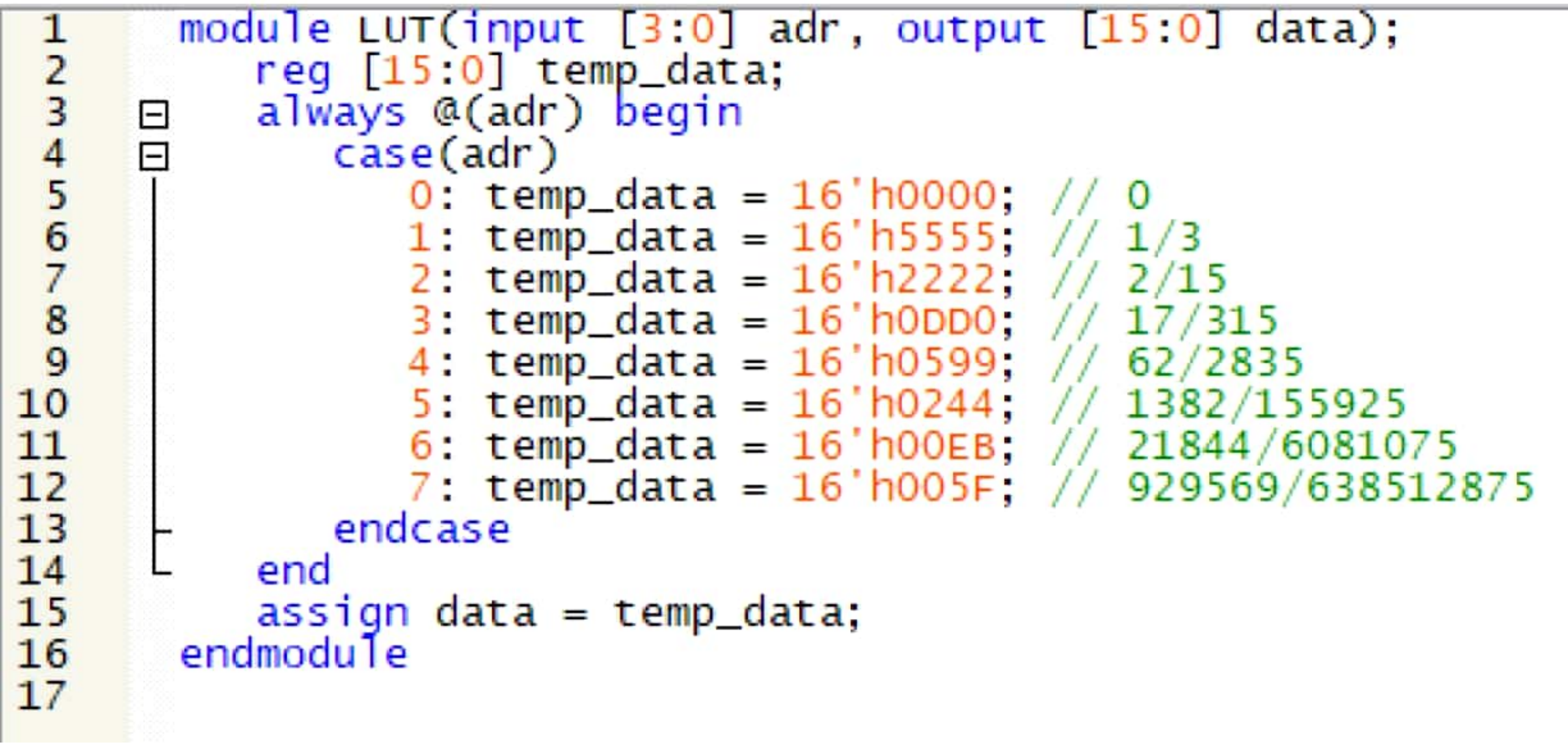
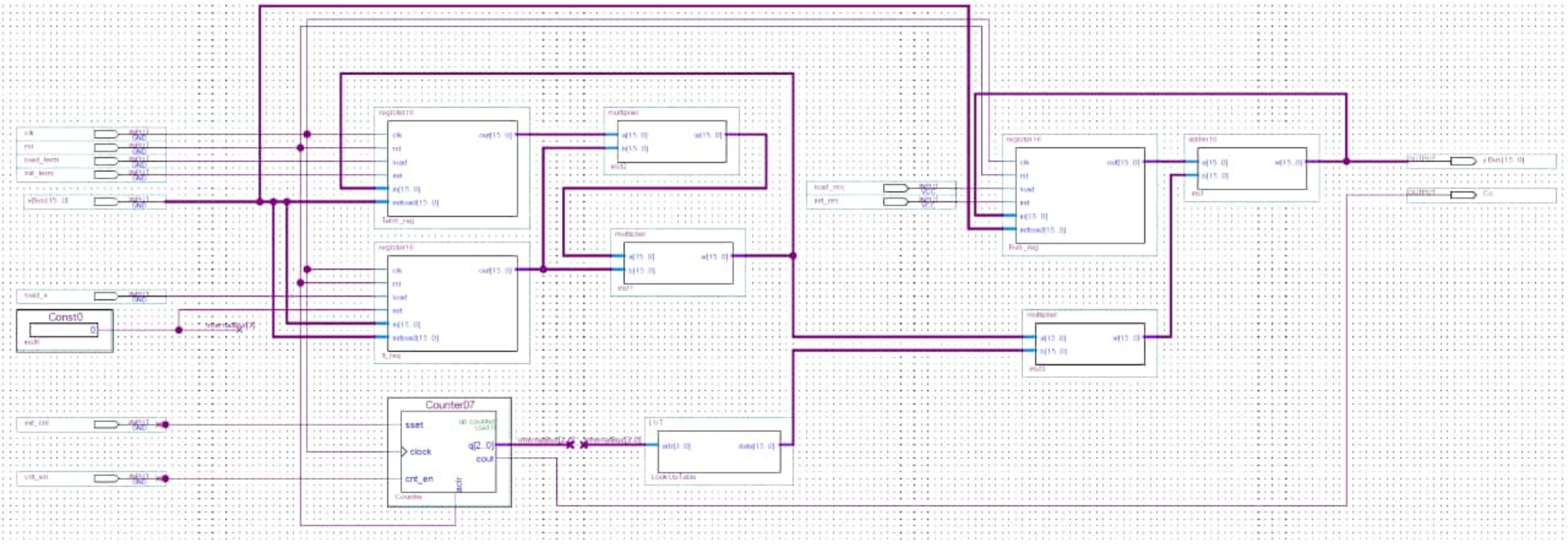
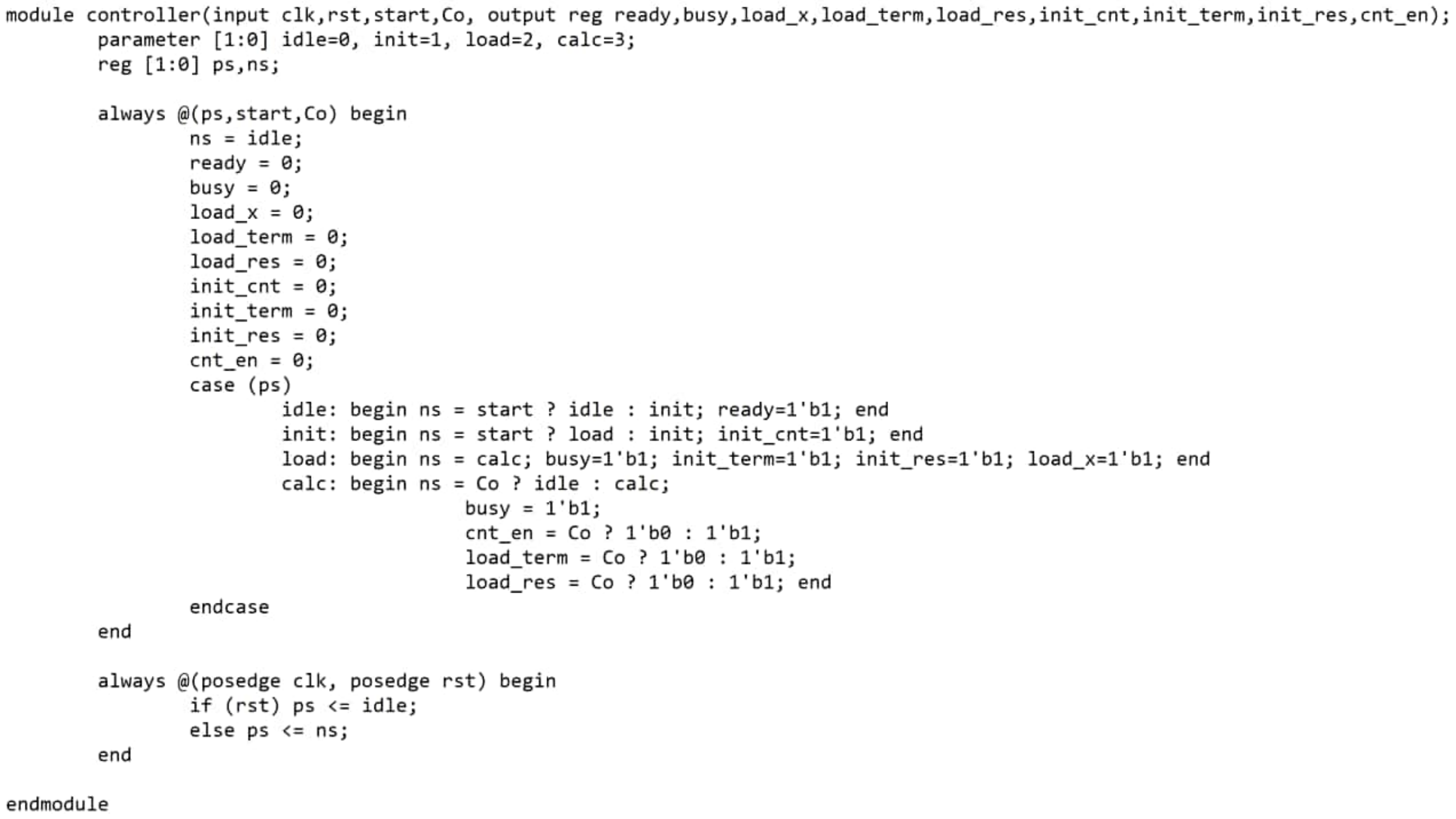
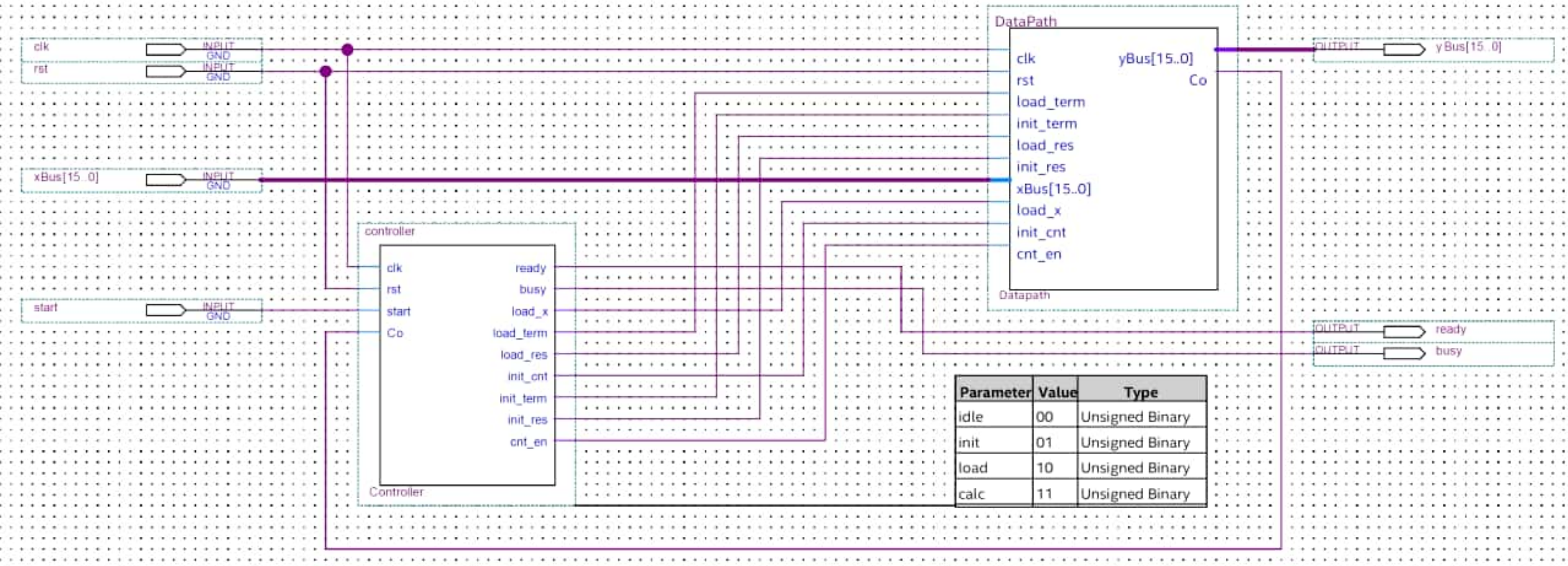


```verilog
module LUT(input [3:0] adr, output [15:0] data);
  reg [15:0] temp_data;
  always @(adr) begin
    case(adr)
      0: temp_data = 16'h0000; // 0
      1: temp_data = 16'h5555; // 1/3
      2: temp_data = 16'h2222; // 2/15
      3: temp_data = 16'h0DD0; // 17/315
      4: temp_data = 16'h0599; // 62/2835
      5: temp_data = 16'h0244; // 1382/155925
      6: temp_data = 16'h00EB; // 21844/6081075
      7: temp_data = 16'h005F; // 929569/638512875
    endcase
  end
  assign data = temp_data;
endmodule
```

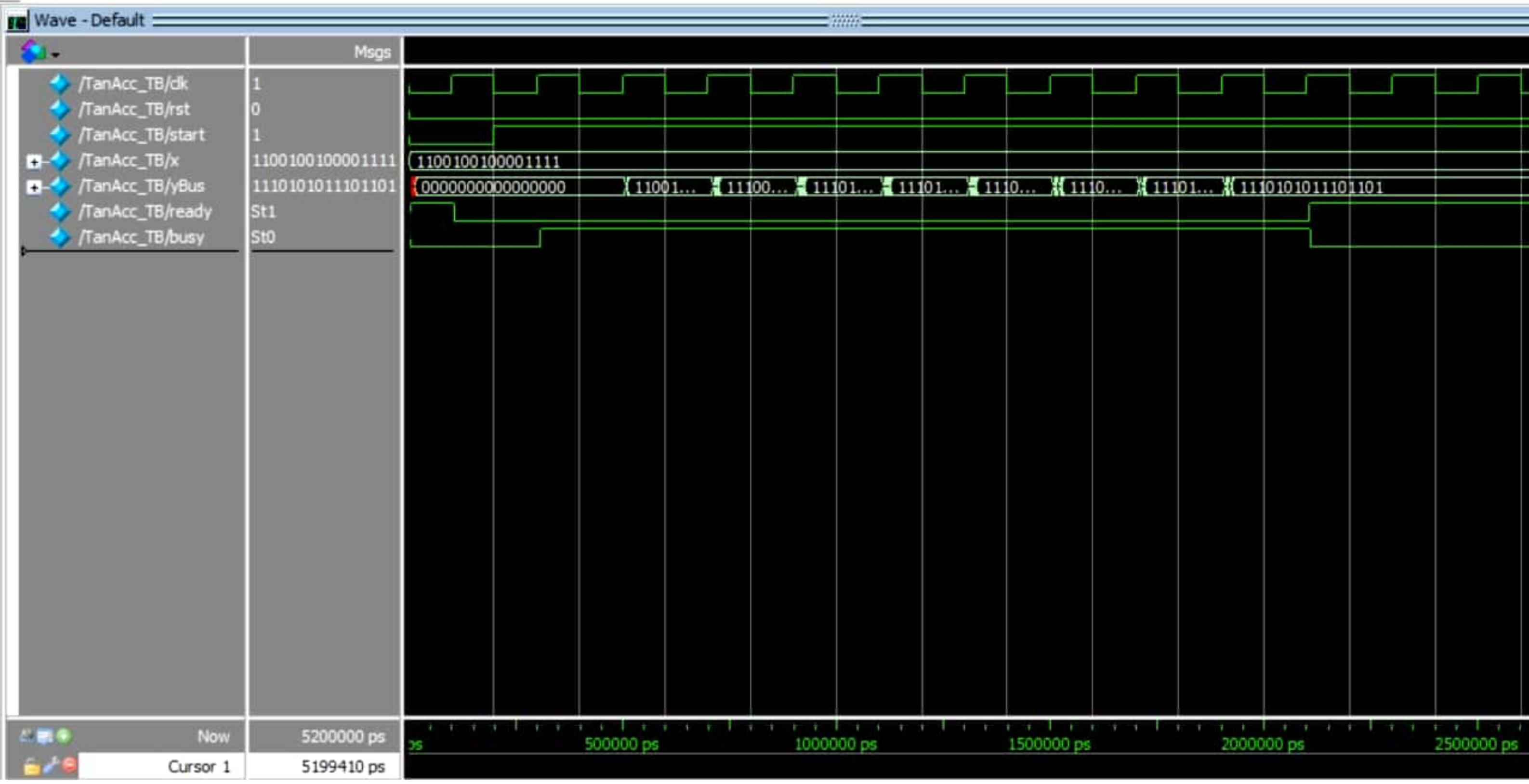

clk
rst
load_term
init_term
xBus[15..0]
load_x
Const0
0
INPUT
GND
register16
clk
rst
load
init
in[15..0]
initload[15..0]
out[15..0]
Term_reg
x_reg
multiplier
a[15..0]
b[15..0]
w[15..0]
inst2
inst1
inst3
Counter07
sset
clock
cnt_en
aclr
q[2..0]
cout
up counter
Counter
LUT
adr[3..0]
data[15..0]
LookUpTable
load_res
init_res
VCC
Res_reg
adder16
inst
OUTPUT
yBus[15..0]
Co



```
module controller(input clk,rst,start,Co, output reg ready,busy,load_x,load_term,load_res,init_cnt,init_term,init_res,cnt_en);
        parameter [1:0] idle=0, init=1, load=2, calc=3;
        reg [1:0] ps,ns;

        always @(ps,start,Co) begin
                ns = idle;
                ready = 0;
                busy = 0;
                load_x = 0;
                load_term = 0;
                load_res = 0;
                init_cnt = 0;
                init_term = 0;
                init_res = 0;
                cnt_en = 0;
                case (ps)
                        idle: begin ns = start ? idle : init; ready=1'b1; end
                        init: begin ns = start ? load : init; init_cnt=1'b1; end
                        load: begin ns = calc; busy=1'b1; init_term=1'b1; init_res=1'b1; load_x=1'b1; end
                        calc: begin ns = Co ? idle : calc;
                                        busy = 1'b1;
                                        cnt_en = Co ? 1'b0 : 1'b1;
                                        load_term = Co ? 1'b0 : 1'b1;
                                        load_res = Co ? 1'b0 : 1'b1; end
                endcase
        end

        always @(posedge clk, posedge rst) begin
                if (rst) ps <= idle;
                else ps <= ns;
        end

endmodule
```

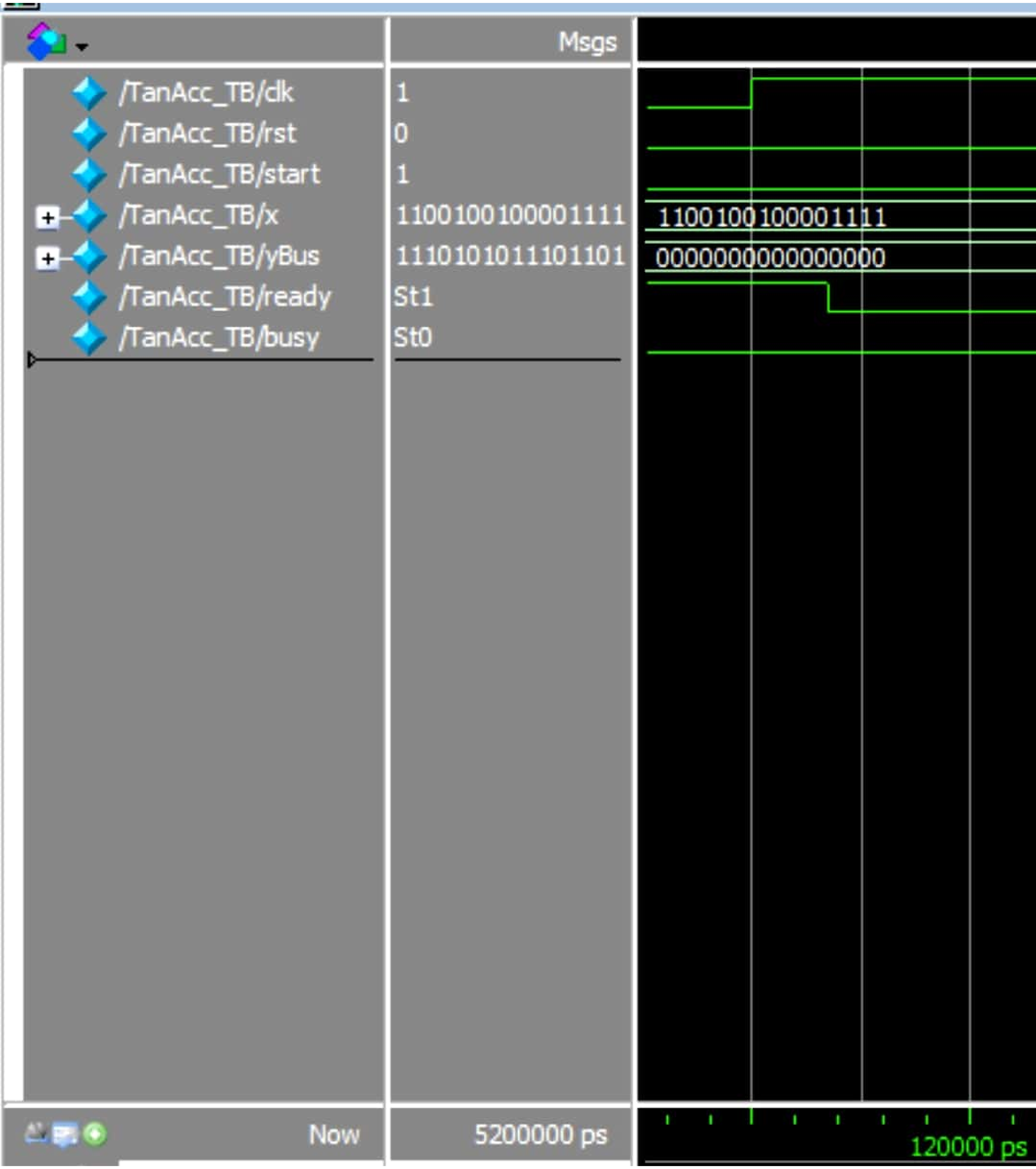


Parameter	Value	Type
idle	00	Unsigned Binary
init	01	Unsigned Binary
load	10	Unsigned Binary
calc	11	Unsigned Binary



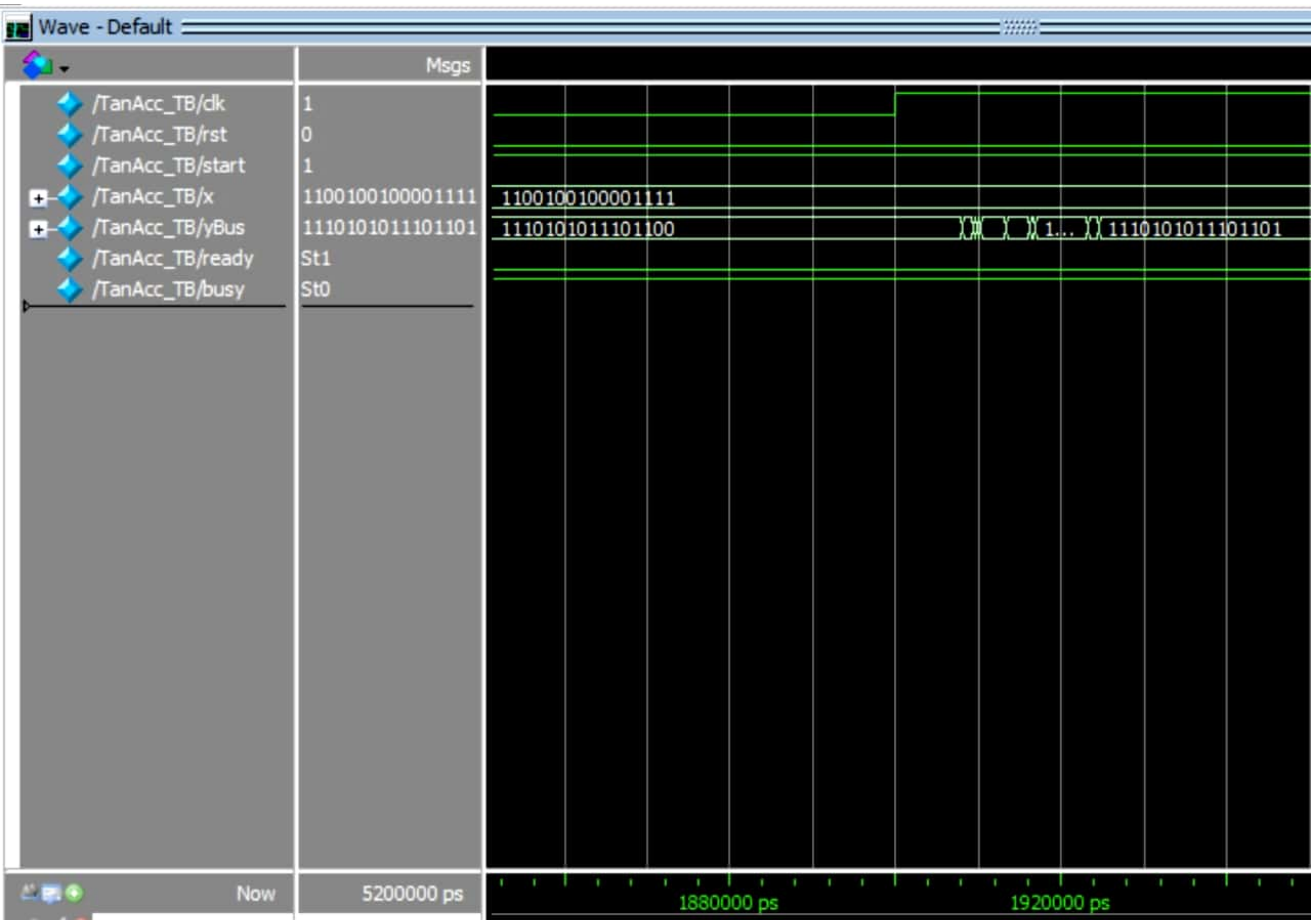
Wave - Default

	Msgs
/TanAcc_TB/clk	1
/TanAcc_TB/rst	0
/TanAcc_TB/start	1
/TanAcc_TB/x	1100100100001111
/TanAcc_TB/yBus	1110101011101101
/TanAcc_TB/ready	St1
/TanAcc_TB/busy	St0
Now	5200000 ps
Cursor 1	5199410 ps



	Msgs
/TanAcc_TB/clk	1
/TanAcc_TB/rst	0
/TanAcc_TB/start	1
/TanAcc_TB/x	1100100100001111
/TanAcc_TB/yBus	1110101011101101
/TanAcc_TB/ready	St1
/TanAcc_TB/busy	St0

Scanned by CamScanner



Wave - Default

```
                        Msgs
/TanAcc_TB/clk          1
/TanAcc_TB/rst          0
/TanAcc_TB/start        1
/TanAcc_TB/x            1100100100001111
/TanAcc_TB/yBus         1110101011101101
/TanAcc_TB/ready        St1
/TanAcc_TB/busy         St0

Now                     5200000 ps
```