

**Laboratorio 1**  
**Electrónica Digital II**

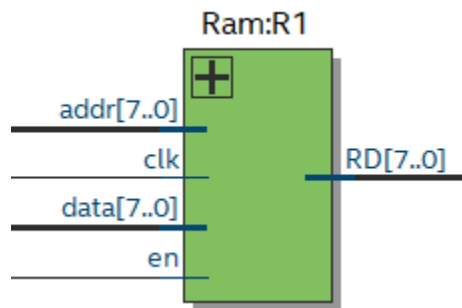
**Alexander Passo**  
**Víctor Manuel Jiménez García**

**Jhon Bairon**  
**Docente**

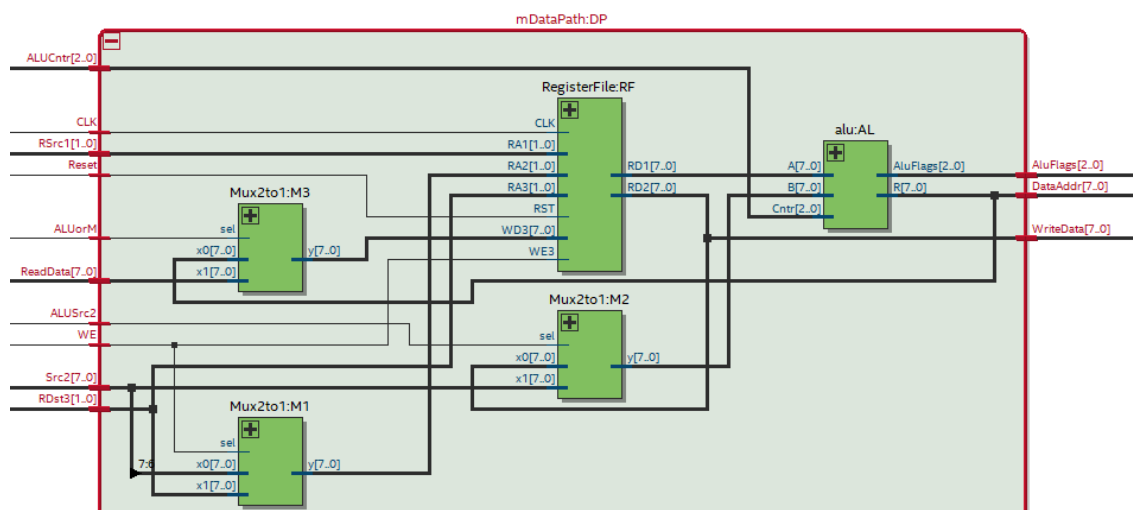


**Universidad de Antioquia**  
**Departamento de Ingeniería electrónica y telecomunicaciones**  
**Ingeniería electrónica**  
**Medellín**  
**2022**

Para el diseño del datapath de la CPU, se tuvo en cuenta cada uno de los módulos que ésta tiene internamente: Alu, multiplexores, memoria RAM y registros de 8 bits. A continuación, se mostrarán los módulos construidos en el System Verilog.



**Figura 1 Memoria RAM**



**Figura 2 Datapath**

En los archivos adjuntos esta la implementación de cada uno de los módulos, en el cual, se llaman de la siguiente forma:

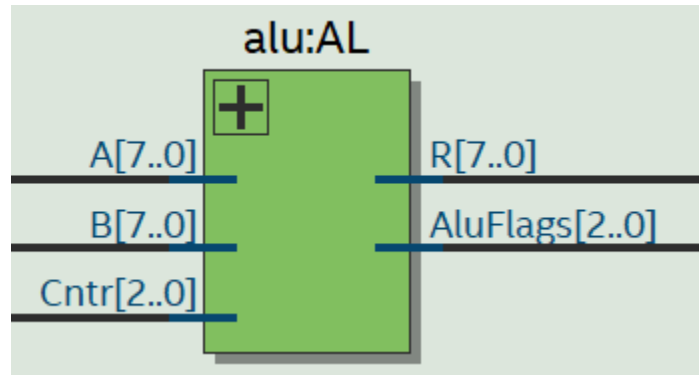
Memoria Ram: Ram.sv

ALU: alu.sv

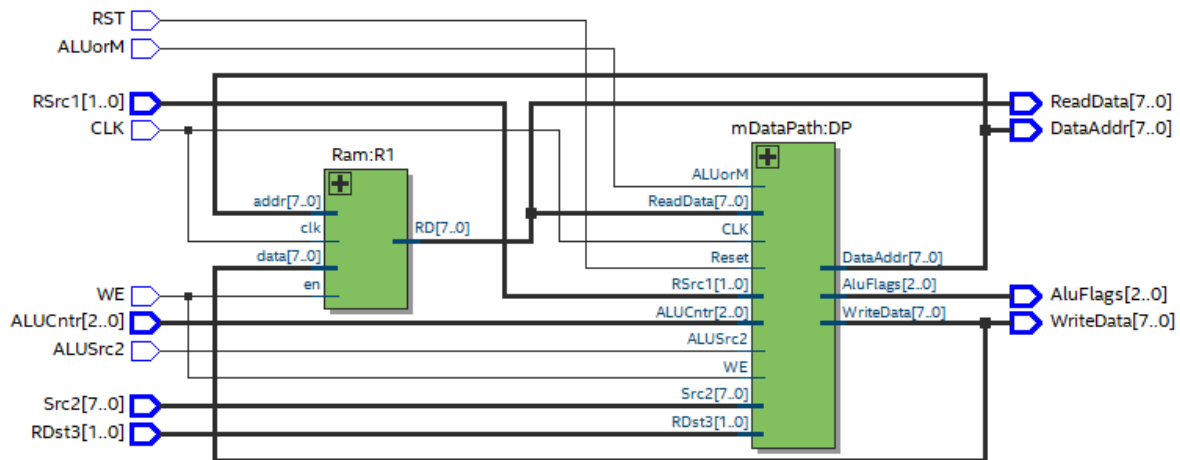
Registros: RegisterFile.sv

Multiplexor 2 a 1: Mux2to1.sv

Sistema Completo: DP\_DM\_AV.sv



**Figura 3 ALU**

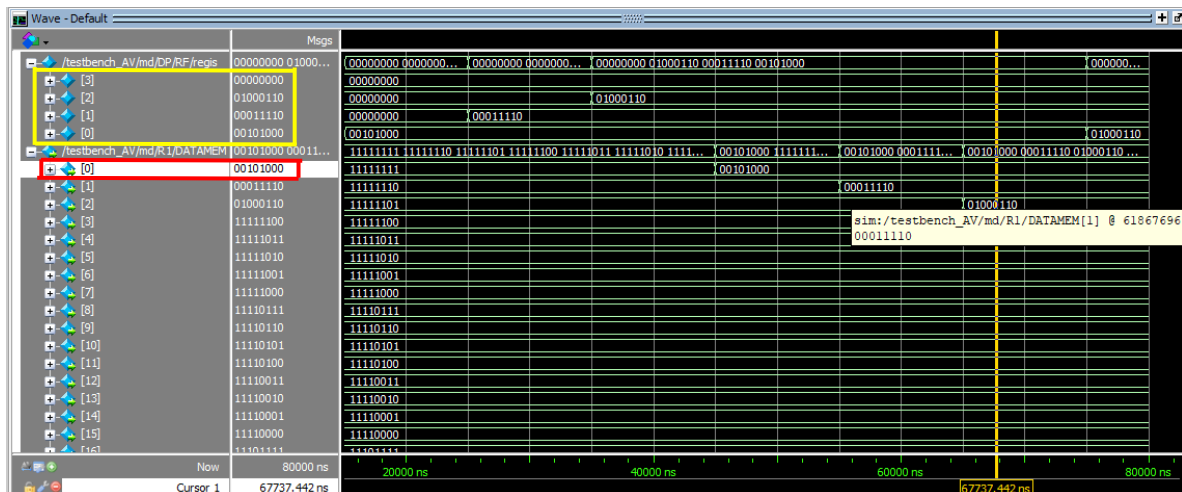


**Figura 4 Sistema Completo**

Para la prueba del sistema completo, se procedió a registrar valores en los registros, en donde se utilizan para realizar la operación suma a partir de la ALU, y donde el resultado se registra en uno de los registros. Luego, para verificar la conexión entre la memoria y los registros, se procedió a realizar un LDR. Los datos utilizados para la simulación se muestran a continuación:

```
ALUCntr = 3'b101; ALUorM = 1'b0; RDst3 = 2'b00; WE=0; RSrc1 = 2'b00; ALUSrc2= 1'b1; Src2 = 8'b00101000; #clock_cycle;
ALUCntr = 3'b101; ALUorM = 1'b0; RDst3 = 2'b01; WE=0; RSrc1 = 2'b01; ALUSrc2= 1'b1; Src2 = 8'b00011110; #clock_cycle;
ALUCntr = 3'b011; ALUorM = 1'b0; RDst3 = 2'b10; WE=0; RSrc1 = 2'b00; ALUSrc2= 1'b0; Src2 = 8'b01000000; #clock_cycle;
//Escribiendo en memoria
ALUCntr = 3'b111; ALUorM = 1'b1; RDst3 = 2'b00; WE=1; RSrc1 = 2'b01; ALUSrc2= 1'b1; Src2 = 8'b00000000; #clock_cycle;
ALUCntr = 3'b101; ALUorM = 1'b1; RDst3 = 2'b01; WE=1; RSrc1 = 2'b01; ALUSrc2= 1'b1; Src2 = 8'b00000001; #clock_cycle;
ALUCntr = 3'b101; ALUorM = 1'b1; RDst3 = 2'b10; WE=1; RSrc1 = 2'b01; ALUSrc2= 1'b1; Src2 = 8'b00000010; #clock_cycle;
//LOAD
ALUCntr = 3'b011; ALUorM = 1'b1; RDst3 = 2'b00; WE=0; RSrc1 = 2'b11; ALUSrc2= 1'b1; Src2 = 8'b00000010; #clock_cycle;
```

**Figura 5 Datos para la simulación. Testbench**



**Figura 6 Resultado de la simulación.**

Se puede ver en la figura 6, que el resultado obtenido era el esperado, al analizar que el dato que se encuentra en el registro cero, se encuentra almacenado en la posición cero de la memoria RAM.

En el testbench (modulo incluido en DP\_DM\_AV.sv) realizado para esta práctica, se encuentra más datos de prueba, sin embargo, por comodidad se muestran solo algunos ciclos de reloj donde se puedan observar el buen funcionamiento del sistema.

## Conclusiones

- Con esta práctica, se logró en primer lugar un acercamiento con el lenguaje de descripción de hardware system Verilog, el cual permitió la implementación del datapath de la CPU y su respectiva simulación.
- Se logró entender el funcionamiento de los registros a partir de flip flop, a pesar de que se implementó a partir de una función específica de lenguaje de descripción.
- Se comprobó que la memoria RAM es un hardware que puede escribir o leer datos según una dirección de memoria, y que puede estar relacionada directamente con unos registros.
- A partir de la simulación, se pudo ver y entender como varían las señales de entrada y salida, teniendo en cuenta el reloj de sincronización, que para este caso fue de 1Khz. A partir de ello, se logró hacer análisis para comprobar los resultados, en donde se pudo confirmar que fueron los esperados.