Managing Hybrid On-chip Scratchpad and Cache Memories for Multi-tasking Embedded Systems

Zimeng Zhou, Lei Ju*, Zhiping Jia, Xin Li

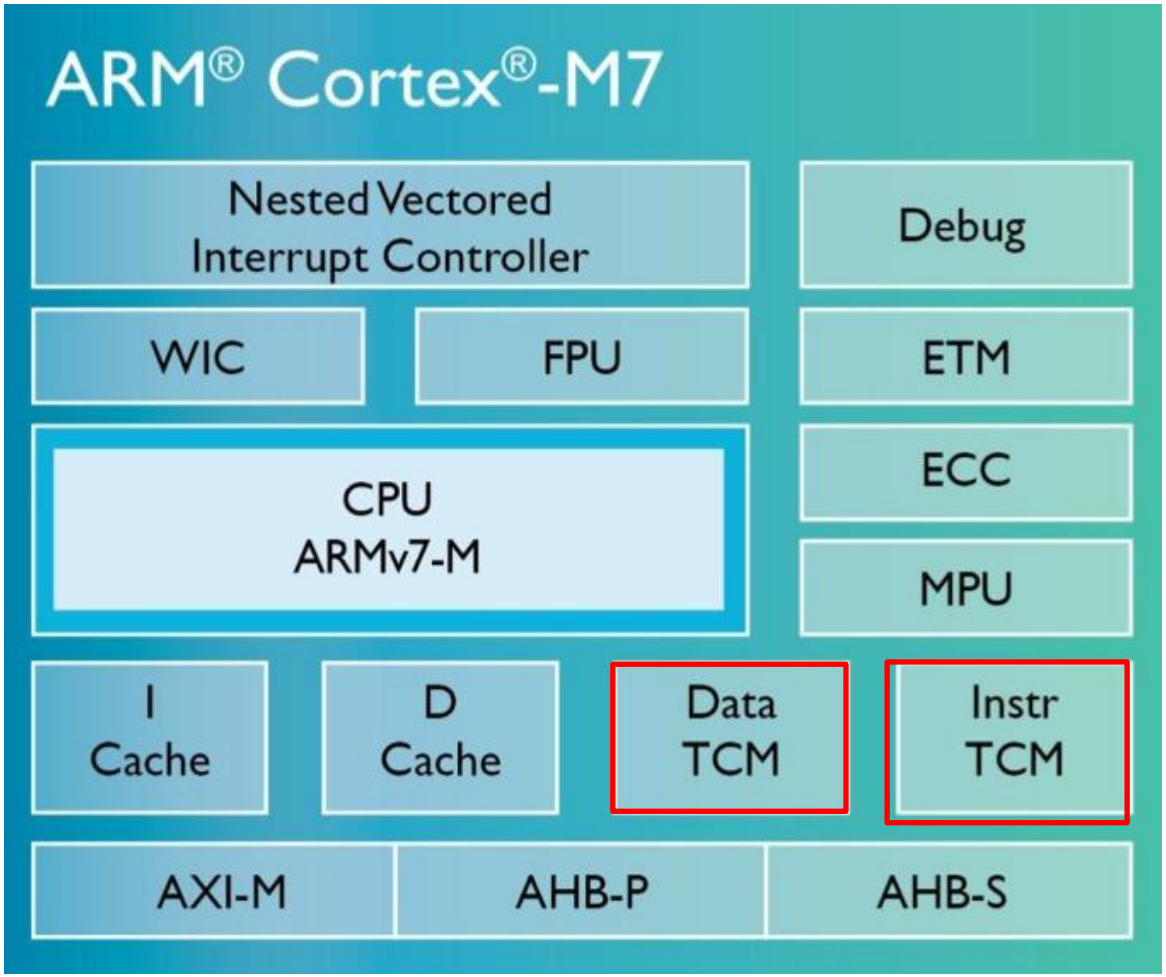
School of Computer Science and Technology
Shandong University, China

Outline

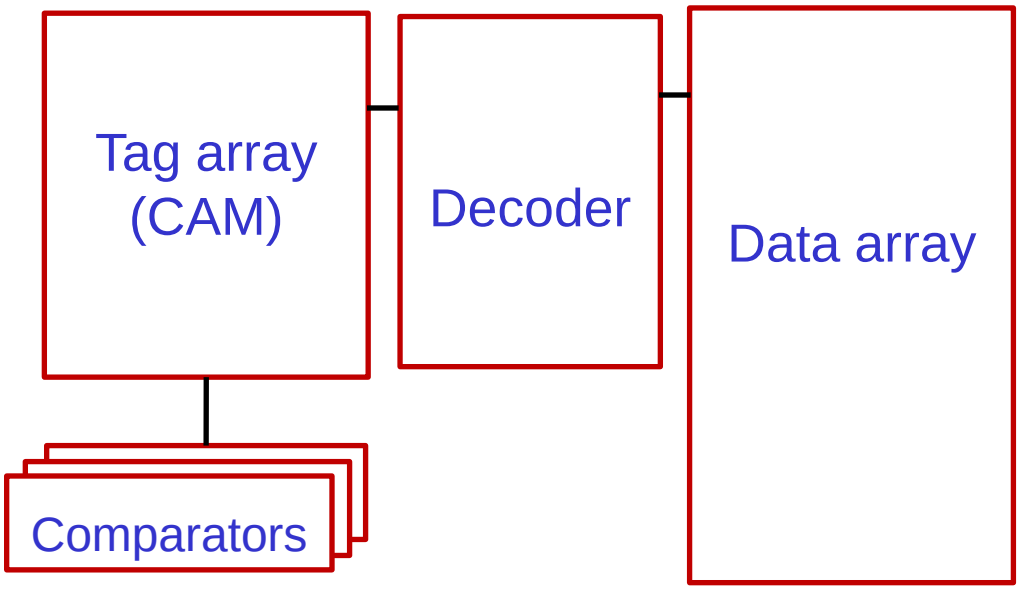
- Introduction
- Motivation
- Methodology
- Experimental Results
- Concluding Remarks

Background

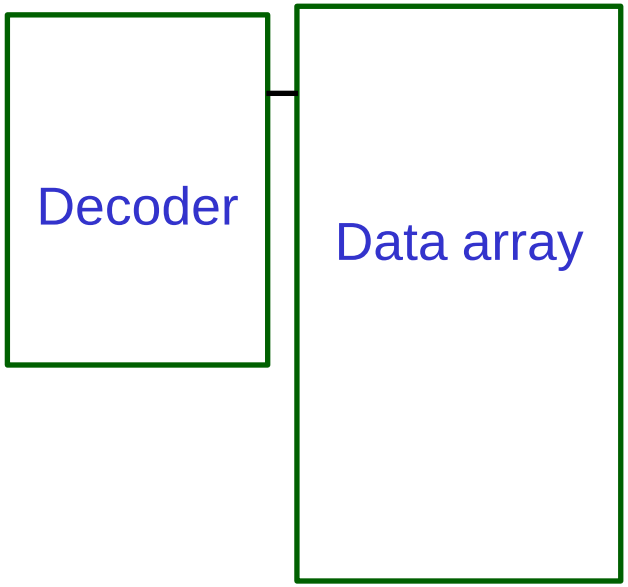
SPMs(scratch-pad memories) are software-controlled on-chip SRAMs (e.g., ARM, CELL, GPU)



Cache vs. SPM



Cache organization



SPM organization

SPM vs. Cache

AREA	• SPM has higher density
ENERGY	• SPM consumes less energy
SPEED	• SPM has slightly faster per word access speed
REAL-TIME	• SPM offers better timing predictability

Moving the design complexity from hardware to system/application software

Research on SPM Allocation

Performance Improvement

- [TECS'06, RTAS'12, DATE'14]

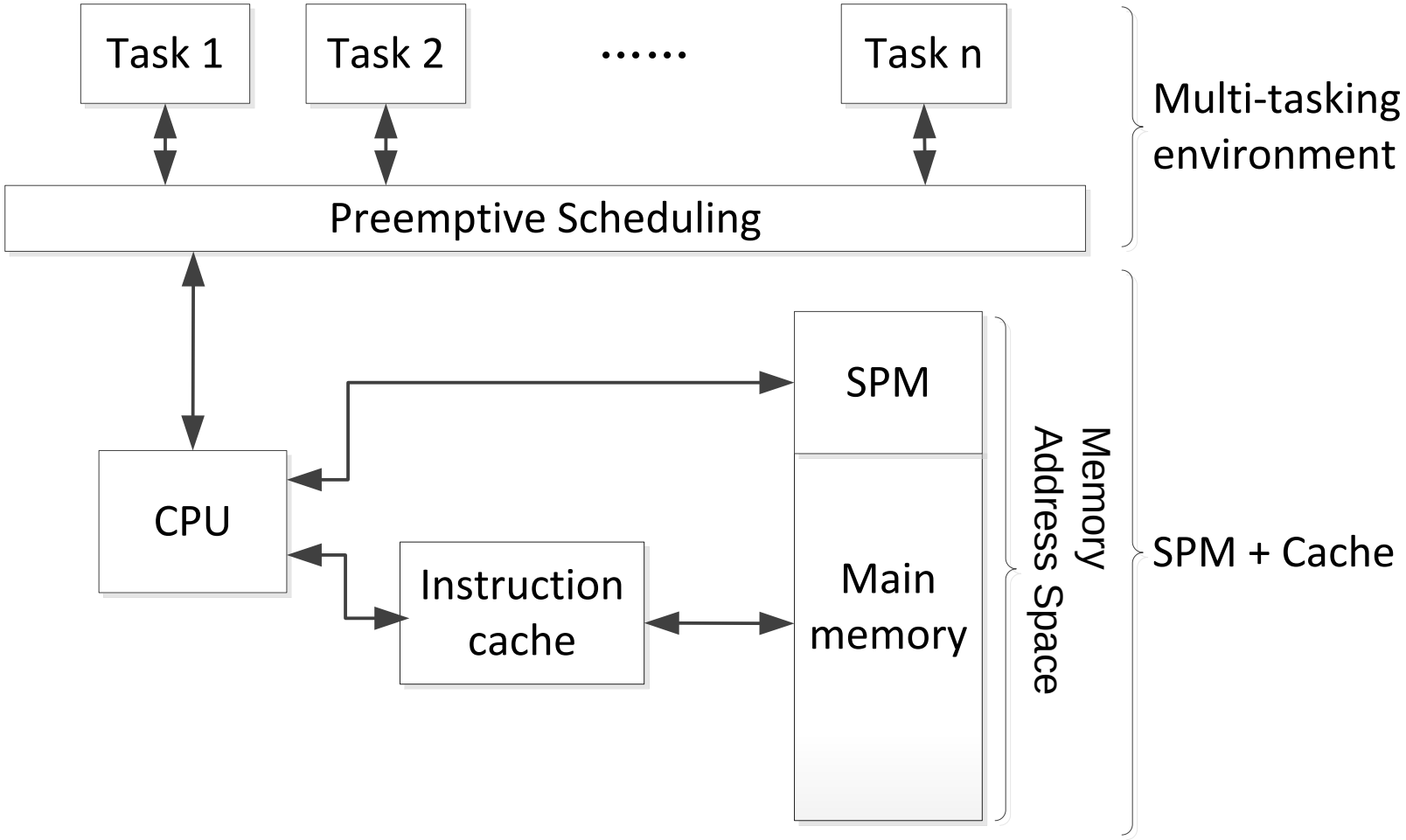
Energy Optimization

- [TCAD'06, DATE'10, ICPP'11, TC'12]

WCET Analysis

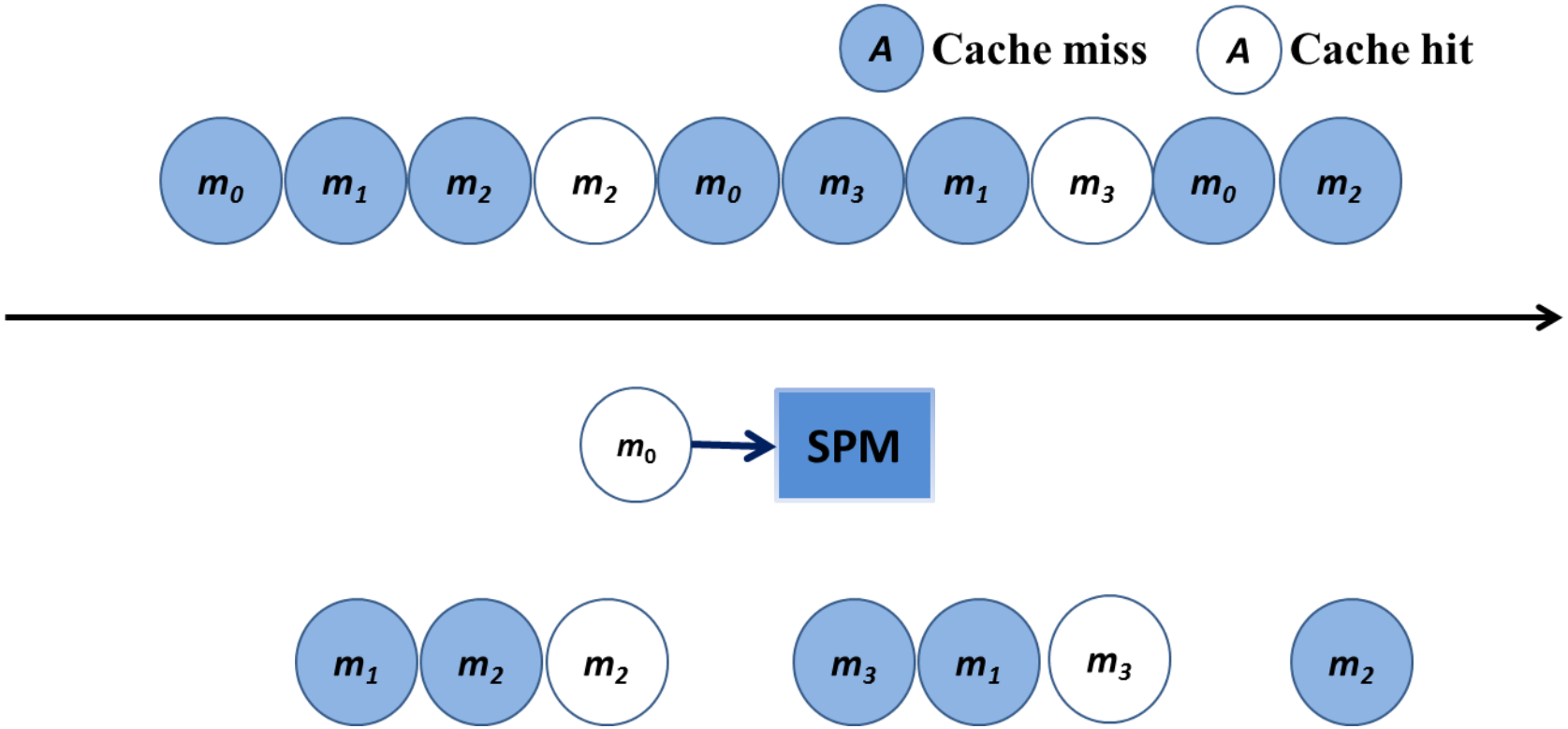
- [TOPLAS'10, LCTES'02, JSA'14]

Target Architecture



Motivation

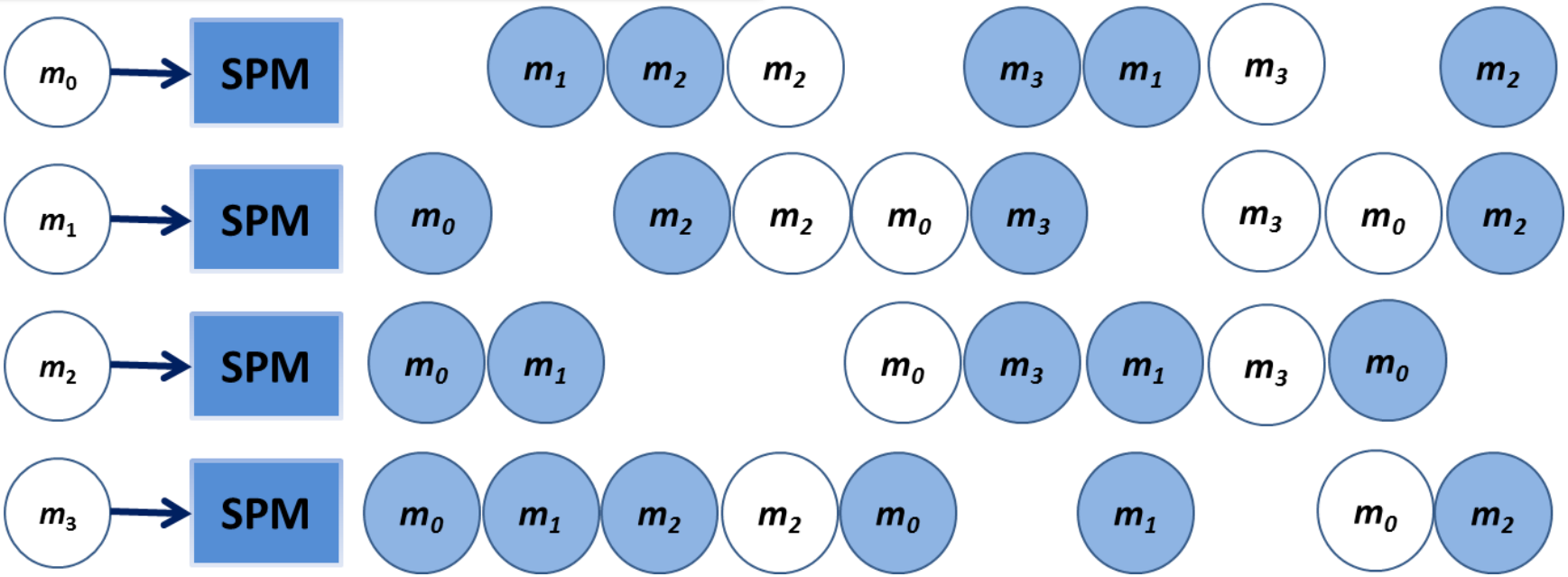
An instruction cache trace of a cache set with 2-way associativity



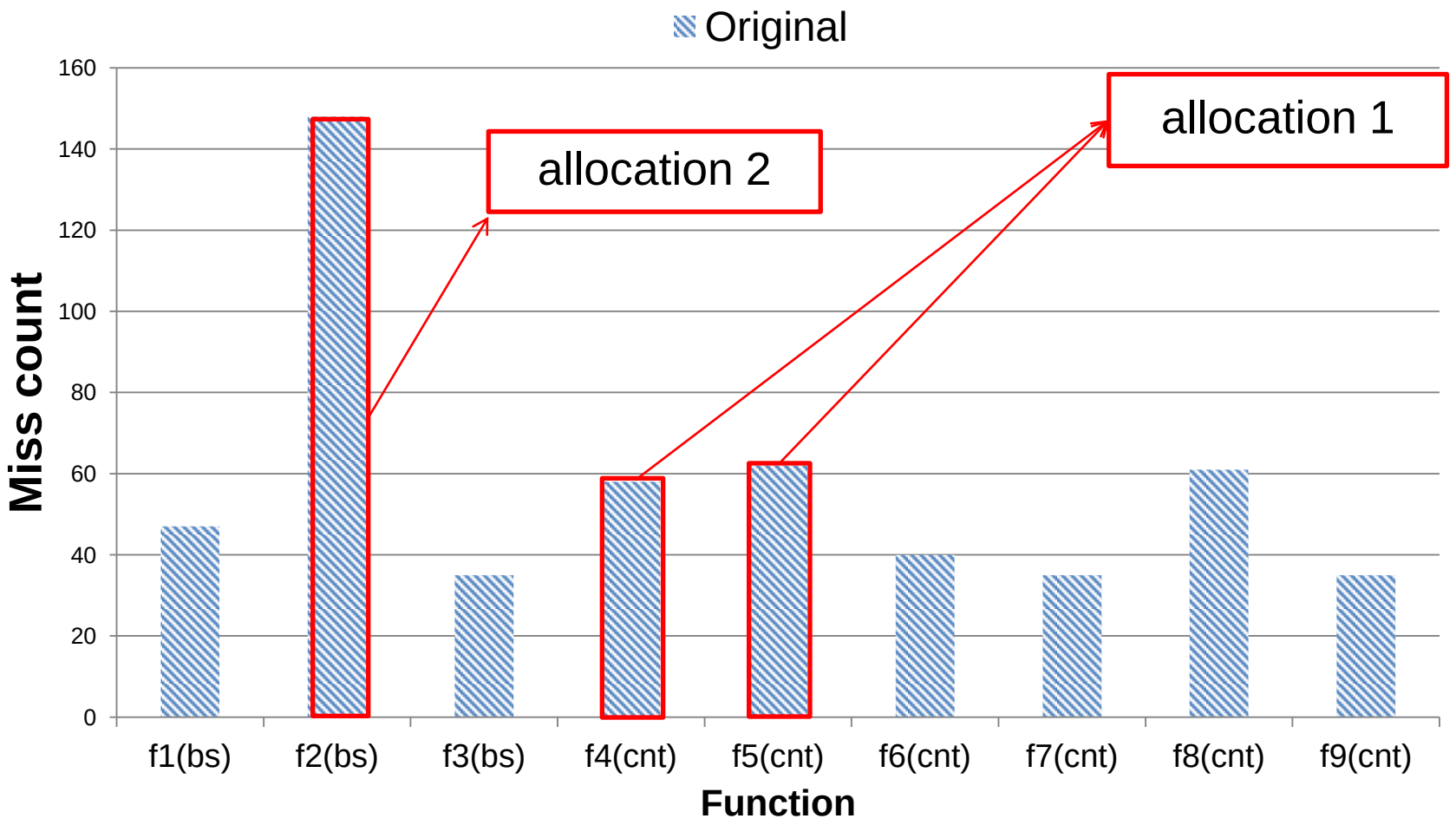
Motivation

	Access priority	Miss priority	Optimal
SPM allocation	m_0 or m_2	m_0	m_1
Miss reduction	3	3	4

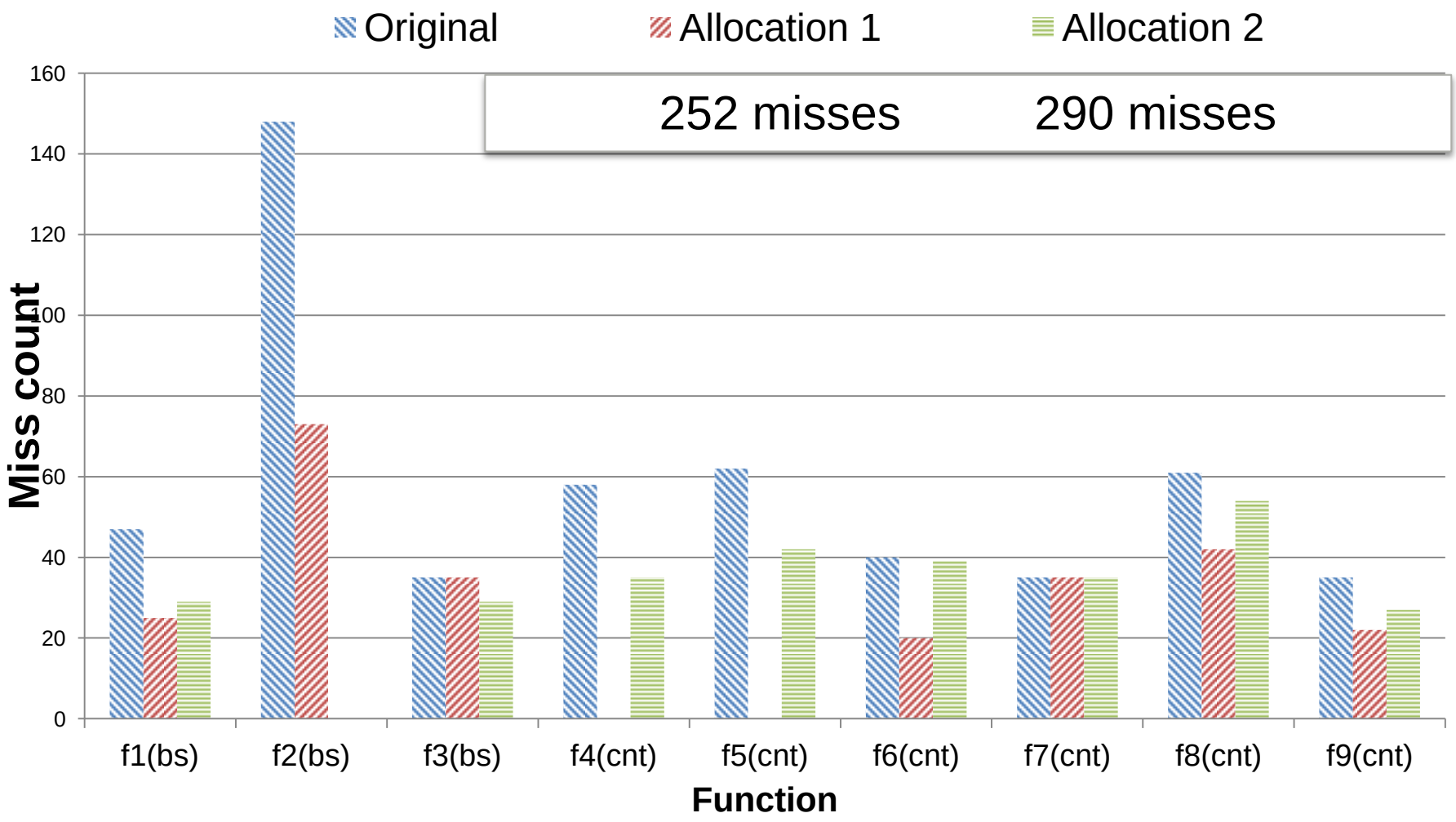
4 Different allocation schemes



Motivation


Original
allocation 1
allocation 2
Miss count
160
140
120
100
80
60
40
20
0
f1(bs)
f2(bs)
f3(bs)
f4(cnt)
f5(cnt)
f6(cnt)
f7(cnt)
f8(cnt)
f9(cnt)
Function

Motivation

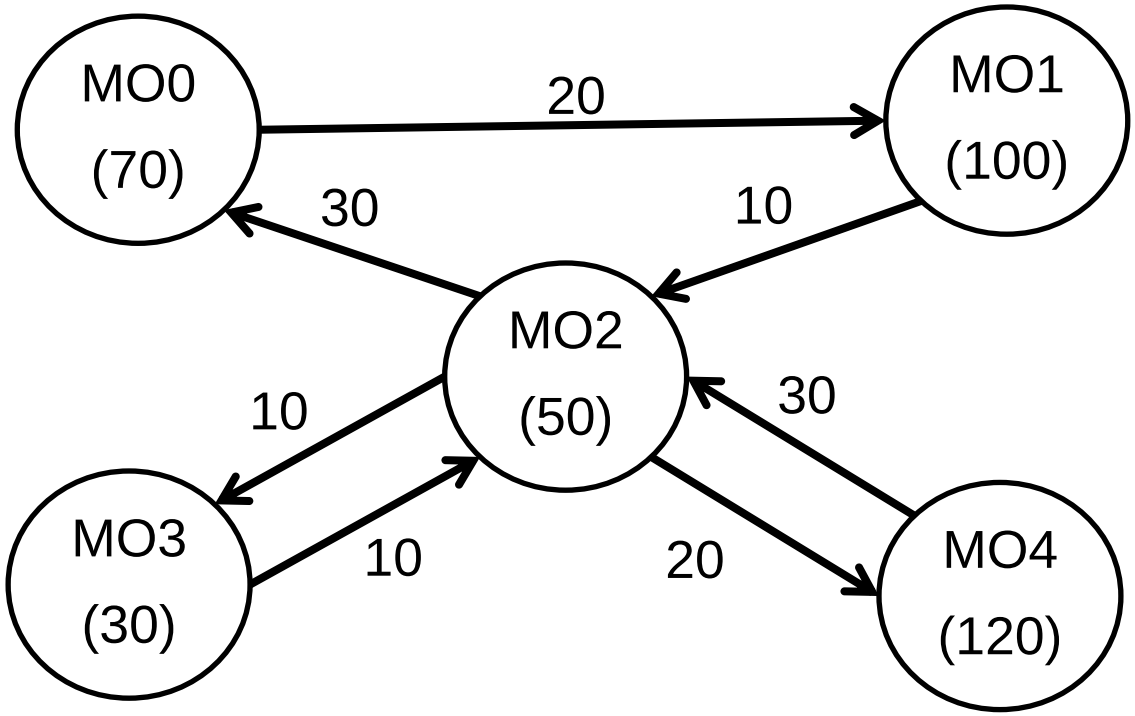

Original
Allocation 1
Allocation 2
252 misses
290 misses
160
140
120
100
80
60
40
20
0
Miss count
f1(bs)
f2(bs)
f3(bs)
f4(cnt)
f5(cnt)
f6(cnt)
f7(cnt)
f8(cnt)
f9(cnt)
Function

Methodology

- Intra- and inter-task cache behavior modeling
- ILP-based allocation strategies
  - Performance optimization
  - Energy optimization

Cache Conflict Graph

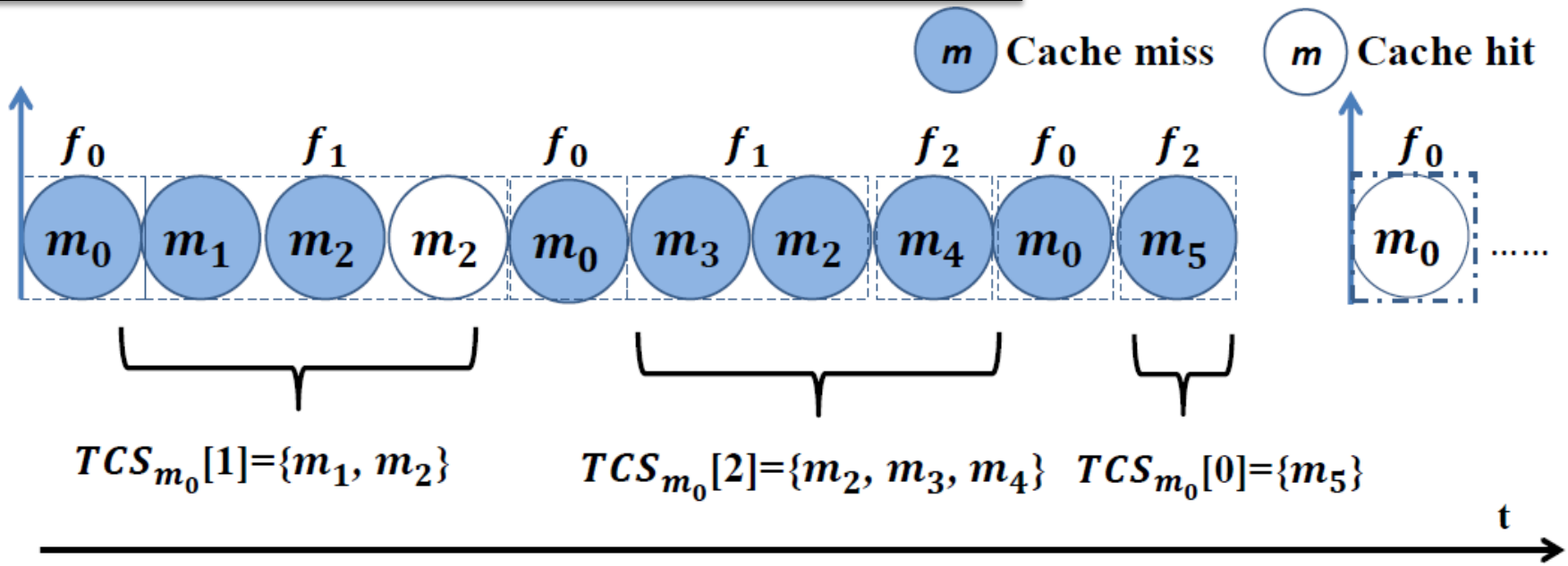
- Adopted to capture the potential gain of SPM allocation in a hybrid SPM-cache architecture
  - [TCAD'06, RTAS'12, TC'12]
  - Coarse-grained, aggregated, pair-wise cache interferences



Temporal Conflict Set

$TCS_{m0}[i]$: set of unique memory blocks referenced between the i-th and (i+1)-th accesses of memory block m_0 in a given trace ([DAC'10])

An instruction cache trace (2-way + LRU):



$$\begin{cases} |TCS| \geq A, & Cache\ miss \\ |TCS| < A, & Cache\ hit \end{cases}$$

0-1 ILP formulation

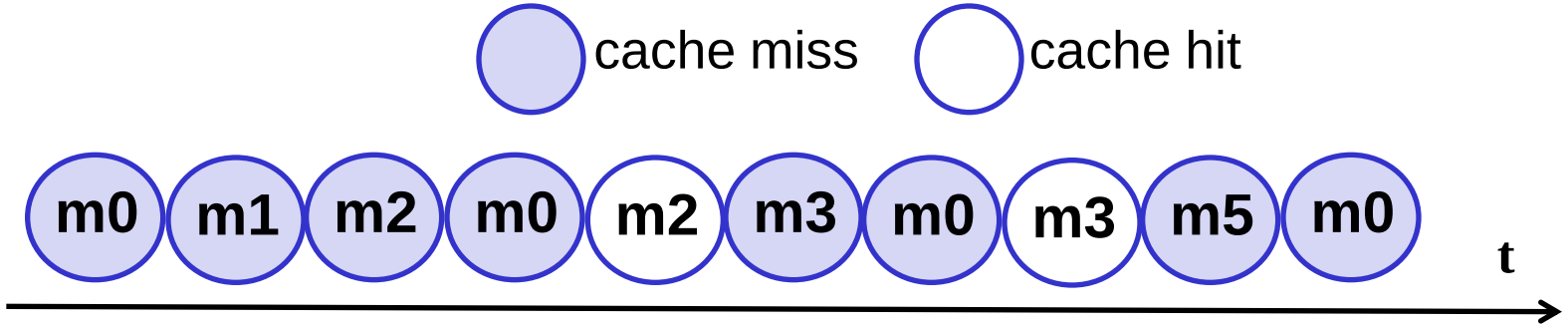
$$x_{f_i} = \begin{cases} 1, & \text{if function } f_i \text{ is allocated to the SPM,} \\ 0, & otherwise. \end{cases}$$

Subject to SPM capacity:

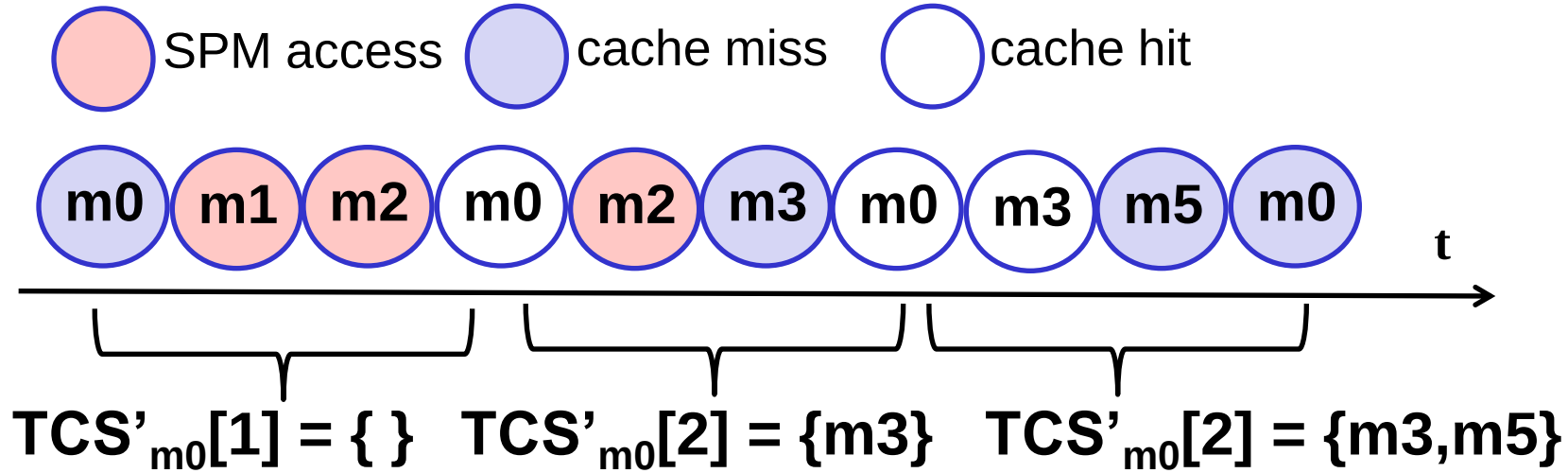
$$\sum_{i=1}^{V} (size_{f_i} \cdot x_{f_i}) \leq SIZE_S$$

Intra-task Cache Interference

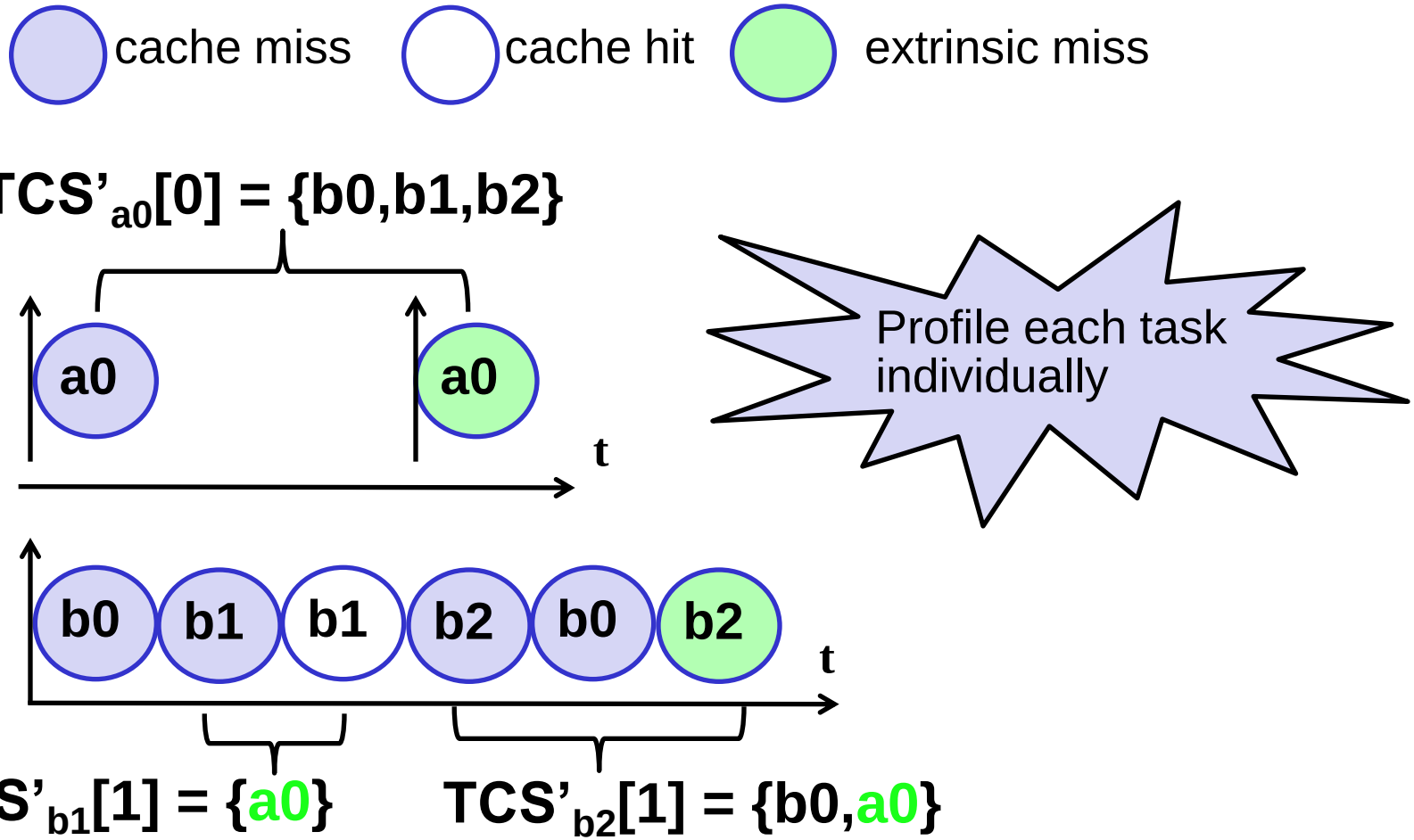
2-way + LRU



After allocate function f1 (with memory blocks m1,m2) into SPM



Inter-task Cache interference



Probability of b2[1] being a extrinsic miss = P_{t1}/P_{t2}

Cache miss count

$$
\begin{aligned}
miss_j &= miss'_j + hmiss'_j + lmiss'_j \\
num_m(m_j) &= miss_j \cdot (1 - x_{fun(m_j)}) \\
num_c(m_j) &= (access_j - miss_j) \cdot (1 - x_{fun(m_j)}) \\
num_s(m_j) &= access_j \cdot x_{fun(m_j)}
\end{aligned}
$$

Optimization Goals

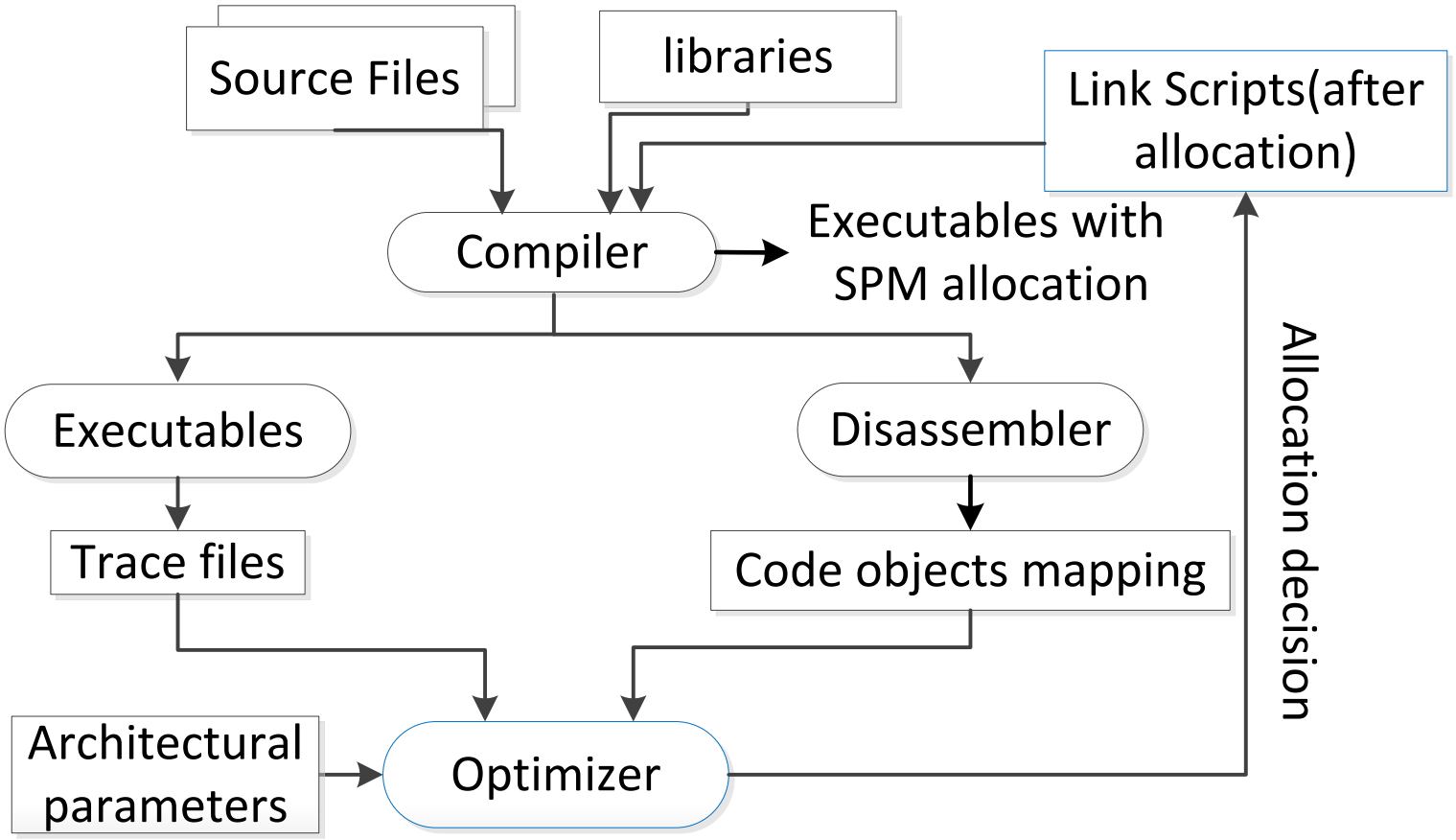
Performance Optimization

$$lat_{access} = \sum_{\forall f_i} \sum_{m_j \in f_i} \frac{hperiod}{pd_{tk(m_j)}} (num_m(m_j) \cdot lat_m + num_c(m_j) \cdot lat_c + num_s(m_j) \cdot lat_s)$$

Energy Optimization

$$E_{access} = \sum_{\forall f_i} \sum_{m_j \in f_i} \frac{hperiod}{pd_{tk(m_j)})} (num_m(m_j) \cdot E_m + num_c(m_j) \cdot E_c + num_s(m_j) \cdot E_s)$$

Framework


Source Files
libraries
Link Scripts(after allocation)
Compiler
Executables with SPM allocation
Executables
Disassembler
Trace files
Code objects mapping
Architectural parameters
Optimizer
Allocation decision

Task sets

Task set	# of tasks	Tasks description	size(kByte)
Set1	3	bsort100, fft1, insertsort	12.9
Set2	4	bs, cnt, fft1, insertsort	17.3
Set3	4	bcnt, bsort100, cnt, qurt	14.4
Set4	3	qsort, bcnt, qurt	13.3
Set5	4	bcnt, bs, cnt, qurt	15.9
Set6	6	bsort100, fft1, insertsort, bcnt, qurt, cnt	24.6

Individual application tasks are taken from WCET and Powerstone benchmarks

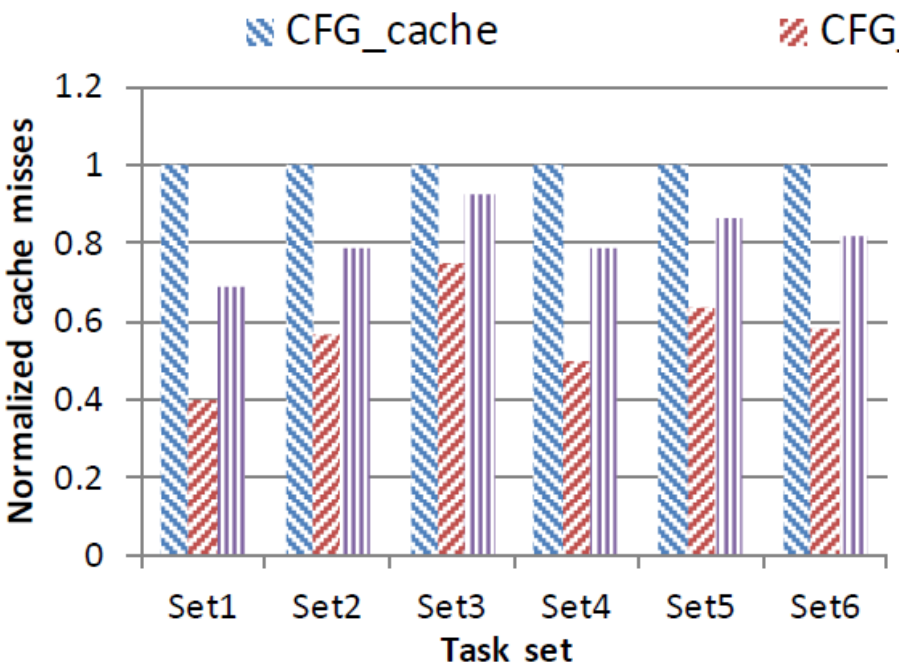
Simulation Parameters

Parameters	Access latency [TECS'06]	Energy consumption [TCAD'06]
2KB SPM	1	1.07nJ
2KB cache	1	4.04nJ
4KB cache	1	4.71nJ
500MB SDRAM	20	49.3nJ

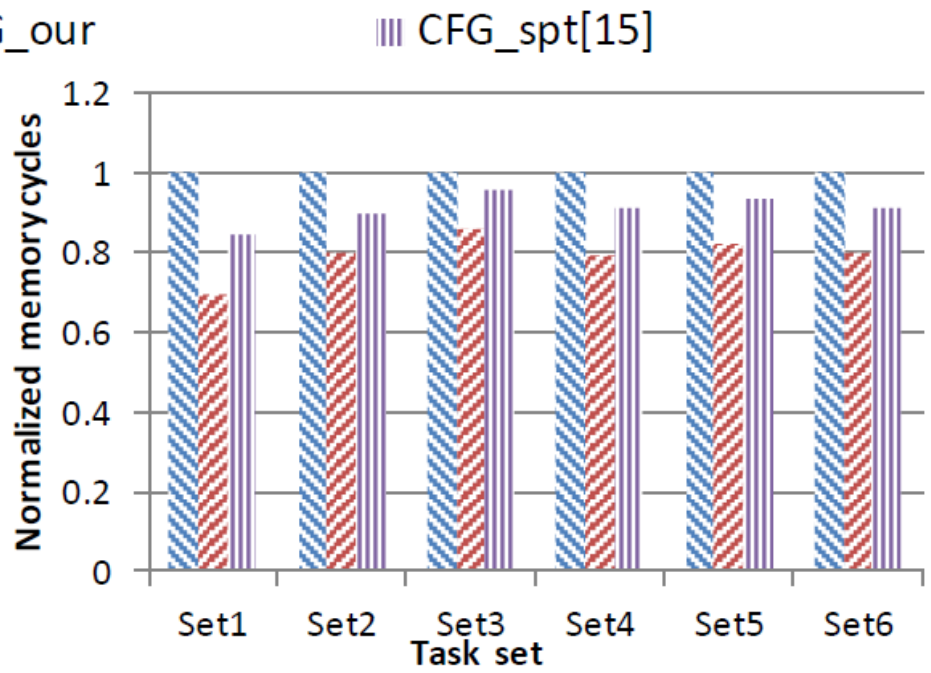
Experiment

- Three configurations：
  - CFG_cache (4K cache)
  - CFG_our (2K cache + 2K SPM)
  - CFG_spt (2K cache + 2K SPM)
    - spatial-based static allocation as in [Takase et al. DATE'10]

Performance Optimization

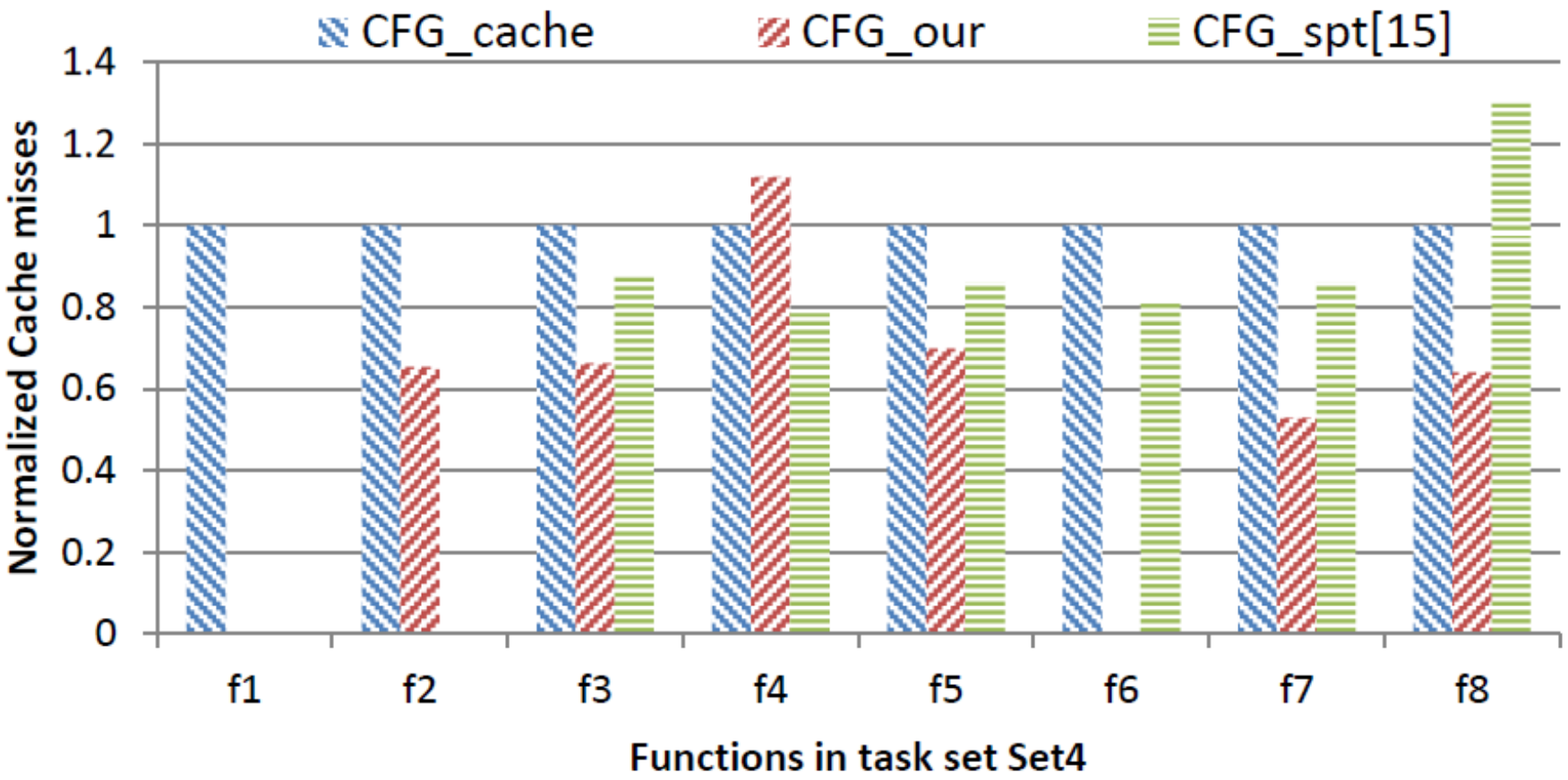


(a) Total cache misses

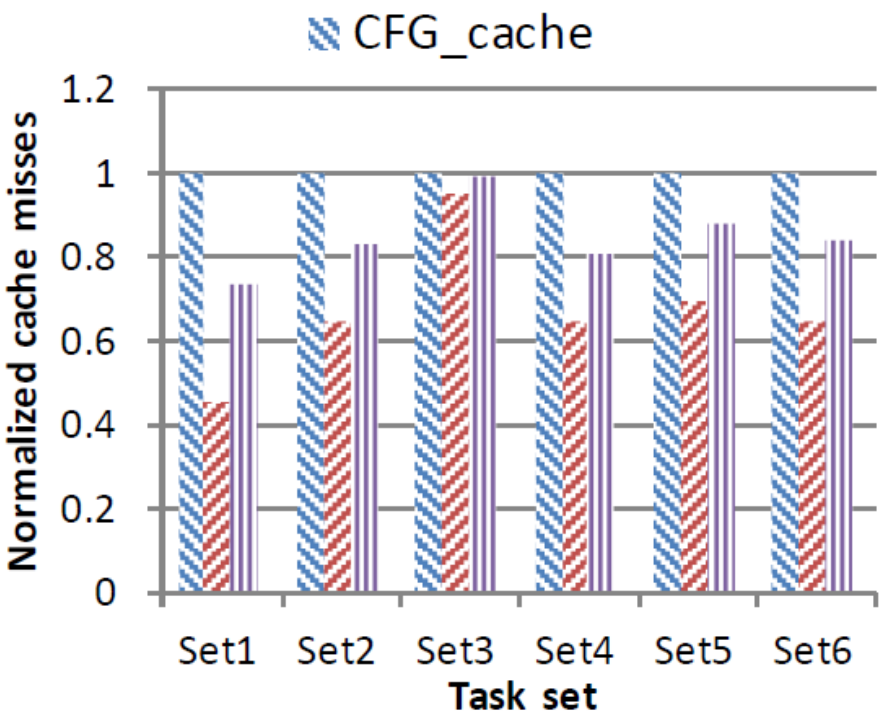


(b) Overall memory cycles

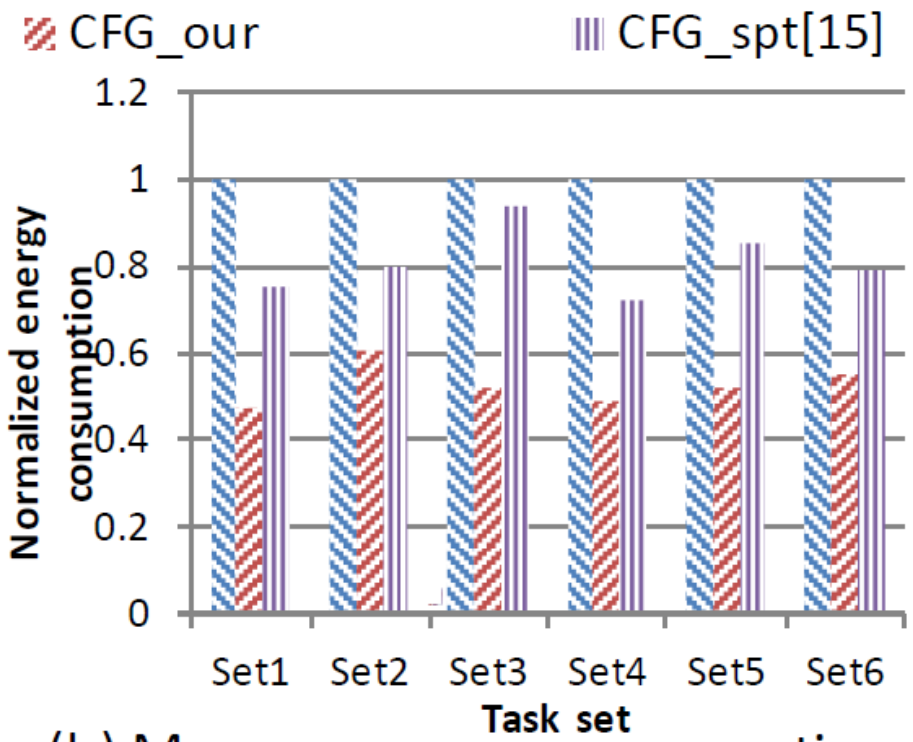
Performance Optimization



Energy Optimization



(a) Total cache misses



(b) Memory energy consumption

Concluding Remarks

- In this work, we have studied the static SPM allocation problem
  - For a hybrid on-chip SPM-cache architecture and multitasking environment
  - A fine-grained temporal cache behavior model captures the SPM-cache synergy
- Future work
  - Heuristic SPM allocation algorithms to support fast design space exploration
  - Sophisticated inter-task cache behavior modeling

Thank you !

Q&A