Future Directions in CLIPPER Processors

Howard Sachs and Harlan McGhan
Intergraph Advanced Processor Division
Palo Alto, California

Abstract

The CLIPPER is both the oldest of the commercial RISC microprocessors, dating from the introduction of the original C100 in October of 1985, and at the same time, with the just-introduced C400, one of the newest. To the list of historic firsts for microprocessors pioneered by the C100, the C400 adds the achievement of being the first RISC microprocessor to maintain binary code compatibility with previous generation machines, while providing a completely new implementation at the hardware level, as well as being the first microprocessor to combine superpipelined functional units with superscalar instruction dispatch. After a brief survey of CLIPPER generations to date, we look ahead to what can be expected from CLIPPER in the 1991-1995 timeframe. We then examine in some depth what we regard as the key trends over the next decade, both in microprocessor architectures and implementation technologies, and indicate how we expect the CLIPPER to relate to these emerging directions for microprocessors. Our conclusion is that the last six years have demonstrated the validity of the balanced path pursued to date with the CLIPPER, and we do not expect the next decade to call for a radical break with the past.

1. Introduction

The CLIPPER RISC processor was originally introduced in October of 1985 by the Advanced Processor Division (APD) of Fairchild Semiconductor. (Rights to the CLIPPER, together with the rest of APD, were acquired by Intergraph Corporation in October of 1987, shortly after the purchase of Fairchild by National Semiconductor.) From the beginning, CLIPPER development has been driven by a single main objective, namely: *The achievement of the highest possible performance consistent with reasonable cost.*

In pursuit of this goal, the CLIPPER development team has always taken a balanced approach to processor design and implementation: trading the virtue of simplicity against the convenience of functionality; trading maximum functionality against the benefits of integration; trading the highest levels of integration against the requirement for performance; and trading ultimate performance against the value of cost-effective implementation.

2. CLIPPERS Past and Present: 1985 - 1990

1985: Marked the introduction of the original "5 MIPS" CLIPPER, the C100. Not so much a microprocessor as that term was understood then (i.e., a standalone CPU chip), the C100 comprised a complete "compute engine": consisting of a combined CPU/FPU chip, a commercial CACHE/MMU chip (abbreviated to "CAMMU"), and a clock chip. One CPU/FPU chip, two of the CAMMU chips, and a clock chip were integrated on a small 3" x 4.5" printed wiring board (PWB), called a "module". The CLIPPER module provided system designers with a complete, ready-to-go, "compute engine", needing only the addition of main memory and I/O to become a full system.

In addition to providing unprecedented levels of performance and integration for a microprocessor, the C100 marked a number of historic firsts: the first commercial Reduced Instruction Set Computer (RISC) among microprocessors; the first microprocessor to integrate CPU and FPU functions on a single chip; the first microprocessor to feature a full Harvard architecture, with separate caches (CAMMUs) for data and instructions; and the first microprocessor to implement a number of advanced features, like bus watch and hardwired scoreboarding, previously found only in mainframes.

1988: Marked the introduction of a second generation CLIPPER, the C300. Packaged in the same module format, the C300 was both binary software and hardware plug-compatible (at the module level) with the earlier C100. As a consequence, users of the C100 were able to upgrade their systems by the simple expedient of powering the machine down, swapping module boards, and rebooting.

The focus of the C300 was greatly improved floating-point performance, thanks to a completely redesigned FPU, together with very high frequency operation. The latter feature is the result of employing a technique that involves segmenting functional units into relatively short (and hence electrically fast) pipe stages, which recently has become widely publicized under the name, "superpipelining". In-

CH2961-1/91/0000/0240$01.00 © 1991 IEEE

troduced at a clock rate of 40 MHz, the C300 is currently in production at frequencies ranging up to 75 MHz.

1990: Marked the introduction of a third, entirely new generation of CLIPPER, the C400. While still preserving the all-important binary software compatibility with earlier generation CLIPPERS, the C400 represents a complete redesign of the CLIPPER hardware architecture. The most visible difference is an entirely new scheme for partitioning functions across chips. The C400 chipset consists, initially, of separate CPU and FPU chips with MMU and cache functions implemented in discrete logic. This strategy allows use of modestly sized die for both CPU and FPU, while at the same time it provides ample space for multiple hardwired functional units on each chip.

Implementation of the MMU and cache functions in discrete logic allows system designers to enjoy maximum flexibility in creating these subsystems – though, to be sure, at the cost of added complexity in the system design. (Relative, that is, to system level designs done with the C100/C300. In fact, other leading RISC microprocessors, such as the SPARC and MIPS R2000/R3000, have required system designers to provide discrete implementations of some or all of these memory functions from the beginning.)

In addition to continuing the high frequency, superpipelined tradition of the C300, the C400 adds another dimension to CLIPPER: namely, "superscalar" issue, i.e., the ability to issue multiple (two) instructions in the same clock cycle. As a consequence, the C400 also offers a dramatically reduced Clocks Per Instruction (CPI) metric. For both scalar and vector floating-point code, CPI will typically drop below the RISC goal of 1.0 for optimized loops (e.g., the famous "Daxpy" inner loop of the Linpack benchmark).

3. A CLIPPER Roadmap for the Next Five Years: 1991 - 1995

An overview of the CLIPPER roadmap for the next five years shows a continuation of the balanced path pursued to date, rather than a dramatic break with the past. Future generations of CLIPPER will extend both the superpipelined tradition of the C300, and the superscalar tradition of the new C400: pushing up the overall clock frequency at which the processor runs; as well as both the percentage of clocks on which it is possible to issue multiple instructions, and the number of instructions it is possible to issue simultaneously.

More specifically, the first half of the 1990s will see the following additions to the CLIPPER family:

1991: The current C400 (separate CPU and FPU chips implemented in 1.0 micron CMOS, with caches and MMU implemented in discrete logic) will enter production at 50 MHz, offering overall combined performance (integer and floating-point) of better than 40 SPECmarks.

1992: The C400 will move to 0.8 micron geometries and 60 MHz operation, with overall performance of about 50 SPECmarks. A 64 KB cache/MMU (CAMMU) chip will be added to the CPU and FPU units, offering a cost-reduced, high-integration alternative to discrete logic implementations of these functions. A full Harvard architecture version of the C400 "compute engine", consisting of four chips – CPU, FPU, and two 64 KB cache chips (one for instructions and the other for data) – will be implemented on a 2-inch-square ceramic substrate.

1993: The C400/4 will be introduced: a symmetric multiprocessor (SMP) version of the C400, consisting of four C400 "compute engines" (16 chips total) mounted on a single "flip chip" ceramic substrate, all running at 60 MHz. The C400/4 will achieve roughly 200 SPECmarks on appropriate multiprocessor tasks.

1994: The C500 will be introduced: the fourth new generation of CLIPPER processors, extending both the superpipelined and superscalar abilities of the C400. Targeted to run at speeds up to 100 MHz and able to issue up to four instructions in a single clock, the C500 will exceed 80 SPECmarks in uniprocessor performance. The C500, like the C400, will be implemented as a chipset rather than as a single "megachip".

4. Architectural Directions for High Performance Microprocessors

A major debate within the RISC camp during the 1980s concerned the choice between an architecture that deliberately exposed details of the pipeline's operation to software or one that deliberately concealed as many details about its operation as possible from software. Advocates of "exposed" pipes argued forcefully that only by exposing the pipelines within the machine was it possible for software writers to construct code that would extract the maximum performance from the machine. Advocates of "hidden" pipes argued equally vigorously that only by hiding implementation details about the pipelines would it be possible to reimplement the hardware while preserving binary software compatibility.

The original C100 chose to conceal the operation of its pipelines from software. Our success with the C400, in reimplementing the CLIPPER hardware base, while preserving binary compatibility with the millions of lines of application code written for the CLIPPER since 1985, does nothing to make us doubt the wisdom of this choice.

Architecturally, the historic CLIPPER strategy of providing a robust (by RISC standards) instruction set will move

closer to the center of the RISC architectural spectrum. Increased silicon resources will lead to moderation of dogmatic RISC tenets about the need for instruction set simplicity. The CLIPPER will add instructions to handle strings, decimal numbers, vectors, transcendentals, etc.

The C400 strategy of combining superpipelining with superscalar abilities also will become more commonplace. In theory, these strategies are sometimes regarded as redundant since they both exploit the same, often scarce, resource – instruction-level parallelism in code. In practice, however, they are generally complementary. Only an approach which combines both is able to make maximum use of the on-chip resources that must be provided in any high-performance processor: since the effect of superpipelining is to increase usage of the serial logic present within given pipelined functional units, and the effect of superscalar dispatch is to increase usage of the parallel logic present across the different functional units.

Moreover, combining the two has the practical advantage that the resulting processor is still able to exploit the very high levels of parallelism typically found in vector code, without need of the extreme measures that would be required to achieve the same degree of parallelism by use of either technique alone. For example, building a "pure" superpipelined microprocessor, able to operate in CMOS at speeds in excess of 100 MHz, requires either a very exotic and expensive memory subsystem, or use of on-chip caches. On-chip caches, in turn, require use of large die, which (as we shall see) carries a severe economic penalty. Also, even the largest die feasible today can support only a relatively small primary cache (typically 8 KB), which means that it is still necessary to construct expensive (and complex) off-chip secondary cache.

Alternatively, building a "pure" superscalar microprocessor, able to issue more than four instructions in a single cycle, requires not only elaborate decode circuitry, but also the construction of very high bandwidth memory with multiple load/store pipes. Multiple paths to and from memory are both expensive and pin-intensive to implement. Elaborate decode circuitry well may have the effect of slowing down the overall processor clock rate, thereby making the entire machine operate at lower (less efficient) frequencies than otherwise would be possible.

5. Technological Directions for High Performance Microprocessors

A. Directions in Chip Technology

Of the major chip technologies that will be competing over the next decade, namely CMOS, ECL, and GaAs, we believe that CMOS will continue to be the standard for the foreseeable future. Indeed, we predict that, before the end of the decade, CMOS will displace ECL as the highest performance technology available for practical, commercial uses, and will begin to threaten the highly specialized niche applications dominated by GaAs. *Figures 1* and *2* summarize the trends for these competing chip technologies in the two performance critical areas of gate and interconnect delays.

B. Directions in Chip Size

Improving geometries will make possible as many as 5 million transistors on a single large production die by 1995. We expect that, over the course of the next five years, many microprocessor vendors will be tempted by this seemingly enormous number into implementing "compute engines" on

Figure 1 Gate Delay (ps)

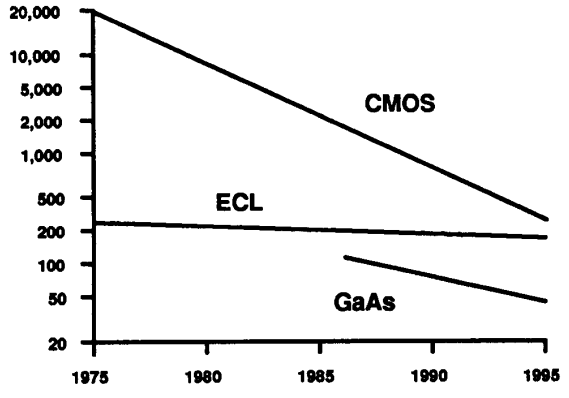
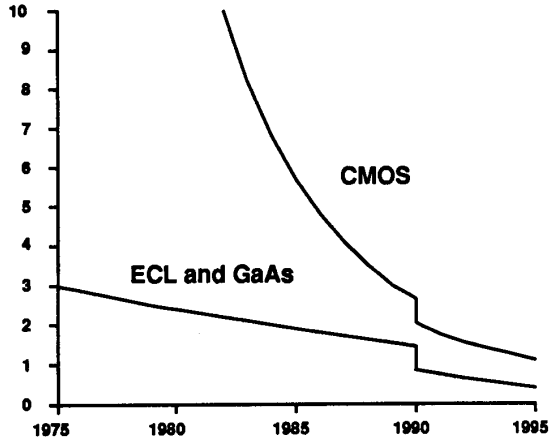


Figure 2 Interconnect Delay (ns)



a single die, complete with CPU, FPU, MMU, cache, special function units, and possibly memory and I/O control. Nonetheless, despite the fact that we see a shift towards building all-in-one "compute engines" on a single "megachip" as an emerging direction for microprocessors, it remains our conviction that the pursuit of this particular holy grail is one more illustration of the fact that things often happen more because they are possible than because they are sensible (especially where technology-driven enterprises are concerned).

The facts as we see them are that, purely on grounds of cost-effectiveness, it will continue to make sense to implement microprocessors as a set of two or more moderately sized die for the foreseeable future. In the first place, even 5 million transistors are not enough to implement a truly high performance compute engine. The IBM RS/6000, for example, uses nearly 7 million transistors (spread out over nine chips); with 4.5 million of that number expended just to implement a rather minimal 64 KB cache. In the second place, even if 5 million transistors were adequate for the job, the performance advantage gotten from implementing a processor as a single "megachip" does not outweigh the economic handicap incurred by the use of large die.

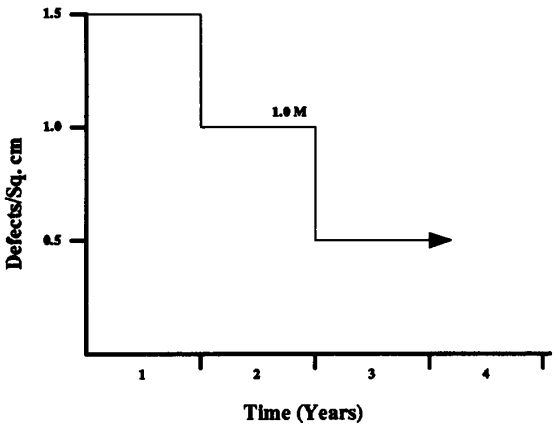
Analyzing this second point requires that we weigh two opposing issues: the cost of large die, i.e., the economic penalty paid for use of a single "megachip" (rather than implementing the same thing using several smaller die); vs. interconnect delays, i.e., the performance penalty paid for using several smaller die (rather than a single "megachip"). The next section will consider the second of these issues, namely, the problem of interchip communications and ways to reduce the penalty paid for crossing chip boundaries. In the remainder of this section, we will look at the first issue mentioned above, namely, the economics of die size.

Extensive experience in the manufacture of semiconductor devices has established a principal widely known as "the learning curve." Briefly stated, the learning curve tells us that when a new process and design first goes into manufacturing at a new geometry (e.g., a leading-edge microprocessor implemented in a leading edge chip technology), the defect density will start at about 1.5 defects per square centimeter, drop to 1.0 defects per square centimeter in about a year, and in another year finally stabilize somewhere around 0.5 defects per square centimeter.

The learning curve is important, because from the changing ratio of defect densities over time, we can calculate the changing cost of silicon. *Table 1* summarizes this economic lesson.

The first (full) row of the table shows die size in half square centimeter increments, ranging up to about the largest die currently practical, 2.0 square centimeters. The second row shows the number of die of the given size it is physically

Figure 3 The Learning Curve



possible to place on a 6" wafer. Given this information about numbers of candidate die, and a defect density ratio taken from the learning curve, it is possible to calculate Y, the effective yield (i.e., number of good die) from a 6" wafer by use of the following formula:

$$Y = e^{-AD}$$

A = area of a die in square centimeters
D = the defect density, in number of defects per square centimeter

Applying this formula, for example, to the column for 2.0 square centimeter die, we note first that this large area die generates just 70 candidates on a 6" wafer (vs., say, 146 candidates for the 1 square centimeter die). Now assume that we are in the first year of production for this die, viz., that the defect ratio is at the typical maximum for a new process of 1.5 defects per square centimeter. Then, on average, just 3 of the 70 candidates on a 6" wafer will be good. As the defect ratio drops with time, the number of good die (out of the 70 candidates) rises accordingly. During the second year of production, at a ratio of 1.0 defects per square centimeter, our yield triples to 9 good die. During the third year of production (and beyond), when the defect ratio has bottomed out at 0.5 defects per square centimeter, the yield nearly triples again, to around 26 good die.

Finally, given yield numbers, it is possible to calculate the cost of (working) silicon – provided that we know the price of a 6" wafer. Assume that, realistically speaking, a 6" wafer must bring about $3500 in revenue (including all packaging and finishing costs, and a reasonable profit margin). It follows that in the first year of manufacturing 2.0 cm^2 die,

Table 1 The Economics of Die Size

Wafer Revenue:	$3500				
Die Area (Sq. cm)		2.00	1.50	1.00	.50
Candidates/Wafer		70	95	146	314
Die/System		1	2	2	4
Extra Si Area		n/a	50%	0%	0%
Year 1: Defects/Sq. cm	1.5				
Number Good Die		3	10	33	148
$/Die		$1,167	$350	$106	$24
$M/100K Systems		$117	$70	$21	$10
Year 2: Defects/Sq. cm	1.0				
Number Good Die		9	21	54	190
$/Die		$389	$167	$65	$18
$M/100K Systems		$39	$33	$13	$7
Year 3: Defects/Sq. cm	.5				
Number Good Die		26	45	89	245
$/Die		$135	$78	$39	$14
$M/100K Systems		$14	$16	$8	$6
Total Si $/3 yrs.		$169	$119	$42	$23
Total $M Saved		n/a	$50	$127	$147
Avg $/Sq. cm Si		$282	$132	$70	$38
$ Increase/Sq. cm Si		n/a	213%	402%	748%

when the defect density hovers around 1.5, the actual cost incurred to produce a (good) chip is nearly $1200. This cost then drops over the next two years, first to $389 a chip (at a yield of 9 die per wafer), and ultimately to $135 a chip (at a yield of 26 die per wafer).

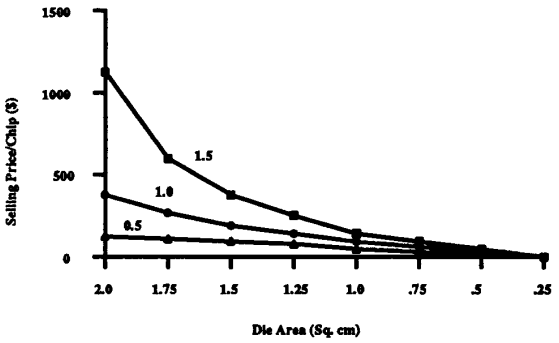
If we now suppose that the single large die represents a complete, all-in-one microprocessor, so that we need just one of these chips in a uniprocessor system, and that we are going to sell 100,000 such systems each of the three years while we are sliding down the learning curve, it turns out that our total silicon costs over this period for micro-processors will add up to $169 million. That expense needs to be compared with the expense of $42 million for silicon, shipping the same 100,000 uniprocessor systems for the same three year period, but with the single large 2.0 cm^2 die replaced by two 1.0 cm^2 die (with exactly the same total silicon area). See column 3 of *Table 1* for details. (In the interest of brevity, we will skip over the tedium of explicitly working through all the rows of this second example.)

The bottom line here, for using the two smaller 1.0 cm^2 chips to replace the one large 2.0 cm^2 chip, is a very real savings of $127 million. To put the same point another way, it turns out to be more than 4 times as expensive to implement a microprocessor as a single "megachip", rather as two smaller chips with exactly the same total silicon area.

Of course, this rather simple "level sales" model can be accused of exaggerating the problem. The truth is that sales are likely to be smallest during the first year, when the cost differential is greatest, and largest in years three and on, when the cost differential is lowest. Even so, a substantial difference must remain. Suppose instead sales of 10,000 systems in the first year, 50,000 systems in the second year, and 100,000 systems in the third year. In this case, the cost of using a single large die over the three year period will drop to a total of $45 million. However, the cost of the two die implementation over the same period will be only $17 million. Product cost of the one large chip is still 2.7 times the cost of the two smaller chips. The savings of $28 million still goes straight to the bottom line.

Even in the least favorable case possible, looking just at the steady cost of silicon from year three on, the expense of building 100,000 systems a year using one large die is $14 million, as opposed to only $8 million for using two smaller die. A savings of $6 million a year (supposing a run rate of 100,000 units a year) is hardly inconsequential. Indeed, at any run rate, the premium of 75% paid on each and every

Figure 4 Price Relative to Defect Densities



"megachip" microprocessor requires a very strong justification.

While *Table 1* shows there is some economic advantage to using even smaller die than 1.0 square centimeter, as *Figure 4* shows, the point of diminishing economic returns has been reached at this size.

Next, we need to consider the other side of the price/performance equation: the performance penalty paid for use of multiple chips, namely the interconnect delays encountered.

C. Directions in Packaging

Traditionally, chipsets have been packaged separately, as individual parts, then integrated on standard PWBs. For example (as mentioned earlier), this is the way both the C100 and C300 modules are built today. *Figure 5* illustrates the transmission delays involved with this packaging technology.

To communicate between CMOS chips, it is necessary to step up the capacitance of 0.6 picofarad (pf) at chip boundary to 20 pf, then drive across 6 inches of wire. The time required to make a round trip journey, going over and back again, is about 5.4 nanoseconds (ns) -- 1500 picoseconds (ps) to cross the series of three inverters shown, plus 1200 ps to cross six inches of wire; times two for the return trip. Assuming the chips them-selves can run at a clock rate of 50 MHz, i.e, with 20 ns intervals between clock edges, this means that a processor built from two chips would run at a clock rate interval of 25.4 ns, or 39 MHz. (The 20 ns of intrachip delay plus the 5.4 ns required for interchip communication.)

The performance penalty, then, for using two chips rather than one is the difference between 39 MHz (the speed possible with two chips, counting the interchip communication delay) and 50 MHz (the internal speed of a single chip); or about 27%. It seems clear that this penalty is modest, compared with the penalty in cost, of approximately 400% (worst case) or 75% (best case), for using a single large die.

However, even this relatively modest performance penalty can be cut roughly in half, by mounting multiple unpackaged chips closely together on a Multi-Chip Molule (MCM). Use of MCMs, e.g., a ceramic substrate, is a relatively new packaging technology that only recently has become commercially feasible as a result of improvements in wire bond yields. Power dissipation is about the same on an MCM as on a single chip, and 250 times lower than chips packaged separately and mounted on a PWB.

Since chip-to-chip capacitance on an MCM is only about 1/10 the capacitance on a PWB (5 pf. vs. 50 pf.), and the wire distances are considerably shorter, by using this packaging technology the total round trip time needed to drive signals chip-to-chip can be reduced to just 2.6 ns. See *Figure 6* for an illustration of the typical interchip communication delays involved on an MCM.

Figure 5 Chip - Chip Communications on a PWB

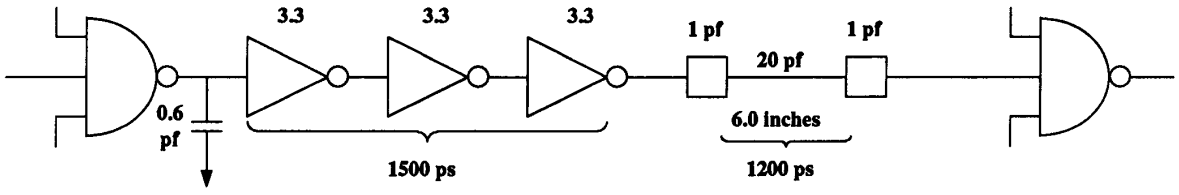
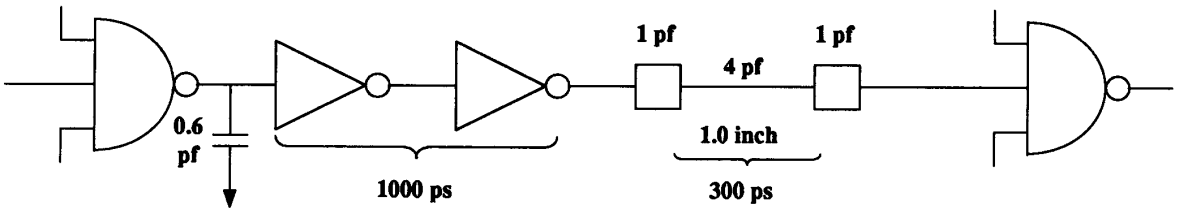


Figure 6 Chip - Chip Communications on an MCM



Again, assuming an intrachip clock rate of 50 MHz, a system of two chips would run at 44.2 MHz. Thus, the performance penalty paid with MCM packaging technology for using two chips rather than one is only 13%

Of course, since the chip-to-chip communication delay on an MCM is essentially constant, while the internal frequency of the chip itself can be controlled by (super)pipelining the functional units more or less heavily, the performance penalty paid for implementing a processor as multiple chips, rather than as a single megachip, will increase with increasing chip frequencies. *Table 2* summarizes this change.

Even at an upper limit of 100 MHz, the clock rate advantage for use of a single chip, rather than multiple chips packaged on a ceramic substrate, only amounts to slightly more than 20%. And at lower frequencies, the advantage is smaller – amounting to less than 8% at 33 MHz.

D. Directions in Systems Technology

Inexorably, as chip geometries continue to shrink from 1.0 micron to 0.8 micron to 0.6 micron and below, more and more of the functions needed in a typical system will be combined into fewer and fewer chips. In the end, the processor will integrate not just CPU, FPU, and MMU functions, together with a relatively generous primary cache (on the order of 128 KB or more), but DRAM control functions for main memory, disk and other I/O controllers, display graphics functions, and so on. It will not, however, be possible to put the 10 million or more transistors required for all this functionality onto a single chip until the second half of the decade just beginning; and even then, it still will be economically sensible to implement this functionality as several modestly sized die, rather than as a single "megachip".

Table 2 Interchip Delay Penalties Relative to Frequency

Chip Frequency (MHz)	33	40	50	60	70	80	90	100
Intrachip Clock (ns)	30.3	25.0	20.0	16.7	14.3	12.5	11.1	10.0
Interchip Clock (ns)	32.9	27.6	22.6	19.3	16.9	15.1	13.7	12.6
Single-Chip Speedup	7.9%	9.4%	11.5%	13.5%	15.4%	17.2%	20.6%	20.6%