3rd Workshop on RISC-V and OpenPOWER in HPC

# OpenMP and RISC-V for Green SuperComputers

Lower the Power

**Bernard Goossens**

Université de Perpignan Via Domitia, DALI-LIRMM



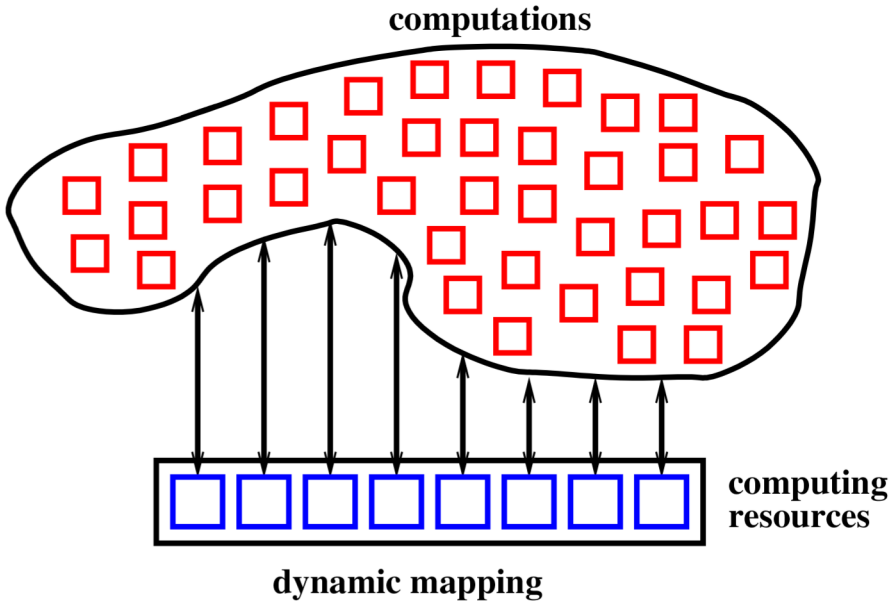
# Outline

# OpenMP and RISC-V: Low-Power Friendly ?

- **OpenMP** fork/join parallel computing model: favorable to **energy saving**.
- **RISC-V** instruction set **modularity**: **energy saving** opportunities.

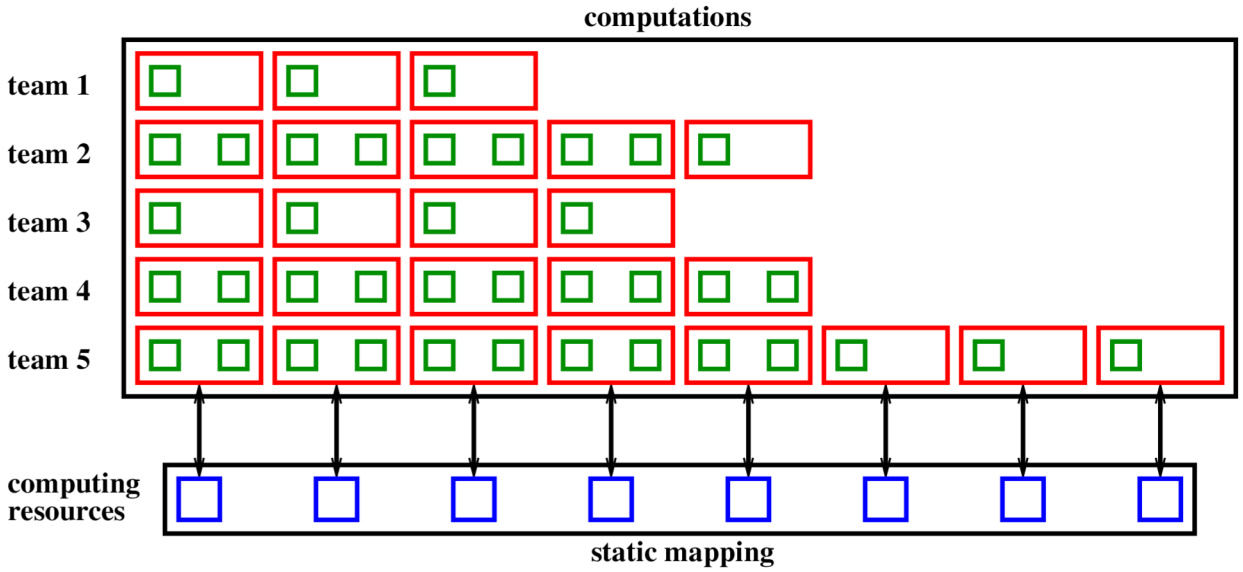
# Outline

# Flexibility is the Enemy of Sobriety




- Computations move in and out of computing resources which must keep **always active**.

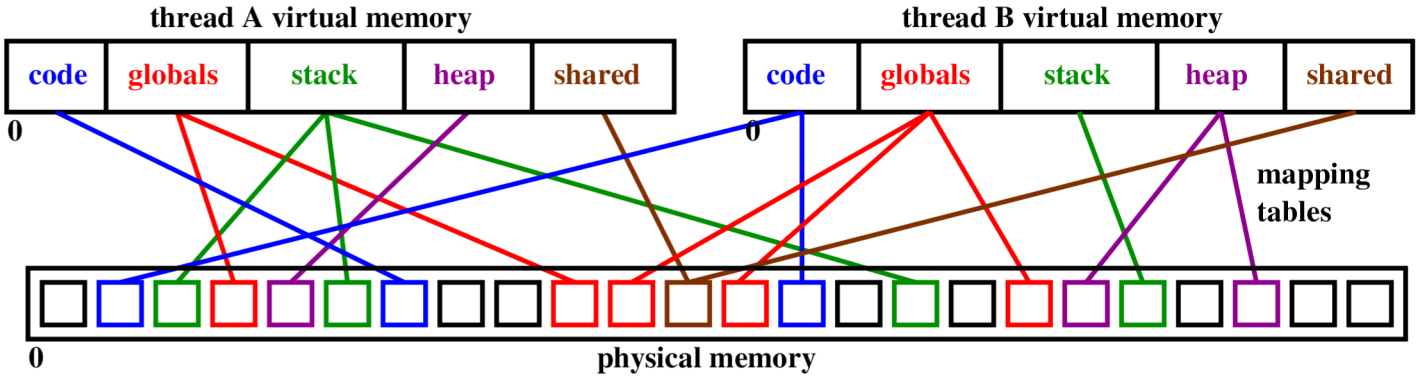
# Structuration is the Price to Pay for Low-Power




computations are grouped (team)
each team member has $\lceil$comp in team/core$\rceil$ computations
a team member is run by its rank matching core

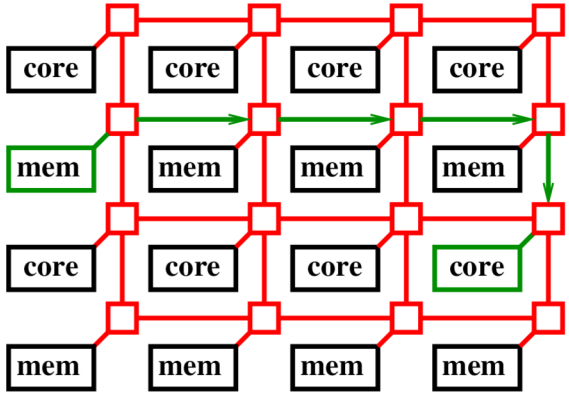
- Matches OpenMP **fork/join** model.
- Cores are powered when used, **unpowered when idle**.

# Memory Paging is the Second Enemy of Sobriety



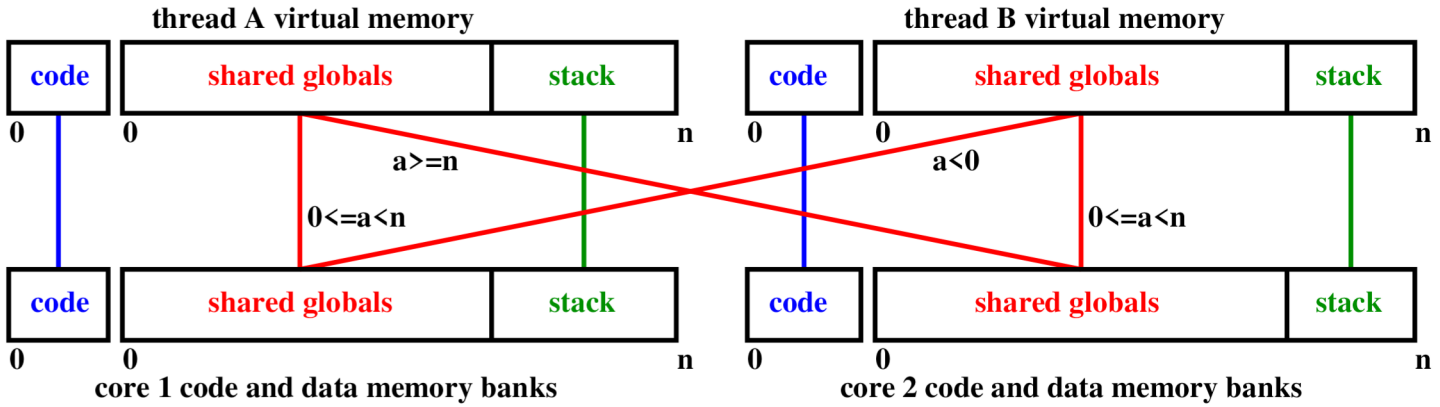

- Dynamic allocation of physical memory
- Physical memory is not related to the accessing thread.

# A NoC to Link Cores to Memory Banks

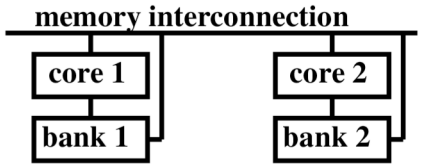



- For every access request, the requesting core is distant from the accessed bank.
- A NoC is necessary to route the requests.
- Caches are necessary to keep latency low.
- The whole NoC and memories must be powered at all time.

# Align Thread, Core, Memory Bank



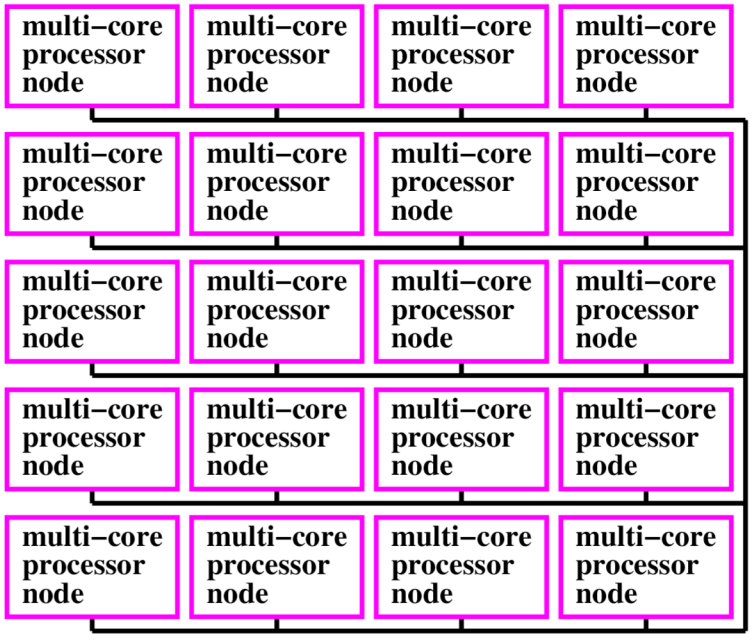

Separate code from data (to align data on 0)
No heap (allocation from thread creation)
Shared = local when 0<=a<n (relative addressing)




- Memory banks are powered by the creation of shared globals, unpowered when they are freed.
- Accesses are either local or to neighbours (OpenMP fork/join): flexible interconnection.

# Outline

# Uniformity is the Second Enemy of Sobriety




32 bit int application

64 bit int application

simple precision FP application

double precision FP application

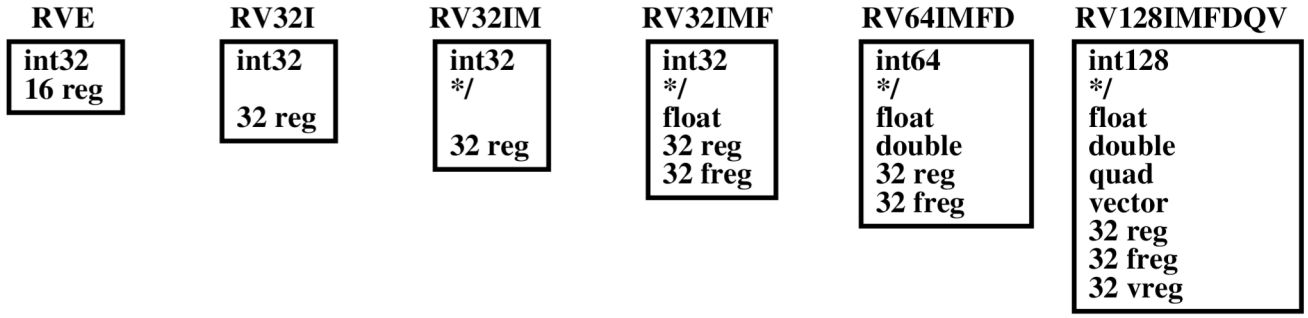
scalar application

vectorized int application

vectorized FP application

- Integer and floating point applications.
- 32 bits, 64 bits, 128 bits computations.
- Scalar, vector processing.

# RISCV: A Modular ISA



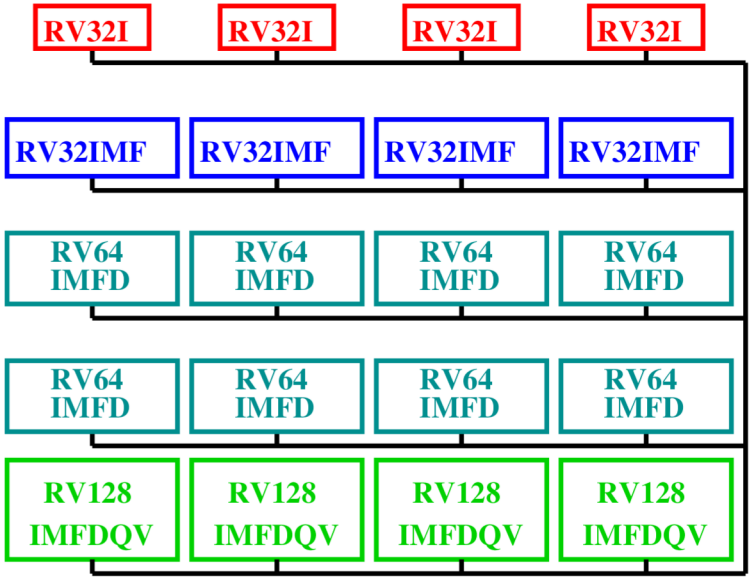
# RISC-V: An Extensible ISA

| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 00 | LOAD | LOAD–FP | custom–0 | MISC–MEM | OP–IMM | AUIPC | OP–IMM–32 | 48b instruct |
| 01 | STORE | STORE–FP | custom–1 | AMO | OP | LUI | OP–32 | 64b instruct |
| 10 | MADD | MSUB | NMSUB | NMADD | OP–FP | reserved | custom–2/rv128 | 48b instruct |
| 11 | BRANCH | JALR | reserved | JAL | SYSTEM | reserved | custom–3/rv128 | >=80b |

**RV32E, RV32I, RV64I, RV128I (and standard extensions) ISA**
**opcode = bits 65 432**
**bits 10: 11 (32b instructions), other value (16b instructions)**
**bits 432 = 111: big length instructions (>32b)**
**reserved = may be used for future extensions (e.g. 10 101 is OP–V)**
**custom/rv128 = custom for rv32 and rv64, OP–64 and OP–IMM–64 for rv128**

- Personal experience on an OpenMP fork/join ISA (X_PAR ISA extension on the **custom-0** opcode space).

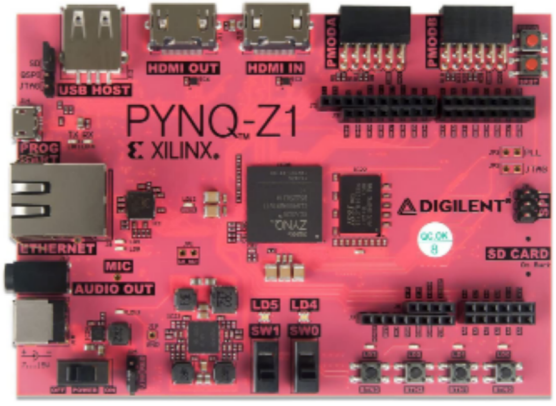
# An Heterogeneous Design with a Modular ISA




- Mix RV32I, RV32IM, RV32IMF.
- Mix RV32IMF, RV64IMFD, RV128IMFDQ.
- Mix RV128IMFDQ and RV128IMFDQV.

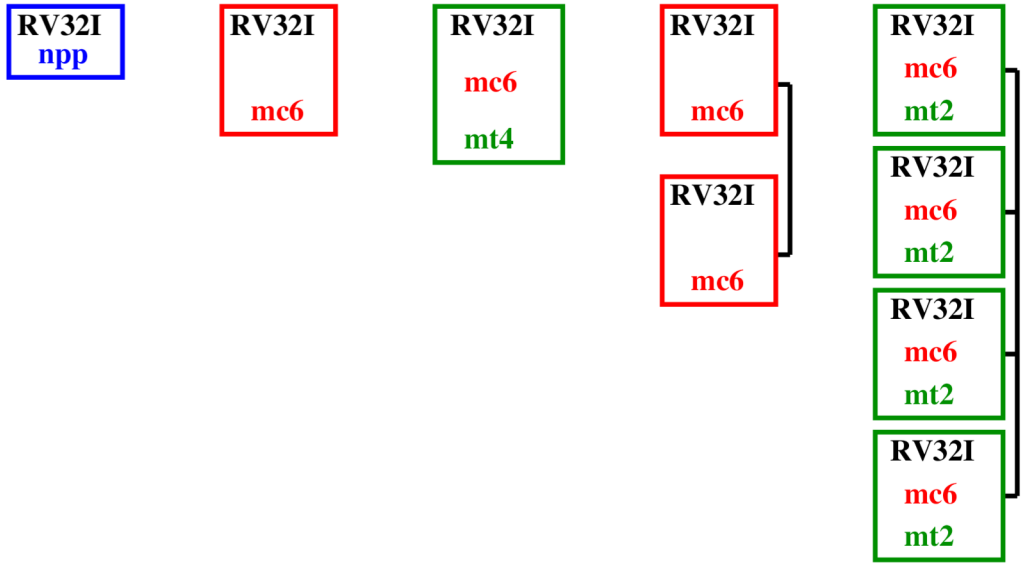
# Outline

# FPGA Implementation on RISC-V Designs

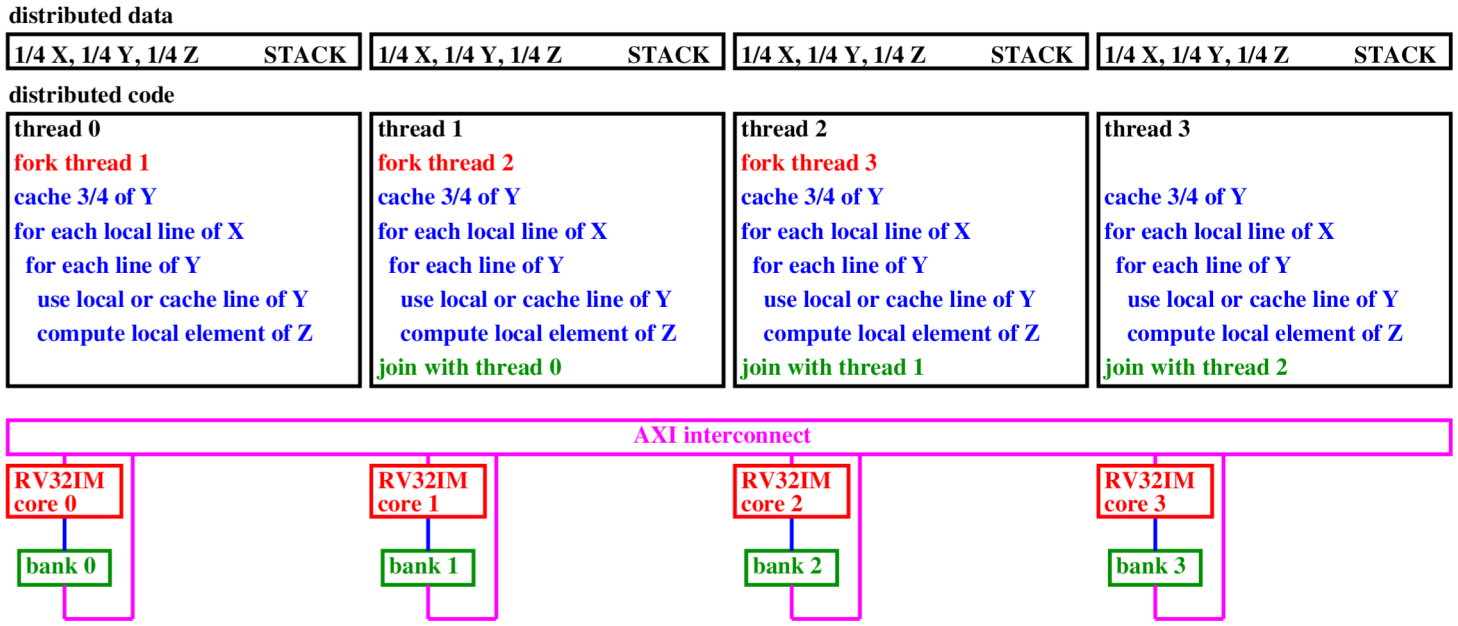

XILINX VITIS

# Multiple RV32I Designs




- RV32I non pipelined: one week.
- RV32I multicycle pipeline, multihart, multicore: one month.

# Parallelized Distributed Matrix Multiplication




- Matrices interleaved and distributed in the memory banks.
- Code forks, runs partial multiplications in parallel and joins.

# Outline

# RISC-V and OpenMP: Two Opportunities for Green SuperComputers

- RISC-V ISA is modular and extensible.
- OpenMP fork/join model enables powering on/off.