intel ®

Pentium® Pro Processor GTL+ Guidelines

CONTENTS

1.0. INTRODUCTION

The Pentium® Pro processor is the next generation in the Intel386™, Intel486™ and Pentium family of microprocessors. The Pentium Pro processor maintains binary compatibility with the 8086/88, 80286, Intel386, Intel486, and Pentium processors. The design of the external Pentium Pro processor bus enables the Pentium Pro processor to be "multiprocessor ready." To relax timing constraints on a bus that supports up to eight loads, the Pentium Pro processor implements a synchronous, latched bus protocol that allows a full clock cycle for signal transmission and a full clock cycle for signal interpretation and generation. This protocol simplifies interconnect timing requirements and supports 66 MHz system designs using standard ASIC interconnect technology. The Pentium Pro processor bus uses low-voltage-swing GTL+ I/O buffers, making high frequency signal communication between many loads easier.

The goal of this layout guideline is to provide a system designer with the information needed for the Pentium Pro processor and 82450 PCIset bus portion of PCB layout. This document provides guidelines and methodologies that are to be used with good engineering practices. It does not provide hard and fast rules. See the Pentium Pro processor specification and the applicable chipset specification for component specific electrical details. Intel strongly recommends running analog simulations using the available I/O buffer models together with layout information extracted from your specific design.

2.0. ABOUT THIS DOCUMENT

2.1. Document Organization

This section defines terms used in the document. Section 3 discusses specific system guidelines. This is a step-by-step methodology that Intel has successfully used to design Pentium Pro processor systems using the 82450 PCIset components. These systems were for validation and feasibility. Section 4 introduces the theories that are applicable to this Layout Guideline. Section 5 contains more details and insights. The items in section 5 expand on some of the rationale for the recommendations in the step-by-step methodology. This section also includes equations that may be used for reference.

The actual guidelines start at Section 3 -
A Recommended GTL+ Design Guideline.

2.2. Definition of Terms

Aggressor - a network that transmits a coupled signal to another network is called the aggressor network.

Bus Agent - a component or group of components that, when combined, represent a single load on the GTL+ bus.

Corner - describes how a component performs when all parameters that could impact performance are adjusted to have the same impact on performance. Examples of these parameters include variations in manufacturing process, operating temperature, and operating voltage. The results in performance of an electronic component that may change as a result of this include, but are not limited to: clock to output time, output driver edge rate, output drive current, and input drive current. Discussion of the "slow" corner would mean having a component operating at it's slowest, weakest performance. Similar discussion of "fast" corner would mean having a component operating at its fastest, strongest performance. Operation or simulation of a component at it's slow corner and fast corner is expected to bound the extremes between slowest, weakest performance and fastest, strongest performance.

- **Crosstalk** - the reception on a victim network of a signal imposed by aggressor network(s) through inductive and capacitive coupling between the networks.

- **Backward Crosstalk** - coupling which creates a signal in a victim network that travels in the opposite direction as the aggressor's signal.

- **Even Mode Crosstalk** - coupling from multiple aggressors when all the aggressors switch in the same direction that the victim is switching

- **Forward Crosstalk** - coupling which creates a signal in a victim network that travels in the same direction as the aggressor's signal.

- **Odd Mode Crosstalk** - coupling from multiple aggressors when all the aggressors switch in the opposite direction that the victim is switching

Flight Time - The delay between the driver and receiver introduced by the printed circuit board interconnects and the component loading effects. Although the name implies that this is the time required for a signal to travel from one end of the interconnect to the other, a better definition of this term is simply that it is the total delay the layout (interconnects plus loads) adds to the component timings. (This is similar to the usage of the

term "derating", but that term fails to acknowledge that transmission line effects are being included in the analysis.)

Flight time is therefore defined as the difference between when a signal at the input pin of a receiving agent crosses V_{REF} and the time that the output pin of the driving agent crosses the V_{REF} were it driving the test load used to specify that driver's AC timings.

T_{REF} for the Pentium Pro processor and the 82450 PCIset component test load is an idealized 25Ω resistor pulled up to 1.5 V, with component delays measured to the V_{REF} value of 1.0 V.

Flight time is defined as:

$$T_{FLIGHT} = T_{RECEIVER} - T_{REF}$$

where T_{REF} is the reference delay discussed above, and $T_{RECEIVER}$ is the time at which the waveform has a valid V_{REF} crossing (as described in the Pentium Pro Processor datasheet).

Figure 1 shows the definition of flight time. Notice that determining flight time requires a minimum of two simulations, one in which the driver is driving the test load, and one in which it is driving the actual system load.

- **Maximum and Minimum Flight Time** - Flight time variations can be caused by many different parameters. The more obvious causes include variation of the board dielectric constant, changes in load condition, variation in termination resistance and differences in I/O buffer performance as a function of temperature, voltage and manufacturing process. Some less obvious causes include effects of multiple signals switching and additional packaging affects. Table 4 includes recommended adjustment factors.

- The **Maximum Flight Time** is the largest flight time a network will experience under all variations of conditions.

- The **Minimum Flight Time** is the smallest flight time a network will experience under all variations of conditions.

GLT+ - is the bus technology used by the Pentium Pro processor. This is an incident wave switching, open drain bus with external pull-up resistors that provide both the high logic level and termination at each end of the bus. It is an enhancement to the GTL (Gunning Transceiver Logic) technology. See the Pentium Pro processor specification for more details of GTL+.

Network - the trace of a Printed Circuit Board (PCB) that completes an electrical connection between two or more components.
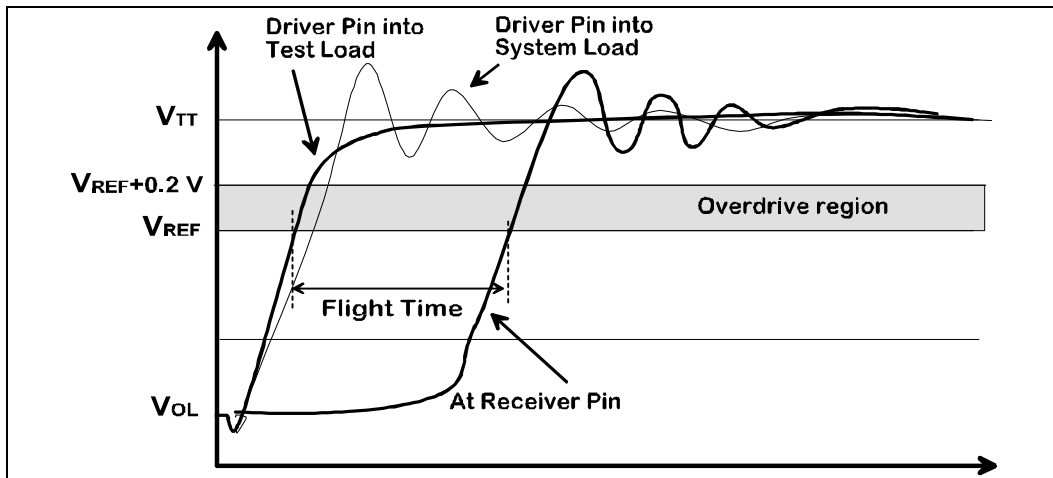

Figure 1. Definition of the Flight Time Criteria

Network Length - the distance between extreme bus agents on the network and does not include the distance connecting the end bus agents to the termination resistors.

Overdrive Region - is the voltage range, at a receiver, from V_{REF} + 200 mV for a low to high going signal and V_{REF} - 200 mV for a high to low going signal.

Overshoot - see appropriate component specification.

Ringback - see appropriate component specification.

Settling Limit - see appropriate component specification.

Setup Window - is the time between the beginning of Setup to Clock (T_{SU_MIN}) and the clock input. This window may be different for each type of bus agent in the system.

Stub - the branch from the trunk terminating at the pad of an agent. Expected to be very short (less than 1.1 inches including the internal package connection).

Trunk - the main connection, excluding interconnect branches, terminating at agent pads.

Undershoot - specified in the component datasheets (Pentium Pro processor and 82450 PCIset).

Victim - a network that receives a coupled crosstalk signal from another network is called the victim network.

3.0.   A RECOMMENDED GTL+ DESIGN GUIDELINE

The following step-by-step guideline was developed for systems based on one to four Pentium Pro processors and up to four 82450 PCIset loads.

The methodology recommended in this section is based on experience developed at Intel while developing many different Pentium Pro processor-based system for validation and for feasibility studies. This methodology relies on spreadsheet type calculations for initial timing analysis and using analog simulation tools to refine the timing analysis and to perform signal integrity/noise analysis. The analog simulations should be validated after actual systems become available. The validation portion of this section describes a method for determining the flight time in an actual system.

Outline of the guideline:

- Determine Components
- Initial Timing Analysis
- Determine General Layout, Routing and Topology Desired
- Estimate Component to Component Spacing for GTL+ Signals
- Route Board
- Simulation
  - Extract Interconnect Information
  - Run Uncoupled Simulation
  - Run Fully Coupled Simulation
- Validation
  - Measurements
  - Variation of V_{REF}
  - Determining Flight Time

3.1.   Determine Components

Determine which components will be used. Determine how many Pentium Pro processors, which and how many 82450 components (one or two memory controllers, one or two PCI bridges, GX or KX), and if any other GTL+ components will be used.

3.2.   Initial Timing Analysis

Do an initial timing analysis of the system. Equation 1 and Equation 2 are the basis for the timing analysis. To complete the timing analysis, values for the clock skew and clock jitter are needed, along with the component specifications. These are sufficient to determine the bounds for the system flight times.

Equation 1. Maximum Frequency

$$T_{CO\_MAX} + T_{SU\_MIN} + CLK_{SKEW} + CLK_{JITTER} + T_{FLT\_MAX} \leq \text{Clock Period}$$

Equation 2. Hold Time

$$T_{CO\_MIN} + T_{FLT\_MIN} \geq T_{HOLD} + CLK_{JITTER}$$

Symbols used in Equation 1 and Equation 2:

- T_{CO_MAX} is the maximum clock to output specification[1].

- T_{SU_MIN} is the minimum required time specified to setup before the clock[1].

- CLK_{JITTER} is the maximum clock edge to edge variation.

- CLK_{SKEW} is the maximum variation between components receiving the same clock edge.

- T_{FLT_MAX} is the maximum flight time as defined in Section 2.2.

- T_{FLT_MIN} is the minimum flight time as defined in Section 2.2.

- T_{CO_MIN} is the minimum clock to output specification[1]

- T_{HOLD} is the minimum specified input hold time.

Note:

1. The Clock to Output (T_{CO}) and Setup to Clock (T_{SU}) timings are both measured from the signals last crossing of V_{REF}, with the requirement that the signal does not violate the Ringback or edge rate limits. See the Pentium Pro processor datasheet for more details.

Solving these equation for T_{FLT} results in the following equations:

Equation 3. Maximum Flight Time

$$T_{FLT\_MAX} \leq \text{Clock Period} - T_{CO\_MAX} - T_{SU\_MIN} - CLK_{SKEW} - CLK_{JITTER}$$

Equation 4. Minimum Flight Time

$$T_{FLT\_MIN} \geq T_{HOLD} + CLK_{JITTER} - T_{CO\_MIN}$$

There are multiple cases to consider. Note that while the same trace connects two components, say A and B, the minimum and maximum flight time requirements for A driving B as well as B driving A must be met. The cases discussed in this document are:

- 150 MHz Pentium Pro processor driving a 150 MHz Pentium Pro processor

- 150 MHz Pentium Pro processor driving a PCIset component

- PCIset component driving a 150 MHz Pentium Pro processor

- PCIset component driving a PCIset component

- $\geq$ 166 MHz Pentium Pro processor, driving a $\geq$ 166 MHz Pentium Pro processor

- $\geq$ 166 MHz Pentium Pro processor driving a PCIset component

- PCIset component driving a $\geq$ 166 MHz Pentium Pro processor

A designer who used components other than those listed above would need to evaluate additional combinations of driver and receiver.

Table 1. Pentium® Pro Processor and 82450 PCIset GTL+ Parameters

IC Parameters	Pentium® Pro Processor at 150MHz	Pentium Pro Processor ≥ 166MHz	82450 PCIset
Clock to Output maximum (T_{CO_MAX}) ns	4.40	4.40	6.00
Clock to Output minimum (T_{CO_MIN}) ns	0.55	0.80	1.00
Setup time (T_{SU_MIN}) ns	2.20	2.20	4.50
Hold time (T_{HOLD}) ns	0.45	0.70	0.30

intel.

Table 2 and Table 3 are derived assuming:

- CLK_{SKEW} = 0.7 ns

- CLK_{JITTER} = 0.2 ns

Table 2. T_{FLT_MAX} Calculations for 66 MHz

Driver	Receiver	Clk Period	T_{CO_MAX}	T_{SU_MIN}	Clk_{SKEW}	Clk_{JITTER}	T_{FLT_MAX}
150 MHz CPU	150 MHz CPU	15.00	4.40	2.20	0.70	0.20	7.50
150 MHz CPU	82450	15.00	4.40	4.50	0.70	0.20	5.20
82450	150 MHz CPU	15.00	6.00	2.20	0.70	0.20	5.90
≥ 166MHz CPU	≥ 166MHz CPU	15.00	4.40	2.20	0.70	0.20	7.50
≥ 166MHz CPU	82450	15.00	4.40	4.50	0.70	0.20	5.20
82450	≥ 166MHz CPU	15.00	6.00	2.20	0.70	0.20	5.90
82450	82450	15.00	6.00	4.50	0.70	0.20	3.60

Table 3. T_{FLT_MIN} Calculations (Frequency Independent)

Driver	Receiver	T_{HOLD}	Clk_{SKEW}	T_{CO_MIN}	T_{FLT_MIN}
150 MHz CPU	150 MHz CPU	0.45	0.70	0.55	0.60
150 MHz CPU	82450	0.30	0.70	0.55	0.45
82450	150 MHz CPU	0.45	0.70	1.00	0.15
≥ 166MHz CPU	≥ 166MHz CPU	0.70	0.70	0.80	0.60
≥ 166MHz CPU	82450	0.30	0.70	0.80	0.20
82450	≥ 166MHz CPU	0.70	0.70	1.00	0.40
82450	82450	0.30	0.70	1.00	0.00

The effective board propagation constant (S_{EFF}) is a function of

- Dielectric constant (ε_r) of the PCB material

- The type of trace connecting the components (stripline or microstrip)

- The length of the trace and the load of the components on the trace. (Note that the board propagation constant multiplied by the trace length is a **component** of the flight time but **not necessarily equal to** the flight time.)

The standard "textbook" equations used to calculate the expected signal propagation rate of a board are included in Section 5.1. Intel recommends some additional adjustment factors which have been derived from empirical testing. These adjustment factors are not found in textbooks but are used to account for differences between the expected values calculated using textbook formulas and values that have been measured in a variety of actual systems.

The adjustment factors to the timing equations account for the following phenomena that Intel has observed:

- The falling edge propagation rate is 8% slower than predicted by the "text-book" equations.

- The 82450 rising edge rate is slower than the GTL+ specification of 0.3 V/ns, requiring extrapolation that causes additional delay.

- The crosstalk on the PCB and internal to the package can cause variation in the signals.

- Delay caused by simultaneous switching noise (SSN) of multiple outputs.

- Edge rate degradation caused by inductance in the current return path.

SSN refers to **S**imultaneous **S**witching **N**oise. That is noise in the design from multiple outputs changing state at the same time.

When doing spreadsheet-based calculations, include the value in the "Total Adjustment" column as part of the flight time. (That is, $T_{FLIGHT} = [S_{EFF} * \text{Trace Length}]+ \text{Adjustment}$.) For uncoupled simulations add the "Package & PCB Coupling & SSN" column plus the "Many Bit Push-Out Due to Connectors" column (if, in fact the design will have series connectors on the GTL+ bus) to the board propagation time calculated by the simulator. Similarly, for fully coupled simulations, when appropriate, add the "Many Bit Push-Out Due to Connectors" column to the board propagation time calculated by the simulator.

Note that the spreadsheet calculation is based on the component specification timing values, which are into a test load. The test load is likely to be different than an actual system. This difference in loads can impact the performance of the output buffer, causing a difference in the component T_{CO} in an actual system. The adjustment factors in Table 4 are from systems with $47\Omega \geq R_{TT} \geq 51\Omega$.

Table 4. Empirical Adjustment Factors

		Sum for Spreadsheet				
				Sum for Uncoupled Sims		
					Use for Fully Coupled Sims	Use for Spreadsheet
Driver	Receiver	Settling from Previous Transition	Slow Edge Rate	Package & PCB Coupling & SSN	Many Bit Push Out Due to Connectors	Total Adjustment in Modular Design
CPU	CPU	0.10	0.00	0.00	0.70	0.80
CPU	82450	0.10	0.00	0.00	0.18	0.28
82450	CPU	0.10	0.45	0.36	0.32	1.23
82450	82450	0.10	0.55	1.00	0.00	1.65

NOTE
- All values are in nanoseconds (ns).

9

3.3.   Determine General Layout, Routing, and Topology Desired

Once the processor bus components have been selected, and the timing budget calculated, then determine their approximate location on the printed circuit board. Estimate the printed circuit board parameters from the placement and other information including the following general layout/routing guidelines:

- Daisy chain all GTL+ signals, keeping stubs to 82450 PCIset components under 0.25 inches and no stubs to the Pentium Pro processor(s).

- Distribute V_{TT} with a wide trace. A 50 mil minimum width is recommended. Route the V_{TT} trace with the same topology as the GTL+ traces.

- Place termination resistors at each end of each GTL+ signal. Minimize the inductance between the V_{TT} distribution and the termination resistors. Provide at least one decoupling capacitor for every four termination resistors.

- Plan to place V_{REF} resistor divider pairs at each 82450 component and a pair of V_{REF} resistor divider pairs at each processor.

- Locate the processor(s) and 82450 PCIset as required to meet timing. Systems with busses greater than 14 inches in length may need to have the 82450 PCIset components in the middle of the bus to minimize the flight time from the 82450 PCIset components to the processors and/or other GTL+ agents.

- Keep the overall length of the bus as short as possible (but don't forget minimum component to component distances to meet hold times).

- Avoid the use of connectors in the GTL+ bus, particularly for heavily loaded designs (long GTL+ bus and or more than 3 GTL+ agents). When connectors are used, the stub and loading requirements must be maintained. This generally means that connectors will be placed in series on the bus. Use quality "high-speed" connectors if connectors are required.

- Plan to minimize crosstalk by

    - Maximizing the line-to-line spacing (at least 10 mils between traces, except when routing between pins of the processor).

    - Minimizing the dielectric used in the system (maximum of 4.6).

    - Minimize the cross sectional area of the traces, (5 mil lines with 1/2 ounce/ft2 copper - but watch out for higher resistivity traces).

    - Eliminating parallel traces between layers not separated by a power or ground plane.

    - Isolate GTL+ signals from other signals (at least 25 mils from non GTL+ signals to GTL+ signals).

    - Route the same type of GTL+ I/O signals in isolated signal groups. That is route the data signals in one group, the address signals in another group. Keep at least 25 mils between each group of signals.

The placement of the Pentium Pro processor, 82450 PCIset and/or custom ASIC(s) on the processor bus must be carefully chosen. The Pentium Pro processor's buffers are faster (shorter clock to output delay), and have faster rising edge rates, than the 82450 PCIset buffers. The 82450 PCIset buffers have faster falling edge rates. These characteristics of the Pentium Pro processor buffers and the 82450 PCIset buffers cause the route order on the board to be very important. Systems with more than two Pentium Pro processor components and/or more than the minimum number of 82450 PCIset components should place an equal number of processors on each end of the network. Having the fast buffers on the ends of the network compensates for the longer flight time needed to go to the opposite end of the network relative to the time from the middle to either end of the network. Having the buffer(s) with the slower rising edge rate in the middle of the network causes less ringing (noise) on the network than having the faster buffer in the middle. Having the buffer(s) with slower clock to output delay in the middle of the bus may allow a longer overall bus. Using a custom ASIC (with different timings than Pentium Pro processor or 82450 PCIset) on the Pentium Pro processor bus will require additional analog simulations to determine the optimum location of each agent along the bus.
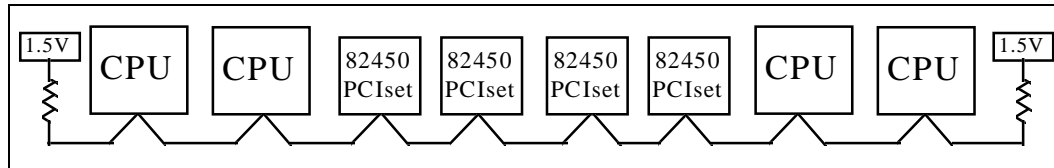
...

3.5.   Route Board

Lay the board out using the guidelines detailed in Section 3.3. Keep the estimated spacing and timing requirements in mind during the layout of the board. If it becomes apparent that the placement and estimated spacing are not going to support the timing requirements, then revise the timing requirements estimates before the routing is complete.

After the GTL+ portion of the system is routed, extract the actual routed line lengths and verify that the actual routing provided acceptable timing.

3.6.   Simulation

Intel strongly suggests running analog simulations for Pentium Pro processor designs. Intel provides the Pentium Pro processor I/O Buffer Models and the 82450 PCIset I/O Buffer Models in IBIS 2.1 formats. These models are available from your local Intel office.

Accurate simulations require that the actual range of parameters be used in the simulations. Intel has consistently measured the cross-sectional resistivity of the PCB copper to be in the order of 1 ohm*mil^2/inch, not the 0.662 ohm*mil^2/inch value for annealed copper that is published in reference material.

Positioning drivers with faster edges closer to the middle of the network results in more noise than positioning them towards the ends. We have also shown that the worst-case noise margin can be generated by drivers located in all positions (given appropriate variations in the other network parameters). Therefore, we recommend stimulating the networks from all driver locations, and analyzing each receiver for each possible driver.

We assumed that it is impractical to terminate each network independently, and that the designer will choose one or two values to terminate all of the networks. Our analysis has shown that increasing the value of R_T results in decreased noise margin on the rising edge, and decreasing the value of R_T results in decreased noise margin on the falling edge. Therefore it is not necessary to budget for R_T variation if the selected value of R_T + 5% is used on the rising edge and R_T-5% is used on the falling edge, since the simulation results will already include the extreme effects. If ±1% resistors are used for R_T the nominal value of R_T can be used to simulate each edge.

Faster edge rates cause increased ringback, which reduces the noise margin on the rising edge (Low to High); therefore only the fast corner (voltage, temperature, and process) I/O buffer model needs to be simulated for the Low to High transitions to evaluate signal quality.

Analysis has also shown that both fast and slow models must be run to verify signal quality on the falling edge (High to Low). The fast corner is needed because the fast edge rate creates the most noise. The slow corner is needed because the buffer's drive capability will be a minimum, causing the V_{OL} to shift up, which may cause the noise from the slower edge to exceed the available budget. The slow corner I/O buffer model is used to check the maximum flight time.

Lengthening the stubs correlates to more (increased) ringback and a corresponding reduction in noise margin on the rising edge. Therefore it is acceptable to only simulate rising edges with all stubs at the maximum value on all bus agents (0.9 inches for the processor, which represents the maximum package stub, and 0.9 inches for 82450 PCIset which including the maximum internal package stub and a 0.25 inch stub on the PCB).

The falling edge analysis did not always show that lengthening the stubs increased the ringback (and therefore reducing the noise margin). Approximately 25% of the networks in the analysis showed increased noise of up to 50 mV for less than maximum stub lengths. Therefore reducing the noise margin available on the falling edge by 50 mV precludes the need to simulate the networks with a variety of stub lengths.

Using maximum length package stubs can be pessimistic. Actual internal package stub lengths are provided with the I/O buffer models for the Pentium Pro processor or the 82450 PCIset devices. The internal package stub lengths **may** change slightly over time with new steppings of the components.

Intel has determined that, to properly model the effects of the "package stub" (connection between the die pad and the external pin), the package traces and pins should be represented using transmission line segments. The length, Z_0 and S_0 of each stub is given in IBIS compatible ".pkg" files. These files include the stub lengths, as well as the package trace resistance. The packaging files are in an IBIS compatible format. Because of differences in stub length between the 150-MHz Pentium Pro processor with the 256kbyte L2 cache and all other versions of the Pentium Pro processor package, two files have been included: "ppromin.pkg" and "ppromax.pkg". Use the stub lengths listed in the

"ppromin.pkg" file when doing simulations that involve minimum hold time. Use the stub lengths listed in "ppromax.pkg" when doing simulations involving maximum setup time or slow corner V_{OL} predictions. For use in helping correlate simulation results to actual measurements, the file "pprolen.txt" contains the actual trace length for each package type.

The transmission line package models must be inserted between the output of the buffer and the net it is driving. Likewise, the package model must also be placed between a net and the input of a receiver model. This is generally done by editing your simulator's net description or topology file.

We have found wide variation in noise margins when we vary the stub impedance and the PCB's Z_0 and S_0. Our analysis has shown that extremes in impedance do NOT necessarily produce the extreme variations in noise margin. We therefore recommend that PCB parameters be controlled as tightly as possible, with a sampling of the allowable Z_0 and S_0 simulated. Intel recommends running uncoupled simulations using the Z_0 of the package stubs; and performing fully coupled simulations if increased accuracy is needed or desired. Accounting for crosstalk within the device package by varying the stub impedance was investigated and was not found to be sufficiently accurate. This lead to the development of full package models for the PQFP packages.

3.6.1. EXTRACT INTERCONNECT INFORMATION

Extract the actual interconnect information for the board from the CAD layout tools.

3.6.2. RUN UNCOUPLED SIMULATION

Intel recommends running uncoupled simulations **at the pin** for timing and **at the pad** for signal quality. Note that simulations at the pin and at the pad can have more than 200 mV difference. The system measurements that Intel has done shows much better correlation to the pin measurements than to the pad measurements for uncoupled simulations.

The timing analysis using flight times extracted from simulations may not have enough timing margin to use T_{CO_MAX} with the fast corner I/O buffer models. If more timing margin is needed, Intel recommends using T_{CO} of 2.4 ns for the Pentium Pro processor at the fast corner and T_{CO} of 3.4 ns for the 82450 fast corner. These T_{CO} values represent a fast output buffer and the inclusion of the worst case internal component parameters (clock skew, clock jitter, etc.). These values are to be used in

conjunction with the other component values included in this document.

Run uncoupled simulations to evaluate the noise in the system. Because these are simulations on the isolated network, be sure to either add the appropriate adjustments from Table 4 or shift the thresholds to include the budget described in Section 4.3.

Shifting the threshold provides a good approximation for actual timings but does not accurately reflect signal quality - particularly when ringback is allowed. ("Shift the thresholds" means, rather than set the high going threshold at V_{REF} + 200 mV it should be set at V_{REF} + 200 mV + Noise Budget and correspondingly the low going threshold at V_{REF} - 200 mV it should be set at V_{REF} - 200 mV-Noise Budget.)

3.6.3. RUN FULLY COUPLED SIMULATION

Intel did achieve good correlation to simulation when using full package models for the PQFP PCIset and fully coupled PCB models. (There is not enough coupling in the Pentium Pro processor package to warrant a package model that includes coupling.) The fully coupled PQFP package models were used to refine the simulation predictions. If resources preclude doing fully coupled simulations on all the networks (including fully coupled package models), then after running uncoupled simulations, approximately the worst 10 signals from the uncoupled simulations should be re-simulated including, coupling and using full package models.

The released I/O buffer models, at the time of this document publication, do not include fully coupled package models. If you require fully coupled packaged models contact, your Intel representative.

Run fully coupled (PCB & package) simulation on the design and evaluate at the PAD. (This simulation can consume LOTS of processor cycles.)

OR

Pick the worst 10 signals from the uncoupled simulation.

Run fully coupled (PCB & package) simulation on selected worst signals and evaluate these signals at the PAD. This assumes that while the single worst signal from the uncoupled simulation may not actually be the worst signal when more factors are considered that the worst signal will be found in one of the worst signals from the uncoupled simulation. Also simulate the following signals if they are not already in the 10 worst: D21#, D26#, A14#, and A34#. These signals represent

13

the longest total package stub length or the most heavily loaded signals.

3.7. Validation

Build systems and validate the design and simulation assumptions.

3.7.1. MEASUREMENTS

Note that the GTL+ specification for signal quality is at the **pad** of the component. The expected method of determining the signal quality is to run analog simulations for the pin and the pad. Then correlate the simulations at the pin against actual system measurements at the pin. Good correlation at the pin leads to confidence that the simulation of the pad is accurate. Controlling the temperature and voltage to correspond to the I/O buffer model extremes should enhance the correlation between simulations and the actual system. Using the actual package stub length information for the simulations should also enhance the correlation.

3.7.2. VARIATION OF V_{REF}

Variation of V_{REF} in a system is one method to empirically determine the noise margin in a particular system.

By modifying the system to allow V_{REF} to vary for each of the GTL+ bus components, then moving V_{REF} higher or lower until a failure occurs, the amount by which V_{REF} can be varied before causing a failure will determine the noise margin under the test conditions.

For systems designed with V_{REF} supplied from its own pair of voltage dividing resistors for each GTL+ bus component, by removing the resistor pair at each component and replacing each pair with a three terminal variable resistor it is possible to individually vary the V_{REF} at each component over the full range from 0V to V_{TT} (the 1.5V GTL+ termination voltage). Intel has been successful at replacing the divider pair with a 1 KΩ, 15 turn trimming resistor. This allowed sufficient adjustment precision to vary V_{REF} by as little as the 1 mV resolution of a digital multimeter. Systems which distribute a single V_{REF} from each end of the bus would need to make an appropriate modification to obtain the same results. (After modification, adjust V_{REF} to the normal 1.000V and test the board to verify correct operation.)

Run the modified system and vary V_{REF} until failures occur. Measure V_{REF} at the failure point and determine the amount of margin in the system under the test conditions.

Each system design may have sensitivity to different code sequences. This test only indicates the amount of margin available in the particular system tested under the specific test conditions. Varying component temperature and voltage across their extremes improves the applicability of the test to other systems as well as giving indications of the sensitivity to these system variables. It would not be practical to perform this test with all combinations of fast corner and slow corner parts. Similarly it is difficult to identify the most stressful software to operate during this test. Still, the test can give a good indication of the relative health of the system. Performing these tests with the processor caches off may increase GTL+ bus traffic. Running tests with the processor caches on may increase PCI bus traffic.

Systems which Intel has performed this test on have all shown at least 200 mV of margin and generally more than 300 mV of margin.

3.7.3. DETERMINING FLIGHT TIME

Flight time is defined as the difference between the time the signal is valid at the receiver and the T_{CO} of the driver into the test load. It is necessary to know the actual T_{CO} of the device being used to make a flight time measurement, but the observed Low to High T_{CO} is a result of the effective R_{TT} and the Z_{EFF} of the PCB, and may be quite different than the T_{CO} into the tester spec load (25Ω). If one assumes the T_{CO} is the T_{CO_MAX} from the specification, then the resulting flight time could be too small by up to 3 ns, leading one to believe there is more margin than actually exists. If one assumes T_{CO_MIN} then the flight time could be overestimated by 4 ns, which is almost sure to cause timing violations.

The best way to determine T_{CO} is to actually have the driver output tied to the tester load (25Ω), but this is rarely possible. (This can be approximated by using a long section of 25Ω coax.) One method to approximate the Low to High T_{CO} is by measuring the High to Low T_{CO} at the driver (clock at 1.5V to output at 1V) and using this to predict the Low to High T_{CO} (our experience has been that the High to Low T_{CO} observed in the system is within 200 ps of the actual T_{CO} and is relatively insensitive to R_{TT} value). The charts in Figures 3 and 4 can be used to predict Low to High T_{CO} given a High to Low T_{CO} measurement. The actual Low to High T_{CO} for any given High to Low T_{CO} will lie between the lines on the chart. Note that this method is relatively

accurate (it over predicts T_{CO} by less than 400 ps for larger values (> 3 ns) of High to Low T_{CO} but can over predict the Low to High T_{CO} by as much as 1 ns for the smaller values of (< 2.4 ns) High to Low T_{CO}'s).
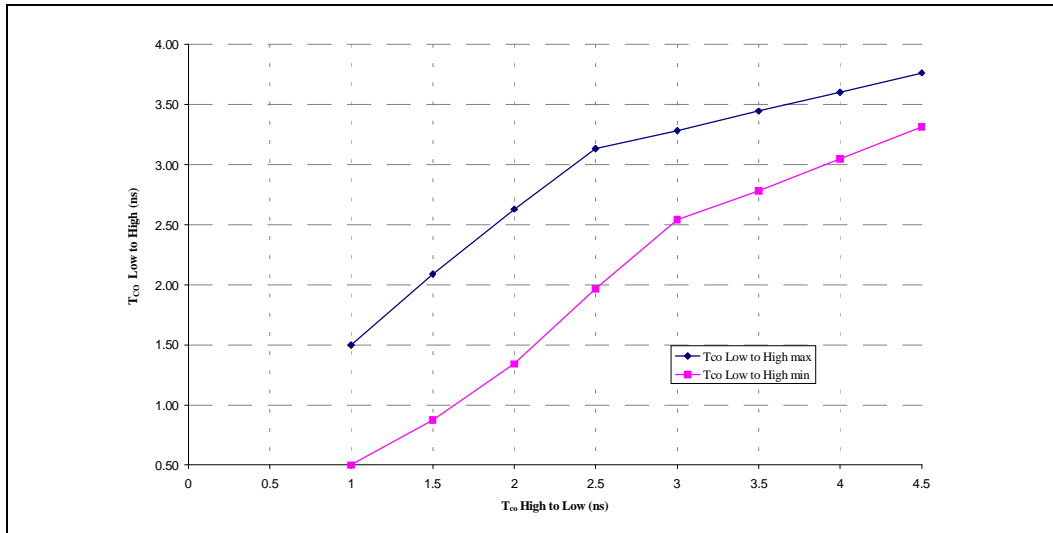

Figure 3. T_{CO} Correlation for Pentium® Pro Processor
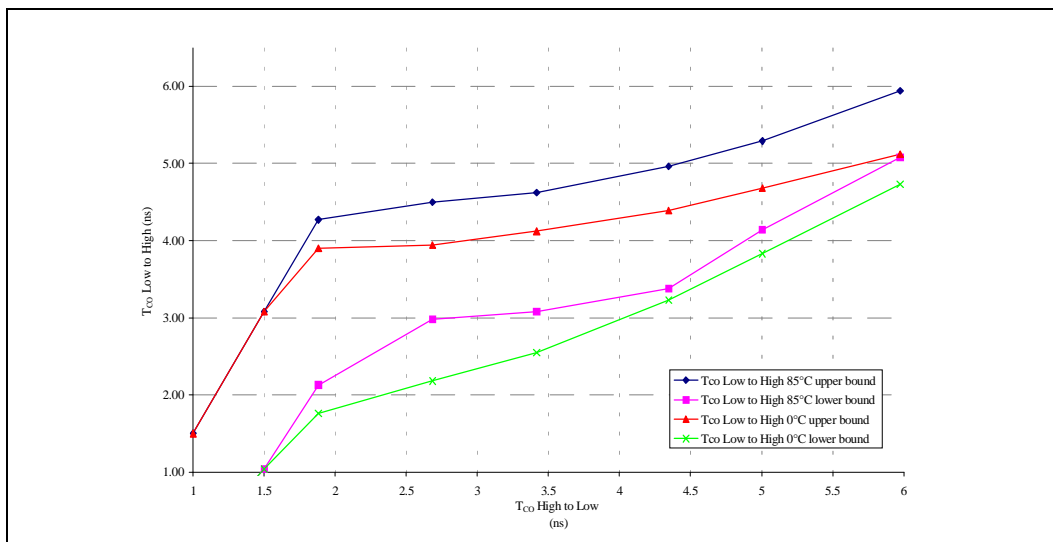

Figure 4. T_{CO} Correlation for 82450

15

4.0. THEORY

4.1. GTL+

GTL+ is the electrical bus technology used for the Pentium Pro processor bus. This is an incident wave switching, open-drain bus with external pull-up resistors that provide both the high logic level and termination at each end of the bus. The specification defines:

- Termination voltage (V_{TT}).
- Termination resistance (R_T).
- Maximum output low voltage (V_{OL}).
- Output driver edge rate under specific load conditions.
- Maximum bus agent loading (capacitance and package stub length).
- Receiver high and low voltage level.
- Receiver reference voltage (V_{REF}) as a function of termination voltage (V_{TT}).
- Receiver ringback characterization.

The complete GTL+ specification can be found in the Pentium Pro processor datasheet. Layout recommendations for the GTL+ bus can be found in Section 3 of this document.

4.2. Timing Requirements

The system timing for GTL+ is dependent on many things. Each of the following elements combine to determine the maximum and minimum frequency the GTL+ bus can support:

- The range of timings for each of the agents in the system.
  - Clock to output [T_{CO}]. (Note that the system load is likely to be different from the "specification" load therefore the T_{CO} observed in the system may not be the same as the T_{CO} from the specification.)
  - The minimum required time to setup to clock [T_{SU_MIN}] for each receiving agent.
- The range of flight time between each component. This includes:
  - The velocity of propagation for the loaded printed circuit board [S_{EFF}].
  - The board loading impact on the effective T_{CO} in the system.

- The amount of skew and jitter in the system clock generation and distribution.
- Changes in flight time due to crosstalk, noise, and other effects.

4.3. Noise Margin

The goal of these sections is to describe the total amount of noise that can be tolerated in a system (the noise budget), identify the sources of noise in the system, and recommend methods to analyze and control the noise so that the allowed noise budget is not exceeded.

There are several sources of noise which must be accounted for in the system noise budget, including:

- V_{REF} variation
- Variation in V_{TT}
- Crosstalk
- Ringback due to impedance variation along the network, termination mismatch, and/or stubs on the network
- Data pattern dependencies

The total noise budget is calculated by taking the difference in the worst case specified input level and the worst case driven output level.

Sections 4.3.1 and 4.3.2 discuss calculating noise margin. These sections do not discuss ringback tolerant receivers which can increase the effective noise margin. See the component datasheet(s) for information about ringback.

4.3.1. FALLING EDGE OR LOW LEVEL NOISE MARGIN

Equation 5. Low Level Noise Margin

$$\text{Noise Margin}_{\text{LOW LEVEL}} = V_{IL\_MAX} - V_{OL\_MAX} \Rightarrow$$
$$(V_{REF\_MIN} - 200 \text{ mV}) - V_{OL\_MAX}$$

Symbols for Equation 5 are:

- V_{IL_MAX} is the maximum specified valid input low level from the component specification.
- V_{IH_MIN} is the minimum specified valid input high level from the component specification.
- V_{OL_MAX} is the maximum output low level the component will drive.
- V_{REF_MIN} is the minimum valid voltage reference used for the threshold reference.

V_{OL_MAX} for the Pentium Pro processor is 600 mV, and is specified into a 25Ω test load tied to 1.5V. This corresponds to the maximum output low current (I_{OL}) of 36 mA. This implies an effective maximum "on" resistance of 16.67Ω. This maximum condition corresponds to the slow corner components and models.

The implied effective minimum "on" resistance is 6.25Ω with the same test load, minimum output low voltage and the specified minimum output low current of 48 mA.

This condition corresponds to the fast corner components and models.

$$V_{REF\_MIN} = [\ 2/3\ (\ V_{TT\_MIN})\ ]\ \text{-}\ 2\%$$
$$= [\ 2/3\ (1.5\ V\ \text{-}\ 10\%)\ ]\ \text{-}\ 2\%$$
$$= 882\ mV$$

The output low current for V_{REF_MIN} can be calculated as shown below:

$$I = V/R$$

$$I = 1.35/(25Ω + 16.67Ω) = 32.4\ mA$$

then the
V_{OL_MAX} for V_{REF_MIN} is (32.4 * 16.67) = 540 mV

So from
CPU Driving Noise Margin$_{LOW\ LEVEL}$
$$= (V_{REF\_MIN}\text{-}200\ mV)\text{-}V_{OL\_MAX}$$
$$= (882\ mV\ \text{-}\ 200\ mV)\ \text{-}\ 540\ mV$$
$$= 142\ mV$$

These calculations are for an effective termination resistance of 25Ω which corresponds to a 50Ω termination at each end of a GTL+ signal. These calculations DO NOT include any resistive drop along the trace. The resistive drop along the trace can be significant with long traces and 1/2 oz/ft^2 copper (>8Ω causing up to 200 mV for a 24 inch 4 mil actual etched trace with the fast corner component driving). Different termination resistors will allow different low level noise margins. Larger value resistors will reduce the current in the line, reducing the V_{OL} and increasing the low level noise margin.

Similar calculations for the fast and slow corners of the Pentium Pro processor driving and the 82450 PCIset driving yield the low level noise margins shown in Table 5.
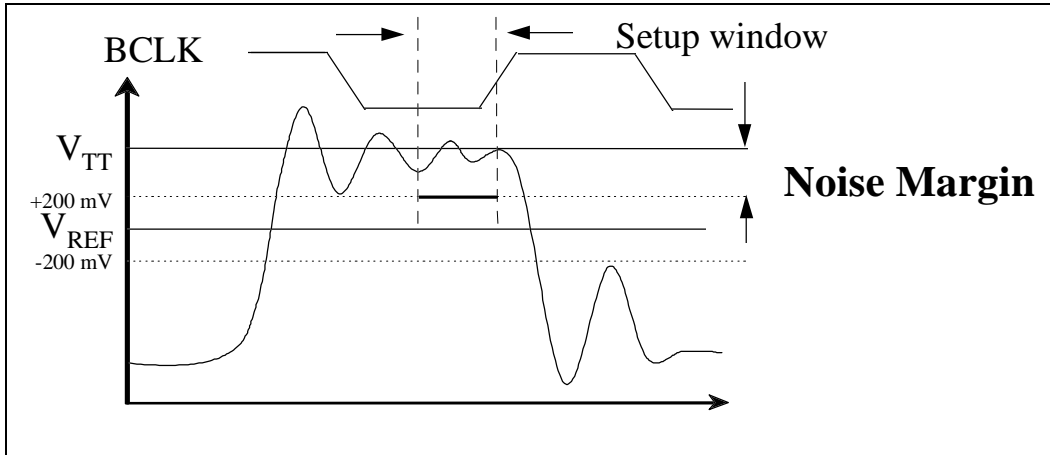

Figure 5. Rising Edge Noise Margin

Table 5. Low Level Noise Margin

Corner & Device	I_{OL}(mA)	V_{TT} (V)	$R_{ON}(\Omega)$	V_{OL} (mV)	Margin (mV)
Slow/CPU	32.40	1.35	16.67	540	142
Slow/CPU	36.00	1.50	16.67	600	180
Slow/CPU	39.60	1.65	16.67	660	218
Fast/CPU	43.20	1.35	6.25	270	412
Fast/CPU	48.00	1.50	6.25	300	480
Fast/CPU	52.80	1.65	6.25	330	548
Slow/82450	34.20	1.35	14.47	498	187
Slow/82450	38.00	1.50	14.47	550	230
Slow/82450	41.80	1.65	14.47	605	273
Fast/82450	43.20	1.35	6.25	270	412
Fast/82450	48.00	1.50	6.25	300	480
Fast/82450	52.80	1.65	6.25	330	548

4.3.2.   RISING EDGE OR HIGH LEVEL NOISE MARGIN

Equation 6. High Level Noise Margin

Noise Margin$_{HIGH\ LEVEL}$ = V_{OH_MIN} - V_{IH_MIN} $\Rightarrow$

V_{TT_MIN} - (V_{REF_MAX} + 200 mV)

Symbols for Equation 6 are:

- V_{IH_MIN} is the minimum specified valid input high level from the component specification.

- V_{OH_MIN} is the minimum output high level the component will drive.

- V_{TT_MIN} is the minimum termination voltage.

- V_{REF_MAX} is the maximum valid voltage reference used for the threshold reference.

- V_{OH_MIN} for the GTL+ signals is V_{TT_MIN}. This can be 1.5V - 10%, or 1.35V. Since V_{REF} is defined as a function of V_{TT} the maximum V_{REF} when V_{TT} is 1.35V is 2/3 *(1.35V) + 2% = 918 mV

- Then Noise Margin$_{HIGH\ LEVEL}$

    = V_{TT_MIN} - (V_{REF_MAX} + 200 mV)

    = 1.35V - 918 mV - 200 mV

    = 232 mV

Note that while the high level noise margin is not sensitive to the value of the termination resistance, using larger value termination resistors would reduce the current in the line, slowing the rising edge rate and hence increasing the flight time.

4.3.3.   RECOMMENDED NOISE BUDGET

The slow corner falling edge noise margin is reduced due to the increase in V_{OL} associated with the reduced drive capability of the worst case buffer, yielding the smallest margin. This requires a different budget than the fast corner falling edge or the rising edges. The slow corner edge rates are slowed by approximately 1/3, resulting in a maximum crosstalk length that is three times longer than the fast corner. Systems that are designed to minimize crosstalk with the fast corner edge rates, are not likely to have the maximum crosstalk lengths at the slow corner. Therefore, maximum coupled noise is unlikely to occur. In addition, the voltage swing is reduced by 15%, reducing the crosstalk budget to 60 mV. This leaves only 100 mV for the ringback portion of the noise budget, can be achieved with the slower edge and reduced voltage swing. The biggest concern for the slow corner signal quality is achieving a sufficiently low V_{OL}. Trace resistance for 1/2 ounce copper on a 24 inch long network can be 8Ω or more. This would increase the V_{OL} at the farthest receiver more than 180 mV (for a nominal 5 mil line with an actual etched width of 5 mils). Using

19

1 ounce copper or shortening the maximum network length may be necessary to minimize the V_{OL} loss along the network. Adjusting R_T to balance the noise margin could also be an option.

A representative noise budget (within the setup window, $V_{TT} = 1.5V$ and $V_{REF} - 2/3\ V_{TT}$) for all rising edges and the typical falling edge is:

V_{REF} variation	20 mV
V_{TT} variation	20 mV
Crosstalk	110 mV
Ringback	150 mV
Total budget	300 mV

A representative noise budget (within the setup window, $V_{TT} = 1.5V$ and $V_{REF} - 2/3\ V_{TT}$) for the slow corner falling edge is:

V_{REF} variation	20 mV
V_{TT} variation	20 mV
Crosstalk	60 mV
Ringback	100 mV
Total budget	200 mV

The V_{REF} variation is based on the +/-2% tolerance in V_{REF}. The V_{TT} variation term is based on shifting V_{OL} closer to V_{REF} when V_{TT} is lowered (simple voltage divider effect). The required margin for these can both be reduced by holding tighter tolerances on V_{REF} and V_{TT}. Note that Table 5 shows 180 mV calculated noise margin which includes 20 mV of noise for V_{REF}.

The crosstalk budget comes from 5 mil lines with 10 mil spacing (5/10), using 1/2 ounce/ft2 copper and a dielectric constant of 4.0. This budget also assumes that there is no doubling; see Sections 4.4 and 4.4.1. Using 1 ounce/ft2 copper (1.4 mil thick) doubles the cross-sectional area of the traces and therefore doubles the crosstalk. Using a dielectric material with a constant higher than 4.0 will cause the signals to propagate at a slower rate, which will increase the maximum coupled length, but using a higher dielectric constant material while maintaining the same impedance will cause the traces to be farther from their reference plane, increasing crosstalk. The total impact of using a higher dielectric material, while keeping the rest of the board parameters the same, is more noise from crosstalk.

Ringback is a function of the following parameters:

- R_T value (and variation)

- Driver's edge rate

- Stubs along the network and their length (including internal package connection)

- Inter-agent spacing

- Total network length

- Bus agent position

- Impedance variations (PCB material and internal package stubs)

4.4. Crosstalk Theory

GTL+ signals swing across a smaller voltage range and have a correspondingly smaller noise margins than technologies that have traditionally been used in personal computer designs. This requires that designers using GTL+ be more aware of crosstalk than they may have been in past designs.

Crosstalk is caused through capacitive and inductive coupling between networks. Crosstalk appears as both backward crosstalk and as forward crosstalk. Backward crosstalk creates an induced signal on a victim network that travels in a direction opposite that of the aggressor's signal. Forward crosstalk creates a signal that travels in the same direction as the aggressor's signal. On the GTL+ bus, a driver on the aggressor network is not at the end of the network, therefore it sends signals in both directions on the aggressor's network. The signal propagating in each direction causes crosstalk on the victim network. Figure 6 shows two aggressors on each side of the victim. A third aggressor on each side of the victim network is not shown, as it has negligible effect on crosstalk. (There may be additional noise from multiple bits switching, but these are not believed to be from crosstalk.) The maximum crosstalk occurs when all the aggressors are switching in the same direction at the same time. Figure 7 shows a driver on the aggressor network and a receiver on the victim network that are not at the ends of the network. There is crosstalk internal to the IC packages, which can also affect the signal quality/noise.
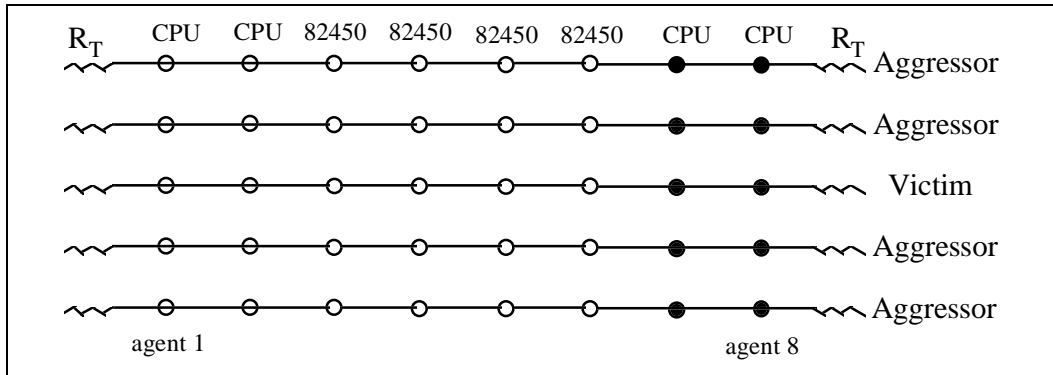
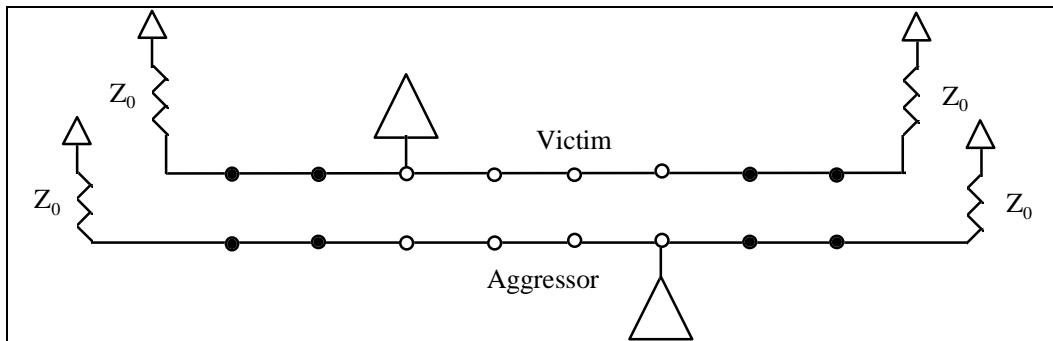**Figure 6. Aggressor and Victim Networks**

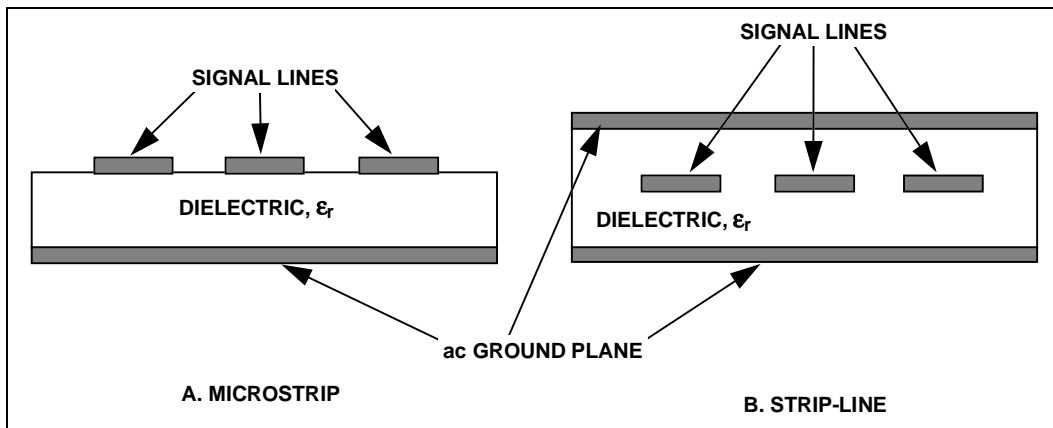**Figure 7. Driver on Aggressor Network: Receiver on Victim Network**

Figure 8. Transmission Line Geometries: (A) Microstrip (B) Stripline

Backward crosstalk is present in both stripline and microstrip geometries (see Figure 8). (A way to remember which geometry is stripline and which is microstrip is that a stripline geometry requires **stripping** a layer away to see the signal lines.) The backward coupled amplitude is proportional to the backward crosstalk coefficient, the aggressor's signal amplitude, and the coupled length of the network up to a maximum which is dependent on the rise time of the aggressor's signal. Backward crosstalk reaches a maximum (and remains constant) when the propagation time on the coupled network length exceeds one half of the rise time of the aggressor's signal. Assuming the ideal ramp on the aggressor from 0% to 100% voltage swing, and the rise time on an unloaded coupled network, then:

$$Length\ for\ Max\ Backward\ Crosstalk = \frac{\frac{1}{2} \times Rise\ Time}{Board\ Delay\ Per\ Unit\ Length}$$

Since the GTL+ aggressor signals are non-ideal steps, and due to the presence of reflective loads on the GTL+ bus, we have used simulations to determine this length for maximum backward crosstalk, and found that it is associated with the 82450 PCIset fast corner falling edge which yields a maximum backward crosstalk length of about four inches.

Agents on the GTL+ bus drive signals in each direction on the network. This will cause backward crosstalk from segments on two sides of a driver. The pulses from the backward crosstalk travel toward each other and will meet and **add** at certain moments and positions on the bus. This can cause the voltage (noise) from crosstalk to double. Backward crosstalk will transition in the same direction as the aggressor's edge.

Forward crosstalk is absent in stripline topologies, but present in microstrip. (This is for the ideal case with a **uniform** dielectric constant. In actual boards, forward crosstalk is **nearly** absent in stripline topologies, but **abundant** in microstrip.) The forward coupled amplitude is proportional to the forward crosstalk coefficient, the aggressor's signal edge rate (dv/dt), and the coupled network's electrical length. The forward crosstalk coefficient is also a function of the geometry. Unlike backward crosstalk, forward crosstalk can grow with coupled section length, and may transition in a direction similar to or opposite to that of the aggressor's edge.

Since forward coupled signals travel in the same direction as the aggressor's, an agent on the GTL+ bus that has coupled sections on both sides of itself will not run the risk of the two forward coupled signals meeting and adding. However, unlike backward crosstalk, each signal will continue to grow as it passes through more coupled length before the aggressor's wave front is absorbed by the termination.

22

4.4.1.   CROSSTALK MANAGEMENT

To minimize crosstalk (and the "cost" of crosstalk) in terms of noise margin budget:

- Route adjacent trace layers in different directions (orthogonal preferred) to minimize the forward and backward crosstalk that can occur from parallel traces on adjacent layers. This reduces the source of crosstalk.

- Maximize the spacing between traces. Where traces have to be close and parallel to each other, minimize the distance that they are close together, and maximize the distance between sections that have close spacing. Routing close together could occur where multiple signals have to route between a pair of pins. When this happens the signals should be spread apart where possible. As an example: Two traces at 5/5 (5 mil lines with 5 mil spaces) for two separate 2 inch sections that are spaced at least one half of the rise time apart is better than having a single 4 inch section at 5/5 spacing. Also note that routing multiple layers in the same direction between reference planes can result in parallel traces that are close enough to each other to have significant crosstalk.

- Minimize the nominal board impedance (Z_0) within the GTL+ specification. For a given dielectric constant, this reduces the spacing between the traces and their reference plane, which reduces the backward and forward crosstalk coefficients. Having reduced crosstalk coefficients reduces the magnitude of the crosstalk.

- Minimize the dielectric constant used in the PCB fabrication. As above, all else being equal, this puts the traces closer to their reference planes and reduces the magnitude of the crosstalk.

- To avoid backward crosstalk at the extreme ends of the bus, connect the end bus agents (each end) to the termination resistors using microstrip traces of the same impedance as the rest of the GTL+ bus (this will have to be evaluated with other system constraints). For a given impedance, microstrip traces will have less crosstalk than stripline traces.

- Watch out for voltage doubling at a receiving agent, caused by the adding of the backward crosstalk on either side of a driver. Minimize the total network length of signals that have coupled sections. If there has to be closely spaced/coupled lines, place them near the center of the net. This will cause the point in time that voltage doubling occurs to be before the setup window.

- Route synchronous signals that could be driven by different components in separate groups to minimize crosstalk between these groups. The Pentium Pro processor uses a split transaction bus. This implies, that in a given clock cycle, the address lines and corresponding control lines could be driven by a different agent than the data lines and their corresponding control lines. If these two agents are at the opposite process corner (one fast and one slow), then separating the signal types will support the budget assumptions in Section 4.4.1.

- Minimize the cross-sectional area of the trace. This can be done by using narrower traces and/or by using thinner copper (1/2 ounce/ft^2 or 0.7 mil thick rather than 1 ounce/ft^2 or 1.4 mil thick). Note that the trade-off for this smaller cross-sectional area is a higher trace resistivity that can reduce the falling edge noise margin because of the increased I*R loss along the trace.

Simulation shows that 5/5 technology (5 mil lines with 5 mil spaces) will have excessive crosstalk between networks on the Pentium Pro processor bus. This is due to the lower voltage swing of GTL+, high frequencies (even with the controlled edge rate buffers) and likely long parallel traces.

4.4.2. POTENTIAL TERMINATION CROSSTALK PROBLEMS

The use of standard "pull-up" resistor networks for termination may not be suitable. These networks have a common power or ground pin at the extreme end of the package, shared by 13 to 19 resistors (for 14- and 20-pin components). These packages generally have too much inductance to maintain the voltage/current needed at each resistive load. Intel recommends using discrete resistors, resistor networks that have separate power/ground pins for each resistor, or working with a resistor network vendor to obtain resistor networks that have acceptable characteristics.

5.0. MORE DETAILS AND INSIGHTS

5.1. Textbook Timing Equations

The textbook equations used to calculate the propagation rate of a PCB are the basis for spreadsheet calculations for timing margin based on the component parameters. These equations are:

Equation 7. Intrinsic Impedance

$$Z_0 = \sqrt{\frac{L_0}{C_0}}$$

Equation 8. Stripline Intrinsic Propagation Speed

$$S_{0\_STRIPLINE} = 1.017 * \sqrt{\varepsilon_r}$$

Equation 9. Microstrip Intrinsic Propagation Speed

$$S_{0\_MICROSTRIP} = 1.017 * \sqrt{0.475 * \varepsilon_r + 0.67}$$

Equation 10. Effective Propagation Speed

$$S_{EFF} = S_0 * \sqrt{1 + \frac{C_D}{C_0}}$$

Equation 11. Effective Impedance

$$Z_{EFF} = \frac{Z_0}{\sqrt{1 + \frac{C_D}{C_0}}}$$

Equation 12. Distributed Trace Capacitance

$$C_0 = \frac{S_0}{Z_0}$$

Equation 13. Distributed Trace Inductance

$$L_0 = Z_0 * S_0$$

Symbols for Equation 7 through Equation 13 are:

- S_0 is the speed of the signal on an unloaded PCB. This is referred to as the board propagation constant.

- S_0 MICROSTRIP and S_0 STRIPLINE refer to the speed of the signal on an unloaded microstrip or stripline trace on the PCB.

- Z_0 is the intrinsic impedance of the line and is a function of the dielectric constant (ε_r), the line width, line height and line space from the plane(s). The equations for Z_0 are not included in this document. See the *MECL System Design Handbook* by William R. Blood, Jr. for these equations.

- C_0 is the distributed trace capacitance per unit length of the network.

- L_0 is the distributed trace inductance per unit length of the network.

- C_D is the sum of the capacitance of all devices and stubs divided by the length of the network's trunk, not including the portion connecting the end agents to the termination resistors.

- S_{EFF} and Z_{EFF} are the effective propagation constant and impedance of the PCB when the board is "loaded" with the components.

5.2. Effective Impedance and Tolerance/Variation

The impedance of the PCB needs to be controlled when the PCB is fabricated. The method of specifying control of the impedance needs to be determined to best suit each situation. Using stripline transmission lines (where the trace is between two reference planes) is likely to give better results than microstrip (where the trace is on an external layer using an adjacent plane for reference with solder mask and air on the other side of the trace). This is in part due to the difficulty of precise control of the dielectric constant of the solder mask, and the difficulty in limiting the plated thickness of microstrip conductors, which can substantially increase crosstalk.

The effective line impedance (Z_{EFF}) is recommended to be between 45Ω to 65Ω. Where Z_{EFF} is defined by the following equation:

Equation 14. Effective Line Impedance

$$Z_{EFF} = \frac{Z_0}{\sqrt{(1 + C_d/C_0)}}$$

Symbols for Equation 14 are:

- Z_0 = Nominal board impedance

- C_D = Sum of the capacitance of all devices and stubs (if any) attached to the network, divided by the length of the network

- C_0 = Intrinsic trace capacitance

To help in this calculation, values for Pentium Pro processor and 82450 PCIset input capacitance are listed below.

- Pentium Pro processor capacitance = 8.5 pF (including 0.5 pF for a socket)

- 82450 PCIset Capacitance = 6 pF (silicon and package)

5.3. Termination Values

Simulations of the Pentium Pro processor/82450 PCIset bus show that smaller values of R_T have better noise margin for the rising edge, and that larger values of R_T have better noise margin for the falling edge. R_T =47Ω is near the minimum that can be driven by the 82450 PCIset slow corner model. Systems with less than maximum total line length may be able to use smaller R_T. Verify with simulation if this is desired. The drive characteristic and maximum V_{OL} for the 82450 PCIset at the slow corner determines the minimum termination resistance value that can be used. R_{TT} + x% value should be used for rising edge simulations, and R_{TT} - x% should be used for falling edge simulations. (x% indicates the tolerance of the resistors used in the system.)

5.4. Reference Planes

Designs using the Pentium Pro processor require several different voltages. The following paragraphs describe some of the impact of three common methods used to distribute the required voltages. Refer to the *Pentium® Pro Processor Power Distribution System Design Guidelines* (Order Number 242764) for more information on power distribution.

The most desirable method of distributing these voltages is for each of them to have a dedicated plane. If any of these planes are used for an "AC ground" reference for traces to control trace impedance on the board, then the plane needs to be well decoupled to the system ground plane. This method may require more total layers in the PCB than other methods.

A second method of power distribution is to use partial planes in the immediate area needing the power, and to place these planes on a routing layer on an as-needed basis. These planes still need to be decoupled to ground to ensure stable voltages for the components being supplied. This method has the disadvantage of reducing area that can be used to route traces. These partial planes may also change the impedance of adjacent trace layers. (For instance, the impedance calculations may have been done for a microstrip geometry, and adding a partial plane on the other side of the trace layer may turn the microstrip into a stripline.)

The third method to distribute the power is to incorporate split power planes. This method is similar to the second method except that the multiple voltages share the conventional power plane layer. The power plane is split so that areas of the board needing separate voltages are divided to provide a separate voltage for each area. These areas still need to be properly decoupled, especially at the edges of each plane. The gap between the different power planes on a layer should be kept to a minimum. There will be a **negligibly small** impedance discontinuity in traces that cross the split and are using the power plane for a reference plane. It is very important when splitting planes that the GROUND plane not be split, as this could create significant length in the ground return path, adding noise in the system. Decoupling the different power planes, which are adjacent on the same layer may also be valuable for signals that use the split power planes for AC reference. The split plane method is not universally agreement upon as engineering good practice. If your company is not comfortable splitting planes, then you should use a different method.

5.5.   PCB Stackup

The type and number of layers for the PCB need to be chosen to balance many requirements. Many of these requirements are technical and include:

- Providing enough routing channels to support the minimum and maximum timing requirements of the components.

- Providing stable voltage distribution for each of the components.

- Providing uniform impedance for the Pentium Pro processor bus and other signals as needed.

- Minimizing coupling/crosstalk between the networks.

- Minimizing RF emissions.

- Maximizing PCB yield.

- Minimizing PCB cost.

- Minimizing cost to assemble PCB.

Design your PCB to meet these technical requirements.

5.6.   Clock Routing

The clock skew in Pentium Pro processor based systems must be kept to a minimum. (The calculations used in this document have a total clock skew of 900 ps, allowing 500 ps skew from the clock driver, 200 ps difference in the board propagation delay, and 200 ps of clock jitter.) To meet these specifications:

- Use a low skew clock driver.

- Have equal electrical length and type of traces on the PCB (microstrip and stripline may have different propagation velocities).

- Maintain consistent impedance for the clock traces.
    - Minimize the number of vias in each trace.
    - Minimize the number of different trace layers used to route the clocks.
    - Keep other traces away from clock traces.

- Lump the loads **at the end** of the trace if multiple components are to be supported by a single clock output.

- Have equal loads at the end of each network.

If the timing between a pair of components is exceptionally tight, and further reducing the clock skew between the components is desirable, then driving the pair of components from a single clock output with a short "T" close to the components may improve the timing. When supporting more than one component from a single clock output, the clock driver skew is eliminated.

The **ideal** way to route each clock trace is on the same single inner layer, next to a ground plane, isolated from other traces, with the same total trace length, to the same type of single load, with an equal length ground trace parallel to it, and driven by a zero skew clock driver. When deviations from ideal are required, going from a single layer to a pair of layers adjacent to power/ground

25

intel ®

planes would be a good compromise. The fewer number of layers the clocks are routed on, the smaller the impedance difference between each trace is likely to be. Maintaining an equal length and parallel ground trace for the **total length of each** clock ensures a low inductance ground return and produces the minimum current path loop area. (The parallel ground trace will have lower inductance than the ground plane because of the mutual inductance of the current flowing through the clock trace.)

The number of components that will need to receive a system clock is dependent on the system size. The following shows the number of clocks needed by each of the Intel bus agents:

- One clock per Pentium Pro processor.
- One clock per 82454 PCI Bridge.
- One clock per 82451/82452/82453 Memory Controller - six clocks per set (DP, DC, and 4 x MIC).