The Case for Soft-CPUs in Accelerator Control Systems

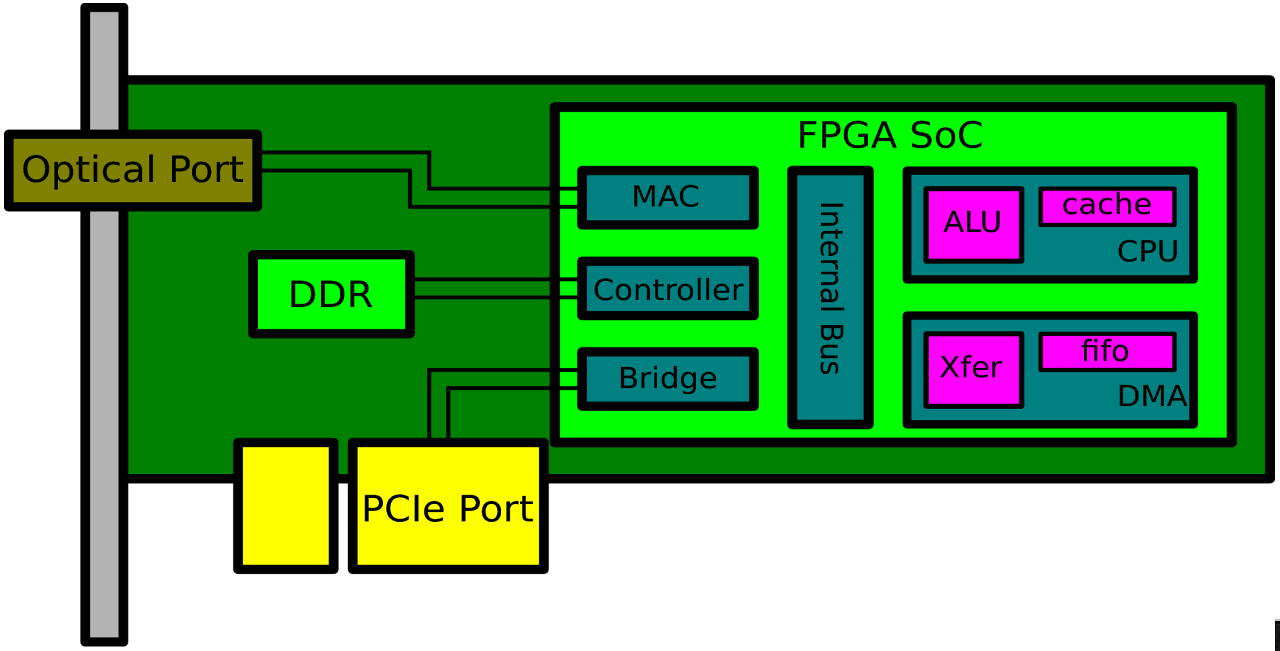
Wesley W. Terpstra

GSI Helmholtz Centre for Heavy Ion Research GmbH

Darmstadt, Germany *http://bel.gsi.de*

What is a Soft-CPU?

- A full CPU
- Implemented in HDL
- Connected to the FPGA internalSoC bus



Why use a Soft-CPU? (vs. custom HDL)

✔ Re-use large body of C/fortran/etc code

✔ Complex execution order

✔ Dynamic resource management

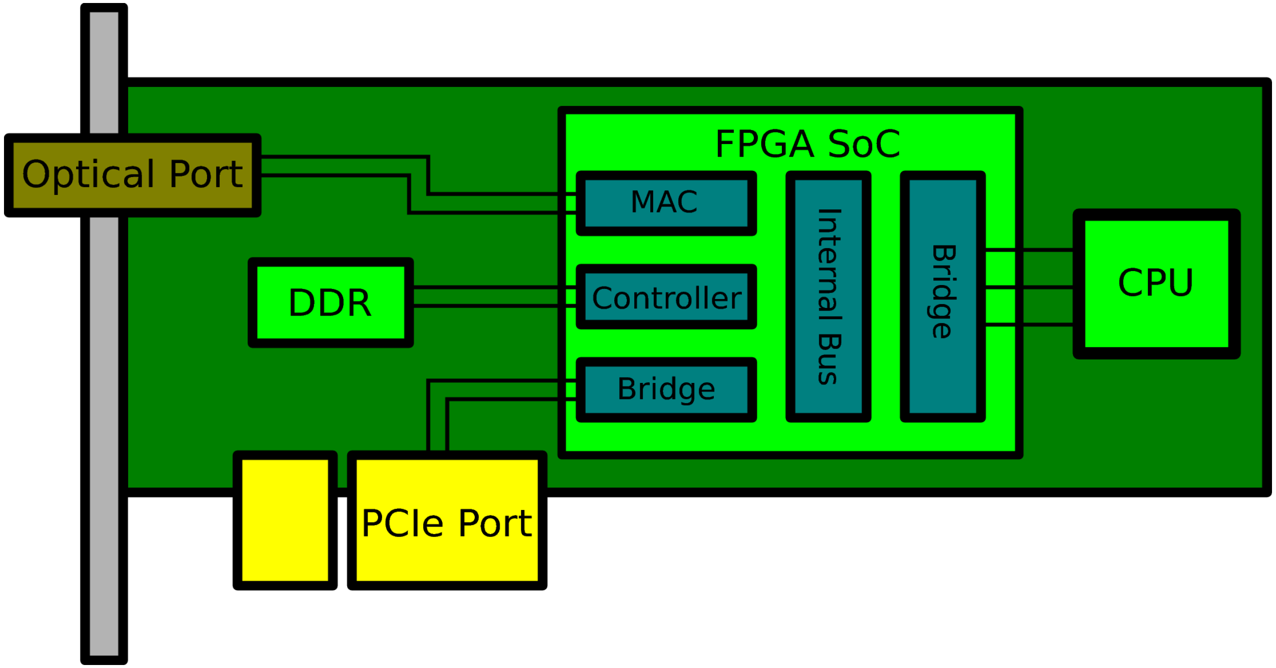
✔ A single component to solve many tasks

✔ Easier to debug and trace

✗ Slower

✗ Requires a memory subsystem

Why use a Soft-CPU? (vs. external CPU)



- Physically separate chip
- ARM/etc integrated into FPGA

Why use a Soft-CPU? (vs. External CPU)

✗ Speed

✗ Standard OS and toolchain

✔ Customizable instructions

✔ One less part

✔ **Runs synchronously with FPGA bus clock**

- Deterministic timing behaviour
- Tight integration with custom HDL

Size? ... Memory!

- (Good) Soft-CPUs take ~2% of EP2AGX125
- A single Soft-CPU can run multiple programs

What's the cost of a Soft-CPU?

The Memory Subsytem

Trades FPGA gates for memory blocks

The more it does, the more it needs

The true cost of more memory

- More memory = further away from CPU
- Further away = more complicated timing

  (bus access cycles, prefetch, cache miss)
- MMU makes it even worse (TLB misses)

→ *Might as well use an external CPU*

Soft-CPU non-issues

- Most (good) Soft-CPUs have similar perfomance
  - Single cycle issue and ~175MHz (Arria2)
  - 3-staged fetch/decode/execute

- Comparable area (at most 3* different)
- Similar executable sizes (32-bit RISC)

Soft-CPUs, but Hard Requirements

- Open Source
  - Portable (Altera, Xilinx, Lattice, ... future proof)
  - Tweakable (custom instructons, bus choice, ...)
- Well documented, tested, and supported
  - Eliminates almost all open source softcores
- Survivors: LM32, LEON3, OpenRISC, (ZPU)

Feature Comparison

Requirement	Purpose	Issues
Gcc-toolchain	re-use of C	-
JTAG access	debug bus, load F/W	ZPU
Debug support	trace program	ZPU
Flex. mem bus	max determinism	OR1k
Documentation		ZPU

Best choices: LM32 and LEON3

Size/speed trade-offs (Ballpark)

CPU	Mhz	Size (LEs)	Cycles/mul
ZPU	~~300~~ 175	400	576+
LM32	~~250~~ 175	900	1
LEON3	175	2500	1
OpenRISC	150	3700	1

(All numbers are for an Altera Arria2)

... and the winner is!

If you need an MMU / linux

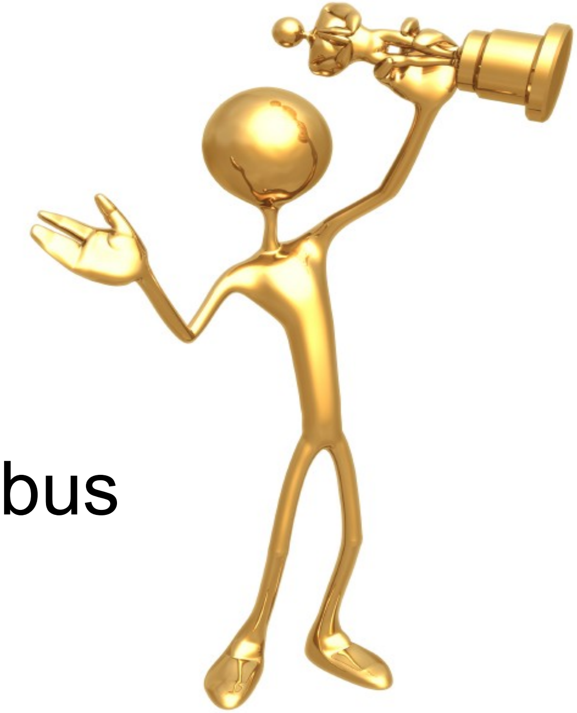
LEON3 **(or an external CPU)**

If you need deterministic execution

If you need a configurable memory bus

If you need small and fast

LM32 !



Questions?

LM32 Processor Overview

- 32-bit RISC architecture
  - 32 registers
  - 8-16 control registers
  - no side-effects / flags
- 6-stage pipeline
  - single cycle issue
  - 1-3 cycle result
- Harvard architecture

What's bad about NIOS/Microblaze?

- Not Open-Source
  - Vendor-specific (Altera/Xilinx)
  - Cannot be debugged / signal trapped
  - Cannot be tweaked or fixed

- On the other hand:
  - more comprehensive feature set
  - more complete instruction set
  - vendor specific HDL = slightly smaller

What's bad about ARM/MIPS?

- Patented instruction set
  - Even if you make your own Soft-CPU... bad!

- (If not Open Source):
  - Cannot be debugged / signal trapped
  - Cannot be tweaked or fixed

- On the other hand:
  - more comprehensive feature set
  - more complete instruction set

LM32 Processor Options

Feature	Size	Impact	Suggestion
• Instruction cache	++	++++	KEEP
• Data cache	+++	+	-
• Multiplier	.	+	KEEP
• Barrel shifter	.	+	KEEP
• Divider	+++	+	-
• JTAG access	++		DEBUG
• Watch/breakpoints	+		DEBUG
• 32 Interrupts	+	++	KEEP