Tutorial Outline

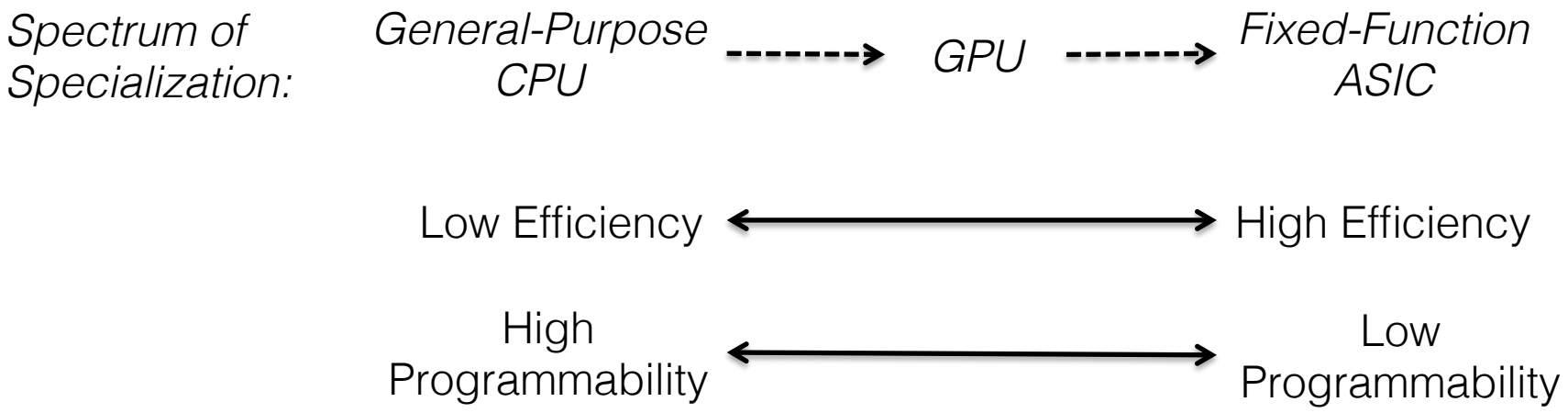
Time	Topic
8:30 am – 9:00 am	Introduction
9:00 am – 10:00 am	Pre-RTL Simulation Framework: Aladdin
10:00 am – 10:30 am	Break
10:30 am – 11:00 am	**Workload Characterization Tool: WIICA**
11:00 am – 12:00 pm	CAD & Benchmarks: HLS & MachSuite
12:00 pm – 2:00 pm	Lunch
2:00 pm – 3:00 pm	Embedded Keynote Talk: Mark Horowitz (Stanford)
3:00 pm – 3:30 pm	Accelerator Selection Tool: Sigil
3:30 pm – 4:00 pm	Break
4:00 pm – 5:00 pm	Hands-on Exercise

WIICA

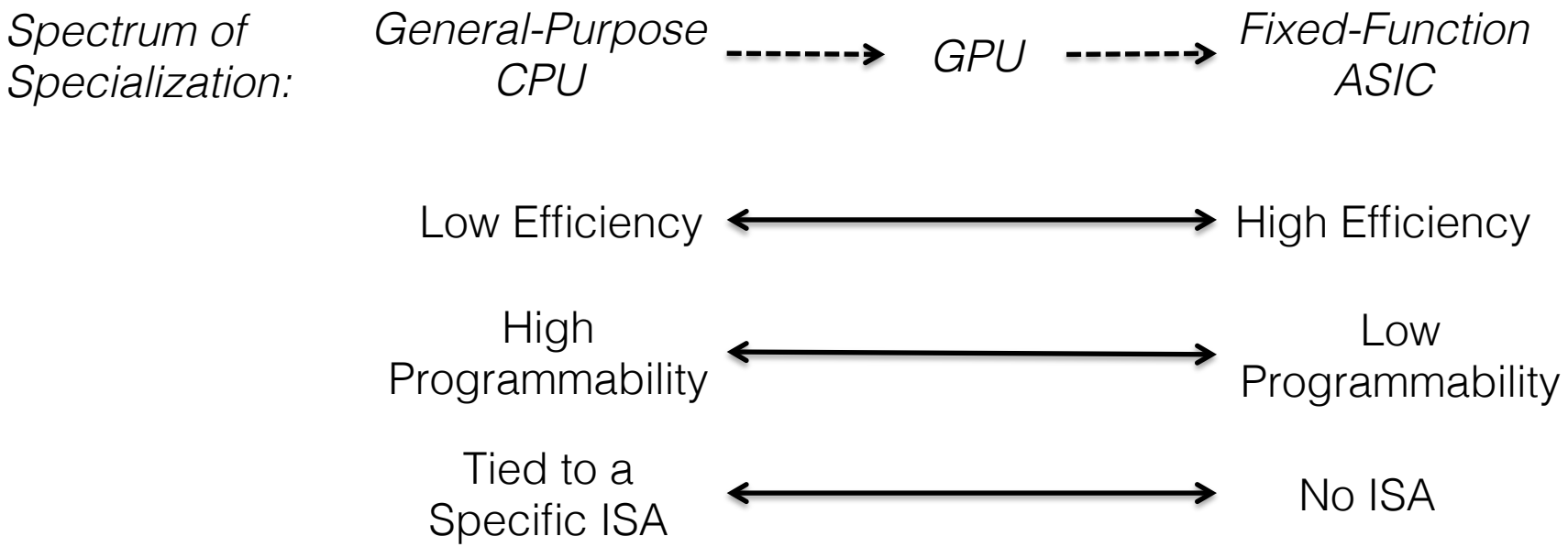
Workload ISA-Independent Characterization for Applications

Yakun Sophia Shao, Emma Wang,
Gu-Yeon Wei, David Brooks
Harvard University

Specialized architectures are decoupled from legacy ISAs.



Specialized architectures are decoupled from legacy ISAs.



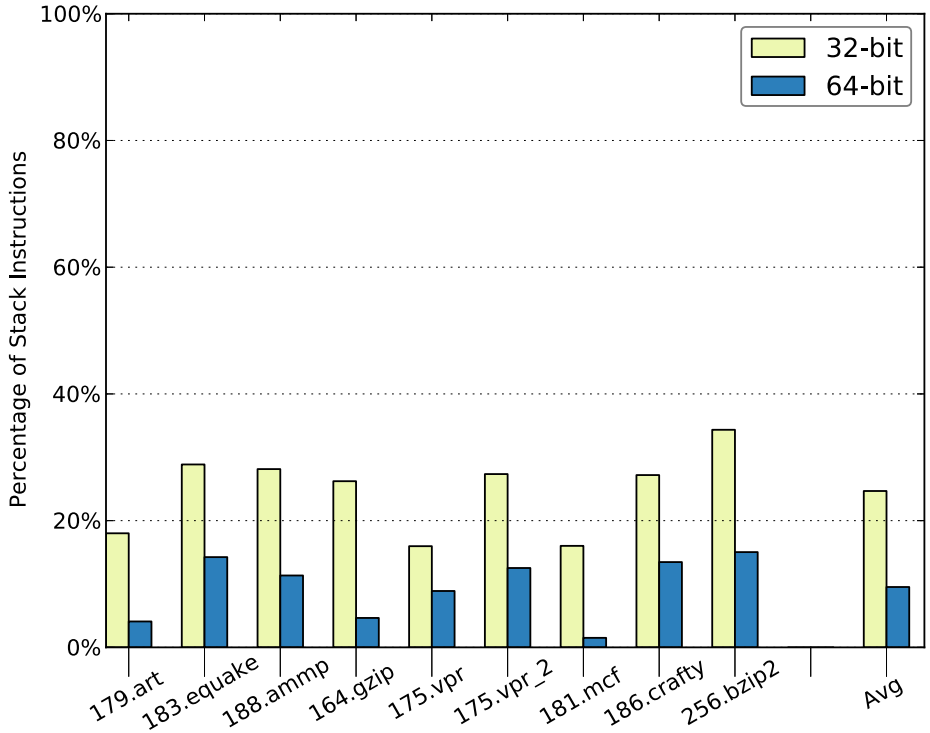
Performance-Counter Based Workload Characterization

- Metrics
  - IPC
  - Cache miss rates
  - Branch mis-prediction rates
  - …
- Microarchitecture-dependent
  - What if there is a bigger cache/a better branch predictor?
  - Not program intrinsic characteristics

ISA impacts program behaviors.

Stack Overhead

- Limited Registers
- Additional Load/Store



ISA impacts program behaviors.

Stack Overhead

- Limited Registers
- Additional Load/Store

Complex Operations

- Memory Operands
- Vector Operations

ISA impacts program behaviors.

Stack Overhead

- Limited Registers
- Additional Load/Store

Complex Operations

- Memory Operands
- Vector Operations

Calling Conventions

WIICA Summary

Goal:

- An analysis tool to characterize workloads ISA-Independent characteristics for specialized architectures

WIICA Summary

Goal:

- An analysis tool to characterize workloads ISA-Independent characteristics for specialized architectures

Methods:

- Leverage compiler's intermediate representation (IR)
- Categorize characteristics into compute, memory, and control

WIICA Summary

Goal:

- An analysis tool to characterize workloads ISA-Independent characteristics for specialized architectures

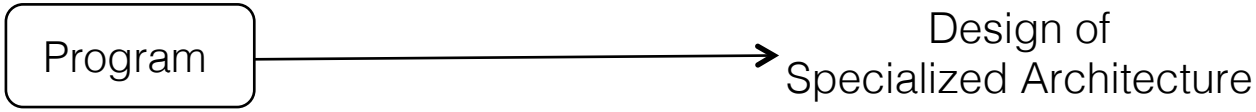
Methods:

- Leverage compiler's intermediate representation (IR)
- Categorize characteristics into compute, memory, and control

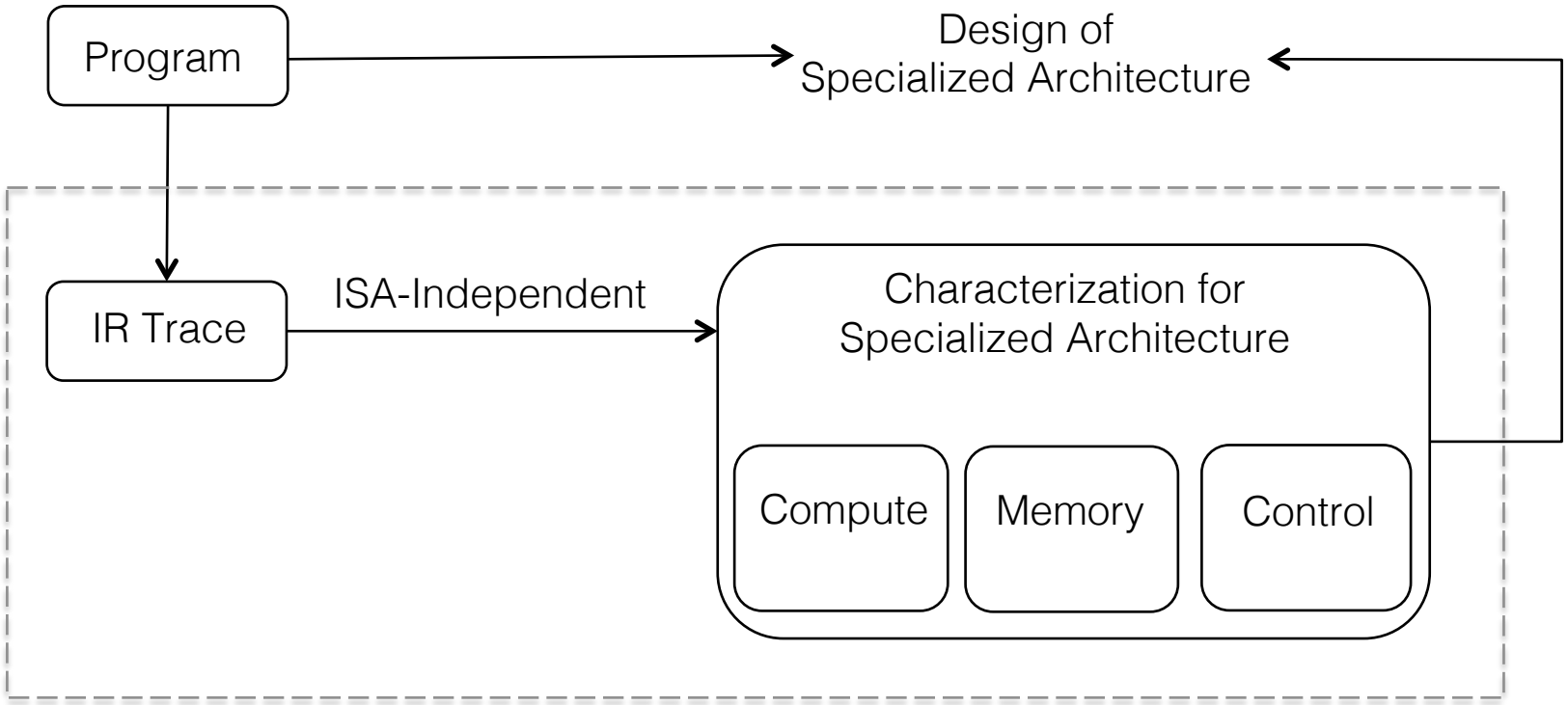
Takeaways:

- ISA-dependent characterization is misleading for specialization.
- ISA-independent characterization allows designers to quickly identify opportunities for specialization.

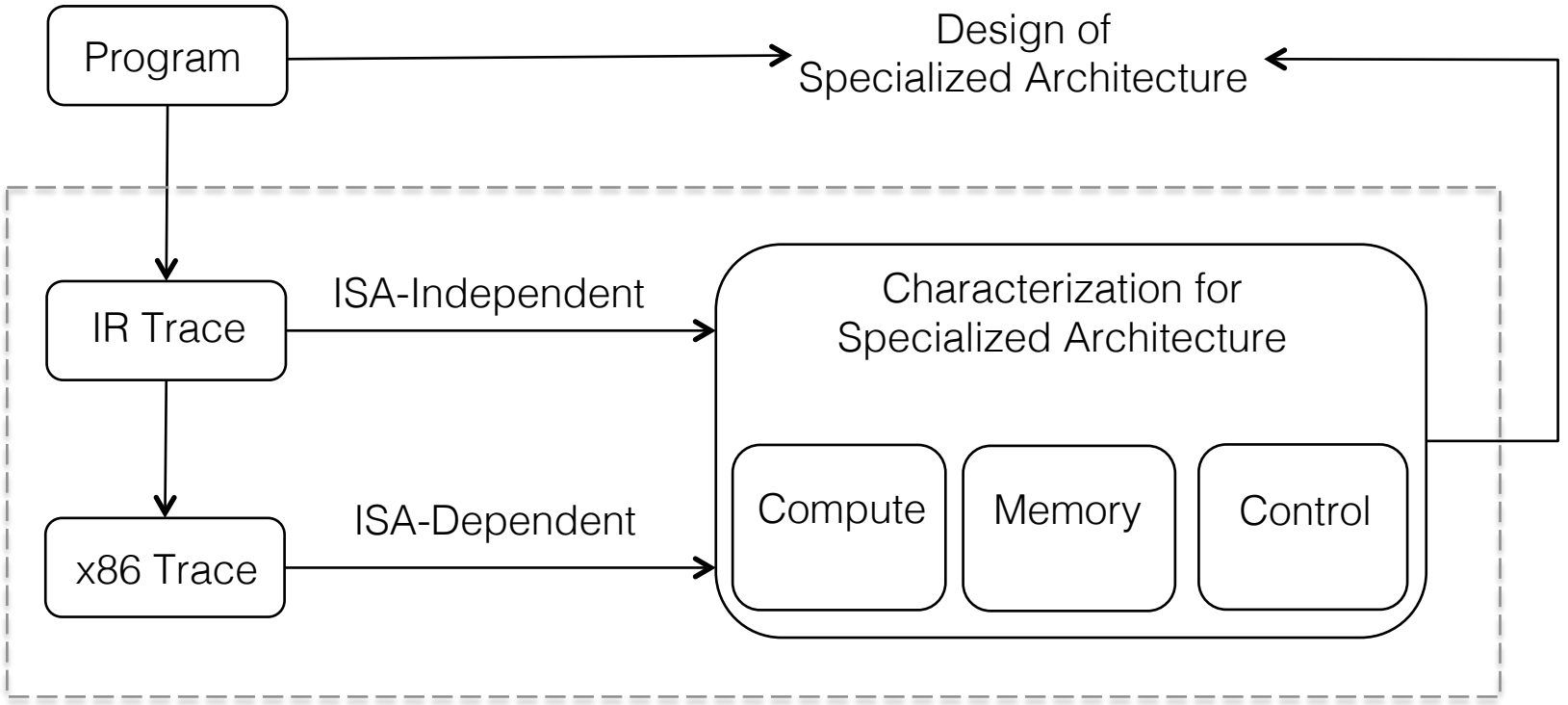
Tool Overview



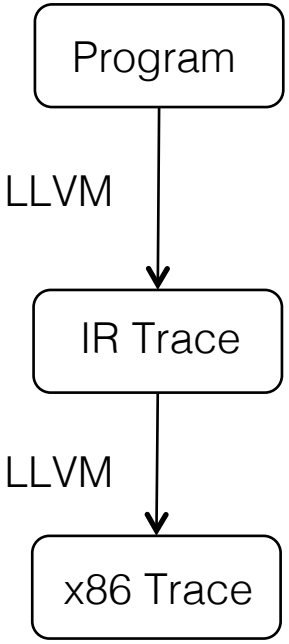
Tool Overview



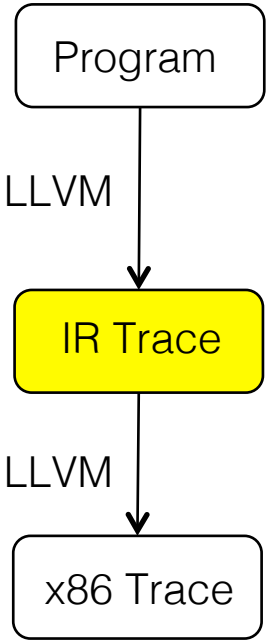
Tool Overview



Program Representations



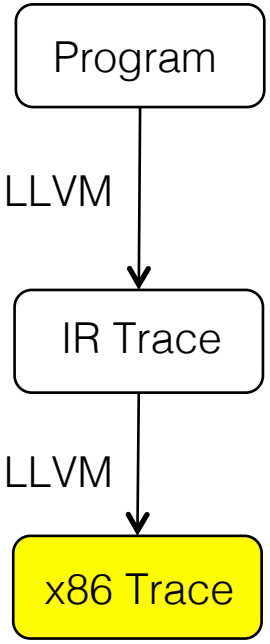
Program Representations



LLVM IR

- High-level IR
- Machine- and ISA-independent
- Features:
  - 60 instructions
  - Unlimited registers
  - Only loads/stores access memory
  - Disable vector operations

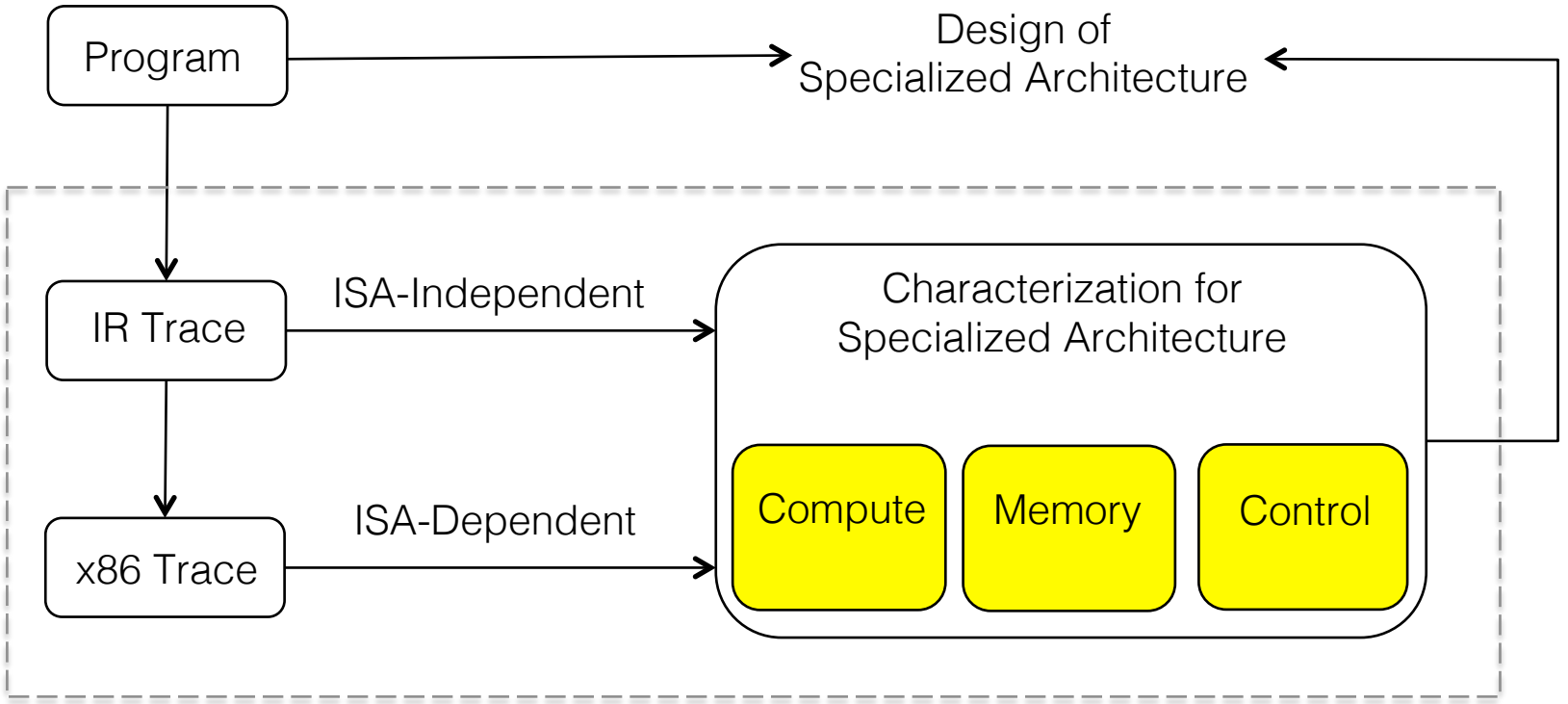
Program Representations



x86 Trace

- Used for ISA-dependent analysis
- Semantically equivalent to the IR code
- Collected with Pin instrumentation

Tool Overview



ISA-Independent Workload Characteristics

Compute

- Opcode Diversity
- Static Instructions (I-MEM)

Memory

- Memory Footprint (D-MEM)
- Memory Entropy
- Locality Score

Control

- Branch Instruction Counts
- Branch Entropy

ISA-Independent Workload Characteristics

Compute

- Opcode Diversity
- **Static Instructions (I-MEM)**

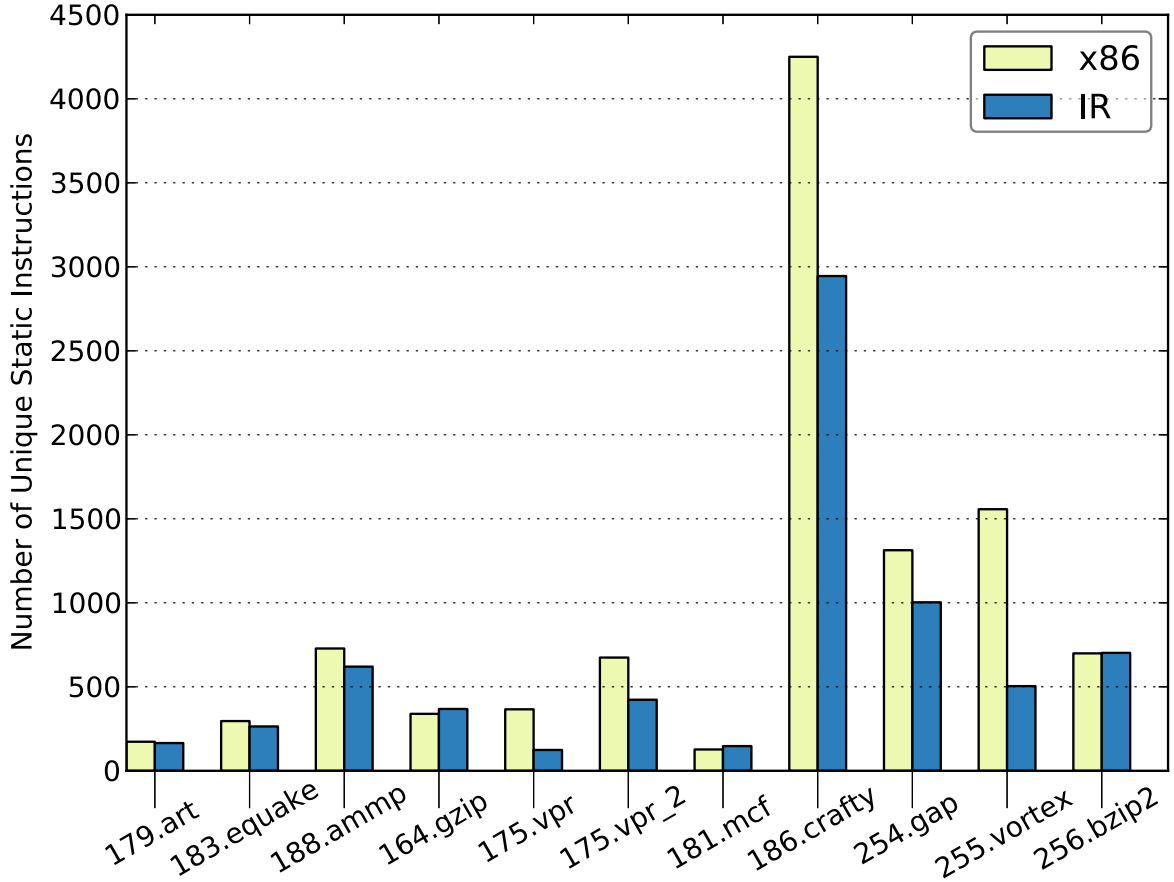
Memory

- Memory Footprint (D-MEM)
- Memory Entropy
- Locality Score

Control

- Branch Instruction Counts
- Branch Entropy

Compute::Static Instructions



ISA-Independent Workload Characteristics

Compute

- Opcode Diversity
- Static Instructions (I-MEM)

Memory

- Memory Footprint (D-MEM)
- **Memory Entropy**
- **Locality Score**

Control

- Branch Instruction Counts
- Branch Entropy

Memory::Entropy

Entropy: a measure of the randomness

$$Entropy = -\sum_{i=1}^{N} p(x_i) * \log_2 p(x_i)$$

Memory::Entropy

Entropy: a measure of the randomness

$$Entropy = -\sum_{i=1}^{N} p(x_i) * \log_2 p(x_i)$$

Case 1:

X is always a constant.

$$p(X) = 1$$
$$\log_2 p(X) = 0$$
$$Entropy = 0$$

Case 2:

N possible outcomes of X occur equally.

$$p(X) = \frac{1}{N}$$
$$\log_2 p(X) = \log_2 N^{-1}$$
$$Entropy = -N * \frac{1}{N} * \log_2 N^{-1}$$
$$Entropy = \log_2 N$$

Memory::Global Address Entropy

Temporal Locality

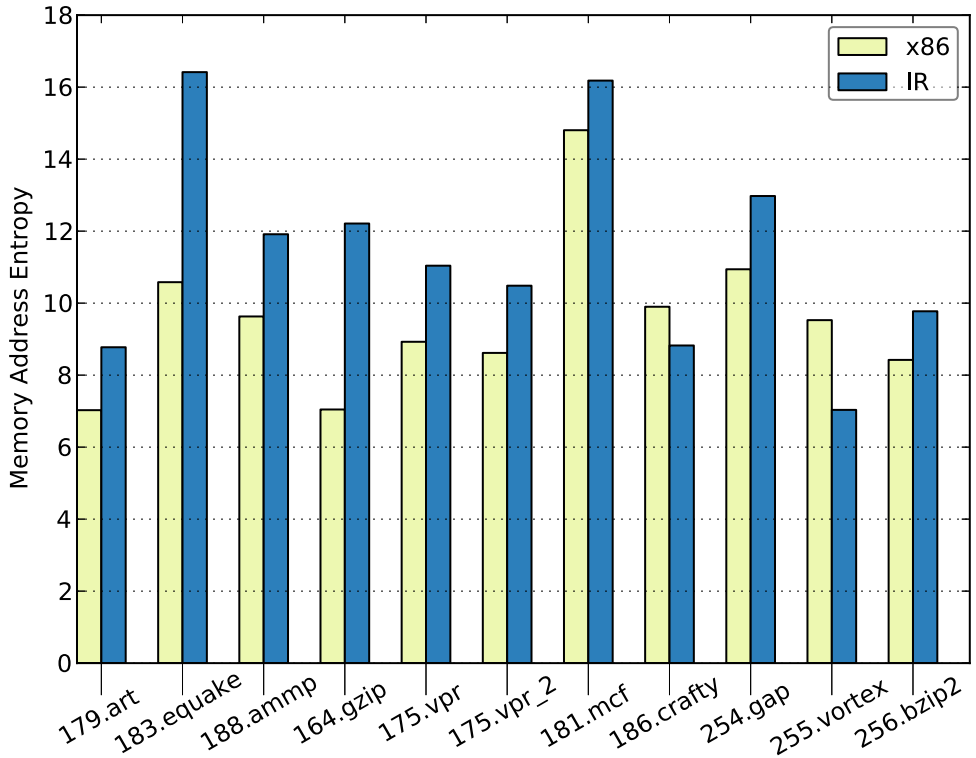
Address Stream A (less temporal locality)	Address Stream B (more temporal locality)
0 0 0 0	0 0 1 1
0 0 0 1	0 0 1 1
0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 1
Entropy = **2**	Entropy = 0

Yen, Draper, and Hill. Notary: Hardware Techniques to Enhance Signatures. MICRO 08

Memory::Global Address Entropy

Temporal Locality

Address Stream A (less temporal locality)	Address Stream B (more temporal locality)
0 0 0 0	0 0 1 1
0 0 0 1	0 0 1 1
0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 1
Entropy = **2**	Entropy = 0

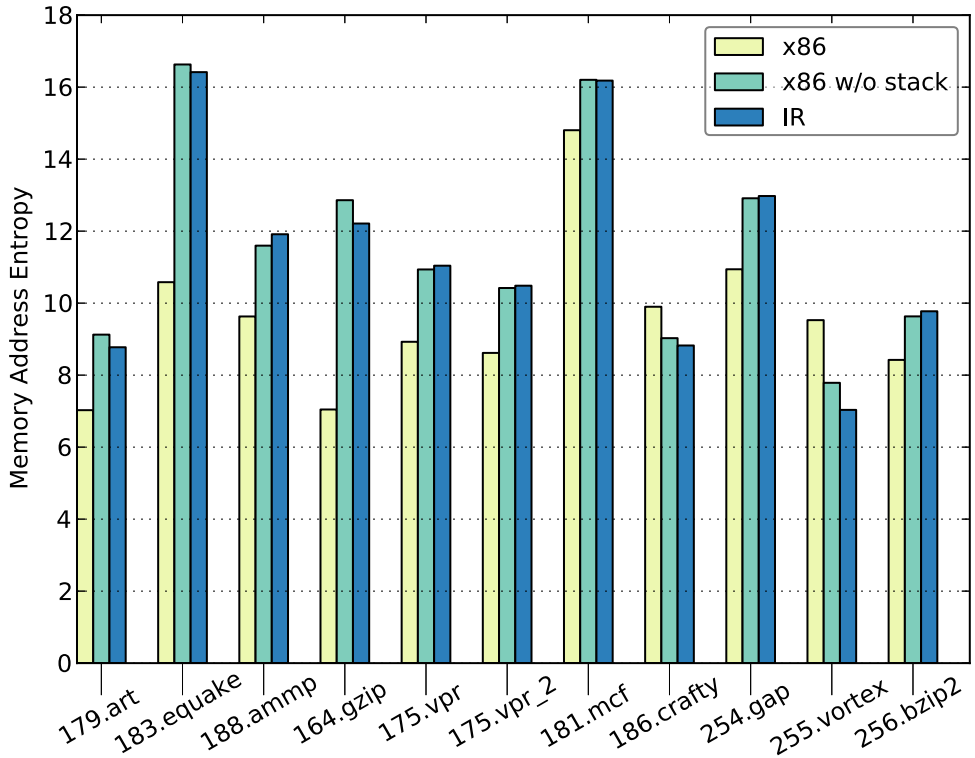


Yen, Draper, and Hill. Notary: Hardware Techniques to Enhance Signatures. MICRO 08

Memory::Global Address Entropy

Temporal Locality

Address Stream A (less temporal locality)	Address Stream B (more temporal locality)
0 0 0 0	0 0 1 1
0 0 0 1	0 0 1 1
0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 1
Entropy = **2**	Entropy = 0



Yen, Draper, and Hill. Notary: Hardware Techniques to Enhance Signatures. MICRO 08

ISA-Independent Workload Characteristics

Compute

- Opcode Diversity
- Static Instructions (I-MEM)

Memory

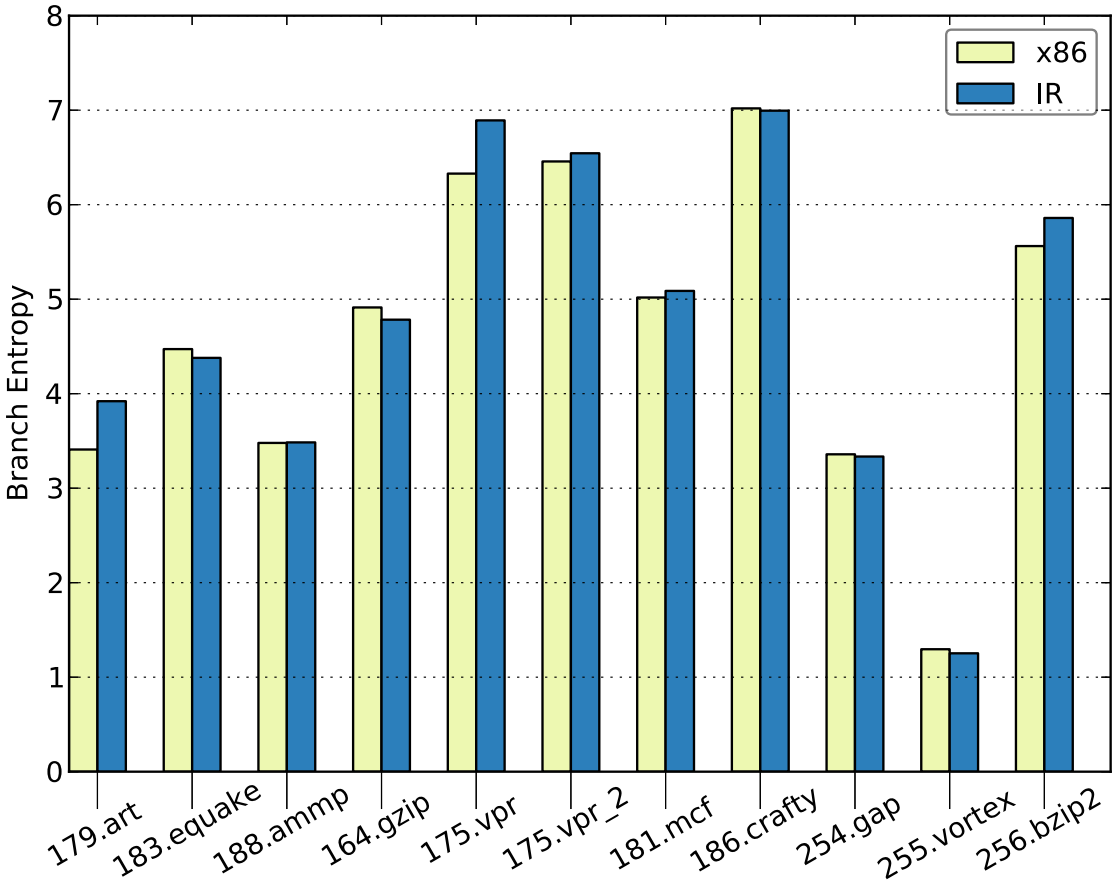
- Memory Footprint (D-MEM)
- Memory Entropy
- Locality Score

Control

- Branch Instruction Counts
- **Branch Entropy**

Yokota, et all, Introducing Entropies for Representing Program Behavior and Branch Predictor Performance, 07

Control::Branch Entropy



ISA-Independent Workload Characteristics

Compute

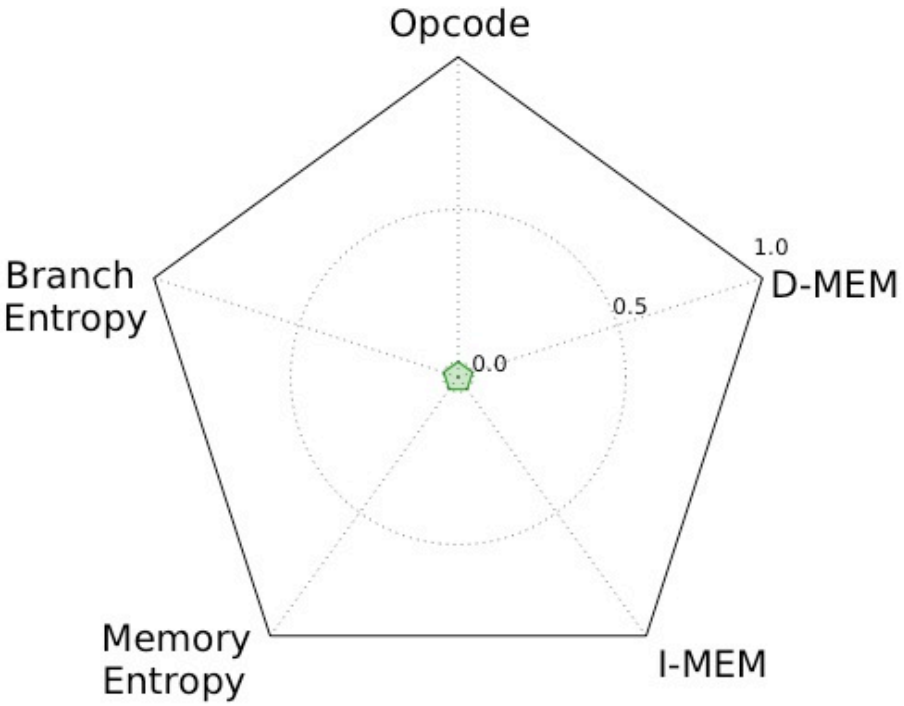
- **Opcode Diversity**
- **Static Instructions (I-MEM)**

Memory

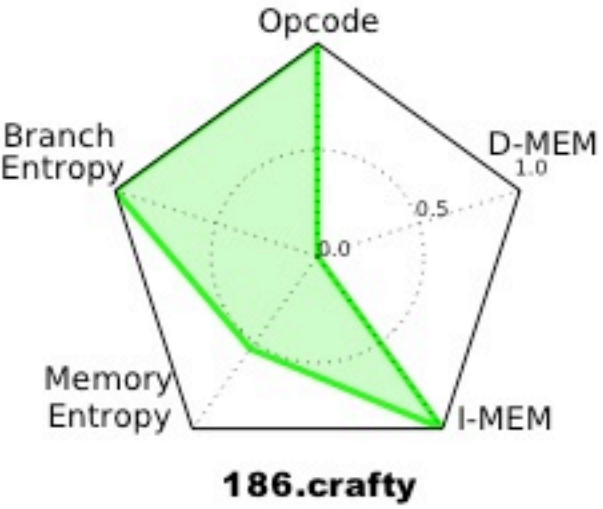
- **Memory Footprint (D-MEM)**
- **Global Address Entropy**
- Local Address Entropy

Control

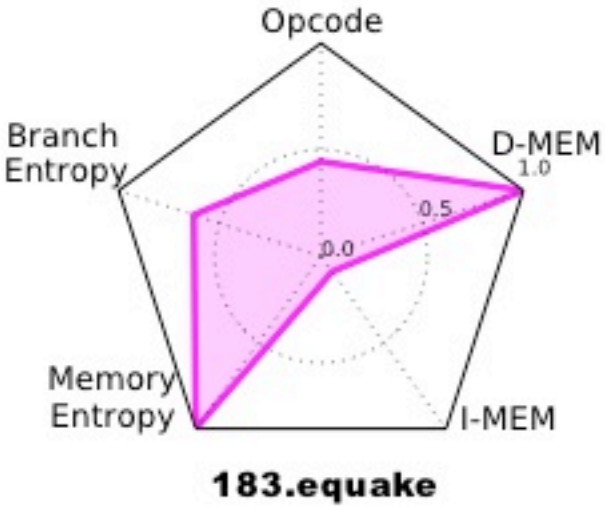
- Branch Instruction Counts
- **Branch Entropy**



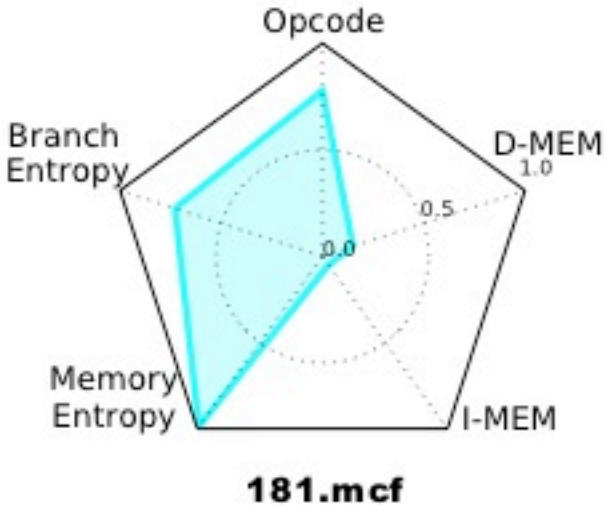
Workload Characterization



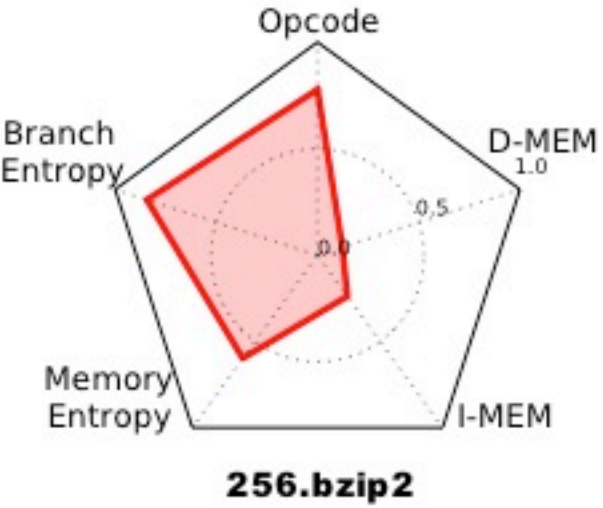
186.crafty



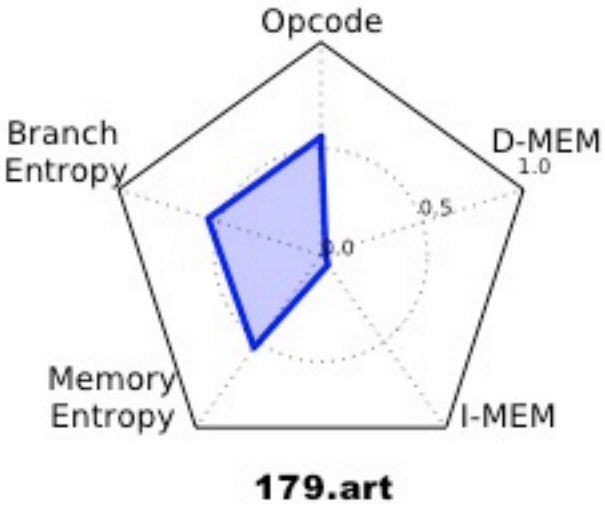
183.equake



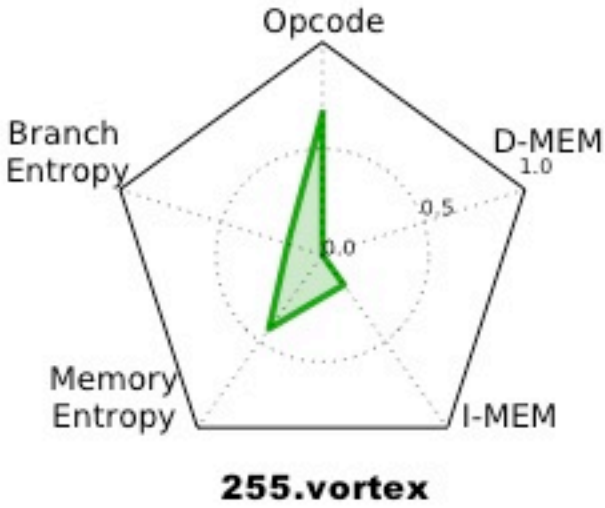
181.mcf



256.bzip2

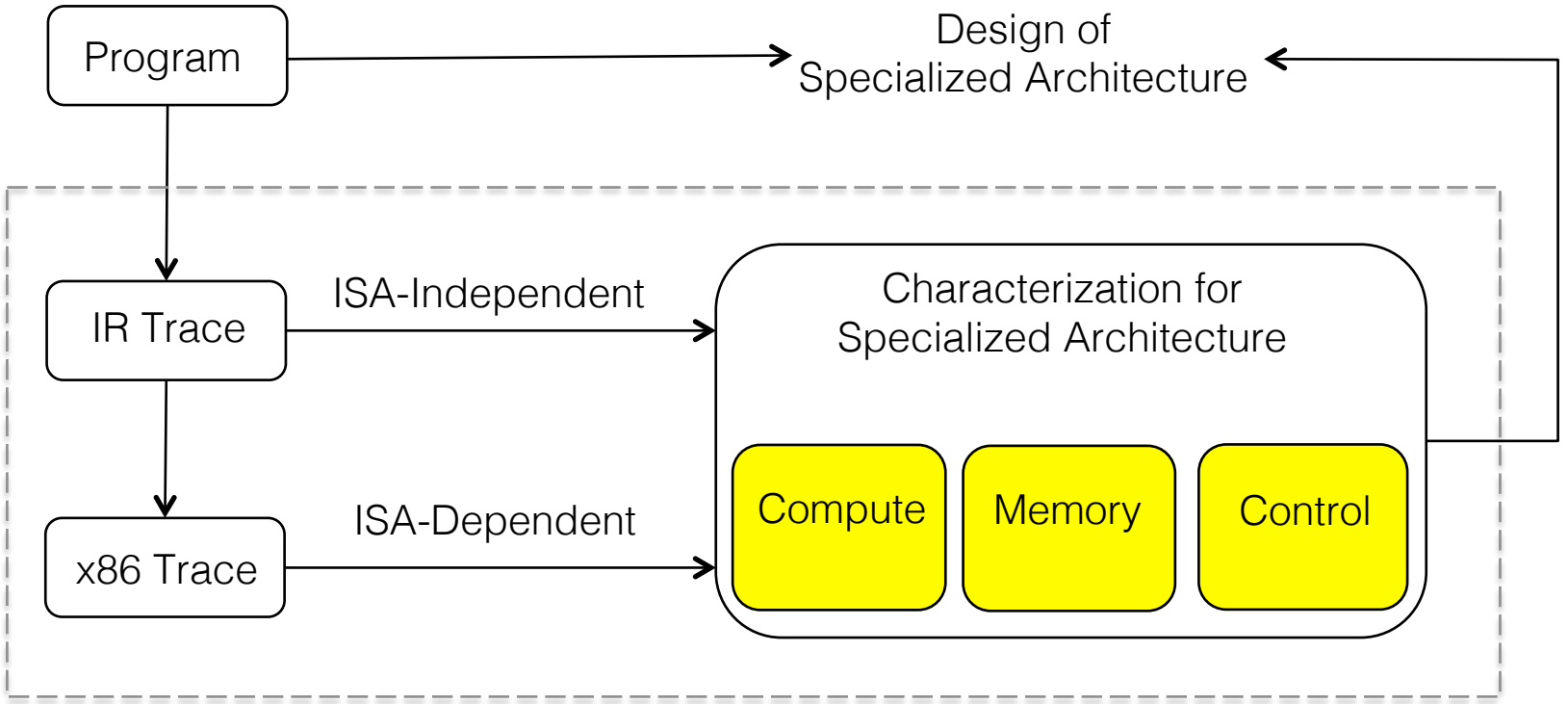


179.art



255.vortex

Tool Overview



Tutorial References

- Y.S. Shao and D. Brooks, “ISA-Independent Workload Characterization and its Implications for Specialized Architectures,” ISPASS’13.
- B. Reagen, Y.S. Shao, G.-Y. Wei, D. Brooks, “Quantifying Acceleration: Power/ Performance Trade-Offs of Application Kernels in Hardware,” ISLPED’13.
- Y.S. Shao, B. Reagen, G.-Y. Wei, D. Brooks, “Aladdin: A Pre-RTL, Power-Performance Accelerator Simulator Enabling Large Design Space Exploration of Customized Architectures,” ISCA’14.
- B. Reagen, B. Adolf, Y.S. Shao, G.-Y. Wei, D. Brooks, “MachSuite: Benchmarks for Accelerator Design and Customized Architectures,” IISWC’14.
- **http://vlsiarch.eecs.harvard.edu/accelerators**