Courses @ NECST

Xilinx Vivado HLS
18/01/2018

Lorenzo Di Tucci <lorenzo.ditucci@polimi.it>
Emanuele Del Sozzo <emanuele.delsozzo@polimi.it>
Marco D. Santambrogio <marco.santambrogio@polimi.it>

Agenda

- Introduction to the Hardware Design Flow
- Vivado HLS:
  - Design Flow
  - Kernel Creation
  - Communication Infrastructure
  - Kernel Optimizations
- Hands-On Example

Installation Party

Use this Google Doc to provide your data

https://goo.gl/FRCG6y

First, install the VPN we have provided you.
(Mac: Tunnelblick - Windows/Linux: OpenVPN)

To **SSH** to the machine:
ssh <name>.<surname>@nags31.local.necst.it
password: *user*

Installation Party

You can change your password here:

http://changepassword.local.necst.it/

You can also **RDP** to the instance using

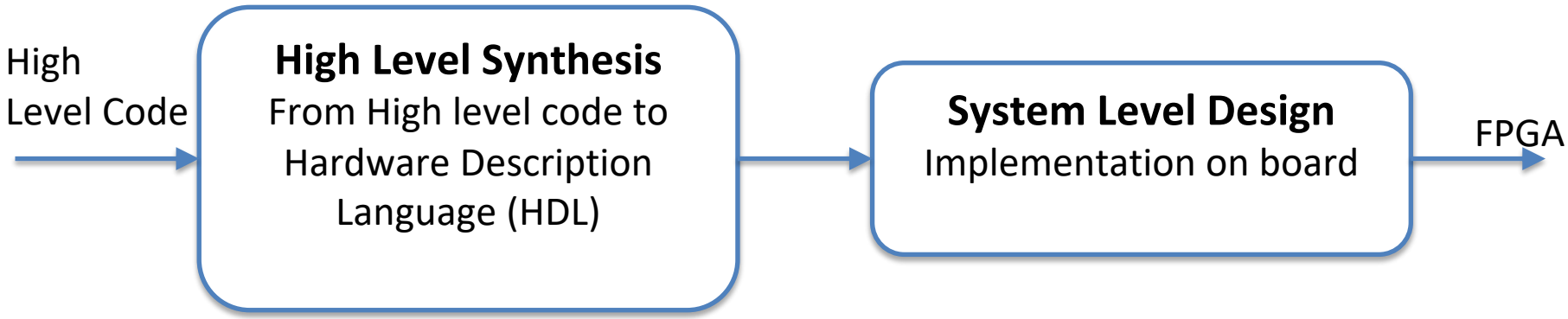
- Microsoft Remote Desktop (Microsoft/Mac OS)
- Remmina (Linux)

To connect to the machine, or change your password you **must** have started the VPN.

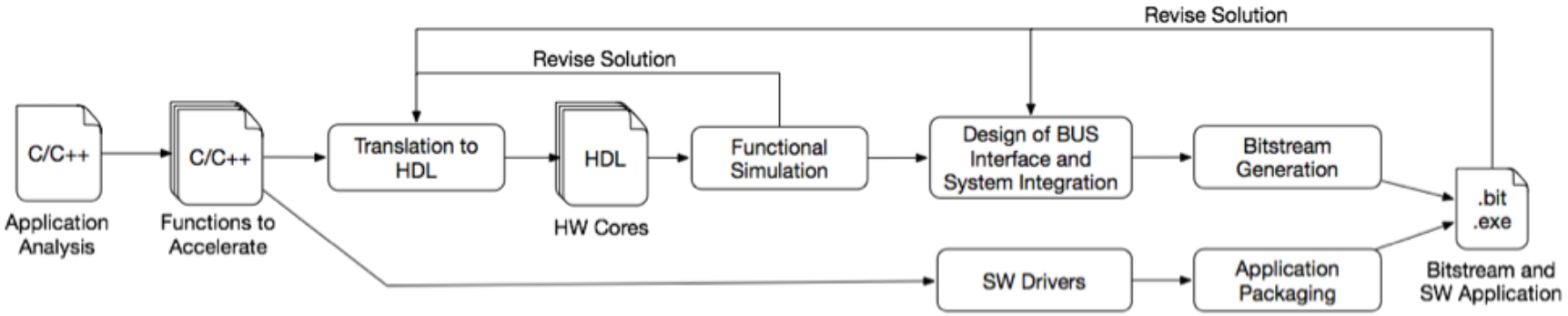
Hardware Design Flow for HPC

- *Hardware Design Flow (HDF)*: process to realize a hardware module

- HDF for FPGAs can be seen as a 2 step process

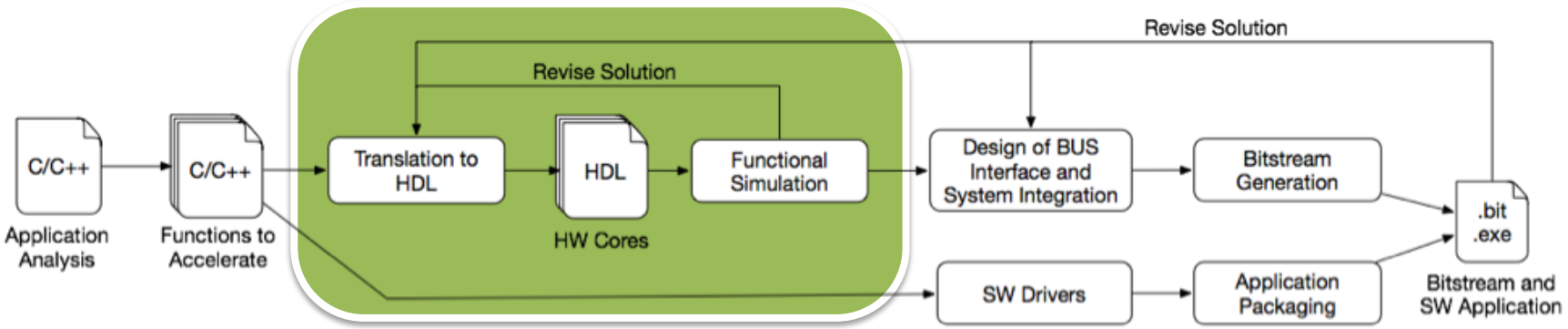


The Hardware Design Flow



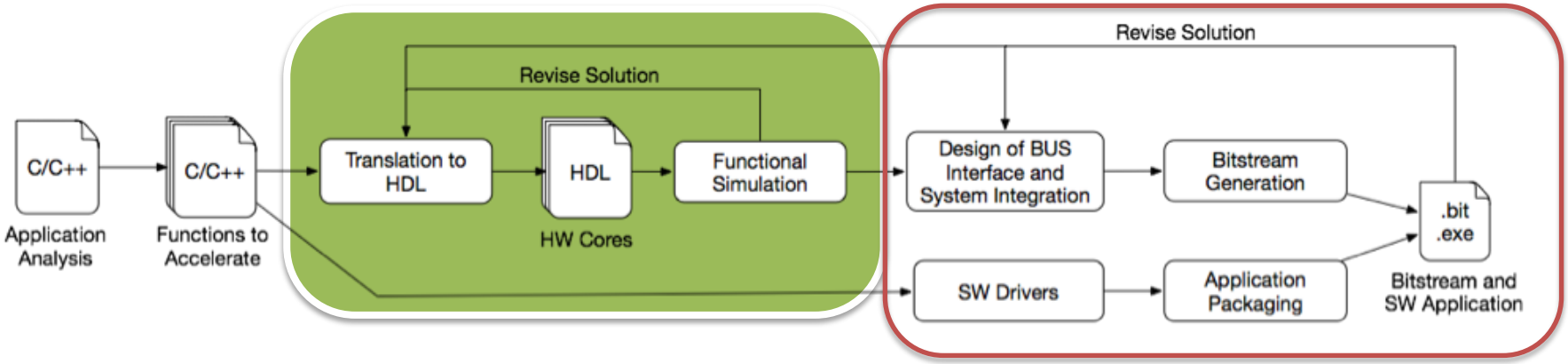
The Hardware Design Flow

- CAD tools aims at alleviating the design on FPGA



The Hardware Design Flow

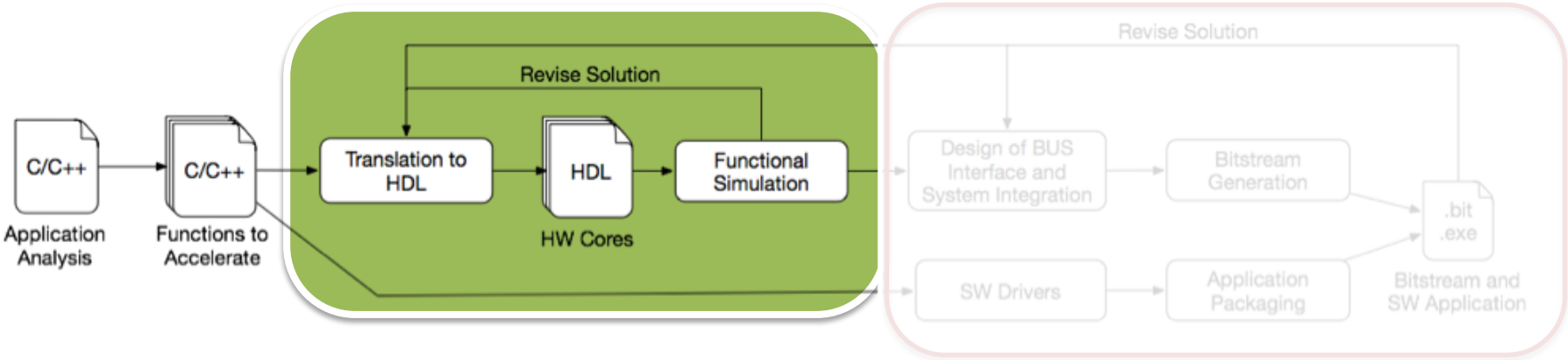
- CAD tools aims at alleviating the design on FPGA



- Designer still need to manually perform system integration, driver generation and runtime management

The Hardware Design Flow

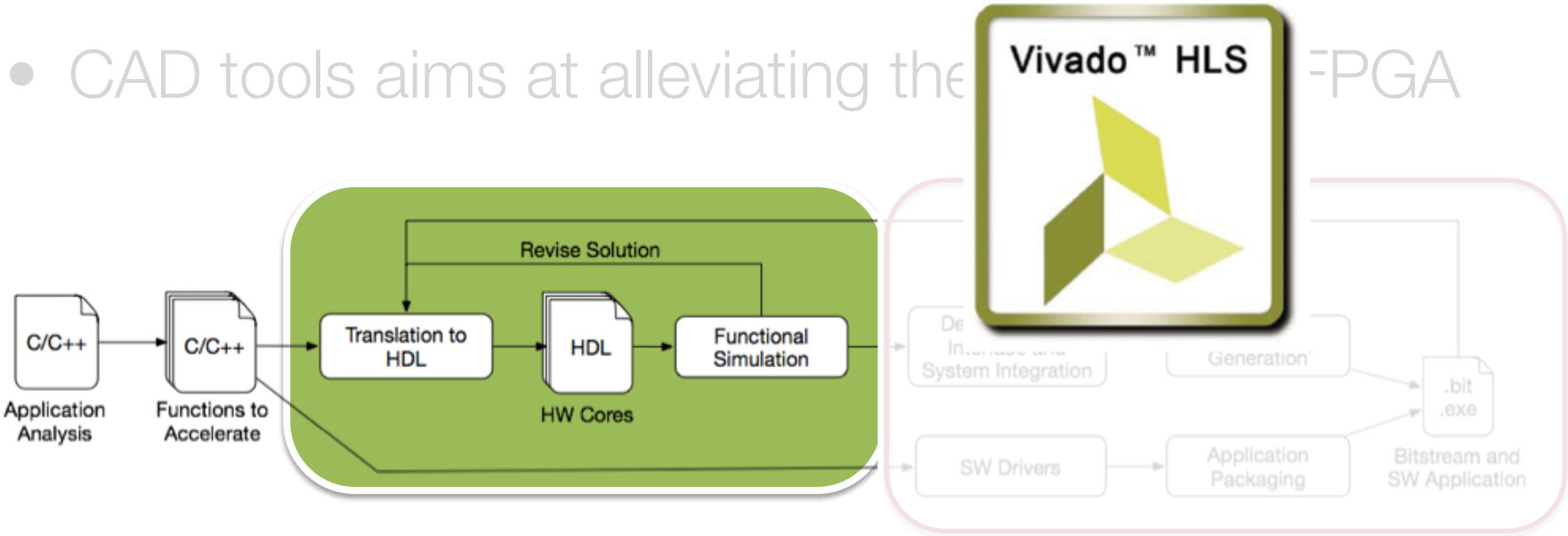
- CAD tools aims at alleviating the design on FPGA



- Designer still need to manually perform system integration, driver generation and runtime management

The Hardware Design Flow

- CAD tools aims at alleviating the [illegible] FPGA



- Designer still need to manually perform system integration, driver generation and runtime management

Vivado HLS

- High Level Synthesis tools
- provided a high level description of a kernel, allows to generate a HDL of the application
- increase the level of **abstraction** to ease the designing process
- provides multiple **hardware optimized** libraries and APIs
- **directives driven architecture-aware** synthesis

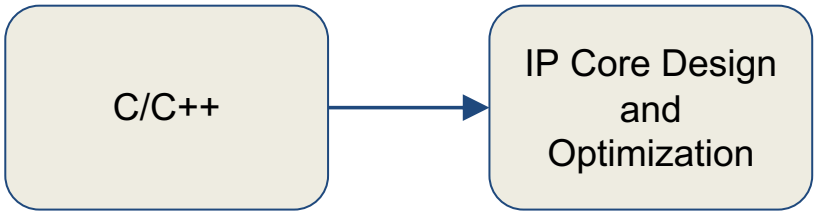
Vivado HLS Flow

C/C++

Possibility to specify the **kernel** using High Level Code

- code must be refactored as a **subset of C/C++** is supported
- **no dynamic memory allocation**
- Multiple **API** provided (arbitrary precision, mathematical functions, video library, etc..)
- Possibility to specify a constraint file (clock uncertainty, clock period, target board)

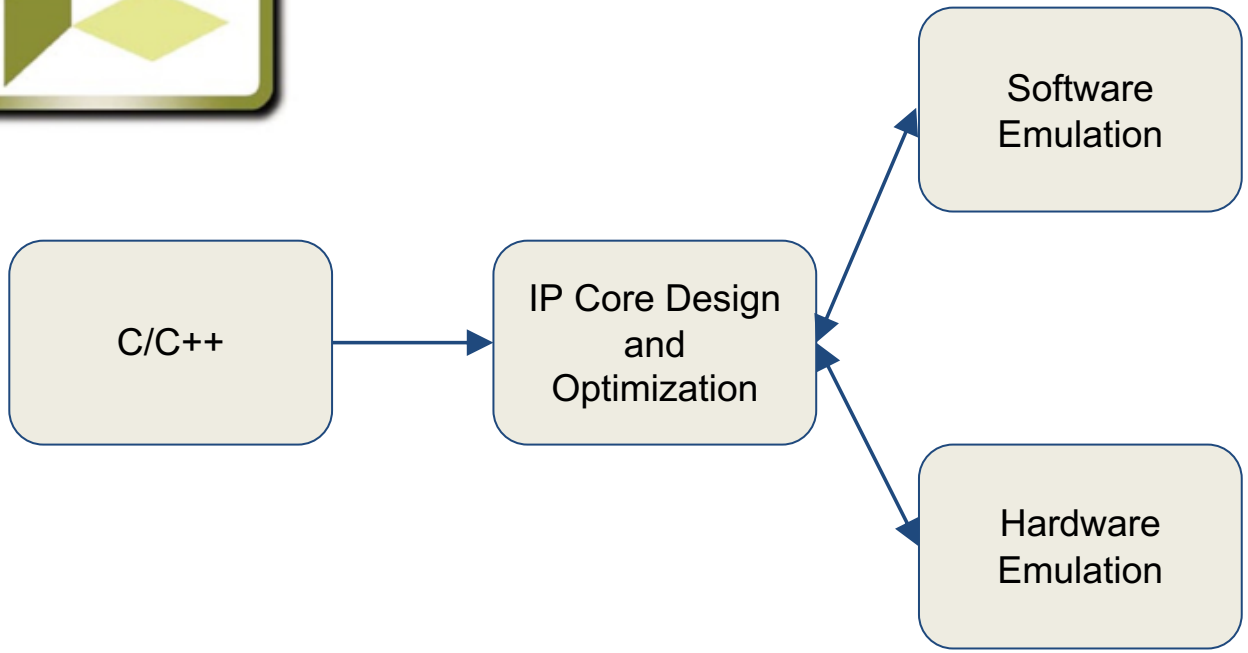
Vivado HLS Flow



Directives driven design and optimization

- necessary to specify interconnection with memory (AXI4, AXI4Lite, AXI4-Stream)
- The designer still needs to have clear in mind what architecture to implement

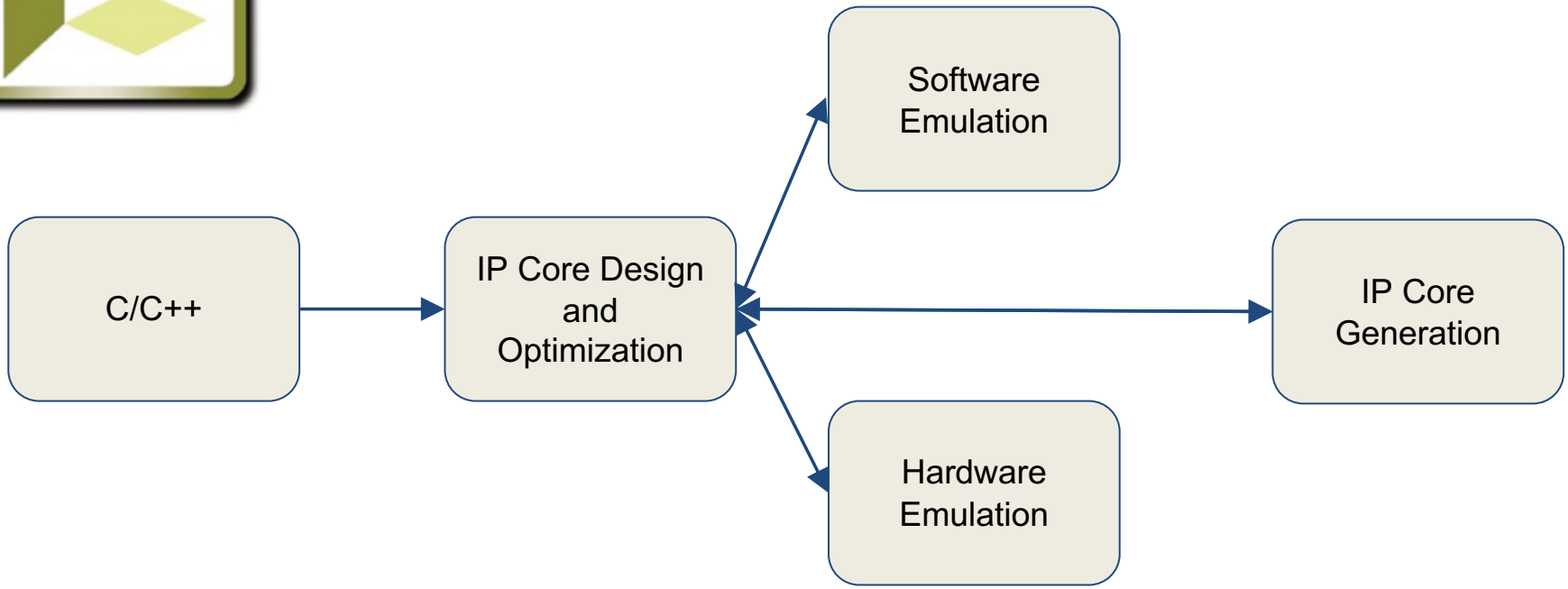
Vivado HLS Flow



2 levels of emulation

- **Software Emulation** used to check *functional correctness* of the application - does not guarantee correctness on the FPGA
- **Hardware Emulation** check the correctness of the logic generated by the synthesizer

Vivado HLS Flow



- Eventually, IP Core can be generated and exported so that it can be used for the **System Level Design** Step.
- At each step of the pipeline, it is possible to come back to the optimization phase

Kernel Creation

- Not all the parts of an algorithm are suitable for hardware acceleration
- First, it is necessary to **profile** the application to identify the bottlenecks of the application, i.e. the most compute intensive parts/kernels
- Then, it is possible to start adapting and optimizing the kernels for the High Level Synthesis process
  - define the communication infrastructure
  - build the kernel architecture

Communication Infrastructure

- Mainly, 3 types of communication may be implemented using AXI4 protocol:
  - AXI-Lite (for control signals)
  - Axi-Stream (to stream data)
  - Axi-Master (direct connection with the memory)

- Vivado HLS provides directives to specify which type of communication to implement

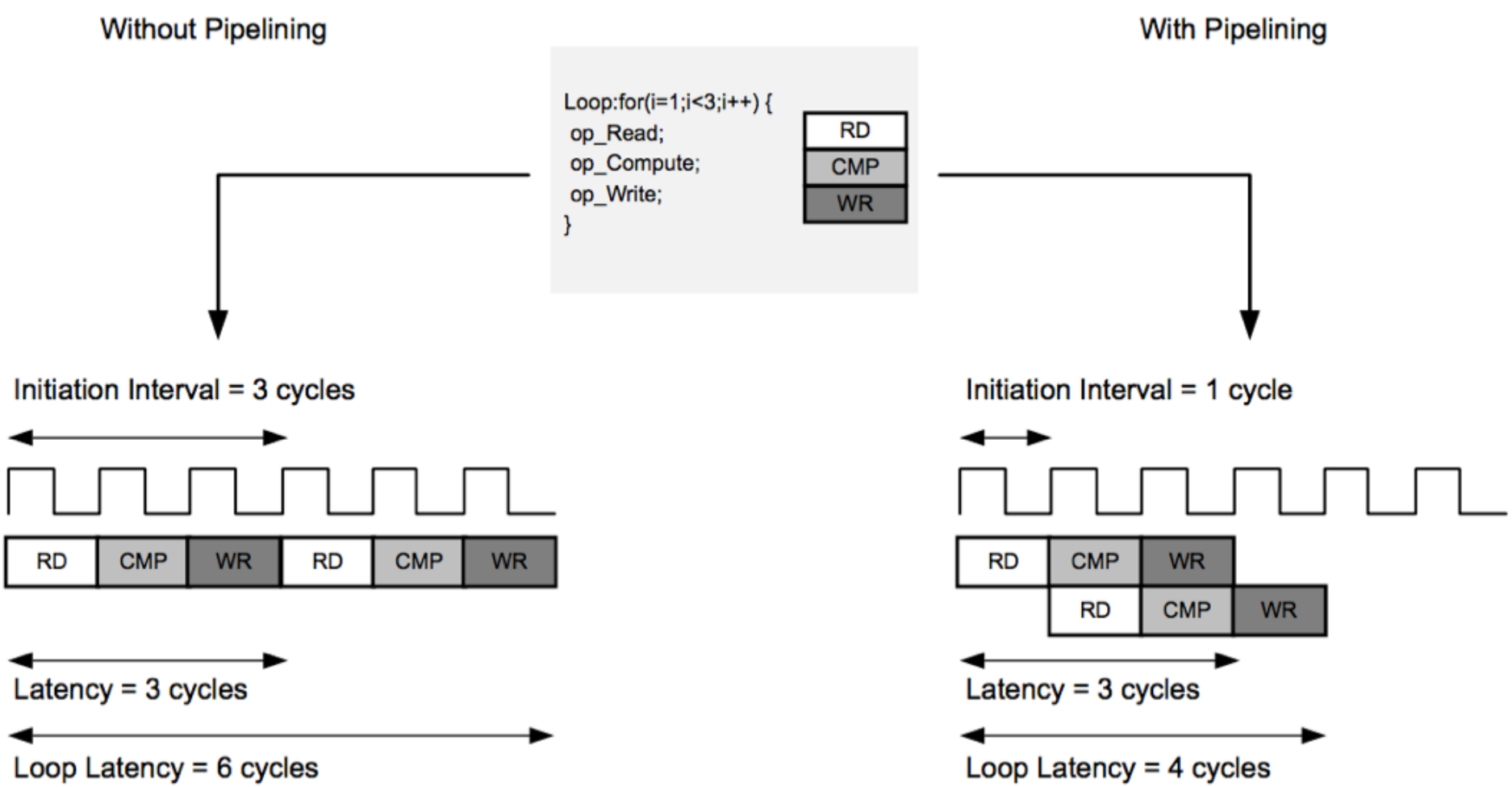
- However, the user has to develop the data transfer in an **efficient** way

Kernel Optimizations

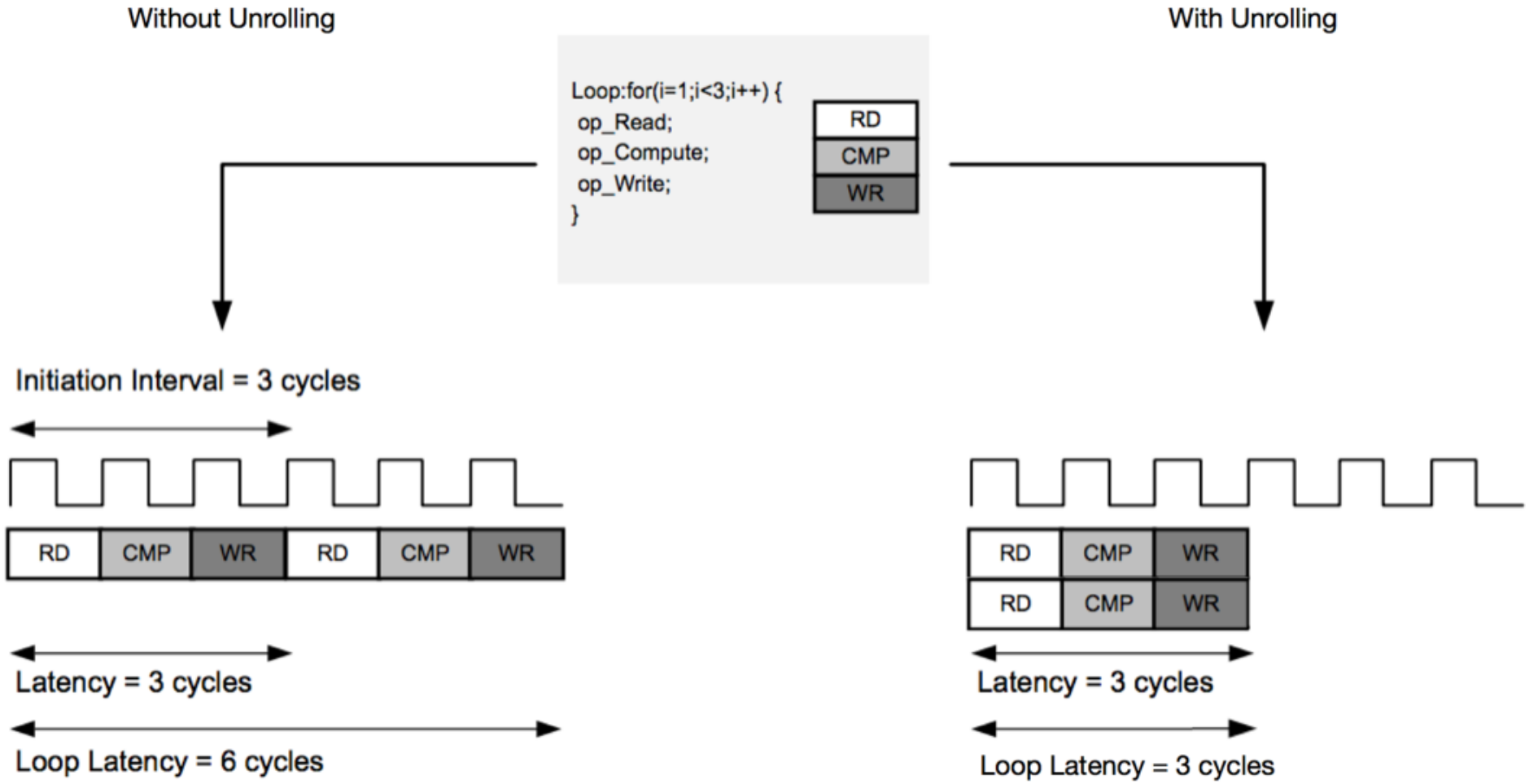
- Vivado HLS provides directives to optimize the kernel

- The directives may refer to:
  - how to carry out the computation
    - loop pipelining/unrolling
    - function inlining
    - dataflow
    - resources to use for a certain computation
  - how to store the data on FPGA memory
    - array partitioning (block, cyclic, complete)

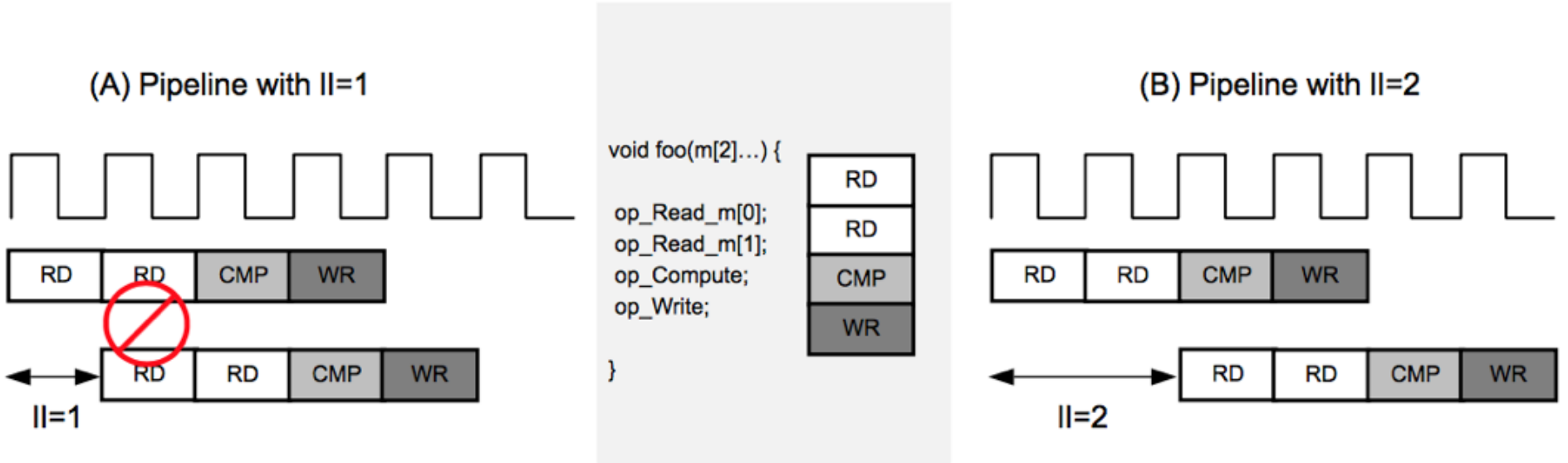
Loop Pipelining



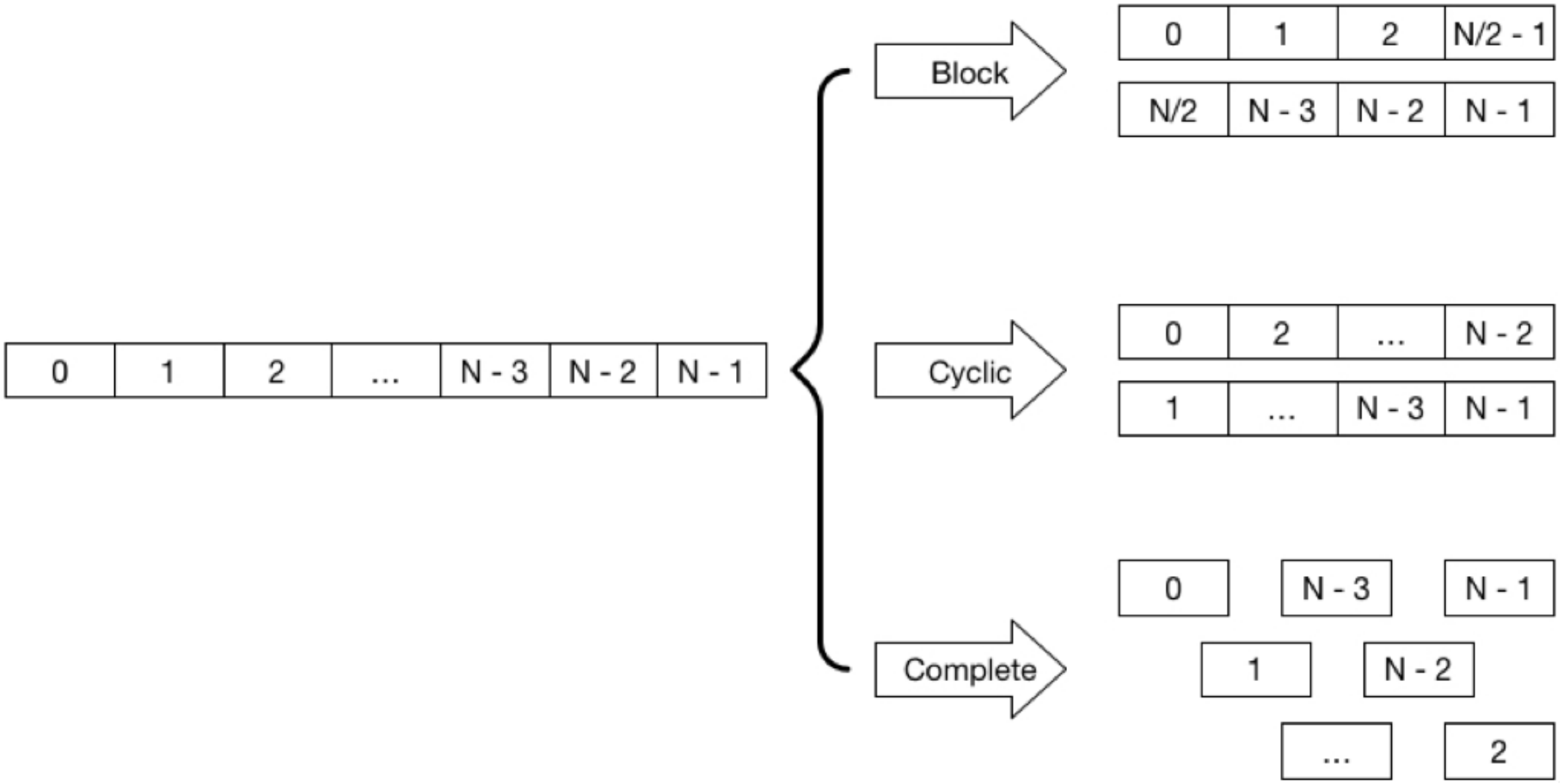
Loop Unrolling



Resource Contention



Array Partitioning



Example

- Now we are going to see how to implement a vector addition of this form:

  $$\mathbf{a} = \mathbf{b} + \mathbf{c}*d$$

  where **a**, **b**, **c** are vectors, and d is a constant

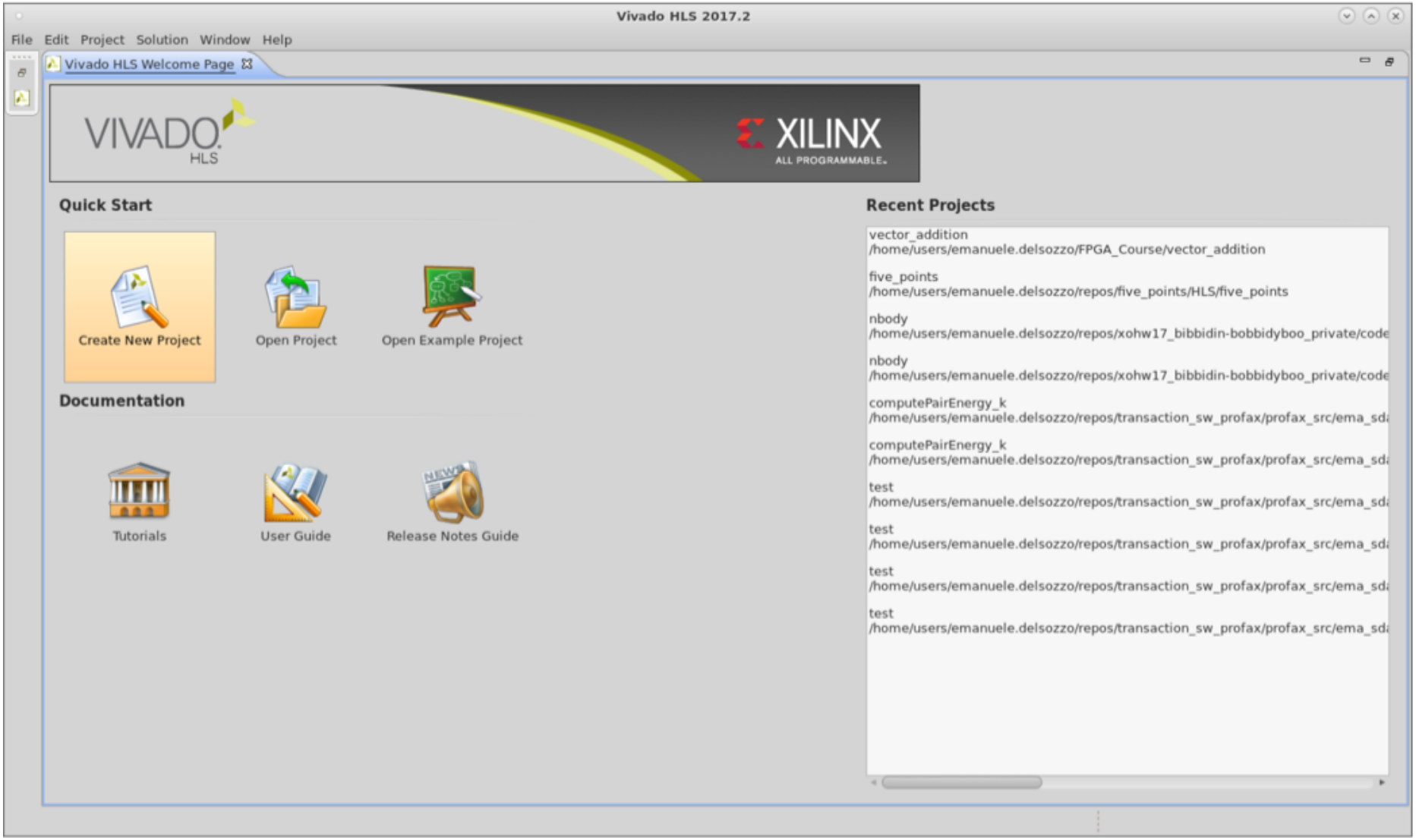
- We will go through all optimization steps with Vivado HLS, from the project creation to the IP core generation

Launch Vivado HLS

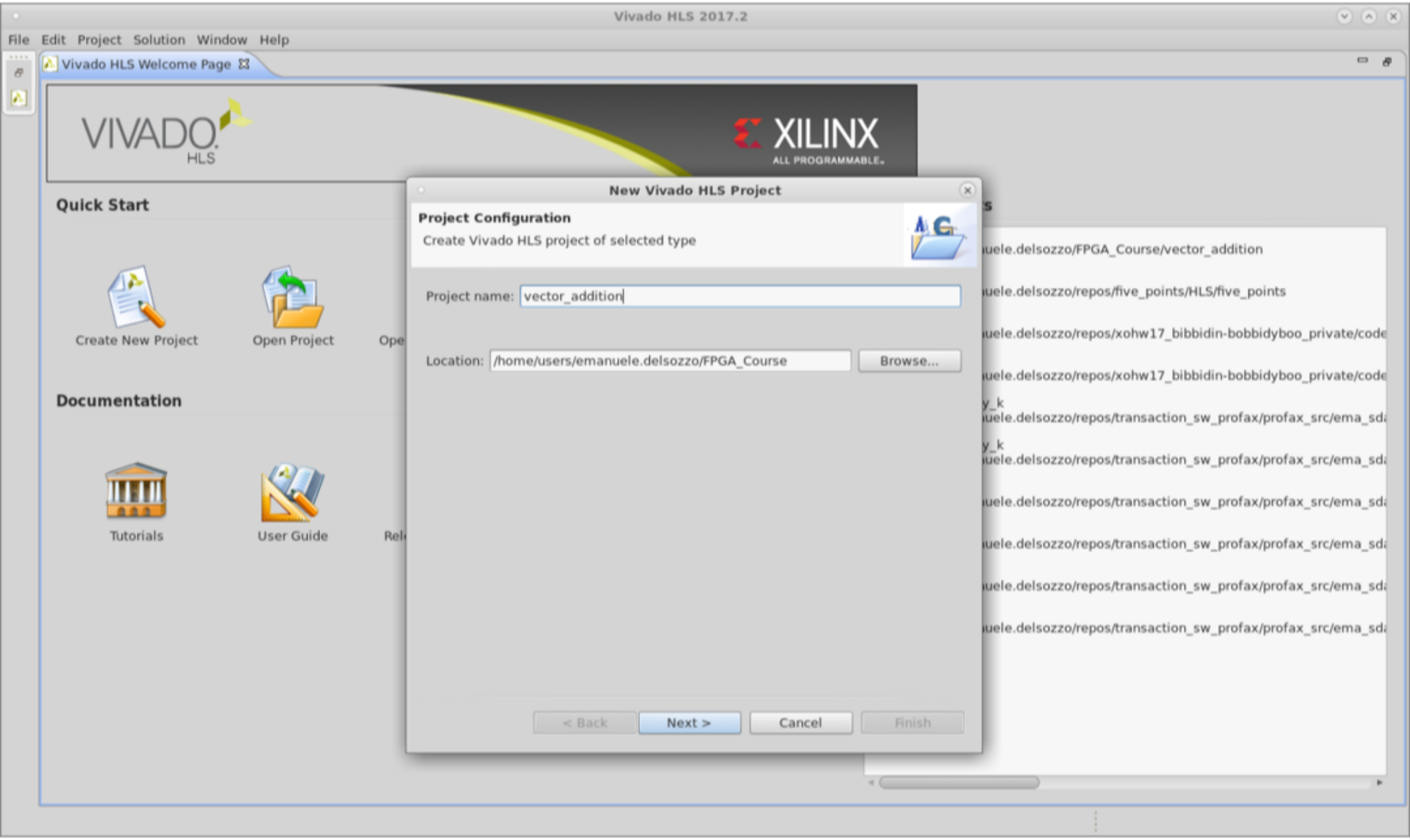
Source *settings64.sh* in Vivado folder, then run Vivado HLS

```
[emanuele.delsozzo@NAGS30 FPGA_Course]$ source /xilinx/software/Vivado/2017.2/settings64.sh
[emanuele.delsozzo@NAGS30 FPGA_Course]$ vivado_hls
****** Vivado(TM) HLS - High-Level Synthesis from C, C++ and SystemC
  Version 2017.2
  Build 1909853 on Thu Jun 15 18:55:24 MDT 2017
  Copyright (C) 1986-2017 Xilinx, Inc. All Rights Reserved.
****** 
INFO: [HLS 200-10] Running '/xilinx/software/Vivado_HLS/2017.2/bin/unwrapped/lnx64.o/vivado_hls'
INFO: [HLS 200-10] For user 'emanuele.delsozzo' on host 'NAGS30' (Linux_x86_64 version 3.10.0-514.
INFO: [HLS 200-10] On os "CentOS Linux release 7.3.1611 (Core) "
INFO: [HLS 200-10] In directory '/home/users/emanuele.delsozzo/FPGA_Course'
INFO: [HLS 200-10] Bringing up Vivado HLS GUI ...
```

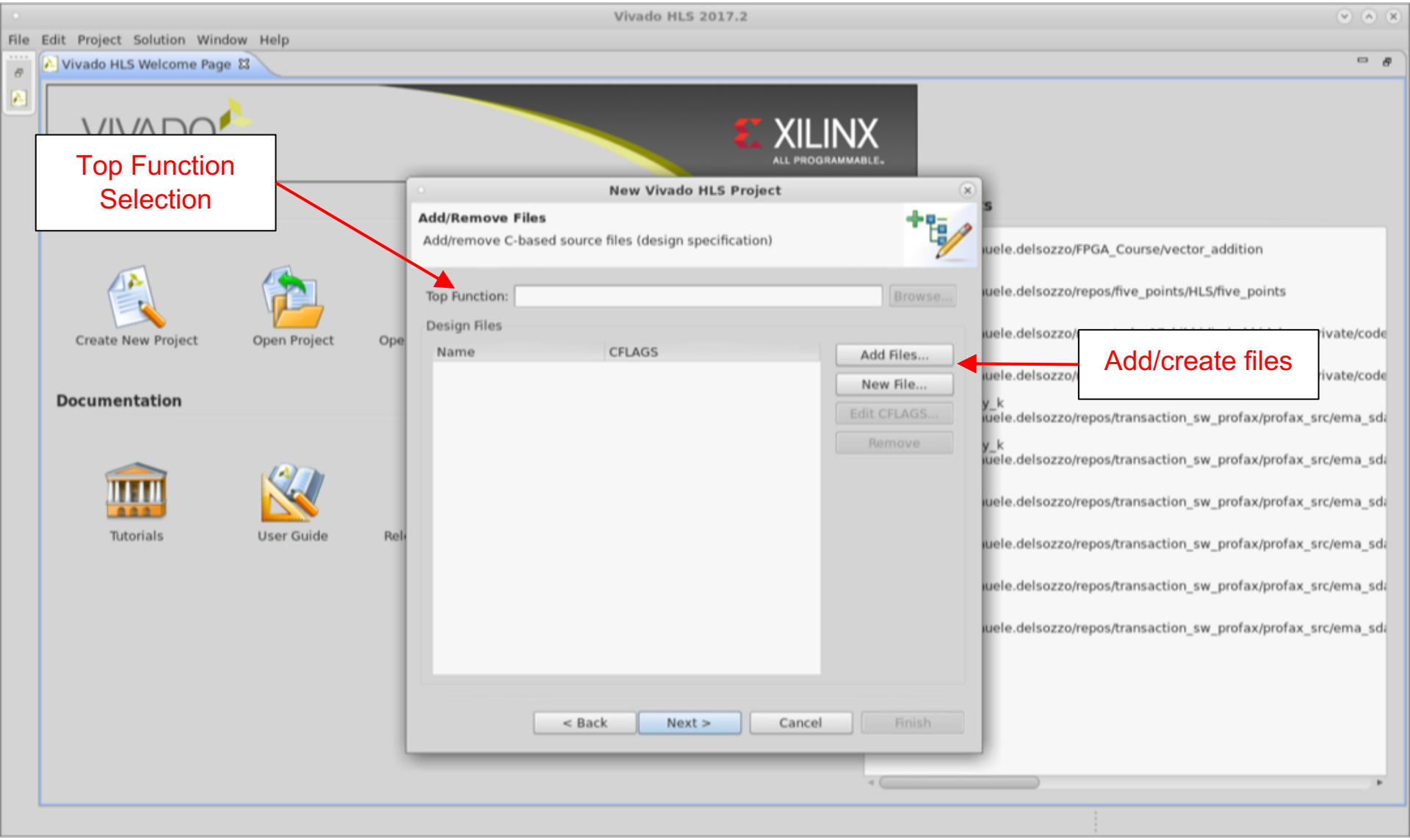
Vivado HLS GUI



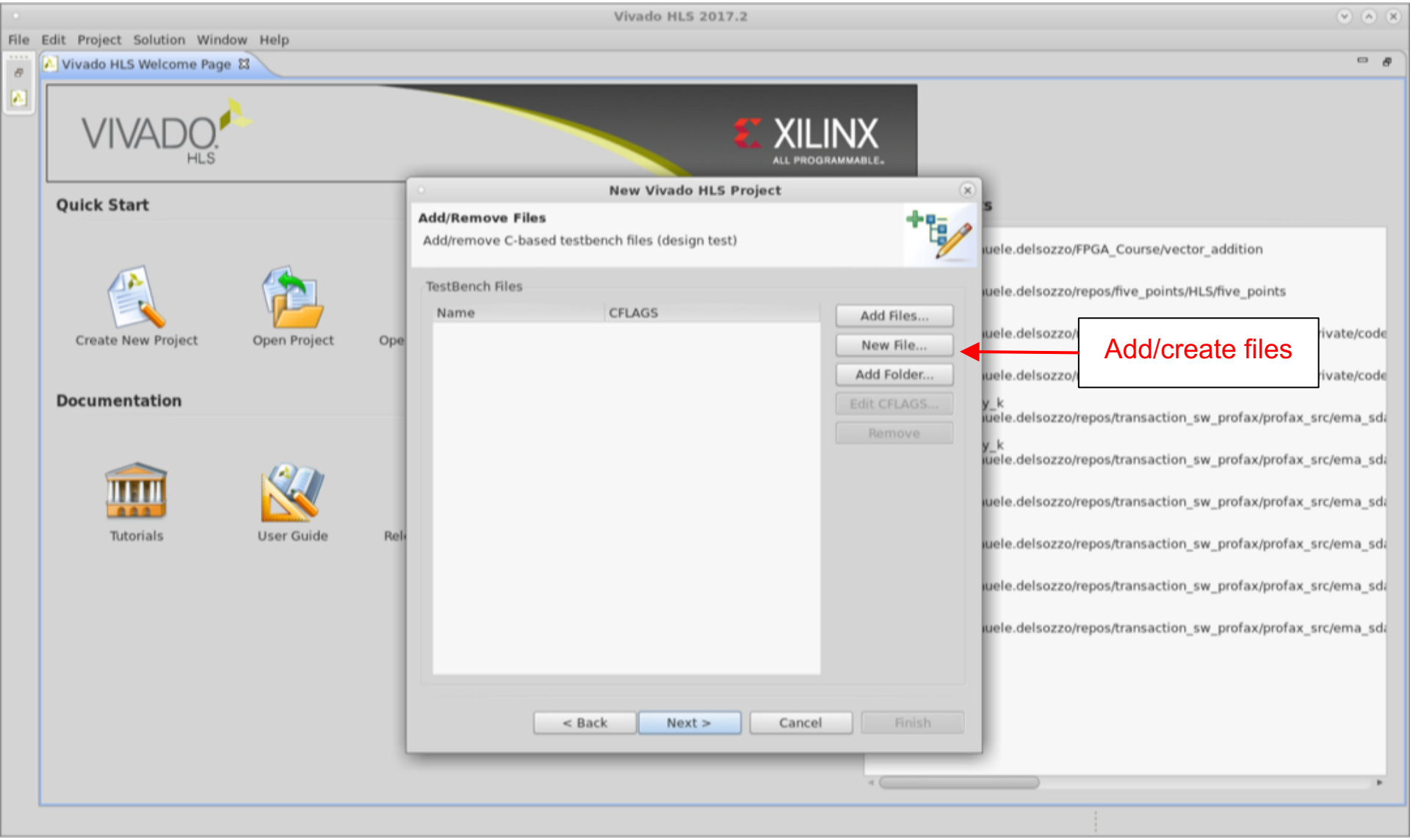
Project Configuration



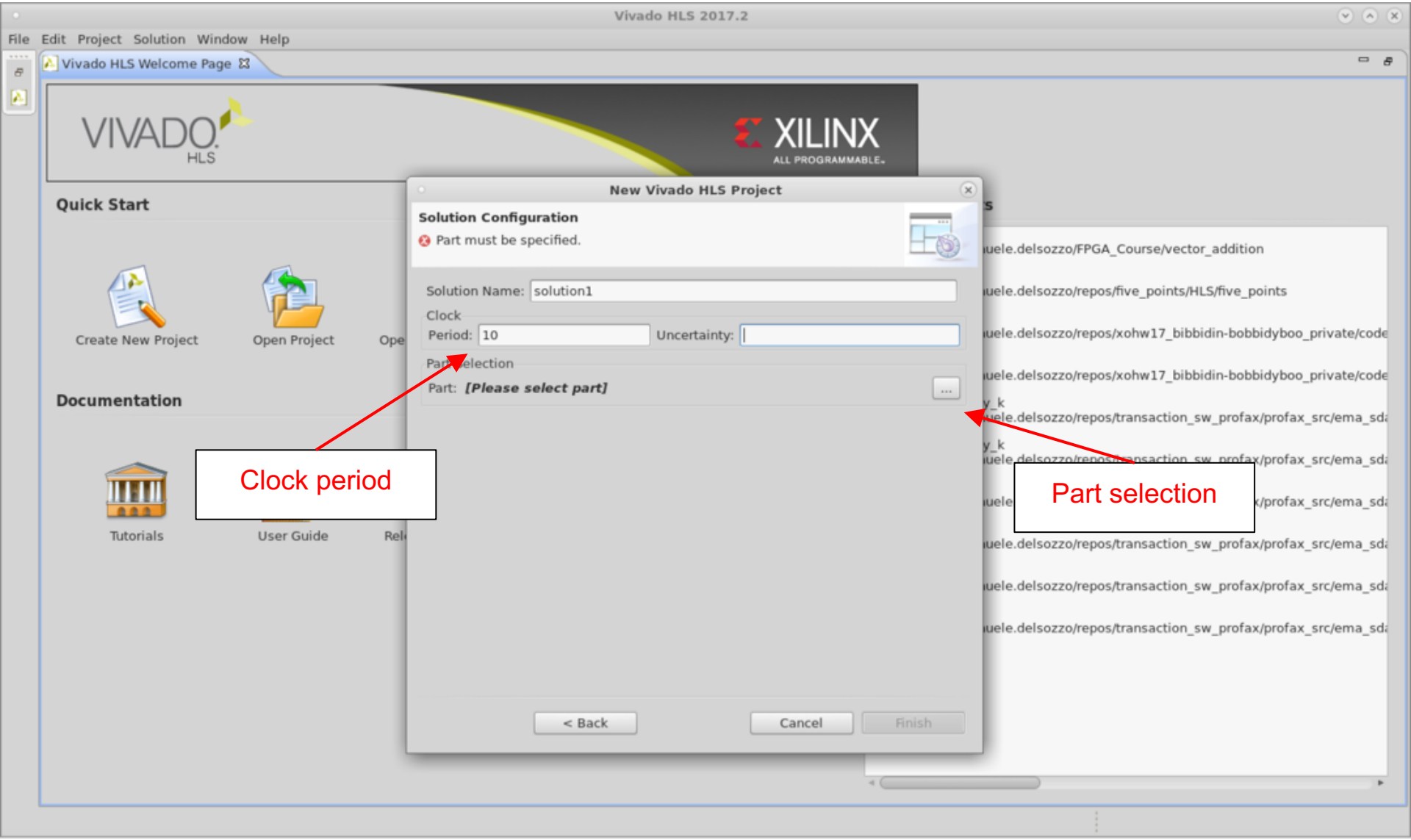
Kernel files



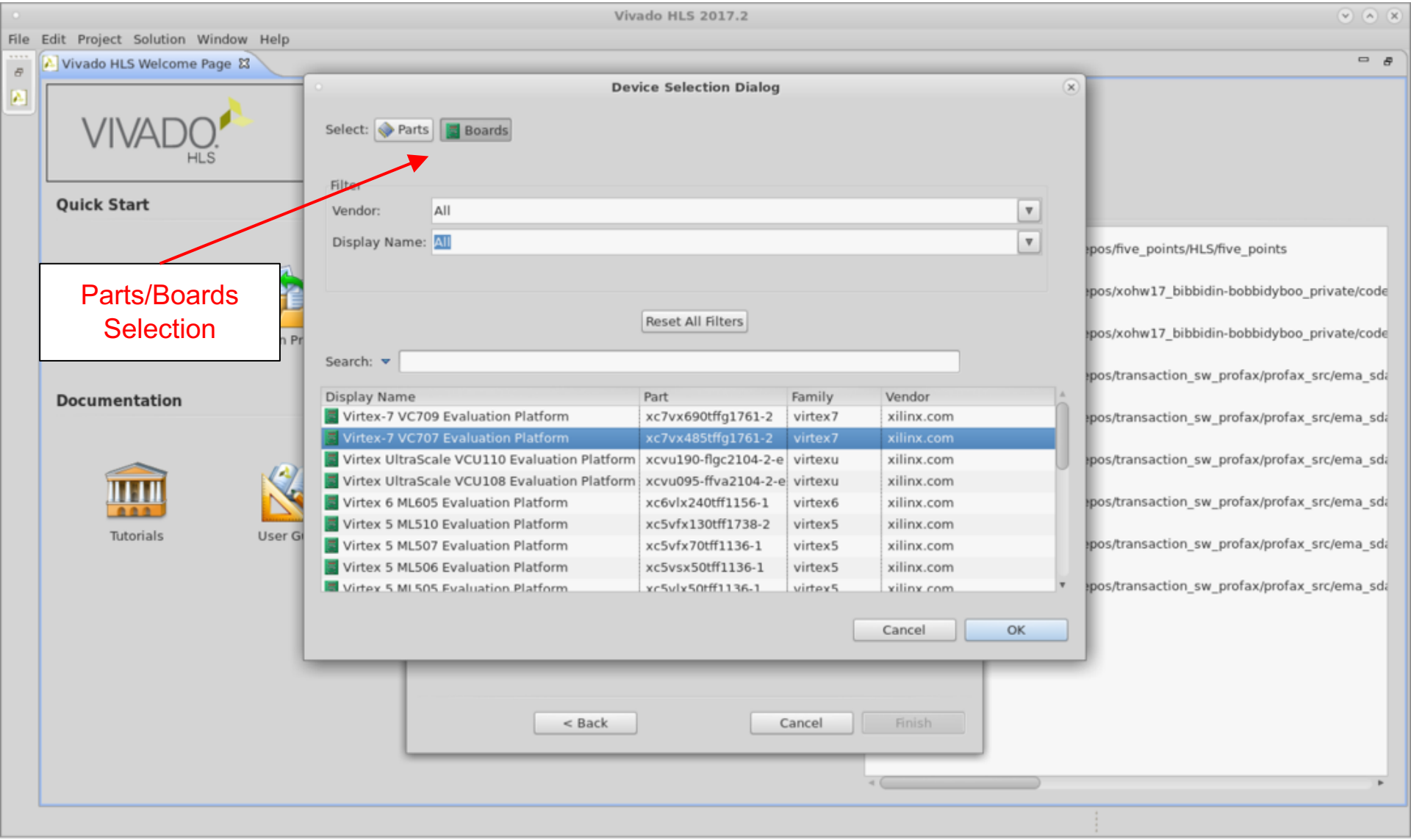
Testbench files



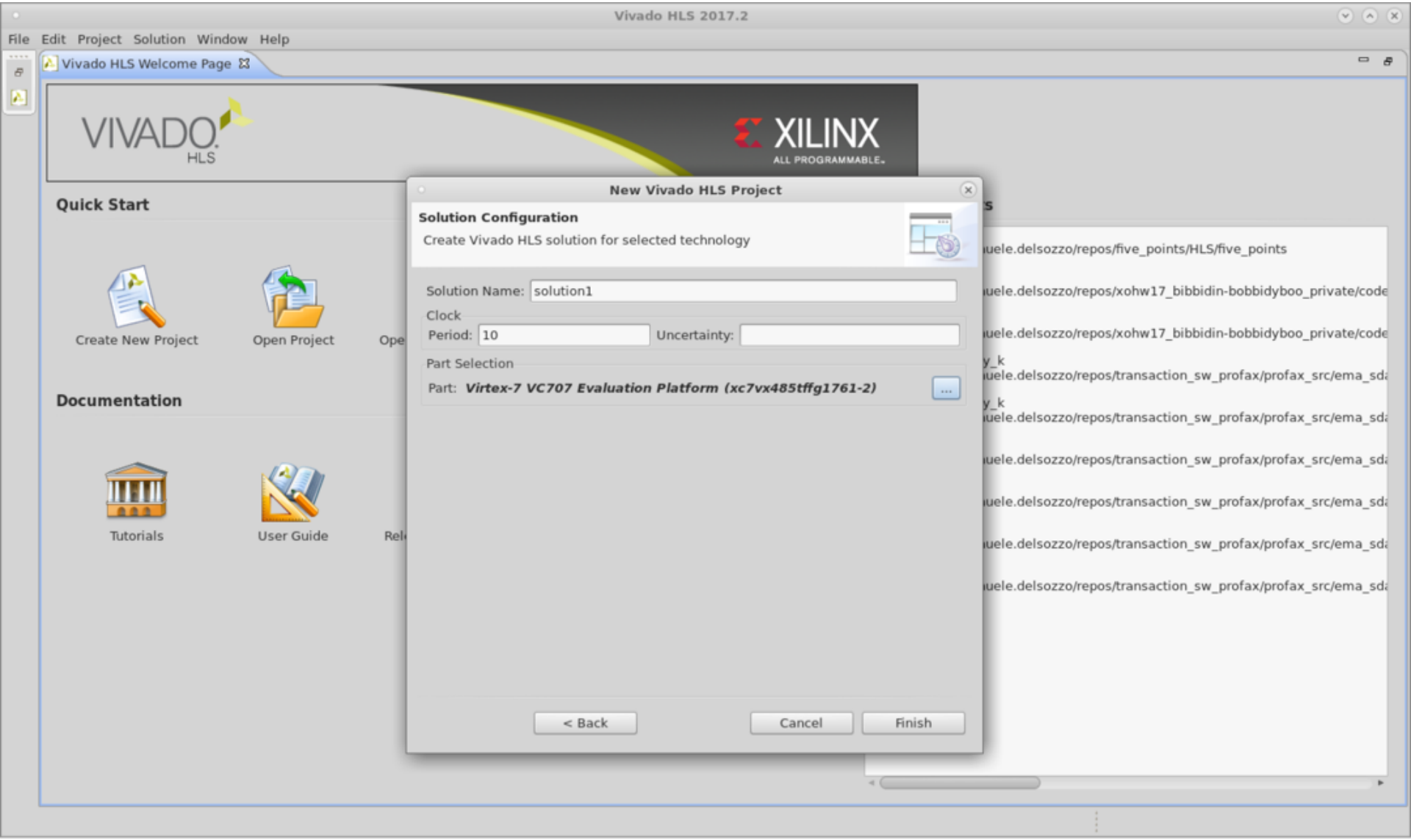
Solution Configuration



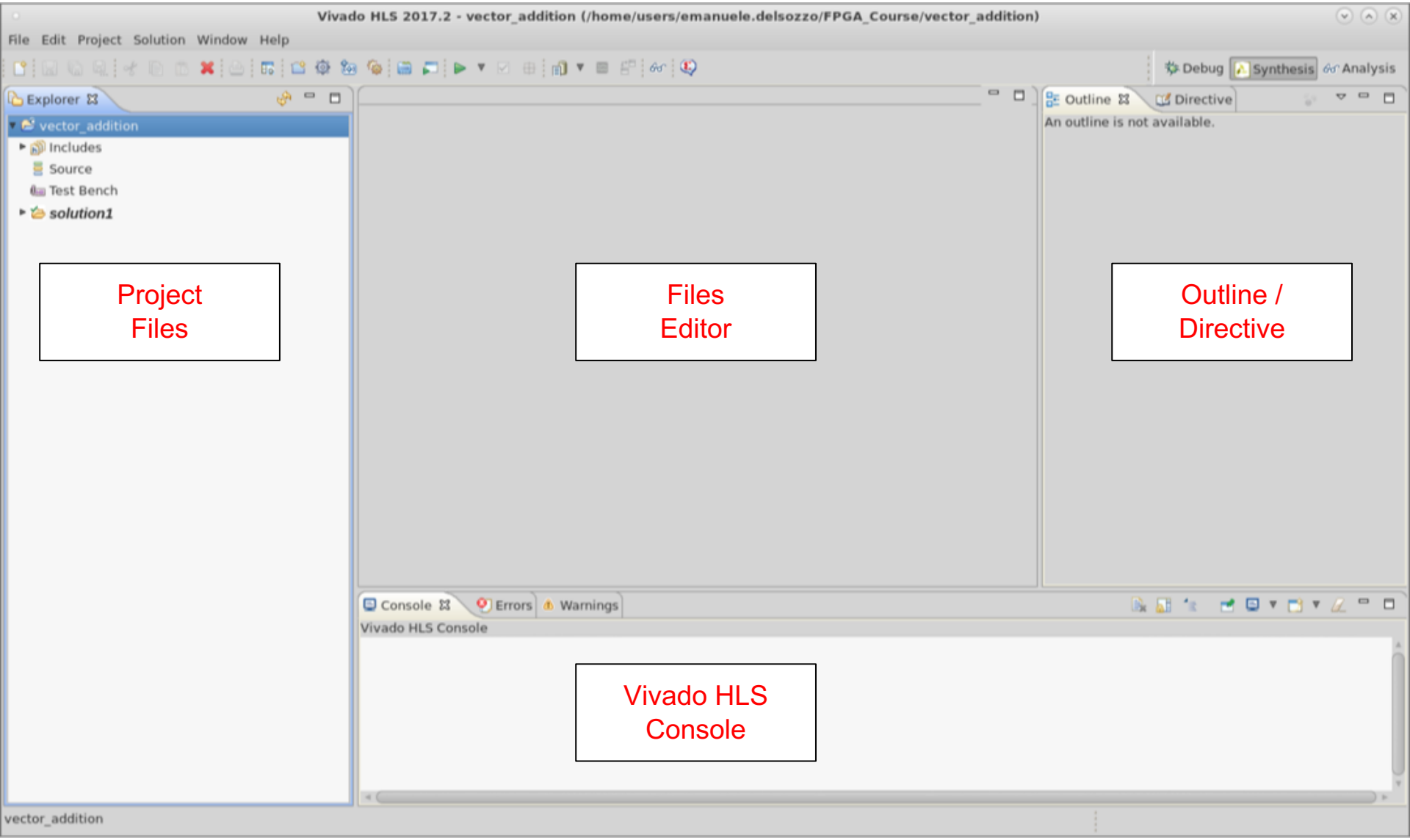
Device Selection



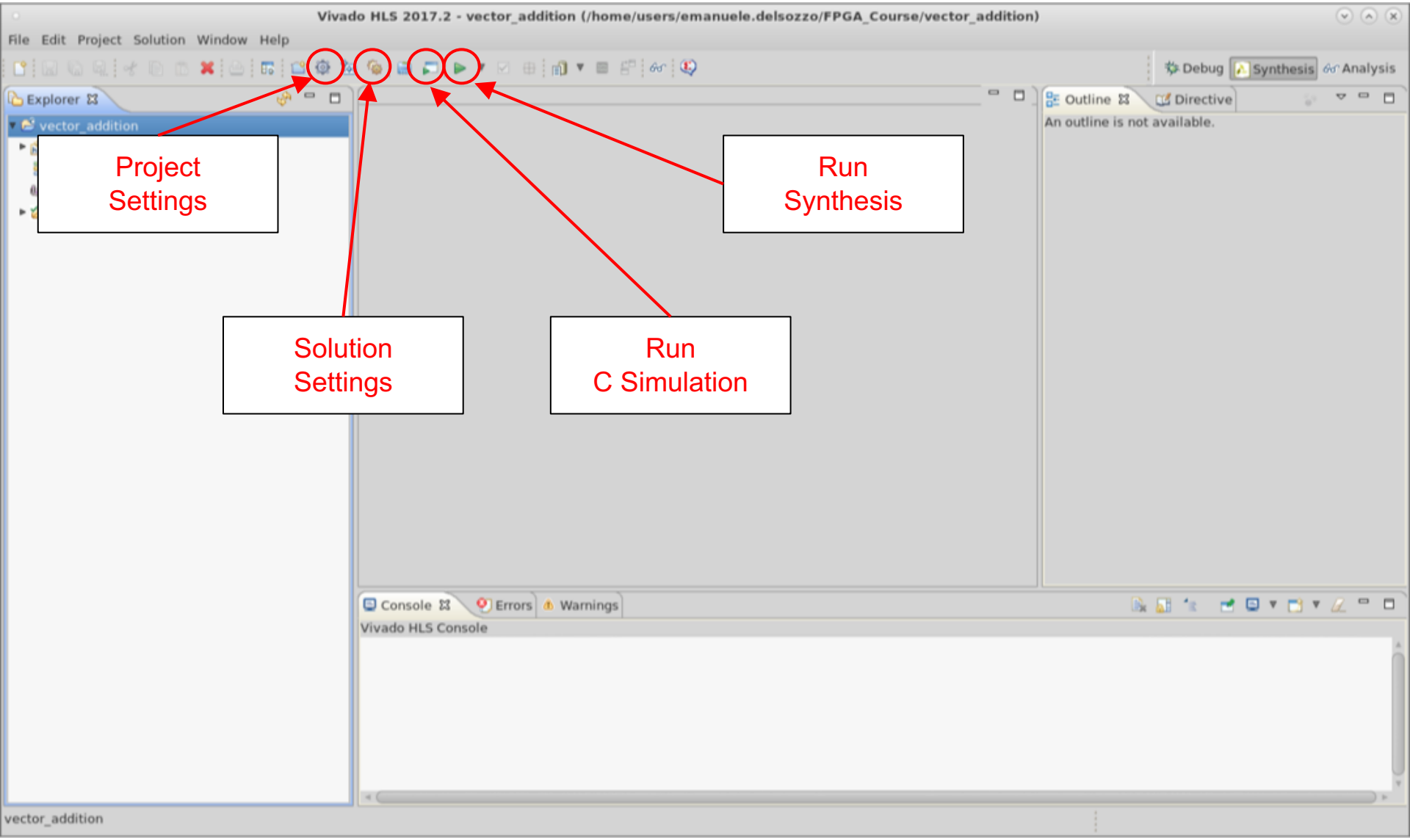
Solution Configuration



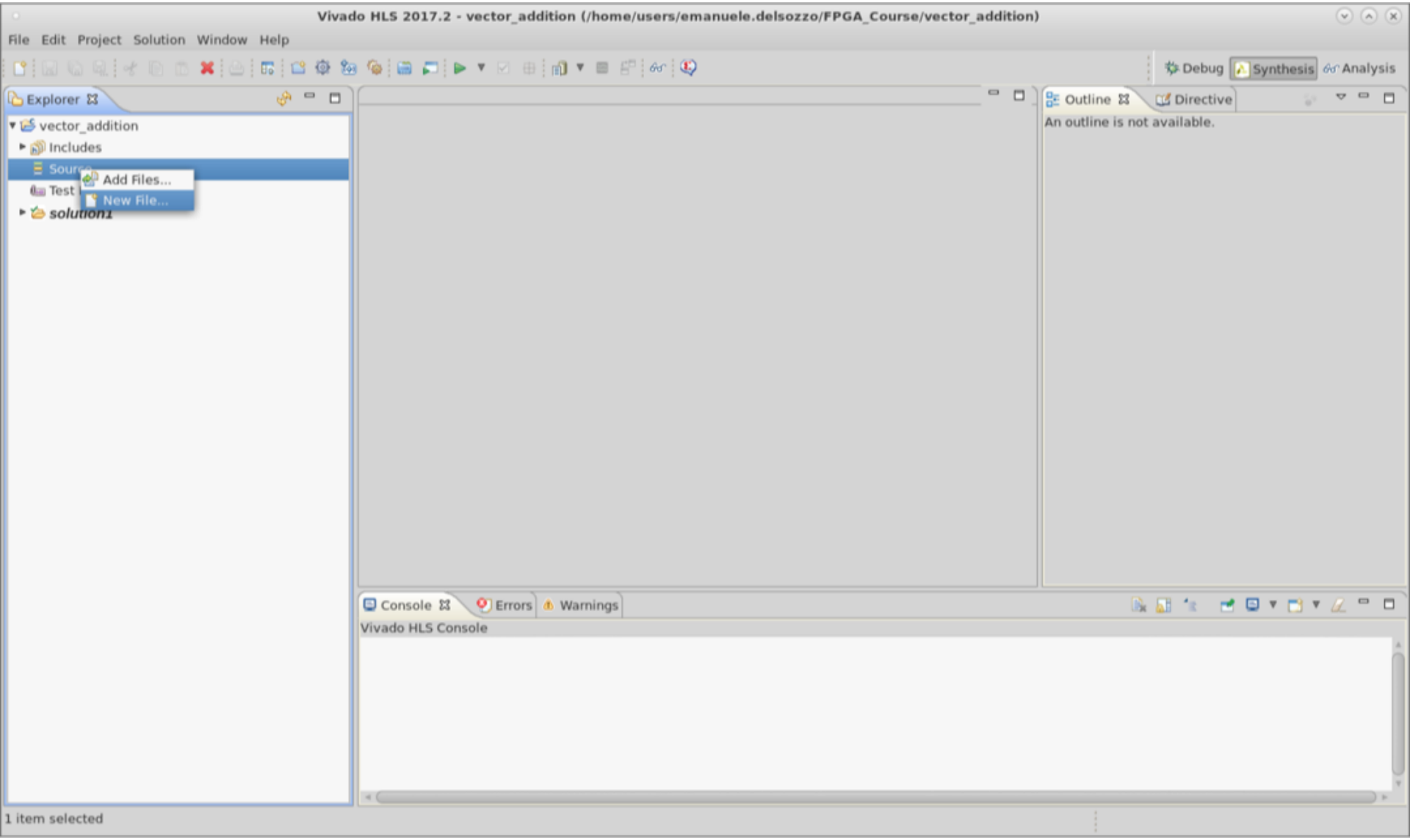
Editor Interface



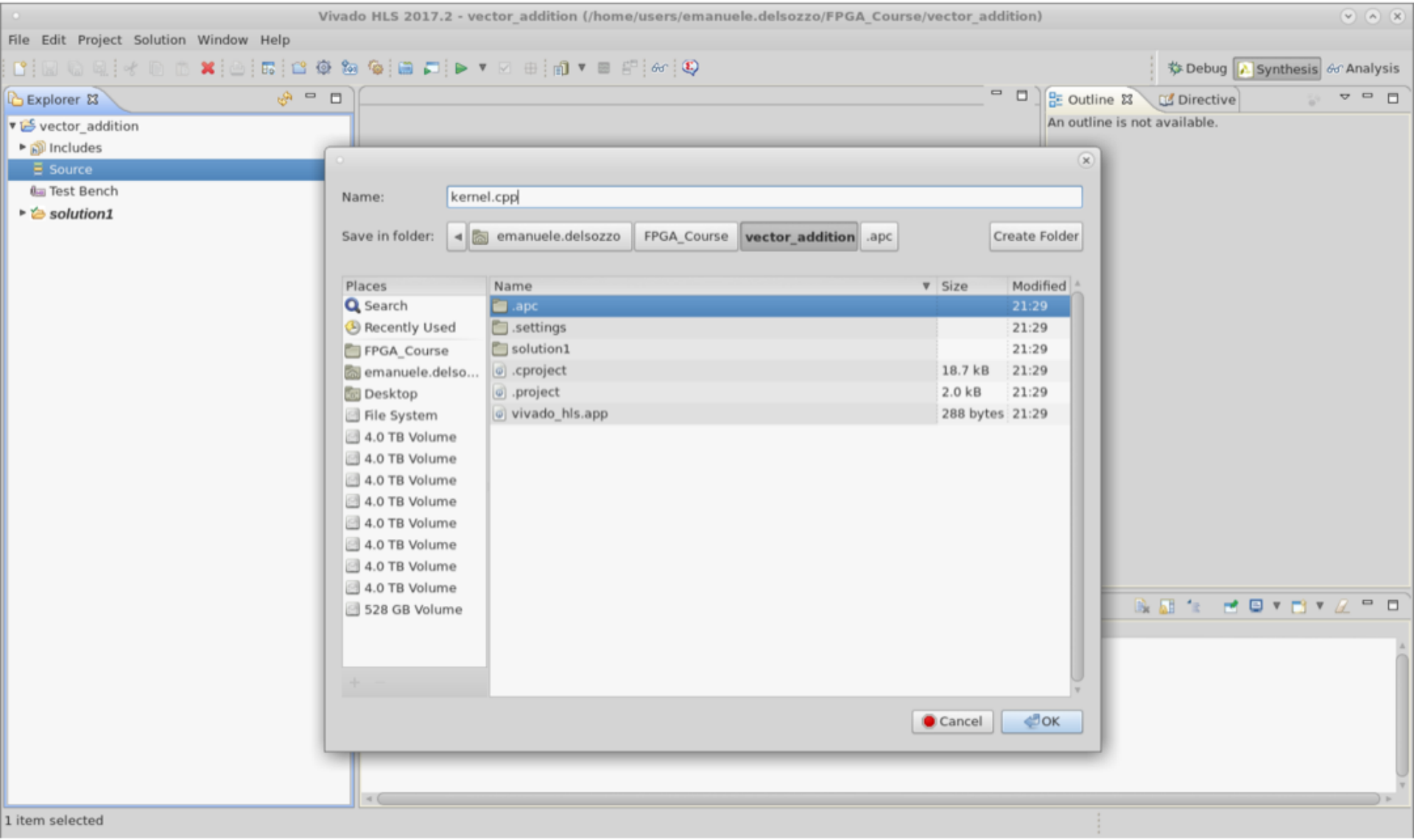
Editor Interface



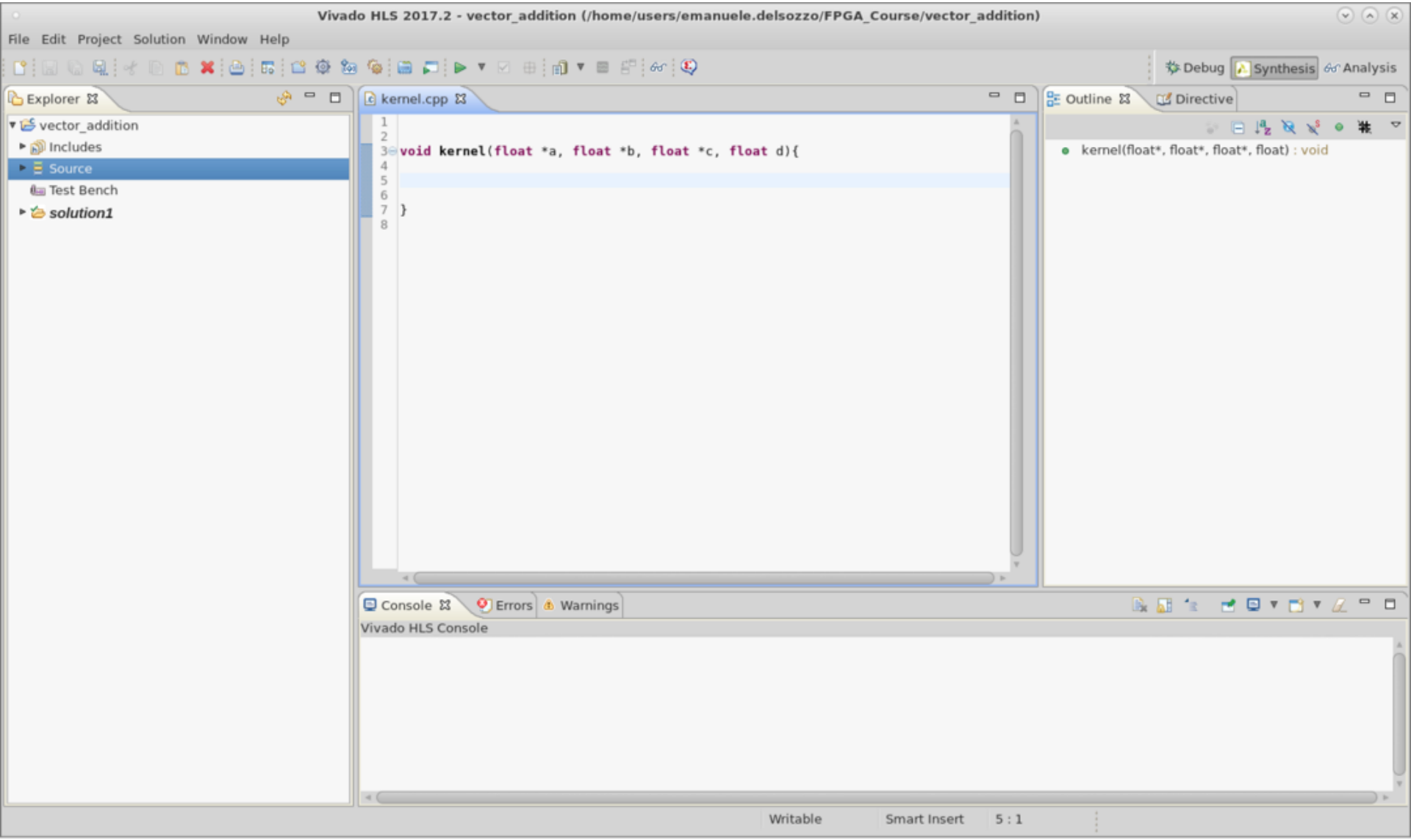
Kernel Creation



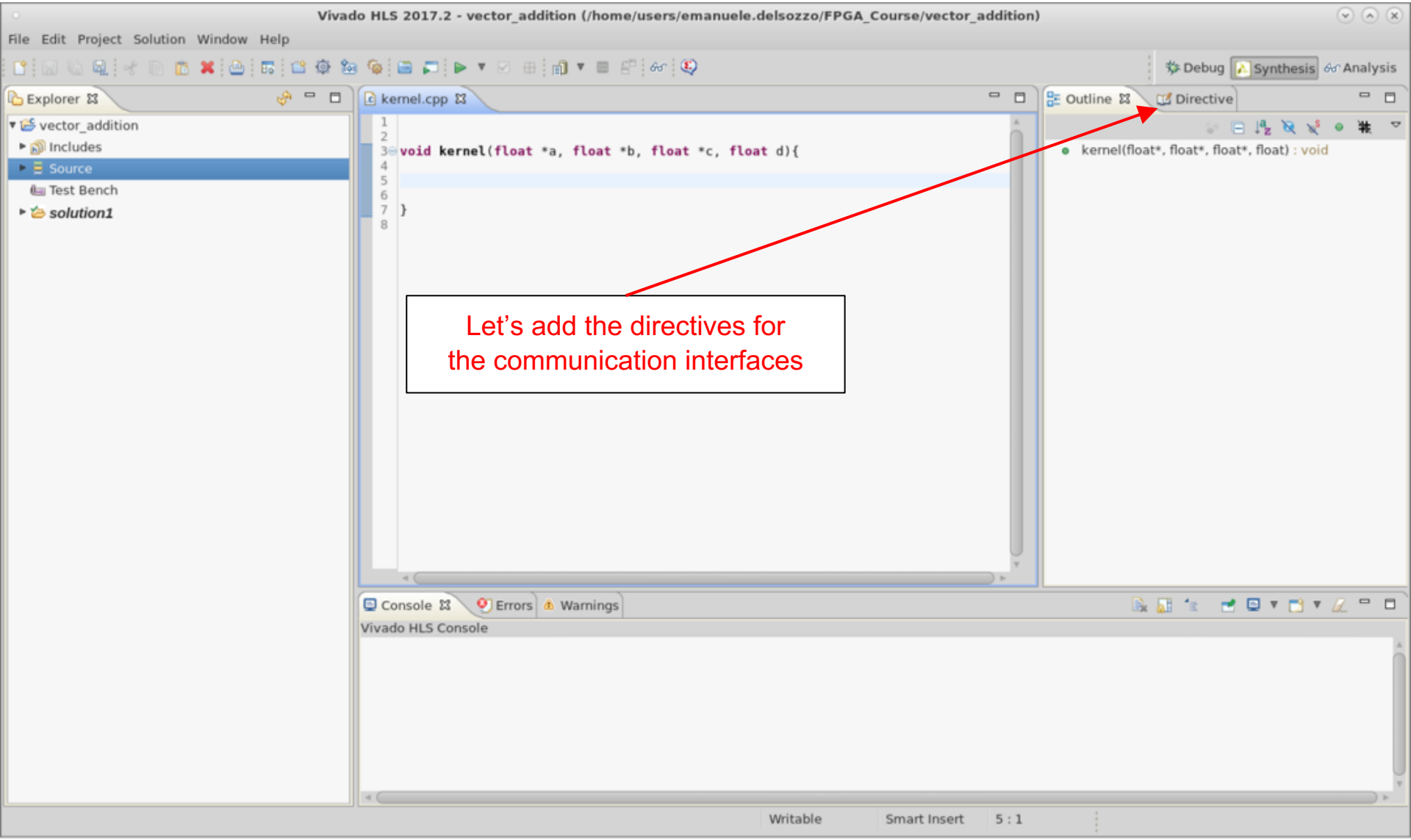
Kernel Creation



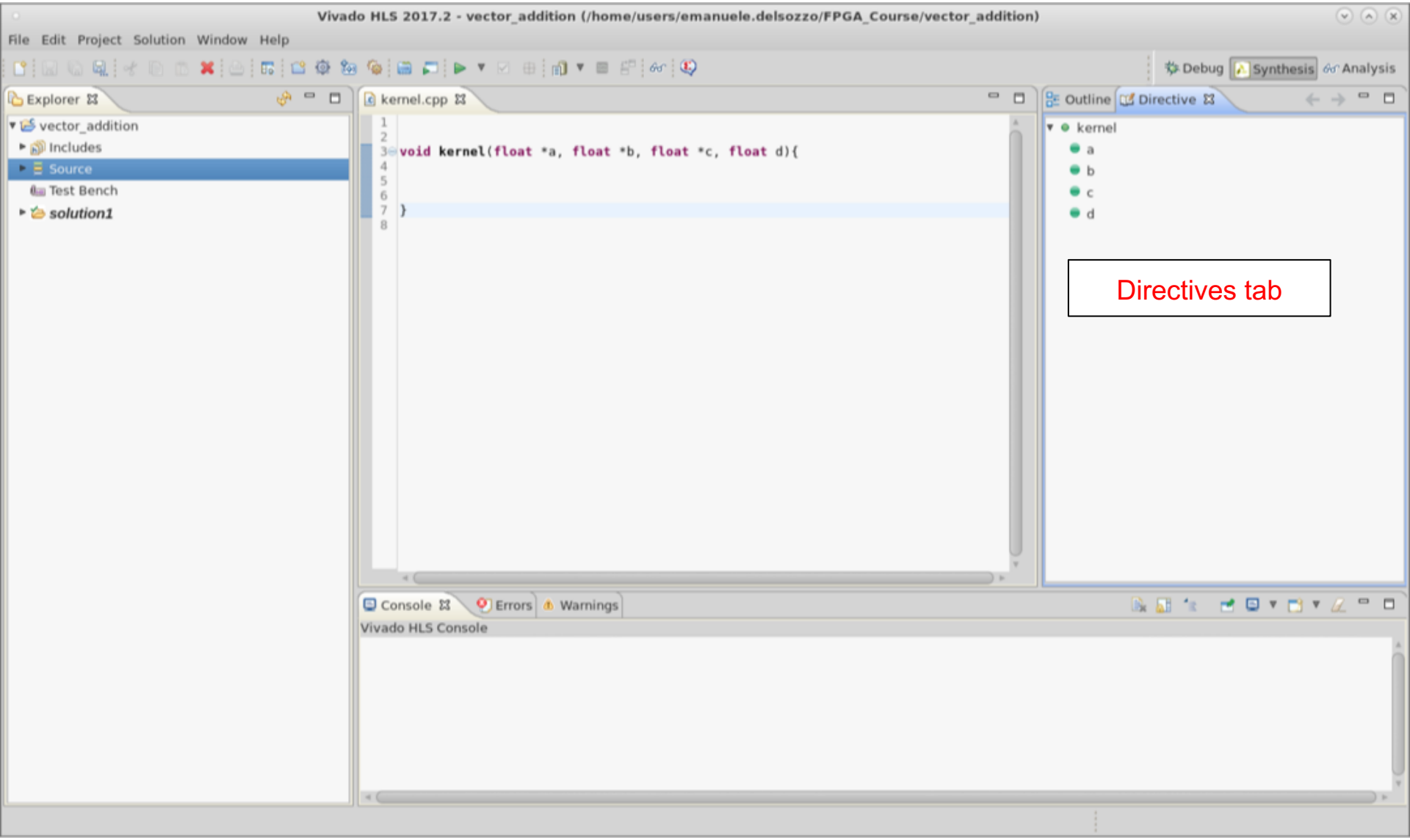
Kernel function



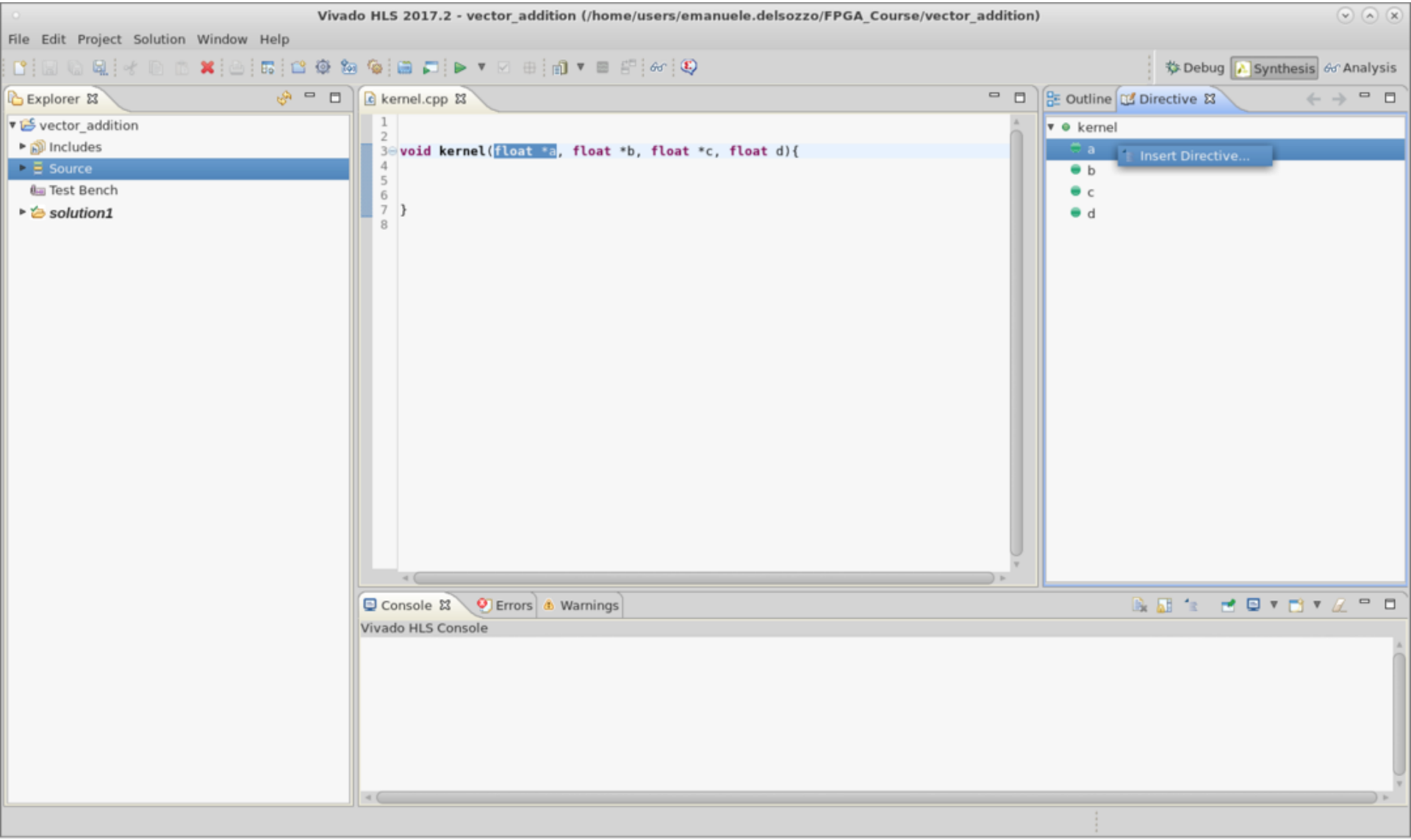
Kernel function



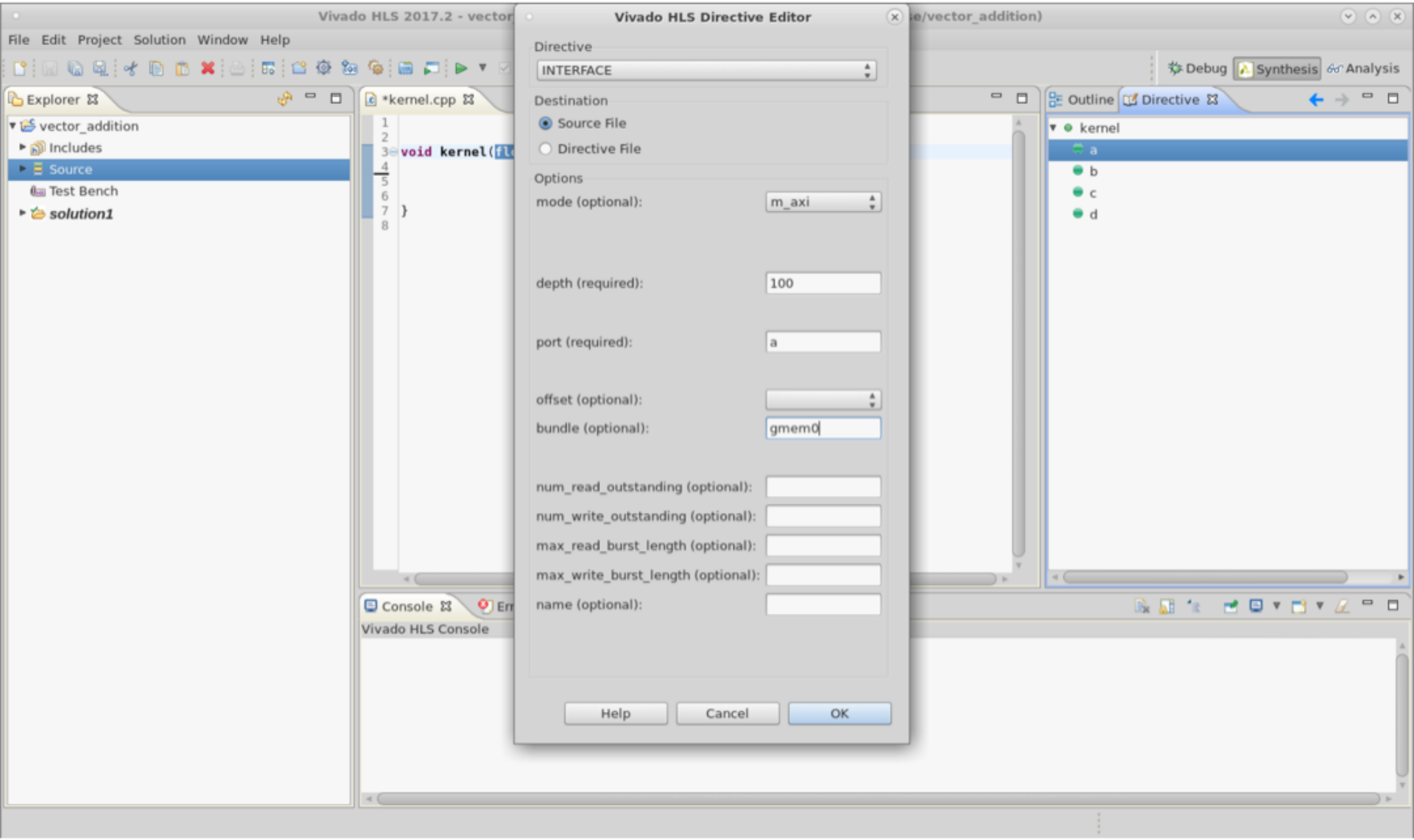
Directives


Vivado HLS 2017.2 - vector_addition (/home/users/emanuele.delsozzo/FPGA_Course/vector_addition)
File Edit Project Solution Window Help
Debug Synthesis Analysis
Explorer
vector_addition
Includes
Source
Test Bench
solution1
kernel.cpp
1
2
3 void kernel(float *a, float *b, float *c, float d){
4
5
6
7 }
8
Outline Directive
kernel
a
b
c
d
Directives tab
Console Errors Warnings
Vivado HLS Console

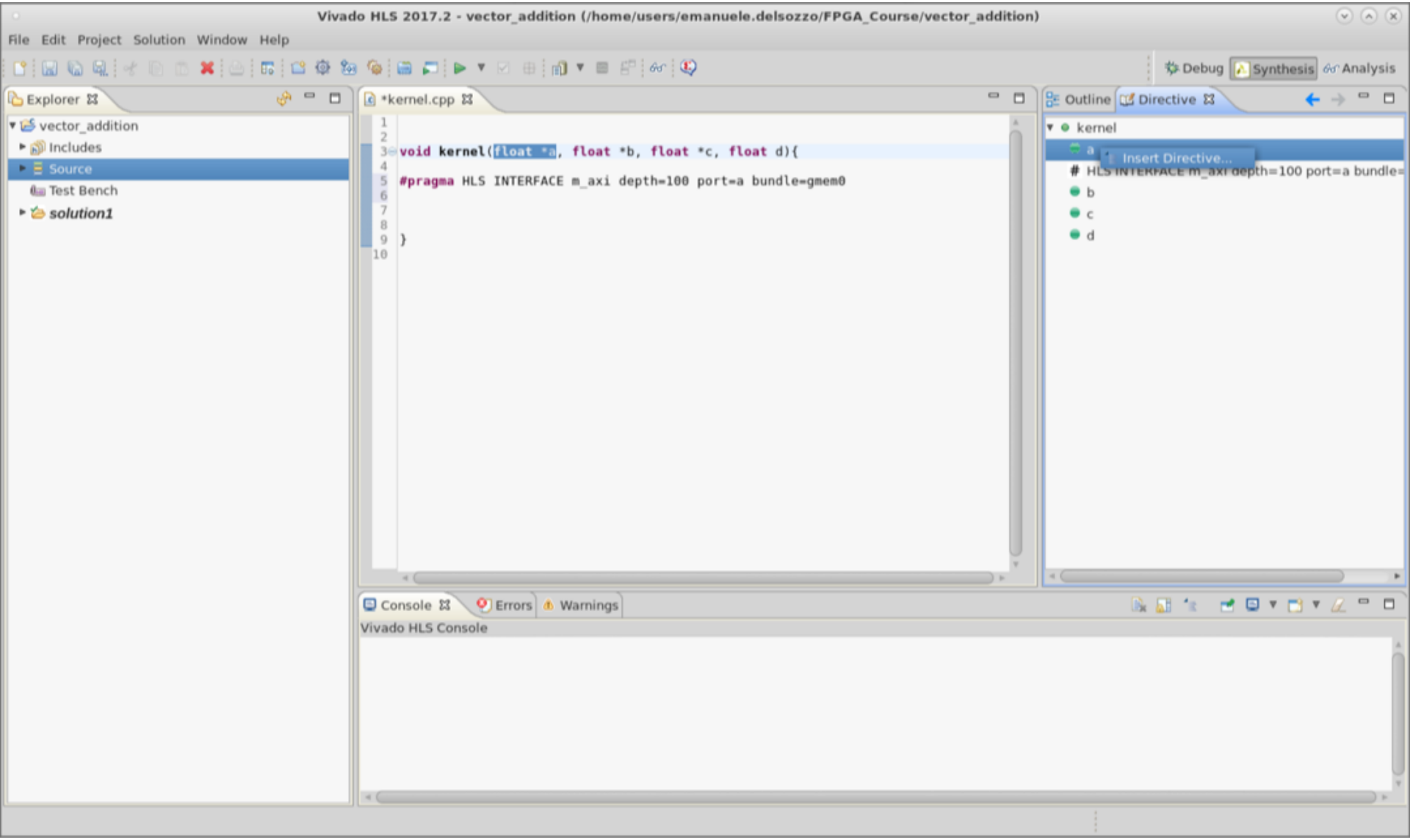
Insert Directive for port **a**



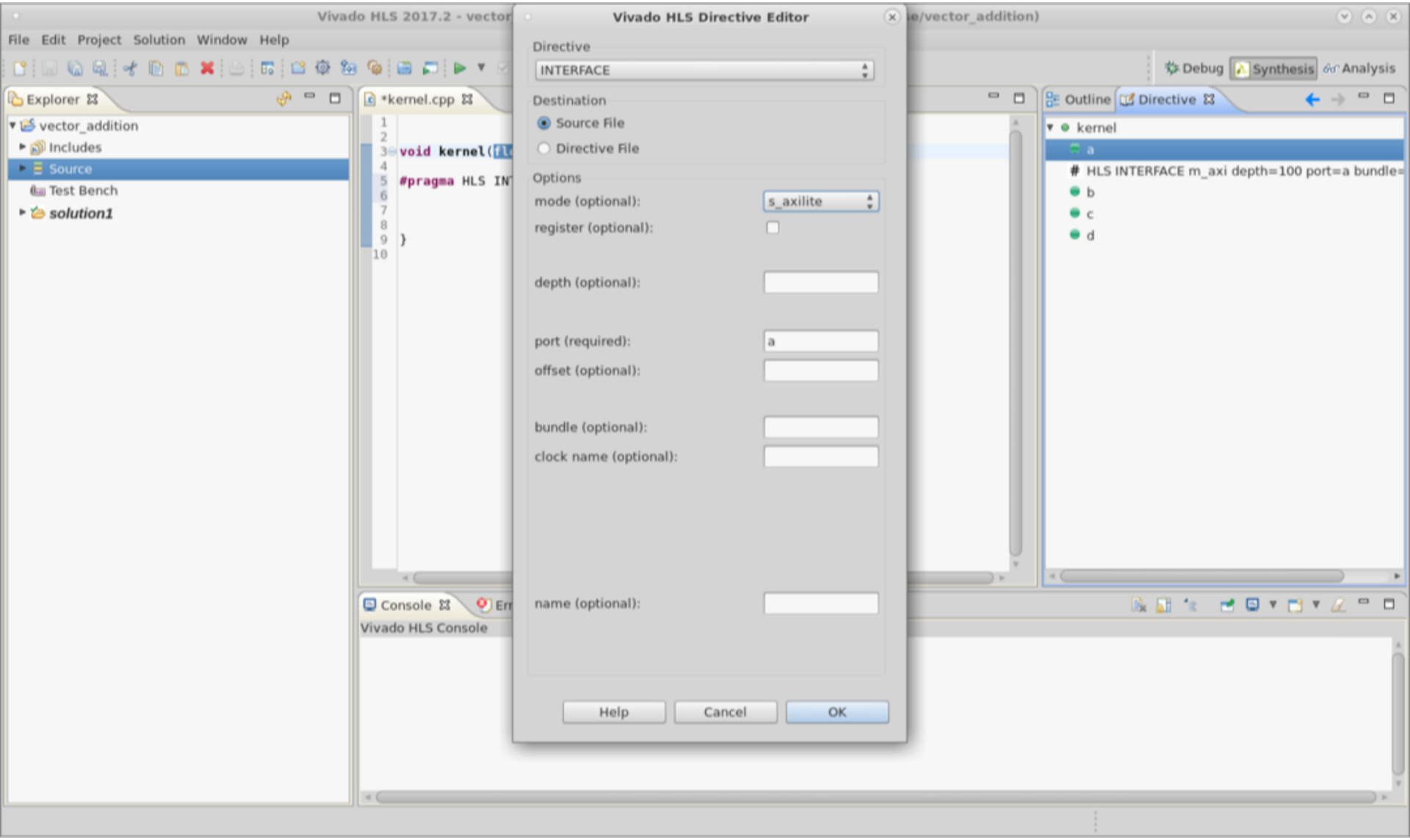
Master Axi Directive



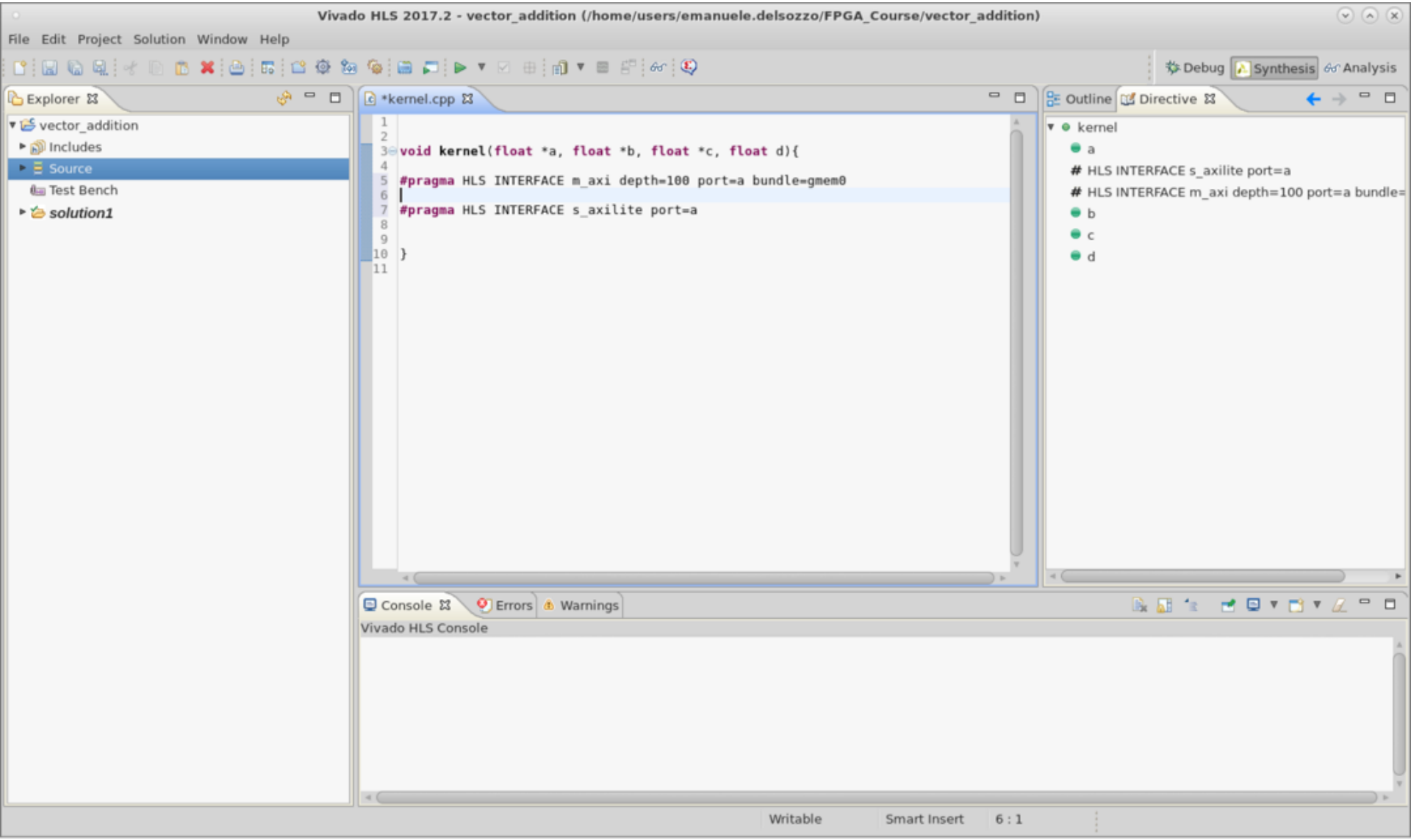
Insert Directive for port **a**



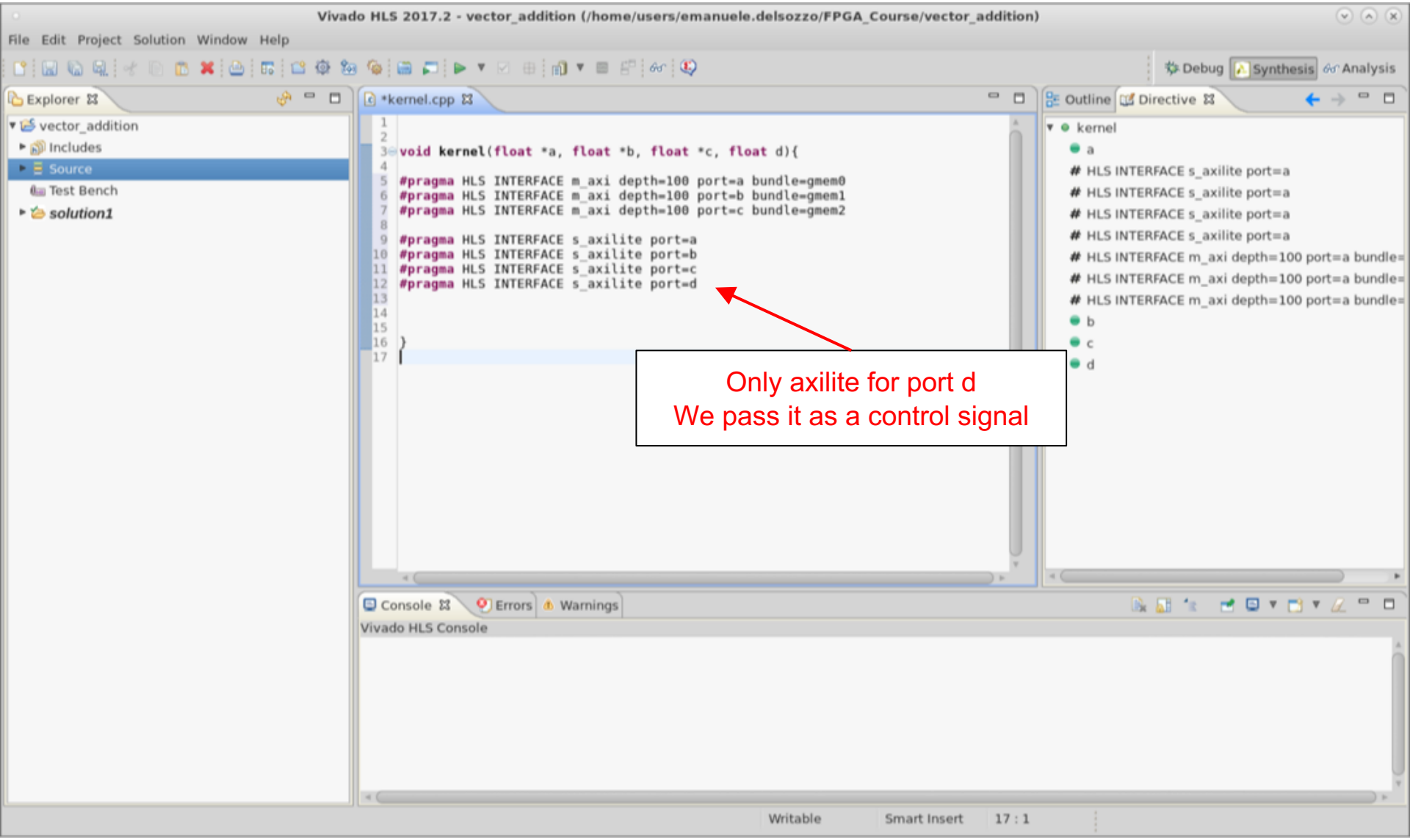
Axilite Directive



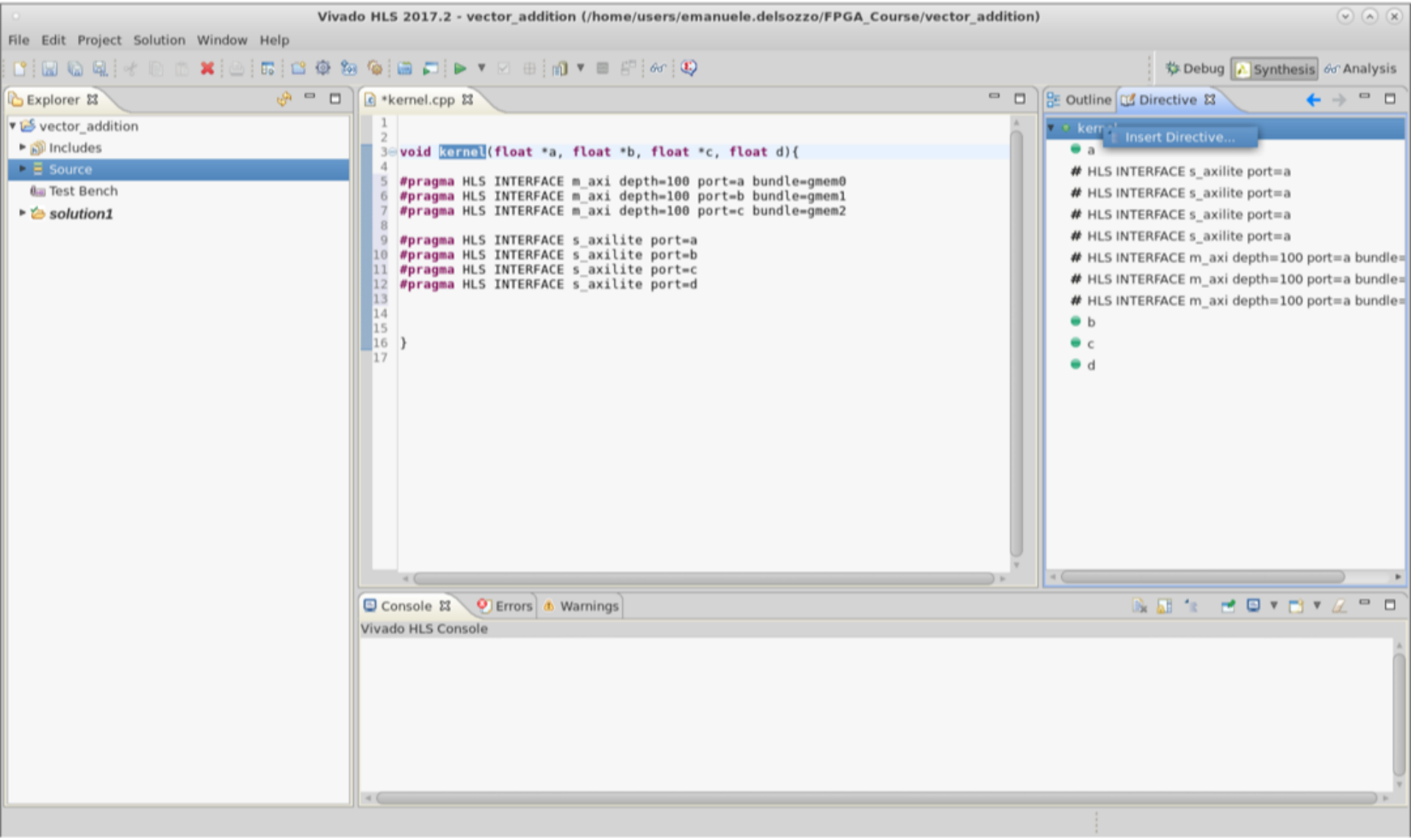
Port **a** directives



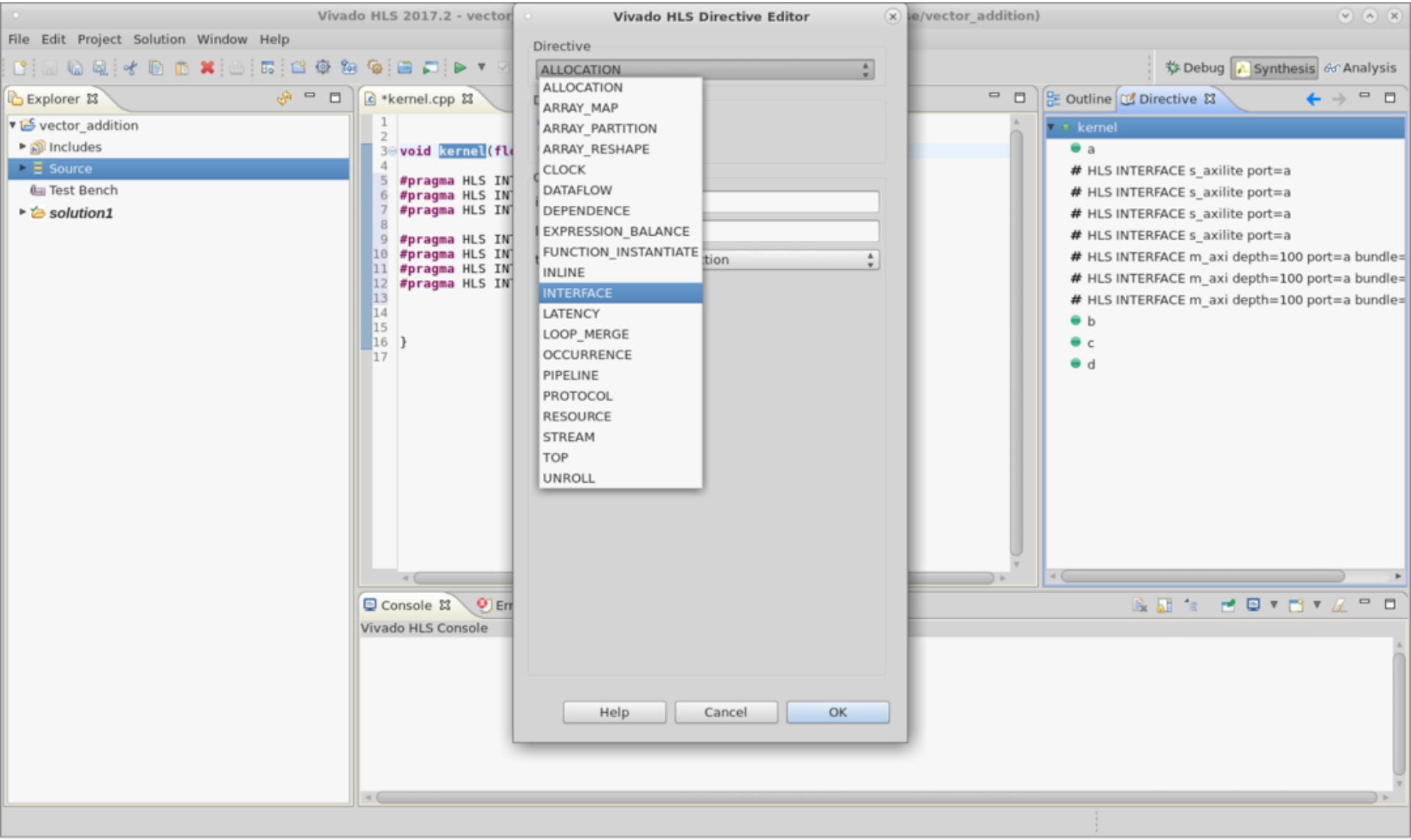
Ports directives



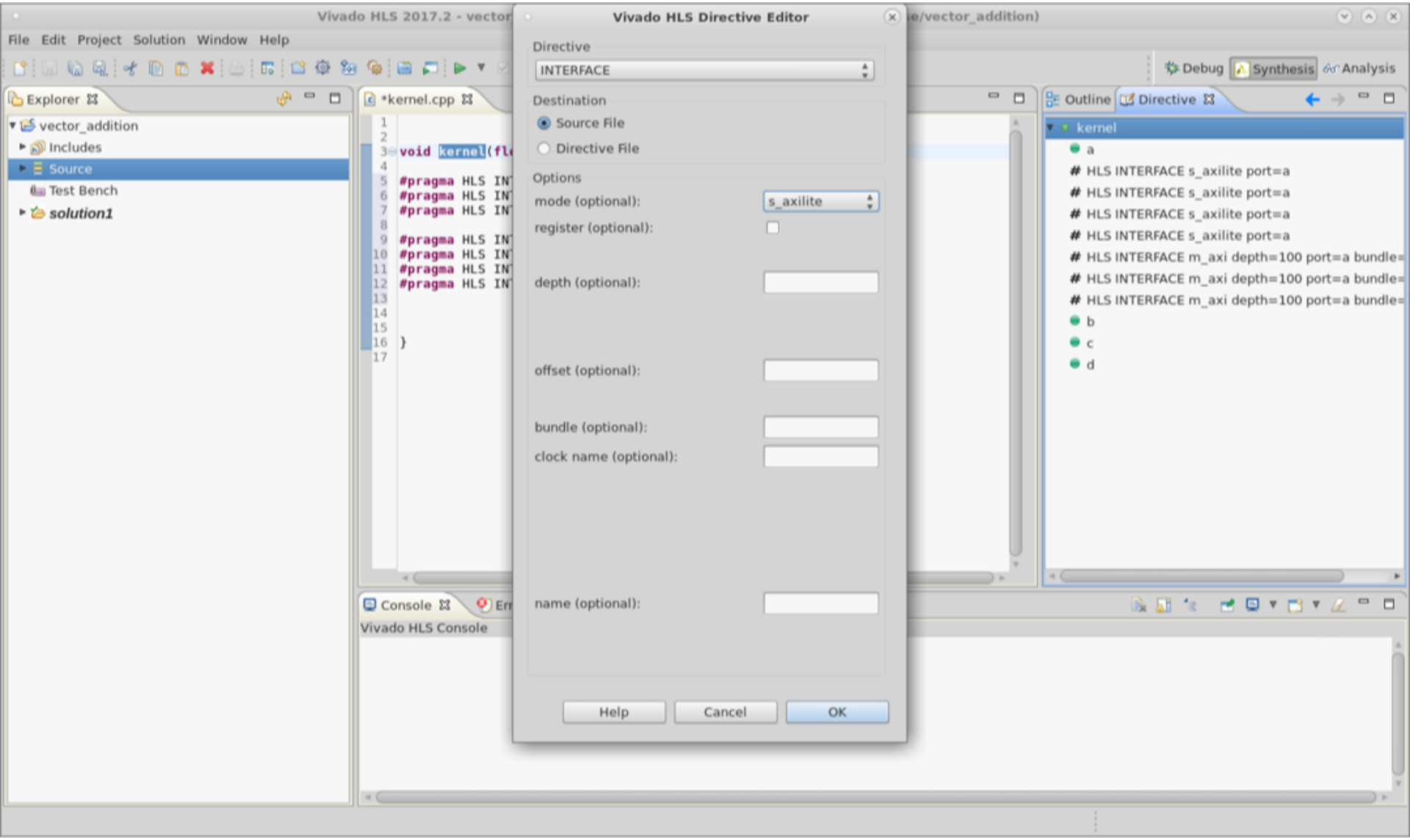
Insert Directive for kernel



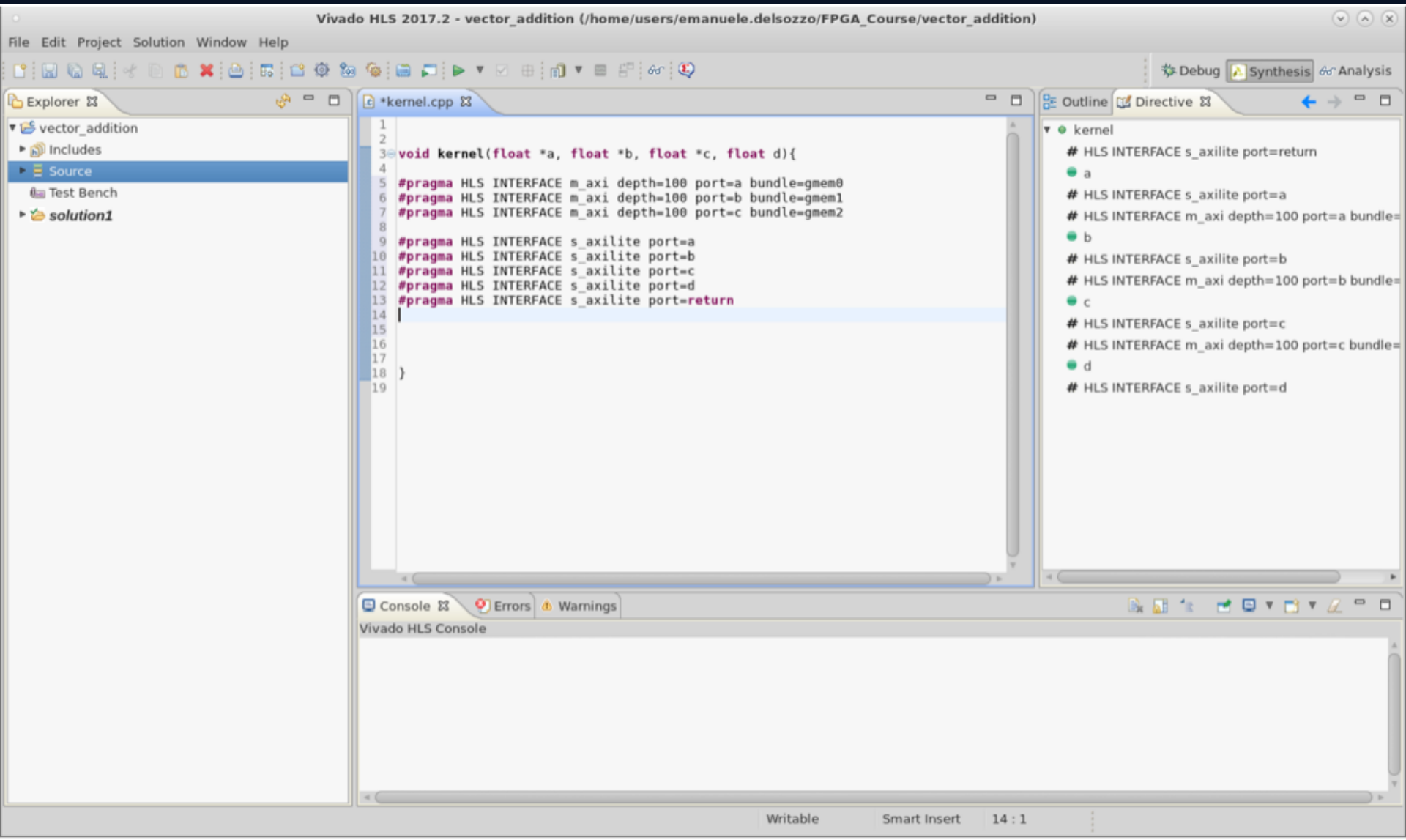
Interface Directive



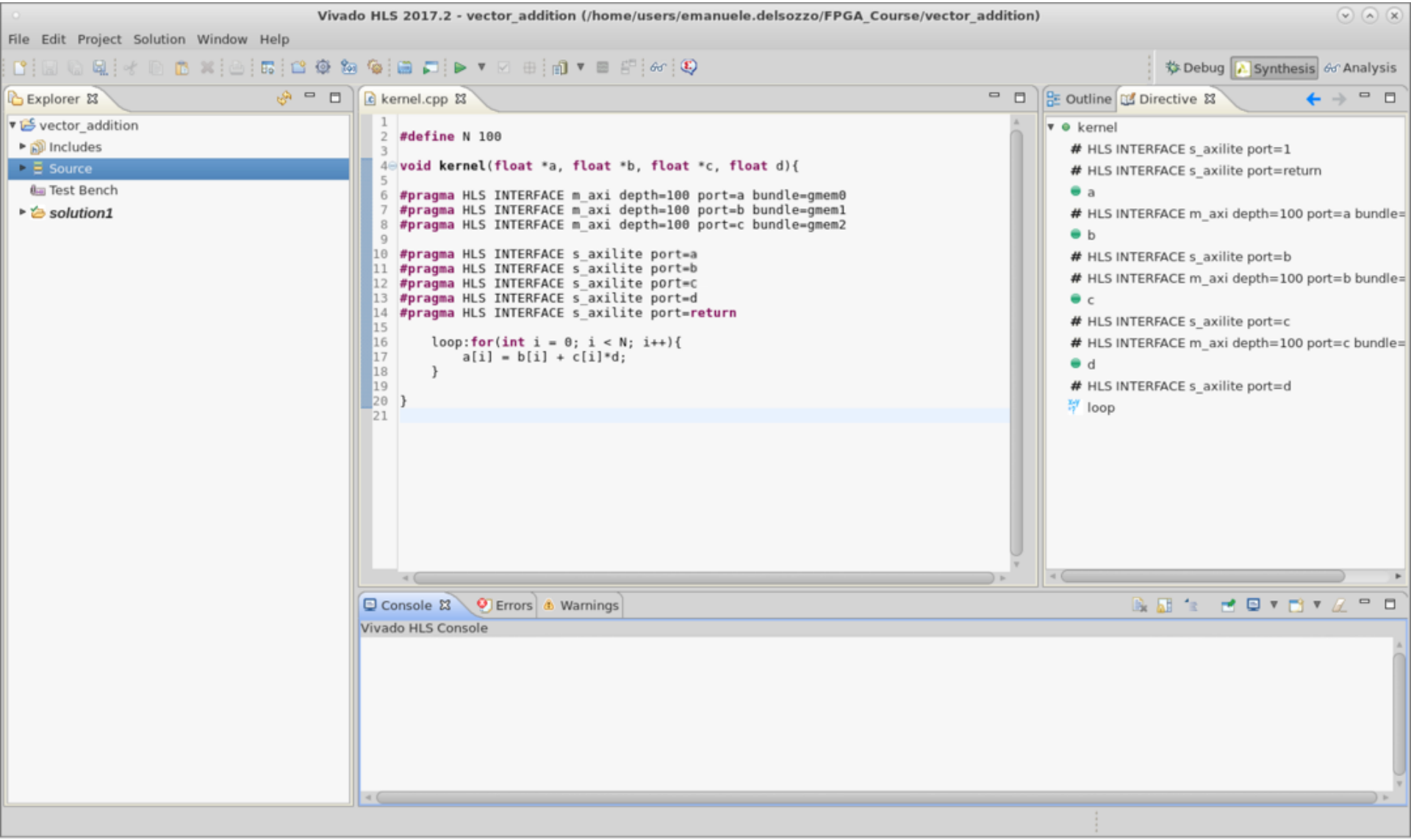
Axilite Directive



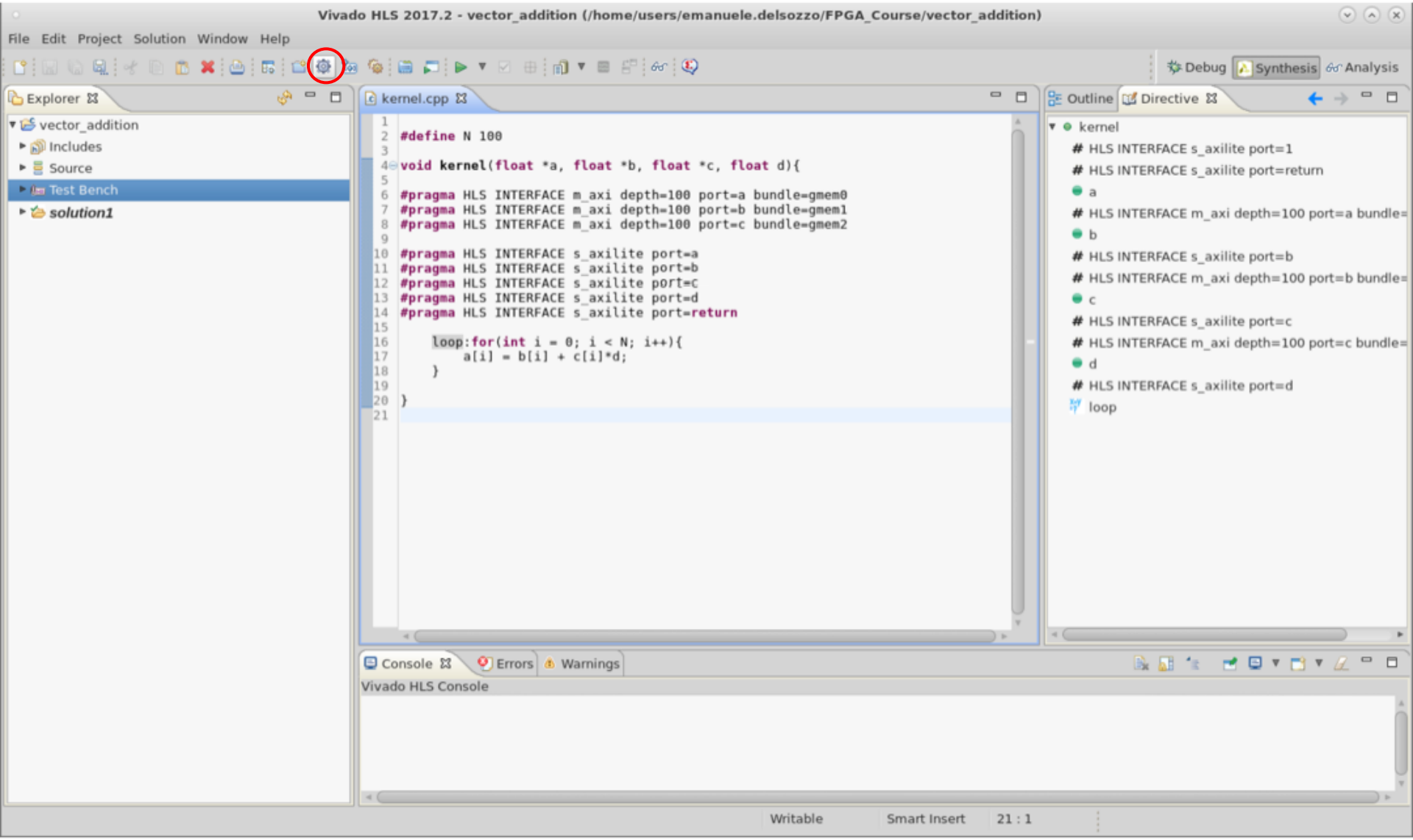
Kernel Directive



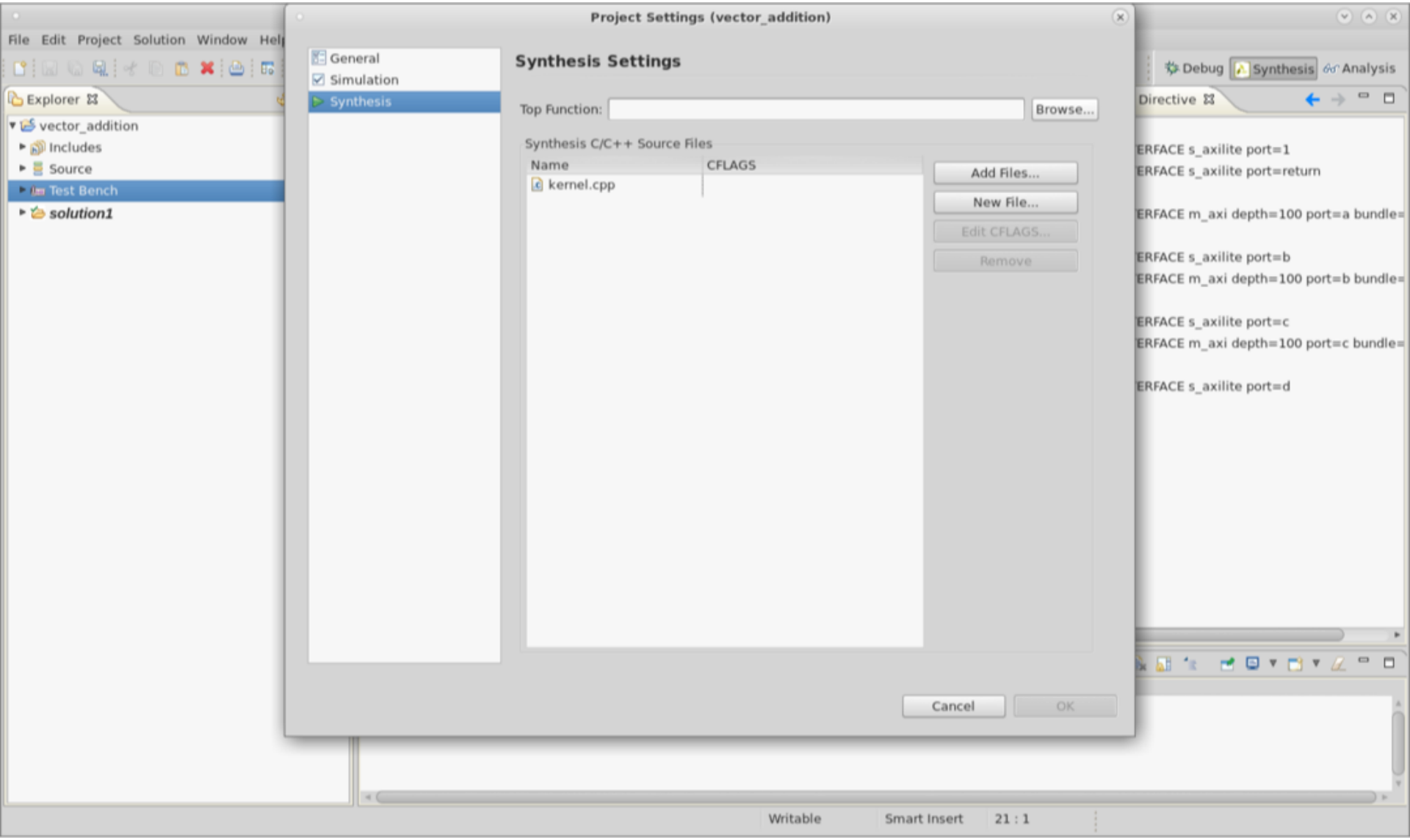
Naive Implementation



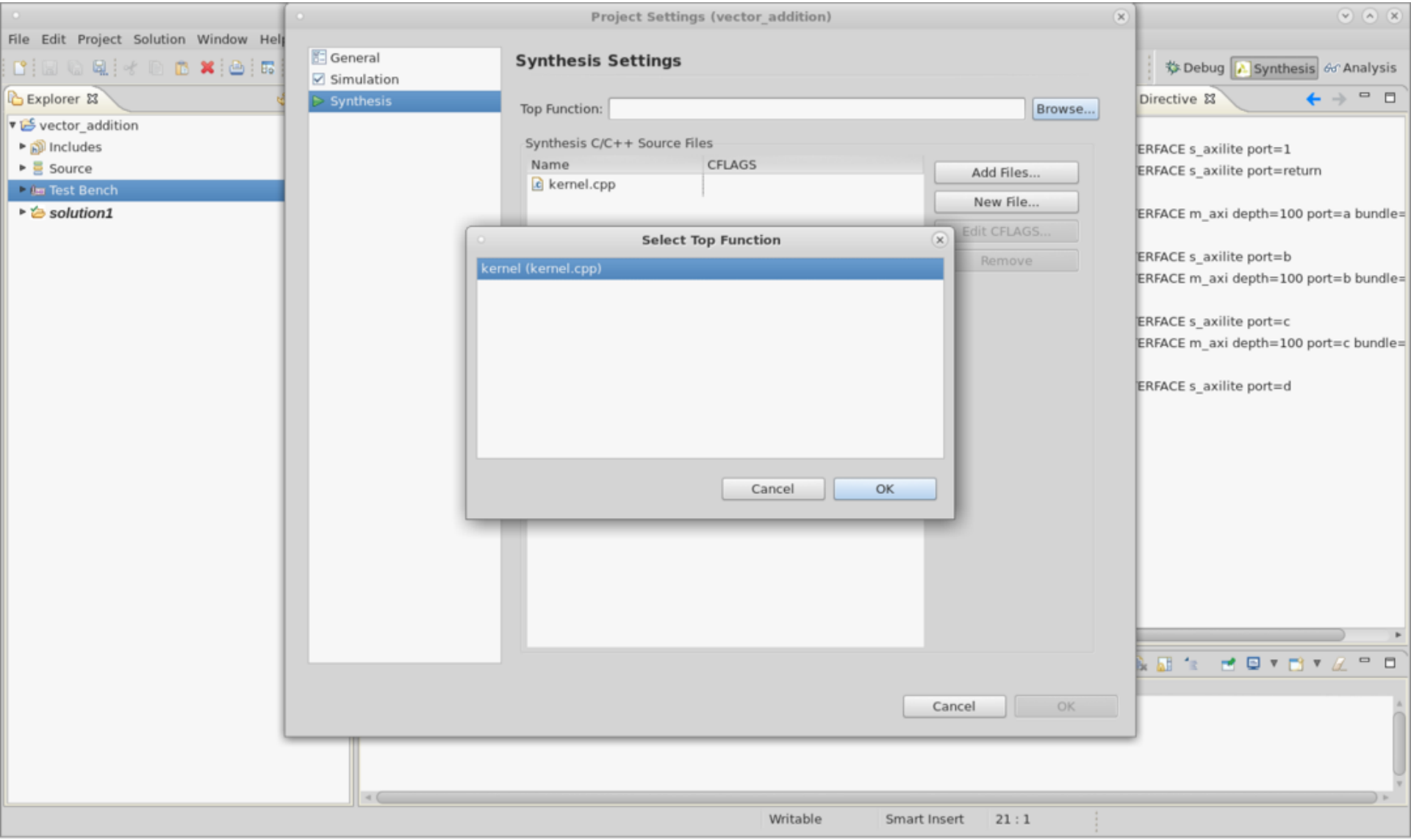
Project Settings



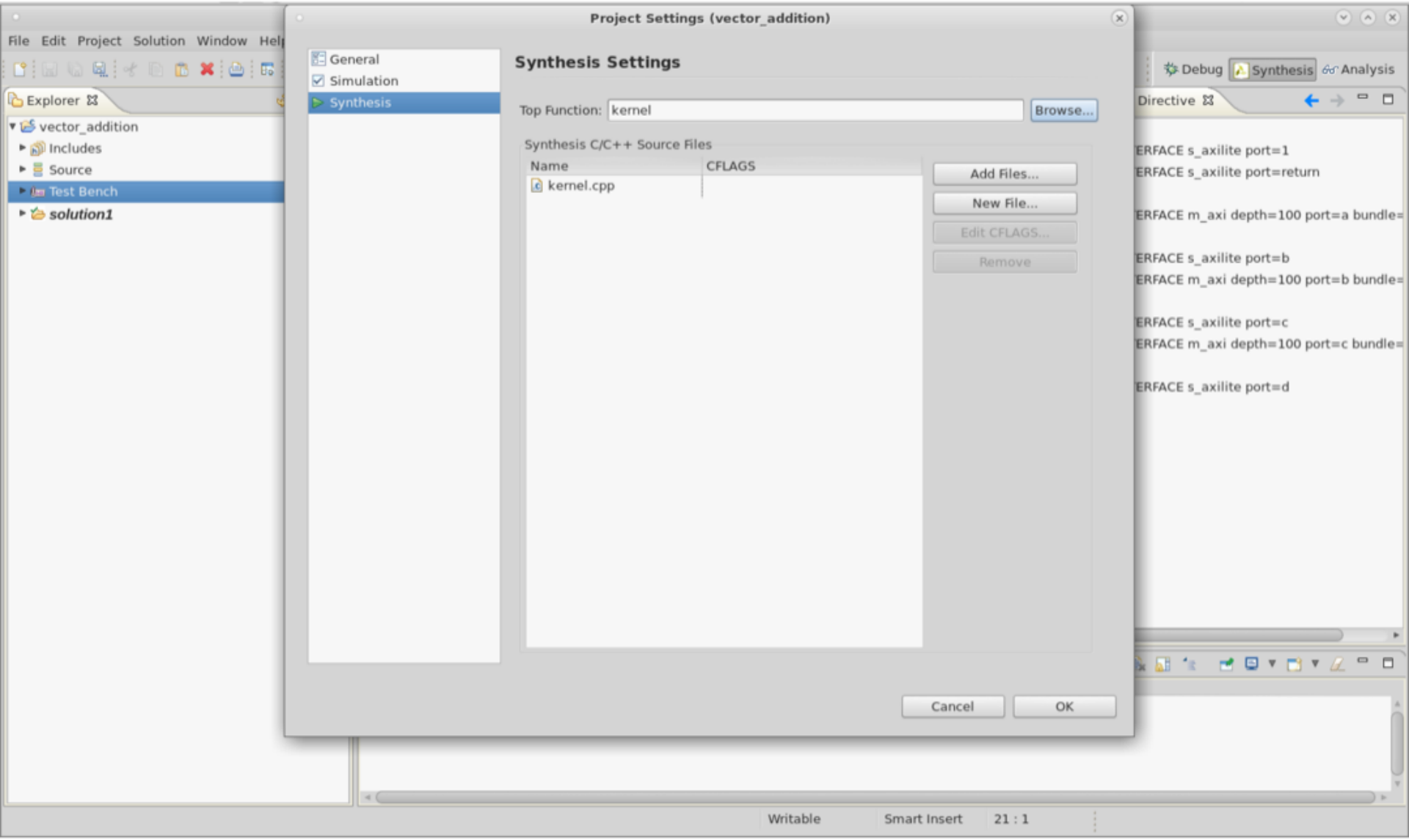
Top Function Selection



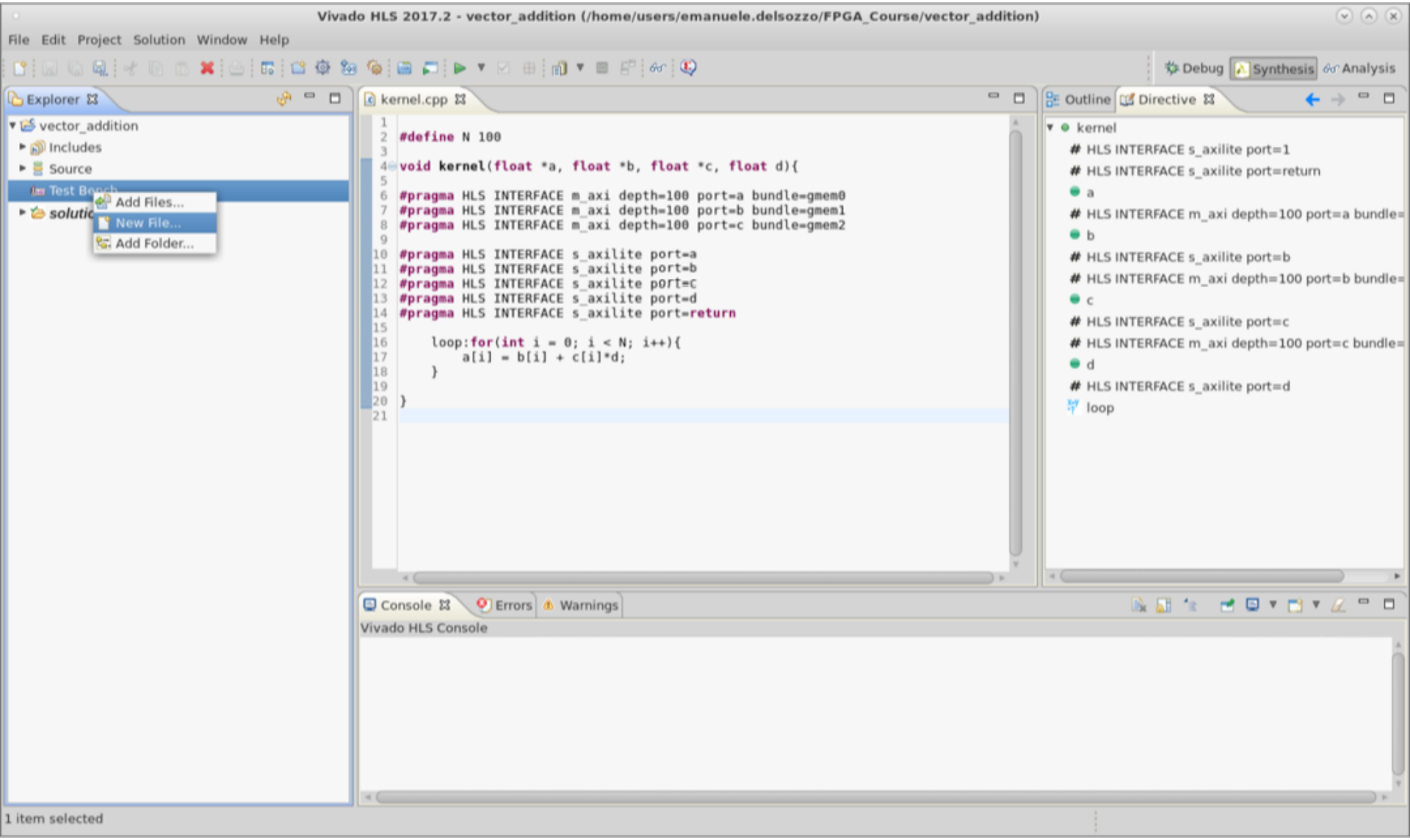
Top Function Selection



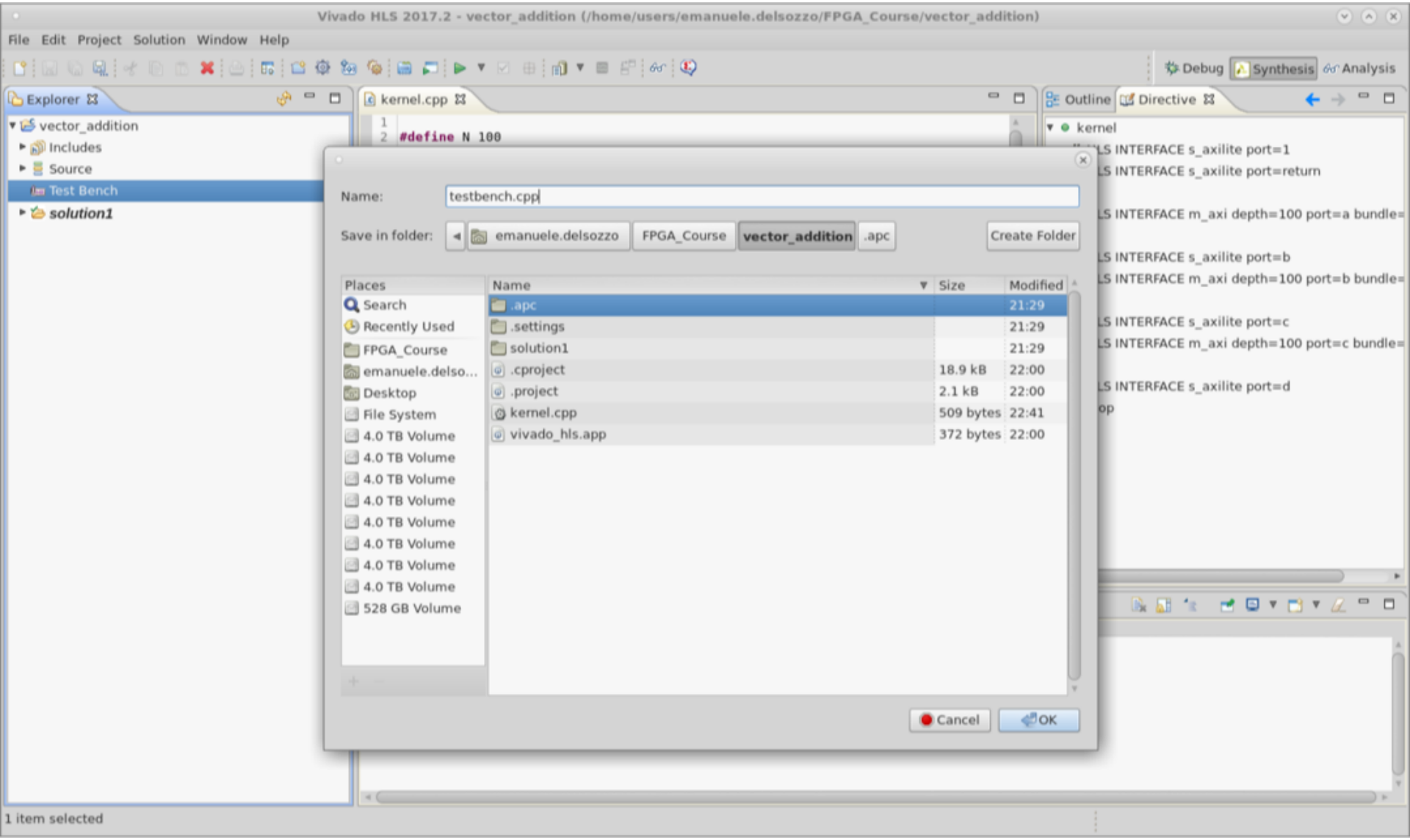
Top Function Selection



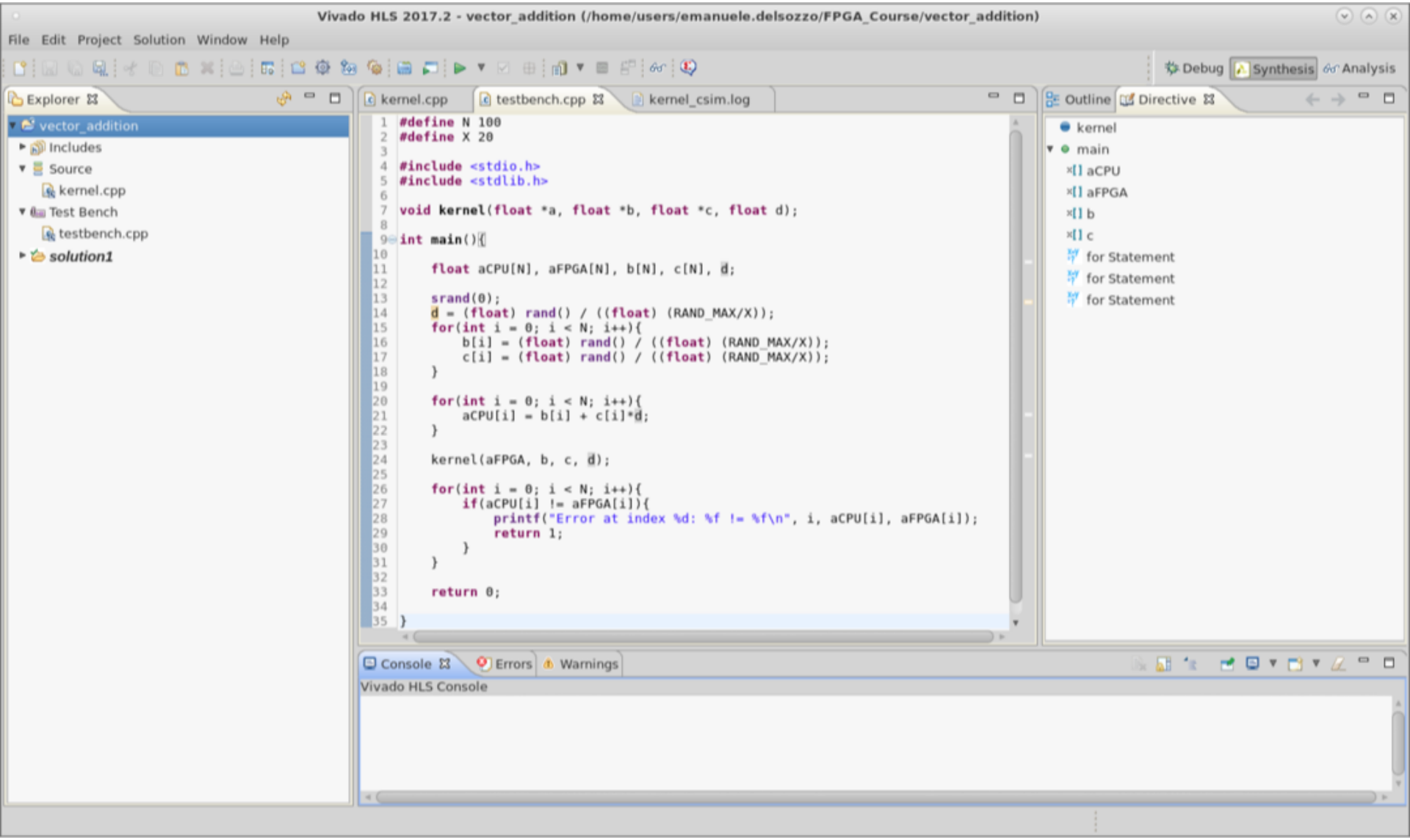
Testbench Creation



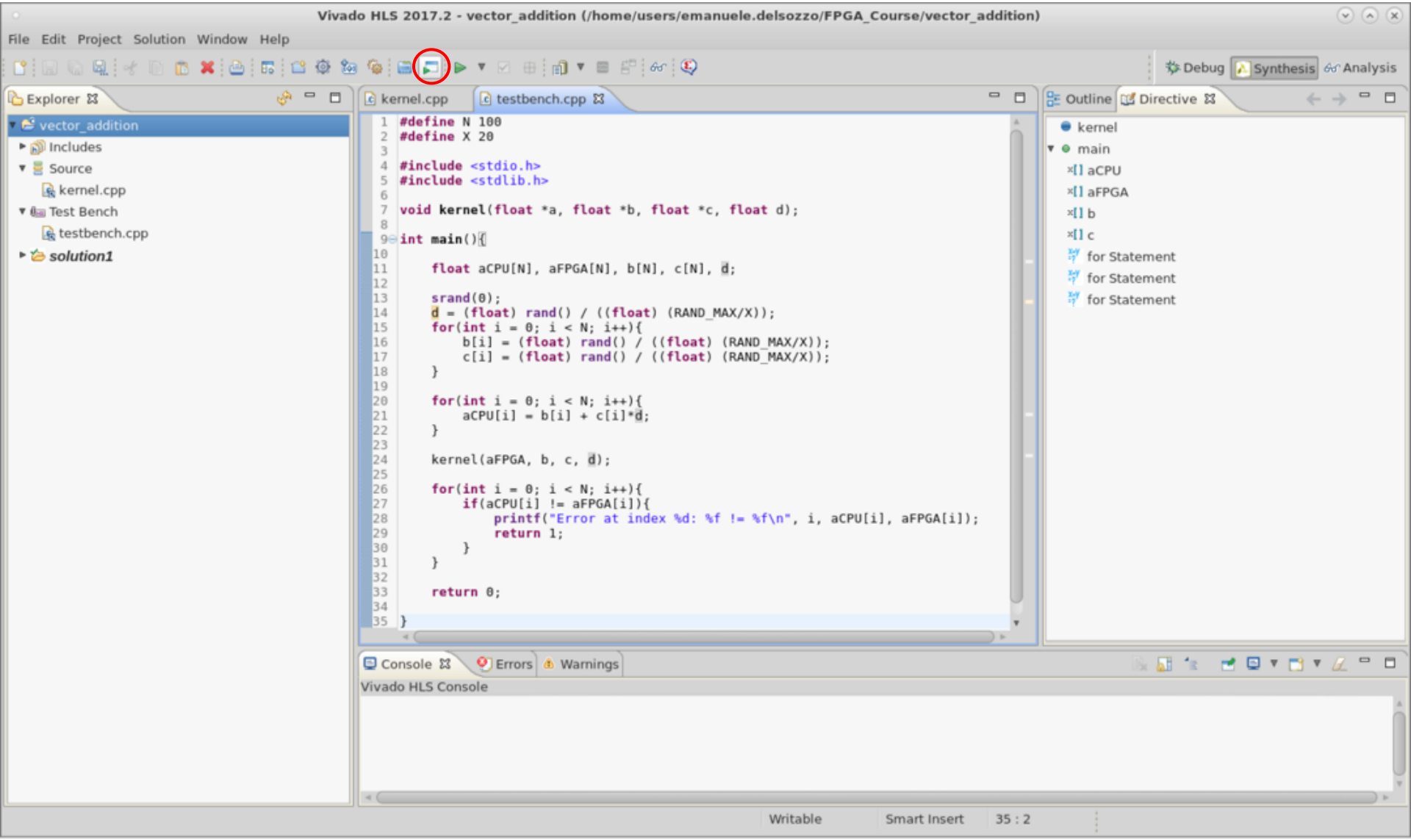
Testbench Creation



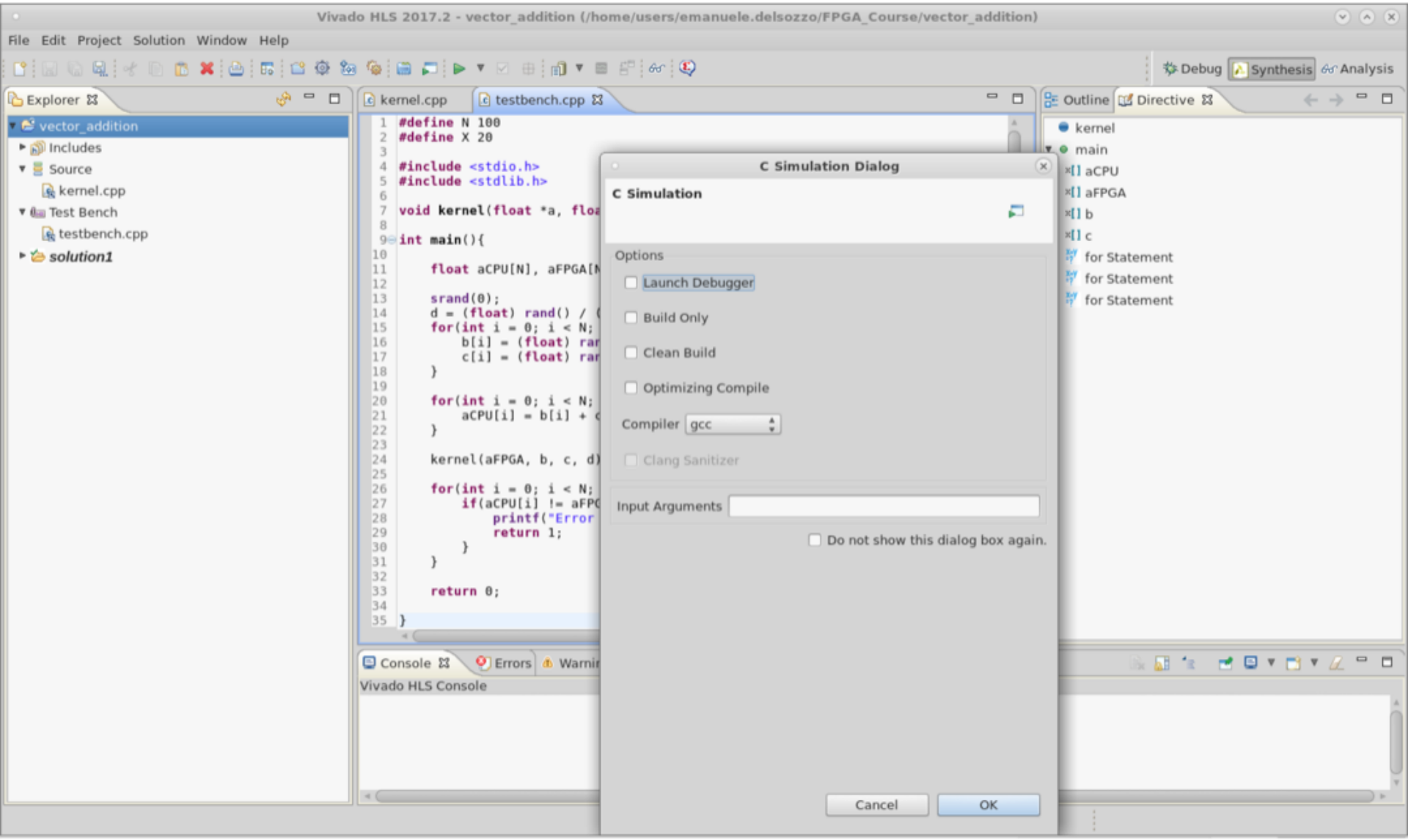
Testbench


Vivado HLS 2017.2 - vector_addition (/home/users/emanuele.delsozzo/FPGA_Course/vector_addition)
File Edit Project Solution Window Help
Debug Synthesis Analysis
Explorer
vector_addition
Includes
Source
kernel.cpp
Test Bench
testbench.cpp
solution1
kernel.cpp testbench.cpp kernel_csim.log
#define N 100
#define X 20
#include <stdio.h>
#include <stdlib.h>
void kernel(float *a, float *b, float *c, float d);
int main(){
float aCPU[N], aFPGA[N], b[N], c[N], d;
srand(0);
d = (float) rand() / ((float) (RAND_MAX/X));
for(int i = 0; i < N; i++){
b[i] = (float) rand() / ((float) (RAND_MAX/X));
c[i] = (float) rand() / ((float) (RAND_MAX/X));
}
for(int i = 0; i < N; i++){
aCPU[i] = b[i] + c[i]*d;
}
kernel(aFPGA, b, c, d);
for(int i = 0; i < N; i++){
if(aCPU[i] != aFPGA[i]){
printf("Error at index %d: %f != %f\n", i, aCPU[i], aFPGA[i]);
return 1;
}
}
return 0;
}
Outline Directive
kernel
main
aCPU
aFPGA
b
c
for Statement
for Statement
for Statement
Console Errors Warnings
Vivado HLS Console

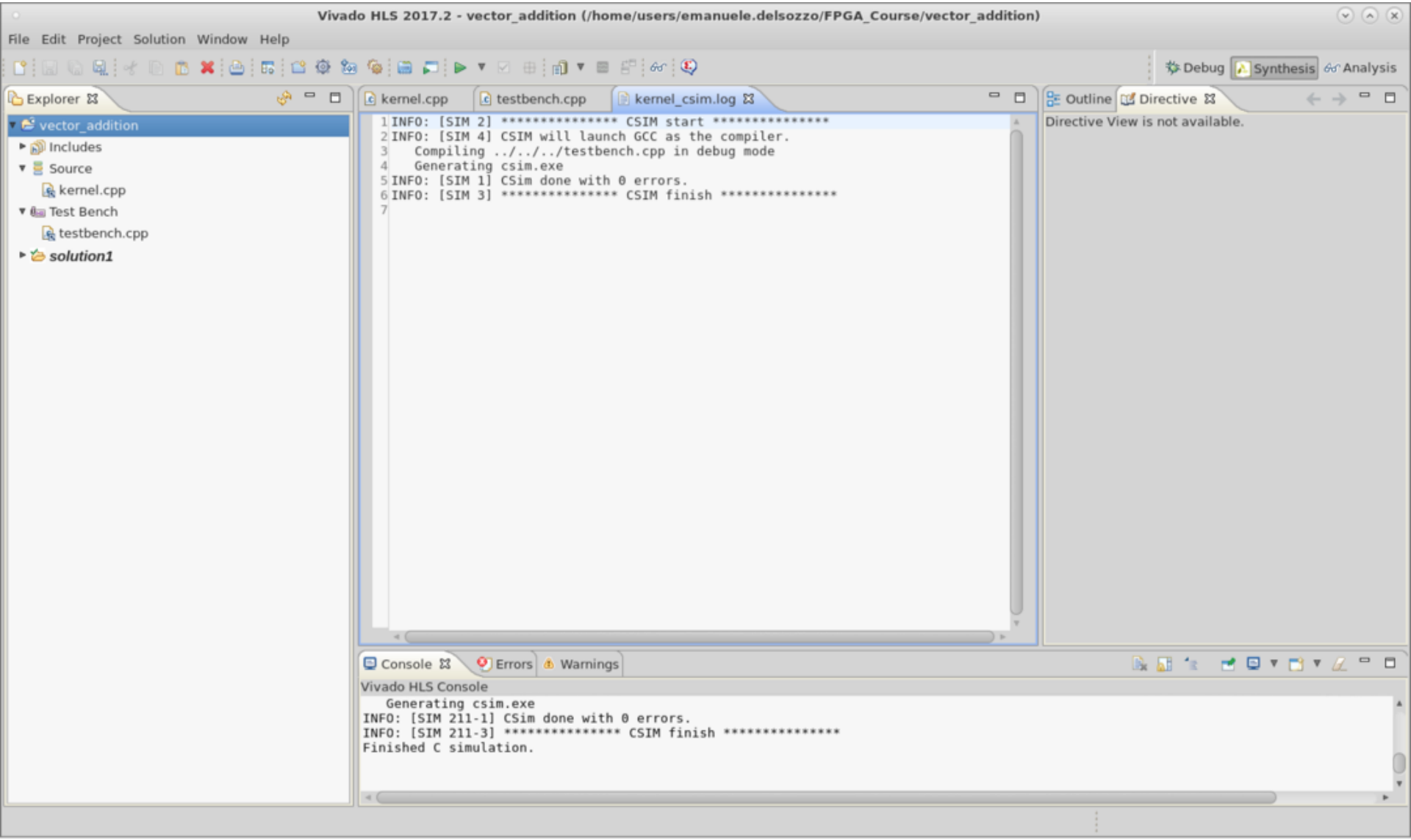
Run C Simulation



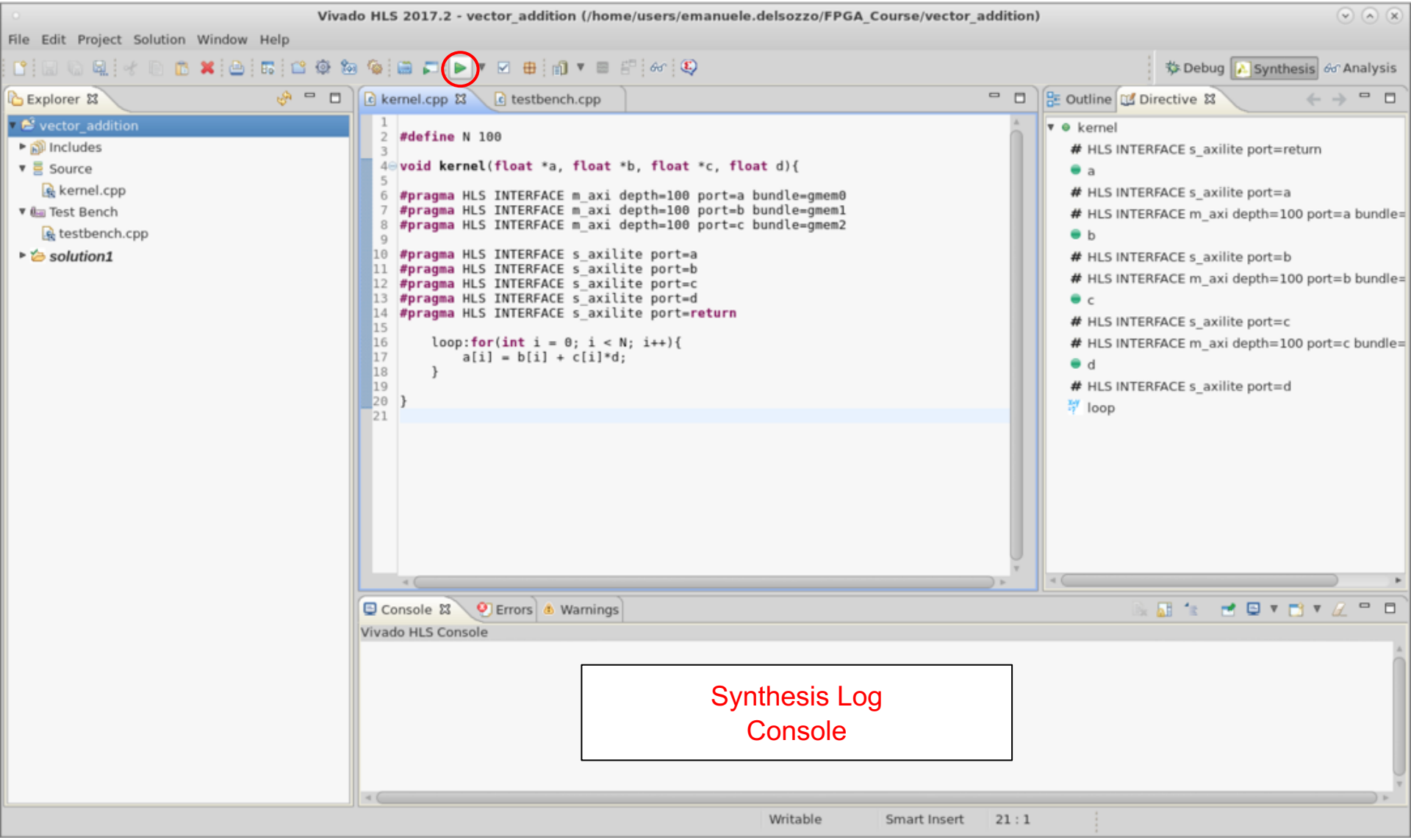
C Simulation Settings



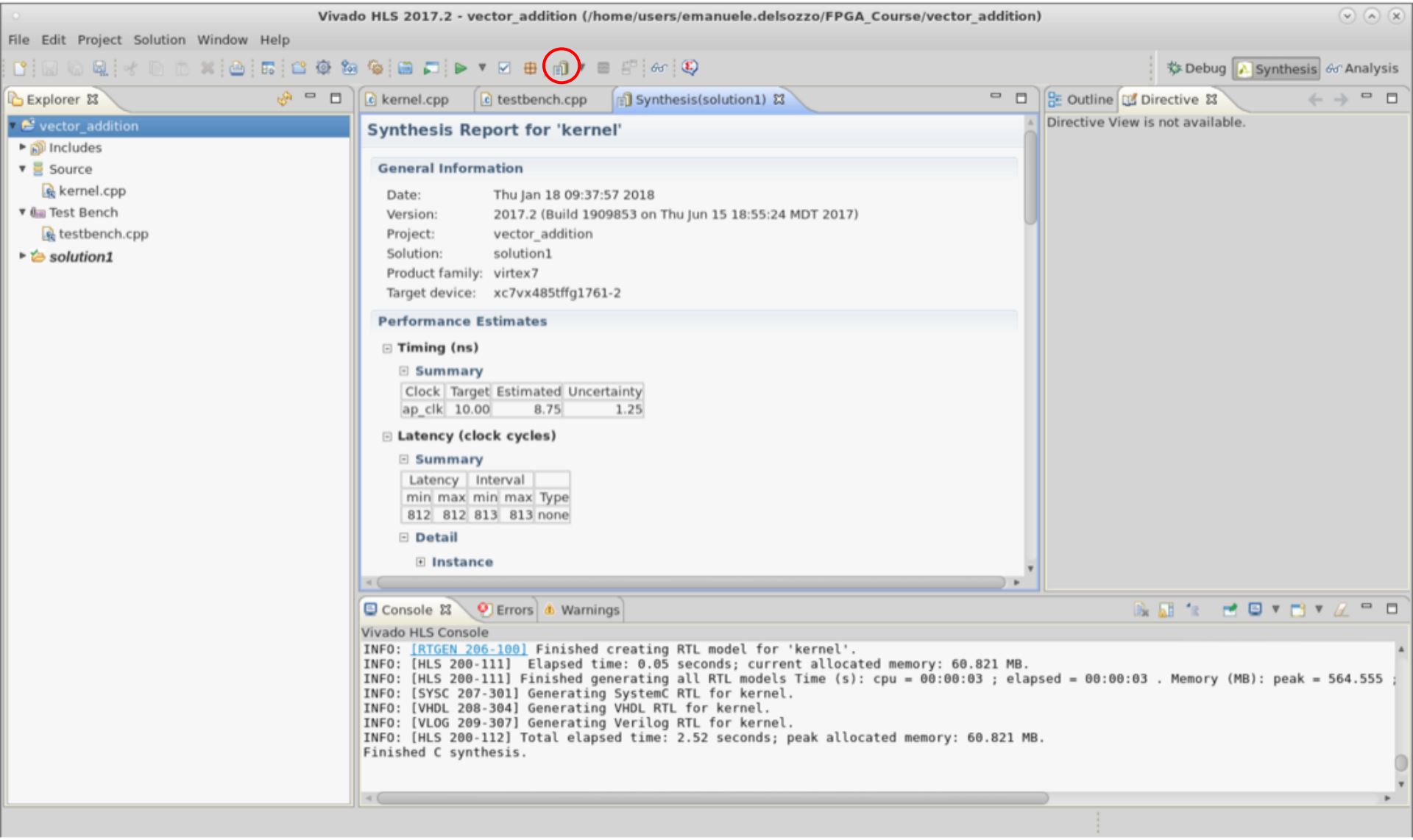
C Simulation Result



Run Synthesis



Synthesis Report



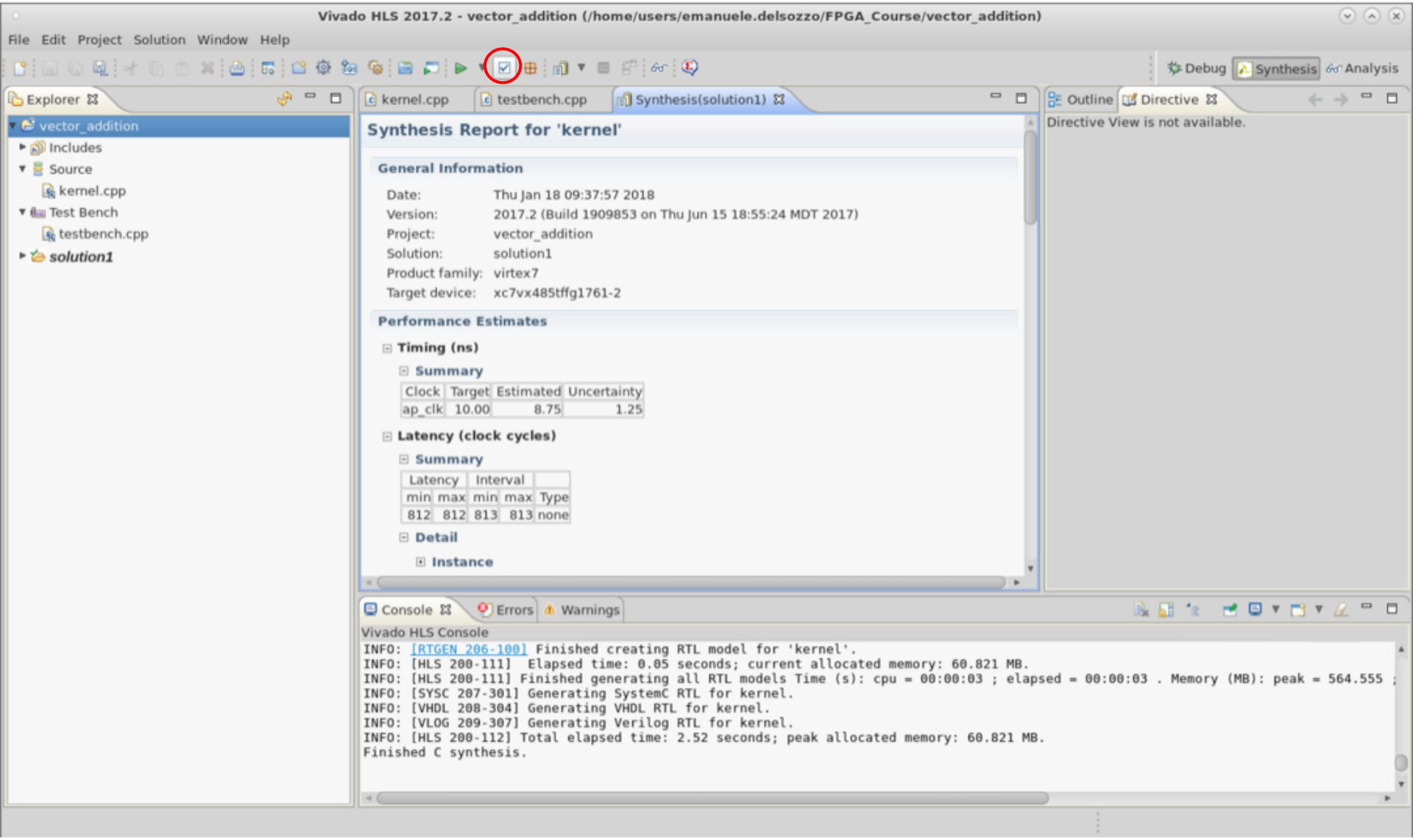
Naive Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.75	1.25

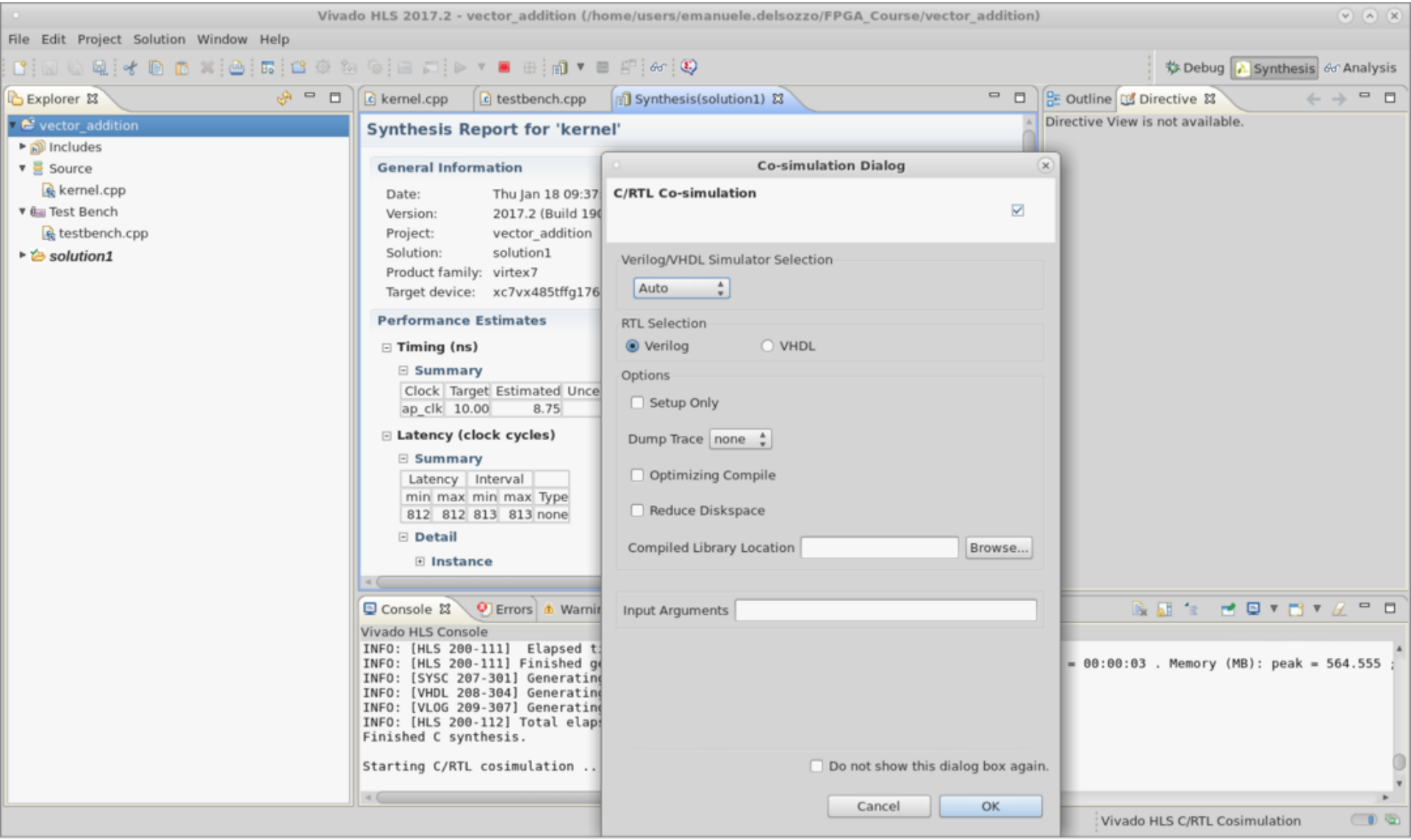
Latency		Interval		
min	max	min	max	Type
812	812	813	813	none

Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	~0%	~0%	~0%	~0%

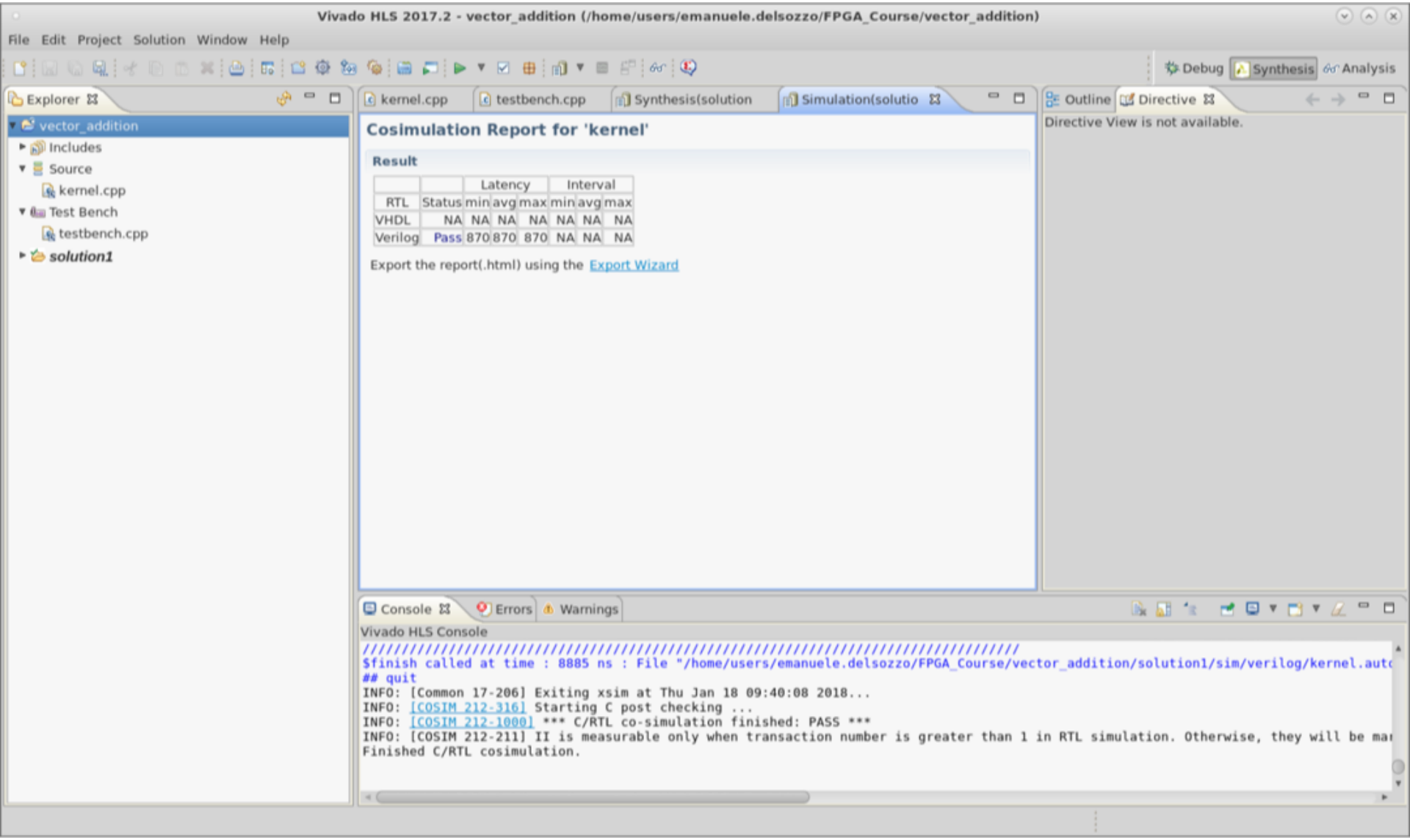
Run Co-Simulation



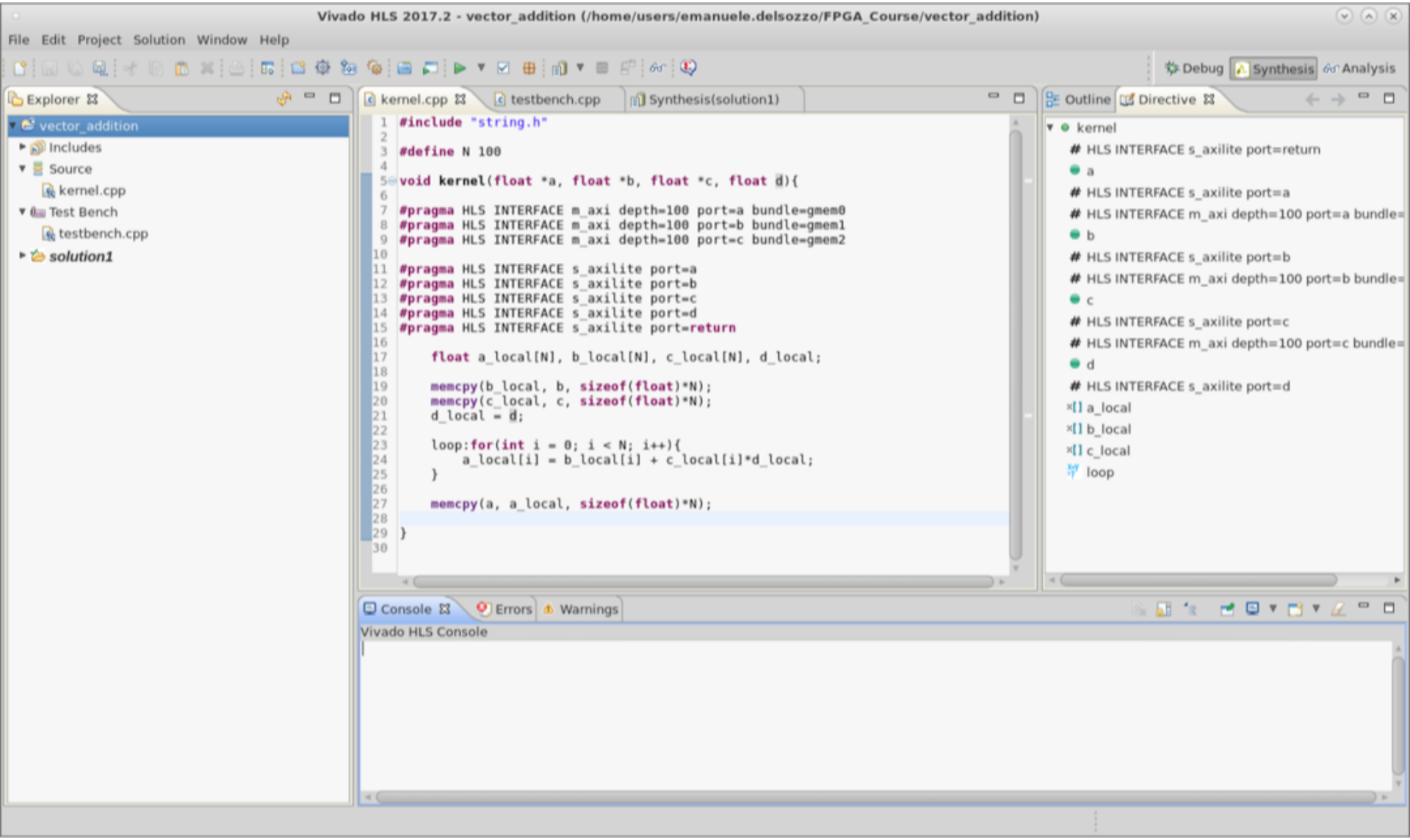
Co-Simulation Settings



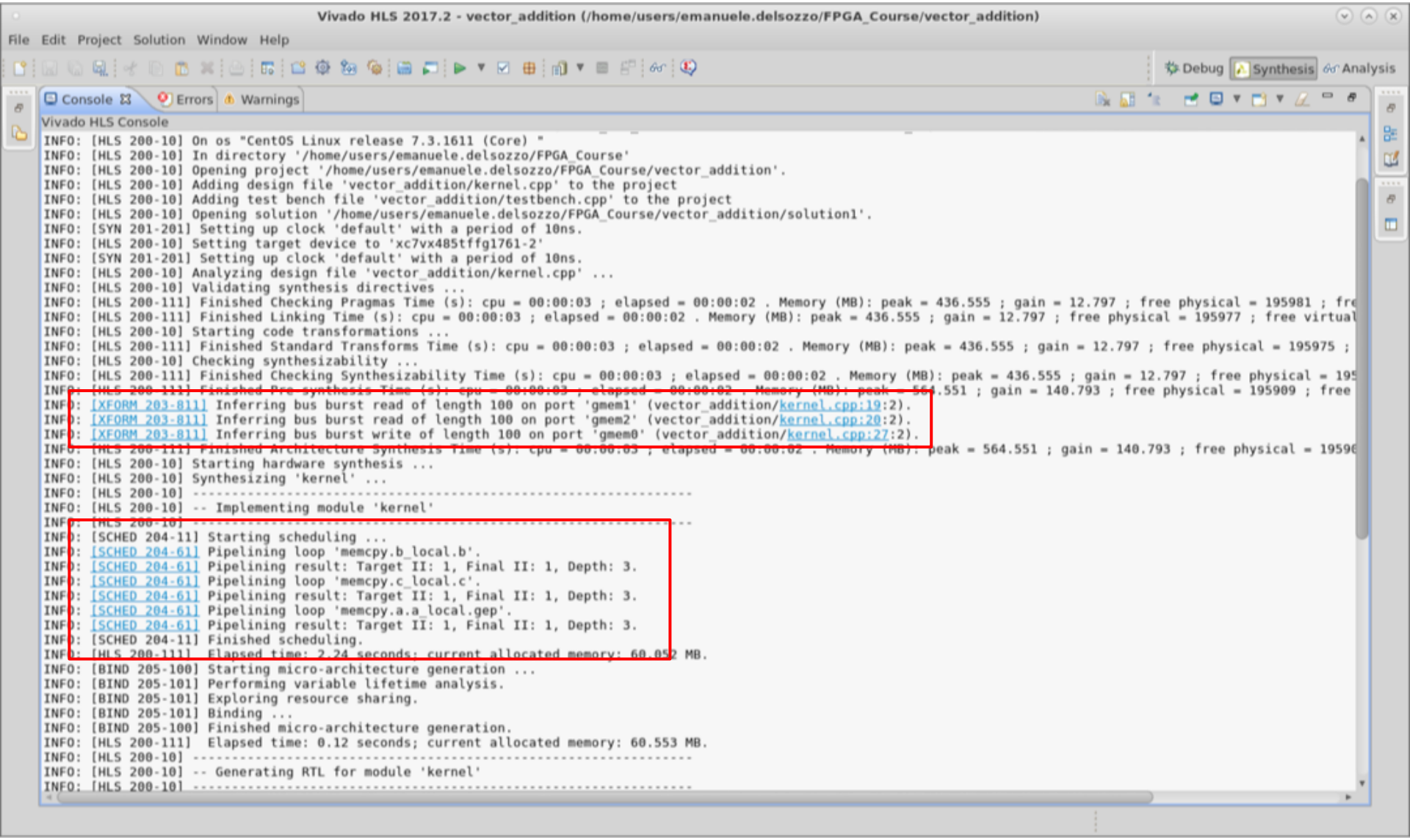
Co-Simulation Results



V1 Implementation



V1 Implementation Log



V1 Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.75	1.25

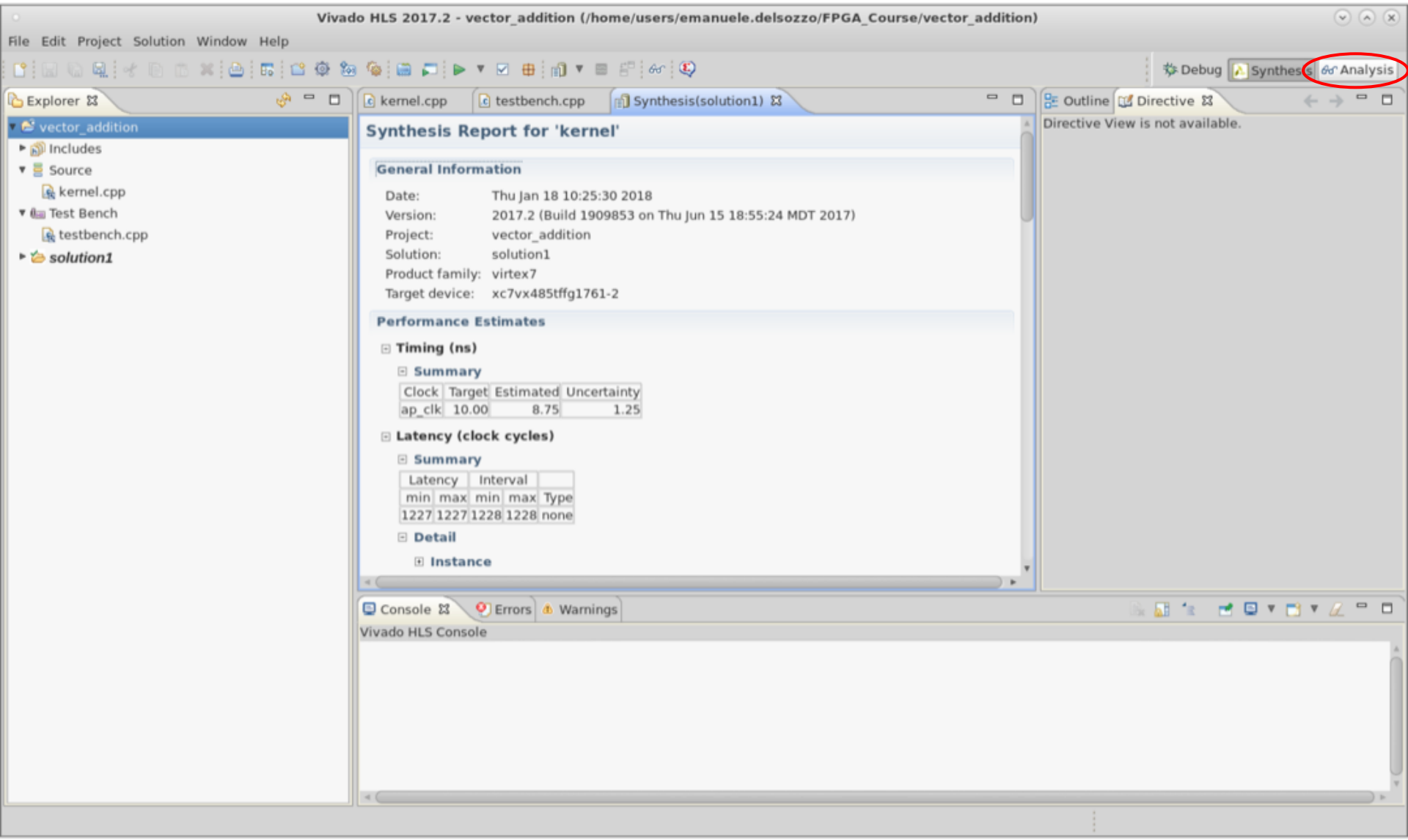
Latency		Interval		
min	max	min	max	Type
1227	1227	1228	1228	none

Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	~0%	~0%	~0%	~0%

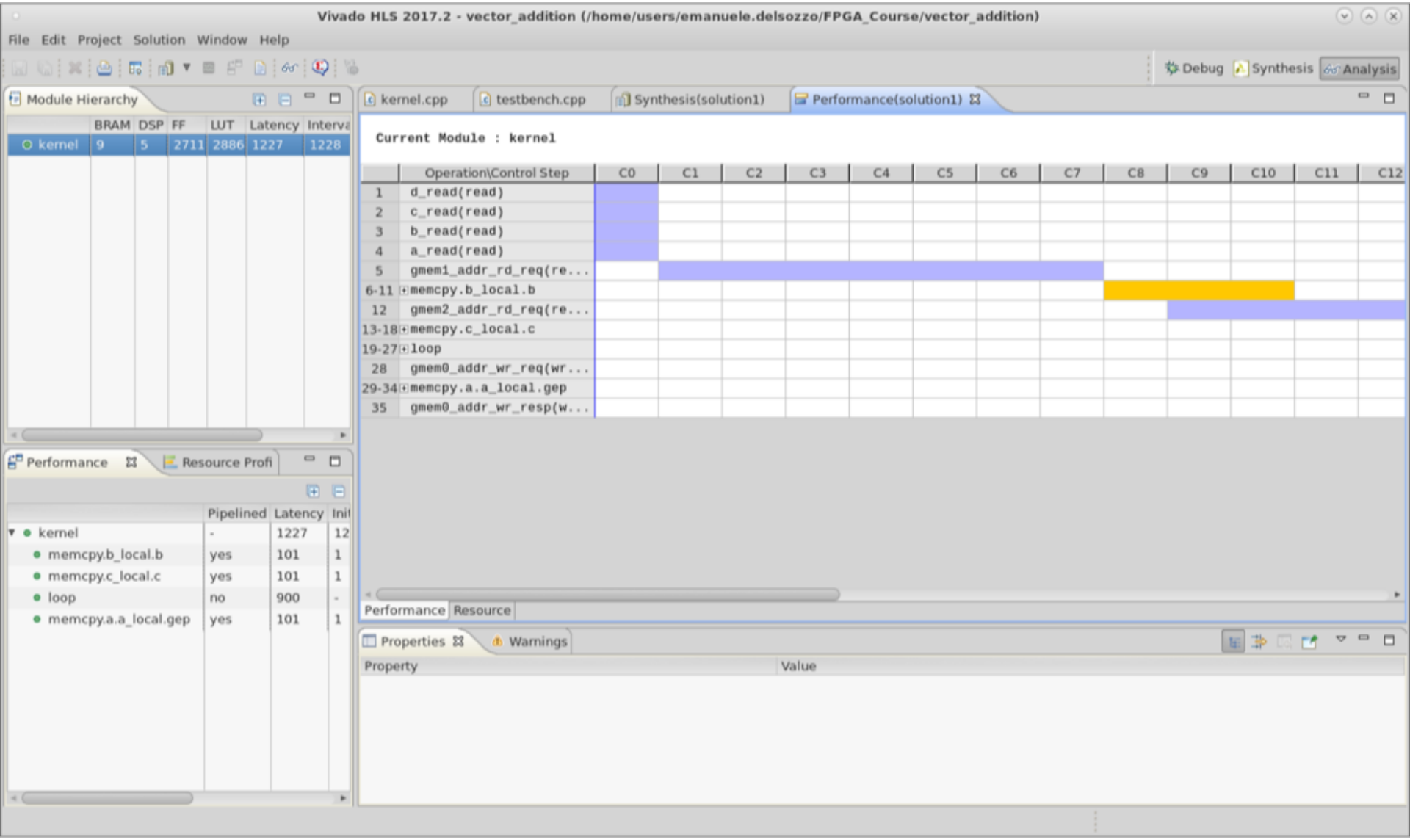
V1 Implementation Loops

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
memcpy b	101	101	3	1	1	100	yes
memcpy c	101	101	3	1	1	100	yes
loop	900	900	9	-	-	100	no
memcpy a	101	101	3	1	1	100	yes

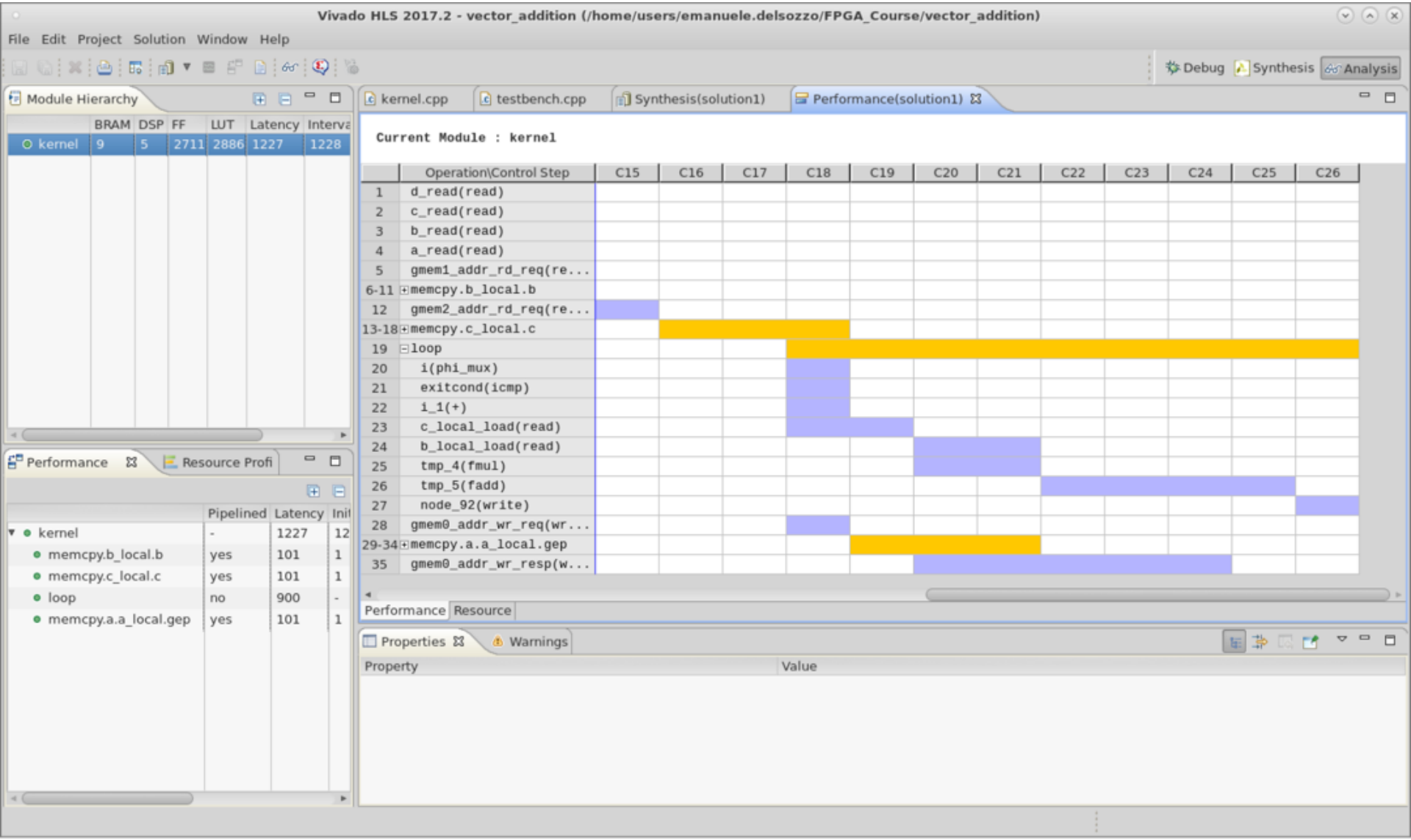
V1 Analysis


Vivado HLS 2017.2 - vector_addition (/home/users/emanuele.delsozzo/FPGA_Course/vector_addition)
File Edit Project Solution Window Help
Debug Synthesis Analysis
Explorer
vector_addition
Includes
Source
kernel.cpp
Test Bench
testbench.cpp
solution1
kernel.cpp testbench.cpp Synthesis(solution1)
Synthesis Report for 'kernel'
General Information
Date: Thu Jan 18 10:25:30 2018
Version: 2017.2 (Build 1909853 on Thu Jun 15 18:55:24 MDT 2017)
Project: vector_addition
Solution: solution1
Product family: virtex7
Target device: xc7vx485tffg1761-2
Performance Estimates
Timing (ns)
Summary
Clock Target Estimated Uncertainty
ap_clk 10.00 8.75 1.25
Latency (clock cycles)
Summary
Latency Interval
min max min max Type
1227 1227 1228 1228 none
Detail
Instance
Outline Directive
Directive View is not available.
Console Errors Warnings
Vivado HLS Console

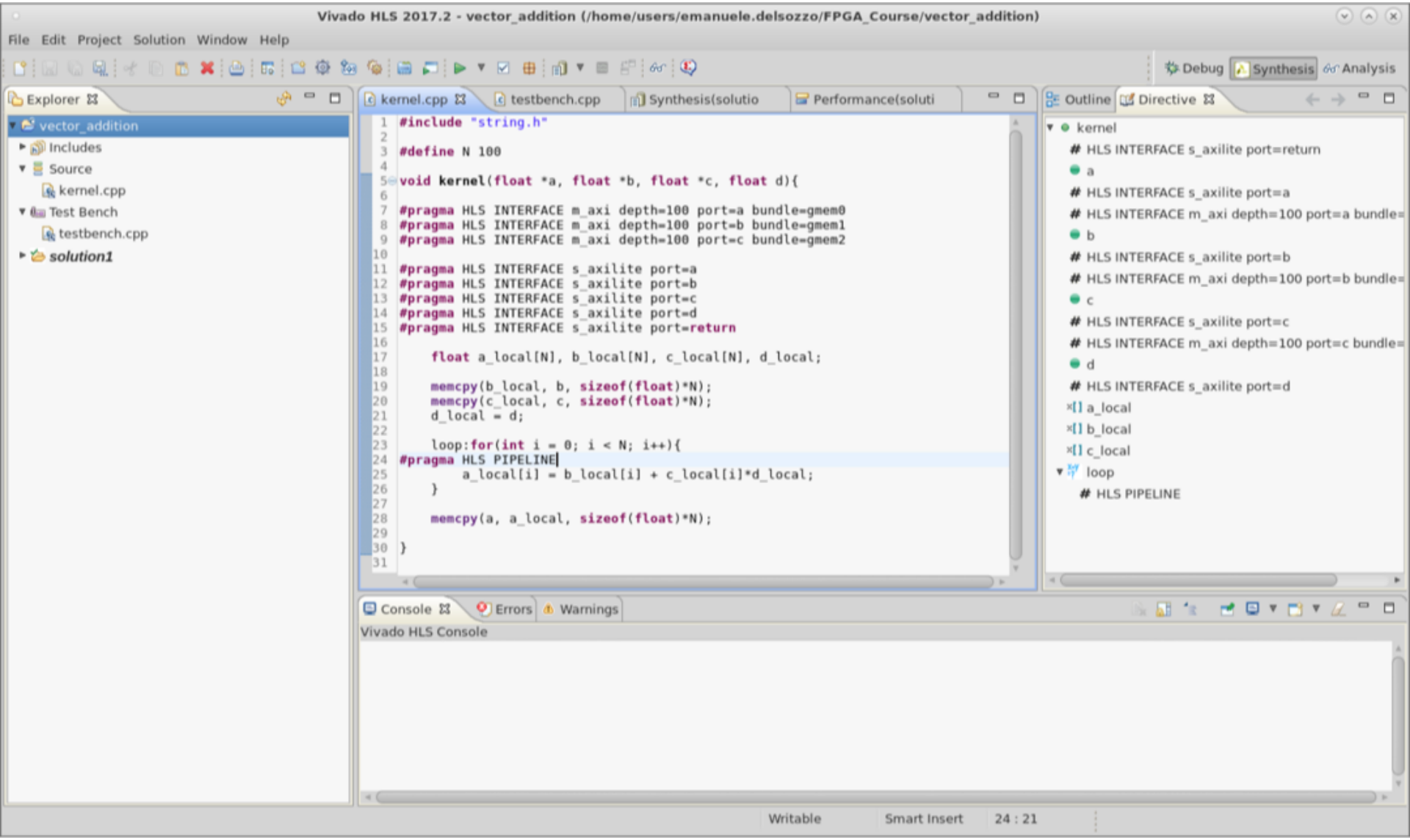
V1 Analysis


Vivado HLS 2017.2 - vector_addition (/home/users/emanuele.delsozzo/FPGA_Course/vector_addition)
File Edit Project Solution Window Help
Debug Synthesis Analysis
Module Hierarchy
BRAM DSP FF LUT Latency Interva
kernel 9 5 2711 2886 1227 1228
kernel.cpp testbench.cpp Synthesis(solution1) Performance(solution1)
Current Module : kernel
Operation\Control Step C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12
1 d_read(read)
2 c_read(read)
3 b_read(read)
4 a_read(read)
5 gmem1_addr_rd_req(re...
6-11 memcpy.b_local.b
12 gmem2_addr_rd_req(re...
13-18 memcpy.c_local.c
19-27 loop
28 gmem0_addr_wr_req(wr...
29-34 memcpy.a.a_local.gep
35 gmem0_addr_wr_resp(w...
Performance Resource
Performance Resource Profi
Pipelined Latency Init
kernel - 1227 12
memcpy.b_local.b yes 101 1
memcpy.c_local.c yes 101 1
loop no 900 -
memcpy.a.a_local.gep yes 101 1
Properties Warnings
Property Value

V1 Analysis


Vivado HLS 2017.2 - vector_addition (/home/users/emanuele.delsozzo/FPGA_Course/vector_addition)
File Edit Project Solution Window Help
Debug Synthesis Analysis
Module Hierarchy
BRAM DSP FF LUT Latency Interva
kernel 9 5 2711 2886 1227 1228
kernel.cpp testbench.cpp Synthesis(solution1) Performance(solution1)
Current Module : kernel
Operation\Control Step C15 C16 C17 C18 C19 C20 C21 C22 C23 C24 C25 C26
1 d_read(read)
2 c_read(read)
3 b_read(read)
4 a_read(read)
5 gmem1_addr_rd_req(re...
6-11 memcpy.b_local.b
12 gmem2_addr_rd_req(re...
13-18 memcpy.c_local.c
19 loop
20 i(phi_mux)
21 exitcond(icmp)
22 i_1(+)
23 c_local_load(read)
24 b_local_load(read)
25 tmp_4(fmul)
26 tmp_5(fadd)
27 node_92(write)
28 gmem0_addr_wr_req(wr...
29-34 memcpy.a.a_local.gep
35 gmem0_addr_wr_resp(w...
Performance Resource Profi
Pipelined Latency Init
kernel - 1227 12
memcpy.b_local.b yes 101 1
memcpy.c_local.c yes 101 1
loop no 900 -
memcpy.a.a_local.gep yes 101 1
Performance Resource
Properties Warnings
Property Value

V2 Implementation



V2 Implementation Log

Vivado HLS 2017.2 - vector_addition (/home/users/emanuele.delsozzo/FPGA_Course/vector_addition)

File Edit Project Solution Window Help

Debug Synthesis Analysis

Console Errors Warnings

Vivado HLS Console

```
Starting C synthesis ...
/xilinx/software/Vivado_HLS/2017.2/bin/vivado_hls /home/users/emanuele.delsozzo/FPGA_Course/vector_addition/solution1/csynth.tcl
INFO: [HLS 200-10] Running '/xilinx/software/Vivado_HLS/2017.2/bin/unwrapped/lnx64.o/vivado_hls'
INFO: [HLS 200-10] For user 'emanuele.delsozzo' on host 'NAGS30' (Linux_x86_64 version 3.10.0-514.16.1.el7.x86_64) on Thu Jan 18 11:02:50 CET 2018
INFO: [HLS 200-10] On os "CentOS Linux release 7.3.1611 (Core) "
INFO: [HLS 200-10] In directory '/home/users/emanuele.delsozzo/FPGA_Course'
INFO: [HLS 200-10] Opening project '/home/users/emanuele.delsozzo/FPGA_Course/vector_addition'.
INFO: [HLS 200-10] Adding design file 'vector_addition/kernel.cpp' to the project
INFO: [HLS 200-10] Adding test bench file 'vector_addition/testbench.cpp' to the project
INFO: [HLS 200-10] Opening solution '/home/users/emanuele.delsozzo/FPGA_Course/vector_addition/solution1'.
INFO: [SYN 201-201] Setting up clock 'default' with a period of 10ns.
INFO: [HLS 200-10] Setting target device to 'xc7vx485tffg1761-2'
INFO: [HLS 200-10] Analyzing design file 'vector_addition/kernel.cpp' ...
INFO: [HLS 200-10] Validating synthesis directives ...
INFO: [HLS 200-111] Finished Checking Pragmas Time (s): cpu = 00:00:02 ; elapsed = 00:00:02 . Memory (MB): peak = 436.559 ; gain = 12.797 ; free physical = 194630 ; fre
INFO: [HLS 200-111] Finished Linking Time (s): cpu = 00:00:02 ; elapsed = 00:00:02 . Memory (MB): peak = 436.559 ; gain = 12.797 ; free physical = 194626 ; free virtual
INFO: [HLS 200-10] Starting code transformations ...
INFO: [HLS 200-111] Finished Standard Transforms Time (s): cpu = 00:00:03 ; elapsed = 00:00:02 . Memory (MB): peak = 436.559 ; gain = 12.797 ; free physical = 194619 ;
INFO: [HLS 200-10] Checking synthesizability ...
INFO: [HLS 200-111] Finished Checking Synthesizability Time (s): cpu = 00:00:03 ; elapsed = 00:00:02 . Memory (MB): peak = 436.559 ; gain = 12.797 ; free physical = 194
INFO: [HLS 200-111] Finished Pre-synthesis Time (s): cpu = 00:00:03 ; elapsed = 00:00:02 . Memory (MB): peak = 564.555 ; gain = 140.793 ; free physical = 194599 ; free
INFO: [XFORM 203-811] Inferring bus burst read of length 100 on port 'gmem1' (vector_addition/kernel.cpp:26:2).
INFO: [XFORM 203-811] Inferring bus burst read of length 100 on port 'gmem2' (vector_addition/kernel.cpp:27:2).
INFO: [XFORM 203-811] Inferring bus burst write of length 100 on port 'gmem0' (vector_addition/kernel.cpp:37:2).
INFO: [HLS 200-111] Finished Architecture Synthesis Time (s): cpu = 00:00:03 ; elapsed = 00:00:02 . Memory (MB): peak = 564.555 ; gain = 140.793 ; free physical = 19466
INFO: [HLS 200-10] Starting hardware synthesis ...
INFO: [HLS 200-10] Synthesizing 'kernel' ...
INFO: [HLS 200-10] ----------------------------------------------------------------
INFO: [HLS 200-10] -- Implementing module 'kernel'
INFO: [HLS 200-10] ----------------------------------------------------------------
INFO: [SCHED 204-11] Starting scheduling ...
INFO: [SCHED 204-61] Pipelining loop 'memcpy.b_local.b'.
INFO: [SCHED 204-61] Pipelining result: Target II: 1, Final II: 1, Depth: 3.
INFO: [SCHED 204-61] Pipelining loop 'memcpy.c_local.c'.
INFO: [SCHED 204-61] Pipelining result: Target II: 1, Final II: 1, Depth: 3.
INFO: [SCHED 204-61] Pipelining loop 'loop'.
INFO: [SCHED 204-61] Pipelining result: Target II: 1, Final II: 1, Depth: 9.
INFO: [SCHED 204-61] Pipelining loop 'memcpy.a.a_local.gep'.
INFO: [SCHED 204-61] Pipelining result: Target II: 1, Final II: 1, Depth: 3.
INFO: [SCHED 204-11] Finished scheduling.
INFO: [HLS 200-111]  Elapsed time: 2.13 seconds; current allocated memory: 60.078 MB.
INFO: [BIND 205-100] Starting micro-architecture generation ...
INFO: [BIND 205-101] Performing variable lifetime analysis.
INFO: [BIND 205-101] Exploring resource sharing.
INFO: [BIND 205-101] Binding ...
```

V2 Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.75	1.25

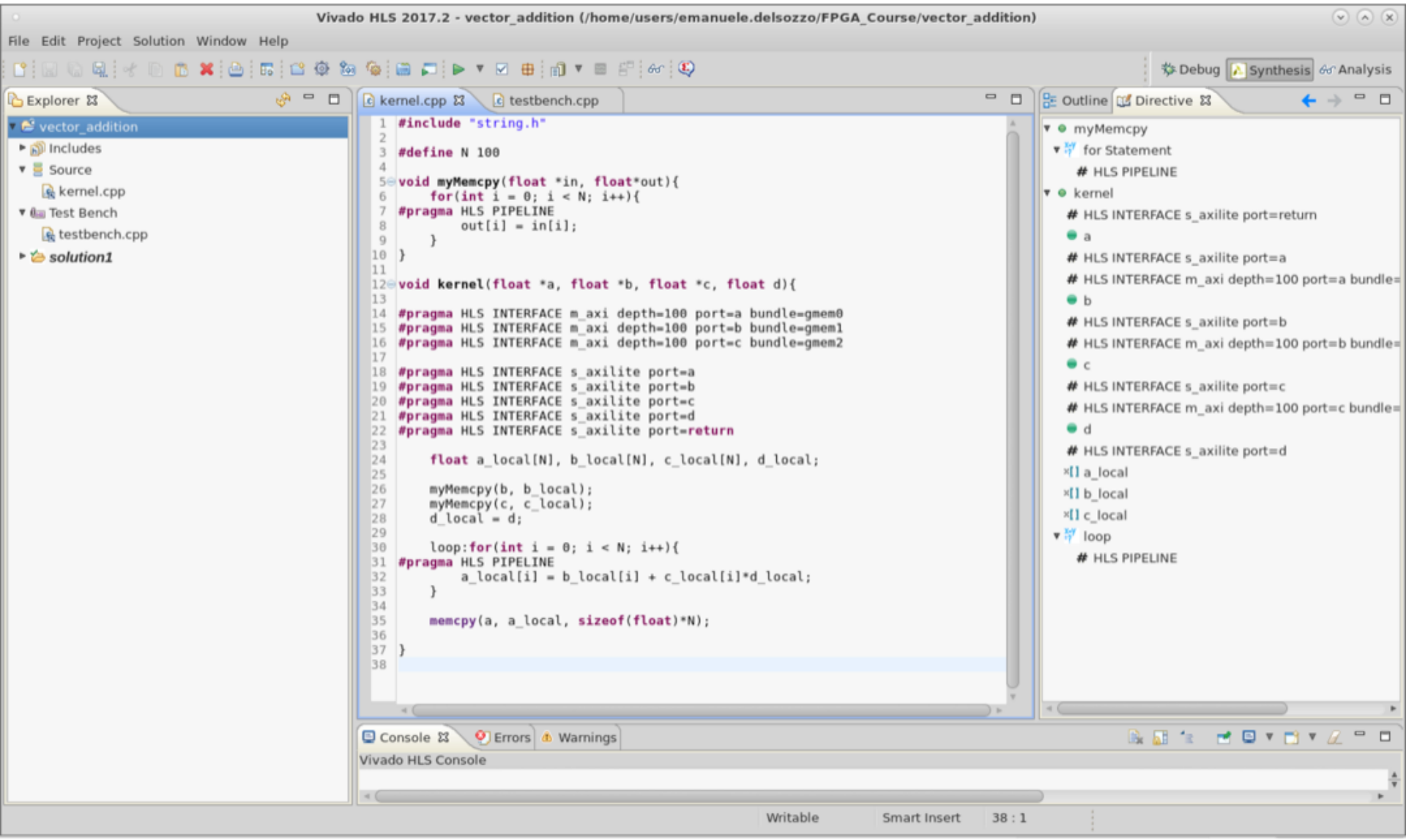
Latency		Interval		
min	max	min	max	Type
435	435	436	436	none

Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	~0%	~0%	~0%	~0%

V2 Implementation Loops

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
memcpy b	101	101	3	1	1	100	yes
memcpy c	101	101	3	1	1	100	yes
loop	107	107	9	1	1	100	yes
memcpy a	101	101	3	1	1	100	yes

V3 Implementation



V3 Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.75	1.25

Latency		Interval		
min	max	min	max	Type
326	326	327	327	none

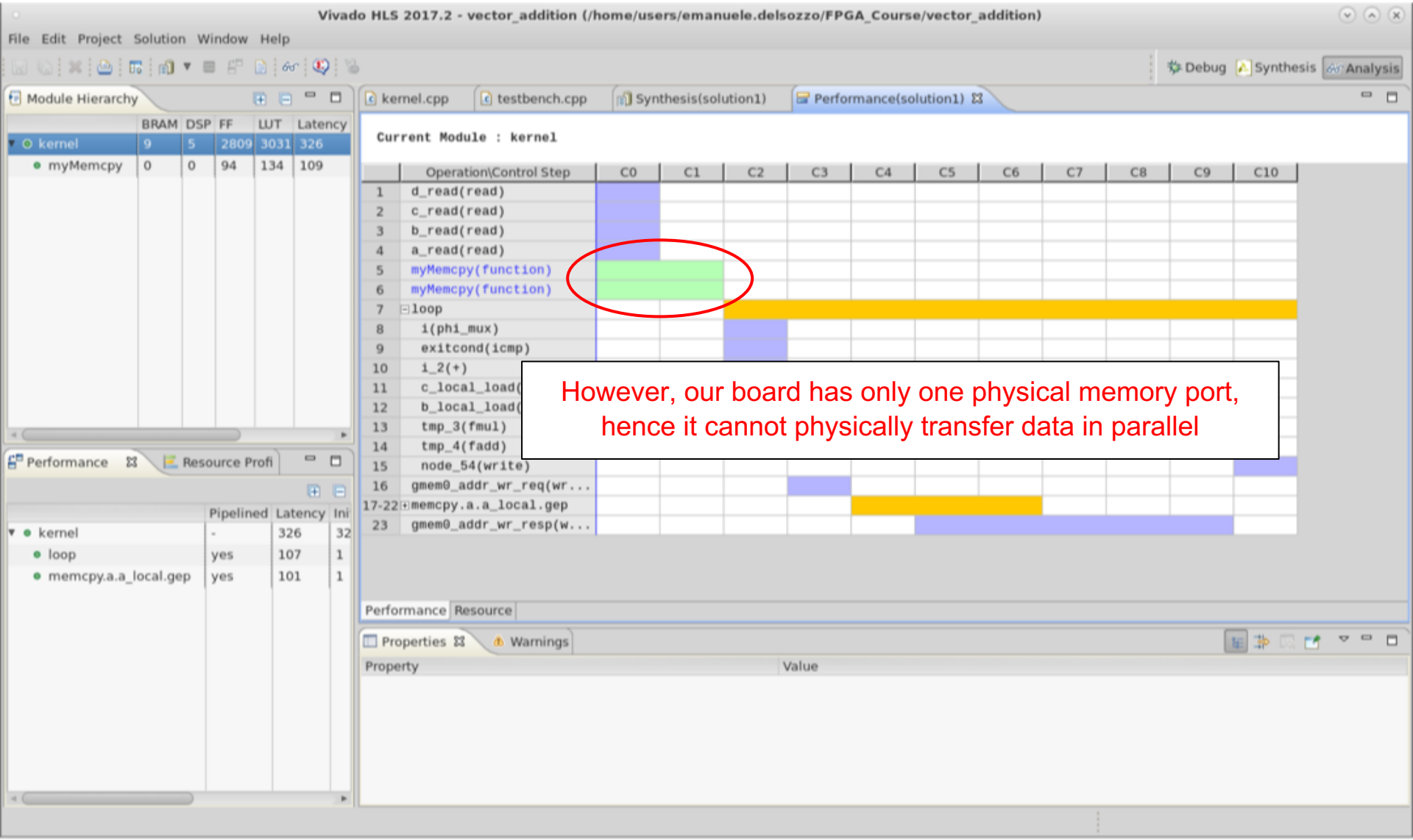
Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	~0%	~0%	~0%	~0%

V3 Implementation Loops

	Latency		Initiation Interval		
Module	min	max	min	max	Type
myMemcpy	109	109	109	109	None
myMemcpy	109	109	109	109	None

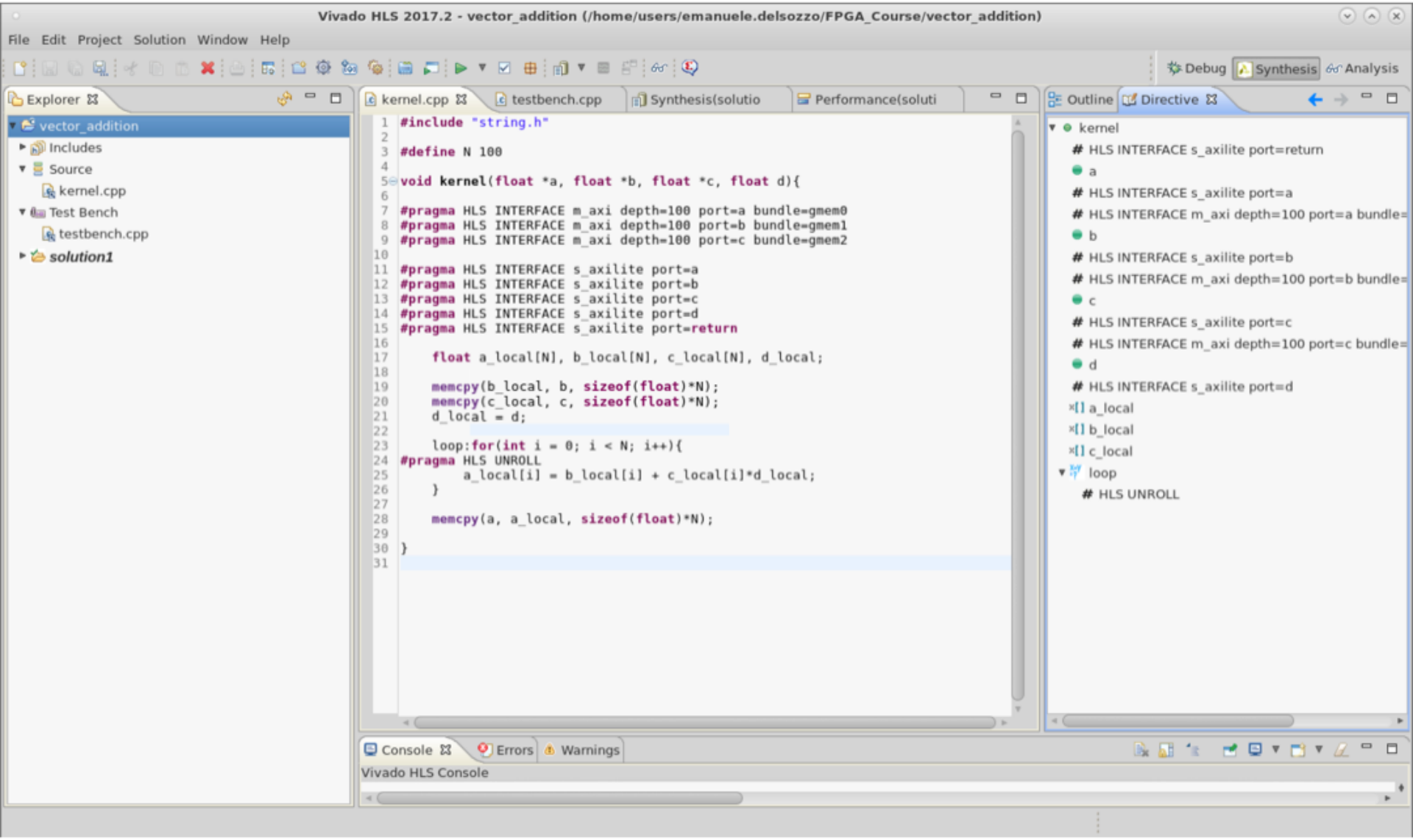
	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
loop	107	107	9	1	1	100	yes
memcpy a	101	101	3	1	1	100	yes

V3 Implementation Analysis



However, our board has only one physical memory port, hence it cannot physically transfer data in parallel

V4 Implementation



V4 Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.75	1.25

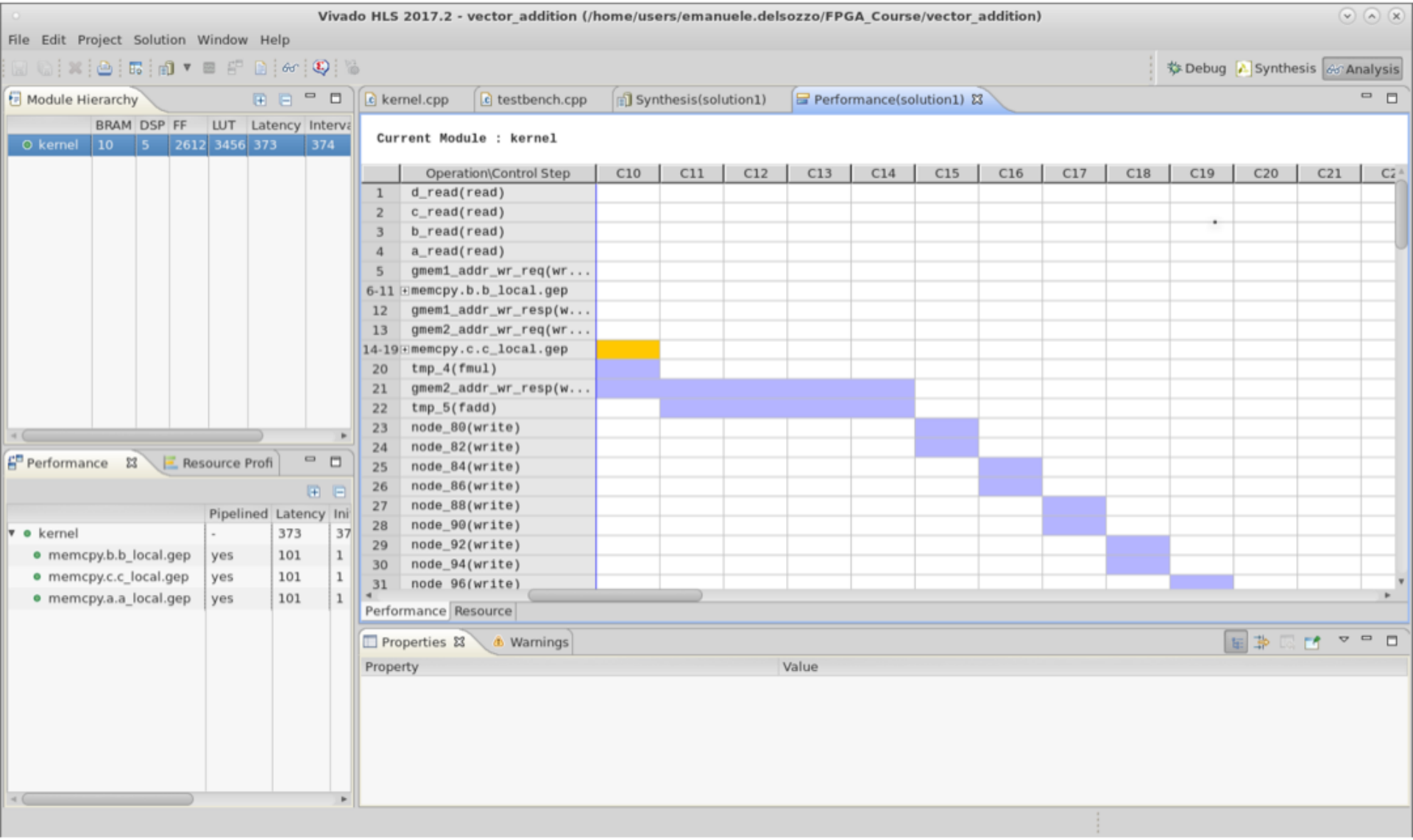
Latency		Interval		
min	max	min	max	Type
373	373	374	374	none

Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	~0%	~0%	~0%	1%

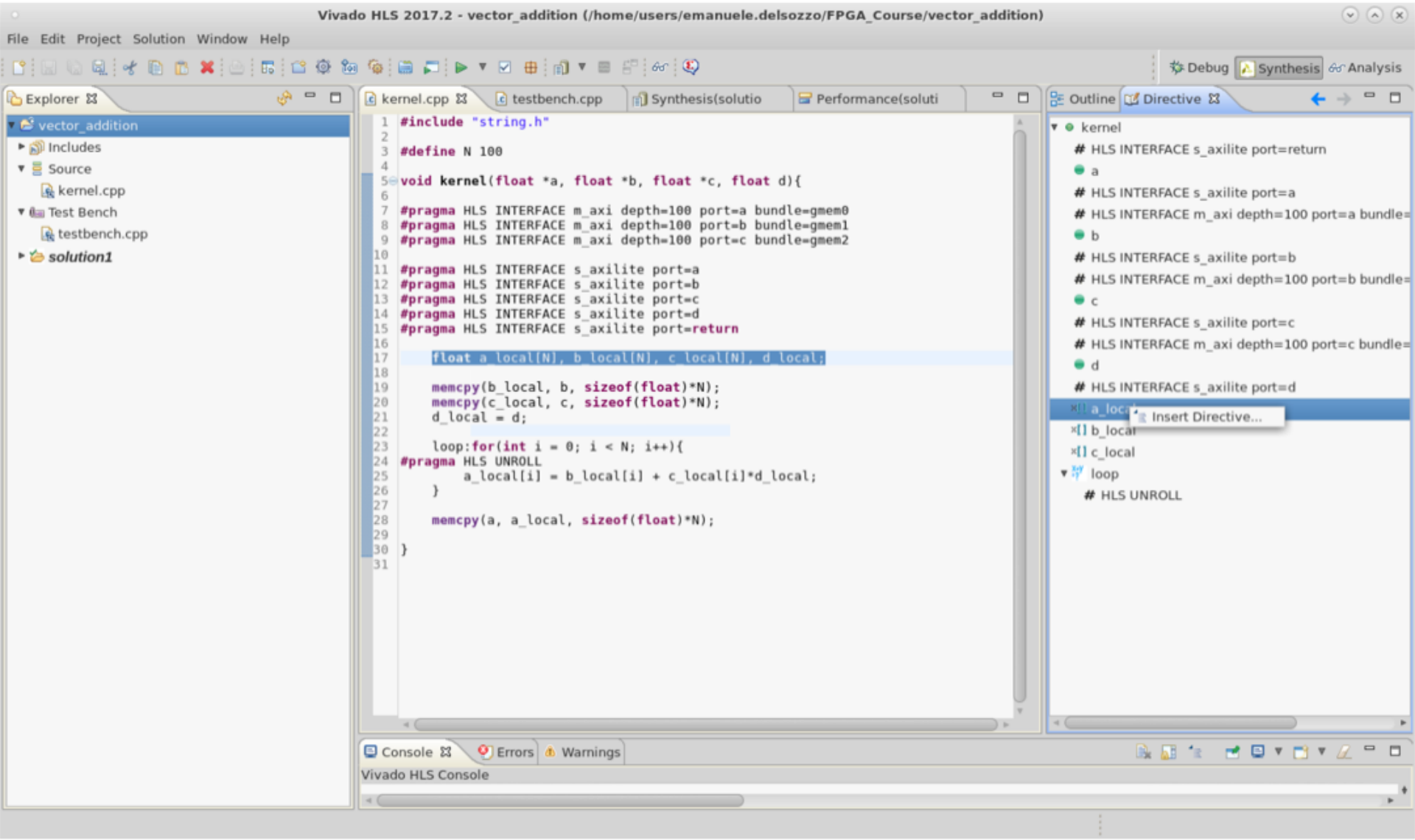
V4 Implementation Loops

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
memcpy b	101	101	3	1	1	100	yes
memcpy c	101	101	3	1	1	100	yes
memcpy a	101	101	3	1	1	100	yes

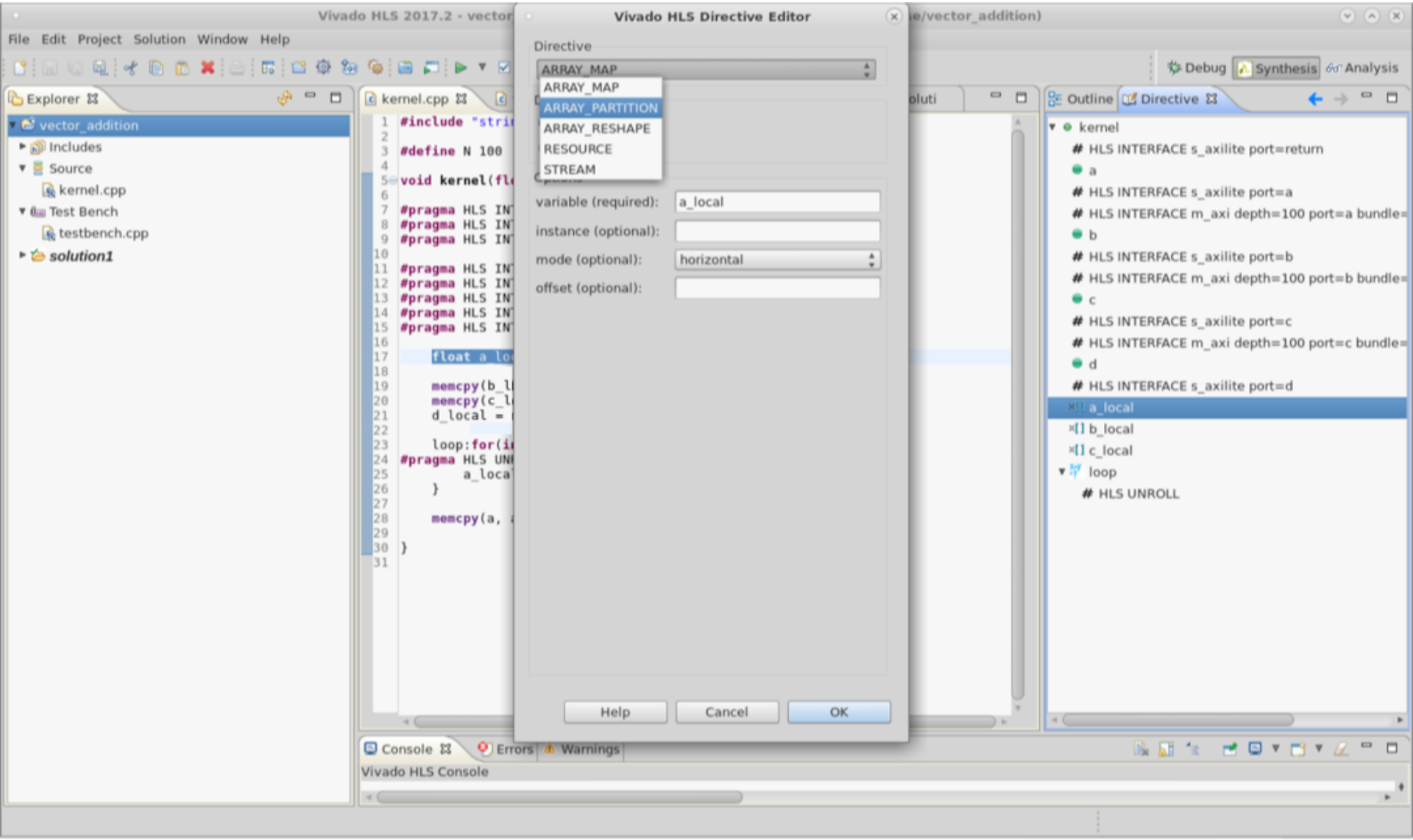
V4 Implementation Analysis



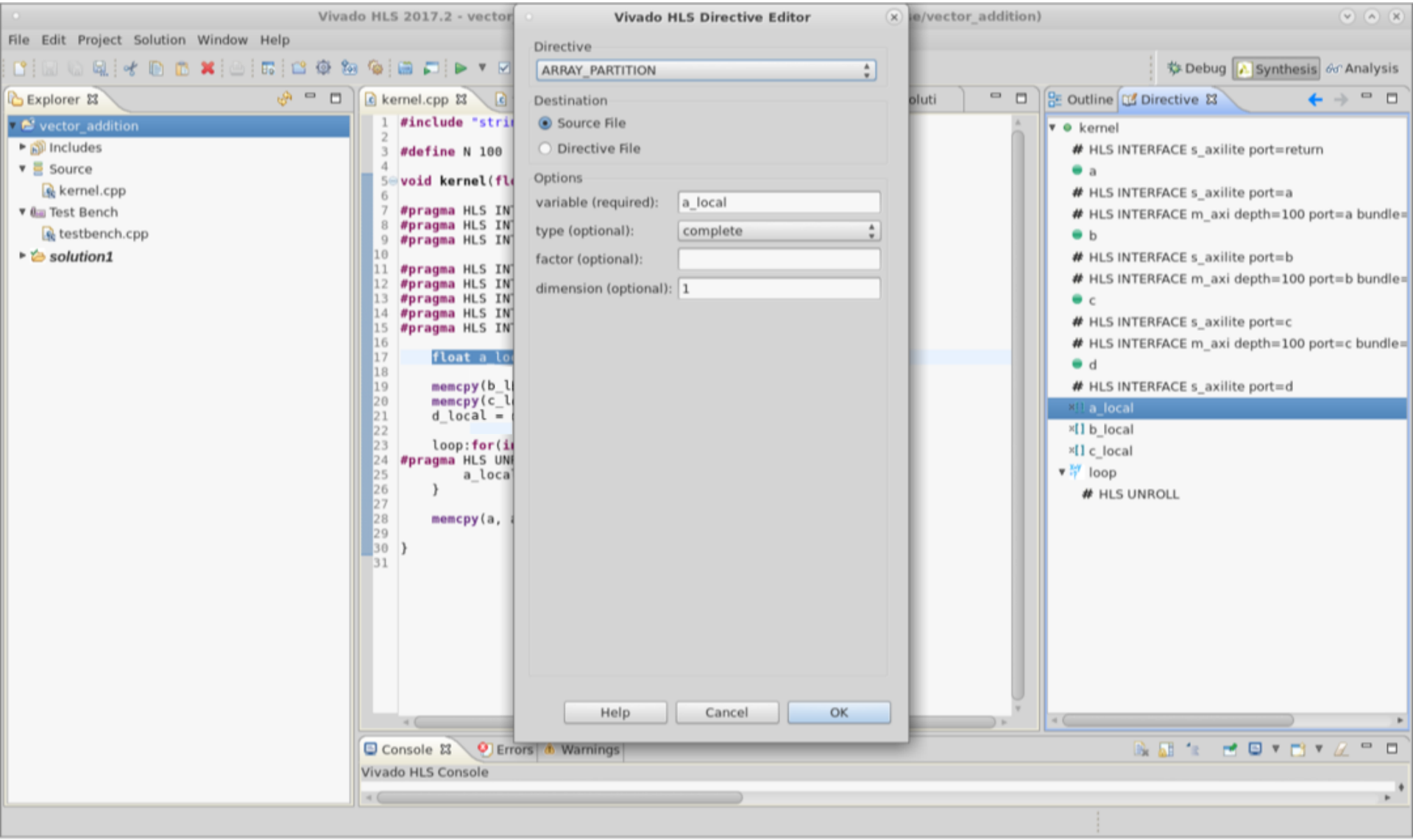
Insert Directive for a_local



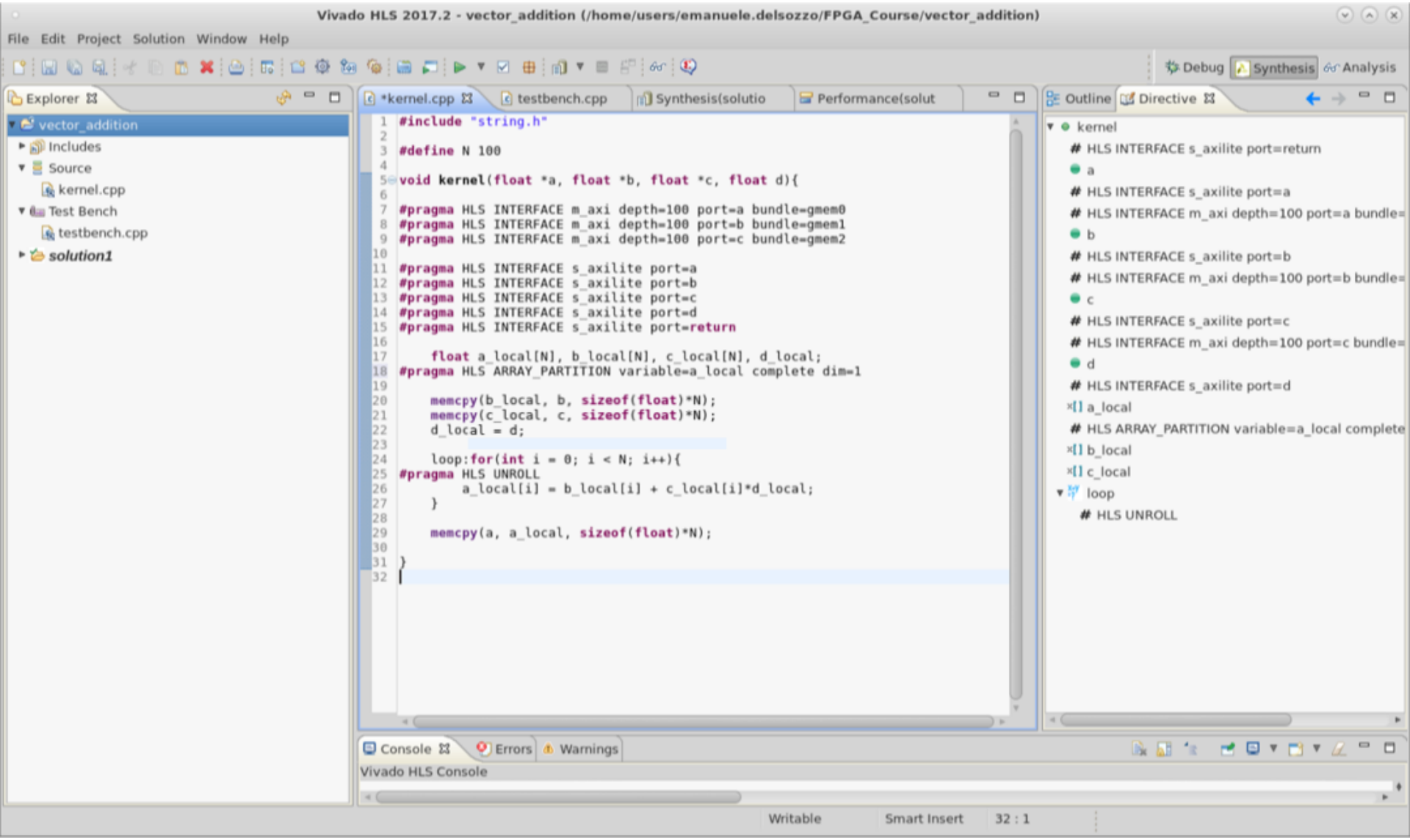
Array Directives



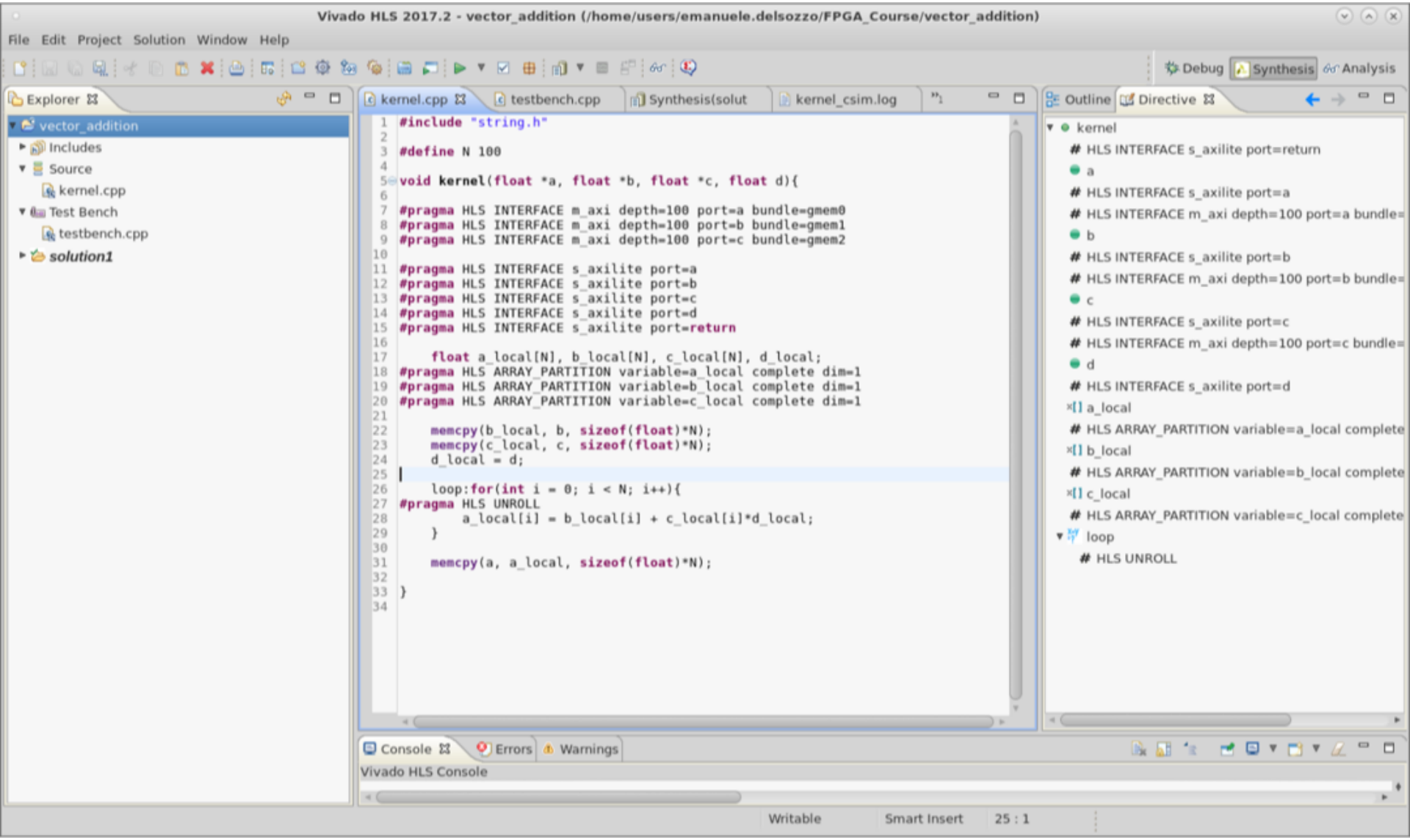
Array Partition Directive



Array a_local directive



V5 Implementation



V5 Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.75	1.25

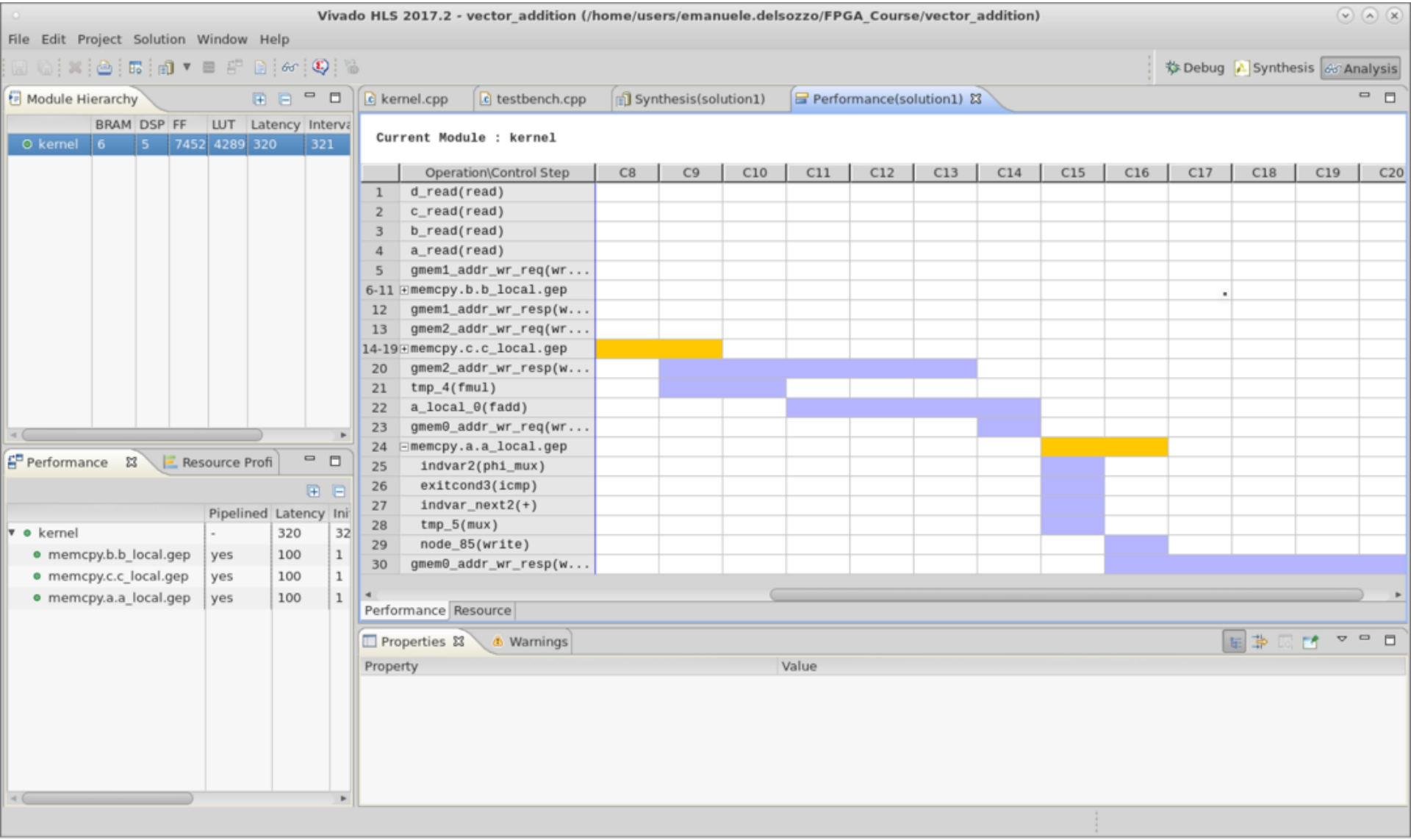
Latency		Interval		
min	max	min	max	Type
320	320	321	321	none

Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	~0%	17%	8%	12%

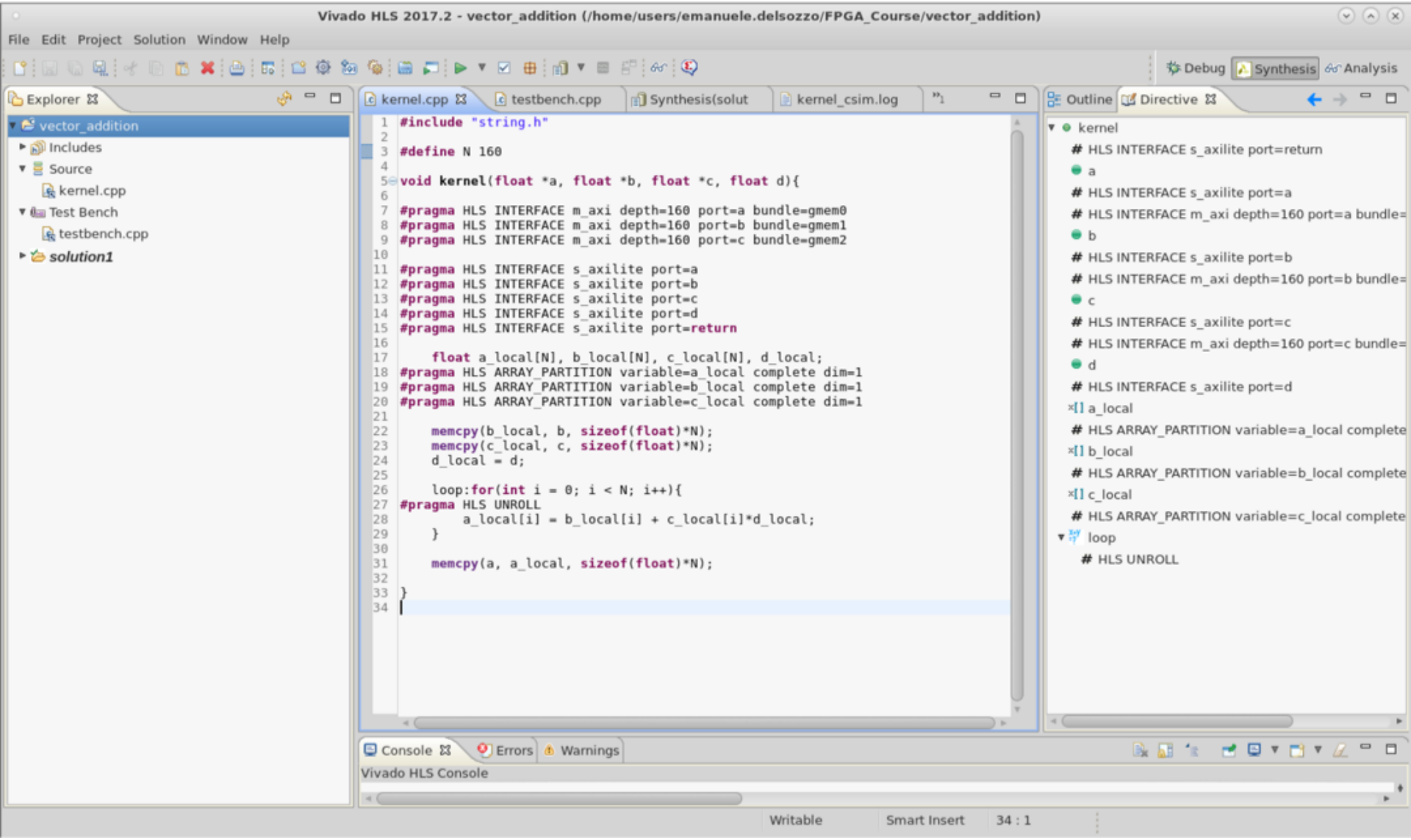
V5 Implementation Loops

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
memcpy b	101	101	3	1	1	100	yes
memcpy c	101	101	3	1	1	100	yes
memcpy a	101	101	3	1	1	100	yes

V5 Implementation Analysis



V5* Implementation



V5* Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.75	1.25

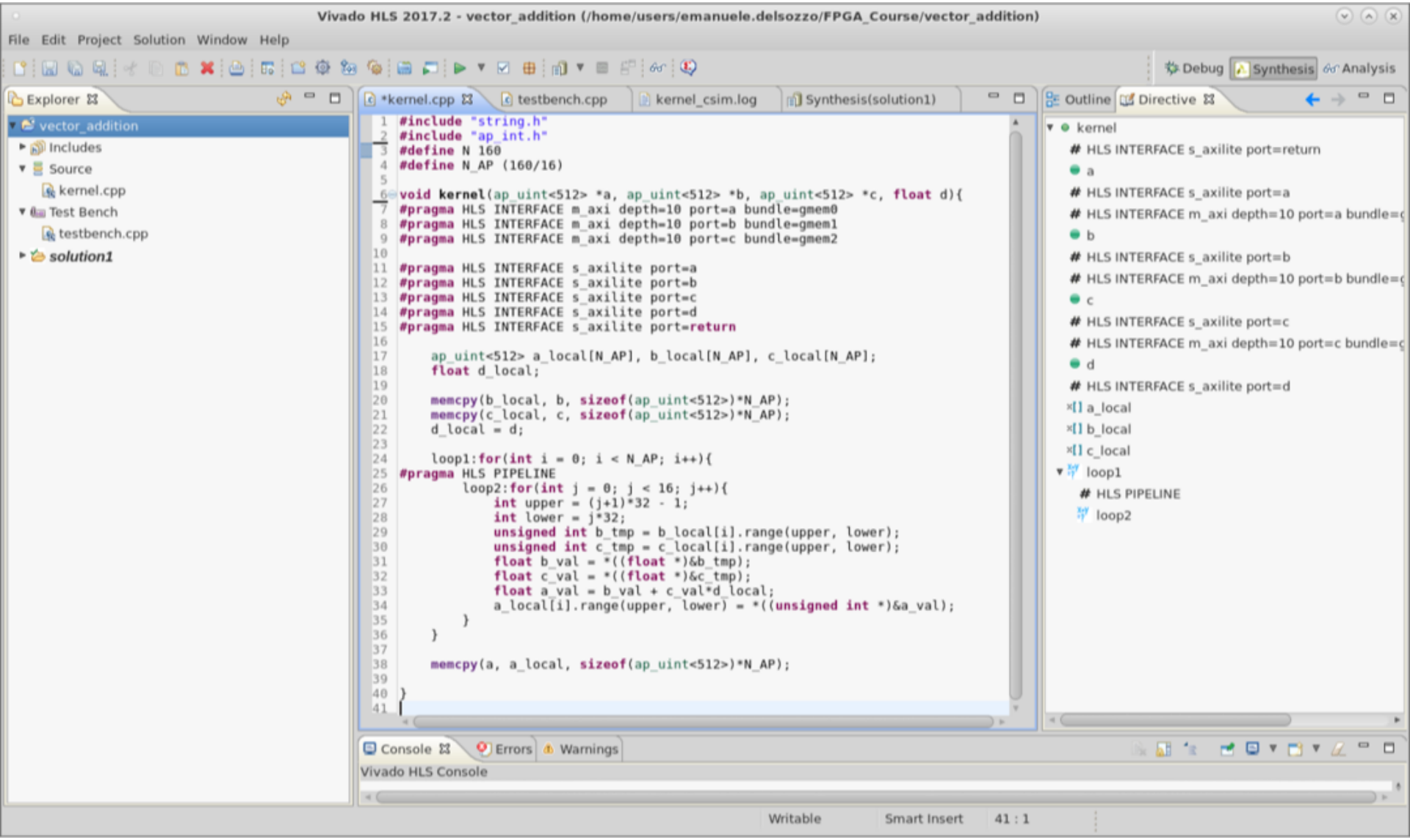
Latency		Interval		
min	max	min	max	Type
508	508	509	509	none

Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	~0%	28%	13%	19%

V5* Implementation Loops

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
memcpy b	160	160	2	1	1	100	yes
memcpy c	160	160	2	1	1	100	yes
memcpy a	160	160	2	1	1	100	yes

V6 Implementation



V6 Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.75	1.25

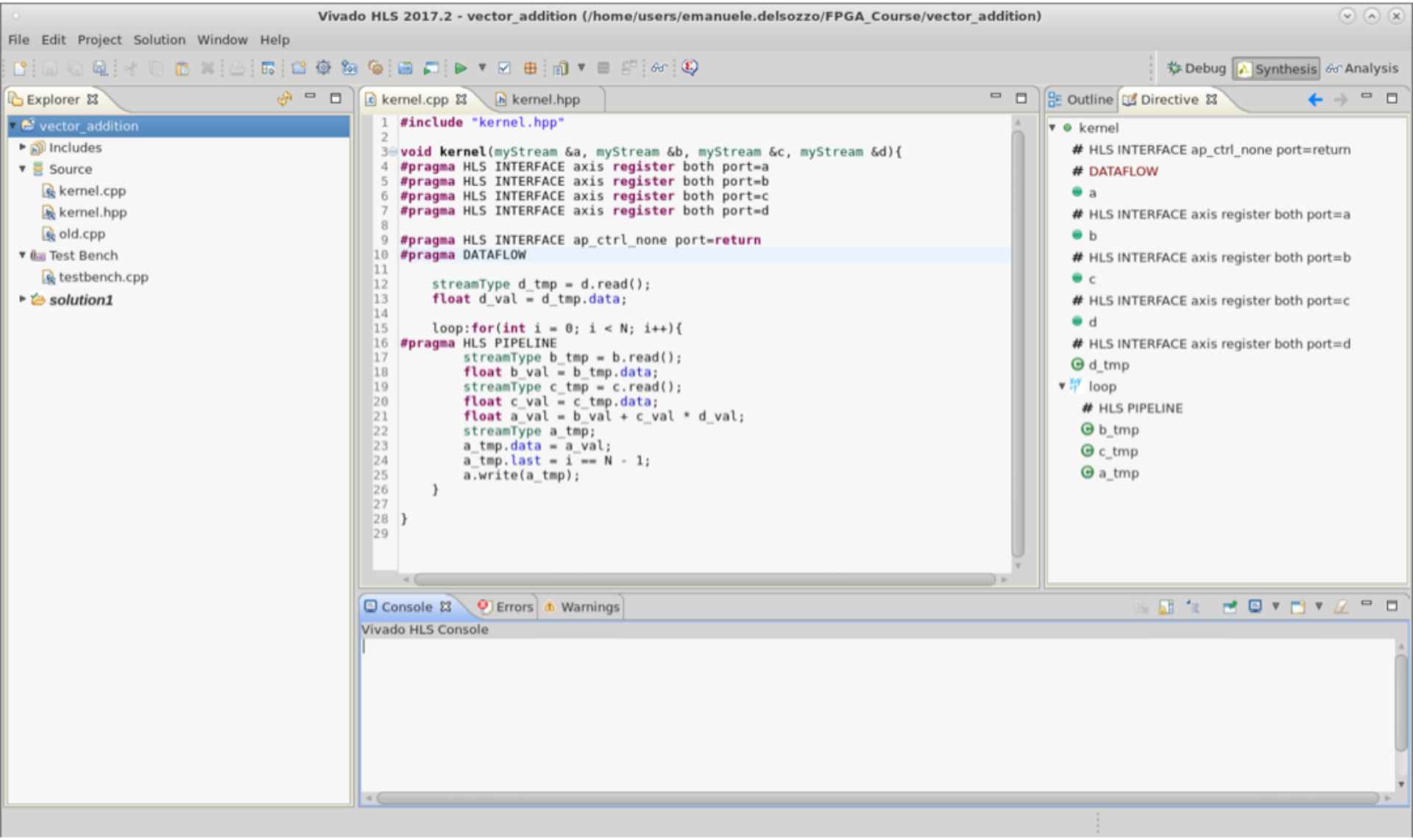
Latency		Interval		
min	max	min	max	Type
75	75	76	76	none

Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	6%	2%	2%	3%

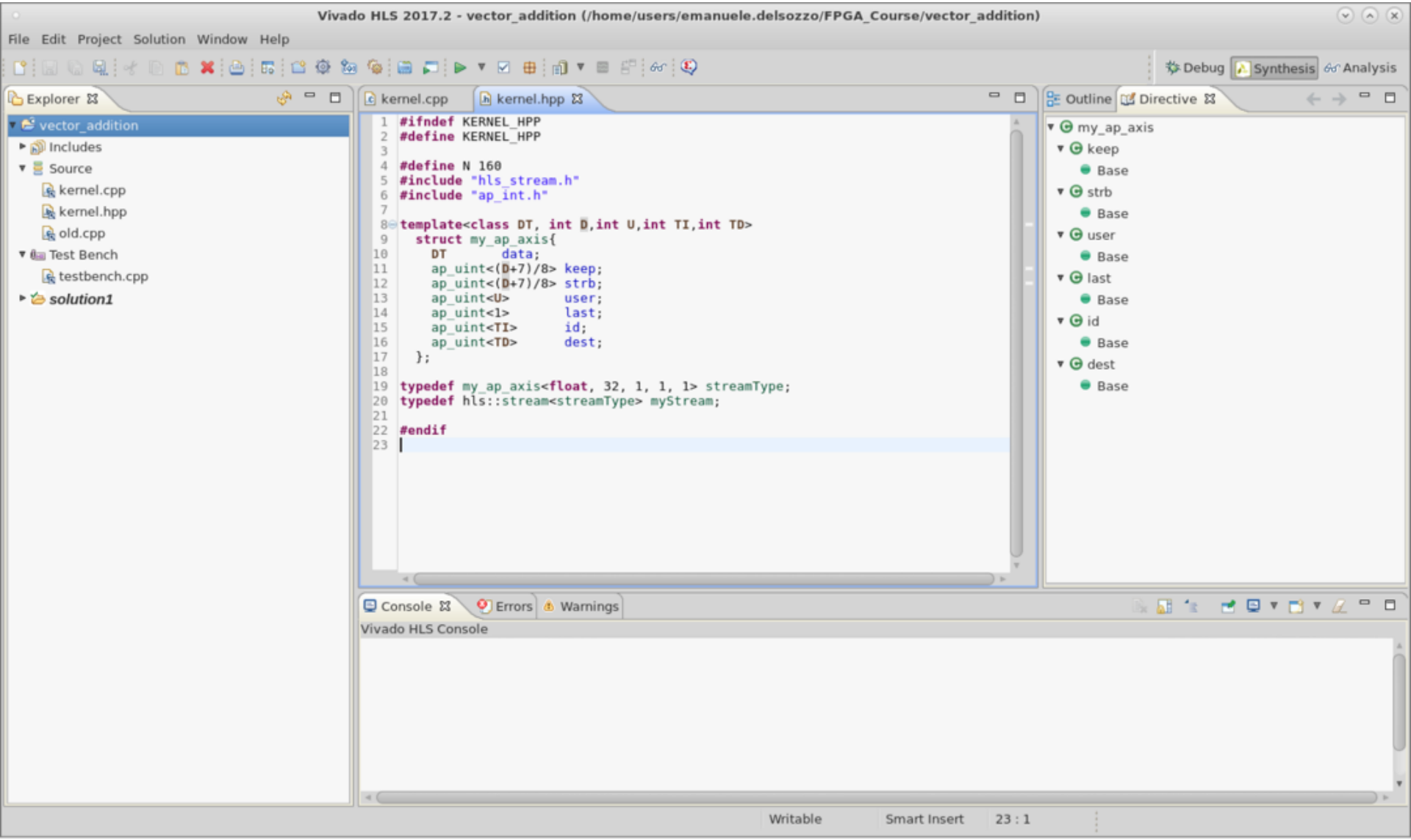
V6 Implementation Loops

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
memcpy b	11	11	3	1	1	10	yes
memcpy c	11	11	3	1	1	10	yes
loop1	17	17	9	1	1	10	yes
memcpy a	11	11	3	1	1	10	yes

V7 Implementation



Streaming Interface



V7 Testbench

Vivado HLS 2017.2 - vector_addition (/home/users/emanuele.delsozzo/FPGA_Course/vector_addition)

File Edit Project Solution Window Help

Debug Synthesis Analysis

kernel.cpp kernel.hpp *testbench.cpp

```
3  #include "kernel.hpp"
4  #define X 20
5
6  void kernel(myStream &a, myStream &b, myStream &c, myStream &d);
7
8  int main(){
9
10     float aCPU[N], aFPGA[N], b[N], c[N], d;
11     myStream aStream, bStream, cStream, dStream;
12     streamType aType, bType, cType, dType;
13
14     srand(0);
15     d = (float) rand() / ((float) (RAND_MAX/X));
16     for(int i = 0; i < N; i++){
17         b[i] = (float) rand() / ((float) (RAND_MAX/X));
18         c[i] = (float) rand() / ((float) (RAND_MAX/X));
19     }
20     for(int i = 0; i < N; i++){
21         aCPU[i] = b[i] + c[i]*d;
22     }
23     for(int i = 0; i < N; i++){
24         bType.data = b[i];
25         bType.last = i == N - 1;
26         bStream.write(bType);
27         cType.data = c[i];
28         cType.last = i == N - 1;
29         cStream.write(cType);
30     }
31     dType.data = d;
32     dType.last = 1;
33     dStream.write(dType);
34
35     kernel(aStream, bStream, cStream, dStream);
36
37     for(int i = 0; i < N; i++){
38         aType = aStream.read();
39         aFPGA[i] = aType.data;
40         if(aCPU[i] != aFPGA[i]){
41             printf("Error at index %d: %f != %f\n", i, aCPU[i], aFPGA[i]);
42             return 1;
43         }
44     }
45
46     return 0;
47
48 }
```

Writable Smart Insert 45 : 1

V7 Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.46	1.25

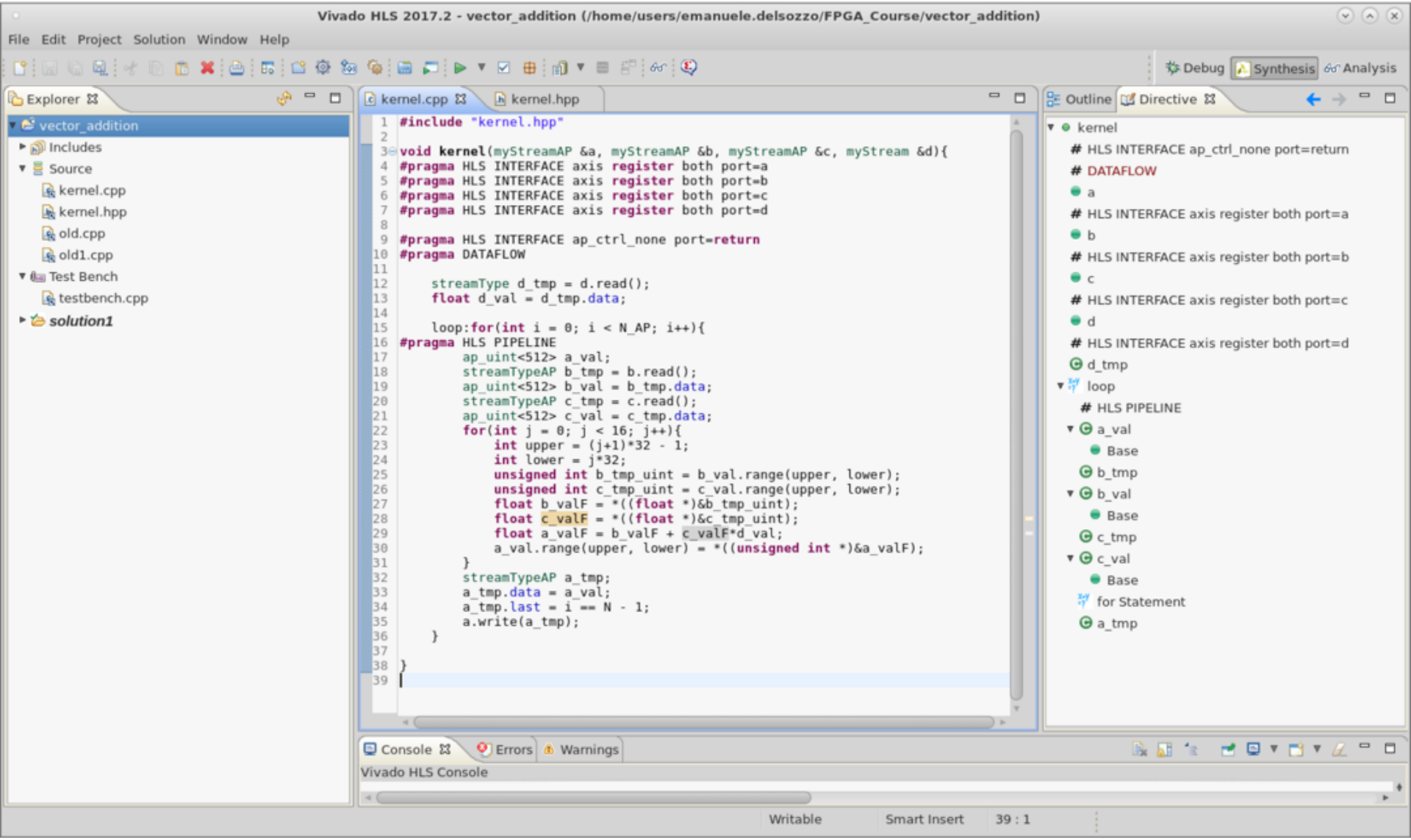
Latency		Interval		
min	max	min	max	Type
169	169	170	170	none

Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	0%	~0%	~0%	~0%

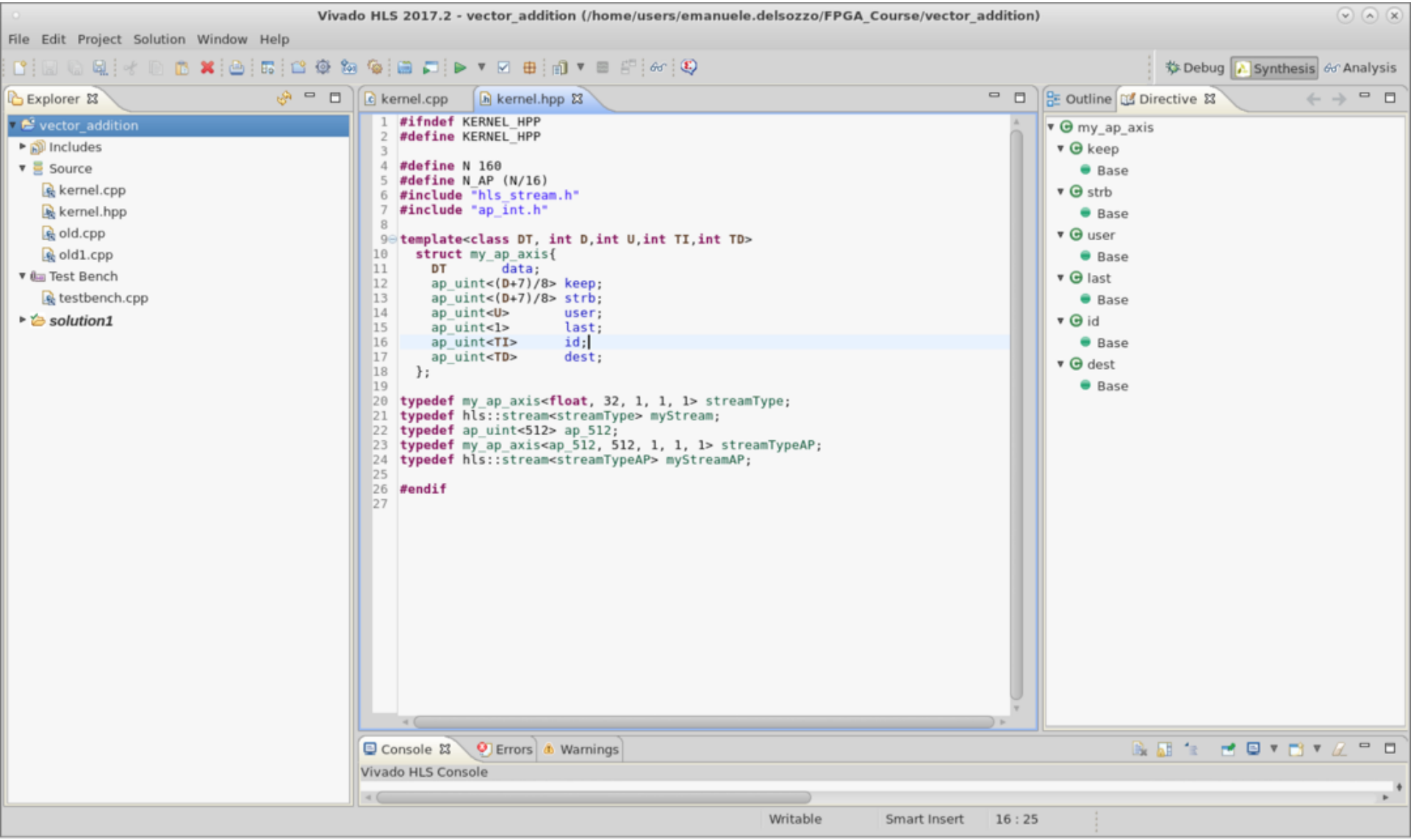
V7 Implementation Loops

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
loop	166	166	8	1	1	160	yes

V8 Implementation



V8 Streaming Interface



V8 Testbench

Vivado HLS 2017.2 - vector_addition (/home/users/emanuele.delsozzo/FPGA_Course/vector_addition)

File Edit Project Solution Window Help

Debug Synthesis Analysis

kernel.cpp kernel.hpp testbench.cpp

```
5
6  void kernel(myStreamAP &a, myStreamAP &b, myStreamAP &c, myStream &d);
7
8  int main(){
9
10     float aCPU[N], aFPGA[N], b[N], c[N], d;
11     myStreamAP aStream, bStream, cStream;
12     myStream dStream;
13     streamType dType;
14     streamTypeAP aType, bType, cType;
15
16     srand(0);
17     d = (float) rand() / ((float) (RAND_MAX/X));
18     for(int i = 0; i < N; i++){
19         b[i] = (float) rand() / ((float) (RAND_MAX/X));
20         c[i] = (float) rand() / ((float) (RAND_MAX/X));
21     }
22     for(int i = 0; i < N; i++){
23         aCPU[i] = b[i] + c[i]*d;
24     }
25     for(int i = 0; i < N_AP; i++){
26         ap_uint<512> bAP;
27         ap_uint<512> cAP;
28         for(int j = 0; j < 16; j++){
29             int upper = (j+1)*32 - 1;
30             int lower = j*32;
31             float bVal = b[i*16 + j];
32             float cVal = c[i*16 + j];
33             bAP.range(upper, lower) = *((unsigned int *)&bVal);
34             cAP.range(upper, lower) = *((unsigned int *)&cVal);
35         }
36         bType.data = bAP;
37         bType.last = i == N_AP - 1;
38         bStream.write(bType);
39         cType.data = cAP;
40         cType.last = i == N_AP - 1;
41         cStream.write(cType);
42     }
43     dType.data = d;
44     dType.last = 1;
45     dStream.write(dType);
46
47     kernel(aStream, bStream, cStream, dStream);
48
49     for(int i = 0; i < N_AP; i++){
50         aType = aStream.read();
```

Writable Smart Insert 38 : 30

V8 Implementation Report

Clock	Target	Estimated	Uncertainty
ap_clk	10.00	8.46	1.25

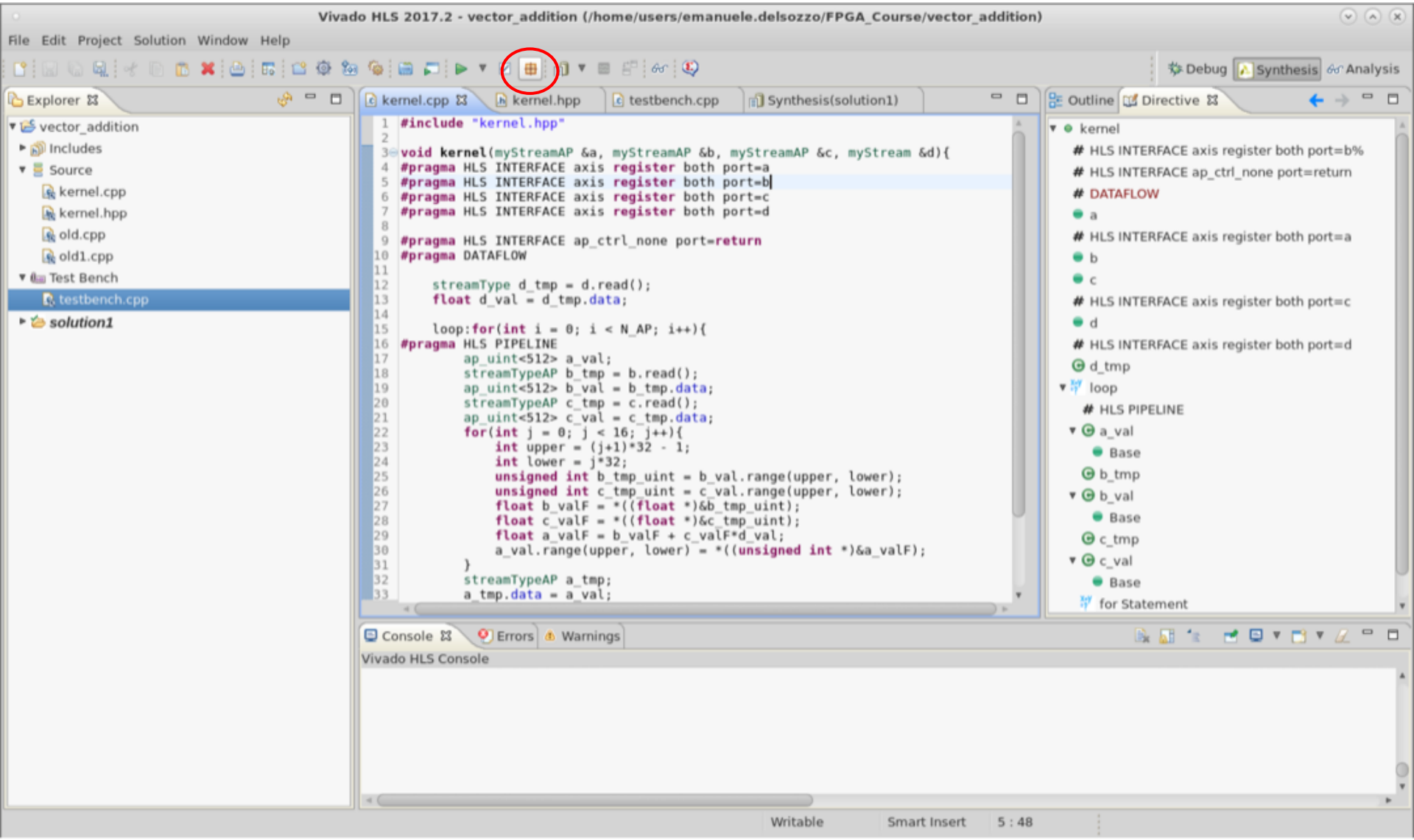
Latency		Interval		
min	max	min	max	Type
19	19	20	20	none

Name	BRAM_18K	DSP48E	FF	LUT
Available	2060	2800	2393	2618
Utilization	0%	2%	1%	1%

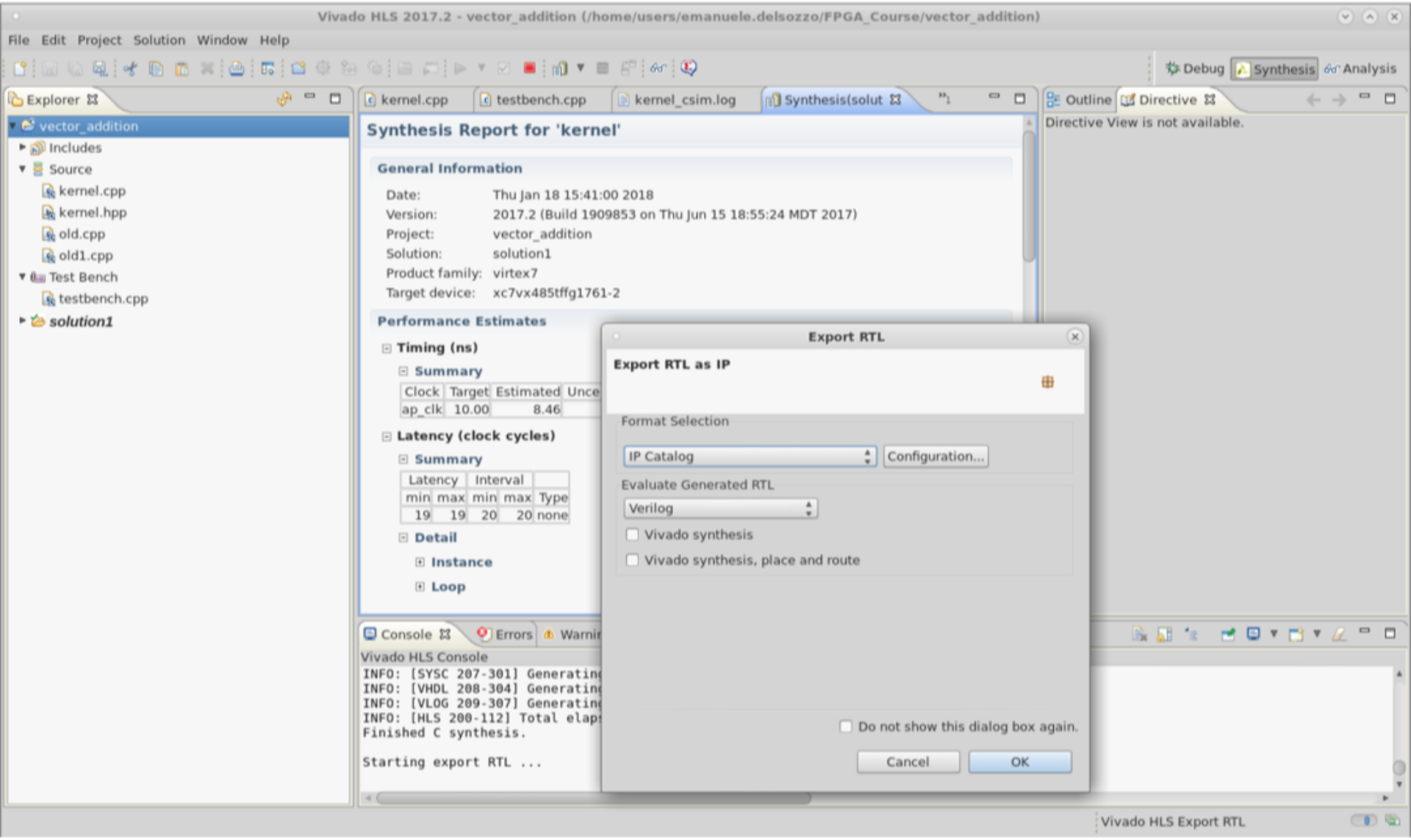
V8 Implementation Loops

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
loop	16	16	8	1	1	10	yes

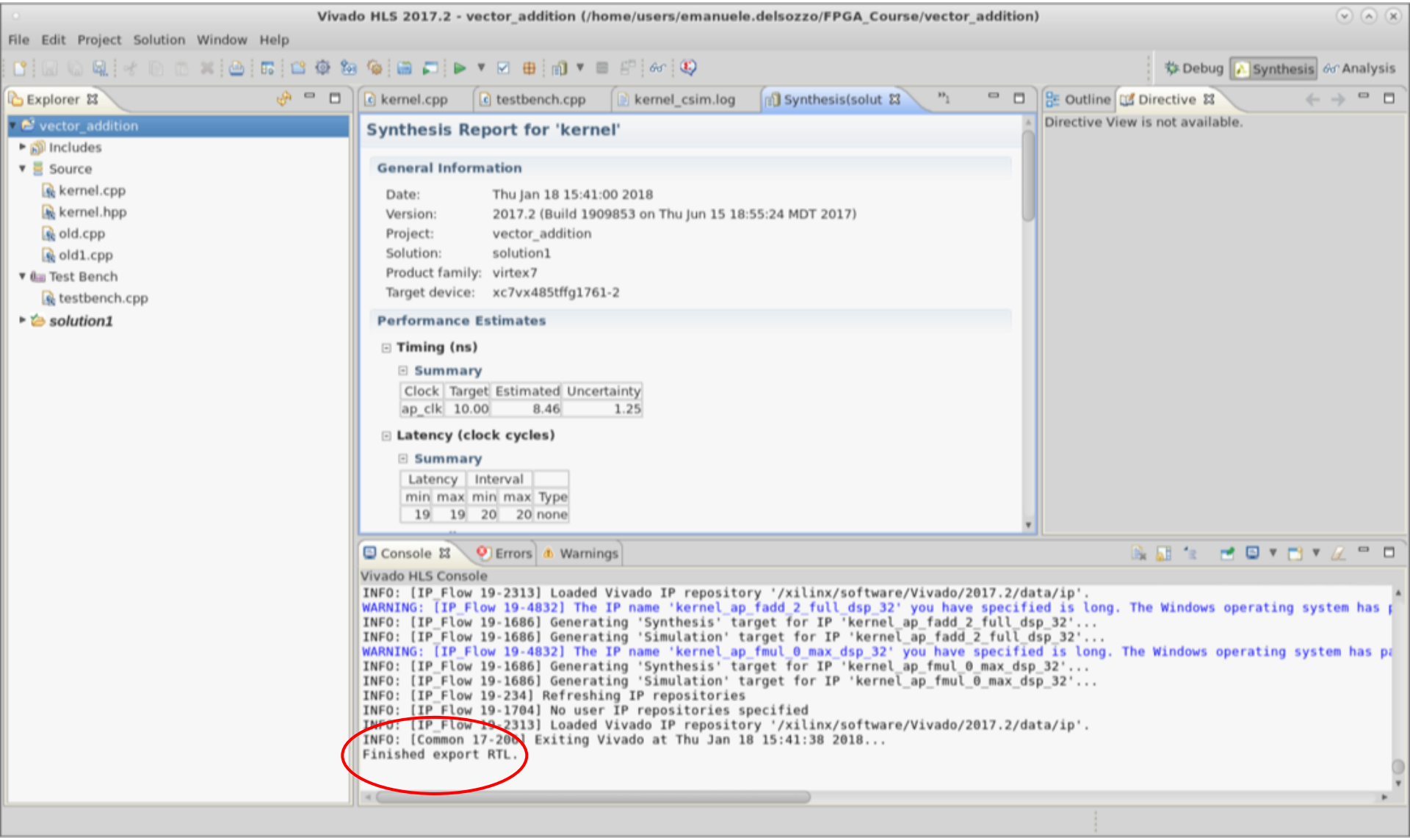
Export IP


Vivado HLS 2017.2 - vector_addition (/home/users/emanuele.delsozzo/FPGA_Course/vector_addition)
File Edit Project Solution Window Help
Debug Synthesis Analysis
Explorer
vector_addition
Includes
Source
kernel.cpp
kernel.hpp
old.cpp
old1.cpp
Test Bench
testbench.cpp
solution1
kernel.cpp kernel.hpp testbench.cpp Synthesis(solution1)
#include "kernel.hpp"
void kernel(myStreamAP &a, myStreamAP &b, myStreamAP &c, myStream &d){
#pragma HLS INTERFACE axis register both port=a
#pragma HLS INTERFACE axis register both port=b
#pragma HLS INTERFACE axis register both port=c
#pragma HLS INTERFACE axis register both port=d
#pragma HLS INTERFACE ap_ctrl_none port=return
#pragma DATAFLOW
streamType d_tmp = d.read();
float d_val = d_tmp.data;
loop:for(int i = 0; i < N_AP; i++){
#pragma HLS PIPELINE
ap_uint<512> a_val;
streamTypeAP b_tmp = b.read();
ap_uint<512> b_val = b_tmp.data;
streamTypeAP c_tmp = c.read();
ap_uint<512> c_val = c_tmp.data;
for(int j = 0; j < 16; j++){
int upper = (j+1)*32 - 1;
int lower = j*32;
unsigned int b_tmp_uint = b_val.range(upper, lower);
unsigned int c_tmp_uint = c_val.range(upper, lower);
float b_valF = *((float *)&b_tmp_uint);
float c_valF = *((float *)&c_tmp_uint);
float a_valF = b_valF + c_valF*d_val;
a_val.range(upper, lower) = *((unsigned int *)&a_valF);
}
streamTypeAP a_tmp;
a_tmp.data = a_val;
Outline Directive
kernel
HLS INTERFACE axis register both port=b%
HLS INTERFACE ap_ctrl_none port=return
DATAFLOW
a
HLS INTERFACE axis register both port=a
b
c
HLS INTERFACE axis register both port=c
d
HLS INTERFACE axis register both port=d
d_tmp
loop
HLS PIPELINE
a_val
Base
b_tmp
b_val
Base
c_tmp
c_val
Base
for Statement
Console Errors Warnings
Vivado HLS Console
Writable Smart Insert 5 : 48

Export IP Settings



Export IP Done



This is only the beginning!!

For more information, read Vivado HLS manual

https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_2/ug902-vivado-high-level-synthesis.pdf

Feedbacks

- We are working at improving this course, would you share your feedback for this lesson?

https://goo.gl/tLcWQj

Thank You for the Attention!

Lorenzo Di Tucci
lorenzo.ditucci@polimi.it
Emanuele Del Sozzo
emanuele.delsozzo@polimi.it
Marco D. Santambrogio
marco.santambrogio@polimi.it