No-ISA is the Best ISA

Shreeyash Pandey, Rishik Ram Jallarapu

Vicharak, India @ vicharak.in

28th September, 2024

About us

Goals:

- Introduce software-level reconfigurability to hardwares
- Develop consumer and industrial computing hardware for a new ecosystem: desktop, mobile, and cloud-compatible

Vicharak's team of 50 engineers and thinkers is committed to realizing this vision across all verticals.
- Akshar Vastarpara (Founder and CEO, Vicharak)

Contents

- Chapter 1 - Motivations for our Work
- Chapter 2 - Introduction To Reconfigurable And Heterogeneous Computing
- Chapter 3 - Need for modern EDA Compilers
- Chapter 4 - Work Done Towards Implementation

Chapter 1 Motivations for our Work

Problems Facing Modern Compute

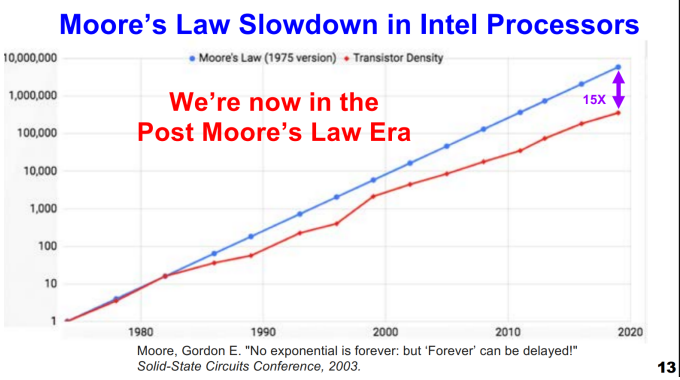

Figure: Moores Law Slowdown

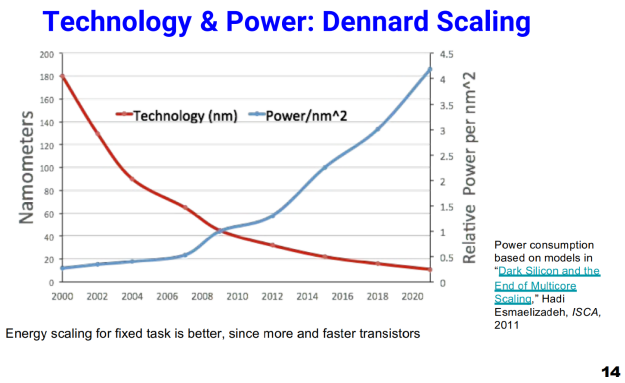

Figure: End of Dennard Scaling

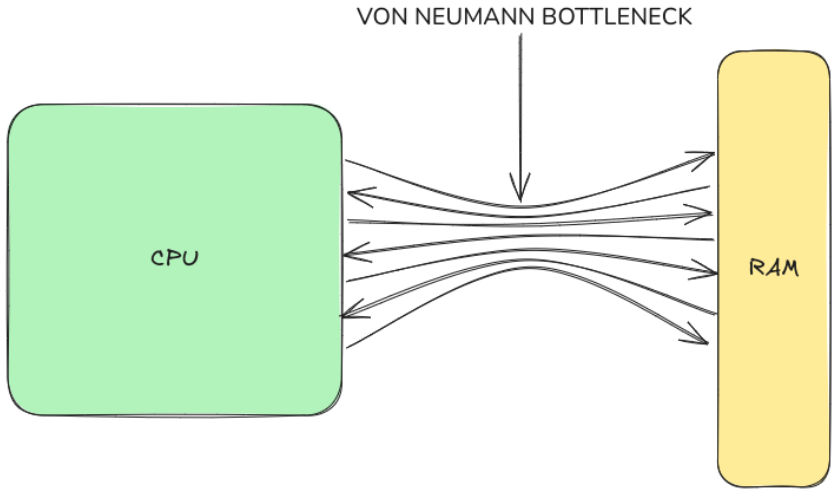

Figure: Von-Neumann Bottleneck

Overview of Modern Compute

- Processor trade-offs: ASICs (high upfront costs), CPUs (performance limitations), GPUs (power-hungry, less flexible)
- Question: Should modern compute be limited to CPUs/GPUs and select ASICs?
- Four scenarios where current compute falls short:
  1. Many peripherals with real-time compute needs
  2. Unusual number representations (e.g., neural network quantization)
  3. Novel architectures for existing problems (e.g., KANs)
  4. Power-efficient yet flexible solutions

How to deal with this?

- The free lunch afforded by hardware improvements over years is coming to an end.
- New, creative, first-principles architectures and compute infrastructure need to be designed along with the software abstractions to use them.

Chapter 2 Introduction To Reconfigurable And Heterogeneous Computing

Setting the stage

Key concepts:

1. Reconfiguration: Reprogramming to implement new circuits (e.g., FPGAs)
2. Heterogeneity: Integrating diverse processors (CPUs/GPUs/DSPs/ASICs) to complement rather than replace existing compute

Challenges:

1. FPGA integration with traditional software is cumbersome
2. Custom FPGA hardware development has a steep learning curve

Reconfigurability: An Introduction to FPGAs

1. Digital circuits are made up of gates and connections between them.
2. FPGAs are a grid of programmable cells and switches that allows a user to create new gates and connections after the IC has been fabricated.
3. Circuits for FPGAs are described using Hardware Descriptions Languages (HDLs) such as Verilog, VHDL.
4. High level description of a circuit is compiled into real hardware (i.e. a representation that only uses FPGA primitives) by a "compiler".

Problem 1: Programming model for FPGAs

A true programming model for FPGAs would heavily exploit reconfigurability.
Proposal: Non-Von Neumann, flow-based reconfigurable architecture

- No instructions or ISA
- Data flows through the chip, transformed by configured hardware

Comparison with a Von-Neumann Computer

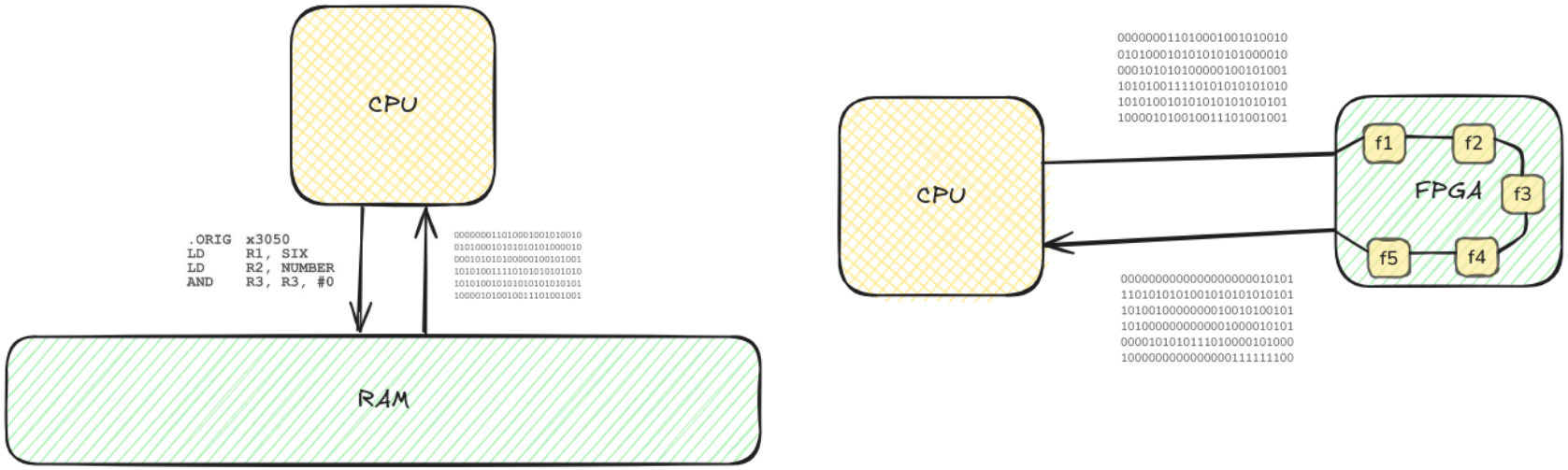

Figure: a) A Von-Neumann Computer b) A Flowing Reconfigurable Computer

An exemplary DSL for reconfigurable compute architectures

```
1       Base *input = new TensorSend("ts");
2       Base *proc_one = new JPEGEncoderTillDct(
            input, "jpeg_encoder");
3       Base *proc_two = new MlEngineCore(
            proc_one, "ml_core");
4       Base *output = new TensorReceive(
            proc_two, "tr");
5       Model m1 = new Model(input, output);
6       m1->compute(input);
```

An exemplary DSL for reconfigurable compute architectures (2)

Continue

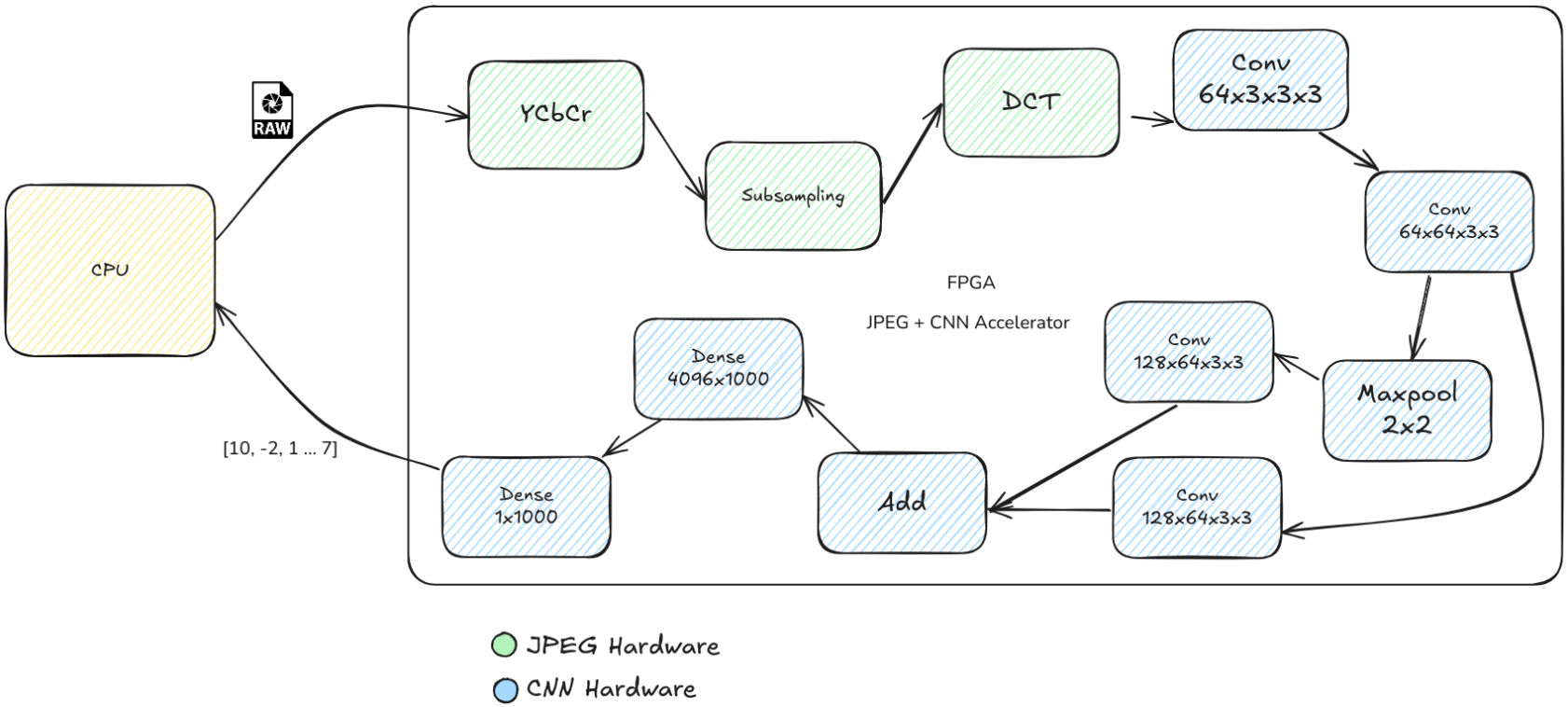

Figure: Data flow for JPEG (Partial) + CNN inference

Chapter 3 Need for modern EDA Compilers

Problem 2: Writing Hardware Is Hard

Current state:

- Proprietary tools and architectures
- NP-Hard and NP-Complete problems in the flow
- Gap between CAD research and production-ready tools

FPGA CAD Toolflow

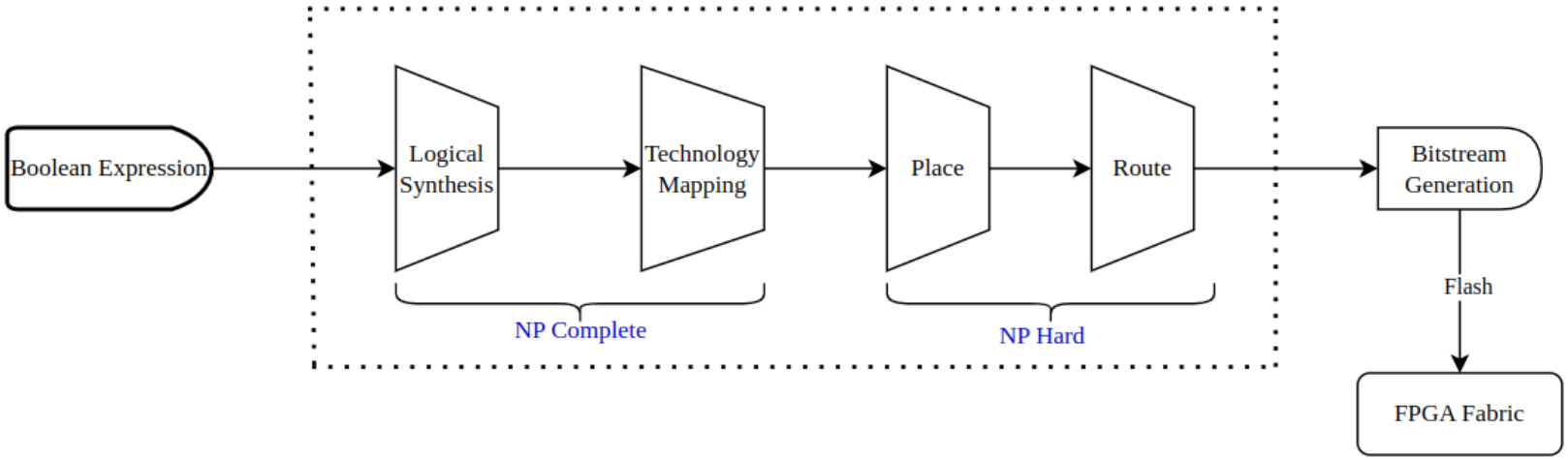

Figure: FPGA CAD Tool Flow

FPGA CAD Toolflow

- Logical Synthesis: HDL to primitive gates (AND/OR).
- Technology Mapping: Mapping generic AND/OR gates to k-LUTs.
- Placement: Placing k-LUTs in a constrained grid representing the FPGA based on a cost model
- Routing: Connection placed LUTs so that the latency is minimum
- Bitstream Generation: Generating a final bitstream that the FPGA can understand.

Optimization Opportunities for EDA Compilers

Optimization opportunities:

- Direct netlist operations in DSL compiler
- Caching optimzations for faster incremental compilation
- GPU-accelerated routing

Chapter 4 Work Done Towards Implementation

Realizing this goal

Vaaman: A reconfigurable heterogeneous single board computer (SBC)

1. Custom-designed hardware
2. Integrates FPGA for reconfigurability
3. Heterogeneous as it integrates CPU with FPGA

The Hardware (Vaaman)

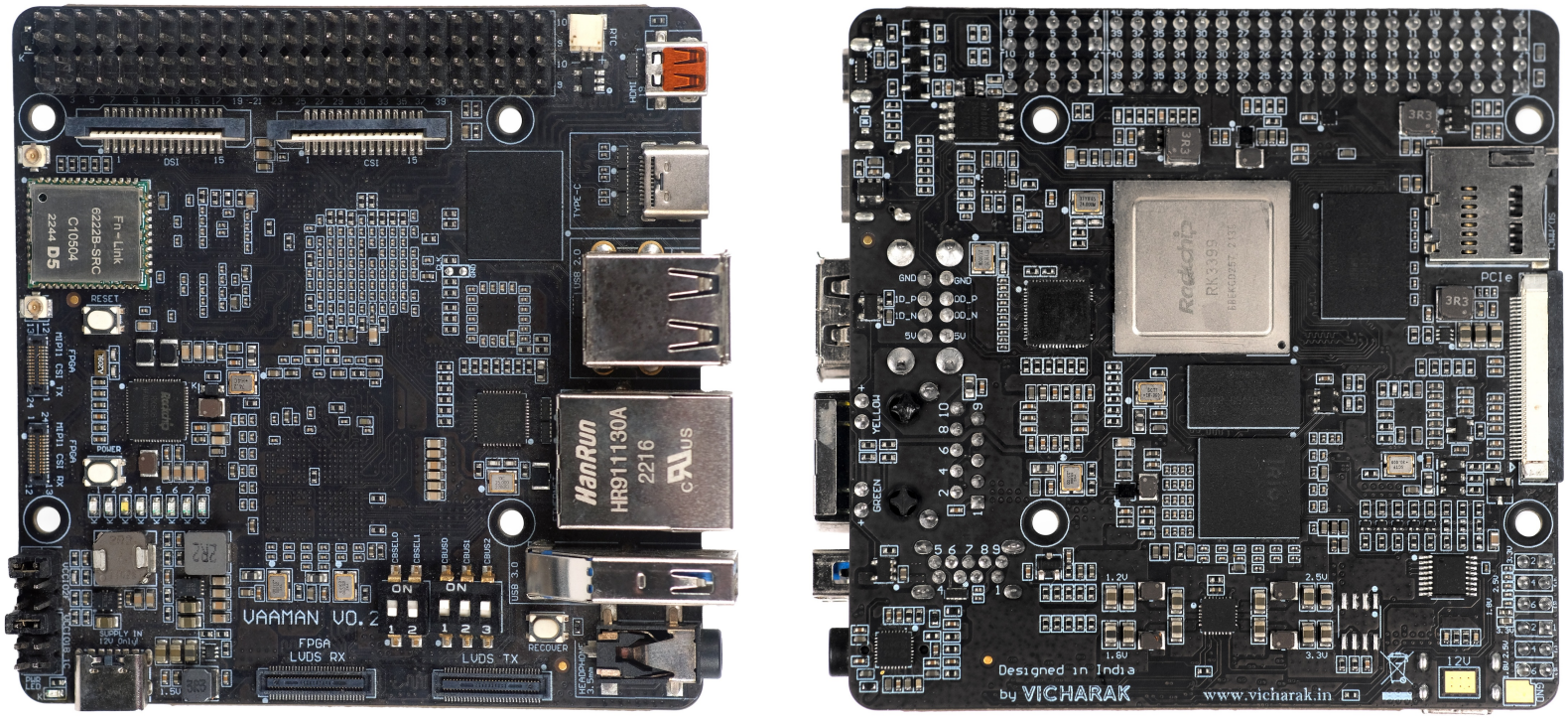

Figure: Vaaman: A heterogenous SBC
`https://shorturl.at/5y9QA`

Features

- **Gati: CNN accelerator using FPGA as co-processor**
  1. Systolic-Array based MAC engine
  2. Macro-ISA for control flow
  3. Compiler and runtime for model partitioning and execution
- **Periplex: On-the-fly peripheral generation**
  1. Software-defined peripherals on FPGAs
  2. JSON-driven configuration
  3. Linux kernel driver integration

Conclusion

1. Reconfigurable architectures address von Neumann bottleneck and modern compute challenges
2. Heterogeneous approach enables integration with existing infrastructure
3. Key challenges: developing reconfigurability-focused programming models and efficient hardware compilers
4. Solutions include novel DSLs, optimized EDA tools, and GPU acceleration

Extended slides and more stuff:

`github.com/vicharak-in/noisa`

We are hiring! Email us at:

`join-the-team@vicharak.in`