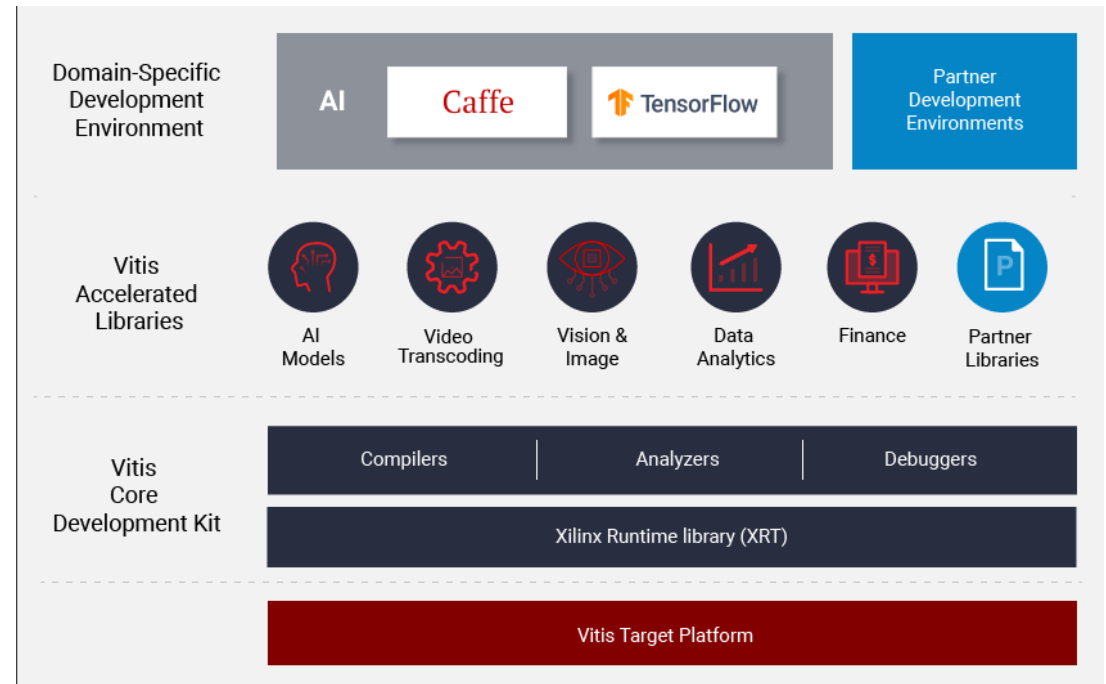


# Getting to grips with Vitis

Adam Taylor

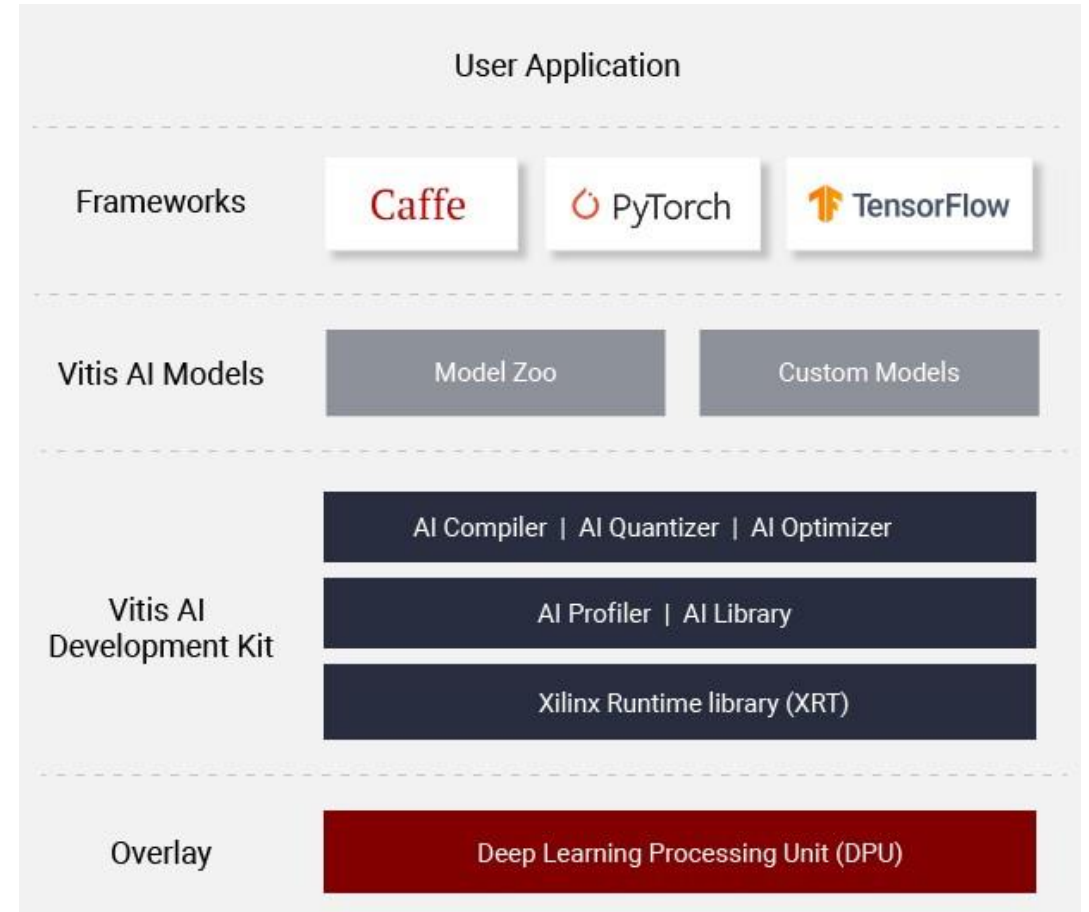
# What is Vitis

- Vitis is unified software development environment from Xilinx
- Offers Unified edge and cloud development methodologies
- Support embedded and accelerated flows



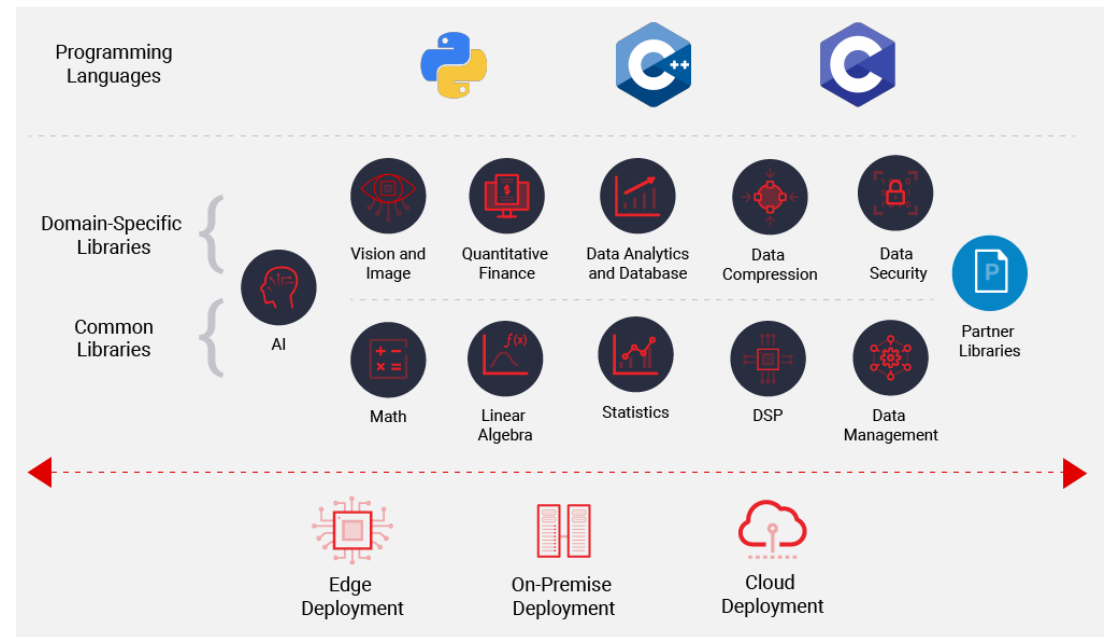
# Vitis AI Development Environment

- Vitis AI enables acceleration of AI inference at edge & Cloud
- Supports leading frameworks such as TensorFlow, Caffe and Pytorch
- Works with fixed point representation and Xilinx Deep Learning Processor Unit



# Vitis Accelerated Libraries

- Several Open Source acceleration ready libraries
- Common Libraries – offer a set of common functionality
- Domain specific libraries – offer out of the box functions for specific domains e.g. vision



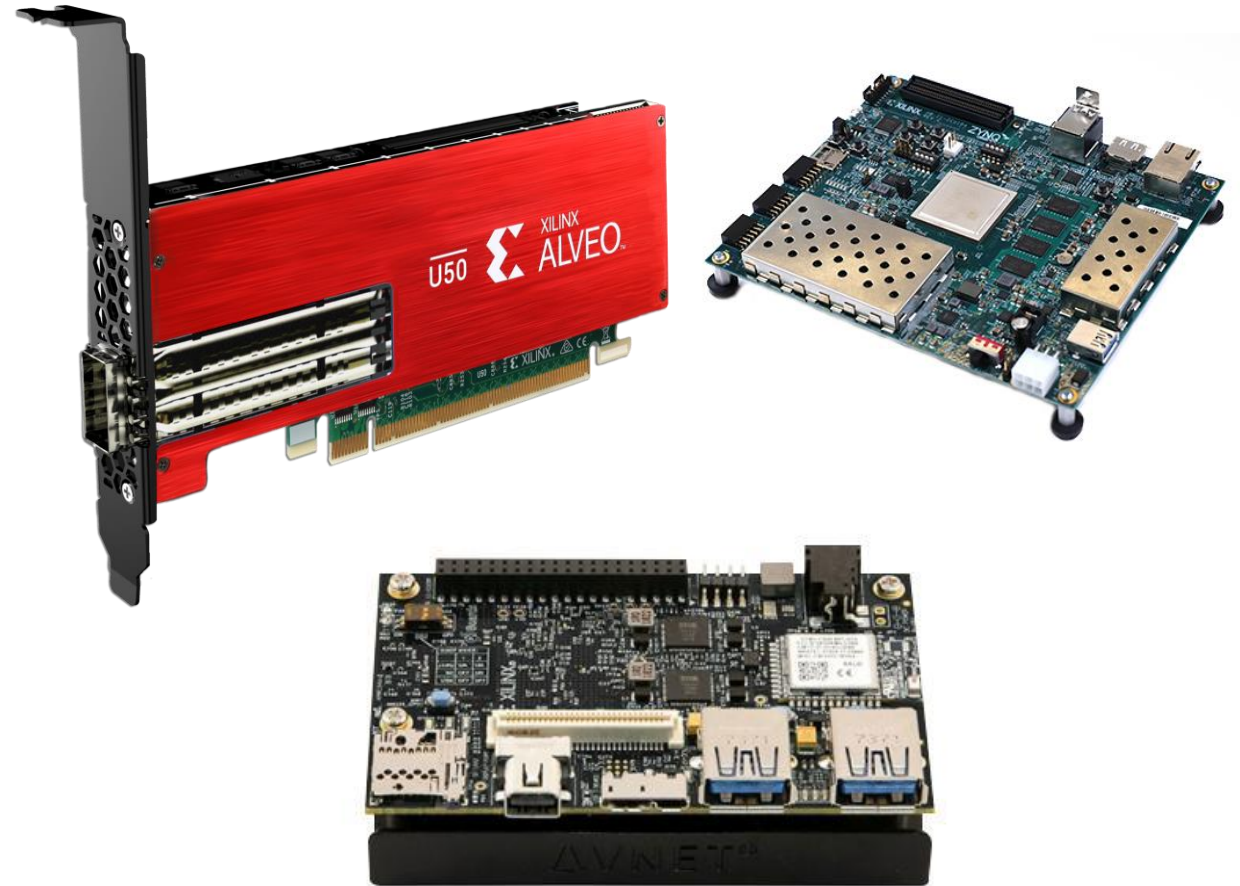
# Vitis Core Development Kit

- GUI & Command line tools for compilation, debug and analysis of C, C++ and OpenCL designs.
- Can use preferred GUI or integrated GUI
- Supports embedded and accelerated flows



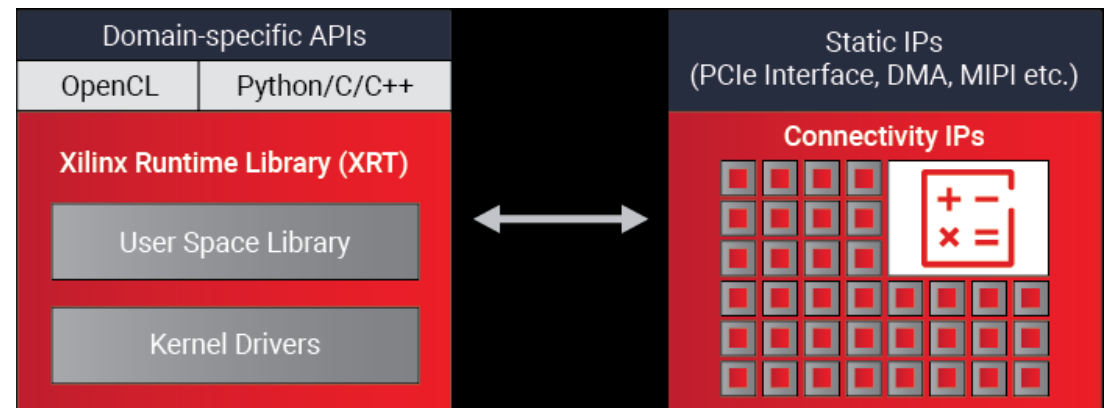
# Vitis Target Platforms

- Embedded
  - SoC – MPSoC, RFSoc, Zynq
  - FPGA – MicroBlaze
- Cloud – Alveo / AWS F1 Instance
- Embedded SoC and Cloud applications can use acceleration flow.
- All required files and boot elements are generated



# Xilinx Runtime library

- Xilinx Runtime library (XRT) enables communication between the host and accelerator
- Cloud based – Host x86
- Embedded – Arm A9 or A53
- Provides all libraries, APIs, drivers and utilities.





# Xilinx Runtime library

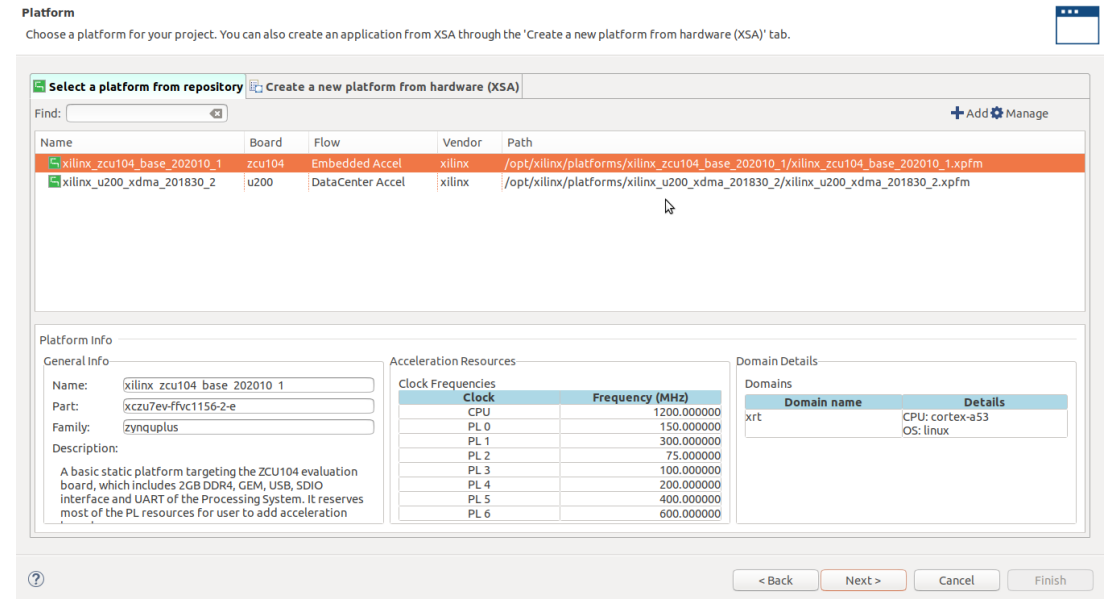
- Key Functions of the Runtime include
- Downloading the FPGA binary
- Memory Management between Host and Accelerator
- Execution Management
- Board Management

```
adiuvo@Adiuvo: ~  
File Edit View Search Terminal Help  
INFO: == Starting PCIE link check:  
LINK ACTIVE, ATTENTION  
Ensure Card is plugged in to Gen3x16, instead of Gen3x4  
Lower performance may be experienced  
WARN: == PCIE link check PASSED with warning  
INFO: == Starting SC firmware version check:  
SC FIRMWARE MISMATCH, ATTENTION  
SC firmware running on board: 1.8. Expected SC firmware from installed Shell: 4.2.0  
Please use "xbmgmt flash --scan" to check installed Shell.  
WARN: == SC firmware version check PASSED with warning  
INFO: == Starting verify kernel test:  
INFO: == verify kernel test PASSED  
INFO: == Starting DMA test:  
Host -> PCIE -> FPGA write bandwidth = 3335.9 MB/s  
Host <- PCIE <- FPGA read bandwidth = 3238.05 MB/s  
INFO: == DMA test PASSED  
INFO: == Starting device memory bandwidth test:  
.....  
Maximum throughput: 52428 MB/s  
INFO: == device memory bandwidth test PASSED  
INFO: == Starting PCIE peer-to-peer test:  
P2P BAR is not enabled. Skipping validation  
INFO: == PCIE peer-to-peer test SKIPPED  
INFO: == Starting memory-to-memory DMA test:  
bank0 -> bank1 M2M bandwidth: 12100 MB/s  
bank0 -> bank2 M2M bandwidth: 12128.7 MB/s  
bank0 -> bank3 M2M bandwidth: 12114.9 MB/s  
bank1 -> bank2 M2M bandwidth: 12116 MB/s  
bank1 -> bank3 M2M bandwidth: 12118.9 MB/s  
bank2 -> bank3 M2M bandwidth: 12116 MB/s  
INFO: == memory-to-memory DMA test PASSED  
INFO: Card[0] validated with warnings.  
  
INFO: All cards validated successfully but with warnings.  
adiuvo@Adiuvo:~$
```



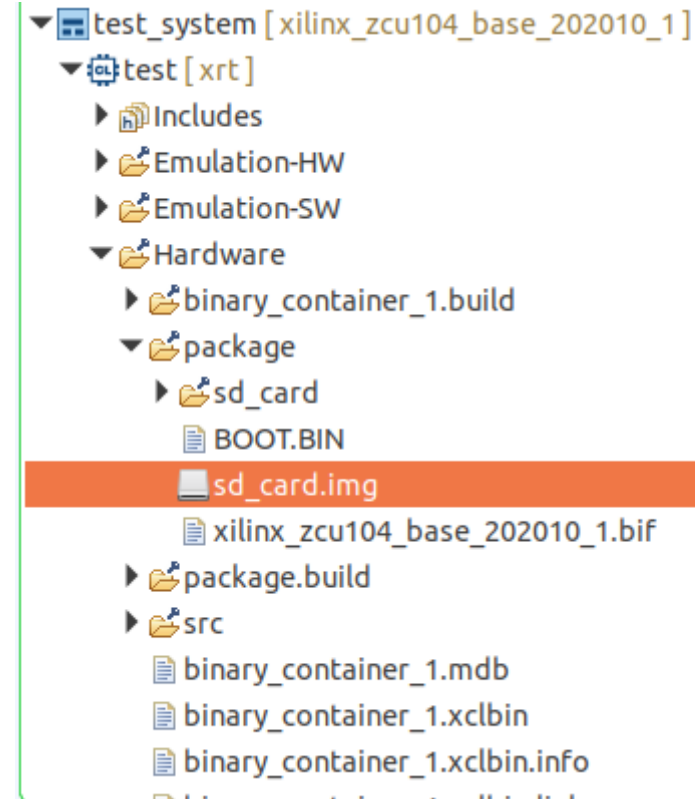
# Element of Vitis

- All projects required a platform
  - Hardware element – makes available AXI connections, clocks and Interrupts in the PL to Vitis Compiler
  - Software element – provides boot, XRT and QEMU support
  - Linux element – FS, Image and SysRoot



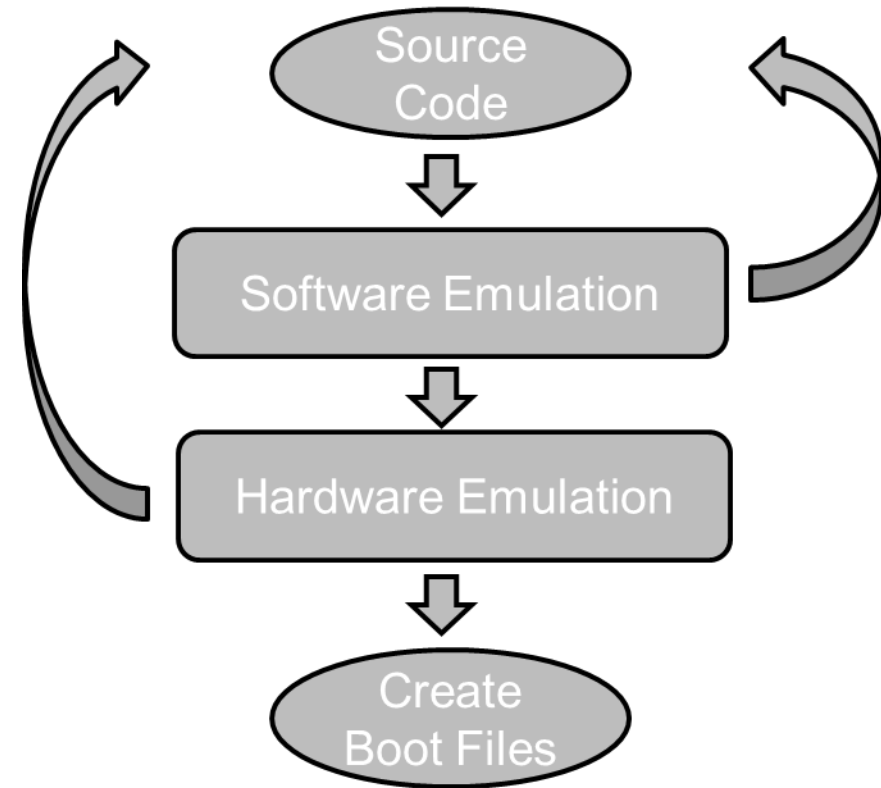
# Vitis Output

- Compiled binary (host) and XCLbin (accelerator)
- Embedded System Output
  - SD Card Image
    - Image
    - File System
    - Binary and XCLBin
- Cloud output
  - Binary and XCLBin



# Vitis Development Flow

- Software Emulation – Syntax errors & algorithm verification
- Hardware Emulation – Optimize Performance, Interfacing & Resources



# OpenCL Framework

An open industry standard

- For parallel computing
- Of heterogeneous systems

Enables cross-platform functional portability

- No code changes
- Portable across CPU, GPU, FPGA, DSP, etc.
  - Can run on cell phones, laptops, super computers
- Important: No performance portability

Wide market adoption

- Support implemented by
  - Apple, AMD, Xilinx, Intel, ARM, Nvidia, Qualcomm, etc.
- Many companies developing applications
  - Image, video, audio processing, scientific calculations, medical imaging, and more

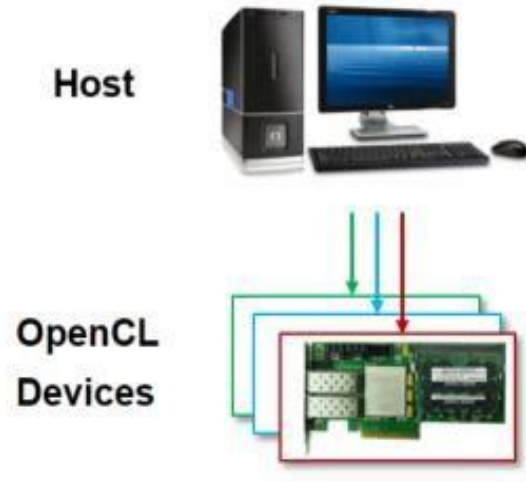


OpenCL

*Khronos Group*

[www.khronos.org](http://www.khronos.org)

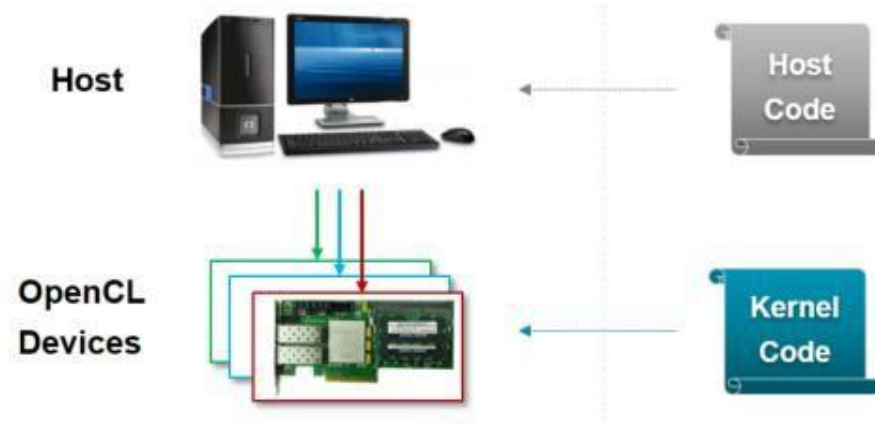
# OpenCL Framework



## Platform model

- Defines representation of ANY platform
- Contains
  - Single host
  - One or more OpenCL devices (compute device)

# OpenCL Framework



## Platform model

- Defines representation of ANY platform
- Contains
  - Single host
  - One or more OpenCL devices (compute device)

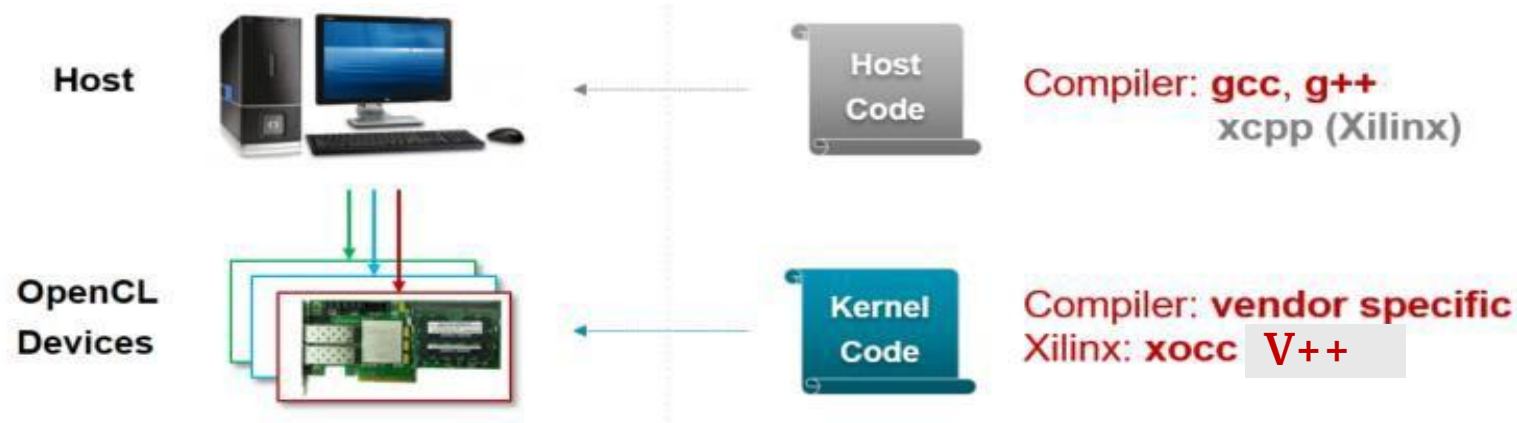
## Execution model

OpenCL application: Two parts

- Host program
  - Manages the entire application: OpenCL APIs
- Kernels (OpenCL C language)
  - Functions to accelerate, run on OpenCL devices



# OpenCL Framework



## Platform model

- Defines representation of ANY platform
- Contains
  - Single host
  - One or more OpenCL devices (compute device)

## Execution model

OpenCL application: Two parts

- Host program
  - Manages the entire application: OpenCL APIs
- Kernels (OpenCL C language)
  - Functions to accelerate: run on OpenCL devices

# Execution Model – Command Queues

Interaction between host and device occurs via command queues

- Created by host
- Attached to a single device
  - **Note:** Multiple command queues can be active within context

Three command types

- Kernel execution commands
- Memory commands
  - Transfer data between host and different memory objects
- Synchronization commands
  - Put constraints on in the order in which commands are executed



# Memory Model

## Three types of memory objects

### – Buffer objects

- Contiguous block of memory
- Available to kernels for read/write
- Programmer can write data to buffers
- Access to data via pointers

### – Image objects (not a part of embedded profile)

- Hold images only
- Storage/format can be optimized for specific OpenCL device
- OpenCL framework provides functions to manipulate images

### – Pipes

- Data organized as FIFO
- Accessed (read/write) via built in
- Pipe not accessible from the host



# Five Sub-regions of Memory Objects

## Host memory

- Visible to host only
- OpenCL framework only defines how host memory interacts with OpenCL objects

## Global memory

- Visible to host and device
- All work items in all workgroups can read/write there
- Global on-chip memory – visible to device only

## Constant memory

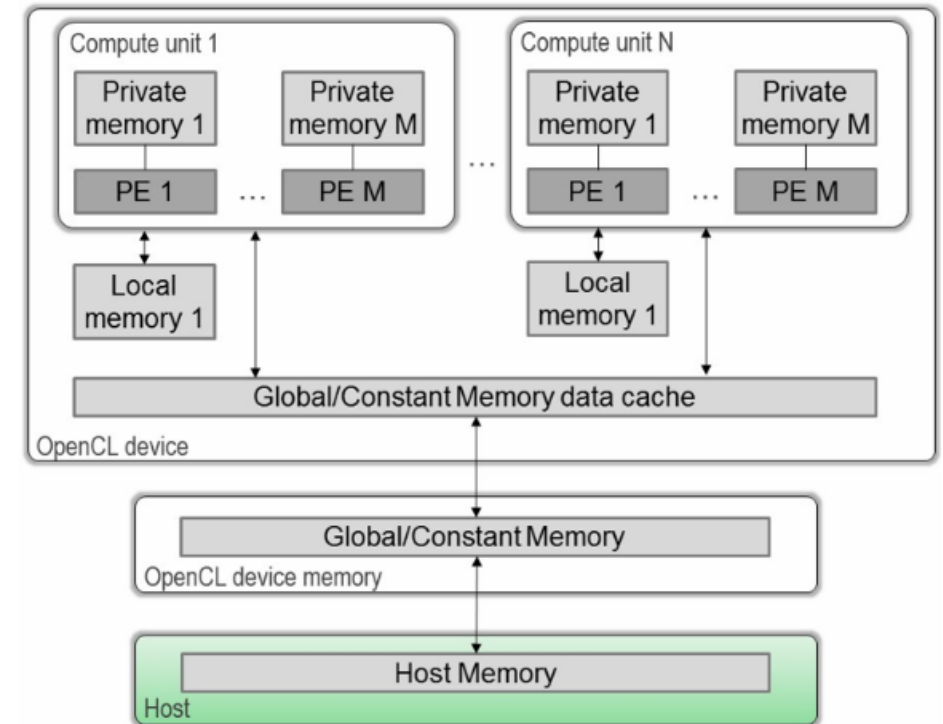
- Region of global memory
- Work items – read access only

## Local memory

- Local to workgroup (shared by all work-items in a group)

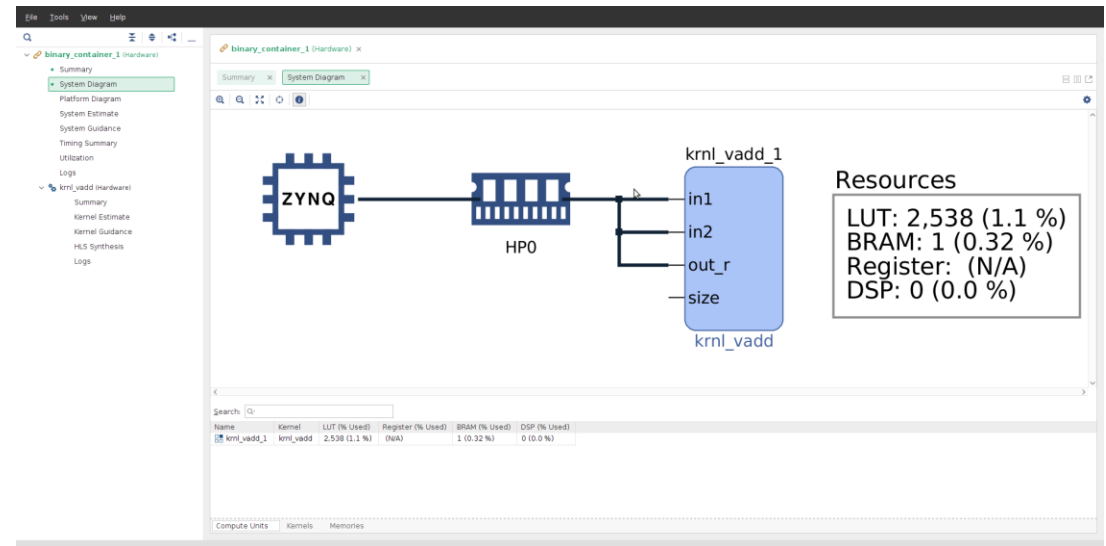
## Private memory

- Accessible by a work-item



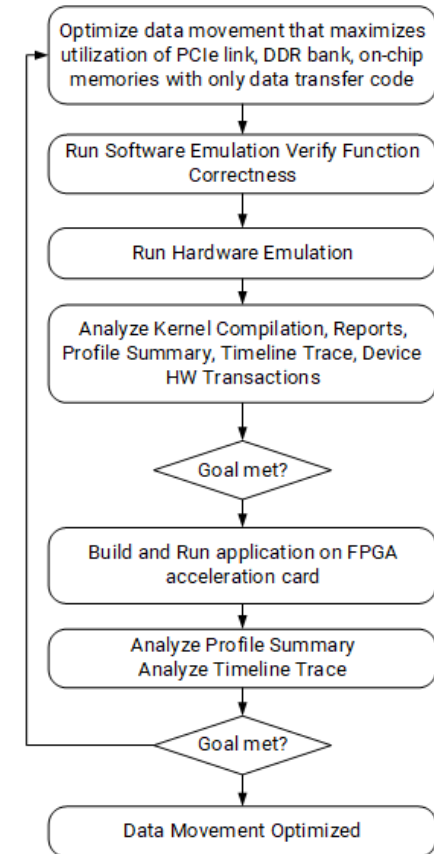
# Optimization

- Optimization Possible at both Host and Kernel
- Enables most responsive solution
- Host optimization
- Kernel optimization possible in OpenCL and C/C++
  - Optimization Syntax differs



# Host Optimization

- Optimize the data movement in the application before optimizing computation
- Compute Unit Scheduling
  - Multiple In-Order Command Queues
  - Single Out-of-Order Command Queue

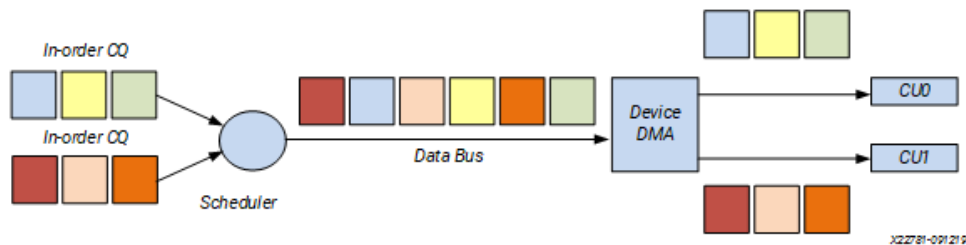


X2239-082719

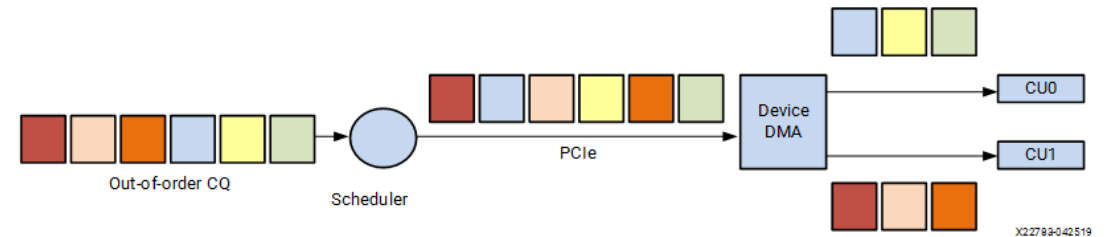


# Host Optimization

- Multiple In-Order Command Queues

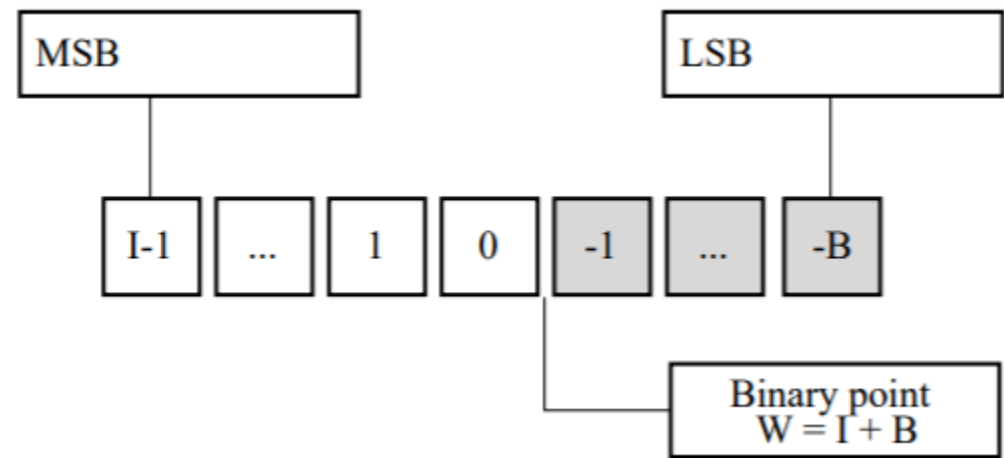


- Single Out-of-Order Command Queue



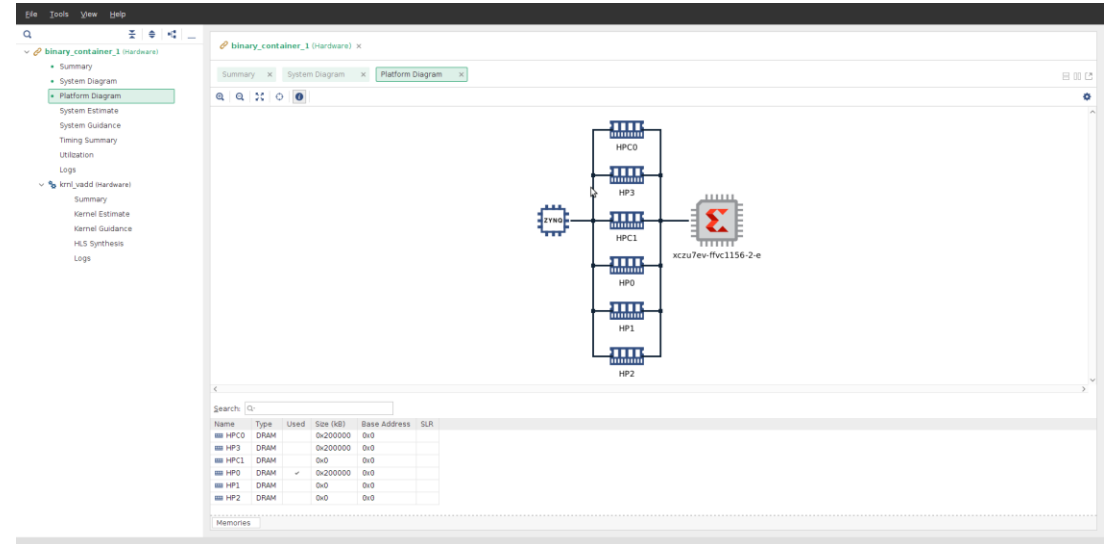
# Kernel Optimization – Data Types

- Avoid native C data types  
e.g. int, float, double
- Best performance is using  
bit accurate types (C/C++  
Kernels)
  - Arbitrary Precision Integer
  - Arbitrary Precision fixed  
point
- Enables smaller & faster  
logic implementations



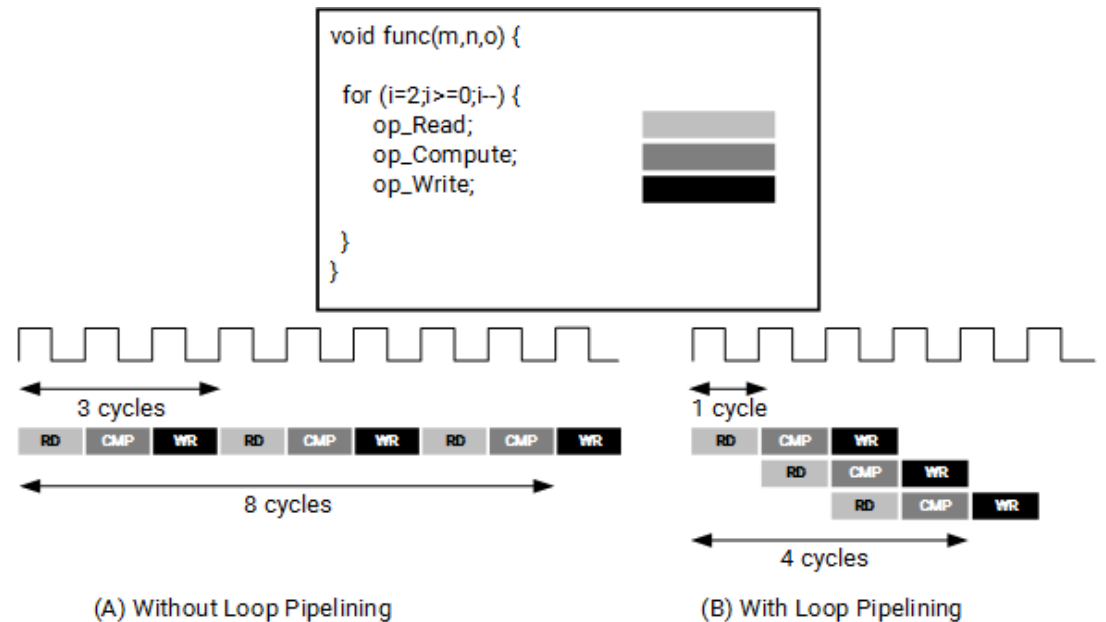
## A decorative graphic consisting of several vertical and diagonal blue lines of varying thicknesses, interspersed with solid blue circles and hollow blue circles, resembling a stylized circuit board or data flow diagram.

- Two types of data transfer
  - Data Pointers via global memory (M\_AXI)
  - Scalar direct to kernel (AXI\_LITE)
- Vitis automatically selects interface type
- Max data width is 512 bits – maximum performance leverages this



# Kernel Optimization – Pipelining

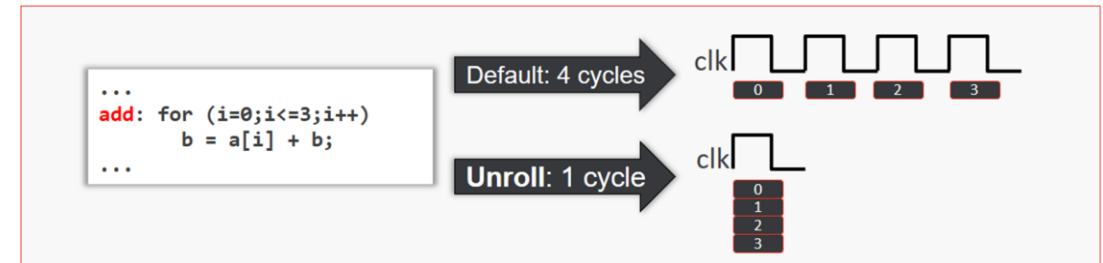
- By default, every iteration of a loop only starts when the previous iteration has finished
- Pipelining the loop executes subsequent iterations in a pipelined manner



X14277-110217

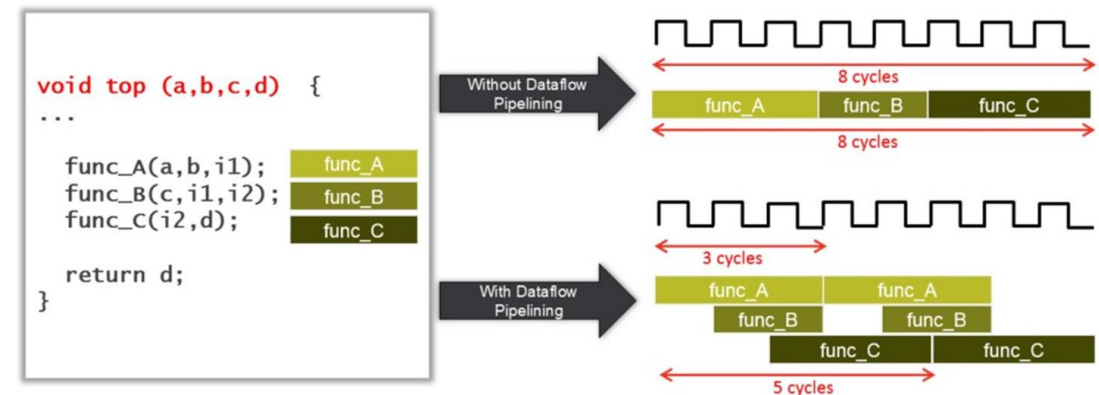
# Kernel Optimization – Unrolling

- Unrolling a loop enables the full parallelism
- Full or Partial Unroll
- Data dependencies in loops can impact the results of loop pipelining or unrolling



# Kernel Optimization – DataFlow

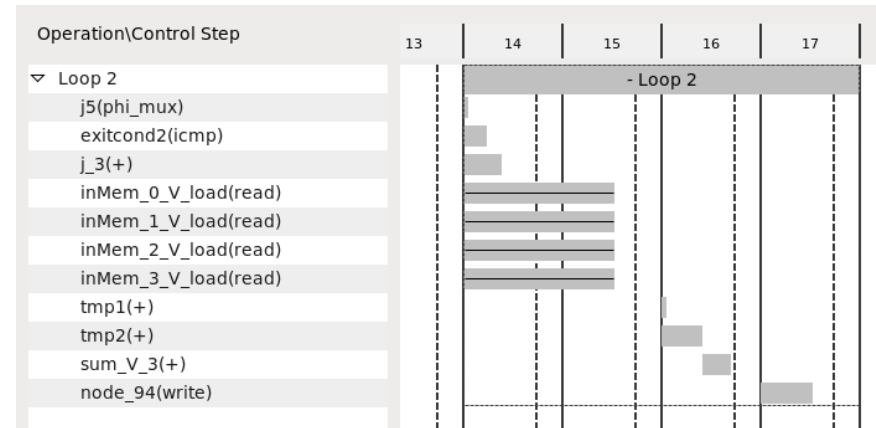
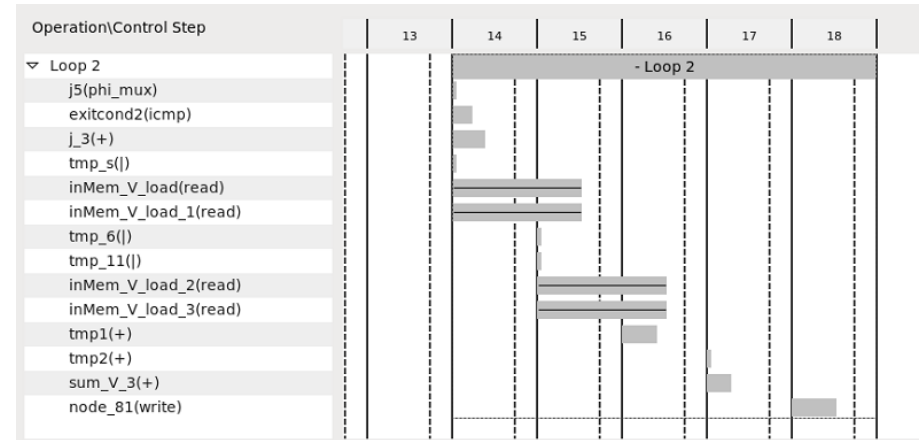
- Improve kernel performance by enabling task-level pipelining
- Be careful of
- Single producer-consumer violations.
  - Bypassing tasks.
  - Feedback between tasks.
  - Conditional execution of tasks.
  - Loops with multiple exit conditions or conditions defined within the loop





# Kernel Optimization – Memory

- Limited BRAM access bandwidth, can heavily impact the overall performance
- Ability to partition and reshape arrays can increase bandwidth
- Partition – Separates into different BRAMS
- Reshape – allows combination of words



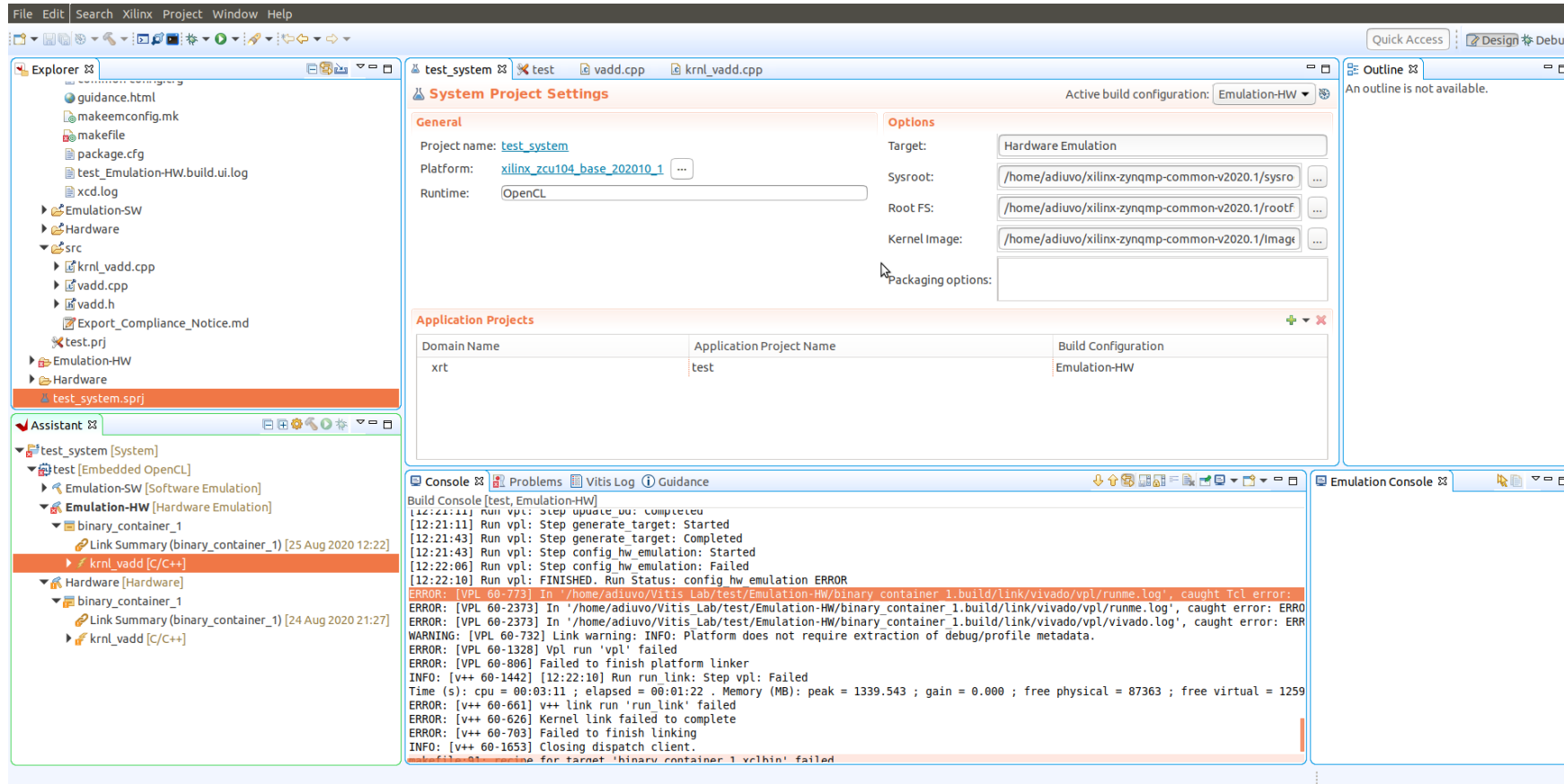
# Kernel Optimization - Pragmas

Optimization	C/C++	OpenCL
Pipeline	#pragma HLS PIPELINE	__attribute__((xcl_pipeline_loop))
Unroll	#pragma HLS UNROLL	__attribute__((opencl_unroll_hint))
DataFlow	#pragma HLS DATAFLOW	__attribute__((xcl_dataflow))
Memory	#pragma HLS ARRAY_PARTITION	

Further information can be found at

[https://www.xilinx.com/html\\_docs/xilinx2020\\_1/vitis\\_doc/optimizingperformance.html#fhe1553474153030](https://www.xilinx.com/html_docs/xilinx2020_1/vitis_doc/optimizingperformance.html#fhe1553474153030)

# Vitis GUI – Project Settings



# Vitis GUI – Project Setting

The screenshot displays the Vitis GUI interface for configuring a project named 'test'. The interface is divided into several panes:

- Explorer:** Shows the project file structure, including 'test.prj', 'Emulation-HW', 'Hardware', and 'test\_system.sprj'.
- Assistant:** Shows the project hierarchy, including 'test [System]', 'test [Embedded OpenCL]', 'Emulation-SW [Software Emulation]', 'Emulation-HW [Hardware Emulation]', 'binary\_container\_1', and 'krnl\_vadd [C/C++]'.
- Application Project Settings:** Contains the main configuration options for the project.
- Console:** Displays the build log output, showing various errors and warnings.
- Emulation Console:** Shows the emulation-specific log output.

**Application Project Settings Details:**

- General:**
  - Project name: test
  - Platform: xilinx\_zcu104\_base\_202010\_1
  - Runtime: OpenCL
  - Number of devices: 1
- Options:**
  - Target: Hardware Emulation
  - Host debug: ☒
  - Kernel debug: ☒
  - Kernel debug mode: Waveform
  - Report level: Default
  - Hardware optimization: Default optimization (-O0)
- Hardware Functions:**

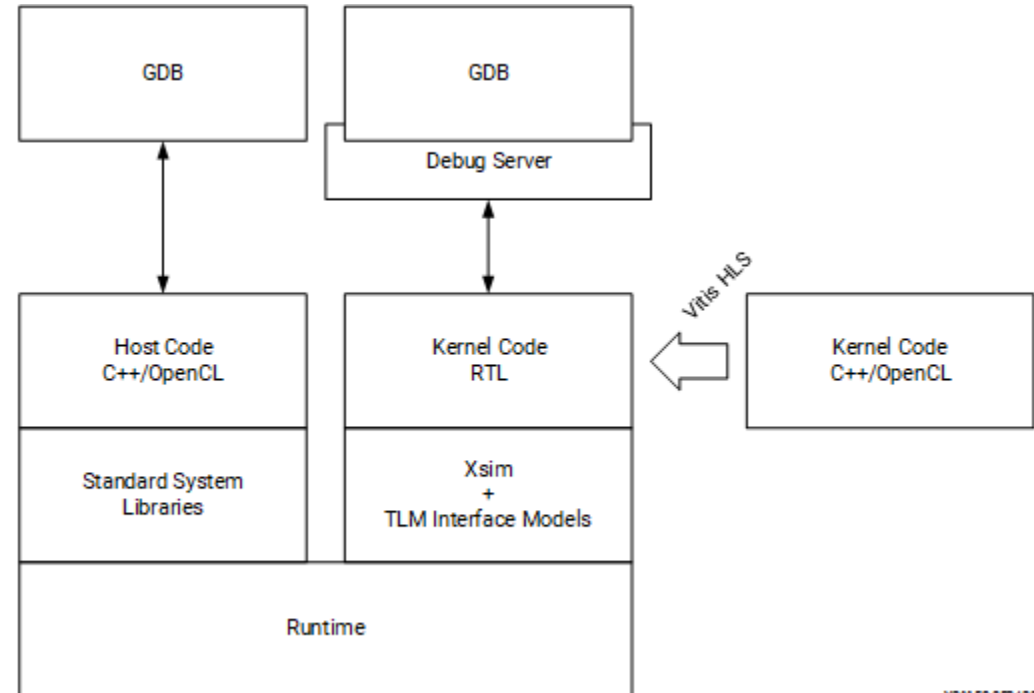
Name	Compute Units	Port Data Width	Max Memory Ports
krnl_vadd	1	Auto	<input type="checkbox"/>

**Console Log Output:**

```
Build Console [test, Emulation-HW]
[12:21:11] Run vpl: Step update_vpl: completed
[12:21:11] Run vpl: Step generate_target: Started
[12:21:43] Run vpl: Step generate_target: Completed
[12:21:43] Run vpl: Step config_hw_emulation: Started
[12:22:06] Run vpl: Step config_hw_emulation: Failed
[12:22:10] Run vpl: FINISHED. Run Status: config hw emulation ERROR
ERROR: [VPL 60-773] In '/home/adiuvo/Vitis_Lab/test/Emulation-HW/binary_container_1.build/link/vivado/vpl/runme.log', caught Tcl error:
ERROR: [VPL 60-2373] In '/home/adiuvo/Vitis_Lab/test/Emulation-HW/binary_container_1.build/link/vivado/vpl/runme.log', caught error: ERR
ERROR: [VPL 60-2373] In '/home/adiuvo/Vitis_Lab/test/Emulation-HW/binary_container_1.build/link/vivado/vpl/vivado.log', caught error: ERR
WARNING: [VPL 60-732] Link warning: INFO: Platform does not require extraction of debug/profile metadata.
ERROR: [VPL 60-1328] Vpl run 'vpl' failed
ERROR: [VPL 60-806] Failed to finish platform linker
INFO: [v++ 60-1442] [12:22:10] Run run_link: Step vpl: Failed
Time (s): cpu = 00:03:11 ; elapsed = 00:01:22 . Memory (MB): peak = 1339.543 ; gain = 0.000 ; free physical = 87363 ; free virtual = 1259
ERROR: [v++ 60-661] v++ link run 'run link' failed
ERROR: [v++ 60-626] Kernel link failed to complete
ERROR: [v++ 60-703] Failed to finish linking
INFO: [v++ 60-1653] Closing dispatch client.
makefile:91: recipe for target 'binary_container_1.xclbin' failed
```

# Vitis-Debug

- Can Debug
  - Software Emulation
  - Hardware Emulation
- Hardware flow insert ILA
- Debugging will use QEMU and Logic Simulator



XZ1159052420



# Hands On Labs





# ADIUVO

ENGINEERING AND TRAINING, LTD.

High Reliability Design | Embedded Systems Expertise | Quality Training

[www.adiuvoengineering.com](http://www.adiuvoengineering.com)