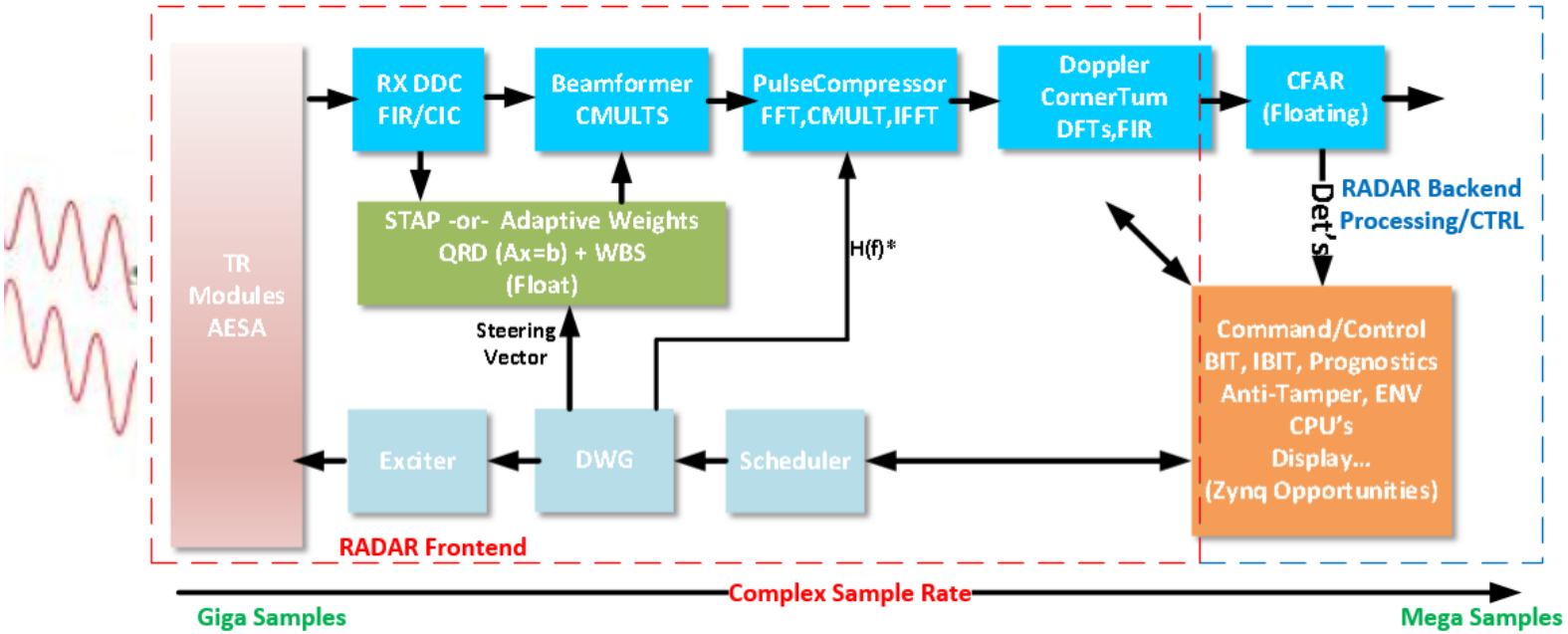
New Design Tools Help You Develop Radar That Sees the Unseeable and Detects the Undetectable

정 웅, DSP Specialist, Xilinx

Nov 18, 2015

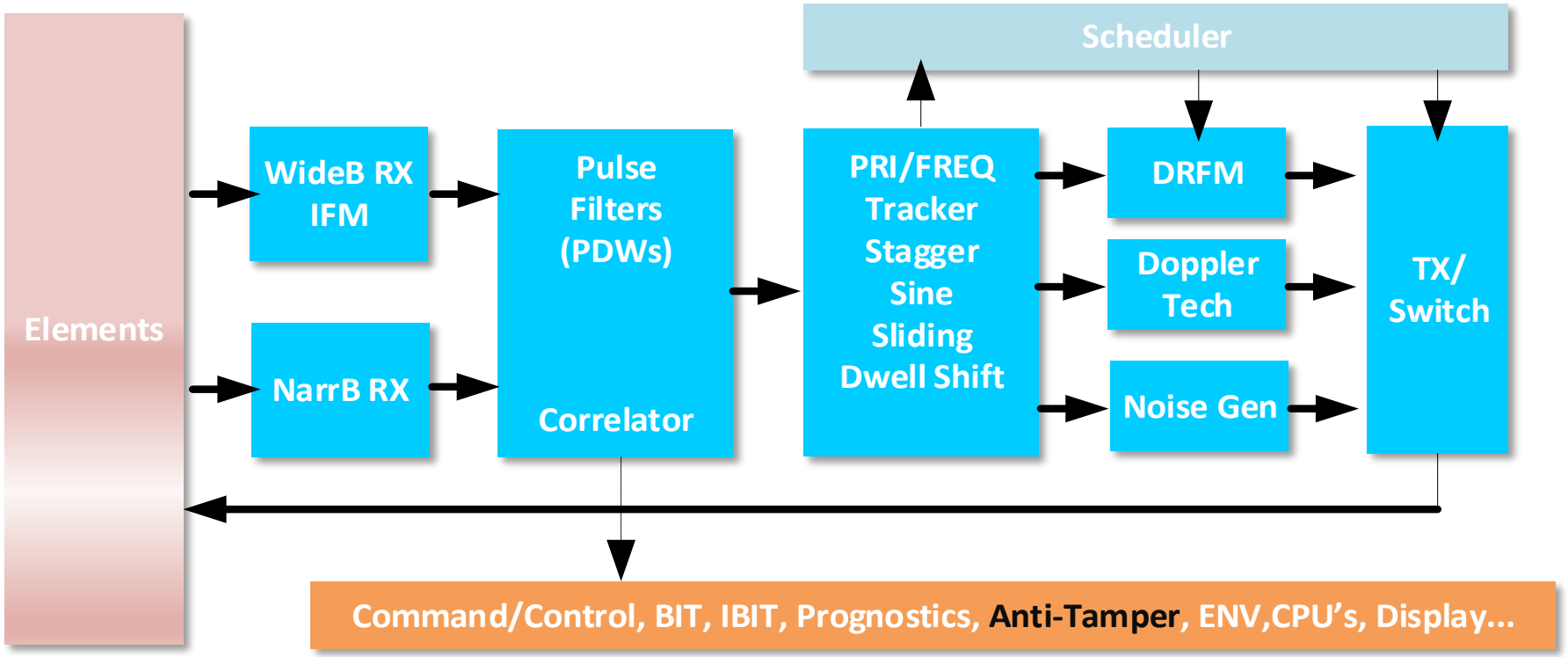
Xilinx In All Aspects of RADAR - Open by Design



- **Xilinx FPGAs Ensure Mission Success for Today and the Future**
  - 20,16,7nm FPGAs and Multi-Processor SoCs ensure world class functionality
  - Xilinx FPGAs have a role in the active array down to backend processing
  - Floating Point in same FPGA allows adaptive solutions at lower power
  - Xilinx FPGAs are open hardware from the start

Xilinx FPGAs Help Stay Ahead of the Enemy by Reprogrammability

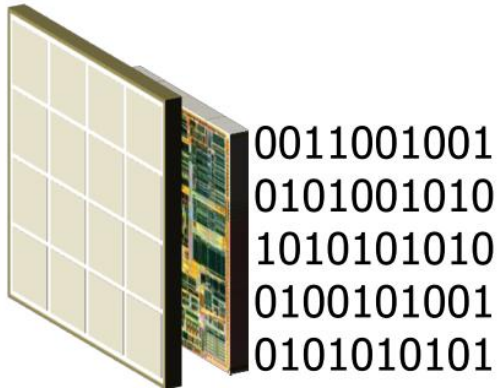
Xilinx In All Aspects of Electronic Warfare



- EW Systems push limits of FPGAs, High ADC/DAC Sample Rates
- EW needs many, many DSPs (FIRs, Channelization…) and I/O
- Latency is Critical → PRI Tracking is done under 200ns
- All RADAR Systems need some EW/ECCM Solution

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

The Transmit Receive Module (TRM)

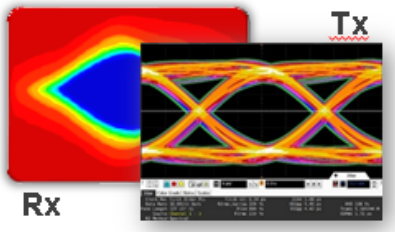


- **Expect Highly Integrated TRMs - Each Module is a Mini-RADAR**
  - Integrated Data Converters, ADCS + DACS - **DIRECT RF is HERE**
  - **AESA Element Level Processing is here as well!**
  - MIMO Beamforming - Goal is to get beams off the array NOT I,Q samples
  - Power and package are a concern, but a DSP/GT challenge overall
  - **This is the future of RADAR systems. These Modules replace:**
    - **Digital Receiver, Digital Waveform Generator, and Beamformer**
    - **STAP, Adaptive Floating Point Calculators**

Moving Data On and Off the RADAR/EW System

Xilinx

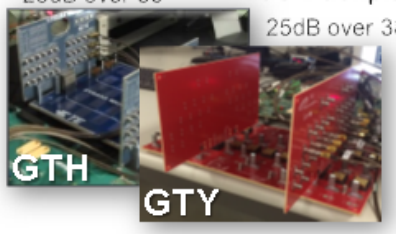
Up to 28Gbps



- ✓ Robust Auto Adaptive DFE
- ✓ Absolute Production Quality

16G Backplane
25dB over 30"

28G Backplane
25dB over 38"



- ✓ 1st Tapeout Success
- ✓ Unmatched Backplane Performance

UltraSCALE+

- ✓ Proven Transceiver Architecture
  - 20nm GTH / GTY foundation
- ✓ Proven Design Team
  - Multi-node track record of success

28nm → 20nm → FinFET

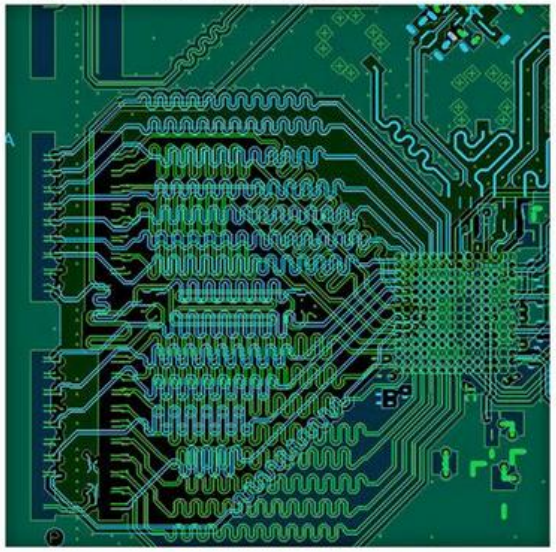
Altera

- ✗ Marginal DFE / Auto Adaptation
- ✗ Production Errata
- ✗ +4 Mask Spins for GT issues
- ✗ Unable to yield target specs
- ✗ New Design Required
  - Adding 28G backplane
- ✗ Poor Foundation
  - 20nm VSR solution struggling

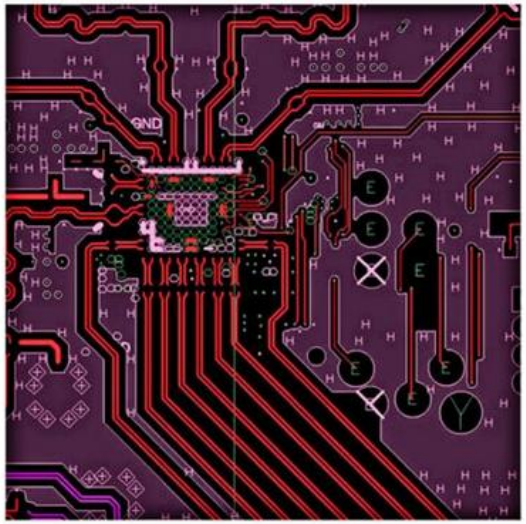
- Xilinx 16nm (33g) 128 GTs allows over 7Tb
- Xilinx 20nm SERDES can support GEN2, 3 HMC
- Xilinx SERDES Reduces Risk for System Requirement Changes

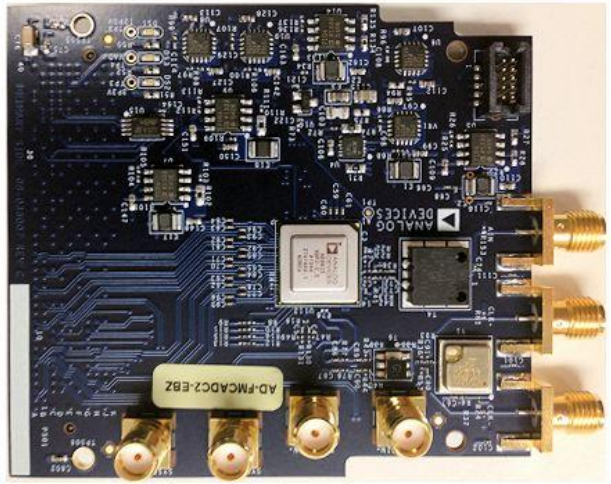
Xilinx's JESD204B IP Enabling RADAR/EW Design

DAC with LVDS Interface



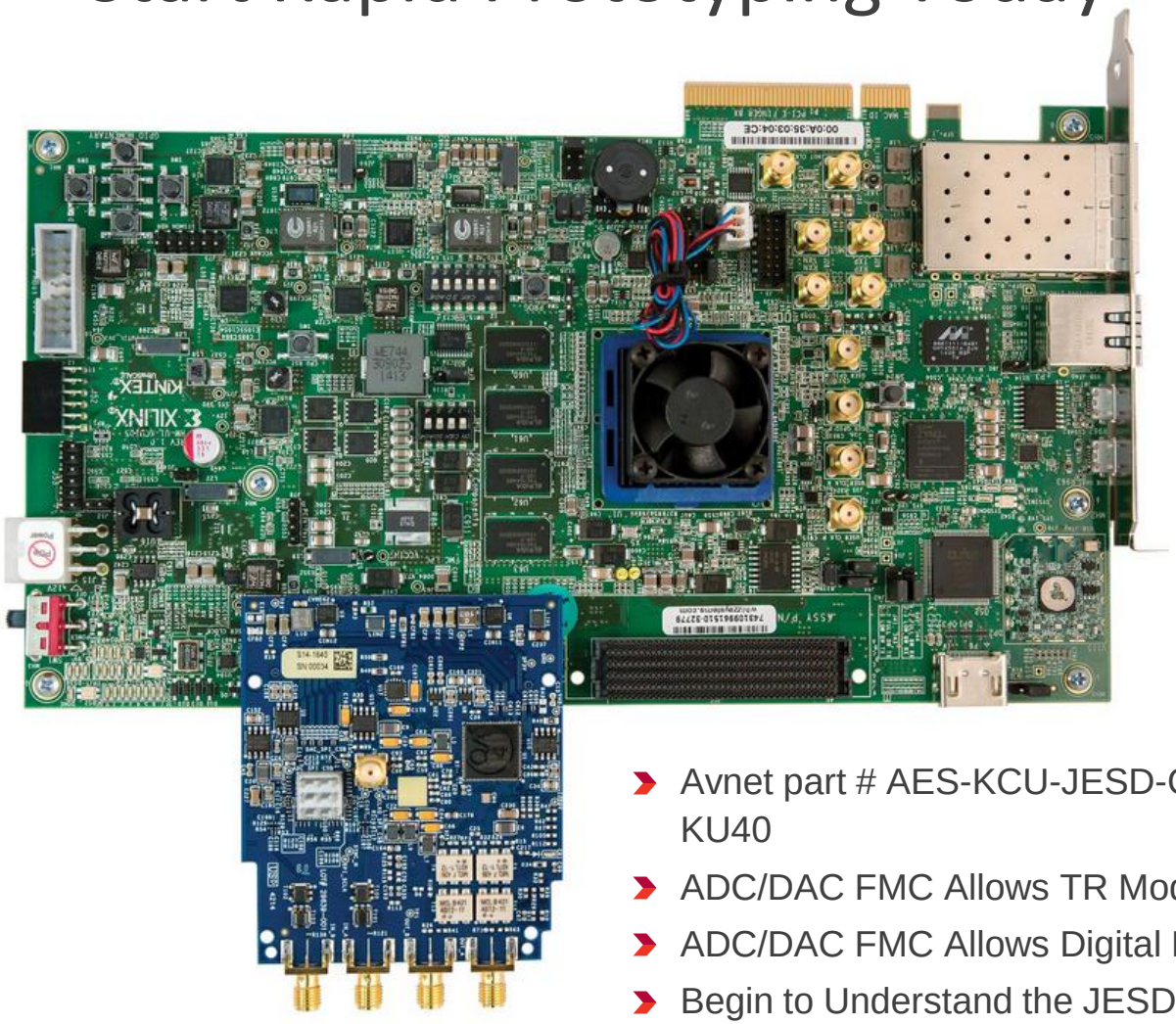
DAC with JESD204B





- **JESD204b Is Great, However…**
  - Latency is too high for EW; Sensor to PRI Tracker sub 100ns latency
- **Expect a New Standard to Address Low Latency – JESD204C?**
- **Xilinx has a working out of the box solution**
  - Partnered with all the major data converter companies
  - ADI, Intersil, TI, IDT

Start Rapid Prototyping Today - Think Demo Schedules



Analog Devices AD-FMCDAQ2-EBZ FMC Module

AD9680 Dual 14-Bit, 1.0 GSPS, A/D Converter

- JESD204B (subclass 1) 8-lane coded serial digital outputs
- SFDR 77 dBc at 500 MHz Ain, 1 GSPS
- ENOBs = 10.9 bits
- 2 GHz of usable analog input full-power bandwidth
- Two integrated DDCs per channel

AD9144 Quad 16-Bit, 2.8 GSPS Transmit D/A Converter

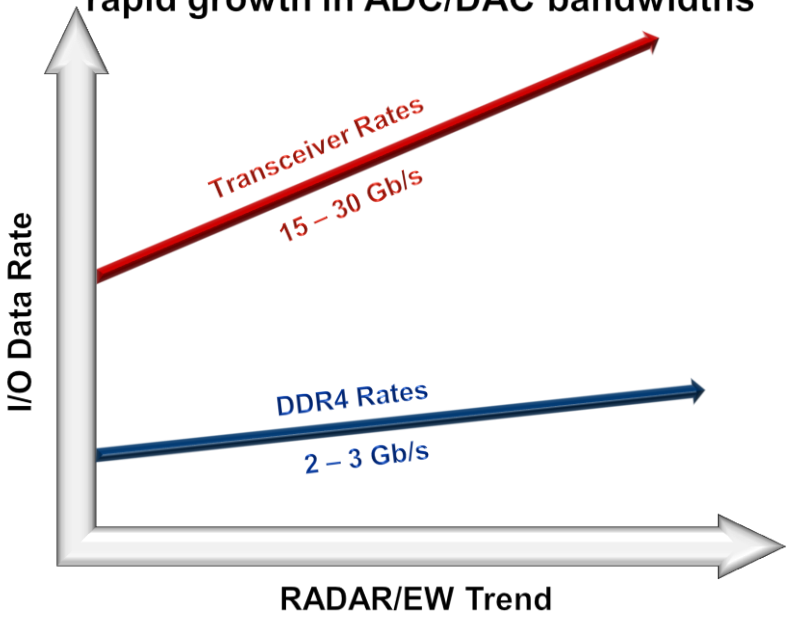
- JESD204B (subclass 1) coded serial digital outputs
- Supports complex signal bandwidths up to 800 MHz
- 6-carrier GSM IMD=75 dBc at 75 MHz IF
- SFDR = 85 dBc (BW=300 MHz) at DC IF
- Selectable 2x, 4x, 8x interpolation filters

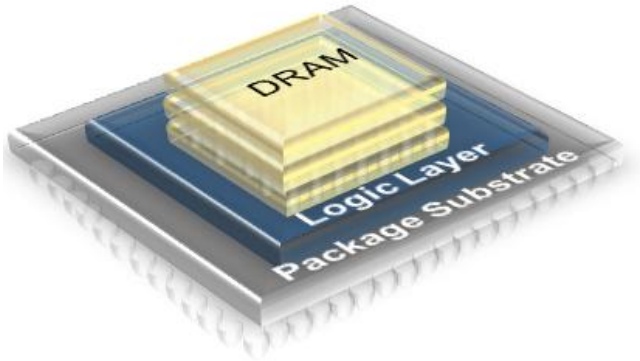
- Avnet part # AES-KCU-JESD-G – Featuring Xilinx 20nm Kintex® KU40
- ADC/DAC FMC Allows TR Module Designs
- ADC/DAC FMC Allows Digital RF Module Design
- Begin to Understand the JESD204b Interface

Check Out Xilinx's Boards and Kits Web Page at
http://www.xilinx.com/products/boards-and-kits.html

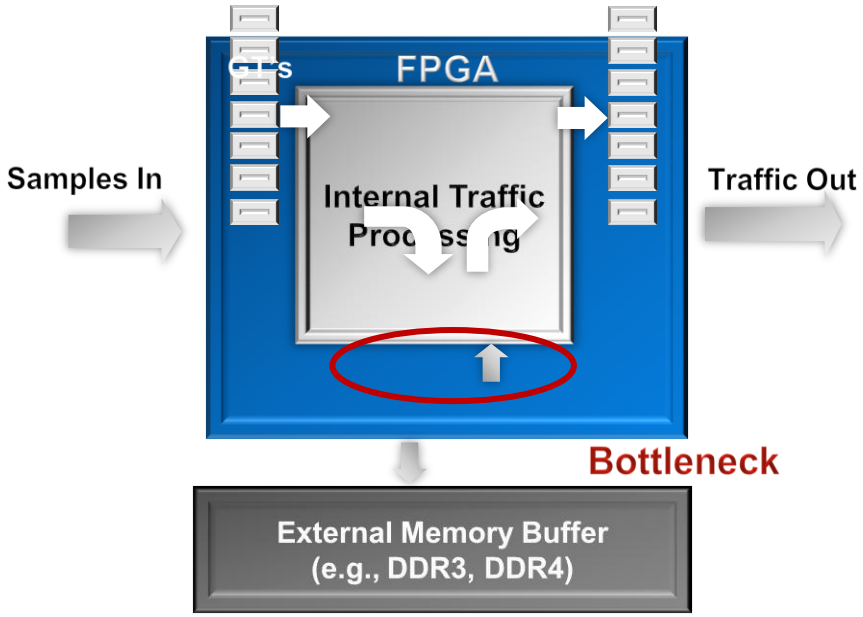
Parallel Memory Struggling to Meet New Bandwidth Demands

Parallel memory rates not keeping up with rapid growth in ADC/DAC bandwidths





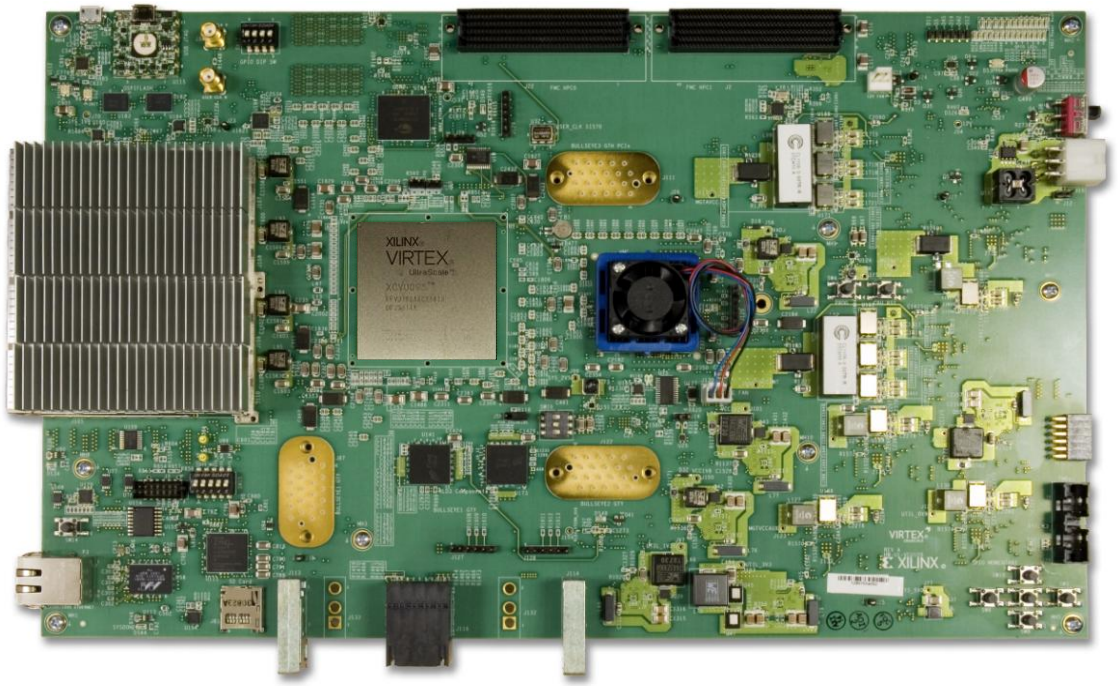
External memory buffering rate must match or exceed RADAR sample rates



2 Types of HMC from Micron

Parameter	BW (GBs)	Line Rates
Gen2 (Current)	120-160	10, 12.5, or 15 Gb/s
Gen 3 (Future)	240-320	15 or 30 Gb/s

UltraScale HMC Evaluation Board: VCU110



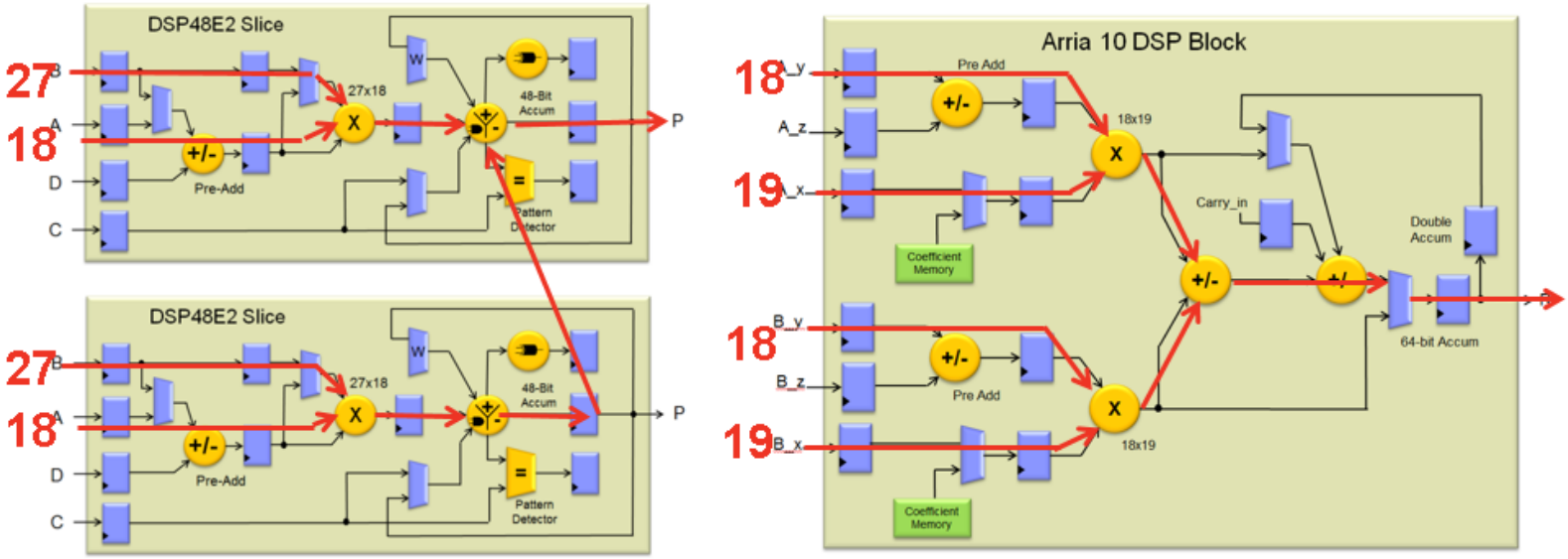
- Benefits of migrating to HMC – above is 2 Full Links = 84 GB/s
  - 15X bandwidth of DDR3, 70% less power, 90% smaller size
- DDR is a board designer's worst nightmare - design is greatly simplified
- EW Systems have aggressive requirements to store all channels at rate
- RADAR Systems need deep buffers for many PRI/CPIs
  - Xilinx's Virtex®-7,UltraScale, and UltraScale+ have world class GT speeds and densities
  - The RADAR/EW developer can evaluate Hybrid Memory Cube

Xilinx Meets the Most Aggressive RADAR/EW

	20nm Kintex UltraScale	16nm Kintex UltraScale+	20nm Virtex UltraScale	16nm Virtex UltraScale+	16nm Zynq UltraScale+
MPSoC Processing System					✓
Logic Cells (K)	355–1,160	205–915	627–4,433	690–2,863	83–915
Block Memory (Mb)	19.0–75.9	5.1–34.6	44.3–132.9	25.3–94.5	4.5–34.6
UltraRAM (Mb)		0–36		90–432	0–36
DSP (Slices)	768–5,520	1,056–3,528	600–2,880	2,280–11,904	240–3,528
DSP Performance (GMAC/s)	8,180	6,287	4,268	21,213	6,287
Transceivers	16–64	16–76	36–120	40–128	0–72
Max. Transceiver Speed (Gb/s)	16.3	32.75	30.5	32.75	32.75
Max. Serial Bandwidth (full duplex) (Gb/s)	2,086	3,268	5,616	8,384	3,268
Integrated Blocks for PCIe®	2–6	0–5	2–6	2–6	0–5
Memory Interface Performance (Mb/s)	2,400	2,666	2,400	2,666	2,666
I/O Pins	312–832	280–668	338–1,456	416–832	76–668
I/O Voltage (V)	1.0–3.3	1.0–3.3	1.0–3.3	1.0–1.8	1.0–3.3

- Over 20 TMACs Real Time DSP Power
- 128, 33g SERDES lanes → 7.7 Tbs of Bandwidth!
  - Paths to JESD204C, HMC Gen3
- ~500 Mb of UltraRAM RAM → Plenty of Capacity for many, many BEAMs
- DDR4 Rates up to 2,666Mbs

UltraScale Architecture DSP Slice Advantage



** Diagram only shows logic for the real component*

*Out_r = (Ar * Br) – (Ai * Bi);*
*Out_i = (Ar * Bi) + (Ai * Br);*

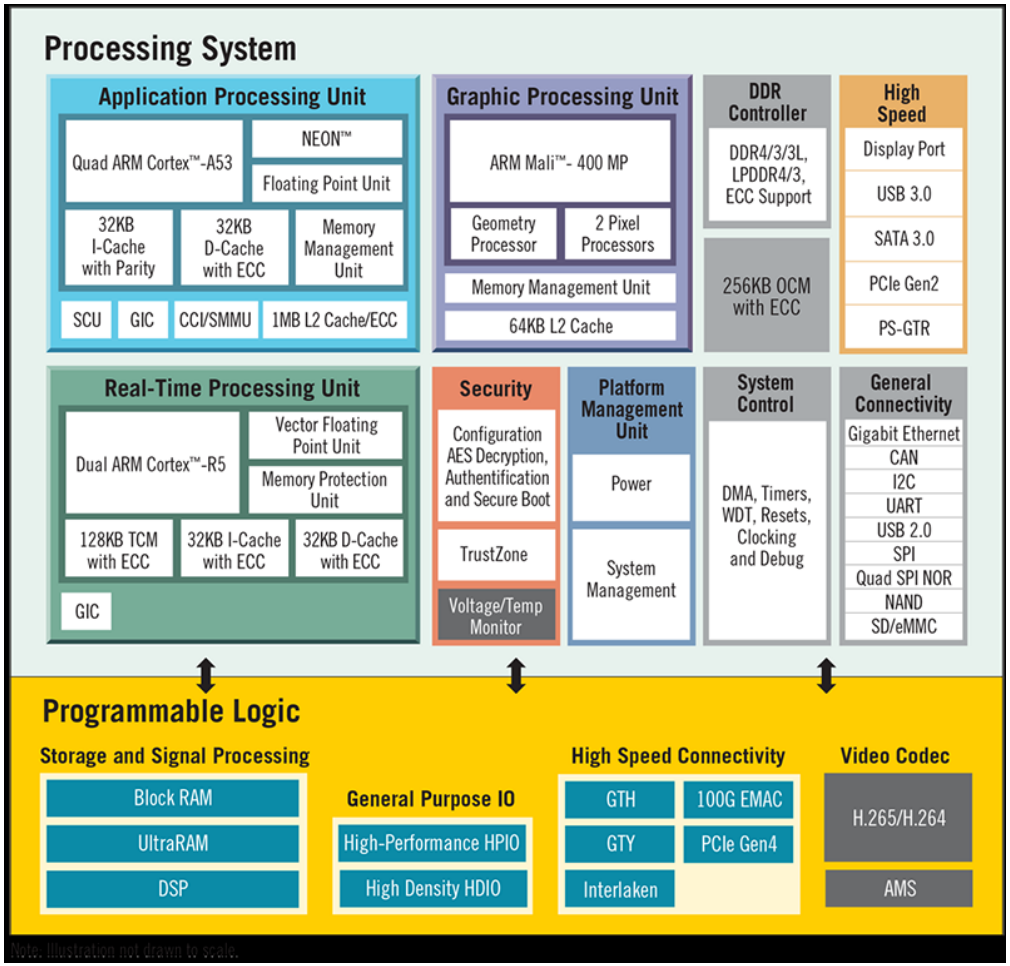
- 18x18 Complex Mult = 1.5 DSP Tiles (3 slices)
- 18x27 Complex Mult = 2 DSP Tiles
- 19x29 Complex Mult = 2 DSP Tiles + 4 LUTs

- 18x18 Complex Mult = 2 DSP Blocks
- 18x25 Complex Mult = 3 DSP Blocks

- Since Altera forces hard floats in Stratix-10, you no longer can perform these fixed point calculations without using valuable FPGA fabric:
- Xilinx lets the RADAR/EW designer choose without penalty
  - Fixed or Float, Xilinx 8.5 TMACs, **Altera 3.5 TMACs**
  - Larger/Wider DSP Port Inputs → Higher Dynamic Range

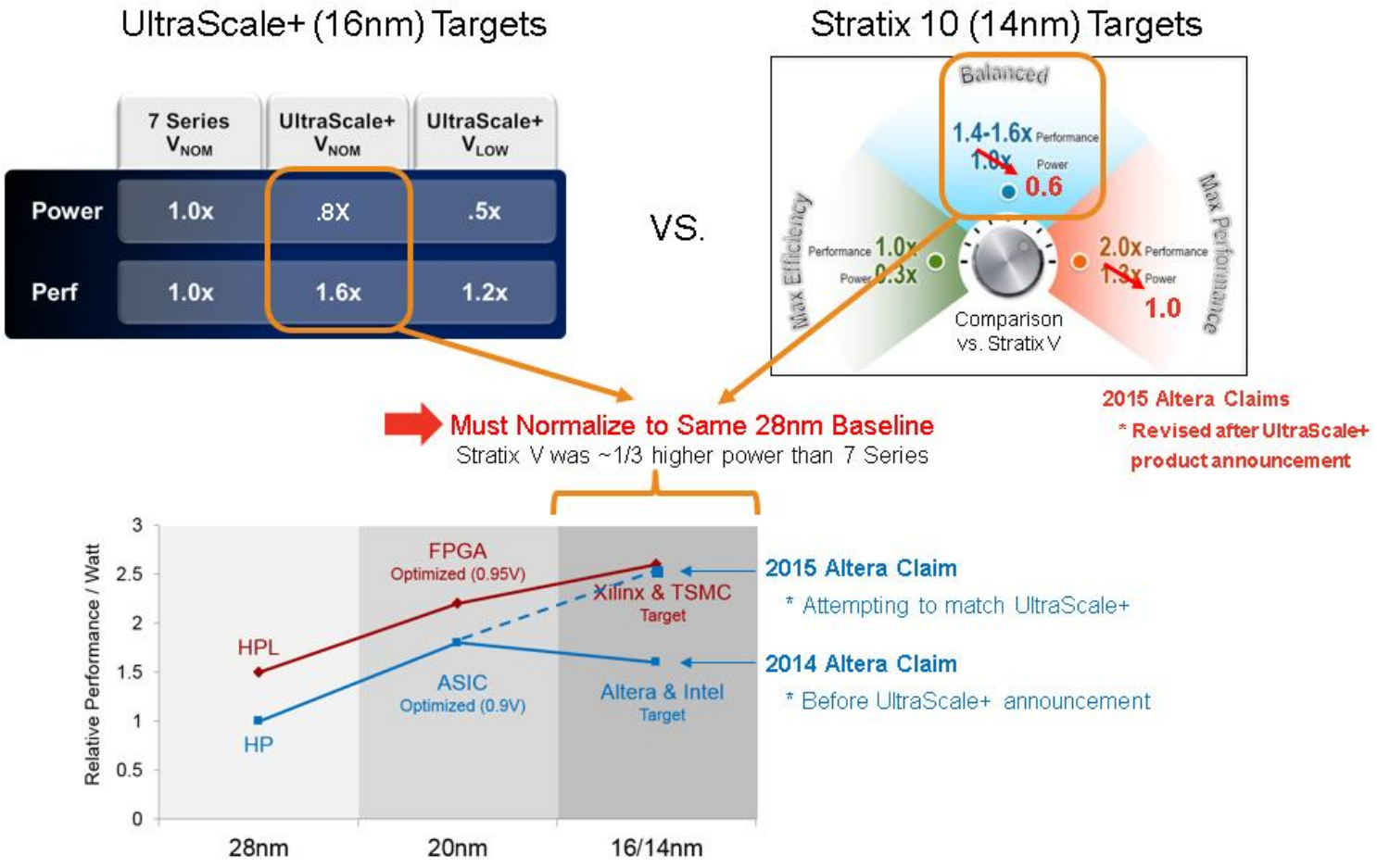
- 18x36 mult
- 36x36 mult
- 54x54 mult
- 18x25 complex mult
- 18x36 complex mult

Xilinx's MPSoC Fits Into Any EW/RADAR System



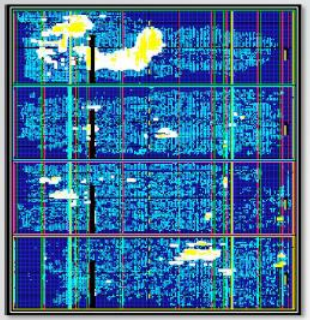
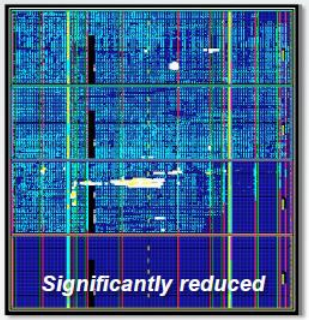
- Perfect for BIT, Status, Control, Cal, Scheduling, Out of Band DSP
- Low Power → Longer UAV Mission Times
- Security for Foreign Military Sales → Anti Tamper

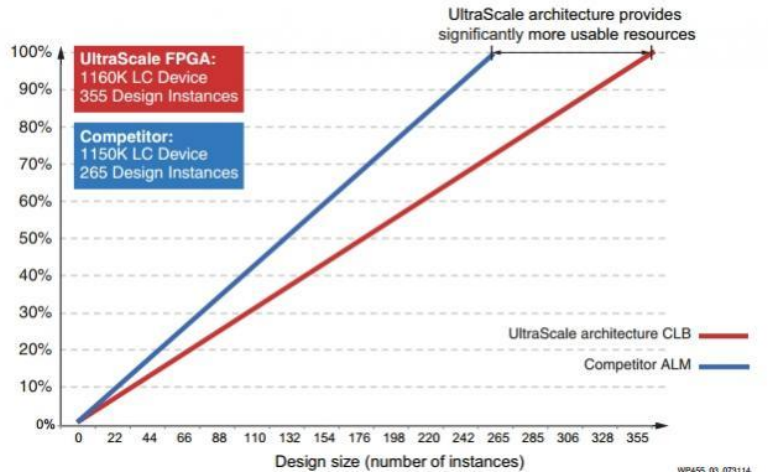
Xilinx Leads the Industry with Low Power by Design

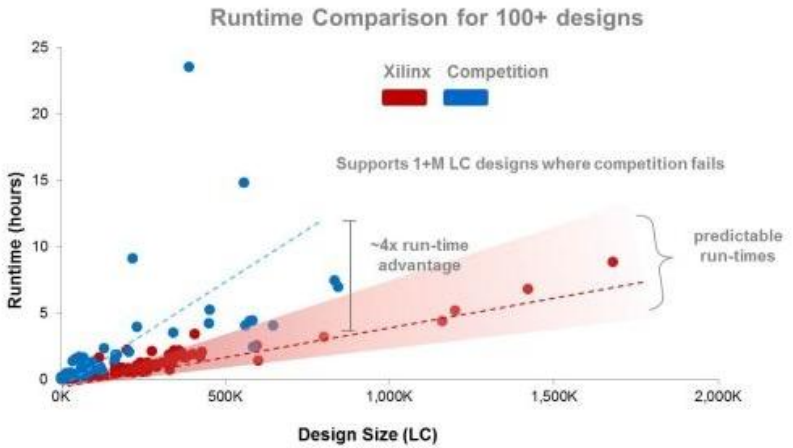


- ✓ Element Level Processing Needs Low Power
- ✓ UAVs, Network of Drones
- ✓ Portable RADAR

Xilinx Reduces the RADAR FPGA Turn Around Time

	ISE	Vivado
P&R runtime	13 hrs	5 hrs
Memory usage	16 GB	9 GB
Reduced Wire length & Congestion		



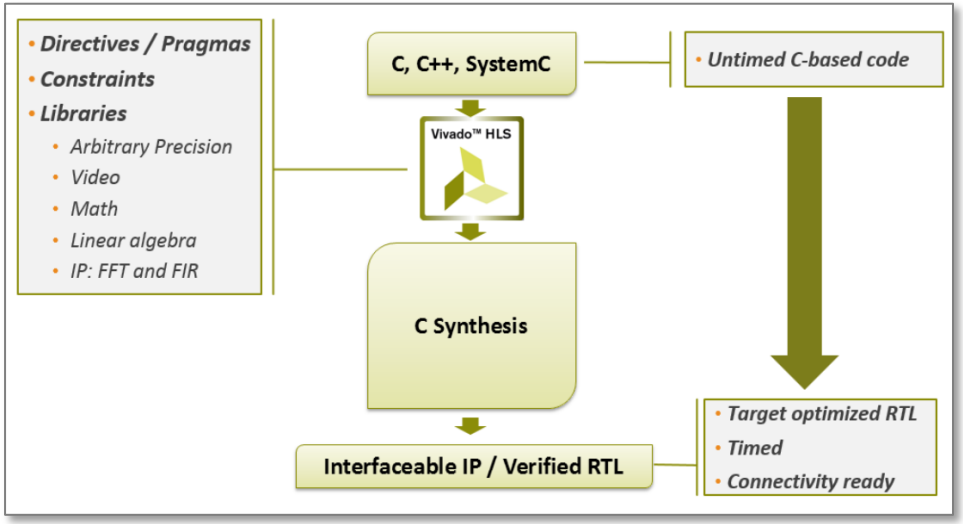


- Implementation Reduction Time
- Pack more design into FPGAs
- Many, design iterations in a day
- Design change was not trivial
- **Predictable runs, every time**

I Know How You Feel During System Integration!

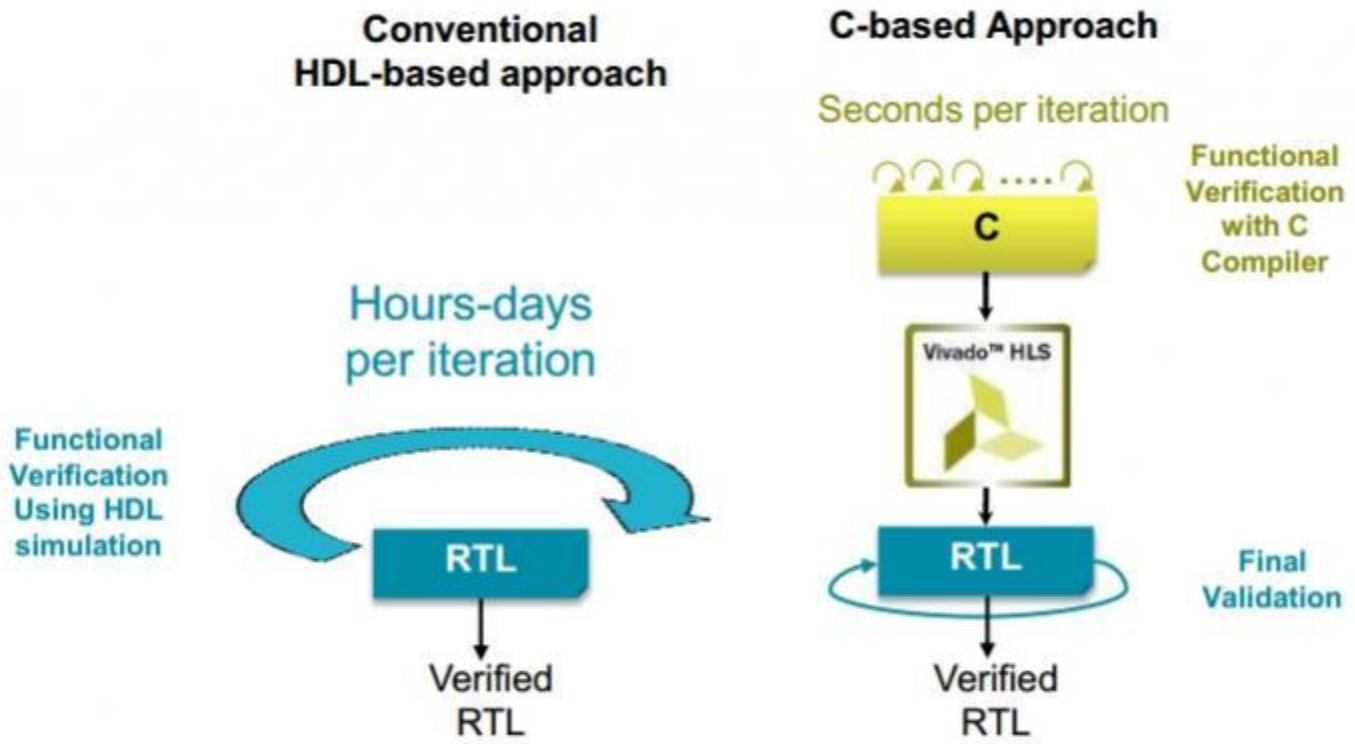
Vivado HLS: Framework for C based Design

- Over 1,000 customers
- Leveraged by LogiCORE IP developers
- C/C++ to optimized RTL



- C to hand-coded quality RTL
  - In weeks not months…
- Accelerated verification
  - Over 100X over RTL
- Ideal for algorithmic designs
  - Excels at math (floating / fixed point)
  - Video, DSP…

C/C++ Simulation is Orders of Magnitude Faster!

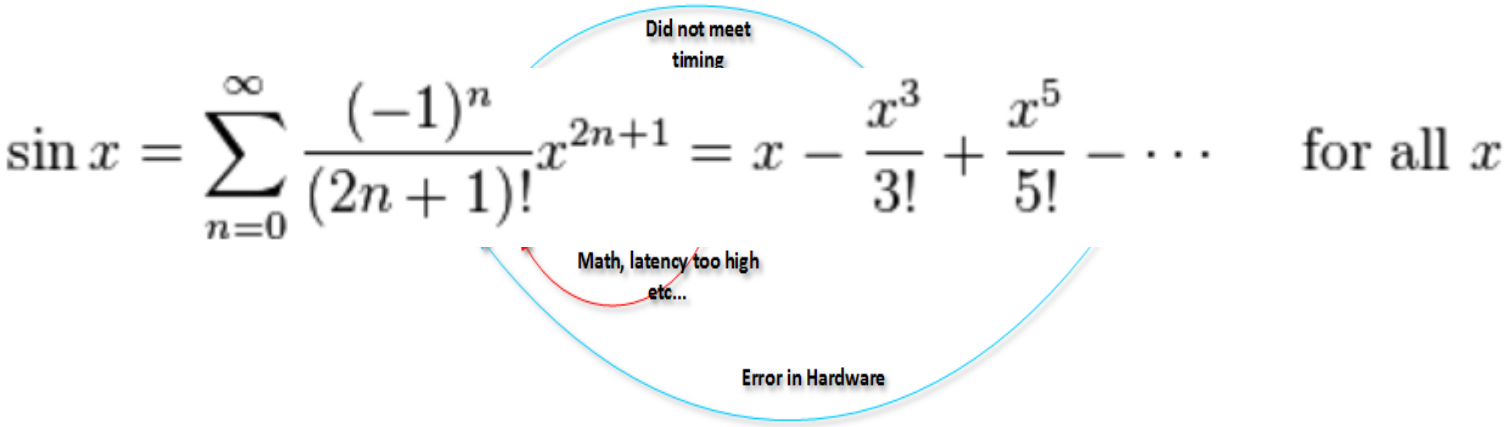


Optical flow video example

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

*RTL Simulations performed using ModelSim

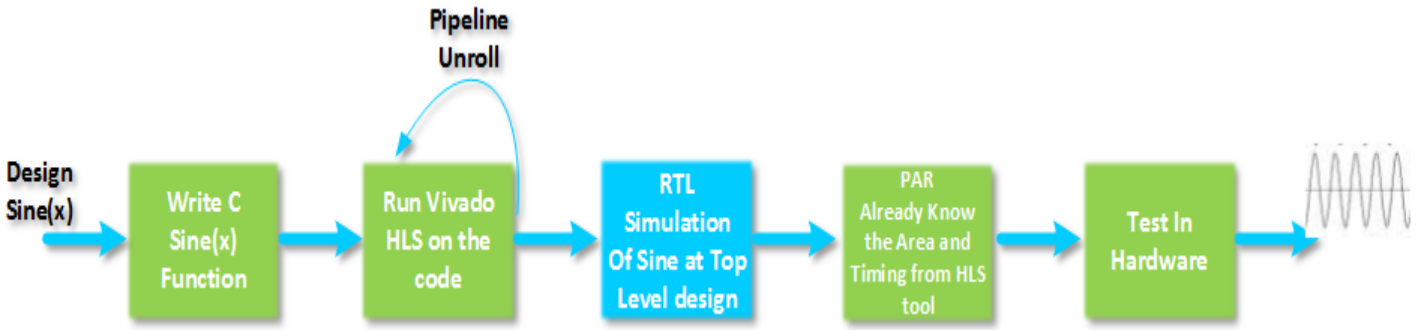
Sine(x) Hand Code Flow



$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots \qquad \text{for all } x$$

- How long would it take for you to hand code a Sine(x) function valid from –pi to pi, 125 MHz clock, floating point , pipelined and latency under 32 clocks?
- Constant churning of RTL simulation just to meet latency and verify math functionality
- Waiting for PAR to finish just to see sizing and actual timing
- Not feasible to hand code RADAR/EW FPGAs anymore giving the enormous densities, FPGA programming need not ASIC skills
  - Error prone
  - Inefficient

Sine(x) HLS Flow



- Sine function latency, area and clock speed are computed in seconds
- Options to unroll, pipeline and other key algorithm optimizations target design requirements
- NO more RTL simulations to verify the Math and Latencies
  - It works - the math works and the boundary conditions are met
  - Only a small set of top level RTL simulations are required to verify the whole design
  - PRI's/CPI's of data are simulated in seconds, not hours or days

Sine(x) The Design – Simple!

top.cpp

```
// sin(x)=x-x^3/3!+x^5/5!-x^7/7!+x^9/9!-x^11/11!-x^13/13!+x^15/15!
// sin(x)=x-x^2/2!+x^4/4!-x^6/6!+x^8/8!-x^10/10!-x^12/12!+x^14/14!

#include "math.h"

void top(float x, float *sinx){
    /*precalculate the 1/factorial terms which will reduce latency and resource as there are no devides*/
    /*1, 1/3!, 1/5!, 1/7!,...*/
    const float facts[6]={-1.6666667e-01,8.3333338e-03,-1.9841270e-04,2.7557319e-06,-2.5052108e-08,1.6059044e-10};

    float x2,x3,x5,x7,x9,x11,x13;


    x2=x*x;
    x3=x2*x;
    x5=x3*x2;
    x7=x5*x2;
    x9=x7*x2;
    x11=x9*x2;
    x13=x11*x2;

    *sinx=x + x3*facts[0] + x5*facts[1] + x7*facts[2] + x9*facts[3] + x11*facts[4] + x13*facts[5];
}
```

Sine(x) Results

Summary of timing analysis

Estimated clock period (ns): 6.09

Summary of overall latency (clock cycles)

Best-case latency:	32
Average-case latency:	32
Worst-case latency:	32
Pipeline initiation interval (II):	1
Pipeline depth:	33

Area Estimates

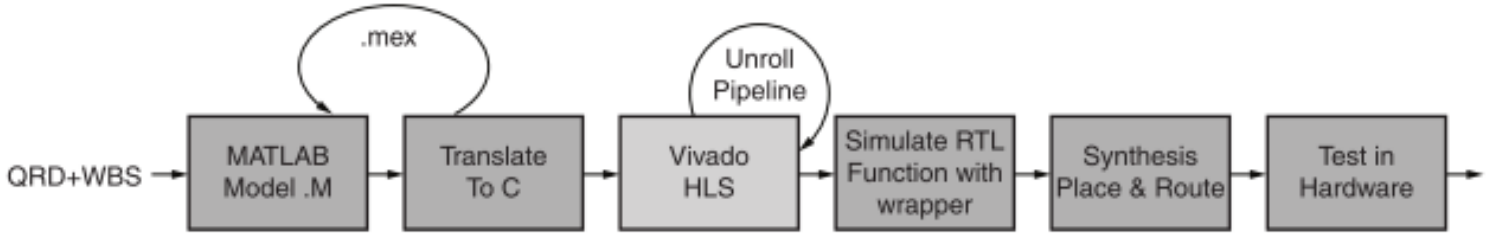
Summary

	BRAM_18K	DSP48E	FF	LUT	SLICE
Component	-	51	3026	3089	-
Expression	-	-	-	-	-
FIFO	-	-	-	-	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	-	-
Register	-	-	641	-	-
ShiftMemory	-	-	0	224	-
Total	0	51	3667	3313	0
Available	1590	1260	728400	364200	91050
Utilization (%)	0	4	~0	~0	0

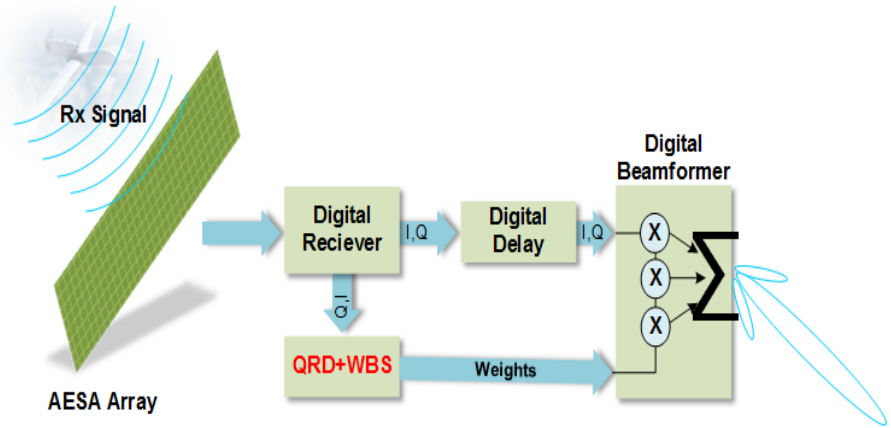
- Took about 5 minutes to design
- This would take more than 4-8 hours depending on experience level…
- 24x improvement in just a Sine function
- The design is pipelined so after 32 clocks, data is streaming out at clock rate

MGS QRD + Weight Back Substitution → STAP

This is the Hardest Floating Point Problem for Hardware




```
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 Real System Design took 6 Months using VHDL
  - Using HLS, it took only 4 hours
- 260x Speed-Up - this does not include system integration which will be faster
- The results are staggering and open's RADAR/EW to a Algorithm Renaissance

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

Zynq A9 : 250ms

- **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

Parameter	Virtex-7 FPGA	ARM-A9
Programmable in C/C++/SystemC:	Y	Y
Flexible I/O	Y	N
HMC/IFSD204b	Y	N
Clock (MHz)	125	667
Latency (ms)	3.3	250
System Power (watts)	75	1,400
System Cost	12.5X lower cost	

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

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**."

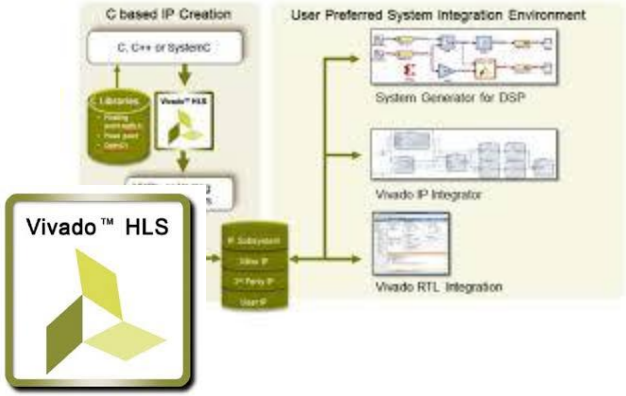
CTO, Major broadcast equipment company

Xilinx's HLS Engine is Creating a Algorithm Renaissance

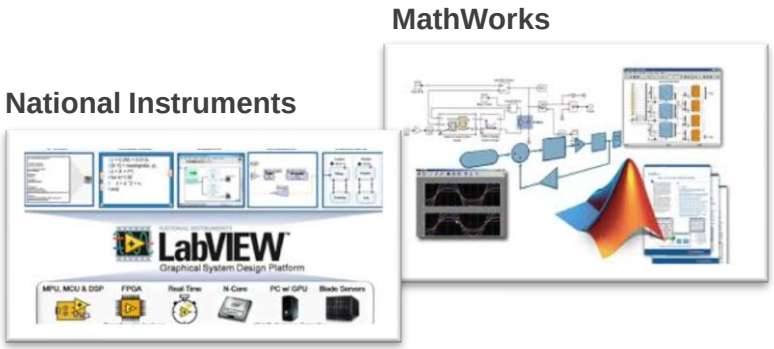
Enabling Smarter Systems

Abstractions

Software Automation

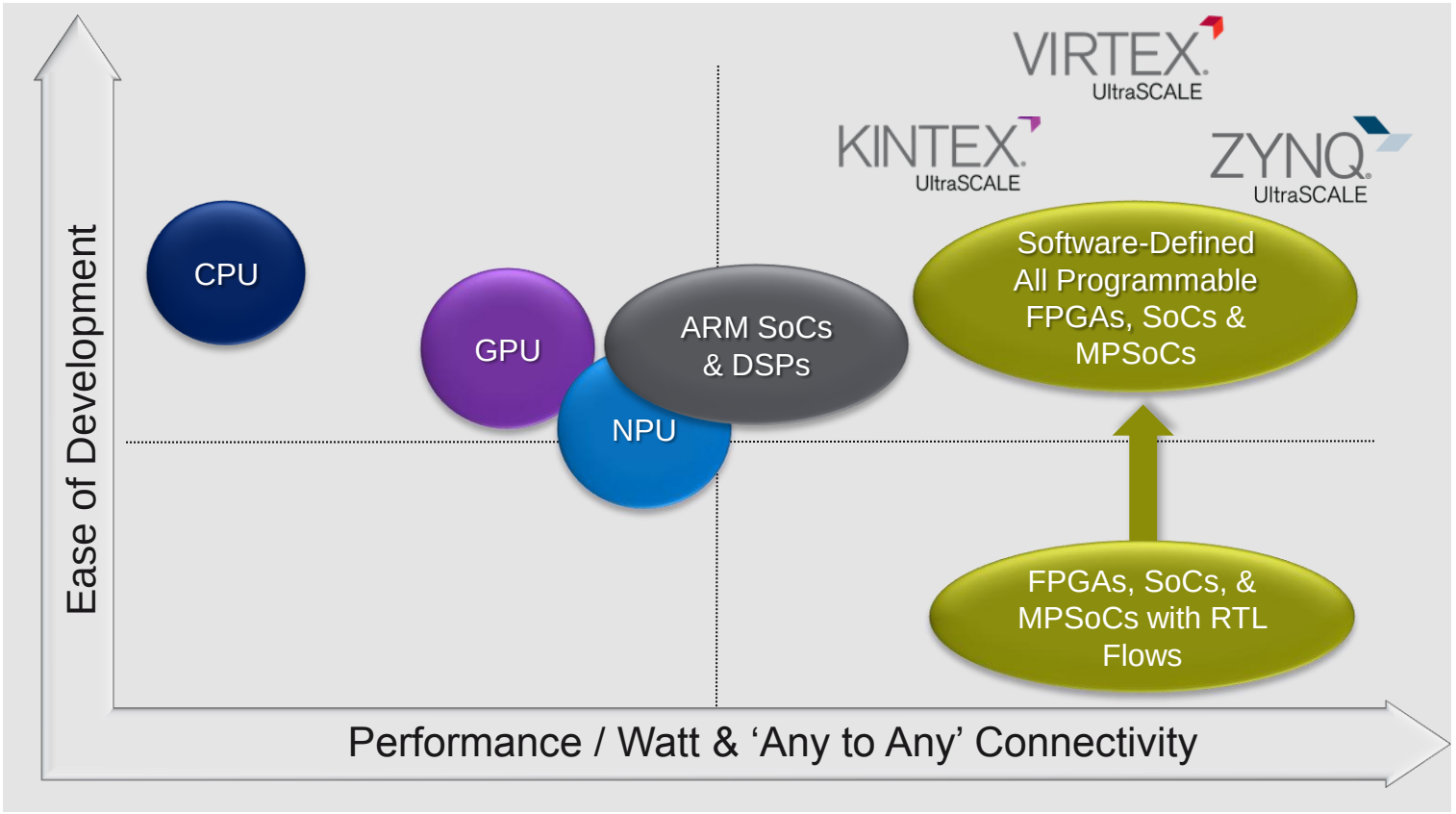


Hardware Automation



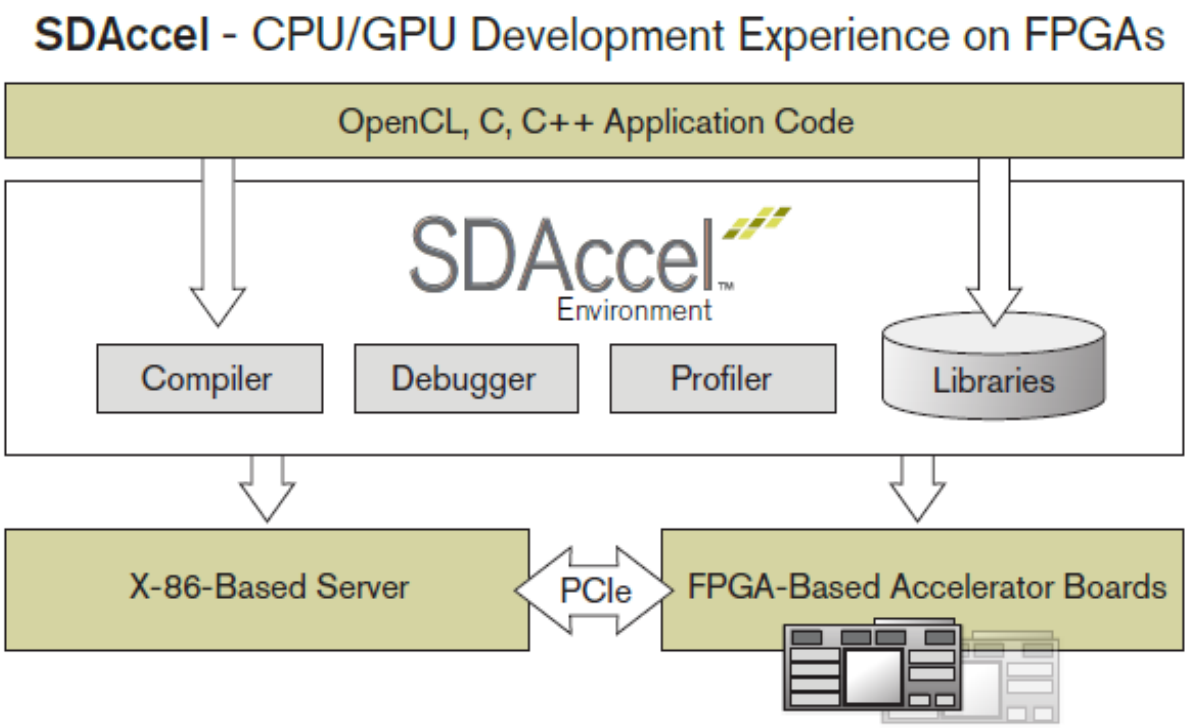
System Automation

Smarter Systems Compute Acceleration Options



Note: Software programmable devices are often paired with FPGAs for connectivity and co-processing

SDAccel: Development Environment for C, C++ and OpenCL



Libraries	Availability
OpenCL built-ins	Included
Video, DSP, Linear Algebra	Included
OpenCV, BLAS, DNN	Auviz Systems
Machine Learning	ArrayFire

Readily Available Boards

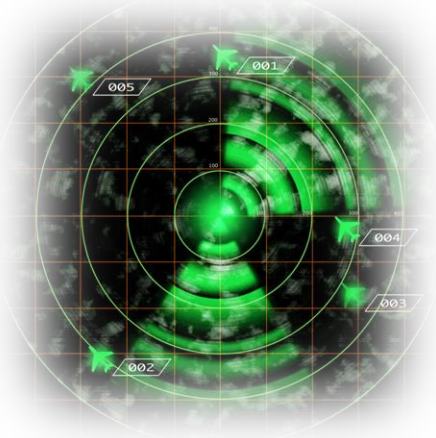
FPGA Optimized Libraries are a Key Advantage

FPGA optimized Libraries built using proven HLS technology

Application	Metric	SDAccel with AuvizCV library	Other FPGA OpenCL	SDAccel Advantage
HD Sobel Filter	Highest frames/ second	650	130	5x faster
HD Image Downscaling	Highest frames/second	465	110	4x faster

Application	Metric	SDAccel with AuvizCV library	Nvidia K20 with CUDA	SDAccel Advantage
HD Sobel Filter	Frames/watt	80	11	7x
HD Image Downscaling	Frames/watt	36	5	7x

GPUs for EW/RADAR ?



- **GPUs will NOT work for EW Systems**
  - Cannot meet sub-micro sec latency
  - Are NOT Deterministic, Are NOT Real-time
  - POOR SEU rates, Xilinx FPGAs 30x Better than GPUs

- **GPUs in 'Theory' Could Work for RADAR with Millisecond PRIs BUT…**
  - GPU + CPU use **200 WATTS**! POOR SWaP
  - Are GPU Vendors committed to High REL and Obsolesce?
  - Are Not 'OPEN' - Xilinx FPGAs are any Interface you want them to be
  - GPUs are inefficient on smaller data sets - think stagger CPIs with varying PRI
  - As shown with Vivado HLS, Xilinx FPGAs float just as easily as fixed point arithmetic
  - FPGAs do it all - DSP, Status, Control and Out of Band Processing

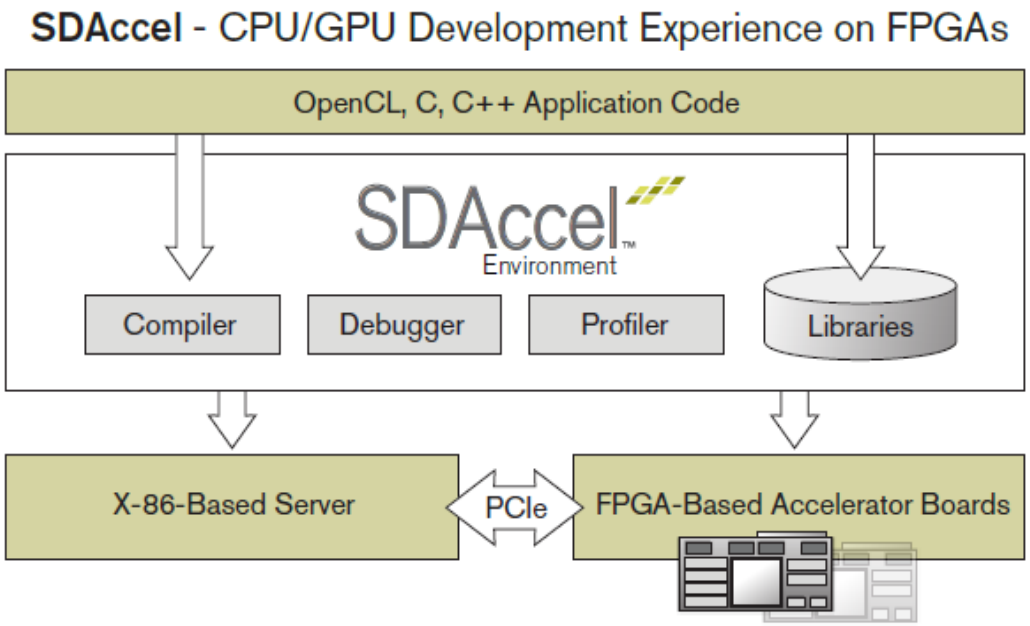
*See Neutron-Induced Multiple Output Errors: a Reality on GPUs

Xilinx Series FPGAs are the safe, efficient choice over GPUs

SDAccel Ecosystem

Application Developers

Auviz Systems | ArrayFire | impulse accelerated technologies | 聯科集團 CLUSTERTECH | ICHEC Irish Centre for High-End Computing | Instigate Design | MULTICORE WARE



Library Providers

Auviz Systems

ArrayFire

COTS Board Partners

ALPHA DATA | pico computing is now Micro | 4DSP

Call To Action

- **Review [WP452] to see how HLS accelerates design for RADAR Adaptive Beamforming**

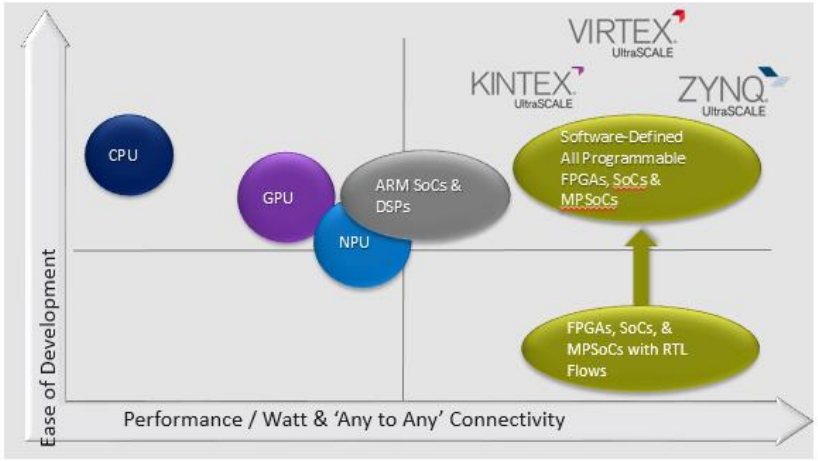
  http://www.xilinx.com/support/documentation/white_papers/wp452-adaptive-beamforming.pdf

- **Start using [Vivado HLS] to accelerate RADAR Electronic Warfare Algorithms**

  http://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html

- **Review [SDAccel Development Environment Backgrounder]**

  http://www.xilinx.com/publications/prod_mktg/sdx/sdaccel-backgrounder.pdf



Note: Software programmable devices often paired with FPGA for connectivity and co-processing