Codes\IF\InstructionMem.v

```
module InstructionMem (Address, Instruction);
    input [31:0] Address;
    output reg [31:0] Instruction;

    always @(Address) begin
        case (Address)
            // 0: Instruction = 32'b1110_01_0_0100_0_0100_1000_000000000000; //STR R8 ,[R4],#0
//MEM[4] = 8
            0:   Instruction = 32'b1110_00_1_1101_0_0000_0000_000000010100; //MOV R0 ,#20 //R0 =
20
            4:   Instruction = 32'b1110_00_1_1101_0_0000_0001_101000000001; //MOV R1 ,#4096 //R1
= 4096
            8:   Instruction = 32'b1110_00_1_1101_0_0000_0010_000100000011; //MOV R2 ,#0xC0000000
//R2 = -1073741824
            12:  Instruction = 32'b1110_00_0_0100_1_0010_0011_000000000010; //ADDS R3 ,R2,R2 //R3
= -2147483648
            16:  Instruction = 32'b1110_00_0_0101_0_0000_0100_000000000000; //ADC R4 ,R0,R0 //R4
= 41
            20:  Instruction = 32'b1110_00_0_0010_0_0100_0101_000100000100; //SUB R5 ,R4,R4,LSL
#2 //R5 = -123
            24:  Instruction = 32'b1110_00_0_0110_0_0000_0110_000010100000; //SBC R6 ,R0,R0,LSR
#1 //R6 = 10
            28:  Instruction = 32'b1110_00_0_1100_0_0101_0111_000101000010; //ORR R7 ,R5,R2,ASR
#2 //R7 = -123
            32:  Instruction = 32'b1110_00_0_0000_0_0111_1000_000000000011; //AND R8 ,R7,R3 //R8
= -2147483648
            36:  Instruction = 32'b1110_00_0_1111_0_0000_1001_000000000110; //MVN R9 ,R6 //R9 =
-11
            40:  Instruction = 32'b1110_00_0_0001_0_0100_1010_000000000101; //EOR R10,R4,R5 //R10
= -84
            44:  Instruction = 32'b1110_00_0_1010_1_1000_0000_000000000110; //CMP R8 ,R6
            48:  Instruction = 32'b0001_00_0_0100_0_0001_0001_000000000001; //ADDNE R1 ,R1,R1
//R1 = 8192
            52:  Instruction = 32'b1110_00_0_1000_1_1001_0000_000000001000; //TST R9 ,R8
            56:  Instruction = 32'b0000_00_0_0100_0_0010_0010_000000000010; //ADDEQ R2 ,R2,R2
//R2 = -1073741824
            60:  Instruction = 32'b1110_00_1_1101_0_0000_0000_101100000001; //MOV R0 ,#1024 //R0
= 1024
            64:  Instruction = 32'b1110_01_0_0100_0_0000_0001_000000000000; //STR R1 ,[R0],#0
//MEM[1024] = 8192
            68:  Instruction = 32'b1110_01_0_0100_1_0000_1011_000000000000; //LDR R11,[R0],#0
//R11 = 8192
            72:  Instruction = 32'b1110_01_0_0100_0_0000_0010_000000000100; //STR R2 ,[R0],#4
//MEM[1028] = -1073741824
            76:  Instruction = 32'b1110_01_0_0100_0_0000_0011_000000001000; //STR R3 ,[R0],#8
//MEM[1032] = -2147483648
            80:  Instruction = 32'b1110_01_0_0100_0_0000_0100_000000001101; //STR R4 ,[R0],#13
//MEM[1036] = 41
            84:  Instruction = 32'b1110_01_0_0100_0_0000_0101_000000010000; //STR R5 ,[R0],#16
//MEM[1040] = -123
            88:  Instruction = 32'b1110_01_0_0100_0_0000_0110_000000010100; //STR R6 ,[R0],#20
//MEM[1044] = 10
            92:  Instruction = 32'b1110_01_0_0100_1_0000_1010_000000000100; //LDR R10,[R0],#4
//R10 = -1073741824
            96:  Instruction = 32'b1110_01_0_0100_0_0000_0111_000000011000; //STR R7 ,[R0],#24
//MEM[1048] = -123
            100: Instruction = 32'b1110_00_1_1101_0_0000_0001_000000000100; //MOV R1 ,#4 //R1 = 4
            104: Instruction = 32'b1110_00_1_1101_0_0000_0010_000000000000; //MOV R2 ,#0 //R2 = 0
            108: Instruction = 32'b1110_00_1_1101_0_0000_0011_000000000000; //MOV R3 ,#0 //R3 = 0
            112: Instruction = 32'b1110_00_0_0100_0_0000_0100_000100000011; //ADD R4 ,R0,R3,LSL #2
            116: Instruction = 32'b1110_01_0_0100_1_0100_0101_000000000000; //LDR R5 ,[R4],#0
            120: Instruction = 32'b1110_01_0_0100_1_0100_0110_000000000100; //LDR R6 ,[R4],#4
            124: Instruction = 32'b1110_00_0_1010_1_0101_0000_000000000110; //CMP R5 ,R6
            128: Instruction = 32'b1100_01_0_0100_0_0100_0110_000000000000; //STRGT R6 ,[R4],#0
            132: Instruction = 32'b1100_01_0_0100_0_0100_0101_000000000100; //STRGT R5 ,[R4],#4
            136: Instruction = 32'b1110_00_1_0100_0_0011_0011_000000000001; //ADD R3 ,R3,#1
            140: Instruction = 32'b1110_00_1_1010_1_0011_0000_000000000011; //CMP R3 ,#3
            144: Instruction = 32'b1011_10_1_0_111111111111111111110111 ; //BLT #-9
            148: Instruction = 32'b1110_00_1_0100_0_0010_0010_000000000001; //ADD R2 ,R2,#1
            152: Instruction = 32'b1110_00_0_1010_1_0010_0000_000000000001; //CMP R2 ,R1
            156: Instruction = 32'b1011_10_1_0_111111111111111111110011 ; //BLT #-13
            160: Instruction = 32'b1110_01_0_0100_1_0000_0001_000000000000; //LDR R1 ,[R0],#0 //R1 = -2147483648
            164: Instruction = 32'b1110_01_0_0100_1_0000_0010_000000000100; //LDR R2 ,[R0],#4 //R2 = -1073741824
            168: Instruction = 32'b1110_01_0_0100_1_0000_0011_000000001000; //LDR R3 ,[R0],#8 //R3 = 41
            172: Instruction = 32'b1110_01_0_0100_1_0000_0100_000000001100; //LDR R4 ,[R0],#12 //R4 = 8192
            176: Instruction = 32'b1110_01_0_0100_1_0000_0101_000000010000; //LDR R5 ,[R0],#16 //R5 = -123
            180: Instruction = 32'b1110_01_0_0100_1_0000_0110_000000010100; //LDR R6 ,[R0],#20 //R4 = 10
            184: Instruction = 32'b1110_10_1_0_111111111111111111111111 ; //B #-1
            default: Instruction = 32'bzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz;
        endcase
    end

endmodule
```