

Advanced Topics ASIC, FPGA, HDL, Advanced Processors

Computer Organization

ASIC

- Application Specific Integrated Circuit
- ASICs are custom circuits
- Very fast and use less power
- Similar in theory to an FPGA
 - Exception is that it is fabricated as a custom circuit
 - This means that – unlike FPGAs – it is not reprogrammable
- Drawback of ASIC
 - Cost
 - To build an ASIC for you will cost hundreds of thousands of dollars as an initial investment
- Both FPGAs and ASICs are designed with a Hardware Description Language (HDL)
- The two most popular hardware description languages are VHDL and Verilog

ASIC

- Designed to perform a particular, specialized function
- Not software programmable
- Not a memory chip, but may contain memory
- Used in many popular electronic devices, like
 - MPEG decoder
 - Audio processor for Dolby noise reduction
 - Image processor for MRI
- Used in critical power-sensitive applications
 - Cell phones, mp3 players, and other battery-operated devices
- High volume, cost sensitive
- High reliability, high performance (mA/Mhz)

FPGA

Field Programmable Gate Array

Control Logic Blocks

- Can be combined as needed
- Logic circuits are implemented as **Programmable Hardware**

HDL

- A Language to describe, simulate, and create hardware
- Cannot be used like typical languages (C and Java)
- Express the dimensions of timing and concurrency
- At RTL, describes a hardware structure, not an algorithm
- At behavioral level, describes behavior with no implied structure
- Very large, complex, high speed ICs can successfully fabricated
- Schematics becoming unwieldy quickly
- The number of transistors and complexity exploded
- Needed a way to describe and simulate complex designs

HDL

- Logic Synthesis
- 10-20K gates/day/designer
- Design changes are fast and easily done (text vs. drawing)
- Leverage of SW design tools
- Optimization of designs is easier
- Exploration of alternative designs can be done quickly
- Easy to trade off time and complexity (speed vs. area)

Differences Traditional PL and HDL

Traditional Programming Language

- Modeled after a sequential process
- Operations performed in a sequential order
- Help human's thinking process to develop an algorithm step by step
- Resemble the operation of a basic computer model

Hardware Description Language

- Characteristics of digital hardware
- Connections of parts
- Concurrent operations
- Concept of propagation delay and timing
- Characteristics cannot be captured by traditional PLs
- Require new languages: VHDL, Verilog
 - Technology/vendor independent, Portable, Reusable

HDL

- VHDL
 - VHSIC (Very High Speed Integrated Circuit) Hardware Description Language
 - Considered as an alternative to IC datasheets, but it is a programming language
 - Designed and optimized for describing the behavior of digital circuits and systems used by industry worldwide
 - Objective is to shorten the time from a design concept to implementation from 18 months to 6 months

HDL

- VHDL
 - More difficult to learn
 - Strongly typed
 - Widely used for FPGAs, military
- Verilog
 - Simpler to learn
 - Looks like C
 - Weakly typed
 - 85% of ASIC designs use Verilog
- Once either is used, the other is learned quickly

- Beginning of Advanced Processor…

Advancement in Processor

- Higher chip densities
 - (Moore's Law) Not Possible
 - Because more transistors are not problem
 - Faster clock rates slice EX into tinier pieces
 - Power is an issue, a BIG ISSUE
 - Lots of delays in the system ... like accessing mem
- 1st idea: Use what we've got better: multithreading
- Keep multiple contexts, switch when there's a stall
- such L1 cache miss
- Need 2x registers, 2 PCs and some care in design
- Threads share caches, TLB, and all HW units

Advancement in Processor - Solution 1

- Multithreading
 - Keep multiple contexts, switch when there's a stall
 - such L1 cache miss
 - Need 2x registers, 2 PCs and some care in design
 - Threads share caches, TLB, and all HW units

Advancement in Processor - Solution

2

- Superscalar
 - Issue (Fetch/Decode) multiple instructions at a time
 - Keep careful track of which operand registers the computation depends on
 - Dependency of registers checked and executes multiple instructions in parallel at *same* stage
 - Actually execute instructions when data available
 - Commit instructions (write to memory, etc.) in order
 - Effective multi-issue machines have CPI < 1

Advancement in Processor - Solution

3

- Add multiple processors -- Parallelism
 - Parallelism is implemented using high transistor counts as multi-core processor chips
 - Using multi-core (and all parallelism) requires multiple instruction streams – threads
 - One option is to use separate tasks: OS, virus checker, multimedia apps, etc.
 - Only way to speed 1 computation is to express it as multiple threads of computation == parallel program
 - After decades of research automatic conversion of sequential programs into parallel programs

Advancement in Processor - Solution

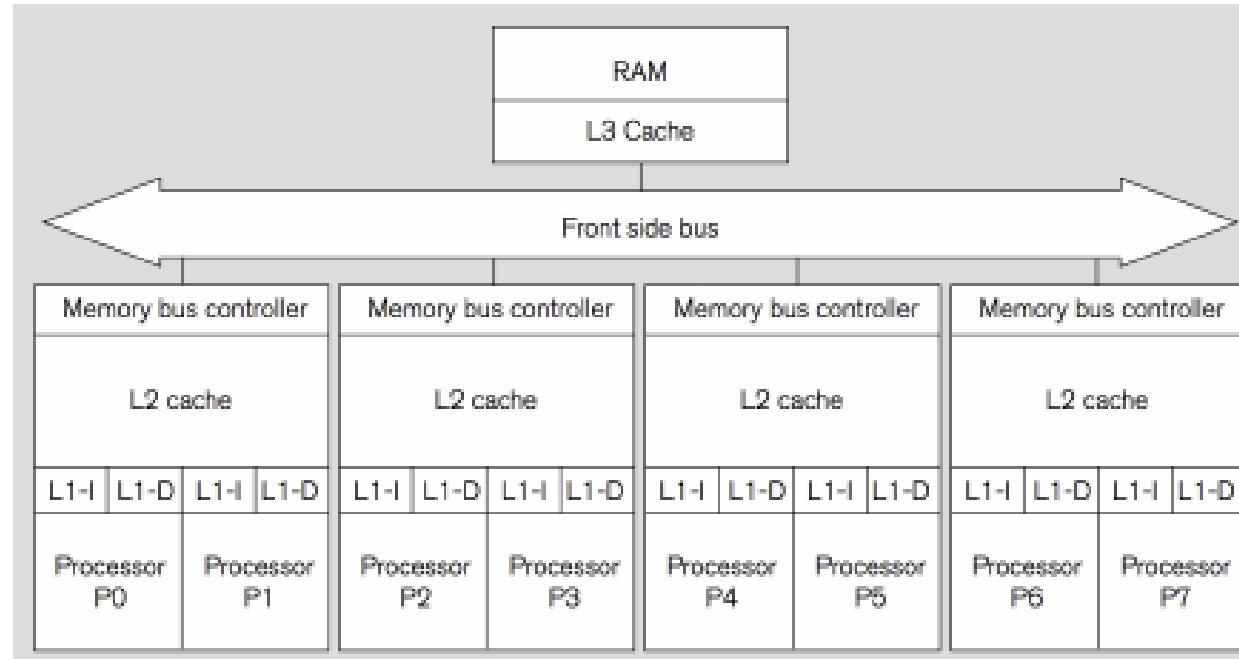
3

- Exploiting Parallelism of the computing problems for which performance is important, many have inherent parallelism
 - Best example: computer games
 - Graphics, physics, sound, AI etc. can be done separately
 - Furthermore, there is often parallelism within each of these:
 - Each pixel on the screen's color can be computed independently
 - Non-contacting objects can be updated/simulated independently
 - Artificial intelligence of non-human entities done independently
 - Another example: Google queries
 - Every query is independent
 - Index is read-only!!

Advancement in Processor - Solution

3

- Quad Core Logical Structure



Features of MultiCore Processors

- Standard pipelined architectures … built with the “best” features
- Often exploit multithreading
- Private L1 Instruction and Data caches
- Shared L2 and L3 combined caches
- Shared off-chip ports

MultiCore Processors

- Having more than one processor on the same chip
 - Soon all PCs/servers and game consoles will be multi-core
 - Results from Moore's law and power constraint
- Exploiting multi-core requires parallel programming
 - Automatically extracting parallelism too hard for compiler, in general
 - But, can have compiler do much of the bookkeeping for us

- All the Best…