THE UNIVERSITY of EDINBURGH
informatics

icsa

EnCore: A Low-power Extensible Embedded Processor

Nigel Topham

The University of Edinburgh

http://groups.inf.ed.ac.uk/pasta/

EPSRC Engineering and Physical Sciences Research Council

HiPEAC COMPILATION ARCHITECTURE

Overview

- EnCore is an embedded soft-core processor developed in the PASTA project – so first an introduction to the project…
- Aims, goals and key results from the PASTA project
- Architecture of the EnCore family
- Implementation results
- Multi-core versions of EnCore
- Summary

Aims and Goals of the PASTA Project

"Rapid, automated creation of high-performance, energy-efficient, application-specific processors"

- The PASTA project began in September 2006, funded by a research grant from EPSRC. Since then the group has grown to team of 10 researchers, addressing a range of research topics in architecture and compiler synthesis

- Primary innovations are:
  - Automated processor customization
  - Automated exploration of many design options
  - Applying statistical methods to allow tools to 'learn' how to optimize
  - Co-design of both the processor and its compiler

- Many tradeoffs involve CPU implementation, so we created EnCore

Why a New Approach to Processor Design?

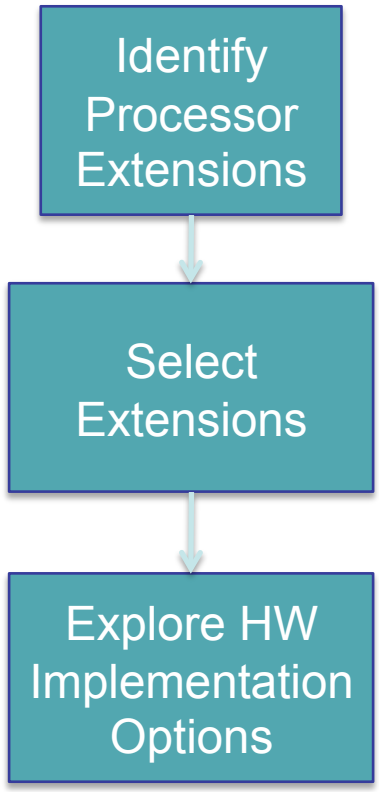
- Moore's Law increases transistor counts at an exponential rate
- Transistor count doubles at each new technology node (130nm, 90nm, 65nn, 45nm, 32nm, etc)
- Chips with more than 1 billion transistors feasible today
- Set to continue until ~2016

- Using the additional resources effectively is a big design problem:
  - How to get best performance from a given silicon area?
  - How to increase performance while decreasing energy usage?

- Basically too many design options, leading to complex trade-offs
- In the past, human designers could handle the complexity
- Now, we need smart design tools to navigate the complexity
  - Low-level choices have high-level impact – must be predicted
  - Prediction requires tools that learn about the design space

Learning how to Optimize Processors

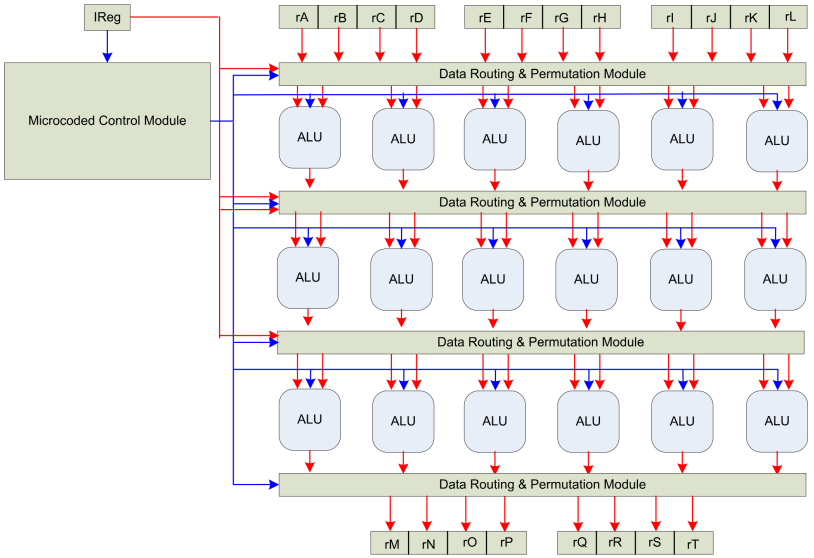
- Instruction Set extensions have been around for some time, so what's different here?
- Extendable architectures create a huge space of design choices in architecture and implementation
- Most prior methods involve heuristics which yield a single solution; this is not what we want…
- Our goals are:
  - To expose the design spaces through parametric algorithms
  - To develop statistical methods for learning how to quickly find good solutions from within those design spaces
- Tools become "expert designers"



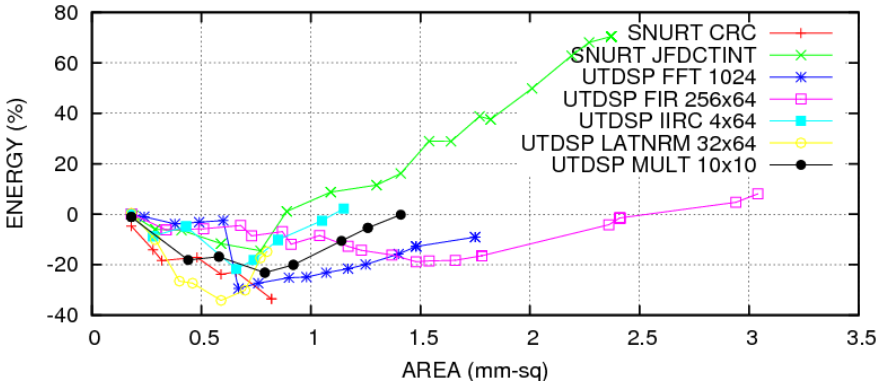
Research Activities in the PASTA Project

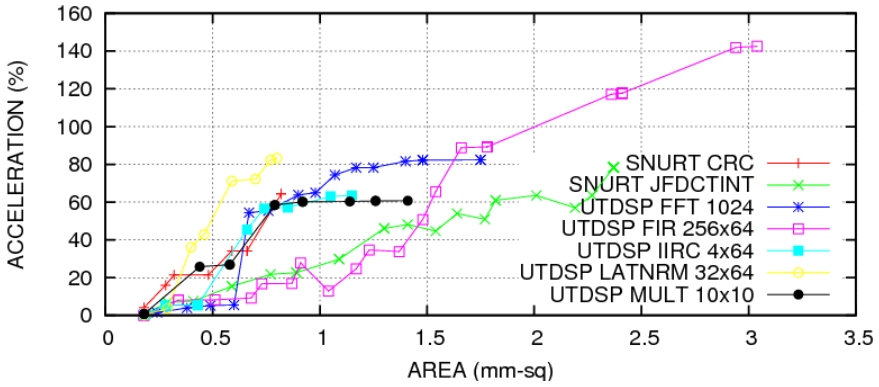
- Application-specific processor synthesis
- Exploring the trade-off space in microarchitecture
- Compiling for customized processors
- High-speed processor simulation
- HW-SW co-design for energy efficiency
- Self-adaptive microarchitecture for energy efficiency
- Proof-of-concept microprocessor implementation (EnCore)
- … will summarize results from a few of these

Application-specific Processor Synthesis

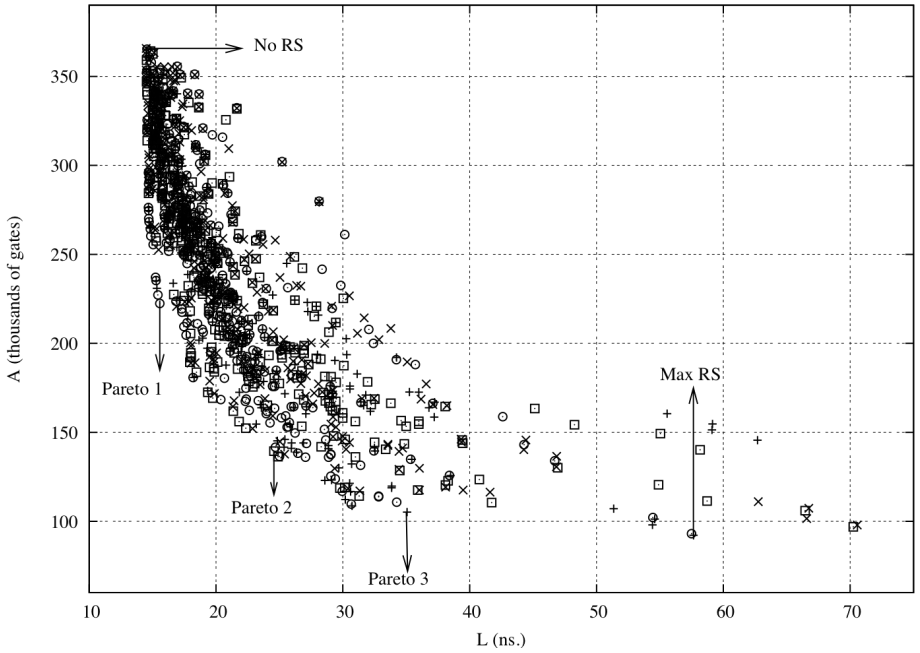


- Extension instructions map to a Configurable Function Accelerator
- Design space of feasible CFAs is searched automatically, under constraints of die area
- Resulting solutions are examined to see how additional area leads to:
  - Performance improvements
  - Energy reductions





Exploring the Trade-off Space in Microarchitecture

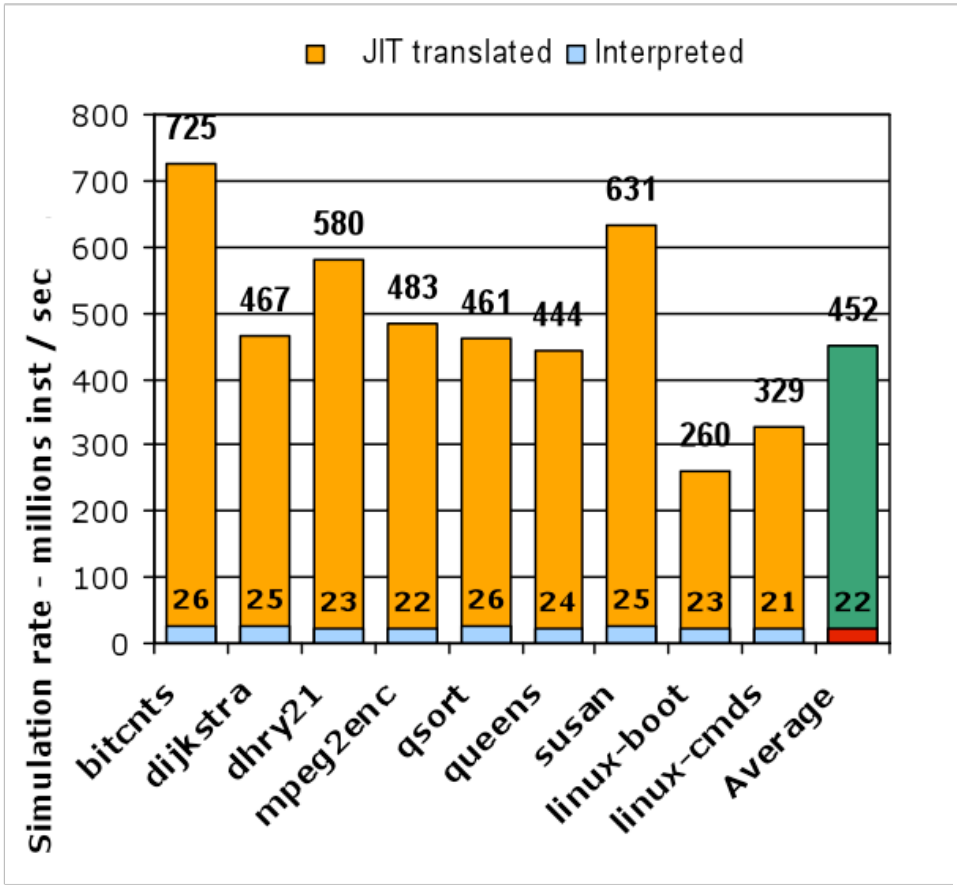


Graph showing trade-off between speed and die area when sharing resources across extension instructions

http://groups.inf.ed.ac.uk/pasta/rareas_rshare.html

- 100's of extra instructions may be added to the processor
- Similar functional units will appear in many instructions
- Small number of instructions will be active at any instant
- Therefore, try to share the logic between disjoint instructions
- Maximum resource sharing leads to maximum slow-down
- Maximum speed achieved at minimum resource sharing
- Thousands of points in the design space, defining a Pareto curve.

High-speed Processor Simulation



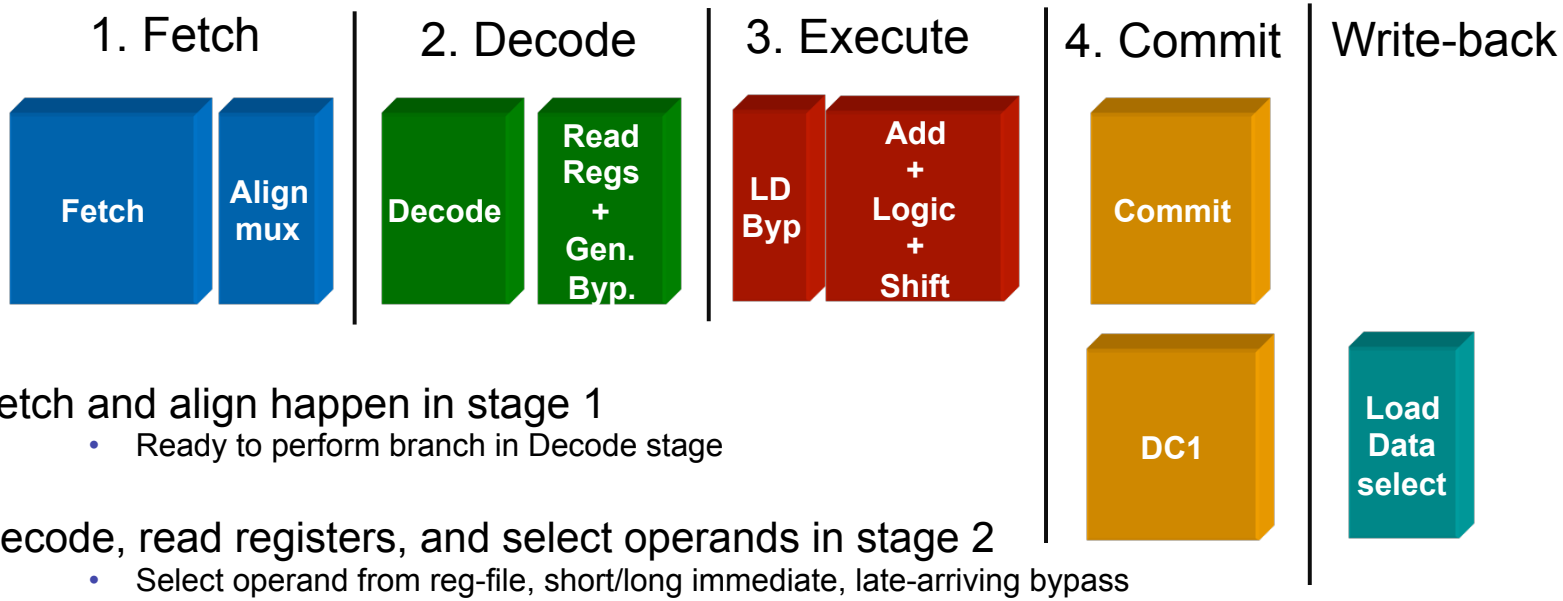
- Training of statistical performance predictors requires high-speed simulation
- Developed fast simulator based on just-in-time binary translation
- Simulates full system at 450 MIPS
- Licensed to ARC in 2007
- Now released as a commercial simulation tool
- New cycle-approximate models
  - Max. error < 2%
  - Only a small slowdown

http://groups.inf.ed.ac.uk/pasta/rareas_fastsim.html

The EnCore Processor Family

- Design status
  - Based on ARCompact ISA for embedded applications
  - Implemented as configurable Verilog IP libraries
  - 5 and 7 stage versions of the microarchitecture (EC5 and EC7)
- Configurability
  - Pipeline depth (I.e. 5 or 7 stages)
  - I-cache and D-cache (capacity, ways, associativity,…)
  - Extended instruction set
  - Various microarchitectural options (instruction packs)
- Tools
  - Fast simulation: dynamic binary translation and/or predictive
  - Co-simulation: Verilog core + fast simulator
  - Extension instruction generation tool
  - Extension microarchitecture synthesis tool
    - Two versions – one static, the other dynamically reconfigurable

Overview of EnCore EC5 pipeline structure



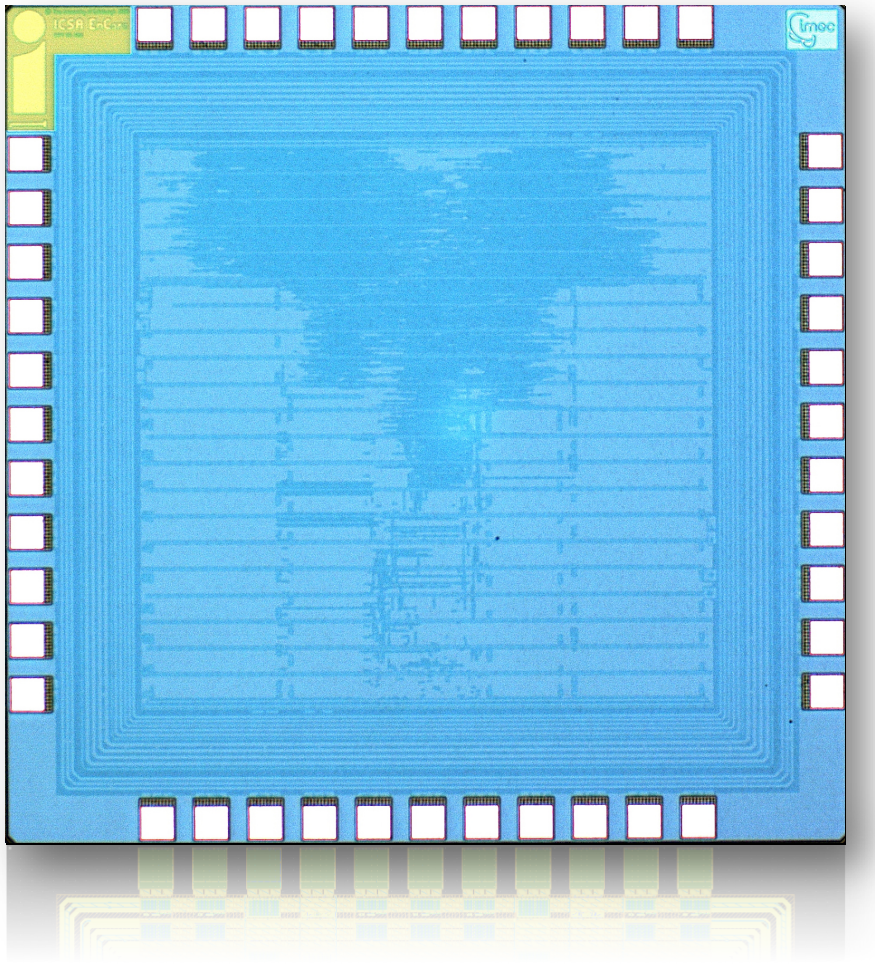
- Fetch and align happen in stage 1
  - Ready to perform branch in Decode stage
- Decode, read registers, and select operands in stage 2
  - Select operand from reg-file, short/long immediate, late-arriving bypass
  - Zero branch cost (with delay-slot or if branch not-taken)
- Execute ALU operations in stage 3
  - Optimized forwarding network, split across stages 2 and 3
- Access data cache or perform min, max, abs operations in stage 4 (commit decision)
  - No stalls between most non-memory instructions
- Write register file and architectural state in write-back stage
  - One load-delay slot

The EnCore “Calton” test chip

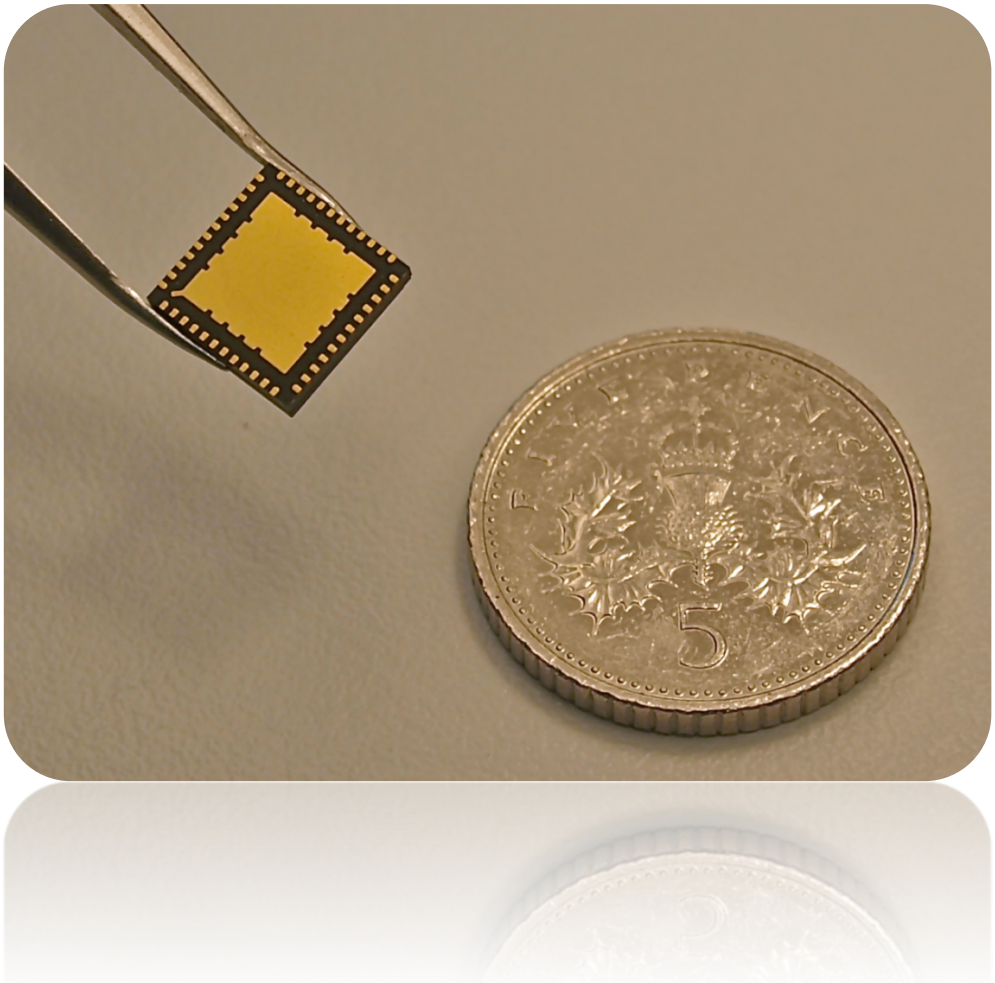
- All chips named after hills in Edinburgh
- Calton Hill is the smallest…

- Baseline CPU core is efficient:
  - 5-stage pipeline, 1.45 DMIPS / MHz
  - Configurable RTL (caches etc.)
  - 24 kgates
  - 250 MHz worst-case, 130G process
  - 70 uW / MHz, 130G process, inc RAM

- `Calton’ test chip taped out Nov. 2008
  - UMCL130 FSG HS
  - 1 sq.mm active area
  - RTL to GDSII (in-house design flow)
  - 375 MHz in 0.13um (slow, free libs)
  - 90 uW/MHz dynamic power (chip level)
  - **100% functional in first silicon!**

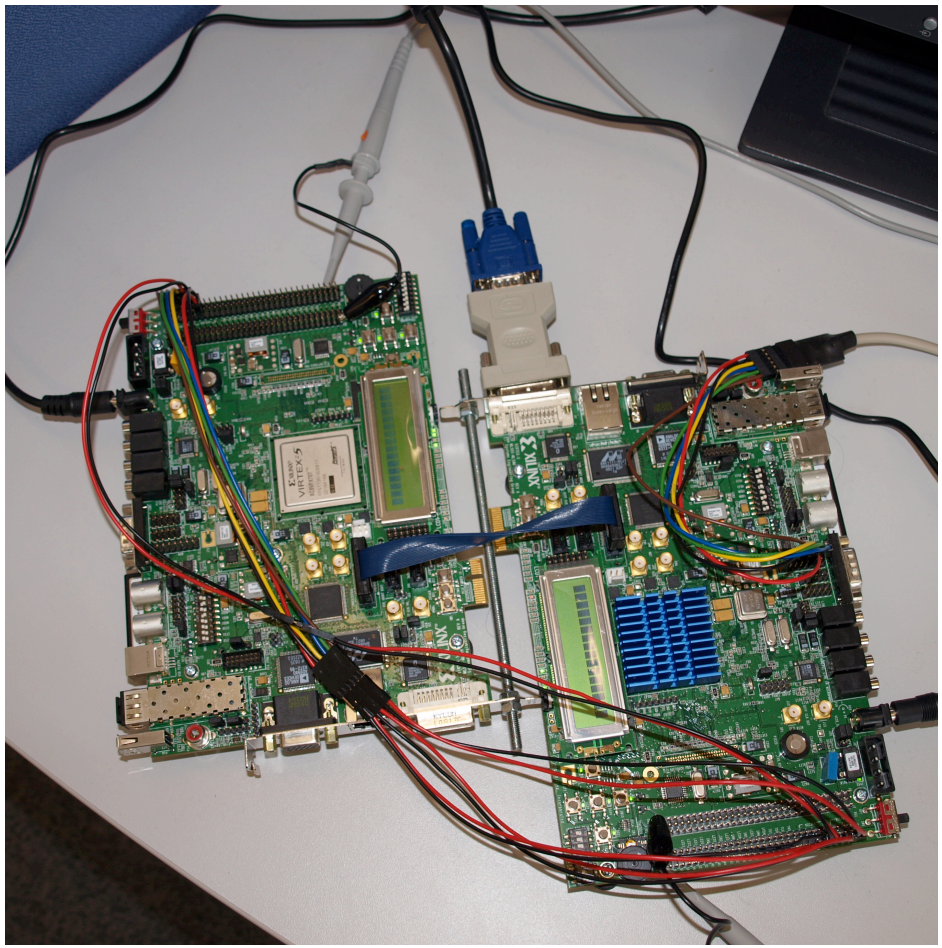


Packaged EnCore `Calton' Test Chip



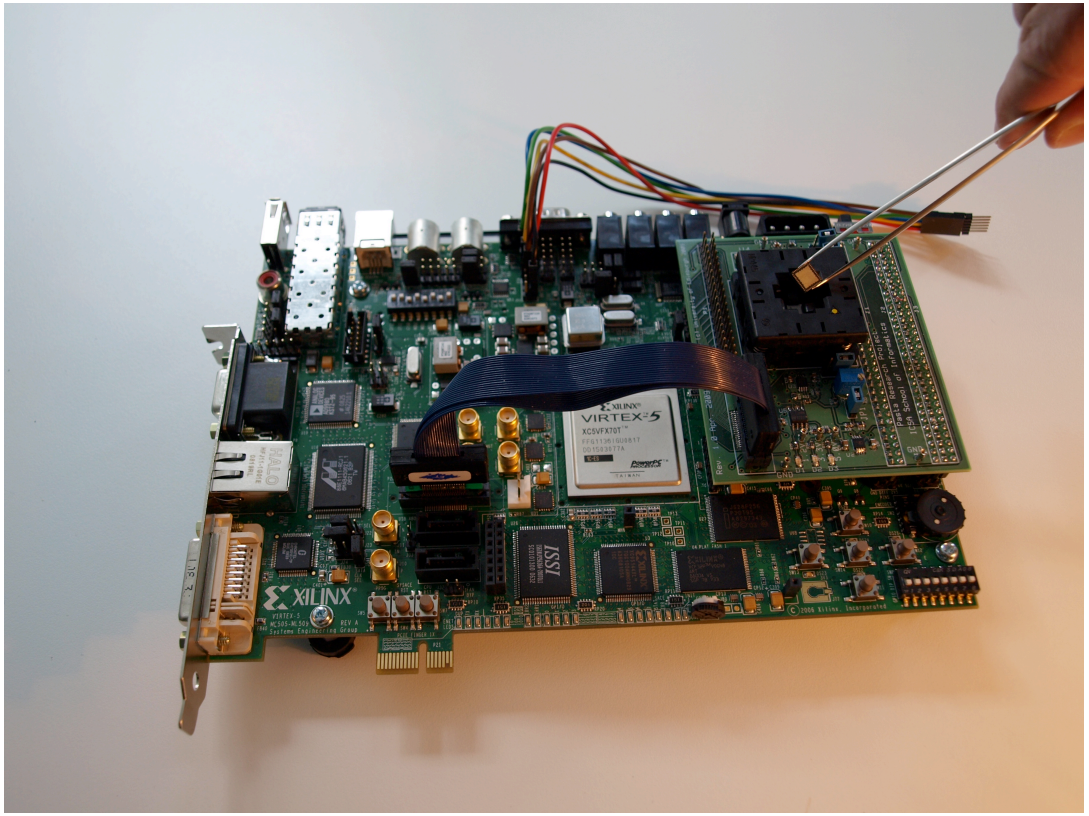
- Chip Scale Packaging
- 7 x 7mm QFN-48
- 5p coin shown for scale
- 100% packaging yield
- Designed & built PCB
- Running code compiled by our compiler
- Fabricated an enhanced version of EC5, with higher performance architecture in May 2009
- Enhanced devices received back in October 2009, these devices were also functional

Hardware Emulation Environment



- Dual FPGA configuration, based on Xilinx ML507 with Xilinx Virtex-5 FPGA
- FPGA-1 runs EnCore SoC
- FPGA-2 runs chipset
- Identical interconnections to those used by test chips post-manufacturing
- Allows functional verification at 75 MHz prior to tape out
- Allows experimentation
- Allows for compiler testing in real-time conditions

Chip-testing environment



- Uses Xilinx ML507 board
- Custom daughter board
- Test socket for MO220
- JTAG link to host
- Xirtex-5 implements the rest of the system
  - Main memory
  - Graphics
  - AC97 audio
  - Etc.
- Re-usable for future devices

Vital Statistics of EnCore

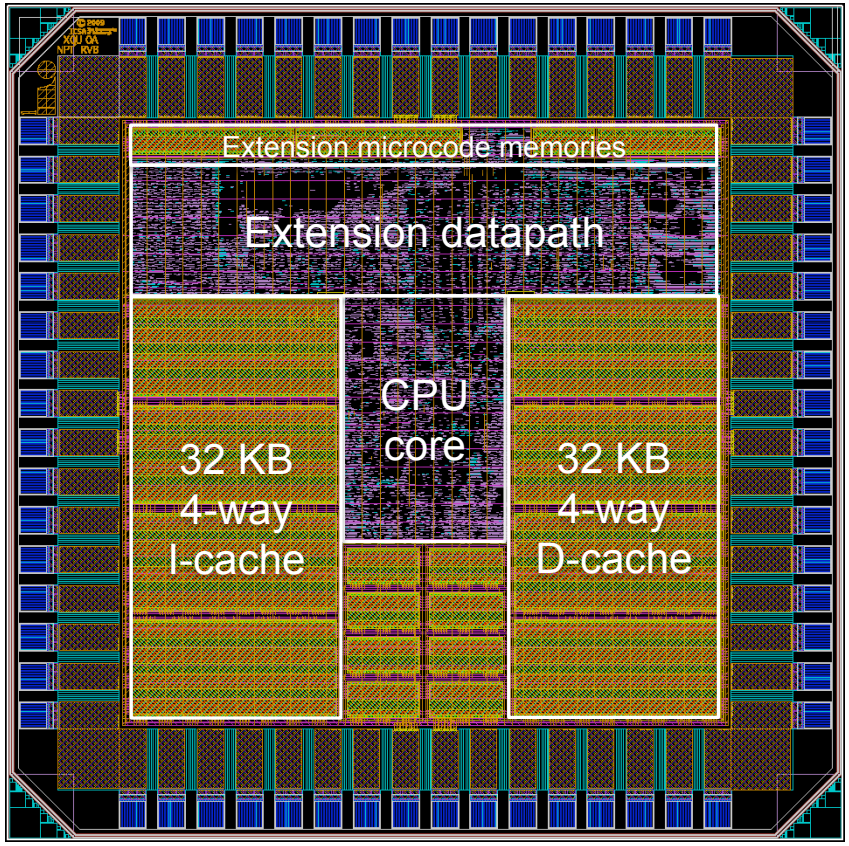
Feature	Silicon proven	EnCore 130nm (speed opt)	ARM Cortex M3 130nm (speed opt)
Core size (baseline)		0.15 mm2	0.43 mm2
DMIPS / MHz		1.45 (single issue)	1.25 (single issue)
Power consumption (250 MHz)		0.070 mW / MHz (Fmax = 250 MHz)	0.140 mW / MHz (Fmax = 135 MHz)

Feature	In fabrication	EnCore 90nm (speed opt)	ARM Cortex M3 90nm (speed opt)
Core size (baseline)		0.071 mm2	0.210 mm2
DMIPS / MHz		1.45 (single issue)	1.25 (single issue)
Power consumption (350 MHz)		0.024 mW / MHz (Fmax = 420 MHz)	0.070 mW / MHz (Fmax = 191 MHz)

Feature	Projected	EnCore 65nm (Low Power)	ARM Cortex M3 65nm
Core size (baseline)		0.037 mm2	N/A
DMIPS / MHz		1.45	1.25 (dual issue)
Power consumption (400 MHz)		0.012 mW / MHz (Fmax = 500 MHz)	N/A

- Core sizes exclude RAM area
- EnCore power measurements include cache RAM power
- ARM data from ARM website (also believed to exclude RAM area)

90nm EnCore chip with Extensions – “Castle”



- EnCore baseline CPU
- 32KB 4-way I-cache and D-cache
- 32-bit external bus interface

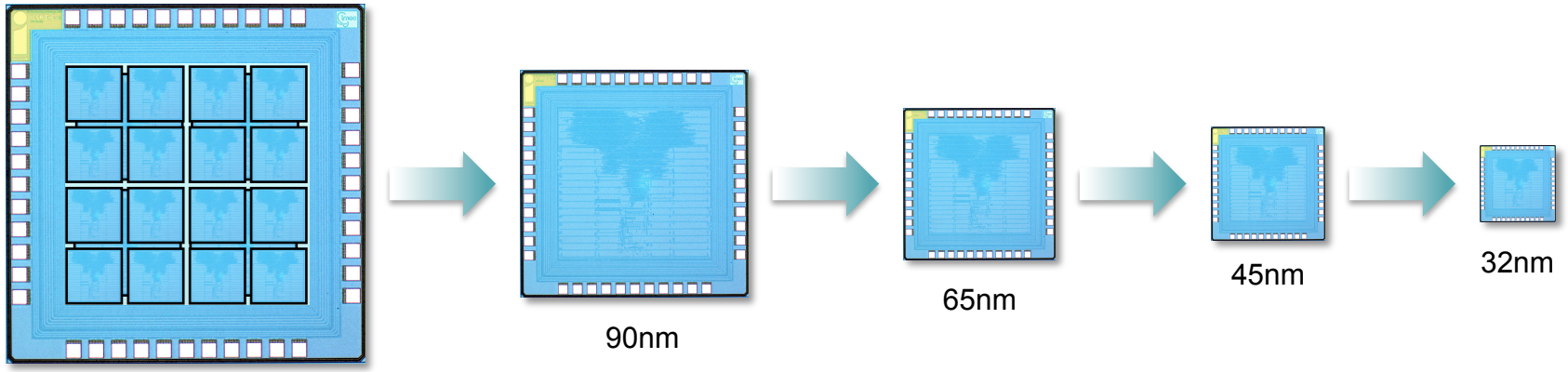
- Includes CFA accelerator
  - Dynamically reconfigurable extensions
  - Optimized for audio decode

- UMC 90nm process, 8LM
- Faraday free libraries
- 580 MHz (sampled from typical Si)
- 350 MHz (worst-case PVT)
- 56 uW / MHz (extended core, inc. RAM)
- 1.875 x 1.875 mm die

- Submitted for fabrication October 2009

Projections to 65nm and Beyond

- At 65nm a quad-core EnCore system fits into same 1 sq.mm of current 130nm single-core
- 16-cores are possible in 1 sq.mm at 32nm
- Energy efficiency grows with each new technology node, for tiny processors such as Encore
  - Localized signalling = short wires = low switching capacitance
  - This is where parallelism achieves a big win
  - Energy-efficient throughput computing through many-core devices
- Virtual memory in development - able to run Linux on next 65nm Quad Core chip

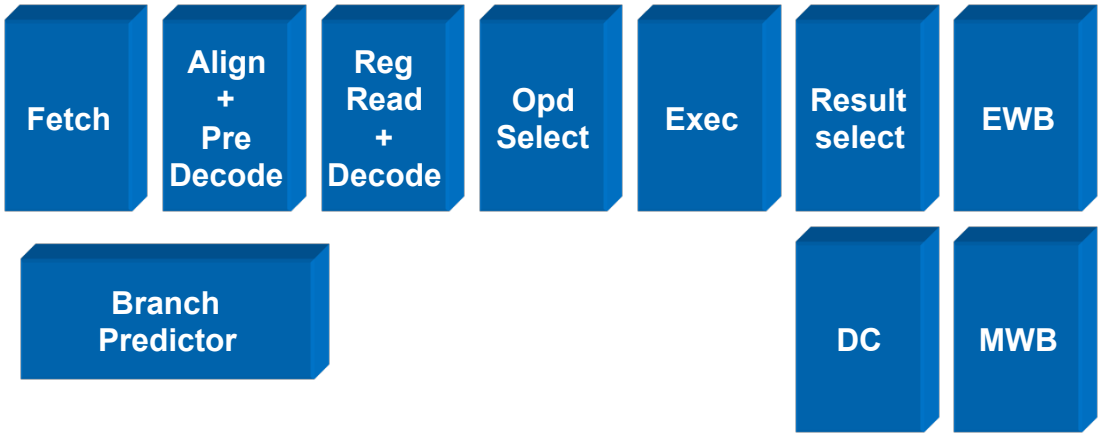


Future Research / Application Potential

- Green Computing
  - Low energy ubiquitous devices
  - Extensions ease load on CPU, reducing energy consumption
- Security, Encryption, Smartcards, RFID devices
  - Low energy ideal for embedded, mobile, smartcard / active RFID applications
  - Customization provides barrier to hackers
  - Extensions allow fast, efficient bit-level algorithms
- Medical
  - Implantable devices enabled by low energy use and relatively high performance achievable through customization
- Many-core Throughput Computing – e.g. web server farms
  - Customize the processor for search algorithms
  - Put 128 cores on a chip, run at 600 MHz, consume 2 Watts
  - 100,000 DMIPS
  - DMIPS comparable with 16 Intel Core2 processors running at 3 GHz

EnCore 7-stage pipeline overview

- Pre-dates EnCore 5-stage pipeline
- Implements same set of instructions as EnCore 5
- Shares many modules in common with EnCore 5
- Decode is pipelined across two stages



- Achieves higher frequency - 550 MHz (worst case) at 90 G
- 30% more silicon area than EC5

Summary

- PASTA project has demonstrated key technology advances in:
  - Energy saving through customization
  - Resource sharing in processor extensions
  - Exploring the vast design space of processor customization
- It is feasible to build real processors in academia !
- Developed a highly efficient microprocessor
  - Proven in silicon
  - Extensible
  - Baseline EnCore is competitive
- Developed matching compiler technology
- 90nm test chip now in the fab.
- Excellent core for energy-efficient multi-core systems
- Thanks to the whole PASTA team !