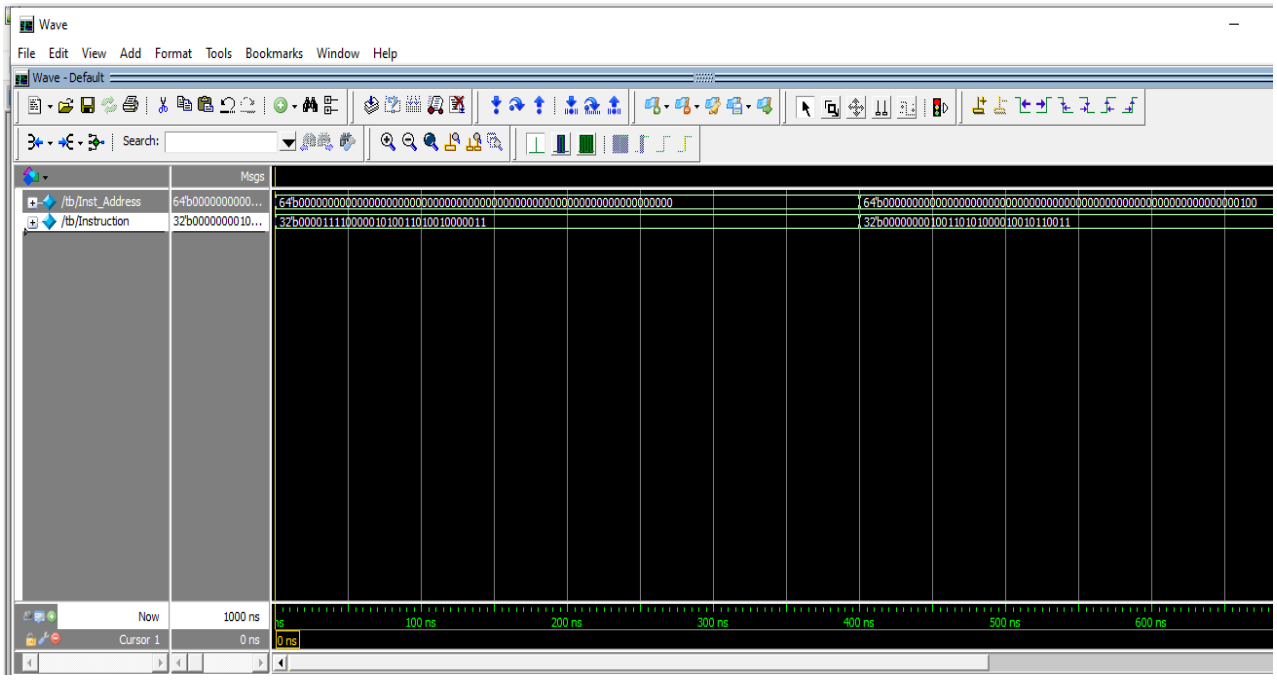
CA Lab 08 Report

Task01

```
module Instruction_Memory
(
	input[63:0] Inst_Address,
	output reg[31:0] Instruction

);
	reg[7:0] Inst_Mem[15:0];

	//initialize acc to fig:8.3
	initial
		begin
			Inst_Mem[0] = 8'b10000011;
			Inst_Mem[1] = 8'b00110100;
			Inst_Mem[2] = 8'b00000101;
			Inst_Mem[3] = 8'b00001111;
			Inst_Mem[4] = 8'b10110011;
			Inst_Mem[5] = 8'b10000100;
			Inst_Mem[6] = 8'b10011010;
			Inst_Mem[7] = 8'b00000000;
			Inst_Mem[8] = 8'b10010011;
			Inst_Mem[9] = 8'b10000100;
			Inst_Mem[10] = 8'b00010100;
			Inst_Mem[11] = 8'b00000000;
			Inst_Mem[12] = 8'b00100011;
			Inst_Mem[13] = 8'b00111000;
			Inst_Mem[14] = 8'b10010101;
			Inst_Mem[15] = 8'b00001110;
		end

	always@(*)
	begin
		//concatenate Inst_Mem[Inst_Address+3:Inst_Address] as output Instruction
		assign Instruction = { Inst_Mem[Inst_Address+3],
Inst_Mem[Inst_Address+2] ,Inst_Mem[Inst_Address+1], Inst_Mem[Inst_Address]};
	end
endmodule
```

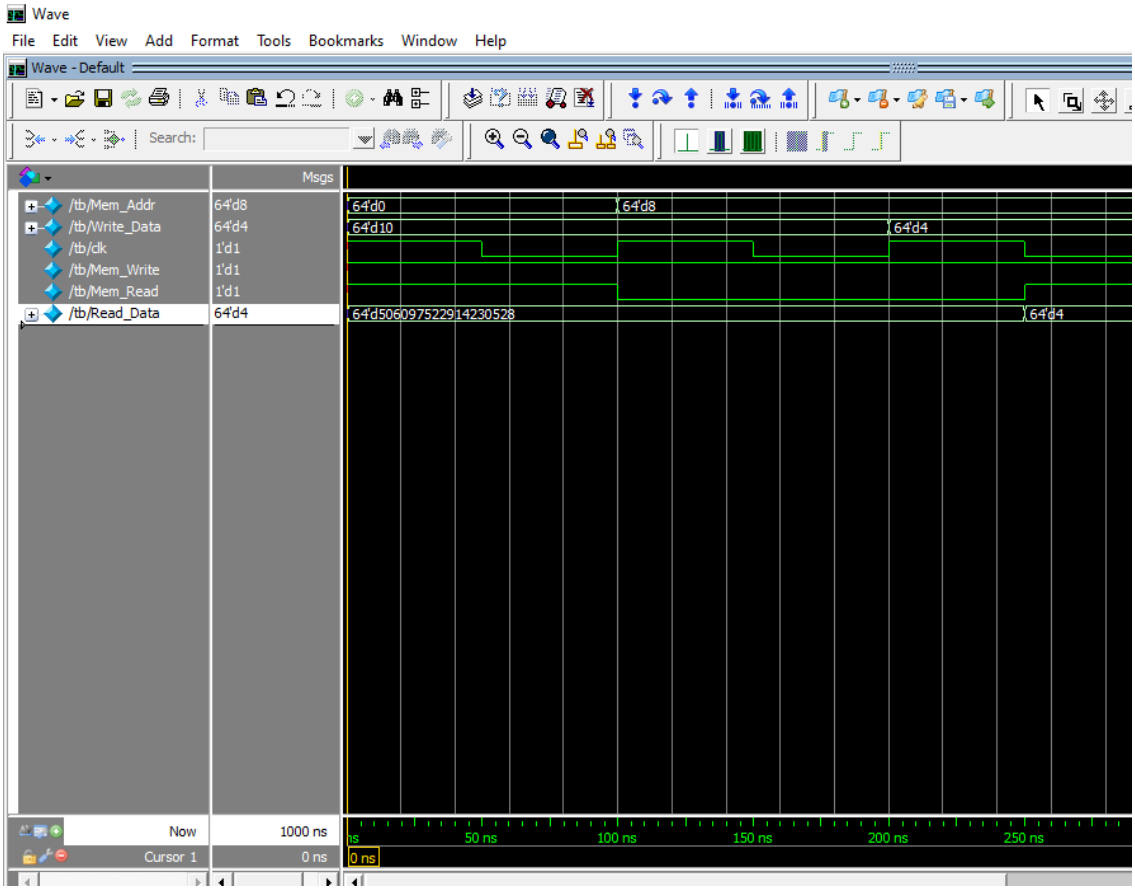
```
module tb
(
);
        reg[63:0] Inst_Address;
        wire[31:0] Instruction;
        Instruction_Memory instruction_memory
        (
                .Inst_Address(Inst_Address),
                .Instruction(Instruction)
        );
        initial
        begin
        Inst_Address = 64'd0;
        #400
        Inst_Address = 64'd4;
        end
endmodule
```



Exercise

```
module Data_Memory
(
	input[63:0] Mem_Addr, Write_Data,
	input clk, Mem_Write, Mem_Read,
	output reg[63:0] Read_Data
);
	reg [7:0] DM [63:0];
	integer i;
	initial
	begin
	for (i=0; i<64; i=i+1)
	DM[i] = i;
	end
	always@(posedge clk)
	begin
		if (Mem_Write == 1)
		begin
			DM[Mem_Addr] = Write_Data[7:0];
			DM[Mem_Addr+1] = Write_Data[15:8];
			DM[Mem_Addr+2] = Write_Data[23:16];
			DM[Mem_Addr+3] = Write_Data[31:17];
			DM[Mem_Addr+4] = Write_Data[39:18];
			DM[Mem_Addr+5] = Write_Data[47:40];
			DM[Mem_Addr+6] = Write_Data[55:48];
			DM[Mem_Addr+7] = Write_Data[63:56];
		end
	end
	always@(Mem_Addr or Mem_Read)
	begin
		if (Mem_Read == 1)
		begin
		Read_Data = { DM[Mem_Addr+7], DM[Mem_Addr+6], DM[Mem_Addr+5],
DM[Mem_Addr+4], DM[Mem_Addr+3], DM[Mem_Addr+2], DM[Mem_Addr+1], DM[Mem_Addr]   };
		end
	end
endmodule
```

```
module tb
(
);
	reg[63:0] Mem_Addr, Write_Data;
	reg clk, Mem_Write, Mem_Read;
	wire[63:0] Read_Data;

	Data_Memory data_memory
	(
		.Mem_Addr(Mem_Addr),
		.Write_Data(Write_Data),
		.clk(clk),
		.Mem_Write(Mem_Write),
		.Mem_Read(Mem_Read),
		.Read_Data(Read_Data)
	);

	initial
	begin
	clk = 0;
	Write_Data = 64'd10;
	Mem_Addr = 64'd0;
	Mem_Write = 1;
	Mem_Read = 1;

	#100
	Mem_Addr = 64'd8;
	Mem_Read = 0;

	#100
	Write_Data = 64'd4;

	#50
	Mem_Read = 1;

	end

	always
	#50 clk = ~clk;
endmodule
```


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/tb/Mem_Addr
/tb/Write_Data
/tb/clk
/tb/Mem_Write
/tb/Mem_Read
/tb/Read_Data
64'd8
64'd4
1'd1
1'd1
1'd1
64'd4
64'd0
64'd8
64'd10
64'd4
64'd506097522914230528
64'd4
Now
1000 ns
Cursor 1
0 ns
50 ns
100 ns
150 ns
200 ns
250 ns
0 ns