**module** instr\_mem(

**input** **logic** [**31**:**0**] addr\_i,

**output** **logic** [**31**:**0**] read\_data\_o

);

**logic** [**31**:**0**] memory [**0**:**1023**];

**initial** **begin**

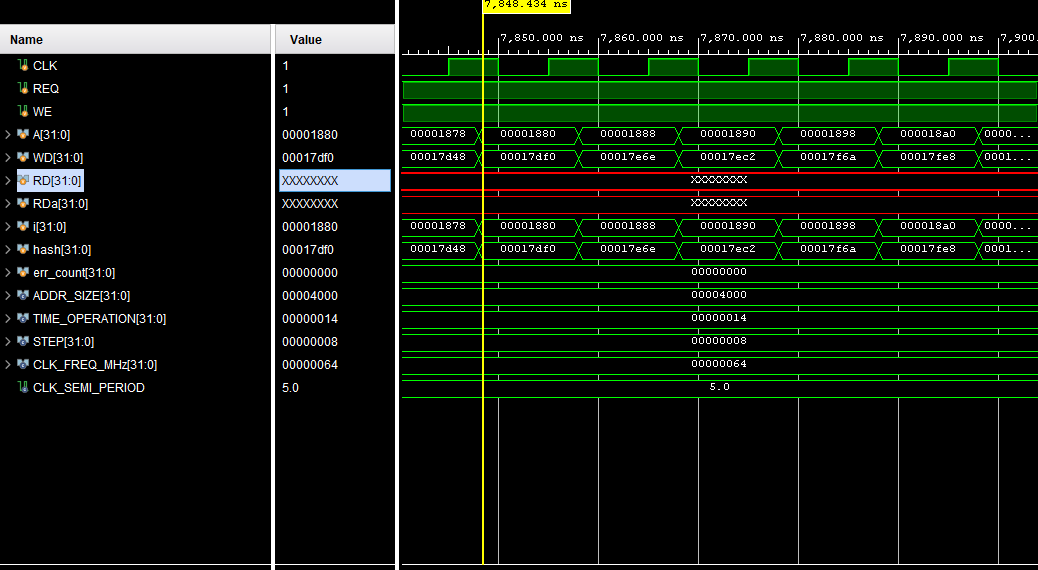
$readmemh("program.mem",memory);

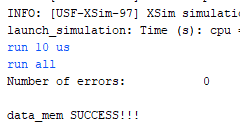
**end**

**assign** read\_data\_o=memory[ addr\_i[**11**:**2**] ];

**endmodule**

**endmodule**





**module** data\_mem(

**input** **logic** clk,

**input** **logic** mem\_req\_i,

**input** **logic** write\_enable\_i,

**input** **logic** [**31**:**0**] addr\_i,

**input** **logic** [**31**:**0**] write\_data\_i,

**output** **logic** [**31**:**0**] read\_data\_o

);

**logic** [**31**:**0**] ram[**0**:**4095**];

**logic** [**13**:**2**]counter;

**logic** [**31**:**0**]data;

**logic** [**1**:**0**] cmd;

**assign** cmd={write\_enable\_i,mem\_req\_i};

**assign** counter=addr\_i[**13**:**2**];

**always\_ff** @(**posedge** clk)

**begin**

**case**(cmd)

**2'b01**: **begin**

read\_data\_o<=ram[counter];

data<=ram[counter];

**end**

**2'b11**: **begin**

read\_data\_o<=ram[counter];

ram[counter]<=write\_data\_i;

data<=ram[counter];

**end**

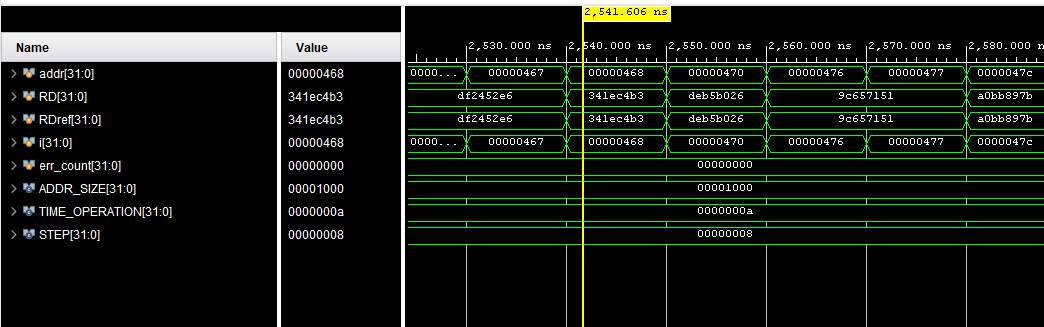
**default**:

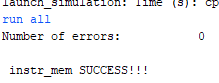
read\_data\_o<=data;

**endcase**

**end**

**endmodule**





**module** rf\_riscv(

**input** **logic** clk,

**input** **logic** [**4**:**0**] addr\_1,

**input** **logic** [**4**:**0**] addr\_2,

**input** **logic** [**4**:**0**] write\_addr,

**input** **logic** write\_enable,

**input** **logic** [**31**:**0**] write\_data,

**output** **logic** [**31**:**0**] read\_data\_1,

**output** **logic** [**31**:**0**] read\_data\_2

);

**logic** [**31**:**0**]rf\_mem[**0**:**31**];

**logic** [**31**:**0**]buf1;

**logic** [**31**:**0**]buf2;

**assign** buf1=rf\_mem[ read\_data\_1 ];

**assign** buf2=rf\_mem[ read\_data\_2 ];

**always\_comb**

**begin**

**case**(read\_data\_1)

**0**: read\_data\_1<=**0**;

**default**:

read\_data\_1<=buf1;

**endcase**

**end**

**always\_comb**

**begin**

**case**(read\_data\_2)

**0**: read\_data\_2<=**0**;

**default**:

read\_data\_2<=buf2;

**endcase**

**end**

**always\_ff** @(**posedge** clk)

**begin**

**if** (write\_enable==**1**)

**begin**

rf\_mem[ write\_addr ]<=write\_data;

**end**

**end**

**endmodule**

\