EDP 2009
"Threads Are Dead"

Practical Realities

John Murphy

LIGA SYSTEMS

Affordable Acceleration

Recent and Relevant Experience

- **Liga Systems 2008-present**
  - **Value proposition:**
    - Reduce risk of logical design errors
  - **Major features:**
    - 10x speed up of logic verification
    - Plug-compatible with software simulation
    - Priced on par with software simulator
  - **Technology (HW and SW)**
    - FPGA-based co-processor with very high bandwidth to memory
    - Compile synthesizable and behavioral Verilog/VHDL
- **Athena Design Systems 2005-2008**
  - **Value propositions:**
    - Close multi-scenario timing fast with sign-off accuracy
    - Execute routing in ¼ of the time using existing computers in the farm
  - **Major features:**
    - Multi-mode, multi-corner timing closure using sign-off tools for each scenario
    - Automatic partitioning and stitching for data independent execution of partitions
  - **Technology (SW)**
    - Multi-scenario timing optimization algorithm
    - Layered software infrastructure "single machine" look and feel using distributed multi-processing
    - Distributed IC physical design database with partitioning and stitching algorithms

EDA Computing Discontinuity

- ***EDA tool run times frustrate designers – Market Opportunity!***
  - ***Designers need to do multiple turns per day or at least one per day***
  - ***Logic verification example***

Design Size: 40+ M Gates	Single CPU Simulation time (hours)
Test	
memory_debug	62
bad_pipe_fuse_read	67
Full_reset	403

Need 10-20X Speedup

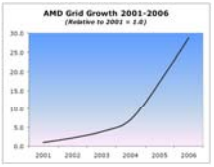
- ***As data sizes grow, turnaround time slows***
  - ***Design data sizes force design centers to upgrade machines fo***
    - *Memory capacity does not increase designer productivity!*
  - ***Routing example:***
    - *DEF size projected above 128 Gb*
    - *6,000 machines in the farm with 32 Gb or less*
    - *Budget allocated for only one big machine, however 5 projects planned next year*

Need Smart Partitioning

3

Alternatives: *More Servers!, BUT with Unintended Consequences*

- **Increases cost and energy consumption of servers, floor space, SW license cost**
- **Not addressing the core problems**
  - It's *not* the processor speed
  - It's more the cache size (*I/O speed* of the processor(s))
- **Rapid grid growth, worldwide**
  - 1,000+: Broadcom, Qualcom, HP, SUN
  - 10,000+: Nvidia, Intel, AMD, IBM
  - Any limit?
- **Designers manually partition to run in parallel where possible**
  - Increases throughput of number of jobs
  - Reduces time to run entire suite, but
    - does <u>not</u> reduce *turn-around-time* for the longest running job
- **Multi-threading has produced limited success**
  - Multi-threading produces 2X speed-up at best while consuming 4 licenses
  - Multi-threading does not address the data size/memory issue



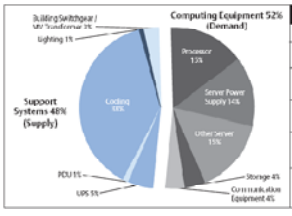
Source: AMD Company Data

Over the five-year period from 2002 to 2007, the HPC server market has grown an aggregate 134% at an average annual compounded rate (CAGR) of 18.8% - IDC

4

Crisis in the Server Room

"An overwhelming majority of facilities managers named power and cooling as the most pressing issues of concern to them." *Source: IDC*

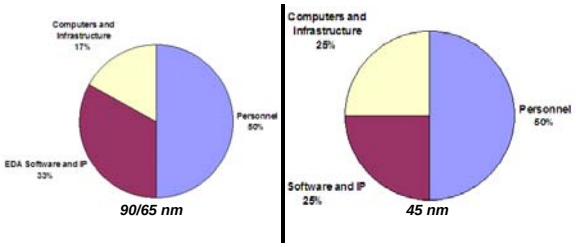


Category	Power Draw*
Computing	588 kW
Lighting	10 kW
UPS and distribution losses	72 kW
Cooling power draw for computing and UPS losses	429 kW
Building switchgear/MV transformer/other losses	28 kW
TOTAL	1127 kW

Typical 5000 sq ft data center daily energy usage. Source: Energy Logic

5

IC Implementation Budget Distribution



Sources: IBS, IDC, five major semiconductor corporations

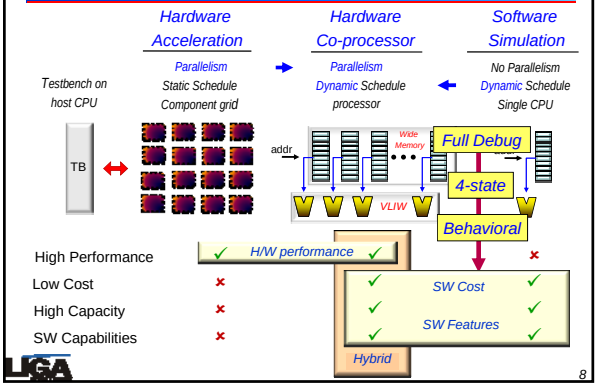
EDA Software and IP <u>lost share</u> to computers and infrastructure

6

How To Reduce Turnaround Times and Memory Footprint?

- **Tools and methodologies must incorporate smart partitioning coupled with parallel Computing**
- **Which path is best?**
  - **Data independent partitions and distributed multi-processing?**
  - **Change processor architecture?**
    - GPU?
    - VLIW?
  - **Multi-threading?**
- **Each choice has unique advantages and issues**
- **Choose carefully, then make the supporting technology a core competency**

Comparing Simulation Technologies



Liga Technology in Action

- **Runtime Comparison**
  - Design Size: 40+ M Gates

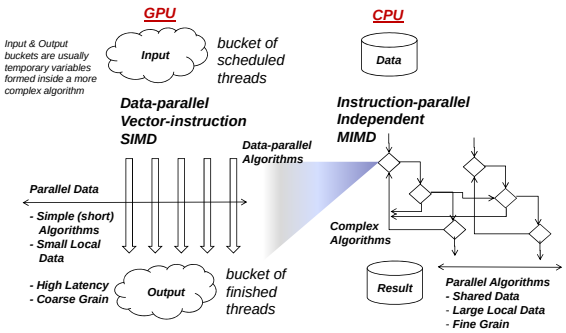
Test	SW Sim	NitroSIM (Liga)	Acceleration
memory_debug	62	7	9x
bad_pipe_fuse_read	67	8	8x
Full_reset	403	92	4x

The runtime unit above is hours

Co-Processor/VLIW Advantages and Issues

- **The hardware is relatively "easy", the compiler is tough**
  - **The Achilles heel of VLIW processors is to make the compiler "anticipate" branches since there is no fetch or cache**
    - Engineering expertise is scarce
    - Difficult to compile behavioral code and test-bench to hardware
      - Degrades performance
      - Inhibits usability/compatibility
- **FPGA based hardware is good and bad**
  - Economical for "low" volume applications like EDA tools
  - Dependent on Altera and Xilinx to address special requirements
    - Difficult to get their attention
    - Creates risk in the business
- **Custom silicon**
  - **More control in processor architecture and I/O**
  - **Economics don't add up**
    - Silicon spins very expensive
      - Lesson learned from Theras
    - A custom processor is a "billion dollar play"

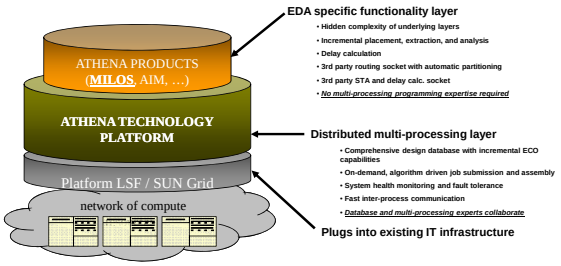
GPU versus CPU



GPU acceleration

- **More Bandwidth**
  - 150 times processing (single precision), 20 times memory bandwidth over single CPU
- **Data-parallel**
  - Very high acceleration possible: 200x or more has been reported (rendering)
  - Usually applicable to mostly local areas inside more complex algorithms
    - *Amdahl's law inhibits performance,*
      - *In hardware accelerated simulation it's the Test Bench/DUT ratio*
      - *In distributed routing it's the partitioning and stitching times*
- **Weakness**
  - Data-parallel imposes coarse granularity, requires complex rework of algorithms
  - Dynamic decisions have high latency – causing wait times
  - Not good for convergence or propagation algorithms or fine granularity
- **Reported Performances – real-life applications**
  - **Varying overall experience, higher if more data-parallel application**
    - Graphics rendering: 100x
    - Data-processing applications: 10x-20x
    - Compute-processing applications: 3x-4x
  - **Example:**
    - Oil reservoir 3D modeling: 20x (Data-parallel)
    - Oil reservoir fluid migration modeling: 1x (Fluid-dynamics)

Distributed: The Athena Technology Platform



EDA specific functionality layer

- Hidden complexity of underlying layers
- Incremental placement, extraction, and analysis
- Delay calculation
- 3rd party routing socket with automatic partitioning
- 3rd party STA and delay calc. socket
- *No multi-processing programming expertise required*

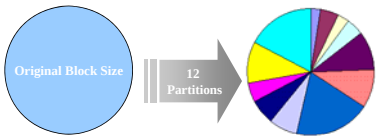
Distributed multi-processing layer

- Comprehensive design database with incremental ECO capabilities
- On-demand, algorithm driven job submission and assembly
- System health monitoring and fault tolerance
- Fast inter-process communication
- *Database and multi-processing experts collaborate*

Plugs into existing IT infrastructure

Athena technology was architected in layers for ease of installation and maintenance, ease of use for the end user, and for building a loosely interdependent development organization based on specific areas of expertise supporting the architecture.

Athena Technology In Action



- **Partitioning for routing**
  - Original block: 5.4 G
  - 12 tiles total 5.6 G
  - Largest block 20% of original block (1 G)
  - Total memory footprint 5% larger than flat
- **Reduced memory footprint by 80% and reduced runtime time by 4X**
- **Methodology:**
  - Only partition and stitch on big memory machine asynchronously
  - Each block independently routed using normal scripting on existing (small memory) farm machines.

Conclusions

- **Our market thirsts for faster EDA tools**
  - Semi companies invested in computers and infrastructure to solve the problem
  - EDA SW lost share of the design budget
- **The door is open for EDA companies to capture value based on parallel processing**
  - Must become a core competency
  - Must be very transparent to users and tool developers
- **Choose your approach to parallel computing carefully**
  - Beware Amdahl's Law

Embarrassingly Parallel Applications	*Dynamic Applications*	
Multi-threading GPU	*Data-Independent Partitions*	*Data-Dependent Partitions*
	Smart Partitioning Distributed MP	*Multi-threading*

Back-up