



Intel PSG's SoC FPGAs

ARROW
Five Years Out

Your User-Customizable System on Chip

Agenda

- High-Level Overview
- System Architecture Overview
- SoC System Design
- System Tool Flow
- SoCKit – Details of the Kit
- Third-Party Solutions
- Next Steps



Strength in numbers

- **Centennial, CO, USA**

HEADQUARTERS

- **ARW**

(NYSE)

TICKER SYMBOL

- **\$29 billion**

SALES

- **110**

FORTUNE 500 RANK

- **336 worldwide**

LOCATIONS

- **19,300**

EMPLOYEES

- **175,000+**

CUSTOMERS

- **1935**

FOUNDED



Intel PSG SoC High-Level Overview

Your User-Customizable System on Chip

First Things First: What Is An FPGA?

FPGA stands for Field Programmable Gate Array

- Field programmable
 - Program (and re-program) product features and functions.
 - Reconfigure hardware for specific applications even after the device is deployed in the field.
- Gate Array – a configurable array of basic building blocks within a specific device
 - Logic Elements
 - DSP Blocks
 - Memory
 - PLLs and clocking resources

Why use FPGAs?

- Flexible
- Customizable
- Low development cost (no expensive ASIC development tools)
- Fast time to market (quickly make modifications to an existing design)

Why do engineers love FPGAs?

- Digital Etch-a-Sketch (no more mistakes, just re-program and start over).
- Easy for anybody to get started with free tools and inexpensive dev kits.



Intel PSG Device Series Overview



*Lowest Cost,
Lowest Power*



*Cost/Power
Balance*



*Mid-range
FPGAs & SoCs*



*Optimized for
Bandwidth*



*Built for the
Data-Centric Era*

***Foundational Across
All Segments***

***Record
Growth,
Yet to Peak***

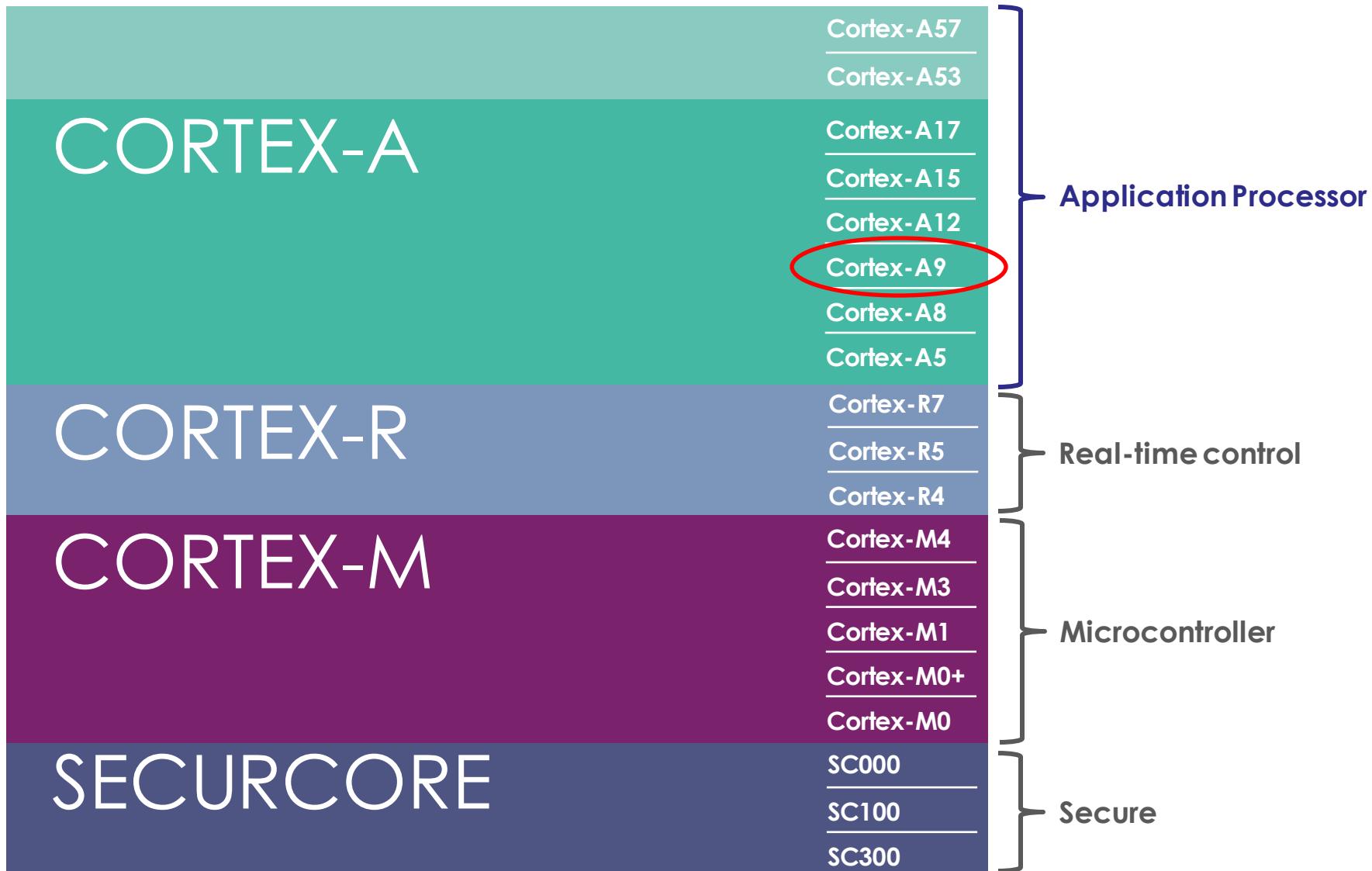
***Just
Starting***

Intel® SoC FPGAs: The Best of Both Worlds



Arm® Cortex® processor + Intel® FPGA

ARM Public Processor Offering



SoC FPGA Key Features

Intel® FPGA	Cyclone® V	Arria® V	Arria 10	Stratix® 10
Processor	Dual core Arm*-Cortex* A9			Quad Core Arm-Cortex A53
Processor performance	925 MHz	1.05 GHz	1.5 GHz	1.35 GHz
Memory controller support	Up to DDR3 400 MHz	Up to DDR3 533 MHz	Up to DDR4 1066 MHz	Up to DDR4 1200 MHz
Logic Density	25-110 KLE	350-460 KLE	160-660 KLE	400K-5.5M LE
Transceivers speed	Up to 6 Gbps	Up to 10 Gbps	Up to 17 Gbps	Up to 28 Gbps

HPS Architectural Differences: Arria10, Stratix10, Agilex

Feature	Intel® Arria® 10 SoCs	Intel® Stratix® 10 SoCs	Intel® Agilex™ FPGAs
Microprocessor Core	Dual 32-bit Cortex*-A9	Quad 64-bit Cortex-A53	Quad 64-bit Cortex-A53
Security	Security Manager inside the HPS	Secure Device Manager (SDM) manages authentication independent of HPS	Secure Device Manager (SDM) manages authentication independent of HPS
System MMU	No	Yes	Yes
Cache Coherency	Accelerator Coherency Port (ACP), 64-bit switching	Cache Coherency Unit (CCU), 128-bit switching, directory based	Cache Coherency Unit (CCU), 128-bit switching, directory based
FPGA-to-SOC (targeting CCU)	32, 64 or 128 bits	128 bits	256 bits
FPGA-to-SOC (targeting SDRAM)	Two by: 32, 64 or 128 bit One by: 32 bit	Three by: 32, 64, or 128 bit	One by: 128, 256, or 512 bit
Hardware-Assisted CPU Virtualization	No	Yes	Yes

28nm SoC Device Family Package Plan

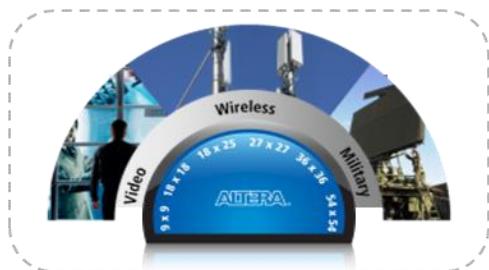
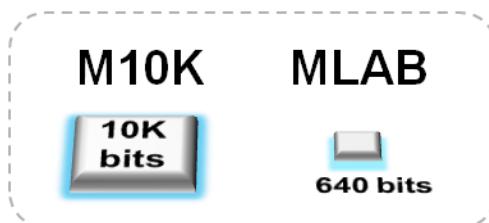
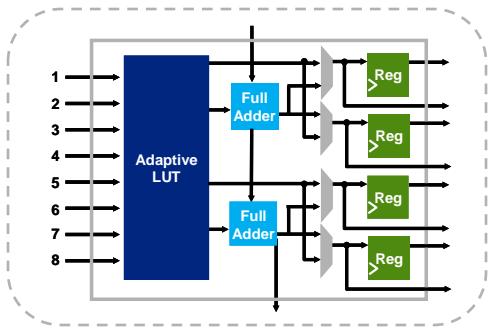
Family	KLE	Non-XCVR Devices (FPGA User I/Os)			XCVR Devices (FPGA User I/Os,XCVRs)				
		U484-WB 19x19	U672-WB 23x23	F896-WB 31x31	U672-WB 23x23 (IO,3G)	F896-WB 31x31 (IO,3G/5G)	F896-FC 31x31 (IO, 6G, 10G)	F1152-FC 35x35 (IO, 6G, 10G)	F1517-FC 40x40 (IO, 6G, 10G)
Cyclone V SoC FPGA	25	66	138		138,6				
	40	66	138		138,6				
	85	66	138	288	138,6	288,9			
	110	66	138	288	138,6	288,9			
Arria V SoC FPGA	350						170,12,4	350,18,8	528,30,16
	460						170,12,4	350,18,8	528,30,16
HPS I/O		161	188	188	188	188	216	216	216

Today's Workshop



V | Five Years Out

Intel PSG 28nm FPGA Core Architecture



■ Enhanced ALM architecture

- Up to 30% increase in logic packing efficiency vs. 4-LUT
- 2x registers in each ALM for easier timing closure

■ Variety of memory structures

- 10K blocks optimized for highest port count
- 640-bit MLABs for efficient data buffers

■ Variable precision DSP block

- One block equals either :
 - One 27x27 multiplication
 - Two 18x19 multiplications
 - Three 9x9 multiplications
- Up to 1,600 GMACS, 300 GFLOPS in Arria V SoC
- Up to 150 GMACS, 100 GFLOPS in Cyclone V SoC

Cyclone V has inherited a high-end FPGA architecture

System Architecture Overview

Your User-Customizable System on Chip

System Architecture

Processor

- Dual-core ARM® Cortex™-A9 MPCore™ processor
- 4,625 MIPS (up to 925 MHz per core)
- NEON coprocessor with double-precision FPU
- 32-KB/32-KB L1 caches per core
- 512-KB shared L2 cache

Multiport SDRAM controller

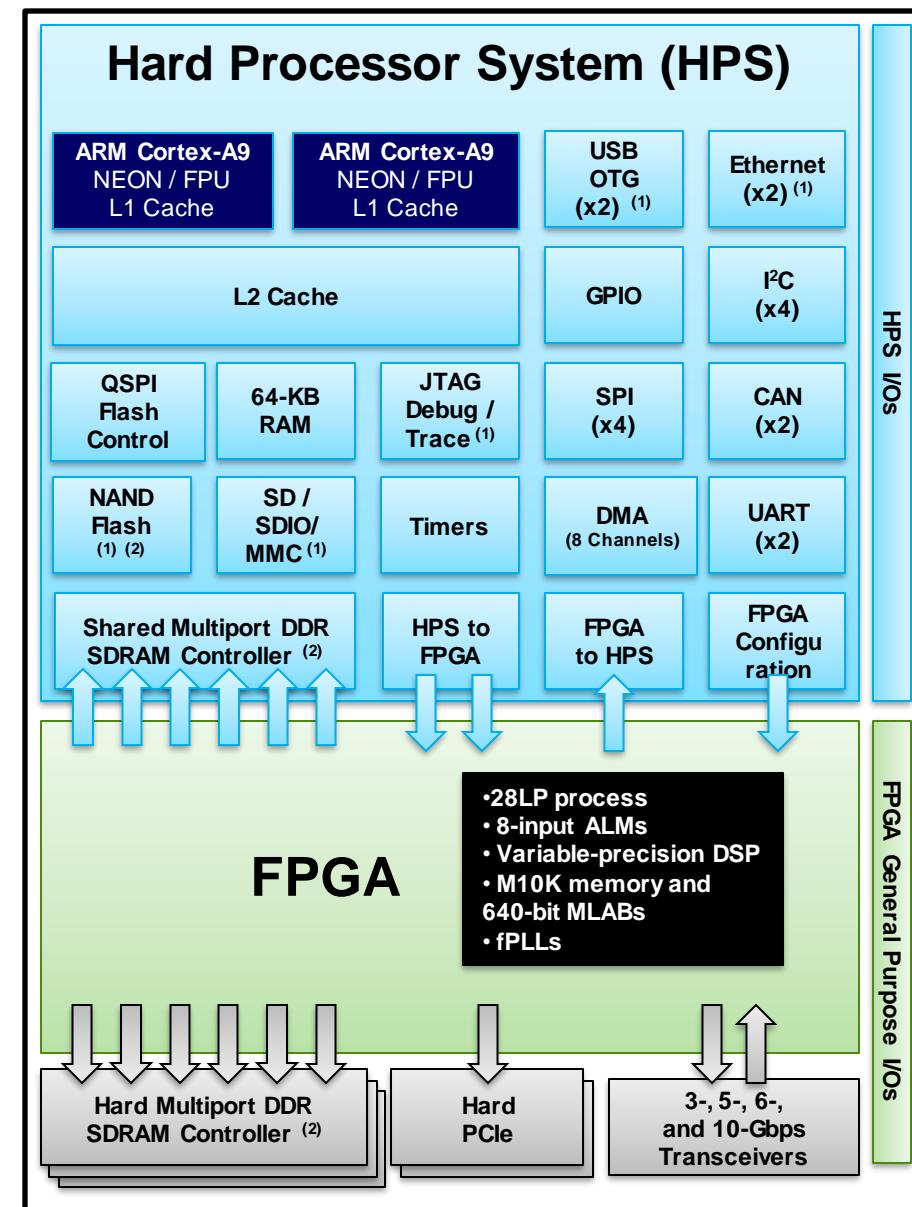
- Up to 533-MHz DDR3 and LPDDR2
- Up to 400-MHz DDR2
- Integrated ECC support

High-bandwidth on-chip interfaces

- Dual HPS-to-FPGA bridge architecture
- FPGA-to-HPS interface
- FPGA-to-SDRAM interface

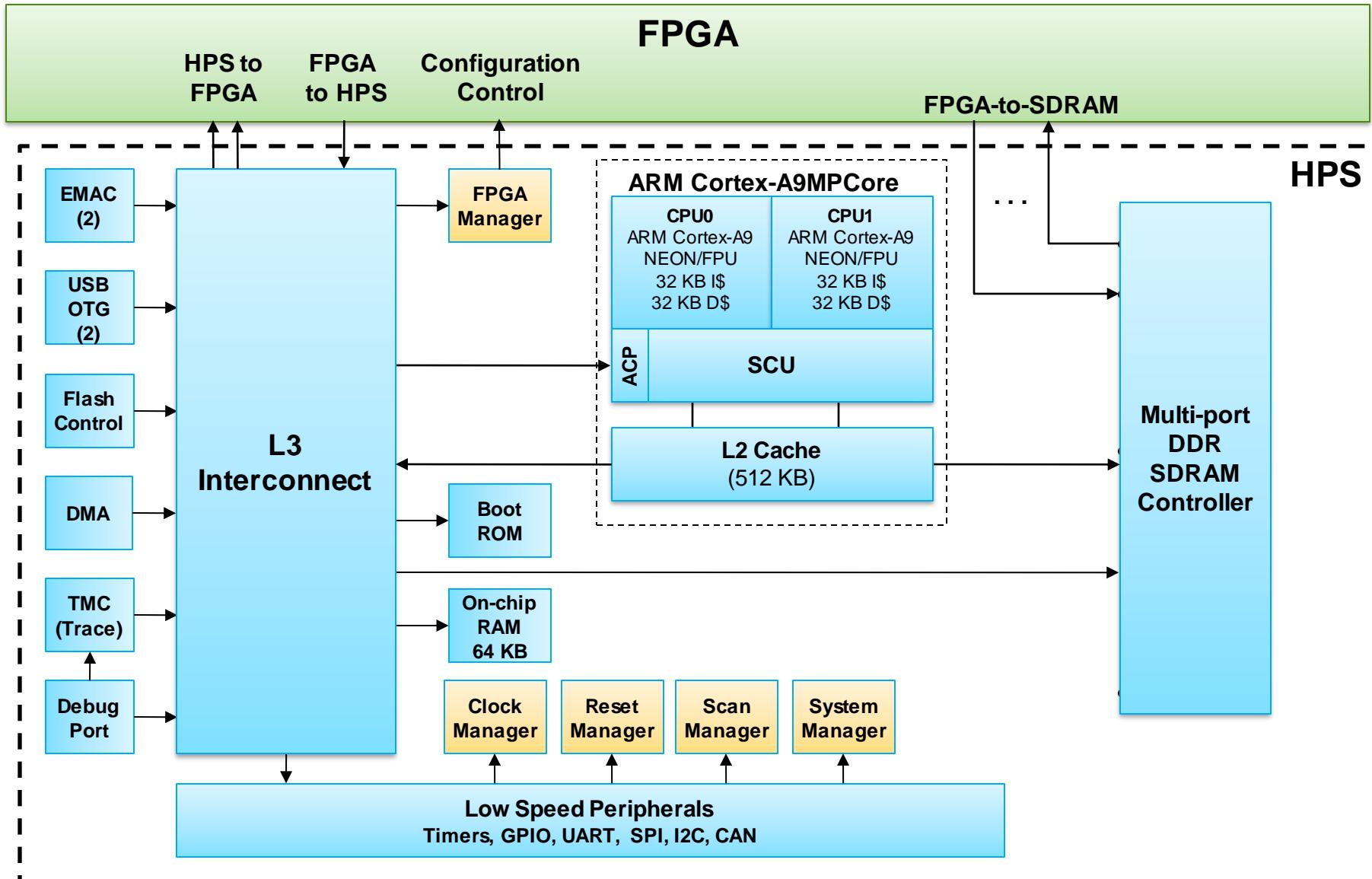
Cost- and power-optimized FPGA fabric

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

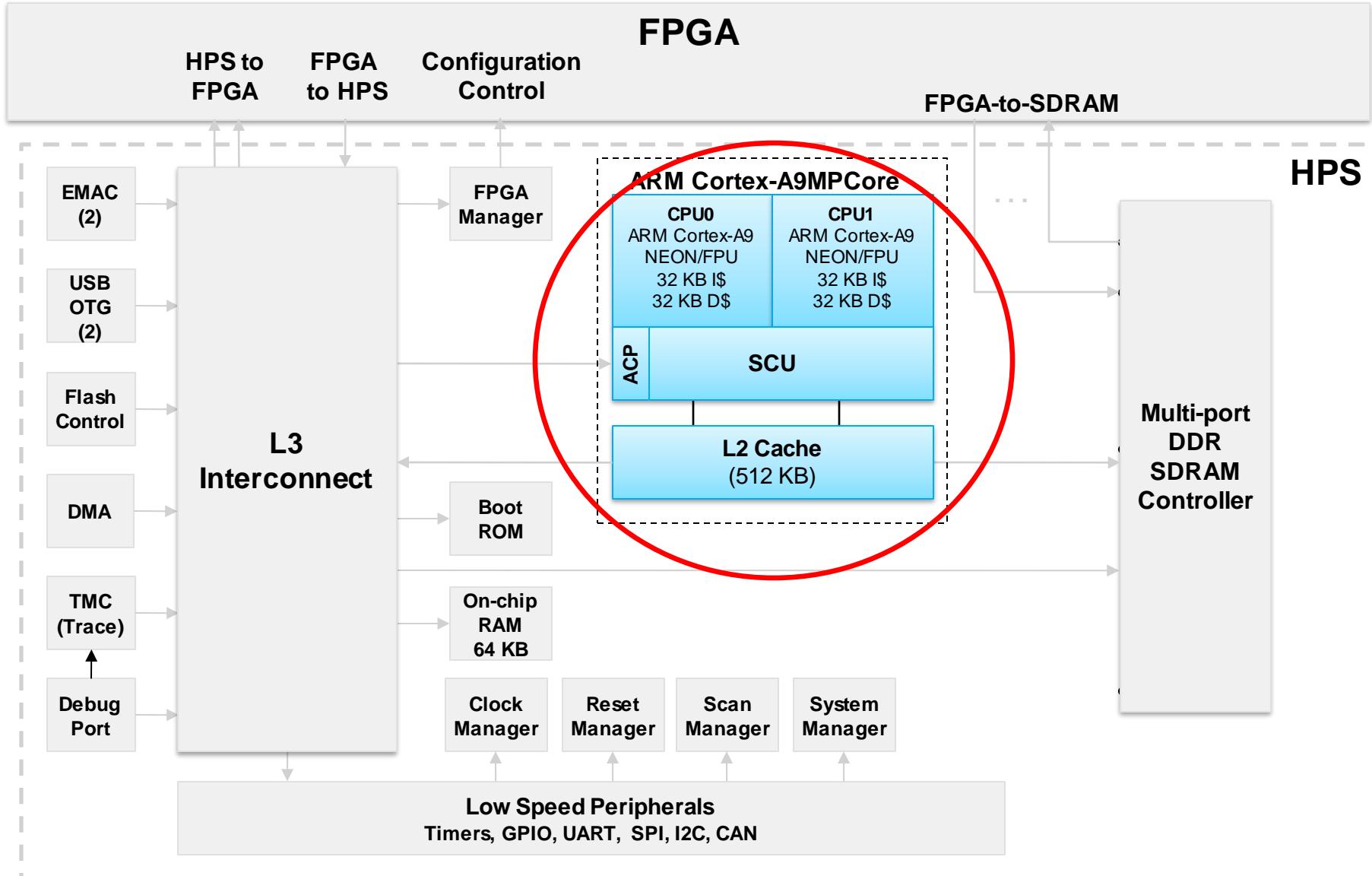


Notes: (1) Integrated DMA, (2) Integrated ECC

Hard Processor System Block Diagram



MPU Subsystem Block Diagram



Components of the MPU Subsystem

- Cortex-A9 MPU subsystem contains:

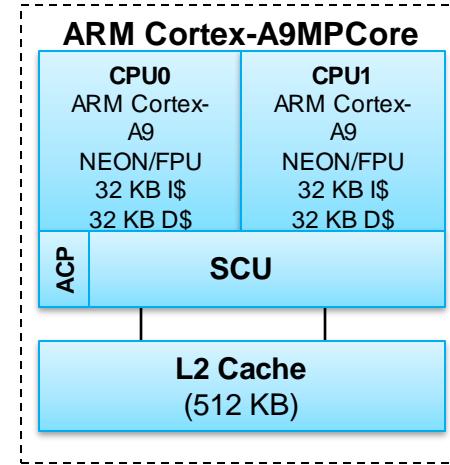
- Cortex-A9 MPCore
 - Level 2 cache
 - Debugging module (*not shown*)
 - Accelerator coherency port (ACP) ID mapper (*not shown*)

- Cortex-A9 MPCore contains:

- Dual Cortex-A9 processor cores
 - Snoop control unit (SCU) with Accelerator Coherency Port (ACP)
 - Global interrupt controller (GIC)
 - Global timer
 - Private timers and watchdogs

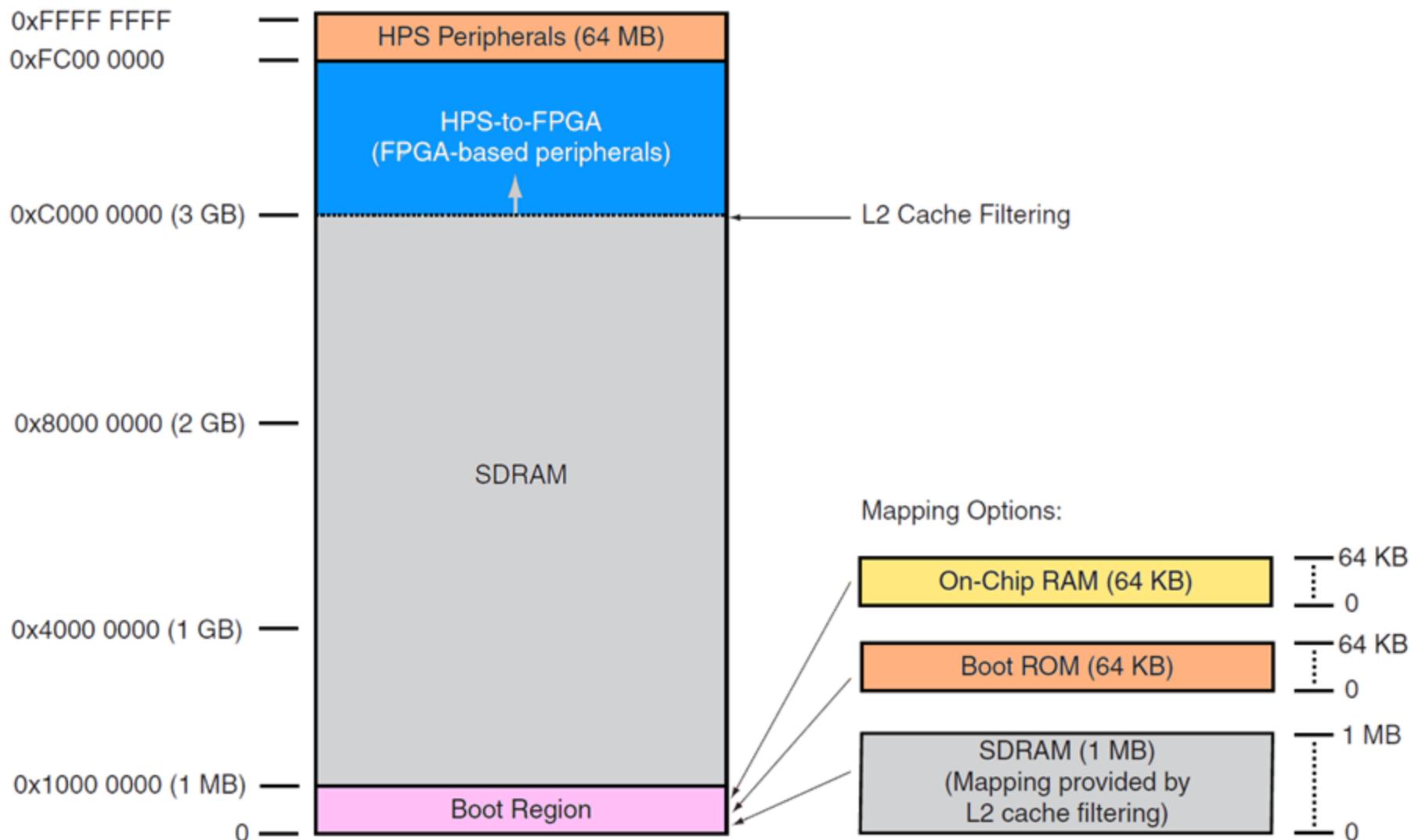
- Cortex-A9 processor core contains:

- ARM Cortex-A9 core
 - NEON™ SIMD engine with scalar FPU
 - Level 1 instruction and data caches (32KB per cache)
 - Memory management unit (MMU)



Cortex A9 MPU Memory Map

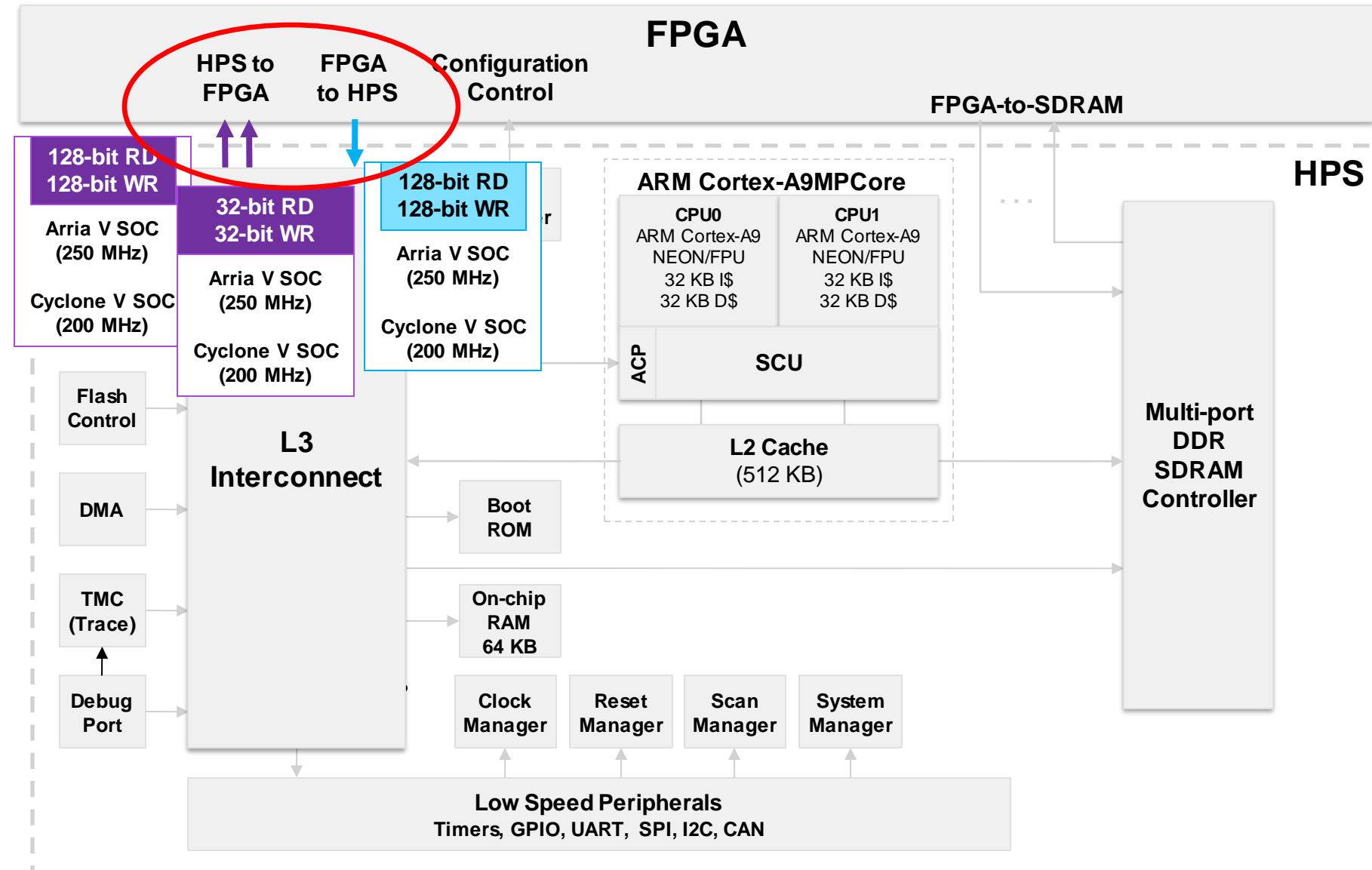
Addresses are not to scale.



Cortex-A9 MPU Memory Views



FPGA-HPS Interfaces



FPGA-HPS Interfaces

- HPS contains three AXI bridges
 - FPGA-to-HPS (F2H) → FPGA logic masters HPS
 - HPS-to-FPGA (H2F) → HPS accesses high speed FPGA slaves
 - Lightweight HPS-to-FPGA (LWH2F) → HPS accesses CSRs
- Each bridge provides clock crossing
 - FPGA ports clocked independently from HPS ports
 - Asynchronous clock crossing
- F2H and H2F bridges support variable widths
 - FPGA port variable width between 32 / 64 / 128-bit
 - HPS port fixed at 64-bit and up to 400MHz
- LWH2F bridge implemented with fixed 32-bit data
 - CSRs typically 32-bit



Dual HPS-to-FPGA Bridge Approach

- Two main types of HPS-to-FPGA traffic
 - (Slow) Control/Status Register (CSR) accesses to peripherals
 - (Fast) Accesses to memories and hardware accelerators
- Dual bridges used to support two types of traffic
 - LWH2F bridge used for short – latency sensitive accesses
 - 32-bit CSR data
 - 32-bit wide interface
 - H2F bridge used for bursting – latency insensitive accesses
 - MPU, DMA or peripheral data transactions
 - Up to a 128-bit wide interface
- Historically, architectures having only a single bridge could either cause latency issues with CSR accesses or throughput issues with high bandwidth transfers.
- Providing dual bridges allows system to maintain low-latency accesses to CSRs in FPGA independent of large data transfers that might be occurring simultaneously on the H2F bridge.

When Architecture Matters

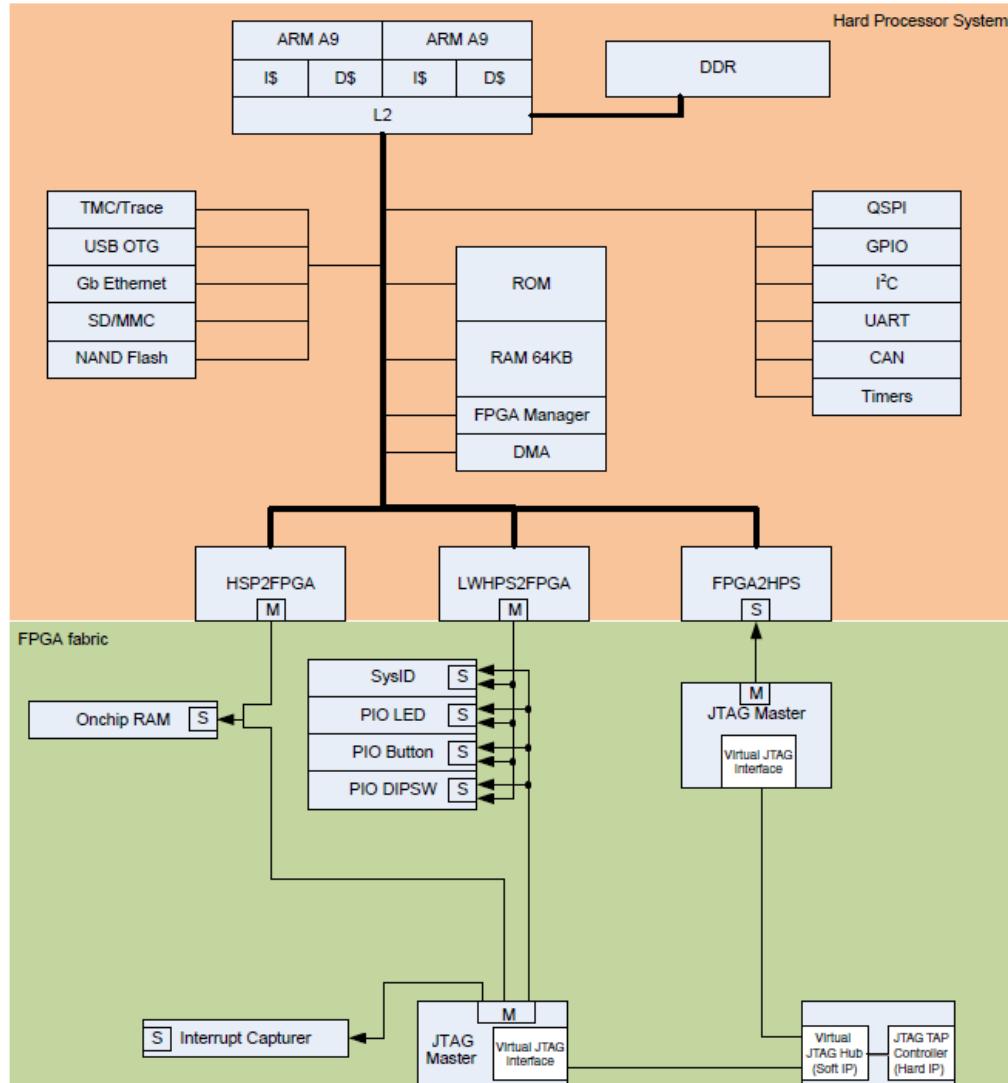


V | Five Years Out

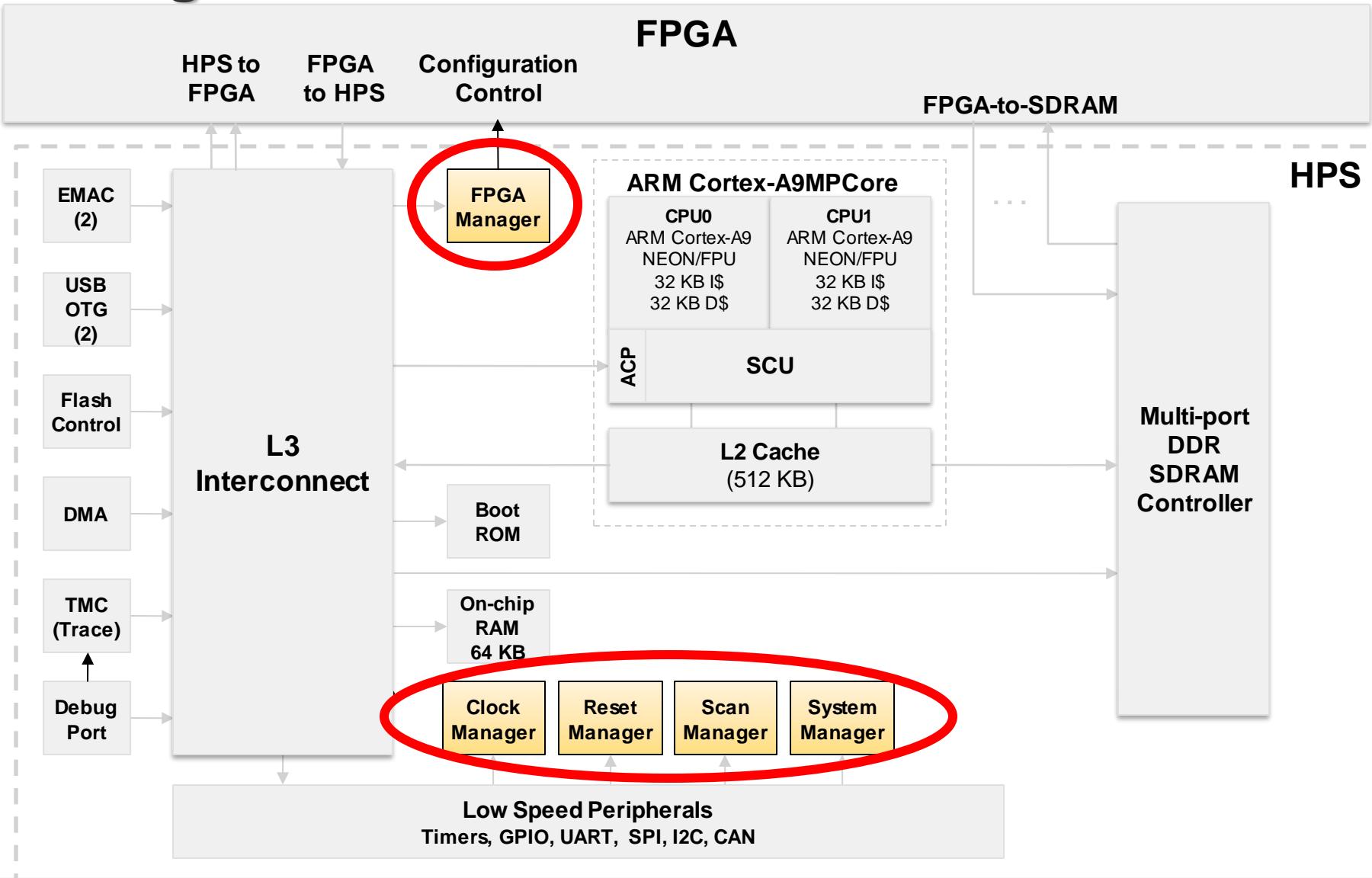
FPGA-HPS Interfaces Workshop Example

In the workshop, you will build the simple example system shown here.

- H2F bridge connects to an On-Chip Memory
- LWH2F bridge connects to a few Parallel IO and System ID peripherals
- F2H bridge is connected to a JTAG Master peripheral



System Management



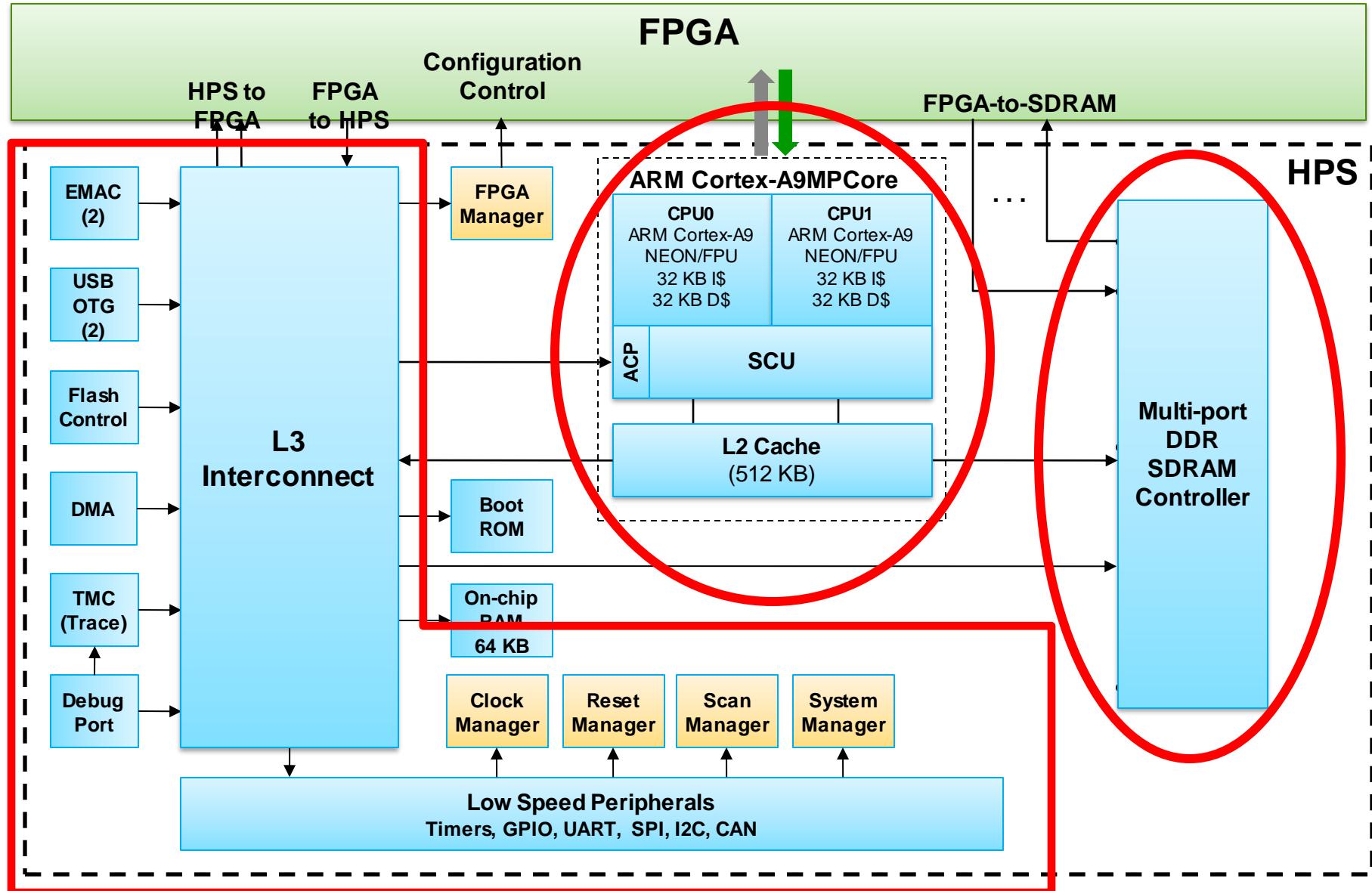
HPS Input Clocks

- HPS requires an external clock source (OSC1, 10-50 MHz)
- Optional second source (OSC2)
- External Clocks connect to 3 PLLs that drive clock groups inside HPS
- FPGA clocks are the same as in Cyclone V
- Optional clocks to and from FPGA
 - HPS-to-FPGA clocks 0/1/2, one from each PLL group
 - FPGA-to-HPS peripheral pll reference clock
 - FPGA-to-HPS SDRAM pll reference clock

Clock Manager

- Generates/manages all system clocks in HPS
- 3 Clock Groups, each with it's own PLL:
 - Main PLL
 - Peripheral PLL
 - SDRAM PLL
- FPGA-to-HPS clocks are enabled/disabled by the Clock Manager but are controlled from the FPGA not the Clock Manager
- Hardware sets the base clock.
- Software sets up the PLL dividers and VCO frequencies. This task is handled by the preloader.

3 HPS Clock Groups



FPGA interface Clocks

- **FPGA Clocks Drive:**

- FPGA AXI Bridges (clock crossing)
- Optional Clock inputs to Peripheral and SDRAM group PLLs

- **3 HPS Clocks routed to FPGA**

- One from each PLL

Clock Manager - PLLs

- Software programmable clock control
 - Clock routing
 - PLL configuration
 - VCO speed
 - Clock Divider setting (per output)
 - Clock output phase (SDRAM group only)
 - Clock gating
- Each PLL can trigger the Clock Manager Interrupt
 - PLL lock / PLL loss of lock
- Clock Manager can trigger a MPU Wake Up Interrupt when clock hardware is stable

HPS Clock Manager Features

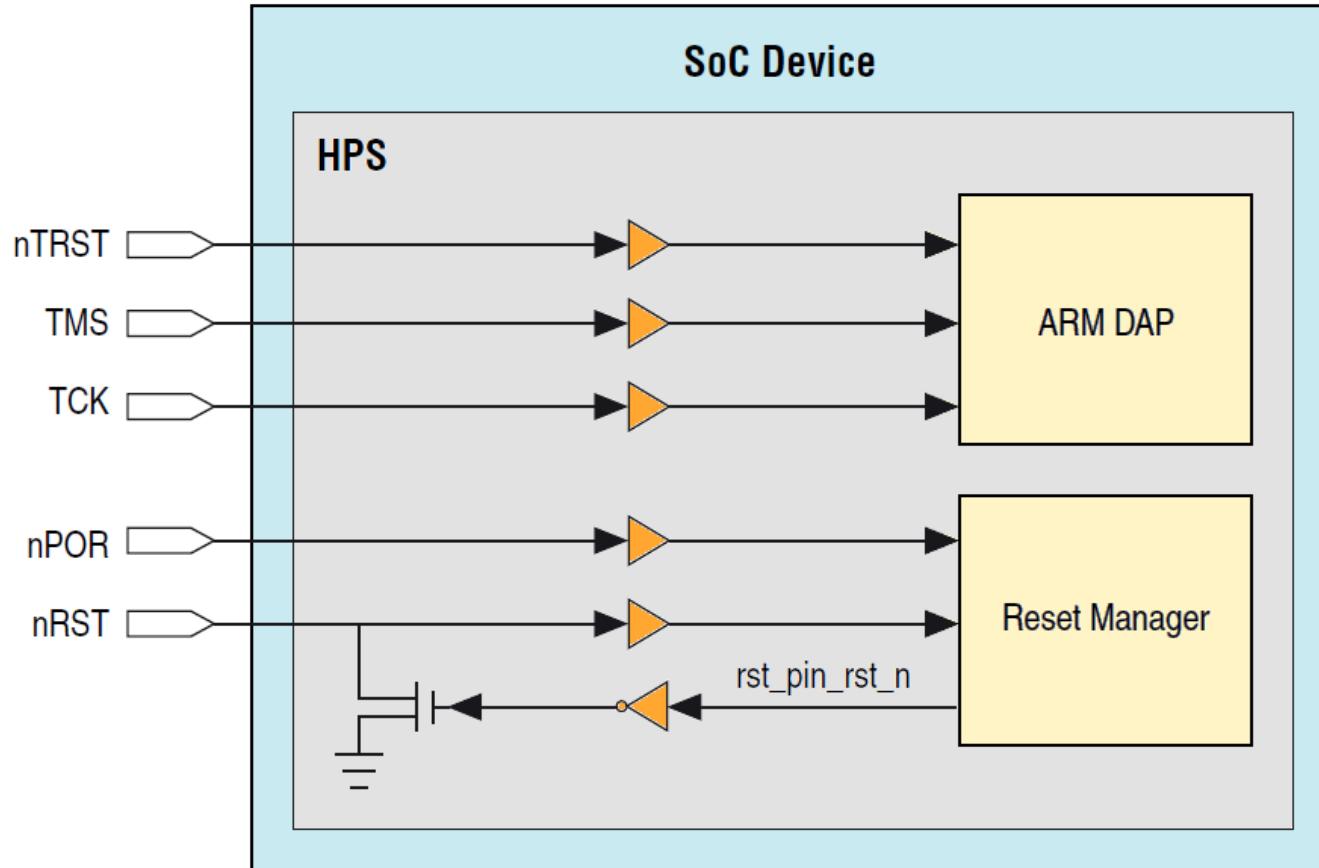
- Enables dynamic scaling of MPU clock frequency
- Delivers base clocks to Ethernet, USB, SDRAM, etc.
- Enables ~all clocks to be gated off (power saving)
- Entry/Exit of ‘Safe Mode’
 - Put clocks into known safe state on cold reset or warm reset request from Reset Manager
 - Bypasses Main PLL C0-C2 over to OSC1
 - Clock manager registers settings, for reset PLL counter & dividers, bypassed to default values
 - Enables all clocks (turns clock gating off)
 - Flash controller clock multiplexer selects output from peripheral PLL
 - Exit Safe Mode by resetting *Safe Mode* bit of *CTRL* register

Clocking Caveats

- Clocks must be very carefully sequenced
- It is possible to lock up system if mishandled
- Intel PSG provides library of functions to assist with clock management. These tasks are included as part of the preloader.
- For more information refer to Volume 3, chapter 2 of the device handbook: [Clock Manager](#)



SoC Device Reset Pins



Cold / Warm / Debug Reset

Cold Reset	Warm Reset	Debug Reset
Affects all reset domains (TAP, Debug, System)	Happens after HPS has already been cold reset	
Places hardware-managed clocks into safe mode	Used to recover system from a non-responsive condition	
Places software managed clocks into their default states	Resets a subset of the HPS	Only affects debug reset domain
Asynchronously resets all registers in the clock manager	Debug & TAP reset domain unaffected	
Resets SDRAM so memory contents and setup lost	SDRAM unaffected so memory contents preserved	
FPGA is not affected across any type of reset		



SoC Resets

- HPS cold and warm resets can be sent into FPGA logic
- FPGA cold and warm resets can be received by HPS
 - FPGA traffic must terminate before HPS warm reset completes
- Implemented and managed by Reset Manager



Reset Manager Overview

- Generates module reset signals based on reset requests from various sources
- Clocked by EOSC1 clock pin
- Three separate reset domains
 - HPS system
 - JTAG port
 - Debug system
- Accepts reset requests from following
 - Software writing module reset registers
 - FPGA control block
 - FPGA fabric
 - External Reset pins
- Implements Cold and Warm reset



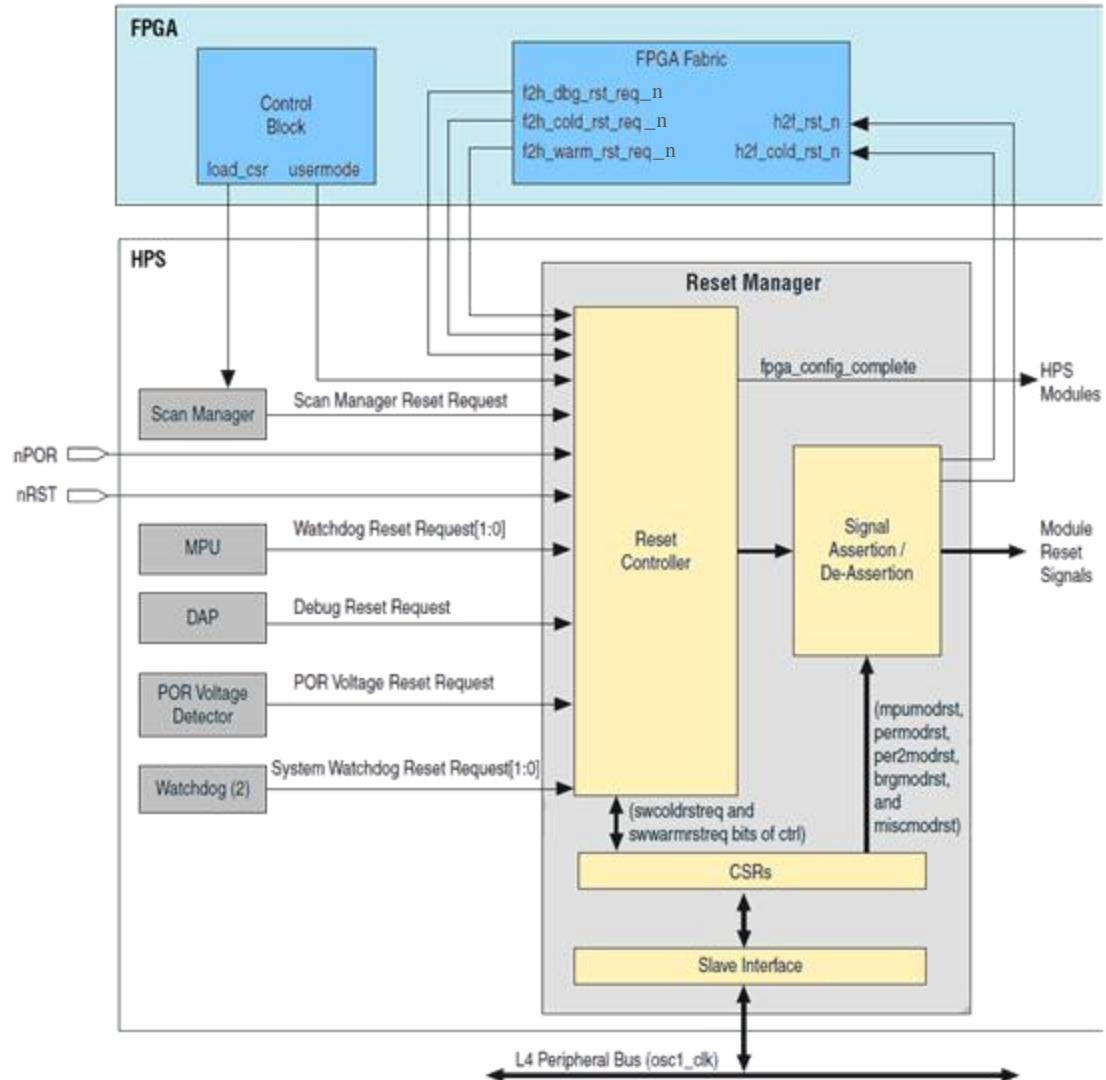
Reset Manager Integration

■ Reset Sources

- Power-On-Reset voltage monitor (cold reset)
- nPOR pin (cold reset)
- nRST pin (warm reset)
- Debug (Debug Reset)
- Watchdogs (SW control)
- FPGA
 - f2h_cold_rst_req_n
 - f2h_warm_rst_req_n
 - f2h_dbg_rst_req_n
 - h2f_cold_rst_n
 - h2f_RST_n (cold or warm)
 - load_csr (cold reset)

■ FPGA monitoring

- Reset from FPGA can occur only if FPGA is configured



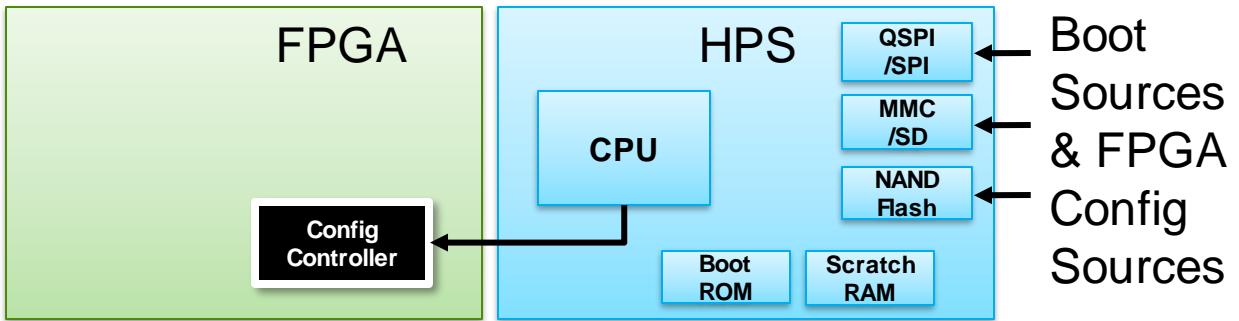
FPGA Manager Overview

- Manages and monitors the FPGA portion of the SoC FPGA device
 - Configure the FPGA fabric
 - Monitor FPGA configuration and power status
 - *INIT_DONE*, *CRC_ERR*, *PR_DONE*, etc.
 - Software configurable interrupts to MPU
 - Can reset the FPGA
 - Drive 32 general purpose outputs to the FPGA
 - Receive 32 general purpose inputs from the FPGA
 - Boot from FPGA handshake signals
 - Used for booting the HPS from the FPGA fabric

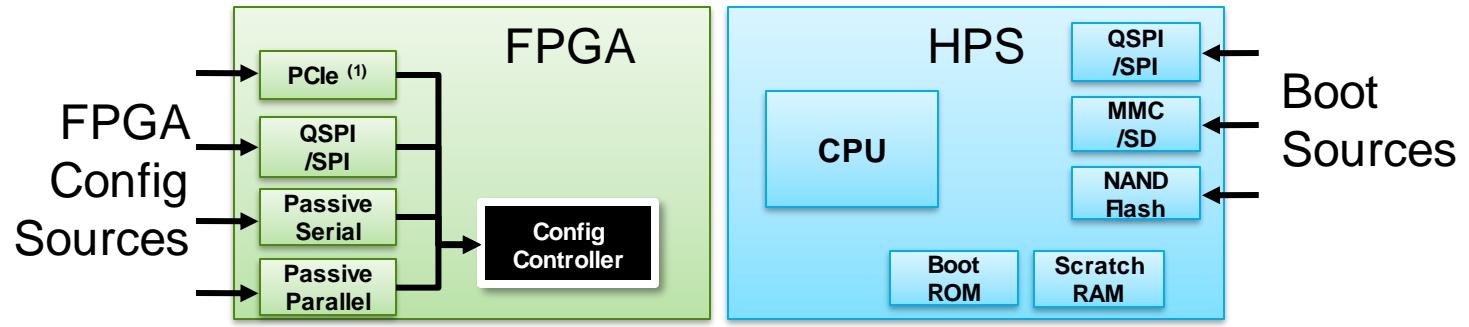


FPGA Configuration / uP Boot : Flexibility

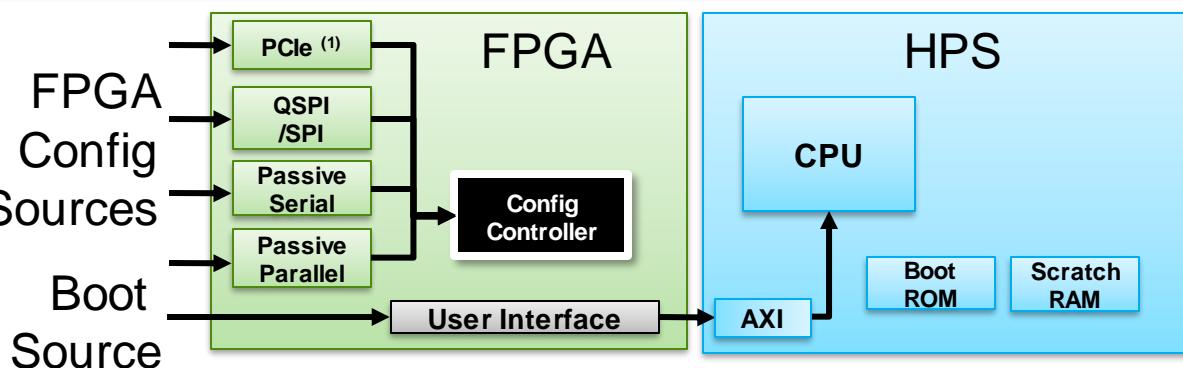
Processor boots first, then configures the FPGA



Independent FPGA config & processor boot



**FPGA config first, CPU boots through FPGA logic
(allows for secure boot)**



System Manager Overview

- Contains memory-mapped control and status registers and logic to:
 - Provide software access to boot configuration information
 - Provide software access to status signals in other HPS modules
 - Enable/disable other HPS modules and Peripherals
 - Enable and controls ECC and parity in HPS modules
 - Control/Manage HPS I/O features
 - Provide eight registers that software can use to pass information between warm boots
 - Pause watchdog timers during debug mode
- Contains a System ID register – can be read from the FPGA side

System Manager HPS I/O Features

- Tri-states HPS-configurable I/O pins during configuration (Freeze controller)
- Controls HPS peripheral I/O pin multiplexing
- Enables/disables various HPS I/O peripherals interfaces
- Enables/disables HPS-FPGA interfaces



System Manager – Managed Peripherals

- EMACs
- USB controllers
- Flash controllers
 - SD/MMC controller
 - NAND controller
 - Quad SPI controller
- SPI masters
- DMA controller
- On-chip RAM
- CAN controllers
- Debug core
- Reset manager



Scan Manager

- **Programs HPS I/O pin configuration**

- Configures I/O bank settings (voltage, drive, pull up, etc.)
- Same IOCSR mechanism as FPGA IO
(scan chains - JTAG Bit-stream)
- HPS I/O must be frozen by System Manager before configuration

- **Gives HPS access to FPGA JTAG**

- Can perform any FPGA JTAG operation
- Disables external FPGA JTAG when it takes over

Summary – Intel PSG Advantages

■ Higher performance

- 2x CPU-to-FPGA & FPGA-to-CPU bandwidth
- 2x FPGA-to-DDR bandwidth
- All masters (FPGA and HPS) can share memory coherently with the processor
- Application-class processing (SMP)

■ Greater memory security & protection

- Broader ECC support in HPS: DDR controller, L2, on-chip RAM, Flash I/F, HPS peripherals
- More robust DDR protection: address range protection per port & per master
- Exclusive memory sharing on all DDR controller ports

■ More design flexibility

- XCVRs/PCIe available in all devices
- Multiple DDR controllers in FPGA
- Finer control over DDR port priorities
- Greater FPGA density
- Can use multiple Flash devices (e.g. QSPI + NAND)
- FPGA / CPU boot options (FPGA first, CPU first, Independent)
- DMA request interfaces for FPGA *and* HPS peripherals
- Coherent memory access for FPGA *and* HPS peripherals
- More processor and system trace options
- No need to add power-down circuitry



SoC System Design

Your User-Customizable System on Chip

Quick Summary

■ **FPGA:**

- Looks like an FPGA
- Works like an FPGA
- Standard FPGA development flow
- QuartusII, Platform Designer, USB Blaster, SignalTap, System Console...

■ **ARM Hard Processor System:**

- Looks like an ARM processor system
- Works like an ARM processor system
- Standard ARM development flow
- ARM Cortex-A9 compiler/debugger, JTAG tools, program trace...



System Development Flow

Standard FPGA Flow



Hardware Development

- Quartus II design software
- Platform Designer system integration tool
- Standard RTL flow
- Intel PSG and partner IP

Design

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

Simulate

- SignalTap™ II logic analyzer
- System Console

Debug

- Quartus II Programmer
- In-system Update

Release



Standard Software Flow



Software Development

- ARM Development Studio 5
- GNU toolchain
- OS/BSP: Linux, VxWorks
- Etc...

Design

- Virtual Target

Simulate

- GNU, Lauterbach, DS5 and ARM ecosystem

Debug

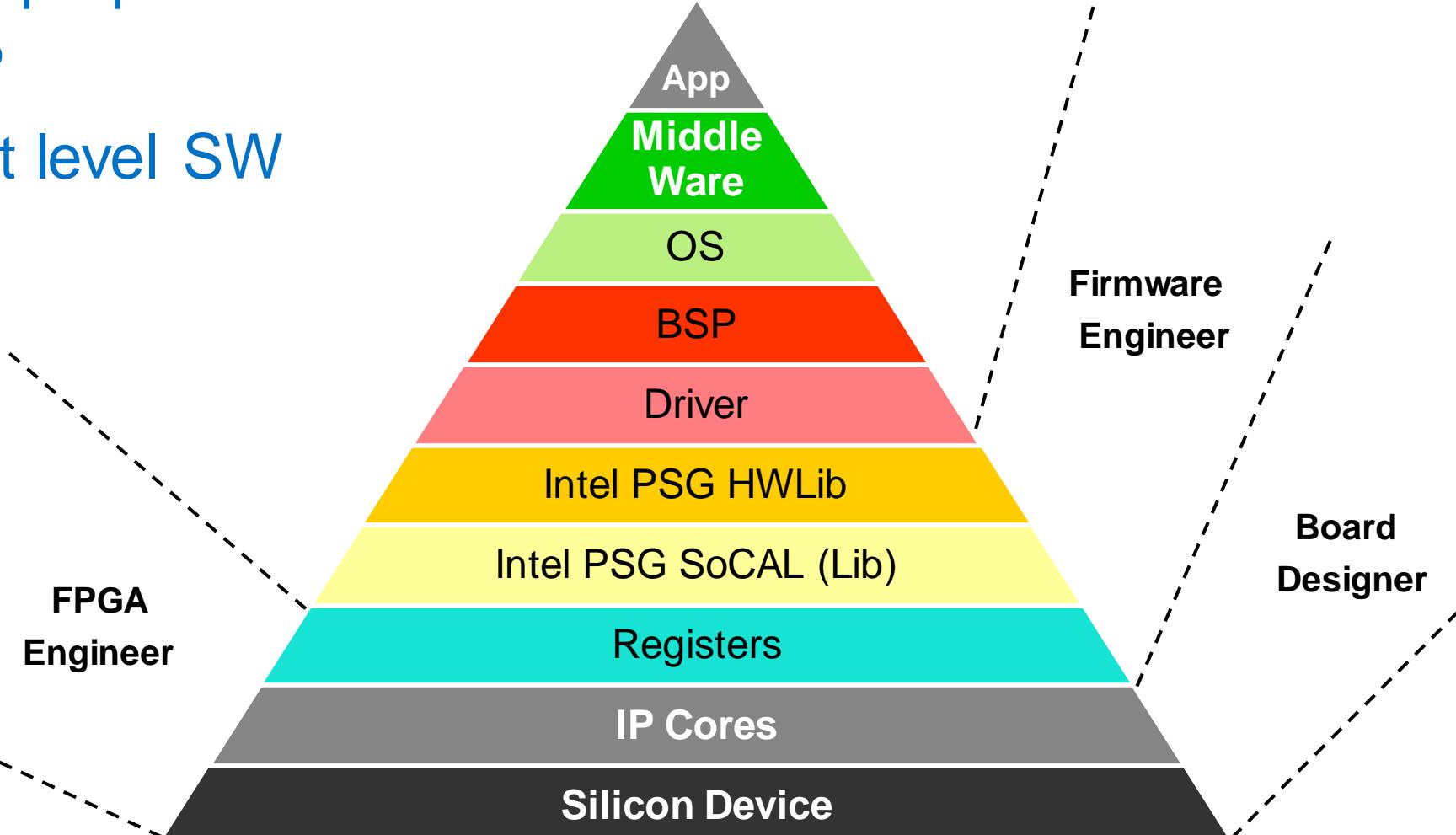
- Flash Programmer

Release

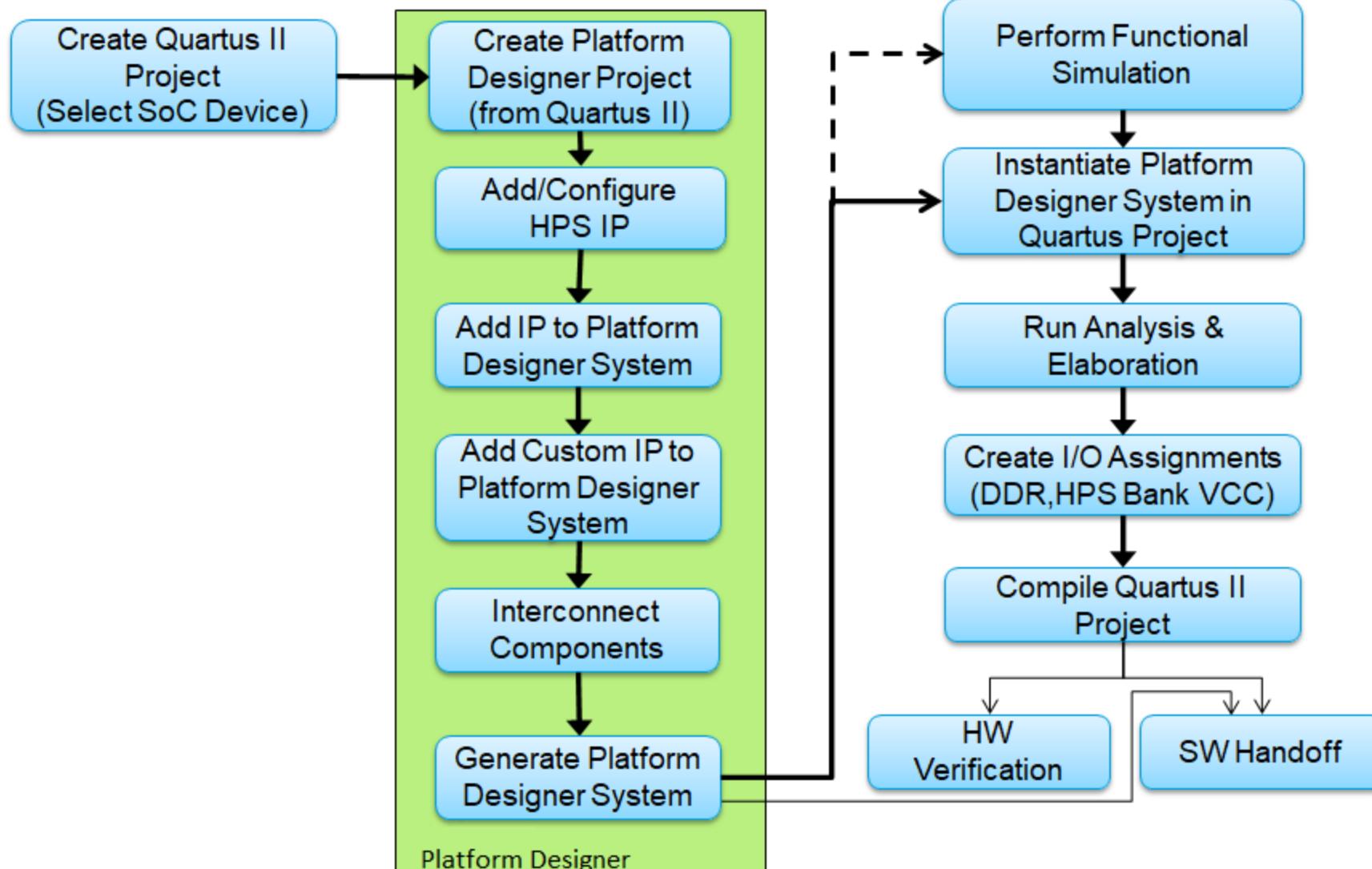
FPGA in the Loop

Hardware Perspective

- Silicon properties
- Soft IP
- Lowest level SW



Typical Hardware Design Flow



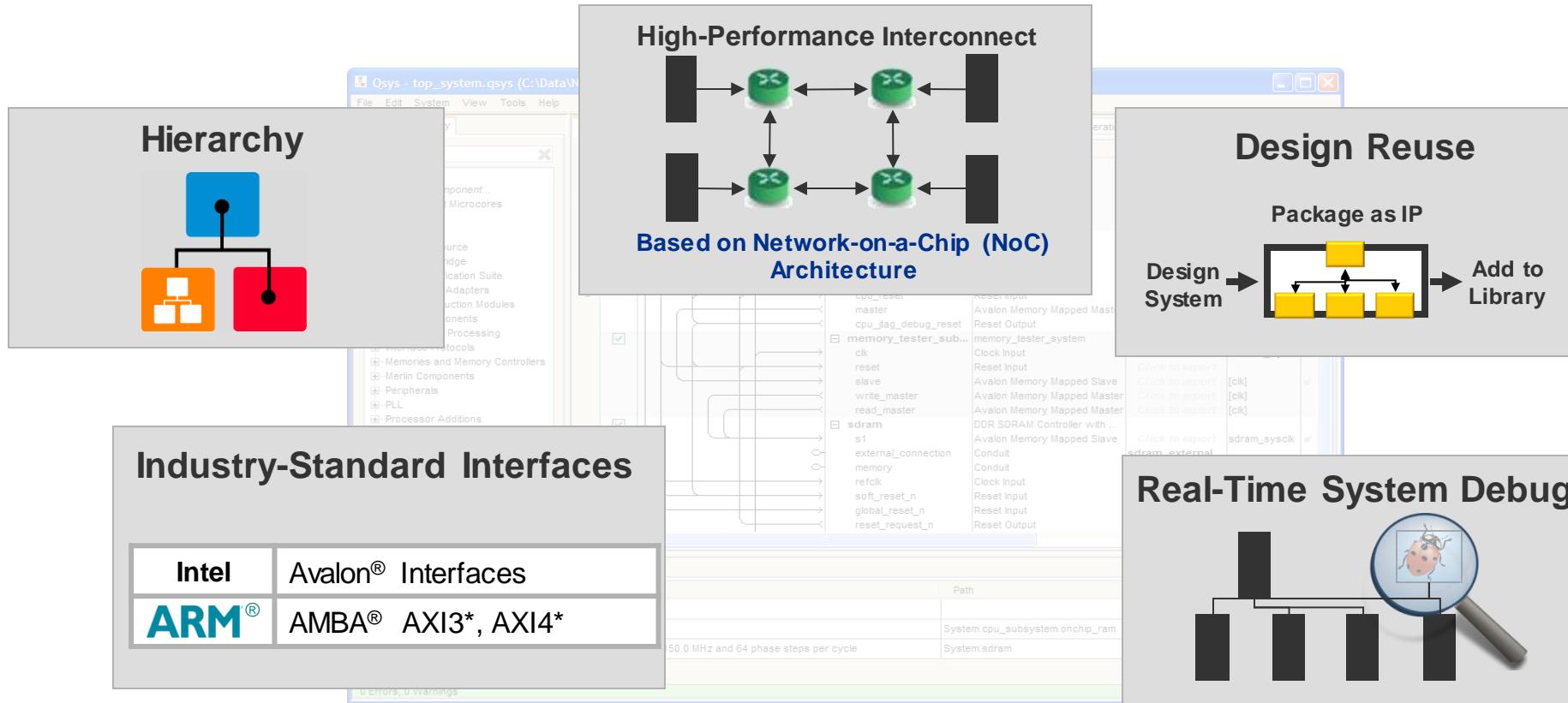
So... what exactly is Platform Designer?

GUI based system integration tool for system design using IP blocks.

- Simplifies complex system development
- Raises the level of design abstraction
- Provides a standard platform:
 - IP integration
 - Custom IP authoring
 - IP verification
- Enables design re-use
- Scales easily to meet the needs of end product
- Reduces time to market



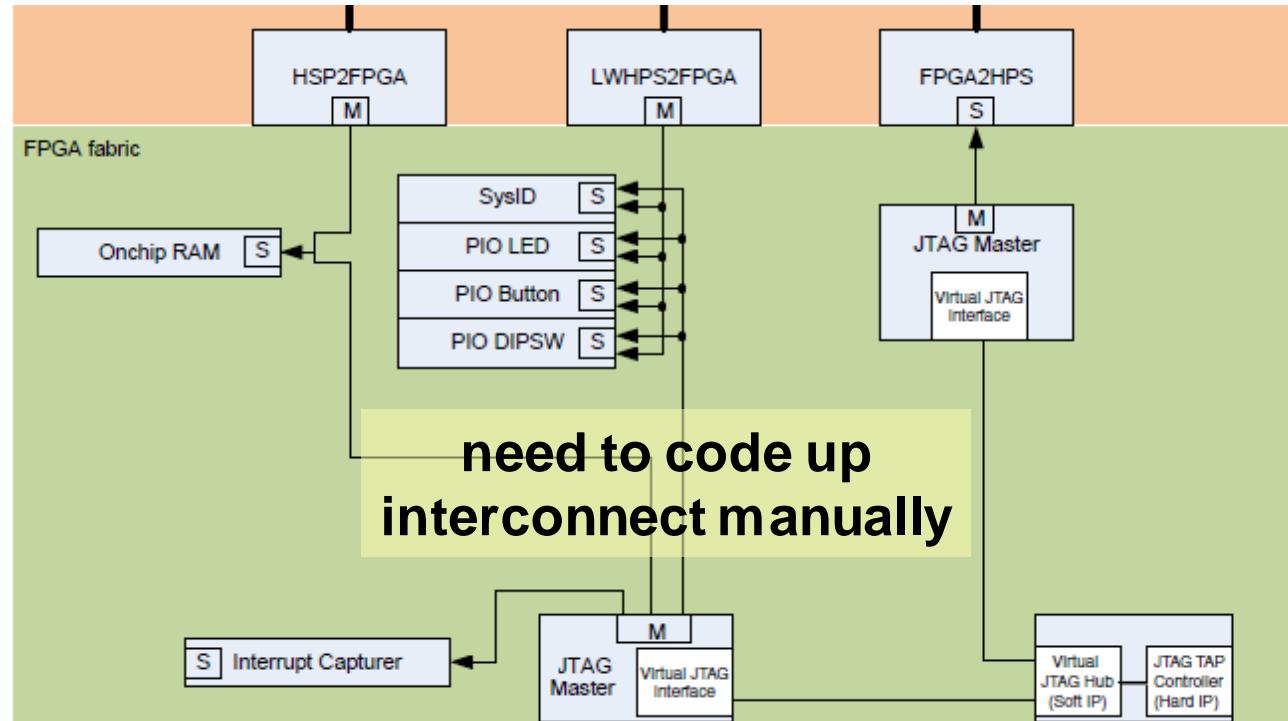
Platform Designer System Integration Platform



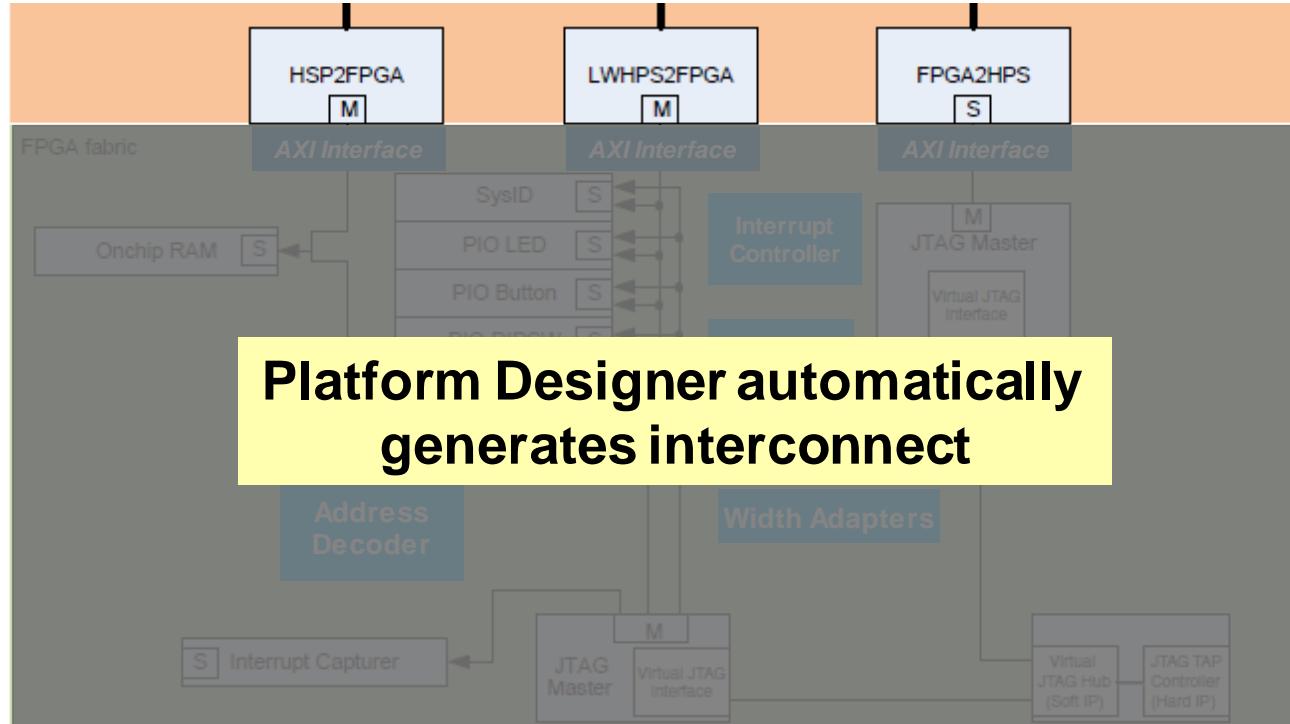
- Platform Designer is Intel PSG's design environment for
 - Deployment of IP
 - Deployment of reference designs and example designs
 - Development platform for Intel PSG custom solutions
 - Design platform for customers to quickly create system designs

Traditional System Design

- Components in system use different interfaces to communicate (some standard, some non-standard)
- Typical system requires significant engineering work to design custom interface logic
- Integrating design blocks and intellectual property (IP) is tedious and error-prone



Automatic Interconnect Generation



- Avoids error-prone integration
- Saves development time with automatic logic & HDL generation
- Enables you to focus on value-add blocks

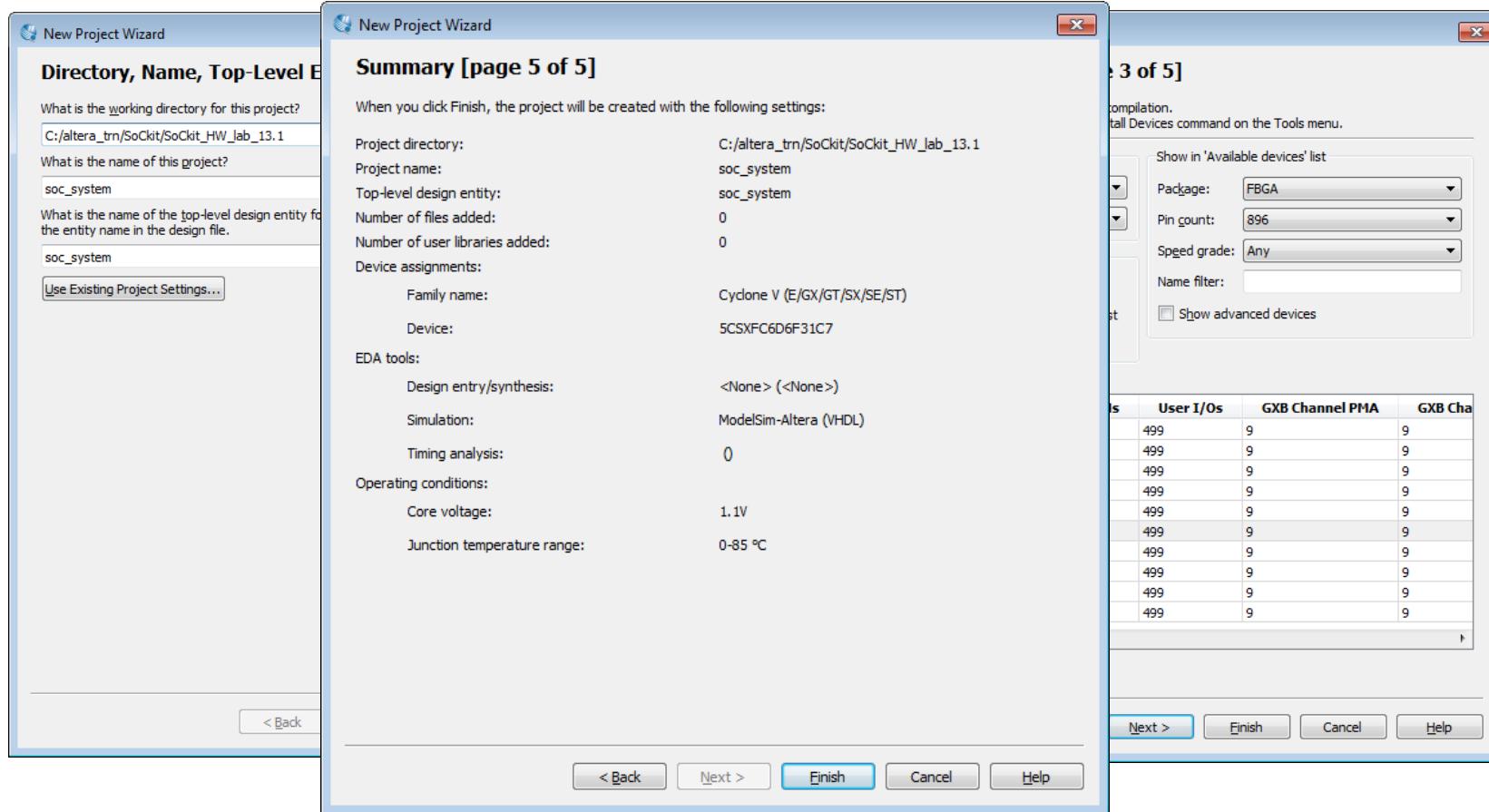
Platform Designer improves productivity by automatically generating the system interconnect logic

System Tool Flow

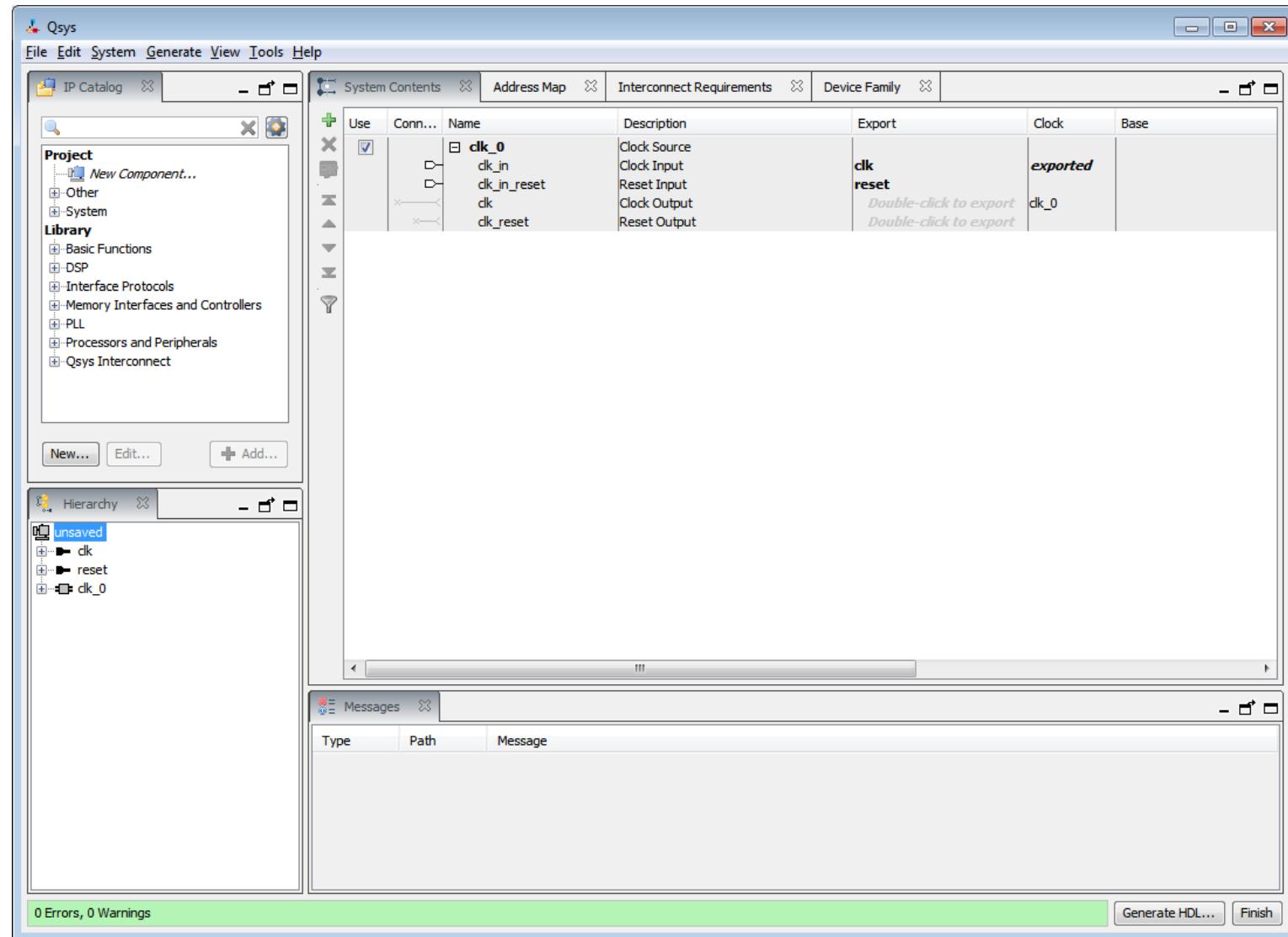
Your User-Customizable System on Chip

Create Quartus II Project for SoC Device

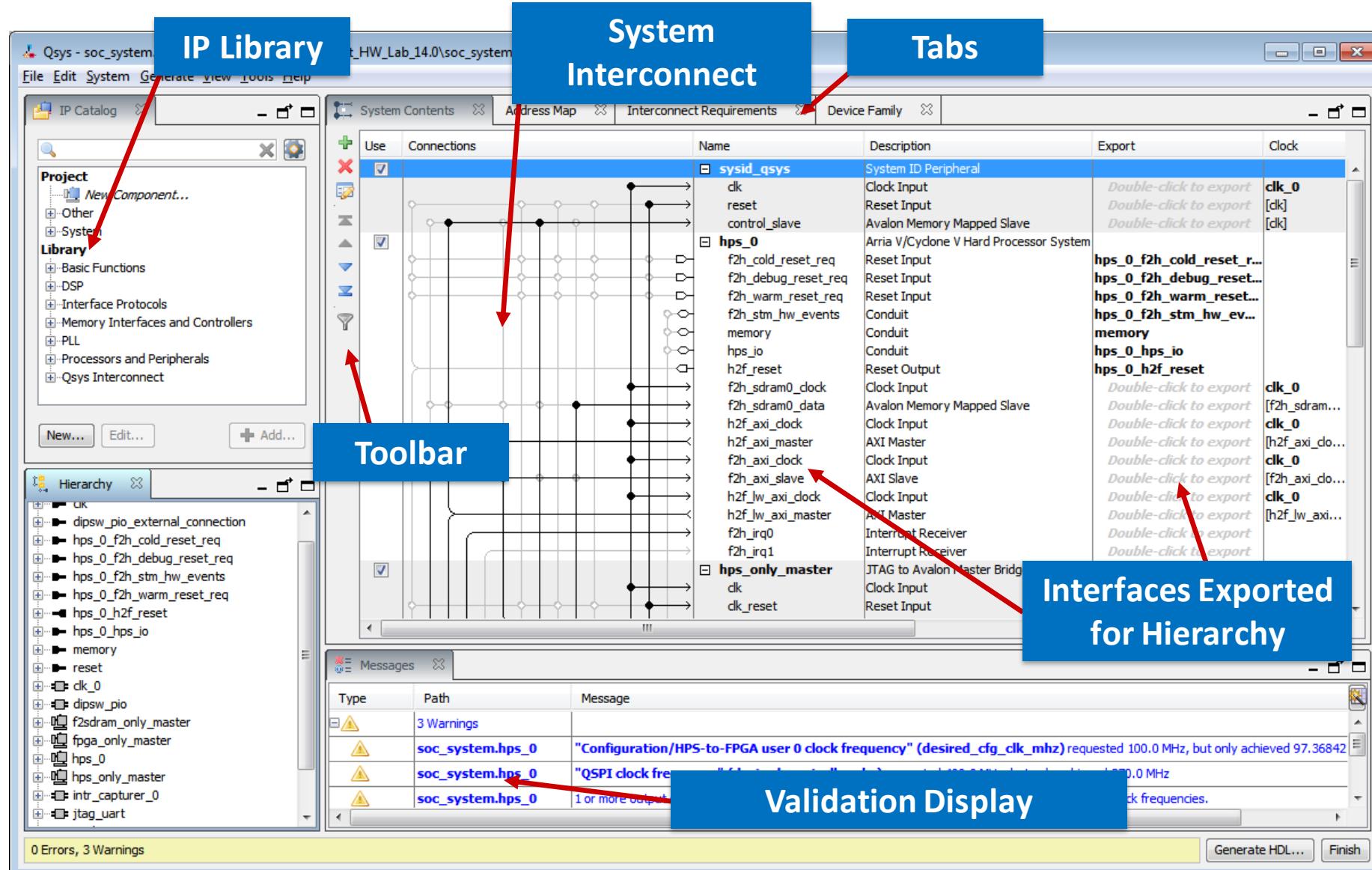
■ Start with a New Quartus II Project



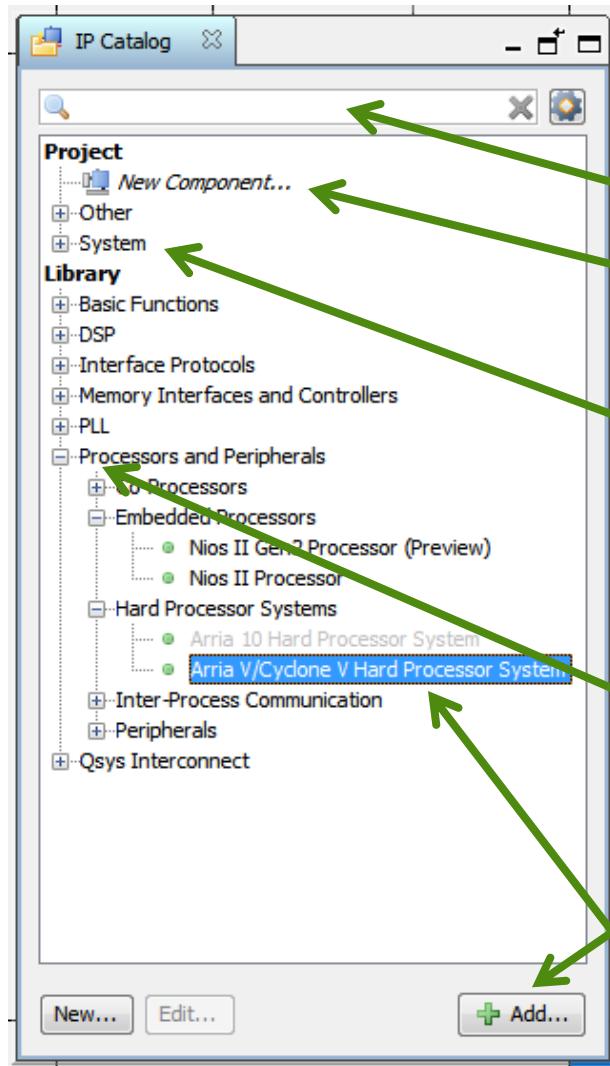
Start a New System in Platform Designer



Platform Designer User Interface



Add IP to Platform Designer System



Lists available IP and systems

Type search string to filter the list

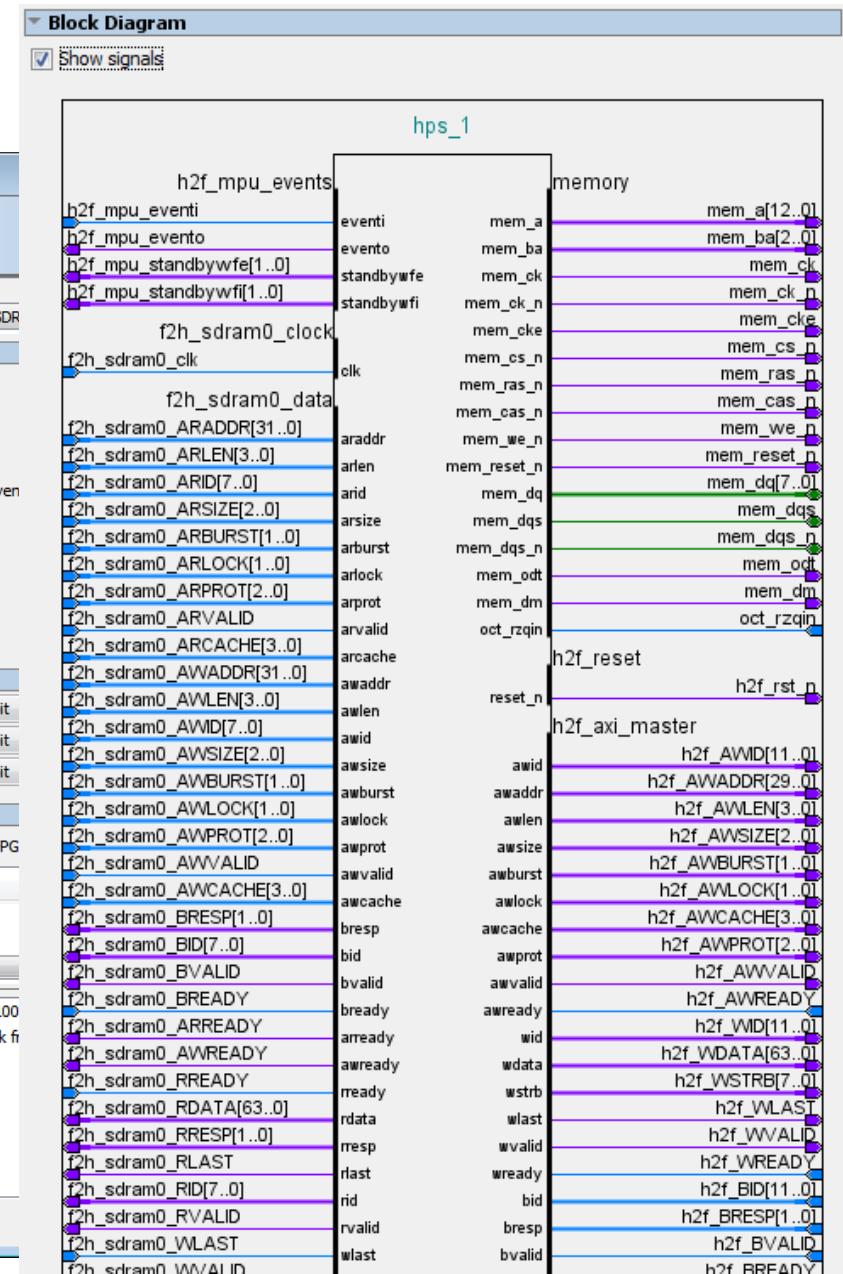
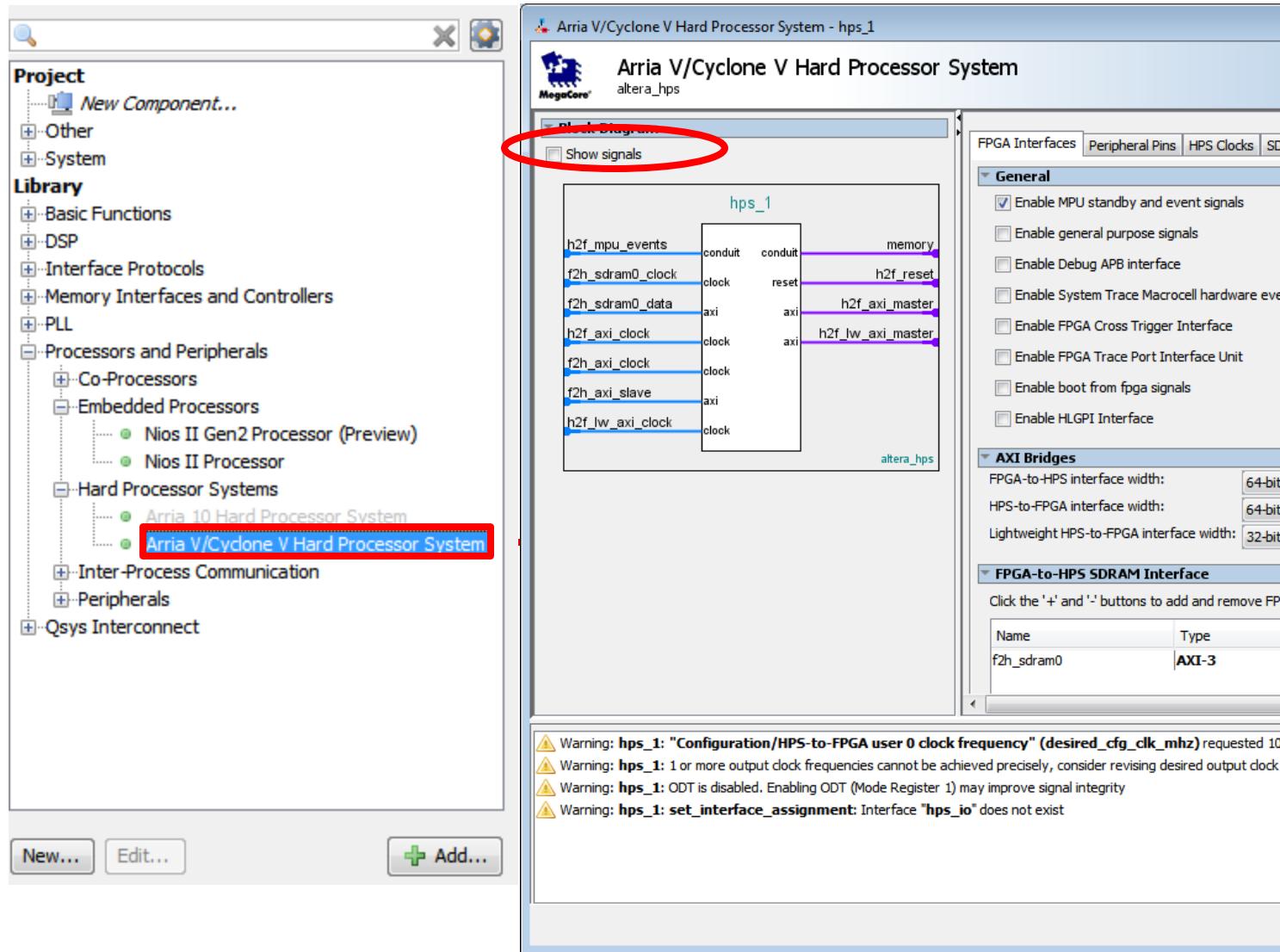
Bring in your custom IP

Reuse systems or custom components

Expand categories to browse components

Double-click component or click Add button
to add selected component to system

Hard Processor System Component

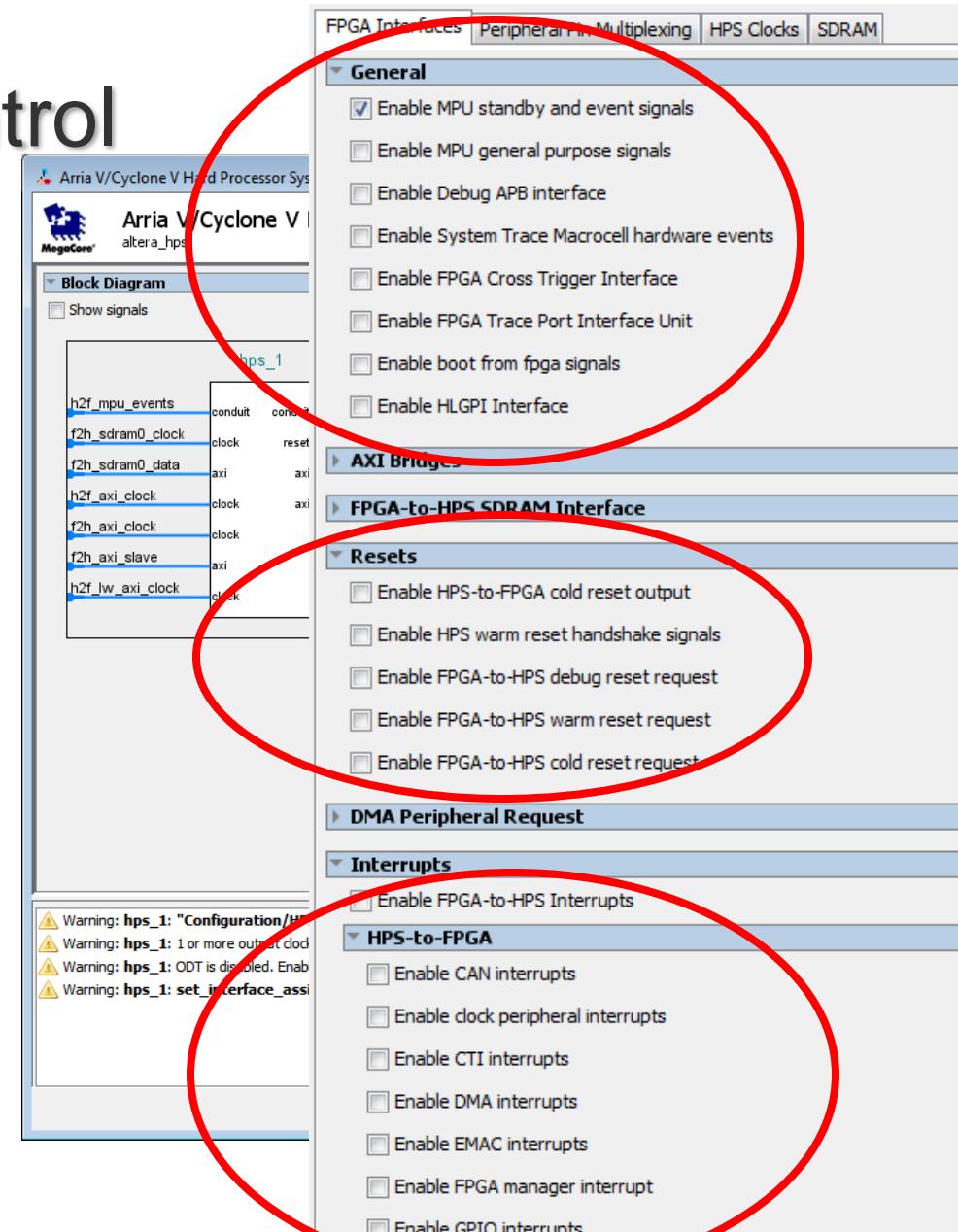


General Options & Boot Control

■ General

- Events
 - Event in and out
 - Wait for event condition
 - Wait for interrupt condition
- GPIO
- Interrupts
- Debug interfaces

■ Boot from FPGA



AXI Bridges

■ FPGA-to-HPS

- Access peripherals & memory
- 4 GB space
- Widths 32, 64, 128

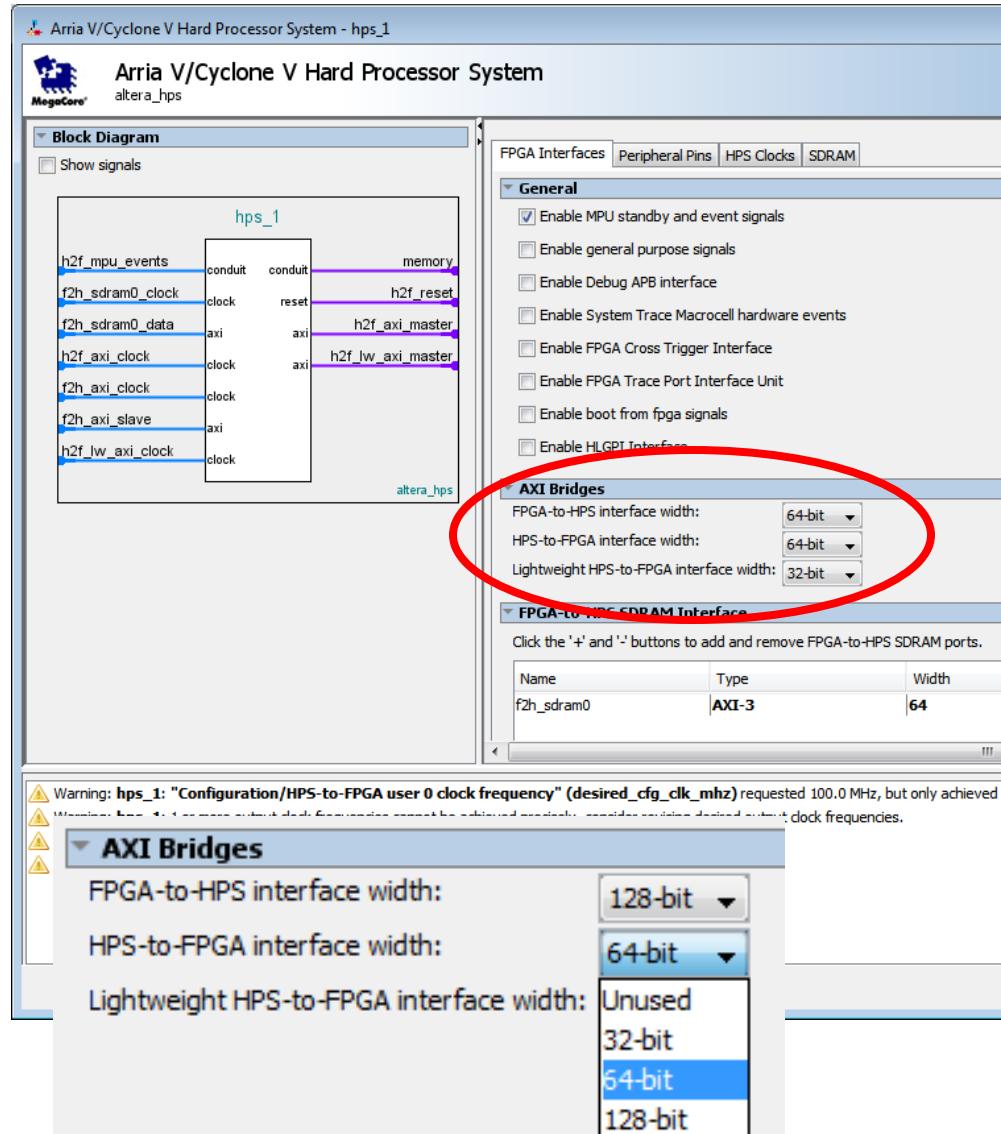
■ HPS-to-FPGA

- 960 MB space
- Widths 32, 64, 128

■ Lightweight HPS-to-FPGA

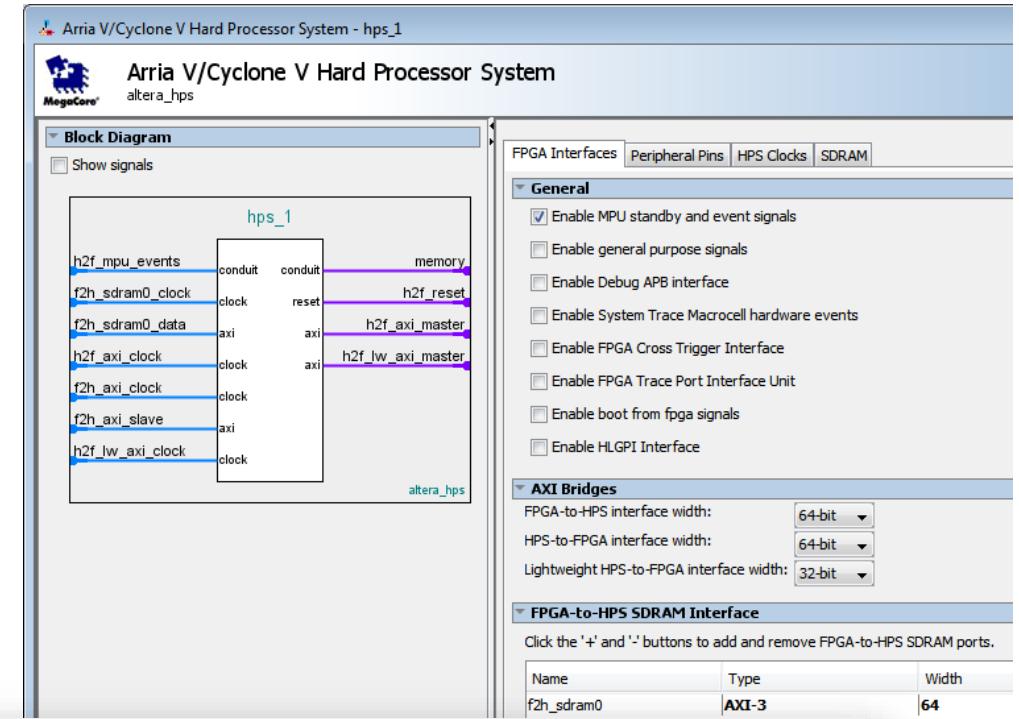
- Lower performance (32 bits)
- Accessing CSRs
- 2 MB space

■ All support soft Avalon connections



FPGA-to-HPS SDRAM Interface

- AXI-3 or Avalon-MM
- Select the number of interfaces
 - Maximum of 3 AXI-3 or
 - Maximum of 6 Avalon-MM
- Data widths:
 - 32, 64, 128, 256



Resets & DMA Control

■ Resets

- Different reset domains
 - Cold
 - Warm
 - Debug
- HPS can drive resets to FPGA
- FPGA can drive resets

■ Resets

- Enable HPS-to-FPGA cold reset output
- Enable HPS warm reset handshake signals
- Enable FPGA-to-HPS debug reset request
- Enable FPGA-to-HPS warm reset request
- Enable FPGA-to-HPS cold reset request

■ DMA Peripheral Request

Peripheral Request ID	Enabled
0	No
1	No
2	No
3	No
4	No
5	No

HPS Peripheral Pin Multiplexing

- Enable peripheral interfaces
- Choose peripheral I/O modes
- Select I/O set

FPGA Interfaces Peripheral Pin Multiplexing HPS Clocks SDRAM

Hover the mouse cursor over the mode parameters for a tooltip regarding signal membership details.

Ethernet Media Access Controller

EMAC0 pin multiplexing: Unused
EMAC0 mode: N/A
EMAC1 pin multiplexing: Unused
EMAC1 mode: N/A

NAND Flash Controller

NAND pin multiplexing: Unused
NAND mode: N/A

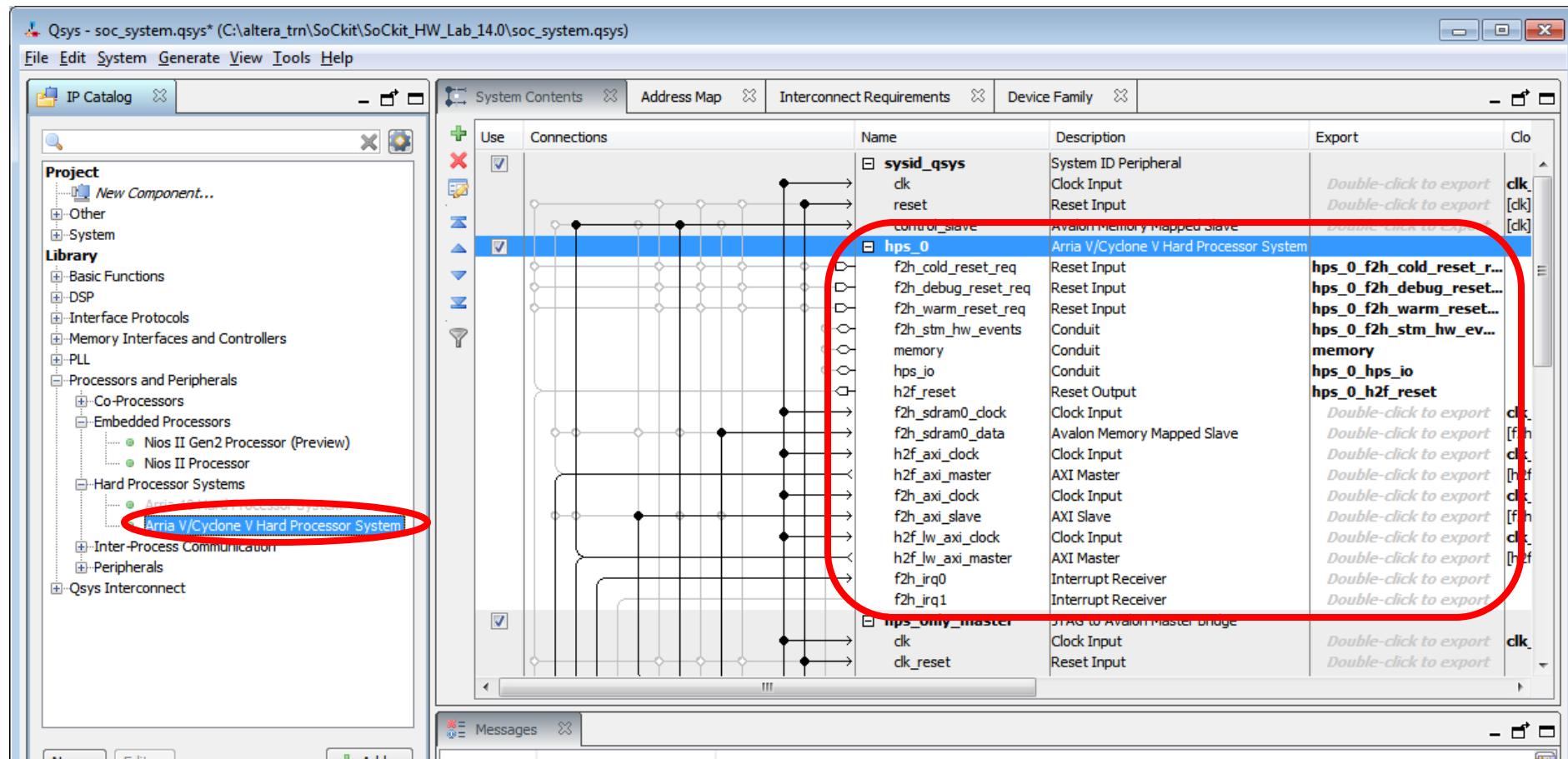
QSPI Flash Controller

QSPI pin multiplexing: Unused
QSPI mode: Unused

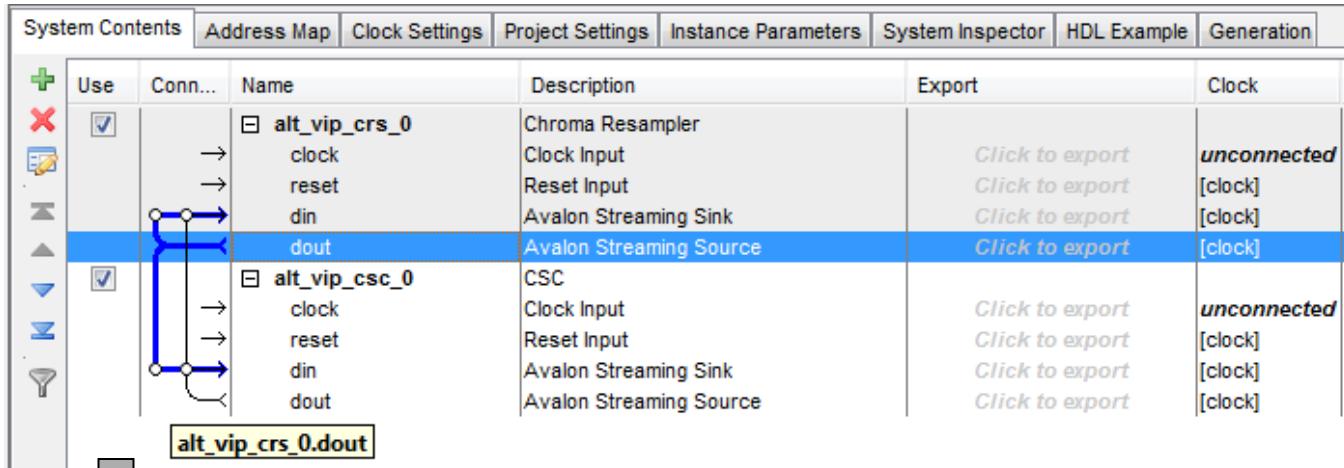
Peripherals Mux Table

PinName	mux_select_1	mux_select_2	mux_select_3
RGMII0_TX_CLK			EMAC0.TX_CLK (Set0)
RGMII0_RXD0		USB1.D0 (Set0)	EMAC0.TXD0 (Set0)
RGMII0_RXD1		USB1.D1 (Set0)	EMAC0.TXD1 (Set0)
RGMII0_RXD2		USB1.D2 (Set0)	EMAC0.TXD2 (Set0)
RGMII0_RXD3		USB1.D3 (Set0)	EMAC0.TXD3 (Set0)
RGMII0_RXD0		USB1.D4 (Set0)	EMAC0.RXD0 (Set0)
RGMII0_MDIO	I2C2.SDA (Set0)	USB1.D5 (Set0)	EMAC0.MDIO (Set0)
RGMII0_MDC	I2C2.SCL (Set0)	USB1.D6 (Set0)	EMAC0.MDC (Set0)
RGMII0_RX_CTL		USB1.D7 (Set0)	EMAC0.RX_CTL (Set0)
RGMII0_TX_CTL			EMAC0.TX_CTL (Set0)
RGMII0_RX_CLK		USB1.CLK (Set0)	EMAC0.RX_CLK (Set0)
RGMII0_RXD1		USB1.STP (Set0)	EMAC0.RXD1 (Set0)
RGMII0_RXD2		USB1.DIR (Set0)	EMAC0.RXD2 (Set0)
RGMII0_RXD3		USB1.NXT (Set0)	EMAC0.RXD3 (Set0)
NAND_ALE	QSPI.SS3 (Set1) (Set0)	EMAC1.TX_CLK (Set0)	NAND.ALE (Set0)
NAND_CE	USB1.D0 (Set1)	EMAC1.TXD0 (Set0)	NAND.CE (Set0)
NAND_CLE	USB1.D1 (Set1)	EMAC1.TXD1 (Set0)	NAND.CLE (Set0)
NAND_RE	USB1.D2 (Set1)	EMAC1.TXD2 (Set0)	NAND.RE (Set0)
NAND_RB	USB1.D3 (Set1)	EMAC1.TXD3 (Set0)	NAND.RB (Set0)
NAND_DQ0		EMAC1.RXD0 (Set0)	NAND.DQ0 (Set0)
NAND_DQ1	I2C3.SDA (Set0)	EMAC1.MDIO (Set0)	NAND.DQ1 (Set0)
NAND_DQ2	I2C3.SCL (Set0)	EMAC1.MDC (Set0)	NAND.DQ2 (Set0)
NAND_DQ3	USB1.D4 (Set1)	EMAC1.RX_CTL (Set0)	NAND.DQ3 (Set0)
NAND_DQ4	USB1.D5 (Set1)	EMAC1.TX_CTL (Set0)	NAND.DQ4 (Set0)
NAND_DQ5	USB1.D6 (Set1)	EMAC1.RX_CLK (Set0)	NAND.DQ5 (Set0)

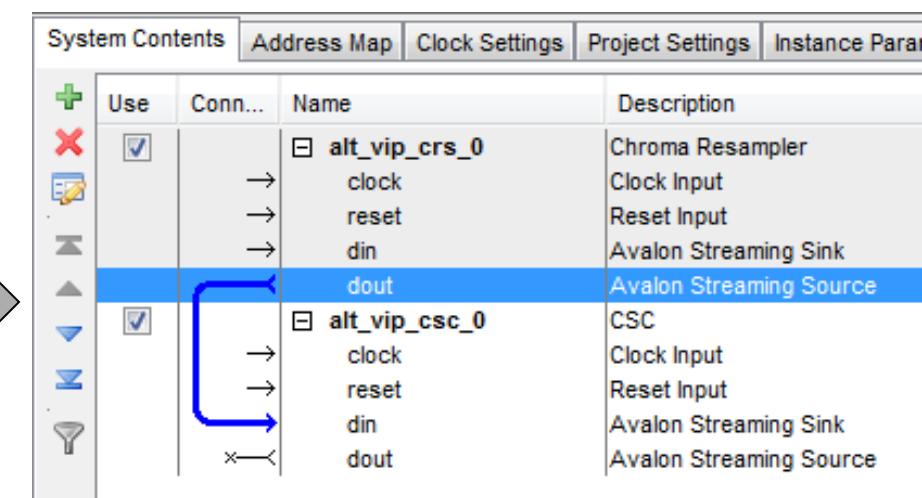
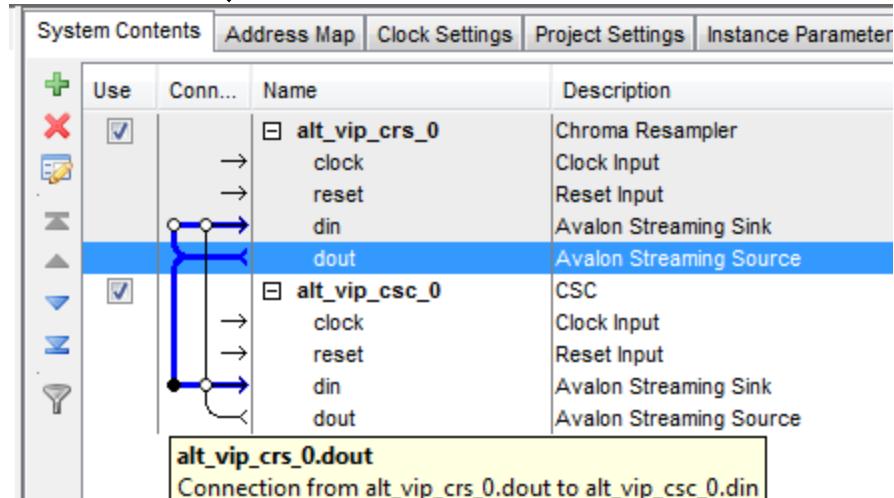
HPS in Platform Designer



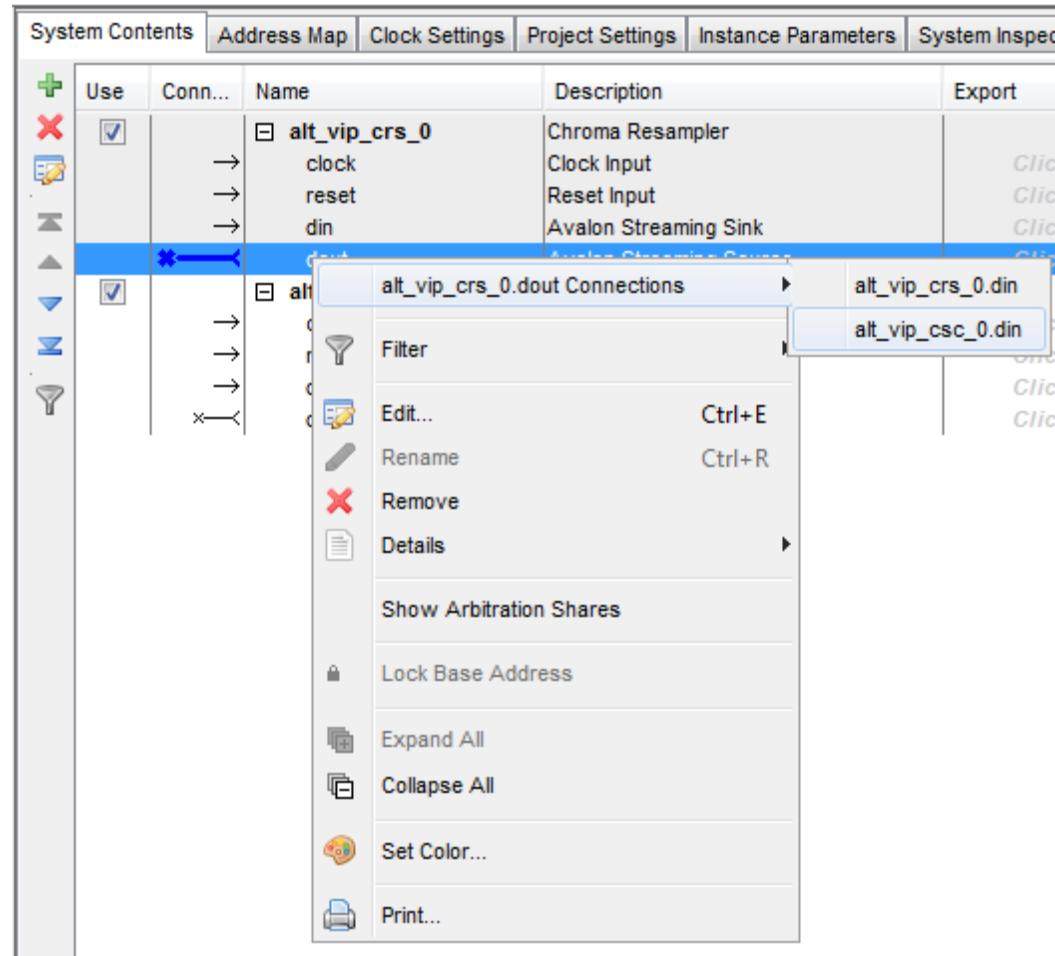
Connect the Components



Click the open dot to make connection

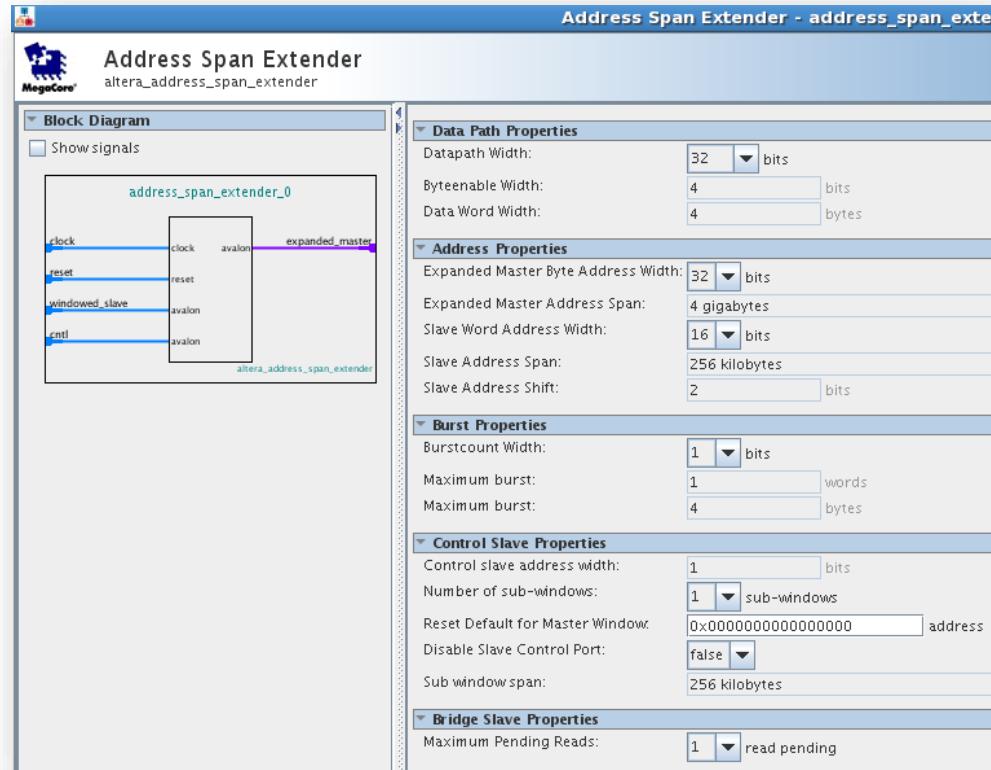
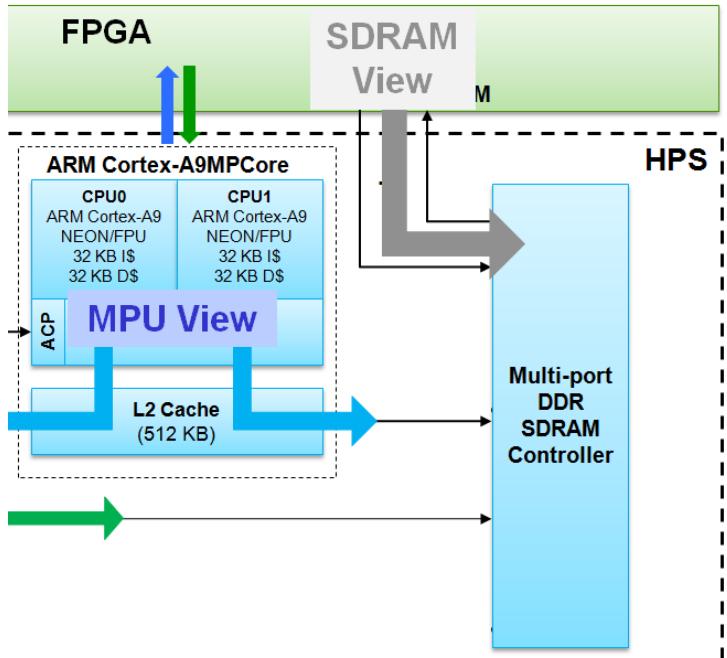


Connect the Components

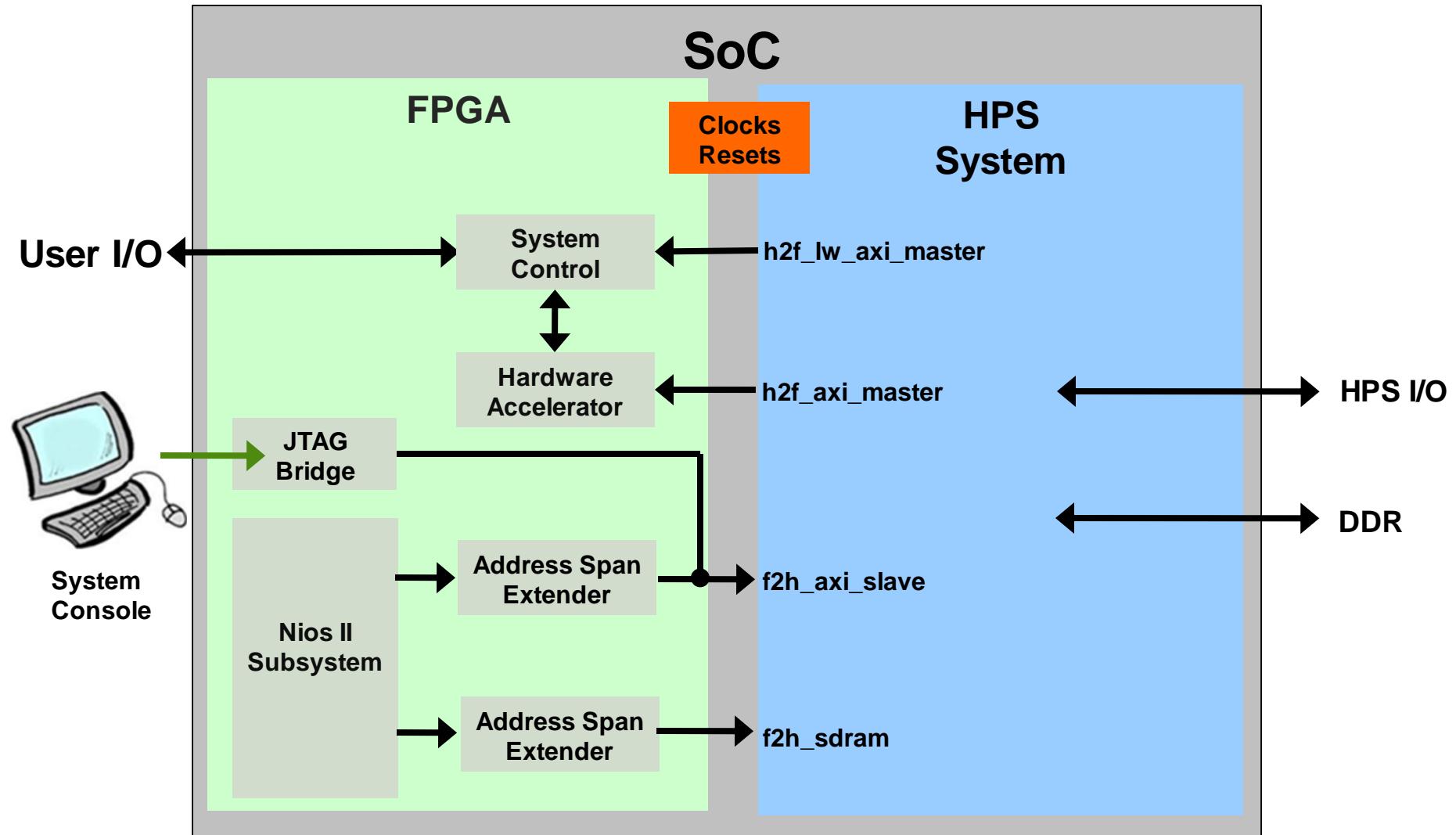


Accessing HPS Memory from FPGA

- 4 GB HPS memory map
- Less capable FPGA masters (e.g. Nios II has 31-bits of address / 2GB)
- Address Span Extender (Windowed Bridge)

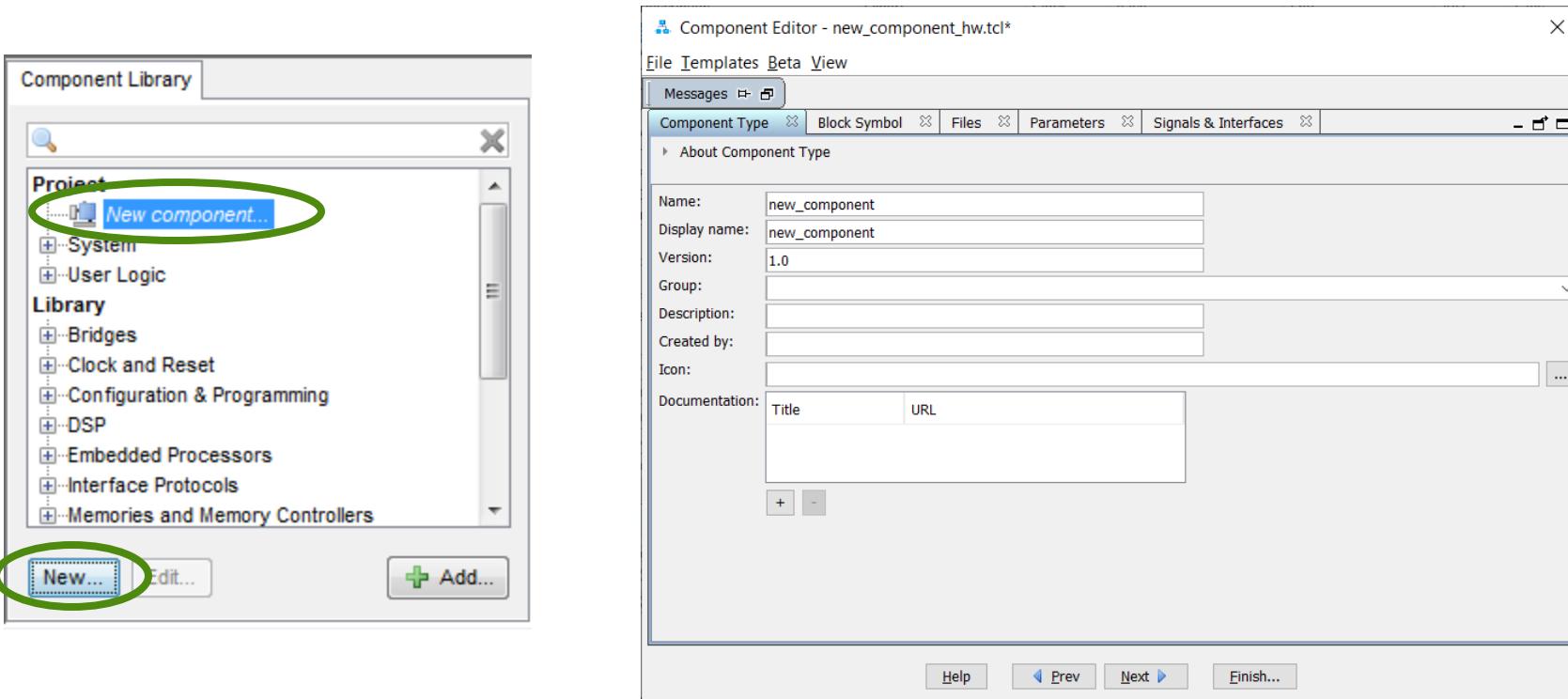


Example: Nios II master in FPGA → HPS



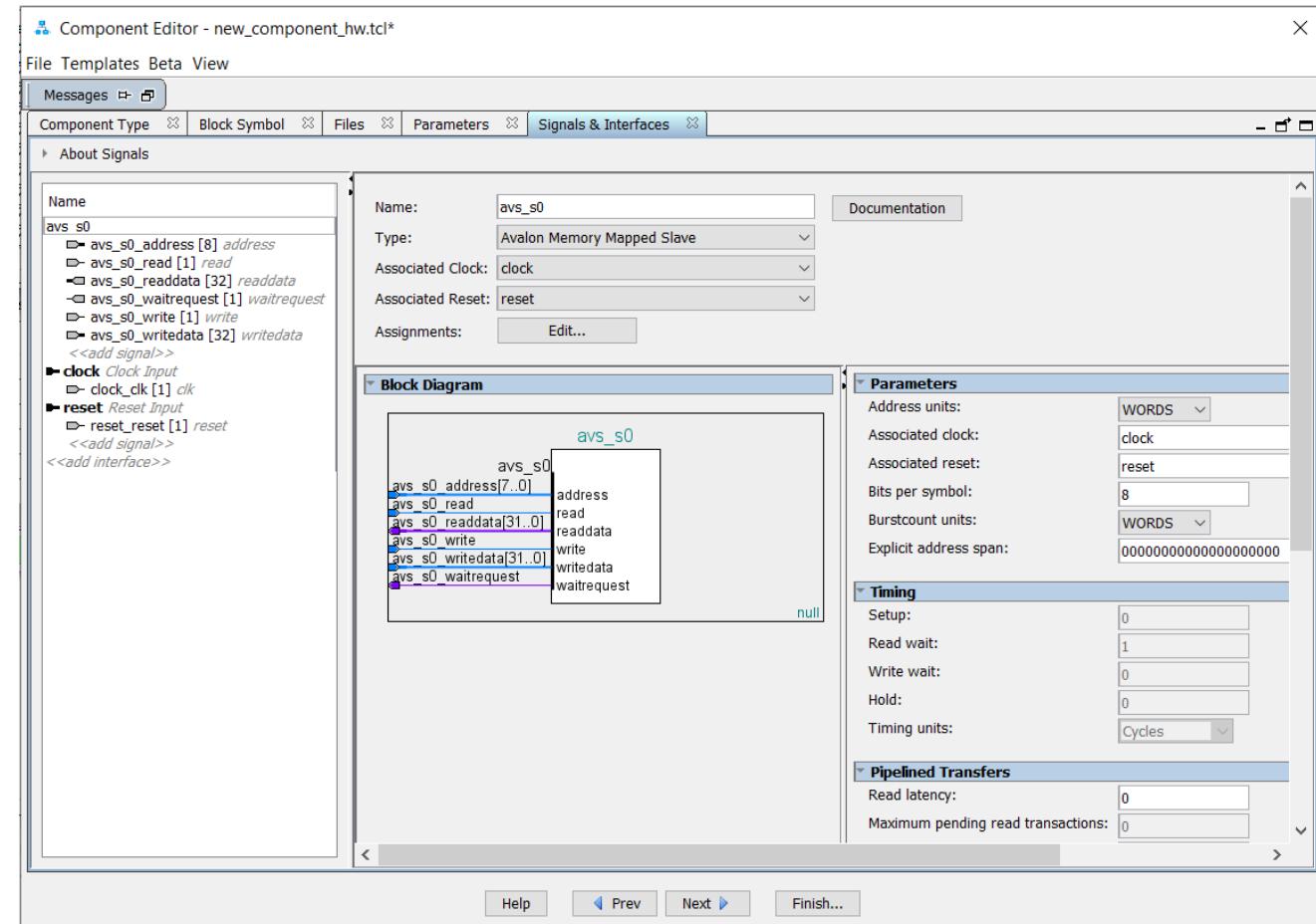
Platform Designer Component Editor

- Import FPGA HDL custom logic as components into a Platform Designer system
 - Launch from Platform Designer pick list or File menu



Create Custom Components

- Design your own logic with Platform Designer supported standard interfaces
- Map component signals to Platform Designer Interface types and Signal Names on the **Signals** tab



Platform Designer-Supported Standard Interfaces

- **Avalon-MM (memory mapped)**

- Little Endian
 - Control plane
 - Master interface makes read and write requests to slave interface

- **Avalon-ST (streaming)**

- Big Endian
 - Data plane
 - Source interface sends data to sink interface (point-to-point)

- **ARM AMBA™ AXI™ 3.0 & 4.0**

- HPS interfaces are AXI 3.0
 - AXI 4.0 is backwards compatible with AXI 3.0

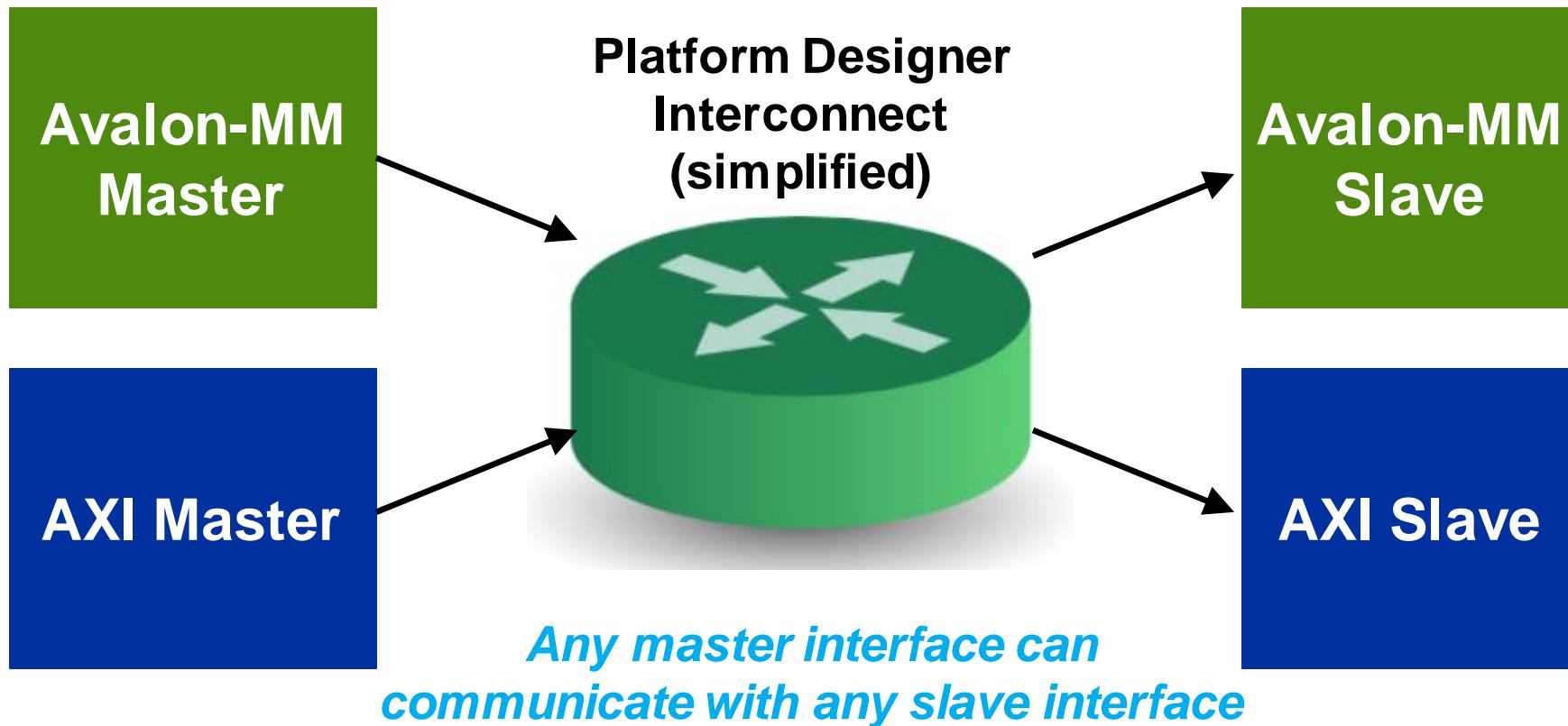


Advantages of Using Standard Interfaces

- Ensure compatibility between IP blocks from different design teams or vendors
 - Any component supporting interface can be connected
- Simplify design entry and team-based design
 - Signal behavior defined by interface
 - Improved understanding, simplified documentation
 - No manual wiring or mapping of control, data, and status signals
 - Fast system-level integration
 - Easy system changes
- Simplify interface verification
 - Use verification infrastructure to verify against standard
 - Bus functional models, interface compliance assertions and monitors, functional coverage

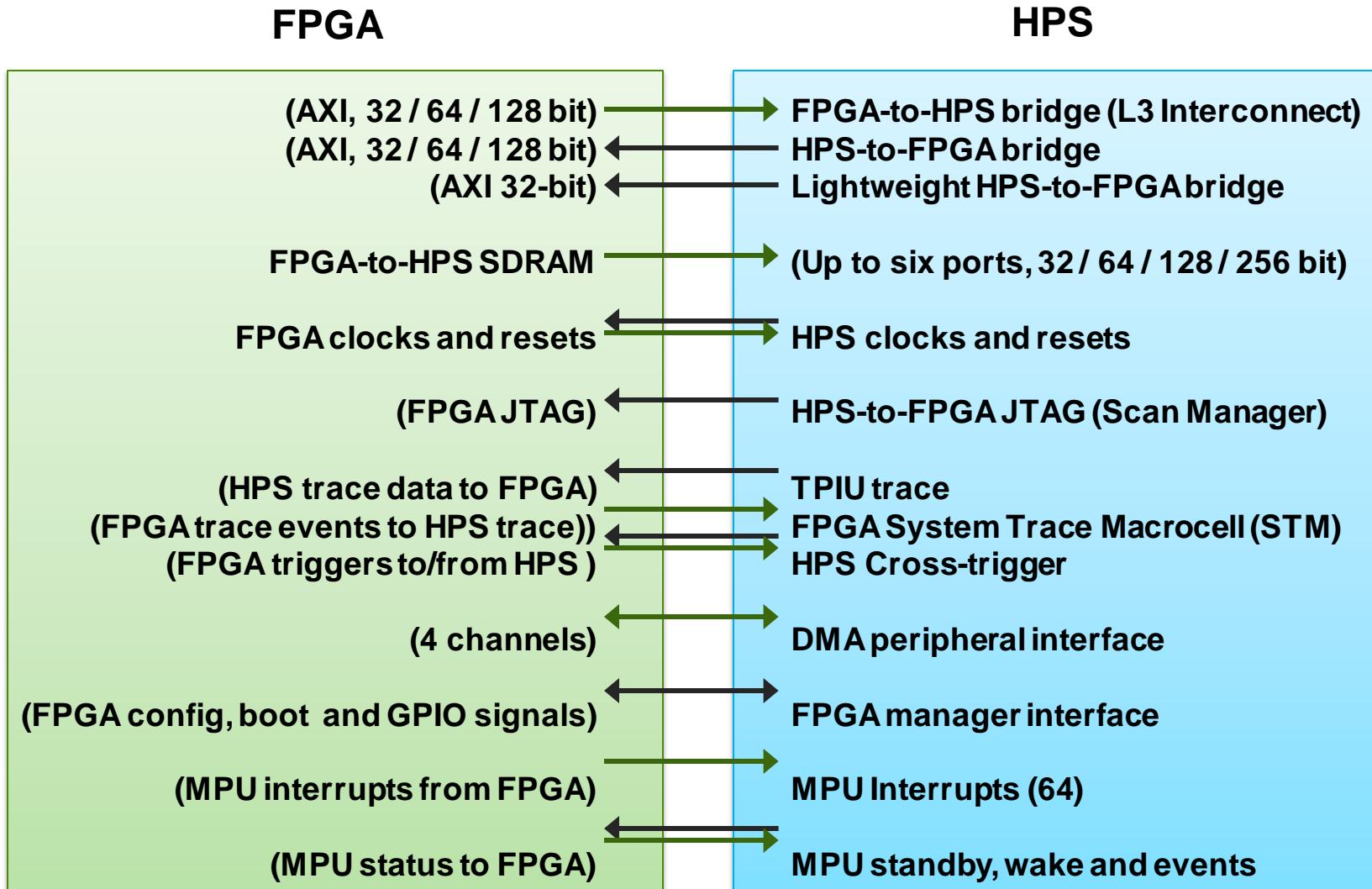


Standard Interface Example



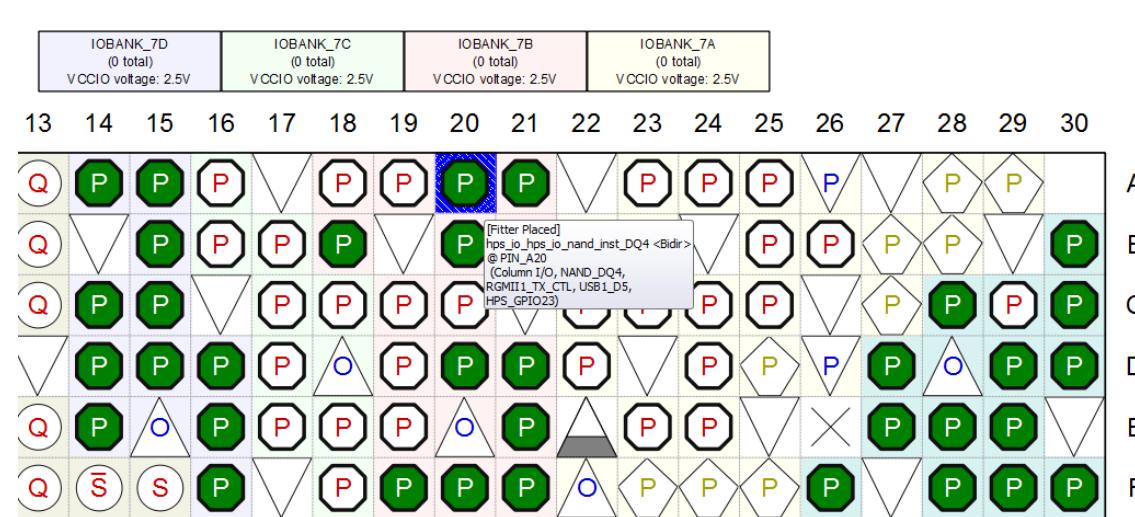
- If Platform Designer system is all AXI components, then Platform Designer will generate interconnect that is natively AXI.
- Otherwise, interconnect is based on Avalon.

Summary: FPGA – HPS Connections

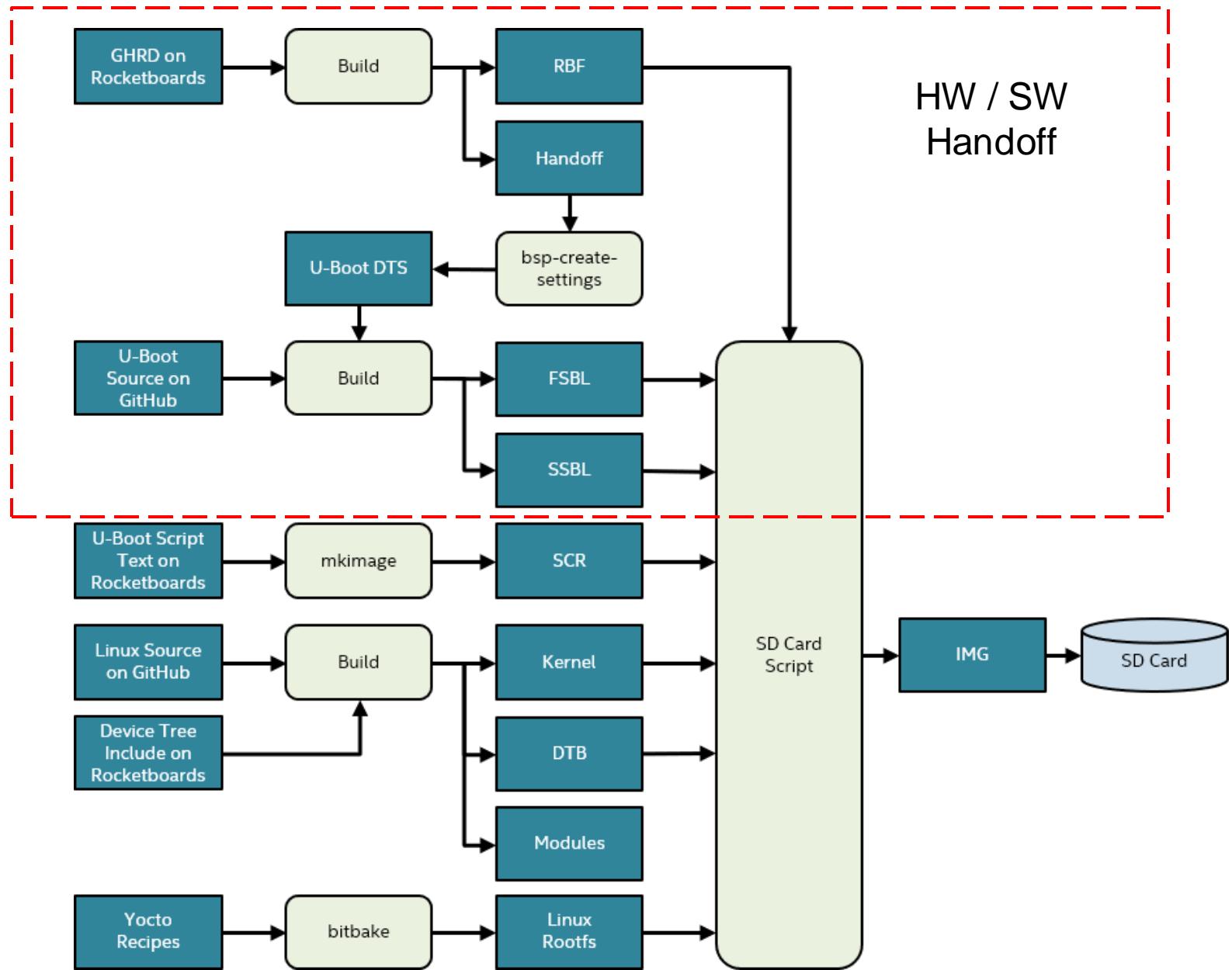


HPS Pin Assignments

- Platform Designer Automatically assigns HPS pins (pin mux settings transferred into Quartus II project)
- Check Assignments in Quartus II pin planner:
 - Drive strength
 - VCCIO for banks



Hardware/Software Design Flow Overview

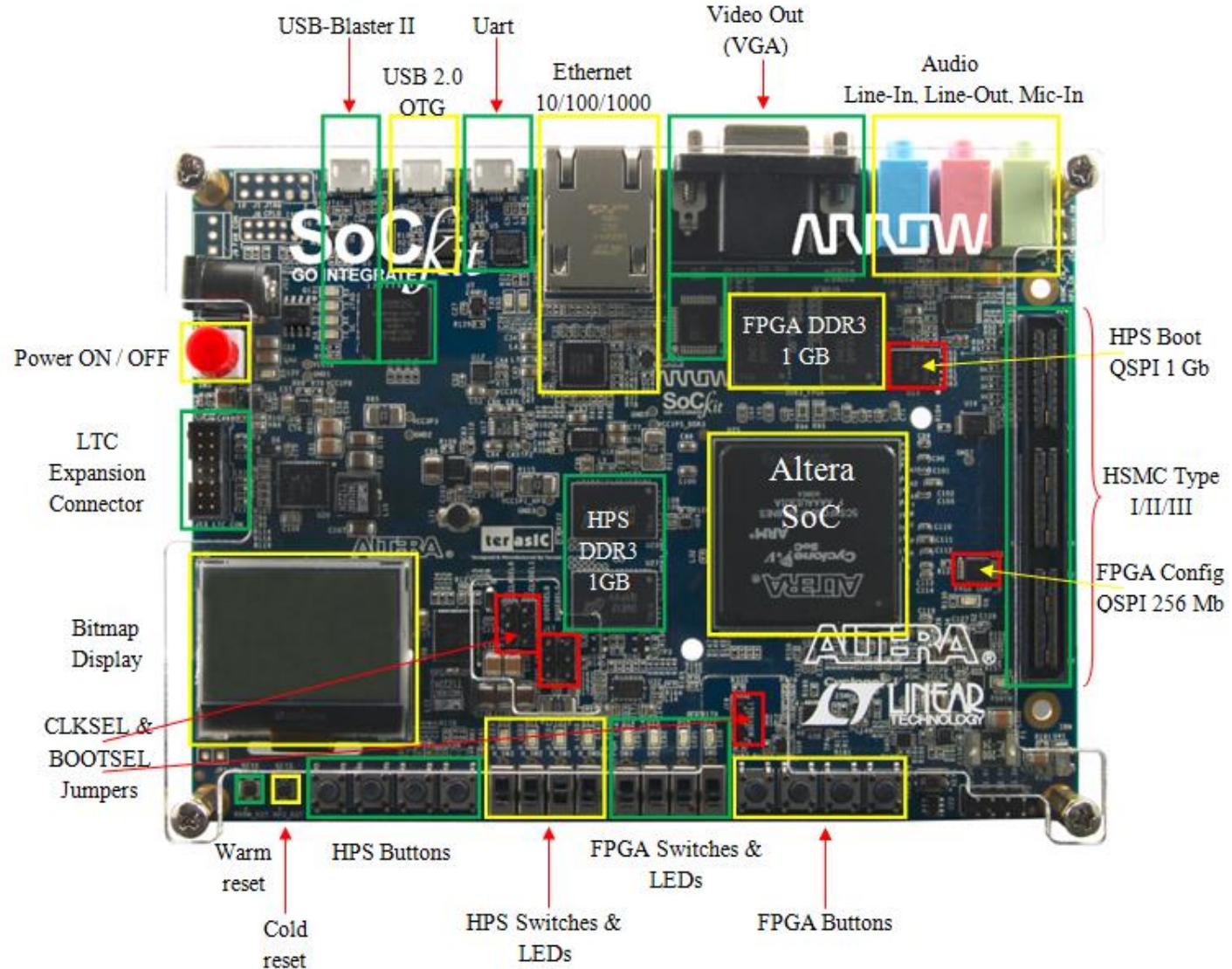




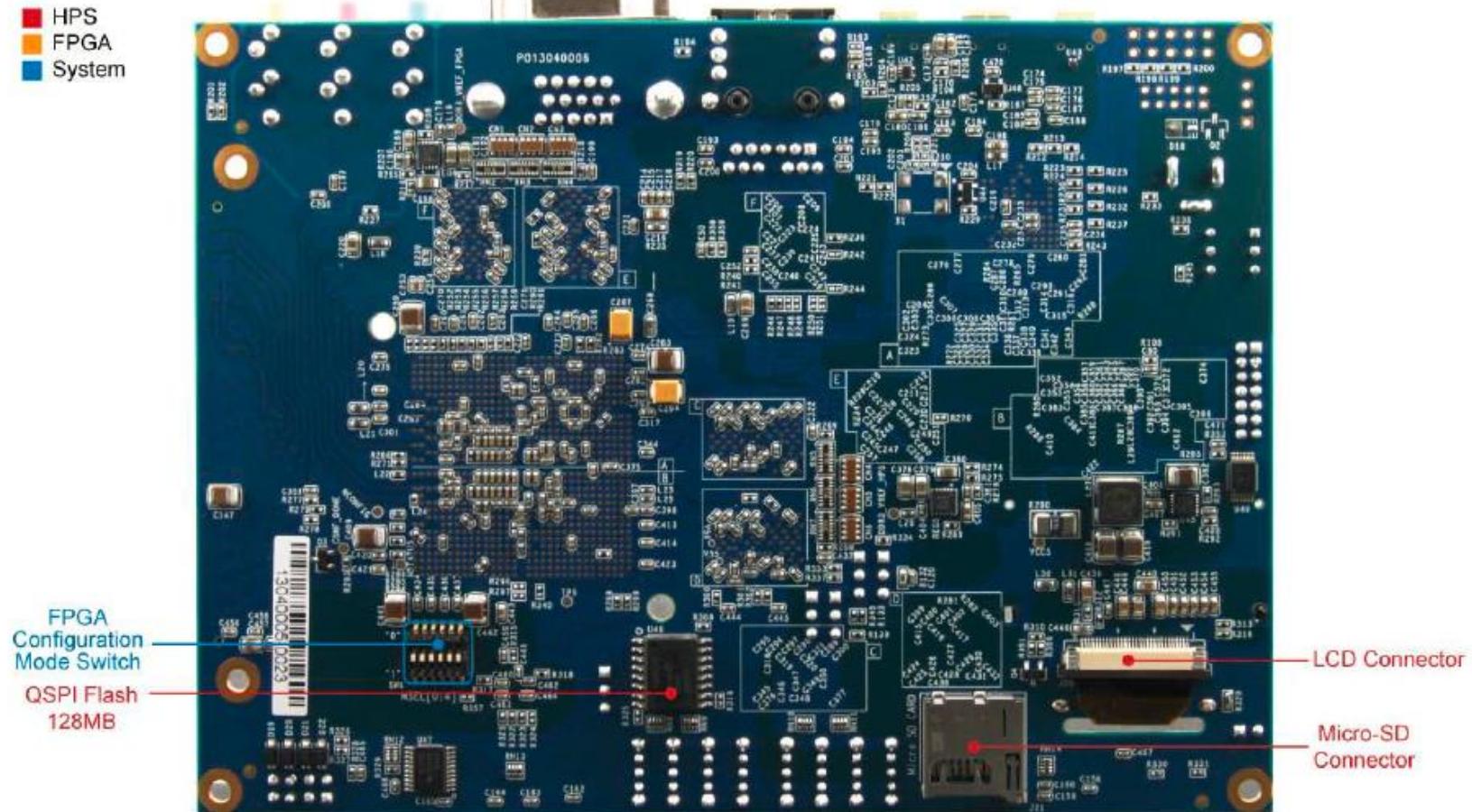
SoCKit – Details of the Kit

Your User-Customizable System on Chip

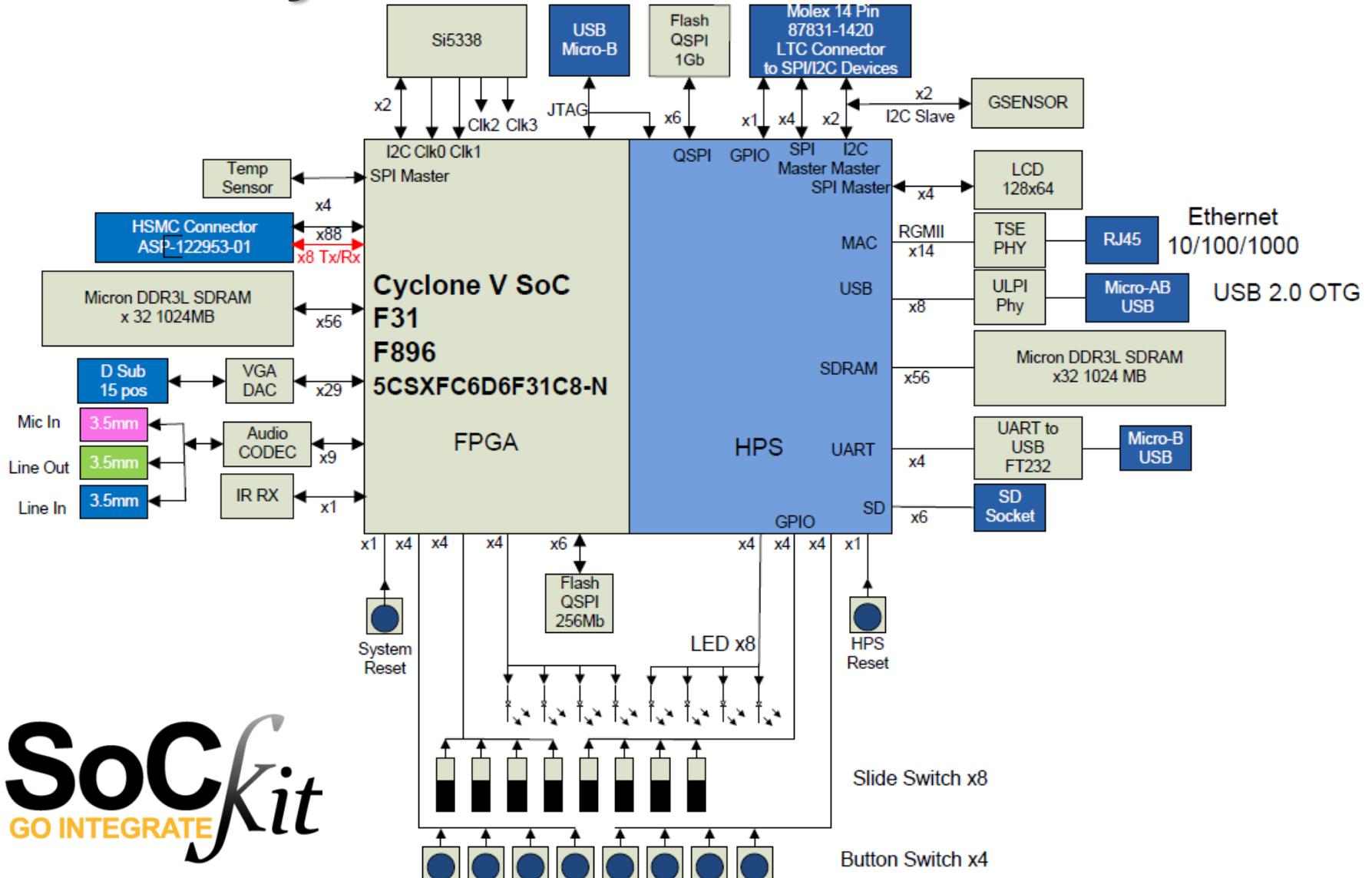
Arrow Cyclone V SoCKit



Arrow Cyclone V SoCKit



Arrow Cyclone V SoCKit



SoC
kit
GO INTEGRATE

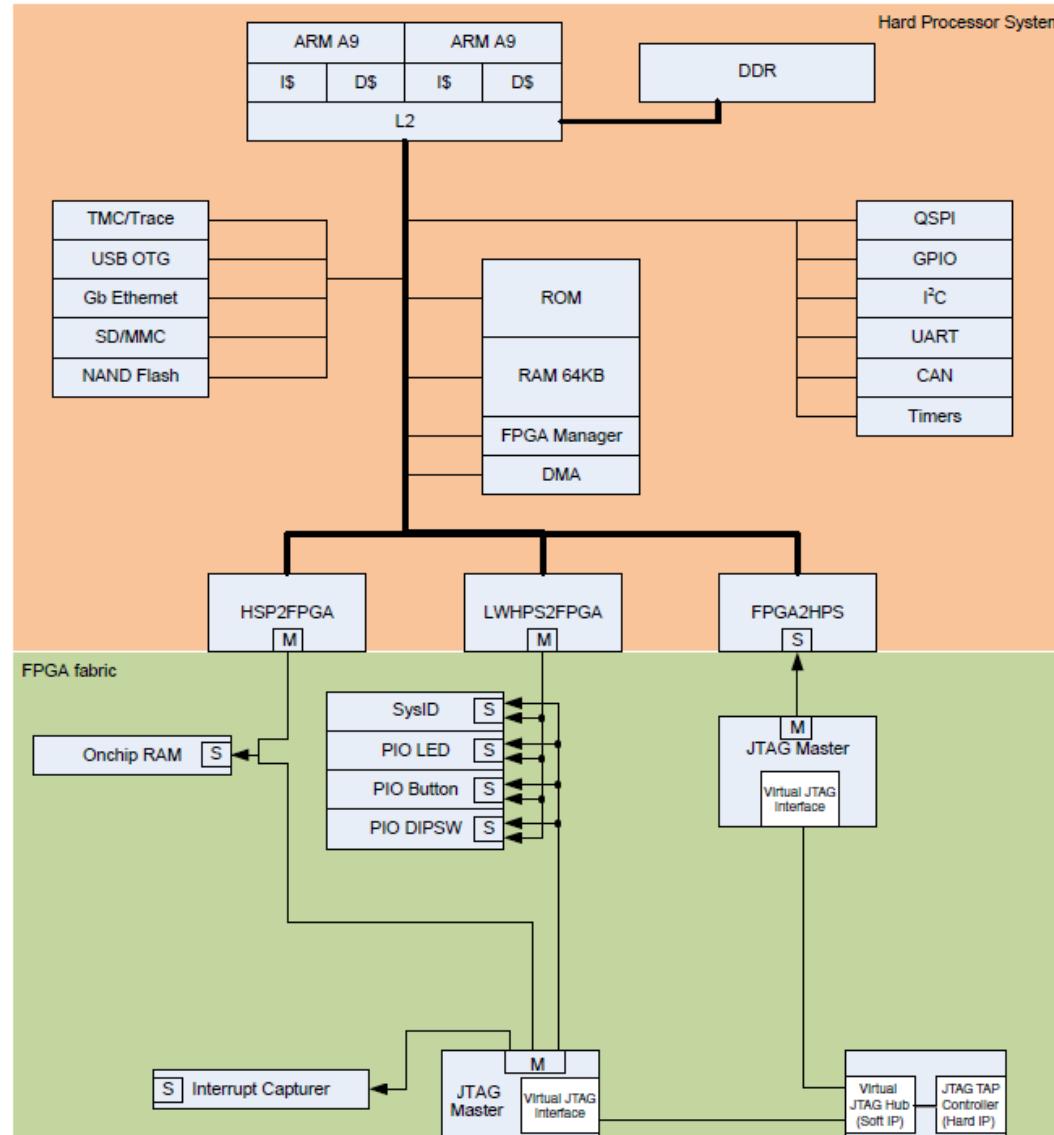
Golden Reference Design

■ HPS configuration

- DRAM, QSPI, SD/Card
- All peripheral functions exposed at least once

■ FPGA configuration

- Simple Platform Designer “sandbox” system
 - First experience in ARM / FPGA integration
- Getting started guide to walk them through the process on integrating IP
 - HW simulation
 - Verification via system console
 - Verification via CPU
 - HW/SW hand-off





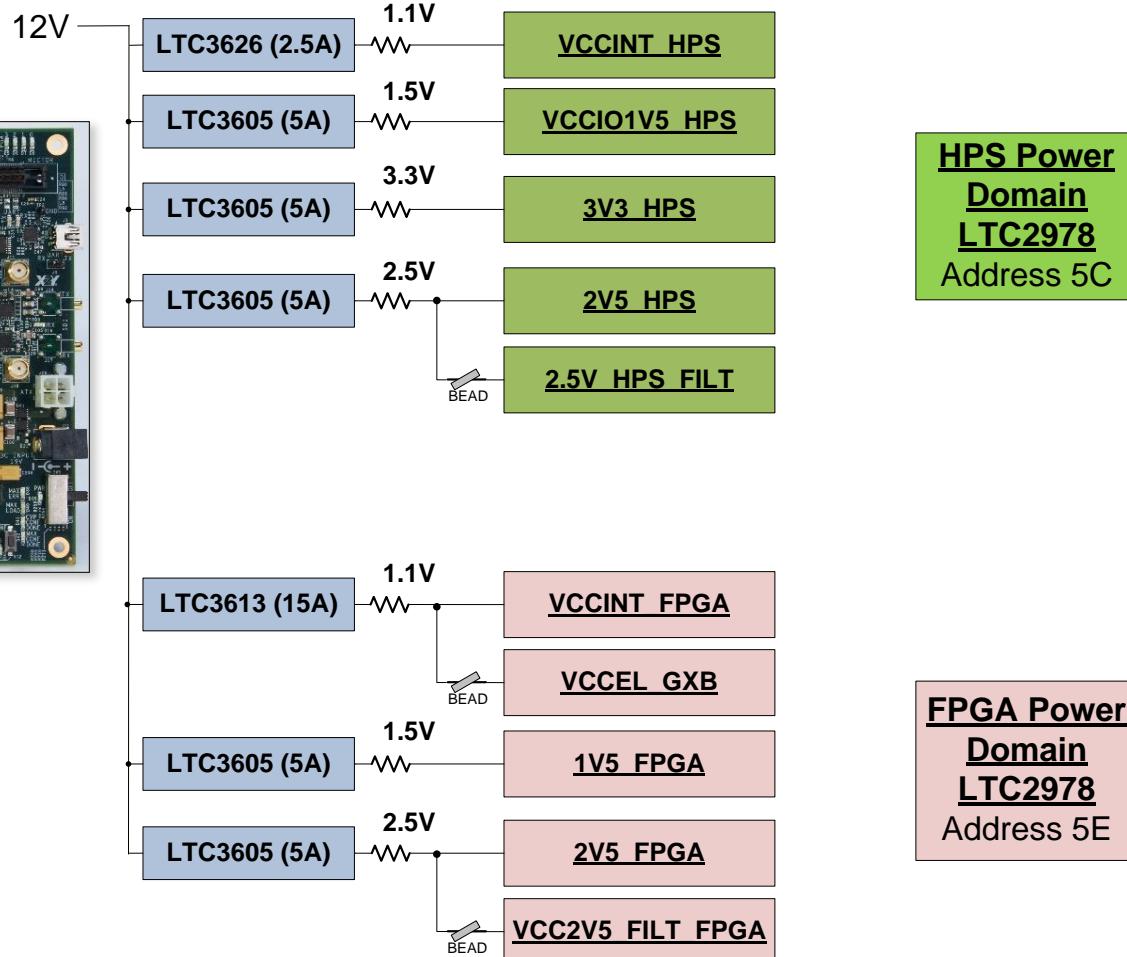
Third-Party Solutions

Analog Devices/Linear Tech
Silicon Labs

Your User-Customizable System on Chip

Intel PSG Cyclone V SoC Block Diagram

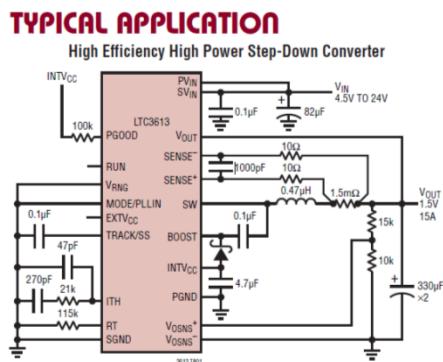
Simplified Cyclone V SoC Kit Power Tree



LTC Regulators Powering FPGA

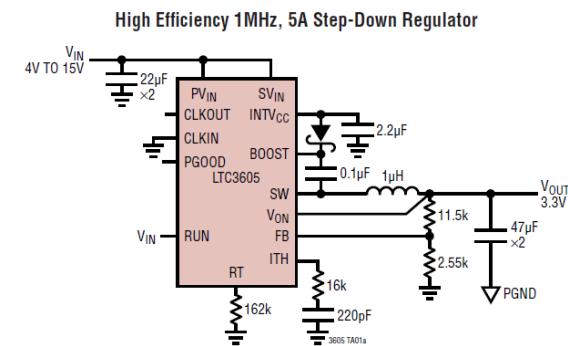
LTC3613 (15A) Monolithic Regulator

- Low I_q
- High efficiency
- Fast transient response
- Overvoltage protection
- Current limit foldback
- Power good status output
- Small package with integrated FETs (7mmx9mm)



2xLTCz3605 (5A) Monolithic Regulator

- Low I_q
- High efficiency
- Excellent line and load transient response
- 0.6V+/- 1% reference accuracy
- Small package with integrated FETs (4mmx4mm)



LTC Regulators Powering ARM

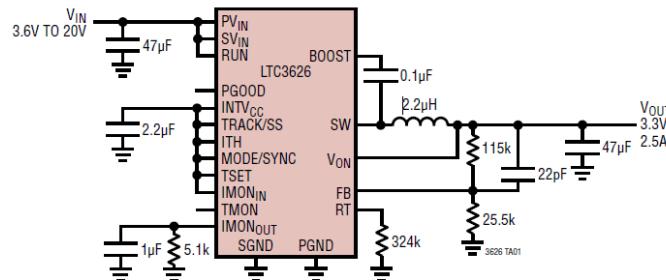
LTC3626 (2.5A) Monolithic Regulator

- Low I_{Q}
- High efficiency
- Programmable average input/output current limit with monitoring
- Short circuit protection
- Power Good status output
- Small package with integrated FETs (7mmx9mm)

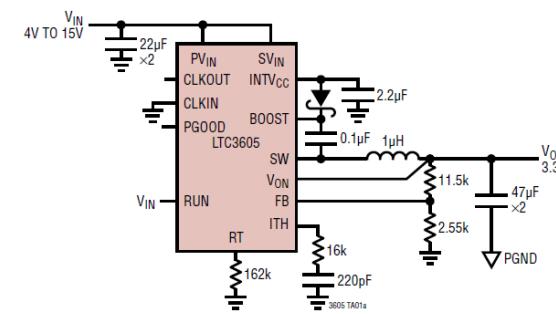
3x LTC3605 (5A) Monolithic Regulator

- Low I_{Q}
- High efficiency
- Excellent line and load transient response
- 0.6V+/- 1% reference accuracy
- Small package with integrated FETs (4mmx4mm)

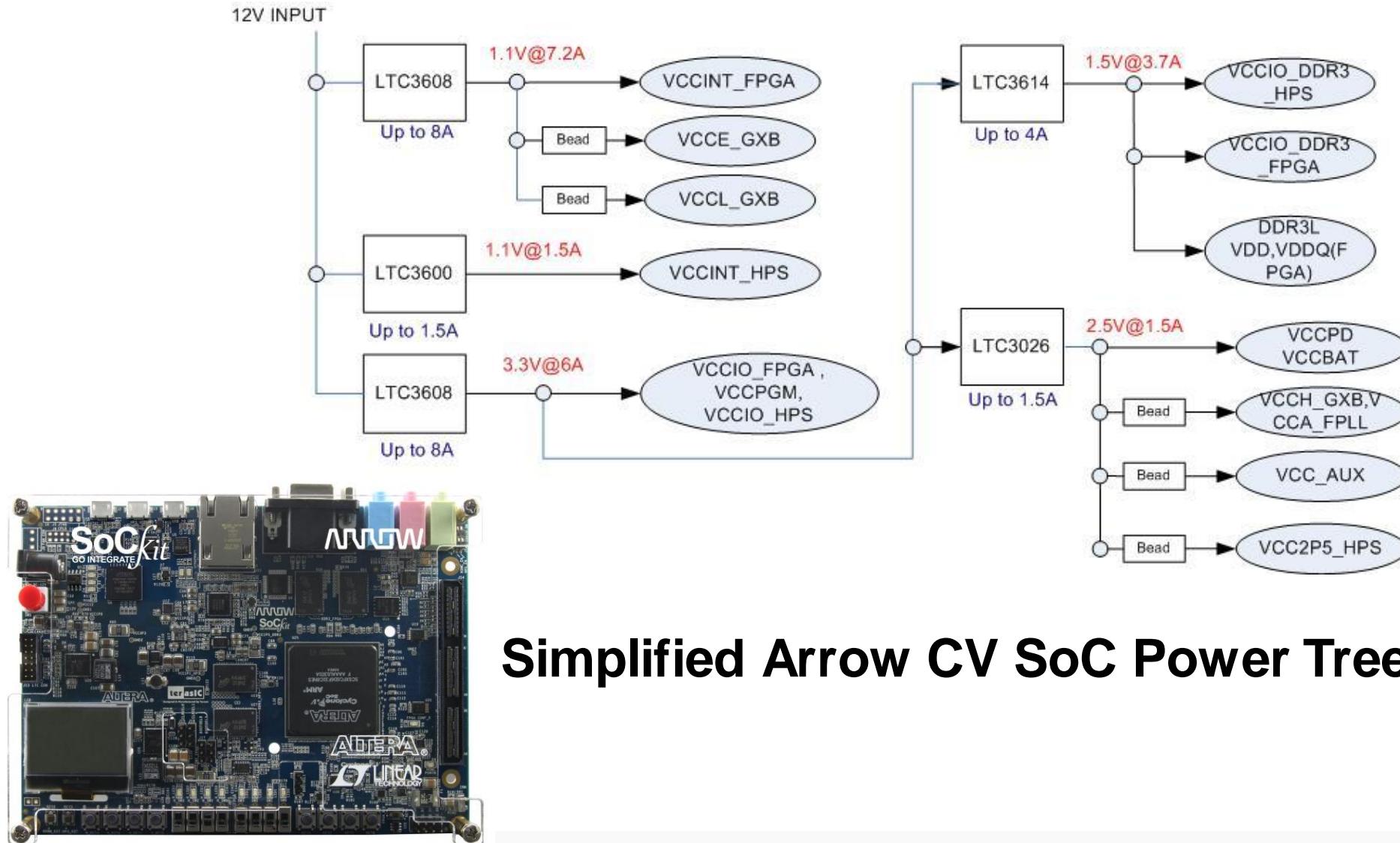
TYPICAL APPLICATION



High Efficiency 1MHz, 5A Step-Down Regulator



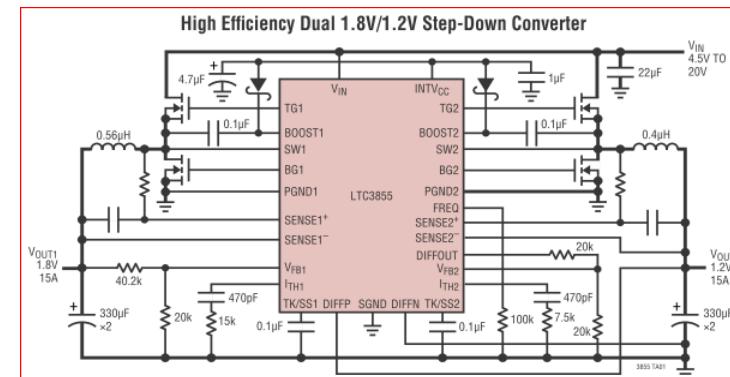
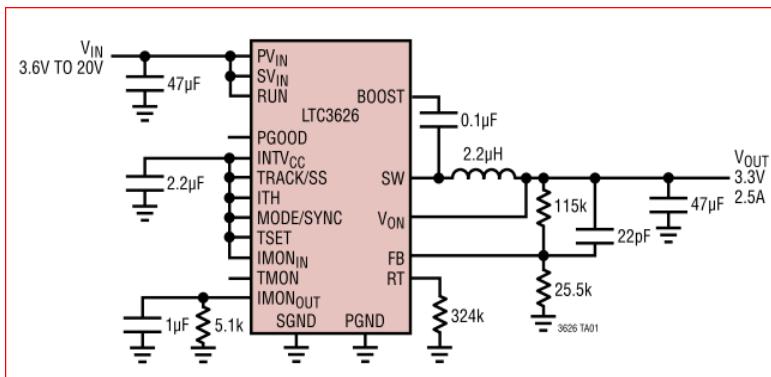
Arrow Cyclone V SoC Block Diagram



Simplified Arrow CV SoC Power Tree

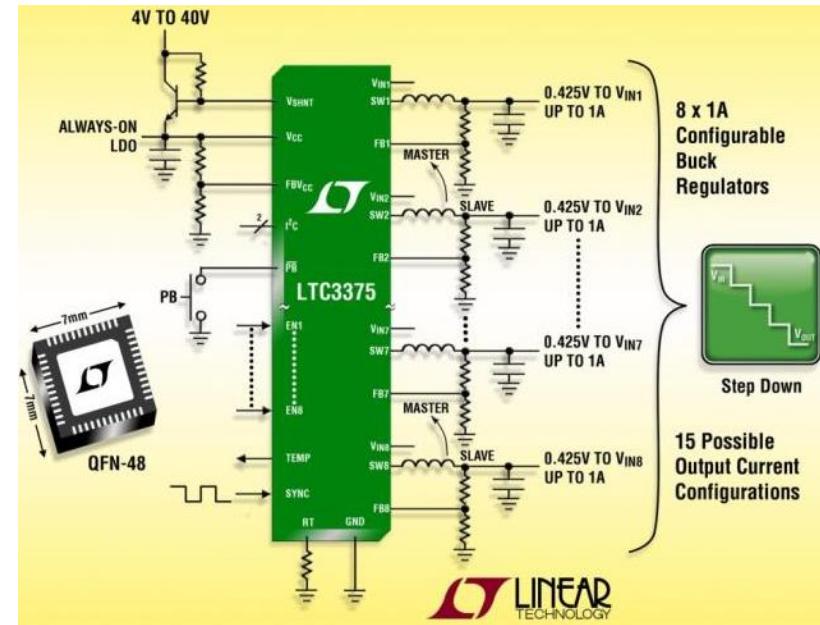
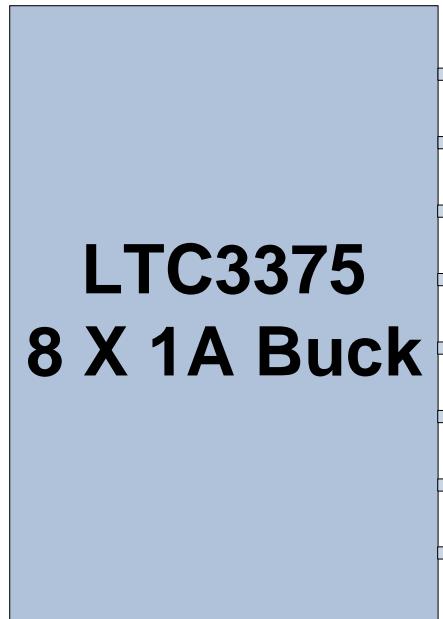
CV SoC Power Supply Options

VIN	Output Current		
	2A	5A	10A
<5V	LTC3615 Monolithic, LTC3618 Monolithic	LTC3616 Monolithic, LTC3415 Monolithic	LTC3613 Monolithic, LTC3610 Monolithic
<12V	LTC3626 Monolithic, LTC3633A Dual Monolithic	LTC3605 Monolithic, LTC3611 Monolithic	LTC3866 Controller, LTC3855 Dual Controller

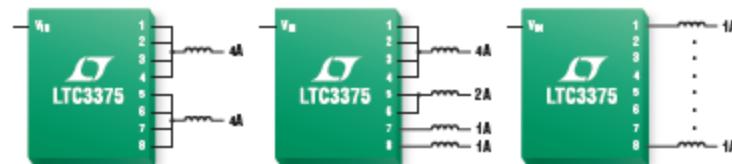


Single IC Option for Applications that only require up to 8A - Eight Output LTC3375 VIN < 5.5V

Potential Example Application



Outputs Easily Parallelized



Silicon Labs – Leading Timing Solutions

■ Si5338 Any-rate Programmable Clock Generators

- **Flexible** – Any frequency, format, voltage on each output clock
 - Si5338: customizable clock with I2C programmability
 - Si5335: customizable clock for cost-sensitive applications
- **Easy to use** – Free customization with two week lead times
- **High performance** – Low phase jitter at less than 1ps-rms



www.silabs.com/timing



■ Si501 CMEMS Programmable Oscillator



- **Flexible** – supports any frequency 32 kHz – 156.25 MHz
 - Si501/2/3/4 single, dual, quad and programmable oscillators
 - Customizable on the web or in the field
- **Simple** – 4-pin DFN in 2025, 3225, 3240 industry-standard packages
- **Reliable** – excellent, guaranteed performance over 10 years
 - Stability includes operating temperature range, initial frequency, power supply, solder shift, load pulling, shock and vibration

CMEMS™

Next Steps

Your User-Customizable System on Chip

Documentation and Useful References

- Intel PSG Cyclone V Documentation page:

<https://www.intel.com/content/www/us/en/programmable/products/soc/portfolio/cyclone-v-soc/support.html>

Useful items found on the documentation page:

- [Cyclone V SoC HPS Address Map and Register Definitions](#)
- [Cyclone V Device Handbook, Volume 3: Hard Processor System Technical Reference Manual \(HPS TRM\)](#)

- Arrow SoCKit page on RocketBoards.org:

<http://rocketboards.org/foswiki/Documentation/ArrowSoCKitEvaluationBoard>

- Support info & Links

- [Intel PSG My Support](#) – for submitting Service Requests – GO SIGN UP!
- [Intel's Design Resources Site](#)

- Intel PSG Forum Community: <https://forums.intel.com/s/>

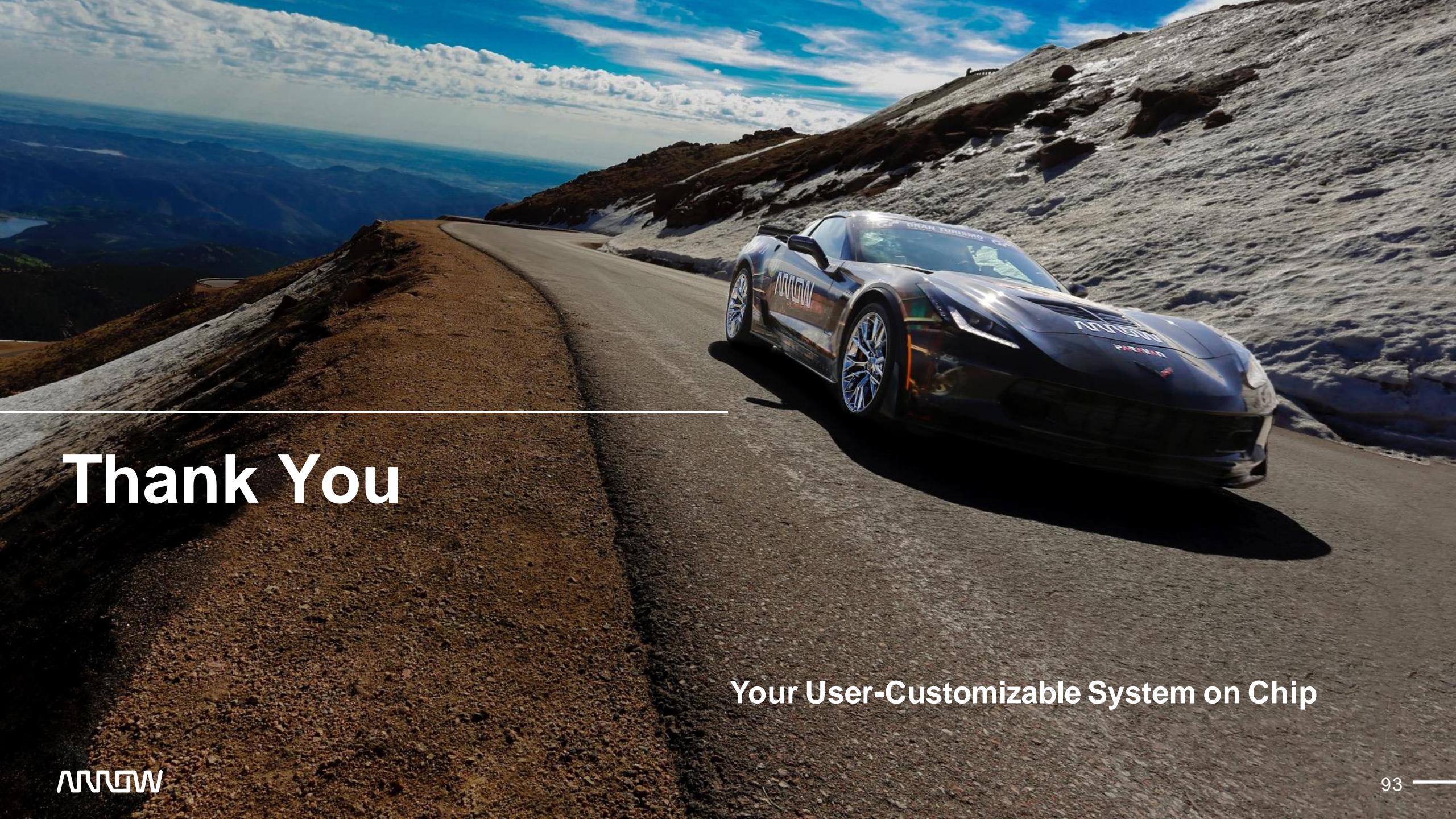
- Intel PSG Wiki Community: www.IntelPSGwiki.com

Attend Intel PSG Training For Further Depth

■ Intel PSG offers several SoC training classes

- [Using Intel® SoC FPGAs \(ISOC103\)](#)
 - This will course focuses on the hardware aspects of using the processor in an SoC from the design, verification and debug hardware perspectives
- [Software Design Flow for an ARM-based SoC \(OSOC2000\)](#)
 - This course is intended for low level software and firmware engineers and examines the software design flow required to implement software for an Intel ® FPGA SoC with the ARM®-based hard processing system (HPS).
- [Getting Started with Linux* OS for Intel® SoC FPGAs \(OSOCLINUX\)](#)
 - This course discusses the various Linux options available for Intel® SoC FPGAs with integrated Arm* Cortex* processors.





A dark blue sports car, possibly a Corvette, is shown driving on a winding asphalt road high in the mountains. The road is bordered by a rocky embankment on the left and a steep, snow-covered hillside on the right. The sky is bright blue with scattered white clouds. The car's body reflects the surrounding landscape.

Thank You

Your User-Customizable System on Chip

APPENDIX



Power Supply Topologies

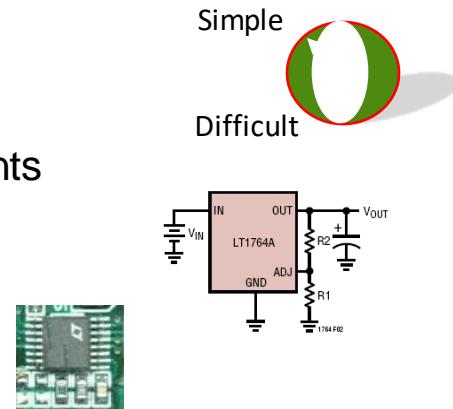
Power Supply Topologies

Your User-Customizable System on Chip

Power Management Solution Options

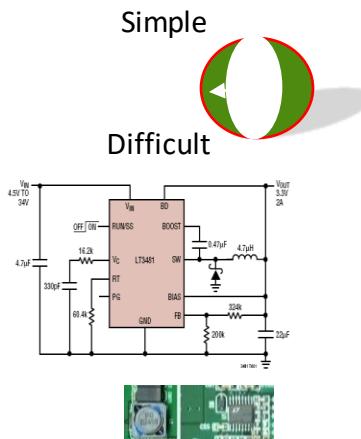
▪ Linear Regulators

- Simple
- Low noise
- Few components
- Low power



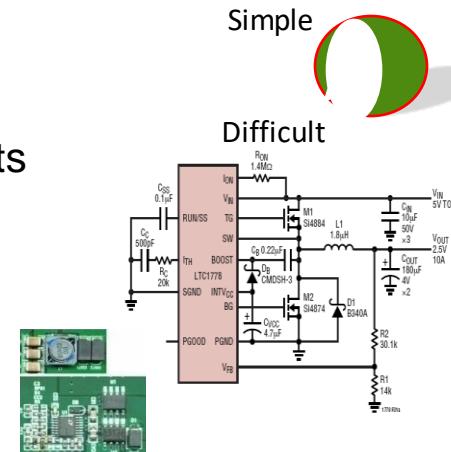
▪ Monolithic Switchmode Regulators

- More components than linear reg.
- Layout Consideration
- Mid-to-high power
- More efficient



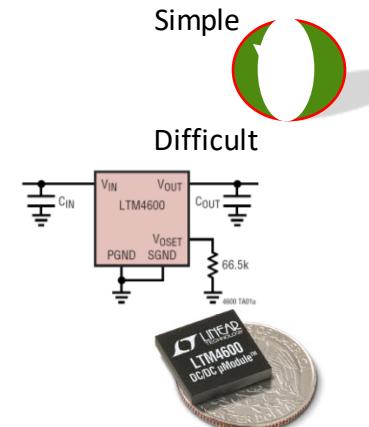
▪ Switchmode Controller Regulators

- More components than monolithic
- More Complex layout
- Mid-to-very high power, efficient



▪ μModule Regulator Systems

- Complete circuit in a tiny package
- Simple layout
- Efficient
- Multi-output
- Low-to-high power
- **New:** Low noise...

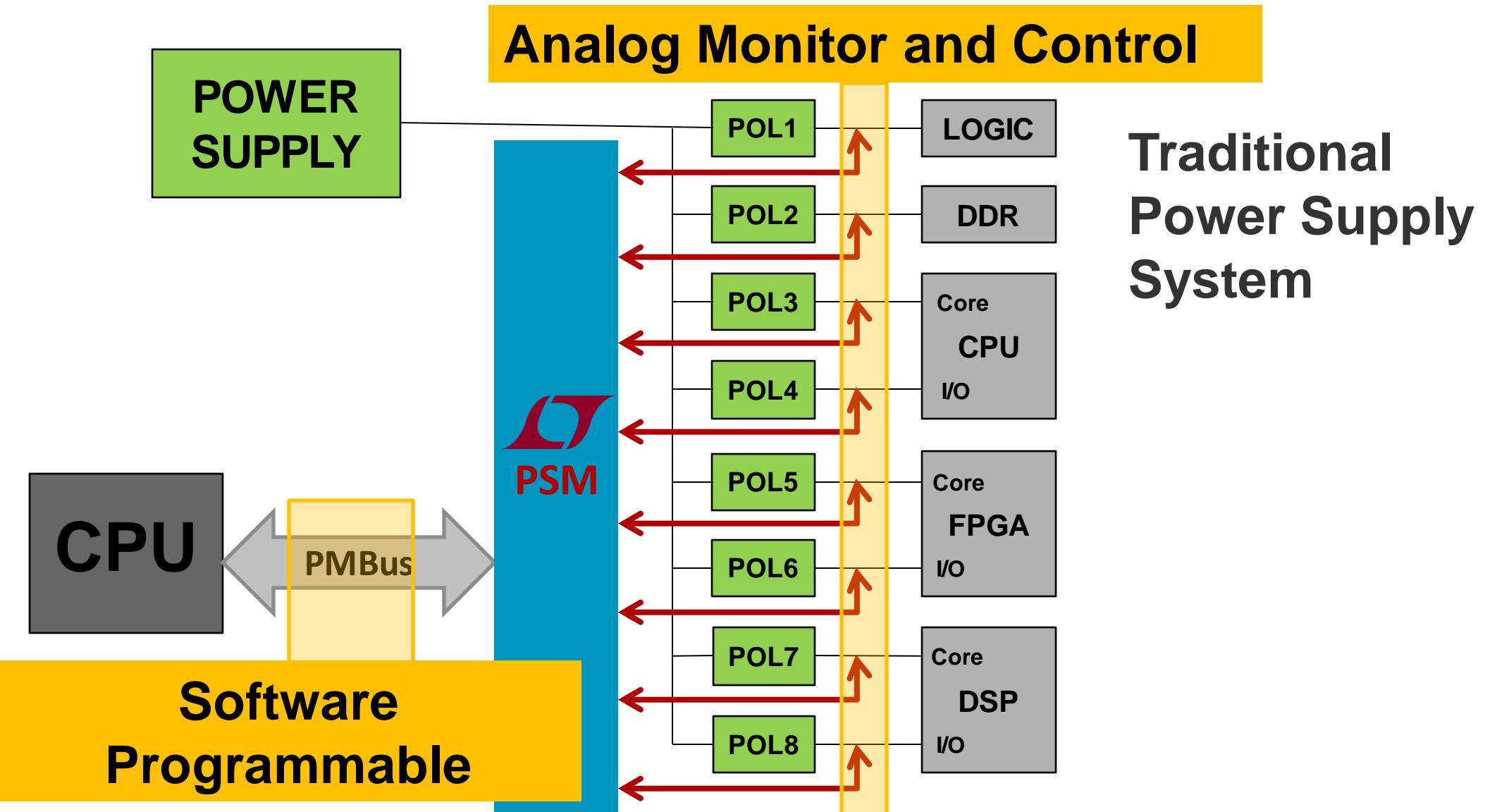


Power System Management

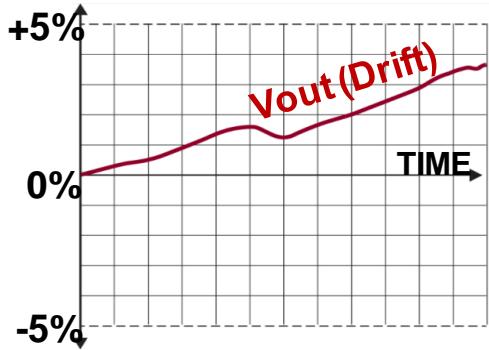
Power System Management

Your User-Customizable System on Chip

What is Power System Management (PSM) ?

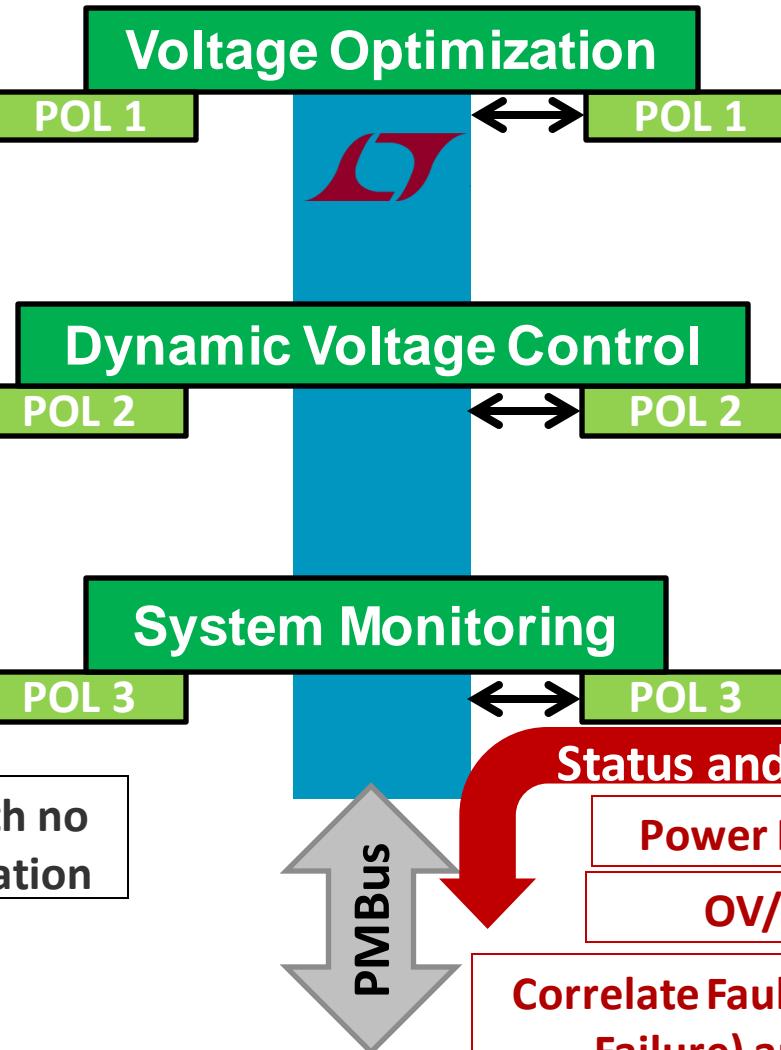


Traditional Power Supply

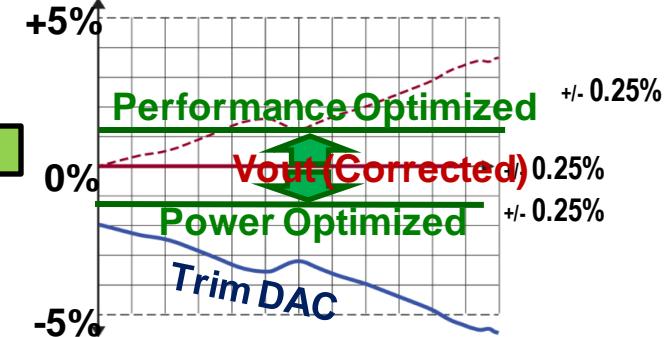


Limited by tolerance % of components.
Still subject to time/temp drift

OV/UV/OC Faults with no other system information



PSM with LTpowerPlay™



DVFS
FPGA/DSP/ASIC
1.50V / 1Ghz
1.15V / .6Ghz



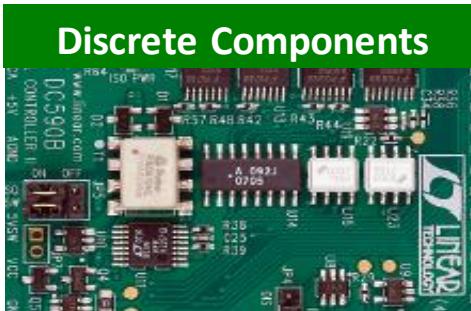
Status and Fault Monitoring

Power In and Power Out
OV/UV/OC Faults

Correlate Faults to System Status (Predict Failure) and "Black Box" Analysis

Traditional Power Supply

PSM with LTpowerPlay™



Discrete Components

Sequence / Margin / Track

POL 1



POL 1

Changes may require board to be reworked



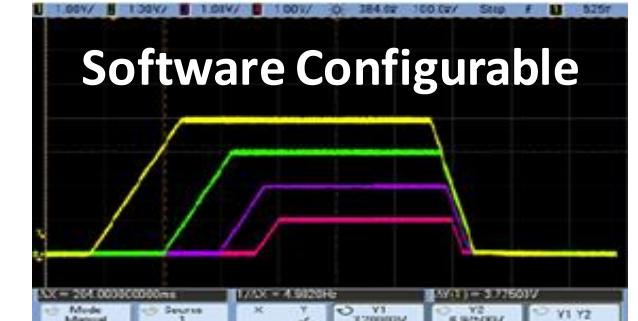
Probes and meters...GOOD LUCK!

Debugging & Troubleshooting

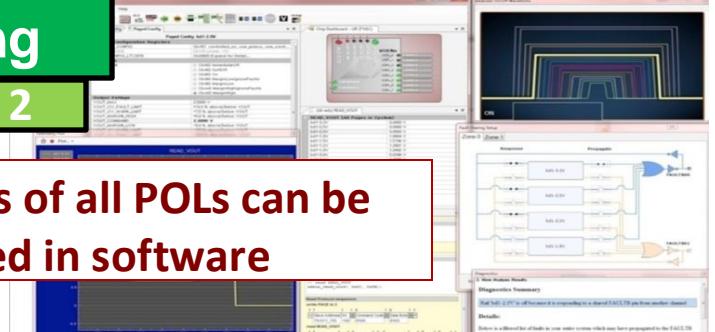
POL 2

POL 2

Status of all POLs can be viewed in software



Unlimited changes by software

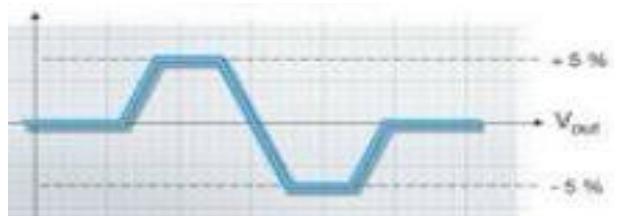


Margin Supplies to Test System Stability

POL 3

POL 3

Can't be easily done



Test system performance with different combinations of voltages



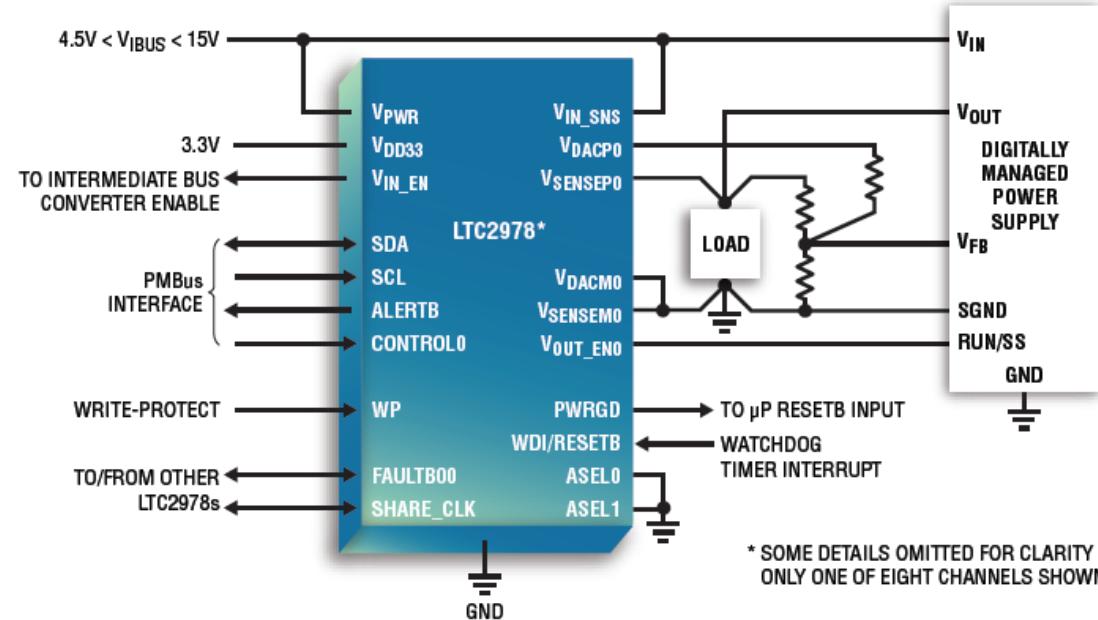
PSM:LTC2978A used on Intel PSG CV SoC Kit



LTC2978
Octal PMBus Power Supply
Monitor and Controller
with EEPROM

Features

- Margin/Trim Eight Supplies to $\pm 0.25\%$ Accuracy
- EEPROM for Configuration, Black Box Fault Logging
- I²C/SMBus Interface, PMBus Compliant
- Supported by LTpowerPlay GUI
- Operates Autonomous Without Additional Software
- Eight-Supply Sequencer, Time Based
- Monitor and Supervise:
 - Input and Eight Output Voltages
 - Optional Current Monitor on Odd Channels
 - Die Temperature
- Watchdog Timer
- Powered from 3.3V, or 4.5V to 15V
- 64-Pin 9mm x 9mm QFN Package



PSM User Friendly Interactive GUI

Complete Development Platform with LTpowerPlay GUI

