



Microtecnología y
Sistemas Embebidos

Instituto Politécnico Nacional

Centro de Investigación en Computación

Lenguajes de descripción de hardware

Práctica 8 - In-System Memory Content Editor

PROFESOR:

M. EN C. OSVALDO ESPINOSA SOSA

POR:

ING. RICARDO ALDAIR TIRADO TORRES

CIUDAD DE MÉXICO, 18 DE JUNIO DE 2024

Tabla de contenido

1. Objetivos	2
2. Manejo de una memoria de 32 localidades de 8 bits	3
3. Conclusiones	9
4. Anexos	10
4.1. Descripciones del hardware	10
4.2. Bancos de pruebas (<i>test benches</i>)	10

1. Objetivos

- Implementar una memoria de 32 localidades de 8 bits utilizando una instancia de la herramienta *IP Catalog*.
- Cargar la memoria en un dispositivo FPGA DE2-115 y manipular el contenido con la herramienta *In-System Memory Content Editor*.

2. Manejo de una memoria de 32 localidades de 8 bits

Actividad 1

Crear una memoria inicializada de 32x8 con archivo MIF como se vio en clase, habilitar la opción de uso del *In-System Memory Content Editor*. Automatizar el despliegue del contenido de las 32 localidades de memoria, usando un contador a una razón de localidad por segundo. Usar la tarjeta DE2-115.

La visualización RTL de la memoria de 32 localidades de 8 bits se muestra en la Figura 1. De manera interna, la implementación en hardware utiliza una instancia de memoria perteneciente a la herramienta de *IP Catalog* (ver Figura 2). Para inicializar la memoria, se generó un archivo *.mif* cuyos valores se presentan en la Figura 3, siendo estos el valor de la dirección incrementado en uno.

La simulación se visualiza en la Figura 4. Por cada ciclo de reloj se incrementa el valor de la dirección en uno y el valor almacenado en la localidad actual se ve reflejado en la salida Q. Una observación importante es que la señal WE esta en bajo, por lo que la memoria esta en modo lectura, siendo los datos de la memoria los observados en la Figura 5.

Con respecto a la implementación en la tarjeta DE2-115, se utilizó la herramienta *In-System Memory Content Editor* para cargar la memoria de datos al FPGA, como se visualiza en la Figura 6. En la Figura 7 se muestra que en un inicio se desconocen los valores almacenados en cada localidad, por lo que se debe hacer un chequeo utilizando la herramienta (ver Figura 8). Realizando un monitoreo continuo, como el de la Figura 9, se pueden ver los cambios realizados en la memoria en tiempo real.

En los Anexos se localiza la descripción de la memoria de 32 localidades de 8 bits. Al emplear un elemento proveniente de la herramienta de *IP Catalog*, el código se reduce a describir las entradas y salidas y llamar a la instancia dentro del módulo principal.

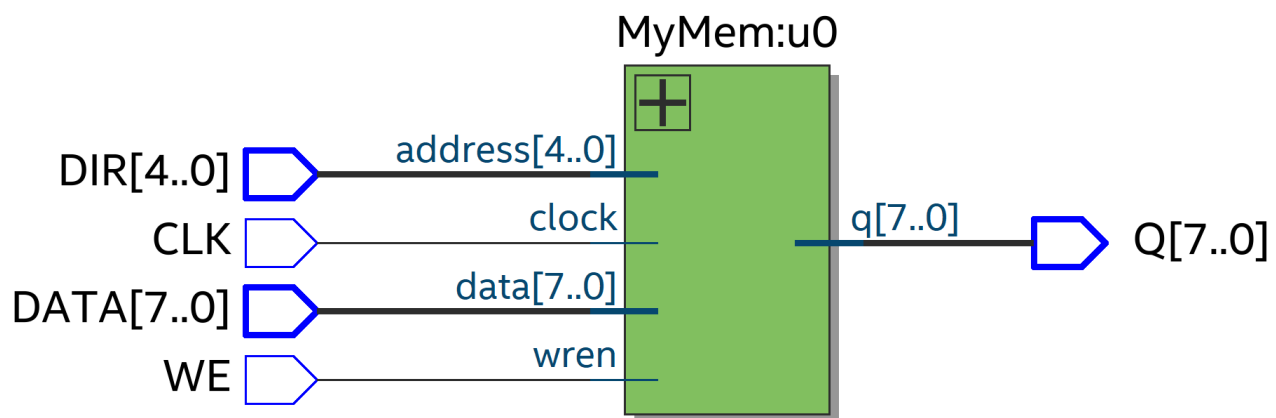


Figura 1: Diagrama RTL de la memoria de 32 localidades de 8 bits, instanciada con un componente de IP Catalog.

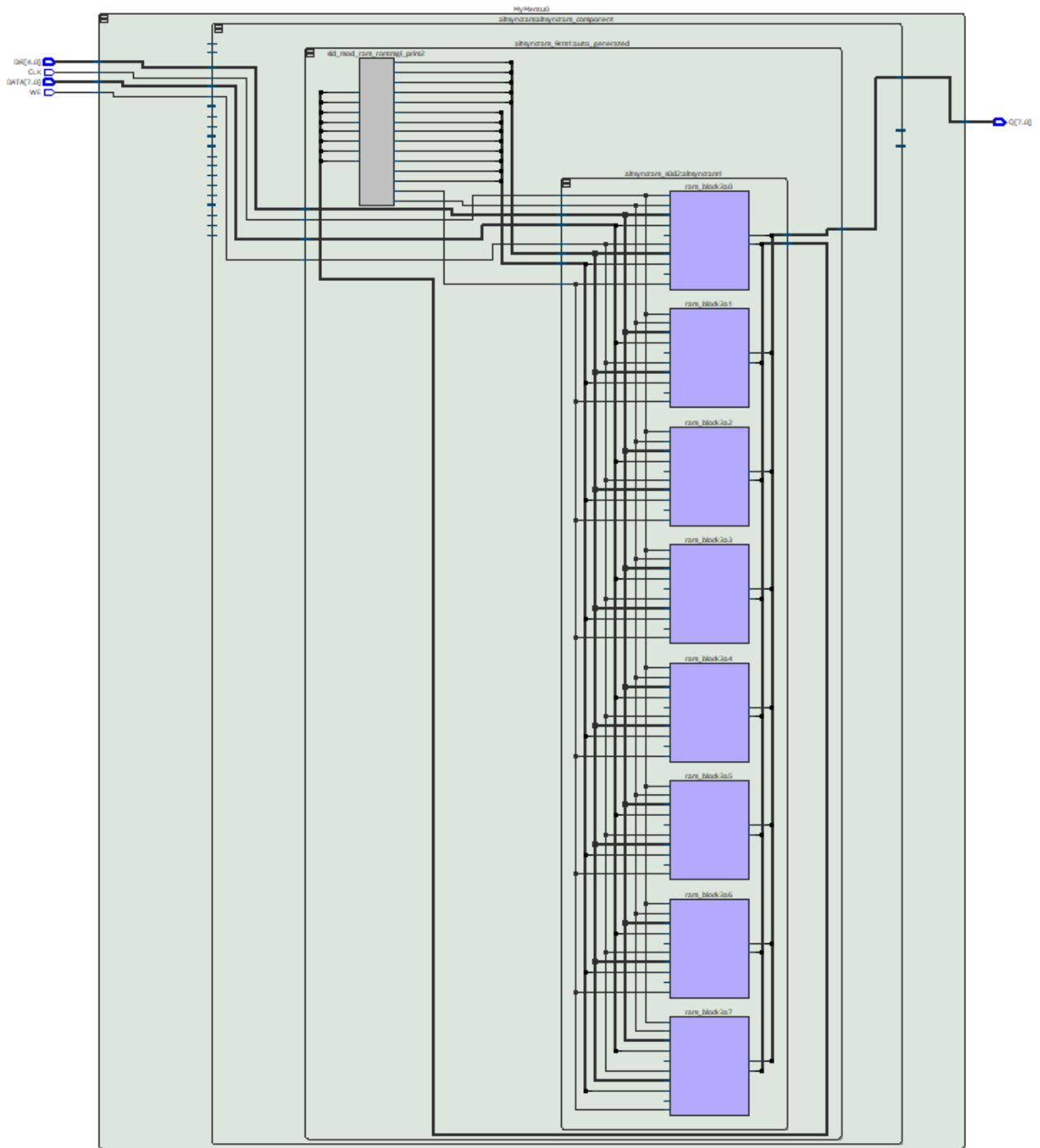


Figura 2: Diagrama RTL de la memoria de 32 localidades de 8 bits, instanciada con un componente de IP Catalog (vista interna).

Initialization.mif									
Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	01	02	03	04	05	06	07	08
8	09	10	11	12	13	14	15	16
16	17	18	19	20	21	22	23	24
24	25	26	27	28	29	30	31	32

Figura 3: Visualización del archivo de inicialización de la memoria de 32 localidades de 8 bits.

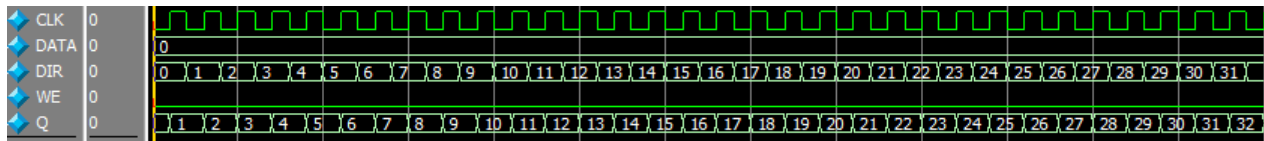


Figura 4: Simulación de la memoria de 32 localidades de 8 bits, en el visor de formas de onda de ModelSim.

Memory List				am_component/m_default/altsyncram_inst/mem_data				
Instance	Range	Depth	Width					
/Mem_vlg_tst...	[0:31]	32	8	00000000	00000001	00000010	00000011	00000100
				00000004	00000101	00000110	00000111	00001000
				00000008	00001001	00001010	00001011	00001100
				0000000c	00001101	00001110	00001111	00010000
				00000010	00010001	00010010	00010011	00010100
				00000014	00010101	00010110	00010111	00011000
				00000018	00011001	00011010	00011011	00011100
				0000001c	00011101	00011110	00011111	00100000

Figura 5: Visualización de los datos de la memoria de 32 localidades de 8 bits, en la lista de memoria de ModelSim.

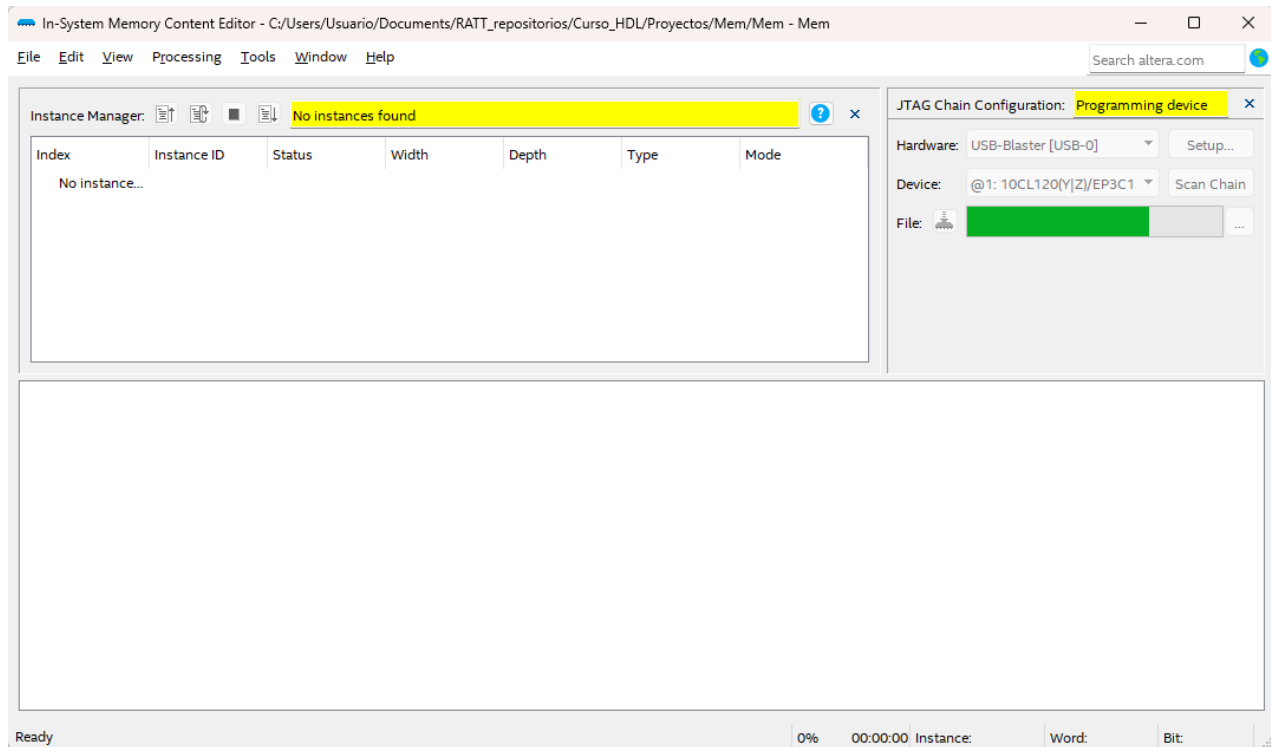


Figura 6: Uso de la herramienta *In-System Memory Content Editor* para cargar la memoria al dispositivo DE2-115 (carga en progreso).

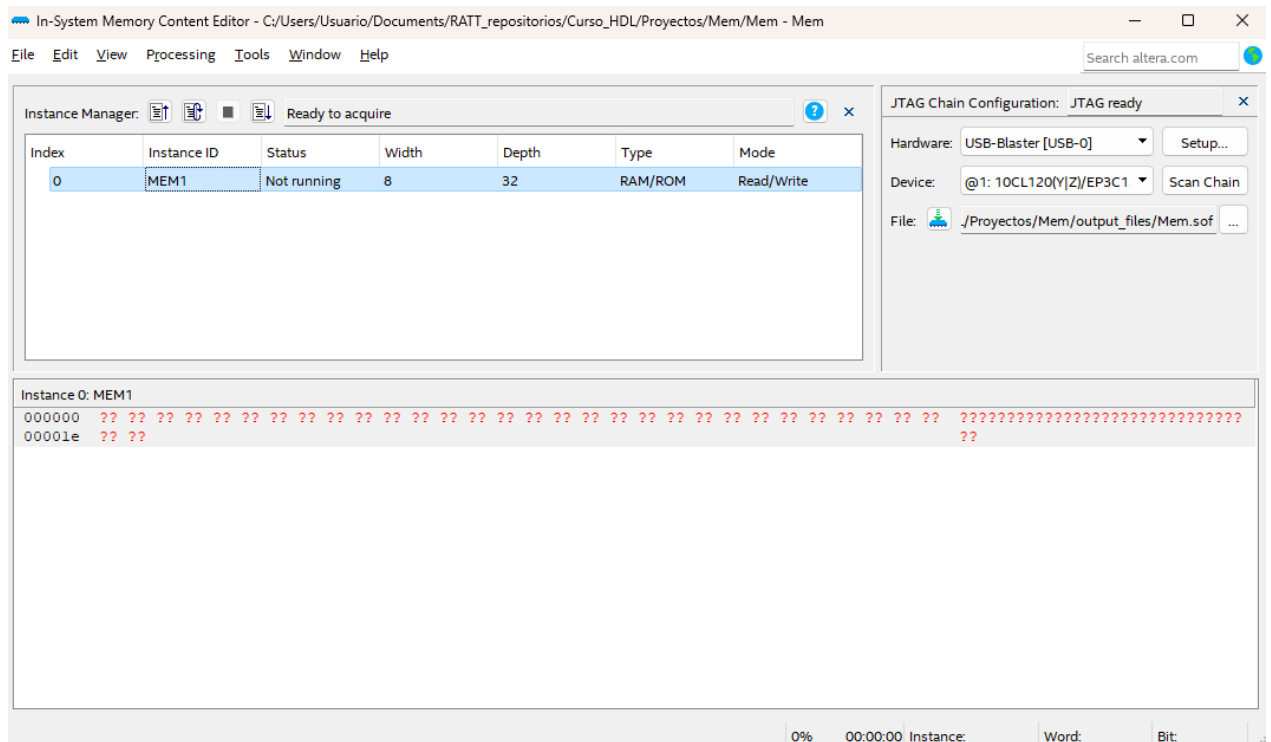


Figura 7: Uso de la herramienta *In-System Memory Content Editor* para cargar la memoria al dispositivo DE2-115 (memoria cargada).

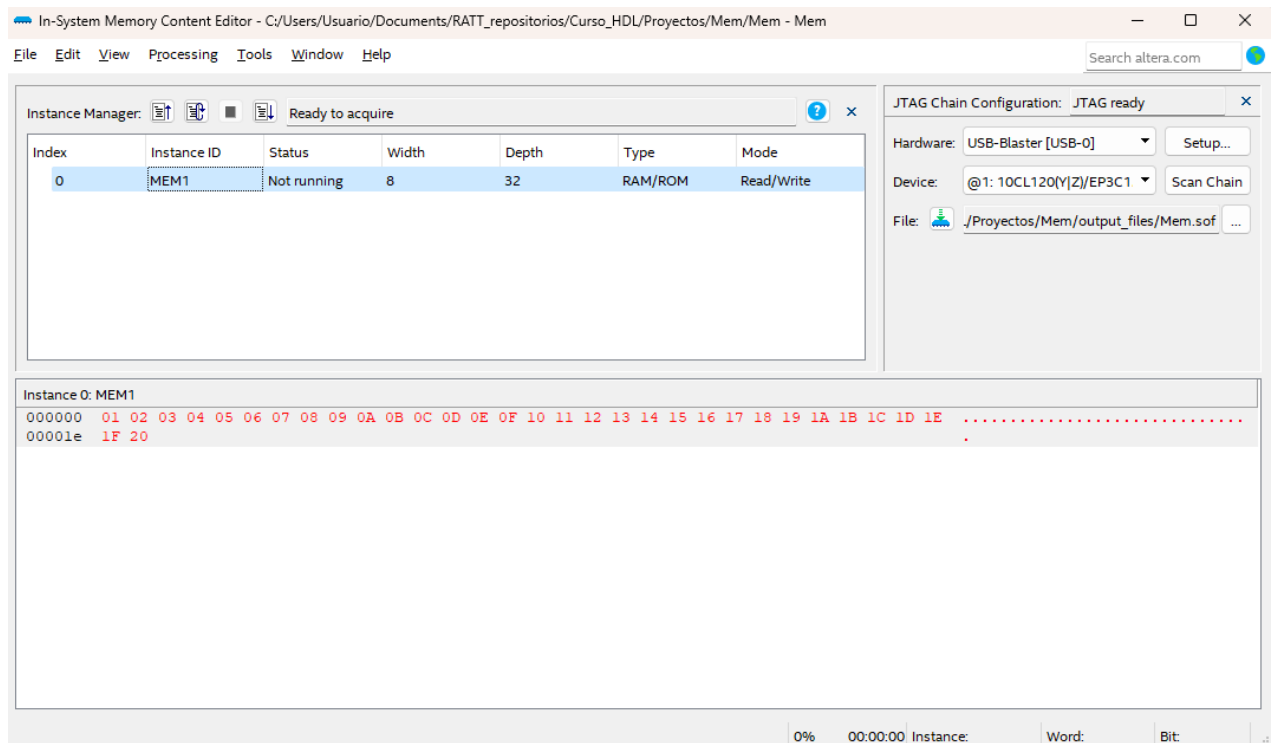


Figura 8: Uso de la herramienta *In-System Memory Content Editor* para observar los datos de la memoria en el dispositivo DE2-115 (monitoreo simple).

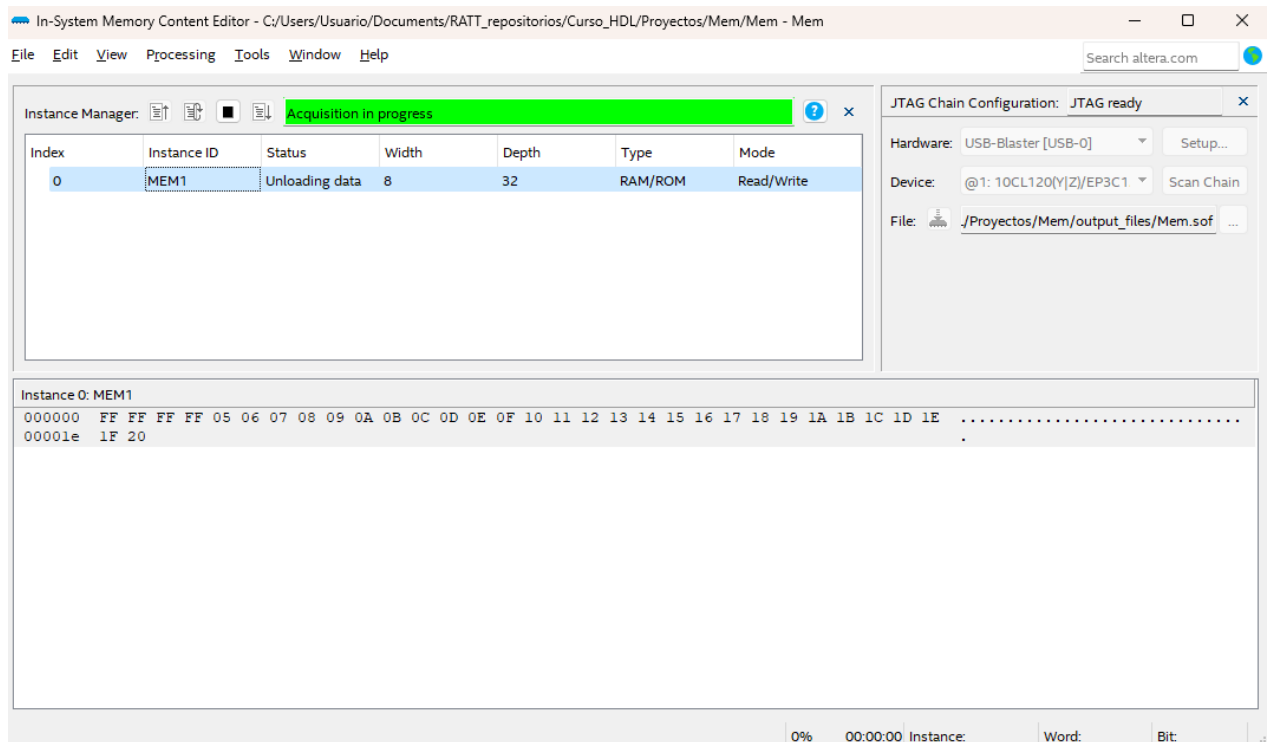


Figura 9: Uso de la herramienta *In-System Memory Content Editor* para observar los datos de la memoria en el dispositivo DE2-115 (monitoreo continuo).

3. Conclusiones

En conclusión, se implementó la memoria dentro del dispositivo de hardware de manera exitosa.

Utilizando el procedimiento de la presentación vista en clase, se creó una memoria de 32 direcciones con un tamaño de 8 bits cada una, empleando una instancia de memoria perteneciente al apartado de *IP Catalog*. Con un archivo *.mif* se inicializó la memoria.

Se realizaron las simulaciones pertinentes para observar los datos de la memoria con el visor de formas de onda de ModelSim.

Con el *Pin Planner* se asignaron los puertos de entradas y salidas y con *In-System Memory Content Editor* se cargó la implementación a la tarjeta DE2-115. Con la herramienta mencionada se leyeron y escribieron datos en la memoria del FPGA en tiempo real.

En los Anexos se pueden encontrar los códigos implementados.

4. Anexos

4.1. Descripciones del hardware

```
1 module Mem(  
2   input      CLK, WE,  
3   input  [4:0] DIR,  
4   input  [7:0] DATA,  
5   output [7:0] Q  
6 );  
7  
8 MyMem u0(DIR, CLK, DATA, WE, Q);  
9  
10 endmodule
```

Programa 1: Descripción en Verilog de la memoria de 32 localidades de 8 bits, utilizando una instancia de *IP Catalog*.

4.2. Bancos de pruebas (*test benches*)

```
1 `timescale 1 ns/ 1 ps  
2 module Mem_vlg_tst();  
3   reg      CLK;  
4   reg  [7:0] DATA;  
5   reg  [4:0] DIR;  
6   reg      WE;  
7   wire  [7:0] Q;  
8  
9   Mem i1 (  
10    .CLK(CLK),  
11    .DATA(DATA),  
12    .DIR(DIR),  
13    .Q(Q),  
14    .WE(WE)  
15  );  
16  
17  initial  
18  begin  
19    $display("Running testbench at CIC");  
20    DIR = 0; WE = 0; DATA = 0; CLK = 0;  
21  end  
22  
23  always
```

```
24 begin
25     #10; CLK = ~CLK;
26     #10; CLK = ~CLK;
27     DIR = DIR + 1;
28 end
29
30 endmodule
```

Programa 2: Banco de prueba para el Programa 1.