Open

ram.v

~/Desktop/ram

Save

ram.v

ram_tb.v

```
1 `timescale 1ns/1ps
2 module ram(di,do,a,w,r,clk);
3 input clk;
4 input [7:0] a;
5 input [31:0] di;
6 input r,w;
7 output  reg [31:0] do;
8 reg [31:0] ram[255:0];
9 always@(posedge  clk)begin
10 if(w)begin
11 ram[a]<=di;
12 end
13 if(r)
14 begin
15 do=ram[a];
16 end
17 else
18 begin
19 ram[a]=32'b0;
20 end
21 end
22 endmodule
```

Verilog

Tab Width: 8

Ln 22, Col 10

INS

Open

ram_tb.v
~/Desktop/ram

Save

ram.v

ram_tb.v

```
`timescale 1ns/1ps
module ram_tb;
reg w,r;
reg [7:0]a;
reg [31:0]di;
reg clk;
wire [31:0] do;
integer i;
ram u0(
       .a(a),
       .di(di),
       .r(r),
       .w(w),
       .do(do),
       .clk(clk));
       always #50 clk=~clk;
       initial begin
       $dumpfile("ram_tb.vcd");
       $dumpvars(0,ram_tb);
       clk=0;
       a=0;di=0;w=0;r=0;#100;
       w=1;a=8'h10;di=32'hdeadbeef;#100;
       w=0;r=1;a=8'h10;#100;
       $display("Data out :%h",do);
       for(i=0;i<10;i++)
       begin
       w=1;r=0;a=i;
       di=i*32'h100;
       #100;
       w=0;r=1;#100;
       $display("Data out at a %d :%b",a,do);
       end
```

Open

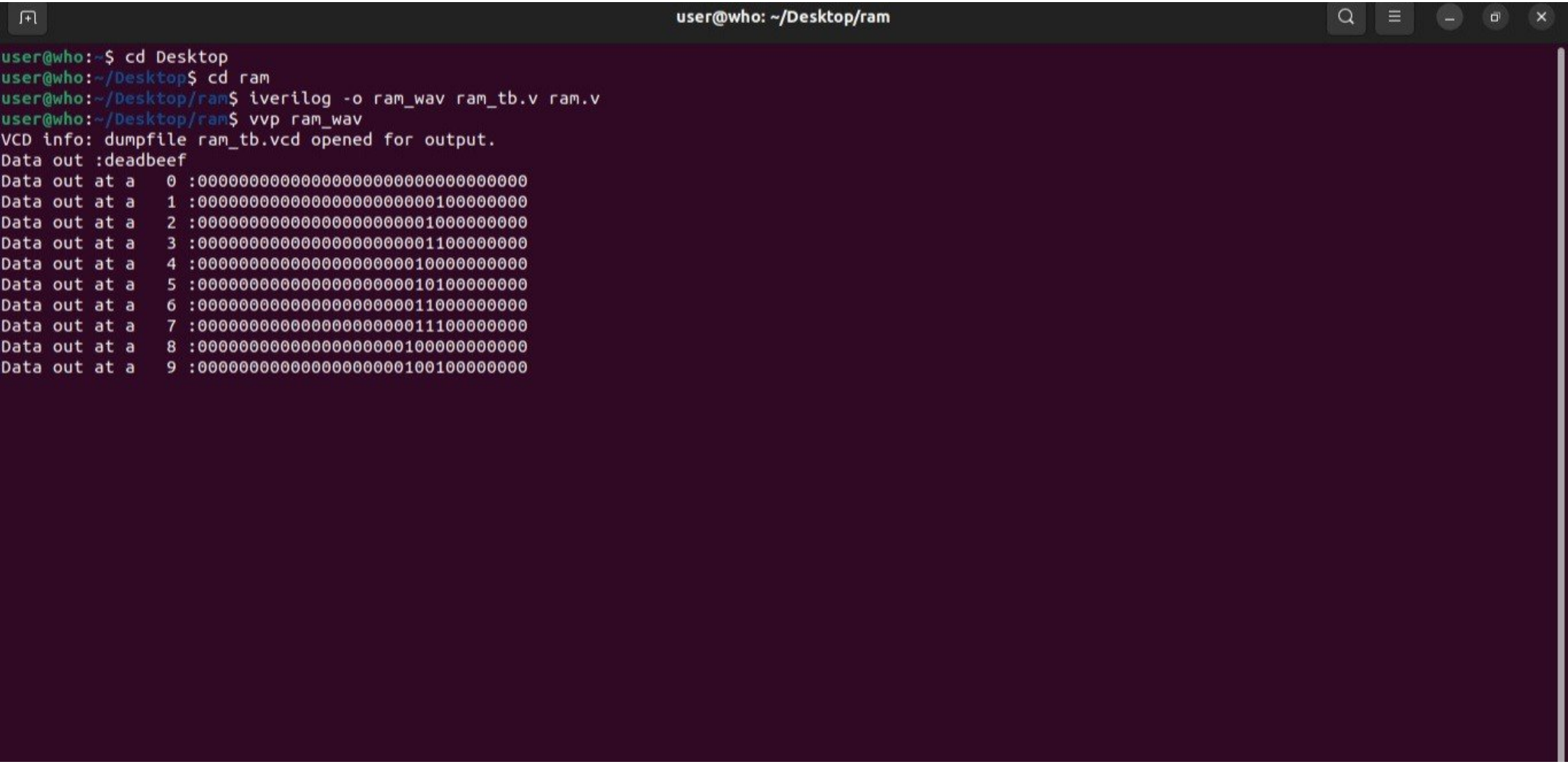
ram_tb.v
~/Desktop/ram

Save

ram.v

ram_tb.v

```
reg w,r;
reg [7:0]a;
reg [31:0]di;
reg clk;
wire [31:0] do;
integer i;
ram u0(
       .a(a),
       .di(di),
       .r(r),
       .w(w),
       .do(do),
       .clk(clk));
       always #50 clk=~clk;
       initial begin
       $dumpfile("ram_tb.vcd");
       $dumpvars(0,ram_tb);
       clk=0;
       a=0;di=0;w=0;r=0;#100;
       w=1;a=8'h10;di=32'hdeadbeef;#100;
       w=0;r=1;a=8'h10;#100;
       $display("Data out :%h",do);
       for(i=0;i<10;i++)
       begin
       w=1;r=0;a=i;
       di=i*32'h100;
       #100;
       w=0;r=1;#100;
       $display("Data out at a %d :%b",a,do);
       end
       end
       endmodule
```


user@who: ~/Desktop/ram
user@who:~$ cd Desktop
user@who:~/Desktop$ cd ram
user@who:~/Desktop/ram$ iverilog -o ram_wav ram_tb.v ram.v
user@who:~/Desktop/ram$ vvp ram_wav
VCD info: dumpfile ram_tb.vcd opened for output.
Data out :deadbeef
Data out at a   0 :00000000000000000000000000000000
Data out at a   1 :00000000000000000000000100000000
Data out at a   2 :00000000000000000000001000000000
Data out at a   3 :00000000000000000000001100000000
Data out at a   4 :00000000000000000000010000000000
Data out at a   5 :00000000000000000000010100000000
Data out at a   6 :00000000000000000000011000000000
Data out at a   7 :00000000000000000000011100000000
Data out at a   8 :00000000000000000000100000000000
Data out at a   9 :00000000000000000000100100000000