



Hls ASSIGNMENT-1

Design an 8bit * 8bit multiplier using HLS. There will be two 8bit inputs (use char data type) and one 16bit output (use short data type). Use ap_none interface for the port interfaces. Write an HLS testbench to verify the output of the design. Pass 10 pair of input values to design and collect the output, and display inputs and outputs as part of the testbench. Verify if your design is working using C simulation first. Run HLS synthesis. Now, verify the design again using C/RTL co-simulation. Report your design and tb code, C simulation printed outputs, HLS resource consumption, HLS timing report, and C/RTL cosimulation printed outputs and report, in one document

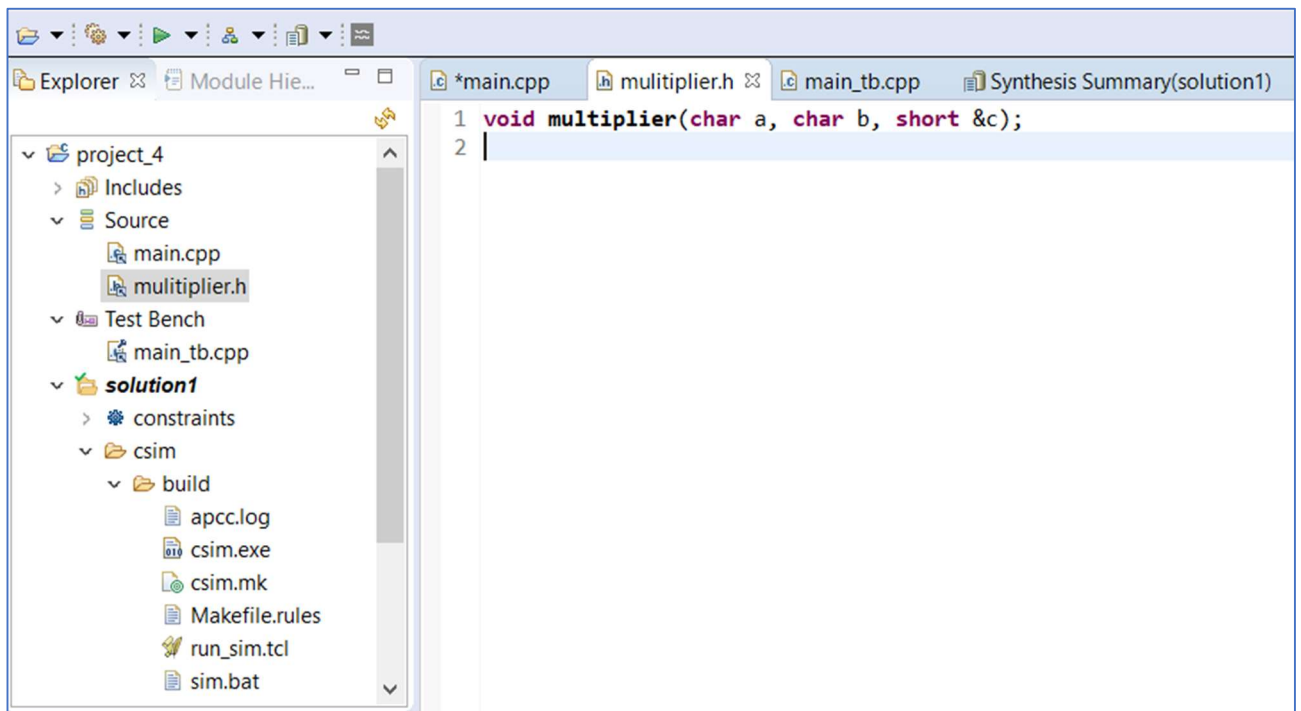
Design:

Multiplier function:

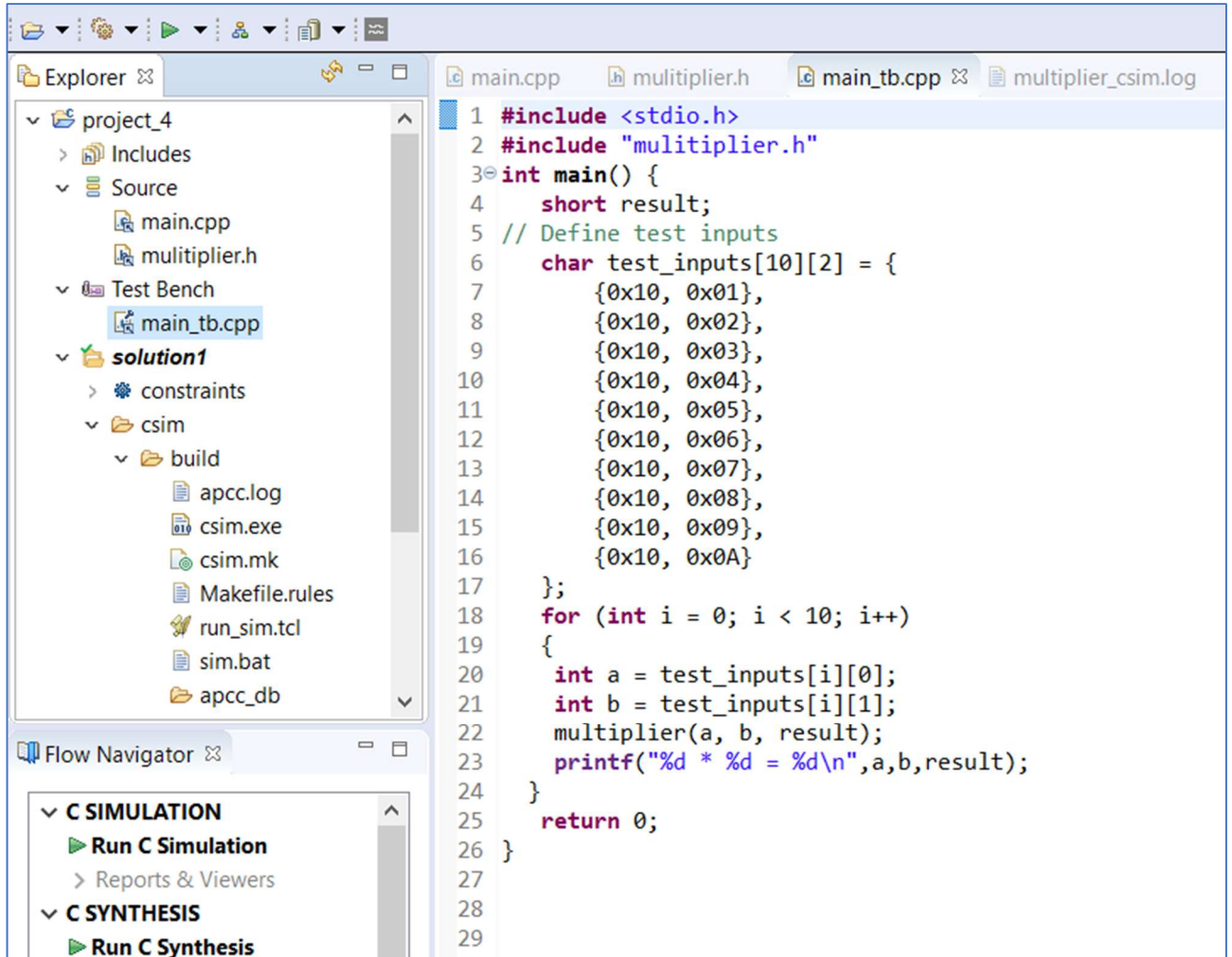
```
1 #include "multiplier.h"
2 void multiplier(char a, char b, short &c)
3 {
4     #pragma HLS INTERFACE ap_none port=a
5     #pragma HLS INTERFACE ap_none port=b
6     #pragma HLS INTERFACE ap_none port=c
7     c = a * b;
8 }
9
10
```

The screenshot shows the Xilinx IDE interface. On the left, the 'Explorer' pane displays the project structure for 'project_4', including 'Includes', 'Source' (with 'main.cpp' and 'multiplier.h'), 'Test Bench' (with 'main_tb.cpp'), and 'solution1' (with 'constraints', 'csim', and 'build' subfolders). The 'build' folder contains files like 'apcc.log', 'csim.exe', 'csim.mk', 'Makefile.rules', 'run_sim.tcl', and 'sim.bat'. The main editor window shows the code for 'main.cpp', which includes 'multiplier.h' and defines a 'multiplier' function. The function takes two 'char' inputs 'a' and 'b' and a 'short' output 'c' by reference. It uses three '#pragma HLS INTERFACE ap_none' directives to specify the port interfaces. The function body simply calculates 'c = a * b;'. The code is numbered from 1 to 10.

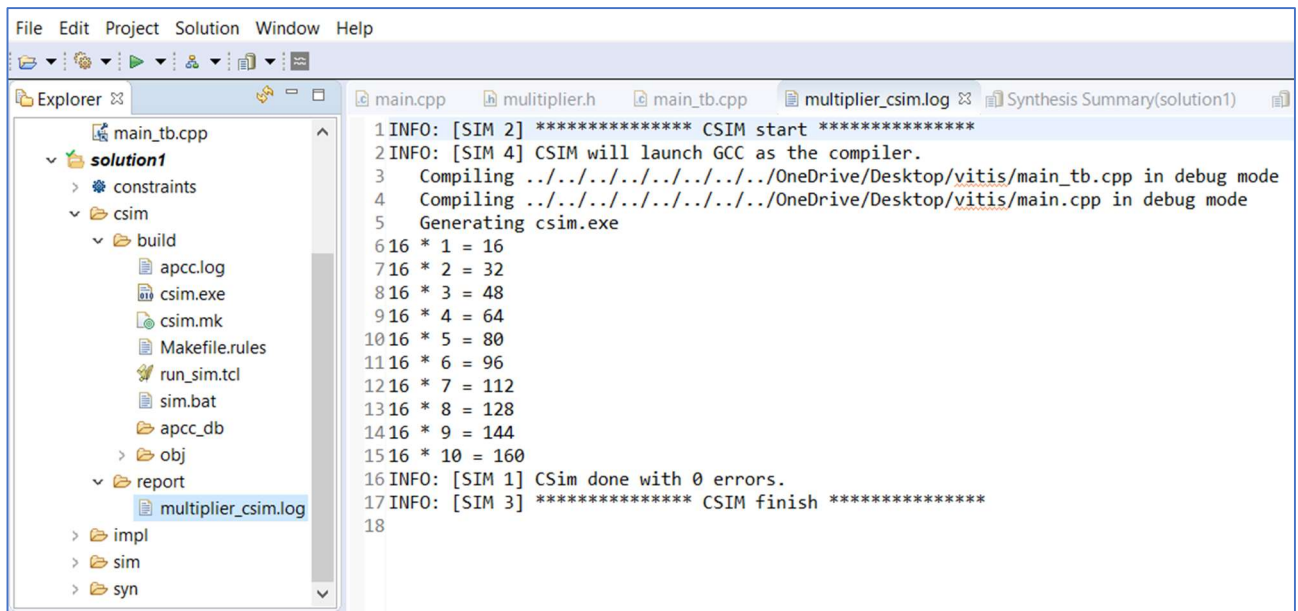
Header file:



Test Bench:



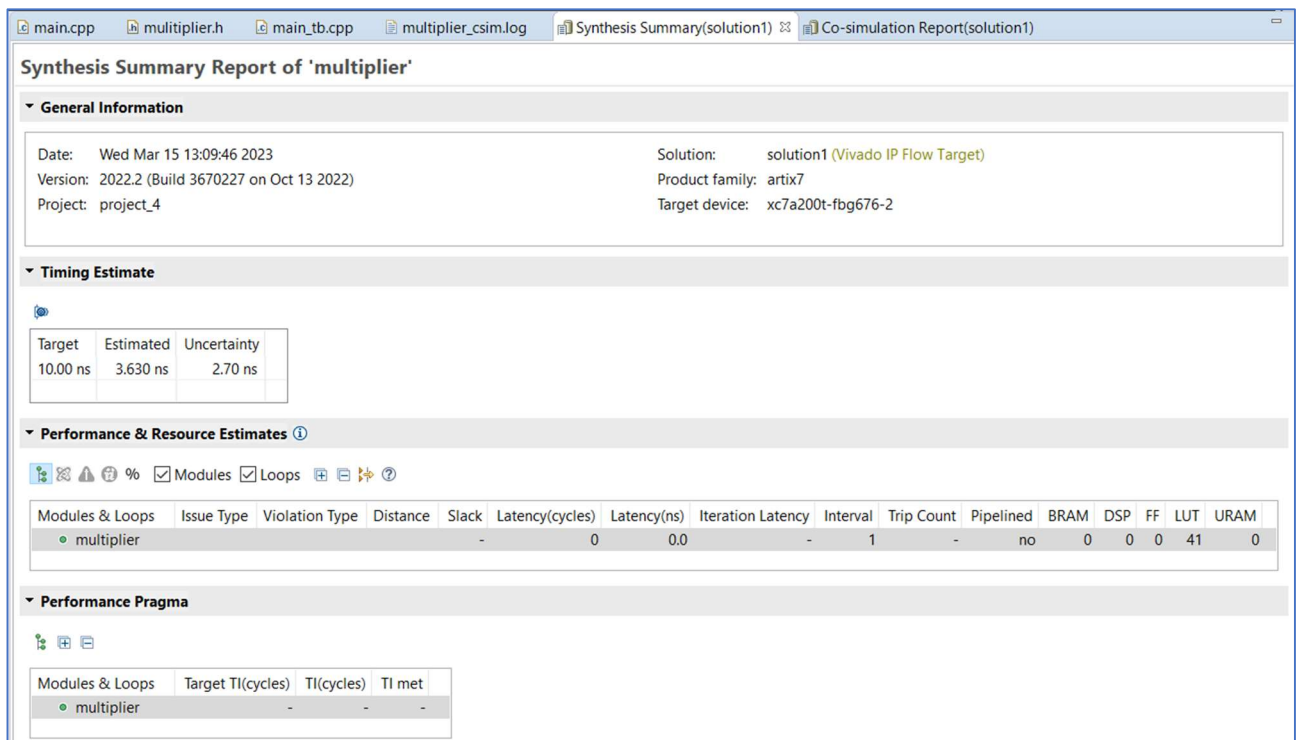
C simulation printed output:



The screenshot shows the Vivado IDE interface. On the left, the Explorer pane displays the project structure for 'solution1', including folders for constraints, csim, build, obj, report, impl, sim, and syn. The 'report' folder is expanded, showing 'multiplier_csim.log'. The main editor pane displays the contents of this log file, which contains the following text:

```
1 INFO: [SIM 2] ***** CSIM start *****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3   Compiling ../../../../../../OneDrive/Desktop/vitis/main_tb.cpp in debug mode
4   Compiling ../../../../../../OneDrive/Desktop/vitis/main.cpp in debug mode
5   Generating csim.exe
6 16 * 1 = 16
7 16 * 2 = 32
8 16 * 3 = 48
9 16 * 4 = 64
10 16 * 5 = 80
11 16 * 6 = 96
12 16 * 7 = 112
13 16 * 8 = 128
14 16 * 9 = 144
15 16 * 10 = 160
16 INFO: [SIM 1] CSim done with 0 errors.
17 INFO: [SIM 3] ***** CSIM finish *****
18
```

Synthesis Report:



The screenshot shows the Vivado Synthesis Summary Report for the 'multiplier' block. The report is organized into several sections:

- General Information**
 - Date: Wed Mar 15 13:09:46 2023
 - Version: 2022.2 (Build 3670227 on Oct 13 2022)
 - Project: project_4
 - Solution: solution1 (Vivado IP Flow Target)
 - Product family: artix7
 - Target device: xc7a200t-fbg676-2
- Timing Estimate**

Target	Estimated	Uncertainty
10.00 ns	3.630 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	
multiplier				-	0	0.0		-	1	-	no	0	0	0	41	0
- Performance Pragma**

Modules & Loops	Target TI(cycles)	TI(cycles)	TI met
multiplier	-	-	-

HW Interfaces

Other Ports

Interface	Mode	Bitwidth
a	ap_none	8
b	ap_none	8
c	ap_none	16

TOP LEVEL CONTROL

Interface	Type	Ports
ap_ctrl	ap_ctrl_hs	ap_done ap_idle ap_ready ap_start

SW I/O Information

Top Function Arguments

Argument	Direction	Datatype
a	in	char
b	in	char
c	out	short&

SW-to-HW Mapping

Argument	HW Interface	HW Type
a	a	port
b	b	port
c	c	port

Pragma Report

Valid Pragma Syntax

Bind Op Report

No filter settings

Name	DSP	Pragma	Variable	Op	Impl	Latency
multiplier	-					
mul_8s_8s_16_1_U1	-		mul_in7	mul	auto	0

No user config_op information

Bind Storage Report

No bind storage, config_storage information

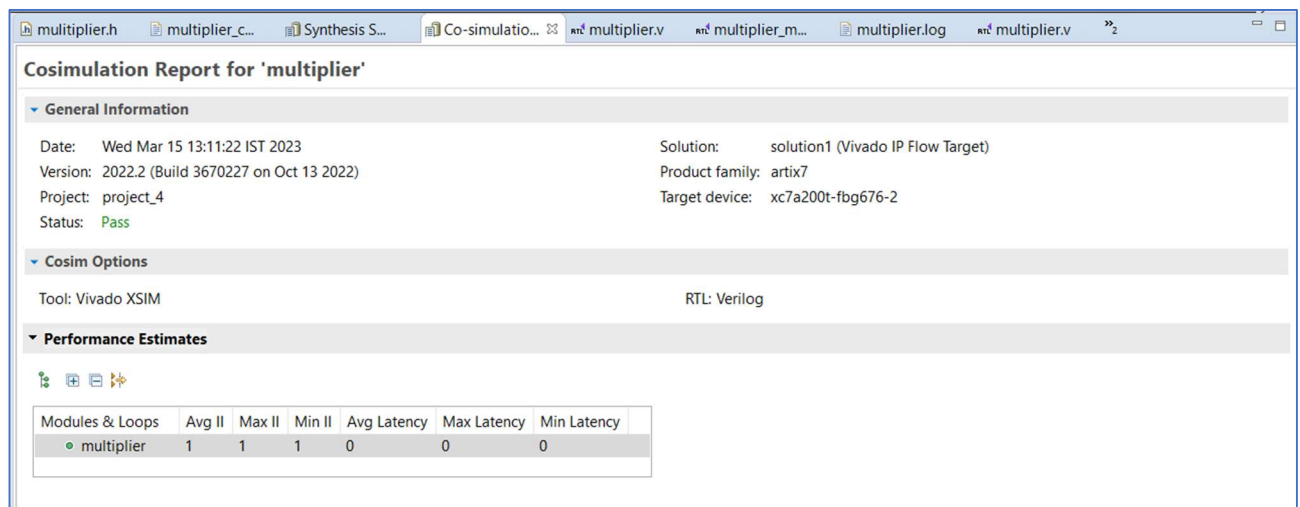
Cosimulation printed output:

```

88## quit]
89 INFO: [Common 17-206] Exiting xsim at Wed Mar 15 13:11:17 2023...
90 16 * 1 = 16
91 16 * 2 = 32
92 16 * 3 = 48
93 16 * 4 = 64
94 16 * 5 = 80
95 16 * 6 = 96
96 16 * 7 = 112
97 16 * 8 = 128
98 16 * 9 = 144
99 16 * 10 = 160
100 INFO: [COSIM-1000] *** C/RTL co-simulation finished: PASS ***
101 INFO: [COSIM-210] Design is translated to an combinational logic. II and Latency will be marked as all 0.
102

```

Cosimulation Report:



The screenshot shows the Vivado IDE interface with the 'multiplier' project open. The 'Cosimulation Report for 'multiplier'' is displayed in the main window. The report is organized into three sections: General Information, Cosim Options, and Performance Estimates.

General Information

Date:	Wed Mar 15 13:11:22 IST 2023	Solution:	solution1 (Vivado IP Flow Target)
Version:	2022.2 (Build 3670227 on Oct 13 2022)	Product family:	artix7
Project:	project_4	Target device:	xc7a200t-fbg676-2
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
• multiplier	1	1	1	0	0	0

Github:

https://github.com/velicharlagokulkumar/vitis-hls/tree/main/project_4