

For Software Engineer, Accelerating your algorithm with Altera SoC FPGA

using OpenCL, OpenCV



■ Introduce to Altera SoC FPGA

- What is an FPGA?
- Advantages of Altera SoC FPGA

■ Running an OpenCL, OpenCV application on SoC FPGA

- OpenCL introduction
- Altera OpenCL SDK overview
- Ray tracing demo
- IpTronix OpenCV introduction

Introduce to FPGA and SoC FPGA

Your User-Customizable System on Chip



A Complete Solutions Portfolio

ALTERA®

POWERING YOUR INNOVATION

MAX®
Series

Cyclone®
Series

Arria®
Series

Stratix®
Series

ENPIRION

CPLDs

**Lowest Cost,
Lowest Power**

FPGAs

**Cost/Power Balance
SoC & Transceivers**

FPGAs

**Mid-range FPGAs
SoC & Transceivers**

FPGAs

**Optimized for
High Bandwidth**

PowerSoCs

**High-efficiency
Power Management**

RESOURCES

**Embedded Soft and
Hard Processors**

Nios® II
ARM®

**Design
Software**



**Development
Kits**



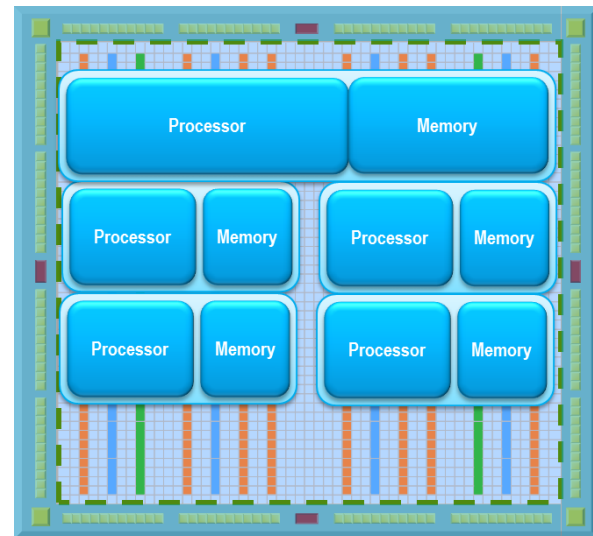
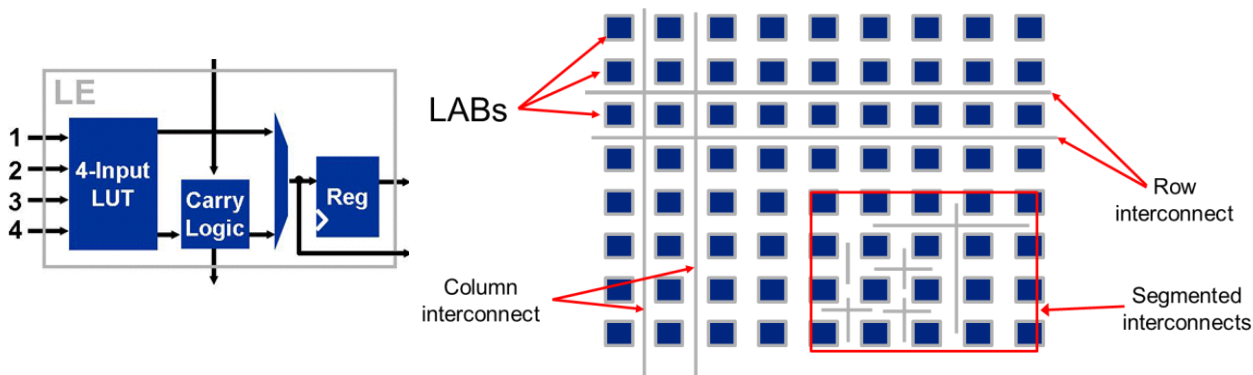
**Intellectual
Property (IP)**

- Industrial
- Computing
- Enterprise



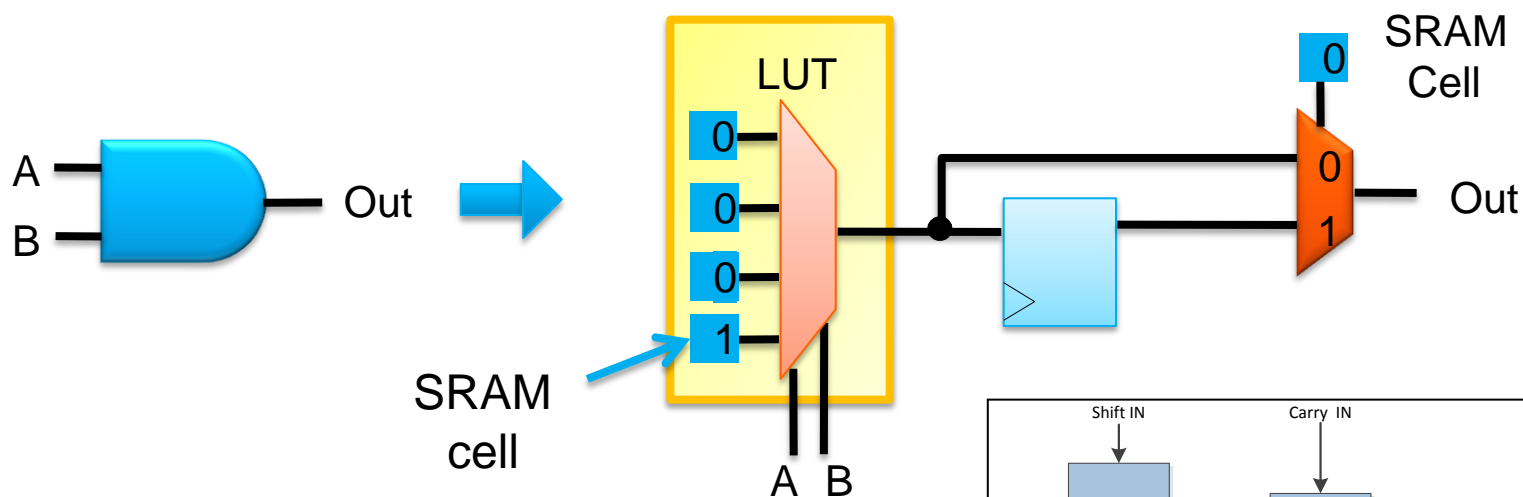
What is an FPGA?

- **Field Programmable Gate Array (FPGA)**
- **LABs (Logic Array Blocks) arranged in an array**
- **Programmable interconnect creates custom functionality regions in the array**
 - Arbitrary complexity control instructions
 - Arbitrary width data instructions
 - Arbitrary size memory
- **Processor units customized to your application**



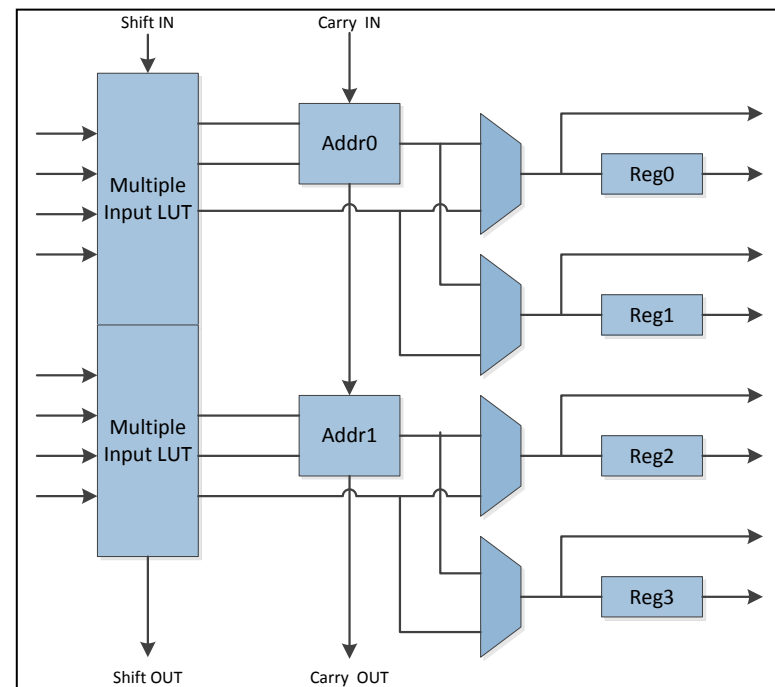
FPGA Logic Elements

■ FPGAs use multiple input LUTs



■ Multiple Families

- Cyclone: cost and power balanced
- Arria: midrange
- Stratix: high bandwidth



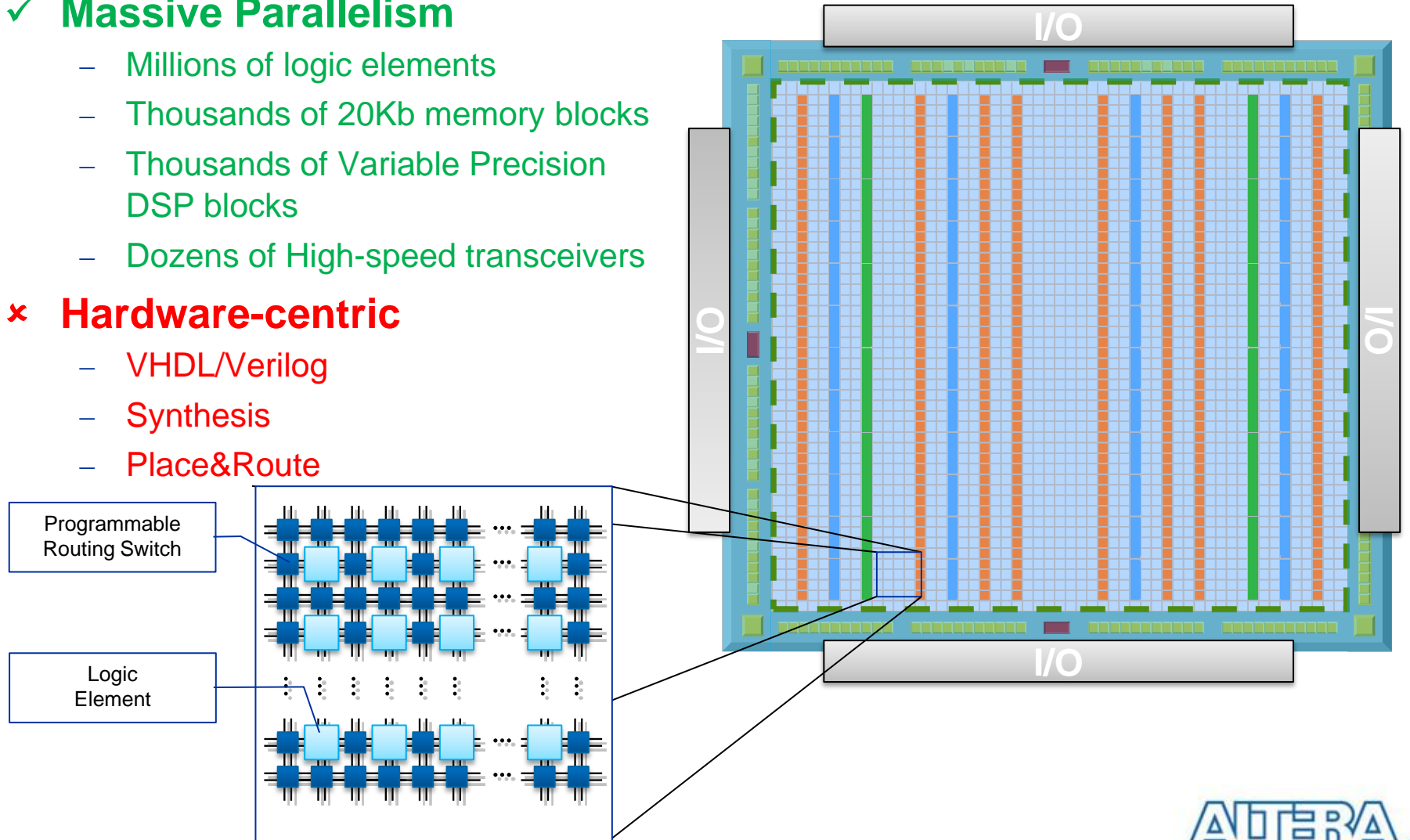
FPGA Architecture

✓ Massive Parallelism

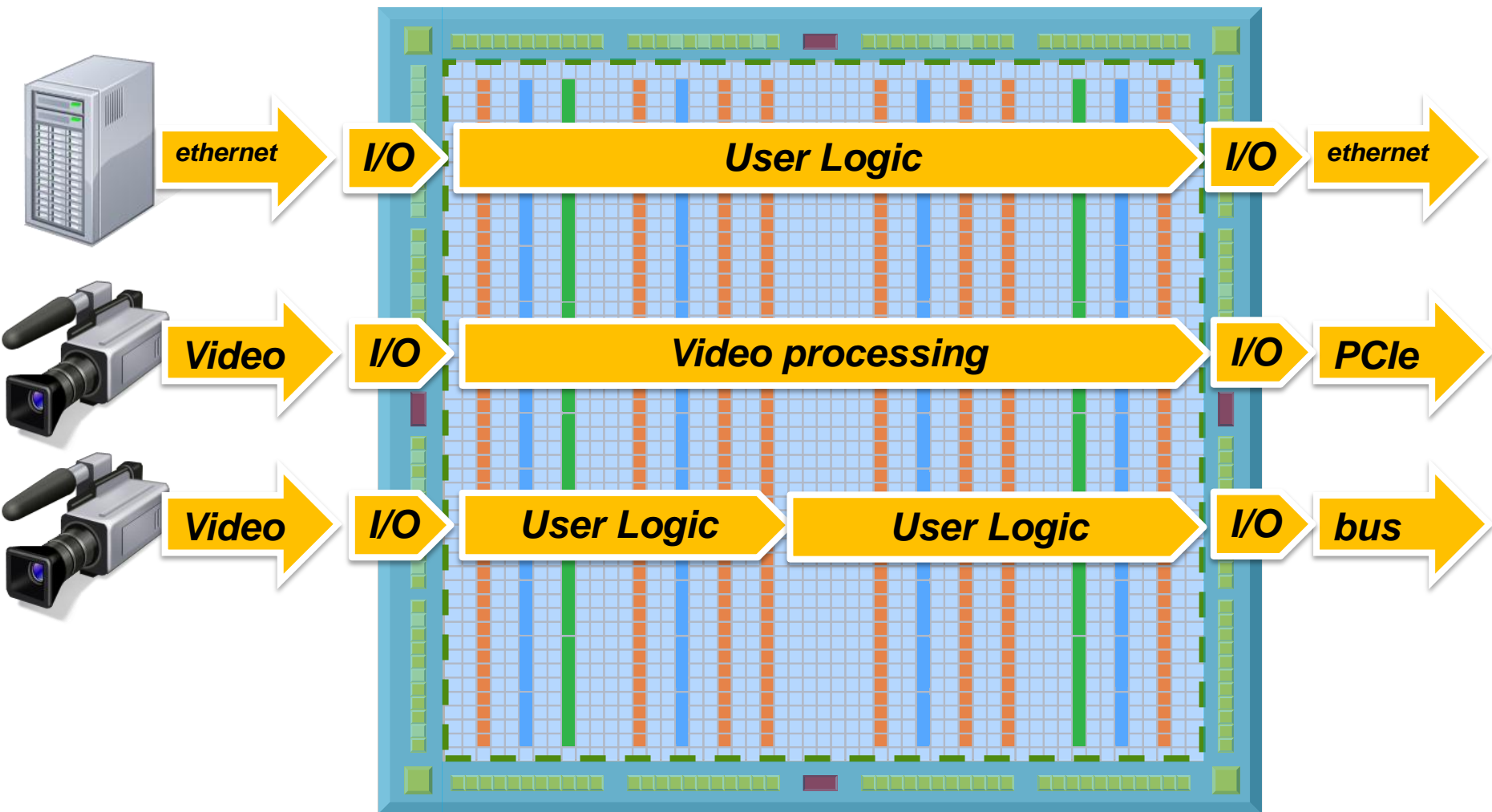
- Millions of logic elements
- Thousands of 20Kb memory blocks
- Thousands of Variable Precision DSP blocks
- Dozens of High-speed transceivers

✗ Hardware-centric

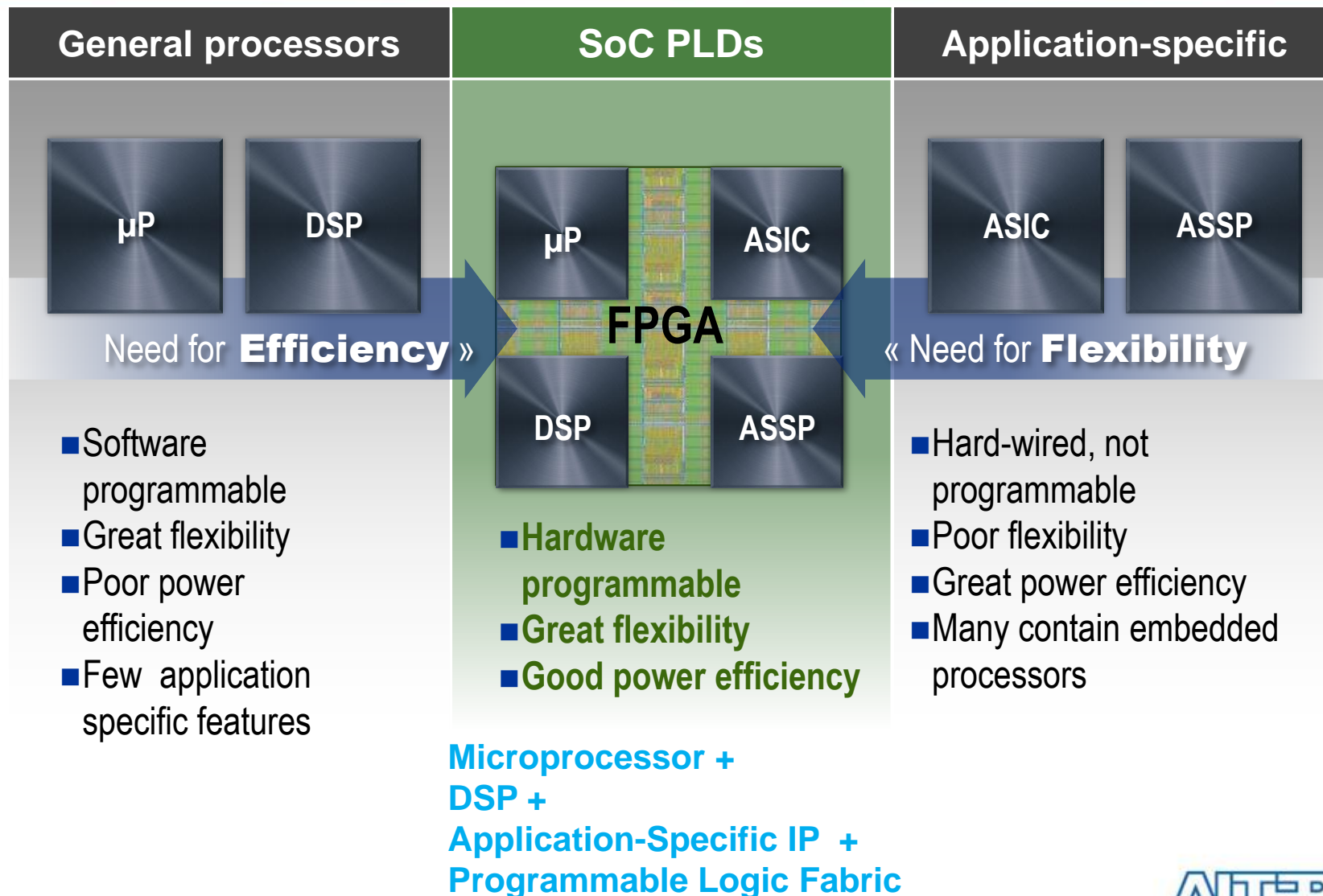
- VHDL/Verilog
- Synthesis
- Place&Route



FPGA Architecture



Altera & Silicon Convergence



Providing the Best of Both Worlds

ARM[®]

ARM Processor System

Dual Core ARM Cortex-A9
MPCore Processor

Hard
Memory
Controller

Peripherals

ALTERA[®]

28-nm FPGA

Cyclone V

Arria V

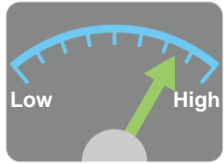


ALTERA[®]

SoC FPGA

ARM + Altera = SoC FPGAs

System-Level Benefits of Integration



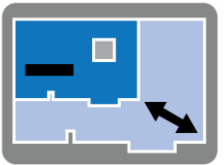
Increased system performance

- Over 4,600 DMIPS for under 1.8W
- Up to 1,600 GMACS, 300 GFLOPS DSP
- >125 Gbps processor to FPGA interconnect
- Cache coherent hardware accelerators



Reduced power consumption

- Up to 30% power savings vs. 2-chip solution



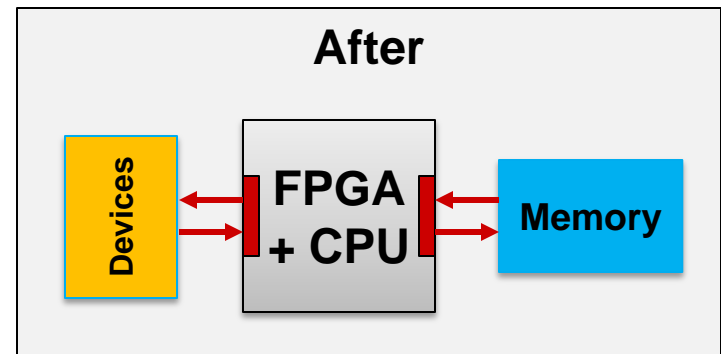
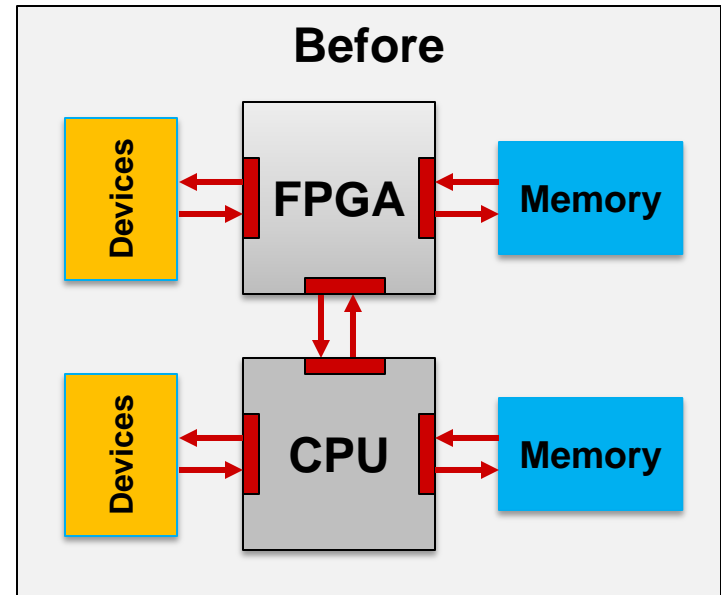
Reduced board size

- Up to 55% form factor reduction
- As few as two power rails



Reduced system costs

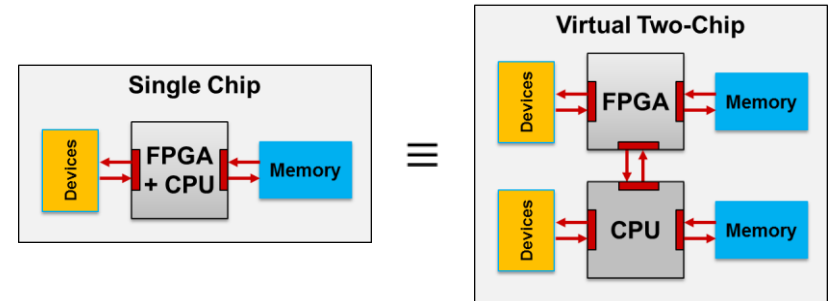
- Lower component cost
- Reduction in PCB complexity and cost
 - Less routing with fewer layers



Architecture Matters

Preserve Independence (virtual 2-chip operation)

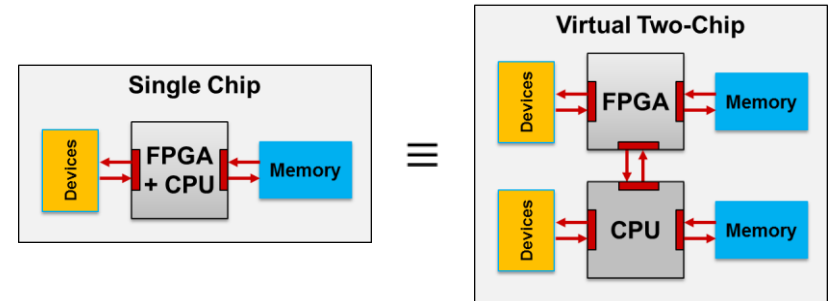
- Processor boot / FPGA configuration
- FPGA operates even with CPU reset
- Independent FPGA / CPU memories



Architecture Matters

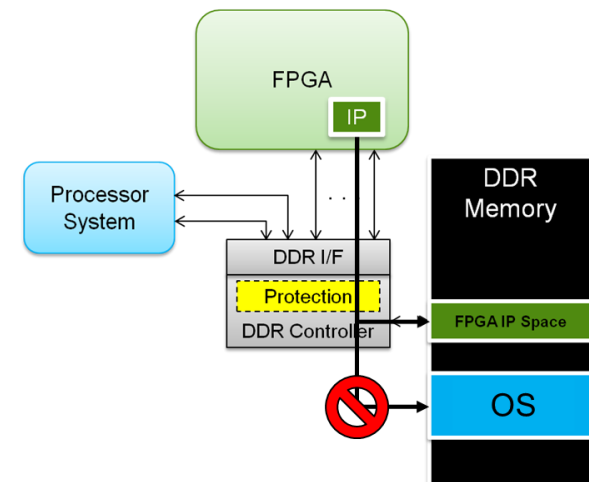
Preserve Independence (virtual 2-chip operation)

- Processor boot / FPGA configuration
- FPGA operates even with CPU reset
- Independent FPGA / CPU memories

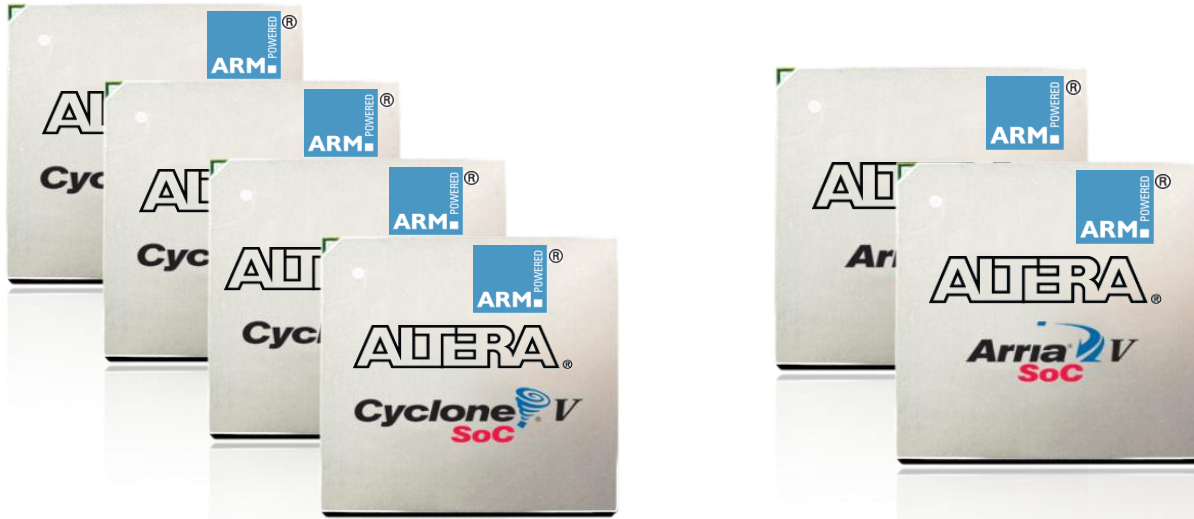


Protect Memory When Shared

- CPU memory protected from FPGA IP



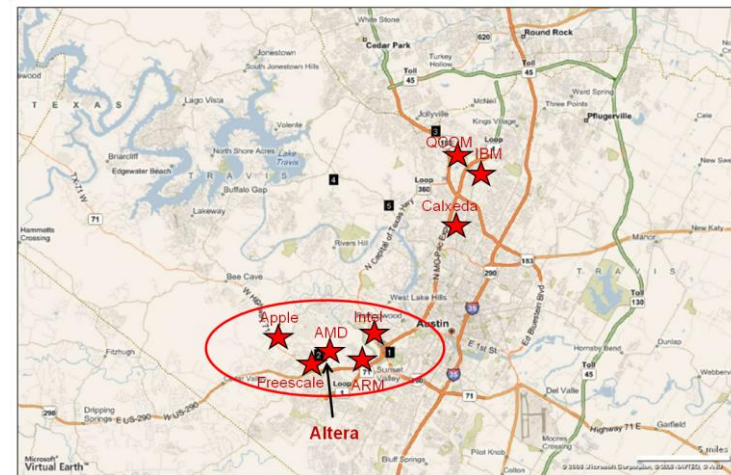
SoC FPGA Device Portfolio



**Tailored to Address
Diverse Application Requirements**

Altera Investment in Embedded Technologies

- Altera established Austin Technology Center (ATC) in 2011
- Altera's primary embedded engineering center
- Austin provides access to one of the richest embedded processing talent bases in the world





Technical Details



System Architecture

■ Processor

- Dual-core ARM® Cortex™-A9 MPCore™ processor
- Up to 5,250 MIPS (1050 MHz per core maximum)
- NEON coprocessor with double-precision FPU
- 32-KB/32-KB L1 caches per core
- 512-KB shared L2 cache

■ Multiport SDRAM controller

- DDR3, DDR3L, DDR2, LPDDR2
- Integrated ECC support

■ High-bandwidth on-chip interfaces

- > 125-Gbps HPS-to-FPGA interface
- > 125-Gbps FPGA-to-SDRAM interface

■ Cost- and power-optimized FPGA fabric

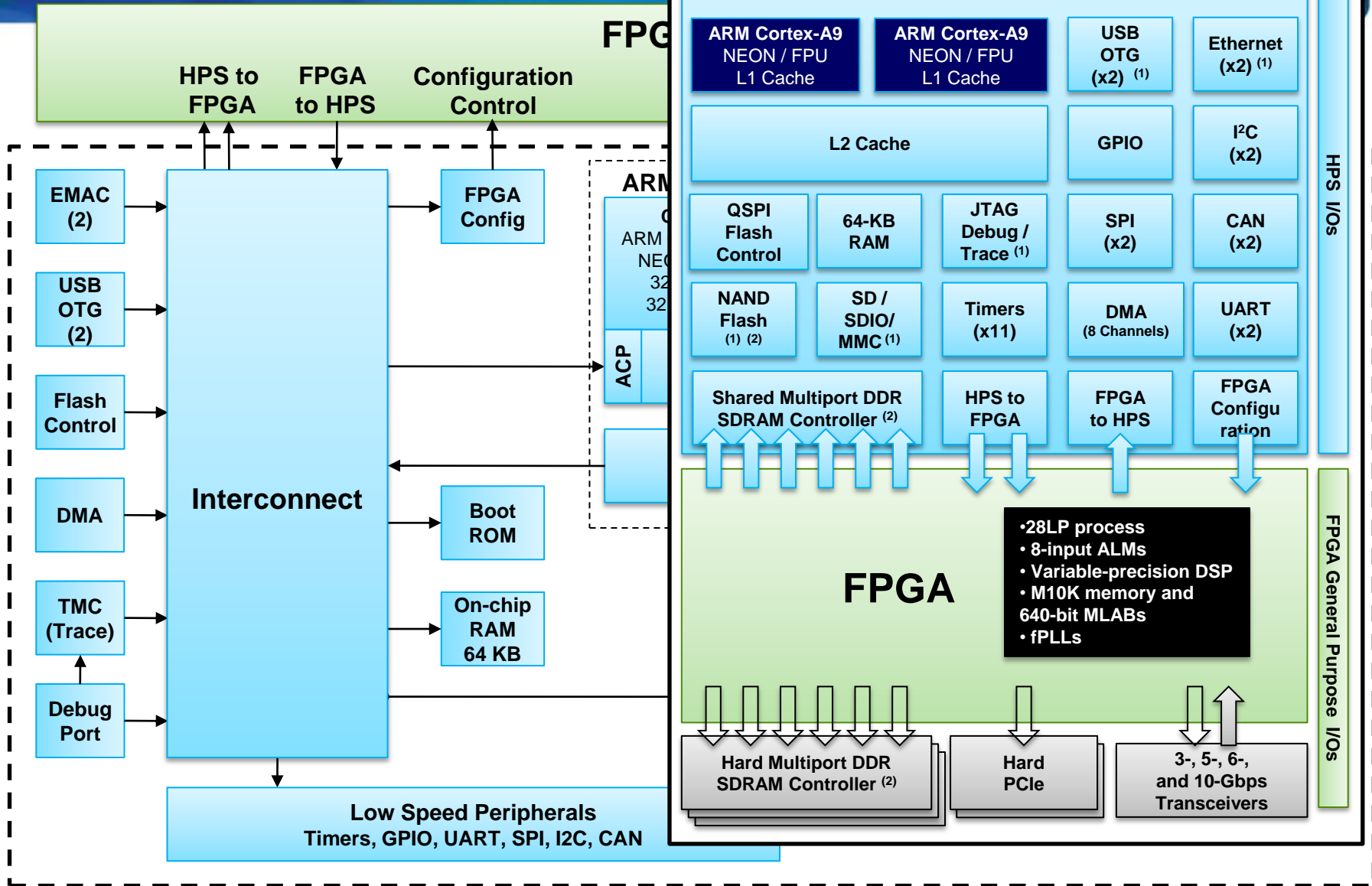
- Lowest power transceivers
- Up to 1,600 GMACS, 300 GFLOPS
- Up to 25Mb on-chip RAM
- More hard intellectual property (IP): PCIe® and memory controllers



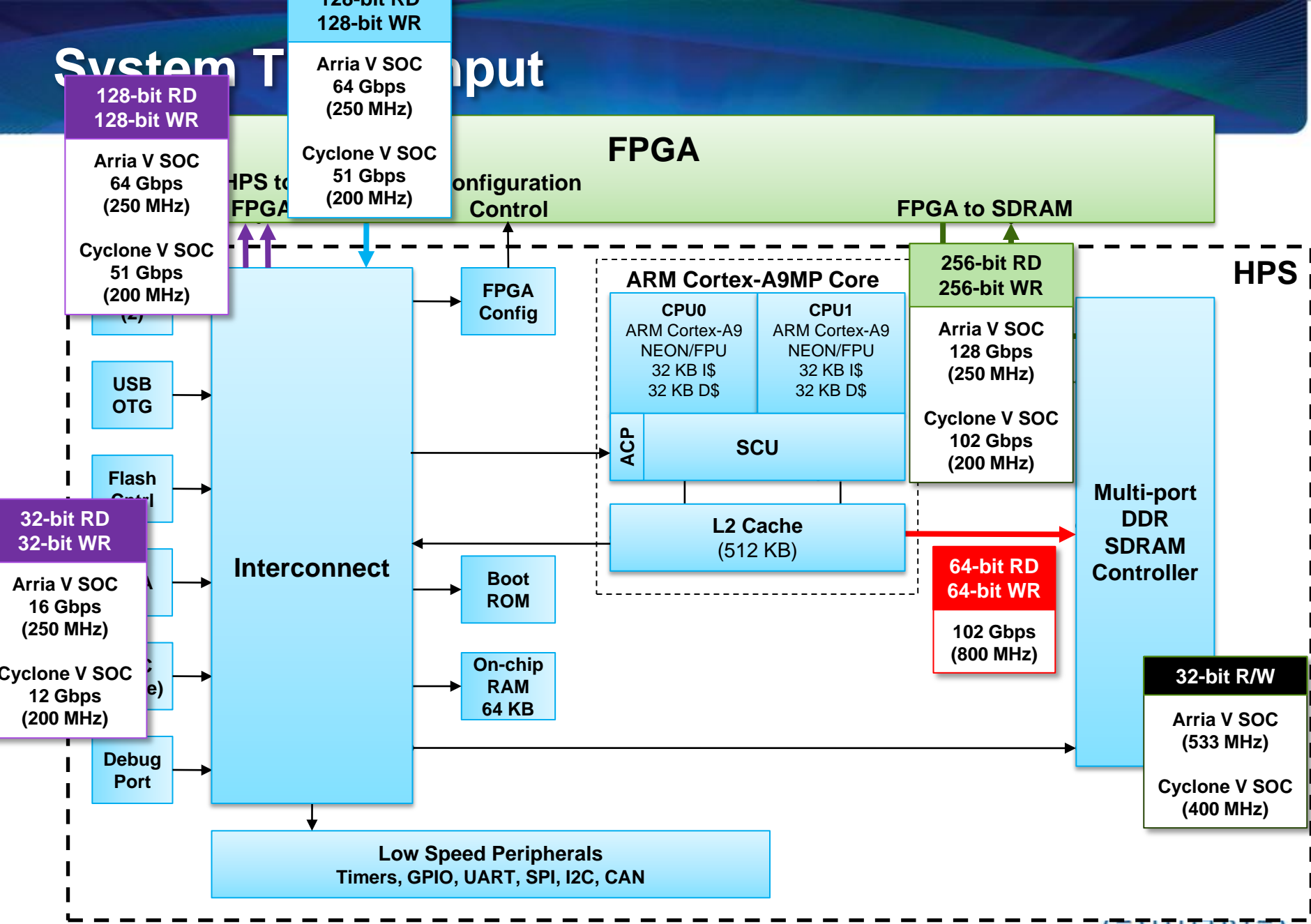
Notes:

- (1) Integrated direct memory access (DMA)
- (2) Integrated ECC

High-Level Block Diagram



System Topology



Coherent Memory Sharing

- **Coherent memory support for all masters**

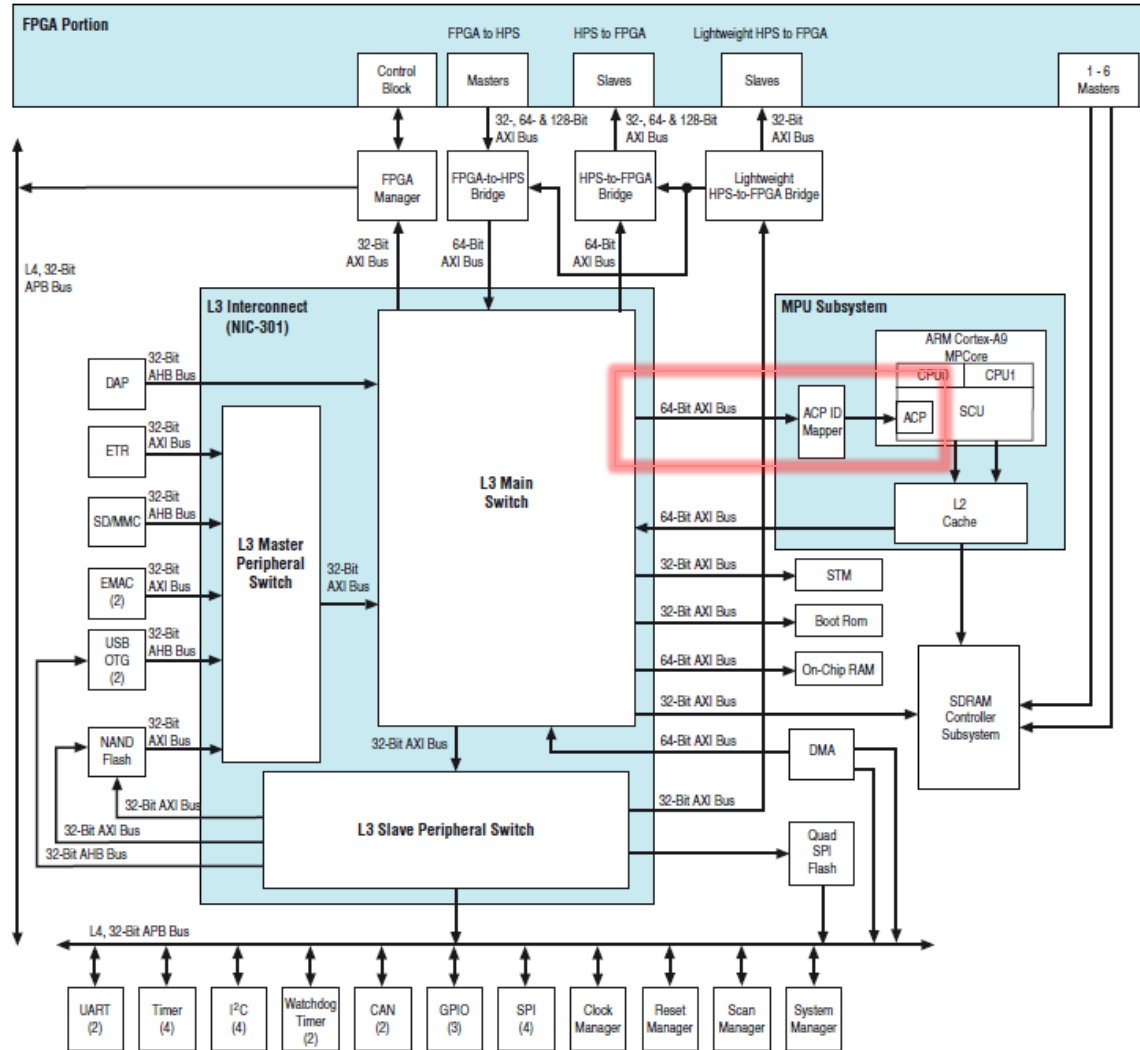
- FPGA and HPS

- **ACP Mapper removes 8-master limit**

- Up to 8 transactions in flight
- Unlimited transactions pending

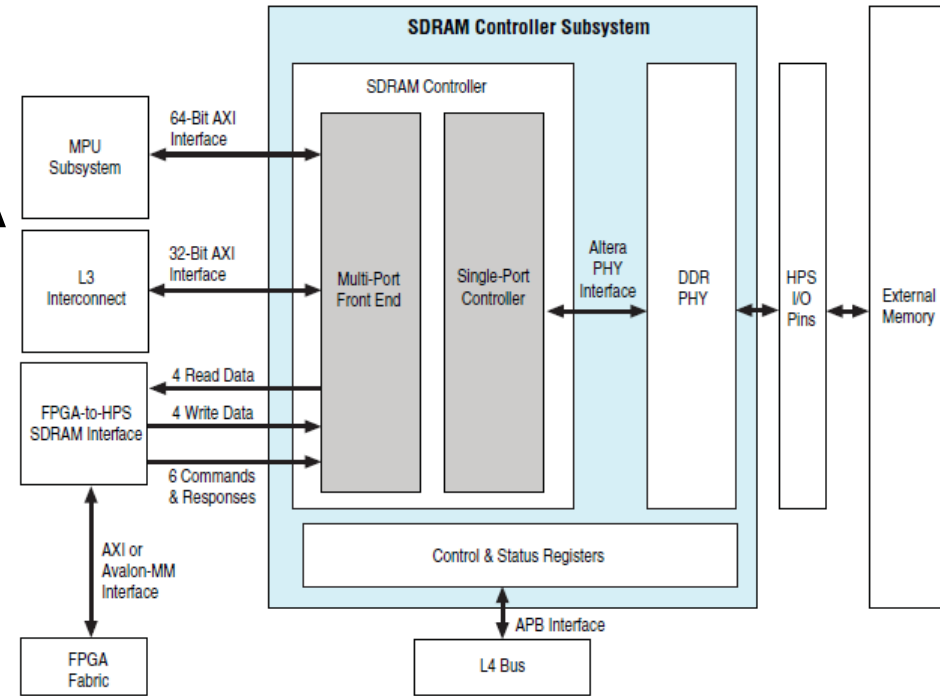
- **High BW**

- 64-bit port running at $\frac{1}{2}$ CPU clock



HPS DDR Memory Controller

- One 64-bit AXI port for CPU
- One 32-bit AXI port for L3
- Up to six ports shared with FPGA
 - Six CMD ports
 - Four 64-bit RD ports
 - Four 64-bit WR ports
 - Direct connection (no switch)
- Variable DRAM port width
 - 8-bit
 - 16-bit, 16-bit + ECC
 - 32-bit, 32-bit + ECC
- Up to 4GB of address map for DDR
- TrustZone regions: 1MB boundaries



FPGA to HPS SDRAM Port Utilization

■ Total ports

- 4 Read
- 4 Write
- 6 Command

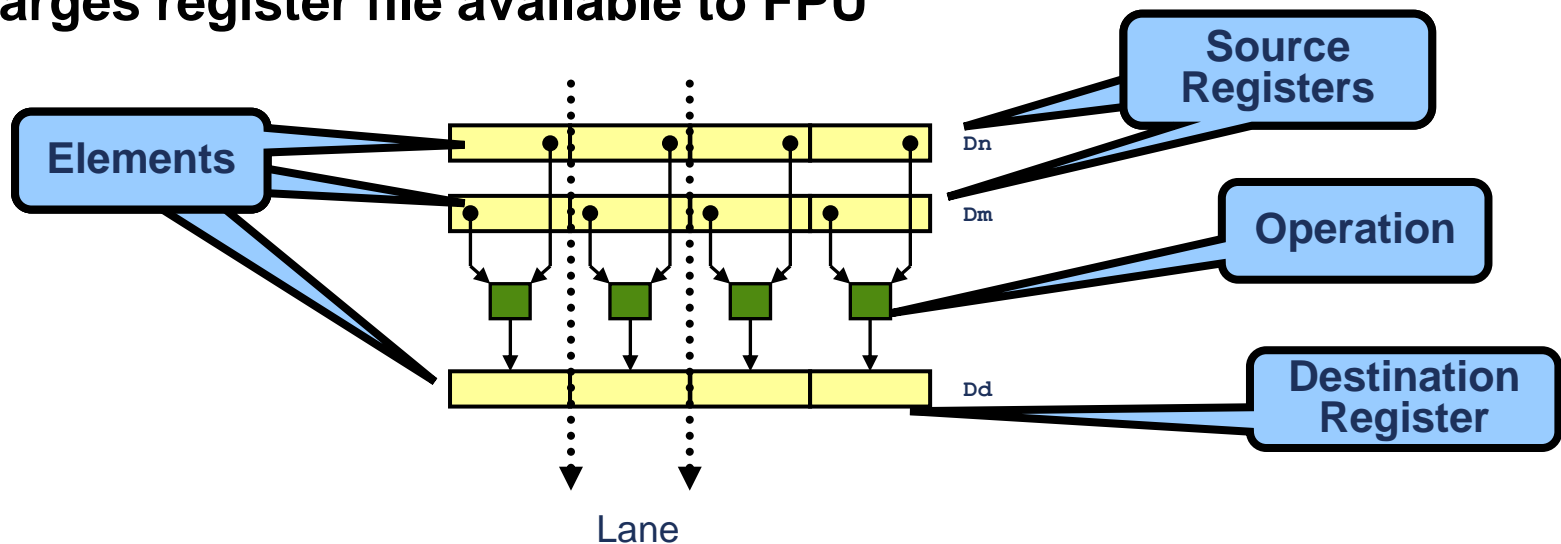
Bus Protocol	Command	Read Data	Write Data
8-, 16-, 32-, and 64-bit AXI	2 ⁽¹⁾	1	1
128-bit AXI	2 ⁽¹⁾	2 ⁽²⁾	2 ⁽²⁾
256-bit AXI	2 ⁽¹⁾	4 ⁽²⁾	4 ⁽²⁾
32- and 64-bit Avalon-MM	1	1	1
128-bit Avalon-MM	1	2	2
256-bit Avalon-MM	1	4	4
32- and 64-bit Avalon-MM write-only	1	0	1
128-bit Avalon-MM write-only	1	0	2
256-bit Avalon-MM write-only	1	0	4
32- and 64-bit Avalon-MM read-only	1	1	0
128-bit Avalon-MM read-only	1	2	0
256-bit Avalon-MM read-only	1	4	0

Note to Table 8-2:

- (1) Because the AXI protocol allows simultaneous read and write commands to be issued, two SDRAM control ports are required to form an AXI interface.
- (2) Because the native size of the data ports is 64 bits, extra read and write ports are required to form an AXI interface.

Cortex-A9 NEON Media Processing Engine

- **NEON is a wide SIMD data processing architecture**
 - Single Instruction Multiple Data (SIMD) extension of the ARM instruction set
 - 64bit and 128bit registers supporting SIMD operations on:
 - 8, 16 and 32bit integer data formats
 - 32bit floating point data
 - Registers are considered as vectors of elements of the same data type
 - Instructions perform the same operation in all lanes
- **Enlarges register file available to FPU**





Development Flow & Tools



System Development Flow

FPGA Design Flow



Hardware Development

- Quartus II design software
- Qsys system integration tool
- Standard RTL flow
- Altera and partner IP

Design

- ModelSim, VCS, NCSim, etc.
- AMBA-AXI and Avalon bus functional models (BFMs)

Simulate

- SignalTap™ II logic analyzer
- System Console

Debug

- Quartus II Programmer
- In-system Update

Release

Software Design Flow

Software Development

- ARM Development Studio 5
- GNU toolchain
- OS/BSP: Linux, VxWorks
- Hardware Libraries
- Design Examples

Design

- Virtual Target

Simulate

- GNU, Lauterbach, DS5

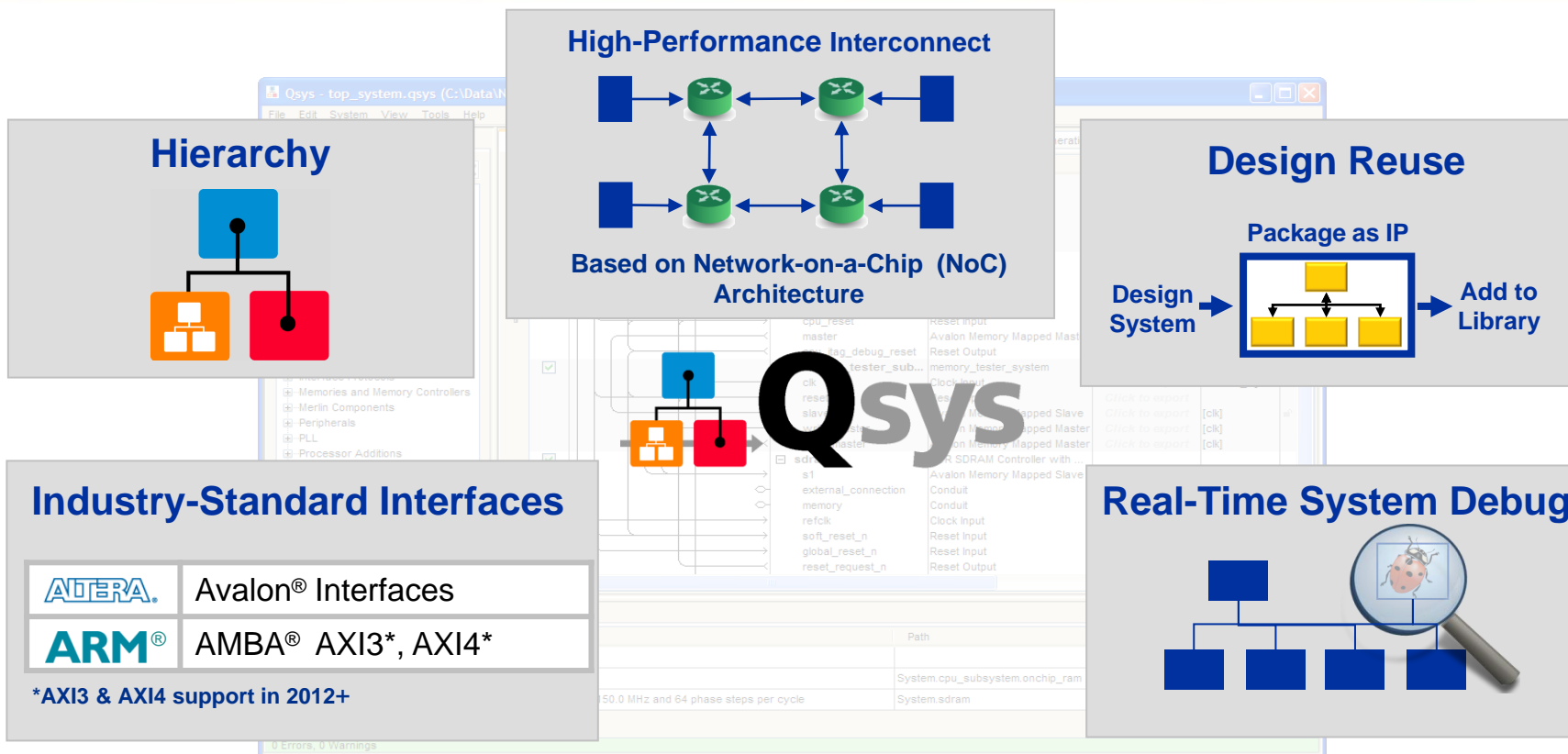
Debug

- Flash Programmer

Release



Qsys System Integration Platform



- Qsys is Altera's design environment for
 - Deployment of IP
 - Deployment of reference designs and example designs
 - Development platform for Altera custom solutions
 - Design platform for customers to quickly create system designs

Altera SoC Embedded Design Suite

FPGA Design Flow

Software Design Flow



Hardware Development

Software Development

Design

HW/SW Handoff

Design

Simulate

Simulate

Debug

FPGA-Adaptive Debugging

Debug

Release

Release

- ARM Development Studio 5
- GNU toolchain
- OS/BSP: Linux, VxWorks
- Hardware Libraries
- Design Examples

Virtual **Software Development**

- GNU, Lauterbach, DS5
- Flash Programmer

- Quartus II design software
- Qsys system integration tool
- Standard RTL flow
- Altera and partner IP

- ModelSim, VCS, NCSim, etc.
- AMBA-AXI and Avalon bus functional models (BFMs)

- SignalTap™ II logic analyzer
- System Console

- Quartus II Programmer
- In-system Update



Industry First: FPGA-Adaptive Debugging



ARM® Development Studio 5 (DS-5™) Altera® Edition Toolkit

- **Best multicore debugger for the ARM architecture**
- **Exclusive OEM agreement between Altera and ARM**
- **Result of innovation in silicon, software, and business model**
 - Automatic register view of FPGA peripherals
 - Hardware cross-triggering between the CPU and FPGA domains
 - Single USB-Blaster target connection for software and hardware debug

Visualization of SoC Peripherals

- **Register views assist the debug of FPGA peripherals**
 - File generated by FPGA tool flow
 - Automatically imported in DS-5 Debugger
- **Debug views for debug of software drivers**
 - Self-documenting
 - Grouped by peripheral, register and bit-field

golden_ref_design_CMSIS_1_1_to_arm_v2.tcf

Registers

Use this tab to create or modify the target registers. Register bitfields can be edited in the title bar. Mandatory fields are indicated by an asterisk.

	*Unique Name	Name	Base Address	*Offset
1	JTAG_UART_inst_0_CON...	Control	JTAG_UAR...	0x00000004
2	JTAG_UART_inst_0_DATA	Data	JTAG_UAR...	0x00000000
3	PIO_inst_0_CLEAR_BITS	Outclear	PIO_inst_0	0x00000014
4	PIO_inst_0_DATA	Data	PIO_inst_0	0x00000000
5	PIO_inst_0_DIRECTION	Direction	PIO_inst_0	0x00000004

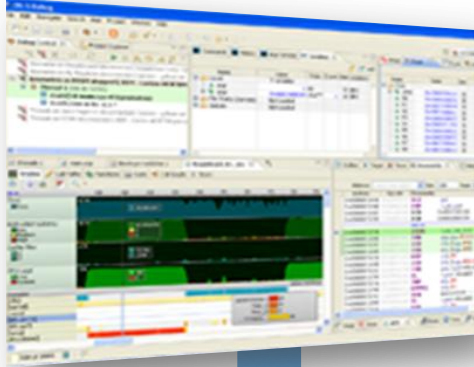
Name	Value	Size	Access
JTAG_UART_inst_0			
JTAG_UART_inst_0_DATA	0x00002000	32	R/W
JTAG_UART_inst_0_CONTROL	0x00402000	32	R/W
PIO_inst_0			
PIO_inst_0_DATA	0x00000000	32	R/W
PIO_inst_0_DIRECTION	0x00000000	32	R/W
PIO_inst_0_IRQ_MASK	0x00000000	32	R/W
PIO_inst_0_EDGE_CAP	0x00000000	32	R/W
PIO_inst_0_SET_BIT	write only	32	WO
PIO_inst_0_CLEAR_BITS	write only	32	WO
SYSID_inst_0			
SYSID_inst_0_ID	0xDEADBEEF	32	R/W
SYSID_inst_0_TIMESTAMP	0x50A91FC1	32	R/W



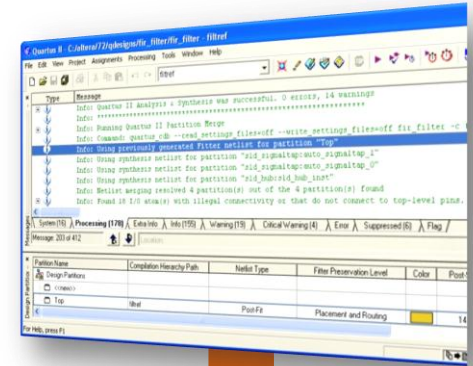
Peripheral register descriptions

One Device, Two Debugging Tools?

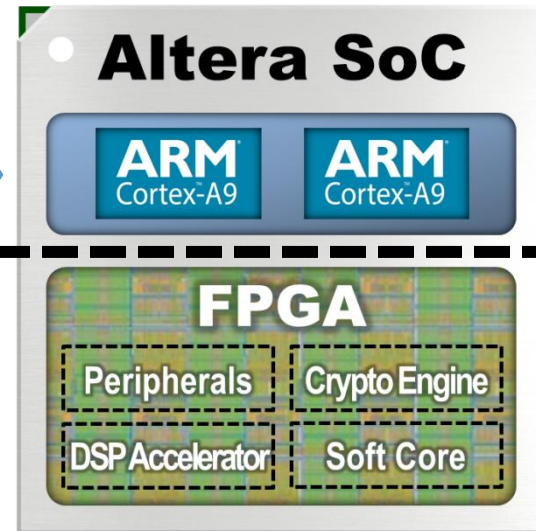
ARM® DS-5™ Toolkit



Altera Quartus™ II Software



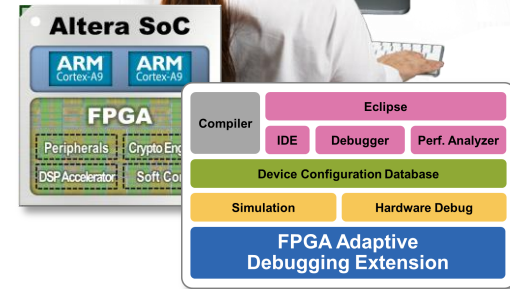
- Dedicated JTAG connection
- Visualize & control CPU subsystem



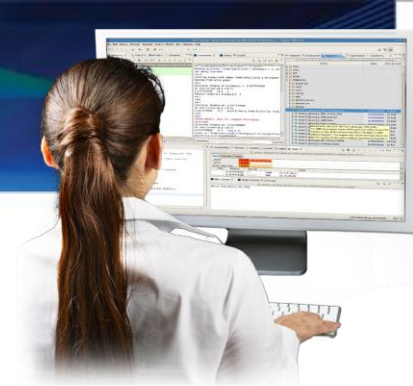
- Dedicated JTAG connection
- Visualize & control FPGA

DS-5 AE: Productivity-Boosting Features

- Single USB-Blaster target connection for software and hardware debug
- Automatic creation of register views of FPGA peripherals
- Non-intrusive trace of CPU software instructions correlated with application events and FPGA hardware events
- Hardware cross-triggering between the CPU and FPGA domains
- Simultaneous debug and trace for Cortex-A9 cores and CoreSight™-compliant cores synthesized on FPGA
- Streamline support: Statistical analysis of software load and bus traffic spanning the CPUs and FPGA

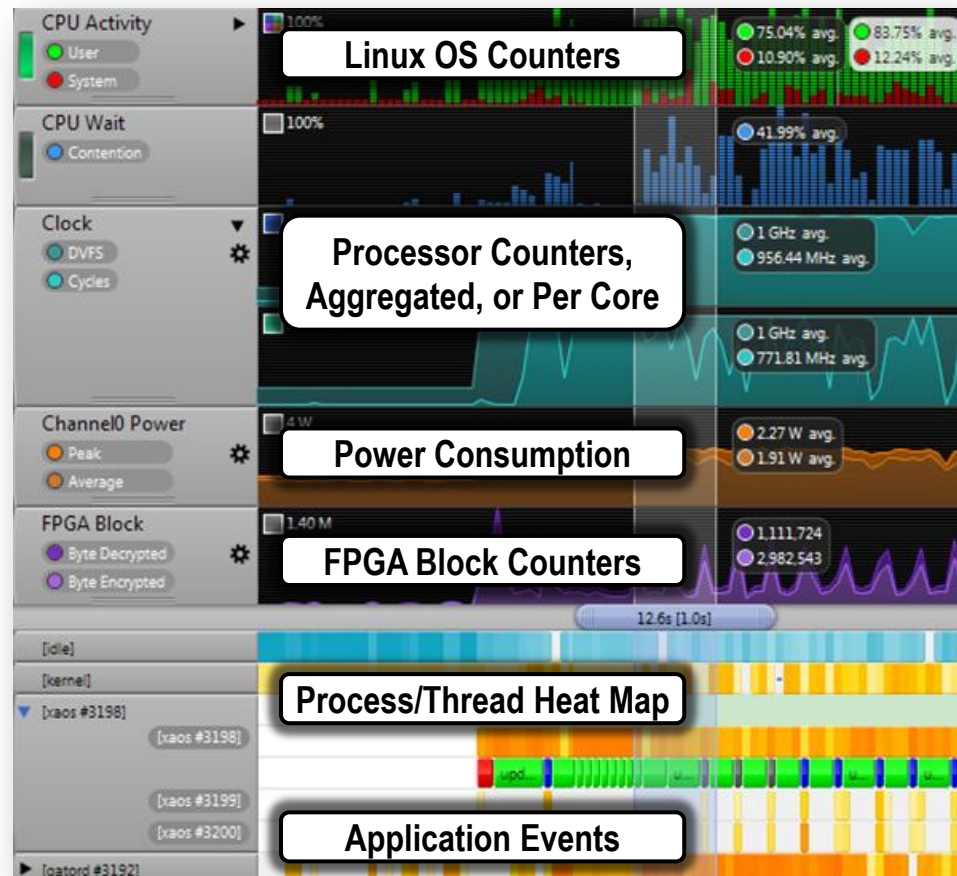


System-Level Performance Analysis



- Performance bottlenecks in SoCs often come from the CPU interaction with the rest of the SoC
- Streamline visualizes software activity with performance counters from the SoC and FPGA to enable full system-level analysis
- Streamline only requires a TCP/IP connection to the SoC

ARM® DS-5™ Streamline





Altera SoC Operating System Support



Embedded SW Availability



Vendor	OS/RTOS	Development Tools	Availability
Open Source	Linux 3.11	Linaro compiler	Now through Altera
Wind River Systems	VxWorks 6.9.3	Workbench 3.3.3	Now through Wind River
Micrium	µC/OS-II, µC/OS-III	GNU compiler	Now through Micrium
Enea	OSE 5.5.3	Optima 2.6	Now through ENEA
Express Logic	ThreadX G5.5.5.0	GNU compiler	Now through Express Logic
Wind River Systems	Wind River Linux 5	Workbench/GNU	Now through Wind River
QNX	QNX/Neutrino 6.5.3	Momentics	Now through QNX
Fujisoft	Android	GNU compiler	Now though Fujisoft
Green Hills	INTEGRITY	Multi/Green Hills	Now through Green Hills
Microsoft	Windows Embedded 7	Microsoft/Studio	Now through iWave
Code Time	Abassi	ARMCC/GCC	Q4 2013
Mentor	Nucleus	GCC	Q4 2013
eCosCentric	ECOSPRO (eCos)	GCC	Q4 2013



SoC Linux Development Branch

- Current Feature Set
 - Sources only
 - Documentation
 - Current configuration:
 - Kernel v3.9, U-Boot 2012.10, Yocto Project v1.3, Poky 8.0
- Linux Kernel v3.9 SMP
 - UART, Ethernet, I2C, LCD, EEPROM, GPIO, SD/MMC, FPGA support, USB, QSPI
 - Location: git.RocketBoards.org
- Update Kernel every 3 months
- Update Development Branch every 2 weeks



RocketBoards.org

Embedded SW Availability (Japan)

Vendor	OS/RTOS	Development Tools	Availability
eSOL	eT-Kernel (uITORN4.0)	eBinder	Now through eSOL (USB:Q4)
eForce	uC3 (uITRON 4.0)	ARMCC/SoCEDs	Now through eForce
Toppers	Toppers (uITRON extended)	EDS/Toppers	Q4 2013
Mispro	NORTi(uITRON 4.0 standard)	ARMCC	Q4 2013

Altera OpenCL SDK

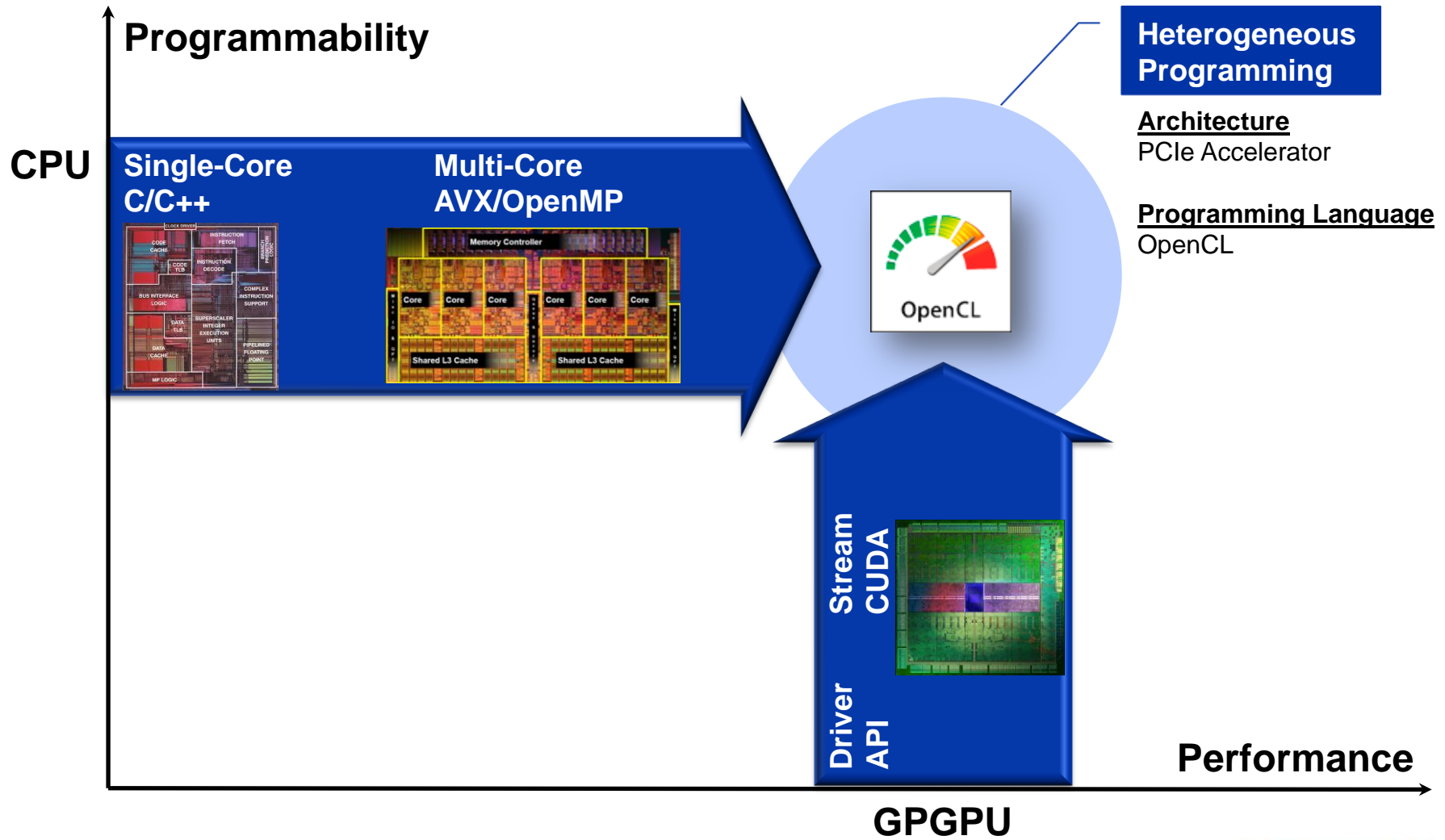




Introduction



The Quest for Performance



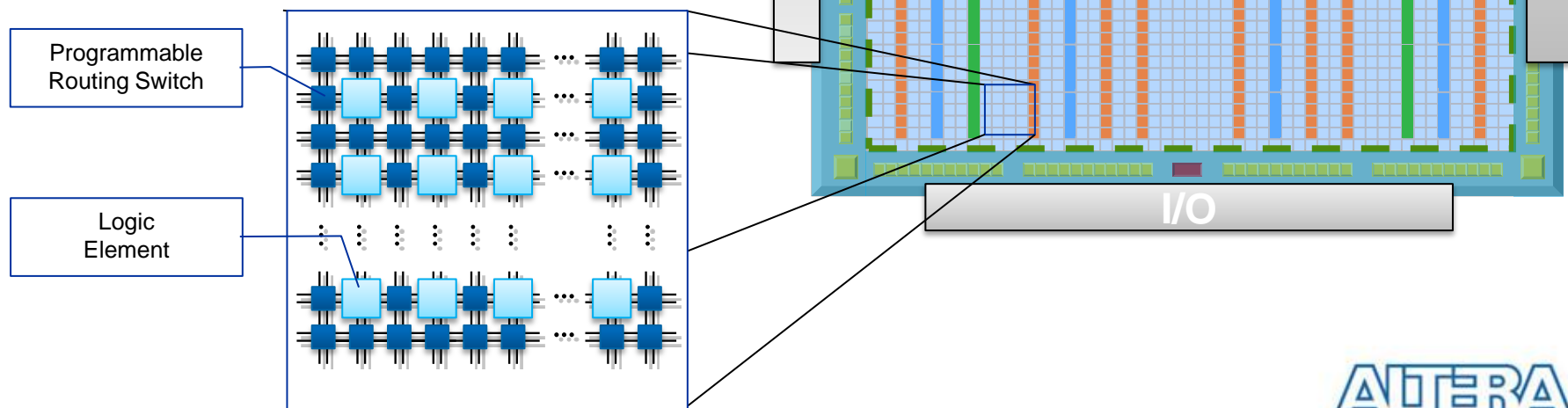
FPGA Architecture

✓ Massive Parallelism

- Millions of logic elements
- Thousands of 20Kb memory blocks
- Thousands of Variable Precision DSP blocks
- Dozens of High-speed transceivers

✗ Hardware-centric

- VHDL/Verilog
- Synthesis
- Place&Route

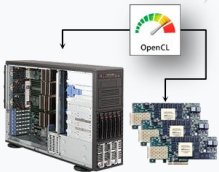
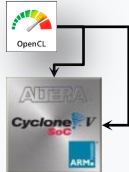






FPGA Sweet Spot

	Low Latency	SMT	SIMD	SPMD	Integer/Bit	Floating Point	Power
CPU	++ (branch prediction)	+	-	-	+	+	-
Control Compute Intensive							
DSP							
GPU							
FPGA							

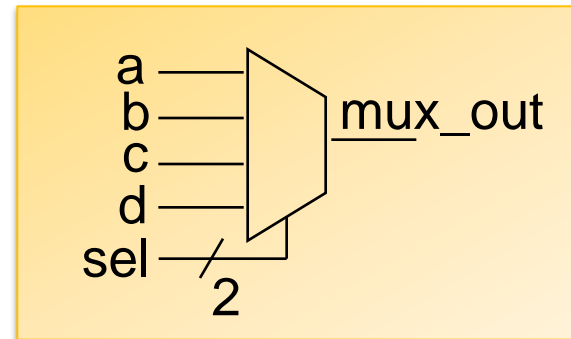


Application Sweet Spot

Current Target	GPU Multi-Core CPU	DSP/ Embedded	FPGA
Scope	System (Heterogeneous Platform) 	Device 	IP Block 
Application	<ul style="list-style-type: none"> High Performance Computing (HPC) Finance (HFT) Energy&Oil Military Medical 	<ul style="list-style-type: none"> Automotive Industrial Communications (baseband) 	<ul style="list-style-type: none"> Broadcast Consumer
Designer	Programmer	Embedded	Hardware Designer
Design Flow	CUDA/OpenCL 	Code Composer Studio (TI C) 	Quartus II (Verilog HDL) 
Design Activity	Task Parallelism Data Parallelism	Real Time Function Acceleration	IP Design and Integration HLS
Design Constraints	<ul style="list-style-type: none"> Throughput/Latency Power Efficiency 	<ul style="list-style-type: none"> Real Time Execution Cost 	<ul style="list-style-type: none"> Clock Frequency Resource Utilization Interface Requirements Power
Data Type	DP/SP Floating Point Today	SP Floating Point 32-bit Fixed Point Beta	Fixed Point Research
Hardware Knowledge	None (Coding Style Guidelines)	Limited (macro architecture bandwidth level)	Yes (protocol-level, timing closure, micro architecture)

Typical FPGA Design Entry

```
always @(a or b or c or d or sel)
begin
  case (sel)
    2'b00: mux_out = a;
    2'b01: mux_out = b;
    2'b10: mux_out = c;
    2'b11: mux_out = d;
  endcase
end
```



- **Hardware circuits are described using Hardware Description Languages (HDL) such as VHDL or Verilog**
- **A designer must describe the behavior of the algorithm to create a low level digital circuit of it**
 - Logic, Registers, Memories, State Machines, etc.

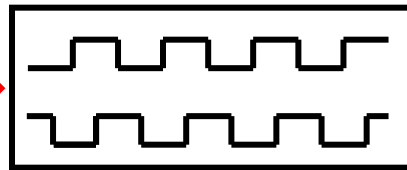
Typical PLD Design Flow

Design Specification



Schematic entry/RTL coding/Qsys/DSP Builder/OpenCL

- Behavioral or structural description of design



RTL simulation

- Functional simulation
- Verify logic model & data flow



Synthesis (Mapping)

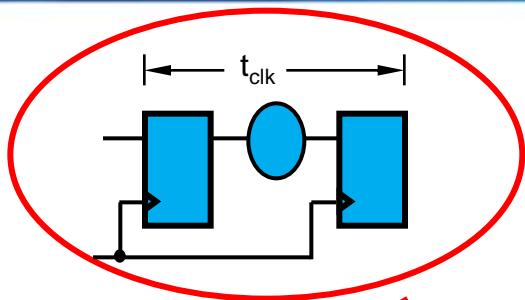
- Translate design into device specific primitives
- Optimization to meet required area & performance constraints
- Quartus II synthesis or 3rd party synthesis tools
- *Result: Post-synthesis netlist*



Place & route (Fitting)

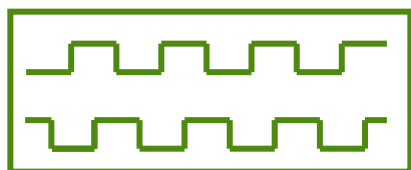
- Map primitives to specific locations inside target technology with reference to area & performance constraints
- Specify routing resources to be used
- Quartus II Fitter
- *Result: Post-fit netlist*

Typical PLD Design Flow



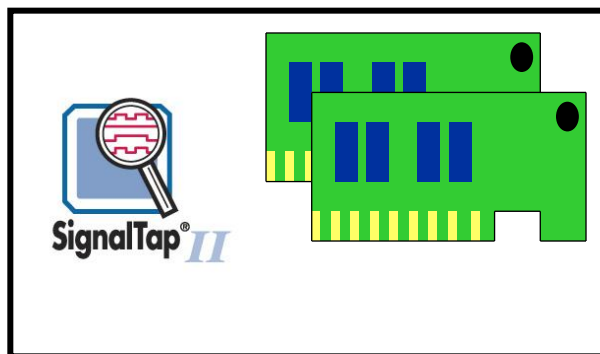
Timing analysis (TimeQuest Timing Analyzer)

- Verify performance specifications were met
- Static timing analysis



Gate level simulation (optional)

- Simulation with timing delays taken into account
- Verify design will work in target technology



PC board simulation & test

- Simulate board design
- Program & test device on board
- Use **SignalTap II** Logic Analyzer or other on-chip tools for debugging

The Key to Performance

Maximize Throughput Minimize Latency

More Operations Per Second

Parallelism

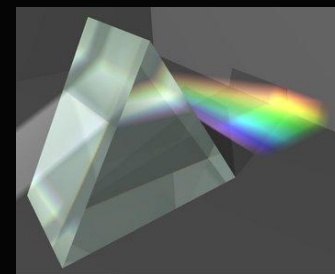
- **Pipelining**
 - Instructions
 - Processes
- Loop unrolling
- Duplication (SPMD)
- Multi-threading (SMT)



Quick Data Access

Memory Access

- Avoid transfer/copy
- Work in local memory instead of shared memory
- Coalesce accesses



FPGA Acceleration

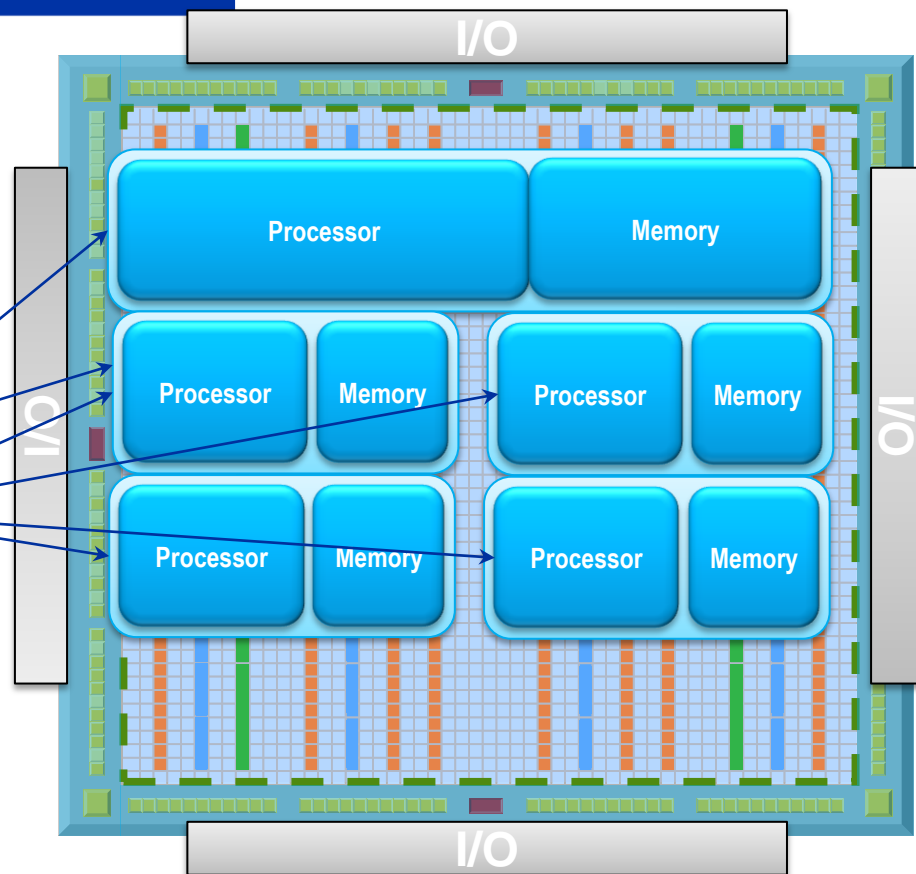
■ Custom processors

- Optimized for task
 - Small soft scalar processor
 - Larger vector processor
 - Hardware pipeline
- Dedicated local memory

■ Acceleration

- Multiple engines for SMT
- Replication for SPMD
- Pipelining for more throughput

Efficient Power





OpenCL Overview



OpenCL (Open Computing Language) Overview

■ Software programming model:

- C/C++ API for host program
- OpenCL C for acceleration device

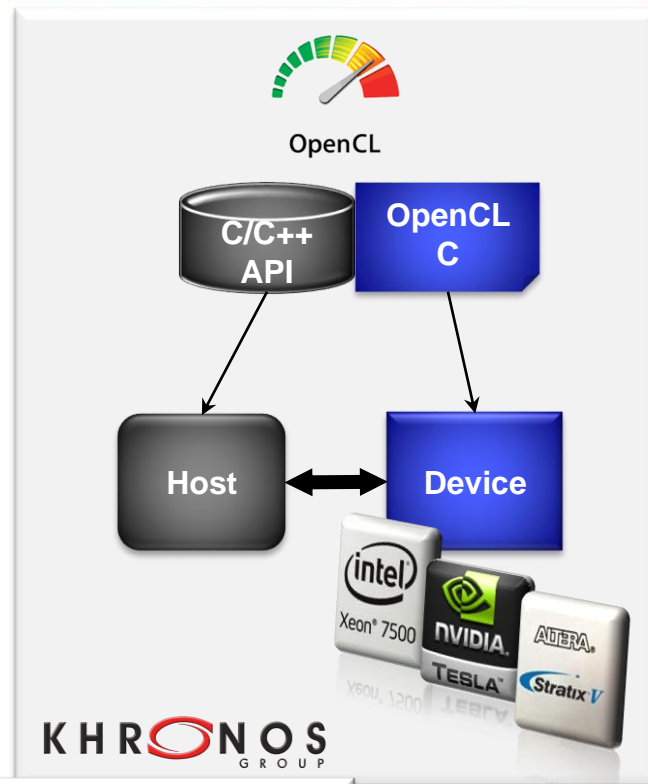
■ Provides increased performance with hardware acceleration

- CPU offload to appropriate accelerator
 - Local Memory
 - Explicit Parallelism
 - Task (SMT)
 - Data (SPMD)

Low Level Programming Language!

■ Open, royalty-free, standard

- Managed by Khronos Group
- Altera active member
- Conformance requirements
 - V1.0 is current reference
 - V2.0 is current release
- <http://www.khronos.org>



Altera OpenCL Program Overview

- **2010 research project**
 - Toronto Technology Center
- **2011 Development started**
 - Proof of concept
 - 9 customer evaluations
- **2012 Early Access Program**
 - Demo's at Supercomputing '12
 - Over 60 customer evaluations
- **2013 First public release**
 - Public announcement with release
 - Passed v1.0 conformance
- **Public release v13.1**
 - **Installation** image accessible from ACDS download infrastructure
 - **Documentation** available online
 - **Boards** available from vendor web site and ACDS installation
 - **Support** flow in place
 - **Optimization** improvements
 - **SoC** support
 - **Design Examples** on Altera.com

Passed OpenCL Conformance!

- **First FPGA to pass OpenCL conformance**
 - OpenCL v1.0 specification
- **>8500 Programs tested**



OpenCL

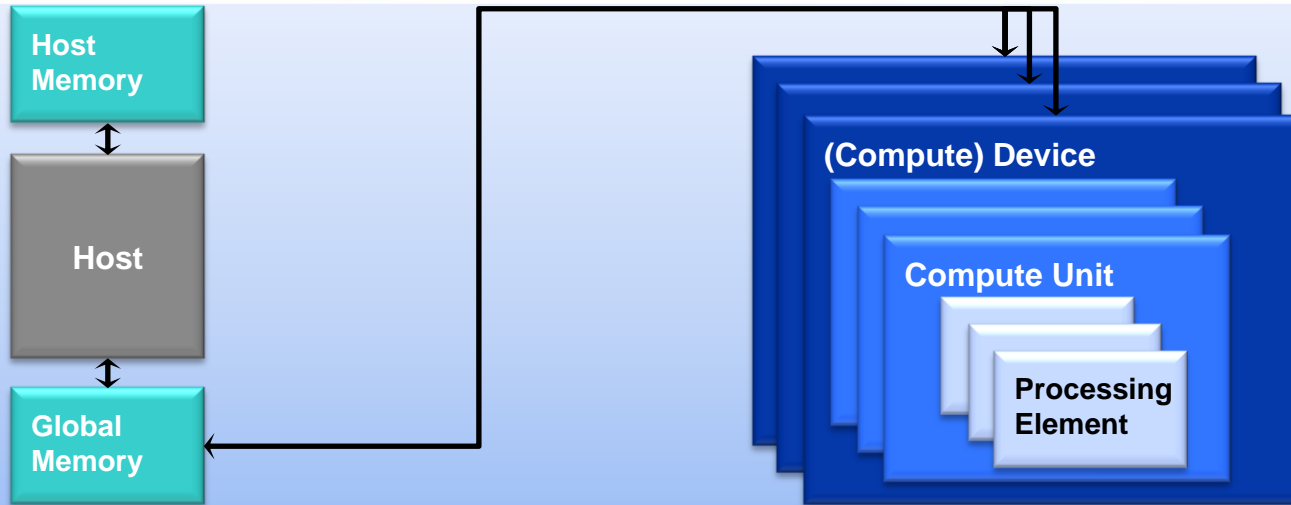
Conformant Products

OpenCL	OpenGL ES	OpenGL SC	OpenKODE	OpenV
Intel	2013-09-11	OpenCL_1_2		
Windows 8.1 64-bit pre-release build running on Confidential Product			x86_32 w	Windows
Intel	2013-09-11	OpenCL_1_2		
Product: Windows 8.1 32-bit pre-release build running on Confidential Product			x86_32 w	Windows
Intel	2013-08-23	OpenCL_1_2		
Windows 8 64-bit 9200 (RTM) running on Confidential Product			x86_32 with minimum requirement of SSE 4_2	Windows 8 64-bit RTM
Altera Corporation	2013-08-03	OpenCL_1_0		
Nallatech PCIe-385N (Altera Stratix V A7)			Altera Stratix V FPGA	
Intel	2013-07-23	OpenCL_1_2		

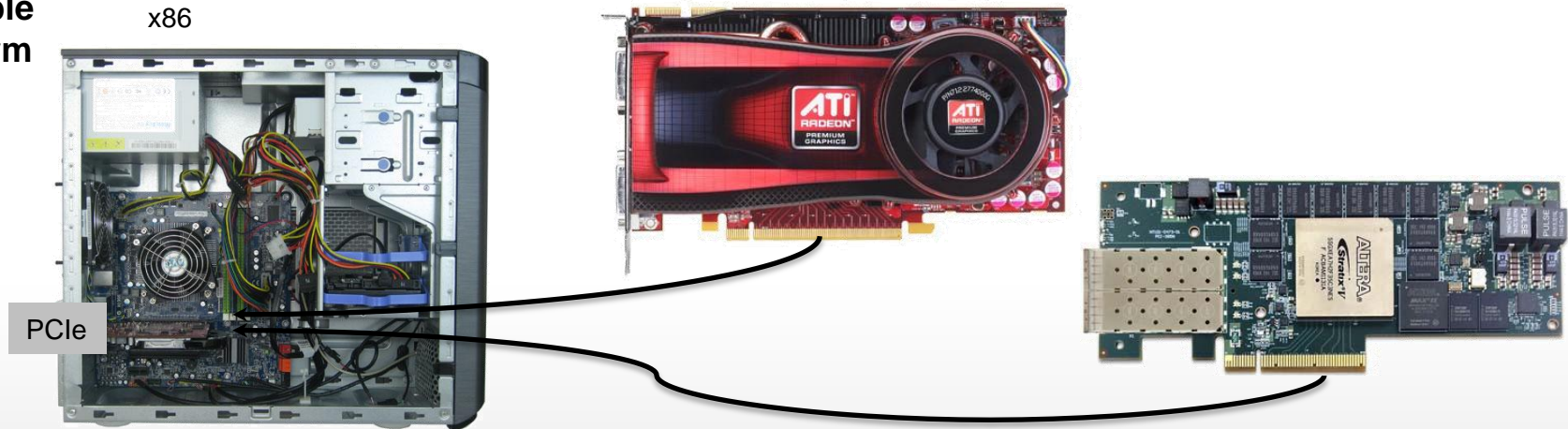
The screenshot shows the Khronos Group website with the 'Conformant Companies' page. The 'OpenCL' section is highlighted with an orange box, showing the Altera Corporation logo and description. The AMD logo is also visible below the Altera logo.

Heterogeneous Platform Model

OpenCL
Platform
Model

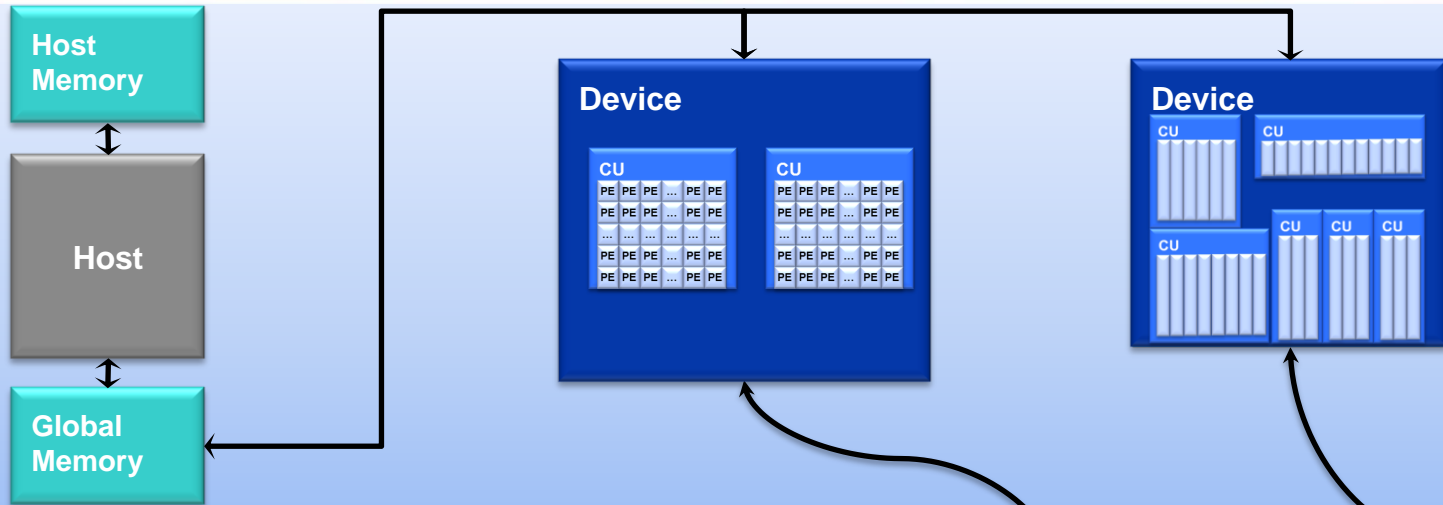


Example
Platform

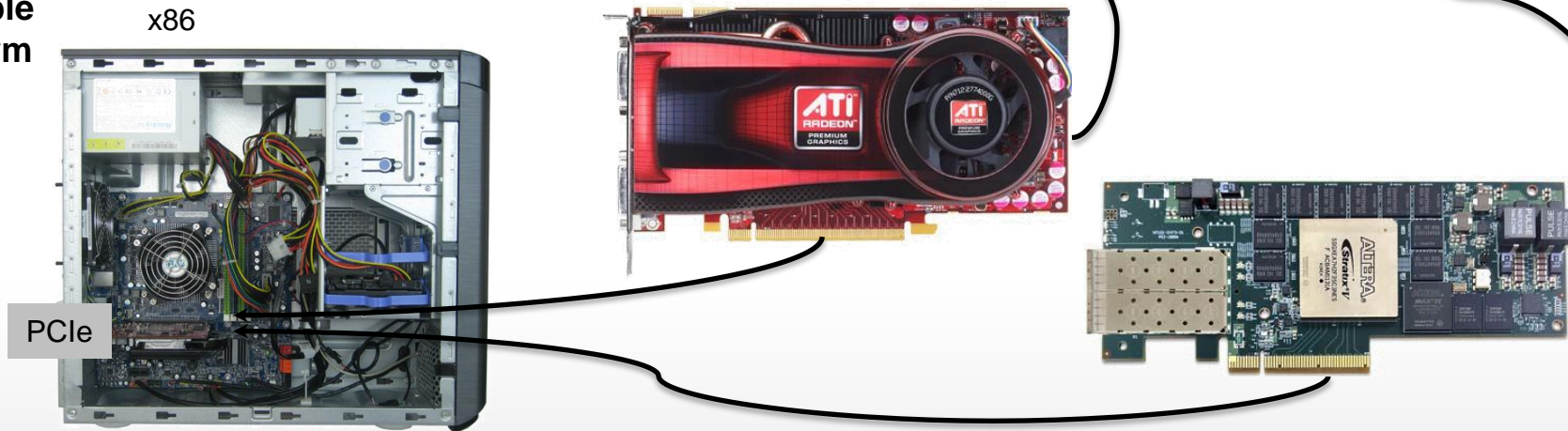


Heterogeneous Platform Model

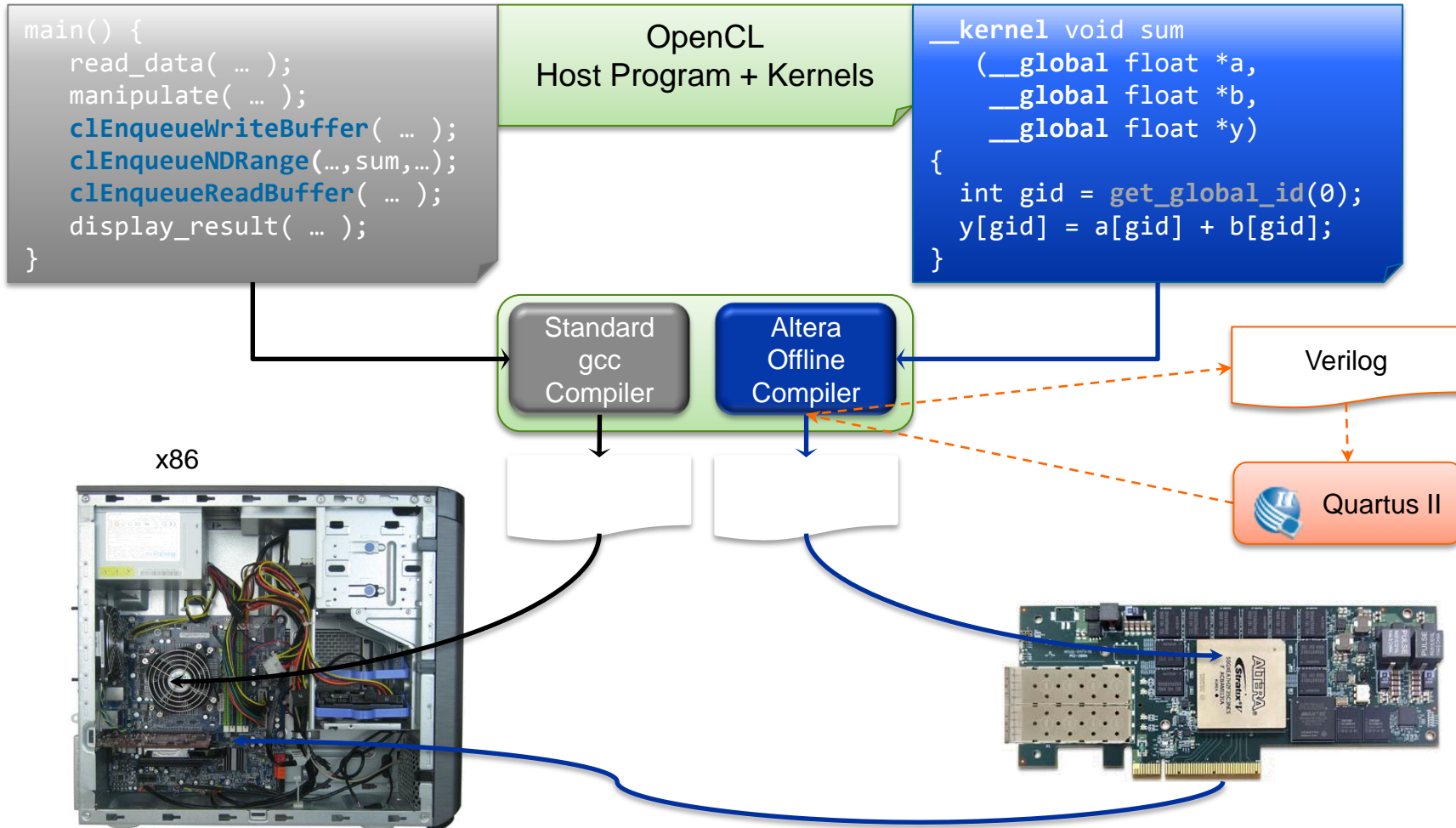
OpenCL Platform Model



Example Platform



Use Model: Abstracting the FPGA away

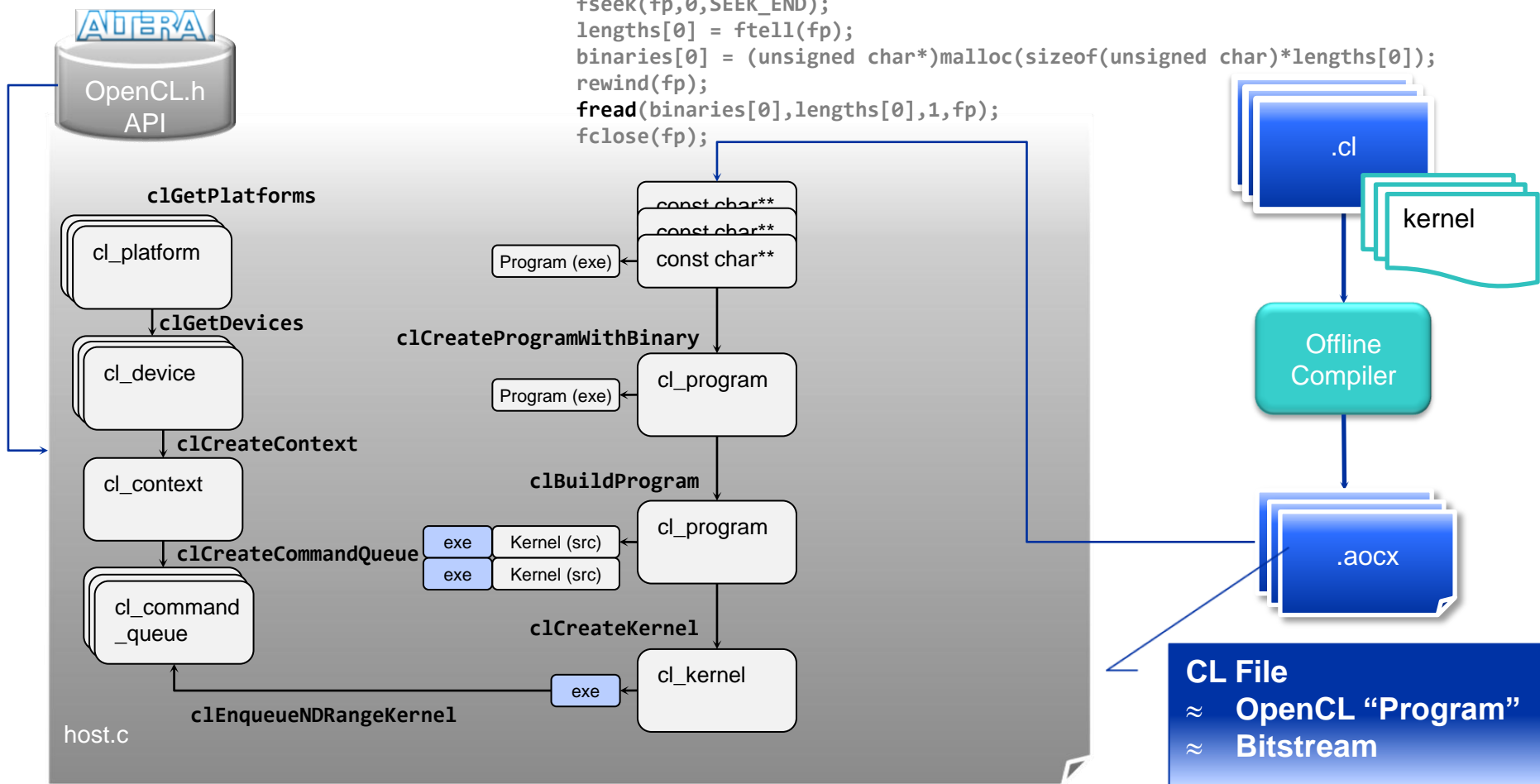


Use Model: clCreateProgramWithBinary

```

fp = fopen("file.aocx", "rb");
fseek(fp, 0, SEEK_END);
lengths[0] = ftell(fp);
binaries[0] = (unsigned char*)malloc(sizeof(unsigned char)*lengths[0]);
rewind(fp);
fread(binaries[0], lengths[0], 1, fp);
fclose(fp);

```



CL File
 ≈ OpenCL "Program"
 ≈ Bitstream

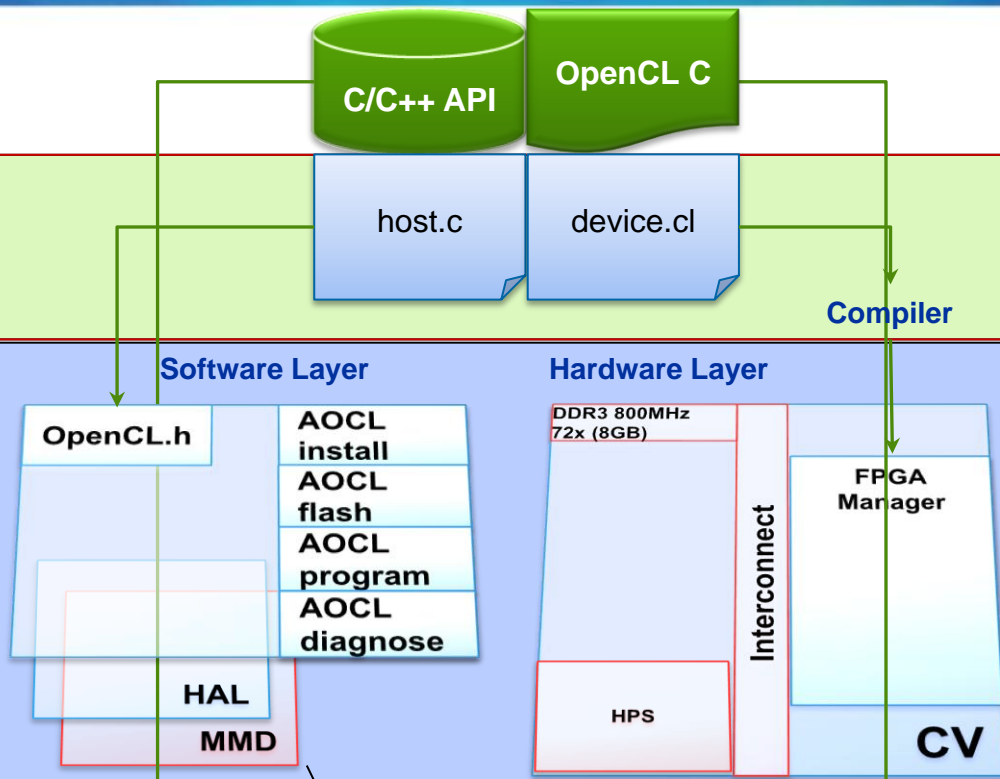
Altera Reference Platforms

	Network Enabled	High Performance Computing (HPC)
Requirement	Low Latency	Compute Power/ Memory Bandwidth
Architecture		
Global Memory	DDR and QDRII+	Large amount of DDR
IO Channels	2x10GbE (MAC/UOE)	None (Minimize IP overhead)
Reference Design	<ul style="list-style-type: none"> OPRA (Streaming) Trading (with global memory access) 	<ul style="list-style-type: none"> Option Pricing

Altera Reference Platforms

	SoC
Requirement	Real Time Acceleration
Architecture	<p>The diagram illustrates the architecture of a Cyclone V SoC FPGA. It features an ARM Host connected to HPS DDR3 (256bit, 100Mhz) via bidirectional arrows. The ARM Host is connected to OpenCL Kernels through LWH2F and F2S paths. A CSR (Control Signal Register) is also connected to the OpenCL Kernels. The OpenCL Kernels are connected to FPGA Memory. A 32bit, 50Mhz H2F/F2H path connects the ARM Host to the OpenCL Kernels. A dashed arrow indicates a connection from the OpenCL Kernels to Scratch DDR3.</p>
Global Memory	DDR
IO Channels	None
Reference Design	<ul style="list-style-type: none"> • Ray tracing (demo) • Optical Flow

Altera SoC Reference Platform for OpenCL



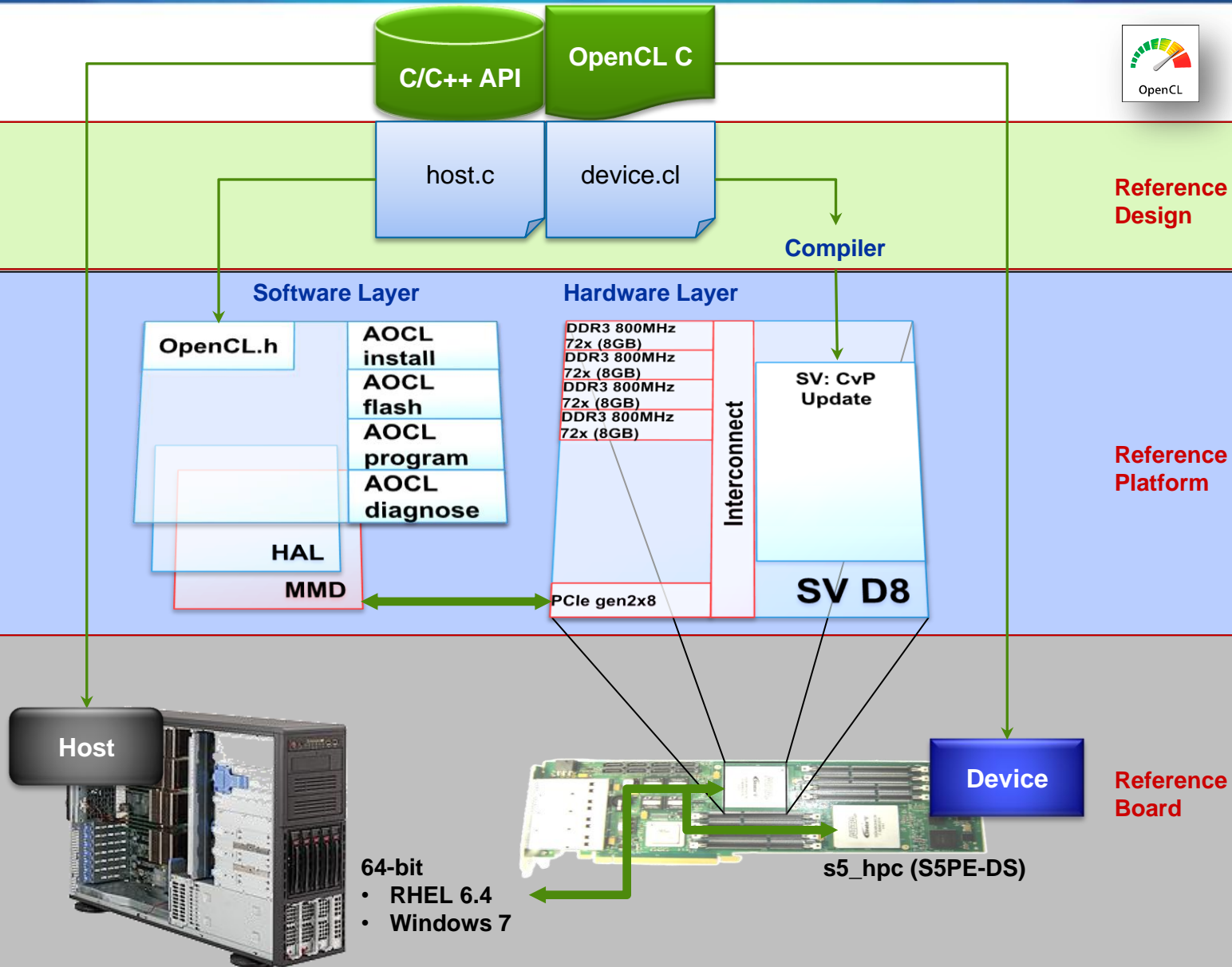
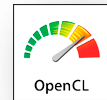
Reference Design

Reference Platform

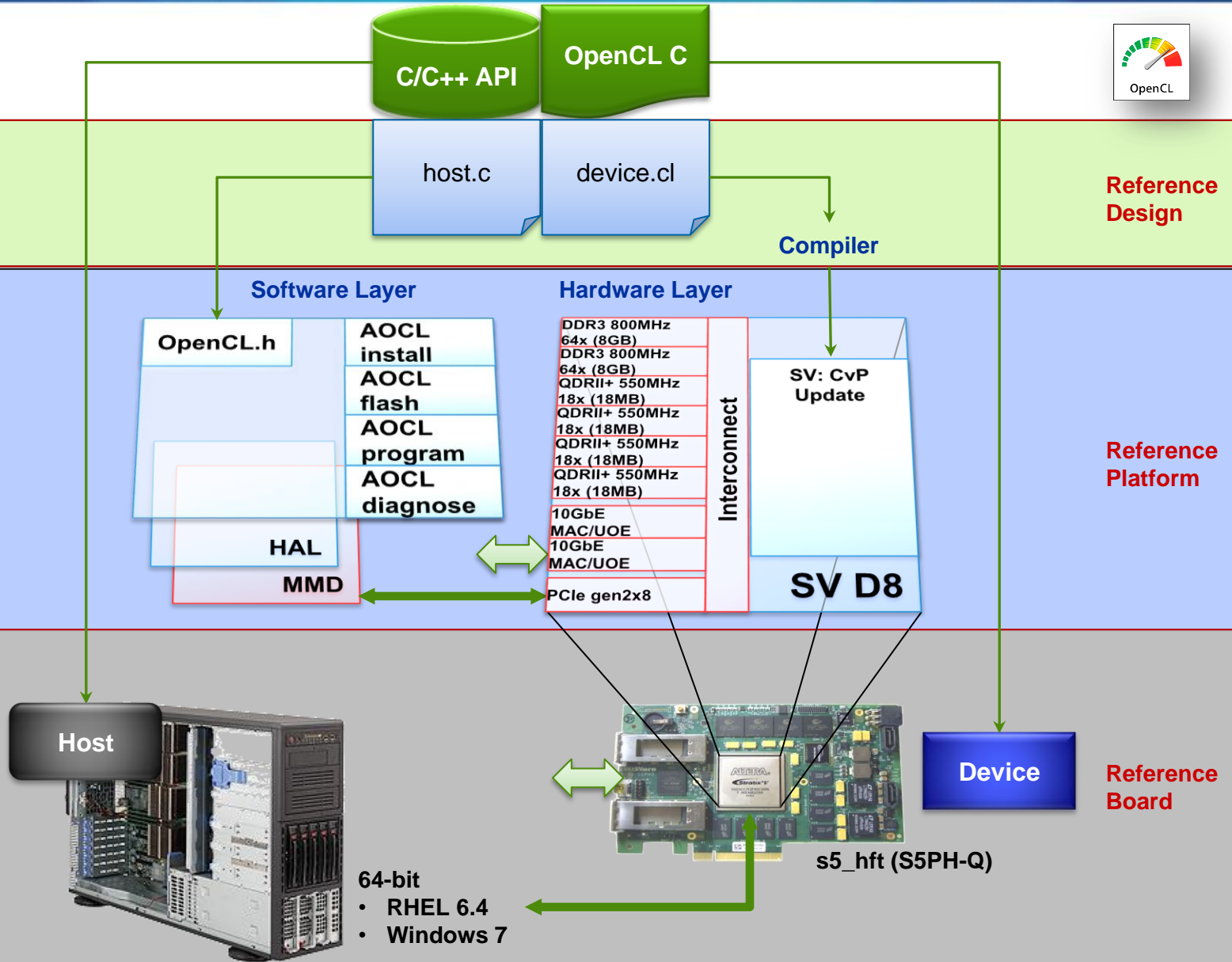
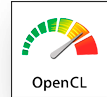
Reference Board

CV-SoC-DevKit
Arrow SoCKit

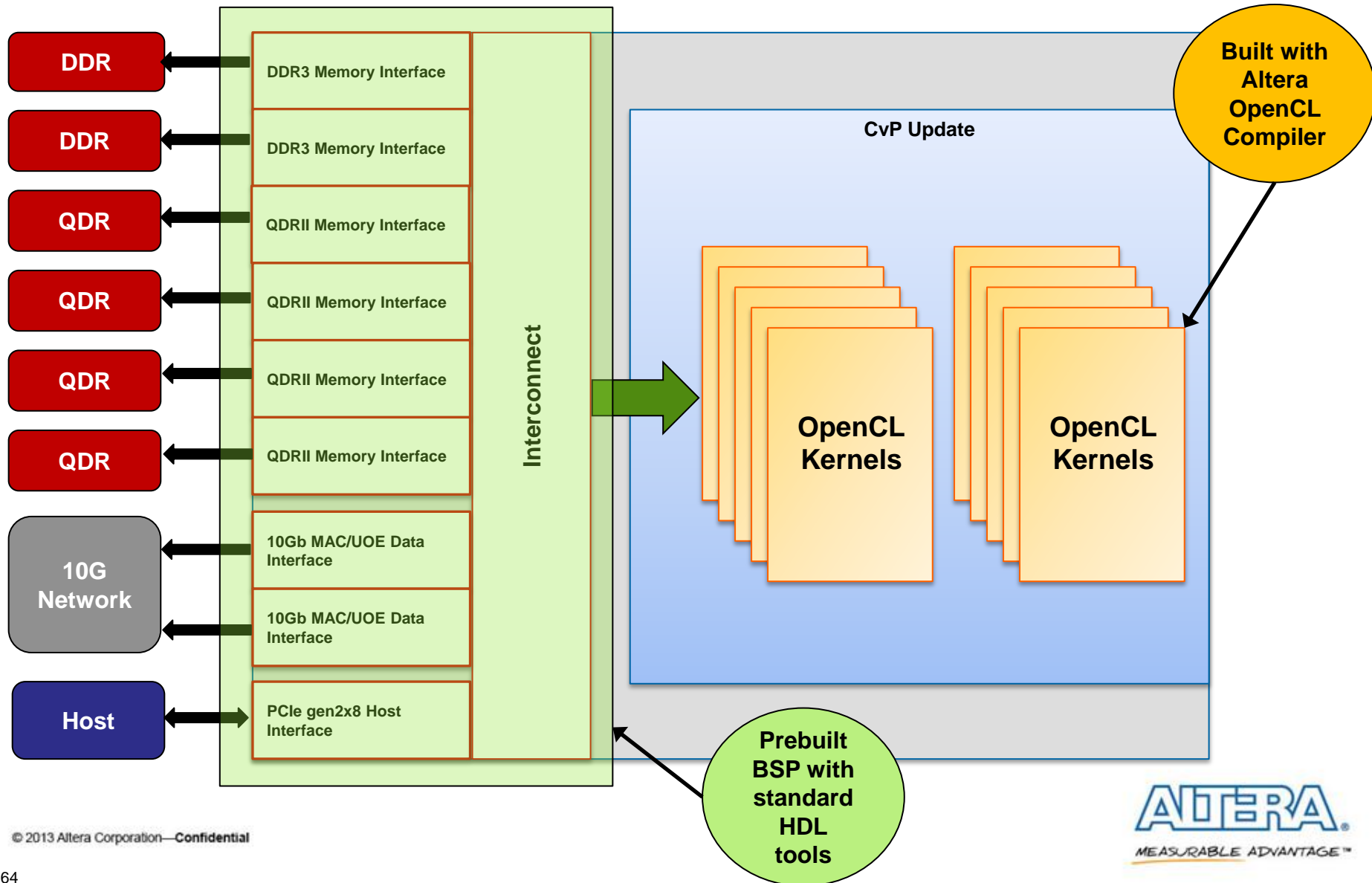
Altera HPC Reference Platform for OpenCL

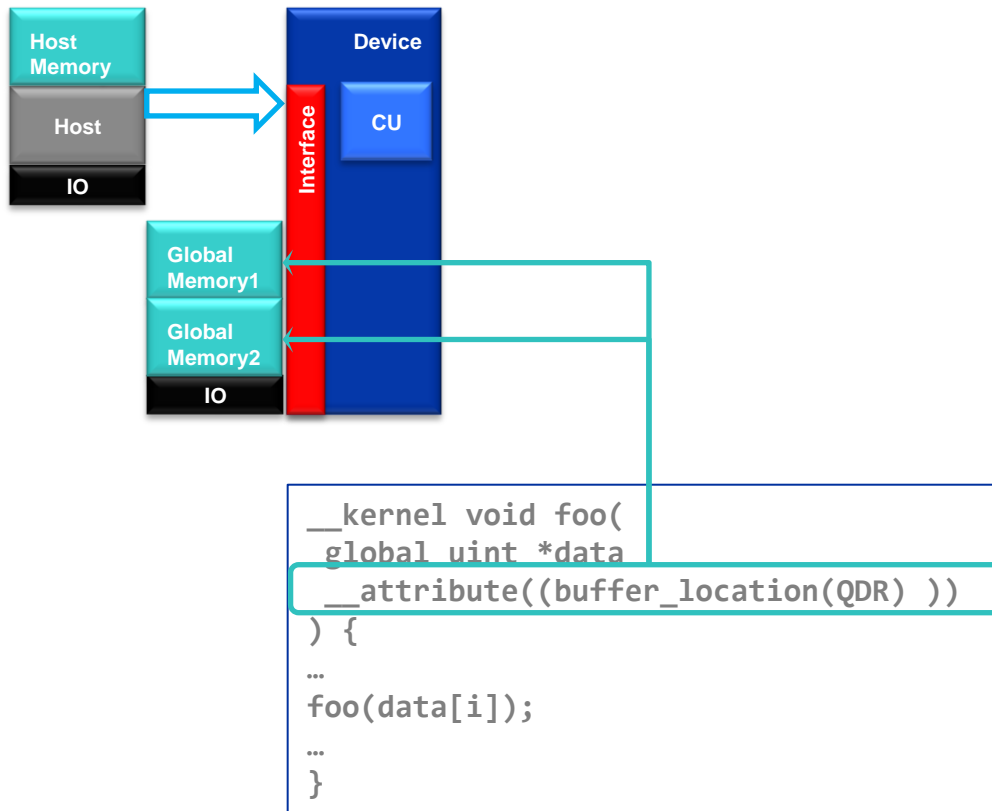


Altera Network Enabled Reference Platform for OpenCL



OpenCL Modular Hardware Architecture





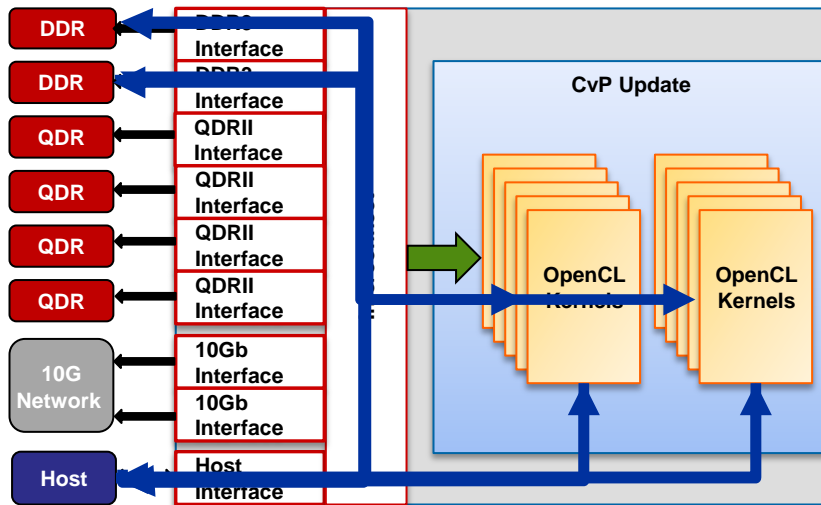
■ Memories with different characteristics

- DDR
 - Sequential Access
- QDR
 - Random Access
- On-Chip Low Latency (v14.1)
- MoSys Efficient (v15.0)
- HMC High Capacity (v15.0)

■ Combine different memories

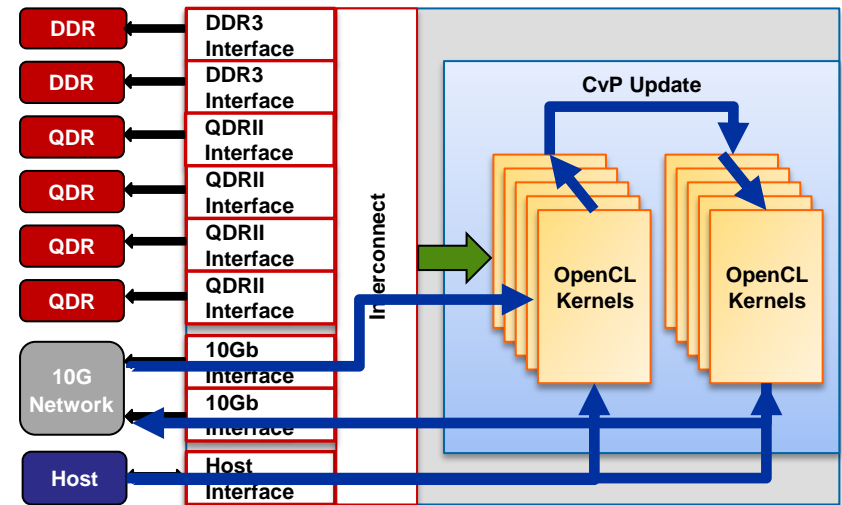
- Attribute-based

Standard OpenCL



Altera Vendor Extension Channels

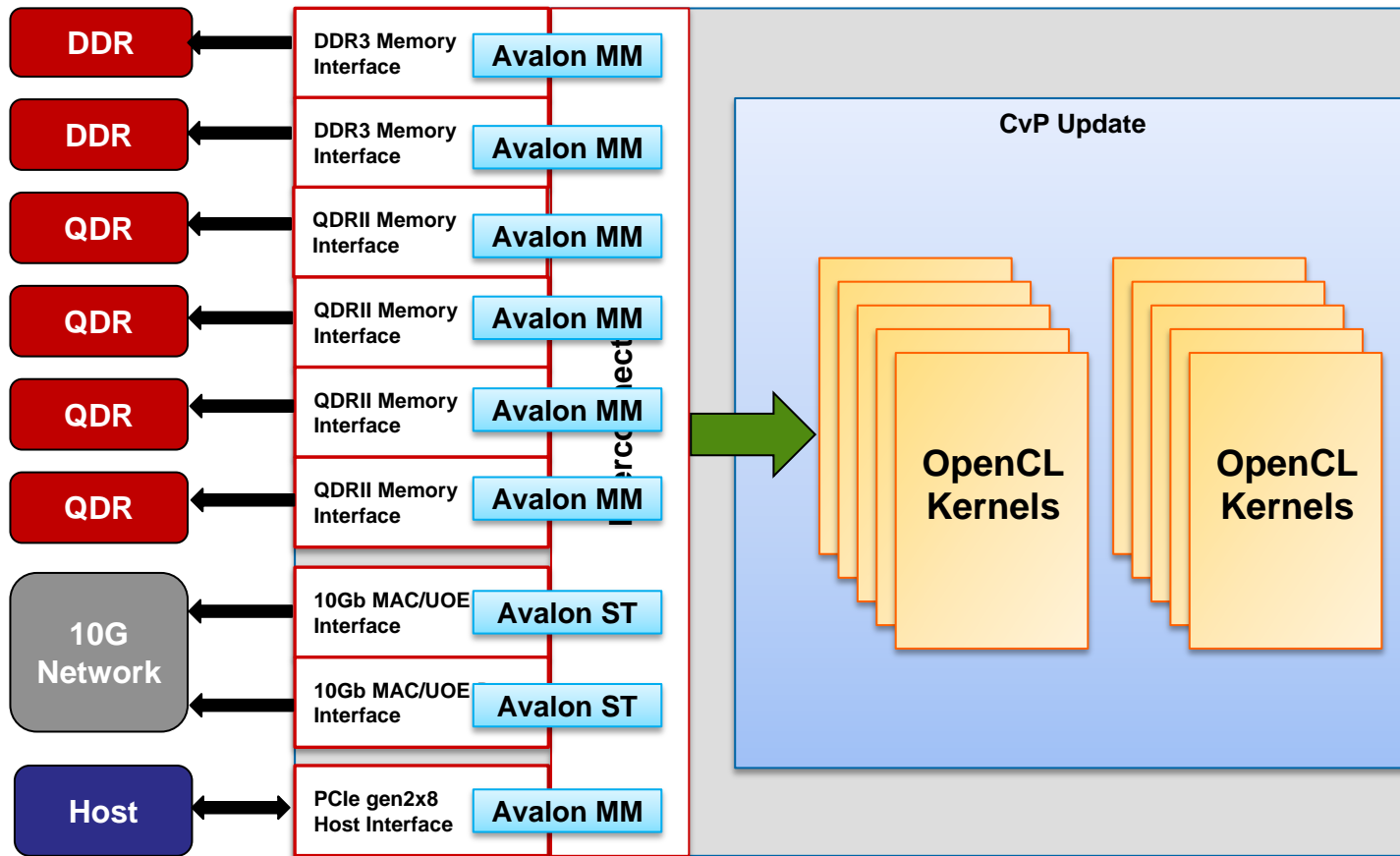
Kernel to kernel and kernel to IO



```
channel int DataChannel;

kernel producer(...) {
    write_channel_altera(DataChannel, value);
}

kernel consumer(...) {
    value = read_channel_altera(DataChannel);
}
```

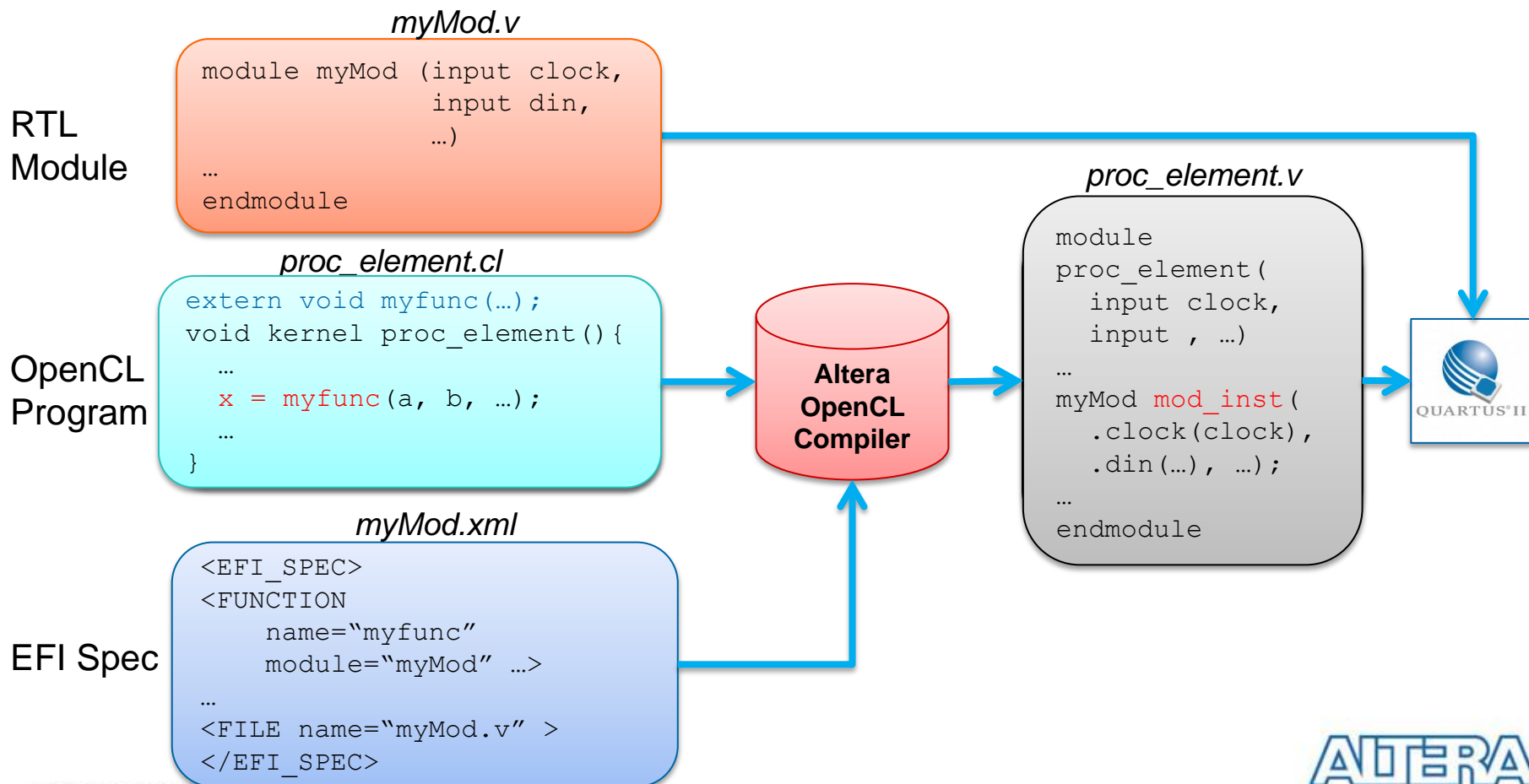


Interfaces to compiler

- Host CPU Interface: *Avalon Memory Map*
- Global Memory: *Avalon Memory Map*
- Option IO: *Avalon Streaming*

■ Ability to quickly integrate IPs in OpenCL

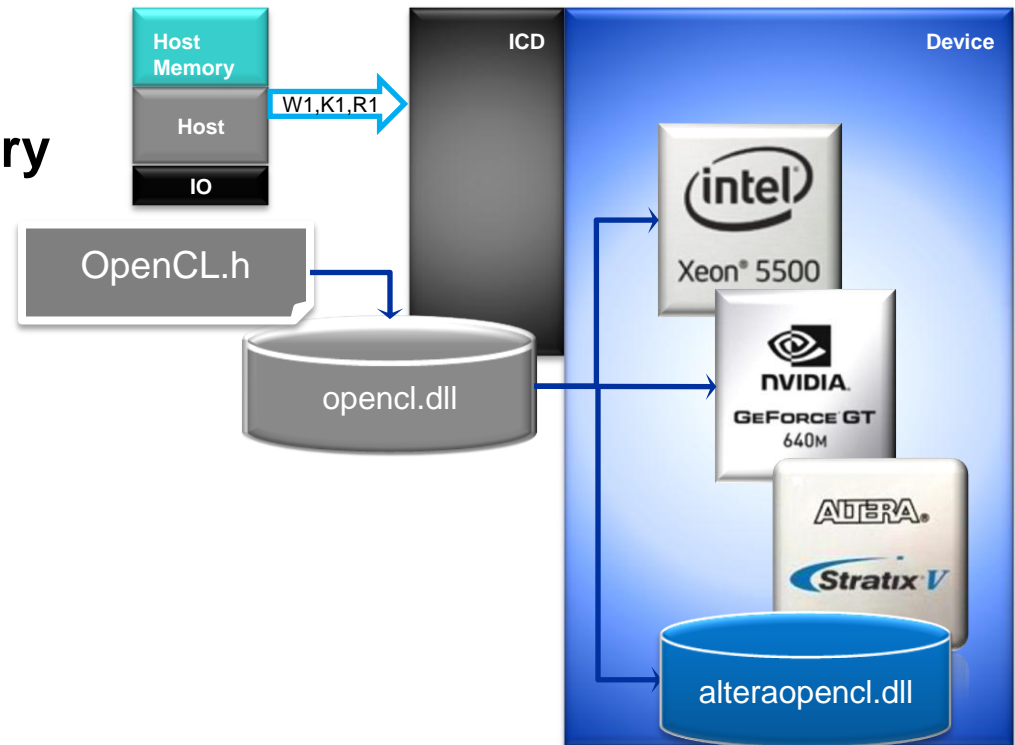
- Support Avalon Streaming based IPs in 14.0



- Different Platforms can co-exist on same host

- Host links against ICD library

- Runtime brings in vendor library



OpenCL + FPGA Key Benefits

- **Higher performance/watt vs. CPU/GPGPU**
 - Offload performance-intensive functions from the host processor to an FPGA
 - Implement exactly what you need
 - Pipeline parallel structures
 - Custom interconnect converging with data processing cores
- **Lower power vs. CPU/GPGPU**
 - Core frequency lower: 200-250MHz vs 1GHz
 - Turn off unused logic
 - Up to 1/5 the power
- **Faster development vs. traditional FPGA design flow**
 - Higher level of design abstraction
- **Higher department productivity**
 - Leverage your software development team
 - Familiar C-based design entry
- **Portability & Obsolescence free**
 - Code can transfer between different HW accelerators (CPU, GPGPU, FPGA, etc)
 - FPGA life cycle considerably longer than CPUs or GPGPUs



Product Information & Design Flow

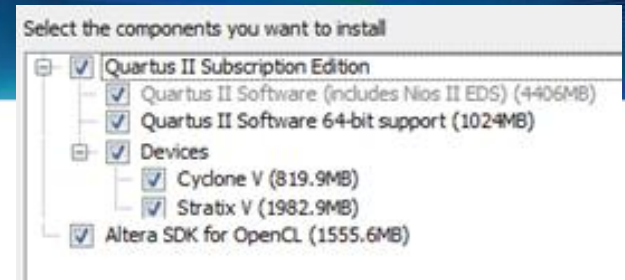


© 2013 Altera Corporation—Confidential



How to Get OpenCL

- Part of ACDS v13.1 installation



- Dedicated download page at <http://dl.altera.com/opencl/>
- Requires licensed Quartus software
- Supported on Windows and Linux
- Still need standard GCC compiler for host side code
 - Visual Studio, Eclipse...etc.

What is included with the Altera SDK for OpenCL?

- **Offline compiler (aoc)**
 - GCC based model
- **Altera OpenCL utility (aocl)**
 - Diagnostics for board installation
 - Flash or program FPGA image
 - Install board drivers (typically PCIe)
- **Host libraries**
 - Required by host code and provided by the vendor (Altera)
- **APB Board driver**
- **Design examples**
 - FFT, vectoradd, matrixmult, moving average

Altera SDK for OpenCL Product Details

■ Altera SDK for OpenCL Licensing

- Purchase a 1 year perpetual license
- Fixed & float available
- 60-day evaluation license available on request
- Requires Quartus II v13.1 Subscription Edition or Development Kit Edition



■ OS

-  – Microsoft 64-bit Windows 7
-  – Red Hat Enterprise 64-bit Linux (RHEL) 6.x

■ Memory requirements

- SDK: Computer equipped with at least 16 GB RAM
- Quartus II: Refer to memory requirements for target FPGA

Altera Preferred Board Partner Program for OpenCL

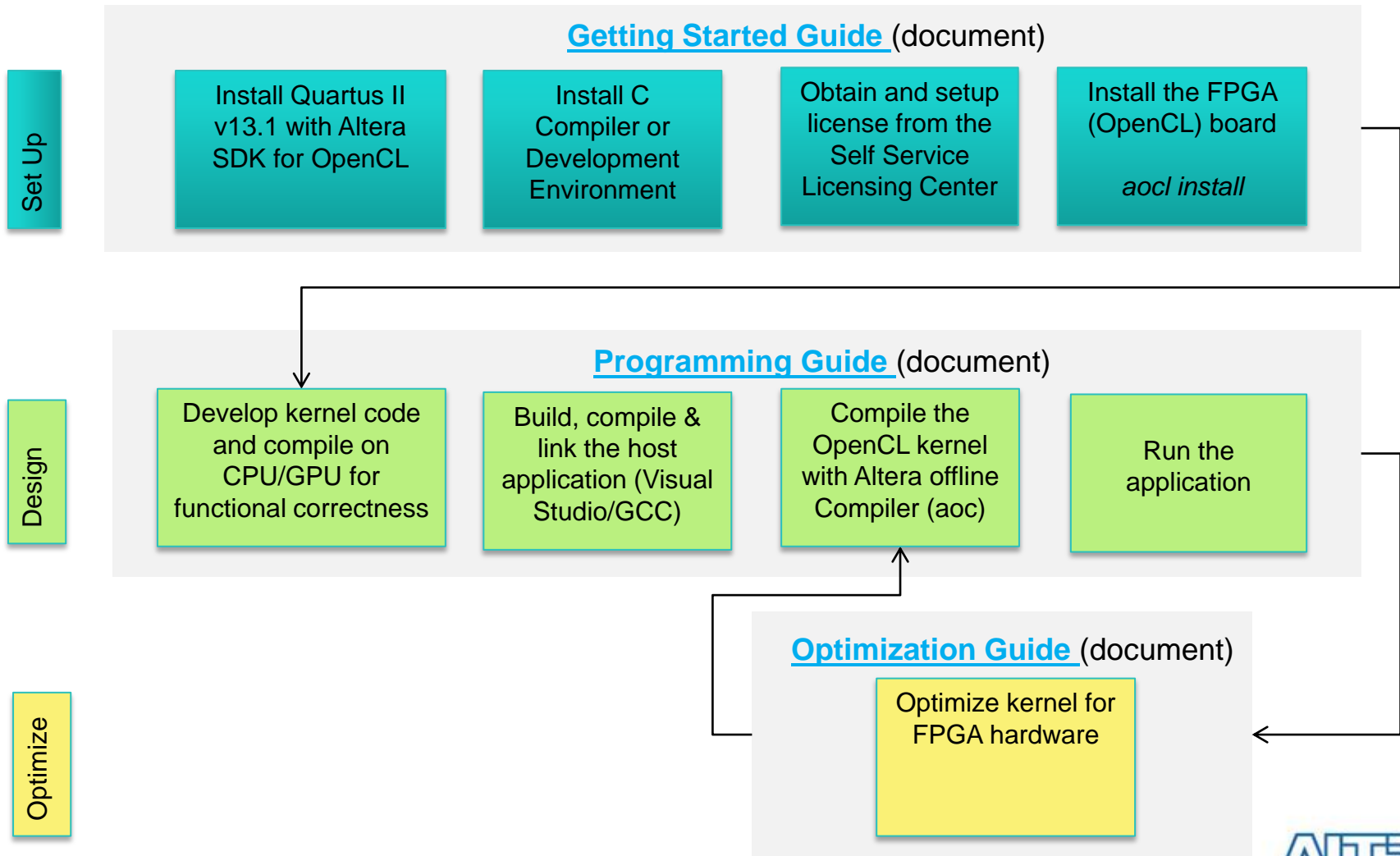
- **Provide customers with a portfolio of COTS boards to evaluate, develop and go-to-production with**
- **Customers can develop code and target that preferred board**
 - Altera SDK for OpenCL flow has been verified on the board
 - Ensures an exceptional out-of-box customer experience for the customer
- **Customer purchase directly from partners**
- **Altera's Preferred Board include:**
 - Includes Quartus II Development Kit Edition Software (one year license)
 - Includes an Altera SDK for OpenCL License (one year, perpetual license)



APBs Available as of 13.1 Release

Partner	Board	Altera Device	Where to Get?
Altera	DK-DEV-5CSXC6NES	Cyclone V SX SoC	Part of ACDS 13.1
BittWare	S5PH-Q	Stratix V D5	Part of ACDS 13.1
	S5PH-Q	Stratix V D8	Part of ACDS 13.1
	S5PH-Q	Stratix V A7	Contact BittWare
	S5PH-Q	Stratix V AB	Contact BittWare
	S5PH-DS	Dual Stratix V AB	Contact BittWare
	Terabox	8, S5PH-DS Boards	Contact BittWare
Nallatech	385-A7 Accelerator Card	Stratix V A7	Part of ACDS 13.1
	385-D5 Accelerator Card	Stratix V D5	Part of ACDS 13.1
PLDA	XP5S620LP-40G	Stratix V A7	Contact PLDA
Terasic	DE5	Stratix V A7	Contact Terasic

Altera SDK for OpenCL Design Flow





Applications



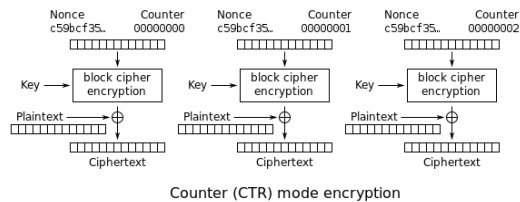
© 2013 Altera Corporation—Confidential



AES Encryption

■ Encryption/decryption

- 256bit key
- Counter (CTR) method

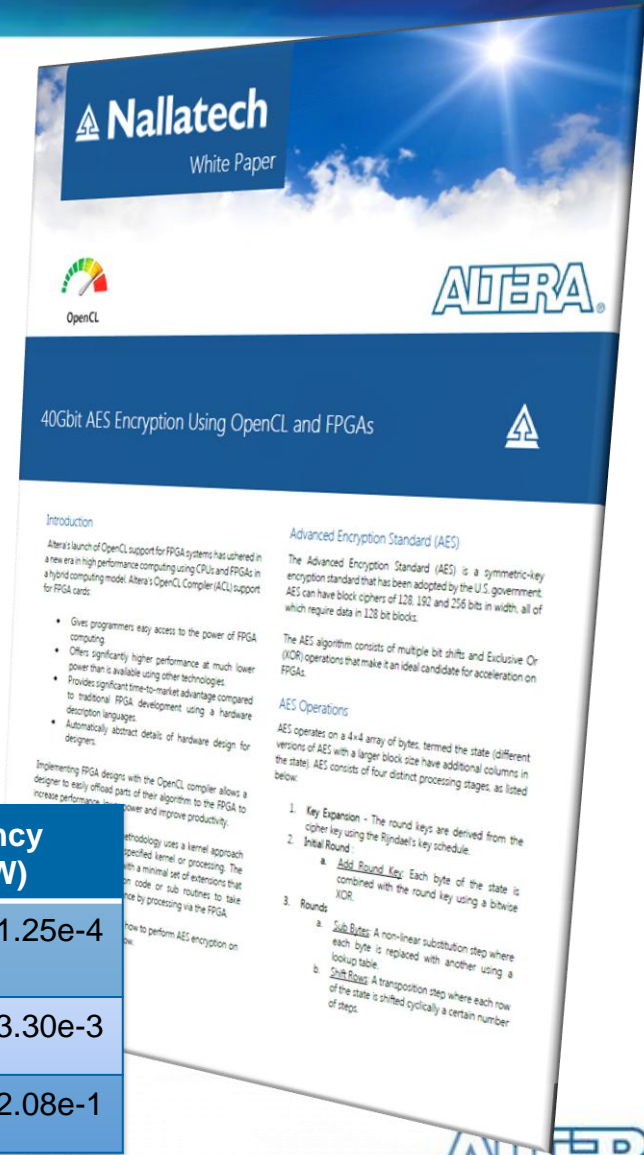


■ Advantage FPGA

- Integer arithmetic
- Coarse grain bit operations
- Complex decision making

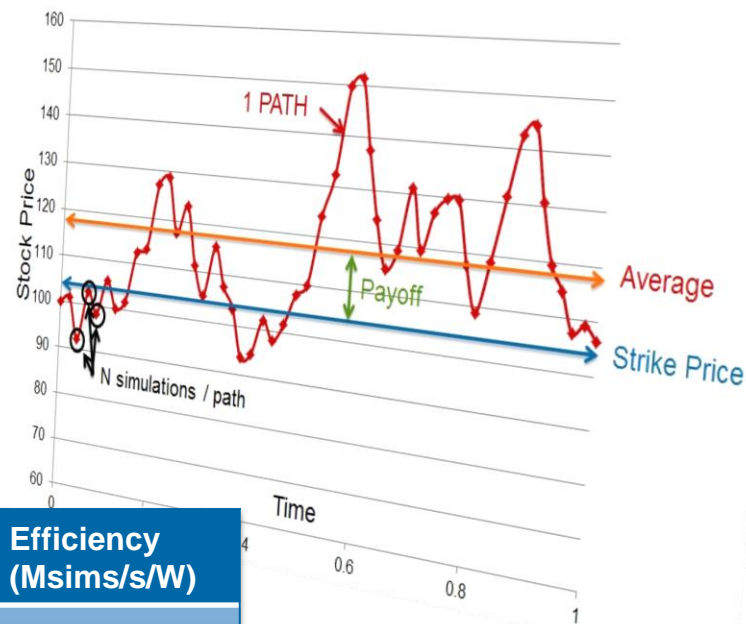
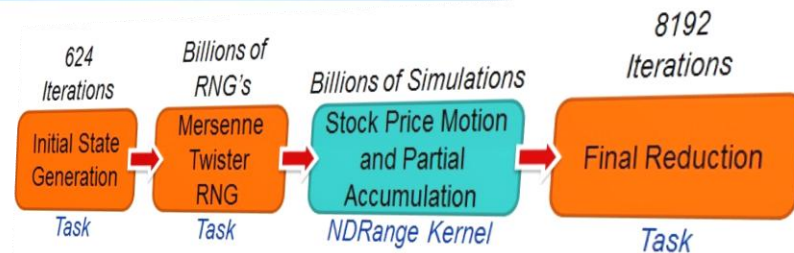
■ Results

Platform	Power (W)	Performance (GB/s)	Efficiency (GB/s/W)
E5503 Xeon Processor (single core)	est 80	0.01	1.25e-4
AMD Radeon HD 7970	est 100	0.33	3.30e-3
PCIe385 A7 Accelerator	25	5.20	2.08e-1



Multi-Asset Barrier Option Pricing

- **Monte-Carlo simulation**
 - No closed form solution possible
 - High quality random number generator required
 - Billions of simulations required
- **Used GPU vendors example code**
- **Advantage FPGA**
 - Complex Control Flow
- **Optimizations**
 - Channels, loop pipelining
- **Results**

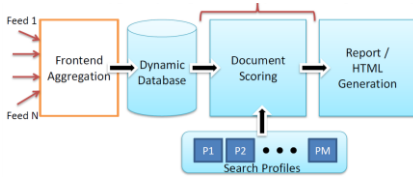


Platform	Power (W)	Performance (Bsims/s)	Efficiency (Msims/s/W)
W3690 Xeon Processor	130	.032	0.0025
nVidia Kepler20	212	10.1	48
Bittware S5-PCIe-HQ	45	12.0	266

Document Filtering

■ Unstructured data analytics

- Bloom Filter



■ Advantage FPGA

- Integer Arithmetic
- Flexible Memory Configuration

■ Results

Platform	Power (W)	Performance (MTs)	Efficiency (MTs/W)
W3690 Xeon Processor	130	2070	15.92
nVidia Tesla C2075	215	3240	15.07
PCIe385 A7 Accelerator	25	3602	144.08

loads from arrays A and B are converted into *load units* which are small circuits responsible for issuing addresses to external memory and processing the returned data. The two returned values are fed directly into an adder unit responsible for calculating the floating point addition of these two values. Finally, the result of the adder is wired directly to a *store unit* that writes the sum back to external memory.

The most important concept behind the OpenCL-to-FPGA compiler is the notion of pipeline parallelism. For simplicity, assume the compiler has created 3 pipeline stages for the kernel as shown in Figure 3. On the first clock cycle, thread

Fig. 3. Example of Pipeline Parallelism

local memory accesses are connected through a specialized interconnect structure to onchip M9K RAMs.

3. DOCUMENT FILTERING

In this section, we explore how OpenCL can be used to express a document filtering algorithm. Document filtering involves looking at an incoming stream of documents and finding the ones that best match a user's interest. An example of such a system would be the use of a filtering mechanism which monitors news feeds and sends relevant articles to a user's email account. In general, this application is an example of performing analytics on unstructured data such as text files, HTML pages, emails, and video files. It has been estimated that up to 80% of all relevant data for businesses is in unstructured form [12].

In [13, 14, 15], it was illustrated that FPGAs can offer compelling advantages for document filtering in terms of performance and power. We briefly review the basics here. The algorithm, illustrated graphically in Figure 5, attempts to find the best matching documents for a specific *search profile*. The search profile is the filter which matches docu-

0 is clocked into the two load units. This indicates that they should begin fetching the first elements of data from arrays A and B. On the second clock cycle thread 1 is clocked in at the same time thread 0 has completed its read from memory and stored the results in the registers following the load units. On cycle 3, thread 2 is clocked in, thread 1 captures its returned data, and thread 0 stores the sum of the two values that it loaded. It is evident that in the steady-state, all parts of the pipeline are active, with each stage processing a different thread.

Figure 4 shows a high level representation of a complete OpenCL system containing multiple kernel pipelines and circuitry connecting these pipelines to off-chip data interfaces. In addition to the kernel pipeline, ACL creates in-

Fig. 5. Unstructured Search Algorithm

ments against the user's topic of interest. To this end, each document is reduced to a set of words, and the frequency of appearance of each word in the document. Each term-frequency pair (t_i, f_i) in the document is represented as a 32-bit integer, with a 24-bit term ID, and a 8-bit frequency of occurrence. Terms are generally words in the document. The search profile consists of a smaller set of terms, and a search profile. The weights specify its relative importance in search. A score is computed for every document to determine its relevance to the given profile.

$$Score(Doc) = \sum_{t_i \in Doc} f_i * w(t_i) \quad (1)$$


Consumer (Japan)

■ Image Processing

- Adaptive weighted images

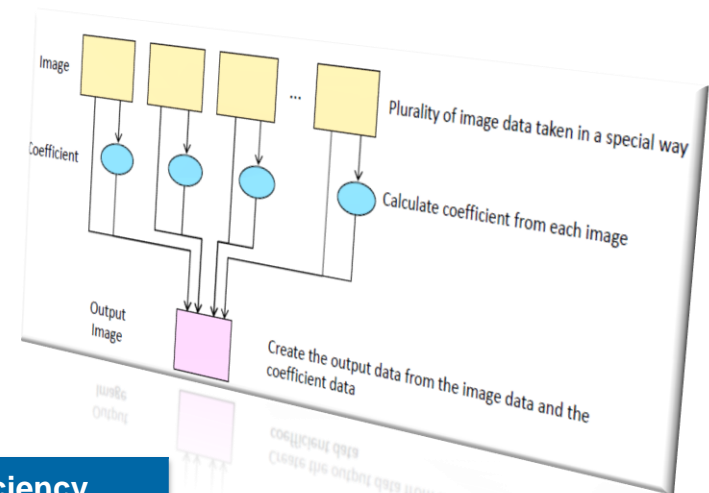
$$p_{xy} = \frac{c_{1ij} d_{1xy} + c_{1(i+1)j} d_{2xy} + c_{2ij} d_{xy} + c_{2(i+1)j} d_{2xy}}{W}$$

■ Advantage FPGA

- Integer Arithmetic

■ Results

Platform	Power (W)	Performance (FPS)	Efficiency (FPS/W)
W3565 Xeon Processor	est 130	0.05	.0004
nVidia Quadro 4000	est 150	2.94	.0200
PCIe385 A7 Accelerator	21	4.29	.2040



Smith-Waterman

■ Sequence Alignment

– Scoring Matrix

$$H(i, 0) = 0, 0 \leq i \leq m$$

$$H(0, j) = 0, 0 \leq j \leq n$$

if $a_i = b_j$ then $w(a_i, b_j) = w(\text{match})$ or if $a_i \neq b_j$ then $w(a_i, b_j) = w(\text{mismatch})$

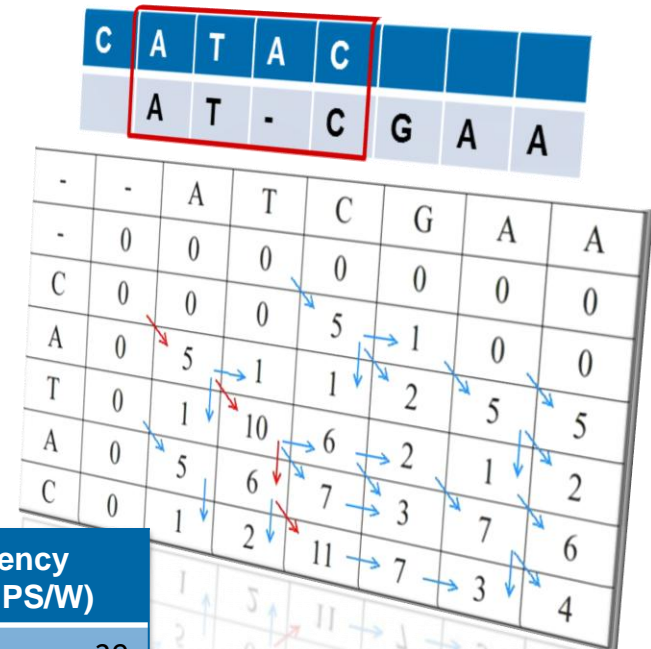
$$H(i, j) = \max \begin{cases} 0 \\ H(i-1, j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1, j) + w(a_i, -) & \text{Deletion} \\ H(i, j-1) + w(-, b_j) & \text{Insertion} \end{cases}, 1 \leq i \leq m, 1 \leq j \leq n$$

■ Advantage FPGA

- Integer Arithmetic
- SMT Streaming

■ Results

Platform	Power (W)	Performance (MCUPS)	Efficiency (MCUPS/W)
W3565 Xeon Processor	140	40	.29
nVidia K20	225	704	3.13
PCIe385 A7 Accelerator	25	32596	1303.00



Multi Function Printer

■ Image Processing

- RGB output of raster scanner converted to CMYK colorants for printing

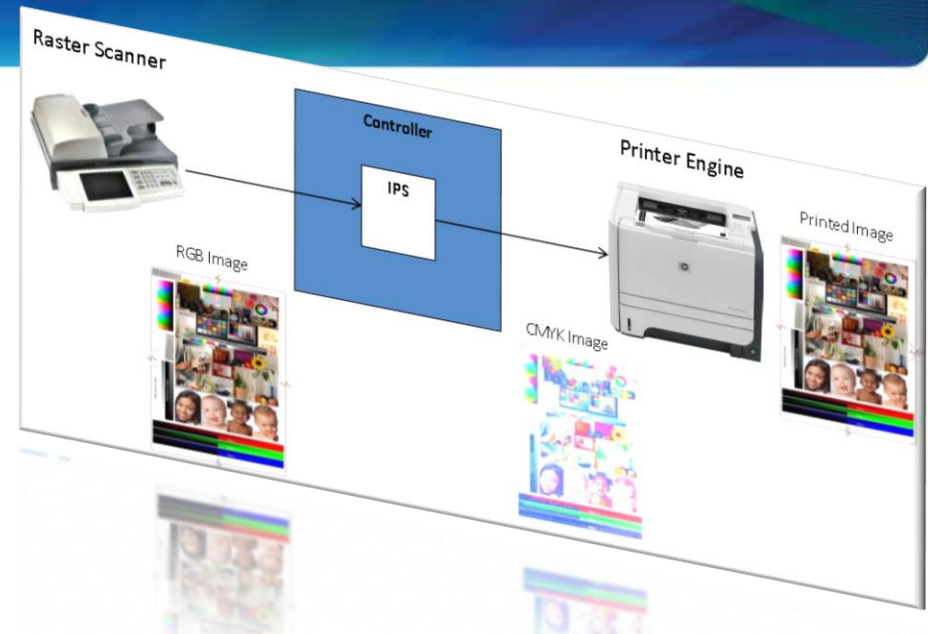
■ Advantage FPGA

- SoC Solution
- IO and Kernel Channels
- Heterogeneous memory accesses

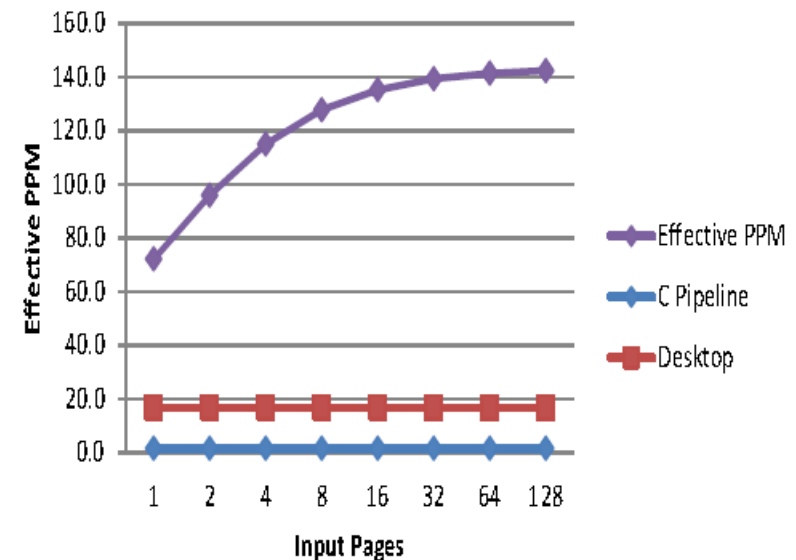
■ Goal 50PPM at A4/letter size

■ Results

- >40X improvement over C based algorithm on ARM only
 - No NEON coprocessor used
- C6 speed grade part improved 20% to 128PPM



Effective PPM vs # of Input Pages



RayTracing Demo using OpenCL on an SoC

October 21, 2013

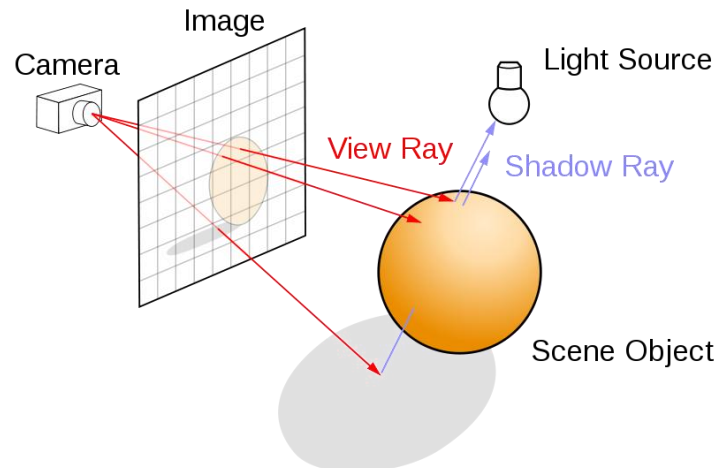


© 2013 Altera Corporation—Confidential

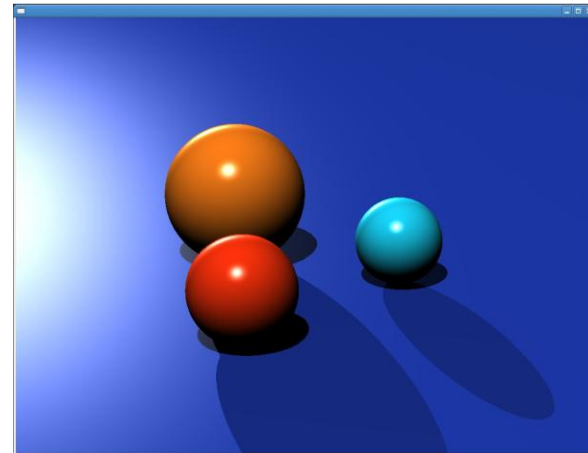


Raytracing Graphics Overview

- **Very high degree of visual realism but high computational cost**
 - Non-realtime rendering (e.g. 3D animation movies)
- **Generates an image by tracing the path of light through pixels in an image plane**
- **Simulates the effects of the rays' encounters with virtual objects**



Source: www.wikipedia.com



Setup

Ethernet Cable (For Video Output)

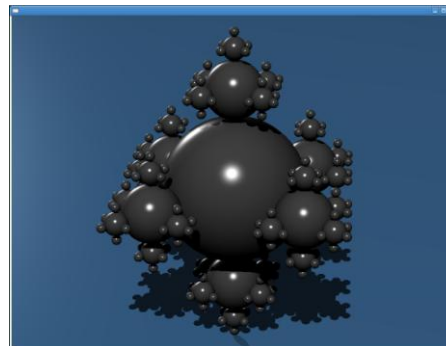
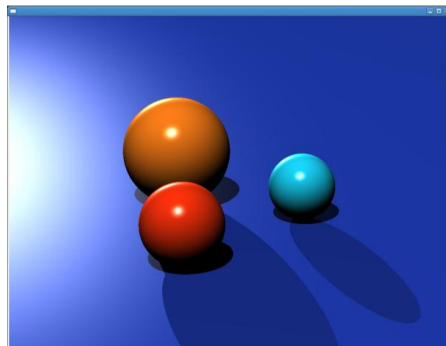


USB to MicroUSB
Cable
(For Serial Terminal)



Cyclone V SoC Development Board

Raytracer demo (using OpenCL on SoC)



Tasks	simple scene			complex scene		
	ARM	FPGA	Speedup	ARM	FPGA	Speedup
init. primary rays	378 ms	X		1063 ms	X	
calc. intersections	1326 ms	X		54914 ms	X	
shadow ray calc. (2 lights)	1727 ms	X		99891 ms	X	
shading	1072 ms	X		3198 ms	X	
TOTAL	4503 ms	95 ms	47x	159066 ms	4303 ms	37x

*all values for 1024x768 resolution

Why SoC and OpenCL

- **Host and accelerator in a single device**
 - Smaller boards
 - Lower board cost
- **Offload ARM with hardware acceleration in the FPGA fabric**
- **OpenCL allows you to stay in the software programming model without having to learn a Hardware Description Language (HDL)**



Thank you



© 2013 Altera Corporation—Confidential

