A Complete HW/SW Co-design Flow for Heterogeneous Platforms

Guy Bois
Professor École Polytechnique de Montréal
President of Space Codesign Systems

MCES Workshop 2013
Space Codesign Systems Inc.

- Electronics Design Software to shorten the design cycle of SoC Embedded Systems
- Established 2008 in Montreal
  - Technology Transfer from Ecole Polytechnique de Montreal’s Microelectronics Research Lab
  - Over 30 man-years of research since 2004
- Worldwide distribution
Agenda

- Space Codesign Systems
- Problem
- Our solution
- Details about the Flow
- Case Study
- Conclusion
Traditional Workflow

- Long design exploration cycles → late problem discovery
- Hardware and Software developed on separate paths
- Long RTL simulation time → few design candidates
- Prolongs time-to-market
SpaceStudio Workflow

- Automated transformation of functions between HW and SW **Without Recoding**
- Fast high level simulation
- No C/C++ to VHDL recoding
- Automatic generation of Firmware and RTL

MCES Workshop 2013
The SpaceStudio Solution

- Rapid decisions at earlier stages 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*)

According to Gary Smith EDA, Space Studio tool suite can be classified as part of the "architect's workbench" category, one of two "killer apps" for ESL and one of the most important.
Space Codesign Technology Steps

- Algorithm / Functional Specification
  - Untimed Simulation
  - Requirements for System Architecture
- Architectural Design Exploration
  - HW/SW Co-design
    - Automated HW/SW Partitioning
  - Loosely Timed (LT) & Approximately Timed (AT) Simulation
    - Non-intrusive Monitoring
  - Development of System Architecture
- Implementation
The Flow

Elix

Simtek

ESL Flow

RTL Flow

The Flow

Architectural Design Exploration Loop

Met?

No

Yes

Co-synthesis

Vivado/Quartus/Fusion Projects

ASIC Development

Mapping

Performance Analysis

Application

Architecture

QoR Constraints

Architecture

Application

Constraints

No

Yes

Co-synthesis

Vivado/Quartus/Fusion Projects

ASIC Development

SW

HW

MCES Workshop 2013

SpaceStudio

HW

SW

Temp
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
- Results guide ASIC architecture & implementation
Zynq -7000 All Programmable Soc
Heterogeneous Virtual Platform

Based on ARM Cortex-A MPCore
Heterogeneous Virtual Platform

Based on QEMU (TLMU)
Heterogeneous Virtual Platform

SoC and SoB Platforms

Many Core Platform

QEMU 1
QEMU 2
Copro Sim

MCES Workshop 2013
3) Function Mapping and Partitioning

- Drag and Drop Mechanism Supports Design Iteration

**1st Mapping Configuration 1:** All SW

**2nd Mapping Configuration 2:** All SW less IDCT
4) Performance Analysis and Evaluation

Evaluation can be automated through Architect’s Dashboard

4. Performance Analysis and Evaluation

1. Specified Architecture & Mapping
2. Virtual Platform Generation
   - TLM Virtual Platform
3. Hardware/Software (HW/SW) Co-Synthesis and HW Estimator
4. Switching Activities*
5. Power Estimator
6. Hardware Resource Metrics
7. Embedded Software Generation
8. Embedded Software
9. HW/SW Co-Simulation and Co-Monitoring
10. Performance
11. Power Metrics
12. HW/SW Co-Simulation and HW Estimator

Evaluation can be automated through Architect’s 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)

- Generate RTL implementation of Architecture
  - RTL platform IP’s, Glue Logic
  - Embedded Firmware and Software
  - Programmables:
    - Project for downstream tools (e.g., ISE, Vivado, Quartus, Fusion, etc.)
    - Support for High Level Synthesis flows for HW accelerators
  - ASIC (for now):
    - Simtek Virtual Platform guides your ASIC architecture and implementation process
5) Co-synthesis (cont’d)

Level 2: Approximately Timed

- AMBA_AxiBus: AMBA_AxiBus1
  - XilinxBRAM: XilinxBRAM1
  - IDCT: IDCT1
- XilinxPIC: XilinxPIC1
- XilinxPIC: XilinxPIC2
- VGA_CONTROLLER: VGA_CONTROLLER1
- armCortexA9: armCortexA91
  - makefile
  - main_armCortexA91_arm.cpp
- Import
  - IQZZ: IQZZ1
  - VLD: VLD1
- armCortexA9.core1: armCortexA91.core1
  - makefile
  - main_armCortexA91_arm.cpp
- Import
  - DEMUX: DEMUX1
  - LIBU: LIBU1

Level 3: Implementation (in Vivado)

- RTL IP Adapters Automatically Generated
- Bus Interface View

MCES Workshop 2013
Case Study 1: M-JPEG Video Decoder

Requirements (Performance, Power, Resources)

<table>
<thead>
<tr>
<th>QoR Constraint Set</th>
<th>Performance (FPS)</th>
<th>Area (Largest Device)</th>
<th>Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24</td>
<td>Zynq-7010</td>
<td>1.10</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>Zynq-7020</td>
<td>0.75</td>
</tr>
<tr>
<td>3</td>
<td>60</td>
<td>Zynq-7020</td>
<td>0.90</td>
</tr>
</tbody>
</table>
Design Space Exploration Results

- Several architectures met FPS but used too much power

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

<table>
<thead>
<tr>
<th>Arch.</th>
<th>FPGA Freq. (MHz)</th>
<th>ARM Freq. (MHz)</th>
<th>Perf. (FPS)</th>
<th>Power (W)</th>
<th>Area (Smallest Device)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>500</td>
<td>24</td>
<td>1.072</td>
<td>Zynq-7010</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>333</td>
<td>54</td>
<td>0.727</td>
<td>Zynq-7020</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>333</td>
<td>72</td>
<td>0.861</td>
<td>Zynq-7020</td>
</tr>
</tbody>
</table>

- No. Cores, Mapping, HW/SW Part., Freq. Scal. $\Leftrightarrow$ QoR
- Total manpower: 1 System Engineer over 25 hours
Case Study 2: Matrix Inversion (30x30 float) on ZedBoard

<table>
<thead>
<tr>
<th>HW/SW with ARM A9 Dual Core</th>
<th>Throughput (matrix/sec)</th>
<th>Latency (sec)</th>
<th>LUT</th>
<th>FF</th>
<th>DSP</th>
<th>BRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% SW - VFP</td>
<td>335</td>
<td>0.00597</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100% SW + VFP*</td>
<td>4675</td>
<td>0.0004278</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Best HW/SW</td>
<td>1411</td>
<td>0.001417</td>
<td>11572</td>
<td>9542</td>
<td>50</td>
<td>118</td>
</tr>
<tr>
<td>100% HW (w/o optimization)</td>
<td>546</td>
<td>0.00366</td>
<td>19214</td>
<td>13540</td>
<td>50</td>
<td>128</td>
</tr>
</tbody>
</table>

* Neon vectorization
Conclusions

• Can start working on your product implementation on day one, and proceed with incremental refinement.

• Using the same models to create HW and SW implementation of a heterogeneous embedded system, and the speed of simulating ESL models for performance analysis of each candidate, it is possible to converge rapidly on a design that fulfills QoR goals.

• If not, one can rapidly examine the ability of the chosen architecture platform and decide which changes to make