V.GOKUL KUMAR
FUTURE WIRELESS COMMUNICATIONS (FWC)
Indian Institute of Technology Hyderabad
**Email:** velicharla@outlook.com

Date:06-04-2023

# Hls ASSIGNMENT-6

Implement a 16bit shift register in HLS. The module should take 3 inputs: 16bit data value, 16bit shift value, and 1 bit left or right shift flag. Shift value is the value by which you need to shift (in the direction denoted by shift flag) the data value and produce the output (also 16bit). Implement it in the most efficient manner possible with what you have learned till now. Do C simulation, synthesis and C/RTL co-simulation like other assignments. The testbench should be a self-checking testbench.

The purpose of this assignment is to familiarise you with LOOP_PIPELINE and LOOP_UNROLL pragmas.

## Design:

Main function:

_csim.log | Synthesis Summary(solution1) | Schedule Viewer(solution1) | shift_reg.cpp

```
#include<hls_stream.h>
#include <ap_int.h>
typedef ap_uint<16> size;
void shift_register(hls::stream<size> &data_in, size shift_value, bool shift_flag,
                        hls::stream<size> &data_out) {
#pragma HLS INTERFACE mode=axis port=data_out register
#pragma HLS INTERFACE mode=axis port=data_in register

    size dout[16];
#pragma HLS ARRAY_PARTITION dim=1 factor=16 type=block variable=dout
    size din[16];
#pragma HLS ARRAY_PARTITION dim=1 factor=16 type=block variable=din

    for (int i = 0; i < 16; i++) {
             #pragma HLS UNROLL
               din[i]=data_in.read();
        }
          if (shift_flag) {    //Right shift
                    for (int i = 0; i < 16; i++) {
                        //#pragma HLS PIPELINE
                         #pragma HLS UNROLL
                        int shift_index = i - shift_value;
                        if (shift_index < 0) {
                            dout[i] = din[16 + shift_index];
                        } else {
                            dout[i] = din[shift_index];
                        }
                    }
                } else {
                    for (int i = 0; i < 16; i++) {
                     // #pragma HLS PIPELINE
                       #pragma HLS UNROLL
                        int shift_index = i + shift_value;
                        if (shift_index >= 16) {
                            dout[i] = din[shift_index - 16];
                         } else {
                             dout[i] = din[shift_index];
                         }
                    }
                 }

         for (int i = 0; i < 16; i++) {
            #pragma HLS UNROLL
             data_out.write(dout[i]);
         }
}

```

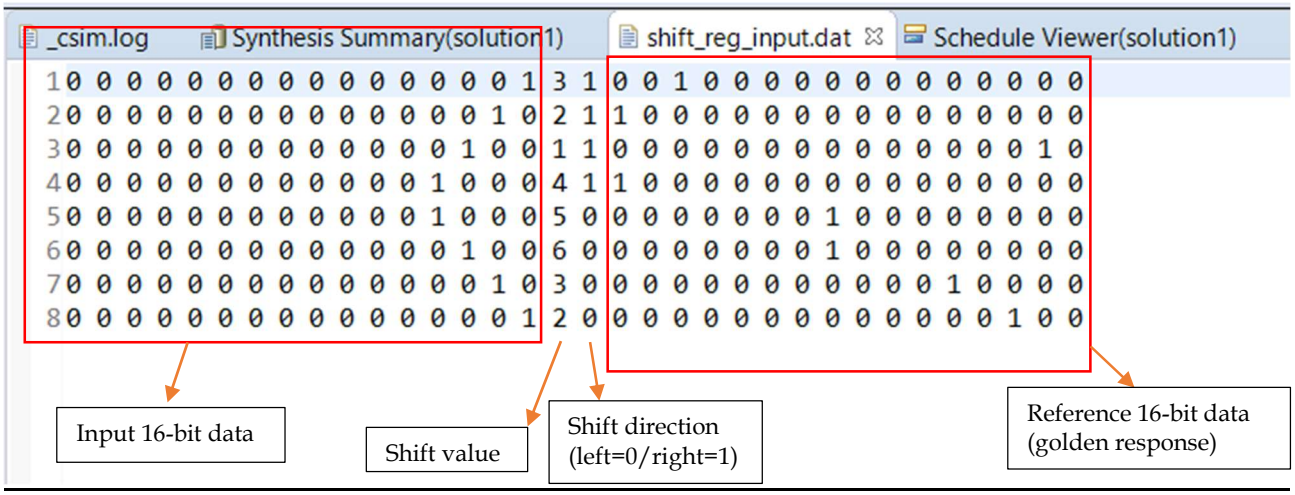
## Self-checking Test Bench:

_csim.log | Synthesis Summary(solution1) | Schedule Viewer(solution1) | shift_reg.cpp | shift_reg_tb.cpp

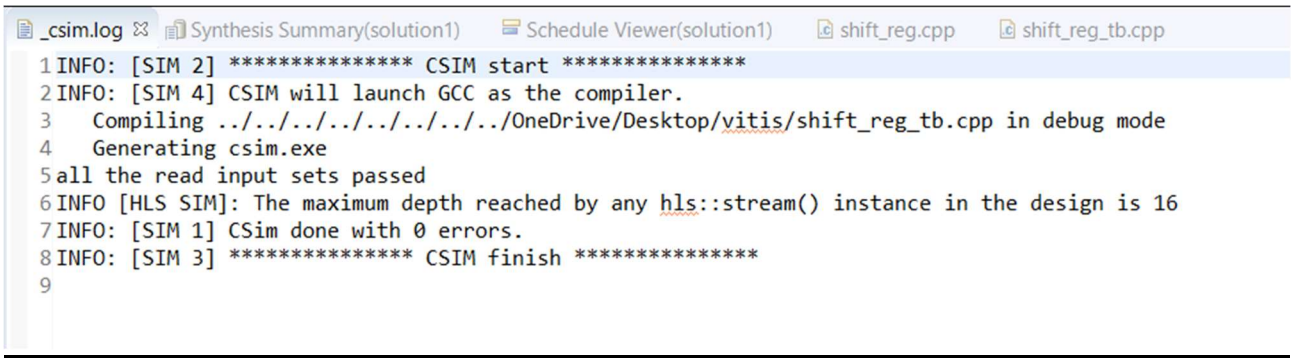
```
#include<hls_stream.h>
#include <ap_int.h>
#include <iostream>
#include <fstream>
using namespace std;

typedef ap_uint<16> size;

void shift_register(hls::stream<size> &data_in, size shift_value, bool shift_flag,
                        hls::stream<size> &data_out);
int main()
{
    hls::stream<size> data_in;
    size shift_value;
    hls::stream<size> data_out;

    size data[8][34];
    size dout[8][34];
    bool shift_flag=0;
    int i=0,j=0,k=0,flag=0,count=0;
        FILE *fp;

    ifstream inputFile("shift_reg_input.dat");

       for (int i = 0; i < 8; i++) {
           for (int j = 0; j < 34; j++) {
               inputFile >> data[i][j];
           }
       }

       inputFile.close();

  ifstream outputFile("output.dat");

     for(int i=0;i<8;i++){
         shift_value=data[i][16];
         shift_flag=data[i][17];
      for(int j=0;j<16;j++){
     data_in.write(data[i][j]);
      }

   shift_register(data_in,shift_value,shift_flag,data_out);

   fp=fopen("output.dat","w");
    for(int k=0;k<16;k++){
     dout[i][k]= data_out.read();
    }
    }

      for (int i = 0; i < 8; i++) {
         for (int j = 0; j < 16; j++){
            if(dout[i][j]==data[i][j+18]){
              count++;
            }
         }
         if(count==16)
         {
            fprintf(fp,"%s\n","Test Passed");
         }
         else
         {
            fprintf(fp,"%s\n","Test Failed!");
            flag=1;
         }
         count=0;
       }
       fclose(fp);

           if(flag==0){
           std::cout<<"all the read input sets passed"<<std::endl;
```

```
        }
         else{
          std::cout<<"few read input sets were not passed"<<std::endl;
        }
      return 0;
    }
```

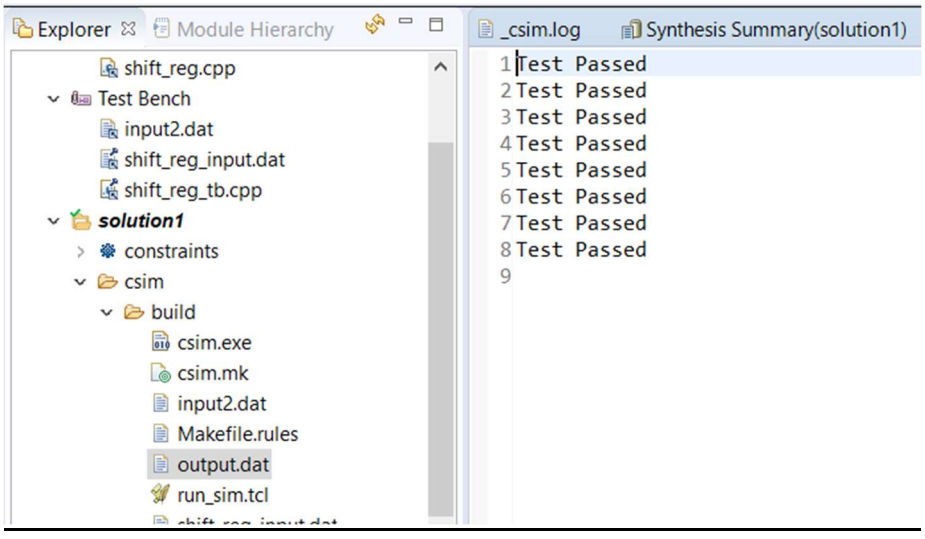
## Input File:




## C simulation printed output:




## Output File:



# Synthesis Report:

## Synthesis Summary Report of 'shift_register'

### General Information

Date: Thu Apr 6 11:22:21 2023
Version: 2022.2 (Build 3670227 on Oct 13 2022)
Project: project_25
Solution: solution1 (Vivado IP Flow Target)
Product family: zynq
Target device: xc7z020-clg484-1

### Timing Estimate

| Target | Estimated | Uncertainty |
|---|---|---|
| 10.00 ns | 4.604 ns | 2.70 ns |

### Performance & Resource Estimates

Modules Loops

| Modules & Loops | Issue Type | Violation Type | Distance | Slack | Latency(cycles) | Latency(ns) | Iteration Latency | Interval | Trip Count | Pipelined | BRAM | DSP | FF | LUT | URAM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| shift_register | | | | - | 31 | 310.000 | - | 32 | - | no | 0 | 0 | 512 | 2974 | 0 |

### Performance Pragma

| Modules & Loops | Target TI(cycles) | TI(cycles) | TI met |
|---|---|---|---|
| shift_register | - | - | - |

### HW Interfaces

#### AXIS

| Interface | Register Mode | TDATA | TREADY | TVALID |
|---|---|---|---|---|
| data_in | both | 16 | 1 | 1 |
| data_out | both | 16 | 1 | 1 |

#### Other Ports

| Interface | Mode | Bitwidth |
|---|---|---|
| shift_flag | ap_none | 1 |
| shift_value | ap_none | 16 |

#### 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 |
|---|---|---|
| data_in | in | stream<ap_uint<16> 0>& |
| shift_value | in | ap_uint<16> |
| shift_flag | in | bool |
| data_out | out | stream<ap_uint<16> 0>& |

#### SW-to-HW Mapping

| Argument | HW Interface | HW Type |
|---|---|---|
| data_in | data_in | interface |
| shift_value | shift_value | port |
| shift_flag | shift_flag | port |
| data_out | data_out | interface |

### Pragma Report

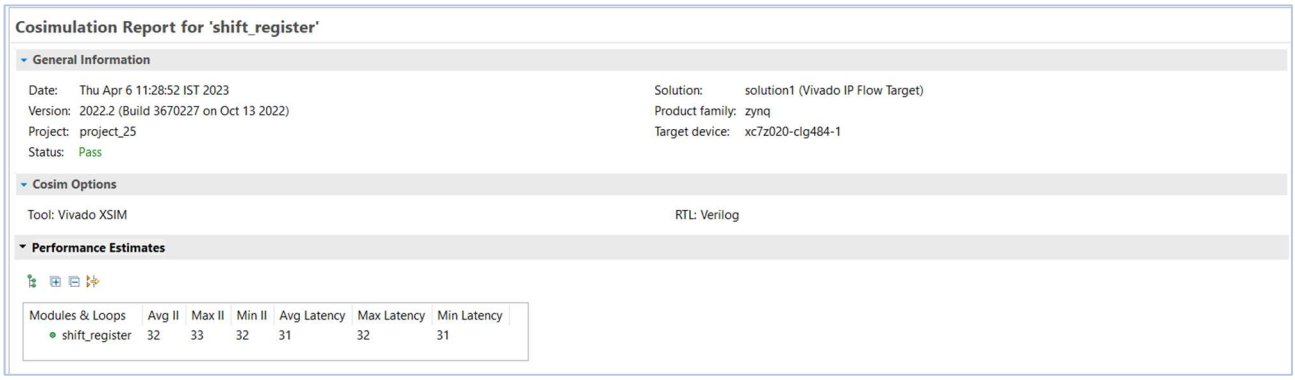
#### Valid Pragma Syntax

| Type | Options | Location | Function |
|---|---|---|---|
| interface | mode=axis port=data_out register | ../../../../OneDrive/Desktop/vitis/shift_reg.cpp:6 | shift_register |
| interface | mode=axis port=data_in register | ../../../../OneDrive/Desktop/vitis/shift_reg.cpp:7 | shift_register |
| array_partition | dim=1 factor=16 type=block variable=dout | ../../../../OneDrive/Desktop/vitis/shift_reg.cpp:10 | shift_register |
| array_partition | dim=1 factor=16 type=block variable=din | ../../../../OneDrive/Desktop/vitis/shift_reg.cpp:12 | shift_register |
| unroll | | ../../../../OneDrive/Desktop/vitis/shift_reg.cpp:15 | shift_register |
| unroll | | ../../../../OneDrive/Desktop/vitis/shift_reg.cpp:21 | shift_register |
| unroll | | ../../../../OneDrive/Desktop/vitis/shift_reg.cpp:32 | shift_register |
| unroll | | ../../../../OneDrive/Desktop/vitis/shift_reg.cpp:43 | shift_register |

## Cosimulation printed output:

```
Compiling module xil_defaultlib.apatb_shift_register_top
Compiling module work.glbl
Built simulation snapshot shift_register

****** 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/shift_register/xsim_script.tcl
# xsim {shift_register} -autoloadwcfg -tclbatch {shift_register.tcl}
Time resolution is 1 ps
source shift_register.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"
find kernel block.
// RTL Simulation : 1 / 8 [100.00%] @ "465000"
// RTL Simulation : 2 / 8 [100.00%] @ "785000"
// RTL Simulation : 3 / 8 [100.00%] @ "1105000"
// RTL Simulation : 4 / 8 [100.00%] @ "1425000"
// RTL Simulation : 5 / 8 [100.00%] @ "1745000"
// RTL Simulation : 6 / 8 [100.00%] @ "2065000"
// RTL Simulation : 7 / 8 [100.00%] @ "2385000"
// RTL Simulation : 8 / 8 [100.00%] @ "2705000"
////////////////////////////////////////////////////////////////////////////////
$finish called at time : 2765 ns : File "C:/Users/velic/AppData/Roaming/Xilinx/Vitis/project_25/solution1/sim/verilog/shift_register.autotb.v" Line 367
## quit
INFO: [Common 17-206] Exiting xsim at Thu Apr  6 11:28:45 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 16
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: 87.919 seconds; current allocated memory: 13.168 MB.
INFO: [HLS 200-112] Total CPU user time: 4 seconds. Total CPU system time: 2 seconds. Total elapsed time: 100.671 seconds; peak allocated memory: 109.113 MB.
Finished C/RTL cosimulation.
```

## Cosimulation Report:



**Cosimulation Report for 'shift_register'**

**General Information**

| | | | |
|---|---|---|---|
| Date: | Thu Apr 6 11:28:52 IST 2023 | Solution: | solution1 (Vivado IP Flow Target) |
| Version: | 2022.2 (Build 3670227 on Oct 13 2022) | Product family: | zynq |
| Project: | project_25 | 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 |
|---|---|---|---|---|---|---|
| shift_register | 32 | 33 | 32 | 31 | 32 | 31 |

Github:

https://github.com/velicharlagokulkumar/vitis-hls/tree/main/project_25