Gowin Software User Messages Reference

SUG937-1.1E, 08/09/2024

Disclaimer

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. 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
06/01/2020	1.0E	Initial version published.
08/06/2020	1.0.1E	● Changed EX0311 ERROR to EX0210 WARN. ● EX0206 WARN code updated. ● EX0310 ERROR code updated.
07/14/2023	1.0.2E	TA1125 NOTE added.
11/30/2023	1.0.3E	TA1052 NOTE and TA1083 NOTE added.
12/08/2023	1.0.4E	TA1125 NOTE updated.
08/09/2024	1.1E	● AG0200, CV0003, CV0026, EX0202, EX0309, EX0314, and so on removed. ● CV0013, EX0201, EX1981, EX1998, EX1999, EX2000, EX2452, and so on updated. ● EX0211, EX0213, EX0315-EX0348, EX4557, and so on added.

Contents

Contents i

Gowin Software User Messages Reference 1

Overview 1

GowinSynthesis User Messages 1

AG0100 1

AG0101 3

CK0013 4

CK2060 5

CV0004 7

CV0005 7

CV0008 8

CV0013 8

CV0014 9

CV0015 9

CV0016 10

CV0017 10

CV0018 11

CV0019 12

CV0020 12

CV0021 13

CV0022 13

CV0023 14

CV0024 15

CV0025 15
CV0027 16
CV0028 16
DI0002 17
DI0003 18
DI0012 18
DI0018 18
DI0019 19
DP0002 19
DP0003 20
DP0004 20
DP0008 20
DP0009 20
EX0200 21
EX0201 22
EX0203 23
EX0205 24
EX0206 25
EX0207 26
EX0208 26
EX0209 26
EX0210 27
EX0211 29
EX0212 30
EX0213 30
EX0214 30
EX0300 31
EX0301 31
EX0302 31

EX0308 31
EX0310 32
EX0311 33
EX0312 34
EX0313 35
EX0315 35
EX0316 35
EX0317 36
EX0318 36
EX0319 36
EX0320 36
EX0321 37
EX0322 37
EX0323 37
EX0324 37
EX0325 38
EX0326 38
EX0327 38
EX0328 38
EX0329 39
EX0330 39
EX0331 39
EX0332 40
EX0333 40
EX0334 40
EX0335 41
EX0336 41
EX0337 41
EX0339 41

EX0340 42
EX0341 43
EX0342 44
EX0343 44
EX0344 45
EX0345 46
EX0346 46
EX0347 47
EX0348 47
EX1981 48
EX1998 49
EX1999 49
EX2000 50
EX2452 51
EX2514 52
EX2526 53
EX2565 53
EX2598 55
EX2629 56
EX2635 57
EX2656 57
EX2664 58
EX2666 59
EX2830 60
EX2855 61
EX2932 61
EX2947 62
EX2987 62
EX3041 63

EX3044 63
EX3073 64
EX3320 65
EX3359 66
EX3413 66
EX3483 67
EX3514 67
EX3534 68
EX3589 69
EX3628 69
EX3638 70
EX3670 71
EX3671 73
EX3680 73
EX3682 74
EX3705 75
EX3706 75
EX3735 76
EX3771 77
EX3779 78
EX3780 80
EX3784 81
EX3786 81
EX3791 82
EX3792 83
EX3794 84
EX3812 85
EX3818 86
EX3827 86

EX3829 87
EX3833 88
EX3834 89
EX3858 90
EX3863 90
EX3864 91
EX3872 92
EX3875 92
EX3900 93
EX3902 94
EX3907 95
EX3916 96
EX3927 97
EX3928 97
EX3937 98
EX3945 99
EX3983 100
EX3988 101
EX4557 102
EX4739 104
IF0003 105
IF0005 106
IF0007 106
IF0008 107
IF0009 107
NL0002 107
NL0003 108
NL0004 108
RP0001 109

RP0002 109
RP0006 109
RP0007 110
RP0008 110
RP0009 110
RP0010 111
RP0011 111
RP0012 111
RP0013 111
RP0014 112
RP0015 112
SC0002 112
SC0003 112
SC0004 113
SC0005 113
SC0006 113
SC0007 114
SC0008 114
SC0010 114
SP0001 115
SP0002 115
SP0003 115
SP0005 115
SP0006 116
SP0007 116
SP0008 116
SP0011 116
SP0012 116
SP00017 117

SP00018 117
SP00019 117
SP00020 117
SP0021 117
SP0022 117
SP0024 118
SP0025 118
SP0026 118
ST0001 118
Place & Route User Messages 119
CT1000 119
CT1003 119
CT1005 119
CT1007 120
CT1097 120
CT1098 121
CT1101 121
CT1102 121
CT1108 121
CT1111 122
CT1112 122
CT1113 122
CT1115 123
CT1116 123
CT1117 123
CT1118 124
FS1008 124
FS2001 125
PA1000 125

PA1001 126
PA1002 128
PA1008 128
PA1010 129
PA2000 130
PA2001 131
PA2004 133
PA2009 135
PA2014 137
PA2017 139
PA2024 139
PA2025 139
PA2039 139
PA2054 141
PA2056 142
PA2058 144
PA2066 145
PR0026 147
PR0027 147
PR0028 147
PR0029 148
PR1011 148
PR1014 149
PR2044 150
PR2045 151
PR2061 151
PR2062 151
PR2063 151
PR2064 152

PR2065 152
PR2066 152
PR2067 152
PR2068 153
PR2069 153
PR2070 153
TA1001 153
TA1004 154
TA1006 154
TA1011 154
TA1012 155
TA1016 155
TA1019 156
TA1027 156
TA1032 156
TA1033 157
TA1048 157
TA1049 158
TA1052 158
TA1058 158
TA1059 159
TA1061 159
TA1068 159
TA1076 160
TA1083 160
TA1109 161
TA1114 161
TA1125 162
TA2002 162

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 will list the <signal> and its line information through the above warning. An example of a logical loop is as follows:

Case 1: In the design, the output port out drives itself, resulting in a logical loop.

```
module test (in,out);

input in;

output out;

assign out = in & !out;

endmodule
```

Action

To eliminate the above warning, you need to check and modify the

RTL design to avoid logical loops.

Case 2: In the design, the r-value of the output signal corresponding to the asynchronous control signal is a signal, causing a logical loop.

```
module top(clk,din,din1,rstn,q);
input clk;
input din,din1;
input rstn;
output reg q;

always@(posedge clk or negedge rstn)begin
    if(!rstn)begin
        q<=din1;
    end
    else begin
        q<=din;
    end
end

endmodule
```

Action

To eliminate the above warning, you need to modify the r-value of the output signal corresponding to the asynchronous control signal to be a constant.

Case 3: The sensitive signals of the timing logic exceed two, causing a loop.

```
module top(clk,d,rstn,ce,set,q);
input clk,rstn,ce,set;
input d;
output reg q;
always@(posedge clk or negedge rstn or posedge set)begin
    if(!rstn)begin
```

```
            q<=1;
        end
        else
            if(set)
                q<=0;
            else
                if(ce)
                    q<=d;
end
endmodule
```

Action

To eliminate the above warning, you need to ensure that the sensitive list of the timing logic does not contain more than two signals.

AG0101

WARN (AG0101): The netlist is not one directed acyclic graph

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

To eliminate the above warning, you need to modify the design to avoid the logical loop.

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_inst_top (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_inst(
        .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.

CK2060

ERROR (CK2060): 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 modify the CLKOUT port of DHCEN to drive the correct instance.

```
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
```

CV0004

WARN(CV0004): Implementing a mos primitive as an active <high/low> control tristate

When implementing the mos primitive as an active high/low control tristate, the above warning will be reported.

Action

To eliminate the above warning, do not implement the mos primitive as an active high/low control tristate.

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): Pin(<pin>) of <inst>(<type>) does not connect to port

When the Pin<pin> of the instance <inst> of Buf type <type> is not connected to a port, the above error will be reported. In the following case, the iopin io_w of the IOBUF uut is not connected to a port.

```
module top(a,b,sel,dout,qout);
input a,b;
input sel;
output dout,qout;
wire io_w;


assign dout=io_w&b;
```

```
    IOBUF uut (
        .O(qout),
        .IO(io_w),
        .I(a),
        .OEN(sel)
    );
    endmodule
```

Action

Check the connection between the intermediate Buf and the port.

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.

CV0015

ERROR(CV0015): Port <port> could not connect other instance except via a buf which should connect PAD

When Port <port> is connected to both a Buf and another instance simultaneously, the above error will be reported. In the following case, the output port dout is not connected to the gate through the buf.

```
module top(a,b,sel,dout,qout);
input a,b;
input sel;
```

```
output dout,qout;


assign dout=qout&b;
   TBUF uut (
      .O(qout),
      .I(a),
      .OEN(sel)
);
endmodule
```

Action

Ensure that the port is either connected to other instances through a Buf or disconnected from other instances

CV0016

WARN(CV0016): Input <port> is unused

When an input port <port> is defined but not used, the above warning will be reported. In the following case, the input port sel is defined but not used.

```
module top(a,b,sel,dout);
input a,b;
input sel;
output dout;


assign dout=a&b;


endmodule
```

Action

To eliminate the warning, remove the defined but unused input port.

CV0017

WARN(CV0017): Inout <port> is unused

When an inout port <port> is defined but not used, the above warning will be reported. In the following case, the inout port io is defined but not used.

```
module top(a,b,io,dout);

input a,b;

inout io;

output dout;


assign dout=a&b;


endmodule
```

Action

To eliminate the warning, remove the defined but unused inout port.

CV0018

WARN(CV0018): Input <port><danglingBit> is unused

When a defined input port <port> is a Bus and one of its bits <danglingBit> is not used, the above warning will be reported. In the following case, the second bit a[1] of the input port a is not used.

```
module top(a,dout);

input [2:0]a;

output dout;


assign dout=a[0]&a[2];


endmodule
```

Action

To eliminate the warning, remove the defined but unused bit.

CV0019

WARN(CV0019): Inout <port><danglingBit> is unused

When a defined inout port <port> is a Bus and one of its bits <danglingBit> is not used, the above warning will be reported. In the following case, the second bit io[1] of the inout port io is not used.

```
module top(sel,a,io,dout);
input sel;
input a;
inout [1:0]io;
output dout;

assign dout=io[0];
assign io[0]=sel ? a : 1'bz;
endmodule
```

Action

To eliminate the warning, remove the defined but unused bit.

CV0020

WARN(CV0020): Input <port><highBit:danglingBit> is unused

When the input port <port> is defined as a Bus and it is unused from a bit <danglingBit> to <highBit>, the above warning will be reported. In the following case, the second through fourth bits b[3:1] of input port b are defined but unused.

```
module top(a,b,dout);
input a;
input[3:0]b;
output dout;

assign dout=a & b[0];
endmodule
```

Action

To eliminate the warning, remove the defined but unused bits.

CV0021

WARN(CV0021): Inout <port><highBit:danglingBit> is unused

When the inout port<port> is defined as a Bus and it is unused from a bit <danglingBit> to <highBit>, the above warning will be reported. In the following case, the second through fourth bits io[3:1] of inout port io are defined but unused.

```
module top(a,sel,io,dout);
input a;
input sel;
inout [3:0]io;
output dout;


assign dout=io[0];
assign io[0]=sel ? a : 1'bz;
endmodule
```

Action

To eliminate the warning, remove the defined but unused bits.

CV0022

WARN(CV0022): Port <port> does not exist, attribute constrain 'black_box_pad_pin' has value error

When using the attribute constraint black_box_pad_pin, if the specified port <port> does not exist, the above warning will be reported and the attribute constraint will be ineffective.

```
module top(a, b, a1, b1, dout, dout1);
input [7:0] a, b, a1, b1;
output [15:0] dout, dout1;


assign dout1 = a1*b1;
```

```
sub uut(a, b, dout);
endmodule


module sub (a, b, dout)/* synthesis black_box_pad_pin="D" */;
input [7:0] a, b;
output [15:0] dout;
endmodule
```

Action

To eliminate the warning, specify an existing port.

CV0023

WARN(CV0023): Sweep user defined iobuf instance <inst> with dangling iopin

When the iopin of an instantiated IOBUF <inst> is left floating, the above warning will be reported, and the IOBUF will be removed. In the following case, the iopin of the IOBUF is floating.

```
module top(a, b,dout, o);
input a, b;
output dout, o;


assign dout = a & b;
IOBUF uut(
    .O(o),
    .IO(io),
    .I(a),
    .OEN(b)
);
endmodule
```

Action

To eliminate the warning, either connect the floating iopin or set the syn_dont_touch attribute for the IOBUF.

CV0024

ERROR(CV0024): Multiple GSR are instantiated

When multiple GSR instances are instantiated with different inputs, the above error will be reported. In the following case, two GSR primitives, GSR and GSR1, are instantiated with different inputs.

```
module top(a, b,dout, o);
input a, b;
output dout, o;

assign dout = a & b;
IOBUF uut(
    .O(o),
    .IO(io),
    .I(a),
    .OEN(b)
);

GSR GSR(1'b1);
GSR GSR1(a);
endmodule
```

Action

Check the connection of the GSR.

CV0025

WARN(CV0025): GSR function will be activated

When the inputs of GSR are all GND, the above warning will be reported, indicating that the GSR function will be activated. In the following case, the inputs of GSR are GND.

```
module top(a, b,dout);
input a, b;
```

```
output dout;


assign dout = a & b;


GSR GSR(1'b0);
endmodule
```

Action

To eliminate the warning, check the connection of the GSR.

CV0027

WARN(CV0027): Iopin of instance <inst> should connect inout port, convert output port <port> to <port type>

When the iopin of instance <inst> must be connected to an inout port, the output port <port> will be converted to an inout port <port type> and the above warning will be reported. In the following case, the iopin CK_N of MIPI_DPHY is connected to the output port ck_n, which will be converted to an inout port.

```
module top(ck_n);
output ck_n;
MIPI_DPHY uut(
    .CK_N(ck_n)
);
endmodule
```

Action

To eliminate the warning, check the connection of iopin in instance <inst>.

CV0028

WARN(CV0028): Iopin <pin type> of instance <primitive> <inst> should only connect inout port, could not connect another instance

When the iopin <pin type> of instance <inst> of the primitive <primitive> is connected to other instances in addition to being connected to an inout port, the above warning will be reported.

Action

To eliminate the warning, check the connection of iopin in instance <inst>.

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
```

DI0003

WARN(DI0003): Latch inferred for net <net>;We do not recommend the use of latches in FPGA designs, as they may lead to timing problems

In FPGA design, the use of Latch types is not recommended due to potential timing issues. When <net> is inferred as a Latch, the above warning will be reported. In the following case, the output port o drives itself, resulting in a logical loop and inferring a latch.

```
module test(d,clear,o);

input d,clear;

output o;

assign o=clear ? d :o;

endmodule
```

Action

To eliminate the warning, check the design to avoid inferring latch-type primitives.

DI0012

ERROR(DI0012): Register <reg> can not be converted in synthesis process

Register <reg> conversion error.

DI0018

WARN(DI0018): Optimizing fanout of <node> may lead to resource limit problems, we do not recommend the use of low number fanout in global FPGA designs

Optimizing the fanout of <node> may lead to resource limitations. It is not recommended to use low number of fanout in global FPGA design.

Action

To eliminate the warning, check the syn_maxfan property and set a

reasonable fanout.

DI0019

WARN(DI0019): Merging used defined instance <inst1> to instance <inst2>, because they are equivalent. If you want to keep the instance, please apply property constraint syn_preserve = 1 on it

When instances <inst1> and <inst2> in the design are equivalent and the property constraint syn_preserve=1 is not set, the synthesis will optimize and merge them, resulting in the above warning. In the following case, cnt0 and cnt1 are equivalent and will be merged.

```
module top(clk,dout,qout);
input clk;
output dout,qout;

reg[7:0]cnt0,cnt1;
always@(posedge clk)begin
    cnt0<=cnt0+1;
end
always@(posedge clk)begin
    cnt1<=cnt1+1;
end

assign dout=&cnt0;
assign qout=|cnt1;
endmodule
```

Action

To eliminate the warning, modify the design. If you want to preserve the merged instances, you can set the property constraint syn_preserve=1.

DP0002

ERROR (DP0002): Unsupported target device <dev>

The above error will be reported when the used device <dev> is not supported.

Action

Check the device information and select a device supported by GowinSynthesis for synthesis.

DP0003

ERROR (DP0003): Unsupported package <package> of <dev>

The above error will be reported when the selected device <dev> does not support the package <package>.

Action

Check the package information for the device and select a supported package for synthesis.

DP0004

ERROR (DP0004): Unsupported speed grade <speed> of <dev>

The above error will be reported when the selected device <dev> does not support the speed grade <speed>.

Action

Check the speed grade of the device and select a supported speed grade for synthesis.

DP0008

ERROR (DP0008): Unsupported partNumber <partNum>

The above error will be reported when the selected part number <partNum> is not supported.

Action

Check the selected part number and choose a supported part number for synthesis.

DP0009

ERROR (DP0009): Unsupported partNumber <partNum> of <dev>

The above error will be reported when the part number <partNumber> of selected device <dev> is not supported.

Action

Check the part number of selected device and choose a supported

part number for synthesis.

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

To eliminate the above warning, 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

To eliminate the above warning, 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

To eliminate the above warning, you need to ensure that the <parameter> of the instantiated primitive is set within a reasonable range 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;
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.DEVICE = "GW1N-4B";
    endmodule
```

Action

To eliminate the above warning, ensure that the device information specified is consistent with the device information in the instantiated Gowin primitive <para>.

EX0207

WARN (EX0207): Not support verilog language <Verilog_language>, using default sysverilog-2017

When the specified Verilog version is not supported during synthesis, the above warning will be reported, and the default version, SystemVerilog 2017, will be used for synthesis.

Action

To eliminate the above warning, you need to specify a supported Verilog version.

EX0208

WARN (EX0208): Not support vhdl language <vhdl_language>, using default vhdl_2008

When the specified VHDL version is not supported during synthesis, the above warning will be reported, and the default version VHDL 2008 will be used for synthesis.

Action

To eliminate the above warning, you need to specify a supported VHDL version.

EX0209

ERROR (EX0209): No valid RTL found to combine with <GAO/GVIO> <gao_file>

When configuring GAO and GVIO, if there are no valid RTL files available for connecting GAO and GVIO, the above error will be reported.

Action

Create valid design files or use existing netlist files to configure GAO and GVIO.

EX0210

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

If the synthesis tool reports the above warning, it indicates that the <freq> of the instantiated Gowin primitive device <inst> is outside the valid range of <num1> to <num2>, and is greater than <num2>, the maximum value. In this case, the warning will be reported, and the corresponding <freq> will be set to the default value. Modify the relevant design based on the reported <inst> name and line information. If the <freq> of the instantiated Gowin primitive <inst> is not set, it will be checked against the default <freq> value.

```
module PLLO_ins (CLKIN, CLKFB, RESET, RESET_P, RESET_I, RESET_S,
FBDSEL, IDSEL, ODSELA, ODSELB, ODSELC, ODSELD, DTA, DTB, ICPSEL, LPFRES,
PSSEL, PSDIR, PSPULSE, ENCLKA, ENCLKB, ENCLKC, ENCLKD, LOCK, CLKOUTA,
CLKOUTB, CLKOUTC, CLKOUTD);

input CLKIN, CLKFB;

input RESET, RESET_P, RESET_I, RESET_S;

input  [5:0] FBDSEL, IDSEL;

input  [6:0] ODSELA, ODSELB, ODSELC, ODSELD;

input  [3:0] DTA, DTB;

input  [4:0] ICPSEL;

input  [2:0] LPFRES;

input  [1:0] PSSEL;

input PSDIR;

input PSPULSE;

input ENCLKA, ENCLKB, ENCLKC, ENCLKD;

output LOCK;

output CLKOUTA, CLKOUTB, CLKOUTC, CLKOUTD;
```

```
PLLO pllo_check(
    .CLKIN(CLKIN),
    .CLKFB(CLKFB),
    .RESET(RESET),
    .RESET_P(RESET_P),
    .RESET_I(RESET_I),
    .RESET_S(RESET_S),
    .FBDSEL(FBDSEL[5:0]),
    .IDSEL(IDSEL[5:0]),
    .ODSELA(ODSELA[6:0]),
    .ODSELB(ODSELB[6:0]),
    .ODSELC(ODSELC[6:0]),
    .ODSELD(ODSELD[6:0]),
    .DTA(DTA[3:0]),
    .DTB(DTB[3:0]),
    .ICPSEL(ICPSEL[4:0]),
    .LPFRES(LPFRES[2:0]),
    .PSSEL(PSSEL[1:0]),
    .PSDIR(PSDIR),
    .PSPULSE(PSPULSE),
    .ENCLKA(ENCLKA),
    .ENCLKB(ENCLKB),
    .ENCLKC(ENCLKC),
    .ENCLKD(ENCLKD),
    .LOCK(LOCK),
    .CLKOUTA(CLKOUTA),
    .CLKOUTB(CLKOUTB),
    .CLKOUTC(CLKOUTC),
    .CLKOUTD(CLKOUTD)
);
```

```
defparam pllo_check.FCLKIN = "100.0";
defparam pllo_check.FBDIV_SEL = 9;
defparam pllo_check.ODIVA_SEL = 120;
defparam pllo_check.IDIV_SEL = 60;


endmodule
```

Action

To eliminate the above error, you need to refer to *SUG283, Gowin Primitives User Guide* to adjust the <freq> value, or modify the specified device information.

EX0211

WARN (EX0211): The output port <port> of module <module> has no driver, assigning undriven bits to Z, simulation mismatch possible

If the output <port> is left floating, it will be connected to GND or high impedance. The simulation behavior of the synthesis results may differ from the RTL. The synthesis tool will issue the above warning and continue with synthesis.

```
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 it a value of GND or VCC.

```
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
```

EX0212

ERROR (EX0212): No valid RTL found to combine with I2C <i2cPath>

When configuring I2C, if there is no valid RTL, the above error will be reported.

Action

Create valid design files or use existing netlist files to configure the I2C.

EX0213

ERROR (EX0213): No valid RTL found to combine with active flash <flashPath>

When configuring active flash, if there is no valid RTL, the above error will be reported.

Action

Create valid design files or use existing netlist files to configure the active flash.

EX0214

WARN (EX0214): Instance <inst> 's parameter <para> value must be an <parity> number from <num1> to <num2>, replaced by default value <para>

When setting the parameter value of an instance, if a parity (odd/even) is specified, and the input parameter value is within the range of <num1> to <num2>, but the parity does not match the specification, the above error will be reported and the parameter value will be set to the default value. In the following case, because the FREQ_DIV parameter of the OSC primitive must be an even number between 2 and 128, setting it to an odd number such as 11 in the instantiation will result in this error.

```
module Gowin_OSC (oscout);

output oscout;

OSC osc_inst (
```

```
        .OSCOUT(oscout)

    );


    defparam osc_inst.FREQ_DIV = 11;

    defparam osc_inst.DEVICE = "GW1N-4D";


    endmodule //Gowin_OSC
```

Action

To eliminate the above warning, refer to the error message and correctly configure the parameter value.

EX0300

ERROR (EX0300): Not support node type : <type>

When performing type conversion, if there is an unsupported primitive type in the user design, the above error will be reported.

EX0301

ERROR (EX0301): Do not support asynchronous write memory operation, signal is <signal>

When performing type conversion, if there is an instance of asynchronous write memory, the above error will be reported.

EX0302

ERROR (EX0302) : No valid top module found

If there are no modules in the design file and the design file is empty, the synthesis tool will report the above error. Synthesizing an empty design file is not allowed.

EX0308

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

The synthesis tool will report the above error if it cannot find the synthesis configuration files primitive.xml or prim_syns in the installation directory IDE\bin. Try moving primitive.xml or prim_syns back to their

original location, or reinstall the software.

EX0310

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

If the <paraValue> of <inst> <para> in Gowin Primitive is invalid, the synthesis tool will report the above error. 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;
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 rpll.FCLKIN = "600.0";
endmodule
```

Action

To eliminate the above warning , you need to make the <paraValue> valid by referring to the reported error information and *SUG283, Gowin Primitives User Guide*.

EX0311

ERROR (EX0311): Invalid <type> frequency <freq> to instance <inst>, suitable range is from <num1>MHz to <num2>MHz

If the synthesis tool reports the above warning, it indicates that the <freq> of the instantiated Gowin primitive device <inst> is outside the valid range of <num1> to <num2>, and is less than <num1>, the minimum value. At this point, the above error will be reported. Modify the relevant design based on the reported <inst> name and line information. If the <freq> of the instantiated Gowin primitive <inst> is not set, it will be checked against the default <freq> value. In the following case, the VCO frequency (FCLKIN*(FBDIV_SEL+1*ODIV_SEL)/(IDIV_SEL+1)) of the rpll primitive is 80MHz, which is not within the specified range of 500MHz to 1250MHz, so this error will be reported.

```
module Gowin_rPLL (clkout, clkin);


output clkout;

input clkin;


wire lock_o;

wire clkoutp_o;

wire clkoutd_o;

wire clkoutd3_o;

wire gw_gnd;


assign gw_gnd = 1'b0;


rPLL rpll_inst (

    .CLKOUT(clkout),

    .LOCK(lock_o),
```

```
        .CLKOUTP(clkoutp_o),
        .CLKOUTD(clkoutd_o),
        .CLKOUTD3(clkoutd3_o),
        .RESET(gw_gnd),
        .RESET_P(gw_gnd),
        .CLKIN(clkin),
        .CLKFB(gw_gnd),
        .FBDSEL({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
        .IDSEL({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
        .ODSEL({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
        .PSDA({gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
        .DUTYDA({gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
        .FDLY({gw_gnd,gw_gnd,gw_gnd,gw_gnd})
    );

    defparam rpll_inst.FCLKIN = "50";
    defparam rpll_inst.IDIV_SEL = 4;
    defparam rpll_inst.FBDIV_SEL = 0;
    defparam rpll_inst.ODIV_SEL = 8;
    endmodule //Gowin_rPLL
```

Action

To eliminate the above error, refer to the reported error information and *SUG283, Gowin Primitives User Guide* to adjust the <freq> value or modify the specified device information.

EX0312

ERROR (EX0312): There is no <node_type> resource in current device, please change device

The above error will be reported when a primitive of type <node_type> is used, but the current device does not have <node_type> resources, such as for PLL types.

Action

Modify the device to one that supports the primitive or use a primitive that is supported by the current device.

EX0313

ERROR (EX0313): <GAO/GVIO> port <port> already defined in top module/unit

The above error will be reported when a port defined in a module/unit has the same name as a GAO or GVIO port.

Action

Modify the port name to ensure it does not conflict with GAO or GVIO ports.

EX0315

ERROR (EX0315): Instantiated primitive <inst> does not match with the definition of <prim>

When instantiating a primitive, if the instance <inst> does not match the definition of the corresponding primitive <prim> and an unknown pin type is encountered, the above error will be reported.

Action

Instantiate the primitive according to the format specified for the corresponding primitive.

EX0316

ERROR (EX0316): Invalid parameterized value <value>(<para1>) specified for instance <inst>, <para1> value need less than or equal to <para2>

The value <value> set for parameter <para1> in instance <inst> is not valid. This value needs to be less than or equal to the value of parameter <para2>.

Action

Refer to the error message and *SUG283, Gowin Primitives User Guide* to set a valid parameter value that meets the requirements, or modify the device.

EX0317

ERROR (EX0317): Can not find module named <GAO/GVIO> from design

When specifying the file for configuring GAO/GVIO, if the file format is incorrect, the GAO/GVIO module cannot be retrieved from the design, and the above error will be reported.

Action

Correctly configure the GAO/GVIO file and specify it during runtime.

EX0318

ERROR (EX0318): Gao module <GAO/GVIO> already be referenced, not one separate module

The above error will be reported if the GAO/GVIO module has already been specified and is not a standalone module.

Action

Use the GAO/GVIO module as a standalone module.

EX0319

ERROR (EX0319): Path of module <GAO/GVIO> is <path>, which is not corresponding with <-gao/-gvio> option

The above error will be reported if the path <path> of the GAO/GVIO module does not match the path specified in the -gao/-gvio option.

Action

Check and modify the file paths to ensure they match.

EX0320

ERROR (EX0320): Can not find <GAO/GVIO> specified net <net> in netlist <netlist>

The above error will be reported if the net specified by GAO/GVIO cannot be found in the corresponding netlist.

Action

Check the netlist and GAO/GVIO configuration, and specify a net that exists in the netlist.

EX0321

ERROR (EX0321): Can not find <GAO/GVIO> specified instance <inst> in netlist <netlist>

The above error will be reported if the instance specified by GAO/GVIO cannot be found in the corresponding netlist.

Action

Check the netlist and GAO/GVIO configuration, and specify an instance that exists in the netlist.

EX0322

ERROR (EX0322): PLLO(<inst>) parameter setting < CLKB/C/D_IN_SEL=value> conflict with 'CLKA_IN_SEL=2'b11', DIVB/C/D source clock must from CLKCAS_A/B/C or CLKIN

When the parameter CLKA_IN_SEL of the PLLO is set to 2'b11, the value of parameter CLKB/C/D_IN_SEL conflict with it. The source clock for DIVB/C/D must come from CLKCAS_A/B/C or CLKIN.

Action

Check the parameter setting of the PLLO and modify the corresponding parameter value according to the error message.

EX0323

ERROR (EX0323): I2C port <port> already defined in top module/unit

The above error will be reported if the port <port> of the I2C is already defined in the top module/unit.

Action

Modify the port name in the top module/unit to avoid the conflict with the I2C port name.

EX0324

ERROR (EX0324): Can not find module named "GW_BACKGROUND_I2C_TO_JTAG" from design

The above error will be reported if the I2C module cannot be found in the design.

Action

Check the I2C configuration.

EX0325

ERROR (EX0325): I2C module "GW_BACKGROUND_I2C_TO_JTAG" already be referenced, not one separate module

The above error will be reported if the I2C module has already been specified and is not a standalone module.

Action

Use the I2C module as a standalone module.

EX0326

ERROR (EX0326): Path of module "GW_BACKGROUND_I2C_TO_JTAG" is <path>, which is not corresponding with -i2c_bgp option

The above error will be reported if the path <path> of the I2C module does not match the path specified in the -i2c_bgp option.

Action

Check and modify the file paths to ensure that they are equal.

EX0327

ERROR (EX0327): Invalid port <port> found in I2C module "GW_BACKGROUND_I2C_TO_JTAG"

The above error will be reported if there is an illegal port <port> in the I2C module.

Action

Check and modify the port names in the I2C module to ensure they conform to the specification.

EX0328

ERROR (EX0328): Invalid direction for port <port>, the direction should be <dir>

The above error will be reported if the direction of a specific port <port> is invalid.

Action

Modify the direction of the port according to the error message to ensure it conforms to the specification.

EX0329

ERROR (EX0329): Illegal recursive module instantiation <modu>

The above error will be reported when illegal recursive instantiation of submodules occurs.

Action

Check the definition and usage of submodules in the design.

EX0330

ERROR (EX0330): Invalid parameter value <para:value>(<inst>), <para> can set <rightValue>, when <expression>

When the logical expression <expression> is satisfied, the legal value of the parameter <para> is <rightValue>. If the parameter of the instance <inst> and its corresponding value <para:value> are not legal, the above error will be reported.

Action

Set the parameter value according to the valid value <rightValue> provided in the error message.

EX0331

ERROR (EX0331): Memory <value> size is too large to synthesize

In the context of synthesizing Memory, if the memory size <value> exceeds the synthesis memory limit of 2520K, the above error will be reported. In the following case, the memory defined by mem exceeds the limit 2520K.

```
module test(clk,addr,din,dout);
input [23:0]addr;
input [15:0]din;
input clk;
output reg[15:0]dout;
reg[15:0]mem[2**24-1:0];
```

```
always@(posedge clk)begin
    mem[addr]<=din;
    dout<=mem[addr];
end
endmodule
```

Action

Reduce the memory size within the synthesis limit.

EX0332

ERROR (EX0332): Active flash port <port> already defined in top module/unit

The above error will be reported if the port <port> of the active flash is already defined in the top module/unit while using active flash mode.

Action

Modify the port name in the top module/unit to avoid conflict with the active flash port name.

EX0333

ERROR (EX0333): Can not find module named "activeFlash" from design

The above error will be reported if the active flash module cannot be found in the design while using active flash.

Action

Check the active flash configuration.

EX0334

ERROR (EX0334): Active flash module \"activeFlash\" already be referenced, not one separate module

The above error will be reported if the active flash module has already been specified and is not a standalone module.

Action

Use the active flash as a standalone module.

EX0335

ERROR (EX0335): Path of module "activeFlash" is <path>, which is not corresponding with -active_flash option

The above error will be reported if the path <path> of the active flash module does not match the path specified in the -active_flash option.

Action

Check and modify the file paths to ensure they are equal.

EX0336

ERROR (EX0336): Invalid port <port> found in active flash module "activeFlash"

The above error will be reported if there is an invalid port <port> in the active flash module.

Action

Check and modify the port names in the active flash module to ensure they conform to the specification.

EX0337

ERROR (EX0337): Constraintend to multiSrc net <net>

The above error will be reported if the net <net> has multiple levels of driving.

Action

Check the connection between nets.

EX0339

ERROR (EX0339): Port <port> drives 1 pad loads(<inst1>) and 1 non-pad loads(pin:<pin> inst:<inst2>)"

The above error will be reported if a port <port> drives both a buffer and a non-buffer. In the following case, the port io drives iopin of IOBUF uut (buffer) and dout (non-buffer).

```
module test(i,oen,din,io,o,dout);

input i,oen;

input din;
```

```
inout io;
output o,dout;
assign dout=io&din;
IOBUF uut(
    .O(o),
    .IO(io),
    .I(i),
    .OEN(oen)
);
endmodule
```

Action

Check the connection of the ports.

EX0340

WARN (EX0340): Invalid attribute constrain location, please use 'full_case' directive after case, casex or casez

The above warning will be reported if the full_case constraint is placed incorrectly. In the following case, the full_case constraint is written at the module definition, where it cannot take effect.

```
module my_mux (a,b,c,sel,out)/*synthesis full_case*/;
input[2:0]       a, b, c;
input[1:0]sel;
output reg    [2:0] out;
always @ (a, b, c, sel) begin
    case(sel)
        2'b00       : out = a;
        2'b01       : out = b;
        2'b10       : out = c;
        default    : out = 0;
    endcase
end
```

```
endmodule
```

Action

Based on the warning message, use the full_case constraint directly after case, casex, or casez.

EX0341

WARN (EX0341): Instance <inst> 's parameter <para> value must be <value> when partNumber is <dev>, replaced by default value <value>

If the value of parameter <para> of instance <inst> is set incorrectly when the selected device is <dev>, the above warning will be reported and the value of the set parameter will be replaced with the default value <value>. In the following case, the value of parameter S_RATE in the OSCZ primitive must be "SLOW" or "FAST" depending on the device selected, and the parameter S_RATE in the instance osc_inst is set incorrectly.

```
module Gowin_OSC (oscout, oscen);


output oscout;
input oscen;


OSCZ osc_inst (
    .OSCOUT(oscout),
    .OSCEN(oscen)
);
defparam osc_inst.FREQ_DIV = 100;
defparam osc_inst.S_RATE = "SLOWA";


endmodule //Gowin_OSC
```

Action

To eliminate the warning message, set the value of the parameter to the default value <value> or leave it unset.

EX0342

WARN (EX0342): The port <port> connected to <inst>(<type>) defined error direction which should be 'INOUT' according to connection

The above warning will be reported if an input port <port> is connected to the iopin of an IOBUF. In the following case, the input port b is connected to the iopin of IOBUF uut.

```
module top (a,b,sel,dout);
input a, b;
input sel;
output dout;

IOBUF uut(
    .O(dout),
    .IO(b),
    .I(a),
    .OEN(sel)
);

endmodule
```

Action

To eliminate the warning, do not connect input ports to the iopin of an IOBUF.

EX0343

ERROR (EX0343): Duplicate module name <modu1> and entity name <modu2>

The above error will be reported if the top-level module name <modu1> in Verilog and the top-level module name <modu2> in VHDL are the same when performing mixed compilation.

Action

Modify the top-level module names so that they are not the same.

EX0344

ERROR (EX0344): Net <net> has multiple drivers, drived by <node>

The above error will be reported if a net has multiple drivers. 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

To eliminate the above error, modify the design so that each signal has only one driver source, 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
```

EX0345

ERROR (EX0345): Instance <inst> 's parameter <para> value invalid

The above error will be reported if the parameter <para> value of instance <inst> is invalid. In the following case of an ALU parameter check, the ALU_MODE parameter value in the ALU primitive instance uut is invalid.

```
module top (a,b,dout);
input a, b;
output[1:0] dout;

  ALU uut (
    .SUM(dout[1]),
    .COUT(dout[0]),
    .I0(a),
    .I1(b),
    .I3(1'b0),
    .CIN(1'b0)
);
defparam uut.ALU_MODE=10;
endmodule
```

Action

Modify the parameter <para> value according to the error message to ensure it is valid.

EX0346

WARN(EX0346): Instance <inst>'s parameter value <oldValue>

truncated to <NewValue> to fit <digit> bits

The above warning will be reported when the parameter value <oldValue> of instance <inst> is truncated to <NewValue> due to a bit-width mismatch with the required <digit> bits. In the case of the LUT2 primitive instance uut, the INIT parameter is set to 8'b11001110. Since the INIT parameter for LUT2 is defined to be 4 bits, the value will be truncated to 4'b1110, resulting in this warning.

```
module test(dout,a,b);
input a,b;
output dout;
LUT2 uut(
    .F(dout),
    .I0(a),
    .I1(b)
);
defparam uut.INIT = 8'b11001110;
endmodule
```

Action

To eliminate the warning, set the parameter value according to the bit-width defined for the parameter.

EX0347

WARN (EX0347): Ignoring <grammar>

When the design contains non-synthesizable grammar <grammar>, it will be ignored, and the above warning will be reported.

Action

To eliminate the warning, avoid using non-synthesizable grammar in the design.

EX0348

ERROR (EX0348): Port <port> is declared repeatedly

When there are duplicate declarations of a port <port> in the design, the above error will be reported. In the following case, if the port mout is declared multiple times, the error will be reported and the process will

terminate.

```
module test(clk,a,b,mout);
input clk;
input signed [3:0] a;
input signed [3:0] b;
output reg signed [3:0] mout;
always@ (posedge clk)begin
    reg mout = a*b;
end
endmodule
```

Action

Based on the error message, remove the duplicate port declarations.

EX1981

ERROR (EX1981): Net <objec> is driven by multiple input ports

If there are multiple assignments to the same variable <object> from input port in the design, the synthesis tool will issue the above warning. In the following case, the variable tmp is 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

To eliminate the above error, you need to remove the duplicate assignments, 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
```

EX1998

WARN (EX1998): Net <object> does not have a driver

When a wire or register variable <object> is defined but not driven in the design, the synthesis tool will issue the above warning. In the following case, the wire a is defined and used but is not driven.

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

Action

To eliminate the above warning, you should either remove the wire a or add the appropriate connections to it, as shown below.

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

EX1999

ERROR (EX1999): Another driver from here

When multiple inputs simultaneously drive the same output in the design, the synthesis tool will report the above error, specifying the locations of the duplicate assignments. In the following case, the out port is driven multiple times and you need to remove one of the assignments.

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

Action

To eliminate the above error, you need to remove the duplicate assignment, as shown below.

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

EX2000

ERROR (EX2000): Net <object> is constantly driven from multiple places

When multiple inputs simultaneously drive the same output <object> in the design, the synthesis tool will report the above error, indicating the locations of the duplicate assignments. In the case where the output port out is driven multiple times, you will need to remove one of the assignments.

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

Action

To eliminate the error, you need to remove the duplicate assignments, as shown below.

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

EX2452

ERROR (EX2452): Invert of if-condition matches sensitivity list edge, this is unconventional

If the reset signal in the always block sensitivity list is posedge/negedge, but the reset signal in the always block statement is enabled by low/high level, the above error will be reported. In the following case, the reset signal in the always block sensitivity list is posedge, but the reset signal in the always block statement is enabled by a low level, so it is necessary to change the posedge to negedge or enable it by a high level.

```
module test (clk,rst,d,q);
input clk,rst,d;
output reg q;
always@(posedge clk or posedge rst)begin
    if(!rst) begin
        q<=0;
    end
    else begin
        q<=d;
    end
end
endmodule
```

Action

To eliminate the error, ensure that the reset signal in the always block sensitivity list corresponds correctly with its levels; posedge means active high, and negedge means active low.

EX2514

ERROR (EX2514):Task call from within a function is not allowed

When a task is found within a function in the design, the following error will be reported. In the case where the function sub contains the task add, it will result in a syntax error.

```
module test(a,b,c,dout);
input [3:0] a,b,c;
output [4:0] dout;
task add;
input [3:0] a,b;
output [4:0] dout;
    begin
        dout = a + b;
    end
endtask

function [4:0] sub;
input [3:0] a,b,c;
reg [4:0] tmp;
    begin
        add(a,b,tmp);
        sub = tmp - c;
    end
endfunction
assign dout = sub(a,b,c);
endmodule
```

Action

Do not use task grammar within a function.

EX2526

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

If the design uses the $readmemh statement but the data width <width> in the corresponding file does not match the expected memory width <memWidth>, the synthesis tool will report the above warning. In the following case, if the memory width is 8 and the data width <initWidth> in the file does not match, this warning will be reported.

```
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

To eliminate the above warning, ensure that the data file specified in the $readmemh() statement and the array width are compatible.

EX2565

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

When a port <port> is defined in the design but not used, the synthesis tool will issue the above warning. In the following case, in the test module, the clk port is defined but not used.

```
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

To eliminate the above warning, you need to remove the unused 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
```

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

To eliminate the above warning, 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

To eliminate the above warning, 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 design uses a generate statement to create modules in a loop, as shown in the flowing example where the test module is generated within a for loop, this usage is only supported in SystemVerilog. Using other versions of Verilog will result in the above warning reported from the synthesis tool.

Action

The above warning is normal for the use of this syntax; note that this usage is only supported in SystemVerilog, to get rid of this warning, use SystemVerilog to synthesize.

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

To eliminate the above error, 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

To eliminate the above warning, 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
```

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

To eliminate the above warning, 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

To eliminate the above warning, 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

To eliminate the above warning, 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

To eliminate the above, 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 <port> of a module is not connected to the corresponding signal during instantiation, the synthesis tool will report the above warning.

Action

To eliminate this warning, you can connect the floating port.

EX2987

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

If the port <port> of a module is not connected to the corresponding signal during instantiation, the synthesis tool will report the above warning.

Action

To eliminate this warning, you need to remove the unused floating port.

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

To eliminate the above warning, 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

To eliminate the above warning, 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

To eliminate the above warning, 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
```

EX3320

ERROR (EX3320): Multiple packed dimensions are not allowed in this mode of Verilog

When using Verilog 2001, if there is a high-dimensional reg definition, the synthesis tool will report the above error. In the following case, mem is defined as reg[255:0][3:0] mem, Verilog 2001 does not support this type of definition.

```
module top (clk,din,dout);
input clk;
input [3:0] din;
input [3:0] dout;
reg[255:0][3:0]mem;
endmodule
```

Action

To eliminate the above error, you can either modify the Verilog version or the design.

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

To eliminate the above warning, 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

To eliminate the above warning, 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

To eliminate the above error, you need to remove either the rising edge trigger or the falling edge trigger of the clk signal. As shown below, only keep the rising edge trigger.

```
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

To eliminate the above warning, you can define an intermediate register and assign the value to the 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

To eliminate the above warning, you can either place the wire declaration before use, or simply adopt the implicit declaration and remove the explicit one.

```
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

To eliminate the above warning, 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;
wire out;
wire out;
assign out = !in;
endmodule
```

Action

To eliminate the above warning, you need to remove the duplicate definition as shown below.

```
module top (in,out);
input in;
output out;
wire out;
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

To eliminate the above warning, 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

To eliminate the above warning, 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

To eliminate the above warning, 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

To eliminate the above error, you need to remove one of the duplicate port assignments 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

To eliminate the above warning, 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. Signal missing is added and assuming complete sensitivity list specified. RTL design and post-synthesis netlist simulations may differ as a result

If the design uses an always statement but <signal> is missing from sensitivity list, the synthesis tool will report the above warning. In the following case, in0 and in1 are sensitivity signals and should be included in the always sensitivity list; otherwise they will be added automatically during

synthesis.

```
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

To eliminate the above warning, 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

To eliminate the above warning, 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

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

If the design uses a <port> with a <width> that exceeds the defined width range <range>, the synthesis tool will report the above warning. In the following case, if the defined width range for out is from 0 to 1, but out[2] is assigned a value, the synthesis tool will report this warning during synthesis.

```
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

To eliminate the above warning, you need to adjust the width of out or remove the using of undefined widths, 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

WARN (EX3786): Assignment to input <port>

If a value is assigned to the input <port> in the design, the synthesis tool will report the above warning. 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

To eliminate the above warning, 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

To eliminate the above warning, 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

To eliminate the above warning, 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): Duplicate <modu> name <name>.

If the design defines the module with the same name <name>, the synthesis tool will report this error. You need to modify the name of the modules <modu> to ensure the names are different. In the following case, if two modules are both named test, this error will occur.

```
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

To eliminate the above error, you need to modify the name of one of the modules to ensure the names are different, 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
```

EX3812

ERROR (EX3812): <signal> is not a constant.

In Verilog design, if the condition of an if statement contains a non-constant variable <signal>, this error will be reported. In the following case, the condition in the if statement in line seven is an input port and needs to be modified to a constant.

```
module top(clk,ADDSUB,a,b,c,mout);
input clk;
input [7:0]a,b,c;
input ADDSUB;

output reg [15:0]mout;
generate
  if(ADDSUB)begin
    always@(posedge clk)begin
      mout<=a*b+a*c;
    end
  end
  else begin
    always@(posedge clk)begin
      mout<=a*b-a*c;
    end
  end
endgenerate
endmodule
```

Action

To eliminate the above error, modify the condition in the if statement to ensure that it does not include non-constant variables.

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 directive is used in the design, 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

This warning is a prompt to add full_case statement, which will reduce the irrelevant conditional logic circuit of the case statement. To eliminate the above warning, you need to complement the 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

To eliminate the above error, 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

To eliminate the above error, 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

To eliminate the above warning, 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>

In the Verilog design, 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

To eliminate the above error, 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

To eliminate the above warning, 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

To eliminate the above error, you need to either delete the undeclared ports or add the port declarations, as shown below.

```
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

To eliminate the above error, 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 <net> is not permitted

If the synthesis tool will report the above error, it indicates that a non-register <net> is being assigned using a non-blocking assignment. In the following case, the declaration of out should be of a register.

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

Action

To eliminate the above error, you need to modify the declaration of out to a register, 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

To eliminate the above error, 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

To eliminate the above error, 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

To eliminate the above warning, 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 Content
/*
parameter size = 2;
*/
```

Action

To eliminate the above error, 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 Content
/*
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 statement 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

To eliminate the above warning, 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

To eliminate the above warning, you need to check the file or get the permission.

EX4557

WARN (EX4557): Actual for formal port 'a_in' is neither a static name nor a globally static expression

In VHDL 1995, calling functions in port mapping (port map) is not allowed, except for type conversions. Otherwise the above warning will be reported.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity top is
      port(
            clk: in std_logic;
            oce: in std_logic;
            ce: in std_logic;
            reset: in std_logic;
            wre: in std_logic;
            a,b:in std_logic_vector(31 downto 0);
            ad: in std_logic_vector(13 downto 0);
```

```
        qout:out std_logic_vector(31 downto 0)
    );
end;

architecture rtl of top is
component SP
        generic (
            READ_MODE: in bit := '0';
            WRITE_MODE: in bit_vector := "00";
            BIT_WIDTH: in integer := 32;
            BLK_SEL: in bit_vector := "000";
            RESET_MODE: in string := "SYNC"
        );
        port (
            DO: out std_logic_vector(31 downto 0);
            CLK: in std_logic;
            OCE: in std_logic;
            CE: in std_logic;
            RESET: in std_logic;
            WRE: in std_logic;
            BLKSEL: in std_logic_vector(2 downto 0);
            AD: in std_logic_vector(13 downto 0);
            DI: in std_logic_vector(31 downto 0)
        );
    end component;

        function add(a,b: std_logic_vector(31 downto 0)) return    std_logic_vector is
        begin
            return a+b;
        end function add;
```

```
begin
        sp_inst_0: SP
        generic map (
            READ_MODE => '0',
            WRITE_MODE => "00",
            BIT_WIDTH => 4,
            RESET_MODE => "SYNC",
            BLK_SEL => "000"
        )
        port map (
            DO => qout,
            CLK => clk,
            OCE => oce,
            CE => ce,
            RESET => reset,
            WRE => wre,
            BLKSEL => "000",
            AD => ad,
            DI => add(a,b)
        );
end;
```

Action

To eliminate the above warning, you can update the version to VHDL 2008 or VHDL 2019.

EX4739

ERROR(EX4739): Syntax error near <object >

In VHDL design, if there are syntax errors around the <object>, the synthesis tool will report this error. In the following entity definition, a semicolon is needed at the end of the tenth line.

```
entity Gowin_MULT is
```

```
port (
dout: out std_logic_vector(35 downto 0):
a: in std_logic_vector(17 downto 0);
b: in std_logic_vector (17 downto 0);
ce: in std_logic;
clk: in std_logic;
reset: in std_logic
);
end entity
```

Action

To eliminate the above error, check and correct the syntax errors at the indicated positions based on the error message.

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.

IF0005

WARN (IF0005): Not support distributed <rom/ram> in current device, please change <ramstyle/romstyle> property setting

In RAM/ROM inference, when specifying the inference as distributed RAM/ROM through property constraints, if the selected device does not have distributed RAM/ROM resources, the warning will be reported, and the RAM/ROM will be synthesized using logic resources.

Action

To eliminate this warning, modify the ramstyle/romstyle properties.

IF0007

WARN (IF0007): Attribute value of 'syn_ramstyle/syn_romstyle' is not applicable

In RAM inference, when the specified syn_ramstyle/syn_romstyle settings are not appropriate, a warning will be issued, and the RAM will be synthesized using logic resources.

Action

To eliminate the above warning, modify the syn_ramstyle/syn_srlstyle attributes.

IF0008

ERROR (IF0008): The number(<needNum>) of <logic> used to infer <ram> exceeds the resource limit(<availableNum>) of current devic(<partNum>)

In RAM inference, when the property constraints are set to logic or when the RAM is split due to insufficient RAM resources, if the number of logic resources <logic> required to split the RAM <ram> exceeds the available logic resource limit <availableNum> of the device <partNum>, the above error will be reported.

Action

To eliminate the above error, modify the property constraints syn_ramstyle/syn_romstyle, reduce the RAM capacity, or select a device with more available logic resources.

IF0009

WARN (IF0009): The number used to infer <type> exceeds the resource limit of current device, please change device or <property> setting

In ROM inference, when the number of inferred ROMs of type <type> exceeds the current device's limit, the above warning will be reported.

Action

Modify the selected device or the property constraint <property>.

NL0002

WARN (NL0002): The module <module> instantiated to <moduleName> is swept in optimizing

If a module is optimized in the synthesis, a warning will be reported. In the following case, the output of the instance uut of the module sub is left floating and gets optimized during synthesis.

```
module test(a,b,c,d,dout);

input a,b,c,d;

output dout;

wire dout_w;

sub uut(
```

```
        .a(a),
        .b(b),
        .dout(dout_w)
    );
    assign dout=dout_w & c ^d;
    endmodule

    module sub(a,b,dout);
    input a,b;
    output dout;
    assign dout=a&b;
    endmodule
```

Action

To eliminate the above warning, modify the module definition.

NL0003

WARN (NL0003): Probe name <probe> already in use, replaced by <probe>

When synthesizing, if the generated probe port name is the same as the original port name after the syn_probe is constrained, the above warning will be reported and the probe port name will be modified to append an underscore + duplication count value.

Action

To eliminate the above warning, modify the syn_probe constraint object name so that it is not the same as the port name, or modify the port name accordingly.

NL0004

WARN (NL0004): Iopin <pin> of instance <inst> should connect inout port

When the iopin <pin> of instance <inst> is not directly connected to an inout port, the above warning will be reported.

Action

To eliminate the obove warning, modify the design to ensure that the pin of instance is connected to an inout port.

RP0001

ERROR (RP0001): The number(<needNum>) of <primitive> in the design exceeds the resource limit(<availableNum>) of current device(<partNum>)

When the number of primitives <primitive> in the design exceeds the device <partNum>'s allowable limit of primitives <availableNum>, the above error will be reported.

Action

Choose a device that can meet the required number of primitives or modify the design.

RP0002

ERROR (RP0002): The number(<needNum>) of <primitive> in the design exceeds the resource limit(<availableNum>) of current device. And <property> maybe the useful user assignment to change the inference result

When performing RAM or DSP inference, if the number of primitives <primitive> in the design <needNum> exceeds the resource limit of the current device <availableNum>, the above error will be reported.

Action

According to the error message, you can modify the ramstyle/dspstyle property <property> to alter the synthesis result.

RP0006

ERROR (RP0006): The number(<logicUsage>(<lutUsage> LUTs, <aluUsage> ALUs, <rom16Usage> ROM16s, <ssramUsage> SSRAMs)) of logic in the design exceeds the resource limit(<logicAvailable>) of current device

When the number of logic resources <logicUsage>, including LUTs <lutUsage>, ALUs <aluUsage>, ROM16s <rom16Usage>, and SSRAMs <ssramUsage>, exceeds the logic resource limits <logicAvailable> of the current device, the above error will be reported.

Action

Choose a device that meets the required logic resource quantity or modify the design.

RP0007

ERROR (RP0007): There is no <type> resource in current device, please use user assignment to change the inference result or change device

Detect DSP and SSRAM resources. When the synthesized primitive type <type> is not supported by the current device, the above error will be reported.

Action

You can modify the ramstyle/dspstyle to change the inference results and ensure that the synthesized primitives are supported by the device.

RP0008

ERROR (RP0008): There is no <type> resource in current device, please change device

When the synthesized primitive type <type> is not supported by the current device, the above error will be reported.

Action

Select a device that supports the primitive type.

RP0009

ERROR (RP0009): The number(<logicUsage>) of logic in the design exceeds the resource limit(<logicAvailable>) of current device, the logic resource usage is <logicUsage> (<lutUsage> LUTs, <aluUsage> ALUs, <rom16Usage> ROM16s)/ <logicAvailable>, <ssramUsage> SSRAMs/<ssramAvailable>

Detect a specific device. When the number of logic resources used in synthesis, including LUTs (lutUsage), ALUs (aluUsage), ROM16s (rom16Usage), and SSRAMs (ssramUsage), exceeds the logic resource limits (logicAvailable) of the current device, the above error will be reported.

Action

Modify the user design.

RP0010

ERROR (RP0010): Cannot instantiate SSRAM with initial value in current device, please delete the initial value of <inst> (<type>), or change other device

When instantiating an SSRAM with an initial value, if the current device does not support it, the above error will be reported.

Action

Delete the initial value of the instance <inst> (of type <type>) or select a device that supports this configuration.

RP0011

ERROR (RP0011): Cannot instantiate <inst>(<type>), there is no <type> resource in current device, or change other device

If the instantiated BSRAM (DPB/DPX9B) is not supported by the current device, the above error will be reported.

Action

Delete the instance <inst> (type <type>) or choose a device that supports this configuration.

RP0012

ERROR (RP0012): Cannot instantiate <inst>(<type>) below <bits>-bit width, please use <type> with bit width of <bits>, or change other device

When instantiating an instance <inst> of type <type>, if the bit width is less than <bits>, the above error will be reported.

Action

When instantiating a primitive of type <type>, ensure the bit width is set to <bits> or choose a different device.

RP0013

ERROR (RP0013): There is no <type1> resource in current device, please use <type2>

Check MIPI_OBUF. The error will be reported when instantiating a primitive of type <type1> with the selected device.

Action

Modify the instantiation to use a primitive of type <type2>.

RP0014

ERROR (RP0014): The peak memory exceeds maximum synthesis memory limit, please check the number of resources in the design

When the memory peak exceeds the maximum limit of Gowinsynthesis during synthesis, the above error will be reported.

Action

Check the resource usage in the design and modify the design accordingly.

RP0015

ERROR (RP0015): <primitive1> and <primitive2> cannot be used together

Primitives <primitive1> and <primitive2> cannot be used simultaneously because they occupy the same location. If both types of primitives appear at the same time in the synthesis, the above error will be reported.

Action

Modify the design to avoid using primitives <primitive1> and <primitive2> simultaneously.

SC0002

ERROR (SC0002): Cannot open property constraint file <file>

Due to reasons such as lack of permission, the specified property constraint file <file> cannot be opened during synthesis, then the above error will be reported.

Action

Check the property file <file> to ensure it is accessible and can be read.

SC0003

ERROR (SC0003): <property> value set invalid <value>

When a property constraint <property> is set to an invalid value <value>, the above error will be reported. For example, in the GowinSynthesis Constraints File (.gsc), if the global constraint syn_ramstyle is set to an invalid value, this error will occur.

```
GLOBAL syn_ramstyle=register
```

Action

Check the value of the property constraint <property> and modify it to a valid value.

SC0004

ERROR (SC0004): <name> is not declared

When the name <name> of the property constraint object is not declared, this error will be reported. For example, in the GowinSynthesis Constraints File (.gsc), if there is no object named "mem" in the design, this error will be reported.

```
INS "mem" syn_ramstyle=registers
```

Action

Check the declaration of the property constraint.

SC0005

ERROR (SC0005): Name <name> does not match with any valid name

When the name of the property constraint object <name> contains a wildcard and is not declared, the above error will be reported.

Action

Check the declaration of the property constraints.

SC0006

ERROR (SC0006): Syntax error near token <text>

When there is a syntax error in the property constraints file, the above error will be reported, indicating the error location as <text>. For example, in the GowinSynthesis Constraints File (.gsc), there is a syntax error in the constraint setting statement.

```
INS "mem" a syn_ramstyle=registers
```

Action

Check the property constraints file and modify the syntax error based on the error message.

SC0007

WARN (SC0007): Override property <property> of <name> to <value>

When a property <property> is set repeatedly for a name <name>, the value <value> will be the last set value, overriding the previous one, and this will trigger the above warning. For example, in the GowinSynthesis Constraints File (.gsc), if the constraint syn_ramstyle is set multiple times for "mem", it will be set to the last specified value, block_ram.

```
INS "mem" syn_ramstyle=registers

INS "mem" syn_ramstyle=block_ram
```

Action

To eliminate this warning, ensure that each property constraint for a given name is assigned a value only once.

SC0008

WARN (SC0007): Override property <property> to <value>

When a global property constraint <property> is set multiple times, the value <value> of the last setting will overwrite the previous one, then the above warning will be reported. For example, in the GowinSynthesis Constraints File (.gsc), if the global constraint syn_ramstyle is set repeatedly, it will be assigned the last specified value, block_ram.

```
GLOBAL syn_ramstyle=registers

GLOBAL syn_ramstyle=block_ram
```

Action

To eliminate this warning, check the property constraint and ensure that each type of global constraint is set only once.

SC0010

WARN (SC0010): Property <property> set invalid in file, maybe it should be set in RTL design

Some property <property> can only be set in RTL design. When these

properties are set in the constraint file, the above warning will be reported, and the property settings will be ignored.

Action

To eliminate this warning and properly set the constraint, configure the constraint in the RTL design.

SP0001

ERROR (SP0001): Check out license failed, please check the gwlicense.ini file and ensure the license is available

When license verification fails, the above message will be reported.

Action

Check the gwlicense.ini file to ensure the license is available.

SP0002

ERROR (SP0002): Corrupted project file: <file>

Corrupted project file <file>. If the project file format is incorrect, the above error will be reported.

Action

Check the project file content to ensure that its format is correct.

SP0003

ERROR (SP0003): No design file specified

When running synthesis without specifying a design file, the above error will be reported.

Action

Create at least one design file and enable the design file to be synthesized before running synthesis.

SP0005

WARN (SP0005): The project file: <file> include illegal file type: <type>

When the project file <file> contains an invalid file type <type>, the above warning will be reported, and the corresponding file will be ignored during synthesis.

Action

To eliminate the warning, check the project file for any invalid file types and either remove or correct them.

SP0006

WARN (SP0006): The project file:<file> include illegal option type: <type>

If the project file <file> contains invalid option types <type>, the corresponding options will be ignored during synthesis, and the above warning will be reported.

Action

To eliminate the above warning, check the project file for any invalid option types and either remove or correct them.

SP0007

ERROR(SP0007): Generate output file <file> failed

Due to a lack of permissions or other reasons, the synthesis output file generation failed.

SP0008

ERROR (SP0008): Cannot open file: <file>

Due to a lack of permissions or other reasons, the file can not be opened during synthesis.

SP0011

ERROR (SP0011): A critical exception has been captured which may trigger application crash

A critical exception that may potentially trigger a program crash has been detected.

SP0012

WARN (SP0012): Include path <path> does not exist

When the specified include path does not exist, the above warning will be reported, and the include path will be ignored.

Action

To eliminate the above warning, verify that the include path exists

SP00017

ERROR(SP00017): Synthesis process cannot run due to error. <error>(<file>:<line>)

GOWIN_ASSERT, an internal error occurred during synthesis.

SP00018

ERROR(SP00018): Synthesis process cannot run due to error.<error>

GOWIN_ASSERT, an internal error occurred during synthesis.

SP00019

ERROR(SP00019): Synthesis process internal error: 10<location><internalid><error>

GOWIN_ASSERT, an internal error occurred during synthesis.

SP00020

ERROR(SP00020): Synthesis process internal error: 10<location><internalid>

GOWIN_ASSERT, an internal error occurred during synthesis.

SP0021

ERROR(SP0021): Could not support backgrand programming set to I2C mode in RTL GAO process.

The above error will be reported when IP I2C background programming files (i2cBgpFile) and .gao files both are present while RTL GAO process configuration is not supported.

Action

Remove one of the files.

SP0022

WARN(SP0022): I2C backgrand programming mode not support

in current device

When the corresponding files are found and the selected device does not support I2C background programming mode, the above warning will be reported.

Action

Choose a device that supports I2C background programming mode.

SP0024

WARN(SP0024): Active flash mode not support in current device.

When the corresponding files are found and the selected device does not support Active Flash mode, the above warning will be reported.

Action

Select a device that supports Active Flash mode.

SP0025

ERROR(SP0025): Read file error: <file>

The above error will be reported if there is a failure to read the file due to reasons such as insufficient permissions.

SP0026

ERROR(SP0026): Global target frequency must be > 0 and <= 1200Mhz

When the STA default global target frequency specified in the project is not within the range, the above error will be reported.

Action

Set the value of the Global target frequency to be within the range of >0 and ≤1200 MHz.

ST0001

WARN(ST0001): Ignored static timing analysis because of the failure of circuit analysis checking

The above warning will be reported due to a timing analysis failure.

Action

To eliminate the above warning, check the netlist timing.

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

Modify the constraint file to remove duplicate constraints.

```
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

Modify the constraint file to remove duplicate constraints..

```
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
```

TA1052

WARN(TA1052): Generated clock is ignored

In create_generated_clock constraint, due to the specified clock name in the -name already being defined in the master clock or if no port is specified for the <object>, the creation of the generated clock fails, then the above warning will pop up.

```
create_ generated_clock -name clk1 -source [get_ports{clkin}] –master_clock clk –edges{2 6 8} –edge_shift {6 5 3}
```

Action

Modify the create_ generated_clock constraint statement to ensure compliance with the requirements.

```
create_ generated_clock -name clk1 -source [get_ports{clkin}] –master_clock clk –edges{2 6 8} –edge_shift {6 5 3} [get_ports {a}]
```

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}]
```

TA1083

WARN(TA1083): <name> is port ,should use "get_ports"

In sdc constraint, when the -through parameter is a port, it is mandatory to use get_ports. In the following example, within the set_multicycle_path constraint, if the -through parameter is a port and get_nets or any other is used, the above warning will pop up.

```
set_multicycle_path -from [get_clocks {clk}] –through [get_nets{clk}] -to [get_clocks {genClk}] -hold
```

Action

The -through parameter utilizes get_ports to obtain the port name.

```
set_multicycle_path -from [get_clocks {clk}] –through [get_ports{clk}] -to [get_clocks {genClk}] -hold
```

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}]
```

TA1125

NOTE (TA1125): More than <num> critical paths are ignored because having large logic level

The number of logic levels on a timing path in the design exceeds the maximum logic levels supported by the software.

Action

Modify the design so that the number of logic levels on all timing paths is less than or equal to 250.

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