caso pulsos de radio. El código Morse se utilizó ampliamente durante la segunda mitad del siglo XIX y siguió utilizándose hasta la aparición de los primeros sistemas de modulación modernos (AM, FM, etc...) (ver Figura 1).

A	.-	J	..---	S	...-	2	..---
B	K	---.	T	-	3
C	----	L	U	..-	4
D	---	M	--	V	5
E	.	N	--	W	---	6
F	O	---	X	----	7	----
G	---	P	Y	----	8	----
H	Q	----	Z	----	9	----
I	..	R	---	1	-----	0	-----

Figura 1. Código Morse

Aunque actualmente el empleo del Morse está prácticamente obsoleto, todavía se usa en algunas aplicaciones del entorno aeronáutico y de la radioafición. Concretamente, las radioayudas tales como: balizas no direccionales (NDB) y omnidireccionales (VOR), así como el sistema de aterrizaje instrumental (ILS), transmiten en Morse la identificación de la citada baliza o del aeropuerto. Las cartas de navegación aérea muestran la identificación de estas balizas, y las pantallas multifuncionales de las aeronaves la presentan tras decodificar los pulsos de radio transmitidos (ver Figura 2).

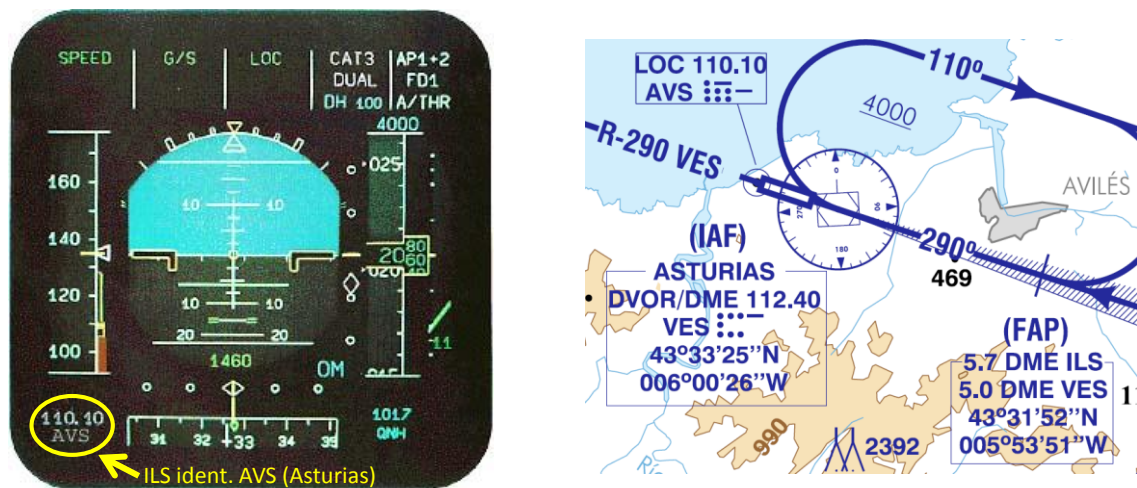


Figura 2. Izquierda: pantalla multifuncional de un avión donde se muestra el identificador del ILS sintonizado (AVS aeropuerto de Asturias). Derecha: carta de navegación aérea donde se muestran diversas radioayudas con sus identificadores Morse.

En la transmisión de un mensaje se produce por tanto una sucesión de pulsos de radio de diferente duración a una frecuencia determinada (transmisión de onda continua o CW [Continuous Wave]). En el receptor, estos pulsos son transportados mediante heterodinaje a una frecuencia más baja dentro del intervalo audible (20 a 20.000 Hz), siendo percibidos como pulsos de audio a través de un altavoz. Un operador experimentado puede decodificar estos pulsos y componer el mensaje original, por eso hasta hace unos años se exigía la interpretación del Morse para la obtención de licencias de operador de radio en los cuerpos militares y en la radioafición. En la actualidad esta tarea la llevan a cabo ordenadores o sistemas digitales como el que vamos a realizar en este proyecto.

2.1 Objetivo del proyecto

En este proyecto se desarrollará un sistema de decodificación morse partiendo de una señal de audio con los pulsos (puntos y rayas) correspondientes a un mensaje que habrá que presentar en los displays del entorno de desarrollo BASYS2 disponible en el laboratorio. Los pulsos de audio tienen una frecuencia de 1 KHz y se obtienen de la salida analógica del entrenador disponible también en el laboratorio. El aspecto en el dominio del tiempo de la citada señal se asemeja a lo que se muestra en la siguiente Figura 3.

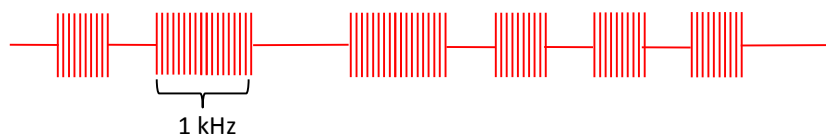


Figura 3. Señal morse suministrada para su decodificación.

El proyecto se compone de dos partes:

- un circuito analógico realizado sobre una placa de inserción que convierte los pulsos de audio en una señal digital (1s y 0s correspondientes a los pulsos y las separaciones entre los pulsos respectivamente).
- un circuito digital implementado sobre una FPGA que interpreta la señal digital obtenida del circuito anterior para decodificar el mensaje morse y presentarlo en los displays.

2.2 Nomenclatura

A lo largo de este proyecto utilizaremos la siguiente nomenclatura (ver Figura 4):

- **Código:** es la representación morse de un carácter. Por ejemplo: código de la B = -...
- **Punto:** pulso de duración corta.
- **Raya:** pulso de duración larga.
- **Símbolo:** cada uno de los puntos y las rayas.
- **Pausa:** separación entre símbolos (duración corta).
- **Espacio:** separación entre códigos (duración larga).

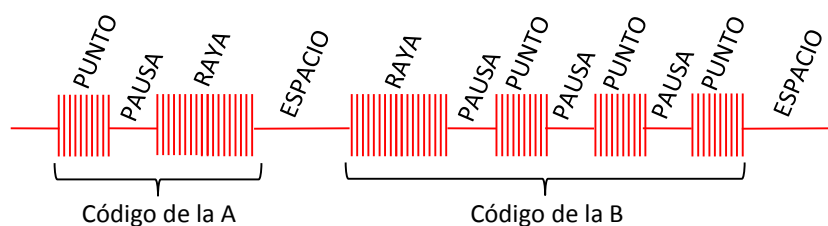


Figura 4. Ejemplo de señal transmitida

2.3 Esquema general: circuito analógico

El circuito analógico se realizará sobre una placa de inserción utilizando componentes electrónicos. Deberá alimentarse con tensión simétrica de +5 y -5 V procedentes de la fuente de alimentación del laboratorio.

La señal de entrada procederá de la salida analógica del entrenador disponible en el laboratorio.

La misión de este circuito consiste en convertir los pulsos de audio en una señal digital entre 0 y 5V utilizando un rectificador tal y como se describe más adelante en esta memoria. Además también deberá amplificar dichos pulsos para escucharlos a través de unos auriculares. El aspecto de este circuito y de las señales obtenidas se muestra a continuación en la Figura 5:

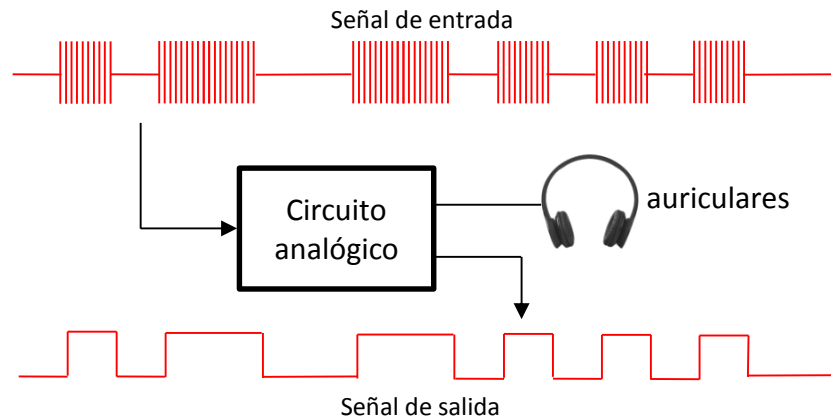


Figura 5. Funcionamiento general del circuito analógico

2.4 Esquema general: circuito digital

El circuito digital se describirá en VHDL y se sintetizará sobre una FPGA. Para ello se utilizará el entorno ISE Webpack y la tarjeta de desarrollo BASYS2 disponible en el laboratorio. El funcionamiento del circuito consiste en medir el tiempo transcurrido entre dos flancos consecutivos de la señal digital. De este modo el circuito puede determinar la duración de cada intervalo a 0 y a 1. Tendremos por tanto 4 posibilidades:

1. **PUNTO:** Intervalo a 1 de duración corta (100 ms).
2. **RAYA:** Intervalo a 1 de duración larga (300 ms).
3. **PAUSA:** Intervalo a 0 de duración corta (100 ms).
4. **ESPACIO:** Intervalo a 0 de duración larga (300 ms).

Cada vez que se recibe un símbolo el circuito deberá almacenarlo. Cuando se recibe un espacio se completa la recepción de un código y por lo tanto deberá ser representado en el display de 7 segmentos presente en la tarjeta BASYS2. Puesto que solamente hay 4 displays, el código irá desplazándose hacia la izquierda a medida que se vaya decodificando el mensaje (Figura 6).



Figura 6. Funcionamiento general del circuito digital

3. REALIZACIÓN DEL PROTOTIPO

3.1 Descomposición en módulos analógicos y digitales

Para la realización de este prototipo se utilizarán los siguientes elementos:

- Para la parte analógica se emplearán amplificadores operacionales del tipo **TL082** alimentados con tensión simétrica de **+5 y -5 V**.
- La parte digital se realizará mediante síntesis en la FPGA Spartan 3E de la tarjeta BASYS2 que encontrará en el laboratorio. Todo el hardware digital se describirá obligatoriamente en VHDL empleando el entorno **ISE Webpack 14.7** instalado en los ordenadores del laboratorio.

NOTA: El entorno ISE 14.7 es de distribución gratuita y puede descargarse abiertamente en la página:

<https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html>

3.2 Forma de trabajar en la asignatura

Los créditos asignados a esta asignatura se encuentran repartidos entre horas prácticas en el aula docente y trabajo personal del alumno. A este respecto, tenga en cuenta que cada semana dispone de 3 horas prácticas en el aula docente. Estas horas deben dedicarse principalmente a realizar medidas, sintetizar código sobre FPGA, obtener capturas de pantalla del osciloscopio para la redacción de la memoria final y resolver posibles problemas que vayan surgiendo durante el desarrollo del prototipo. **NO SE RECOMIENDA** utilizar el tiempo en el aula para diseñar los circuitos, realizar los montajes de los mismos o escribir código VHDL en el ordenador. Dichos diseños y montajes, así como la escritura y depuración del código deben hacerse fuera del aula en las horas de trabajo personal.

No obstante, para cualquier duda referente al diseño o el montaje del circuito, así como para cualquier problema referente al VHDL, su simulación y síntesis, puede acudir a cualquiera de los profesores de la asignatura.

Aproveche el tiempo en el aula para aprender a utilizar los equipos de laboratorio, realizar medidas y resolver problemas. Sea crítico y trate de buscar soluciones acotando el problema que pueda surgir. En la parte analógica revise las líneas de alimentación de los integrados y examine los módulos por separado. En la parte digital simule los módulos individuales antes de sintetizar e intente analizar su circuito por partes.

Para el circuito analógico ponga especial cuidado en la instalación de **condensadores de desacoplo** (ver apartado 3.4)

3.3 Equipo necesario

Para la realización de este prototipo serán necesarios los siguientes elementos: una placa de inserción, 4 cables banana-banana, 3 cables BNC-pinzas y varios componentes electrónicos.

3.4 Montaje

La parte analógica de este prototipo deberá ser montada en una placa de inserción. La alimentación se realizará mediante la fuente de alimentación del laboratorio en modo simétrico con +5 y -5 V (vea la Figura 7 y siguientes). A la salida del circuito analógico se obtendrá una señal digital de datos con tensiones entre 0 y 5 V que deberá conectar a una de las entradas de la FPGA. **Solamente se permite la conexión de dos cables entre la parte analógica y la FPGA: el cable de masa y el cable de datos.**

Además, **se utilizarán condensadores de desacoplo** en la placa de inserción para reducir el ruido que pueda producirse en los circuitos de conmutación. Estos condensadores son fundamentales para el correcto funcionamiento del proyecto. Se utilizarán 3 en cada una de las alimentaciones: 100 μ F, 100 nF y 100 pF para cada uno de los intervalos de frecuencia (bajas, medias y altas respectivamente). Tenga especial cuidado con la polaridad de los condensadores electrolíticos en el caso de 100 μ F (fíjese en la Figura 8).

Por otro lado, es recomendable añadir dos LEDs (rojo y verde) junto con dos resistencias, conectados a cada una de las líneas de alimentación (+5 y -5) para saber que dichas líneas están funcionando correctamente y que no hay cortocircuitos. En caso de producirse alguno, el LED correspondiente a esa alimentación se apagará advirtiéndolo del hecho. Observe la Figura 8 para la correcta conexión de estos elementos. Repetimos que debe prestar especial atención a la polaridad de los condensadores electrolíticos.

La polaridad de los LEDs la indican la longitud de sus patillas. La patilla más larga se corresponde con el ánodo (terminal positivo).

La justificación de la necesidad de condensadores de desacoplo, así como su función, se describen en detalle en la referencia [6].

Le será más sencillo seguir el curso de los posibles errores de montaje si utiliza cables de colores identificativos para las alimentaciones. **Se sugiere negro para la masa, rojo para la alimentación positiva (+5) y azul para la negativa (-5).**

La distribución de conexiones internas de la placa de inserción, así como posibles sugerencias de cómo colocar los componentes sobre la placa se muestran en las figuras posteriores.

3.5 Descripciones en VHDL

La descripción del hardware se hará exclusivamente empleando el lenguaje VHDL utilizando para ello el entorno de desarrollo ISE Webpack 14.7 que se indica en el apartado 3.1.

Para el correcto funcionamiento de los módulos es imprescindible que importe siempre las siguientes librerías al comienzo de cada módulo en el código VHDL:

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

Tenga en cuenta que **el entorno ISE no importa todas estas librerías automáticamente**, por lo que deberá importarlo manualmente en cada módulo.

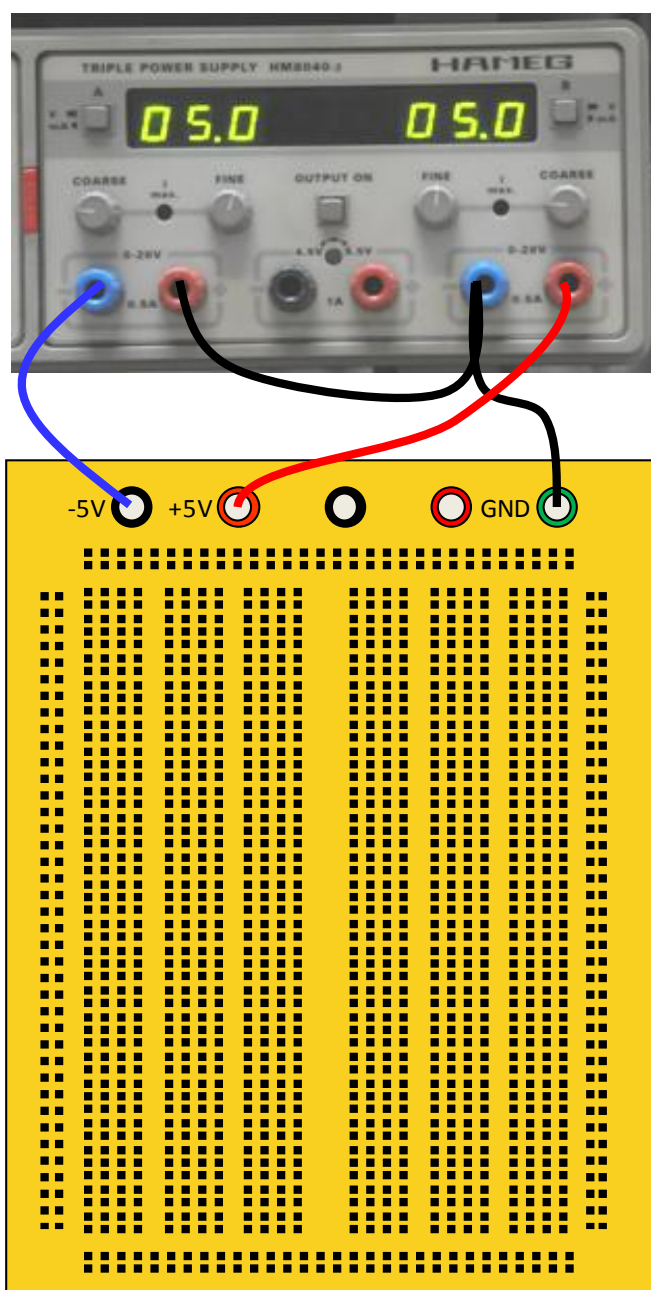


Figura 7 Esquema con la placa de inserción y la conexión de las bornas de alimentación.

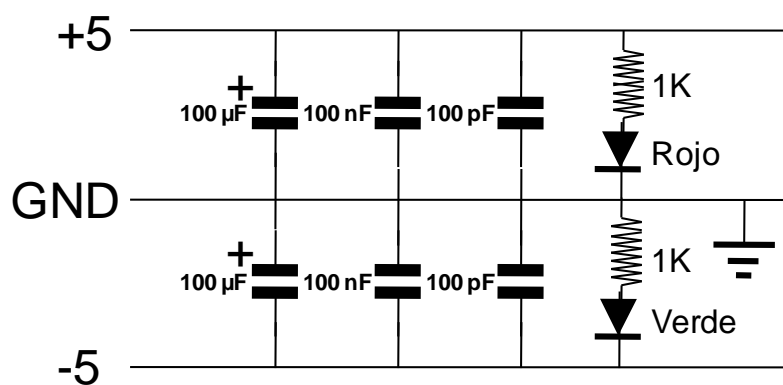
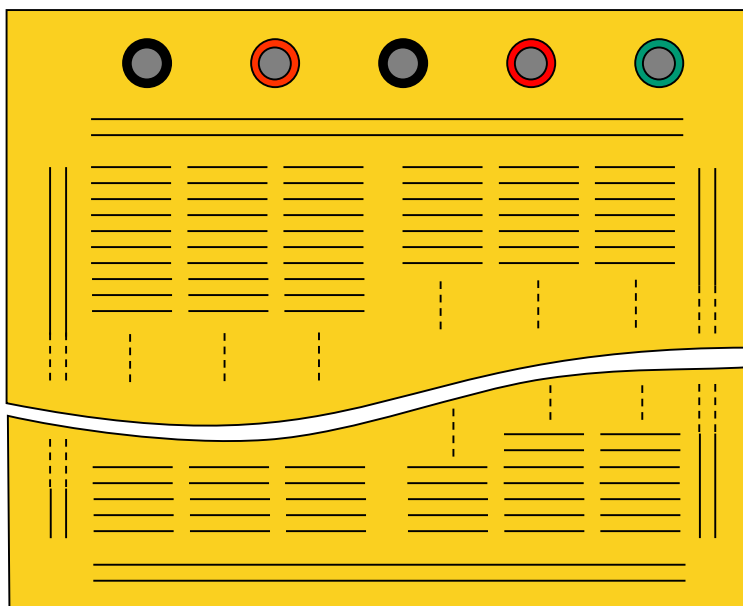
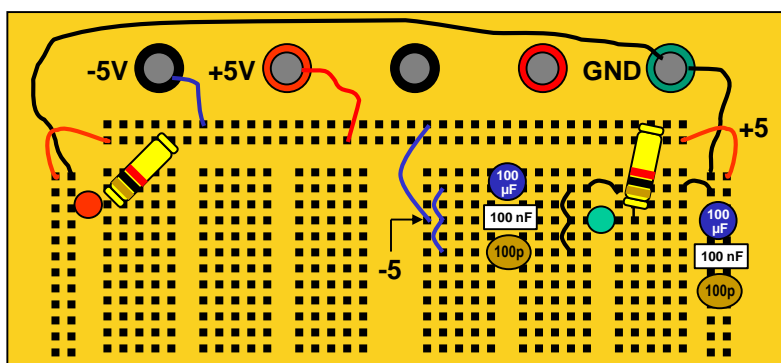


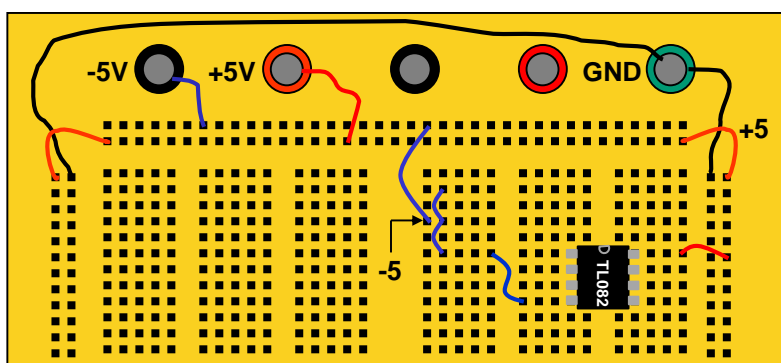
Figura 8 Condensadores de desacoplo y LEDs para la detección rápida de cortocircuitos. Tenga cuidado con la polaridad de los condensadores electrolíticos.



Conexiones internas de la placa de inserción. Las líneas negras indican orificios cortocircuitados.



Sugerencia sobre cómo colocar los condensadores de desacoplo y los LEDs de detección de cortocircuitos



Sugerencia sobre cómo colocar un circuito integrado TL082 y su alimentación.

3.6 Obtención de la señal de pulsos

La señal de pulsos para la realización de este proyecto se obtiene de la salida analógica del entrenador (conector BNC en la parte frontal). Para conectar la señal al circuito necesitará utilizar un cable coaxial BNC-pinzas a la salida analógica del entrenador tal como se indica en la Figura 9.

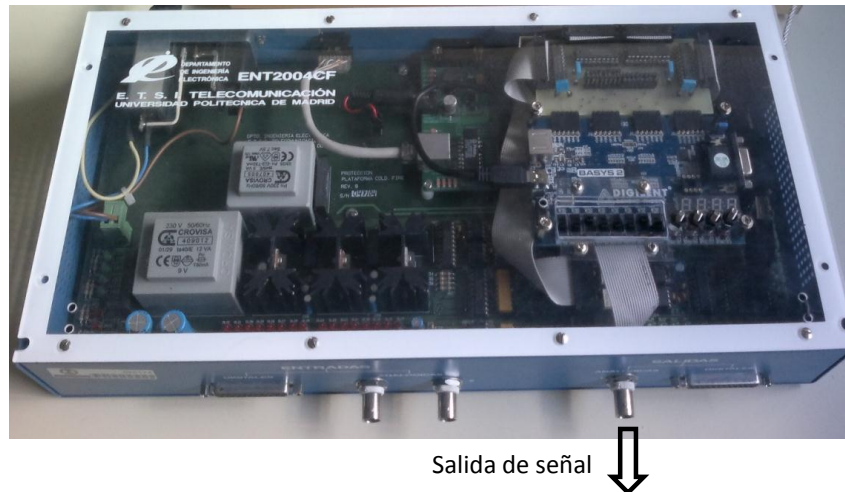


Figura 9 Salida analógica del entrenador donde se obtiene la señal de pulsos.

Para generar esta señal deberá generar un proyecto en el entorno ISE con el fichero **gen_senal.vhd** y el archivo de asociaciones **asoc.ucf** suministrados en la plataforma MOODLE. Compile estos ficheros y genere el bitstream. A continuación cargue el bitstream en la FPGA mediante la aplicación ADEPT.

El hardware toma como entradas los dos pulsadores de la derecha de la placa BASYS 2 (BTN0 y BTN1, Figura 10).

- **Pulse BTN0 para comenzar a generar la secuencia** a través de la salida analógica.
- **Pulse BTN1 para detener la secuencia.**

Posteriormente pulsaciones de BTN0 vuelven a comenzar la secuencia desde el principio.

El hardware también utiliza tres salidas de la FPGA para programar el convertor digital-analógico interno del entorno de entrenamiento. Sobre la tabla de madera donde se encuentran los conectores de entrada y salida hay que realizar las conexiones siguientes: Salida-14 con Entrada-10, Salida-15 con Entrada-11 y Salida-17 con Entrada-13, según se muestra en la Figura 10 (derecha).

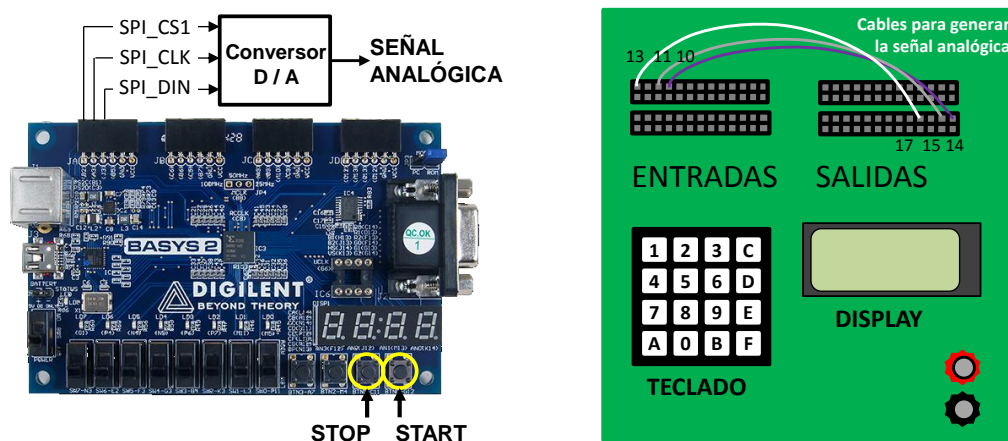


Figura 10 Izquierda: entradas y salidas empleadas por el hardware gen_senal.vhd para generar la señal analógica. Derecha: disposición de cables para generar la señal analógica de salida del entrenador ENT2004CF.

4. FUNCIONAMIENTO DETALLADO

4.1 SUBSISTEMA ANALÓGICO

Antes de comenzar a describir las partes que componen este sistema queremos insistir en una serie de cuestiones fundamentales para el correcto desarrollo del proyecto:

- Este circuito recibirá alimentación mediante la fuente del laboratorio a +5 y -5 V en modo simétrico (ponga especial atención en el uso de condensadores de desacoplo como se describe en el apartado 3.4).
- Los amplificadores operacionales TL082 **deben alimentarse** con tensión simétrica de -5 y +5V.
- Siga siempre las recomendaciones de diseño y asegúrese que un módulo funciona correctamente antes de conectarlo al siguiente. Si piensa que puede realizar alguna simplificación, idea o esquema alternativo, por favor consulte con cualquiera de los profesores de la asignatura antes de abordar su idea. Es posible que proyectos que parecen muy simples se conviertan en una gran dificultad a la hora de llevarlos a la práctica.
- Por último, se recomienda apagar la alimentación antes de realizar cambios sobre el circuito. Muchas veces se producen cortocircuitos durante el movimiento de cables que dan lugar al deterioro de módulos que ya estaban funcionando correctamente.
- Siempre que vaya a utilizar un integrado, descargue y lea su hoja de características hasta que comprenda su funcionamiento correcto.

4.1.1 Descripción detallada del circuito analógico

El circuito analógico debe convertir los pulsos de audio en una señal continua de 5V de la misma duración. Para ello se filtrará la señal y se aprovechará la energía transmitida en dichos pulsos para conseguir una señal continua mediante rectificación y filtrado. Emplearemos diferentes bloques que se muestran en la Figura 11 junto con las formas de onda en cada punto del circuito.

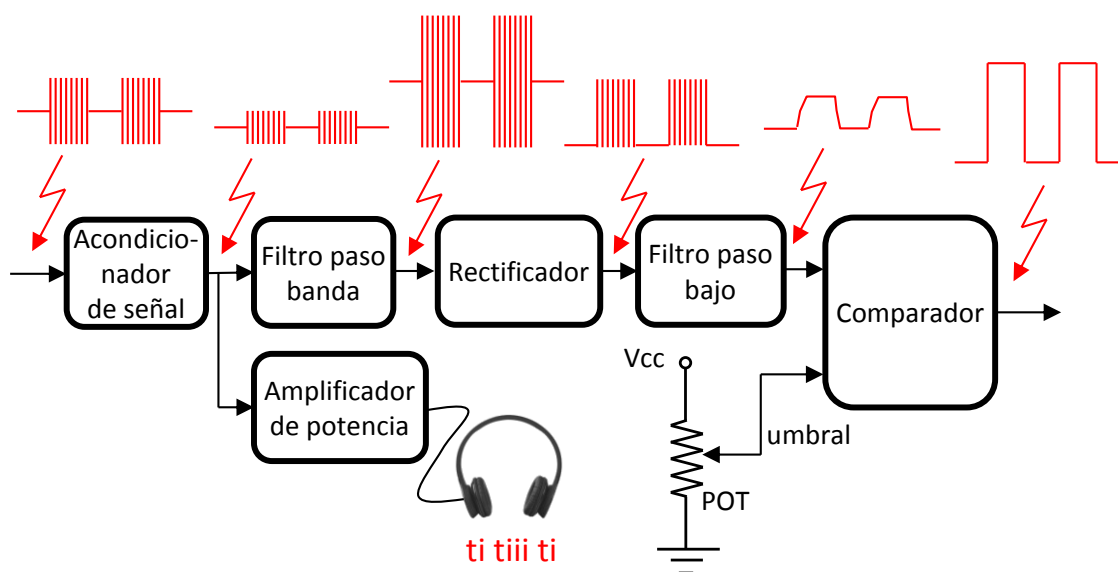


Figura 11 Esquema detallado del circuito analógico y de las formas de onda en cada punto del circuito.

El funcionamiento de estos módulos se describe a continuación:

Acondicionador de señal: La señal que se obtiene a la salida del entrenador tiene una componente continua que no es apropiada para su reproducción en unos auriculares. Además su nivel de amplitud es demasiado elevado (2 Vpp). Esta etapa elimina la tensión continua de la señal de entrada y atenúa su amplitud hasta 1 Vpp.

Amplificador de potencia: Permite amplificar la señal en corriente para poder escuchar los pulsos a través de unos auriculares externos. **Se prohíbe expresamente el uso de altavoces.**

Filtro paso banda: Este bloque tiene como misión limitar en banda la señal que pasa al rectificador, reduciendo de este modo la posible interferencia producida por señales de frecuencia diferente a los pulsos de 1 KHz. La ganancia de esta etapa eleva el nivel de señal a 6 Vpp.

Rectificador: Es un rectificador de media onda que elimina los semiciclos negativos de los pulsos de audio. Se trata básicamente de un diodo.

Filtro paso bajo: Atenúa la componente de 1 KHz de la señal resultante. Como consecuencia convierte cada pulso en una componente continua con flancos exponenciales y un pequeño rizado.

Comparador: Se trata de un operacional en bucle abierto que tiene la señal anterior en una de sus entradas y una tensión umbral (obtenida de un potenciómetro) en la otra. A la salida se obtiene una señal entre 0 y 5 V con flancos definidos.

4.1.2 Acondicionador de señal

Este circuito (Figura 12) adapta la salida del entrenador del laboratorio y atenúa la señal para obtener 1 Vpp a su salida. El conjunto formado por el condensador C y el potenciómetro conforman un filtro paso alto que elimina la componente continua procedente del entrenador. Además, el potenciómetro actúa como un divisor de tensión reduciendo el nivel de señal a la salida. El amplificador operacional en configuración de seguidor conforma un buffer o separador que permite aislar la impedancia de salida de esta etapa evitando afectar a la etapa siguiente.

En primer lugar diseñe el valor de C que hace que la frecuencia de corte del citado filtro paso alto ($f_c = \frac{1}{2\pi \cdot R \cdot C}$) sea menor de 20 Hz (límite audible inferior). Un valor entre 10 y 20 Hz es aceptable.

Ajuste a continuación el cursor del potenciómetro para conseguir un nivel de señal de **1Vpp** a la salida del operacional.

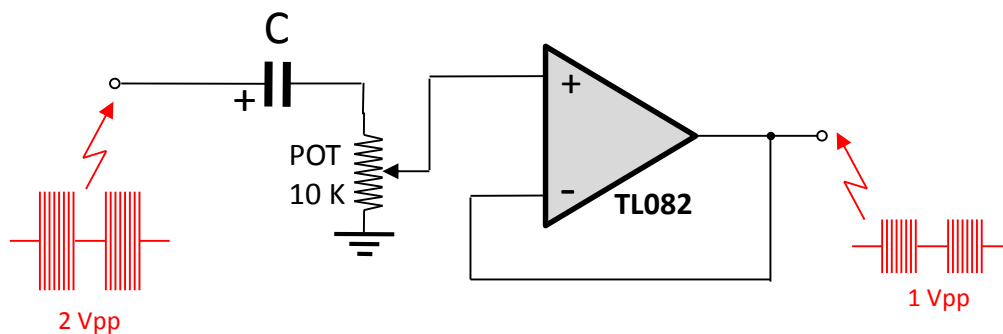


Figura 12 Filtro paso alto, atenuador y separador del acondicionador de señal.

4.1.3 Amplificador de potencia para auriculares

El amplificador de potencia (Figura 13) se realizará con un circuito integrado LM386. Este componente contiene un amplificador de audio de ganancia 20 junto con una etapa de potencia que puede suministrar la corriente suficiente para excitar unos auriculares. **Se prohíbe expresamente el uso de altavoces en el laboratorio.** Utilice un jack hembra para conectar a la placa de inserción y poder utilizar los auriculares. El potenciómetro de 10 kΩ se emplea para regular el volumen. **Se recomienda además colocar un condensador de 100 μF entre alimentación y masa** próximo a este circuito integrado (patillas 6 y 4) para evitar posibles oscilaciones.

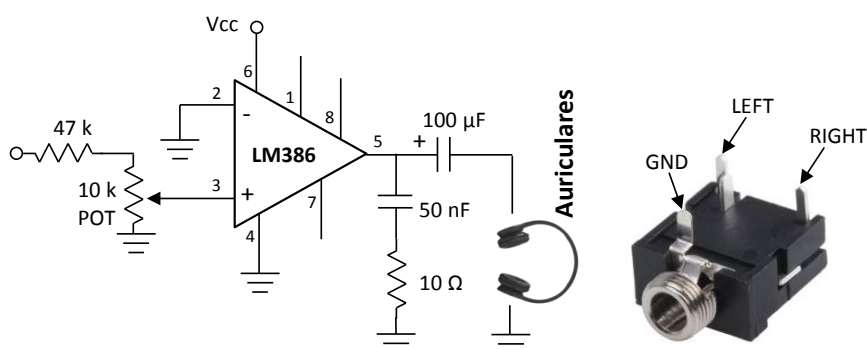


Figura 13 Amplificador de potencia basado en el integrado LM386 y patillaje del jack

4.1.4 Filtro paso banda

Vamos a emplear un **filtro paso banda** de realimentación múltiple (lea las referencias [4] o [5] para una descripción detallada). Este filtro se realiza mediante un amplificador operacional según el esquema que se muestra en la Figura 14.

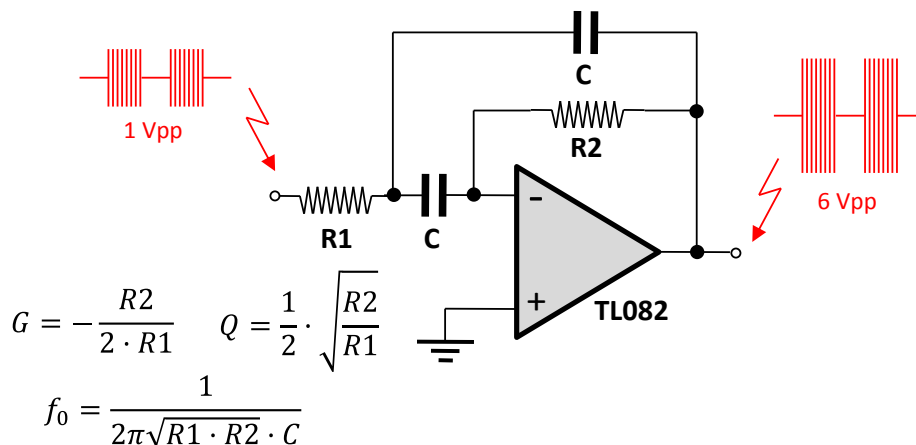


Figura 14 Filtro paso banda de realimentación múltiple de adaptación al canal.

Deberá diseñarse para que la frecuencia central f_0 sea igual a 1 KHz. G es el valor de ganancia en el centro de la banda de paso. El valor de Q indica la selectividad del filtro, de tal modo que el ancho de banda a 3 dB (B) cumple necesariamente la expresión: $B = f_0 / Q$.

Diseñe el filtro para obtener una ganancia $|G| = 6$. De esta manera el nivel de señal a la salida del mismo deberá ser de **6 Vpp**.

Este filtro deberá ser convenientemente caracterizado para la primera entrega (semana 5), tal y como se describe en el apartado 4.3.2

4.1.5 Rectificador

El elemento necesario para producir la rectificación de la señal es un diodo. Dado que a la salida del filtro paso banda tenemos un nivel de señal de 6 Vpp, esta tensión es suficiente para poner el diodo en su zona de conducción directa ($V > 0,6$ V). Se utilizará un diodo **1N4148** que tiene buena respuesta a la frecuencia de 1 KHz.

En la Figura 15 se muestra el esquema del rectificador. Observe que aparece una resistencia de carga a su salida indicada como R_L . Dicha resistencia cierra el bucle de corriente del diodo y es fundamental para el funcionamiento del circuito (un valor de 1 KΩ es suficiente).

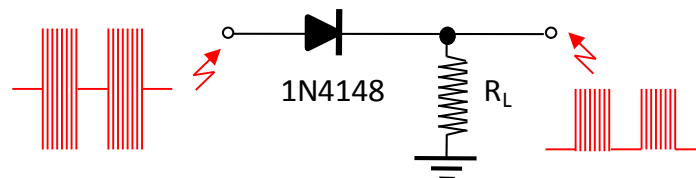


Figura 15 Rectificador construido con un diodo.

4.1.6 Filtro paso bajo

El filtro paso bajo se construirá según un esquema Sallen-Key de dos polos como el que se muestra en la Figura 16.

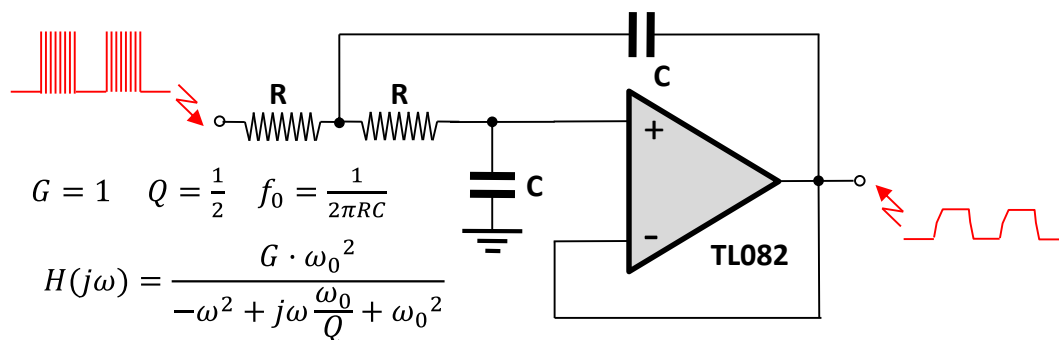


Figura 16 Filtro paso bajo Sallen-Key

Recuerde que este filtro se emplea para atenuar la señal senoidal de 1 kHz, dejando pasar la continua de los pulsos tras la rectificación. En un filtro de dos polos, la frecuencia de corte f_c (frecuencia donde la ganancia cae 3 dB respecto a la banda de paso) solamente coincide con f_0 ($\omega_0 = 2\pi \cdot f_0$) cuando $Q=1/\sqrt{2}$. No obstante, en este caso G y Q están fijados ($G=1$ y $Q=0,5$) y por tanto será necesario buscar el valor de ω_0 que nos permita colocar la frecuencia de corte f_c en el valor que deseemos.

Teniendo esto en cuenta, siga la siguiente secuencia para su diseño:

- 1.- Calcule en primer lugar el módulo de la función de transferencia $|H(j\omega)|$.
- 2.- Escoja un valor para la frecuencia de corte del filtro (f_c)
- 3.- Sustituya este valor en ω ($\omega = 2\pi \cdot f_c$), y calcule el valor de ω_0 que hace que la función de transferencia caiga 3 dB ($1/\sqrt{2} = 0,7$) respecto al valor en la banda de paso.

$$1/\sqrt{2} \cdot G = |H(j\omega)|$$

- 4.- Por último escoja valores para R y C que cumplan $\omega_0 = 1 / RC$.

Este filtro deberá ser convenientemente caracterizado para la segunda entrega (semana 8), tal y como se describe en el apartado 4.3.2

4.1.7 Comparador

El comparador nos permite obtener una señal digital a partir de la señal obtenida tras el filtro paso bajo. Es la etapa que separa físicamente la parte analógica de la parte digital.

Generalmente la señal que se obtiene tras el filtro no tiene amplitud suficiente para ser utilizada en un circuito digital. Además, los flancos exponenciales y el rizado no son apropiados para la aplicación que se propone en este proyecto. Es necesario obtener una señal digital cuadrada entre 0 y 5 V para conectarla a una de las entradas de la FPGA.

Para ello vamos a utilizar un comparador del tipo **LM311** (vea la Figura 17). Este circuito actúa como un operacional en bucle abierto, es decir, cuando la tensión en V+ supera a la tensión en V- entrega +Vcc a la salida. Cuando sucede lo contrario entrega 0V. **Es imprescindible colocar una resistencia de pull up de 1K entre Vcc y la salida del LM311 para que funcione correctamente.**

Se seleccionará una tensión umbral como la que se ve dibujada en el centro de la gráfica de la figura (línea negra). El comparador convertirá entonces la señal con flancos exponenciales en una señal digital con flancos definidos (en verde en la figura). La tensión umbral **debe obtenerla con un potenciómetro** conectado entre Vcc y masa (ver Figura 11).

Para establecer correctamente la posición del umbral tiene que visualizar simultáneamente en el osciloscopio la señal en las dos patillas de entrada del operacional, con la misma escala y el mismo nivel de 0 en ambos canales. Regule el potenciómetro para situar el umbral en el punto medio de las transiciones.

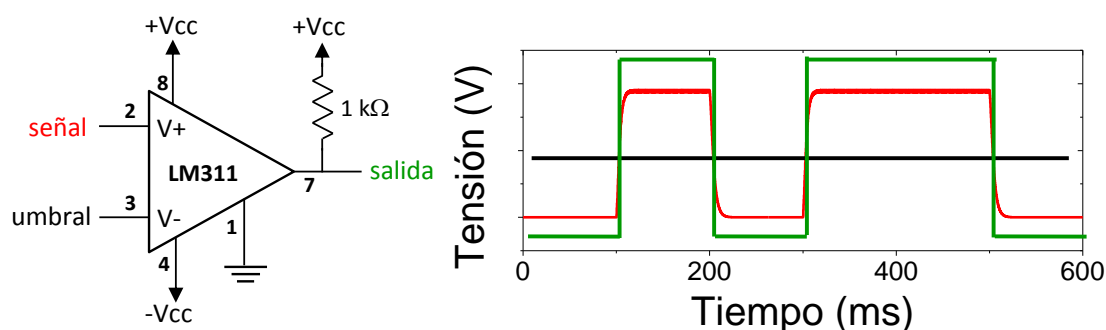


Figura 17 El comparador LM311 y esquema del proceso de comparación. ATENCIÓN: las señales de umbral/salida (negra/roja) NO están representadas en la misma escala que la señal de salida (verde).

4.2 SUBSISTEMA DIGITAL

Antes de comenzar a describir las partes que componen este sistema queremos insistir en una serie de cuestiones fundamentales para el correcto desarrollo del proyecto:

- Toda la lógica digital se describirá mediante VHDL y se sintetizará en la FPGA Spartan 3E presente en el laboratorio.
- La señal procedente del circuito analógico deberá conectarse a una de las entradas externas presentes en los conectores de la tabla de madera. **Sólo se permite la conexión de dos cables entre ambos subsistemas (analógico y digital): el cable de datos procedente del comparador y el cable de masa.**
- Tenga en cuenta que para entender este enunciado, así como el funcionamiento del sistema de desarrollo BASYS 2 y los elementos básicos del entorno de desarrollo ISE Webpack **deberá leer y entender el documento: “Manual de referencia de la tarjeta BASYS2”** [7]. Concretamente compile y sintetice los ejemplos que vienen en el citado manual, cuyo código se encuentra disponible en los ordenadores del laboratorio en la carpeta \BASYS2\ejemplos.

- Cree su proyecto dentro de una **carpeta creada en el Escritorio**. Llévase consigo todos los días el código que vaya generando (en un pendrive o a través de las distintas opciones por Internet). **Las carpetas se borrarán diariamente**.
- **No trabaje** directamente desde la unidad de memoria (pendrive). Copie los ficheros necesarios en la carpeta mencionada en el punto anterior.
- Escriba el **código respetando los nombres de las entidades, de las entradas y de las salidas** que se le suministran en este enunciado para cada módulo.
- Recuerde que en la tabla de madera que contiene los conectores de entrada y salida **hay que realizar una serie de conexiones** para generar la señal analógica en la salida del entrenador (ver Figura 10).

4.2.1 Estructura general del subsistema digital

El subsistema digital estará compuesto por dos partes:

- El módulo de generación de señal analógica suministrado y disponible en el MOODLE de la asignatura (archivo **gen_senal.vhd**).
- El circuito receptor (archivo **receptor.vhd**) que deberán realizar los alumnos.

Estos dos módulos vhd tendrán que estar siempre enlazados en un único archivo **main.vhd** que será el objetivo final de este proyecto. Junto con estos 2 módulos vhd también se necesitará el archivo de asociaciones (**asoc.ucf**) que también deberá compilarse con el código.

El módulo de generación de señal (**gen_senal.vhd**) tiene 3 entradas: el reloj de la FPGA (CLK) y los botones de START y STOP (BTN_START y BTN_STOP) y 3 salidas: (SPI_CS1, SPI_CLK y SPI_DIN). Estas últimas se emplean para controlar el convertidor DAC que genera la señal analógica, y del que no se darán detalles en este enunciado. **No se permite tocar ni modificar este archivo.**

El módulo del circuito receptor se realizará en archivos aparte y tendrá dos entradas: **el reloj de la FPGA (CLK)** y **la línea de entrada de datos (LIN)**. Como salidas tendrá las señales de control de los displays de la tarjeta BASYS2: **4 salidas para activar los displays (AN₃..AN₀)** y **7 salidas para controlar los segmentos de los displays (SEG₇..SEG₆)**.

La estructura de bloques del subsistema digital se muestra a continuación con más detalle en la Figura 18.

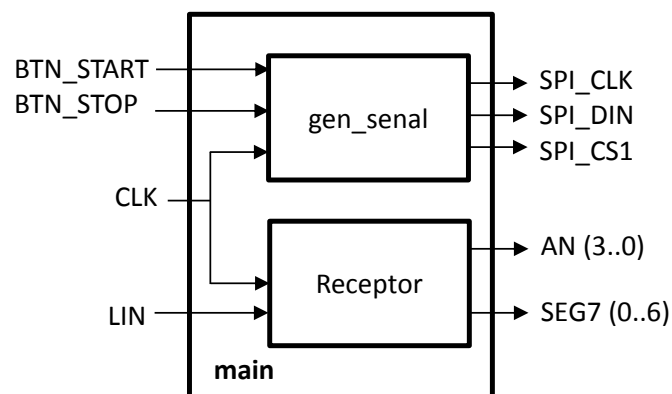


Figura 18 Esquema de bloques del subsistema digital completo

La descripciones en VHDL de las entidades de esta estructura se muestran a continuación y **deberán respetarse en el circuito final de este proyecto.**

ARCHIVO gen_senal.vhd

```
entity gen_senal is
    Port ( CLK      : in  STD_LOGIC;
          BTN_START : in  STD_LOGIC;
          BTN_STOP  : in  STD_LOGIC;
          SPI_CLK   : out  STD_LOGIC;
          SPI_DIN   : out  STD_LOGIC;
          SPI_CS1   : out  STD_LOGIC);
end gen_senal;
```

ARCHIVO receptor.vhd

```
entity receptor is
    Port ( CLK : in  STD_LOGIC; -- reloj de la FPGA
          LIN  : in  STD_LOGIC; -- Línea de entrada de datos
          AN   : out STD_LOGIC_VECTOR (3 downto 0); -- Activación individual
          SEG7 : out  STD_LOGIC_VECTOR (0 to 6)); -- Salida para los displays
end receptor;
```

ARCHIVO main.vhd

```
entity main is
    Port ( CLK      : in  STD_LOGIC;
          BTN_START : in  STD_LOGIC;
          BTN_STOP  : in  STD_LOGIC;
          SPI_CLK   : out  STD_LOGIC;
          SPI_DIN   : out  STD_LOGIC;
          SPI_CS1   : out  STD_LOGIC;
          LIN  : in  STD_LOGIC; -- Línea de entrada de datos
          AN   : out STD_LOGIC_VECTOR (3 downto 0); -- Activación individual displays
          SEG7 : out  STD_LOGIC_VECTOR (0 to 6)); -- Salida para los displays
end main;

architecture a_main of main is

    component gen_senal
        Port ( CLK      : in  STD_LOGIC;
              BTN_START : in  STD_LOGIC;
              BTN_STOP  : in  STD_LOGIC;
              SPI_CLK   : out  STD_LOGIC;
              SPI_DIN   : out  STD_LOGIC;
              SPI_CS1   : out  STD_LOGIC);
    end component;

    component receptor
        Port ( CLK : in  STD_LOGIC; -- reloj de la FPGA
              LIN  : in  STD_LOGIC; -- Línea de entrada de datos
              AN   : out STD_LOGIC_VECTOR (3 downto 0); -- Activación individual
              SEG7 : out  STD_LOGIC_VECTOR (0 to 6)); -- Salida para los displays
    end component;

    begin

    U1 : gen_senal port map
        (CLK => CLK,
         BTN_START => BTN_START,
         BTN_STOP  => BTN_STOP,
         SPI_CLK   => SPI_CLK,
         SPI_DIN   => SPI_DIN,
         SPI_CS1   => SPI_CS1);

    U2 : receptor port map
        (CLK => CLK,
         LIN  => LIN,
         AN   => AN,
         SEG7 => SEG7);

    end a_main;
```

El módulo **main.vhd** será el que se compile junto con el archivo de asociaciones (**asoc.ucf**) para obtener el bitstream que configura la FPGA.

4.2.2 Estructura del receptor

Centrándonos ya en el desarrollo del receptor pasaremos a analizar en detalle su estructura. Este circuito está compuesto por los bloques que se muestran a continuación en la Figura 19.

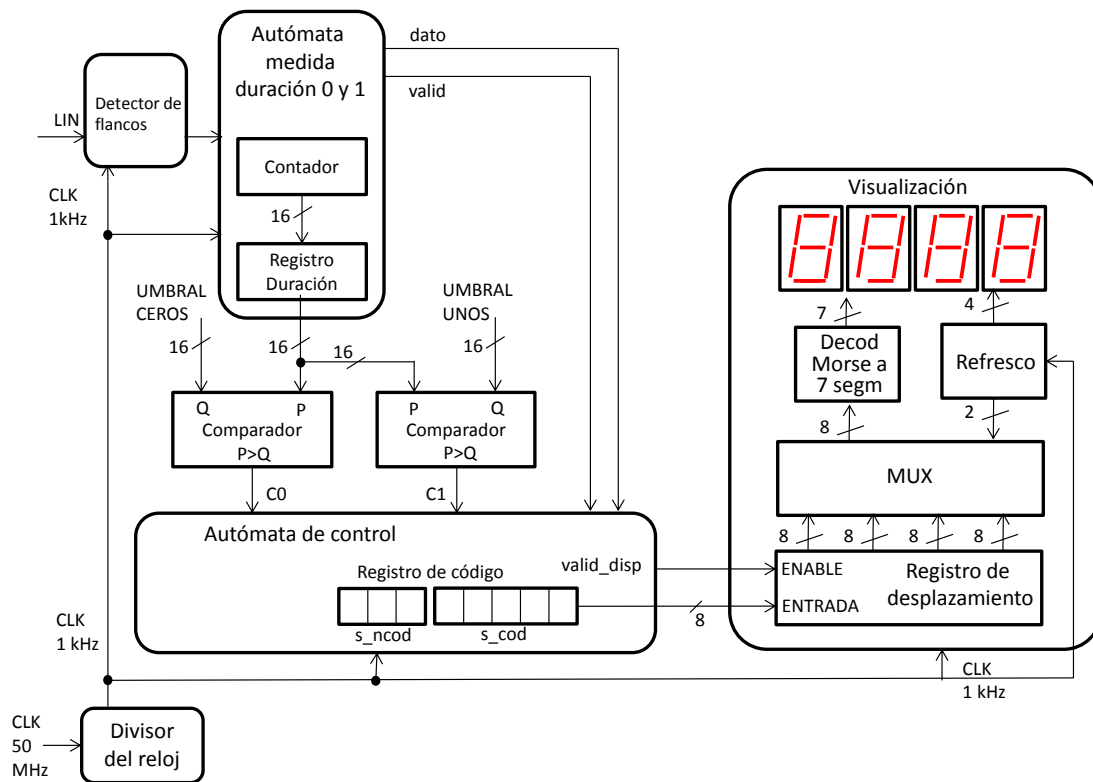


Figura 19 Estructura del receptor con sus bloques fundamentales

La función de cada bloque es la siguiente:

Divisor del reloj: Toda la lógica interna del receptor estará sincronizada con un reloj de 1 KHz de frecuencia. Por tanto es necesario desarrollar un divisor de frecuencia que obtenga un reloj de 1 KHz a partir del reloj de la FPGA (50 MHz).

Detector de flancos: Permite detectar con precisión los cambios entre 0 y 1 (y entre 1 y 0), que son los flancos de subida y de bajada de la señal de entrada. Este módulo elimina posibles glitches y rebotes en la conmutación.

Autómata de medida de duración: Tomando como referencia los flancos de la señal de entrada, este autómata mide la duración en ms de los intervalos a 0 y a 1 contando los ciclos de reloj que transcurren entre cada dos flancos, indicando si se trata de un intervalo a '1' ó a '0' (señal **DATO**) y validando dicho intervalo al llegar el siguiente flanco (mediante la señal **VALID**). Si la línea está a 0 durante más de 600 ms devuelve este valor como duración.

Comparadores: Son circuitos combinacionales que comparan la duración de un intervalo con dos umbrales. Las salidas de estos circuitos (**C0** y **C1**) junto con la señal **DATO** (del autómata de medida) permiten determinar si los intervalos definen una PAUSA, un ESPACIO, un PUNTO o una RAYA.

Autómata de control: Este circuito toma la información de los comparadores y del autómata de medida. Acumula los diferentes símbolos y determina el código que debe representarse en los displays cada vez que aparece un ESPACIO.

Visualización: Este módulo toma el código del carácter a representar, desplaza todos los displays una posición hacia la izquierda y coloca el carácter recibido en el display de la derecha.

4.2.3 Representación binaria del morse y caracteres asociados en los displays

Debemos buscar una forma de representar el código morse mediante un código digital. En este proyecto utilizaremos una representación con 8 bits para el código asociado a cada carácter. Los 3 bits más significativos indican el número de símbolos que componen el código, mientras que los 5 últimos representan el código, siendo 0 el PUNTO y 1 la RAYA (alineados a la izquierda). Por ejemplo:

Carácter	Código Morse	Representación binaria
A	.-	010 01000
B	-...	100 10000
3	...--	101 00011

También debemos asociar una visualización en los displays para cada uno de los caracteres alfabéticos y numéricos. Si bien esto no es posible para todos los caracteres en un display de 7 segmentos, en este proyecto utilizaremos la representación mostrada en la Figura 20. Observe que la representación de los caracteres: K, M, Q, V, W, X, Z es un display en blanco (apagado). La S y el 5 comparten la misma representación.

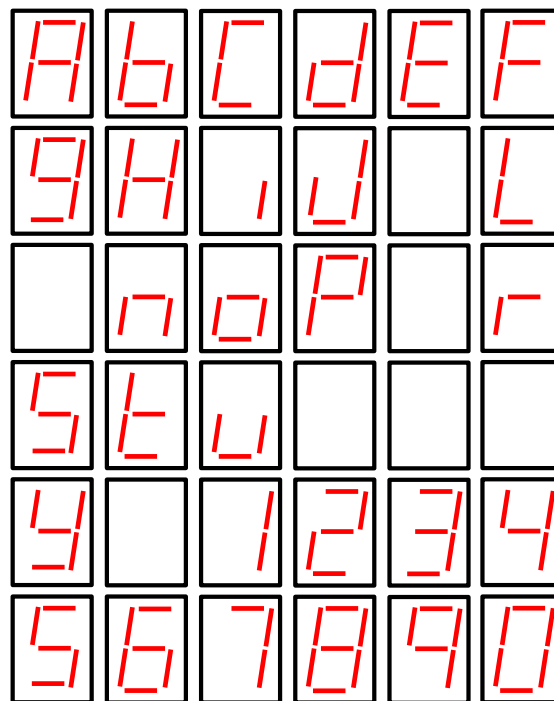


Figura 20 Representación en los displays de todos los caracteres

4.2.4 Funcionamiento del receptor

Los flancos de la señal de entrada son filtrados por el detector de flancos para evitar posibles glitches y rebotes. Estos flancos determinan la duración en el tiempo de los ceros y los unos. El autómata de medida de duración mide el número de ciclos de reloj de 1 ms entre cada dos flancos y lo coloca en la salida **DURACION**, validando dicha salida con la señal **VALID** (activa durante un ciclo de reloj). La salida **DATO** contiene el valor digital del dato medido (0 o 1).

La duración es entonces comparada con dos umbrales ($DURACION > UMBRAL$). Dado que en este proyecto los PUNTOS y las PAUSAS duran **100 ms**, mientras que las RAYAS y los ESPACIOS duran **300 ms** el valor de ambos umbrales será de **200**. El resultado de esta comparación genera las señales C0 y C1, de manera que:

Para **DATO='0'** Si **C0** = 0 se trata de una PAUSA, y si **C0** = 1 se trata de un ESPACIO,

Para **DATO='1'**; Si **C1** = 0 se trata de un PUNTO, y si **C1** = 1 se trata de una RAYA.

A continuación se genera el código en el autómata de control. Este autómata tiene las siguientes entradas y salidas:

ENTRADAS : **DATO**, **VALID**, **C0** y **C1**

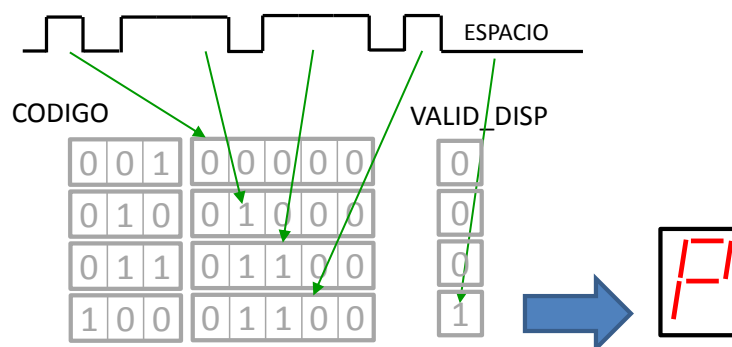
SALIDAS : **CODIGO** (a visualizar en el display) y **VALID_DISP** (validación del display).

El autómata de control se especifica con más detalle en el apartado 4.2.10. El funcionamiento es el siguiente (los datos llegan cuando se activa la señal **VALID**).

Comienza en el estado de RESET

1. RESET. Pone **n=4** y **CODIGO**="00000000". Si llega un SIMBOLO (**DATO=1** y **VALID=1**) pasa al punto 2. En otro caso se queda en el 1.
2. Almacena el símbolo (**C1** indica punto o raya) en la posición **n** de **CODIGO**, incrementa el valor de los tres bits más significativos de CODIGO y decrementa **n** en una unidad. Pasa al punto 3.
3. Si llega un SIMBOLO (**DATO=1** y **VALID=1**) pasa al punto 2. Si llega un ESPACIO (**DATO=0**, **C0=1** y **VALID=1**) pasa al punto 4. En cualquier otro caso se queda en el 3.
4. Ha llegado un ESPACIO. Valida el display activando **VALID_DISP**. Pasa al punto 1.

De este modo, los símbolos que van llegando se van acumulando en la señal **CODIGO** y cuando llega el ESPACIO se valida el display (activándose la señal **VALID_DISP**) apareciendo el código representado.



4.2.5 Divisor del reloj

El divisor del reloj tiene como entrada el reloj de la FPGA (50 MHz) y como salida debe generar una señal cuadrada de 1 KHz. Esto se lleva a cabo mediante división de frecuencia con un contador y una señal que cambia de valor lógico cuando la cuenta alcanza un valor determinado.



ARCHIVO div_reloj.vhd

```

entity div_reloj is
    Port ( CLK      : in  STD_LOGIC;      -- Entrada reloj de la FPGA 50 MHz
          CLK_1ms   : out STD_LOGIC);    -- Salida reloj a 1 KHz
end div_reloj;

architecture a_div_reloj of div_reloj is

    signal contador : STD_LOGIC_VECTOR (15 downto 0);
    signal flag : STD_LOGIC;

begin

    process (CLK)
    begin
        if (CLK'event and CLK='1') then
            contador<=contador+1;
            if (contador=XXXXXX) then
                contador<=(others=>'0');
                flag<=not flag;
            end if;
        end if;
    end process;

    CLK_1ms<=flag;

end a_div_reloj;
  
```

Calcule el valor de **XXXXXX** para que el módulo realice la división de frecuencia adecuada.

Para comprobar que este módulo funciona correctamente asocie la salida CLK_1ms a un terminal de salida de la FPGA y mida con el osciloscopio la frecuencia obtenida. Por ejemplo puede utilizar las siguientes asociaciones:

```

# Reloj principal del sistema

NET "CLK" LOC = "M6"; #Señal de reloj del sistema

#Salida externa donde se visualiza la señal de reloj dividida

NET "CLK_1ms" LOC = "A13"; #Salida terminal 19
  
```


4.2.6 Descripción del módulo de visualización:

El módulo de visualización será la primera tarea que deberá llevar a cabo en este proyecto. Consiste en un módulo capaz de representar un carácter en el display de la derecha, desplazando todos los demás una posición hacia la izquierda. El código a representar se debe presentar en la ENTRADA y será capturado por el registro de desplazamiento en el flanco activo del reloj cuando la señal ENABLE se active (a nivel alto). El esquema general de este módulo se puede ver en la Figura 21.

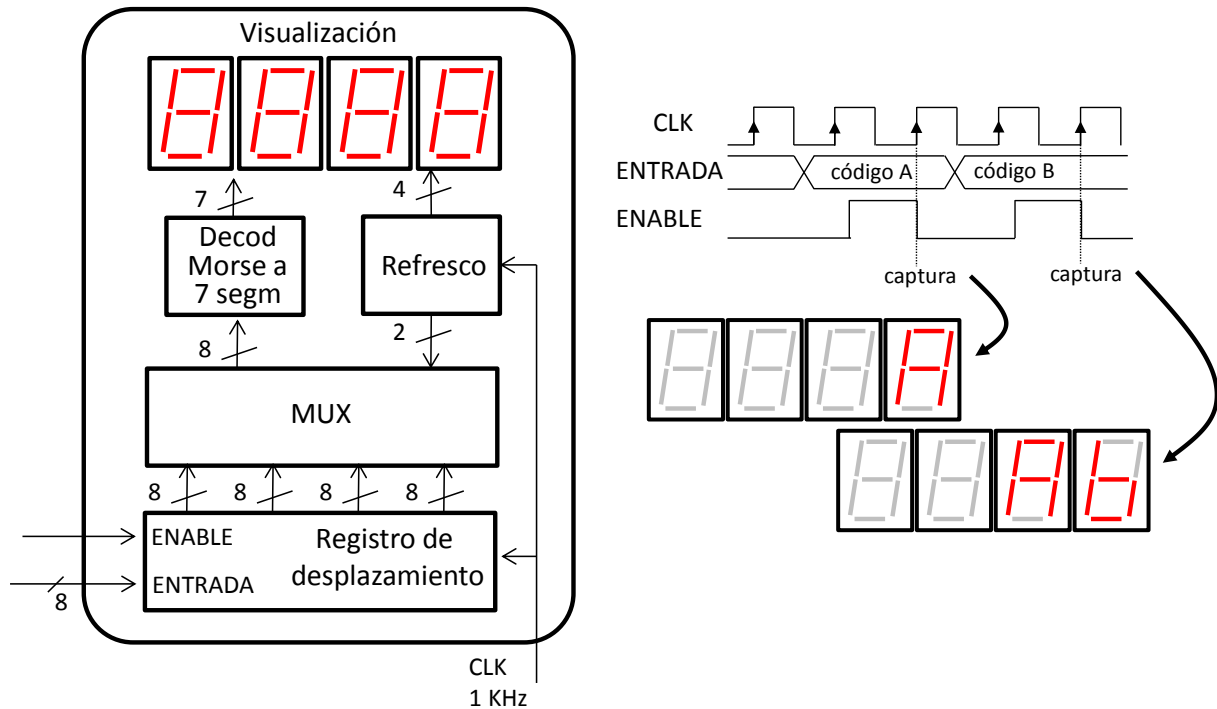


Figura 21 Diagrama de bloques del módulo de visualización y cronograma de su funcionamiento.

4.2.6.1 Visualización: Decodificador de Morse a 7 segmentos

Para visualizar la imagen correspondiente a cada código debe crearse un decodificador que asocie un código morse expresado con 8 bits a un vector de 7 bits que indique los segmentos que deben encenderse. **Este módulo se proporciona ya codificado** para ahorrar tiempo y evitar errores, ya que no supone mayor dificultad. Descargue el fichero **decodmorsea7s.vhd** que contiene dicho decodificador.

Las entradas y salidas asociadas al mismo se resumen a continuación:

ARCHIVO **decodmorsea7s.vhd**

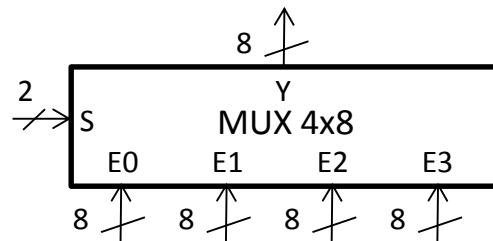
```
entity decodmorsea7s is
  Port ( SIMBOLO    : in STD_LOGIC_VECTOR (7 downto 0); -- código morse
        SEGMENTOS  : out STD_LOGIC_VECTOR (0 to 6)); -- segmentos display
end decodmorsea7s;
```

4.2.6.2 Visualización: Multiplexor

Este es el elemento que permite visualizar caracteres diferentes en cada display. Tenga en cuenta que los 4 displays presentes en la BASYS2 comparten las mismas líneas CA,CB,CC,CD,CE,CF,CG, aunque poseen señales de activación diferentes. Por tanto para visualizar cifras diferentes en cada uno, es necesario presentar cada una de ellas por separado de forma muy rápida para que el ojo no

perciba el parpadeo. (Remitimos al lector a los ejemplos 3 y 4 del “manual de referencia de la tarjeta BASYS 2” donde se puede encontrar una descripción detallada).

En este caso necesitamos un multiplexor de 4 entradas de 8 bits y una salida también de 8 bits. El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación. Complete su descripción funcional.

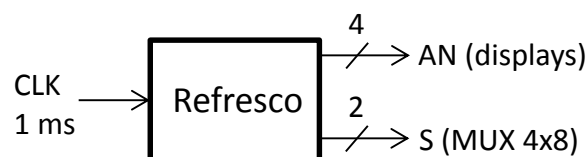


ARCHIVO MUX4x8.vhd

```
entity MUX4x8 is
    Port ( E0 : in  STD_LOGIC_VECTOR (7 downto 0); -- Entrada 0
          E1 : in  STD_LOGIC_VECTOR (7 downto 0); -- Entrada 1
          E2 : in  STD_LOGIC_VECTOR (7 downto 0); -- Entrada 2
          E3 : in  STD_LOGIC_VECTOR (7 downto 0); -- Entrada 3
          S : in  STD_LOGIC_VECTOR (1 downto 0);  -- Señal de control
          Y : out STD_LOGIC_VECTOR (7 downto 0)); -- Salida
end MUX4x8;
```

4.2.6.3 Visualización: Refresco

Este bloque controla la visualización sucesiva de cada carácter en cada display. Para ello va alternando las entradas del multiplexor con la activación del display correspondiente. Como entrada toma la señal de reloj de 1 KHz y como salidas: la entrada de selección del multiplexor y las entradas de activación de los displays. El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación. Complete su descripción funcional (fíjese en los ejemplos 3 y 4 del “manual de referencia de la tarjeta BASYS 2”).



ARCHIVO refresco.vhd

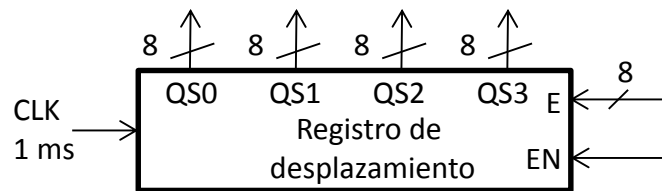
```
entity refresco is
    Port ( CLK_1ms : in STD_LOGIC; -- reloj de refresco
          S : out STD_LOGIC_VECTOR (1 downto 0); -- Control para el mux
          AN : out STD_LOGIC_VECTOR (3 downto 0)); -- Control displays
end refresco;
```

4.2.6.4 Visualización: Registro de desplazamiento

El registro de desplazamiento es el que permite que los caracteres se desplacen hacia la izquierda cada vez que llega un nuevo carácter. Se trata de un registro con entrada serie y salida paralelo de 4 posiciones de 8 bits. Tiene una entrada de enable que lo activa. Cuando ENABLE='0' el registro se

encuentra en modo inactivo. Cuando ENABLE='1' el registro desplaza los datos hacia la izquierda con el flanco del reloj y almacena el dato de entrada en la salida QS3.

El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación. Complete su descripción funcional.



ARCHIVO rdesp_disp.vhd

```
entity rdesp_disp is
  Port( CLK_1ms : in  STD_LOGIC;           -- entrada de reloj
        EN      : in  STD_LOGIC;           -- enable
        E       : in  STD_LOGIC_VECTOR (7 downto 0); -- entrada de datos
        Q0      : out STD_LOGIC_VECTOR (7 downto 0); -- salida Q0
        Q1      : out STD_LOGIC_VECTOR (7 downto 0); -- salida Q1
        Q2      : out STD_LOGIC_VECTOR (7 downto 0); -- salida Q2
        Q3      : out STD_LOGIC_VECTOR (7 downto 0)); -- salida Q3
end rdesp_disp
```

4.2.6.5 Visualización: Cableado final y pruebas

Finalmente es necesario cablear todos los elementos anteriores en un solo módulo. Para ello realizaremos un código VHDL con una descripción arquitectural de interconexión. Deberán declararse todos los módulos anteriores como **component** y realizar el cableado entre ellos.

A continuación se muestra parte del código VHDL que deberá escribir. Complete la descripción arquitectural de interconexión de la forma adecuada según la Figura 21.

ARCHIVO visualizacion.vhd

```
entity visualizacion is
  Port ( E0 : in STD_LOGIC_VECTOR (7 downto 0); -- Entrada sig. carácter
        EN : in STD_LOGIC;           -- Activación para desplazamiento
        CLK_1ms : in STD_LOGIC;       -- Entrada de reloj
        SEG7 : out STD_LOGIC_VECTOR (0 to 6); -- Segmentos displays
        AN   : out STD_LOGIC_VECTOR (3 downto 0)); -- Activación displays
end visualizacion;

architecture a_visualizacion of visualizacion is

  component MUX4x8
    Port (
      E0 : in  STD_LOGIC_VECTOR (7 downto 0); -- Entrada 0
      E1 : in  STD_LOGIC_VECTOR (7 downto 0); -- Entrada 1
      E2 : in  STD_LOGIC_VECTOR (7 downto 0); -- Entrada 2
      E3 : in  STD_LOGIC_VECTOR (7 downto 0); -- Entrada 3
      S  : in  STD_LOGIC_VECTOR (1 downto 0); -- Señal de control
      Y  : out STD_LOGIC_VECTOR (7 downto 0)); -- Salida
  end component;

  -- Cableado de los componentes
  -- (Este código se completará según la Figura 21)
```

```

component decodmorsea7s
  Port ( SIMBOLO   : in  STD_LOGIC_VECTOR (7 downto 0);
        SEGMENTOS : out  STD_LOGIC_VECTOR (0 to 6));
end component;

component refresco
  Port ( CLK_lms : in  STD_LOGIC; -- reloj
        S       : out  STD_LOGIC_VECTOR (1 downto 0); -- Control para el mux
        AN      : out  STD_LOGIC_VECTOR (3 downto 0)); -- Control displays
end component;

component rdesp_disp
  Port ( CLK_lms : in  STD_LOGIC; -- entrada de reloj
        EN      : in  STD_LOGIC; -- enable
        E       : in  STD_LOGIC_VECTOR (7 downto 0); -- entrada de datos
        Q0      : out  STD_LOGIC_VECTOR (7 downto 0); -- salida Q0
        Q1      : out  STD_LOGIC_VECTOR (7 downto 0); -- salida Q1
        Q2      : out  STD_LOGIC_VECTOR (7 downto 0); -- salida Q2
        Q3      : out  STD_LOGIC_VECTOR (7 downto 0)); -- salida Q3
end component;

. . . POSIBLES SEÑALES NECESARIAS

begin

. . . . . COMPLETE AQUÍ LA DESCRIPCIÓN

end a_visualizacion;

```

Pruebas del módulo de visualización:

Para probar el módulo de visualización puede descargar el archivo: **prueba_visualizacion.vhd**. Este archivo, junto con los generados por el alumno permiten generar una secuencia donde aparecen secuencialmente todos los caracteres en los displays. Cargue también el archivo de asociaciones: **asoc_prueba_vis.ucf** y genere el bitstream. Al cargarlo sobre la FPGA deberán aparecer en los displays todos los caracteres sucesivamente, primero las letras y después los números, tal y como se describen en la sección 4.2.3 (Figura 20). Note que algunos de los caracteres no tienen representación.

4.2.7 Detector de flancos

El detector de flancos toma como entrada la línea de datos (LIN) y el reloj, y genera una salida (VALOR) 0 o 1 dependiendo de si el flanco que llega es de bajada o de subida respectivamente. Para ello muestrea constantemente la señal de entrada acumulando las muestras dentro de un registro de desplazamiento en sincronía con el reloj de 1 KHz de modo que siempre haya 20 bits. Además calcula constantemente la suma de los 20 bits que están dentro del registro. Si la suma supera el umbral de 15 interpreta que llega un flanco de subida y la salida (VALOR) toma el valor 1, si la suma pasa por debajo del umbral de 5 interpreta que llega un flanco de bajada y la salida (VALOR) toma el valor '0'. En otro caso la salida no cambia y mantiene su valor. Este detector se comporta como un autómata que implementa un comparador con histéresis. El diagrama de estados correspondiente a este autómata, así como la curva de histéresis se muestran en la Figura 22. Este procedimiento

permite ignorar los flancos correspondientes a glitches o los múltiples flancos en el caso de que aparezcan rebotes.

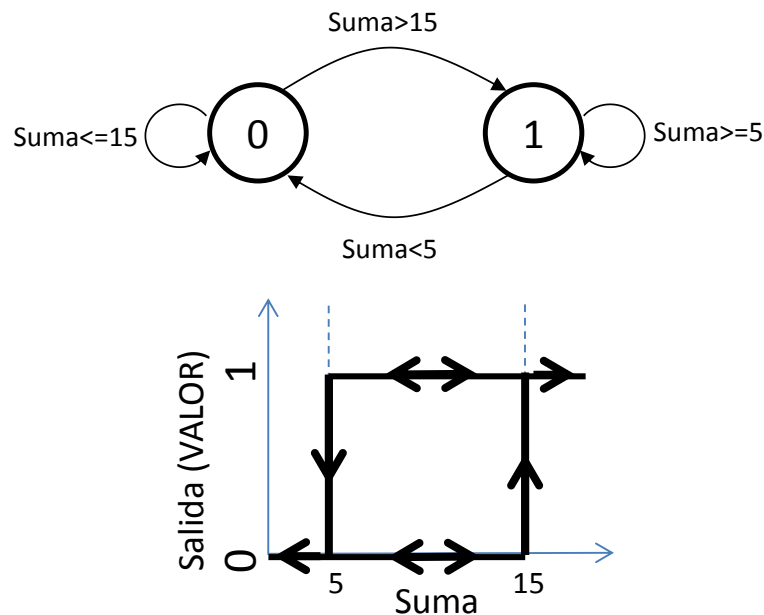


Figura 22 Diagrama de estados del detector de flanco. Se comporta como un comparador con histéresis

A continuación, la Figura 23 ilustra este hecho con dos ejemplos. En el caso a) la salida (VALOR) está a 0 y los glitches van desplazándose dentro del registro haciendo que la suma de los datos (2) sea siempre menor que el umbral del 1 (15), por lo tanto el autómata permanece en el estado 0 ignorándolos. En el caso b) aparece un flanco con rebotes. Mientras los datos se desplazan y la suma no alcanza el umbral de 15, la salida (VALOR) sigue estando a 0, pero cuando la señal de entrada es estable tras los rebotes, la suma alcanza este umbral, y entonces la salida (VALOR) conmuta a 1. De este modo se pueden detectar flancos y proporcionar una señal de salida limpia y con flancos definidos¹.

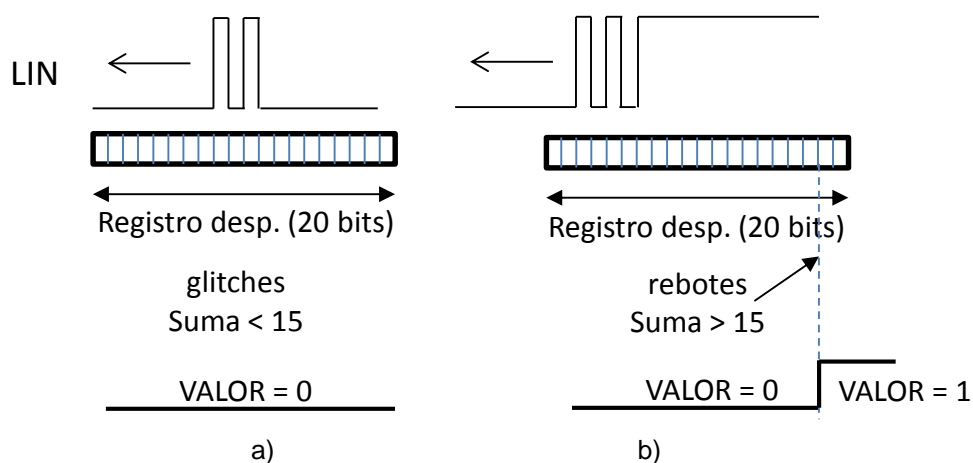
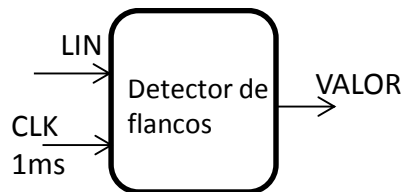


Figura 23 Muestreo constante de la señal en el tiempo. a) Aparecen glitches, pero la suma acumulada no alcanza el umbral y la salida (VALOR) se mantiene a 0 ignorando el glitch. b) En un flanco con rebotes, solamente cuando la suma alcanza el umbral es cuando la salida (VALOR) toma el valor 1 eliminando los rebotes.

¹ Obviamente este sistema tiene sus limitaciones. Si el número de glitches es tan alto que la suma supera el umbral, el autómata dará una salida de 1 cuando en realidad no hay un flanco.

El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación. Complete su descripción funcional según se describe en los comentarios incluidos.



ARCHIVO detector_flanco.vhd

```

entity detector_flanco is
    Port ( CLK_1ms : in  STD_LOGIC;    -- reloj
           LIN     : in  STD_LOGIC;    -- Línea de datos
           VALOR    : out STD_LOGIC);  -- Valor detectado en el flanco
end detector_flanco;

architecture a_detector_flanco of detector_flanco is

    constant UMBRAL0 : STD_LOGIC_VECTOR (7 downto 0) := "00000101"; -- 5 umbral para el 0
    constant UMBRAL1 : STD_LOGIC_VECTOR (7 downto 0) := "00001111"; -- 15 umbral para el 1

    signal reg_desp : STD_LOGIC_VECTOR (19 downto 0) := "00000000000000000000";
    signal suma     : STD_LOGIC_VECTOR (7 downto 0) := "00000000";
    signal s_valor  : STD_LOGIC;

begin

    process (CLK_1ms)
    begin
        if (CLK_1ms'event and CLK_1ms='1') then
            -- Calcular la suma de los elementos del registro
            -- (Para ello suma la nueva muestra que llega y reste la última que sale)
            -- Desplazar los datos del registro capturando la nueva muestra en el registro
            -- Ver si la suma supera los umbrales y asignar a s_valor el valor adecuado
        end if;
    end process;

    VALOR<=s_valor;

end a_detector_flanco;
  
```

Prueba del detector de flancos:

Para probar el funcionamiento de este módulo descargue el archivo de test_bench: **test_detector_flanco.vhd** y realice la simulación. Tenga en cuenta que el simulador, por defecto, solamente simula 1 us. Escriba en la consola del simulador ISIM el comando **run 2us** para simular más tiempo. Este archivo genera varios glitches y rebotes en la señal de entrada. Si el módulo funciona correctamente deberá observar un único flanco de subida en la señal VALOR en 715 ns y un único flanco de bajada en 1.415 ns.

4.2.8 Autómata de medida de duración de 0s y 1s

Este autómata toma como entrada la salida del detector de flancos y genera varias salidas: DURACIÓN (registro de 16 bits que almacena el número de ciclos de reloj contados), DATO (valor binario que indica si se trata de un intervalo a 0 o a 1) y VALID (señal de validación que indica que el intervalo en cuestión ha terminado (al llegar un flanco) y por lo tanto que la cuenta ha terminado. Esta señal dura un ciclo de reloj).

El autómata cuenta los ciclos de reloj transcurridos entre dos flancos consecutivos. Para ello debe definirse un contador interno (**cont**) además de un registro para almacenar el resultado final (**reg**). Ambos serán declarados como *signal*. Si la señal de entrada se mantiene a 0 durante más de 600 ms, el autómata activa la señal **VALID** y devuelve este valor. De este modo se evita que el sistema se quede en un bucle infinito al final de un mensaje. Por lo demás el autómata pasará por una serie de estados los cuales se muestran a continuación en la Figura 24.

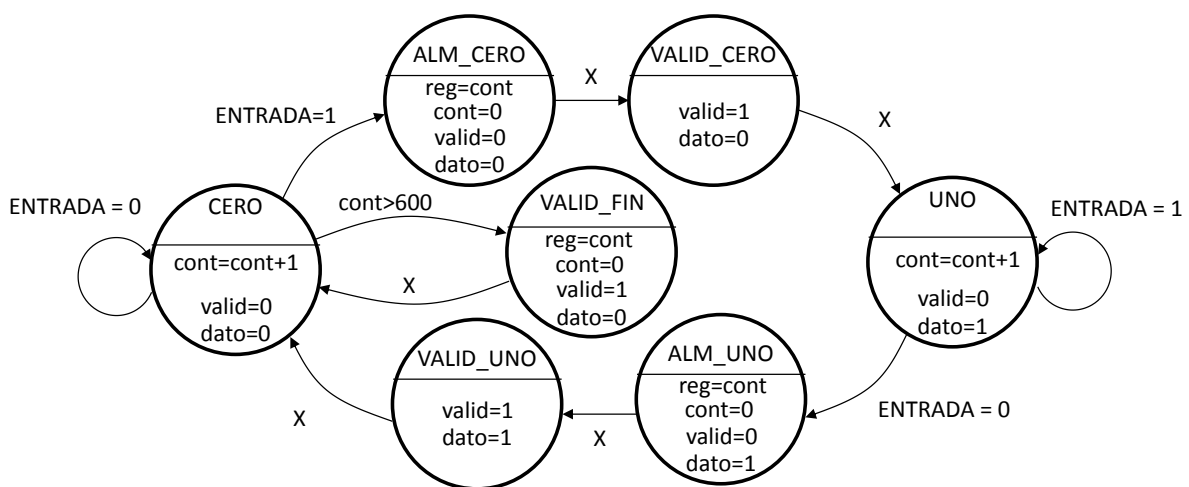
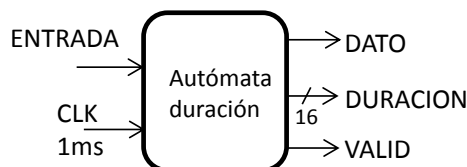


Figura 24 Diagrama de estados del autómata de medida de la duración de 0s y 1s

El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación. Complete su descripción funcional según se describe en los comentarios incluidos. **Realice el autómata con un solo proceso.** Asigne el valor del contador (**cont**) y del registro (**reg**) dentro de cada estado. **Los valores de las salidas deberán cablearse fuera del proceso.** Como puede ver, parte del código se muestra como ejemplo.



ARCHIVO aut_duracion.vhd

```

entity aut_duracion is
  Port ( CLK_1ms : in  STD_LOGIC;    -- reloj de 1 ms
        ENTRADA  : in  STD_LOGIC;    -- línea de entrada de datos
        VALID    : out STD_LOGIC;    -- salida de validación de dato
        DATO     : out STD_LOGIC;    -- salida de dato (0 o 1)
        DURACION : out STD_LOGIC_VECTOR (15 downto 0)); -- salida de duración del dato
end aut_duracion;
  
```

```

architecture a_aut_duracion of aut_duracion is

type STATE_TYPE is (CERO,ALM_CERO,VALID_CERO,UNO,ALM_UNO,VALID_UNO,VALID_FIN);
signal ST : STATE_TYPE := CERO;
signal cont : STD_LOGIC_VECTOR (15 downto 0):="0000000000000000";
signal reg : STD_LOGIC_VECTOR (15 downto 0) :="0000000000000000";

begin

process (CLK_1ms) -- autómeta
begin
if (CLK_1ms'event and CLK_1ms='1') then
case ST is
when CERO =>
cont<=cont+1;
if (ENTRADA='0') then
ST<=CERO;
else
ST<=ALM_CERO;
end if;

. . . OTROS ESTADOS

end case;
end if;
end process;

-- PARTE COMBINACIONAL

VALID<='1' when (ST=VALID_CERO or ST=VALID_UNO or ST=VALID_FIN) else '0';
DATO <= . . . COMPLETE LA DESCRIPCION;
DURACION<= . . . COMPLETE LA DESCRIPCION;

end a_aut_duracion;

```

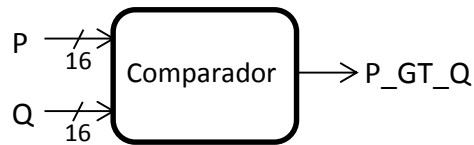
Prueba del autómeta de medida de duración:

Para probar el funcionamiento de este módulo descargue el archivo de test_bench: **test_aut_duracion.vhd** y realice la simulación. En la consola del ISIM escriba **run 2 us** para ampliar el tiempo de simulación. Aparecerán los valores de todas las entradas y las salidas. Si todo funciona correctamente deberá observar al término de cada periodo (a 0 o a 1) el valor medido en ciclos de reloj como resultado de la salida DURACION. En este caso aparecen los valores: 0,101, 298, 98, 298, 98, 602. Además aparecerán pulsos de duración 1 ciclo de reloj en la señal VALID cuyos flancos de subida están en los tiempos: 101,5; 401,5; 501,5; 601,5; 901,5; 1201,5; 1301,5; 1401,5; 2004,5 y 2607,5 ns. La salida DATO debe coincidir con la entrada.

4.2.9 Comparadores

Los comparadores son circuitos combinatoriales que comparan los valores binarios de sus dos entradas de 16 bits (P y Q) y generan una salida P>Q activa a nivel alto.

El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación. Complete su descripción funcional.



ARCHIVO comp_16.vhd

```
entity comp_16 is
    Port ( P : in  STD_LOGIC_VECTOR (15 downto 0); -- Entrada P
          Q : in  STD_LOGIC_VECTOR (15 downto 0); -- Entrada Q
          P_GT_Q : out  STD_LOGIC);               -- Salida P > Q
end comp_16;
```

4.2.10 Autómata de control

Este autómata es el encargado de componer el código y validar su visualización en el display cuando se recibe un ESPACIO. Su organización en estados responde al diagrama que se muestra en la Figura 25.

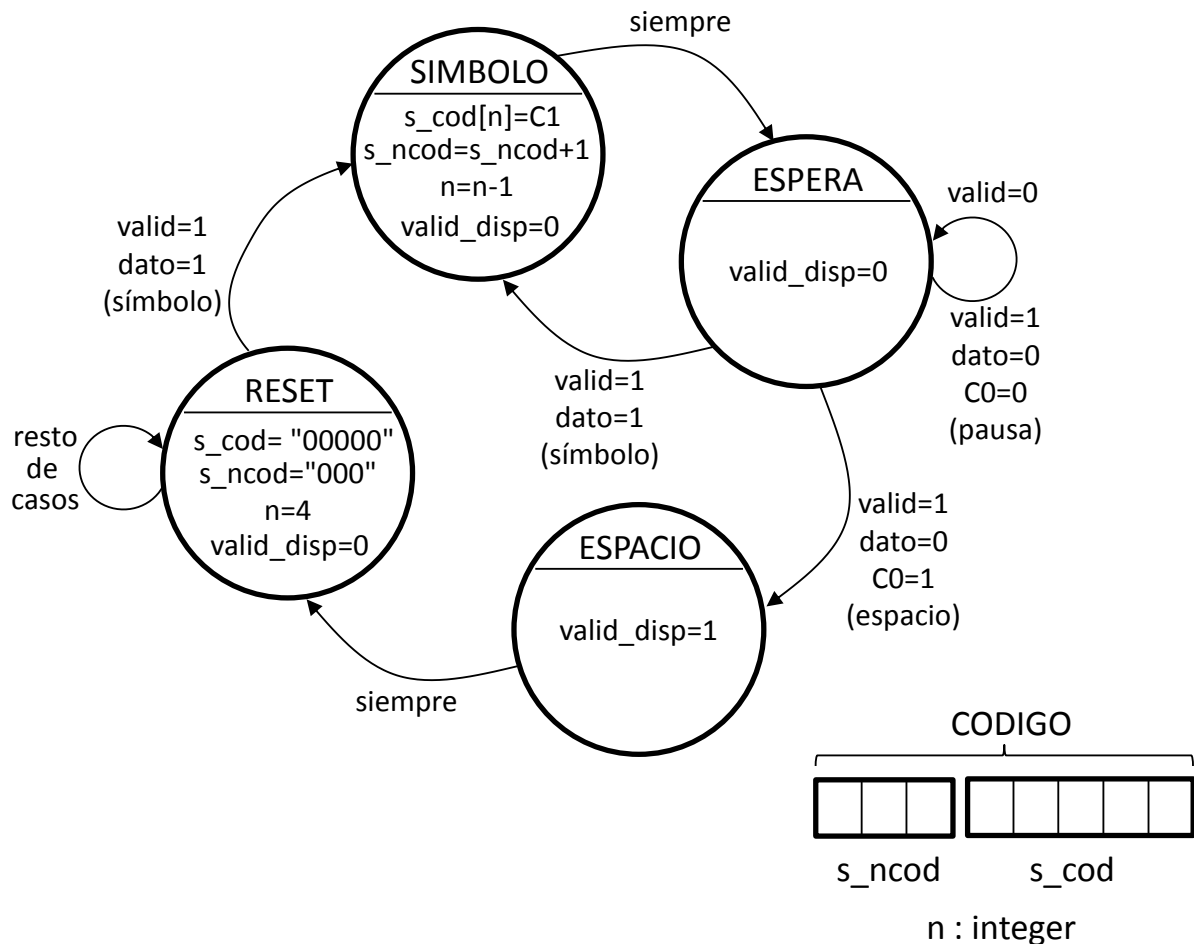
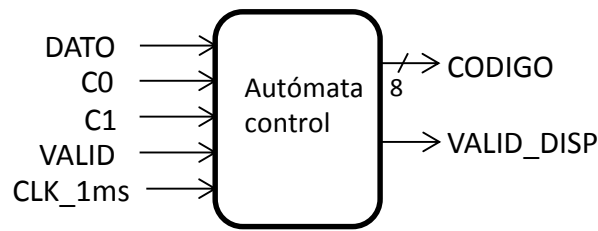


Figura 25 Diagrama de estados del autómata de control

El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación. Complete su descripción funcional según se describe en los comentarios incluidos. **Realice el autómata con un solo proceso.** Asigne el valor de los registros s_ncod y s_cod dentro de cada estado y declare n como un entero (integer). **Los valores de las salidas deberán cablearse fuera del proceso.** Como puede ver, parte del código se muestra como ejemplo.


ARCHIVO aut_control.vhd

```

entity aut_control is
    Port ( CLK_1ms    : in  STD_LOGIC;           -- reloj
           VALID     : in  STD_LOGIC;           -- entrada de dato válido
           DATO       : in  STD_LOGIC;           -- dato (0 o 1)
           C0         : in  STD_LOGIC;           -- resultado comparador de ceros
           C1         : in  STD_LOGIC;           -- resultado comparador de unos
           CODIGO      : out STD_LOGIC_VECTOR (7 downto 0); -- código morse obtenido
           VALID_DISP  : out  STD_LOGIC;         -- validación del display
    end aut_control;

    architecture a_aut_control of aut_control is

        type STATE_TYPE is (ESPACIO, RESET, SIMBOLO, ESPERA);
        signal ST : STATE_TYPE := RESET;

        signal s_ncod : STD_LOGIC_VECTOR (2 downto 0) := "000";
        signal s_cod  : STD_LOGIC_VECTOR (4 downto 0) := "00000";
        signal n : INTEGER range 0 to 4;

    begin

        process (CLK_1ms)
        begin
            if (CLK_1ms'event and CLK_1ms='1') then
                case ST is
                    when SIMBOLO =>
                        s_ncod<=s_ncod+1;
                        s_cod(n)<=C1; -- el resultado del comparador indica punto o raya
                        n<=n-1;
                        ST<=ESPERA;

                        . . . . . COMPLETE AQUÍ EL RESTO DE LOS ESTADOS

                    when RESET =>
                        n <= 4;
                        s_ncod<="000";
                        s_cod<="00000";
                        if (VALID='1' and dato='1') then
                            ST<=SIMBOLO;
                        else
                            ST<=RESET;
                        end if;

                    end case;
                end if;
            end process;

            -- PARTE COMBINACIONAL

            VALID_DISP<='1' when . . . else '0';
            CODIGO(4 downto 0)<= . . .; -- Complete las asignaciones
            CODIGO(7 downto 5)<= . . .

        end a_aut_control ;
    
```

Prueba del autómata de control:

Para probar el funcionamiento de este autómata descargue el archivo **test_aut_control.vhd** y realice la simulación. En la consola del ISIM escriba **run 3 us** para ampliar el tiempo de simulación. La señal

valid_disp debe tener solamente dos pulsos de duración 1 ciclo de reloj. El primero situado en 803,5 ns. En ese instante, el valor de la salida **CODIGO** debe ser 01001000 (correspondiente a la A). El segundo pulso estará situado en 2011,5 ns. En ese instante, el valor de la salida **CODIGO** debe ser 10010000 (correspondiente a la B).

4.2.11 Cableado del circuito completo y fichero de asociaciones:

Finalmente, tendrá que crear el fichero “**receptor.vhd**” que contiene la descripción arquitectural de conexiones de todo el hardware digital. Además también tendrá que crear el fichero de asociaciones que asocie las salidas y entradas del circuito con los recursos reales conectados a las patillas de la FPGA.

Un posible ejemplo de cómo llevar a cabo esta tarea se muestra a continuación. Complete la descripción arquitectural (conexiones) de este módulo y complete también el fichero de asociaciones.

ARCHIVO: **receptor.vhd**

```
entity receptor is
    Port ( CLK      : in  STD_LOGIC;           -- reloj de la FPGA
          LIN       : in  STD_LOGIC;           -- Línea de entrada de datos
          SEG7      : out STD_LOGIC_VECTOR (0 to 6); -- Salida para los displays
          AN        : out STD_LOGIC_VECTOR (3 downto 0)); -- Activación individual
end receptor;

architecture a_receptor of receptor is

    constant UMBRAL0 : STD_LOGIC_VECTOR (15 downto 0) := "0000000011001000"; -- 200 umbral ceros
    constant UMBRAL1 : STD_LOGIC_VECTOR (15 downto 0) := "0000000011001000"; -- 200 umbral unos

    component div_reloj
        Port ( CLK : in  STD_LOGIC;
              CLK_lms : out STD_LOGIC);
    end component;

    component detector_flanco
        Port ( CLK_lms : in  STD_LOGIC; -- reloj
              LIN      : in  STD_LOGIC; -- Línea de datos
              VALOR     : out STD_LOGIC); -- Valor detectado en el flanco
    end component;

    component aut_duracion
        Port ( CLK_lms : in  STD_LOGIC;           -- reloj de 1 ms
              ENTRADA  : in  STD_LOGIC;           -- línea de entrada de datos
              VALID     : out STD_LOGIC;           -- salida de validación de dato
              DATO      : out STD_LOGIC;           -- salida de dato (0 o 1)
              DURACION  : out STD_LOGIC_VECTOR (15 downto 0)); -- salida de duración del dato
    end component;

    component comp_16
        Port ( P : in  STD_LOGIC_VECTOR (15 downto 0);
              Q : in  STD_LOGIC_VECTOR (15 downto 0);
              P_GT_Q : out STD_LOGIC);
    end component;

    component aut_control
        Port ( CLK_lms : in  STD_LOGIC;           -- reloj
              VALID     : in  STD_LOGIC;           -- entrada de dato válido
              DATO      : in  STD_LOGIC;           -- dato (0 o 1)
              C0        : in  STD_LOGIC;           -- resultado comparador de ceros
              C1        : in  STD_LOGIC;           -- resultado comparador de unos
              CODIGO     : out STD_LOGIC_VECTOR (7 downto 0); -- código morse obtenido
              VALID_DISP : out STD_LOGIC);         -- validación del display
    end component;

    component visualizacion
        Port ( E0      : in  STD_LOGIC_VECTOR (7 downto 0); -- Entrada siguiente carácter
              EN       : in  STD_LOGIC;                     -- Activación para desplazamiento
              CLK_lms   : in  STD_LOGIC;                     -- Entrada de reloj de refresco
              SEG7      : out STD_LOGIC_VECTOR (0 to 6);     -- Salida para los displays
              AN        : out STD_LOGIC_VECTOR (3 downto 0)); -- Activación individual
    end component;
```

```

. . . SEÑALES NECESARIAS PARA LAS INTERCONEXIONES

begin

-- Interconexiones de módulos

U1 : div_reloj
    Port map . . . COMPLETE LAS INTERCONEXIONES

end a_receptor;

```

Fichero de asociaciones para compilar con el fichero main.vhd

```

# Reloj principal del sistema
NET "CLK" LOC = "M6"; # Señal de reloj del sistema

# Conexiones de los DISPLAYS
NET "SEG7<0>" LOC = "L14"; # señal = CA
NET "SEG7<1>" LOC = "--"; # Señal = CB
NET "SEG7<2>" LOC = "--"; # Señal = CC
NET "SEG7<3>" LOC = "--"; # Señal = CD
NET "SEG7<4>" LOC = "--"; # Señal = CE
NET "SEG7<5>" LOC = "--"; # Señal = CF
NET "SEG7<6>" LOC = "--"; # Señal = CG

# Señales de activación de los displays
NET "AN<0>" LOC = "K14"; # Activación del display 0 = AN0
NET "AN<1>" LOC = "--"; # Activación del display 1 = AN1
NET "AN<2>" LOC = "--"; # Activación del display 2 = AN2
NET "AN<3>" LOC = "--"; # Activación del display 3 = AN3

#Entrada externa donde se conecta la señal entrante
NET "LIN" LOC = "--"; # Entrada de datos

#Salidas para generar la señal analógica (no tocar estas líneas)
NET "SPI_CLK" LOC = "A9"; # Salida externa terminal 14
NET "SPI_DIN" LOC = "B9"; # Salida externa terminal 15
NET "SPI_CS1" LOC = "C9"; # Salida externa terminal 17

#Entradas para arrancar y parar la señal analógica (no tocar estas líneas)
NET "BTN_START" LOC = "G12"; # Entrada externa BTN0
NET "BTN_STOP" LOC = "C11"; # Entrada externa BTN1

```

4.3 MONTAJE DEL PROTOTIPO

Tenga en cuenta las siguientes consideraciones a la hora de montar el prototipo:

- Intente realizar diseños con cables cortos y componentes con patillas cortas lo más próximos a la placa de inserción que sea posible. Evite las patillas largas pues suelen ser una fuente constante de cortocircuitos entre ellas.
- Utilice valores de resistencias por encima de 100 ohm y por debajo de 100 Kohm siempre que sea posible. Valores muy bajos hacen que fluyan altas corrientes que pueden destruir algún componente. Por el contrario valores muy altos tienden a producir ruido en el circuito.
- Repetimos que los condensadores de desacoplo son fundamentales para el desarrollo del prototipo. No se atenderán cuestiones relacionadas con problemas de funcionamiento del circuito si dichos condensadores no están instalados.

4.3.1 Montaje del circuito por etapas:

El proyecto está pensado para ser montado en dos fases. En la primera fase se montarán todas las etapas de la parte analógica y se comprobará el funcionamiento correcto. Al mismo tiempo deberá ir familiarizándose con los equipos de laboratorio y con el entorno de desarrollo ISE Webpack 12.4.

Posteriormente, en una segunda fase se desarrollará toda la parte digital, completando de esta forma el prototipo.

En las semanas 5, 8 y 11 se deberán entregar las medidas y capturas de pantalla que se indican en las páginas 44, 46 y 48 de este enunciado.

Más adelante se recomienda un calendario para el desarrollo del proyecto donde se detallan los módulos que deben comprobarse en cada sesión (apartado 6).

Recuerde una vez más que el tiempo de laboratorio debe dedicarse a la medida y resolución de problemas, así como para aprender a manejar la instrumentación del puesto. No utilice las sesiones para realizar cálculos, montar módulos o escribir código VHDL puesto que el tiempo es limitado.

4.3.2 Medidas que deben realizarse sobre el circuito:

Deberá llevarse a cabo la caracterización detallada de los filtros Sallen-Key y paso banda, las cuales deberán entregarse. Para realizar esta caracterización se necesitan tres cables coaxiales terminados en pinzas.

Cada uno de los filtros se mide de la siguiente forma:

1. Se mide la salida del generador de funciones en vacío (sin cargar con el filtro). Debe ajustarse para observar una senoide de 0,5 Vp de amplitud (1 Vpp). Asegúrese siempre de medir la salida del generador antes de conectarla al circuito.
2. A continuación se montará el circuito mostrado en la Figura 26 y se irá variando la frecuencia del generador de funciones (periodo T) para obtener la función de transferencia (módulo y fase) del filtro. **Los filtros deben medirse aisladamente**, es decir, deberá desconectar su entrada y salida del circuito para poder medirlos individualmente. Para conseguir una curva fiable, asegúrese de medir varios puntos por década (por lo menos 5). Para dibujar la gráfica de la fase utilice valores de Φ entre -180° y $+180^\circ$. Es decir, si mide un desfase mayor de 180° , considere el valor negativo $\Phi-360^\circ$.

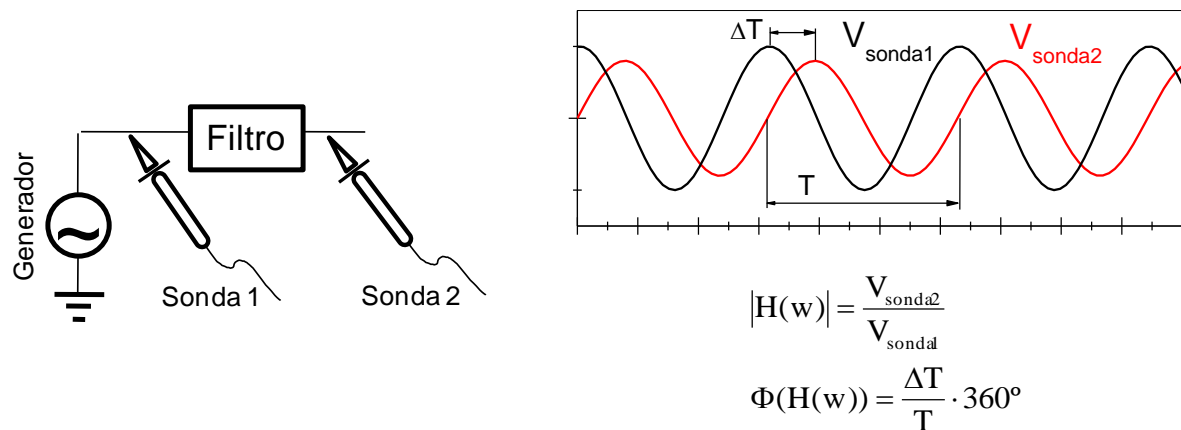


Figura 26 Esquema del circuito de medida de los filtros. Se muestra un ejemplo real de cómo calcular el módulo y la fase de la función de transferencia.

4.3.3 Realización de diagramas de Bode:

Se pide también la realización de los diagramas asintóticos de Bode de módulo y fase correspondientes a los filtros Sallen Key y paso banda. Se recuerda que el diagrama de Bode es un **diagrama asintótico aproximado del comportamiento del filtro**, que se obtiene como resultado del análisis de los polos y los ceros de las funciones de transferencia. Para más información consulte la referencia [3].

En este caso, para mayor facilidad, se indican a continuación las funciones de transferencia de los diferentes filtros para la realización de los citados diagramas de Bode:

Filtro paso bajo de Sallen-Key:

$$H(j\omega) = \frac{G \frac{1}{R^2 C^2}}{-\omega^2 + j\omega \frac{3-G}{RC} + \frac{1}{R^2 C^2}}$$

Filtro paso banda:

$$H(j\omega) = \frac{-j\omega \cdot \frac{1}{R_1 C}}{-\omega^2 + j\omega \frac{2}{R_2 C} + \frac{1}{R_1 R_2 C^2}}$$

Los diagramas de Bode y las medidas deberán representarse en una gráfica donde el eje X muestre la frecuencia en modo logarítmico y el eje Y muestre la amplitud en dB y la fase en grados. En las hojas de entrega (páginas 45 y 47) se proporcionan plantillas de estas gráficas.

4.3.4 Utilización del osciloscopio:

En este proyecto será imprescindible el uso del osciloscopio. En particular, será muy necesaria la función que permite capturar una señal en la pantalla. Esto nos va a permitir capturar parte de la secuencia de puntos y rayas y comprobar el funcionamiento de las distintas etapas. También puede emplear la función que congela la imagen de la pantalla del osciloscopio.

Las señales mostradas en la pantalla del osciloscopio pueden ser además capturadas mediante el ordenador del puesto para su posterior presentación en la memoria.

5. ESPECIFICACIONES DEL SISTEMA

El sistema que se diseñe deberá cumplir las siguientes especificaciones:

1. Alimentación simétrica con +5 y -5 V procedente de la fuente de alimentación del laboratorio.
2. Introducción de la señal de datos mediante el entrenador del laboratorio.
3. Parte analógica realizada con amplificadores operacionales del tipo TL082.
4. Parte digital descrita en VHDL y sintetizada sobre FPGA Spartan 3E.
5. Tenga en cuenta los niveles de señal exigidos en distintos puntos a lo largo del enunciado.
6. Solamente se permite la utilización de dos cables entre la parte analógica y la digital: el cable de datos y el de masa.
7. Se prohíbe el uso de altavoces.
8. Será suficiente con entregar el proyecto montado sobre placa de inserción.

6. DESARROLLO RECOMENDADO POR SESIONES

Este apartado constituye una guía para la realización del prototipo, si bien la planificación real puede diferir puesto que es difícil tener en cuenta todos los contratiempos posibles. Sirva de ayuda para que cada grupo pueda organizar el tiempo de acuerdo a su situación particular.

Desde el comienzo, realice todas las tareas sobre el prototipo con el máximo cuidado. Por ejemplo, montaje de la alimentación, emplazamiento de los componentes, pelado y conexión de los cables, etc. Aunque al principio parezca que todo es manejable, a medida que el montaje crece perderá el control sobre el mismo si empiezan a aparecer incertidumbres en la fiabilidad de las conexiones, falta de espacio para nuevos circuitos, amontonamiento de componentes que dificultan el empleo de las sondas del osciloscopio, etc.

Tenga en cuenta otra vez que el laboratorio debería servirle para medir y buscar y solucionar problemas, no para montar circuitos. Se debe realizar el montaje de circuitos fuera de las horas de laboratorio.

Se recomienda entonces, seguir la siguiente distribución del trabajo en semanas:

6.1 Semana 1

Familiarización con los equipos del laboratorio y el entorno ISE.

- Emplee el tiempo para familiarizarse con el funcionamiento de la fuente de alimentación, el generador de señal y el osciloscopio.
- Aprenda a utilizar el programa de captura de pantalla del osciloscopio instalado en el ordenador del puesto.
- Arranque el entorno ISE, compile y sintetice sobre la FPGA los ejemplos 1 y 2 contenidos en el "Manual de referencia de la tarjeta BASYS 2". (Intente avanzar lo más posible, dedique tiempo a compilar y sintetizar más ejemplos)
- Realice modificaciones sobre los ejemplos y aprenda a utilizar el archivo de asociaciones.
- Trate de sacar provecho de esta sesión para familiarizarse lo más posible con el entorno de trabajo que va a utilizar durante el curso.

6.2 Semana 2

Etapas de acondicionamiento de señal (ap. 4.1.2)

- Ajuste el generador de señal para obtener una salida de 2 Vpp y 1 KHz de frecuencia. Conecte esta señal a la entrada de la etapa de acondicionamiento.
- Varíe la frecuencia del generador y asegúrese que la frecuencia de corte inferior del filtro paso alto está de acuerdo con su diseño. (A la frecuencia de corte, el nivel de salida del filtro debería ser $1/\sqrt{2}$ veces la amplitud de entrada).
- Cargue y compile el fichero gen_senal y sintetice el hardware sobre la FPGA mediante el entorno ADEPT. Mida la salida del entrenador de señal y asegúrese de observar una señal senoidal de 1 KHz y 2 Vpp de amplitud con una continua de 1 V.
- Conecte el entrenador a la entrada de la etapa y ajuste el potenciómetro para obtener una señal de 1 Vpp a la salida del operacional.

6.3 Semana 3

Filtro paso banda (ap. 4.1.4) y rectificador (ap. 4.1.5)

- Mida el filtro según se indica en el apartado 4.3.2 de este enunciado. Utilice como entrada una senoidal de 1 Vpp de amplitud. Tome varios puntos por década (un mínimo de 5). Mida tantas décadas como le sea posible tanto por debajo como por encima de la frecuencia de paso.
- Anote los valores de módulo y fase de la señal obtenida a la salida del filtro. Dibuje una curva del comportamiento del filtro en módulo y fase en función de la frecuencia. Estas medidas serán necesarias para la primera entrega.
- Obtenga el valor del cero y de los dos polos de la función de transferencia del filtro paso banda (apartado 4.3.3). Para los polos obtendrá dos valores complejos conjugados. Calcule su módulo.
- Conecte el filtro a las etapas anterior (acondicionador) y posterior (rectificador), introduzca la señal a la entrada a través del entrenador y compruebe que el rectificador funciona adecuadamente.

6.4 Semana 4

Etapas de amplificación de potencia: (ap. 4.1.3)

- Esta etapa debería funcionar sin ajustes. Tenga en cuenta no obstante el siguiente consejo: coloque un condensador de 100 μF entre alimentación y masa cerca del LM386 (patillas 6 y 4). Esto filtra la alimentación y ayuda a eliminar posibles oscilaciones.
- Ajuste el potenciómetro para regular el volumen.
- **Se recuerda que está prohibido el uso de altavoces.**

6.5 Semana 5

Filtro paso bajo y comparador (ap. 4.1.6 y 4.1.7)

- Mida el filtro según se indica en el apartado 4.3.2 de este enunciado. Utilice como entrada una senoidal de 1 Vpp de amplitud. Tome varios puntos por década (un mínimo de 5). Mida tantas décadas como le sea posible a partir de 1 Hz.
- Anote los valores de módulo y fase de la señal obtenida a la salida del filtro. Dibuje una curva del comportamiento del filtro en módulo y fase en función de la frecuencia. Estas medidas serán necesarias para la segunda entrega.
- Obtenga el valor de los dos polos de la función de transferencia del filtro paso bajo (apartado 4.3.3). Obtendrá un polo real doble. Dibuje el **diagrama asintótico de Bode** correspondiente sobre la curva de comportamiento del filtro.
- Conecte el filtro a la etapa anterior (rectificador), introduzca la señal a la entrada a través del entrenador y compruebe que el filtro funciona adecuadamente.
- Por último conecte la salida del filtro al comparador y sitúe el umbral del comparador en el valor necesario para obtener una señal digital entre 0 y 5 V con flancos definidos.

ENTREGA 1: Esta semana deberá entregar las medidas realizadas sobre el filtro paso banda según el formato que figura en la página 44 de este documento.

Al final de esta sesión debería tener terminada la parte analógica.

Durante las sesiones 2, 3, 4 y 5 intente dedicar tiempo a compilar, sintetizar y entender los ejemplos 3 y 4 del “Manual de referencia de la tarjeta BASYS 2”. Intente también simular los ejemplos ISIM 1 y 2 del citado manual.

6.6 Semana 6

Familiarización con las entradas y salidas externas del entrenador BASYS 2

- Dedique esta sesión a entender, compilar y sintetizar los ejemplos 5 a 8 del “Manual de referencia de la tarjeta BASYS 2”.
- Al final de esta semana debería tener un conocimiento suficiente sobre el entorno ISE y el entorno de desarrollo BASYS 2 como para desarrollar el hardware que se describe en este enunciado.
- Aproveche el tiempo para realizar algunas de las medidas que se exigen para la tercera entrega (página 49).

6.7 Semana 7

Divisor del reloj (ap. 4.2.5) y módulo de visualización: multiplexor, refresco y registro de desplazamiento (ap. 4.2.6.2, 4.2.6.3 y 4.2.6.4)

- Asocie (mediante un fichero de asociaciones) las salidas del divisor del reloj hacia un terminal externo de la FPGA y mida con el osciloscopio la frecuencia de la señal obtenida.
- Los módulos del circuito de visualización son muy simples. Dedique tiempo a entender su funcionamiento y realice simulaciones mediante test_bench de los módulos del multiplexor, refresco y registro de desplazamiento para asegurarse de que funcionan adecuadamente.
- Apóyese en todo momento en los ejemplos 3 y 4 del “Manual de referencia de la tarjeta BASYS2”.

6.8 Semana 8

Módulo de visualización: cableado (ap. 4.2.6.5)

- Emplee el archivo **prueba_visualizacion.vhd**. Este archivo contiene un *component* que se refiere al módulo de visualización realizado previamente. Compile este archivo junto con su código y con el archivo de asociaciones: **asoc_prueba_vis.ucf**.
- Si todo funciona adecuadamente deberá observar la secuencia de caracteres: A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,1,2,3,4,5,6,7,8,9,0.

ENTREGA 2: Esta semana deberá entregar las medidas realizadas sobre el filtro paso bajo según el formato que figura en la página 46 de este documento.

6.9 Semana 9

Detector de flanco (ap. 4.2.7) y autómata de medida de duración (4.2.8)

- Para probar el funcionamiento del detector de flanco descargue el archivo de test_bench: **test_detector_flanco.vhd** y realice la simulación. Siga las instrucciones indicadas en el apartado correspondiente.
- Para probar el funcionamiento del autómata medidor de duración descargue el archivo de test_bench: **test_aut_duracion.vhd** y realice la simulación. Aparecerán los valores de todas las entradas y las salidas. Siga las instrucciones indicadas en el apartado correspondiente.

6.10 Semana 10

Comparadores (ap. 4.2.9) y autómata de control (ap. 4.2.10)

- El comparador es un elemento combinacional muy sencillo. Realice alguna simulación con un test_bench para asegurar que este módulo funciona adecuadamente.
- Para probar el funcionamiento del autómata descargue el archivo **test_aut_control.vhd** y realice la simulación. Siga las instrucciones indicadas en el apartado correspondiente.

6.11 Semana 11

Cableado del circuito completo y asociaciones (ap. 4.2.11)

- Deberá completar la descripción arquitectural del circuito y asociar sus terminales lógicos con las patillas físicas de la FPGA.
- Modifique el archivo de asociaciones que se proporciona (**asoc.ucf**) para incluir las entradas y salidas propias del circuito implementado. Fíjese para ello en el ejemplo propuesto en el apartado 4.2.11 (en verde).

ENTREGA 3: Esta semana deberá entregar las medidas que se indican en la página 48 de este documento.

Al final de esta sesión debería tener terminado el prototipo completo.

6.12 Semana 12

Examen escrito (tipo test), redacción de la memoria final y realización de posibles mejoras

- Examen escrito (tipo test)
- Preparación del examen oral y redacción de la memoria
- Realización de posibles mejoras

6.13 Semana 13: Turnos libres

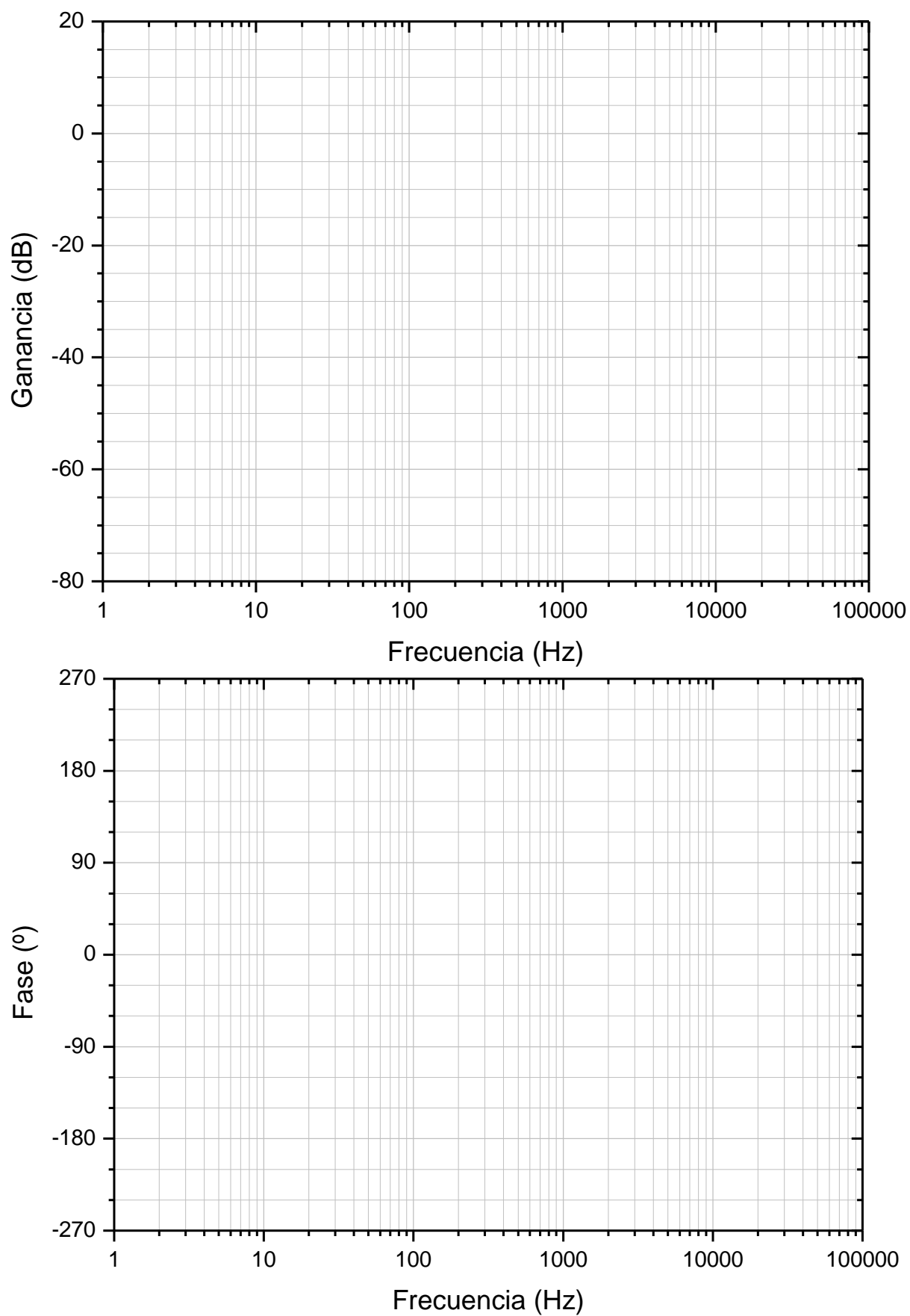
Redacción de la memoria final y realización de posibles mejoras

- Preparación del examen y redacción de la memoria
- Realización de posibles mejoras

6.14 Semana 14 – 17 al 21 de Diciembre

Examen oral

- Se realizarán los exámenes orales en el horario de cada turno.

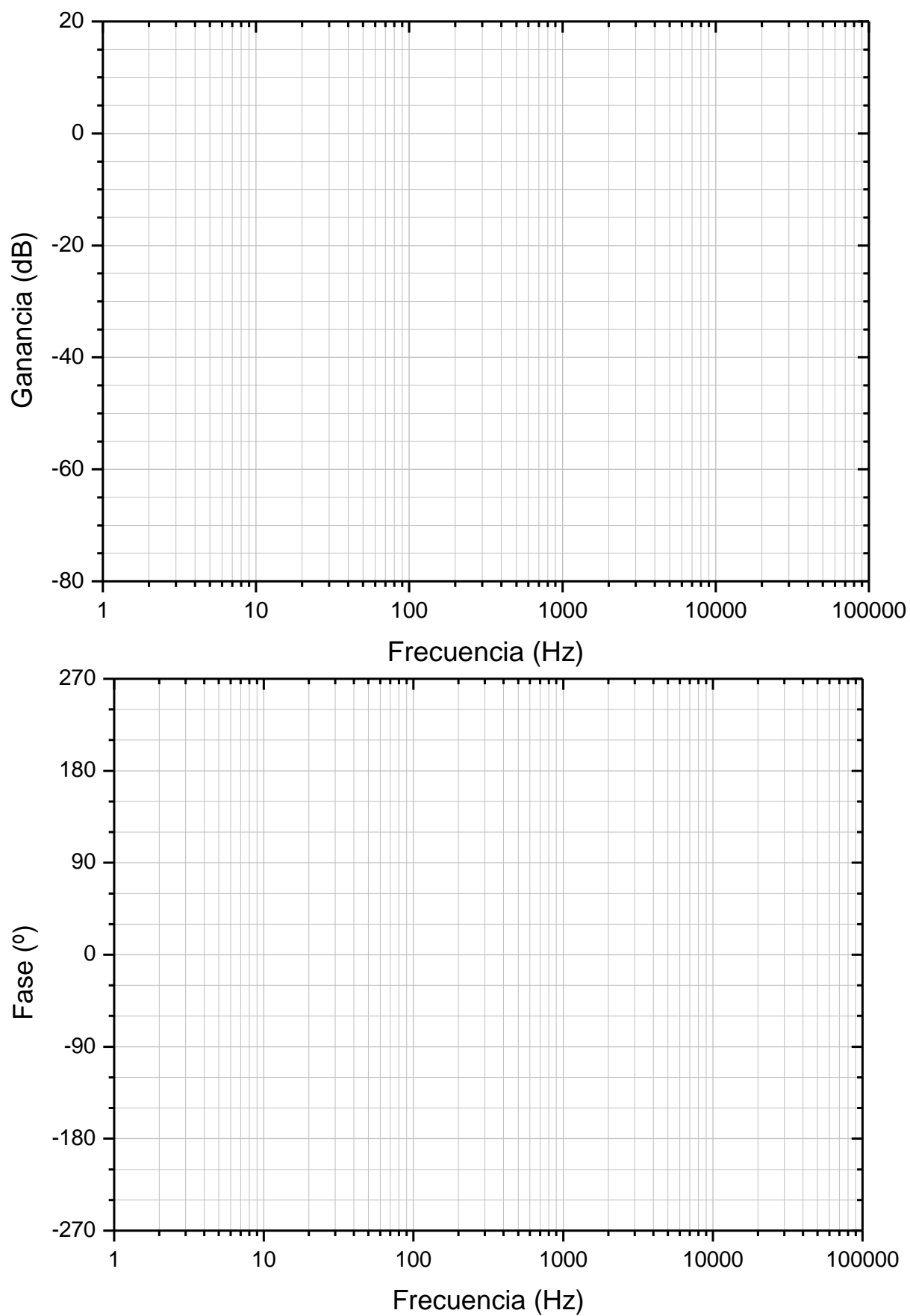
Medidas del comportamiento del filtro paso banda y diagrama asintótico de Bode

Código de pareja	
Alumno 1 (Nombre y apellidos)	
Alumno 2 (Nombre y apellidos)	

Tabla de valores medidos

[illegible]

Especifique aquí el valor de los dos polos. Represente en la página siguiente **el diagrama asintótico de Bode** de la función de transferencia junto con la medida del filtro.

Medidas del comportamiento del filtro paso bajo y diagrama asintótico de Bode

ENTREGA 3

Código de pareja	
Alumno 1 (Nombre y apellidos)	
Alumno 2 (Nombre y apellidos)	

CAPTURAS DE PANTALLA DEL OSCILOSCOPIO

Incluya a continuación las siguientes capturas de pantalla del osciloscopio. En todos los casos la escala deberá ser la adecuada para apreciar con precisión la amplitud de la señal medida:

1. Señal a la salida del entrenador. Mida en modo DC para poder apreciar la componente continua de la señal. Congele la imagen y realice una captura donde se puedan apreciar al menos un PUNTO, una RAYA, una PAUSA y un ESPACIO.
2. Señal a la salida del acondicionador de señal. Mida en modo DC para poder apreciar la componente continua de la señal. Congele la imagen y realice una captura donde se puedan apreciar al menos un PUNTO, una RAYA, una PAUSA y un ESPACIO.
3. Señal a la salida del filtro paso banda. Mida en modo DC para poder apreciar la componente continua de la señal. Congele la imagen y realice una captura donde se puedan apreciar al menos un PUNTO, una RAYA, una PAUSA y un ESPACIO.
4. Señal a la salida del rectificador. Mida en modo DC para poder apreciar la componente continua de la señal. Congele la imagen y realice una captura donde se puedan apreciar al menos un PUNTO, una RAYA, una PAUSA y un ESPACIO.
5. Señal a la salida del filtro paso bajo. Mida en modo DC para poder apreciar la componente continua de la señal. Congele la imagen y realice una captura donde se puedan apreciar al menos un PUNTO, una RAYA, una PAUSA y un ESPACIO.
6. Señales a la entrada del comparador. Mida en modo dual con una señal en cada canal del osciloscopio. Utilice el modo DC con ambos canales en la misma escala y nivel de 0. Congele la imagen y realice una captura donde se puedan apreciar al menos un PUNTO, una RAYA, una PAUSA y un ESPACIO, y donde se vea además el umbral correctamente situado.
7. Salida de 1000 Hz obtenida mediante división de frecuencia de la señal de reloj de la FPGA (señal CLK_1ms). Mediante el archivo de asociaciones conecte esta señal con algún pin de salida de la FPGA y médala con el osciloscopio. Capture la señal de tal manera que puedan apreciarse claramente su amplitud y su frecuencia en la imagen.

7. MEMORIA FINAL PARA EXAMEN ORAL

Antes del examen oral deberá entregar una memoria escrita del trabajo realizado. La memoria deberá contener **obligatoriamente** como mínimo las siguientes partes, medidas y datos:

1. Una portada indicando: nombre de la asignatura, título del proyecto, nombre completo de los autores y código correspondiente (día de la semana, número de turno y puesto).
2. Diseño **detallado y razonado** de cada una de las etapas que integran el diseño, explicando las razones de la elección de los valores de los componentes utilizados.
3. Un esquema eléctrico completo del circuito analógico
4. Un diagrama lógico completo del circuito digital (módulos e interconexión entre ellos).

Parte analógica:

- Diseño detallado y justificación de todos los componentes empleados

Parte digital:

- Es obligatorio entregar el código VHDL junto con el fichero de asociaciones a través del portal de la asignatura.
- Pruebas de simulación (si las ha hecho). Resultados y conclusiones de las mismas.

A lo largo del curso se publicará una plantilla para realizar la memoria.

Insistimos en que las versiones en formato electrónico de los documentos generados (memoria, simulaciones, código VHDL, etc.) deberán entregarse a través del portal de la asignatura.

En particular es muy importante entregar todo el código VHDL, junto con el fichero de asociaciones a través del portal de la asignatura. Tenga en cuenta que todo el código será analizado mediante un software de control de copias para detectar posibles copias entre alumnos.

El examen se realizará descargando y compilando el código VHDL entregado en el portal de entrega del MOODLE por lo que no se permiten modificaciones posteriores a la entrega.

Documente sus diseños, cálculos, justificaciones, esquemas, medidas, observaciones, dificultades, etc., a medida que los realiza, semana por semana, módulo a módulo. De este modo le será sencillo componer finalmente la memoria.

LA MEMORIA DEBERÁ SER ENTREGADA LOS DÍAS 10, 11 y 12 de Diciembre del 2018 SEGÚN SU TURNO DE PRÁCTICAS. CONSULTE EL CALENDARIO PARA MÁS DETALLE.

8. MEJORAS

En los apartados precedentes se ha hecho una descripción bastante detallada de los subsistemas a diseñar, así como de alguno de los esquemas circuitales utilizables. Salvo que se haya indicado lo contrario, lo descrito corresponde a las especificaciones mínimas que deberá cumplir el diseño realizado (el prototipo básico), y constituirá el requisito mínimo para aprobar la asignatura.

Por otro lado, la puntuación máxima alcanzable con este prototipo básico es de 8 puntos, partiendo de la base de que el funcionamiento es correcto y de que se han comprendido los fundamentos teórico-prácticos de todo ello, lo que será verificado a través de las memorias, el examen oral y la prueba escrita a realizar.

Para incrementar la calificación puede abordarse alguna realización opcional, como las que se plantean a continuación o cualquier otra que se le ocurra (consulte con un profesor). En todo caso, no se trata de multiplicar innecesariamente el número de circuitos integrados en su prototipo, ni de replicar módulos idénticos.

Recomendamos encarecidamente a los alumnos que antes de abordar cualquier mejora hagan un estudio pormenorizado de las implicaciones de la misma. Tómense el tiempo necesario en la fase de diseño y no ataquen el montaje de forma impulsiva. Una mejora en apariencia sencilla puede volverse sumamente engorrosa, bien debido al número de circuitos a interconectar o por incluir detalles y complicaciones no suficientemente previstos.

Dicho esto, se proponen a continuación algunas mejoras que se pueden realizar:

8.1 Introducción de un separador entre palabras (dificultad baja):

El proyecto básico contempla únicamente la separación entre letras (ESPACIO). Sin embargo las palabras aparecen seguidas puesto que no existe un separador entre ellas. Esta mejora propone la existencia de un nuevo símbolo (SEPARADOR) con la duración de 700 ms que se utiliza como separador de palabras.

Descargue el fichero: **gen_senal_2.vhd** y añádalo como componente en el fichero **main.vhd** en lugar del **gen_senal.vhd**. Este fichero contiene un mensaje donde las letras están separadas por espacios de 300 ms de duración, pero las palabras están separadas por espacios más largos, de 700 ms (SEPARADOR).

Para realizar esta mejora se recomienda añadir un nuevo comparador que permita distinguir la mayor duración de este separador, y añadirla como entrada en el autómata de control. También deberá de modificar el diagrama de estados del autómata para tener en cuenta este nuevo símbolo. En el caso de que aparezca un separador se visualizará un espacio en blanco.

8.2 Incorporación de una señal asíncrona de reset (dificultad baja):

Se trata de añadir una señal de reset en el módulo main.vhd, asíncrona y activa a nivel alto. La activación de esta señal deberá borrar el contenido de los displays y llevar al sistema a un estado que le permita la recepción de un nuevo mensaje.

8.3 Detección del final del mensaje (dificultad media):

Generalmente, en las antiguas transmisiones morse, los mensajes se terminaban con una K. Esta mejora consiste en detectar la aparición de una K y detener el autómata de control encendiendo además uno de los LEDs de la placa BASYS2. El autómata debe permanecer detenido hasta que se pulse el pulsador de la izquierda de la tarjeta BASYS2 (BTN3). En ese momento se apagará el LED y el autómata quedará en disposición de recibir y decodificar un nuevo mensaje. Deberá modificar el

citado autómata y añadir una nueva entrada para el botón y una nueva salida para contemplar el encendido del LED. El mensaje contenido en el archivo **gen_senal.vhd** termina con una K para poder implementar esta mejora.

8.4 Detección avanzada del final del mensaje (dificultad alta):

Esta mejora implica realizar primero la 8.1. Dado que las transmisiones antiguas terminaban con una K esta mejora consiste en detectar la aparición de una K como una letra única, no como parte de una palabra. En este caso el autómata se detendrá iluminando un LED de la placa BASYS 2. Al igual que en la mejora 8.3 el autómata volverá a ponerse en marcha al pulsar el botón de la izquierda (BTN3) de la placa BASYS2, apagando el LED y quedando dispuesto para la recepción y decodificación de un nuevo mensaje. Descargue el fichero: **gen_senal_2.vhd** y añádalo como componente en el fichero **main.vhd** en lugar del **gen_senal.vhd**. Este fichero termina con una K como letra independiente.

8.5 Uso de esquemas circuitales alternativos a los propuestos

Se valorará positivamente la inclusión de circuitos distintos a los propuestos, siempre que:

- Impliquen una mayor dificultad o una novedad interesante
- No se limiten a duplicar subsistemas ya construidos

DIFICULTAD: en función del esquema alternativo, atendiendo tanto a la complejidad conceptual como de implementación.

8.6 Simulación con PSICE ó LTSPICE (dificultad alta)

En este curso se propone una mejora consistente en la realización, mediante el programa LTSPICE o PSICE, de diversas simulaciones de la parte analógica del montaje.

Para que esta mejora se valore adecuadamente será necesario incluir lo siguiente:

- Simulaciones temporales (análisis transitorio) del circuito, que incluya los siguientes elementos: filtro paso banda, rectificador y filtro paso bajo Sallen Key (Apartados 4.1.4, 4.1.5 y 4.1.6) utilizando como excitación a la entrada de la cadena la secuencia PAUSA PUNTO PAUSA RAYA PAUSA modulada con la señal de 1 KHz (no se recomienda emplear secuencias de más símbolos por la complejidad del análisis).
- Simulaciones en AC (barrido en frecuencia) del filtro paso banda (apartado 4.1.4) y del filtro paso bajo Sallen Key (apartado 4.1.6) por separado. Se piden las gráficas de respuesta en amplitud (dB) y fase (°).

En la memoria será necesario incluir los diagramas esquemáticos utilizados, así como las gráficas de las simulaciones obtenidas, discutiendo la adecuación de dichos resultados a las previsiones teóricas y a las medidas experimentales.

Puede descargar el software LTSPICE gratuitamente desde la página:

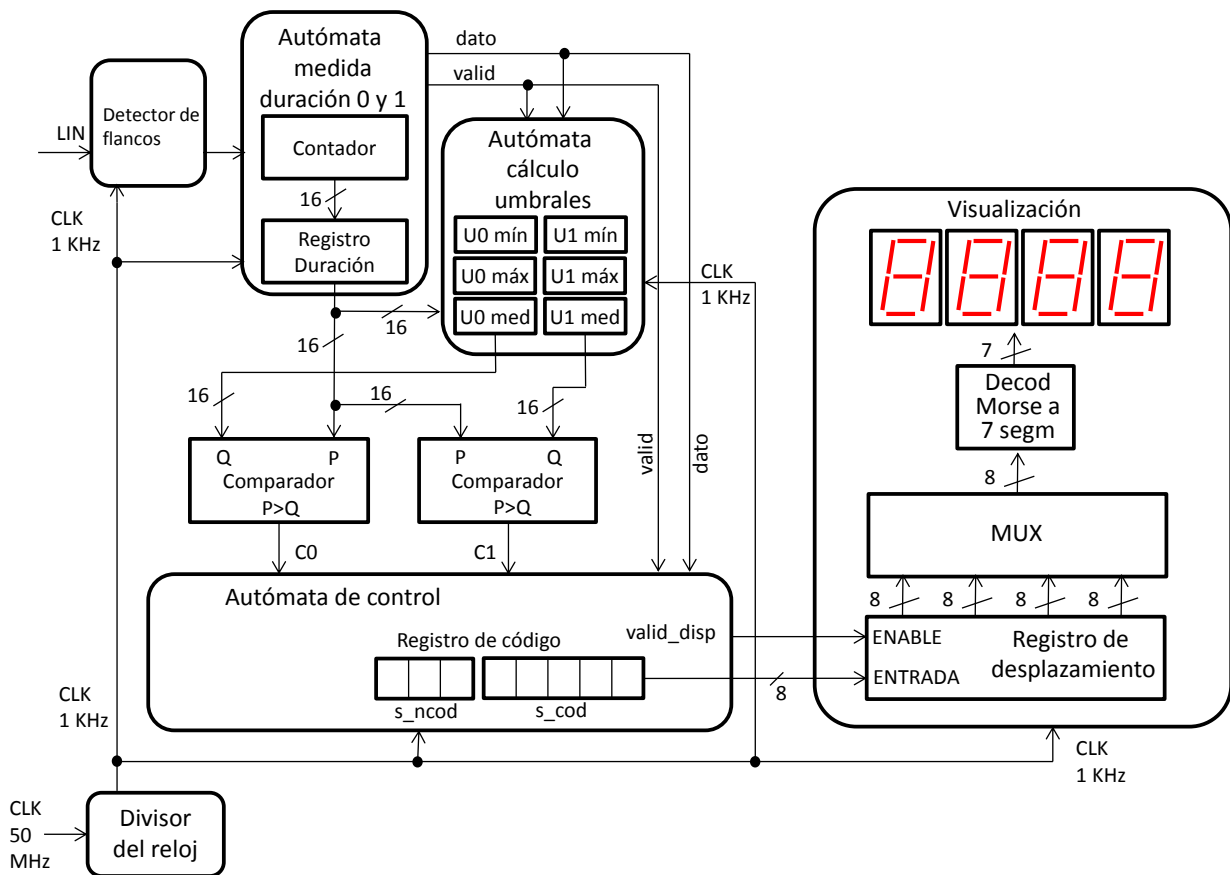
<http://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>.

8.7 Esquema alternativo para la detección de código morse transmitido manualmente (dificultad alta)

El proyecto que se presenta en esta memoria contempla la detección de código morse donde los puntos, las rayas, los espacios y las pausas tienen una duración fija y exacta. En la realidad, cuando se transmite código morse de manera manual, las duraciones son aleatorias y dependen de la pericia y la precisión del operador.

En esta mejora se pretende realizar un esquema alternativo que sea capaz de detectar código con longitudes de símbolos aleatorias. Para ello deberá desarrollarse otro autómata que sea capaz de almacenar las diferentes duraciones de unos y ceros y calcular una media que será utilizada como umbral de comparación para distinguir los símbolos.

Un esquema posible es el siguiente:



Si quiere realizar esta mejora consulte con tiempo al profesor Álvaro de Guzmán Fernández (guzman@die.upm.es).

Referencias

- [1] Alan V. Oppenheim y Alan S. Willsky, Señales y Sistemas, 2ª edición, Prentice-Hall, 1998.
- [2] A. Bruce Carlson, Communication systems: An Introduction to Signals and Noise in Electrical Communication, 3ª edición, McGraw-Hill, 1986.
- [3] Norbert R. Malik, Circuitos Electrónicos: Análisis, Diseño y Simulación, Prentice-Hall, 1996.
- [4] Sergio Franco, Design with Operational Amplifiers and Analog Integrated Circuits, 2ª edición, McGraw-Hill, 1997.
- [5] Sergio Franco, Diseño con Amplificadores Operacionales y Circuitos Integrados Analógicos, 3ª edición, McGraw-Hill, 2005.
- [6] Aspectos Prácticos de Diseño y Medida en Laboratorios de Electrónica, 2ª edición, Dpto. de Publicaciones de la ETSIT (UPM), 2002.
- [7] Manual de referencia de la tarjeta BASYS2, Álvaro de Guzmán Fernández, disponible en la página de la asignatura.