A High Speed Optical Radix Sorter*

Harry F. Jordan[1], Kyungsook Y. Lee[2], Rajgopal Kannan[2], Coke Reed[3], Jon R. Sauer[1], Douglas E. Straub[1]

1. Dept. of Electrical and Computer Engineering, University of Colorado, Boulder, CO 80309-0425

2. Dept. of Mathematics and Computer Science, University of Denver, Denver, CO 80208

3. Center for Computing Sciences, 17100 Science Drive, ∫owie, MD 20715

Abstract

We discuss the design of an optical radix sorter capable of multiple channel operation at 100 Gb/s on each of about 16 channels, for an aggregate rate of about 1.6 Tb/s. The high bit rate is obtained by partitioning the problem into a minimal amount of single bit switching and confining switching to packets consisting of full words in the rest of the system. A high speed optical gate called a TOAD is suitable for the bit rate switching, and lower speed, electro-optic directional couplers are used to temporally and spatially rearrange packets. The sorting logic has been verified by constructing a low speed version that uses directional couplers for both bit rate and packet rate logic.

* This research was supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while an author (Jordan) was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

1. Introduction

Dedicated hardware can be competitive with general purpose supercomputers for certain applications where a small fraction of the full supercomputer functionality is needed. A mismatch of data size to supercomputer word length can also favor special purpose hardware. We show how to implement a high speed sorter using optics. The underlying structure is serial, but the bit rate can be very high. A target rate of 100 Gigabits per second for each serial stream yields a system competitive with a general purpose computer. This paper discusses a sorting algorithm [1] that is amenable to optical implementation at this rate. The algorithm minimizes the number of high speed switches needed while still maintaining a high bit rate. A preliminary low speed optical implementation at 50 Mbit/s has been built that demonstrates the techniques involved.

A key feature of the design is distinguishing bit rate and word rate operation. A few expensive high-speed optical components do the minimal amount of bit rate processing required. The major part of the processing logic only switches full words at a time. This is done with lower cost electro-optic devices that have high throughput bandwidth but lower switching speeds.

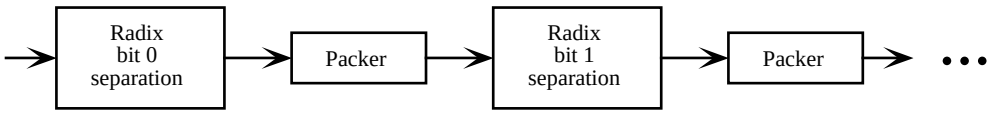
2. System overview

We assume sets of $N = 2^n$ input numbers which are, or can be configured, in a serial stream. It is assumed that successive sets of N numbers do not depend on the results of sorting previous sets, so that pipelined processing is feasible. The overall scheme is based on a binary radix sort [2]. Each stage performs operations based on one bit of each number, say the kth. Figure 1 shows two possible configurations for the overall system. For high speed, pipelined operation, a series of M stages can be used for M bit numbers, each stage devoted to one bit of the numbers, as shown in Fig. 1(a). For less expensive hardware, the number stream can be recirculated through a single stage M times, changing the bit k that the stage deals with each time, as shown in Fig. 1(b).

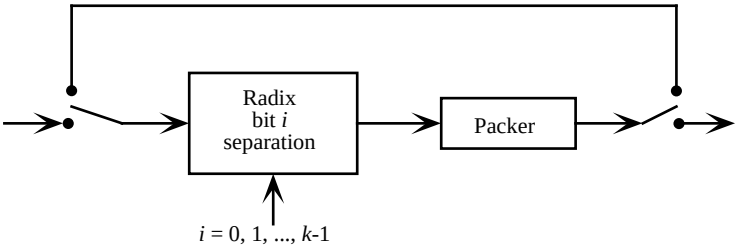
The job of stage k of the radix sorter is to place all numbers with bit $k = 0$ ahead of all numbers with bit $k = 1$ while preserving the input ordering for numbers within each of the two groups. Application of this stable binary rearrangement for k = 0, 1, ..., M-1 gives a full binary radix sorter for N numbers of M bits.

In the serial domain the stage k operation is performed as follows. The dense input stream is partitioned into two sparse streams, consisting of time slots containing either numbers or no data. Stream 0 contains the numbers having bit $k = 0$ and stream 1 contains those having bit $k = 1$. Stream 1 is then delayed by N time slots and appended to stream 0. This results in a frame of $2N$ time slots, N of which are occupied by numbers in the correct order for the output of the stage and N of which are empty. The rest of the stage consists of a packer that turns the sparse frame of $2N$ slots into a dense frame of N numbers, preserving their order, and an empty frame of N slots.

An example of sorting on one bit is shown in Fig. 2. We will see later that efficiency is



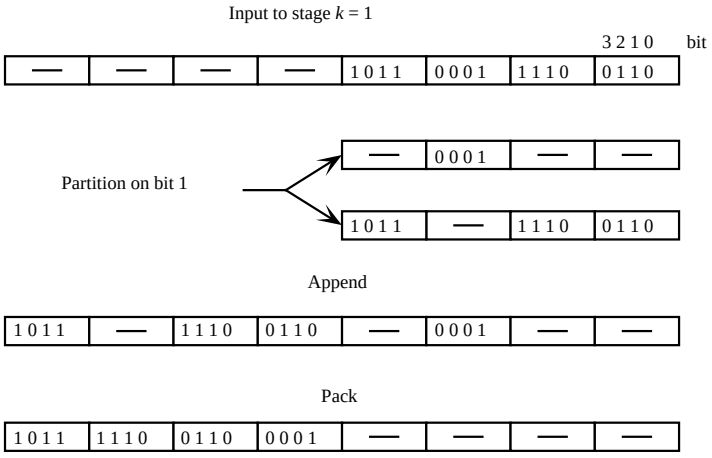
(a) Pipelined parallel radix sorter



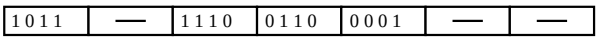
(b) Serial recirculating radix sorter

FIGURE 1. Radix sorter architectures

gained if the sparse frame to be packed has 2*N*-1 rather than 2*N* slots. A modified append operation based on where the last number was placed by the partition makes this possible and is shown in the figure also.



(a) Single radix bit separation and packing



(b) Smart append based on routing of last key

FIGURE 2. Example of serial sorting on one radix bit

The division of a stage into radix bit partition and packer portions is consistent with the goals of very high bit rate operation and reasonable cost. Only the radix bit detection must operate at the bit rate. The partitioning switch and the packer deal only with switching complete numbers. If a number is considered a packet, *M* is on the order of 64, and some

switching gap is allowed between packets, the packet rate is two orders of magnitude slower than the bit rate. The partitioning of the hardware for the sorter stage for a single radix bit is shown in Fig. 3. The five terminal device is a controlled 2×2 exchange switch,

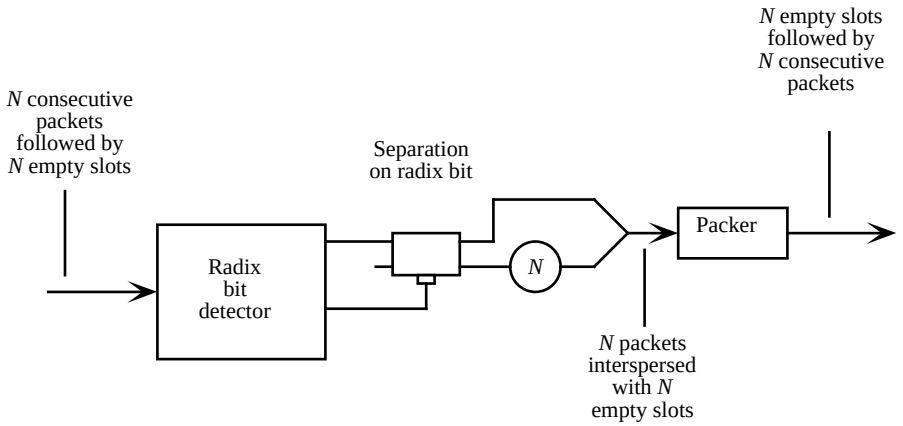


FIGURE 3. Sorter Stage for One Radix Bit

and the circle is an *N* packet delay.

The bit rate of a serial sorter needs to be in the 100 GHz to 1 THz range to make it competitive with other techniques. Two possible optical switching devices allow sampling a bit out of $M \approx 64$ bits at that rate. One is the Sagnac [3] interferometer gate and the other is the Terahertz Optical Asymmetric Demultiplexer, or TOAD [4]. The disadvantage of the Sagnac gate is its extremely long latency, on the order of 10^6 bit times. The disadvantage of the TOAD is that, after sampling one bit, there is a recovery time of tens of bit times before another can be sampled. In sampling the *k*th bit of successive *M* bit numbers the recovery time is not a problem, so the TOAD device is favored.

Once the *k*th bit has been detected and extended to the length of a time slot, it is used to control the switch that partitions the input into two sparse streams. This switch changes no faster than the packet rate, and with the introduction of some gap between packets, a device like a switched optical directional coupler [5], [6] can be used for partitioning. The second stream is then delayed by *N* slot periods and appended to the first to obtain the sparse stream of 2*N* slots having the *N* full slots in the correct order for the stage, but interspersed with *N* empty slots. Hardware to produce the sparse stream for one stage is shown in Fig. 4. Modified hardware to reduce the length of the sparse stream from 2N to 2N-1 is described in Sect. 3.2.

The two challenging parts of the sorter design are the radix bit detector and the sparse stream packer. The radix bit detector is challenging because of the very high bit rate at which it must operate. The problems to be addressed involve producing devices to operate at this rate, and supplying timing and control to them. The sparse stream packer is challenging as a system design problem. Its switches and control logic operate at a speed which is still high, so a minimal switch count design is strongly favored from a cost point of view. A minimum frame delay is also desirable, especially for recirculating operation with one stage.

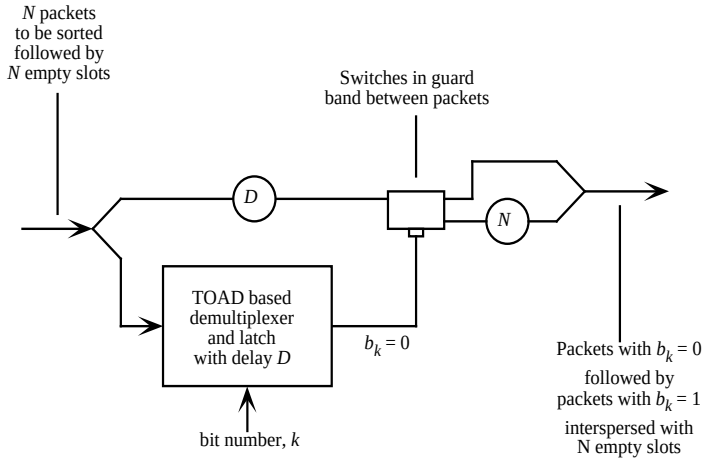


FIGURE 4. Separation on One Radix Bit

3. Packer network

The packer network takes as input a sparse frame of $2N$ ($N = 2^n$) packets. N of the packets are empty, and N contain numbers being sorted. The network compacts the full packets, maintaining their order, so that all appear adjacent in the output frame after a fixed frame delay. Switches need only change at the packet rate and can use the same technology as the partitioning switch of the radix separation section. The network to be described works on the principle that full packets must be moved forward relative to the start of the frame by a number of packet slots equal to the count of empty packets preceding them at the input to the packer network. This is accomplished by causing full packets preceded by empty slots to bypass powers of 2 delays whose total makes up the overall frame delay of the system. The design resembles the tunable optical delay described in [7] and the time slot interchangers of [8].

An important feature of the solution presented below is the restriction of switch control logic to serial hardware operating at a bit rate equal to the number of packets per second. This eliminates, for example, the possibility of attaching a multi-bit count to each full packet moving through the network and using it for routing purposes. Instead, the circuit described distributes the counting of empty packets over both time and the stages of the network using simple bit serial logic operating at the packet rate. The control logic is similar to the optical logic used in demonstrating an all-optical, stored program computer [9].

3.1 Packer architecture and operation

The packer consists of a serial array of n+1 stages. Let packets enter the network at discrete times t, starting with t=0. An empty packet at time t is denoted by $E(t) = 1$. A binary count of the number $n_e(t)$ of empty packets from the beginning of the frame up to time t is kept. The Boolean variables $S_j(t)$, j = 0, 1, ...,n represent the count in the format

$$n_e(t) = S_n(t + 2^n - 1)\ldots S_j(t + 2^j - 1)\ldots S_1(t + 1)S_0(t). \quad \text{(EQ 1)}$$

The count is kept by associating a bit of S with each of n+1 stages of the packer network. Empty packets are counted starting in stage 0 and carries are transmitted from stage j to stage j+1 with a delay of 2^j time units. A full packet at stage j is delayed by 2^j on the way to stage j+1 if and only if $S_j(t) = 0$ at its arrival time t at stage j. The packer delay line and its control are shown in Fig. 5. The delay line uses 2×2 exchange switches that are in the cross state if their control input is zero and in the bar state if it is one. Delays are measured in word times, and switches change no more often than once per word. The control consists of half adders that form a distributed serial counter. The bit rate of the counter is the word rate of the sorter. As long as there is no carry in, the sum bit is effectively stored by the one bit delay at the output of each half adder.

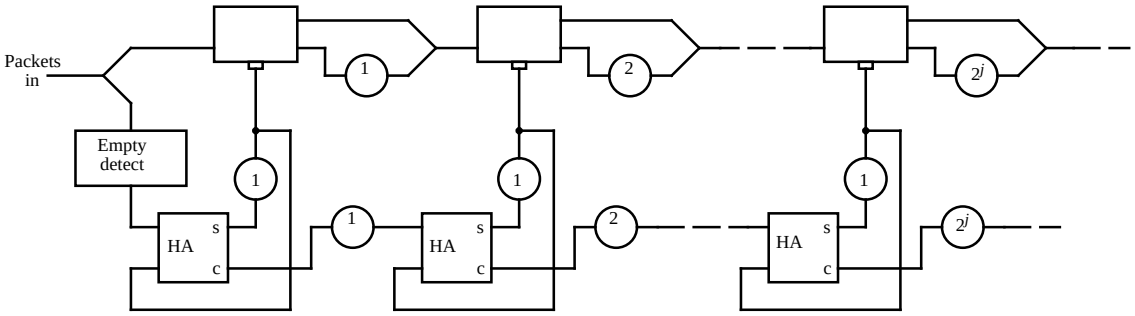


FIGURE 5. The packer line and its control

Three arguments are involved in the proof of correct operation of the packer network. The first relates the value of the bit $S_j(t)$ seen by a full packet arriving at stage j at time t to the count of empty packets preceding it at the network input. The second shows that the output consists of N adjacent full packets starting at a fixed frame delay. The third demonstrates that there is no collision between full packets within the network. The tricky part is that bits of one counter value are distributed both in space and time, as shown by Eq. 1.

The temporal behavior of a count bit in a stage is as follows.

Lemma 1: A packet arriving at stage j with a delay of $2^j - 1$ "sees" the same count bit as one that arrives earlier by $n_e(t) \bmod 2^j$ packet times, i.e.

$$S_j(t + 2^j - 1 - (n_e(t) \bmod 2^j)) = S_j(t + 2^j - 1). \quad \text{(EQ 2)}$$

The proof compares the times that a carry and a full packet reach stage j.

The hardware diagram shows that the top row of switches cause a packet to be delayed in passing from stage j to stage j+1 by 2^j slot times if $S_j = 0$ when it arrives at stage j. This leads to a calculation of the arrival times of full packets at different network stages.

Lemma 2: A full packet $F(t)$ arriving at the network input at time t, arrives at stage j of the network at time $t + 2^j - 1 - (n_e(t) \bmod 2^j)$.

The proof uses finite induction on the stage number, *j*.

We are now in a position to prove that the full packets arrive densely packed at the output, with the first full packet appearing after a fixed frame delay of 2*N*-1.

Corollary: The $N = 2^n$ full packets F_i, $i = 0, 1, ..., N\text{-}1$ arrive at the network output sequentially at times $2N\text{-}1+i$.

The proof results from Lemma 2 by noting that if the *i*th full packet F_i arrives at the network input at time t_i, then the number of empty packets preceding it is $n_e(t_i) = t_i - i$.

The last task is to show that there is no collision between full packets within the network, or alternatively, that no two full packets arrive simultaneously at a common point in the network. This is done by showing that the difference in arrival times of two different packets, given by Lemma 2, must be nonzero.

Lemma 3: Two distinct full packets F_i and F_m, $i \neq m$, do not arrive at any stage *j* of the network simultaneously.

Thus the hardware presented takes a sparse stream of packets formed by concatenating the two radix bit separated streams of *N* slots and compacts the full packets into a dense frame of *N*. The compaction is done in an order preserving manner so that its use with separation on a radix bit constitutes a stage of a stable sort.

3.2 Optimization of the packer

The packer structure described in the last section makes clear the usefulness of the smart append mentioned in Sect 2. With *N* full packets and *N* empty slots in the input stream to the packer, a packet might need to move at most $N = 2^n$ slots toward the start of the frame. The structure of the *n*+1 stage network makes it possible to advance a packet by $2^{n+1} - 1 = 2N - 1$ slots. An *n* stage packer can compact a stream with at most *N*-1 empty slots. Fortunately, a simple modification of the radix bit separation hardware can produce an input stream of length 2*N*-1 consisting of *N* packets and *N*-1 empty slots. This saves one full stage in the packer and reduces the packer frame delay from 2*N*-1 to *N*-1.

Let the output of the radix bit detector for packet *i* be $b_k(i)$, $0 \le i < N$. If $b_k(N-1) = 0$, the last packet is not delayed in Fig. 4, and the length 2*N* output stream has an empty packet at its end, allowing it to be treated as a length 2*N*-1 stream. If $b_k(N-1) = 1$, then the *N* packet slots emerging from the non-delayed switch output in Fig. 4 have an empty slot at the end. If the delay is made *N*-1 packets in this case, the first slot of the delayed stream overlaps the empty last slot in the non-delayed stream, again giving a length 2*N*-1 stream. The circuit to accomplish this is shown in Fig. 6, and Fig. 2 showed an example where the second partition is delayed by N-1 rather than *N*.

Another drawback of the packer that can be addressed by a change in architecture is the 3dB signal loss of each passive coupler in the data path. The data path is singled out because regeneration of the high bit rate signal is potentially costly, so losses should be minimized. The passive couplers can be eliminated by using the second input of the

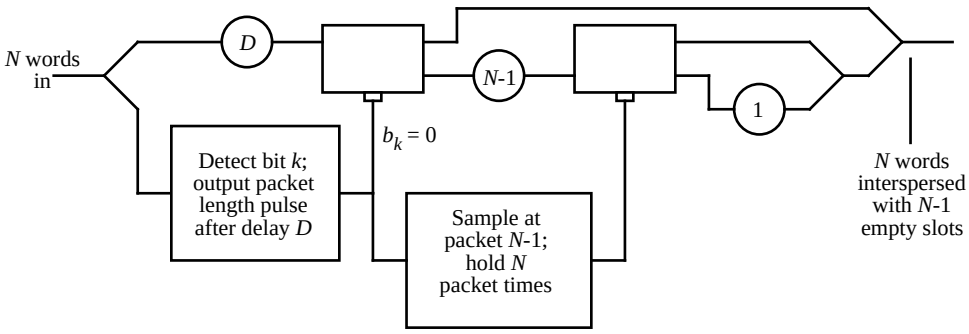


FIGURE 6. Smart append for length 2*N*-1 sparse stream

exchange switch in the next stage. With modified control, a packet on one of two data input lines to a stage can be placed on the correct (delayed or non-delayed) output line.

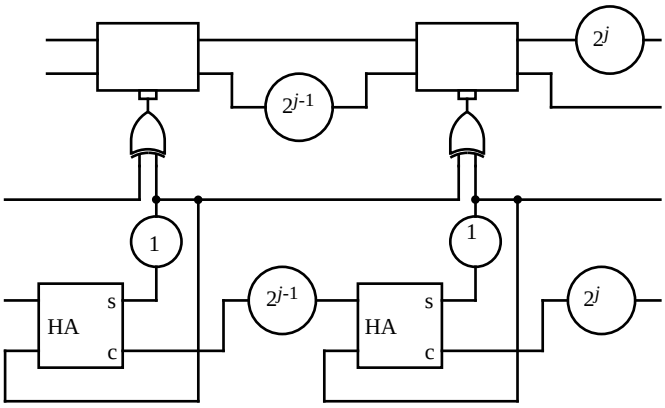


FIGURE 7. Eliminating 3dB couplers with modified control

As shown in Fig. 7, the two outputs of each stage connect to the two inputs of the next and the position of the delay alternates in each successive stage. An extra control connection is added between stages. The control of the modified stage j exclusive ORs the count variable S_{j-1} with S_j to control the data switch in stage j. The signal S_{j-1} is forwarded to stage j with no delay. Let the time of arrival of full packet F_i at stage j be

$$t(i,j) \;=\; t_i + 2^j - 1 - n_e(t_i) \bmod 2^j . \qquad \text{(EQ 3)}$$

The following lemma proves the correctness of this control function.

Lemma 4: Full packet F_i switches from the delayed to the non-delayed line (or from the non-delayed to the delayed line) at stage j iff

$$S_{j-1}(t(i,j)) \oplus S_j(t(i,j)) \;=\; 1 . \qquad \text{(EQ 4)}$$

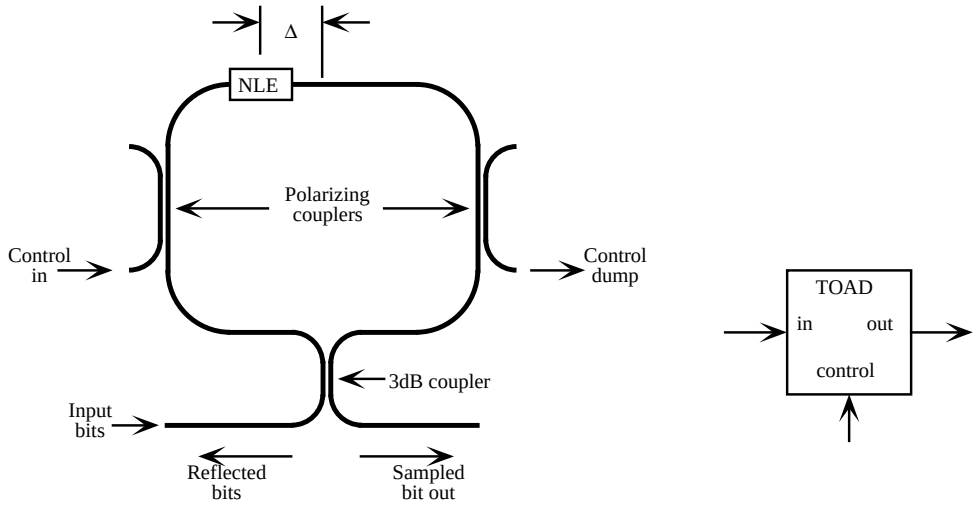
The proof separates into two cases based on the value of bit j-1 of $n_e(t_i)$, which determines whether packet F_i is delayed between stages j-1 and j. If it is not delayed, the switch control function of Eq. 4 is clearly correct, and if it is delayed, the structure of the

carry delays shows that bit S_{j-1} is still in the right state when F_i reaches stage j.

4. Radix bit partition section

The task of the partitioner is to test the *k*th bit of each input number and convert the bit to a packet length signal timed to arrive at the partitioning switch synchronized with the word whose bit was tested. The high speed radix bit tester must be able to isolate the *k*th bit from those on either side and hold its value to stretch it into a packet length signal. The value of *k* is constant for *N* words and increases by one for the next N words in a recirculating system. Thus, although the bit rate may be very high, a recovery period of one packet time is guaranteed between sampled bits. A gap between words can be designed into each packet slot to eliminate the need for bit time precision in switching full words. If the delay through the radix bit tester is to be independent of *k*, it must be at least one word time because the sampled bit may be at the end.

Optical implementation of the partition section relies on the TOAD [4] device. A TOAD is a form of nonlinear optical loop mirror (NOLM). Its operation is briefly described



(a) TOAD loop mirror structure

(b) TOAD symbol

FIGURE 8. TOAD gate structure and symbol

referring to Fig. 8. Incoming bits are split by a 3dB coupler into two equal intensity streams which counterpropagate around a closed loop. Half pulses propagating in opposite directions see precisely the same optical path length, provided they are not strong enough to alter the state of the non-linear optical element NLE. Thus they will be in phase when they return to the 3dB coupler and combine to reconstruct the input pulse traveling in the opposite direction, and this reflection gives rise to the name loop mirror.

A control pulse large enough to change the state of NLE alters the behavior. The NLE responds quickly to a sufficiently large light pulse with a change in its index of refraction. It recovers slowly to its steady state. The control pulse is timed to arrive at the NLE just

after the clockwise propagating half of the bit of interest passes the NLE but before the counterclockwise propagating half reaches it. The NLE index change is designed to shift the phase 180 degrees, so the two halves of the sampled pulse interfere destructively at the original input to the 3dB coupler. Since the reflected power is zero, the pulse power must exit through the other arm of the coupler.

Halves of an input pulse arriving after the control pulse see nearly the same index of refraction in the NLE because it recovers slowly, and they are reflected just as the pulses preceding the sampled one. The size of the time window created by a control pulse for sampling an input pulse is determined by the offset Δ of the NLE from the center of the loop. The control and input signals are orthogonally polarized so that the control pulse can be removed from the loop before it affects the output.

The discussion of the TOAD makes clear the role it can play in the radix bit partition section and also shows that some other devices are needed. The control pulse is timed to sample the *k*th bit using a variable delay, and if it originates as a data signal, it must be amplified to trigger the NLE. The *k*th bit isolated at the output is detected and used to produce a packet length switching signal. The additional devices are shown in Fig. 9.

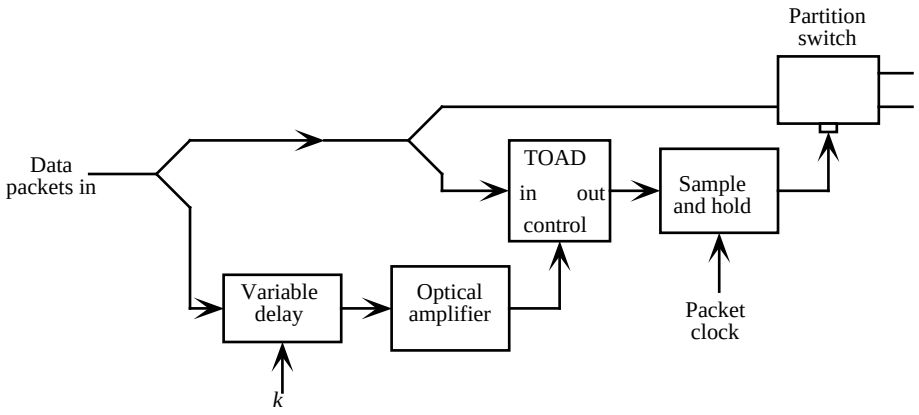


FIGURE 9. Devices making up the radix bit partition section

The control signal must precisely sample the *k*th bit under uncertainty in arrival time of packets. Bit level synchronization can be supplied by providing each packet with a start pulse, to which other bits can be referenced. If this is done, a copy of the data packet can be used to control the TOAD. The start bit, which can differ from data bits in amplitude, polarization, or other characteristics, is delayed by *k* bit times to trigger the NLE. Either the other bits of the word are capable of triggering the NLE or the control input must be turned off in, say, half a packet time to let the NLE to recover for the next word.

The variable delay must have a path length precision on the order of a bit time (1 mm at 100 Gb/s), but only needs to change in the gap between the last packet in one recirculation and the first packet of the next. Directional coupler switches and controlled waveguide lengths can satisfy these requirements.

5. Multiple Channel Sorting

To gain sorting speed, the input data set can be partitioned over several, say *m* serial optical channels. A small space switch among the channels enables sorting in both time and space with the same basic algorithm previously described. We describe a system that sorts in space as the major dimension and time as the minor dimension as shown in Fig. 10.

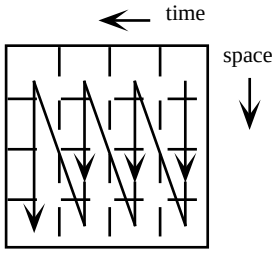


FIGURE 10. Sorting order for the multi channel space-time sort

Sorting within the *m* serial channels proceeds independently, with the only interconnection between channels being an *m* input, *m* output banyan network with a control that can be distributed over its stages. Each channel has *N*/*m* numbers, and within any radix bit stage, the radix bit separation and packer portions behave exactly as described earlier. The banyan switch interconnects the channels between the radix bit separation and packer portions of a stage, as shown for 4 channels in Fig. 11.

For each set of 4 empty or full packets appearing at one time slot, a modulo 4 counter starts with the count from the last time slot and assigns channel numbers to the current full packets. The banyan network then routes each full packet to the channel corresponding to its assigned number. If a reverse banyan is used, the low order count bits control the first stage, and successively higher order bits control subsequent stages. The channels to which the empty packets are routed are unimportant. The guarantee that a reverse banyan network can route the required permutation comes from Theorem 1 of [10], as follows.

Theorem: An $N \times N$ reverse banyan network remains non-blocking when the input packets satisfy the following constraints:

- The set of destination addresses form a permutation π on $[0, 1, ..., N\text{-}1]$.
- π can be decomposed into two disjoint sets $\pi_1 : S_1 \rightarrow [k, k\hat{+}1, \ldots, k\hat{+}(m-1)]$ and $\pi_2 : S_2 \rightarrow [k\hat{+}m, k\hat{+}(m+1), \ldots, k\hat{-}1]$, where $k \in [0, 1, \ldots, N-1]$ is an arbitrary destination address, $1 \le m < N$, and $\hat{+}$ ($\hat{-}$) denotes addition (subtraction) modulo *N*. $S_1 = [i_0, i_1, \ldots, i_{m-1}]$ and $S_2 = [j_0, j_1, \ldots, j_{N-m-1}]$ are disjoint, ordered index sets with $S_1 \cup S_2 = [0, \ldots, N-1]$ and $i_0 < i_1 < \ldots < i_{m-1}$, $j_0 < j_1 < \ldots < j_{N-m-1}$. Additionally,
  $\pi_i(i_t) = \pi(i_t) = k\hat{+}t$, $0 \le t \le m-1$, and
  $\pi_2(j_s) = \pi(j_s) = k\hat{+}m\hat{+}s$, $0 \le s \le N-m-1$.
  In words, packets residing on input lines belonging to the set S_1 are routed to consecu-

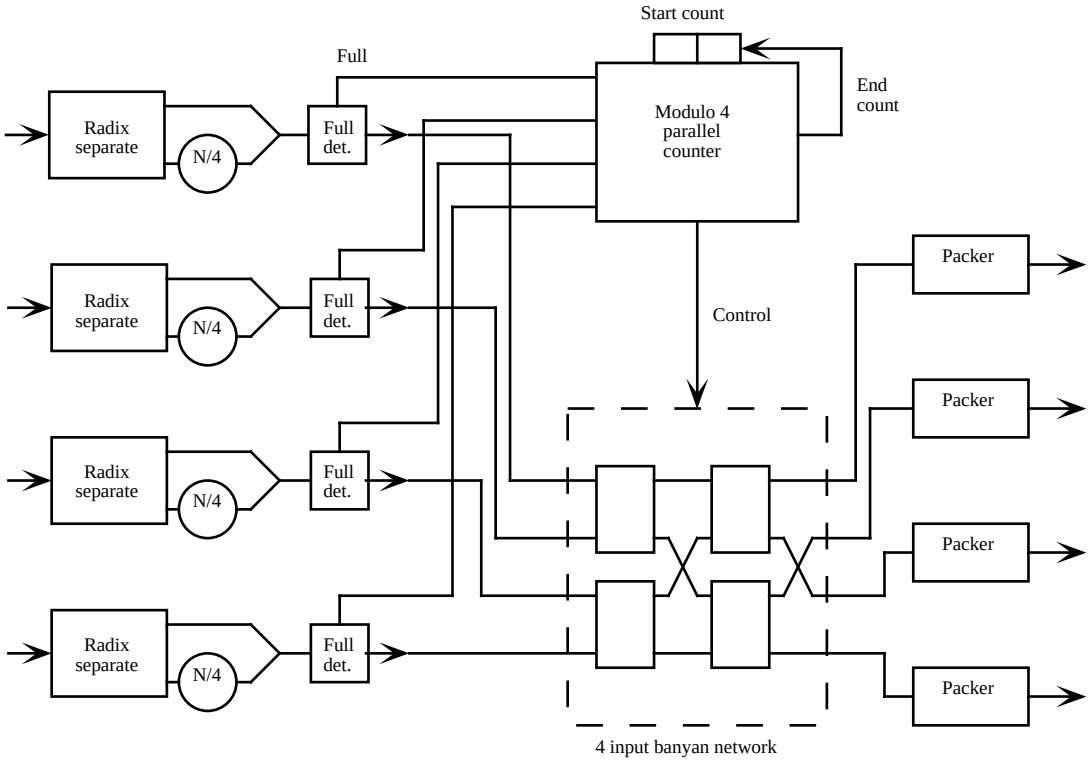


FIGURE 11. Space-time radix sorter stage with 4 parallel channels

tive output ports k through $k \hat{+} (m-1)$ while packets belonging to the set S_2 are routed to output ports $k \hat{+} m$ through $k \hat{+} (N-1)$.

In the present application of this theorem, the two disjoint and exhaustive subsets are provided by the full and empty packet slots. The full packets are routed to consecutive channel numbers, modulo 4, starting at the channel after the last one assigned in the previous time slot. The effect of this operation is to assign all full packets to the space channel that they should occupy at the end of this radix bit stage. The time ordering is not disturbed by the space permutation, so all packets with this radix bit =0 still precede all those with this bit =1, and there are still interspersed empty slots. The independent packers then remove the empty slots leaving a space-time matrix of packets sorted on this radix bit. Multiple passes through the same or different identical stages for each radix bit completes the sort.

6. System Issues

Two overall system concerns must be addressed for a complete single stream sorter design. They apply to both recirculating and pipelined systems, but are more easily discussed by focusing on one of the two. We thus assume a recirculating system in what follows. The issues are elimination of the N packet idle period for the radix bit partition section and signal quality restoration.

6.1 Eliminating the idle period for radix bit partition

Although the packer is occupied by the sparse stream for 2*N* packet times, the first packed result is available after a delay of only *N*-1 packet times. This is early enough to start information recirculating into the radix bit separator as soon as it becomes idle. The separator can partition on the next bit of the radix as the data arrives. It is only necessary to provide a second packer to accept the first *N* packets of the next sparse frame while the last N packets of the previous sparse frame are exiting the first packer. At this point, packets begin to exit the second packer, but now both the radix bit separator and the first packer are idle. By alternately using two packers the radix bit separator can be kept fully occupied, and the time to complete a sort is reduced by a factor of two.

6.2 Signal quality restoration

Three aspects of optical signal quality must be controlled: maintaining high amplitude one bits, controlling low amplitude noise in zero bits, and maintaining signal timing (synchronization). Noise control may be separated from amplification because high speed semiconductor or erbium doped fiber amplifiers are a choice at high bit rates. These devices introduce and amplify low level noise. A closed loop system using optical amplifiers thus needs separate noise suppression to counteract the noise gain of the amplifiers.

The most promising device for attenuating noise in zeros seems to be a NOLM with an asymmetric amplifier/attenuator [11]. Published data shows a low level threshold behavior for output versus input power. The work cited uses linear attenuation but nonlinear amplification for a differential effect on pulses traveling in opposite directions around the loop mirror. Fast saturable absorbers offer another way to achieve this effect.

Combining a zero noise attenuator with the natural gain saturation of a semiconductor or erbium optical amplifier would give a system which both limited the power in one bits and maintained low power in zeros. The third issue of maintaining bit level synchronization may not be a problem in the present application. With timing derived from a start bit in each packet, the only question is whether dispersion causes bit overlap over the total fiber length traversed by a packet. This effect is small and appears only for very long fibers.

7. Comparison with sorting by electronic computer

We compare the optical radix sorter with results from sorting on a 64 processor Cray T3D computer. The number of words to be sorted is *N*, and the number of bits per word is *k*=64. The Cray T3D results* come from a distributed sorting algorithm which has an execution time that is dependent on the original order of the data. The results reported for random initial order show a 64 processor, 150 MHz, Cray T3D sorting 64×10^6 numbers in 2.41 seconds. The time on the same system for originally sorted data (suspected to be near the worst case) is 8.47 seconds.

---

* Lance Joneckis, private communication.

We compare against two versions of the optical radix sorter, one with a recirculating structure and the other pipelined with repeated hardware. The optical sorter uses m spatial channels to do a time/space sort of a single set of N words, partitioned into N/m words per channel. We take m=32 as a practical limit for the number of space channels, based on the complexity of the switching network connecting the channels. We take the word length to be k=64 to correspond to the Cray T3D results. This means k recirculations of the data in the recirculating system, and a factor of k more hardware in the pipelined version.

We measure the operating rate of the optical system by the single channel word rate R (in Gwords/sec). This is the switching rate required of the directional couplers in all but the radix bit sampling parts of the system. We also consider limited wavelength multiplexing of bits in a word. If the bits in a word are wavelength multiplexed with w wavelengths, then the bit rate for radix bit sampling is $64R/w$ Gbits/sec. Wavelength multiplexing on bits of a word thus has no effect on R but decreases the bandwidth demands on bit rate operations. The system parameters are summarized in Table 1.

Symbol	Meaning	Value used in comparison
N	number of words to be sorted	64×10^6
k	bits per word	64
m	number of optical channels	32
w	number of multiplexed wavelengths	8
S	target speedup of optics over electronics	10^3
R	word rate of optical channel	to be determined

TABLE 1. System parameters for comparison calculations

For either the recirculating or pipelined architecture, the latency from inserting the first bit of the input data set to delivery of the first bit of the result is $4\times10^{-9}\,(N/R)$ sec. The time from start to delivery of the last result for a complete sort is $4.03\times10^{-9}\,(N/R)$ sec.

For the non-recirculating sorter, the fiber length for all m channels is $2.56\,(N/R)$ m. For 125 micron diameter fiber, the volume occupied by the glass in all the sorter fiber is $0.315\times10^{-6}\,(N/R)$ m^3.

For the recirculating sorter, the fiber length for all m channels is reduced by the number of bits in a word, and becomes $0.4\,(N/R)$ m. The volume of glass for 125 micron diameter fiber becomes $4.92\times10^{-9}\,(N/R)$ m^3.

We compare with the T3D result of 2.41 seconds to sort $N = 64\times10^6$ words on a 64 processor Cray T3D, remembering that this can go up to 8.47 seconds for badly ordered data. The optical sorter results have no dependence on initial data order.

We start with the recirculating version of the optical sorter, which must complete one sort before starting another. For $N = 64\times10^6$ words, the time to complete an optical sort from input of the first bit is $0.258R^{-1}$ sec. In order to have a speedup of 10^3 over the sorting time on a 64 processor Cray T3D, R would have to be 107 Gwords/sec. The bit rate with

w=8 way wavelength multiplexing would be 856 Gbits/sec.

The fiber length required by the recirculating sorter would be 239 km. The volume of glass for 125 micron fiber diameter would be 5.8×10^{-3} m^3.

With the non-recirculating, pipelined sorter, sorting multiple sets of N numbers can produce a new result every $2Nm^{-1}r^{-1}\times10^{-9}$ sec. Thus a series of $N = 64\times10^6$ word sorts with $m = 32$ optical channels could complete at a rate of one sort every $4\times10^{-3}R^{-1}$ sec. For this mode of operation to have a speedup of 10^3 over the 64 processor T3D would require R = 1.66 Gwords/sec. With 8 fold wavelength multiplexing, the bit rate for radix bit sampling is 13.3 Gbits/sec, and the total fiber length required by the non-recirculating sorter is 987×10^3 km. The glass volume of 125 micron diameter fiber would be 12.1 m^3. The longest single fiber in this sorter would be 241 kilometers.

In summary, gaining a factor of 1,000 over the 64 processor Cray T3D results with a recirculating optical sorter would require the unreasonably high word rate of 107 Gwords/sec with the system parameters assumed. The pipelined sorter without recirculation can operate on multiple input data sets and gain a factor of 1,000 over the 64 processor Cray T3D with a reasonable word rate of R = 1.66 Gwords/sec. The fiber length and volume, however, become fairly impractical. It is possible to vary the number of pipeline stages p anywhere between one and 64 to get compromises in these numbers. For example, p = 8 fold pipelining would require R = 13.4 Gwords/sec for a 1,000 fold speedup over the 64 processor T3D and would need 15.3×10^3 kilometers of fiber for a glass volume of 0.191 m^3.

The stretching of practical limits results from the large number, 64 million, of words in the sort. Sort set sizes on the order of 10^4 give reasonable values for fiber lengths while maintaining very high speeds.

8. Conclusions

A design for a high speed optical radix sorter has been presented. It is primarily serial in nature but can process parallel space channels for improved speed. Bit rates of several tens of Gbits/second/channel are feasible. The high speed comes from using full packet switching for most of the processing and using bit level operations only to test the one bit per word required by each stage of a radix sort. Testing one radix bit per word fits the fast response and slower recovery that is characteristic of the TOAD gate and makes it feasible to target a 100 MHz bit rate per stream. The packer that compacts the correctly ordered, but sparse, stream of $2N$ packet slots down to N consecutive packets can switch a factor of M more slowly, where M is the number of bits per word.

The packer uses passive fiber delay to accomplish the time shifts required by sparse stream compression. The amount of fiber required for delay lines becomes excessive for very large sorts, say 10^6 to 10^7 numbers. The high storage densities and low cost of semiconductor memory prevent the optical architecture from being competitive for sorting problems of this scale. Because each successive stage of the packer switches a factor of two more slowly, there is a potential to use a hybrid electronic design for later packer

stages. Signal conversion would be the major obstacle to using electronic RAM delay in slower switching stages.

An important possible application of the sorter is in optical communications network routing by sorting destination tags. The parallel channel version of the sorter does simultaneous arrangement in space and time, so it could be used to order packets on multiple time-multiplexed space channels.

References

[1] K. Bergman, G. Burdge, D. Carlson, N. Coletti, H. Jordan, R. Kannan, K. Lee, P. Merkey, P. Prucnal, C. Reed, and D. Straub, "Optical sorting, the fast fourier transform, and data packing," *Tech. Rept. CSDG 95-1,* Computer Systems Design Group, University of Colorado, Boulder, CO 80309-0425 (Nov. 1995).

[2] D. E. Knuth, *The Art of Computer Programming: Vol. 3/Sorting and Searching,* Addison-Wesley, New York (1994).

[3] A. Huang, et al., "Sagnac fiber logic gates and their possible applications: a system perspective," *Applied Optics,* Vol. 33, No. 26, pp. 6254-6267 (Sept. 1994).

[4] J. P. Sokoloff, P. R. Prucnal, I. Glesk, and M. Kane, "A terahertz optical asymmetric demultiplexer (TOAD)," *Photonics Technology Letters,* Vol. 5, No. 7, pp. 787-790 (July 1993).

[5] S. K. Korotky, et al., "Optical intensity modulation to 40 GHz using a waveguide electrooptic switch," *Appl. Phys. Lett.,* Vol. 50, pp. 1631-1633 (1987).

[6] United Technology Photonics, Part No. APE-YBBM-1.5-18-T-02.

[7] P. R. Prucnal, "Photonic fast packet switching," in *Photonics in Switching Vol. II,* Academic Press (1993).

[8] H. F. Jordan, D. Lee, K. Y. Lee, and S. V. Ramanan, "Serial array time slot interchangers and optical implementations," *IEEE Trans. on Computers,* Vol. 43, No. 11, pp. 1309-1318 (Nov. 1994).

[9] Harry F. Jordan, Vincent P. Heuring, and Robert F. Feuerstein, "Optoelectronic time-of-flight design and the demonstration of an all-optical, stored program, digital computer," *Proc. IEEE,* Special Issue on Optical Computing, Vol. 82, No. 11 (Nov. 1994).

[10] Rajgopal Kannan, Harry F. Jordan, Kyungsook Y. Lee, and Coke Reed, "A pipelined self-routing optical multichannel time slot permutation network," *Proc. 2nd Int'nl Conf. on Massively Parallel Processing Using Optical Interconnections,* IEEE Press, pp. 271-278, Oct. 23-24, 1995, San Antonio, TX.

[11] A. W. O'Neill and R. P. Webb, "All-optical loop mirror switch employing an asymmetric amplifier/attenuator combination," *Electronics Letters*, Vol. 26, No. 24, pp. 1008-09 (22 Nov. 1990).