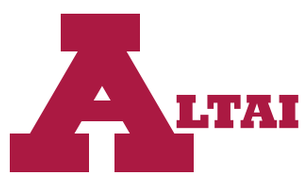
AISC
Approximate Instruction Set Computer

Alexandra Ferrerón, Darío Suárez-Gracia
Jesús Alastruey-Benedé, **Ulya R. Karpuzcu**

University of Zaragoza
University of Minnesota, Twin-Cities

WAX'18, March 25 2018

UNIVERSITY OF MINNESOTA



Instruction Set Architecture (ISA)

- View from hardware stack:
  - Behavioral design specification
  - **Determines hardware complexity**
- View from software stack:
  - Definition of the machine capabilities
  - **Determines functional completeness**

Approximate Instruction Set Computer (AISC)

- Single-ISA heterogeneous fabric
- Each compute engine (core or fixed function) supports a subset of the ISA
  - Component (functionally-incomplete) ISA may overlap
  - Incomplete ISA can reduce microarchitectural complexity
    - thereby improve performance per Watt
- The union of incomplete ISA subsets renders a functionally complete ISA

Approximate Instruction Set Computer (AISC)

- Single-ISA heterogeneous fabric
- Each compute engine (core or fixed function) supports a subset of the ISA
  - Component (functionally-incomplete) ISA may overlap
  - Incomplete ISA can reduce microarchitectural complexity
    - thereby improve performance per Watt
- The union of incomplete ISA subsets renders a functionally complete ISA

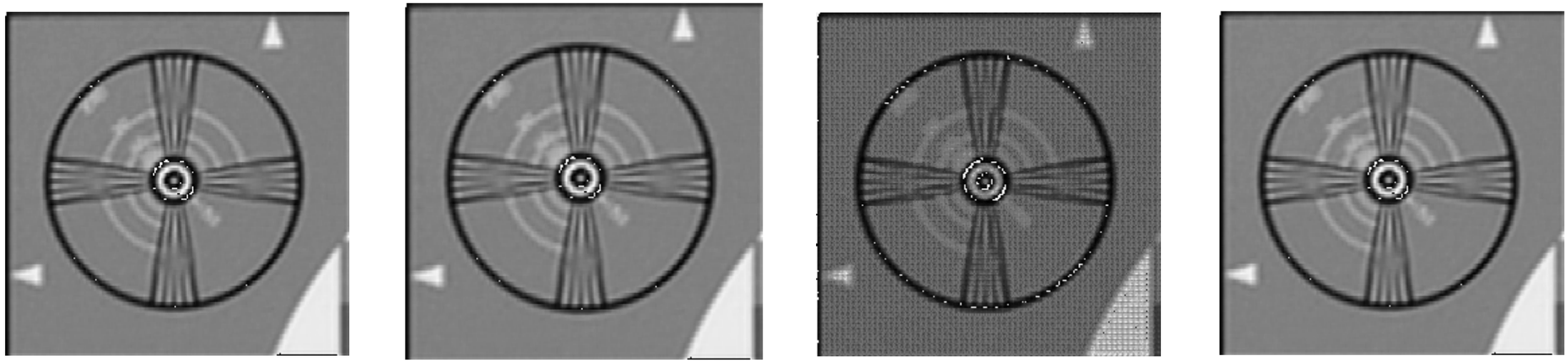
Code that cannot be approximated can run at full accuracy

Improved energy efficiency if ISA-induced approximation is tolerable

Approximate Instruction Set Computer (AISC)

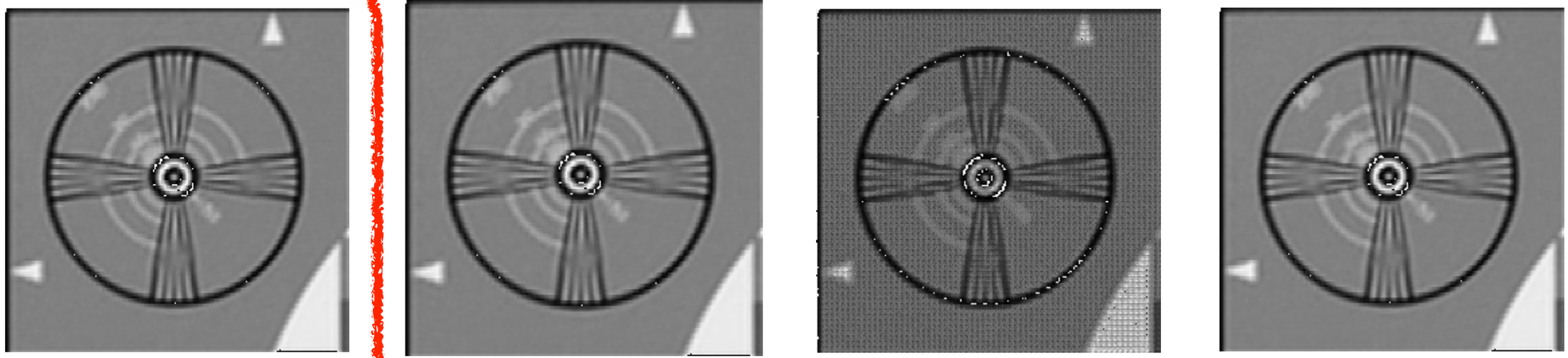
- How to determine incomplete, approximate ISA subsets?
  - Vertical approximation:
    - Exclude less frequently used complex instructions
  - Horizontal approximation:
    - Simplifies each instruction, by e.g., precision reduction

Proof-Of-Concept



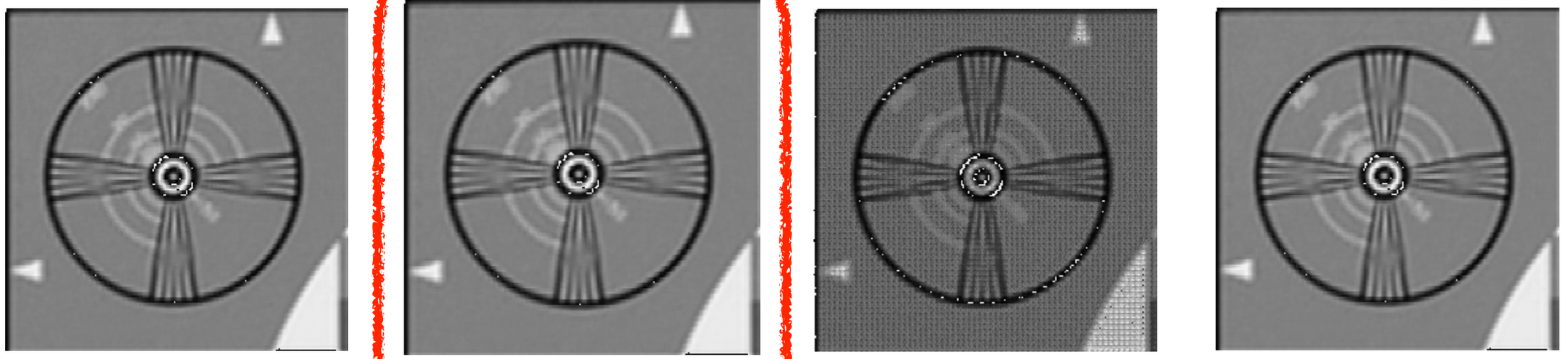
(a) Native (b) Vertical (c) Horizontal (d) Horiz.+Vert.

Proof-Of-Concept



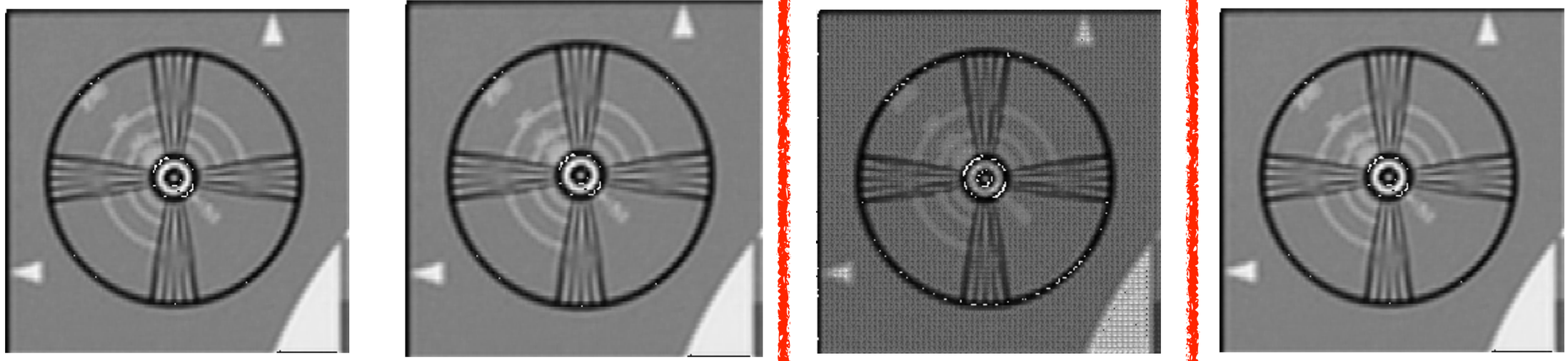
(a) Native (b) Vertical (c) Horizontal (d) Horiz.+Vert.

Proof-Of-Concept



(a) Native (b) Vertical (c) Horizontal (d) Horiz.+Vert.

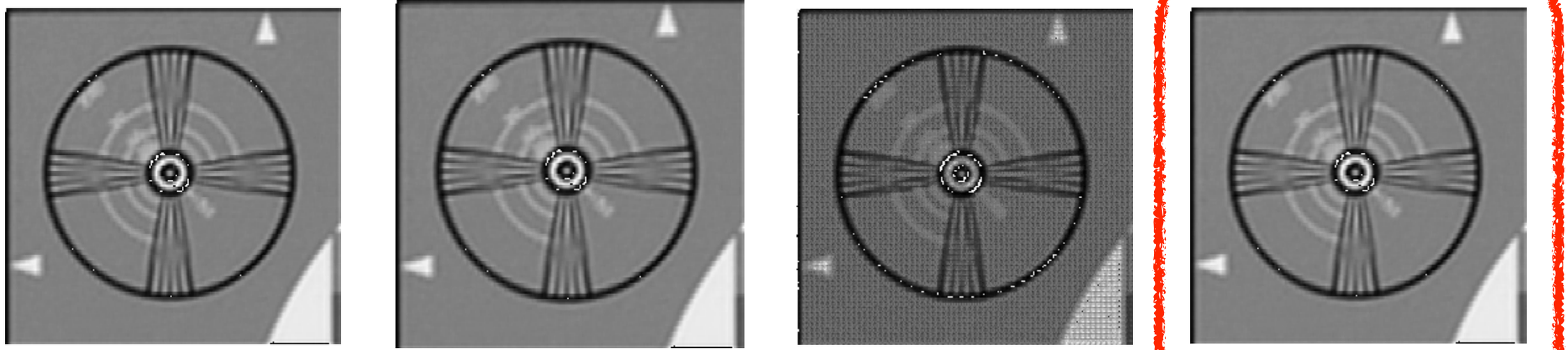
Proof-Of-Concept



(a) Native (b) Vertical (c) Horizontal (d) Horiz.+Vert.

double precision → half-precision

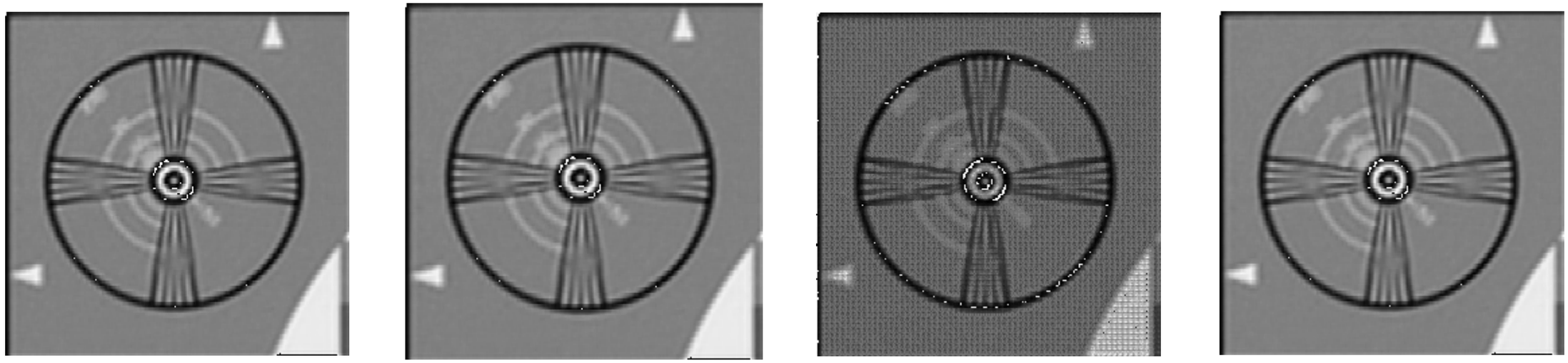
Proof-Of-Concept



(a) Native (b) Vertical (c) Horizontal (d) Horiz.+Vert.

DIV → MUL

Proof-Of-Concept



(a) Native (b) Vertical (c) Horizontal (d) Horiz.+Vert.

Up to 37% energy cut at around 10% accuracy loss

Design Aspects

- Which subset of the ISA should each compute engine support?
- How to map instruction sequences to compute engines?
- How to keep the potential accuracy loss bounded?
- How to orchestrate migration of code sequences
  - from one compute engine to another within the course of computation
  - tolerance to noise may vary for different application phases

Design Aspects

- Which subset of the ISA should each compute engine support?
- How to map instruction sequences to compute engines?
- How to keep the potential accuracy loss bounded?
- How to orchestrate migration of code sequences
  - from one compute engine to another within the course of computation
  - tolerance to noise may vary for different application phases

- **Most critical design aspect: migration granularity**
  - **A break-even point exists for migration granularity (and frequency)**

AISC
Approximate Instruction Set Computer

Alexandra Ferrerón, Darío Suárez-Gracia
Jesús Alastruey-Benedé, **Ulya R. Karpuzcu**

ukarpuzc@umn.edu

WAX'18, March 25 2018

UNIVERSITY OF MINNESOTA

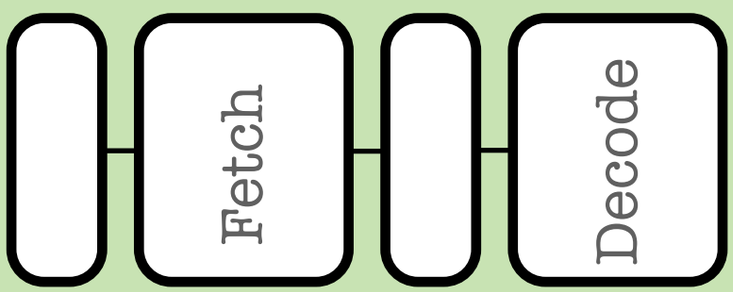
Instruction Set Architecture (ISA)

```
int main(void)
{
   ...
}
```

Software

```
DIV.D     F0, F2, F4
ADD.D     F10, F0, F8
SUB.D     F8, F8, F14
```

Architecture



Hardware