

# Clock Design and Measurement Issues in Pentium™ Systems

Design difficulties in producing a statistically stable 66-MHz Pentium system are reviewed. The information is pertinent to many other new, high-speed processors as well. A new, more informed approach to designing well-timed systems in this performance class is proposed. Measurements that support this approach are examined, particularly those made with the HP 8133A pulse generator.

by Michael K. Williams and Andreas M.R. Pfaff

Clock rates in all classes of computational systems, from PCs to supercomputers, have been escalating exponentially for years. Computational systems formerly considered simpler have come to run at speeds that were previously found only in more complex and aggressive systems. Before this happened, systems at the simpler end of this spectrum (PCs and workstations) operated at clock rates that don't present very difficult clock distribution and reception problems.

Recent introductions of new processor types have given PC and workstation system designers new chips and chipsets that enable system designs that deliver much higher levels of performance.<sup>1</sup> Most of these devices employ internal structures that come directly from the world of mainframes and supercomputers: pipelining, 64-bit data buses, on-board floating-point units, instruction prefetching, and sophisticated caching schemes. Many of these processors are summarized in Table I. These new device families include Intel's Pentium processor, Digital's Alpha, the Apple/IBM/Motorola PowerPC, and others. These ICs have clock rates that range from tens to hundreds of MHz. Some are expected eventually to exceed 1 GHz.

**Table I**  
**Some New Processor Types and their Clock Rates**

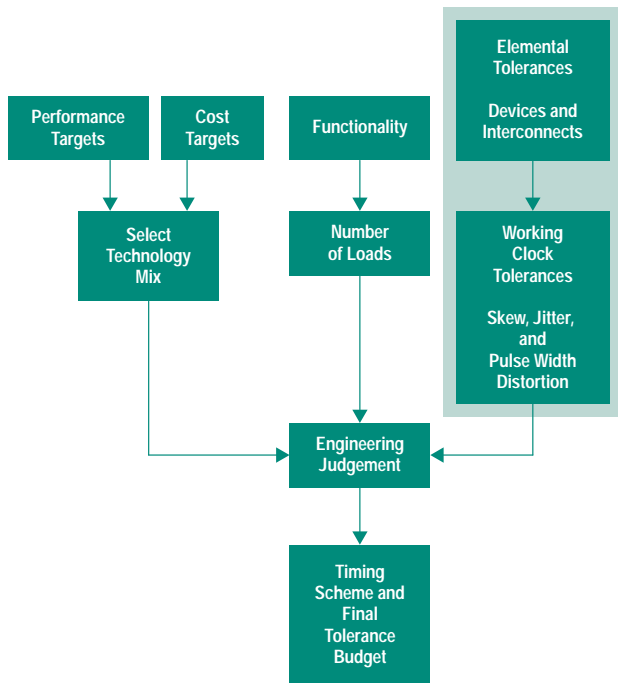
Manufacturer	Processor	Clock Rate
Intel	Pentium	60 and 66 MHz
Intel	P54C	99 MHz
Intel	486	66 and 99 MHz
Apple/IBM/Motorola	PowerPC	80 MHz
Digital Equipment Corp.	Alpha	> 100 MHz
MIPS	R4400SC	150 MHz

With all of the sophisticated internal structures and faster operating speeds comes a price to be paid by the design team. Specifically, successful system design at these speeds requires very careful consideration of many mechanisms, such as timing and pulse fidelity, that are unimportant at lower speeds (16 to 33 MHz).

Pulse fidelity, sometimes referred to as signal integrity, is that part of high-speed digital design that is concerned with managing the analog effects that prevent signals from being reliably recognized at their destinations. This includes ensuring that edges arrive at their loads with proper edge speeds and proper shapes, and controlling the various types of noise (crosstalk, EMI, reflections, ground bounce, etc.) that can cause unreliable or false triggering. The extent to which these issues are important has increased dramatically in PC and workstation designs. Wider buses and faster clocks and edges (higher waveform spectral content) are the primary sources of these problems. The classic discussion of all of these analog effects can be found in reference 2.

Timing, or clock distribution and reception, is the other critical facet of design in these faster systems, and is possibly the most significant and least well-understood aspect of the design. Timing environment design is the process of specifying how the clock is to be distributed and received throughout the system such that the state architecture is reliably synchronized. Reliable means that synchronization is guaranteed on every cycle of every copy of the design that is manufactured, despite the presence of a variety of statistical tolerancing mechanisms (skew, jitter, etc.) which reduce the precision with which the clock can be delivered. These tolerancing mechanisms are described in detail on page 70. When reliable synchronization is ensured by sound design practices, the design is said to be statistically stable.

The question of exactly how to ensure this statistical stability is one that each design team must face as they adopt these new devices into their designs. Success at answering this question brings with it higher yields, fewer design turns, and the elimination of extremely subtle timing failures. Methods for doing this, while relatively new to designs at the workstation and PC level, have been commonplace in the design of higher capability systems (mainframes and supercomputers) for many years. A descriptive term for the approach that is common to all of these methods is informed design.

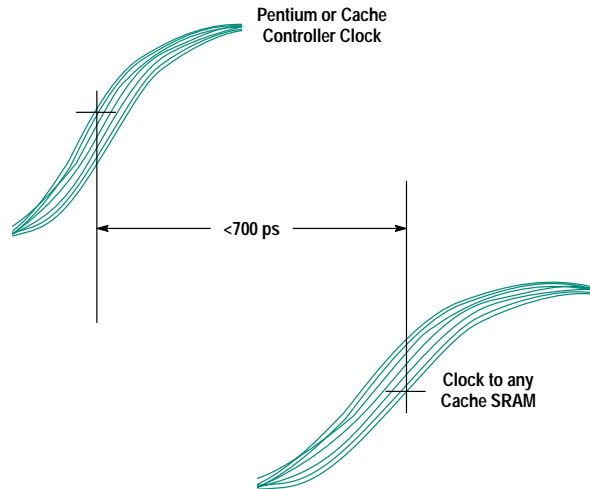


**Fig. 1.** Timing environment design process. Design-specific and/or unrated parametric information must be incorporated into the engineering decision making process from the outset.

Two results are produced by an informed approach to the design of a timing environment. The obvious one is a specification of a clock distribution scheme. Equally important, however, is a detailed knowledge of the tolerance on the arrival time of any clock waveform emerging from any output of any copy of that network, on any cycle of its operation. This knowledge, that is, the tolerance budget, is used by the timing verification software to determine if the rest of the system is correctly timed. Obviously the quality of this determination is a function of the quality of the tolerance budget. Informed design, as it applies to timing, can be viewed as the practice of ensuring that all of the mechanisms that contribute to the overall tolerancing of the clock have been accurately assessed.

Measurement is used to characterize devices and printed circuit board processes to see how they tolerance. This device-level tolerance data is used to compute the overall tolerance on the system clock. And this system-level tolerance is used within the timing verifier to ensure the creation of a statistically stable system. Fig. 1 illustrates where device-level parametric data fits into the overall decision making process.

In this article we examine some of the difficulties a designer will encounter in specifying, analyzing, and verifying a timing scheme for a 66-MHz Pentium system. This falls in the lower speed range for the new round of processors. However, ultratight timing specifications coupled with the currently available implementation technologies (clock buffers, printed circuit boards, etc.) make 66-MHz Pentium systems among the most difficult from a timing environment design perspective. We will see, for example, that the timing within the CPU complex (processor, cache controller, and cache RAMs) is very sensitive to clock jitter. This sensitivity, and others,



**Fig. 2.** The difference in arrival time between either the Pentium clock or the cache controller clock and the clock arriving at any SRAM must be less than 700 ps in every system on every cycle.

make 66-MHz systems ideal for the informed design approach. Furthermore, the issues and methods presented are general and extend to other processor types as well.

### Pentium Characteristics and Requirements

An understanding of the difficulties of distributing a clock within a Pentium design must begin with an understanding of Pentium timing requirements. Our discussion of this aspect of the design will be in summary form, and the reader is referred to the Intel documentation<sup>3-6</sup> for a more complete discussion of requirements. Also, reference 7 discusses both the requirements and the various design decisions in much deeper detail than can be done here.

A variety of system configurations are supported by the Pentium processor. The clock rate can be either 60 or 66 MHz. The system can use either no second-level caching, or it can have 256K-byte or 512K-byte cache memories. Systems with 256K-byte caches can operate at either clock rate, while 512K-byte systems are limited to 60 MHz. A "typical" Pentium design is expected to operate at 66 MHz and have a 256K-byte second-level cache. For such systems, there are 12 clock loads within the CPU complex. Depending upon how the rest of the system is designed, the total number of clock loads will typically be in the range of 15 to 20, although in some server systems, this number can range an order of magnitude higher.

The Pentium specification dictates that the arrival times of the clock at the processor and at the cache controller never differ by more than 200 ps. It also states that the difference in arrival times between the processor and any cache memory, and the cache controller and any cache memory, can never exceed 700 ps (Fig. 2). These tolerance specifications must be met at 0.8, 1.5, and 2.0 volts. In any design, there will be other tolerance requirements that state how much difference in arrival time is permitted between clocks at loads within the CPU complex and clocks at loads external to it (external loads). These requirements will always be directly determined by the design itself. However, the overall tolerance budget will usually be driven by the timing within the CPU complex.

## Tolerance Mechanisms in Clock Distribution Networks

As described in the accompanying article, we are attempting to guard against a number of statistical tolerancing mechanisms, such as skew and jitter, that reduce the precision with which a clock signal can be delivered. Here we present an overview of these mechanisms.<sup>1</sup> For the purpose of considering system timing issues, it is useful to separate the system state architecture into a timing environment and a computation environment (see Fig. 1). The boundary between these two parts of the system is composed of the system state devices. Except for segment delay times and communications locality, we don't address the details of the computation environment here. The timing environment can be further broken down into three sections: the clock or phase generator, the clock distribution network, and the memory elements.

The clock generator supplies the signal whose edges eventually dictate when switching occurs throughout the system. The clock generator determines the period, pulse width, number of phases, and relative phase separation of the clock waveform. The primary attributes of the generator to be specified at design time are the waveform period and stability or jitter. For systems that use a processor chip, the period is usually specified by the manufacturer of the processor. Instability (jitter) in the waveform emerging from the generator detracts from either performance or reliability. Beyond these, there are frequently secondary issues and features that contribute to system testability—frequency and duty cycle adjustability, overtone suppression, modes (burst, single-step, fast, and slow), scan-path drive and timing, and others.

The state devices are flip-flops, latches, or memory devices of some type. New devices with enhanced testability features are appearing more frequently. The state devices play an important role in determining the low-level timing constraints in that their setup, hold, and minimum pulse width requirements must be satisfied at full clock speed.

The clock distribution network is a network of buffers and interconnects that conveys the clock signal to the clock consumers. It is responsible for fanout amplification and is generally tree-structured. In simpler systems, all of the fanout can occur in a single buffer. In larger systems, thousands of copies of the clock can be produced, requiring many levels of buffering (12 to 15 levels in some supercomputers). From a timing perspective, the ideal situation is for all of the copies of the clock waveform to emerge from the leaves of the clock distribution network at the same moment. However, the devices (both buffers and interconnects) that make

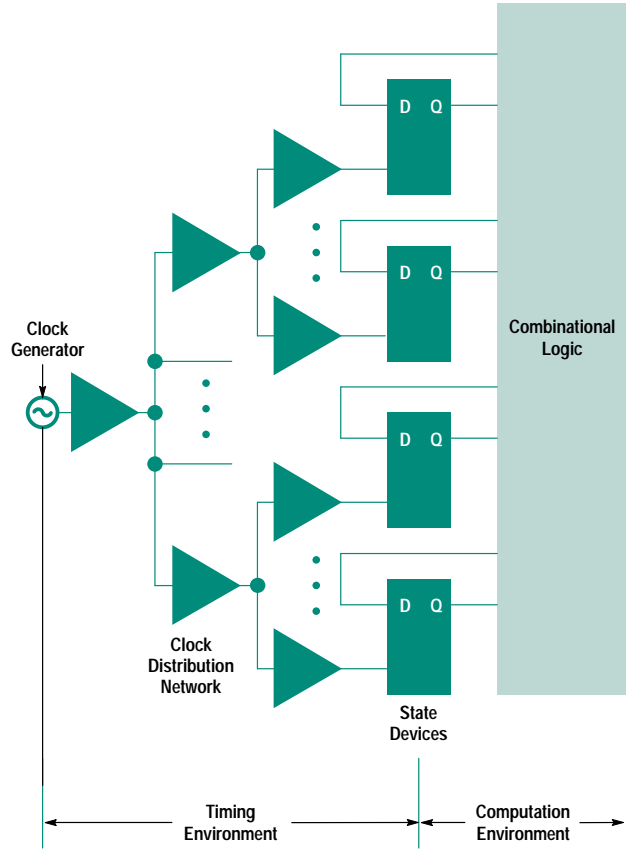


Fig. 1. State architecture model. Any synchronous digital system can be decomposed into a timing environment and a compute environment. The design issues specific to the timing environment are becoming critical in PC and workstation designs.

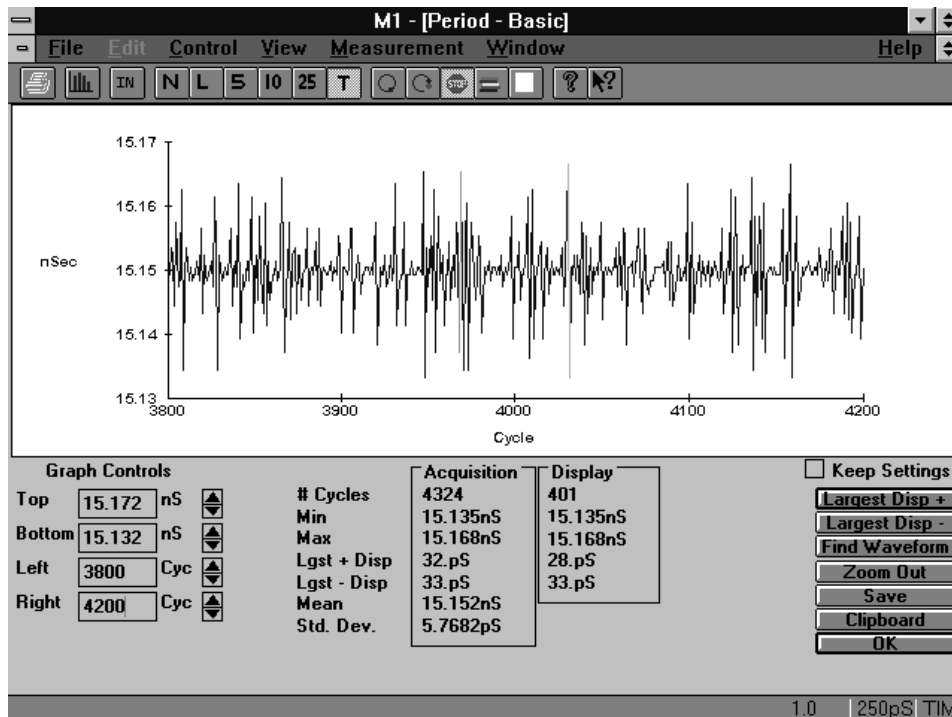


Fig. 2. There are a number of metrics of jitter. This measurement shows the cycle-to-cycle variation in the period of a 66-MHz clock. This was made using the Amherst Systems Associates M1 time interval measurement software, which analyzes digitized waveform data from an HP 54720D oscilloscope for jitter in a variety of ways.

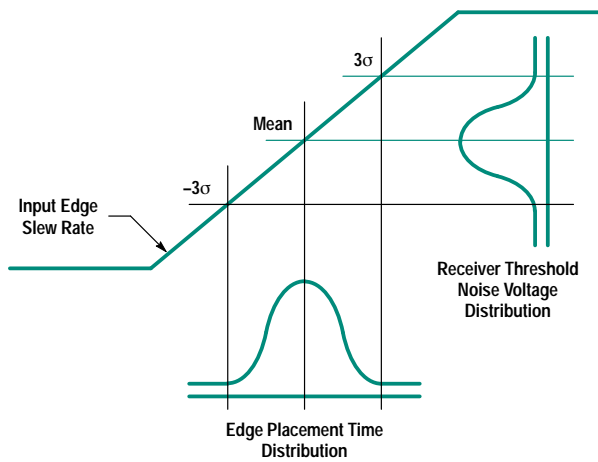


Fig. 3. Jitter, as it occurs in clock buffers, is generally the result of noise in the power environment (return currents, image currents, etc.) modulating the switching threshold of the buffer.

up the paths through the clock distribution network have a statistically distributed delay. These distributions can be time-invariant (static) or time-variant (dynamic). An example of a statically distributed tolerance is skew in clock buffers. This is the variation in delay either from pin to pin in a single package or from part to part. Interconnects can also exhibit tolerancing. This is most easily thought of as a variation in the propagation rate of several picoseconds per inch (10 to 40 ps/in). Interconnect tolerancing is frequently a source of unanticipated timing failures.

An example of a dynamically distributed tolerance is jitter. The placement in time of a waveform edge that has jitter varies from one cycle to another. It can be thought of as having a period that changes from one cycle to the next. Fig. 2 shows an example of this variation. Jitter can be added to the clock waveform in two places: at the generator or in the buffers. At the generator, jitter can occur through either internal noise or dynamic temperature or supply voltage instabilities. Jitter added in the clock buffers is caused primarily by noise in the power environment (return and image currents in power planes sweeping past power and ground pins, etc.) causing time-varying shifts in the device's switching threshold. This is illustrated in Fig. 3. Note that jitter (an expansion of the distribution of the edge placement) is increased when the noise voltage is increased or the edge rate of the signal arriving at the buffer is decreased. The management of jitter at consistent and acceptably low levels is perhaps the single greatest challenge for designers of systems that incorporate many of the new high-performance processors. A more in-depth discussion of jitter measurement and management can be found in references 1 and 2.

There are also statistical variations in how two identical parts are used. For example, one system may run a little warmer than another, another may have a little more noise in the power environment, and so on. Some of these tolerances

are time-variant and some are not. As shown in Fig. 4, these device-level distributions can be statistically combined† to give a system-level distribution on the path delays in the clock distribution network. This system-level path delay distribution has a mean value that is sometimes called the nominal delay. By statistically combining the individual nominal delays along the path, one computes the nominal delay for that path.

When using the nominal delays, it is important to keep in mind that there is actually a delay distribution. This means that even if every path in the design is specified to be identical, when the product is manufactured there will be product-to-product variations in the propagation delay of any given path, there will be path-to-path variations within any given machine, and there will be cycle-to-cycle variations on a given path in a given machine. The result is that one must design the system in a manner that both suitably minimizes these tolerances and consciously considers the fact that the tolerances will always be nonzero. The design is said to be statistically stable when it has this characteristic.

When the tolerances in the system accumulate beyond the value anticipated by the designer, the design is said to be statistically unstable. In statistically unstable designs, some small fraction of the manufactured systems will experience timing failures despite the absence of any physical defects. In these systems, the clock can arrive at times other than the designer anticipated, and this can mean that one or more of the state device timing requirements (setup time, hold time, or minimum pulse width) will be violated.

Violations of any of the device-level timing requirements can result in statistically unreliable switching at the state devices. This can cause unpredictable deviations in normal system-level behavior. These faults can be extremely difficult and time-consuming to isolate. In fact, the failure modes exhibited by systems with internal timing problems are easily among the most difficult to diagnose using conventional troubleshooting methods. It is frequently necessary to employ an analytic approach to find these faults in any sort of efficient manner. These failure modes include:

- Intermittent or nonrepeating
- Low frequency of occurrence (minutes through weeks)
- Migration of the symptom location through the system
- Hibernation (failures occur as device parameters change slightly with age)
- Statistical.

#### References

1. M.K. Williams, "Distortion and Tolerance Mechanisms in High-Speed Clock Delivery," *Proceedings of the 1993 Hewlett-Packard High-Speed Digital Symposium*, pp. 4-1 to 4-41. Also available as Application Note ASA 93-1 from Amherst Systems Associates.
2. M.K. Williams, "Design Trade-offs in High-Speed Clock Distribution and Reception," *Proceedings of the 1993 Hewlett-Packard High-Speed Digital Symposium*, pp. 6-1 to 6-34. Also available as Application Note ASA 93-2 from Amherst Systems Associates.

† The combination of these subordinate distributions is more complicated than direct addition. It must also take into account correlations that occur in such tree-structured circuits, and other related mechanisms called tracking effects.

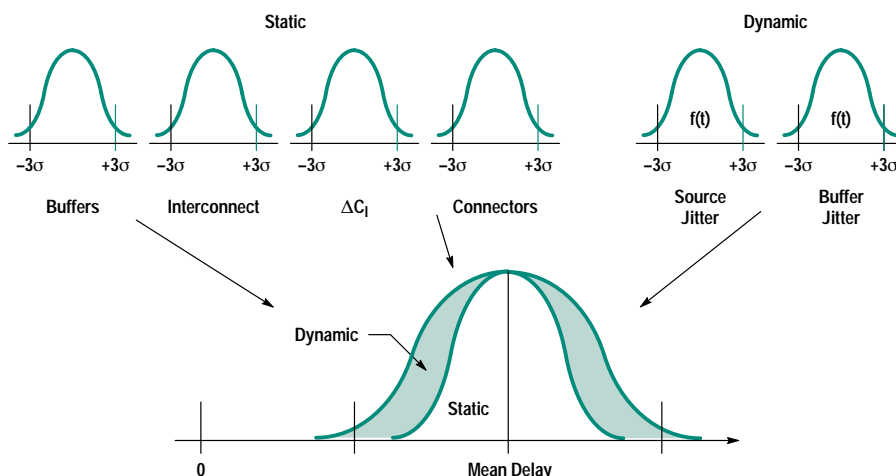


Fig. 4. A variety of tolerancing mechanisms contribute to the uncertainty in the arrival time of the clock edge at any clock load. Generally the only one that is available in catalogs or data sheets is the buffer tolerancing.

In general, the difficulty of any particular timing environment design can be estimated from two facets of the design: the number of clock loads and the amount of allowable clock tolerance, expressed as a fraction of the period. One threshold of difficulty occurs at about ten board-level clock loads† and tolerance budgets that are less than 10% or so of the period. For the typical 66-MHz system we have assumed that the loading (15 to 20 clock loads) ranks it as somewhat difficult. The tolerances within the CPU complex of 200 and 700 ps represent 1.3% and 4.7%, respectively, of the 15-ns cycle time. This represents a very challenging timing requirement. Table II summarizes Pentium clock tolerancing for various system configurations.

**Table II**  
Clock Tolerancing and Loading within the Pentium CPU Complex

Clock Speed (MHz)	Cache Size (bytes)	Tolerance (ps)	Number of Loads
60 or 66	None	N/A	1 (CPU only)
66	256K	700	12 (CPU, cache control, 10 SRAM)
60	512K	800	20 (CPU, cache control, 18 SRAM)
60	256K	800	12 (CPU, cache control, 10 SRAM)

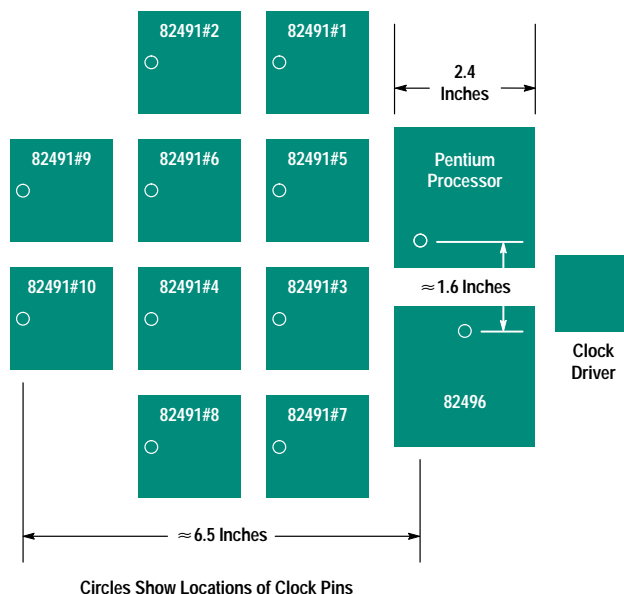
### Design Example

In this section, we attempt to impart some insight as to where the tolerance budget comes from. We illustrate some of the aspects of the design that are major drivers of this budget. Our goal is to show the importance of having complete and accurate design information at every step of the process. However, the process of completely and precisely evaluating each component of that budget is complicated, and is beyond the scope of this article. The interested reader is directed to References 8 and 9 for a more in-depth discussion of the design decisions presented here.

Before describing the design, we encourage the reader to adopt the view that every design decision that pertains to clock paths should be made very carefully and considered from the perspective of how that decision impacts the tolerancing of the clock. It is a fact that every physical design decision (buffer selection, transmission line geometry and impedance, termination, grounding schemes, etc.) that relates to the clock paths impacts clock tolerancing.

**Preliminary Decisions.** Our example design here is a fully synchronous 66-MHz system with a 256K-byte second-level cache. It is based on the use of the Intel 82496 cache controller and the 82491 cache SRAMs. In this discussion, we

† Most clock buffers have 10 or fewer outputs. When the number of loads in a design exceeds this level, either multiple loads must be clustered on each output or a multichip solution is required. The former increases the load capacitance range ( $C_{max} - C_{min}$ ) any output can see, which increases the difference in arrival time between the fastest and slowest conditions. The latter solution, using additional devices, increases cost and the length of the clock paths, which in turn increases the opportunity for tolerancing to occur in the clock.



**Fig. 3.** Intel suggests this placement for use with their second-level cache chipset.

make almost no assumptions†† about circuitry beyond the CPU complex, since the design challenge lies with the clocks within the complex. Beyond this, we assume the Intel suggested device placement (Fig. 3). Placement must be very carefully considered for these devices not only from a clock-distribution perspective, but also from the perspective of the times of flight of all of the data, address, and control signals. These times are very precisely specified in the Pentium specification.

As stated earlier, the typical design is expected to have a total of 15 to 20 board-level clock loads. To minimize clock tolerancing caused by variations in the load capacitance, it is desirable to drive the system in a point-to-point fashion. This means one clock load per clock buffer pin. We have selected a 20-output static clock buffer for this role. It has a pin-to-pin tolerance (skew) of 500 ps.

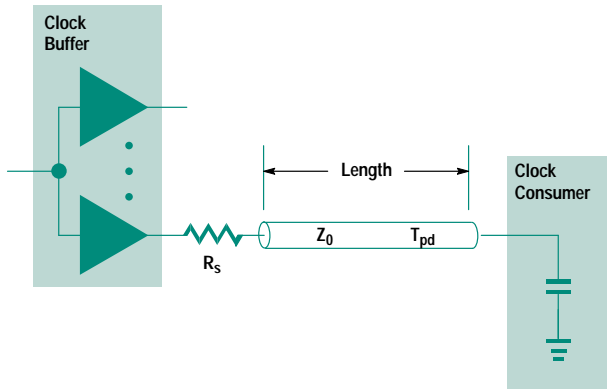
The interconnect for the design being described here was also very carefully considered. It was decided to route all clocks in microstrip (typically less tolerancing than stripline because of a faster propagation rate). An interactive field solver was used to design the microstrip. The resulting propagation rate is 146.4 ps/in.

**Predicting Actual Clock Tolerances.** A good way to begin is to do an inventory of where the clock loads in the system are expected to be placed and get as much information as possible about what types of loading they will present. Intel provides very complete pi-models††† for all of the pins on the devices within the Pentium CPU complex (also known as the “optimized interface group”). These models provide minimum, maximum, and typical values. The minimum and

†† We will make reference to worst-case external clock loading when we do the load/placement inventory.

††† A pi-model is a standard ac model of an input pin, consisting of a parallel inductor, a series capacitor, and another parallel inductor.





**Fig. 4.** Most of the clock nets in our design can be viewed as simple series-terminated transmission lines driving single capacitive loads.

maximum ratings permit an accurate determination of the range of distortion delay† that will occur at any pin type. Usually, the best a designer can hope for in terms of published parametric pin data is typical input capacitance values. This only permits estimation of the typical distortion delay, not the range.

When the clock load inventory is completed, the designer will know approximately how far most loads are from the clock buffer and which buffers are most heavily loaded (typical). This information lets the designer estimate how late the slowest load typically reaches threshold. From this value, the other clock paths can be adjusted (e.g., by serpentine) to align their typical delays with the slowest one in the system. For this design, the result of the inventory is that the largest mean path delay is 1586 ps. The delay ranges for all of the other paths in the system are centered on this value.

Since we used point-to-point distribution for most†† of the clock paths, the general structure of the clock nets is shown in Fig. 4. The general formula for computing the tolerancing at this point is:

$$\text{tolerance}_{\text{net}} = \text{skew}_{\text{int}} + \text{skew}_{\text{ext}} + \text{jitter},$$

Skew<sub>int</sub> is the intrinsic skew, that is, the delay variation of the buffer (pin-to-pin in this case). For our buffer, this is 500 ps. Skew<sub>ext</sub> is the extrinsic skew, that is, the delay variation along the net. Jitter is the peak value, rather than rms or some other statistical jitter metric.

Extrinsic skew is not a single mechanism. It can be broken down into two major components:

$$\text{skew}_{\text{ext}} \cong \Delta L T_{\text{pd}} + \text{tol}_{\text{mfg}},$$

where  $\Delta L T_{\text{pd}}$  is the variation in the propagation delay of a signal down a loaded transmission line. It takes into account

† Distortion delay is that component of the delay that a clock edge experiences as it arrives at the load and enters the die. The parametrics of the pin, as represented by the pi-model, act as a filter. The more the high-end spectral content of the edge is attenuated, the more the slope of the edge is reduced, adding delay in the amount of time it takes the waveform to climb to threshold. What is important with the clock is not absolute delay, but delay variation, so when the parametrics vary more widely, more variation (tolerance) can occur in the timing at the pin. This variation is often referred to simply as load capacitance variation.

†† Because of a 200-ps allowable difference in arrival time between the processor and the cache controller, these two loads are actually clustered at the end of a single clock net. This is discussed in much more detail in Reference 4.

the range of loads seen at the end of a net. Tol<sub>mfg</sub> is the manufacturing tolerance of the interconnect. It ranges from about 1 ps/in to about 50 ps/in times the length of the interconnect.

$\Delta L T_{\text{pd}}$  can be computed from:

$$\Delta L T_{\text{pd}} = L T_{\text{pd}} \left( \sqrt{1 + \frac{C_{\text{lmax}}}{L C_0}} - \sqrt{1 + \frac{C_{\text{lmin}}}{L C_0}} \right),$$

where L is the length of the net in inches and  $T_{\text{pd}}$  is the unloaded propagation rate in ps/in.  $C_{\text{lmax}}$  and  $C_{\text{lmin}}$  are the maximum and minimum values of load capacitance. To compute the difference in arrival times between two clock loads, these values will be from different pins. Equivalent values for  $C_l$  can be computed from pi-models.  $C_0$  is the intrinsic capacitance of the net.

Following this general format for computing the tolerances, we can compute a worst-case difference in arrival times of the clocks to the cache controller and the cache RAMs.

$$700 \text{ ps} \geq \text{skew}_{\text{int}} + \Delta L T_{\text{pd}} + \text{tol}_{\text{mfg}} + \text{jitter}.$$

Plugging in what we computed,

$$700 \text{ ps} \geq 500 + 90 + 60 + \text{jitter},$$

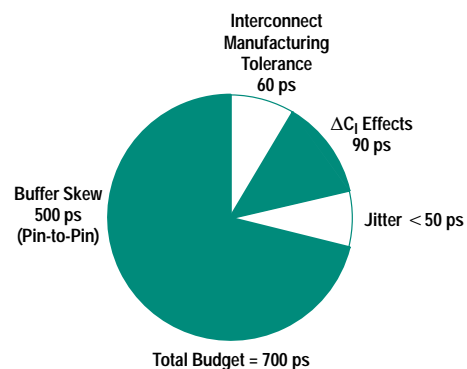
which gives us the constraint on clock jitter:

$$50 \text{ ps} \geq \text{jitter}.$$

The overall tolerance budget is summarized in Fig. 5. The jitter constraint is very aggressive for PC and workstation class computers. Normally, this constraint is a full order of magnitude higher. Keeping noise levels low enough to meet this constraint will present some unique measurement requirements, as we shall see in the next section.

### Incorporating Measurement Information

We have, thus far, described a number of the more challenging issues that must be addressed in producing a 66-MHz Pentium design with statistically stable timing. We have attempted to emphasize the importance of employing informed design practices. The basic tenet of these practices is that important design decisions (e.g., timing verification) are based upon deliberately and accurately gathered design information. The better this design data is, the better the design decisions that are based upon this data. In the case of timing,



**Fig. 5.** After accounting for all of the tolerancing mechanisms we have little or no control over, our typical Pentium design can tolerate approximately 50 ps of jitter.

we are talking about all of the low-level tolerance information required to compute an accurate tolerance budget.

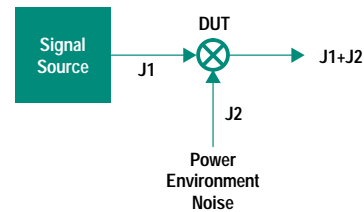
As noted on page 70, the only significant component of the tolerance budget that can be found in data sheets is the buffer tolerance. All of the other low-level tolerance information must be determined through measurement. It cannot be generated for a design at one company and shared with others. The tolerance information is determined by the specific methods and devices employed in a particular design, and each design is unique in these regards.

Perhaps the most notable measurement information relates to the very tight jitter allowance. An upper limit of 50 ps will require exploration and experimentation of various design alternatives (device placement, bypass filtering, ground plane cuts, etc.) to determine their exact effect on jitter. Jitter caused by switching noise will be first-order sensitive to clock buffer placement. And this may involve some measurement activities that are very new to PC and workstation design activities.

Measurement is usually viewed as a stimulus and response process. Stimulus gear includes pulse and function generators and waveform synthesizers. Response gear includes oscilloscopes, time-interval analyzers, spectrum analyzers, and so on. Response is unquestionably important when the measurement of very low-amplitude jitter (10 to 50 ps) is being performed. However, one of the less well-understood facets of precision measurement relates to the specification of stimulus gear and methods for these measurements. In the high-speed PC and workstation designs we're discussing here, stimulus issues center primarily in two areas: characterization activities and applications calling for an alternate, adjustable clock source. As we shall see, the precision of the waveform submitted to a device under test has a significant impact on the quality of the design data that results from the measurement. In this section, we discuss a number of measurement methods that apply to these two areas.

**Instrumentation Issues.** For all of the measurements described in this section, the way the measurement is made and the quality of the instrumentation employed in the measurement are issues of genuine importance. The importance of precision cannot be underestimated. Any tolerance on the measurement must also be included in the final tolerance data. That, of course, means that measurement tolerance directly detracts from system performance.

The very low levels of jitter allowed in the systems we're discussing makes the measurements very challenging. For example, the waveform timing uncertainty or jitter of the source (pulse generator) must be much less than the jitter of the device under test (DUT). There are two reasons for this. The first is to avoid corrupting the measurement. A good rule of thumb is to try to keep stimulus jitter an order of magnitude below what you are expecting to measure. In that way, the majority of the jitter measured is what occurs within the DUT. The second reason for low source jitter is that the tolerance budget establishes an upper bound on the amount of jitter permitted on the clocks distributed to the loads, and the total jitter on those clock signals includes



**Fig. 6.** When the device under test is a static clock buffer it acts as a jitter mixer, combining noise-induced jitter with jitter coming in from the signal source. For tight systems like Pentiums, it is clear that both the source jitter and the power environment jitter must be kept to a minimum to permit reliable testing and characterization.

jitter from the signal source. Consider, for example, Fig. 6, which shows a clock buffer being driven by an external signal source. The buffer can be viewed as a "jitter mixer," that is, the total jitter transmitted to the clock loads is the sum of the jitter that the buffer adds because of noise (J2) and the jitter on the externally generated waveform that drives the buffer (J1). If J1 is significant with respect to J2, it will swamp the measurement. Furthermore, if J1 + J2 exceeds the jitter limit, the system will not function properly during the measurement. This brings up an interesting point. If you plan to make these sorts of measurements and use an external signal source, you must account for the jitter of whatever signal source you may use in the tolerance budget.

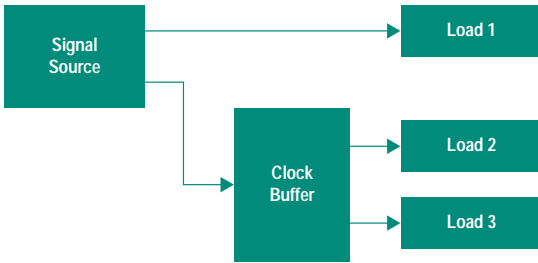
In our Pentium design, our 50-ps allowance for jitter means that if we plan to use a signal source with 15 ps of jitter, we should limit jitter in the system to less than 35 ps. A 10-ps source will permit the design to work with 40 ps of in-system jitter. However, to use a source with much more than 15 ps of jitter means greater design difficulty in minimizing in-system jitter,<sup>†</sup> and increasing difficulty in interpreting system-level jitter measurements because of the difficulty of determining how much of the jitter is from the source and how much is from the system.

**Substitute Clock Measurements.** The most common reason for using an external source to drive the clock is to do system-level timing margin testing and verification. The fundamental question behind these measurements is how sensitive the system is to imperfect device timing. In other words, the sensitivity of the system to variations in parameters such as frequency, duty cycle, skew/jitter, or phase separation<sup>††</sup> is being determined.

Fig. 7 illustrates a measurement setup for one type of margin testing. Specifically, the setup permits investigating how sensitive load number one is to various types of parametric tolerancing by controllably varying the parameters of the waveforms produced by the signal source. For example, by advancing the phase of the waveform to load 1 and noting where unreliable switching occurs, and then retreating the phase of load 1 and again noting where unreliable switching

<sup>†</sup> It is probably a useful rule of thumb that when the stability requirement of the clock in a mass-produced computer system exceeds the stability found in precision pulse generators, the requirement is perhaps too aggressive.

<sup>††</sup> Phase separation is a parameter in systems with multiphase clocks. It is the minimum separation between an edge in one clock phase and an edge in another.



**Fig. 7.** The margin available at a specific load in a system can be examined by driving that load from a two-channel signal source and carefully adjusting the relative phase of the channels until unreliable switching is detected.

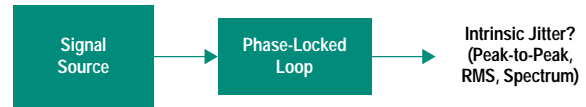
occurs, the operating limits of the load 1 clock can be estimated.† It is common during the course of a design to need to adjust or ascertain the tolerance at a specific point in the system (i.e., a point tolerance). For example, the clock to a particular point in the system may have to be forced to be earlier, or less tolerated than originally assumed, because some aspect of the segment bounded by that state device has changed.

Another critical verification is that the jitter that actually occurs in the final hardware is acceptably low. The designer starts with an assumption of what can be achieved. However, accurately predicting jitter is difficult, even with “representative” assessment boards and experiments. Front-end assessment of jitter is important, but only an estimate can be produced without final hardware. Only the final hardware will have the actual switching activity, the actual return and image currents, and the actual paths and obstacles that steer these currents. To verify jitter, it’s necessary to measure it in a variety of locations and switching conditions.

One other significant application of an alternate, adjustable clock source occurs during debugging. The external clock can bypass either the clock source or paths through the clock distribution network to permit the investigation of a timing problem at the loads. The benefit of this, of course, is that a defective source or clock distribution network can be bypassed or the loads supplied with a clock with jitter reduced to below normal operational levels.

**Characterization Measurements.** The verification activities described in the previous section are intended to determine how sensitive the system is to imperfect timing. Device characterization measurements ask the question from the other side—how imperfect might the timing be? This class of measurements includes fixtured device measurements.

For example, phase-locked loop clock buffers are basically active signal sources. As such, they have jitter of their own (intrinsic jitter). To characterize various facets of this jitter (cycle-to-cycle deviation, phase noise, jitter spectrum, settling time, susceptibility to power supply noise, etc.) without corruption from some external effect, it is important to supply the device with a stable reference signal and a clean power environment (Fig. 8). Note that a measurement of the intrinsic jitter of a well-fixtured phase-locked loop clock buffer does not establish how the device will perform in the

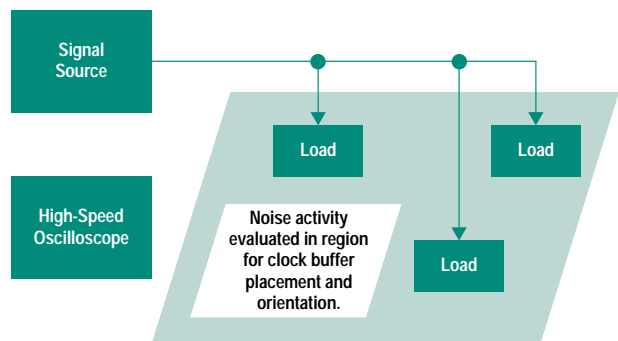


**Fig. 8.** A stable reference signal should be supplied while characterizing a phase-locked loop.

system. Instead, it establishes an upper bound on stability. The live system will have a noisier power environment and less stable reference signals. The spectrum of these disturbances will not likely be consistent in every application, nor will it be easily predicted. The behavior of the phase-locked loop is affected in a very complex way by the superposition of these various external processes.

Another measurement process that involves the clock buffers is the determination of so-called “derating factors.” The published tolerances for clock buffers include not only process and manufacturing variations, but also consideration that the parts may be operated across a range of temperatures, operating voltages, and loadings. The system designer has no control over how buffers vary because of process variations, but does have control over the range of temperature, voltage, and loading in the design, and may wish to attempt to remove that part of the buffer tolerance that is attributable to these margins on the operating ranges. A series of fixtured device measurements made while carefully varying some environmental variable can yield estimations of how much of the published tolerance is attributable to the environmental operating range.

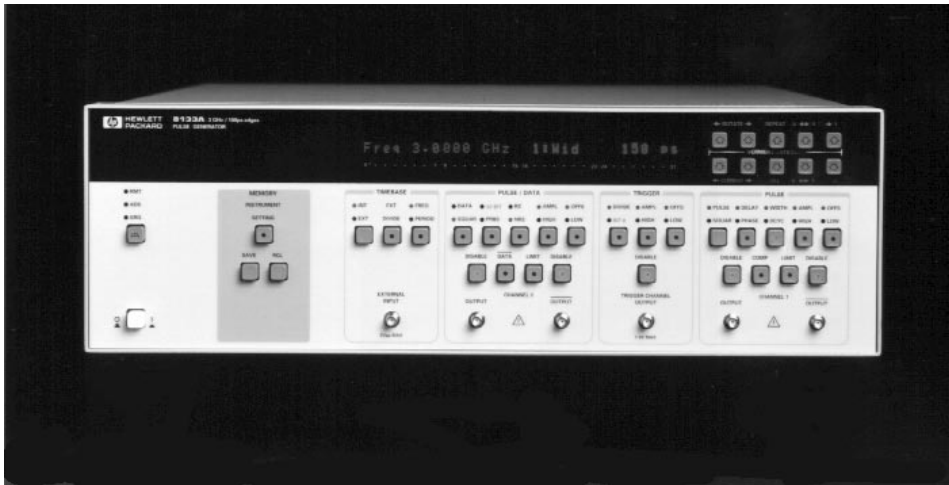
There is also a role for board-level measurements. As stated earlier, the jitter of a clock buffer (phase-locked loop or static) is affected to a large extent by the level of noise in the power environment. More specifically, it is determined by the noise where the device power and ground pins attach to the power and ground planes. This noise is caused by image and return currents in those planes. There are places on the board where this noise is significantly higher than other places. Furthermore, the gradient of these changes can be fairly tight, with quiet points existing millimeters from points that carry high image currents. All this means that buffer placement and orientation on the board have an impact on clock jitter. It is possible to evaluate approximately where the quiet locations are on a “technology board.” Fig. 9 shows such an experiment. The external signal source is used to drive a representative collection of switching gates.



**Fig. 9.** By examining the power environment noise in the region where the clock buffer is expected to be placed, the quietest power and ground connection points can be determined.

† Of course, this only shows the sensitivity of that particular system. However, that result can then be guardbanded to take into account what might happen across a larger population of systems.





**Fig. 10.** The HP 8133A 3-GHz pulse generator is an excellent candidate for use as a high-stability, high-resolution signal source for testing Pentium and other high-speed processor designs.

It is unlikely that the gates on the technology board will be exactly the circuitry that appears in the final design, since this sort of activity is most likely performed very early in the design process. A grid of test points in the region where the buffer is likely to be placed offers visibility into the power and ground planes, and these can be evaluated by a high-speed oscilloscope or spectrum analyzer. Once it is known where the quiet locations are, the placement and orientation of the buffer can be specified.

**HP 8133A Pulse Generator.**<sup>10,11</sup> For many of the measurements described in this section, it is critical to use a high-precision adjustable signal source. The HP 8133A pulse generator (Fig. 10) is an excellent choice for the stimulus instrument in these measurements. It is stable, accurate, and precise. The rms jitter for this instrument is warranted to be less than 5 ps. Both authors have had the opportunity to characterize a number of these instruments. The result of these characterizations is that the distribution is approximately Gaussian. Furthermore, for pulse repetition rates below 500 MHz, the rms jitter of the instruments is typically 1.2 to 1.3 ps. Rms jitter is equal to one standard deviation of the jitter distribution. Worst-case jitter can be taken to be six standard deviations. For a Gaussian distribution, this means that worst-case jitter is approximately 8 ps. Applying this to our 50-ps Pentium tolerance budget, we would have to ensure that the system jitter is less than 42 ps to ensure that the system functions correctly during testing. This also means that most of the jitter of the measurement comes from the system and not from the external source.

When the HP 8133A is configured as a multichannel instrument (Option 003 is recommended for clock characterization and testing activities), the phase delay from one channel to the other can be adjusted in 1-ps increments from the front panel or in 300-fs steps over the HP-IB (IEEE 488, IEC 625).

If a less stable or precise source is used for these measurements, the quality of the results could be compromised. For

example, if we assume jitter levels of just a few tens of picoseconds, the system may not even function properly during testing and the measurement of any jitter in the system will be less meaningful since the majority of the jitter will come from the external source.

### Summary

In this article, we have reviewed some of the significant challenges that exist in designing a statistically stable timing environment for a 66-MHz Pentium system. Many of the difficulties described easily generalize to most of the other new high-speed processors as well. We have advanced the argument that a new, more informed approach to designing the timing for these more aggressive systems is required. This informed design approach requires the determination of important design information at the front end of the design process so that important subsequent design decisions can be made knowledgeably, with more predictable results.

We also examined a variety of measurements that support this approach. Our tolerance budget for a typical Pentium system revealed much more sensitivity to jitter than has been common for designs at this level. Our discussion centered on the measurement of jitter-related design information. In the course of discussing these measurements, we also examined the role of stimulus equipment. Specifically, we discussed what impact various facets of the performance of a high-stability pulse generator would have on the quality of the measurement data. For example, the simple decision to use a higher-stability pulse generator as an adjustable substitute for the clock means that the design can have higher levels of intrinsic jitter (i.e., a simpler design) and continue to function during testing. In the course of our discussion, we showed how the HP 8133A pulse generator can be employed in designs as aggressively timed as Pentium and others.

## References

1. L. Geppert, "The New Contenders," *IEEE Spectrum*, Vol. 30, no. 12, December 1993, pp. 20-25.
2. W.R. Blood, Jr., *MECL System Design Handbook, Fourth Edition*, Motorola Inc., 1988.
3. *Pentium Processor User's Manual—Volume 1: Pentium Processor Data Book*, Intel Corporation, 1993.
4. *Pentium Processor User's Manual—Volume 2: 82496 Cache Controller and 82491 Cache SRAM Data Book*, Intel Corporation, 1993.
5. D. Lin and J. Reilly, *Pentium Processor Clock Design*, Application Note AP-479, Intel Corporation, March 1993.
6. R. Jolly, *Clock Design in 50-MHz Intel486 Systems*, Application Note AP-453, Intel Corporation, 1991.
7. M.K. Williams, *Clock Design in Intel Pentium Systems*, Amherst Systems Associates Application Note ASA 93-3, 1993.
8. M.K. Williams, "Design Trade-offs in High-Speed Clock Distribution and Reception," *Proceedings of the 1993 Hewlett-Packard High-Speed Digital Symposium*, pp. 6-1 to 6-34. Also available as Application Note ASA 93-2 from Amherst Systems Associates.
9. M.K. Williams, "Timing Considerations in Clock Distribution Networks," *Proceedings of the 1992 Hewlett-Packard High-Speed Digital Symposium*, pp. 2-1 to 2-21. Also available as Application Note ASA 92-2 from Amherst Systems Associates.
10. H.J. Wagner, "A Programmable 3-GHz Pulse Generator," *Hewlett-Packard Journal*, Vol. 44, no. 2, April 1993, pp. 52-55.
11. P. Schinzel, A. Pfaff, T. Dippon, T. Fischer, and A.R. Armstrong, "Design of a 3-GHz Pulse Generator," *Hewlett-Packard Journal*, Vol. 44, no. 2, April 1993, pp. 60-72.

Pentium is a U.S. trademark of Intel Corporation.