

Diseño de SoC mínimo para reloj despertador

David Cordero Chavarría, Emanuel Esquivel López, Ricardo Valverde Hernández

dcorderoch@ieee.org email1412@estudiantec.cr rivh@estudiantec.cr

Área Académica de Ingeniería en Computadores

Instituto Tecnológico de Costa Rica

Abstract—Embedded systems usually implement a custom-tailored system to perform a specific function, this can range from sensor reporting, monitoring, to home automation, complex systems can be made by combining several of them, providing resiliency to errors by making each an independent component, capable of operating even if other components fail. These systems are very much in vogue, so understanding their capabilities and challenges can be very advantageous, it is therefore very important to know these systems. In this paper we present a custom-made alarm, implemented in an FPGA, that could be extended to use more of the capabilities provided by the FPGA, to provide a proof-of-concept of a relatively simple smart device, be it offline or online, though we will focus solely on the alarm, without any external connectivity.

Palabras clave—FPGA, SoC, C, Verilog.

I. INTRODUCCIÓN

Los sistemas embebidos son muy importantes en la actualidad ya que estos nos permiten realizar de mejor manera distintos proyectos pequeños los cuales nos funcionan como pequeños sistemas independientes, en general estos son capaces de funcionar independientemente de su localización siempre y cuando tengan la energía necesaria para su operación, por lo que es muy común tener sistemas con sus propias alimentaciones y funcionan de manera independiente.

En este documento primero introduciremos al lector un posible sistema de alarma, un posible diseño de este, luego una posible implementación en una FPGA, resultados de la implementación, y por último una reseña de lo aprendido en el desarrollo de este proyecto.

Un sistema muy común utilizado en la actualidad son los despertadores o alarmas, los cuales son sistemas independientes que funcionan para llevar un control del tiempo, como se sabe estos normalmente cuentan 2 modos, para el correcto funcionamiento, los cuales son las:

- Set de la hora y ver como reloj.
- Set de alarma.

Con esto podemos manejar un despertador de manera independiente, y con ayuda de controladores y hardware independiente podemos hacer una muy eficiente, mediante el uso de un SoC FPGA, para poder realizar todas las pruebas necesarias.

Otro aspecto de mucha importancia es la creación de procesadores embebidos mediante el uso de la plataforma de diseño

II. SISTEMA DESARROLLADO

En el diagrama de la figura 1 se puede ver como esta estructurado el sistema, como se ve las entradas a este son el

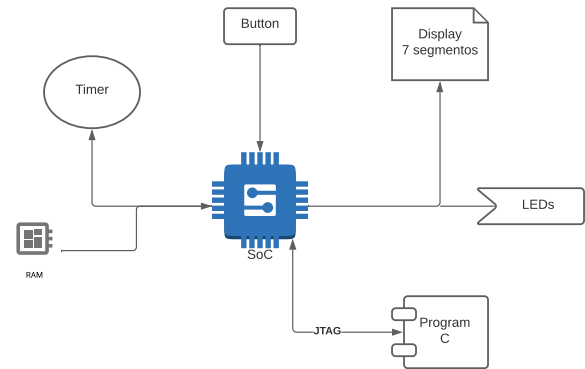


Figure 1. Diagrama de Arquitectura del sistema

programa principal el cual esta en C, este es cargado mediante conexión JTAG, desde una computadora, así como tenemos de entrada botones los cuales funcionan para setear la alarma, tenemos un timer, el cual se encarga de llevar el control de la hora así como de salidas los leds y display de 7 segmentos.

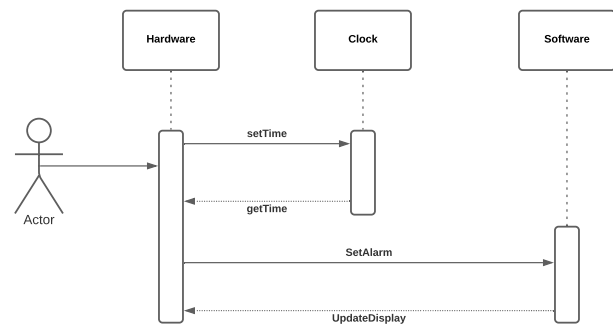


Figure 2. Diagrama de secuencia del sistema

Para este diagrama se puede ver que el usuario interactúa directamente con el hardware, el cual se utiliza para monitorear la alarma y esta interactúa directamente con el software el cual se encarga de mantener el control del tiempo.

III. RESULTADOS

En la figura 3 se puede ver el resultado del reloj en los displays de 7 segmentos de la tarjeta de desarrollo, y en la figura 4 se puede ver la alarma (un LED bajo un sticker, sobre el punto en que acaban los switches y empiezan botones los botones).

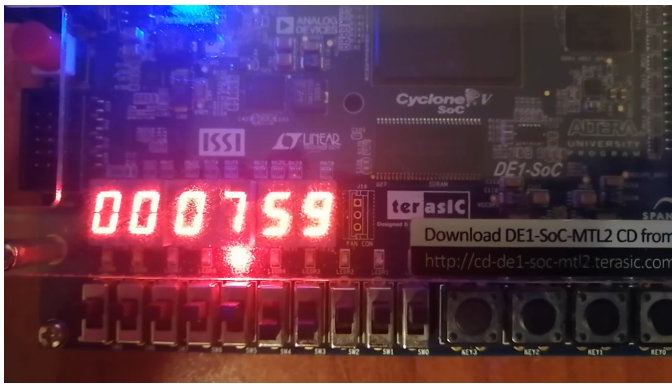


Figure 3. Reloj funcionando

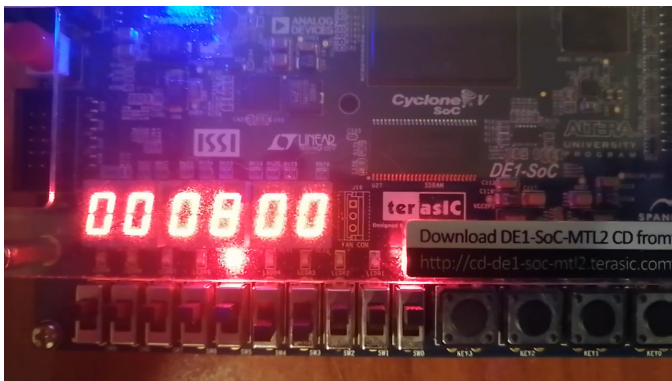


Figure 4. Alarma funcionando

IV. CONCLUSIONES

Es posible crear soluciones personalizadas a partir de componentes sencillos (como los displays de 7 segmentos), que puedan además tener comportamiento complejo, si se tiene el suficiente conocimiento de una plataforma, como la FPGA de Intel/Altera, y se entiende bien lo que se desea realizar. Aunque este sistema tal vez no puede ser usado como una alarma en su forma presente, es una buena ilustración de cómo usar un sistema hecho a la medida de componentes ya existentes, o de tarjetas de desarrollo.

REFERENCES

- [1] S. Pedre, "Sistemas embebidos," *Laboratorio de Robótica y Sistemas Embebidos, Departamento de computación FCEN UBA*, 2017.
- [2] A. Mardan, *Express.js Guide: The Comprehensive Book on Express.js*. Azat Mardan, 2014.
- [3] M. Desnoyers and M. R. Dagenais, "The ltnng tracer: A low impact performance and behavior monitor for gnu/linux," in *OLS (Ottawa Linux Symposium)*, vol. 2006. Citeseer, 2006, pp. 209–224.