Gowin Software User Messages Reference

SUG937-1.01E, 2020-08-06

Disclaimer

GOWINSEMI®, LittleBee®, Arora, and the GOWINSEMI logos are trademarks of GOWINSEMI and are registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders, as described at www.gowinsemi.com. GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. All information in this document should be treated as preliminary. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

Revision History

Data	Version	Description
2020/06/01	1.0E	Initial version published.
2020/08/06	1.01E	● Changed EX0311 ERROR to EX0210 WARN; ● EX0206 WARN code updated; ● EX0310 ERROR code updated.

Contents

Gowin Software User Messages Reference

Overview

This manual illustrates Gowin software user messages to help you deal with the warnings and errors when using the software. There are GowinSynthesis and Place & Route user messages in this manual. As the software is subject to change without notice, some information may not remain relevant and may need to be adjusted according to the software that is in use.

GowinSynthesis User Messages

AG0100

WARN (AG0100) : Find logical loop signal : <signal>

If there is a logical loop in the design, the synthesis tool lists the line information of <signal> in the loop according to the above warning. You can modify the loop to avoid this warning. In the following case, the output out drives itself, resulting in a logical loop.

```
module test (in,out);
input in;
output out;
assign out = in & !out;
endmodule
```

Action

You need to modify the design to avoid the logical loop.

AG0101

WARN (AG0101) : The netlist is not one directed acyclic graph including user instantiated primitives

If there are logical loops in the design, and there are user-instantiated logic primitives in the logical loops, such as LUT and ALU, etc., the synthesis tool will pop up the above warning and continues to synthesize. In the following case, out1 and out2 drive each other resulting in logical loops, and there is user-instantiated LUT3 ins1 in the loop.

```
module test (a,b,out1,out2);
input a,b;
output out1,out2;
assign out1 = out2 & b;
LUT3 ins1(
    .I0(a),
    .I1(b),
    .I2(out1),
    .F(out2)
);
defparam ins1.INIT=8'hAB;
endmodule
```

Action

You need to modify the design to avoid the logical loop.

AG0200

ERROR (AG0200): The netlist is not one directed acyclic graph

Similar to AG0100, if there is a logical loop in the design, the synthesis tool will report the above error and stop synthesis. You need to remove the logical loop according to AG0100 and re-synthesize.

```
module test (in,out);
input in;
output out;
assign out = in & !out;
endmodule
```

Action

You need to modify the design to avoid the logical loop and re-synthesize.

CK0013

ERROR (CK0013) : <signal> is not connected to buf or iodelay.

In the chip design, there is a fixed connection in some logical units, so there is a connection limit for the drive source or destination. If it is beyond the limit, the synthesis tool will report the above error. In the following case, Q0 drives Q0&Q1 in oser4 instance, but it can not be implemented in placement.

```
    module OSER4_ins (Q0_test, D0, D1, D2, D3, TX0, TX1, PCLK, FCLK, RESET);
    input D0, D1, D2, D3;
    input TX0, TX1;
    input PCLK, FCLK, RESET;
    output Q0_test;
    wire Q0;
    wire Q1;
    OSER4 oser4(
   .Q0(Q0),
   .Q1(Q1),
   .D0(D0),
   .D1(D1),
   .D2(D2),
   .D3(D3),
   .TX0(TX0),
   .TX1(TX1),
   .PCLK(PCLK),
   .FCLK(FCLK),
   .RESET(RESET)
);
defparam oser4.GSREN = "false";
defparam oser4.LSREN = "true";
defparam oser4.HWL = "false";
defparam oser4.TXCLK_POL = 1'b0;
    assign Q0_test = Q0 & Q1;
    endmodule
```

Action

You need to modify the design to make Q0 drive output port.

CK2027

ERROR (CK2027) : The connection between Instance <inst1> and instance <inst2> is not correct!

In the chip design, there is a fixed connection between <inst1> and <inst2>, so there is a connection limit for the drive source or destination. If it is beyond the limit, the synthesis tool will report the above error. In the following case, the output port CLKOUT is CLKOUT pin of DHCEN instance. In the design, CLKOUT of DHCEN can only drive clock ports of IOLOGIC/CLKDIV/DLL/PLL/DQS and so on.

```
module DHCEN_ins (CLKOUT, CLKIN, CE);
input CLKIN,CE;
output CLKOUT;
DHCEN dhcen(
   .CLKOUT(CLKOUT),
   .CLKIN(CLKIN),
   .CE(CE)
);
endmodule
```

Action

You need to add a correct drive for CLKOUT.

```
module DHCEN_ins (Q0, CLKIN, CE);
input CLKIN,CE;
output Q0;
wire Q1;
wire D0;
wire D1;
wire D2;
wire D3;
wire TX0;
wire TX1;
wire PCLK;
wire RESET;
wire CLKOUT;
DHCEN dhcen(
   .CLKOUT(CLKOUT),
   .CLKIN(CLKIN),
   .CE(CE)
);
OSER4 oser4(
   .Q0(Q0),
   .Q1(Q1),
   .D0(D0),
   .D1(D1),
   .D2(D2),
   .D3(D3),
   .TX0(TX0),
   .TX1(TX1),
   .PCLK(PCLK),
   .FCLK(CLKOUT),
   .RESET(RESET)
);
defparam oser4.GSREN = "false";
defparam oser4.LSREN = "true";
defparam oser4.HWL = "false";
defparam oser4.TXCLK_POL = 1'b0;
endmodule
```

CV0003

WARN (CV0003) : Output <port> has undriven bits, assigning undriven bits to 0, simulation mismatch possible

Output <port> is dangling, and GND or high impedance will be added to it. The synthesis simulation may be different from that of rtl. The synthesis tool will pop up the above warning and continue synthesis. In the following case, Output port o2 is dangling.

```
module test(a,b,o1,o2);
input a,b;
output o1,o2; // o2 dangling
assign o1 = a & b;
endmodule
```

Action

If an output port is connected to an internal signal, assign GND or VCC to it.

```
module test(a,b,o1,o2);
input a,b;
output o1,o2;
assign o1 = a & b;
assign o2 = 1’b0; // assign GND to dangling output port
endmodule
```

CV0005

ERROR (CV0005) : Tran switch which all inputs are connected to inout port can not be converted

All the pins of tran switch cannot be connected to the inout port, otherwise the data will be conflicted. The synthesis tool will pop up the above warning and stop synthesis. In the following case, the two pins of tran are connected to inout port.

```
module test(io1,io2,control);
inout io1,io2;
input control;
tran t(io1,io2); //tran D0 and D1 all connect inout port
```

```
endmodule
```

Action

You need to connect the two pins to different ports.

```
module test(o1,control);
output o1;
input control;
tran t(control,o1);//tran D0 connect input port, D1 connect output port
endmodule
```

CV0008

ERROR (CV0008) : Convert tran switch <object> failed

If a pin of tran switch <object> is connected illegally, which makes it impossible to synthesize. The synthesis tool will report the above error and stop synthesis. In the following case, a pin of tran is dangling.

```
module test(o1);
output o1;
wire control;
tran t(o1,control); // tran D1 dangling
endmodule
```

Action

You need to connect the dangling pin to a port or a signal.

```
module test(o1,control);
output o1;
input control;
tran t(control,o1); // connect tran D1 to output port
endmodule
```

CV0013

ERROR (CV0013) : Tri-state signal <signal> connected ERROR, it should only be connected to PAD.

The tri-state Buf signal can be only connected to PAD. In the following case, the signal io is connected to PAD, but it also driven by GND(2’b00). The synthesis tool will report above error.

```
module InterFace(
    inout io,
    input  in, io_en,
    output lrck
);
IOBUF gwBuf(
.I(in),
.OEN(io_en),
.IO(io),
.O(lrck)
);
assign io = 2'b00;  //assign GND to io
endmodule
```

Action

You need remove 2’b00.

```
module InterFace(
    inout io,
    input  in, io_en,
    output lrck
);
IOBUF gwBuf(
.I(in),
.OEN(io_en),
.IO(io),
.O(lrck)
);
endmodule
```

CV0014

ERROR (CV0014) : Not Support MOS switch <signal> synthesis

The instance cmos/rcmos synthesis is not supported. The synthesis tool will report the above error. In the following case, instantiate a cmos.

```
module test(in,control1,control2,o);

input in,control1,control2;

output o;

cmos c(o,in,control1,control2);

endmodule
```

Action

You need to remove cmos/rcmos instance and replace it with other logic gate.

DI0002

WARN (DI0002) : Asynchronous register < asynReg > initial values do not match with the Gowin library, simulation mismatch possible

In the chip design, the initial value of the asynchronous set flip-flop <asynReg> can only be set to 1, and the initial value of the asynchronous reset flip-flop <asynReg> can only be set to 0. Otherwise, the above warning will pop up. In the following case, Register o is an asynchronous reset D flip-flop with an initial value of 0, but it is set to 1 in RTL.

```
    module test(clk,d,clear,o);
    input clk,d,clear;
    output o;
    reg o = 1'b1;
    always @(posedge clk or posedge clear) // async register
    if(clear)
    o = 0; // register output 0 if clear, this register will be synthesized
to DFFC
    else
    o = d;
    endmodule
```

Action

The initial value of <asynReg> can not be set, or the initial value is consistent with the reset/set value of the clear/preset signal of <asynReg>.

```
    module test(clk,d,clear,o);
    input clk,d,clear;
    output o;
    reg o = 1'b0;
    always @(posedge clk or posedge clear) // async register
    if(clear)
    o = 0; // register output 0 if clear, this register will be synthesized
to DFFC
    else
    o = d;
    endmodule
```

EX0200

WARN (EX0200): Property <prop> set invalid for <object>

When <object> sets the property constraint <prop> in the design, but the value is invalid, the above warning will pop up and the software will use the default initial value. For example, in the following test case, the property constraint syn_ramstyle is null and the above warning will be reported.

```
module normal1(data_out, data_in, addr, clk, wre,rst);
output [1:0]data_out;
input [1:0]data_in;
input [6:0]addr;
input clk,wre,rst;
reg [1:0] mem [127:0] /* synthesis syn_ramstyle = "" */;
reg [1:0] data_out;
always@(posedge clk or posedge rst)
 if(rst ==1)
        data_out <= 0;
 else
    if(wre == 0)
       data_out <= mem[addr];
 always @(posedge clk)
    if (wre) mem[addr] <= data_in;
endmodule
```

Action

To make sure property constraint value is valid, registers value is assigned to syn_ramstyle in the following corrected test case.

```
module normal1(data_out, data_in, addr, clk, wre,rst);
output [1:0]data_out;
input [1:0]data_in;
input [6:0]addr;
input clk,wre,rst;
reg [1:0] mem [127:0] /* synthesis syn_ramstyle = "registers" */;
reg [1:0] data_out;
always@(posedge clk or posedge rst)
 if(rst ==1)
        data_out <= 0;
 else
    if(wre == 0)
       data_out <= mem[addr];
 always @(posedge clk)
    if (wre) mem[addr] <= data_in;
endmodule
```

EX0201

WARN (EX0201): Missing INIT parameter on <object> and using

default value

When <object> is instantiated in the design, but it does not set the initial value, the above warning will pop up and the software will use the default initial value. In the following test case, the LUT3 ins1 is instantiated in the design, but the ins1 does not set the INIT initial value.

```
module test(a,b,c,out);
input a,b,c;
output out;
LUT3 ins1(
    .I0(a),
    .I1(b),
    .I2(c),
    .F(out)
);
/*lack defparam of LUT3*/
endmodule
```

Action

You need to assign an initial value to the instance <object>, as shown below.

```
module test(a,b,c,out);
input a,b,c;
output out;
LUT3 ins1(
    .I0(a),
    .I1(b),
    .I2(c),
    .F(out)
);
defparam ins1.INIT=8'hEF;
endmodule
```

EX0203

WARN (EX0203) : Top module <modu> has no ports

If the top module does not have any ports, the above warning will pop up. In the following case, test module does not have any ports.

```
module test();
wire a,b,out;
assign a = 1'b0;
assign b = 1'b1;
assign out = a ^ b;
endmodule
```

Action

You need to set at least one input or output port in the <modu>.

EX0205

WARN (EX0205) : Instance <inst> 's parameter <para> value invalid, replaced by default value <para>

If the parameter < para> of the instantiated primitive <inst> is invalid, the above warning will pop up and the software will use the default value. In the following case, <para> of LUT4 is invalid.

```
module unitest ( i,out);
input [3:0] i;
output out;
LUT4 lut4_0 (
    .I0(i[0]),
    .I1(i[1]),
    .I2(i[2]),
    .I3(i[3]),
    .F(out)
);
defparam lut4_0.INIT = "wabajd";///< right: 16'h0000 to 16'hFFFF
endmodule
```

Action

You need to set the valid value by referring to SUG283, Gowin Primitives User Guide.

```
module unitest ( i,out);
input [3:0] i;
output out;
LUT4 lut4_0 (
    .I0(i[0]),
    .I1(i[1]),
    .I2(i[2]),
    .I3(i[3]),
    .F(out)
);
defparam lut4_0.INIT = 16'h1234;///< right: 16'h0000 to 16'hFFFF
endmodule
```

EX0206

WARN (EX0206) : Instance <inst> 's parameter <para> value invalid

When the currently specified device is inconsistent with the device specified in <para> of instantiated Gowin Primitive <inst>, such as rPLL, the above warning will pop up. In the following case, the specified device is inconsistent with rPLL GW1N-4B.

```
module test(i,out);
input [35:0]i;
output [4:0]out;
rPLL rpll (
    .CLKIN(i[0]),
    .CLKFB(i[1]),
    .FBDSEL(i[7:2]),
    .IDSEL(i[13:8]),
    .ODSEL(i[19:14]),
    .DUTYDA(i[23:20]),
    .PSDA(i[27:24]),
    .FDLY(i[31:28]),
    .RESET(i[32]),
    .RESET_P(i[33]),
    .CLKOUT(out[0]),
    .CLKOUTP(out[1]),
    .CLKOUTD(out[2]),
    .CLKOUTD3(out[3]),
    .LOCK(out[4])
);
defparam pll.DEVICE = "GW1N-4B";
endmodule
```

Action

You need to make the device consistent with the one of <para> in instantiated Gowin Primitives.

EX0210

WARN (EX0210) : Invalid input frequency <freq> to instance <inst>, suitable range is from <num1>MHz to <num2>MHz

If the synthesis tool reports the above error, it indicates that the <freq> of <inst> is not in the range from <num1> to <num2>. Please modify the design according to the name and line information. The instantiated primitive <inst> that has not set <freq> will be checked as the default value.

```
module test(i,out);
input [35:0]i;
output [4:0]out;
PLL pll (
    .CLKIN(i[0]),
    .CLKFB(i[1]),
    .FBDSEL(i[7:2]),
    .IDSEL(i[13:8]),
    .ODSEL(i[19:14]),
    .DUTYDA(i[23:20]),
    .PSDA(i[27:24]),
    .FDLY(i[31:28]),
    .RESET(i[32]),
    .RESET_P(i[33]),
    .RESET_I(i[34]),
    .RESET_S(i[35]),
    .CLKOUT(out[0]),
    .CLKOUTP(out[1]),
    .CLKOUTD(out[2]),
    .CLKOUTD3(out[3]),
    .LOCK(out[4])
);
defparam pll.FCLKIN = "100.0";
defparam pll.DYN_IDIV_SEL = "false";
defparam pll.IDIV_SEL = 0;
defparam pll.DYN_FBDIV_SEL = "false";
defparam pll.FBDIV_SEL = 0;
defparam pll.DYN_ODIV_SEL = "false";
defparam pll.ODIV_SEL = 32;
defparam pll.PSDA_SEL = "0000";
defparam pll.DYN_DA_EN = "false";
defparam pll.DUTYDA_SEL = "1000";
defparam pll.CLKOUT_FT_DIR = 1'b1;
defparam pll.CLKOUTP_FT_DIR = 1'b1;
defparam pll.CLKOUT_DLY_STEP = 0;
defparam pll.CLKOUTP_DLY_STEP = 0;
defparam pll.CLKFB_SEL = "internal";
defparam pll.CLKOUT_BYPASS = "false";
defparam pll.CLKOUTP_BYPASS = "false";
defparam pll.CLKOUTD_BYPASS = "false";
defparam pll.DYN_SDIV_SEL = 2;
defparam pll.CLKOUTD_SRC = "CLKOUT";
defparam pll.CLKOUTD3_SRC = "CLKOUT";
defparam pll.DEVICE = "GW1N-4B";
endmodule
```

Action

You need to modify the specified device or refer to the reported error and the <freq> value in SUG283, Gowin Primitives User Guide to avoid this error.

EX0302

ERROR (EX0302) : No valid top module found

If there is no module in the design, the synthesis tool will report the above error.

EX0308

ERROR (EX0308) : GowinSynthesis can not find file \"primitive.xml\". Please reinstall the product

If the primitive.xml file under the IDE\bin directory is removed, the synthesis tool will report the above error. Try to move the primitive.xml back or reinstall the software.

EX0309

ERROR (EX0309): Net <object> has multiple drivers

If there is a signal with multiple non-tristate drivers in the design, the synthesis tool will report the above error. It can be found and modified according to the signal name and line information prompted by the error. In the following case, the output port out is driven by both rotate_1 and rotate_2.

```
module rotate (q1, data, sel1) ;
output [7:0] q1;
input [7:0] data;
input sel1;
endmodule

module top (out, ci, data1, data2);
output [7:0] out;
input [7:0] data1, data2;
input ci;
rotate rotate_1 (out, data1, ci);
rotate rotate_2 (out, data2, ci);
```

```
endmodule
```

Action

You need to make sure that a signal has only one driver, as shown below.

```
module rotate (q1, data, sel1) ;
output [7:0] q1;
input [7:0] data;
input sel1;
endmodule

module top (out, ci, data1);
output [7:0] out;
input [7:0] data1;
input ci;
rotate rotate_1 (out, data1, ci);
endmodule
```

EX0310

ERROR (EX0310) : Invalid parameterized value <paraValue>(<para>) specified for instance <inst>

If the <paraValue> of <inst> <para> in Gowin Primitives is invalid, the synthesis tool will report the above error. Please modify the design according to the name and line information. For example, the FCLKIN value of rPLL is out of range.

```
module test(i,out);
input [35:0]i;
output [4:0]out;
PLL pll (
    .CLKIN(i[0]),
    .CLKFB(i[1]),
    .FBDSEL(i[7:2]),
    .IDSEL(i[13:8]),
    .ODSEL(i[19:14]),
    .DUTYDA(i[23:20]),
    .PSDA(i[27:24]),
    .FDLY(i[31:28]),
    .RESET(i[32]),
    .RESET_P(i[33]),
    .CLKOUT(out[0]),
    .CLKOUTP(out[1]),
    .CLKOUTD(out[2]),
    .CLKOUTD3(out[3]),
    .LOCK(out[4])
);
defparam pll.DEVICE = "GW1N-4B";
defparam pll.FCLKIN = "600.0";
endmodule
```

Action

You need to make the <paraValue> valid by referring to SUG283, Gowin Primitives User Guide.

EX2526

WARN (EX2526) : Entry size <width> at <initvalue>:<initWidth> does not match memory width <memWidth>

If the $readmemh is used and the file data width does not match with <memWidth>, the above warning will pop up. In the following case, mem width is 8, and the data width <initWidth> of file <initvalue> can not match.

```
module test(in,clk,addr_in,addr_out,out);

input in,clk;

input addr_in;

input addr_out;

output out;

reg mem[7:0];

 always @ (posedge clk)

     mem[addr_in] <= in;

 assign out = mem[addr_out];

initial begin

    $readmemh("initvalue", mem);

  end

endmodule
```

Action

You need to make the two widths match with each other.

EX2598

WARN (EX2598) : <design> might have mixed concurrent and procedural assignment

If there are blocking assignment and nonblocking assignment in <design>, the synthesis tool will pop up the warning. In the following case, there are blocking assignment and nonblocking assignment for d_reg in different conditions.

```
module gene_if(t0,t1,t2,d,clk,t);

input t0,t1,t2,clk,t;

output d;

reg d_reg;

localparam S=6;

generate

    if(S<7)

        assign d_reg=t0|t1|t2;

    else

        assign d_reg=t0&t1&t2;

endgenerate

generate

    if(S>7)

        always @(posedge clk)

        d_reg<=t;

        else

        always @(posedge clk)

        d_reg<=t0&t1&t2;

endgenerate

assign d=d_reg;

endmodule
```

Action

You need to delete one of assignment as shown below.

```
module gene_if(t0,t1,t2,d,clk,t);

input t0,t1,t2,clk,t;

output d;

reg d_reg;

localparam S=6;

generate

    if(S>7)

        always @(posedge clk)

        d_reg<=t;

        else

        always @(posedge clk)

        d_reg<=t0&t1&t2;

endgenerate

assign d=d_reg;

endmodule
```

EX2629

WARN (EX2629) : Delay control is not supported for synthesis

The delay control is not supported. If the design file includes delay, such as #10 in the following case, the synthesis tool will pop up the warning. All the delays will be ignored.

```
module top (in0,in1,clk,out);
input in0,in1;
input clk;
output reg out;
always @( posedge clk)
    begin
        out <= #10 in0&in1;
    end
endmodule
```

Action

You need to remove the delay control in the design file as shown below.

```
module top (in0,in1,clk,out);
input in0,in1;
input clk;
output reg out;
always @( posedge clk)
    begin
       out <= in0&in1;
    end
endmodule
```

EX2635

WARN (EX2635) : Generate block is allowed only inside loop and conditional generate in SystemVerilog mode

If the generate block is used in the design, the above warning will pop up. In the following case, the test module is generated in the for loop, which is supported only in SystemVerilog.

```
module top(in,out);
input [1:0]in;
output [1:0]out;
generate
begin
   genvar i;
   for (i=0;i<2;i=i+1)
   begin : reg_loop1
    test test1(in[i],out[i]);
   end
end
endgenerate
endmodule
module test(in,out);
input in;
output out;
assign out = !in;
```

```
endmodule
```

Action

It should be noted that it can only be used in SystemVerilog.

EX2656

ERROR (EX2656) : SystemVerilog keyword <word> used in incorrect context

If the definition is the same as a keyword, the synthesis tool will report the above error. Null is a keyword in systemVerilog and can not be used as a definition.

```
module top (in0,in1,out);
input in0,in1;
output out;
wire null;
assign  out= in0&in1;
endmodule
```

Action

You need to avoid using the keyword for definition as shown below.

```
module top (in0,in1,out);
input in0,in1;
output out;
assign  out= in0&in1;
endmodule
```

EX2664

WARN (EX2664) : Variable <vari> may be used before assigned in always_comb or always @* block : might cause synthesis - simulation differences

If there is a <vari> change in always sensitivity list, the above warning will pop up. In the following case, the tmp is always changing in sensitivity list.

```
module top(in,sel,out);

input in,sel;

output reg out;

reg tmp;
```

```
always@(*)
    begin
        if(sel)
            tmp <= in;
        else
            tmp <= !tmp;
    end
assign out= tmp;
endmodule
```

Action

You need to add clock signal and remove tmp from sensitivity list as shown below.

```
module top(in,sel,out,clk);
input in,sel,clk;
output reg out;
reg tmp;
always@(posedge clk)
    begin
        if(sel)
            tmp <= in;
        else
            tmp <= !tmp;
    end
assign out= tmp;
endmodule
```

EX2565

WARN (EX2565) : Port <port> is not connected on this instance

If there is a port that is not connected on this instance, the above warning will pop up. In the following case, the clk is not connected in test module.

```
module top (in0,in1,out,out1);

input in0,in1;

output out,out1;

assign out = in0 & !in1;

test test1(

       .data(in0),

       .out(out1)

);

endmodule

module test (data,out,clk);

input data,clk;

output out;

assign out = !data;

endmodule
```

Action

You need to remove the dangling port as shown below.

```
module top (in0,in1,out,out1);

input in0,in1;

output out,out1;

assign out = in0 & !in1;

test test1(

       .data(in0),

       .out(out1)

);

endmodule

module test (data,out);

input data;

output out;

assign out = !data;
```

```
endmodule
```

EX2666

WARN (EX2666) : Unsupported use of clock signal <signal>, clock used as data

If there is a signal <signal> both as clock and data, the above warning will pop up. In the following case, the clk is both as clock and data.

```
module top (clk,out);

input clk;

output reg out;

always@(posedge clk)

    out <= clk;

endmodule
```

Action

You need to separate the data and clock as shown below.

```
module top (in,clk,out);

input in,clk;

output reg out;

always@(posedge clk)

    out <= in;

endmodule
```

EX2830

WARN (EX2830) : Data object <object> is already declared

If the <object> has been defined repeatedly, the above warning will pop up. In the following case, the wire has been defined twice.

```
module top (in,out);
input in;
output out;
wire out;
wire out;
assign out = !in;
endmodule
```

Action

You need to delete one as shown below.

```
module top (in,out);
input in;
output out;
wire out;
assign out = !in;
endmodule
```

EX2855

WARN (EX2855) : Result of this <oper> operation does not fit in <width> bits

If the <oper> is used in the design, but the result of this operation exceeds the assigned bit width, the above warning will pop up. In the following case, the result of power operation does not fit in out port width.

```
module top(in,out);
input in;
output [1:0]out;
assign out = 6'd2 ** (16'h77)+in;
endmodule
```

Action

You need to keep the two bit widths same as shown below.

```
module top(in,out);
input in;
output [1:0]out;
assign out = 2'b01 ** (2'b10)+in;
endmodule
```

EX2932

WARN (EX2932) : Unknown system task <task> ignored for synthesis

If there is an unknown system task <task>, the above warning will pop up. In the following case, the $fsdbDumpMDA will be ignored when synthesized.

```
module test(in,out);

input in;

output out;

reg mem;

assign out = in;

initial begin

    $fsdbDumpMDA(mem);

end

endmodule
```

Action

You need to delete the unknown system task as shown below.

```
module test(in,out);

input in;

output out;

reg mem;

assign out = in;

endmodule
```

EX2947

WARN (EX2947) : Input port <port> remains unconnected for this instance

If the input port in the module is instantiated, and no signal is connected, the above warning will pop up. In the following case, there is no in1 connection in sub instance.

```
module top (top_in0,top_in1,top_out);
input top_in0,top_in1;
output top_out;
sub sub1(
.in0(top_in0),
.in1(),
.out(top_out)
);
endmodule
module sub (in0,in1,out);
```

```
input in0,in1;
output out;
assign out = in0 & !in1;
endmodule
```

Action

You can connect the dangling port as shown below.

```
module top (top_in0,top_in1,top_out);
input top_in0,top_in1;
output top_out;
sub sub1(
.in0(top_in0),
.in1(top_in1),
.out(top_out)
);
endmodule
module sub (in0,in1,out);
input in0,in1;
output out;
assign out = in0 & !in1;
endmodule
```

EX2981

WARN (EX2981) : Net <objec> is driven by multiple input ports

If the <object> is driven by multiple input ports in the design, the above warning will pop up. In the following case, tmp has been assigned twice.

```
module test(in,out);

input [3:0]in;

output [3:0]out;

wire [3:0]tmp;

assign tmp[3:0] = in;

assign tmp[1:0] = in;

assign out = tmp;

endmodule
```

Action

You need to delete the repeated value as shown below.

```
module test(in,out);

input [3:0]in;

output [3:0]out;

wire [3:0]tmp;

assign tmp[3:0] = in;

assign out = tmp;

endmodule
```

EX2987

WARN (EX2987) : Input port <port> is not connected on this instance

If the input port in the module is instantiated, and no signal is connected, the above warning will pop up. In the following case, there is no in port connection when test module is instantiated.

```
module top (in0,out1);
input in0;
output out1;
test test1(
.in1(in0),
.in(),
.out(out1)
);
endmodule
module test (in1,in,out);
input in1;
input in;
output out;
assign out = !in1;
endmodule
```

Action

You need to delete dangling port as shown below.

```
module top (in0,out1);
input in0;
output out1;
test test1(
.in1(in0),
.out(out1)
);
endmodule
module test (in1,out);
input in1;
output out;
assign out = !in1;
endmodule
```

EX2997

WARN (EX2997) : Net <net> is already driven by input port <port>

If a value is assigned to input <port> in the design, the above warning will pop up. The input port d is driven by other input port in the following case.

```
module test(b,c,d,f);
input b,c,d;
output f;
assign d = c&b;
assign f = b&d;
endmodule
```

Action

You need to delete it as shown below.

```
module test(b,d,f);
input b,d;
output f;
assign f = b&d;
endmodule
```

EX2998

WARN (EX2998): Net <object> does not have a driver

If a wire or register is defined, but the <object> does not have a driver, the above warning will pop up. In the following case, the wire a is defined but without a driver.

```
module top (in0,in1,out);
input in0,in1;
output out;
wire a;
assign out = in0&in1|a;
endmodule
```

Action

You need to delete the wire a or add the corresponding connection as shown below.

```
module top (in0,in1,out);
input in0,in1;
output out;
assign out = in0&in1;
endmodule
```

EX2999

ERROR (EX2999): Another driver from here

If one output is constantly driven from multiple inputs, the synthesis tool will report the above error. In the following case, the out port is assigned twice.

```
module test (in0,in1,out);
input in0,in1;
output out;
assign out = in0 & !in1;
assign out = in0 & in1;
endmodule
```

Action

You need to delete one of them as shown below.

```
module test (in0,in1,out);
input in0,in1;
output out;
assign out = in0 & !in1;
endmodule
```

EX3000

ERROR (EX3000) : Net <object> is constantly driven from multiple places

If one output <object> is constantly driven from multiple inputs, the synthesis tool will report the above error. In the following case, the out port is assigned twice.

```
module test (in0,in1,out);
input in0,in1;
output out;
```

```
assign out = in0 & !in1;
assign out = in0 & in1;
endmodule
```

Action

You need to delete one of them as shown below.

```
module test (in0,in1,out);
input in0,in1;
output out;
assign out = in0 & !in1;
endmodule
```

EX3041

WARN (EX3041) : <object> shift count >= width of value

If there is a shift in the design, but the shift count is greater than the bit width of <object>, the above warning will pop up. In the following case, in1 is 1 bit width and shift left 2 bits, and the value is always 0 and is invalid.

```
module top (in0,in1,out);

input in0,in1;

output reg out;

assign out = in0&(in1 << 2);

endmodule
```

Action

You need to delete the invalid shift.

EX3044

WARN (EX3044) : Overwriting previous value of parameter <para>

If you assign a value to parameter <para> repeatedly, the above warning will pop up. In the following case, the INIT of LUT2 is assigned twice.

```
module top (in0,in1,out);

input in0,in1;

output reg out;

LUT2 lut2(
.I0(in0),

.I1(in1),

.F(out));

defparam lut2.INIT = 4'h4;

defparam lut2.INIT = 4'h6;

endmodule
```

Action

You need to delete one of them as shown below.

```
module top (in0,in1,out);

input in0,in1;

output reg out;

LUT2 lut2(

.I0(in0),

.I1(in1),

.F(out));

defparam lut2.INIT = 4'h6;

endmodule
```

EX3073

WARN (EX3073) : Port <port> remains unconnected for this instance

If the defined <port> is not in the instantiated ports list, the above warning will pop up. In the following case, the out1 port is not assigned in sub instance.

```
module top (top_in,top_out);
input top_in;
output top_out;
sub sub1(
.in(top_in),
.out0(top_out)
);
endmodule
module sub (in,out0,out1);
input in;
```

```
output out0;
output out1;
assign out0 = !in;
endmodule
```

Action

You need to delete out1 port or add out1 connection when instantiated as shown below.

```
module top (top_in,top_out);
input top_in;
output top_out;
sub sub1(
.in(top_in),
.out0(top_out)
);
endmodule
module sub (in,out0);
input in;
output out0;
assign out0 = !in;
endmodule
```

EX3359

ERROR (EX3359) : Null as source expression is not allowed here

If null is assigned to a signal, the synthesis tool will report the above error. Null is the keyword of systemVerilog.

```
module top (in,out);
input in;
output out;
assign  out= null;
endmodule
```

Action

You need to avoid using keyword.

EX3413

ERROR (EX3413): Second argument of '$<object> must be a memory

If the$<object> is used incorrectly in the design, the synthesis tool will report the above error. In the following case, the second parameter mem of the $readmemh should be a two-dimensional array.

```
module test(in,clk,addr_in,addr_out,out);
input in,clk;
input addr_in;
input addr_out;
output out;
reg [7:0]mem;
 always @ (posedge clk)
    mem[addr_in] <= in;
 assign out = mem[addr_out];
initial begin
    $readmemh("initvalue", mem);
  end
endmodule
```

Action

You need to define mem as a two-dimensional array as shown below.

```
module test(in,clk,addr_in,addr_out,out);
input in,clk;
input addr_in;
input addr_out;
output out;
reg [7:0]mem[7:0];
 always @ (posedge clk)
    mem[addr_in] <= in;
 assign out = mem[addr_out];
initial begin
    $readmemh("initvalue", mem);
  end
endmodule
```

EX3483

ERROR (EX3483) : Cannot open Verilog file <file>

If the design file does not exist or you have no permissions to open it, the synthesis tool will report the above error. You need to check the file or get the permission.

EX3514

ERROR (EX3514) : Module <modu> in library <lib> is not yet analyzed

If there is no specified module <modu> in the library <lib>, the synthesis tool will report the above error. You need to check the module name.

EX3534

ERROR (EX3534) : Assignment under multiple single edges is not supported for synthesis

If there is an assignment under multiple single edges in always statement, the synthesis tool will reporlt the above error. In the following case, the clk signal is under multiple single edges, which is not allowed in Gowin primitives.

```
module top (in,out,clk,clear);
input in,clk,clear;
output reg out;
always @(posedge clk or negedge clk)
    if(clear)
        begin
            out <= 1'b0;
        end
    else
        begin
            out <= in;
        end
endmodule
```

Action

You need to delete one of them as shown below.

```
module top (in,out,clk,clear);
input in,clk,clear;
```

```
output reg out;
always @(posedge clk)
    if(clear)
        begin
            out <= 1'b0;
        end
    else
        begin
            out <= in;
        end
endmodule
```

EX3589

ERROR (EX3589): Keyword <object> is not allowed here in this mode of Verilog

If the keyword <object> is not allowed here, the synthesis tool will report the above error.

EX3628

WARN (EX3628) : Redeclaration of ansi port <port> is not allowed

If the output port is assigned a rvalue, the above warning will pop up. In the following case, the ClkOut will perform an inverse operation as a rvalue in always.

```
module top(

    input ClkIn,

    input rst,

    output ClkOut

    );

    reg ClkOut;

    always@(posedge ClkIn)

    begin

    if(rst) ClkOut = 1'b0;

    else ClkOut = ~ClkOut;

    end
```

```
endmodule
```

Action

You can define a register and then assign a value to output port as shown below.

```
module top(
    input ClkIn,
    input rst,
    output ClkOut
    );
    reg tmp;
    always@(posedge ClkIn)
    begin
    if(rst) tmp = 1'b0;
    else tmp = ~tmp;
    end
    assign ClkOut = tmp;
endmodule
```

EX3638

WARN (EX3638) : <object> is already implicitly declared on line <lineInfo>

If the <object> is already implicitly declared on line <lineInfo>, but then explicitly declared, the above warning will pop up. In the following case, the wire tmp first implicitly declared then explicitly declared in an instance.

```
module top (in0,in1,out);
input in0,in1;
output  out;
aa ins(in0,in1,tmp);
wire tmp;
assign out = tmp;
endmodule
```

```
module aa(in0,in1,out);
input in0,in1;
output out;
assign out=in0|| in1;
endmodule
```

Action

You can put <object> ahead or use implicit declaration and delete explicit declaration.

```
module top (in0,in1,out);
input in0,in1;
output  out;
wire tmp;
aa ins(in0,in1,tmp);
assign out = tmp;
endmodule
module aa(in0,in1,out);
input in0,in1;
output out;
assign out=in0|| in1;
endmodule
```

EX3670

WARN (EX3670) : Actual bit length <actlen> differs from formal bit length <forlen> for port <port>

When <port> is instantiated, if the actual bit length <actlen> does not match with formal bit length <forlen>, the above warning will pop up. In the following case, the widths of the in port and top_in are different, and the widths of the out port and top_out are different.

```
module top (top_in,top_out);
input top_in;
output top_out;
test test1(
.in(top_in),
.out(top_out)
);
endmodule
module test (in,out);
input [2:0]in;
output [1:0]out;
assign out[0] = in[0];
assign out[1] = in[1] & !in[2];
endmodule
```

Action

You need to keep the widths consistent as shown below.

```
module top (top_in,top_out);
input [2:0]top_in;
output [1:0]top_out;
test test1(
.in(top_in),
.out(top_out)
);
endmodule
module test (in,out);
input [2:0]in;
output [1:0]out;
assign out[0] = in[0];
assign out[1] = in[1] & !in[2];
endmodule
```

EX3671

WARN (EX3671) : Second declaration of <object> ignored

If the <object> has been defined repeatedly, the above warning will pop up. In the following case, the out has been defined twice.

```
module top (in,out);
input in;
output out;
wireout;
wireout;
assign out = !in;
endmodule
```

Action

You need to delete one as shown below.

```
module top (in,out);
input in;
output out;
wireout;
assign out = !in;
endmodule
```

EX3680

WARN (EX3680): Concatenation with unsized literal,will interpret as 32 bits

If there is concatenation with undefined bit-width in your design, the above warning will pop up. In the following case, 'b0 has not defined and will use 32 bits, which may cause bit-width mismatches or missing bits.

```
module test (in,out);

input in;

output [15:0]out;

assign out = {'b0,in};

endmodule
```

Action

The bit width must be fixed as shown below.

```
module test (in,out);

input in;

output [15:0]out;

assign out = {15'b0,in};

endmodule
```

EX3682

WARN (EX3682) : Variable <vari> might have multiple concurrent drivers

If the output port <vari> has multiple drivers, the above warning will pop up. In the following case, the out port may have multiple drivers.

```
module top(in,sel,out);

input in,sel;
```

```
output reg out;

reg tmp;

always@(*)

    begin

        if(sel)

            tmp <= in;

        else

            out <= !tmp;

    end

assign out= tmp;

endmodule
```

Action

You need to avoid multiple inputs driving one port.

EX3705

WARN (EX3705) : Macro <object> redefined

If the <object> has been redefined by the define statement, the above warning will pop up. The later definition will replace the previous one. In the following case, INIT has been redefined resulting in different values for out0 and out1 ports.

```
`define INIT 1'b0
module test (in,out0,out1);
input in;
output out0,out1;
assign out0 = !in|`INIT;
`define INIT 1'b1
assign out1 = !in|`INIT;
endmodule
```

Action

You need to give two different definitions or use if define statement.

```
`define INIT0 1'b0
`define INIT1 1'b1
module test (in,out0,out1);
input in;
```

```
output out0,out1;
assign out0 = !in|`INIT0;
assign out1 = !in|`INIT1;
endmodule
```

EX3706

WARN (EX3706) : Empty port in <modu> declaration

If the <modu> declaration is incorrect, the above warning will pop up. In the following case, the declaration is followed by a comma.

```
module test(in,out,);

input in;

output out;

assign out = !in;

endmodule
```

Action

You need to delete the comma as shown below.

```
module test(in,out);

input in;

output out;

assign out = !in;

endmodule
```

EX3735

ERROR (EX3735) : Port <port> is already connected

If you repeatedly assign a value to a port <port> when instantiated, the synthesis tool will report the above error. In the following case, two different values are assigned to in port as shown below.

```
module test (in,out);
input in;
output out;
assign out = !in;
endmodule
module top (in0,in1,out0);
input in0,in1;
```

```
output out0;
test test1(
.in(in0),
.in(in1),
.out(out0)
);
endmodule
```

Action

You need to delete the repeated port assignment as shown below.

```
module test (in,out);
input in;
output out;
assign out = !in;
endmodule
module top (in0,out0);
input in0;
output out0;
test test1(
.in(in0),
.out(out0)
);
endmodule
```

EX3771

WARN (EX3771) : <modu> instance should have an instance name

If the instantiated module <modu> does not have a name, the above warning will pop up. In the following case, the test module is without a name and a default name will be given when synthesized.

```
module test (in,out);
input in;
output [1:0]out;
assign out = in+1'b1;
endmodule
module top (top_in,top_out);
input top_in;
output [1:0]top_out;
test (
.in(top_in),
.out(top_out)
);
endmodule
```

Action

You need to add a name to the instantiated module as shown below.

```
module test (in,out);
input in;
output [1:0]out;
assign out = in+1'b1;
endmodule
module top (top_in,top_out);
input top_in;
output [1:0]top_out;
test test1(
.in(top_in),
.out(top_out)
);
endmodule
```

EX3779

WARN (EX3779) : <signal> should be on the sensitivity list

If the always statement is used in the design but the <signal> is missing in the sensitivity list, the above warning will pop up. In the following case, in0 and in1 are sensitive signals and should be added to the always sensitivity list, or they will be added automatically when synthesized.

```
module top (sel,in0,in1,out);
input sel,in0,in1;
output reg out;
always@(sel /*or in0 or in1*/ )
    if(sel == 1'b0)
        begin
            out <= in0;
        end
    else
        begin
            out <= in1;
        end
endmodule
```

Action

You need to add the signal to the always sensitivity list as shown below.

```
module top (sel,in0,in1,out);
input sel,in0,in1;
output reg out;
always@(sel or in0 or in1)
    if(sel == 1'b0)
        begin
            out <= in0;
        end
    else
        begin
            out <= in1;
        end
endmodule
```

EX3780

WARN (EX3780) : Using initial value of <vari> since it is never assigned

If a register variable <vari> is defined in the design, but only the initial value is assigned as a constant, the above warning will pop up. In the following case, register tmp is used only as 1 'b0.

```
module test(in,out);

input in;

output reg out;

reg tmp;

initial begin

        tmp = 0;

    end

always@(in or tmp)

    if(in==tmp)

        out <= in;

    else

        out <= !in;

endmodule
```

Action

You need to replace tmp with 1’b0 as shown below.

```
module test(in,out);
input in;
output reg out;
always@(in)
    if(in== 1'b0)
        out <= in;
    else
        out <= !in;
endmodule
```

EX3784

WARN (EX3784) : Index <width> is out of range <range> for <port>

If the <width> is out of the range <range> for <port>, the above warning will pop up. In the following case, the out width is from 0 to1, and a value is assigned to out[2], which will be ignored when synthesized.

```
module test (in,out);
input [2:0]in;
output [1:0]out;
assign out[0] = in[0];
assign out[2] = in[1] & !in[2];
assign out[1] = in[1] & in[2];
endmodule
```

Action

You need to modify the out width or delete the undefined width as shown below.

```
module test (in,out);
input [2:0]in;
output [2:0]out;
assign out[0] = in[0];
assign out[2] = in[1] & !in[2];
assign out[1] = in[1] & in[2];
endmodule
```

EX3786

ERROR (EX3786) : Assignment to input <port>

If a value is assigned to the input <port> in the design, the synthesis tool will report the above error. The input port d is driven by other input port in the following case.

```
module test(b,c,d,f);
input b,c,d;
output f;
assign d = c&b;
assign f = b&d;
endmodule
```

Action

You need to delete it as shown below.

```
module test(b,d,f);
input b,d;
output f;
assign f = b&d;
endmodule
```

EX3791

WARN (EX3791): Expression size <size> truncated to fit in target size <tarSize>

If the data width <tarSize> before assignment is different from data width <size> after assignment in the design, the above warning will pop up. In the following case, the data width of out is 1, and the data width of in0 & in1 is 3, at which time the additional width of in0 & in1 is invalid.

```
module top (in0,in1,clk,out);
input [2:0]in0,in1;
input clk;
output reg out;
always @( posedge clk)
    begin
        out <= in0&in1;
    end
endmodule
```

Action

You need to modify the width of out or in0 & in1 as shown below.

```
module top (in0,in1,clk,out);
input [2:0]in0,in1;
input clk;
output reg [2:0] out;
always @( posedge clk)
    begin
       out <= in0&in1;
    end
endmodule
```

EX3792

WARN (EX3792): Literal value truncated to fit in <num> bits

If the parameter value <num> defined in the design is out of range, the above warning will pop up. In the following case, the INIT value of LUT2 ranges from 4 'h0 to 4' hF, and the 4 'h14 is an illegal value. The last four bits are 4' h4 when synthesized.

```
module top (in0,in1,out);
input in0,in1;
output out;
LUT2 lut2(
.I0(in0),
.I1(in1),
.F(out)
);
defparam lut2.INIT = 4'h14;
endmodule
```

Action

You need to modify the value to a reasonable range as shown below.

```
module top (in0,in1,out);
input in0,in1;
output out;
LUT2 lut2(
.I0(in0),
.I1(in1),
.F(out)
);
defparam lut2.INIT = 4'h4;
endmodule
```

EX3794

ERROR (EX3794) : Overwriting previous definition of module <modu>

If there are two modules <modu> of the same name, the synthesis tool will report the above error.

```
module test (in0,in1,out);
input in0,in1;
output out;
assign out = in0 & !in1;
endmodule
module test (data,out);
input data;
output out;
assign out = !data;
endmodule
```

Action

You need to modify one of them as shown below.

```
module test (in0,in1,out);
input in0,in1;
output out;
assign out = in0 & !in1;
endmodule
module test0 (data,out);
input data;
output out;
assign out = !data;
endmodule
```

EX3818

ERROR (EX3818): <inst> expects <num> arguments

If the synthesis tool reports the above error, it indicates that the number of ports given by the instantiated module exceeds the required. In the following case, the ALU needs six ports, and in fact, seven ports are given.

```
module alu_1bit(a,b,din1,din2,sum,cout);
input din1,din2,a,b;
output cout,sum;
ALU sum_cry_0_0 (cout, sum, 0, din2, din1, a, b);
```

```
defparam sum_cry_0_0.ALU_MODE=0;
endmodule
```

Action

You need to change the number of ports as shown below.

```
module alu_1bit(a,b,din1,din2,sum,cout);
input din1,din2,a,b;
output cout,sum;
ALU sum_cry_0_0 (cout, sum, din2, din1, a, b);
defparam sum_cry_0_0.ALU_MODE=0;
endmodule
```

EX3827

WARN (EX3827): Full_case directive is effective : might cause synthesis - simulation differences

If full_case is used, the above warning will pop up. It might cause synthesis-simulation differences.

```
module top (sel,in0,out);

input sel,in0;

output reg out;

always@(sel or in0)

    begin

        case(sel)/*synthesis full_case*/

            1'b0:

                begin

                    out <= in0;

                end

        endcase

    end

endmodule
```

Action

You need to complete other case conditions.

EX3829

ERROR (EX3829) : Port <port> is not defined

If the declared <port> has not been added to the ports list, the synthesis tool will report the above error, such as out1 in the following case.

```
module test (in,out);
input in;
output out;
output out1;
assign out = !in;
endmodule
```

Action

You need to delete out1 declaration as shown below.

```
module test (in,out);
input in;
output out;
assign out = !in;
endmodule
```

EX3833

ERROR (EX3833): If-condition does not match any sensitivity list edge

If there are multiple sensitivity signals in always statement and the if-condition does not match any signal in the list, the synthesis tool will report the above error. In the following case, the clear signal is out of the list.

```
module top (in,out,clk1,clk2,clear);
input in,clk1,clk2,clear;
output reg out;
always @(posedge clk1 or posedge clk2)
begin
    if(clear)
        out <= 1'b0;
    else
        out <= in;
end
endmodule
```

Action

You need to remove clk2 and add clear to the sensitivity list as shown below.

```
module top (in,out,clk1,clear);
input in,clk1,clear;
output reg out;
always @(posedge clk1 or posedge clear)
begin
    if(clear)
        out <= 1'b0;
    else
        out <= in;
end
endmodule
```

EX3834

WARN (EX3834): Case condition never applies

If the case statement is used in the design but some conditions never apply, the above warning will pop up. In the following case, the 3 'b101 bit width is different from the sel signal, and this condition will never occur.

```
module top (sel,in0,in1,out);
input sel,in0,in1;
output reg out;
always@(sel or in0 or in1 )
    begin
        case(sel)
            1'b0:
                begin
                    out <= in0;
                end
            1'b1:
                begin
                    out <= in1;
                end
            3'b101:
                begin
                    out <= 1'b0;
                end
        endcase
    end
endmodule
```

Action

You need to remove the useless condition as shown below.

```
module top (sel,in0,in1,out);
input sel,in0,in1;
output reg out;
always@(sel or in0 or in1 )
    begin
        case(sel)
            1'b0:
                begin
                    out <= in0;
                end
            1'b1:
                begin
                    out <= in1;
                end
        endcase
    end
endmodule
```

EX3858

WARN (EX3858) : System task <task> ignored for synthesis

If there is task statement <task> in the design, the <task> will be ignored in synthesis, and the synthesis tool will report the above error.

EX3863

ERROR (EX3863) : Syntax ERROR near <object>

If there is syntax error near <object>, the synthesis tool will report the above error. In the following case, the first line should be followed by a semicolon.

```
module test (in,out)
input in;
output out;
assign out = !in;
endmodule
```

Action

You need to check the syntax to find the reason.

```
module test (in,out);
```

```
input in;
output out;
assign out = !in;
endmodule
```

EX3864

WARN (EX3864) : <port> was previously declared with a different range

If a port is declared in the design, but it also declared as a wire or a register with different widths, the above warning will pop up. In the following case, out is defined as wire with different widths.

```
module top(in,out);

input in;

output [1:0]out;

wire [2:0] out = 3'b0+in;

endmodule
```

Action

You need to keep the port width and wire width consistent as shown below.

```
module top(in,out);

input in;

output [2:0]out;

wire [2:0] out = 3'b0+in;

endmodule
```

EX3872

ERROR (EX3872) : <port> is not declared

If <port> is not defined and used after declaration, the synthesis tool will report the above error. The out1 is not used after declaration in the following case.

```
module test (in,out,out1);
input in;
output out;
assign out = !in;
```

```
endmodule
```

Action

You need to delete the undeclared port or add the port definition.

```
module test (in,out);
input in;
output out;
assign out = !in;
endmodule
```

EX3875

ERROR (EX3875) : No definition for port <port>

If there is no definition for port <port>, the synthesis tool will report the above error. In the following case, there is no I/O definition for out.

```
module test(in,out);
input in;
assign out = !in;
endmodule
module top (top_in,top_out);
input top_in;
output top_out;
test test1(
.in(top_in),
.out(top_out)
);
endmodule
```

Action

You need to define the I/O for out as shown below.

```
module test(in,out);
input in;
output out;
assign out = !in;
endmodule
module top (top_in,top_out);
input top_in;
output top_out;
test test1(
.in(top_in),
.out(top_out)
);
```

```
endmodule
```

EX3900

ERROR (EX3900) : Procedural assignment to a non-register <nreg> is not permitted

If there is an nonblocking assignment to a non-register <nreg>, the synthesis tool will report the above error. For example, the out should be reg out in the following case.

```
module top (in,out,clk);
input in,clk;
output out;
wire out;
always @( posedge clk)
    begin
        out <= in;
    end
endmodule
```

Action

You need to modify wire out to reg out as shown below.

```
module top (in,out,clk);
input in,clk;
output out;
reg out;
always @( posedge clk)
    begin
        out <= in;
    end
endmodule
```

EX3902

ERROR (EX3902) : Port <port> is already defined

If the <port> has been defined repeatedly, the synthesis tool will report the above error. In the following case, the port out has been defined twice.

```
module top (in,out);
input in;
output out;
output out;
assign out = !in;
```

```
endmodule
```

Action

You need to delete one as shown below to avoid this error.

```
module top (in,out);
input in;
output out;
assign out = !in;
endmodule
```

EX3907

ERROR (EX3907) : Parameter <para> is not defined in this module

If an instance set a <para> that has not defined by this module, the synthesis tool will report the above error. It can be found and modified by the name and line information. In the following case, ins1 sets INIT_0 parameter, which is out of module DFF parameters.

```
module test(a,clk,out);
input a,clk;
output out;
DFF ins1(
    .D(a),
    .CLK(clk),
    .Q(out)
);
defparam ins1.INIT=1'b0;
defparam ins1.INIT_0=8'hEF;
endmodule
```

Action

You need to delete this parameter as shown below.

```
module test(a,clk,out);
input a,clk;
output out;
DFF ins1(
    .D(a),
    .CLK(clk),
    .Q(out)
);
defparam ins1.INIT=1'b0;
endmodule
```

EX3916

WARN (EX3916) : No support for synthesis of mixed edge and level triggers. Assume level triggers only.

If always statement contains both mixed edge and level triggers, the edge trigger will be ignored. The synthesis tool will report the above error.

```
module top (in,out,clk,clear);
input in,clk,clear;
output reg out;
always @(posedge clk or clear)
    if(clear)
        begin
            out <= 1'b0;
        end
    else
        begin
            out <= in;
        end
endmodule
```

Action

You need to remove the clear signal from the sensitivity list as shown below.

```
module top (in,out,clk,clear);
input in,clk,clear;
output reg out;
always @(posedge clk)
    if(clear)
        begin
            out <= 1'b0;
        end
    else
        begin
            out <= in;
        end
endmodule
```

EX3927

ERROR (EX3927) : Module <modu> remains a black box, due to ERRORs in its contents

This error pops up with other errors at the same time. If the synthesis tool reports the above error, it indicates that there is still other error in the <modu>, and synthesis error exits.

EX3928

ERROR (EX3928) : Module <modu> ignored due to previous ERRORs

This error pops up with other errors at the same time. If the synthesis tool reports the above error, it indicates that there is still other error in the <modu>, and synthesis error exits.

EX3937

ERROR (EX3937) : Instantiating unknown module <modu>

If you instantiate an unknown module in your design, the synthesis tool will report the above error. In the following case, the test module has not been defined.

```
module top (in,out);
input in;
output out;
test test1(
.in0(in),
.out0(out)
);
endmodule
```

Action

You need to add <modu> definition to avoid this error. The module implementation can be null. If it is null, it will be converted to black box as shown below.

```
module top (in,out);
input in;
output out;
test test1(
.in0(in),
.out0(out)
);
endmodule
module test(in0,out0);
input in0;
output out0;
```

```
assign out0 = !in0;
endmodule
```

EX3945

ERROR (EX3945) : Incorrect use of predefined macro <include>. Expected <filePath>

If the <filePath> specified by <include> is incorrect, the synthesis tool will report the above error. In the following case, there is no double quotation mark on the file path.

```
`include param.v;

module top(in,sel,out);

input in,sel;

output reg [size:0]out;

assign out = in+sel;

endmodule

//param.v 文件内容

/*

parameter size = 2;

*/
```

Action

You need to add the double quotation mark on the file path in include as shown below.

```
`include "param.v";

module top(in,sel,out);

input in,sel;

output reg [size:0]out;

assign out = in+sel;

endmodule

//param.v 文件内容

/*

parameter size = 2;
```

```
*/
```

EX3983

WARN (EX3983) : Case condition never applies due to comparison with x or z

If the case statement you used contains X and Z in your design, the above warning will pop up. In the following example, the case contains X and Z, which are ignored when synthesized.

```
module top (sel,in0,in1,out);
input sel,in0,in1;
output reg out;
always@(sel or in0 or in1 )
    begin
        case(sel)
            1'b0:
                begin
                    out <= in0;
                end
            1'b1:
                begin
                    out <= in1;
                end
            1'bX:
                begin
                    out <= 1'b0;
                end
            1'bZ:
                begin
                    out <= 1'b1;
                end
        endcase
    end
endmodule
```

Action

You need to delete the X and Z as shown below.

```
module top (sel,in0,in1,out);
input sel,in0,in1;
output reg out;
always@(sel or in0 or in1 )
    begin
```

```
        case(sel)
            1'b0:
                begin
                    out <= in0;
                end
            1'b1:
                begin
                    out <= in1;
                end
        endcase
    end
endmodule
```

EX3988

WARN (EX3988) : Cannot open file <file>

If there is no configuration file <file> or you have no permission to open it, the above warning will pop up. In the following case, initvalue file of $readmemh can not open.

```
module test(in,clk,addr_in,addr_out,out);
input in,clk;
input addr_in;
input addr_out;
output out;
reg [7:0]mem [7:0];
 always @ (posedge clk)
     mem[addr_in] <= in;
 assign out = mem[addr_out];
initial begin
    $readmemh("initvalue", mem);
  end
endmodule
```

Action

You need to check the file or get the permission.

IF0003

ERROR (IF0003) : Cannot infer <signal> due to multiple write clocks

RAM Interface can only support up to two write clocks. If it exceeds two, the synthesis tool will report the above error.

```
    module normal5(data_out0,data_out1, data_in0,
data_in1,data_in2,addr,addr0, addr1,addr2,clk0,clk1,clk2,ce, wre,rst);
    input [2:0]data_in0;
    input [2:0]data_in1;
    input [2:0]data_in2;
    input [3:0]addr,addr0, addr1,addr2;
    input clk0,clk1,clk2,wre,ce,rst;
    reg [2:0] mem [7:0] ;
    output reg [2:0] data_out0;
    output reg [2:0] data_out1;
    always@(posedge clk0)
         if(ce==1 & wre == 0)
            data_out0 <= mem[addr0];
    always@(posedge clk1)
         if(ce==1 & wre == 0)
            data_out1 <= mem[addr1];
     always @(posedge clk0)
        if (ce & wre) mem[addr0] <= data_in0;
     always @(posedge clk1)
        if (ce & wre) mem[addr1] <= data_in1;
     always @(posedge clk1)
        if (ce & wre) mem[addr2] <= data_in2;
    endmodule
```

Action

You can not write more than two sets of data to one ARM. After optimized, if it is still more than two, you need to modify rtl.

Place & Route User Messages

CT1000

WARN (CT1000) : <file>:<line> | This constraint of <name> is defined again, so this will overwrite the previous

The constraint has already defined, and the later will overwrite the previous.

```
INS_LOC uut R3C4;
INS_LOC uut R4C5;
```

Action

You need to delete one of them.

```
INS_LOC uut R4C5;
```

CT1003

WARN (CT1003) : <file>:<line> | Group(<name>) location is already defined, so this will overwrite the previous

The group constraint has already defined, and the later will overwrite the previous.

```
GROUP grp = {“ins1” “ins2”}
GRP_LOC grp R3C[3:5];
GRP_LOC grp R[4:5]C8;
```

Action

You need to delete one of them.

```
GROUP grp = {“ins1” “ins2”}
GRP_LOC grp R[4:5]C8;
```

CT1005

WARN (CT1005) : Conflicting multiple constraints specified for location of Instance <name>(type: <type>); Or constrained location for the Instance is not available; Or constrained location type is not matched with the instance

There are constraints conflicting, or the constrained location is not available, or the constrained location type does not match with the instance.

Action

You need to modify the file to avoid conflicts and constrain the object to an available location.

CT1007

WARN (CT1007) : There is no intersection between multiple group

constraints specified for instance <name>

There is no intersection location between multiple group constraints specified for instance <name>, resulting in an incorrect constraint location for <name>.

```
GROUP  grp1 = {“ins1” “ins2” “ins3”};

GRP_LOC  grp1  R2C[5:6];

GROUP  grp2 = {“ins1” “ins4”};

GRP_LOC  grp2  R4C[5:6];
```

Action

You need to modify the file to avoid putting a group member to multiple groups.

```
GROUP  grp1 = {“ins2” “ins3”};

GRP_LOC  grp1  R2C[5:6];

GROUP  grp2 = {“ins1” “ins4};

GRP_LOC  grp2  R4C[5:6];
```

CT1097

WARN (CT1097) : <file>:<line> | Please define group <name> first before define the constraint at line <number>

Constrain the location for undefined group.

```
GRP_LOC  grp1  R2C[5:6];
```

Action

Define the group first before location constraint.

```
GROUP  grp1 = {“ins2” “ins3”};

GRP_LOC  grp1  R2C[5:6];
```

CT1098

WARN (CT1098) : <file>:<line> | Group name <name> is already defined

The group has already been defined.

```
GROUP  grp1 = {“ins2” “ins3”};
```

```
REL_GROUP  grp1 = {“ins4” “ins5”};

GRP_LOC  grp1  R2C[5:6];
```

Action

You need to modify the constraint file to avoid repeated definition.

```
GROUP  grp1 = {“ins4” “ins5”};

GRP_LOC  grp1  R2C[5:6];
```

CT1101

WARN (CT1101) : <file>:<line> | Location column <number> is out of chip range(<maxColumn>)

The location column is out of the chip range.

Action

You need to modify the file to make the column in the range.

CT1102

WARN (CT1102): <file>:<line> | Location row <number> is out of the chip range(<maxRow>)

The location row is out of the chip range.

Action

You need to modify the file to make the row in the range.

CT1108

WARN (CT1108) : <file>:<line> | Illegal port attribute value specified <attribute> = <value> on <instName>

The value does not match with attribute.

```
IO_PORT  bufIns  DRIVE=20;
```

Action

You need to modify the attribute value.

```
IO_PORT  bufIns  DRIVE=8;
```

CT1111

WARN (CT1111) : Instance <name>(<type>) constrained to

unsuitable location

The constraint is constrained to an unsuitable location.

```
INS_LOC  dll_inst_2  PLL_R;
```

Action

You need to constrain it to an suitable location according to the type.

```
INS_LOC  dll_inst_2  DLL_BR;
```

CT1112

WARN (CT1112) : <file>:<line> | Invalid range location <location>, please constrained in the same side

The start and end should be constrained in the same side.

```
INS_LOC bufIns IOR4:IOL9;
```

Action

You need to make the start and end in the same side.

```
INS_LOC bufIns IOR4:IOR9;
```

CT1113

WARN (CT1113) : <file>:<line> | Cannot find pad location <pin> in current package

You can not find pad location in current package.

Action

You need to modify to make the pad location available in current package.

CT1115

WARN (CT1115) : Attribute <name> can only be set when the port is located to bank <index>. Please set the corresponding location constraint of port <portName>

The corresponding location constraint should be first when the port attribute is constrained.

```
IO_PORT i0 IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

Action

The location constraint is first then comes the attribute constraint.

```
    IO_LOC i0 IOT4;

    IO_PORT i0 IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

CT1116

WARN (CT1116) : Attribute <name> can only be set when the port is located to bank <index>. Please set the corresponding location constraint of port <portName> or <portName>

The corresponding location constraint should be first when the differential port attribute is constrained.

```
    IO_PORT I IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

Action

The location constraint of IB or its differential I constraint is first then comes the attribute constraint.

```
    IO_LOC IB IOT4;

    IO_PORT I IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

Or

```
    IO_LOC I IOT4;

    IO_PORT I IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

CT1117

WARN (CT1117) : Attribute <name> can only be set when the port is located to bank <index>, but the constraint location of port <portName> include other bank

The attribute value does not match with the location.

```
    IO_LOC i0 IOB4;

    IO_PORT i0 IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

Action

You need to modify the attribute or location constraints.

```
    IO_LOC i0 IOT4;

    IO_PORT i0 IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

CT1118

WARN (CT1118) : Attribute <name> can only be set when the port is located to bank <index>, but the constraint location of port <portName> or <portName> include other bank

The value does not match with the location.

```
IO_LOC I IOB4;

IO_PORT I IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

Action

You need to modify the attribute or location constraints of I or its differential IB.

```
IO_LOC I IOT4;

IO_PORT I IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

OR

```
IO_LOC IB IOT4;

IO_PORT I IO_TYPE=RSDS25E DIFF_RESISTOR=ON;
```

FS1008

WARN (FS1008) : Device <device type> is not supported AES encryption, please uncheck in bitstream configurations

The device does not support AES encryption.

Action

You need to cancel encryption.

FS2001

ERROR (FS2001) : Cannot read corrupted fse file

It can not read fse file.

Action

The fse file must match with the software and please do not delete or modify fse file.

PA1000

WARN (PA1000) :Dangling net <netName> in module <moduleName> has no source instance

The net in module has no source instance.

```
module test (i0,i1,i2,i3,out);

input i0;

input i1;

input i2;

input i3;

output out;

wire i4;

LUT4 uut (

    .I0(i0),

    .I1(i1),

    .I2(i2),

    .I3(i4),

    .F(out)

);

endmodule
```

Action

You need to ensure each net has a signal source. If the net should be dangling in the design, ignore this warning.

```
module test (i0,i1,i2,i3,out);

input i0;

input i1;

input i2;

input i3;

output out;

LUT4 uut (

    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out)
);
endmodule
```

PA1001

WARN (PA1001) : Dangling net <netName>(source:<instanceName>) in module <moduleName> has no destination

The net in the specified module has no destination.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
wire out_c;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
endmodule
```

Action

You need to ensure the signal has destination. If the net should be dangling, ignore this warning.

```
module test (i0,i1,i2,i3,out);
```

```
input i0;

input i1;

input i2;

input i3;

output out;

wire out_c;

LUT4 uut (

    .I0(i0),

    .I1(i1),

    .I2(i2),

    .I3(i3),

    .F(out_c)

);

OBUF buf_ins (

    .I(out_c),

    .O(out)

);

endmodule
```

PA1002

WARN (PA1002): <file>:<line> | Invalid parameterized value <value>(<parameter>) specified for instance <instanceName>

The specified instance sets invalid parameter.

Action

You need to set valid parameter.

PA1008

WARN (PA1008): <file>:<line> | Object <name> is already defined

The wire or port has already defined in the specified location.

```
module test (i0,i1,i2,i3,out);

input i0;
input i1;

input i2;

input i3;

input i3;

output out;

LUT4 uut (

    .I0(i0),

    .I1(i1),

    .I2(i2),

    .I3(i3),

    .F(out)

);

endmodule
```

Action

You need to delete one of the definition.

```
module test (i0,i1,i2,i3,out);

input i0;

input i1;

input i2;

input i3;

output out;

LUT4 uut (

    .I0(i0),

    .I1(i1),

    .I2(i2),

    .I3(i3),

    .F(out)

);

endmodule
```

PA1010

WARN (PA1010): <file>:<line> | Dangling pin(<name>) is not connect with net

The pin of the specified instance is not connected.

```
module test (i0,i1,i2,i3,out);

input i0;

input i1;

input i2;

input i3;

output out;

LUT4 uut (

    .I0(i0),

    .I1(i1),

    .I2(i2),

    .I3(i3),

    .F()

);

endmodule
```

Action

You need to make the pin connection correct. If the pin is dangling, ignore this warning.

```
module test (i0,i1,i2,i3,out);

input i0;

input i1;

input i2;

input i3;

output out;

LUT4 uut (

    .I0(i0),

    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out)
);
endmodule
```

PA2000

ERROR (PA2000): <file>:<line> | Syntax error near token <name>

There is a syntax error in the specified location.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
ouput out;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out)
);
endmodule
```

Action

You need to check and correct the syntax according to the error.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
```

```
output out;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out)
);
endmodule
```

PA2001

ERROR (PA2001): <file>:<line> | Module <moduleName> is already defined

The module has already defined.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out)
);
endmodule

module test (I0,I1,OUT);
```

```
input I0;
input I1;
output OUT;
LUT2 uut (
    .I0(I0),
    .I1(I1),
    .F(OUT)
);
endmodule
```

Action

You need to modify one of the module names.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out)
);
endmodule

module testLut2 (I0,I1,OUT);
input I0;
input I1;
output OUT;
```

```
LUT2 uut (
    .I0(I0),
    .I1(I1),
    .F(OUT)
);
endmodule
```

PA2004

ERROR (PA2004): <file> | In module <name>: Net <netName> driven by multiple source instances

The net in the specified module is driven by multiple source instances.

```
module test (i0,i1,i2,i3,out, out1);
input i0;
input i1;
input i2;
input i3;
output out;
output out1;
wire out_c;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
LUT2 uut2 (
    .I0(i0),
    .I1(i1),
    .F(out_c)
);
OBUF bufIns (
    .I(out_c),
    .O(out)
);
endmodule
```

Action

You need to modify the connection in the specified module.

```
module test (i0,i1,i2,i3,out,out1);
input i0;
input i1;
input i2;
input i3;
output out;
output out1;
wire out1_c;
wire out_c;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
LUT2 uut2 (
    .I0(i0),
    .I1(i1),
    .F(out1_c)
);
OBUF bufIns (
    .I(out_c),

    .O(out)

);

OBUF bufIns1 (

    .I(out1_c),

    .O(out1)

);

endmodule
```

PA2009

ERROR (PA2009): The port <name> connected to <instName>(instType) defined error direction which should be <portType> according to connection

The port type can not match with the instance port.

```
    module test (i0,i1,i2,i3,out);

    input i0;

    input i1;

    input i2;

    input i3;

    input out;

    LUT4 uut (

        .I0(i0),

        .I1(i1),

        .I2(i2),

        .I3(i3),

        .F(out)

    );

    endmodule
```

Action

You need to change the connection or type.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out)
);
endmodule
```

PA2014

ERROR (PA2014): Pin(<name>) of <instName>(<instType>) does not connect to port

The pin of the specified instance is not connected to the port.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
wire i0_c;
wire VCC;
wire io;
LUT4 uut (
    .I0(i0_c),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out)
);
IOBUF bufIns (
    .I(i0),
    .O(i0_c),
    .IO(io),
    .OEN(VCC)
);
VCC vcc (
    .V(VCC)
);
endmodule
```

Action

You need to connect the pin to the port.

```
module test (i0,i1,i2,i3,out,io);
input i0;
input i1;
input i2;
input i3;
output out;
inout io;
wire i0_c;
wire VCC;
wire io;
LUT4 uut (
    .I0(i0_c),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out)
);
IOBUF bufIns (
    .I(i0),
    .O(i0_c),
    .IO(io),
    .OEN(VCC)
);
VCC vcc (
    .V(VCC)
);
endmodule
```

PA2017

ERROR (PA2017): The number(<value>) of <instType> in the design exceeds the resource limit(<maxValue>) of current device

The number of <instType> in the design exceeds the resource limit.

Action

You need to reduce the number of the instances or use a device with larger resources.

PA2024

ERROR (PA2024): The number(<value>) of ports exceeds the resource limit <maxValue> regular I/Os(include <value> dedicated I/Os) and <value> shared I/Os of current device

The number of ports in top module exceeds the one of the current device.

Action

You need to use other package or use a device with large resources.

PA2025

ERROR (PA2025): No <instType> resource in current device

There is resource that is not supported by the device in the design.

Action

You need use other series of device supporting this resource.

PA2039

ERROR (PA2039): Net <name> is used in module <moduleName> but not declared in wire list

The net used in module has not declared.

```
module test (i0,i1,i2,i3,out);

input i0;

input i1;

input i2;

input i3;

output out;

LUT4 uut (

    .I0(i0),

    .I1(i1),

    .I2(i2),

    .I3(i3),

    .F(out_c)

);

OBUF obufIns (

    .O(out),

    .I(out_c)

);

endmodule
```

Action

You need to declare the net.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
wire out_c;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
OBUF obufIns (
    .O(out),
    .I(out_c)
);
endmodule
```

PA2054

ERROR (PA2054): <file>:<line> | <name> is already declared

The instance name has already declared in the design.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
wire out_c;
```

```
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
OBUF uut (
    .O(out),
    .I(out_c)
);
endmodule
```

Action

You need to change the name to avoid repeated name.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
wire out_c;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
OBUF obufIns (
    .O(out),
    .I(out_c)
);
endmodule
```

PA2056

ERROR (PA2056): <file>:<line> | Error pin name(<name>) found in instance <instName>

The pin name of instance does not match with the primitive.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
wire out_c;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I4(i3),
    .F(out_c)
);
OBUF bufIns (
    .O(out),
    .I(out_c)
);
endmodule
```

Action

You need to check the pin and correct the name.

```
module test (i0,i1,i2,i3,out);
```

```
input i0;

input i1;

input i2;

input i3;

output out;

wire out_c;

LUT4 uut (

    .I0(i0),

    .I1(i1),

    .I2(i2),

    .I3(i3),

    .F(out_c)

);

OBUF bufIns (

    .O(out),

    .I(out_c)

);

endmodule
```

PA2058

ERROR (PA2058): <file>:<line> | Error pin number within instance <name>(<type>) of module <name>

The number of instance pins is incorrect.

```
module test (i0,i1,i2,i3,out);

input i0;

input i1;

input i2;

input i3;

output out;

wire out_c;
```

```
LUT4 uut (
    .I0({i0,i1}),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
OBUF bufIns (
    .O(out),
    .I(out_c)
);
endmodule
```

Action

You need to delete or add pins as required.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
wire out_c;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
OBUF bufIns (
    .O(out),
    .I(out_c)
);
endmodule
```

PA2066

ERROR(PA2066): <file>:<line> | Invalid parameter name <name> setting to object <instName>

The parameter set is invalid.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
wire out_c;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
defparam uut.INIT_1=16'h0000;
OBUF bufIns (
    .O(out),
    .I(out_c)
);
endmodule
```

Action

You need to set valid parameter as shown below.

```
module test (i0,i1,i2,i3,out);
input i0;
input i1;
input i2;
input i3;
output out;
wire out_c;
LUT4 uut (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .F(out_c)
);
defparam uut.INIT=16'h0000;
OBUF bufIns (
    .O(out),
    .I(out_c)
);
endmodule
```

PR0026

ERROR (PR0026) : CLKOUTN pin of <name> is not connected to any other iologic

The CLKOUTN pin of DHCENC is not driving any IOLOGIC.

Action

You need to make the CLKOUTN pin of DHCENC drive IOLOGIC reasonably.

PR0027

ERROR (PR0027) : Instance <name> connected to CLKIN pin of

instance <name> is unsupported

The instance does not support to connect CLKIN pin of PLL or DLL.

Action

You need to make this instance not connect CLKIN pin of PLL or DLL.

PR0028

ERROR (PR0028) : Instance <name> connected to CLKFB pin of instance <name> is unsupported

The instance does not support to connect CLKFB pin of PLL.

Action

You need to make this instance not connect CLKFB pin of PLL.

PR0029

ERROR (PR0029) : Instance <name>(INS_DHCENC) cannot drive two CLKDIVs

DHCENC can not drive two CLKDIVs at the same time.

Action

You need to modify the design so that DHCENC do not drive two CLKDIVs at the same time.

PR1011

ERROR (PR1011): Failed to capture gao signal:<name>, because there's no wire to route for the signal

It fails to capture GAO signal. In the following case, there is no wire to route the signal "c0_c".

```
module test (i0, i1, i2, i3, o0, o1, o2);

input i0, i1, i2, i3;

output o0, o1, o2;

wire i0_c, i1_c, i2_c, i3_c, c0_c, c1_c, s0_c, s1_c, GND;

GND GND_C(.G(GND));

IBUF ibuf_i0(.I(i0), .O(i0_c));

IBUF ibuf_i1(.I(i1), .O(i1_c));
```

```
   IBUF ibuf_i2(.I(i2), .O(i2_c));
   IBUF ibuf_i3(.I(i3), .O(i3_c));
   ALU
alu_0(.I0(i0_c),.I1(i1_c),.I3(GND),.CIN(GND),.COUT(c0_c),.SUM(s0_c));
   defparam alu_0.ALU_MODE = 0;
   ALU
alu_1(.I0(i2_c),.I1(i3_c),.I3(GND),.CIN(c0_c),.COUT(c1_c),.SUM(s1_c));
   defparam alu_1.ALU_MODE = 0;
   OBUF obuf_sum0(.I(s0_c), .O(o0));
   OBUF obuf_sum1(.I(s1_c), .O(o1));
   OBUF obuf_cout(.I(c1_c), .O(o2));
   endmodule
```

Action

You need to make sure the signal can be captured. When it can not be captured, you can capture the signal from the previous level or backward level. In the following case, you can capture I0 and I1 to analyze signal "c0_c".

```
   module test (i0, i1, i2, i3, o0, o1, o2);
   input i0, i1, i2, i3;
   output o0, o1, o2;
   wire i0_c, i1_c, i2_c, i3_c, c0_c, c1_c, s0_c, s1_c, GND;
   GND GND_C(.G(GND));
   IBUF ibuf_i0(.I(i0), .O(i0_c));
   IBUF ibuf_i1(.I(i1), .O(i1_c));
   IBUF ibuf_i2(.I(i2), .O(i2_c));
   IBUF ibuf_i3(.I(i3), .O(i3_c));
   ALU
alu_0(.I0(i0_c),.I1(i1_c),.I3(GND),.CIN(GND),.COUT(c0_c),.SUM(s0_c));
   defparam alu_0.ALU_MODE = 0;
   ALU
alu_1(.I0(i2_c),.I1(i3_c),.I3(GND),.CIN(c0_c),.COUT(c1_c),.SUM(s1_c));
   defparam alu_1.ALU_MODE = 0;
```

```
OBUF obuf_sum0(.I(s0_c), .O(o0));
OBUF obuf_sum1(.I(s1_c), .O(o1));
OBUF obuf_cout(.I(c1_c), .O(o2));
endmodule
```

PR1014

WARN(PR1014) : Generic routing resource will be used to clock signal <name> by the specified constraint. And then it may lead to the excessive delay or skew

Gowin Router detects that there is generic routing in clock routing, and it may lead to clock delay or skew. In the following case (GW1N-4), the signal clk_c source is constrained to non-clock port so that there is logical wire in clock.

```
top.vm
Module test_clk()
input i0, i1
output o0
IBUF ibuf_data(.I(i0), .O(d_c));
IBUF ibuf_clk(.I(i1), .O(clk_c));
DFF dff_c(.D(d_c), .CLK(clk_c), .Q(q_c));
OBUF obuf_c(.I(q_c), .O(o0));
endmodule


top.cst
IO_LOC “ibuf_data” IOB18A;
```

Action

You need to make sure that the source is clock source and the physical constraint location is clock port. In the following case (GW1N-4), you need to constrain clk_c to IOB20A.

```
top.vm
Module test_clk()
input i0, i1
```

```
output o0
IBUF ibuf_data(.I(i0), .O(d_c));
IBUF ibuf_clk(.I(i1), .O(clk_c));
DFF dff_c(.D(d_c), .CLK(clk_c), .Q(q_c));
OBUF obuf_c(.I(q_c), .O(o0));
endmodule


top.cst
IO_LOC “ibuf_data” IOB20A;
```

PR2044

WARN (PR2044) : FCLK port of <name> conflicts with FCLK port of <name> and <FCLK or HCLKIN> port of <name>

FCLK of first instance, FCLK of second instance and FCLK or HCLKIN of the third instance are not collinear.

Action

You need to make them collinear or find other available location.

PR2045

WARN (PR2045) : FCLK port of <name> conflicts with FCLK port of <name>

FCLK of the first instance, the FCLK of the second instance are not collinear.

Action

You need to make them collinear or find other available location.

PR2061

ERROR (PR2061) : There is no position to place <name>

There is no position to place instance.

Action

You need to optimize the placement of other instances to have position to place this instance.

PR2062

ERROR (PR2062): Objects drived by CLKOUT pin of <name> must be placed on same side

All IOLOGIC driven by CLKOUT of the same DHCENC should be placed on the same side.

Action

You need to find other available position.

PR2063

ERROR (PR2063) : Objects drived by CLKOUTN pin of <name> must be placed on same side with buffer <name>

All IOLOGIC driven by CLKOUTN of the same DHCENC should be placed on the same side with buffer.

Action

You need to find other available location.

PR2064

ERROR (PR2064) : Buffer <name> driving DHCENC must be placed to GCLK

The BUFFER driving DHCENC must be placed to GCLK location.

Action

You need to find other available GCLK location.

PR2065

ERROR (PR2065) : Buffer <name> driving DLLDLY must be placed to GCLK

The BUFFER driving DLLDLY must be placed to GCLK location.

Action

You need to find other available GCLK location.

PR2066

ERROR (PR2066) : Iologics need more than two hclk on <chip side>

IOLOGICs need more than two hclks, but the hclk resource is not enough on this side.

Action

You need to find other available location meeting hclk requirements.

PR2067

ERROR (PR2067) : Instance <name> must have constraint

GW1N-9C and GW1NR-9C require that the instance must be constrained.

Action

You need to add suitable constraint to this instance.

PR2068

ERROR (PR2068) : Instance <name> must have unique constraint

GW1N-9C and GW1NR-9C require that the instance must have an unique constraint.

Action

You need to add a unique constraint to this instance.

PR2069

ERROR (PR2069) : The constrainted location of <name> cannot be IO BLOCK

The constraint location of this instance can not be IOB.

Action

You need to find other available constraint location.

PR2070

ERROR (PR2070) : Instance <name> connected to IODELAYC cannot be placed to bottom side

The BUFFER connected to IODELAYC can not be constrained to the bottom side.

Action

You need to find other available location.

TA1001

WARN(TA1001) : Either option "-name" or option "<source objects>" should be specified

There is no specified object and clock name in create_clock constraint.

```
create_clock -period 10 -waveform {0 5}
```

Action

You need to modify sdc constraint and add specified object and clock name.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]
```

TA1004

WARN(TA1004) : Clock waveform should not contain two edges with time larger than one clock period

The clock edge specified by –waveform is greater than the period specified by –period in create_clock constraint.

```
create_clock -name clk1 -period 10 -waveform {0 15} [get_ports {clk}]
```

Action

You need to modify –waveform or –period to make sure the clock edge is in one clock period.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]
```

TA1006

WARN(TA1006) : The waveform list is not monotonically increasing

The waveform list is not monotonically increasing in create_clock constraint.

```
create_clock -name clk1 -period 10 -waveform {5 0} [get_ports {clk}]
```

Action

You need to modify –waveform in create_clock constraint to make sure the clock edge is monotonically increasing.

```
create_clock -name clk1 -period 10 -waveform {5 10} [get_ports {clk}]
```

TA1011

WARN(TA1011) : Option "-rise" and option "-fall" are mutually exclusive

-rise and –fall are used simultaneously in set_input_delay/set_output_delay constraint. -rise and –fall are mutually exclusive, and only one of them can be used in one constraint.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]
set_input_delay -clock clk1 1 -rise –fall -max [get_ports {in01}]
```

Action

You can only use one of them in one constraint.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]
set_input_delay -clock clk1 1 -rise -max [get_ports {in01}]
set_input_delay -clock clk1 1 -fall -max [get_ports {in01}]
```

TA1012

WARN(TA1012) : Option "-max" and option "-min" are mutually exclusive

-max and -min are used simultaneously in set_input_delay/set_output_delay constraint. –max and -min are mutually exclusive, and only one of them can be used in one constraint.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]
set_input_delay -clock clk1 1 -rise –max -min [get_ports {in01}]
```

Action

You can only use one of them in one constraint.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]
set_input_delay -clock clk1 1 -rise -max [get_ports {in01}]
set_input_delay -clock clk1 1 -rise -min [get_ports {in01}]
```

TA1016

WARN(TA1016):Options "-edges -edge_shift" and options "-divide_by -multiply_by -duty_cycle -phase -offset" are mutually exclusive: specify either of the two ways

There are two ways to create generated clock frequency and phrase in create_generated_clock constraint: -edges /-edge_shift, and -divide_by/ -multiply_by/ -duty_cycle/ -phase /-offset. The two ways are mutually exclusive.

```
   create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

   create_generated_clock -name genClk -source [get_ports {clk}]
-master_clock clk1 -edges {1 3 5} -edge_shift {1 1 1} -divide_by 2  [get_pins
{reg0_0_Z/Q}]
```

Action

You can only use one of the ways.

```
   create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

   create_generated_clock -name genClk -source [get_ports {clk}]
-master_clock clk1 -divide_by 2  [get_pins {reg0_0_Z/Q}]

   create_generated_clock -name genClk2 -source [get_ports {clk}]
-master_clock clk1 -edges {1 3 5} -edge_shift {1 1 1}  [get_pins
{reg0_0_Z/Q}] –add
```

TA1019

WARN(TA1019) : Option "-edges " must be in non-decreasing order

The edges parameters are in non-increasing order in create_generated_clock constraint.

```
   create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

   create_generated_clock -name genClk -source [get_ports {clk}]
-master_clock clk1 -edges {3 1 5} -edge_shift {1 1 1} [get_pins {reg0_0_Z/Q}]
```

Action

You need to make sure the edges parameters are in non-decreasing order in create_generated_clock constraint.

```
   create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

   create_generated_clock -name genClk -source [get_ports {clk}]
-master_clock clk1 -edges {1 3 5} -edge_shift {1 1 1} [get_pins {reg0_0_Z/Q}]
```

TA1027

WARN(TA1027) : Missing required clock latency delay

There is a missing required clock latency value in set_clock_latency constraint.

```
   create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

   set_clock_latency -source -late -fall  [get_ports {clk}] -clock
[get_clocks {clk1}]
```

Action

You need to specify the clock latency value.

```
   create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

   set_clock_latency -source -late -fall  [get_ports {clk}] -clock
[get_clocks {clk1}] 1
```

TA1032

WARN(TA1032) : Option "-from" must be used with "get_clocks"

-from must be followed by get_clocks in set_clock_uncertainty constraint, otherwise it will pop up the above warning.

```
   create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

   set_clock_uncertainty 1 -setup -from [get_ports {clk}] -to [get_clocks
{clk1}]
```

Action

You need to make sure –from is followed by get_clocks in set_clock_uncertainty constraint.

```
   create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

   set_clock_uncertainty 1 -setup -from [get_clocks {clk1}] -to [get_clocks
{clk1}]
```

TA1033

WARN(TA1033) : Option "-to" must be used with "get_clocks"

-to must be followed by get_clocks in set_clock_uncertainty constraint, otherwise it will pop up the above warning.

```
   create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

   set_clock_uncertainty 1 -setup -from [get_clocks {clk1}] -to [get_ports
{clk}]
```

Action

You need to make sure –to is followed by get_clocks in set_clock_uncertainty constraint.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    set_clock_uncertainty 1 -setup -from [get_clocks {clk1}] -to [get_clocks
{clk1}]
```

TA1048

WARN(TA1048) : Existing clock <name> is overwritten

Existing clock is overwritten.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    create_clock -name clk1 -period 20 -waveform {0 5} [get_ports {clk}] –
add
```

Action

You need to make sure there is no repeated clock name in sdc constraint.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    create_clock -name clk2 -period 20 -waveform {0 5} [get_ports {clk}] –
add
```

TA1049

WARN(TA1049) : Object <name> already has one clock applied on it, if you want one more, please use -add option

If there has already one clock applied on it, if you want one more, use –add option.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    create_clock -name clk2 -period 20 -waveform {0 5} [get_ports {clk}]
```

Action

You need to use –add option from the second clock.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    create_clock -name clk2 -period 20 -waveform {0 5} [get_ports {clk}] –
add
```

TA1058

WARN(TA1058) : Input ports list has output ports <name>

There is ouput port in input ports list in set_input_delay constraint. set_input_delay can only be added in input port, otherwise the above warning will pop up..

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    set_input_delay -clock clk1 1 -min -fall [get_ports {out}]
```

Action

set_input_delay can only be added to input port.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    set_input_delay -clock clk1 1 -min -fall [get_ports {in}]
```

TA1059

WARN(TA1059) : Output ports list has input ports <name>

There is input port in output ports list in set_output_delay constraint. set_output_delay can only be added to output port, otherwise the above warning will pop up..

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    set_output_delay -clock clk1 1 -min -fall [get_ports {in}]
```

Action

set_output_delay can only be added to output port.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    set_output_delay -clock clk1 1 -min -fall [get_ports {out}]
```

TA1061

WARN(TA1061) : Cannot find objects matching with <name>

It can not find the specified object in sdc constraint, and the above warning will pop up.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports
{port_no_exist}]
```

Action

You need to modify the object name to make it correct.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]
```

TA1068

WARN(TA1068) :Previous IO timing constraints are overwritten

IO timing constraints are overwritten. You can use -max/-min and -rise/-fall to constrain. If -max/-min are not specified, they are both analyzed, and -rise/-fall is the same. The later will overwrite the previous.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    set_input_delay -clock clk1 1 -max -fall [get_ports {in01}]

    set_input_delay -clock clk1 2 [get_ports {in01}]
```

Action

It is recommended that you had better specify -max/-min and -rise/-fall.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    set_input_delay -clock clk1 1 -max -fall [get_ports {in01}]

    set_input_delay -clock clk1 2 -min -rise [get_ports {in01}]
```

TA1076

WARN(TA1076) : Generated clock Source list has source object <object> which specified by option "-source", this generated clock will be ignored

In create_generated_clock constraint, the object is the master clock source. The object can not be the same as the source, otherwise the above warning will pop up.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    create_generated_clock -name genClk -source [get_ports {clk}]
-master_clock clk1 -divide_by 2 [get_ports {clk}]
```

Action

The object should be the one other than the master clock source.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]
    create_generated_clock -name genClk -source [get_ports {clk}]
-master_clock clk1 -divide_by 2 [get_pins {reg0_0_Z/Q}]
```

TA1109

WARN (TA1109) : Invalid speed grade is specified"

There is an invalid speed grade in sdc. The commercial speed grade of the GW1N series is 5,6; The industrial speed grade is 4,5; The auto speed grade is 4. The commercial speed grade of the GW2A series is 7,8; The industrial speed grade is 6,7; The auto speed grade is 6. This warning will pop up if the specified speed grade does not meet the above requirements.

```
    set_operating_conditions -grade c -model slow -speed 1
```

Action

You need to make sure the device series, temperature grade and speed grade meet the requirements.

```
    set_operating_conditions -grade c -model slow -speed 5
```

TA1114

WARN(TA1114) : Invalid access is specified

There is invalid access in sdc for the following three reasons：

1. The –group is not followed by get_clocks or all_clocks in set_clock_groups constraint.
2. In the constraints of set_false_path /set_max_delay /set_min_delay /set_multicycle_path, -rise_from /-fall_from/ -rise_to/ -rise_fall is not followed by get_clocks or all_clocks.
3. report_timing/ report_exceptions is followed by both -from_clock and -from[get_clocks{}] or both -to_clock and -to[get_clocks{}].

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    create_clock -name clk2 -period 10 -waveform {0 5} [get_ports {clk}] -add

    set_clock_groups -exclusive -group [get_ports {clk}]

    set_false_path -rise_from [get_ports {in1}] -fall_to [get_ports
{out00}]

    report_timing -setup -from_clock [get_clocks {clk1}] -to_clock
[get_clocks {clk1}] -from [get_clocks {clk1}] -to [get_clocks {clk2}]
```

Action

You need avoid the above three conditions.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    create_clock -name clk2 -period 10 -waveform {0 5} [get_ports {clk}] -add

    set_clock_groups -exclusive -group [get_clocks {clk1 clk2}]

    set_false_path –rise_from [get_clocks {clk1}] –fall_to [get_clocks
{clk2}]

    report_timing -setup -from_clock [get_clocks {clk1}] -to_clock
[get_clocks {clk1}]
```

TA2002

ERROR(TA2002) : Cannot get clock with name <name>

If the clock is not correctly created, the above error will be reported.

```
    create_generated_clock -name genClk -source [get_ports {clk}]
-divide_by 2
```

Action

You need to make sure the clock is correctly created.

```
    create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk}]

    create_generated_clock -name genClk -source [get_ports {clk}]
-master_clock clk1 -divide_by 2
```

GOWIN

PROGRAMMING FOR THE FUTURE