

Diseño de Procesador RISC en FPGA

Ramon Ulises-Almada Prieto^a, Roberto Herrera-Charles^a, Teodoro Álvarez-Sánchez^a, Gilberto Enrico-Vazquez Alcaraz^b.

^aInstituto Politécnico Nacional, IPN-CITEDI, ramon_ulises@citedi.mx, nunez@citedi.mx, robcharles@citedi.mx, talvarez@ipn.mx, Tijuana, Baja California, México.

^bTecnológico Nacional de México, Instituto Tecnológico de Tijuana, gilberto.vazquez@tectijuana.edu.mx, Tijuana, Baja California, México.

Resumen

Reconfigurable computing (on FPGAs) opens the doors to the design of both well-known and recent innovative devices. In this article, a soft-core processor was implemented on an FPGA board, the XC7Z010clg400-1 from the Zynq-7000 family. This processor was designed in VHDL language using functions representing each of its stages. Similarly, the processor's ALU features 7 distinct operations and can expand up to 16 operations. The main contribution of this research lies in the design of a 4-bit RISC-type soft-core processor. The characteristic of this processor is that instructions are executed in a single machine cycle, which can be easily scaled to N number of bits per input.

Palabras clave— FPGA, Procesador, soft-core, RISC.

Abstract

Reconfigurable computing (FPGAs) opens the doors to designs of various devices already known, as well as recent and innovative ones. In this article, a soft-core processor with RISC-type architecture was implemented on an FPGA board, specifically the XC7Z010clg400 from the Zynq-7000 family. This processor was designed in VHDL language using functions that represent each of the processor's stages. Similarly, the ALU of the processor has 7 different operations and can easily be expanded to 16 if required. The main contribution of this research lies in the design of a 4-bit soft-core processor with RISC architecture with one machine cycle for every five clock cycles, which can be easily scalable to N number of bits per input.

Keywords— FPGA, Processor, soft-core, RISC.

1. INTRODUCCIÓN

Los procesadores están presentes en cualquier ámbito de la vida, estos se encuentran en distintas presentaciones, algunos ejemplos son las computadoras, dispositivos celulares, electrodomésticos, entre otros. Los procesadores son una gran herramienta para resolver una gran variedad de problemas y simplificar la vida de quienes las utilizan.

De igual forma, las tarjetas de arreglos de compuertas programables en campo (en inglés, FPGA), comienzan a cobrar gran relevancia y a volverse más utilizables debido a su gran versatilidad. Una gran ventaja que presentan los dispositivos FPGA es que permiten realizar ingeniería inversa

a distintos aparatos electrónicos con mayor facilidad para así emular su funcionamiento. Adicionalmente, las tarjetas FPGA son de bajo consumo y poseen altas velocidades de procesamiento, por lo que es posible utilizarlas para diversas aplicaciones, desde la emulación de un procesador, hasta el procesamiento de imágenes o video en tiempo real.

Tomando en consideración lo anterior, para emular un procesador o cualquier dispositivo electrónico, es importante que se conozca a la perfección el funcionamiento de dichos dispositivos a diseñar, se deben respetar a la perfección la cantidad de entradas y salidas que posee el sistema, así como los tiempos de retardo o de procesamiento que les toma a los dispositivos diseñados generar una salida en función de las señales de entrada. De igual forma, para la emulación de un dispositivo que se encuentra en constante comunicación con otros, es importante conocer a la perfección el funcionamiento del protocolo de comunicación que poseen dichos dispositivos, de lo contrario, aunque el dispositivo emulado se comporte de manera correcta, al fallar en el protocolo de comunicación, la transmisión de los datos será errónea.

Actualmente existen trabajos en los que se reporta el diseño de dispositivos electrónicos, tal es el caso de D. Lorente en [1] quien diseño un procesador digital de audio y lo implemento en la tarjeta FPGA Basys 3. Por otro lado, también existen investigaciones en donde se diseñan procesadores soft-core (procesadores diseñados por software), tal es el caso de F. Olivero, F. Iturriaga y C. Almaraz quienes en [2] diseñaron un procesador soft-core de tipo OISC (una sola instrucción) de 8-bits en una tarjeta FPGA, dicha implementación requería únicamente 5 slices (3 CBL: Bloques Lógicos Configurables) y 8 LUTs (en inglés, Look-up Tables) de recursos lógicos, mientras que al aumentar el número de bits, los recursos requeridos aumentaban, teniendo que para un procesador de 15-bits el sistema utiliza 52 slices y 81 LUTs.

En este trabajo de investigación se diseñó un soft-core de 4 bits tipo RISC y se implementó dicho sistema en la tarjeta FPGA xc7z010clg400-1. Para cumplir cada una de las operaciones del procesador el reloj necesita de 5 periodos de por reloj, por lo que 5 periodos representan un ciclo de máquina. Dicho soft-core posee los distintos elementos de un procesador tipo RISC, tales como: contador de programa (en inglés, *program counter*), registro de datos, registro de dirección, unidad aritmética lógica, acumuladores, entre otros. La principal aportación es el diseño de un *soft-core* de tipo RISC de 4-bits el cual pueda ser fácilmente escalable: 8, 16, 32, 64 o 128 bits de direccionamiento, depende de la implementación que se requiere.

Este artículo está organizado de la siguiente manera, en la sección 1 se abordan los antecedentes de procesadores *soft-core* implementados en FPGA. En la sección 2 contiene teoría relevante para el desarrollo de los objetivos de este artículo, por lo que se encuentran temas relacionados con procesadores y FPGAs. En la sección 3 se muestra la

implementación del código en VHDL. En la sección 4 se presentan los resultados. Por último, en la sección 5 se presentan las conclusiones finales.

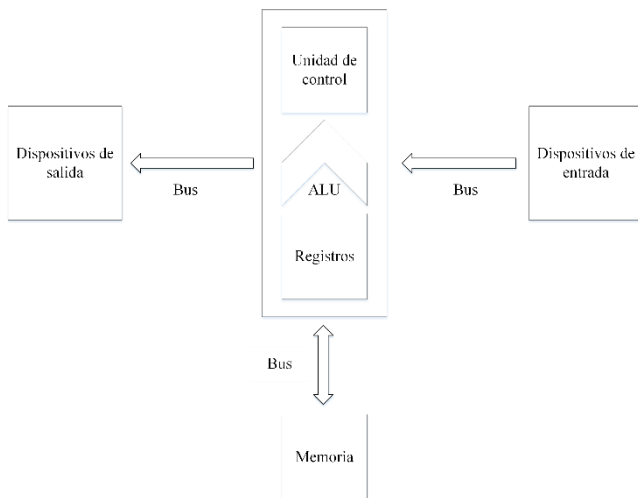
2. MARCO TEÓRICO

En esta sección se presentan los fundamentos teóricos sobre los procesadores, sus tipos de arquitectura, elementos básicos, así como fundamentos sobre las tarjetas FPGAs.

2.1. Procesadores

El procesador es el componente más importante dentro de un sistema de cómputo, es el encargado de que todo funcione correctamente en el sistema. La estructura general de sistema de cómputo, se conforma: procesador, dispositivos de entrada y salida, memoria y un bus de comunicación [3]. En la figura 1 se aprecia un diagrama general de la organización de la a bloques de una computadora.

Fig. 1. Organización general de una computadora.



Fuente: elaboración propia

Como ya se mencionó, el procesador es la parte importante de un sistema de cómputo (computadora), por ende, para su correcto funcionamiento, del sistema posee diversos elementos que se encargan de realizar sus distintas operaciones. Los elementos más básicos para un procesador son:

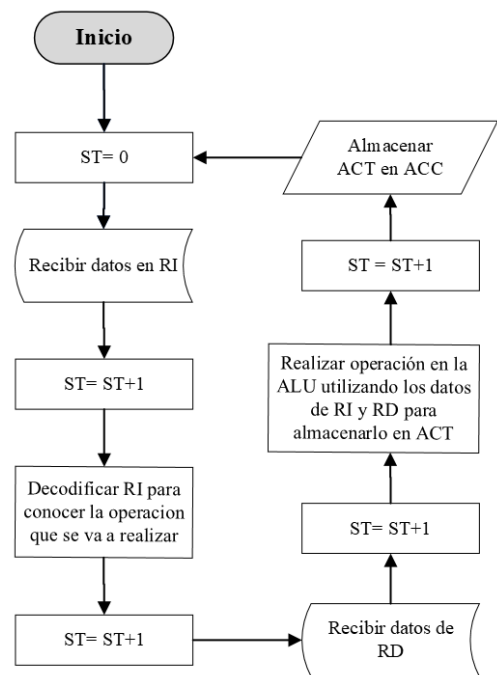
- Contador de programa (PC): este apuntador se encarga de apuntar a cada instrucción en memoria, de esta forma el procesador sabe que instrucción se está ejecutando y cuál es la siguiente.
- Registro de instrucciones (RI): En este registro se almacena la instrucción que se desea ejecutar.
- Registro de datos (RD): se encuentra el dato de entrada para realizar la operación que indique el RI.
- Unidad Lógica Aritmética (ALU): Como su nombre lo indica, es la responsable de realizar todas las operaciones lógicas o aritméticas del sistema, es decir, operaciones lógicas como *and*, *or* y *not* u

operaciones aritméticas como, suma, resta, multiplicación, entre otros.

- Acumulador temporal (ACT): Una vez la ALU realiza su operación, el resultado de dicha operación es enviado al acumulador temporal.
- Acumulador (ACC): Finalmente, el valor del ACT se envía al ACC para almacenarse en el mismo o utilizarse en futuras operaciones, dicho valor también se puede almacenar en la memoria de la computadora si así se desea, siempre y cuando dicha instrucción sea dada al procesador [3].

En la figura. 2 se observa un diagrama de flujo con el proceso al realizar instrucciones de un procesador.

Fig. 2. Diagrama de flujo del proceso.



Fuente: elaboración propia

2.2. Sistema de instrucciones entre RISC vs CISC

Antes de realizar comparaciones entre sistema de instrucciones es necesario establecer los siguientes términos:

- CISC (en inglés, *Complex Instruction set computer*): Computadora con un conjunto de instrucciones complejo.
- RISC (en inglés, *Reduced Instrucción set computer*): Computadora con un conjunto de instrucciones simple o reducido.

Tomando en cuenta lo anterior, las arquitecturas de los procesadores en este trabajo, se refiere al diseño y la estructura interna de un procesador de un sistema de cómputo en la que el procesador recibe instrucciones para realizar su ejecución de dicha instrucción

Como su nombre lo indica, el CISC está conformada por instrucciones complejas, las cuales se basan en un conjunto de instrucciones del procesador, el problema es que, algunas instrucciones son tan complejas que tardan mucho más tiempo en ejecutarse lo que puede provocar problemas en tiempo de respuesta en su ejecución o tiempos de espera.

Por otro lado, el RISC se basa en simplificar las instrucciones para aumentar la velocidad de procesamiento. Una gran ventaja al tener instrucciones simples es que la mayoría de estas se cumplen en un ciclo de reloj, por lo que es más difícil tener problemas al ejecutar varias instrucciones o tiempos de espera. De igual forma, al trabajar con instrucciones más sencillas, es posible migrar dichos modelos a tecnologías más recientes y con mayor eficiencia como lo son las tarjetas FPGAs [4].

2.3. Introducción a los FPGA

Un FPGA (*Field Programmable Gate Array*) es un dispositivo hardware el cual, como su nombre lo indica, está conformado por bloques lógicos que se pueden configurar o programar para realizar alguna tarea en específico, en su mayoría, dichos bloques lógicos poseen LUT (Tablas de búsqueda) y FF (Flip-Flops). Para realizar la programación de estos dispositivos hay dos lenguajes que dominan el mercado, Verilog y VHDL. VHDL es un lenguaje de descripción de hardware el cual se utiliza para modelar y diseñar sistemas digitales, este lenguaje permite al usuario modelar el diseño y la estructura interna de un procesador, así como diseñar su funcionamiento interno. VHDL es un lenguaje complejo y sofisticado, el cual es un estándar en la industria debido a que es un estándar oficial de la IEEE [5].

Una de las ventajas clave de VHDL es su capacidad para sintetizar programas en un archivo de tipo BIT. Esto permite la implementación directa del código en un arreglo de compuertas lógicas o bloques lógicos, lo que resulta en la obtención de resultados idénticos a los obtenidos durante la simulación cuando se utiliza un FPGA

Por último, al realizar implementaciones en una tarjeta FPGA es importante conocer las capacidades físicas de la tarjeta que se va a utilizar, pues, si el programa a ejecutar sobrepasa las capacidades del FPGA, este no podrá ser implementado y por ende no cumplirá con la tarea solicitada. En este trabajo se utilizó una tarjeta de la familia Zynq-7000, en concreto el modelo XC7Z010clg400-1. Dicha tarjeta FPGA posee una velocidad de procesamiento de 650MHz, 4400 unidades lógicas, 6 botones pulsadores, 4 interruptores, 5 leds, 6 puertos *Pmod*, entre otras especificaciones [6]. En la figura 3 se observa una imagen de la tarjeta utilizada.

Fig. 3. Tarjeta xc7z010clg400-1.



Fuente: Manual Zybo [6]

3. IMPLEMENTACIÓN EN FPGA

En esta sección se muestran las implementaciones del código tanto de la entidad como del procesador de tipo RISC en el FPGA.

3.1. Implementación de la entidad

Con base en las características de un procesado tipo RISC dadas en la sección 2 se diseñó la entidad ver figura 4.

Fig. 4. Estructura de la programación VHDL.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith;

entity MAIN is
  Port ( Elreloj : in STD_LOGIC;
        reset : in STD_LOGIC;
        RO : inout std_logic_vector (3 downto 0) ;
        RI : inout std_logic_vector (3 downto 0) ;
        ACC : inout std_logic_vector (3 downto 0) := "0101"
        );
end MAIN;
```

Fuente: elaboración propia

3.2. Implementación de la arquitectura

La implementación de la arquitectura se refiere al diseño y la estructura interna de un procesador. Define cómo se organizan los diferentes componentes del procesador, cómo se comunican entre sí y cómo procesan las instrucciones y los datos. Aquí hay algunos elementos clave que suelen estar presentes en una arquitectura de procesador en la sección 2.

Por lo que, además de poseer distintas operaciones lógicas en la ALU, es necesario establecer una secuencia para determinar en qué momento se leerán los datos, se realizaran operaciones o se enviaran los datos a la memoria. Dicha

implementación se observa en el pseudocódigo de la figura. 5.

Fig. 5. El código de soft core.

```

process(clk,reset)
variable AEBCD: std_logic_vector (4 downto 0);
variable Q: std_logic_vector (3 downto 0 ):="0000";
variable RIaux: std_logic_vector (3 downto 0);

begin
  if (RESET = '1') then
    Q := "0000" ;
  elsif (rising_edge(clk)) then
    case Q is
      when "0101" => Q:= "0000";

      when others => Q := Q+"0001";
    end case;
    AEBCD:= gcm(Q);
    A<= AEBCD(4);
    E<= AEBCD(3);
    B<= AEBCD(2);
    C<= AEBCD(1);
    D<= AEBCD(0);
    if AEBCD(4) = '1' then -- A
      RI<= reg_ins(RO);
    end if;
    if ri =ro and AEBCD(3)='1' then
      DI <= deco_ins(RI);
    end if;
    if AEBCD(2) = '1' then -- B
      RD <= reg_dat(RI);
    end if;
    if (ri = rd and AEBCD(3)='1') then
      OP <= alu(RD,ACC,DI);
    end if;
    if AEBCD(3)='1' then
      PC <= pcount(DI,PC);
    end if;
    if AEBCD(1)='1' then
      ACT <= acct(OP);
    end if;
    if AEBCD(0)='1' then
      ACC <= acct(ACT);
    end if;
    ffD <= flip_flop_D(ACT,reset);

  end if;

end process;

```

Fuente: elaboración propia

4. RESULTADOS

La sinterización el código diseñado en VHDL el sistema se tomó 1:08 minutos y requirió de 42 LUT y 32 FF, mientras

que para diseñar la implementación el sistema se tomó 1:31 minutos y requirió de 42 LUT y 36 FF. Tomando en cuenta que se requieren 6 periodos para completar un ciclo de máquina y que la tarjeta seleccionada posee una velocidad de procesamiento de 650 MHz, la tarjeta FPGA es capaz de realizar 1.5 millones de operaciones por segundo. Por último, el sistema consume 7.4 W de energía, de los cuales únicamente el 4% corresponde a la ejecución de la lógica del programa, el otro 96% corresponde al consumo energético de los dispositivos de entrada y salida.

5. CONCLUSIONES

En este trabajo de investigación, se presenta un diseño sencillo y su implementación en un FPGA un procesador *soft-core* con tipo RISC de cuatro bits que realiza siete distintas operaciones en un ciclo de máquina con 6 periodos por ciclo. Aunque, el procesador diseñado es de cuatro bits, tiene la facilidad de escalar a N cantidad de bits de información y hasta 16 operaciones distintas, siempre y cuando se respete y utilice el ciclo de máquina del RISC y sean operaciones simples.

El *soft-core* diseñado es capaz de realizar 1.5 MIPS con un consumo de sistema consume 28 mWatts de energía, el 4% corresponde a la ejecución de la lógica del programa, el otro 96% 6.72 Watts corresponde al consumo energético de los dispositivos de entrada y salida.

Discusiones

Para implementar el procesador en la tarjeta de la familia Zynq-7000 fue necesario el diseño de la entidad y de las instrucciones tipo RISC para cada una de las operaciones del procesador, de igual forma fue necesaria la asignación de puertos para lograr visualizar correctamente las entradas y salidas del procesador, todo esto fue parte del proyecto del curso de sistemas digitales en el CITEDI.

Reconocimiento:

Agradecemos las facilidades otorgadas para la realización de este trabajo al Consejo Nacional de Ciencia y Tecnología (CONAHCYT), Instituto Politécnico Nacional a través de la Secretaría de Investigación y Posgrado con los proyectos SIP 202401430, SIP 20241219. Además, a la Unidad Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas y Centro de Investigación y Desarrollo de Tecnología Digital. Asimismo, el "Programa de estímulos al desempeño de la investigación" (EDI), el "Programa de estímulos al desempeño docente" (EDD) y la "Comisión de Operación y promoción de actividades académicas" (COFAA).

6. REFERENCIAS

- [1] D. Lorente, "Implementación de un procesador digital de audio basado en FPGA", Tesis de maestría,

Universidad Politécnica de Valencia, Gandía, España, 2013.

- [2] F. Olivero, F. Iturriaga, C. Almaraz, “Procesador soft-core de una única instrucción” *Research in Computing Science*, 137(1), pp. 9-17, 2017.
- [3] A. Tanenbaum, “Organización de computadoras: un enfoque estructurado”, 4ta edición, Prentice Hall, 2000.
- [4] J. I. Vega-Luna, R. Sánchez-González, G. Salgado-Guzmán, L. A. Sánchez-González, “Arquitectura RISC vs CISC”, *PC/TIPS BYTE*, 5(50), pp. 1-9, 1992.
- [5] S. Brown, Z. Vranesic, “Digital Logic with VHDL Design” 3ra edición, Mc Graw Hill, 2009.
- [6] Digilent, “ZYBO™ FPGA Board Reference Manual”, USA, 2017.