**Project Design**

I prepare to design a verification module which can be inserted into the designer's module. In order to facilitate a quick analysis, it is therefore necessary to do it in a host. The data is downloaded to the FPGA to start testing according to the instructions of the host.

**Design Diagram**

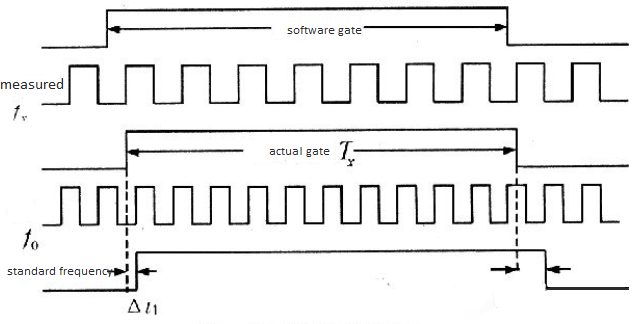
The design diagram is as shown below.



It will describe the four modules in details in the followings.

1. Frequency measurement module

Equal precision frequency measurement principle is as shown below.



According to the principle of equal precision frequency measurement, the coding diagram is as follows.

 The pre-gate coding is shown below.

always @(posedge I\_sys\_clk or negedge I\_rst\_n) begin

     if(!I\_rst\_n) begin

         gate\_cnt <= 32'd0;

         R\_done\_pre <= 0;

         gate\_[**ti**](https://bbs.elecfans.com/zhuti_715_1.html)me\_pre <= 0;

         R\_start <= 0;

     end

     else begin

         if(W\_start\_pos) begin

             R\_start <= 1;

         end

         if(R\_start) begin

             if(gate\_cnt == SET\_1S\_PERD) begin

                 gate\_time\_pre <= 0;

                 R\_done\_pre <= 1;

                 R\_start <= 0;

             end

             else begin

                 gate\_cnt <= gate\_cnt + 32'd1;

                 gate\_time\_pre <= 1;

                 R\_done\_pre <= 0;

             end

         end

         else begin

             gate\_time\_pre <= 0;

             gate\_cnt <= 32'd0;

             R\_done\_pre <= R\_done\_pre;

         end

     end

 end

The code for pre-gate synchronized to the clock domain to be measured is shown below.

1. reg gate\_time;
2. always @(posedge I\_clk\_fx or negedge I\_rst\_n) begin
3. if(!I\_rst\_n) begin
4. gate\_time <= 0;
5. end
6. else begin
7. gate\_time <= gate\_time\_pre;
8. end
10. end

The code for the reference signal and the signal to be measured is shown below.

1. assign O\_done = ({R\_done\_pre,gate\_time} == 2'b10) ? 1 : 0;
2. always @(posedge I\_clk\_fx or negedge I\_rst\_n) begin
3. if((!I\_rst\_n)) begin
4. O\_fx\_cnt <= 32'd0;
5. end
6. else begin
7. if(gate\_time) begin
8. O\_fx\_cnt <= O\_fx\_cnt + 32'd1;
9. end
10. else begin
11. O\_fx\_cnt <= O\_fx\_cnt;
12. end
13. end
14. end
15. always @(posedge I\_sys\_clk or negedge I\_rst\_n) begin
16. if((!I\_rst\_n)) begin
17. O\_f0\_cnt <= 32'd0;
18. end
19. else begin
20. if(gate\_time) begin
21. O\_f0\_cnt <= O\_f0\_cnt + 32'd1;
22. end
23. else begin
24. O\_f0\_cnt <= O\_f0\_cnt;
25. end
26. end
27. end
28. Digital tube sampling module

The diagram is as shown below.

The code is as follows.

1. always @(posedge I\_sys\_clk or negedge I\_rst\_n) begin
2. if(~I\_rst\_n) begin
3. R\_test\_num <= 0;
4. O\_data1 <= 0;
5. O\_data0 <= 0;
6. R\_01\_flag <= 1'b0;
7. R\_02\_flag <= 1'b0;
8. R\_03\_flag <= 1'b0;
9. R\_04\_flag <= 1'b0;
10. end
11. else begin
12. if(R\_start\_en & I\_test\_en) begin
13. case (I\_test\_sel)
14. 4'b0001 : begin
15. O\_data1[15:0] <= I\_test\_disp\_data;
16. O\_data0[7:0] <= I\_test\_seg;
17. R\_01\_flag <= 1'b1;
18. if(R\_01\_flag) begin
19. R\_test\_num <= R\_test\_num;
20. end
21. else begin
22. R\_test\_num <= R\_test\_num + 1;
23. end
25. end
26. 4'b0010 : begin
27. O\_data0[15:8] <= I\_test\_seg;
28. R\_02\_flag <= 1'b1;
29. if(R\_02\_flag) begin
30. R\_test\_num <= R\_test\_num;
31. end
32. else begin
33. R\_test\_num <= R\_test\_num + 1;
34. end
35. end
36. 4'b0100 : begin
37. O\_data0[23:16] <= I\_test\_seg;
38. R\_03\_flag <= 1'b1;
39. if(R\_03\_flag) begin
40. R\_test\_num <= R\_test\_num;
41. end
42. else begin
43. R\_test\_num <= R\_test\_num + 1;
44. end
45. end
46. 4'b1000 : begin
47. O\_data0[31:24] <= I\_test\_seg;
48. R\_04\_flag <= 1'b1;
49. if(R\_04\_flag) begin
50. R\_test\_num <= R\_test\_num;
51. end
52. else begin
53. R\_test\_num <= R\_test\_num + 1;
54. end
55. end
56. default: O\_data0 <= O\_data0;
57. endcase
58. end
59. else  begin
60. if(I\_test\_en) begin
61. R\_test\_num <= R\_test\_num;
62. O\_data1 <= O\_data1;
63. O\_data0 <= O\_data0;
64. end
65. else begin
66. R\_test\_num <= 0;
67. O\_data1 <= 0;
68. O\_data0 <= 0;
69. R\_01\_flag <= 1'b0;
70. R\_02\_flag <= 1'b0;
71. R\_03\_flag <= 1'b0;
72. R\_04\_flag <= 1'b0;
73. end
74. end
75. end
76. end
77. Serial transceiver module

The diagram is as shown below.



For the code, you can refer to the source.

1. Control module

The diagram is as shown below.

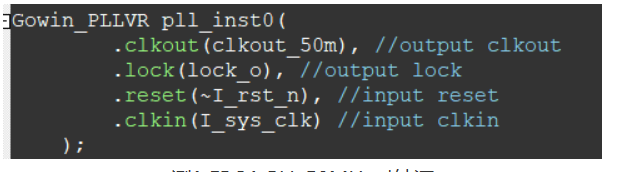
**Host Design**

The source code is as follows.

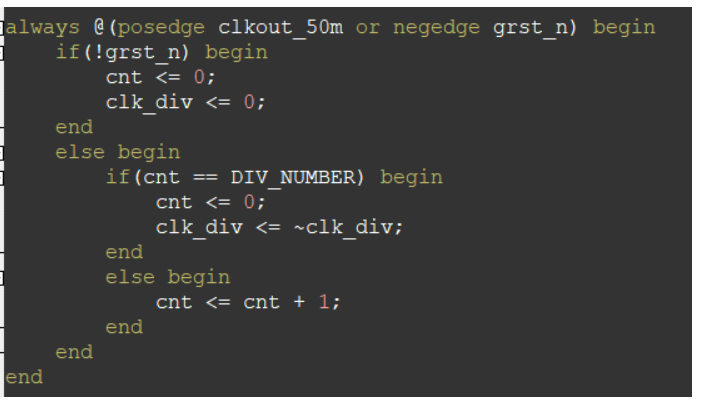
1. HeadByte[0] = 0xEC;
2. HeadByte[1] = 0x01;
3. HeadByte[2] = 0x00;
4. HeadByte[3] = 0x00;
5. HeadByte[4] = 0x00;
6. HeadByte[5] = 0x00;
7. HeadByte[6] = 0x00;
8. HeadByte[7] = 0xa5;
9. void Widget::on\_pushButtonP1\_clicked()
10. {
11. HeadByte[1] = 0x01;
12. //serialport->write(HeadByte);
13. QByteArray tmp;
14. tmp.resize(1);
15. for(int idx = 0;idx < 8;idx++){
16. tmp[0] = HeadByte.at(idx);
17. serialport->write(tmp);
18. Sleep(10);
19. }
20. ui->pushButtonP1->setEnabLED(false);
21. ui->pushButtonP2->setEnabled(false);
22. flag\_state = 1;
23. ui->textCommd->append("Frequence measure cmd has sended!\n");
24. }

**Testing**

Frequency testing



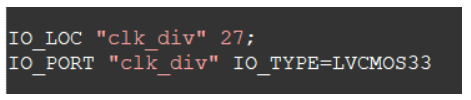
FPGA PLL 50MHz Clock Source



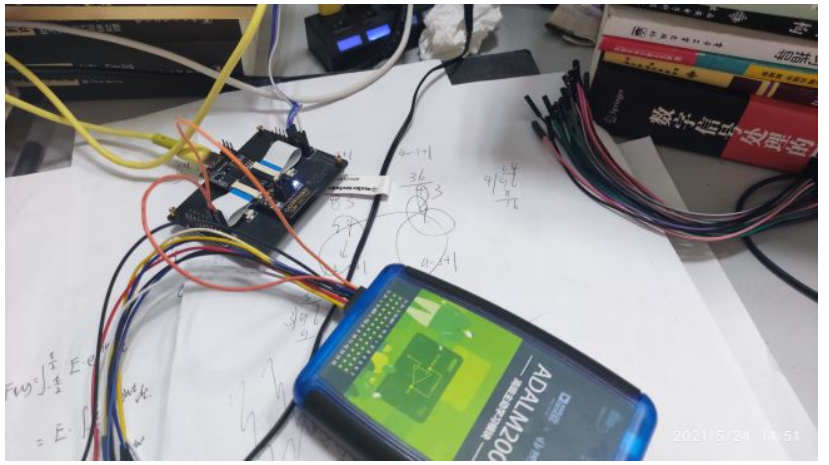
Division code

C:\Users\DELL\AppData\Local\Temp\企业微信截图_16306539205408.png

100KHz division coefficient

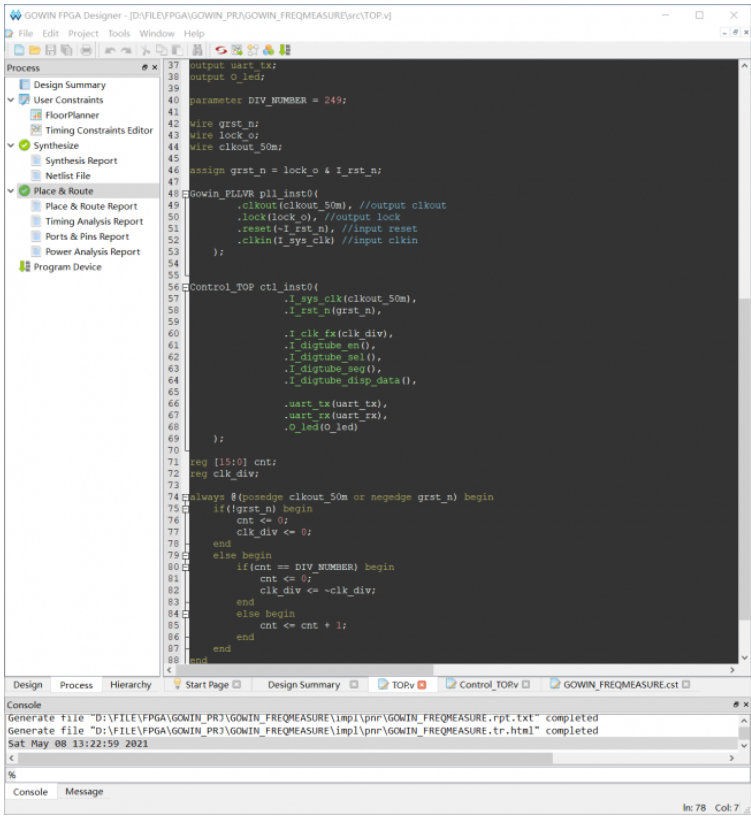


Pinout



Connection

100K frequency Testing



User design clock 100KHz

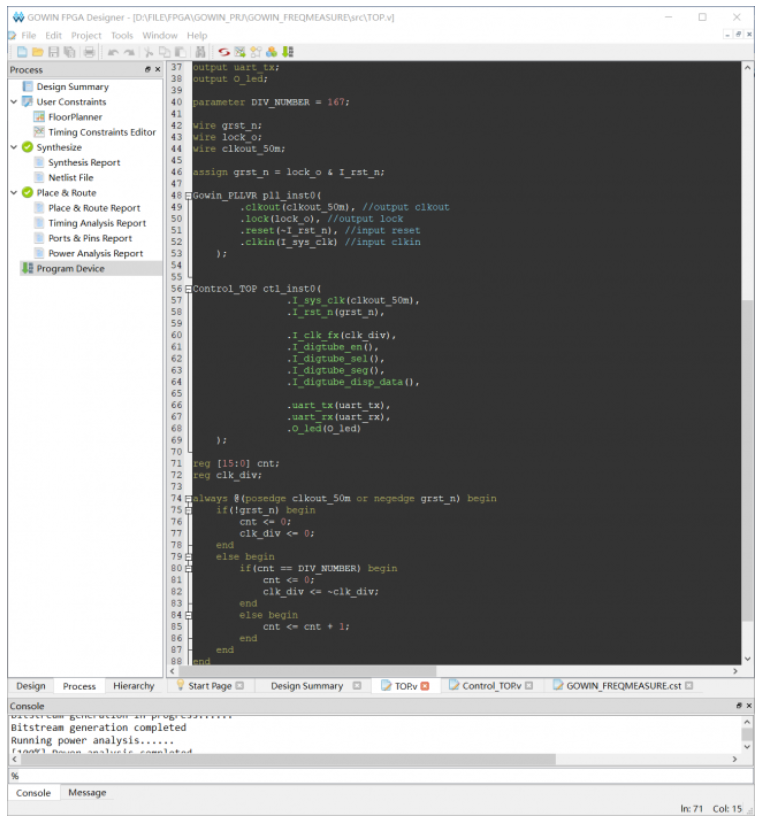


Testing by host



Testing by ADALM2000

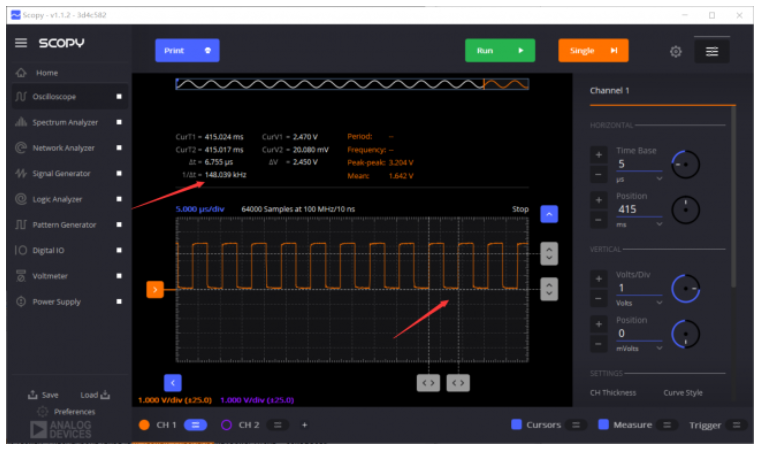
149K frequency Testing



User design 149KHz

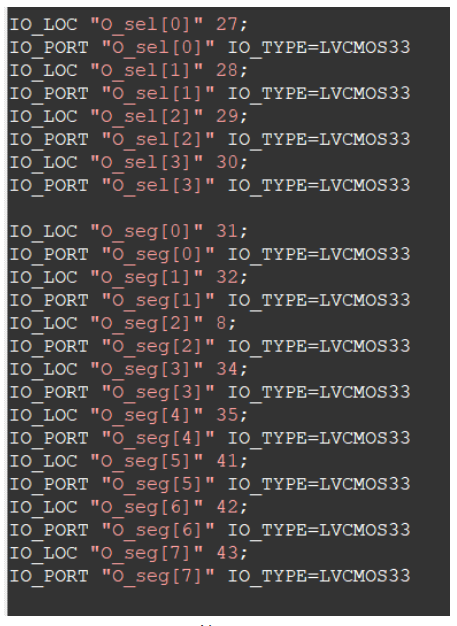


Testing by host

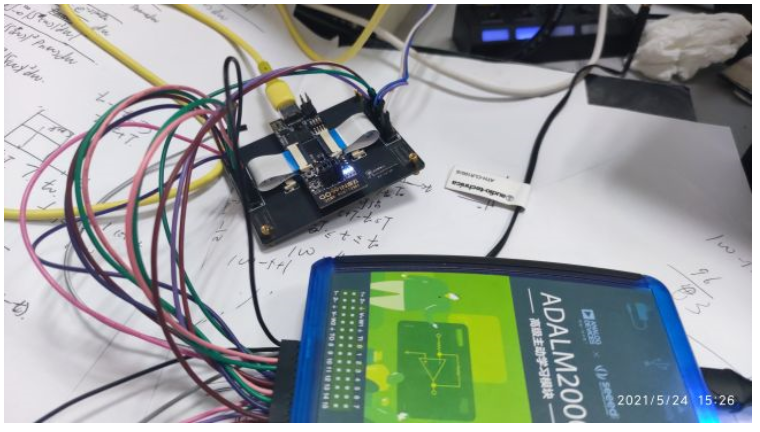


Testing by ADALM2000

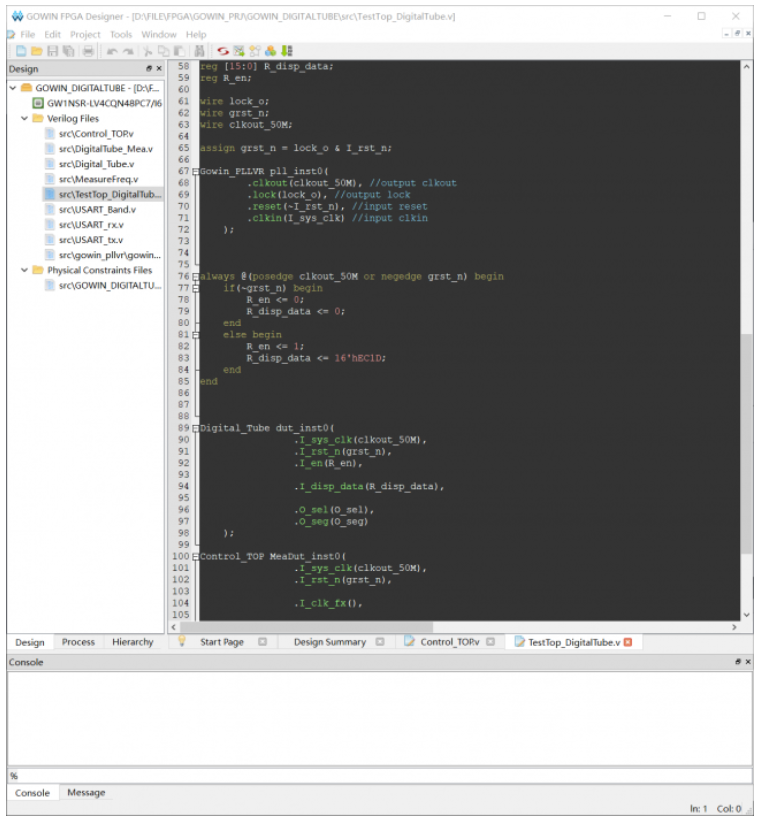
Digital tube testing



Output Pinout



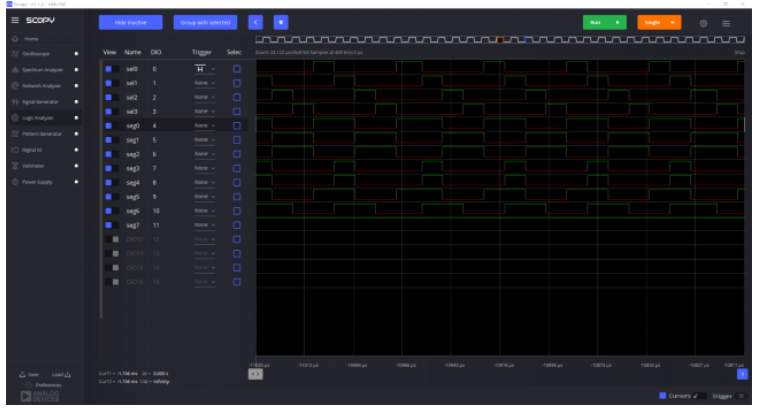
Connection



Gowin editor



Testing by host



Testing by ADALM2000

**Source code**

You can click this link <https://github.com/kevinliuyunfeng/GOWIN_FPGA.git> to get the source code.