THE CASE FOR FIBER-TO-THE PROCESSOR

P. Wijetunga and A. F. J. Levi
Department of Electrical Engineering
University of Southern California, CA 90089
http://www.usc.edu/alevi, wije@ieee.org, alevi@usc.edu

Abstract

In the near future electronics will fail to deliver the interconnect bandwidth density required to match microprocessor performance. This, combined with increased electrical power consumption, is driving microprocessor development to a crisis. Solutions based on conventional electronics and packaging will increasingly fail to effectively remove the stress imposed on system performance. One promising alternative approach exploits recent advances in photonic technology to create opportunities for new system architecture optimization using a concept called fiber-to-the-processor.

INTRODUCTION

Moore's Law [1] has dominated the development of microprocessors. As shown in Figure 1, the number of transistors per chip doubles every two years. This creates both opportunity and challenges.

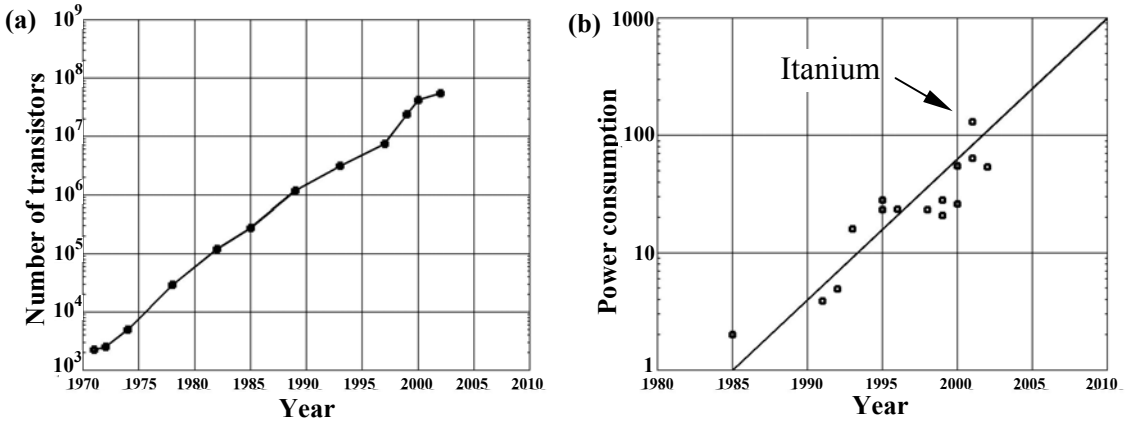

Figure 1 – (a) Moore's Law has successfully predicted that the total number of transistors on a microprocessor doubles every two years [2]. (b) Microprocessor power dissipation as a function of year [2].

During the last decade (1990 – 2000), CMOS technology has scaled from 1.0 μm to 0.18 μ*m* minimum feature size enabling an increase in the number of transistors per chip from about 1.2 million to 42 million [2] (Figure 1(a)). Over the same period, the associated improvement in transistor performance, greater power consumption and improved architecture has allowed clock rates to increase by 30 times (i486DX – P4) [2]. The increase in number of transistors per chip has provided opportunity for innovation in

micro-architecture by, for example, increasing the number of special-function logic blocks, implementing out-of-order speculative execution, deep pipelining and increasing cache size. The combination of increased number of transistors, increased clock rate and improved architecture has dramatically enhanced total microprocessor performance.

Moore's Law will continue to influence microprocessor design for the present decade until 0.03 μm CMOS technology is implemented. By the year 2010 architectures calling for a billion transistors on a chip operating at clock rates in excess of 10 GHz and delivering nearly a trillion instructions per second will be considered. Multi-threading and multiple-processor architectures on a single-chip will be adopted to increase throughput and the number of operations per second. Unfortunately, these trends will result in dramatically increased power consumption and exert significant bandwidth performance demands at the platform level.

Power dissipation will become a critical factor in the coming years. As shown in Figure 1(b), simple extrapolation of current trends predicts microprocessors with power dissipation approaching 1000 W by 2010. Power consumption at these levels in a single die is not practical. Reduction of power consumption and power management will become a dominant aspect of design. This is a significant challenge that will increasingly preoccupy design effort. System architects will seek solutions from new and emerging technologies to find better system design points. A promising approach is to emphasize high-speed IO and de-emphasize increase in number of transistors per chip.

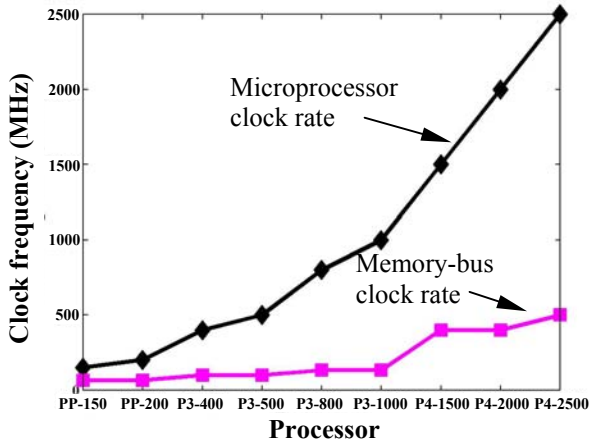

Figure 2 – The imbalance between microprocessor clock rate and memory bus clock rates continue to grow with successive generations of microprocessor [3].

In the coming years, the *imbalance between microprocessor performance and memory access* will be driven to a crisis point. As illustrated by Figure 2, with successive generations, the difference between microprocessor and memory-bus clock rate continues to grow. Without a new approach, the microprocessor will lose hundreds of process cycles while waiting for a single read from main memory. Again, system architects will push electrical bus rates into the GHz range and increase bus widths. Unfortunately, this approach becomes significantly more difficult with increasing clock frequency.

At higher frequencies memory interface requires controlled impedance lines, and due to the nature of the periodically loaded bus, existing bus-based memory architectures must be replaced with switch-based point-to-point architectures. For bus architectures, when the device distance between two memory devices equal the quarter wavelength it creates constructive interference of reflections and as a result a periodically loaded memory array reflects all the energy beyond its cut-off frequency. The hard cut-off frequency of RAMBUS is 1.5 GHz [4]. Work on electrical bus interfaces have also shown that device loss of periodically loaded bus is proportional to the frequency square, and consequently, device losses further worsens the frequency and power consumption performance of high-speed buses.

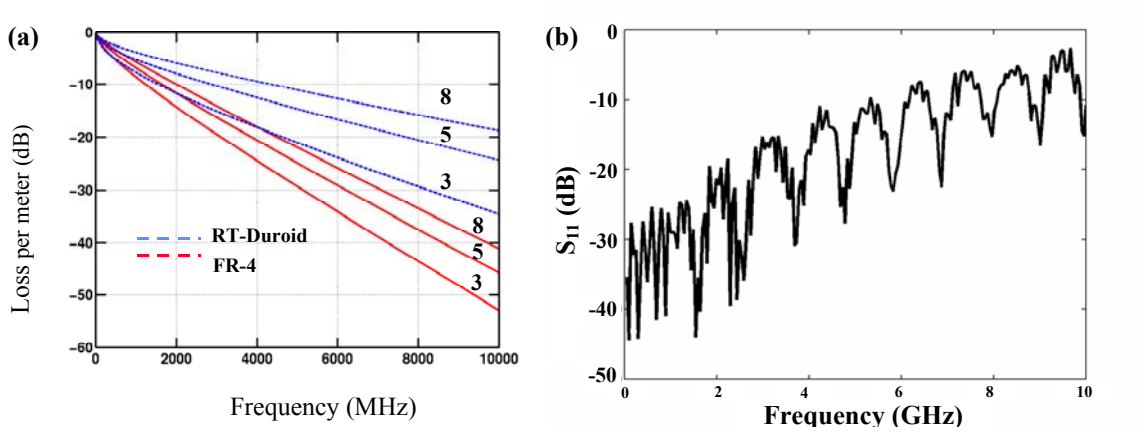

Figure 3 – (a) Simulated loss per meter for 50 Ω microstrip including skin-effect and dielectric losses for FR-4 (ε_r = 4.5, tan δ = 0.02) and RT-Duroid (ε_r = 2.35, tan δ = 0.005) for trace widths 8, 5 and 3 mils in 1 *oz* copper [5]. (b) Measured S_{11} as a function of frequency for a standard 50 Ω high-speed electrical test-fixture (Tektronix #671-3273-00). Inadequacies of the SMA launch onto a 3"-long, 60 mil wide microstrip trace result in severe reflections at 10 GHz [5].

Although, pure-electrical point-to-point architectures scale better than bus architectures, they are also unsuitable for future low-latency high-speed memory interfaces.

As illustrated in Figure 3(a), losses in micro-strip lines implemented in FR4 increase significantly with frequency. However, the processor memory-controller interface length is expected to remain below 20 cm in most systems, and for such short lengths attenuation is not a significant issue. On the other hand, a very serious issue that relates to cost-effective packaging is the difficulty in launching high-speed signals. This is illustrated in Figure 3(b) where S_{11} is shown as a function of frequency for a standard high-frequency test fixture. Controlled launch for each signal across a wide bus and maintaining impedance through vias and electrical board connectors is a major challenge for an all-electronic approach. In addition, electrical crosstalk and radiation from a bus operating at GHz rates makes it difficult to maintain signal integrity at the platform level.

The overall trend in microprocessor development is one that is being driven to a crisis in both power dissipation and memory access. Solutions based on conventional electronics and packaging will increasingly fail to effectively remove the stress imposed on system performance. Fiber-optics is the radically different technology which will provide the path forward for future system design. Here, the electronics industry will

benefit from the development and maturation of fiber-based telecommunication technologies.

In the following we will show that optical interconnects provide immediate benefits for system area networks (SANs) in scalable multiprocessor systems. What is more difficult to show is the performance advantage of using optics for local processor to main-memory data transfers, because this involves more complex analysis. We first discuss the impact of memory interface performance on future microprocessors.

IMPACT OF MEMORY INTERFACE PERFORMANCE ON FUTURE PROCESSORS

Processor pipeline and clock speeds continue to improve. Multi-threading, multiple issue pipelines with parallel and out-or-order execution are already employed in high-end processors and will be extensively used in the future [6][7]. In addition there is considerable interest in single-chip multi-processing and super-scalar architectures. Such enhancements are expected to reduce pipeline execution latencies and latency between consecutive memory-access. However, overall performance will only increase if processor stalls due to memory access can be avoided. Prefetching and the use of multi-level caches deliver limited gains in overall performance. To keep the processor busy, latency and bandwidth of the memory interface must be improved.

To understand the impact of memory latency and bandwidth on future processors we analyze the execution delay of double precision matrix multiplication on a single-issue 10 GHz processor. For the purpose of simulation, a 2-level cache hierarchy is assumed. Similar to a Pentium 4 processor, the L1 cache size is 8 KB with 2-cycle access penalty (4-way set set-associative) and L2 cache size is 256 KB with 7-cycle access penalty (8-way set-associative, 9-cycle total L2 hit penalty). A first-byte memory access latency of between 10.0 ns to 50.0 ns is used in the simulations (expected latency for FTTP is ~40.0 ns, and for E-FTTP it is ~30.0 ns).

Because matrix multiplication involves predictable memory access patterns and reasonably high memory locality, and also since the memory interface has a greater impact on super scalar architectures, the selected example will give us a conservative estimate. In matrix multiplication, (A[256][128]*B[128][64] = C[256][64], Figure 4) previously accessed element has a high probability of being accessed again resulting in high temporal locality. Also when an element is accessed there is a high probability that its neighbors will be accessed in the near future resulting in high spatial locality. Compared to other applications, matrix multiplication has high cache hit rates and few prefetch requirements (less than 15% memory references and 99.8% cache hit rate without any prefetching). Thus, when locality and predictability of access is combined, memory interface bandwidth should not greatly impact the performance of the matrix multiplication. Also, since there is only a limited number of predictable prefetches, regardless of the memory interface bandwidth, there should not be memory interface saturation and prefetching should be able to improve L2 miss rate. This is, therefore, a harsh test of optical interconnect technology whose primary attribute is high bandwidth density.

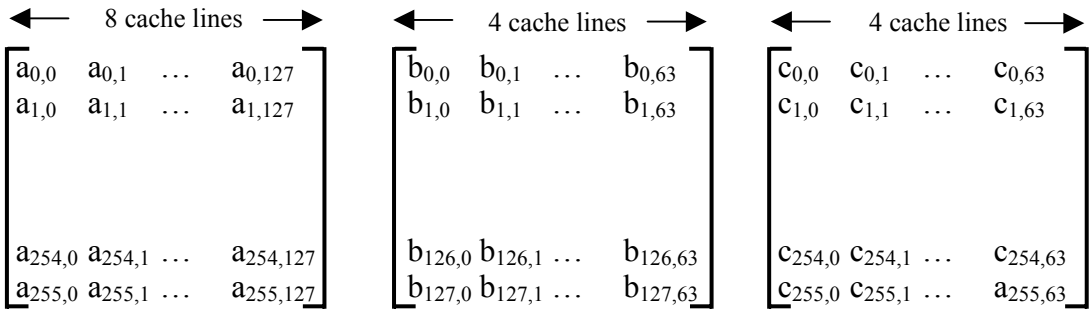

Figure 4 – The data pattern for the matrix multiplication test program. Each element of matrixes requires 8-bytes of space. Cache line corresponding to $A_{x,y}$ is accessed 512 (64×8) times, $B_{x,y}$ is accessed 1024 (256×8) times, and $C_{x,y}$ is accessed 256 (32×8) times. 99.8% of the 2097152 memory references to 'A' are repetitions.

For the selected example, the impact of prefetching on processor performance at 3.2 GB/s and 32 GB/s memory interface bandwidths are given in Table 1. Two prefetching schemes that are used to improve L2 are shown in Figure 5.

Table 1: Impact of prefetching on a 10.0 GHz processor with a first-byte memory access latency of 40.0 ns

Memory Interface (GB/s)	Prefetching scheme	L2 Misses (1000)	Number of instructions executed (Million)	Execution time (Million cycles)
3.2	No-prefetching	60	31.3	92
3.2	Figure 5.(a)	60	33.5	94
3.2	Figure 5.(b)	53	35.4	91
32.0	No-prefetching	60	31.3	86
32.0	Figure 5.(a)	60	33.5	88
32.0	Figure 5.(b)	16	35.4	70

From Table 1 it is evident that, *even* for limited memory access, memory interface bandwidth impacts the overall processor performance. In addition it can be seen that data prefetching only hides memory access latency if adequate bandwidth is available. Ineffectiveness of prefetching for low memory bandwidth can be understood by examining the code expansion and subsequent prefetch invalidations. Compared to a simple nested for loop, when prefetching about 3.9 million additional instructions are executed. The additional assembly instructions are needed to accommodate prefetch instructions and related logic. 66% of all scheduled prefetch instructions are invalidated for a 3.2 GB/s memory interface. In comparison less than 2% prefetch invalidation occurs for a 32 GB/s memory interface. This suggests even though the prefetches can be correctly predicted, if adequate bandwidth is unavailable, the memory interface saturates and data cannot be prefetched on time (delayed prefetch). Prefetching is unable to hide the memory access latency in the presence of a low-bandwidth memory interface.

(a)

```
for (i = 0; i < 256; i++)
  for (j = 0; j < 64; j++) {
    if (j % 8 == 0)
      prefetch (c[i][j]);
    for (k = 0; k < 128; k+=8) {
      prefetch(a[i][k]);
       c[i][j] = c[i][j] +
            a[i][k] * b[k][j] +
            a[i][k+1] * b[k+1][j] +
            a[i][k+2] * b[k+2][j] +
            a[i][k+3] * b[k+3][j] +
            a[i][k+4] * b[k+4][j] +
            a[i][k+5] * b[k+5][j] +
            a[i][k+6] * b[k+6][j] +
            a[i][k+7] * b[k+7][j];
    }
  }
```

(b)

```
for (i = 0; i < 256; i++)
  for (j = 0; j < 64; j++) {
    if (j % 8 == 0)
      prefetch (c[i][j]);
    for (k = 0; k < 128; k+=8) {
      prefetch(a[i][k]);
      if (j % 8 == 0) {
        prefetch(b[k][j]);
        prefetch(b[k+1][j]);
           •
           •
           •
        prefetch(b[k+7][j]);
      }
       c[i][j] = c[i][j] +
            a[i][k] * b[k][j] +
            a[i][k+1] * b[k+1][j] +
            a[i][k+2] * b[k+2][j] +
            a[i][k+3] * b[k+3][j] +
            a[i][k+4] * b[k+4][j] +
            a[i][k+5] * b[k+5][j] +
            a[i][k+6] * b[k+6][j] +
            a[i][k+7] * b[k+7][j];
    }
  }
```

Figure 5 – Conventional prefetching strategies used to reduce the L2 miss rate. A prefetch instruction will be executed and a prefetch will be scheduled only if the target cache line is unavailable in L1 or L2 cache. If it is currently available in the cache, processor will ignore the prefetch instruction. (a) Elements of the array *a*, and *c* are prefetched. From the data pattern we see that for every inner most loop iteration *a* should be prefetched, while *c* should be prefetched when *j* is a multiple of 8. Array *b* is accessed in a column major order and therefore not prefetched. (b) Even elements of *b* are prefetched. For the inner loop the first access of *b* (b[k][j]) may result in a memory stall. Since prefetches already in the prefetch buffers are executed independent of the main execution pipeline, if there is sufficient bandwidth, the prefetching *b* can eliminate subsequent L2 misses for *b*.

The importance of bandwidth is further illustrated in Table 2. Although prefetching is used, a significant performance improvement is not seen when the first-byte access latency is reduced from 90.0 ns to 60.0 ns (only 1.14× improvement in performance). However, if access latency is kept constant at 90.0 ns and bandwidth is increased to 32 GB/s a 1.51× improvement in the performance is seen.

Table 2: Impact of first-byte access latency on a 10.0 GHz processor

First-byte latency (ns)	Bandwidth (GB/s)	Execution time (Million cycles)	Relative improvement
90	3.2	116	1
	32.0	77	1.51×
60	3.2	101	1.14×
	32.0	73	1.56×

Table 3: Predicted improvements in program execution for a memory interface with 3.2 GB/s and 32 GB/s bandwidth and 40.0 ns first-byte access latency.

Memory Interface Bandwidth	First-byte latency (ns)	L2 Misses (1000)	Prefetch invalidations (1000)	Execution	
				Time (M cycles)	Relative Improvement
3.2 GB/s	40	53	35	91	1
32.0 GB/s	40	16	0.3	70	1.3×

As seen in Table 3, compared to a 3.2 GB/s memory interface there is a 3× reduction in the L2 miss rate for a 32.0 GB/s memory interface. The resulting improvement due to the reduced L2 miss rate is only 1.3×. This suggests that for high-bandwidth memory interfaces reducing the L2 size (lower hit penalty) at the cost of higher L2 miss-rate should improve the overall performance. The above observation is verified through simulation and the results are given in Table 4. A 128 KB L2 cache with 4-cycle access latency outperformed a 256 KB L2 cache by more than 10%. The results in Table 4 show that for a 32 GB/s memory interface a 128 KB L2 cache with lower hit penalties will outperform a 256 KB L2 cache.

Table 4: Comparison of program execution time for L2 cache size of 256 K and 128 K with access latency of 7 and 4 cycles**.**

First-byte latency	L2 Cache		L2 Misses (1000)	Prefetch Invalidations (1000)	Execution time (Million cycles)
	Size (KB)	Latency (Cycles)			
40	256	7	16	0.3	70
	128	4	25	0.3	66
30	256	7	16	0.3	69
	128	4	25	0.3	64
20	256	7	16	0.3	68
	128	4	24	0.3	62

The impact of bandwidth is more pronounce for applications with a higher amount of memory references (memory intensive) and/or less locality. In such situations, memory stalls can only be avoided by improving the L2 hit rate using prefetching. To achieve this adequate memory bandwidth is required to avoid memory interface saturation. A program with less memory locality requires a greater number of prefetches. This results in a greater number of prefetches, increases the bus usage, increases bus contention, and results in delayed prefetch and prefetch invalidations. Thus, when prefetching is used, a program with less memory locality is expected to have higher L2 hit rates if adequate memory bandwidth can be guaranteed. The advantage of higher memory bandwidth for programs with less memory access locality is shown in Figure 6. Applications with 1/10 and 1/100 the memory reference locality of our example can achieve at least 3× and 4.5× performance improvement respectively by increasing the memory interface bandwidth from 3.2 GB/s to 32.0 GB/s. In practice, since the probability of memory interface saturation increases with increased memory traffic, a performance increase of better than 4× and 6× should be expected respectively. Further research in this subject is needed to understand the relationship between L2 miss rates, memory reference locality and memory interface bandwidth.

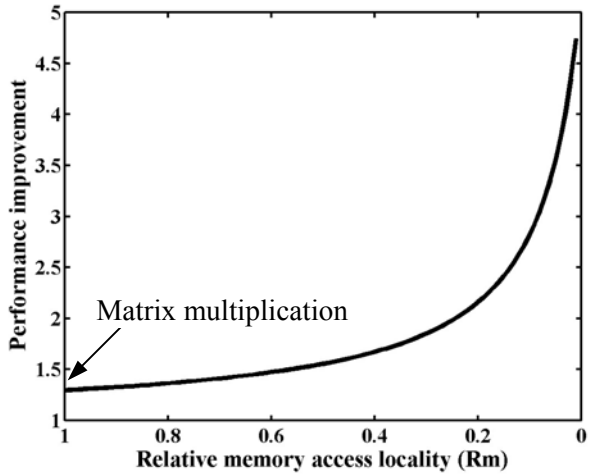

Figure 6 – The expected performance improvement for 32 GB/s memory interface for different memory access localities compared with a 3.2 GB/s memory interface (400 MHz bus × 64). The memory locality is given relative to our matrix multiplication test program. A program with relative memory access locality of 1 ($R_m = 1$) will have 0.2% cache miss rate when no prefetching is used. A program with relative memory access locality of 0.1 ($R_m = 0.1$) will have 2% cache miss rate when no prefetching is used. $R_m = 0.01$ corresponds to 20% cache miss rate. When prefetching is used in a way which is dependent on the on the memory interface bandwidth the cache miss rate will change. A higher bandwidth memory interface will outperform lower-bandwidth memory interface as memory access locality reduces. High bandwidth memory interfaces are able to utilize cache and prefetching more effectively. A first-byte access latency of 40.0 ns and a linear relationship between L2 miss rate and memory access locality is assumed.

Additional simulations are needed to understand the impact of memory bandwidth on the performance of super-scalar processors, multi-threading and single-chip multiprocessing. Super-scalar architectures have multiple parallel execution units and as a result less cycles are required for the actual computation portion of a program. In our example program if we assumed each computation on average required 1 cycle, then moving to a 2-issue pipeline will improve the execution latency 75 M cycles for a 3.2 GB/s memory interface and 55 M cycles for a 32 GB/s memory interface (1.36× improvement).

IMPACT OF HIGH-BANDWIDTH, LOW-LATENCY INTERCONNECTS FOR SYSTEM AREA NETWORKS

To take advantage of multiprocessing, for a given problem set, the communication time must be kept low compared to the computation time. The communication overhead must be minimized (Figure 7).

Improving communication locality, the amount of communication, and improving the interconnection network latency and bandwidth, can reduce the communication overhead. A considerable amount of research is focused on reducing the amount of communication and on increasing problem set locality. Unfortunately for most problem sets such enhancements carry a size and/or accuracy tradeoff. In addition some problems

are still unmanageable without both system area network (SAN) and algorithmic improvements. Thus, as processor performance increases, it is important to scale the interconnection network performance accordingly to make full use of the processors' resources.

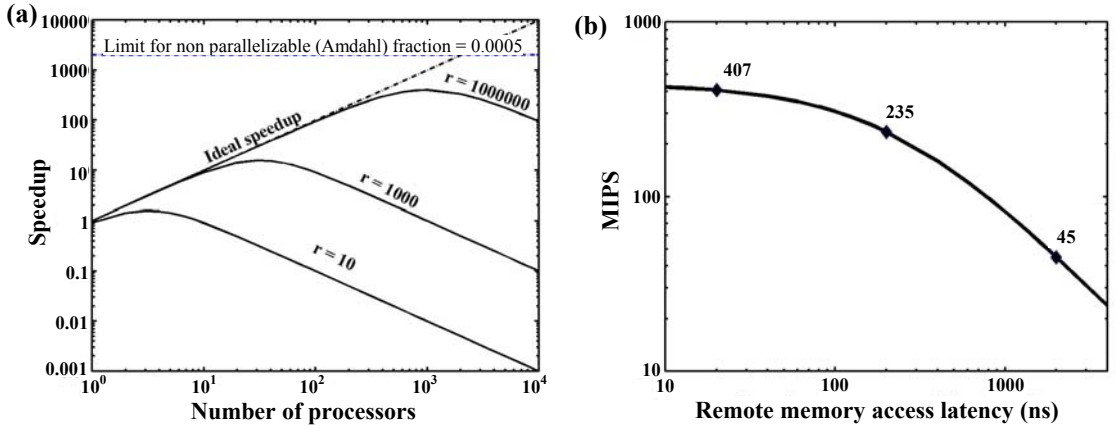

Figure 7 – Results of modeling the impact of inter processor communication delay. (a) The relationship between the total achievable speedup, number of processors and the ratio between the computation time and communication time (r). This kind of speed up is seen in FFT algorithms (seismic activity, weather modeling, etc.). The graph assumes a 0.05% serial portion and that the communication time is proportional to the number of working processors. The maximum speed up (S) is given by $S = \frac{1}{2}\sqrt{r}$. (b) Impact of remote memory access on processor performance. A 94% cache hit rate, 1% remote memory access, 1.3 cycles per instruction and a 5.0 GHz processor speed is assumed. The processor performance for 20 ns, 200 ns and 2.0 μs remote access latencies were 407, 235 and 45 million-instructions-per-second (MIPS) respectively.

SAN performance may be improved by improving the raw data bandwidth and latency performance and/or by improving the efficiency of the switch fabric. Until recently, interconnect physical medium was dominated by pure-electrical interconnects, and the interconnect fabric performance was limited to sub-giga-bit-per-second per-signal-line bandwidths. CMOS technology is now able to out-perform the physical electrical interconnect. Today limited bandwidth and bandwidth density of electrical interconnects are the true SAN bottlenecks. To circumvent this electrical interconnect bottleneck, complex routing and flow control algorithms are used. Under loaded conditions these algorithms may improve effective network bandwidth by reducing network congestion. Nevertheless, they have no impact on the aggregate network bandwidth or the unloaded communication latency. At the same time even the most efficient algorithms still saturate when the attempted load is near 75% of the aggregate network bandwidth [8]-[10]. Due to an increasing processor-interconnect performance gap and the bursty nature of SAN communication, the attempted load can easily achieve the SAN saturation point rendering the improvements gained by complex electronic control logic useless.

Increasing SAN bandwidth is a simple way to improve the no-load and the loaded network latency and bandwidth. As seen in Figure 8, if the interconnection bandwidth is increased from 3.2 GB/s/port, to 32 GB/s/port the no-load latency of a 2-D torus network is expected to increase by 6× (6× for 16 nodes, and 5.5× for 1024 nodes). It can also be

seen that for an equal amount of attempted load, there will be more congestion in the 3.2 GB/s/port network. Thus under equal loading conditions, the latency difference between the 32 GB/s/port network and 3.2 GB/s/port network will be greater than six-fold. Since a network with 3.2 GB/s ports saturates at per-node sustained data rates of greater than 2.5 GB/s (75% attempted load), above 2.5 GB/s the performance gap between the two networks is expected to increase exponentially. As a result of increased processing power and adoption of single-chip multi-processing, the interconnection bandwidth required to avoid network saturation is also expected to grow exponentially in the next few years. To keep pace with such improvements new Tb/s/port scalable interconnection networks are needed in the near future. Fiber-optic interconnect technology is a natural choice for such networks.

There are additional reasons to maintain a focus on improving system interconnect bandwidth. As the Internet matures, bandwidth to a given node will dramatically increase. User applications will evolve to exploit IP and high-bandwidth connectivity. The influence of IP-centric applications in determining future system specification should not be underestimated. Optimizing system performance will require emphasizing high-speed IO. In general, a high-bandwidth SAN also allows intelligence (and the associated power dissipation) to be distributed throughout the system. A processor and translation look-aside buffer *in* main memory is one such example.

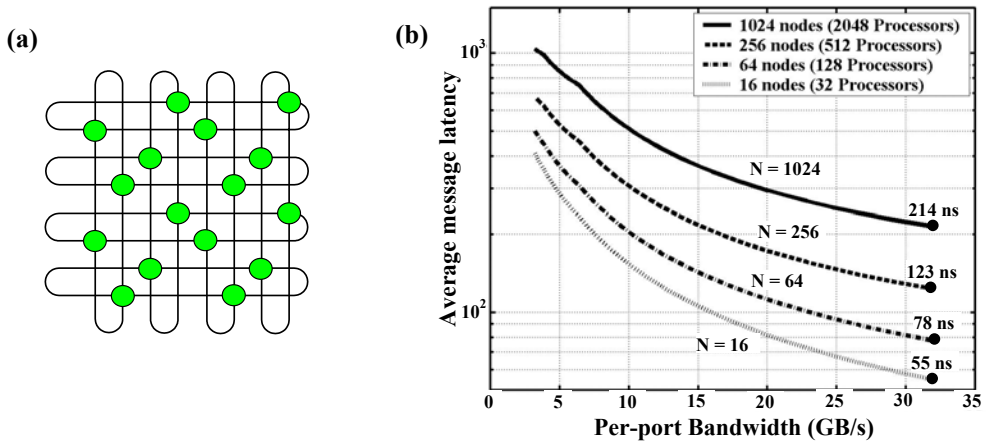

Figure 8 – Modeling the bisection bandwidth and latency of a 2-D torus. (a) Example of a 4-array 2-cube network. A 2-D torus network with 4-nodes in each dimension. (b) Bisection-bandwidth and message latency modeling for a 2-D torus network. Simulation assumes 2-ports per dimension and a physical spacing of 1 m between adjacent nodes. The assumed per-node header latency is 40 ns for clock frequencies below 250 MHz and 16 clock cycles for clock frequencies above 250 MHz. The bandwidth is defined as $2\times\phi\times W$, (where ϕ is the clock frequency and W = 64 is the port width).

THE EMERGENCE OF FIBER-OPTIC TECHNOLOGIES

The success of fiber-optic insertion in telephone systems and the promise of economies-of-scale from a larger component market has resulted in adoption of fiber for Metropolitan Area Networks (MAN) and Local Area Network (LAN) connectivity where

low-cost is a dominant factor. The Gigabit Ethernet standard (IEEE 802.3z) of 1998 and the 10 Gigabit Ethernet standard (IEEE 802.3ae) in progress in 2002 are representative of the adoption of fiber-optics for the LAN environment. Importantly, the reduction in implementation cost has allowed other optical networks such as Fibre Channel (FC) to link machine-room facilities to remote disk storage.

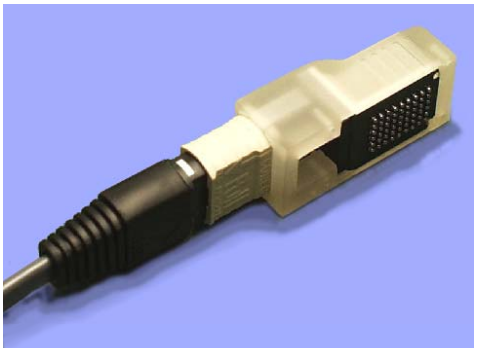

Figure 9 – Example of a prototype parallel fiber-optic transmitter module using VCSEL technology developed by Agilent [11]. The BGA for surface mount to a PCB and the 12 *b*-wide parallel fiber-optic push-pull connector are clearly visible. Today, such commercially available modules have a bandwidth density of 30 Gb/s/cm.

Recently, fiber-optics has been used to solve a different class of problems in machine-room and system interconnect. Here, the difficulty is an edge-connection IO bottleneck at the box-to-box and board-to-board level [12]. In these very short reach applications [13] link distance is less than 300 m so the advantage of fiber-optics for long-distance transmission is not important. However, electrical interconnects simply fail to provide the needed edge-connection bandwidth density (measured in units of Gb/s/cm) and this is where fiber optics has another distinct advantage. A popular solution is use of parallel fiber-optic transmitter and receiver modules [11] which today provide up to twelve independent links with an edge-connection bandwidth density near 30 Gb/s/cm (3.7 GB/s/cm). A parallel fiber-optic module is shown in Figure 9. Future, straightforward scale-up of this technology should achieve bandwidth density of 120 Gb/s/cm (15 GB/s/cm) and adoption of Wavelength Division Multiplexing (WDM) is capable of increasing bandwidth density by an additional factor of ten to deliver 1.2 Tb/s/cm (150 GB/s/cm).

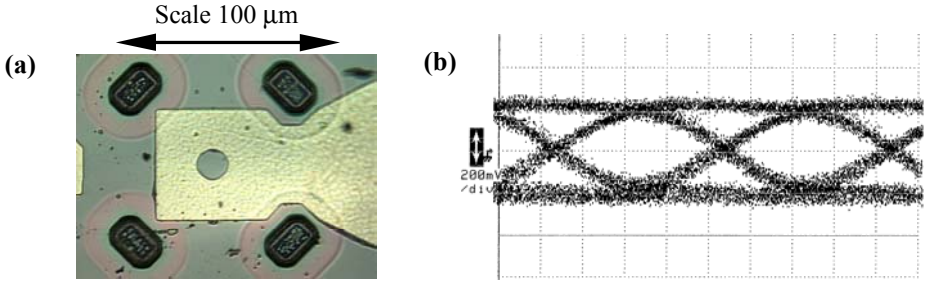

Figure 10 – (a) Photograph of a typical GaAs/AlGaAs oxide-confined VCSEL viewed from above. The light-emitting area is the small region in the center of the image. (b) Transmitted eye-diagram at 2.5 Gb/s of a VCSEL with threshold current 0.5 mA and 1.6 mA_{pp} drive current. Horizontal scale is 200 ps/div [14].

Key to recent advances in fiber-optic interconnects is the development of efficient, high-speed Vertical Surface Emitting Lasers (VCSELs) of the type shown in Figure 10(a). Importantly, as illustrated in Figure 10(b), these devices can consume less power than an equivalent all-electrical LVDS transmitter.

It is the remarkable bandwidth density scaling, the use of power-efficient vertical-cavity surface-emitting lasers (VCSELs), low-cost interface electronics, and inexpensive packaging that make fiber-optics so attractive for addressing the needs of microprocessor platforms. Already, there is some movement in this direction with widespread industry acceptance of Infiniband (IBA) as a System Area Network (SAN) based on 2.5 Gb/s, 10 Gb/s, and 30 Gb/s links. Ultimately, however, fiber-optic interconnect solutions will be inserted directly into a new fiber-to-the-processor platform as a means to solve the power and bandwidth bottleneck crisis that will envelope microprocessors in the next few years.

THE CASE FOR FIBER-TO-THE-PROCESSOR

While the advantages of high memory bandwidth are understood, conventional bus-based electrical solutions are performance limited. Today, a Pentium 4 processor with 2.0 GHz clock has an internal bandwidth of 16.0 GB/s, DDR SRAM has an internal burst bandwidth of at least this, but the system memory interface bandwidth is only 3.2 GB/s. The reason for the low memory bandwidth is easy to understand. For example, the periodically loaded 16 *b*-wide bus used in Rambus designs has a hard cut-off frequency at 1.5 GHz [4] and requires tight manufacturing tolerances. A 4-level electrical signaling scheme proposed by Rambus maintains a manageable bus clock frequency of 400 MHz at the expense of increased power dissipation, reduced noise tolerance, and some latency. Due to the complexity associated with the 4-level signaling, it has recently been abandoned by RAMBUS in favor of a more conventional approach. Today, the RAMBUS 5-year roadmap states next-generation RAMBUS memory interface signaling rate will be increased to 1.2 Gb/s and the bus width will be increased to 64-bits for an aggregate memory bandwidth of 9.6 GB/s (3× improvement over current RAMBUS dual channel architecture) [15]. However, such bandwidths are still much less than required by processors with a 10 GHz clock rate that will be available in the same time period. Electrical interconnects cannot deliver the needed performance and new approaches to system interconnect implementation need to be used. Advances in optical interconnects and WDM technology provide an opportunity to solve these system problems.

The advantages of optical-solutions are clear. There is reduced power dissipation from high-speed chip IO. Photonics provides improved edge-connection density and bandwidth. The optical transmission medium has low crosstalk and zero EMI. These features make optical interconnects the best solution for improving memory access and SAN bandwidth. Incorporating the advantages of fiber-optics with the integration capability of scaled CMOS electronics leads to a new microprocessor design-point called *the encapsulated processor*.

Figure 11 is a schematic of an encapsulated processor. In this case, the encapsulated processor is a single CMOS chip with photonic ports as the only means of external high-speed data communication. The processor includes two CPUs with L1 and L2 cache

connected by a crossbar switch. The crossbar connects to on-chip shared L3 cache and multiple high-speed fiber-optic ports. The processor IC and optical port have separate thermal management. There is a short electrical link from the processor IC to the optical port IC embedded in the sockets shown in Figure 12. The electrical link is low power because there is no need for controlled impedance. The optical port IC decodes and multiplexes signals for the optical sub-assembly that contains low-power VCSEL transmitters, PIN receivers and the fiber interface. The bandwidth density of the fiber communication channel is significantly greater than an electrical alternative. Each photonic port is capable of sustaining 40 GB/s (320 Gb/s) data throughput in each direction and one such port is dedicated to local main memory. Main memory may have its own processors (PIM) and pipelined translation look-aside buffer (TLB) whose purpose is to efficiently feed the encapsulated processor. The remaining optical ports are available for IO and scalable SAN interconnect.

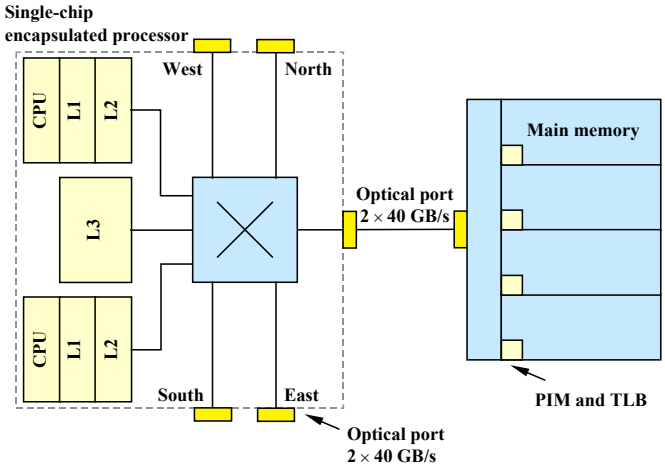

Figure 11 – The encapsulated processor is a single CMOS chip with optical ports as the only means of external high-speed data communication. The processor consists of two CPUs with L1 and L2 cache connected by a crossbar switch. The crossbar connects to on-chip L3 cache and multiple high-speed fiber-optic ports. Each fiber-optic port is capable of sustaining 40 GB/s (320 Gb/s) data throughput in each direction and one such port is dedicated to local main memory. Main memory could be configured to have its own processors and TLB. The remaining optical ports are available for IO and scalable SAN interconnect.

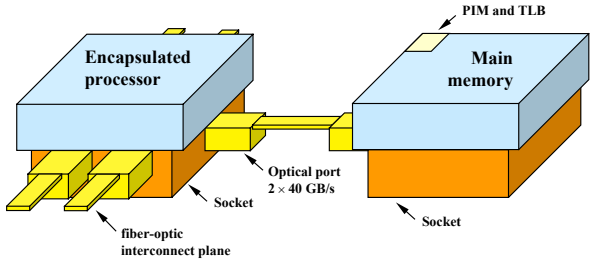

Figure 12 – The encapsulated processor includes a socket that supplies DC current and ground. Incorporated into the socket are the physical optical ports, each of which provide 2 × 40 GB/s data bandwidth external to the encapsulated processor. One optical port is dedicated to local main memory.

The direct replacement of an electrical link with optics introduces an electrical-to-optical and optical-to-electrical conversion delay. Typical values for this delay are less than 0.5 ns for a complete link. In practical applications this is compensated for by the reduced time-of-flight of an optical signal traveling in glass fiber (relative index n = 1.5) compared to an electrical signal propagating in FR4 dielectric (relative index n = 2.19). Such signal delays are essentially insignificant compared to other latencies in the system (DRAM core access latency $t_{RAS} \approx 20.0$ ns).

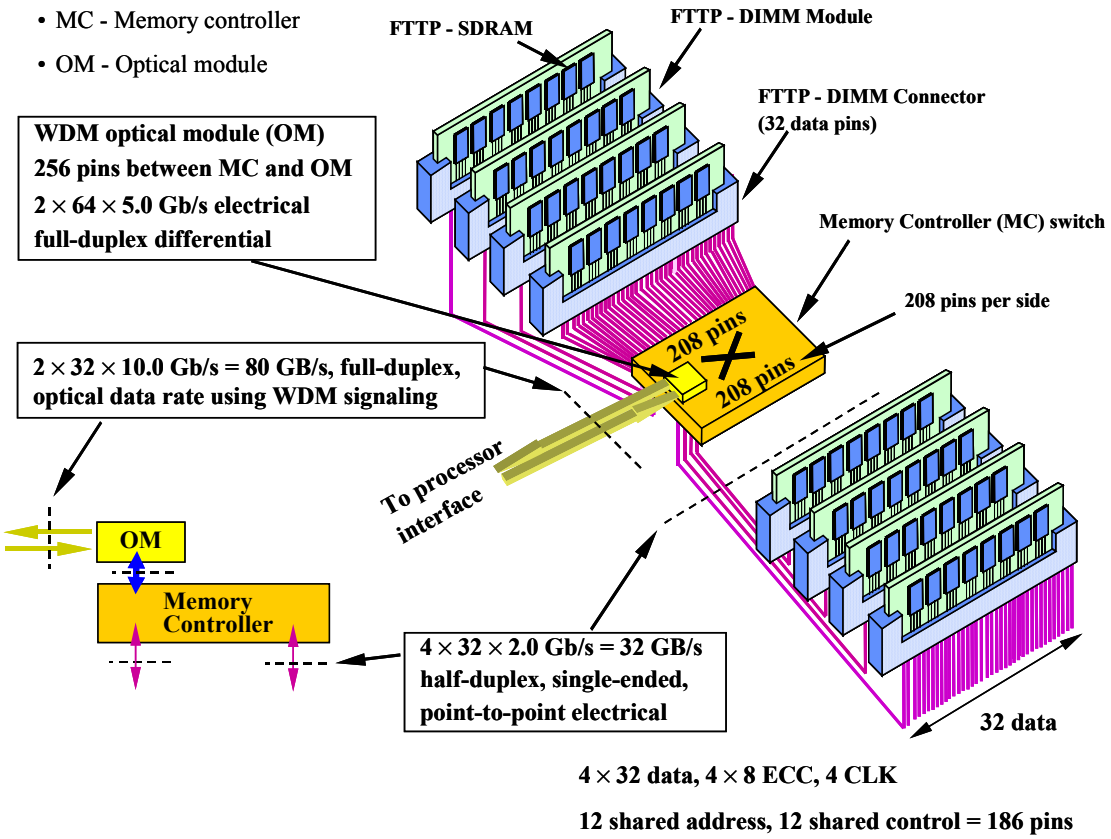

Figure 13 – Conceptual sketch of switch-based fiber-optic memory interface architecture. The interface between the MC and memory modules is 32-bit half-duplex point-to-point (2.0 Gb/s/bit). The bandwidth of each memory module interconnect is 8 GB/s, for a total bandwidth of 64 GB/s for the 8 memory modules. The aggregate bandwidth of the memory modules is supported by 64 GB/s full-duplex interface between processor and memory using 100 Gb/s optical WDM interconnect. Short electrical interconnects are used at the MC-OM interface to avoid use of transmission lines.

To break the latency and bandwidth bottleneck, the existing bus-based main memory architecture is abandoned for a scalable switch based point-to-point architecture. Adoption of the switch based architecture for the main memory requires a high-speed crossbar switch at the main memory controller, and high-speed point-to-point connections between the memory controller and the memory modules. Figure 13 is a conceptual sketch of a switch-based fiber-optic memory interface architecture. Current PC board and connector technologies can support 64 2.0 Gb/s full or half-duplex point-to-point connections. The electrical interconnect bandwidth of the point-to-point connections can scale up to 4.0 Gb/s/pin in FR4. To reduce the packaging complexity of the memory controller (MC) the MC-to-memory interface is limited to 32 2.0 Gb/s half-

duplex links per memory module, and can be scaled to wider interfaces with improved packing technologies. The 64 GB/s full-duplex optical interconnect between the processor and memory controller supports the aggregate bandwidth of the main memory. The total latency to request data from main memory and return it to the microprocessor of about 37 ns is dominated by the 25 ns core row-access (t_{RAS}) latency of DRAM itself (Table 5). This is a 3× improvement over the estimated RAMBUS latencies. As future memory designs improve on this value, the advantages of using a WDM optical port in combination with crossbar switches become more evident. The high bandwidth port consumes less power and less board area compared to any all-electrical alternative. In addition, memory can be scaled incurring minimal additional latency by adding crossbar switches.

Table 5: Comparison of RAMBUS and FTTP delay estimates. The latencies are given in nano (10^{-9}) seconds. The latencies do not include the latency at processor's memory controller.

Architecture	FSB - Chipset	Chipset	Chipset - RAM	RAM	RAM - Chipset	Chipset	Chipset - FSB	Total
RAMBUS	5.0	40.0	1.25 - 6.25	40 - 50	1.25 - 6.25	40.0	5.0	132.5 - 152.5
FTTP	3.0	3.0	3.0	25.0	1.0	3.0	3.0	37.0

Table 6: Comparison between the expected program execution time of RAMBUS (5 year outline) and FTTP for the matrix multiplication example. Results assume a 10 GHz single-issue processor.

Architecture	First-byte latency (ns)	Bandwidth (Gb/s)	L2 miss-rate (1000)	Execution time (Million cycles)
RAMBUS	150.0	9.6	37	115
FTTP	37.0	40	16	70

Scalability of the SAN is dependent on there being enough high-speed ports available for the network. In Figure 11, the bisection bandwidth of an 8-port crossbar switch integrated into the encapsulated processor is 640 GB/s (5.12 Tb/s). Crossbar switches are symmetric structures where the performance is limited by the distributed RC effects of the signal wires and the signal and power/ground routing complexity. Existing logic styles such as static CMOS are based on full-swing operation and are unsuitable for designs that have large RC delays such as the case with crossbar switches. Conventional differential architectures are based on low-swing operation. Nevertheless, compared to static, they have a higher signal routing complexity and consume more area. On the other hand, low-swing logic styles based on pass-transistor logic and high-speed low-power sense-amplifiers are able to achieve high-bandwidth and low-power consumption. Such, innovative circuit design using 0.1 μm CMOS technology predicts that a switch-core with 5.12 Tb/s bandwidth will consume less than 6 W [16].

CONCLUSION

The trends in processor design have dramatically increased power consumption and exerted significant bandwidth performance demands at the platform level. The required low-latency, high-bandwidth, local and remote memory access memory performance cannot be achieved using traditional all-electrical approaches or through latency hiding techniques. In the coming years, the imbalance between processor performance and memory access will be driven to a crisis point.

The adoption of new photonic interconnect technology is the paradigm shift which can provide low-power, low-latency high-bandwidth data-delivery direct to the processor. It is also the only scalable technology that will allow the seamless integration of the processor, local memory and the interconnection network (remote processors and memory).

Fiber-to-the-processor is a natural technology convergence point. It drives the evolution of the PC, workstation, server, mainframe, and router to one basic entity: *The encapsulated processor*.

The promise of fiber-optics is so great that it cannot be ignored. In fact, it is inevitable that, just as fiber-optics migrated from WAN to LAN and then to SAN, it will be embraced as the enabling technology to propel microprocessor platforms to the next level of performance. There are just too many good reasons for adopting the technology.

REFERENCES

1. G. E. Moore, Electronics **38**, 114 (1965). Also reprinted in Proc. IEEE **86**, 82 (1998).
2. Data taken from the Intel web site, http://www.intel.com/research/silicon/mooreslaw.htm
3. Intel data sheets.
4. H. J. Liaw, G. J. Yeh, P. S. Chau and G. Pitner "A 1.6 Gbit/s/pin Multilevel Parallel Interconnection," Designcon, 2001
5. B. Raghavan and A. F. J. Levi, unpublished.
6. G. Hinton, et.al. IEEE J. of Solid-State Circuits, **36**, 1617 (2001)
7. J. Hennessy and D. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kauffmann, San Francisco (1990)
8. J. Duato, S. Yalamanchili and L. Ni, Interconnection Networks, IEEE Comp. Soc. Press, Washington (1997)
9. L. M. Li, W. Qiao, M. Yang, "Switches and Switch Interconnects," Proc. of Massively Parallel Processing using Optical Interconnects, pp. 122 – 129, 1997.
10. W. J. Dally, H. Aoki, IEEE Trans. Parallel and Distributed Systems, **4**, 466 (1993).
11. Now an Agilent Technologies product number HFBR 712BP. Also see http://www.agilent.com/
12. A. F. J. Levi, Proc. IEEE **88**, 750-757 (2000).

13. http://www.oiforum.com/
14. B. Madhavan and A. F. J. Levi, Electron. Lett. **34**, 178 (1998).
15. -, "Rambus Announces RDRAM and RIMM Module Roadmap Through 2005," Rambus News Letter, Vol 3, June, 2001. "http://www.rambus.com/company/press/pressreleases/2001/010613.html"
16. P. Wijetunga and A. F. J. Levi, unpublished