Toward Composable Multimedia MP-SoC Design

H. Nikolov, T. Stefanov, E. Deprettere – Leiden University
M. Thompson, A. Pimentel, S. Polstra – University of Amsterdam
C. Zissulescu, R. Bose – Chess B.V. (Image Processing Solutions)
Market pull: “Design better products faster”

- **Productivity**
  - Design technology: architectures, methods, tools

- **Quality**
  - Low cost, low power, flexible, no bugs

- **Complexity**
  - Need to support multiple applications / standards

- **Emergence of**
  - MP-SoCs to cope with computing power demand
    - Embedded processor cores and dedicated HW
    - Network on Chip (NoC)
  - System-level design to cope with design complexity
System-level MP-SoC Design

- Platform-based design
  - Domain-specific platform architectures
  - Re-use of IP

- High-level modeling and simulation of system components and their interactions

- Separation of concerns
  - application vs. architecture
  - computation vs. communication

Although system-level design is very promising, it still remains very complex.
Some important ingredients
- Decomposing applications for mapping onto MP-SoC
- Hardware/software partitioning of applications
- Modeling and simulating MP-SoC architecture(s)
- Efficient (and early!) exploration of design options
  - Architecture trade-offs
  - Different mappings and HW/SW partitionings
- Mapping application(s) / system implementation

Different tools / tool-flows are usually needed
- Interoperability problems!

There typically remains a large gap between system-level models and actual implementations
Daedalus: *Making system-level design take off*

- Building upon a decade of system-level design research
- From sequential application to MP-SoC implementation/prototype in a matter of hours
- Unparalleled degree of design automation
- Fully integrated tool-flow, no interoperability problems
- High-quality, open source framework
  - [http://daedalus.liacs.nl](http://daedalus.liacs.nl)
- Presented at CODES+ISSS, 2007, pp. 9-14
Daedalus Design-flow

**Library of IP cores**
- System-level specification
- RTL-level specification

**Sesame**
- Explore, modify, select instances
- Platform specification
- Mapping specification

**KPNgen**
- Sequential application

**ESPAM**
- Synthesizable VHDL
- C/C++ code for processors
- Xilinx Platform Studio (XPS)

**MP-SoC**
Daedalus – Chess B.V.
Case Study
Daedalus-Chess case study

- Image processing solutions for customers that build
  - Medical appliances
    - Very high resolution images
  - Industrial process monitoring
    - Very high frame rate

Joint project: Chess B.V. uses Daedalus:

- Still image JPEG/JPEG2000 compression system
- Very fast evaluation (exploration and implementation) of alternative systems (MP-SoCs)
- Trade-off between
  - Cost, Design time, Performance, etc.
Initial case study – JPEG encoder

Tile = 128 MacroBlocks

- Packet of bytes
- Compressed byte sequence for Tile

MacroBlock = 2Yblocks + 1Ublock + 1Vblock
- Yblock = 64 pixels,
- Ublock = 64 pixels,
- Vblock = 64 pixels,
MP-SoCs consist of MicroBlaze softcores and/or dedicated HW components

- Point-to-point connections
- IP component library contains
  - High-level HW component model for all tasks (Sesame)
  - Contains RTL HW model only for DCT task (ESPAM)

MP-SoCs exploit both task (pipeline) parallelism and data parallelism

MP-SoC implementations on FPGA are constrained by the on-chip memory (288KB)
Example architecture instances for a single-tile JPEG encoder:

- **2 MicroBlaze processors (50KB)**
  - 16KB Vin,DCT
  - 2KB
  - 32KB Q,VLE,Vout

- **6 MicroBlaze processors (120KB)**
  - 8KB Vin
  - 4x2KB
  - 4x2KB
  - 4x16KB
  - 32KB DCT, Q
  - 2KB VLE, Vout

- **1 MicroBlaze, 1HW DCT (36KB)**
  - 32KB Vin,Q,VLE,Vout
  - 4KB
  - 4KB DCT

- **4 MicroBlaze, 2HW DCT (68KB)**
  - 2KB Vin
  - 8KB
  - 32KB
  - 8KB
  - 8KB
  - 4KB
  - 4KB
  - 4KB
  - 4KB
Sesame’s DSE results
Sesame DSE results

Single JPEG encoder DSE:

Performance-memory trade-off DSE

- **Implementable**
- **Implementable pareto front**
- **Non-implementable**
- **Homogeneous**
Sesame DSE results (cont’d)

Predicted speed-ups of multi JPEG encoder MP-SoCs:

Projected speed-ups

- Homogeneous MP-SoCs
- Heterogeneous MP-SoCs

Estimated speed-up vs Number of processing cores in MP-SoC
ESPAM’s synthesis results
ESPAM synthesis results

JPEG case study, homogeneous systems (32 tiles):
ESPAM synthesis results (cont’d)

JPEG case study, heterogeneous systems (32 tiles):

Memory utilization (KB)

Million clock cycles

#Cores: 2 4 6 8 10 12 14 16 18 20 22 24
MB/DCT: 1/1 2/2 3/3 4/4 5/5 6/6 7/7 8/8 11/7 13/7 14/8 16/8

ESPAM synthesis results (cont’d)
The best of all, we performed

- The DSE study (≤ 5% error) and
- The implementation of 25 MP-SoC JPEG encoder variations on an FPGA in only 5 days!

- Combining data and task parallelism:
  24 cores, 19.7x speed-up, 288KB memory
So, what about the name Daedalus?

- Daedalus means “cunning worker” in Latin
- He was an innovator in many arts
- Daedalus was the father of Icarus

Analogy:
- It’s new, disruptive technology
- But there are still limitations

“Don’t fall into the sea”!

*Daedalus and Icarus*, by Charles Paul Landon, 1799
Conclusions

**Merits of the Daedalus design-flow:**

- Automated parallelization of media/streaming applications into parallel specifications (KPNs)
- Automated synthesis of MP-SoC platforms at system level, in a plug-and-play fashion
- Automated mapping of parallel application specifications onto MP-SoC platform
- Steering by means of efficient system-level design space exploration
- All of this in a matter of hours