Heuristic Optimisation Methods for System Partitioning in HW/SW Co-Design

Univ.Prof. Dipl.-Ing. Dr.techn. Markus Rupp
Vienna University of Technology
Institute of Communications and Radio-Frequency Engineering

Dr. Bastian Knerr, Dr. Martin Holzer
• Embedded system design

• System partitioning
  • Heterogeneous platforms
  • Mapping system graphs to platforms
  • Heuristic optimisation for platform mapping
  • Multi-objective optimization (Area, Timing)

• Conclusions
An embedded system is a computing device
- which is in general subject to a specific purpose.
- whose implementation is predominantly determined by this purpose.
- which entails a complete encapsulation into the environment where the purpose is located at.
Embedded system design

Spec sheets

Source code

System Graphs

Platform Models

Abstract high

HW/SW Partitioning

Scheduling

Cosimulation

Executable System Description

System Requirements

Performance Estimation

Hardware Modelling

Communication Modelling

Software Modelling

Hardware Synthesis

Bus/Memory/DMA Definition

Code Generation

Integration & Test

Prototyping/Implementation

Performance Analysis

Prototype or Product

Prototype or Product

Software

Hardware
Outline

• Embedded system design

• System partitioning
  • Heterogeneous platforms
  • Mapping system graphs to platforms
  • Heuristic optimisation for platform mapping
  • Multi-objective optimization (Area, Timing)

• Conclusions
Heterogeneous platforms

• Classical HW/SW co-design platform

  - First published in 1992
  - Served well to get a first grip on (bi-)partitioning
  - Recent work still uses this classical model (Kalavade 2002, Wiangtong 2002)
  - This model does not sufficiently represent state-of-the-art platforms in embedded systems
Heterogeneous platforms

- Modern system-on-chip platform
- UMTS baseband transceiver chip (2003)
- DSP+Microcontroller
- ASICs
- Busses and bridges
- RAM and registers
- Interfaces
Heterogeneous platforms

Contribution: Design of platform composition library

- Processor
  - DSP
  - FPGA
  - ASIC
- Communication
  - Bus
  - FIFO
  - Direct
- Memory


• Embedded system design

• System partitioning
  • Heterogeneous platforms
  • Mapping system graphs to platforms
  • Heuristic optimisation for platform mapping
  • Multi-objective optimization (Area, Timing)

• Conclusions
Mapping system graphs to platforms
Mapping system graphs to platforms

The constrained mapping problem

Proven to be NP-hard → Use heuristics
Outline

- Embedded system design
- **System partitioning**
  - Heterogeneous platforms
  - Mapping system graphs to platforms
  - Heuristic optimisation for platform mapping
  - Multi-objective optimization (Area, Timing)
- Conclusions
• Popular heuristics for system partitioning

✓ Gradient search
✓ Simulated annealing
✓ Tabu search
✓ Kernighan-Lin min-cut
✓ Genetic algorithm
✓ Global criticality/local phase
✓ Restricted range exhaustive search
Heuristic optimisation

Contribution: New or substantially improved methods

• Kernighan-Lin min-cut
  B. Knerr, M. Holzer, M. Rupp, HW/SW Partitioning Using High Level Metrics, Int. Conf. on Computing, Communications, and Control (CCCT), Austin, TX, USA, August 2004.

• Genetic algorithm

• Global criticality/local phase (GCLP)

• Restricted range exhaustive search (RRES)
Contribution:
- Analysis of representative graphs for signal processing
- Identification of their specific typical properties

Density

- Density/Sparsity
- Degree of parallelism
- Avg number of parallel nodes
- Rank-locality

\[ r_{\text{loc}} \in [1.0 \ldots 3.0] \]

dense \( (\rho = 21/8) \)

sparse \( (\rho = 10/8) \)
Heuristic optimisation

- Consider graphs with strong locality property
- Compare GA with different genome codings
  - Randomly ordered
  - Rank ordered
  - Locality ordered
- Multi-core platform
- 3 Objectives
  - Area, Code, Time

\[ k_{loc} = 5 \]
\[ |V| = 25 \]
Heuristic optimisation

• Analysed the dependency of different algorithms on graph locality

\[ \frac{\Omega_{GA}}{\Omega_{min}} \quad \frac{\Omega_{RRES}}{\Omega_{min}} \quad \frac{\Omega_{TS}}{\Omega_{min}} \]

\[ \rho = [1..10] \]

• Comparison of GA, TS, and RRES over \( \kappa \) on large sparse graphs \( (|V| = 100) \)
• Embedded system design

• **System partitioning**
  • Heterogeneous platforms
  • Mapping system graphs to platforms
  • Heuristic optimisation for platform mapping
  • Multi-objective optimization (Area, Timing)

• Conclusions
Execution Time Estimation

- **Metrics**
  - Number of operations
  - Parallelism
  - Available resources
Execution Time Profile

- **Path analysis**
  - Best Case Execution Time (BCET)
  - Worst Case Execution Time (WCET)
  - Infeasible paths
    - Condition \((A < 1) \&\& (A \geq 1)\) cannot be fulfilled

- Narrow bounds for the execution time interval
Comparison Synthesis/Estimation

- Benchmark algorithms
  - UMTS cell searching
  - MPEG algorithm

- Comparison with results from high level synthesis (SPARK)

- **Fidelity** value counts how often the relation between two estimations \((E(i), E(j))\) is preserved also for the real values \((R(i), R(j))\)

- Fidelity of 100% achieved

\[
Fidelity = 100 \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mu_{ij}
\]

\[
\mu_{ij} = \begin{cases} 
1 & \text{if } \begin{cases} 
R(i) < R(j) \wedge E(i) < E(j) \\
R(i) > R(j) \wedge E(i) > E(j) \\
R(i) = R(j) \wedge E(i) = E(j)
\end{cases} \\
0 & \text{otherwise}
\end{cases}
\]
void f()
{
    ...
    ...
    for(i = 0; i < 4; i++)
    {
        g[i] = h[i] * k;
    }
    ...
    ...
    return;
}
void f()
{
    ...
    z = (a+b)*c + (d+e)*f;
    ...
}
return;

Multi-objective Optimization

• Minimization of a set of conflicting functions
  \[
  \min \{f_1(x), f_2(x), \ldots, f_n(x)\}
  \text{ subject to } x \in S
  \]

• A decision \( x \) is Pareto optimal if there is no other decision that dominates \( x \)

• Set of Pareto optimal points is called Pareto front

• Evolutionary algorithm approach to compute Pareto front
Evolutionary Optimisation

![Graph showing area complexity against cycle count]

- Area complexity decreases as cycle count increases.
• Performance measure: hyper volume indicator

• Improvement of design space coverage by 20%

Pareto Front Examples

- Control flow graph
  - 10 basic blocks
  - No loops
  - $10^{10}$ design points
  - 14 Pareto optimal design points

- Control flow graph
  - 15 basic blocks
  - 2 loops
  - $10^{20}$ design points
  - 29 Pareto optimal design points
Conclusions

- Formulation of the system partitioning problem for state-of-the-art embedded systems

- Design of modular platform library
  - Allowing for arbitrary platform compositions
  - Including board level and SoC models

- Analysis of typical graphs in digital signal processing
  - Identification of specific value ranges for the graph properties

- Multi-objective optimization algorithm
  - Coverage improvements of 20% increases the number of implementation variants
Conclusions

• Evaluation of a large variety of heuristics
  • Including ES, GS, SA, TS, GA, RRES, GCLP, KL
  • Analysis of the impact of system graph properties on the algorithm's performance
  • Identification of appropriate algorithms depending on

• Improved genome sequence for genetic algorithms
  • Proposed genome sequence according to graph properties
  • Cost reduction of ~50% compared to random sequence

• Proposed new algorithm (RRES)
  • RRES exploits strong locality component of system graphs
  • Outperforms any other algorithm on a distinct subset of system graphs
Thank you for your attention