Analysis and Modelling of Optoelectronic Systems - AMOS

Keith J. Symington

Gordon A. Russell

Theo Lim

John F. Snowdon

Heriot-Watt University

School of Engineering and Physical Sciences

February 2003

Contents

Abstract

This technical brief examines commodity computer architectures and the flow of data within such systems. It specifically examines I/O bus, RAM and cache architectures with attention paid to real world performance and associated limitations. The case is made for the move from electronic to optical interconnection and a detailed examination is undertaken of core optoelectronic interconnect components. Finally, an architecture is outlined that exploits the advantages of optical interconnection between commodity system called the optical highway. Conclusions are drawn as to the implications of optoelectronics and their potential impact on modern day computer architectures.

1 Introduction

As the speed of electronic components increases, both the distance and parallelism to which they can be electrically interconnected decreases. This rule is governed by the basic laws of physics, in this case coulomb interaction [1], and is forcing a deceleration in the bandwidth growth of electrical interconnect. Nevertheless, it is possible to sidestep this limitation by moving to a different technology governed by a different set of rules, in this case optical interconnection.

It is important to understand that this technical brief does not advocate the complete replacement of electronics with optics. The coulomb interaction of electrons means that electronics are superior to optics for switching. However, the complete lack of interaction between photons makes them ideal for data transmission. We believe that the physical characteristics of each should be taken advantage of in the appropriate situation thus making the best of both worlds.

This brief begins in Chapter 2 by examining the evolution, present day and future of *input*/*output* (I/O) buses and their realistic throughput rates in conventional computer architectures. Comparisons are made between theoretical with PCI bus measurements presented for reference.

Chapter 3 moves into an architectural examination of computer systems. The focus here is primarily on processor to memory data transfer rates and caching. The reasoning behind the design decisions taken to create such systems is examined.

Chapter 4 examines optical interconnect technologies that could be used to enhance data transfer. It looks at emitters, modulators, detectors and even optical interconnect elements.

Chapter 5 introduces the *optical highway* (OH) concept. This is an architecture designed to connect commodity PC components using a high spatial bandwidth optical interconnect. Detailed analyses of architectures, bandwidth, latency and optical component tolerances are undertaken.

Conclusions are finally drawn as to the potential of optical interconnection as a whole, whether it be via the architecture described here or by using another medium. Please note that there is an extensive variable reference list in Section 7 that details all the variables used in this technical brief.

2 Computer Buses

This section examines the evolution of various computer bus types [2]-[3], assessing both level of integration and associated performance. It is written historically from an IBM PC point of view as this is currently the dominant commodity computer architecture. This section quotes data throughputs as both *theoretical* B_{IOB} and *available* B_{avail}. Theoretical describes a data rate based on bus width and frequency. However, in actual operation a combination of wait states, interrupts and various protocol factors will reduce this figure to the maximum available. This available bandwidth is a range of data rates that are sustainable over an extended period of time given minimal loading from all but essential devices and signals as well as negligible background processing.

2.1 PC Bus

The *personal computer* (PC) bus is a direct extension of the Intel 8088 address and data lines. Its layout was standardised by IBM in 1981 and the bus is sometimes referred to as 8-bit ISA.

Bus Type	Data	Address	Freq.	Throughput B_{IOB} (B_{avail})
PC	8 bit	20 bit	4.77MHz	4.77MBs^{-1} (0.5-2MBs^{-1})

Table 1: PC Bus Performance
Throughput measured in bytes per second with the first value indicating the theoretical maximum and the second the sustained maximum normally achieved in an actual system.

The PC bus requires a minimum of two clock cycles to transfer a single byte giving a best case throughput of 2.38MBs^{-1}. Sometimes this delay can be extended up to 8 wait states.

2.2 ISA and EISA Buses

The *industry standard architecture* (ISA) bus was introduced in 1984 by IBM and is backwards compatible with the PC bus. It was designed to complement the speed and power of newly introduced 286 processors but was, in some ways, a mistake as main memory moved from this bus to a dedicated bus shortly after. The advent of 32-bit

processors, specifically the 386 and 486, resulted in the development of the *extended ISA* (EISA) bus in 1988. EISA enhanced functionality and increased both address and data bus sizes while maintaining backwards compatibility with ISA devices.

Bus Type	Data	Address	Freq.	Throughput B_{IOB} (B_{avail})
ISA	16 bit	24 bit	8MHz	$16MBs^{-1}$ ($1\text{-}3MBs^{-1}$)
EISA	32 bit	32 bit	8MHz	$32MBs^{-1}$ ($8\text{-}12MBs^{-1}$)

Table 2: ISA and EISA Bus Performance
Throughput measured in bytes per second with the first value indicating the theoretical maximum and the second the sustained maximum normally achieved in an actual system.

The bus frequency of both ISA and EISA varies depending on version, but is typically 8MHz. Earlier bus versions used 4.77MHz whereas later implementations permitted clock rates of 12MHz or more [4]. As with the PC bus, a minimum of two clock cycles are required to transfer information thus halving the theoretical maximum throughput shown.

2.3 MCA Bus

Worthy of mention, although not particularly influential, was the *micro channel architecture* (MCA) bus introduced by IBM in 1987. It was an asynchronous bus, therefore clock rates were specified by the attached devices rather than a universal clock. Typical clock rates did not exceed 10MHz.

Bus Type	Data	Address	Freq.	Throughput B_{IOB} (B_{avail})
MCA	32 bit	32 bit	10MHz	$40MBs^{-1}$ ($10\text{-}20MBs^{-1}$)

Table 3: MCA Bus Performance
The MCA bus had a burst transfer mode which could support up to $160MBs^{-1}$. Throughput measured in bytes per second with the first value indicating the theoretical maximum and the second the sustained maximum normally achieved in an actual system.

The bus architecture was developed as a closed system and as such was only ever implemented by IBM in its PS/2 computers and by a small number of its vendors. Unfortunately all practical differences over EISA, which was introduced as a direct

competitor, were negligible. Nevertheless, this architecture lived on for a considerable time in certain IBM server systems.

2.4 VL Bus

The *VESA local* (VL) bus [5] was developed by the *video electronics standards agency* (VESA) in 1991 to specifically address graphical data transfer issues that were beginning to cripple previous buses with the introduction of Microsoft Windows. To improve performance, the architecture was directly linked to Intel CPU control lines making the bus exclusive to Intel machines.

Bus Type	Data	Address	Freq.	Throughput B_{IOB} (B_{avail})
VL	32 bit	32 bit	33MHz	132MBs^{-1} (40-50MBs^{-1})

Table 4: VL Bus Performance
Throughput measured in bytes per second with the first value indicating the theoretical maximum and the second the sustained maximum normally achieved in an actual system.

Although the VL bus could support additional devices such as mass storage controllers or high speed network interfaces, it lacked sufficient signalling to completely replace expansion bus architectures such as EISA or MCA. This contributed significantly to its downfall.

2.5 PCI Bus

The *peripheral component interconnect* (PCI) bus [6]-[7] was introduced by Intel in 1992 and was a revolutionary design in that it was a *mezzanine* bus or a bus between buses. This meant that its architecture is not specifically linked to Intel processors, as with the VL bus, making it equally implementable on PC, Macintosh or RISC platforms.

Bus Type	Data	Address	Freq.	Throughput B_{IOB} (B_{avail})
PCI 32/33	32 bit	32 bit	33MHz	132MBs^{-1} (50-80MBs^{-1})
PCI 64/33	64 bit	32 bit	33MHz	256MBs^{-1} (95-155MBs^{-1})
PCI 32/66	32 bit	32 bit	66MHz	264MBs^{-1} (100-160MBs^{-1})
PCI 64/66	64 bit	32 bit	66MHz	528MBs^{-1} (200-320MBs^{-1})

Table 5: PCI Bus Performance
Throughput measured in bytes per second with the first value indicating the theoretical maximum and the second the sustained maximum normally achieved in an actual system.

The PCI bus has a CPU independent frequency of either 33MHz or 66MHz and bus widths of 32 or 64 bit. Commodity PC systems currently implement PCI at 33MHz with 32 bits of data. It is an adaptable bus which usually hosts most other external buses such as SCSI, USB and Firewire. It uses the concept of bridges to connect different buses or indeed to extend the PCI bus. The *north bridge* (NB) interfaces the PCI bus with its host computer's architecture whereas the *south bridge* (SB) interfaces other buses such as ISA, USB or even PCI. Although the concept of a PCI-to-PCI bridge seems unnecessary, it is required due to signal frequency and skew limitations across a *printed circuit board* (PCB). If run at 33MHz a PCI bus can support up to 8 devices whereas at 66MHz only 4 or 5 devices can be supported. Thus such a bridge must be installed if six or more devices are required at 66MHz.

2.6 AGP Bus

The *accelerated graphics port* (AGP) is an extension of the PCI bus with the addition of extra signalling lines. However, it is physically, logically and electrically independent of the PCI bus and has been optimised for 3D graphics applications. There is normally only one device on an AGP bus thus the device does not have to compete with any others and as such may better be described as a port. Thus actual data transfer rates can approach the theoretical maximum when moving data from main memory to the AGP device. However, AGP devices are not designed to return data to main memory thus data transfer rates in the reverse direction rarely exceed half the theoretical maximum.

Bus Type	Data	Address	Freq.	Throughput B_{IOB} (B_{avail})
AGP 1×	32 bit	32 bit	66MHz	264MBs^{-1} (100-264MBs^{-1})
AGP 2×	32 bit	32 bit	66MHz	528MBs^{-1} (200-528MBs^{-1})
AGP 4×	32 bit	32 bit	66MHz	1056MBs^{-1} (400-1056MBs^{-1})

Table 6: AGP Bus Performance
Throughput measured in bytes per second with the first value indicating the theoretical maximum and the second the sustained maximum normally achieved in an actual system.

AGP has an extension to the PCI timing cycle which supports multiple data transfers in a single cycle as indicated by the multiplier.

2.7 Future Buses

The future of computer bus technologies remains unclear, however this section will examine what appear to be the key architectures. It is probable that one, or perhaps even two, of these architectures will become dominant within the next few years. Due to the independence of each architecture from connected component type, it is not inconceivable that all will survive to a varying extent in suited applications.

The successor to the PCI bus architecture is due to be deployed in server architectures and is called *PCI extended* (PCI-X). It enhances PCI performance by improving equivalent bus efficiencies by around 20%. Unfortunately, its high speed of operation at 133MHz limits the bus to one or two devices before another bridge must be inserted.

Bus Type	Data	Address	Freq.	Throughput B_{IOB} (B_{avail})
PCI-X 66	64 bit	32 bit	66MHz	528MBs^{-1} (240-384MBs^{-1})
PCI-X 100	64 bit	32 bit	100MHz	800MBs^{-1} (360-584MBs^{-1})
PCI-X 133	64 bit	32 bit	133MHz	1064MBs^{-1} (479-776MBs^{-1})

Table 7: PCI-X Bus Performance
Throughput measured in bytes per second. As these buses have not been implemented, available data transfer rates are estimates based on predicted performance figures.

The days of computer buses using address and data lines seem to be numbered. Current thinking indicates that a return to high speed serial transmission is about to be made as

the cost and complexity of hardware required to maintain highly parallel buses at every increasing speeds becomes prohibitive. Probably the best example of this is RDRAM recently released by Rambus Inc (see Section 3.2). However, the biggest problem that bus type interfaces face is that they are inherently unsuitable for high speed processing of streamed data. This is because multiple devices share common signalling lines, resulting in a considerable drop in effective data transfer rates when more than one device is streaming data. Thus replacement bus architectures are based on the mature technology of packet switched networks [8] only using a much simplified protocol. There are three major contenders at present: HyperTransport, RapidIO and 3GIO.

HyperTransport [9] is being developed by AMD and uses multiple independent data channels, each of 8 bits wide, to create data paths of up to 32 bits wide. These 8 bit data channels each have their own clock signal which reduces the chance of skew and allows faster data transmission rates. However, the clocks must be synchronised if multiple 8 bit data channels are used to create virtual data channels of 16 or 32 bits. These channels are full duplex, transferring data on both rising and falling clock edges. Link efficiencies are expected to be around 80%.

Unlike previous buses, HyperTransport is designed to interface with any *integrated circuit* (IC) on a PCB, be it a CPU, RAM or I/O, and appears to such hardware to be a PCI bus.

Bus Type	Data	Max. Packet	Freq.	Throughput B_{IOB} (B_{avail})
HyperTransport	32 bit	64 bytes	1.6GHz	25.6GBs^{-1} (15.4-20.5GBs^{-1})

Table 8: HyperTransport Bus Performance
Packet switched architectures put the destination address into the packet, thus there are no address lines. Throughput measured in bytes per second. As this bus has not been implemented, the available data transfer rate is an estimate based on predicted performance figures.

RapidIO from Motorola is another packet switched contender for the next generation computer architecture. It again groups 8 data lines with one clock line and quadruples data throughput by a combination of duplex communication and data sampling on both clock edges. Link efficiencies are expected to range from 50-90%, depending on packet size.

Bus Type	Data	Max. Packet	Freq.	Throughput B_{IOB} (B_{avail})
RapidIO	16 bit	256 bytes	1.0GHz	8.0GBs^{-1} (4.0-7.2GBs^{-1})

Table 9: RapidIO Bus Performance
Packet switched architectures put the destination address into the packet, thus there are no address lines. Throughput measured in bytes per second. As this bus has not been implemented, the available data transfer rate is an estimate based on predicted performance figures.

3rd generation I/O (3GIO) [10] is considered to be the direct successor to PCI and is endorsed by Intel and the PCI-SIG group. At the time of writing, there has been speculation that this bus will eventually be called *PCI Express*, however this has not yet been confirmed. Although initial development lagged behind both HyperTransport and RapidIO, it has recently begun to catch up rapidly. 3GIO uses high speed duplex lanes to transfer data that are arranged into groups of up to 32. Although initial clock rates are 2.5GHz it is expected that the bus will soon reach 10GHz. 3GIO multiplexes the clock onto the data signal, as does Ethernet, using a technique patented by IBM called 8B/10B. This significantly reduces skew but also effective bandwidth by exactly 20%. Factor in other protocol overheads, and the architecture is expected to achieve 65% efficiency. 3GIO is generally considered to be the successor to HyperTransport and not a direct competitor.

Bus Type	Data	Max. Packet	Freq.	Throughput B_{IOB} (B_{avail})
3GIO	32 bit	256 bytes	10GHz	80GBs^{-1} (40-52GBs^{-1})

Table 10: 3GIO Bus Performance
Packet switched architectures put the destination address into the packet, thus there are no address lines. Throughput measured in bytes per second. As this bus has not been implemented, the available data transfer rate is an estimate based on predicted performance figures.

2.8 Overview

Figure 1 summarises the computer buses discussed in this chapter, giving an overview of theoretical best performance B_{IOB} and actual performance. Note that the actual available bandwidth B_{avail} is an average of minimum and maximum practically available bandwidths.

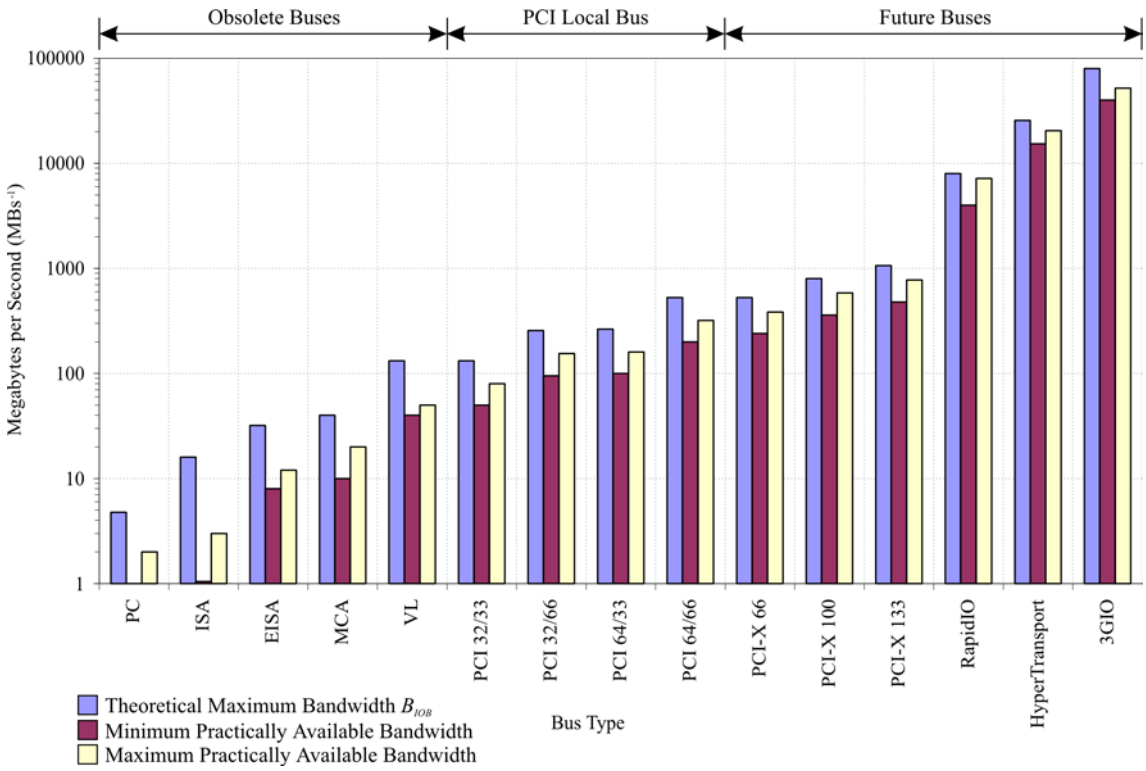

Figure 1: Computer Bus Performance Comparison
Values plotted on a logarithmic scale. The bus era is indicated at the top of the figure. Note that AGP is not included here as it is a dedicated bus rather than general purpose.

Bus eras are indicated on the top of the graph with obsolete buses ranging from 1981 to 1991, PCI local bus from 1992 to 2002 and future buses from 2003 to ~2010. Available bandwidths have been predicted for future buses as they have not been implemented yet. Therefore the figures for future buses are, at time of writing in 2002, purely estimations.

2.9 Data Transfer Impact

In earlier non-local bus based architectures, data transfer required that the CPU was halted as the address and data lines used were the same as those for main memory. This also meant that all data transfer was at the speed of the I/O card even if the main memory bus could run significantly faster than the I/O bus. Therefore sustained data transfer at maximum bus capacity would prevent any processing from occurring at all.

Today's local bus based architectures, such as PCI, alleviate this problem in two ways. Firstly, the main memory bus is independent of the I/O bus and linked through the north bridge. Thus data transfer can simultaneously take place on both buses. Secondly, *speed matching buffers* (SMB) are used in the bridges. Data from a typically slower I/O device is buffered until a reasonable amount is accumulated to make a transfer to main

memory worthwhile. In a similar manner, data is buffered from main memory to the I/O bus thus allowing it to be transferred at high speed from main memory rather than at a speed artificially matched to that of the I/O bus. Such buffers are obviously architecture dependent, but for PCI 32/33 they are usually in the region of 256 bytes. Since the PCI specification guarantees a maximum wait time before a device is serviced of 3µs, bus usage levels of up to ~65% can therefore be sustained with no loss of data.

2.10 PCI Bus Throughput Measurements

The sustained data transfer rate figures in this section presume that nothing else is using the data bus and that the appropriate device is operating as a bus master. In real life situations, devices such as graphics cards, sound cards and hard disk drives use a percentage of the total theoretically available bandwidth B_{IOB}. Experiments have been carried out to measure bus usage of a PCI bus in a commodity PC (1.4GHz Athlon, 0.5GB RAM, AGP 4×) under different sets of circumstances. The measurements were taken using an *RD2 PC Geiger* [11] as shown in Figure 2.

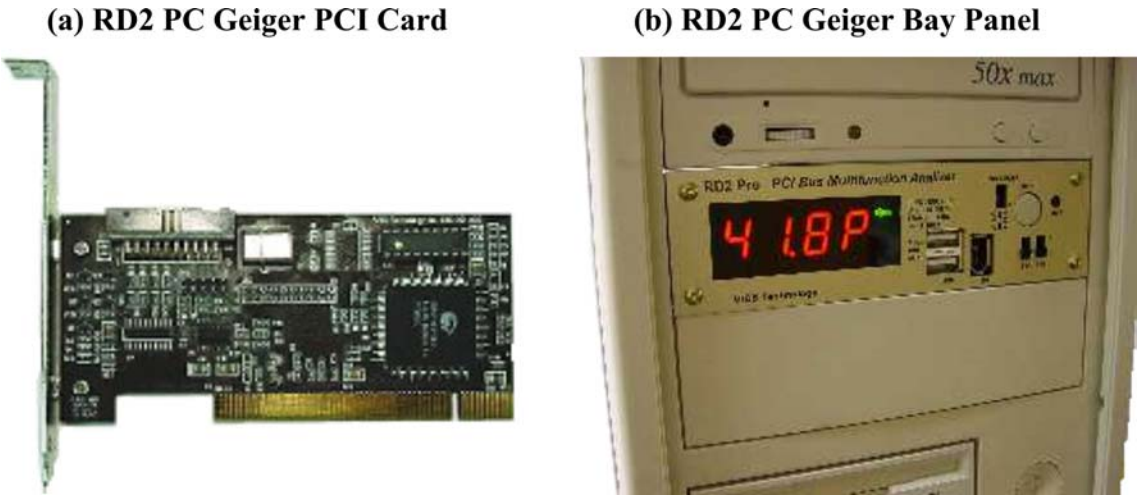

Figure 2: RD2 PC Geiger
The card in (a) uses a PCI slot to monitor bus activity, the result of which is displayed on an externally visible bay panel as seen in (b). This device is OS independent.

The PC Geiger monitors activity on the PCI bus using the least significant byte. It assumes that if data is present there then the entire bus is in use. Although not technically exact, to all intents and purposes this is correct as any unused bandwidth in such a cycle would be lost. The device measured the following peripheral loads B_{per} under a specific set of circumstances:

- 1 to 5% ±1% of B_{IOB} load while running the system idle process or during activities where the majority of processing is RAM based. Such processing entails activities such as word processing.
- 15 to 25% ±1% of B_{IOB} load was obtained under reasonable load situations. Such a situation is defined as a fair amount of both processor and I/O activity. For example, load the operating system, opening a large software package etc.
- 50-80% ±1% of B_{IOB} loads were achieved thorough intensive disk and network activity. This was induced by launching multiple processes with I/O requests such as copying large files both between multiple local drives and a network drive simultaneously. Note that peak transfer rates of ~80% were noticed for short periods of time. This is analogous with bus mastering.

As predicted, the peak bandwidth usage did not exceed an average of 60% ±5%. These results help verify the theory presented in this section.

2.11 Conclusion

I/O bus throughputs have a tendency to be quoted as theoretical maximum values and thereby fail to take account any practical considerations [12]. For example, the ISA bus normally requires a minimum of two clock cycles to complete an I/O operation however wait states may extend this to as much as eight. Therefore practical data transfer rates on an ISA bus rarely exceeded $500kBs^{-1}$ [13], even though ISA has a quoted bandwidth of around $16MBs^{-1}$. Fortunately, such poor efficiencies are continually improving with each successive generation of commodity bus architecture. For instance, AGP buses are now capable of sustaining four sets of data transfers during a single clock cycle with an efficiency that approaches 100%.

Aside from the efficiency argument, there lies yet another minefield – algorithmic efficiency. What is done with the data when it reaches main memory? Given that the memory in a computer system is finite, sustained transfers from high bandwidth I/O devices will eventually fill main memory. Assuming that some processing is performed on any input data stream, a result will be generated that will either need to be stored or retransmitted. Such actions also consume available I/O bandwidth and must be considered if throughput is to be sustained.

Creative enhancement of existing buses, such as PCI to AGP, indicates that higher performance can almost always be squeezed out of an implementation. Unfortunately AGP does not support anything other than a graphics card, although it very existence shows that the concept of a dedicated and optimised bus for high bandwidth device(s) is both sound and viable. However, implementing such a system is encroaching on the realm of custom built hardware and is not the enhancement of a commodity architecture.

3 Computer Architectures

The commercial microprocessors of today offer impressive raw performance and provide an attractive option in the assembly of cost-effective parallel machines. Significant effort has been directed into the development of efficient parallel architectures, networks and interconnects to exploit such components. However, these scalable architectures have yet to fully address communication delays between tasks on different processors and, perhaps more importantly, between processors and main memory modules.

It has been reported [14] that although processor performance improves at a rate of 50 to 100 percent per year, memory performance remains dismal at approximately 7 percent per year. Even network performance has had a greater improvement than that of memory. In an effort to optimise performance, system designers will inevitably have to overcome the limitations imposed by memory architectures. It is therefore common practice to incorporate increasingly deep memory hierarchies through multiple levels of cache to maintain the balance between processor and memory system performance [15]. One consequence of these deep memory hierarchies is that cache miss latencies have become extremely large when counted in processor clock cycles. Thus, if an application is to attain good performance, it must exhibit memory reference behaviour that exploits caches as well – in other words the memory access pattern must exhibit high spatial, temporal and processor locality [16].

As the gap between processor data requirements and main memory bandwidth continues to widen, economic considerations will prevent the use of dramatically faster main memory and impede the use of high bandwidth interconnect technology in the commodity class of machines. Therefore, network subsystems must take care to minimise the number of trips that any network data takes across the CPU to main memory data path [17]. Failure to do so may result in memory bandwidth being overwhelmed as faster network interfaces become not only available but commonplace.

To aid understanding of the issues in commodity computer architectures, the four major memory hierarchies of such architectures must first be outlined:

- **Registers:** These can be considered as small amounts of full speed memory within the CPU, the presence of which is essential to it operation. They are normally

managed by the compiler which decides what value is to be kept where at every point in the program. Registers are normally few in number, hardly ever reaching a fraction of 1kB of data storage.

- **Cache:** This is a small amount of fast memory located close to the CPU that holds the most recently accessed code or data. Caches generally comprise of *static random access memory* (SRAM). SRAM is built up of an array of switches where data is represented by switch state. SRAM is low latency, but cost is high as a fair amount of silicon real estate is required by a single bit. A cache hit occurs if the CPU finds the requested data item in the cache, otherwise the data has to be fetched either from the next level of cache or main memory. There are two major levels of cache: *level 1* (L1) and *level 2* (L2). L1 cache resides on the CPU die, running near, but not always at, core frequency with a size in the region of 16-256kB. L2 cache resides on the motherboard, physically close to the CPU. This allows L2 cache to be slightly larger, with sizes of typically 256-1024kB, but it also tends to be slightly slower due to the distance from the CPU. As an aside, a third level of cache, *level 3* (L3), is beginning to appear in the specifications of new CPUs. However, as of 2002, there is not any large scale implementation of this level of caching.
- **Main memory:** This is made up of *dynamic random access memory* (DRAM), or a variation thereof, which serves both the I/O interface and the demand of caches. It has significantly larger storage capacities and longer latencies than those seen in SRAM, all achieved through the use of micro capacitors to store data. The silicon real estate required to store a bit in DRAM is significantly smaller than that required by the switches in SRAM. However, capacitors are subject to leakage and therefore main memory must be refreshed every few milliseconds, usually by integrated control circuitry. The use of capacitive devices to store data also introduces intrinsic limitations to device access speed. However, their simple design structure leads to economic viability and extremely high density, making DRAM the definitive choice for main memory. Typical DRAM main memory sizes are between 128-512MB.
- **Virtual memory:** All systems generally require a greater number of memory locations than are available in the main memory. *Virtual memory* (VM) allows an entire portion of address space to be temporarily stored on a large magnetic or optical disk usually tens of gigabytes in size. The most frequently used sections of main memory are kept in main memory and the others are swapped in and out as required.

However, VM is no substitute for main memory as its access times have consistently remained around three orders of magnitude longer that that of DRAM for the past few decades.

3.1 Cache Architectures

SRAM caches work by exploiting the principle of spatial locality of reference by pre-fetching a cache line or block, otherwise defined as a fixed amount of contiguous data, from the referenced location. The widening gap between the processor and memory has lead to a keen interest in multi-level caches. Indeed, the fact that L1 and L2 caches exist at all is testament to this. Although adding additional levels in the hierarchy is relatively straightforward, it significantly complicates design and performance analysis [18]. This section outlines the factors that influence the design and performance of caches:

- **CPU Scheduling:** This may cause the execution of processes to be interleaved. When processing resumes, cached portions of data will most likely have been replaced by other data. In multiprocessor systems, processing can be simply resumed if processes have their own data cache. Situations such as hardware interrupts and the events they signal can also trigger CPU rescheduling. More common though are situations in which scheduling occurs during the processing of a data unit as it is passed to another thread or *queued*. Queuing typically occurs in certain protocols and protocol layers as data is moved from the device driver interrupt handler to the top half of the driver at the user or kernel boundary.
- **Cache Write and Read:** Most uniprocessor data caches employ a write-through policy such that every store operation requires a write to main memory. Although write buffers are used in tandem with write-through caches, many consecutive read and writes could cause the CPU to stall due to queuing in the memory system [17], [19]. Misses on data writes vary in importance depending on allocation/management policy and size of the cache. Since data write requests are often smaller than a block, which is of an arbitrary size and often sequential, spatial locality is an important issue [20]. The size of data reads is dependent on the I/O libraries used and the application. The cost of a read miss is the overhead of the context switch, the penalty of lost local state (in additional to CPU memory cache misses and page faults when restarted), and any idle cycles resulting from disk access [20], [21].

- **Cache Lookup:** Two types of cache lookup exist - those that are virtually indexed and tagged and those that are physically tagged. The former does not require a virtual-to-physical address translation to access cached data. Also, cached data from virtually shared pages cannot remain valid across protection domain boundaries. Physically cached data does not have this problem but does require that a *translation lookaside buffer* (TLB) entry is active for the page that contains the referenced data [22].
- **Cache size:** Cache size depends on a cost to performance ratio: small caches for low cost machines, large caches for high performance machines. Loading and storing every word during any data modify or inspection step requires the cache size to be at least twice the size of the loaded data unit. Cache size requirements are also increased by cache line collisions due to limited associativity of the cache and by access to program variables during and between data manipulation [22].

3.1.1 Data Caches and Translation Lookaside Buffer

It is well know that cache access times and cache misses are the most influential performance constraining factors [23]. However, several other factors also affect the effectiveness of both cache and TLB, such as size and organisation, locality of data access and processor scheduling [17], [23]-[24].

To quantify the effectiveness of the data cache in handling network data, the amount of network data resident in the cache after *operating system* (OS) processing needs to be determined. This measure of cache residency indicates the potential benefit to a user process due to caching of network data, thereby avoiding CPU to main memory transfers. It also provides an upper bound on the benefits provided by cache when the network data is processed without the need for copying.

In the case of systems that support the manipulation of high-bandwidth data, the CPU is required to inspect and possibly modify all the information in single data unit potentially multiple times. It has been reported [17] that data caches are not overly effective in reducing CPU to main memory transfers in the case of network data. The experiments demonstrated three significant contributors to limitations in network buffer data access times. First, *direct memory access* (DMA) used by other system devices may increase contention for access to main memory. Such contention effectively increases the amount of time it takes to load data into the processor's cache, thus

increasing average data access times. Second, on processors with physically tagged caches, contention for TLB slots, caused by network data fragmentation over virtual memory pages, produced significant access overheads. Finally, even with TLB slot contention eliminated, data caches have minimal effect in reducing CPU/memory traffic for network data.

Later analysis [23] of the effectiveness of data caches and TLB in processing network I/O has shown that operating system structure impacts cache behaviour. These experiments suggest three general rules for network subsystem design. First, since TLB usage has significant importance, the OS should be designed to take advantage of group TLB entries. Second, context switches adversely effect cache performance. Therefore, network data should be brought into the cache as late as possible, such as when first accessed by check summing, and that all access to network data should happen in the same context such as the processes protection domain. Finally, network buffers should be laid out contiguously and allocated in a *last-in-first-out* (LIFO) manner so as to minimise self-interference.

The effectiveness of data caches also depends on locality in executed software. Analysis of the amount and characteristics of memory reference locality [25] allows understanding of the reasons for large amounts of cache misses. In looking at how much memory access belongs to immediate spatial sequences, it was observed that many accesses are to data blocks that have recently been brought into cache. The study indicated two principle sources for spatial data sequences. First, accesses to data structures that are larger than the block size or that cross block boundaries, where the spatial sequence is often generated by a different load instruction. Second, accesses to an array or linked data structure, where the spatial sequence is often generated by the same load instruction within a loop. Thus, for many applications, increasing the block size would lead to lower utilisation of the cache.

The use of dynamic caching is also ineffective [19]. In this study, it was found that the percentage of live data in caches was typically under 20%. Dynamic object allocation also usually stresses the randomness of data memory usage since the variables of a dynamic cache working set are, in general, distributed stochastically in both virtual and physical address space. However, since caches and TLBs are typically not fully associative, the effects of stochastically structured working sets is not obvious. Thus, random influences result in an increase of cache conflicts and reduced hit rates [26].

Although a plethora of literature is abound with arguments for particular types of caching, it cannot bridge the growing processor to memory gap. Therefore, to preserve the bandwidth on the data path from the network device through the OS and application to a sink device such as a display, multiple transfers of data between the CPU and main memory must either be avoided or reduced. This suggests that serious problems lie in the implementation of a commodity architecture that uses a high bandwidth optical networking strategy.

3.1.2 I/O Subsystem Caching

Technology advances continue to rapidly increase storage device capacities but the difference in access times between main memory and such devices still remains around three orders of magnitude. This limits overall performance due to the I/O subsystem's response time leaving the CPU under utilised.

Typically, a data storage hierarchy comprises of main memory, disks and optical or tape drives. The OS uses device drivers to route I/O requests to device controllers, which in turn transfer data to memory via bus adapters. To alleviate mechanical latencies, I/O subsystem designers have increased disk rotation speeds, used intelligent scheduling algorithms, increased I/O bus speed and implemented caches in various places along the I/O stream.

Unlike processor caches which have request sizes of up to 32 bytes, low access times and are implemented in hardware, I/O caches have substantially greater request sizes, data sets and backing store latencies. To reduce cost, I/O caches are usually implemented and managed in software with limited hardware assistance.

In an I/O data path, the location of the cache is an important issue [22]. There are three possible locations for a cache: in each host, in each storage device such as a magnetic drive or in a multi-device storage controller. For I/O subsystems there are two possible configurations. Low-end systems use dedicated cache controller embedded in the drive itself which is tasked to read, write, prefetch and transfer data. On the other hand, high-end mainframes typically use a multi-device controller to perform complex operations such as command reordering, seek optimisations and intelligent caching.

Most studies have agreed that host caching significantly reduces the host's response time resulting from network or bus latency and disk access time [27]. Host caching reduces latency due to faster data access and lower cache pollution. With intelligent

prefetching and the ability to dynamically vary cache size, low miss rates for a given cache size are achieved thus reducing contention for storage devices or the bus interconnect. Some disadvantages of host caching are data volatility, ensuring cache coherency among different hosts' caches, assuming that they are write-back caches, and CPU overhead for cache management. If the CPU is already heavily utilised, then host caching could also degrade system performance.

Caching at the controller in a multi-device or host-based adapter has several advantages. Functions such as DMA transfers, command queuing and other optimisations are performed by the controller completely transparent to the host. Cache coherency is improved since the controller sees the I/O streams from all hosts and since data from multiple hosts resides in the controller cache itself. Cache pollution at controller level is minimised since only the most frequently used portions of the prefetched data from each drive migrates upward to the controller. More importantly, effective allocation of controller cache is possible among the different drives based on their utilisation and workload profiles.

The drawback of multi-device controllers is higher miss rates compared to a host cache due to the controller seeing a random mix of data requests from different hosts. For multi-device hierarchical storage controllers, write-back caching is more difficult to implement with dual-pathing due to the difficulties of keeping both controller caches consistent in case of a fail-over.

The majority of commodity drives have embedded controllers with on-board caches (drive-level caching) which is similar to caching at a single device driver. However, it remains an open question as to how well such caches perform in conjunction with a multi-device controller or a host-based adapter cache. Similarly, whether caching at multiple levels within a system is more effective than a single large cache placed strategically in the I/O path is not clear. Further work is necessary to evaluate the performance of multilevel I/O caches.

3.1.3 Buffering

Buffering can have significant impact on *network interface* (NI) performance. A large amount of buffering could be considered necessary for NI adapters because of four main reasons. Buffering smoothes out temporary mismatch rates on loosely coupled microprocessors, network switches and a variety of user-level communication protocols

thus creating a more balanced system. With larger amounts of buffering, a processor can relax its monitoring of the NI status. The degree of multiprogramming can also be increased. Finally, large buffers may avoid clogging due to unreliable flow-control schemes in networks such as that seen in the Myricom Myrinet [28]. Note that the NI cannot rely on network switches and routers to provide a high level of buffering so the location of buffers and processor involvement in buffering messages is important.

The effects of loading data into the cache too early can decrease overall system performance by evicting live data from the cache. This calls for a substantial amount of buffer space in the NI adapter [17]. In the case of DMA, incoming data can be buffered into main memory. Using main memory to buffer network data is advantageous since a single pool of memory resources can be dynamically shared among applications, OS, and the network subsystem.

Processors require rapid access to NI buffers. Also, NI buffers need to be plentiful. This presents a conflict since having large amounts of dedicated NI memory to buffer messages is not economical. In contrast, main memory can support plentiful buffering but may not allow rapid data transfer. This paradox can be compromised by logically allocating message buffers in coherent, shared memory but physically locating NI buffers in processor caches, main memory, or NI memory. When NI buffers are located in dedicated NI memory and main memory, then either the processor or NI must manage the transfers. Otherwise, the system will be clogged and may even cause the system to deadlock [28].

Buffer management is required since buffer editing occurs frequently in network protocol implementations and as such it is needed in OS network subsystems. Buffer editing comprises of operations to create, share, clip, split, concatenate and destroy buffers. The buffer manager is restricted to a single protected domain, typically a kernel. In most systems, a software copy into a contiguous buffer is necessary when a data unit crosses a protection domain boundary. To facilitate copy-free buffer editing a lazy evaluation of buffers is used [17].

3.2 DRAM Types

This section outlines and quantifies the variations on DRAM technology used in main memory. The majority of today's computer systems tend use *synchronous* DRAM (SDRAM). This is a variant of DRAM which transfers data in lock step with the rising

edge of an applied square wave clock signal, the same clock signal that operates the memory control chip set. Since the timing of SDRAM is predictable, data can be transferred at a much higher rate than is possible with older DRAM technologies. In an attempt to boost the bandwidth of SDRAM, an extension has been developed called *double data rate* (DDR) SDRAM which transfers data on both rising and falling edges of the square wave in a similar manner to AGP 2×.

Despite the relatively high access latency of DRAMs, memory subsystems achieve high bandwidths using some form of pipelining such as *interleaving* or *page mode*. Next-generation workstations are expected to support terabit per second network adapters that will enable transfer of data into main memory at network speeds.

By increasing the memory data path width, peak memory bandwidth can be improved. However, cache line size must also be increased proportionally to achieve a substantial increase in CPU bandwidth. The cache line sizes that result in optimal hit rates are typically too small to achieve a substantial fraction of the peak memory bandwidth at present. Thus, for small transfer sizes, start-up latency contributes significantly to memory transfer times [29].

One approach to reduce transfer latencies has been to integrate some form of a cache with a DRAM [30]. The integration of a second-level caches use large cache lines which are connected to the DRAM by wide data paths. However, as for any cache, the hit rate of these components still depends on locality of reference. More recently, a pipelined microarchitecture that allows direct control of all DRAM row and column resources concurrently during data transfer, known as direct *Rambus* DRAM (RDRAM), has been proposed to bridge the current performance gap [31]. As noted [32], this new architecture merits further study as its peak bandwidth attains 1.6GBs^{-1}, which is a significant improvement when compared to conventional DRAM. Its architecture, interface and timing are unique and provide a greatly improved bus efficiency. In the advent of optical and electro-optical components, photonic networks and so on, this device has the potential to be a platform for optical DRAM. However, at time of writing, RDRAM appears to be loosing out to DDR-SDRAM technology due to licensing fees, technical problems and double the cost per megabyte for marginally better performance.

The first metric of memory performance examined here is bandwidth B_{mem}. Table 11 lists the throughputs of some commodity DRAM modules and their efficiency ξ_{mem}.

DRAM Memory	Data	Freq. (Hz)	Efficiency (ξ_{mem})	Throughput (B_{mem})
PC66 SDRAM	64 bit	66MHz	0.60	533MBs^{-1}
PC100 SDRAM	64 bit	100MHz	0.60	800MBs^{-1}
PC133 SDRAM	64 bit	133MHz	0.60	1060MBs^{-1}
PC150 SDRAM	64 bit	150MHz	0.60	1300MBs^{-1}
PC1600 DDR-SDRAM	64 bit	100MHz	0.40	1600MBs^{-1}
PC2100 DDR-SDRAM	64 bit	133MHz	0.40	2100MBs^{-1}
PC2700 DDR-SDRAM	64 bit	167MHz	0.40	2700MBs^{-1}
PC800 DDR-RDRAM	16 bit	400MHz	0.80	1600MBs^{-1}
2 Ch. PC800 DDR-RDRAM	2×16 bit	400MHz	0.80	3200MBs^{-1}

Table 11: DRAM Bandwidth

Efficiencies reflect random access. Throughputs are measured in bytes per second.

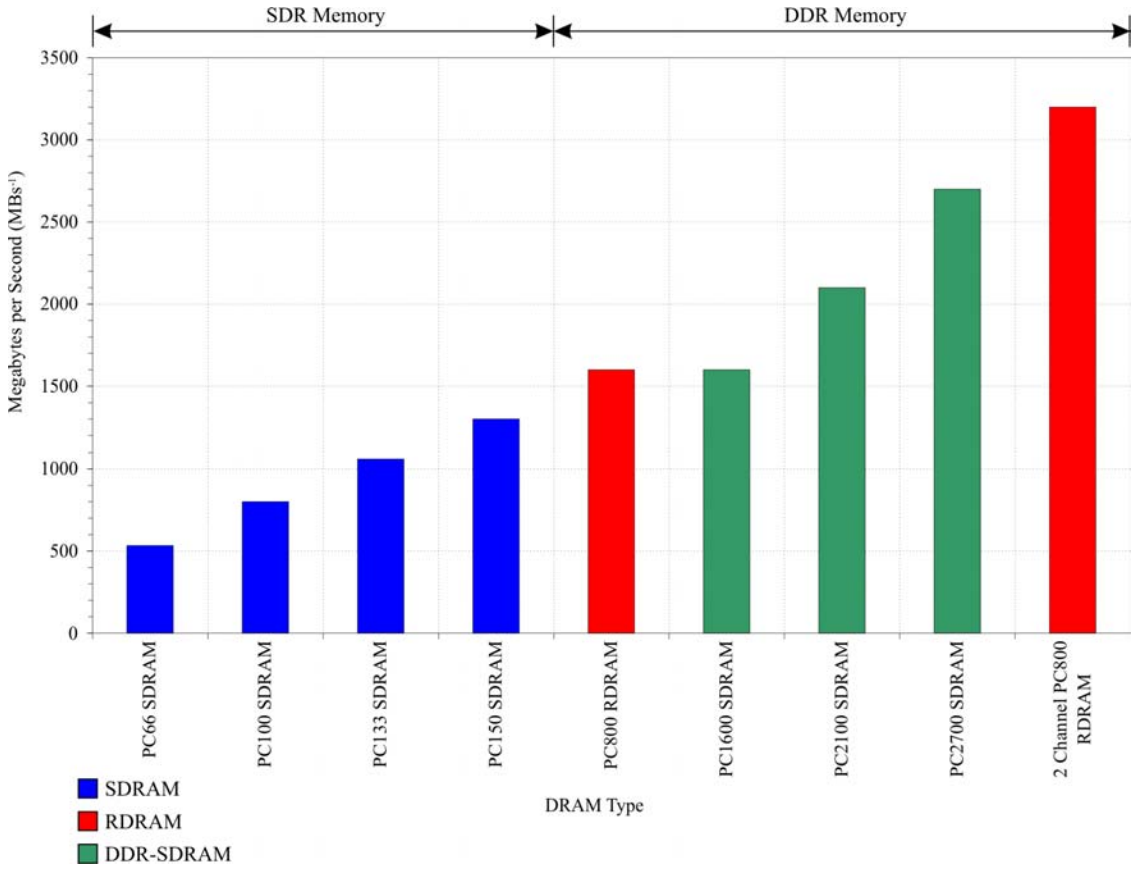

Figure 3: DRAM Bandwidth Comparison

Data rate type, whether single or double, is indicated at the top of the figure.

Figure 3 graphs the information found in Table 11. The bandwidths shown here are the peak achievable given 100% memory bus efficiency. Needless to say, such a level of efficiency is unachievable in real world situations however the average efficiency for each architecture is given. In fact the sources of variance are great and range from memory module components and *column address strobe* (CAS) latency through to CPU speed, cache size and motherboard chipset. Therefore a sustained efficiency variance of as great as ±0.2 may sometimes be encountered. For further analysis of DRAM technologies see [33].

The second metric of memory performance is its latency L_{mem}, which is a combination of *address latency*, *component latency* and *data delay*. These three latencies occur in the order written and their values are highly dependent on memory architecture. They are defined as:

- **Address Latency:** This is the time required for the data address to reach every pin on every memory device. In an SDRAM based system this is almost always two clock cycles, which is referred to as the *two cycle addressing problem*. Unfortunately DDR-SDRAM based systems have not managed to overcome this limitation and still require two full clock cycles, rather than two half clock cycles as may have been envisaged. It is possible to reduce this figure by tightly integrating memory however this is only really feasible in custom designed systems such as graphics cards. RDRAM eases this problem by reducing the period of a clock cycle significantly so that the RDRAM delay of 4 clock cycles is only 10ns, a feat which uses both clock edges to transmit the address.
- **Component Latency:** Regardless of interface method, the DRAM components used are identical and as such have a fixed latency. This latency is defined as the delay between the moment the *row address strobe* (RAS) line is activated and when the first data bit becomes valid. In an SDRAM based system, this is the sum of t_{RCD} and *CAS latency* (CL) values. At this point a full bus width of data has just been transferred. However, this is not the exact time that it takes for data to appear on the data bus. Since SDRAMs are synchronised to the memory clock, an equivalent or greater time period must have elapsed before data is valid. For example, interfacing a 40ns component to a 100MHz bus requires 4 cycles of 10ns before a 40ns delay has been achieved. However, if this were a 133MHz bus with a cycle time of 7.5ns then 6 cycles would be required totalling 45ns. One less cycle would result in a

delay of 37.5ns which is less than the 40ns component delay and therefore not long enough to guarantee that the data would be valid on the component's output pins. In RDRAM, component latency is computed slightly differently. It is the sum of both t_{RCD} and t_{CAC} plus one half clock cycle for the data to become valid on the data bus [34]. Current component latencies for both systems are in the 35ns to 45ns region (2002) and are not foreseen to decrease rapidly within the next few years.

- **Data Latency:** Assuming a favourable memory access pattern, the remaining information is finally transferred at the theoretical bandwidth of the bus. In the case of conventional SDRAM memory this is one bus width of data every cycle. DDR-SDRAM is capable of transferring two memory bus widths every cycle however any spare half cycles will be lost as signalling operates using whole cycles only. RDRAM also transmits two memory bus widths every cycle however a precharge time is required before another access can occur to ensure that the bank and associated sense amps are charged.

The total latency is the sum of these three latencies. In systems such as DDR-SDRAM this must be matched to the next clock cycle since addressing cannot make use of half clock cycles. Table 12 shows the combined latency for the return of 32 bytes of data to the CPU, known as a *cache line fill*, using different DRAM interface methods. Cache line fills are a common metric since the presence of L2 cache memory means that transactions occur as bursts of fixed sized memory blocks with continuous address ranges covering 32 consecutive bytes of memory in a Pentium or equivalent class processor.

It should be noted that exact latency prediction is complex for RDRAM and is affected by additional parameters such as bank precharge times and component location as the serial nature of RDRAM means that extra cycles may be required to access remote memory modules. For further information on RDRAM see [34]-[38].

Research has indicated that as DRAM access latencies grow, a cache miss may eventually take hundreds of CPU cycles to satisfy [19], [32]. Several latency tolerance techniques have been proposed to increase a processor's memory bandwidth. However, these solutions often trade improved latency at the expense of increased traffic, or higher bandwidth for increased latency. At any rate, their relative affects on memory access performance are important.

DRAM Memory	Address	Component	Data	L_{mem}
PC66 SDRAM	2×15ns	45ns	3×15ns	120ns
PC100 SDRAM	2×10ns	40ns	3×10ns	90ns
PC133 SDRAM	2×7.5ns	45ns	3×7.5ns	82.5ns
PC150 SDRAM	2×6.7ns	40ns	3×6.7ns	73.3ns
PC1600 DDR-SDRAM	2×10ns	40ns	1.5×10ns	80ns
PC2100 DDR-SDRAM	2×7.5ns	45ns	1.5×7.5ns	75ns
PC2700 DDR-SDRAM	2×6ns	42ns	1.5×6ns	66ns
PC800 DDR-RDRAM	4×2.5ns	45ns	1.25×16ns	75ns
2 Channel PC800 DDR-RDRAM	4×2.5ns	45ns	1.25×8ns	65ns

Table 12: DRAM Latency
All values shown here are matched to the nearest clock cycle. Latencies measure a complete cache line fill of 32 bytes. Component latencies for SDRAM are assumed to be 40ns when unmatched to clock frequency and 45ns for RDRAM.

Various high performance DRAM interfaces have been proposed with lower cost in mind. Generic designs that target main memory as well as graphics are typically single ported. Others are dual ported and concentrate specifically on enhancing rendering performance for graphics applications [39]. However, such solutions remain in the realm of custom and not commodity components.

3.3 SRAM Types

SRAM memory differs from DRAM in that the latency figures for both read and write are small to negligible. Indeed there are no-latency SRAMs available such as the *zero-bus-turnaround* (ZBT) architecture, capable of delivering 100% data-bus efficiency assuming that the system has previously filled the address pipeline. Therefore the latency figures for SRAM is simply the time required to transfer data to and from the SRAM device.

The drawback however is that it is not possible to integrate as much memory onto an SRAM chip as can be done with DRAM due to chip real estate considerations. Thus,

SRAM maintains a consistent price premium of two orders of magnitude over DRAM per megabyte

3.4 Data Transfer Modes

Communication performance on the I/O bus is highly dependent on the size of data transfer and processor involvement [28]. Two techniques commonly employed to move data between devices on the I/O bus and main memory are *programmed input/output* (PIO) and *direct memory access* (DMA). The units of data exchanged between host driver software and on-board device controllers is a physical buffer organised as a set of memory locations with contiguous physical addresses. The descriptors used in transmit and receive requests contain the physical address and the length of a buffer.

3.4.1 Programmed Input Output (PIO)

PIO requires the processor to handle all data transfer between main memory and an I/O device, thus only a small fraction of peak I/O bandwidth can be achieved. The adapter's control and data ports can be mapped either as cacheable or non-cacheable memory locations. Bandwidth is improved with cacheable locations as the I/O bus transfers can occur at cache line length. However, it may still be well below the peak I/O bus bandwidth and data cache flushing is required to maintain consistency with the adapter's ports.

3.4.2 Direct Memory Access (DMA)

DMA enables direct data transfer between I/O adapters and main memory to free the host processor from the burden of transfer. A DMA operation typically has three virtual arguments: a source, a destination and a size. Its function is to transfer a number of bytes from virtual address source to virtual address destination. DMA engines usually operate only on physical addresses. To operate on virtual addresses, the DMA hardware requires translation tables to translate virtual to physical addresses, while the OS software would need to keep these tables updated. This requires most DMA engines to have physical addresses as arguments.

The OS kernel traditionally does DMA management. The copying of network data between application memory and kernel buffers is usually achieved by statically allocating contiguous physical pages to the fixed set of kernel buffers. However, this approach does not generalise to a copy-free data path [17] since applications generally

cannot be allowed to hold buffers from a statically allocated pool. Before initiating DMA transfers, the host driver software must set up a map which contains the appropriate mappings for all the fragments to a buffer. When data is transferred to and from application buffers, it may be necessary to update the map for each individual message. Thus, even when virtual DMA is available, physical buffer fragmentation becomes a potential performance concern.

Host devices with cache subsystems do not guarantee a coherent view of memory contents after a DMA transfer into main memory [40]. CPU reads from cached main memory locations that were overwritten by a DMA transfer may return stale data. Consequently, the appropriate portions of the data cache must be invalidated by the OS [17].

For high bandwidth interconnects, the software protocol overheads and time spent handling cache misses to load data from a recipient's main memory to its highest level cache are major sources of communication latency [15], [41]. As traditional DMA requires the OS to perform many tasks to initiate transfer, the overhead can be in the region of thousands of instructions. This makes DMA highly inefficient for small data transfers that underlie fine-grain communications [28]. However, there are several innovative solutions to help alleviate this NI access bottleneck. Two examples are user level-block load and store by SPARC or *user-level DMA* (UDMA) [40], [42] and the coherent network interface [43]. Note that user-level DMA should not be confused with *ultra DMA* (UDMA).

In local bus based systems, the separation of memory and I/O bus allows a technique called *bus mastering* to be used. In this case, the device initiating DMA can request, and be given control of, the I/O bus by the host system controller. Information is then transferred to the north bridge and speed matched to main memory. This method is the only way to achieve the bus throughput figures quoted previously. It should be noted that DMA implementations are entirely system and chipset specific. Therefore there has been no attempt to quantify performance.

3.4.3 Managing Data Transfer

There are three alternatives for a network interface to transfer data. First the processor can initiate the transfer and allow the NI to manage the rest of the transfer. Second, the processor can actively manage all data transfer. Finally, a dedicated device or DMA

controller can manage the transfer. A scatter-gather capability in DMA based devices is important for reducing memory traffic.

Network interfaces that manage transfers typically require only the processor to initiate the transfer between the interface and the internal memory structures of a node. Using uncached loads and stores from processor to a memory-mapped register is rapid. However, the NI requires physical memory addresses of data buffers to obtain the data to be transferred. Unfortunately, users cannot provide authenticated physical addresses of data buffers without violating an OS protection domain. UDMA circumvents the need for authentic physical buffers by having users provide authentic physical addresses to the NI via a sequence of two-level instructions: an uncached store and an uncached load [42]. UDMA allows direct deposit of data into user data structures. The key limitation of UDMA is that there is no known generic technique to extend it to multi-programmed *symmetric multiprocessing* (SMP) [28]. The multi-programming problem faced by UDMA can be overcome by having processors and NI communicate via cacheable, shared memory [40]. The drawback of this approach is that the NI must poll the cached, shared locations to check for any new messages.

With processor-managed transfers, the NI design is simplified since the NI does not require authentic physical addresses to access a message. However, processor involvement incurs precious resources, which can be used for other purposes, primarily computation. Applying UDMA and cache block transfers avoids processor involvement. This reduces processor occupancy and allows overlap of computation with data transfer [28], [40] and [42].

3.5 Conclusion

Improvements to microprocessors other than latency reduction techniques will increase the bandwidth requirements across the processor module boundary. Faster processor clocks will run programs in a shorter time, increasing off-chip bandwidth requirements. *Integrated layer processing* (ILP) similarly reduces execution time but increases need bandwidth. *Speculative execution* also necessitates increased bandwidth as future microprocessors which rely on coarse-grained speculative threads to improve ILP, such as multi-scalar processors, increase memory traffic whenever they must squash a task after incorrect speculation. The emergence of embedded multiprocessors would increase the amount of data loaded per cycle in a manner similar to multi-scalar

processors due to shared-cache interference and from multiple, concurrent running contexts and threads. The primary barrier to the implementation of embedded multiprocessors will not be transistor availability but off-chip memory bandwidth. Thus if one processor looses performance due to limited pin bandwidth, multiple on-chip processors will lose far more performance for the same reason [19].

The advancement of other techniques to improve memory bandwidth includes new architectures that allow wider or faster connections to memory, larger and more efficient on-chip caches, traffic efficient requests, high-bandwidth or intelligent DRAM interfaces and memory-centric architectures. Optimised I/O subsystems dedicated to communication and memory control, multi-device I/O controllers, network interfaces and other host or bus adapters have also been developed to lower latency and allow higher bandwidths. However, with the introduction of optics, the bandwidth of the communication subsystem has ceased to be the bottleneck. Rather, the support for high speed transfer, and in particular, the management of the relatively large quantity of high speed buffers, has become the performance-limiting factor [44].

It is clear from the issues raised above that mitigating communication performance in one device aggravates the performance in another. This is largely due to the mismatch in the bandwidth available for the concerned device controllers and subsystems. Implicit in any system, the processor's bandwidth and clock rate governs the maximum processing rate. This indicates that even with the high-bandwidth intrinsic in optics, no matter what amount of hardware or software tweaking is done, unless both inter- and intra- processor memory device bandwidths accrue simultaneously, performance trade-offs are inevitable.

Perhaps the key question is not whether providing enough memory bandwidth with each new generation is possible, but whether providing enough for each new generation is cost effective. In particular providing enough pins at a low enough cost will be a significant challenge in the future.

4 Optical Technology

This chapter examines the optical technologies that enable optoelectronic interconnection. It summarises their mode of operation, development and intrinsic limitations.

4.1 Emitters

A component is considered to be an emitter when photons are produced within the device. The wavelength of the photons is material dependent. This section examines three semiconductor based optical emitters considering their construction, uses, limitations and modes of operation.

The first device which we will consider is the *light emitting diode* (LED), as it was the first to be discovered. In 1907, Round noticed *photoluminescence* in Carborundum (SiC) [45] when a current was applied essentially creating a Schottky device [46]. Today's LEDs and indeed most if not all optoelectronic devices rely on the p-n junction [47]. The junction shown here is classified as a *homojunction* structure as both sides are made of the same material with the same bandgap even though their electrical characteristics are different.

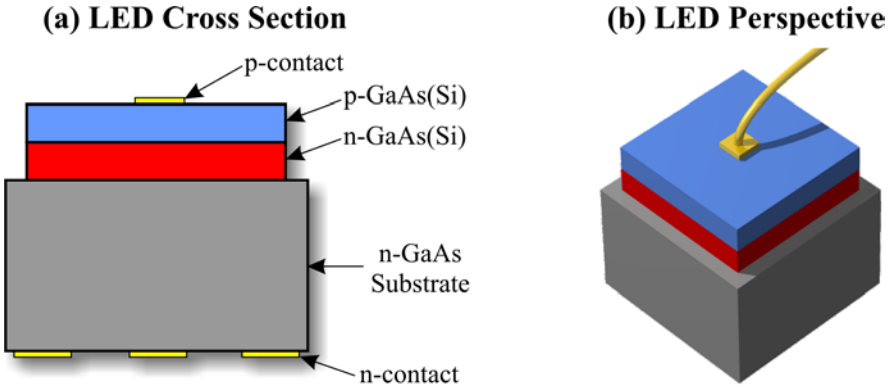

Figure 4: Near Infrared Light Emitting Diode (LED)
Typical NIR-LED structure. Photon emission is spontaneous and not in any particular direction. Doping behaviour of Si is controlled by the temperature of epitaxial growth [48]. Temperatures of above 820ºC form an n-type layer and below form a p-type layer. Patterned n-contact ensures minimum absorption of incident radiation by contact. Layers not to scale.

The first commercial LEDs became available in 1969 and were quickly incorporated as visual indicators on a wide variety of products. Production techniques improved with

consequent increases in not only yield but also device efficiency. Today's LEDs are almost in a position to take over from incandescent filament bulbs and fluorescent tubes as they provide a power efficient "cold" light. Not only that, but their lifetimes are an order of magnitude longer with graceful degradation rather than instantly burning out. The demand for LEDs has continued to grow since their inception and reached the $1billion mark in 2001. However, LEDs for communications purposes have been superseded by laser diodes. This is because the LED is highly divergent and slow by comparison. LEDs rely on *spontaneous emission* of light on application of a current to modulate their optical output. Unfortunately, spontaneous emission is a slow process and is not normally faster than 1ns [49]-[50], limiting throughput on an LED communications channel to around 100MHz.

The *laser diode* (LD) has become a fundamental building block in many of today's communication and storage systems. Again, it is a p-n junction device, but beam divergence is usually a few degrees with potential bandwidths in excess of 20Ghz [51]. This is because laser based devices use *stimulated emission* rather than spontaneous to produce light, thus the name *light amplification by stimulated emission of radiation* (LASER). Nevertheless, spontaneous emission and absorption will still occur, however probability remains heavily on stimulated emission's side.

A laser diode works by exciting the *electron population* in the active region to an upper energy band by application of a current. When the majority of electrons are excited, the population is said to be *inverted*, a state which is inherently unstable. If a photon passes through a medium with an inverted population, it can stimulate an electron in the upper energy band return to the lower one emitting another photon which is identical in every way to the first. Partially mirrored surfaces on the device, due to its refractive index, cause internal reflection returning photons back through the cavity and stimulating the emission of further identical photons in a cascade like manner. This causes light amplification within the resonant cavity, presuming there is sufficient current to sustain the inverted population, and is the basis of laser action.

Laser action was discovered almost simultaneously by four research groups in 1962 [52]-[55]. However, all of these groups created homojunction lasers which required high current densities and could not provide *continuous wave* (CW) operation at room temperature. One year later it was postulated that a *heterojunction* design could considerably improve semiconductor lasers [56], a theory which would later award

Alferov and Kroemer the Nobel prize. Unfortunately, the technology to fabricate such structures would not be available until 1969. It took just one year before room temperature continuous wave operation was demonstrated in 1970 [57]-[58]. Figure 5 shows the construction of a typical *double heterojunction* (DH) laser diode.

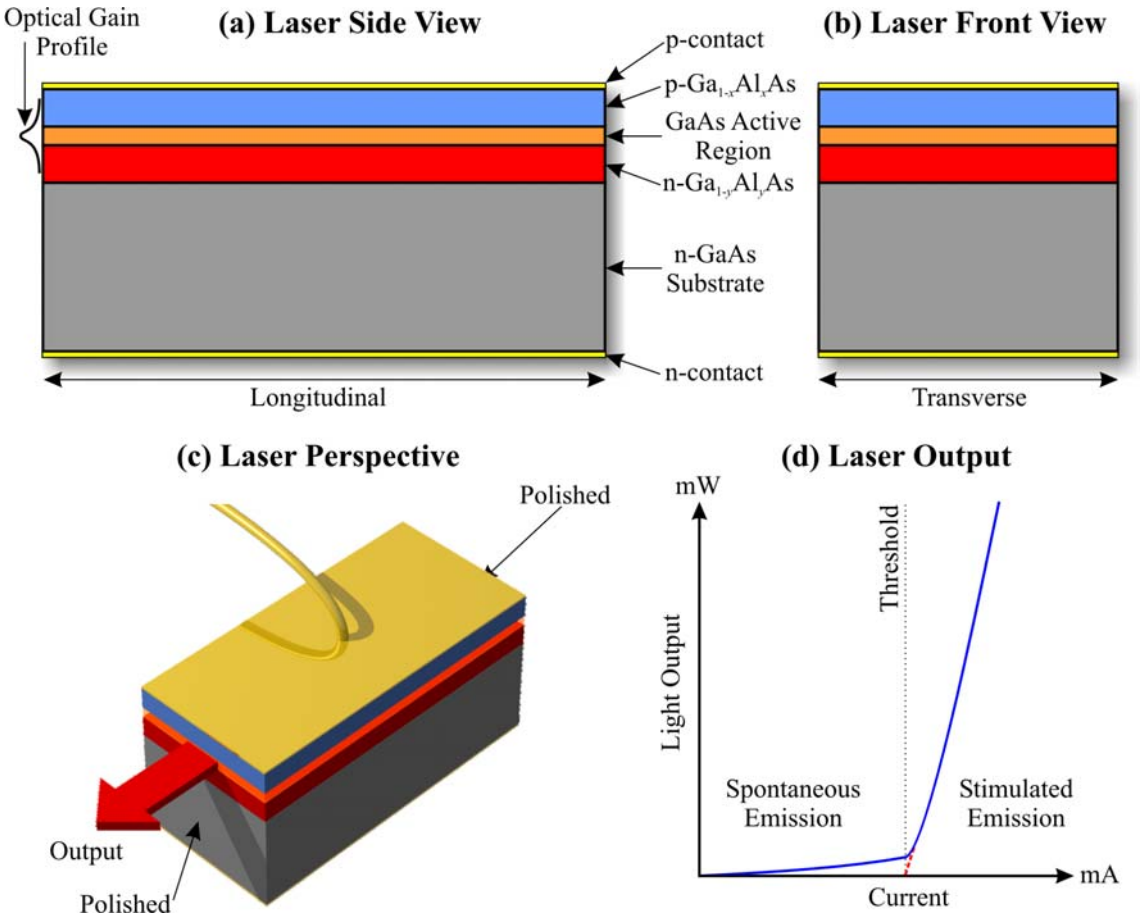

Figure 5: Stripe Laser Diode (LD)
Optical gain is supported in the active region (a). Polishing opposite ends (c) and leaving the remaining sides rough favours laser oscillation along this axis. Below laser threshold (d), spontaneous emission is dominant whereas stimulated emission is dominant above. Layers not to scale.

A heterojunction is a junction between two different materials, in this case GaAlAs and GaAs. Due to the *bandgap* differences at the two junctions, there is greater confinement of electrons and holes to the active region. In addition, the larger refractive index of GaAs also aids confinement of radiation to the active region. Most stripe diode lasers have an elliptical beam output profile, however this is not regarded as a problem since not only is the output stable but it is easier to couple into an optical fibre. For further information on laser diodes see [46], [51], [59]-[61].

The problem with creating laser diodes is *dicing*. A single substrate can contain hundreds of devices each being an edge emitting device. Thus the substrate must be carefully cut using a precision diamond saw and the appropriate edges of each device polished. Considering that these devices are usually measured in microns, this can be an awkward and time consuming process. This problem is addressed by the third and final device that we will consider in this section, the *vertical cavity surface emitting laser* (VCSEL).

The VCSEL, as shown in Figure 6, emits perpendicular to the surface of the chip, simplifying fabrication and lowering production costs. Since there is no longer any need for dicing, smaller structures can be created that consume less power.

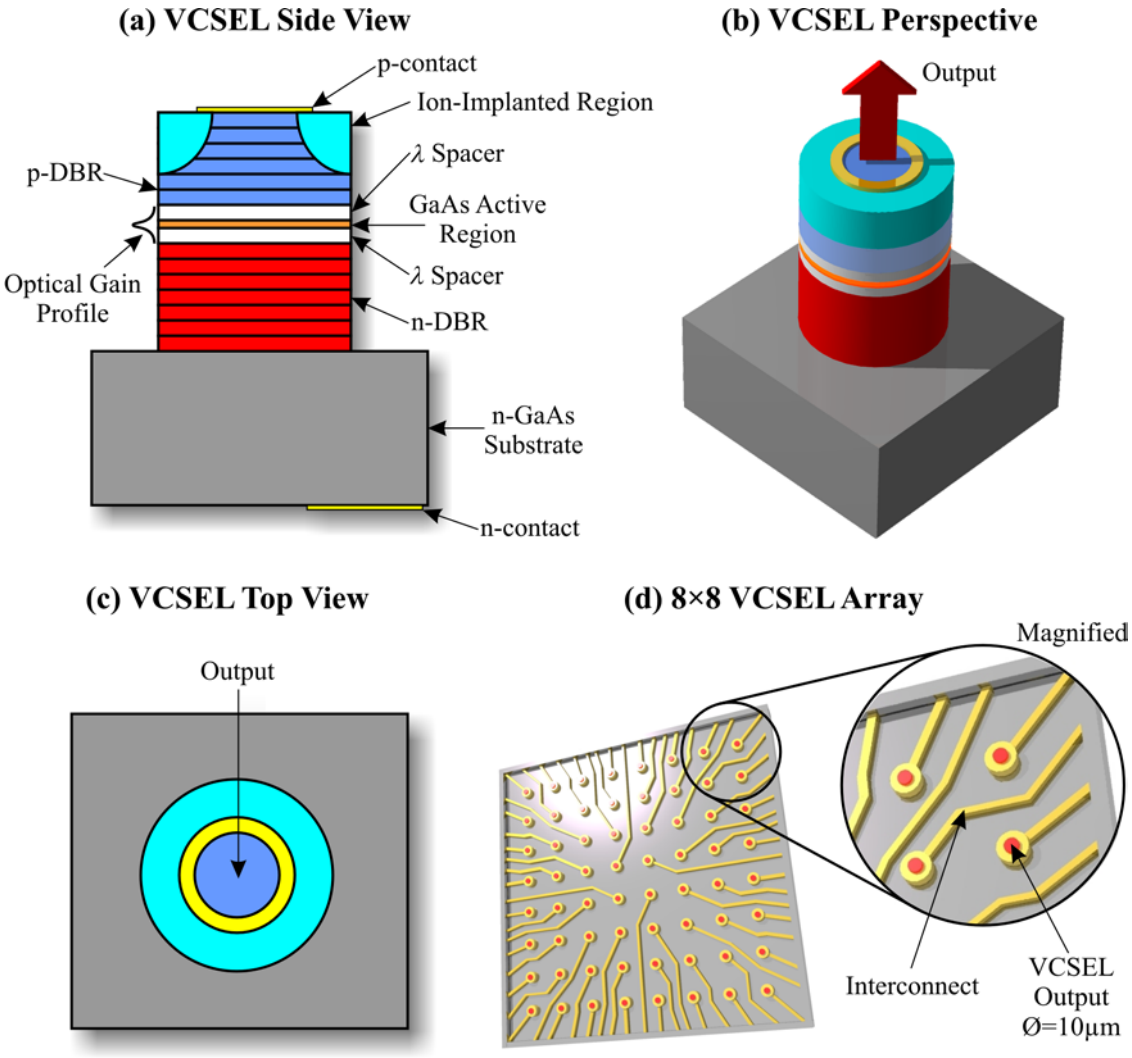

Figure 6: Vertical Cavity Surface Emitting Laser (VCSEL)
Distributed Bragg reflectors (DBR) act as mirrors in a VCSEL. Emission can either be through the top of the VCSEL, as shown, or through the substrate assuming it is transparent at emission wavelength or that a well has been etched. Layers not to scale.

The first VCSEL was built in 1979 [62]. It lased at a temperature of 77K, had a high threshold current and used metal mirrors which had substantial absorption. The construction of efficient mirrors was to remain a problem until 1989 when advances in epitaxial growth enabled the construction of remarkably effective *distributed Bragg reflectors* (DBRs) and subsequently the first room temperature CW VCSEL [63]. DBRs are created by fabricating quarter wavelength $\lambda/4$ thick layers of alternating high and low refractive index substances. Well fabricated layers can have reflectivities approaching 99% [64], however poor fabrication results in poor reflectivity and prevents the VCSEL from lasing. The cavity in a VCSEL is fabricated such that a standing wave is formed between upper and lower DBRs, where the maximum is centred on the active region. Centring is ensured by adding spacers so that the cavity length is an integer multiple of wavelengths. Emission from the top of the VCSEL is forced by leaving the upper interface open to air as shown in Figure 6. To force emission from the bottom surface, the top layer is fully metallised with the contact layer. Both configurations have GaAs as the lower DBR interface.

A now common VCSEL enhancement called *oxide confinement* uses implantation of heavy ions, usually fluorine (F) or oxygen (O), to form a circular aperture in the upper Bragg reflector. This channels the carriers into the active region resulting in increased performance from the VCSEL. Unfortunately, *ion-implantation* is, by its very nature, imprecise. This results in crystal damage, creating fuzzy boundaries around the aperture in the active region with consequent increases in diffraction and divergence. For more information on oxide confined VCSEL arrays refer to [64]-[66].

VCSELs have already taken over from LDs at wavelengths around 850nm, however manufacturing techniques have so far limited their production in the communications wavelength window due to poor DBR reflectivities. This is about to change as improved epitaxial processes are approaching commercial viability which would allow efficient DBR construction for 1.3μm wavelengths [67].

VCSELs are ideal for high density optical interconnects, however array sizes are currently limited to 8×8 or 16×16 devices due to poor yield. Since there is no theoretical limit involved, investment in process technology will allow larger arrays to be fabricated with each VCSEL providing GHz bandwidths.

4.2 Modulators

Modulators differ from emitters in that they transfer information onto an incident optical channel either by changing transmission, reflection or re-routing the beam. Figure 7 shows the construction of a typical *multiple quantum well* (MQW) modulator. Depending on the potential difference across the active region, this modulator will either absorb any input signal or reflect it along the modulated output path. This device is an example of an *absorptive* modulator.

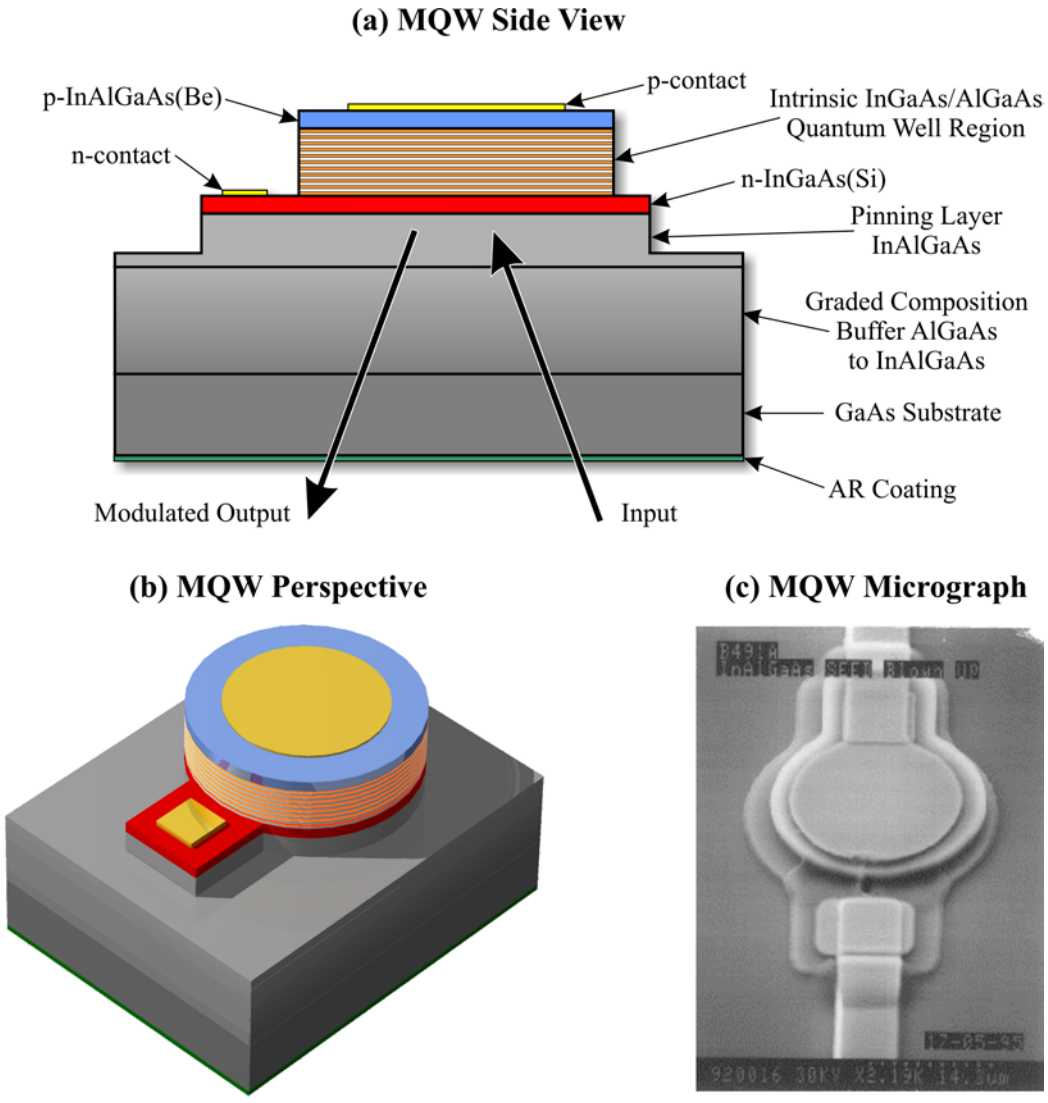

Figure 7: Multiple Quantum Well (MQW) Modulator
The quantum well region modulates information onto an incident optical beam. The substrate is transparent at operational wavelength ~1.04µm. Lattice constants of GaAs/AlGaAs are fairly closely matched, however a graded composition buffer eases lattice tension for the transition to indium (In). Ratios of elements are not included. For more information on the device pictured here see [68]. Layers not to scale.

Absorptive modulators can be created by using one of two effects, either the *Franz-Keldysh effect* (FKE) [69]-[70] or the *quantum confined Stark effect* (QCSE) [71]. The Franz-Keldysh effect states that in the presence of a field in bulk semiconductor, the wavefunctions of electrons and holes tunnel into the bandgap region allowing limited absorption of photons just below the bandgap energy. *Excitonic* interaction [72] adds to, and possibly dominates, this effect allowing photons to be absorbed in a material that would normally appear transparent. Creating multiple quantum wells of alternating high and low bandgap materials allows use of the quantum confined Stark effect. This differs from the FKE in that application of a field shifts rather than broadens the peak excitonic absorption energy. This is because the quantum wells confine electron-hole pairs, preventing ionisation and allowing the application of larger fields resulting in an increased Stark shift. Note that the QCSE can be shown to be a quantised version of the FKE [73]. A typical MQW device has around 50 to 100 layers each of 5 to 10nm thick with a single chip able to sustain thousands of these devices [74]. Unfortunately, coupling light into a semiconductor is problematic at best leading to lower signal powers than with active emitters. Interestingly, absorptive modulators can be used as detectors if field polarity is reversed.

Reflective and refractive devices work by changing the optical properties of a structure such that the path length is altered [51]. Interference or propagation effects are then used to modulate the beam. Although the crystals in such devices are optimal for waveguides as they provide high transmission, they are comparatively large for semiconductor devices and therefore cannot be directly integrated.

The fact that modulators are not subject to carrier or photon build-up problems as seen in active emitters allows them to be used at higher speeds under certain circumstances. Unfortunately, this is also the principal drawback of modulators: they rely on an external optical source. Manufacturers generally wish to construct systems as efficiently as possible using a single process technology and, if feasible, on a single chip. The requirement of separate modulator and emitter, plus associated alignment, may add unnecessary complexity. The author believes that applications requiring extreme performance will push modulator technology with recent technological breakthroughs placing them well for future development. Specifically, Si compatible polymeric modulators have been demonstrated with a measured bandwidth of 110GHz given a meagre drive voltage of 0.8V [75].

4.3 Detectors

Although many types of optical detector exist, this section will concentrate specifically on semiconductor devices by examining their construction, defining characteristics and electrical properties.

Semiconductor detectors can be broadly categorised into three: *photoresistors*, *phototransistors* and *photodiodes* (PD). The resistance of a photoresistor, such as a cadmium sulphide (CdS) cell, changes depending on the amount of incident light. Unfortunately, their response is normally in the millisecond range and composition not compatible with conventional substrate materials. Phototransistors, which can be fabricated in Si, control the flow of a current based on incident light intensity essentially providing amplification. However, at low light levels their amplification is poor and their frequency response limited to around 200kHz due to carrier diffusion times [47].

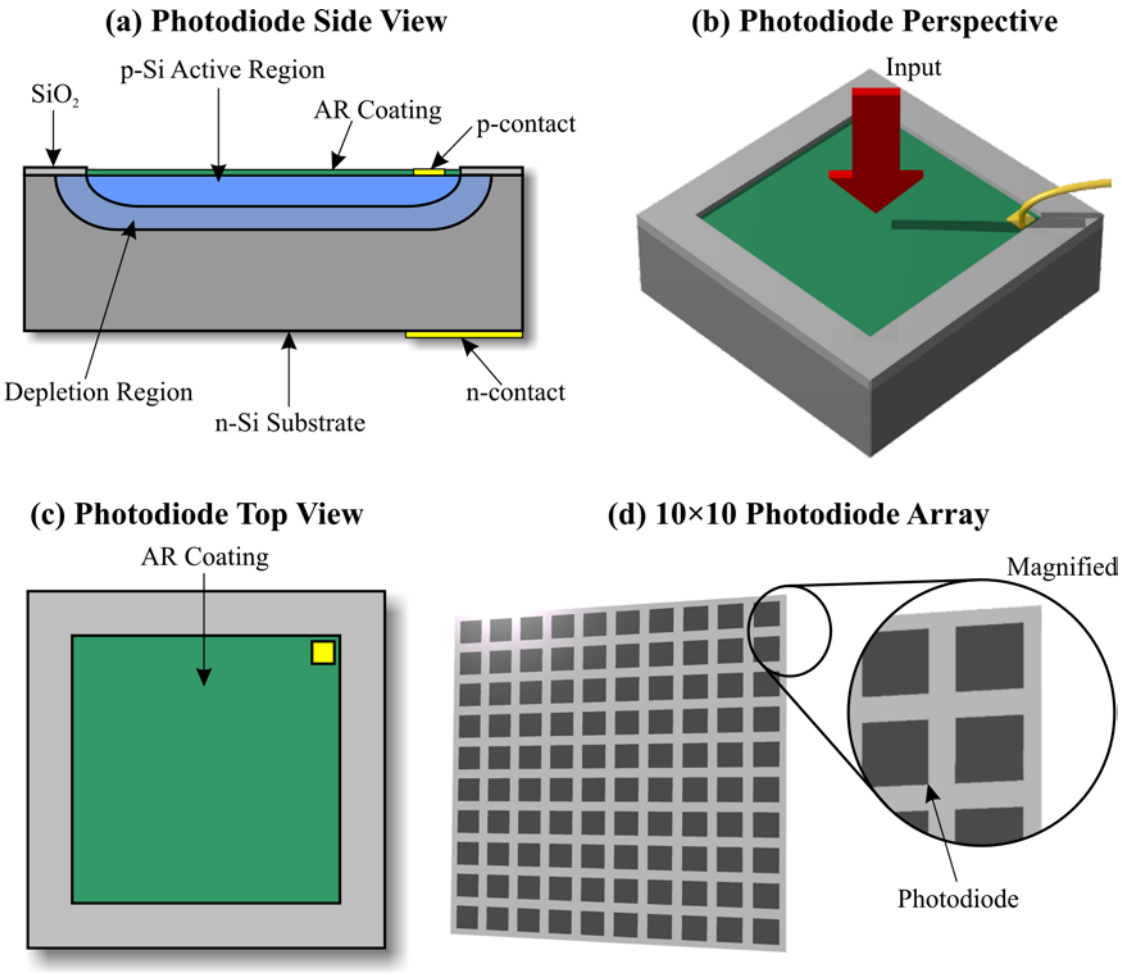

Figure 8: Photodiode Detector
SiO_2 layer masks all but the active region. Array shown in part (d) is not colour coded - each square represents a photodiode. Layers not to scale.

Finally we have the photodiode. It is an Si compatible device that converts any incident light into a current. Figure 8 shows a typical photodiode. Photodiodes are the dominant optical detector technology. They can be easily fabricated in large arrays using existing technology, offer fast response times and can even count single photons at picosecond speeds in incarnations such as the *avalanche photodiode* (APD) [76]. Photodiodes are by definition efficient, however recovering a useable signal requires power input which scales directly with bandwidth.

4.3.1 The Photodiode

When light is incident on a photodiode, a current is produced through external circuitry which is proportional to the light intensity. The photodiode works by exploiting the *photovoltaic effect*. This current response is usually nearly, but not perfectly, linear. Electrons in the semiconductor junction of either p-n or p-i-n type are excited from the valence band into the conduction band by incident photons thus creating a current. This can only happen if the photons carry an energy greater than the bandgap of the detector material. At every wavelength the detector is therefore said to have a *responsivity* $\Re$. It is defined as the ratio of generated photocurrent I_p over incident optical power P_i:

$$\Re = \frac{I_p}{P_i} \qquad \textbf{Equation 1}$$

A real photodiode can be electrically modelled as shown in Figure 9.

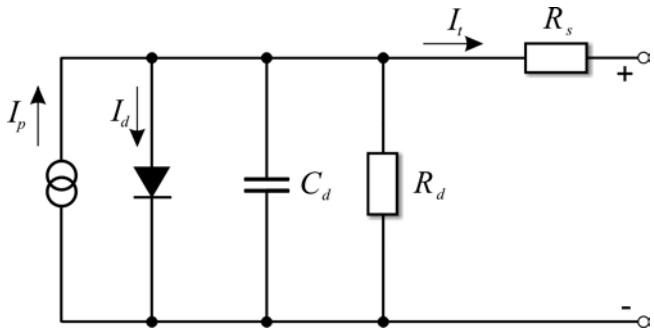

Figure 9: Photodiode Electrical Model
Arrows indicate flow of current.

The photodiode is considered to be an ideal current source, which produces a current I_p as examined above, electrically connected to components used to characterise the photodiode. C_d is the *junction capacitance* since the depletion region acts as the plates of a capacitor. This value limits the maximum detectable frequency and can be lowered by applying a reverse bias voltage. R_d is the *shunt resistance* of the photodiode which is

used to determine the noise current in the photodiode when no light is incident. Ideally, the shunt resistance should be infinite, however typical values lie between 1 MΩ and 1,000 MΩ. Shunt resistance can be measured by applying 10mV to the photodiode, reading the current and calculating the resistance. Finally, R_s is the *series resistance* of the photodiode. It determines the linearity of the photodiode when operated in photovoltaic mode and should ideally be zero. A device's series resistance R_s can be calculated by adding together both junction and contact resistances. Typical devices have resistances of less than 1 kΩ. Under normal circumstances R_s and R_d are considered negligible. C_d is the most important parameter when designing a photodiode receiver.

Photodiodes can be operated in a number of different ways depending on application, as shown in Figure 10.

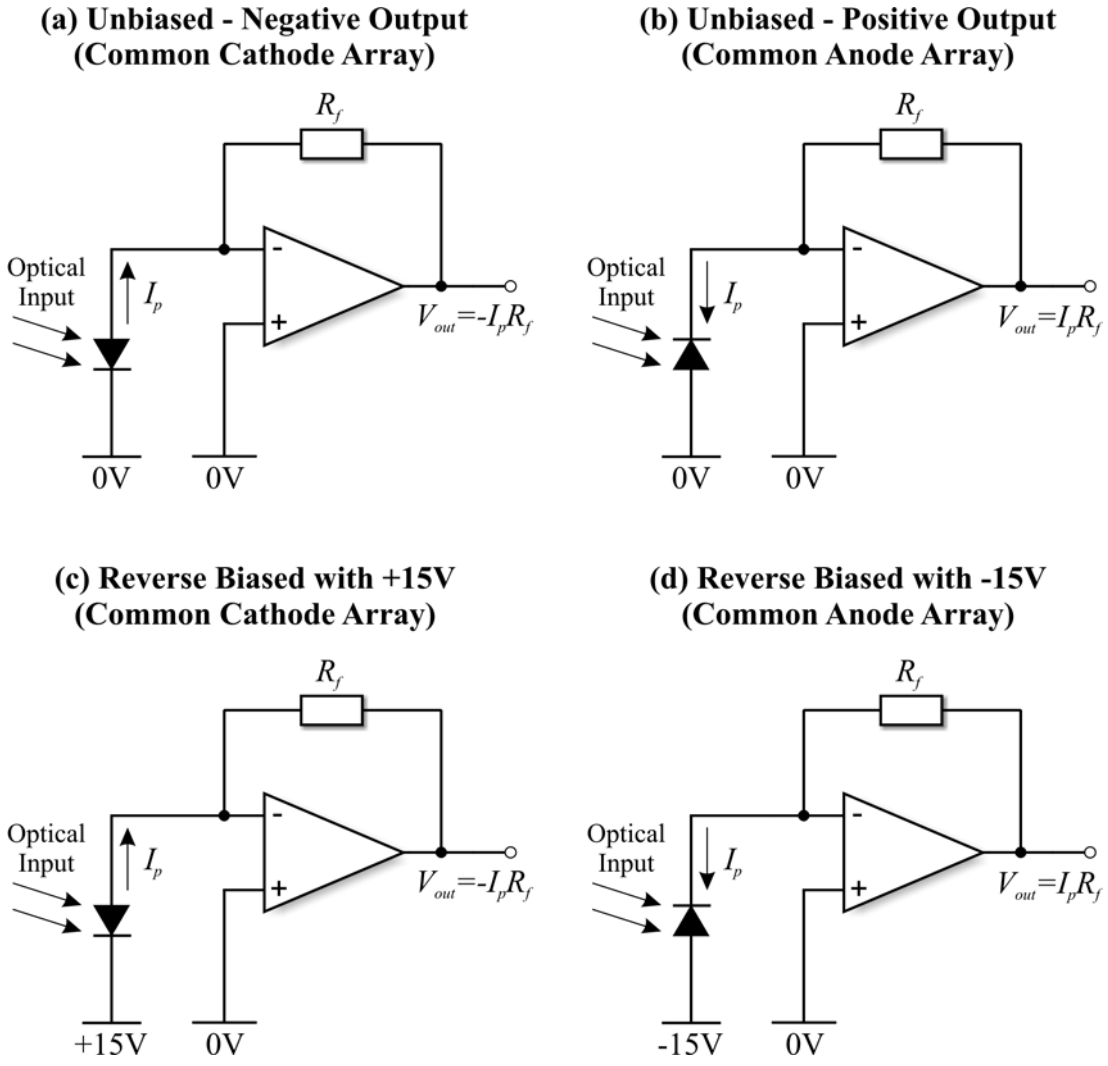

Figure 10: Photodiode Modes of Operation
Unbiased mode has high sensitivity but low bandwidth. Reverse biasing improves bandwidth response but adds to noise.

Firstly, there is *unbiased* or *photovoltaic* mode, as shown in Figure 10(a) and (b), where reversing the photodiode will invert the output signal's polarity. This mode offers low noise and high sensitivity but suffers from reduced bandwidth: such circuits are not normally operated above 350kHz, even when optimised. Secondly, there is *reverse biased* or *photoconductive* mode, as shown in Figure 10(c) and (d). The reverse bias voltage sweeps electrons out of the junction improving both responsivity and bandwidth. Unfortunately, this also results in a *dark current* I_{dk} which adds to noise. Finally there is *forward bias*. Photodiodes are not normally used in this mode as they simply conduct. Returning to the photodiode electrical model in Figure 9,

$$I_t = I_p - I_d \qquad \textbf{Equation 2}$$

Under reverse or unbiased conditions I_d=0, so the total current is I_t=I_p. However, if a forward bias current is applied then there is a decrease in I_t. If I_t exceeds $-100\,\text{mA}$ the photodiode is usually destroyed as this is enough current to burn the contacts off.

Noise is intrinsic in almost all real systems and the photodiode is no exception. There are two main sources of noise in photodiodes. The first is *thermal noise*, referred to as *Johnson* [77] or *Nyquist* [78] noise. At absolute zero all electrons in the junction remain in the valence band but as the temperature increases they become excited and randomly elevate into the conduction band. This results in an r.m.s. current I_j of:

$$I_j = \sqrt{\frac{4kT\Delta f}{R_d}} \qquad \textbf{Equation 3}$$

where k is the Boltzmann constant $1.38\times10^{-23}\,\text{JK}^{-1}$, T is the absolute temperature in Kelvin, R_d is the photodiode shunt resistance and Δf the bandwidth over which the noise is measured (usually 1Hz). Note that thermal noise is also present in detector electronics and not exclusively in the photodiode.

The second source of noise is *shot noise* [79]-[80]. Shot noise (or *white noise*) is a statistical variation in the current generated by both incident optical power I_p and dark current I_{dk}.

$$I_s = \sqrt{2q\Delta f\left(I_p + I_{dk}\right)} \qquad \textbf{Equation 4}$$

This is again an r.m.s. value where q is electronic charge $1.60\times10^{-19}\,\text{C}$ and Δf the bandwidth over which the noise is measured (usually 1Hz). Dark current I_{dk} is the current that flows in a photodetector when no optical radiation is incident and an

operating voltage is applied. It is a combination of surface leakage, generation and recombination of carriers within the depletion region and diffusion to the depletion region of thermally generated minority carriers.

The total r.m.s. noise I_n is a combination of both Johnson and shot noise:

$$I_n = \sqrt{I_j^2 + I_s^2} \qquad \textbf{Equation 5}$$

Noise dominance in this equation depends on the mode of operation. In reverse biased mode, the dark current I_{dk} increases so shot noise I_s becomes dominant over thermal noise I_j. Unbiased mode does not apply a potential difference over the junction, eliminating the dark current I_{dk} and causing Johnson noise I_j to become the dominant term. Therefore, unbiased mode is well suited to ultra-low light level applications as Johnson noise is significantly smaller than the dark current.

The total r.m.s. noise I_n is important as it helps to define the *noise equivalent power* (NEP) at a specific wavelength:

$$\text{NEP} = \frac{I_n}{\Re} \qquad \textbf{Equation 6}$$

This is the amount of incident optical power required ($\text{WHz}^{-\frac{1}{2}}$) to provide a *signal-to-noise ratio* (SNR) of 1. Average values range from $1\times10^{-11}\,\text{WHz}^{-\frac{1}{2}}$ for large area photodiodes to $1\times10^{-15}\,\text{WHz}^{-\frac{1}{2}}$ for small area ones. NEP can be used to compare two similar detectors.

Photodiodes are temperature sensitive but their response to temperature change is dependent on the mode of operation. In unbiased mode, an increase in temperature will result in a decrease of shunt resistance R_d. R_d is halved for every 6K increase in temperature [81]. In reverse biased mode, the dark current I_{dk} is doubled for every 10K increase in temperature [82]. Although the exact change is device dependent, the trend remains the same with unbiased mode being more sensitive to temperature increase. It should be noted that there is also a responsivity change with temperature but this is material dependent. For example, lowering an Si photodiode's temperature improves responsivity at shorter wavelengths (blue to ultra-violet) and increasing it improves responsivity at longer wavelengths (near infra-red). This change does not normally exceed a few percent under normal operating conditions.

Connecting a photodiode is essentially a speed to noise trade-off. The faster a photodiode goes, the more noise it generates. Noise in turn limits the minimum

detectable optical power, and thereby the maximum distance between optical transmitter and receiver given that signal diminishes with distance.

4.4 Optical Interconnect Elements

This section examines optical components that can be used to create static or dynamic two dimensional interconnect patterns in free space.

The first component considered is the *diffractive optic element* (DOE), the operation of which was first demonstrated in 1967 [83]. In the same way that a diffraction grating divides an incident beam in one dimension, the DOE shapes a beam in two dimensions to create a desired intensity profile in the far field of a Fourier lens. These devices are planar elements consisting of areas which retard incident light. They can perform complex optical functions which may have previously required several optical components, that is if the function was at all feasible in any other manner.

DOEs are compact, can be constructed using robust materials such as silica and are simple to manufacture using existing *very large scale integration* (VLSI) fabrication techniques. They can be mathematically described by *kinoforms* [84]-[85] and are fabricated as either binary or multilevel. Binary elements use a single stage fabrication process resulting in good uniformity across the array. Multilevel structures, as shown in Figure 11, require as many fabrication steps as there are layers, leading to an increase in non-uniformity due to alignment mismatch of 1% to 2% per layer. However, multilevel structures result in substantially improved transmissions. Note that light which is not transmitted is normally scattered outside the diffraction window and not absorbed by the DOE.

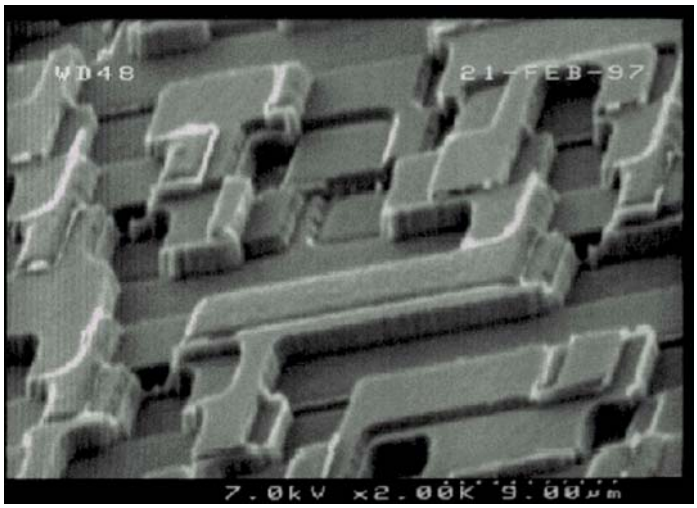

Figure 11: Multilevel Diffractive Optic Element
Surface micrograph of a multilevel DOE.

A repeating pattern exists in the DOE which is referred to as its period and defines the maximum angular divergence. As the grating period gets smaller, the maximum diffraction angle increases. DOEs are matched to a specific wavelength to minimise or eliminate the zero diffraction order. Complex and large scale interconnection patterns can be created using a DOE. Heriot-Watt University has fabricated devices in-house capable of fanning-out to 128×128 elements. Obviously such large scale interconnects are limited by input beam intensity since every fanned-out channel must have a large enough fraction of the input beam power to make it detectable.

The second component examined is the *spatial light modulator* (SLM) [86] which works in the same way as the DOE except that it is programmable. These devices are based on *liquid crystal displays* (LCD) in that they contain a large number of individually addressable voltage controlled pixels. Figure 12(a) illustrates the operation of a *twisted nematic* (TN) device with no voltage applied. The liquid crystal cell rotates the polarisation state of any incident light thus allowing it to pass through the analyser. In Figure 12(b), a voltage is applied across the liquid crystal which prevents rotation of polarisation state. Since vertically polarised light cannot pass through a horizontal analyser there can be no transmission of light.

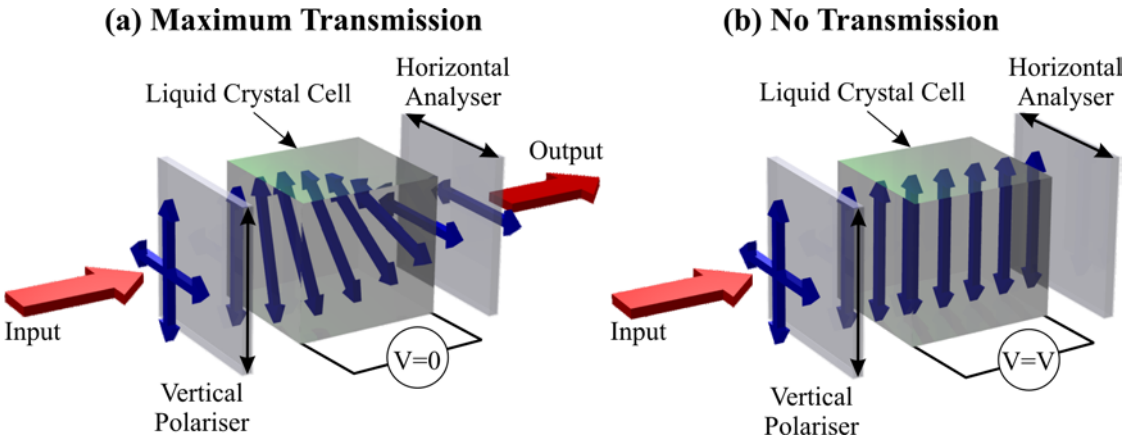

Figure 12: Spatial Light Modulator
Vertical polariser converts the input beam to a single vertical polarisation state. Horizontal analyser only allows transmission of horizontally polarised light. This device is configured to modulate amplitude. Replacing the horizontal analyser with a vertical one inverts the effect of any applied voltage.

When used as shown to modulate amplitude, the SLM can be used to control the routing of a specific transmission channel. Removal of both polariser and analyser generates phase lag rather than amplitude modulation allowing an SLM to be used as a programmable DOE.

The use of SLMs tends to be limited by their relative expense, complex control logic and slow refresh rates. If used to control phase, configurations need to be stored in memory as computation of a new configuration requires a large amount of processing power and is therefore not feasible in real time. The devices do not particularly suffer from fabrication limitations, indeed megapixel devices already exist [87], but rather from liquid crystal response times and serial reconfiguration. As SLMs are addressed in a serial manner, larger arrays require longer reconfiguration times consequently reducing the refresh rate of the entire array. These disadvantages mean that SLMs rarely have frame refresh rates of greater than a few kilohertz.

4.5 Conclusion

This chapter has examined the current technologies that enable optical interconnection. Systems that use this technology are referred to as *smart pixel arrays* (SPAs) and are defined as an optoelectronic device that may have memory, intra-pixel processing, inter-pixel communication and an optical input or output element.

The process technologies that enable smart pixels are beginning to mature in their own right. However, direct integration and exploitation of their full bandwidth potential is still some way off. This has previously been due to fabrication difficulties, however alternative methods of integration are beginning to emerge that can reduce integration complexity and improve yield. Viability and reliability of such systems has recently been clearly demonstrated by the company Terraconnect [88]. In 2001, it had just reached prototype stage with a single general purpose optical interconnect module that integrates 320 oxide confined VCSELs in a 16×20 array, 320 VCSEL driver amplifiers, 320 GaAs p-i-n photodiodes, 320 transimpedance amplifiers and all the coding and switching logic to ensure error free data transmission. At the time of writing, this system had been running continuously for just over eight weeks without so much as a single bit error. This is further proof that not only are these systems feasible but they are becoming a reality.

5 The Optical Highway

This section introduces the *optical highway* (OH) as an interconnection solution. It then proceeds to examine variations on the architecture including their associated scalability, bandwidth and latency attributes. The equations derived here allow detailed modelling of any proposed implementation and thereby assessment of the impact such an optical highway would have in a particular situation.

5.1 Optical Highways

The concept of optical highways [89]-[90] envisages a high bandwidth low latency general purpose multiprocessor interconnection architecture. Figure 13 schematically shows such a highway which is used to connect nodes in an arbitrary topology where each node has access to more than 1,000 channels.

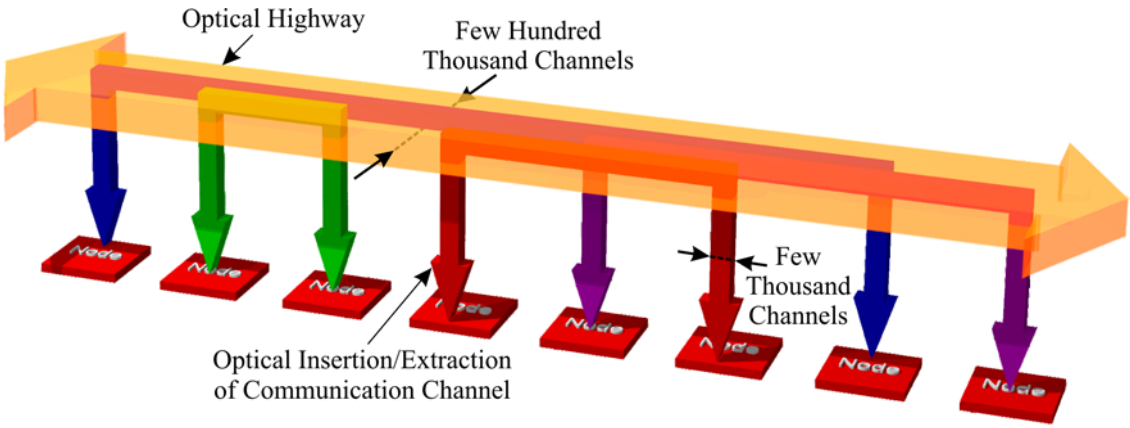

Figure 13: Optical Highway
Free space optical highways interconnect multiple nodes through a series of relays that are used to add or drop thousands of channels at a time.

A node is considered to consist of a processing system and a custom optoelectronic interface chip such as an SPA. The interconnect is point to point and hard wired, with several thousand channels being passed to and from each node via an optoelectronic interface into a free space optical relay system which can hold several hundred thousand channels. The number of nodes that can be interconnected in a specific network topology is primarily limited by aberrations in bulk optic lenses.

The optical highway abstract model defines six parameters that will be used as an interface. Some of these parameters are specified and the others are calculated.

However, these parameters do not take into account network size, topology or particular communicating nodes - they simply provide a generalised model. This allows the model to generalise an architecture rather than specific nodes in an architecture. Figure 14 indicates the logical part of the model to which each parameter is associated:

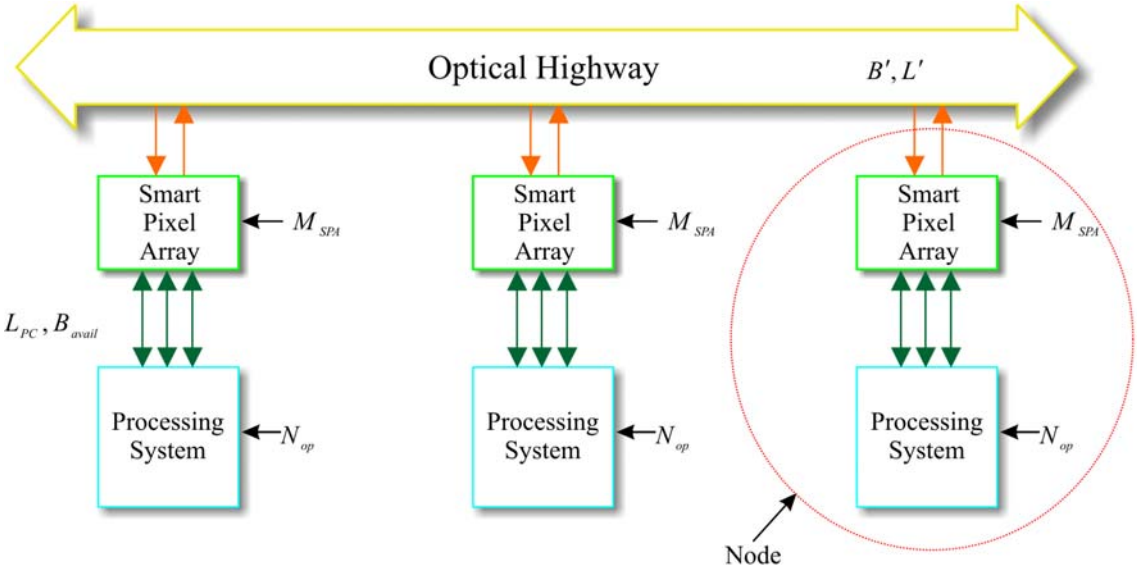

Figure 14: Optical Highway Abstract Model
Six parameters are used to define the optical highway. This diagram indicates to which logical part of the model each parameter is associated.

The parameters are defined as follows:

B' defines the optical node-to-node bandwidth measured in bytes per second (Bs^{-1}). It is the maximum available capacity on an optical channel in a single direction. An identical channel of the same bandwidth is required in the other direction to provide duplex connection.

B_{avail} is the best practically available, and sustainable, bandwidth on the I/O bus of the processing system to which the SPA is connected. It is measured in bytes per second (Bs^{-1}) and is irrespective of the direction in which data travels. The sustained data rate of B' should never exceed B_{avail}. Note that processing system peripherals must also compete for available I/O bus bandwidth.

L' is the optical system latency from either a source node's smart pixel array memory or the end of the processing system I/O bus to one of the same on the destination node. This value is obviously both architecture and route dependent. It is measured in seconds (s).

L_{PC} is the latency from main memory to end of I/O bus. It is highly dependent on processing system and is measured in seconds (s). Note that most commodity architectures have memory latency that is not related to memory bandwidth. This has been examined earlier.

M_{SPA} defines the smart pixel array memory buffer size in bytes (B). This size must be balanced to minimise the risk of data loss due to overrun and maximise cost efficiency.

N_{op} is an estimation of the number of operations per second (ops^{-1}) that a processing system has left given that a specified amount of information is transferred to or from main memory. This value is heavily influenced by the amount of processor cache and memory bandwidth load.

5.2 Optical Highway Construction

There are many possible designs of optical hardware in an optical highway. Two examples are give here which use polarising optics to route data channels as shown in Figure 15. A *polarising beam splitter* (PBS) deflects channels of a specific polarisation to a node with each channel's polarisation state determined by patterned *half wave plates* (HWP).

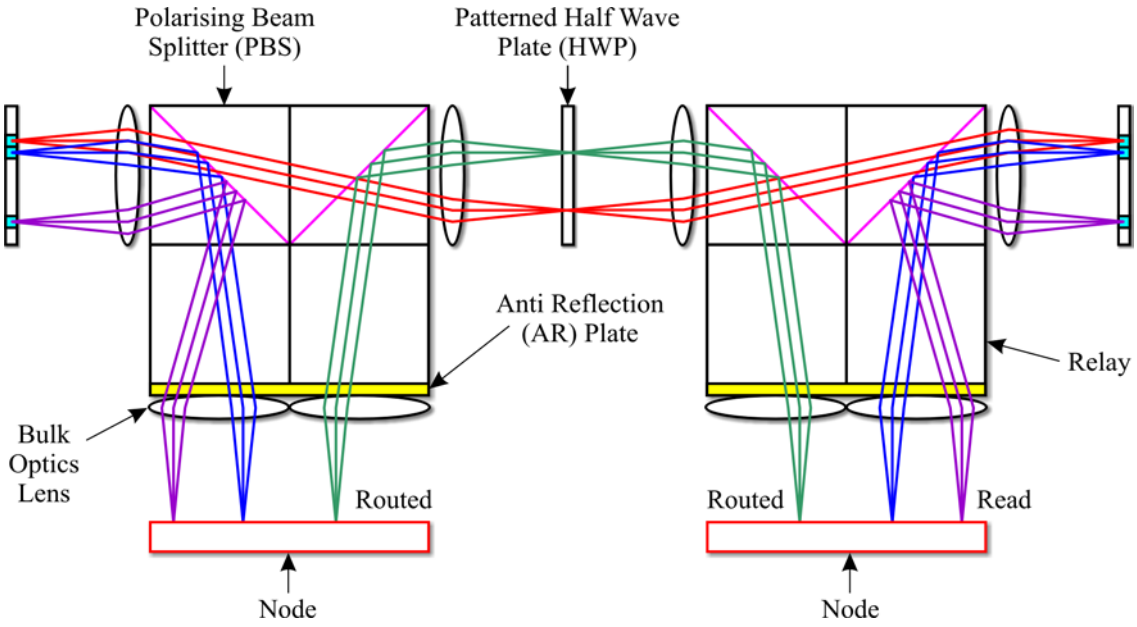

Figure 15: Dual PBS Optical Highway Construction
Polarising optics define a fixed network topology.

The second, and similar, system can be constructed using a single PBS as shown in Figure 16.

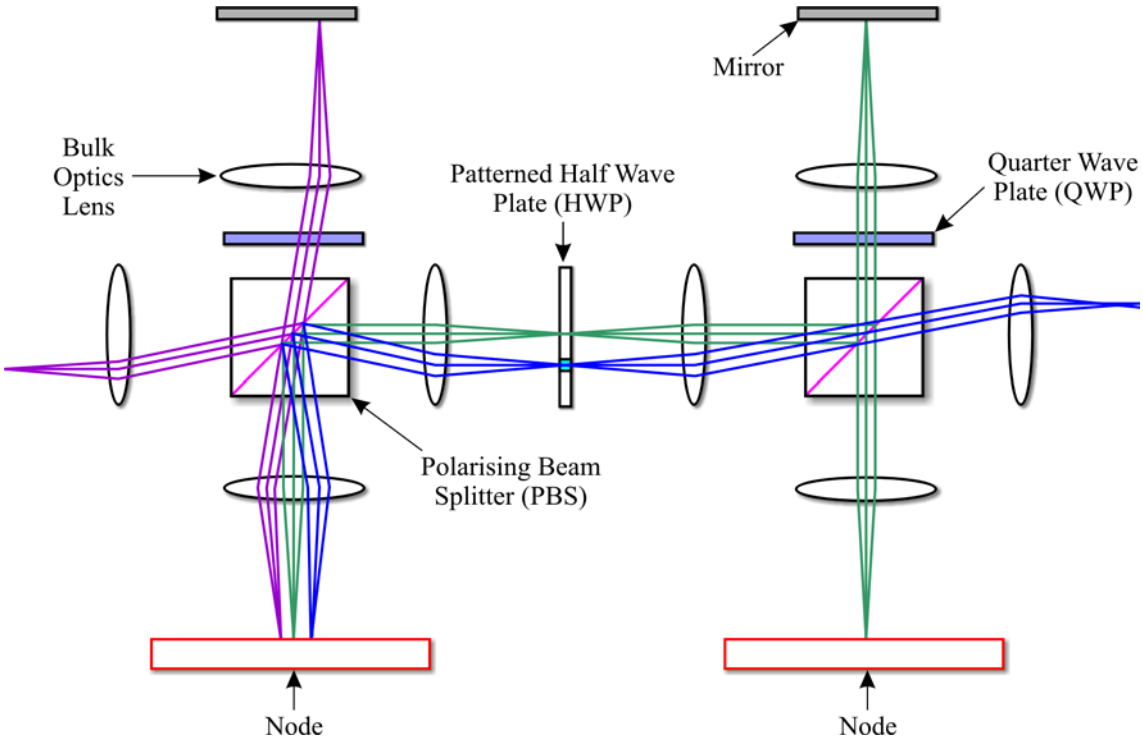

Figure 16: Single PBS Optical Highway Construction

Polarising optics define a fixed network topology. If modulators are used instead of VCSELs the mirror must be patterned and a second QWP added between the node and the PBS to allow a read beam into the system.

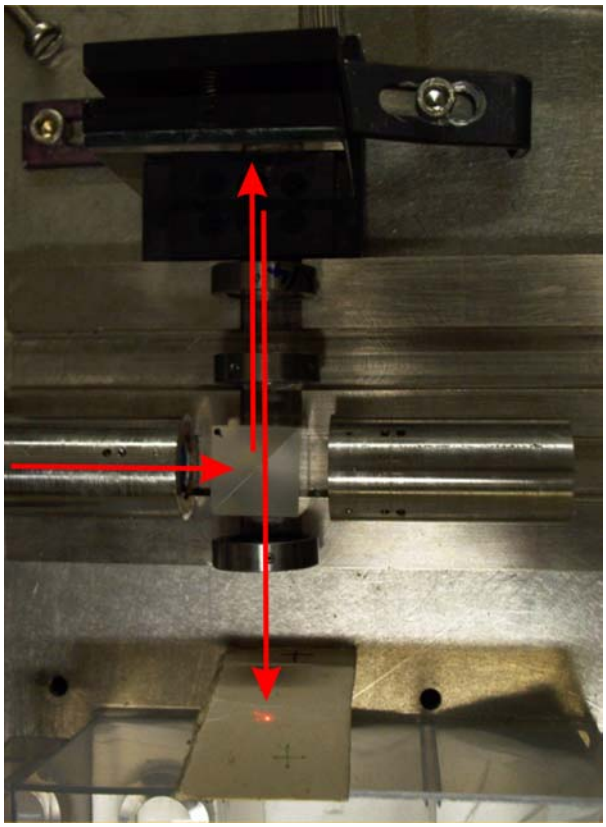

Figure 17: Single PBS Experimental Construction

The red arrows indicated the path taken by a data channel.

The single PBS version was constructed at Heriot-Watt University and can be seen in Figure 17. This system was designed as a proof of principle and used an LCD to route channels to and from their destination node. It is superior as it halves the number of beam splitters required, reducing the power loss and increasing the maximum size of the network. This is the topology that will be considered in the rest of this section.

Optical highways can be made reconfigurable using an SLM in place of the patterned HWP. In order to maintain efficiency, run-time reconfiguration cannot be performed since it requires a reconfiguration controller with associated hardware reconfiguration delays. Compile time reconfiguration is desirable but not necessarily a requirement. Regardless of when reconfiguration occurs, most algorithms will need to be adjusted for computation on a multiprocessor system. It would therefore be simpler to fit the algorithm to a fixed topology optical highway. Thus complex reconfiguration control hardware can be eliminated, including potential run time issues such as determination of the entire optical highway's current state.

Given components similar to those already constructed and in use by the SCIOS project in this research group [91], we can extrapolate the potential bandwidth of such a system. Considering that 2,500 MQW based optical channels off-chip are feasible at data rates of 250MHz each, a two node system has a bisection bandwidth in excess of 1Tbs^{-1}. Depending on topology [89], the potential bisection bandwidth is therefore far in excess of any existing electronic architecture.

5.3 Optical Highway Configuration

An optical highway is considered to consist of a total of *p processing system* (PS) and *smart pixel array* (SPA) pairs. Given that the combination of PS and SPA is referred to as a node, any optical highway must have *p* nodes. Since the topology is fixed and that links are dedicated, we can calculate the distance between a source node p_s and a destination node p_d. Both of these must have a value which lies between 1 and *p* inclusive. There is no distance between nodes if $p_s = p_d$. The distance value is measured by an integer number of hops *h*.

5.3.1 Linear Optical Highway

The *linear optical highway* (LOH) is an optical highway in its most basic form and can be seen in Figure 18. Presuming an even point to point load, the central section of this highway has the highest bisection bandwidth and the outlying sections the lowest.

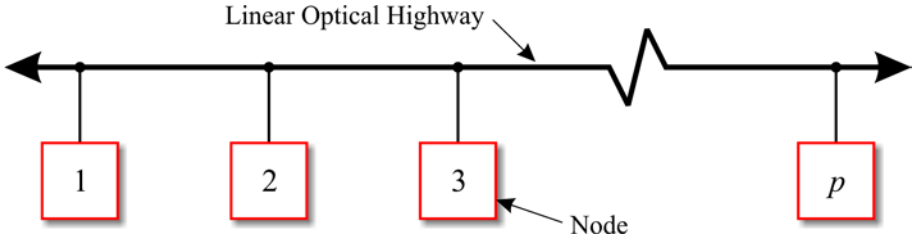

Figure 18: Linear Optical Highway
A linear optical highway has a single and fully interconnected optical system.

The maximum distance across the highway can be calculated using:

$$h_{\max} = p - 1 \qquad \textbf{Equation 7}$$

The number of hops point to point, given that source and destination nodes are know, can be calculated using:

$$h = |p_d - p_s| \qquad \textbf{Equation 8}$$

This number of hops allows the latency of a particular connection to be calculated.

5.3.2 Circular Optical Highway

In a *circular optical highway* (COH), as shown in Figure 19, nodes are arranged in a linear manner with the last node looped back to connect with the first node thus forming a ring topology. With loopback, network load is evenly distributed throughout the highway rather than concentrated at any particular point. This loopback also reduces the average distance from node to node by a factor of two.

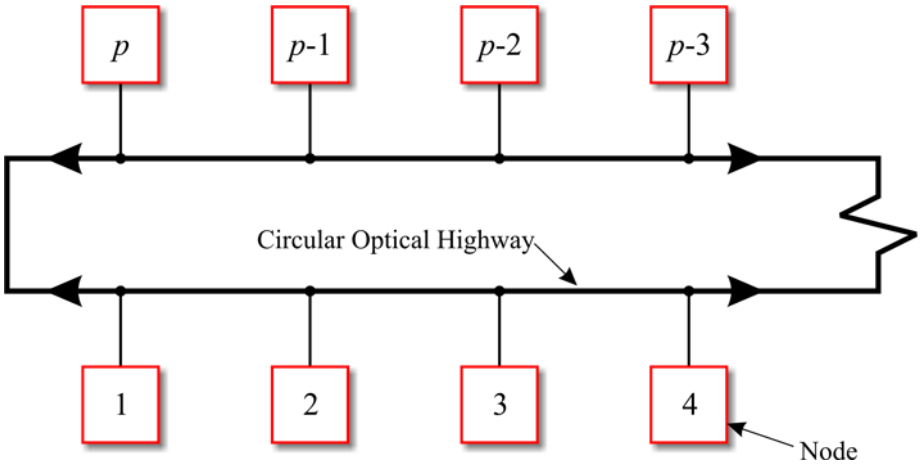

Figure 19: Circular Optical Highway
A circular optical highway has a single and fully interconnected optical system looped back on itself to reduce overall system latency and distribute bandwidth.

The maximum number of hops $h_{\max}$ required to traverse a such a network is dependent on whether p is even or odd. If p is odd there is one distinct best path giving a maximum of:

$$h_{\max} = \frac{p-1}{2} \qquad \textbf{Equation 9}$$

If p is even there are two possible paths where the one used is determined by architecture. The maximum path length can be determined using:

$$h_{\max} = \frac{p}{2} \qquad \textbf{Equation 10}$$

Assuming that the nodes are numbered linearly, a set of rules can be given to calculate the number of hops h between source node p_s and destination node p_d. The rule to be used depends on which way the signal should pass through the network. If the following rule is true:

$$|p_d - p_s| > h_{\max} \qquad \textbf{Equation 11}$$

Then calculate the number of hops using:

$$h = p - |p_d - p_s| \qquad \textbf{Equation 12}$$

Otherwise, calculate the number of hops using:

$$h = |p_d - p_s| \qquad \textbf{Equation 13}$$

This number of hops allows the latency of a particular connection to be calculated.

5.3.3 HyperCube Plus Optical Highway

A *hypercube plus optical highway* (HC+OH) architecture uses multiple optical highways in a variation on both mesh and hypercube topologies. This further reduces overall system latency and distributes bandwidth. However, a routing delay through a node when travelling between vertical and horizontal highways will be encountered.

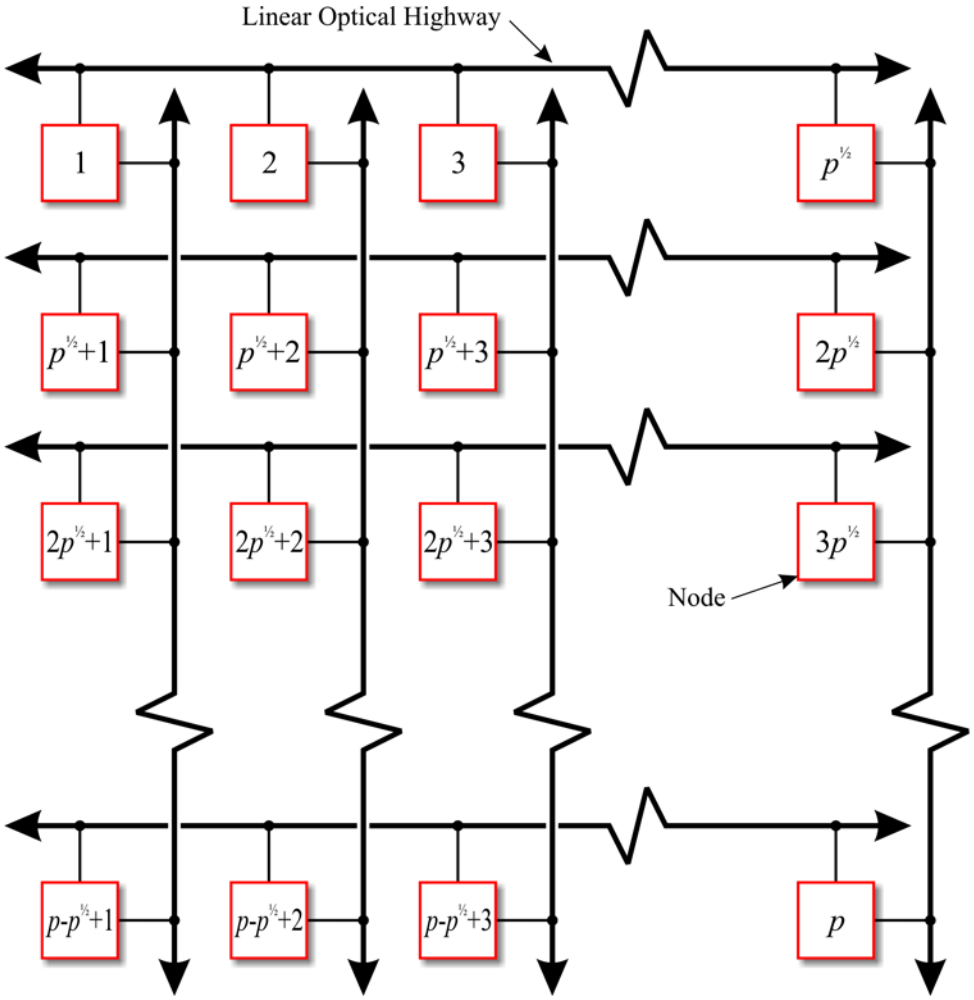

Figure 20: Hypercube Plus
Multiple optical highways can be used in a variation on both mesh and hypercube topologies to reduces overall system latency and distributes bandwidth.

Note that the this highway must be square, with equal lengths of both horizontal and vertical highways, for the equations in this section to work.

The maximum number of hops $h_{\max}$ required to traverse a such a network is defined as:

$$h_{\max} = 2\left(\sqrt{p} - 1\right) \qquad \textbf{Equation 14}$$

To determine the number of hops between source and destination nodes, first use the equation below to determine the *y* coordinate of the source node. The division in this equation must be rounded up to the nearest integer otherwise the formula will not work.

$$p_{sy} = \frac{p_s}{\sqrt{p}} \qquad \textbf{Equation 15}$$

Next, the *x* coordinate of the source node must be determined using:

$$p_{sx} = p_s - \sqrt{p}\left(p_{sy} - 1\right) \qquad \textbf{Equation 16}$$

The same process is followed through for the destination node. Again, the division in this formula must be *rounded up* for it to work:

$$p_{dy} = \frac{p_d}{\sqrt{p}} \qquad \textbf{Equation 17}$$

This allows the *x* coordinate of the destination node to be determined:

$$p_{dx} = p_d - \sqrt{p}\left(p_{dy} - 1\right) \qquad \textbf{Equation 18}$$

The four coordinates calculated above allow the distance to be determined as:

$$h = \left|p_{dx} - p_{sx}\right| + \left|p_{dy} - p_{sy}\right| \qquad \textbf{Equation 19}$$

Unless $p_{sx} = p_{dx}$ or $p_{sy} = p_{dy}$, an additional routing latency will be incurred as information traverses from one optical highways to another. This latency is described as the routing latency L_r and will be in addition to optical-to-electronic L_{oe}, electronic-to-optical L_{eo} and encode-decode L_{ed} latencies. These exact values will be discussed later.

5.3.4 Optical Highway Latency

Given that we know the number of hops *h* and optical highway path dimensions, we can calculate the optical path length *q* in meters:

$$q = hq' + 2q'' + 2q''' \qquad \textbf{Equation 20}$$

This allows us to calculate the node to node latency however the value is dependent on the refractive index of the guiding medium:

$$L_{OH} = \frac{nq}{c} \qquad \textbf{Equation 21}$$

where the constant $c = 3.00 \times 10^8 \text{ms}^{-1}$. In the typical optical highway setup more than 95% of the system is air, thus $n \approx 1$. If the guiding medium were glass then $n \approx 1.56$.

5.4 Optical Highway Channels

Transferring information from point to point on the optical highway is done using a series of *physical optical channels*. A physical channel consists of an emitter-detector pair, each on a different node, optically interconnecting the nodes. These components have intrinsic bandwidth limitations, the faster of the devices having to run at the bandwidth of the slower. Multiple physical channels need to be used to connect nodes as shown in Figure 21 as data transmission on a physical channel is unidirectional:

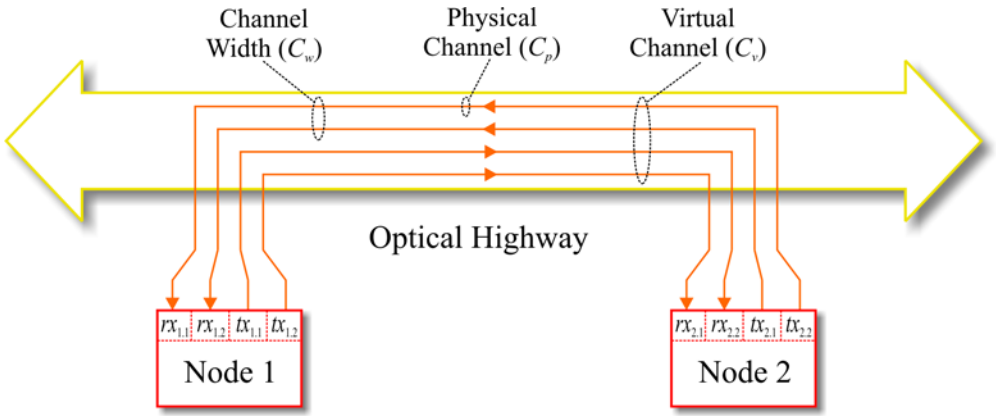

Figure 21: Channel Configuration
Multiple emitter-detector pairs on different nodes are combined to create a virtual channel.

Optical node to node bandwidth in one direction B' can be calculated using:

$$B' = B_{optical} \xi_{ed} C_w \qquad \textbf{Equation 22}$$

This equation considers the raw bandwidth of a physical channel to be $B_{optical}$ bytes per second (Bs^{-1}) in one direction. For this analysis 8B/10B [92] encoding is assumed, i.e. 8 bytes of data are encoded in 10 bytes of information. This encoding is modelled as an efficiency ξ_{ed} for transmission in the optical domain. Typical values lie around 0.8, which is also the value used for 8B/10B clock encoding. To boost the available node to node bandwidth simply increase C_w. Note that any increase in C_w does not reduce latency – it simply increases the available bandwidth once data transfer has started.

The number of physical interconnection channels C_p required to create a single bi-directional node to node virtual channel can be calculated using:

$$C_p = 2C_w$$ **Equation 23**

The 2 in this equation comes from the requirement for bi-directional communication.

5.5 Optical System Constraints

The characteristics of any optical component used to construct an optical highway places constraints on maximum achievable interconnectivity. These characteristics are *optical power*, *aberration* and *device densities* on the optoelectronic chip. For the purposes of modelling, these limits are assumed to be completely independent of each other and that the most significant of the three is the final limit. Realistically, their independence does not hold under extreme circumstances where tradeoffs need to be made. For example, the power calculations are dependent on both NEP and VCSEL optical output power, but these are both dependent on device density even though this is not taken into consideration. Therefore, as a general rule, do not assume that any theoretically calculated component values are readily available or even technologically feasible.

5.5.1 Power Limit

NEP is the amount of incident optical power required ($\mathrm{WHz}^{-\frac{1}{2}}$) to provide a signal-to-noise ratio of 1. Average values range from $1\times10^{-11}\,\mathrm{WHz}^{-\frac{1}{2}}$ for large area photodiodes to $1\times10^{-15}\,\mathrm{WHz}^{-\frac{1}{2}}$ for small area ones. Photodiode arrays are assumed to be the far end of diode sizes ($1\times10^{-15}\,\mathrm{WHz}^{-\frac{1}{2}}$). Minimum power at the detector is therefore:

$$P_{det} = \mathrm{NEP}\sqrt{8B_{optical}}$$ **Equation 24**

where $B_{optical}$ is the point to point bandwidth, in bits per second, and can be calculated from B', the optoelectronic bandwidth as defined in the abstract model, using:

$$B_{optical} = \frac{B'}{\xi_{ed} C_W}$$ **Equation 25**

where ξ_{ed} is the coding efficiency and C_W the channel width.

Efficiency per optical stage ξ, where an optical stage is considered to be from one PBS to the next in the highway, is therefore:

$$\xi = \xi_{lens}^2 \xi_{PBS} \xi_{HWP}$$ **Equation 26**

where ξ_{lens}, ξ_{PBS} and ξ_{HWP} are the efficiencies of the individual components. The total efficiency is $\xi^h \xi_{mirror} \xi_{QWP}$ where h is the total number of relay stages and ξ_{mirror} and ξ_{QWP} are the efficiencies of the mirror and *quarter wave plate* (QWP) used to route the signal in or out of the system.

From this it is possible to find out the maximum number of processors that the system can support before the power loss becomes too great:

$$P_{VCSEL} \xi^{h_{\max}} \xi_{mirror} \xi_{QWP} = \mathrm{NEP} \sqrt{8 B_{optical}} \qquad \textbf{Equation 27}$$

where $h_{\max}$ is the maximum number of hops, or relay stages, between nodes and the input optical power of a single VCSEL is denoted by P_{VCSEL}.

Substituting into Equation 25 gives:

$$B' = \frac{\xi_{ed} C_W P_{VCSEL}^2 \xi^{2h_{\max}} \xi_{mirror} \xi_{QWP}^2}{8\mathrm{NEP}^2} \qquad \textbf{Equation 28}$$

To find the maximum number of processors $p_{p\max}$, Equation 28 can be rearranged assuming that there is a distinct path for a circular topology, or rather that p is odd, to give:

$$p_{p\max} = \frac{2}{\ln(\xi)} \ln\left(\frac{144\mathrm{NEP} \sqrt{\dfrac{8B'}{\xi_{ed} C_W \xi_{mirror} \xi_{QWP}^2}}}{P_{VCSEL}} \right) + 1 \qquad \textbf{Equation 29}$$

Note that if $\xi = 1$ then the power limited number of processors is infinite, or better said there is no limit. The value of 144 is necessary here to ensure that the *bit error rate* (BER) is 10^{-9} [93].

P_{VCSEL} can be taken from a real device specification or modelled using:

$$P_{VCSEL} = \frac{\dfrac{\xi_l}{V_{th}}}{\left(1 - \dfrac{\xi_l}{V_{th}}\right)} \left(P_{Velec} - I_{th} V_{th} \right) \qquad \textbf{Equation 30}$$

where ξ_l is the laser slope efficiency, I_{th} is the laser threshold current, V_{th} the laser threshold voltage and P_{Velec} the VCSEL electrical power as shown in Figure 5(d).

5.5.2 Aberration Limit

The diameter of the spot after $h_{\max}$ $4f$ image relays can be calculated using:

$$s_{\max} = \sqrt{s_0^2 + h_{\max}\left(s_1^2 + s_2^2 + \ldots\right) + s_{VCSEL}^2} \qquad \textbf{Equation 31}$$

Where s_1, s_2 etc are the contributions to the spot size due to the aberrations in the system, s_0 is the contribution from the diffraction limit and s_{VCSEL} is the size of the VCSEL.

5.5.3 Effective Aperture

Many of the standard equation quoted are for a point source illuminating the entire aperture of the lens. In our case the source is a laser diode and may only be illuminating a small area of the lens. I have introduced an effective aperture to modify the standard equations. The effective aperture is the area of the lens 'seen' by the rays from the source, as shown in Figure 22.

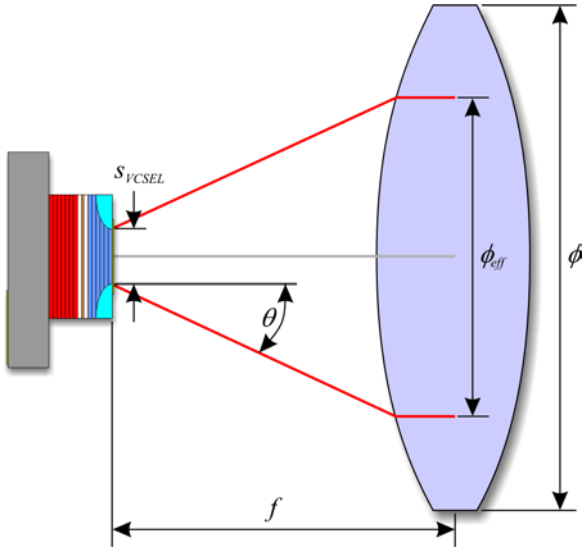

Figure 22: Effective Aperture
Diagram showing effective aperture, ϕ_{eff}.

To get the effective aperture, the effect of diffraction on the beam leaving the source aperture is calculated using:

$$\theta = \sin^{-1}\left(\frac{\lambda}{2s_{VCSEL}}\right) \qquad \textbf{Equation 32}$$

then:

$$\phi_{eff} = 2f\tan(\theta) + s_{VCSEL} = \frac{f\lambda}{s_{VCSEL}\left(1 - \frac{\lambda^2}{8s_{VCSEL}^2}\right)} + s_{VCSEL} \qquad \textbf{Equation 33}$$

If $\phi_{eff} > \phi$, use ϕ as ϕ_{eff}. There will be a loss of optical power:

$$\xi_{couple} = 1 - \frac{\phi_{eff}^2 - \phi^2}{\phi^2} \qquad \textbf{Equation 34}$$

that may affect the result in Equation 29 by reducing P_{VCSEL} to $\xi_{couple} P_{VCSEL}$.

5.5.4 Diffraction Limit s_0

The contribution to the spot size due to the diffraction limit is calculated using the effective aperture from above using the standard *Raleigh criteria* (RC). The use of the effective aperture insures that the correct stop in the system is used, i.e. if $\phi_{eff} < \phi$ the laser aperture is the limiting stop not the lens aperture and vice versa. s_0 is then:

$$s_0 = \left[-0.5\ln\left(1 - \frac{P}{P_0}\right)\right]^{1/2} \frac{4\lambda f}{\pi\phi_{eff}} \qquad \textbf{Equation 35}$$

Where P/P_0 is the fraction of the encircled optical power. For 99% encircled power Equation 35 becomes:

$$s_0 = 0.96\frac{\lambda f}{\phi_{eff}} \qquad \textbf{Equation 36}$$

5.5.5 Spherical Aberration s_1

The effect of *spherical aberration* (SA) is to bring rays further from to optic axis to a quicker focus. The new focal point for rays at the extreme of the lens aperture brought in by A_l, the *longitudinal spherical aberration* (LSA):

$$A_l = \frac{kf}{(f/\#)^2} \qquad \textbf{Equation 37}$$

where k is the SA coefficient as shown in Figure 23, f the focal length and $f/\#$ is the f-number.

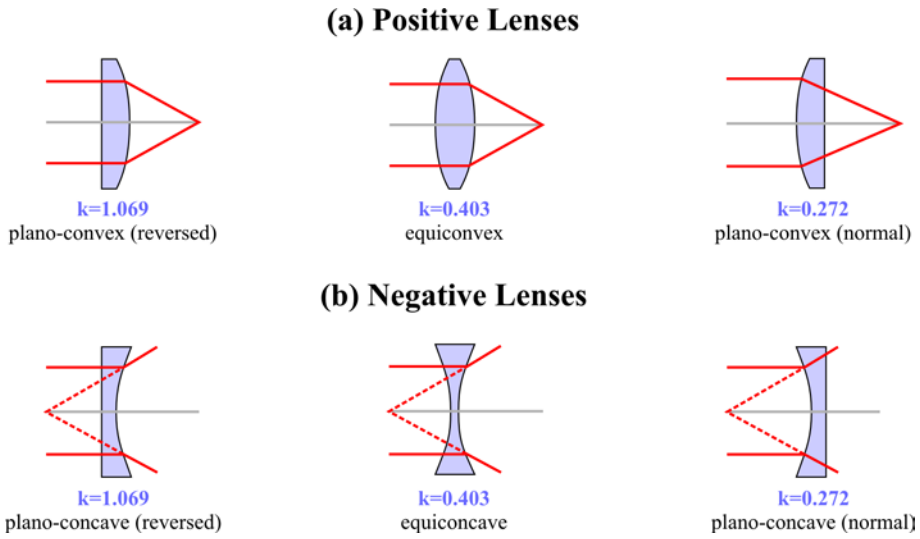

Figure 23: Spherical Aberration Constant *k* for Various Lens Singlets
By combining a positive and negative lens of different refractive index the aberration can be balanced and k=0 for the compound lens [94].

The first approximation is the extreme rays from the VCSEL are assumed to be parallel to the optic axis. The focal point for the extreme rays from the VCSEL is then pulled in by:

$$A_l^{'} = A_l \frac{\phi_{eff}^2}{\phi^2} \qquad \textbf{Equation 38}$$

The second approximation is that the second element in the 4f system has no spherical aberration. In this case the two approximations made should have an effect equal in magnitude but opposite on the outcome. The increase in spot size can then be calculated via the geometry of the system, as shown in Figure 24, giving:

$$s_1 = \frac{\phi_{eff}\left(f + A_l^{'}\right)}{\left(f - \frac{\phi_{eff}^2}{\phi^2}\right)} - \phi_{eff} \qquad \textbf{Equation 39}$$

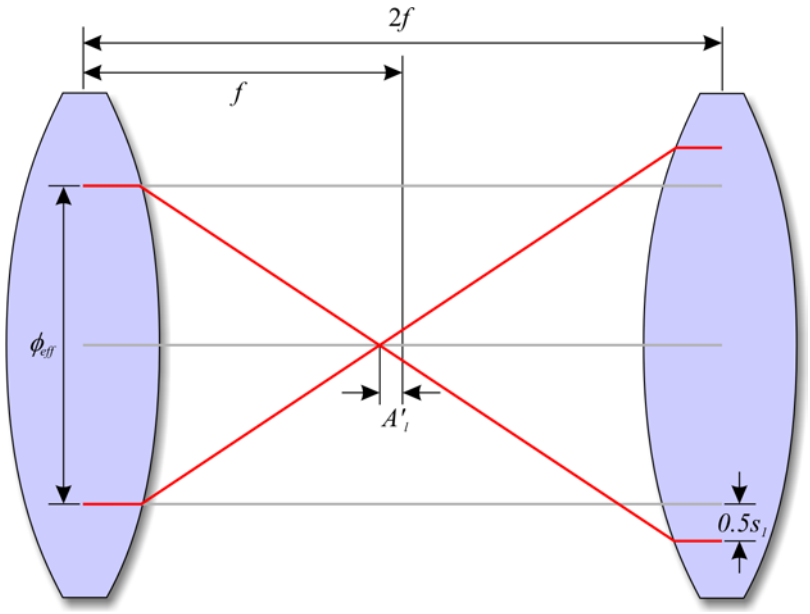

Figure 24: Optics Geometry
A'_l is the modified longitudinal spherical aberration for the effective aperture.

5.5.6 Other Aberrations s_2 etc.

All the optical systems considered in this analysis are monochromatic, usually at 850nm, so *chromic* aberrations are not an issue. Apart from monochromatic aberration, the only other concern is *distortion*. This form of aberration has little effect on the resolution, i.e. number of channels in the system, but affects the positioning of the channels spatially. This could be an issue if the channels were closely packed in the OH or the *optoelectronic* (OE) chip was very densely packed. Also the assumption in the spherical aberration analysis that the $fov < \phi$ reduces the effect of distortion.

Nevertheless, the effects of these aberrations can be dealt with through careful design of the system such that their impact becomes negligible.

5.5.7 Simulation and Results

Figure 25 shows the input beam waist (VCSEL size) vs. output beam waist (detector size) for 0, 10 and 200 relay stages. The lens parameters used were f=30mm, ϕ=7.5mm, k=0.272 and λ=850nm which are the SPOEC [95] lens parameters with a greatly exaggerated spherical aberration term k.

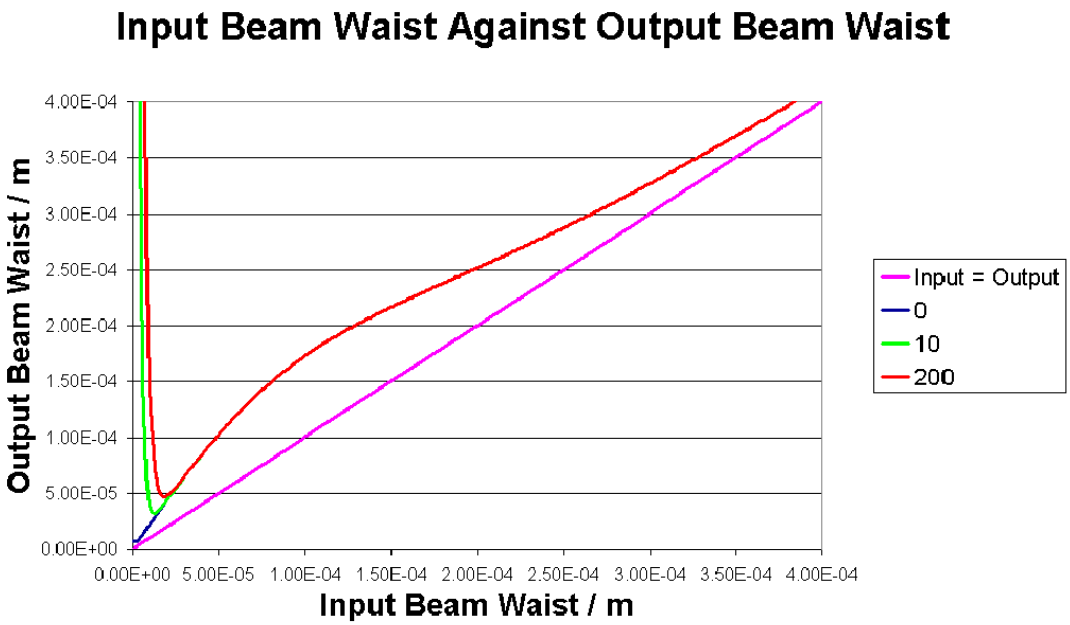

Figure 25: Input Beam Waist vs. Output Beam Waist
Graph of input beam waist (VCSEL size) vs. output beam waist (detector size) for 0, 10 and 200 relay stages.

From small input beams the spherical aberration of the system causes the sharp rise in the output beam size. At large beam sizes it is the diffraction limit due to the lens stops that is the limiting factor. The hump like feature between the two extremes is the area where the VCSEL source provides the system beam stop.

The total number of optical emitters N_{tx} and receivers N_{rx} required at a single node to fully interconnect a circular network with an even number of nodes is:

$$N_{tx} = N_{rx} = \frac{C_w p(p-1)}{2h_{\max}} \qquad \textbf{Equation 40}$$

This calculation allows estimation of the feasibility of any system. Current system scalability, or the maximum number of nodes, is limited by the number of emitters N_{tx} that can be fabricated on a single device and not the number of detectors N_{rx}. Table 13 summarises optical device characteristics based on what is available, what is due to market and what has already been built in the lab.

Generation	N_{tx}	f_{tx}	Raw Throughput ($N_{tx}B_{optical}$)	Technology
Available	320	0.5GHz	$20GBs^{-1}$	VCSEL
Due	1024	2.5GHz	$320GBs^{-1}$	VCSEL
Lab.	4096	110GHz	$56.32TBs^{-1}$	MQW

Table 13: Transmitter Scalability and Bandwidth
Optoelectronic transmitter scalability and performance limits. Raw throughput measured in bytes per second. These figures were valid in December 2001.

Note that the emitters and receivers must be capable of a comparable data rates as any physical link will run at the speed of the slowest component.

Assuming that circular spots are mapped onto a square grid, this makes the limit on the number of processors due to aberrations $p_{ab\max}$:

$$p_{ab\max} < \frac{\pi\phi^2}{2C_w s_{\max}^2} - 1 \qquad \textbf{Equation 41}$$

5.6 Node to Node Latency

This section defines latency given that the data path through the system is known. This latency figure could be a data request or the beginning of data transfer itself and will be incurred with each transfer that is initialised. By defining the data source and destination, be it processing system or smart pixel array memory, the latency figure L' can be calculated. The exact value is dependent on the components used in any implementation and is broken up into a series of latencies as shown in Figure 26.

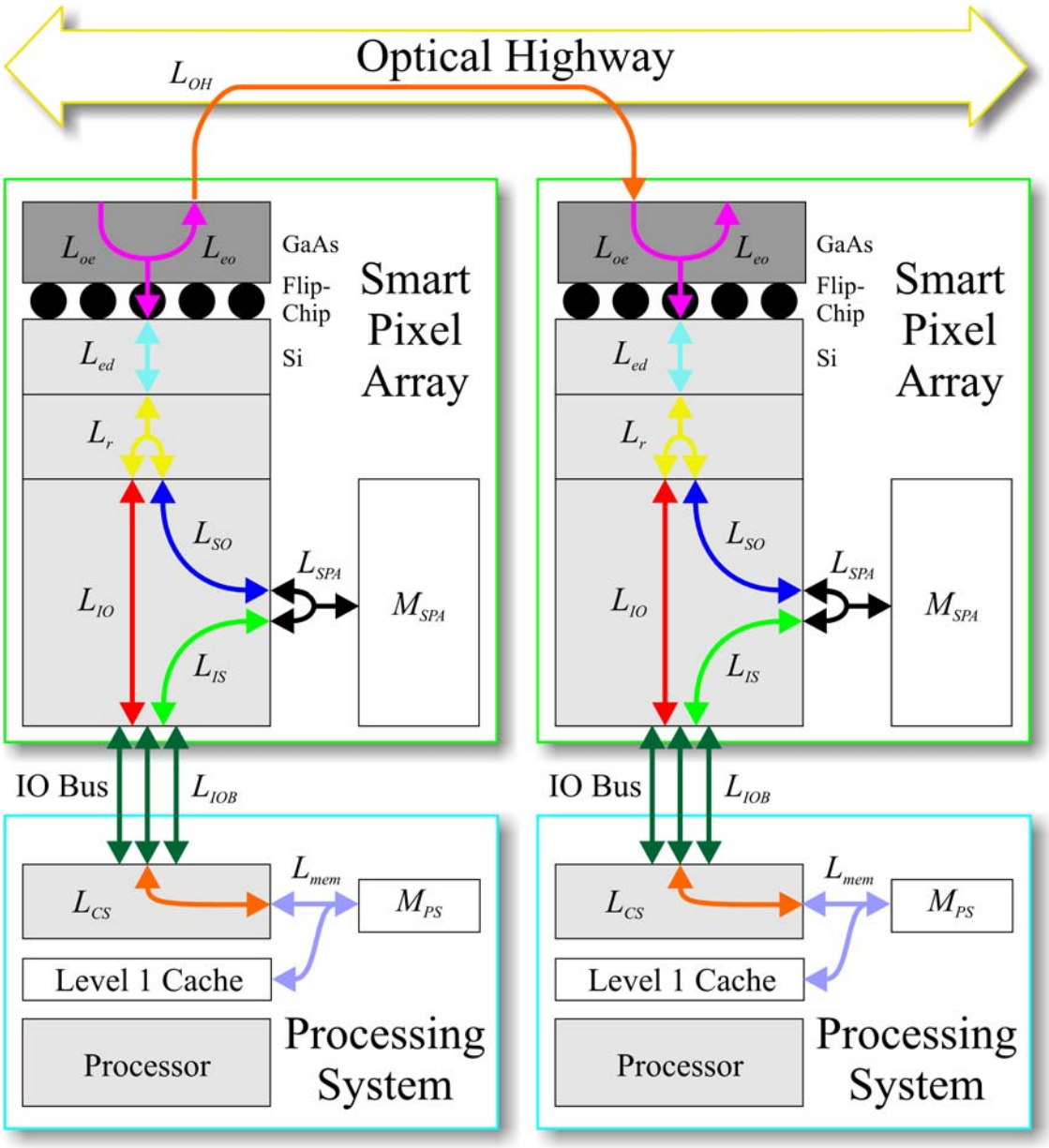

Figure 26: Node to Node Data Path Latency

The latency between selected source and destination nodes can be calculated by summing the above latencies for the appropriate path.

A description of each latency and its approximate magnitude is shown in Table 14.

Latency Description	Symbol	Typical Values	Ref.
SPA memory latency	L_{SPA}	65 to 120ns	Sections 3.2/3.3
SPA memory to OH	L_{SO}	5 to 15ns	[96]
I/O bus to SPA Memory	L_{IS}	5 to 15ns	[96]
SPA bypass	L_{IO}	5 to 15ns	[96]
Routing	L_r	10ns to 1µs	[97]
De/encoding	L_{ed}	20 to 200ns	[98]
Electronic-optical conversion	L_{eo}	100ps to 1ns	-
Optical-electronic conversion	L_{oe}	100ps to 1ns	-
Optical time of flight	L_{OH}	Architecture dependant	Section 5.3
I/O bus latency	L_{IOB}	20 to 200ns	[98]
Chip set latency	L_{CS}	5 to 15ns	[96]
Main memory latency	L_{mem}	65 to 120ns	Section 3.2

Table 14: Latency Figures
Order of magnitude values of latencies in the system. A *field programmable gate array* (FPGA) based SPA system is assumed. Latencies are highly implementation dependent and therefore cannot be accurately predicted.

L_{ed} is the latency associated with encoding the data to allow for error correction and clock reconstruction. For this analysis 8B/10B encoding is assumed, i.e. 8 bytes of data are encoded as 10 bytes. From [98], encoding takes 6 FPGA clocks. Decoding takes the same number of clocks so L_{ed} is also used as the decoding latency.

The smart pixel array has an associated controller latency which is incorporated into L_{IO}, L_{SO} and L_{IS}. This is not well defined as it is highly implementation dependent but empirical studies indicate 5-10ns values for conventional devices.

L_{CS} is a chipset specific latency delay. This value will typically be very small in conventional systems but will be of increasing significance as packet switched I/O bus architectures begin to appear.

The purpose of these figures is to allow calculation of both L' and L_{PC}. Taking the latter first, L_{PC} can be approximated as:

$$L_{PC} = L_{mem} + L_{CS} + L_{IOB} + L_{IOBld} \quad \textbf{Equation 42}$$

This is the latency from main memory to the end of the I/O bus. The one component that remains undefined is the I/O bus load latency (L_{IOBld}). This is the additional delay cycles required for the first bus width of information to reach the destination when the I/O bus is in use. If the maximum available and sustainable bandwidth is in use on a PCI bus, the specification guarantees that the delay will be, at worst, 3µs. Assuming that requests are randomly timed, the average delay would therefore be 1.5µs. Thus an estimation of latency based on load L_{IOBld} is:

$$L_{IOBld} = \frac{1.5 \times 10^{-6} \left(B_{per} + B_{act} \right)}{B_{avail}} \quad \textbf{Equation 43}$$

However, the size of data chunk transmitted is limited. Thus any reasonably large data transfer will be fragmented.

Calculating L' is slightly more complex since both source and destination must be determined. There are three possible routes for data depending on whether the source or destination is the smart pixel array or is directly to the I/O bus:

I/O bus to I/O bus:

$$L' = L_{OH} + L_{oe} + L_{eo} + 2L_{ed} + 2L_r + 2L_{IO} \quad \textbf{Equation 44}$$

I/O bus to SPA:

$$L' = L_{OH} + L_{oe} + L_{eo} + 2L_{ed} + 2L_r + L_{IO} + L_{SO} + L_{SPA} \quad \textbf{Equation 45}$$

SPA to SPA:

$$L' = L_{OH} + L_{oe} + L_{eo} + 2L_{ed} + 2L_r + 2L_{SO} + 2L_{SPA} \quad \textbf{Equation 46}$$

Note that L_r is a routing delay and will only be incurred if a signal must traverse two independent optical highways. This situation arises when a signal on an HC+ optical highway moves from, for example, a horizontal highway to a vertical one in order to reach its destination. In such a case, L_{OH} in the above three equations should be extended to:

$$L_{OH} = L_{OH}^{vert} + L_{oe} + L_{eo} + 2L_{ed} + L_r + L_{OH}^{horiz} \quad \textbf{Equation 47}$$

5.7 I/O Bus Bandwidth Modelling

The modelling of a node's I/O bus must account for bus bandwidth, bus efficiency and peripheral load. Therefore the bus bandwidth available for interconnection B_{avial} is defined as:

$$B_{avail} = B_{IOB}\xi_{IOB} - B_{per} \qquad \textbf{Equation 48}$$

where B_{IOB} is the theoretical maximum transfer rate of the I/O bus in bytes per second, ξ_{IOB} is the efficiency of the I/O bus from 0 to 1 and B_{per} the average bandwidth used by additional peripherals on the I/O bus again in bytes per second. If B_{avail} is less than or equal to zero then there is no free bandwidth for data transfer. Given that we have B_{avail} bytes/second at our disposal we can request a portion of this bandwidth. The actual requested portion is B_{act} and must be less than or equal to B_{avail}.

5.8 Memory Transfer Overheads

It is unavoidable that every I/O device in a system places resource demands on the CPU. These demands include [14]:

- Clock cycles for the instructions used to initiate I/O, to support operation of an I/O device such as handling interrupts, and to complete I/O.
- CPU clock stalls due to waiting for I/O to finish using the memory, bus or cache.
- CPU clock cycles to recover from an I/O activity, such as a cache flush.

To illustrate these points take for example access to a hard disk. A request is first sent over the I/O bus by the CPU to the hard drive controller for data from a specific block. There is a delay while the drive heads are positioned, usually around 5 to 10ms, and while reading takes place. Data is then read into a buffer within the hard drive. This delay time is the commonest definition of a blocked process waiting on I/O. The operating system may switch to another process and continue as there is no direct access to main memory and millisecond delay times are significant in computer systems. However, the fastest way to currently transfer data from hard drive to main memory is using DMA. Such a method halts the CPU, takes over control of the memory bus and transfers then data. During this time, it is not possible for the processor to access main memory as the address and data lines are being used by the DMA controller. In the best situation, a CPU can finish work on any data already present in the cache.

Although DMA would seem a near optimal solution, it does have its problems mainly arising from the alteration of a memory segment held in cache. Under such circumstances, and depending on caching strategy used, the CPU may have to flush its pipeline and cache to ensure data integrity. This can be costly in terms of machine cycles. The hope would be that process requesting data is blocked and therefore not held in cache memory.

The optical highway concept assumes that large volumes of data can be transferred with minimal delay. This technically means that it should be possible to saturate the I/O bus with information transfers. Such transfers hold the CPU, thereby preventing any further processing. It is possible to circumvent this problem in either one of two ways:

- Use a multiprocessor architecture with multiple memory banks. Therefore one bank could be put out of operation while DMA occurs. However, the optoelectronic highway is designed to interconnect such systems rather than have multiple processors at each node.
- Use speed matching buffers that accumulate data from the slower I/O bus and transfer it in a burst over the main memory bus. Thus CPU downtime is related to the bandwidth consumed on the memory bus plus any protocol overhead.

The latter method has become commonplace in today's I/O bus architectures such as PCI. This behaviour will be assumed to be that used by a processing system at any node. This information allows us to estimate the CPU time lost to data transfer as a ratio:

$$t_{CPU} = \frac{(B_{per} + B_{act})p_{ov}}{B_{mem}} \qquad \textbf{Equation 49}$$

where B_{mem} is the main memory bandwidth in bytes per second. If t_{CPU} is 0.1 then 10% of the CPU's time in the next second will be taken up by data transfer. If this value is 1 then 100% will be taken up by data transfer. A value of greater than 1 indicates that memory bandwidth B_{mem} is insufficient to continually handle this level of I/O.

A very important machine dependent variable here is p_{ov}. This value indicates the processor overhead for transfer. A value of zero indicates that the CPU and I/O interfaces are somehow independent and the transfer does not influence cache etc. This could be the case in a shared memory system. A value of 0.2 indicates a 20% overhead and could be the case in a shared memory system where data is written to a location in

use by a process. This write has resulted in an invalidation of data in CPU cache and pipeline stall. Although not a problem to recover from, the transfer is still noticeable. A value of 1 indicates that there is a 100% overhead. This is the case in most systems as a CPU cannot use the same memory lines while another device is. A special case may arise if portions of memory are cached and the CPU can continue working on these while access occurs resulting in an overhead of 0.8. In most commodity systems this overhead will unfortunately exceed 1. This indicates that in addition to raw data transfer there is additional signalling taking place. The problem could then be exaggerated by access to a memory location currently in cache, usually the OS process requesting I/O, resulting in a cache flush and pipeline stall. For example, 1.2 indicates an additional 20% overhead.

This allows a quick estimate of the number of operations a CPU has left N_{op} per second from its maximum N_{max}:

$$N_{op} = N_{\max} - N_{\max} t_{CPU} - N_{ov} \qquad \textbf{Equation 50}$$

Note that N_{ov} allows representation of any fixed overheads and will under most circumstances be zero.

5.9 Conclusions

Extensive simulation using the equations detailed in this document allow us to reach the following conclusions about the optical highway hardware:

- The limiting factors in the construction of such a system are the optical power output of emitter devices and the fabricated density of these devices.
- The receiver electronics are the primary bandwidth limit and not, as initially suspected, the optical system.
- The number of spatial channels can be traded off against the bandwidth of each channel.
- The electronic to optical conversion latencies are two orders of magnitude smaller than the data encode and decode latencies. Thus the penalty paid to convert a signal to the optical domain and back is much smaller than the penalty paid when an electrical signal must be converted and encoded for transmission over a similar distance.

- The latencies involved in a conventional I/O bus are significantly higher than those in the optical highway.

The limiting factor in this system is the location of the processing system. Having to traverse an I/O bus drastically increases latency and decreases bandwidth. Ideally, all processing should be performed as close to the optical highway as possible. This alludes to intrinsic problems in the architectures of conventional computer systems. Indeed, the authors believe that the optical highway should be far more closely intertwined with both processor and memory to the point of being the transmission medium between memory and the processor itself.

The main advantage of having a commodity optics-based intelligent interconnect is that it is independent of the nuances attributed to the performance gap between advanced processors and other developing system components. With a modular, 'plug-and-play' implementation and capabilities for reconfiguration one can deploy it as a router or for other dedicated purposes.

Specialised hardware is required in most cases to address issues such as physically independent memory blocks at a single CPU node with dedicated memory channels. Integration of optical systems at an I/O level does not seem to address bandwidth issues. Tighter integration appears to be required, precluding any application in commodity systems.

6 Conclusions

Parallel buses in their current incarnations, be they I/O, address or data, are reaching the end of their working life. Packet based serial networking technologies that started out as wide area network protocols are ever encroaching on short range interconnection. There is a large amount of evidence to support this conclusion in that such protocols are appearing in processor to memory interconnect in the form of Rambus, in I/O bus technologies such as Hypertransport, RapidIO, 3GIO, in HDD interfaces like serial ATA (SATA) and in peripheral interconnect such as USB versions 1, 2 and IEEE 1394 Firewire 400 and 800.

The increasing shortfall in bus bandwidths have thus far been eased using brute force, in particular judicious caching. Processors now sport up to three different levels of cache which can easily be megabytes in size. Increasing the memory bus bandwidth to that required by the processor would mean that the concept of caching must be re-evaluated considering only latency. Thus, certain caching levels would become redundant, freeing up real estate on a chip and decreasing overall die size.

The results described here indicate that the current I/O architectures in commodity systems are the limiting factor to the introduction of Optical Highway style systems. However, they also show that it would be advantageous to integrate an optical system tightly with a computer system where the optical highway itself is used to interconnect a processor with its main memory at essentially a physically separate node. Admittedly, this would be a custom implementation and not commodity system, but the bandwidth capacity of even main memory buses is not growing rapidly enough to sustain development. Such integration would result in a logically flat multiprocessor architecture with many processors capable of addressing many different memory banks, each with an latency based on physical separation.

Ultimately, the algorithm is the deciding factor. The adaptation of any algorithm into its parallel form for use on the optical highway may prove difficult, if not impossible. However, regardless of the compute and interconnect capability available, a poorly designed algorithm can reduce a potentially powerful system to nothing more than a crawl.

7 Variable Definitions

Note that capitals are used in subscripts to represent an acronym. If lowercase is used then the letters represent a part or all of a single word.

A_l Longitudinal spherical aberration or L.SA. Measured in meters (m).

$A_l^{'}$ Modified longitudinal spherical aberration for the effective aperture. Measured in meters (m).

B' Optical node-to-node bandwidth in one direction. Measured in bytes per second (Bs^{-1}).

B_{act} Bandwidth transfer requested by algorithm through the I/O bus. This must be equal to or less than the available bandwidth. Measured in bytes per second (Bs^{-1}).

B_{avail} Best practically available bandwidth on I/O bus given a defined efficiency. Measured in bytes per second (Bs^{-1}).

B_{IOB} Maximum theoretical bandwidth transfer available on the I/O bus. Measured in bytes per second (Bs^{-1}).

B_{mem} Maximum theoretical bandwidth transfer available to processing system main memory. Multiply by memory architecture efficiency for the actual figure. Measured in bytes per second (Bs^{-1}).

$B_{optical}$ Bandwidth of data through the optical highway, node to node, down a single physical channel. It is essentially the emitter/detector raw transmission rate. Encoding must be included in this figure. Measured in bytes per second (Bs^{-1}).

B_{per} Maximum bandwidth transfer used on I/O bus by peripheral devices. This is dependent on architecture; for example some architectures use the I/O bus to access the hard-disk. This must be equal to or less than the available bandwidth. Measured in bytes per second (Bs^{-1}).

c Universal constant speed of light $c = 3.00 \times 10^8 \, ms^{-1}$.

C_d The junction capacitance of a photodiode. Measured in Farads (F).

C_v Number of virtual interconnection channels. Virtual channels are considered to carry data transparently in both directions. This is a dimensionless number.

C_p Number of physical interconnection channels. This number represents the number of physical links that need to be established and thus there must be an equivalent number of VCSELs and detectors. This is a dimensionless number.

C_w Number of physical channels used to construct a single virtual channel in one direction. This is normally 1. Any increase in this number will decrease node-to-node transmission times through increased bandwidth. This is a dimensionless number.

f Focal length of optical highway lenses. Measured in meters (m).

$f/\#$ Lens *f*-number or the proportional inverse of the numerical aperture. This is a dimensionless number.

fov Field of view. Measured in meters (m).

f_{tx} Maximum operational speed of a specific transmitter device. Measured in Hertz (Hz).

Δf The bandwidth over which noise is measured in a photodiode. Usually assumed to be 1 Hertz (Hz).

h The distance between two nodes or hops. Moving from one node to a directly adjacent node in any direction is counted as a single hop. It is a dimensionless integer number.

$h_{\max}$ The maximum number of hops across an optical highway. This is a dimensionless integer number.

I_d Current in a photodiode under forward bias conditions. Under normal circumstances, when the photodiode is in reverse or unbiased, this value will be 0. Measured in amps (A).

I_{dk} The current present in a photodiode when no optical radiation is incident and a reverse bias is applied. Measured in amps (A).

I_j Johnson or Nyquist noise generated by random thermal excitation of electrons. Measured in amps (A).

I_n Total rms noise current. This is a combination of both Johnson and Shot noise. Measured in amps (A).

I_p Total current in the photodiode. Measured in amps (A).

I_s Shot or white noise. This is a statistical variation calculated using both incident optical power and dark current. Measured in amps (A).

I_{th} Threshold current at which the laser begins to lase. Measured in amps (A).

I_t The current generated in a photodiode by incident light. Measured in amps (A).

k Boltzmann's constant $1.38 \times 10^{-23}\,\mathrm{JK}^{-1}$.

k Spherical aberration constant of lenses used in the optical highway. 0 indicates a perfect design however 0.2 to 0.3 is typical for a well made component. Poor quality components may exceed 1. This constant is a dimensionless number.

L' Latency from a designated source, either smart pixel array cache or I/O bus, to an equivalently designated destination on a different node. Measured in seconds (s).

L_{CS} Latency of the system chip-set which includes traversal of the south bridge to the north bridge. The remainder of the journey is considered to be memory latency L_{mem}. Measured in seconds (s).

L_{ed} Smart pixel array encoding or decoding latency. Measured in seconds (s).

L_{eo} Latency of electronic to optical conversion including transmission time to the optoelectronic device. Measured in seconds (s).

L_{IO} Latency of data transfer through the smart pixel array between the I/O bus and optical system. Measured in seconds (s).

L_{IOB} Latency from the north bridge to the end of the I/O bus. Measured in seconds (s).

L_{IOBld} IO bus latency with respect to load. In a PCI based system this can be anything from 0 to the guaranteed maximum of 3µs. This is a statistical process but a value of 1.5µs is assumed throughout this document given maximum load conditions. Measured in seconds (s).

L_{IS} Latency of data transfer through the smart pixel array between the I/O bus and cache memory. Measured in seconds (s).

L_{mem} Latency of main memory per read cycle including traversal of the north bridge. Measured in seconds (s).

L_{oe} Latency of optical to electronic conversion including transmission time from the optoelectronic device to its destination. Measured in seconds (s).

L_{OH} Latency due to time of flight through the optical system. This will vary from a minimum when only a single hop is traversed to a maximum when $h_{\max}$ hops are traversed. Measured in seconds (s).

L_{PC} Latency from main memory to end of I/O bus. Measured in seconds (s).

L_r Smart pixel array routing latency. Measured in seconds (s).

L_{SO} Latency of data transfer through the smart pixel array between the cache memory and optical system. Measured in seconds (s).

L_{SPA} Latency of smart pixel array cache memory. Measured in seconds (s).

M_{PS} Memory available on processing system. Measured in bytes (B).

M_{SPA} Smart pixel array memory buffer size. Measured in bytes (B).

n Refractive index of majority optical carrier in the optical highway. Assuming the system is more than 95% air, $n \approx 1$. It is a dimensionless number.

NEP Noise equivalent power is the amount of incident optical power required on a photodetector to generate a signal of equal magnitude to inherent device noise. In such a case, SNR=1. It is measured in $\mathrm{WHz}^{-\frac{1}{2}}$. Typical values range from $1\times10^{-11}\,\mathrm{WHz}^{-\frac{1}{2}}$ for large active area photodiodes to $1\times10^{-15}\,\mathrm{WHz}^{-\frac{1}{2}}$ for small area photodiodes.

$N_{\max}$ Maximum number of operations per second that can be performed by a processing system. Measured in operations per second (ops^{-1}).

N_{op} Number of operations per second that a processing system has left given that a specified amount of information is transferred to or from main memory. Measured in operations per second (ops^{-1}).

Variable Definitions

N_{ov} Constant processing system overhead given sustained transfer through the I/O bus. Measured in operations per second (ops^{-1}).

N_{rx} Number of receivers at a single node. This is an integer number.

N_{tx} Number of transmitters at a single node. This is an integer number.

p The number of processing system and smart pixel array pairs. This combination is referred to as a node. It is a dimensionless integer number.

$p_{ab\max}$ Maximum number of processing systems that an optical highway can support based on lens system aberrations. It is a dimensionless integer number.

p_d Destination processing node. This is a dimensionless integer number where 1 represents the first node and p the last.

p_{dx} Destination processing node x coordinate. This is a dimensionless integer number where 1 represents the first column in an HC+ architecture and $\sqrt{p}$ the last.

p_{dy} Destination processing node y coordinate. This is a dimensionless integer number where 1 represents the first row in an HC+ architecture and $\sqrt{p}$ the last.

$p_{p\max}$ Maximum number of processing systems that an optical highway can support based on available optical power. It is a dimensionless integer number.

p_s Source processing node. This is a dimensionless integer number where 1 represents the first node and p the last.

p_{sx} Source processing node x coordinate. This is a dimensionless integer number where 1 represents the first column in an HC+ architecture and $\sqrt{p}$ the last.

p_{sy} Source processing node y coordinate. This is a dimensionless integer number where 1 represents the first row in an HC+ architecture and $\sqrt{p}$ the last.

P_{det} Minimum power detectable at the detector in watts (W).

P_i Incident optical power on a detector in watts (W).

P_{VCSEL} Power emitted by a single VCSEL in watts (W).

P_{Velec} VCSEL electrical power in watts (W).

P/P_0 Fraction of encircled optical power. This is a dimensionless number.

q Optical highway node to node distance. It is dependent on selected source and destination nodes. Measured in meters (m).

q' Universal distance between two adjacent nodes along the optical highway. Measured in meters (m).

q'' Distance from node emission/detection point to insertion point beam splitter on the optical highway. Measured in meters (m).

q''' Distance between beam splitter and mirror at extraction point on optical highway. Measured in meters (m).

$rx_{i,j}$ A receiver on a node indexed by the subscript. Node number is identified by i and detector device number by j. This is a dimensionless integer number.

R_d Photodiode shunt resistance. Used to determine the noise current with no incident light. Measured in Ohms (Ω).

R_f Photodiode amplifier feedback resistance. Determines the level of photodiode signal amplification. Measured in Ohms (Ω).

R_s Photodiode series resistance. Determines the linearity of response of the photodiode in photovoltaic mode. Measured in Ohms (Ω).

$s_{\max}$ Spot diameter of VCSEL image on last detector after $h_{\max}$ hops. Measured in meters (m).

SNR Signal to noise ratio. This is a dimensionless number.

s_{VCSEL} Input beam waist to system, equivalent to maximum diameter of VCSEL spot. Measured in meters (m).

$s_{\max}$ Spot diameter after $h_{\max}$ hops. Measured in meters (m).

s_0 The contribution to the spot size due to the diffraction limit. Measured in meters (m).

$s_1, s_2, \ldots$ The contributions to the spot size due to the aberrations in the system. Measured in meters (m).

Variable Definitions

t_{CAC} Column access delay. A preprogrammed number of clock cycles to allow for signal propagation on the memory bus, in this case that of RDRAM. The exact value is dependent on the number of RDRAM devices on the channel and the RDRAM timing bin. Normally written as clock cycles and measured in seconds (s).

t_{CPU} CPU time lost due to data transfer between I/O and system memory. This value is dimensionless and ranges from 0 (0%) for no impact to 1 (100%) for complete bus usage and thereby overloading the system.

t_{RCD} The delay between RAS and CAS signals when opening a memory page. Assuming that subsequent accesses are to the same page then this delay will only occur once during any memory operation. Typically written as clock cycles and measured in seconds (s).

$tx_{i,j}$ A transmitter on a node indexed by the subscript. Node number is identified by i and transmitter device number by j. This is a dimensionless integer number.

T Absolute temperature. Measured in Kelvin (K).

V_{out} Final output voltage from the photodiode amplifier. Measured in volts (V).

V_{th} Threshold voltage at which the laser begins to lase. Measured in volts (V).

ϕ Diameter of lens aperture. Measured in meters (m).

ϕ_{eff} An approximation of the diameter of the effective limiting stop aperture. Measured in meters (m).

λ Optical highway operational wavelength at which peak power exists. Measured in meters (m).

π Constant Pi which has a value of 3.14159.

θ The effect of diffraction on a beam leaving the source aperture. This is a half angle and is measured in degrees (º).

$\Re$ The responsivity of a photodiode at a specific wavelength. This is the amount of current generated given a watt of incident optical power. The units AW^{-1}.

ξ Efficiency per optical stage. This value is a dimensionless number.

ξ_{couple} Coupling efficiency of the light into the first stage of the system. This value is a dimensionless number.

ξ_{ed} Encoding efficiency for transmission in the optical domain. This value ranges between 0.0 for total inefficiency and 1.0 for perfect efficiency. Typical values are around 0.8, if 8B/10B clock encoding is used, which indicated that 10 bytes are used to encode 8 bytes of information. This value is a dimensionless number.

ξ_{HWP} Transmission efficiency of a single half wave plate (HWP). This value is dimensionless and ranges from opaque at 0.0 (0%) to perfectly transparent at 1.0 (100%).

ξ_{IOB} Practical efficiency of I/O bus. This value is dimensionless and ranges from unusable at 0.0 (0%) to a perfect bus with no overhead at 1.0 (100%).

ξ_{l} Laser slope efficiency. This value is dimensionless and ranges from a poor power conversion efficiency at 0.0 (0%) to perfect at 1.0 (100%).

ξ_{lens} Transmission efficiency of a single system lens. This value is dimensionless and ranges from opaque at 0.0 (0%) to perfectly transparent at 1.0 (100%).

ξ_{mem} Efficiency of main memory implementation. This value is dimensionless and ranges from unusable at 0.0 (0%) to a perfect system with no overhead at 1.0 (100%).

ξ_{mirror} Reflection efficiency of a single mirror. This value is dimensionless and ranges from transparent at 0.0 (0%) to perfectly reflective at 1.0 (100%).

ξ_{PBS} Transmission efficiency of a single polarising beam splitter (PBS). This value is dimensionless and ranges from opaque at 0.0 (0%) to perfectly transparent at 1.0 (100%).

ξ_{QWP} Transmission efficiency of a single quarter wave plate (QWP). This value is dimensionless and ranges from opaque at 0.0 (0%) to perfectly transparent at 1.0 (100%).

8 Glossary

3GIO	3rd Generation Input/Output
AGP	Accelerated Graphics Port
APD	Avalanche Photodiode
AR	Anti Reflective
BER	Bit Error Rate
CAS	Column Address Strobe
CL	CAS Latency
COH	Circular Optical Highway
CPU	Central Processing Unit
CW	Continuous Wave
DBR	Distributed Bragg Reflector
DDR	Double Data Rate
DH	Double Heterojunction
DMA	Direct Memory Access
DOE	Diffractive Optic Element
DRAM	Dynamic Random Access Memory
EISA	Extended Industry Standard Architecture
FKE	Franz-Keldysh Effect
FPGA	Field Programmable Gate Array
HC+OH	Hypercube Plus Optical Highway
HDD	Hard Disk Drive
HWP	Half Wave Plate
IBM	International Business Machines
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers

ILP	Integrated Layer Processing
I/O	Input/Output
ISA	Industry Standard Architecture
L1	Level 1
L2	Level 2
L3	Level 3
LASER	Light Amplification by Stimulated Emission of Radiation
LCD	Liquid Crystal Display
LD	LASER Diode
LED	Light Emitting Diode
LIFO	Last-in-First-out
LOH	Linear Optical Highway
LSA	Longitudinal Spherical Aberration
MCA	Micro Channel Architecture
MQW	Multiple Quantum Well
NB	North Bridge
NEP	Noise Equivalent Power
NI	Network Interface
NIR	Near Infra Red
OE	Opto-Electronic
OH	Optical Highway
OS	Operating System
PBS	Polarising Beam Splitter
PC	Personal Computer
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
PCI-SIG	Peripheral Component Interconnect Special Interest Group

PCI-X	Peripheral Component Interconnect Extended
PD	Photodiode
PIO	Programmed Input/Output
PS	Processing System
QCSE	Quantum Confined Stark Effect
QWP	Quarter Wave Plate
RAM	Random Access Memory
RAS	Row Address Strobe
RC	Raleigh Criteria
RDRAM	Rambus Dynamic Random Access Memory
RISC	Reduced Instruction Set Computer
SA	Spherical Aberration
SATA	Serial Advanced Technology Attachment
SB	South Bridge
SCIOS	Scottish Collaborative Initiative in Optical Sciences
SCSI	Small Computer System Interface
SDRAM	Synchronous Dynamic Random Access Memory
SLM	Spatial Light Modulator
SMB	Speed Matching Buffers
SMP	Symmetric Multiprocessing
SNR	Signal to Noise Ratio
SPA	Smart Pixel Array
SPOEC	Smart Pixel Opto-Electronic Connections
SRAM	Static Random Access Memory
TLB	Translation Lookaside Buffer
TN	Twisted Nematic
UDMA	Ultra Direct Memory Access

UDMA	User-level Direct Memory Access
USB	Universal Serial Bus
VCSEL	Vertical Cavity Surface Emitting LASER
VESA	Video Electronics Standards Agency
VL	VESA Local
VLSI	Very Large Scale Integration
VM	Virtual Memory
ZBT	Zero Bus Turnaround

9 References

[1] K. J. Symington, *Optically Interconnected Computing Systems*, Ph.D. Thesis, Heriot-Watt University, November 2001.

[2] W. Buchanan, *Computer Busses: Design and Application*, Arnold Publishers, 2000.

[3] H. Gilbert, *The I/O Bus*, http://sophia.dtp.fmph.uniba.sk/pchardware/bus.html, May 2002.

[4] J. Ögren, "ISA (Technical)", *The Hardware Book*, http://sunsite.tut.fi/hwb/co_ISA_Tech.html, January 2002.

[5] W. L. Rosch, "Local Bus: Speeding Video Display", *PC Magazine*, pp. 325-331, November 1992.

[6] *PCI V2.3 Specifications*, PCI Special Interest Group (PCI-SIG), http://www.pci-sig.com, January 2002.

[7] *ISA or PCI? Which is better for your business?*, IBM Corporation, http://www.networking.ibm.com/trl/trlwhit.html, 1996.

[8] H. Eda and M. Oishi, "Post-PCI Interfaces to Adopt Packet Switching", AsiaBizTech, http://www.nikkeibp.asiabiztech.com/nea/200109/peri_140009.html, January 2002.

[9] *HyperTransport I/O Link Specifications*, The HyperTransport Consortium, http://www.hypertransport.org/documentation/specifications_html, May 2002.

[10] M. Bedford, "Next Generation Input Output", *PC Plus*, pp. 27, April 1999.

[11] IOSS, *RD2 PC Geiger*, http://www.ioss.com.tw/web/English/RD2PCGeiger.html, May 2002.

[12] L. Logan, *Matching Data Bus Throughput to Data Acquisition Needs*, http://www.evaluationengineering.com/pctest/articles/e707data.htm, May 2002.

[13] M. James, "Critical Computing - Babbage's Bag", *Computer Shopper*, pp. 679-680, August 1999.

[14] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, Palo Alto, CA, Morgan Kaufmann Publishers, 1996.

[15] J. B. Carter, A. Davies, R. Kuramkote, C-C Kou, L. B. Stoller and M. Swanson, *Avalanche: A Communication and Memory Architecture for Scalable Parallel Computing*, Technical Report UUCS-95022, University of Utah, 1995.

[16] A. Agarwal and A. Gupta, "Memory Reference Characteristics of Multiprocessor Applications under MACH", *Proc. ACM Sigmetrics Conference on Measurement and Modelling of Computer Systems*, pp. 215-225, May 1988.

[17] P. Druschel, M. B. Abbott, M. A. Pagels and L. L. Peterson, "Network Subsystem Design", *IEEE Network*, vol. 74, pp. 8-17, July 1993.

[18] N. R. Mahapatra and B. Venkatrao, "The Processor-Memory Bottleneck: Problems and Solutions", *ACM Crossroads*, http://www.acm.org/crossroads/xrds5-3/pmgap.html, May 2002.

[19] D. C. Burger, J. R. Goodman and A. Kagi, "Limited Bandwidth To Affect Processor Design", *IEEE Micro*, vol. 17, no. 6, pp. 55-62, November/December 1997.

[20] K. J. Richardson and M. J. Flynn, "Strategies to Improve I/O Cache Performance", *Proc. of the 26th Hawaii Int. Conf. on System Sciences*, vol. 1, pp. 31-39, 1993.

[21] J. C. Mogul and A. Borg, "The Effect of Context Switches on Cache Performance", *4th Int. Conf. on Architectural Support for Programming Languages and Operating Systems*, SIGARCH Computer Architecture News, Santa Clara, CA, vol. 19, pp. 75-84, April 1991.

[22] A. J. Smith, "Cache Memories", *ACM Computing Surveys*, vol. 14, no. 3, pp. 473-530, September 1982.

[23] M. A. Pagels, P. Druschel and L. L. Peterson, "Cache and TLB Effectiveness in Processing Network I/O", *Technical Report TR 94 08*, Dept. of Computer Science, University of Arizona, Tucson, Arizona, 1994.

[24] R. H. Saavedra and A. J. Smith, "Measuring Cache and TLB Performance and Their Effect on Benchmark Run Times", *IEEE Transactions on Computers*, pp. 1223-1235, October 1995.

[25] M. Kampe and F. Dahlgren, "Exploration of the Spatial Locality on Emerging Applications and the Consequences for Cache Performance", *Proc. of 14th Int. Parallel and Distributed Processing Symposium*, pp. 163-170, 2000.

[26] J. Liedtke, "Caches Versus Object Allocation", *Proc. of the 5th Int. Workshop on Object-Orientation in Operation Systems*, pp. 95-101, 1996.

[27] K. Bates, *VAX I/O Subsystems: Optimizing Performance*, Professional Press Books, Horshmam, PA, pp. 81-90, 1991.

[28] S. S. Mukherjee and M. D. Hill, "The Impact of Data Transfer and Buffering Alternatives on Network Interface Design", *Proc. 4th Int. Symp. on High Performance Computer Architecture*, pp. 207-218, 1998.

[29] P. Druschel and L. L. Peterson, "Fbufs: A High-Bandwidth Cross-Domain Transfer Facility", *Proc. 14th Symp. on Operating Systems Principles*, 1993.

[30] R. Comerford and G. F. Watson, "Memory Catches Up", *IEEE Spectrum*, vol. 29, no. 10, pp.34-57, Oct. 1992.

[31] Rambus Inc., *64/72M Direct RDAM Data Sheet*, DL 0035-00.c0.5.28, http://www.rambus.com/documentation.html, March 1998.

[32] S. I. Hong, S. A. McKee, M. H. Salinas, R. H. Klenke, J. H. Aylor and W. A. Wulf, "Access Order and Effective Bandwidth for Streams on a Direct Rambus Memory", *Proc. of the 5th Int. Symp. on High Performance Computer Architecture*, pp.80-89, 1999.

[33] S. Przyblski, *New DRAM Technologies: A Comprehensive Analysis of the New Architectures*, 2nd Edition, MicroDesign Resources, Sunnyvale, CA, 1997.

[34] Rambus Inc., *Direct RDRAM*, Document DL0059, Version 1.11, http://www.rdram.com/downloads/rdram.128s.0059-1.11.book.pdf, August 2002.

[35] N. Hendrich, “Speicherhierarchie: DRAM, Cache, IRAM”, *Vorlesung PC-Technologie,* Kapitel 4, http://tech-www.informatik.uni-hamburg.de/lehre/pc-technologie/, April 2001.

[36] Rambus Inc., *RDRAM Memory: Leading Performance and Value over SDRAM and DDR,* Document WP0001-R, Version 1.2, August 2002.

[37] R. Crisp, "Direct Rambus Technology: The New Main Memory Standard", *IEEE Micro*, pp. 18-28, November/December 1997.

[38] S. Sassen, “Lies, Damned Lies, and a Different Perspective: RDRAM vs. SDRAM Performance”, Hardware Central, http://www.hardwarecentral.com/hardwarecentral/reports/1686/4/, August 2002.

[39] M. Kumanoya, T. Ogawa and K. Inoue, "Advances in DRAM Interfaces", *IEEE Micro*, vol. 15, no. 6, pp. 30-36, December 1995.

[40] E. P. Markatos and M. G. H. Katevenis, "User-Level DMA without Operating System Kernel Modification", *3rd Int. Symp. on High Performance Computer Architecture*, pp. 322-331, 1997.

[41] X. Zhang, Y. Yan and K. He, "Evaluation of Multiprocessor Latency Patterns", *Proc. of the 8th Int. Parallel Processing Symposium*, pp. 845-852, 1994.

[42] M. A. Blumrich, C. Dubnicki, E. W. Felten and K. Li, "Protected, User-level DMA for the SHRIMP Network Interface", *Proc. 2nd Int. Symp. on High Performance Computer Architecture*, San Jose, CA, pp. 154-165, February 1996.

[43] S. S. Mukherjee, B. Falsafi, M. D. Hill and D. A. Wood, "Coherent Network Interfaces for Fine-Grain Communication", *Proc. 23rd Annual Int. Symp. on Computer Architecture*, pp. 247-258, May 1996.

[44] I. Chlamtac and A. Ganz, "A Study of Communication Resource Allocation in a Distributed System", *Proc. of the 10th Int. Conf. of Distributed Computing Systems*, pp. 530-536, 1990.

[45] H. J. Round, "A Note on Carborundum", *Electrical World*, vol. 49, pp. 309, 1907.

[46] S. M. Sze, *Semiconductor Devices: Physics and Technology*, John Wiley & Sons, ISBN 0471837040, pp. 9-15, 1985.

[47] S. D. Smith, *Optoelectronic Devices*, Prentice-Hall, ISBN 0131437690, 1995.

[48] S. A. Campbell, *The Science and Engineering of Microelectronic Fabrication*, Oxford University Press, ISBN 0195105087, 1996.

[49] K. Gillessen and E. Schairer, *Light Emitting Diodes: An Introduction*, Prentice-Hall, ISBN 0135365333, 1987.

[50] Eds. S. Nakamura and S. F. Chichibu, *Introduction to Nitride Semiconductor Blue Lasers and Light Emitting Diodes*, Taylor & Francis, ISBN 0748408363, 2000.

[51] A. Yariv, *Quantum Electronics*, Third Edition, John Wiley & Sons, ISBN 0471617717, 1987.

[52] R. N. Hall, G. E. Fenner, J. D. Kingsley, T. J. Soltys and R. O. Carlson, "Coherent Light Emission From GaAs Junctions", *Phys. Rev. Lett.*, vol. 9, no. 9, pp. 366-368, November 1962.

[53] N. Holonyak Jr. and S. F. Bevacqua, "Coherent (Visible) Light Emission From $Ga(As_{1-x}P_x)$ Junctions", *Appl. Phys. Lett.*, vol. 1, no. 4, pp. 82-83, December 1962.

[54] M. I. Nathan, W. P. Dumke, G. Burns, F. H. Dill Jr. and G. Lasher, "Stimulated Emission of Radiation From GaAs p-n Junctions", *Appl. Phys. Lett.*, vol. 1, no. 3, pp. 62-64, November 1962.

[55] T. M. Quist, R. H. Rediker, R. J. Keyes, W. E. Krag, B. Lax, A. L. McWhorter and H. J. Zeiger, "Semiconductor Maser of GaAs", *Appl. Phys. Lett.*, vol. 1, no. 4, pp. 91-92, December 1962.

[56] H. Kroemer, "A Proposed Class of Heterojunction Injection Lasers", *Proc. IEEE*, vol. 51, pp. 1782-1783, 1963.

[57] I. Hayashi, M. B. Panish, P. W. Foy and S. Sumski, "Junction Lasers Which Operate Continuously at Room Temperature", *Appl. Phys. Lett.*, vol. 17, no. 3, pp. 109-111, 1970.

[58] Z. I. Alferov, V. M. Andreev, D. Z. Garbuzov, Y. V. Zhilyaev, E. P. Morozov, E. I. Portnoi and V. G. Trofim, "Effect of the Heterostructure Parameters on the Laser Threshold Current and the Realization of Continuous Generation at Room Temperature", *Fiz. Tekh. Poluprovodn.*, vol. 4, pp. 1826, 1970 and *Sov. Phys. Semicond.*, vol. 4, pp. 1573, 1971.

[59] J. T. Verdeyen, *Laser Electronics*, Third Edition, Prentice Hall, ISBN 013706666X, 1995.

[60] O. Svelto, *Principles of Lasers*, Third Edition, Plenum Press, ISBN 0306429675, 1989.

[61] A. E. Siegman, *Lasers*, University Science Books, ISBN 0935702113, 1986.

[62] H. Soda, K. Iga, C. Kitahara and Y. Suematsu, "GaInAsP/InP Surface Emitting Injection Lasers", *Jpn. J. Appl. Phys.*, vol. 18, pp. 2329-2330, 1979.

[63] J. L. Jewell, A. Scherer, S. L. McCall, Y. H. Lee, S. Walker, J. P. Harbison and L. T. Florez, "Low-Threshold Electrically Pumped Vertical-Cavity Surface-Emitting Microlasers", *Electronics Lett.*, vol. 25, pp. 1123-1124, 1989.

[64] Ed. G. P. Agrawal, *Semiconductor Lasers: Past, Present and Future*, AIP Press, ISBN 156396211X, 1995.

[65] L. M. F. Chirovsky, W. S. Hobson, J. Lopata, S. P. Hui, G. Giaretta, A. V. Krishnamoorthy, K. W Goossen, G. I. Zydzik and L. A. D'Asaro, "Novel Ion-Implanted VCSEL Structures", Technical Digest of *Spatial Light Modulators and Integrated Optoelectronic Arrays*, OSA, Aspen, CO, pp. 38-39, April 1999.

[66] I. Aeby, "Oxide VCSELs Rise to the Challenge", *Compound Semiconductor Mag.*, http://www.compoundsemiconductor.net/7-5Final/CSJunEmcore.htm, vol. 7, no 5, June 2001.

[67] M. Fuller, "New Research Bolsters Development of 1,300nm VCSELs", *Lightwave*, July 2001.

[68] C. R. Stanley, M. McElhinney, F. Pottier, Y. P. Song, C. D. W. Wilkinson and D. J. Goodwill, "The MBE Growth and MCP/RIE Processing of InGaAs-GaAs MQW Structures for 1047-1064nm S-SEED Arrays", IEEE/LEOS Topical Meeting on *Optoelectronic Material Growth and Processing*, Lake Tahoe, NV, July 11-13, 1994.

[69] W. Franz, *Z. Naturforsch.*, vol. A13, pp. 484, 1958.

[70] L. V. Keldysh, "The Effect of a Strong Electric Field on the Optical Properties of Insulating Crystals", *Zh. Eksp. Teor. Fiz.*, vol. 34, pp. 1138-1141, 1958 and *Sov. Phys.*, vol. 7, pp. 788-790, 1958.

[71] D. A. B. Miller, D. S. Chemla, T. C. Damen, A. C. Gossard, W. Wiegmann, T. H. Wood and C. A. Burrus, "Electric Field Dependence of Optical Absorption

Near the Bandgap of Quantum Well Structures", *Phys. Rev. B*, vol. 32, pp. 1043-1060, 1985.

[72] J. D. Dow and D. Redfield, "Electroabsorption in Semiconductors: The Excitonic Absorption Edge", *Phys. Rev. B*, vol. 1, pp. 3358-3370, 1970.

[73] D. A. B. Miller, D. S. Chemla and S. Schmitt-Rink, "Relation Between Electroabsorption in Bulk Semiconductors and in Quantum Wells: The Quantum-Confined Franz-Keldysh Effect", *Phys. Rev. B*, vol. 33, pp. 6976-6982, 1986.

[74] A. L. Lentine, F. B. McCormick, R. A. Novotny, L. M. F. Chirovsky, L. A. D'Asaro, R. F. Kopf, J. M. Kuo and G. D. Boyd, "A Two kbit Array of Symmetric Self-Electro-Optic Effect Devices", *IEEE Photon. Technol. Lett.*, vol. 2, pp. 51, 1990.

[75] L. Geppert, "Opto-Chips Shatter Records for Bandwidth and Low Voltage", *IEEE Spectrum*, vol. 37, no. 6, pp. 28-29, June 2000.

[76] G. E. Stillman and C. M. Wolfe, "Avalanche Photodiodes", *Semiconductors and Semimetals*, vol. 12, Infrared Detectors II, Academic Press, pp. 291-393, 1977.

[77] J. B. Johnson, "Thermal Agitation of Electricity in Conductors", *Phys. Rev.*, vol. 32, no. 1, pp. 97-109, July 1928.

[78] H. Nyquist, "Thermal Agitation of Electrical Charge in Conductors", *Phys. Rev.*, vol. 32, no. 1, pp. 110-113, July 1928.

[79] S. O. Rice, "Mathematical Analysis of Random Noise", *Bell Syst. Tech. J.*, vol. 23, pp. 282-332, July 1944.

[80] S. O. Rice, "Mathematical Analysis of Random Noise", *Bell Syst. Tech. J.*, vol. 24, pp. 46-156, 1945.

[81] UDT Sensors Inc., *Photodiode Characteristics*, http://www.udt.com/pdf/pd_char.pdf, June 2000.

[82] P. Horowitz and W. Hill, *The Art of Electronics*, Second Edition, Cambridge University Press, Cambridge, ISBN 0521370957, 1980.

[83] A. W. Lohmann and D. P. Paris, "Binary Fraunhofer Holograms, Generated by Computer", *Appl. Opt.*, vol. 6, no. 10, pp. 1739-1748, October 1967.

[84] L. B. Lesem, P. M. Hirsch and J. A. Jordon Jr., "The Kinoform: a New Wavefront Reconstruction Device," *IBM J. Res. Develop.*, vol. 13, pp. 150-155, March 1969.

[85] P. Blair, *Diffractive Optical Elements: Design and Fabrication Issues*, Ph.D. Thesis, Heriot-Watt University, September 1995.

[86] J. W. Goodman, *Introduction to Fourier Optics*, Second Edition, McGraw-Hill, ISBN 0071142576, 1996.

[87] J. L. Sanford, P. F. Greier, K. H. Yang, M. Lu, R. S. Olyha Jr., C. Narayan, J. A. Hoffnagle, P. M. Alt and R. L. Melcher, "A One-Megapixel Reflective Spatial Light Modulator System for Holographic Storage", *IBM J. of Res. and Develop.*, High-Resolution Displays, vol. 42, no. 3/4, pp. 411-427, 1998.

[88] P. Riley, *TeraConnect Demonstrates World's Densest Optical Interconnect*, http://www.teraconnect.com/pr041601.html, October 2001.

[89] J. A. B. Dines, J. F. Snowdon, M. P. Y. Desmulliez, D. B. Barsky, A. V. Shafarenko and C. R. Jesshope, "Optical Interconnectivity in a Scalable Data-Parallel System", *J. of Parallel and Distributed Computing*, vol. 41, pp. 120-130, 1997.

[90] G. A. Russell, J. F. Snowdon, T. Lim, J. Casswell, P. Dew and I. Gourlay, "The Analysis of Multiple Buses in a Highly Connected Optical Interconnect", Technical Digest of *Quantum Electronics and Photonics 15*, IoP Publishing, Glasgow, pp. 75, September 2001.

[91] A. C. Walker, M. P. Y. Desmulliez, M. G. Forbes, S. J. Fancey, G. S. Buller, M. R. Taghizadeh, J. A. B. Dines, C. R. Stanley, G. Pennelli, P. Horan, D. Byrne, J. Hegarty, S. Eitel, K.-H. Gulden, A. Gauthier, P. Benabes, J. L. Gutzwiller and M. Goetz, "Design and Construction of an Optoelectronic Crossbar Containing a Terabit Per Second Free-Space Optical Interconnect", *IEEE J. of Selected Topics in Quantum Electronics*, vol. 5, no. 2, pp. 236-249, March/April 1999.

[92] R. Seifert, *Gigabit Ethernet: Technology and Applications for High-Speed LANs*, Addison Wesley, ISBN 0201185539, 1998.

[93] J. M. Senior, *Optical Fiber Communications: Principles and Practice*, second edition, Prentice Hall, ISBN 0136354262, pp. 622-629, 1992.

[94] http://www.mellesgriot.com/glossary/wordlist/glossarydetails.asp?wID=77

[95] A. C. Walker, M. P. Y. Desmulliez, M. G. Forbes, S. J. Fancey, G. S. Buller, M. R. Taghizadeh, J. A. B. Dines, C. R. Stanley, G. Pennelli, A. R. Boyd, P. Horan, D. Byrne, J. Hegarty, S. Eitel, H.-P. Gauggel, K.-H. Gulden, A. Gauthier, P. Benabes, J. L. Gutzwiller and M. Goetz, "Design and Construction of an Optoelectronic Crossbar Switch Containing a Terabit/s Free-space Optical Interconnect", Invited Paper for the Special Issue on Smart Photonic Components, Interconnects and Processing, *IEEE J. Selected Topics in Quantum Electronics*, vol. 5, no. 2, pp. 236-249, 1999

[96] S. Brown and J. Rose, "Architecture of FPGAs and CPLDs: A Tutorial", Department of Electrical and Computer Engineering, University of Toronto.

[97] S. J. Ben Yoo, Ultra-low Latency, Multi-protocol Optical Routers for the NGI, http://www.darpa.mil/ipto/psum2001/K210-0.html, August 2002.

[98] R. Kunisawa, Gigabit Network with Co-operative Functions for General Purpose Massively Parallel OS, http://www-hiraki.is.s.u-tokyo.ac.jp/ssscore/paper/master97-kunisawa.ps.gz, August 2002.