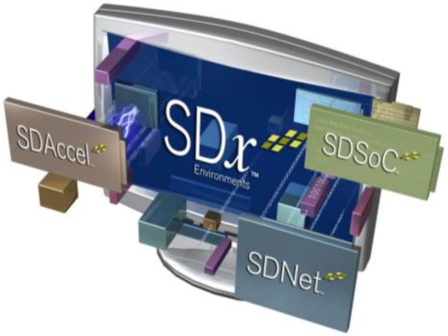
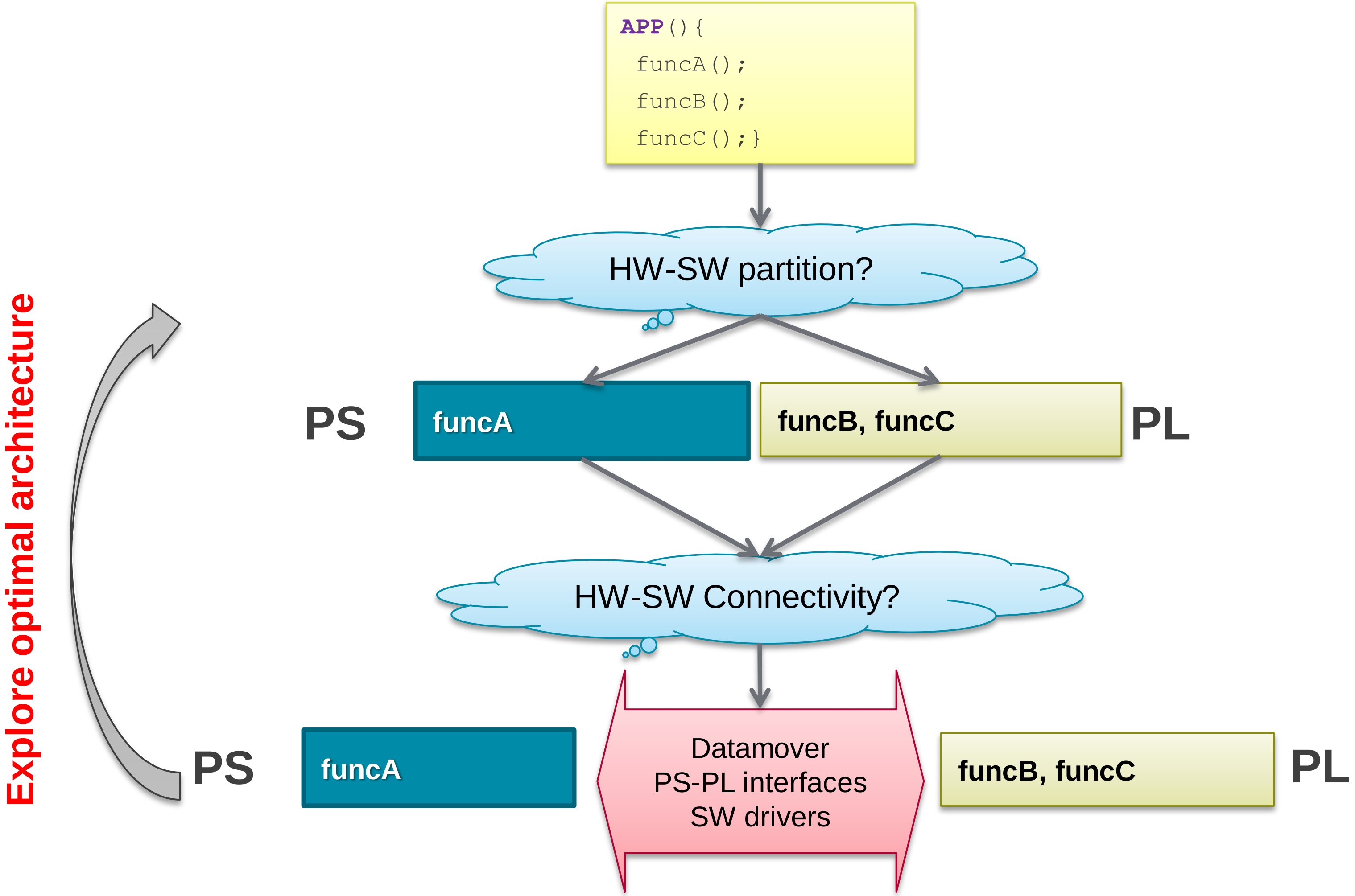
XILINX

ALL PROGRAMMABLE™

SDSoC Technical Introduction



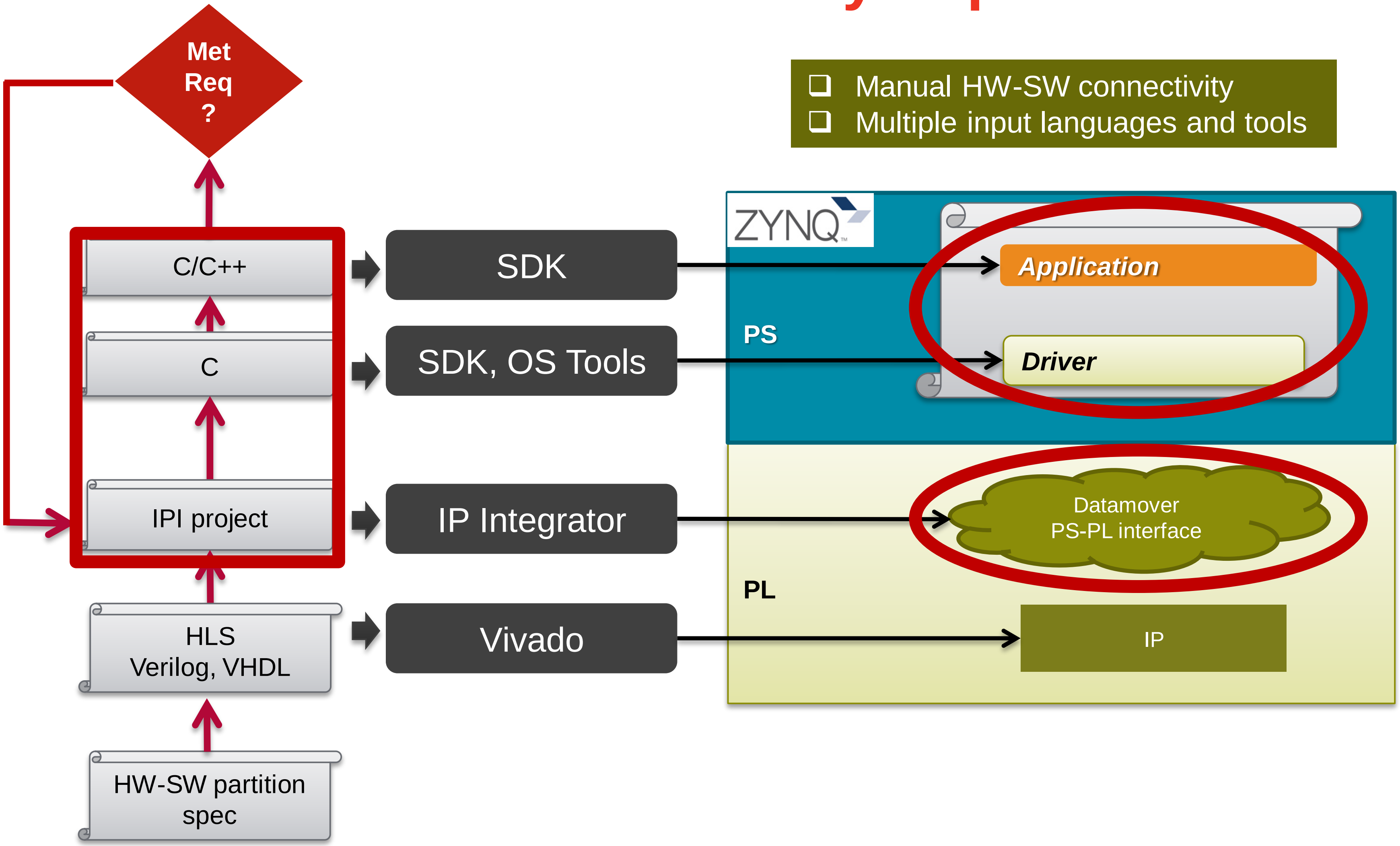
All Programmable SoCs Development Flow



```
APP(){
  funcA();
  funcB();
  funcC();}
```

Before SDSoC: Connectivity Exploration

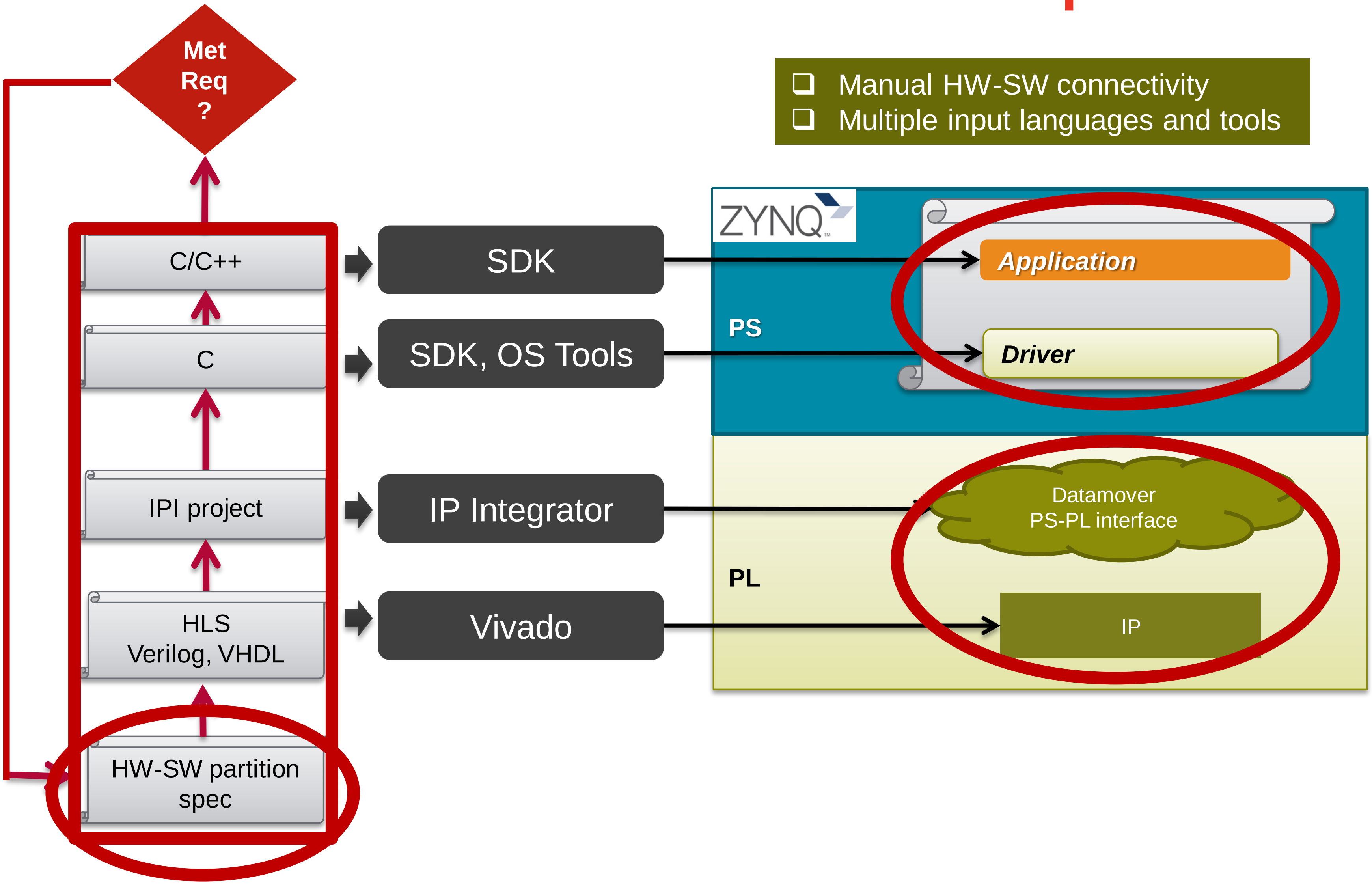
- ❑ Manual HW-SW connectivity
- ❑ Multiple input languages and tools



Need to modify multiple levels of design entry

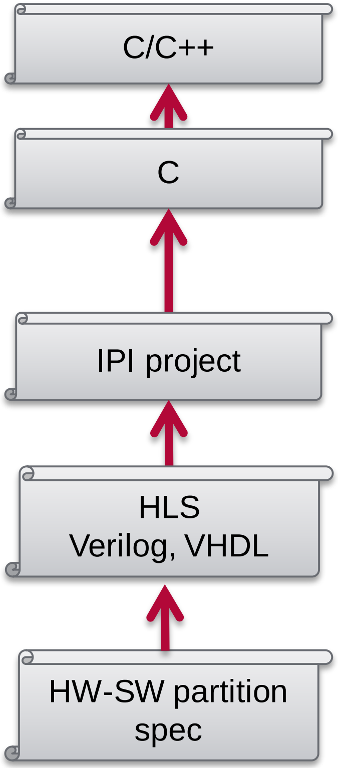
Before SDSoC: HW-SW Partition Exploration

- ❑ Manual HW-SW connectivity
- ❑ Multiple input languages and tools

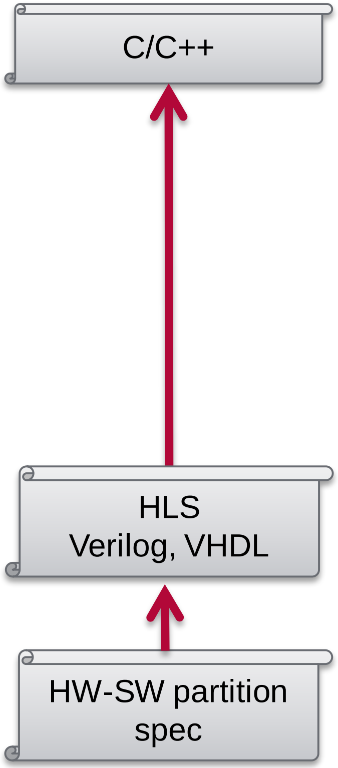


Involving multiple discipline to explore architecture

After SDSoC:

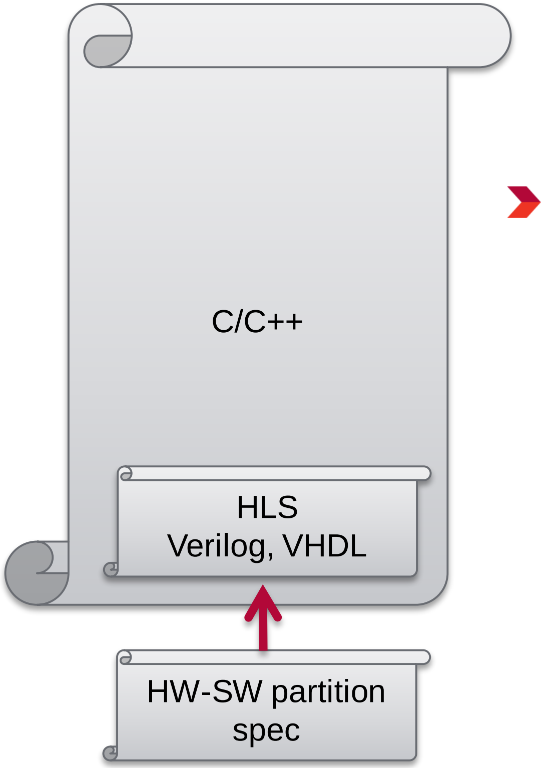


After SDSoC:



- Remove the manual design of SW drivers and HW connectivity

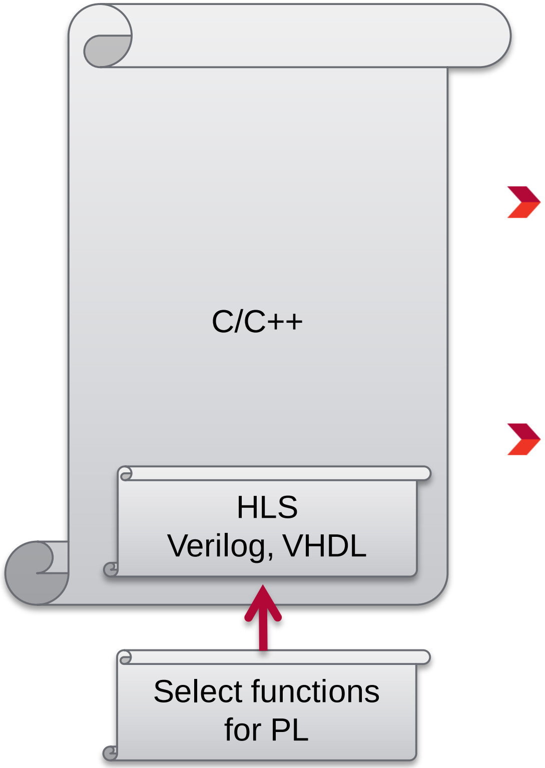
After SDSoC:



- Remove the manual design of SW drivers and HW connectivity

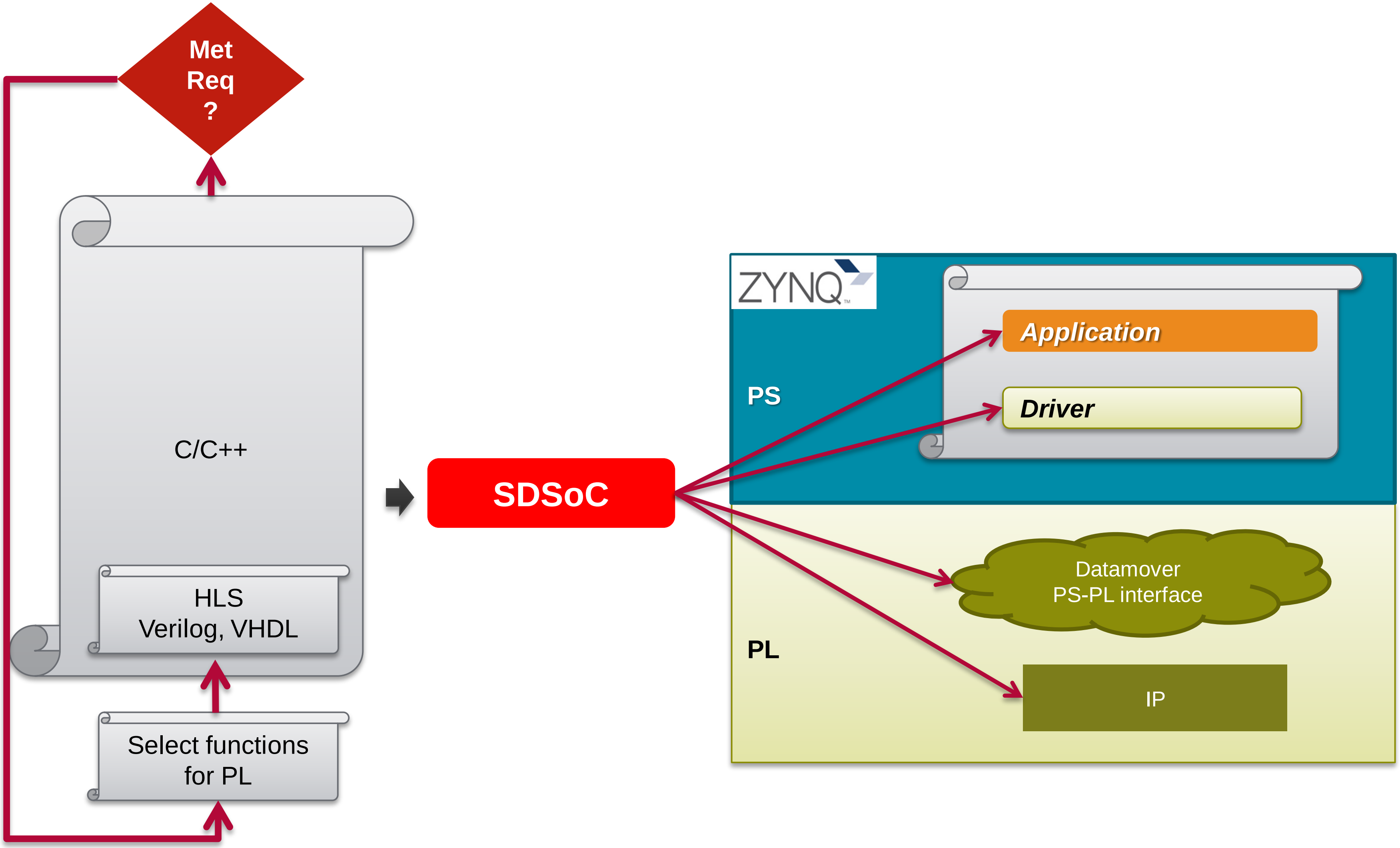
- Use the C/C++ end application as the input calling the user algorithm IPs as function calls

After SDSoC:



- Remove the manual design of SW drivers and HW connectivity
- Use the C/C++ end application as the input calling the user algorithm IPs as function calls
- Partition set of functions to Programmable Logic by a single click

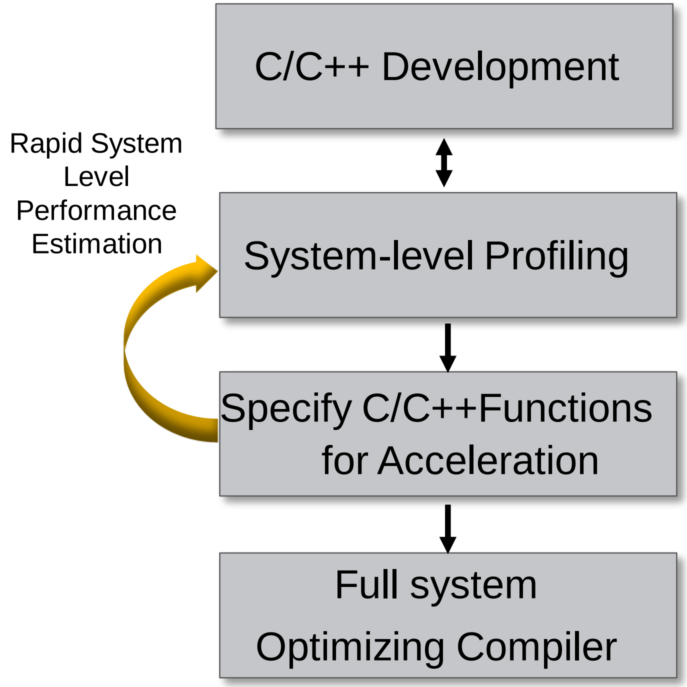
After SDSoC: Automatic System Generation



C/C++ to System in hour, days

The SDSoC Development Environment

SDSoC™
Environment


C/C++ Development

System-level Profiling

Specify C/C++Functions for Acceleration

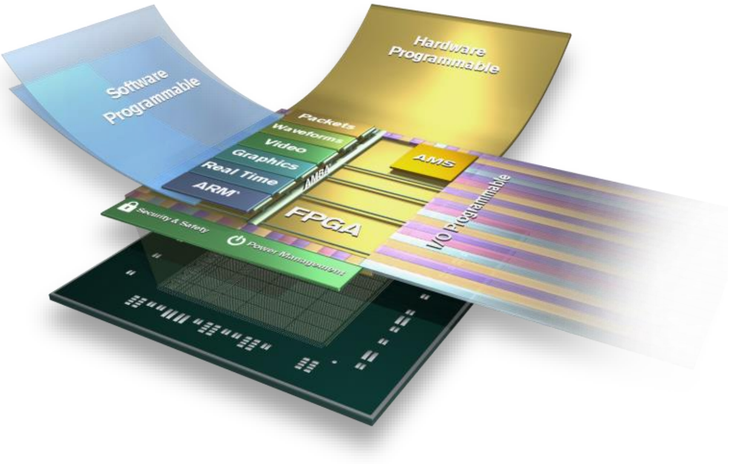
Full system Optimizing Compiler

Rapid System Level Performance Estimation

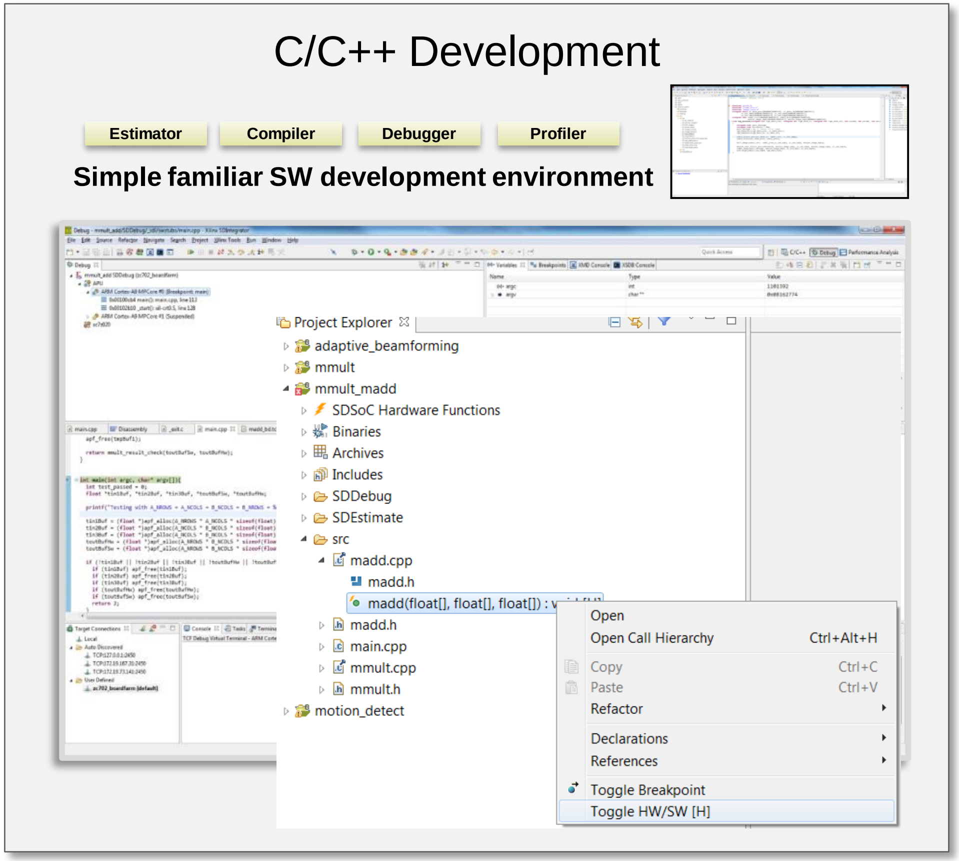
ZYNQ® SoC

ZYNQ® MPSoC

- ASSP-like programming experience
- System-level profiling
- Full system optimizing compiler
- Expert use model for platform developers and system architects



SDSoC's ASSP-like Programming Experience

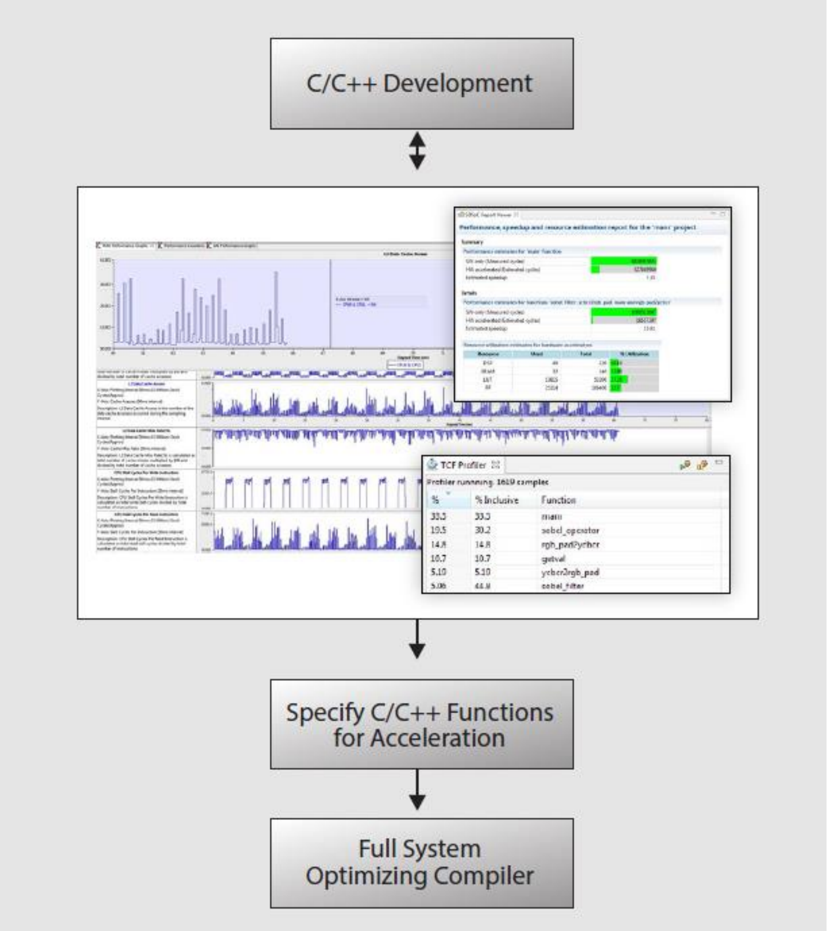


C/C++ Development

Simple familiar SW development environment

- **Easy to use Eclipse IDE**
- **One click to accelerate functions in Programmable Logic (PL)**
- **Optimized libraries**
  - Xilinx, ARM and Partners
  - DSP, Video, fixed point, linear algebra, BLAS, OpenCV
- **Support for Linux, FreeRTOS, bare metal**
  - Additional OS support in future releases

SDSoC’s System Level Profiling


C/C++ Development
Specify C/C++ Functions for Acceleration
Full System Optimizing Compiler
SDSoC Report Viewer
TCF Profiler

Performance, speedup and resource estimation report for the 'main' project

Summary

Performance estimates for 'main' function

- SW-only (Measured cycles)
- HW accelerated (Estimated cycles)
- Estimated speedup

Details

- SW-only (Measured cycles)
- HW accelerated (Estimated cycles)
- Estimated speedup

Resource utilization estimates for hardware accelerators

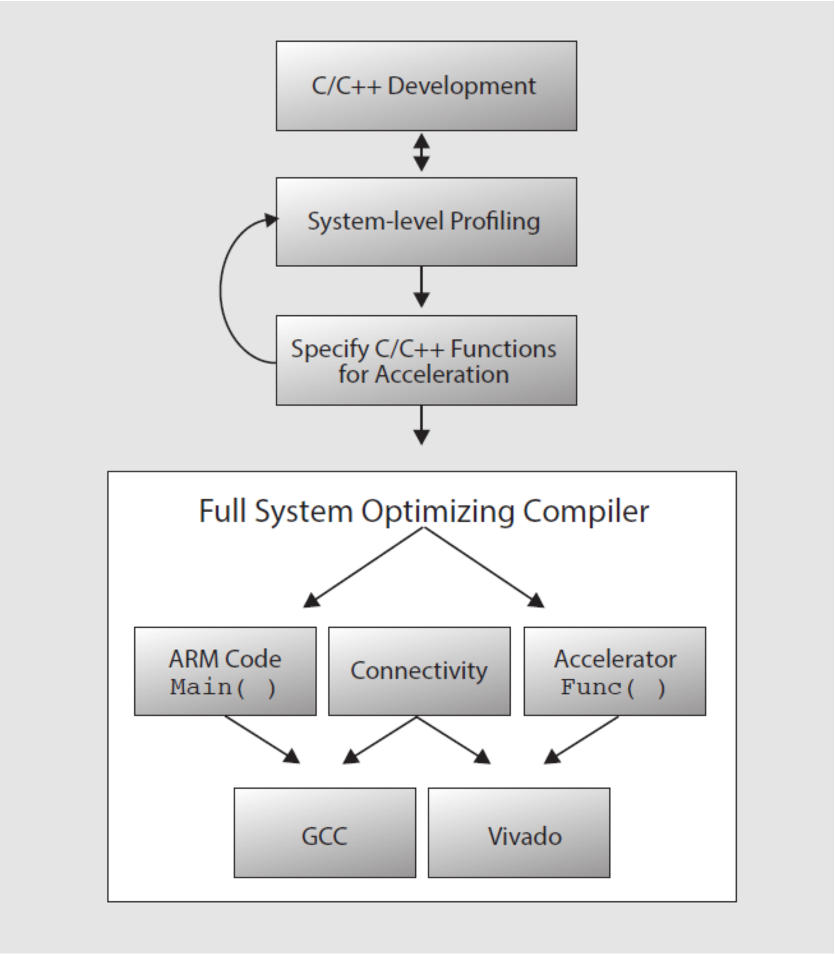
Resource	Used	Total	% Utilization
DSP	[illegible]	[illegible]	[illegible]
BRAM	[illegible]	[illegible]	[illegible]
LUT	13825	53200	[illegible]
FF	15504	106400	[illegible]

Profiler running. 1619 samples

%	% Inclusive	Function
33.3	33.3	main
19.5	30.2	sobel_operator
14.8	14.8	rgb_pad2ycbcr
10.7	10.7	getval
5.10	5.10	ycbcr2rgb_pad
5.06	44.8	sobel_filter

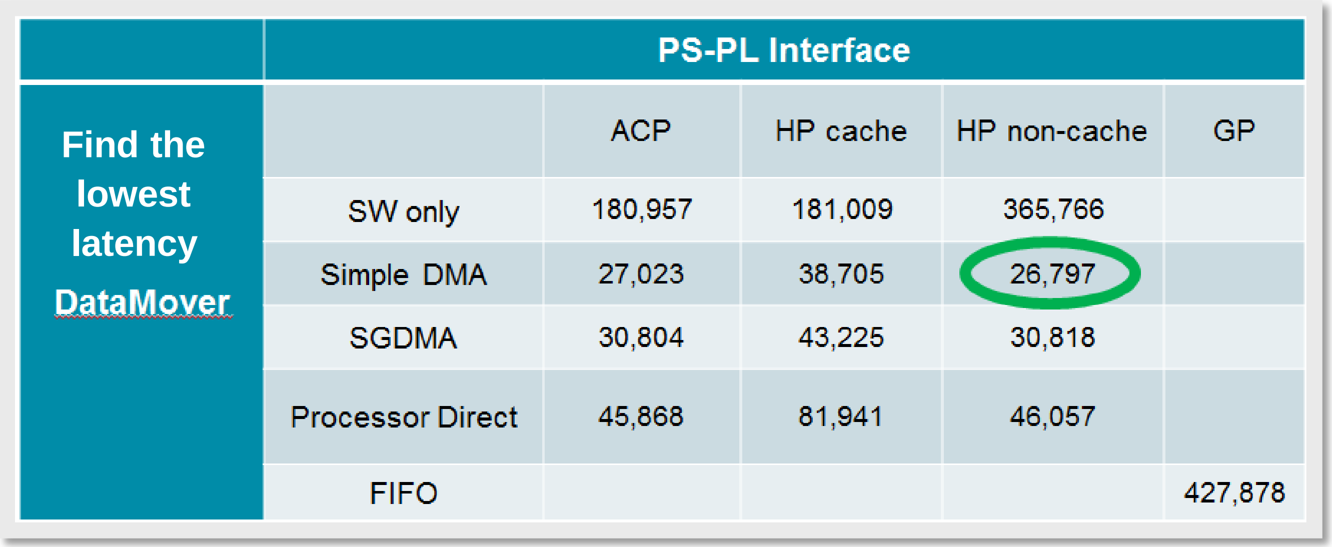
- **Rapid system performance estimation**
  - Full system estimation (programmable logic, data communication, processing system)
  - Reports SW/HW cycle level performance and hardware utilization
- **Automated performance measurement**
  - Runtime measurement by instrumentation of cache, memory, and bus utilization

Full System Optimizing Compiler



- **Rapid software configurable application acceleration using C/C++**
  - Automated function acceleration in programmable logic
  - Up to 100X increase in performance vs. software
  - System optimized for latency, bandwidth, and hardware utilization

Automated Generation of System Connectivity

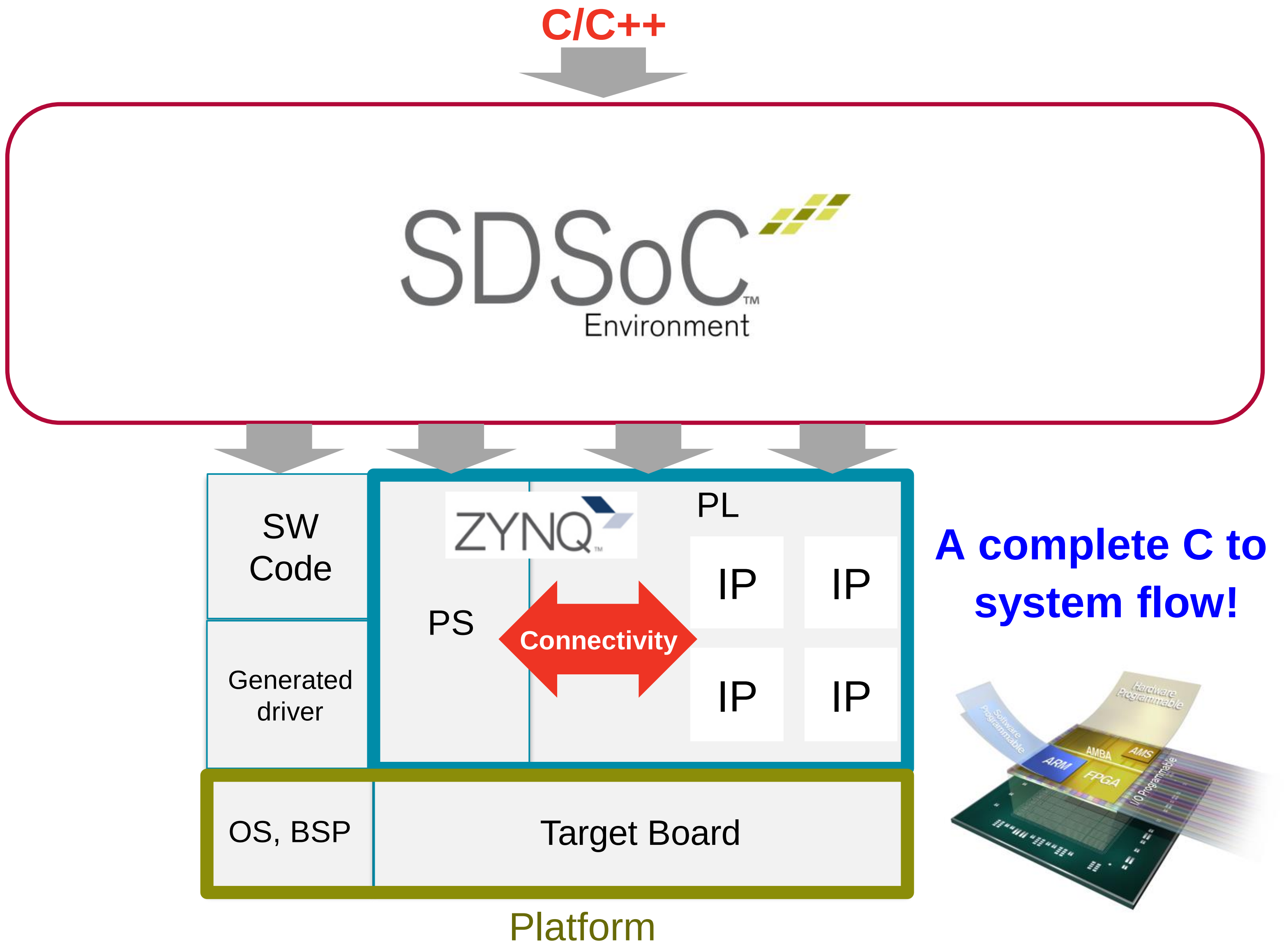


	PS-PL Interface				
Find the lowest latency DataMover		ACP	HP cache	HP non-cache	GP
	SW only	180,957	181,009	365,766	
	Simple DMA	27,023	38,705	26,797	
	SGDMA	30,804	43,225	30,818	
	Processor Direct	45,868	81,941	46,057	
	FIFO				427,878

32X32 floating point matrix multiply (latency in processor cycles)

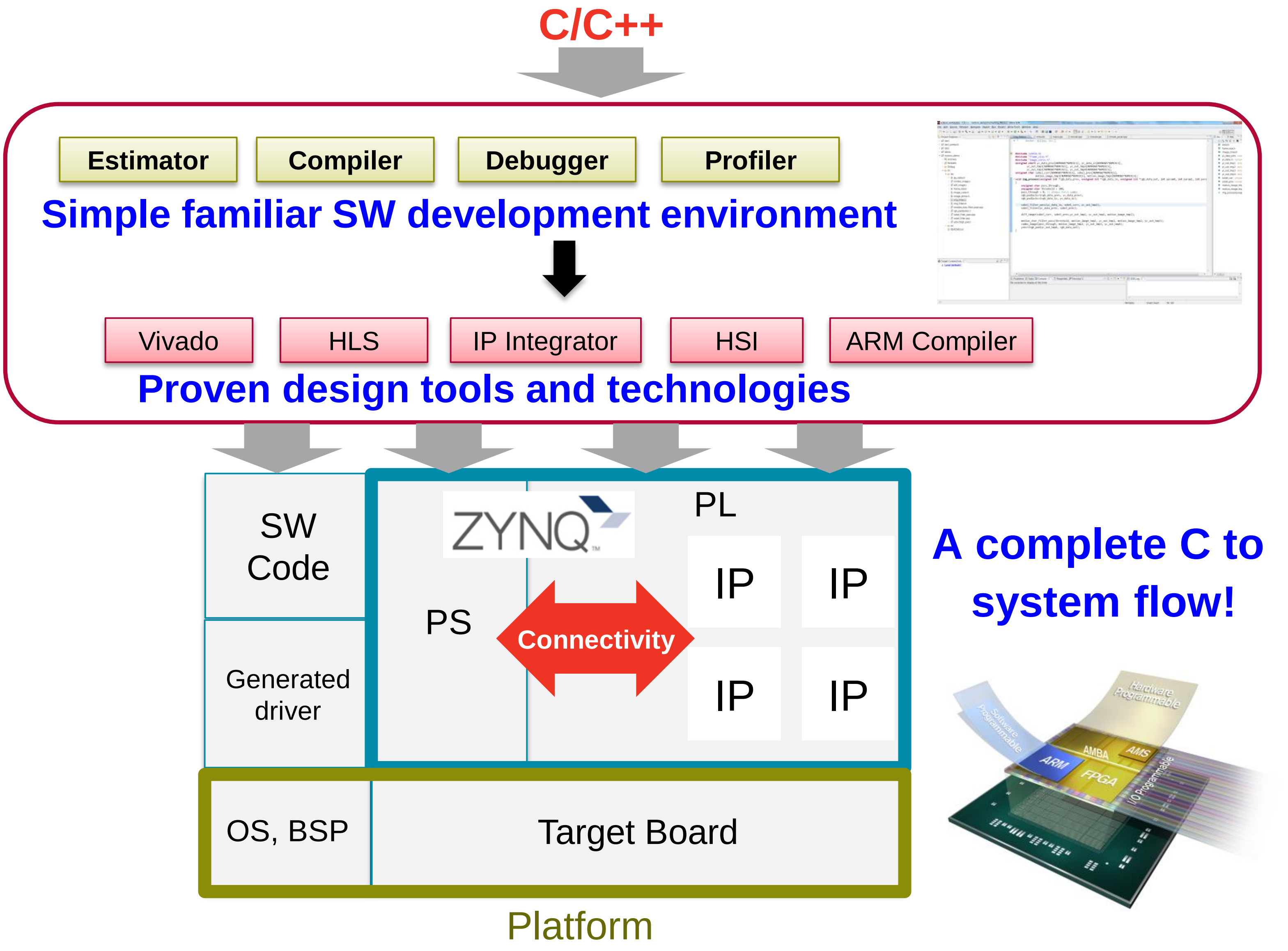
- **Explore system performance and power**
  - Rapidly configure, generate macro and micro architectures
  - Explore optimal interconnect and memory interfaces
  - Automatic insertion of AXI-Performance Monitor (APM) to obtain detailed cache/port/memory performance data
- **Shorten development time over traditional Hardware/Software flows**

SDSoC: Complete End-to-End Flow



A complete C to system flow!

SDSoC: Complete End-to-End Flow


C/C++

Estimator | Compiler | Debugger | Profiler

Simple familiar SW development environment

Vivado | HLS | IP Integrator | HSI | ARM Compiler

Proven design tools and technologies

Platform

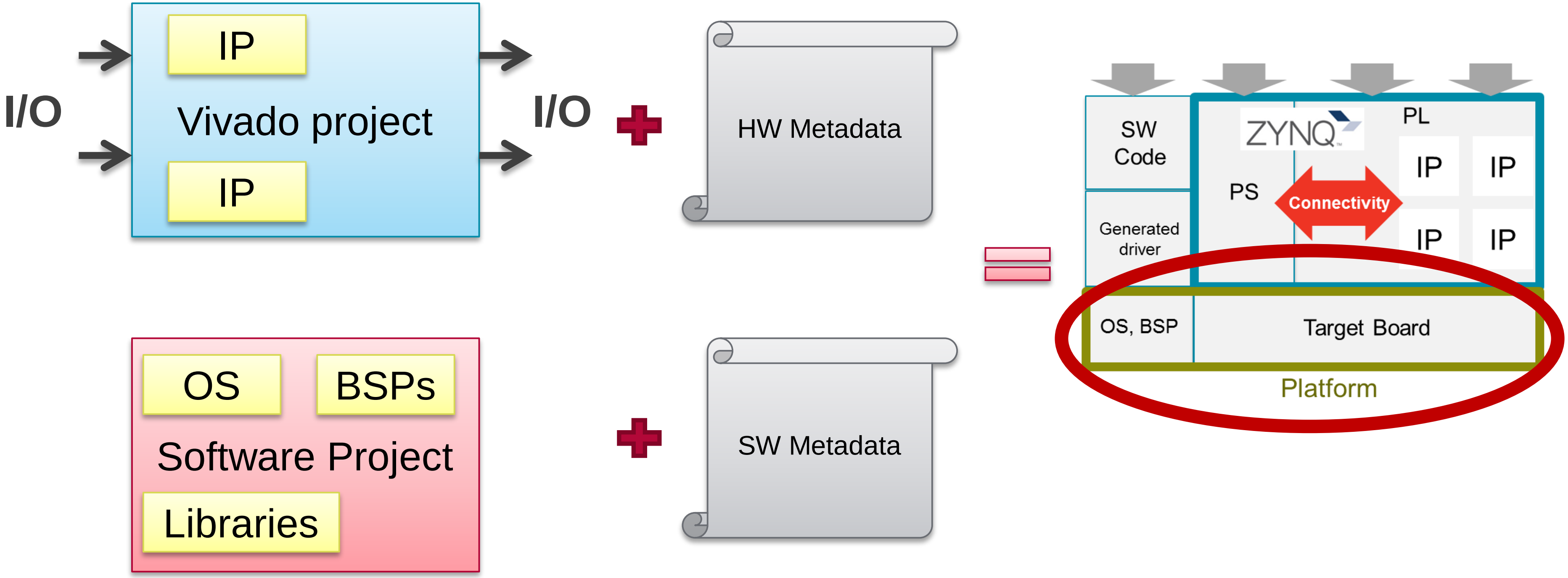
A complete C to system flow!

Supported Development Platforms

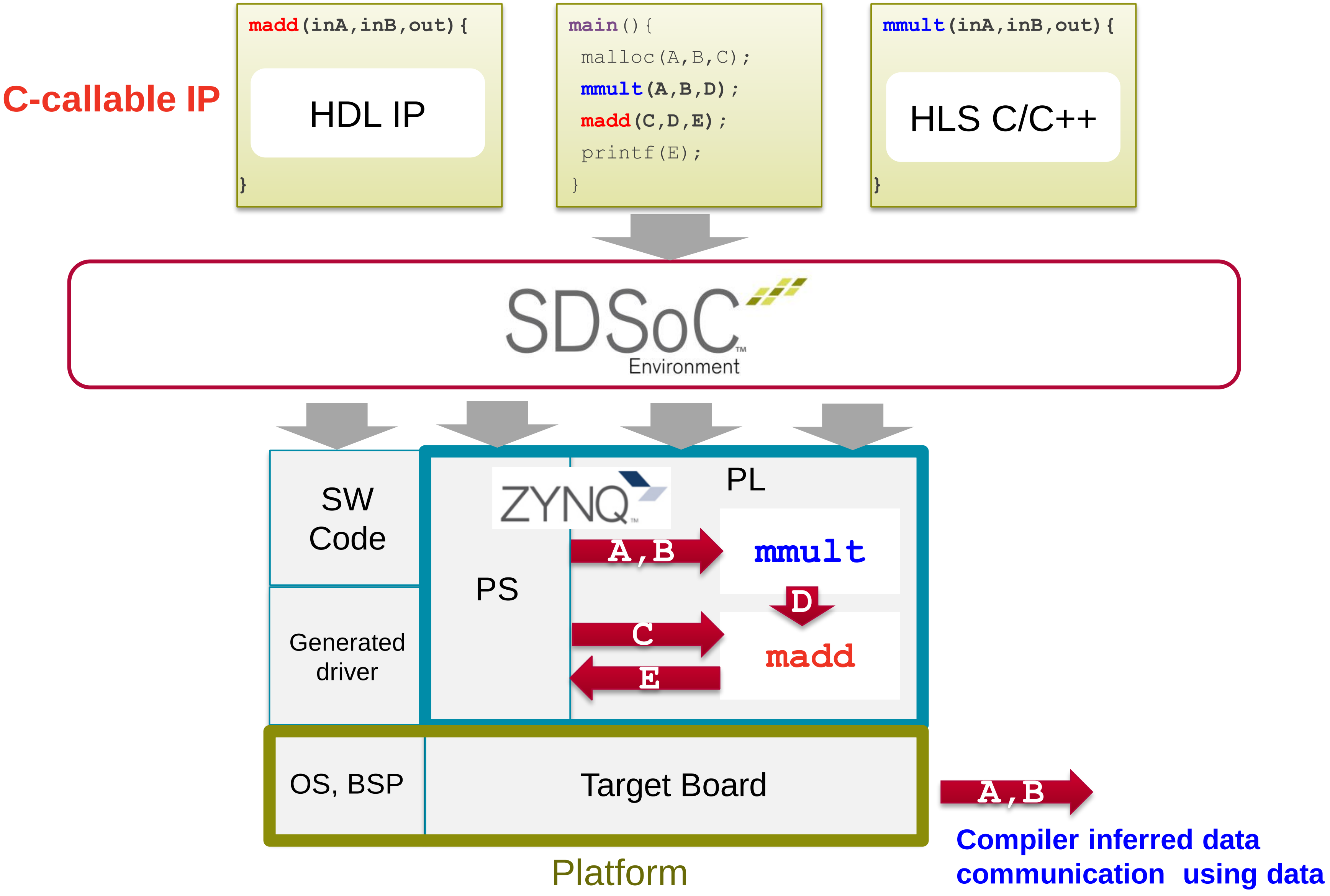
- **Start today with:**
  - Xilinx development platforms: ZC702 & ZC706,
  - Alliance Member platforms: Zedboard, MicroZed, ZYBO, Zynq SDR, ZC702+HDMI IO, SVDK, any many more
  - Visit http://www.xilinx.com/products/design-tools/software-zone/sdsoc.html#boardskits
- Customer platform flow supported today (UG1146)

Custom Platform Creation

- Target customer's own board for production design with SDSoC
- Add metadata to existing Vivado project and existing Software projects (OS, BSPs and libraries)



An Example – Matrix Multiply + Add



C-callable IP

```
madd(inA,inB,out){
    HDL IP
}
```

```
main(){
 malloc(A,B,C);
 mmult(A,B,D);
 madd(C,D,E);
 printf(E);
}
```

```
mmult(inA,inB,out){
    HLS C/C++
}
```

SDSoC™ Environment

SW Code

Generated driver

ZYNQ

PS

PL

A,B

mmult

D

C

madd

E

OS, BSP

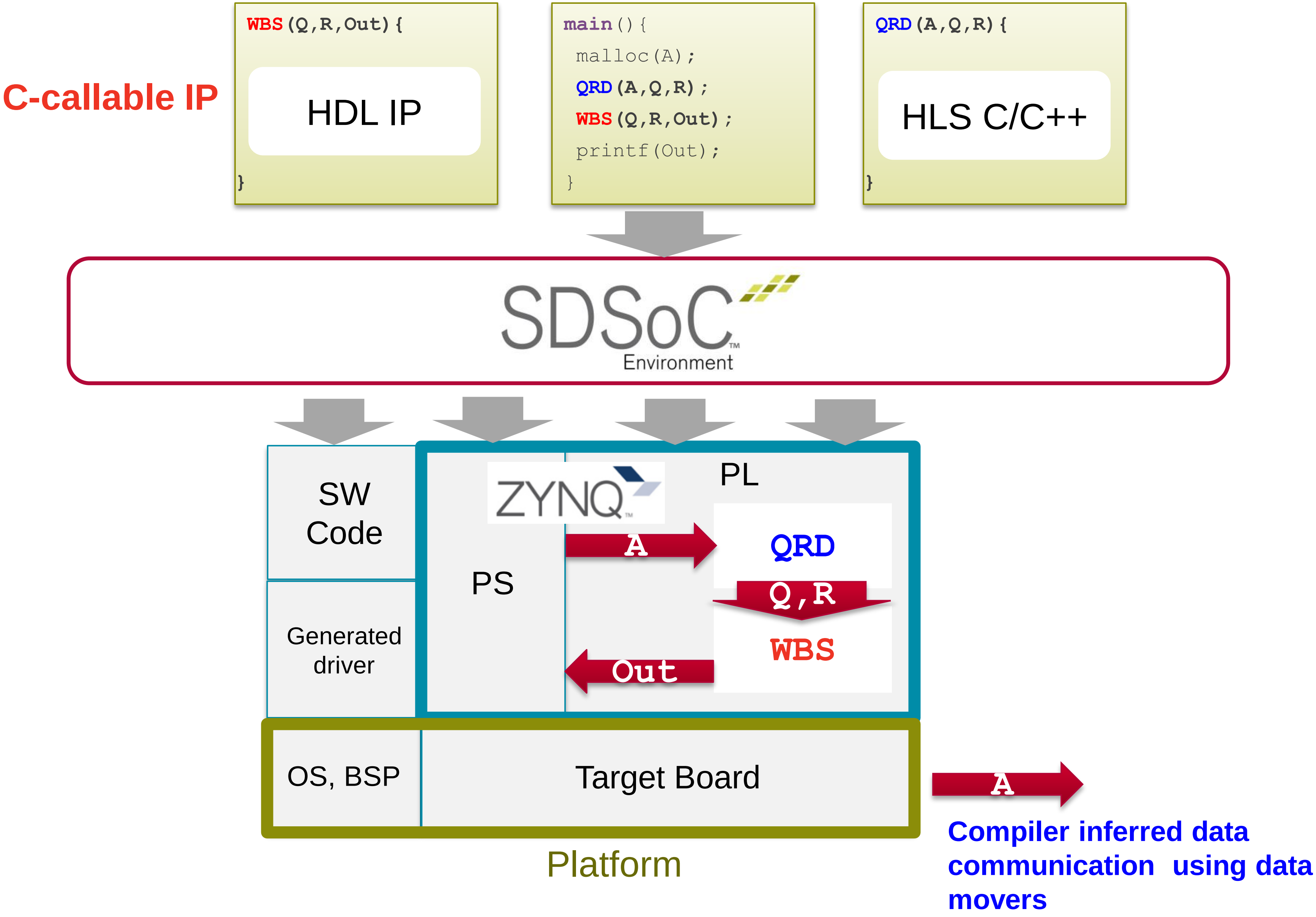
Target Board

Platform

A,B

Compiler inferred data communication using data movers

An Example - Beamforming



C-callable IP

```
WBS(Q,R,Out){
    HDL IP
}
```

```
main(){
 malloc(A);
 QRD(A,Q,R);
 WBS(Q,R,Out);
 printf(Out);
}
```

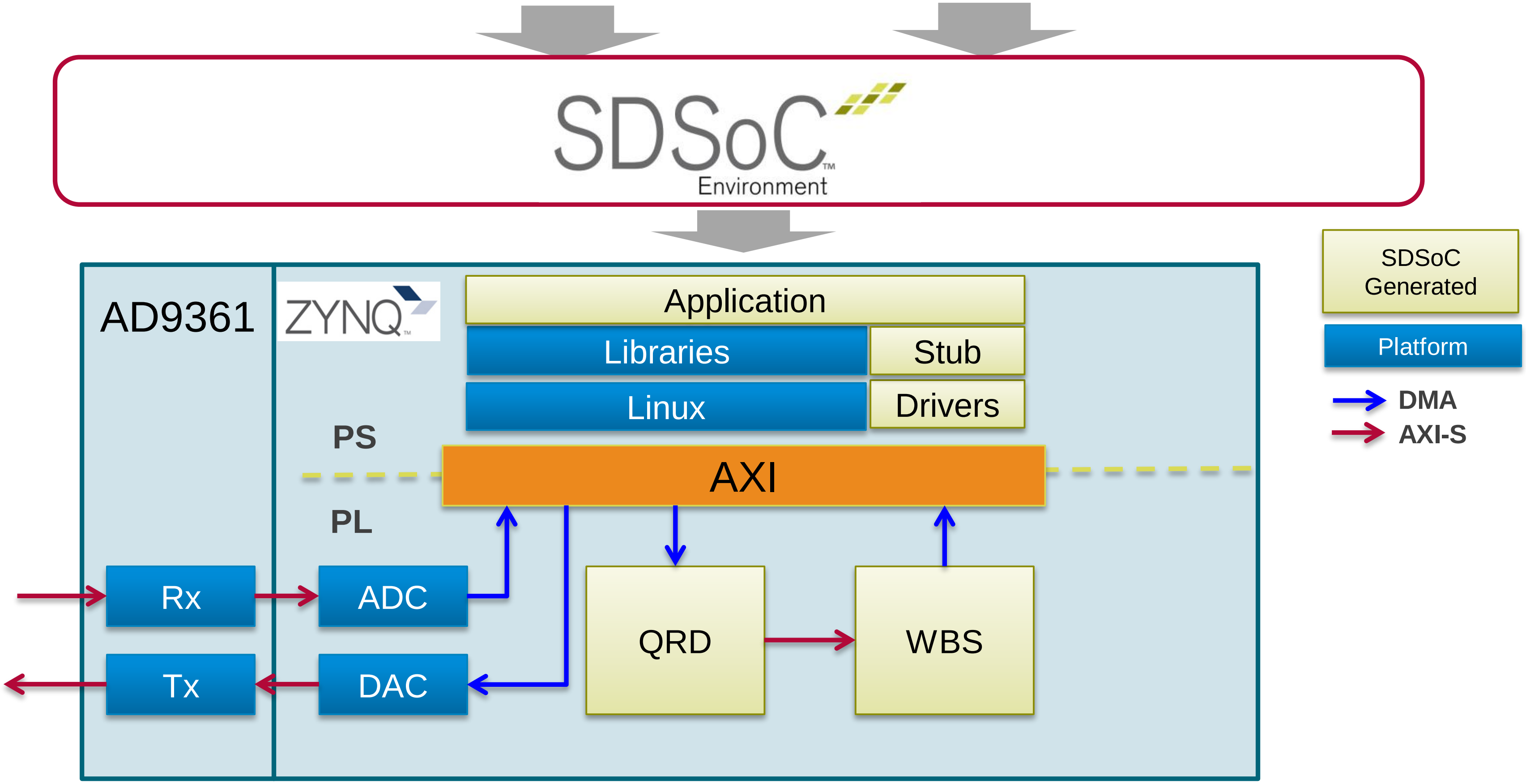
```
QRD(A,Q,R){
    HLS C/C++
}
```

Compiler inferred data communication using data movers

Beamforming with AD9361

```
main(){
  rxADC(A);
  QRD(A,Q,R);
  WBS(Q,R,out);
  txDAC(out);
}
```

Zynq SDR platform

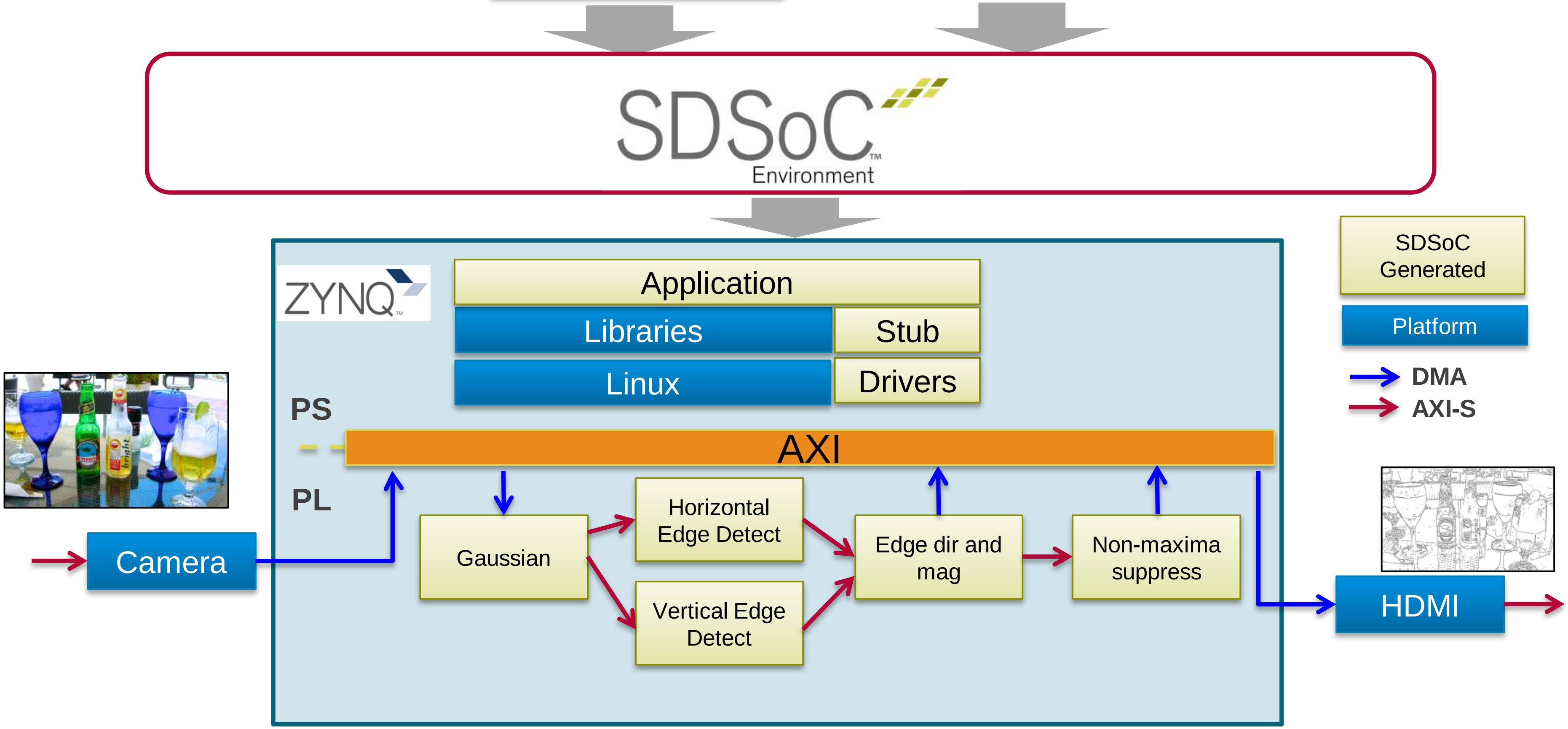


Custom Platform and HLS optimized IPs

Canny Edge Detection with SDSoC

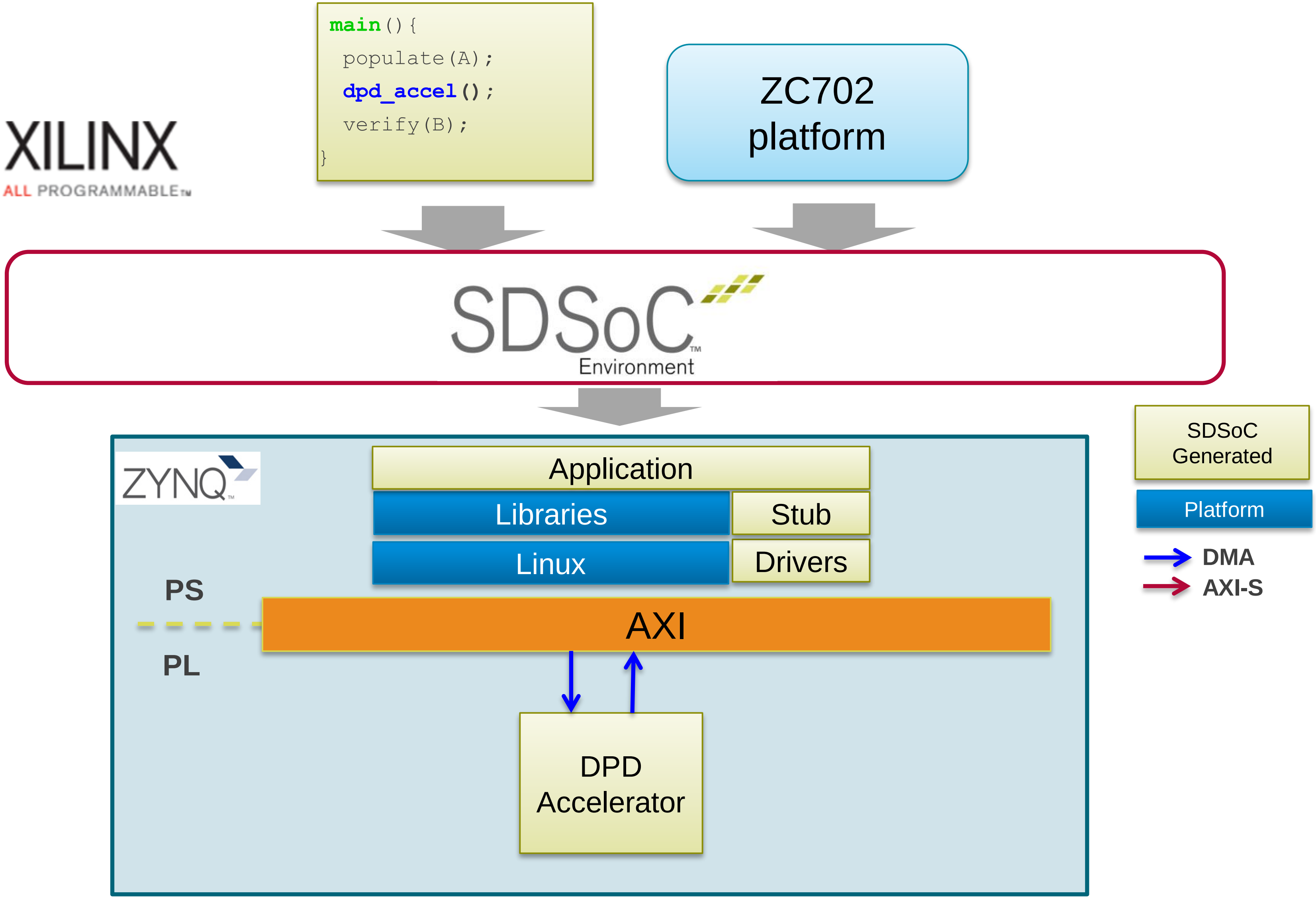
```
main(){
 getCamera(A);
 gaussian(A,B);
 horEdge (B,C);
 verEdge (B,D);
 …
 display(out);}
```

iVeia Vision Platform



Custom Platform C-callable IPs and HLS optimized IPs

Digital Pre-Distortion (DPD) Accelerator



```
main(){
 populate(A);
 dpd_accel();
 verify(B);
}
```

ZC702 platform

SDSoC Environment

ZYNQ

Application

Libraries | Stub

Linux | Drivers

PS

PL

AXI

DPD Accelerator

SDSoC Generated

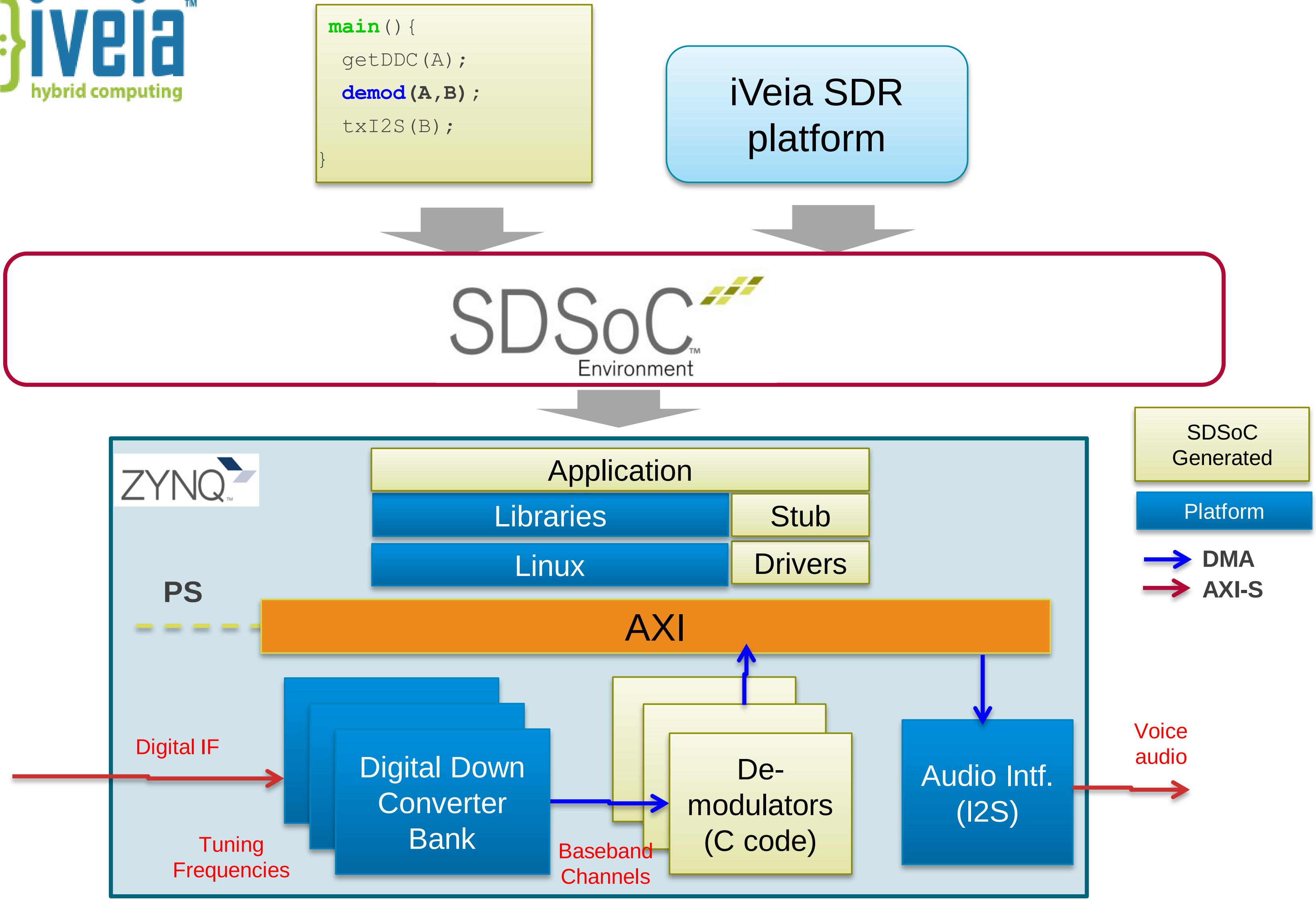
Platform

→ DMA

→ AXI-S

Custom Platform and C-callable RTL IP

Demodulation



```
main(){
  getDDC(A);
  demod(A,B);
  txI2S(B);
}
```

Custom Platform and C-callable RTL IP

1080p60 Motion Detection using SDSoC

SDSoC Generated

Platform

DMA

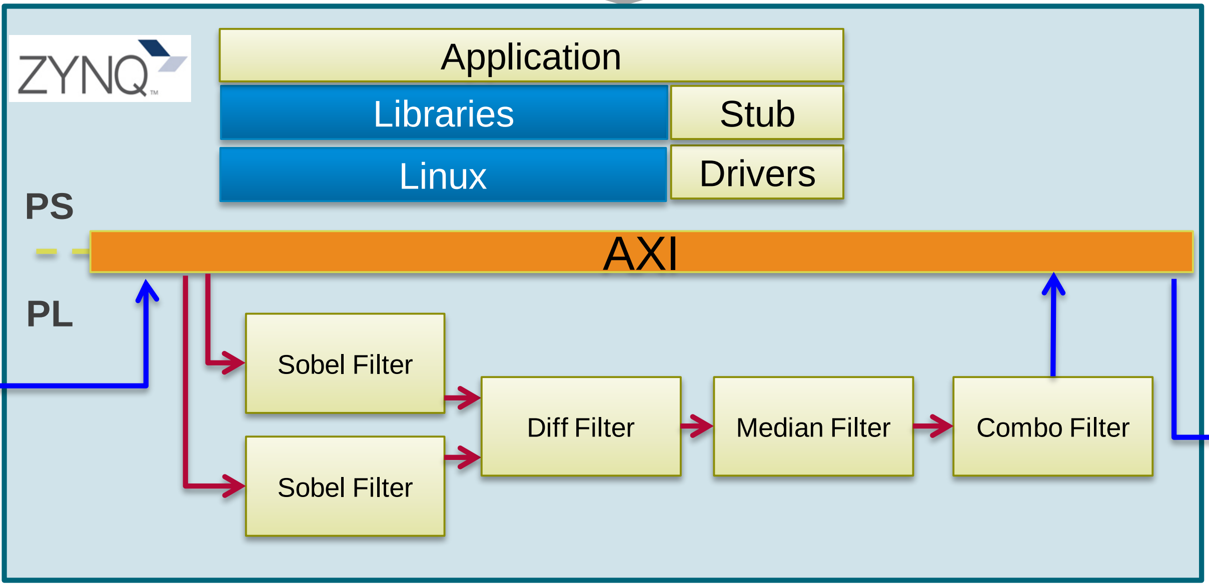
AXI-S

```
main(){
  get_camera(A);
  sobel(A,B);
  diff(B,C);
  …
  display(out);
}
```

ZC702 + HDMI FMC Platform

SDSoC Environment

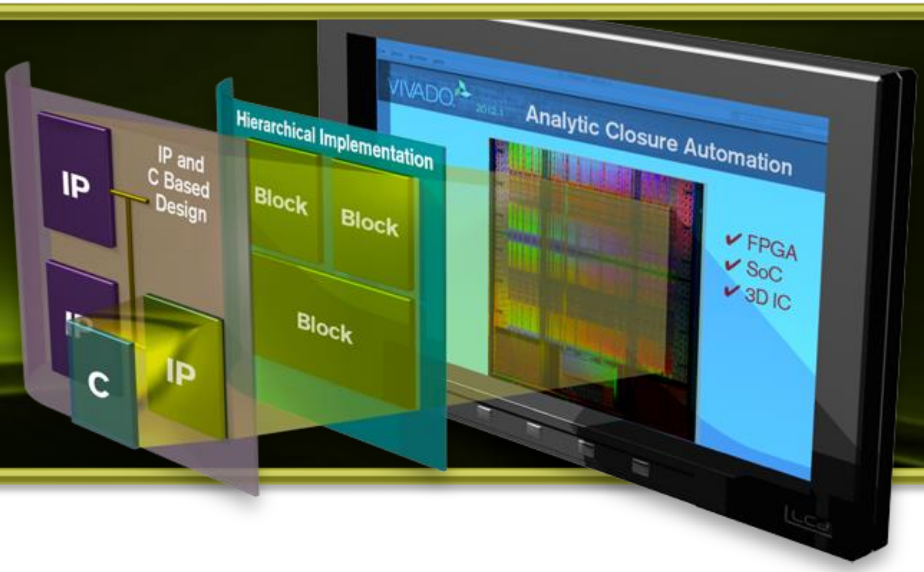
HDMI



HDMI

Entire design completed in 2-weeks with multiple iterations of algorithm

Thank you


IP
IP and C Based Design
IP
C
IP
Hierarchical Implementation
Block
Block
Block
VIVADO
Analytic Closure Automation
✓ FPGA
✓ SoC
✓ 3D IC