Open Processor Foundation

The Project : An Open Platform

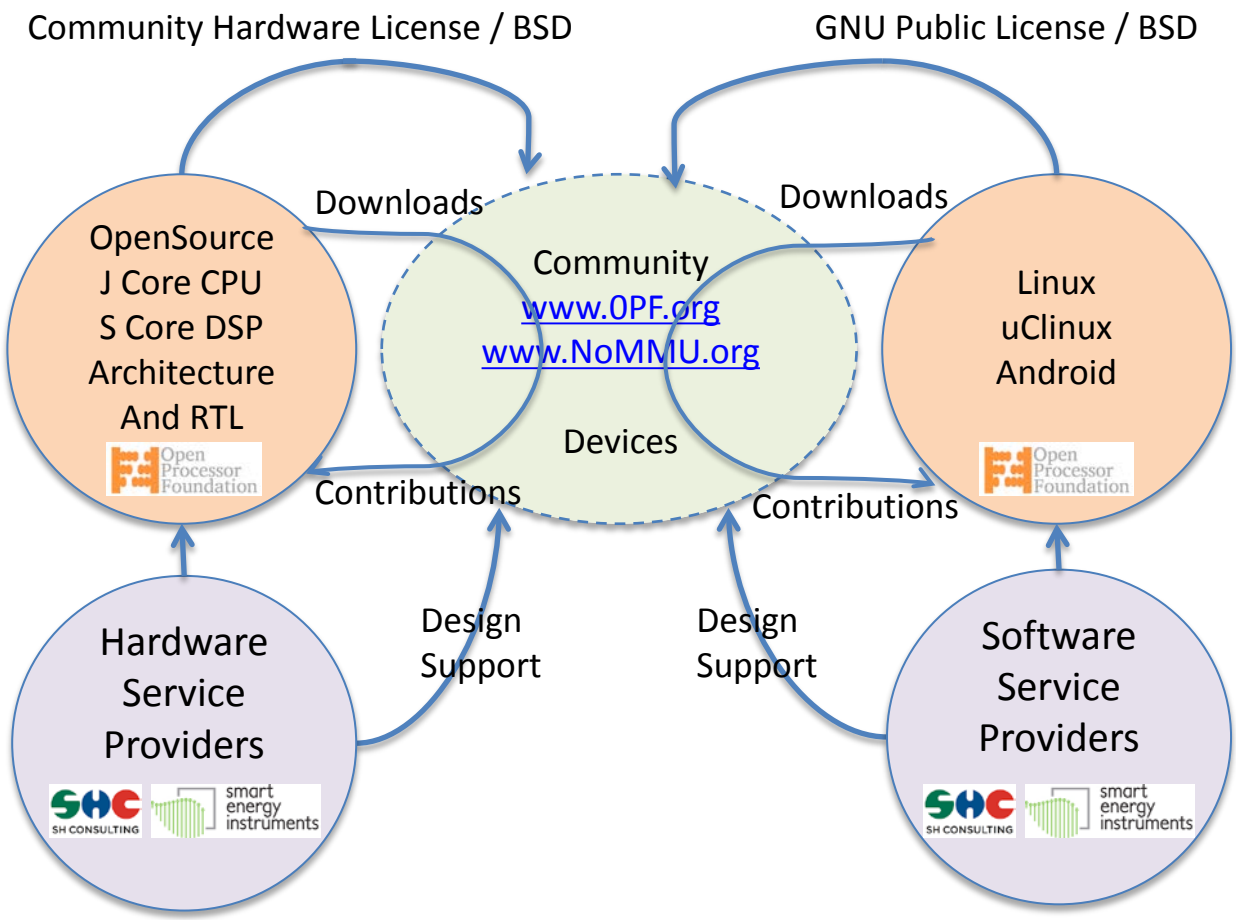
- SuperH Compatible Open Chip Platform & DSPs
  - Why these choices, and why now?

- Open Chip Designs, Open Hardware
- Open Software Platform
- Community
  - www.NoMMU.org www.0PF.org www.uClinux.org

- Silicon, Hardware, Signal Processing, OS & Software
  - IoT Platform

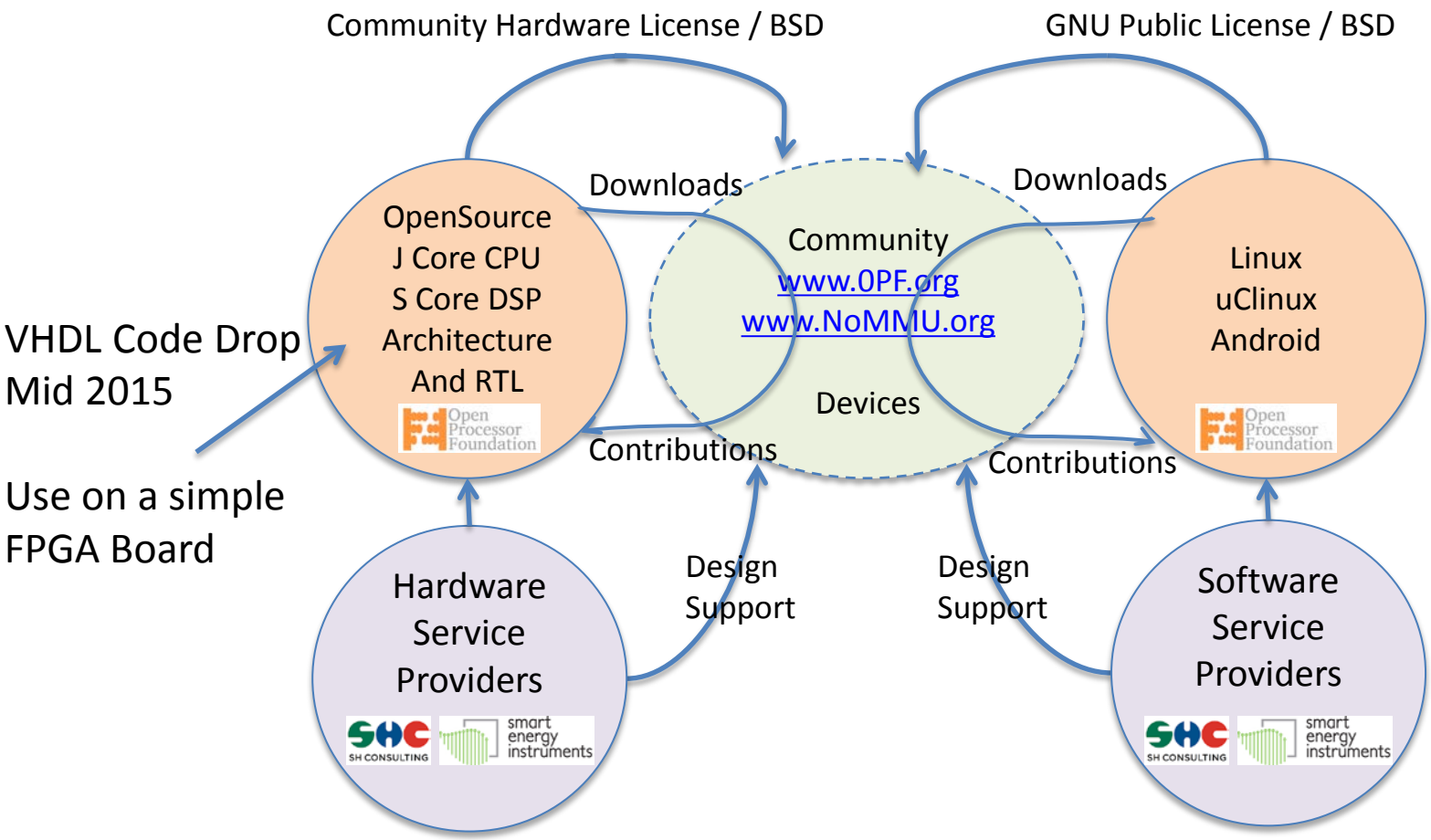
Community Design & License

- OPF Certifies FPGA and ASIC. Validation Vectors.
- Professional Support from Multiple Service Providers Distributed Globally
- Liberal License Terms: BSD and Community HW License



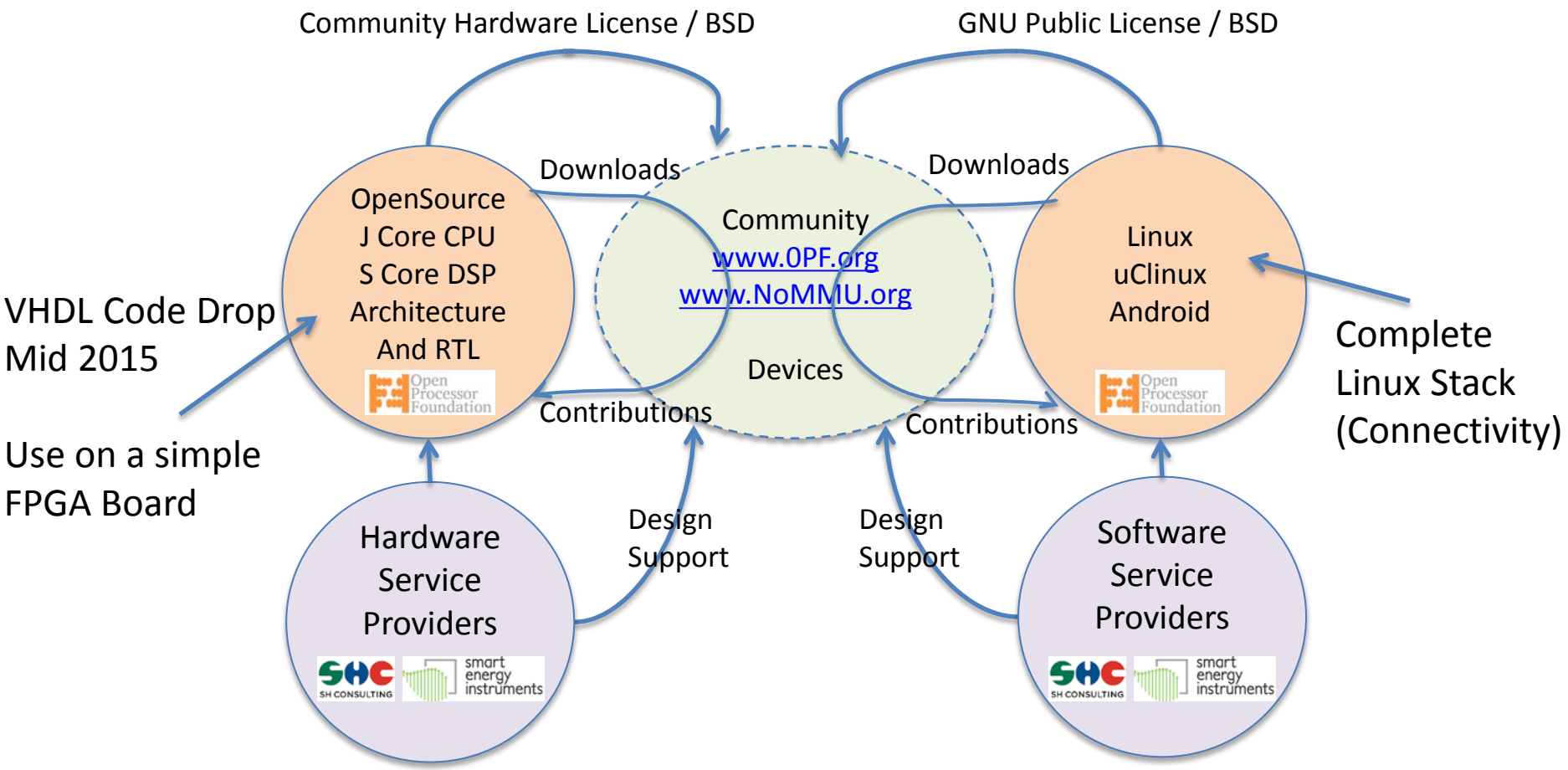
Community Design & License

- OPF Certifies FPGA and ASIC. Validation Vectors.
- Professional Support from Multiple Service Providers Distributed Globally
- Liberal License Terms: BSD and Community HW License



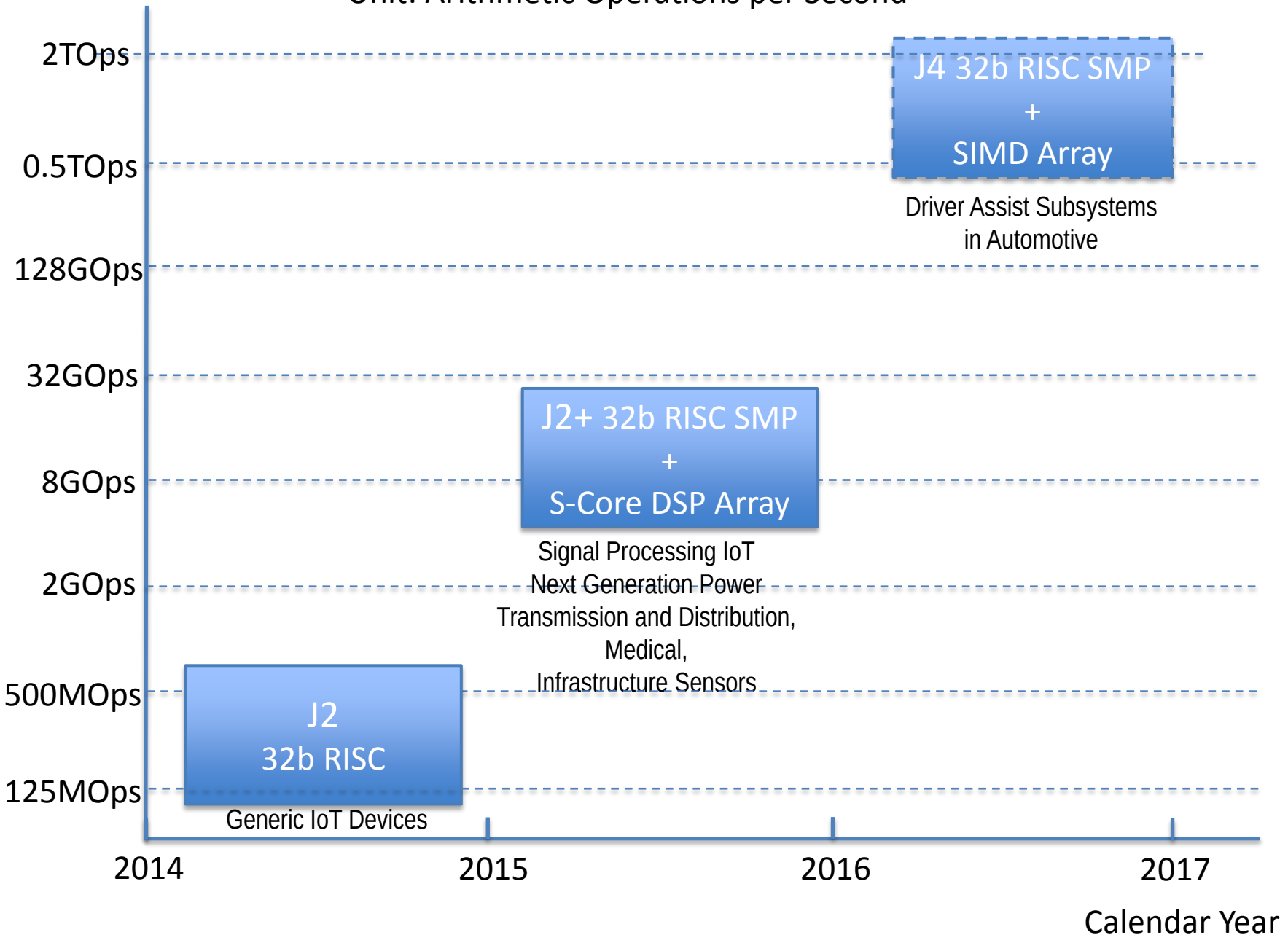
Community Design & License

- OPF Certifies FPGA and ASIC. Validation Vectors.
- Professional Support from Multiple Service Providers Distributed Globally
- Liberal License Terms: BSD and Community HW License



J Series Computation Core Cluster Roadmap

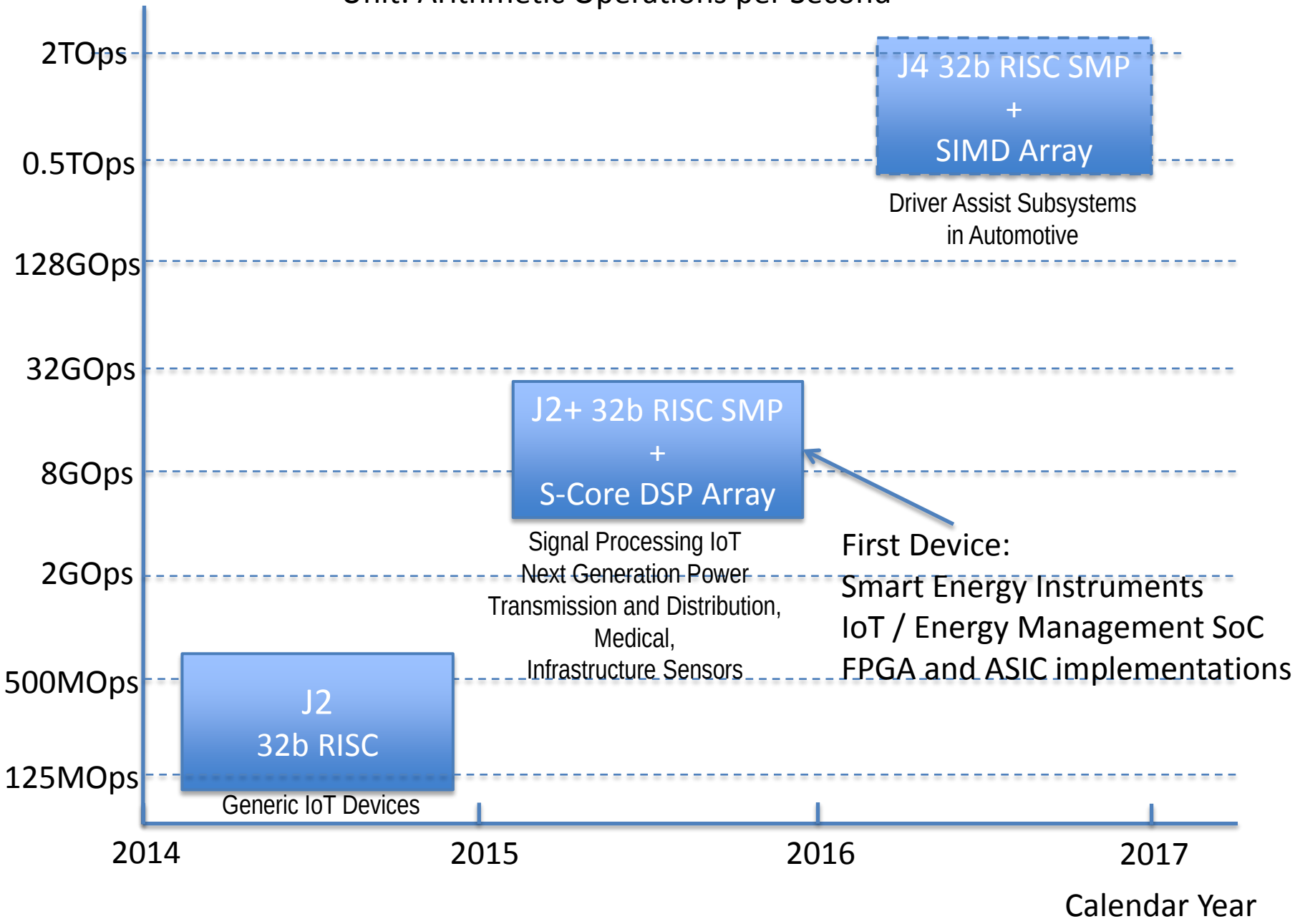
Unit: Arithmetic Operations per Second



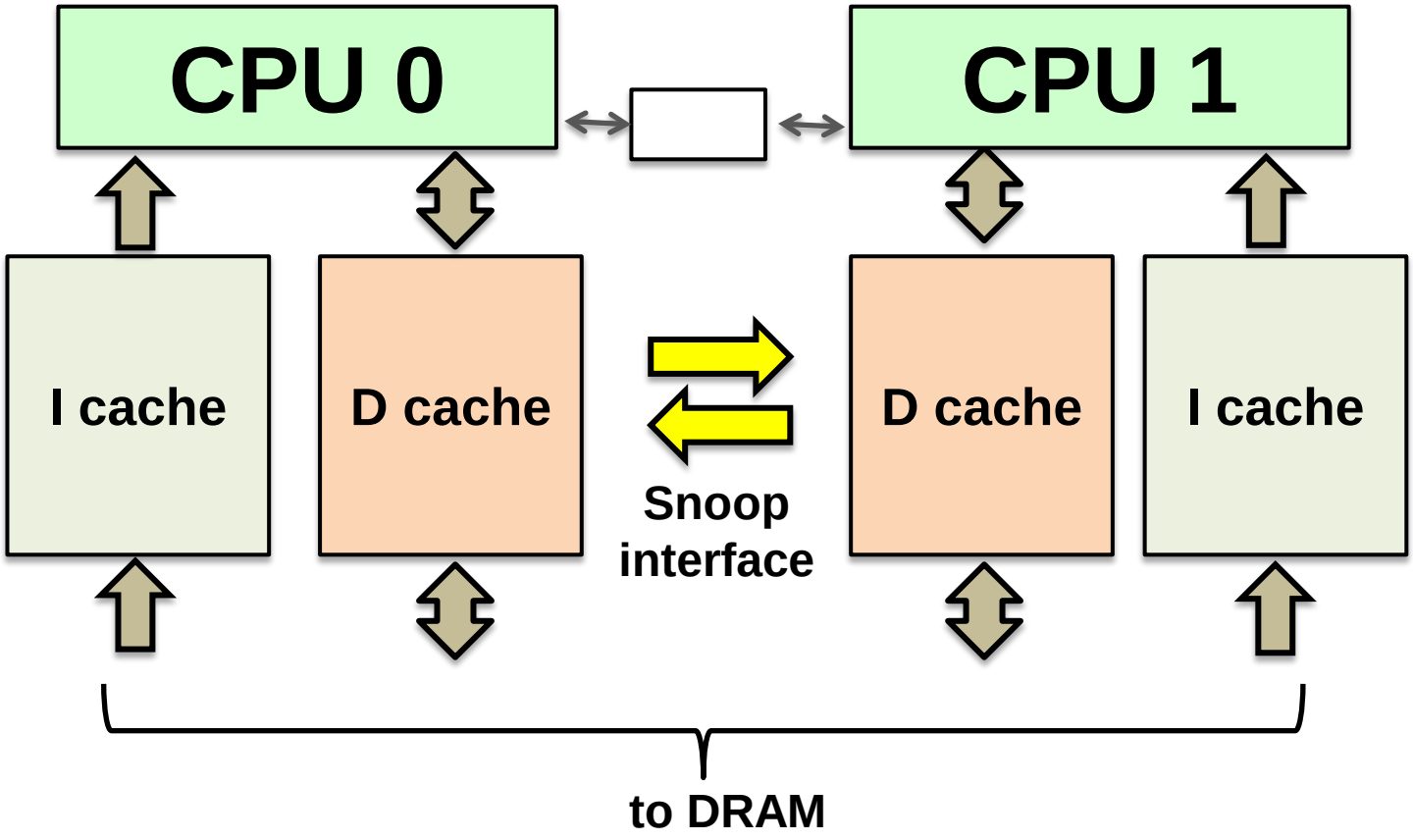
J Series Computation Core Cluster Roadmap

Open Processor Foundation

Unit: Arithmetic Operations per Second



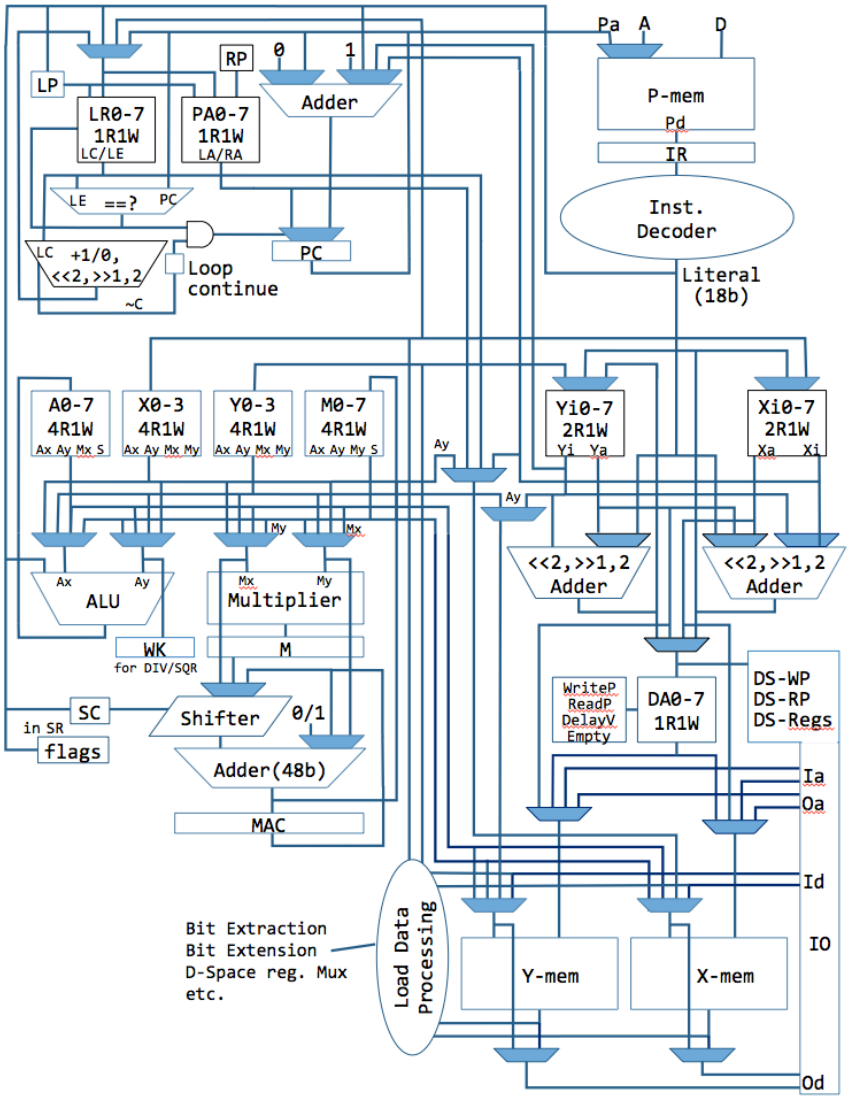
Multiprocessor data coherency



D-cache hardware-based snooping

18/36b S-Core DSP

- Development in Progress (Target Completion: June 2015)



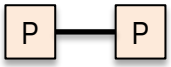
DSP ISA and Code Map

ALU & MAC	35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	primary set
	0 0 ALU Xa Xd X Y MAC Ya Yd	Xa0(Xi4),Xc0(Xi6),Xi0,Xi2
CAX Xin	Xin: Xi0-2, Xa:Xi{2b'10,n&1b'1} Xi	Ya0(Yi4),Yc0(Yi6),Yi0,Yi2
NOP	1 1 Xa+=Xi(w/Mask & Bit Rev.)	secondary set
STX Xd	Xd: A0,2,4,6,M0,2,4,6 Xd 0	Xa1(Xi5),Xc1(Xi7),Xi1
LDX Xd	Xd: X0-4 0 Xd 1 @Xa = Xd(Xa -> DA[rp,wp]*) *)use DA	Ya1(Yi5),Xc1(Xi7),Yi1
LSWX	1 1 0 1 Xd = @Xa(Xa -> DA[wp] *) in some casess	Xi3,Yi3: always "0"
NOP	1 1 1 1 LP = ~LP (for two set case)	
reserved	others	Xc/Yc
CAY Yin	Yin: Yi0-3, Ya:Yi{2b'10,n&1b'1} Ya+=Yi(w/Mask & Bit Rev.) Yi	[17] use cntrol field
NOP	Ya =Yi4 1 1	[16] "M" mask
STY Xy	Yd: A1,3,5,7,M1,3,5,7 0 @Ya = Yd(Ya -> DA[rp,wp]*) Yd	[15] "R" bit reverse
LDY Xy	Yd: Y0-4 1 Yd = @Ya(Ya -> DA[wp] *) 0 Xd	[14:13] DA mode
LSWY	1 LP = ~LP (for two set case) 1 1 0	00: "_" nop
NOP	1 1 1 1	01: "D" 1-cycle Delay
reserved	1 others	10: "P" Pair
ADD Ax,Ay,Az	0 0 Ax Ay Az Ax:X0,2/Y0,2/A0-3, Ay:X1,3/Y1,3/M0-3, Az:all (C,Az)= Ax + Ay	11: "F" Pair Load/Store
SUB Ax,Ay,Az	0 1 Ax Ay Az Ax:X0,2/Y0,2/A0-3, Ay:X1,3/Y1,3/M0-3, Az:all (C,Az)= Ax - Ay	(Full four accesses)
ADD Ax,Ay,Az	1 0 0 0 Ay Ax Az Ax:X0,2/Y0,2/A0-3, Ay:A2-3, Az:all (C,Az)= Ax + Ay	[8:4] encoded mask position
SUB Ax,Ay,Az	1 1 0 0 Ay Ax Az Ax:X0,2/Y0,2/A0-3, Ay:A2-3, Az:all (C,Az)= Ax - Ay	1-17, the other is reerved
ADD Ax,Ay,Az	1 0 0 1 Ax Ay Az Ax:M0-1, Ay:X1,3/Y1,3/M0-3, Az:all (C,Az)= Ax + Ay	[1:0] size
SUB Ax,Ay,Az	1 1 0 1 Ax Ay Az Ax:M0-1, Ay:X1,3/Y1,3/M0-3, Az:all (C,Az)= Ax - Ay	00: 18b
ADC Ax,Ay,Az	1 1 1 0 0 Ax Ay Az Ax:X2/Y2/A0,2, Ay:X3/Y3/M1,3, Az:A1,3,5,7 (C,Az)= Ax + Ay + C	01: 1b(±1)
SBC Ax,Ay,Az	1 1 1 0 1 Ax Ay Az Ax:X2/Y2/A0,2, Ay:X3/Y3/M1,3, Az:A1,3,5,7 (C,Az)= Ax +~Ay + C	
ADC Ax,Ay,Az	1 1 1 1 0 0 0 Ax Az Ax:X2/Y2/A0,2, Ay:A3, Az:A1,3,5,7 (C,Az)= Ax + Ay + C	
ADC Ax,Ay,Az	1 1 1 1 0 0 1 Ay Az Ax:M1, Ay:X3/Y3/M1,3, Az:A1,3,5,7 (C,Az)= Ax + Ay + C	
SBC Ax,Ay,Az	1 1 1 1 0 1 0 Ax Az Ax:X2/Y2/A0,2, Ay:A3, Az:A1,3,5,7 (C,Az)= Ax +~Ay + C	
SBC Ax,Ay,Az	1 1 1 1 0 1 1 Ay Az Ax:M1, Ay:X3/Y3/M1,3, Az:A1,3,5,7 (C,Az)= Ax +~Ay + C	
ATN Ay,Az	1 1 1 1 1 0 0 Ax Az Ay:X3/Y3/M1,3, Az:A1,3,5,7 explained later	
SCN0 Ax,Ay,A4	1 1 1 1 1 0 1 Ay Ax Ax:X0,2/Y0,2, Ay:X3/Y3/M1,3 explained later	
SCN1 Ay,A4	1 1 1 1 1 1 0 0 0 Ax Ay:X3/Y3/M1,3 explained later	
LSWX #n	1 1 1 1 1 1 1 0 0 n LP = n (Swich to n-th LOOP control set)	
NOP	1 1 1 1 1 1 1 1 1 1	
reserved	other patterns	
MAD Mx,My,Mz	Mx:X0-1/Y0-1/A4-7,My:X2-3/Y2-3/M4-7,Mz:all 0 0 Mx My Mz (Mz,MAC)= S(MAC + Mx * My)	
MSB Mx,My,Mz	Mx:X0-1/Y0-1/A4-7,My:X2-3/Y2-3/M4-7,Mz:all 0 1 Mx My Mz (Mz,MAC)= S(MAC - Mx * My)	
MUL Mx,My,Mz	Mx:X0-1/Y0-1/A4-7,My:X2-3/Y2-3/M4-7,Mz:all 1 0 Mx My Mz (Mz,MAC)= SR(Mx * My)	
MAD Mx,Mz	Mx:X0-1/Y0-1/A4-7, Mz:all 1 1 0 0 0 Mx Mz (Mz,MAC)= S(MAC + Mx^2)	
MAD My,Mz	My:X2-3/Y2-3/M4-7,Mz:all 1 1 0 0 1 My Mz (Mz,MAC)= S(MAC + My^2)	
MSB Mx,Mz	Mx:X0-1/Y0-1/A4-7, Mz:all 1 1 0 1 0 Mx Mz (Mz,MAC)= S(MAC - Mx^2)	
MSB My,Mz	My:X2-3/Y2-3/M4-7,Mz:all 1 1 0 1 1 My Mz (Mz,MAC)= S(MAC - My^2)	
MUL Mx,Mz	Mx:X0-1/Y0-1/A4-7, Mz:all 1 1 1 0 0 Mx Mz (Mz,MAC)= SR(Mx^2)	
MUL My,Mz	My:X2-3/Y2-3/M4-7,Mz:all 1 1 1 0 1 My Mz (Mz,MAC)= SR(My^2)	
MULD Mx,My,Mz	Mx:X01/Y01/A45,67,My:X23/Y23/M45,67,Mz:M0123,4567 1 1 1 1 0 0 Mx My Mz MAC=UU(MxL*MyL) ,MzLL=MACL	
MULD Mx,Mx,Mz	Mx:X01/Y01/A45,67, Mz:M0123,4567 1 1 1 1 0 1 0 0 Mx Mz MAC=UU(MxL^2) ,MzLL=MACL	
MULD My,My,Mz	My:X23/Y23/M45,67,Mz:M0123,4567 1 1 1 1 0 1 0 1 My Mz MAC=UU(MyL^2) ,MzLL=MACL	
(2nd step)	MAC=US(MxH*MyL)+MAC>>18	
(3rd step)	MAC=US(MyH*MxL)+MAC ,MzLH=MACL	
(4th step)	MAC=SS(MyH*MxH)+MAC>>18,MzHL=MACL	
(5th step)	MzHH=MACH	
LSWY #n	1 1 1 1 1 1 1 0 0 n LP = n (Swich to n-th LOOP control set)	
NOP	1 1 1 1 1 1 1 1 1 1	

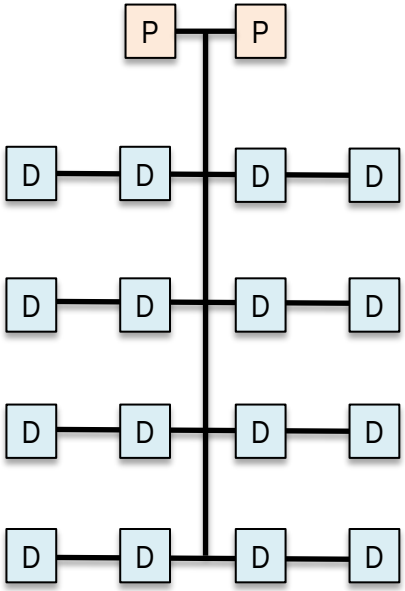
J / S Core On-Chip Computational Clusters

- J2 : RISC (Prototype SoC in FPGA, Customer Projects)
- J2+: RISC SMP + DSP Array (Product SoC in Progress, 152nm silicon process)
- J4 : RISC SMP + N-Dimensional SIMD Array (Under Planning)

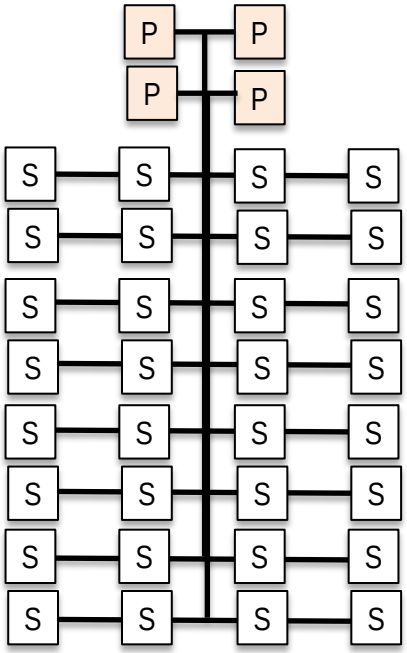
RISC SMP



RISC SMP
+
DSP Array



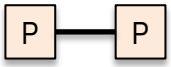
RISC SMP
+
N-Dimensional SIMD Array



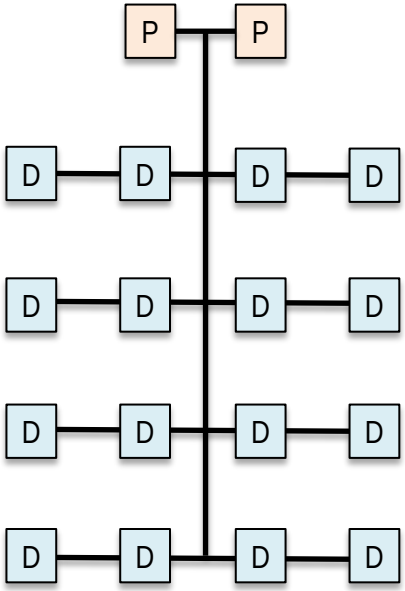
J / S Core On-Chip Computational Clusters

- J2 : RISC (Prototype SoC in FPGA, Customer Projects)
- J2+: RISC SMP + DSP Array (Product SoC in Progress, 152nm silicon process)
- J4 : RISC SMP + N-Dimensional SIMD Array (Under Planning)

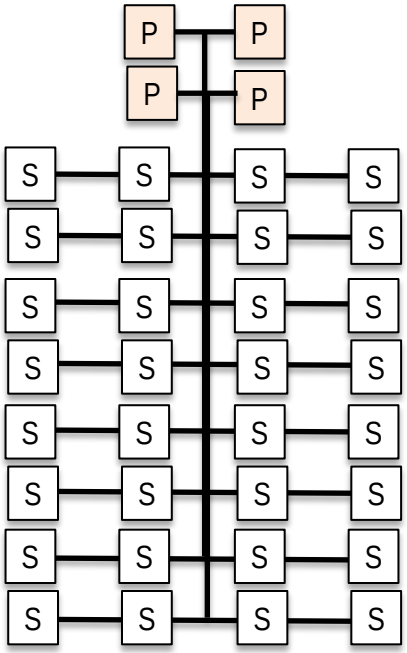
RISC SMP



RISC SMP
+
DSP Array



RISC SMP
+
N-Dimensional SIMD Array



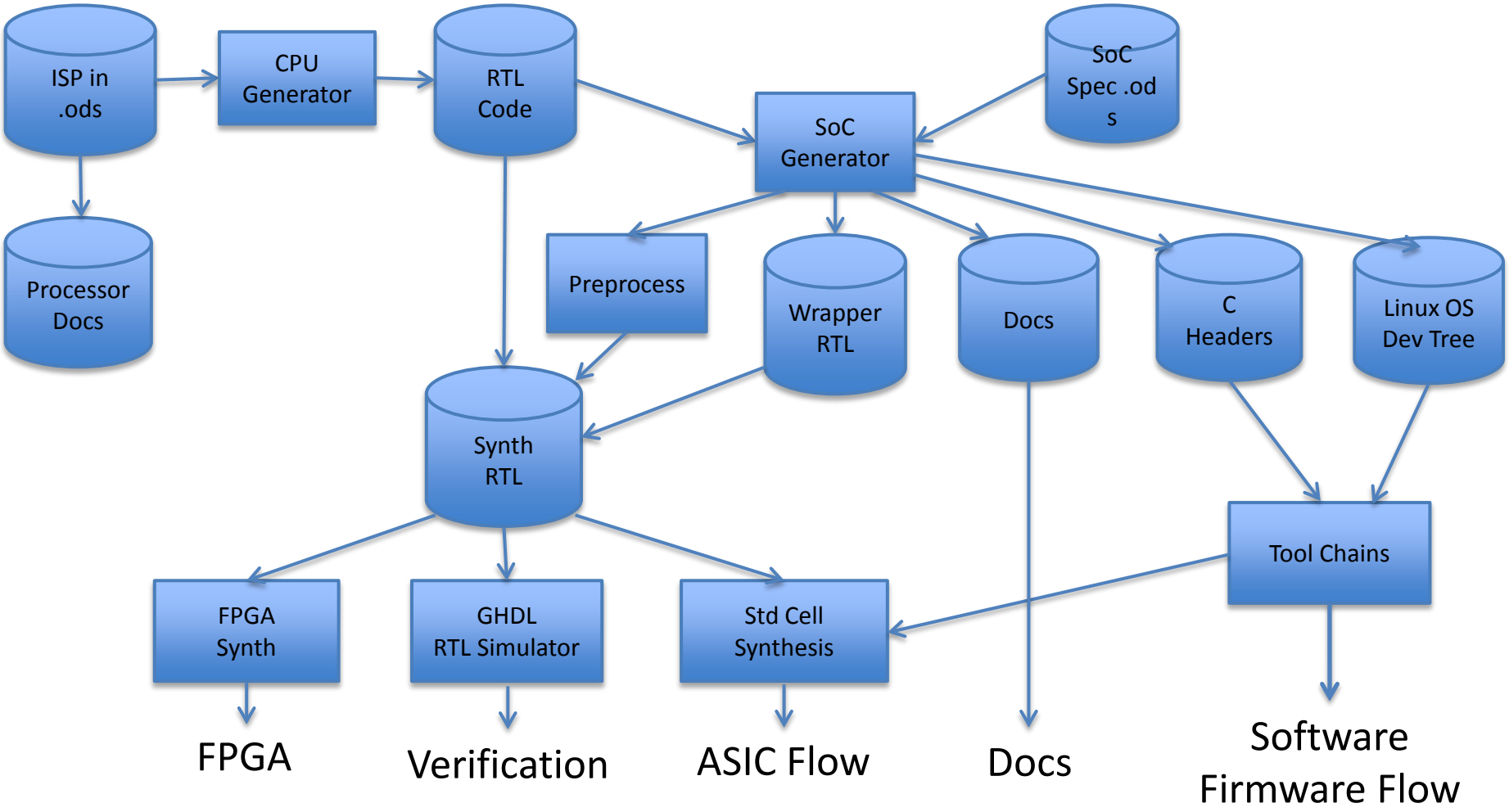
IoT devices are all about multiple Real World Signals…

Signals, Silicon, Software … and Network

Hardware Development Environment

- Simulation & Synthesis
  - GHDL (open VHDL sim)
  - Xilinx ISE (Spartan FPGA)
  - Cadence ASIC toolflow
  - JTAG w/GDB proxy
- FPGA
  - Low Cost Dev Boards
  - System on Module
    - Multiple Vendors
  - Open FPGA Board design
    - (mid 2015)
- ASIC
  - Silicon Proven (2015)
  - Cadence flow, TSMC
  - Low Die Area
    - J2 0.45mm^2 in 0.152nm

RTL Sim->FPGA->ASIC Tool Flow



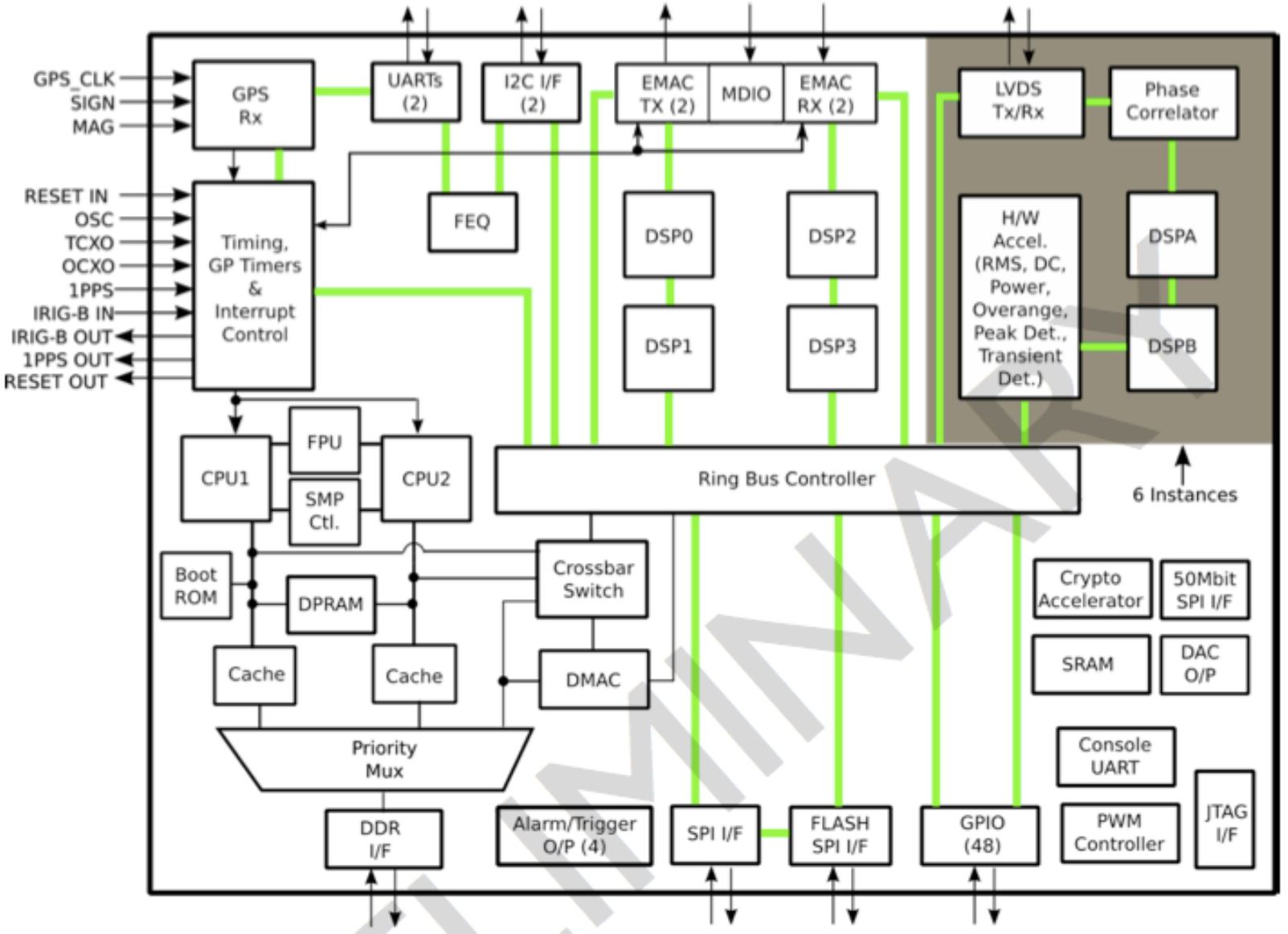
RTL IP

- Patents:
  - All SH2 patents expired in October, 2014 (RIP)
  - SH4 patents expiring in 2016
- Copyrights:
  - New Canadian Engineers wrote initial RTL.
  - Then Original Hitachi Engineers Validated
- Trade Secret:
  - The Biggest Secret is there is no Secret.
- Contracts:
  - Don’t Apply
- Trademarks
  - This is not an SH, but J series is instruction set compatible

First Real World Device!

SGS1010D

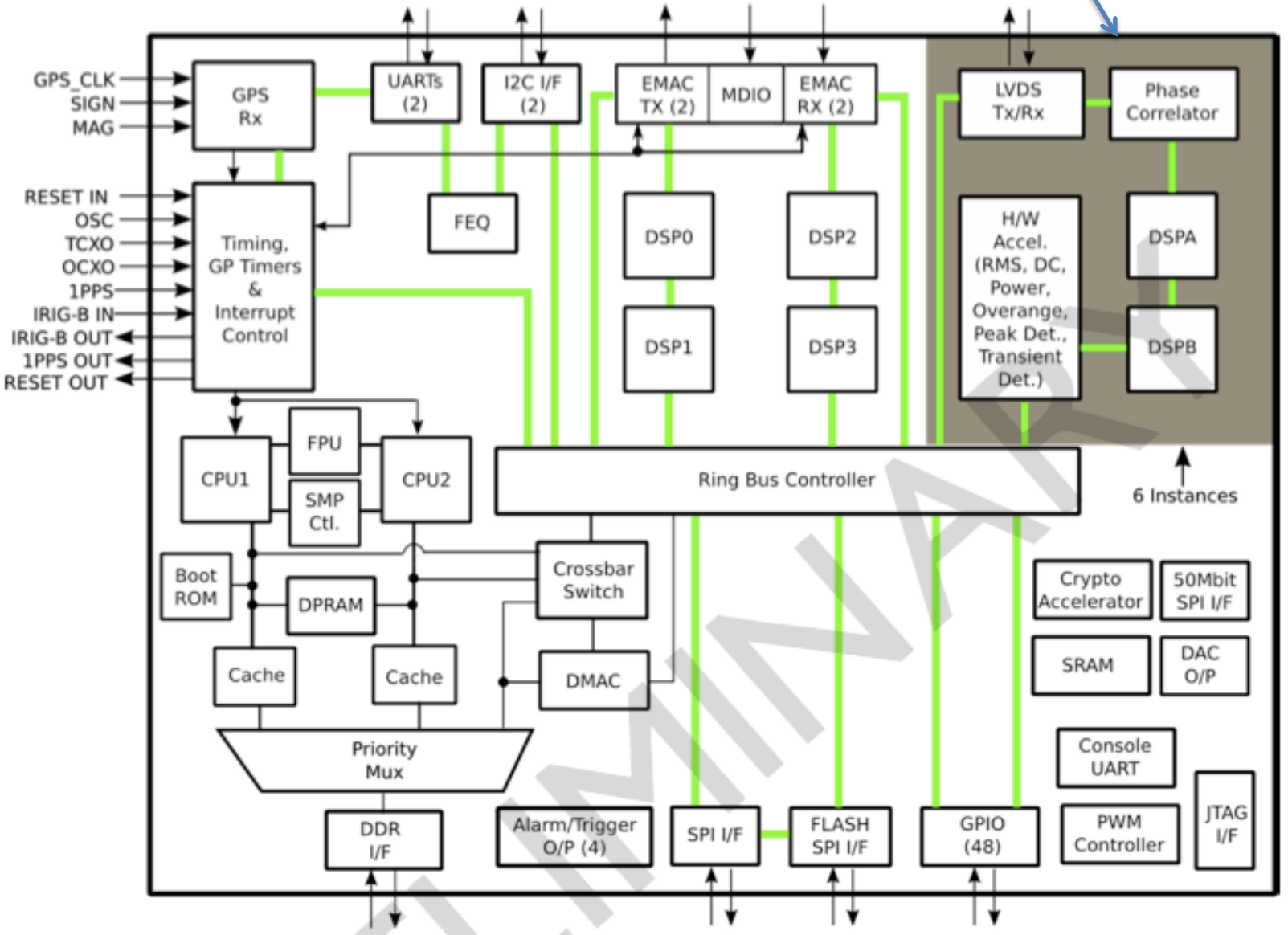
J2+ SoC with 16DSPs and Energy Hardware Measurement Accelerators



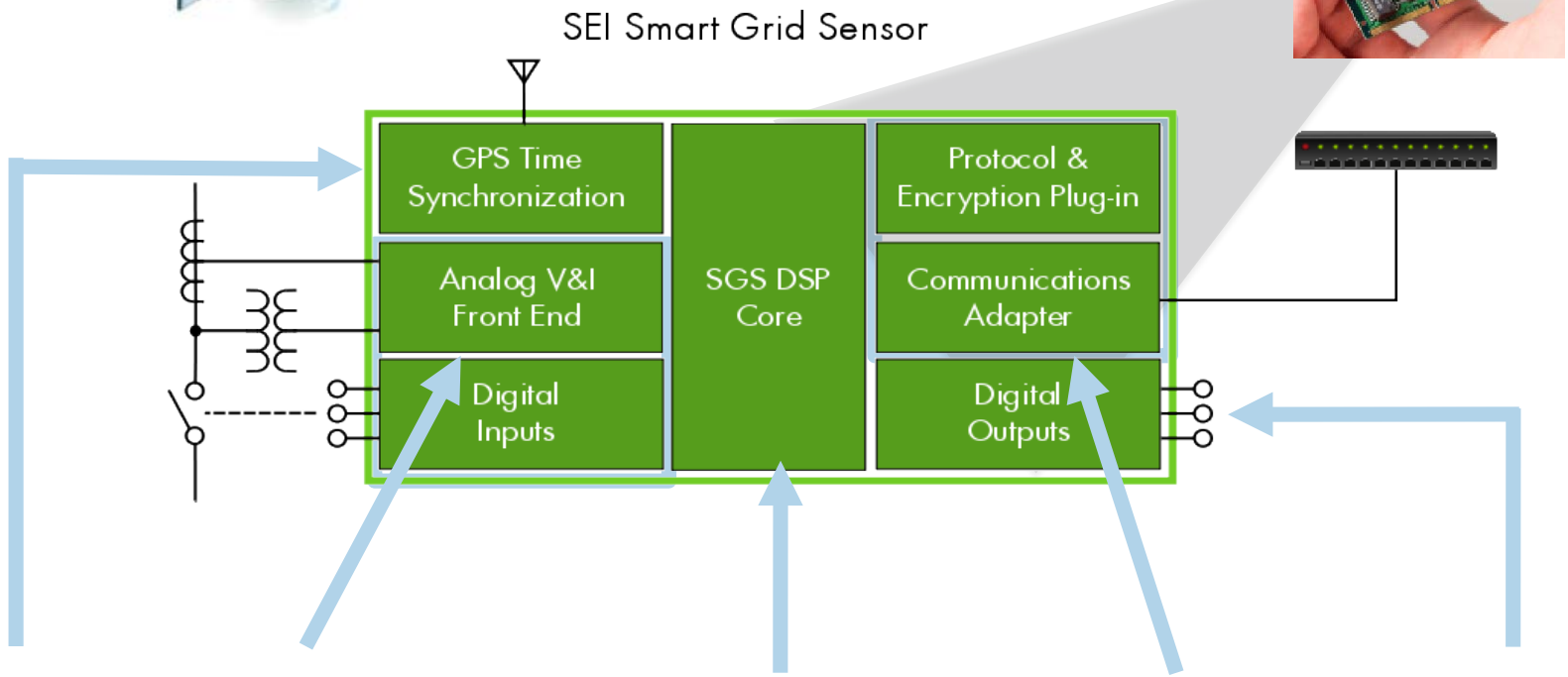
First Real World Device!

SGS1010D

J2+ SoC with 16DSPs and Energy Hardware Measurement Accelerators



Application : A Measurement & Communications Core Platform for Power Grid Devices



Timing – Measurement – Computation – Communication – Control

Software

What Does Software Stack Look Like?

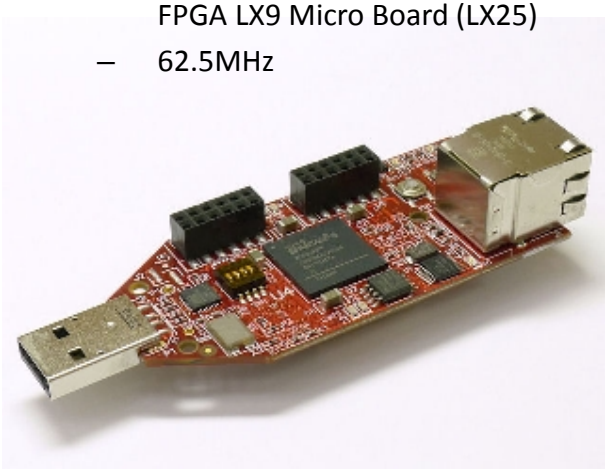
Standard Linux Environment

- Toolchains
  - GCC
  - BinUtils
  - ELF2FLT
  - GDB
- Kernel
  - CPU Specific Patches
  - Board Support
    - Memory, Interrupt Ctrl
    - Drivers (Serial, Eth, SD)
- Userspace
  - uClibc (Moving to MUSL)
  - Toybox + uClinux Dist
  - Initramfs (SD for Config)

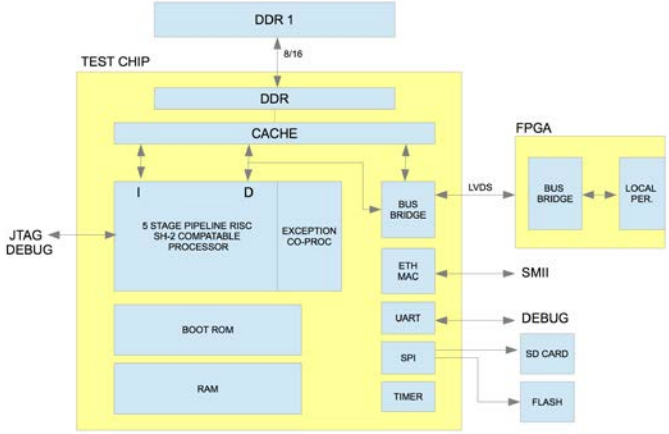
Demo

J2 / S2 Evaluation Boards

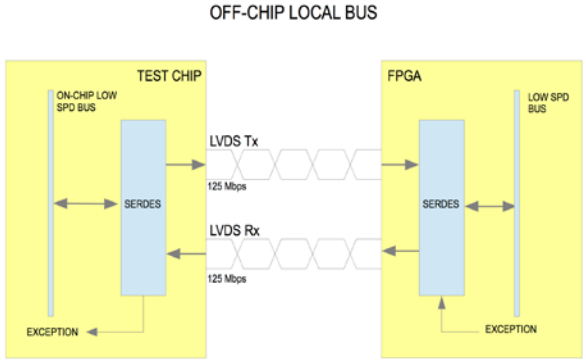
- FPGA Development System
  - VHDL on ISE® Design Suite
  - Xilinx AES-S6MB-LX6-G Spartan FPGA LX9 Micro Board (LX25)
  - 62.5MHz



- CPU (~45K Gates per Core)
  - 5 Stage RISC pipeline
  - Full Harvard (separate I and D)
  - 2 Processor SMP Configuration



- DSP Array
  - 4 Operation SIMD DSP
  - P / X / Y Memories
  - e.g. 16 DSPs on a Single SoC
- Memory Subsystem
  - 16b LPDDR @200MHz: PLL off mode, low EMI
  - On Chip Boot ROM
  - 0 wait state scratch pad SRAM (64KB)
- Full chip RTL / C co-simulation
  - JTAG Co-simulation and Debug



- Off-Chip Local Bus Access
  - 8B/10B High Speed Serial LVDS Bus
  - 4 pins : 12.5 or 25MByte/sec
  - Interrupts and exceptions also on the same bus
  - Framed protocol, simple (with example)
  - Purpose: Allow test chip to be used w/FPGA
  - Prototyping with FPGA of full speed peripherals
  - Highly flexible, low technical barrier for RTL design
  - Both bus master and slave

New Resource Sites (mid 2015)

- www.NoMMU.org
  - Development HOWTO
    - Platform information
      - Cortex-M, Armv7-j, J2, coldfire, blackfin…
    - Application development
      - Fixed stack, fork vs vfork, elf vs binflat/fdpic, memory fragmentation,
      - Toolchains and test environments
    - System development
      - Existing Linux root filesystem packages
      - QEMU coldfire emulation test platform for package dev
    - Education
      - Tutorials, mailing list
  - Upstream staging
    - Kernel, llvm, musl
    - Buildroot, openembedded

New Resource Sites (mid 2015)

- www.0pf.org
  - VHDL
    - Git repository with full history, under BSD license
    - J2 processor, S1 DSP, SOC with peripherals, makefiles
  - Bitstreams
    - Release binaries built from VHDL for lx9, lx25, and lx45
      - Lx9 entry level FPGA board, <$100 but only space for basic J2
      - Lx25 midrange, space for icache/dcache and ethernet
      - Lx45 space for multiple DSPs
  - Documentation
    - Toolchain install, VHDL howto, community mailing list