



INFORME DE LABORATORIO 5

Autores: *María Del Mar Arbeláez Sandoval, Julián Mauricio Sánchez Ceballos*

*Laboratorio de Electrónica Digital 2
Departamento de Ingeniería Electrónica y de Telecomunicaciones
Universidad de Antioquia*

Resumen

El trabajo de esta practica consiste en la implementación de un procesador ARM monociclo, con un conjunto de instrucciones básicas como lo son: ADD, SUB, Bxx, LDR y STR, dentro de los objetivos de la practica se plantea la ampliación del conjunto de instrucciones buscando implementar las instrucciones de desplazamiento (LSL, LSR, ASR, ROR y MOV) y una instrucción de comparación CMP. Luego de la implementación de estas nuevas instrucciones se plantea el uso de este procesador para realizar la operación $A+B$, donde las entradas son obtenidas mediante unos switches integrados en la fpga y previamente mapeados en la memoria de datos y estos datos deben ser mostrados en los displays de 7 segmentos también previamente mapeados en la memoria de datos.

Palabras clave: ARM, ensamblador, Registros, Memoria, Instrucciones, periféricos.

INTRODUCCIÓN.

En esta practica de laboratorio se implementa un versión del procesador ARM con ejecución de instrucciones en un solo ciclo, junto con un programa escrito en ensamblador que pueda ejecutarse en el mencionado procesador haciendo uso de los periféricos del sistema. La descripción de hardware, implementación del procesador y las nuevas instrucciones, será realizada mediante System Verilog, el despliegue del procesador se realiza en el sistema de desarrollo FPGA DE10-LITE.

IMPLEMENTACIÓN DE LAS NUEVAS INSTRUCCIONES.

Para la implementación de las nuevas instrucciones (LSL, LSR, ASR, ROR, MOV y CMP) se parte del siguiente diagrama:

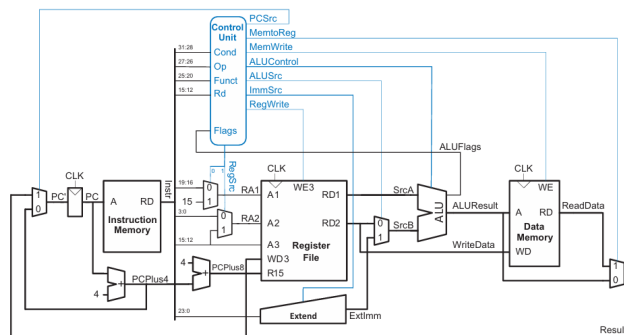


Figura 0-1: Procesador monociclo.

En primer lugar se implementan las instrucciones de desplazamiento, para esto, es necesario introducir un modulo que realice estas operaciones, la ubicación del modulo es entre la RD2 del Register File y el MUX que escoge entre si la operación es con un inmediato o con un registro. las entradas de este modulo será las RD2 Y INSTR[11:5], esta última contiene los argumentos necesarios para realizar la tarea que se requiere, tendrá una sola salida la cual se conecta a la primera entrada del MUX. El nuevo diagrama del procesador se incluye a continuación:

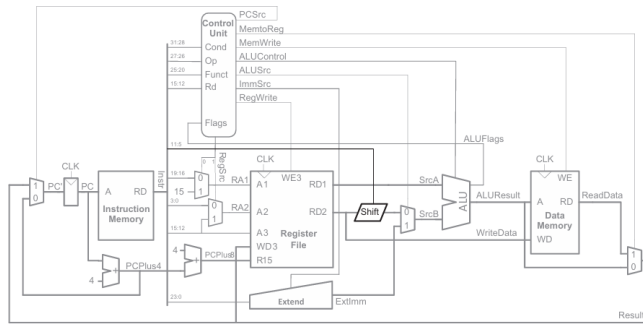


Figura 0-2: Introducción del modulo shift.

El modulo shift realiza las siguientes operaciones:

```

1 module shift (
2   input logic [31:0] entrada,
3   input logic [6:0] instr,
4   output logic [31:0] b
5 );
6
7   logic [1:0] shtype;
8   assign shtype = instr [1:0];
9
10  always_comb begin
11    case (shtype)
12      2'b00: b = entrada << instr[6:2]; //LSL
13      2'b01: b = entrada >> instr[6:2]; //LSR
14      2'b10: b = entrada >> instr[6:2]; //ASR
15      2'b11: b = (entrada >> instr[6:2]) | (entrada << (32-instr[6:2])); //ROR
16    endcase
17  end
18 endmodule

```

Figura 0-3: descripción de hardware del modulo shift.

Con esto, además de la ampliación de la señal ALU-Control de 2 bits a tres bits (Una más para que se pueda implementar un bypass en la ALU) estaría implementadas las instrucciones de desplazamiento.

Para la instrucción CMP, es necesario alterar la unidad de control del procesador. En un inicio, la unidad de control está compuesta por dos bloques, un decodificador y un lógico condicional, como se muestra a continuación:

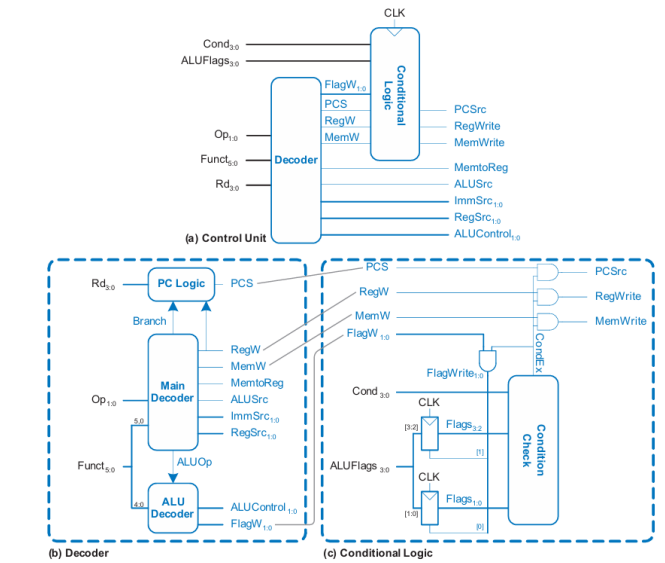


Figura 0-4: Unidad de control para el procesador monociclo

La instrucción CMP, resta SrcB de SrcA y activa las banderas de la ALU, esta resta no se escribe en Rd, por lo que la modificación a la unidad de control será agregar una señal que evite escribir en Rd, durante la ejecución de la instrucción CMP. La conexión de la señal se realiza de la siguiente manera:

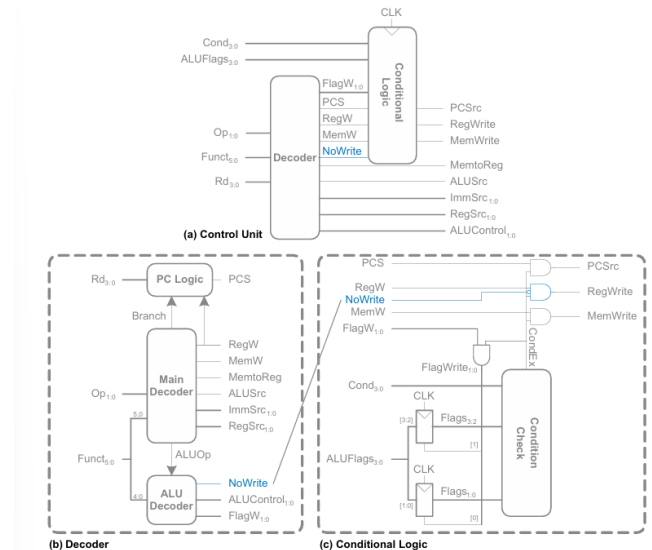


Figura 0-5: Unidad de control modificada para la instrucción CMP

Esta señal es creada en los archivos correspondientes

SIMULACIÓN.

den entradas e interacción con el usuario durante tiempo de ejecución, la simulación brinda información sobre el apartado inicial de la practica (entrada del dato A y su correspondiente representación en el display de 7 segmentos). Esto es lo que se muestra en el imagen anterior.

Los debidos archivos como la memoria de instrucciones, la memoria de datos y el programa en ensamblador se adjuntan debidamente junto a la entrega de este informe.

CONCLUSIONES.

- El procesador ARM aunque inusualmente usado en la industria, tiene grandes potenciales académicos en forma de demostración de las múltiples formas y estrategias por las cuales un procesador puede ser mejorado mediante la inclusión de nuevas instrucciones, como se pudo ver en el desarrollo de este laboratorio.
- Mediante de esta practica es posible comprender las limitaciones, características y ventajas que se tiene en un conjunto de instrucciones sobre una versión con más ó mejores versiones de estas mismas instrucciones, el lenguaje ensamblador permite un conocimiento intimo y estrecho con la arquitectura de un procesador (Aunque en principio este lenguaje sea más complejo de entender e implementar), a medida de que el lenguaje es complementado con más instrucciones se convierte en un lenguaje con mas accesibilidad a las personas y con un carácter tendiente a los lenguajes de mayor nivel.

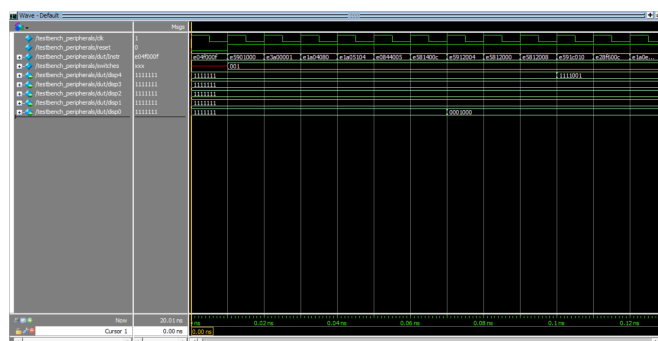


Figura 0-6: Simulación del procesador.

Durante la simulación es necesario tener en cuenta, que debido a la naturaleza del programa, donde se pi-