Software Performance Estimation in MPSoC Design

Marcio Seiji Oyamada[1,2], Flávio Rech Wagner[1],
Wander Cesario[2], Marius Bonaciu[2],
Ahmed Jerraya[2]

UFRGS[1]
Instituto de Informática
Porto Alegre, Brazil
http://www.inf.ufrgs.br/~lse

TIMA Laboratory[2]
SLS Group
Grenoble, France
http://tima.imag.fr/sls

Motivation

- Very large design space of embedded MPSoCs
  - High-level performance estimation tools are required
  - Must be combined with fast design exploration strategies
- Embedded systems are software-dominated
  - Evaluation of processor performance under various workloads
    - e.g. exploring the allocation of tasks to various processors

Motivation

- Support for SW performance estimation and analysis at different abstraction levels
- At specification level
  - Estimation must be fast, for fast design space exploration
  - Some inaccuracy is accepted
- At RT level
  - Accurate performance analysis after the architecture definition
  - Evaluation of OS and communication overhead in an MPSoC environment

Motivation

- For accurate performance analysis and identification of bottlenecks, simulation models must be instrumented with appropriate profiling resources
- These models must be generated according to virtual prototypes coming from the synthesis flow
- However, there is a poor integration between the synthesis flow and the performance evaluation flow
  - Manual configurations of performance models are often required

Goals of this work

- Integrated methodology for software performance analysis at different abstraction levels
  - Different trade-offs between speed and accuracy of the performance analysis
  - High-level performance evaluation for fast design space exploration
- Performance evaluation flow tightly coupled with an MPSoC synthesis flow

Outline

1. MPSoC design flow
2. Software performance estimation methodology
3. Neural network-based performance estimation
4. Virtual prototype-based performance estimation
5. Case study: MPEG4 encoder

Outline

1. MPSoC design flow

2. Software performance estimation methodology

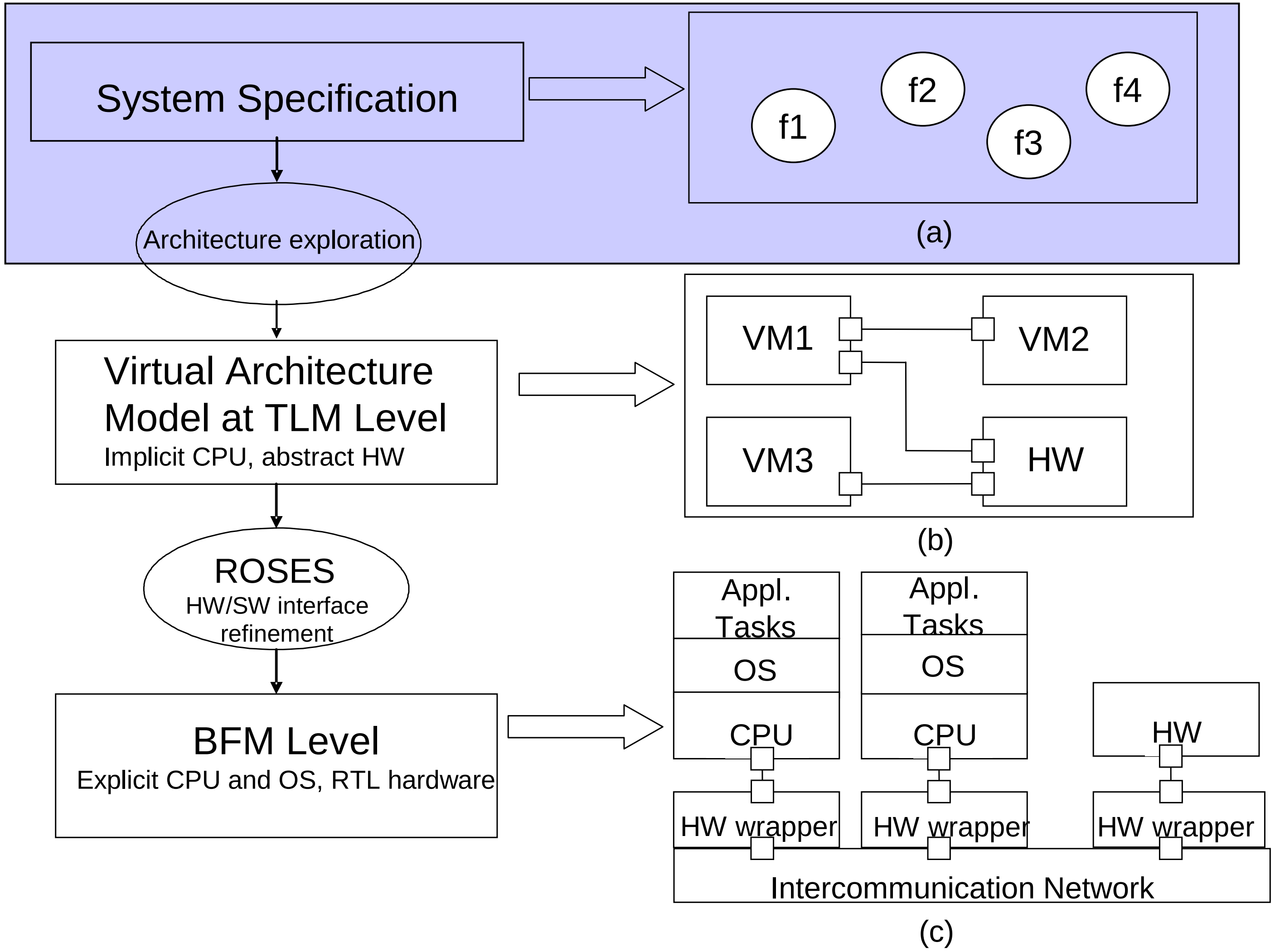
3. Neural network-based performance estimation

4. Virtual prototype-based performance estimation

5. Case study: MPEG4 encoder

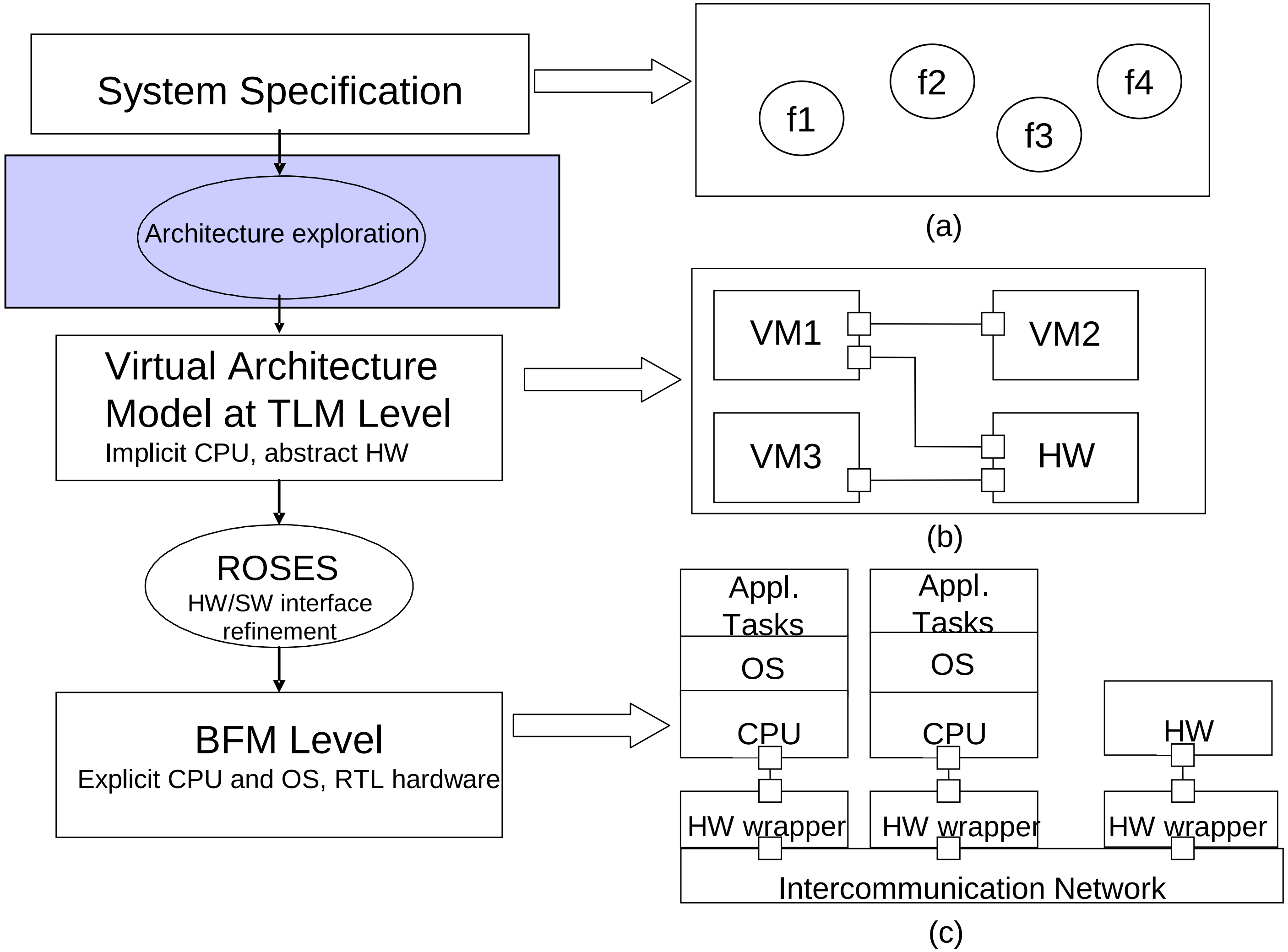
MPSoC Design Flow

- System specification: functional components



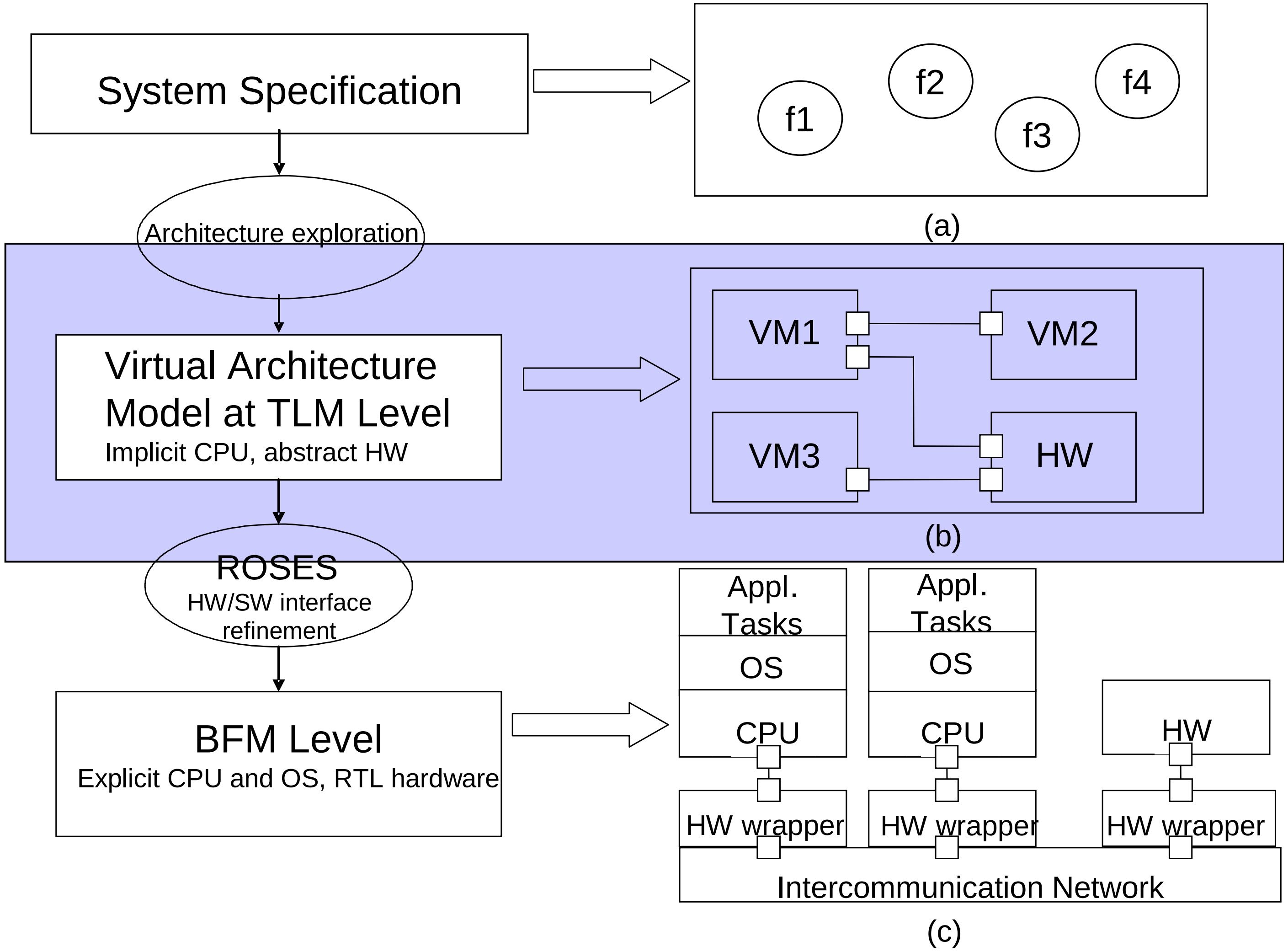
MPSoC Design Flow

- System specification: functional components
- Architecture exploration maps the functionalities in HW and SW components
- Virtual architecture: hardware and software components with abstract communication channels



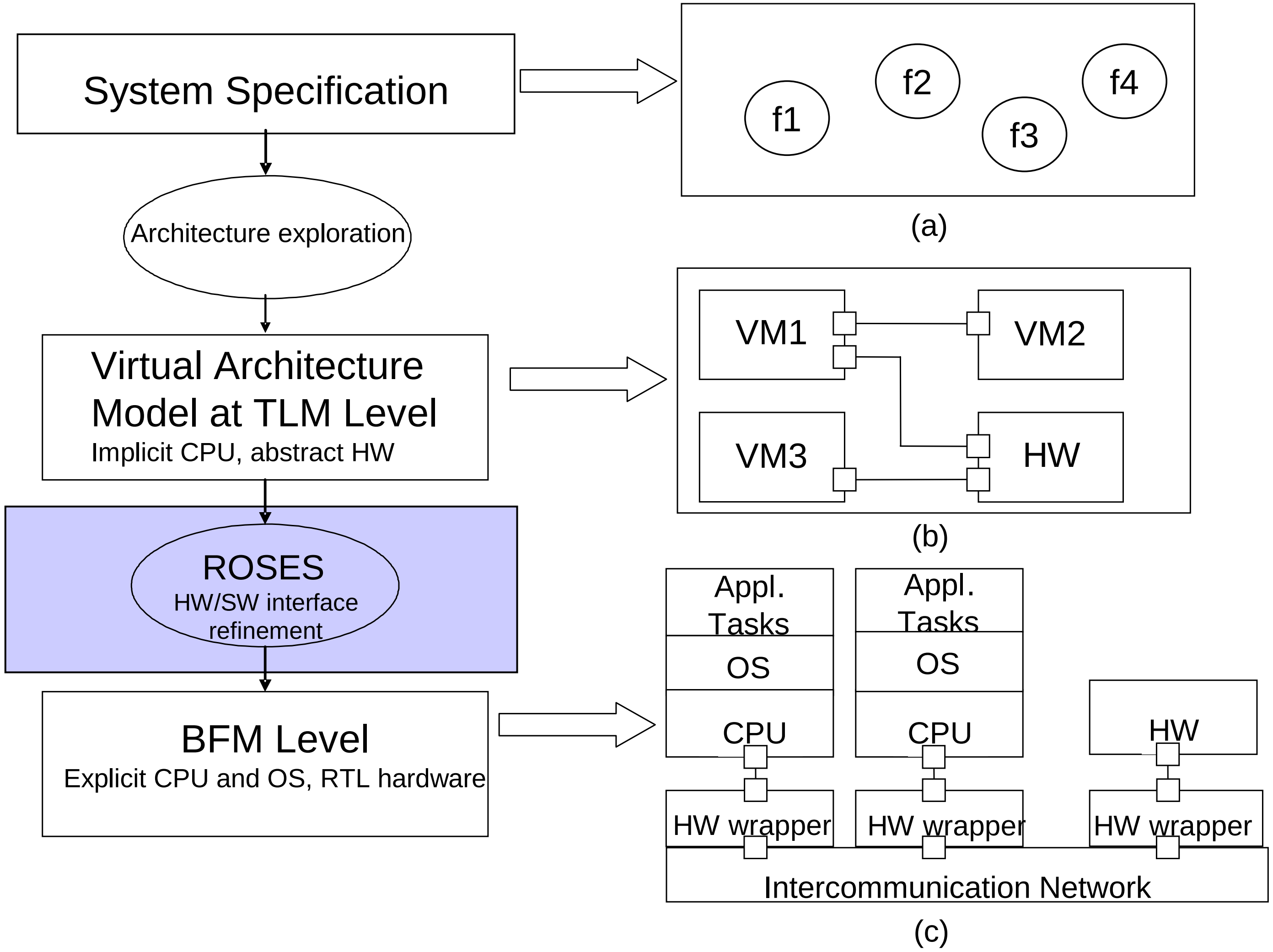
MPSoC Design Flow

- System specification: functional components
- Architecture exploration maps the functionalities in HW and SW components
- Virtual architecture: hardware and software components with abstract communication channels



MPSoC Design Flow

- System specification: functional components
- Architecture exploration maps the functionalities in HW and SW components
- Virtual architecture: hardware and software components with abstract communication channels
- Abstract interfaces are refined in hardware and software interfaces
- BFM Level:
  - CPU
  - Interconnection network and adapters



System Specification

Architecture exploration

Virtual Architecture Model at TLM Level

Implicit CPU, abstract HW

ROSES

HW/SW interface refinement

BFM Level

Explicit CPU and OS, RTL hardware

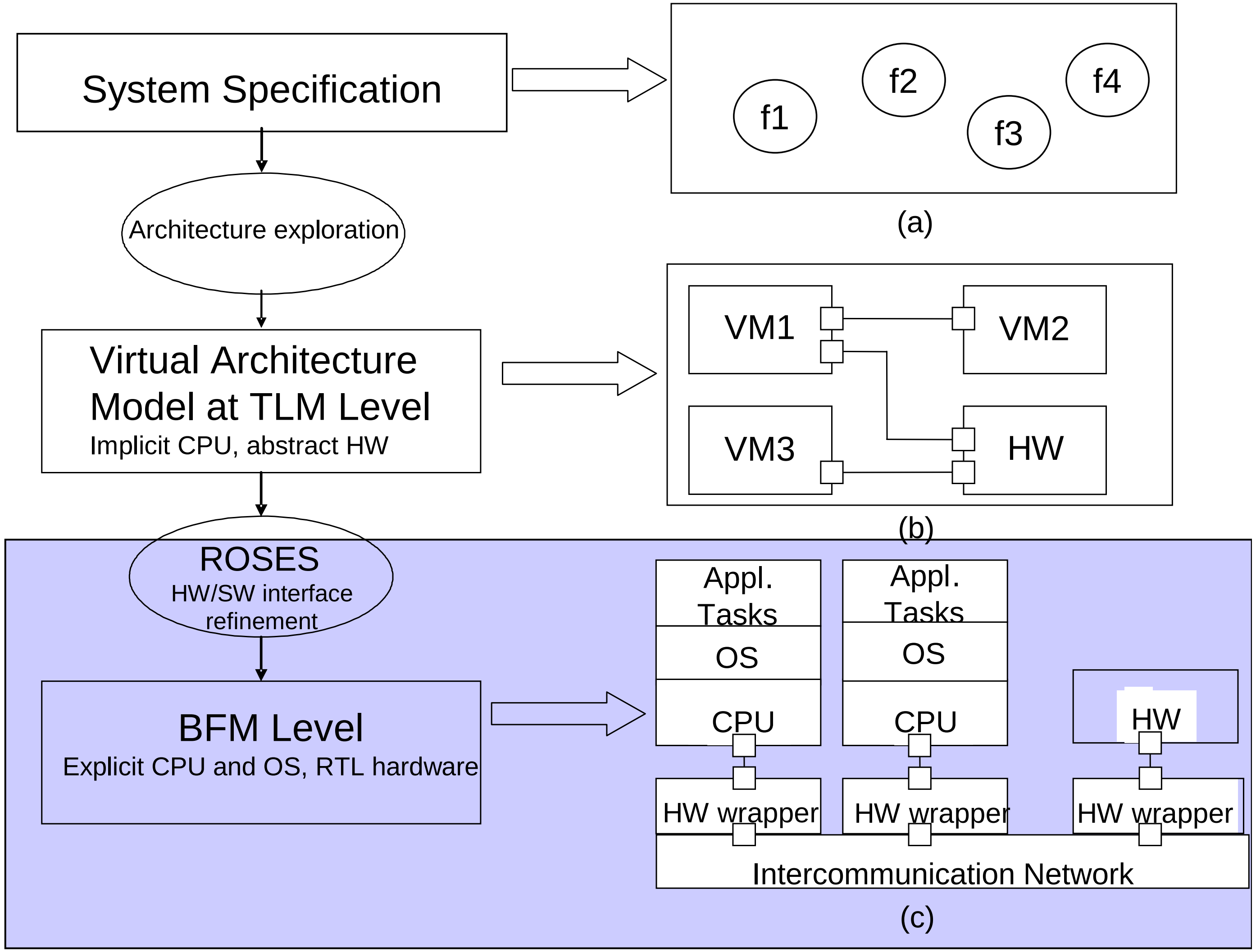
(a)

(b)

(c)

MPSoC Design Flow

- System specification: functional components
- Architecture exploration maps the functionalities in HW and SW components
- Virtual architecture: hardware and software components with abstract communication channels
- Abstract interfaces are refined in hardware and software interfaces
- BFM level:
  - CPU
  - Interconnection network and adapters



Outline

1. MPSoC design flow

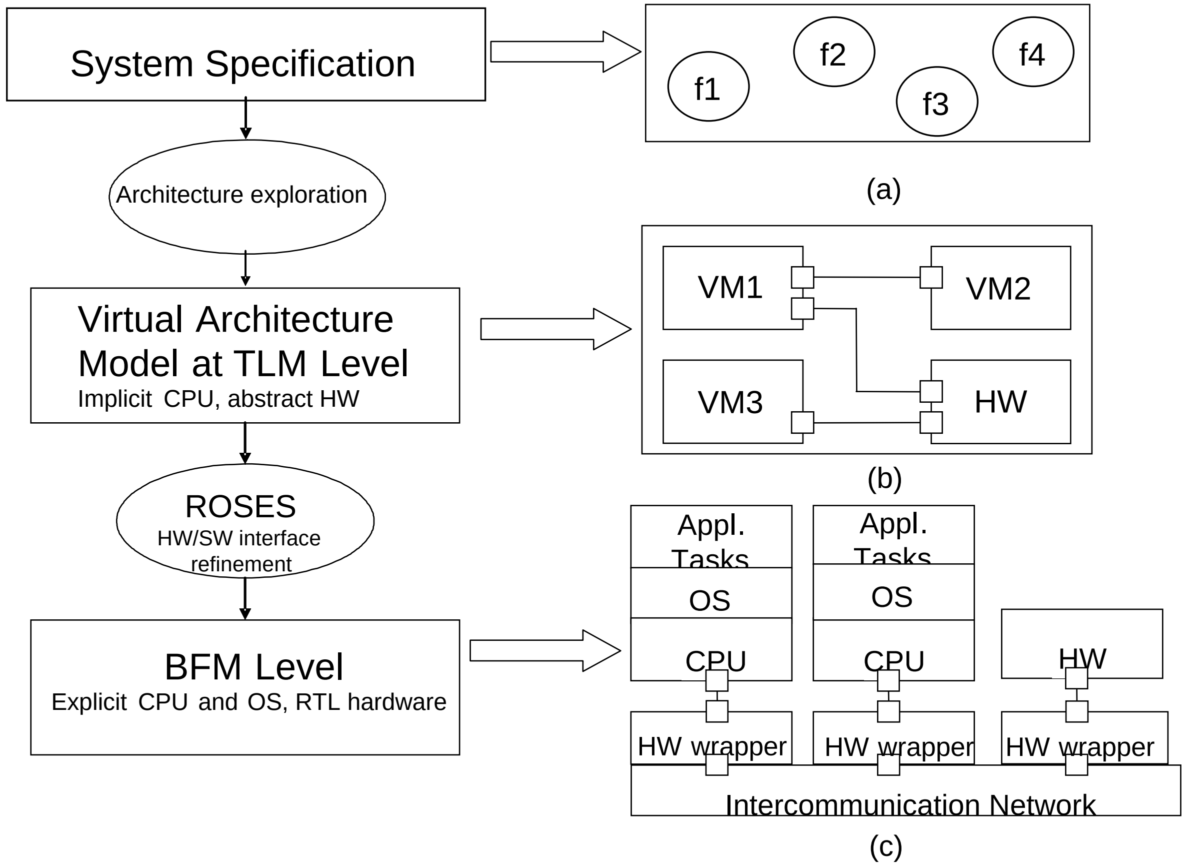
2. Software performance estimation methodology

3. Neural network-based performance estimation

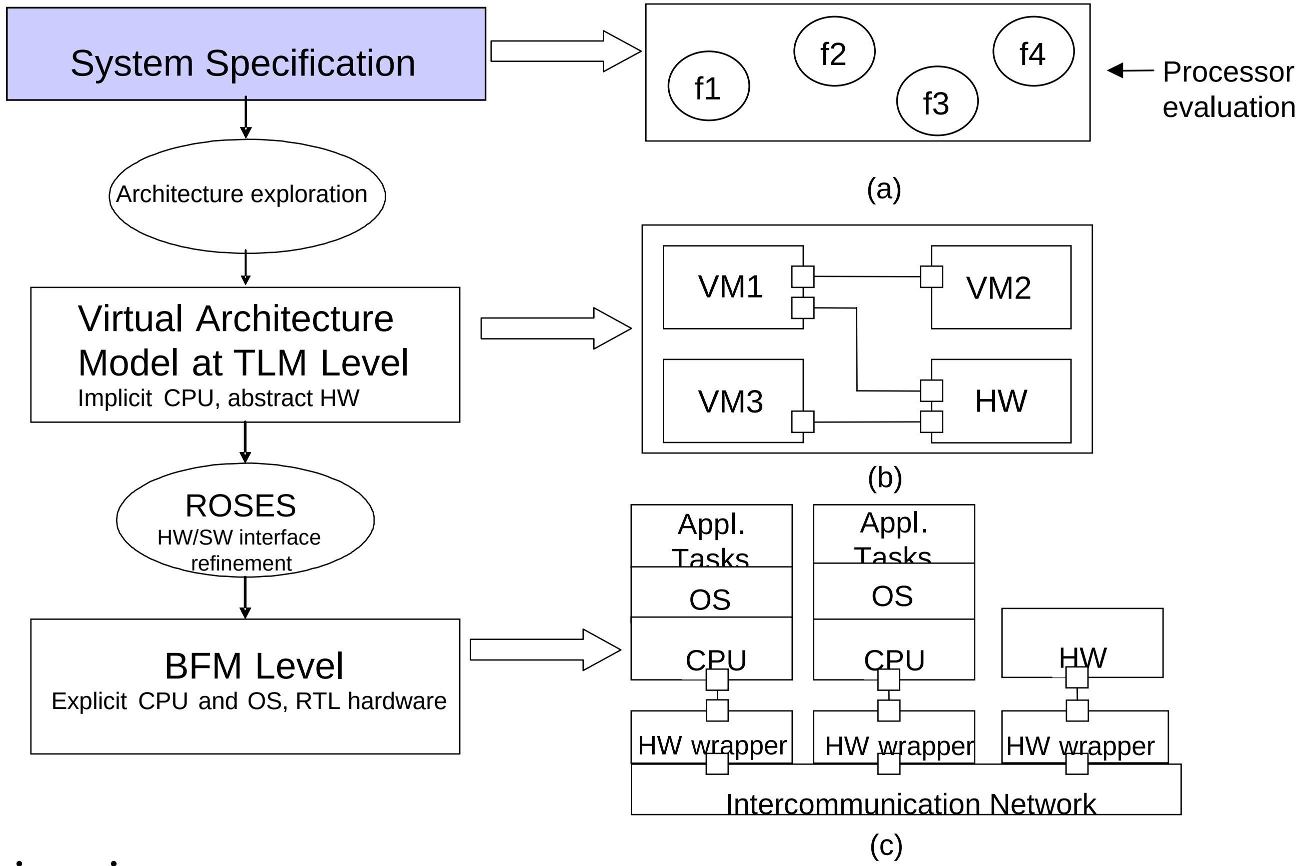
4. Virtual prototype-based performance estimation

5. Case study: MPEG4 encoder

Software Performance Estimation – Integrated with Design Flow

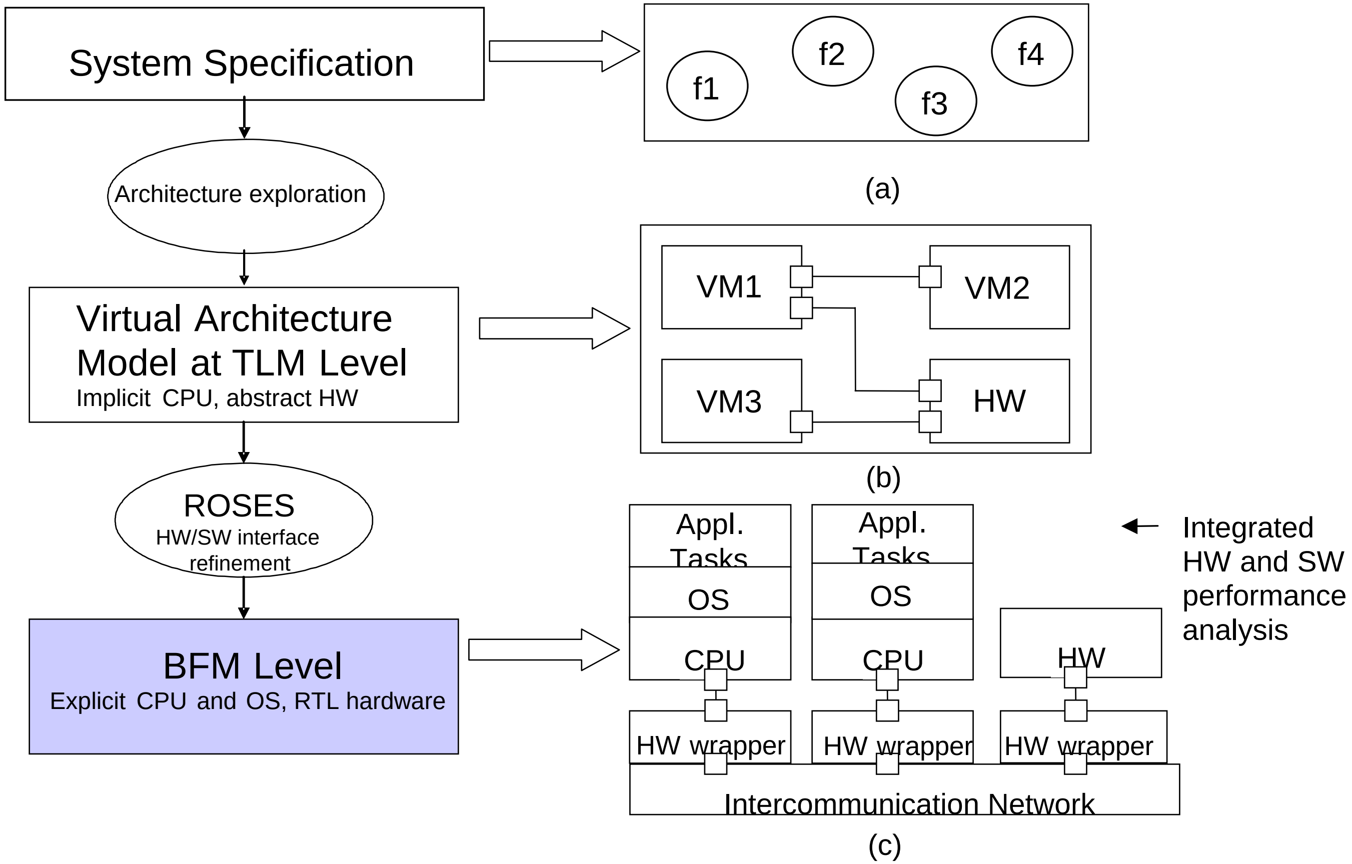


Software Performance Estimation – Processor Evaluation



- SW performance estimation
  - Goal: fast processor evaluation under a given workload
  - Analytical-based, using neural networks
  - High-level, thus some inaccuracies are allowed

Software Performance Estimation – Virtual Prototype


System Specification
f1
f2
f3
f4
(a)
Architecture exploration
Virtual Architecture Model at TLM Level
Implicit CPU, abstract HW
VM1
VM2
VM3
HW
(b)
ROSES
HW/SW interface refinement
BFM Level
Explicit CPU and OS, RTL hardware
Appl. Tasks
OS
CPU
Appl. Tasks
OS
CPU
HW
HW wrapper
HW wrapper
HW wrapper
Intercommunication Network
(c)
Integrated HW and SW performance analysis

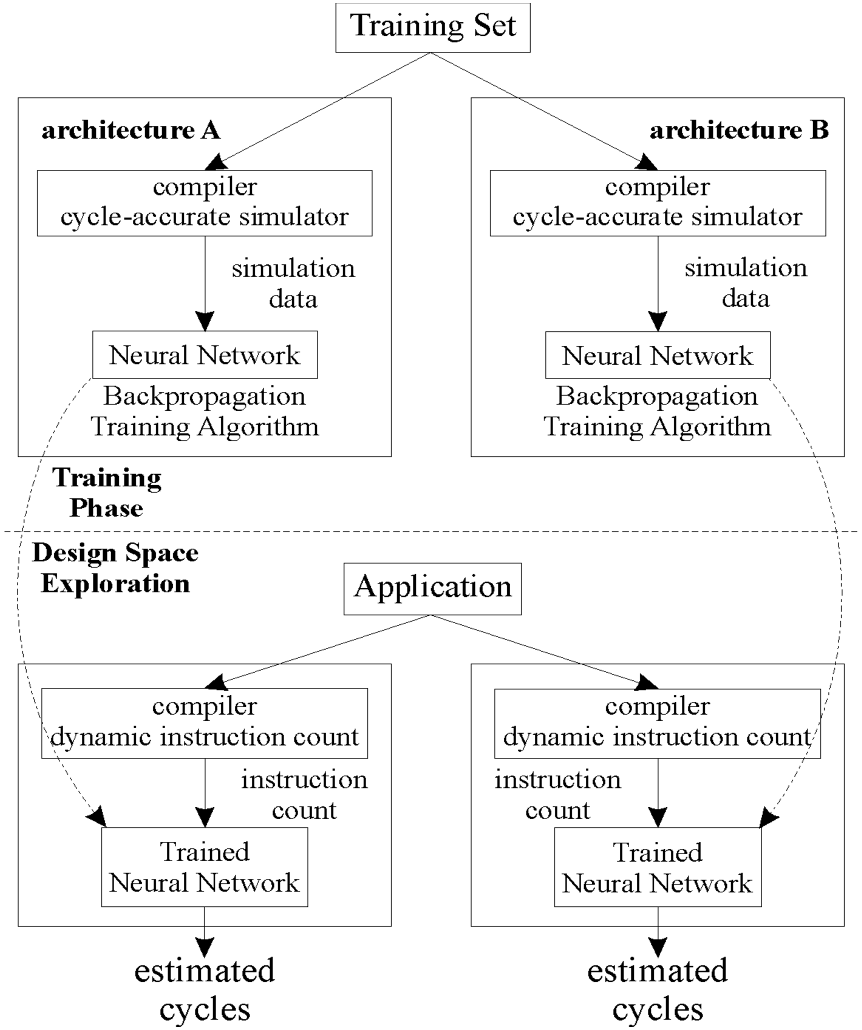
- SW performance estimation using a virtual prototype
  - Simulation-based
  - Detailed analysis of interaction between hardware and software components

Outline

1. MPSoC design flow
2. Performance estimation methodology
3. **Neural network-based performance estimation**
4. Virtual prototype-based performance estimation
5. Case study: MPEG4 encoder

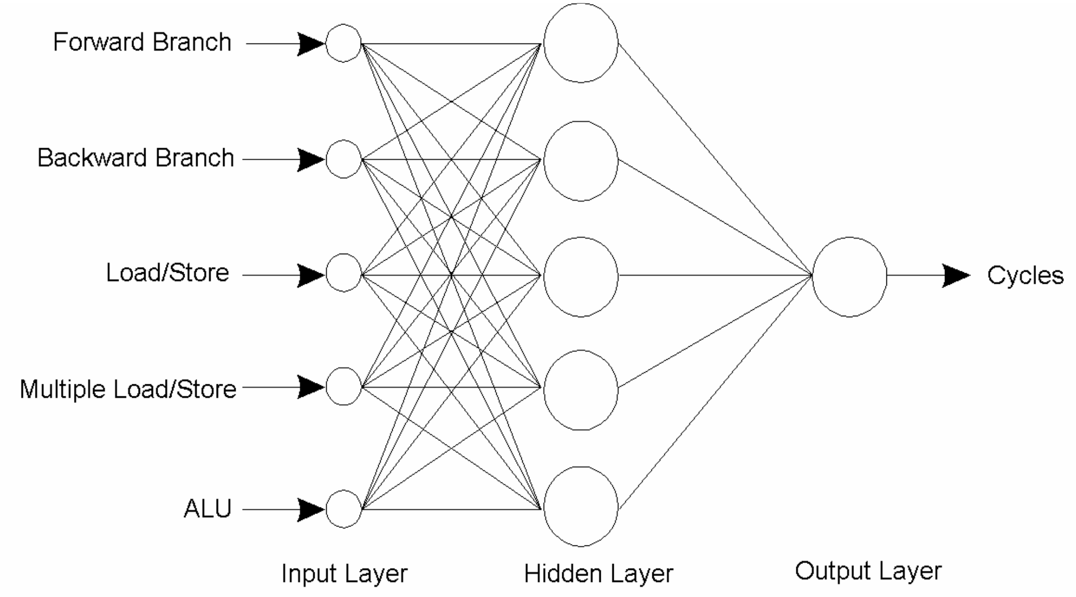
Neural Network Performance Estimation

- Why neural networks?
  - Non-linear prediction – state-of-the-art processors
  - Very fast estimation
- Training phase
  - Set of benchmarks
  - Cycle-accurate simulation: MaxSim ARM9
  - Neural network training and simulation: Matlab
- Utilization phase
  - Dynamic instruction count: instruction-accurate simulator



Neural Network Performance Estimation

- Neural network configuration
  - Input: instruction count
  - Output: # of cycles
  - Input layer and output layer: *linear* transfer function
  - Hidden layer: *tansig* transfer function



- Back-propagation training algorithm

NN Estimation Results for ARM9

- Benchmark set composed by 32 applications and algorithms
  - Total of 41 samples (some benchmarks were executed with different inputs)
  - Different domains
    - Numerical
    - Sort and search algorithms
    - Data processing
    - Synthetic algorithms
  - 20 benchmarks used as training set

	Max underestim	**Max overestim**	**Mean error**	**Std deviation**
All benchmarks	-35.59%	29.75%	9.05%	8.90%
Training set	-18.30%	29.75%	7.62%	7.97%
Test set	-35.59%	11.58%	10.08%	9.54%

NN Estimation Results for ARM9

- Benchmark set composed by 32 applications and algorithms
  - Total of 41 samples (some benchmarks were executed with different inputs)
  - Different domains
    - Numerical
    - Sort and search algorithms
    - Data processing
    - Synthetic algorithms
  - 20 benchmarks used as training set

	Max underestim	**Max overestim**	**Mean error**	**Std deviation**
All benchmarks	-35.59%	29.75%	9.05%	8.90%
Training set	-18.30%	29.75%	7.62%	7.97%
Test set	**-35.59%**	**11.58%**	**10.08%**	**9.54%**

Outline

1. MPSoC design flow

2. Performance estimation methodology

3. Neural network-based performance estimation

4. Virtual prototype-based performance estimation

5. Case study: MPEG4 encoder

Virtual Prototype-based Performance Estimation

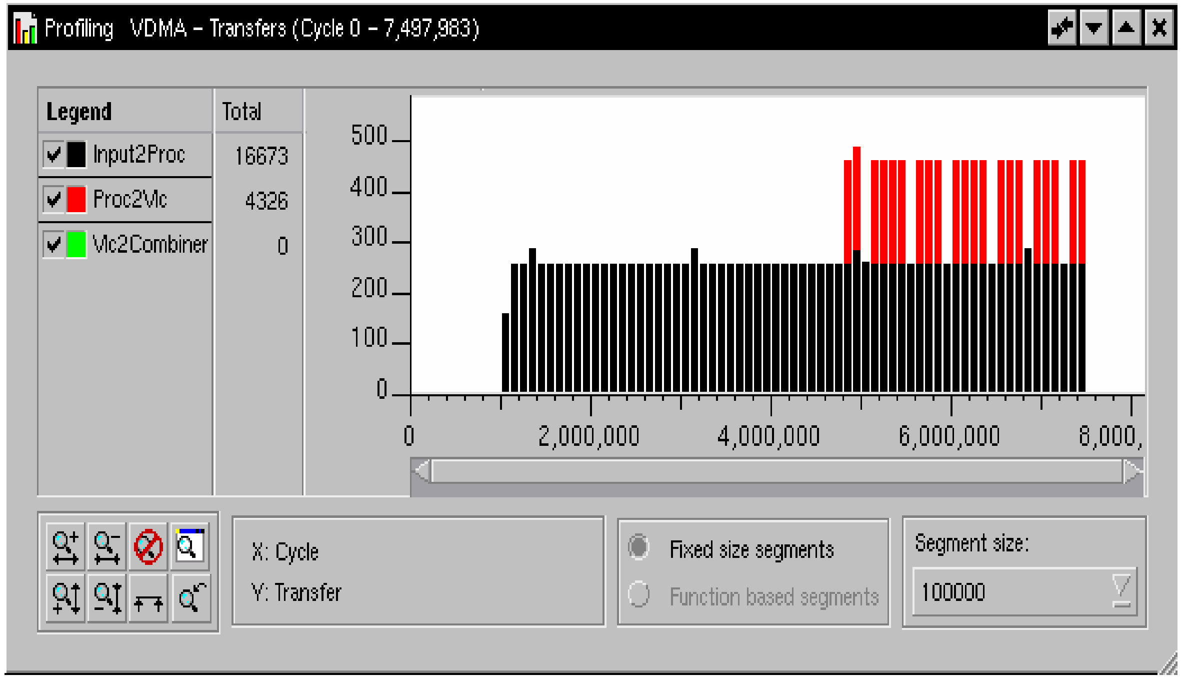
- Virtual prototype for the MaxSim environment, for performance evaluation, is generated from the architecture model
- Processor model: cycle-accurate model provided in the MaxSim library
- Other HW components are provided in SystemC
- Hardware and software simulators run in synchronized way
  - Detection of problems arising from communication between HW and SW components

Virtual Prototype-based Performance Estimation

- Software analysis support by MaxSim
  - Timeline charts for evaluating application functions
  - Cache performance
- Analysis of hardware components
  - Custom profiling of user-defined components

Virtual Prototype – Custom Profiling

- Using the profiling interface, custom analysis is implemented in user-defined components
- Example: Analysis of transfers managed by the DMA component in the MPEG4 case study



Profiling VDMA – Transfers (Cycle 0 – 7,497,983)

Legend	Total
☑ ■ Input2Proc	16673
☑ 🟥 Proc2Vlc	4326
☑ 🟩 Vlc2Combiner	0

X: Cycle
Y: Transfer

● Fixed size segments
○ Function based segments

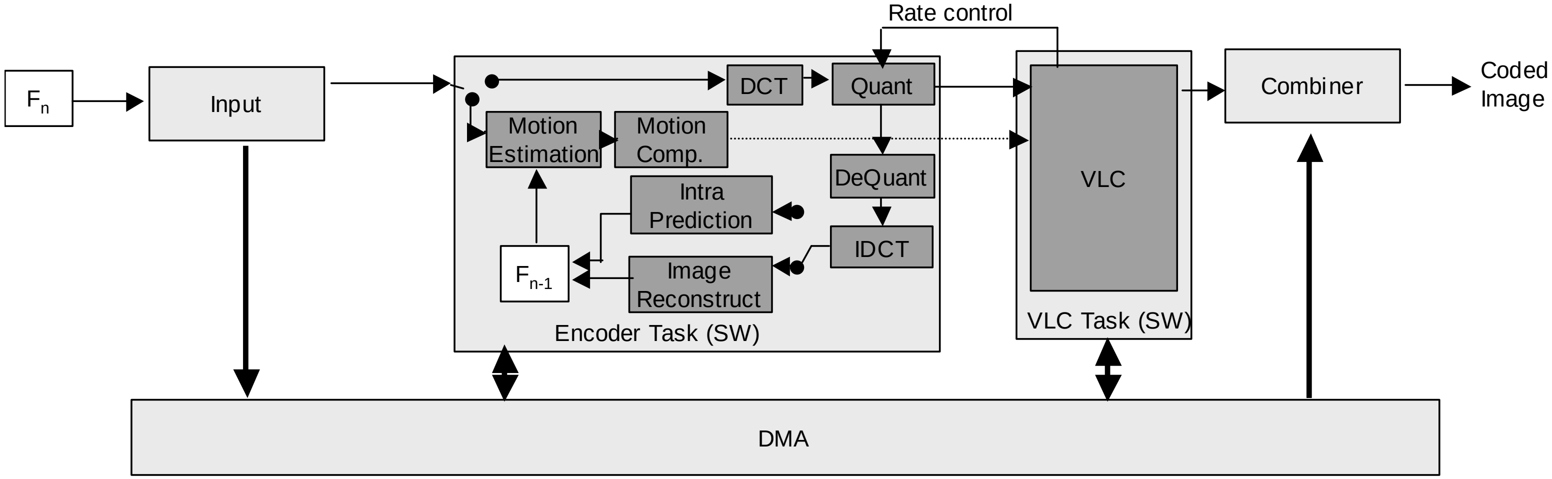
Segment size: 100000

Outline

1. MPSoC design flow
2. Performance estimation methodology
3. Neural network-based performance estimation
4. Virtual prototype-based performance estimation
5. **Case study: MPEG4 encoder**

MPEG4 Encoder

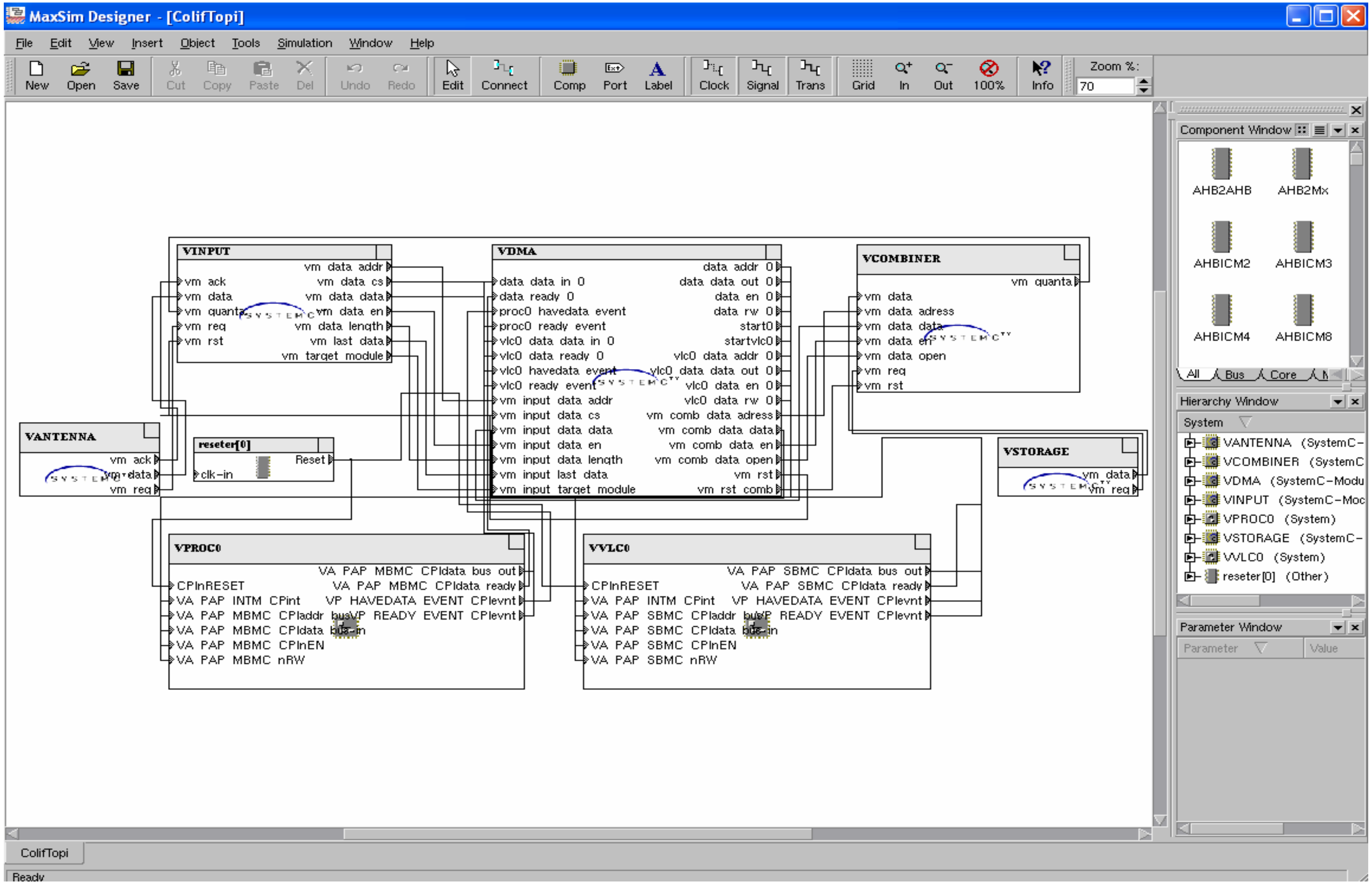
- Two software tasks
  - Encoder Task: core algorithms
  - VLC Task: compression algorithm
- Hardware components
  - DMA, INPUT, COMBINER



MPEG4 Encoder Software Performance Estimation

- At specification level
  - NN estimator used to estimate the software performance
  - Choice of several ARM processor models
  - ARM9 has been selected by using estimation results
- At BFM level
  - Virtual prototype used for detailed SW performance estimation
  - Simulation model uses ARM MaxSim tool
    - CPU: cycle-accurate model
    - Hardware components: RTL models described in SystemC and instrumented with MaxSim profiling interface

Virtual Prototype: MaxSim model



MPEG4 Encoder
NN Estimation Errors

	NN estimation	VP: cycle-accurate	Estimation Error
Encoder Task	122,910 cycles	137,000 cycles	10%
VLC Task	21,613 cycles	26,179 cycles	17%

- Estimation errors have two sources
  - Intrinsic error of the neural network method
  - Communication and OS overheads are neglected
- Communication overhead
  - The NN estimator was trained for a monoprocessor architecture
  - DMA provides point-to-point communication without contention
  - In architectures with shared resources (memories and buses), the contentions could result in a larger error of the NN estimator

NN Estimation Speed-up

- NN network costs
  - NN trained just once, in about 1.5 hours
  - NN utilization
    - Dynamic instruction count using instruction-accurate simulators (much faster than cycle-accurate simulators)
    - NN execution: very fast, just a matrix multiplication
- Virtual prototype: simulation of cycle-accurate CPU + RTL hardware components

Benchmark	Cycle-accurate execution time	Estimation time	Speed-up	Estimation error (%)
Matrix sum	9 sec	0.39 sec	23	3%
LMS filter	12 sec	0.52 sec	23	1%
MPEG encoder	600 sec	17 sec	35	17%

Conclusions

- Integrated MPSoC design and estimation methodology
  - Performance data support the design decisions through the design flow
- Software performance estimation
  - At specification level: processor evaluation using a neural network estimator
    - High-level
    - Fast
  - Virtual prototype
    - After the HW and SW interface refinement
    - Cycle-accurate processor model with instrumented RTL hardware modules
    - Detailed performance analysis
- Offers an interesting trade-off between estimation speed and accuracy
- Case study: MPEG4 encoder
  - Neural network estimation errors up to 17%

Software Performance Estimation in MPSoC Design

Thanks
Questions?

Contact
{marcio, flavio}@inf.ufrgs.br
{ahmed.jerraya, marius.bonaciu}@imag.fr