**CSE4117 – Microprocessors**



**THE DESIGN OF RAM**

**NAME SURNAME** **STUDENT NUMBER**

Aydın DUYGU 150118981

Tugay SARICI 150119829

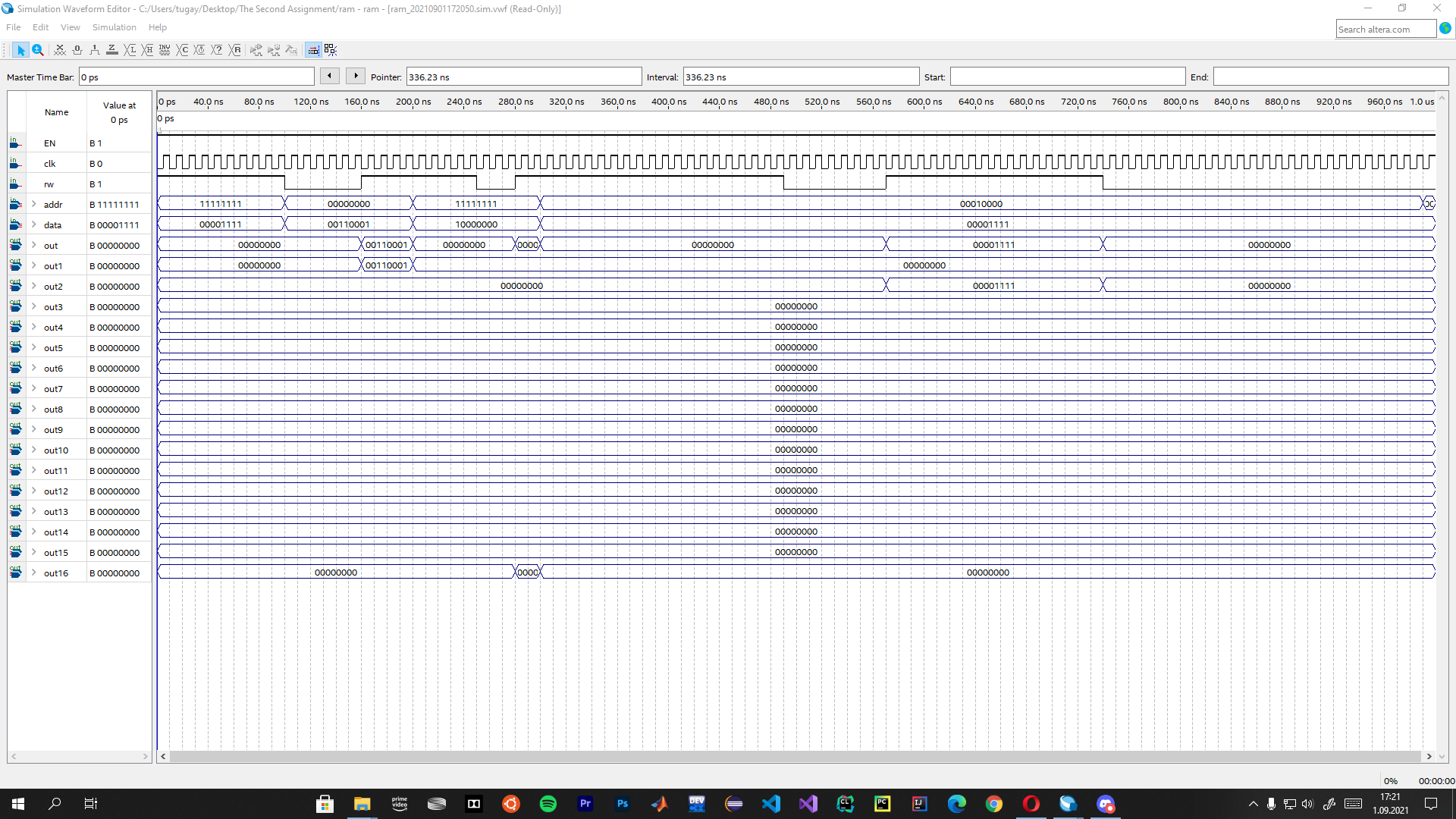
**INTRODUCTION**

In this assignment, we designed a RAM (256 addresses x 8 bits) by using binary cells and decoders. Firstly, we connected the most significant 4 bits of the address with an enable bit to the an decoder (4 x 16). And then we connected the outputs of this decoder (as an enable bit) together with the least significant 4 bits of the address to the 16 seperate decoders. In this way, we got 256 different addresses to store any data.

At the last part, we connected each 16 bits part of the addresses that we got to binary cells with the data, clk and rw signals. At this place, each bit of the 16 bits parts is used as a selection for the binary cells. As a result, the data is stored at the specified address.

We have seen that our design yielded successful results in our simulations. We attach the simulation below.

**TESTS**



**SOURCE CODE**

|  |
| --- |
| module dflipflop(d, q, clk);  input d, clk;  output q;  reg q;  always @ (posedge clk)  begin  q <= d;  end  endmodule  //------------------------------------------------------------  module binarycell(sel, dataIn, rw, clk, dataOut);  input dataIn, clk, sel, rw;  output dataOut;  wire w1, w2, q;  reg dataOut;  assign w1 = (sel & ~rw);  assign w2 = w1 ? dataIn : dataOut;  dflipflop binflop(w2, q, clk);  always @ ( \* ) begin  dataOut <= (sel & rw & q);  end  endmodule  //------------------------------------------------------------  module decoder\_4to16(a1, a2, a3, a4, e, dOut);  input a1, a2, a3, a4, e;  output [15:0] dOut;  assign dOut[0] = (~a1) & (~a2) &(~a3) & (~a4) & (e);  assign dOut[1] = (~a1) & (~a2) &(~a3) & (a4) & (e);  assign dOut[2] = (~a1) & (~a2) &(a3) & (~a4) & (e);  assign dOut[3] = (~a1) & (~a2) &(a3) & (a4) & (e);  assign dOut[4] = (~a1) & (a2) &(~a3) & (~a4) & (e);  assign dOut[5] = (~a1) & (a2) &(~a3) & (a4) & (e);  assign dOut[6] = (~a1) & (a2) &(a3) & (~a4) & (e);  assign dOut[7] = (~a1) & (a2) &(a3) & (a4) & (e);  assign dOut[8] = (a1) & (~a2) &(~a3) & (~a4) & (e);  assign dOut[9] = (a1) & (~a2) &(~a3) & (a4) & (e);  assign dOut[10] = (a1) & (~a2) &(a3) & (~a4) & (e);  assign dOut[11] = (a1) & (~a2) &(a3) & (a4) & (e);  assign dOut[12] = (a1) & (a2) &(~a3) & (~a4) & (e);  assign dOut[13] = (a1) & (a2) &(~a3) & (a4) & (e);  assign dOut[14] = (a1) & (a2) &(a3) & (~a4) & (e);  assign dOut[15] = (a1) & (a2) &(a3) & (a4) & (e);  endmodule  //------------------------------------------------------------  module RamPart(dOut, data, rw, clk, out);  input [15:0] dOut;  input [7:0] data;  input rw, clk;  output [7:0] out;  wire out0000\_1 , out0001\_1 , out0010\_1 , out0011\_1 , out0100\_1 , out0101\_1 , out0110\_1 , out0111\_1 , out1000\_1 , out1001\_1 , out1010\_1 , out1011\_1 , out1100\_1 , out1101\_1 , out1110\_1 , out1111\_1;  wire out0000\_2 , out0001\_2 , out0010\_2 , out0011\_2 , out0100\_2 , out0101\_2 , out0110\_2 , out0111\_2 , out1000\_2 , out1001\_2 , out1010\_2 , out1011\_2 , out1100\_2 , out1101\_2 , out1110\_2 , out1111\_2;  wire out0000\_3 , out0001\_3 , out0010\_3 , out0011\_3 , out0100\_3 , out0101\_3 , out0110\_3 , out0111\_3 , out1000\_3 , out1001\_3 , out1010\_3 , out1011\_3 , out1100\_3 , out1101\_3 , out1110\_3 , out1111\_3;  wire out0000\_4 , out0001\_4 , out0010\_4 , out0011\_4 , out0100\_4 , out0101\_4 , out0110\_4 , out0111\_4 , out1000\_4 , out1001\_4 , out1010\_4 , out1011\_4 , out1100\_4 , out1101\_4 , out1110\_4 , out1111\_4;  wire out0000\_5 , out0001\_5 , out0010\_5 , out0011\_5 , out0100\_5 , out0101\_5 , out0110\_5 , out0111\_5 , out1000\_5 , out1001\_5 , out1010\_5 , out1011\_5 , out1100\_5 , out1101\_5 , out1110\_5 , out1111\_5;  wire out0000\_6 , out0001\_6 , out0010\_6 , out0011\_6 , out0100\_6 , out0101\_6 , out0110\_6 , out0111\_6 , out1000\_6 , out1001\_6 , out1010\_6 , out1011\_6 , out1100\_6 , out1101\_6 , out1110\_6 , out1111\_6;  wire out0000\_7 , out0001\_7 , out0010\_7 , out0011\_7 , out0100\_7 , out0101\_7 , out0110\_7 , out0111\_7 , out1000\_7 , out1001\_7 , out1010\_7 , out1011\_7 , out1100\_7 , out1101\_7 , out1110\_7 , out1111\_7;  wire out0000\_8 , out0001\_8 , out0010\_8 , out0011\_8 , out0100\_8 , out0101\_8 , out0110\_8 , out0111\_8 , out1000\_8 , out1001\_8 , out1010\_8 , out1011\_8 , out1100\_8 , out1101\_8 , out1110\_8 , out1111\_8;    //module binarycell(en,data,rw,clk,out);  binarycell cell\_0000\_1(dOut[0],data[0],rw,clk,out0000\_1);  binarycell cell\_0001\_1(dOut[1],data[0],rw,clk,out0001\_1);  binarycell cell\_0010\_1(dOut[2],data[0],rw,clk,out0010\_1);  binarycell cell\_0011\_1(dOut[3],data[0],rw,clk,out0011\_1);  binarycell cell\_0100\_1(dOut[4],data[0],rw,clk,out0100\_1);  binarycell cell\_0101\_1(dOut[5],data[0],rw,clk,out0101\_1);  binarycell cell\_0110\_1(dOut[6],data[0],rw,clk,out0110\_1);  binarycell cell\_0111\_1(dOut[7],data[0],rw,clk,out0111\_1);  binarycell cell\_1000\_1(dOut[8],data[0],rw,clk,out1000\_1);  binarycell cell\_1001\_1(dOut[9],data[0],rw,clk,out1001\_1);  binarycell cell\_1010\_1(dOut[10],data[0],rw,clk,out1010\_1);  binarycell cell\_1011\_1(dOut[11],data[0],rw,clk,out1011\_1);  binarycell cell\_1100\_1(dOut[12],data[0],rw,clk,out1100\_1);  binarycell cell\_1101\_1(dOut[13],data[0],rw,clk,out1101\_1);  binarycell cell\_1110\_1(dOut[14],data[0],rw,clk,out1110\_1);  binarycell cell\_1111\_1(dOut[15],data[0],rw,clk,out1111\_1);  binarycell cell\_0000\_2(dOut[0],data[1],rw,clk,out0000\_2);  binarycell cell\_0001\_2(dOut[1],data[1],rw,clk,out0001\_2);  binarycell cell\_0010\_2(dOut[2],data[1],rw,clk,out0010\_2);  binarycell cell\_0011\_2(dOut[3],data[1],rw,clk,out0011\_2);  binarycell cell\_0100\_2(dOut[4],data[1],rw,clk,out0100\_2);  binarycell cell\_0101\_2(dOut[5],data[1],rw,clk,out0101\_2);  binarycell cell\_0110\_2(dOut[6],data[1],rw,clk,out0110\_2);  binarycell cell\_0111\_2(dOut[7],data[1],rw,clk,out0111\_2);  binarycell cell\_1000\_2(dOut[8],data[1],rw,clk,out1000\_2);  binarycell cell\_1001\_2(dOut[9],data[1],rw,clk,out1001\_2);  binarycell cell\_1010\_2(dOut[10],data[1],rw,clk,out1010\_2);  binarycell cell\_1011\_2(dOut[11],data[1],rw,clk,out1011\_2);  binarycell cell\_1100\_2(dOut[12],data[1],rw,clk,out1100\_2);  binarycell cell\_1101\_2(dOut[13],data[1],rw,clk,out1101\_2);  binarycell cell\_1110\_2(dOut[14],data[1],rw,clk,out1110\_2);  binarycell cell\_1111\_2(dOut[15],data[1],rw,clk,out1111\_2);  binarycell cell\_0000\_3(dOut[0],data[2],rw,clk,out0000\_3);  binarycell cell\_0001\_3(dOut[1],data[2],rw,clk,out0001\_3);  binarycell cell\_0010\_3(dOut[2],data[2],rw,clk,out0010\_3);  binarycell cell\_0011\_3(dOut[3],data[2],rw,clk,out0011\_3);  binarycell cell\_0100\_3(dOut[4],data[2],rw,clk,out0100\_3);  binarycell cell\_0101\_3(dOut[5],data[2],rw,clk,out0101\_3);  binarycell cell\_0110\_3(dOut[6],data[2],rw,clk,out0110\_3);  binarycell cell\_0111\_3(dOut[7],data[2],rw,clk,out0111\_3);  binarycell cell\_1000\_3(dOut[8],data[2],rw,clk,out1000\_3);  binarycell cell\_1001\_3(dOut[9],data[2],rw,clk,out1001\_3);  binarycell cell\_1010\_3(dOut[10],data[2],rw,clk,out1010\_3);  binarycell cell\_1011\_3(dOut[11],data[2],rw,clk,out1011\_3);  binarycell cell\_1100\_3(dOut[12],data[2],rw,clk,out1100\_3);  binarycell cell\_1101\_3(dOut[13],data[2],rw,clk,out1101\_3);  binarycell cell\_1110\_3(dOut[14],data[2],rw,clk,out1110\_3);  binarycell cell\_1111\_3(dOut[15],data[2],rw,clk,out1111\_3);  binarycell cell\_0000\_4(dOut[0],data[3],rw,clk,out0000\_4);  binarycell cell\_0001\_4(dOut[1],data[3],rw,clk,out0001\_4);  binarycell cell\_0010\_4(dOut[2],data[3],rw,clk,out0010\_4);  binarycell cell\_0011\_4(dOut[3],data[3],rw,clk,out0011\_4);  binarycell cell\_0100\_4(dOut[4],data[3],rw,clk,out0100\_4);  binarycell cell\_0101\_4(dOut[5],data[3],rw,clk,out0101\_4);  binarycell cell\_0110\_4(dOut[6],data[3],rw,clk,out0110\_4);  binarycell cell\_0111\_4(dOut[7],data[3],rw,clk,out0111\_4);  binarycell cell\_1000\_4(dOut[8],data[3],rw,clk,out1000\_4);  binarycell cell\_1001\_4(dOut[9],data[3],rw,clk,out1001\_4);  binarycell cell\_1010\_4(dOut[10],data[3],rw,clk,out1010\_4);  binarycell cell\_1011\_4(dOut[11],data[3],rw,clk,out1011\_4);  binarycell cell\_1100\_4(dOut[12],data[3],rw,clk,out1100\_4);  binarycell cell\_1101\_4(dOut[13],data[3],rw,clk,out1101\_4);  binarycell cell\_1110\_4(dOut[14],data[3],rw,clk,out1110\_4);  binarycell cell\_1111\_4(dOut[15],data[3],rw,clk,out1111\_4);  binarycell cell\_0000\_5(dOut[0],data[4],rw,clk,out0000\_5);  binarycell cell\_0001\_5(dOut[1],data[4],rw,clk,out0001\_5);  binarycell cell\_0010\_5(dOut[2],data[4],rw,clk,out0010\_5);  binarycell cell\_0011\_5(dOut[3],data[4],rw,clk,out0011\_5);  binarycell cell\_0100\_5(dOut[4],data[4],rw,clk,out0100\_5);  binarycell cell\_0101\_5(dOut[5],data[4],rw,clk,out0101\_5);  binarycell cell\_0110\_5(dOut[6],data[4],rw,clk,out0110\_5);  binarycell cell\_0111\_5(dOut[7],data[4],rw,clk,out0111\_5);  binarycell cell\_1000\_5(dOut[8],data[4],rw,clk,out1000\_5);  binarycell cell\_1001\_5(dOut[9],data[4],rw,clk,out1001\_5);  binarycell cell\_1010\_5(dOut[10],data[4],rw,clk,out1010\_5);  binarycell cell\_1011\_5(dOut[11],data[4],rw,clk,out1011\_5);  binarycell cell\_1100\_5(dOut[12],data[4],rw,clk,out1100\_5);  binarycell cell\_1101\_5(dOut[13],data[4],rw,clk,out1101\_5);  binarycell cell\_1110\_5(dOut[14],data[4],rw,clk,out1110\_5);  binarycell cell\_1111\_5(dOut[15],data[4],rw,clk,out1111\_5);  binarycell cell\_0000\_6(dOut[0],data[5],rw,clk,out0000\_6);  binarycell cell\_0001\_6(dOut[1],data[5],rw,clk,out0001\_6);  binarycell cell\_0010\_6(dOut[2],data[5],rw,clk,out0010\_6);  binarycell cell\_0011\_6(dOut[3],data[5],rw,clk,out0011\_6);  binarycell cell\_0100\_6(dOut[4],data[5],rw,clk,out0100\_6);  binarycell cell\_0101\_6(dOut[5],data[5],rw,clk,out0101\_6);  binarycell cell\_0110\_6(dOut[6],data[5],rw,clk,out0110\_6);  binarycell cell\_0111\_6(dOut[7],data[5],rw,clk,out0111\_6);  binarycell cell\_1000\_6(dOut[8],data[5],rw,clk,out1000\_6);  binarycell cell\_1001\_6(dOut[9],data[5],rw,clk,out1001\_6);  binarycell cell\_1010\_6(dOut[10],data[5],rw,clk,out1010\_6);  binarycell cell\_1011\_6(dOut[11],data[5],rw,clk,out1011\_6);  binarycell cell\_1100\_6(dOut[12],data[5],rw,clk,out1100\_6);  binarycell cell\_1101\_6(dOut[13],data[5],rw,clk,out1101\_6);  binarycell cell\_1110\_6(dOut[14],data[5],rw,clk,out1110\_6);  binarycell cell\_1111\_6(dOut[15],data[5],rw,clk,out1111\_6);  binarycell cell\_0000\_7(dOut[0],data[6],rw,clk,out0000\_7);  binarycell cell\_0001\_7(dOut[1],data[6],rw,clk,out0001\_7);  binarycell cell\_0010\_7(dOut[2],data[6],rw,clk,out0010\_7);  binarycell cell\_0011\_7(dOut[3],data[6],rw,clk,out0011\_7);  binarycell cell\_0100\_7(dOut[4],data[6],rw,clk,out0100\_7);  binarycell cell\_0101\_7(dOut[5],data[6],rw,clk,out0101\_7);  binarycell cell\_0110\_7(dOut[6],data[6],rw,clk,out0110\_7);  binarycell cell\_0111\_7(dOut[7],data[6],rw,clk,out0111\_7);  binarycell cell\_1000\_7(dOut[8],data[6],rw,clk,out1000\_7);  binarycell cell\_1001\_7(dOut[9],data[6],rw,clk,out1001\_7);  binarycell cell\_1010\_7(dOut[10],data[6],rw,clk,out1010\_7);  binarycell cell\_1011\_7(dOut[11],data[6],rw,clk,out1011\_7);  binarycell cell\_1100\_7(dOut[12],data[6],rw,clk,out1100\_7);  binarycell cell\_1101\_7(dOut[13],data[6],rw,clk,out1101\_7);  binarycell cell\_1110\_7(dOut[14],data[6],rw,clk,out1110\_7);  binarycell cell\_1111\_7(dOut[15],data[6],rw,clk,out1111\_7);  binarycell cell\_0000\_8(dOut[0],data[7],rw,clk,out0000\_8);  binarycell cell\_0001\_8(dOut[1],data[7],rw,clk,out0001\_8);  binarycell cell\_0010\_8(dOut[2],data[7],rw,clk,out0010\_8);  binarycell cell\_0011\_8(dOut[3],data[7],rw,clk,out0011\_8);  binarycell cell\_0100\_8(dOut[4],data[7],rw,clk,out0100\_8);  binarycell cell\_0101\_8(dOut[5],data[7],rw,clk,out0101\_8);  binarycell cell\_0110\_8(dOut[6],data[7],rw,clk,out0110\_8);  binarycell cell\_0111\_8(dOut[7],data[7],rw,clk,out0111\_8);  binarycell cell\_1000\_8(dOut[8],data[7],rw,clk,out1000\_8);  binarycell cell\_1001\_8(dOut[9],data[7],rw,clk,out1001\_8);  binarycell cell\_1010\_8(dOut[10],data[7],rw,clk,out1010\_8);  binarycell cell\_1011\_8(dOut[11],data[7],rw,clk,out1011\_8);  binarycell cell\_1100\_8(dOut[12],data[7],rw,clk,out1100\_8);  binarycell cell\_1101\_8(dOut[13],data[7],rw,clk,out1101\_8);  binarycell cell\_1110\_8(dOut[14],data[7],rw,clk,out1110\_8);  binarycell cell\_1111\_8(dOut[15],data[7],rw,clk,out1111\_8);  assign out[0] = out0000\_1 || out0001\_1 || out0010\_1 || out0011\_1 || out0100\_1 || out0101\_1 || out0110\_1 || out0111\_1 || out1000\_1 || out1001\_1 || out1010\_1 || out1011\_1 || out1100\_1 || out1101\_1 || out1110\_1 || out1111\_1;  assign out[1] = out0000\_2 || out0001\_2 || out0010\_2 || out0011\_2 || out0100\_2 || out0101\_2 || out0110\_2 || out0111\_2 || out1000\_2 || out1001\_2 || out1010\_2 || out1011\_2 || out1100\_2 || out1101\_2 || out1110\_2 || out1111\_2;  assign out[2] = out0000\_3 || out0001\_3 || out0010\_3 || out0011\_3 || out0100\_3 || out0101\_3 || out0110\_3 || out0111\_3 || out1000\_3 || out1001\_3 || out1010\_3 || out1011\_3 || out1100\_3 || out1101\_3 || out1110\_3 || out1111\_3;  assign out[3] = out0000\_4 || out0001\_4 || out0010\_4 || out0011\_4 || out0100\_4 || out0101\_4 || out0110\_4 || out0111\_4 || out1000\_4 || out1001\_4 || out1010\_4 || out1011\_4 || out1100\_4 || out1101\_4 || out1110\_4 || out1111\_4;  assign out[4] = out0000\_5 || out0001\_5 || out0010\_5 || out0011\_5 || out0100\_5 || out0101\_5 || out0110\_5 || out0111\_5 || out1000\_5 || out1001\_5 || out1010\_5 || out1011\_5 || out1100\_5 || out1101\_5 || out1110\_5 || out1111\_5;  assign out[5] = out0000\_6 || out0001\_6 || out0010\_6 || out0011\_6 || out0100\_6 || out0101\_6 || out0110\_6 || out0111\_6 || out1000\_6 || out1001\_6 || out1010\_6 || out1011\_6 || out1100\_6 || out1101\_6 || out1110\_6 || out1111\_6;  assign out[6] = out0000\_7 || out0001\_7 || out0010\_7 || out0011\_7 || out0100\_7 || out0101\_7 || out0110\_7 || out0111\_7 || out1000\_7 || out1001\_7 || out1010\_7 || out1011\_7 || out1100\_7 || out1101\_7 || out1110\_7 || out1111\_7;  assign out[7] = out0000\_8 || out0001\_8 || out0010\_8 || out0011\_8 || out0100\_8 || out0101\_8 || out0110\_8 || out0111\_8 || out1000\_8 || out1001\_8 || out1010\_8 || out1011\_8 || out1100\_8 || out1101\_8 || out1110\_8 || out1111\_8;  endmodule  //------------------------------------------------------------  module ram(data, rw, clk, out, addr, EN);  input [7:0] addr;  input rw, clk, EN;  input[7:0] data;  output[7:0] out;  wire [7:0] out1, out2,out3,out4,out5,out6,out7,out8,out9,out10,out11,out12,out13,out14,out15,out16;  wire [15:0] dOut;  wire [15:0] d1\_Out;  wire [15:0] d2\_Out;  wire [15:0] d3\_Out;  wire [15:0] d4\_Out;  wire [15:0] d5\_Out;  wire [15:0] d6\_Out;  wire [15:0] d7\_Out;  wire [15:0] d8\_Out;  wire [15:0] d9\_Out;  wire [15:0] d10\_Out;  wire [15:0] d11\_Out;  wire [15:0] d12\_Out;  wire [15:0] d13\_Out;  wire [15:0] d14\_Out;  wire [15:0] d15\_Out;  wire [15:0] d16\_Out;  decoder\_4to16 decoder(addr[7], addr[6], addr[5], addr[4], EN, dOut);  decoder\_4to16 decoder1(addr[3], addr[2], addr[1], addr[0], dOut[0], d1\_Out);  decoder\_4to16 decoder2(addr[3], addr[2], addr[1], addr[0], dOut[1], d2\_Out);  decoder\_4to16 decoder3(addr[3], addr[2], addr[1], addr[0], dOut[2], d3\_Out);  decoder\_4to16 decoder4(addr[3], addr[2], addr[1], addr[0], dOut[3], d4\_Out);  decoder\_4to16 decoder5(addr[3], addr[2], addr[1], addr[0], dOut[4], d5\_Out);  decoder\_4to16 decoder6(addr[3], addr[2], addr[1], addr[0], dOut[5], d6\_Out);  decoder\_4to16 decoder7(addr[3], addr[2], addr[1], addr[0], dOut[6], d7\_Out);  decoder\_4to16 decoder8(addr[3], addr[2], addr[1], addr[0], dOut[7], d8\_Out);  decoder\_4to16 decoder9(addr[3], addr[2], addr[1], addr[0], dOut[8], d9\_Out);  decoder\_4to16 decoder10(addr[3], addr[2], addr[1], addr[0], dOut[9], d10\_Out);  decoder\_4to16 decoder11(addr[3], addr[2], addr[1], addr[0], dOut[10],d11\_Out);  decoder\_4to16 decoder12(addr[3], addr[2], addr[1], addr[0], dOut[11],d12\_Out);  decoder\_4to16 decoder13(addr[3], addr[2], addr[1], addr[0], dOut[12],d13\_Out);  decoder\_4to16 decoder14(addr[3], addr[2], addr[1], addr[0], dOut[13],d14\_Out);  decoder\_4to16 decoder15(addr[3], addr[2], addr[1], addr[0], dOut[14],d15\_Out);  decoder\_4to16 decoder16(addr[3], addr[2], addr[1], addr[0], dOut[15],d16\_Out);  RamPart h1(d1\_Out, data, rw, clk, out1);  RamPart h2(d2\_Out, data, rw, clk, out2);  RamPart h3(d3\_Out, data, rw, clk, out3);  RamPart h4(d4\_Out, data, rw, clk, out4);  RamPart h5(d5\_Out, data, rw, clk, out5);  RamPart h6(d6\_Out, data, rw, clk, out6);  RamPart h7(d7\_Out, data, rw, clk, out7);  RamPart h8(d8\_Out, data, rw, clk, out8);  RamPart h9(d9\_Out, data, rw, clk, out9);  RamPart h10(d10\_Out, data, rw, clk, out10);  RamPart h11(d11\_Out, data, rw, clk, out11);  RamPart h12(d12\_Out, data, rw, clk, out12);  RamPart h13(d13\_Out, data, rw, clk, out13);  RamPart h14(d14\_Out, data, rw, clk, out14);  RamPart h15(d15\_Out, data, rw, clk, out15);  RamPart h16(d16\_Out, data, rw, clk, out16);  assign out = out1 | out2 | out3 | out4 | out5 | out6 | out7 | out8 | out9 | out10 | out11 | out12 | out13 | out14 | out15 | out16;  endmodule |

**The RTL View**

