CSE 291-E00 and ECE 260C

Spring Quarter: Application Specific Processors

Class Time: Tue, Thu, 2:20-3:40pm, Center 202

Instructors:
Brad Calder: calder@cs.ucsd.edu
Bill Lin: billlin@ece.ucsd.edu
Tim Sherwood: sherwood@cs.ucsd.edu

Office Hrs: By appointment

URL: http://www-cse.ucsd.edu/classes/sp02/cse291_E/index.html
Why Application Specific Processors?

• At 0.13μm, Pentium 4 die size can fit about 50 ARM9 cores, 80 at 0.10μm

• At 0.13μm at 250MHz clock, ARM9 dissipates about 0.1W. 50x = 5W

• 150M transistors single-chip at 0.13μm, 250M at 0.10μm

\[ \text{Pentium4 at 0.13mm} = 144 \text{ mm}^2 \]
\[ \text{ARM9 at 0.13mm} = 3 \text{ mm}^2 \]
Why Application Specific Processors?

• System-on-Chip = Boards-on-Chip?
• Instead of Pentium + PCI + PCI cards + DRAM, System-on-Chip = CPU cores (e.g. ARM) + on-chip networks (e.g. AMBA bus) + ASIC blocks + SRAM
• Many issues are similar: e.g. need standardization process for components supplied by 3rd party to interoperate
• What’s different? Components come as “Intellectual Property” (i.e. “Designs”) rather than IC’s
• So why not make the components customizable? Anyway need to fab specific instantiations
Why Application Specific Processors?

- The promise: Customization enables better performance and/or better fit
- Example of customization
  - Add additional “data paths”
  - Add application-specific “data paths” and/or application-specific instructions
  - Vary processor parameters (e.g. #registers, cache sizes)
  - Add application-specific accelerators (as “peripherals”)

Example: Tensilica’s XTenسا Processor

- Basic architecture
- Comparable performance of 32-bit RISC
- 0.7mm\(^2\) in 0.18µm
- 0.4mw/MHz
Adding Additional “Data Paths”

- Can exploit instruction-level parallelism
- e.g. two independent adds can happen in same instruction cycle
Adding Specialized “Data Paths”

- Multiply-accumulate standard computation for DSP applications
Adding Customized “Data Paths”

- TCP processing is “state machine” like
- Typically consumes many instruction cycles
Vary Processor “Parameters”

• Vary #registers
• Vary instruction or data cache sizes
• Possibly add speculative execution logic?
Add “Accelerators” as “Peripherals”

- Specialized “accelerators” can communicate with processor via standardized bus protocol (e.g. ARM’s AMBA bus)
Example Applications

• Because we have been able to embed 10-50 processors and components on to a single-chip, many more to come, almost all IC’s designed today can be regarded as an “embedded processor” design, with each embedded processor having the potential of being “application-specific”

• Typical cited examples
  – Cell phones, PDA’s, game consoles, printers …
  – But there is so much more …
Class Project

• Grade based on class project tentatively in teams of 2’s

• Tentative scope of project
  – Pick an application
  – Design a customizable/accelerator core design in Verilog that can be attached to a processor core e.g. as a “data path” or a “peripheral” on the core bus
  – Demonstrate the benefits of the customizable/accelerator core using the Simple Scalar simulator or other means
  – The project can also leverage multiple “data path” units (e.g. multiple integer units), varying the number of registers, varying the cache sizes, etc.
Tools

- Verilog as HDL
- Leonardo for synthesis, mapping to cell library (e.g. TSMC), and timing analysis
- Mentor for logic simulation
- Simple Scalar simulator for processor modeling and simulation
Tentative Lecture Plan

- Lectures on embedded processors (2)
- Lectures on Verilog, CAD tools & Simple Scalar simulator (2)
- Lectures on application areas and case for application-specific processors (4)
- Lectures on system-on-chip concepts (2)
- Lectures on commercial & academic examples of application-specific processors (3)
- Lectures on network & graphics processors (2)
- Lectures on compiler & operating system issues for embedded processors
- Rest of class time used for project presentations & discussions
Information Flow

• Check the class website regularly !!!

http://www-cse.ucsd.edu/classes/sp02/cse291_E/index.html