ELECTRONIC DESIGN PROCESS SYMPOSIUM
at Monterey Beach, California

End-to-End Automated Hardware/Software Co-Design for Reconfigurable SoC

Guy Bois[1,2], Laurent Moss[1],
Gary Dare[1], Hubert Guérard[1]

1. Space Codesign Systems
2. École Polytechnique de Montreal

Agenda

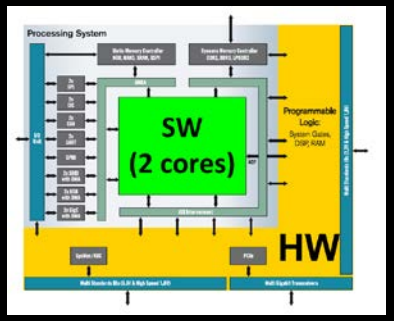
- Problem
- Our solution
- Details about the Flow
- Case Study
- Conclusion

Problem

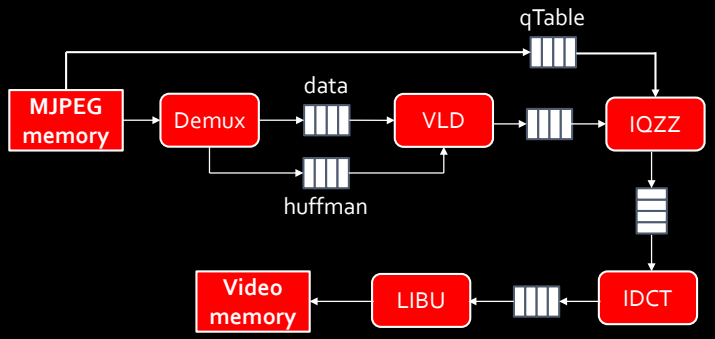
- FPGA design is dealing with larger, more complex Hardware and Software systems
- System Architects must perform rapid decision-making on functions to be realized in HW or SW to meet specific application requirements
- HW and SW engineers must refine for optimal HW/SW partitioning to fit system requirements in implementation.
- The last two points (namely HW/SW codesign and HW/SW cosynthesis) are not well established in FPGA design flow.

Problem (cont'd)

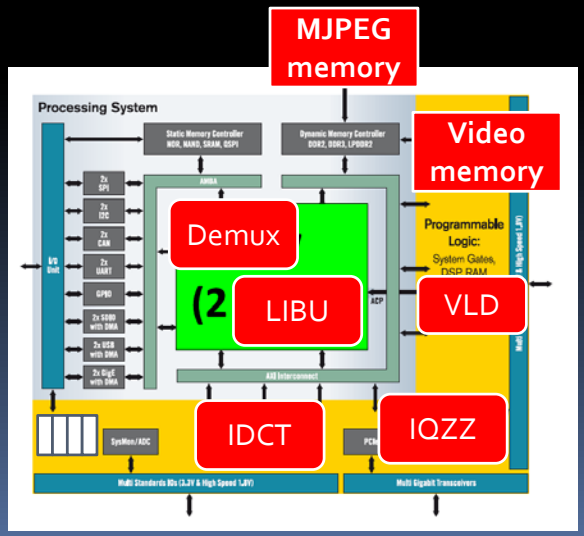
Today's FPGA workflow may take many iterations
And many hours per iteration



Architecture

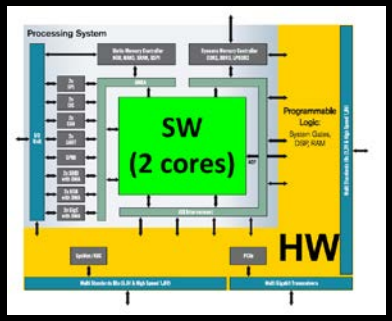


Application

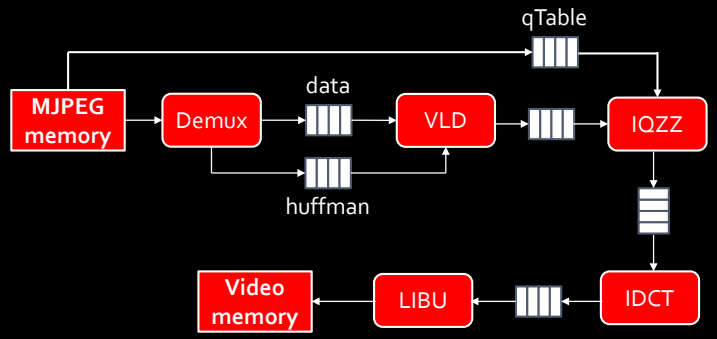


Problem (cont'd)

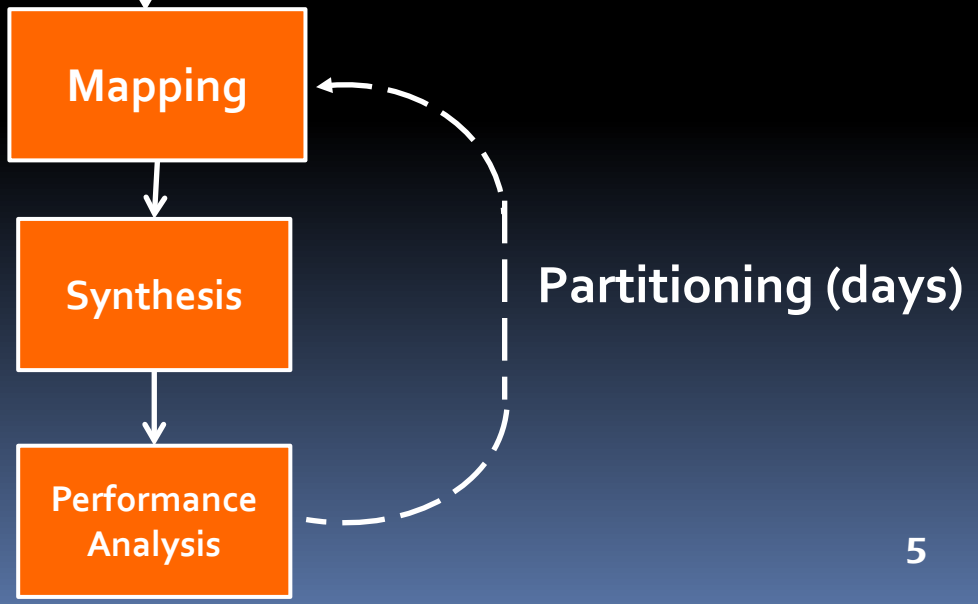
Today's FPGA workflow may take many iterations
And many days per iteration



Architecture

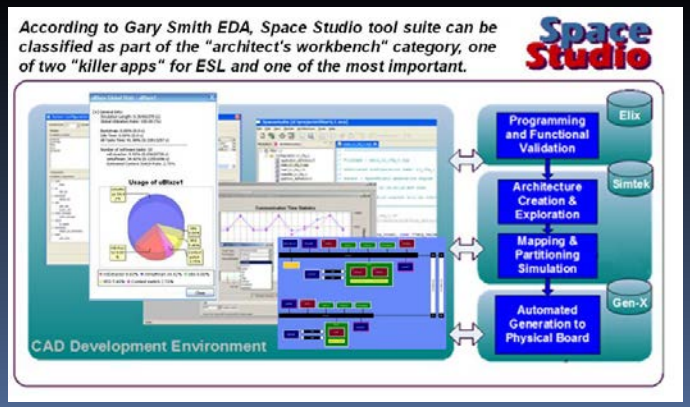


Application (Algorithm)



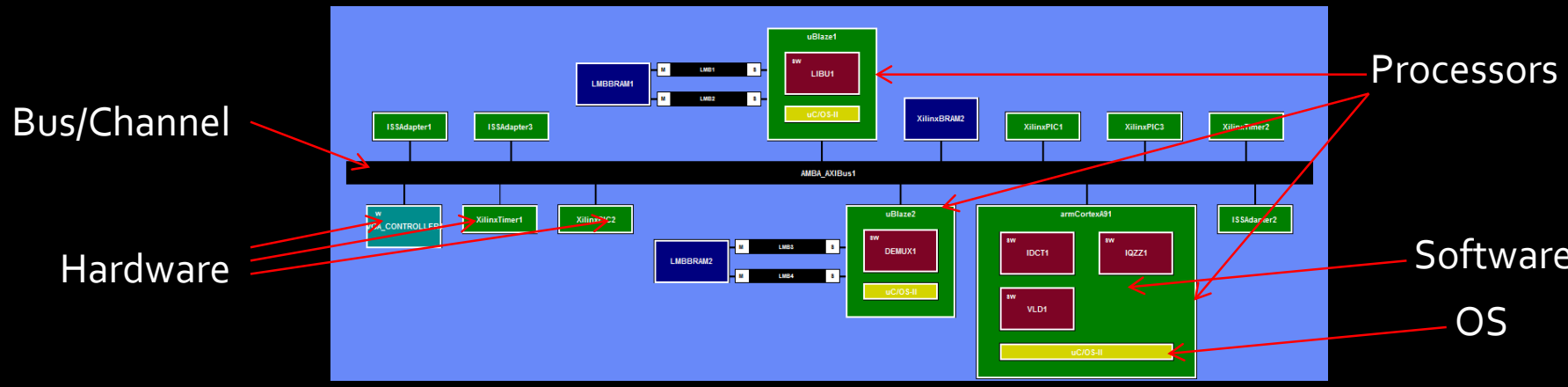
Our Solution

- Rapid decisions at front end of design process
  - Electronic System Level (ESL)
  - Create Large Complex Systems at Higher Level → reduce complexity of details …
  - Co-design of Software AND Hardware – together (Software content is *increasing*)



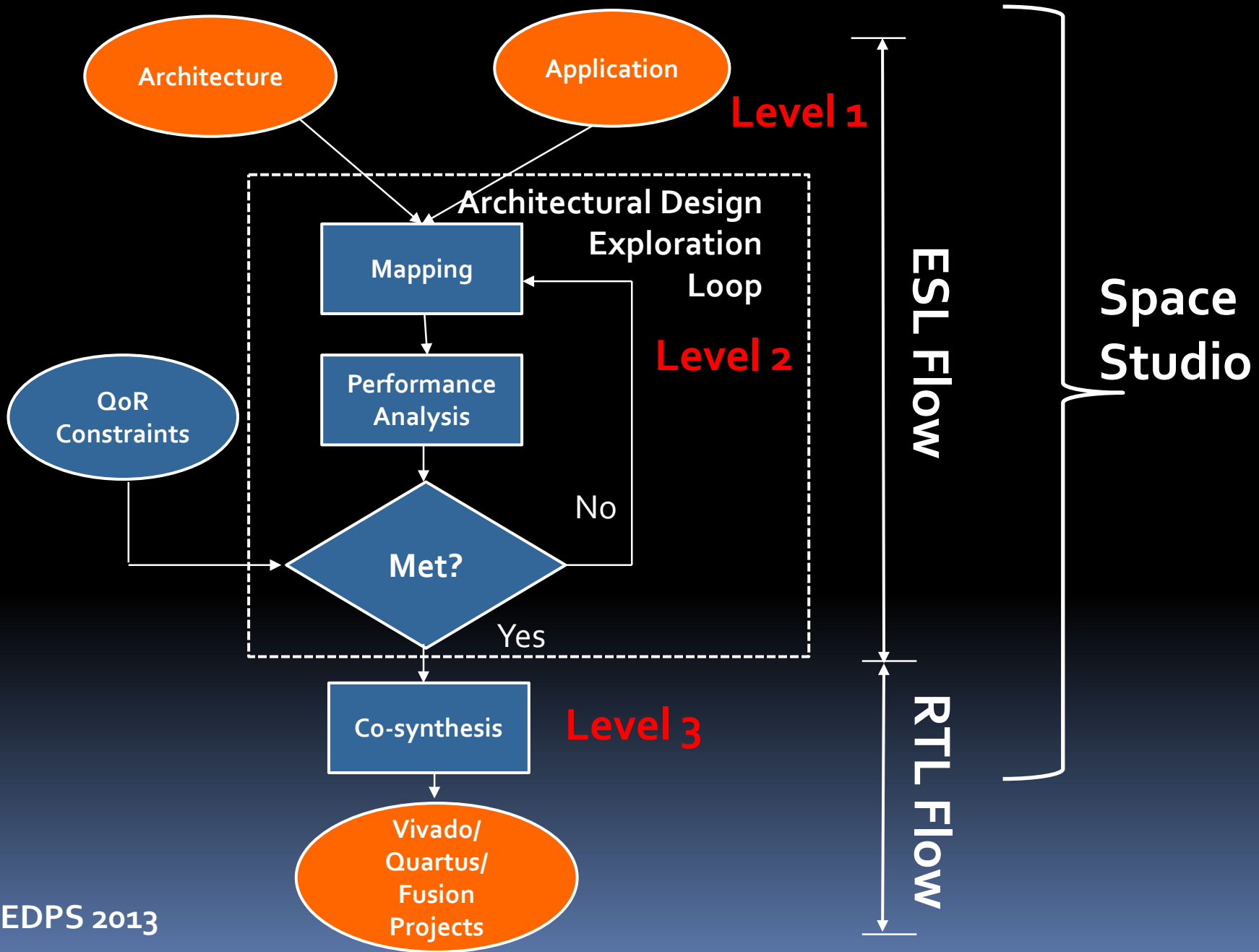
Our Solution (cont'd))

- Level 1 – Algorithm/Functional Specification
- Level 2 – Architectural Design Exploration



- Level 3 – Implementation Manager (IP Mapping, HLS) Programmable right now – ASIC tbd

Our Solution (Cont'd)



1) Application (Algorithm)

- Multi-task application specification at a high level of abstraction using C/C++ blocks

- Supported communication semantics
  - FIFO-based message passing
  - Shared memory
  - Memory-mapped I/O

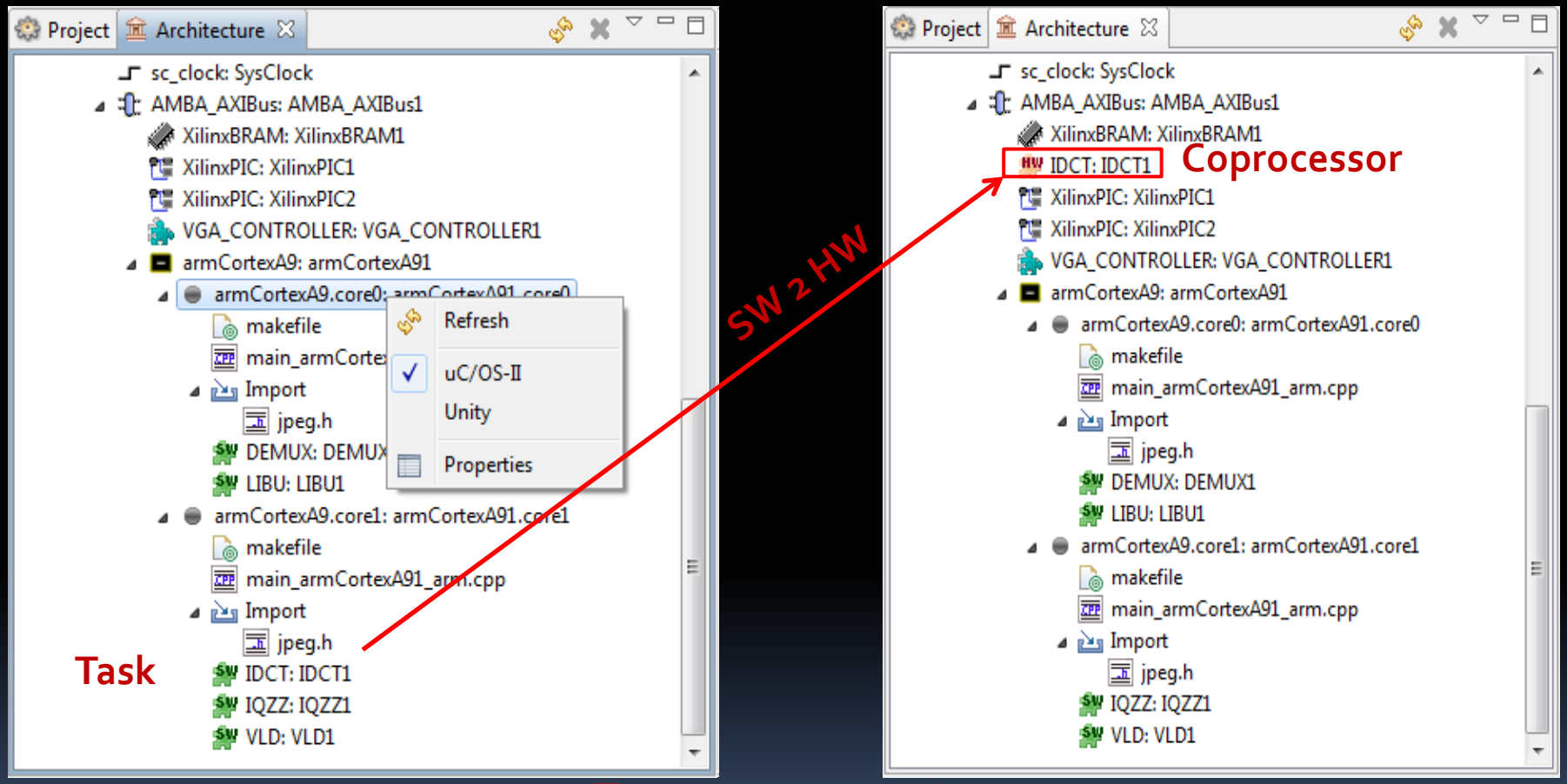
2) Architecture Design (Virtual Platform)

- Library of TLM-2.0 models for:
  - Processors
  - OS (BareMetal, uC, Linux, etc.), AMP, SMP
  - Busses and interconnects
  - Memories
  - I/O peripherals

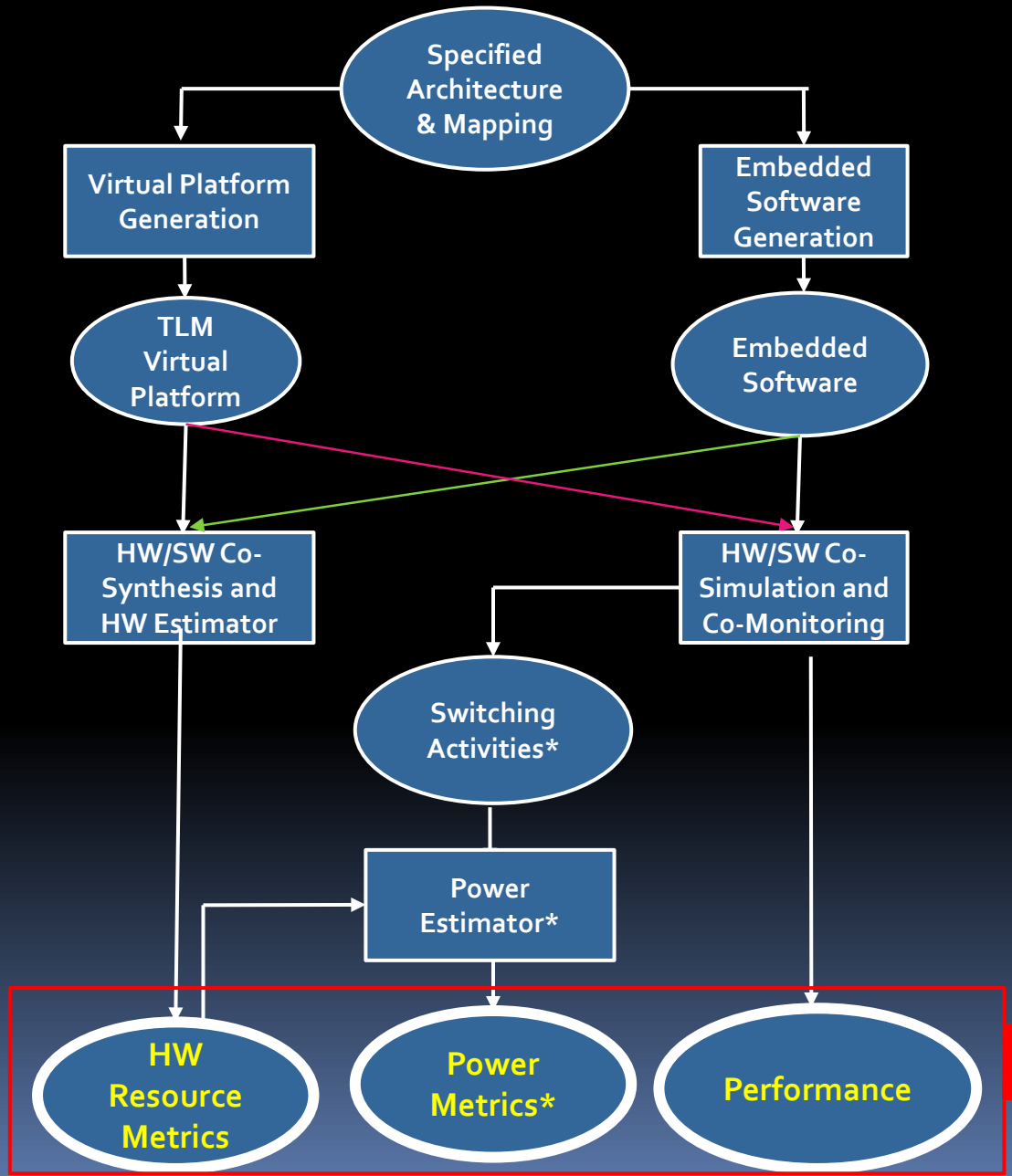
- Extensions for third-party or user-defined IPs
  - User IP Import
    - C/C++ model import w/SystemC + TLM-2.0
    - IP-XACT description for parameters and interfaces of the component

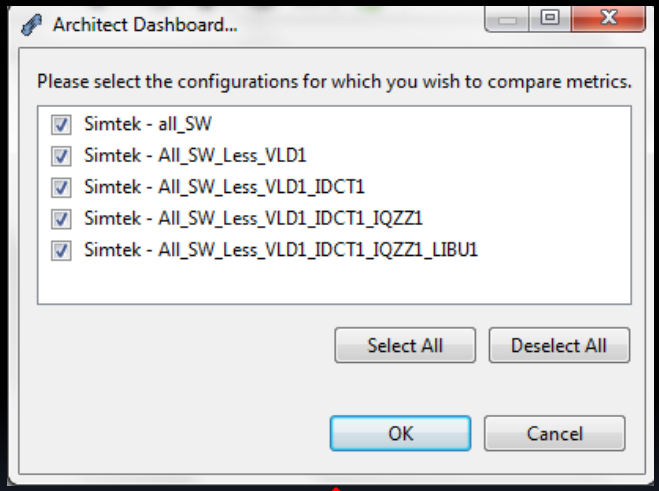
3) Function Mapping and Partitioning

- Drag and Drop Mechanism Supports Design Iteration



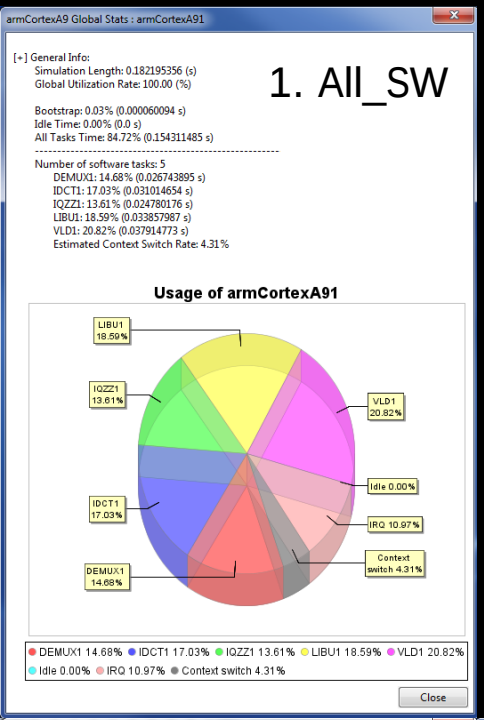
4) Performance Analysis and Evaluation



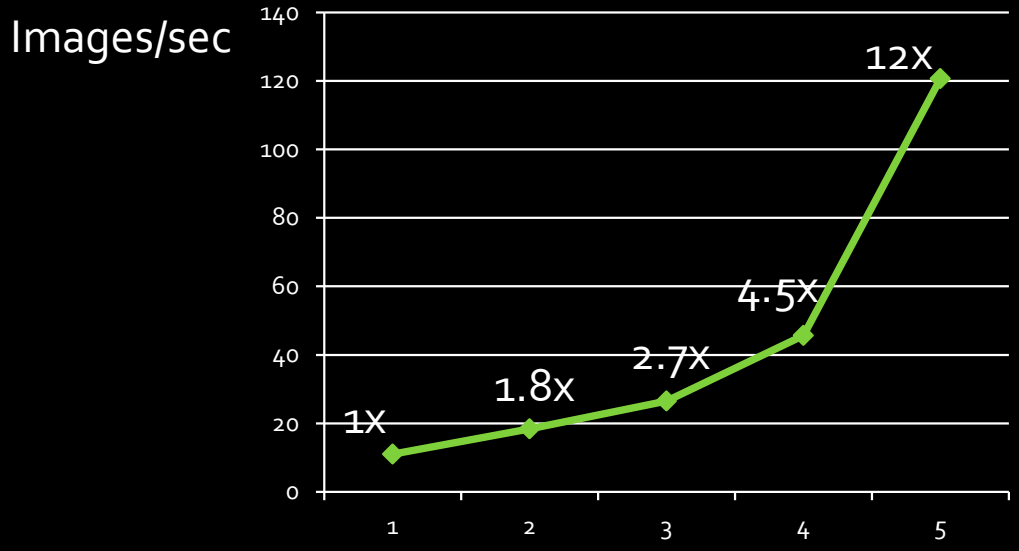


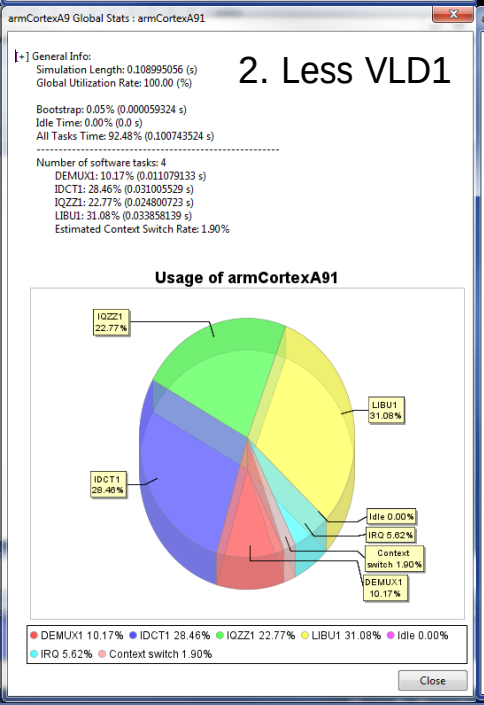
Evaluation can be automated through an Architect Dashboard

Example: Load on a single ARM Cortex-A9 core

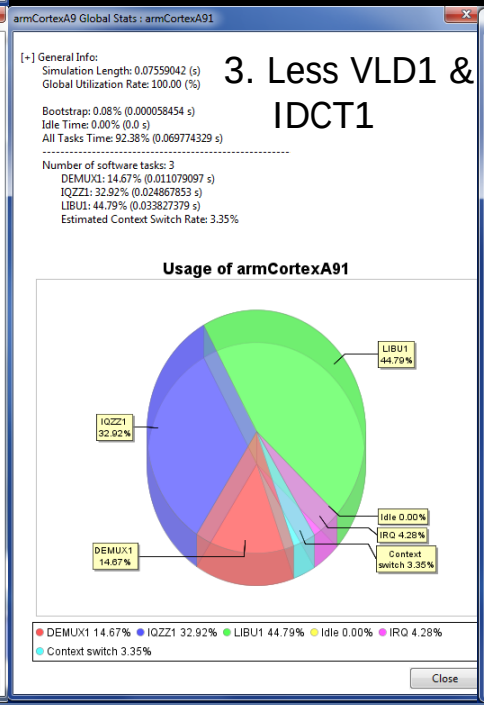


1. All_SW

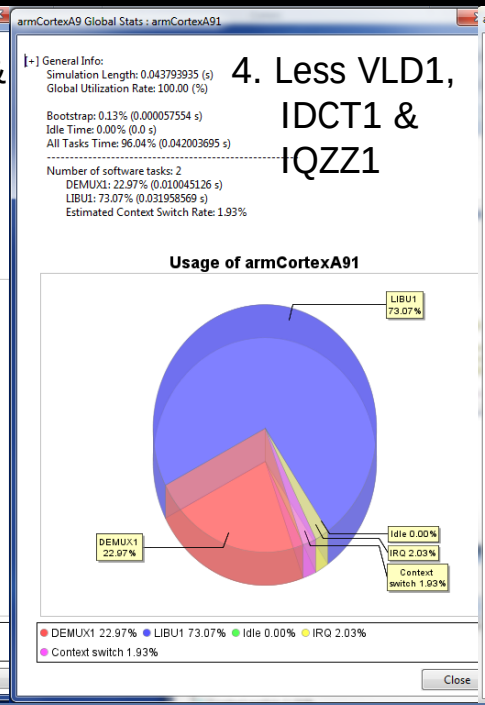




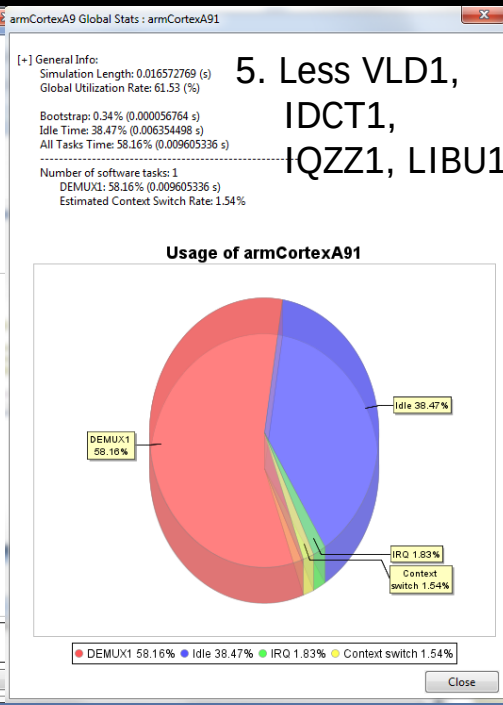
2. Less VLD1



3. Less VLD1 & IDCT1

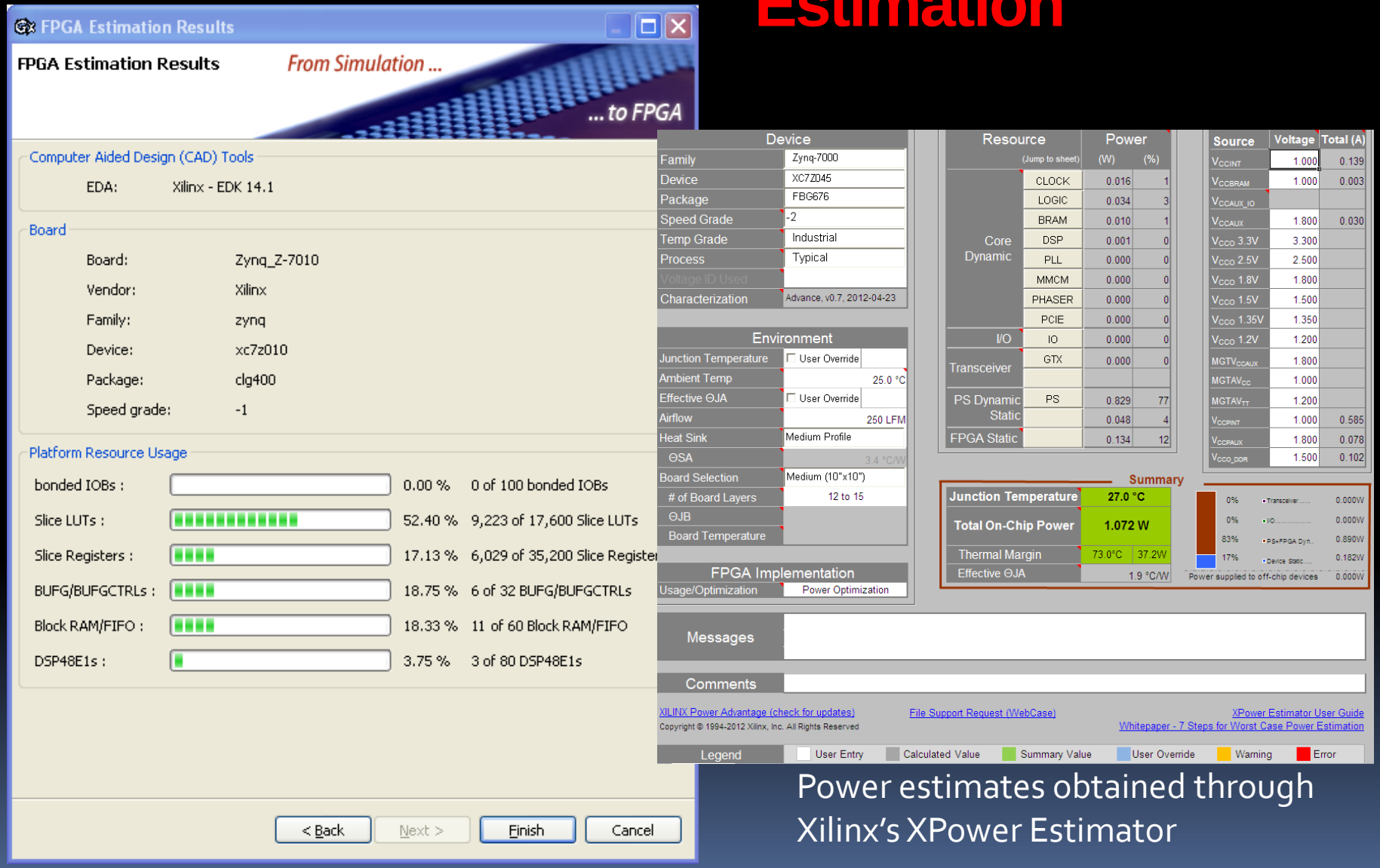


4. Less VLD1, IDCT1 & IQZZ1



5. Less VLD1, IDCT1, IQZZ1, LIBU1

Another example: Resource and Power Estimation

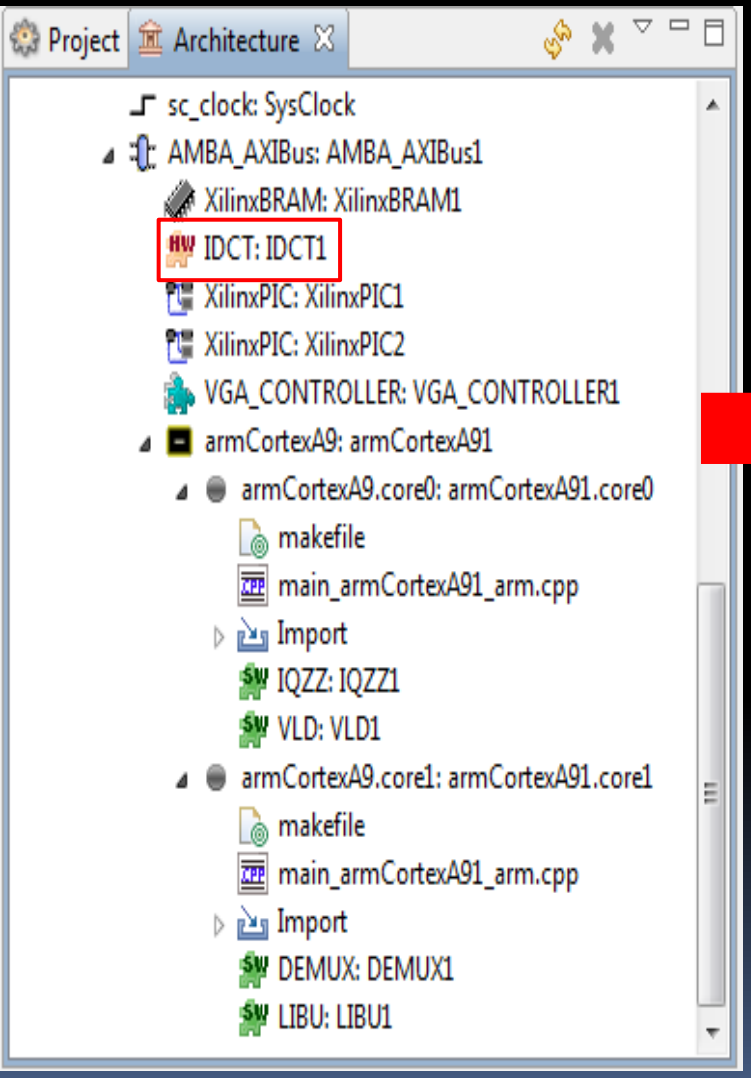


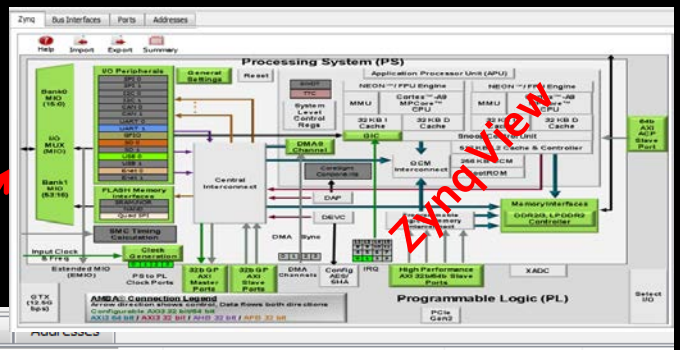
Power estimates obtained through Xilinx's XPower Estimator

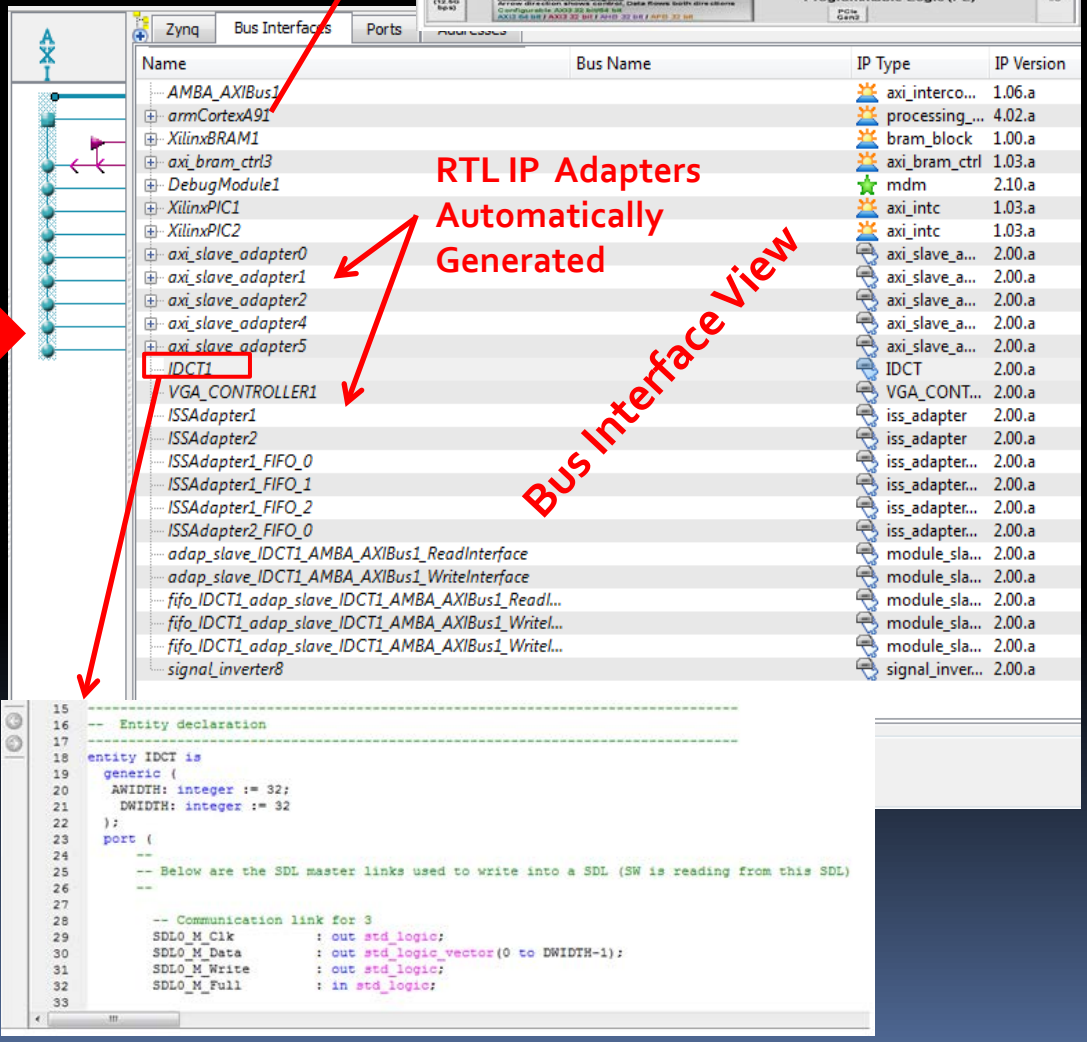
5) Co-synthesis (From ESL 2 RTL)

- Export RTL implementation of Architecture
  - RTL platform IP's, Glue Logic
  - Embedded Firmware and Software
  - Project for downstream tools (e.g., ISE, Vivado, Quartus, Fusion, etc.)
  - Support for High Level Synthesis flows for HW accelerators

5) Co-synthesis (cont'd)





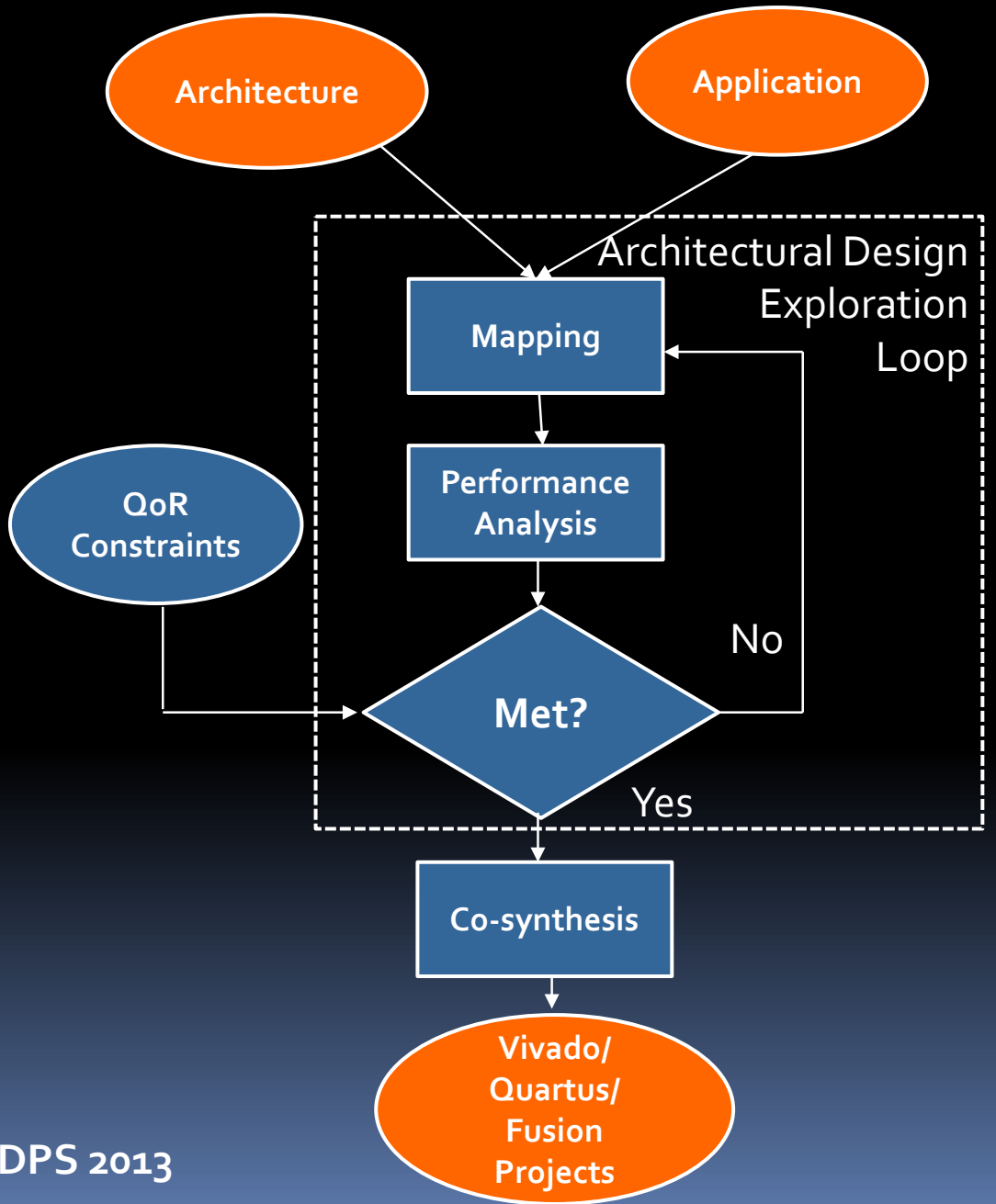



```
15  ------------------------------------------------------------------------------
16  --  Entity declaration
17  ------------------------------------------------------------------------------
18  entity IDCT is
19    generic (
20     AWIDTH: integer := 32;
21      DWIDTH: integer := 32
22    );
23    port (
24        --
25        -- Below are the SDL master links used to write into a SDL (SW is reading from this SDL)
26        --
27
28          -- Communication link for 3
29          SDL0_M_Clk          : out std_logic;
30          SDL0_M_Data         : out std_logic_vector(0 to DWIDTH-1);
31          SDL0_M_Write        : out std_logic;
32          SDL0_M_Full         : in std_logic;
33
```

Level 2: Approximately Timed

Level 3: Implementation (in Vivado)

In summary



ESL Flow

RTL Flow

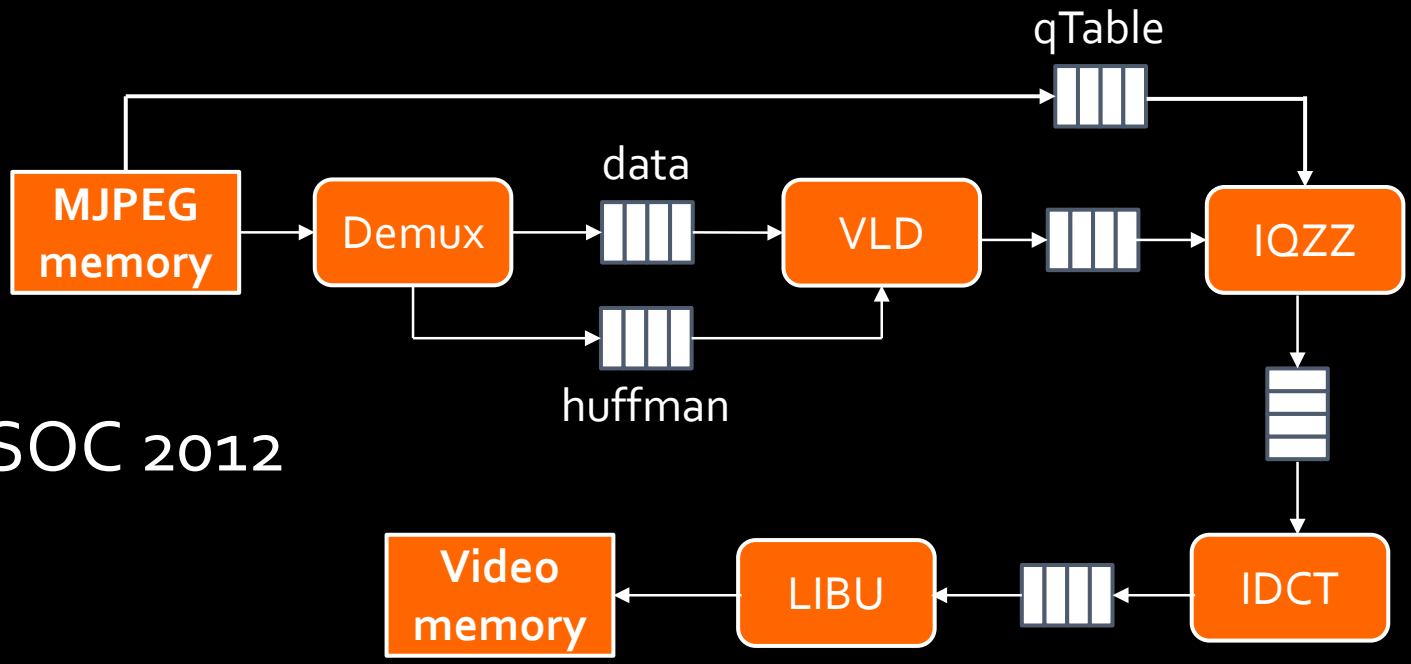
Benefits

Extensive automation:

- Iterations within minutes for all mapping changes
  - HW→HW, SW→SW, HW→SW, SW→HW
- OS as an IP block
- Fully transparent non-intrusive monitoring for performance analysis

- No manual recoding to RTL
- Firmware and adapters are automatically generated

Case Study: M-JPEG Video Decoder



MPSOC 2012

QoR Constraint Set	Performance (FPS)	Area (Largest Device)	Power (W)
1	24	Zynq-7010	1.10
2	30	Zynq-7020	0.75
3	60	Zynq-7020	0.90

Requirements (Performance, Power, Resources)

Design Space Exploration (Part 1)

- Rapid spec. & evaluation of 26 different architectures over 2.5 days
  - 1 using 1 ARM Cortex-A9 core
  - 5 using 2 ARM Cortex-A9 cores
  - 7 using 2 ARM Cortex-A9 cores + 1 MicroBlaze soft-core
  - 3 using 2 ARM Cortex-A9 cores + 2 MicroBlaze soft-cores
  - 1 using 1 ARM Cortex-A9 core + HW accelerators
  - 5 using 2 ARM Cortex-A9 cores + HW accelerators
  - 4 using mix of MicroBlaze soft-cores + HW accelerators
- Architecture has significant impact on QoR
  - Performance goes from 10 FPS to 200+ FPS
  - Power goes from 1.15 W to 1.75 W
  - Number of HW resources used goes from 1X to 6.3X

Design Space Exploration (Part 2)

- Several architectures met FPS but used too much power

- Selected 3 architectures for power/perf. trade-off
  1. LIBU → MicroBlaze, VLD → ARM core 1, others → ARM core 2
  2. DEMUX , LIBU → ARM core 1, others → HW accelerators
  3. DEMUX → ARM core1, LIBU → ARM core 2, others → HW

Arch.	FPGA Freq. (MHz)	ARM Freq. (MHz)	Perf. (FPS)	Power (W)	Area (Smallest Device)
1	100	500	24	1.072	Zynq-7010
2	50	333	54	0.727	Zynq-7020
3	50	333	72	0.861	Zynq-7020

- No. Cores, Mapping, HW/SW Part., Freq. Scal. ⇔ QoR
- Total manpower: 1 System Engineer over 25 hours

Conclusions

- From the B. Bailey's Blog:

"FPGA companies would be some of the first to implement true ESL flows"

Xilinx Builds Vivado Into Full ESL Solution, 4/3/2013

- We agree with this (e.g. , for HW/SW integration, HLS and IP's), but a complete HW/SW Co-design flow is still missing

- As a design creation front end for Xilinx Vivado, we have shown that SpaceStudio fills this gap.