

# Sintetizador De Audio

Juan Alonso Rubiano Portela  
*juarubianopo@unal.edu.co*

Laboratorio de Digital I  
 Departamento de Ingeniería Eléctrica y Electrónica  
 Universidad Nacional de Colombia.  
 Bogotá. Colombia.

**Abstract**—The objective of this report is to show the hardware description of a basic audio synthesizer module, by means of the control of a numerical keyboard, the desired musical octave is selected and when pressing the key corresponding to the musical note, a sound will be obtained, it is also possible to select the shape of the output signal between square, triangular and sinusoidal.

**keywords**— sintetizador, frecuencia, nota musical, señal de reloj, registro

## I. INTRODUCCIÓN

EL siguiente informe mostrara la descripción de hardware de un sintetizador de audio muy básico, el cual consta solamente de las 12 teclas de las notas musicales que componen una octava, un teclado que permite seleccionar entre la tercera y la sexta octava, también si se desea una onda de salida triangular, senoidal o la onda cuadrada que se obtiene de la FPGA, para la onda triangular y senoidal se usaron dos filtros pasa bajos los cuales permiten la transformación de la onda.

### I-A. Problemática

Hoy en día ya no es difícil acceder a un instrumento musical para practicar, sin embargo en los colegios del distrito o las diferentes gobernaciones del país no es posible acceder a clases de piano por los altos costos de este instrumento, esto genera poca motivación en la población haciendo que el tiempo se utilice en otras actividades que no fomentan el desarrollo intelectual y social de un individuo en formación.

### I-B. Solución

La solución que se pretende plantear es desarrollar un modulo de sintetizador de audio con teclado para así poder disminuir los costos de producción de este instrumento y poder promover su aprendizaje en las escuelas del país, de esta forma con la practica de un instrumento musical los niños quienes son el futuro del país mejoraran en sus habilidades cognitivas, sociales y académicas, permitiendo asegurar un mejor futuro para sus vidas.

### I-C. Objetivos

**I-C1. General:** Desarrollar un modulo de sintetizador de audio en lenguaje verilog con el fin de fomentar el gusto por la música y el aprendizaje del instrumento.

### I-C2. Específicos:

- Mitigar costos de producción de un instrumento de este calibre, ya que en la actualidad es muy difícil acceder a él, con esto no se pretende competir con las compañías desarrolladoras de los instrumentos musicales, el objetivo es puramente académico.
- Se pretende servir de escalón para las personas que han buscado en la red sin encontrar información de utilidad acerca del tema o códigos complicados difíciles de entender, dejar el código abierto a cualquier persona aficionada a la electrónica que desee seguir con el desarrollo y mejoramiento de este mismo.

### I-D. Marco teórico

**I-D1. El sintetizador:** Uno de los grandes descubrimientos de la era digital fue el sintetizador, al menos para la industria de la música, desde su invención la música no volvería a ser la misma, aunque desde el siglo XIX ya se venían desarrollando varios avances para la industria musical no fue sino hasta 1978 con el Propet – 5 que se abrió la puerta a este tipo de instrumentos, el Propet – 5 era un microcontrolador controlado por teclado. Este dispositivo permite que al generar ondas eléctricas que vibran en diferentes frecuencias se generen ondas de sonido, al combinar estas frecuencias de forma correcta se puede reproducir la escala musical, a medida que fue avanzando la tecnología se le fueron dando mas funciones a estos instrumentos, por ejemplo reproducir el sonido de diferentes instrumentos musicales, cambiar el tipo de onda a senoidal, cuadrada o triangular, entre otras funciones.



Figura 1: Sintetizador Early Minimoog de R.A.Moog (1970)

**I-D2. La frecuencia de las notas musicales:** Como se mencionó anteriormente es posible reproducir cualquier nota

musical si se conoce su frecuencia, la siguiente tabla se utilizó para crear el divisor de frecuencia con la frecuencia de las notas musicales, estas notas están distribuidas en grupos de doce llamadas octavas y cada una representa las notas que se conocen de la escala musical, la primera octava esta formada por la frecuencia fundamental y las siguientes son múltiplos pares de la primera.

		Frecuencia (en Hertzios) de las notas musicales											
		x											
		0	1	2	3	4	5	6	7	8			
n=1	do		32.7	65.41	130.81	261.63	523.25	1046.50	2093.00	4186.01			
n=2	do#		34.65	69.30	138.59	277.18	554.37	1108.73	2217.46	4434.92			
n=3	re		36.71	73.42	146.83	293.66	587.33	1174.66	2349.32	4698.64			
n=4	re#		38.89	77.78	155.56	311.13	622.25	1244.51	2489.02	4978.03			
n=5	mi		41.2	82.41	164.81	329.63	659.26	1318.51	2637.02	5274.04			
n=6	fa	21.826	43.65	87.31	174.61	349.23	698.46	1396.91	2793.83	5587.65			
n=7	fa#	23.125	46.25	92.50	185.00	369.99	739.99	1479.98	2959.96	5919.91			
n=8	sol	24.50	49.00	98.00	196.00	392.00	783.99	1567.98	3135.96	6271.93			
n=9	sol#	25.96	51.91	103.83	207.65	415.30	830.61	1661.22	3322.44				
n=10	la	27.50	55.00	110.00	220.00	440.00	880.00	1760.00	3520.00				
n=11	la#	29.14	58.27	116.54	233.08	466.00	932.33	1864.66	3729.31				
n=12	si	30.87	61.74	123.47	246.94	493.88	987.77	1975.53	3951.07				

Figura 2: Tabla de frecuencias de las notas musicales

*I-D3. El teclado:* El teclado matricial de dieciséis botones permite dar a un proyecto un toque mas sofisticado ya que tiene una estética bastante agradable, funciona con ocho pines digitales cuatro que corresponden al barrido de las columnas y otros cuatro que corresponden a las filas, es necesario conectar ya sea las filas o las columnas a una resistencia de descarga para evitar alguna corriente parásita que genere algún inconveniente en el proyecto, se conecta de la siguiente forma:

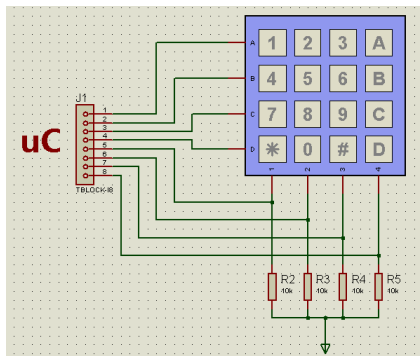


Figura 3: Conexión del teclado a la FPGA

Para el funcionamiento del teclado fue necesario desarrollar varios módulos, el barrido que permite que las cuatro columnas funcionen dinámicamente intercálándose en una frecuencia de 1Khz, el divisor de frecuencia que genera la señal que controla al barrido y el comparador que tiene la función de comparar cuando se haga presión de un botón del teclado las filas y las columnas y asigna un valor al botón seleccionado, a continuación el esquema RTL del modulo.

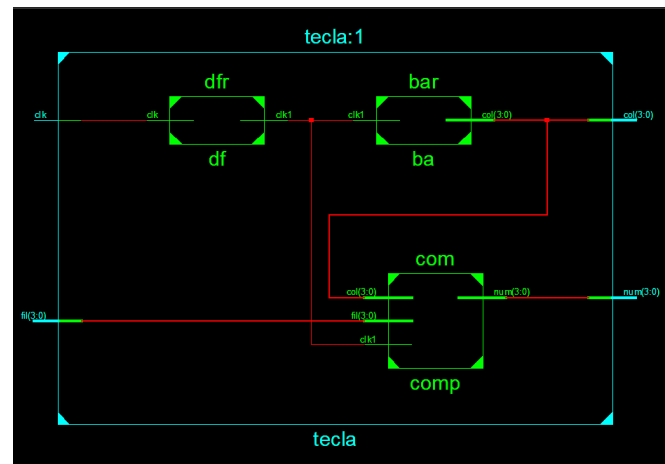


Figura 4: Esquema RTL del modulo teclado

*I-D4. Filtro Pasa bajos:* Un filtro pasa bajo en términos generales es un filtro que permite seleccionar frecuencias mas a bajo de una frecuencia de corte de una señal, hay de diferentes tipos pero en este caso se utilizara un filtro pasa bajos RC, el cual esta conformado por una resistencia y un condensador conectados de la siguiente forma:

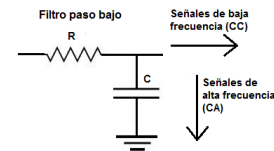


Figura 5: Esquema del filtro Pasa Bajo

Para el calculo del filtro se utilizo la siguiente formula  $f = \frac{1}{2\pi RC}$ , siendo f la frecuencia de corte, en este caso esta ecuación se acomodo fijando los valores de la frecuencia y la capacitancia para poder seleccionar que resistencia poder utilizar.

$$f = \frac{1}{2\pi RC}$$

Para implementarla se descubrió que las notas musicales funcionan a diferentes frecuencias se fijaron los valores de la tercera octava musical para crear el filtro, es decir pasan solo las frecuencias mas bajas de 496Hz y así lograr la onda triangular, se conectó otra etapa del filtro para lograr la onda senoidal, a continuación la ecuación con los valores utilizados y el circuito diseñado en la plataforma multisim online.

$$R = \frac{1}{2\pi 10\mu F 496Hz}$$

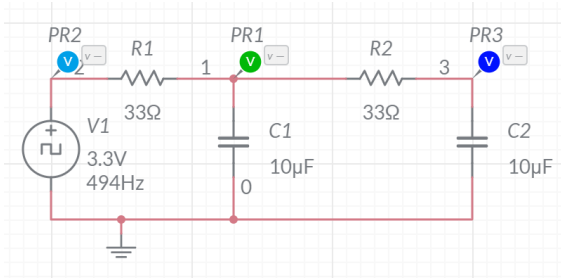


Figura 6: Diseño del circuito filtro Pasa Bajo

## II. IMPLEMENTACIÓN

Para este proyecto se utilizó una FPGA Nexys 2 y se describirá lenguaje verilog, en la primera parte del código se describe la creación del modulo y se crean las variables de entrada y de salida, también los registros necesarios para la implementación y las conexiones necesarias para el funcionamiento, a continuación el código en cuestión:

```
module tecla(clk, fil, col, num);
    input clk;
    input [3:0]fil;
    output [3:0]col;
    output [3:0]num;
    input [11:0]bottom;
    output reg led;
    reg [27:0]clk0;
    reg [27:0]clk_1;
    reg [27:0]clk_2;
    reg [27:0]clk3;
    reg [27:0]clk4;
    reg [27:0]clk5;
    reg [27:0]clk6;
    reg [27:0]clk7;
    reg [27:0]clk8;
    reg [27:0]clk9;
    reg [27:0]clk10;
    reg [27:0]clk11;

    wire [3:0]num;
    wire clk1;
```

Figura 7: creación del modulo e implementación de las variables

La segunda parte del código describe los módulos que se utilizaron para el funcionamiento del teclado dfr que es el divisor de frecuencia, bar que es el barrido de columnas y com que es el comparador de filas y columnas:

```
dfr df(.clk(clk), .clk1(clk1));
bar ba(.clk1(clk1), .col(col));
com comp(.clk1(clk1), .fil(fil), .col(col), .num(num));
```

Figura 8: descripción de los módulos que componen el teclado

La tercera parte del código describe como funciona el sintetizador, aunque el código resultante es bastante largo se implementó de la siguiente forma, varias instrucciones if anidadas que indican que cuan se oprima un numero en el teclado numérico se activan los botones que según el numero oprimido permitirá una frecuencia determinada, el código en su totalidad será compartido en GitHub para su futura revisión

si se desea, la siguiente imagen muestra el código que fue utilizado en su mayoría para los botones utilizados y el resto de octavas:

```
always @(posedge clk)begin
    //Tercera Octava
    if(num == 4'b0001)begin
        if(!bottom[0])begin
            clk0 = 1;
        end
        else begin
            if(clk0 < 'd382263)begin
                clk0 = clk0 + 1;
            end
            else begin
                led = !led;
                clk0 = 0;
            end
        end

        if(!bottom[1])begin
            clk_1 = 1;
        end
        else begin
            if(clk_1 < 'd360750)begin
                clk_1 = clk_1 + 1;
            end
            else begin
                led = !led;
                clk_1 = 0;
            end
        end

        if(!bottom[2])begin
            clk2 = 1;
        end
        else begin
            if(clk2 < 'd331477)begin
                clk2 = clk2 + 1;
            end
            else begin
                led = !led;
                clk2 = 0;
            end
        end
    end
```

Figura 9: descripción de la implementación del sintetizador

## III. FUNCIONAMIENTO

Después de sintetizar el código correctamente, generar la implementación del diseño y crear el archivo .bit, se cargó el código a la FPGA por medio del software de Adept.

Tanto la salida led como la entrada bottom fueron conectadas a los puertos periféricos (pmod) de la FPGA.

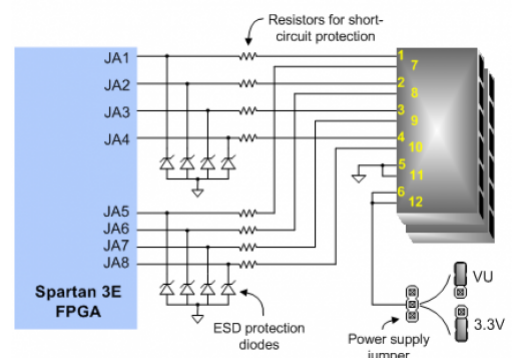


Figura 10: descripción de los puertos Pmod de la FPGA

para que funcionaran los botones fue necesario crear el circuito descrito a continuación en una protoboard, la imagen muestra como funcionan los botones de la Nexys 2, sin embargo se realizó una conexión similar en la protoboard.

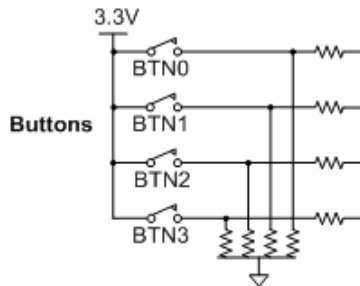


Figura 11: descripción del circuito para conectar los botones

Adjunto a este informe se anexará un vídeo que evidencia el funcionamiento de esta implementación preliminar del proyecto, a continuación una imagen que muestra la disposición de los periféricos conectados, las conexiones y los botones puestos en las protoboard.

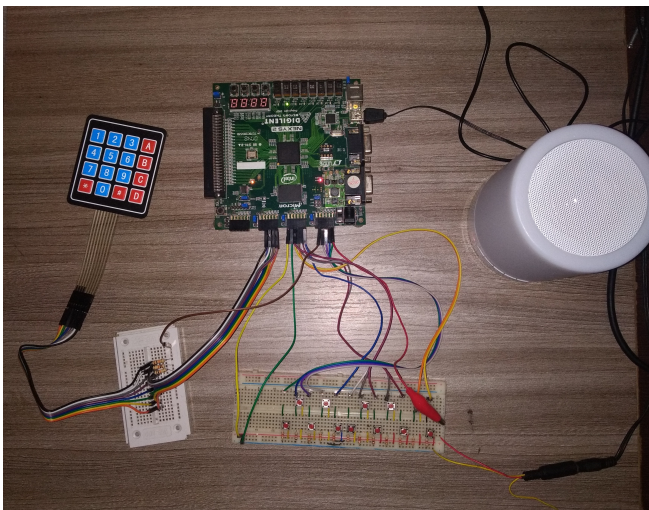


Figura 12: disposición de los periféricos

A continuación se presentan las imágenes obtenidas en el osciloscopio digital de las formas de onda de las tres diferentes salidas:

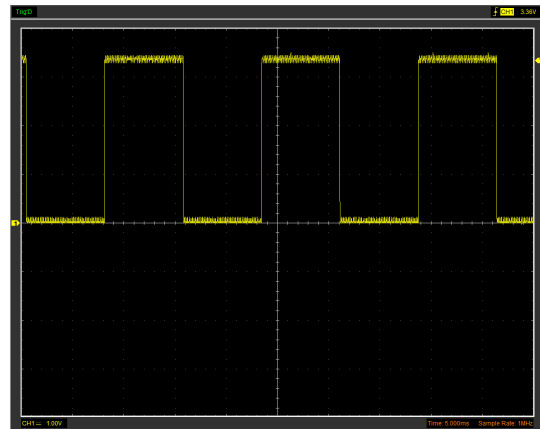


Figura 13: Onda cuadrada obtenida de la FPGA

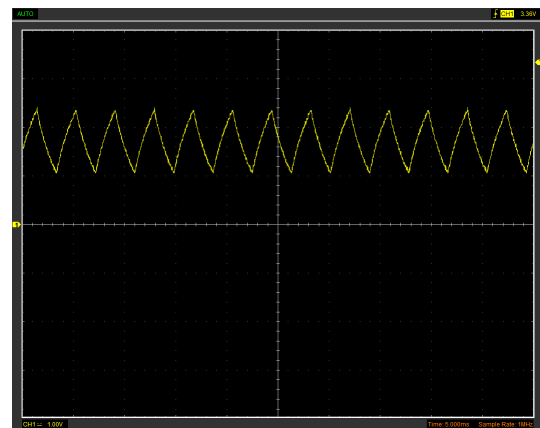


Figura 14: Onda obtenida del primer filtro

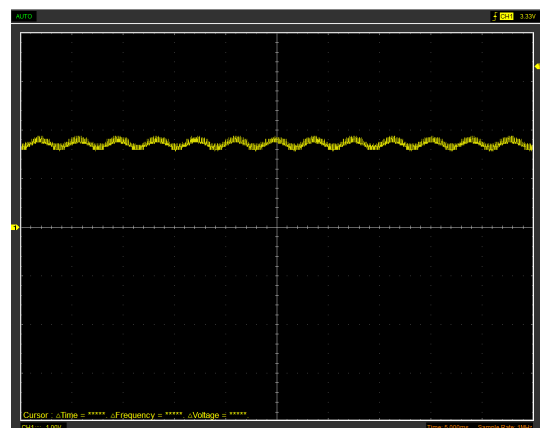


Figura 15: Onda obtenida de las dos etapas de filtro

#### IV. CONCLUSIONES

- es posible usar un divisor de frecuencia para generar sonido en un parlante si se configura la frecuencia adecuada.
- Para hacer funcionar un botón como periférico externo es necesario alimentarlo con la tensión entregada por la FPGA ya que de no hacerlo la FPGA lo tomara como un cero lógico.

- Al conectar un parlante a la FPGA es necesario que se conecte un amplificador externo o que el parlante cuente con uno ya que la Nexys no entrega la potencia suficiente para tener un sonido satisfactorio.
- es necesario conectar resistencias de descarga para el teclado numérico ya que así se evita la circulación de corrientes parásitas que puedan afectar la integridad del proyecto y de la FPGA.
- Con la instrucción if anidada es posible hacer que al oprimir un numero en el teclado numérico se activen diferentes funciones, por ejemplo activar un led con un botón o activar un sonido de una frecuencia específica.
- A medida que aumentan las etapas de filtro disminuye la amplitud de la señal de salida, es necesario conectar un circuito amplificador para corregir esto.

#### REFERENCIAS

- [1] John F.Wakerly (2001). Diseño Digital Principios y Practicas 3ra Edicion. PEARSON EDUCACIÓN S.A. ISBN: 970-J7-0404-5
- [2] Thomas L. Floyd. (2006). FUNDAMENTOS DE SISTEMAS DIGITALES. PEARSON EDUCACIÓN S.A., Madrid, 2006. ISBN 10: 84-8322-085-7
- [3] eference.digilentinc.com. 2020. Nexys 2 Reference Manual [Digilent Documentation]. [online] Available at: <https://reference.digilentinc.com/reference/programmable-logic/nexys-2/reference-manual> [Accessed 18 November 2020].
- [4] .bp.blogspot.com. 2020. [online] Available at: <http://2.bp.blogspot.com/-tkgzpdpfZKk/UmMn6QiOGII/AAAAAAAAAAu0/b-cGI9cfmFU/s1600/Frecuencia+de+las+notas+musicales+en+hertzios.jpg> [Accessed 18 November 2020].
- [5] s.wikipedia.org. 2020. Sintetizador. [online] Available at: <https://es.wikipedia.org/wiki/Sintetizador> [Accessed 18 November 2020].
- [6] ttps://www.naylampmechatronics.com/. 2020. Teclado matricial 4x4 - Tipo membrana. [online] Available at: <https://www.naylampmechatronics.com/interfaz-de-usuario/19-teclado-matricial-4x4-tipo-membrana.html> [Accessed 10 December 2020].
- [7] ttps://3.bp.blogspot.com/. 2020. Teclado Numerico. [online] Available at: <https://3.bp.blogspot.com/-vwgQNksjaQ/VSWGAgTNumI/AAAAAAAAAAmA/jipkKQmoyQ/s1600/Captura.PNG> [Accessed 10 December 2020].
- [8] ttp://www.learningaboutelectronics.com/. 2020. Filtro Pasa Bajo. [online] Available at: <http://www.learningaboutelectronics.com/Articulos/Filtro-paso-bajo.php> [Accessed 12 December 2020].