An Optimized Methodology for High-Speed Design

DESIGNCON 98

Roger Perger, Intel Corporation
Donald Telian, Cadence Design Systems

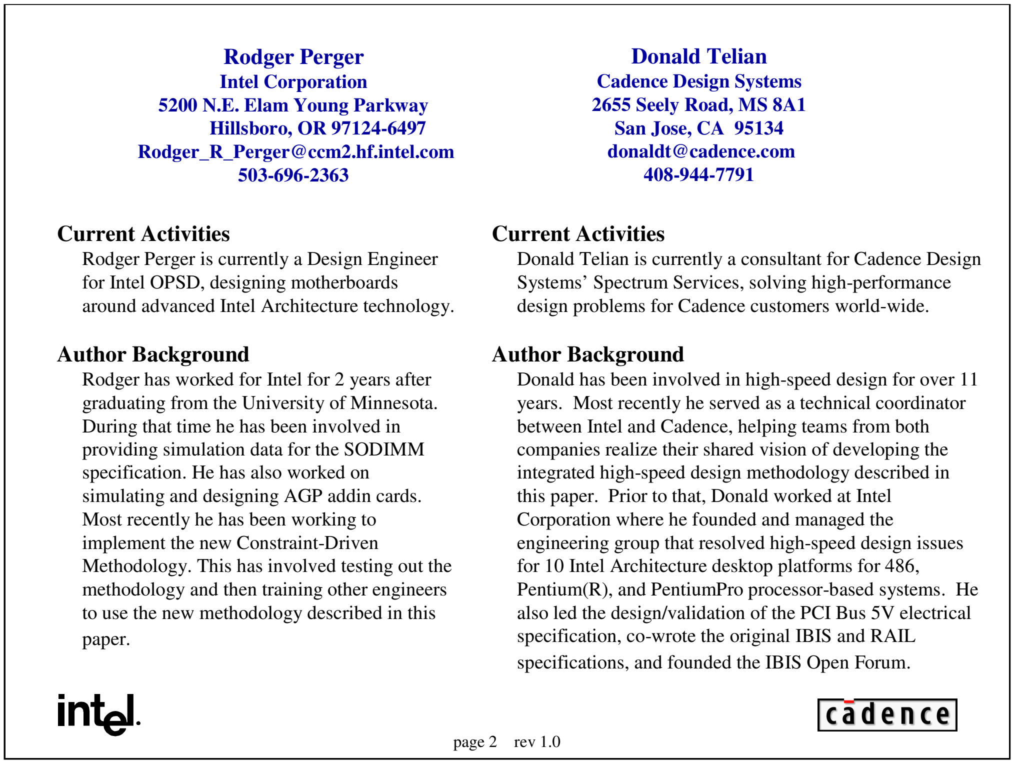
intel.

cadence

page 1 rev 1.0

ABSTRACT:

In recent years High-Speed PCB Design has become more the norm than the exception. Even the once simple desktop PC - now produced in millions of units per year - has most of it's digital interfaces running above 50 MHz. Engineers transitioning into the High-Speed world need new design processes and methodologies to properly address the challenge. This is largely due to the fact that yesterday's PCB netlist describes only logical connectivity and not electrical performance criteria or physical requirements. This paper describes a newly developed High-Speed Design Methodology focused on automating the derivation and passing of all relevant constraints required for a successful end product. Process steps are illustrated by examples from state-of-the-art Intel PC Motherboard system design challenges. Through moving analysis up-front in the development cycle, tangible savings in cost and time are demonstrated.



Rodger Perger
Intel Corporation
5200 N.E. Elam Young Parkway
Hillsboro, OR 97124-6497
Rodger_R_Perger@ccm2.hf.intel.com
503-696-2363

Donald Telian
Cadence Design Systems
2655 Seely Road, MS 8A1
San Jose, CA 95134
donaldt@cadence.com
408-944-7791

Current Activities

Rodger Perger is currently a Design Engineer for Intel OPSD, designing motherboards around advanced Intel Architecture technology.

Author Background

Rodger has worked for Intel for 2 years after graduating from the University of Minnesota. During that time he has been involved in providing simulation data for the SODIMM specification. He has also worked on simulating and designing AGP addin cards. Most recently he has been working to implement the new Constraint-Driven Methodology. This has involved testing out the methodology and then training other engineers to use the new methodology described in this paper.

Current Activities

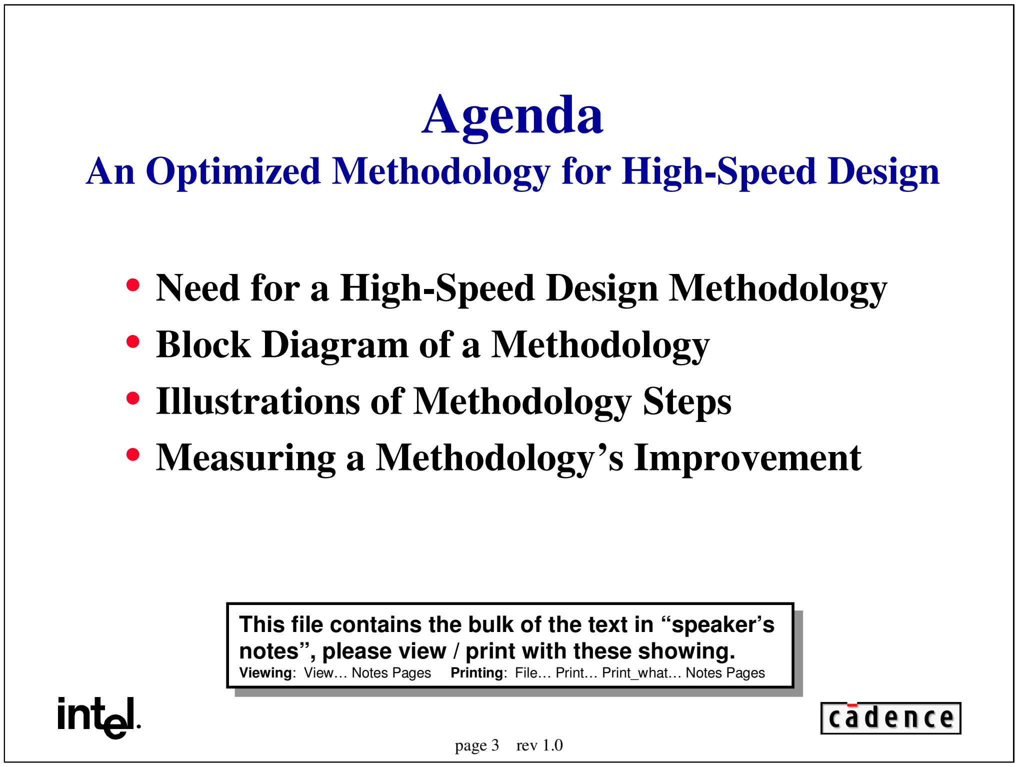
Donald Telian is currently a consultant for Cadence Design Systems' Spectrum Services, solving high-performance design problems for Cadence customers world-wide.

Author Background

Donald has been involved in high-speed design for over 11 years. Most recently he served as a technical coordinator between Intel and Cadence, helping teams from both companies realize their shared vision of developing the integrated high-speed design methodology described in this paper. Prior to that, Donald worked at Intel Corporation where he founded and managed the engineering group that resolved high-speed design issues for 10 Intel Architecture desktop platforms for 486, Pentium(R), and PentiumPro processor-based systems. He also led the design/validation of the PCI Bus 5V electrical specification, co-wrote the original IBIS and RAIL specifications, and founded the IBIS Open Forum.

This paper was co-authored by Rodger Perger of Intel and Donald Telian at Cadence. The Methodology itself was developed by teams of engineers from both companies.

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.



Agenda

An Optimized Methodology for High-Speed Design

- Need for a High-Speed Design Methodology
- Block Diagram of a Methodology
- Illustrations of Methodology Steps
- Measuring a Methodology’s Improvement

This file contains the bulk of the text in “speaker’s notes”, please view / print with these showing.
Viewing: View… Notes Pages **Printing**: File… Print… Print_what… Notes Pages

This is a paper about the changing process of digital design. Higher speed signals and interfaces have caused the need for some new methods and design techniques.

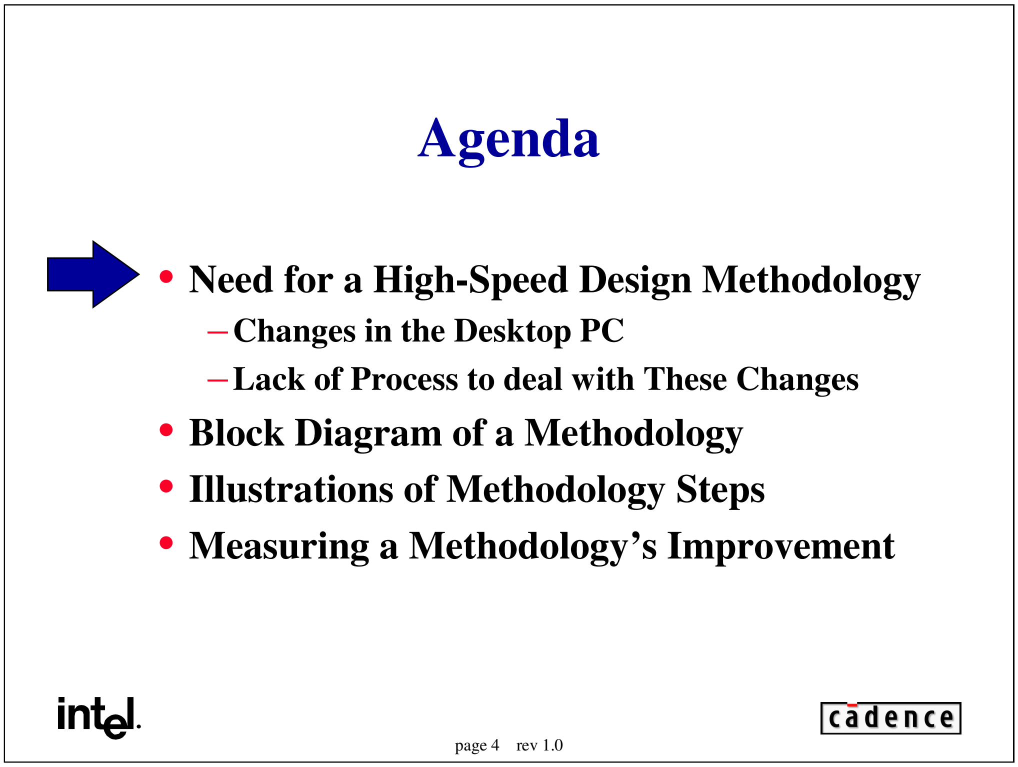
“High-Speed” has carried with it a sense of mystery, creating problems meant only for “experts” to solve. However, as higher speed Integrated Circuits (ICs) have become widely used in systems of all types, it has become important to derive a process by which the issues can be understood, addressed, and solved by every digital design engineer during the course of their normal duties. THAT is what this paper is about: describing an optimal process or *methodology* that can be applied by any engineer to solve the high-speed issues associated with today’s typical designs.

To do that, we’ll begin by showing how each of the interfaces in even a desktop PC now presents a high-speed design challenge. This change alone has made the problem more widespread, and brought about **the need for a high-speed design methodology** all design engineers can easily use. The problems can not only be solved by “experts”. There simply aren’t enough of them to go around.

Next, we’ll show the **block diagram** of an optimized methodology. We say it’s “optimized” primarily because it causes the high-speed issues to be addressed and solved early in the design cycle before they delay production schedules and become more difficult and costly to solve.

To better **illustrate the methodology steps** in the block diagram, we will then show examples of exactly how the methodology was used on state-of-the-art Intel motherboard designs.

And finally we will offer some metrics aimed at **measuring the methodology’s improvement** (savings in time and cost) once it is deployed in a mainstream hardware development process.



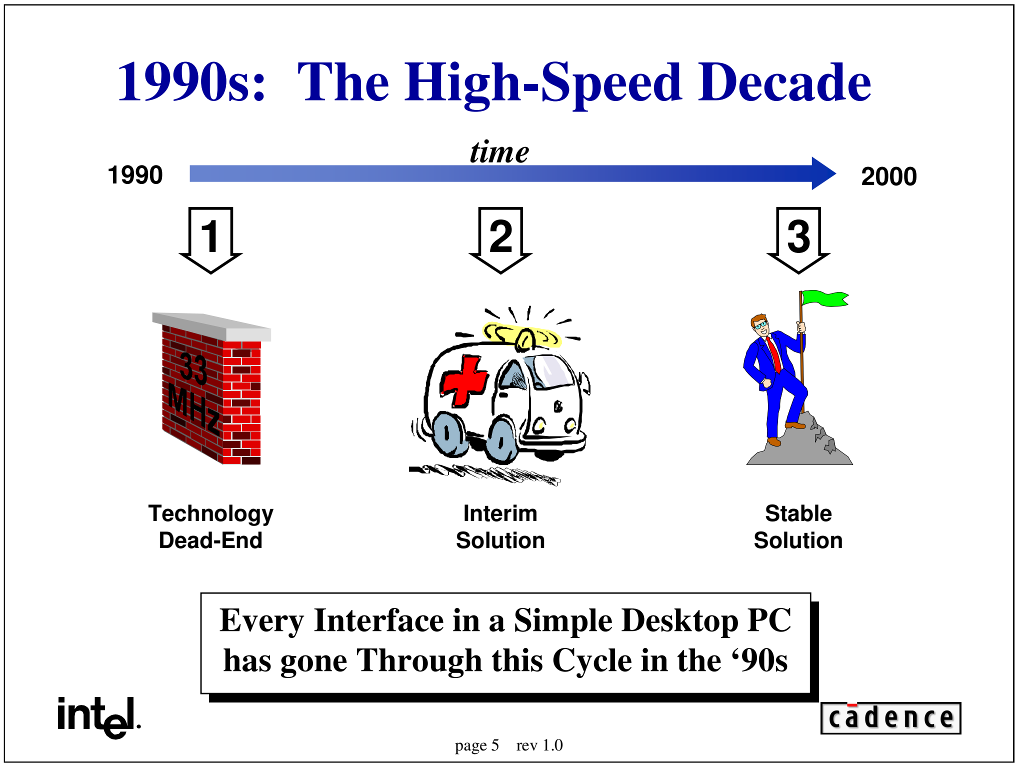
Agenda

- Need for a High-Speed Design Methodology
  - Changes in the Desktop PC
  - Lack of Process to deal with These Changes
- Block Diagram of a Methodology
- Illustrations of Methodology Steps
- Measuring a Methodology’s Improvement

page 4 rev 1.0

As we move through this paper, keep in mind that we will be focusing on superimposing a “high-speed” methodology onto a typical schematic/layout oriented digital design process. As such, familiarization with the typical process is assumed and not addressed here. Instead, we will show how the high-speed process steps interact with the typical process used by most companies today.

We begin here by looking at why, at this point in time, it has become imperative to have a high-speed design methodology.



1990s: The High-Speed Decade

time
1990
2000
1
2
3
33 MHz

Technology Dead-End

Interim Solution

Stable Solution

Every Interface in a Simple Desktop PC has gone Through this Cycle in the '90s

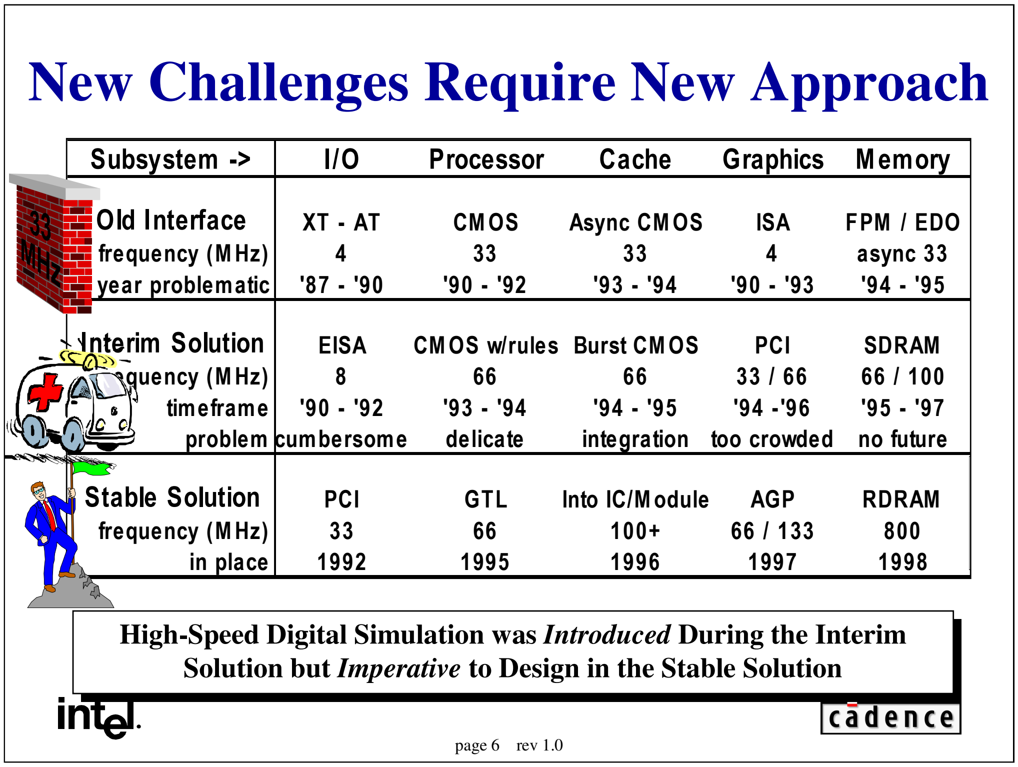
The 1990s have literally been the "High-Speed Decade". The simple desktop PC, in its quest for higher performance, multimedia extensions, TV-like video, and arcade-style 3D graphics, has been required to raise the frequency of all of its internal busses.

If you study the path that each interface has taken, you'll see three basic steps. Originally, the interface operated at *lower speeds*. Functionality was achieved by integrating the basic logical functions into a set of ICs, and then connecting them together however you pleased. As the quest for higher performance ensued, the frequencies on these lower speed interfaces approached 33 MHz.

At 33 MHz a **Technology Dead End** occurred, where it was no longer possible to simply continue to raise the frequency on the same set of physical wires and technology. As such, 33 MHz has become the de facto standard on when/where digital signals (on FR4 PCB) change from "low-speed" to "high-speed". (Actually, there are other phenomena at work here such as rise/fall times, but that's another discussion).

Since market pressures really don't care about technology dead ends, some **Interim Solution** had to be found to allow the interfaces to continue to increase in performance. For these solutions, since the frequencies pushed past 33 MHz, high-speed simulation was normally introduced to explore ways to mesh the older interfaces with the future technologies.

These simulations (and a lot of hard work) would then converge on a more **Stable Solution** of which high-speed design was an integral part.



New Challenges Require New Approach

Subsystem ->	I/O	Processor	Cache	Graphics	Memory
Old Interface	XT - AT	CMOS	Async CMOS	ISA	FPM / EDO
frequency (MHz)	4	33	33	4	async 33
year problematic	'87 - '90	'90 - '92	'93 - '94	'90 - '93	'94 - '95
Interim Solution	EISA	CMOS w/rules	Burst CMOS	PCI	SDRAM
frequency (MHz)	8	66	66	33 / 66	66 / 100
timeframe	'90 - '92	'93 - '94	'94 - '95	'94 -'96	'95 - '97
problem	cumbersome	delicate	integration	too crowded	no future
Stable Solution	PCI	GTL	Into IC/Module	AGP	RDRAM
frequency (MHz)	33	66	100+	66 / 133	800
in place	1992	1995	1996	1997	1998

High-Speed Digital Simulation was *Introduced* During the Interim Solution but *Imperative* to Design in the Stable Solution

page 6 rev 1.0

To better illustrate exactly how this has occurred in a desktop PC, consider this chart carefully.

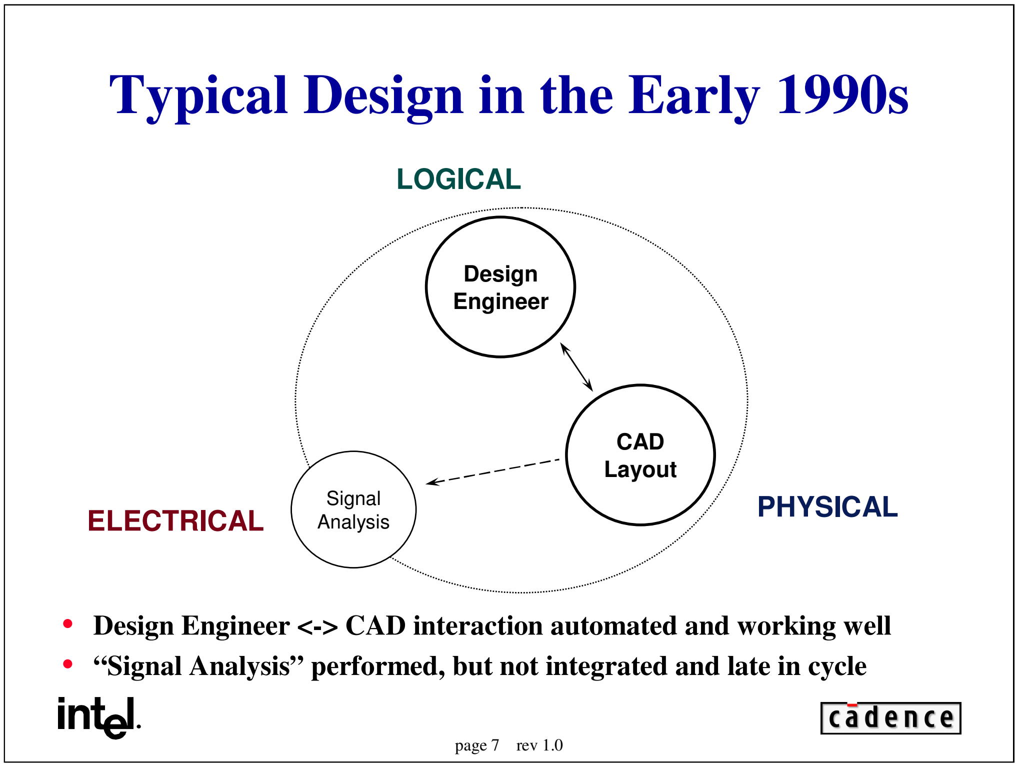
Notice (in the first row) how each of the older interfaces, and even the 386 and 486, topped out in the early 1990s at 33 MHz or less.

Interim Solutions emerged in the mid-90s at higher frequencies to help performance continue to rise, but eventually gave place to the (even higher frequency) Stable Solutions primarily due to the problems stated.

But even more interesting is looking at how high-speed design and simulation was used to arrive at both the Interim and Stable solutions. The Interim solutions *were* derived by “experts” on very advanced, but less user friendly tools using a process that was kind of “make it up as you go.” In contrast, the simulations of the Stable solutions are now being performed by engineers everywhere on tools supplied by leading EDA vendors that are much easier to use. And the process used is becoming more crystallized into a common Methodology - largely what we are attempting to illustrate in this paper.

Converging on an understandable and repeatable methodology will be the primary focus of high-speed design during this latter portion of the 1990s.

So that is how technology has migrated to higher performance. Now let’s look at how design processes have changed in the 1990s to accommodate these changes.



Typical Design in the Early 1990s

LOGICAL

Design Engineer

CAD Layout

Signal Analysis

ELECTRICAL

PHYSICAL

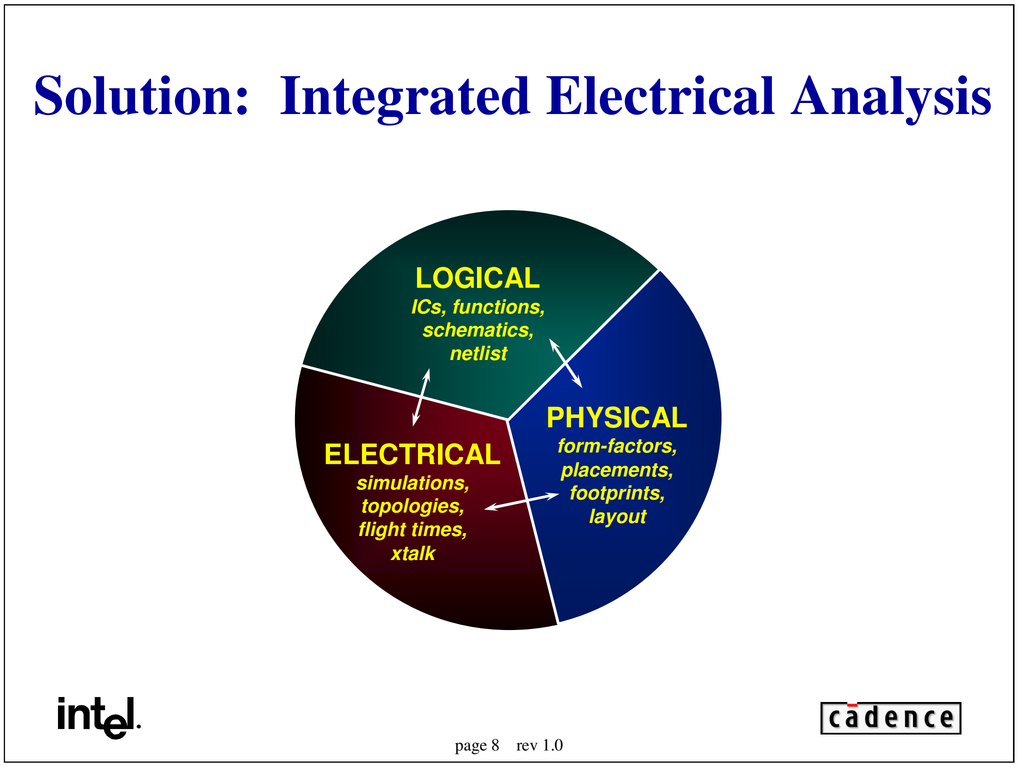
- Design Engineer <-> CAD interaction automated and working well
- “Signal Analysis” performed, but not integrated and late in cycle

page 7 rev 1.0

Entering the 1990s, the logical and physical portions of digital design worked fairly well. The LOGICAL details of the design were primarily driven by the Design Engineer, who captured the design content in a Schematic. Connectivity was then compiled into a “netlist” and electronically transferred into CAD Layout. In CAD, the printed circuit board’s PHYSICAL implementation was drawn using a standard PCB Layout environment. The back-and-forth sharing of electronic information between Design Engineer and CAD Layout worked well as forward and backward annotation were used to keep the databases (Schematic and Layout) synchronized.

Signal Analysis (the predecessor of today’s “High-Speed Design”) was performed in a variety of ways, mainly by carefully measuring each signal’s performance on physical hardware. In addition, completed layouts were electronically translated into various simulation environments to try to analyze their performance prior to physical hardware. Both of these attempts were effective at finding problems, but were too late in the design cycle to adequately *influence* the design and layout *while it was being developed*. These methods proved to be too slow and cumbersome because they caused too many physical design iterations, and were very hard to carry out due to the growing number of nets requiring attention. Consequently, an increasing amount of “up front” analysis was done to try to catch problems earlier in the design cycle and define a “solution space” in which design and layout could achieve first pass success. However, no good (electronic) “integrated” mechanism existed to capture and communicate this solution space back to the rest of the design team, and information was often mis-communicated, not followed, lost, and/or unavailable to re-use on later designs. As shown in the figure, information only electronically flowed one way : from CAD layout TO signal analysis.

Obviously, a more optimized process would *integrate* the ELECTRICAL portion of the design into the larger design process to achieve “Constraint Driven Design” (I.e., passing the electrical rules FROM Signal Analysis TO CAD to guide or “constrain” the design as it is developed), better first pass success, and hence faster time-to-market.



Solution: Integrated Electrical Analysis

LOGICAL
ICs, functions, schematics, netlist

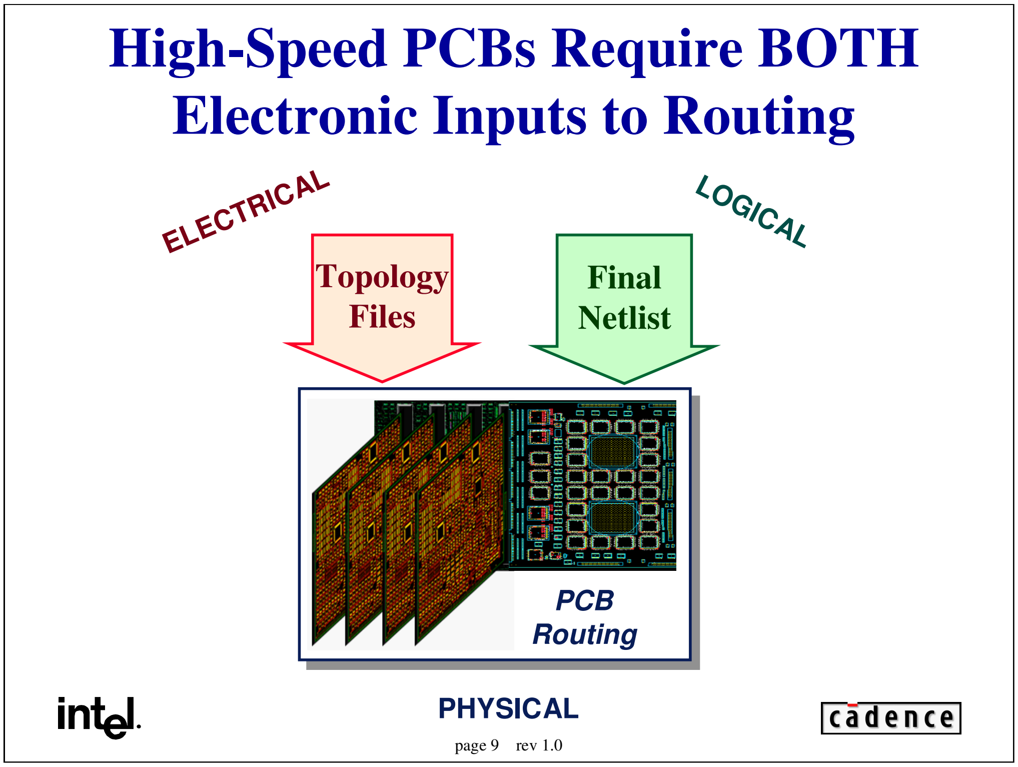
PHYSICAL
form-factors, placements, footprints, layout

ELECTRICAL
simulations, topologies, flight times, xtalk

page 8 rev 1.0

As High-Speed Design becomes more critical in today’s mainstream designs, it too must be more tightly integrated into the hardware development process. An ideal solution would integrate the bilateral electronic sharing of design information between each of the disciplines as shown in the diagram.

While yesterday’s design success hinged on accurate passing of information between Logical and Physical domains (to/from schematic and layout), today’s high-speed boards require close coupling of the Electrical and Physical domains (shown in the lower-most arrow).



High-Speed PCBs Require BOTH Electronic Inputs to Routing

ELECTRICAL

LOGICAL

Topology Files

Final Netlist

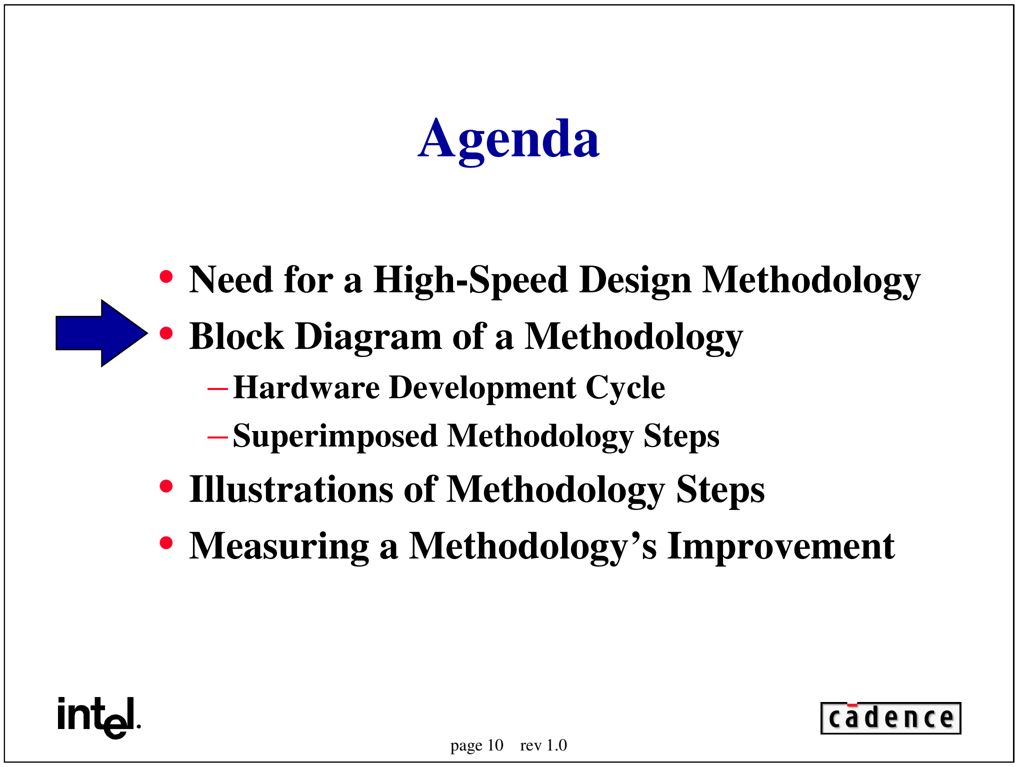
PCB Routing

PHYSICAL

page 9 rev 1.0

Said another way, the proper functioning of lower-speed boards hinged only on routing the connectivity expressed in the Final Netlist generated by the Logical design. But understand that logical connectivity expressed *what* was to be connected, not *how* the connection must be made (for example: in what order?, over what impedance?, with what acceptable delay?, etc.).

In contrast, today's high-speed boards require BOTH the connectivity requirements expressed in the logical netlist AND ALSO the electrical and topological information (shown here in "Topology Files"). This is the primary change caused by higher speed devices: to successfully route a high-speed board both sets of information must be fed into the physical layout process. Actually, that's been fairly obvious for awhile, but the fact that now *most of the board* has high-speed ICs and nets has made the automation of this task imperative. As such, this paper will focus on exactly how this can be done.

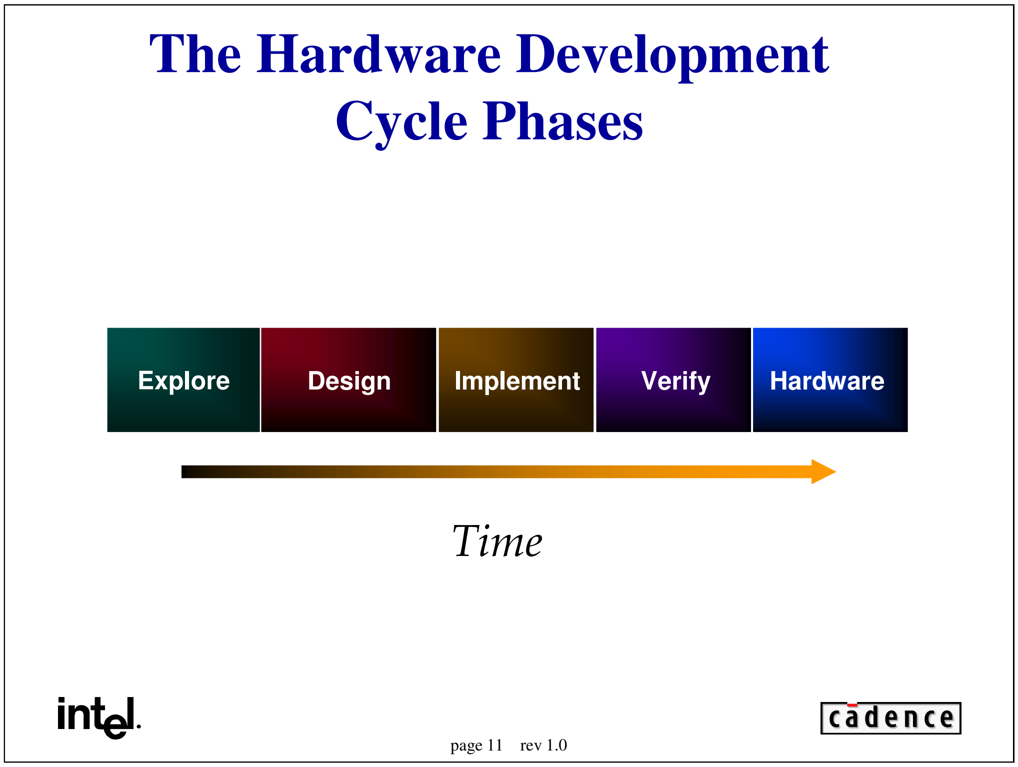


Agenda

- Need for a High-Speed Design Methodology
- Block Diagram of a Methodology
  - Hardware Development Cycle
  - Superimposed Methodology Steps
- Illustrations of Methodology Steps
- Measuring a Methodology's Improvement

page 10 rev 1.0

Now we have explained exactly why having a high-speed design methodology is imperative. Next we will show a block diagram picture of the Methodology, and show how it relates to a normal Hardware Development Cycle (shown first).



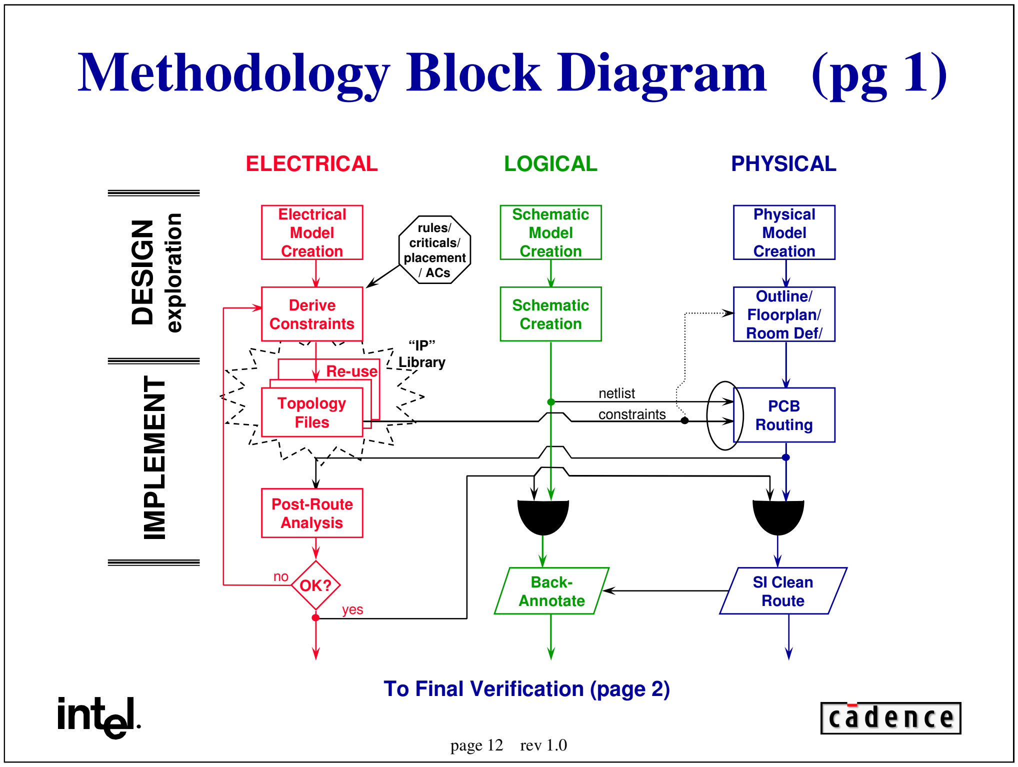
The Hardware Development Cycle Phases

To understand how the Methodology steps relate to product development, let's first look at the phases in a typical hardware development cycle.

Prior to developing a product it first must be well-defined. With a solid definition in hand, during the "**explore**" phase technology options and choices are examined and selected.

During the "**design**" phase these choices are carefully organized and analyzed, leading to a more physical "**implementation**" phase. During the implementation phase the design layout (and analysis of it) would occur. Once the design database is fully "**verified**", it can be realized in "**hardware**".

The Methodology on the next two pages shows how the Electrical, Logical, and Physical design tasks map onto these phases over time.



Methodology Block Diagram (pg 1)

ELECTRICAL

LOGICAL

PHYSICAL

DESIGN exploration

IMPLEMENT

Electrical Model Creation

rules/ criticals/ placement / ACs

Derive Constraints

"IP" Library

Re-use

Topology Files

Post-Route Analysis

no

OK?

yes

Schematic Model Creation

Schematic Creation

netlist

constraints

Back-Annotate

Physical Model Creation

Outline/ Floorplan/ Room Def/

PCB Routing

SI Clean Route

To Final Verification (page 2)

intel.

cadence

page 12 rev 1.0

This page and the next page contain the Block Diagram of the Methodology. The block diagram superimposes the hardware development phases (horizontally) onto the design domains (vertically, in colored columns). Each vertical column from left to right lists the various tasks performed for Electrical, Logical, and Physical design respectively.

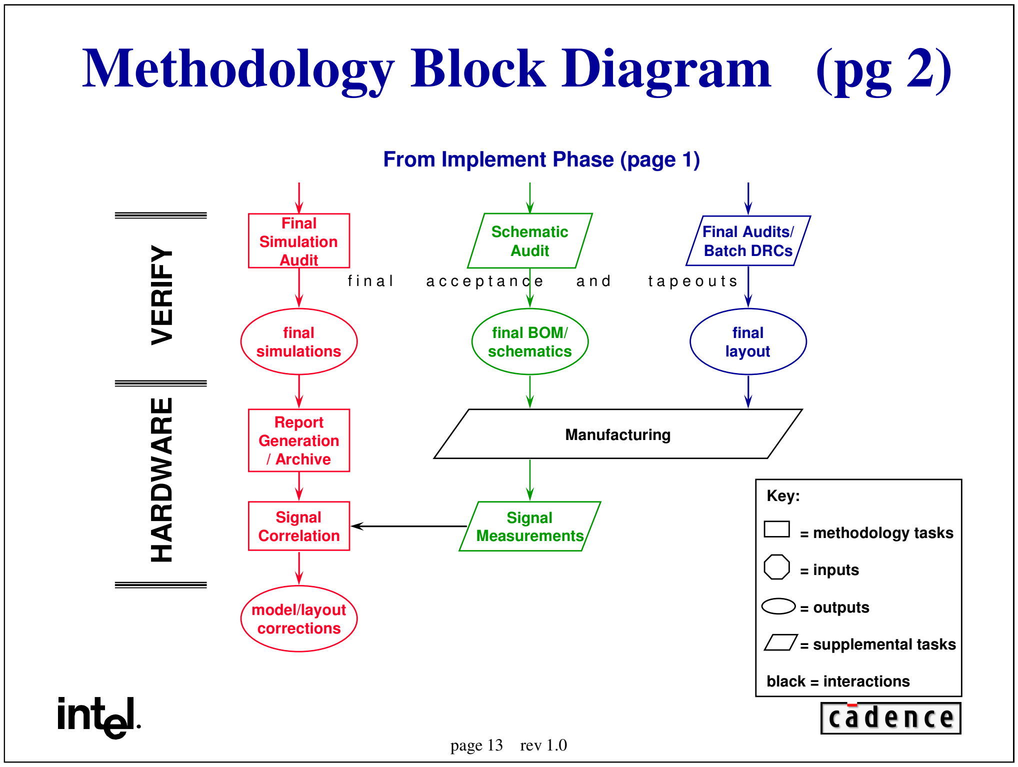
The hardware development flow progresses from top to bottom (and then onto the next page) showing the aspect of time as explained on the previous page. Some steps are further illustrated in the next section, and a simple explanation is given here.

ROW 1: The top row shows part of the design preparation. In each of the domains models (or symbols) must be created before the design can begin. Planning time for this stage, and ensuring it is done correctly and completely is key to success in the later stages.

ROW 2: With the model building blocks in place, (from left to right) constraints can be derived, schematics created, and a rough physical placement performed. For many designs, there would be more Electrical interaction and constraints involved in Floorplanning. However, in PC motherboard design most of the floorplan is pre-determined by standard chassis form factors.

ROW 3: As already stated, a key component of a Methodology is the generation of Topology Files. The Cadence software environment provides convenient graphical interfaces to draw and simulate topologies, sweep various options, enter desired constraints, and then use a "save" button to create the Topology Files. The files should be organized into a library that will capture this work in an electronic and re-usable format. Over time, these constraint files will become a key component of a design team's Intellectual Property (or "IP"). In row 3 the topology files are applied to the board layout. PCB Routing can now be performed as guided by both ELECTRICAL constraints (contained in topology files) and LOGICAL connectivity (contained in the netlist).

ROW 4: After the board is routed, "Post-Route Analysis" simulations are performed to confirm that the route did indeed satisfy the previously entered constraints. This analysis is shown as gating the progress of both the schematic and PCB layout to ensure that proper electrical performance is achieved on the first board fab.



Methodology Block Diagram (pg 2)

From Implement Phase (page 1)

VERIFY

HARDWARE

Final Simulation Audit

Schematic Audit

Final Audits/ Batch DRCs

final acceptance and tapeouts

final simulations

final BOM/ schematics

final layout

Report Generation / Archive

Manufacturing

Signal Correlation

Signal Measurements

model/layout corrections

Key:

▭ = methodology tasks

⬡ = inputs

⬭ = outputs

▱ = supplemental tasks

black = interactions

intel

cadence

page 13 rev 1.0

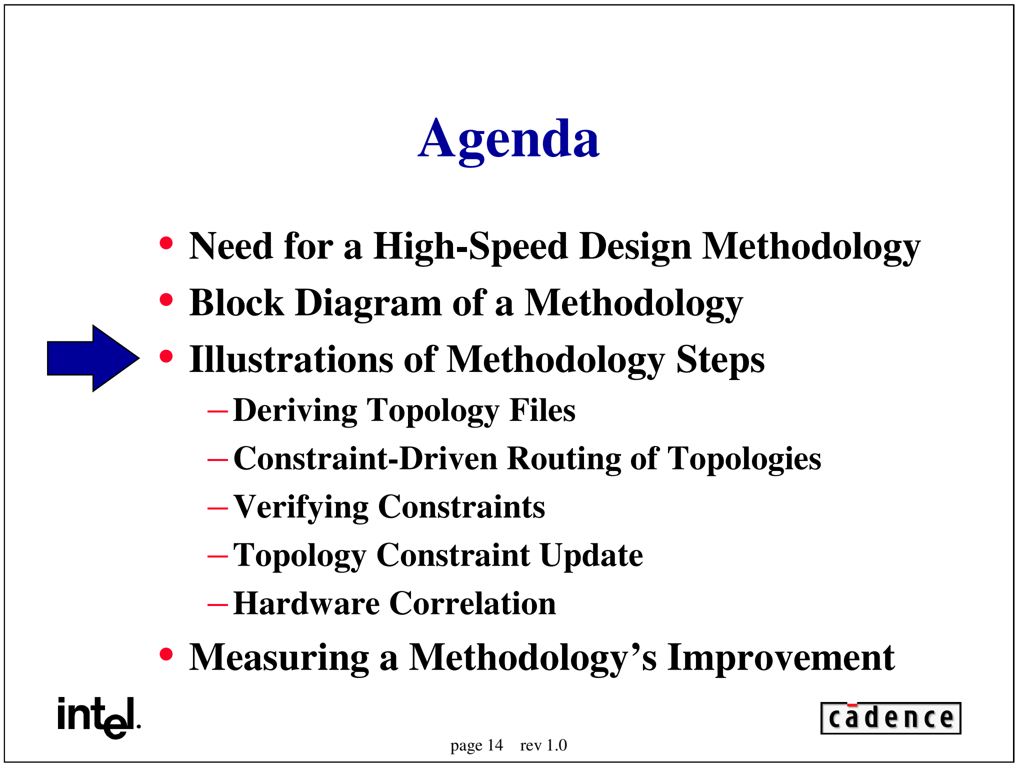
ROW 1: Once the implementation (and the careful analysis of it) is complete the design must be verified before committing it to hardware. Each column has a series of final audits to perform as indicated in row 1 on this page. Obviously, audits had occurred throughout design and implementation, but this row represents the one “last check” on adherence to company-dependent acceptance criteria.

ROW 2: Now final outputs are created. While Design and CAD assemble the various components of the manufacturing package (BOM, PCB artwork, assembly drawings, etc.), the Design Engineer can save off the final simulations for later correlation.

ROW 3: While manufacturing occurs the Design Engineer can archive the various simulation reports and setups. This is important, because if problems are found later in debug it will be important to know what the original assumptions were. A correlation testplan should also be written.

ROW 4: Once hardware is available, the test engineers should collaborate on physical signal verification and correlation, primarily on the most critical signals (or, others where you may want to verify the constraints derived with approximate models). Various methods exist to superimpose simulation waveforms on oscilloscope measurements to perform correlation.

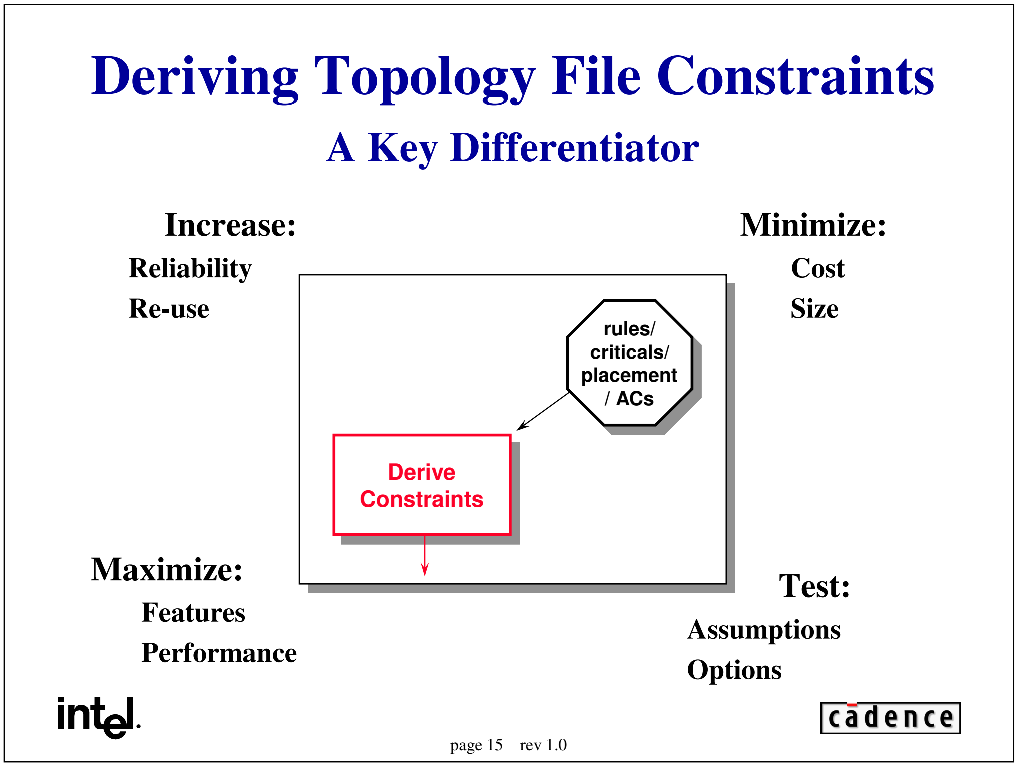
The final output (aside from a working board of course!) would be corrections to the measurements, models or layout as motivated by the correlation effort. This step is important to “close the loop”, improve all databases, and ensure there is sufficient margin in the design.



Agenda

- Need for a High-Speed Design Methodology
- Block Diagram of a Methodology
- Illustrations of Methodology Steps
  - Deriving Topology Files
  - Constraint-Driven Routing of Topologies
  - Verifying Constraints
  - Topology Constraint Update
  - Hardware Correlation
- Measuring a Methodology’s Improvement

Now that we have shown the overall Methodology steps, and how they relate to the standard development process, we will now illustrate some of the major points with some examples from a recent Pentium II Processor motherboard design. Our focus will be on the usage of topologies and electrical constraints within the design. Note that we will be highlighting some of the important blocks in the original Block Diagram shown on the last two pages.



Deriving Topology File Constraints

A Key Differentiator

Increase:

Reliability

Re-use

Minimize:

Cost

Size

rules/ criticals/ placement / ACs

Derive Constraints

Maximize:

Features

Performance

Test:

Assumptions

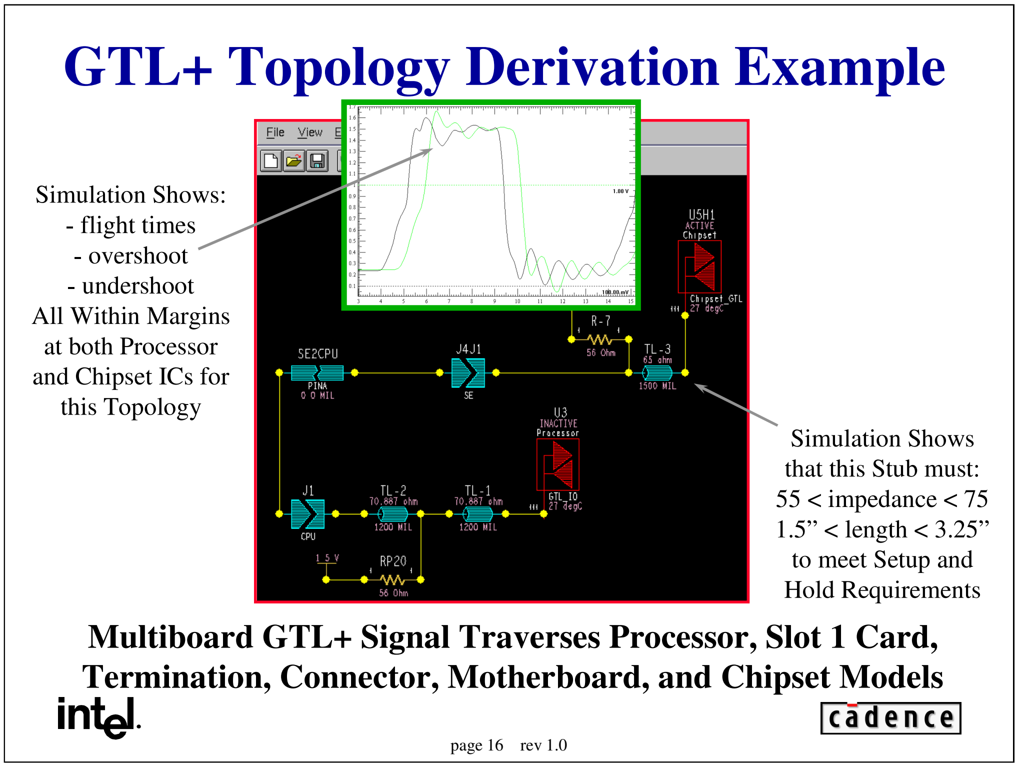
Options

page 15 rev 1.0

A key differentiator in any company's products and design flow is their ability to derive constraints that capture the electrical requirements of their state-of-the-art high-speed designs. These constraints should allow realization of minimal cost while achieving maximum design density and performance.

Some previously derived constraints may come in documentation associated with various processors, chipsets, or other high-speed components. Other constraints must be derived from scratch amidst various timing, buffer, loading, and placement requirements. Wherever constraints originate, each new design should test the assumptions under which they were previously derived and make sure they are applicable to the design at hand.

Remember that the output of the Derive Constraints process is to create Topology Files that will be applied to the design to guide the high-speed layout process. A single interface such as DRAM may require a number of topology files (and hence constraint derivations) to describe it (e.g., MAA.top, CAS_RAS.top, MD.top, and so on), while a single GTL.top file may work for all the signals in a processor interface.



GTL+ Topology Derivation Example

Simulation Shows:
- flight times
- overshoot
- undershoot
All Within Margins at both Processor and Chipset ICs for this Topology

Simulation Shows that this Stub must:
55 < impedance < 75
1.5” < length < 3.25”
to meet Setup and Hold Requirements

Multiboard GTL+ Signal Traverses Processor, Slot 1 Card, Termination, Connector, Motherboard, and Chipset Models

intel. cadence

page 16 rev 1.0

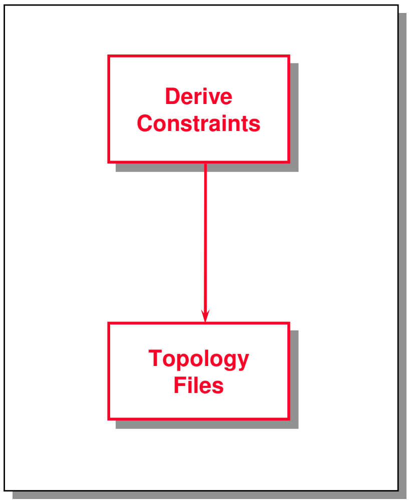
One of the critical steps to ensure that the methodology is successful is to derive the constraints that will drive the routing of critical nets. To successfully determine the constraints requires that several parameters must be varied.

These simulations should determine several things about the topology. Most importantly is how the net should be routed (e.g., Star or T topology). In this example we are most concerned with determining the minimum and maximum line lengths for each of the line segments required to meet the correct electrical performance and system timing parameters. By varying the buffer strength and the length, velocity, and impedance of each of the line segments a *solution space* can be determined. Waveforms should be checked for non-monotonic edges and flight time data should be gathered to determine setup and hold time violations.

Provided that a good simulation plan was followed the constraints that are derived should not be too prohibitive or too lax. It is important to find this middle ground when deriving the constraints so that the CAD engineer is not constantly running into DRC (Design Rule Check) errors or allowed to freely route critical nets without getting important DRC’s.

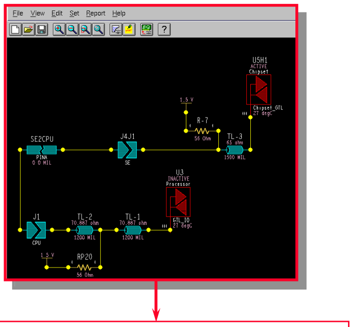
In this GTL+ example we carefully analyzed the hold time path requirements and simulated the net to determine the minimum acceptable line length. Overshoot, undershoot, and monotonicity can also be viewed to ensure that there are no violations. After viewing waveforms and gathering flight time data we have determined the constraints on the TL-3 line segment. This particular net segment has constraints of 55 to 75 Ohms for the impedance and has to be between 1.5” and 3.25” for line length. In a similar manner the constraints on the remaining T-lines can be determined.

Generating GTL+ Topology File



Derive Constraints

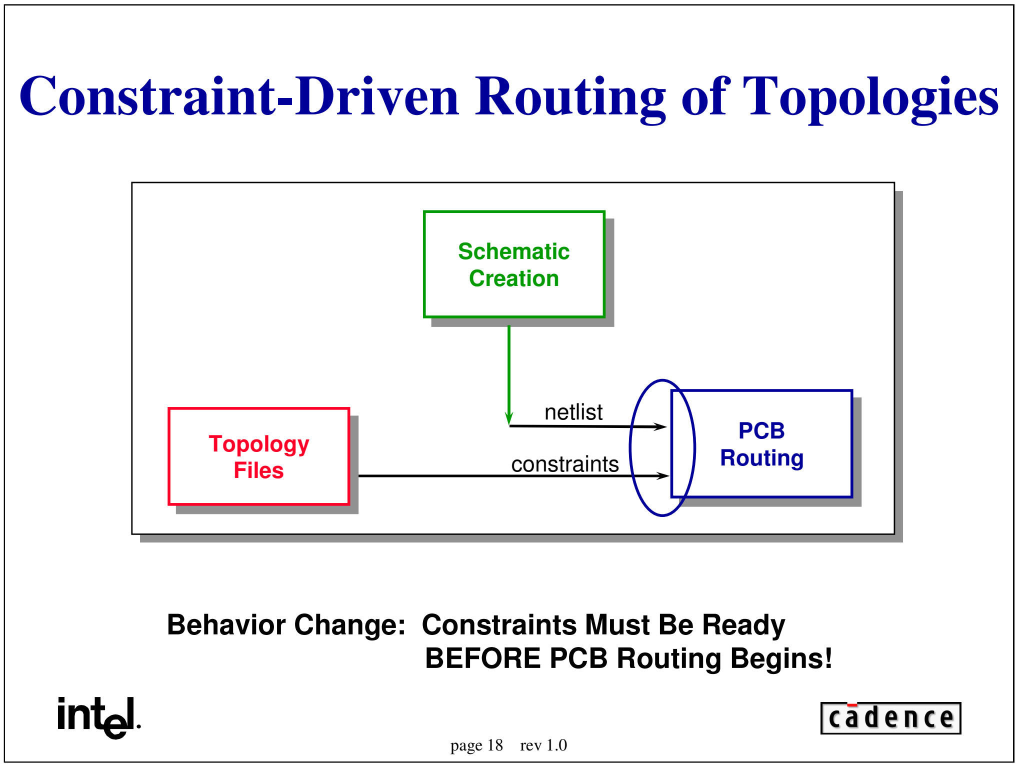
Topology Files



```
       (Nodes
         ("_idlSysNode1" "CPU.@@J1.B72.2" "pinB.72.1" )
         ("_idlSysNode2" "SE.@@J4J1.B72.2" "pinB.72.2" )
         ("SE.X77750Y29100L1LRATS"
"SE.T@@RATS@@U5H1.B22[illegible]J4J1.B72.1" "SE.@@U5H1.B22.1" )
       (Constraints
         ("MAX_UNDE[illegible]
         ("MAX_XTALK" [illegible]
         ("SWITCH_SETTLE_DEL[illegible]
           ("ALL DR[illegible]" "ALL RCVRS" [illegible]000e-009")
         ("MAX_OVERS[illegible]" "0.7")
         ("MAX_VIA_[illegible]")
       (Params
     (TlineGeometry ((di[illegible] 4.5[illegible] (conductorType
"stripLine")))
     (Trace ((length "0.01")))
     (HiClampTerm ((cutoffVoltage "0.7") (voltage "5")
(maxDelay "1e-10")))
     (IbisIO ((temperature "27")))
```

Just hit 'Save'!

Once the constraints for the nets have been determined the next step is to get the constraints into an electronic format. This will allow the constraints to be applied easily to the board file for routing. Having an electronic version of the constraints will prevent miscommunication between the design engineer and the CAD engineer. Also, these electronic constraints can be re-used and will prevent information from being lost. In our implementation, a file “save” of the derived constraint screen automatically generates a generic Topology File that can be applied to one or many nets in a design.



Constraint-Driven Routing of Topologies

Schematic Creation

Topology Files

netlist

constraints

PCB Routing

Behavior Change: Constraints Must Be Ready BEFORE PCB Routing Begins!

Once the Topology Files are in place, they must be applied to the design. This represents a BEHAVIORAL CHANGE, requiring the constraints to be ready BEFORE routing begins on the board.

Achieving this requires a pro-active approach:

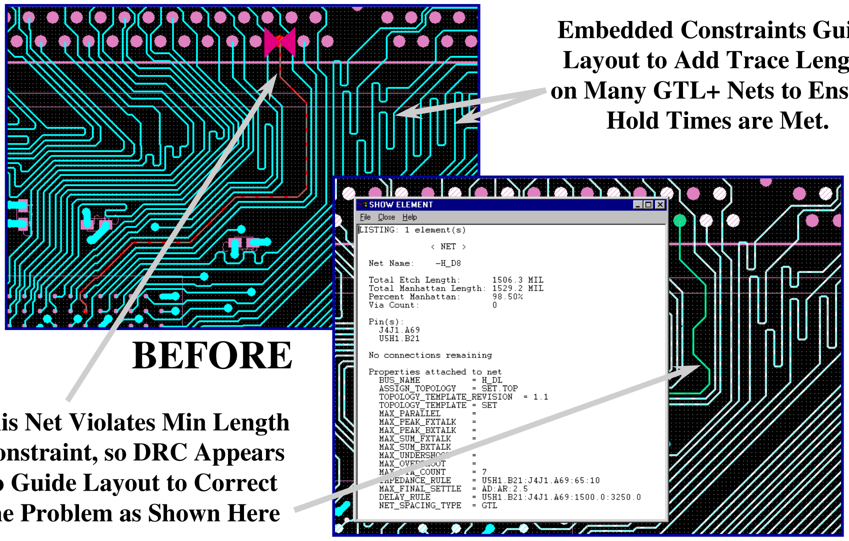
1) Very early in the design cycle the Design Engineer must determine all nets/net_groups that will need to have constraints defined.

2) All the required constraints must be created before the final netlist is generated.

3) The electronic constraints must be applied to all the critical nets/busses before layout begins. In the Cadence environment, this occurs automatically when the netlist is loaded by the layout software provided the topology files are available.

All this implies a different approach to project scheduling, primarily ensuring engineers are on the project early enough to derive all the constraints and apply them to the design prior to layout.

Constraint-Driven GTL+ Layout



Embedded Constraints Guide Layout to Add Trace Length on Many GTL+ Nets to Ensure Hold Times are Met.

BEFORE

This Net Violates Min Length Constraint, so DRC Appears to Guide Layout to Correct the Problem as Shown Here

```
LISTING: 1 element(s)

              < NET >

  Net Name:     -H_D8

  Total Etch Length:        1506.3 MIL
  Total Manhattan Length: 1529.2 MIL
  Percent Manhattan:        98.50%
  Via Count:                0

  Pin(s):
    J4J1.A69
    U5H1.B21

  No connections remaining

  Properties attached to net
    BUS_NAME          = H_DL
    ASSIGN_TOPOLOGY   = SET.TOP
    TOPOLOGY_TEMPLATE_REVISION  = 1.1
    TOPOLOGY_TEMPLATE = SET
    MAX_PARALLEL      =
    MAX_PEAK_FXTALK   =
    MAX_PEAK_BXTALK   =
    MAX_SUM_FXTALK    =
    MAX_SUM_BXTALK    =
    MAX_UNDERSHOOT    =
    MAX_OVERSHOOT     =
    MAX_VIA_COUNT     = 7
    IMPEDANCE_RULE    = U5H1.B21:J4J1.A69:65:10
    MAX_FINAL_SETTLE  = AD:AR:2.5
    DELAY_RULE        = U5H1.B21:J4J1.A69:1500.0:3250.0
    NET_SPACING_TYPE  = GTL
```

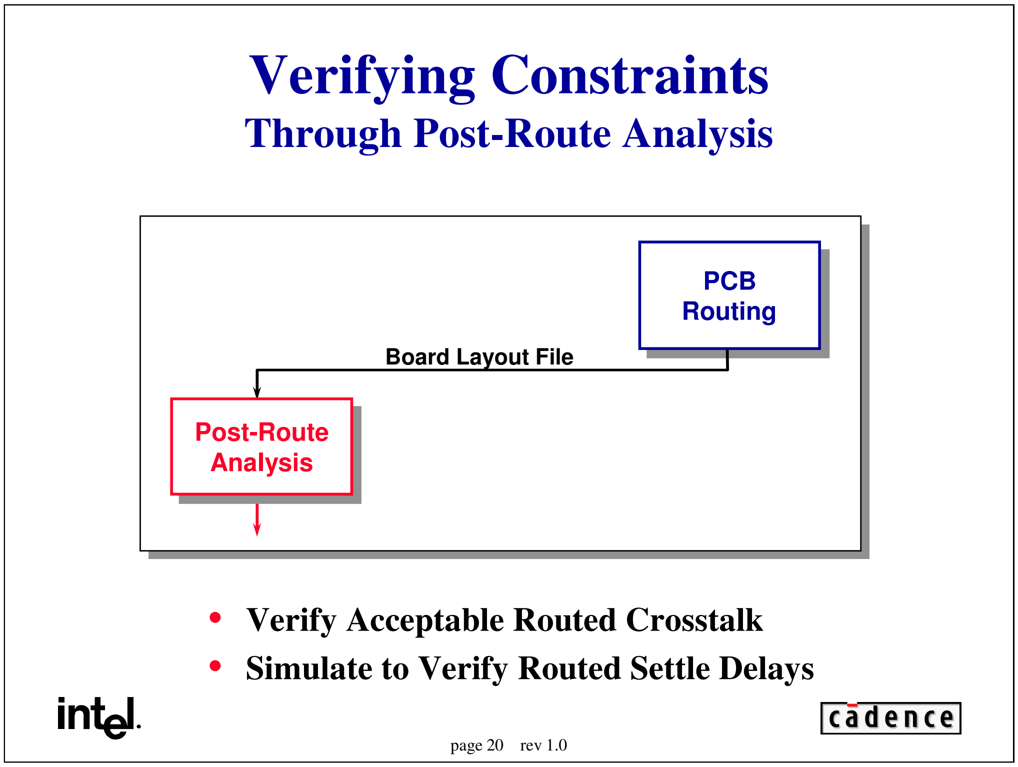
AFTER

intel. cadence

page 19 rev 1.0

Here is where the time and effort spent deriving the constraints pays off. As the CAD engineer is routing, a DRC will appear if a physical constraint is violated. In the example shown on the left the minimum line length constraint has been violated, so the net is modified to eliminate the DRC as shown on the right. Notice the serpentine on several other GTL+ nets to ensure that the minimum trace length constraint has been met.

A great time savings can be had by ensuring that all the physical constraints are met before the board enters post-route simulation. If the constraints were properly derived there should be very few instances requiring the topology files to be modified. This will prevent the need to waste time ripping up large sections of etch to re-route nets.



Verifying Constraints

Through Post-Route Analysis

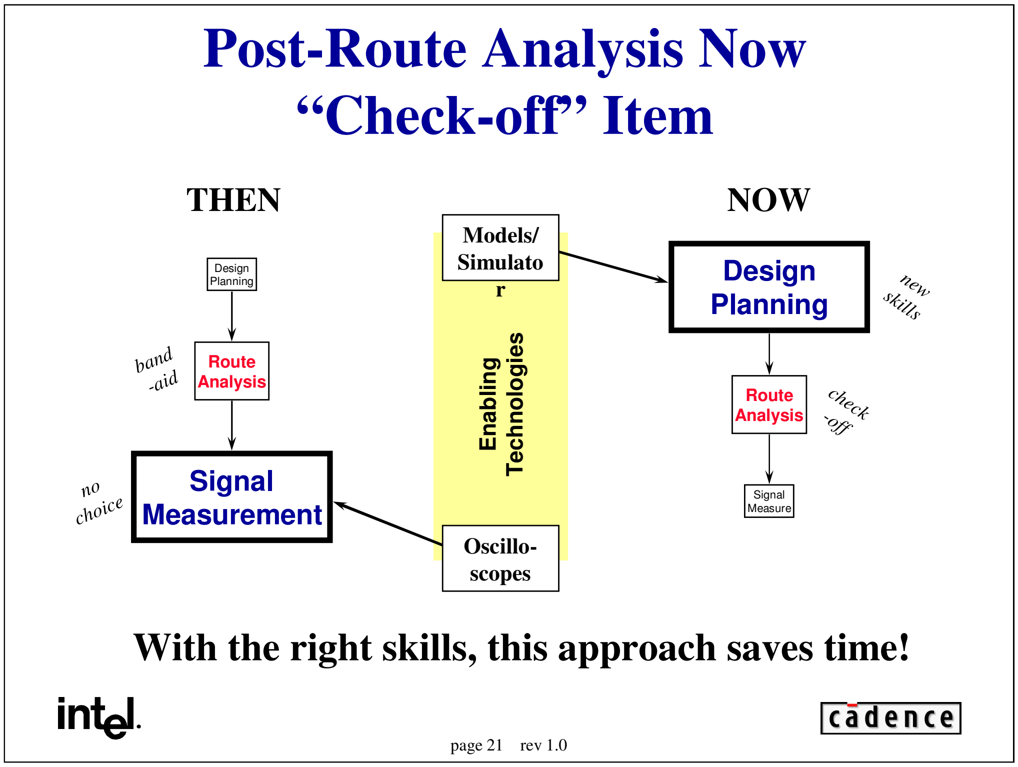
- Verify Acceptable Routed Crosstalk
- Simulate to Verify Routed Settle Delays

page 20 rev 1.0

Since the constraints have already been applied, during routing the CAD engineer sees DRCs when the layout-oriented constraints (such as min length or parallelism) are violated.

After various interfaces are routed, copies of the board file can be passed directly to the simulation engineer to perform “Post-Route Analysis” simulations. Note that since the constraints flowed INTO CAD and were applied, the board file already has the various simulation-oriented constraints (e.g., settle delays, overshoot limits, etc.) attached to the nets.

At this step, simulations are performed on the routed nets and the output report files are checked for various flags showing if and where parameters may have been violated. If the constraints were setup correctly *prior* to CAD layout, the number of violations found at this stage should be minimal.



Post-Route Analysis Now “Check-off” Item

THEN

Design Planning

band -aid

Route Analysis

no choice

Signal Measurement

Models/ Simulato r

Enabling Technologies

Oscillo- scopes

NOW

Design Planning

new skills

Route Analysis

check -off

Signal Measure

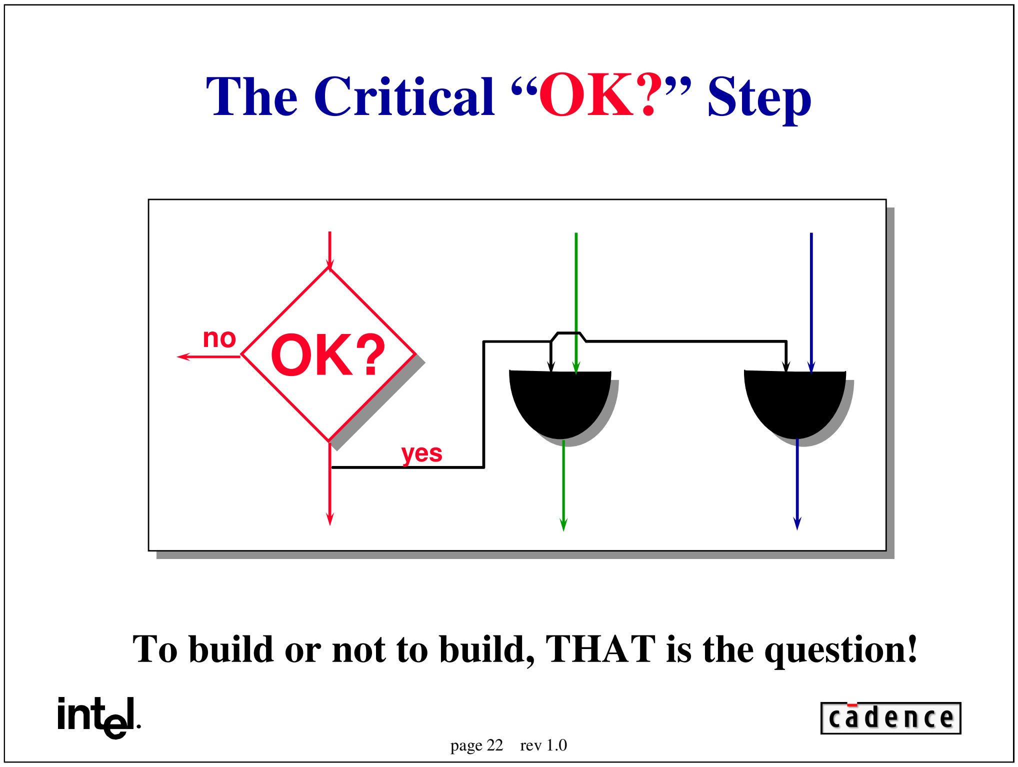
With the right skills, this approach saves time!

intel.

cadence

So we have basically flip-flopped our design approach to keep pace with the ever increasing need for faster technology. With less than 33MHz bus technology it was possible to route a board without concern that there would be serious signal quality problems. Problems that did exist were found with oscilloscopes and corrected.

Now with improved design tools and the need to become more efficient in the design process we have been able to shift our focus away from bench measurements to design planning. IBIS models and simulation tools allow the design engineer to effectively plan his design before the first trace is layed out on the board. This allows the post-route analysis to become more of a “constraint checker” than an evaluation process.



The Critical “OK?” Step

To build or not to build, THAT is the question!

And now we come to the moment of truth. IF the constraints were adequately defined AND the layout engineer successfully routed the PCB to the requirements, all the simulations pass with no problems.

HOWEVER, sometimes there are problems. If so, engineers will be asked to make a decision. Now is the time for ENGINEERING JUDGMENT. Neither the software nor the Methodology will make decisions for you. Instead, they will provide you with the tools, process, and data to make your decisions. And, your manager will provide you with the schedule. Alas, what will you do?

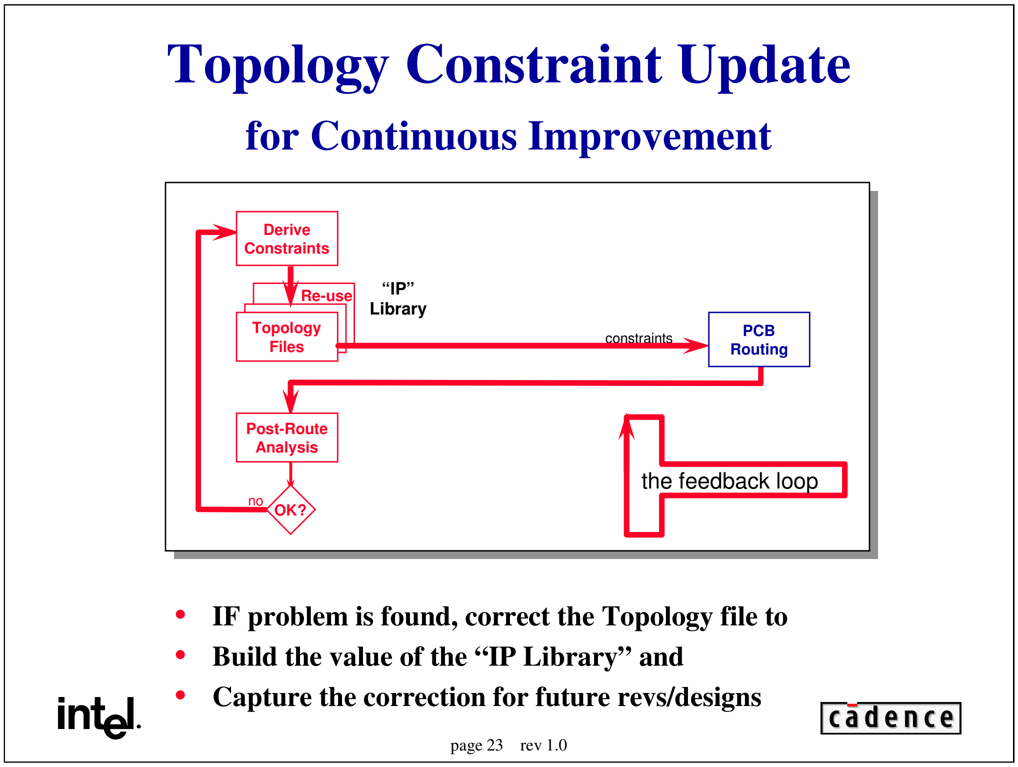
The max flight time is 3.2 ns, but this net is 3.41894 ns?

The overshoot limit on this SDRAM is 1.5V, but you’re at 1.67V worst case?

Tapeout awaits your decision. What will you do?

This may well be the most important part of the Methodology, and it is difficult to explain exactly what you should do. Different situations demand different responses. Now is the time to collect the data, gather the assumptions, and leverage the EXPERIENCE contained within your company to take the appropriate action.

If the answer to “OK?” is YES, the design proceeds through the gates. If the answer is NO, go to the next page.



Topology Constraint Update

for Continuous Improvement

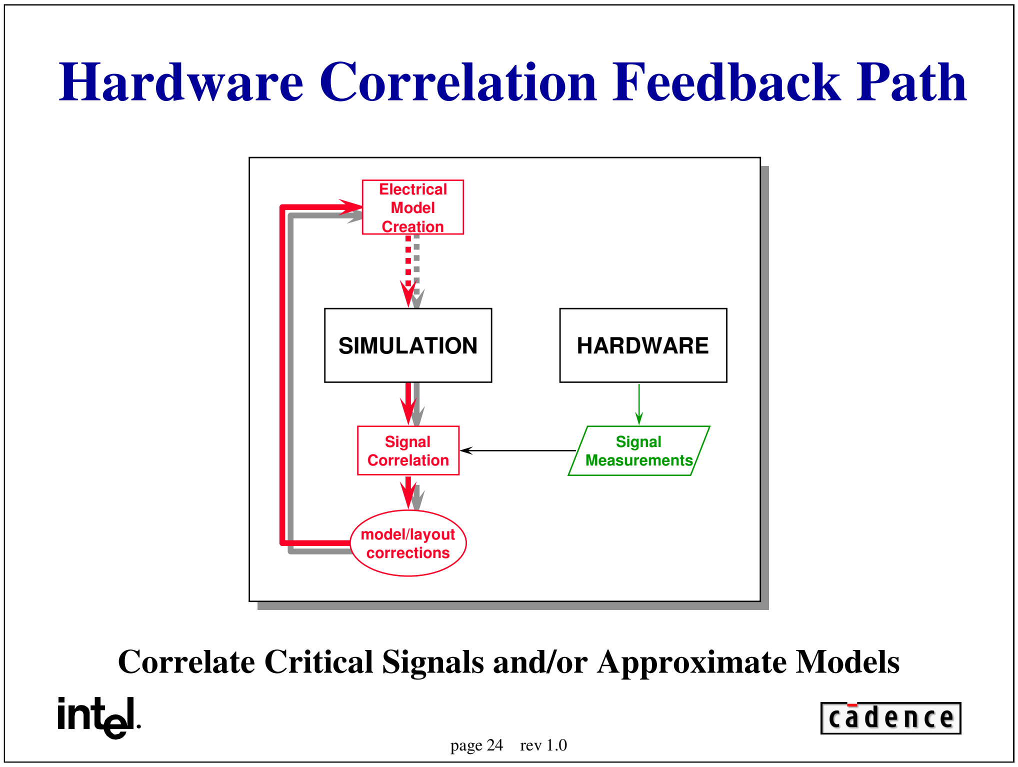
- IF problem is found, correct the Topology file to
- Build the value of the "IP Library" and
- Capture the correction for future revs/designs

Every good process has a feedback path that allows it to improve. This Methodology has such a path, as shown by the highlighted arrows (creating a loop).

If problems are found at the "OK?" step, it is important that changes are not directly made in CAD. If that occurred it may fix the design at hand, but the corrections would not be captured for later designs. This reusability of the constraints on future designs is an important part of this methodology. A great deal of time and money can be saved by reusing constraints and not having to redefine the same constraints for every new board project.

If there are failures in post_route these problems point to an inadequacy in the constraints and hence the original constraint derivation. As such, the proper action is to reload the original constraint file and re-adjust the constraints so that subsequent route attempts will not fail in Post-Route Analysis. This path is shown in the "no" arrow on the left in the diagram.

Note that since a loop has been created, CRAFTSMANSHIP must be exercised to minimize the number of loops while ensuring that the constraints are completely comprehensive in specifying the electrical requirements of the high-speed signals.



Hardware Correlation Feedback Path

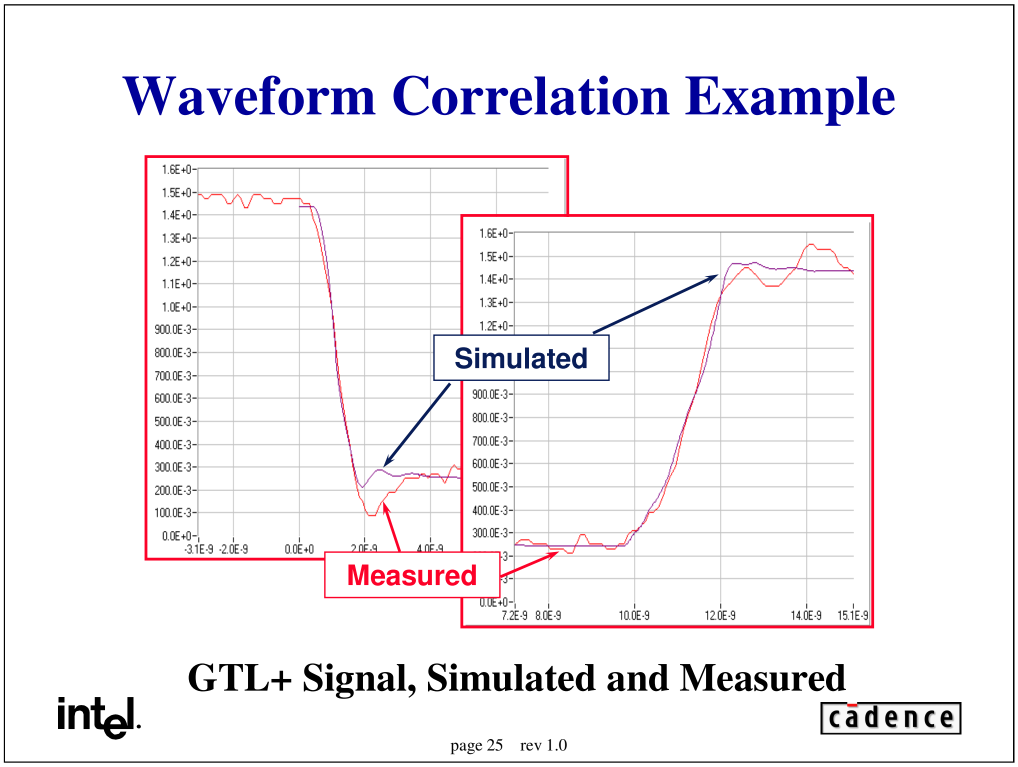
Correlate Critical Signals and/or Approximate Models

intel. cadence

page 24 rev 1.0

Another feedback path exists in the Methodology. This path flows from the very last step to the first.

Though it's very late in the development cycle, signal correlation yields excellent data that can be used to confirm Electrical Models. In extreme cases, the changes may influence the current design. But the primary goal is to verify the models and bring them to "correlated" revision levels (if they aren't already) for the benefit of future projects that will use them. This continues to raise the confidence in pre-hardware simulations, further decoupling development of a product from manufacturing/verification cycles.

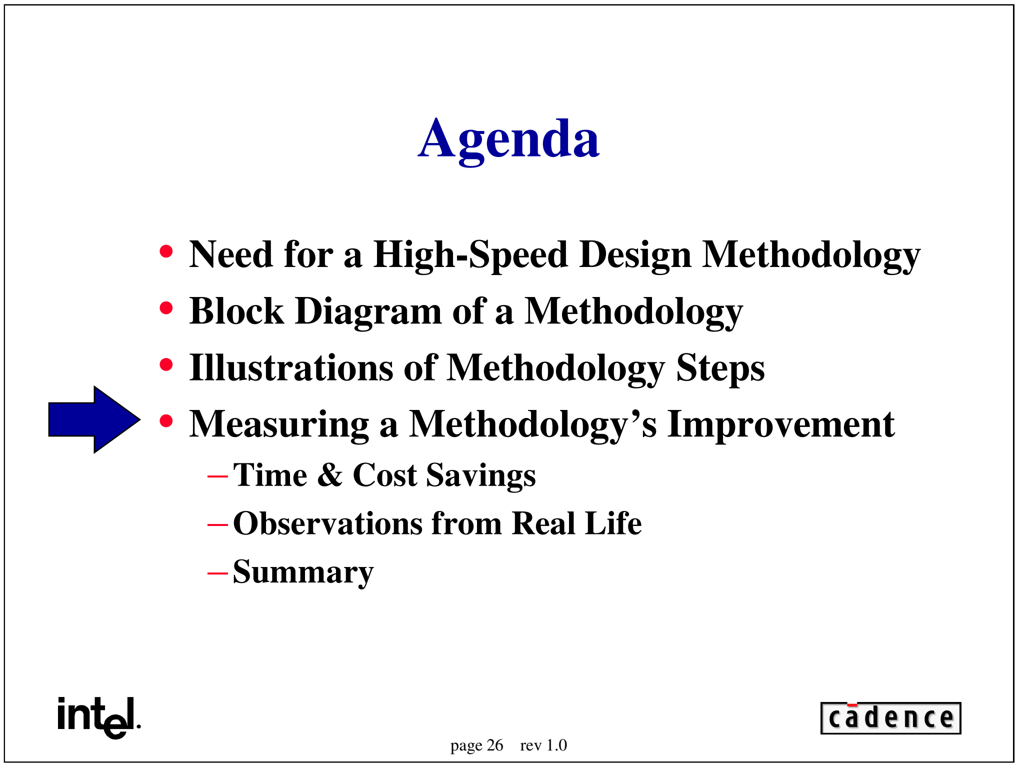


Waveform Correlation Example

GTL+ Signal, Simulated and Measured

page 25 rev 1.0

Correlation involves gathering and comparing waveforms measured on real hardware with those taken from the simulator. By comparing the waveforms, decisions can be made about the constraints defined in the topologies. For example, if the measured waveform showed a non-monotonic edge, it might be wise to re-simulate and tighten the line length constraint on one or more of the T-lines. This feedback path allows for models, topologies, and constraints to be updated to ensure that the next generation of boards are using the most accurate constraints possible.



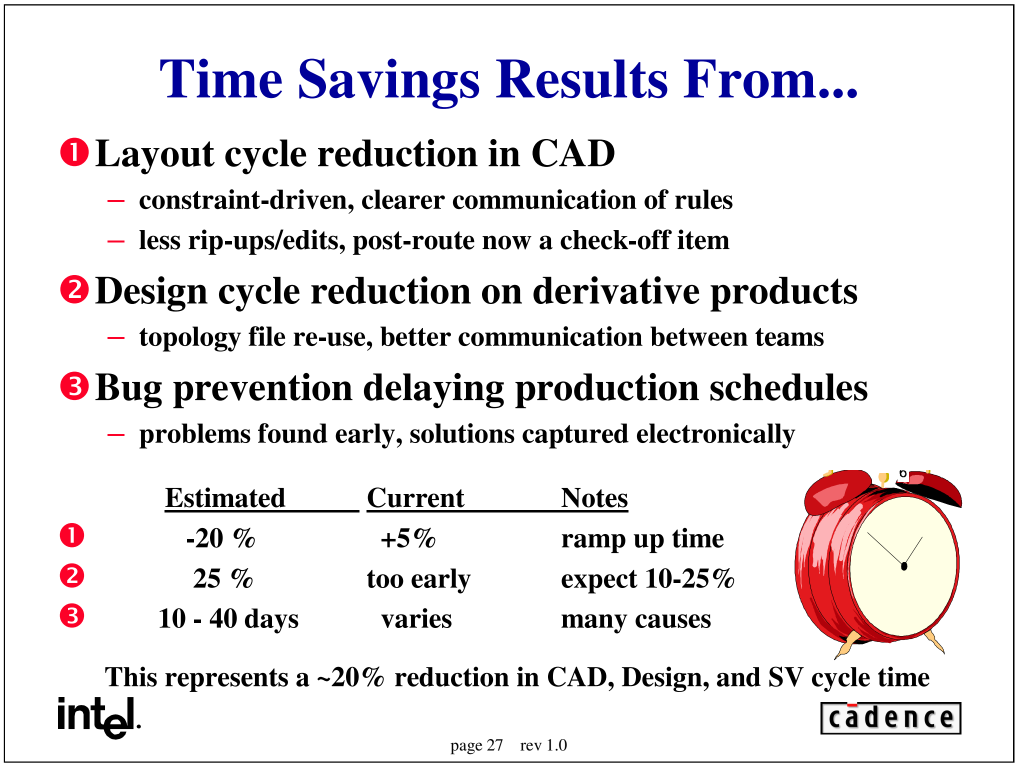
Agenda

- Need for a High-Speed Design Methodology
- Block Diagram of a Methodology
- Illustrations of Methodology Steps
- Measuring a Methodology’s Improvement
  - Time & Cost Savings
  - Observations from Real Life
  - Summary

intel.   cadence

page 26   rev 1.0

A good methodology should also include some ways to monitor and measure its success. So now that we have described the primary technical aspects of an optimized high-speed design methodology, we’ll take a look at some ways to measure its performance and improvement. We’ll quantify its benefits in both time and cost, and also share some key learnings gleaned from the early phases of deploying the methodology.



Time Savings Results From...

❶ Layout cycle reduction in CAD

- constraint-driven, clearer communication of rules
- less rip-ups/edits, post-route now a check-off item

❷ Design cycle reduction on derivative products

- topology file re-use, better communication between teams

❸ Bug prevention delaying production schedules

- problems found early, solutions captured electronically

	Estimated	Current	Notes
❶	-20 %	+5%	ramp up time
❷	25 %	too early	expect 10-25%
❸	10 - 40 days	varies	many causes

This represents a ~20% reduction in CAD, Design, and SV cycle time

page 27 rev 1.0

Though the Methodology is recently installed, we can begin to estimate and measure time and cost savings in several areas. The three primary areas of time savings affect the three main areas involved in the development of a new Motherboard: Design Engineering, CAD Layout, and System Validation. The three areas achieve these savings:

(1) Layout cycle reduction in CAD. Because constraints are now automated and passed in electronically, there is less time required to explain exactly how the routes must be made. Also, since topologies are pre-analyzed and pre-defined for success there are much fewer changes motivated by post-route analysis. That represents a huge savings on boards that are carefully routed manually since changes normally require ripping up large portions of the route. The estimated savings in the route process are:

1 week saved / 5 week process = 20% reduction in effort

(2) Design cycle reduction on derivative products. Traditionally, design teams have had a very difficult time capturing high-speed constraints and passing them on to other teams. Since topology files are saved electronically and built into a library, designers on derivative (similar, but changed slightly in features and implementation) products can either re-use them directly, or use them as a starting point to revise for their unique design. This capability saves time, estimated at:

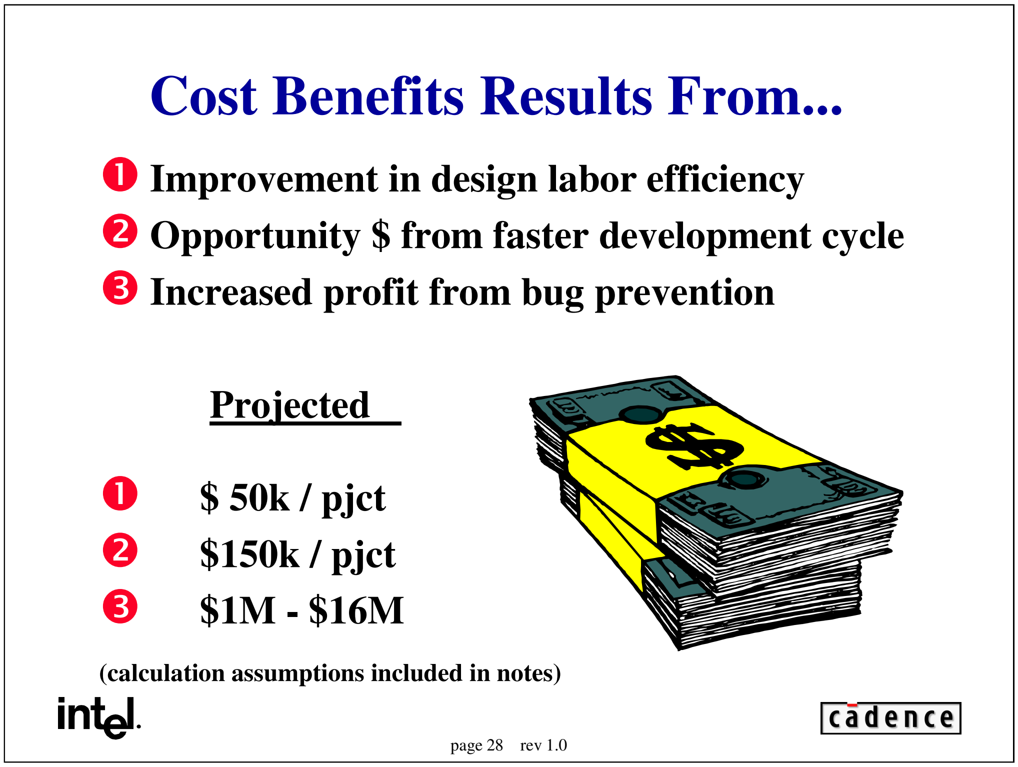
1 engr_month saved / 4 engr_months simulation process = 25% reduction in effort

(3) Bug prevention delaying production schedules. Depending on when they're discovered, high-speed issues often delay production while they are corrected. If they are discovered in System Validation (SV), they typically require a change in the PCB layout which takes at least 7 days. Fixing bugs has a number of steps, which have delays ranging between:

min = 10 days = 0.5 (find) + 0.5 (fix) + 1 (verify) + 7 (HW) + 1 (re-verify)

max = 40 days = 20 (find) + 5 (fix) + 3 (verify) + 7 (HW) + 5 (re-verify)

Since the methodology has impacted Design Engineering, CAD Layout, and System Validation, overall it should achieve a 20% reduction in the time required to execute these serial processes.



Cost Benefits Results From...

❶ Improvement in design labor efficiency

❷ Opportunity $ from faster development cycle

❸ Increased profit from bug prevention

Projected

❶ $ 50k / pjct

❷ $150k / pjct

❸ $1M - $16M

(calculation assumptions included in notes)

page 28 rev 1.0

Monetary value from using a High-Speed Design Methodology results from three primary areas, from which we can calculate an estimated dollar value.

(1) Improvement in design labor efficiency. As shown on the previous slide, installing a Methodology to solve high-speed design issues early in the hardware development cycle can result in roughly a 20% decrease in required resources to complete a project. Assuming a project requires 5 engineers for 6 months, at an expense of $100k (loaded) per engineer year, the savings per project is:

0.20/pjct * 5_engr * 6_mos * $100k/engr_year * 1year/12mos = $50k/pjct

Of even greater value (although not calculated here) would be the value of being able to complete 20% more product designs with the same amount of resources.

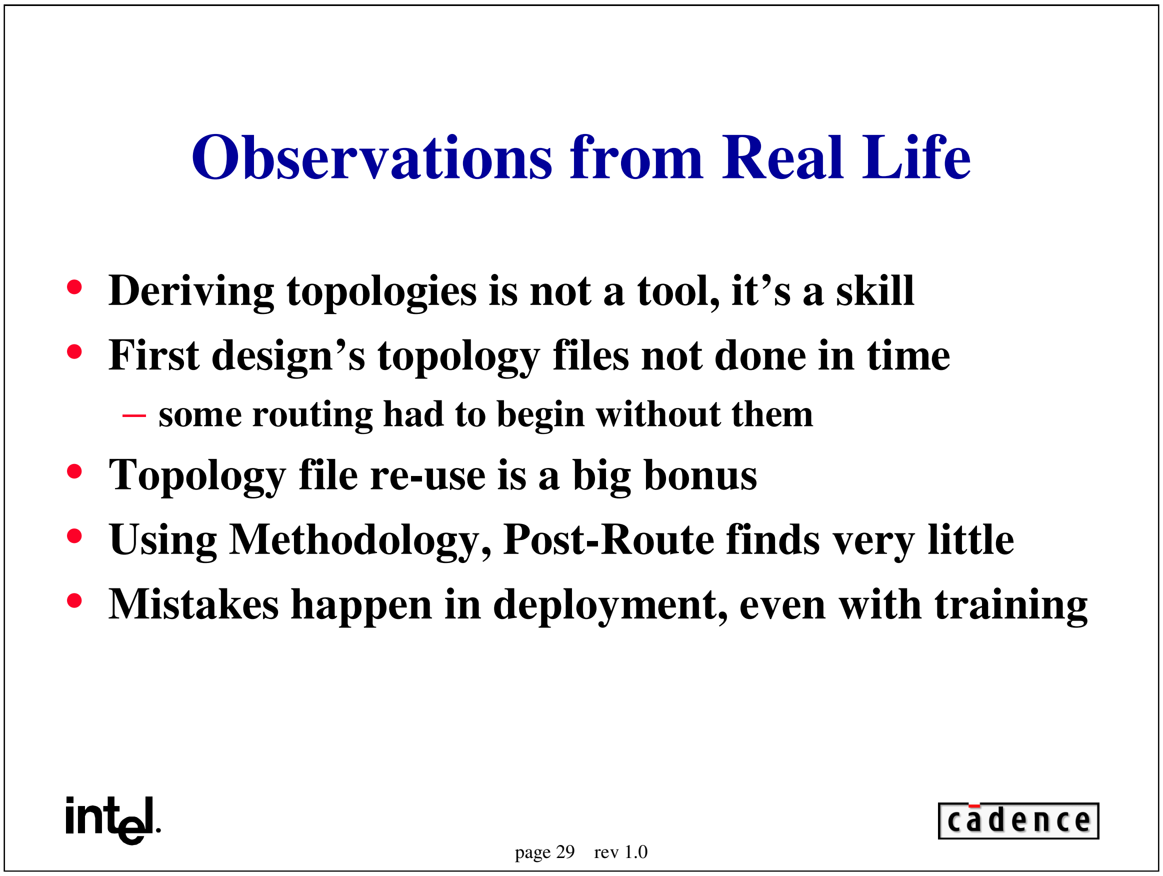
(2) Realized opportunity dollars from faster development cycle. PC Motherboards have a relatively short lifespan. Extra profit can be realized by releasing a product sooner than later. Assume we can get a product out 2 months faster by using the methodology to increase throughput and reduce bugs. If we assume that the profit erodes at a rate of $10/month, and production is 200k/month, the increased profit is:

$20_mo#1 * 200k/mos + $10_mo#2 * 200k/mos = $150k / project

(3) Increased profit from bug prevention. Any bug can cause a production slip if it misses the normal development/validation cycle, which often occurs with high-speed design problems. As shown on the previous slide, depending on the severity/elusiveness of the bug, it can take between 10 and 40 days to correct a hardware problem. With production ranging between 50k to 200k_units/month at $60_profit/unit, bugs can cost (neglecting NRE):

10 day_stall * 50k_units/month * $60/unit * month/30days = $1M minimum

40 day_stall * 200k_units/month * $60/unit * month/30days = $16M maximum



Observations from Real Life

- Deriving topologies is not a tool, it’s a skill
- First design’s topology files not done in time
  - some routing had to begin without them
- Topology file re-use is a big bonus
- Using Methodology, Post-Route finds very little
- Mistakes happen in deployment, even with training

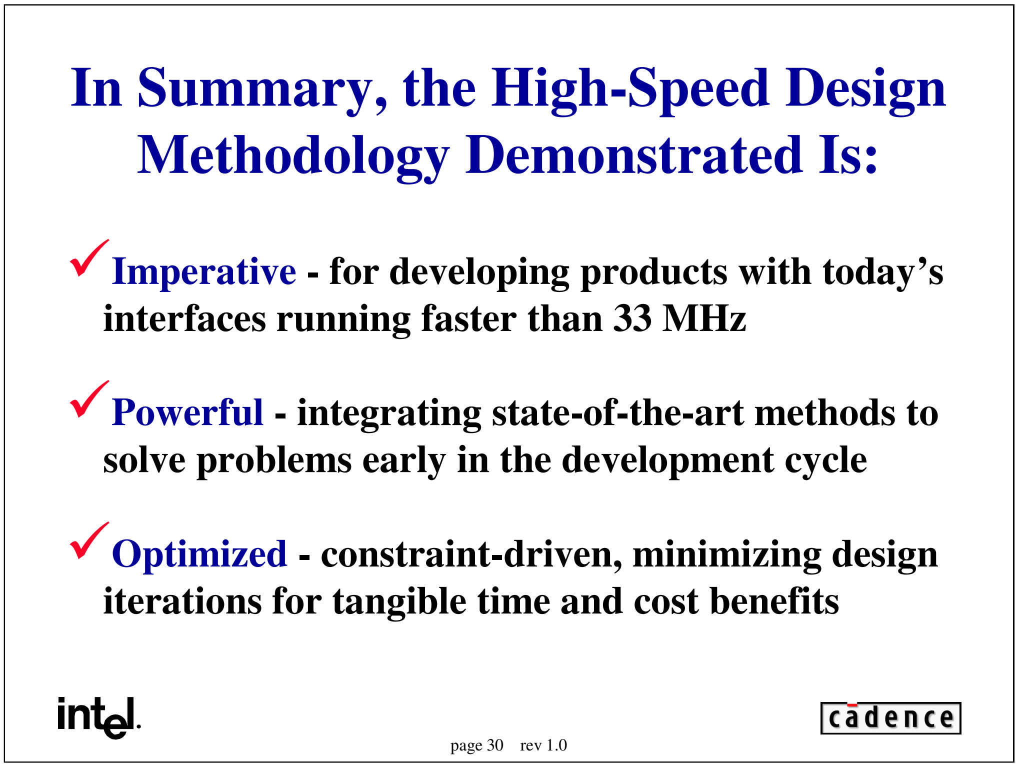
Throughout this presentation deriving topologies and constraints has been stressed. It should be said, however, that to derive these constraints and topologies requires a lot of skill on the part of the design engineer. The software tools make it possible to derive and apply the constraints to a board, but good engineering is required to ensure that the constraints are accurate and usable. Constraints that are poorly defined will prevent this methodology from doing what it was intended to do, which is to avoid multiple iterations through CAD.

In the first pilot program for this methodology we were not entirely successful in getting the topologies completed before routing had begun. This reinforces the point made earlier about a behavior change in the way designs are done. One of the most difficult parts of this new methodology is changing how engineers do their designs.

One of the biggest benefits of this new methodology is being able to reuse topologies and constraints for future boards. The savings in time and money has been addressed in the Measuring a Methodologies Improvement section of this paper.

As stated earlier, the post-route analysis has now really become a check off item because of this new methodology. During post-route we really just need to ensure that the constraints that were derived earlier in the process are reasonable. Provided there are no electrical constraint failures an in depth post-route analysis doesn’t need to be done.

While good training is important to ensure this methodology is successful, some things can still cause problems when using the methodology the first time. We have found that details like not upreving the topology after constraints have been changed or differences in pinuse between the topology and the netlist can cause problems. While it is easy to work around these problems once they are found, it is much harder to prevent them until you have experienced them.



In Summary, the High-Speed Design Methodology Demonstrated Is:

✓ **Imperative - for developing products with today's interfaces running faster than 33 MHz**

✓ **Powerful - integrating state-of-the-art methods to solve problems early in the development cycle**

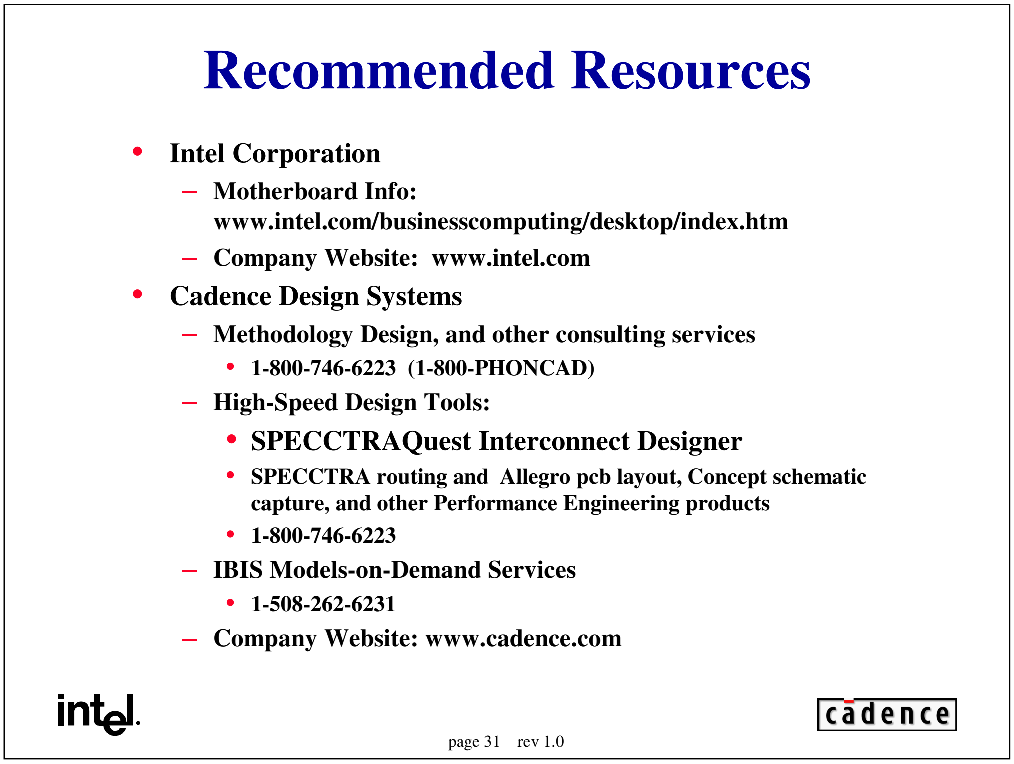
✓ **Optimized - constraint-driven, minimizing design iterations for tangible time and cost benefits**

intel.
cadence

page 30 rev 1.0

In this paper, we have demonstrated an optimized methodology for designing high-speed digital products that is now in place at Intel using Cadence's design environment. It was imperative that this methodology be developed and deployed for the successful design of today's and tomorrow's high-speed motherboards.

The final solution is simple, yet powerful and will empower engineers to carry out the development process in a way that is both optimized and leverageable from design to design.



Recommended Resources

- Intel Corporation
  - Motherboard Info: www.intel.com/businesscomputing/desktop/index.htm
  - Company Website: www.intel.com
- Cadence Design Systems
  - Methodology Design, and other consulting services
    - 1-800-746-6223 (1-800-PHONCAD)
  - High-Speed Design Tools:
    - SPECCTRAQuest Interconnect Designer
    - SPECCTRA routing and Allegro pcb layout, Concept schematic capture, and other Performance Engineering products
    - 1-800-746-6223
  - IBIS Models-on-Demand Services
    - 1-508-262-6231
  - Company Website: www.cadence.com

More information on advanced Intel motherboards and Cadence Design Systems products and consulting services can be found at the locations listed.

At Cadence, an entire Performance Engineering Business Unit is focused on solving customer's system design problems. Cadence focuses on providing *total solutions focused on business objectives;* packaging together hard-to-find solutions that normally include combinations of advanced design software technology, experienced design consultants, and design process/methodology services.

For companies already involved in high-speed design, Cadence now offers an IBIS Models-on-Demand service. IBIS models can be generated and delivered by Cadence for your critical components very early in the design cycle - right when you need them to make this Methodology work.