Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica
IEE2473 – Laboratorio de Electrónica Analógica y Digital

Experiencia 3: Piano Digital

Grupo 17
Juan Lorca - Kevin Justiniano

Índice

1. Trabajo de Laboratorio

1.1. Teclado: Driver PS2

La tarjeta *Basys*-3 posee de un puerto USB que permite conducir dos buses de bits por medio de un *open-collector driver*. Por medio de estos dos buses es que se lleva a cabo el protocolo $PS2$ que nos permitió comunicar uno de los teclados disponibles en el Laboratorio con la PFGA *Basys*-3.

Para lograr tal comunicación se programó en el lenguaje $Verilog$ el driver `Keyboard` obtenido en [1]. La siguiente figura ejemplifica la transmisión de los datos `ps2c` y `ps2d` que son enviados por el protocolo:

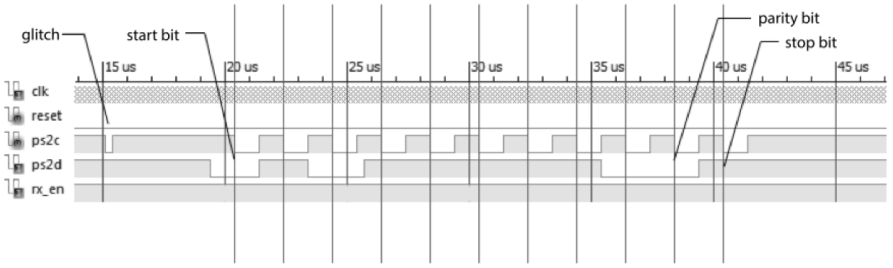

Figura 1: Protocolo $PS2$ [2]

Así, el módulo que fue programado como driver para este protocolo debía ser capaz de:

- No leer los *glitches* enviados por la señal `ps2c`.
- Detectar el bit de partida y leer los siguientes bits dados por `ps2d` en los flancos de bajada de la señal de reloj.
- Hacer una revisión si es que el bit de paridad era igual a 1.
- Guardar los 8 bits asociados a la codificación ASCII de la tecla presionada en un registro momentáneo que actuara como salida del módulo cada vez que se leía el *stop bit*.

El código implementado logra cumplir con los requisitos anteriormente mencionados. A continuación se muestra el bloque asocidado al driver y una descripción de su funcionamiento.

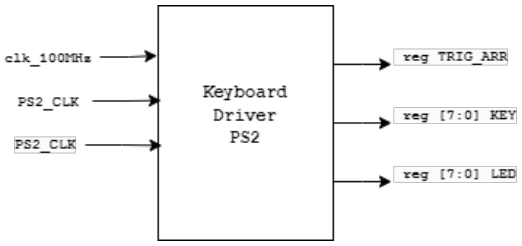

Figura 2: Diagrama Driver PS2

El módulo posee de los registros enseñados en la Tabla 1. Estos registros permitieron el seguimiento y control de las señales de entrada `ps2c` y `ps2c`:

Nombre `reg`	bits
`read`	1
`count_reading`	12
`PREVIOUS_STATE`	1
`scan_err`	1
`scan_code`	11
`CODEWORD`	8
`COUNT`	4
`TRIGGER`	1
`DOWNCOUNTER`	8

Tabla 1: Registros Driver

Además, el módulo posee de 5 procesos asociados al flanco positivo de la señal de entrada `clk_100MHz`:

(1) El primer proceso consiste en utilizar un contador dado por el registro `DOWNCOUNTER` para dividir el clock de entrada en 250 y asignar al registro `TRIGGER` el nuevo reloj deivido. Este úlitmo registro es utilizado en los siguientes procesos para tener un seguimiento mas preciso de las otras señales de entrada.

$$f_{trigger} = \frac{100\ MHz}{250} = 400\ kHz \tag{1}$$

(2) El segundo proceso es para ir aumentando el contador `count_reading` si es que el registro `read` = 1. Este contador nos fue útil ya que nos permite tener un registro de la cantidad de ciclos de $f_{trigger}$ que pasaron desde la última palabra de control enviada dada por el bir de partida, los datos, el bit de paridad y el bit de parada.

(3) En este proceso se leen los bits de la palabra de control enviada por `ps2d`. Cada vez que el clock entregado por el registro `TRIGGER` es 1, se verifica que el registro `PREVIOUS_STATE` sea distinto del valor del clock muestreado de la señal y `ps2c` para evitar *glitches* (ocurren a una frecuencia mayor que `TRIGGER`) y luego se lee la data en cada flanco negativo de `ps2c`. Esta data es almacenada en el registro `scan_code` utilizando el MSB como el primer bit de almacenamiento y el LSB como el último. Además para saber qué número de bit del mensaje se está muestreando se utiliza el contador `COUNT`. Cuando este útlimo llega a 11 (`COUNT` = 11) se calcula el registro `scan_error`.

(4) Este proceso está asociado a la revisión de los bits enviados dado el bit de paridad y el valor asociado a `scan_error`.

(5) Este último proceso viene a ser que que asigna a los registros de las salidas `KEY` y `LED` los datos del código asociado a la tecla enviada, almacenados desde el bit 8 al bit 1 del registro `scan_code`.

1.2. Oscilador de Frecuencias

La idea del oscilador de frecuencias radica en recorrer una memoria que tiene almacenada un cuarto de sinudoside con distintos saltos de indexación de de distintas formas. La memoria posee 128 muestras de un cuarto de sinusoide cuya amplitud se encuentra codificada en 8 bits. Así, si se recorre la memoria con una indexación constante, igual a 1 y de distintas formas se puede reconstruir una sinusoide completa de un máximo de 512 muestras.

1.2.1. Diagrama de Bloques

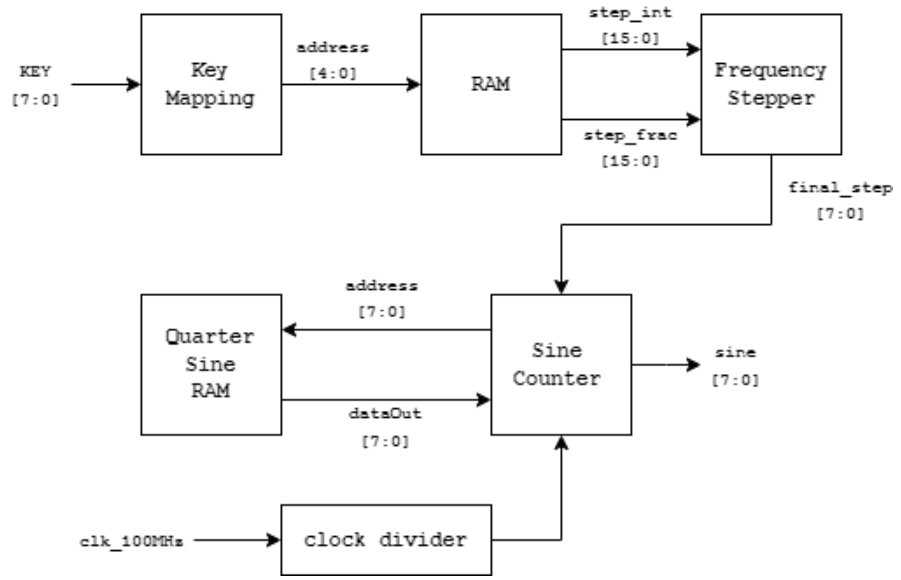

Figura 3: Diagrama de Bloques Oscilador Local

Considerar que todos los bloques requieren de un clock el cual es el reloj maestro de la la *Basys*-3 de 100 MHz.

1.2.2. Indexación de RAM

Para la indexación de la memoria que contiene al cuarto de ciclo de una sinusoide de amplitud 127.5 y una componente DC del mismo valor, fue necesario considerar lo siguiente:

- El salto entre muestras de la sinusoide iba a ser variable y su valor estaba dado por la entrada `final_step`. Esto último implicaba que iba a haber casos donde el valor del índice fuera mayor o menor que 0 o 127 (valores mínimos y máximos de las direcciones de la memoria).
- Restando el valor de la memoria indexada a 255 (valor máximo sinusoide) se lograba invertir la señal, si la memoria era recorrida en orden creciente y decreciente. Por lo tanto, existen 4 estados a analizar: 2 para un semi-ciclo positivo y 2 para un semi-ciclo negativo.

- Implementar por medio de registros todo lo que sea cálculos o almacenamientos. Los registros logran cambiar de acuerdo a un proceso dado por un reloj (flanco positivo o flanco negativo). Esto último permite ir actualizando los valores necesarios sin retrasos y así lograr una forma de onda bien digitalizada y sin ninguna muestra equivocada.

Considerando el primer punto se utilizó el registro `counter` de 8 bits para llevar a cabo la indexación. Este registro se incrementa o disminuye de acuerdo al estado en que se encuentre otro registro llamado `enable_back`. Para que el contador no se pasara de los límites superiores e inferiores dados por las direcciones de la memoria, para el caso del límite superior, con un condicional '`if`' se determinaba si el valor de:

$$\texttt{counter} + \texttt{step} > \texttt{RAM\_DEPTH}$$

y luego se le asignaba al registro `counter` el valor:

$$\texttt{counter} <= \texttt{RAM\_DEPTH} - (\texttt{counter} + \texttt{step} - \texttt{RAM\_DEPTH})$$

a su vez que se activaba el registro `enable_back` para indicar que el contador comenzara a disminuir. Para el caso del límite inferior, la lógica fue similar si considerabamos `counter` como una variable *unsigned* o sin signo. Esto útlimo significaba que si `counter` llegaba ser menor que 0, su valor pasaba a estar mas cerca o ser igual a 255. Esto útlimo hizo que se implementara la siguiente condición:

$$\texttt{counter} - \texttt{step} > \texttt{RAM\_DEPTH}$$

y luego se le asignaba al registro `counter` el valor:

$$\texttt{counter} <= 2 \cdot \texttt{RAM\_DEPTH} + 1 - (\texttt{counter} - \texttt{step})$$

Tanto la memoria para almacenar el ciclo de sinusoide como las frecuencias necesarias para generar los distintos tonos fueron implementadas utilizando el *IP Core Block Memory Genrator* de Vivado. Para la memoria de las frecuencias, se almacenaron los steps asociados en 24 direcciones y se utilizaron 32 bits para cada step, dónde 16 fueron para la parte entera y los otros 16 para la parte fraccionaria.

1.3. Conversor Análogo-Digital (DAC)

Se realiza el circuito DAC con resistencias de presición de valor, donde R es 3.57 kΩ y 2R se logra con un arreglo de resistores en paralelo, tal de repromediar el error que tienen los elementos debido a su tolerancia. En base a esto se envían dos notas LA, una estando una escala sobre la otra, las señales de voltaje se pueden observar en las imágenes 4 y 5

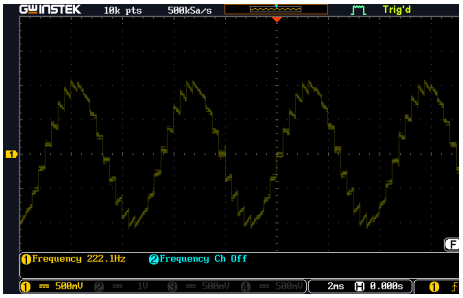

Figura 4: señal de nota LA de 220 Hz

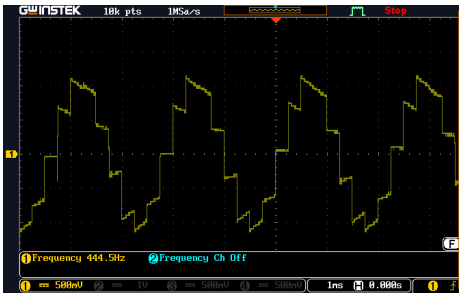

Figura 5: señal de nota La de 440 Hz

Ambas señales ostentan de igual amplitud, con saltos más o menos grandes debido a la entrega de señal cada entrada relacionada al bit de emisión desde la basys. Posterior a esto, se reordenan las señales de entrada del circuito DAC, tal de tener siete bits de entrada en vez de ocho como se ve en la figura 6. Al compararla con una su misma señal de 8 bits, se observa una atenuación del 50 % de la de 7 bits frente a la original.

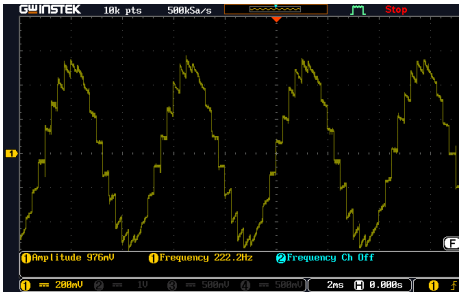

Figura 6: señal de nota La de 220 Hz cuantizada por 7 bits

A continuación, se conecta el DAC al amplificador operacional y filtro activo, en este caso el filtro es un integrador con pérdidas a a altas frecuencias. El diseño del sistema es para lograr entregar los 3 V a la salida del amplificador clase AB, así como de tener una frecuencia de corte del filtro pasabajos en 500 Hz aproximadamente. La señal de salida total se muestra en la figura 7 donde se puede apreciar la señal amarilla de entrada y la azul que es la señal de salidacon una amplitud de salida es de aproximadamente 9 V.

1.4. Etapa de salida clase AB sin realimentación

Se contruye el amplificador clase AB tal de obtener una corriente de gate lo suficientemente grande como para polarizar correctamente los gates de los BJT, para

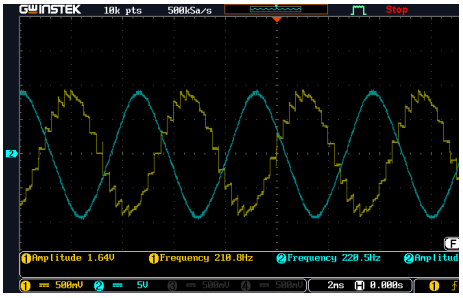

Figura 7: señal de voltaje de entrada y salida de amplificado y filtro

lograr dicha polarización se usaron resistores de potencia máxima de 0.25 W y que permitiesen pasar 10 mA, en base a estas condiciones se utilizaron resistores de 2200 Ω. Del mismo modo, para evitar el ciclo de retroalimentación positivo generado por el aumento de temperatura, se hizo uso de resistores de 6 Ω.

Una vez armado el circuito, se procede a hacerlo funcionar y medir las señales de voltaje de la salida del DAC, del filtro y amplificador, y del amplificador AB, los resultados se pueden ver en la imagen 8, donde se ve que la tensión de salida es de aproximadamente 2 V peak-peak (señal verde), mientras que las señales azul y verde, son la de la salida del DAC y filtro respectivamente.

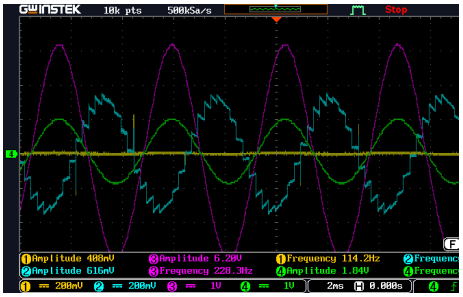

Figura 8: señales totales del circuito

Por último se calcula el THC para el total de notas teniendo en cuenta los tres primeros armónicos. obteniendo los resultados que se muestran en la tabla 2. Del mismo modo se muestra el espectograma de una señal de 440 Hz en la imagen 9, donde se puede evidenciar como la frecuencia fundamental es la que posee la gran mayoría de la potencia, pero como los ármonicos que son divisibles por la fundamental tienen un peak importante de igual manera.

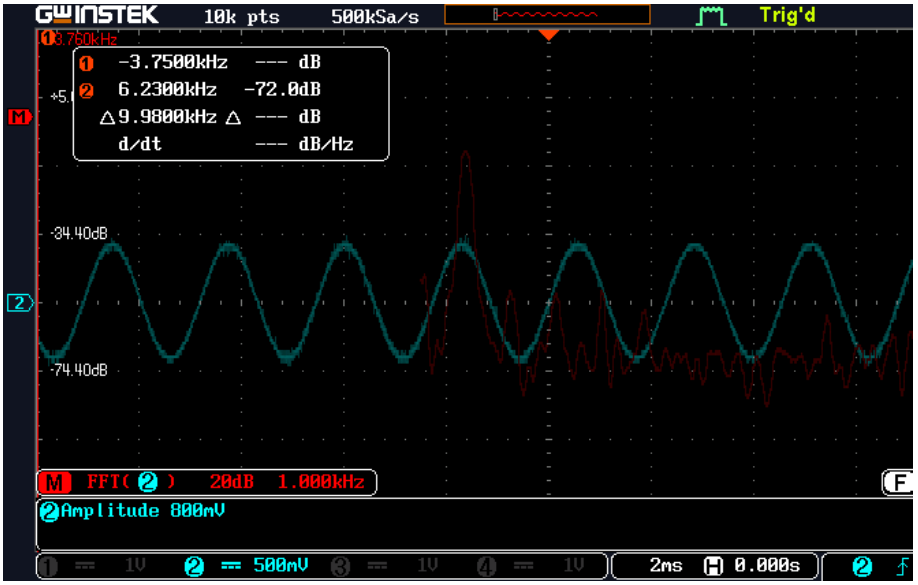

Figura 9: espectograma de La de 440 Hz

La mayor diferencia frente a la simulación son las amplitudes de las señales de salida, se registra una mayor atenuación en el circuito elavorado que en el simulado. Del mismo modo, al mover el potenciómetro P_f se ve como la señal se distorsiona colapsando la amplitud poco a poco a 0, esto debido a que al existir una menor resistencia en la rama paralela a los didos, la corriente fluye por allí, bajando la tensión de polarización que generan los diodos para el BJT.

Nota	Fundamental	1er H	2do H	THC [%]
DO	-11.2	-46.4	-60	2.10
Do	-10.4	-49	-49	1.70
Re	-10.4	-52	-56	1.31
Re	-11.2	-46.4	-56.8	2.10
Mi	-11.2	-46.4	-60	2.10
Fa	-10.4	-49	-60	2.0
FA	-10.4	-52	-52	1.66
Sol	-10.4	-46.4	-52	2.42
Sol	-11.2	-46.4	-52	2.10
La	-10.4	-52	-52	1.66
La	-11.2	-52	-52	1.82
Si	-10.4	-52	-56	1.36
DO	-10.4	-46.4	-52	2.42
Do	-10.4	-46.4	-56	2.11
Re	-11.2	-52	-56	1.49
Re	-10.4	-52	-52	1.66
Mi	-10.4	-52	-52	2.41
Fa	-11.2	-46.4	-52	2.65
FA	-11.2	-49	-56	1.86
Sol	-11.2	-52	-52	2.20
Sol	-11.2	-49	-52	2.01
La	-10.4	-52	-52	1.66
La	-10.4	-52	-52	1.66
Si	-10.4	-52	-52	1.66

Tabla 2: resultados de harmónicos y THC

Referencias

[1] Autodesk Instructables. (2021). *PS2 Keyboard for FPGA*. Instructables. Recuperdado de: `https://www.instructables.com/PS2-Keyboard-for-FPGA/`

[2] C. Garcés, *Experiencia 3: Piano Digital*. Laboratorio de Electrónica Analógica y Digital. IEE2473, Pontificia Universidad Católica de Chile, 2023.