**The Digital Race**

**Understanding the problem:**

* We have to use single 8-bit input data bus on which we obtain 3 inputs A, B and C in 3 consecutive clock cycles.
* This means each clock pulse and input will arrive at the same time.
* Hence, in the test bench we gave each input at the positive edge of clock pulse.
* We have to create single output 9-bit bus which will give 3 required sums in 3 consecutive clock cycles in the specified sequence.

**Devising the plan:**

* We receive inputs and display outputs in 6 consecutive clock cycles in proper sequence i.e. ‘A’ at the arrival of 1st clock pulse,

‘B’ at the arrival of 2nd clock pulse and so on.

* So, there is a need to count clock pulses. Hence, we used counter to count clock pulses.
* We have made 3 input variables A,B & C for the 3 inputs.
* Due to the counter we have the number(1st,2nd) of the clock pulse. Depending on the number we assign data on the input bus to our variable. i.e. on 1st Clock pulse we assign the data to A. Similarly for B & C.
* We get the each input(A,B,C) on the same input 8 bit data bus for only one clock cycle. Hence we need to store it as in the next clock cycle new input will be obtained on the same data bus.
* Hence we need to create a 8 bit Latch for each input. This is achieved by module ALU\_mini.
* For finding the required 3 sums we require 3 adders. We used CLAs for this purpose. This is achieved by module LAC.
* For the respective clock pulses we display the sums and sig\_out signal.

**Carrying out the plan:**

Verilog code:

Module LAC:

module FA(a,b,c\_in,sum,c\_out);

input a;

input b;

input c\_in;

output sum;

output c\_out;

assign sum=a^b^c\_in;

assign c\_out=(a&b)|(b&c\_in)|(a&c\_in);

endmodule

module adder1(a,b,c\_in,sum,c\_out);

input[3:0] a,b;

input c\_in;

output[3:0]sum;

output c\_out;

wire cA,cB,cC;

FA inst1(a[0],b[0],c\_in,sum[0],cA);

FA inst2(a[1],b[1],cA,sum[1],cB);

FA inst3(a[2],b[2],cB,sum[2],cC);

FA inst4(a[3],b[3],cC,sum[3],c\_out);

endmodule

module x(a,b,c\_in,sum,c\_out);

input [3:0] a;

input [3:0] b;

input c\_in;

output [3:0]sum;

output c\_out;

wire g1;

reg p1;

wire y1;

wire c\_out13;

adder1 inst\_1(a,b,c\_in,sum,c\_out13);

always @ (a,b)

begin

if(a[0]==~b[0] && a[1]==~b[1] && a[2]==~b[2] && a[3]==~b[3])

p1=1'b1;

else

p1=1'b0;

end

assign g1=(a[3]&b[3])+((a[3]^b[3])&(a[2]&b[2]))+((a[3]^b[3])&(a[2]^b[2])&(a[1]&b[1]))+((a[3]^b[3])&(a[2]^b[2])&(a[1]^b[1])&(a[0]&b[0]));

and(y1,p1,c\_in);

or(c\_out,g1,y1);

endmodule

module LAC(a,b,p,q,g,h,sum1,sum2,sum3,c\_out1,c\_out3,c\_out5);

input[7:0]a;

input[7:0]b;

input[7:0]p;

input[7:0]q;

input[7:0]g;

input[7:0]h;

output[7:0]sum1;

output[7:0]sum2;

output[7:0]sum3;

output c\_out1;

output c\_out3;

output c\_out5;

wire c\_in=1'b0;

wire c\_out6=1'b0;

wire c\_out7=1'b0;

x x1(a[3:0],b[3:0],c\_in,sum1[3:0],c\_out);

x x2(a[7:4],b[7:4],c\_out,sum1[7:4],c\_out1);

x x3(p[3:0],q[3:0],c\_out6,sum2[3:0],c\_out2);

x x4(p[7:4],q[7:4],c\_out2,sum2[7:4],c\_out3);

x x5(g[3:0],h[3:0],c\_out7,sum3[3:0],c\_out4);

x x6(g[7:4],h[7:4],c\_out4,sum3[7:4],c\_out5);

endmodule

Module ALU\_mini:

module ALU\_mini(

input [7:0]A,B,C,

input [2:0]count,

output reg[7:0]x,y,z

);

always@(\*)

begin

if(count==3'b001)

x<=A;

else

x<=x;

end

always@(\*)

begin

if(count==3'b010)

y<=B;

else

y<=y;

end

always@(\*)

begin

if(count==3'b011)

z<=C;

else

z<=z;

end

endmodule

Module Processor:

module processor(

input sig\_in,

input clk,

input [7:0]bus,

output reg sig\_out,

output reg [8:0]sum

);

reg [2:0]count=3'b000;

reg [7:0]A,B,C;

wire [8:0]Y1,Y2,Y3;

wire [7:0]sum1,sum2,sum3;

wire carry1,carry2,carry3;

wire [7:0]x,y,z;

LAC lac(x,y,x,z,y,z,sum1,sum2,sum3,carry1,carry2,carry3);

ALU\_mini inst(A,B,C,count,x,y,z);

assign Y1={carry1,sum1};

assign Y2={carry2,sum2};

assign Y3={carry3,sum3};

always @ (posedge clk)

begin

count<=count+1;

end

always @ (\*)

begin

if(count==3'b001 && sig\_in==1'b1)

A=bus;

else

A=8'b00000000;

end

always @ (\*)

begin

if(count==3'b010 && sig\_in==1'b0)

B=bus;

else

B=8'b00000000;

end

always @ (\*)

begin

if(count==3'b011 && sig\_in==1'b0)

C=bus;

else

C=8'b00000000;

end

always @ (\*)

begin

if(count==3'b100)

begin

sum=Y1;

sig\_out=1'b1;

end

else if(count==3'b101)

begin

sum=Y2;

sig\_out=1'b0;

end

else if(count==3'b110)

begin

sum=Y3;

sig\_out=1'b0;

end

else

begin

sum=9'b000000000;

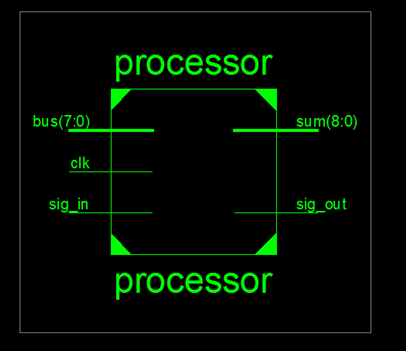
sig\_out=1'b0;

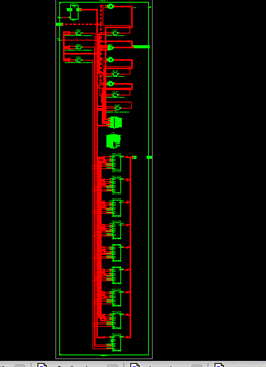
end

end

endmodule

**RTL Schematic :**





**Test Bench:**

module last\_tb;

// Inputs

reg sig\_in;

reg clk;

reg [7:0] bus;

// Outputs

wire sig\_out;

wire [8:0] sum;

// Instantiate the Unit Under Test (UUT)

processor uut (

.sig\_in(sig\_in),

.clk(clk),

.bus(bus),

.sig\_out(sig\_out),

.sum(sum)

);

initial begin

// Initialize Inputs

sig\_in = 0;

clk = 0;

bus = 0;

// Wait 100 ns for global reset to finish

#100;

bus=8'b11111111;

sig\_in=1'b1;

clk=1'b1;

#25;

clk=1'b0;

#25;

bus=8'b00110011;

sig\_in=1'b0;

clk=1'b1;

#25;

clk=1'b0;

#25;

bus=8'b11111001;

sig\_in=1'b0;

clk=1'b1;

#25;

clk=1'b0;

#25;

bus=8'b00000000;

clk=1'b1;

#25;

clk=1'b0;

#25;

clk=1'b1;

#25;

clk=1'b0;

#25;

clk=1'b1;

#25;

clk=1'b0;

#25;

clk=1'b1;

#25;

clk=1'b0;

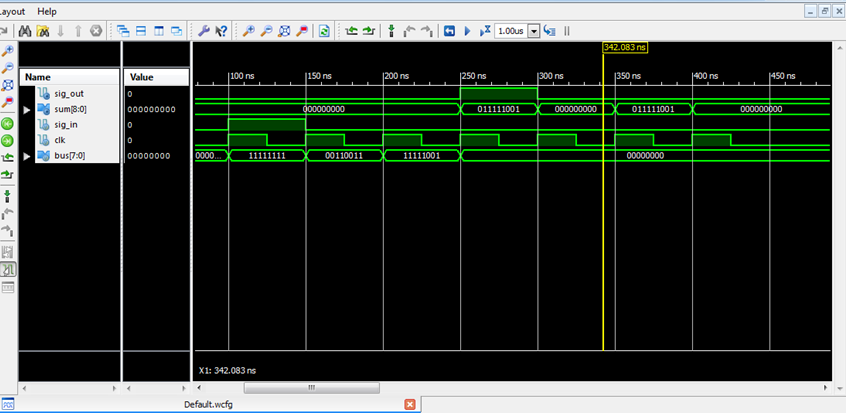
#25;

// Add stimulus here

end

endmodule

**Output:**



**Looking Back:**

* Our goal was to obtain 3 inputs on the single input data bus and obtain outputs on the single output data bus in total 6 consecutive clock cycles.
* So, in this program, we learned how to use basic components of sequential logic design like latches and counters.
* We learned how to give clock pulse and input synchronously in the test bench.

**Synthesis Report:**

=========================================================================

\* Synthesis Options Summary \*

=========================================================================

---- Source Parameters

Input File Name : "processor.prj"

Input Format : mixed

Ignore Synthesis Constraint File : NO

---- Target Parameters

Output File Name : "processor"

Output Format : NGC

Target Device : xc3s200-5-pq208

---- Source Options

Top Module Name : processor

Automatic FSM Extraction : YES

FSM Encoding Algorithm : Auto

Safe Implementation : No

FSM Style : lut

RAM Extraction : Yes

RAM Style : Auto

ROM Extraction : Yes

Mux Style : Auto

Decoder Extraction : YES

Priority Encoder Extraction : YES

Shift Register Extraction : YES

Logical Shifter Extraction : YES

XOR Collapsing : YES

ROM Style : Auto

Mux Extraction : YES

Resource Sharing : YES

Asynchronous To Synchronous : NO

Multiplier Style : auto

Automatic Register Balancing : No

---- Target Options

Add IO Buffers : YES

Global Maximum Fanout : 500

Add Generic Clock Buffer(BUFG) : 8

Register Duplication : YES

Slice Packing : YES

Optimize Instantiated Primitives : NO

Use Clock Enable : Yes

Use Synchronous Set : Yes

Use Synchronous Reset : Yes

Pack IO Registers into IOBs : auto

Equivalent register Removal : YES

---- General Options

Optimization Goal : Speed

Optimization Effort : 1

Library Search Order : processor.lso

Keep Hierarchy : NO

Netlist Hierarchy : as\_optimized

RTL Output : Yes

Global Optimization : AllClockNets

Read Cores : YES

Write Timing Constraints : NO

Cross Clock Analysis : NO

Hierarchy Separator : /

Bus Delimiter : <>

Case Specifier : maintain

Slice Utilization Ratio : 100

BRAM Utilization Ratio : 100

Verilog 2001 : YES

Auto BRAM Packing : NO

Slice Utilization Ratio Delta : 5

=========================================================================

=========================================================================

\* HDL Compilation \*

=========================================================================

Compiling verilog file "LAC.v" in library work

Module <FA> compiled

Module <adder1> compiled

Module <x> compiled

Compiling verilog file "ALU\_mini.v" in library work

Module <LAC> compiled

Compiling verilog file "processor.v" in library work

Module <ALU\_mini> compiled

Module <processor> compiled

No errors in compilation

Analysis of file <"processor.prj"> succeeded.

=========================================================================

\* Design Hierarchy Analysis \*

=========================================================================

Analyzing hierarchy for module <processor> in library <work>.

Analyzing hierarchy for module <LAC> in library <work>.

Analyzing hierarchy for module <ALU\_mini> in library <work>.

Analyzing hierarchy for module <x> in library <work>.

Analyzing hierarchy for module <adder1> in library <work>.

Analyzing hierarchy for module <FA> in library <work>.

=========================================================================

\* HDL Analysis \*

=========================================================================

Analyzing top module <processor>.

Module <processor> is correct for synthesis.

Analyzing module <LAC> in library <work>.

Module <LAC> is correct for synthesis.

Analyzing module <x> in library <work>.

Module <x> is correct for synthesis.

Analyzing module <adder1> in library <work>.

Module <adder1> is correct for synthesis.

Analyzing module <FA> in library <work>.

Module <FA> is correct for synthesis.

Analyzing module <ALU\_mini> in library <work>.

Module <ALU\_mini> is correct for synthesis.

=========================================================================

\* HDL Synthesis \*

=========================================================================

Performing bidirectional port resolution...

Synthesizing Unit <ALU\_mini>.

Related source file is "ALU\_mini.v".

WARNING:Xst:737 - Found 8-bit latch for signal <x>. Latches may be generated from incomplete case or if statements. We do not recommend the use of latches in FPGA/CPLD designs, as they may lead to timing problems.

WARNING:Xst:737 - Found 8-bit latch for signal <y>. Latches may be generated from incomplete case or if statements. We do not recommend the use of latches in FPGA/CPLD designs, as they may lead to timing problems.

WARNING:Xst:737 - Found 8-bit latch for signal <z>. Latches may be generated from incomplete case or if statements. We do not recommend the use of latches in FPGA/CPLD designs, as they may lead to timing problems.

Unit <ALU\_mini> synthesized.

Synthesizing Unit <FA>.

Related source file is "LAC.v".

Found 1-bit xor3 for signal <sum>.

Summary:

inferred 1 Xor(s).

Unit <FA> synthesized.

Synthesizing Unit <adder1>.

Related source file is "LAC.v".

Unit <adder1> synthesized.

Synthesizing Unit <x>.

Related source file is "LAC.v".

WARNING:Xst:646 - Signal <c\_out13> is assigned but never used. This unconnected signal will be trimmed during the optimization process.

Found 1-bit adder for signal <g1>.

Found 1-bit adder for signal <g1$addsub0000> created at line 73.

Found 1-bit adder for signal <g1$addsub0001> created at line 73.

Found 1-bit xor2 for signal <g1$xor0000> created at line 73.

Found 1-bit xor2 for signal <g1$xor0001> created at line 73.

Found 1-bit xor2 for signal <g1$xor0002> created at line 73.

Found 1-bit xor2 for signal <p1$xor0000> created at line 67.

Found 1-bit xor2 for signal <p1$xor0001> created at line 67.

Found 1-bit xor2 for signal <p1$xor0002> created at line 67.

Found 1-bit xor2 for signal <p1$xor0003> created at line 67.

Summary:

inferred 3 Adder/Subtractor(s).

Unit <x> synthesized.

Synthesizing Unit <LAC>.

Related source file is "LAC.v".

Unit <LAC> synthesized.

Synthesizing Unit <processor>.

Related source file is "processor.v".

Found 3-bit up counter for signal <count>.

Summary:

inferred 1 Counter(s).

Unit <processor> synthesized.

=========================================================================

HDL Synthesis Report

Macro Statistics

# Adders/Subtractors : 18

1-bit adder : 18

# Counters : 1

**3-bit up counter : 1**

# Latches : 3

**8-bit latch : 3**

# Xors : 66

1-bit xor2 : 42

1-bit xor3 : 24

=========================================================================

=========================================================================

\* Advanced HDL Synthesis \*

=========================================================================

=========================================================================

Advanced HDL Synthesis Report

Macro Statistics

# Adders/Subtractors : 12

1-bit adder : 6

1-bit adder carry in : 6

# Counters : 1

3-bit up counter : 1

# Latches : 3

8-bit latch : 3

# Xors : 66

1-bit xor2 : 42

1-bit xor3 : 24

=========================================================================

=========================================================================

\* Low Level Synthesis \*

=========================================================================

Optimizing unit <processor> ...

Mapping all equations...

Building and optimizing final netlist ...

Found area constraint ratio of 100 (+ 5) on block processor, actual ratio is 4.

FlipFlop count\_1 has been replicated 1 time(s)

Final Macro Processing ...

=========================================================================

Final Register Report

Macro Statistics

# Registers : 4

Flip-Flops : 4

=========================================================================

=========================================================================

\* Partition Report \*

=========================================================================

Partition Implementation Status

-------------------------------

No Partitions were found in this design.

-------------------------------

=========================================================================

\* Final Report \*

=========================================================================

Final Results

RTL Top Level Output File Name : processor.ngr

Top Level Output File Name : processor

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : NO

Design Statistics

# IOs : 20

Cell Usage :

**# BELS : 122**

# LUT2 : 11

# LUT3 : 43

# LUT4 : 59

# MUXF5 : 8

# VCC : 1

# FlipFlops/Latches : 28

# FD : 3

# FDR : 1

# LD : 24

# Clock Buffers : 1

# BUFGP : 1

# IO Buffers : 19

# IBUF : 9

# OBUF : 10

=========================================================================

Device utilization summary:

---------------------------

Selected Device : 3s200pq208-5

Number of Slices: 64 out of 1920 3%

Number of Slice Flip Flops: 28 out of 3840 0%

Number of 4 input LUTs: 113 out of 3840 2%

Number of IOs: 20

Number of bonded IOBs: 20 out of 141 14%

Number of GCLKs: 1 out of 8 12%

---------------------------

Partition Resource Summary:

---------------------------

No Partitions were found in this design.

---------------------------

=========================================================================

TIMING REPORT

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.

FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT

GENERATED AFTER PLACE-and-ROUTE.

Clock Information:

------------------

---------------------------------------+------------------------+-------+

Clock Signal | Clock buffer(FF name) | Load |

---------------------------------------+------------------------+-------+

clk | BUFGP | 4 |

inst/z\_cmp\_eq0000(inst/z\_cmp\_eq00001:O)| NONE(\*)(inst/z\_0) | 8 |

inst/y\_cmp\_eq0000(inst/y\_cmp\_eq00001:O)| NONE(\*)(inst/y\_0) | 8 |

inst/x\_cmp\_eq0000(inst/x\_cmp\_eq00001:O)| NONE(\*)(inst/x\_0) | 8 |

---------------------------------------+------------------------+-------+

(\*) These 3 clock signal(s) are generated by combinatorial logic,

and XST is not able to identify which are the primary clock signals.

Please use the CLOCK\_SIGNAL constraint to specify the clock signal(s) generated by combinatorial logic.

INFO:Xst:2169 - HDL ADVISOR - Some clock signals were not automatically buffered by XST with BUFG/BUFR resources. Please use the buffer\_type constraint in order to insert these buffers to the clock signals to help prevent skew problems.

Asynchronous Control Signals Information:

----------------------------------------

No asynchronous control signals found in this design

Timing Summary:

---------------

Speed Grade: -5

Minimum period: **2.513ns (Maximum Frequency: 398.010MHz)**

Minimum input arrival time before clock: 3.192ns

Maximum output required time after clock: 14.525ns

Maximum combinational path delay: No path found

Timing Detail:

--------------

All values displayed in nanoseconds (ns)

=========================================================================

Timing constraint: Default period analysis for Clock 'clk'

Clock period: 2.513ns (frequency: 398.010MHz)

Total number of paths / destination ports: 8 / 4

-------------------------------------------------------------------------

Delay: 2.513ns (Levels of Logic = 0)

Source: count\_0 (FF)

Destination: count\_0 (FF)

Source Clock: clk rising

Destination Clock: clk rising

Data Path: count\_0 to count\_0

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

---------------------------------------- ------------

FDR:C->Q 13 0.626 0.994 count\_0 (count\_0)

FDR:R 0.892 count\_0

----------------------------------------

Total 2.513ns (1.518ns logic, 0.994ns route)

(60.4% logic, 39.6% route)

=========================================================================

Timing constraint: Default OFFSET IN BEFORE for Clock 'inst/z\_cmp\_eq0000'

Total number of paths / destination ports: 16 / 8

-------------------------------------------------------------------------

Offset: 3.192ns (Levels of Logic = 2)

Source: sig\_in (PAD)

Destination: inst/z\_0 (LATCH)

Destination Clock: inst/z\_cmp\_eq0000 falling

Data Path: sig\_in to inst/z\_0

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

---------------------------------------- ------------

IBUF:I->O 24 0.715 1.822 sig\_in\_IBUF (sig\_in\_IBUF)

LUT3:I0->O 1 0.479 0.000 C<7>1 (C<7>)

LD:D 0.176 inst/z\_7

----------------------------------------

Total 3.192ns (1.370ns logic, 1.822ns route)

(42.9% logic, 57.1% route)

=========================================================================

Timing constraint: Default OFFSET IN BEFORE for Clock 'inst/y\_cmp\_eq0000'

Total number of paths / destination ports: 16 / 8

-------------------------------------------------------------------------

Offset: 3.192ns (Levels of Logic = 2)

Source: sig\_in (PAD)

Destination: inst/y\_0 (LATCH)

Destination Clock: inst/y\_cmp\_eq0000 falling

Data Path: sig\_in to inst/y\_0

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

---------------------------------------- ------------

IBUF:I->O 24 0.715 1.822 sig\_in\_IBUF (sig\_in\_IBUF)

LUT3:I0->O 1 0.479 0.000 B<7>1 (B<7>)

LD:D 0.176 inst/y\_7

----------------------------------------

Total 3.192ns (1.370ns logic, 1.822ns route)

(42.9% logic, 57.1% route)

=========================================================================

Timing constraint: Default OFFSET IN BEFORE for Clock 'inst/x\_cmp\_eq0000'

Total number of paths / destination ports: 16 / 8

-------------------------------------------------------------------------

Offset: 3.192ns (Levels of Logic = 2)

Source: sig\_in (PAD)

Destination: inst/x\_0 (LATCH)

Destination Clock: inst/x\_cmp\_eq0000 falling

Data Path: sig\_in to inst/x\_0

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

---------------------------------------- ------------

IBUF:I->O 24 0.715 1.822 sig\_in\_IBUF (sig\_in\_IBUF)

LUT3:I0->O 1 0.479 0.000 A<7>1 (A<7>)

LD:D 0.176 inst/x\_7

----------------------------------------

Total 3.192ns (1.370ns logic, 1.822ns route)

(42.9% logic, 57.1% route)

=========================================================================

Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'

Total number of paths / destination ports: 59 / 10

-------------------------------------------------------------------------

Offset: 11.315ns (Levels of Logic = 4)

Source: count\_1 (FF)

Destination: sum<1> (PAD)

Source Clock: clk rising

Data Path: count\_1 to sum<1>

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

---------------------------------------- ------------

FD:C->Q 20 0.626 1.608 count\_1 (count\_1)

LUT3:I0->O 5 0.479 1.078 sum<0>21 (N25)

LUT4:I0->O 1 0.479 0.976 sum<1>229\_SW0 (N52)

LUT4:I0->O 1 0.479 0.681 sum<1>229 (sum\_1\_OBUF)

OBUF:I->O 4.909 sum\_1\_OBUF (sum<1>)

----------------------------------------

Total 11.315ns (6.972ns logic, 4.343ns route)

(61.6% logic, 38.4% route)

=========================================================================

Timing constraint: Default OFFSET OUT AFTER for Clock 'inst/y\_cmp\_eq0000'

Total number of paths / destination ports: 123 / 9

-------------------------------------------------------------------------

Offset: 14.525ns (Levels of Logic = 8)

Source: inst/y\_1 (LATCH)

Destination: sum<7> (PAD)

Source Clock: inst/y\_cmp\_eq0000 falling

Data Path: inst/y\_1 to sum<7>

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

---------------------------------------- ------------

LD:G->Q 4 0.551 1.074 inst/y\_1 (inst/y\_1)

LUT4:I0->O 3 0.479 0.771 lac/x5/inst\_1/inst2/c\_out1 (lac/x5/inst\_1/cB)

MUXF5:S->O 3 0.540 0.830 lac/x5/Madd\_g1\_Madd\_xor<0>11\_f5 (lac/c\_out4)

LUT3:I2->O 2 0.479 0.804 lac/x6/inst\_1/inst1/c\_out1 (lac/x6/inst\_1/cA)

LUT3:I2->O 2 0.479 0.915 lac/x6/inst\_1/inst2/c\_out1 (lac/x6/inst\_1/cB)

LUT4:I1->O 1 0.479 0.740 sum<7>205\_SW1 (N102)

LUT3:I2->O 1 0.479 0.000 sum<7>220\_G (N125)

MUXF5:I1->O 1 0.314 0.681 sum<7>220 (sum\_7\_OBUF)

OBUF:I->O 4.909 sum\_7\_OBUF (sum<7>)

----------------------------------------

Total 14.525ns (8.709ns logic, 5.816ns route)

(60.0% logic, 40.0% route)

=========================================================================

Timing constraint: Default OFFSET OUT AFTER for Clock 'inst/z\_cmp\_eq0000'

Total number of paths / destination ports: 124 / 9

-------------------------------------------------------------------------

Offset: 14.473ns (Levels of Logic = 8)

Source: inst/z\_0 (LATCH)

Destination: sum<7> (PAD)

Source Clock: inst/z\_cmp\_eq0000 falling

Data Path: inst/z\_0 to sum<7>

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

---------------------------------------- ------------

LD:G->Q 6 0.551 1.023 inst/z\_0 (inst/z\_0)

LUT4:I1->O 3 0.479 0.771 lac/x5/inst\_1/inst2/c\_out1 (lac/x5/inst\_1/cB)

MUXF5:S->O 3 0.540 0.830 lac/x5/Madd\_g1\_Madd\_xor<0>11\_f5 (lac/c\_out4)

LUT3:I2->O 2 0.479 0.804 lac/x6/inst\_1/inst1/c\_out1 (lac/x6/inst\_1/cA)

LUT3:I2->O 2 0.479 0.915 lac/x6/inst\_1/inst2/c\_out1 (lac/x6/inst\_1/cB)

LUT4:I1->O 1 0.479 0.740 sum<7>205\_SW1 (N102)

LUT3:I2->O 1 0.479 0.000 sum<7>220\_G (N125)

MUXF5:I1->O 1 0.314 0.681 sum<7>220 (sum\_7\_OBUF)

OBUF:I->O 4.909 sum\_7\_OBUF (sum<7>)

----------------------------------------

Total 14.473ns (8.709ns logic, 5.764ns route)

(60.2% logic, 39.8% route)

=========================================================================

Timing constraint: Default OFFSET OUT AFTER for Clock 'inst/x\_cmp\_eq0000'

Total number of paths / destination ports: 126 / 9

-------------------------------------------------------------------------

Offset: 14.378ns (Levels of Logic = 8)

Source: inst/x\_1 (LATCH)

Destination: sum<7> (PAD)

Source Clock: inst/x\_cmp\_eq0000 falling

Data Path: inst/x\_1 to sum<7>

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

---------------------------------------- ------------

LD:G->Q 4 0.551 1.074 inst/x\_1 (inst/x\_1)

LUT4:I0->O 3 0.479 0.771 lac/x3/inst\_1/inst2/c\_out1 (lac/x3/inst\_1/cB)

MUXF5:S->O 3 0.540 0.830 lac/x3/Madd\_g1\_Madd\_xor<0>11\_f5 (lac/c\_out2)

LUT3:I2->O 2 0.479 0.804 lac/x4/inst\_1/inst1/c\_out1 (lac/x4/inst\_1/cA)

LUT3:I2->O 2 0.479 0.768 lac/x4/inst\_1/inst2/c\_out1 (lac/x4/inst\_1/cB)

LUT4:I3->O 1 0.479 0.740 sum<7>54 (sum<7>54)

LUT4:I2->O 1 0.479 0.000 sum<7>220\_F (N124)

MUXF5:I0->O 1 0.314 0.681 sum<7>220 (sum\_7\_OBUF)

OBUF:I->O 4.909 sum\_7\_OBUF (sum<7>)

----------------------------------------

Total 14.378ns (8.709ns logic, 5.669ns route)

(60.6% logic, 39.4% route)

=========================================================================

Total REAL time to Xst completion: 3.00 secs

Total CPU time to Xst completion: 3.18 secs

-->

Total memory usage is 189144 kilobytes

Number of errors : 0 ( 0 filtered)

Number of warnings : 4 ( 0 filtered)

Number of infos : 1 ( 0 filtered)