HP Archive

QUICKTURN SYSTEMS

TESLA CORPORATION

Optimizing Your Design Flow... How to Use Microprocessor and ASIC Emulators

Eric Amador

TESLA Corporation
1025 Buckland Avenue
San Carlos, CA 94070
Tel: (415) 637-9479
Fax: (415) 637-9361

Tim Chambers

Hewlett-Packard Co.
8245 N. Union Blvd.
P.O. Box 617
Colorado Springs, CO 80901
Tel: (719) 590-5570
Fax: (719) 590-7679

Naeem Zafar

Quickturn Systems, Inc.
325 East Middlefield Road
Mountain View, CA 94043
Tel: (415) 967-3300
Fax: (415) 967-3199

1993
High Speed Digital
Symposium

QUICKTURN SYSTEMS

TESLA CORPORATION

HEWLETT PACKARD

Abstract

The dramatic increase in the complexity of microprocessor-based designs and their associated system software has caused companies to employ larger design teams, and consequently to incur higher hardware and software design costs. In addition, modern digital systems now invariably include custom digital circuits in the form of proprietary Application-Specific Integrated Circuits (ASICs). Managing the technical design issues associated with these projects, and meeting time-to-market constraints, requires the use of advanced software and hardware development tools.

This paper discusses two important tools useful to designers of complex digital systems. The first is Hewlett-Packard's HP 64700 Microprocessor Emulation System. The second tool is comprised of Quickturn's Enterprise and RPM ASIC Emulation Systems.

The HP 64700 provides in-circuit emulation of commercial microprocessors. In this role it serves as an advanced Hardware-Software development and integration platform. The various features and options of the HP 64700 are discussed, along with the emulator's role in assisting system design and integration.

Quickturn System's ASIC Emulation systems provide complete in-circuit emulation of proprietary ASICs. In addition to complete in-circuit functionality, these systems provide access to all internal circuit nodes of the ASIC being emulated. The architecture, options, capabilities and applications of these versatile instruments is discussed. Case study data demonstrates "real-life" applications.

Authors

Tim Chambers

Current Activities:
Tim is currently with Hewlett-Packard as a Sales Development Engineer working at HP's Colorado Springs Division. He is responsible for sales development of HP's Microprocessor Development Tools.

Author Background:
Tim has worked at HP for 11 years. Prior activities with HP included research and development of debugger simulators supporting the Motorola family of Microprocessors. Tim graduated from M.I.T. where he earned a degree in Computer Science and Engineering.

Eric Amador

Current Activities:
Eric is President of TESLA Corporation. He is involved in advanced system design consulting activities for TESLA Corporation's clients. In addition to managing a small team of highly skilled consultants, Eric's principal design responsibilities include development of high-performance VLSI ASIC's and the complex board-level assemblies in which these ASIC's operate. TESLA Corporation specializes in ASIC emulation, and advanced system simulation technologies.

Author Background:
Eric has been an independent system design consultant for the past 16 years. During this time he has consulted on numerous projects for major corporations. He is named inventor on several US and foreign patents. Before becoming a consultant, Eric was Engineering Manager at Arnold Magnetics Corporation where he was involved in the design and manufacture of high-density switching power supplies. Eric earned his Bachelor's degree in Biochemistry at Michigan State University.

Authors (Cont'd)

Naeem Zafar

Current Activities:
Naeem is Director of Product Marketing at Quickturn Systems, Inc. Naeem is responsible for all product marketing activities including new product definition. Naeem's responsibilities include the RPM Emulation system and the Enterprise Emulation system. Naeem is actively involved in promoting Quickturn's strategy in the emerging re-programmable hardware market.

Author Background:
Naeem has 12 years experience in the electronic design and CAE industries. Naeem designed VLSI chips and worked in the area of computer architecture at Honeywell, Inc. Naeem was founder of a company, XCAT, which developed hardware accelerators for logic and fault simulation. He joined Quickturn Systems in 1988 and has held positions in engineering, technical marketing and product marketing. Naeem has a Sc.B. degree in Electrical Engineering from Brown University and an MSEE from the University of Minnesota.

Slide #1

Optimizing Your Design Flow... How to Use Microprocessor and ASIC Emulators

Hewlett-Packard
Quickturn Systems
TESLA Corporation

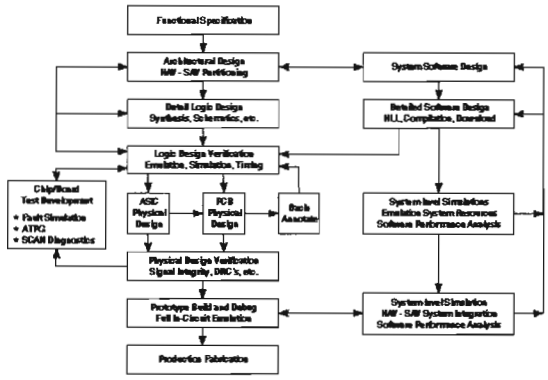
Slide #2

Agenda

- ☞ ❑ Digital System Design Process
- ☞ ❑ System Design & Integration Factors
- ☞ ❑ The Need for Emulation & Simulation
- ❑ HP 64700 Microprocessor Emulator
- ❑ Quickturn's RPM ASIC Emulator
- ❑ Summary of Methodologies

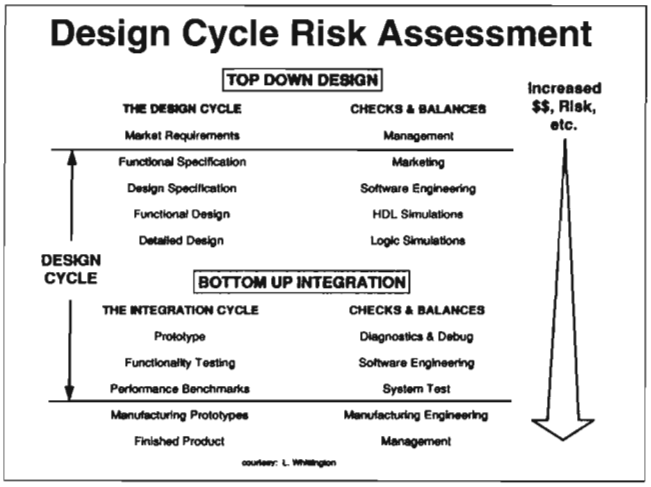
Slide #3

Simplified HS Digital Design Process



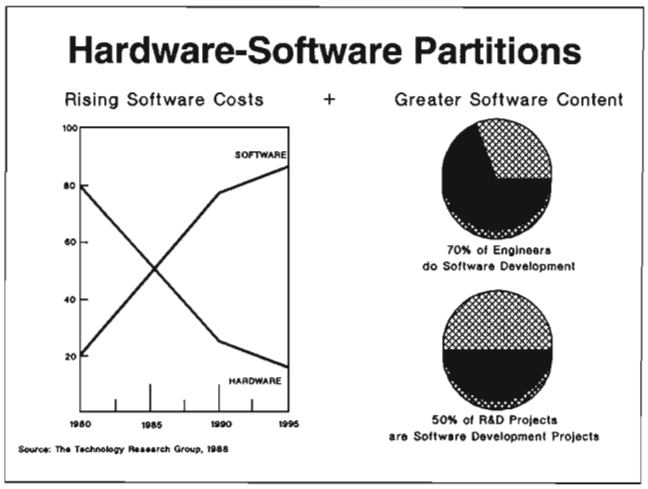
As shown here, in a complex system, the various engineering tasks in both hardware and software design have an iterative nature. The "top-level" aspects of a project are closely related to "bottom-level" activities. In order to save costs and meet project schedule constraints it is very important to minimize iterations through these "top-to-bottom" paths. Successful management of the complex interactions among these system design and integration activities is greatly improved through the use of the advanced design methodologies implemented in Microprocessor and ASIC emulators.

Slide #4



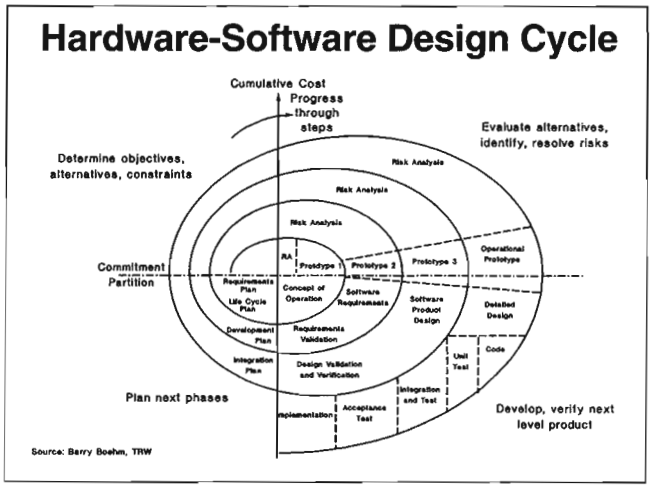
This diagram describes a "typical" design cycle and its associated check & balances. The "Increased Costs and Risk" along the right side of the diagram. Risk is illustrated by considering the effect on project costs which is caused by excessive iterations in the area defined as "Design Cycle". The role played by the Checks & Balances column in this description of the design cycle process is one which associates Design Tasks with Personnel Functions. That is to say, we are able to observe how personnel from various departments (Marketing, Software Engineering, etc.) must interact with each other as an automatic result of the performance of detailed design tasks (Logic Simulations, Diagnostics & Debug, etc.) This diagram serves to make clear the complex nature of present day system design. Excessive time spent in iterations of the "Design Cycle" dramatically increase the cost of a project and consequently place the success of the project at "Increased Risk".

Slide #5



The cost allocations between Hardware & Software are shown here. During the past decade software costs rose dramatically while hardware costs declined proportionately. Of course, overall total project costs have increased significantly as well, giving added importance to the need for timely project completion. Two additional design relationships are described here in the form of pie-charts. The lower pie-chart shows the increased emphasis on software-only projects. The upper pie-chart shows the distribution of manpower between Software Development and Hardware Development engineers. As indicated, in 1988 70% of Engineers were involved in Software Development. Keeping software development costs and schedules under control is of paramount importance to the success of a complex digital system project. Of equal importance, hardware engineers can benefit significantly from the use of tools which assist them in their work with software engineers. In this regard, HP's 64700 Microprocessor Development Tools provide hardware and software development engineers with highly efficient and powerful hardware/software design and debug capabilities.

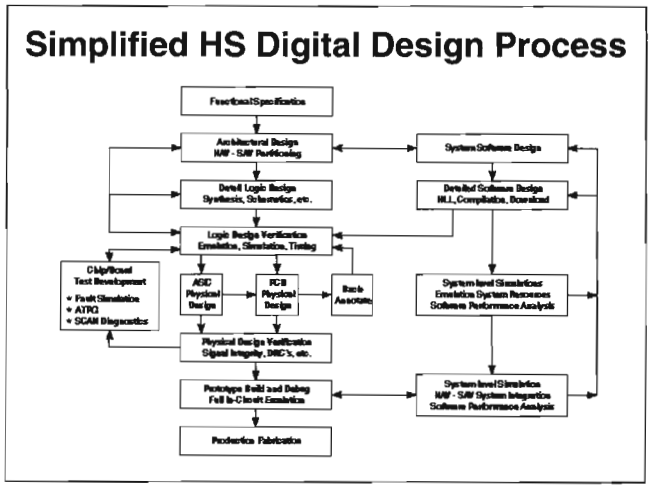
Slide #6



This diagram illustrates in a different fashion ideas similar to those described by the "Design Cycle Risk Assessment" slide. This representation accentuates the iterative nature of the design process. A typical system design begins at the center of the spiral with a Requirements Plan. The system progresses through Prototype 1 and Concept of Operation phases. This leads to a Life Cycle Plan. As shown here, at each iteration Risk Analysis is being performed. In our previous slide, the Risk Analysis was shown as "Checks & Balances"; here it is defined as an overall management function which takes place at various points in the lifecycle of a project. This chart emphasizes the planning phases of a major software project. As shown here, there are three Prototypes leading to the development of an Operational Prototype and a Detailed Design Specification. The subsequent phases following the Detailed Design are shown as ending in a finished Implementation of the Software System.

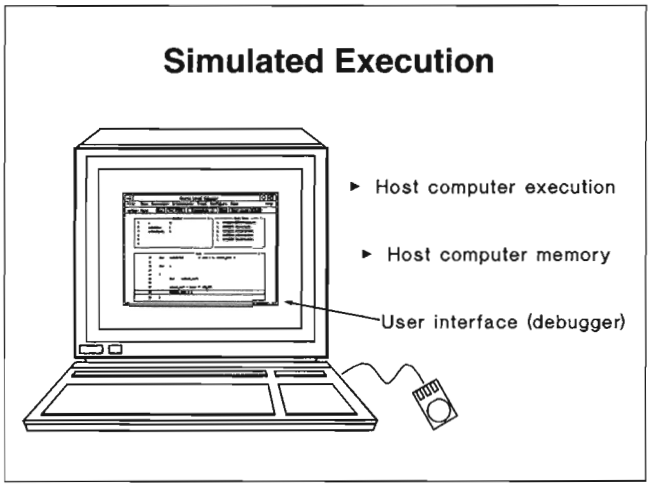
Slide #7

Slide #8



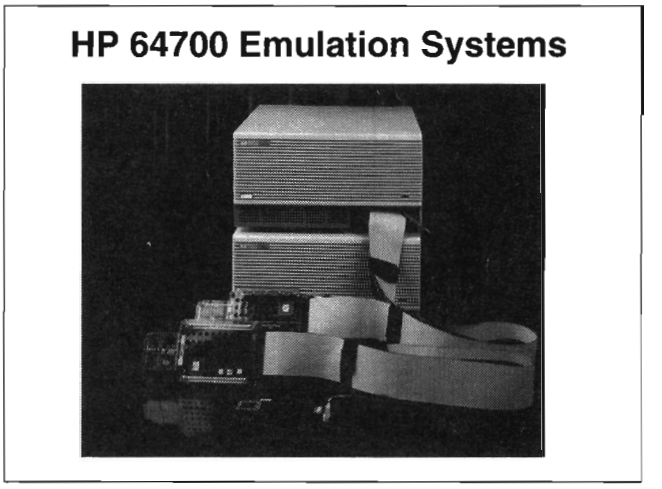
The HP 64700 Microprocessor Emulation system serves a valuable role in system design. As shown here, starting at the Hardware/Software partitioning function. The HP 64700 emulator plays a vital role in the design cycle all the way through the final stages of prototype verification. The emulator's real-time hardware/software analysis capabilities make it an indispensable tool initially during hardware debug and later during hardware/software integration.

Slide #9



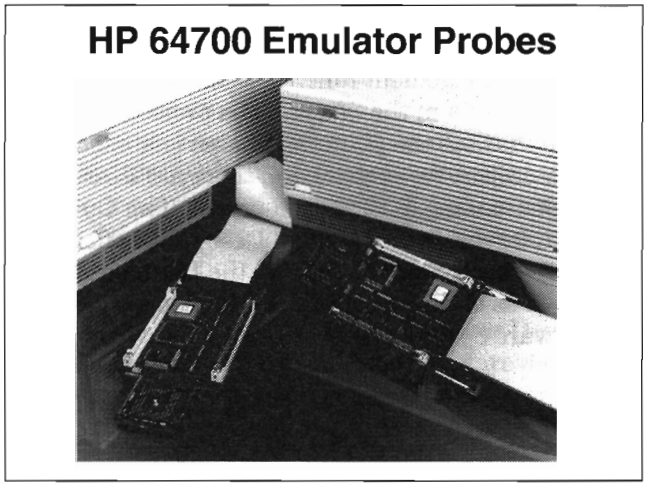
This slide shows an HP workstation being used to display the simulated execution of system software. This type of software simulation can also be performed using a remote host. In a case where a remote host is used to compile, run and debug the target code, potential exists for errors since the execution is not being performed by the actual target microprocessor. As it is shown here, the target microprocessor is being used to actually execute the code, thus providing a high measure of accuracy because the real system is being approximated much more closely. Simulations such as this rely on host computer memory to provide simulated target system resources. Extensive debugging capabilities are provided, including structured breakpoints and full-featured software performance analysis. As implemented on the HP 64000 System, simulated execution provides **hardware and** software development teams the capability to advance rapidly. Extensive algorithm debug can be performed without having to wait for target hardware prototypes. Once target hardware is available, the same environment can be incrementally mapped over during initial debug. When completed, execution of the target software via the Emulator permits exhaustive analysis and optimization of the target system.

Slide #10



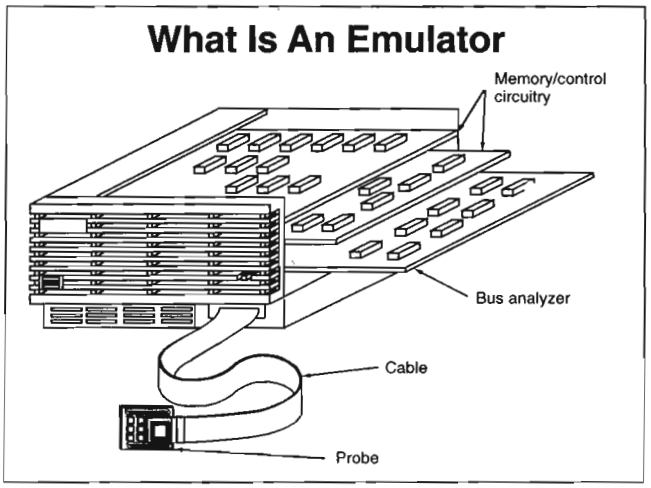
Two HP 64700 Emulator Systems are shown here. Their in-circuit pods are displayed, as is a small test-case printed circuit board. Comprehensive cross-triggering capabilities permit these instrument to be used in conjunction with each other, enabling hardware and software development of multi-processor designs.

Slide #11



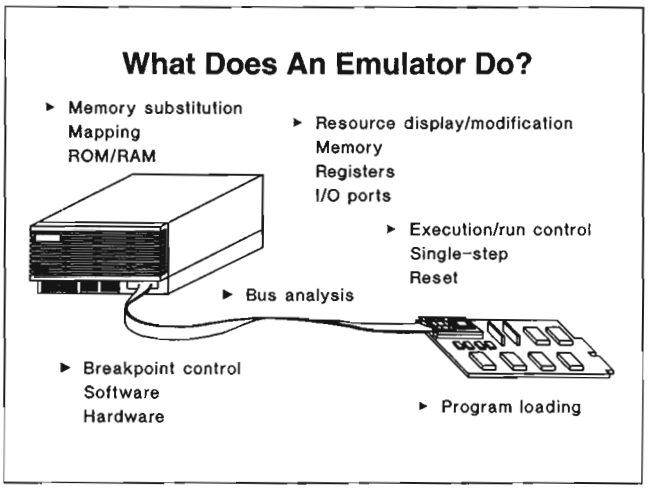
This photograph shows a close-up view of two HP 64700 in-circuit microprocessor pods. Real-time in-circuit performance is supported by the close proximity of the microprocessor device on the probe card. The ribbon cables connect the probes to their respective HP 64700 internal instrumentation cards. The "tip" of these probe cards provide a PGA plug which connects the emulator pod to the microprocessor socket(s) on the target hardware circuit boards.

Slide #12



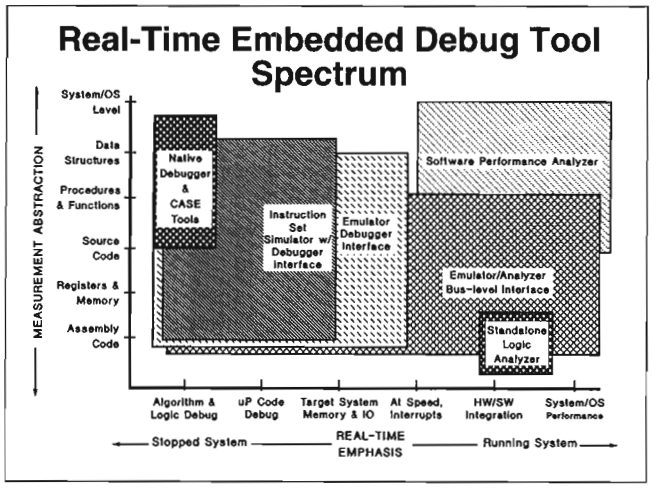
The architecture of HP's 64000 System is shown in this slide. As mentioned in the previous slide, the emulator permits the same code to be initially simulated and then later run in the target hardware. No changes to the user interface or tools are required. The probe assembly can be plugged into the target hardware system's microprocessor socket when hardware prototypes become available. Prior to a system prototype becoming available, the probe assembly can be used to execute code which is resident in memory provided by the HP 64700 emulator. The Bus Analyzer card permits the emulator to display cycle timing information, as well as other types of measurements useful during debug. Although not shown in this slide, the emulator chassis can hold additional options such as a State/ Timing Analyzer which permits independent probing of target hardware. Another option, the Software Performance Analyzer (SPA) can perform detailed real-time system performance measurements, which are often valuable in the course of real-time system hardware design and debug.

Slide #13



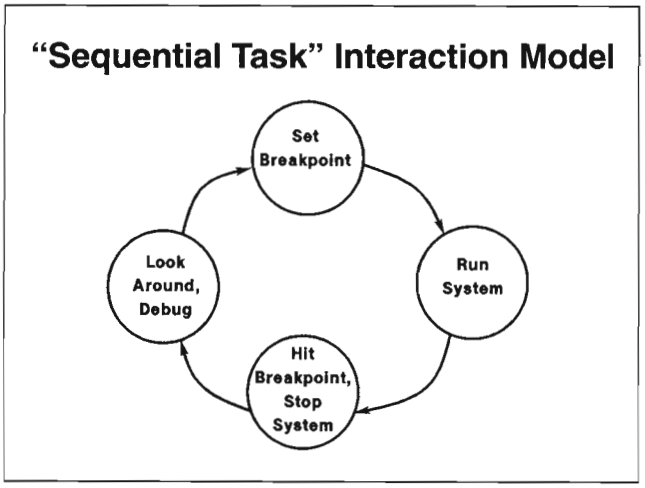
This slide depicts an overview of emulator functionality. The HP 64700 emulator provides memory resources which can be mapped so as to simulate target ROM or RAM. The mapping scheme is very flexible, thus allowing a hardware designer to rapidly change system memory configurations as required. Internal resources of the target microprocessor can be displayed and modified readily. This feature gives design engineers valuable insight into the operation of their selected microprocessor. Internal registers, I/O ports, memory, etc. can also be accessed. Program execution is under the complete control of the designer. Powerful breakpoint capabilities support parameters based on address, data, type of cycle, etc. The user can control execution in single-step and other modes. Of course, the microprocessor can be readily reset at any time. Loading of program code is extremely fast due to the real-time nature of the emulator itself. Complete analysis capabilities of the microprocessor's bus activity are provided directly to the hardware engineer via the probe's resources. Additional logic and timing analysis is supported via Logic Analyzer probes. In this fashion, both software and hardware environments can be fully observed and controlled during system development.

Slide #14



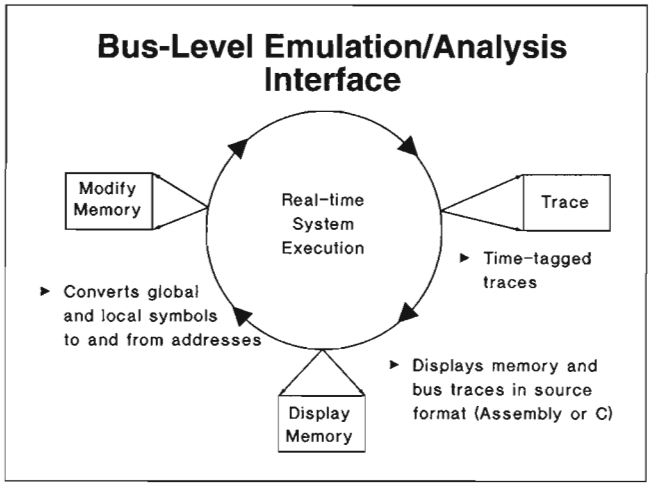
Shown here are the various resources provided by HP's 64000 Microprocessor Emulation System. The vertical axis of this graph is used to represent Software Measurements, characterized by their "Level of Abstraction". The software level of abstraction is a measure of "closeness" to the execution of individual instructions. Thus, the Operating System itself (System/OS) is at the top and hence the most abstract element, while individual instruction are at the bottom and represent fundamental "primitive elements" of the target microprocessor. The horizontal axis represents the dependence of the target system on real-time issues. Overall performance is judged and optimized by adjusting the various elements which are placed along this axis. At the origin we show individual algorithms as the "primitive elements" of real-time performance. At the extreme right of the horizontal axis we again show the System/OS performance, since it incorporates the elements of complex real-time system behavior.

Slide #15



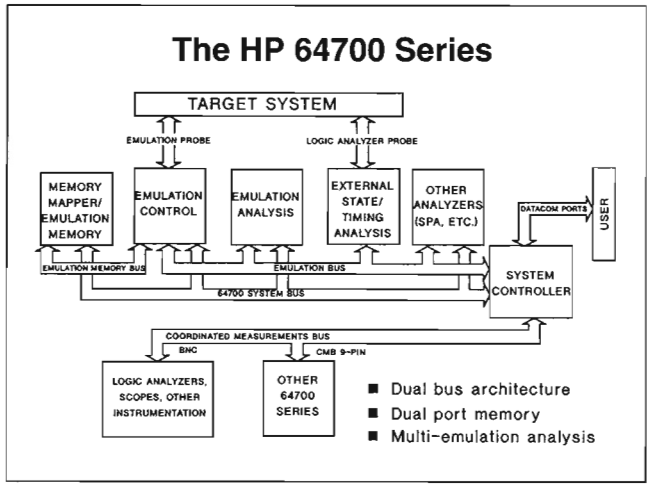
This slide graphically describes a "Sequential Task" debug strategy in which the user sets a breakpoint then starts execution of system code. Upon encountering the breakpoint, the system is halted and the user proceeds to examine the state of the system. The user can inspect registers, memory locations, etc. If the user uncovers a bug as a result of hitting the breakpoint, he proceeds to debug the system code or hardware, or both. The cycle is then repeated with the same or different breakpoints. This style of debug is sequential and, for the most part, not sufficient in the debug of complex systems. One of its disadvantages is the lack of real-time system monitoring. There are other disadvantages as well, including the lack of control and visibility over system hardware, etc. This strategy should be used only for simple debug scenarios where a single, static breakpoint will suffice.

Slide #16



This slide shows a "Parallel Task" debug strategy which makes full use of the Emulator's resources. At the center of this slide we show the target software as it executes code in real-time. Large amounts of data can be gathered and processed by the Software and Bus Analyzers. The user thus can derive useful conclusions regarding the operation of complex system software. In addition, this strategy provides time-tagged software execution traces permitting detailed timing analysis of software execution. Advanced high-level debugging facilities include global and local symbol mapping, as well as complete control over system memory. When combined with a Logic Analyzer, events in the hardware can be displayed correlated to microprocessor code execution. Triggering capabilities are synchronous across both Logic and Software Analyzers. Source level software debugging provides advanced trace capabilities which make source code debug possible in either Assembly or C. This slide summarizes state-of-the-art embedded controller debug techniques.

Slide #17



The HP 64700 Series architecture is described in this figure. The various functional assemblies are labelled as Memory Mapper/Emulation Memory, Emulation Controller, Emulation Analyzer, External State/ Timing Analyzer, Software Performance Analyzer, etc. The HP 64700 System Controller, shown to the right, provides the user and external interfaces. Shown at the top is the Target System. Both Emulation and Logic Analyzer Probes are shown connecting to the Target System. Finally, along the bottom we have ancillary debug instruments which can communicate with the Microprocessor Emulation System. These instruments include external Logic Analyzers, Oscilloscopes, and other laboratory instruments. It is important to note that design and debug of systems comprised of multiple processor systems can be supported by using additional 64700 Systems. The HP 64700 contains an internal dual-bus architecture. The microprocessor emulator itself is controlled and analyzed using a high-speed Emulation Bus. This Emulation Bus is completely dedicated to providing real-time support of the Microprocessor Emulator and the various Analyzers. The second bus is the 64700 System Bus. This bus is used for slower non-real-time system operations such as downloading code or recovering trace information from the Analyzers for post-processing and display to the user. The combination of these two busses provides the high-bandwidth data processing capabilities required to sustain real-time system debug. A third bus, called the Coordinated Measurements Bus provides synchronous and state information useful when external instrumentation is connected to the HP 64700 System. Complex cross-triggering among the various internal Analyzers and/ or external instruments is possible. Extensive timing and performance parameters permit extremely flexible instrumentation setups. The overall result is that users can readily perform complex debug tasks and accurately measure system performance.

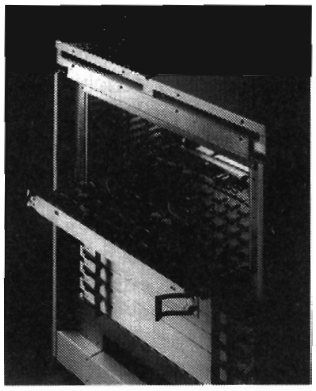
Slide #18

Agenda

- ❑ Digital System Design Process
- ❑ System Design & Integration Factors
- ❑ The Need for Emulation & Simulation
- ❑ HP 64700 Microprocessor Emulator
- ☞ ❑ Quickturn's RPM ASIC Emulator
- ❑ Summary of Methodologies

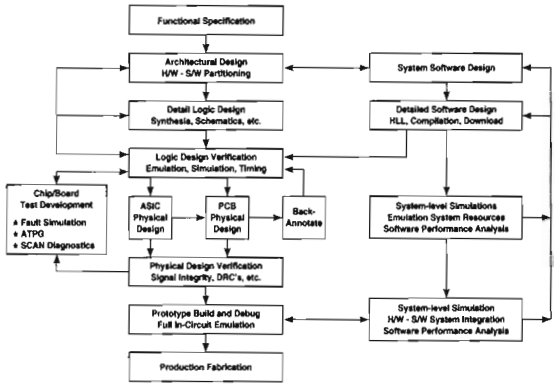
Slide #19

RPM Emulation System



Slide #20

Simplified HS Digital Design Process



QUICKTURN SYSTEMS

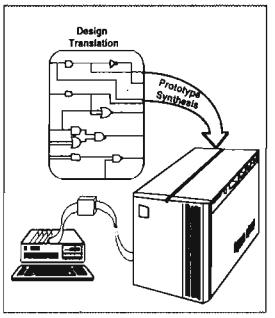
TESLA CORPORATION

HEWLETT PACKARD

Slide #21

Computer-Aided Prototyping

- *Automatic Creation of Hardware Prototypes ...*
- Read any design Netlist & automatically partition, place & route on array of FPGAs
- Map memory & logic elements automatically
- Access to all internal nodes through built-in logic analyzer
- Ability to make incremental design changes



Computer-aided prototyping (CAP) is a methodology for achieving system design verification. It provides for early system integration and concurrent design verification. CAP includes CAE interface software, design-to-prototype translation software, timing analysis software, reprogrammable hardware (including logic and memory), debug instrumentation and in-circuit interface cabling.

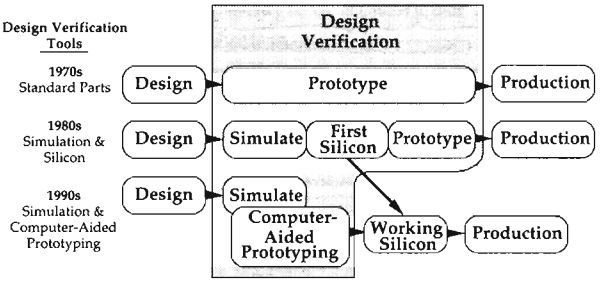
Interfaces to all popular design environments with netlists generated from schematics or synthesis when designs are captured in VHDL.

CAP provides design prototypes in hardware that can be verified in the system being designed where all software, hardware and interfaces can be thoroughly debugged before committing chips to silicon fabrication. It provides an environment for making hardware and software trade-offs, optimizing architectural approaches, and the development of more complete diagnostics software.

CAP prototypes are real hardware prototypes that use real gates and real wires. The implementation technology is reprogrammable logic using FPGAs. All interconnect between gates, interconnect to debug instrumentation, and interconnect to in-circuit interfaces is completed automatically with reprogrammable interconnect. In-circuit adapters plug directly into sockets where the ASIC chip will go when it has been fabricated in silicon.

Slide #22

Computer-Aided Prototyping

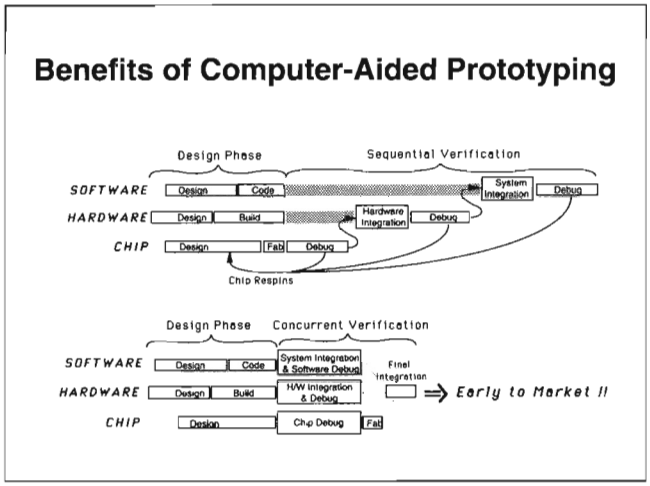


All systems are prototyped whether there is a plan to do so or not. The first article is always a prototype. Unfortunately most system prototypes are done with ASICs in silicon where debug and design changes are difficult and time consuming.

CAP allows the first article of an entire system to be prototyped before fabricating silicon so that design verification and debug can be completed in an environment where all internal nodes of ASICs can be observed, controlled, and changed quickly and conveniently.

CAP moves the first article of the system into the domain of design verification where it belongs and reduces the total development time, produces higher quality products, and reduces risk of schedule slips.

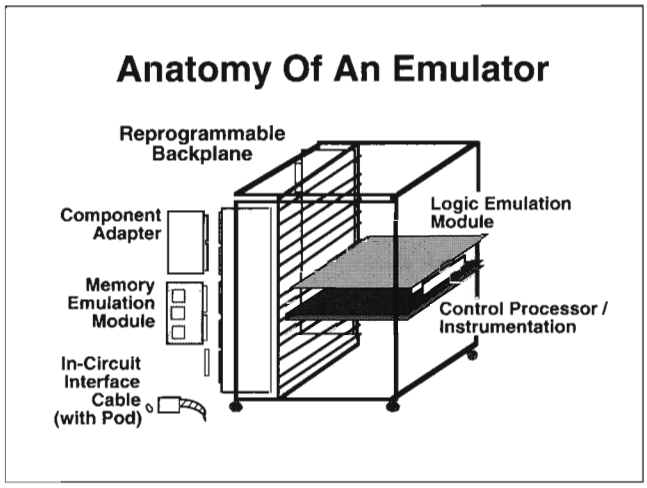
Slide #23



Various pieces of the product are designed by the user separately but they all come together at one point. Usually boards cannot be assembled until the chips are verified and the whole system verification has to wait for the hardware to be fully functional. This traditional approach is inflexible. Often problems are found very late in the development cycle and are not always fixable without changing hardware, compromising software or worse, re-spinning the chips.

CAP provides a way to do product integration much earlier in this cycle. All pieces of the system can be brought together and verified before ever fabricating chips. Silicon is made only after all pieces of software and hardware have been verified together. This results in much shorter system integration phase, higher quality due to fewer compromises and being earlier to market.

Slide #24

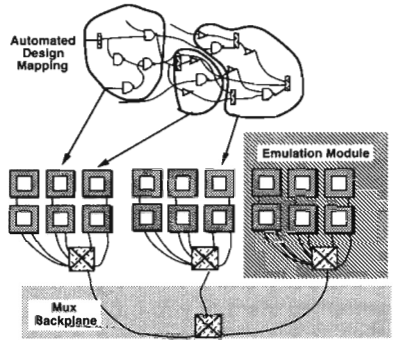


The emulator accepts 11 cards, called Emulation Modules. Each emulation module has a usable gate capacity of 30K gates. There is a Control Processor card which is used to communicate to the host workstation and contains a 1024 channel logic analyzer. The interactive backplane provides electronic switching of signals among these Emulation Modules. Memory emulation cards are plugged into the backplane. These cards are automatically personalized by the mapping software and can emulate up to 2MByte of RAM per card, including multiport RAM (up to 32 port wide). User can plug up to 32 Memory Emulation Boards in a system. User has access to over 6000 I/O signals. Component Adapter cards allow users to place existing ICs and plug them into the emulator backplane. In-Circuit interfaces provide the connection to the user's target system.

Designs are automatically read in from just about any format including Verilog, EDIF and ASIC vendor formats. They are automatically partitioned and mapped on to the emulation boards. A graphical user interface gives user's complete access to the internals of their designs.

Slide #25

Automatic Design Mapping



Designs are automatically read in from just about any format including Verilog, EDIF and ASIC vendor formats. They are automatically partitioned and mapped on to the emulation boards. A graphical user interface gives user's complete access to the internals of their designs.

A portion of the user's netlist is assigned to an FPGA and all FPGAs are connected via a custom interconnect chip (MIC) which acts as a cross-bar. Then all emulation modules can be connected electronically using another rank of MIC chips. This provides an expandable architecture where user can add more capacity modularly as his needs grow.

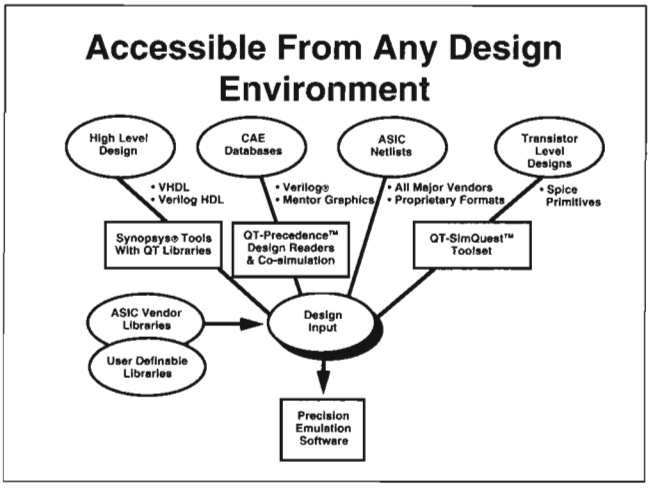
Slide #26

Enterprise Emulation System

- **Higher Capacity System**
  - » 330K Gates, 64Myte of Memory per Enterprise Emulation System
  - » 6M+ Gates per Multi-Enterprise Cluster
- **Faster Emulation Speed**
  - » 2 to 4 MHz typical
  - » Up to 8 MHz
- **Most Advanced Software**
  - » Connected with Major CAE Environment
  - » Most Efficient Mapping of Design

Enterprise emulation system is the next generation emulation system. Based on custom interconnect devices, latest FPGAs and a patented new Hierarchical Multiplexed Architecture this system delivers emulation for ASICs and custom ICs from 30K gate to 330K gates. Memory emulation is handled just as easily through special reprogrammable cards called Memory Emulation Modules. Enterprise emulation system connects to many CAE systems. It allows co-simulating mapped design from your software simulation environment before you may plug it in-circuit for system verification at hardware speeds.

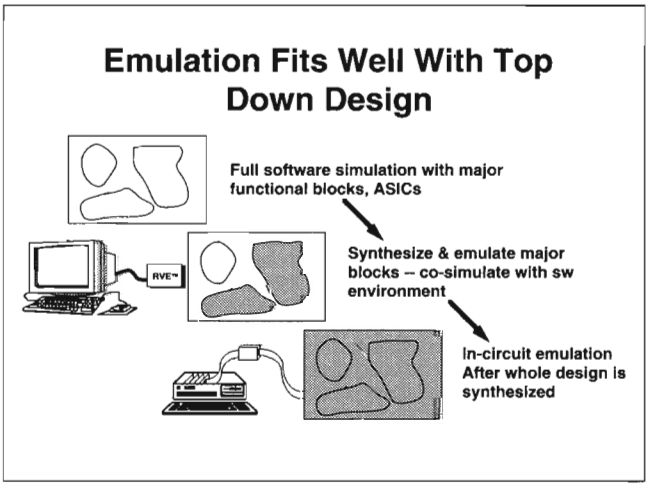
Slide #27



A rich set of design readers allow users to read-in a design from just about any design environment. Many translators allow users to read in design in SPICE, Verilog® or Mentor Graphics databases or just about any ASIC vendor format. Designs from VHDL can be synthesized using Synopsys™ emulation library for optimal synthesis for emulation.

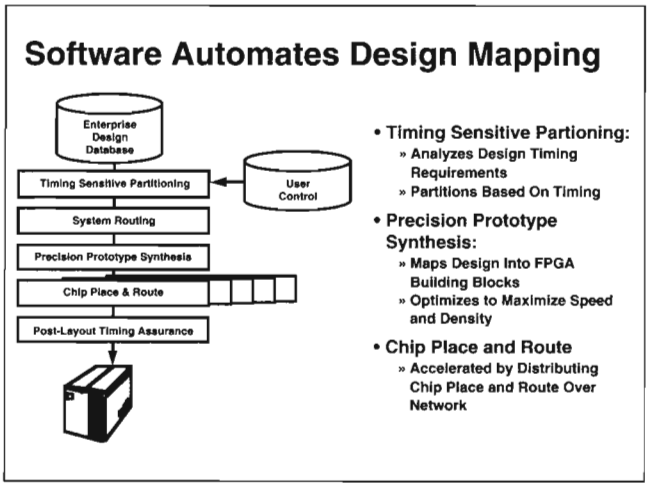
Many ASIC vendor libraries are supported. User can develop their own libraries using the Library Development Kit for proprietary cells.

Slide #28



As part of the top-down design methodology users do a complete behavior simulation first. They synthesize block at a time and as a portion of the design becomes available in gate level netlist, it can be emulated. The emulator connects with the software simulator via a device called Rapid Vector Evaluator (RVE™). RVE along with special co-simulation software permits gradual and smooth transition into the full emulation.

Slide #29



Precision Emulation Software™ automatically maps designs read-in by first partitioning into logic blocks which will fit into an FPGA. Once the design is partitioned, using timing driven techniques, logic cells are routed among FPGAs using system router through custom interconnect chip. The FPGA place and route can be spread among multiple workstation on the network. Once place and route is completed, a comprehensive post-layout static timing analysis is run to assure full functional equivalence. Software calculates emulation speed, identifies and fixes any potential hold violations and allows users to time certain asynchronous paths.

Slide #30

Alcatel (Rockwell International)

First Time Success Rate Restored

	Total # ASICs	# Worked First Time	% First Time Success	
Prior to 1986:	36	36	100%	< 15K Gates/ASIC Well-Defined Specification Gate-Level Simulation
1986 - 1989:	17	4	23%	Increased Complexity (20K-50K gates/ASIC) Unproven Specification RTL + Gate-Level Sim.
1990 - 1991:	12	12	100%	Increased Complexity (30K-70K gates/ASIC) RTL, Gate Simulation + Quickturn Emulation

Emulation is the way to manage increasing complexity in telecommunication systems

Rockwell International— now Alcatel— were one of the first users of Computer-aided prototyping. They realized that as chips got above 20K gates the test vectors alone could not be relied up comprehensive testing of designs. The first silicon failure rate decreased significantly. New design methodology, including emulation, changed those statistics for a 100% success even after the designs grew over 50K gates.

Slide #31

5370 Superminicomputer Project Overview

New Generation 50 Series Dual Processor

- Same Architecture as the 6650 ECL Processor
- Uses CPU ASIC Components from 5340 CMOS Processor

Very Complex CISC Architecture

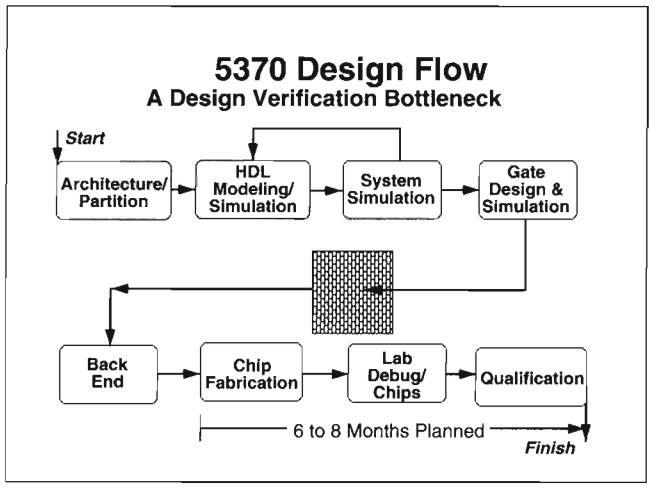
- Dual CPU with Crossbar Switches for Memory & I/O
- New Memory/Cache Controller

Required Development of Multiple New ASICs

Prime Computer of Natick, Mass. used CAP from Quickturn Systems to develop a new generation of superminicomputers, the 5370. The 5370 replaced a previous generation computer. ASICs were used to reduce the number of boards in the system and several new features were added including multi-processing.

The goal was to put all the logic that once required 8 boards onto a single board by integrating much of the logic into 50K CMOS gate arrays. The system achieved every objective and outperformed the previous generation computer by 25% to 41%.

Slide #32



Prime's design flow started with architectural partitioning and a description of the design in HDL. The design system was simulated with an HDL simulator and when the design was relatively stable the ASIC portion was recaptured with schematic capture at the gate level. Then gate-level simulation was performed using vectors captured during system simulation.

This design flow was an advance from previous flows by the addition of HDL system simulation because it reduced the time required to complete the total simulation task from the previous flow which used only gate-level simulation. Unfortunately Prime calculated that it would take thousands of years to simulate the equivalent of 30 minutes of diagnostics software running in real-time.

The original schedule for the time it would take to ship systems to customers after they sent ASIC data bases to fabrication was 6 to 8 months. This process included prototype chip fabrication, production chip ramp up, software check out, and final system Q/A and test.

Slide #33

Design Verification Bottleneck

Microdiagnostics Were Simulated

- Simulations Began to Take Multiple Days
- Clock Cycles Took 6 Seconds on Sun 4/330
- Diagnostics Would Have Taken 7600 Years

Simulation Alternatives

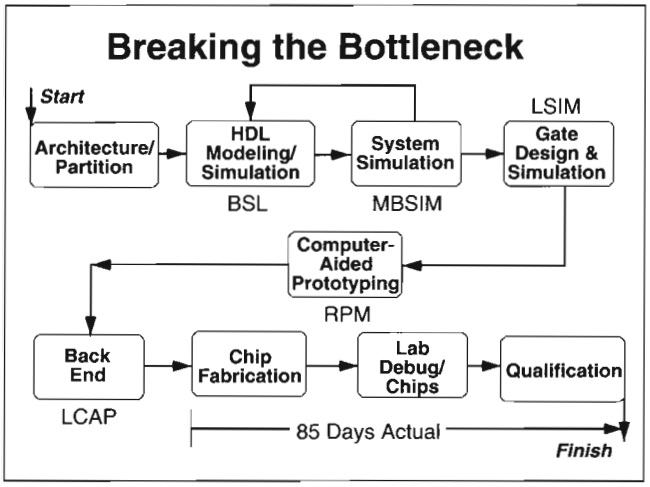
- More Compute Power
- Custom Simulation Language
- Hardware Acceleration
- Hardware Modeling
- FPGA Breadboards
- Computer-Aided Prototyping

Prime's development strategy was to simulate portions of the diagnostic programs (called microdiagnostics) that were used with the previous generation of super-minicomputer and when the diagnostics executed correctly the system design including the ASICs would be known to be correct.

Simulations of the microdiagnostics began to take multiple days to complete. Prime knew it had to evaluate alternative approaches to design verification if it were to avoid major schedule slips.

Prime evaluated several alternative ways to accelerate the design verification process. They considered more compute power, a new simulation language, hardware accelerators for their simulators, the addition of hardware modeling to their system simulator, building FPGA breadboards manually, and CAP with Quickturn's RPM Emulation System.

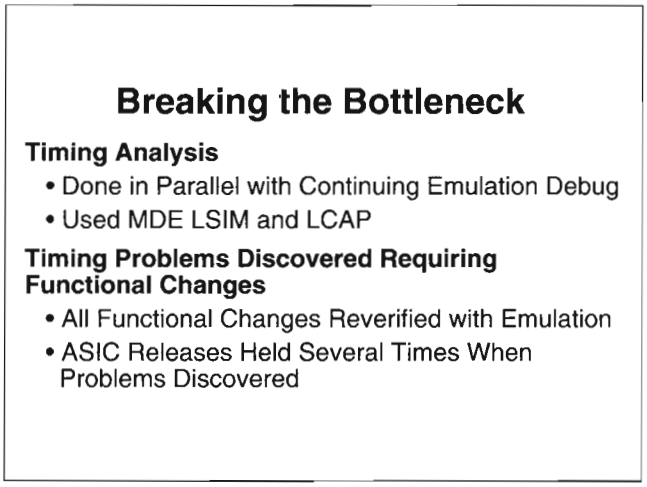
Slide #34



Prime selected CAP as the only viable way to achieve their product and schedule goals. They built a full system prototype using CAP for the ASICs. The design verification strategy was enhanced with CAP to execute the full diagnostics suite before fabricating silicon.

The design was simulated at the HDL-level, then the gate-level version of the design was recaptured and simulated at the gate-level to assure correlation with the HDL-level. Then a system prototype was built, the gate-level netlists were loaded into the RPM and plugged into the system in the sockets where the ASICs would eventually go when they were fabricated in silicon.

Slide #35



Timing analysis was run in parallel with CAP because CAP is a functional verification methodology and is ASIC technology independent. Timing analysis uncovered several flaws requiring functional changes to the design. All functional changes to the design were reverified with CAP before designs were released for silicon fabrication.

The net result was a reduction of many months in the delivery time to customers. The delivery time was actually cut by 6 months compared to a previous project of lesser complexity.

Slide #36

Breaking the Bottleneck

Run PXIO Assembly Language Diagnostics

- Emulation Took 12 Hours; Real-Time Takes 1/2 Hour
- Several "Show Stopper" Bugs Discovered

Boot Primos Operating System

- Not Part of Emulation Debug Test Plan
- Uncovered Further Hardware "Show Stopper" Bugs
- Uncovered Many Microcode and System Software Bugs

Saved Multiple Respins

The full diagnostics suite was run on CAP and Prime was ready to send out for silicon prototypes. The diagnostics took about 12 hours with system clocks running at approximately 1.2 MHz. These diagnostics take about 1/2 hour in real-time. During the time the diagnostics were run several show stopper bugs were discovered that would have required silicon to be respun had they not been caught ... and simulation would not have caught them.

Prime had met its objective of executing full diagnostics before fabricating silicon and so they went ahead and sent data bases to LSI Logic for silicon fabrication.

It was not part of the plan but they invited the operating system people to run the Prime operating system on the test bed (system Prototype) while silicon prototypes were being fabricated. Much to their surprise they discovered additional show stopper bugs the diagnostics had not uncovered.

They stopped the silicon prototype fabrication process and continued to run the operating system software and found more bugs.

In the final analysis several silicon prototype respins were saved.

Slide #37

Record Hardware "Bring Up" Time

Fast Silicon Fabrication

- 2 1/2 Weeks to Prototypes
- Risk Production Started

Chips Replace Emulation Cables

- Primos Booted at Emulation Speed (1.2 MHz) 1 Hour After Receiving Chips
- Systems Running at Full Speed (55 MHz) in 24 Hours

Alpha Systems Installed in 2 Weeks

Prime achieved a record hardware "bring up" time after releasing the final designs for silicon fabrication. They received silicon prototypes 2 and 1/2 weeks after releasing design data bases. They unplugged the RPM Emulation Systems, plugged in the chips, and within 1 hour after receiving the prototype chip the system was running at 1.2 MHz.

Within 24 hours after the chips were received the system was running at the full design target design speed of 55 MHz and alpha sites were installed within 2 weeks.

Slide #38

More Than Time-To-Market Benefits

- **Problems Uncovered Were Not Limited to ASICs**
- **Emulation Uncovered Problems Much Earlier In Design Cycle**
- **Problem Fixes Designed In ... Not Added On**
- **Shorter Qualification Time and Higher Quality Products!**
- **Team Worked Harder ... New Sense of Pride**

Prime's experience demonstrates that CAP enhances the system design verification environment in ways beyond ASIC chip verification.

• Over 50 board-level bugs were discovered by debugging with CAP in the system that had nothing to do with the ASIC designs themselves.

• Bugs were discovered much earlier in the development cycle than would otherwise have been possible with simulation alone.

• Problems within the ASICs were discovered before chips were fabricated so the fixes were made by correcting the chip designs instead of making patches external to the chip or in software.

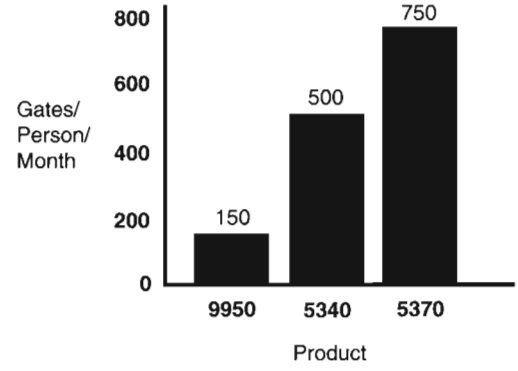
• The product qualification time was dramatically reduced because the complete product was so thoroughly verified before the qualification process was even started.

• The product quality was enhanced because the diagnostics programs that would be shipped with the product were refined for test coverage and changes to improve testability were made to the chip designs.

• Finally the team seemed to have a renewed sense of enthusiasm when they could get hands on the real system with CAP.

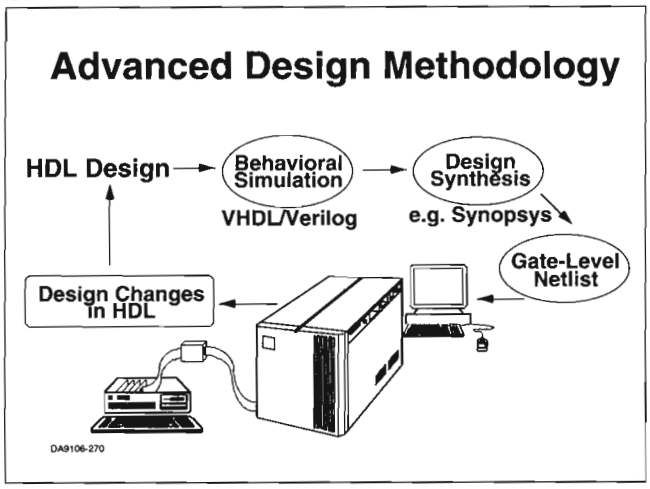
Slide #39

Increasing Design Productivity



Design productivity shot up dramatically on the 5370 project. They designed more gates in less time than the previous project such that the average number of gates per designer per month increased by 50%.

Slide #40

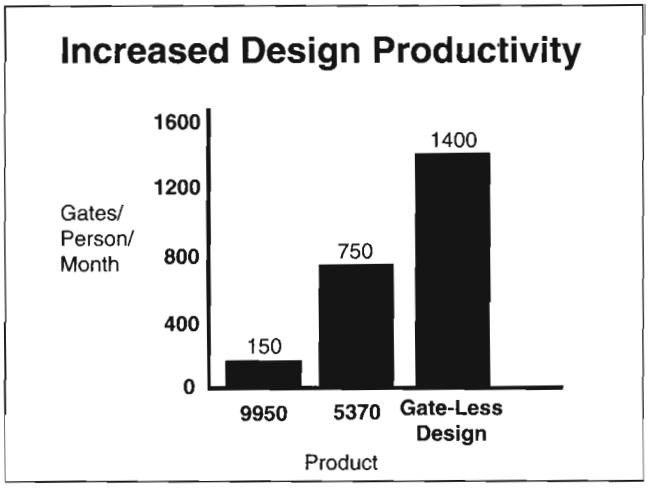


This advanced design methodology is getting to be very popular among the designers of high performance digital systems. A collection of advanced tools is resulting is increased productivity for the designer and allowing them to create and deliver more complex products in record time.

Using HDL to input design and doing behavioral simulation gives early affirmation of the design. Users can then synthesize the design and is able to see the design working in real hardware using the Computer-Aided Prototyping. Users debug the system with CAP and are able to modify designs as they debug, with a very short turn-around cycle.

Using CAP for comprehensive functional verification gives user more time to do timing analysis or timing simulation of their design using the conventional tools.

Slide #41



Prime expects to achieve another dramatic improvement in designer productivity by going to the gateless design methodology. They completely eliminate gate-level design. This saves the time required to recapture the design at the gate-level and the time required to correlate it with the HDL representation.

Slide #42

Feature Summary

	HP 64700 MicroProc. Emulator	Quickturn RPM ASIC Emulator
Commercial MicroProcessors	Real-time S/W & H/W Support	
Commercial VLSI Devices		Limited Capabilities
Proprietary VLSI Devices		Near Real-time S/W & H/W Support
Internal Register Visibility	Complete Access	Complete Access
Software Performance Analyzer	Real-time S/W & H/W Support	
Logic Analysis Capability	Comprehensive Feature Set	Comprehensive Feature Set
VLSI Design Modifications		Incremental Netlist Changes
Software Design Modifications	Complete S/W Tools Available	
Scope of Support	Mainstream Microprocessors	Major ASIC Vendors Fully Supported

Hewlett-Packard, c/o: Dave Bellandi, 1900 Garden of the Gods Road, Colorado Springs, CO 80907

QUICKTURN SYSTEMS TESLA CORPORATION HEWLETT PACKARD