FLINT
Flexible Latency-InseNsiTive SPARC

http://ramp.eecs.berkeley.edu

Alex Williams (awilliams@berkeley.edu)
Chris Fletcher (http://cwfletcher.net)
Ilia Lebedev (ilial@berkeley.edu)

Deliverable

- SPARC V8 processor implementation
  - Support for most instructions
  - Support for traps
  - Compact design with higher CPI
  - Tolerance for variable latency of major components
    - Memory, ALU, Register File, etc
  - “One button click to board”

Statistics & Resources

- FLINT (without test harness)
  - SLICE LUTs: 5,254
    - As Logic: 5,252
  - Occupied SLICEs: 2,021
  - Block RAM: 1
  - Clock: ~95 MHz

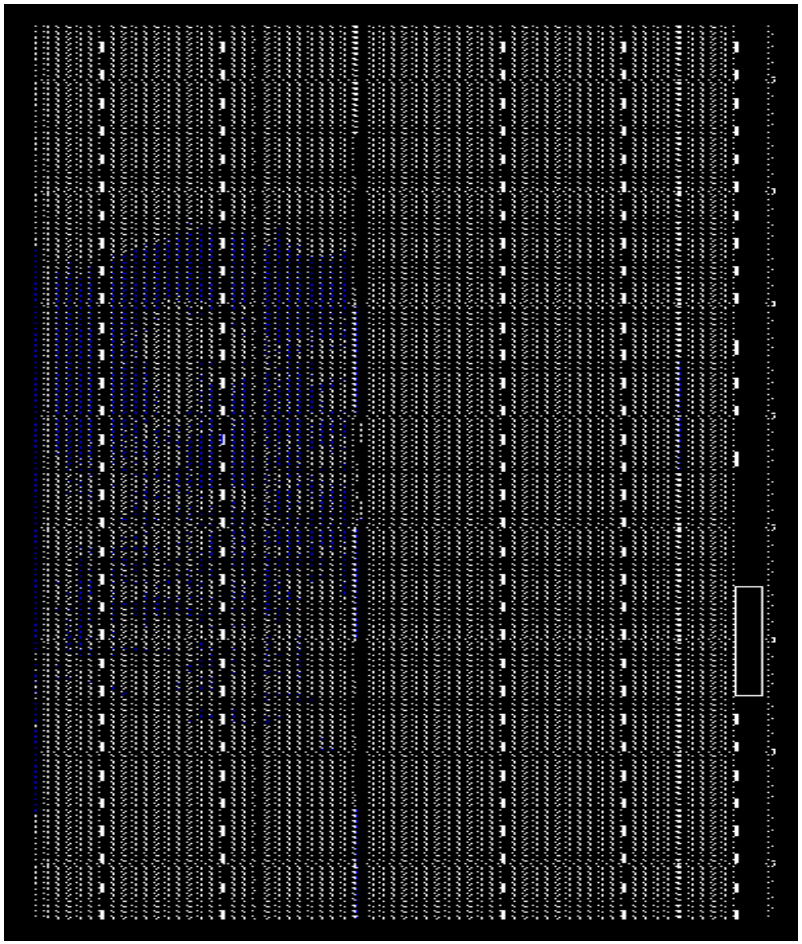
Statistics & Resources (2)

- Test harness
  - SLICE LUTs: 1,911
    - As Logic: 1,317
    - As Memory: 594
  - Occupied SLICEs: 554

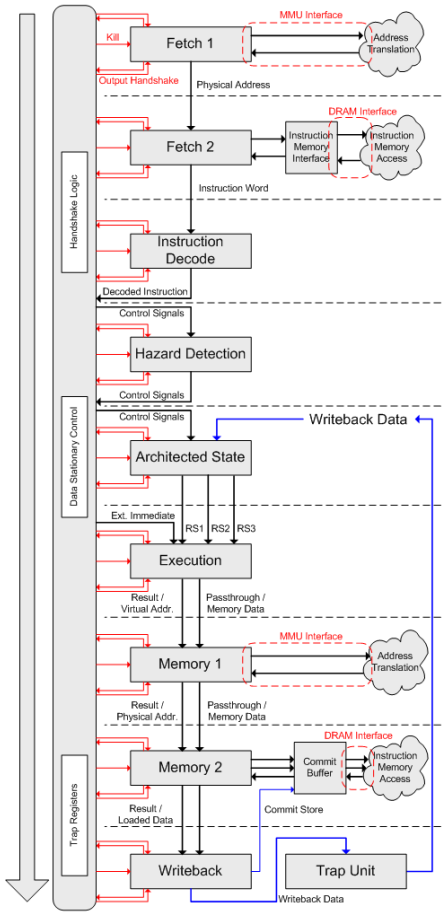
Statistics & Resources (2)

- Placed, Routed design in FPGA Editor
- S for

"Stanford Stinks"



Block Diagram



Physical Address → MMU → Virtual Address

Virtual Address → DRAM Backend → Instruction

Instruction → Control Signals

Stalls on Data Hazards

Register Read/Write: Status Registers and Register Windows

ALU: Add/Subtract, Shifts, Multiply, Logical Operations, Selects Operators

Pass-through for normal instructions,
Physical Address → MMU → Virtual Address for memory instructions

Pass-through for normal instructions
Buffers data for stores
Fetches data for loads

Pass-through for normal instructions
Commits data buffered in Memory 2 for stores
All traps are detected and handled

Demo

- Tests passing in software
- Tests passing in hardware
- User demonstration:
  - Calculating X^(Y+Z), Fib, and some other stuff.