Custom Chip Design
Open Source Chip Development using OpenPower Microwatt

Satyadhyan Chickerur
and
Ganesan Narayanasamy

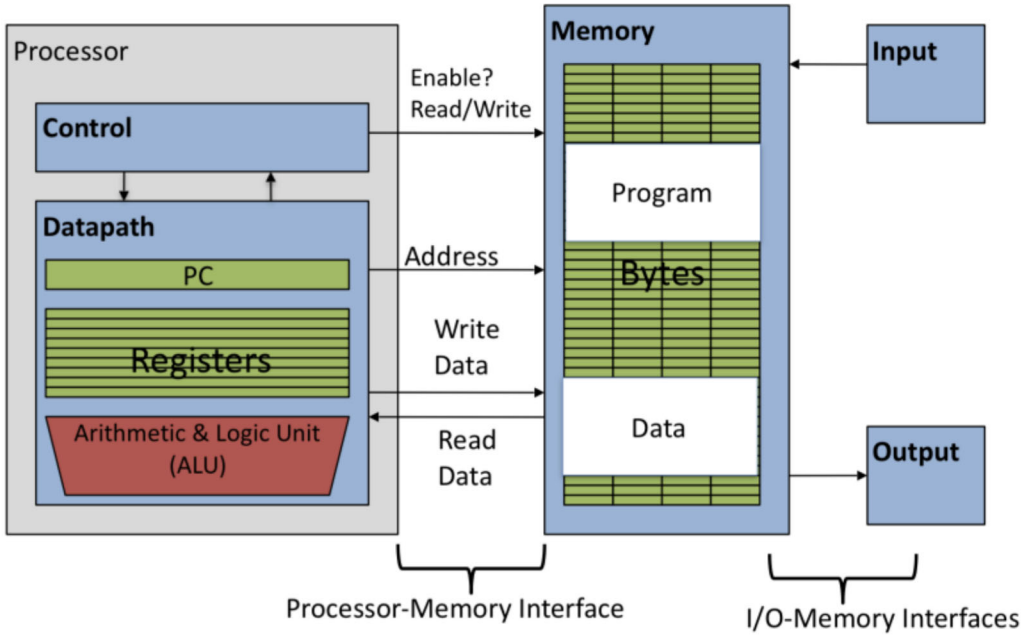
27 June 2022

Agenda

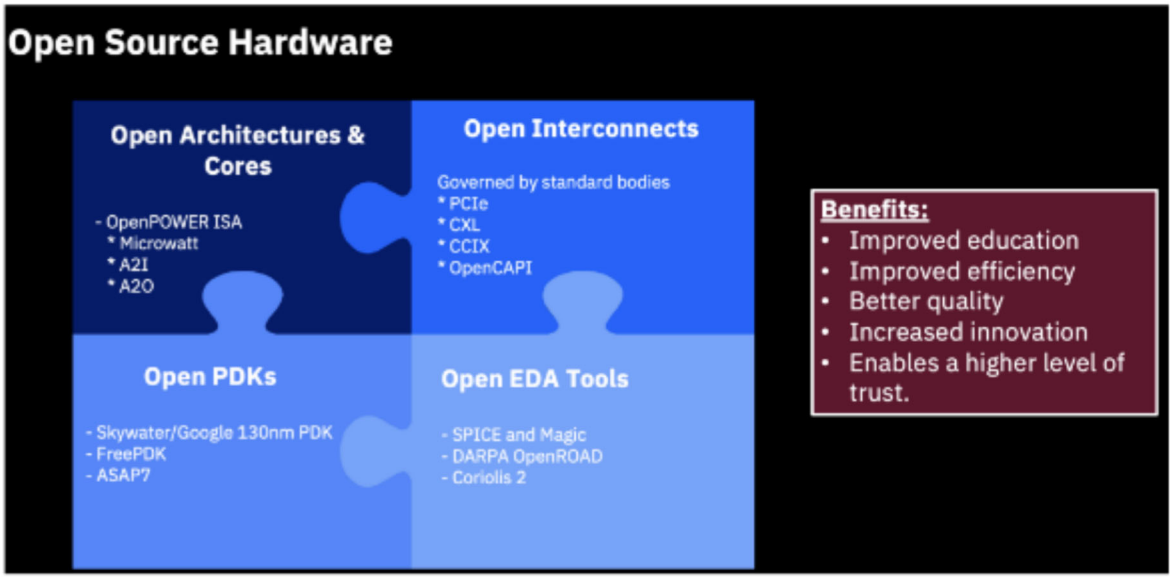
- Components of a Computer and it's relation to the Open Source Ecosystem
- Levels of representation
- Instruction Set Archtecture (ISA) - A Perspective based on complexity
- SoftCore Processors
- Microwatt - Brief Introduction
- Demo - Microwatt
- References

KLE Technological University
Creating Value
Leveraging Knowledge

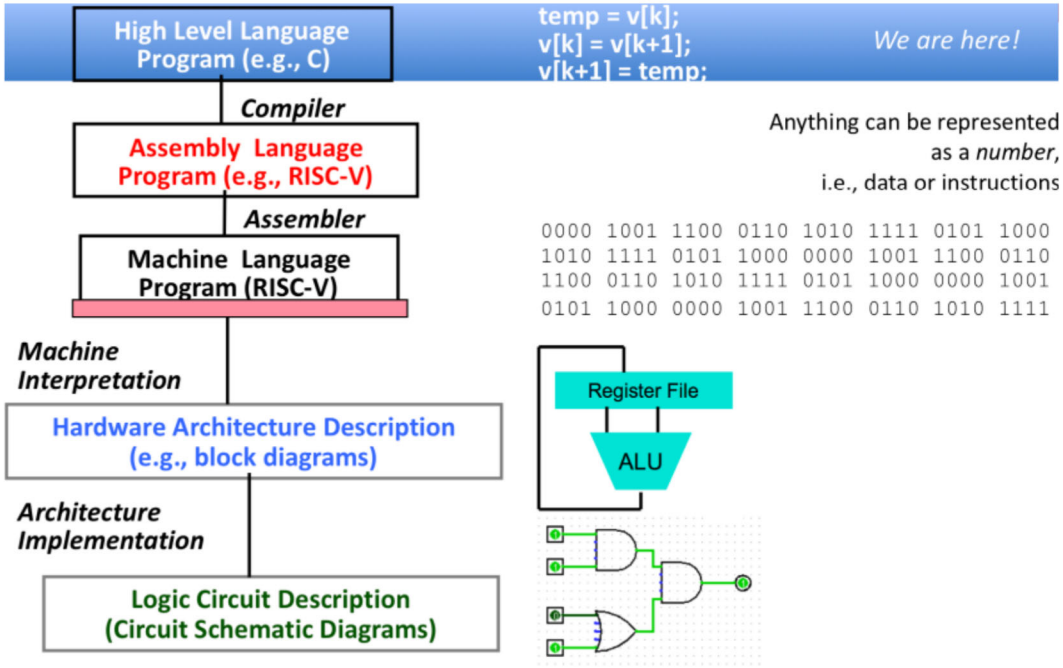
Components of a Computer and its' relation to the Open Source Ecosystem



OpenPOWER


Open Source Hardware
Open Architectures & Cores
- OpenPOWER ISA
* Microwatt
* A2I
* A2O
Open Interconnects
Governed by standard bodies
* PCIe
* CXL
* CCIX
* OpenCAPI
Open PDKs
- Skywater/Google 130nm PDK
- FreePDK
- ASAP7
Open EDA Tools
- SPICE and Magic
- DARPA OpenROAD
- Coriolis 2
Benefits:
• Improved education
• Improved efficiency
• Better quality
• Increased innovation
• Enables a higher level of trust.

Levels of representation



Instruction Set Architecture (ISA)-1

- Job of a CPU (Central Processing Unit, aka Core): execute instructions
- Instructions: CPU's primitives operations
  - Instructions performed one after another in sequence
  - Each instruction does a small amount of work (a tiny part of a larger program).
  - Each instruction has an operation applied to operands,
  - Might be used change the sequence of instruction.
- CPU's particular set of instructions implements an Instruction Set Architecture (ISA)
  - Examples: Intel x86, AMD, ARM, IBM PowerPC, Intel IA64, ... RISC-V, OpenPower-microwatt
- CPUs belong to "families," each implementing its own set of instructions
  - CISC
  - RISC

Instruction Set Architecture (ISA)-2

- Early trend: add more instructions to new CPUs for elaborate operations
  - Made assembly language programming easier.
  - VAX architecture had an instruction to compute polynomials! result = C[0]+x**0 + x*(C[1] + x*(C[2] + ... x*C[d]))
- RISCphilosophy(CockeIBM,PattersonUCB,Hennessy Stanford, 1980s) – Reduced Instruction Set Computing
  - Keep the instruction set small and simple, in order to build fast hardware
  - Let compiler generate software do complicated operations by composing simpler ones

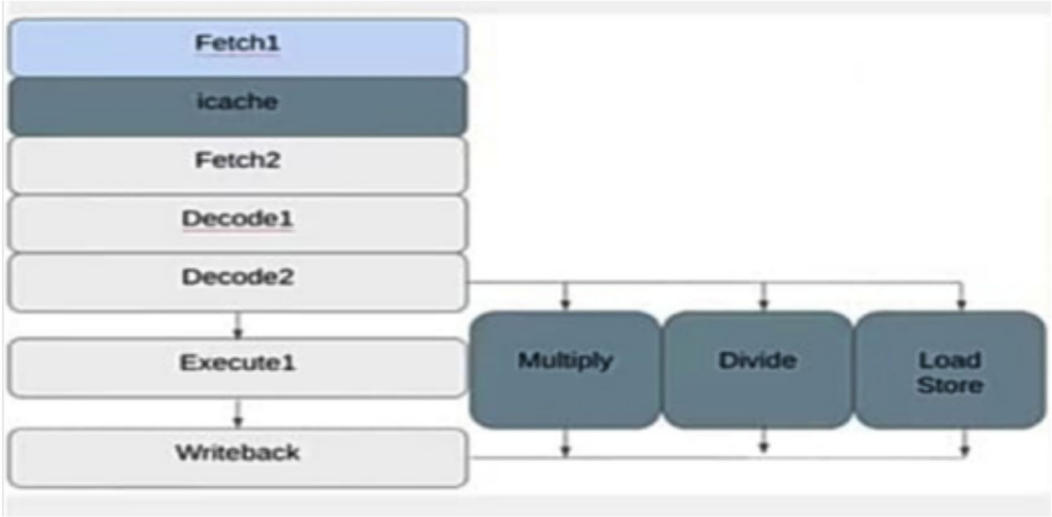
SoftCore Processors

- A soft-core processor is one that is implemented entirely in the logic primitives of an FPGA using VHDL.
- core processor does not have the speeds or performance characteristics of a hard-core or an discrete processor.
- A soft-core processor allows a designer to add or subtract peripherals from the SoC with ease.
- A soft-core processor also offers the flexibility of configuring the core itself for the application.
- Using a SoPC solution also offers flexibility outside the FPGA while designing the circuit board.
- Since a SoPC exists in an FPGA, the pinout is flexible, which gives the board designer almost complete freedom with component placement while meeting the timing constraints

OpenPOWER Microwatt

- Microwatt is an open source soft processor core originally written in VHDL by Anton Blanchard at IBM.
- It adheres to the Power ISA 3.0 instruction set and can be run on FPGA boards, booting Linux, MicroPython and Zephyr.
- Microwatt is a tiny 64-bit bi-endian scalar integer processor core, implementing a subset of the Power ISA 3.0 instruction set.
- It has 32 x 64 -bit general purpose registers and 32 x 64 -bit floating-point registers.
- It uses Wishbone for the memory interface.
- No Memory management unit (MMU) and no floating-point unit.

Microwatt : Implementation



MicroWatt Core-1

- core:
  - decode_types.vhdl
  - wishbone_types.vhdl
  - common.vhdl
  - fetch1.vhdl
  - decode1.vhdl
  - helpers.vhdl
  - decode2.vhdl
  - register_file.vhdl
  - cr_file.vhdl
  - crhelpers.vhdl
  - ppc_fx_insns.vhdl
  - sim_console.vhdl
  - logical.vhdl
  - countzero.vhdl
  - control.vhdl

MicroWatt Core-2

- core:
  - execute1.vhdl
  - fpu.vhdl
  - loadstore1.vhdl
  - mmu.vhdl
  - dcache.vhdl
  - divider.vhdl
  - rotator.vhdl
  - pmu.vhdl
  - writeback.vhdl
  - insn_helpers.vhdl
  - core.vhdl
  - icache.vhdl
  - plru.vhdl
  - cache_ram.vhdl
  - core_debug.vhdl
  - utils.vhdl

KLE Technological University
Creating Value
Leveraging Knowledge

Demo- Snapshots-1

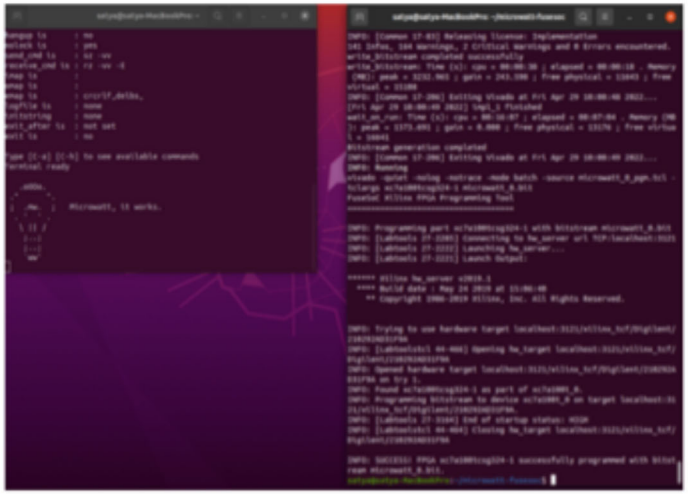
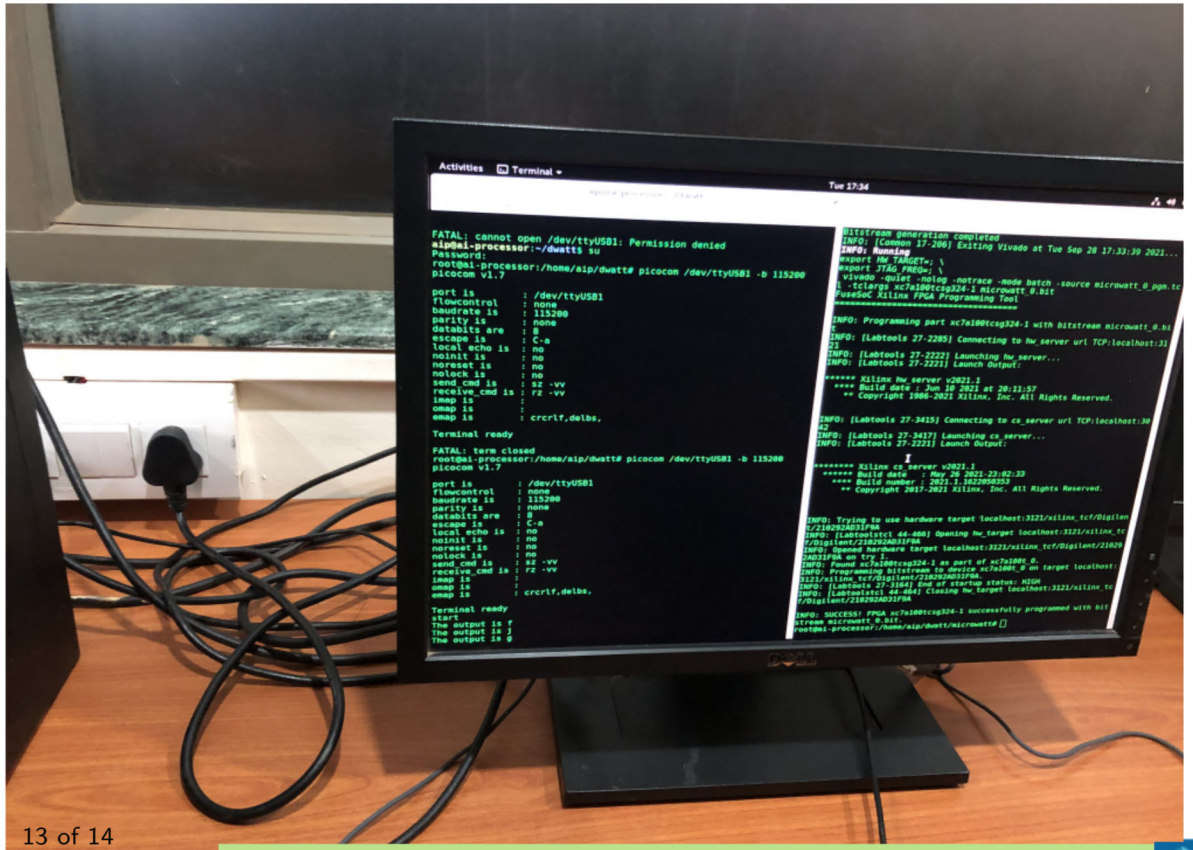


Figure 1: OpenPower Microwatt on a FPGA

OpenPOWER

KLE Technological University
Creating Value
Leveraging Knowledge

Demo- Snapshots-2



OpenPOWER™

OpenCAPI / OMI

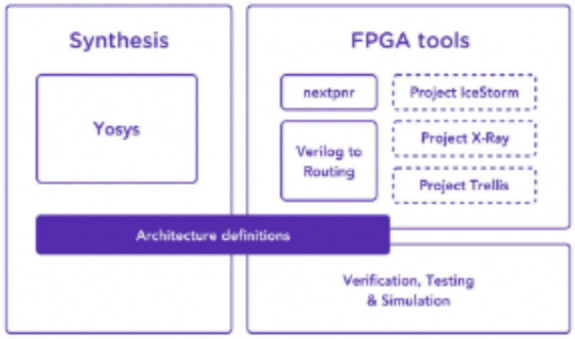
KLE Technological University
Creating Value Leveraging Knowledge

- Open Coherent Accelerator Processor Interface that allows any microprocessor to attach to
  - Coherent user-level accelerators and I/O devices
  - Advanced memories accessible via read/write or user-level DMA semantics
  - Agnostic to processor architecture
- Open Memory Interface
  - maximize memory bandwidth between processors and attached devices

POWER ISA implementations could be used in server, consumer, or embedded applications. An OMI host interface can be used in any ASIC that supports DRAM to reduce the ASIC pin-count by ~4x or increase the bandwidth by 4x

Ref: https://[illegible]

Symbiflow



- SymbiFlow is an end-to-end FPGA synthesis toolchain with the goal to provide a fully open-source, multi-platform, and vendor-neutral design tool option for FPGA developers. It contains all of the necessary tools to convert a Verilog design to a final bitstream.

Ref: https://www.hackster.io/news/gsoc-for-fpga-symbiflow-open-source-toolchain-254c7ab15ff

nMigen

- **a Python toolbox for building complex digital hardware.**
- nMigen itself provides the core language and is complemented by several external components.
- The nmigen-boards package contains definition files for various FPGA boards, providing information such as pin locations and clocks.
- The nmigen-stdio package provides libraries to interface to common I/O protocols such as I2C and SPI.
- The nmigen-soc package is a library that provides bus interconnect and configuration and status register (CSR) management.

Ref: https://m-labs.hk/gateware/nmigen/

Coriolis2

- Coriolis is a set of tools for the VLSI backend.
- The complete design flow has more advanced Foss alternatives:-
  - Yosys for logical synthesis.
  - GHDL for VHDL simulation.

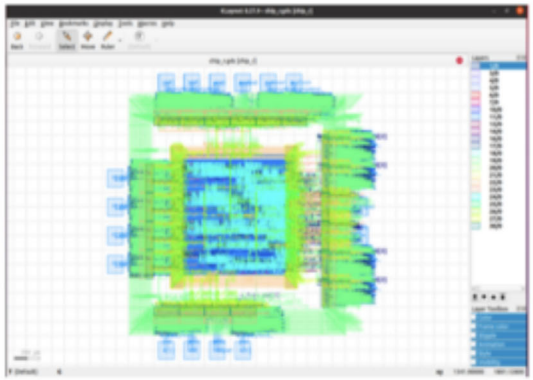


Figure 2: Layout done using Coriolis2

References

- Figures from slides of Wawrzynek and Weaver for Computer Science 61C Spring 2018 course at Berkely EECS
- https://github.com/antonblanchard/microwatt