

Circuitos Electrónicos (CELT)

Descripción del proyecto

Curso 2017-2018

Demodulador y decodificador de la señal DCF77

Miguel Ángel Sánchez García
Álvaro de Guzmán Fernández

ÍNDICE GENERAL

1. INTRODUCCIÓN	4
Calendario de la asignatura	5
2. DESCRIPCIÓN GENERAL	6
2.1 La señal DCF77 original	7
2.2 La señal empleada en este proyecto	7
2.3 Objetivo de este proyecto	8
2.4 Esquema detallado: circuito analógico	9
2.5 Esquema detallado: circuito digital	9
3. REALIZACIÓN DEL PROTOTIPO	11
3.1 Descomposición en módulos analógicos y digitales	11
3.2 Forma de trabajar en la asignatura	11
3.3 Equipo necesario	12
3.4 Montaje	12
4. FUNCIONAMIENTO DETALLADO	14
4.1 SUBSISTEMA ANALÓGICO	14
4.1.1 Acondicionador de señal	14
4.1.1.1 Amplificador y eliminación de posible componente continua	14
4.1.1.2 Filtro paso banda	15
4.1.2 Recuperación de la señal de datos a partir de la señal modulada	16
4.1.3 El demodulador AM	17
4.1.4 Rectificador	18
4.1.5 Filtro paso bajo	19
4.1.6 Comparador	19
4.2 SUBSISTEMA DIGITAL	20
4.2.1 Funcionamiento del subsistema digital	21
4.2.2 Estructura del código VHDL	22
4.2.3 Generador de reloj	23
4.2.4 Registro de desplazamiento de 40 bits	24
4.2.5 Sumador de 40 bits de entrada	25
4.2.6 Comparador	27
4.2.7 Puerta AND	27
4.2.8 Registro de desplazamiento	28
4.2.9 Registro	29
4.2.10 Autómata de MOORE	29
4.2.11 Módulo de visualización	32
4.2.12 Cableado del circuito completo y fichero de asociaciones	33
4.3 MONTAJE DEL PROTOTIPO	36
4.3.1 Montaje del circuito por etapas	36
4.3.2 Medidas que deben realizarse sobre el circuito	36
4.3.3 Realización de diagramas de Bode	37
4.3.4 Utilización del osciloscopio	37
5. ESPECIFICACIONES DEL SISTEMA	38
6. DESARROLLO RECOMENDADO POR SESIONES	39
ENTREGA 1	44
ENTREGA 2	46
ENTREGA 3	48
7. MEMORIA FINAL PARA EXAMEN ORAL	49
8. MEJORAS	50
8.1 Presentación de la hora en formato AM / PM (dificultad baja)	50
8.2 Visualización de hora y fecha (dificultad media)	50
8.3 Visualización de hora, fecha y día de la semana (dificultad alta)	51
8.4 Uso de esquemas circuitales alternativos a los propuestos	51
8.5 Simulación con PSPICE (1 PUNTO) (dificultad alta)	51
8.6 Montaje en PCB (1,5 PUNTOS) (dificultad muy alta)	52
Referencias	53

ÍNDICE DE FIGURAS

Figura 1 Cobertura de la señal DCF77	6
Figura 2 Forma de onda de la señal DCF77.....	7
Figura 3 Forma de onda genérica, secuencia binaria correspondiente a la hora 19:30 y estructura del fichero mp3 suministrado.	8
Figura 4 Esquema general del circuito propuesto, compuesto por un circuito analógico y un hardware digital sintetizado en FPGA.....	9
Figura 5 Esquema general de bloques del circuito analógico	9
Figura 6 Valor de la energía de la señal en cada periodo de bit en función de los datos transmitidos	10
Figura 7 Esquema general de bloques de la parte digital	11
Figura 8 Esquema con la placa de inserción y la conexión de las bornas de alimentación.....	13
Figura 9 Condensadores de desacoplo y LEDs para la detección rápida de cortocircuitos.	13
Figura 10 Amplificador y adaptador de impedancia del acondicionador de señal	15
Figura 11 Conexión correcta del conector “jack” para el reproductor MP3.	15
Figura 12 Filtro paso banda de realimentación múltiple de adaptación al canal.	15
Figura 13 Señal de datos en banda base y espectro en frecuencia de la misma	16
Figura 14 Portadora modulada por la señal de datos y espectro en frecuencia correspondiente	17
Figura 15 Señal obtenida tras la rectificación y espectro en frecuencia.	17
Figura 16 Señal obtenida tras un filtrado paso bajo de orden 2 y espectro en frecuencia.	18
Figura 17 Esquema del demodulador	18
Figura 18 Rectificador construido con un diodo.....	18
Figura 19 Filtro paso bajo Sallen-Key	19
Figura 20 El comparador LM311 y esquema del proceso de comparación	20
Figura 21 Esquema de bloques del subsistema digital completo.....	21
Figura 22 Ejemplos de muestreo correcto e incorrecto sobre la señal entrante	22
Figura 23 Forma de conseguir el sincronismo entre la señal entrante y el muestreo del autómata	22
Figura 24 Diagrama de estados del autómata.....	30
Figura 25 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.	37

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 clases y tutorías 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 semanas 5,8 y 11 del curso, junto con una memoria escrita final** (para entregar en la penúltima semana) 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: <http://celt.die.upm.es>

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 de la Práctica. Este cuaderno **será de gran utilidad a la hora de escribir los documentos entregables y la memoria**.
2. La evaluación continua se realizará en base a 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 la práctica 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.**

Para cualquier consulta, puede dirigirse al coordinador de la asignatura, Miguel Ángel Sánchez García (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. Durante la primera semana estará abierto el periodo de inscripción y matrícula en la asignatura y deberá hacerse a través del portal. Es obligatoria la asistencia a una de las dos clases introductorias que se impartirán los días 7 y 11 de Septiembre.

CALENDARIO CELT 2017-2018

	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
SEP	4	5	6	7	8
		MATRICULA ON LINE		ABIERTA	
				Clase introdutora	Publicación listas definitivas
	11	12	13	14	15
	Clase introdutora LT1	MT1	XT1	JT1	VC1
	18	19	20	21	22
	LT2	MT2	XT2	JT2	VC2
	25	26	27	28	29
		MT3	XT3	JT3	VC3
OCT	2	3	4	5	6
	LT3	MT4	XT4	JT4	VC4
	9	10	11	12	13
	LT4	MT5	XT5	ENT1- XT	No Lectivo
	16	17	18	19	20
	LT5	MT6	XT6	JT5	VC5
	ENT1- LT			ENT1- JT	ENT1- VC
	23	24	25	26	27
	LT6	MT7	XT7	JT6	VC6
NOV	30	31	1	2	3
	LT7	MT8	ENT2- MT	JT7	VC7
	6	7	8	9	10
	LT8	JT8	XT8	ENT2- XT	VC8
	ENT2- LT	ENT2- JT			ENT2- VC
	13	14	15	16	17
	LT9	MT9	XT9	JT9	VC9
	20	21	22	23	24
	LT10	MT10	XT10	JT10	VC10
DIC	27	28	29	30	1
	LT11	MT11	XT11	JT11	VC11
	ENT3- LT	ENT3- MT	ENT3- XT	ENT3- JT	ENT3- VC
	4	5	6	7	8
	VC12	MT12		No Lectivo	
	11	12	13	14	15
	LT12	Turnos Libres	XT12	JT12	Turnos Libres
	18	19	20	21	22
		EXÁMENES ORALES			

2. DESCRIPCIÓN GENERAL

La señal **DCF77** es una señal de radio utilizada para el sincronismo de relojes que se transmite desde la ciudad alemana de Mainflingen, a una frecuencia de 77,5 KHz. La emisora es de gran potencia (50 KW) por lo que dicha señal puede captarse a distancias muy lejanas del transmisor (ha llegado a captarse hasta a 2000 Km de distancia). Como puede verse en la siguiente Figura 1, la zona de cobertura alcanza la práctica totalidad de la Península Ibérica.

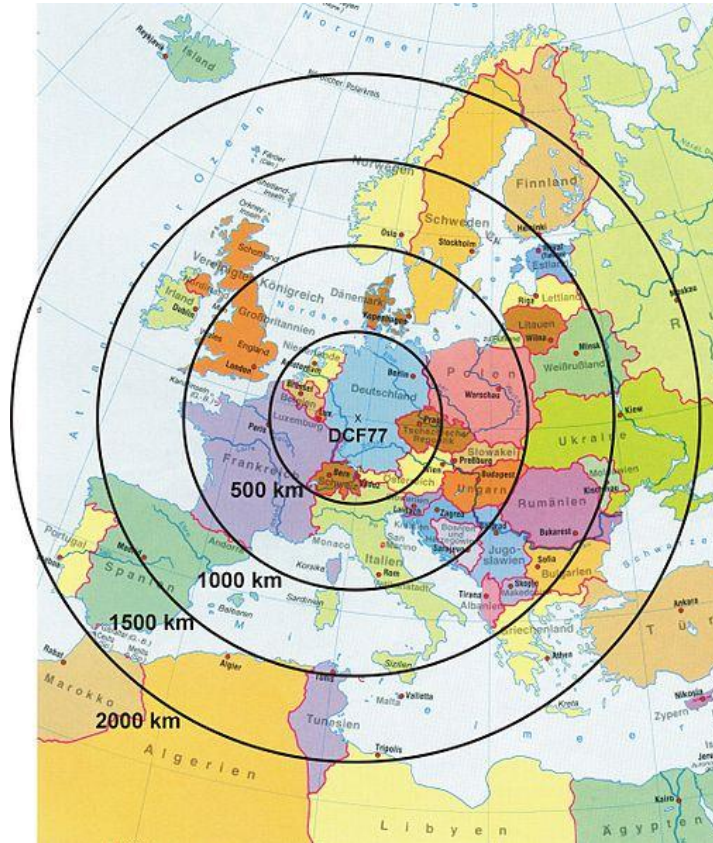


Figura 1 Cobertura de la señal DCF77

Dicha señal consta de una portadora modulada en amplitud por una señal de bits con una tasa binaria de 1 bit/s. Los datos digitales contienen información acerca de la hora, la fecha, el día de la semana y alguna información adicional y se obtienen a partir de un reloj atómico, por lo que la precisión es muy alta y está principalmente limitada por los retardos de propagación desde el transmisor a los distintos receptores.

La señal DCF77 está activa desde 1959, aunque se hizo muy popular en la década de los años 80 para la sincronización de relojes de uso doméstico y personal, donde la precisión no es extremadamente importante. Actualmente sigue siendo muy utilizada tanto en estos elementos, como en algunos instrumentos meteorológicos que pueden encontrarse en el mercado.

En este prototipo vamos a realizar un demodulador y decodificador a partir de una versión simplificada de la señal DCF77. Se proporcionará un archivo MP3 con una señal digital de hora y minutos (exclusivamente), modulada sobre una portadora de 1 KHz. Este archivo puede reproducirse con cualquier dispositivo portátil MP3 evitando de esta forma la construcción de un receptor complejo.

2.1 La señal DCF77 original

La señal original DCF77 consiste en una portadora de 77,5 KHz modulada en amplitud con un porcentaje de modulación del 25%. La modulación está formada por una trama de 59 bits transmitidos a una tasa de 1 bit/s. Cada bit tiene una duración de 100 ms (para el 0) o 200 ms (para el 1). Durante el segundo 60 se transmite la portadora sin modulación para poder determinar el comienzo de la trama binaria. Ver Figura 2.

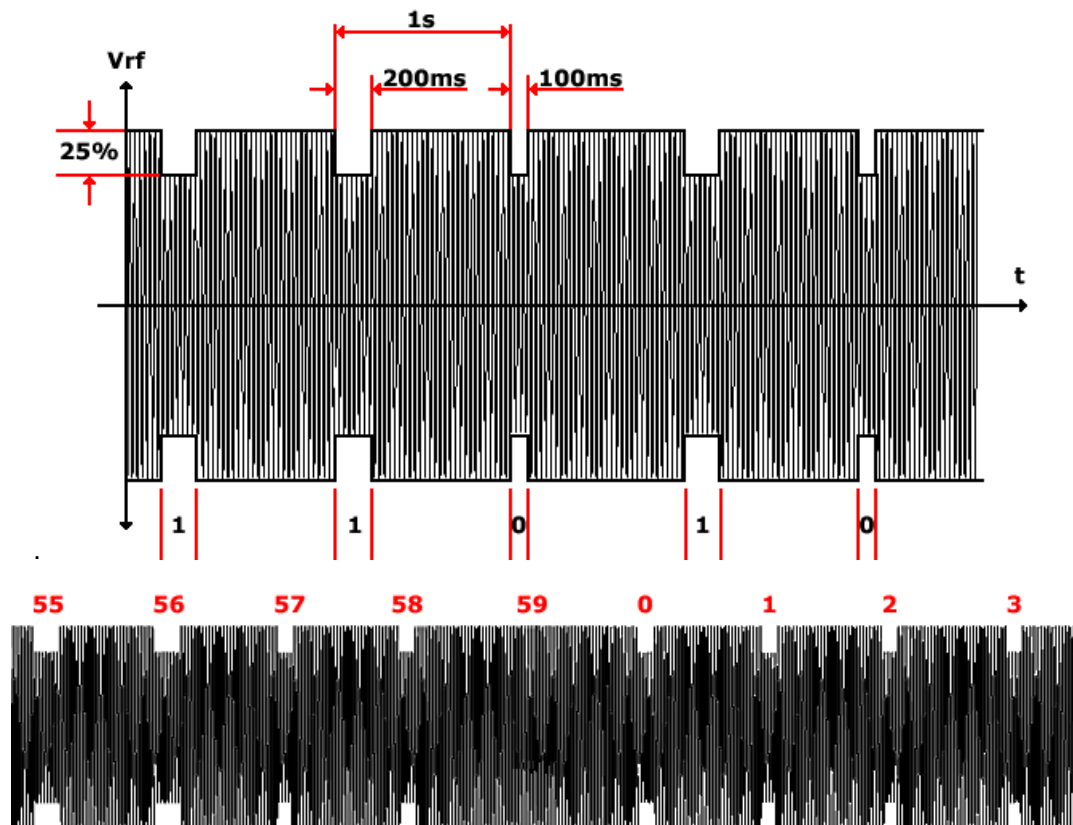


Figura 2 Forma de onda de la señal DCF77

Los detalles acerca de la información contenida en los bits quedan fuera del contenido de esta asignatura. No obstante, aquellos alumnos que tengan interés en conocer sus detalles pueden consultar las referencias al final de este documento.

2.2 La señal empleada en este proyecto

En este proyecto emplearemos una señal más simplificada que consistirá en una portadora de 1 KHz, modulada en amplitud con un porcentaje del 50%. La trama binaria se transmite a una tasa de 1 bit/s y consistirá en 15 bits. Los 14 primeros bits contendrán la hora y el minuto expresados en BCD, donde las decenas de la hora y el minuto se codifican con 3 bits y las unidades con 4 bits. Durante el segundo 15 la portadora no contiene modulación para poder determinar el comienzo de la trama siguiente (a partir de este momento nos referiremos al segundo 15 como sincronismo o SYNC).

Se proporciona un fichero mp3 que está disponible en la página web de la asignatura ([hora_19_30.mp3](#)) y que puede ser reproducido con cualquier aparato reproductor. Este fichero contiene la citada señal modulada con 10 tramas de 15 bits que se transmiten de forma continua. Cada trama comienza con la transmisión de la hora y a continuación un SYNC, empezando en las 19:30 y terminando en las 19:39. La duración total del fichero en tiempo es de 150 segundos.

En concreto, la hora 19:30 expresada en BCD con las decenas representadas con 3 bits y las unidades con 4 bits, resulta en la secuencia binaria: 001 1001 011 0000 (1 9 3 0 en decimal respectivamente). Estos son los 14 bits que componen la primera trama. Las siguientes se codifican de igual modo teniendo en cuenta el incremento del minuto desde 30 hasta 39. La Figura 3 muestra de forma gráfica la estructura de la señal contenida en el citado fichero mp3.

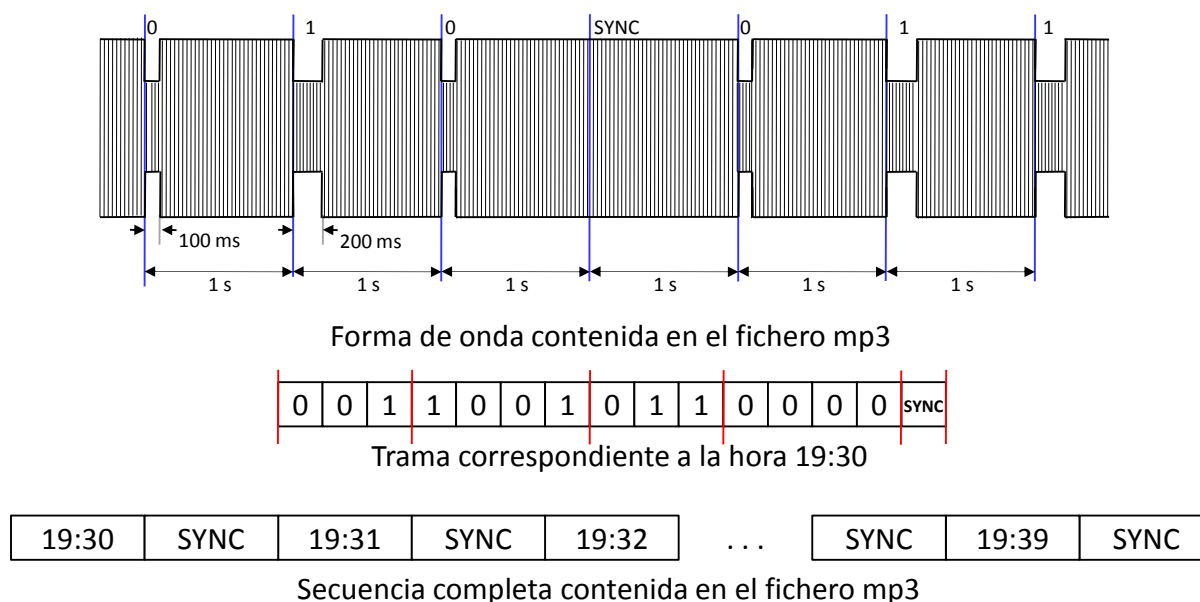


Figura 3 Forma de onda genérica, secuencia binaria correspondiente a la hora 19:30 y estructura del fichero mp3 suministrado.

2.3 Objetivo de este proyecto

El objetivo de este proyecto es diseñar y construir un sistema analógico/digital para demodular la señal horaria descrita en el apartado anterior y presentar su valor en 4 displays de 7 segmentos. El circuito analógico se realizará con componentes discretos, mientras que el hardware digital se desarrollará describiendo sus módulos en VHDL y sintetizándolo finalmente sobre la FPGA Spartan 3E contenida en la tarjeta BASYS2 del laboratorio.

El cometido del circuito analógico será tomar como entrada la señal de salida de un reproductor mp3 y demodularla para obtener una señal digital donde la portadora modulada al 50% se represente por un nivel de tensión de 0V y la portadora no modulada se represente por un nivel de tensión de 5 V.

El circuito digital deberá tomar como entrada esta secuencia de niveles de tensión y decodificarla para obtener la señal BCD correspondiente a la hora, presentándola en los 4 displays de 7 segmentos de la tarjeta BASYS2. Este circuito funcionará de forma continua de manera que cada 15 segundos se actualizará la lectura de la hora en los displays.

A modo de ejemplo, se presenta en la Figura 4 un esquema genérico de las partes que integran este prototipo. En los displays de la BASYS2 se muestra la hora (19:32).

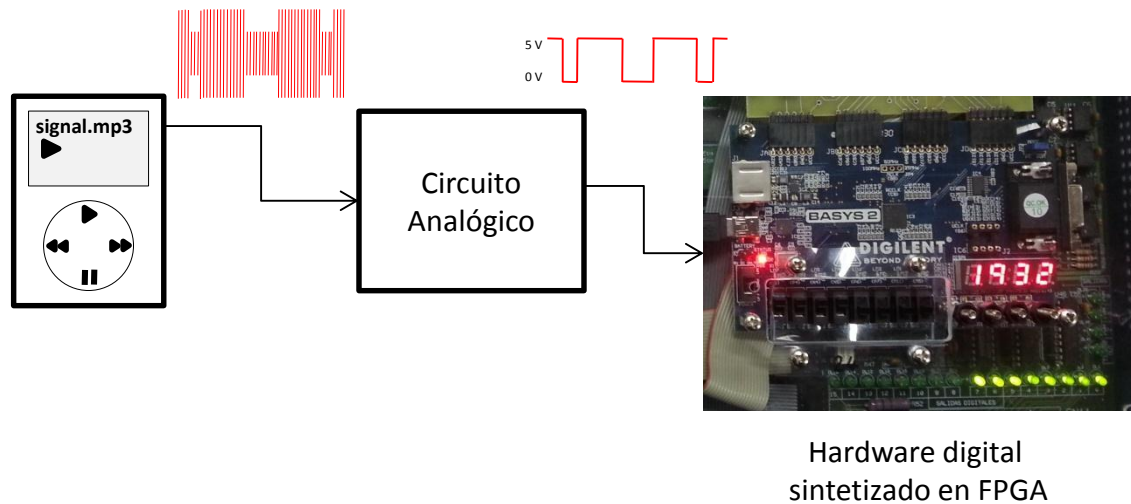


Figura 4 Esquema general del circuito propuesto, compuesto por un circuito analógico y un hardware digital sintetizado en FPGA.

2.4 Esquema detallado: circuito analógico

El circuito analógico deberá demodular la señal AM y obtener una señal digital con valores de tensión de 0 y 5V en concordancia con la amplitud de la portadora en cada instante del tiempo. En una primera aproximación, el citado circuito se compone de tres partes (ver Figura 5):

- Acondicionador de señal: este módulo toma la señal de salida de audio de un reproductor MP3 adaptando su impedancia. Posteriormente la amplifica y la filtra para obtener un nivel de tensión adecuado para el demodulador. Deberá ser ajustable para poder utilizar distintos reproductores en función de su nivel de señal a la salida.
- Demodulador AM: este módulo obtiene una señal de tensión proporcional a la amplitud de la portadora en cada instante de tiempo.
- Comparador: este módulo compara la señal de salida del demodulador con un umbral de tensión continua, dando como resultado 0V cuando dicha señal se encuentra por debajo de este umbral o 5V cuando se encuentra por encima.

Como resultado del circuito analógico se obtiene la señal que contiene los datos codificada con niveles de tensión que pueden ser procesados por un circuito digital.

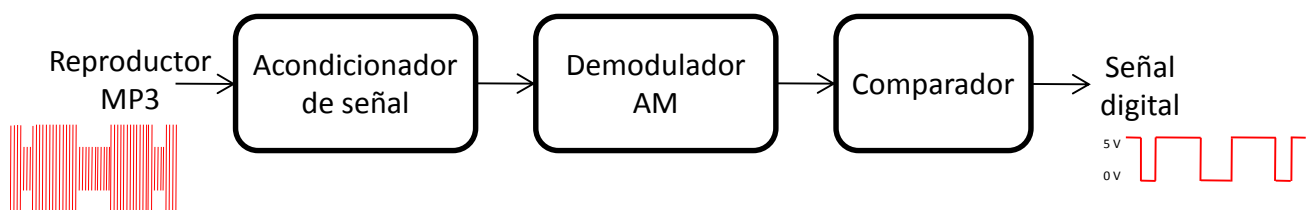


Figura 5 Esquema general de bloques del circuito analógico

2.5 Esquema detallado: circuito digital

El circuito digital toma como entrada la señal de datos obtenida por la parte analógica. A partir de esta señal debe extraerse la información contenida en cada periodo de bit de 1s ("0", "1" o SYNC) y presentar la hora en los displays. El funcionamiento debe ser continuo, de tal manera que se actualice periódicamente cada vez que llega una trama nueva.

El proceso que se empleará para decidir si la información contenida en un periodo de bit (1 seg) corresponde a un "0", un "1" o una señal SYNC será el siguiente: se tomarán 40 muestras de la señal entrante por cada periodo de bit y se integrará la energía acumulada en ese periodo (la energía se obtendrá a partir de la suma binaria de las muestras).

En el caso ideal de una señal pura sin ruido, debemos esperar los siguientes valores de energía:

- En un periodo de bit correspondiente a un SYNC (donde la señal mantiene su valor a nivel alto durante todo el tiempo), el total de la suma de muestras debe ser 40 (se muestrean 40 "1s").
- En un periodo de bit correspondiente a un "0" (donde la señal cae a nivel bajo durante 100 ms = 4 muestras), el valor de la suma debe ser 36 (se muestrean 36 "1s" y 4 "0s").
- En un periodo de bit correspondiente a un "1" (donde la señal cae a nivel bajo durante 200 ms = 8 muestras), el valor de la suma debe ser 32 (se muestrean 32 "1s" y 8 "0s").

Tras calcular la energía, se realizará una comparación de este valor con unos umbrales prefijados y a partir del resultado se decidirá si durante ese segundo se está transmitiendo un "0", un "1" o un SYNC (la Figura 6 muestra los valores de energía en diferentes periodos de bit).

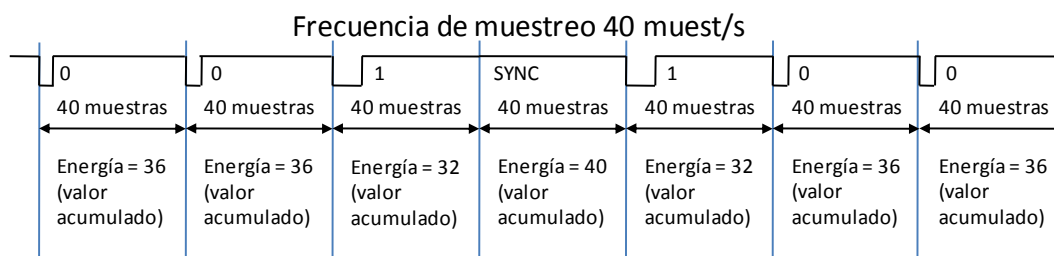


Figura 6 Valor de la energía de la señal en cada periodo de bit en función de los datos transmitidos

Este circuito digital se describirá en VHDL y se sintetizará sobre la FPGA del laboratorio y estará compuesto por los siguientes módulos (ver Figura 7):

- **Muestreo de señal:** este módulo se encarga de realizar el muestreo periódico de la señal a una frecuencia de 40 muestras por segundo. De esta forma tomará 40 muestras por cada periodo de bit (1 seg.). Además almacenará el valor de las muestras para su análisis posterior.
- **Integrador:** a partir de las muestras adquiridas se calcula la energía del periodo de bit integrando el valor de todas ellas. La integración se realizará sumando las muestras con un sumador binario.
- **Detección de umbral:** con el valor de la energía deberá decidirse si en ese periodo de bit se está transmitiendo un "0", un "1" o se trata del SYNC. Para ello se usarán dos comparadores con dos umbrales que se fijarán más adelante en este documento (apartado 4.2.1).
- **Registro de desplazamiento:** Una vez tomada la decisión, si se trata de un valor diferente de SYNC deberá almacenarse para la visualización posterior cuando toda la trama haya sido recibida. Esto se realizará mediante un registro de desplazamiento de carga serie y salida paralelo.
- **Registro (de validación):** Si el dato contenido en el periodo de bit es un SYNC, significa que la trama binaria ha terminado y por tanto se validará para su presentación en los displays. Esto se realizará mediante un registro que capturará el valor presente en el registro de desplazamiento.

- Visualización: Este módulo toma como entrada los valores BCD de la hora y el minuto (contenidos ya en el registro) y los presenta en los displays de 7 segmentos de la tarjeta BASYS 2. Utiliza un reloj de diferente frecuencia al muestreo para la actualización periódica de dichos displays
- Autómata de MOORE: Para el funcionamiento continuo de todos estos elementos es necesario un autómata secuencial de control que se encargue de cargar y validar los datos en función de la decisión tomada en los detectores de umbral.

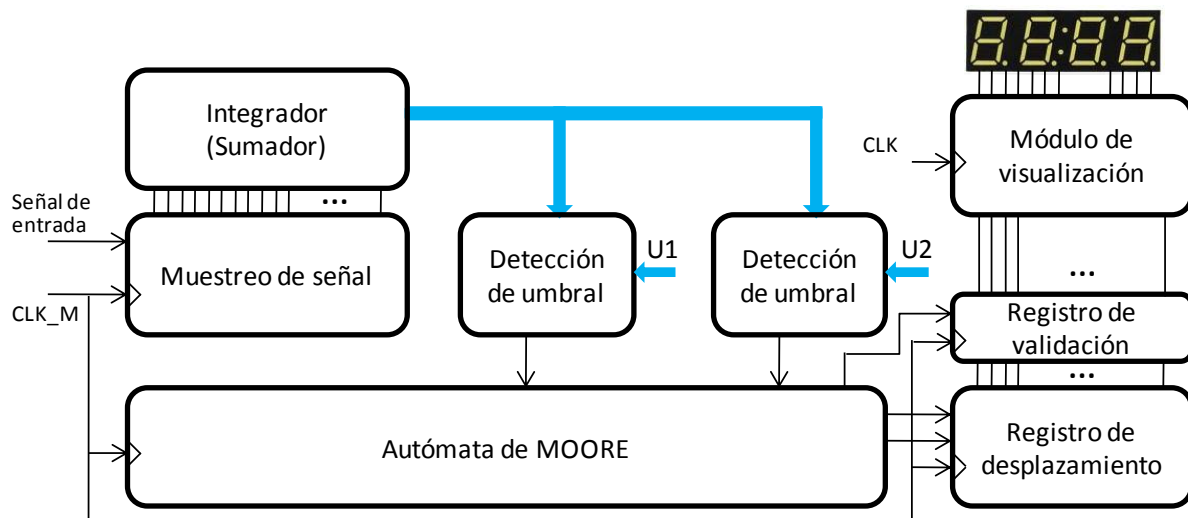


Figura 7 Esquema general de bloques de la parte digital

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:

<http://www.xilinx.com/products/silicon-devices/fpga/spartan-6.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 8). 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 de la práctica. Se utilizarán 3 en cada una de las alimentaciones: 100 μF , 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 μF .

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 advirtiendo del hecho. Observe la Figura 9 para la correcta conexión de estos elementos. Repetimos que debe prestar especial atención a la polaridad de los condensadores electrolíticos.

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).

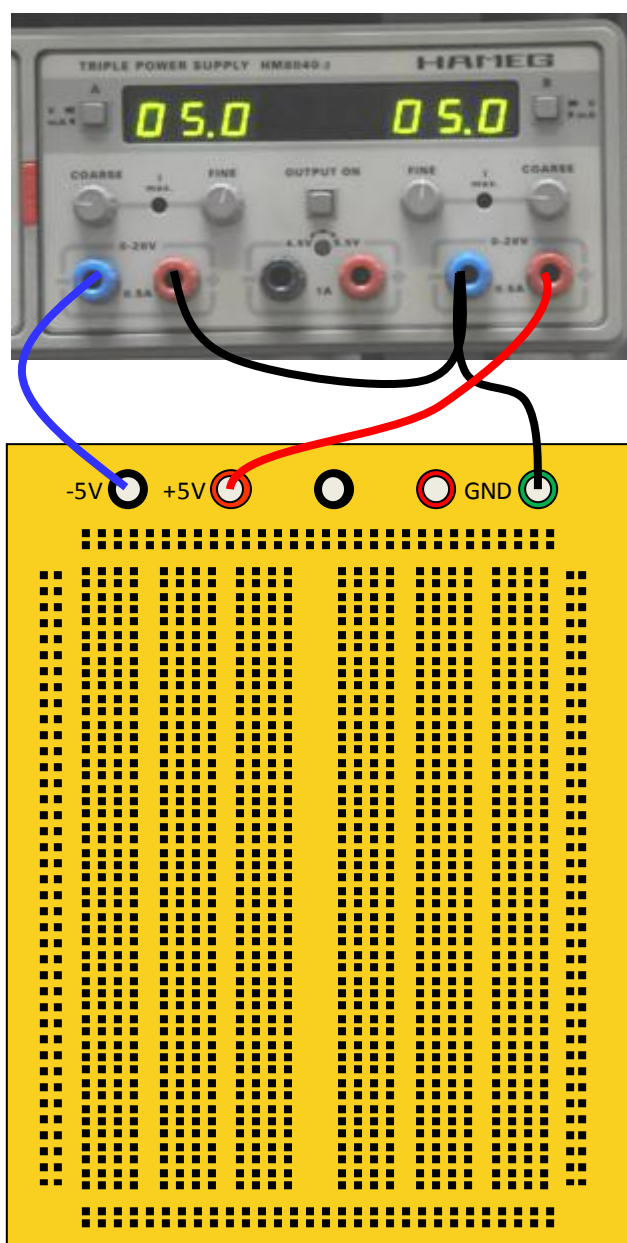


Figura 8 Esquema con la placa de inserción y la conexión de las bornas de alimentación.

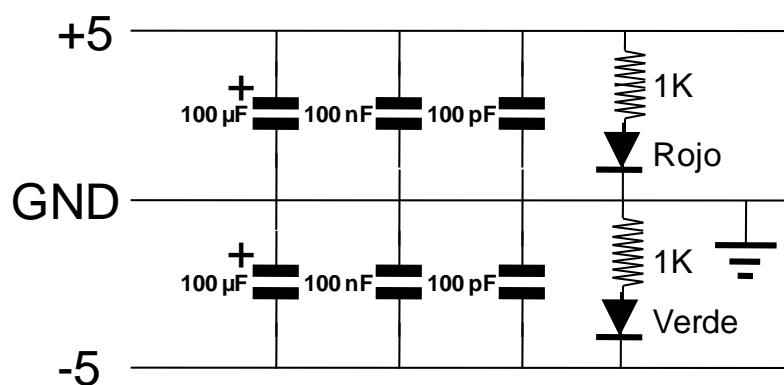


Figura 9 Condensadores de desacoplo y LEDs para la detección rápida de cortocircuitos. Tenga cuidado con la polaridad de los condensadores electrolíticos.

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 de la práctica:

- 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 se alimentarán con tensión simétrica de -5 y +5V.
- En los siguientes apartados se va a detallar el diseño de los diferentes módulos que componen el subsistema analógico. Tenga en cuenta que en cada caso se resumen los criterios generales de diseño, pero para tener información detallada deberá consultar las referencias indicadas en las referencias al final de este documento.
- 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 Acondicionador de señal

El acondicionador de señal está formado por dos partes: un amplificador y un filtro paso banda:

4.1.1.1 Amplificador y eliminación de posible componente continua:

Este amplificador (Figura 10) adapta la impedancia del reproductor MP3 y amplifica la señal para obtener 1 Vpp a su salida. La resistencia de 10 Ω simula la impedancia de los auriculares. El conjunto formado por el condensador de 1 μF y la resistencia R_f conforman un filtro paso alto que elimina la posible componente continua procedente del reproductor. A continuación, el amplificador aumenta el nivel de señal en función de su ganancia G. Deberá colocar un potenciómetro POT para poder conseguir ganancia variable en función del nivel de señal del dispositivo de entrada.

En primer lugar diseñe el valor de R_f que hace que la frecuencia de corte del citado filtro paso alto ($f_c = \frac{1}{2\pi \cdot R_f \cdot C}$) sea menor de 20 Hz (límite audible inferior). Un valor entre 10 y 15 Hz es aceptable.

Diseñe a continuación los valores de R_a y POT para conseguir una ganancia que permita obtener un nivel de señal de **1Vpp** a la salida del operacional. Ajuste el volumen de su reproductor y diseñe la etapa de tal modo que la citada ganancia se obtenga a mitad del recorrido del cursor del potenciómetro.

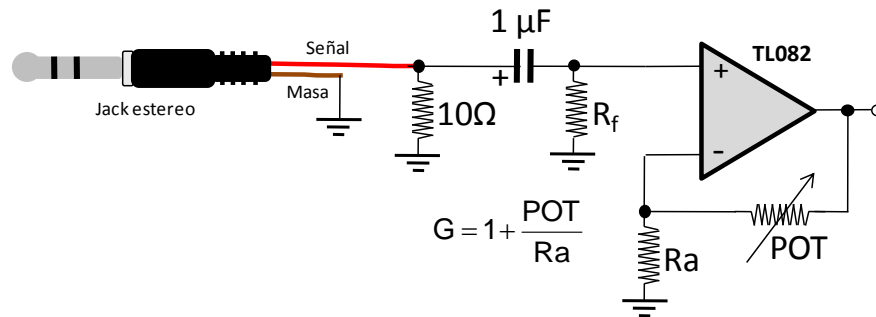


Figura 10 Amplificador y adaptador de impedancia del acondicionador de señal

Para conectar su reproductor al circuito necesitará comprar un conector “jack” estéreo y soldar los terminales correctamente. **No utilice en ningún caso un “jack” monoaural**, pues podría dañar el reproductor MP3. Para el correcto funcionamiento deberá soldar un cable a masa y otro a una cualquiera de las salidas (canal izquierdo L o derecho R), tal como se indica en la Figura 11.

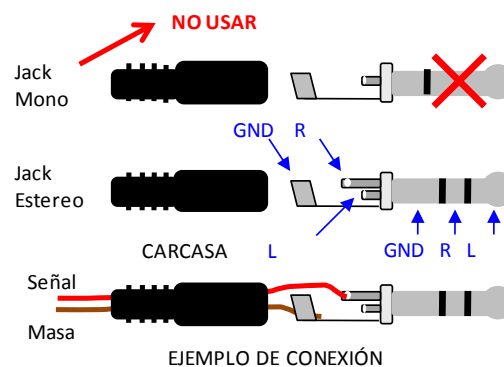


Figura 11 Conexión correcta del conector “jack” para el reproductor MP3.

4.1.1.2 Filtro paso banda:

Este bloque tiene como misión limitar en banda la señal que pasa al demodulador, reduciendo de este modo la posible interferencia producida por señales diferentes a la señal modulada de 1 KHz. En nuestro caso 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 12.

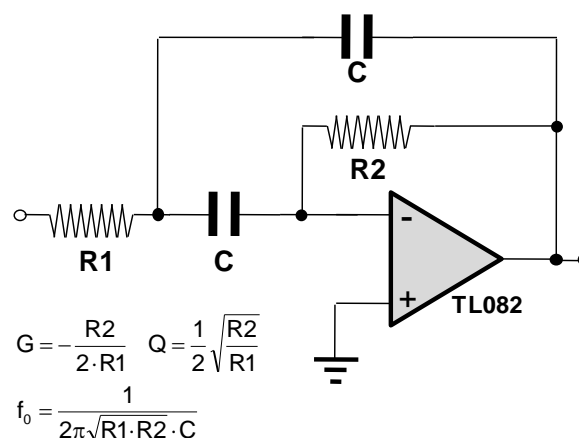


Figura 12 Filtro paso banda de realimentación múltiple de adaptación al canal.

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 **6Vpp**.

Este filtro deberá ser convenientemente caracterizado para la primera entrega (semana 5 o 6, dependiendo del grupo), tal y como se describe en el apartado 4.3.2

4.1.2 Recuperación de la señal de datos a partir de la señal modulada

Para entender el proceso de extracción de la señal de datos, es necesario comprender previamente ciertos conceptos de Teoría de la Señal que se resumen a continuación [1]:

La señal digital que contiene los datos, es una señal rectangular de periodo 1s. Dicha señal mantiene un nivel de tensión V durante gran parte del periodo, y los datos se codifican al principio de cada intervalo como tramos con un nivel de tensión más bajo ($0,75 \cdot V$) y diferente duración: 100ms para el "0", 200 ms para el "1" y 0 ms para el SYNC. Esta señal tiene un espectro en frecuencia paso bajo con componentes que se extienden hasta los 100 Hz aproximadamente, tal como se muestra en la Figura 13.

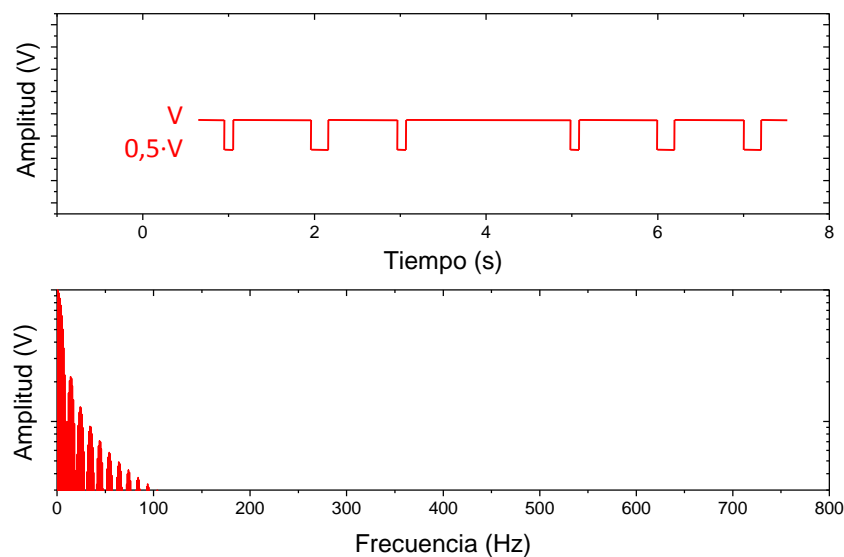


Figura 13 Señal de datos en banda base y espectro en frecuencia de la misma

No obstante, en el fichero mp3 esta señal de datos se encuentra modulada sobre una portadora senoidal de 1 KHz. Por tanto, su espectro en frecuencia estará centrado en torno a 1 KHz con unas bandas laterales que aparecen como consecuencia de dicha modulación (ver Figura 14).

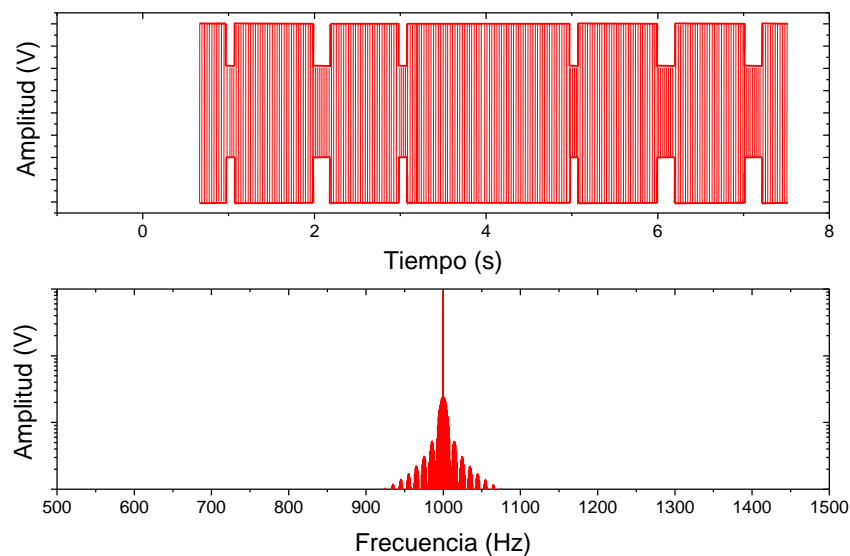


Figura 14 Portadora modulada por la señal de datos y espectro en frecuencia correspondiente

La información deseada se encuentra en las bandas laterales, por lo que para su extracción es necesario devolver la señal a su banda base original. Esto se realiza en varias etapas, la primera de las cuales consiste en rectificar la señal, quedándose con los semiciclos positivos de la misma. La Figura 15 muestra la señal rectificada y su espectro en frecuencia.

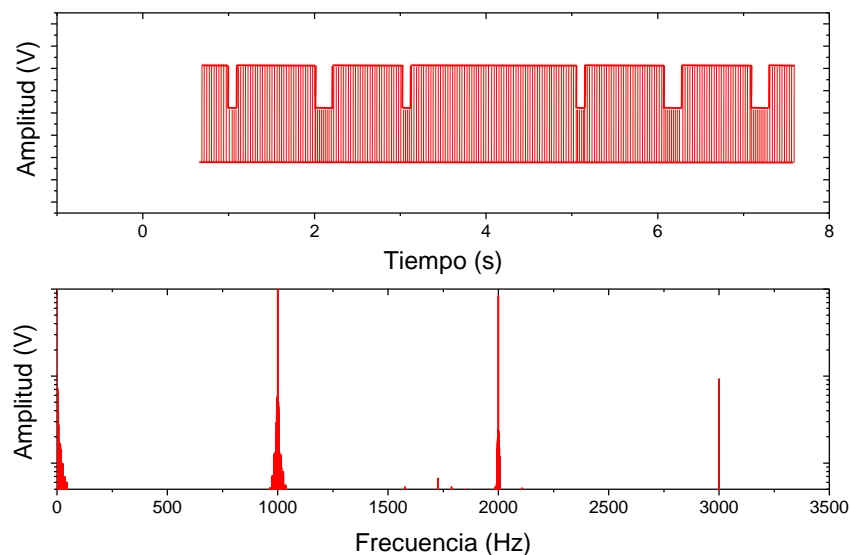


Figura 15 Señal obtenida tras la rectificación y espectro en frecuencia.

Esta señal ya posee componentes en banda base correspondientes a los datos, además de la portadora modulada y algunos armónicos de la misma. Para recuperar la señal de datos es necesario eliminar todas las componentes de alta frecuencia, quedándose exclusivamente con la parte paso bajo del espectro. Esto se realiza mediante un filtro paso bajo que atenúe las componentes por encima de 100 Hz (ancho de banda original de la señal banda base tal como se muestra en la Figura 13).

No obstante, si se utiliza un filtro con dos polos, no será posible eliminar completamente todas las componentes de frecuencia superior a la de corte, por lo que la señal recuperada tendrá flancos exponenciales y un rizado adicional. Para el citado filtro de orden 2 se obtienen las señales que se muestran a continuación en la Figura 16.

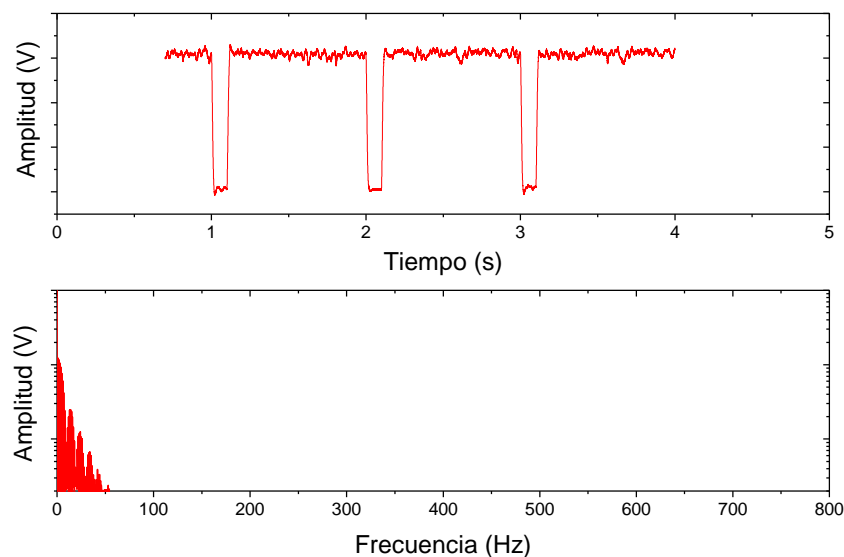


Figura 16 Señal obtenida tras un filtrado paso bajo de orden 2 y espectro en frecuencia.

4.1.3 El demodulador AM

El proceso descrito en el apartado anterior se conoce como demodulación. En nuestro caso, tal como se ha detallado, vamos a construir un demodulador de amplitud muy sencillo basado en un rectificador, un filtro paso bajo, y un comparador (vea la Figura 17).

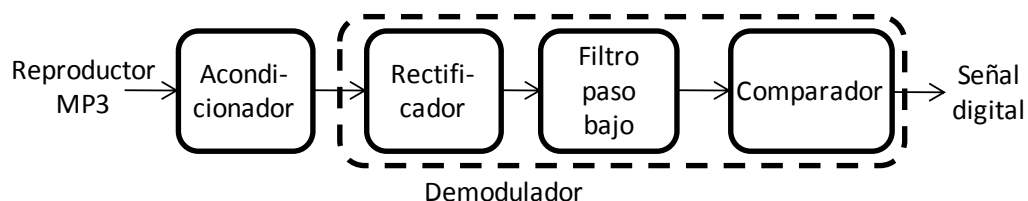


Figura 17 Esquema del demodulador

4.1.4 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 ($V > 0,6 \text{ V}$). Se utilizará un diodo **1N4148** que tiene buena respuesta a la frecuencia de la portadora.

En la Figura 18 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 K Ω es suficiente).

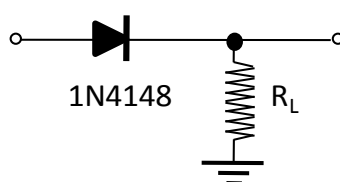


Figura 18 Rectificador construido con un diodo.

4.1.5 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 19.

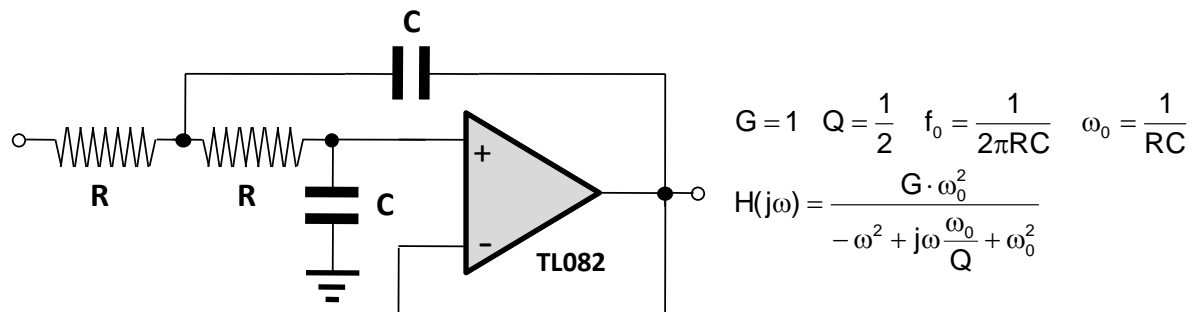


Figura 19 Filtro paso bajo Sallen-Key

Recuerde que este filtro se emplea para atenuar la portadora senoidal de 1 KHz y sus armónicos, dejando pasar la banda base correspondiente a la señal de datos. Teniendo esto en cuenta, siga la siguiente secuencia para su diseño:

- 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 $f_0 = 2\pi \cdot \omega_0$ 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.
- Calcule en primer lugar el módulo de la función de transferencia $|H(j\omega)|$.
- Escoja un valor para la frecuencia de corte del filtro (f_c)
- 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.

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

- 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 o 9 dependiendo del grupo), tal y como se describe en el apartado 4.3.2

4.1.6 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 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 20). Este circuito actúa como un operacional en bucle abierto, es decir, cuando la tensión en $V+$ supera a la tensión en $V-$ entrega $+V_{cc}$ a la salida. Cuando sucede lo contrario entrega 0V. **Es imprescindible colocar una resistencia de pull up de 1K entre V_{cc} 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.

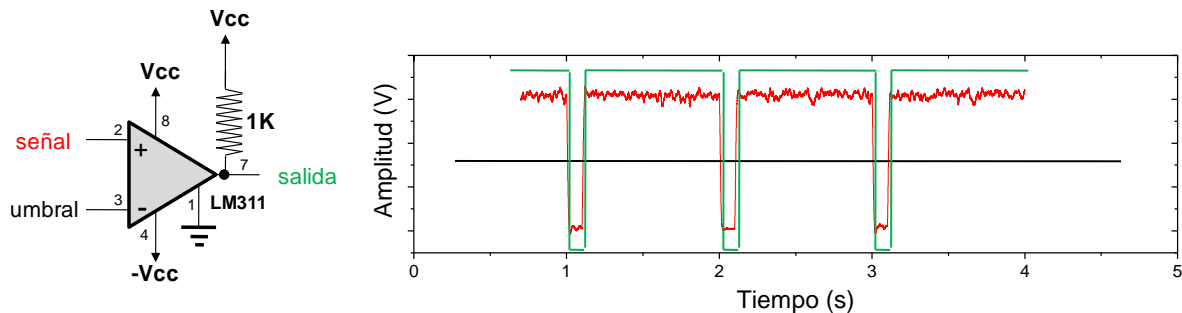


Figura 20 El comparador LM311 y esquema del proceso de comparación

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 de la práctica:

- 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 14.7 **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.
- **Parte de los módulos que integran este proyecto se encuentran ya desarrollados en los citados ejemplos.** Por tanto siempre puede utilizarlos como referencia y ayuda para describir su hardware. Ponga atención y dedique tiempo a entender su funcionamiento durante las primeras semanas mientras se trabaja en la parte analógica.
- **No utilice nunca letras acentuadas, espacios, ñ o ü en la ruta de carpetas donde se encuentra su proyecto puesto que el entorno ISE falla si se emplean esos caracteres.**
- **Cree su proyecto dentro de la carpeta c:\alumnos. Es la única carpeta con permisos de escritura.**
- **Llévese consigo todos los días el código que vaya generando (en un pendrive o a través de las distintas opciones por Internet). La carpeta alumnos se borrará periódicamente sin previo aviso.**

4.2.1 Funcionamiento del subsistema digital

La estructura de bloques del subsistema digital se muestra a continuación con más detalle en la Figura 21.

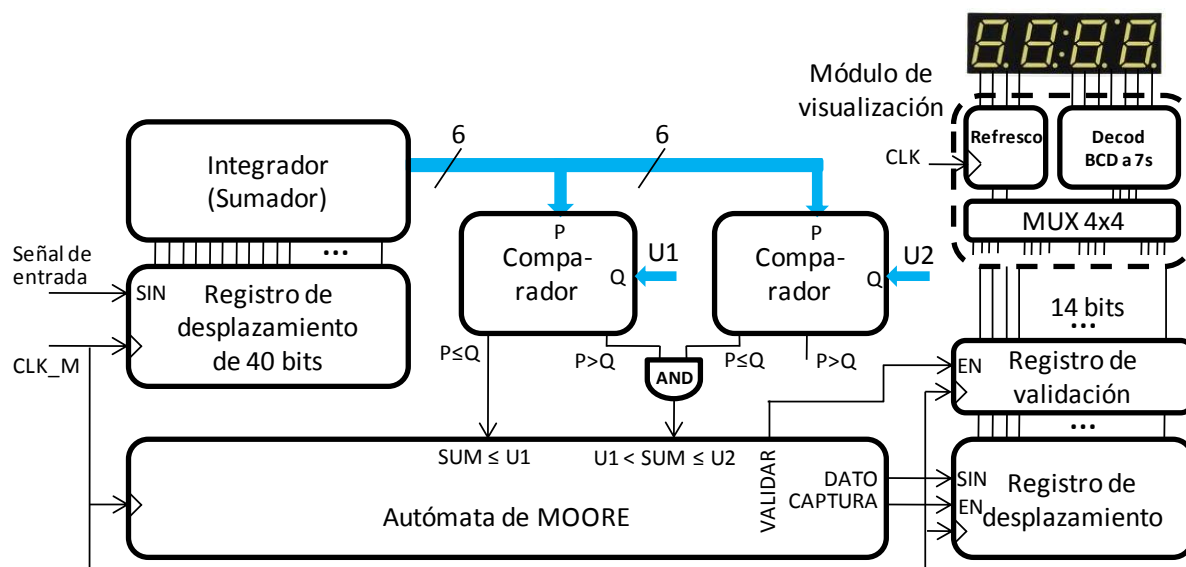


Figura 21 Esquema de bloques del subsistema digital completo

Como puede verse, se compone de un registro de desplazamiento de 40 bits que captura la señal de entrada a una frecuencia de 40 muestras/s, de esta forma se obtienen 40 muestras en cada tiempo de bit.

Por otro lado, un sumador calcula el valor de la suma binaria de las 40 muestras y obtiene un número binario de 6 bits (para poder representar valores decimales entre 0 y 40).

Dicho número de 6 bits se compara con dos umbrales para poder distinguir si en ese tiempo de bit se está transmitiendo un "0", un "1" o es una señal SYNC. Como se indicaba en el apartado 2.4 de este enunciado, la energía (suma) correspondiente es diferente en cada caso. Para una señal perfecta sin ruido, el valor de la energía es de 40 para la señal SYNC, 36 para el "0" y 32 para el "1". No obstante, la señal recibida dependerá en general de la calidad del circuito analógico realizado, y por tanto pueden darse casos donde los valores de energía sean ligeramente diferentes. Es razonable entonces, situar dos umbrales de decisión (U1 y U2) en los valores intermedios de las energías para el caso ideal, de tal modo que si llamamos SUM al valor obtenido por el sumador, se tiene:

- **Umbral U1** = valor intermedio entre 32 y 36 = **34**
- **Umbral U2** = valor intermedio entre 36 y 40 = **38**

Por tanto:

- Si $SUM \leq U1$ decidimos que se trata de un "1".
- Si $SUM > U1$ y $SUM \leq U2$ (es decir, $U1 < SUM \leq U2$) decidimos que se trata de un "0".
- Si $SUM > U2$ decidimos que se trata de un SYNC.

El autómata espera por la recepción de las 40 muestras y una vez recibidas decide el tipo de dato contenido en ese tiempo de bit. Si se trata de un "0" o un "1" lo almacena en el registro de desplazamiento de 14 bits poniendo su valor en la salida DATO y activando el ENABLE del citado registro. Si se trata de un SYNC, entonces activa la señal VALIDAR, la cual permite la captura por parte del registro colocado a la salida del de desplazamiento, validando la trama.

No obstante, para que este procedimiento funcione, debemos asegurarnos que la ventana de 40 muestras está sincronizada correctamente con la señal entrante de tal modo que el pulso completo a nivel bajo se encuentra dentro de la misma. La Figura 22 muestra un ejemplo de señal entrante y la

ventana de muestreo (40 muestras). Si se comienza el muestreo en un momento aleatorio del tiempo puede ocurrir que dentro de la citada ventana se adquieran muestras del pulso correspondiente a ese tiempo de bit con algunas correspondientes al siguiente. En ese caso el valor de la suma será incorrecto (en rojo en la figura). Existe un intervalo temporal (marcado también en la figura), dentro del cual puede comenzar el muestreo, puesto que asegura que todo el pulso a nivel bajo se encuentra dentro de la ventana.

Una vez sincronizado, el muestreo puede continuar indefinidamente.

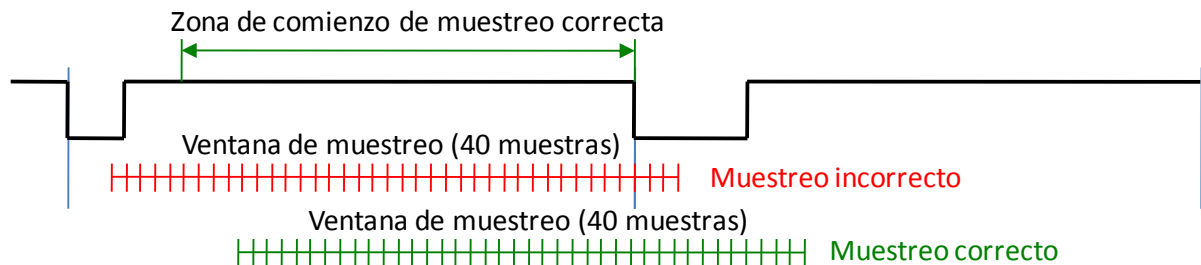


Figura 22 Ejemplos de muestreo correcto e incorrecto sobre la señal entrante

La forma de asegurar el sincronismo inicial consiste en muestrear constantemente hasta que la energía acumulada se interpreta como un SYNC. En ese momento debemos desplazarnos a la zona de comienzo de muestreo correcta. Como puede verse en la Figura 23 bastaría con adquirir 5 muestras para entrar en esta región. No obstante, para hacer el sistema más robusto esperaremos 20 muestras para conseguir que los pulsos a nivel bajo se encuentren centrados en la ventana de muestreo.

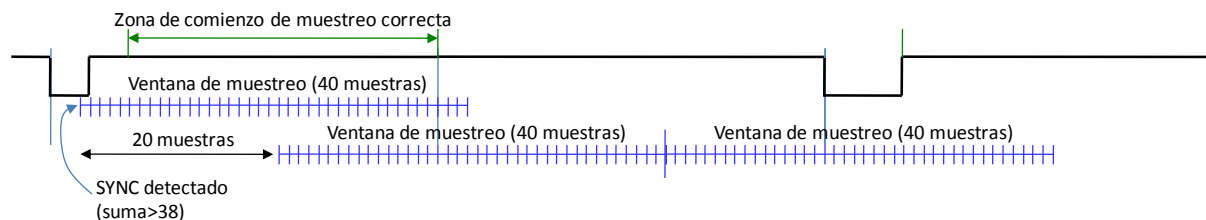


Figura 23 Forma de conseguir el sincronismo entre la señal entrante y el muestreo del autómata

El autómata debe por tanto realizar inicialmente la sincronización y posteriormente entrar en un bucle infinito donde tome 40 muestras y realice la decisión acerca del dato transmitido.

El módulo de visualización presenta en los displays los valores BCD contenidos a la salida del registro, por lo tanto cambiará cada vez que se valide un dato nuevo, lo cual ocurrirá siempre que se reciba una señal SYNC, cada 15 seg.

4.2.2 Estructura del código VHDL

Para la realización de este proyecto deberá describir diversos módulos en VHDL. Todos estos módulos deberán ser posteriormente conectados entre sí para generar el hardware digital completo. Con el objeto de normalizar la implementación del hardware digital en VHDL, se sugiere al alumno que siga la siguiente estructura de archivos:

- principal.vhd: fichero con descripción arquitectural (conexiones) de interconexión de módulos:
 - gen_reloj.vhd: divisor del reloj del sistema para obtener la frecuencia adecuada
 - reg_desp40.vhd: registro de desplazamiento de 40 bits
 - sumador40.vhd: sumador de 40 bits de entrada
 - comparador.vhd: comparador binario
 - AND2.vhd: puerta AND de 2 entradas

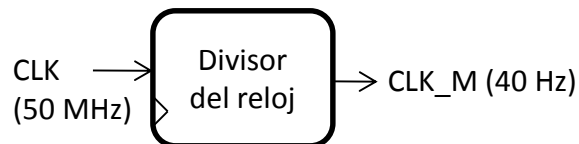
- reg_desp.vhd: registro de desplazamiento para almacenar la trama
- registro.vhd: registro que captura la trama, validándola, cuando está completa
- autómata.vhd: autómata que controla la captura y validación de la trama
- visualización.vhd: fichero arquitectural que implementa el módulo de visualización:
 - MUX4x4.vhd: Multiplexor de 4 entradas de 4 bits de datos
 - decod7s.vhd: decodificador de BCD a 7 segmentos
 - refresco.vhd: circuito secuencial que refresca periódicamente los displays

4.2.3 Generador de reloj:

El circuito digital propuesto necesita una señal de reloj (CLK_M) para muestrear la señal de entrada a 40 Hz.

La FPGA posee un oscilador interno muy estable que produce una frecuencia de 50 MHz. La frecuencia necesaria se obtendrá mediante división de la frecuencia del reloj utilizando contadores binarios.

El módulo que realiza la división responde por tanto al siguiente esquema y su descripción en VHDL se presenta también a continuación:



FICHERO: gen_reloj.vhd

```

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

entity gen_reloj is
    Port ( CLK      : in  STD_LOGIC;    -- Reloj de la FPGA
          CLK_M     : out STD_LOGIC);   -- Reloj de muestreo
end gen_reloj;

architecture a_gen_reloj of gen_reloj is

    signal cont_M : STD_LOGIC_VECTOR (31 downto 0) := (others=>'0'); -- contador 1
    signal S_M    : STD_LOGIC := '0';

begin
    PROC_CONT : process (CLK)
    begin
        if CLK'event and CLK='1' then
            cont_M <= cont_M + 1;

            if cont_M >= AAAAA then -- división de frecuencia a 40 Hz
                S_M <= not S_M;
                cont_M <= (others=>'0');
            end if;
        end if;
    end process;

    CLK_M <= S_M;

end a_gen_reloj;
  
```

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

Para comprobar que este módulo funciona correctamente asocie la salida CLK_M 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

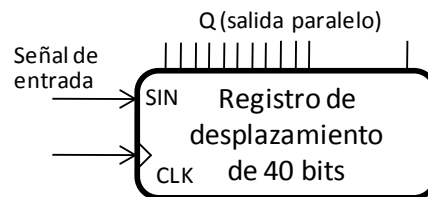
NET "CLK" LOC = "M6"; #Señal de reloj del sistema

#Salidas externas donde se visualiza la señal de reloj dividida

NET "CLK_M" LOC = "A9"; #Salida terminal 14
```

4.2.4 Registro de desplazamiento de 40 bits:

Este módulo será empleado para capturar las 40 muestras correspondientes a un tiempo de bit. Se trata de un registro de desplazamiento con entrada serie y salida paralelo. El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación.



FICHERO: reg_desp40.vhd

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

entity reg_desp40 is
    Port ( SIN : in  STD_LOGIC;  -- Datos de entrada serie
          CLK : in  STD_LOGIC;  -- Reloj de muestreo
          Q  : out STD_LOGIC_VECTOR (39 downto 0)); -- Salida paralelo
end reg_desp40;

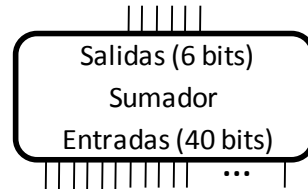
architecture a_reg_desp40 of reg_desp40 is
begin
    .....
end a_reg_desp40;
```

Complete la descripción funcional de este módulo en VHDL.

4.2.5 Sumador de 40 bits de entrada:

El sumador calcula la energía acumulada durante el periodo de bit y para ello suma todas las muestras capturadas durante ese tiempo. Se trata de un sumador de 40 bits de entrada y 6 bits de salida, puesto que dicha salida podrá tomar valores entre 0 y 40.

El esquema del módulo se muestra a continuación. Dado que su descripción en VHDL necesita de conocimientos algo más extensos sobre la arquitectura de la FPGA, se le suministra completo también a continuación.



FICHERO: sumador40.vhd

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

entity sumador40 is
    Port ( ENT : in  STD_LOGIC_VECTOR (39 downto 0); -- entradas (40 bits)
          SAL : out  STD_LOGIC_VECTOR (5 downto 0)); -- salida (6 bits)
end sumador40;

architecture a_sumador40 of sumador40 is

    signal E0 : STD_LOGIC_VECTOR (5 downto 0);
    signal E1 : STD_LOGIC_VECTOR (5 downto 0);
    signal E2 : STD_LOGIC_VECTOR (5 downto 0);
    signal E3 : STD_LOGIC_VECTOR (5 downto 0);
    signal E4 : STD_LOGIC_VECTOR (5 downto 0);
    signal E5 : STD_LOGIC_VECTOR (5 downto 0);
    signal E6 : STD_LOGIC_VECTOR (5 downto 0);
    signal E7 : STD_LOGIC_VECTOR (5 downto 0);
    signal E8 : STD_LOGIC_VECTOR (5 downto 0);
    signal E9 : STD_LOGIC_VECTOR (5 downto 0);
    signal E10 : STD_LOGIC_VECTOR (5 downto 0);
    signal E11 : STD_LOGIC_VECTOR (5 downto 0);
    signal E12 : STD_LOGIC_VECTOR (5 downto 0);
    signal E13 : STD_LOGIC_VECTOR (5 downto 0);
    signal E14 : STD_LOGIC_VECTOR (5 downto 0);
    signal E15 : STD_LOGIC_VECTOR (5 downto 0);
    signal E16 : STD_LOGIC_VECTOR (5 downto 0);
    signal E17 : STD_LOGIC_VECTOR (5 downto 0);
    signal E18 : STD_LOGIC_VECTOR (5 downto 0);
    signal E19 : STD_LOGIC_VECTOR (5 downto 0);
    signal E20 : STD_LOGIC_VECTOR (5 downto 0);
    signal E21 : STD_LOGIC_VECTOR (5 downto 0);
    signal E22 : STD_LOGIC_VECTOR (5 downto 0);
    signal E23 : STD_LOGIC_VECTOR (5 downto 0);
    signal E24 : STD_LOGIC_VECTOR (5 downto 0);
    signal E25 : STD_LOGIC_VECTOR (5 downto 0);
```

```

signal E26 : STD_LOGIC_VECTOR (5 downto 0);
signal E27 : STD_LOGIC_VECTOR (5 downto 0);
signal E28 : STD_LOGIC_VECTOR (5 downto 0);
signal E29 : STD_LOGIC_VECTOR (5 downto 0);
signal E30 : STD_LOGIC_VECTOR (5 downto 0);
signal E31 : STD_LOGIC_VECTOR (5 downto 0);
signal E32 : STD_LOGIC_VECTOR (5 downto 0);
signal E33 : STD_LOGIC_VECTOR (5 downto 0);
signal E34 : STD_LOGIC_VECTOR (5 downto 0);
signal E35 : STD_LOGIC_VECTOR (5 downto 0);
signal E36 : STD_LOGIC_VECTOR (5 downto 0);
signal E37 : STD_LOGIC_VECTOR (5 downto 0);
signal E38 : STD_LOGIC_VECTOR (5 downto 0);
signal E39 : STD_LOGIC_VECTOR (5 downto 0);

begin

    E39<="00000"&ENT(39);
    E38<="00000"&ENT(38);
    E37<="00000"&ENT(37);
    E36<="00000"&ENT(36);
    E35<="00000"&ENT(35);
    E34<="00000"&ENT(34);
    E33<="00000"&ENT(33);
    E32<="00000"&ENT(32);
    E31<="00000"&ENT(31);
    E30<="00000"&ENT(30);
    E29<="00000"&ENT(29);
    E28<="00000"&ENT(28);
    E27<="00000"&ENT(27);
    E26<="00000"&ENT(26);
    E25<="00000"&ENT(25);
    E24<="00000"&ENT(24);
    E23<="00000"&ENT(23);
    E22<="00000"&ENT(22);
    E21<="00000"&ENT(21);
    E20<="00000"&ENT(20);
    E19<="00000"&ENT(19);
    E18<="00000"&ENT(18);
    E17<="00000"&ENT(17);
    E16<="00000"&ENT(16);
    E15<="00000"&ENT(15);
    E14<="00000"&ENT(14);
    E13<="00000"&ENT(13);
    E12<="00000"&ENT(12);
    E11<="00000"&ENT(11);
    E10<="00000"&ENT(10);
    E9<="00000"&ENT(9);
    E8<="00000"&ENT(8);
    E7<="00000"&ENT(7);
    E6<="00000"&ENT(6);
    E5<="00000"&ENT(5);
    E4<="00000"&ENT(4);
    E3<="00000"&ENT(3);
    E2<="00000"&ENT(2);
    E1<="00000"&ENT(1);
    E0<="00000"&ENT(0);

    SAL<=E39+E38+E37+E36+E35+E34+E33+E32+E31+E30+
        E29+E28+E27+E26+E25+E24+E23+E22+E21+E20+
        E19+E18+E17+E16+E15+E14+E13+E12+E11+E10+

```

```
E9+E8+E7+E6+E5+E4+E3+E2+E1+E0;
```

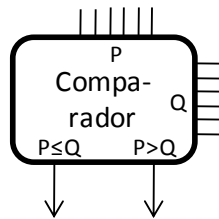
```
end a_sumador40;
```

Utilice este código tal cual se muestra.

4.2.6 Comparador:

El comparador debe tomar dos entradas de 6 bits (P y Q) y entregar dos bits como valores de salida. Uno de ellos indicará la condición $P \leq Q$ y el otro la condición $P > Q$. Ambas señales deben ser activas a nivel alto, esto es: $P > Q$ será "1" cuando P sea mayor que Q y "0" en caso contrario.

El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación:



FICHERO: comparador.vhd

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

entity comparador is
    Port ( P : in  STD_LOGIC_VECTOR (5 downto 0); -- Entrada P
          Q : in  STD_LOGIC_VECTOR (5 downto 0); -- Entrada Q
          PGTQ : out  STD_LOGIC; -- Salida P>Q
          PLEQ : out  STD_LOGIC; -- Salida P≤Q
end comparador;

architecture a_comparador of comparador is

begin
    . . . . .
end a_comparador;
```

Complete la descripción funcional de este módulo en VHDL.

4.2.7 Puerta AND:

La puerta AND permite obtener la condición $U1 < SUM \leq U2$ a partir de las salidas de los dos comparadores. El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación:

FICHERO: AND2.vhd

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

entity AND_2 is
    Port ( A : in  STD_LOGIC; -- Entrada A
```



```

        B : in  STD_LOGIC;    -- Entrada B
        S : out  STD_LOGIC);  -- Salida
end AND_2;

architecture a_AND_2 of AND_2 is
begin
    . . . . .
end a_AND_2;

```

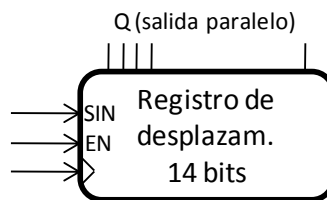
Complete la descripción funcional de este módulo en VHDL.

4.2.8 Registro de desplazamiento:

Este registro, controlado por el autómata, será el encargado de ir almacenando los valores “0” y “1” que componen la trama y que darán lugar a la hora que se representará en los displays. Dado que funciona conectado al reloj principal del sistema (CLK_M), es necesario utilizar una patilla de ENABLE para que el autómata pueda activar su funcionamiento al final de cada periodo de bit.

Se trata de un registro de desplazamiento de entrada serie y salida paralelo de 14 bits (la trama completa contiene 14 bits), con entrada de ENABLE. Este registro solamente desplazará los datos cuando la señal ENABLE se encuentre activa a nivel alto. En otro caso, la salida no varía, independientemente del reloj.

El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación:



FICHERO: reg desp.vhd

```

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

entity reg_desp is
    Port ( SIN : in  STD_LOGIC;  -- Datos de entrada serie
          CLK : in  STD_LOGIC;  -- Reloj
          EN  : in  STD_LOGIC;  -- Enable
          Q   : out STD_LOGIC_VECTOR (13 downto 0)); -- Salida paralelo
end reg_desp;

architecture a_reg_desp of reg_desp is
begin
    . . . . .
end a_reg_desp;

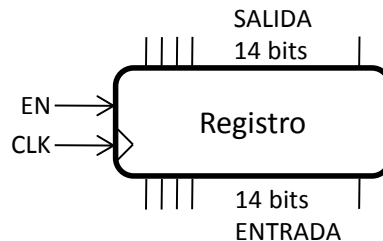
```

Complete la descripción funcional de este módulo en VHDL.

4.2.9 Registro (de validación):

El registro es un conjunto de 14 biestables tipo D en paralelo que se activan con una señal de reloj común. Este dispositivo copia la entrada en su salida cuando se recibe un flanco positivo del reloj y la señal Enable está activa. En otro caso, la salida no varía. **Tenga en cuenta que debe ser activo por flanco** y no por nivel (se trata de un registro, no de un latch)

El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación:



FICHERO: registro.vhd

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

entity registro is
    Port ( ENTRADA : in  STD_LOGIC_VECTOR (13 downto 0); -- Entradas
          SALIDA  : out STD_LOGIC_VECTOR (13 downto 0); -- Salidas
          EN      : in  STD_LOGIC;                    -- Enable
          CLK     : in  STD_LOGIC);                   -- Reloj
end registro;

architecture a_registro of registro is
begin
    . . . . .
end a_registro;
```

Complete la descripción funcional de este módulo en VHDL.

4.2.10 Autómata de MOORE:

La descripción del funcionamiento del autómata ha quedado establecida en el apartado 4.2.1 de este enunciado. Según esta descripción deberá realizar las siguientes tareas:

1. Realizar el sincronismo inicial de la señal entrante con la ventana de muestreo
2. Leer 40 muestras
3. Tomar la decisión sobre el dato transmitido
4. Si se trata de "0" o "1" deberá almacenarlo en el registro de desplazamiento
5. Si se trata de un SYNC validará la trama activando el registro
6. En cualquier caso, vuelve a 2.

Este procedimiento se plasma en el siguiente diagrama de estados del autómata (vea Figura 24):

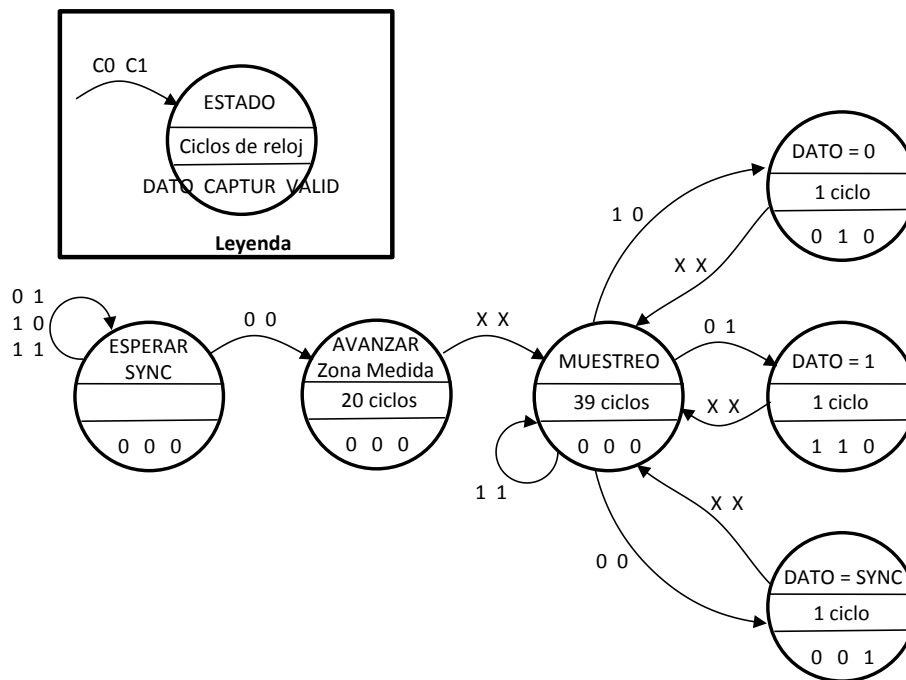


Figura 24 Diagrama de estados del autómata

Este autómata tiene las siguientes características:

Entradas:

C0 = ($SUM > U1$ y $SUM \leq U2$) y **C1** = ($SUM \leq U1$) que son las condiciones de decisión para el “0” y el “1” respectivamente. El caso 00 indica que se trata de una señal SYNC ($SUM > U2$), y el caso 11 no debería darse nunca (pues la suma no puede ser al mismo tiempo menor que $U1$ y estar entre $U1$ y $U2$).

Ciclos de reloj:

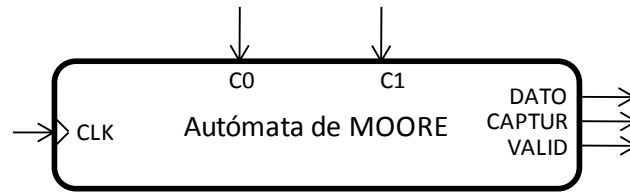
Indica los ciclos de reloj que dura ese estado. Lo normal es que un estado dure un ciclo de reloj. Para esperar varios ciclos de reloj tendríamos que colocar tantos estados como ciclos necesitemos. Existe otra forma más sencilla que consiste en hacer que el autómata espere dentro de un estado durante varios ciclos de reloj. Esto es fácil de codificar en VHDL y más adelante se indica cómo hacerlo. En este caso las decisiones sobre el cambio de estado deberán hacerse cuando hayan transcurrido los ciclos correspondientes.

Salidas:

El autómata controla el registro de desplazamiento y el registro de validación. Esto lo hace mediante 3 señales: **DATO** (dato a cargar en caso que se trate de “0” o “1”), **CAPTUR** (que activa el ENABLE del registro de desplazamiento permitiéndole cargar ese dato) y **VALID** (que activa el ENABLE del segundo de validación capturando la trama completa que se encuentra en la salida del registro de desplazamiento).

Tenga en cuenta que el hecho de que el estado MUESTREO dure 39 ciclos, es debido a que la transición hacia DATO 0, DATO 1 o DATO SYNC necesita un flanco de reloj más otro de vuelta, en total los 40 ciclos que debe durar el muestreo.

El esquema de este módulo y parte de su descripción en VHDL se muestran a continuación:

**FICHERO: automata.vhd**

```

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

entity automata is
    Port ( CLK      : in  STD_LOGIC;      -- Reloj del autómata
          C0       : in  STD_LOGIC;      -- Condición de decision para "0"
          C1       : in  STD_LOGIC;      -- Condición de decision para "1"
          DATO      : out STD_LOGIC;      -- Datos a cargar
          CAPTUR    : out STD_LOGIC;      -- Enable del reg. de desplaz.
          VALID     : out STD_LOGIC);     -- Enable del reg de validación
end automata;

architecture a_automata of automata is
    type TIPO_ESTADO is (ESP_SYNC, AVAN_ZM, MUESTREO, DATO0, DATO1, DATOSYNC);
    signal ST : TIPO_ESTADO := ESP_SYNC ; -- Estado inicial en que arranca
    signal salidas : : STD_LOGIC_VECTOR (2 downto 0) := "000";

begin
    process (CLK)
        variable cont : STD_LOGIC_VECTOR (7 downto 0) := "00000000"; -- contador
                                -- para contar ciclos en un estado, iniciado a 0
        begin
            if (CLK'event and CLK = '1') then
                case ST is
                    when ESP_SYNC =>      -- Estado normal, dura 1 ciclo de reloj

                        if {CONDICIONES DE LAS ENTRADAS} then
                            ST<={NUEVO ESTADO};
                            . . . . {otras condiciones}

                    when AVAN_ZM =>      -- Estado que dura 20 ciclos de reloj

                        cont:= cont+1;      -- Se incrementa el contador.
                        if (cont=20) then  -- Si llega a 20
                            cont:=(others=>'0'); -- Poner el contador a 0
                            ST<=MUESTREO;    -- Y cambiar de estado
                        else
                            ST<=AVAN_ZM;    -- Si no ha llegado a 20 permanecer
                        end if;              -- en el mismo estado

                        . . . . {Otros estados}
                end case;
            end if;
        end process;
    end process;

```

```

with ST select
    salidas<=
        "000" when ESP_SYNC,
        ...   when AVAN_ZM,
        ...   when ...,
        ...
        "000" when others;

DATO <= salidas(2);
CAPTUR <= salidas(1);
VALID <= salidas(0);

end a_automata;

```

En este caso, al tratarse de un autómata de MOORE las salidas dependen solamente del estado. Por tanto, se puede utilizar un bloque combinacional (with...select), en paralelo con el proceso de cambio de estado, para asignar los valores de las salidas en función del estado.

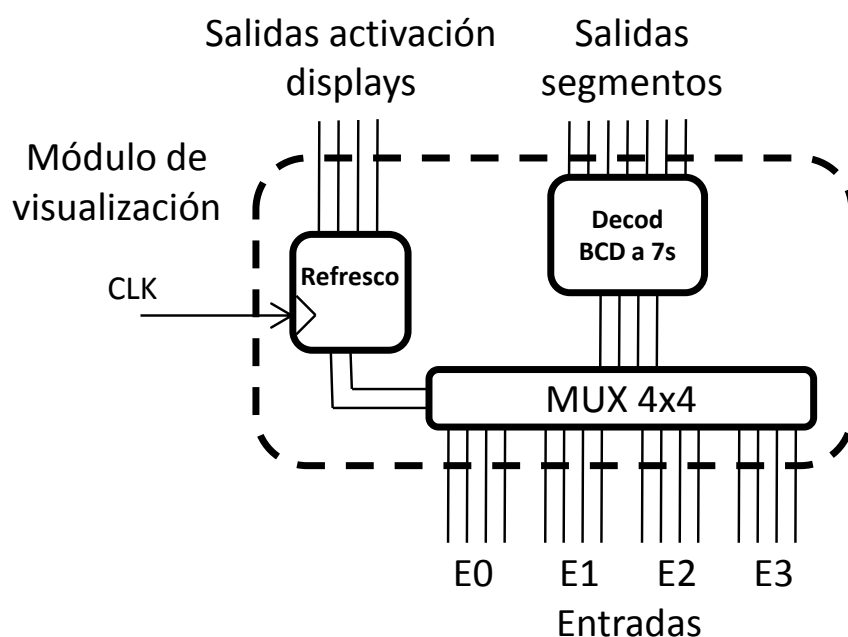
Complete por tanto la descripción funcional de este módulo en VHDL. Tenga en cuenta que en el estado MUESTREO, la condición sobre las entradas debe comprobarse al final de los 39 ciclos (momento en el que todas las muestras se encuentran en la ventana).

4.2.11 Módulo de visualización:

El módulo de visualización está detalladamente explicado en el ejemplo 4 del “Manual de referencia de la tarjeta BASYS2” [7]. Está formado a su vez por tres subsistemas: un decodificador BCD a 7 segmentos, un multiplexor de 4 entradas de 4 bits y un bloque de control que realiza el refresco periódico de los displays.

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.

Aproveche el código que se muestra en los ejemplos para desarrollar un módulo con una descripción arquitectural que responda a la siguiente definición en VHDL:



FICHERO: visualizacion.vhd

```

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

entity visualizacion is
    Port ( E0 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada MUX 0
          E1 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada MUX 1
          E2 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada MUX 2
          E3 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada MUX 3
          CLK : in  STD_LOGIC; -- Entrada de reloj
          SEG7 : out STD_LOGIC_VECTOR (6 downto 0); -- Salida para los displays
          AN : out  STD_LOGIC_VECTOR (3 downto 0)); -- Activación individual
end visualizacion;

architecture a_visualizacion of visualizacion is

    . . . . {Posibles señales necesarias (NODOS)}

    component MUX4x4
        Port ( E0 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada 0
              E1 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada 1
              E2 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada 2
              E3 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada 3
              S : in  STD_LOGIC_VECTOR (1 downto 0); -- Señal de control
              Y : out  STD_LOGIC_VECTOR (3 downto 0)); -- Salida
    end component;

    component decod7s
        Port ( D : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada BCD
              S : out  STD_LOGIC_VECTOR (6 downto 0)); -- Salida para excitar los displays
    end component;

    component refresco
        Port ( CLK : 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 individuales
    end component;

begin

    U1 : {Interconexiones}

end a_visualizacion;

```

Complete la descripción arquitectural (conexiones) de este módulo.

4.2.12 Cableado del circuito completo y fichero de asociaciones:

Finalmente, tendrá que crear el fichero “principal.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:

FICHERO: principal.vhd

```

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

entity principal is
    Port ( CLK : in  STD_LOGIC; -- entrada de reloj
          SIN : in  STD_LOGIC; -- entrada de datos
          AN  : out STD_LOGIC_VECTOR (3 downto 0); -- control de displays
          SEG7 : out STD_LOGIC_VECTOR (6 downto 0)); -- segmentos de displays
end principal;

architecture a_principal of principal is

    -- Constantes del circuito (umbrales de decisión)

    constant UMBRAL1 : STD_LOGIC_VECTOR (5 downto 0) := "100010"; -- 34
    constant UMBRAL2 : STD_LOGIC_VECTOR (5 downto 0) := "100110"; -- 38

    . . . . {Posibles señales necesarias (NODOS)}

    component gen_reloj
        Port ( CLK : in  STD_LOGIC;      -- Reloj de la FPGA
              CLK_OUT : out STD_LOGIC); -- Reloj de frecuencia dividida
    end component;

    component reg_desp40
        Port ( SIN : in  STD_LOGIC; -- Datos de entrada serie
              CLK : in  STD_LOGIC; -- Reloj de muestreo
              Q : out  STD_LOGIC_VECTOR (39 downto 0)); -- Salida paralelo
    end component;

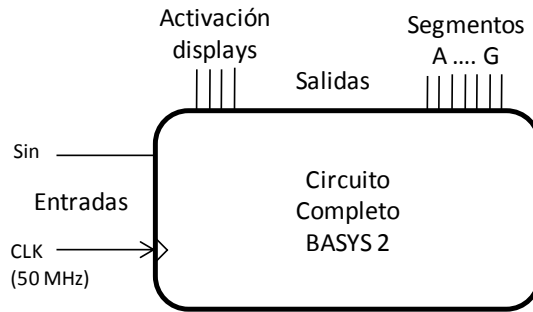
    component sumador40
        Port ( ENT : in  STD_LOGIC_VECTOR (39 downto 0);
              SAL : out  STD_LOGIC_VECTOR (5 downto 0));
    end component;

    component comparador
        Port ( P : in  STD_LOGIC_VECTOR (5 downto 0);
              Q : in  STD_LOGIC_VECTOR (5 downto 0);
              PGTQ : out STD_LOGIC;
              PLEQ : out STD_LOGIC);
    end component;

    component AND_2
        Port ( A : in  STD_LOGIC;
              B : in  STD_LOGIC;
              S : out  STD_LOGIC);
    end component;

    component reg_desp
        Port ( SIN : in  STD_LOGIC; -- Datos de entrada serie
              CLK : in  STD_LOGIC; -- Reloj
              EN  : in  STD_LOGIC; -- Enable
              Q : out  STD_LOGIC_VECTOR (13 downto 0)); -- Salida paralelo
    end component;

```



```

component registro
  Port ( ENTRADA : in  STD_LOGIC_VECTOR (13 downto 0);
        SALIDA  : out STD_LOGIC_VECTOR (13 downto 0);
        EN      : in  STD_LOGIC;                -- Enable
        RCLK    : in  STD_LOGIC);
end component;

component automata
  Port ( CLK      : in  STD_LOGIC;    -- Reloj del autómata
        C0       : in  STD_LOGIC;    -- Condición de decision para "0"
        C1       : in  STD_LOGIC;    -- Condición de decision para "1"
        DATO      : out STD_LOGIC;    -- Datos a cargar
        CAPTUR    : out STD_LOGIC;    -- Enable del reg. de desplaz.
        VALID     : out STD_LOGIC);  -- Activación registro
end component;

component visualizacion
  Port ( E0 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada MUX 0
        E1 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada MUX 1
        E2 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada MUX 2
        E3 : in  STD_LOGIC_VECTOR (3 downto 0); -- Entrada MUX 3
        CLK : in  STD_LOGIC;                -- Entrada de reloj FPGA
        SEG7 : out STD_LOGIC_VECTOR (6 downto 0); -- Salida para los displays
        AN  : out STD_LOGIC_VECTOR (3 downto 0)); -- Activación individual
end component;

begin

U1 : {Interconexiones}

end a_circuito;

```

Fichero de asociaciones

```

# 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 "SIN" LOC = "--"; #Entrada terminal 1

```

Complete la descripción arquitectural (conexiones) de este módulo y complete también el fichero de asociaciones.

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.
- Como regla general **se prohíbe el uso de potenciómetros**. Solamente podrán emplearse dos: uno para ajustar la ganancia variable en el acondicionador de señal (apartado 4.1.1) y otro en el comparador para seleccionar el umbral (apartado 4.1.6). Todas las resistencias deben ser fijas y deberá justificarse adecuadamente su valor.
- 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:

La práctica está pensada para ser montada 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 45, 47 y 49 de este enunciado.

Más adelante se recomienda un calendario para el desarrollo de la práctica 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 25 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 $\Phi-360^\circ$.

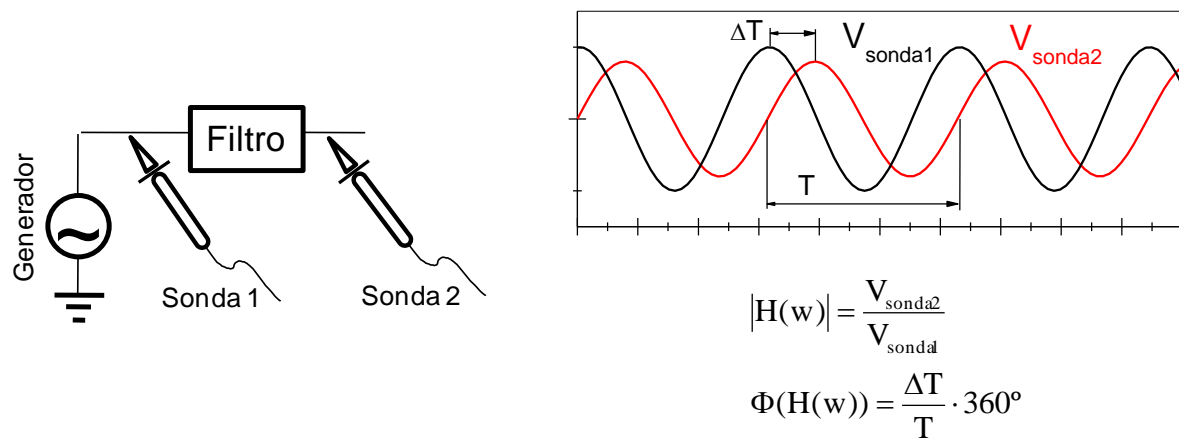


Figura 25 Esquema del circuito de medida de los filtros. Se muestra un ejemplo real de cómo calcular el módulo y la fase de la función de transferencia.

4.3.3 Realización de diagramas de Bode:

Se pide también la realización de los diagramas 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:

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

Filtro paso banda:

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

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 46 y 48) se proporcionan plantillas de estas gráficas.

4.3.4 Utilización del osciloscopio:

En esta práctica será imprescindible el uso del osciloscopio. En la plataforma de la asignatura está disponible el manual de uso del osciloscopio, así como una guía rápida.

5. ESPECIFICACIONES DEL SISTEMA

El sistema que se diseñe deberá cumplir las siguientes especificaciones:

Emisor:

1. Alimentación simétrica con +5 y -5 V procedente de la fuente de alimentación del laboratorio.
2. Introducción de la señal de datos mediante un reproductor MP3.
3. Parte analógica realizada con amplificadores operacionales del tipo TL082
4. Parte digital descrita en VHDL y sintetizada sobre FPGA Spartan 3E.
5. Frecuencia de muestreo de la señal digital: 40 Hz.
6. Tenga en cuenta los niveles de señal en distintos puntos que se exigen a lo largo del enunciado.
7. Solamente se permite la utilización de dos cables entre la parte analógica y la digital: el cable de datos y el de masa.
8. Será suficiente con entregar la práctica montada 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 – 11 al 15 de Septiembre

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 y trate de compilar y sintetizar 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)
- 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 – 18 al 22 de Septiembre

Etapas de acondicionamiento de señal: amplificador (ap. 4.1.1.1)

- Compruebe el funcionamiento correcto de la etapa. Para ello mida la salida del generador de señal y ajuste los controles para observar una señal senoidal de 1 KHz y 100 mVpp de amplitud.
- A continuación quite la resistencia de 10 Ω y conecte el generador de señal a la entrada de la etapa.
- Mida la salida de la etapa y asegúrese que la ganancia se ajusta con los cálculos.
- Varíe la frecuencia del generador y asegúrese que la frecuencia de corte inferior del filtro paso alto está de acuerdo con su diseño
- Por último, vuelva a poner la resistencia de 10 Ω y conecte el dispositivo MP3.
- Ajuste el potenciómetro para obtener 1 Vpp de amplitud a la salida.

6.3 Semana 3 – 25 al 29 de Septiembre (lunes 25 es Festivo)

Etapas de acondicionamiento de señal: filtro paso banda (ap. 4.1.1.2) y rectificador (ap. 4.1.4)

- 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 (amplificador) y posterior (rectificador), introduzca la señal a través del dispositivo MP3 y compruebe que el rectificador funciona adecuadamente.

6.4 Semana 4 – 2 al 6 de Octubre

Filtro paso bajo y comparador (ap. 4.1.5 y 4.1.6)

- 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 banda (apartado 4.3.3). Obtendrá un polo real doble. Dibuje el diagrama de Bode correspondiente sobre la curva de comportamiento del filtro.
- Conecte el filtro a la etapa anterior (rectificador), introduzca la señal a través del dispositivo MP3 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 5V con flancos definidos.

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

Durante las sesiones 2, 3 y 4 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.

6.5 Semana 5 – 9 al 13 de Octubre (el jueves es Festivo y el viernes No Lectivo)

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.

ENTREGA 1: Esta semana deberá entregar las medidas realizadas sobre el filtro paso banda según el formato que figura en la página 45 de este documento.

6.6 Semana 6 – 16 al 20 de Octubre

Generador de reloj y módulo de visualización (ap. 4.2.3 y 4.2.11)

- Asocie (mediante un fichero de asociaciones) la única salida del generador de reloj hacia un terminal externo de la FPGA y mida con el osciloscopio la frecuencia de la señal obtenida..
- Implemente el módulo de visualización (recuerde que puede encontrar los submódulos que lo integran en los ejemplos [7]). Escriba un fichero de asociaciones que conecte los microinterruptores de la tarjeta BASYS 2 a dos de las entradas de 4 bits y compruebe que esos dos displays visualizan correctamente las cifras..
- Cambie ahora los microinterruptores a las otras dos entradas y compruebe que los displays asociados visualizan correctamente las cifras.

6.7 Semana 7 – 23 al 27 de Octubre

Registro de desplazamiento y sumador (ap. 4.2.4 y 4.2.5)

- Utilice los testbench que se suministran en la página web de la asignatura para simular y comprobar el funcionamiento de estos elementos.
- El fichero “**test_reg_desp40.vhd**” permite comprobar el funcionamiento del registro de desplazamiento de 40 bits. Este código introduce una secuencia de 8 “0” y 32 “1” en sincronía con el reloj. Deberá observar cómo se desplaza esta señal a través del registro en los 40 terminales de salida y decidir si funciona correctamente.
- El fichero “**test_sumador40.vhd**” permite realizar una comprobación parcial del funcionamiento del sumador. Dado que la prueba exhaustiva sería imposible debido al gran número de entradas, este código introduce un valor de 40 “0”, a continuación 20 “0” y 20 “1”, posteriormente 20 “1” y 20 “0”, luego una secuencia de 40 bits “1010... 1010”, después una secuencia “0101... 0101”, y por último 40 “1”. Los valores (en decimal) de las salidas en cada caso deberían ser 0,20,20,20,20 y 40 respectivamente. Si esto funciona bien es bastante probable que el código del sumador esté copiado correctamente. Puede cambiar las entradas para comprobar otros valores de suma.

6.8 Semana 8 – 30 de Octubre al 3 de Noviembre (el miércoles es Festivo)

Comparador, puerta AND y registro de desplazamiento (ap. 4.2.6, 4.2.7 y 4.2.8)

- El fichero “**test_comparador.vhd**” permite realizar una comprobación parcial del funcionamiento del comparador. Se introducen todos los valores posibles en las entradas P (64 valores) y se fija un valor para la entrada Q. Observe los valores de las salidas $P \leq Q$ y $P > Q$ para ver que se activan correctamente. Haga pruebas con diferentes valores de Q.
- Para comprobar el funcionamiento de la puerta AND puede crear un fichero de simulación. Es un circuito muy sencillo
- El fichero “**test_reg_desp.vhd**” permite comprobar el funcionamiento del registro de desplazamiento de 14 bits. Este código introduce una secuencia de 3 “0” y 11 “1” en sincronía con el reloj. Primero lo hace con el ENABLE desactivado y a continuación repite la secuencia con el ENABLE activado. Deberá observar cómo se desplaza esta señal a través del registro en los 14 terminales de salida cuando el ENABLE está activo y decidir si funciona correctamente

ENTREGA 2: Esta semana deberá entregar las medidas realizadas sobre el filtro paso bajo según el formato que figura en la página 47 de este documento.

6.9 Semana 9 – 6 al 10 de Noviembre (el martes es clase de jueves, el jueves es Festivo)

Registro y autómata (ap. 4.2.9 y 4.2.10)

- El fichero “**test_registro.vhd**” permite comprobar el funcionamiento del registro. Este fichero introduce diferentes entradas en varios instantes de tiempo. Deberá comprobar que el dispositivo sólo captura los datos (los transfiere a la salida) en los flancos positivos de la señal de reloj.
- En referencia al autómata, deberá codificar en VHDL su funcionamiento. Es el elemento de mayor dificultad de simulación. Se suministra un fichero: “**test_automata.vhd**” que hace que el autómata comience en el estado ESP_SYNC y vaya pasando por varios estados. Cuente los ciclos de reloj en cada estado, así como las transiciones en función de las entradas y las salidas con referencia a cada estado. Asegúrese que se corresponde con el diagrama de la Figura 24.

6.10 Semana 10 – 13 al 17 de Noviembre

Autómata, cableado del circuito completo y asociaciones (ap. 4.2.10 y 4.2.12)

- Se programan dos semanas para la codificación depuración y síntesis del autómata.
- Deberá además completar la descripción arquitectural del circuito y asociar sus terminales lógicos con las patillas físicas de la FPGA.

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

6.11 Semana 11 – 20 al 24 de Noviembre**Redacción de la memoria final y realización de posibles mejoras**

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

ENTREGA 3: Esta semana deberá entregar las medidas que se indican en la página 49 de este documento.

6.12 Semana 12 – 27 de Noviembre al 1 de Diciembre**Redacción de la memoria final y realización de posibles mejoras**

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

6.13 Semana 13 – 4 al 8 de Diciembre (el lunes es clase de viernes, el miércoles y viernes es Festivo, el jueves es No Lectivo)**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 – 11 al 15 de Diciembre (el martes y viernes son días de Turnos Libres)**Examen escrito (lunes 11)****6.15 Semana 15 - 18 al 22 de Diciembre****Examen oral (toda la semana)**

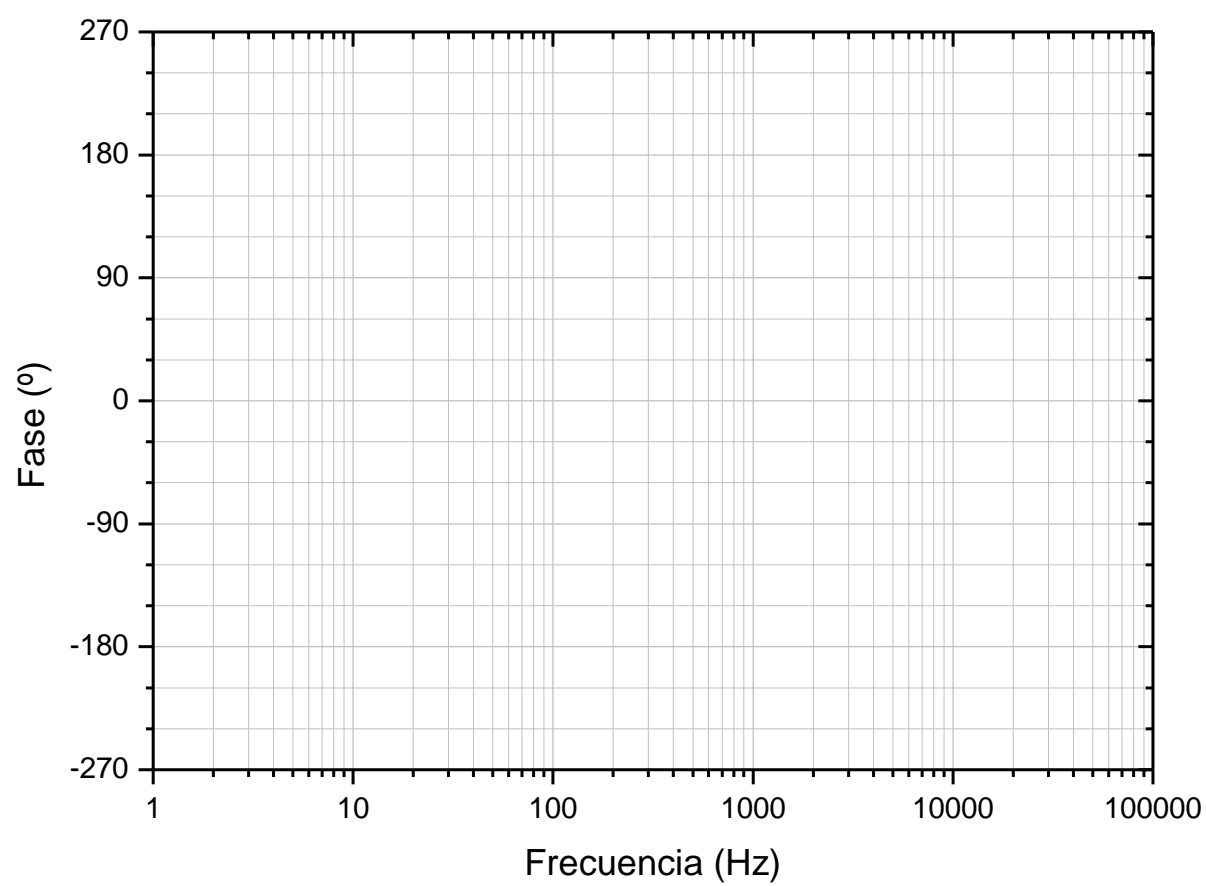
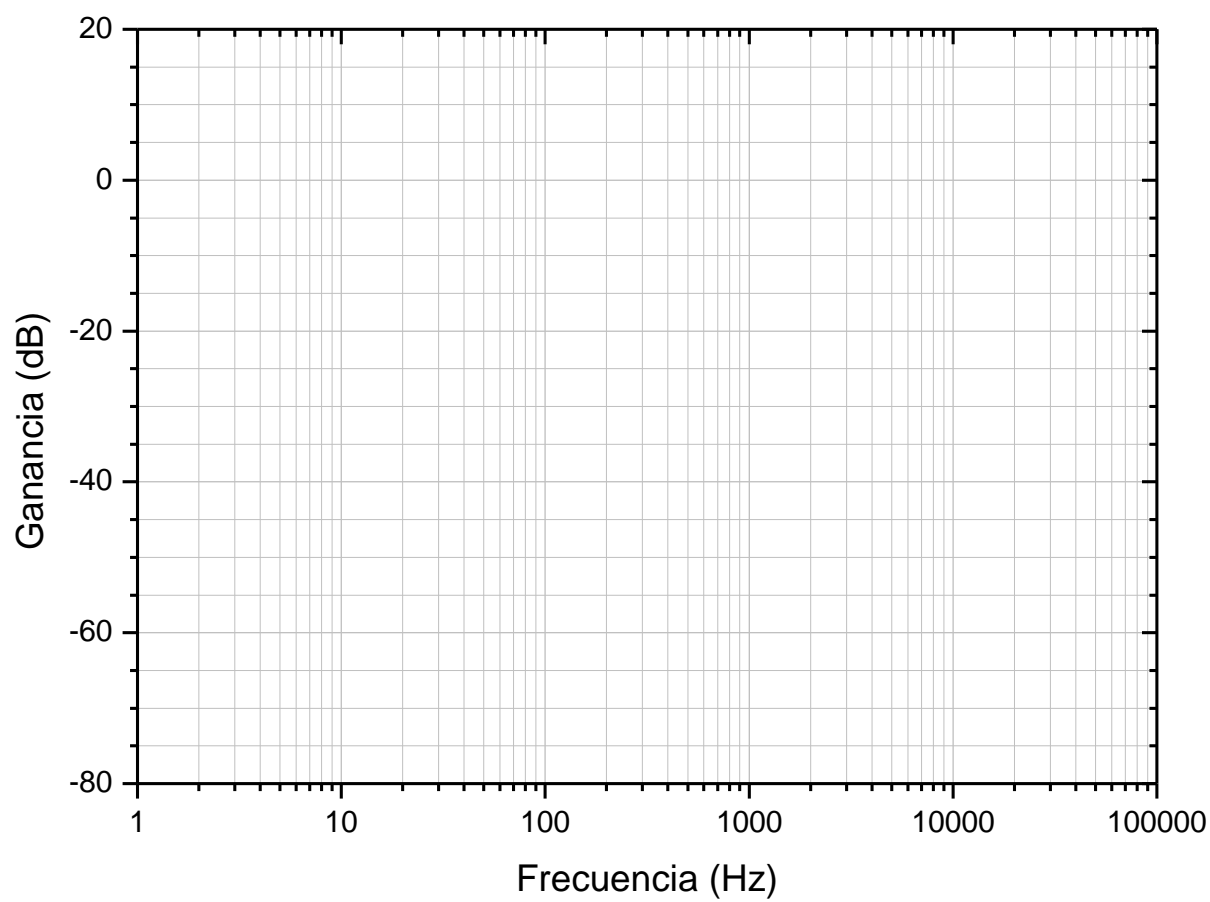
- Se realizarán los exámenes orales.

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

Tabla de valores medidos

[illegible]

Especifique aquí el valor del cero, de los dos polos complejos conjugados y del módulo de estos últimos.

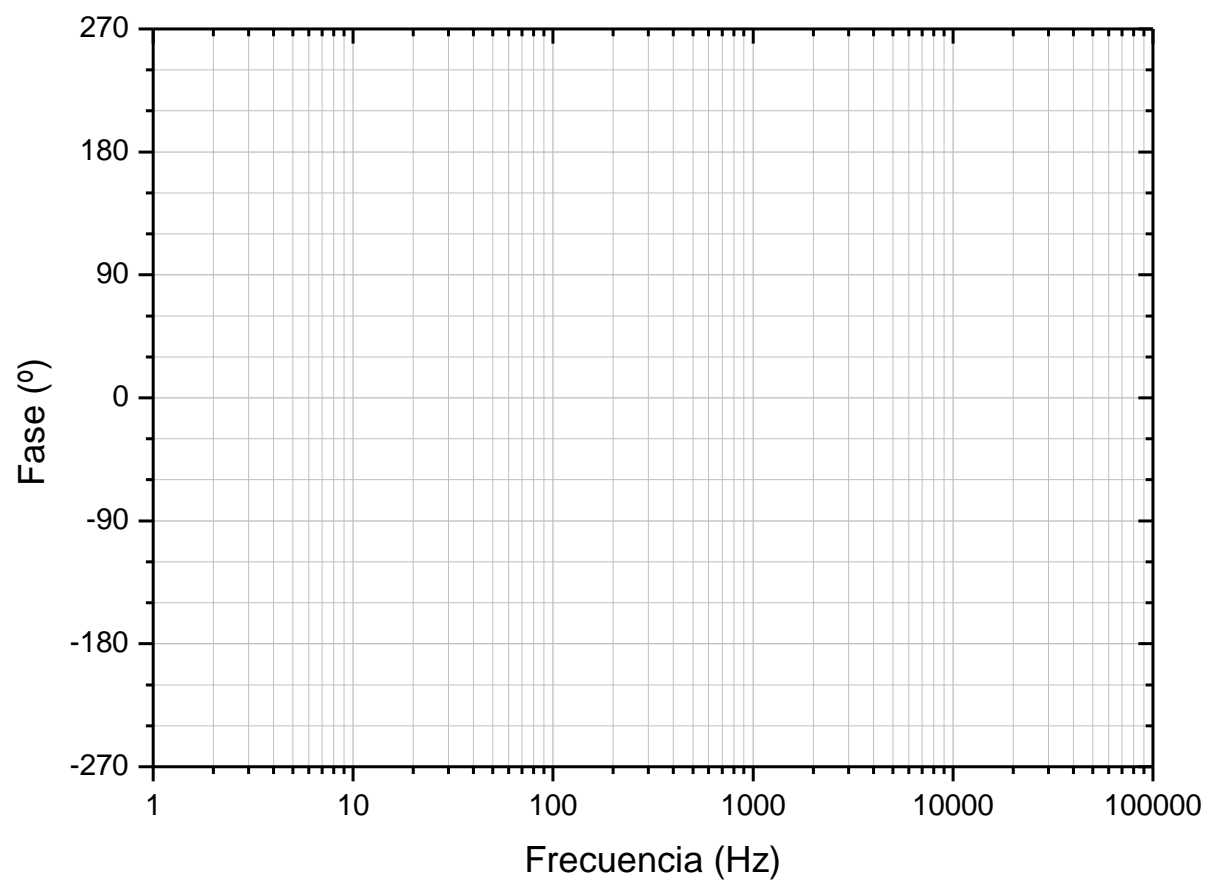
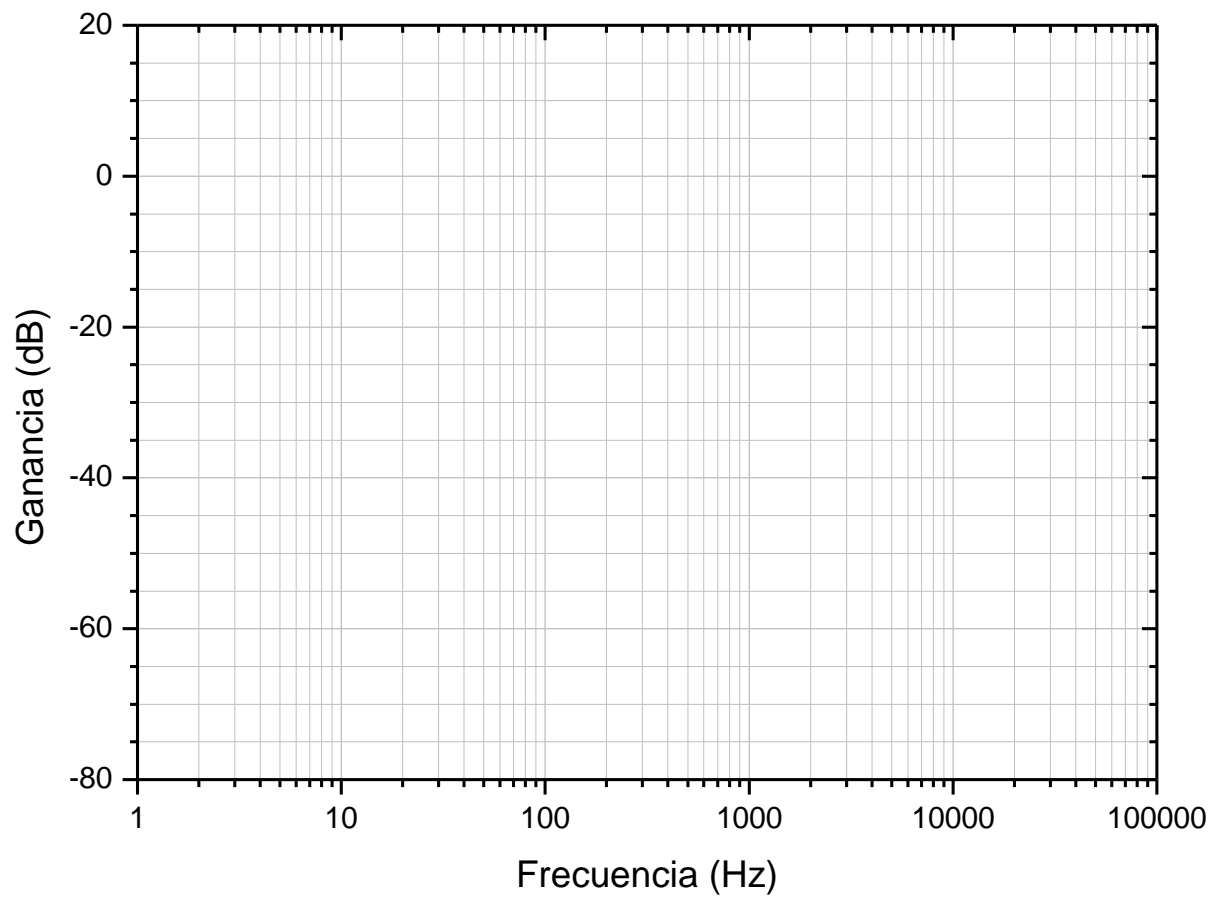
Medidas del comportamiento del filtro

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 de Bode de la función de transferencia junto con la medida del filtro.

Medidas del comportamiento del filtro y diagrama 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:

1. Señal a la salida del dispositivo MP3. Realice una captura donde se puedan apreciar al menos 2 bits de información. (un cero y un uno)
2. Señal a la salida del filtro paso banda. Realice dos capturas, una donde se pueda apreciar al menos 2 bits de información (un cero y un uno) y otra donde se pueda apreciar una trama completa (14 bits de información más el bit de sincronismo).
3. Señal a la salida del rectificador. Realice dos capturas, una donde se pueda apreciar al menos 2 bits de información (un cero y un uno) y otra donde se pueda apreciar una trama completa (14 bits de información más el bit de sincronismo).
4. Señal a la salida del filtro paso bajo. Realice una captura donde se puedan apreciar al menos 2 bits de información (un cero y un uno) y el nivel de referencia del comparador, utilizando los dos canales del osciloscopio.
5. Salida de 40 Hz obtenida mediante división de frecuencia de la señal de reloj de la FPGA (señal CLK_M). Capture la señal de tal manera que pueda apreciarse 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 de la Práctica, 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: celt.die.upm.es.
- 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.

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 tanto en la plataforma Moodle (formato digital) como en el laboratorio (en formato papel).

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 entregas, la memoria, 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 pastillas 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 Presentación de la hora en formato AM / PM (dificultad baja):

Esta mejora consiste en visualizar la hora con el formato AM/PM por medio de uno de los LED presentes en la tarjeta BASYS 2.

Descargue el fichero: **hora_11_57_AM.mp3**. Este fichero contiene 10 tramas de 15 bits seguidos de un SYNC. En las tramas se encuentra codificada en BCD la hora desde las 11:57 AM hasta las 12:06 PM. Cada trama consta de las decenas de la hora (BCD con 3 bits), unidades de la hora (BCD con 4 bits), decenas del minuto (BCD con 3 bits), unidades del minuto (BCD con 4 bits) y un bit adicional que indica AM ("0") o PM ("1"). Total 15 bits. Posteriormente se transmite un SYNC.

Para realizar esta mejora deberá ampliar el registro de desplazamiento para capturar un bit más, presentando su valor en uno de los LED disponibles en la tarjeta BASYS 2.

8.2 Visualización de hora y fecha (dificultad media):

Esta mejora consiste en visualizar la hora y la fecha utilizando los displays. Dado que solamente existen 4 displays se propone la utilización de uno de los pulsadores de la tarjeta BASYS 2 para alternar la visualización entre hora y fecha. Cuando el pulsador está suelto, se visualiza la hora, cuando el pulsador está pulsado se visualiza la fecha.

Descargue el archivo: **horayfecha_23_58_05_06.mp3**. Este archivo contiene 10 tramas de 28 bits seguidos de un SYNC. En las tramas se contiene la hora y la fecha desde las 23:58 del 5/6 hasta las 00:07 del 6/6. Cada trama está compuesta por las decenas de la hora (BCD con 3 bits), unidades de la hora (BCD con 4 bits), decenas del minuto (BCD con 3 bits), unidades del minuto (BCD con 4 bits), las decenas del día (BCD con 3 bits), unidades del día (BCD con 4 bits), decenas del mes (BCD con 3 bits) y unidades del mes (BCD con 4 bits). Total 28 bits. Seguidamente se transmite un SYNC.

Para realizar esta mejora deberá extender la longitud del registro de desplazamiento para capturar todos los bits de la trama. Posteriormente deberá presentar la información de hora o fecha sobre los

displays en función del valor de uno de los pulsadores de la tarjeta BASYS 2. Deberá emplear un multiplexor cuya entrada de control se encuentre conectada al citado pulsador.

8.3 Visualización de hora, fecha y día de la semana (dificultad alta):

Esta mejora consiste en visualizar la hora, la fecha, y el día de la semana utilizando los displays y los LEDs presentes en la tarjeta de desarrollo. Dado que solamente existen 4 displays se propone la utilización de uno de los pulsadores de la tarjeta BASYS 2 para alternar la visualización entre hora y fecha. Cuando el pulsador está suelto, se visualiza la hora, cuando el pulsador está pulsado se visualiza la fecha. El día de la semana se presentará encendiendo el LED correspondiente: LED7 = lunes, LED6 = martes ...LED 2=domingo. Hágalo así por la disposición de los LEDs de izquierda a derecha en la tarjeta.

Descargue el archivo: **horafechaydia_23_58_05_06_martes.mp3**. Este archivo contiene 10 tramas de 31 bits seguidos de un SYNC. En las tramas se contiene la hora, la fecha y el día de la semana desde las 23:58 del 5/6 (martes) hasta las 00:07 del 6/6 (miércoles). Cada trama está compuesta por las decenas de la hora (BCD con 3 bits), unidades de la hora (BCD con 4 bits), decenas del minuto (BCD con 3 bits), unidades del minuto (BCD con 4 bits), las decenas del día (BCD con 3 bits), unidades del día (BCD con 4 bits), decenas del mes (BCD con 3 bits), unidades del mes (BCD con 4 bits) y el día de la semana codificado en binario (000 para el lunes y 110 para el domingo). Total 31 bits. Seguidamente se transmite un SYNC.

Para realizar esta mejora deberá extender la longitud del registro de desplazamiento para capturar todos los bits de la trama. Posteriormente deberá presentar la información de hora o fecha sobre los displays en función del valor de uno de los pulsadores de la tarjeta BASYS 2. Deberá emplear un multiplexor cuya entrada de control se encuentre conectada al citado pulsador. Para visualizar el día deberá decodificar el valor binario del día y encender el LED correspondiente.

8.4 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.

8.5 Simulación con PSPICE (dificultad media)

En este curso se propone una mejora puntuada consistente en la realización mediante el programa PSPICE de diversas simulaciones de la parte analógica del montaje, de forma similar a como se llevaría a cabo durante un diseño realista en un entorno profesional.

Para obtener la puntuación indicada 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 4.1.1.2, 4.1.4 y 4.1.5) utilizando como excitación a la entrada de la cadena la secuencia "10" modulada en amplitud sobre la portadora de 1 KHz (no se recomienda emplear secuencias de más bits por la complejidad del análisis).

- Simulaciones en AC (barrido en frecuencia) del filtro paso banda (apartado 4.1.1.2) y del filtro paso bajo Sallen Key (apartado 4.1.5) por separado. Se piden las gráficas de respuesta en amplitud (dB) y fase ($^{\circ}$).

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.

Los ordenadores del Laboratorio disponen del software necesario para realizar las simulaciones descritas anteriormente.

8.6 Montaje en PCB (dificultad muy alta)

En la Práctica Básica se exige el montaje, como requisito mínimo, en placa de inserción, de modo que se valorará positivamente la construcción de los prototipos en placa de circuito impreso. En este caso, considere la utilización de zócalos para facilitar el cambio de integrados.

Será imprescindible presentar igualmente el prototipo previo (placa de inserción), como también los documentos generados durante el empleo de las herramientas software necesarias para el diseño.

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.