NASA Contractor Report 187520

ICASE Report No. 91-18

ICASE

DIGITAL OPTICAL COMPUTERS AT THE OPTOELECTRONIC COMPUTING SYSTEMS CENTER

Harry F. Jordan

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

Digital Optical Computers at the Optoelectronic Computing Systems Center

Harry F. Jordan

Center for Optoelectronic Computing Systems, Campus Box 525
University of Colorado, Boulder, CO 80309-0525

ABSTRACT

The Digital Optical Computing Program within the National Science Foundation Engineering Research Center for Optoelectronic Computing Systems has as its specific goal research on optical computing architectures suitable for use at the highest possible speeds. The program can be targeted toward exploiting the time domain because other programs in the Center are pursuing research on parallel optical systems, exploiting optical interconnection and optical devices and materials. Using a general purpose computing architecture as the focus, we are developing design techniques, tools and architectures for operation at the speed of light limit. Experimental work is being done with the somewhat low speed components currently available but with architectures which will scale up in speed as faster devices are developed. The design algorithms and tools developed for a general purpose, stored program computer are being applied to other systems such as optically controlled optical communications networks.

INTRODUCTION

The Digital Optical Computing Program within the Optoelectronic Computing Systems Center at the University of Colorado at Boulder is centered around the design and construction of an "all-optical," general purpose, stored program, digital computer. It is "all-optical" in the sense that logic level components have only optical inputs and outputs, with all inter-component signals restricted to light. It is digital because this type of operation has proven successful in representing both arbitrary precision numbers and control information. The computer science term, stored program, means that instructions are stored as data to be manipulated by the computer itself. Thus it can "write its own programs" by, for example, running compilers. Finally, "general purpose" implies an instruction set which supports both the symbolic processing needed to manipulate programs as well as numeric computation. The design is bit serial to minimize the number of active optical devices. Fiber delay lines are used for storage because they are passive elements, suited for storing serial information. Waveguide switches using the electro-optic effect are used to do logic. The bit serial design uses bandwidth, or time domain capacity, to achieve processing power. Since terabits per second are possible in one optical channel, much complexity can be put into the time domain, making possible prototypes with few components. To minimize active elements, we have adopted a simple but complete instruction set without floating point arithmetic. Instructions have one address with no complex addressing modes. A carefully optimized design gives a complete computer using only a few tens of switches. Optical fibers form all memory and interconnecting components. There are no synchronizing elements such as flip flops, so all signal storage is in passive fiber delays. More important than demonstrating an optical computer is gaining more understanding of the use of the time domain in computer architecture and of time-space trade offs. Another goal is to transfer digital electronics knowledge to optics. There may be new ways to use optics which have no analogs in electronics, but it would be unwise to assume either a complete break with the extensive knowledge base in digital computing or to assume that it all transfers unchanged to optics.

Prior work in optics which applies most directly to the current work is in communications and signal processing. Single and multi-mode optical fiber and connector systems have been developed and commercialized[1]. Static directional couplers are available with specified power splitting ratios and can be used for fan-out or for combining noninterfering signals. Electrically switched directional couplers[2] have reached a reasonable degree of maturity, and are available from more than one source. They are used for modulation, multiplexing and demultiplexing[3] of optical communications signals. To get a component with all inputs and outputs optical, we add an a photodetector, amplifier and electrode driver to allow the switch to be optically controlled. The above devices are used as shown in Fig. 1 to provide an implementation domain for digital optical computing. It uses intensity encoding of bits and synchronous operation. A light pulse at a clock time is a logic one, and no light represents a zero. The waveguide switch computes the multiplexer function shown. Interconnection is done with single mode fiber and fanout by 3dB fiber couplers, which are also used to merge signals from two sources when only one source carries a signal at a time. Memory is accomplished entirely by the propagation time of optical pulses in fiber. The delay schematic shown in the figure represents a coil of fiber of delay $K \Delta$.

2

$$D = AC + B\bar{C}$$
$$E = BC + A\bar{C}$$

*LiNbO*₃ Waveguide Switch

as fanout   3 dB Coupler   as wired OR
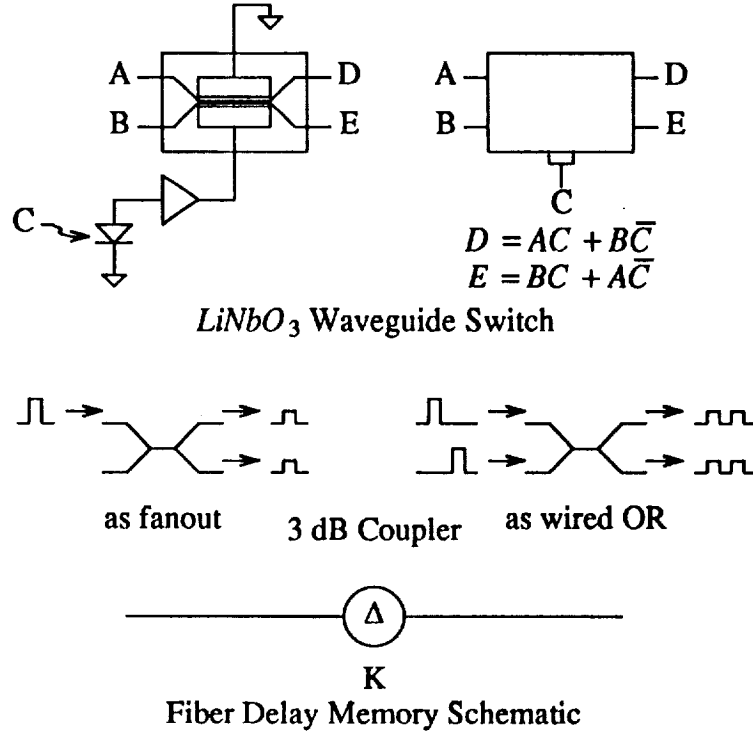
K

Fiber Delay Memory Schematic

Figure 1: Fiber and Waveguide Switch Implementation Domain.

For our purposes, an electro-optically switched directional coupler can be viewed as a controlled exchange element with two optical waveguide inputs, the signals on which can be copied directly or exchanged onto two optical waveguide outputs. The direct, "bar," state or exchange, "cross," state are under the control of an electrical potential. Its physical properties can be summarized at the systems level by loss and crosstalk from inputs to outputs in both states. Loss can be kept under 5dB and crosstalk less than -20 dB. Optical fiber has an index of refraction of about 1.5, which implies a distance-time correspondence of 20 cm/nanosecond. At a wavelength of 1300 nanometers, losses of a few tenths of a dB/km are achievable with low chromatic dispersion. Standard connector technology yields 1/2 dB or less loss per connection. See, for example, Cherin[4].

The photodetector and electronics encapsulated in the logic component limit its speed, so the impact of this limit on our work must be assessed. Our emphasis is on architecture, so the question is whether useful concepts and techniques can be developed in spite of the limitation. Logic is done entirely by the waveguide switches, and any other logically complete optical component can be used with little impact on the system. As far as interconnections and delays are concerned, the clock frequency can be increased simply by scaling down all fiber lengths. This is the essence of the concept of a *speed scalable* architecture. At the architectural level, all physical time constants other than the speed of light are encapsulated in the switching component. The speed of a system with a *speed scalable* architecture can be increased by replacing the logic element with one *m* times as fast and scaling down all fiber lengths by a factor of *m*. The architecture remains unchanged.

3

To understand the potential value of speed scalable architectures, one can extrapolate system speeds for devices which are still in the research stage. The time domain capacity of optical data transmission is important because transmission is becoming more of a limit to computer speed than switching. It is physically possible to produce and propagate 10 femtosecond optical pulses, which translates to a bandwidth of 100 terabits/second. Haner[5] has actually demonstrated 100 femtosecond resolution in a time compressed waveform, which promises that 10 terabits/second may actually be achievable. A fast, logically complete optical switch has been demonstrated by Islam[6] who built NOR gates using 300 femtosecond solitons. His gates show that optical switching and transmission may attain similar speeds. A smaller, but significant, speed improvement is expected from integrated electro-optic switches, waveguides, detectors and electronics in a III-V materials system[7].

Using such bandwidths requires a speed scalable architecture. The architectural drivers implied by speed scalability are:

all inter-component signals are restricted to light;
there are no synchronizing memory elements;
synchronization is done by controlling optical delays;
optical signal quality must be restored, both in amplitude and timing;
any logically complete device can substitute for the switches.

Although a general purpose digital computer focuses the work, it is not expected to be the first competitive success of optical architectures. A near term application is in optically switched optical communication networks. Packet switching requires only a small amount of logic. Even simple optical state machines can improve high speed communications systems, which now require conversion between optics and electronics for the simplest switching. High speed controllers in hostile environments, such as particle accelerators, are also a potential application. In general purpose computing, optics will complement electronics long before it replaces it. Time (or frequency) multiplexing can make high speed serial systems effective adjuncts to slower, parallel, electronic ones.

BUILDING BLOCKS

A digital computing architecture must include logic, interconnection, signal restoration and memory. The ability to restore signals in both amplitude and timing is not strictly a logical function, since it depends on the physical characteristics of signals. It
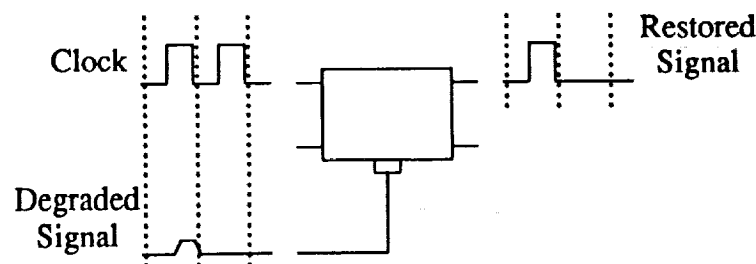

Figure 2: Signal Restoration in Amplitude and Timing.

4

can be accomplished with a switch component by gating the system clock as shown in Fig. 2. Amplitude is restored because the incoming optical pulse is physically switched to the output. For timing restoration to be effective, the control signal must arrive earlier than the clock and then be amplified and broadened in order to switch the full, correctly timed clock pulse to the output, while the second output receives a restored complemented copy of the control signal. Clock gating was used in electronic computers to restore timing. Here it also restores the optical power level. This makes supplying optical power a problem of producing multiple copies of a synchronized clock.

The multiplexer function, $D = AC + B\bar{C}$, shown for the switch of Fig. 1, is logically complete given the constants zero and one. In the pulse coded representation, zero is the absence of light, and one is a copy of the system clock. A circuit with both logic and memory is the serial binary adder which will add two binary numbers presented to its inputs low order bit first. It consists of a single full adder and a one clock period delay to store the carry used in computing the next higher order sum bit. Figure 3 shows the circuit built from waveguide switches, 3 dB couplers for fanout, and a fiber delay for the carry. The switches connected to the inputs not only complement them but also do signal restoration. Switches S3 and S4 demonstrate AND, OR and exclusive OR functions.
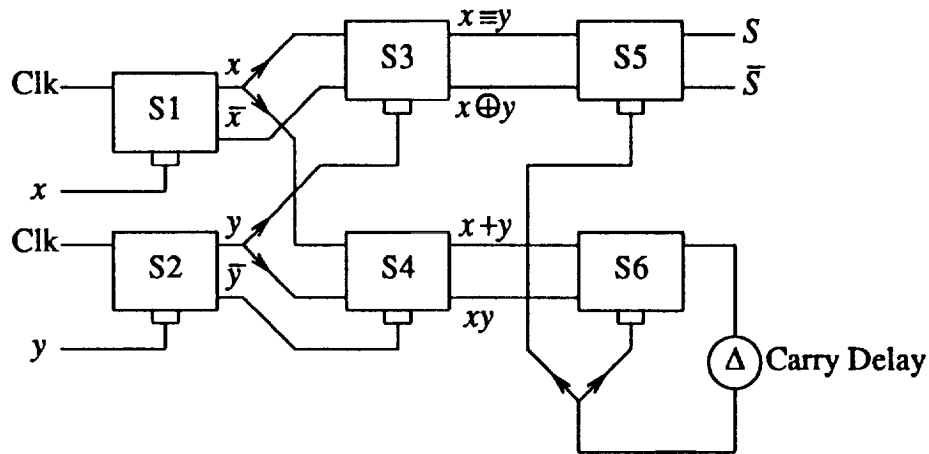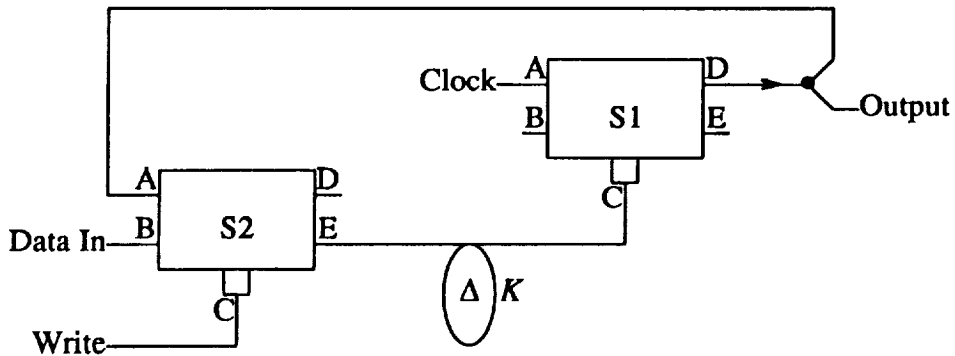

Figure 3: Serial Binary Adder with Carry Delay

Figure 4: Fiber Delay Line Memory Loop.

5

The circuit is independent of the length N of the binary numbers, but end conditions, such as initializing the carry delay to a zero or discarding the high order carry, require more switches and timing signals to mark word boundaries.

Memory registers extend the carry delay of the binary adder with signal regeneration, read and write access to the register. Figure 4 shows a K bit register. Switch S1 regenerates data on every circulation through the loop, or once every K clock periods. With switch S2 in the cross state, a one bit emerging from the K bit storage loop causes switch S1 to copy a clock pulse into the loop. Zeros route the unconnected input B of S1 to the D output. The 3 dB coupler allows the register to be read and switch S2 provides the ability to write new information by holding the Write input at logic one for K bit periods. When such a delay loop is used as a register in a serial machine, its length is usually equal to the computer word length, and its contents can be read or written once per word time.

Multiple word memories use the same kind of storage loop shown in Fig. 4. With K bits per word, the length an N word memory loop is NK bits. A scale of N counter incremented once per word determines which word is currently passing switch S2, where it can be read or written. This counter requires $m = \lceil \log_2 N \rceil$ bits. If $m < K$, the m bit counter can be incremented during the passage of one word. To access a word, its address is compared to the counter value at each word time until a match occurs. A large memory can have several loops and an address of two parts, one of which selects a loop, while the other is compared to a counter. The number and size of loops is determined by the acceptable waiting time for an address match and by the physical limits on the loop capacity. Sarrazin[8] has examined the several physical limitations on memory loop capacity to establish a resolution of one part in 10^4 per degree C for a synchronous storage loop.

The serial counter design will be referred to several times. Figure 5 is a block diagram of a four bit, scale of 16 serial counter. On the left is a four bit increment signal consisting of a one in the low order bit position and three zeros. Below the half adder is a stored four bit count value, with low order bit at the left. Above the half adder is a carry bit. It and the count are stored in delay lines of one and four bit durations, respectively. Use of an m bit delay line and placing the increment signal at the low order bit of an m bit period are the only changes required to make it an m bit, scale of 2^m counter. Figure 6(a) shows a logic description of the counter and its optical design is shown in
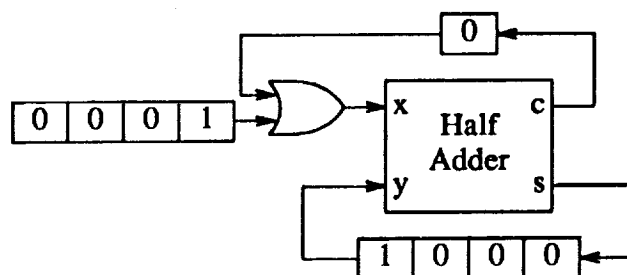

Figure 5: Block Diagram of a Bit Serial Counter.

(a) Bit-Serial Counter Logic
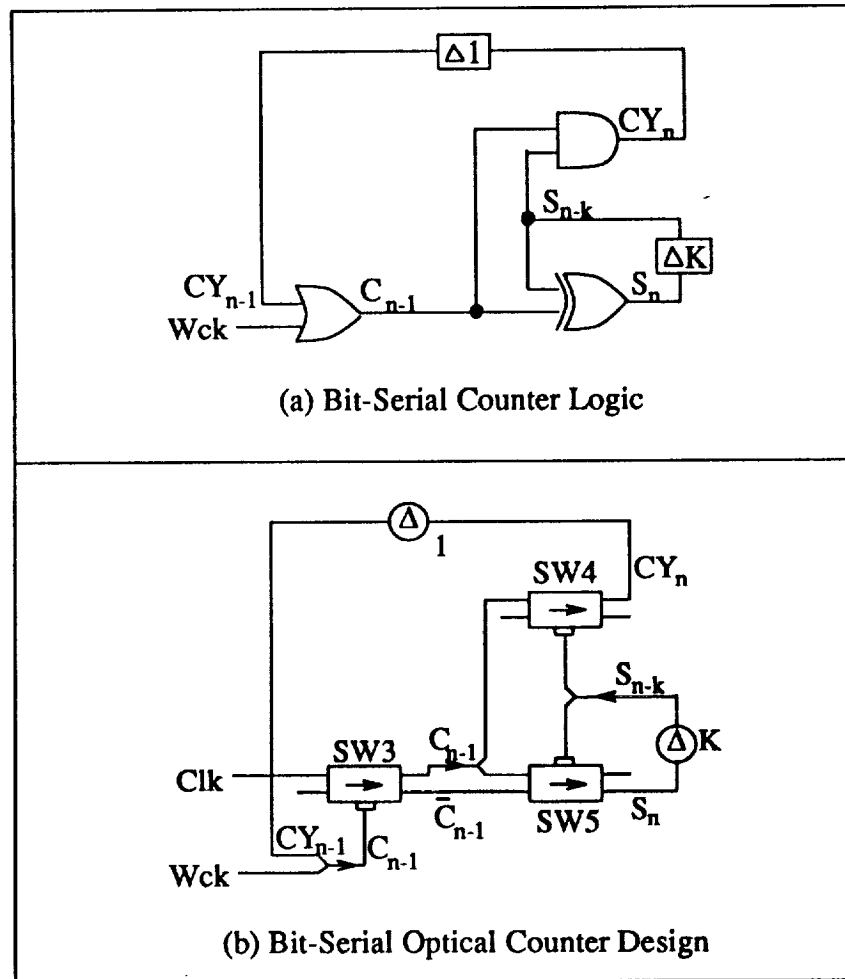
(b) Bit-Serial Optical Counter Design

Figure 6: Implementation of the Bit Serial Counter.

Fig. 6(b). The signal labeled Clk is an optical oscillator with pulses appearing once every bit time. The signal Wck times a word of the binary count by producing a pulse every four bit times. Including two switches required to derive Wck from Clk, the complete design requires only five switches, making it a simple implementation target.

The design of Fig. 6(b) neglects a fundamental problem of speed scalable design. Delays in the circuit are taken to be zero except for those associated with explicit delay elements. Delays are actually distributed throughout the circuit, in connections, switches and electrode drivers. The delay distribution problem is to distribute delays to coordinate signal arrival times at the inputs to a switch. It is a result of interconnection delay being of the same order as logic delay and the absence of flip flop synchronization. Lumped delay designs using familiar digital design techniques must be transformed into realistic ones with delays which meet the physical requirements of components and layout. Otherwise, signals to be logically combined will not arrive simultaneously at the proper logic element.

Figure 7 shows a two switch circuit to derive the word clock, Wck, for the counter of Fig. 6 from the master clock, Clk. Part (a) shows the lumped delay design, which produces one pulse out for every four in, and part (b) shows the design with a delay associated with each signal path. Also shown are two equations which ensure that corresponding inputs arrive simultaneously at switches S1 and S2, respectively, and inequalities which characterize the minimum delays in the paths between outputs and inputs. The delays, δ_A, δ_B, δ_C, δ_D and δ_E are associated with the five terminals of a waveguide switch, while δ_S is associated with the fixed 3dB coupler used as a signal splitter. By adding length to the interconnections and adjusting the lumped delays, the equations and constraints can be satisfied, provided no feedback loop has an inherent physical delay longer than the specified lumped delay. Since the minimum lumped delay in a feedback loop for a non-trivial sequential circuit is usually one clock period, the "optical length" or latency of a switching element puts a lower limit on the clock cycle time. The latency is not necessarily related to the bandwidth of an element. The topic of time multiplexed architectures, which make use of high bandwidth logic in spite of long latency, will be discussed later.

SYSTEM ARCHITECTURES

At this point essential building blocks for a stored program computer — logic, registers and multi-word memory — have been discussed. The experimental question is whether a complete, stored program computer can be built with few enough switches to be feasible as a near term prototype. The current cost and size of available waveguide switches imply that a computer requiring hundreds to thousands of them would remain only a paper design yielding no practical experience with speed scalable architectures. The SCAMP[9] architecture is a carefully minimized design containing all of the features

(a) Lumped Delay Design

$$\lambda1 + \lambda2 + \Delta = \lambda1 + \lambda3 + \lambda5$$

$$\lambda1 + 2\Delta = \lambda1 + \lambda3 + \lambda4$$

$$\lambda1 \geq \delta_A, \lambda2 \geq \delta_B + \delta_D, \lambda3 \geq \delta_E + \delta_S$$

$$\lambda4 \geq \delta_S + \delta_C, \lambda5 \geq \delta_S + \delta_C$$
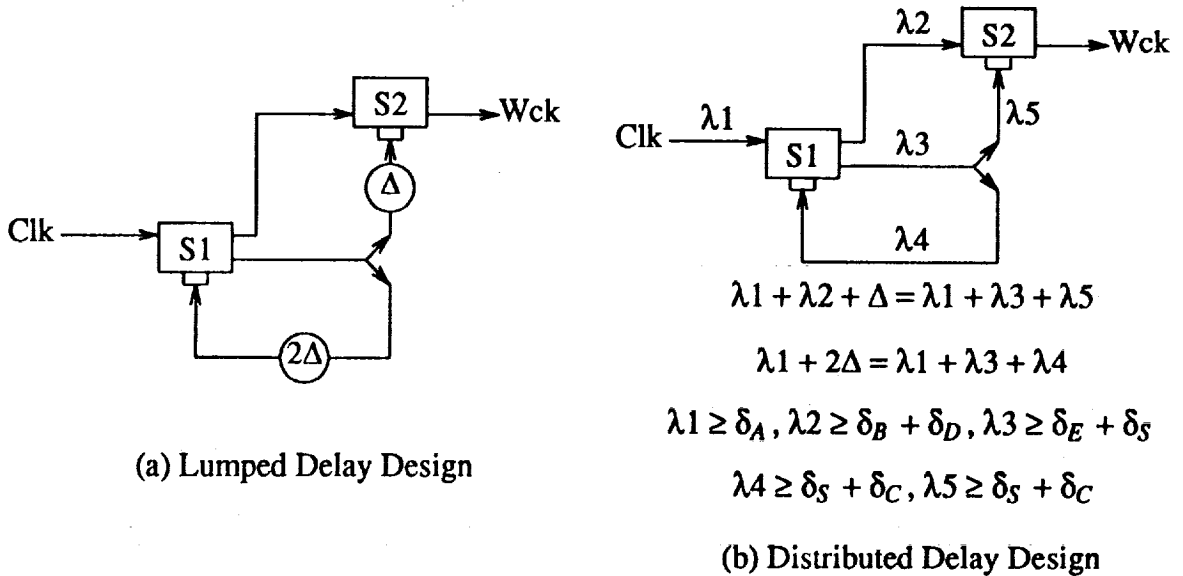
(b) Distributed Delay Design

Figure 7: The Delay Distribution Problem.

of a general purpose computer except input/output. I/O will initially be supplied by the monitor subsystem[10] necessary to control and make measurements on the computer. For minimality, general registers are represented by a single accumulator. It, the program counter, the instruction register and the memory counter are the only registers accessible every word time.

Along with minimizing registers, the instruction set is also kept small. Multiply, divide and floating point arithmetic are left for software, although preliminary work on multiply[11] and divide[12] hardware is in progress. The arithmetic logic unit (ALU) is limited to *and, or, not, add* and *shift*. This has sufficed for many microprocessors and is a reasonable first step. The design proceeded in two phases. First, logic, registers and memory were assembled under the assumption that the waveguide switches implemented perfect multiplexers. The complete design required only about 50 waveguide switches. The second phase used measured loss and crosstalk values to determine where to place signal restorers to meet the physical specifications of the switches and photodetectors. This second phase resulted in a design using about 75 switches. Although the design uses a 16 bit word length, only delay line lengths change to accommodate any word length which is no shorter than the memory address length plus six bits. Simulations verified the design at both the logic level and the physical level.

The soliton gates[6] cited as a demonstration of very high speed logic used 20 meters of fiber to obtain sufficient interaction length given the weak nonlinearity of glass. Such extreme ratios of reciprocal latency to bandwidth are not expected in mature optical devices, but interaction lengths of a few centimeters in terahertz bandwidth gates would not be surprising because of the large power densities which would otherwise be required. Although long latency limits the minimum feedback loop length, such gates would have the potential for hundred-fold time multiplexing or pipelining. Decoupling the duration of a switched pulse from the latency of the switching element by this means opens up important possibilities for optical logic devices.

An architectural technique to make use of devices with high bandwidth but long latency decouples consecutive bits by time multiplexing serial data streams. One can multiplex several bit streams on the hardware of the SCAMP to yield several independent computers. Such a time multiplexed multiprocessor requires multiplexing of processor inputs and demultiplexing of outputs, as shown in Fig. 8. Since multiplexing and demultiplexing do not require feedback, they can be implemented with long latency
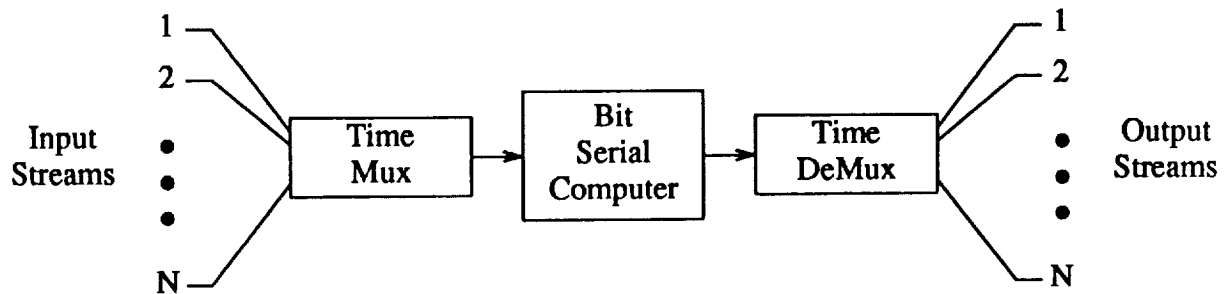

Figure 8: Time Multiplexed Multiprocessor

devices. Time multiplexed multiprocessors have been built with electronics. An early commercial one implemented the ten peripheral processors of the CDC6600[13], and a more recent pipelined multiprocessor, the Denelcor HEP[14], multiplexed up to 128 instruction streams on one set of processor hardware. Pipelined vector units in current supercomputers use time multiplexing of independent vector components to achieve high speed. Latency tolerance is incorporated at the level of numeric operations in arithmetic pipelines and systolic arrays, but only in the highest speed designs is it a gate level concern of the sort addressed by the delay distribution problem. More research is needed on trade-offs and optimizations possible in designing systems pipelined at the gate level, especially if such designs use no latches. The elimination of latches is not intrinsically desirable, but since latching implies a device entering a stable state, and since time constants associated with stable states are long compared to those of unstable states, the highest speed designs may well avoid latches.

The immediate future of high speed optical logic is probably in communications rather than general purpose computing. Packet switched communication networks, controlled by information contained in the data being transmitted, have great utility. Since information in high speed fiber networks arrives and departs in optical form, optically controlled optical switching would benefit such networks. Since network control logic can be simple, and the network is extended in space, existing expensive and large waveguide switches are not as severe a limitation as in general purpose computing. A specific architecture for a high speed, packet switched, optical communications network is being studied[15]. It is based on three interacting ideas: 1) a network of $N \log N$ nodes of fixed fan-in and fan-out in which a message needs to pass through only order $\log N$ intermediate nodes to reach its destination; 2) "hot potato" routing, in which messages are not stored in intermediate nodes; and 3) optical compression of data packets to release bandwidth for use in network synchronization and control. Nodes in the network not only do switching but are associated with hosts which originate and consume messages. A ShuffleNet[16] network of $N \log_2 N$ nodes with two inputs and two outputs per node and a maximum distance of $2\log_2 N - 1$ between any two nodes will be used. When two incoming messages need to use the same output port, electronic switching nodes typically store one of the conflicting messages for later transmission. Rather than do electro-optic conversion for storage, the network uses "hot potato" or deflection routing to send one of the messages through the wrong output port to make its way by another path to its destination. Finally, conflicts are minimized and synchronization is simplified if message packets are separated in time by large gaps. Time compression of the optical data by wavelength multiplexing or grating techniques can create such gaps, thus trading potential data bandwidth for ease of network control.

Another important architecture uses fiber delays and exchange switches for time slot interchange. This time domain permutation is useful both in accessing information from a serial delay line in an order different from that in which it is stored and to allow the time multiplexed independent processors of Fig. 8 to exchange information. Time is divided into slots containing information, with a frame consisting of N sequential time slots. Time slot interchange means moving information from slots of an input signal into slots in different relative positions of the output frame. such a permutation is associated with a frame delay. A time slot interchange architecture has immediate application in time multiplexed telecommunications channels. Time multiplexed signals are most often

switched by demultiplexing into separated channels, switching in the space domain, and re-multiplexing the result. Thompson's[17] architecture uses waveguide switches to demultiplex an input stream into individual time slots, uses fiber loops to individually delay them, and uses more switches to multiplex them into the output stream. Leaving out switches needed to vary the delays, $2N - 2$ switches are used in the multiplexer and demultiplexer.

Ramanan[18] applied techniques developed for multistage switching networks in the space domain to time domain permutation. The basic building block of the architecture uses a switch connected to a delay loop of size Δ in a feedback configuration to selectively interchange pairs of time slots separated by a fixed time Δ, a multiple of the slot time. Figure 9 shows the situation for a Δ of one slot time. Any number of pairs can be interchanged by setting the control for exchange (x) for all time slots except the second of a pair to be exchanged, for which it is set for straight connection (=). The Benes[19] network, with $2N\log_2 N - 1$ exchange switches, is a universal space domain switch. Ramanan's time domain analog of this network can perform any time slot permutation on a frame of $N = 2^k$ slots with only $2\log_2 N - 1$ of the above building blocks.

One block with delay loop of length $N/2$ can selectively exchange any pair of slots separated by $N/2$ units. The frame suffers an overall delay of $N/2$ slot times. If we now use an $N/2$ exchange switch at both input and output of a universal interchanger for frames of length $N/2$, as shown in Fig. 10, we have a recursive construction for a universal interchanger of length N. The input stage allows time slots to be selectively exchanged between first and last half frames, the center section permutes each half frame arbitrarily, and the output stage again allows selective exchange of pairs between half frames. This is sufficient to apply the Benes looping algorithm[20] to show that if the center can permute frames of $N/2$ slots, the whole network can permute frames of length N. If $N = 2^k$ is a power of two, continuing the recursion until a one block exchanger for adjacent slots is left in the center yields a general time slot interchanger with $2\log_2 N - 1$ switches and delay loops, as shown in Fig. 11. An alternative design in which the delays increase toward the center as powers of two is also possible but more difficult to

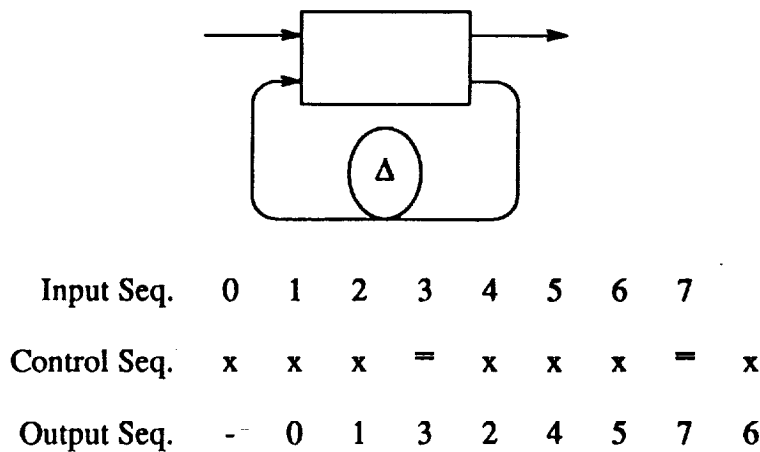| Input Seq.	0	1	2	3	4	5	6	7	
Control Seq.	x	x	x	=	x	x	x	=	x
Output Seq.	-	0	1	3	2	4	5	7	6

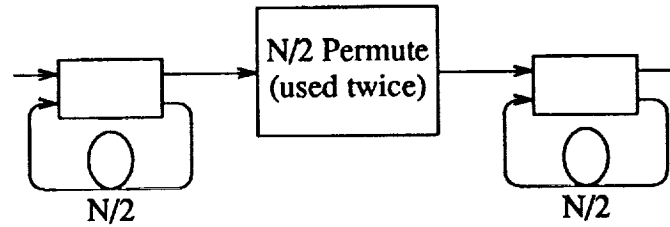Figure 9: Exchange of Time Slot Pairs

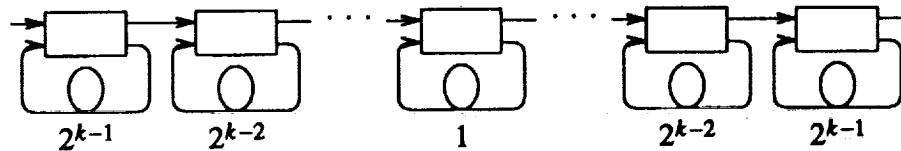Figure 10: Recursive Construction of a General Time Slot Interchanger.

Figure 11: A Time Slot Interchanger with $2\log_2 N - 1$ Switches.

describe. Thompson's design requires $2N - 2$ switches for the demultiplexer and multiplexer alone. For permuting 1024 time slots, the new design requires 19 switches compared with more than 2046 for the other architecture. This new architecture shows how optics can give insight into time-space tradeoffs which may even have advantages for electronic implementation. Since time slot interchange forms a large fraction of all telecommunications switching, the practical value of the result may be large.

TOOLS AND TECHNIQUES

Simulation is an important tool in realizing computer architectures, which by nature involve high complexity. The SCAMP design uses many clever tricks to reduce the number of switches. Since clever tricks can backfire, a logic level verification of the design is the first important step. The tool built to do this is an event driven simulator called HATCH[21]. As a result of the absence of flip flops in the design, it is a continuous time simulator. Clocked timing is introduced, as in the actual system, by an object called a clock which produces a standard repetitive signal. The HATCH software is object oriented so that it can evolve to meet new simulation needs by the addition of new object types and methods.

The first evolutionary challenge met by HATCH was to solve the delay distribution problem, described in connection with Fig. 7. The circuit is taken as a graph whose nodes are waveguide switches or 3 dB couplers. The edges of the graph represent interconnections between elements. A delay vector, with one component for each edge, characterizes the delays in the design. For the lumped delay design, many of these components are zero. In a real design, each delay vector component is always greater than some minimum which represents the path length through components, length of couplers, length of the interconnecting fiber, and for some edges, the latency of photodetector and electrode driver circuitry. The physical constraints are thus embodied in a minimum

12

delay vector over the edges of the graph. The linear equations ensuring synchronized signal arrival are derived from the lumped delay design. A delay vector having each component greater than or equal to that of the minimum vector and satisfying the linear system is a possible design, and that having the least extra delay is the solution. Three algorithms for solving this constrained minimization problem were studied and compared: the simplex method[22], the shortest path method[23], and the local distribution method[24]. The study[24] showed that the local distribution method converged well, with delays increasing monotonically up from the lumped delay values to those of the solution vector. The simplicity of this algorithm gave it better performance than the other two, so it was therefore included in HATCH to do delay distribution.

Signal quality management is also included in HATCH. Power losses can make ones appear to be zeros while crosstalk in the switches may cause zeros to accumulate noise and appear to be ones. At each switch control terminal, a threshold decision distinguishes zeros from ones. A signal restorer must be placed in any optical path from a standard clock which has enough loss for a logical one to be below threshold or enough crosstalk for a zero to be above threshold. If loss and crosstalk specifications are associated with each device, HATCH[25] can compute signal degradation associated with a specified path or identify the worst case path. A designer can thus use it to add restoring switches to a design which assumes ideal elements. This was done for the SCAMP design assuming a loss of -5 dB, a crosstalk of less than -20 dB and a control terminal photodetector threshold of -19 dBm, obtaining a signal restored design for SCAMP requiring only 75 switches.

The extended HATCH is a general tool to design fiber optic and waveguide switch based systems. Starting with a lumped delay design, logic simulation with ideal gates verifies the sequential behavior. Component delays then allow HATCH produce fiber lengths for a distributed delay design. When loss and crosstalk specifications are added, HATCH identifies critical paths for insertion of signal restoring switches. The final design is then simulated with delay, loss and crosstalk specifications to produce logarithmically scaled plots of signal amplitudes versus time under worst case loss and crosstalk assumptions. An overview of the functionality of HATCH is illustrated in Fig. 12.

It has been mentioned that techniques for gate level time multiplexing can help overcome the effects of latency. A specific example is the serial counter design. The shortest feedback loop in the counter of Fig. 6 has a length of one bit time. Since it passes through two switches and a 3 dB coupler, it sets a lower limit on the bit rate of the counter. Time multiplexing can increase the effective counter bandwidth by multiplexing more than one independent bit stream on one set of counter hardware. This gives the effect of several simultaneous counters, each running at the original bit rate. A block diagram for this scheme with two multiplexed counts appears in Fig. 13. The counter associated with the bits in the white boxes is about to be incremented from 3 to 4 while that associated with the stippled boxes is about to change from 8 to 9. A carry feedback generated from a bit at time t need not combine with a bit arriving at the increment input any sooner than two bit times later. The two Wck input streams can be multiplexed using only differential delay and a 3 dB coupler, and the count outputs can be demultiplexed by one switch toggling at the effective bit rate.
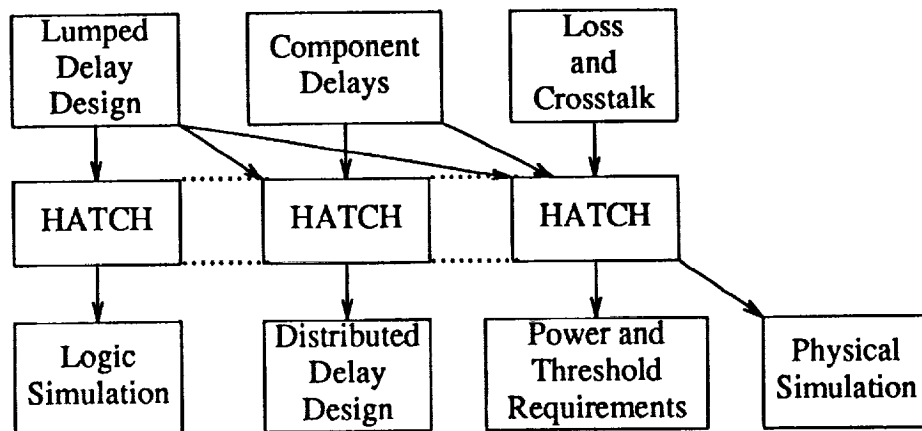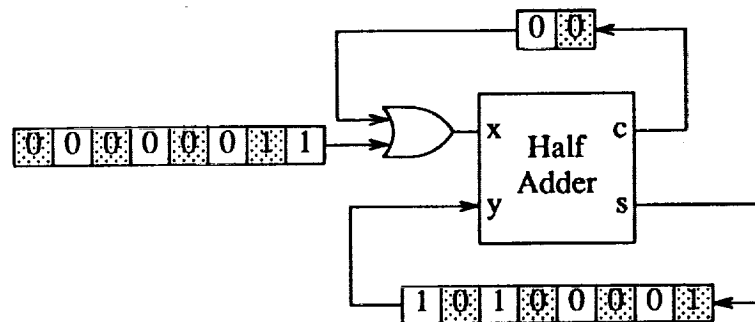
Figure 12: The HATCH Design Support System.

Figure 13: Two Independent Counters Multiplexed on the Same Hardware

EXPERIMENTS

The demonstration of a prototype optical computer involves several intermediate experiments. From the architecture viewpoint, a simple feedback state machine is the first step. We chose the one out of four scaler of Fig. 7 driving the counter of Fig. 6. The count value delay line demonstrates one word storage, so the second step is the multi-word memory loop, which also requires a binary counter and a serial comparator. The memory and one word registers will hold operands and result for an arithmetic unit, which will be the third subunit built. The instruction fetch, decode and execute cycle will be implemented last. The current status of experiments is between the counter and memory demonstrations.

The scaler of Fig. 7 is a feedback state machine, but is simpler than the counter because it is self stabilizing if a bit is lost or gained. An infrequent bit error in the scaler would go undetected on an oscilloscope. The counter, on the other hand, has a period of 64 bits, and single bit errors have a large influence on its output. The optical scaler and binary counter combination has been built and tested[26] yielding the output waveform for a 50 MHz clock rate shown in Fig. 14. The complemented count available at the unused output of SW5 in Fig. 6 is shown, low order bit first reading left to right. Changing two fiber lengths yields a six bit, scale of 64, counter, and this device was also built
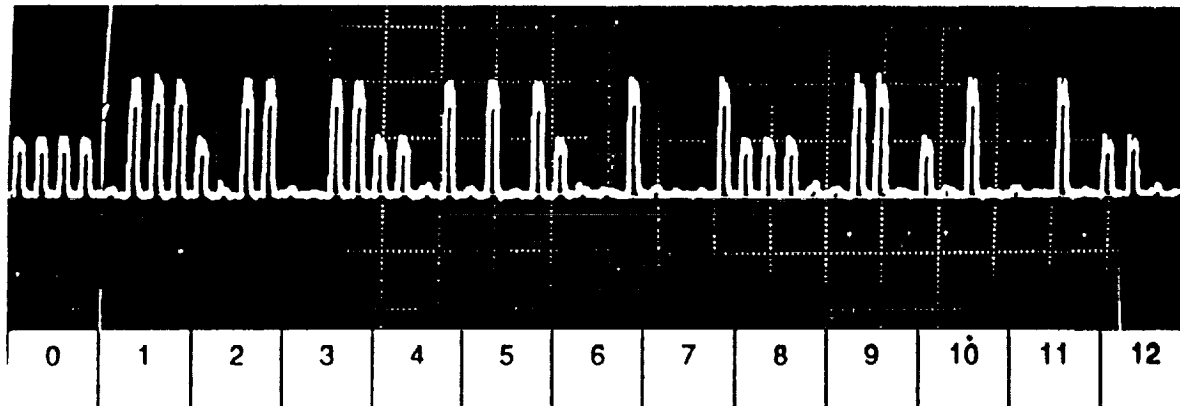
14

Figure 14: Output of the Four Bit Optical Counter

and operated at a 50 MHz clock rate. A modified design with a shortened carry feedback loop was built and operated at 100 MHz. The technique of using time multiplexing to increase the effective counter bandwidth was also applied to obtain 100 MHz operation by interleaving two independent count values, independently incremented by interleaved Wck signals. The dual counter was also operated successfully at an effective 100 MHz rate.

CONCLUSIONS

The work discussed here primarily exploits the time domain to make potential use of high optical bandwidth,although the packet switched communications network also includes significant spatial parallelism. If this work does not directly address the use of spatial parallelism in optics, it also does not conflict with it. The ideas of speed scalable architectures should ideally be combined with the parallel optical designs being pursued effectively by other groups [27] [28] using synchronous operation and latching gates. The most parallel system running at the highest possible speed is the ideal optical computer, although the time slot interchanger shows that at least some interesting systems are strictly serial.

This work also does not directly address the problem of producing or using an ideal optical device, which is fast, small, can be highly integrated, and uses little power. These architectures would smoothly scale up in speed with the availability of such a device, but our work has no device development component as such. The size, speed and cost of the $LiNbO_3$ waveguide switches is such that the specific implementation of the prototype architecture discussed here would not be competitive as a general purpose computer, although it could have special purpose application as a very high speed controller in systems where data is optical to begin with.

Optical computing helps in understanding the architectural problems associated with very high speed digital computing. Electromagnetic radiation and induction effects are avoided, and experimental demonstrations of both communications and switching at terabit per second bandwidths exist. Current digital architectures are heavily influenced

15

by the assumptions of arbitrary fanout and instantaneous signal propagation within moderately complex subsystems. As switching speeds become faster and power more of a concern, both assumptions prevent architectures from scaling up in speed. This work involves latch-free designs in which finite signal propagation time is fundamental. Such speed scalable designs can take advantage of higher speed devices as they become available. Tools such as the delay distribution algorithms are essential to this style of design. Optics provides an excellent environment in which to study speed of light limited architectures, which are becoming of increasing concern in electronic computer design also.

The systems described here are not general purpose supercomputers. The results show that designing an optical computer involves much more than simply inserting an "optical transistor" into an existing design. The maturity and commercial development of digital electronics suggests that an all-optical computer is not imminent. Optics will probably find its way gradually into digital computers, starting from the fibers already used to connect cabinets in large, high speed systems. Although optical architectures may well be different from electronic ones in important respects, they will probably build on the digital design knowledge base on which electronic computers rest. Optical computers will eventually combine spatial parallelism with high speed design constrained by the speed of light limit, as will future electronic computers. In the meantime, a better understanding of speed of light limited digital systems shows great promise for immediate applications. Communications systems can benefit from even limited optical processing. Time critical tasks in signal processing are another area in which significant applications may exist. Perhaps even more important is the fact that the speed of light limit is a universal phenomenon, not just an optical one. By studying the time-space tradeoffs in the optical domain, insight may be gained into the fundamental nature of physical realizations of the mathematical model which constitutes computation.

REFERENCES

[1] C.K. Kao, *Optical Fiber Systems: Technology, Design and Applications*, McGraw-Hill (1986)

[2] S.K. Korotky and R.C. Alferness, "Waveguide Electro-optic Devices for Optical Fiber Communication," in *Optical Fiber Telecommunications*, I.P. Kaminow and S.E. Miller, Eds., Academic Press (1988)

[3] R.S. Tucker, S.K. Korotky, G. Eisenstein, U. Koren, G. Raybon, J.J. Veselka, L.L. Buhl, B.L. Kasper and R.C. Alferness, "4 Gb/s optical time-division multiplexed system experiments using Ti:LiNbO$_3$ switch/modulators," *Tech. Digest Top. Meet. on Photonic Switching*, Paper FD3, Incline Village, NV (1987)

[4] A. H. Cherin, *An Introduction to Optical Fibers*, McGraw-Hill, New York (1983)

[5] M. Haner and W.S. Warren, "Generation of arbitrarily shaped picosecond optical pulses using an integrated electrooptic waveguide modulator," *Applied Optics*, V. 26, No. 17, pp. 3687-3694 (1987)

[6] M.N. Islam, C.E. Soccolich and D.A.B. Miller, "Low energy ultrafast fiber soliton logic gates," submitted to *Optics Letters*, (Jan. 1990)

[7]   O. Wada, "Optoelectronic integration based on GaAs material, a tutorial review," *Optical and Quantum Electronics,* V. 20, p. 441 (1988)

[8]   D.B. Sarrazin, H.F. Jordan and V.P. Heuring, "Fiber optic delay line memory," *Applied Optics,* V. 29, No. 5, pp. 627-637 (1990)

[9]   V.P. Heuring, H.F. Jordan and J.P. Pratt, "A Bit Serial Architecture for Optical Computing," *OCS Tech. Rept. 88-01,* Center for Optoelectronic Computing Systems, U. of Colo., Boulder, CO (1988)

[10] E.T. Lode and V.P. Heuring, "A Monitor for the Bit Serial Optical Computer," *OCS Tech. Rept. 90-17,* Center for Optoelectronic Computing Systems, U. of Colo., Boulder, CO (1990)

[11] A.R. Pleszkun and M.P. Dickman, "Design Alternatives for Optical Bit-Serial Multiplication," *OCS Tech. Rept. 89-37,* Center for Optoelectronic Computing Systems, U. of Colo., Boulder, CO (1989)

[12] M.P. Dickman and A.R. Pleszkun, "A Bit-Serial Optical Divider", *OCS Tech. Rept. 90-22,* Center for Optoelectronic Computing Systems, U. of Colo., Boulder, CO (1990)

[13] J. E. Thornton, *Design of a Computer: The Control Data 6600,* Scott, Foresman and Co., Glenview, Ill. (1970)

[14] J. S. Kowalik, Ed., *Parallel MIMD Computation: The HEP Supercomputer and its Applications,* MIT Press (1985)

[15] J.R. Sauer, "An Optoelectronic Multi-Gb/s Packet Switching Network", *OCS Tech. Rept. 89-06,* Optoelectronic Computing Systems Center, U. of Colo., Boulder, CO (1989)

[16] A.S. Acampora, M.J. Karol and M.G. Hluchyj, "Terabit lightwave networks: The multihop approach," *AT&T Tech. J.,* p. 21 (Nov./Dec. 1987)

[17] R. A. Thompson, "Architectures with Improved Signal-to-Noise Ratio in Photonic Systems with Fiber-Loop Delay Lines", *IEEE J. on Selected Areas in Comm.,* V. 6, pp. 1096-1106 (1988)

[18] S.V. Ramanan, H.F. Jordan and J.R. Sauer, "A new time domain, multi-stage permutation algorithm," to appear *IEEE Trans. on Info. Theory,* V. 36, No. 1, pp. 171-173 (1990)

[19] V.E. Benes, *Mathematical Theory of Connecting Networks and Telephone Traffic,* Academic Press (1965)

[20] D. C. Opferman and N. T. Tsao-Wu, "On a Class of Rearrangeable Switching Networks - Part I: Control Algorithm; Part II: Enumeration Studies of Fault Diagnosis", *Bell System Technical Journal,* pp. 1579-1618 (1971)

[21] J.P. Pratt, "HATCH User's Manual," *OCS Tech. Rept. TR 89-31,* Optoelectronic Computing Systems Center, U. of Colo., Boulder, CO (1989)

[22] F.A. Ficken, *The Simplex Method of Linear Programming*, Holt, Rinehart and Winston, New York (1961)

[23] N. Christofides, *Graph Theory*, Academic Press, London (1975)

[24] J.P. Pratt and V.P. Heuring, "A methodology for the design of continuous-dataflow synchronous systems," *OCS Tech. Rept. TR 89-23*, Optoelectronic Computing Systems Center, U. of Colo., Boulder, CO (1989)

[25] J.P. Pratt and V.P. Heuring, "Synchronization, Crosstalk and Loss in Optical Systems," Submitted to *Applied Optics*, (August, 1990)

[26] A.F. Benner, J. Bowman, T. Erkkila, R.J. Feuerstein, V.P. Heuring, H.F. Jordan, J.R. Sauer and T. Soukup, "Digital Optical Counter Using Directional Coupler Switches," subitted to *Applied Optics*, (Sept. 1990)

[27] M.J. Murdocca, A. Huang, J. Jahns and N. Streibl, "Optical design of programmable logic arrays," *Applied Optics*, V. 27, p. 1651 (1988)

[28] B.K. Jenkins and A.A. Sawchuk, "Optical cellular logic processors," *J. Optical Soc. Amer. A*, V. 2, No. 13, p. 26 (1985)

NASA Report Documentation Page

NASA — National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA CR-187520 ICASE Report No. 91-18	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle DIGITAL OPTICAL COMPUTERS AT THE OPTOELECTRONIC COMPUTING SYSTEMS CENTER	5. Report Date February 1991
	6. Performing Organization Code

7. Author(s) Harry F. Jordan	8. Performing Organization Report No. 91-18
	10. Work Unit No. 505-90-52-01

9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	11. Contract or Grant No. NAS1-18605
	13. Type of Report and Period Covered

12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	Contractor Report
	14. Sponsoring Agency Code

16. Abstract

The Digital Optical Computing Program within the National Science Foundation Engineering Research Center for Optoelectronic Computing Systems has as its specific goal research on optical computing architectures suitable for use at the highest possible speeds. The program can be targeted toward exploiting the time domain because other programs in the Center are pursuing research on parallel optical systems, exploiting optical interconnection and optical devices and materials. Using a general purpose computing architecture as the focus, we are developing design techniques, tools and architectures for operation at the speed of light limit. Experimental work is being done with the somewhat low speed components currently available but with architectures which will scale up in speed as faster devices are developed. The design algorithms and tools developed for a general purpose, stored program computer are being applied to other systems such as optimally controlled optical communications networks.

17. Key Words (Suggested by Author(s)) optical computing, computer architecture, optical communication networks	18. Distribution Statement 33 – Electronics and Electrical Engineering 62 – Computer Systems Unclassified – Unlimited

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 20	22. Price A03