Hardware-Software Codesign

7. Design Space Exploration

Lothar Thiele
System Design

- **specification**
- **system synthesis**
- **estimation**

- **SW-compilation**
- **instruction set**
- **HW-synthesis**

- **intellectual prop. code**
- **machine code**
- **net lists**

- **intellectual prop. block**
Optimization-Analysis Cycle

**decision vector** $X$

**evaluation model** (e.g., simulation, analytic)

**objective vector** $f(X)$

**optimization algorithm**

make decisions only by knowing (and comparing) $f$
Example: Simple Mapping Model

EA

1. selection
2. recombination
3. mutation

“chromosome” = encoded allocation + binding

search algorithm

solutions

analysis of individual solutions

allocation

binding

decode allocation

decode binding

scheduling

fitness evaluation

fitness

user constraints

design point (implementation)

binding β

scheduling τ

allocation α

fitness evaluation

user constraints
Remember …

**Definition:** A specification graph is a graph $G_S=(V_S,E_S)$ consisting of a data flow graph $G_P$, an architecture graph $G_A$, and edges $E_M$. In particular, $V_S=V_P \cup V_A$, $E_S=E_P \cup E_A \cup E_M$.
**Definition**: Given a specification graph $G_S$, an implementation is a triple $(\alpha, \beta, \tau)$, where $\alpha$ is a feasible allocation, $\beta$ is a feasible binding, and $\tau$ is a schedule.
Challenges of EAs in DSE

- encoding allocation+binding
  - simple encoding
    e.g., one bit per resource, one variable per binding
    - easy to implement
    - ... however, it may lead to (many) infeasible partitioning solutions
  - encoding + repair
    e.g. simple encoding AND modify
    s.t. for each \( v_p \in V_p \) there exists at least one \( v_a \in V_A \) with \( \beta(v_p) = v_a \)
    - reduces number of infeasible partitioning solutions

- ("smart") generation of initial population

- ("smart") neighborhood operations, e.g., mutation, crossover
Example Network Processors - Definition

- Typically, network processors serve as bridge between the network and the source/sink audio/video device (or set of devices)

  - **implementation**: high-performance, programmable devices optimized for (real-time) network packet processing

  - **features**: complex packet processing capabilities at high line speeds (routing, forwarding, de-/encryption, de-/compression, ...) and means to guarantee quality-of-service
Network Processor Architecture (*)

Network processor heterogeneous hardware/software architecture:

- available processing units
  - … are described in resource set $R = \{\text{ARM9, PowerPC, DSP, MEngine, Classifier, Cipher, LookUp, CheckSum}\}$
  - … have a relative implementation cost $\text{cost}(r) \geq 0$, $r \in R$
  - … and are selected for a specific architecture during the allocation step
    - with $\text{alloc}(r) = 1$ if a resource is selected and 0 otherwise

Network Processor Task Model

application structure: set of streams $s \in S$ and set of tasks $t \in T$
- each stream includes an ordered sequence of tasks $V(s) = [t_0, \ldots, t_n]$

example:
$S = \{\text{RTSend}, \text{NRTDecrypt}, \text{NRTEncrypt}, \text{RTRecv}, \text{NRTForward}\}$
Problem: Optimal Design of Network Processor

- mappings $M \subseteq T \times R$: all possible bindings of tasks
  - i.e., if $(t, r) \in M$, then task $t$ could be executed on resource $r$
- request $w(r, t) \geq 0$
  - i.e., execution of one packet in $t$ would use $w$ computing units of $r$
- resource allocation cost $c(r) \geq 0$
- binding $Z$ of tasks to resources $Z \subseteq M$ (leading to actual implementation)
  - subset of mappings $M$ s.t. every task $t \in T$ is bound to exactly one allocated resource $r \in R$ with $\text{alloc}(r) = 1$ and $r = \text{bind}(t)$
The design of network processors typically faces conflicting goals:

- **Delay constraints**
  - e.g., maximal time a packet is processed within NP

- **Throughput maximization**
  - e.g., maximum throughput of NP (packets per second)

- **Cost minimization**
  - implementation with small amount of resources (e.g., processing units, memory, and communication networks)

- ... and conflicting usage scenarios
  - usually, a packet processor is used in several different systems (e.g., router or consumer multimedia processing device) and might have different implementations with different throughput/delay requirements
NP Design Space Exploration

**issues to be considered during system-level design (and synthesis):**

- **allocation**
  - determine hardware components of the network processor

- **binding**
  - for each process of the software application choose an allocated hardware unit which executes it

- **scheduling**
  - for the set of tasks mapped onto a specific resource choose scheduling policy/parameters – from available run-time environment, e.g., a fixed priority for each stream $s$: $\text{prio}(s) > 0$
Design Space Exploration Flow

- Hardware Architecture Template
- Software Application
- Run-Time Environment
- Application Scenarios

Variation:
- Allocation
  - alloc(r) = 0/1
  - HW Architecture
- Binding
  - r = bind(t)
  - HW/SW Architecture
- Scheduling
  - prio(s)>0
  - HW/SW Architecture
- Performance Analysis

Selection:
- Multiobjective Evolutionary Selection
- Constraints Opt. Criteria
- HW/SW Architectures Cost and Performance
Tools and a Small Demo
... Some Results

- Performance of encryption/decryption
- Performance of RT voice processing

**Results Summary**

- **DSP**
  - NRT: 64%
  - RT: 39%

- **Cipher**
  - NRT: 71%
  - RT: 0%

- **LookUp**
  - NRT: 15%
  - RT: 6%

- **Classifier**
  - NRT: 27%
  - RT: 11%

- **DSP**
  - NRT: 35%
  - RT: 39%

- **LookUp**
  - NRT: 1%
  - RT: 6%

- **Classifier**
  - NRT: 1%
  - RT: 11%

**Cost**
Example: Wave Field Synthesis

What is wave field synthesis (WFS)?

- high quality spatial sound reproduction system for huge listening areas
- 32 sound sources and 300 loudspeakers for medium sized reproduction rooms
System Specification: WFS Application

Parallel application modeled as Kahn process network

structure: XML

functionality: ANSI C & DOL(*) API

Algorithm 1 Process Model
1: procedure INIT(DOLProcess p) ▷ initialization
2: initialize local data structures
3: end procedure
4: procedure FIRE(DOLProcess p) ▷ execution
5: DOL_read(INPUT, size, buf) ▷ blocking read
6: manipulate
7: DOL_write(OUTPUT, size, buf) ▷ blocking write
8: end procedure

(*) DOL – distributed operation layer: http://www.tik.ee.ethz.ch/~shapes/dol.html
System Specification: Architecture

- Architecture is modeled at abstract level in XML format
- Modeled elements:
  - processors, buses, memories
  - communication paths between these elements
  - ... parameters are included in the model
Application-to-Architecture Mapping

parallel application

heterogeneous architecture

design space exploration
(performance analysis & mapping optimization)

software synthesis
Simple Analysis Model

\[ \text{max processor load} \]

\[ obj_1 = \max_{c \in C} \left\{ \sum_{p \text{ mapped to } c} n(p) \cdot r(p, c) \right\} \]

number of activations of process \( p \)

runtime of process \( p \) on processor \( c \)

\[ \text{max bus load} \]

\[ obj_2 = \max_{g \in G} \left\{ \sum_{s \text{ mapped onto } g} \frac{b(s)}{t(g)} \right\} \]

communication link with worst load

communication request from channel \( s \)

bandwidth of communication link \( g \)
Where Are Data Obtained From?

- **Static parameters**: bandwidth of buses \( t(g) \)
- **Functional simulation**: number of activations for each process \( n(p) \), amount of data for each channel \( b(s) \)
- **Instruction-set simulation**: runtime of each process on different processors \( r(p, c) \) by using benchmark mappings
Design Space Exploration Cycle – An Example
EXPO: Example

- microphones
- convolution
- sum
- loudspeakers

max. bus load

search direction

single processor mapping

max. processor load

EXPO, Institute TIK, ETH Zurich

current population

x10^2

x axis

y

5

4

3

2

1

0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1

x10^{2}

ARM

mAgic

AHBO

Swiss Federal Institute of Technology

Computer Engineering and Networks Laboratory