**PROBLEM STATEMENT**

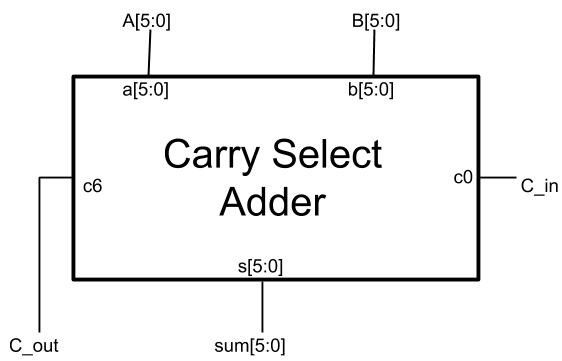
**KEYWORDS:**

1. **INTRODUCTION**

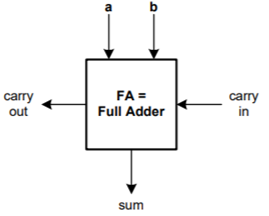
//Adder Types (Ripple adder,

//CSA Overview

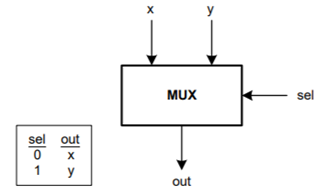
*Figure 6.1.a* CSA Top Level Overview Diagram



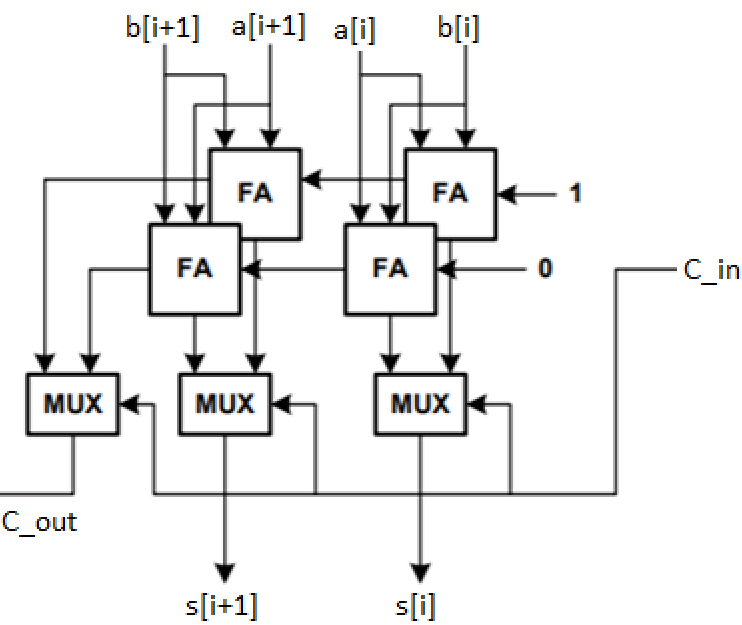
*Figure 6.2* Full Adder Diagram



*Figure 6.3* 2-input Multiplexer Diagram



*Figure 6.4* CSA Mid-Level Diagram



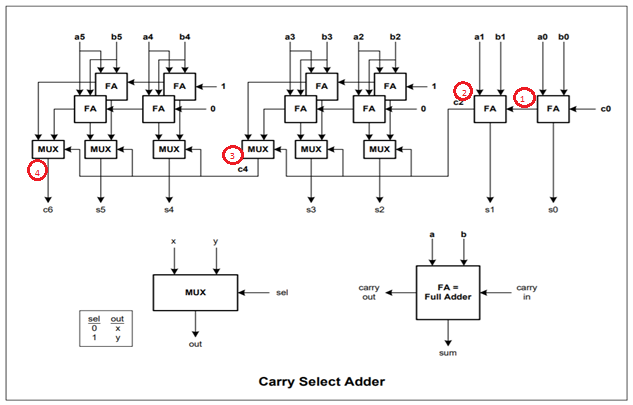
*Table 6.1* CSA Block Delay Table

|  |  |
| --- | --- |
| **Operation** | **Delay (ns)** |
| *Full Adder* |  |
| a to sum | 6 ns |
| b to sum | 6 ns |
| a to carry out | 2 ns |
| b to carry out | 2 ns |
| carry in to carry out | 2 ns |
| carry in to sum | 3 ns |
| *2-input Multiplexer* |  |
| x to out | 2 ns |
| y to out | 2 ns |
| sel to out | 3 ns |

The critical path associated with the CSA block diagram shown in Figure 6.x.a is shown below in red. By adding the associated delays of each block, the delay of the circuit can be calculated. This calculation is performed in Eq 6.1(1) below.

(1)

Figure 6.1.b CSA Top Level Detail Diagram



1. **METHODOLOGY**

//Data Requirements

//Sum of two six bit numbers as a six bit sum with carry overflow.

//To verify the CSA, all input combinations must be tested and the computed sum must be checked against a behavioral adder.

//Testing Strategy

//Non Exhaustive approach: For the non-exhaustive approach, a small subset of 10 test vectors are selected at the boundaries of each case of input. Test vectors 1,2, and 6 handle cases where one input is non-zero and the other inputs are 0. Test vector 3 handles the case where both A and B inputs are non-zero but carry in (c0) is zero. Test Vectors 4, 5, 7 & 8 handle cases where the output carry is set due to binary addition overflow when the input carry is 0. Test Vector 9 handles the case where all inputs are non-zero but there is no overflow. Test Vector 10 handles the case where all inputs are non-zero and the output carry is set.

In the exhaustive approach each combination of inputs is explored using loops. The testbench displays a detailed subset of these exhaustive vectors while confirming the test in comparison to a behaviorally coded adder (also included in the test bench).

To check the output of the CSA, a behavioral test-adder is constructed in the testbench to compare all computed sums. If a case arises where the sum of the CSA and the sum of the test-adder are different, the error condition is thrown by the testbench program and testing is halted. The testbench also reports the successful testing of every 128 test vectors and concludes when each test vector has been computed and verified.

//Modules Constructed

//Mux\_2\_1.sv

//Full\_Adder.sv

//Carry\_Sel\_Adder\_Mid.v

//Carry\_Sel\_Adder.v

//tb\_Carry\_Sel\_Adder.v

The testbench and behavioral modules must then be compiled using the Verilog Compile Simulator tool (VCS). If compiled with no warnings or errors, the behavioral simulation will be run and the output recorded. Figure 6.x.a and 6.x.b shows the captured behavioral waveforms of the modules when no error condition is forced. For the sake of visual representation, each of these waveform figures contain only the initial and terminating 12 test vectors respectively. Other test vectors are verified in the simulation log output. Figure 6.X shows the captured behavioral waveforms of the modules when an error condition is forced. This waveform figure only displays the 12 preceding vectors as well as the error test vector.

1. **MODULE FILES & SIMULATION RESULTS**

*Module 6.1*— Mux\_2\_1.sv

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* \*\*\*

\*\*\* EE 526 L Experiment #6 Kyle E. Keislar, Spring, 2020 \*\*\*

\*\*\* \*\*\*

\*\*\* Carry Select Adder \*\*\*

\*\*\* \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Filename: Mux\_2\_1.sv \*\*\*

\*\*\* Author: Kyle E. Keislar \*\*\*

\*\*\* Date: 03/17/2020 \*\*\*

\*\*\* Version: 1.0 \*\*\*

\*\*\* Revised: \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Module Description: 2 input multiplexer \*\*\*

\*\*\* \*\*\*

\*\*\* \*\*\*

\*\*\* \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

`timescale 1ns **/** 10ps

// Module Declaration

**module** Mux\_2\_1**(** X**,** //First data input of the Multiplexer;

Y**,** //Second data input of the Multiplexer;

SEL**,** //Multiplexer Select Line;

OUT**);** //Multiplexer output bus;

//Module Parameters

// I/O port assignment

**output** **reg** OUT**;**

**input** **wire** X**,**Y**;**

**input** **wire** SEL**;**

**specify**

**(**X **=>** OUT**)=** **(**2**);**

**(**Y **=>** OUT**)=** **(**2**);**

**(**SEL **=>** OUT**)** **=** **(**3**);**

**endspecify**

//Mux Logic

**always\_comb** **begin**

**if(**SEL**&&**1'B1**)** OUT**=**X**;** //if select is 1 choose Y

**else** **if(!(**SEL**||**1'B0**))** OUT**=**Y**;** //if it is 0 select X

**end**

**endmodule**

*Module 6.2*— Full\_Adder.sv

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* \*\*\*

\*\*\* EE 526 L Experiment #6 Kyle E. Keislar, Spring, 2020 \*\*\*

\*\*\* \*\*\*

\*\*\* Carry Select Adder \*\*\*

\*\*\* \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Filename: Full\_Adder.sv \*\*\*

\*\*\* Author: Kyle E. Keislar \*\*\*

\*\*\* Date: 03/18/2020 \*\*\*

\*\*\* Version: 1.0 \*\*\*

\*\*\* Revised: \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Module Description: single bit Full\_Adder with sum, carry outputs \*\*\*

\*\*\* \*\*\*

\*\*\* \*\*\*

\*\*\* \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

`timescale 1ns **/** 10ps

// Module Declaration

**module** Full\_Adder**(** A**,** //First operand bit

B**,** //Second operand bit

CIN**,** //carry in

SUM**,** //SUM

COUT**);** //cARRY OUT;

//Module Parameters

// I/O port assignment

**output** **reg** SUM**,** COUT**;**

**input** **wire** A**,**B**,**CIN**;**

**specify**

**(**A **=>** SUM**)=** **(**6**);**

**(**B **=>** SUM**)=** **(**6**);**

**(**A **=>** COUT**)=** **(**2**);**

**(**B **=>** COUT**)=** **(**2**);**

**(**CIN **=>** COUT**)=** **(**2**);**

**(**CIN **=>** SUM**)** **=** **(**3**);**

**endspecify**

**always\_comb** **begin**

//full adder logic

COUT**=** A**&**B **|** A**&**CIN **|** B**&**CIN**;**

SUM **=** A**^**B**^**CIN**;**

**end**

**endmodule**

*Module 6.3*— Carry\_Sel\_Adder\_Mid.v

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* \*\*\*

\*\*\* EE 526 L Experiment #6 Kyle E. Keislar, Spring, 2020 \*\*\*

\*\*\* \*\*\*

\*\*\* Carry Select Adder \*\*\*

\*\*\* \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Filename: Carry\_Sel\_Adder\_Mid.v \*\*\*

\*\*\* Author: Kyle E. Keislar \*\*\*

\*\*\* Date: 03/20/2020 \*\*\*

\*\*\* Version: 1.0 \*\*\*

\*\*\* Revised: \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Module Description: Midlevel Module for CSA. Instantiates two adder \*\*\*

\*\*\* pairs, two muxes. Takes Two input bits and a \*\*\*

\*\*\* carry to output a two bit sum and a 1 bit \*\*\*

\*\*\* carry out \*\*\*

\*\*\* \*\*\*

\*\*\* \*\*\*

\*\*\* \*\*\*

\*\*\* Sub Modules: Full\_Adder.sv, Mux\_2\_1.sv \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

`timescale 1ns **/** 10ps

// Module Declaration

**module** Carry\_Sel\_Adder\_Mid**(** A**,** //First operand bit pair

B**,** //Second operand bit pair

SUM**,** //SUM bits out

CIN**,**

COUT1**,** //First Carry out;

COUT2**);** //Second Carry out;

//Module Parameters

// I/O port assignment

**output** **reg** **[**1**:**0**]** SUM**;**

**output** **reg** COUT1**,** COUT2**;**

**input** **wire** **[**1**:**0**]** A**,**B**;**

**input** **wire** CIN**;**

//Internal Signals

**supply1** HIGH**;** //first adder carry input; set to logic 1

**supply0** LOW**;** //second adder carry input; set to logic 0

**wire** s1**,** s2**;** //first pair of internal wires driving multiplexer inputs

**wire** s3**,** s4**;** //first pair of internal wires driving multiplexer inputs

**wire** cin3**,**cin4**;**//Wires into cin3 and cin 4

//Module Instantiations

//Full adders

Full\_Adder FA1**(**A**[**0**],**B**[**0**],**HIGH**,**s1**,**cin3**);**

Full\_Adder FA2**(**A**[**0**],**B**[**0**],**LOW**,**s2**,**cin4**);**

Full\_Adder FA3**(**A**[**1**],**B**[**1**],**cin3**,**s3**,**COUT1**);**

Full\_Adder FA4**(**A**[**1**],**B**[**1**],**cin4**,**s4**,**COUT2**);**

//Multiplexers

Mux\_2\_1 Mux1**(**s1**,**s2**,**CIN**,**SUM**[**0**]);**

Mux\_2\_1 Mux2**(**s3**,**s4**,**CIN**,**SUM**[**1**]);**

**Endmodule**

*Module 6.4*— Carry\_Sel\_Adder.v

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* \*\*\*

\*\*\* EE 526 L Experiment #6 Kyle E. Keislar, Spring, 2020 \*\*\*

\*\*\* \*\*\*

\*\*\* Carry Select Adder \*\*\*

\*\*\* \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Filename: Carry\_Sel\_Adder\_Mid.v \*\*\*

\*\*\* Author: Kyle E. Keislar \*\*\*

\*\*\* Date: 03/20/2020 \*\*\*

\*\*\* Version: 1.0 \*\*\*

\*\*\* Revised: \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Module Description: Top level Module for CSA. Takes two 6 bit \*\*\*

\*\*\* inputs and an input carry and outputs the \*\*\*

\*\*\* sum and an additoinal carry \*\*\*

\*\*\* \*\*\*

\*\*\* \*\*\*

\*\*\* \*\*\*

\*\*\* \*\*\*

\*\*\* Sub Modules: Full\_Adder.sv, Mux\_2\_1.sv, Carry\_Sel\_Adder\_Mid.v \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

`timescale 1ns **/** 10ps

// Module Declaration

**module** Carry\_Sel\_Adder**(** c6**,** //Output Carry

sum**,** //Output sum of A and B

A**,** //Second Operand

B**,** //First Operand

c0**);** //carry in

//Module Parameters

// I/O port assignment

**output** **reg** **[**5**:**0**]** sum**;**

**output** **reg** c6**;**

**input** **wire** **[**5**:**0**]** A**,**B**;**

**input** **wire** c0**;**

//Internal Signals

**wire** c1**,** c2**,** c4**;** //CSA internal carry signals

**wire** cout1**,**cout2**,**cout3**,**cout4**;** //carry out signals from mid level module

//Module Instantiations

//Full adders

Full\_Adder FA1**(**A**[**0**],**B**[**0**],**c0**,**sum**[**0**],**c1**);**

Full\_Adder FA2**(**A**[**1**],**B**[**1**],**c1**,**sum**[**1**],**c2**);**

//Multiplexers

Mux\_2\_1 Mux\_c4**(**cout1**,**cout2**,**c2**,**c4**);**

Mux\_2\_1 Mux\_c6**(**cout3**,**cout4**,**c4**,**c6**);**

//Mid level Modules

Carry\_Sel\_Adder\_Mid CSAM\_1**(**A**[**3**:**2**],**B**[**3**:**2**],**sum**[**3**:**2**],**c2**,**cout1**,**cout2**);**

Carry\_Sel\_Adder\_Mid CSAM\_2**(**A**[**5**:**4**],**B**[**5**:**4**],**sum**[**5**:**4**],**c4**,**cout3**,**cout4**);**

**Endmodule**

*Module 6.5*— tb\_Carry\_Sel\_Adder.v

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* \*\*\*

\*\*\* EE 526 L Experiment #6 Kyle E. Keislar, Spring, 2020 \*\*\*

\*\*\* \*\*\*

\*\*\* Carry Select Adder \*\*\*

\*\*\* \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Filename: Carry\_Sel\_Adder\_Mid.v \*\*\*

\*\*\* Author: Kyle E. Keislar \*\*\*

\*\*\* Date: 03/20/2020 \*\*\*

\*\*\* Version: 1.0 \*\*\*

\*\*\* Revised: \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Test Modules: Mux\_2\_1.sv, Full\_Adder.sv, Carry\_Sel\_Adder\_Mid.v, \*\*\*

\*\*\* & Hierarchy : Carry\_Sel\_Adder.v \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Test Strategy: Two approaches are taken to test the Carry Select

Adder: First a small set of test vectors are applied

Then An exhaustave test is performed for all bits of

inputs A, B and c0;

For the Non-Exhaustive approach, a subset of 10 test

vectors was selected. Test vectors 1,2, and 6 handle

cases where one input is non-zero and the other inputs

are 0. Test vector 3 handles the case where both A and

B inputs are non-zero but carry in is zero. Test Vectors

4,5,7 & 8 handle cases where the output carry is set

due to binary addition overflow when the input carry is 0.

Test Vector 9 handles the case where all inputs are non-

zero but there is no overflow. Test Vector 10 handles the

case where all inputs are non-zero and the output carry

is set.

In the Exhaustive approach Each combination of inputs is

explored using loops. The testbench displays a detailed

subset of these exhaustive vectors while confirming the

test in comparision to a behaviorally coded adder (also

included in the test bench).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Timescale

`timescale 1ns **/** 10ps

`define runtime 500000**;**

`define csa\_del 10**;**

//`define ERROR\_COND 1; //condition to force error during exhaustive testing

// Module Instantiation

**module** tb\_Carry\_Sel\_Adder**();**

//test bench

**integer** tv\_count**=**0**;** //integer variable to count test vectors during exhaustive test

**integer** tst\_sum**;** //behaviroal adder sum

**integer** tst\_c\_out**;** //test carry out

**integer** err\_flag **=**0**;** //error flag, set if error conditions are defined

//Input & Output Defintions

//reg inputs

**reg** **[**5**:**0**]** a**,**b**;** //a,b input bus

**reg** c0**;** //input carry

//wire outputs

**wire** **[**5**:**0**]** sum**;**//output sum

**wire** c6**;** //output carry

//Carry Select Adder Instantation

Carry\_Sel\_Adder UUT**(**c6**,**sum**,**a**,**b**,**c0**);**

//Initial Conditions

**initial** **begin**

$vcdpluson**;** //include waveforms in simulation

a**=**6'b0**;** b**=**6'b0**;** c0**=**1'b0**;**

$display**(**""**);**

**#**`csa\_del $monitor**(**"\n\t\t c6 = %d\t Decimal Sum = %d\tBehav Add Carry = %d\tBehav Add Decimal Sum = %d \n"**,**c6**,{**c6**,**sum**},**tst\_c\_out**,**tst\_sum**);**

$monitoroff**;**

**#**`csa\_del

//Non-Exhaustive Testing

//Tv 1

$display**(**"Test Vector # 0.1"**);**

//0+1+0

**#**`csa\_del a**=**6'b0**;** b**=**6'b000001**;** c0**=**1'b0**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

//Tv 2

$display**(**"Test Vector # 0.2"**);**

//1+0+0

**#**`csa\_del a**=**6'b000001**;** b**=**6'b0**;** c0**=**1'b0**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

//Tv 3

$display**(**"Test Vector # 0.3"**);**

//1+1+0

**#**`csa\_del a**=**6'b000001**;** b**=**6'b000001**;** c0**=**1'b0**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

//Tv 4

$display**(**"Test Vector # 0.4"**);**

//32+32+0

**#**`csa\_del a**=**6'b100000**;** b**=**6'b100000**;** c0**=**1'b0**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

//Tv 5

$display**(**"Test Vector # 0.5"**);**

//52+21+0

**#**`csa\_del a**=**6'b110010**;** b**=**6'b010101**;** c0**=**1'b0**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

//Tv 6

$display**(**"Test Vector # 0.6"**);**

//0+63+0

**#**`csa\_del a**=**6'b000000**;** b**=**6'b111111**;** c0**=**1'b0**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

//Tv 7

$display**(**"Test Vector # 0.7"**);**

//1+63+0

**#**`csa\_del a**=**6'b000001**;** b**=**6'b111111**;** c0**=**1'b0**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

//Tv 8

$display**(**"Test Vector # 0.8"**);**

//63+63+0

**#**`csa\_del a**=**6'b111111**;** b**=**6'b111111**;** c0**=**1'b0**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

//Tv 9

$display**(**"Test Vector # 0.9"**);**

//19+4+1

**#**`csa\_del a**=**6'b010011**;** b**=**6'b000100**;** c0**=**1'b1**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

//Tv 10

$display**(**"Test Vector # 0.10"**);**

//63+0+1

**#**`csa\_del a**=**6'b111111**;** b**=**6'b000010**;** c0**=**1'b1**;**

**#**`csa\_del $display**(**"\n\n\ta=%d\tb=%d\tcarry\_in=%d\n\t\t\tsum=%d\tcarry\_out=%d\toutput=%d"**,**a**,**b**,**c0**,**sum**,**c6**,{**c6**,**sum**});**

$display**(**""**);**

//Exhaustive Testing

**for(int** J**=**0**;** J**<**64**;**J**=**J**+**1**)** **begin**

$display**(**"Testing Vectors: %4d\t-\t%4d\t"**,(**tv\_count**)\***128**+**1**,(**tv\_count**+**1**)\***128**);**

tv\_count**=**tv\_count**+**1**;**//simple counter to enumerate test vectors

**for(int** I**=**0**;** I**<**128**;** I **=** I**+**1**)** **begin**

$write**(**"%c[1;37m"**,**27**);**

a**=**J**;** b**=**I**/**2**;** c0**=**I**%**2**;** //test each case with & without carry in

**#**`csa\_del

//adder logic

//Error Condition

`ifdef ERROR\_COND

**force** err\_flag**=**1**;**//set error flag to 1 to show preceeding TVs before error

**if(**J**==**48 **&** I**>**40**)** **force** tst\_sum**=-**1**;**

**else** tst\_sum**=**a**+**b**+**c0**;**

`else tst\_sum**=**a**+**b**+**c0**;**

`endif

**if((**a**+**b**+**c0**)>=**64**)** tst\_c\_out**=**1**;**

**else** tst\_c\_out**=**0**;**

//display Test results for current TV

**if((**J**==**0 **&** I**<**12**)** **|** **(**J**==**63 **&** I**>**115**)|(**J**==**48 **&** I**>**28 **&** err\_flag**==**1**))**

**begin**

$display**(**"\nTV#: %2d"**,(**tv\_count**-**1**)\***128**+**I**+**1**);**

$display**(**"\n\ta=%d\tb=%d\tcarry\_in=%d"**,**a**,**b**,**c0**);**

**end**

**if((**J**==**0 **&** I**<**12**)** **|** **(**J**==**63 **&** I**>**116**)|(**J**==**48 **&** I**>**28 **&** err\_flag**==**1**))** $monitoron**;** //first and last test vectors to be displayed

**#**`csa\_del**;**

**if((**J**==**0 **&** I**<**12**)** **|** **(**J**==**63 **&** I**>**116**)|(**J**==**48 **&** I**>**28 **&** err\_flag**==**1**))** $monitoroff**;**

**if(({**c6**,**sum**}** **!=** tst\_sum**)** **|** **(**c6 **!=** tst\_c\_out**)** **)** **begin** //if the ader's sums are not equal or c\_out is not equal

//error

$write**(**"%c[1;31m"**,**27**);**

$display**(**"\n\t\t::|| ERROR || ::\t\t CSA output: %d != Behav Add output: %d \n"**,{**c6**,**sum**},**tst\_sum**);**

$display**(**"\n\t\t::|| ERROR || ::\t\t c6 = %d\t sum = %d\ttst\_sum = %d\ttst\_carry = %d \n"**,**c6**,**sum**,**tst\_sum**,**tst\_c\_out**);**

$write**(**"%c[1;37m"**,**27**);**

$finish**;**//stop testing after the first error

**end**

//if test confirmed

**if((**I**==**127**)** **&** **(**J**==**63**))begin**

$write**(**"%c[1;32m"**,**27**);**

$write**(**"|Exhaustive Test Compelte: All TVs Confirmed|\n"**);**

$write**(**"%c[1;37m"**,**27**);**

**end;**

**if(**I**==**127**)** $write**(**"DONE\n"**);**

**end**

**end**

**end**

//Asynchronous Behavior

//Synchronous Behavior

//Check starting values & Finish the simulation at runtime

**always**

**begin**

**#**`runtime

$display**(**""**);**//new line

$finish**;**

**end**

**endmodule**

Figure 6.x.a Simulation Behavioral Waveforms for Error-Free Case, Initial Vectors

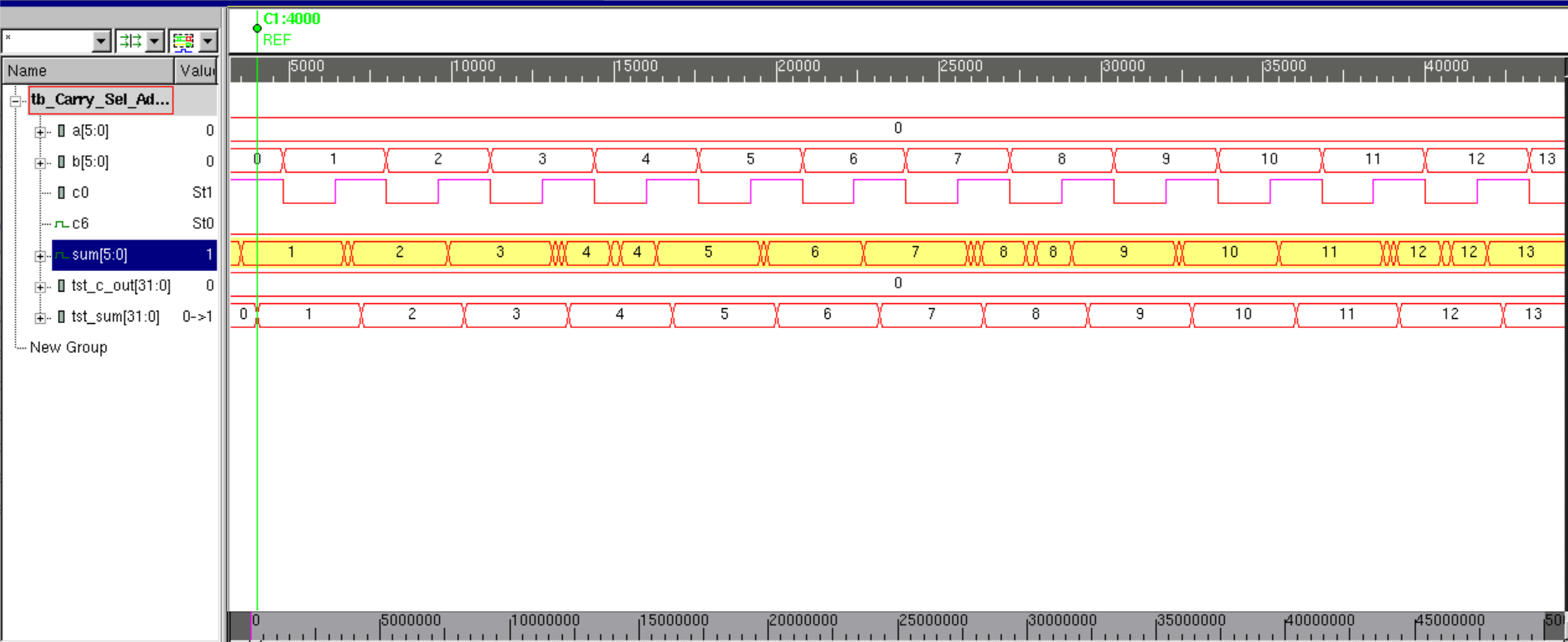


Figure 6.x.b Simulation Behavioral Waveforms for Error-Free Case, Ending Vectors

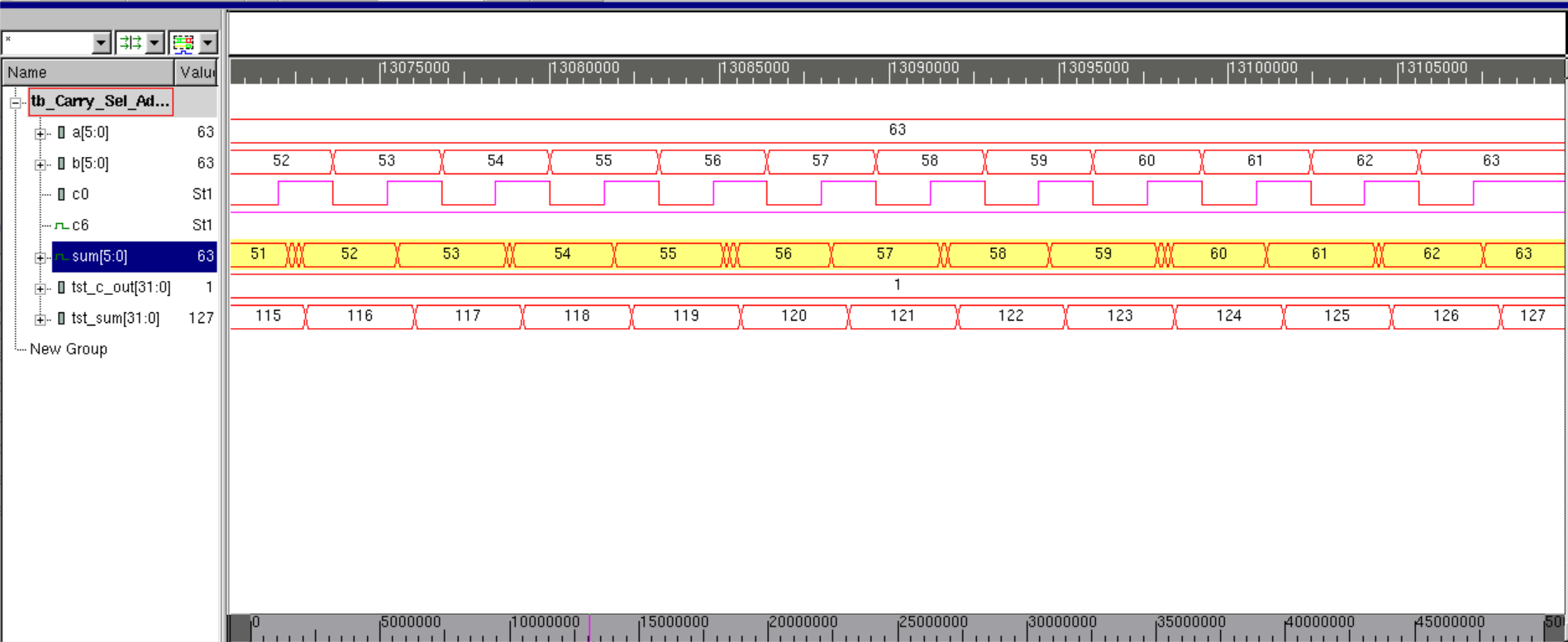
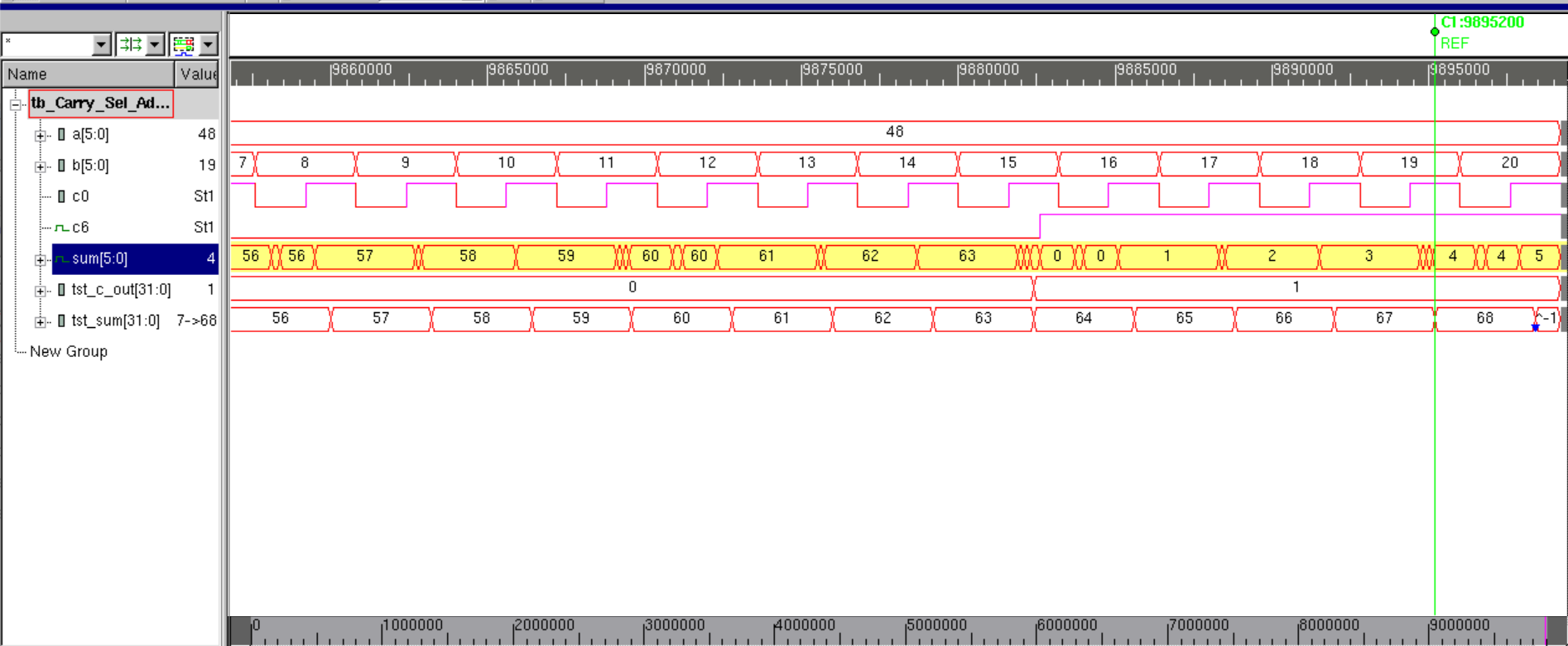


Figure 6.X Simulation Behavioral Waveforms for Error Case

****

1. **ANALYSIS**

//Did it function as Adder

//CSA behavior

//Non-Exhaustive Analysis

//Sum when all inputs are 0

//Sum when all inputs are non-zero

//Sum when carry in is 1

//Sum when carry in is 0

//Sum when overflow

//Exhaustive Analysis

//Error free Case

//Error forced case

1. **CONCLUSION**

//CSA

**APPENDIX**

1. **SIMULATION LOG A: No Error Condition**

Chronologic VCS simulator copyright 1991-2017

Contains Synopsys proprietary information.

Compiler version N-2017.12-SP2-2\_Full64; Runtime version N-2017.12-SP2-2\_Full64; Apr 5 23:04 2020

VCD+ Writer N-2017.12-SP2-2\_Full64 Copyright (c) 1991-2017 by Synopsys Inc.

Testing Vectors: 1 - 128

TV#: 1

a= 0 b= 0 carry\_in=0

c6 = 0 Decimal Sum = 0 Behav Add Carry = 0 Behav Add Decimal Sum = 0

TV#: 2

a= 0 b= 0 carry\_in=1

c6 = 0 Decimal Sum = 1 Behav Add Carry = 0 Behav Add Decimal Sum = 1

TV#: 3

a= 0 b= 1 carry\_in=0

c6 = 0 Decimal Sum = 1 Behav Add Carry = 0 Behav Add Decimal Sum = 1

TV#: 4

a= 0 b= 1 carry\_in=1

c6 = 0 Decimal Sum = 2 Behav Add Carry = 0 Behav Add Decimal Sum = 2

TV#: 5

a= 0 b= 2 carry\_in=0

c6 = 0 Decimal Sum = 2 Behav Add Carry = 0 Behav Add Decimal Sum = 2

TV#: 6

a= 0 b= 2 carry\_in=1

c6 = 0 Decimal Sum = 3 Behav Add Carry = 0 Behav Add Decimal Sum = 3

TV#: 7

a= 0 b= 3 carry\_in=0

c6 = 0 Decimal Sum = 3 Behav Add Carry = 0 Behav Add Decimal Sum = 3

TV#: 8

a= 0 b= 3 carry\_in=1

c6 = 0 Decimal Sum = 4 Behav Add Carry = 0 Behav Add Decimal Sum = 4

TV#: 9

a= 0 b= 4 carry\_in=0

c6 = 0 Decimal Sum = 4 Behav Add Carry = 0 Behav Add Decimal Sum = 4

TV#: 10

a= 0 b= 4 carry\_in=1

c6 = 0 Decimal Sum = 5 Behav Add Carry = 0 Behav Add Decimal Sum = 5

TV#: 11

a= 0 b= 5 carry\_in=0

c6 = 0 Decimal Sum = 5 Behav Add Carry = 0 Behav Add Decimal Sum = 5

TV#: 12

a= 0 b= 5 carry\_in=1

c6 = 0 Decimal Sum = 6 Behav Add Carry = 0 Behav Add Decimal Sum = 6

DONE

Testing Vectors: 129 - 256

DONE

Testing Vectors: 257 - 384

DONE

Testing Vectors: 385 - 512

DONE

Testing Vectors: 513 - 640

DONE

Testing Vectors: 641 - 768

DONE

Testing Vectors: 769 - 896

DONE

Testing Vectors: 897 - 1024

DONE

Testing Vectors: 1025 - 1152

DONE

Testing Vectors: 1153 - 1280

DONE

Testing Vectors: 1281 - 1408

DONE

Testing Vectors: 1409 - 1536

DONE

Testing Vectors: 1537 - 1664

DONE

Testing Vectors: 1665 - 1792

DONE

Testing Vectors: 1793 - 1920

DONE

Testing Vectors: 1921 - 2048

DONE

Testing Vectors: 2049 - 2176

DONE

Testing Vectors: 2177 - 2304

DONE

Testing Vectors: 2305 - 2432

DONE

Testing Vectors: 2433 - 2560

DONE

Testing Vectors: 2561 - 2688

DONE

Testing Vectors: 2689 - 2816

DONE

Testing Vectors: 2817 - 2944

DONE

Testing Vectors: 2945 - 3072

DONE

Testing Vectors: 3073 - 3200

DONE

Testing Vectors: 3201 - 3328

DONE

Testing Vectors: 3329 - 3456

DONE

Testing Vectors: 3457 - 3584

DONE

Testing Vectors: 3585 - 3712

DONE

Testing Vectors: 3713 - 3840

DONE

Testing Vectors: 3841 - 3968

DONE

Testing Vectors: 3969 - 4096

DONE

Testing Vectors: 4097 - 4224

DONE

Testing Vectors: 4225 - 4352

DONE

Testing Vectors: 4353 - 4480

DONE

Testing Vectors: 4481 - 4608

DONE

Testing Vectors: 4609 - 4736

DONE

Testing Vectors: 4737 - 4864

DONE

Testing Vectors: 4865 - 4992

DONE

Testing Vectors: 4993 - 5120

DONE

Testing Vectors: 5121 - 5248

DONE

Testing Vectors: 5249 - 5376

DONE

Testing Vectors: 5377 - 5504

DONE

Testing Vectors: 5505 - 5632

DONE

Testing Vectors: 5633 - 5760

DONE

Testing Vectors: 5761 - 5888

DONE

Testing Vectors: 5889 - 6016

DONE

Testing Vectors: 6017 - 6144

DONE

Testing Vectors: 6145 - 6272

DONE

Testing Vectors: 6273 - 6400

DONE

Testing Vectors: 6401 - 6528

DONE

Testing Vectors: 6529 - 6656

DONE

Testing Vectors: 6657 - 6784

DONE

Testing Vectors: 6785 - 6912

DONE

Testing Vectors: 6913 - 7040

DONE

Testing Vectors: 7041 - 7168

DONE

Testing Vectors: 7169 - 7296

DONE

Testing Vectors: 7297 - 7424

DONE

Testing Vectors: 7425 - 7552

DONE

Testing Vectors: 7553 - 7680

DONE

Testing Vectors: 7681 - 7808

DONE

Testing Vectors: 7809 - 7936

DONE

Testing Vectors: 7937 - 8064

DONE

Testing Vectors: 8065 - 8192

TV#: 8181

a=63 b=58 carry\_in=0

TV#: 8182

a=63 b=58 carry\_in=1

c6 = 1 Decimal Sum = 122 Behav Add Carry = 1 Behav Add Decimal Sum = 122

TV#: 8183

a=63 b=59 carry\_in=0

c6 = 1 Decimal Sum = 122 Behav Add Carry = 1 Behav Add Decimal Sum = 122

TV#: 8184

a=63 b=59 carry\_in=1

c6 = 1 Decimal Sum = 123 Behav Add Carry = 1 Behav Add Decimal Sum = 123

TV#: 8185

a=63 b=60 carry\_in=0

c6 = 1 Decimal Sum = 123 Behav Add Carry = 1 Behav Add Decimal Sum = 123

TV#: 8186

a=63 b=60 carry\_in=1

c6 = 1 Decimal Sum = 124 Behav Add Carry = 1 Behav Add Decimal Sum = 124

TV#: 8187

a=63 b=61 carry\_in=0

c6 = 1 Decimal Sum = 124 Behav Add Carry = 1 Behav Add Decimal Sum = 124

TV#: 8188

a=63 b=61 carry\_in=1

c6 = 1 Decimal Sum = 125 Behav Add Carry = 1 Behav Add Decimal Sum = 125

TV#: 8189

a=63 b=62 carry\_in=0

c6 = 1 Decimal Sum = 125 Behav Add Carry = 1 Behav Add Decimal Sum = 125

TV#: 8190

a=63 b=62 carry\_in=1

c6 = 1 Decimal Sum = 126 Behav Add Carry = 1 Behav Add Decimal Sum = 126

TV#: 8191

a=63 b=63 carry\_in=0

c6 = 1 Decimal Sum = 126 Behav Add Carry = 1 Behav Add Decimal Sum = 126

TV#: 8192

a=63 b=63 carry\_in=1

c6 = 1 Decimal Sum = 127 Behav Add Carry = 1 Behav Add Decimal Sum = 127

|Exhaustive Test Compelte: All TVs Confirmed|

DONE

$finish called from file "tb\_Carry\_Sel\_Adder.v", line 209.

$finish at simulation time 50000000

V C S S i m u l a t i o n R e p o r t

Time: 500000000 ps

CPU Time: 0.280 seconds; Data structure size: 0.0Mb

Sun Apr 5 23:04:16 2020

1. **SIMULATION LOG B: Error Condition**

Chronologic VCS simulator copyright 1991-2017

Contains Synopsys proprietary information.

Compiler version N-2017.12-SP2-2\_Full64; Runtime version N-2017.12-SP2-2\_Full64; Apr 5 23:02 2020

VCD+ Writer N-2017.12-SP2-2\_Full64 Copyright (c) 1991-2017 by Synopsys Inc.

Testing Vectors: 1 - 128

TV#: 1

a= 0 b= 0 carry\_in=0

c6 = 0 Decimal Sum = 0 Behav Add Carry = 0 Behav Add Decimal Sum = 0

TV#: 2

a= 0 b= 0 carry\_in=1

c6 = 0 Decimal Sum = 1 Behav Add Carry = 0 Behav Add Decimal Sum = 1

TV#: 3

a= 0 b= 1 carry\_in=0

c6 = 0 Decimal Sum = 1 Behav Add Carry = 0 Behav Add Decimal Sum = 1

TV#: 4

a= 0 b= 1 carry\_in=1

c6 = 0 Decimal Sum = 2 Behav Add Carry = 0 Behav Add Decimal Sum = 2

TV#: 5

a= 0 b= 2 carry\_in=0

c6 = 0 Decimal Sum = 2 Behav Add Carry = 0 Behav Add Decimal Sum = 2

TV#: 6

a= 0 b= 2 carry\_in=1

c6 = 0 Decimal Sum = 3 Behav Add Carry = 0 Behav Add Decimal Sum = 3

TV#: 7

a= 0 b= 3 carry\_in=0

c6 = 0 Decimal Sum = 3 Behav Add Carry = 0 Behav Add Decimal Sum = 3

TV#: 8

a= 0 b= 3 carry\_in=1

c6 = 0 Decimal Sum = 4 Behav Add Carry = 0 Behav Add Decimal Sum = 4

TV#: 9

a= 0 b= 4 carry\_in=0

c6 = 0 Decimal Sum = 4 Behav Add Carry = 0 Behav Add Decimal Sum = 4

TV#: 10

a= 0 b= 4 carry\_in=1

c6 = 0 Decimal Sum = 5 Behav Add Carry = 0 Behav Add Decimal Sum = 5

TV#: 11

a= 0 b= 5 carry\_in=0

c6 = 0 Decimal Sum = 5 Behav Add Carry = 0 Behav Add Decimal Sum = 5

TV#: 12

a= 0 b= 5 carry\_in=1

c6 = 0 Decimal Sum = 6 Behav Add Carry = 0 Behav Add Decimal Sum = 6

DONE

Testing Vectors: 129 - 256

DONE

Testing Vectors: 257 - 384

DONE

Testing Vectors: 385 - 512

DONE

Testing Vectors: 513 - 640

DONE

Testing Vectors: 641 - 768

DONE

Testing Vectors: 769 - 896

DONE

Testing Vectors: 897 - 1024

DONE

Testing Vectors: 1025 - 1152

DONE

Testing Vectors: 1153 - 1280

DONE

Testing Vectors: 1281 - 1408

DONE

Testing Vectors: 1409 - 1536

DONE

Testing Vectors: 1537 - 1664

DONE

Testing Vectors: 1665 - 1792

DONE

Testing Vectors: 1793 - 1920

DONE

Testing Vectors: 1921 - 2048

DONE

Testing Vectors: 2049 - 2176

DONE

Testing Vectors: 2177 - 2304

DONE

Testing Vectors: 2305 - 2432

DONE

Testing Vectors: 2433 - 2560

DONE

Testing Vectors: 2561 - 2688

DONE

Testing Vectors: 2689 - 2816

DONE

Testing Vectors: 2817 - 2944

DONE

Testing Vectors: 2945 - 3072

DONE

Testing Vectors: 3073 - 3200

DONE

Testing Vectors: 3201 - 3328

DONE

Testing Vectors: 3329 - 3456

DONE

Testing Vectors: 3457 - 3584

DONE

Testing Vectors: 3585 - 3712

DONE

Testing Vectors: 3713 - 3840

DONE

Testing Vectors: 3841 - 3968

DONE

Testing Vectors: 3969 - 4096

DONE

Testing Vectors: 4097 - 4224

DONE

Testing Vectors: 4225 - 4352

DONE

Testing Vectors: 4353 - 4480

DONE

Testing Vectors: 4481 - 4608

DONE

Testing Vectors: 4609 - 4736

DONE

Testing Vectors: 4737 - 4864

DONE

Testing Vectors: 4865 - 4992

DONE

Testing Vectors: 4993 - 5120

DONE

Testing Vectors: 5121 - 5248

DONE

Testing Vectors: 5249 - 5376

DONE

Testing Vectors: 5377 - 5504

DONE

Testing Vectors: 5505 - 5632

DONE

Testing Vectors: 5633 - 5760

DONE

Testing Vectors: 5761 - 5888

DONE

Testing Vectors: 5889 - 6016

DONE

Testing Vectors: 6017 - 6144

DONE

Testing Vectors: 6145 - 6272

TV#: 6174

a=48 b=14 carry\_in=1

c6 = 0 Decimal Sum = 63 Behav Add Carry = 0 Behav Add Decimal Sum = 63

TV#: 6175

a=48 b=15 carry\_in=0

c6 = 0 Decimal Sum = 63 Behav Add Carry = 0 Behav Add Decimal Sum = 63

TV#: 6176

a=48 b=15 carry\_in=1

c6 = 0 Decimal Sum = 48 Behav Add Carry = 1 Behav Add Decimal Sum = 64

c6 = 1 Decimal Sum = 64 Behav Add Carry = 1 Behav Add Decimal Sum = 64

TV#: 6177

a=48 b=16 carry\_in=0

c6 = 1 Decimal Sum = 64 Behav Add Carry = 1 Behav Add Decimal Sum = 64

TV#: 6178

a=48 b=16 carry\_in=1

c6 = 1 Decimal Sum = 65 Behav Add Carry = 1 Behav Add Decimal Sum = 65

TV#: 6179

a=48 b=17 carry\_in=0

c6 = 1 Decimal Sum = 65 Behav Add Carry = 1 Behav Add Decimal Sum = 65

TV#: 6180

a=48 b=17 carry\_in=1

c6 = 1 Decimal Sum = 66 Behav Add Carry = 1 Behav Add Decimal Sum = 66

TV#: 6181

a=48 b=18 carry\_in=0

c6 = 1 Decimal Sum = 66 Behav Add Carry = 1 Behav Add Decimal Sum = 66

TV#: 6182

a=48 b=18 carry\_in=1

c6 = 1 Decimal Sum = 67 Behav Add Carry = 1 Behav Add Decimal Sum = 67

TV#: 6183

a=48 b=19 carry\_in=0

c6 = 1 Decimal Sum = 67 Behav Add Carry = 1 Behav Add Decimal Sum = 67

TV#: 6184

a=48 b=19 carry\_in=1

c6 = 1 Decimal Sum = 68 Behav Add Carry = 1 Behav Add Decimal Sum = 68

TV#: 6185

a=48 b=20 carry\_in=0

c6 = 1 Decimal Sum = 68 Behav Add Carry = 1 Behav Add Decimal Sum = 68

TV#: 6186

a=48 b=20 carry\_in=1

c6 = 1 Decimal Sum = 69 Behav Add Carry = 1 Behav Add Decimal Sum = -1

::|| ERROR || :: CSA output: 69 != Behav Add output: -1

::|| ERROR || :: c6 = 1 sum = 5 tst\_sum = -1 tst\_carry = 1

$finish called from file "tb\_Carry\_Sel\_Adder.v", line 185.

$finish at simulation time 9899200

V C S S i m u l a t i o n R e p o r t

Time: 98992000 ps

CPU Time: 0.270 seconds; Data structure size: 0.0Mb

Sun Apr 5 23:02:13 2020

1. **SIMULATION LOG C: Non-Exhaustive**

Chronologic VCS simulator copyright 1991-2017

Contains Synopsys proprietary information.

Compiler version N-2017.12-SP2-2\_Full64; Runtime version N-2017.12-SP2-2\_Full64; Apr 5 23:06 2020

VCD+ Writer N-2017.12-SP2-2\_Full64 Copyright (c) 1991-2017 by Synopsys Inc.

Test Vector # 0.1

a= 0 b= 1 carry\_in=0

sum= 1 carry\_out=0 output= 1

Test Vector # 0.2

a= 1 b= 0 carry\_in=0

sum= 1 carry\_out=0 output= 1

Test Vector # 0.3

a= 1 b= 1 carry\_in=0

sum= 2 carry\_out=0 output= 2

Test Vector # 0.4

a=32 b=32 carry\_in=0

sum= 0 carry\_out=1 output= 64

Test Vector # 0.5

a=50 b=21 carry\_in=0

sum= 3 carry\_out=1 output= 67

Test Vector # 0.6

a= 0 b=63 carry\_in=0

sum=39 carry\_out=0 output= 39

Test Vector # 0.7

a= 1 b=63 carry\_in=0

sum=48 carry\_out=0 output= 48

Test Vector # 0.8

a=63 b=63 carry\_in=0

sum= 2 carry\_out=1 output= 66

Test Vector # 0.9

a=19 b= 4 carry\_in=1

sum=28 carry\_out=0 output= 28

Test Vector # 0.10

a=63 b= 2 carry\_in=1

sum=10 carry\_out=1 output= 74

$finish called from file "tb\_Carry\_Sel\_Adder.v", line 211.

$finish at simulation time 50000000

V C S S i m u l a t i o n R e p o r t

Time: 500000000 ps

CPU Time: 0.230 seconds; Data structure size: 0.0Mb

Sun Apr 5 23:06:21 2020