PRU: Probabilistic Reasoning processing Unit for resource-efficient AI

Nimish Shah, Laura I. Galindez Olascoaga, Wannes Meert, and Marian Verhelst

nimish.shah@esat.kuleuven.be

KU LEUVEN

micas

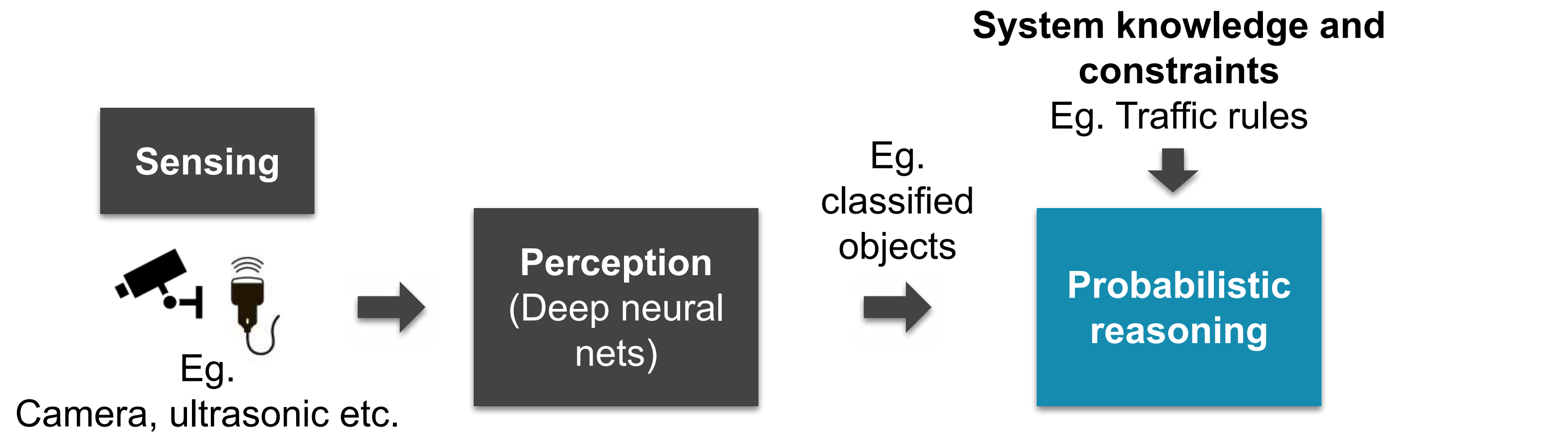
European Research Council

Motivation

Combining probabilistic reasoning techniques with Deep learning is crucial to handle real-world uncertainty and constraints

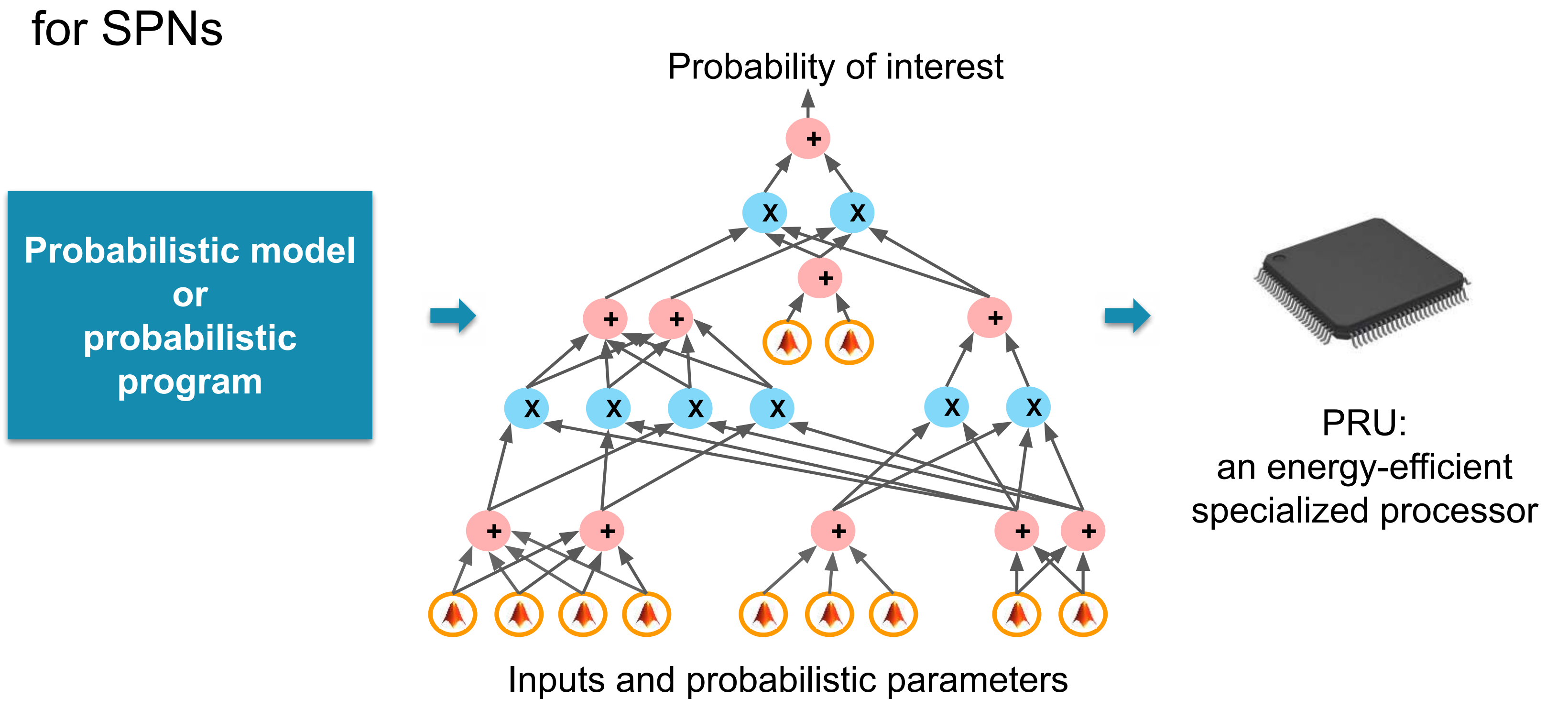
Probabilistic programming + DL frameworks: PYRO DeepProbLog Edward

Probabilistic + DL system for safety-critical applications like self-driving vehicles:



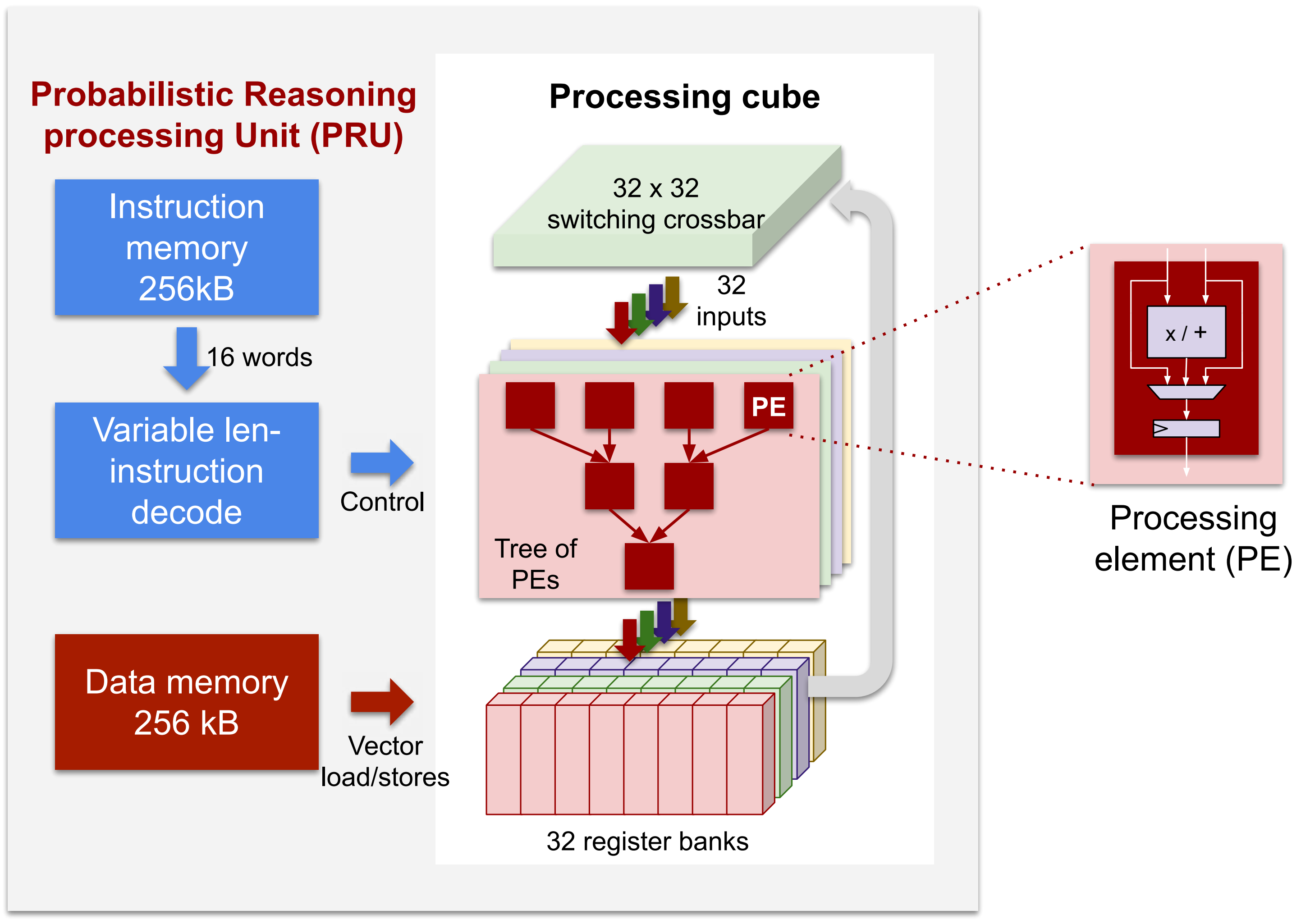
Sum-Product Networks (SPN)

- Probabilistic models are typically implemented as a network of sums and products called Sum-product network (SPN)
- SPNs are not suitable for GPUs or vector processors due to highly irregular graph structure
- Aim of this work is to develop PRU, an energy-efficient custom processor, for SPNs


Probability of interest
Probabilistic model or probabilistic program
Inputs and probabilistic parameters
PRU:
an energy-efficient specialized processor

Sum-Product Networks (SPN)

PRU architecture



Hardware properties

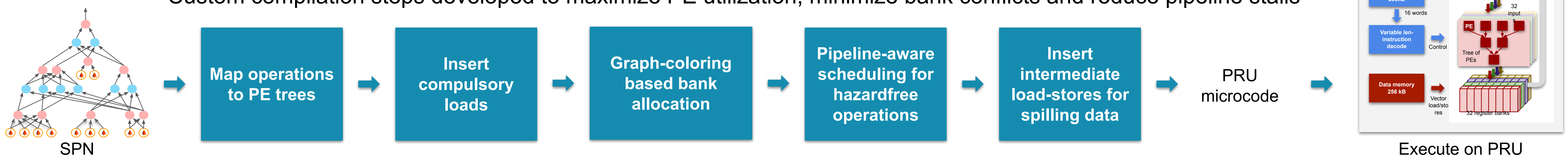
- Processing elements (PEs) arranged in a tree structure for efficient data reuse
- Hardware flexibility to support indirect-addressing based computation in SPNs:
  - 32 independent register banks
  - Switching crossbars for efficient shuffling of data during register read
- Instruction programmable to execute any SPN
- Automatic register writing scheme to avoid write address field in instructions

Instruction set

Inst. name	Function	Len (x32b)
ld	Loads a vector of 32 words from memory	2
st	Stores a word from 8 register banks to memory	4
sh	Shuffle 8 words across register banks	4
main	Main tree instruction that performs the compute by configuring the trees	16

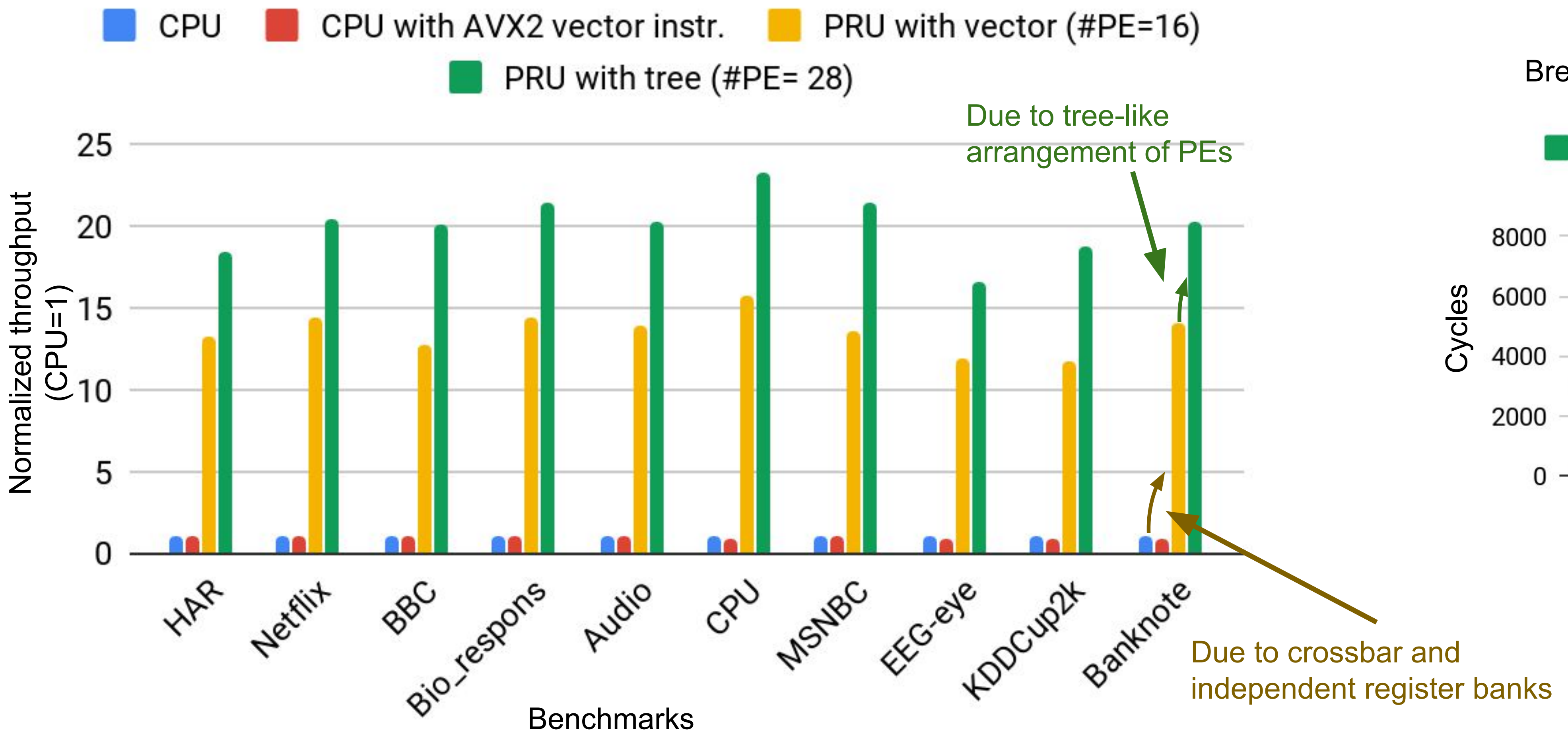
Compilation

Custom compilation steps developed to maximize PE utilization, minimize bank conflicts and reduce pipeline stalls


SPN
Map operations to PE trees
Insert compulsory loads
Graph-coloring based bank allocation
Pipeline-aware scheduling for hazardfree operations
Insert intermediate load-stores for spilling data
PRU microcode
Execute on PRU

Results

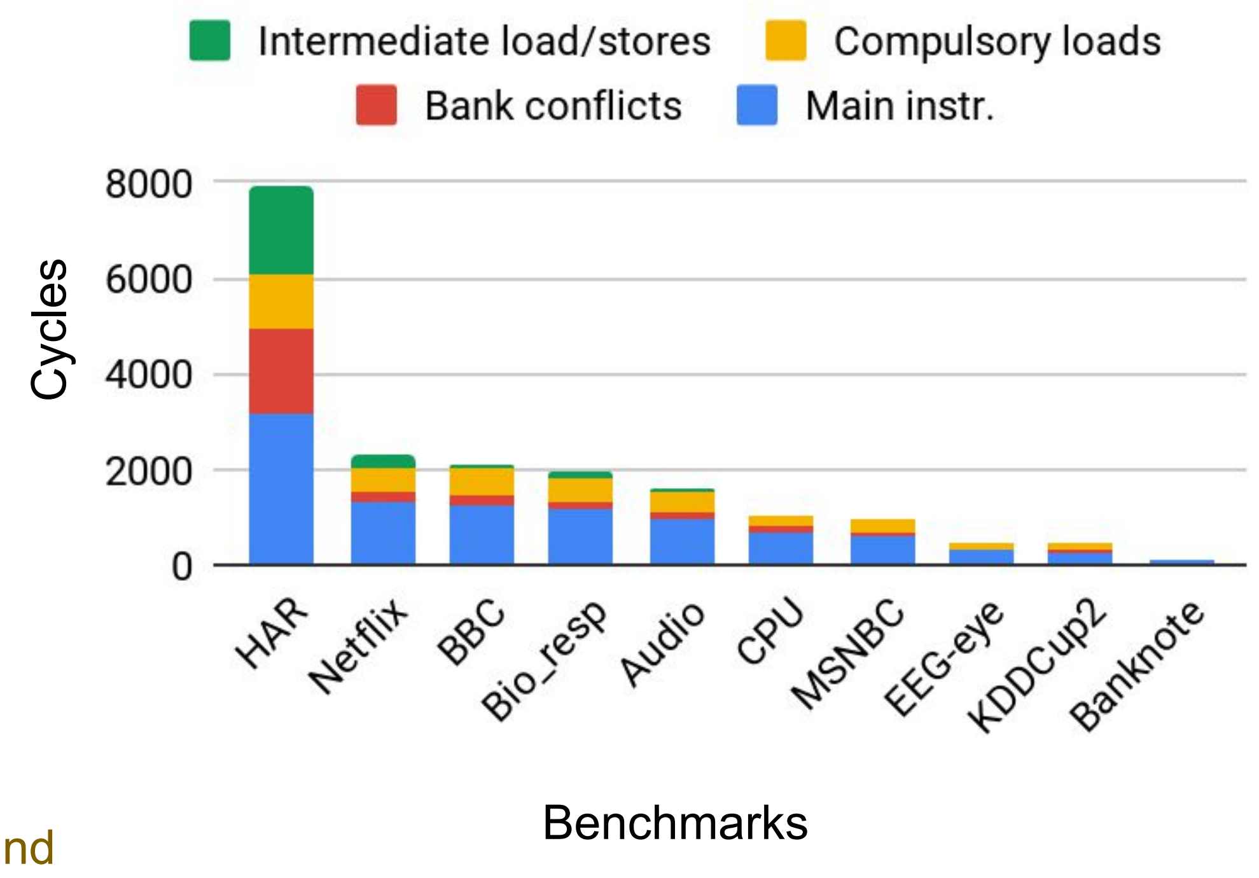
PRU achieves at least 15x higher throughput than Intel i5 CPU



Due to tree-like arrangement of PEs

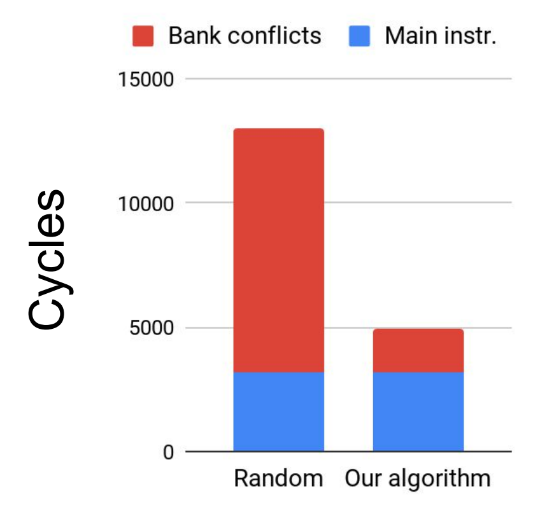
Due to crossbar and independent register banks

Breakup of execution time for different benchmarks



Impact of using a graph-coloring based bank-allocation scheme

Significant reduction in bank-conflicts



Bank allocation scheme

Conclusion

Future intelligent systems will have, besides CPU, GPU and NPU, also a PRU to support reasoning tasks at 15x improved efficiency and throughput