**module ROM\_32x8 (addrb, datab, read\_en);**

**input [4:0] addrb;**

**output [7:0] datab;**

**input read\_en;**

**wire [3:0] OUT;**

**Dec\_24 d1 (addrb[4], addrb[3], OUT);**

**ROM1\_8x8 rc1 (OUT[0], addrb[2:0], datab, read\_en);**

**ROM2\_8x8 rc2 (OUT[1], addrb[2:0], datab, read\_en);**

**ROM3\_8x8 rc3 (OUT[2], addrb[2:0], datab, read\_en);**

**ROM4\_8x8 rc4 (OUT[3], addrb[2:0], datab, read\_en);**

**endmodule**

**module Dec\_24 (A, B, OUT);**

**input A, B;**

**output [3:0] OUT;**

**assign OUT[0] = ~A & ~B;**

**assign OUT[1] = ~A & B;**

**assign OUT[2] = A & ~B;**

**assign OUT[3] = A & B;**

**endmodule**

**module ROM1\_8x8 (cs, addrb, datab, read\_en);**

**input [2:0] addrb;**

**output [7:0] datab;**

**input read\_en, cs;**

**reg [7:0] datab;**

**always @ (\*)**

**if (read\_en & cs)**

**case (addrb)**

**0: datab = 21;**

**1: datab = 255;**

**2: datab = 33;**

**3: datab = 99;**

**4: datab = 127;**

**5: datab = 13;**

**6: datab = 10;**

**7: datab = 88;**

**endcase**

**else**

**datab = 8'bz;**

**endmodule**

**module ROM2\_8x8 (cs, addrb, datab, read\_en);**

**input [2:0] addrb;**

**output [7:0] datab;**

**input read\_en, cs;**

**reg [7:0] datab;**

**always @ (\*)**

**if (read\_en & cs)**

**case (addrb)**

**0: datab = 28;**

**1: datab = 38;**

**2: datab = 48;**

**3: datab = 58;**

**4: datab = 68;**

**5: datab = 78;**

**6: datab = 88;**

**7: datab = 98;**

**endcase**

**else**

**datab = 8'bz;**

**endmodule**

**module ROM3\_8x8 (cs, addrb, datab, read\_en);**

**input [2:0] addrb;**

**output [7:0] datab;**

**input read\_en, cs;**

**reg [7:0] datab;**

**always @ (\*)**

**if (read\_en & cs)**

**case (addrb)**

**0: datab = 128;**

**1: datab = 138;**

**2: datab = 148;**

**3: datab = 158;**

**4: datab = 168;**

**5: datab = 178;**

**6: datab = 188;**

**7: datab = 198;**

**endcase**

**else**

**datab = 8'bz;**

**endmodule**

**module ROM4\_8x8 (cs, addrb, datab, read\_en);**

**input [2:0] addrb;**

**output [7:0] datab;**

**input read\_en, cs;**

**reg [7:0] datab;**

**always @ (\*)**

**if (read\_en & cs)**

**case (addrb)**

**0: datab = 1;**

**1: datab = 2;**

**2: datab = 3;**

**3: datab = 4;**

**4: datab = 5;**

**5: datab = 6;**

**6: datab = 7;**

**7: datab = 111;**

**endcase**

**else**

**datab = 8'bz;**

**endmodule**

**module tst\_ROM\_32x8;**

**reg [4:0] addrb;**

**wire [7:0] datab;**

**reg read\_en;**

**ROM\_32x8 rom (addrb, datab, read\_en);**

**initial begin**

**#5 addrb = 0; //00 000**

**#10 addrb = 7; //00 111**

**#10 addrb = 8; //01 000**

**#5 addrb = 15; //01 111**

**#5 addrb = 24; //11 000**

**#5 addrb = 31; //11 111**

**#5 addrb = 20; //10 100**

**#5 addrb = 21; //10 101**

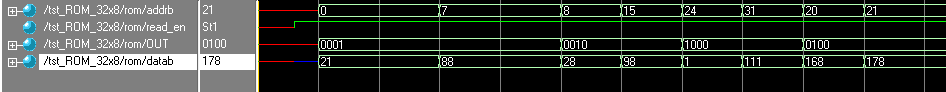
**end**

**initial begin**

**#3 read\_en <= 1;**

**end**

**endmodule**

****