

Laboratorio de Electrónica Digital II

Práctica No. 5

Calculadora RPN en Procesador LC3

Profesor Lab.: Luis Germán García M. (german.garcia@udea.edu.co)

Septiembre 15, 2019



Fecha de entrega: Octubre 9 de 2019
Medio de entrega: <http://www.ingeniaudea.co>
Sustentación: Octubre 10-11 de 2019.
Valor Práctica: 9% del curso (22,5% laboratorio)

1 Objetivo de la Práctica

Diseñar una calculadora básica con notación polaca inversa (RPN - Reverse Polish Notation) empleando lenguaje ensamblador para el procesador LC3 implementado en FPGA.

2 Objetivos Específicos

- (i) Establecer los periféricos de hardware para el procesador LC3 que realicen las operaciones de multiplicación y división, junto con las respectivas subrutinas en lenguaje ensamblador.
- (ii) Definir el sistema de almacenamiento de operandos en memoria para hacer un uso eficiente de la notación polaca inversa (RPN).

- (iii) Describir la funcionalidad de las operaciones suma, resta, multiplicación, división y módulo a partir de instrucciones de bajo nivel para el procesador LC3.
- (iv) Demostrar el correcto funcionamiento de la calculadora RPN programada para el procesador LC3, el cual es implementado en una FPGA.

3 Calculadora RPN en Procesador LC3

La calculadora RPN debe estar en capacidad de realizar cualquiera de las operaciones matemáticas básicas tales como suma (+), resta (-), multiplicación (\times), división (\div) y módulo (%). Una terminal de comunicación serial (e.j. CuteCom) servirá de interfaz entre el usuario y la calculadora. Todas las operaciones a realizar sobre un conjunto de operandos se deben indicar en notación polaca inversa. En RPN, las operaciones se indican después de proveer los operandos. Por ejemplo, la operación algebraica $(2 + 5) \times 3$ se representa en RPN como: $2\ 5 + 3 \times$, donde cada operación toma dos operandos únicamente. También sería posible expresar la operación mediante: $3\ 2\ 5 + \times$. Note que la operación matemática toma los dos operados más cercanos a ella. Para facilitar la funcionalidad RPN, los operandos y resultados de cada operación se almacenan en un *stack*, el cual corresponde a una estructura de datos en memoria. Cualquier operación de +, -, \times , \div o %, opera sobre los dos últimos operandos almacenados en el *stack*, a la vez que los elimina de allí. El resultado de la operación se almacena como un nuevo dato en el *stack*.

Para la operación de la calculadora, ésta debe ofrecer un menú con las siguientes opciones:

- (1) Insertar operandos al *stack*.
- (2) Ver valores en el *stack*.
- (3) Realizar operaciones.
- (4) Limpiar el *stack*.

La opción 1 le permite al usuario ingresar valores en el *stack* que serán utilizados más adelante para llevar a cabo las operaciones. Los operandos se deben ingresar separados por comas (,). El caracter ENTER (LF) debe ir después del último operando. Por ejemplo, para insertar 4 operandos en el *stack*, utilice la siguiente sintaxis: "1025,32767,15,-18,LF". La opción 2 imprime en la terminal los valores actualmente almacenados en el *stack*, incluyendo el resultado de la última operación. La opción 3 permite indicar las operaciones a realizar mediante los símbolos: + para suma, - para resta, * para multiplicación, / para división y % para el módulo, separados por comas (,). El caracter ENTER (LF) debe ir después de la última operación. Por ejemplo, para realizar las operaciones con los operandos dados previamente: $15 \times (-18) + 32767 \div 1025$, utilice la siguiente sintaxis: "*+,/LF". Cuando la calculadora recibe el caracter LF, ésta realiza todas las operaciones solicitadas y muestra el resultado final. Finalmente, la opción 4 elimina todos los valores presentes en el *stack*. Una vez se procese la opción solicitada (1-4), la calculadora debe desplegar nuevamente el menú ya descrito.

4 Procedimiento

Tenga en cuenta los siguientes pasos para el correcto diseño de su calculadora:

- a. Descargar de la página del curso el código VHDL dado para el procesador LC3, el módulo de comunicación serial, el módulo de lectura/escritura de la memoria y otros módulos, agregarlos a un proyecto de Xilinx ISE, comprender su funcionamiento, sintetizarlos y programar la FPGA.
- b. Describir los periféricos (hardware) para la multiplicación y división empleando lenguaje de descripción de hardware VHDL y comprobar su funcionamiento. Llevar a cabo las correcciones pertinentes.
- c. Mapear los periféricos de multiplicación y división en la memoria del procesador LC3 a través del módulo *Peripheral*, disponible en el código VHDL dado en la página WEB del curso.
- d. Desarrollar subrutinas en ensamblador para el acceso a los periféricos de multiplicación y división a través de direcciones de memoria y comprobar su funcionamiento. Llevar a cabo las correcciones pertinentes.
- e. Comprender la notación polaca inversa y el concepto de *stack* para aplicarlos a su diseño.
- f. Desarrollar el código ensamblador encargado de manejar la RPN y el *stack*.
- g. Componer el código de la calculadora empleando los códigos previamente desarrollados para establecer la funcionalidad indicada, ensamblarlo, y comprobar su funcionamiento mediante el uso del simulador. Llevar a cabo las correcciones pertinentes.
- h. Realizar los ajustes finales al código de la calculadora para su funcionamiento en el procesador LC3 implementado en la FPGA, ensamblarlo, descargarlo en la memoria del sistema Nexys-2 y comprobar su funcionamiento. Llevar a cabo las correcciones pertinentes.
- i. Enviar el código fuente de su programa (ASM, HEX, etc.) en lenguaje ensamblador para el procesador LC3, el código fuente del hardware (VHDL), junto con un reporte, antes de la fecha límite.
- j. Sustentar el diseño en el horario de laboratorio correspondiente.

5 Consideraciones

Tenga en cuenta las siguientes consideraciones:

- * Los operandos son valores enteros con signo de 16-bits.
- * La unidad de multiplicación debe recibir dos operandos de 16-bits (valores enteros) y retornar un resultado de 32-bits.
- * La unidad de división debe recibir dos operandos de 16-bits (valores enteros) y retornar dos resultados de 16-bits, el cociente y el residuo (requerido para el módulo).
- * Las operaciones se realizan en 16-bits.
- * Todos los operandos se ingresan/imprimen en formato decimal. Formato hexadecimal no está permitido.

- * Valores negativos se ingresan/muestran en formato -dddddd (ej. -32768). Sin embargo, en la memoria del procesador LC3 deben ser almacenados en complemento a dos.
- * Realice sub-rutinas en su programa para hacer el código modular.
- * Evite el uso de multiplicadores/divisores provistos por las librerías de VHDL. En su lugar, diseñe las unidades de multiplicación y división empleando técnicas conocidas.
- * El estudiante puede emplear el compilador C para el procesador LC3 si lo considera pertinente.
- * El profesor podrá solicitar cambios al diseño durante las diferentes sesiones del laboratorio sin que queden consignadas en esta guía.

6 Entrega

El grupo de trabajo deberá escribir un reporte (4-5 páginas) que contenga los siguientes elementos:

- a. **Abstract:** resumen del diseño del programa en ensamblador y periféricos para el procesador LC3.
- b. **Implementación:** Diagrama de bloques de las unidades de multiplicación y división. También se debe indicar el pseudo-código para las subrutinas que acceden a los periféricos de multiplicación/división, para el código que maneja el *stack* y para el código que implementa la ejecución de las operaciones usando RPN.
- c. **Conclusiones**

Crear un archivo comprimido que incluya el reporte y los archivos importantes de su programa en ensamblador y proyecto en Xilinx ISE como se describe a continuación:

- a. Reporte: archivo con extensión .pdf
- b. Archivos en ensamblador: *.asm, *.txt, *.hex, etc.
- c. Archivos de su proyecto en Xilinx: *.xise, *.vhd, *.bit, *.ucf, etc.

El nombre del archivo comprimido debe tener el siguiente formato:

p5_primerapellidointegrante1_primerapellidointegrante2_horariolaboratorio.zip.

Ejemplo: si el primer apellido de ambos integrantes es **Cabarcas** y **García**, respectivamente, y el laboratorio es el Martes de 9-12, entonces el archivo debe ser nombrado: *p5_cabarcas_garcia_m9-12.zip*.

7 Evaluación

La evaluación de la práctica se divide en tres partes, funcionamiento (55%), sustentación (40%) y reporte (5%). Las notas para el funcionamiento y el reporte se asignan por igual a todos los integrantes del grupo de trabajo, mientras que la nota de sustentación es individual. En caso un estudiante obtenga una nota inferior a 3.0 en la sustentación, su nota final de la práctica será la

que obtuvo en la sustentación, es decir, no se tendrá en cuenta el funcionamiento ni el reporte en el cálculo.

Cada grupo de trabajo deberá sustentar la práctica en un tiempo de 15 minutos, 7 minutos para mostrar el funcionamiento y 8 minutos para sustentar el diseño. Durante la sustentación, el profesor hará entre dos (2) y cuatro (4) preguntas a cada uno de los integrantes del grupo de trabajo. Entre las preguntas que se podrían hacer estarían las siguientes:

- a. Indicar ventajas y desventajas de la notación polaca inversa (RPN).
- b. Explicar el manejo del *stack* en memoria.
- c. Explicar el diseño de los periféricos de multiplicación y división.
- d. Indicar la forma como mapeó los diferentes periféricos a la memoria del procesador LC3.
- e. Explicar cualquier parte del código ensamblador.
- f. Realizar cambios al código ensamblador o VHDL.

8 Referencias

- a. Introduction to Computing Systems, Yale N. Patt, Sanjay J. Patel.
Diapositivas sobre la arquitectura LC3.
https://highered.mheducation.com/sites/0072467509/student_view0/powerpoints_nc_state.html
Visitado: Agosto, 2019.
- b. Ensamblador y Simulador para LC3 (Windows/Linux)
https://highered.mheducation.com/sites/0072467509/student_view0/lc-3_simulator.html
Visitado: Agosto, 2019.
- c. The LC-3, University of Texas at Austin
http://www.cs.utexas.edu/users/fussell/cs310h/lectures/Lecture_10-310h.pdf
Visitado: Agosto, 2019.
- d. Manual Sistema de Desarrollo Nexys 2.
<https://reference.digilentinc.com/reference/programmable-logic/nexys-2/reference-manual>
Visitado: Agosto, 2019.