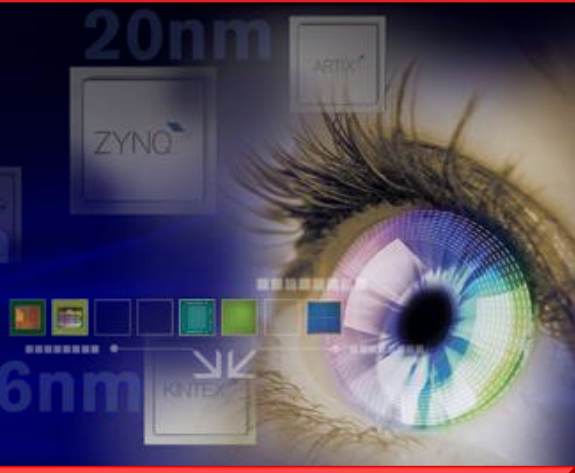
HIGH PERFORMANCE
SIGNAL PROCESSING 2014

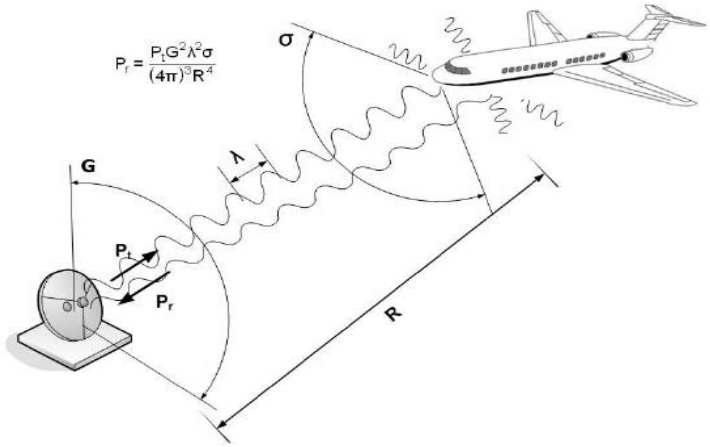
XILINX
ALL PROGRAMMABLE™

DSP design on Xilinx FPGA with High Level Synthesis C/C++ compiler



Daniele Bagni

DSP Specialist for EMEA



Outlines

- about Xilinx
- Xilinx DSP design flows
- Vivado High Level Synthesis (HLS)
- Linear Algebra with HLS: QRD for Adaptive Beamforming
- Conclusion
- References

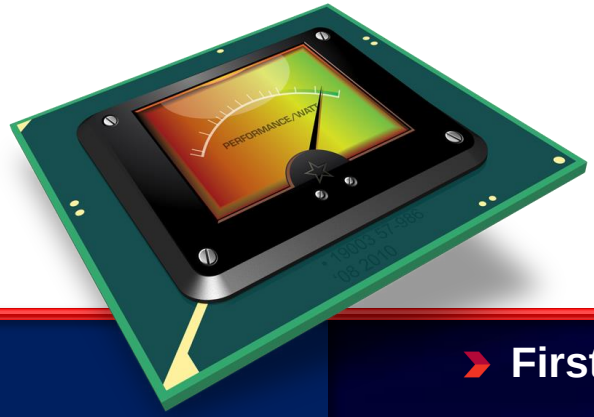
Outlines

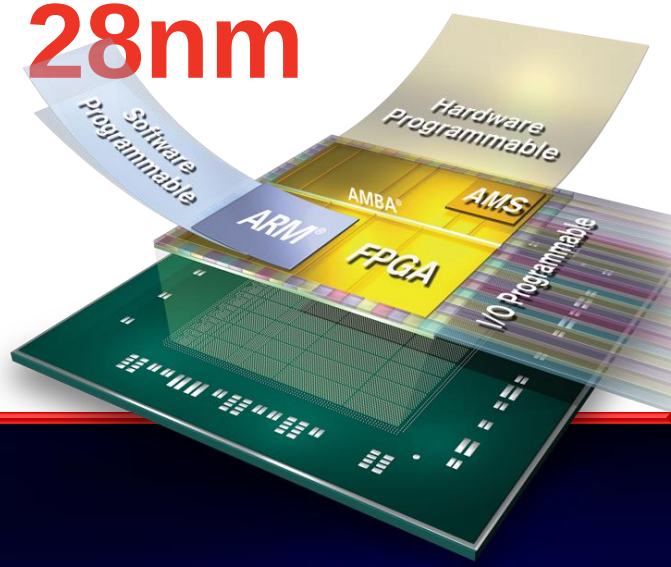
- about Xilinx
- Xilinx DSP design flows
- Vivado High Level Synthesis (HLS)
- Linear Algebra with HLS: QRD for Adaptive Beamforming
- Conclusion
- References

Semiconductor Market Leader in All Programmable Devices

- Founded 1984
- $2.17B FY13 revenue
- ~50% market segment share
- 3,000 employees worldwide
- 20,000 customers worldwide
- 3,500 patents
- 60 industry firsts

A Generation Ahead at 28nm

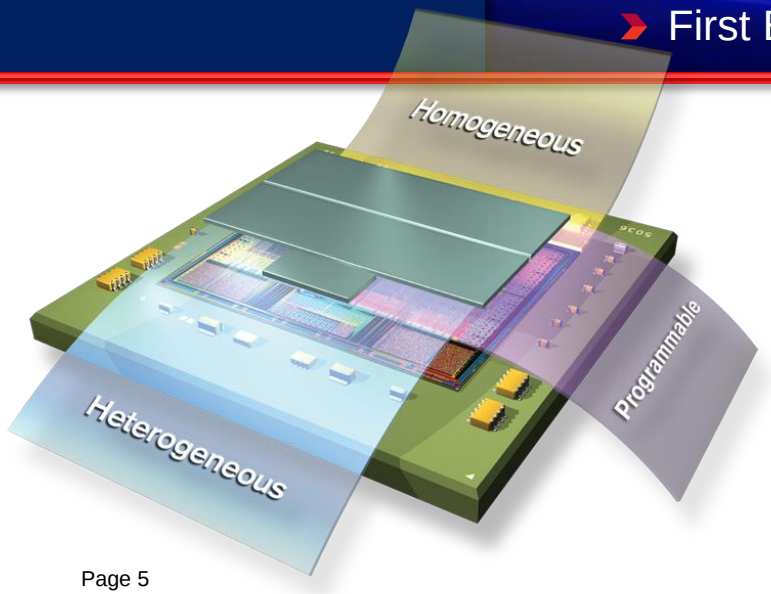




28nm

- **First** 28nm tape-out
- **First** All Programmable SoC
- **First** All Programmable 3D IC
- **First** SoC-strength design suite
- First Errata-Free Production Release

70%
Market Share



XILINX ALL PROGRAMMABLE

Delivering the Industry Firsts

First to 28nm

Mar 2011

First to SoC

Dec 2011

First to 4M LCs

Dec 2013

First to 20nm

June 2013

First ASIC-Class Architecture

Nov 2013

First to Ship

1 month for first shipment,
1+ yr & counting for quality

1+ years for first shipment,
2+ yrs and counting for production quality

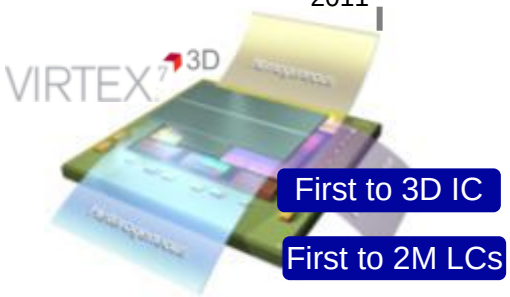
4+ mos. & counting

2011 | 2012 | 2013

2+ years, 2 generations & counting

1.5+ years & counting

Oct 2011



First to 3D IC

First to 2M LCs

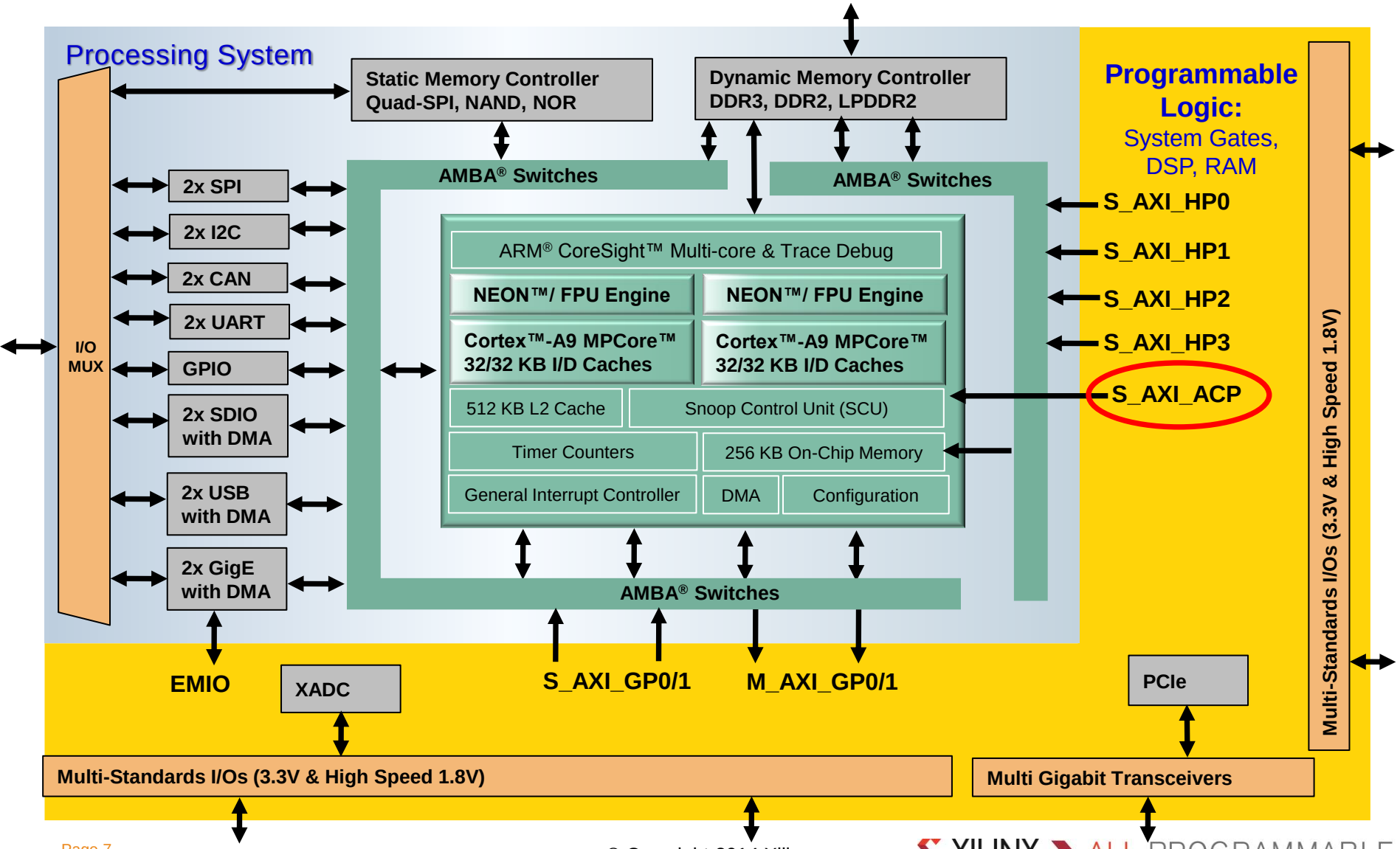
Jul 2012

First ASIC / SoC Class Tools

First Design Methodology

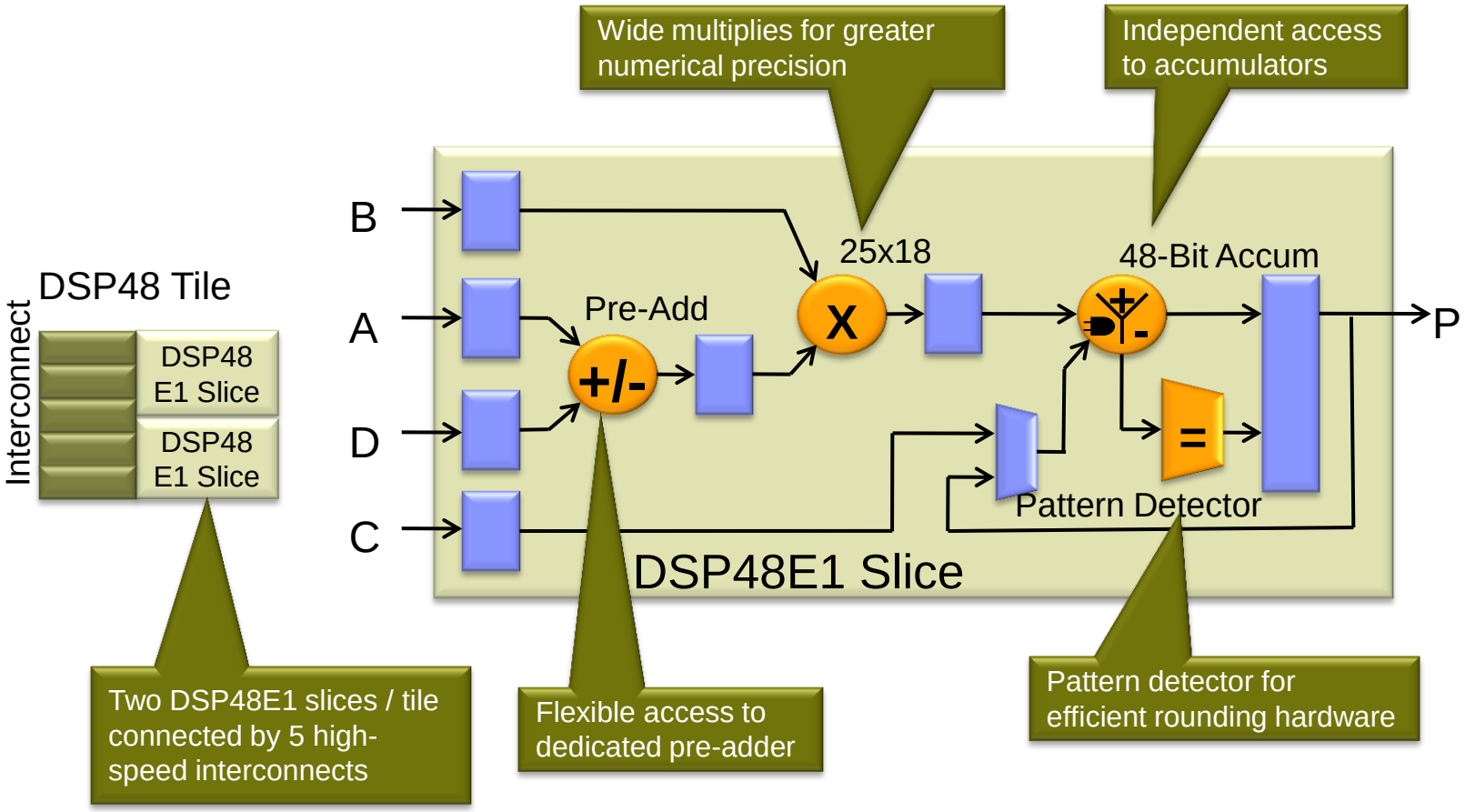
Lead on Competitor

Zynq-7000 AP SoC block diagram



Industry's most Advanced DSP Slice

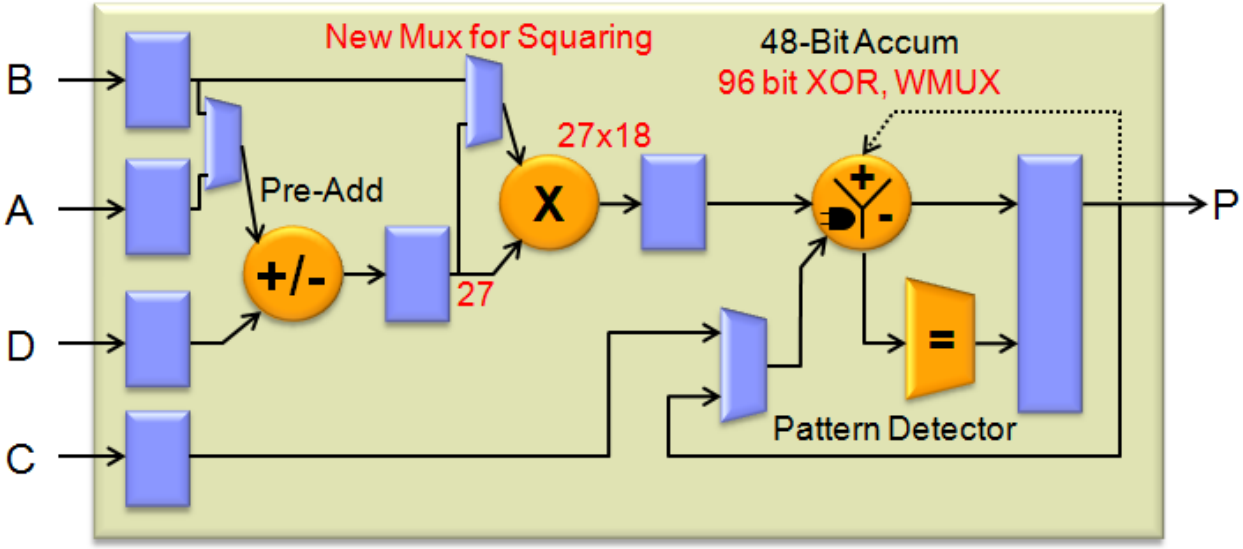
Artex-7, Kintex-7, Virtex-7, Zynq-7000



Resources per Family	Artix	Kintex	Virtex	Zynq
Max DSP48E1 Fmax	628 MHz	741 MHz	741 MHz	741 MHz
Max DSP48E1 Count	740	1920	3600	2020

UltraScale Architecture (20nm silicon node) DSP Slice Features

- **The UltraScale DSP slice is a superset of 7 Series DSP48E2**
  - 27x18 multiplier and 27-bit pre-adder
  - Flexible pipeline
  - Cascade in and out
  - Carry in and out
  - 96-bit MACC
  - SIMD support
  - 48-bit ALU
  - Pattern detect
  - 17-bit shifter
  - Dynamic operation (cycle by cycle)
  - Squaring: $(A+/-D)^2$, Wide XOR (up to 96 bit) and WMUX product feedback

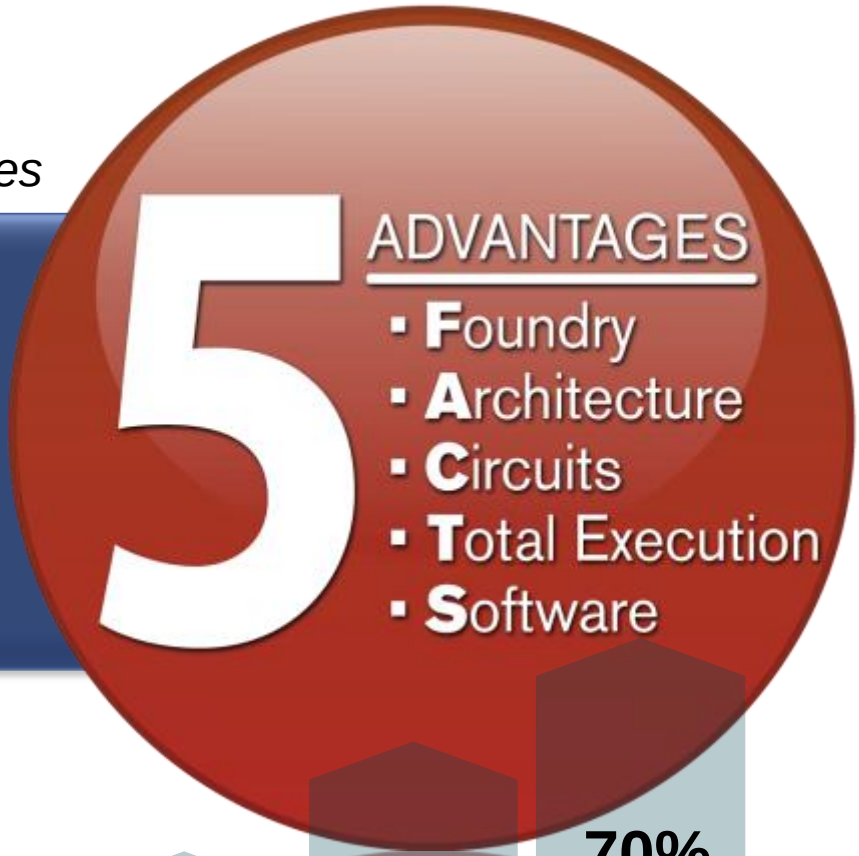


Over 8 TMACs Available in KU115! 1.3 TFLOPS!

FACTS Versus Promises

Proven Formula
Repeated Across Nodes

28nm
20nm
16nm



XILINX
ALL PROGRAMMABLE

50% FY12

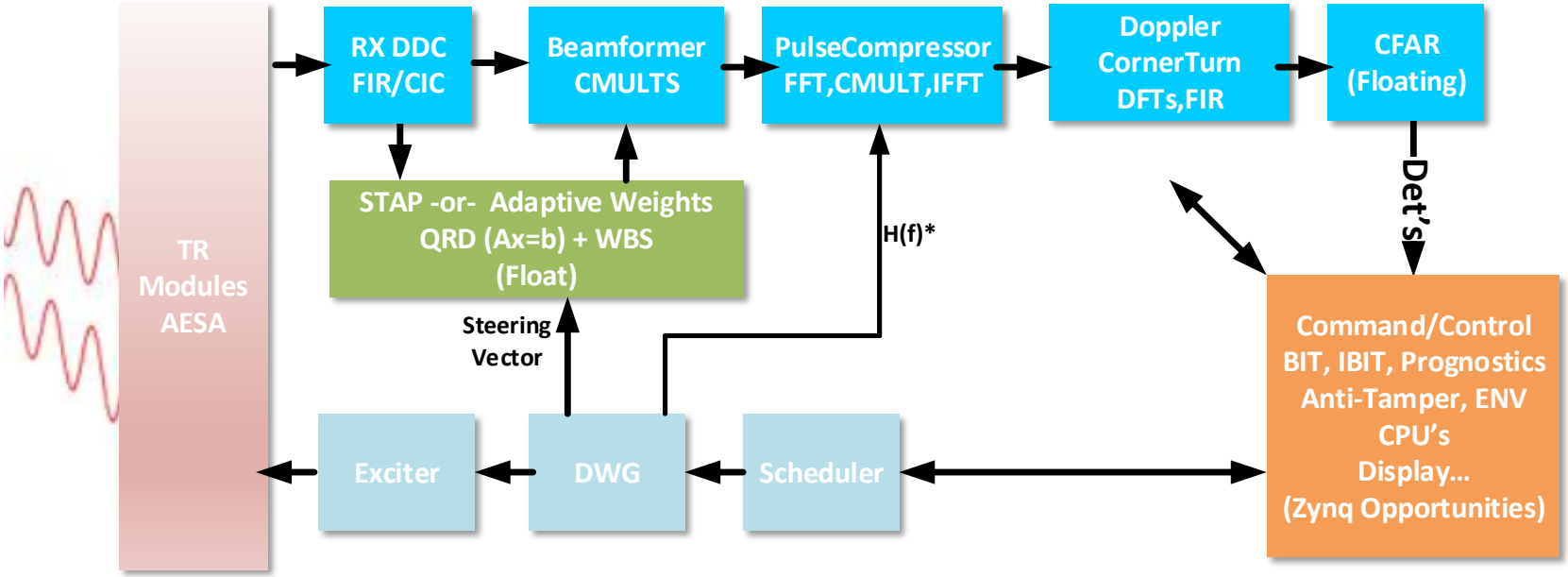
60% FY13

~70% FY14

1Q-3Q FY14 numbers from public competition and Xilinx financial data

De facto 28nm Segment Share

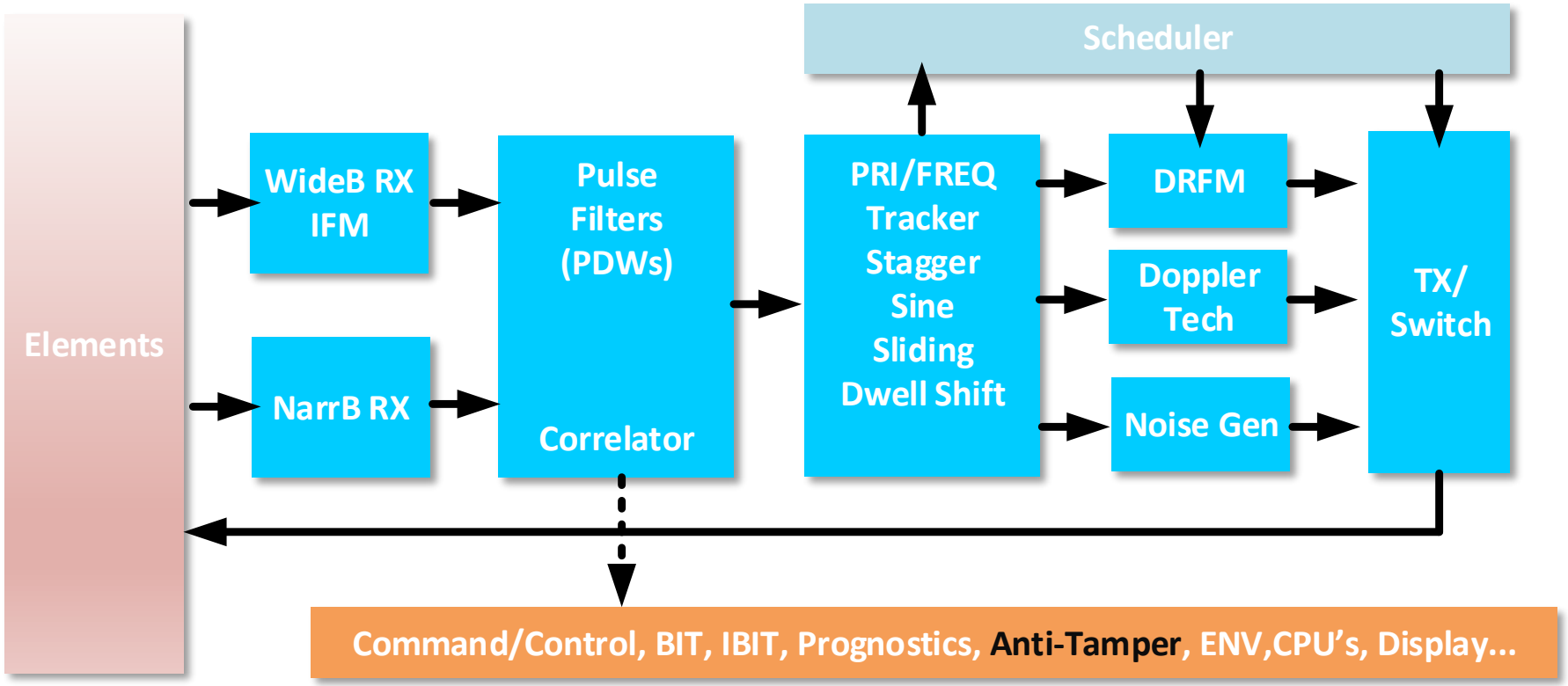
Xilinx in all aspects of RADAR



- **Xilinx FPGAs Ensure Mission Success for Today and the Future**
  - Xilinx's Virtex/Kintex7, Zynq and UltraScale Ensure World Class Functionality
  - Leader in DSP, GT, RAM Density and Fast Path into Silicon
  - Floating Point allows QRD, CFAR Solutions, reduce Power & BOM
  - Reprogrammable and IO Agnostic, PCIe, SRIO, GbE, Fiber Channel...

Xilinx FPGAs Allow Upgrades & Capability without overhauling the System

Xilinx in all aspects of Electronic Warfare



- EW Systems push limits of FPGAs, High ADC/DAC Rates
- Push limits on DSPs (FIRs, Channelization…) and Clock Frequencies
- Latency is **Critical → High End/Performance FPGAs**

Xilinx enables the World's State of the ART EW Systems Today!

Key Technologies Enabling EW/RADAR

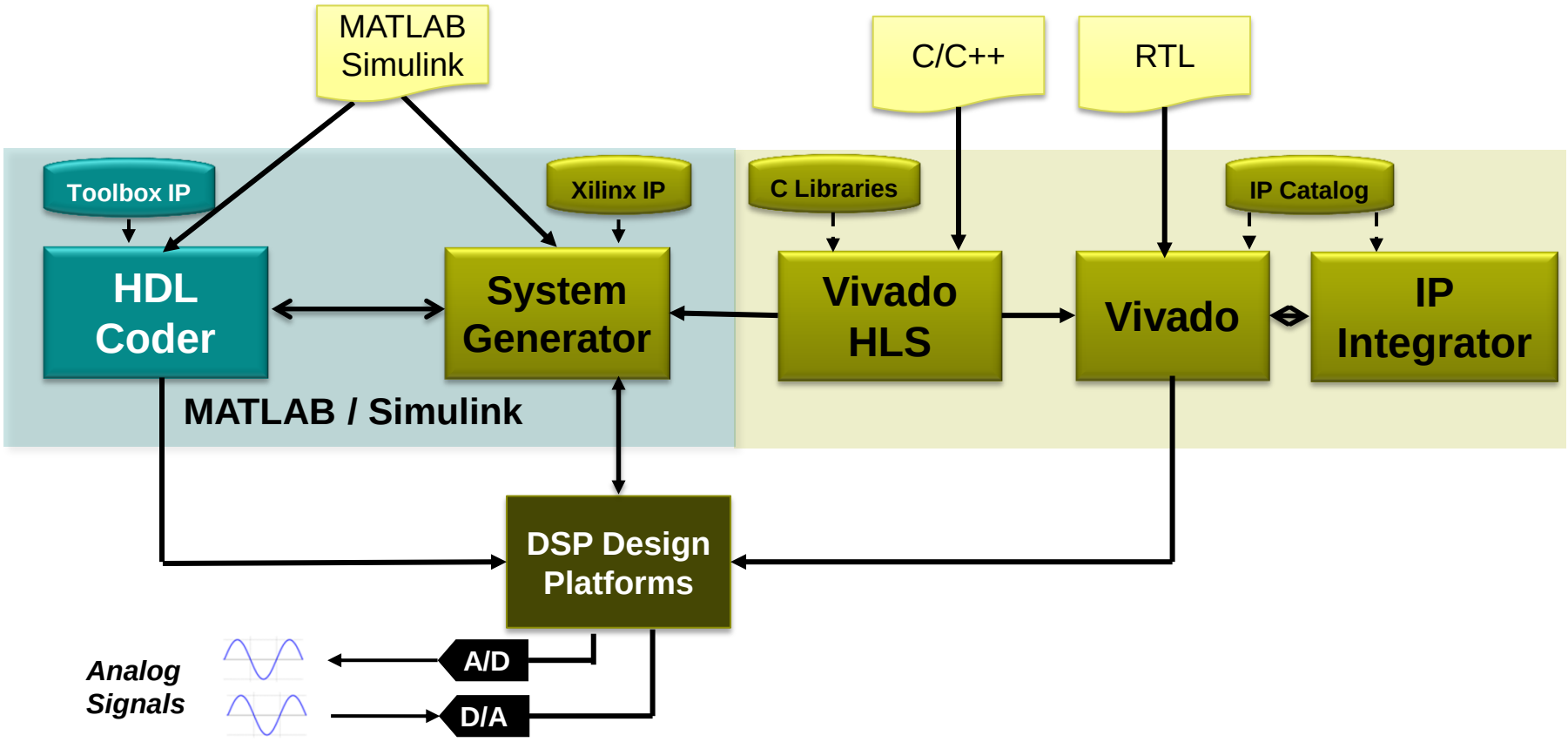
- **Xilinx JESD204b IP ADC/DAC (Serial Interfaces are the future…)**
  - Allows Dense Digital RX Designs, Significant lower power, migration/upgrades
  - Enabler for element level processing, TR Modules + ADC
  - Xilinx is the GT Leader
- **Xilinx Hybrid Memory Cube IP**
  - Same story, Uses GT lanes to reduce many DDR Pairs
  - 160GB/s Bandwidth, Lower Power once again
- **Xilinx DSP's are the Widest and Fastest in the Industry**
  - Fmax over 740 MHz, or Massively Parallel functions
  - Wide Bit widths allow efficient **Floating Point** Operations
- **Embedded ARM Processors – Zynq Family**
  - Remove the need of Centralized CPU processors
  - C/C++ libraries that can be Migrated and Maintained
- **Xilinx Vivado High Level Synthesis**
  - C/C++ FPGA Design Entry Allows The Worlds Fastest FPGA Exploration, Key for RFP
  - Removes Burden of Time Consuming RTL Simulation
- **Analytical Place and Route**
  - Xilinx FPGAs no longer the bottle neck in System Development and Integration
  - Many FPGA Bit Images in a day can be verified on a system
- **Xilinx Commitment to Secure Designs**
  - Anti-Tamper Solutions
  - Secure Monitor IP

Outlines

- about Xilinx
- **Xilinx DSP design flows**
- Vivado High Level Synthesis (HLS)
- Linear Algebra with HLS: QRD for Adaptive Beamforming
- Conclusion
- References

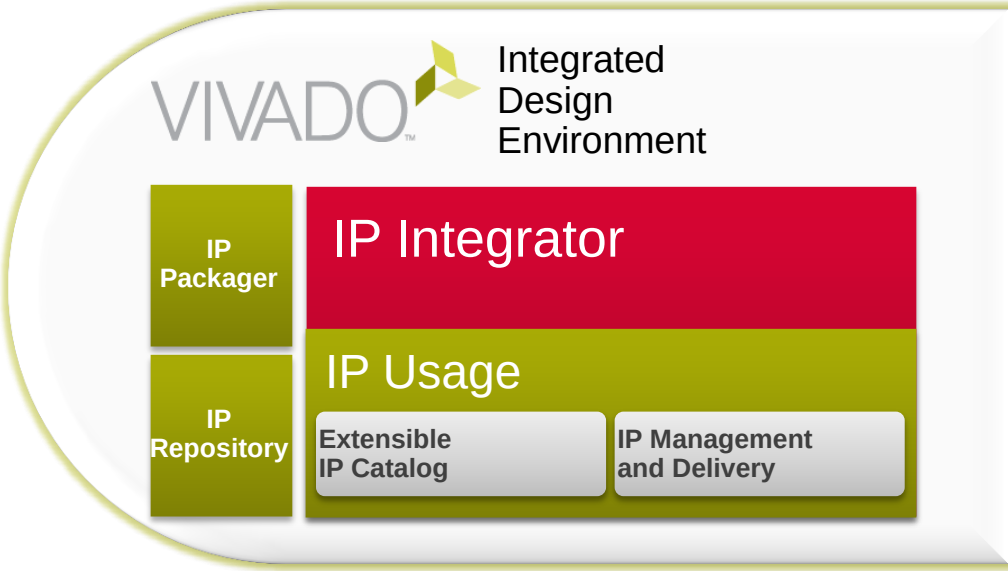
Xilinx DSP Design Flows

- Flexible design environment
- Floating-point and Fixed-point Hardware Generation
- Real time analog data acquisition
- World class C design flow

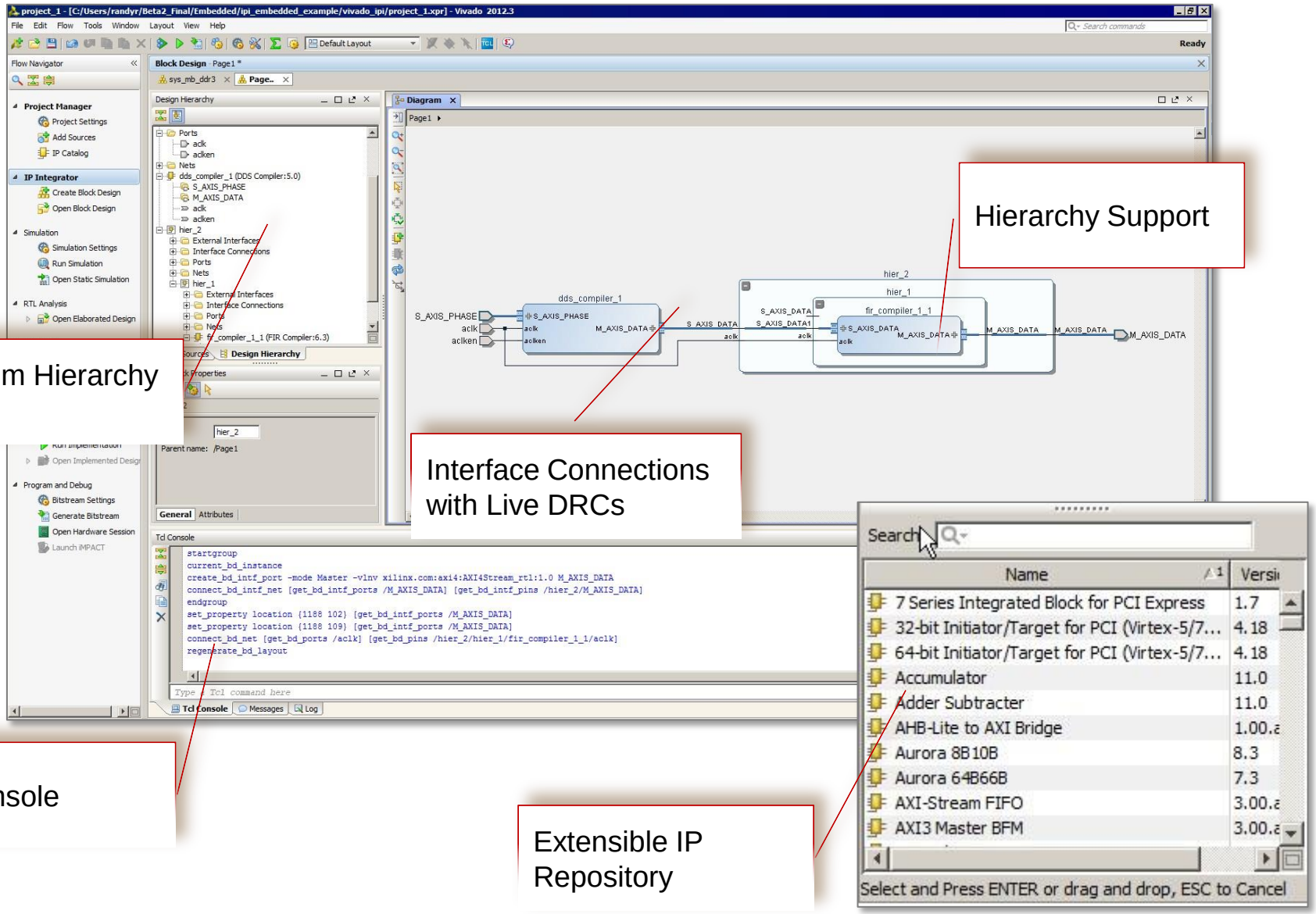


Vivado IP Integrator

- **A hierarchical IP integration tool for processor based and non-processor based systems**
- **A graphical and scriptable IP configuration and connection environment**
- **Integration and Reuse of IP**
  - Rapid creation of complex IP by packaging the contents of a diagram
- **Automatic Generation of RTL**
  - Instantiates all IP in a diagram and makes all the interconnections



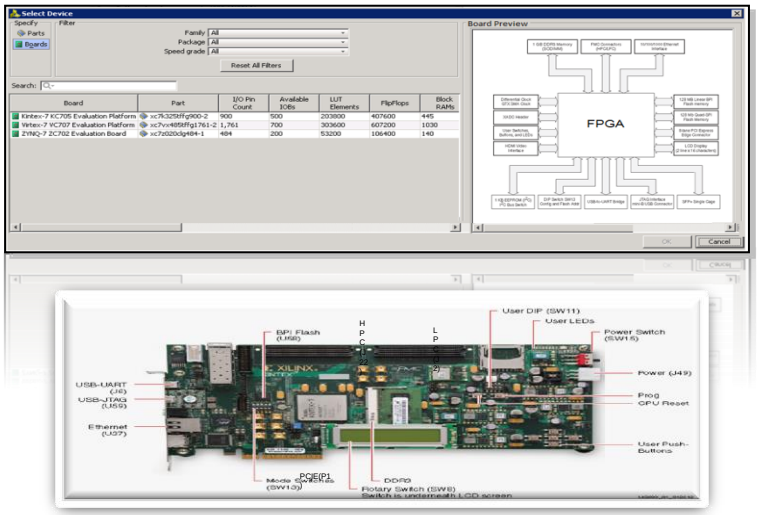
Vivado IP Integrator User Interface



Vivado IP Integrator: Platform Aware

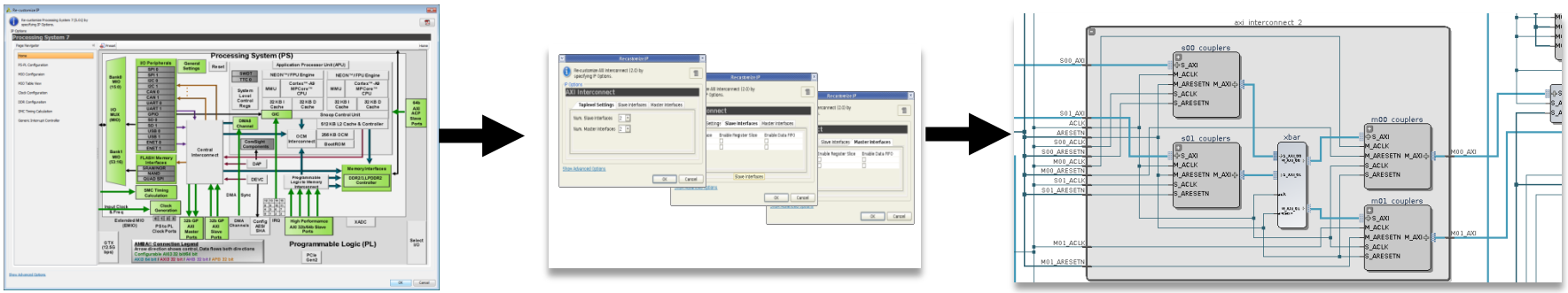
Co-Optimized for platforms

- Target platform aware
- Supports All Programmable Zynq and 7 series kits

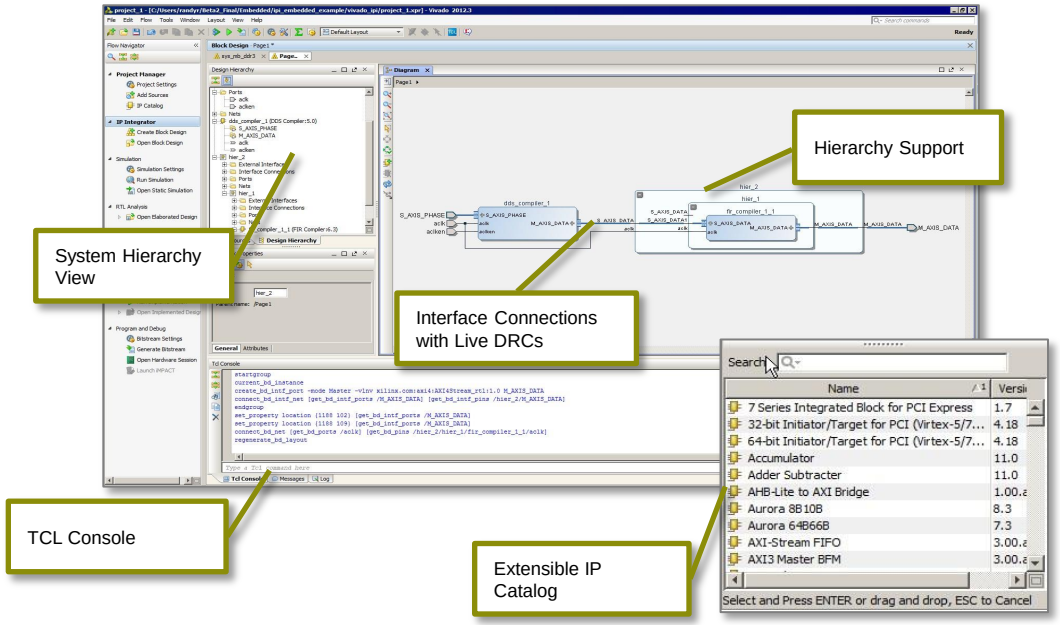


Co-Optimized for silicon

- IP aware automated AXI Interconnects for maximum performance or area
- Automated interface, device driver & address map generation for Zynq and MicroBlaze

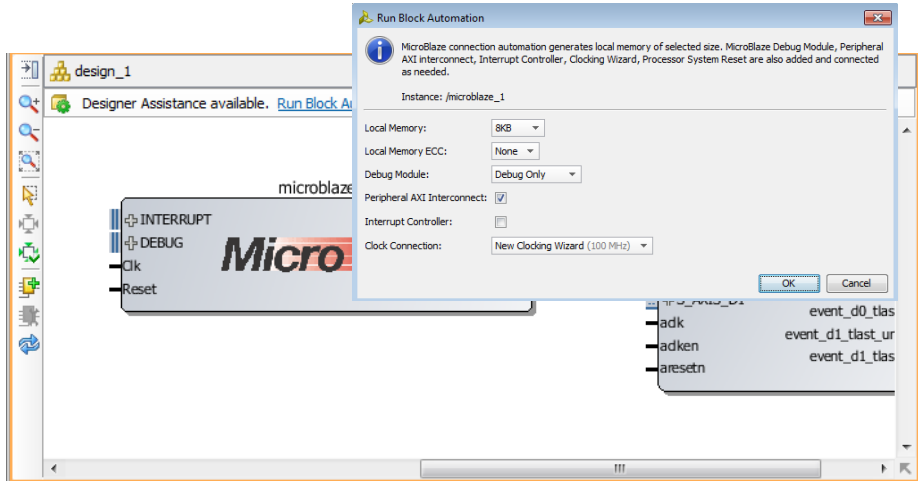


Vivado IP Integrator: Intelligent IP Integration



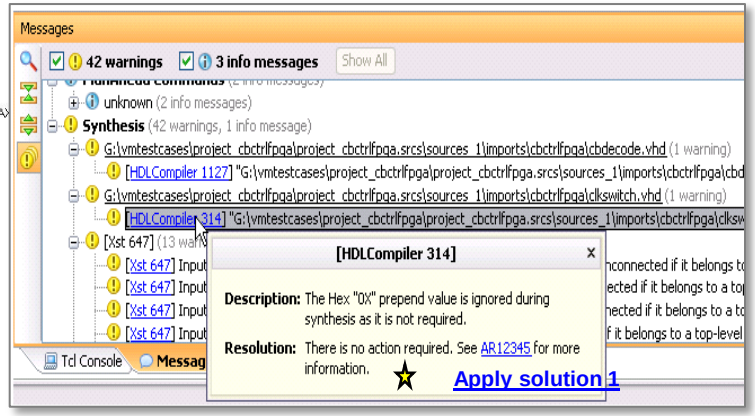
- **Correct-by-construction**
  - Extensible IP repository
  - Real-time DRCs and parameter propagation/resolution
  - Designer Assistance

- **Automated IP Subsystems**
  - Block automation for rapid design creation
  - One click IP customization

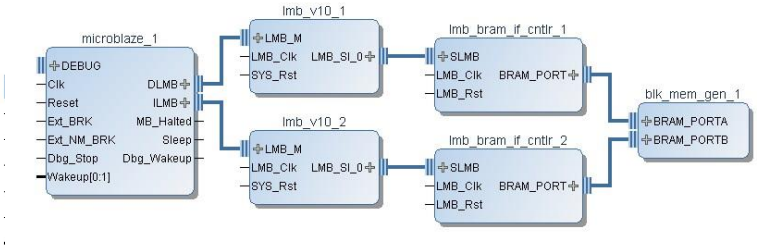


Intelligent IP Integration:
Correct by Construction Design

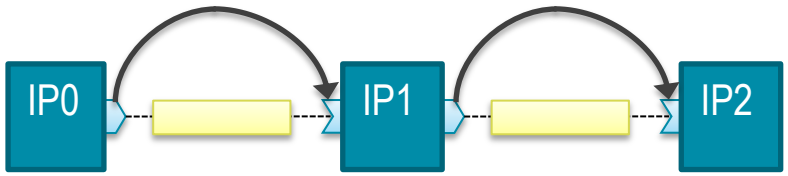
Real-time DRCs



Auto Connection



Parameter Propagation

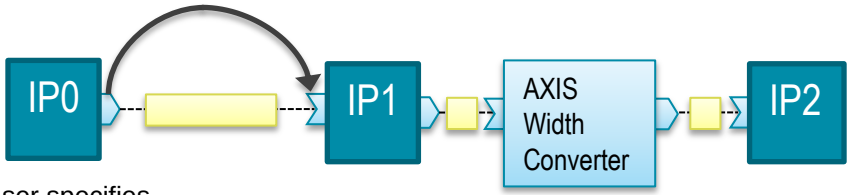


User specifies width=64

Default width=32
Is updated to propagated value

Input width=64 after propagation

Parameter Resolution



User specifies width=64

IP2 only supports width=32

XAPP1170

Application Note: Vivado HLS

XILINX®

A Zynq Accelerator for Floating-Point Matrix Multiplication Designed with Vivado HLS

Author: Daniele Bagni, Juanjo Noguera, Fernando Martinez Vallina

Summary

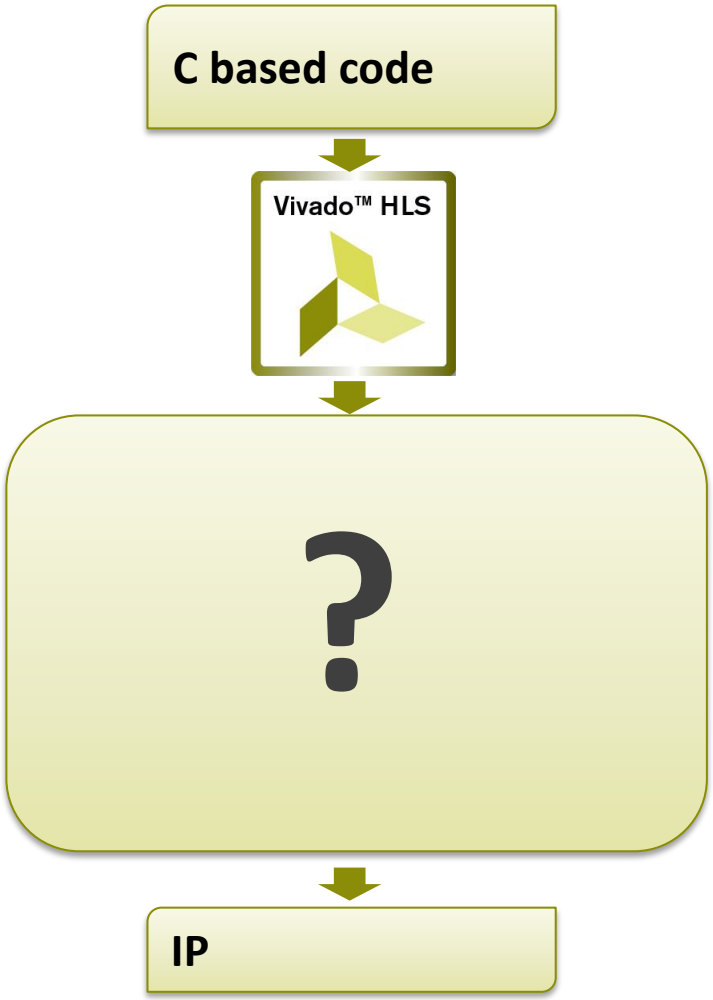
This application note describes how using Vivado HLS (High Level Synthesis) to develop a floating-point matrix multiplication accelerator interfaced via AXI4-Streaming to the Accelerator Coherency Port (ACP) of the ARM CPU in the Zynq-7000 All Programmable SOC device. The floating-point matrix multiplication modeled in C/C++ code can be quickly implemented and optimized into an RTL design using Vivado HLS and then exported as a pcore that is connected with AXI4-Streaming interface, generated automatically by the tool itself, to the ACP of the Zynq Programmable Subsystem (the Cortex-A9 ARM CPU, so called "PS") through a Direct Memory Access (DMA) core in the Programmable Logic (PL) subsystem of the Zynq device. The usage of Xilinx Platform Studio (XPS) and Software Development Kit (SDK) tools is illustrated with great detail to design respectively the hardware of the Zynq PL – which includes the matrix multiplier peripheral, the DMA engine and an AXI timer – and the software running on the Zynq PS to manage such peripherals.

From 3 weeks full time of embedded HW design with the old tool-chain to only 4h in Vivado-IPI

Outlines

- about Xilinx
- Xilinx DSP design flows
- **Vivado High Level Synthesis (HLS)**
- Linear Algebra with HLS: QRD for Adaptive Beamforming
- Conclusion
- References

Vivado HLS



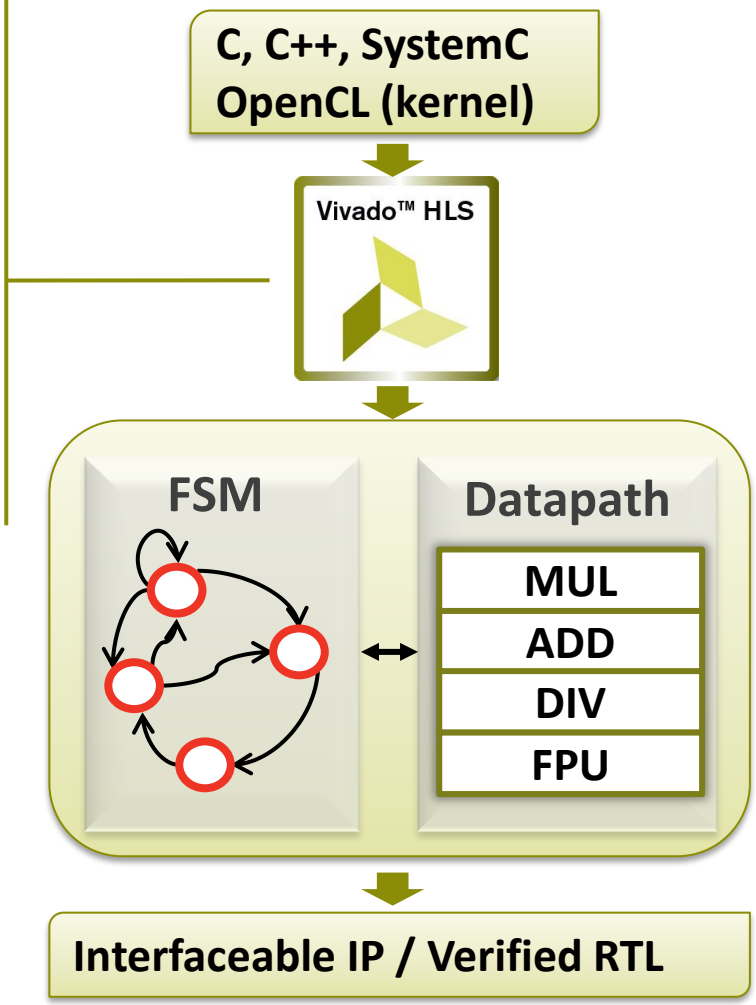
Abstract, no clocks!

Specific, timed

Accelerates Algorithmic C to RTL Creation

Vivado HLS – Synthesis

- ***Directives / Pragmas***
- ***Constraints***
- ***Libraries***
  - *Arbitrary Precision*
  - *Video*
  - *Math*
  - *Linear algebra*
  - *IP: FFT and FIR*

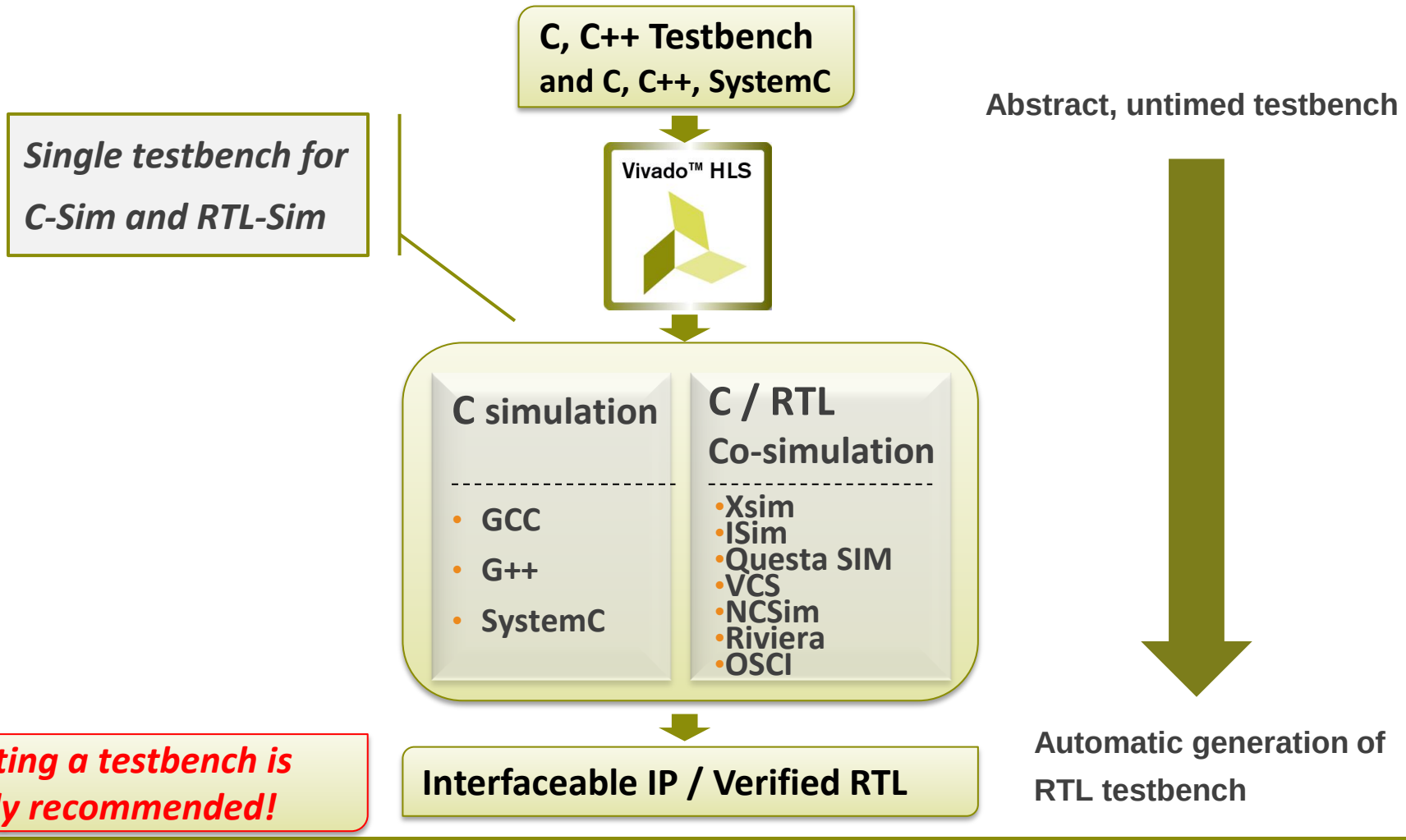


Abstract, untimed

Target optimized, timed,
Connectivity ready

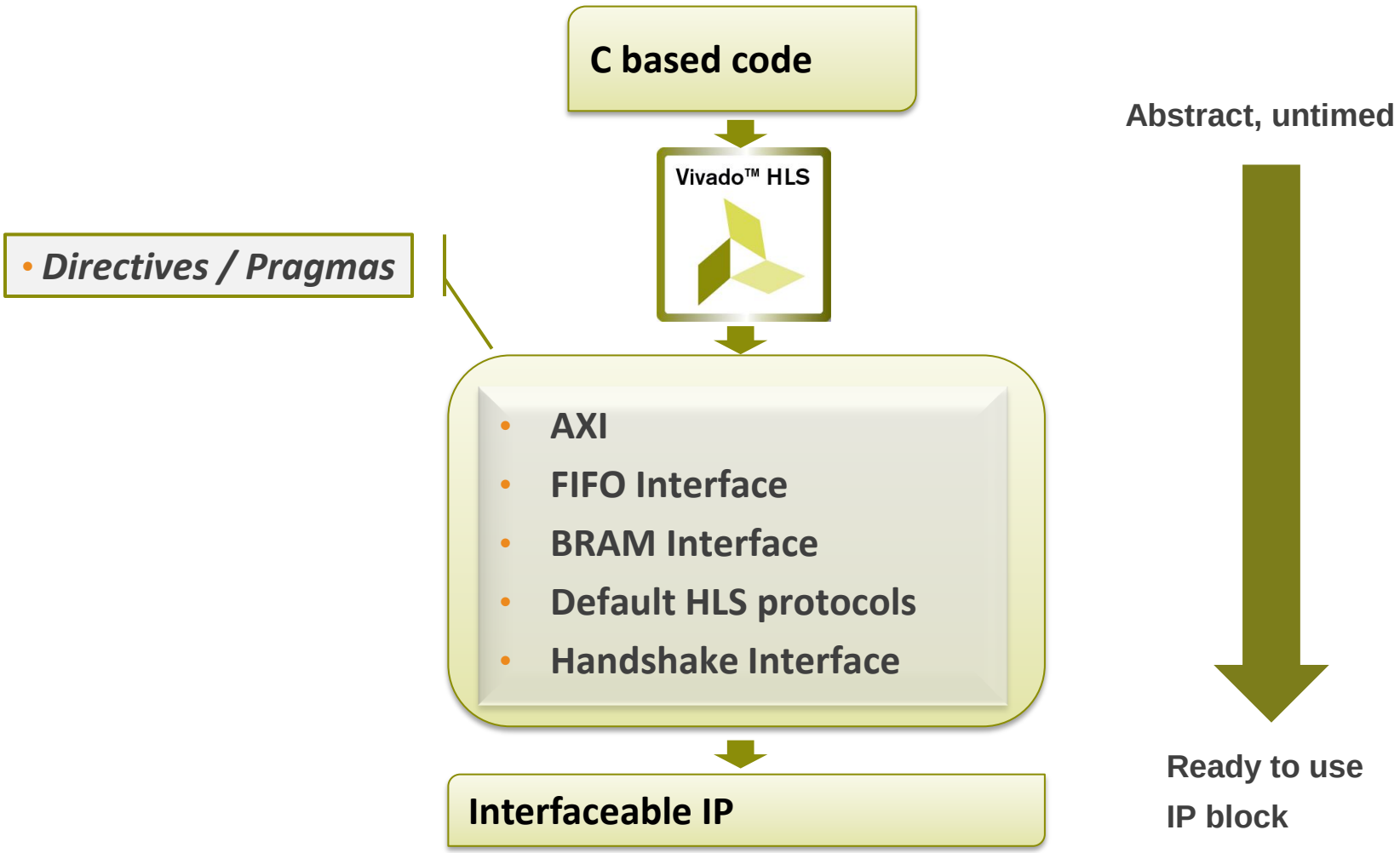
Accelerates Algorithmic C to RTL Creation

Vivado HLS – Verification



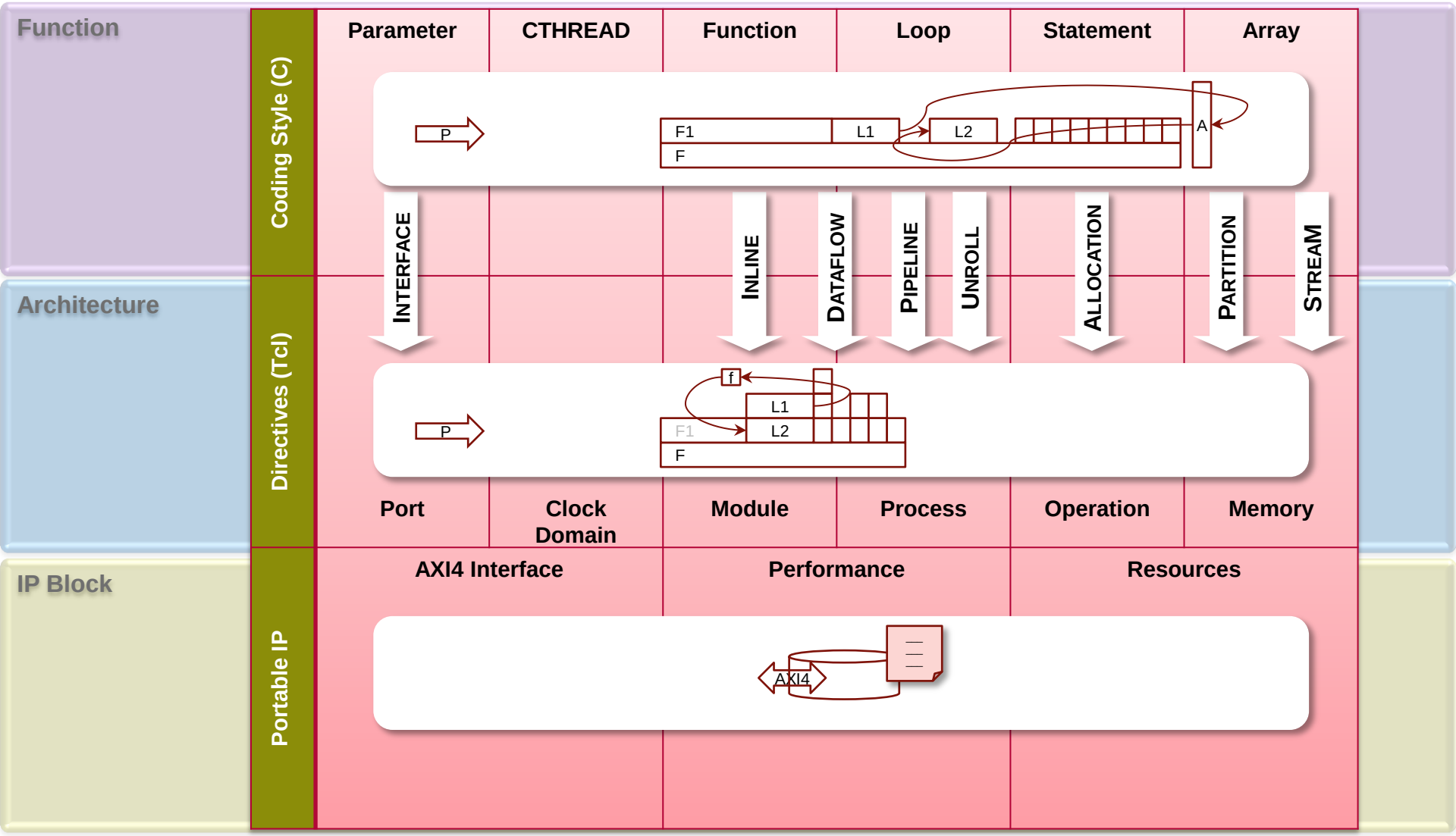
Accelerates Algorithmic C to RTL Creation

Vivado HLS – Interfaces



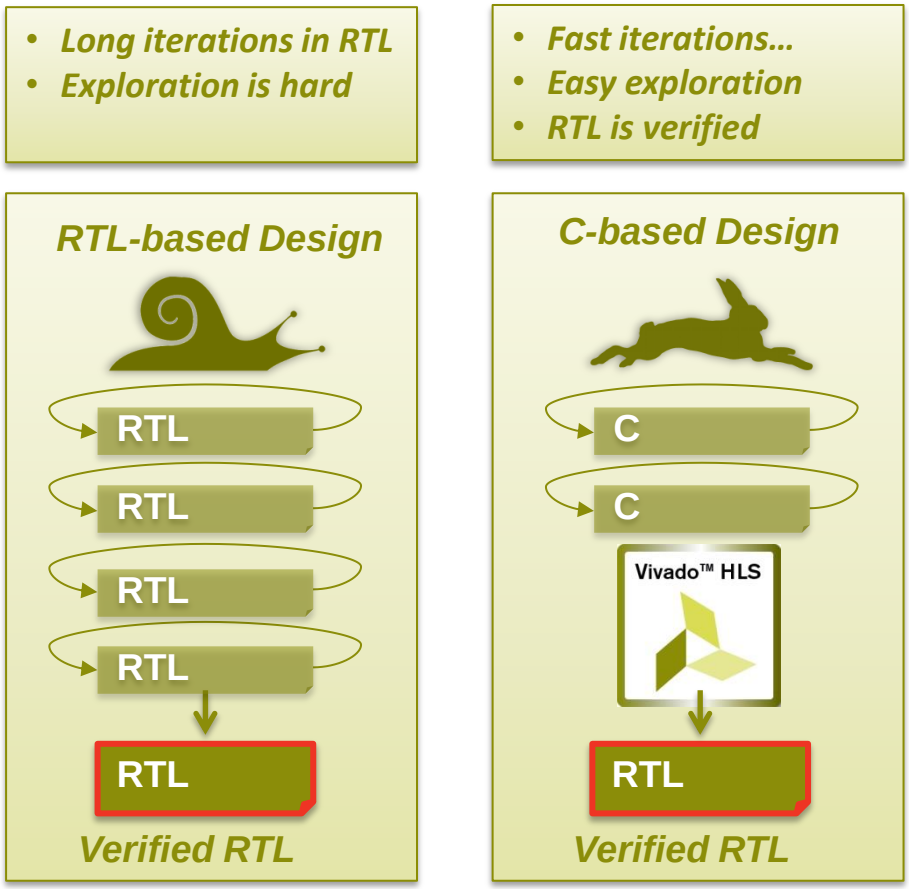
Accelerates Algorithmic C to RTL Creation

Core Technology



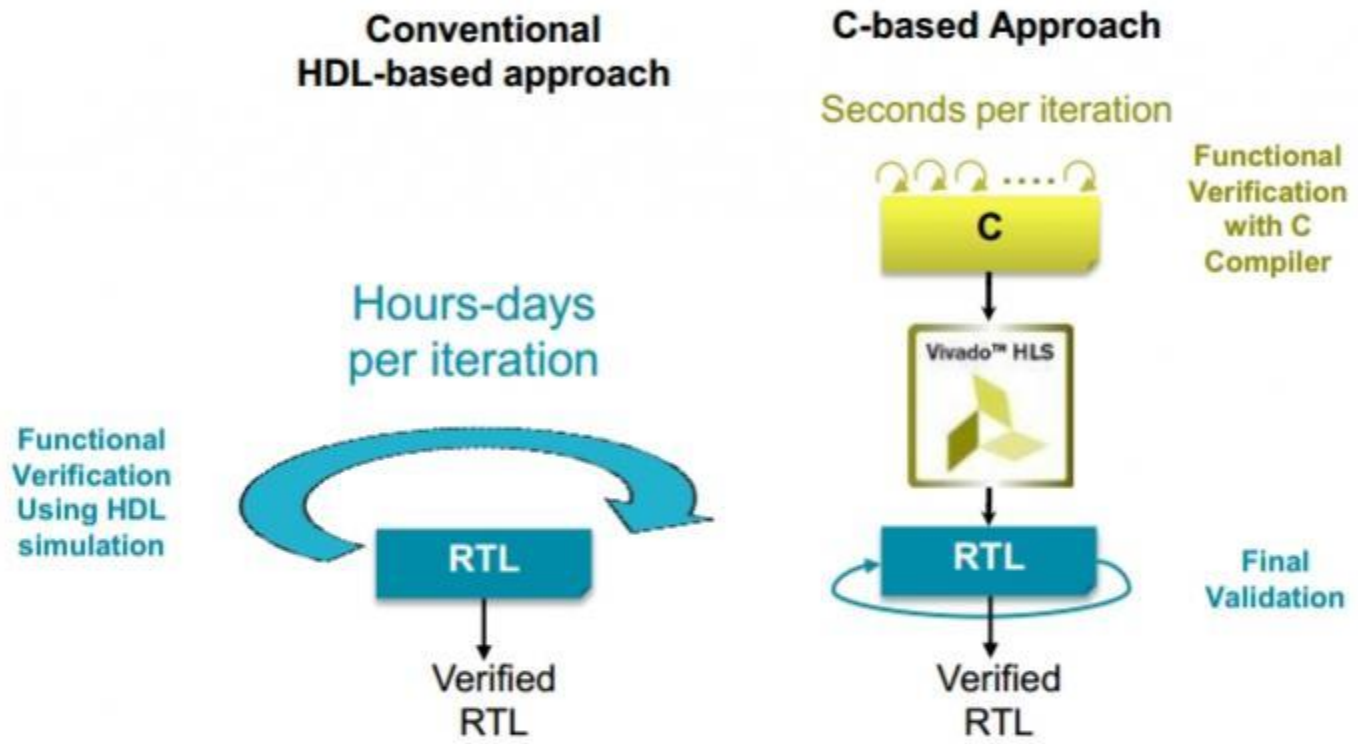
The HLS Productivity Boost…

- **Quick algorithmic convergence**
  - Fast simulation and synthesis
  - Automatic generation of RTL testbench
- **Optimized RTL Output**
  - Code is architecture-aware
- **Interfaces**
  - Included in generated RTL



RTL simulation verification time reduced from days to seconds

C/C++ Simulation is Orders of Mag. Faster!



Optical flow video example

Input	RTL Simulation Time	C Simulation Time	Acceleration
10 frames of video data	~2 days	10 seconds	~12,000X

*RTL Simulations performed using ModelSim

Vivado HLS System IP Integration Flow

C-based IP Creation

C, C++, SystemC

Libraries

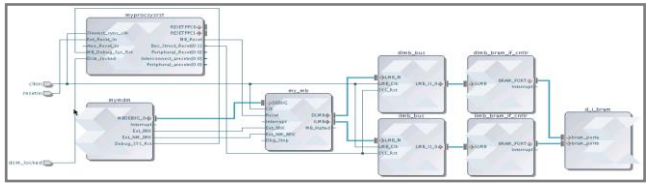
- Arbitrary Precision
- Video
- Math
- Linear algebra
- IP: FFT and FIR

Vivado™ HLS

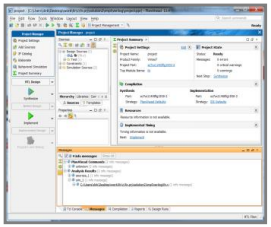
VHDL or Verilog

System Integration

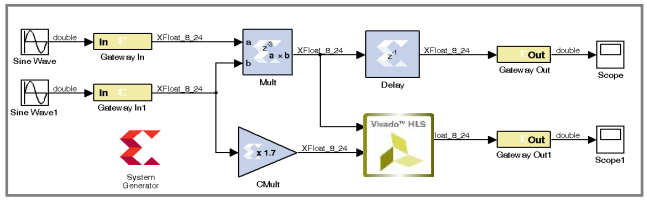
IP Catalog



Vivado IP Integrator



Vivado RTL



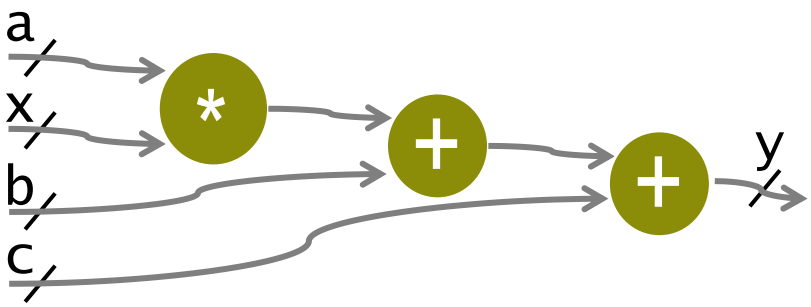
System Generator for DSP

Vivado HLS Integrates into System Flows

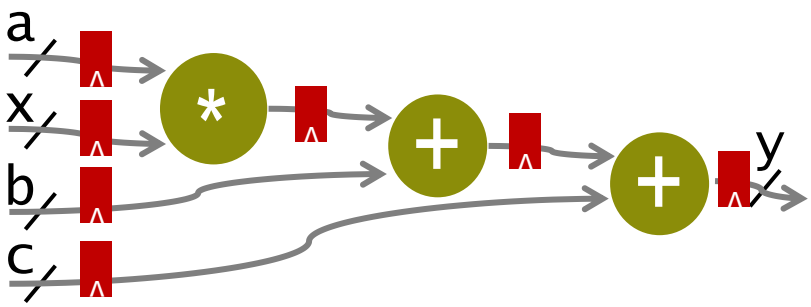
Datapath Synthesis

*Example: y = a*x + b + c;*

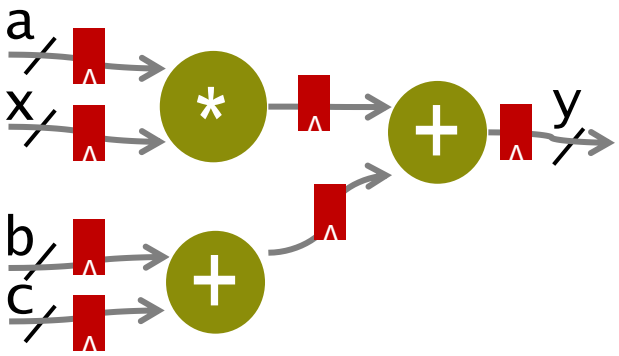
1. HLS begins by extracting a data flow graph (DFG), a functional representation



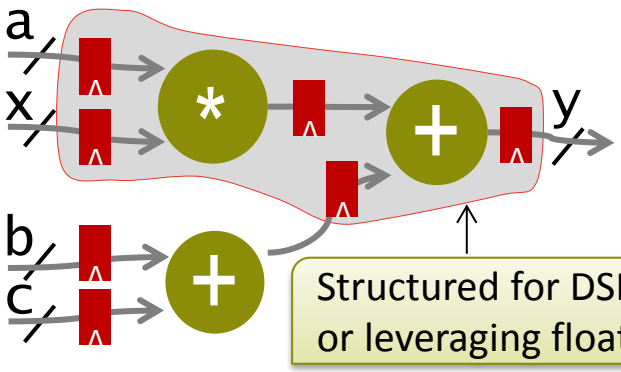
2. Accounts for target Fmax to determine minimum required pipelining (not yet the optimal implementation)



3. Expression balancing for latency reduction
4. Restructuring to optimize fabric resources



5. Restructuring for optimized DSP48

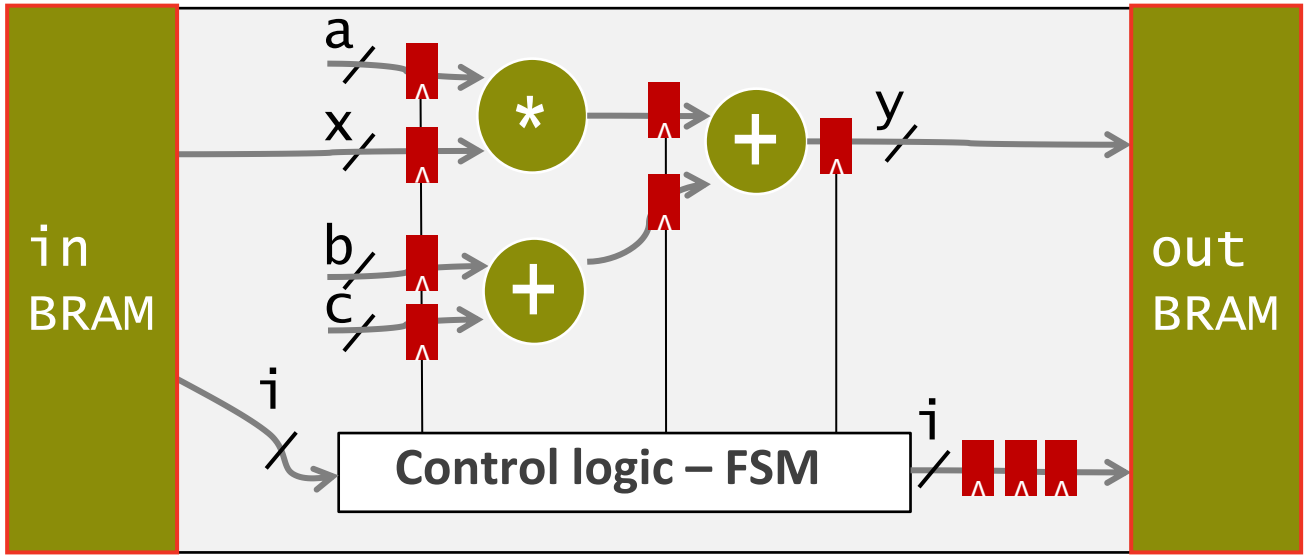


Structured for DSP inference or leveraging floating point IPs

Interface Synthesis – Completing the design…

- C function arguments become RTL interface ports

```
f(int in[20], int out[20]) {
  int a,b,c,x,y;
  for(int i = 0; i < 20; i++) {
    x = in[i]; y = a*x + b + c; out[i] = y;
  }
```



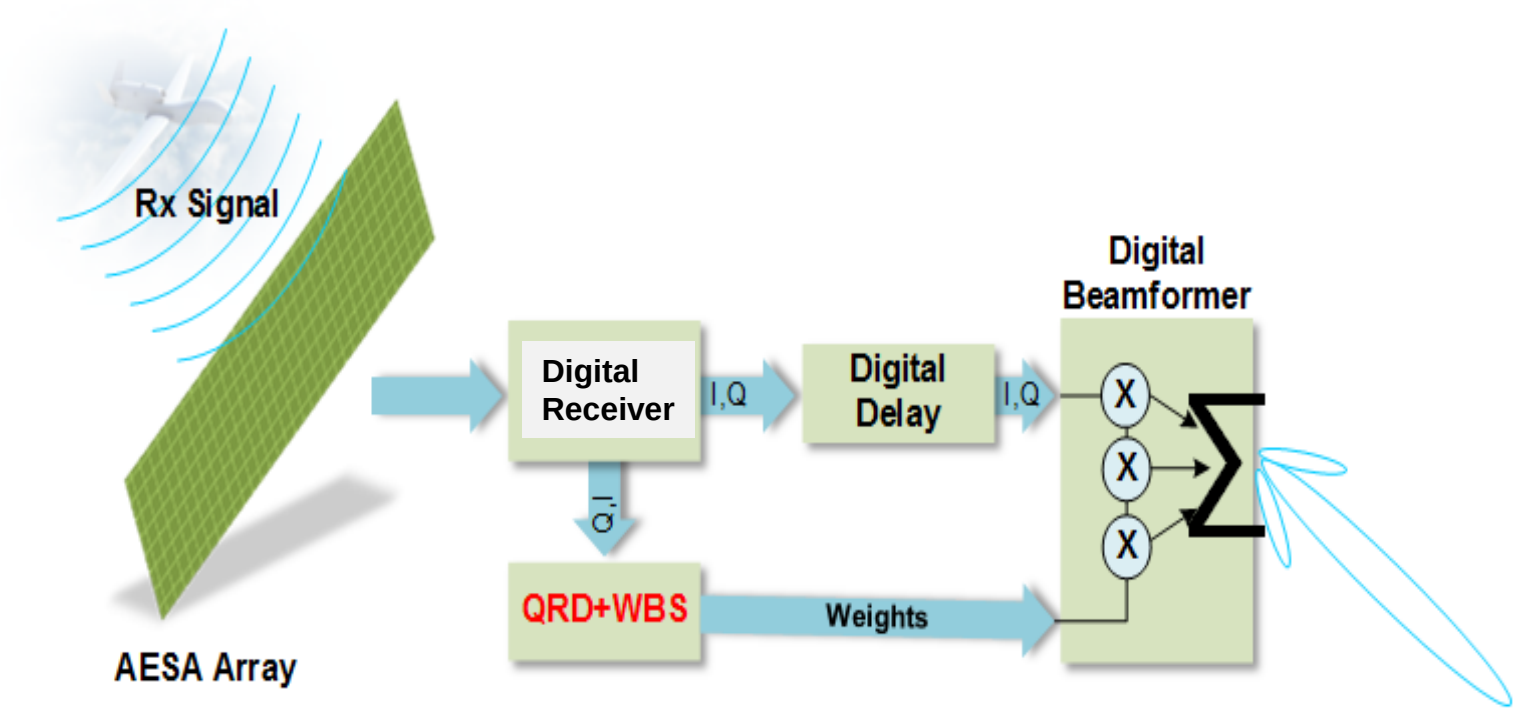
The State Machine Automatically Adapts to the Design Interface

Outlines

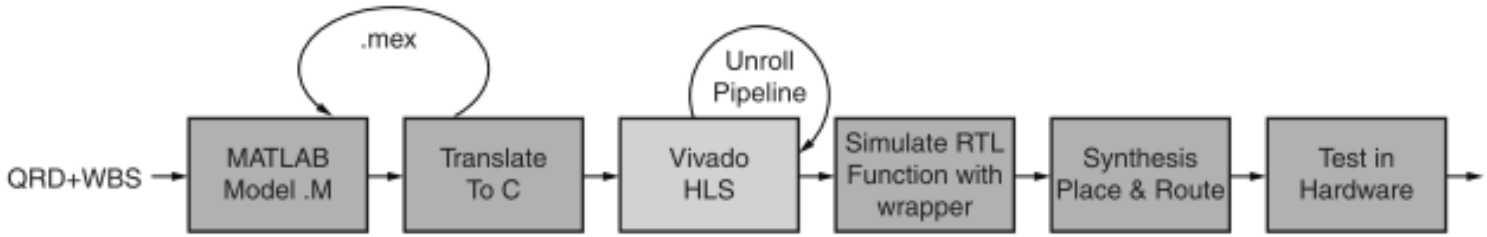
- about Xilinx
- Xilinx DSP design flows
- Vivado High Level Synthesis (HLS)
- **Linear Algebra with HLS: QRD for Adaptive Beamforming**
- Conclusion
- References

Adaptive Digital Beamforming

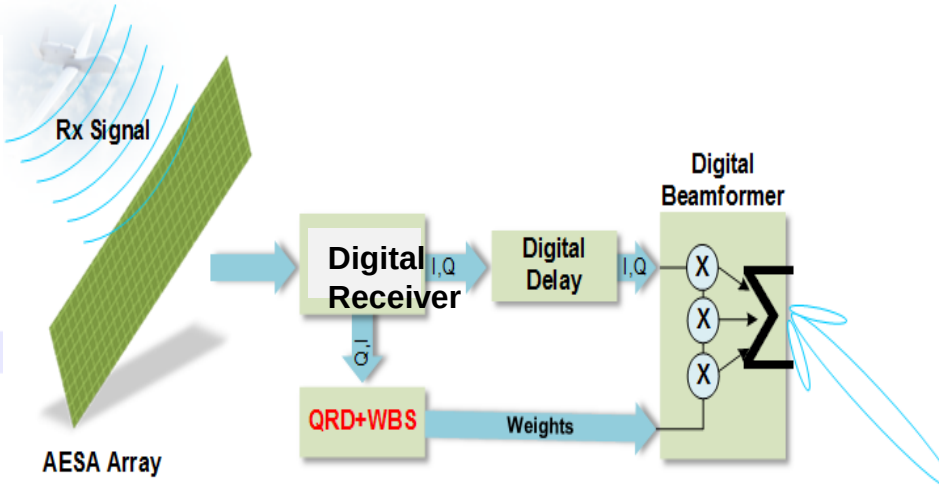
- A beam-agile RADAR can be created by calculating complex floating point adaptive weights, received the during Pulse Repetition Intervals
- Modified Gram-Schmidt (MGS) QR Decomposition (QRD) computes the matrix inversion – The Heart of Adaptive Beamforming



MGS QRD + Weight Back Substitution




```
4   for i=1:cols,
5       Q(:,i)=A(:,i);
6           for j=1: i -1,
7               R(j,i)=Q(:,j)'*Q(:,i);
8               Q(:,i)=Q(:,i) - (R(j,i)*Q(:,j) );
9           end
10      R(i,i)=norm(Q(:,i) );
11      Q(:,i)=Q(:,i)/R(i,i);
12  end
```



- This is the Hardest Floating Point Problem for HW design
- This Real System Design took 6 Months using VHDL. Using HLS, it took 4 hours!
- 260x Speed-Up. This does not include system integration which will also be Faster!

MGS QRD+WBS RESULTS

Summary of timing analysis

Estimated clock period (ns): 6.79

Summary of overall latency (clock cycles)

- Best-case latency: 3
- Average-case latency: 93027
- Worst-case latency: 420163

Summary of loop latency (clock cycles)

- I3
- I9
- I11

Area Estimates

Summary

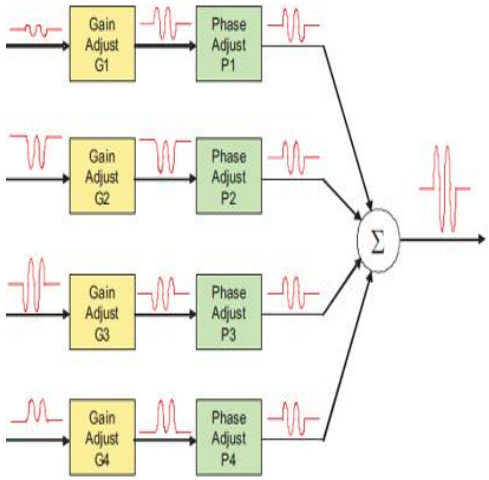
	BRAM_18K	DSP48E	FF	LUT	SLICE
Component	-	392	44457	47081	-
Expression	-	-	0	6559	-
FIFO	-	-	-	-	-
Memory	128	-	0	0	-
Multiplexer	-	-	-	21984	-
Register	-	-	19130	-	-
ShiftMemory	-	-	0	276	-
Total	128	392	63587	75900	0
Available	2940	3600	866400	433200	108300
Utilization (%)	4	10	7	17	0

- **128x64 Complex FP ~3.3ms (125 MHz)**
- **FP Matrix Inversion thanks to Vivado HLS is trivial in FPGAs**
- **If it can be done Mathematically, It can be done in HLS**
- **C/C++ easily moves to CPUs**
  - Truly Portable Libraries
  - Rapid Trade Space Exploration

Look for WP452 'Adaptive Beamforming for Radar: Floating-Point QRD+WBS in an FPGA'

HLS Beamformer— Easy for Vivado HLS
16 channels - 32k samples @ 200 MHz

```
Floating Point Beamformer
*******************************************************************************/
#include <math.h>
#include <stdio.h>

 void bfr_fp( float chr[16][32000], float chi[16][32000], float wr[16][32000], float wi[16][32000], float br[32000],float bi[32000]) {

    float rr,ii;
    int i,k,bb;


    u0: for(i=0; i<32000; i++){
    #pragma HLS PIPELINE
        rr=0; ii=0;


        u1: for(k=0;k<16;k++){
        #pragma HLS UNROLL
            rr = rr + (chr[k][i] * wr[k][i]) - chi[k][i]*wi[k][i];
            ii = ii + ((chr[k][i] + chi[k][i])*(wr[k][i]+wi[k][i])) - chr[k][i]*wr[k][i] - chi[k][i]*wi[k][i];
        }

            br[i] = rr;
            bi[i] = ii;

    }

 }
```



16 Channel Beamformer RESULTS

Summary of timing analysis

Estimated clock period (ns): 3.46

Summary of overall latency (clock cycles)

Best-case latency: 256442

Average-case latency: 256442

Worst-case latency: 256442

Summary of loop latency (clock cycles)

u0

Area Estimates

Summary

	BRAM_18K	DSP48E	FF	LUT	SLICE
Component	-	46	5442	4110	-
Expression	-	-	0	309	-
FIFO	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	2902	-
Register	-	-	7594	-	-
ShiftMemory	-	-	0	2176	-
Total	0	46	13036	9497	0
Available	2940	3600	866400	433200	108300
Utilization (%)	0	1	1	2	0

- **Took about 15 minutes to design**
- **Uses only 46 DSPs!**
- **Beamforms—**
  - 16 Complex channels
  - 32k Complex Samples per Channel
  - Every 1.3 ms in Floating Point per beam
- **Since its floating point—**
  - RADAR front end can change and the design will still work
- **To process more beams—**
  - Just change the C code and rerun HLS
- **Can optimize to reduce latency—**
  - At expense of increased DSP48 usage
- **This is just a starting point…**
  - Infinite designs are possible

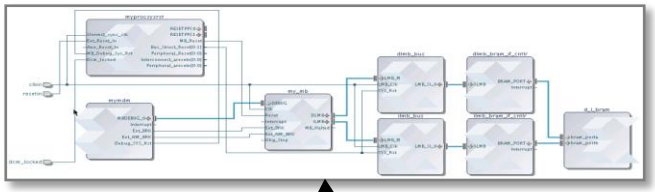
THINK ABOUT WHAT YOU CAN DO WITH VIVADO HLS!

Outlines

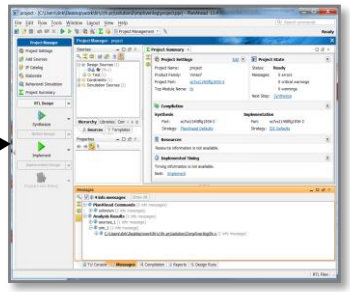
- about Xilinx
- Xilinx DSP design flows
- Vivado High Level Synthesis (HLS)
- Linear Algebra with HLS: QRD for Adaptive Beamforming
- **Conclusion**
- References

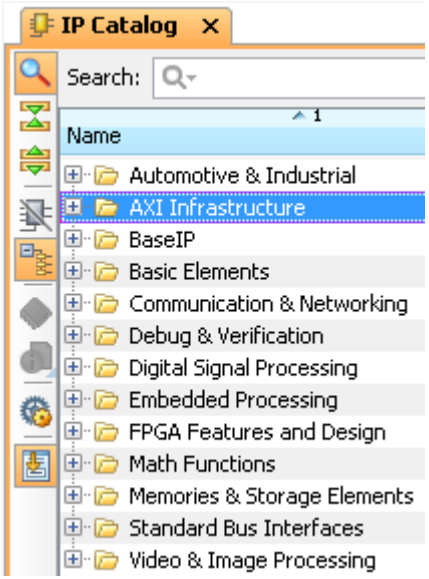
Complete IP & System Centric Design Flow

Vivado IP Integrator



Vivado RTL Integration

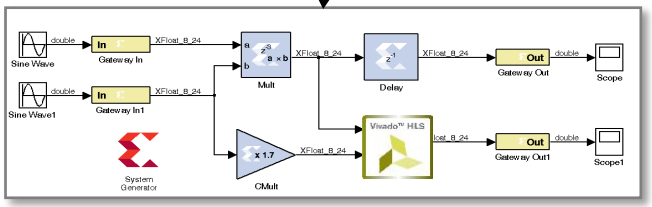




Vivado IP Catalog

Vivado™ HLS

Vivado HLS

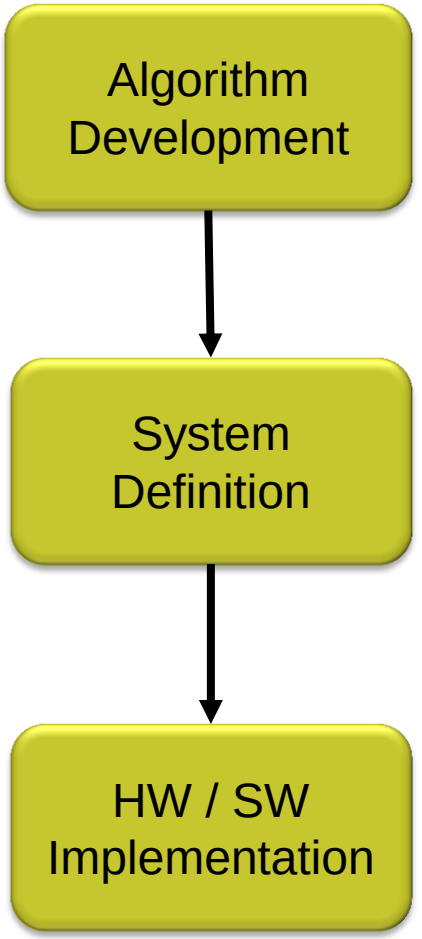


System Generator for DSP

Vivado 2014.2 provides direct flows from all four design environments into and out of the Vivado IP catalog

Benefits of High Level Synthesis

- **Embedded and DSP SW engineers**
  - Take advantage of the performance of custom hardware without RTL design experience
  - Optimize system performance through rapid HW / SW partitioning “what if” analysis and design exploration
  - Accelerates software development productivity by enabling early software development, integration and test without hardware



Vivado HLS Customer Praise…

"In an **HDL** design, each scenario would likely cost an **additional day of writing code** …

With Vivado **HLS** these changes **took minutes**"

R&D Engineer, Agilent Technologies

"I was able to design complex linear algebra algorithms **10x faster** than before with VHDL, and yet achieved **better QoR** with Vivado HLS."

Design Engineer, Major A&D contractor

"..we always use **C** to quickly build a **system-level model** for validation of **key algorithms**.. **problem** .. **quickly and efficiently convert C into a HDL**".

"With Xilinx Vivado HLS, …. … used **C to implement a key algorithm** … **into Verilog**. We **verified** both the functionality and performance in **Xilinx devices** … "

Central R&D Data Center CTO, ZTE Inc.

Radar Design 1024 x 64 QRD Floating-Point data path	Conventional Hand-coded HDL Approach	Using Vivado High Level Synthesis
Design Language	VHDL (RTL)	C
Design Time (weeks)	12	1
Latency (ms)	37	21
Memory (RAMB18E1)	134 (16%)	10 (1%)
Memory (RAMB36E1)	273 (65%)	138 (33%)
Registers	29686 (9%)	14263 (4%)
LUTs	28152 (18%)	24257 (16%)

Source: *Design Engineer at Major A&D contractor*

"For each project where we used Vivado HLS, we **saved 2-3 weeks** of engineering **time**."

CTO, Major broadcast equipment company

THINK ABOUT WHAT YOU CAN DO WITH VIVADO HLS!

Outlines

- about Xilinx
- Xilinx DSP design flows
- Vivado High Level Synthesis (HLS)
- Linear Algebra with HLS: QRD for Adaptive beamforming
- Conclusion
- **References**

HLS documentation

- **There are a lot of HLS related articles published in the last 3 years of XCELL Magazine, available here:**
  - http://www.xilinx.com/publications/xcellonline/xcell-past-issues.htm,
- **for example:**
  - Using Vivado HLS to Design a Median Filter and Sorting Network for video www.xilinx.com/publications/archives/xcell/Xcell86.pdf
  - Wireless MIMO Sphere Detector Implemented in FPGA www.xilinx.com/publications/archives/xcell/Xcell74.pdf

- **Xilinx Application notes:**
  - Accelerating OpenCV Applications with Zynq-7000 All Programmable SoC using Vivado HLS Video Libraries http://www.xilinx.com/support/documentation/application_notes/xapp1167.pdf
  - Zynq-7000 All Programmable SoC Accelerator for Floating-Point Matrix Multiplication using Vivado HLS http://www.xilinx.com/support/documentation/application_notes/xapp1170-zynq-hls.pdf

HLS documentation

- Implementing Carrier Phase Recovery Loop Using Vivado HLS http://www.xilinx.com/support/documentation/application_notes/XAPP1173-carrier-loop.pdf
- Floating-Point PID Controller Design with Vivado HLS and System Generator for DSP http://www.xilinx.com/support/documentation/application_notes/xapp1163.pdf
- Implementing Memory Structures for Video Processing in the Vivado HLS Tool http://www.xilinx.com/support/documentation/application_notes/xapp793-memory-structures-video-vivado-hls.pdf
- Floating-Point Design with Vivado HLS http://www.xilinx.com/support/documentation/application_notes/xapp599-floating-point-vivado-hls.pdf
- Processor Control of Vivado HLS Designs http://www.xilinx.com/support/documentation/application_notes/xapp745-processor-control-vhls.pdf
- Zynq All Programmable SoC Sobel Filter Implementation Using the Vivado HLS Tool http://www.xilinx.com/support/documentation/application_notes/xapp890-zynq-sobel-vivado-hls.pdf

HLS documentation

- **Xilinx white paper 452 on Linear Algebra via HLS**
  - http://www.xilinx.com/support/documentation/white_papers/wp452-adaptive-beamforming.pdf
- **Xilinx white paper 437 on encryption via HLS**
  - http://www.xilinx.com/support/documentation/white_papers/wp437-ZUC-Cipher-Zynq.pdf
- **Xilinx User Guide (for FPGA and HLS beginners)**
  - Introduction to FPGA Design with Vivado High-Level Synthesis http://www.xilinx.com/support/documentation/sw_manuals/ug998-vivado-intro-fpga-design-hls.pdf

White Paper 452: Linear Algebra via HLS

White Paper: 7 Series FPGAs and Zynq-7000 AP SoCs

WP452 (v1.0) June 24, 2014

Adaptive Beamforming for Radar: Floating-Point QRD+WBS in an FPGA

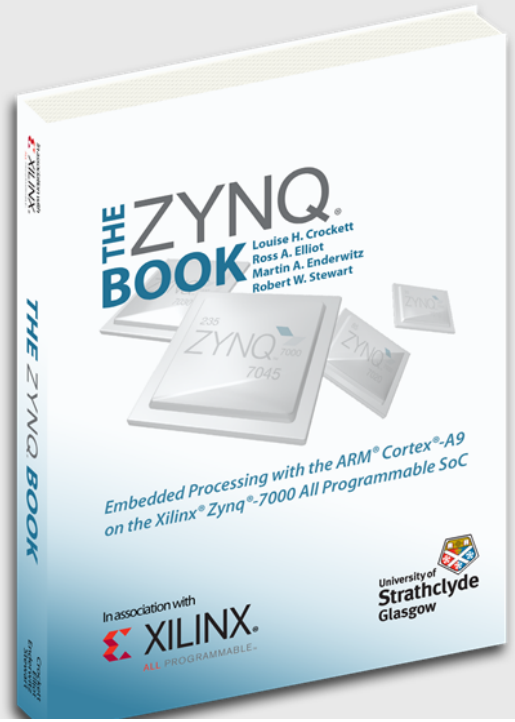
By: Luke Miller

The Vivado® HLS tool allows development of fully verified floating-point radar QRD+WBS algorithms for Xilinx® FPGAs in hours, not weeks or months, in a native C environment.

http://www.zynqbook.com/

THE ZYNQ BOOK

HOME ABOUT AUTHORS CONTACT FAQ DOWNLOADS BUY



Welcome to the online home of The Zynq Book

The Zynq Book is all about the Xilinx Zynq®-7000 All Programmable System on Chip (SoC) from Xilinx. This is the online home of The Zynq Book, designed to raise awareness of the book and host the accompanying tutorials. Thanks for finding us!

The Zynq Book is the first book about Zynq to be written in the English language. It has been produced by a team of authors from the University of Strathclyde, Glasgow, UK, with the support of Xilinx. We wanted to create an accessible, readable book that would benefit people just starting out with Zynq, and engineers already working with Zynq. We hope that it will prove a handy reference that remains on your desktop! You can find out more about the book's contents on the **About** page.

Buy a Hardcopy of The Zynq Book

The book comprises 24 themed chapters, and is printed in full colour throughout (including over 150 figures). It is available internationally for a suggested price of $30 (US), £21.50 (UK), €23 (Europe), and similar prices in other countries. The **Buy** page provides more details about how to obtain your copy.

PDF copy The Zynq Book

You can download a free pdf copy of the Zynq Book from the **Downloads** area.

We hope you will enjoy The Zynq Book! If you would like to get in touch with us with questions, feedback etc., then please use the **Contact** form – we look forward to hearing from you!

Louise, Ross, Martin and Bob
July 2014

XAPP 1163

Application Note: Vivado HLS

XILINX®

XAPP1163 (v1.0) January 23, 2013

Floating-Point PID Controller Design with Vivado HLS and System Generator for DSP

Author: Daniele Bagni, Duncan Mackay

Summary

This application note describes how to quickly implement and optimize floating-point Proportional-Integral-Derivative (PID) control algorithms specified in C/C++ code into an RTL design using Vivado HLS. You can use System Generator for DSP to easily analyze and verifiy the design. This enables floating-point algorithm designers to take advantage of high-performance, low cost, and power efficient Xilinx® FPGA devices.

Introduction

Floating-point algorithms are widely used in industries from analysis to control applications. Traditionally, such algorithms have been implemented on microprocessors. The primary reason for using microprocessors has been the ease with which floating-point algorithms can be captured, validated, and debugged in C/C++ code, therefore avoiding the complexity and skills required to implement them in hardware. However, implementing these algorithms on optimized and dedicated hardware provides lower cost, higher performance, and power benefits over a standard, or even optimized microprocessor.

This application note presents a new design flow enabled by the Xilinx Vivado™ Design Suite, which allows floating-point algorithms to be quickly specified in C/C++ code, optimized for performance, and implemented on Xilinx FPGA devices. This flow delivers on the cost,

http://www.xilinx.com/publications/archives/xcell/Xcell86.pdf

ASK FAE-X

Median Filter and Sorting Network for Video Processing with Vivado HLS

by Daniele Bagni
DSP Specialist
Xilinx, Inc.
daniele.bagni@xilinx.com

Daniele Bagni *is a DSP Specialist FAE for Xilinx EMEA in Milan, Italy. After earning a degree in quantum electronics from the Politecnico of Milan, he worked for seven years at Philips Research labs in real-time digital video processing, mainly motion estimation for field-rate upconverters. For the next nine years he was project leader at STMicroelectronics' R&D labs, focusing on video coding algorithm development and optimization for VLIW architecture embedded DSP processors, while simultaneously teaching a course in multimedia information coding as an external professor at the State University of Milan. In 2006 he joined the Xilinx Milan sales office. What Daniele enjoys most about his job is providing customers with feasibility studies and facing a large variety of DSP applications and problems. In his spare time, he likes playing tennis and jogging.*