Microsemi

a Microchip company

SoC FPGAs: Opportunities and Challenges

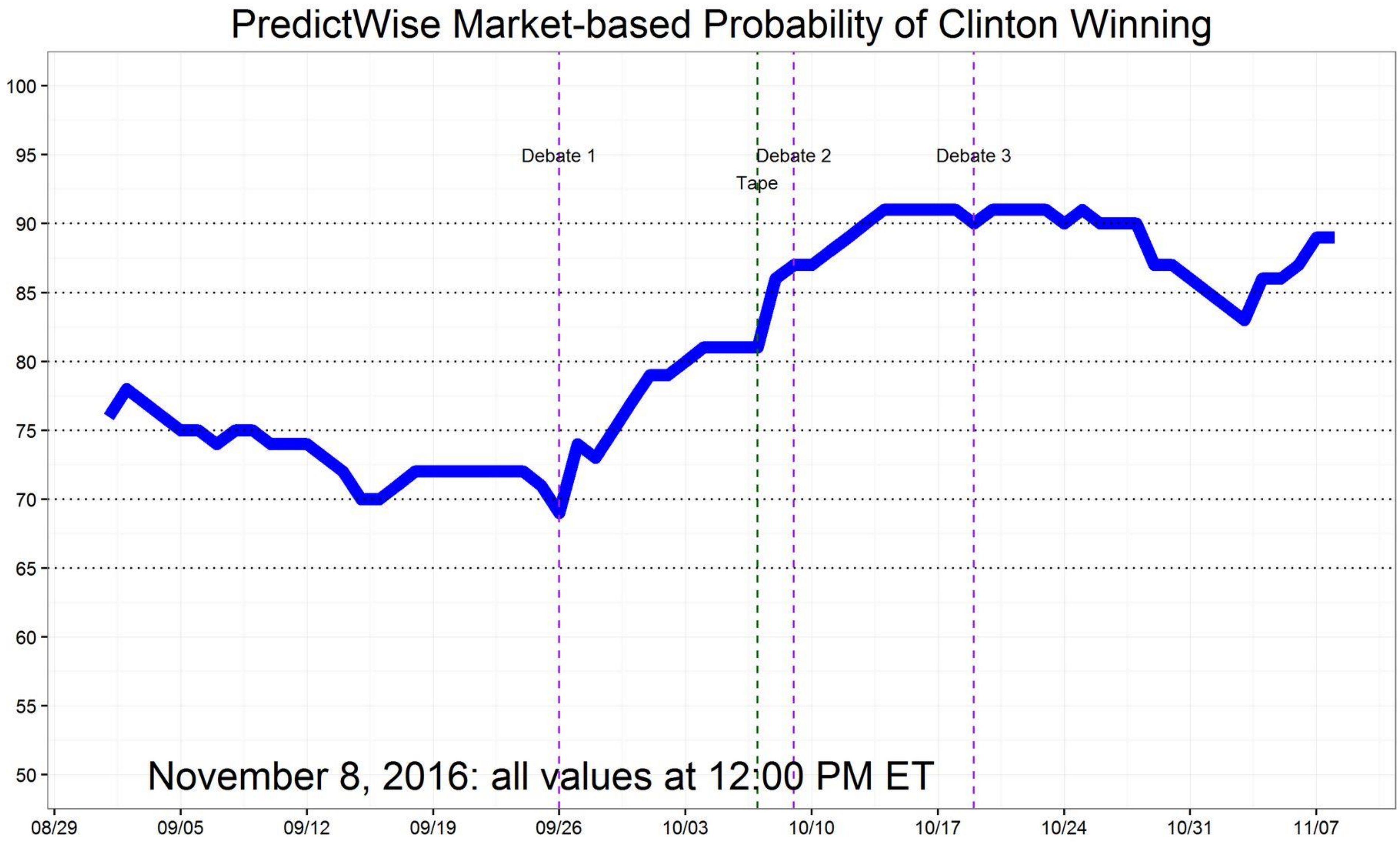
Advanced Workshop on Modern FPGA Based Technology for Scientific Computing

Nizar Abdallah

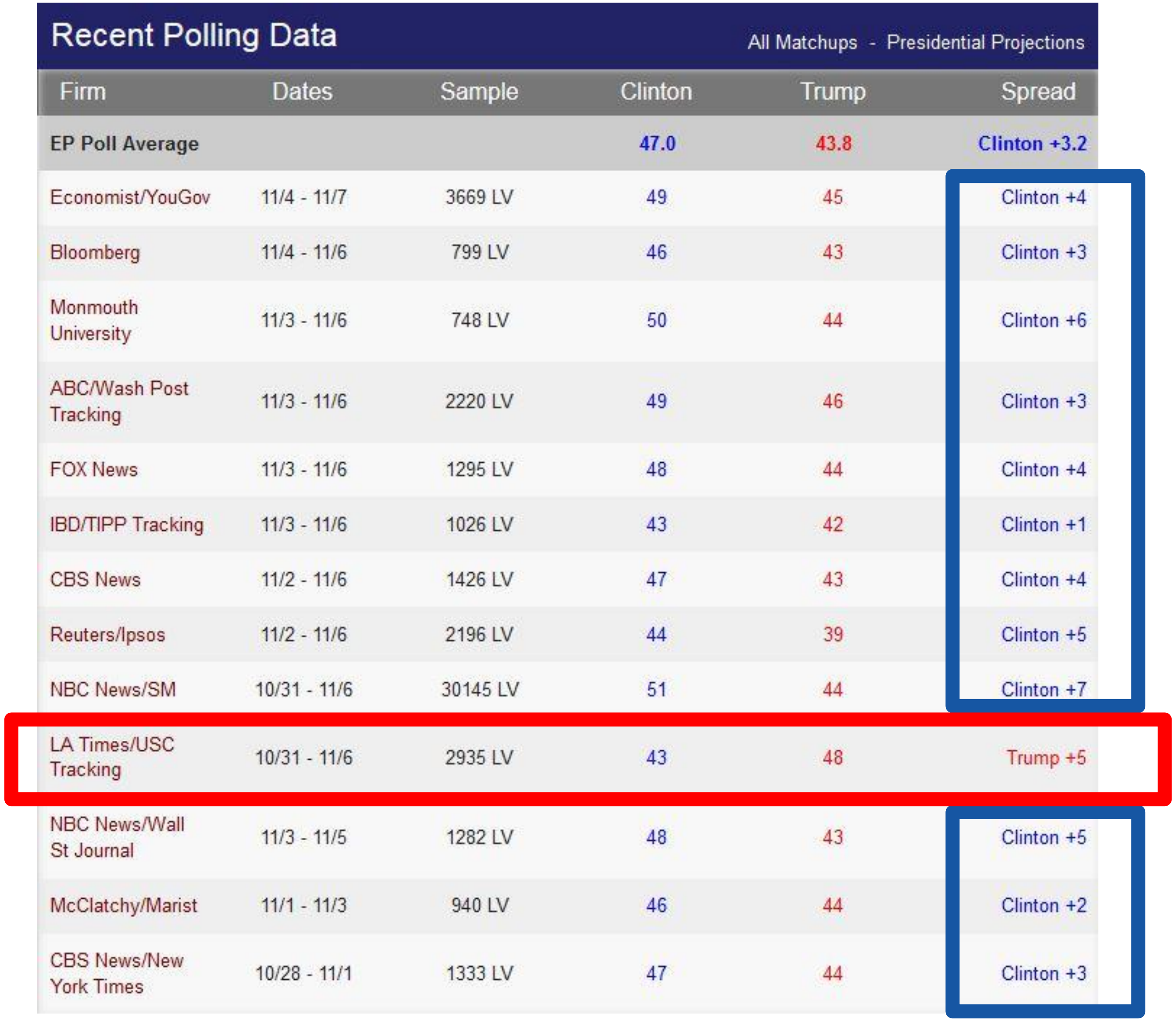
May 2019

- A short bio…
- What will we be talking about?

Disclaimer–On Our Ability to Predict the Future



Disclaimer–On Our Ability to Predict the Future



Recent Polling Data

All Matchups - Presidential Projections

Firm	Dates	Sample	Clinton	Trump	Spread
EP Poll Average			**47.0**	**43.8**	**Clinton +3.2**
Economist/YouGov	11/4 - 11/7	3669 LV	49	45	Clinton +4
Bloomberg	11/4 - 11/6	799 LV	46	43	Clinton +3
Monmouth University	11/3 - 11/6	748 LV	50	44	Clinton +6
ABC/Wash Post Tracking	11/3 - 11/6	2220 LV	49	46	Clinton +3
FOX News	11/3 - 11/6	1295 LV	48	44	Clinton +4
IBD/TIPP Tracking	11/3 - 11/6	1026 LV	43	42	Clinton +1
CBS News	11/2 - 11/6	1426 LV	47	43	Clinton +4
Reuters/Ipsos	11/2 - 11/6	2196 LV	44	39	Clinton +5
NBC News/SM	10/31 - 11/6	30145 LV	51	44	Clinton +7
LA Times/USC Tracking	10/31 - 11/6	2935 LV	43	48	Trump +5
NBC News/Wall St Journal	11/3 - 11/5	1282 LV	48	43	Clinton +5
McClatchy/Marist	11/1 - 11/3	940 LV	46	44	Clinton +2
CBS News/New York Times	10/28 - 11/1	1333 LV	47	44	Clinton +3

Introduction

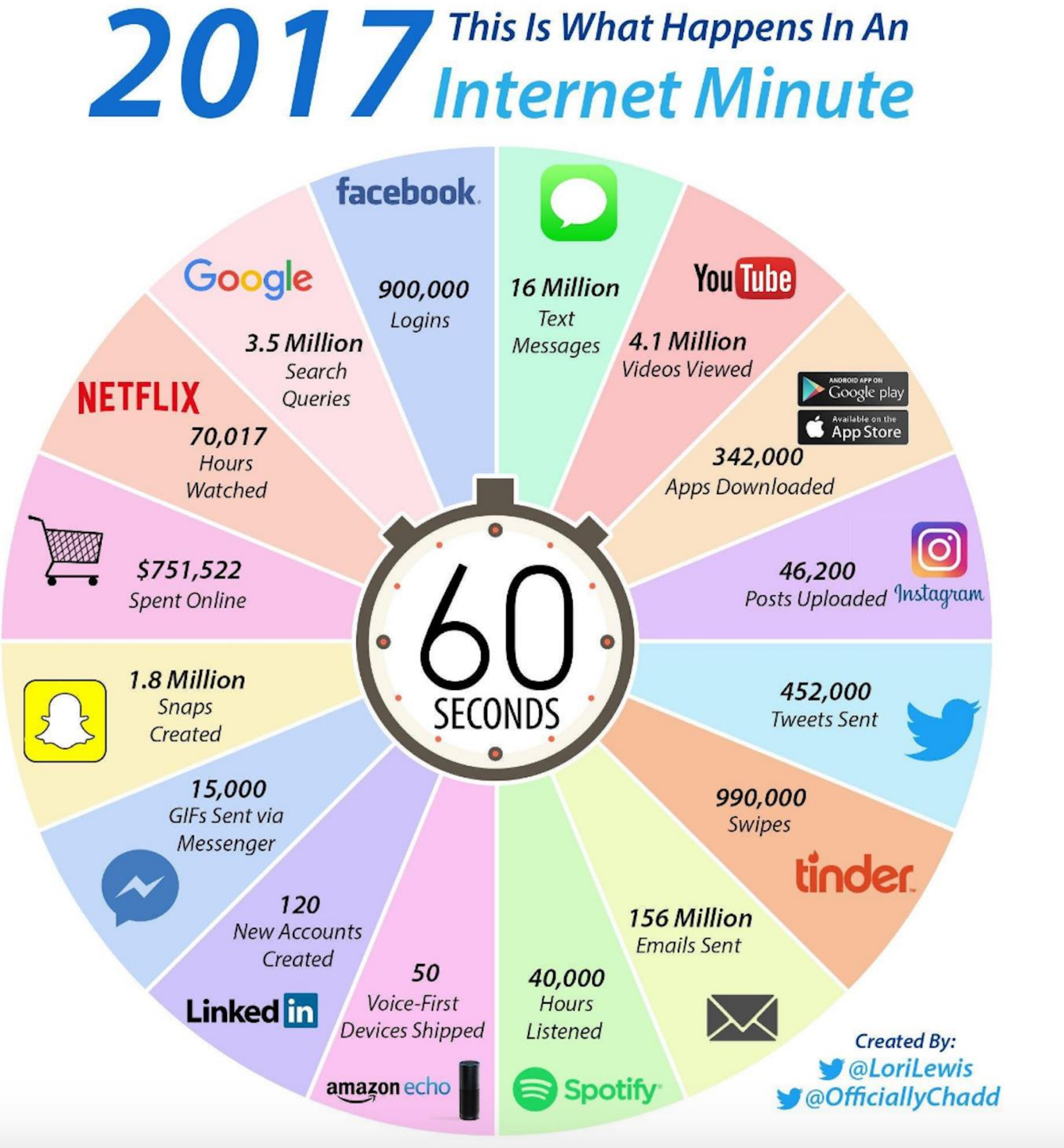
JAN 2017

GLOBAL DIGITAL SNAPSHOT

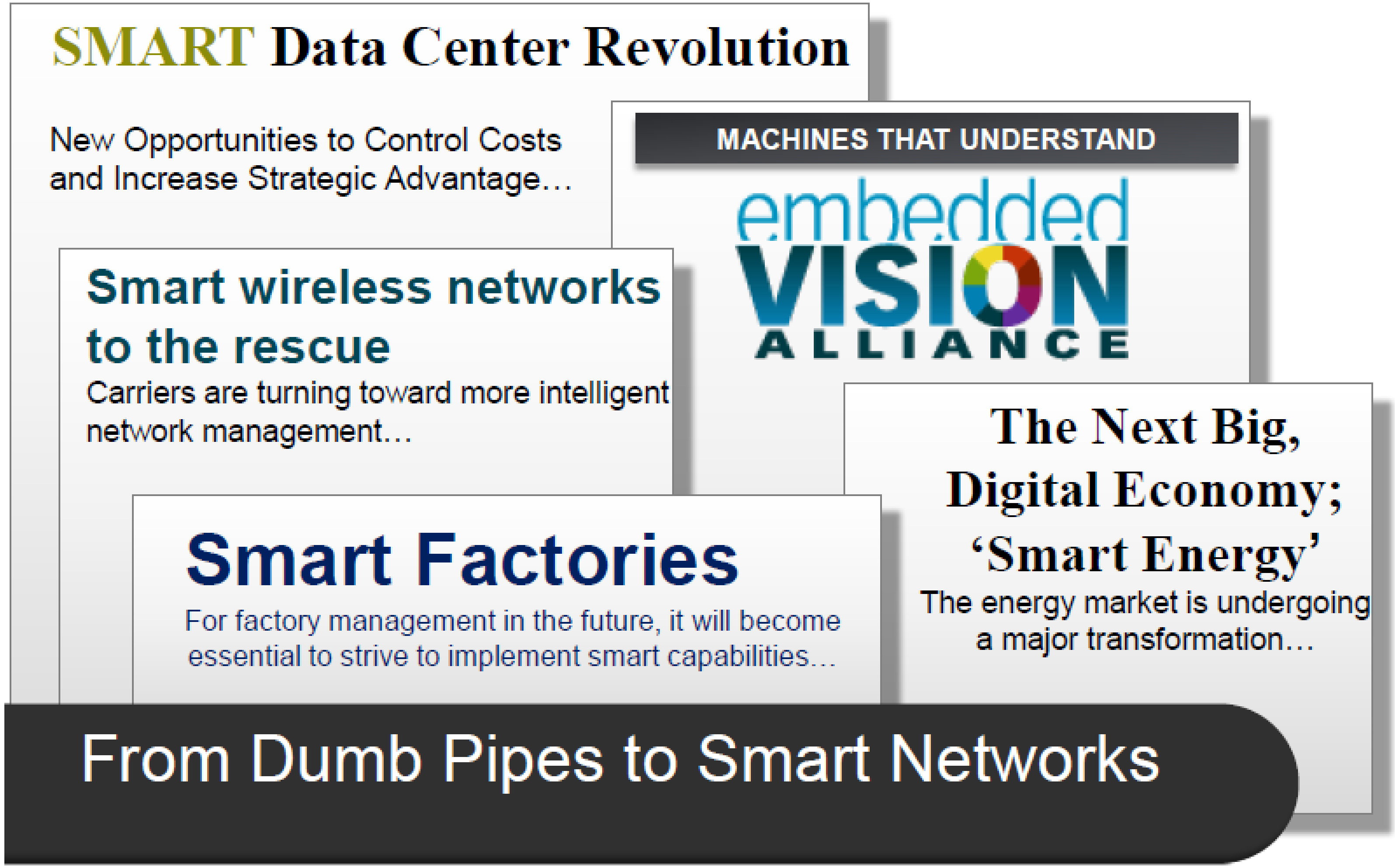
KEY STATISTICAL INDICATORS FOR THE WORLD'S INTERNET, MOBILE, AND SOCIAL MEDIA USERS

TOTAL POPULATION	INTERNET USERS	ACTIVE SOCIAL MEDIA USERS	UNIQUE MOBILE USERS	ACTIVE MOBILE SOCIAL USERS
7.476 BILLION	3.773 BILLION	2.789 BILLION	4.917 BILLION	2.549 BILLION
URBANISATION: 54%	PENETRATION: 50%	PENETRATION: 37%	PENETRATION: 66%	PENETRATION: 34%

SOURCES: POPULATION: UNITED NATIONS; U.S. CENSUS BUREAU; INTERNET: INTERNETWORLDSTATS; ITU; INTERNETLIVESTATS; CIA WORLD FACTBOOK; FACEBOOK; NATIONAL REGULATORY AUTHORITIES; SOCIAL MEDIA AND MOBILE SOCIAL MEDIA: FACEBOOK; TENCENT; VKONTAKTE; LIVEINTERNET.RU; KAKAO; NAVER; NIKI AGHAEI; CAFEBAZAAR.IR; SIMILARWEB; DING; EXTRAPOLATION OF TNS DATA; MOBILE: GSMA INTELLIGENCE; EXTRAPOLATION OF EMARKETER AND ERICSSON DATA.


2017 This Is What Happens In An Internet Minute
facebook
900,000 Logins
16 Million Text Messages
YouTube
4.1 Million Videos Viewed
Google play
App Store
342,000 Apps Downloaded
46,200 Posts Uploaded
Instagram
452,000 Tweets Sent
990,000 Swipes
tinder
156 Million Emails Sent
40,000 Hours Listened
Spotify
50 Voice-First Devices Shipped
amazon echo
120 New Accounts Created
LinkedIn
15,000 GIFs Sent via Messenger
1.8 Million Snaps Created
$751,522 Spent Online
NETFLIX
70,017 Hours Watched
Google
3.5 Million Search Queries
60 SECONDS
Created By:
@LoriLewis
@OfficiallyChadd

More Intelligence in Every System



SMART Data Center Revolution

New Opportunities to Control Costs and Increase Strategic Advantage…

MACHINES THAT UNDERSTAND

embedded VISION ALLIANCE

Smart wireless networks to the rescue

Carriers are turning toward more intelligent network management…

The Next Big, Digital Economy; ‘Smart Energy’

The energy market is undergoing a major transformation…

Smart Factories

For factory management in the future, it will become essential to strive to implement smart capabilities…

From Dumb Pipes to Smart Networks

Society’s Top Challenges and Trends

- IoT => distributed intelligence & interconnectivity
- Data explosion, processing (AI & ML) => computation, response time

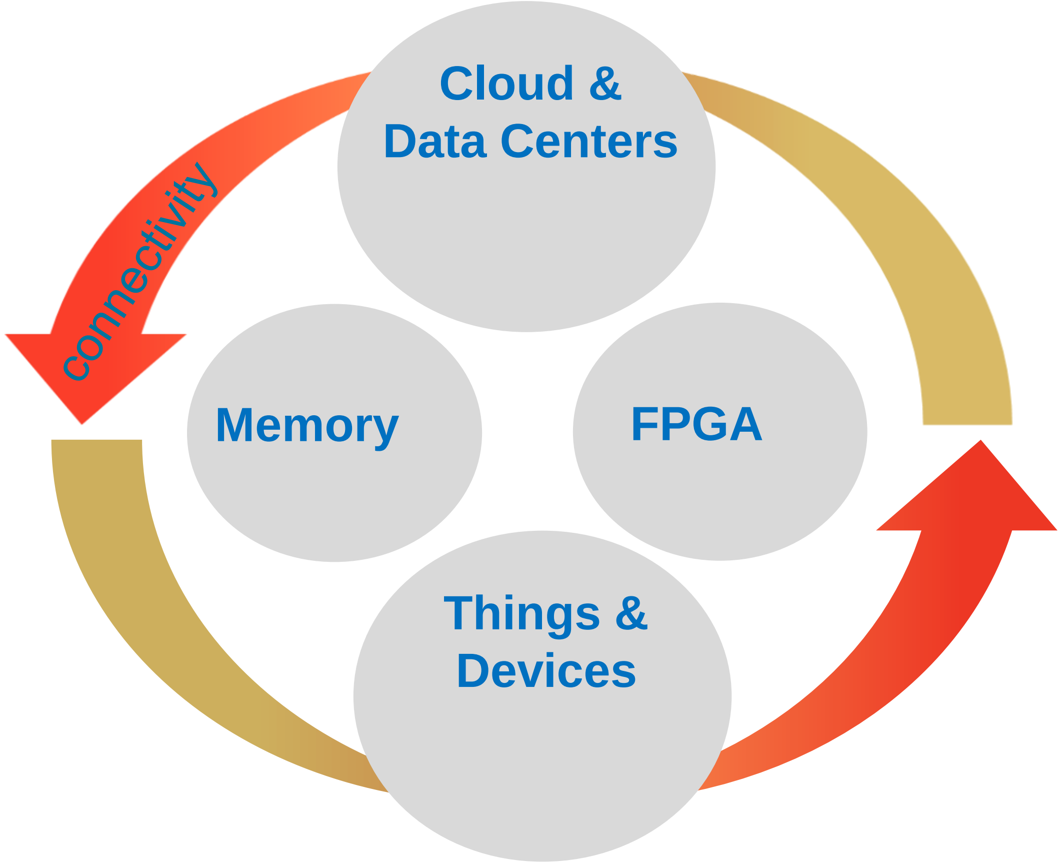
- Energy needs => decentralized processing at end points + reconfigurability
- Environmental sustainability (product as a service) => cloud storage, response time, adaptability

- Social sustainability => cost & access
- Security => response time
- Well being (robotics, medical, comfort) => response time, adaptability, portability

- Theoretical research: space exploration => reliability

At the Center of the Digital Transformation

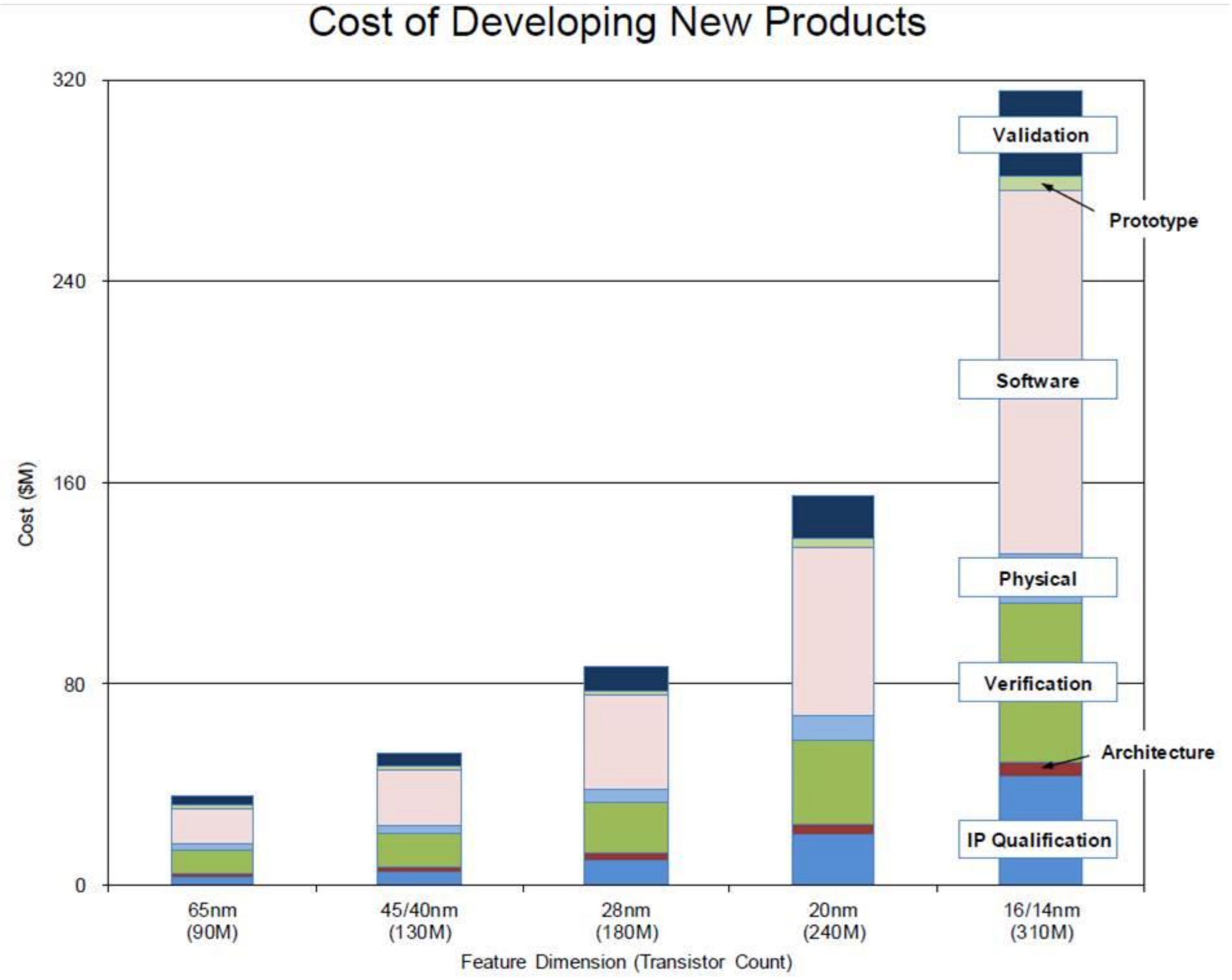
- Everything we do has a digital component
- Push to the cloud
- Run analytics
- Push back to the source



Inspired from Intel CEO Brian Krzanich, IDF 2016

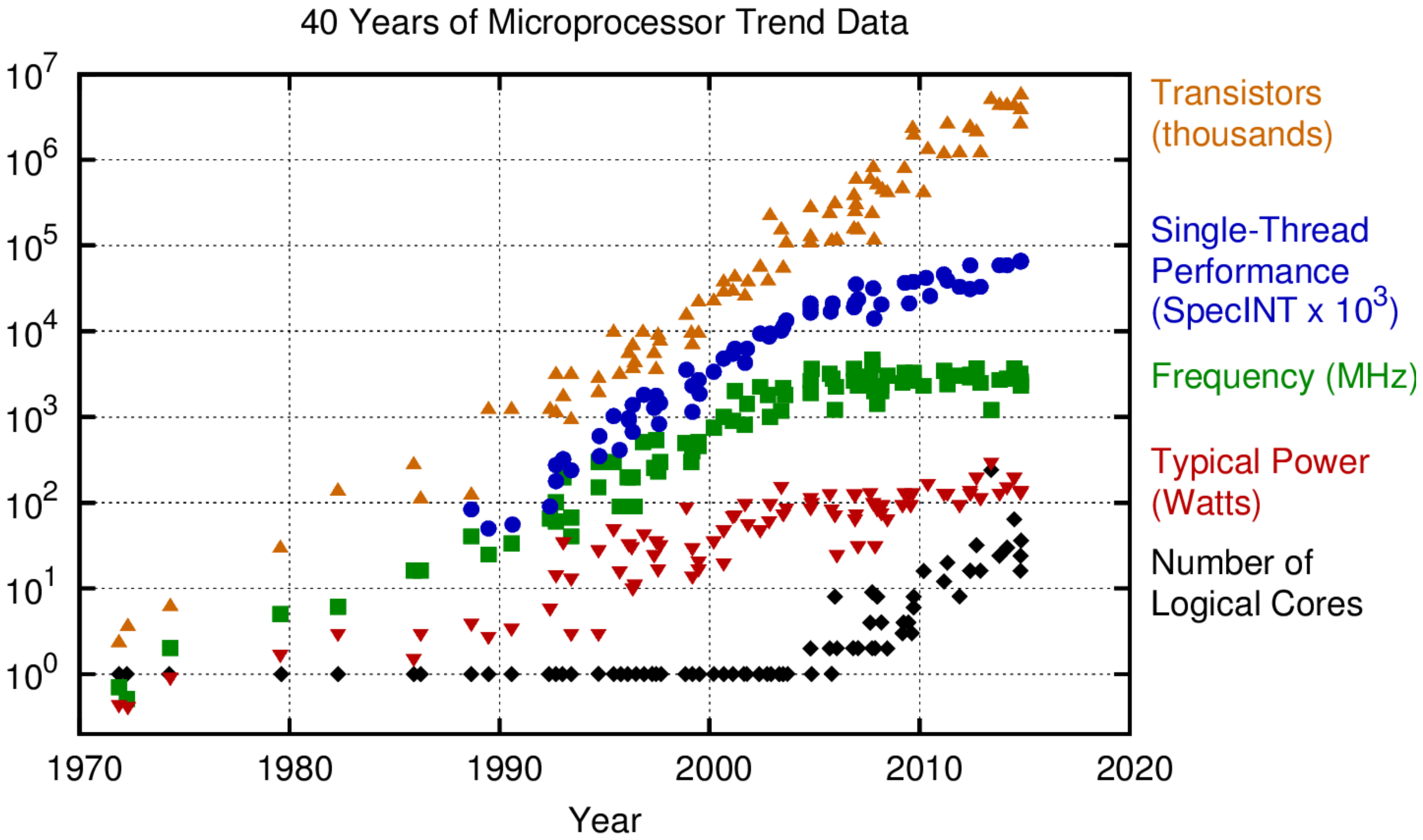
Design Cost

- Up to $300M for a new SoC
- Between $20M - $50M with reuse



Source: IBS

Performance Improvement Trend



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Key Requirements

Cost + Speed + Power + Flexibility

↓

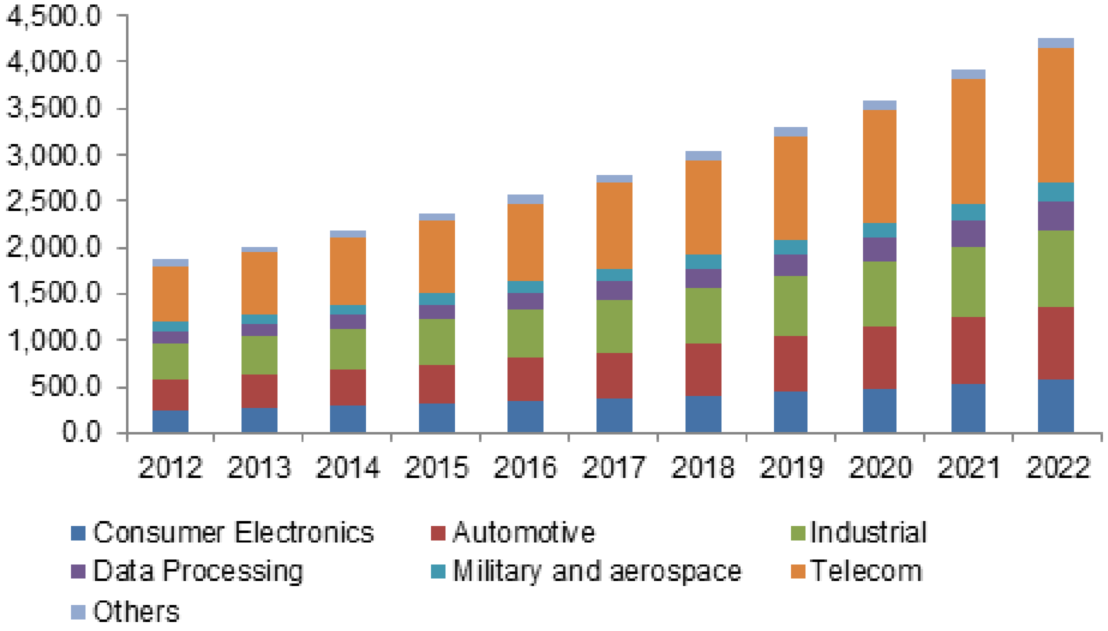
Integration

[Re] Programmability

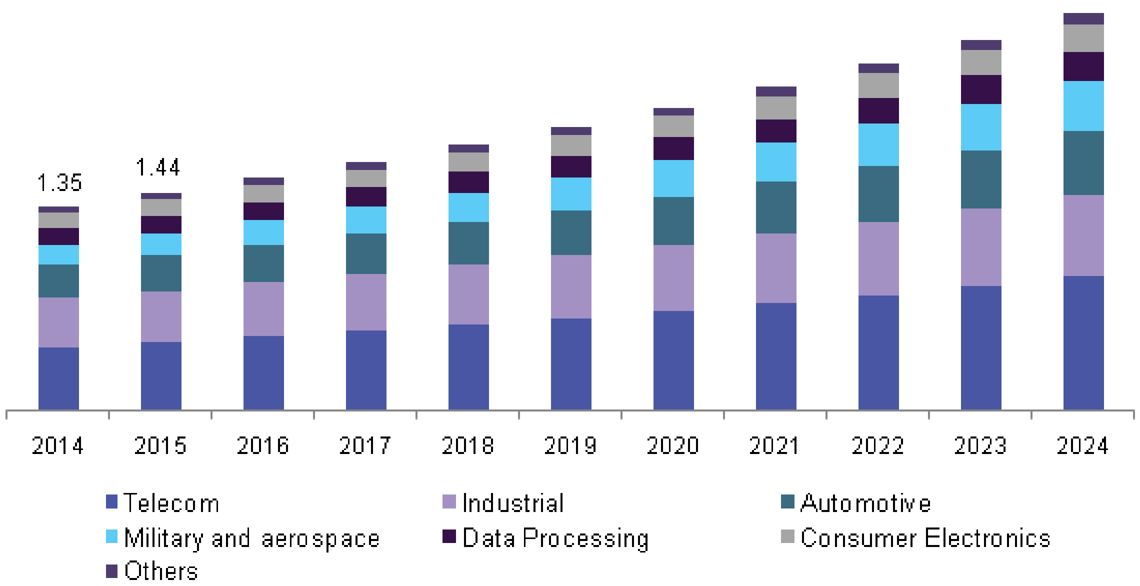
↓

FPGA SoC

FPGA Economics



Asia-Pacific FPGA market growth. Source: Global Market Insights

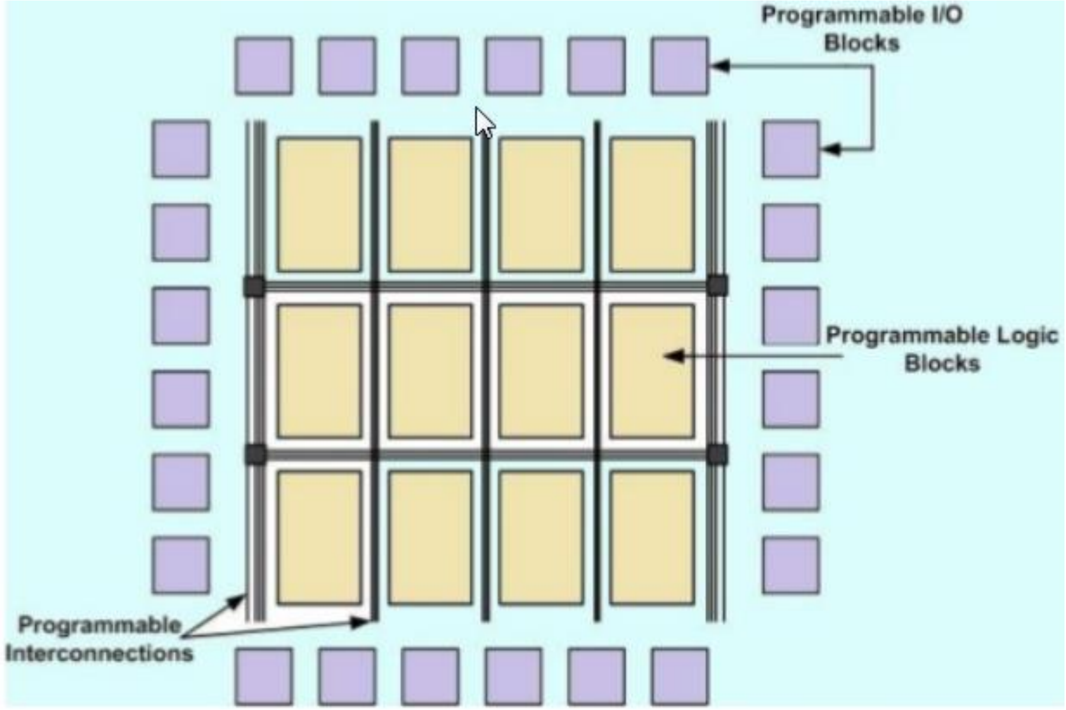


U.S. FPGA market Growth. Source: Grand View Research

FPGA Basics

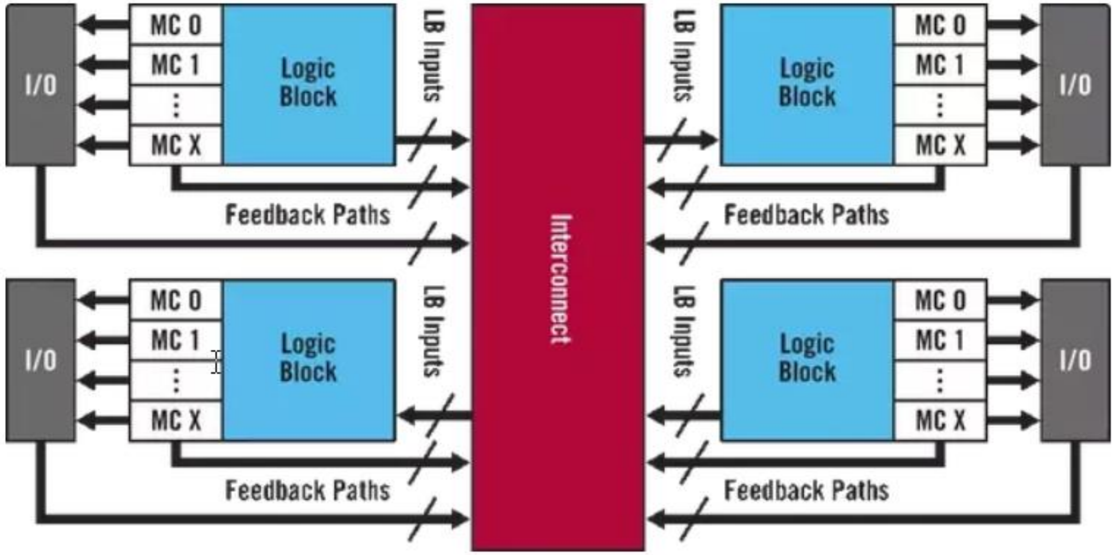
Programmable Logic Basics

Field Programmable Gate Array (FPGA)



- Flexible architecture with virtually no restrictions – any pin to any logic block. LUT4 + FF = an LE
- Large Density, High pin count – SRAM based or Flash
- Highly integrated with memory, DSP, transceivers
- Variable timing dependent on routing

Complex Programmable Logic Device (CPLD)



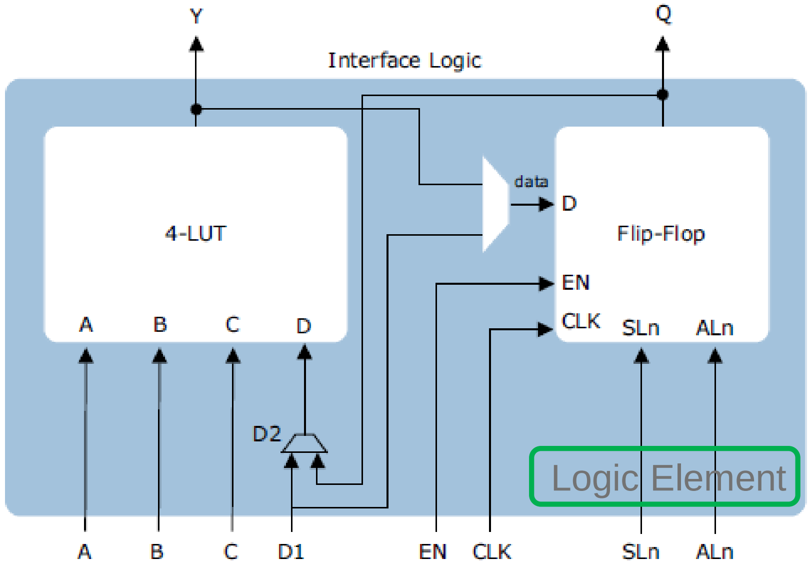
- Less flexible architecture with routing limitations – uses Macrocell blocks to build functions
- Low density EEPROM based but with fixed timing
- Great for simple logic functions
- No investment being made in this technology

Why Did FPGAs Become So Popular?

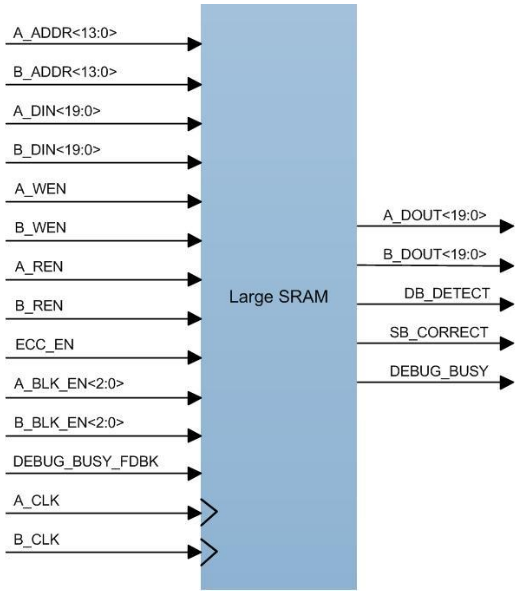
- FPGAs are “field” re-programmable which gives them many advantages over traditional custom ASICs
  - Great deal of application flexibility
  - Quick time to market
  - Low risk
  - Field upgrade-ability
- They are much higher density and have a more flexible architecture than CPLDs
- FPGAs use either SRAM memory or Flash memory to provide their re-programmability
- The ability to conceive of a logic function and realize that function in silicon within weeks or even days was a revolutionary concept that fundamentally changed how digital systems are developed today
- Furthermore, the capability of being able to upload a “bug-fix” to a fielded FPGA-based product was something that designers had only dreamed about
  - Using an internet connection to the fielded product, a configuration flash memory can be programmed and then the FPGA instructed to load the new image
- With the great growth of the FPGA market over the last 20 years (now at ~$5.5B), ASICs have been relegated to only very high volume applications

FPGA Architecture

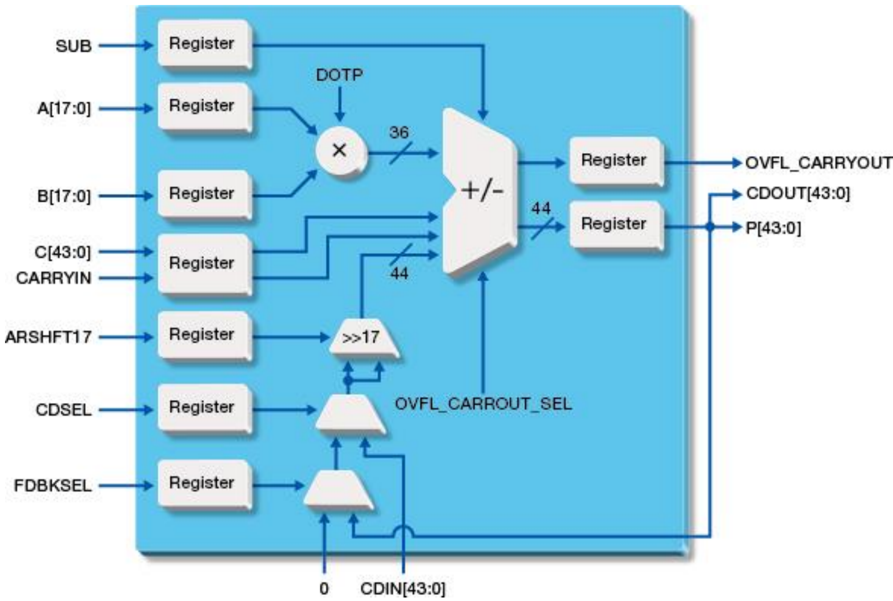
4-Input Look-up-table (LUT4) plus FF



20Kbit Large SRAM (LSRAM)

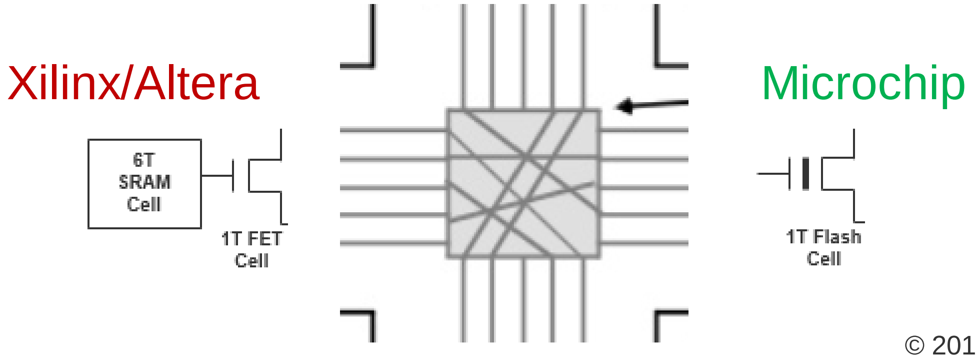


18x18 Multiply Math Block



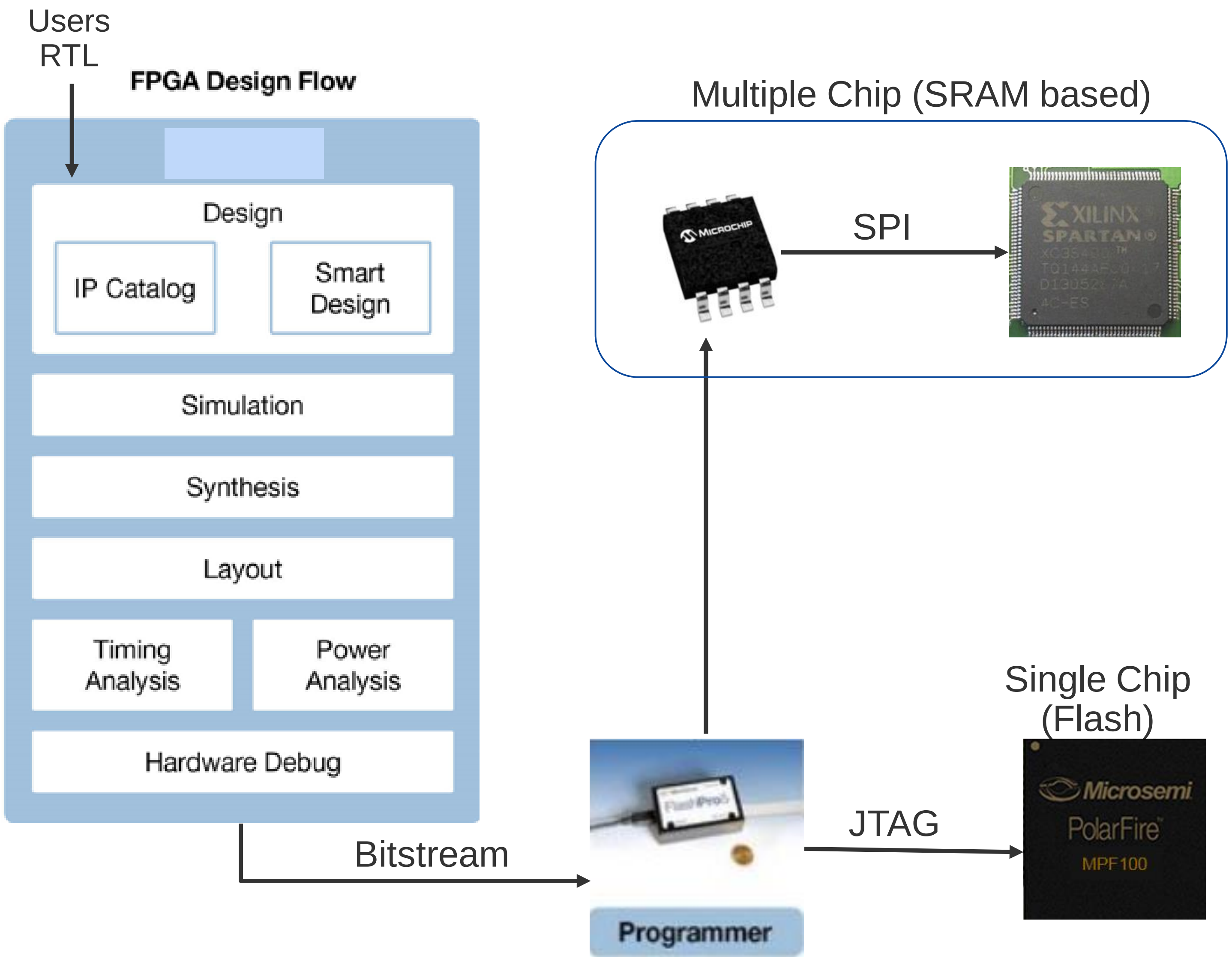
- All FPGAs have some combination of these building blocks and switch matrices
- Other blocks that are becoming more popular within FPGAs are
  - Transceiver Blocks from 5G to 25G+
  - Analog PLLs, crypto processors, DDR controllers, ADCs
  - Embedded processors (Arm and RISCV)

Routing Switch Matrix



How Do FPGAs Work?

- The designer describes his design in
  - RTL (VHDL or Verilog)
  - Uses IP blocks (UART or SPI for example)
- Simulates the RTL using Modelsim
- Synthesizes the design using Synplify from Synopsys
  - This converts the RTL into an edif netlist
- After place and route, timing and power analysis is performed
- Programming takes place using a USB dongle to
  - Program a SPI flash or NAND flash for Xilinx and Altera
    - Once the chip is told to re-program, it “re-boots” itself with the new image
  - Program our FPGA directly since the FPGA is flash based (no “boot” prom needed)



Users RTL

FPGA Design Flow

Design

IP Catalog

Smart Design

Simulation

Synthesis

Layout

Timing Analysis

Power Analysis

Hardware Debug

Bitstream

Programmer

Multiple Chip (SRAM based)

SPI

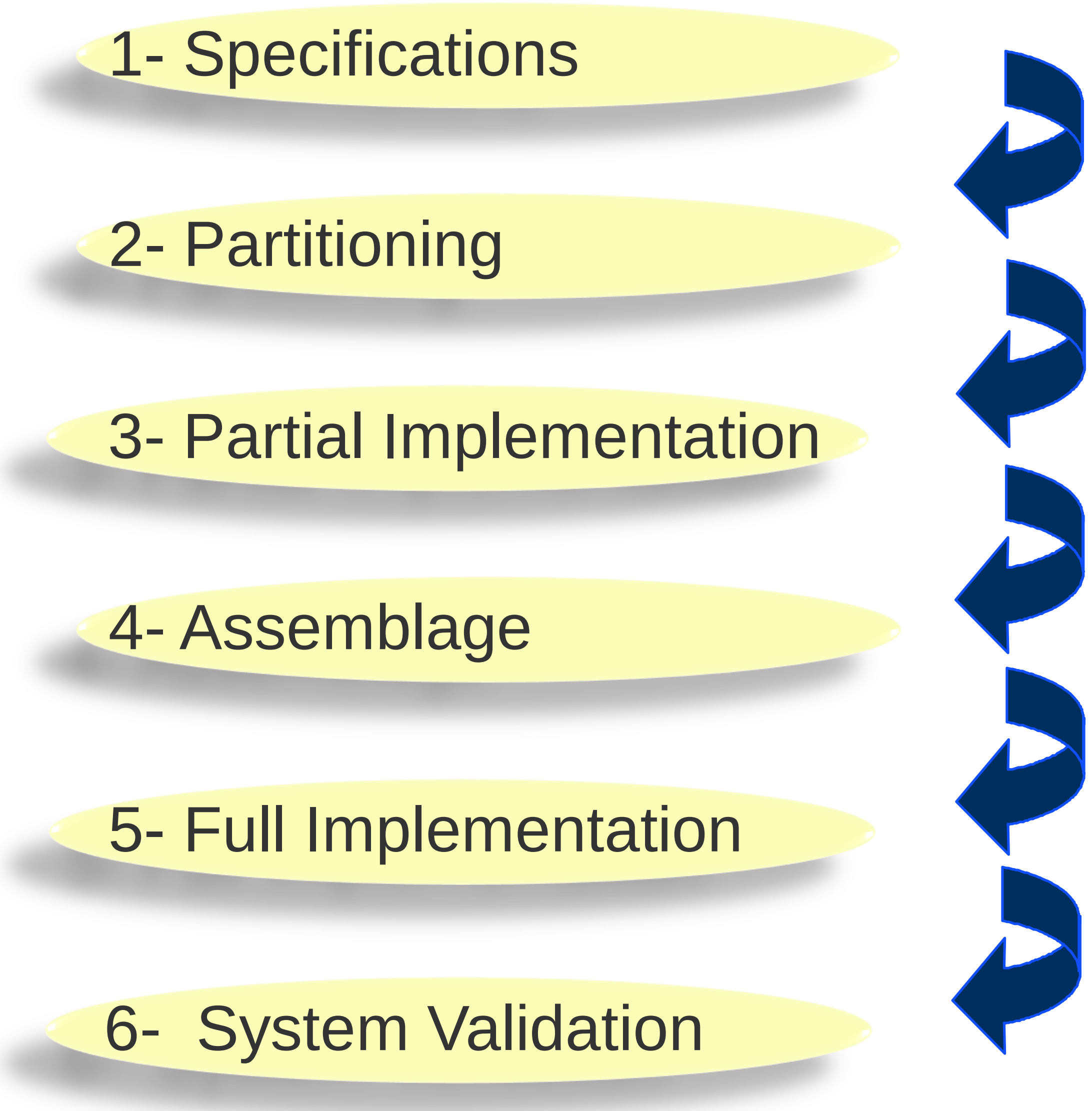
JTAG

Single Chip (Flash)

[FPGA] Design Methodology

- Top-Down design methodology

Critical Importance



1- Specifications

2- Partitioning

3- Partial Implementation

4- Assemblage

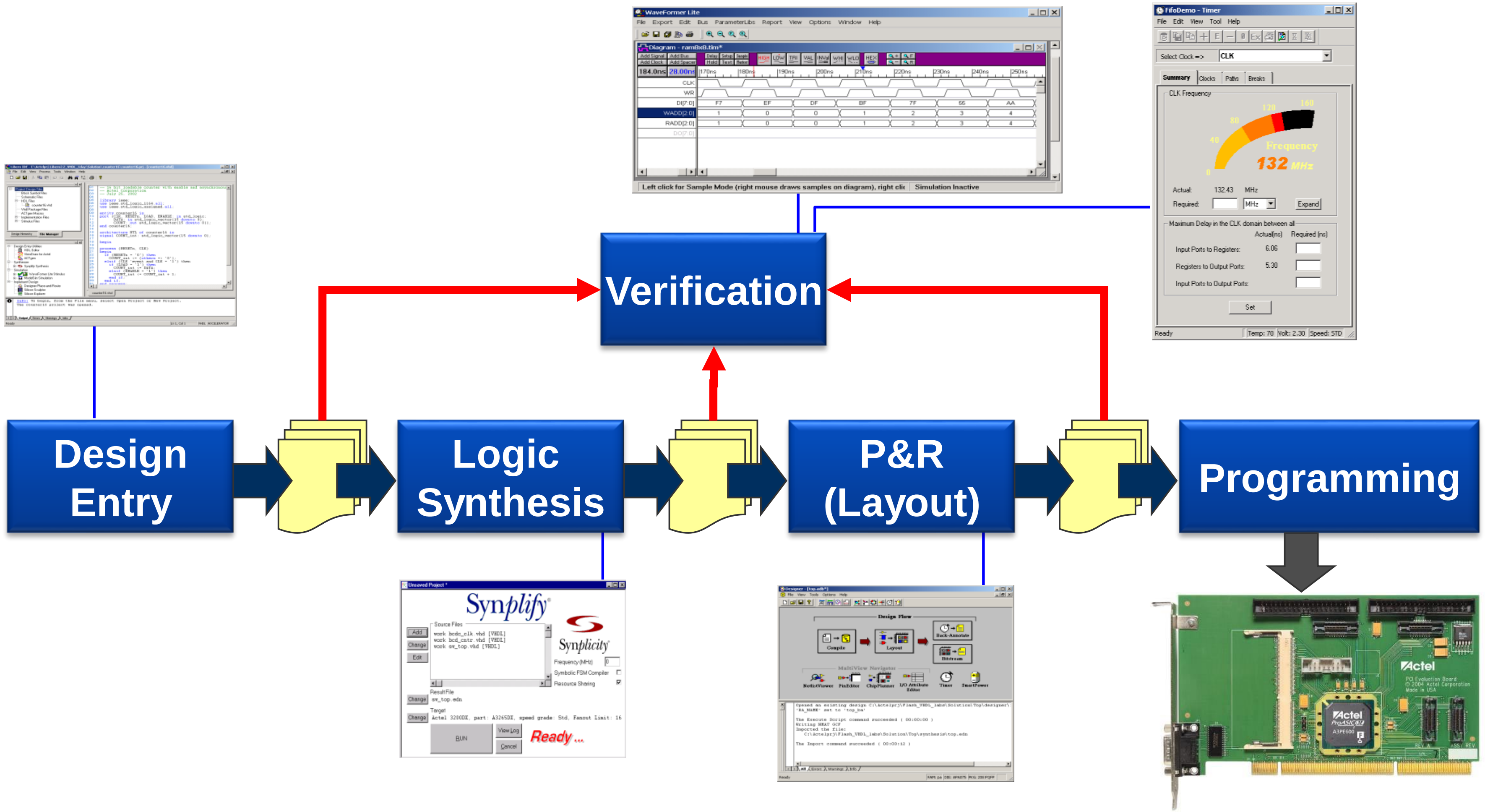
5- Full Implementation

6- System Validation

FPGA Implementation Flow

- Simplified FPGA design implementation flow

Critical Importance



System Validation Flow

- Simulation
  - Bus Functional Model (BFM)
  - Mixed language HDL simulation

- Hardware Prototype for system validation

Remember Barto's Law

“Every circuit is considered guilty until proven innocent”

Example of a FPGA Family Table (PolarFire)

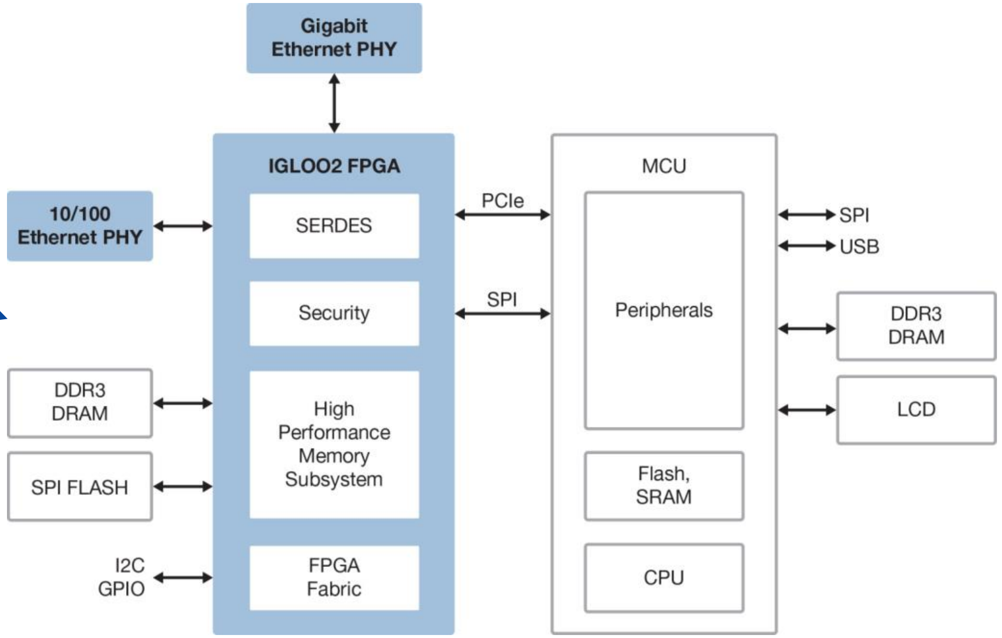
	Features	PolarFire FPGA			
		MPF100	MPF200	MPF300	MPF500
FPGA Fabric	Logic Elements (4LUT + DFF)	109K	192K	300K	481K
	Math Blocks (18x18 MACC)	336	588	924	1480
	LSRAM Blocks (20 kbit)	352	616	952	1520
	uSRAM Blocks (64x12)	1008	1764	2772	4440
	Total RAM (Mbits)	7.6 Mbits	13.3 Mbits	20.6 Mbits	33 Mbits
	uPROM (kbits)	297 Kbits	297 Kbits	459 Kbits	513 Kbits
	User DLL's/PLL's	8 each	8 each	8 each	8 each
High Speed I/O	250 Mbps -12.7 Gbps Transceiver Lanes	8	16	16	24
	PCIe Gen2 Endpoints/Root Ports	2	2	2	2
Total I/O	Total User I/O	296	364	512	584
Packaging	Type / Size / Pitch	Total User I/O (HSIO / GPIO) GPIO CDRs / XCVRs			
	FCSG325 (11x11, 11x14.5*, 0.5 mm)	170(84/86) 8/4	🟩 170(84/86) 8/4*		
	FCSG536 (16x16, 0.5 mm)		300(120/180) 15/4	🟩 300(120/180) 15/4	
	FCVG484 (19x19, 0.8 mm)	284(120/164) 14/4	284(120/164)14/4	🟩 284(120/164) 14/4	
	FCG484 (23x23, 1.0 mm)	244(96/148) 13/8	244(96/148) 13/8	🟩 244(96/148) 13/8	
	FCG784 (29x29, 1.0 mm)		364(132/232) 20/16	388(156/232) 20/16	🟩 388(156/232) 20/16
	FCG1152 (35x35, 1.0 mm)			512(276/236) 24/16	🟩 584(324/260) 24/24
		Devices in the same package and family type are pin compatible			

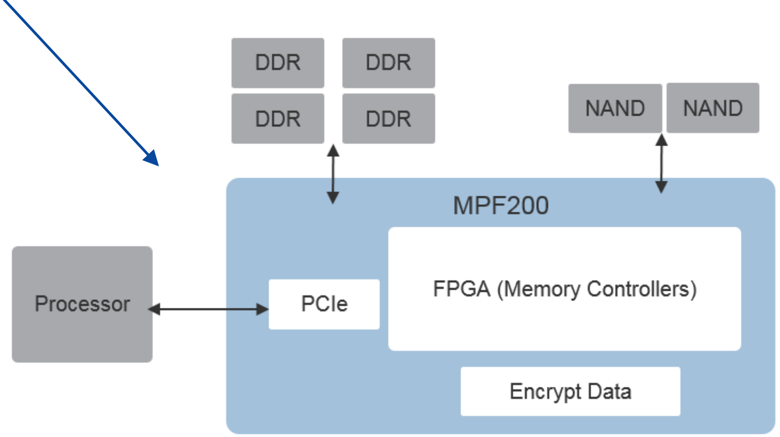
Extended Commercial (0ºC-100ºC) and Industrial (-40ºC-100ºC) Temperature Support for all Die Package Combinations - RoHS only

🟩 ***Additional Temp Grade: Military (-55ºC-125ºC) - Leaded packages only***

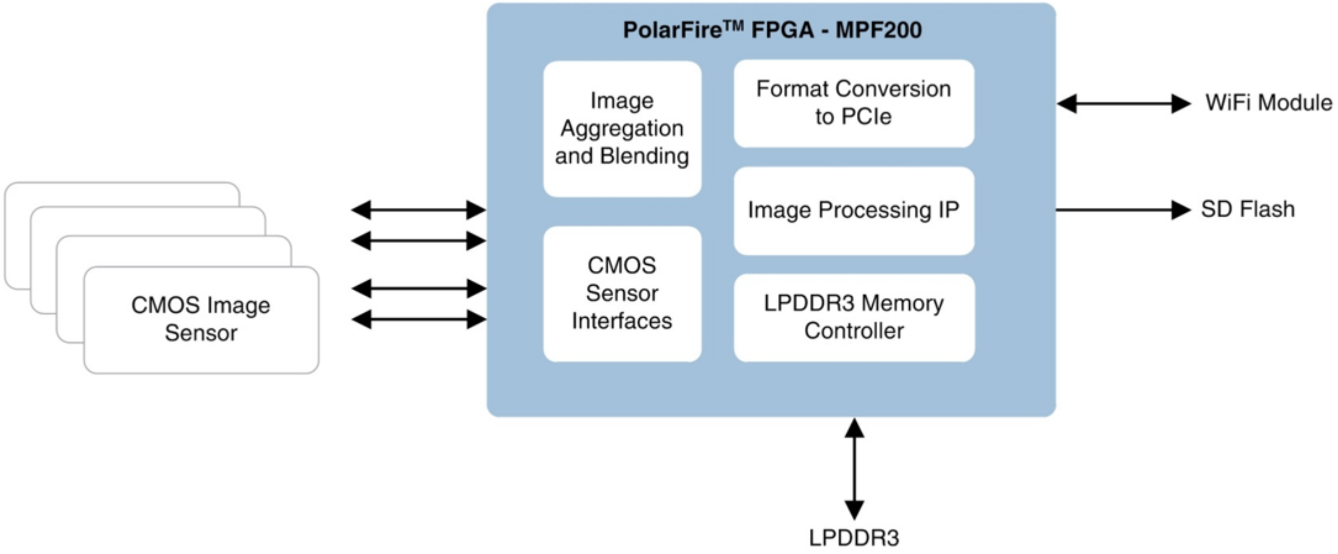
Where FPGAs Fit Within Electronic Systems?

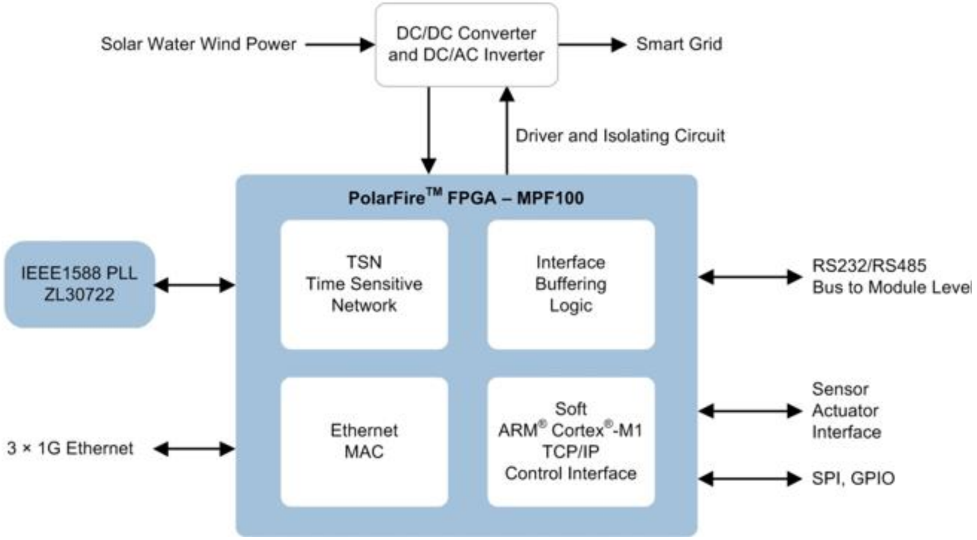
- A common application for an FPGA is as an companion chip to a CPU
  - Functions could range from interface expansion, to DSP offload algorithms, to motor control, to image processing, to NVDIMM controllers
- High-end FPGAs are used in many communications data-path designs
  - Microchip FPGAs fit better in the control plane
- With the advent of embedded processors on FPGAs, a new class of applications is emerging where the entire System is On a Chip (SOC)
  - SmartFusion2 (M3) and Zynq/Cyclone 5 (dual A9 CPUs
- Military and Space applications
  - Mil Avionics, missile guidance, soldier hand held
  - Nasa based space programs and military communications programs
  - Higher volume new space applications – RT plastic



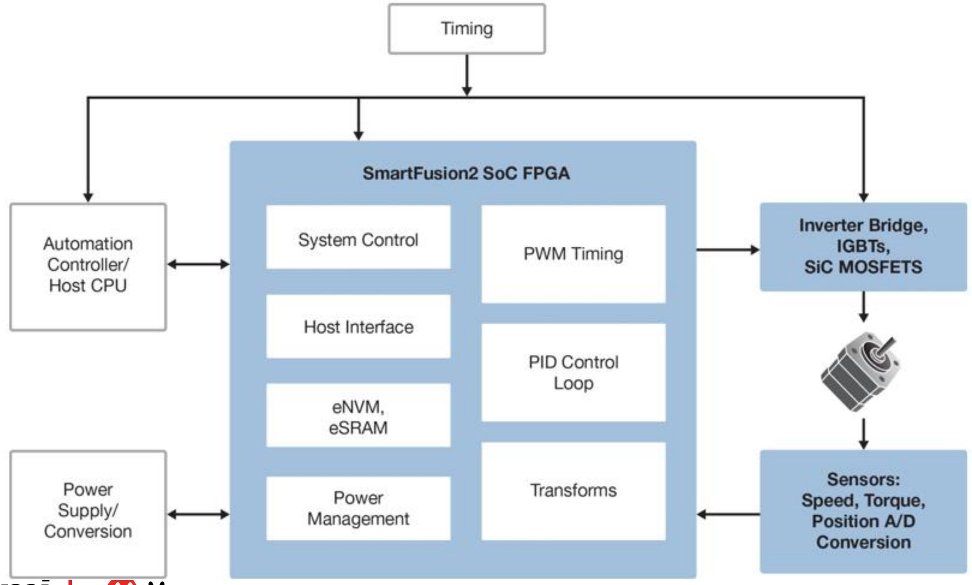


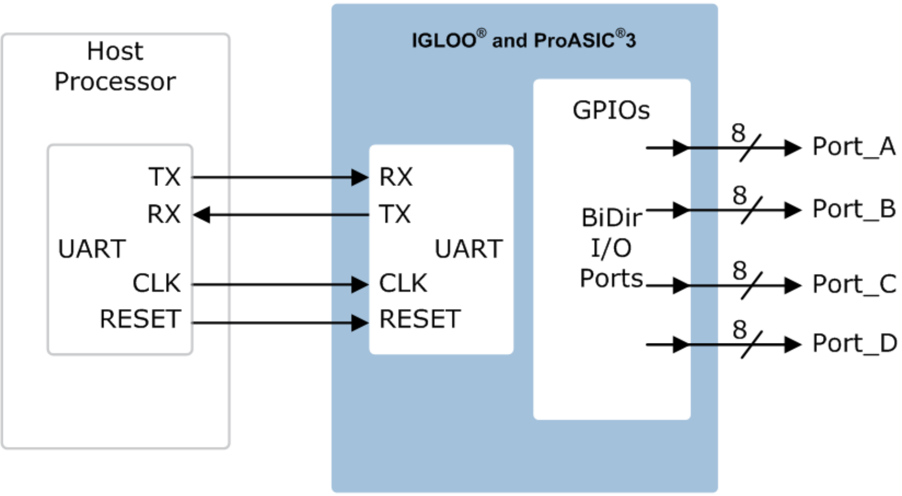
Additional Application Examples





Solar Water Wind Power
DC/DC Converter and DC/AC Inverter
Smart Grid
Driver and Isolating Circuit
PolarFire™ FPGA – MPF100
IEEE1588 PLL ZL30722
TSN Time Sensitive Network
Interface Buffering Logic
RS232/RS485 Bus to Module Level
3 × 1G Ethernet
Ethernet MAC
Soft ARM® Cortex®-M1 TCP/IP Control Interface
Sensor Actuator Interface
SPI, GPIO





SoC FPGA

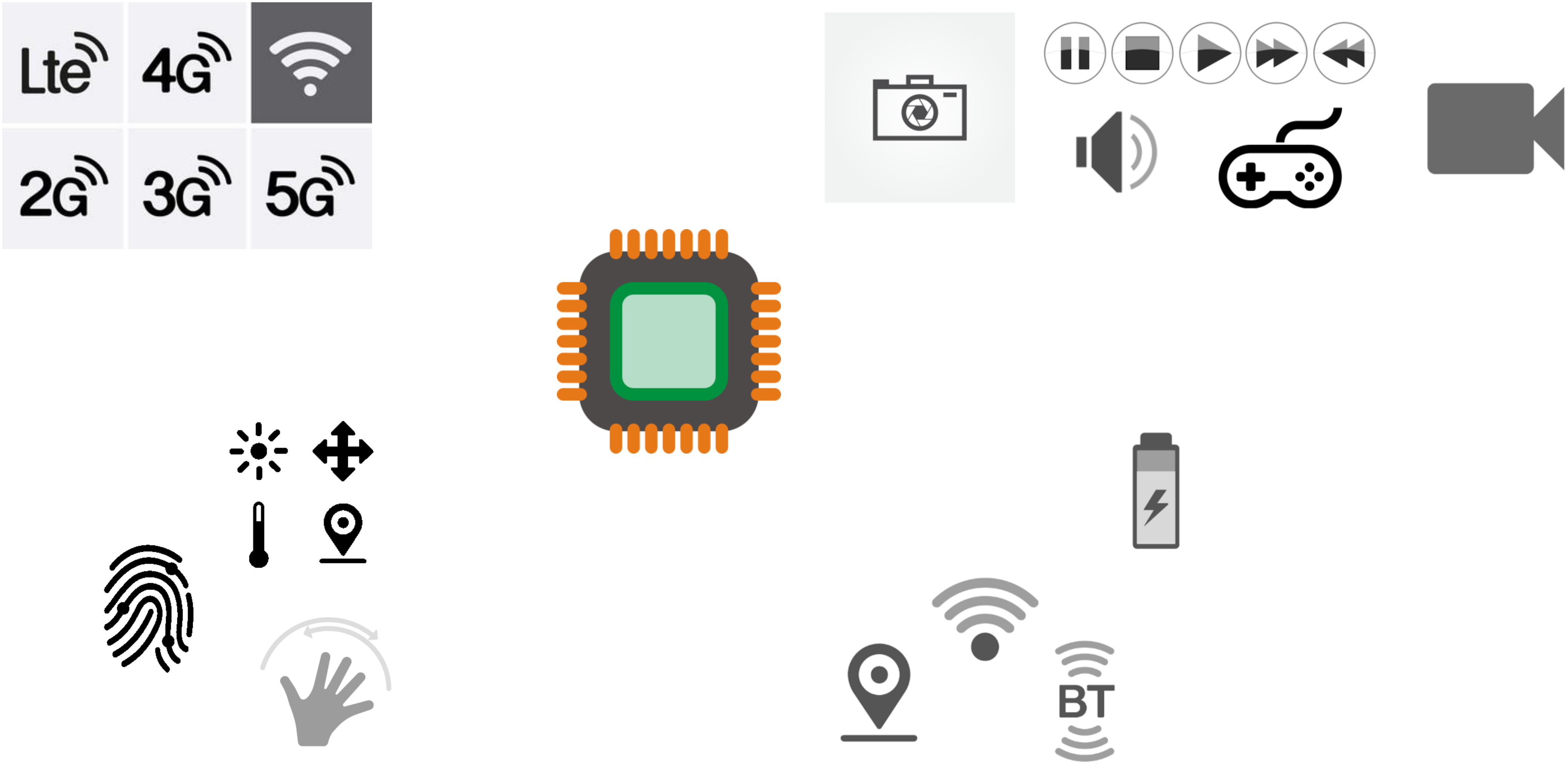
SoC and SoC FPGA

- FPGA
  - Field Programmable Gate Array
  - Plenty of I/O options
  - Extremely parallel architecture
  - Programmable hardware

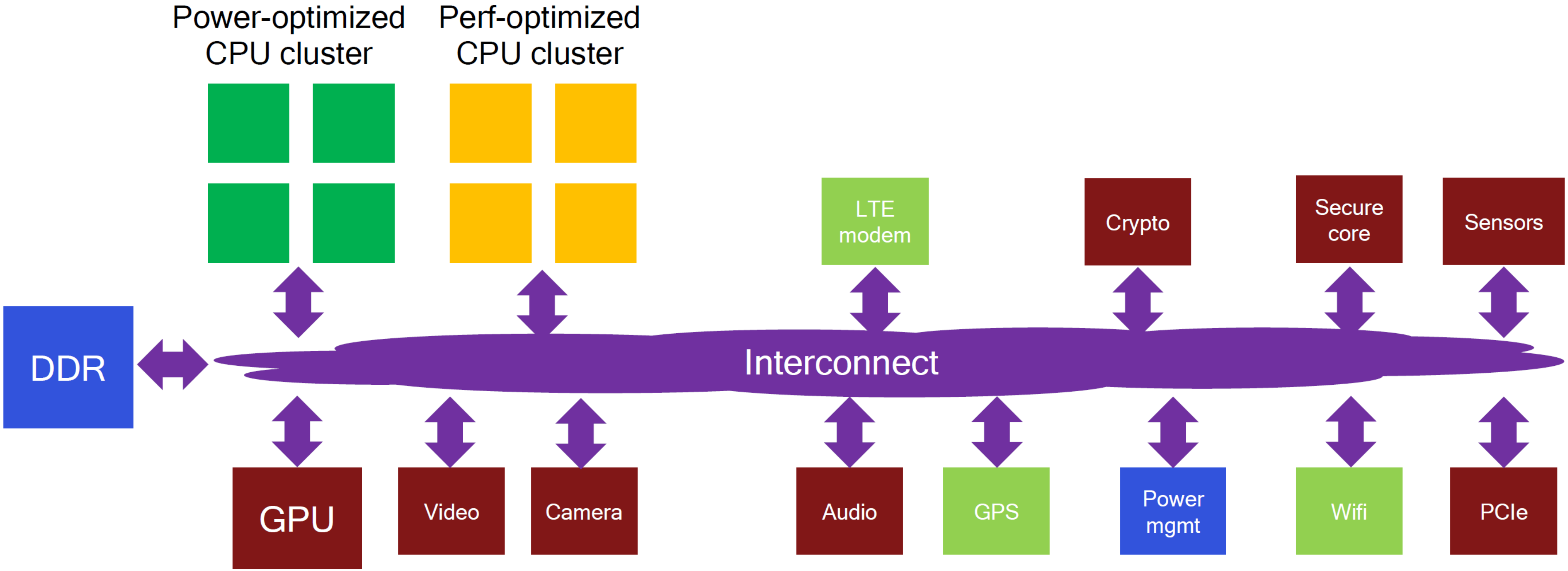
- SoC
  - System on Chip
  - CPU Core + Peripherals
  - Programmable software

- SoC FPGA
  - SoC & FPGA on a single chip
  - Connected through on-chip bus

SoC Overview



SoC Overview



Source: Greg Wright - RISC-V Summit December 2018

Very Complex System

- Many CPUs => Many ISAs + Many SW Stacks
- Multiple clock domains
- Multiple power domains
- Multiple third party IPs
- Different development cycles for each component type
- Multiple skills required

- A validation nightmare

Why SoC FPGA?

- Reduce size => Reduce overall system cost
- Increase performance
- Lower power consumption
- Increase system reliability
- Need for special bus interface for a CPU
- Need for obscure amount of IOs
- Need for extra CPU power for your FPGA
- Need for extra FPGA speedup for your CPU functions

RISC-V: The Opportunity

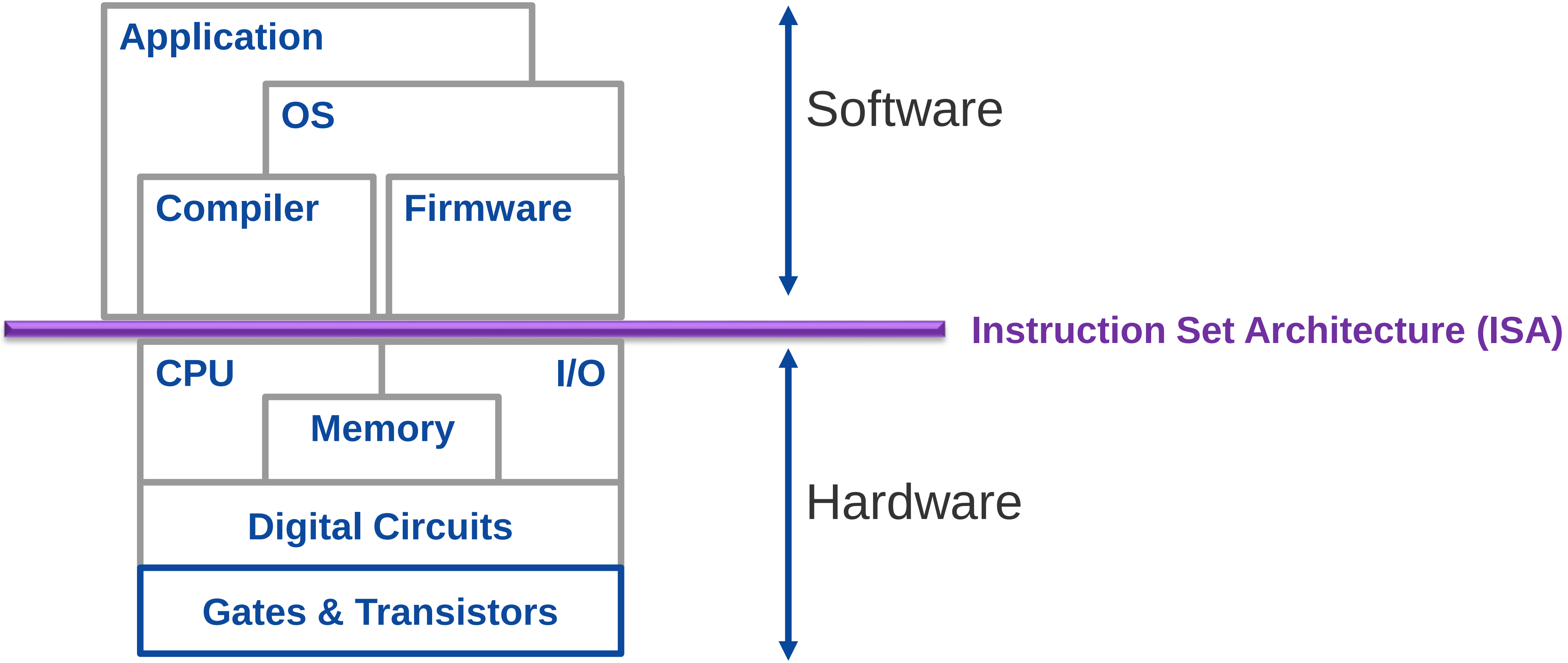
Homogenization

- Common base (and frozen)
- Shared frameworks and libraries
- Rich ecosystem (and growing)

Customization

- Mix-and-match extensions
- Domain specific features
- Proprietary extensions
- Freedom in implementing

Instruction Set Architecture (ISA)



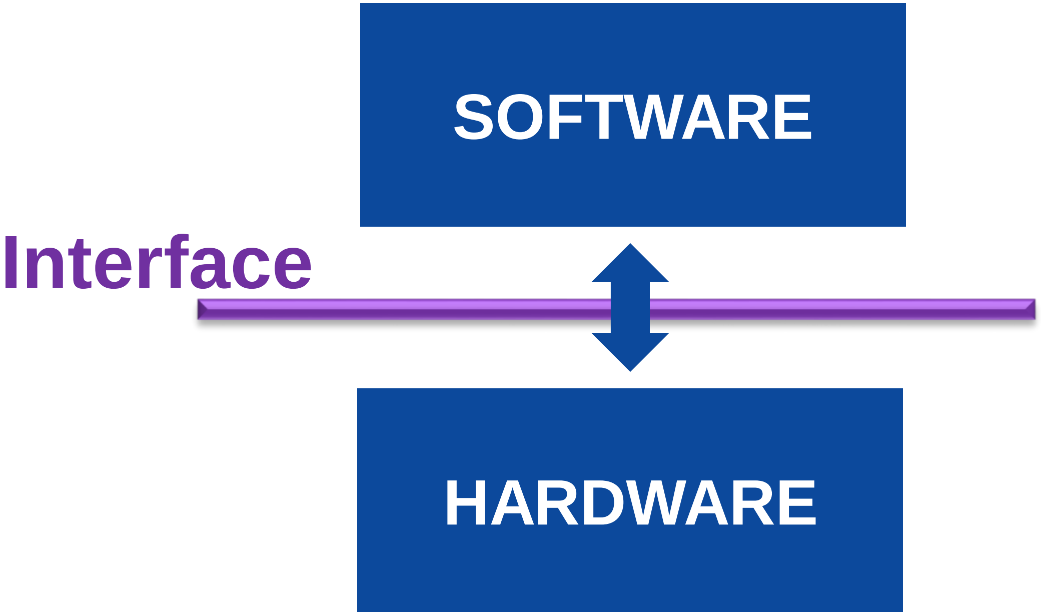
Instruction Set Architecture (ISA)

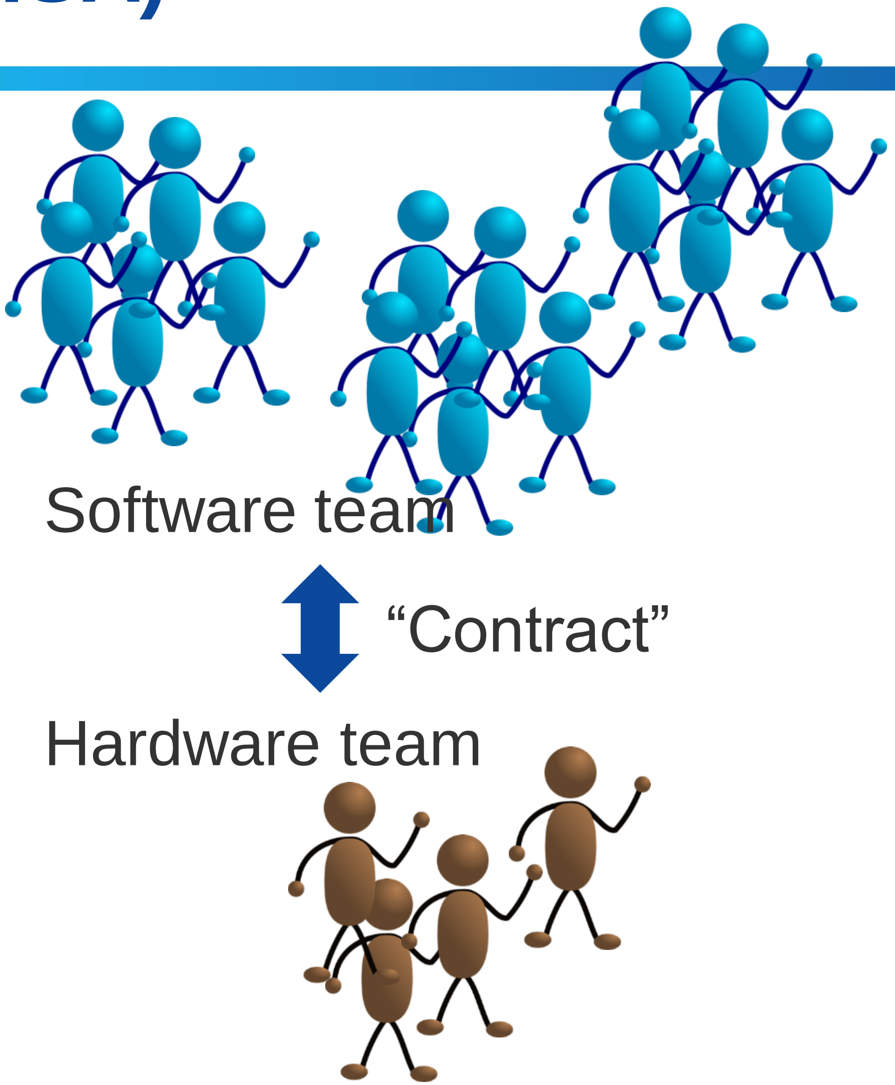
- A well-defined hardware/software interface

- The “contract” between software and hardware
  - Functional definition of operations, modes, and storage locations supported by hardware
  - Precise description of how to invoke, and access them

- Not in the “contract”: non-functional aspects
  - How operations are implemented
  - Which operations are fast and which are slow and when
  - Which operations take more power and which take less

Instruction Set Architecture (ISA)

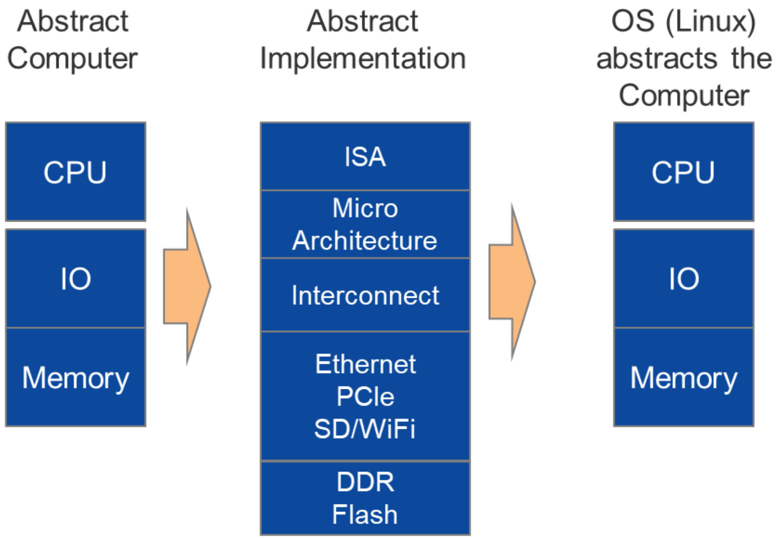




Why RISC-V for a SoC FPGA?

- The RISC-V ISA is Open. Open ISAs allow for …
  - Low cost migration to ASICs, royalty free usage
  - Innovation for custom architectures – free "architectural license" doesn't restrict usage to fixed architectures

- The RISC-V ISA is simple. Simplicity …
  - Allows for a low power implementation
  - Lower cost of ownership - easier to learn, customize and debug
  - Simple architectures are easier to secure against threats



Open, Lowest Power, Cost Optimized, Programmable SoC

When does it make sense?

- Consider the following scenarios:

1. The existing design uses an FPGA and a separate microprocessor?
   - Comparable functionality and performance, but lower board space, power, and cost by as much as 50% less
2. The current generation uses a proprietary ASIC that includes a microprocessor?
   - No expensive mask charges or minimum purchase requirements
   - Faster time to market with lower risk
   - Adapt to changing markets requirements and emerging standards
3. A microprocessor being used today, but would benefit from a peripheral set more tailored to the application?
   - If forced to accept compromises due to the lack of off-the-shelf processor derivatives
   - The design can be differentiated both in hardware and software, making it more difficult for competitors to copy or emulate

In Any Case…

Architecture Matters

Criteria for Choosing an SoC FPGA

- Design considerations & engineering trade-off decisions

- The selection criteria centers on the following areas:
  - Existing ecosystem (legacy IPs, Software…)
  - System performance
  - System reliability
  - System flexibility
  - System cost
  - Power consumption
  - Continuity (product roadmap)
  - Quality of the software solution (development tools)

System Performance

- Industrial Example: Motor Control
  - The processing must be deterministic (i.e. complete within a given window in time, every time)

System Performance

- The processor performance
- The fabric performance
- The interconnect between fabric and processor
- Memory bandwidth

System Performance

The interconnect between fabric and processor

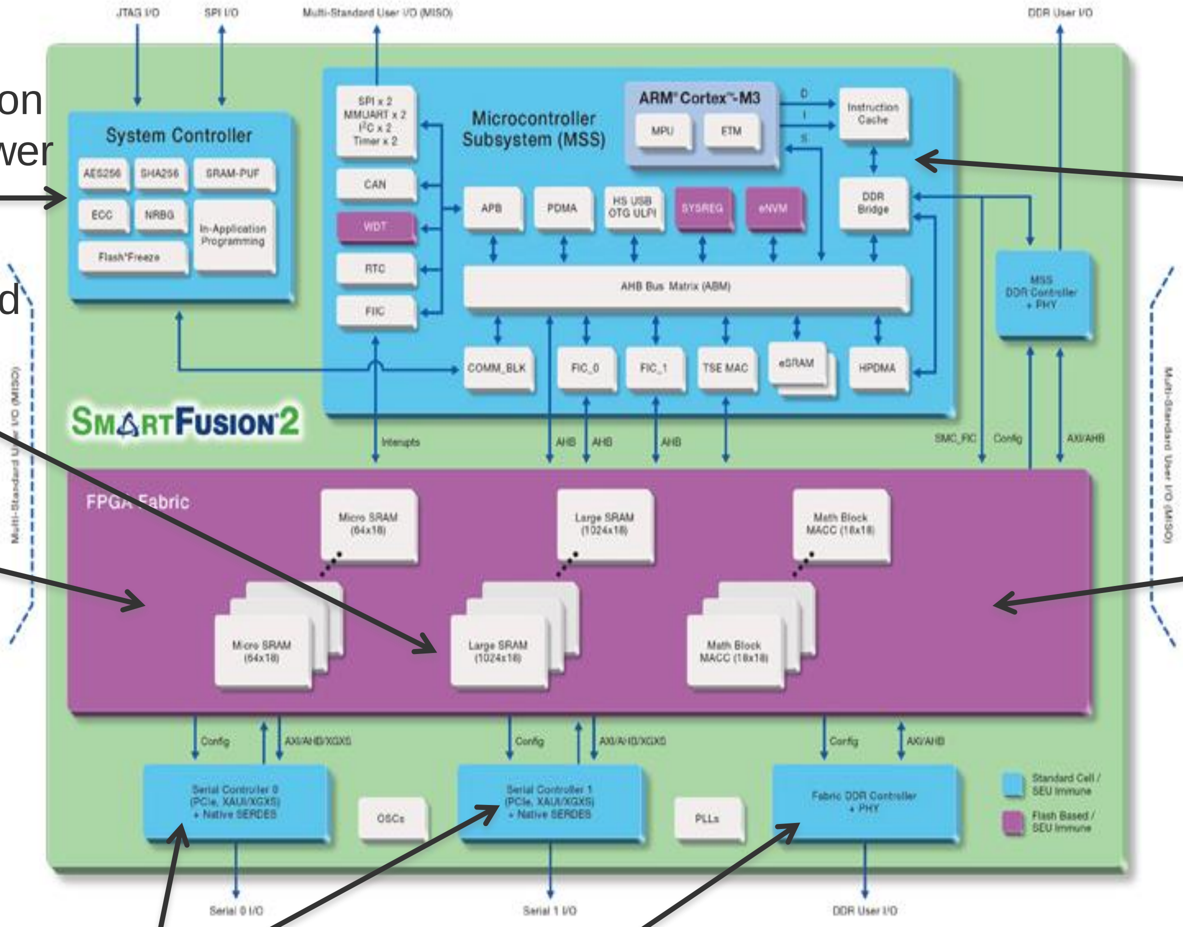
Encryption,
Error Detection
And Low Power
Control

SEU Protected
SRAM Blocks

SEU-Free
Flash FPGA
Configuration
Memory

Hardened
MCU

FPGA Logic



SerDes Channels

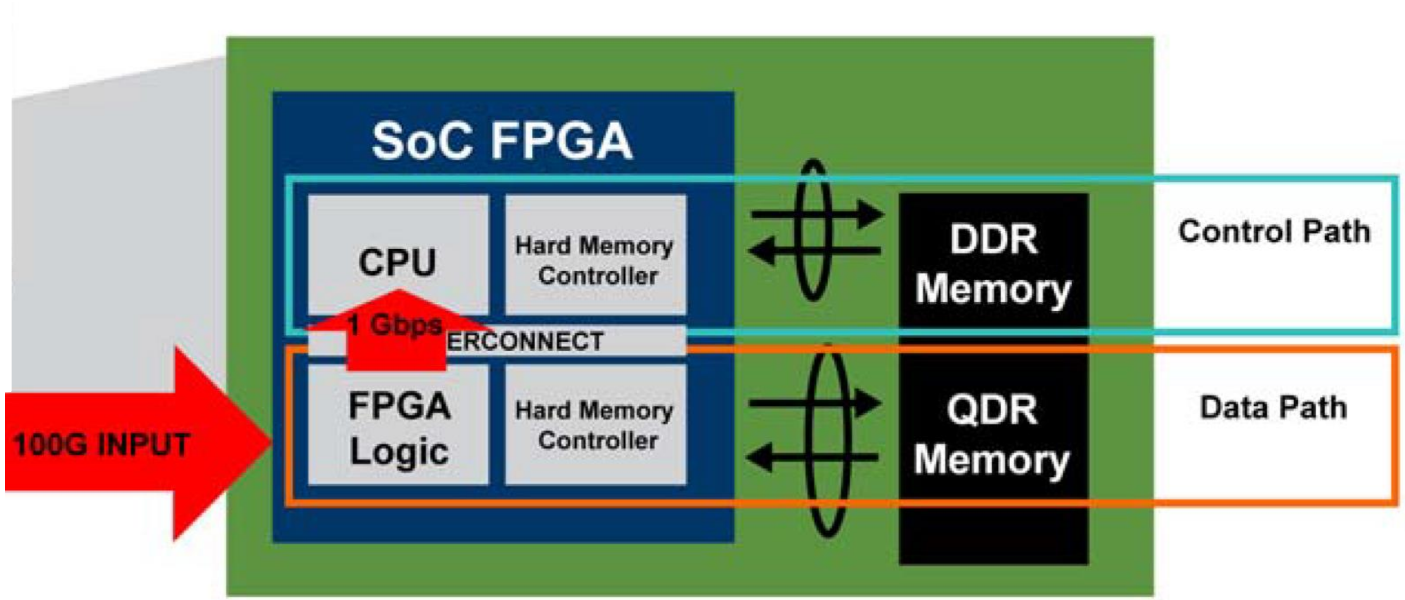
SECDED
Memory Interface

Data transfer between the memory, FPGA fabric, processor, and peripherals

System Performance

The interconnect between fabric and processor

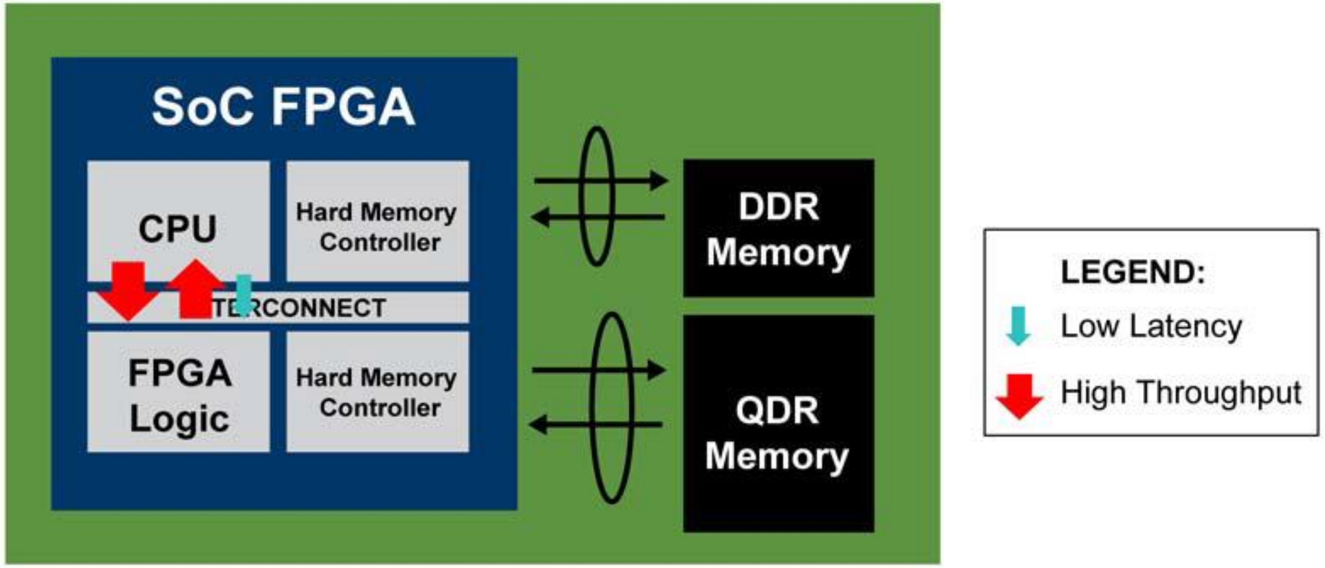
- Question: Does the interconnect support high throughput path?



System Performance

The interconnect between fabric and processor

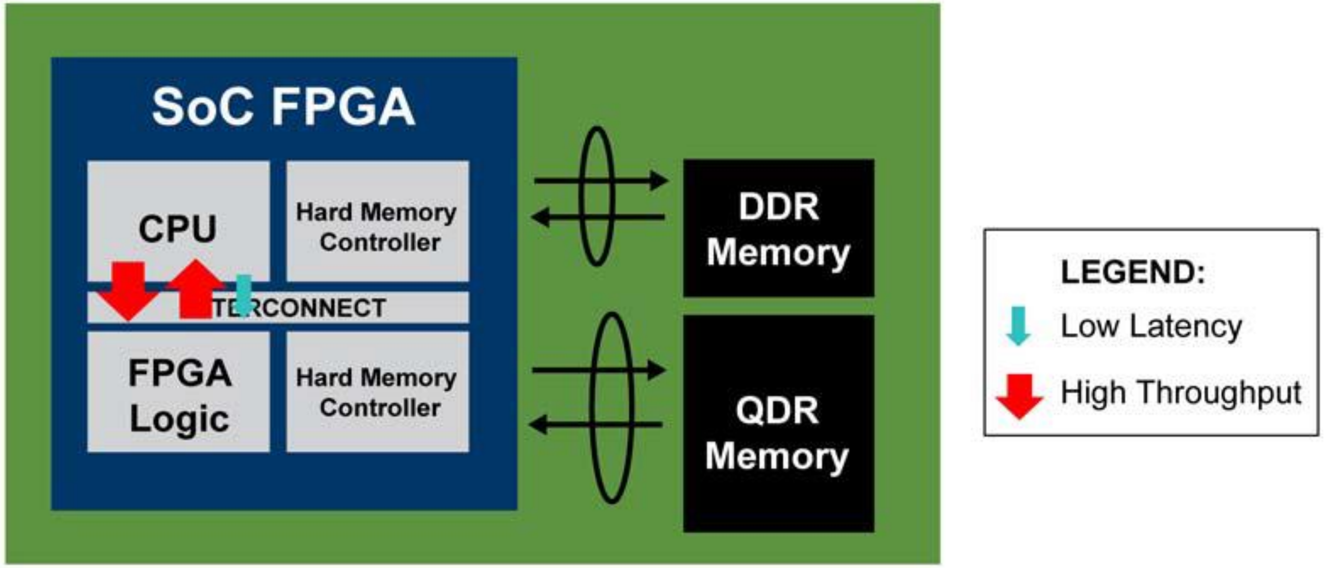
- If needed, a low latency non-blocking bridge for allowing control access to the FPGA



System Performance

The interconnect between fabric and processor

- Hardware acceleration example: When the acceleration results are needed by the processor
- In this case, in the other direction: Does the architecture include an Accelerator Coherency Port (ACP)?

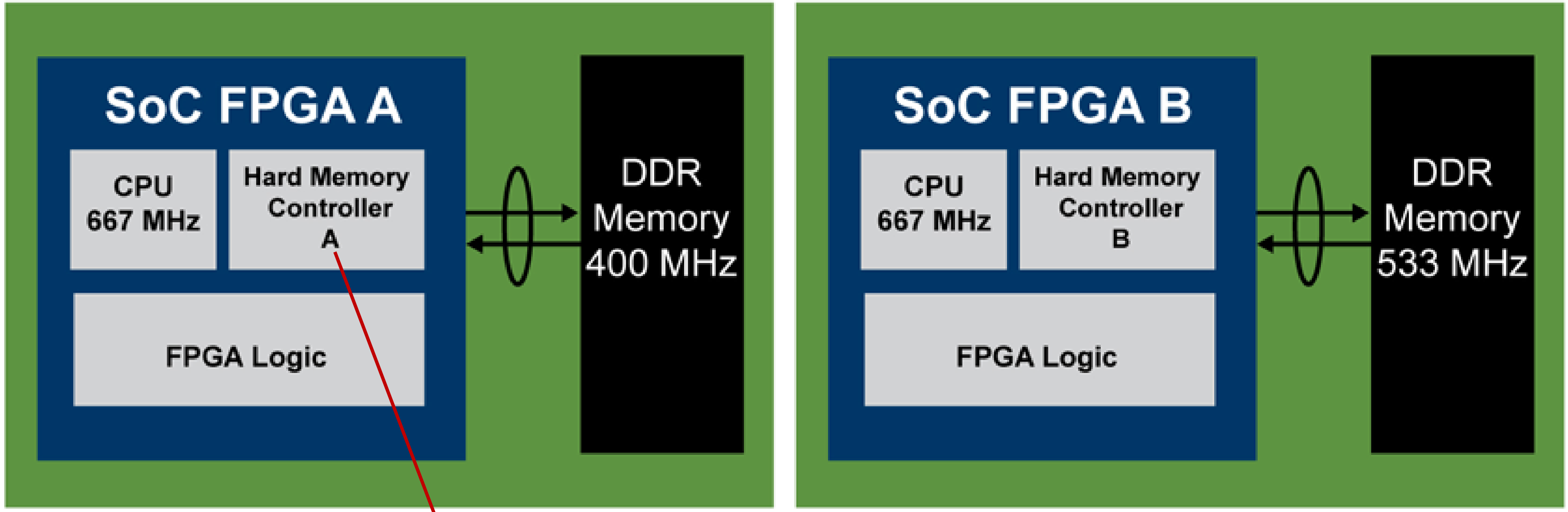


- ARM Cortex-A9-based SoC FPGAs include a feature called an Accelerator Coherency Port (ACP). Through the ACP, new data produced by an FPGA-based hardware accelerator is transferred directly to the processor’s L2 cache via a low-latency direct connection—not just quickly but coherently

System Performance

Memory bandwidth

- Memory controllers are as important as the memory speed
- Question: Do you have separate hard memory controllers?
- Question: How smart is the memory controller?

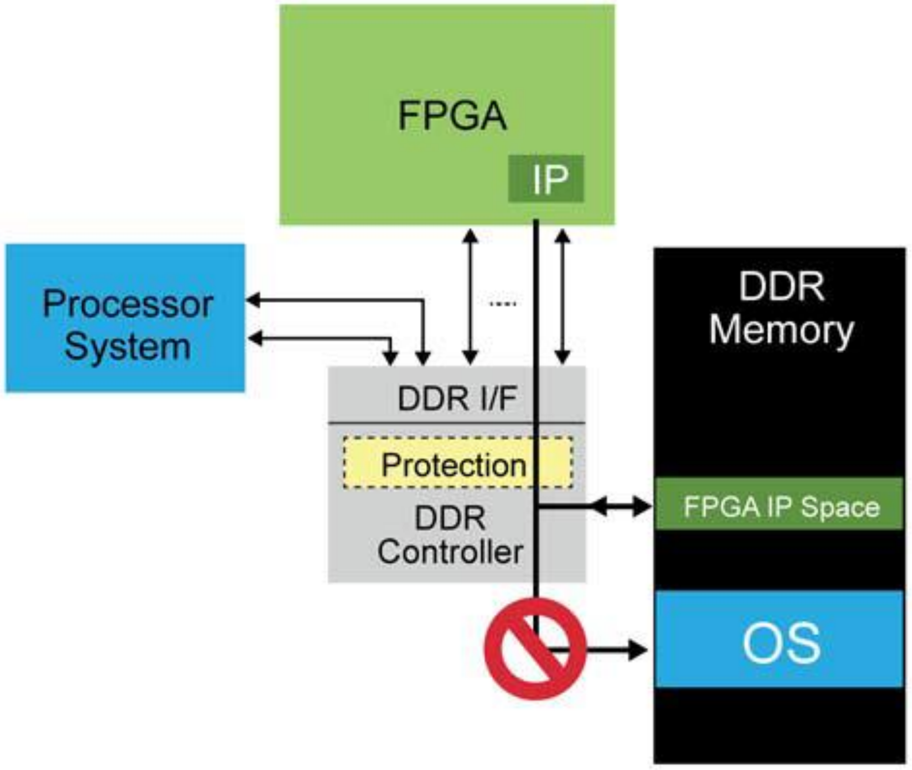


17% Faster using a smarter scheduling algorithm (e.g. deficit weight round robin)

System Reliability

- Supporting ECC Memory for content protection
  - On-Chip RAM
  - External DDR Memory Controller
  - L1 Cache & L2 Cache
  - SPI Controller
  - DMA Controller
  - 10/100/1G Ethernet Controller
  - USB 2.0 OTG Controller
  - …

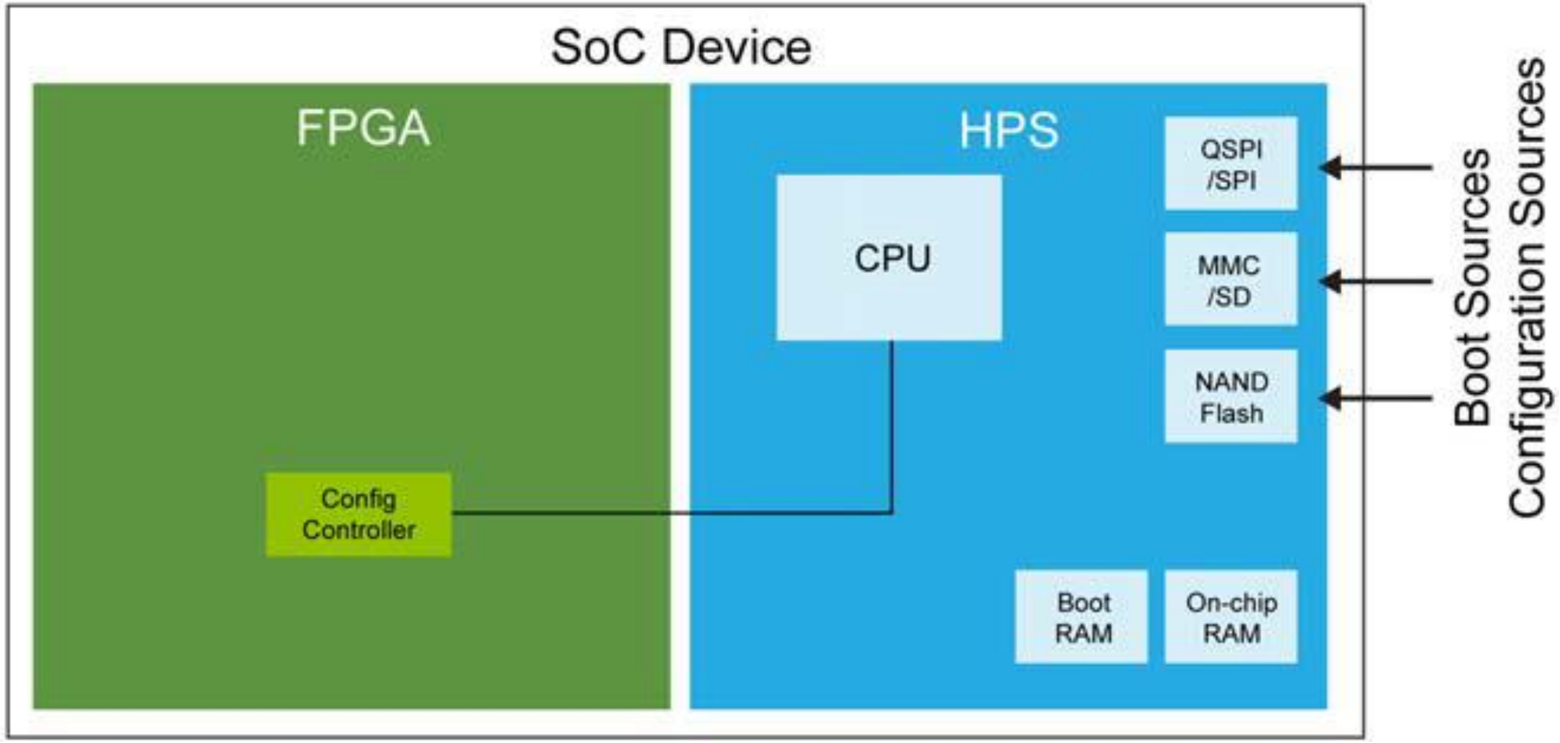
- Protection for shared memory
  - Arm has the concept of “trust zone”



System Flexibility

- Extending the flexibility to the system level

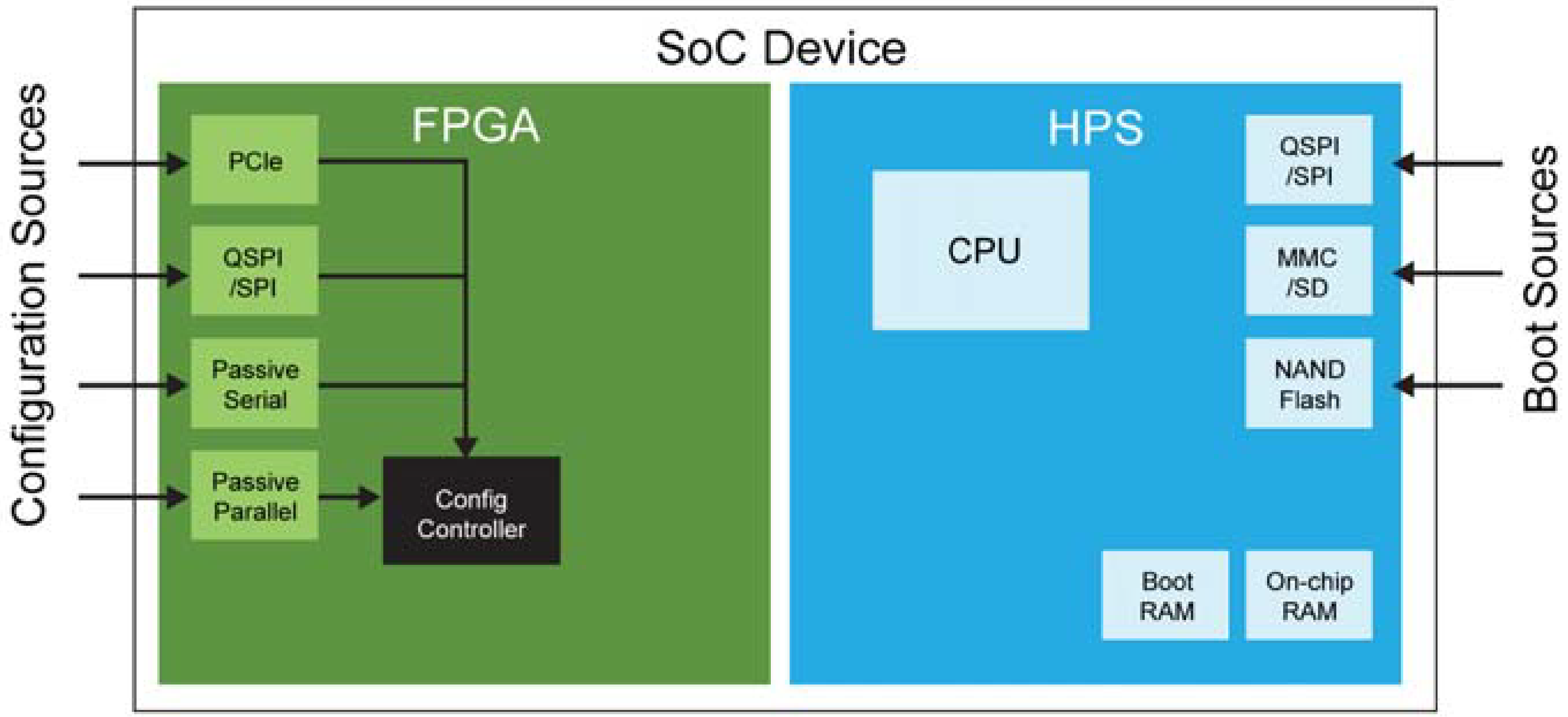
Processor boots first, then configures the FPGA



System Flexibility

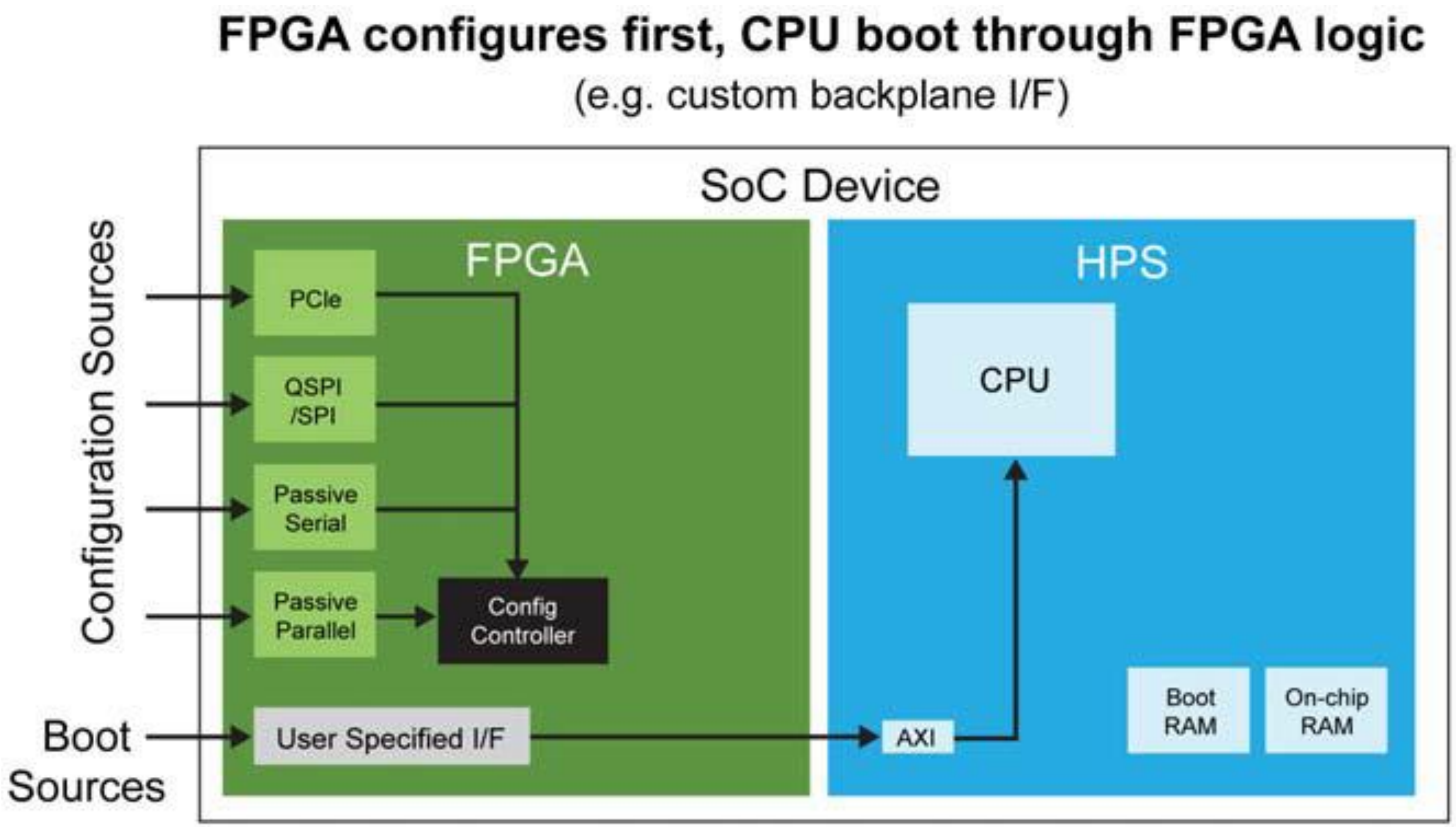
- Extending the flexibility to the system level

Independent FPGA configuration and processor boot



System Flexibility

- Extending the flexibility to the system level



Criteria for Choosing an SoC FPGA

- Design considerations & engineering trade-off decisions

- The selection criteria centers on the following areas:
  - Existing ecosystem (legacy IPs, Software…)
  - System performance
  - System reliability
  - System flexibility
  - System cost
  - Power consumption
  - Continuity (product roadmap)
  - Quality of the software solution (development tools)

PolarFire SoC FPGA Architecture

Mi-V: Microsemi’s RISC-V Processor

PolarFire SoC FPGA

Industry’s First RISC-V SoC FPGA Architecture Brings Real-Time to Linux, Giving Developers the Freedom to Innovate in Low-Power, Secure and Reliable Designs

- First SoC FPGA with deterministic, coherent CPU cluster and a deterministic L2 memory subsystem enabling Linux + real-time applications
- First SoC FPGA architecture integrating a RISC-V processor subsystem and low-power FPGA technology
- PolarFire SoC FPGA with a mid-range low-power SoC FPGA and high levels of security and reliability

Freedom to implement RISC-V in hardware

SiFive

SoftConsole

RISC-V

0x5
HEX–Five

Zephyr

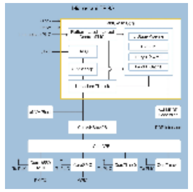
antmicro

Microsemi

Mi-V™

DIGITAL CORE
TECHNOLOGIES
QUALITY BY DESIGN

Libero
System-on-Chip



EmCraft
systems

aries
embedded
Intelligence for Industry

APACHE
mynewt

expresslogic

free RTOS

Micrium®
Embedded Software

PHYTEC

ultrasoc

µC/OS™
The Real-Time Operating System

imperas

DornerWorks

Award Winning PolarFire FPGA as an SoC platform

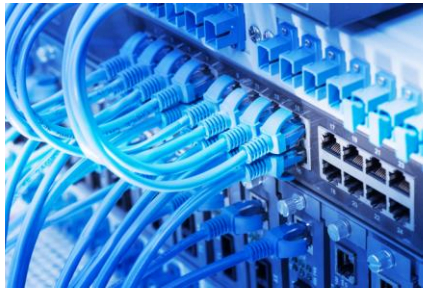
- Lowest Power
  - Low static power technology
  - Power optimized transceivers
  - Up to 50% lower than SRAM FPGAs

- Proven Security
  - Defense-grade security
  - DPA safe Crypto coprocessor
  - Built-in anti-tamper

- Exceptional Reliability
  - SEU immune configuration
  - Block RAM with ECC
  - Extended temperatures

10G Bridging & Aggregation

Winner
Electronic Products 2017 Product of the Year
PolarFire FPGA
Winner
21ic Electronic Products China 2017 Product of the Year
PolarFire FPGA
Winner
China Electronics Market 2017 Editor's Choice Award
PolarFire FPGA
Finalist
Elektra 2017 Digital Semiconductor Product of the Year
PolarFire FPGA



Control Plane

Video & Image Processing

Microsemi
Power Matters.™

POLARFIRE™
FPGA

COST-OPTIMIZED
LOWEST POWER
MID-RANGE DENSITIES

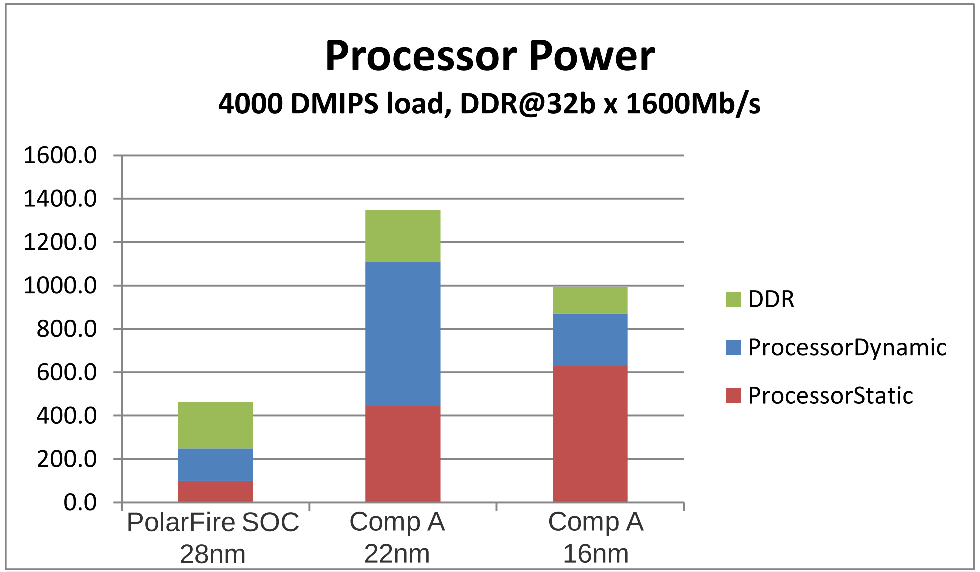
Hardware Acceleration

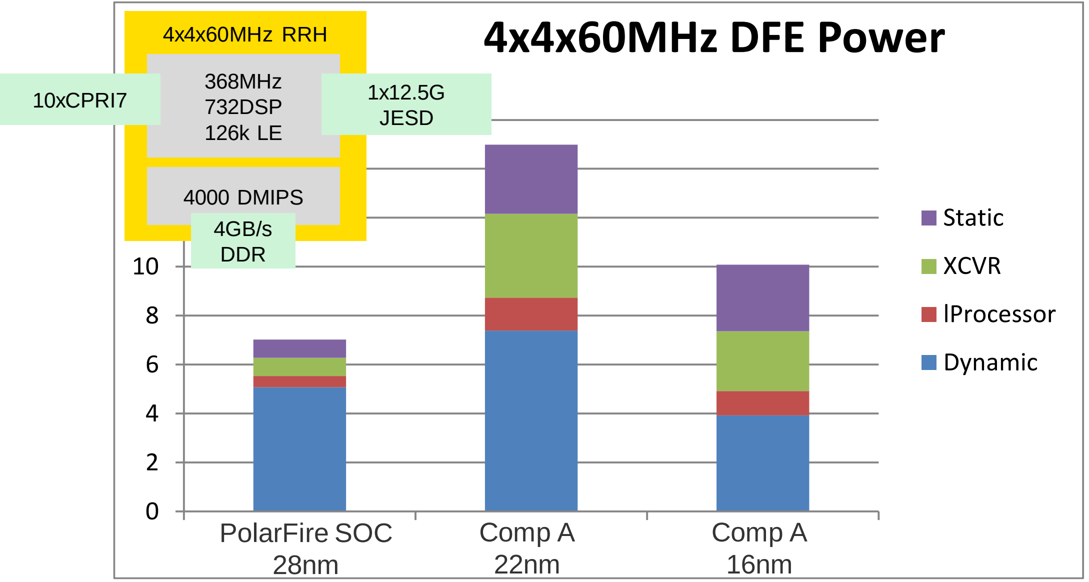
Portable Equipment

Signal Processing

Low Power Optics

Low Power: RISC-V MSS vs. Alternatives





<u>Low Power MSS</u>

Low power RISC-V micro-architecture

2MB L2 cache: increases cache hit/miss ratio

0.5-0.9W lower power

Power-optimized transceivers

<u>LTE Digital Front End Application</u>

3-7W lower power for 60MHz 4x4 MIMO implementation

Security Ready for IoT

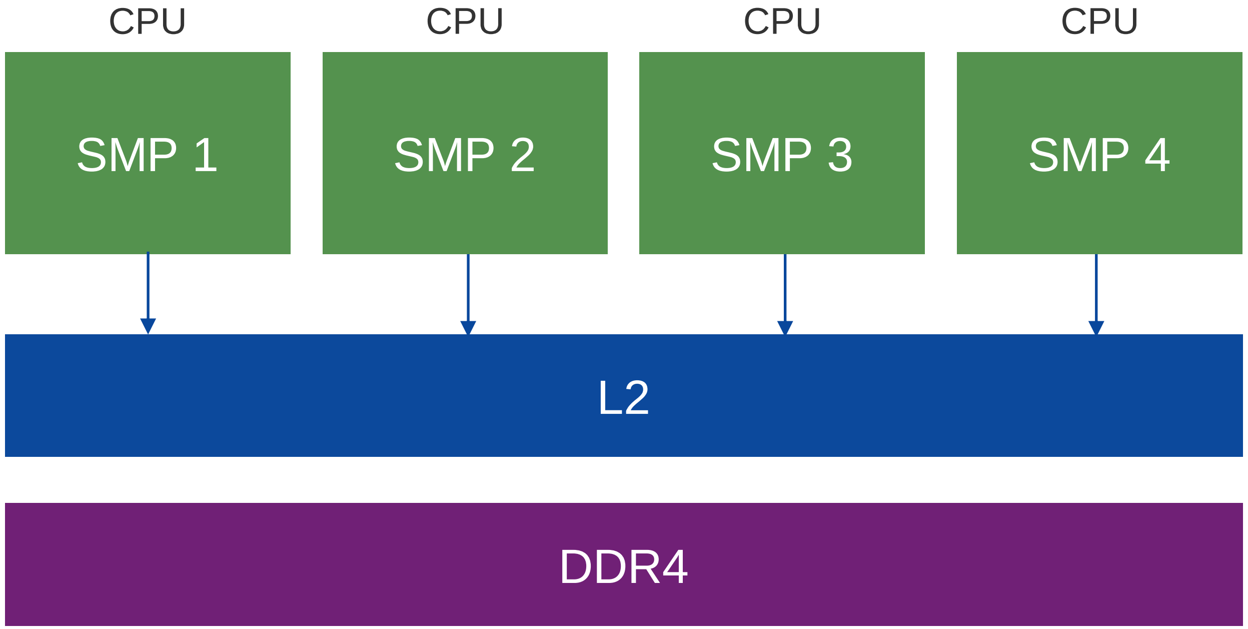
- PolarFire SoC inherits best in class PolarFire FPGA Security
  - DPA resistant bitstream programming
  - Anti-tamper
  - Cryptographical bound supply chain assurance
  - Physically unclonable function
  - True random number generator
  - Side channel resistant crypto coprocessor

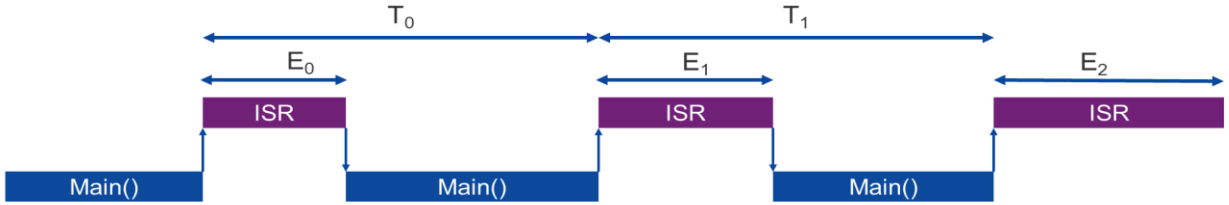
- **PolarFire SoC has:**
  - + **Secure Boot**
  - + **Spectre and Meltdown immunity**
  - + **Physical memory protection**
  - + **SECDED on all memories**
    (single-error correction and double-error detection)

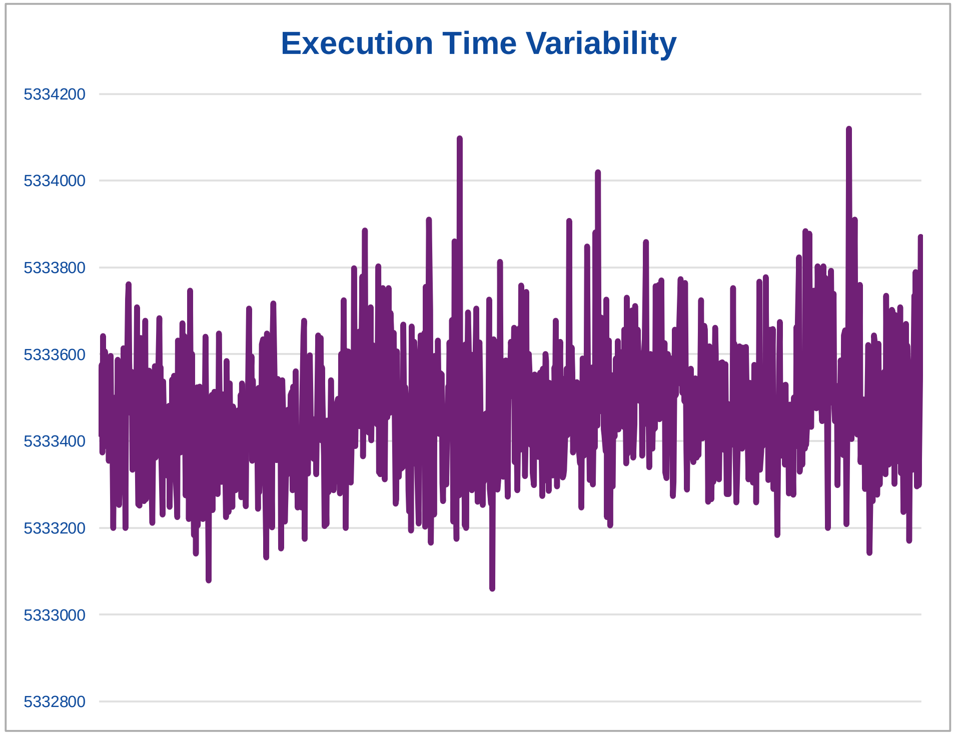
Real-time Linux?

- Wide spread Linux adoption
  - Rich OS with thousands of applications to choose from
- Requirements still exist for real-time while running Linux
  - Safety critical
    - The ability to deterministically monitor the execution environment.
  - Real-time system control
    - Completing tasks deterministically, on time every time.
  - Securing the IoT
    - Execute a trusted execution environment deterministically for consistent results.
- Working with SiFive
  - We have been able to architect a complex SoC FPGA that provides
    - Determinism and a rich OS within the same multi-core CPU cluster

Variable Execution Time in Typical Application Processors







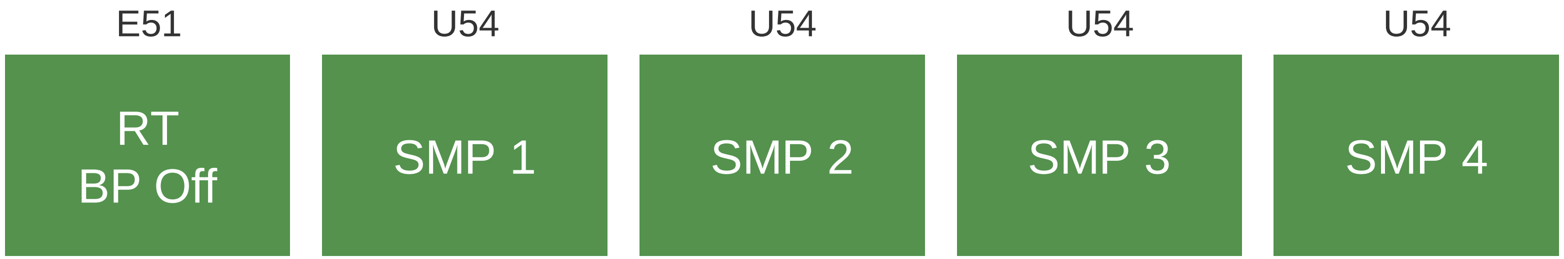
- Periodic Interrupts
  - $T_0 = T_1$
- Inconsistent Execution Times
  - $E_0 != E_1 != E_2$

Branch predictors, cache misses and lack of coherency ***affect determinism negatively***

Microsemi | a Microchip company

Solution

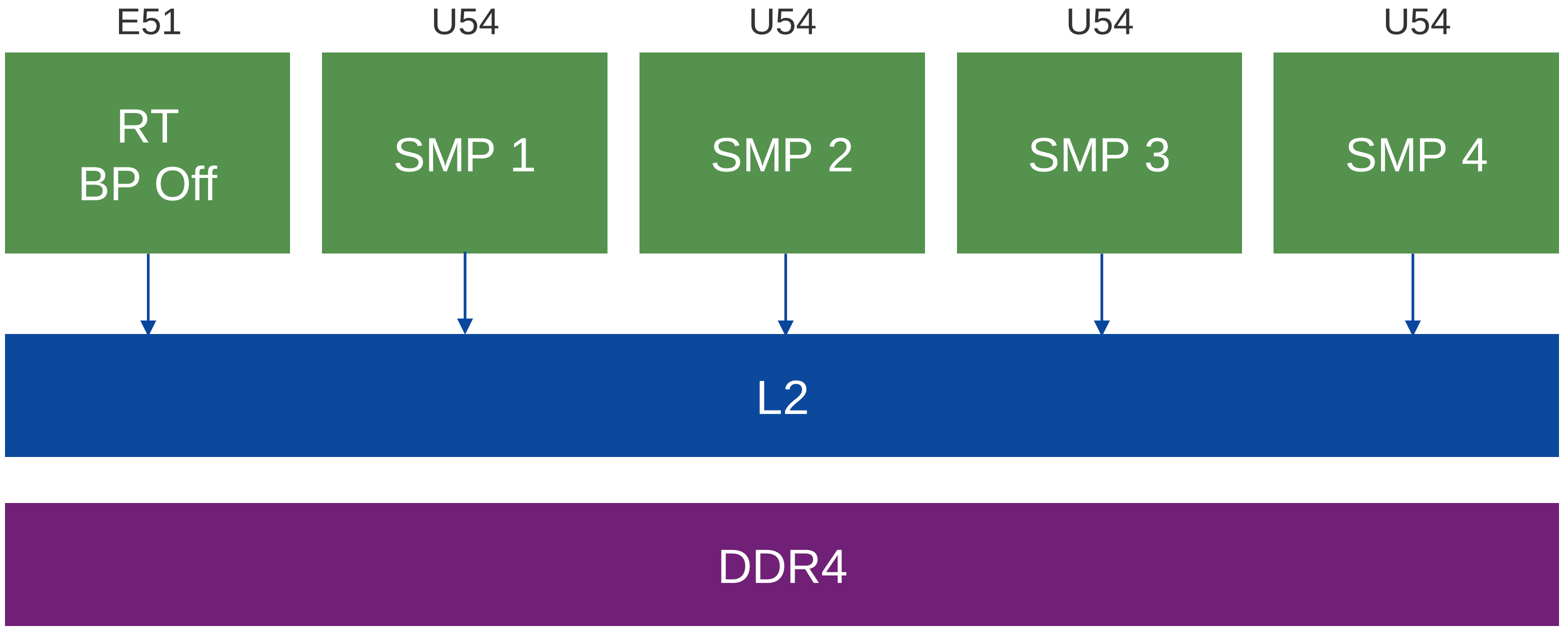
- Turn off the CPU branch predictors



E51	U54	U54	U54	U54
RT BP Off	SMP 1	SMP 2	SMP 3	SMP 4

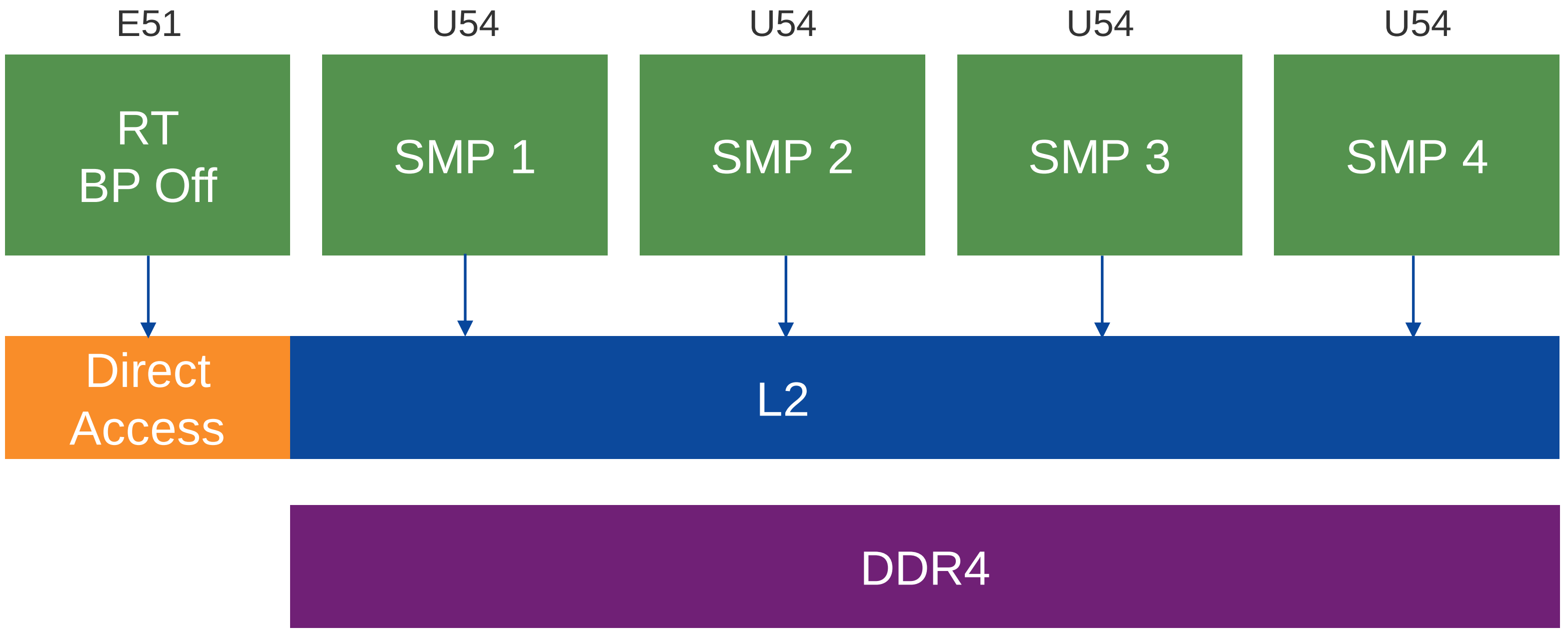
Solution

- Turn off the CPU branch predictors
- Make sure all cores coherent to the memory subsystem



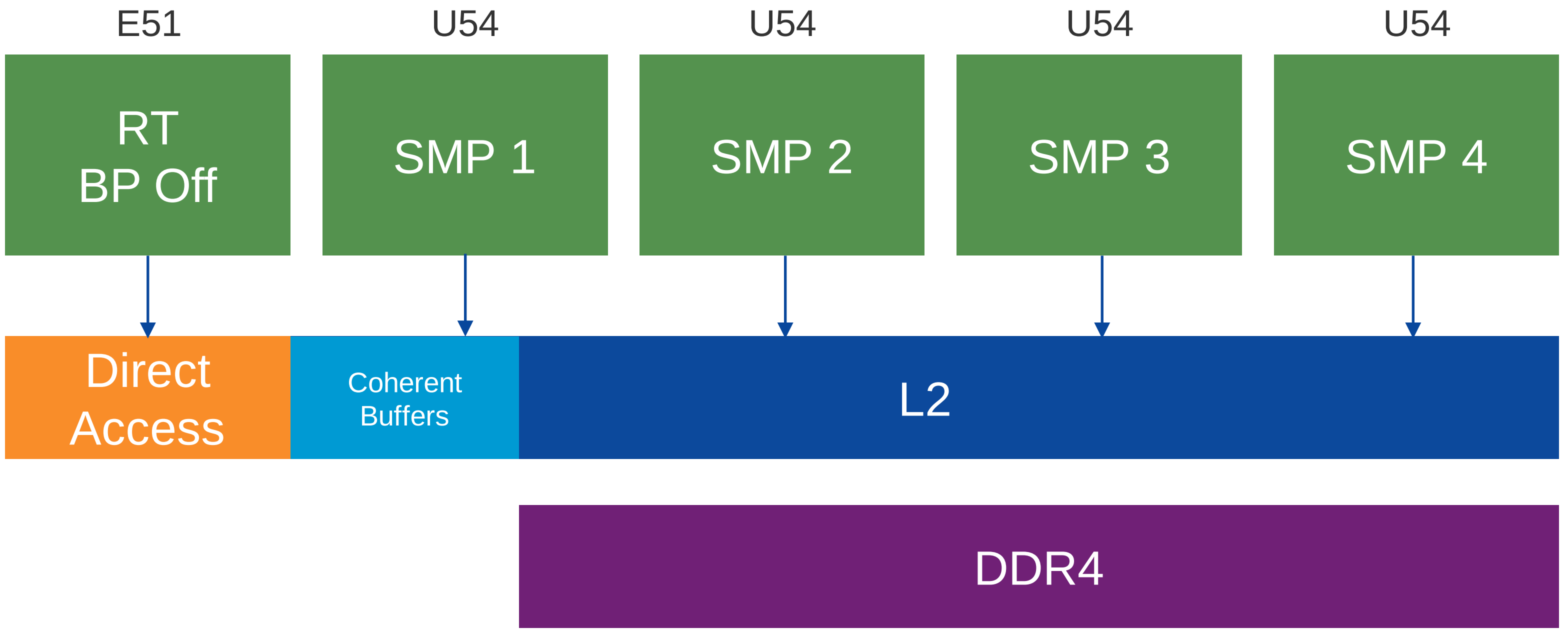
Solution

- Turn off the CPU branch predictors
- Make sure all cores coherent to the memory subsystem
- Make the memory system deterministic



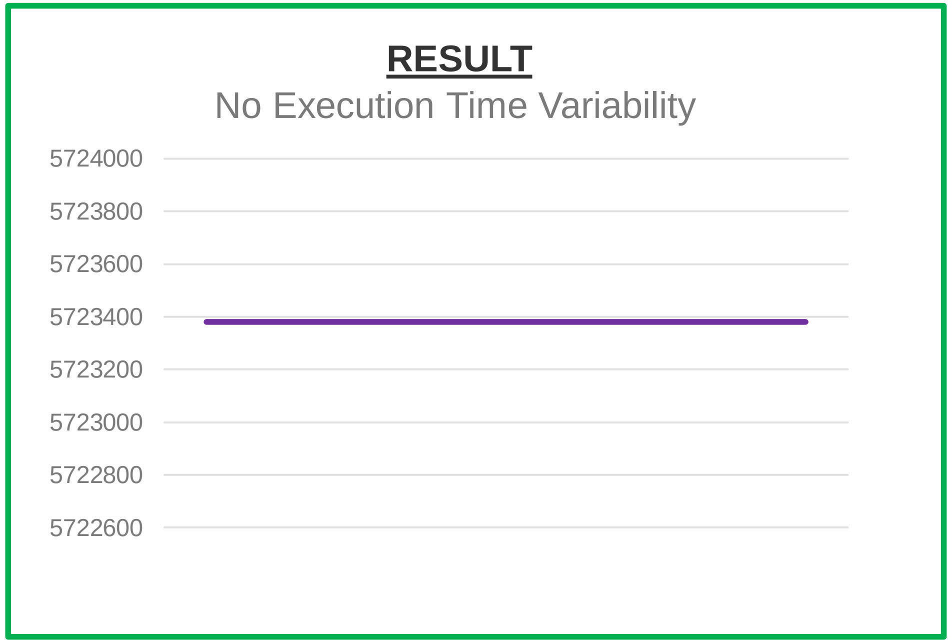
Solution

- Turn off the CPU branch predictors
- Make sure all cores coherent to the memory subsystem
- Make the memory system deterministic
- Share coherent memory for message passing



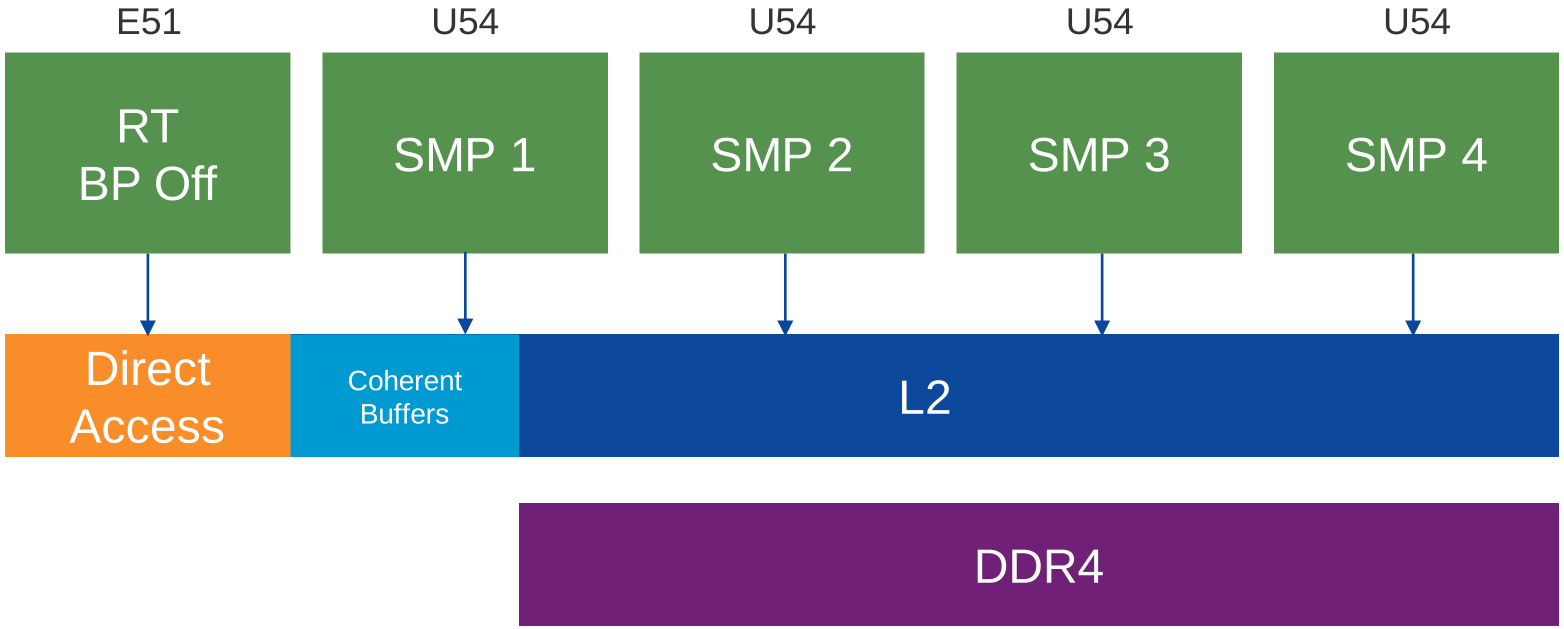
Solution

- Turn off the CPU branch predictors
- Make sure all cores coherent to the memory subsystem
- Make the memory system deterministic
- Share coherent memory for message passing
- Linux and real time in a coherent memory subsystem



RESULT

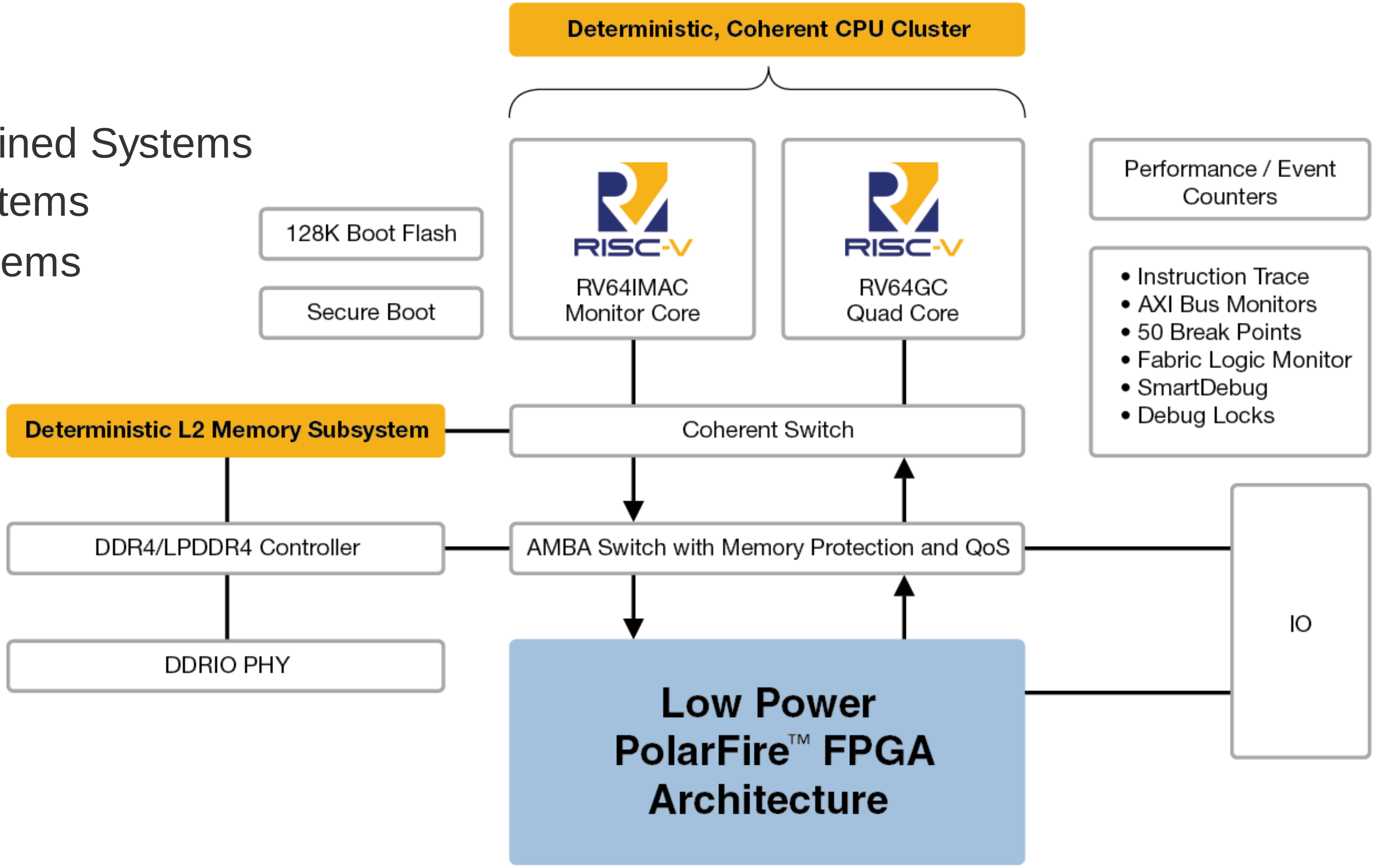
No Execution Time Variability



PolarFire SoC - RISC-V enabled innovation platform

Freedom to Innovate in

- Linux and Real-Time
- Thermal and Power Constrained Systems
- Securely Connected IoT systems
- High-Rel Safety Critical Systems

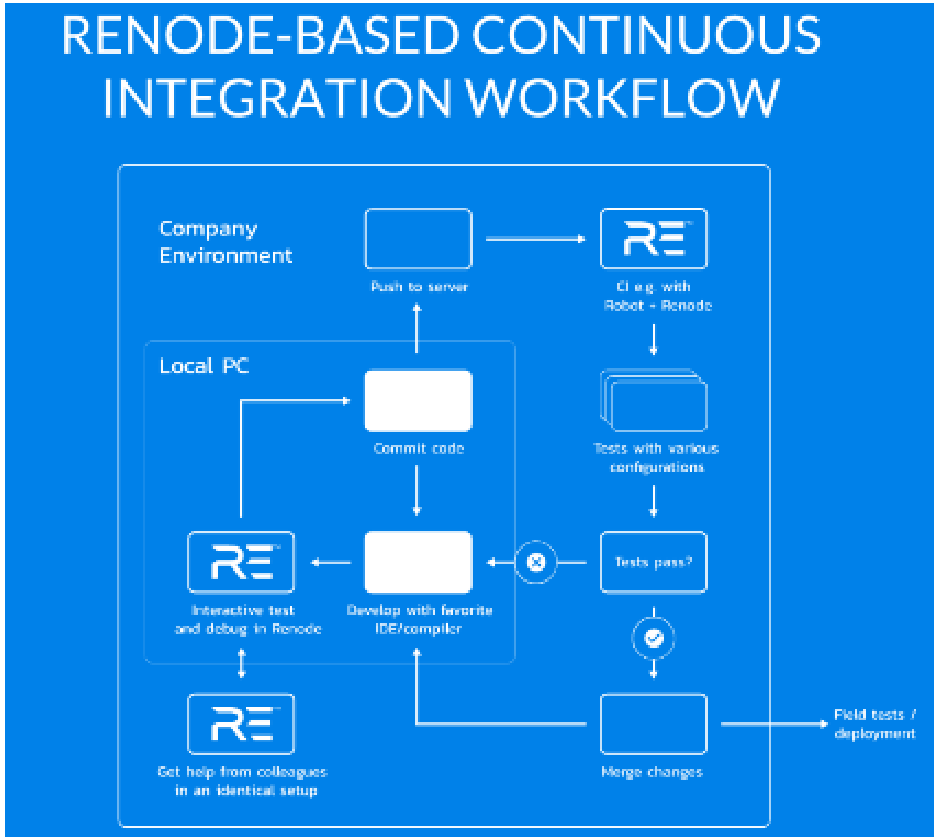


PolarFire SoC Summary

- PolarFire FPGA Award Winning Features
  - 30-50% Lower power
  - Defense grade security
  - Exceptional reliability
  - Smallest, lowest power, secure form factors – 11x11, 16x16, 19x19

- PolarFire Microprocessor Subsystem
  - Linux and real time in a deterministic, coherent CPU cluster
  - 30-50% Lower power
  - Defense grade secure boot
  - Spectre/Meltdown immune
  - SECDED on all memories

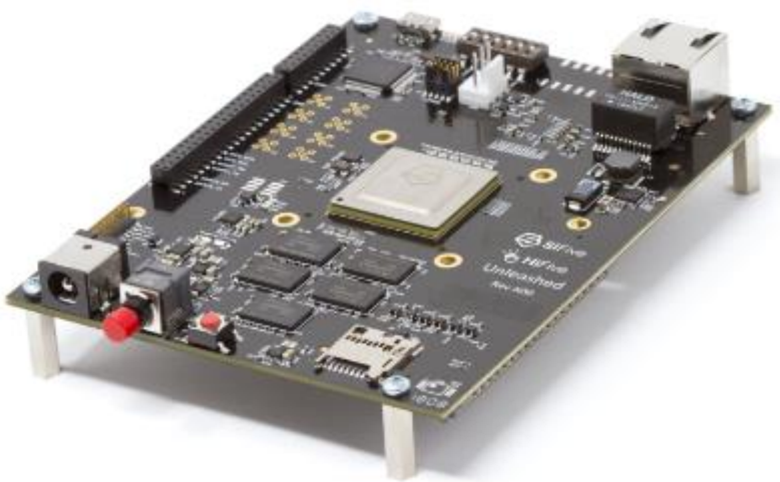
Freedom to start software development


RENODE-BASED CONTINUOUS INTEGRATION WORKFLOW
Company Environment
Push to server
CI e.g. with Robot + Renode
Local PC
Commit code
Tests with various configurations
Interactive test and debug in Renode
Develop with favorite IDE/compiler
Tests pass?
Field tests / deployment
Get help from colleagues in an identical setup
Merge changes

- Free Rapid Software Development and Debug Capabilities Without Hardware
- Complete PolarFire SoC Processor Subsystem Model
- Available now!

Freedom to begin hardware development

PolarFire SoC Embedded Experts Development Kit



[HiFive Unleashed Expansion Board](#)

[HiFive Unleashed Development Board](#)

Freedom to engage with the Mi-V Ecosystem

New Mi-V Embedded Experts Network

antmicro

aries embedded
Intelligence for Industry

DIGITAL CORE TECHNOLOGIES
QUALITY BY DESIGN

DORNERWORKS

EmCraft systems

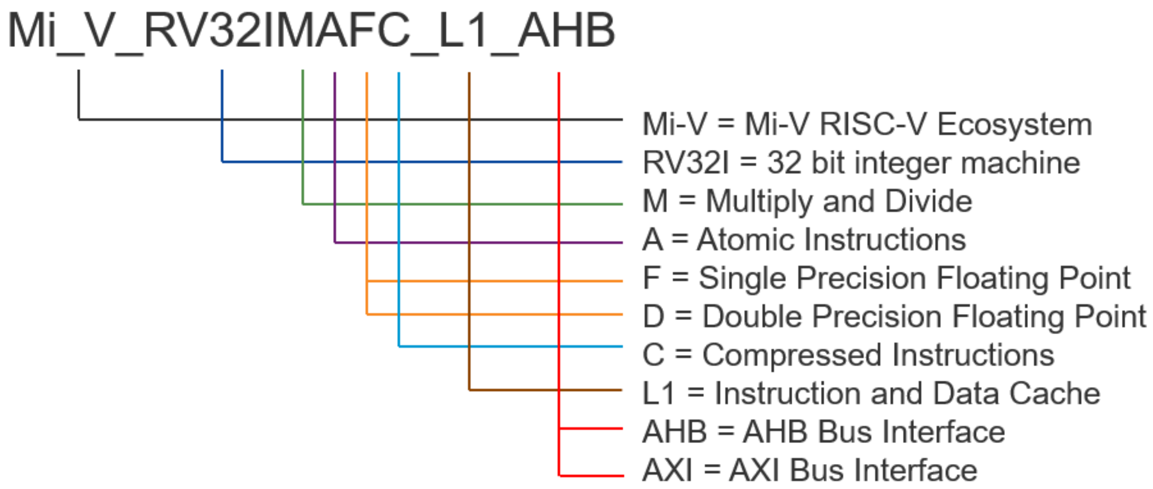
EMDALO TECHNOLOGIES
Embedded Intelligent Systems

PHYTEC

RISC-V CPUs

RISC-V Soft CPU	LE'S	CoreMark Score	Cache Size	Mul/Div	Floating Point	Availability
CORE_RISCV_AXI4*	10K	2.01	8K I and D	Yes	N/A	Now
Mi_V_RV32IMAF_L1_AHB*	26K	2.01	8K I and D	Yes	Single Precision	Now
Mi_V_RV32IMA_L1_AHB*	10K	2.01	8K I and D	Yes	N/A	Now
Mi_V_RV32IMA_L1_AXI*	10K	2.01	8K I and D	Yes	N/A	Now

*Click the RISC-V Soft CPU to download the Handbook



Mi_V_RV32IMAFC_L1_AHB

Mi-V = Mi-V RISC-V Ecosystem
RV32I = 32 bit integer machine
M = Multiply and Divide
A = Atomic Instructions
F = Single Precision Floating Point
D = Double Precision Floating Point
C = Compressed Instructions
L1 = Instruction and Data Cache
AHB = AHB Bus Interface
AXI = AXI Bus Interface

Summary

- ***PolarFire SoC*** gives designers the freedom to create innovative low power systems by enabling Linux and deterministic architectures in novel ways
  - First SoC FPGA with deterministic, coherent CPU cluster and a deterministic L2 memory subsystem enabling Linux + real-time applications
  - First SoC FPGA architecture integrating a RISC-V processor subsystem and low-power FPGA technology
  - PolarFire SoC addresses the industry’s need for a mid-range low-power SoC FPGA with high levels of security and reliability
- Developers can begin development today
  - antmicro Renode platform for software development
  - PolarFire SoC Embedded Experts Development Kit for hardware development
  - New Mi-V Embedded Experts Network

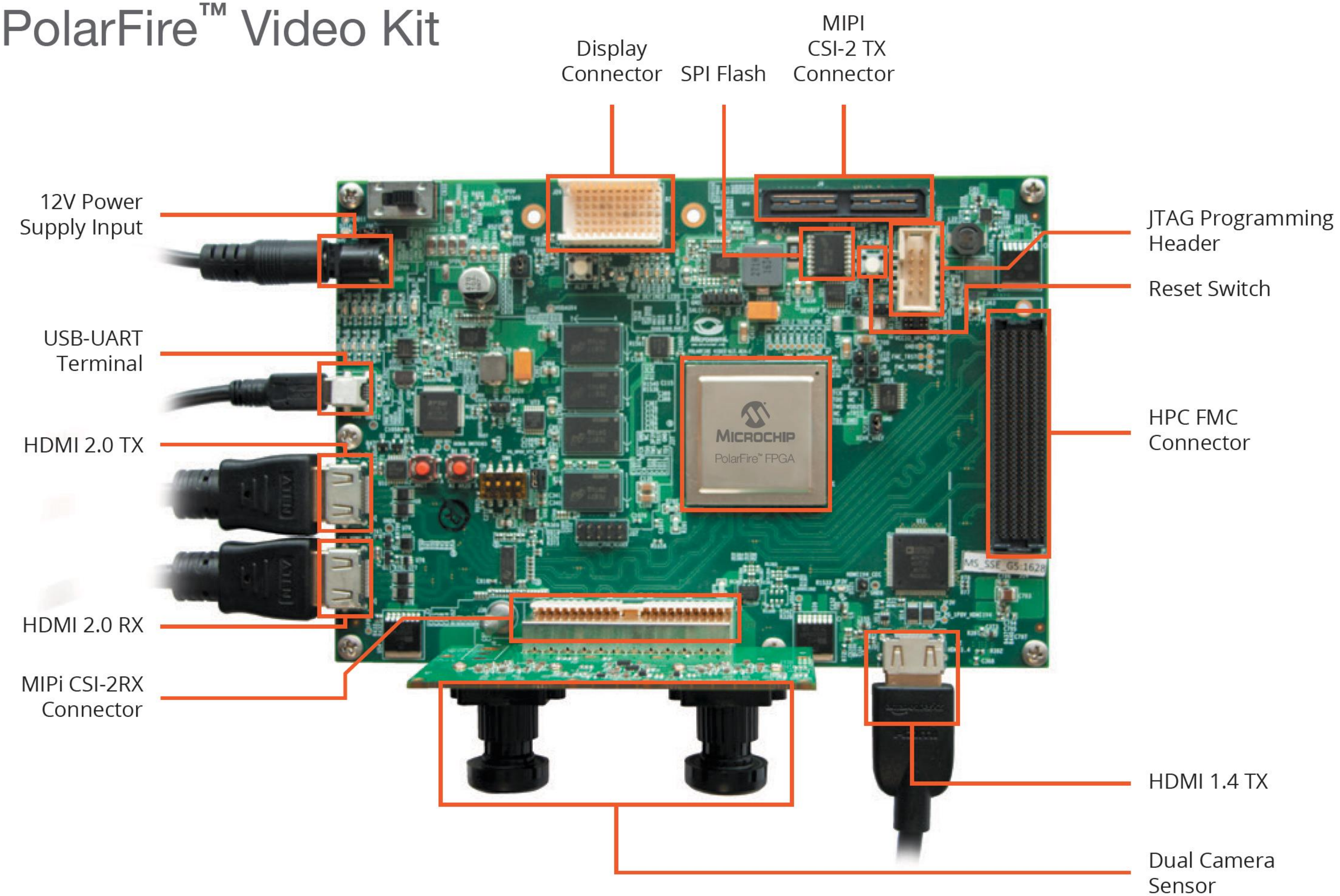
PolarFire Dual Camera Video Kit

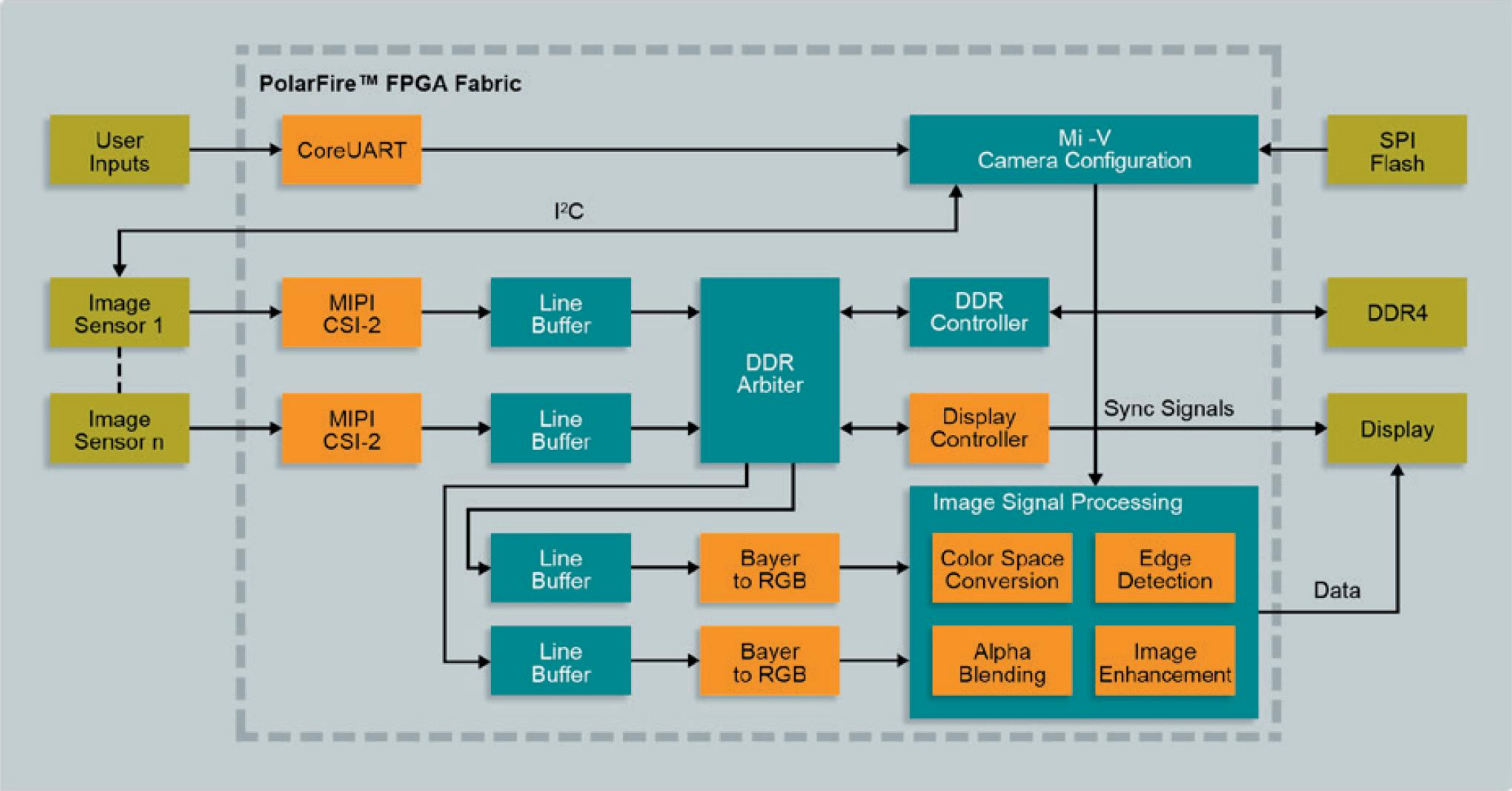
Example using a soft RISC-V

Imaging and Video Solutions for Smart Embedded Vision

- Based on PolarFire™
- Ideal for mid-bandwidth (4K/2K) imaging/video
- Up to 50 percent lower power compared to SRAM-based FPGAs
- Target Applications
  - Machine learning
  - Automotive
  - Surveillance
  - Medical imaging

PolarFire™ Video Kit

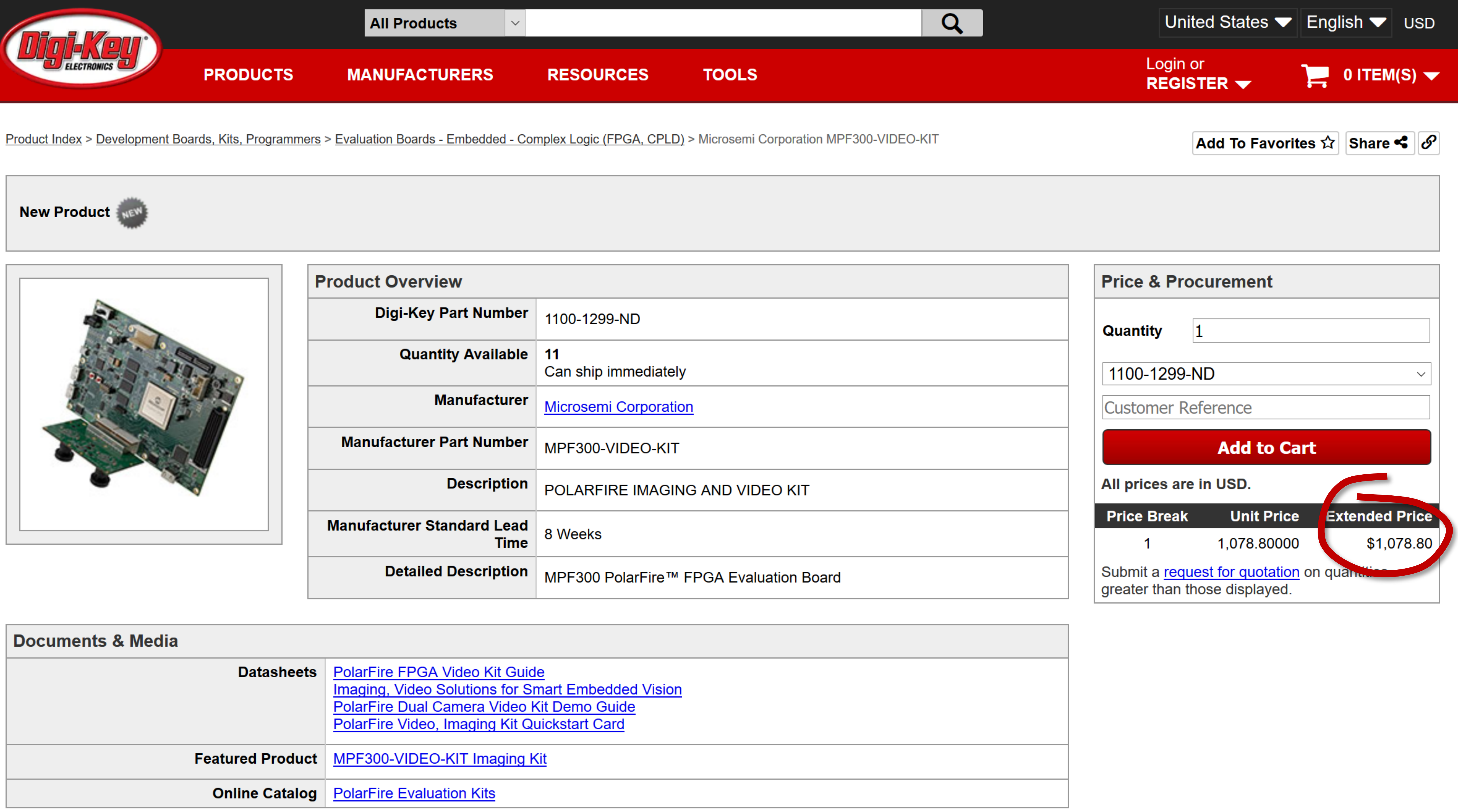



PolarFire™ FPGA Fabric
User Inputs
CoreUART
Mi -V Camera Configuration
SPI Flash
I²C
Image Sensor 1
MIPI CSI-2
Line Buffer
DDR Arbiter
DDR Controller
DDR4
Image Sensor n
MIPI CSI-2
Line Buffer
Display Controller
Sync Signals
Display
Image Signal Processing
Line Buffer
Bayer to RGB
Color Space Conversion
Edge Detection
Line Buffer
Bayer to RGB
Alpha Blending
Image Enhancement
Data

Feature	PolarFire™
Performance	● Tranceivers@12.7G, DDR4@1.6G, CSI-2 Rx@1.0G, LVDS@1.6G ● 1657 GMAC/s Max DSP Performance
Resolution	Up to 4K (3840 x2160)
Protocol Support	MIPI CSI-2 (RX, TX),HDMI 2.0 ,3G SDI,DSI
Speed and Frame Rate	● MIPI: RX at 4.8 Gbps (4 x 1.0 Gbps), TX at 3.2 Gbps (4x 800 Mbps) ● 4K @ 30 fps

Board Components and Operations

- PF_Dual_Camera_Video_Kit_Demo\Microsemi_PolarFire_FPGA_Video_Kit_User_Guide_UG0856_V1.pdf



Product Index > Development Boards, Kits, Programmers > Evaluation Boards - Embedded - Complex Logic (FPGA, CPLD) > Microsemi Corporation MPF300-VIDEO-KIT

New Product

Product Overview

Digi-Key Part Number	1100-1299-ND
Quantity Available	**11** Can ship immediately
Manufacturer	Microsemi Corporation
Manufacturer Part Number	MPF300-VIDEO-KIT
Description	POLARFIRE IMAGING AND VIDEO KIT
Manufacturer Standard Lead Time	8 Weeks
Detailed Description	MPF300 PolarFire™ FPGA Evaluation Board

Price & Procurement

Quantity: 1

1100-1299-ND

Customer Reference

All prices are in USD.

Price Break	Unit Price	Extended Price
1	1,078.80000	$1,078.80

Submit a request for quotation on quantities greater than those displayed.

Documents & Media

Datasheets	PolarFire FPGA Video Kit Guide Imaging, Video Solutions for Smart Embedded Vision PolarFire Dual Camera Video Kit Demo Guide PolarFire Video, Imaging Kit Quickstart Card
Featured Product	MPF300-VIDEO-KIT Imaging Kit
Online Catalog	PolarFire Evaluation Kits

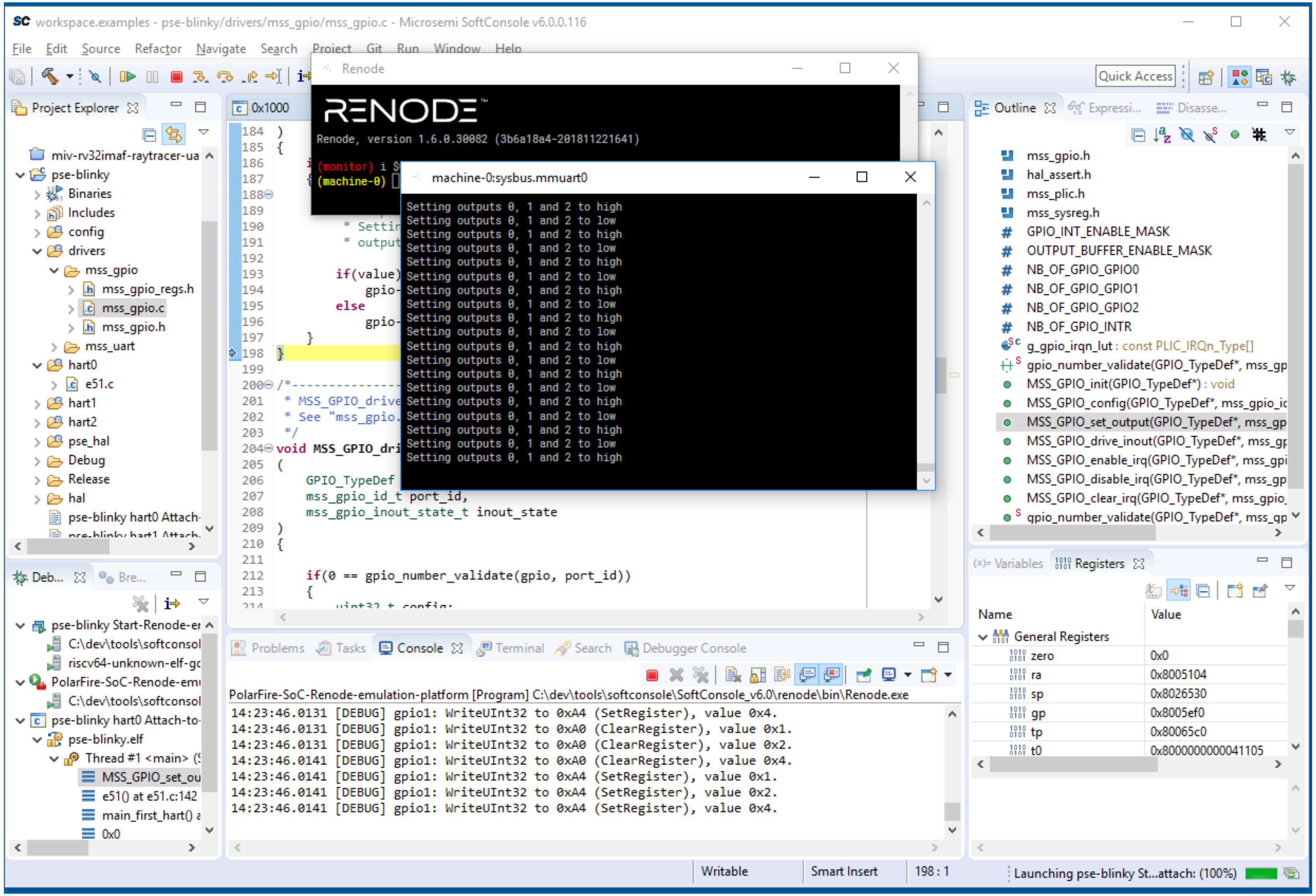
How do you make a design with a Mi-V

How do you make a design with a Mi-V

- The design suite is called Libero SoC
- You can run through a full design flow, from synthesis to programming your device
- Libero includes the firmware catalog to create drivers for IP cores

How do you write a program and run it on the Mi-V

- Softconsole Eclipse IDE required for the development
- Runs on Windows and Linux
- Eclipse-based IDE (Integrated Development Environment)
- Uses OpenOCD and GDB for debugging
- Comes bundled with several sample projects
- Most common demo is called “Blinky” which demos UART, GPIO and Interrupts
- If running correctly, the LEDs on the device should blink on and off, and when the project is first run, it will send a “hello” message over UART
- Softconsole has a built-in terminal emulator
- Resources at github.com/riscv-on-Microsemi-fpga


workspace.examples - pse-blinky/drivers/mss_gpio/mss_gpio.c - Microsemi SoftConsole v6.0.0.116
File Edit Source Refactor Navigate Search Project Git Run Window Help
Quick Access
Project Explorer
miv-rv32imaf-raytracer-ua
pse-blinky
Binaries
Includes
config
drivers
mss_gpio
mss_gpio_regs.h
mss_gpio.c
mss_gpio.h
mss_uart
hart0
e51.c
hart1
hart2
pse_hal
Debug
Release
hal
pse-blinky hart0 Attach-
0x1000
Outline
Expressi...
Disasse...
mss_gpio.h
hal_assert.h
mss_plic.h
mss_sysreg.h
GPIO_INT_ENABLE_MASK
OUTPUT_BUFFER_ENABLE_MASK
NB_OF_GPIO_GPIO0
NB_OF_GPIO_GPIO1
NB_OF_GPIO_GPIO2
NB_OF_GPIO_INTR
g_gpio_irqn_lut : const PLIC_IRQn_Type[]
gpio_number_validate(GPIO_TypeDef*, mss_gp
MSS_GPIO_init(GPIO_TypeDef*) : void
MSS_GPIO_config(GPIO_TypeDef*, mss_gpio_ic
MSS_GPIO_set_output(GPIO_TypeDef*, mss_gp
MSS_GPIO_drive_inout(GPIO_TypeDef*, mss_gp
MSS_GPIO_enable_irq(GPIO_TypeDef*, mss_gpi
MSS_GPIO_disable_irq(GPIO_TypeDef*, mss_gp
MSS_GPIO_clear_irq(GPIO_TypeDef*, mss_gpio_
gpio_number_validate(GPIO_TypeDef*, mss_gp
Deb...
Bre...
pse-blinky Start-Renode-er
C:\dev\tools\softconsol
riscv64-unknown-elf-gc
PolarFire-SoC-Renode-emi
C:\dev\tools\softconsol
pse-blinky hart0 Attach-to-
pse-blinky.elf
Thread #1 <main> (S
MSS_GPIO_set_ou
e51() at e51.c:142
main_first_hart() a
0x0
Variables
Registers
Problems Tasks Console Terminal Search Debugger Console
Writable
Smart Insert
198 : 1
Launching pse-blinky St...attach: (100%)

```
)
{
    i
    {

        * Settir
        * output

        if(value)
            gpio-
        else
            gpio-
    }
}

/*---------------
 * MSS_GPIO_drive
 * See "mss_gpio.
 */
void MSS_GPIO_dri
(
    GPIO_TypeDef
    mss_gpio_id_t port_id,
    mss_gpio_inout_state_t inout_state
)
{

    if(0 == gpio_number_validate(gpio, port_id))
    {
        uint32_t config;
```

Renode

```
RENODE™
Renode, version 1.6.0.30082 (3b6a18a4-201811221641)

(monitor) i $
(machine-0)
```

machine-0:sysbus.mmuart0

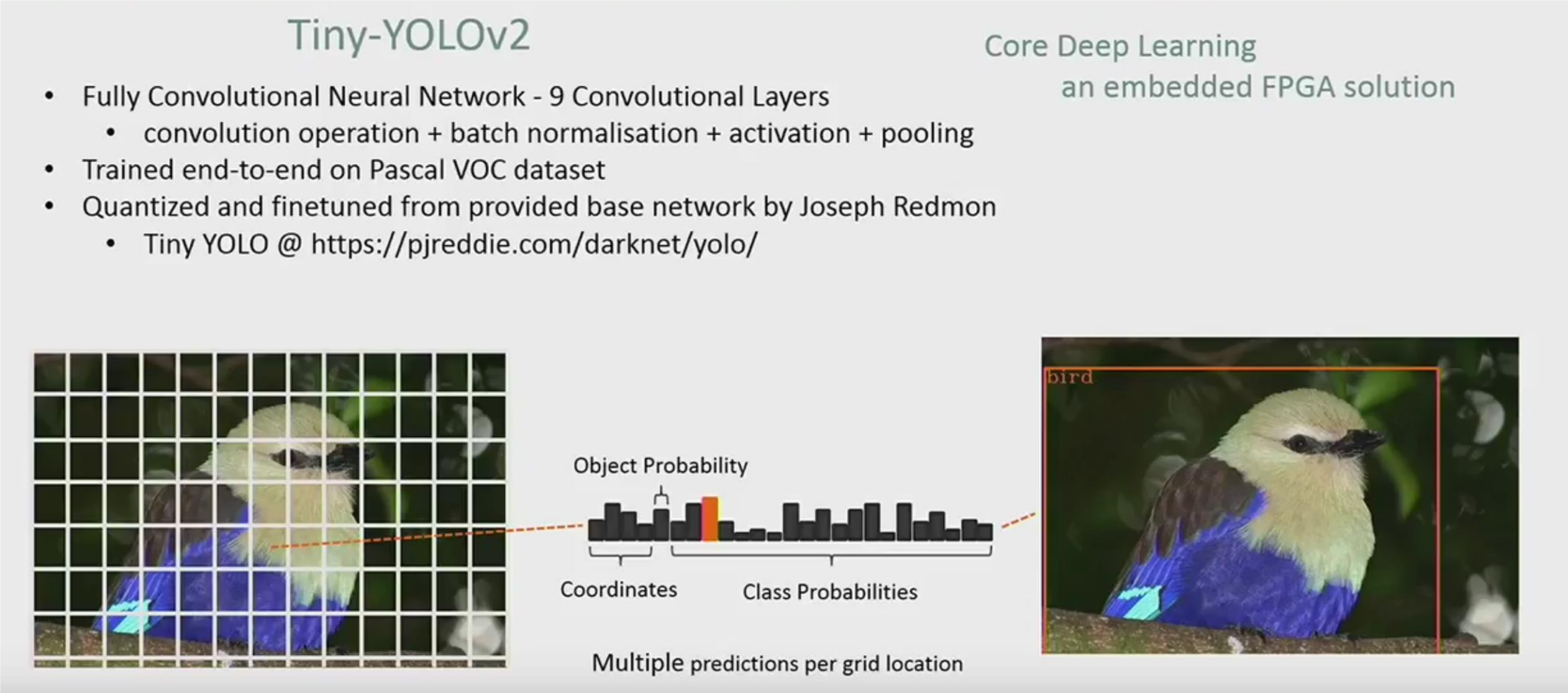
```
Setting outputs 0, 1 and 2 to high
Setting outputs 0, 1 and 2 to low
Setting outputs 0, 1 and 2 to high
Setting outputs 0, 1 and 2 to low
Setting outputs 0, 1 and 2 to high
Setting outputs 0, 1 and 2 to low
Setting outputs 0, 1 and 2 to high
Setting outputs 0, 1 and 2 to low
Setting outputs 0, 1 and 2 to high
Setting outputs 0, 1 and 2 to low
Setting outputs 0, 1 and 2 to high
Setting outputs 0, 1 and 2 to low
Setting outputs 0, 1 and 2 to high
Setting outputs 0, 1 and 2 to low
Setting outputs 0, 1 and 2 to high
Setting outputs 0, 1 and 2 to low
Setting outputs 0, 1 and 2 to high
Setting outputs 0, 1 and 2 to low
Setting outputs 0, 1 and 2 to high
```

PolarFire-SoC-Renode-emulation-platform [Program] C:\dev\tools\softconsole\SoftConsole_v6.0\renode\bin\Renode.exe

```
14:23:46.0131 [DEBUG] gpio1: WriteUInt32 to 0xA4 (SetRegister), value 0x4.
14:23:46.0131 [DEBUG] gpio1: WriteUInt32 to 0xA0 (ClearRegister), value 0x1.
14:23:46.0131 [DEBUG] gpio1: WriteUInt32 to 0xA0 (ClearRegister), value 0x2.
14:23:46.0141 [DEBUG] gpio1: WriteUInt32 to 0xA0 (ClearRegister), value 0x4.
14:23:46.0141 [DEBUG] gpio1: WriteUInt32 to 0xA4 (SetRegister), value 0x1.
14:23:46.0141 [DEBUG] gpio1: WriteUInt32 to 0xA4 (SetRegister), value 0x2.
14:23:46.0141 [DEBUG] gpio1: WriteUInt32 to 0xA4 (SetRegister), value 0x4.
```

Name	Value
General Registers	
zero	0x0
ra	0x8005104
sp	0x8026530
gp	0x8005ef0
tp	0x80065c0
t0	0x8000000000041105

Another Example (ASIC Design Services)



Tiny-YOLOv2

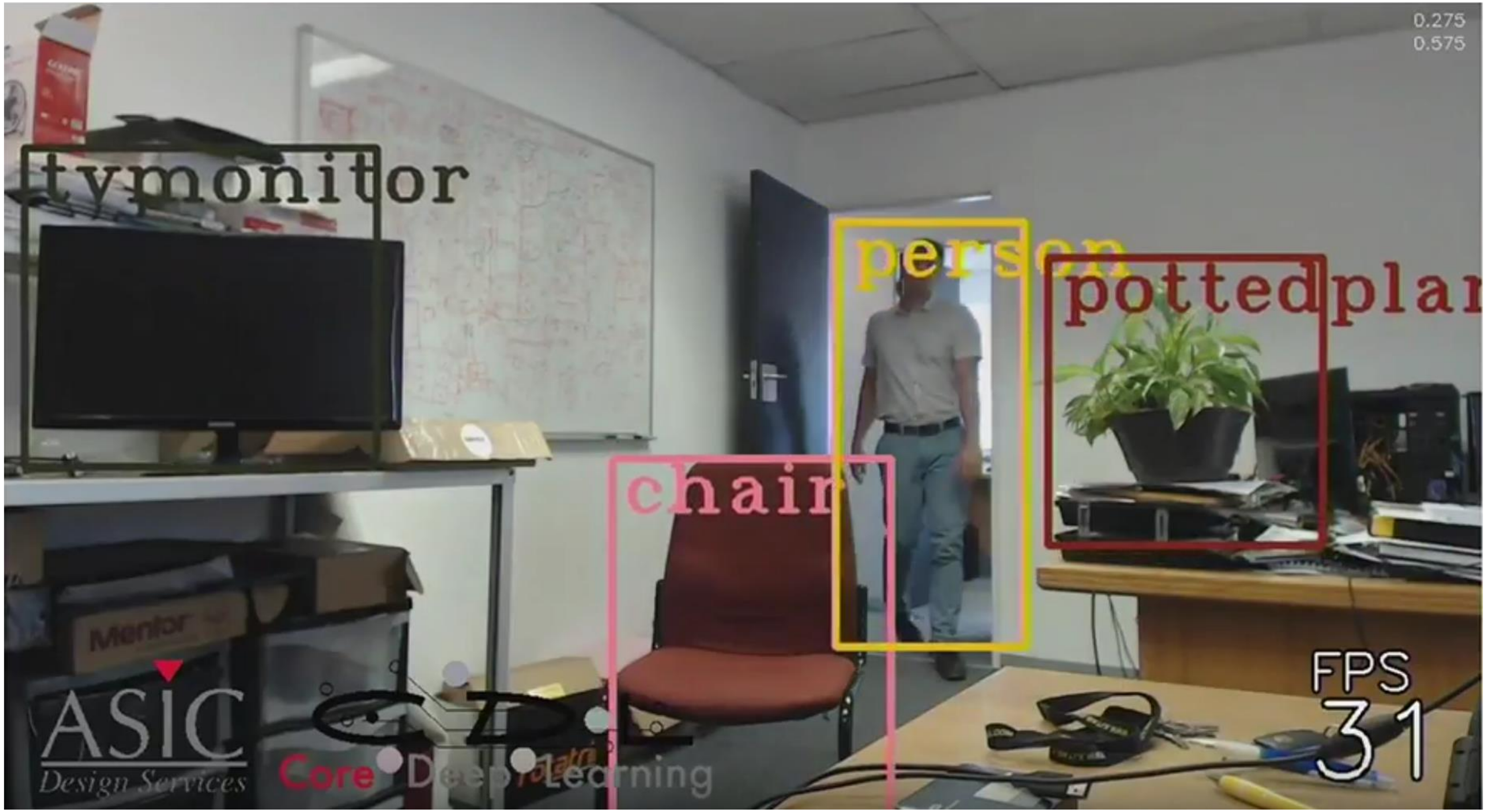
Core Deep Learning
an embedded FPGA solution

- Fully Convolutional Neural Network - 9 Convolutional Layers
  - convolution operation + batch normalisation + activation + pooling
- Trained end-to-end on Pascal VOC dataset
- Quantized and finetuned from provided base network by Joseph Redmon
  - Tiny YOLO @ https://pjreddie.com/darknet/yolo/

Multiple predictions per grid location

Another Example (ASIC Design Services)

- PolarFire + Mi-V in Tiny Yolo Video


0.275
0.575
tvmonitor
person
pottedplar
chair
FPS
31
ASIC
Design Services
Core Deep Learning