

Implementación de una Caja Musical para la canción del “Feliz Cumpleaños”

Nicolás Luna, Mercedes Gorría
Universidad Nacional del Litoral - FICH

Resumen—El objetivo principal de este trabajo es explorar el proceso de diseño, desarrollo e implementación de una caja musical digital capaz de reproducir la canción del "Feliz Cumpleaños" sobre una FPGA. A lo largo del informe, se describe paso a paso la metodología utilizada para llevar a cabo el diseño, así como las decisiones tomadas para resolver los desafíos encontrados durante el proceso. Para lograr el objetivo, se adoptó un enfoque basado en la integración de los conceptos vistos de electrónica digital y programación en verilog. Además, se utilizó la modularización de las tareas del proyecto, para que se pueda desempeñar cada función específica dentro del sistema.

En resumen, este informe documenta el diseño, desarrollo y los retos superados en la implementación de una caja musical digital para reproducir la melodía del "Feliz Cumpleaños". El enfoque modular y las soluciones implementadas subrayan la importancia de aplicar tanto los fundamentos teóricos como las técnicas prácticas en el diseño y desarrollo de sistemas digitales.

Palabras clave: Caja Musical, Generación de Señales Cuadradas, Feliz Cumpleaños FPGA.

I. INTRODUCCIÓN

El presente informe tiene como objetivo principal abordar el desarrollo de una caja musical digital programable a través del lenguaje verilog, capaz de reproducir el arreglo musical de la canción "Feliz Cumpleaños" utilizando un altavoz (speaker) con conexión a una FPGA. El diseño propuesto consta de varios módulos interrelacionados que representan funciones específicas utilizadas en conjunto para poder llevar a cabo el objetivo con el fin de simplificar la explicación del código y obtener una abstracción que cree un entorno más ordenado y legible para el lector.

En primer lugar, se optó por un módulo principal, es decir, un módulo capaz de encapsular todo el conjunto de funciones o representaciones a utilizar para la composición de la canción propuesta. Además, este será el responsable de generar la señal de audio que se envía.

Seguidamente, se emplean los demás módulos contenidos dentro del módulo principal los cuales permiten almacenar nuestro arreglo de notas y controlar el flujo de la melodía. Juntos, permiten obtener la frecuencia sonora de cada nota a través de un contador y el tiempo de cada nota para la reproducción de la melodía.

En este informe, se detalla tanto sobre los aspectos teóricos y prácticos del diseño, describiendo el funcionamiento de cada uno de los módulos utilizados y dándole una suma importancia a la integración de conceptos de electrónica digital, programación en verilog y física conjunto a la teoría musical para crear un dispositivo interactivo.

Este proyecto se centra inicialmente sobre la melodía del "Feliz Cumpleaños", sin embargo, la metodología desarrollada trasciende esta canción en específico. Gracias al enfoque modular y la comprensión de la relación entre notas musicales y frecuencias sonoras, se podría decir que este sistema es fácilmente adaptable a una amplia variedad de canciones. Esta versatilidad demuestra cómo se pueden crear plataformas de reproducción de señales flexibles, creando diversas experiencias musicales cotidianas, desde melodías tradicionales como el “Feliz Cumpleaños” hasta composiciones contemporáneas.

II. MATERIALES

A continuación, se presentan los materiales y especificaciones técnicas que se utilizaron para la implementación del trabajo, así como también descripciones justificando el uso de cada uno.

- i. FPGA (field programmable gate array) : El trabajo se impactó sobre una EDU-CIAA-FPGA, la cual es una placa diseñada por el Grupo de Investigación y Desarrollo de Sistemas Embebidos (ASE).
- ii. Software: Se utilizó el software VSCode como entorno de desarrollo para los módulos en Verilog, junto con GTKWave para la verificación y visualización de señales.
- iiii. Hardware adicional: Se reprodujo la melodía a través de un parlante (speaker) y una resistencia.

III. MÉTODOS Y DISEÑO

Análisis del Problema:

El objetivo de este proyecto es crear una caja musical que pueda reproducir la melodía del "Feliz Cumpleaños" utilizando Verilog y conceptos de Electrónica Digital. La solución implica diseñar un sistema que genere una señal de audio precisa, capaz de reproducir las notas de la melodía mediante la implementación de tres módulos interrelacionados:

1. Generador de Señal de Audio: Este módulo es responsable de generar la señal de audio que se enviará al altavoz.
2. Almacenamiento de Frecuencias de Notas: Este módulo almacena las frecuencias de las notas necesarias para la melodía.
3. Controlador de Flujo de la Melodía: Este módulo determina cuándo cambiar de una nota a la siguiente, controlando el flujo de la melodía.

Especificaciones Técnicas:

- Conexión del Parlante: El parlante se conectará al PIN 107.
- Frecuencia del Clock de la FPGA: Se utilizará el clock disponible en el PIN 94, con una frecuencia de 12 MHz (12,000,000 Hz).

Generación de la Señal Cuadrada

La siguiente tabla muestra las notas musicales utilizadas en la melodía "Feliz Cumpleaños" junto con sus frecuencias aproximadas (en Hertz) y los valores correspondientes que se utilizan en el diseño.

TABLA I

ASIGNACIÓN FINAL DE NOTAS Y SU RESPECTIVA FRECUENCIA UTILIZADAS EN EL DISEÑO.

Nota	Frecuencia (Hz)	Número de ciclos
Do	262	22900
Re	294	20407
Mi	330	18181
Fa	349	17191
Sol	392	15305

La	440	13635
La#	466	12875
+Do	523	11471

La fórmula a continuación,

$$N = \frac{f_{clk}}{2 \cdot f_{nota}} \quad (1)$$

se utiliza para calcular el número de ciclos del reloj necesarios para generar una señal cuadrada de una frecuencia específica. Aquí están los componentes clave de la fórmula:

- N : Número de ciclos del reloj que determinan el periodo de la señal cuadrada.
- f_{clk} : es la frecuencia del reloj.
- f_{nota} : es la frecuencia de la nota musical.

Para generar una onda cuadrada con una frecuencia específica de una nota musical a partir de una señal de reloj (CLK), se utilizará un contador y un comparador. El procedimiento es el siguiente:

1. Cálculo de ciclos: Se calcula el número de ciclos de reloj necesarios para cada nota usando la fórmula mencionada. Este valor será guardado en la rom.
2. Contador: Un contador incrementa su valor con cada ciclo del reloj. Este contador se compara continuamente con el valor N .
3. Comparador y Oscilación de la Señal: Cuando el valor del contador alcanza N , se produce un cambio en la señal de salida. Esta cambia de 0 a 1 o de 1 a 0, generando así una onda cuadrada. Después, el contador se reinicia y el ciclo comienza de nuevo.

Diseño e implementación de módulos:

Los módulos descritos a continuación detallan la implementación del sistema que reproduce la melodía de "Feliz cumpleaños" utilizando tres módulos principales: happybday, contador y rom. A continuación, se presenta una explicación detallada sobre el diseño de cada uno de estos módulos y cómo funcionan.

A. Módulo happybday.

Este módulo se creó con el objetivo de generar la señal final que reproducirá la canción del "Feliz cumpleaños". Este cumple el rol de módulo principal en el sistema, el cual gestiona tanto la señal de reloj de entrada como la señal de salida. Además, se encarga de coordinar la interacción entre los demás submódulos del sistema.

El módulo tiene una entrada principal, la señal de reloj (clk). Esta entrada sincroniza todas las operaciones en el sistema. La salida principal, *speaker*, es una señal que se conecta al altavoz, alternando entre 0 y 1 para generar una onda cuadrada que produce el sonido.

Existen señales intermedias cruciales para el funcionamiento del módulo. La señal *ciclo_de_nota*, que es un cable de 16 bits, transporta el número de ciclos para la nota actual desde el módulo ROM. La señal *direccion_nota*, que es un cable de 5 bits, transporta la dirección de la nota actual desde el módulo Controlador. Finalmente, la señal *contador_tiempo*, que es un cable de 24 bits, transporta el tiempo de conteo actual desde el módulo Controlador para controlar la duración de la nota.

"Happybday" también utiliza registros internos para su operación, como el registro *contador*, de 16 bits, que cuenta los ciclos del reloj hasta alcanzar el valor de ciclos necesarios de la nota actual.

Para asegurarnos que el sistema comience en un estado conocido y controlado, inicializamos las variables *speaker* y *contador* en 0. Esto lo realizamos en el bloque inicial de dicho módulo:

1. Fragmento de Código del módulo “Happybday”.

```
initial begin
    speaker = 0;
    contador = 0;
end
```

En cada flanco positivo del reloj (*posedge clk*), dentro del bloque *always* se incrementa *contador* hasta que alcanza el valor de *ciclos_de_nota*. Cuando esto sucede, se alterna el valor de *speaker* y se reinicia el contador:

2. Fragmento de Código del módulo “Happybday”.

```
always @(posedge clk) begin
    if (contador >= ciclos_de_nota) begin
        speaker <= ~speaker;
        contador <= 0;
    end else begin
        contador <= contador + 1;
    end
end
```

Este proceso de alternancia de *speaker* entre 0 y 1 a intervalos regulares genera una onda cuadrada con una frecuencia que corresponde a la nota actual. El registro contador se incrementa en cada ciclo de reloj y, cuando *contador* alcanza el valor de *ciclos_de_nota*, *speaker* cambia de estado, de 0 a 1 o de 1 a 0, y *contador* se reinicia a 0. Esto crea un ciclo repetitivo que genera la onda cuadrada.

De esta manera, el módulo happybday convierte la información de frecuencia de reloj en nuestra señal final, que se escuchará luego a través del parlante.

B. Módulo ROM.

El módulo ROM es una memoria de solo lectura que es responsable de almacenar los ciclos de reloj necesarios para generar las notas que componen la melodía "Feliz Cumpleaños". Su principal función es proporcionar el número correcto de ciclos de reloj para cada nota, según la dirección dentro de la memoria, al módulo principal que genera la señal de audio. Esto se logra mediante el uso de una memoria interna implementada como un arreglo de registros, donde cada posición almacena los ciclos de reloj para una nota específica. Cabe destacar, que cada registro almacena el número de ciclos previamente calculado con la ecuación (1) para nuestra FPGA. En una implementación diferente, pueden necesitar ajustes basados en el reloj del sistema.

Se optó por utilizar una memoria ROM por motivos de practicidad, ya que el sistema no requiere la necesidad de escritura de datos. Además, tiene ventajas como el poder acceder rápidamente a sus elementos sin necesidad de realizar cálculos complejos en tiempo real.

Además, resultó conveniente utilizar 16 bits para cada registro, a pesar que técnicamente es posible utilizar solo 15 bits para esto. Utilizar un tamaño de palabra que sea una potencia de dos (como 8, 16, 32 bits) es una práctica común en el diseño de hardware ya que simplifica la lógica de control y la eficiencia. Esta decisión ofrece mayor flexibilidad, proporcionando un margen adicional para futuros cambios o expansiones.

C. Módulo controlador:

El módulo controlador es responsable de generar la secuencia de direcciones de nota que determina la melodía que se reproduce. Utiliza una máquina de estados simple para contar el tiempo y controlar la transición entre las diferentes notas en la melodía. Su objetivo principal es dar la dirección de la nota actual al módulo rom, de modo que este pueda recuperar el número de ciclos correspondiente a dicha nota almacenada en su memoria interna. Además, el controlador se encarga de la importante tarea de controlar la duración de cada nota, asegurando que se mantenga durante un tiempo específico antes de avanzar a la siguiente nota.

El módulo tiene como entrada la señal de reloj que sincroniza todas las operaciones. Tiene dos salidas, *direccion_nota* que es una salida de 5 bits que representa la dirección de la nota actual y *contador_tiempo*, de 24 bits, que cuenta el tiempo y determina cuándo cambiar a la siguiente nota. Además, utiliza registros internos para su operación, como *tiempo_por_nota*. Este parámetro define la duración fija de cada nota. Así, para que cada nota tenga una duración de medio segundo para el clk de 12 MHz, *tiempo_por_nota* debe ser igual a 6000000. En una implementación diferente, pueden necesitar ajustes basados en el reloj del sistema.

El contador se inicializa en el “bloque inicial” con la dirección de la nota y el tiempo de conteo establecidos en 0.

3. Fragmento de Código del módulo “Controlador”.

```
initial begin
    contador_tiempo = 0;
    direccion_nota = 0;
end
```

El módulo controlador utiliza un bloque always sensible al flanco positivo del reloj (*posedge clk*) para incrementar el tiempo de conteo (*contador_tiempo*). Cuando *contador_tiempo* alcanza el valor predefinido de *tiempo_por_nota*, se incrementa la dirección de la nota (*direccion_nota*). Si *direccion_nota* alcanza el límite de notas, en este caso 24, se reinicia a 0 para comenzar nuevamente la melodía.

El módulo controlador es fundamental para controlar el tiempo y las transiciones entre notas en la melodía "Feliz Cumpleaños". Su diseño simple y eficiente permite que el sistema cambie de una nota a otra de manera precisa y sincronizada con el reloj del sistema. De esta manera, el módulo genera una secuencia de direcciones de nota que se repite cíclicamente, controlando la duración de cada nota y sincronizando la reproducción de la melodía con el módulo rom y el módulo principal happybday.

Verificación

La verificación del diseño se realiza mediante un testbench, que simula el funcionamiento del módulo happybday y comprueba que las señales de salida sean correctas. El testbench genera una señal de reloj (clk) de 12 MHz, cuyo periodo (T) es de 1/12,000,000 segundos. Convertido a nanosegundos, T es aproximadamente 83.3333 ns. La mitad de este período, que corresponde al tiempo durante el cual el reloj debe permanecer en un estado antes de cambiar, es 41.6667 ns. Para generar esta señal de reloj, se utiliza la siguiente declaración:

4. Fragmento de Código del “Testbench”.

```
always begin
    #41.6667 clk = ~clk;
end
```

La señal de clk alterna entre 0 y 1 cada 41.6667 nanosegundos, creando un ciclo completo de 83.3333 nanosegundos, lo que corresponde a la frecuencia deseada de 12 MHz. Esta señal de reloj es utilizada por el testbench para simular el comportamiento del circuito digital a la velocidad correcta.

Se instancia el módulo happybday dentro del testbench, conectando la señal de reloj y observando la salida del altavoz (speaker). Se utiliza *\$dumpfile* y *\$dumpvars* para registrar todas las señales durante la simulación en un archivo VCD. Este archivo puede ser visualizado con herramientas de simulación como GTKWave para observar las transiciones de las señales y verificar el correcto funcionamiento del sistema. La simulación se ejecuta por un tiempo suficiente para cubrir al menos una estrofa de la melodía. En este caso, se simulan 3 segundos, lo que permite verificar las primeras notas de la melodía "Feliz Cumpleaños".

La señal de salida alterna en los momentos correctos para cada nota, según las frecuencias almacenadas en la ROM. La transición entre notas ocurre en los intervalos de tiempo adecuados. Se puede observar la reproducción de la melodía en el archivo VCD generado, lo que verifica que el diseño implementado en Verilog cumple con los requisitos funcionales del proyecto.

IV. RESULTADOS

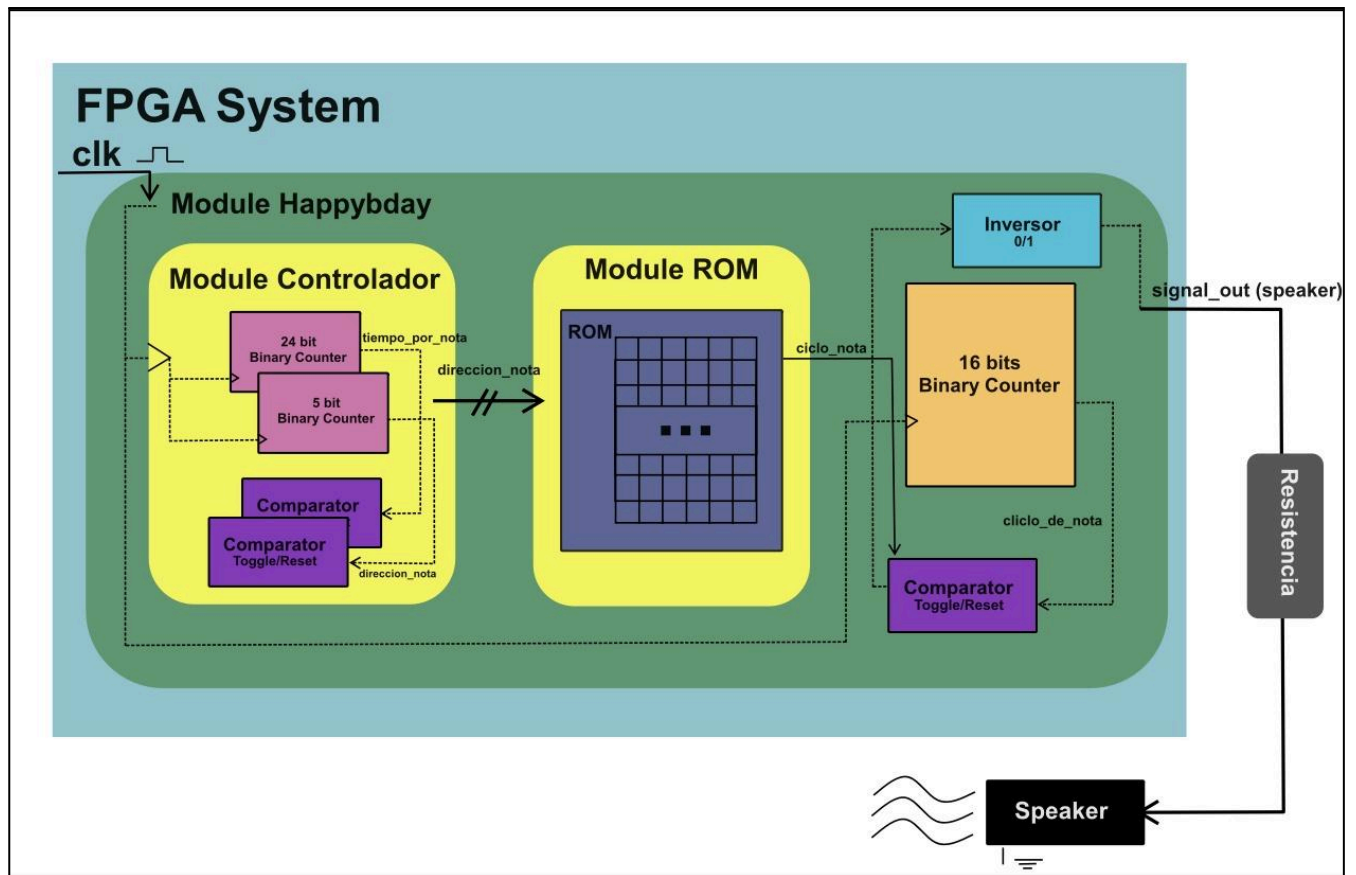


Fig. 1: Representación del Sistema sobre FPGA para la Reproducción de la melodía "Feliz Cumpleaños".

Se obtuvo como resultado final el sistema descrito anteriormente en la etapa de métodos y diseño, ilustrado en la figura 1, con sus respectivos módulos Happybday, Controlador y ROM interactuando de manera capaz de generar y reproducir la canción deseada. Impactando sobre una FPGA (EDU-CIAA-FPGA), este diseño muestra la implementación efectiva del proyecto dado y se confirma en la práctica la funcionalidad efectiva del diseño propuesto, además de su potencial de aplicabilidad en proyectos similares de generación de audio con cualquier otra melodía posible mediante sistemas digitales.

IV. CONCLUSIONES

En conclusión, se diseñó una caja de música que reproduce la canción "Feliz Cumpleaños" utilizando una FPGA y módulos programados en el lenguaje Verilog. Además, se logró conectar de manera óptima los distintos componentes del sistema para poder generar la canción y reproducirla a través del altavoz.

Se crearon varios módulos en Verilog, como: happybday (encargado de coordinar todo y generar una señal cuadrada), rom (que registra las frecuencias de las notas) y controlador (que se encarga de cambiar una nota a otra en el momento adecuado). Cada uno de estos, jugó un papel importante en la reproducción de las notas para que sonaran correctamente.

Se utilizaron herramientas como VSCode para la programación en Verilog y GTKWave para verificar las señales generadas. Esta verificación fue muy importante ya que nos permitió asegurar que las señales de audio se generan correctamente y que las notas se tocan en el orden y tiempo correcto, además de los “testbench” que nos ayudaron a probar todo el sistema antes de afectar la FPGA, evitando así cualquier posible error.

Por último, se pudieron aplicar los conocimientos obtenidos a lo largo de la asignatura sobre el diseño e implementación de sistemas digitales para procesar, transmitir y almacenar información digital, demostrando el amplio campo de aplicación de dichos conocimientos.

BIBLIOGRAFÍA

- Fundamentos de Sistemas Digitales, vol 1, 1997, Thomas L. Floyd.
- Diseño digital: Principios y prácticas, vol 1, 1994, John F. Wakerly.