

# SPARTAN6 - DSP48A1

Project 1

Saeed Maher Saeed

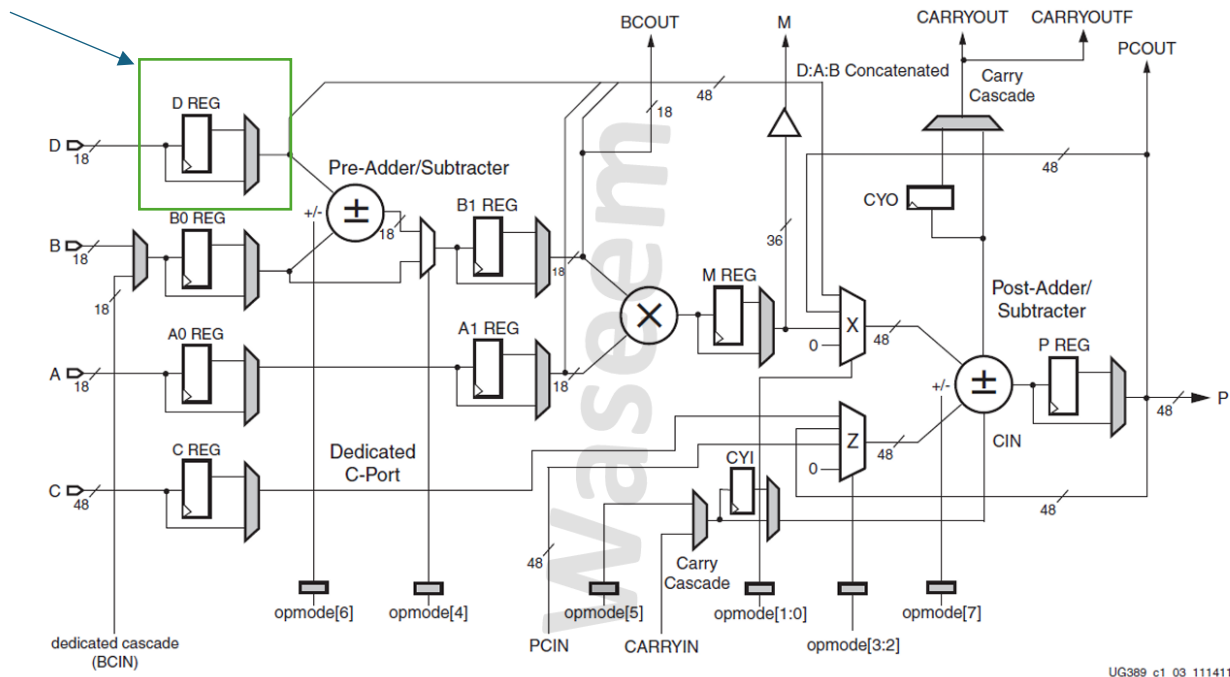
## Table of Contents

Introduction .....	3
Register and MUX part: .....	4
The RTL code design: .....	4
The testbench code with waveform: .....	5
For register = 0 & TYPE = “SYNC”: .....	5
For register = 0 & TYPE = “ASYNCR””: .....	6
For register = 1 & TYPE = “SYNC”: .....	7
For register = 1 & TYPE = “ASYNCR””: .....	8
The lint tool: .....	9
Main DSB part: .....	11
The RTL code design: .....	11
The testbench code with waveform: .....	13
➤ Code: .....	13
➤ Waveform: .....	18
The lint tool: .....	19
The synthesis tool “Vavido”: .....	19
➤ Elaboration: .....	19
1. MSG of no errors or critical warning: .....	19
2. Schematic: .....	20
➤ Synthesis: .....	20
1. MSG of no errors or critical warning: .....	20
2. Schematic: .....	21
3. Utilization report: .....	22
4. Timing report: .....	23
➤ Implementation: .....	23
1. MSG of no errors or critical warning: .....	23
2. Device: .....	24
3. Utilization report: .....	24

<b>4. Timing report:</b>	25
➤ Clock:	25

# Introduction

The design of the DSP is divided into instantiation of mux register as in the square of the schematic and then implement the other parts of the design.



The file will be divided into 2 parts one for the mux register and the other part for the main design each part will have its schematic from lint and vavido to make sure it is right also its testbench with the design code.

You can find here a link with the codes and do file to run their testbenches also a text file called description to guide you each do file do what. [The link](#)

## Register and MUX part:

### The RTL code design:

```
module REG_MUX(clk,rst,clk_en,D,Q);
parameter register = 1;
parameter TYPE = "SYNC";
parameter WIDTH = 1;
input [WIDTH-1:0]D;
input clk,rst,clk_en;
output reg [WIDTH-1:0]Q;
generate
    if(register == 1) begin
        if(TYPE == "SYNC") begin
            always @(posedge clk) begin
                if(rst) begin
                    Q<=0;
                end
                else if(clk_en) begin
                    Q<=D;
                end
            end
        end
        else begin
            always @(posedge clk or posedge rst) begin
                if (rst) begin
                    Q<=0;
                end
                else begin
                    if(clk_en) begin
                        Q<=D;
                    end
                end
            end
        end
    end
end
else begin
    always @(*) begin
        Q=D;
    end
end
endgenerate
endmodule
```

## The testbench code with waveform:

For register = 0 & TYPE = "SYNC":

```
module REG_MUX_tb1();
reg D_tb,clk_tb,rst_tb,clk_en_tb;
wire Q_tb;

REG_MUX #(.register(0), .TYPE("SYNC")) dut (clk_tb,rst_tb,clk_en_tb,D_tb,Q_tb);

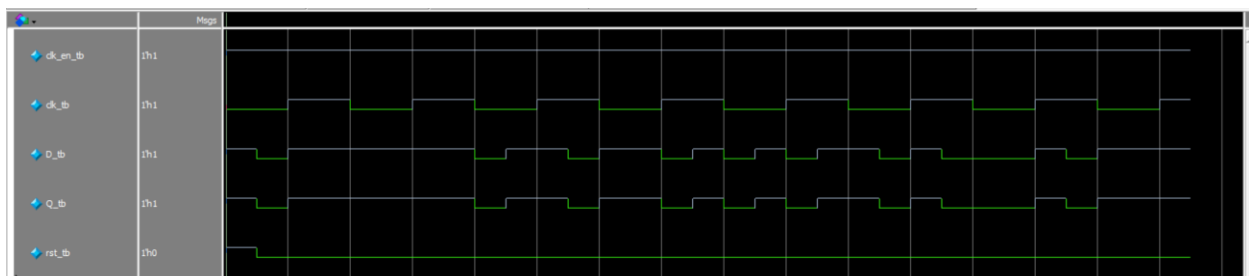
initial begin
    clk_tb=0;
    forever begin
        #10
        clk_tb = ~clk_tb;
    end
end

initial begin
    rst_tb=1;
    D_tb=1;
    clk_en_tb=1;
    #5

    rst_tb=0;
    repeat (30) begin
        D_tb=$random;
        #5;
    end

    $stop;
end

endmodule
```



since the reset signal doesn't affect it and also the output is changing with D immediately without waiting the clock then it is passed.

For register = 0 & TYPE = "ASYNC":

```
module REG_MUX_tb2();
reg D_tb,clk_tb,rst_tb;
wire Q_tb;

REG_MUX #(.register(0), .TYPE("ASYNC")) dut (clk_tb,rst_tb,D_tb,Q_tb);

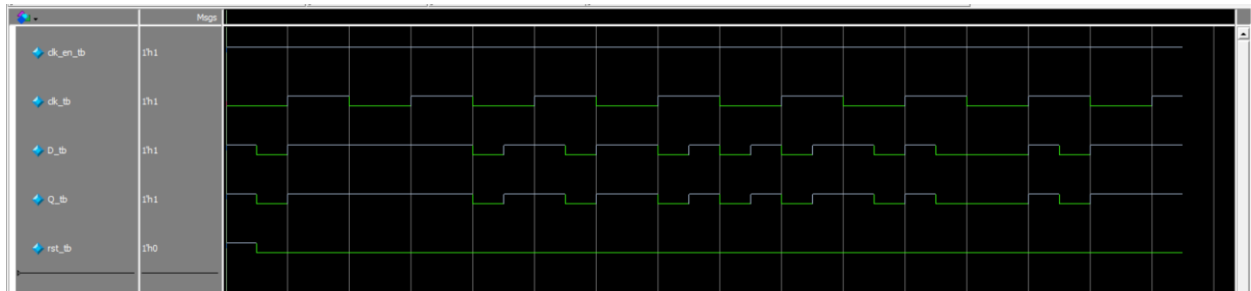
initial begin
    clk_tb=0;
    forever begin
        #10
        clk_tb = ~clk_tb;
    end
end

initial begin
    rst_tb=1;
    D_tb=1;
    #5

    rst_tb=0;
    repeat (30) begin
        D_tb=$random;
        #5;
    end

    $stop;
end

endmodule
```



since the reset signal doesn't affect it and also the output is changing with D immediately without waiting the clock then it is passed.

For register = 1 & TYPE = "SYNC":

```
module REG_MUX_tb3();
reg D_tb,clk_tb,rst_tb,clk_en_tb;
wire Q_tb;

REG_MUX #(.register(1), .TYPE("SYNC")) dut (clk_tb,rst_tb,clk_en_tb,D_tb,Q_tb);

initial begin
    clk_tb=0;
    forever begin
        #10
        clk_tb = ~clk_tb;
    end
end

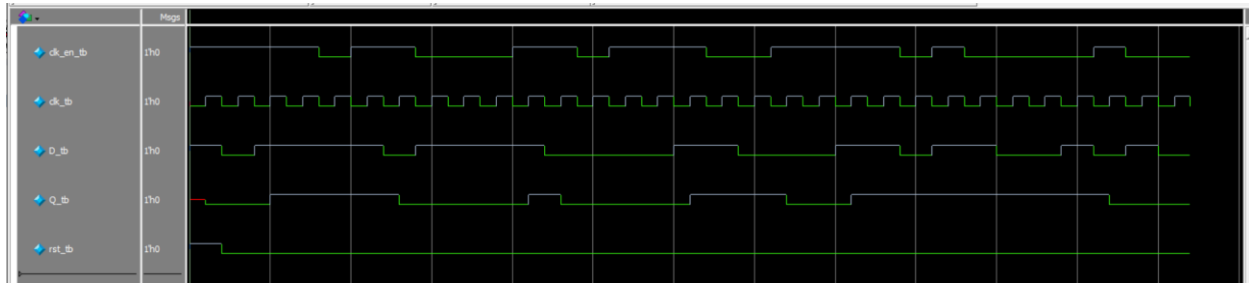
initial begin
    rst_tb=1;
    D_tb=1;
    clk_en_tb=1;
    @(negedge clk_tb);

    rst_tb=0;
    repeat (30) begin
        D_tb=$random;
        clk_en_tb=$random;
        @(negedge clk_tb);
    end

    $stop;
end

endmodule
```





Since that the output appears at the edge of the clock when the clock enable signal is high and that the reset signal waits for the clock and not reset the output immediately then it is passed.

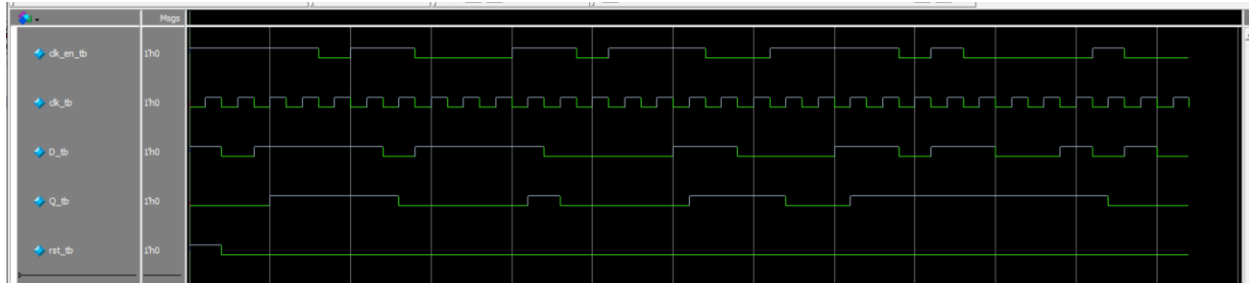
For register = 1 & TYPE = "ASYNC":

```
module REG_MUX_tb4();
reg D_tb,clk_tb,rst_tb,clk_en_tb;
wire Q_tb;
REG_MUX #(.register(1), .TYPE("ASYNC")) dut (clk_tb,rst_tb,clk_en_tb,D_tb,Q_tb);

initial begin
    clk_tb=0;
    forever begin
        #10
        clk_tb = ~clk_tb;
    end
end

initial begin
    rst_tb=1;
    D_tb=1;
    clk_en_tb=1;
    @(negedge clk_tb);

    rst_tb=0;
    repeat (30) begin
        D_tb=$random;
        clk_en_tb=$random;
        @(negedge clk_tb);
    end
    $stop;
end
endmodule
```



Since that the output appears at the edge of the clock and that the reset signal doesn't wait for the clock and resets the output immediately then it is passed.

## The lint tool:

In this part, it is check that this module is synthesizable or not also if it shows the correct schematic.

Questa Lint 2021.1 (D:/last/lint.db)

File Edit View Schematic Window Help

Design

Search: Type Search... Exact Hier

Instance Module De

REG\_MUX REG\_MUX

Flow Navigator Design

Lint Checks

Filter: Type here

Severity Status Check Alias Message Module Category State Owner STARC Reference

Severity	Status	Check	Alias	Message	Module	Category	State	Owner	STARC Reference
Warning	Warning	multi_ports_in_single_line		Multiple ports are declared in one line. Module REG_MUX...	REG_MUX	Rtl Design Style	open	unassign...	3.5.6.3

Transcript Message Viewer Lint Checks Design Metrics Design Information Status History Lint Dashboard

D:/last/MUX\_REG.v [REG\_MUX]

Schematic 1

Expr

1'b0 0 0D1 D0

D

rst

clk

clk\_en

Schematic 1 Lint Summary Schematic 1

Questa Lint 2021.1 (D:/last/lint.db)

File Edit View Schematic Window Help

Design

Search: Type Sear... Exact Hier

Instance Module De

REG\_MUX REG\_MUX To

1 module REG\_MUX(clk,rst,clk\_en,D,Q);

2 parameter

3 register = 1;

4 parameter TYPE = "ASYNC";

5 parameter ASYNC

6 parameter WIDTH = 1;

7 input [WIDTH-1:0]D;

8 input clk,rst,clk\_en;

9 output reg [WIDTH-1:0]Q;

10 generate

11 if(register == 1) begin

12 if(TYPE == "SYNC") begin

13 ASYNC

14 always @(posedge clk) begin

15 if(rst) begin

16 Q<=0;

17 end

18 else if(clk\_en) begin

19 Q<=D;

20 end

21 end

22 end

Schematic 1

Q

D

clk\_en

EN

clk

RST

rst

Q

Schematic 1

Schematic 1 Lint Summary

Lint Checks

Filter: Type here

Severity Status Check Alias Message Module Category State Owner STARC Reference

async\_reset\_active\_high Asynchronous reset is active high. Reset rst, Module ... REG\_MUX Clock open unassign... 2.3.6.2

multi\_ports\_in\_single\_line Multiple ports are declared in one line. Module REG\_MUX REG\_MUX Rtl Design Style open unassign... 3.5.6.3

Transcript Message Viewer Lint Checks Design Metrics Design Information Status History Lint Dashboard

D:/last/MUX\_REG.v [REG\_MUX]

Questa Lint 2021.1 (D:/last/lint.db)

File Edit View Schematic Window Help

Design

Search: Type Sear... Exact Hier

Instance Module De

REG\_MUX REG\_MUX To

1 module REG\_MUX(clk,rst,clk\_en,D,Q);

2 parameter

3 register = 0;

4 parameter TYPE = "ASYNC";

5 parameter ASYNC

6 parameter WIDTH = 1;

7 input [WIDTH-1:0]D;

8 input clk,rst,clk\_en;

9 output reg [WIDTH-1:0]Q;

10 generate

11 if(register == 1) begin

12 if(TYPE == "SYNC") begin

13 ASYNC

14 always @(posedge clk) begin

15 if(rst) begin

16 Q<=0;

17 end

18 else if(clk\_en) begin

19 Q<=D;

20 end

21 end

22 end

Schematic 1

clk\_en

D

Q

clk

rst

Schematic 1

Schematic 1 Lint Summary

Lint Checks

Filter: Type here

Severity Status Check Alias Message Module Category State Owner STARC Reference

multi\_ports\_in\_single\_line Multiple ports are declared in one line. Module REG\_MUX REG\_MUX Rtl Design Style open unassign... 3.5.6.3

Transcript Message Viewer Lint Checks Design Metrics Design Information Status History Lint Dashboard

D:/last/MUX\_REG.v [REG\_MUX]

From the above snippets, it appears that the schematic is right and synthesizable.

## Main DSB part:

### The RTL code design:

```
module
DSB(A,B,C,D,BCIN,CARRYIN,M,P,CARRYOUT,CARRYOUTF,clk,opmode,CEA,CEB,CEC,CECARRYIN,
CED,CEM,CEOPMODE,CEP,RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP,BCOUT,PCO
UT,PCIN);
// parameters
parameter A0REG = 0;
parameter A1REG = 1;
parameter B0REG = 0;
parameter B1REG = 1;
parameter CREG = 1;
parameter DREG = 1;
parameter MREG = 1;
parameter PREG = 1;
parameter CARRYINREG = 1;
parameter CARRYOUTREG = 1;
parameter OPMODEREG = 1;
parameter CARRYINSEL = "OPMODE5";
parameter B_INPUT = "DIRECT";
parameter RSTTYPE = "SYNC";
// data ports
input [17:0]A,B,D,BCIN;
input [47:0]C;
input CARRYIN;
output [35:0]M;
output [47:0]P;
output CARRYOUT,CARRYOUTF;
// control input ports
input clk;
input [7:0]opmode;
// clock enable input ports
input CEA,CEB,CEC,CECARRYIN,CED,CEM,CEOPMODE,CEP;
// Reset Input Ports
input RSTA,RSTB,RSTC,RSTCARRYIN,RSTD,RSTM,RSTOPMODE,RSTP;
// cascade ports
output [17:0]BCOUT;
output [47:0]PCOUT;
input [47:0]PCIN;
// internal wires connection
wire [17:0] B0_reg_in;
wire[7:0]OPMODEREG_out;
wire [17:0] A0REG_out,B0REG_out,DREG_out;
```

```

wire [47:0] CREG_out;
wire [17:0] A1REG_out;
wire [17:0] pre_AS_result; // Pre adder/subtractor result
wire [17:0] B1_REG_in,B1REG_out;
wire [35:0] mult,MREG_out;
wire CYI_in,CYI_REG_out;
reg [47:0] X_MUX,Z_MUX;
wire [47:0] post_AS_result; // post adder/subtractor result
wire CYO_in;

assign B0_reg_in = (B_INPUT == "DIRECT")? B:
                   (B_INPUT == "CASCADE")? BCIN:0;

  REG_MUX #(.WIDTH(8), .register(OPMODEREG), .TYPE(RSTTYPE)) REG_OPMODE
(clk,RSTOPMODE,CEOPMODE,opmode,OPMODEREG_out);
  REG_MUX #(.WIDTH(18), .register(A0REG), .TYPE(RSTTYPE)) REG_A0
(clk,RSTA,CEA,A,A0REG_out);
  REG_MUX #(.WIDTH(18), .register(B0REG), .TYPE(RSTTYPE)) REG_B0
(clk,RSTB,CEB,B0_reg_in,B0REG_out);
  REG_MUX #(.WIDTH(48), .register(CREG), .TYPE(RSTTYPE)) REG_C
(clk,RSTC,CEC,C,CREG_out);
  REG_MUX #(.WIDTH(18), .register(DREG), .TYPE(RSTTYPE)) REG_D
(clk,RSTD,CED,D,DREG_out);
  REG_MUX #(.WIDTH(18), .register(A1REG), .TYPE(RSTTYPE)) REG_A1
(clk,RSTA,CEA,A0REG_out,A1REG_out);

assign pre_AS_result = (OPMODEREG_out[6])? (DREG_out-
B0REG_out):(DREG_out+B0REG_out);
assign B1_REG_in = (OPMODEREG_out[4])? pre_AS_result:B0REG_out;
  REG_MUX #(.WIDTH(18), .register(B1REG), .TYPE(RSTTYPE)) REG_B1
(clk,RSTB,CEB,B1_REG_in,B1REG_out);

assign mult = B1REG_out*A1REG_out;
  REG_MUX #(.WIDTH(36), .register(MREG), .TYPE(RSTTYPE)) REG_M
(clk,RSTM,CEM,mult,MREG_out);

assign CYI_in = (CARRYINSEL == "CARRYIN")? CARRYIN:
                (CARRYINSEL == "OPMODE5")? OPMODEREG_out[5]:0;

  REG_MUX #(.WIDTH(1), .register(CARRYINREG), .TYPE(RSTTYPE)) REG_CYI
(clk,RSTCARRYIN,CECARRYIN,CYI_in,CYI_REG_out);

always @(*) begin
    case (OPMODEREG_out[1:0])
        0: X_MUX = 0;

```

```

        1: X_MUX = {12'b0,MREG_out};
        2: X_MUX = P;
        3: X_MUX = {DREG_out[11:0],A1REG_out,B1REG_out};
    endcase
end

always @(*) begin
    case (OPMODEREG_out[3:2])
        0: Z_MUX = 0;
        1: Z_MUX = PCIN;
        2: Z_MUX = P;
        3: Z_MUX = CREG_out;
    endcase
end

assign M = MREG_out;

assign {CYO_in,post_AS_result} = (OPMODEREG_out[7])? (Z_MUX-
(X_MUX+CYI_REG_out)):(Z_MUX+X_MUX+CYI_REG_out);

    REG_MUX #(.WIDTH(1), .register(CARRYOUTREG), .TYPE(RSTTYPE)) REG_CYO
(clk,RSTCARRYIN,CECARRYIN,CYO_in,CARRYOUT);
    REG_MUX #(.WIDTH(48), .register(PREG), .TYPE(RSTTYPE)) REG_P
(clk,RSTP,CEP,post_AS_result,P);

assign PCOUT=P;
assign CARRYOUTF=CARRYOUT;
assign BCOUT=B1REG_out;
endmodule

```

The testbench code with waveform:

➤ Code:

```

module DSP_tb();
reg [17:0]A_tb,B_tb,D_tb,BCIN_tb;
reg [47:0]C_tb;
reg CARRYIN_tb;
wire [35:0]M_tb;
wire [47:0]P_tb;
wire CARRYOUT_tb,CARRYOUTF_tb;
reg clk_tb;
reg [7:0]opmode_tb;
reg CEA_tb,CEB_tb,CEC_tb,CECARRYIN_tb,CED_tb,CEM_tb,CEOPMODE_tb,CEP_tb;

```

```

reg RSTA_tb,RSTB_tb,RSTC_tb,RSTCARRYIN_tb,RSTD_tb,RSTM_tb,RSTOPMODE_tb,RSTP_tb;
wire [17:0]BCOUT_tb;
wire [47:0]PCOUT_tb;
reg [47:0]PCIN_tb;
// self checking regs
reg [35:0]M_expected;
reg [47:0]P_expected;
reg [17:0]BCOUT_expected;
reg [47:0]PCOUT_expected;
reg CARRYOUT_expected,CARRYOUTF_expected;

DSP dut
(A_tb,B_tb,C_tb,D_tb,BCIN_tb,CARRYIN_tb,M_tb,P_tb,CARRYOUT_tb,CARRYOUTF_tb,clk_tb
,opmode_tb,CEA_tb,CEB_tb,CEC_tb,CECARRYIN_tb,CED_tb,CEM_tb,CEOPMODE_tb,CEP_tb,RST
A_tb,RSTB_tb,RSTC_tb,RSTCARRYIN_tb,RSTD_tb,RSTM_tb,RSTOPMODE_tb,RSTP_tb,BCOUT_tb,
PCOUT_tb,PCIN_tb);

initial begin
    clk_tb=0;
    forever begin
        #10
        clk_tb = ~clk_tb;
    end
end

initial begin
    // reset check
    RSTA_tb=1;
    RSTB_tb=1;
    RSTC_tb=1;
    RSTCARRYIN_tb=1;RSTD_tb=1;
    RSTM_tb=1;
    RSTOPMODE_tb=1;
    RSTP_tb=1;
    A_tb=$random;
    B_tb=$random;
    D_tb=$random;
    C_tb=$random;
    BCIN_tb=$random;
    PCIN_tb=$random;
    CARRYIN_tb=$random;
    opmode_tb=$random;
    CEA_tb=0;
    CEB_tb=0;
    CEC_tb=0;

```

```

CECARRYIN_tb=0;
CED_tb=0;
CEM_tb=0;
CEOPMODE_tb=0;
CEP_tb=0;
@(negedge clk_tb);
M_expected=0;
P_expected=0;
BCOUT_expected=0;
PCOUT_expected=0;
CARRYOUT_expected=0;
CARRYOUTF_expected=0;
if (M_expected != M_tb || P_expected != P_tb || BCOUT_expected != BCOUT_tb ||
PCOUT_expected != PCOUT_tb || CARRYOUT_expected != CARRYOUT_tb ||
CARRYOUTF_expected != CARRYOUTF_tb) begin
    $display("Error in reset");
    $stop;
end
RSTA_tb=0;
RSTB_tb=0;
RSTC_tb=0;
RSTCARRYIN_tb=0;RSTD_tb=0;
RSTM_tb=0;
RSTOPMODE_tb=0;
RSTP_tb=0;
CEA_tb=1;
CEB_tb=1;
CEC_tb=1;
CECARRYIN_tb=1;
CED_tb=1;
CEM_tb=1;
CEOPMODE_tb=1;
CEP_tb=1;

// path 1 check
A_tb=20;
B_tb=10;
C_tb=350;
D_tb=25;
BCIN_tb=$random;
PCIN_tb=$random;
CARRYIN_tb=$random;
opmode_tb=8'b1101_1101;
repeat(4) @(negedge clk_tb);
M_expected='h12c;

```



```

P_expected='h32;
BCOUT_expected='hf;
PCOUT_expected='h32;
CARRYOUT_expected=0;
CARRYOUTF_expected=0;
if (M_expected != M_tb || P_expected != P_tb || BCOUT_expected != BCOUT_tb ||
PCOUT_expected != PCOUT_tb || CARRYOUT_expected != CARRYOUT_tb ||
CARRYOUTF_expected != CARRYOUTF_tb) begin
    $display("Error in path 1");
    $stop;
end

// check path 2
BCIN_tb=$random;
PCIN_tb=$random;
CARRYIN_tb=$random;
opmode_tb=8'b0001_0000;
repeat(3) @(negedge clk_tb);
M_expected='h2bc;
P_expected='h0;
BCOUT_expected='h23;
PCOUT_expected='h0;
CARRYOUT_expected=0;
CARRYOUTF_expected=0;
if (M_expected != M_tb || P_expected != P_tb || BCOUT_expected != BCOUT_tb ||
PCOUT_expected != PCOUT_tb || CARRYOUT_expected != CARRYOUT_tb ||
CARRYOUTF_expected != CARRYOUTF_tb) begin
    $display("Error in path 2");
    $stop;
end

// check path 3
BCIN_tb=$random;
PCIN_tb=$random;
CARRYIN_tb=$random;
opmode_tb=8'b0000_1010;
repeat(3) @(negedge clk_tb);
M_expected='hc8;
BCOUT_expected='ha;
if (M_expected != M_tb || P_expected != P_tb || BCOUT_expected != BCOUT_tb ||
PCOUT_expected != PCOUT_tb || CARRYOUT_expected != CARRYOUT_tb ||
CARRYOUTF_expected != CARRYOUTF_tb) begin
    $display("Error in path 3");
    $stop;
end

```

```

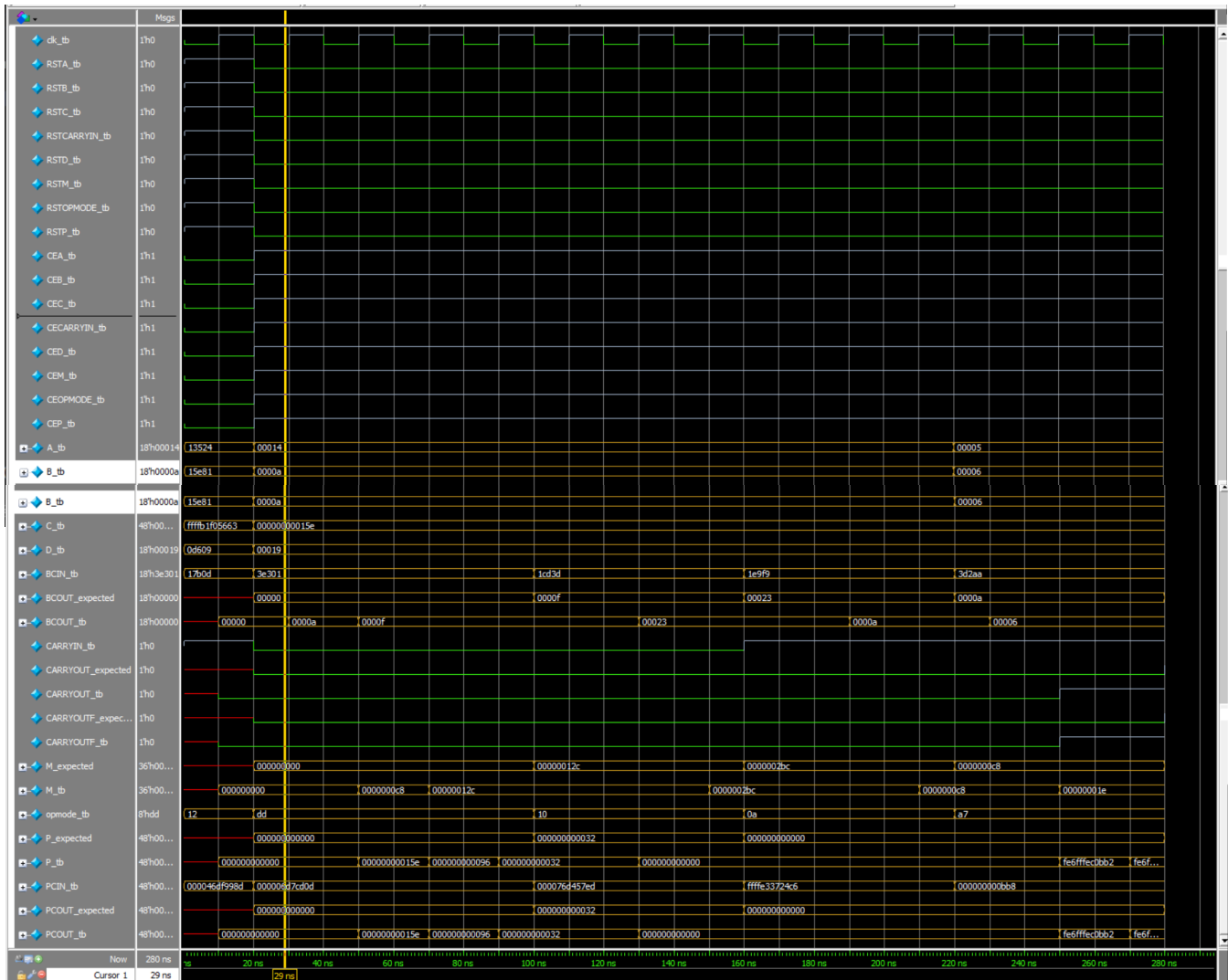
// check path 4
A_tb=5;
B_tb=6;
C_tb=350;
D_tb=25;
BCIN_tb=$random;
PCIN_tb=3000;
CARRYIN_tb=$random;
opmode_tb=8'b1010_0111;
repeat(3) @(negedge clk_tb);
M_expected='h1e;
BCOUT_expected='h6;
P_expected='hfe6ffffec0bb1;
PCOUT_expected='hfe6ffffec0bb1;
CARRYOUT_expected=1;
CARRYOUTF_expected=1;
if (M_expected != M_tb || P_expected != P_tb || BCOUT_expected != BCOUT_tb ||
PCOUT_expected != PCOUT_tb || CARRYOUT_expected != CARRYOUT_tb ||
CARRYOUTF_expected != CARRYOUTF_tb) begin
    $display("Error in path 4");
    $stop;
end

$stop;
end

endmodule

```

## ➤ Waveform:



Note: the waveform is too long to be in one picture so I divided it into 2 and tried to make them under each other for easier tracing.

## DO file:

vlib work

vlog MUX\_REG.v MAIN\_DSP.v DSP\_tb.v

vsim -voptargs=+acc work.DSP\_tb

add wave \*

run -all

#quit -sim

## The lint tool:

Questa Lint 2021.1 (D:/PROJ1/LINT/lint.db)

File Edit View Schematic Window Help

Design

Search: Type Search...

Instance Module

DSB (11) DSB

module DSB(A,B,C,D,BCIN,CARRYIN,M,P,  
// parameters  
parameter A0REG = 0;  
0  
parameter A1REG = 1;  
1  
parameter B0REG = 0;  
0  
parameter B1REG = 1;  
1  
parameter CREG = 1;  
1  
parameter DREG = 1;  
1  
parameter MREG = 1;  
1  
parameter PREG = 1;  
1  
parameter CARRYINREG = 1;  
1  
parameter CARRYOUTREG = 1;  
1  
parameter OPMODEREG = 1;  
1  
parameter CARRYINSEL = "OPMODES";  
OPMODES

Schematic 1

Lint Summary

Lint Checks

Filter: Type here

Severity Status Check Alias Message Module Category State Owner STARC Reference

Warning	condition_const	Condition expression is a constant. Module DSB, File ...	DSB	Rtl Design Style	open	unassign...	
Warning	condition_const	Condition expression is a constant. Module DSB, File ...	DSB	Rtl Design Style	open	unassign...	
Warning	multi_ports_in_single_line	Multiple ports are declared in one line. Module DSB, ...	DSB	Rtl Design Style	open	unassign...	3.5.6.3
Warning	multi_ports_in_single_line	Multiple ports are declared in one line. Module REG_...	REG_MUX	Rtl Design Style	open	unassign...	3.5.6.3

Transcript Message Viewer Lint Checks Design Metrics Design Information Status History Lint Dashboard

D:/PROJ1/LINT/MAIN\_DSB.v (DSB)

## The synthesis tool "Vavido":

### ➤ Elaboration:

#### 1. MSG of no errors or critical warning:

project\_1 - [D:/last/project\_1\_xpr] - Vivado 2018.2

File Edit Flow Tools Repgrts Window Layout View Help Quick Access

Flow Navigator

PROJECT MANAGER

- Settings
- Add Sources
- Language Templates
- IP Catalog

IP INTEGRATOR

- Create Block Design
- Open Block Design
- Generate Block Design

SIMULATION

- Run Simulation

RTL ANALYSIS

- Open Elaborated Design
- Report Methodology
- Report DRC
- Report Noise
- Schematic

SYNTHESIS

ELABORATED DESIGN - xc7a200tfg1156-3 (active)

Sources

- Design Sources (1)
- DSP (MAIN\_DSB.v) (11)
- Constraints (1)
- consts\_1 (1)
- Simulation Sources (1)
- Constraints\_basys3.xdc

Source File Properties

Constraints\_basys3.xdc

Enabled

Location: D:/last

General Properties

Project Summary

Constraints\_basys3.xdc

D:/last/Constraints\_basys3.xdc

```
1: ## This file is a general .xdc for the Basys3 rev B board
2: ## To use it in a project:
3: ## - uncomment the lines corresponding to used pins
4: ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5:
6: ## Clock signal
7: set_property -dict { PACKAGE_PIN W5 IOSTANDARD LVCMOS33 } [get_ports clk]
8: create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
9:
10:
11:
12: ## Configuration options, can be used for all designs
13: set_property CONFIG_VOLTAGE 3.3 [current_design]
14: set_property CPGSVS VCCO [current_design]
15:
16: ## SPI configuration mode options for QSPI boot, can be used for all designs
17: set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
```

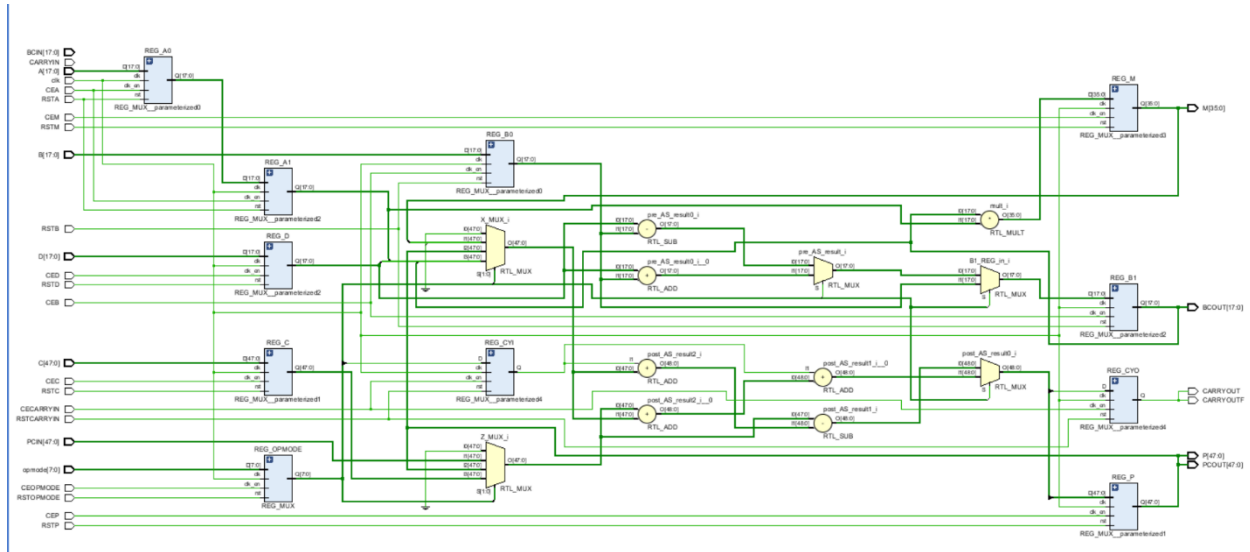
Tcl Console Messages Log Reports Design Runs

Elaborated Design (22 warnings)

- General Messages (22 warnings)
- [Synth 8-3331] design REG\_MUX\_\_parameterized0 has unconnected port clk (21 more like this)

Note: the warnings is that the BCIN and the CARRYIN are not used due to the pre defined MUX\_REG as in description nothing more than that.

## 2. Schematic:



## ➤ Synthesis:

### 1. MSG of no errors or critical warning:

project\_1 - [D:/last/project\_1.xpr] - Vivado 2018.2

File Edit Flow Tools Repgrts Window Layout View Help Quick-Access

Synthesis Complete

Debug

Flow Navigator

PROJECT MANAGER

- Settings
- Add Sources
- Language Templates
- IP Catalog

IP INTEGRATOR

- Create Block Design
- Open Block Design
- Generate Block Design

SIMULATION

- Run Simulation

RTL ANALYSIS

- Open Elaborated Design
- Report Methodology
- Report DRC
- Report Noise
- Schematic

SYNTHESIS

- Run Synthesis

SYNTHESIZED DESIGN - xc7a200t@y1156-3 (active)

Sources

- REG\_B1 (REG\_MUX\_\_parameterized2\_0)
- REG\_C (REG\_MUX\_\_parameterized1)
- REG\_C1 (REG\_MUX\_\_parameterized4)
- REG\_C2 (REG\_MUX\_\_parameterized4\_1)
- REG\_D (REG\_MUX\_\_parameterized2\_2)
- REG\_M (REG\_MUX\_\_parameterized3)
- REG\_OPMODE (REG\_MUX)
- REG\_P (REG\_MUX\_\_parameterized1\_3)

Cell Properties

REG\_C1

Name: REG\_C1

General Properties Statistics Nets Cell

Schematic

Constraints\_basys3.xdc

```

1: ## This file is a general .xdc for the Basys3 rev B board
2: ## To use it in a project:
3: ## - uncomment the lines corresponding to used pins
4: ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5:
6: ## Clock signal
7: set_property -dict { PACKAGE_PIN W6 IOSTANDARD LVCMOS33 } [get_ports clk]
8: create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
9:
10:
11:
12: ## Configuration options, can be used for all designs
13: set_property CONFIG_VOLTAGE 3.3 [current_design]
14: set_property CFGMEM_VCCO [current_design]
15:
16: ## SPI configuration mode options for QSPI boot, can be used for all designs
17: set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]

```

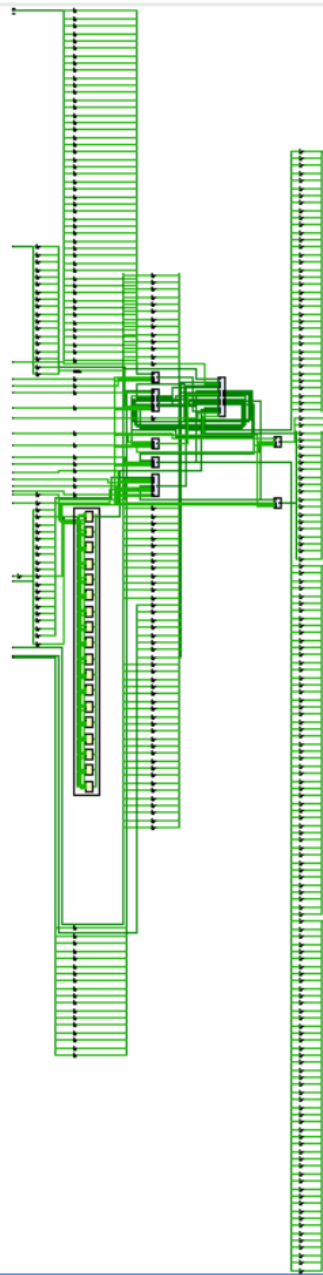
Tcl Console Messages Log Reports Design Runs Debug

Warning (42) Info (41) Status (19) Show All

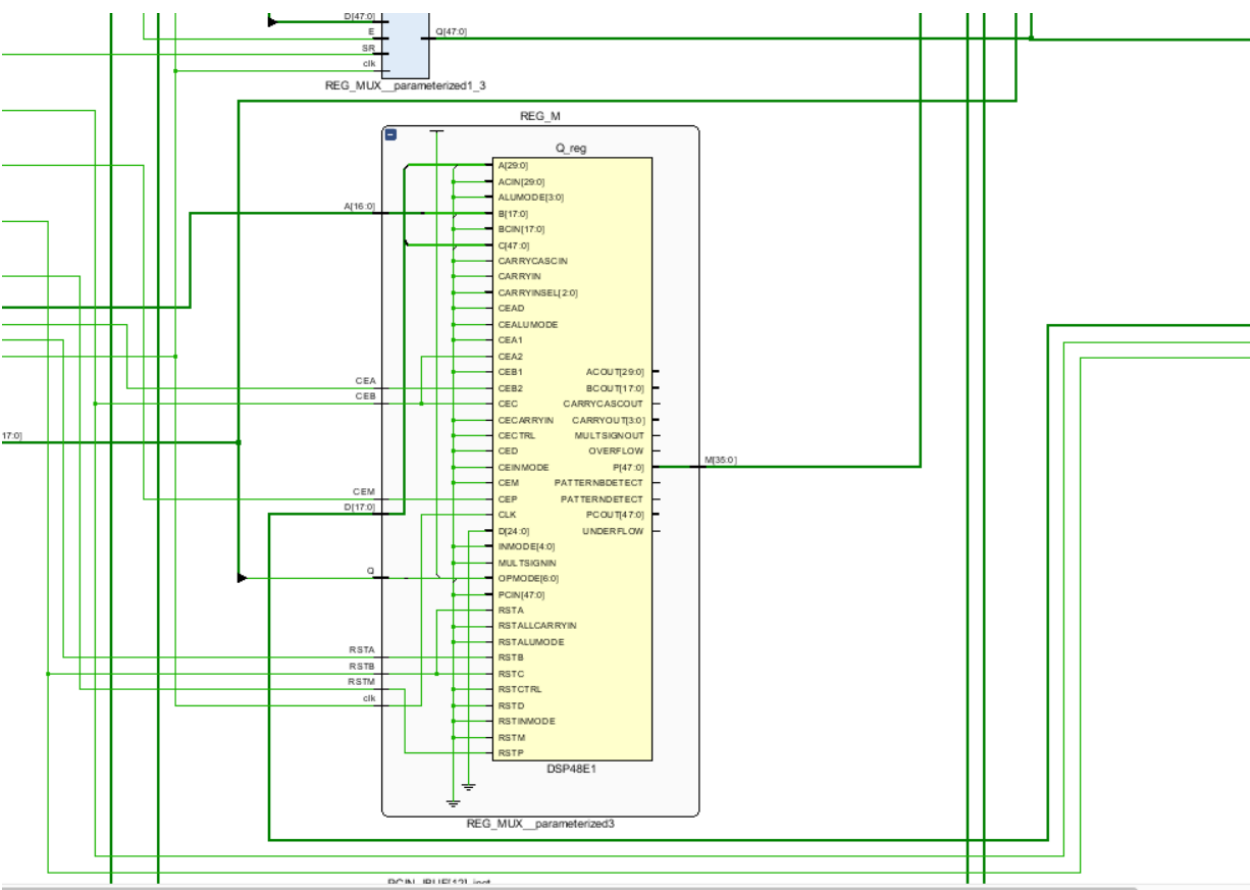
Synthesis (42 warnings)

- [Synth 8-3331] design REG\_MUX\_\_parameterized0 has unconnected port dk (40 more like this)
- [Constraints 18-5210] No constraint will be written out

## 2. Schematic:



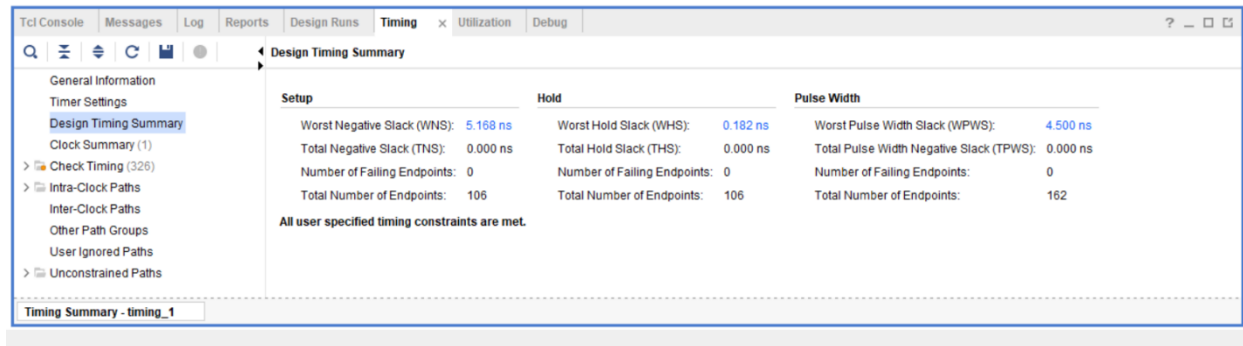
The DSP block would be found inside the REG\_M as shown in the following figure:



### 3. Utilization report:

Tcl Console Messages Log Reports Design Runs Utilization x Debug						
Hierarchy						
Hierarchy						
Summary						
▼ Slice Logic						
▼ Slice LUTs (<1%)						
LUT as Logic (<1%)						
▼ Slice Registers (<1%)						
Register as Flip Flop (<1						
Memory						
▼ DSP						
▼ DSPs (<1%)						
DSP48E1 only						
▼ IO and GT Specific						
▼ Bonded IOB (65%)						
IOB Master Pads						
▼ Clocking						
BUFGCTRL (3%)						
Name	1	Slice LUTs (134600)	Slice Registers (269200)	DSP s (740 )	Bonded IOB (500)	BUFGCTRL (32)
▼ N DSP		230	160	1	327	1
I REG_A1 (REG_MUX_...		0	18	0	0	0
I REG_B1 (REG_MUX_...		0	18	0	0	0
I REG_C (REG_MUX_...		0	48	0	0	0
I REG_CY1 (REG_MUX_...		1	1	0	0	0
I REG_CYO (REG_MUX_...		0	1	0	0	0
I REG_D (REG_MUX_...		0	18	0	0	0
I REG_M (REG_MUX_...		0	0	1	0	0
I REG_OPMODE (REG_...		228	8	0	0	0
I REG_P (REG_MUX_...		0	48	0	0	0

## 4. Timing report:



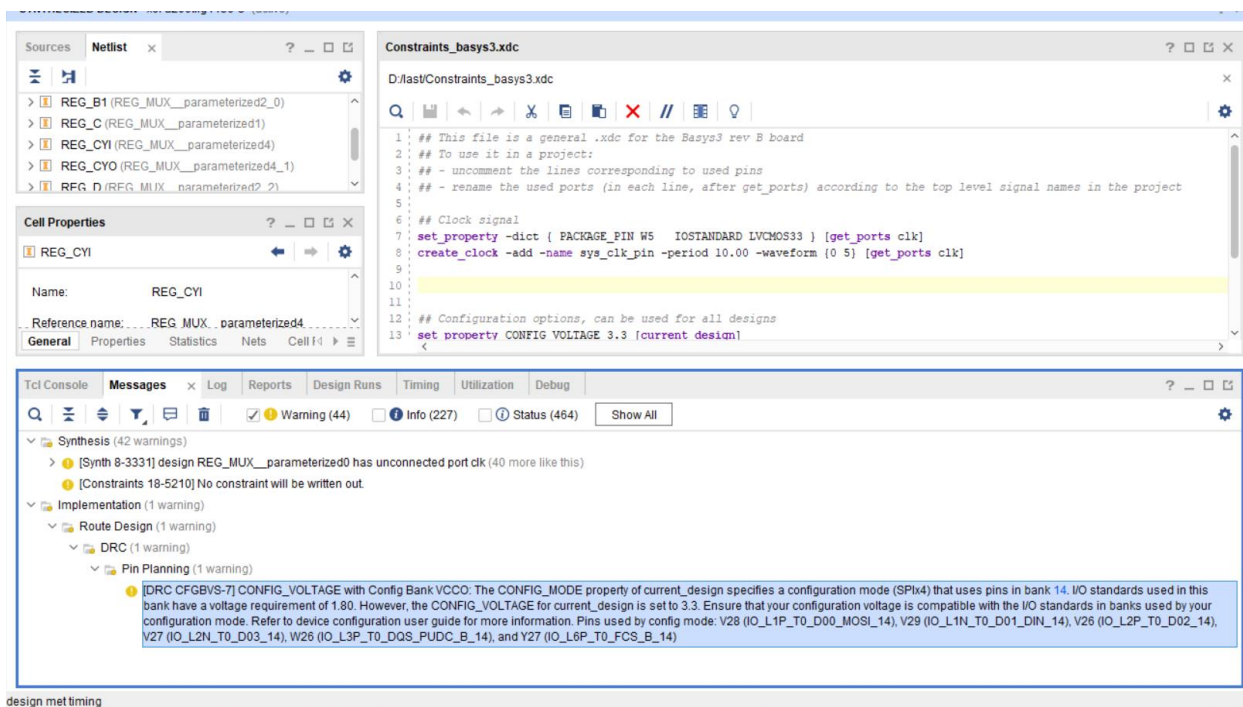
The screenshot shows the 'Design Timing Summary' window. It contains a table with three columns: Setup, Hold, and Pulse Width. The data is as follows:

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.168 ns	Worst Hold Slack (WHS): 0.182 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 106	Total Number of Endpoints: 106	Total Number of Endpoints: 162

Below the table, it states: 'All user specified timing constraints are met.'

## ➤ Implementation:

### 1. MSG of no errors or critical warning:



The screenshot shows the 'Messages' window with a list of warnings. The first warning is:

[Synth 8-3331] design REG\_MUX\_\_parameterized0 has unconnected port clk (40 more like this)

The second warning is:

[Constraints 18-5210] No constraint will be written out.

The third warning is:

[DRC 1 warning]

The fourth warning is:

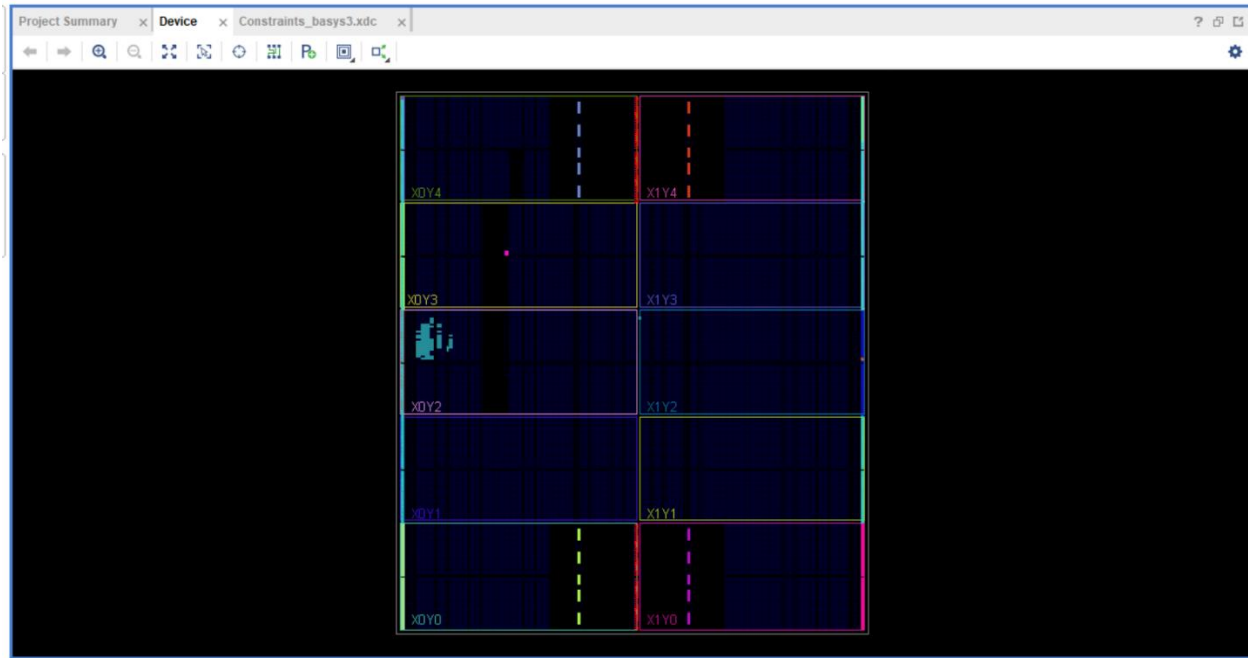
[DRC 1 warning]

The fifth warning is:

[DRC CFGBVS-7] CONFIG\_VOLTAGE with Config Bank VCC0: The CONFIG\_MODE property of current\_design specifies a configuration mode (SPH4) that uses pins in bank 14. I/O standards used in this bank have a voltage requirement of 1.80. However, the CONFIG\_VOLTAGE for current\_design is set to 3.3. Ensure that your configuration voltage is compatible with the I/O standards in banks used by your configuration mode. Refer to device configuration user guide for more information. Pins used by config mode: V28 (IO\_L1P\_T0\_D00\_MOSI\_14), V29 (IO\_L1N\_T0\_D01\_DIN\_14), V26 (IO\_L2P\_T0\_D02\_14), V27 (IO\_L2N\_T0\_D03\_14), V26 (IO\_L3P\_T0\_D0S\_PUOC\_B\_14), and V27 (IO\_L6P\_T0\_FCS\_B\_14)



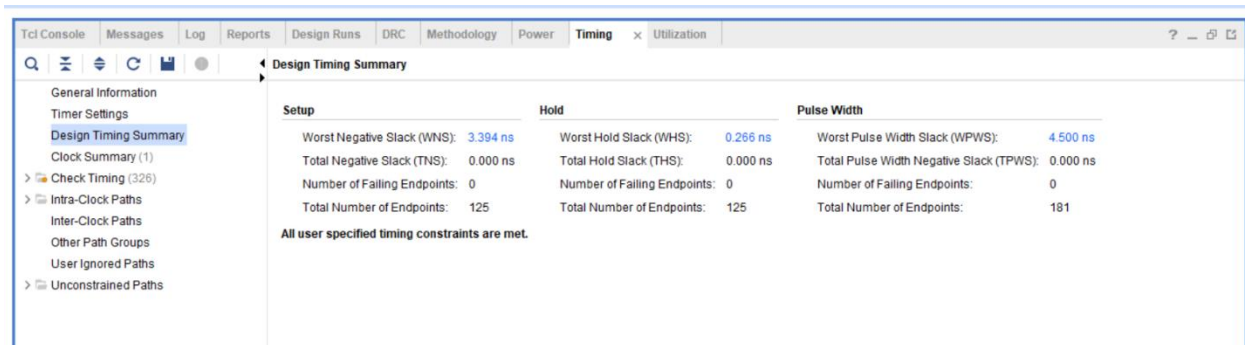
## 2. Device:



### 3. Utilization report:

Tcl Console Messages Log Reports Design Runs DRC Methodology Power Timing Utilization x									
Hierarchy									
Name	Slice LUTs (133800)	Slice Registers (267600)	Slice (33450)	LUT as Logic (133800)	LUT Flip Flop Pairs (133800)	DSPs (740)	Bonded IOB (500)	BUFGCTRL (32)	
<b>Hierarchy</b>									
Summary									
▼ Slice Logic									
▼ Slice LUTs (<1%)									
LUT as Logic (<1%)									
▼ Slice Registers (<1%)									
Register as Flip Flop (<1%)									
▼ Slice Logic Distribution									
▼ Slice (1%)									
SLICEM									
SLICEL									
▼ LUT Flip Flop Pairs (<1%)									
LUT-FF pairs with one									
LUT-FF pairs with one									
▼ LUT as Logic (<1%)									
using O5 and O6									
using O6 output only									
▼ DSP	229	179	104	229	51	1	327	1	
REG_A1 (REG_MUX_...)	0	18	7	0	0	0	0	0	
REG_B1 (REG_MUX_...)	0	36	17	0	0	0	0	0	
REG_C (REG_MUX_...)	0	48	17	0	0	0	0	0	
REG_CY1 (REG_MUX_...)	1	1	1	1	1	0	0	0	
REG_CYO (REG_MUX_...)	0	2	2	0	0	0	0	0	
REG_D (REG_MUX_...)	0	18	8	0	0	0	0	0	
REG_M (REG_MUX_...)	0	0	0	0	0	1	0	0	
REG_OPMODE (REG_...)	228	8	70	228	0	0	0	0	
REG_P (REG_MUX_...)	0	48	12	0	0	0	0	0	

## 4. Timing report:

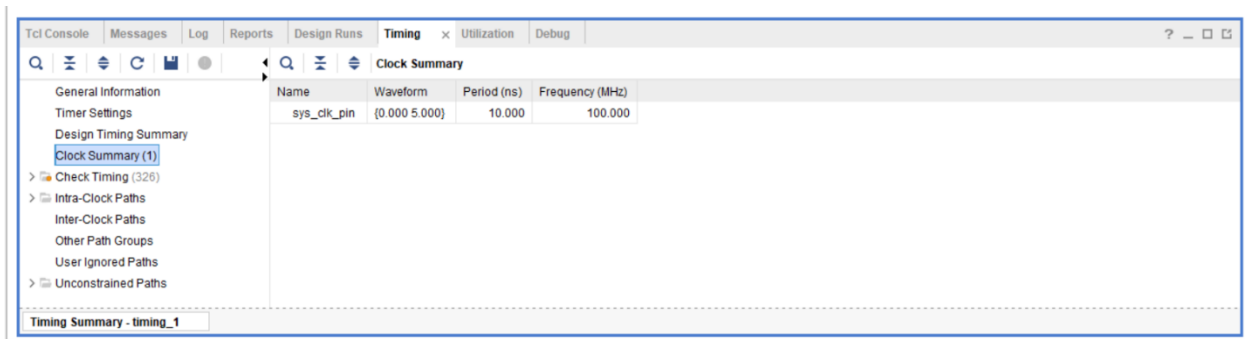


The screenshot shows the 'Design Timing Summary' window. The left sidebar contains a tree view with 'Design Timing Summary' selected. The main area displays timing metrics for Setup, Hold, and Pulse Width constraints.

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 3.394 ns	Worst Hold Slack (WHS): 0.266 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 125	Total Number of Endpoints: 125	Total Number of Endpoints: 181

All user specified timing constraints are met.

### ➤ Clock:



The screenshot shows the 'Clock Summary' window. The left sidebar contains a tree view with 'Clock Summary (1)' selected. The main area displays a table with clock signal details.

Name	Waveform	Period (ns)	Frequency (MHz)
sys_clk_pin	{0.000 5.000}	10.000	100.000

## The constraint file:

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top
level signal names in the project

## Clock signal
set_property -dict { PACKAGE_PIN W5    IOSTANDARD LVCMOS33 } [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]

## Configuration options, can be used for all designs
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCC0 [current_design]
## SPI configuration mode options for QSPI boot, can be used for all designs
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
```