Qsys (Platform Designer) and IP Core Integration

Stephen A. Edwards
(after David Lariviere)

Columbia University

Spring 2023
IP Cores

Altera’s IP Core Integration Tools

Connecting IP Cores
IP Cores
Cyclone V SoC: A Mix of Hard and Soft IP Cores

IP = Intellectual Property
Hard = wires & transistors

Core = block, design, circuit, etc.
Soft = implemented w/ FPGA
Example IP Cores

**CPUs:** ARM (hard), NIOS-II (soft)

**Highspeed I/O:** Hard IP Blocks for High Speed Transceivers (PCI Express, 10Gb Ethernet)

**Memory Controllers:** DDR3

**Clock and Reset signal generation:** PLLs
Cyclone V SoC: FPGA layout

6.144-Gbps Transceivers
ALMs and Distributed Memory
PLLs
6.144-Gbps Transceivers PCS
Hard IP Blocks for PCIe Gen 2 and PCIe Gen 1
External Memory Interface Controllers
HPS I/O
ARM Cortex-A9 MPCore HPS
M10K Embedded Memory Blocks
Variable-Precision Digital Signal Processing (DSP) Hard IP Blocks
Up to 469 I/O pins (HPS + FPGA)
Two Core/Transceiver Power Regulators Required (1.1V, 2.5V)

Source: Altera
Cyclone V SoC: HPS Layout

Source: NARD, LLC.
Stratix V: FPGA Layout

Source: Altera
A bus bridge connects two, often different, buses.

Enables multiple clock domains, different protocols (e.g., AXI ↔ Avalon), bus widths, etc.

**Example Bridge Types:**

- SOC HPS ↔ FPGA Bridge
- Avalon MM Clock Crossing Bridge
- Avalon MM Pipeline Bridge
Cyclone V SoC: FPGA ↔ HPS Bridge

Source: Altera
Altera’s IP Core Integration Tools
The Quartus Megawizard

Source: Altera
Example:

10Gb Ethernet PHY
Megawizard IP Cores

Core-specific interfaces on each

**Arithmetic:** $+, -, \times, \div$, Multiply-Accumulate, ECC

**Floating Point:** $+, -, \times, \div$

**Gate Functions:** Shift Registers, Decoders, Multiplexers

**I/O Functions:** PLL, temp sensor, remote update, high speed transceivers

**Memory:** Single/Dual-port RAMs, Single/Dual-clock FIFOs, Shift registers

**DSP:** FFT, ECC, FIR, etc.

**Video:** large suite

Some megafuinctions are only available on certain FPGAs
Platform Designer (formerly Qsys) is Altera’s system integration tool for building Network-on-Chip (NoC) designs connecting multiple IP cores.

<table>
<thead>
<tr>
<th>You</th>
<th>Qsys</th>
</tr>
</thead>
<tbody>
<tr>
<td>List the IP components and how you want them connected</td>
<td>Generates the interconnect (arbiters, etc.), adds adapters as necessary, warns of errors</td>
</tr>
</tbody>
</table>
Qsys: Raising Level of Abstraction

Source: Altera
System Level Design: Why Use Qsys

Avoids manually developing custom interconnect fabrics and signaling.

Instead of cycle-to-cycle coordination between every individual IP core, focus on transaction-level designs.

Design IP without knowing exactly when data will transfer and instead only focus on how (once it does).

(Only valid if you design your individual components to one of the standardized interfaces)
Qsys-based Method of Design

Source: Altera
Connecting IP Cores
Interface Types

**Memory-mapped** Interfaces:

Avalon MM (Altera)

AXI (ARM, supported by Qsys now for SoC)

**Streaming** Interfaces: Avalon ST:

Avalon ST source port: outputs streaming data

Avalon ST sink port: receives incoming streaming data
Control vs. Data Planes

Control Plane: Memory mapped registers typically used for configuring devices, querying status, initiating transactions, etc (low bandwidth)

Data Plane: Streaming directed graphs for actually moving and processing large amounts of data (audio/video, network packets, etc); high bandwidth

A single IP core can have both MM and ST interfaces (including multiple of each).
Control and Data Planes Example
Additional References

Altera online training lectures: (HIGHLY recommended; many of these slides are taken directly from them)

http://www.altera.com/education/training/curriculum/trn-curriculum.html

Introduction to Qsys

Advanced System Design Using Qsys

Custom IP Development Using Avalon and AXI Interfaces

(Everything has moved to Intel; above link still works)