



Date:02-04-2023

## Hls ASSIGNMENT-5

### Part A:

Implement a DUT that accesses 8 elements from BRAM (the BRAM should be contained within the DUT, you can choose to populate the BRAM in any way you like) and gives out all 8 values as HLS streaming output in a single bundle in a single clock cycle. The starting index will be an input to the DUT and it will be a multiple of 8.

The way to access these 8 elements are as follows:

1. 8 consecutive elements are to be chosen and these elements will make up a bundle.
2. Every 8th element is to be chosen and 8 such elements will make up a bundle.

### Part B:

Implement a DUT that takes 4 inputs in 4 clock cycles and then saves all the inputs in a BRAM (The BRAM should be contained within the DUT) at a single address. At the same time, it also gives out all the 4 inputs as a single bundle output in the same clock cycle when it receives the final input. The BRAM index will overflow after the BRAM is full and it should overwrite the old values as more inputs keep coming in.

You can assume the data elements to be 8 bit values for both parts.

The purpose of this assignment is to understand the working of the pragmas, ARRAY\_PARTITION (block, cyclic, complete) and ARRAY\_RESHAPE (horizontal, vertical). Report all the source code files, self-checking testbench files, input and output files, and "ALL" the HLS reports.

## PART-A\_1

### Design:

Main function:

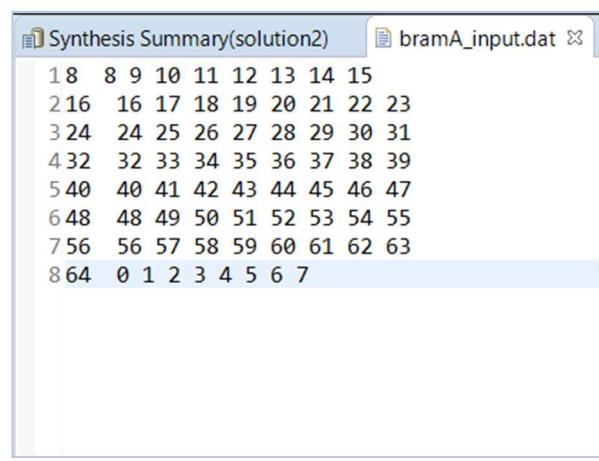


```
1 #include <hls_stream.h>
2 #include <iostream>
3 #include <fstream>
4 #include<stdio.h>
5 #define BRAM_DEPTH 64
6 #define BUNDLE_SIZE 8
7 #include "ap_int.h"
8 typedef ap_uint <8> size;
9 void dut(size start_index, hls::stream<size> & output) {
10     #pragma HLS INTERFACE ap_none port=start_index
11     #pragma HLS INTERFACE axis register port=output
12
13     size bram1[BRAM_DEPTH];
14     #pragma HLS BIND_STORAGE variable=bram1 type=ram_1p impl=bram
15     #pragma HLS ARRAY_PARTITION variable=bram1 cyclic factor=8
16
17     size i=0,j=0;
18     for ( i = 0; i < BRAM_DEPTH; i++) {
19         bram1[i]=i;
20     }
21     /*FILE *fp;
22     fp=fopen("populate.dat","r");
23     while(fscanf(fp,"%d",&bram1[i])==1)
24     {
25         i++;
26     }*/
27     size bundle1[8];
28     for (j= 0; j < 8; j++) {
29         #pragma HLS PIPELINE II=1
30         bundle1[j] = bram1[(start_index +j)%BRAM_DEPTH];
31         output.write(bundle1[j]);
32     }
33 }
```

## Self-checking Test Bench:

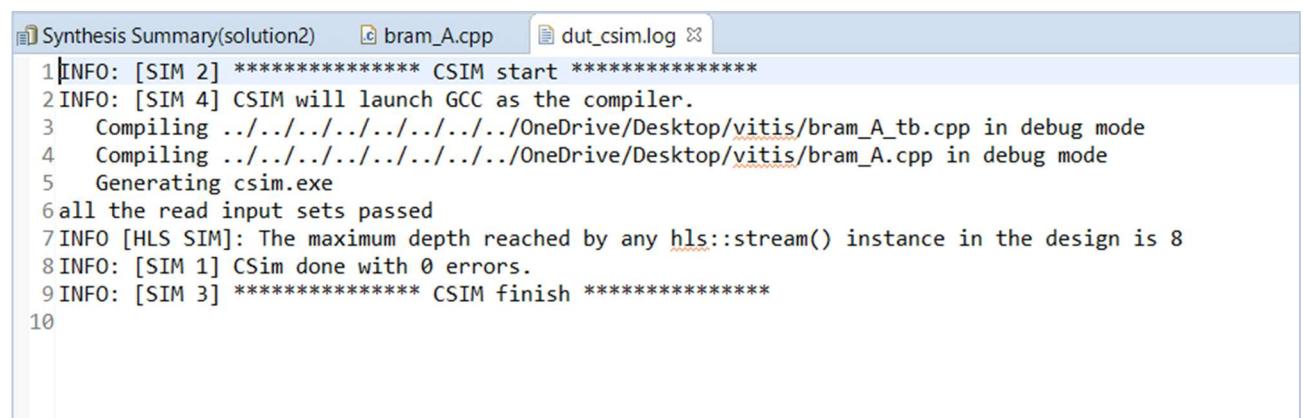
```
Synthesis Summary(solution2)  bramA_input.dat  bram_A.cpp  dut_csim.log  bram_A_tb.cpp ✘
1 #include "bram.h"
2 #include <hls_stream.h>
3 #include <iostream>
4 #define BRAM_DEPTH 64
5 #define BUNDLE_SIZE 8
6 #include <ap_int.h>
7 typedef ap_uint<8> size;
8 void dut(size start_index, hls::stream<size>& output);
9
10 int main()
11 {
12     hls::stream<size> output;
13     size out,start_index;
14     size golden_data[8][8];
15     size func_data[8][8];
16     size index[8];
17     int i=0,j=0,flag=0,count=0;
18     FILE *fp;
19
20     fp=fopen("bramA_input.dat","r");
21     if (fp == NULL) {
22         printf("Error opening file\n");
23         return 1;
24     }
25     while(fscanf(fp,"%d %d %d %d %d %d %d %d",&index[i],&golden_data[i][0],&golden_data[i][1],
26                 &golden_data[i][2],&golden_data[i][3],&golden_data[i][4],&golden_data[i][5],&golden_data[i][6],
27                 &golden_data[i][7]) == 9) {
28         i++;
29     }
30     fclose(fp);
31
32
33     for(i=0; i<8; i++){
34         start_index=index[i];
35         dut(start_index,output);
36         //output >> out;
37         //std::cout<<start_index<<std::endl;
38         for (size j=0 ; j < BUNDLE_SIZE; j++) {
39             func_data[i][j]=output.read();
40             //std::cout<< func_data[i][j]<<std::endl;
41         }
42     }
43     fp=fopen("output.dat","w");
44     for (int i = 0; i < 8; i++) {
45         for (int j = 0; j < BUNDLE_SIZE; j++) {
46             if(func_data[i][j]==golden_data[i][j]){
47                 count++;
48             }
49         }
50         if(count==8)
51         {
52             fprintf(fp,"%s\n","Test Passed");
53         }
54         else
55         {
56             fprintf(fp,"%s\n","Test Failed!");
57             flag=1;
58         }
59         count=0;
60     }
61     fclose(fp);
62
63     if(flag==0){
64         std::cout<<"all the read input sets passed"<<std::endl;
65     }
66     else{
67         std::cout<<"few read input sets were not passed"<<std::endl;
68     }
69     return 0;
70 }
71
72
```

## Input File:



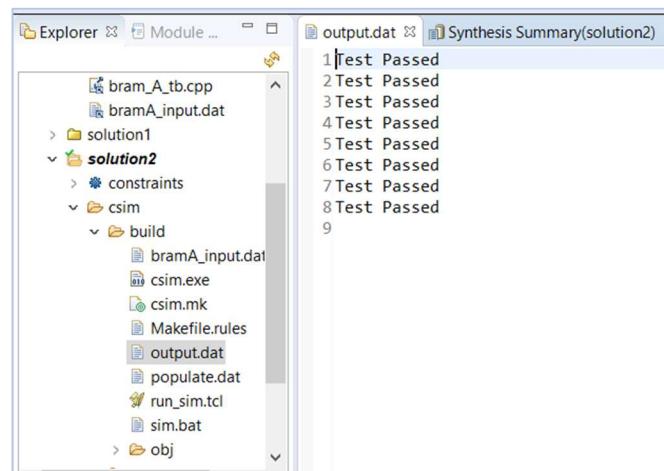
```
18  8  9  10 11 12 13 14 15  
216 16 17 18 19 20 21 22 23  
324 24 25 26 27 28 29 30 31  
432 32 33 34 35 36 37 38 39  
540 40 41 42 43 44 45 46 47  
648 48 49 50 51 52 53 54 55  
756 56 57 58 59 60 61 62 63  
864 0 1 2 3 4 5 6 7
```

## C simulation printed output:



```
1[INFO: [SIM 2] ***** CSIM start *****  
2INFO: [SIM 4] CSIM will launch GCC as the compiler.  
3 Compiling ../../../../../../OneDrive/Desktop/vitis/bram_A_tb.cpp in debug mode  
4 Compiling ../../../../../../OneDrive/Desktop/vitis/bram_A.cpp in debug mode  
5 Generating csim.exe  
6 all the read input sets passed  
7INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 8  
8INFO: [SIM 1] CSim done with 0 errors.  
9INFO: [SIM 3] ***** CSIM finish *****  
10
```

## Output File:



The Explorer view shows the following files and folders:

- Module ...
- solution1
- solution2** (selected)
- constraints
- csim
  - build
    - bramA\_input.dat
    - csim.exe
    - csim.mk
    - Makefile.rules
    - output.dat** (highlighted)
    - populate.dat
    - run\_sim.tcl
    - sim.bat
  - obj

The 'output.dat' file contains the following test results:

```
1Test Passed  
2Test Passed  
3Test Passed  
4Test Passed  
5Test Passed  
6Test Passed  
7Test Passed  
8Test Passed  
9
```

```

Synthesis Summary(solution2) bramA_input.dat bram_A.cpp bram_A_tb.cpp dut_csim.log
1 INFO: [SIM 2] **** CSIM start ****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3 Compiling ../../../../../OneDrive/Desktop/vitis/bram_A_tb.cpp in debug mode
4 Compiling ../../../../../OneDrive/Desktop/vitis/bram_A.cpp in debug mode
5 Generating csim.exe
6 few read input sets were not passed
7 INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 8
8 INFO: [SIM 1] CSim done with 0 errors.
9 INFO: [SIM 3] **** CSIM finish ****
10

```

	output.dat	Synthesis Summary(solution2)
1	Test Passed	
2	Test Passed	
3	Test Passed	
4	Test Passed	
5	Test Passed	
6	Test Failed!	
7	Test Passed	
8	Test Passed	
9		

```

Synthesis Summary(solution2) bramA_input.dat
18 8 9 10 11 12 13 14 15
216 16 17 18 19 20 21 22 23
324 24 25 26 27 28 29 30 31
432 32 33 34 35 36 37 38 39
540 40 41 42 43 44 45 46 47
648 48 49 50 51 52 53 54 58
756 56 57 58 59 60 61 62 63
864 0 1 2 3 4 5 6 7

```

## Synthesis Report:

Synthesis Summary Report of 'dut'

**General Information**

Date: Mon Apr 3 20:00:14 2023	Solution: solution2 (Vivado IP Flow Target)
Version: 2022.2 (Build 3670227 on Oct 13 2022)	Product family: zynq
Project: project_20	Target device: xc7z020-clg484-1

**Timing Estimate**

Target	Estimated	Uncertainty
10.00 ns	5.558 ns	2.70 ns

**Performance & Resource Estimates**

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
v dut	-	-	-	-	11	110.000	-	12	-	no	8	0	12	144	0
v VITIS_LOOP_28_2	-	-	-	-	9	90.000	3	1	8	yes	-	-	-	-	-

**Performance Pragma**

Modules & Loops	Target TI(cycles)	TI(cycles)	TI met
v dut	-	-	-
v VITIS_LOOP_28_2	-	-	-

**HW Interfaces**

- AXIS**

Interface	Register Mode	TDATA	TREADY	TVALID
output_r	both	8	1	1
- Other Ports**

Interface	Mode	Bitwidth
start_index	ap_none	8
- TOP LEVEL CONTROL**

Interface	Type	Ports
ap_clk	clock	ap_clk
ap_rst_n	reset	ap_rst_n
ap_ctrl	ap_ctrl_hs	ap_done ap_idle ap_ready ap_start

**SW I/O Information**

- Top Function Arguments**

Argument	Direction	Datatype
start_index	in	ap_uint<8>
output	out	stream<ap_uint<8>>&
- SW-to-HW Mapping**

Argument	HW Interface	HW Type
start_index	start_index	port
output	output_r	interface

**Pragma Report**

- Valid Pragma Syntax**

Type	Options	Location	Function
interface	ap_none port=start_index	./././OneDrive/Desktop/vitis/ <a href="#">bram_A.cpp:10</a>	dut
interface	axis register port=output	./././OneDrive/Desktop/vitis/ <a href="#">bram_A.cpp:11</a>	dut
bind_storage	variable=bram1 type=ram_1p impl=bram	./././OneDrive/Desktop/vitis/ <a href="#">bram_A.cpp:14</a>	dut
array_partition	variable=bram1 cyclic factor=8	./././OneDrive/Desktop/vitis/ <a href="#">bram_A.cpp:15</a>	dut
pipeline	II=1	./././OneDrive/Desktop/vitis/ <a href="#">bram_A.cpp:29</a>	dut

**Bind Op Report**

No filter settings

Name	DSP	Pragma	Variable	Op	Impl	Latency
✓ dut	-	-	-	-	-	-
>  VITIS_LOOP_28_2	-	-	-	-	-	-

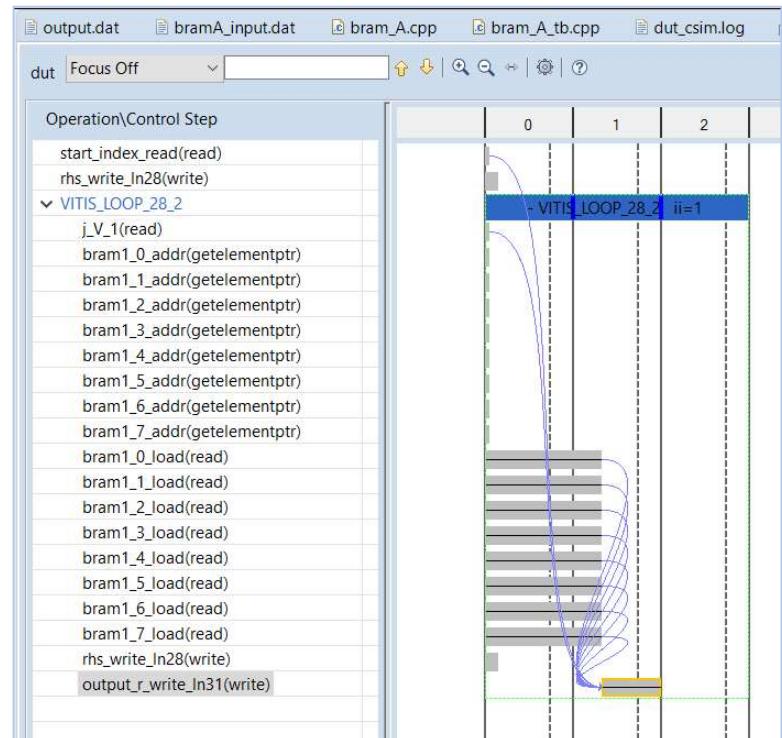
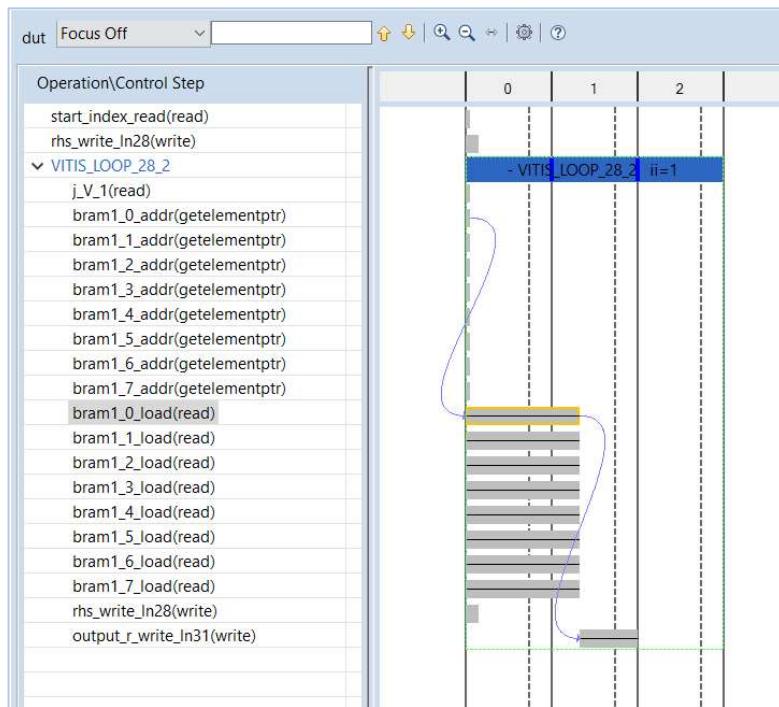
No user config\_op information

**Bind Storage Report**

No filter settings

Name	BRAM	URAM	Pragma	Variable	Storage	Impl	Latency
✓ dut	8	-	-	-	-	-	-
bram1_0_U	1	-	yes	bram1_0	ram_1p	bram	1
bram1_1_U	1	-	yes	bram1_1	ram_1p	bram	1
bram1_2_U	1	-	yes	bram1_2	ram_1p	bram	1
bram1_3_U	1	-	yes	bram1_3	ram_1p	bram	1
bram1_4_U	1	-	yes	bram1_4	ram_1p	bram	1

## Schedule Viewer result:



## Cosimulation printed output:

```
Vitis HLS Console
Compiling module xil_defaultlib.dataflow_monitor_1
Compiling module xil_defaultlib.aptb_dut_top
Compiling module work.glbl
Built simulation snapshot dut

***** xsim v2022.2 (64-bit)
*** SW Build 3671981 on Fri Oct 14 05:00:03 MDT 2022
*** IP Build 3669848 on Fri Oct 14 08:30:02 MDT 2022
** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

source xsim.dir/dut/xsim_script.tcl
# xsim {dut} -autoloadwcfg -tclbatch {dut.tcl}
Time resolution is 1 ps
source dut.tcl
## run all
///////////////////////////////
// Inter-Transaction Progress: Completed Transaction / Total Transaction
// Intra-Transaction Progress: Measured Latency / Latency Estimation * 100%
//
// RTL Simulation : "Inter-Transaction Progress" ["Intra-Transaction Progress"] @ "Simulation Time"
/////////////////////////////
// RTL Simulation : 0 / 8 [0.00%] @ "125000"
// RTL Simulation : 1 / 8 [100.00%] @ "235000"
// RTL Simulation : 2 / 8 [100.00%] @ "345000"
// RTL Simulation : 3 / 8 [100.00%] @ "455000"
// RTL Simulation : 4 / 8 [100.00%] @ "565000"
// RTL Simulation : 5 / 8 [100.00%] @ "675000"
// RTL Simulation : 6 / 8 [100.00%] @ "785000"
// RTL Simulation : 7 / 8 [100.00%] @ "895000"
// RTL Simulation : 8 / 8 [100.00%] @ "1005000"
/////////////////////////////
$finish called at time : 1065 ns : File "C:/Users/velic/AppData/Roaming/Xilinx/Vitis/project_20/solution2/sim/verilog/dut.autotb.v" Line 270
## quit
INFO: [Common 17-206] Exiting xsim at Mon Apr 3 20:07:20 2023...
INFO: [COSIM 212-316] Starting C post checking ...
all the read input sets passed
INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 8
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 2 seconds. CPU system time: 0 seconds. Elapsed time: 97.482 seconds; current allocated memory: 11.754 MB
INFO: [HLS 200-112] Total CPU user time: 4 seconds. Total CPU system time: 2 seconds. Total elapsed time: 112.961 seconds; peak allocated memory: 108.031 MB.
Finished C/RTL cosimulation.
```

## Cosimulation Report:

Cosimulation Report for 'dut'

General Information

Date: Mon Apr 3 20:07:28 IST 2023	Solution: solution2 (Vivado IP Flow Target)
Version: 2022.2 (Build 3670227 on Oct 13 2022)	Product family: zynq
Project: project_20	Target device: xc7z020-clg484-1
Status: Pass	

Cosim Options

Tool: Vivado XSIM	RTL: Verilog
-------------------	--------------

Performance Estimates

Modules & Loops	Avg II	Max II	Min II	Avg Latency	Max Latency	Min Latency
dut	11	11	11	9	9	9
© VITIS_LOOP_28_2	11	11	10	10	10	10

## PART-A\_2

### Design:

Main function:

```
1 #include <hls_stream.h>
2 #include <iostream>
3 #include <fstream>
4 #include<stdio.h>
5 #define BRAM_DEPTH 64
6 #define BUNDLE_SIZE 8
7 #include "ap_int.h"
8 typedef ap_uint <8> size;
9 void dut(size start_index, hls::stream<size> & output) {
10     #pragma HLS INTERFACE ap_none port=start_index
11     #pragma HLS INTERFACE axis register port=output
12
13     size bram1[BRAM_DEPTH];
14     #pragma HLS BIND_STORAGE variable=bram1 type=ram_1p impl=bram
15     #pragma HLS ARRAY_PARTITION variable=bram1 block factor=8
16
17     size i=0,j=0;
18     for ( i = 0; i < BRAM_DEPTH; i++) {
19         bram1[i]=i;
20     }
21     /*FILE *fp;
22     fp=fopen("populate.dat","r");
23     while(fscanf(fp,"%d",&bram1[i])==1)
24     {
25         i++;
26     }*/
27     size bundle1[BUNDLE_SIZE];
28     for (j= 0; j < 8; j++) {
29         #pragma HLS PIPELINE II=1
30         bundle1[j] = bram1[(start_index +j*8)%BRAM_DEPTH];
31         output.write(bundle1[j]);
32     }
33 }
34 }
```

## Self-checking Test Bench:

```
dut_csim.log *bram2_A_tb.cpp Synthesis Summary(solution1) Schedule Viewer(solution1) bram2_A.cpp
1 #include <hls_stream.h>
2 #include <iostream>
3 #define BRAM_DEPTH 64
4 #define BUNDLE_SIZE 8
5 #include <ap_int.h>
6 typedef ap_uint<8> size;
7 void dut(size start_index, hls::stream<size>& output);
8
9 int main()
10 {
11     hls::stream<size> output;
12     size out,start_index;
13     size golden_data[8][8];
14     size func_data[8][8];
15     size index[8];
16     int i=0,j=0,flag=0,count=0;
17     FILE *fp;
18
19     fp=fopen("bram2A_input.dat","r");
20     if (fp == NULL) {
21         printf("Error opening file\n");
22         return 1;
23     }
24     while(fscanf(fp,"%d %d %d %d %d %d %d %d",&index[i],&golden_data[i][0],&golden_data[i][1],
25                 &golden_data[i][2],&golden_data[i][3],&golden_data[i][4],&golden_data[i][5],&golden_data[i][6],
26                 &golden_data[i][7]) == 9) {
27         i++;
28     }
29     fclose(fp);
30
31
32     for(i=0; i<8; i++){
33         start_index=index[i];
34         dut(start_index,output);
35         //output >> out;
36         for (size j=0 ; j < BUNDLE_SIZE; j++) {
37             func_data[i][j]=output.read();
38         }
39     }
40     fp=fopen("output.dat","w");
41     for (int i = 0; i < 8; i++) {
42         for (int j = 0; j < BUNDLE_SIZE; j++){
43             if(func_data[i][j]==golden_data[i][j]){
44                 count++;
45             }
46         }
47         if(count==8)
48         {
49             fprintf(fp,"%s\n","Test Passed");
50         }
51         else
52         {
53             fprintf(fp,"%s\n","Test Failed!");
54             flag=1;
55         }
56         count=0;
57     }
58     fclose(fp);
59
60     if(flag==0){
61         std::cout<<"all the read input sets passed"<<std::endl;
62     }
63     else{
64         std::cout<<"few read input sets were not passed"<<std::endl;
65     }
66     return 0;
67 }
68
```

## Input File:

The screenshot shows a code editor window with three tabs: "bram2A\_input.dat", "bram2\_A\_tb.cpp", and "Synthesis S". The "bram2A\_input.dat" tab is active, displaying the following binary data:

```
18 8 16 24 32 40 48 56 0  
216 16 24 32 40 48 56 0 8  
324 24 32 40 48 56 0 8 16  
432 32 40 48 56 0 8 16 24  
540 40 48 56 0 8 16 24 32  
648 48 56 0 8 16 24 32 40  
756 56 0 8 16 24 32 40 48  
864 0 8 16 24 32 40 48 56
```

## C simulation printed output:

The screenshot shows a terminal window with several tabs: "bram2\_A.cpp", "bram2A\_input.dat", "bram2\_A\_tb.cpp", "dut\_csim.log", "Synthesis Summary(solution1)", and "Co-". The "dut\_csim.log" tab is active, displaying the following simulation log:

```
1 INFO: [SIM 2] **** CSIM start ****  
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.  
3 Compiling ../../../../../OneDrive/Desktop/vitis/bram2_A_tb.cpp in debug mode  
4 Compiling ../../../../../OneDrive/Desktop/vitis/bram2_A.cpp in debug mode  
5 Generating csim.exe  
6 all the read input sets passed  
7 INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 8  
8 INFO: [SIM 1] CSim done with 0 errors.  
9 INFO: [SIM 3] **** CSIM finish ****  
10
```

## Output File:

The screenshot shows a code editor window with three tabs: "bram2\_A.cpp", "bram2A\_input.dat", and "br". The "bram2\_A.cpp" tab is active, displaying the following test results:

```
1 Test Passed  
2 Test Passed  
3 Test Passed  
4 Test Passed  
5 Test Passed  
6 Test Passed  
7 Test Passed  
8 Test Passed  
9
```

```

bram2A_input.dat dut_csim.log bram2_A_tb.cpp Synthesis Summary(solution1) Schedule Viewer(solution1)
1 INFO: [SIM 2] **** CSIM start ****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3 Compiling ../../../../../../OneDrive/Desktop/vitis/bram2_A_tb.cpp in debug mode
4 Compiling ../../../../../../OneDrive/Desktop/vitis/bram2_A.cpp in debug mode
5 Generating csim.exe
6 few read input sets were not passed
7 INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 8
8 INFO: [SIM 1] CSim done with 0 errors.
9 INFO: [SIM 3] **** CSIM finish ****
10

```

## Output File:

The screenshot shows two windows. On the left is a terminal window titled 'bram2A\_input.dat' displaying a sequence of binary values: 18 8 16 24 32 40 48 56 0, followed by several lines of 16-bit pairs. On the right is an 'Explorer' window showing the project structure. The 'solution1' folder contains a 'build' directory which includes 'bram2A\_input.dat', 'csim.exe', 'csim.mk', 'Makefile.rules', 'output.dat', 'run\_sim.tcl', 'sim.bat', and 'obj'. An orange arrow points from the 'output.dat' file in the terminal to the 'output.dat' file in the 'solution1\build' folder.

## Synthesis Report:

The screenshot shows the 'Synthesis Summary Report of 'dut''. It includes sections for General Information, Timing Estimate, Performance & Resource Estimates, and Performance Pragma. The 'Performance & Resource Estimates' section contains a table with columns: Modules & Loops, Issue Type, Violation Type, Distance, Slack, Latency(cycles), Latency(ns), Iteration Latency, Interval, Trip Count, Pipelined, BRAM, DSP, FF, LUT, URAM. The table shows data for the 'dut' module and its sub-loop 'dut\_Pipeline\_VITIS\_LOOP\_28\_2'.

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
✓ dut	-	-	-	-	14	140.000	-	15	-	no	8	0	64	160	0
✓ dut_Pipeline_VITIS_LOOP_28_2	-	-	-	-	10	100.000	-	10	-	no	0	0	15	127	0
✓ VITIS_LOOP_28_2	-	-	-	-	8	80.000	2	1	8	yes	-	-	-	-	-

**HW Interfaces**

- AXIS**

Interface	Register Mode	TDATA	TREADY	TVALID
output_r	both	8	1	1

- Other Ports**

Interface	Mode	Bitwidth
start_index	ap_none	8

- TOP LEVEL CONTROL**

Interface	Type	Ports
ap_clk	clock	ap_clk
ap_rst_n	reset	ap_rst_n
ap_ctrl	ap_ctrl_hs	ap_done ap_idle ap_ready ap_start

**SW I/O Information**

  - Top Function Arguments**

Argument	Direction	Datatype
start_index	in	ap_uint<8>
output	out	stream<ap_uint<8> 0>&

  - SW-to-HW Mapping**

Argument	HW Interface	HW Type
start_index	start_index	port
output	output_r	interface

**Pragma Report**

    - Valid Pragma Syntax**

Type	Options	Location	Function
interface	ap_none port=start_index	./././OneDrive/Desktop/vitis/bram2_A.cpp:10	dut
interface	axis register port=output	./././OneDrive/Desktop/vitis/bram2_A.cpp:11	dut
bind_storage	variable=bram1 type=ram_1p impl=bram	./././OneDrive/Desktop/vitis/bram2_A.cpp:14	dut
array_partition	variable=bram1 block factor=8	./././OneDrive/Desktop/vitis/bram2_A.cpp:15	dut
pipeline	ll=1	./././OneDrive/Desktop/vitis/bram2_A.cpp:29	dut

    - Bind Op Report**

No filter settings

Name	DSP	Pragma	Variable	Op	Impl	Latency
du	-	-	-	-	-	-
du_Pipeline_VITIS_LOOP_28_2	-	-	-	-	-	-

No user config\_op information

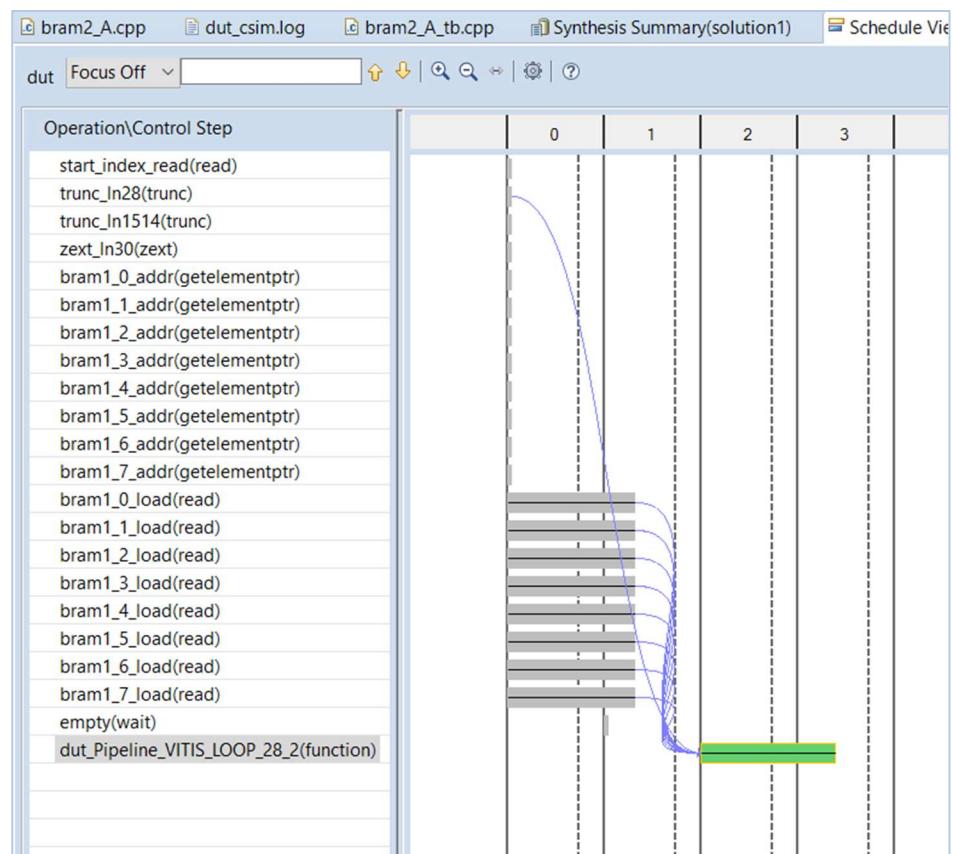
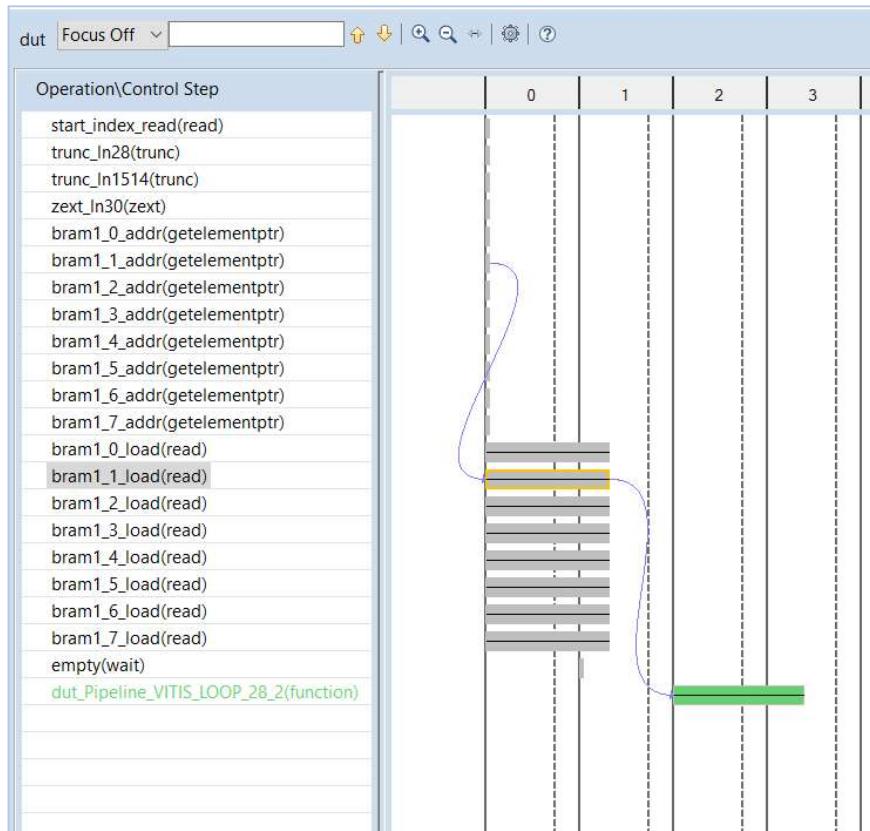
    - Bind Storage Report**

No filter settings

Name	BRAM	URAM	Pragma	Variable	Storage	Impl	Latency
du	8	-	-	-	-	-	-
bram1_0_U	1	-	yes	bram1_0	ram_1p	bram	1
bram1_1_U	1	-	yes	bram1_1	ram_1p	bram	1
bram1_2_U	1	-	yes	bram1_2	ram_1p	bram	1
bram1_3_U	1	-	yes	bram1_3	ram_1p	bram	1
bram1_4_U	1	-	yes	bram1_4	ram_1p	bram	1

No user config\_op information

## Schedule Viewer result:



## Cosimulation printed output:

```
Vitis HLS Console
Compiling module xil_defaultlib.dataflow_monitor_1
Compiling module xil_defaultlib.apatb_dut_top
Compiling module work.glbl
Built simulation snapshot dut

***** xsim v2022.2 (64-bit)
**** SW Build 3671981 on Fri Oct 14 05:00:03 MDT 2022
**** IP Build 3669848 on Fri Oct 14 08:30:02 MDT 2022
** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

source xsim.dir/dut/xsim_script.tcl
# xsim {dut} -autoloadwcfg -tclbatch {dut.tcl}
Time resolution is 1 ps
source dut.tcl
## run all
///////////////////////////////
// Inter-Transaction Progress: Completed Transaction / Total Transaction
// Intra-Transaction Progress: Measured Latency / Latency Estimation * 100%
//
// RTL Simulation : "Inter-Transaction Progress" ["Intra-Transaction Progress"] @ "Simulation Time"
/////////////////////////////
// RTL Simulation : 0 / [0.00%] @ "125000"
// RTL Simulation : 1 / 8 [100.00%] @ "265000"
// RTL Simulation : 2 / 8 [100.00%] @ "395000"
// RTL Simulation : 3 / 8 [100.00%] @ "525000"
// RTL Simulation : 4 / 8 [100.00%] @ "655000"
// RTL Simulation : 5 / 8 [100.00%] @ "785000"
// RTL Simulation : 6 / 8 [100.00%] @ "915000"
// RTL Simulation : 7 / 8 [100.00%] @ "1045000"
// RTL Simulation : 8 / 8 [100.00%] @ "1175000"
/////////////////////////////
$finish called at time : 1235 ns : File "C:/Users/velic/AppData/Roaming/Xilinx/Vitis/project_21/solution1/sim/verilog/dut.autotb.v" Line 270
## quit
INFO: [Common 17-206] Exiting xsim at Mon Apr 3 22:33:35 2023...
INFO: [COSIM 212-316] Starting C post checking ...
all the read input sets passed
INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 8
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 1 seconds. CPU system time: 1 seconds. Elapsed time: 83.422 seconds; current allocated memory: 11.984 MB.
INFO: [HLS 200-112] Total CPU user time: 3 seconds. Total CPU system time: 2 seconds. Total elapsed time: 95.967 seconds; peak allocated memory: 108.586 MB.
Finished C/RTL cosimulation.
```

## Cosimulation Report:

Cosimulation Report for 'dut'

General Information

Date: Mon Apr 3 22:33:42 IST 2023	Solution: solution1 (Vivado IP Flow Target)
Version: 2022.2 (Build 3670227 on Oct 13 2022)	Product family: zynq
Project: project_21	Target device: xc7z020-clg484-1
Status: Pass	

Cosim Options

Tool: Vivado XSIM	RTL: Verilog
-------------------	--------------

Performance Estimates

Modules & Loops

	Avg II	Max II	Min II	Avg Latency	Max Latency	Min Latency
dut	13	13	13	12	12	12
dut_Pipeline_VITIS_LOOP_28_2	13	13	13	8	8	8

## PART-B

### Design:

Main function:

```
1 #include <ap_int.h>
2 #include <hls_stream.h>
3
4 const int BRAM_DEPTH = 4;
5 const int INPUT_WIDTH = 8;
6 const int OUTPUT_WIDTH = 32;
7
8 typedef ap_uint<INPUT_WIDTH> input_t;
9 typedef ap_uint<OUTPUT_WIDTH> output_t;
10 int index=0;
11
12 void dut(hls::stream<input_t>& in_stream, hls::stream<output_t>& out_stream) {
13     #pragma HLS INTERFACE mode=axis port=out_stream register
14     #pragma HLS INTERFACE mode=axis port=in_stream register
15     input_t bram[4];
16     // #pragma HLS ARRAY_PARTITION factor=4 type=block variable=bram
17     #pragma HLS ARRAY_RESHAPE dim=1 type=complete variable=bram
18     bram[0] = in_stream.read();
19     bram[1] = in_stream.read();
20     bram[2] = in_stream.read();
21     bram[3] = in_stream.read();
22     for (int i = 0; i < BRAM_DEPTH; i++) {
23         #pragma HLS PIPELINE II=1
24         out_stream.write(bram[i]);
25     }
26 }
27 }
```

### Self-checking Test Bench:

```
1 #include <iostream>
2 #include <fstream>
3 #include <hls_stream.h>
4 #include <ap_int.h>
5 const int BRAM_DEPTH = 4;
6 const int INPUT_WIDTH = 8;
7 const int OUTPUT_WIDTH = 32;
8
9 typedef ap_uint<INPUT_WIDTH> input_t;
10 typedef ap_uint<OUTPUT_WIDTH> output_t;
11
12 void dut(hls::stream<input_t>& in_stream, hls::stream<output_t>& out_stream);
13 int main() {
14     hls::stream<input_t> in_stream;
15     hls::stream<output_t> out_stream;
16     input_t test_inputs[10][4];
17     input_t golden_data[10][4];
18     input_t bram_func_data[10][4];
19     int i=0,j=0,k=0,flag=0,count=0;
20     FILE *fp;
21
22     fp=fopen("input.dat","r");
23     if (fp == NULL) {
24         printf("Error opening file\n");
25         return 1;
26     }
27     while(fscanf(fp,"%d %d %d %d %d %d %d %d",&test_inputs[i][0],&test_inputs[i][1],&test_inputs[i][2],
```

```

28         &test_inputs[i][3],&golden_data[i][0],&golden_data[i][1],&golden_data[i][2],&golden_data[i][3]) == 8) {
29     i++;
30 }
31 fclose(fp);
32 for (int i = 0; i < 10; i++) {
33     for (int j = 0; j < BRAM_DEPTH; j++) {
34         input_t data = test_inputs[i][j];
35         in_stream.write(data);
36     }
37 }
38 fp=fopen("output.dat","w");
39 for (int i = 0; i < 10; i++) {
40     dut(in_stream, out_stream);
41     for (int j = 0; j < BRAM_DEPTH; j++){
42         //std::cout << "Output: " << out_stream.read() << std::endl;
43         bram_func_data[i][j]=out_stream.read();
44     }
45 }
46 for (int i = 0; i < 10; i++) {
47     for (int j = 0; j < BRAM_DEPTH; j++){
48         if(bram_func_data[i][j]==golden_data[i][j]){
49             count++;
50         }
51     }
52     if(count==4)
53     {
54         //fprintf(fp,"%d %d %d %d %s\n",bram_func_data[k][0],bram_func_data[k][1],bram_func_data[k][2],
55         //        bram_func_data[k][3],"Test Passed");
56         //k++;
57         fprintf(fp,"%s\n","Test Passed");
58     }
59     else
60     {
61         //fprintf(fp,"%d %d %d %d %s\n",bram_func_data[k][0],bram_func_data[k][1],bram_func_data[k][2],
62         //        bram_func_data[k][3],"Test Failed!");
63         fprintf(fp,"%s\n","Test Failed!");
64         flag=1;
65         //k++;
66     }
67     count=0;
68 }
69 fclose(fp);
70
71     if(flag==0){
72         std::cout<<"all the read input sets passed" << std::endl;
73     }
74     else{
75         std::cout<<"few read input sets were not passed" << std::endl;
76     }
77 return 0;
78 }
79

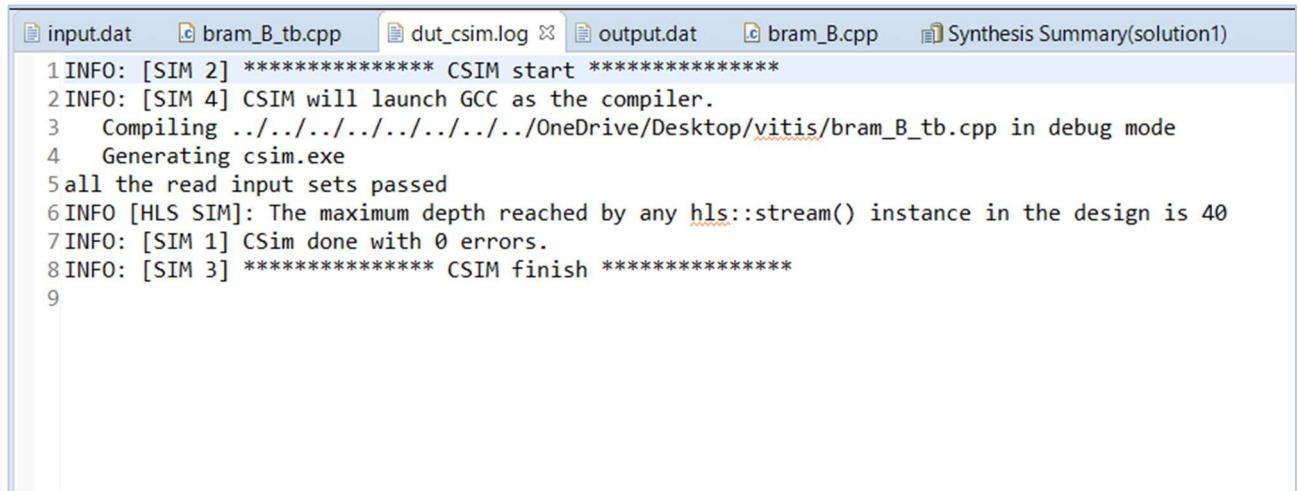
```

## Input File:

input.dat    bram\_B\_tb.cpp    dut\_csim.lc

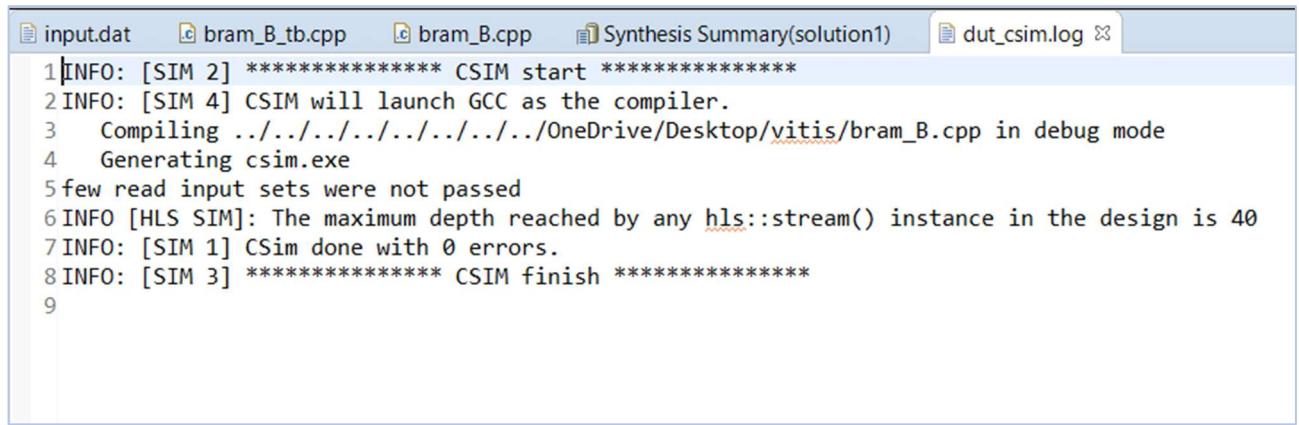
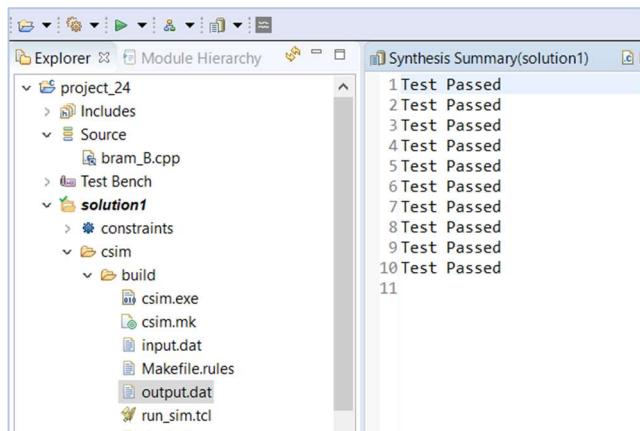
11 2 3 4 1 2 3 4
25 6 7 8 5 6 7 8
33 6 9 8 3 6 9 8
41 4 7 8 1 4 7 8
52 5 8 7 2 5 8 7
67 4 1 2 7 4 1 2
71 2 3 6 1 2 3 6
84 5 6 9 4 5 6 9
93 6 5 4 3 6 5 4
101 5 9 6 1 5 9 6

## C simulation printed output:



```
1 INFO: [SIM 2] **** CSIM start ****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3   Compiling ../../../../../../OneDrive/Desktop/vitis/bram_B_tb.cpp in debug mode
4   Generating csim.exe
5 all the read input sets passed
6 INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 40
7 INFO: [SIM 1] CSim done with 0 errors.
8 INFO: [SIM 3] **** CSIM finish ****
9
```

## Output File:



```
1 INFO: [SIM 2] **** CSIM start ****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3   Compiling ../../../../../../OneDrive/Desktop/vitis/bram_B.cpp in debug mode
4   Generating csim.exe
5 few read input sets were not passed
6 INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 40
7 INFO: [SIM 1] CSim done with 0 errors.
8 INFO: [SIM 3] **** CSIM finish ****
9
```

## Output File:

The image shows two terminal windows side-by-side. The left window displays a series of test results:

```
1 Test Passed
2 Test Passed
3 Test Passed
4 Test Passed
5 Test Failed!
6 Test Passed
7 Test Passed
8 Test Passed
9 Test Passed
10 Test Failed!
11
```

The right window displays a 10x4 matrix of integers:

11	2	3	4
25	6	7	8
33	6	9	8
41	4	7	8
52	5	8	7
67	4	1	2
71	2	3	6
84	5	6	9
93	6	5	4
101	5	9	6

Arrows from the failed test numbers in the left window point to the corresponding rows in the matrix.

## Synthesis Report:

The image shows the 'Synthesis Summary Report of 'dut'' in the Vivado interface. It includes sections for General Information, Timing Estimate, Performance & Resource Estimates, Performance Pragma, HW Interfaces, and SW I/O Information.

**General Information:**

Date: Mon Apr 3 17:51:16 2023	Solution: solution1 (Vivado IP Flow Target)
Version: 2022.2 (Build 3670227 on Oct 13 2022)	Product family: artix7
Project: project_24	Target device: xc7a200t-fbg676-2

**Timing Estimate:**

Target	Estimated	Uncertainty
10.00 ns	3.308 ns	2.70 ns

**Performance & Resource Estimates:**

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
dut	-	-	-	-	12	120.000	-	13	-	no	0	0	86	215	0
dut_Pipeline_VITIS_LOOP_22_1	-	-	-	-	6	60.000	-	6	-	no	0	0	14	169	0
VITIS_LOOP_22_1	-	-	-	-	4	40.000	2	1	4	yes	-	-	-	-	-

**Performance Pragma:**

Modules & Loops	Target TI(cycles)	TI(cycles)	TI met
dut	-	-	-
dut_Pipeline_VITIS_LOOP_22_1	-	-	-

**HW Interfaces:**

**AXIS**

Interface	Register Mode	TDATA	TREADY	TVALID
in_stream	both	8	1	1
out_stream	both	32	1	1

**TOP LEVEL CONTROL**

Interface	Type	Ports
ap_clk	clock	ap_clk
ap_rst_n	reset	ap_rst_n
ap_ctrl	ap_ctrl_hs	ap_done ap_idle ap_ready ap_start

**SW I/O Information:**

**Top Function Arguments:**

Argument	Direction	Datatype
in_stream	in	stream<ap_uint<8> 0>&
out_stream	out	stream<ap_uint<32> 0>&

**SW-to-HW Mapping:**

Argument	HW Interface	HW Type
in_stream	in_stream	interface
out_stream	out_stream	interface

▼ Top Function Arguments

Argument	Direction	Datatype
in_stream	in	stream<ap_uint<8> 0>&
out_stream	out	stream<ap_uint<32> 0>&

▼ SW-to-HW Mapping

Argument	HW Interface	HW Type
in_stream	in_stream	interface
out_stream	out_stream	interface

▼ Pragma Report

▼ ValidPragma Syntax

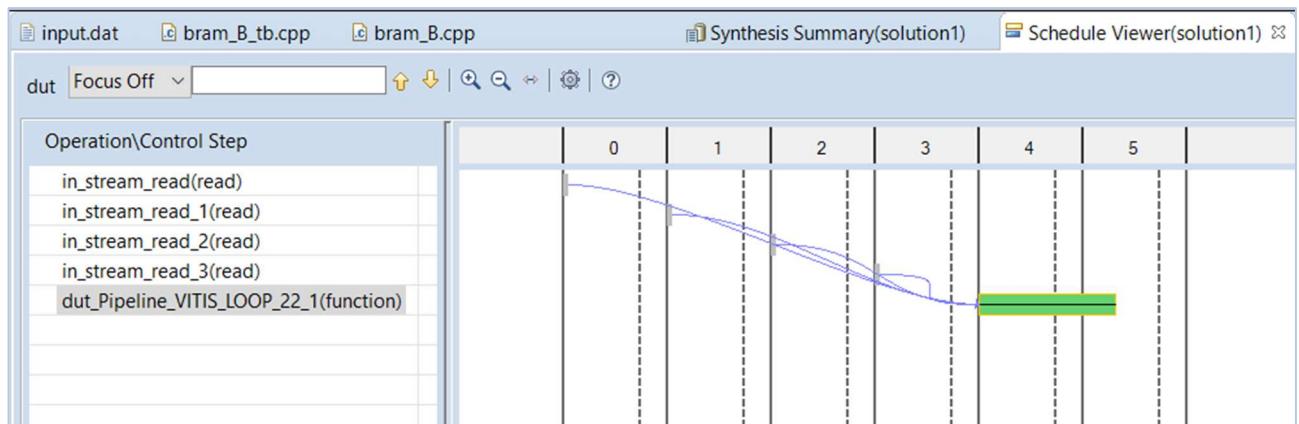
Type	Options	Location	Function
interface	mode=axis port=out_stream register	./././OneDrive/Desktop/vitis/ <b>bram_B.cpp:13</b>	dut
interface	mode=axis port=in_stream register	./././OneDrive/Desktop/vitis/ <b>bram_B.cpp:14</b>	dut
array.reshape	dim=1 type=complete variable=bram	./././OneDrive/Desktop/vitis/ <b>bram_B.cpp:17</b>	dut
pipeline	ll=1	./././OneDrive/Desktop/vitis/ <b>bram_B.cpp:23</b>	dut

▼ Bind Op Report

No filter settings

Name	DSP	Pragma	Variable	Op	Impl	Latency
✓ dut	-	-	-	-	-	-
> dut_Pipeline_VITIS_LOOP_22_1	-	-	-	-	-	-

## Schedule Viewer result:



## Cosimulation printed output:

```

Console ✘ Errors ✘ Guidance ✘ Properties ✘ Man Pages ✘ Git Repositories ✘ Modules/Loops
Vitis HLS Console
Built simulation snapshot dut

***** xsim v2022.2 (64-bit)
***** SW Build 3671981 on Fri Oct 14 05:00:03 MDT 2022
***** IP Build 3669848 on Fri Oct 14 08:30:02 MDT 2022
** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

source xsim.dir/dut/xsim_script.tcl
# xsim {dut} -autoloadwcfg -tclbatch {dut.tcl}
Time resolution is 1 ps
source dut.tcl
## run all
///////////////////////////////
// Inter-Transaction Progress: Completed Transaction / Total Transaction
// Intra-Transaction Progress: Measured Latency / Latency Estimation * 100%
//
// RTL Simulation : "Inter-Transaction Progress" ["Intra-Transaction Progress"] @ "Simulation Time"
/////////////////////////////
// RTL Simulation : 0 / 10 [0.00%] @ "125000"
find kernel block.
// RTL Simulation : 1 / 10 [100.00%] @ "255000"
// RTL Simulation : 2 / 10 [100.00%] @ "365000"
// RTL Simulation : 3 / 10 [100.00%] @ "475000"
// RTL Simulation : 4 / 10 [100.00%] @ "585000"
// RTL Simulation : 5 / 10 [100.00%] @ "695000"
// RTL Simulation : 6 / 10 [100.00%] @ "805000"
// RTL Simulation : 7 / 10 [100.00%] @ "915000"
// RTL Simulation : 8 / 10 [100.00%] @ "1025000"
// RTL Simulation : 9 / 10 [100.00%] @ "1135000"
// RTL Simulation : 10 / 10 [100.00%] @ "1245000"
/////////////////////////////
$finish called at time : 1305 ns : File "C:/Users/velic/AppData/Roaming/Xilinx/Vitis/project_24/solution1/sim/verilog/dut.autob.v" Line 247
## quit
INFO: [Common 17-206] Exiting xsim at Mon Apr 3 17:55:12 2023...
INFO: [COSIM 212-316] Starting C post checking ...
all the read input sets passed
INFO [HLS SIM]: The maximum depth reached by any hls::stream() instance in the design is 40
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
INFO: [HLS 200-111] Finished Command cosim_design CPU user time: 2 seconds. CPU system time: 1 seconds. Elapsed time: 94.965 seconds; current allocated memory: 13.066 MB.
INFO: [HLS 200-112] Total CPU user time: 4 seconds. Total CPU system time: 2 seconds. Total elapsed time: 117.179 seconds; peak allocated memory: 111.305 MB.
Finished C/RTL cosimulation.

```

## Cosimulation Report:

The screenshot shows the Vivado Co-simulation Report for the 'dut' component. It includes sections for General Information, Cosim Options, and Performance Estimates.

**General Information:**

- Date: Mon Apr 3 17:55:19 IST 2023
- Version: 2022.2 (Build 3670227 on Oct 13 2022)
- Project: project\_24
- Status: Pass
- Solution: solution1 (Vivado IP Flow Target)
- Product family: artix7
- Target device: xc7a200t-fbg676-2

**Cosim Options:**

- Tool: Vivado XSIM
- RTL: Verilog

**Performance Estimates:**

Module	Avg II	Max II	Min II	Avg Latency	Max Latency	Min Latency
dut	11	12	11	10	11	10
dut_Pipeline_VITIS_LOOP_22_1	11	11	4	4	4	4
VITIS_LOOP_22_1	11	11	11	5	5	5

Github:

[https://github.com/velicharlagokulkumar/vitis-hls/tree/main/project\\_20](https://github.com/velicharlagokulkumar/vitis-hls/tree/main/project_20)  
[https://github.com/velicharlagokulkumar/vitis-hls/tree/main/project\\_21](https://github.com/velicharlagokulkumar/vitis-hls/tree/main/project_21)

[https://github.com/velicharlagokulkumar/vitis-hls/tree/main/project\\_24](https://github.com/velicharlagokulkumar/vitis-hls/tree/main/project_24)