

Synthesis C++ target
C++ testbench

The screenshot displays the Vitis HLS 2020.2 IDE interface. The top menu bar includes File, Edit, Project, Solution, Window, and Help. The Explorer pane on the left shows the project structure for 'hks_hls_template', including 'Includes', 'Source' (with 'hks_hls_template.cpp' and 'hks_hls_template_tb.cpp'), 'Test Bench', 'solution1', 'constraints', 'directives.tcl', 'script.tcl', 'csim', 'impl', 'misc', 'verilog', and 'vhdl'. The Git Repositories pane shows the current branch as 'fe_fw [dev]' at the path '/home/braydo/Projects/fe_fw'. The main editor window shows the C++ testbench code in 'hks_hls_template_tb.cpp', which includes 'hks_hls_template.h' and implements a multi-trigger function. The Outline pane on the right shows the hierarchy of the 'hks_hls_template.h' file. The Console pane at the bottom displays the synthesis and simulation logs, indicating that the design was successfully synthesized and simulated using CSIM.

```
#include "hks_hls_template.h"

ap_uint<1> multi_trg(
    ap_uint<256> fadc_hits,
    ap_uint<9> multiplicity_thr
)
{
    ap_uint<9> hit_cnt = 0;
    for(int i=0; i<256; i++)
        hit_cnt += fadc_hits[i] ? 1 : 0;
    if(hit_cnt >= multiplicity_thr)
        return 1;
    return 0;
}

ap_uint<1> or_trg(
    ap_uint<256> fadc_hits
)
{
    return fadc_hits.or_reduce();
}

void hks_hls_template(
    ap_uint<9> multiplicity_thr,
    hls::stream<fadc_vxs_hits_t> &s_fadc_vxs_hits,
    hls::stream<trig_t> &s_trig
)
{
    #pragma HLS INTERFACE ap_fifo port=s_trig
    #pragma HLS INTERFACE ap_fifo port=s_fadc_vxs_hits
    #pragma HLS stable variable=multiplicity_thr
    #pragma HLS PIPELINE II=1 style=flp
    trig_t trig;
    fadc_vxs_hits_t fadc_vxs_hits = s_fadc_vxs_hits.read();

    // since we're running with 31.25MHz clock, but
    // have 4ns resolution data we must perform 8 iterations per clock cycle
    for(int t=0; t<8; t++)
    {
        // ...
    }
}
```

Vitis HLS Console

```
INFO: [HLS 200-1510] Running: create_clock -period 32 -name default
INFO: [HLS 200-1510] Running: set_directive_top -name hks_hls_template hks_hls_template
INFO: [HLS 200-1510] Running: csim_design -quiet
INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
s_trig_answer.size() = 100
s_trig.size() = 100
The maximum depth reached by any of the 3 hls::stream() instances in the design is 100
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
INFO: [HLS 200-111] Finished Command csim_design CPU user time: 0.08 seconds. CPU system time: 0.03 seconds. Elapsed time: 0.06 seconds; current allocated memory: 85.654 MB.
Finished C simulation.
```

Main functions to run:

- 1) C Synthesis
- 2) C Simulation
- 3) Co-Simulation

The screenshot displays the Vitis HLS 2020.2 interface. The left sidebar shows the project explorer with files like `hks_hls_template.cpp`, `hks_hls_template_tb.cpp`, and `hks_hls_template.h`. The main window shows the **Synthesis Summary Report of 'hks_hls_template'**.

General Information

Date:	Thu Nov 16 21:45:22 2023	Solution:	solution1 (Vivado IP Flow Target)
Version:	2020.2 (Build 3064766 on Wed Nov 18 09:12:47 MST 2020)	Product family:	virtex7
Project:	hks_hls_template	Target device:	xc7vx550t-ffg1927-1

Timing Estimate

Target	Estimated	Uncertainty
32.00 ns	17.762 ns	8.64 ns

Performance & Resource Estimates

Modules & Loops	Issue Type	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
hks_hls_template	-	-	0	0.0	-	1	-	yes(flp)	0	0	1	23444	0

HW Interfaces

AP_FIFO

Interface	Data Width
s_fadc_vxs_hits_V	2048
s_trig_V	48

REGISTER

Interface	Mode	Bitwidth
multiplicity_thr_ap_none		9

Console

16 Guidance-Infos 0 Guidance-Warnings 0 Guidance-Errors

Web Help

Name **Details**

SCHEDULE

- [HLS 200-1470]

RUNTIME

- [HLS 200-111]
- [HLS 200-111]

Pipelining result : Target II = 1, Final II = 1, Depth = 1, function 'hks_hls_template'

Finished Command cosim_design CPU user time: 32.12 seconds, CPU system time: 1.14 seconds, Elapsed time: 18.39 seconds; current allocated memory: 90.599 MB.

Finished File checks and directory preparation: CPU user time: 0 seconds, CPU system time: 0 seconds, Elapsed time: 0 seconds; current allocated memory: 85.528 MB.

Finished Source Code Analysis and Preprocessing: CPU user time: 1.75 seconds, CPU system time: 0.14 seconds, Elapsed time: 1.33 seconds; current allocated memory:

C-Synthesis – synthesizes VHDL/Verilog source files from C/C++ code

C-Synthesis result: latency(cycles) should be small (<30), Interval=1, BRAM<200, DSP<200, FF<200k, LUT<200k, URAM=0

The screenshot displays the Vitis HLS 2020.2 software interface. The main window shows the 'Synthesis Summary Report of 'hks_hls_template''. The report is divided into several sections: General Information, Timing Estimate, Performance & Resource Estimates, and HW Interfaces.

General Information

- Date: Thu Nov 16 21:45:22 2023
- Version: 2020.2 (Build 3064766 on Wed Nov 18 09:12:47 MST 2020)
- Project: hks_hls_template
- Solution: solution1 (Vivado IP Flow Target)
- Product family: virtex7
- Target device: xc7vx550t-ffg1927-1

Timing Estimate

Target	Estimated	Uncertainty
32.00 ns	17.762 ns	8.64 ns

Performance & Resource Estimates

Modules & Loops	Issue Type	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
hks_hls_template		-	0	0.0	-	1	-	yes(flp)	0	0	1	23444	0

HW Interfaces

AP_FIFO

Interface	Data Width
s_fadc_vxs_hits_V	2048
s_trig_V	48

REGISTER

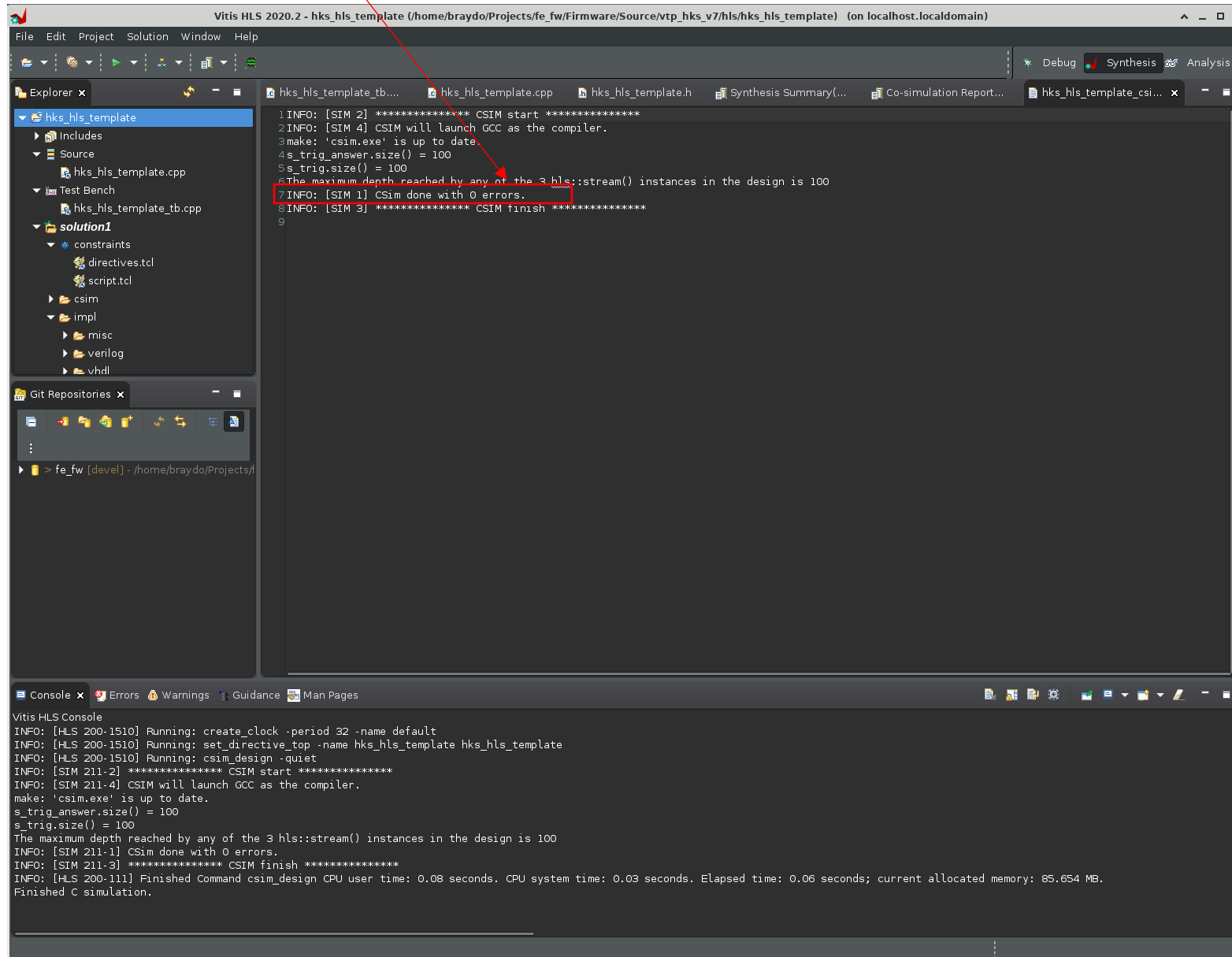
Interface	Mode	Bitwidth
multiplycty_thr_ap_none		9

Console

```
Vitis HLS Console
INFO: [HLS 200-1510] Running: create_clock -period 32 -name default
INFO: [HLS 200-1510] Running: set_directive_top -name hks_hls_template hks_hls_template
INFO: [HLS 200-1510] Running: csim_design -quiet
INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
s_trig_answer.size() = 100
s_trig.size() = 100
The maximum depth reached by any of the 3 hls::stream() instances in the design is 100
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
INFO: [HLS 200-111] Finished Command csim_design CPU user time: 0.08 seconds. CPU system time: 0.03 seconds. Elapsed time: 0.06 seconds; current allocated memory: 85.654 MB.
Finished C simulation.
```

C-Simulation – Verify C/C++ code works by feeding it data and checking result

C-Simulation result: no errors (based on hks_hls_template_tb.cpp main()) returning value of 0



The screenshot displays the Vitis HLS 2020.2 IDE interface. The top toolbar includes buttons for File, Edit, Project, Solution, Window, and Help. The Explorer pane on the left shows the project structure for 'hks_hls_template', including 'Includes', 'Source' (containing 'hks_hls_template.cpp'), 'Test Bench' (containing 'hks_hls_template_tb.cpp'), and 'solution1' (containing 'constraints', 'directives.tcl', 'script.tcl', 'csim', 'impl', 'misc', 'verilog', and 'vhdl'). The Git Repositories pane at the bottom left shows the current directory as '/home/braydo/Projects/fe_fw'. The main editor window displays the 'hks_hls_template_tb.cpp' file, which contains the following code:

```
1 INFO: [SIM 2] ***** CSIM start *****
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3 make: 'csim.exe' is up to date.
4 s_trig_answer.size() = 100
5 s_trig.size() = 100
6 The maximum depth reached by any of the 3 hls::stream() instances in the design is 100
7 INFO: [SIM 1] CSim done with 0 errors.
8 INFO: [SIM 3] ***** CSIM finish *****
9
```

The console output at the bottom shows the execution of the C-simulation:

```
Vitis HLS Console
INFO: [HLS 200-1510] Running: create_clock -period 32 -name default
INFO: [HLS 200-1510] Running: set_directive_top -name hks_hls_template hks_hls_template
INFO: [HLS 200-1510] Running: csim_design -quiet
INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
s_trig_answer.size() = 100
s_trig.size() = 100
The maximum depth reached by any of the 3 hls::stream() instances in the design is 100
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
INFO: [HLS 200-111] Finished Command csim_design CPU user time: 0.08 seconds. CPU system time: 0.03 seconds. Elapsed time: 0.06 seconds; current allocated memory: 85.654 MB.
Finished C simulation.
```

Co-Simulation – Simulates HDL solution and verifies it with C++ testbench input/output check. Waveform can be viewed as shown to see digital waveforms and inspect inside the HLS VHDL/Verilog solution if needed

The image displays the Vivado 2020.2 interface for a co-simulation project named 'hks_hls_template'. The main window shows the 'Cosimulation Report for 'hks_hls_template'', which includes general information, cosim options, and performance estimates.

Cosimulation Report for 'hks_hls_template'

General Information

- Date: Thu Nov 16 21:46:34 EST 2023
- Version: 2020.2 (Build 3064766 on Wed Nov 18 09:12:47 MST 2020)
- Project: hks_hls_template
- Status: **Pass**
- Solution: solution1 (Vivado IP Flow Target)
- Product family: virtex7
- Target device: xc7vx550t-ffg1927-1

Cosim Options

- Tool: Vivado XSIM
- Dump Trace: port

Performance Estimates

The report also lists modules and performance metrics (Avg II, Max II, Min II, Avg Latency).

The bottom panel shows the 'Vitis HLS Console' with the following output:

```
INFO: [HLS 200-1510] Running: create_clock -period 32 -name default
INFO: [HLS 200-1510] Running: set_directive_top -name hks_hls_template hks_hls_template
INFO: [HLS 200-1510] Running: csim_design -quiet
INFO: [SIM 211-2] ***** CSIM start *****
INFO: [SIM 211-4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
s_trig_answer.size() = 100
s_trig.size() = 100
The maximum depth reached by any of the 3 hls::stream() instances in the design is 100
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
INFO: [HLS 200-111] Finished Command csim_design CPU user time: 0.08 seconds. CPU system time: 0.0
Finished C simulation.
```

The right panel shows the 'hks_hls_template.wcfg' waveform, displaying digital signals over time (0.000 ns to 2.000 ns). The signals include:

- C Outputs:** s_trig(fifo), s_trig_V_write, s_trig_V_full_n, s_trig_V_din[47:0]
- C Inputs:** s_fadc_vws_hits(fifo), s_fadc_vws_hits_V_read, s_fadc_vws_hits_V_empty_n, s_fadc_vws_hits_V_dout[2047:0]
- multiply_thr(wire):** multiply_thr[8:0]
- Block-level IO Handshake:** Reset
- Clock:** ap_clk

The waveform shows the timing of these signals, with the clock (ap_clk) running at 3536 ns.