**PERANCANGAN SISTEM DIGITAL**

NAME: MUHAMMAD AIMAN BIN ABD AZIS

NPM : 41155045160075

COURSE: TEKNIK ELEKTRO



JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

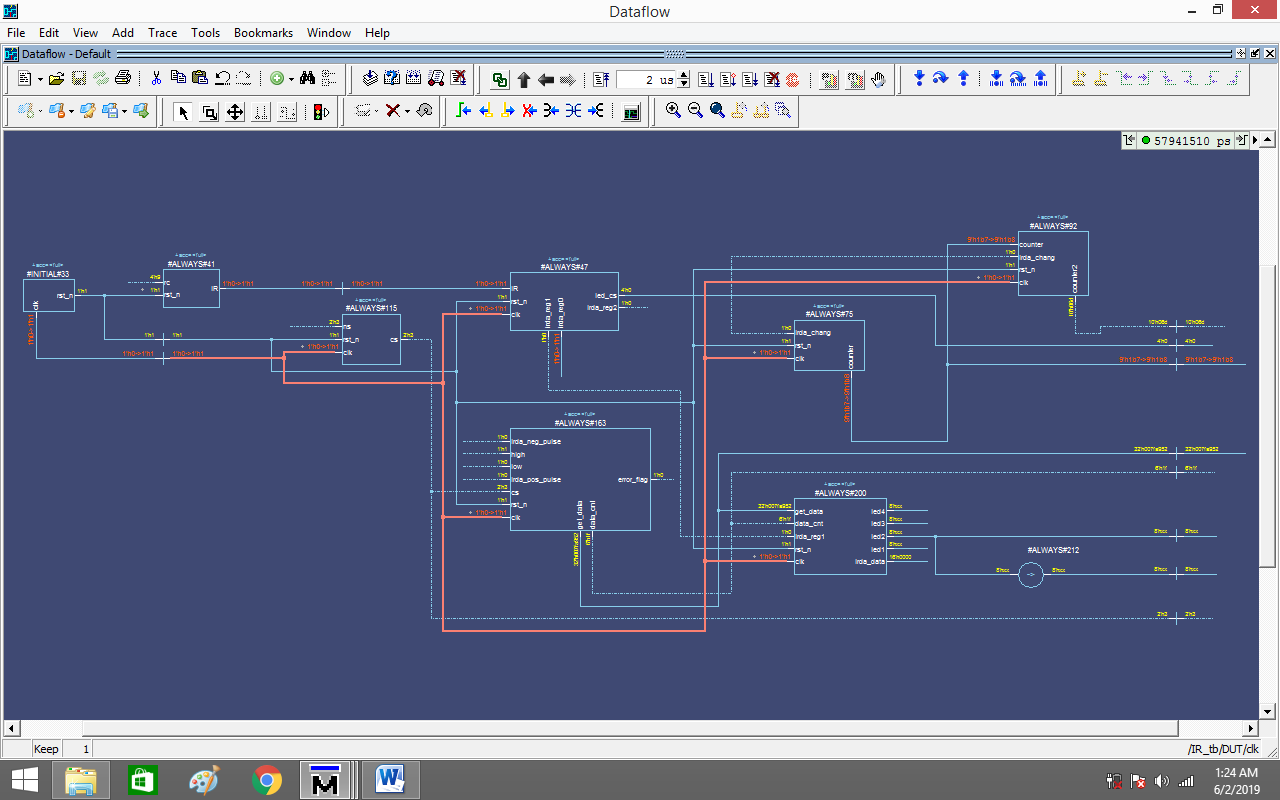
UNVERSITAS LANGLANGBUANA

BANDUNG

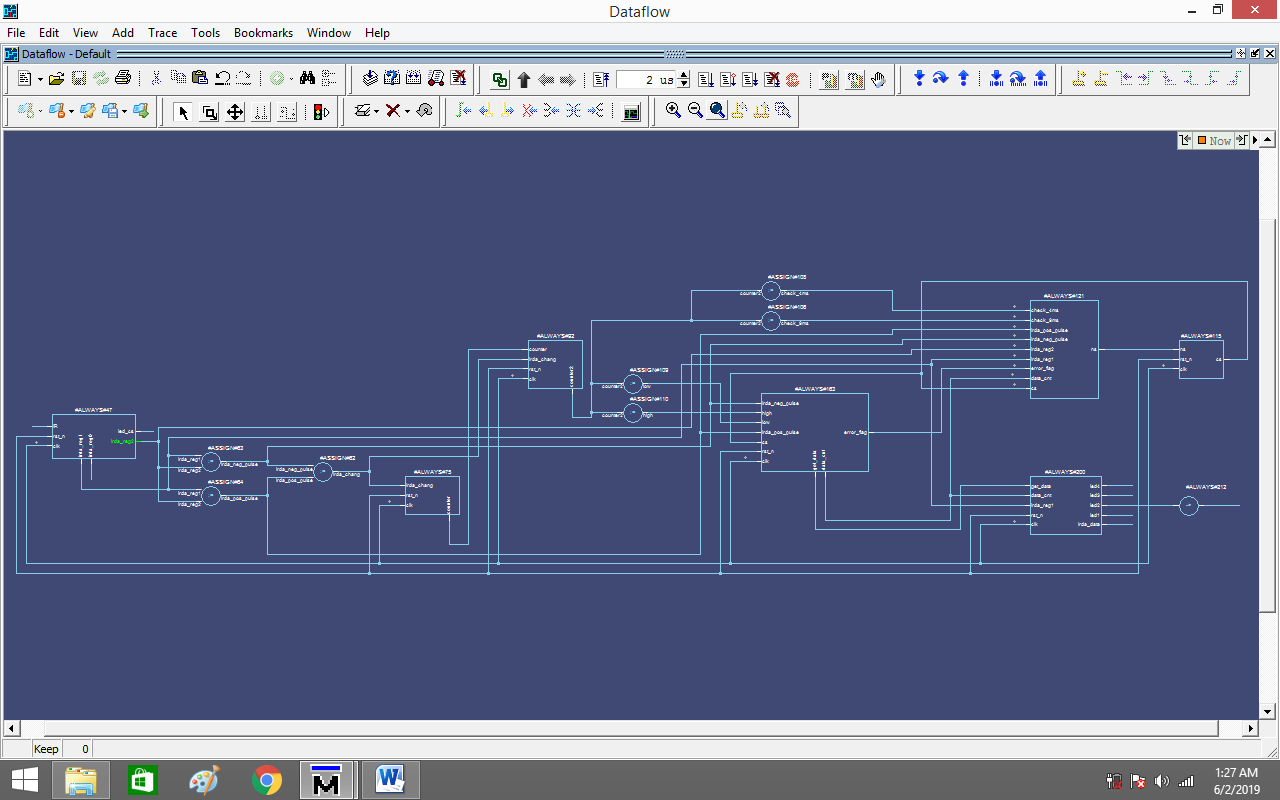
DISEMBER 2019

**IR REMOTE CONTROL interface with 8051 based on STANDARD PROTOCOL NEC**

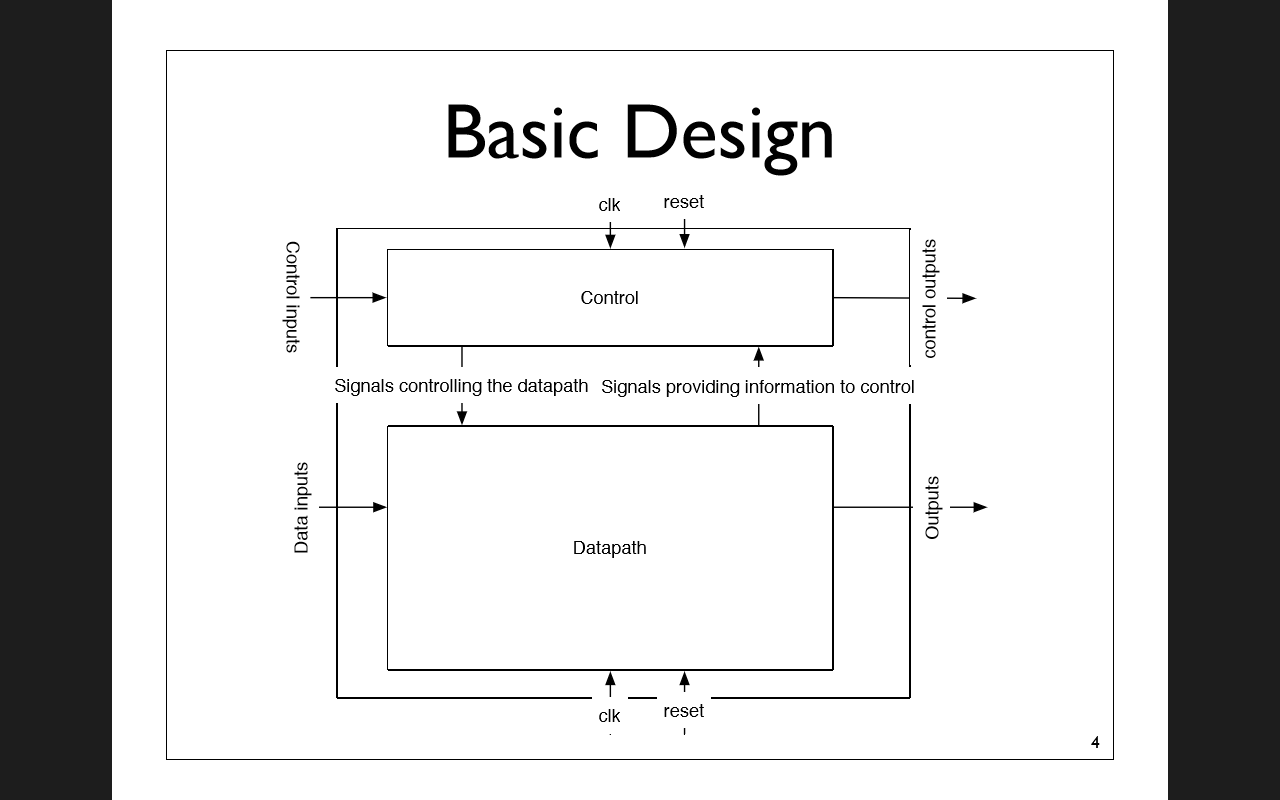
**A. Data path and Control path**



*Picture 1: Dataflow for IR\_tb*



*Picture 2: Dataflow for IR*



*Picture 3: Basic Design for Datapath and Controlpath*

The datapath is where data moves from place to place.

• Computation happens in the datapath

• No decisions are made here.

• Things you should ﬁnd in a datapath

• Muxes

• Registers

• ALUs

• Wide busses (34 bits for data. 17 bits for instructions)

• These components are physically large

• In a real machine, their spatial relationship are important.

Control is where decisions are made

• Things you will there

• State machines

• Random lots of complex logic

• Little state (maybe just a single register)

• Spatial relationships are harder to reason about or exploit.

**B. ModelSim verilog coding**

**At line 4**

module IR (clk,rst\_n,IR,rc ,led\_cs,led\_db,cs,counter,counter2,data\_cnt,led2,get\_data);

-The coding of IR module start with input and output desired variable.

- For each input will be declare as wire in module while reg in testbench respectively.

- For each output will be declare as reg in module while wire in testbench respectively.

**At line 56**

always @ (posedge clk) **//triggers the block on the positive (rising) edge of a clock signal**

if(!rst\_n) **//(rst\_n = 0 is true, rst\_n = 1 is false)**

begin

irda\_reg0 <= 1'b0;**//initialize bit register**

irda\_reg1 <= 1'b0;**//initialize bit register**

irda\_reg2 <= 1'b0;**//initialize bit register**

end

else

begin

led\_cs = 4'b0000;

irda\_reg0 <= IR;

irda\_reg1 <= irda\_reg0;

irda\_reg2 <= irda\_reg1;

end

* irda\_reg is 3 bit shift register
* irda\_reg will non-blocking assignment which is to describe sequential logic
* irda\_reg execute in parallel for each line

**At line 62**

assign irda\_chang = (irda\_pos\_pulse|irda\_neg\_pulse); // **This line is to detect when IR have any change in logic value**

assign irda\_neg\_pulse = (irda\_reg2 & (~irda\_reg1)); //**This line is to detect when IR have change from 1 to 0 in logic value**

assign irda\_pos\_pulse = ((~irda\_reg2) & irda\_reg1); **// This line is to detect when IR have change from 0 to 1 in logic value**

**At line 75**

always @ (posedge clk)

if (!rst\_n)

begin

counter <= 9'b0;

end

else if (irda\_chang) //**When irda\_chang is true, counter will turn to 0**

begin

counter <= 9'b0;

end

else if (counter == 9'b111111111) // **When counter = 9'b111111111, counter will turn to 0**

begin

counter <= 9'b0;

end

else//**Other than that, counter will up by 1 bit per 1 cyle clock**

counter <= counter + 9'b1;

//----------------------------------------------------------------------------

always @ (posedge clk)

if (!rst\_n)

begin

counter2 <= 10'b0;

end

else if (irda\_chang)

begin

counter2 <= 10'b0;

end

else if (counter == 9'b111111111)//**Only when counter count to 511 times, the counter2 will goes up by 1 bit.**

Begin

counter2 <= counter2 +10'b1;

end

**At line 113**

assign check\_9ms = ((868 < counter2) & (counter2 < 898));

**//878 = counter2 will assign check\_9ms as true**

assign check\_4ms = ((429 < counter2) & (counter2 < 449));

**//439 = counter2 will assign check\_4ms as true**

assign low = ((26 < counter2) & (counter2 < 56));

**//35 = counter2 will assign low as true**

assign high = ((154 < counter2) & (counter2 < 174));

**//164 = counter2 will assign high as true**

**At line 178**

else if (cs == DATA\_STATE) //**If cs is DATA\_STATE then it is true**

if (irda\_pos\_pulse) // **If irda\_pos\_pulse = 1 then it is true**

begin

if (low) // **low is true**

begin

get\_data <= get\_data << 1; //**left shift register**

get\_data[0] <= 1'b0; // **1’b0 will shift into lsb**

data\_cnt <= data\_cnt + 1'b1; // **data\_cnt goes up by 1-bit**

end

else if (high) // **high is true**

begin

get\_data <= get\_data << 1; //**left shift register**

get\_data[0] <= 1'b1; // **1’b0 will shift into lsb**

data\_cnt <= data\_cnt + 1'b1; // **data\_cnt goes up by 1-bit**

end

end

else if (irda\_neg\_pulse) // **If irda\_neg\_pulse = 1 then it is true**

begin

if (!low) //**if low = 0,then it is true**

error\_flag<=1'b1;

end

**At line 203**

else if ((data\_cnt ==6'd32) & irda\_reg1) // **If data\_cnt decimal value is 32 and irda\_reg1 binary value is 1, then it is true**

begin

led1 <= get\_data[7:0]; // **Command(Logical inverse)**

led2 <= get\_data[15:8]; // **Command that will determine output led\_db value**

led3 <= get\_data[23:16];// **Address(Logical inverse)**

led4 <= get\_data[31:24];// **Address**

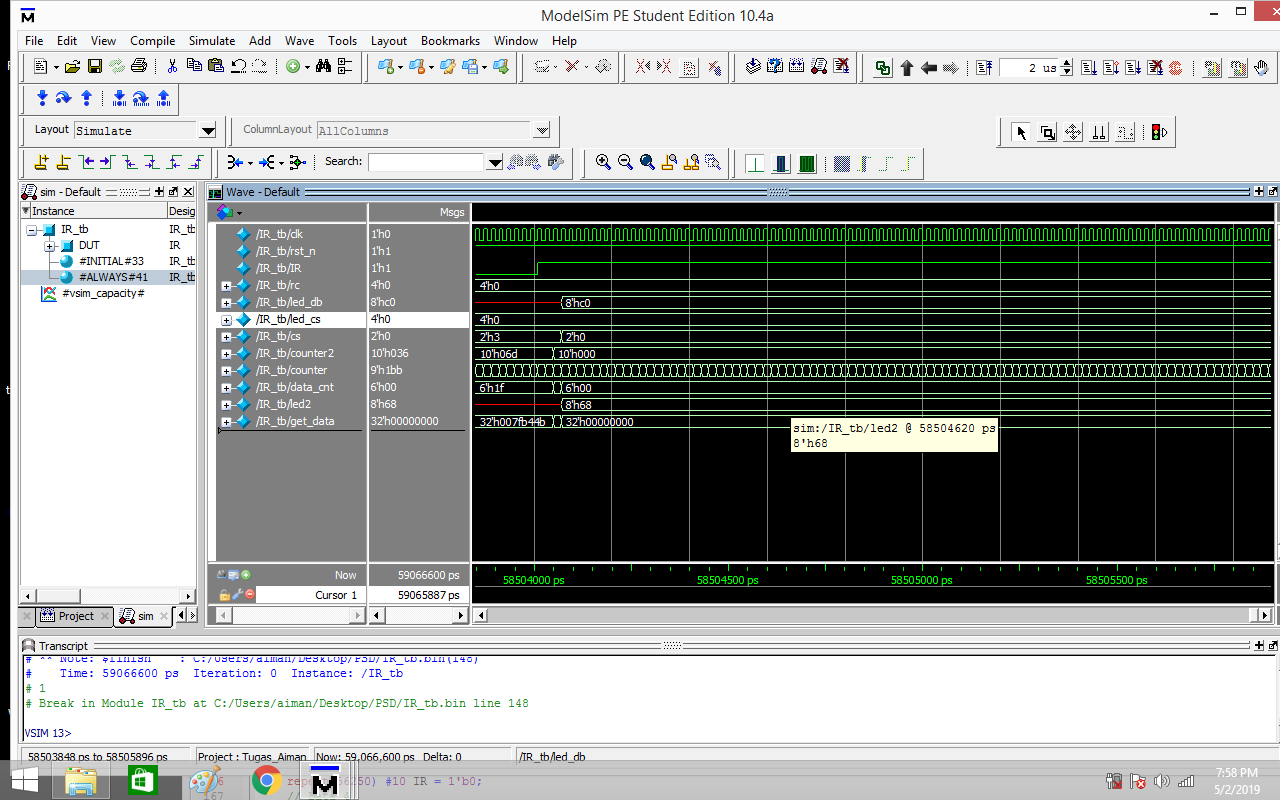
end

In file IR\_tb.v

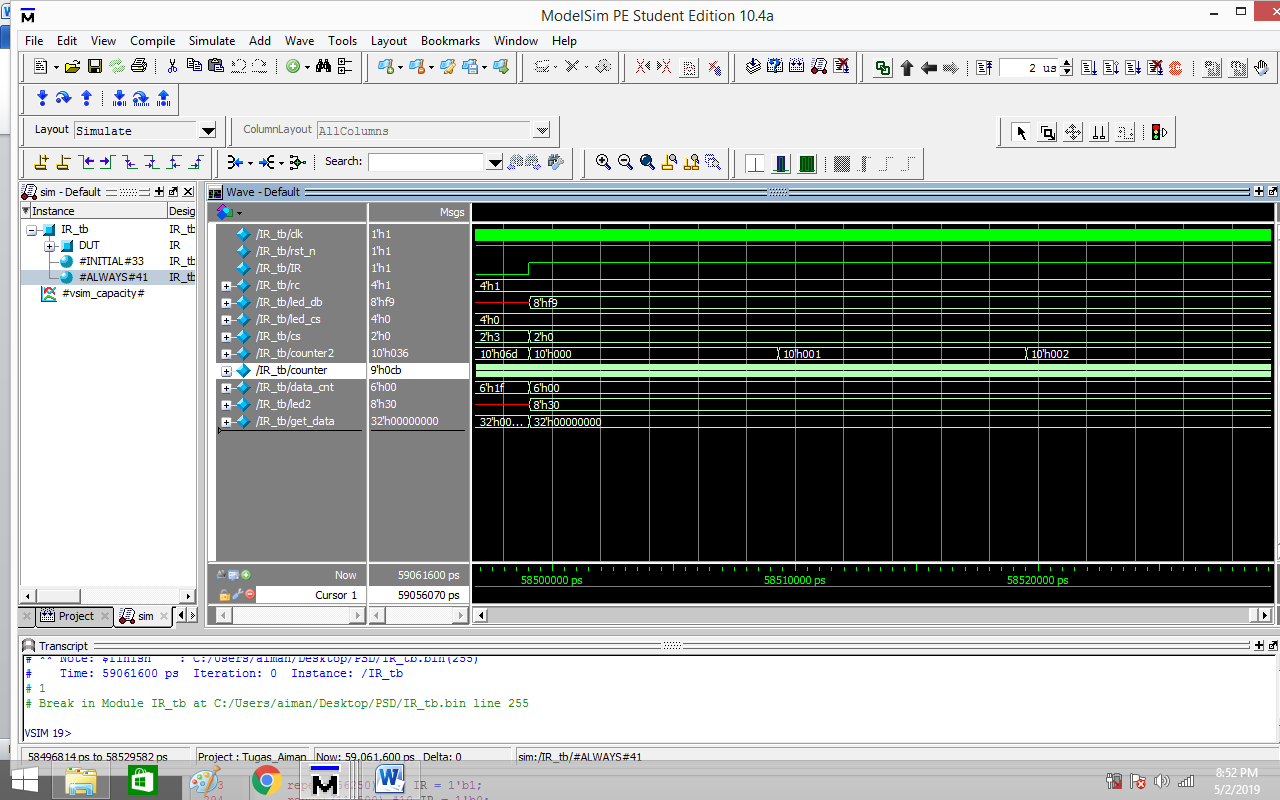
* Reg and wire will be shown in simulation wave
* DUT means Device Under Test
* In DUT is for sensitive list, could be seen in IR.v which is always(\*)
* Clock cyle is 20ps and reset start as 0 for 10ps and 1 for the rest of simulation
* rc value must be force input first to give an information which IR signal will we use, for example rc = 4’h0 is for 0 in seven-segment

**C. Simulation using Model Sim for input remote switch from 0 until 9**

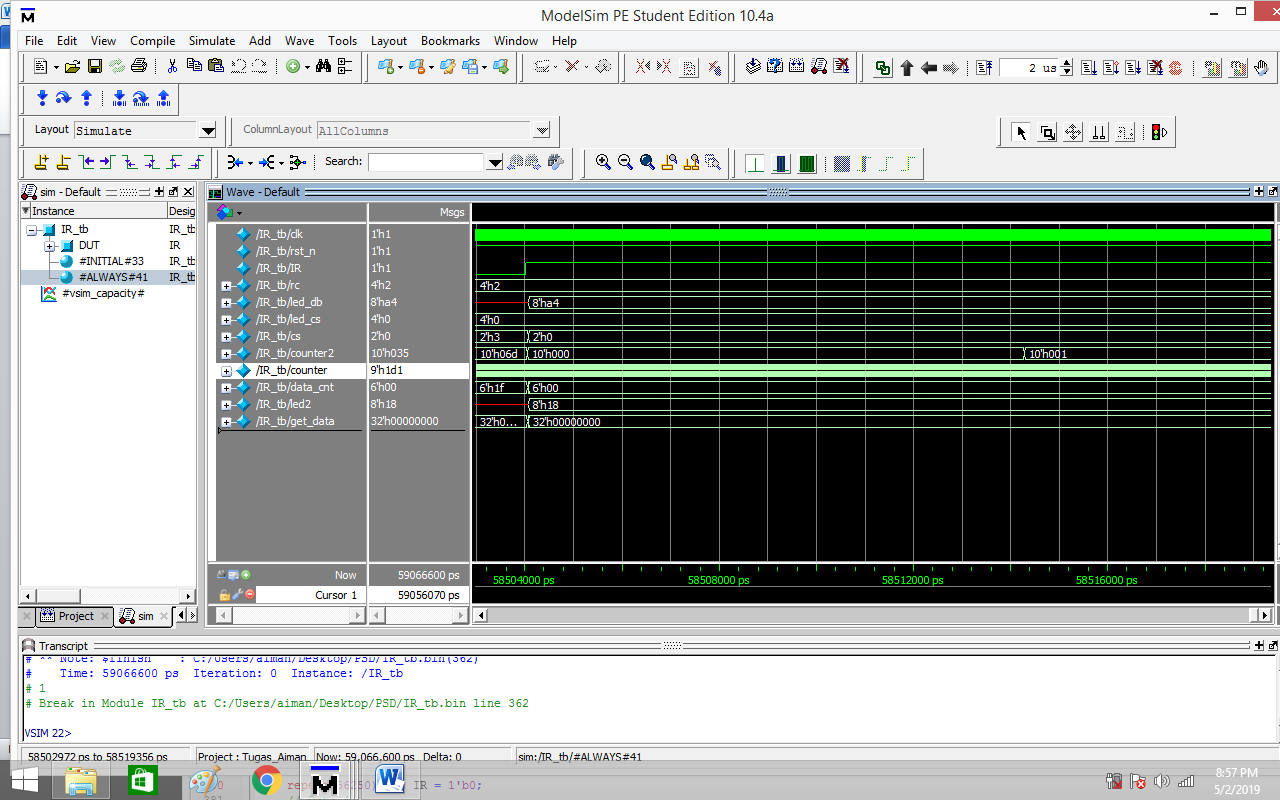
RESULTS FOR INFRARED SIGNAL OF REMOTE CONTROL



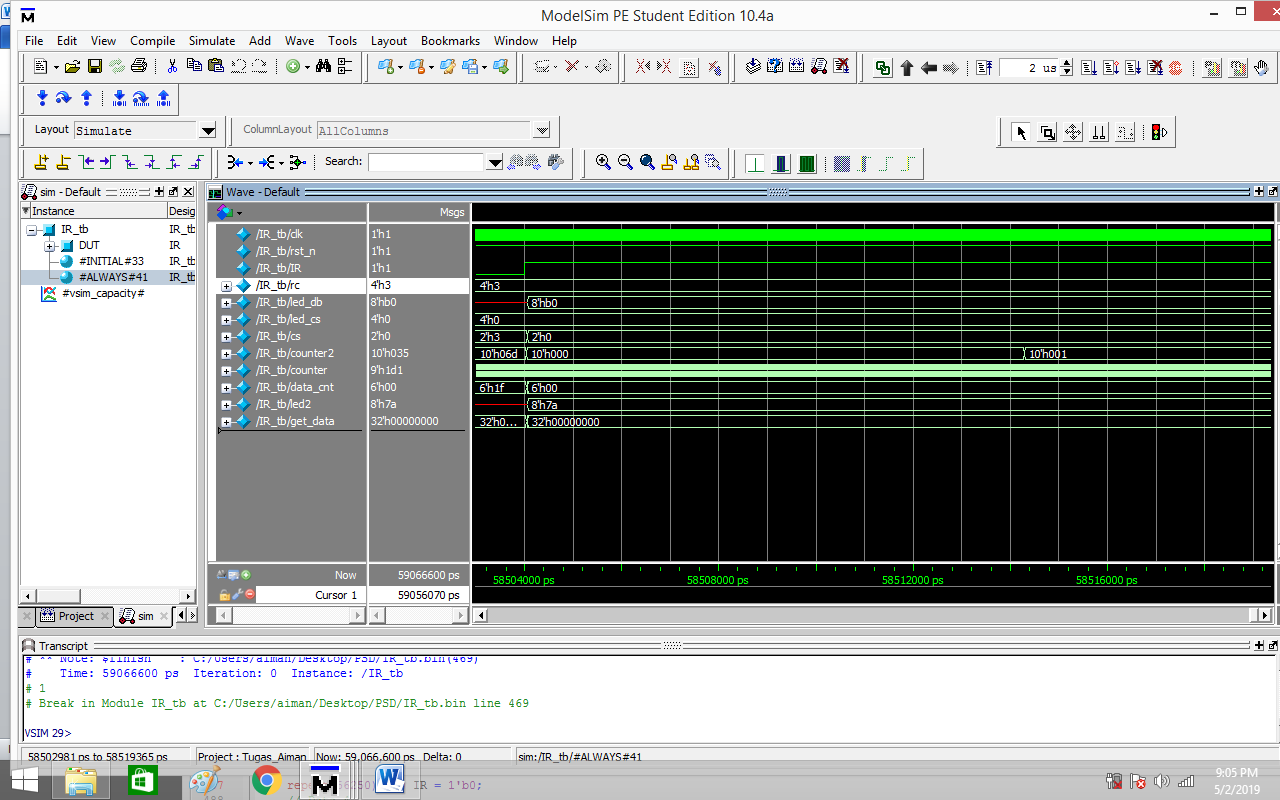
*Figure 1: rc = 4’d0, led2 = 8’h68, led\_db = 8’hc0*

**

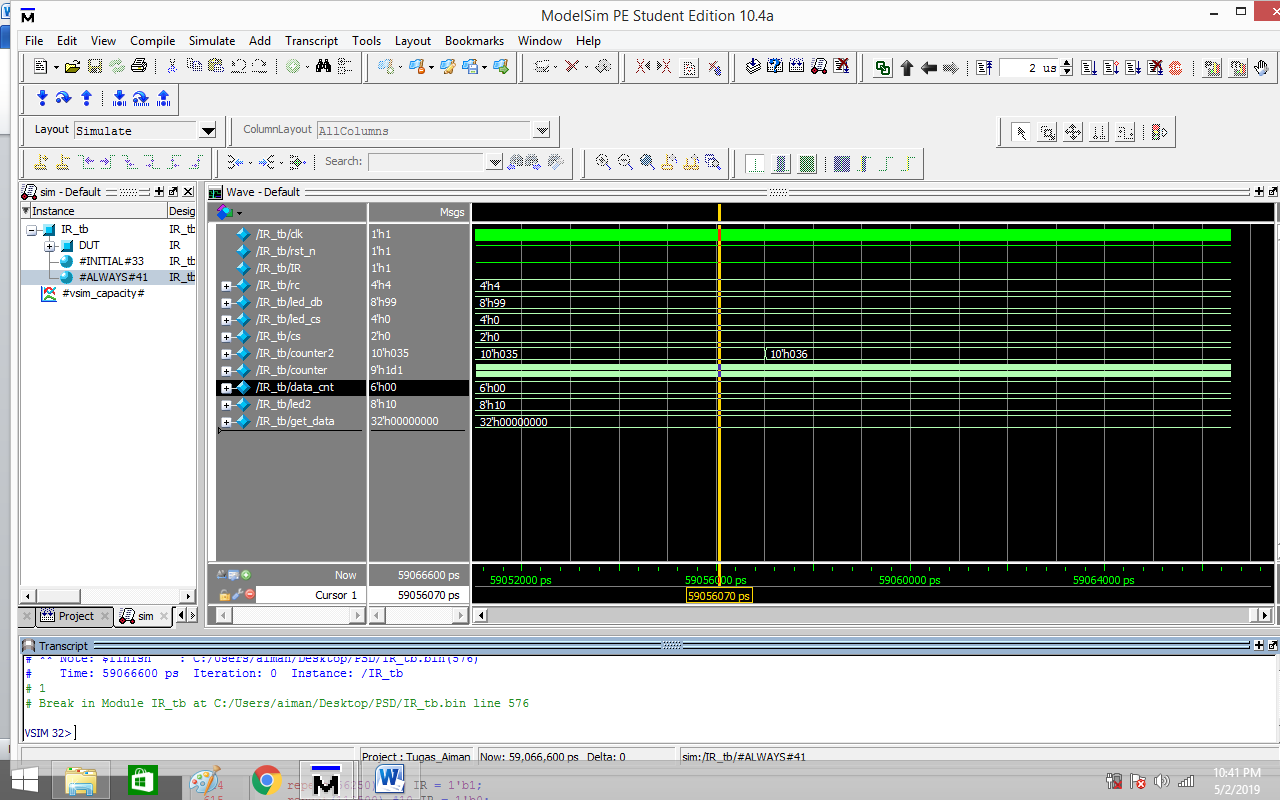
*Figure 2: rc = 4’h1, led2 = 8’h30, led\_db = 8’hf9*

**

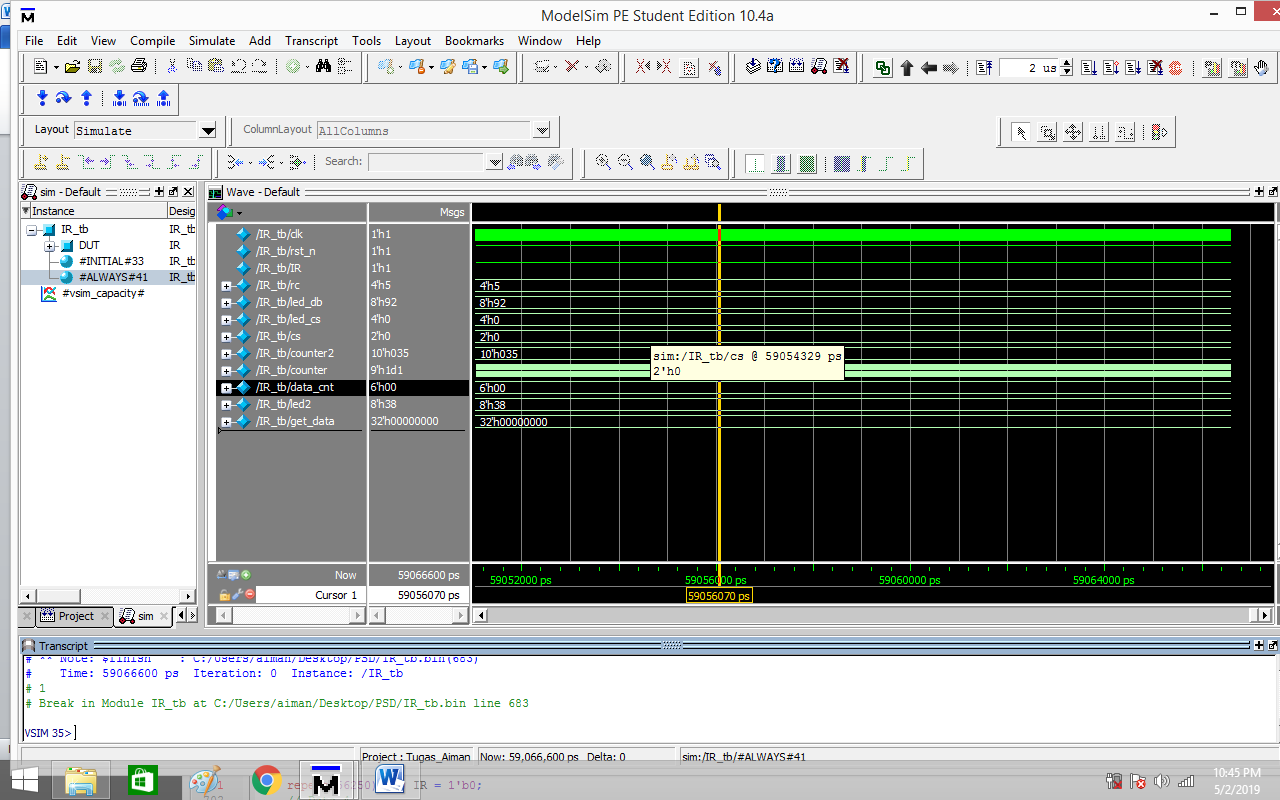
*Figure 3: rc = 4’h2, led2 = 8’h18, led\_db = 8’ha4*

**

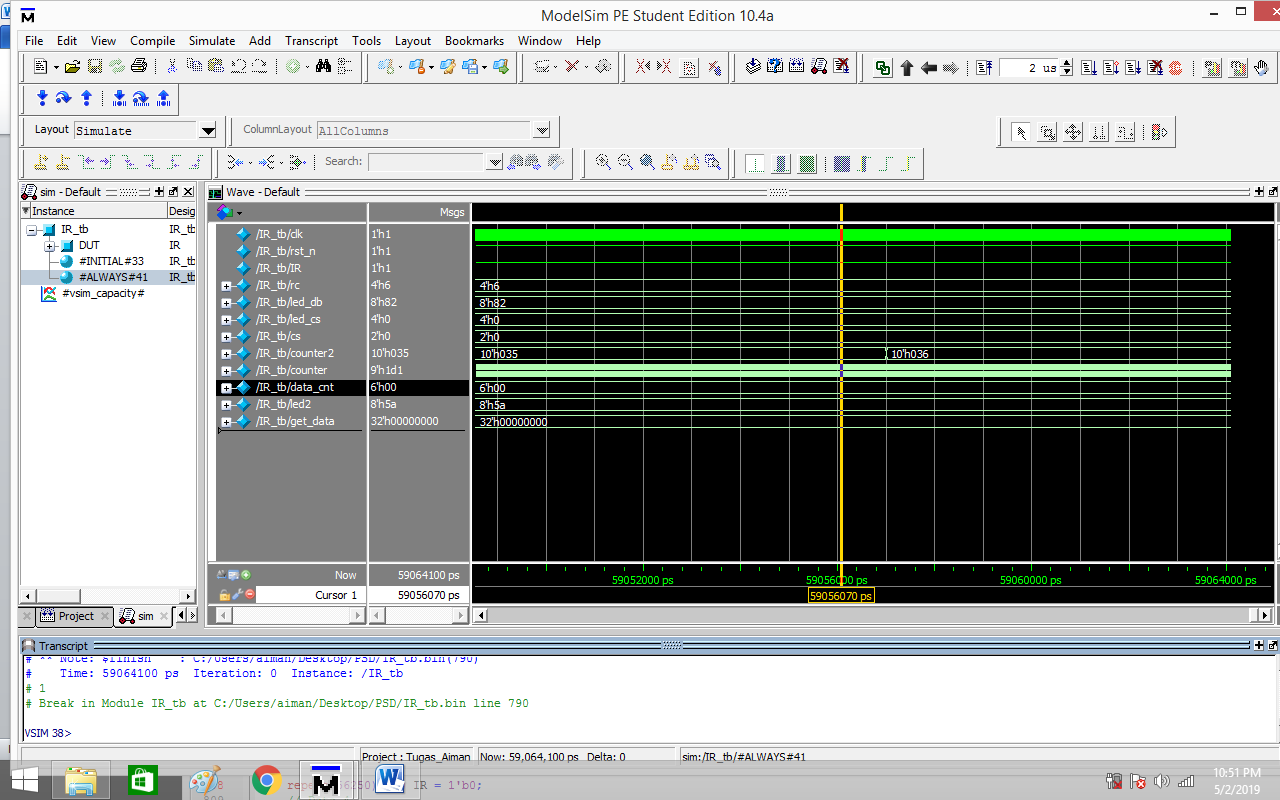
*Figure 4: rc = 4’h3, led2 = 8’h7a, led\_db = 8’hb0*

**

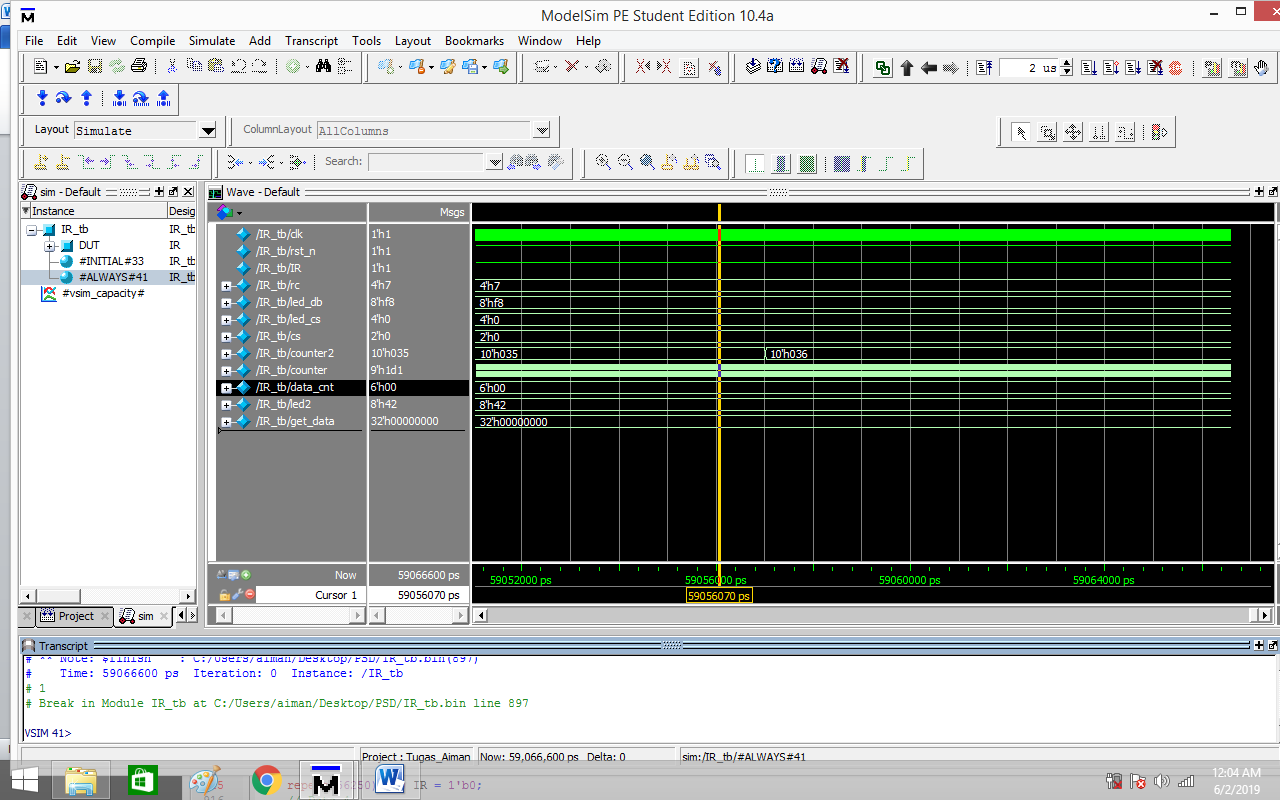
*Figure 5: r4 = 4’h4, led2 = 8’h10, led\_db = 8’h99*

**

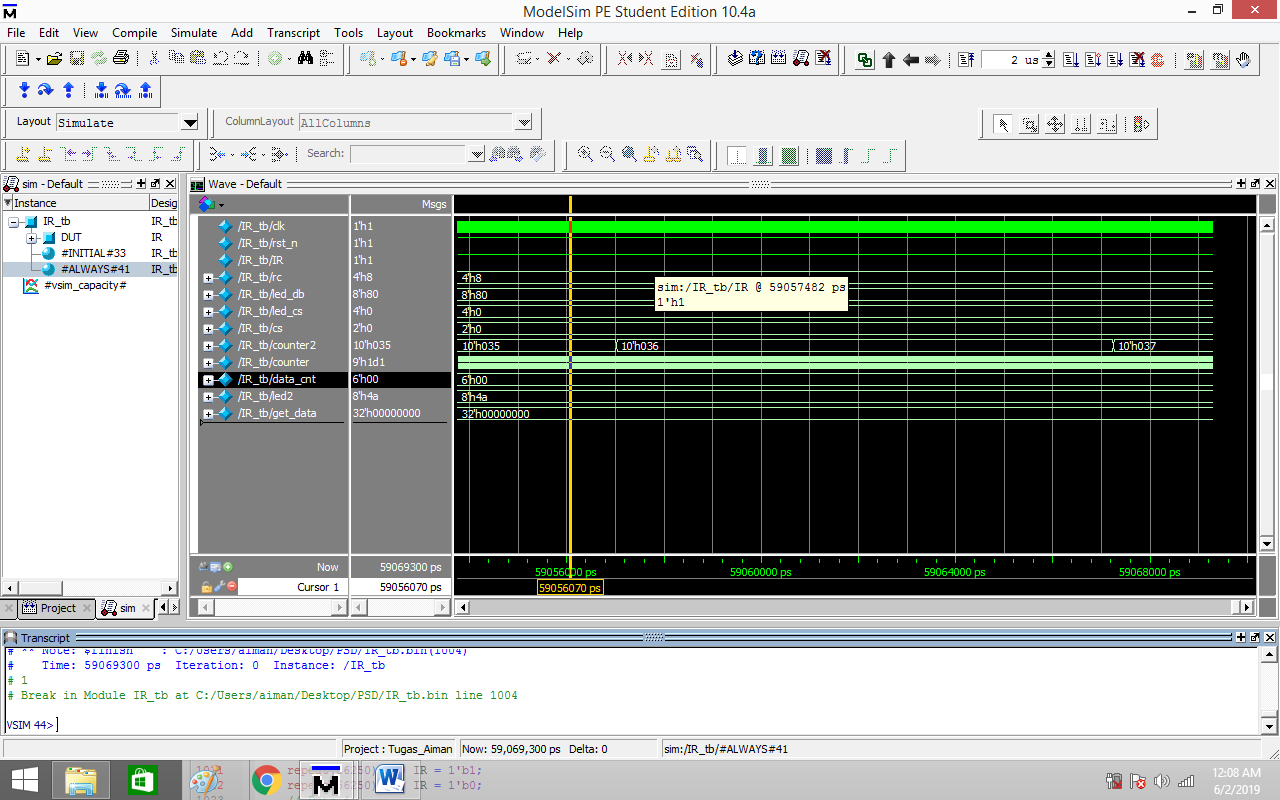
*Figure 6: rc = 4’h5, led2 = 8’h38, led\_db = 8’h92*

**

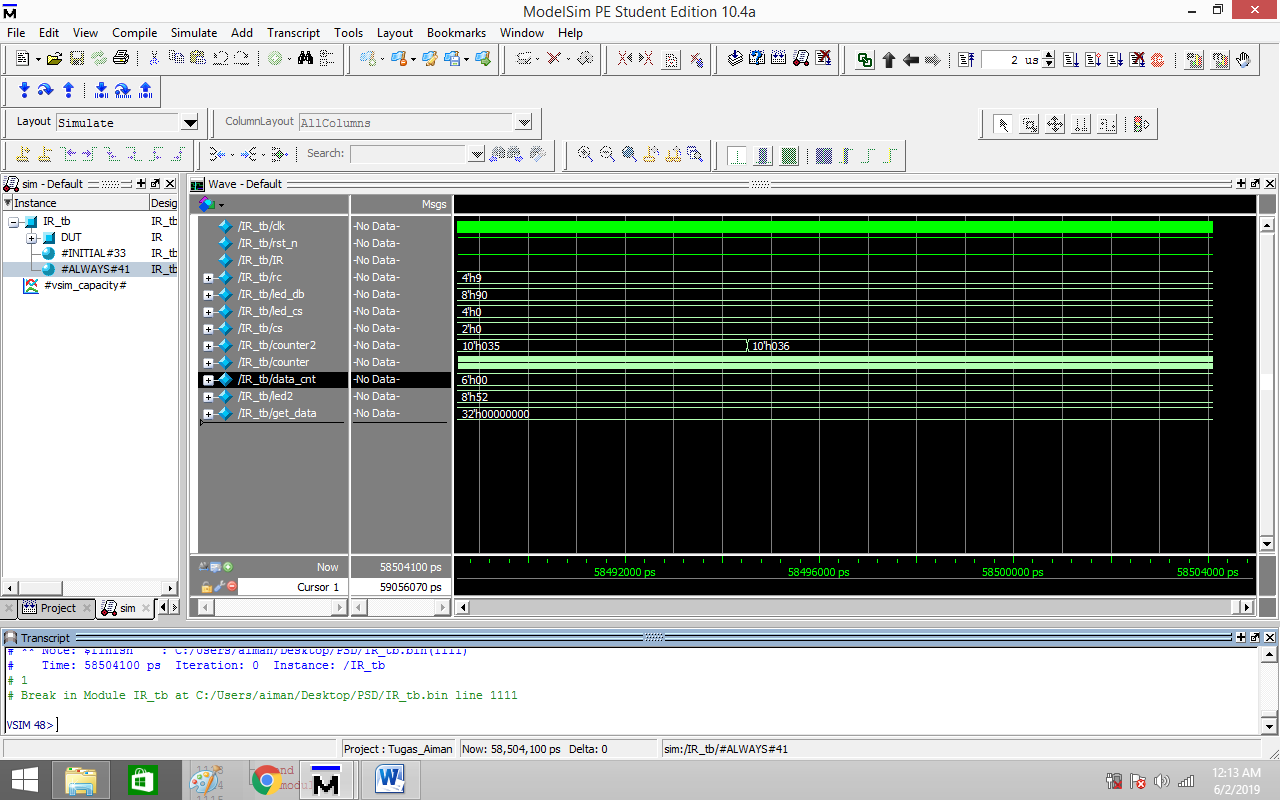
*Figure 7: rc = 4’h6, led2 = 8’h5a, led\_db = 8’h82*

**

*Figure 8: rc7 = 4’h7, led2 = 8’h42, led\_db = 8’hf8*

**

*Figure 9: rc8 = 4’h8, led2 = 8’h4a, led\_db = 8’h80*

**

*Figure 10: rc9 = 1h’1, led2 = 8h’52, led\_db = 8h’90*