General-Purpose Code Acceleration with Limited-Precision Analog Computation

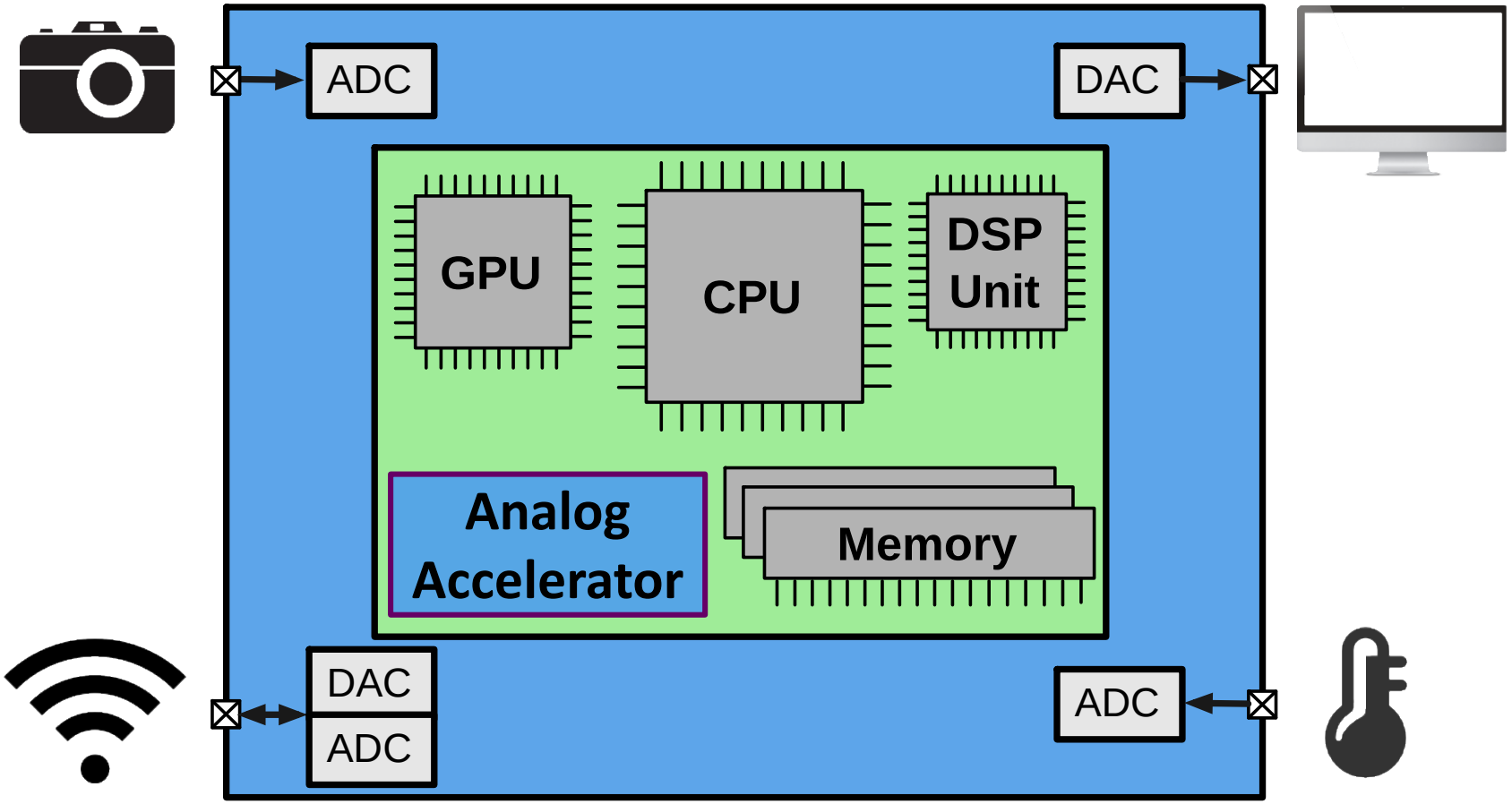
Renée St. Amant **Amir Yazdanbakhsh** Jongse Park Bradley Thwaites
Hadi Esmaeilzadeh Arjang Hassibi Luis Ceze Doug Burger

Georgia Institute of Technology
Alternative Computing Technologies (ACT) Lab

Georgia Institute of Technology
University of Washington

The University of Texas at Austin
Microsoft Research

ISCA 2014



Input and Output
Display
Communication
Sensing

Analog Domain

Processing
Storage

Digital Domain

How to use analog circuits for accelerating programs written in conventional languages?

1) Neural transformation
   [Esmaeilzadeh et. al., MICRO 2012]

2) Analog neurons

3) Compiler-circuit co-design

Challenges

- Analog circuits are mainly single function
- Instruction control cannot be analog
- Storing intermediate results in analog domain is not effective
- Analog circuits have limited operational range

1) Neural transformation

2) Analog neurons

3) Compiler-circuit co-design

Challenges

- Analog circuits are mainly single function
- Instruction control cannot be analog
- Storing intermediate results in analog is not effective
- Analog circuits have limited operational range

1) Neural transformation

2) Analog neurons

3) Compiler-circuit co-design

Challenges

- Analog circuits are mainly single function
- Instruction control cannot be analog
- Storing intermediate results in analog domain is not effective
- Analog circuits have limited operational range

1) Neural transformation

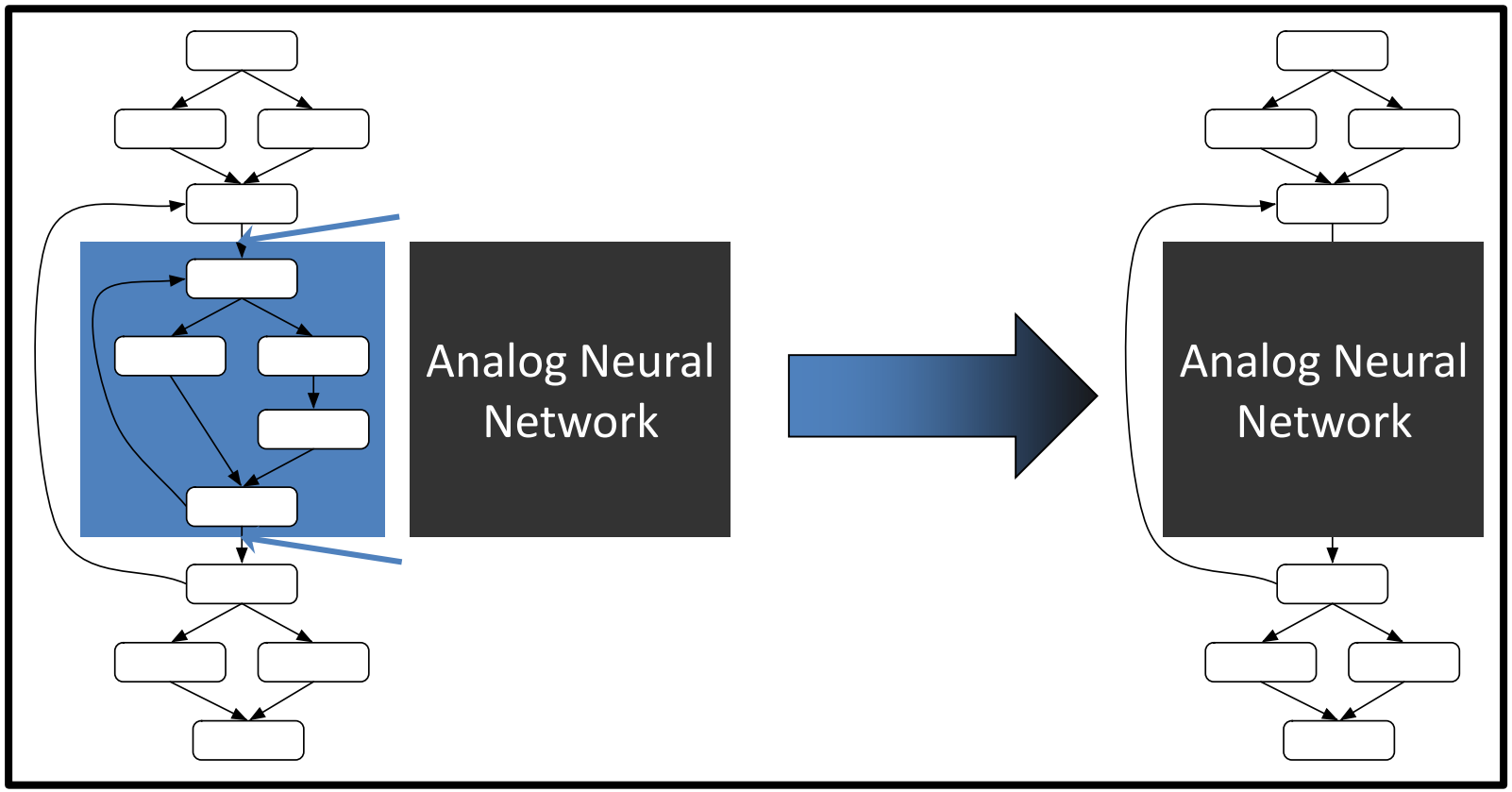
2) Analog neurons

3) Compiler-circuit co-design

1st Design Principle

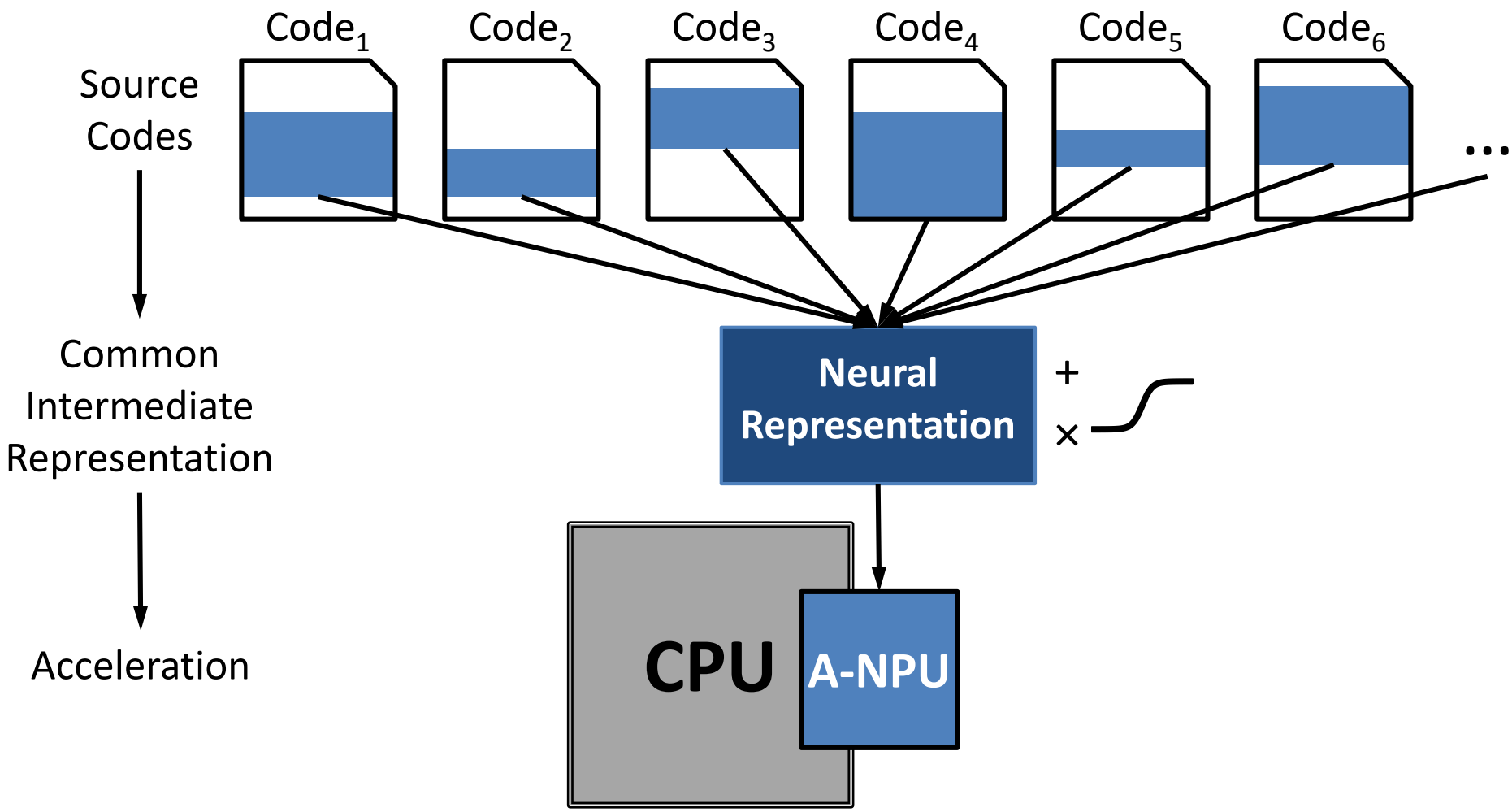
Neural Transformation

Neural Transformation



Esmaeilzadeh, Sampson, Ceze, Burger, “Neural Acceleration for General-Purpose Approximate Programs,” MICRO 2012.

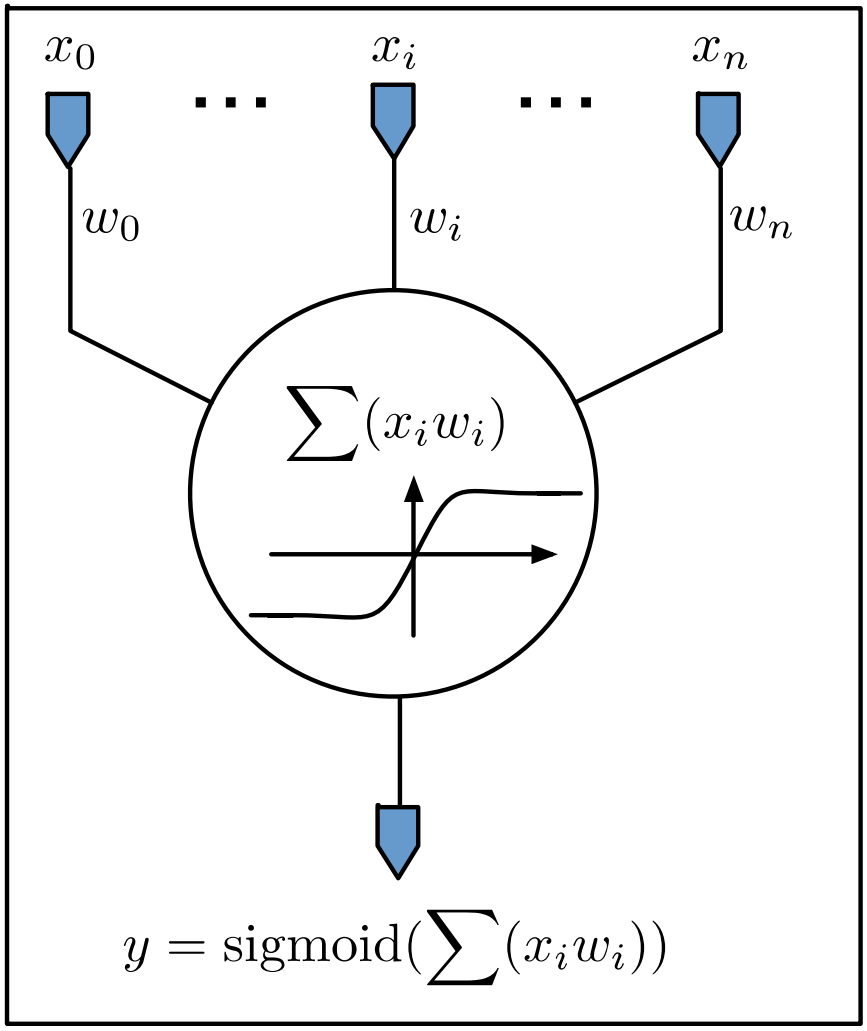
A-NPU acceleration

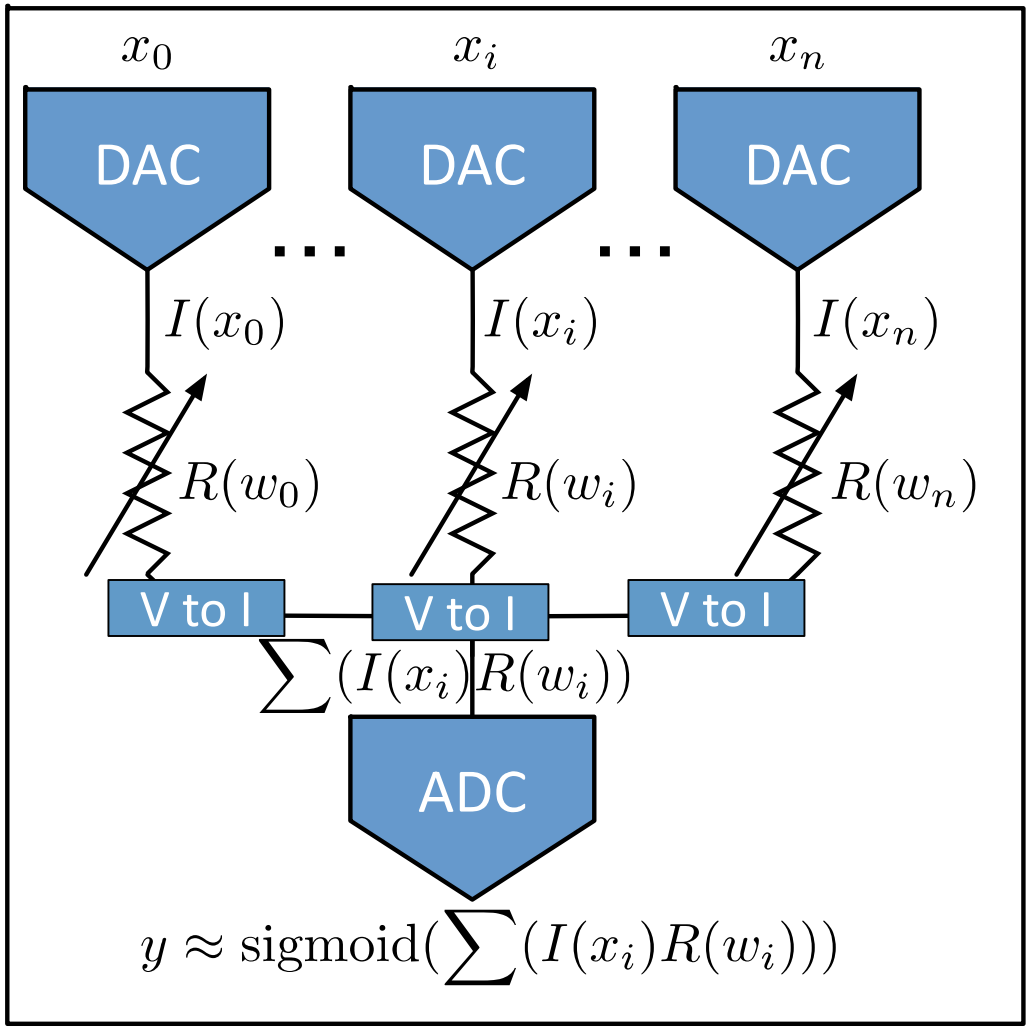


2nd Design Principle

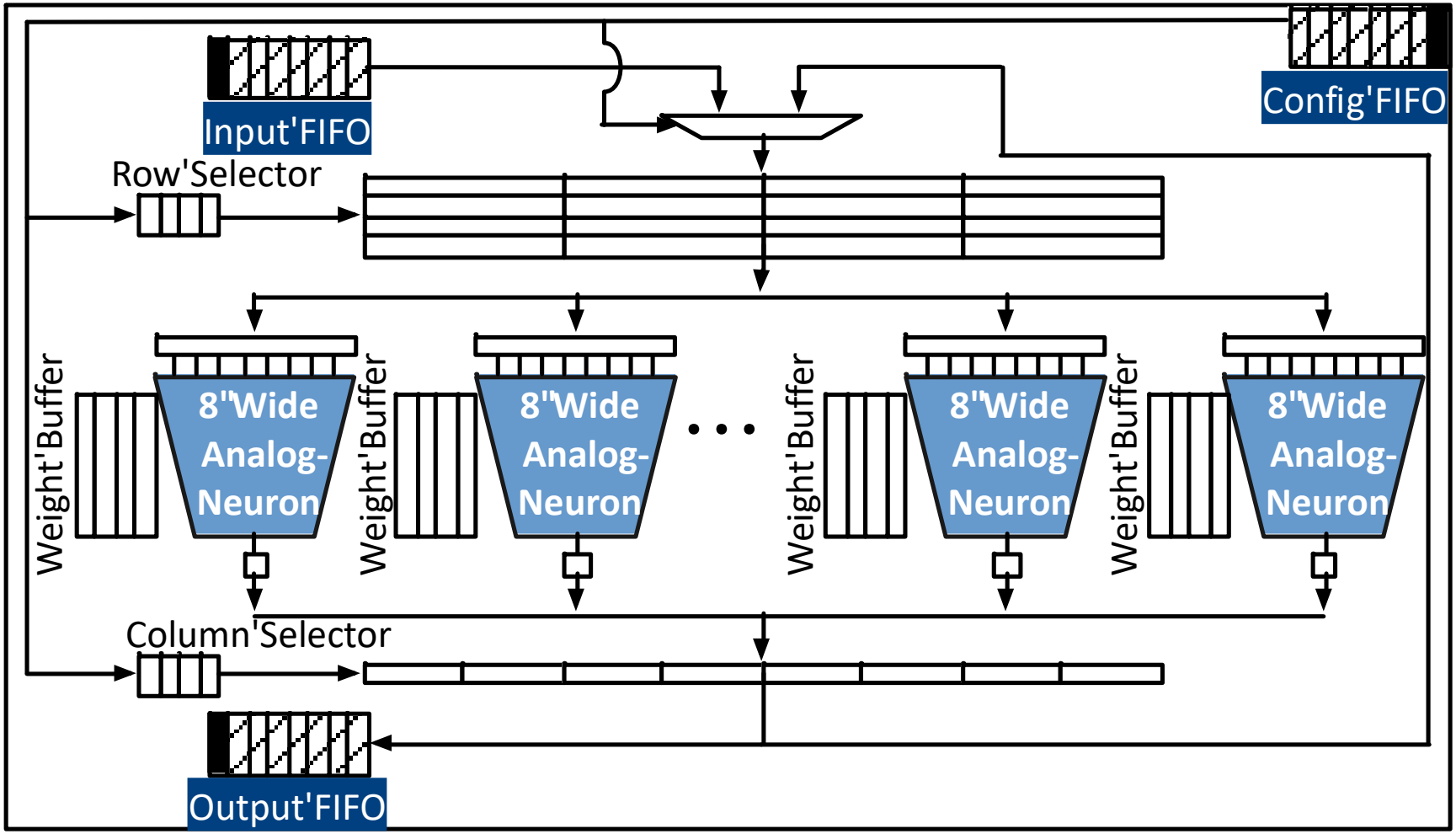
Analog Neurons

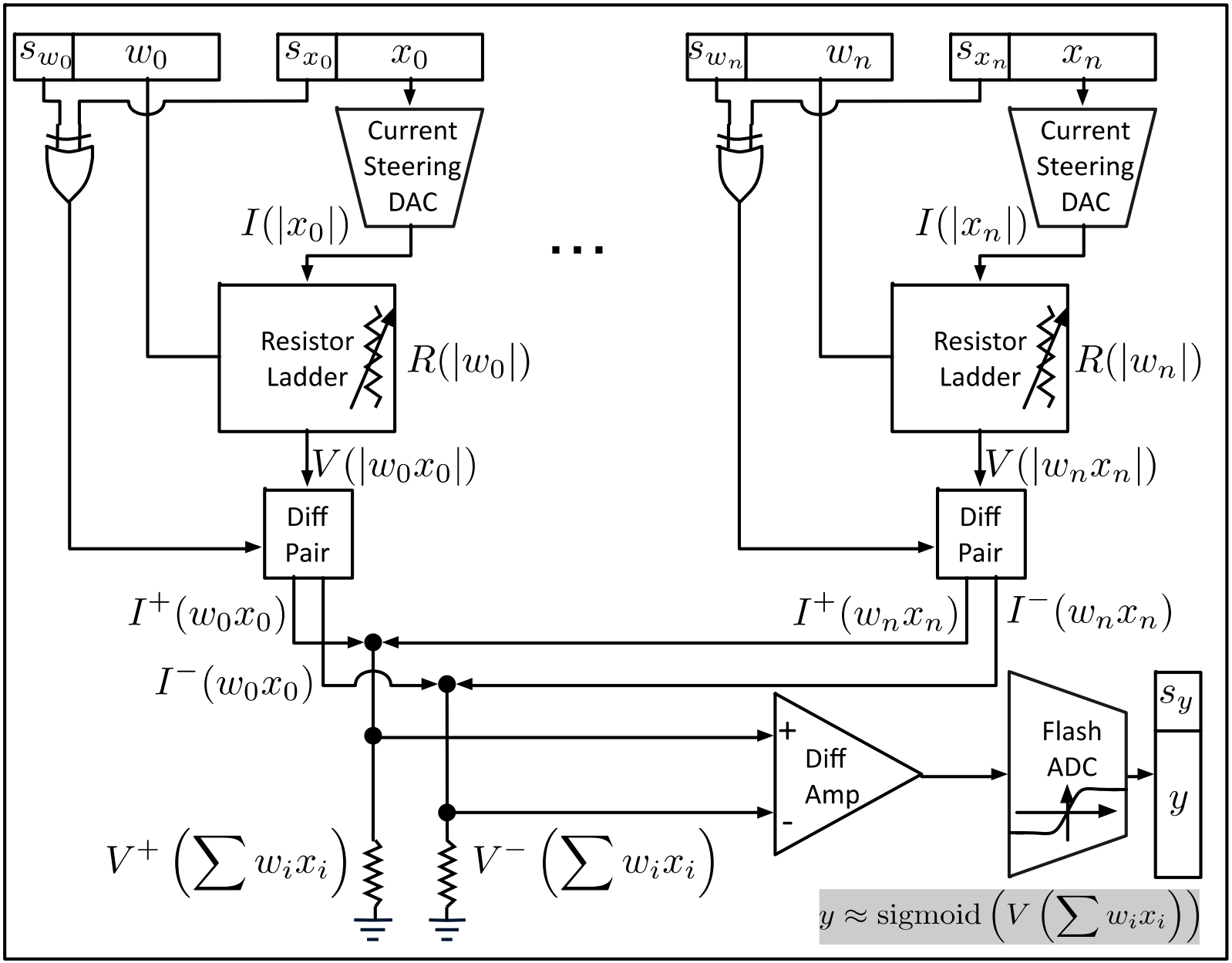
Analog Neurons for Accelerated Computation





Mixed-signal A-NPU




s_{w_0}
w_0
s_{x_0}
x_0
Current Steering DAC
$I(|x_0|)$
Resistor Ladder
$R(|w_0|)$
$V(|w_0 x_0|)$
Diff Pair
$I^+(w_0 x_0)$
$I^-(w_0 x_0)$
s_{w_n}
w_n
s_{x_n}
x_n
Current Steering DAC
$I(|x_n|)$
Resistor Ladder
$R(|w_n|)$
$V(|w_n x_n|)$
Diff Pair
$I^+(w_n x_n)$
$I^-(w_n x_n)$
...
+
Diff Amp
-
Flash ADC
s_y
y
$V^+\left(\sum w_i x_i\right)$
$V^-\left(\sum w_i x_i\right)$
$y \approx \text{sigmoid}\left(V\left(\sum w_i x_i\right)\right)$

Limitations of Analog Neuron

Limited range of operation (e.g. 600mV)

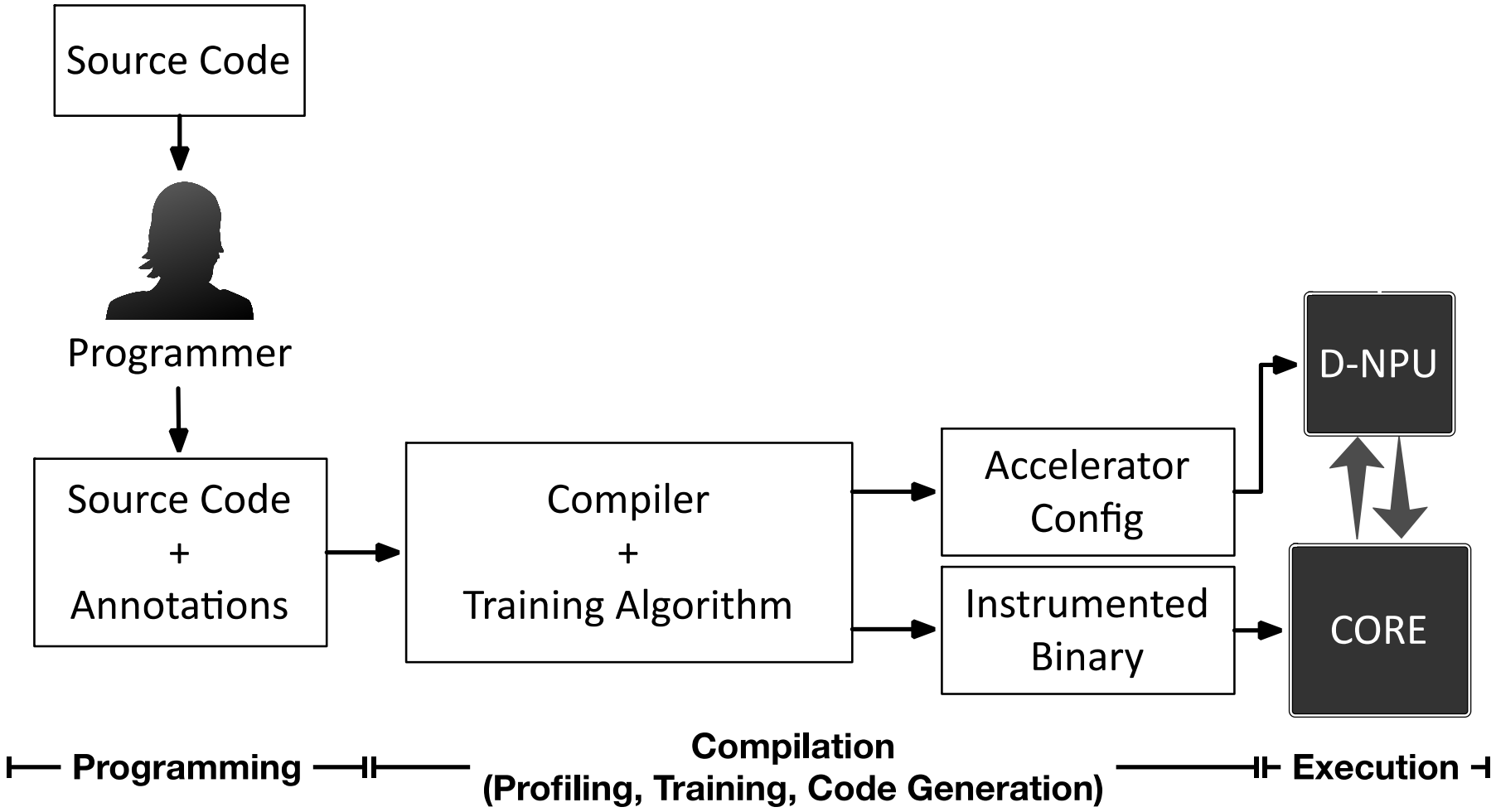
Margins for noise resiliency (2-3 mV)

Limited Bit-width
Topology Restriction
Circuit Non-idealities (e.g., Sigmoid)

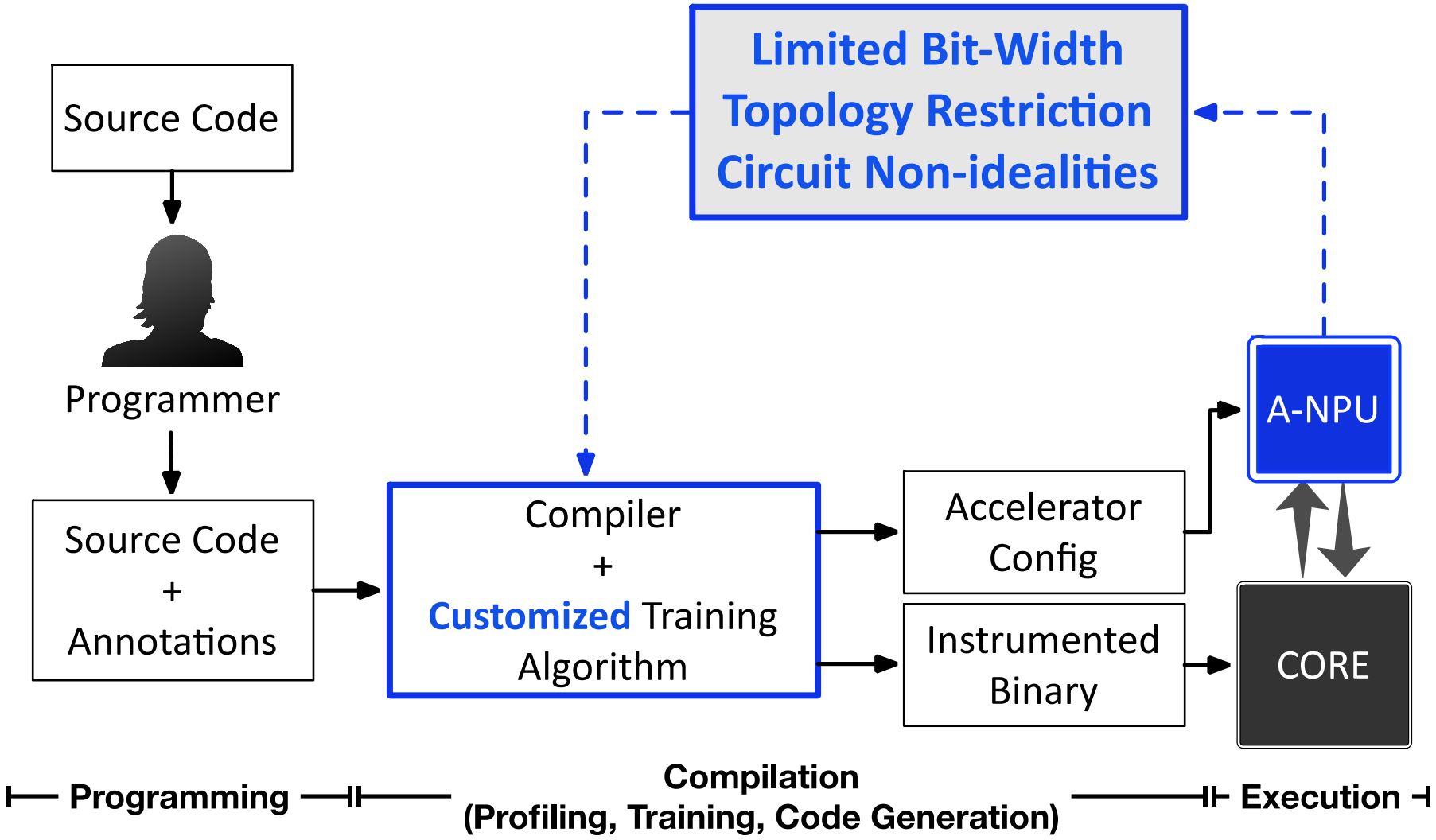
3rd Design Principle

Compiler-Circuit Co-design

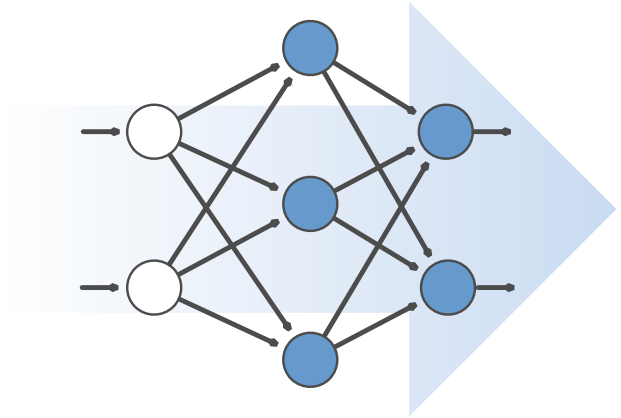
Digital Compilation Workflow



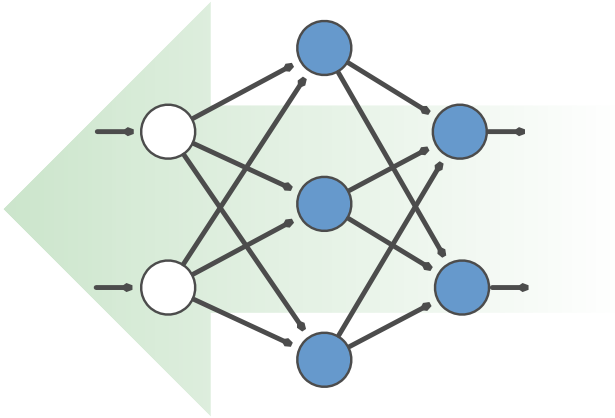
Analog Compilation Workflow



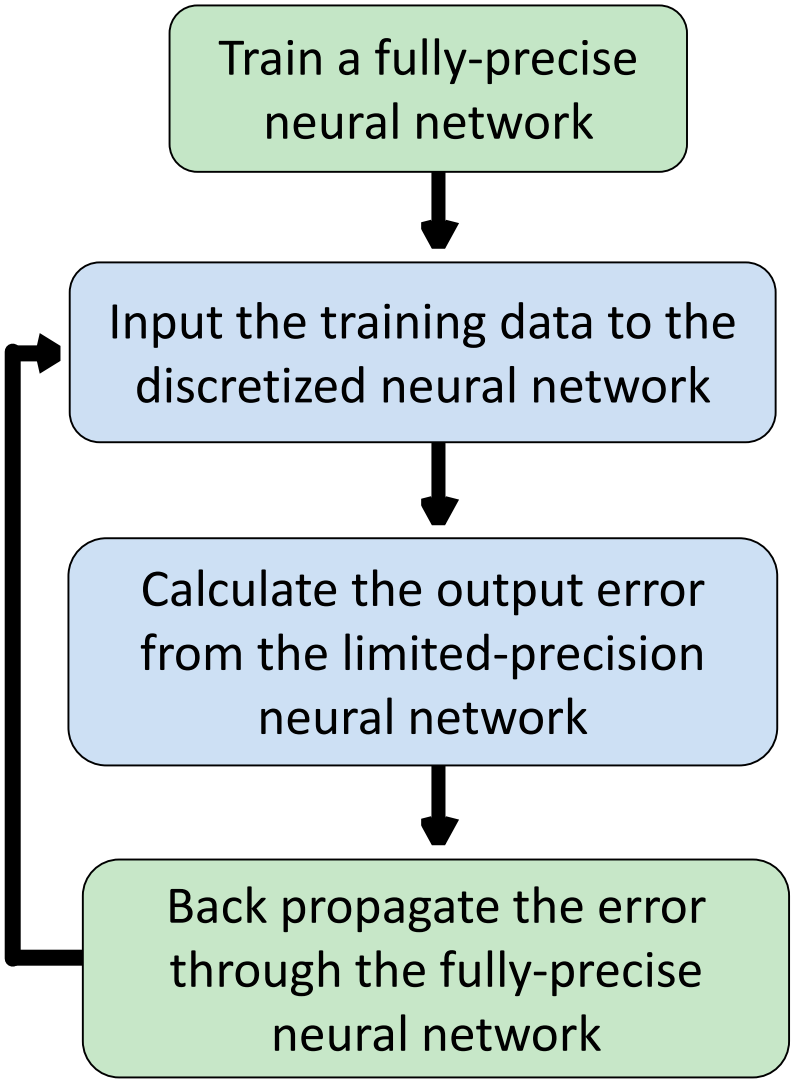
(1) Training with Limited Bit-width



Limited-Precision Network

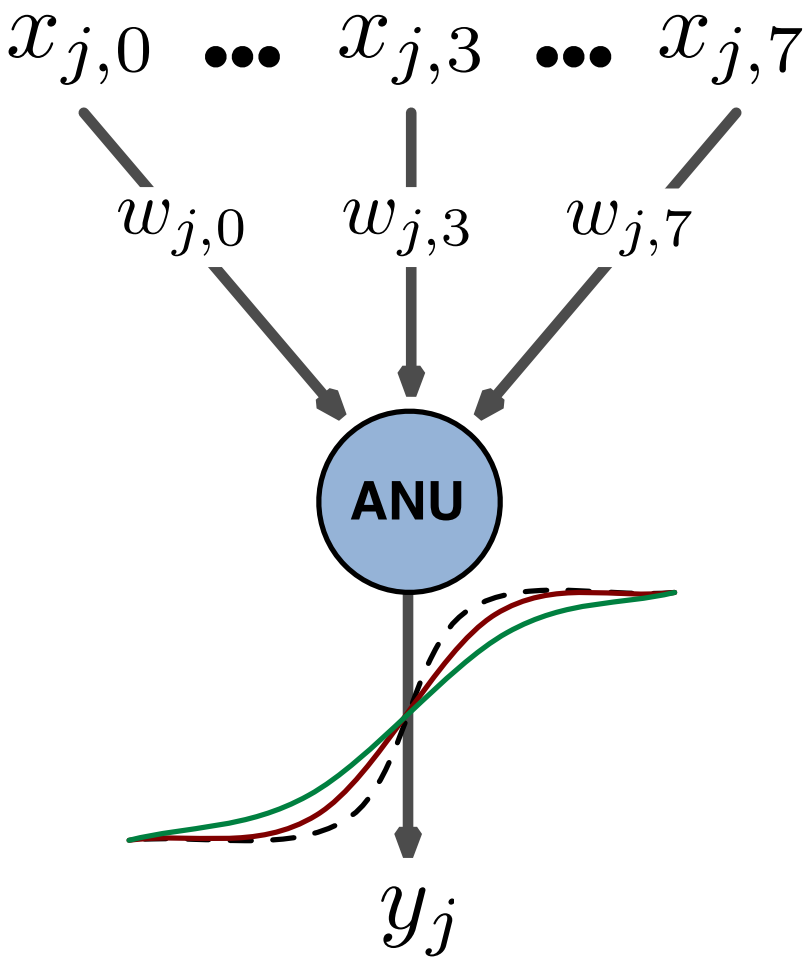


Fully-Precise Network



Continuous-Discrete Learning Method (CDLM), E. Fiesler, 1990

(2) Training with topology restrictions and non-idealities



1) **Robust** to the topology restrictions

2) Tolerate a more **shallow sigmoid** activation steepness over all applications

Resilient Back Propagation (RPROP), M. Riedmiller, 1993

Measurements

Signal Processing, Robotics, 3D Gaming, Financial Analysis, Compression, Machine Learning, Image Processing

Analog A-NPU with 8 Analog Neurons

- Transistor-Level HSPICE Simulation
- Predictive Technology Models (PTM), 45nm
- Vdd: 1.2 V, f: 1.1 GHz

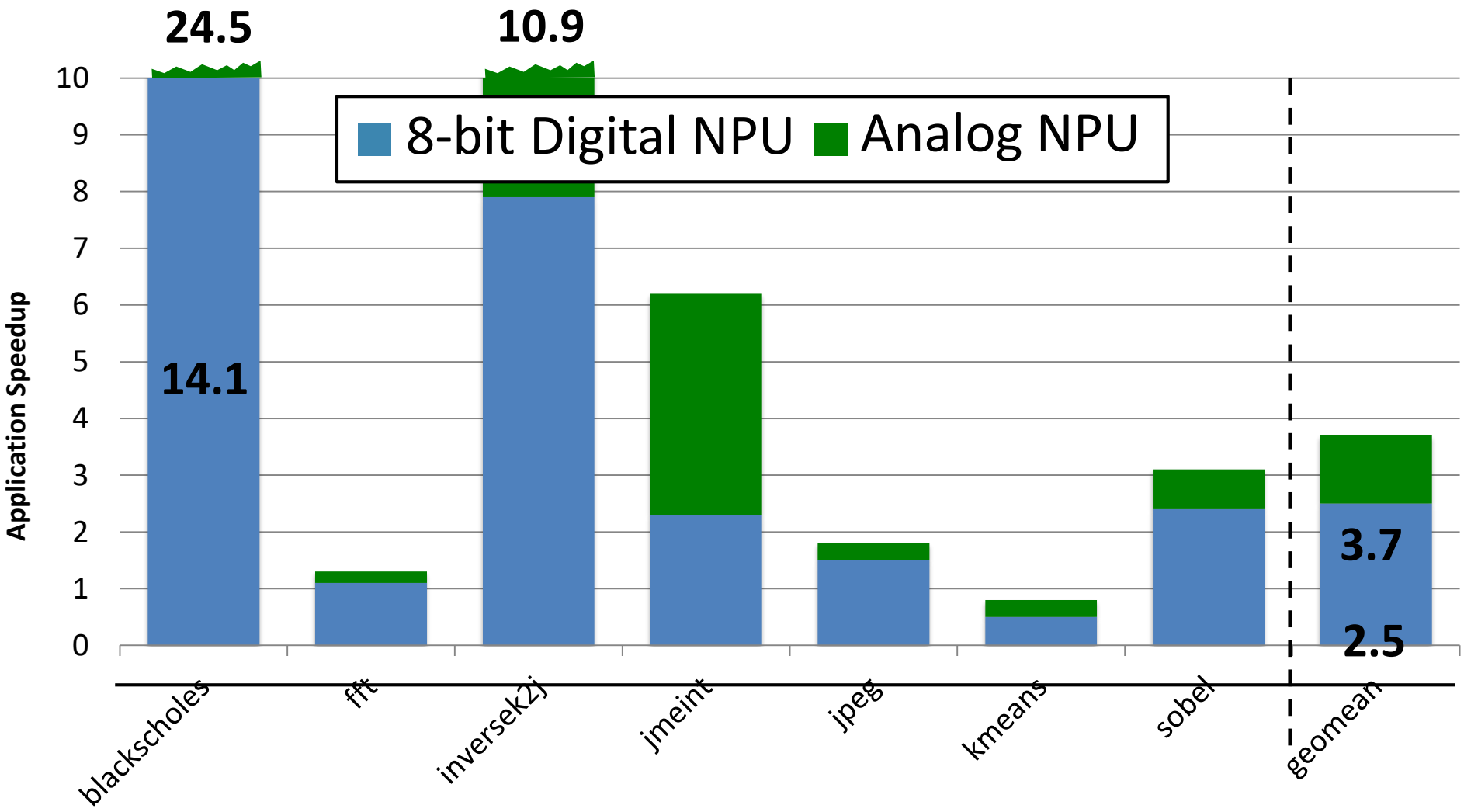
Digital Components

- Power Models: McPAT, CACTI, and Verilog

Processor Simulator

- Marssx86 Cycle-Accurate Simulation
- Intel Nehalem-like 4-wide/5-issue OoO processor
- Technology: 45 nm, Vdd: 0.9 V, f: 3.4 GHz

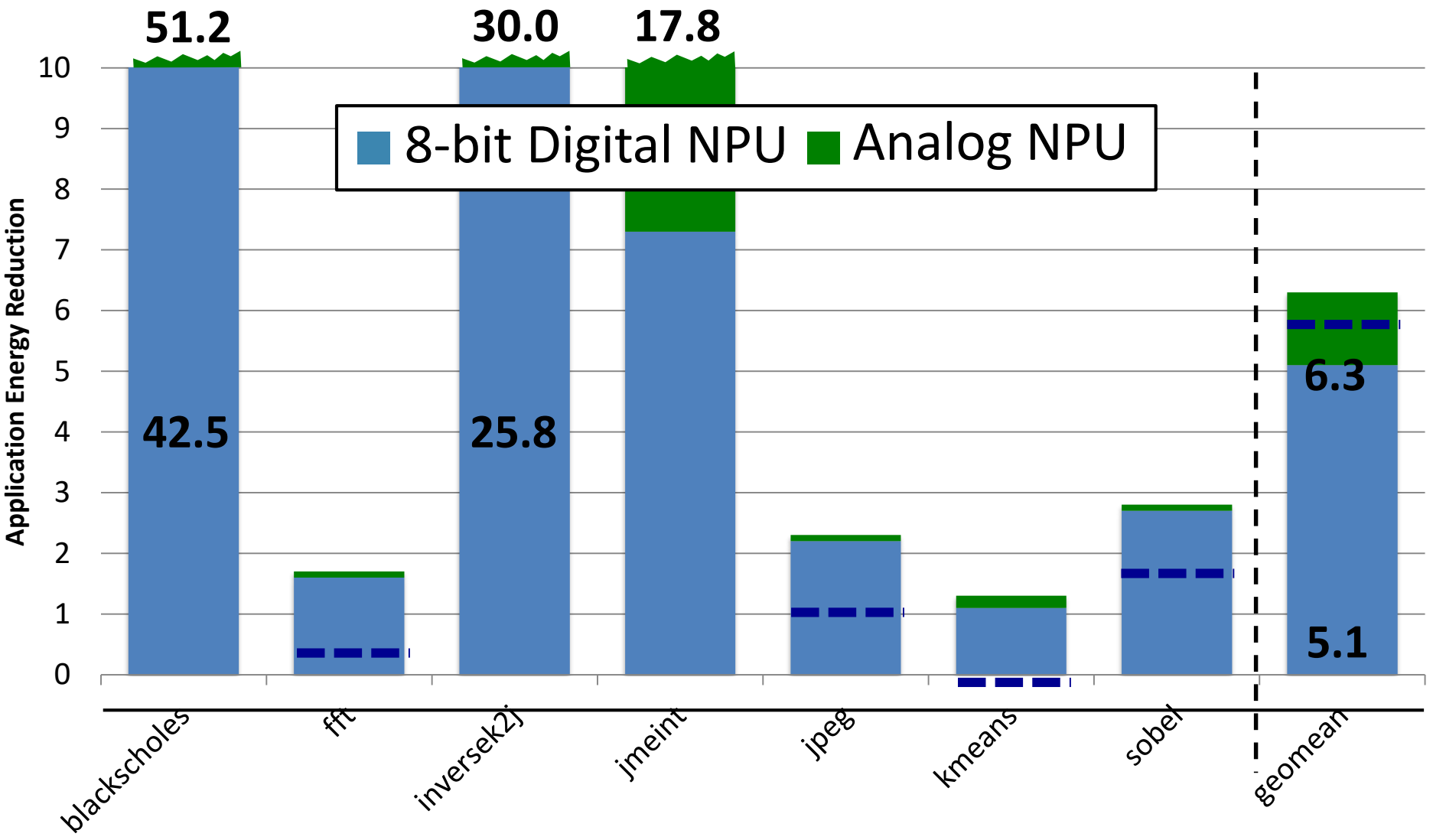
Speedup



Ranges from 0.8× to 24.5× with Analog NPU

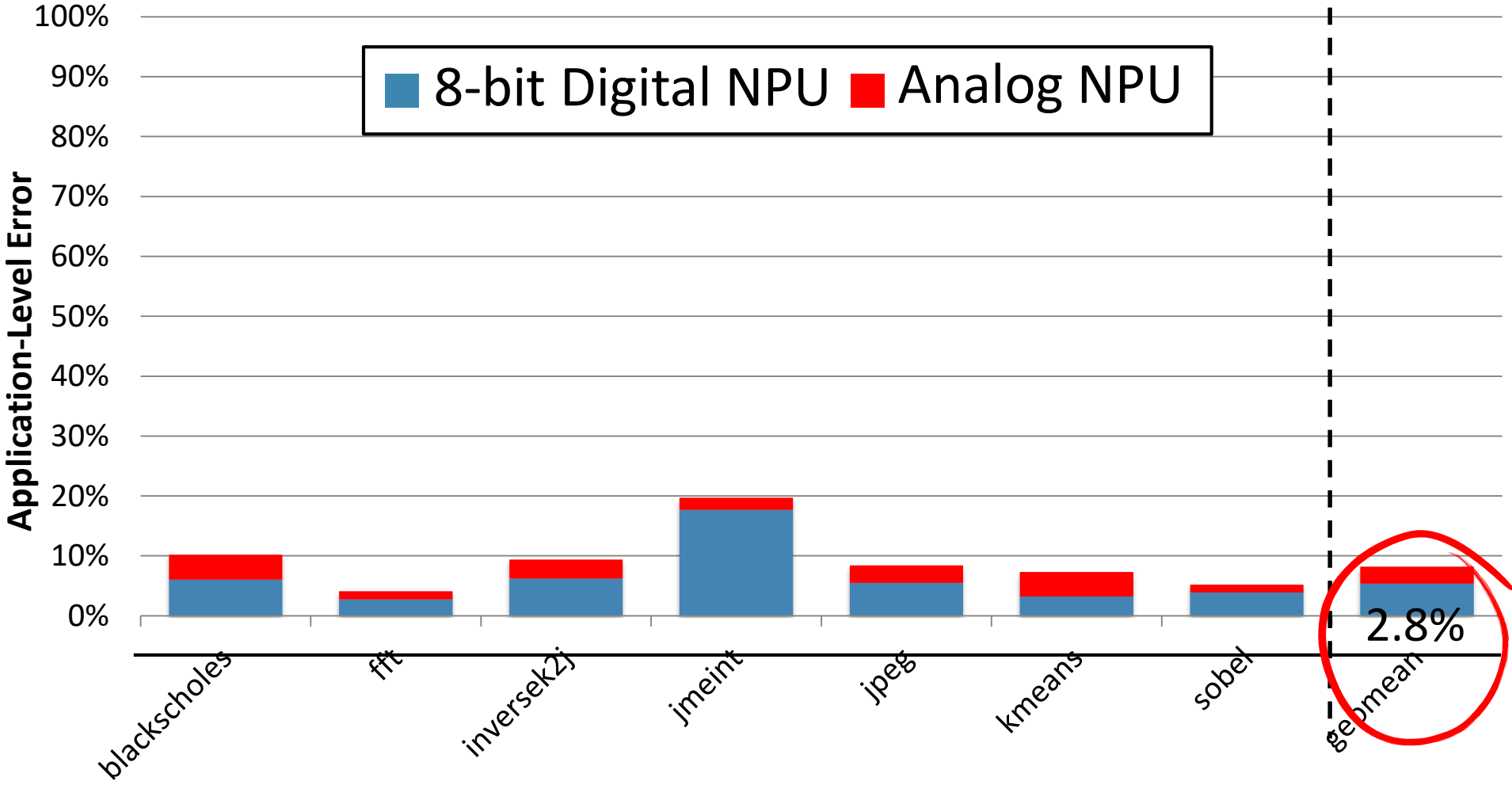
1.2× increase in application speedup with Analog over Digital NPU

Energy Savings



Energy saving with Analog NPU is very close to ideal case (6.5x)

Application quality loss



Quality loss is below 10% in all cases but one

Based on application-specific quality metric

What is left?

Energy Reduction

46%

Speedup

We can not reduce the energy of the computation much more.

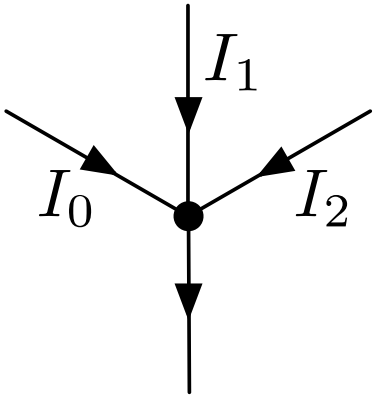
3.7x × 6.3x

Speedup **Energy Reduction**

≈23x

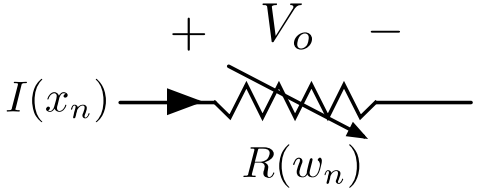
Energy-Delay Product

Quality Degradation: Avg. 8.2%, Max. 19.7%



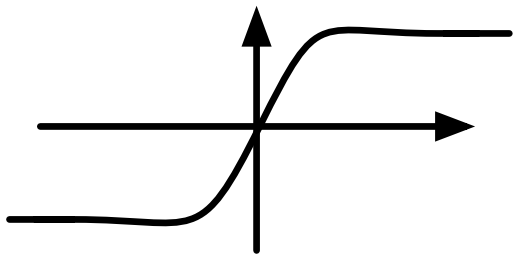
$$I_{out} = I_0 + I_1 + I_2$$

Kirchhoff's Law



$$V_o = I(x_n).R(w_n)$$

Ohm's Law



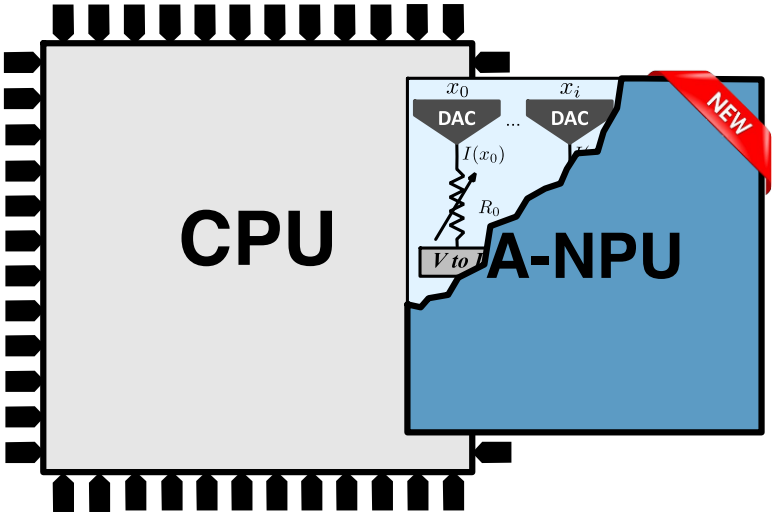
Saturation Property of Transistors

It is still the beginning...

1) **Broad applicability** of the analog computation

2) Prototyping and integrating A-NPU within **noisy** high performance processors

3) Reasoning about the **acceptable level of error** at the programming level



Backup Slides

Area Breakdown

Sub-circuit	Area
A-NPU	
8x8-bit DAC	3,096 T
8xResistor Ladder (8-bit weights)	4,096 T + 1 KΩ (≈450 T)
8xDifferential Pair	48 T
I-to-V Resistors	20 KΩ (≈30 T)
Differential Amplifier	244 T
8-bit ADC	2,550 T + 1KΩ (≈ 450)
Total	**≈10,964 T**
D-NPU	
8x8-bit multiply-adds	≈56,000 T
8-bit Sigmoid lookup table	16,456 T
Total	**≈72,456**

6.6x fewer transistors in the analog neuron implementation

Power Breakdown

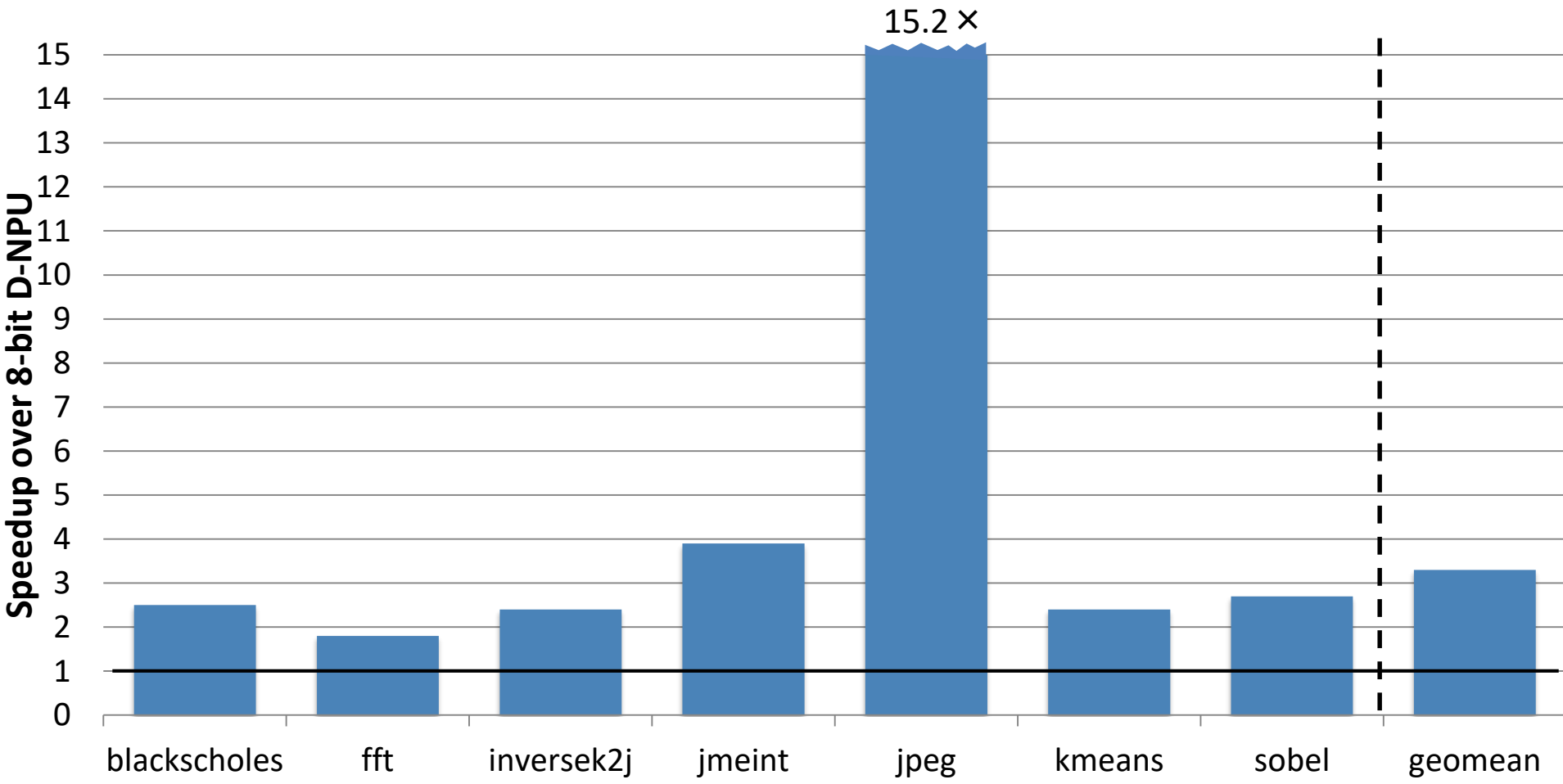
Sub-circuit	Percentage of total power
A-NPU	
SRAM-accesses	13%
DAC-Resistor Ladder-Diff Pair-Sum	54%
Sigmoid-ADC	33%

Power numbers vary with applications

Applications

Financial **blackscholes** 309 x86 instructions 97.2% dynamic instructions	6→8→8→1 Error: 10.2%
Signal Processing **fft** 34 x86 instructions 67.4% dynamic instructions	1→4→4→2 Error: 4.1%
Compression **jpeg** 1,257 x86 instructions 56.3% dynamic instructions	64→16→8 →64 Error: 8.4%
Robotics **inversek2j** 100 x86 instructions 95.9% dynamic instructions	2→8→2 Error: 9.4%
Machine Learning **kmeans** 26 x86 instructions 29.7% dynamic instructions	6→8→4→1 Error: 7.3%
3D Gaming **jmeint** 1,079 x86 instructions 95.1% dynamic instructions	18→32→8 →2 Error: 19.7%
Image Processing **sobel** 88 x86 instructions 57.1% dynamic instructions	9→8→1 Error: 5.2%

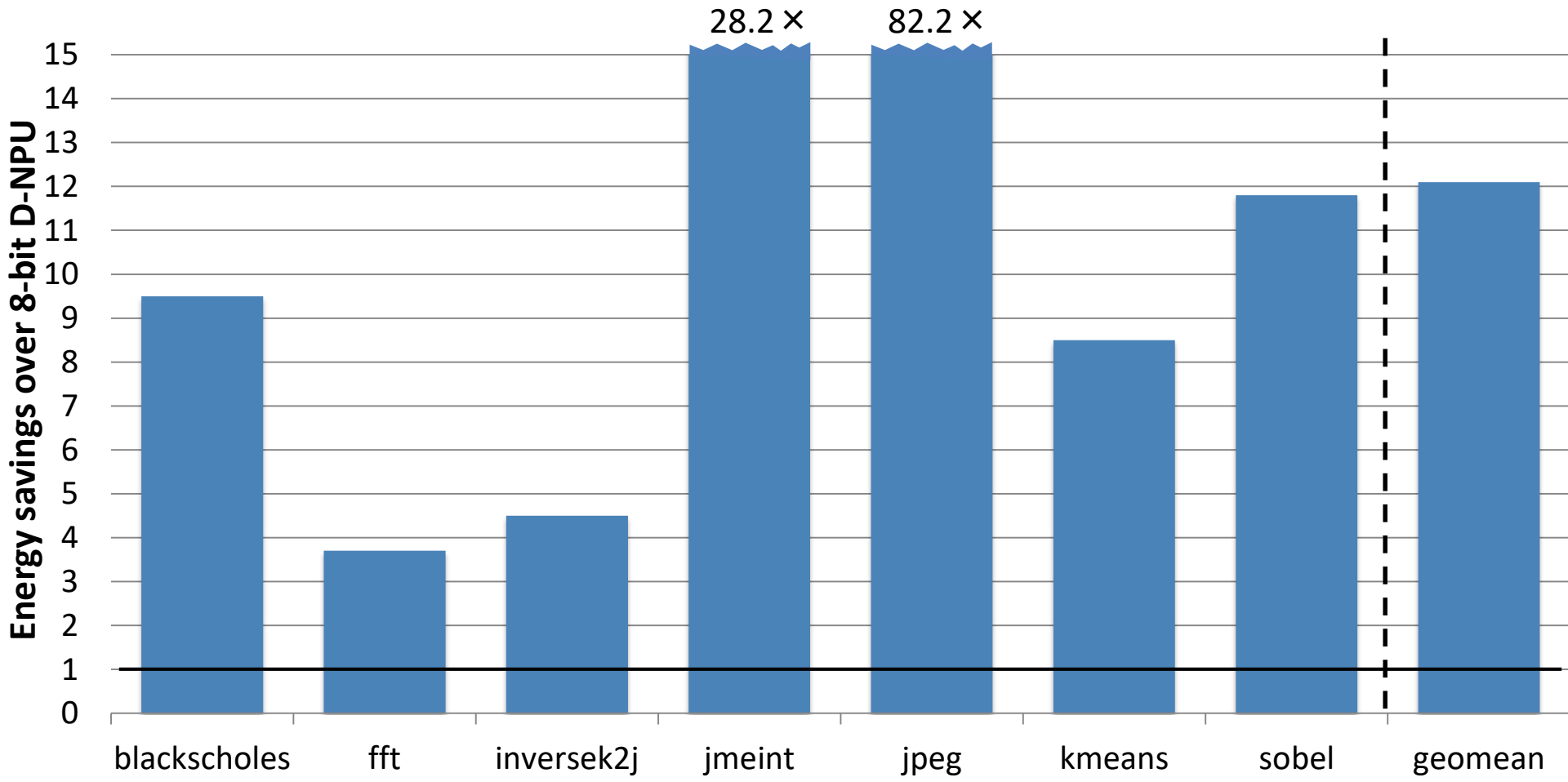
Speedup with A-NPU over 8-bit D-NPU



3.3× geometric mean speedup

Ranges from 1.8× to 15.2×

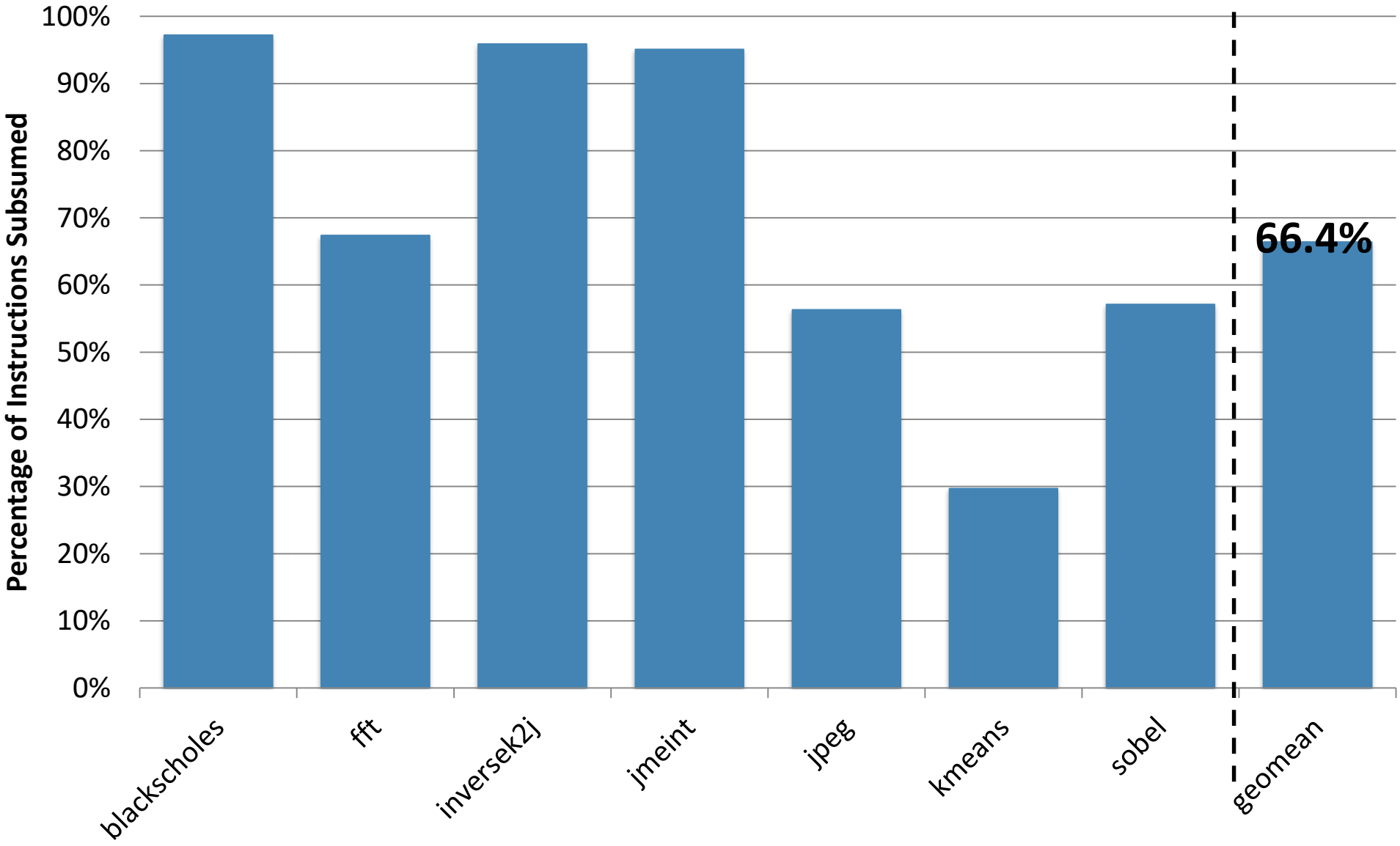
Energy savings with A-NPU over 8-bit D-NPU



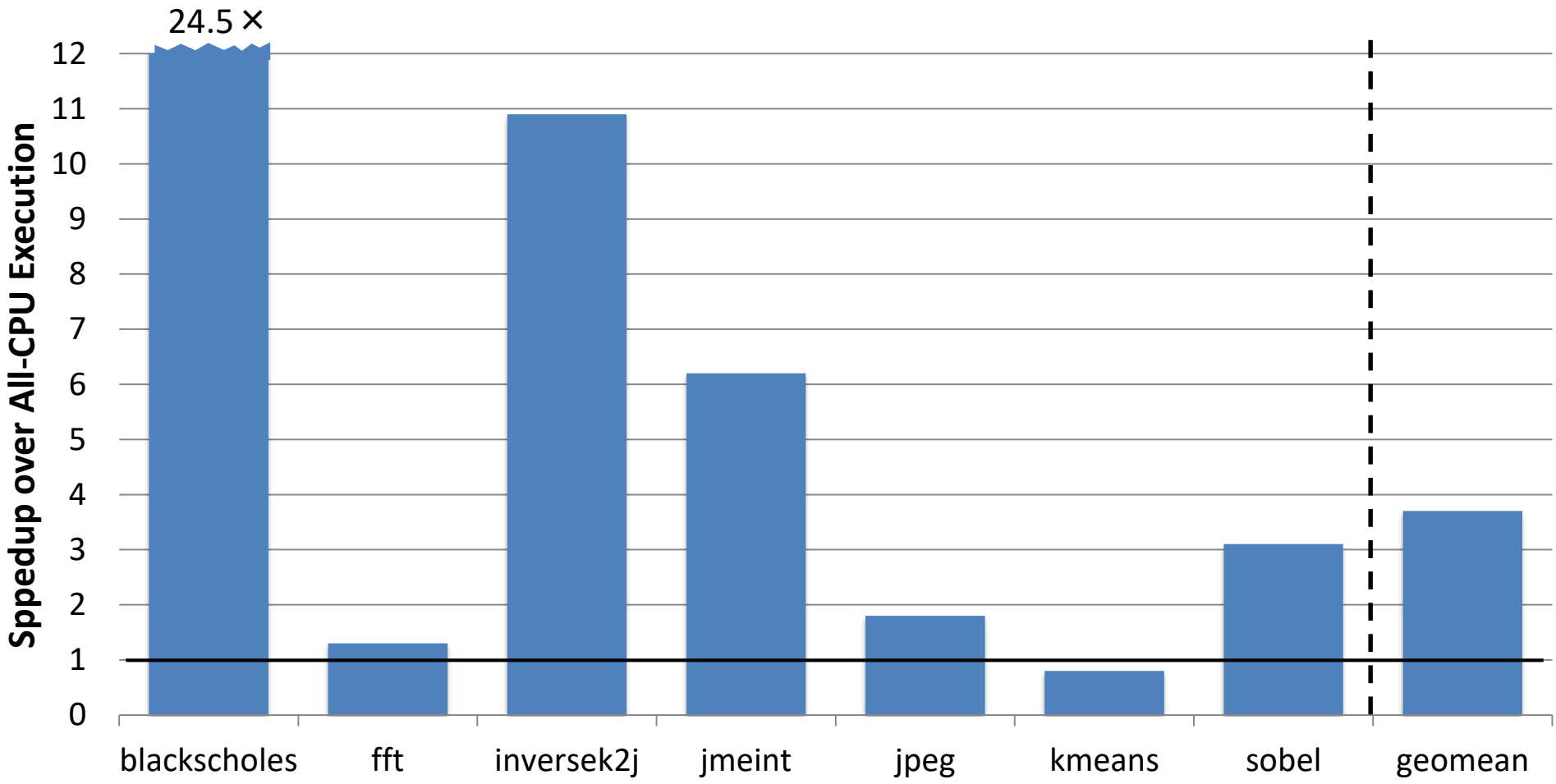
12.1× geometric mean speedup

Ranges from 3.7× to 82.2×

Dynamic Instruction Reduction



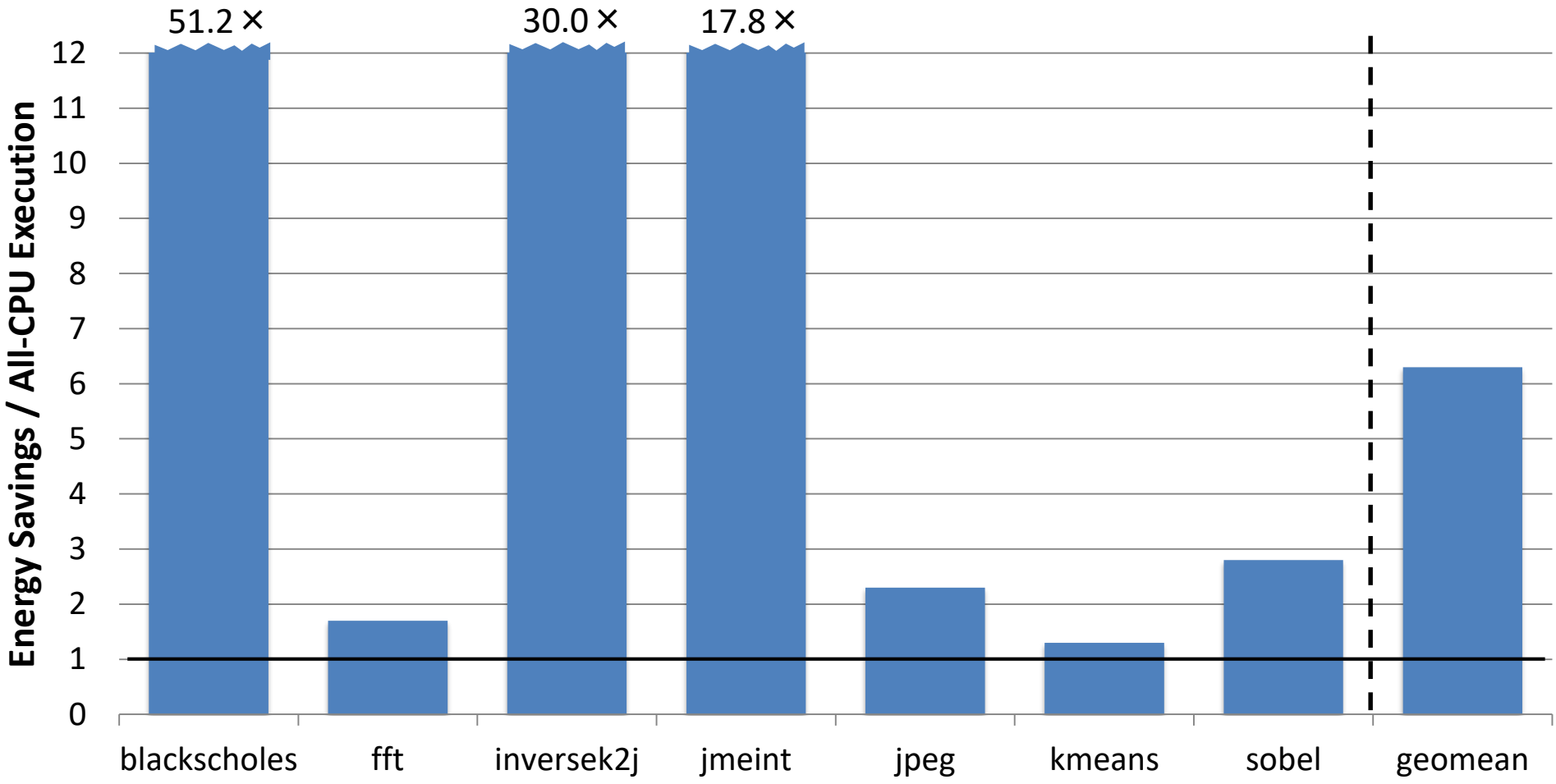
Speedup with A-NPU acceleration



3.7× geometric mean speedup

Ranges from 0.8× to 24.5×

Energy savings with A-NPU acceleration



6.3× geometric mean energy reduction

All benchmarks benefit