Terabit Burst Switching
Progress Report (7/99–12/99)

Jonathan S. Turner
jst@cs.wustl.edu

WUCS-99-32

December 30, 1999

Department of Computer Science
Campus Box 1045
Washington University
One Brookings Drive
St. Louis, MO 63130-4899

Abstract

This report summarizes progress on Washington University's *Terabit Burst Switching Project*, supported by DARPA and Rome Air Force Laboratory. This project seeks to demonstrate the feasibility of *Burst Switching*, a new data communication service which can more effectively exploit the large bandwidths becoming available in WDM transmission systems, than conventional communication technologies like ATM and IP-based packet switching. Burst switching systems dynamically assign data bursts to channels in optical data links, using routing information carried in parallel control channels. The project will lead to the construction of a demonstration switch with throughput exceeding 200 Gb/s and scalable to over 10 Tb/s.

[0]This work is supported by the Advanced Research Projects Agency and Rome Laboratory (contract F30602-97-1-0273).

Terabit Burst Switching
Progress Report (7/99–12/99)

Jonathan S. Turner
jst@cs.wustl.edu

This report summarizes progress on the Terabit Burst Switching Project at Washington University for the period from July 1, 1999 through December 31, 1999.

1. Prototype Burst Switch Progress

The following paragraphs summarize status and progress on the various components being developed for the prototype burst switch. Figure 1 shows the overall structure of the prototype and details the location of each component in the system architecture.

- *Crossbar* (XBAR). The crossbar is the principal component of the burst switch datapath. It accepts 256 1 Gb/s data streams and switches each to one of 256 outputs for an aggregate data rate of 256 Gb/s. It uses a bit-sliced organization with nine parallel planes for carrying the data. Control inputs allow an input port to be selected for each output port, and an input with null data is selected when an output is unused. This avoids potential security problems that would arise if output ports were always transmitting data from a valid input. Successive groups of 32 outputs have independent control sections. This feature enables different Burst Processors to manage connections to the outputs they are responsible for without contention from other Burst Processors.

  The crossbar design has been restructured in order to allow implementation with Xilinx Virtex FPGAs rather than with a custom ASIC as originally planned. Reducing the single-signal data rate from 125 Mb/s to 62.5 Mb/s (and doubling the number of backplane signals to maintain the same total bit rate), along with fully synchronous data transfers allows use of commercially available FPGAs. Original estimates of allowed backplane density had precluded this solution. Four times as many crossbar chips are required (each is 256x64 at 62.5 Mb/s rather than 256x128 at 125 Mb/s for the ASIC), but this approach eliminates the need for a custom ASIC layout and fabrication costs.

  The crossbar will be composed of 72 FPGA chips with 8 chips for each of 9 bits (8 data, one control bit) on an individual card, and a total of 9 cards. Each FPGA implements a 256x64 crossbar at 62.5 Mb/s, and 8 are assembled on a single card to implement a 256x256 crossbar at 125 Mb/s. To avoid problems with multiplexing the outputs from several small crossbars

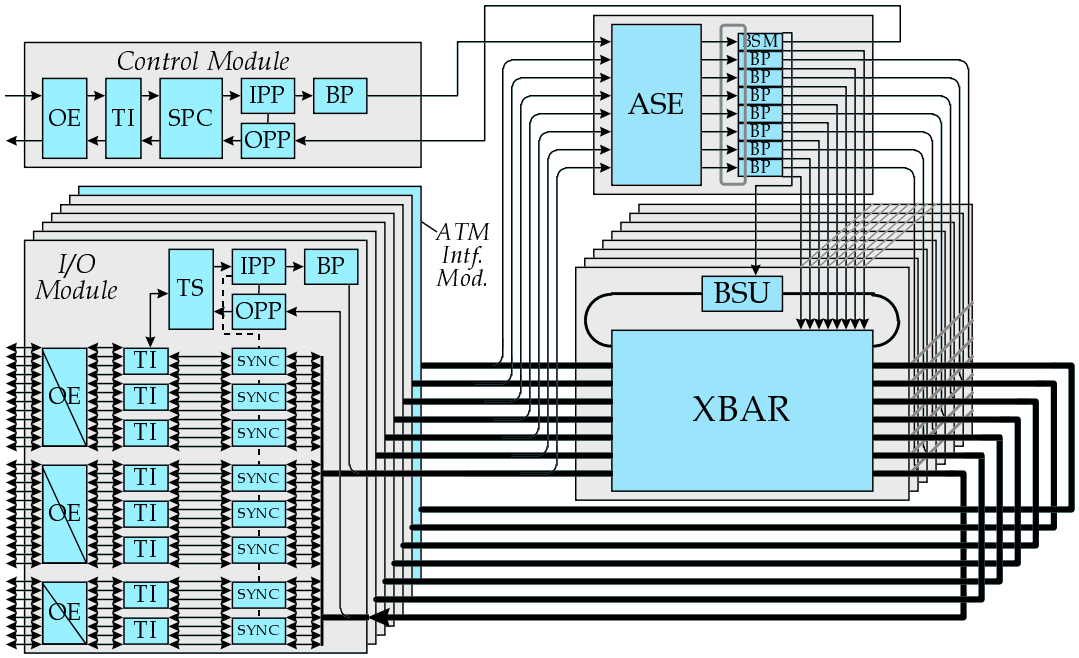

Figure 1: Prototype Burst Switch

to implement a single larger one, each individual FPGA accepts all inputs, and produces a subset of the outputs; no output multiplexing is required.

- *Synchronization Chip* (SYNC). The SYNC circuit accepts data from Serializer/Deserializer (SERDES) chips, delays it for up to 50 μs, and passes it on to the crossbar chips. Data returned from the crossbar chips is returned to the SYNC circuit and passed to the SERDES for transmission over the optical fibers. This was also originally planned to be implemented as an ASIC, however a test circuit to exercise the SERDES circuits and PAROLI electo-optical converters was implemented in an FPGA and showed that it was possible (with careful design) to obtain the necessary 125 Mb/s operation required for the SERDES. Present plans are to implement the SERDES to crossbar path with a FIFO and some minimal control within an FPGA, and to implement the path from crossbar back to SERDES with an FPGA. The delay in the SERDES to crossbar path will be accomplished by allowing the FIFO to fill with a specified number of words before it is read. Following this, words will be written and read at the 125 MHz rate to maintain a constant number of values in the FIFO (and a constant delay).

- *Burst Storage Unit* (BSU). The BSU provides an interface between the crossbar and the memory in which bursts are stored when they cannot be sent directly to the outgoing links. Each BSU supports 32 channels and has an aggregate throughput of 4 Gb/s. It uses a 128 bit wide memory, made up of four 1 MB static RAM chips, plus two additional memory chips which hold linked list pointers to enable the BSU to manage the memory in a flexible fashion. It is being implemented using an FPGA, to minimize risk and provide flexibility for alternative

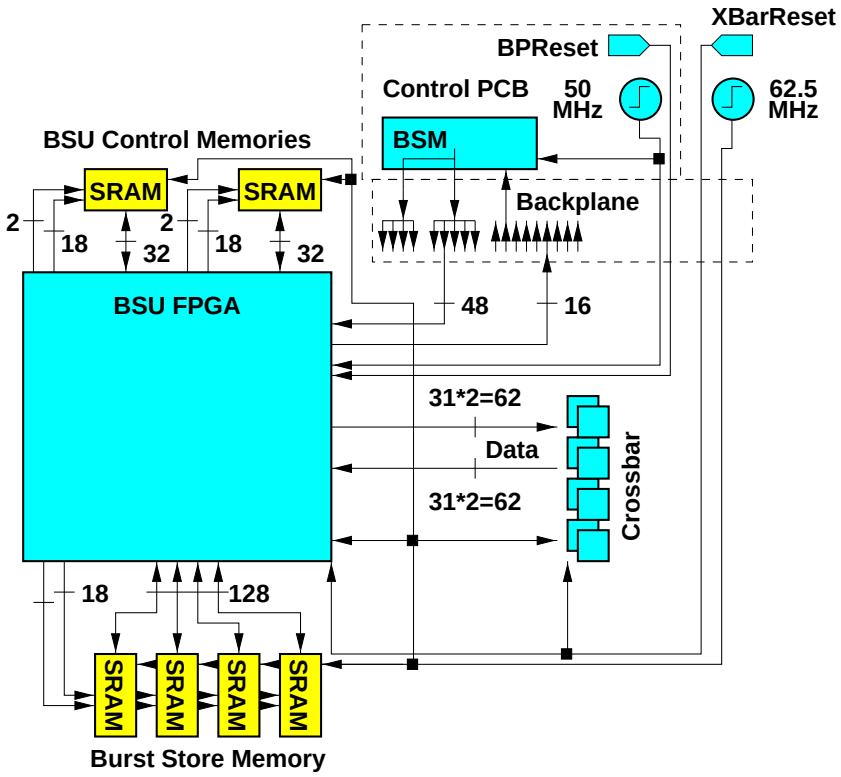

Figure 2: Burst Storage Unit

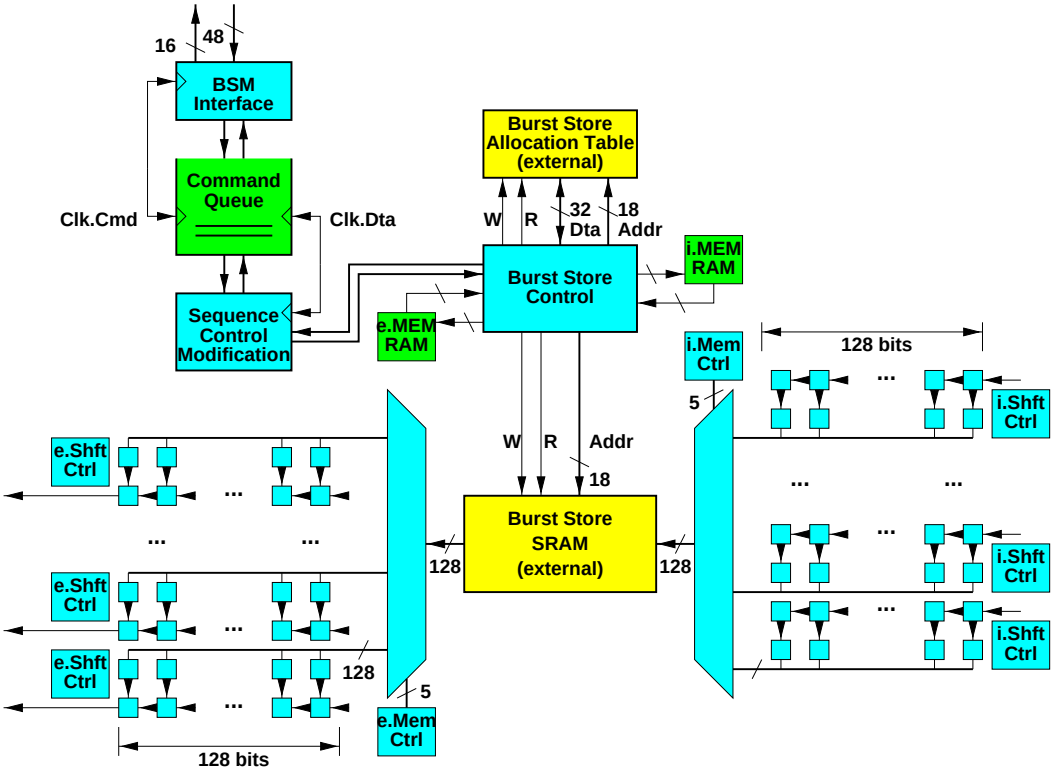

Figure 3: Burst Storage Unit Controller Block Diagram

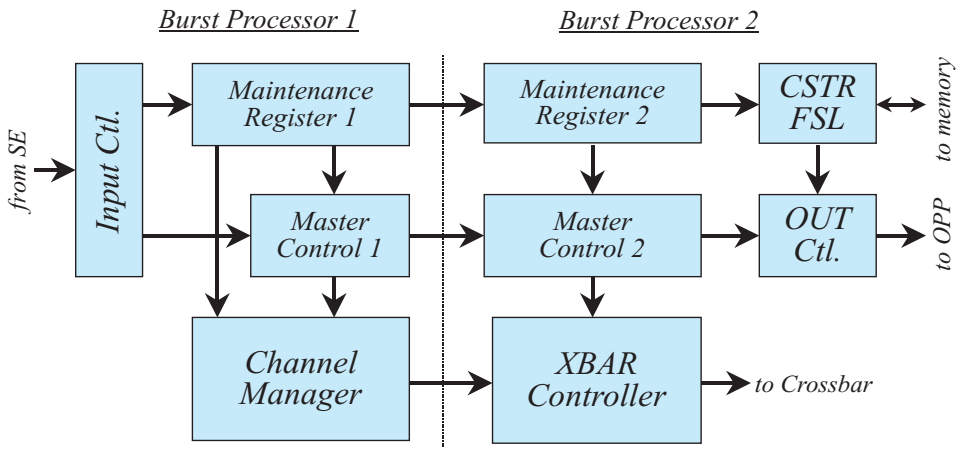

Figure 4: Phase 1 Burst Processor

implementations. Figure 2 shows a single BSU (there are nine in the prototype), its associated memories and its connections to the crossbar and Burst Storage Manager (BSM). Figure 3 is a block diagram of the BSU FPGA. Arriving data enters through one of 32 shift registers and is then multiplexed through to the memory interface where it is stored. Departing data passes through a similar data path.

- *Burst Processor* (BP). The BP is the most important single component of the burst switch. Each BP is responsible for managing 31 outgoing channels from the crossbar. It maintains a schedule for those outgoing channels, and assigns incoming bursts to places in the schedule, using the information it receives in *Burst Header Cells* that come to it through the ATM Switch Element (ASE). The BP also communicates with the Burst Storage Manager (BSM) through a local control ring and has connections that can be used to communicate with upstream and downstream neighbors in a multistage configuration.

  Figure 4 shows the organization of the Phase 1 BP. It is being implemented using a pair of FPGAs. Each has an external memory that can be used for either data storage or for control information. The Phase 1 BP logic includes a horizon link scheduler and a crossbar controller for managing the creation and removal of connections in the crossbar at the appropriate times. The Phase 1 BP logic has been designed, synthesized, simulated, placed and routed. Timing verification shows that it will run at the designed clock rate (50 MHz). The logic uses about 30% of the logic blocks in the two FPGAs.

  Figure 5 shows the Phase 2 BP. It will use the same physical hardware as the Phase 1 BP. We will merely re-program the FPGAs to provide the required expanded functionality. The only completely new blocks are the ring interfaces that connect to the control ring. The control ring has been split into two parts. One is for communicating with the Burst Storage Manager, and the other enables the BPs within a BSE to exchange status information. Only the first part is needed in Phase 2. The Phase 2 BP maintains many of the same components as in Phase 1, but significant changes are needed to the Channel Manager and Master Control 1. The modified channel manager generates a storage request, if an arriving burst cannot be directly switched through to its output. On receiving a reply, it either completes scheduling

of the selected outgoing channel (if the BSM accepts the storage request) or cancels the scheduling operation. While the BP is waiting for the reply, the tentatively selected channel is unavailable for selection by other bursts.

- *Burst Storage Manager* (BSM). The BSM schedules the storage of bursts in the BSU. This component is not required in Phase 1, but the design is now under way in preparation for Phase 2. A block diagram is shown in Figure 6. The physical hardware configuration of the BSM is identical to the BP; just the programming of the FPGAs is different. The BSM requires separate channel managers for its input and output interfaces. In addition, it has a controller to manage the storage within the BSU.

  For Phase 2, we have decided to use a simplified version of the general storage management data structure that we have developed. It involves a differential search tree, but we allocate storage to a burst from the time a burst header cell is received, rather than waiting until the burst arrives. There is little performance penalty incurred by this, in systems where there is only a small variation in the time between arrival of a Burst Header Cell and arrival of the corresponding burst. Section 3.1 provides further details on this.

- *Time Stamp Chip* (TS).

  The TS chip adds a system-wide timestamp to arriving BHCs and provides delay compensation on both the input and output sides of the system. On the input side, this is intended to enable compensation of known variable delays associated with different channels (in a system with WDM links, such delay variations are caused by the wavelength dependence of the speed of light). On the output side, it compensates for varying delays that BHCs experience when passing through the system. The TS also converts between conventional time units used on the external links and internal time units based on the switch's internal clock frequency. This allows various internal components to perform timing in terms of clock ticks and reduces the number of components that require precise timing calibration.

  A block diagram of the TS chip is shown in Figure 7. It contains two main data paths, an *ingress path* and an *egress path.* It also has a separate pair of interfaces used for control purposes. The TS chip is being implemented in an FPGA. The logic has been designed, simulated, placed and routed and the device is expected to run at the required clock rates (125 MHz for the interface to the transmission circuits and 50 MHz for other parts).

- *ATM Interface Module.* The ATM interface module is an IO card that allows data to be received from an ATM switch and converted into a burst that is suitable for transmission through a burst switching network. Details of the AIM were given in a previous progress report [11].

- *ATM Switch Components.* Three components from Washington University Gigabit ATM switch are being used within the burst switch prototype. The next section describes progress on the 160 Gb/s configuration of that switch that is now being assembled. It includes descriptions of the ATM components being used in the prototype burst switch, and their status.

- *Transmission Interfaces* (TI). Transmission formatting will be provided using quad gigabit serial link components made by AMCC. Each of these components has four gigabit transmitters

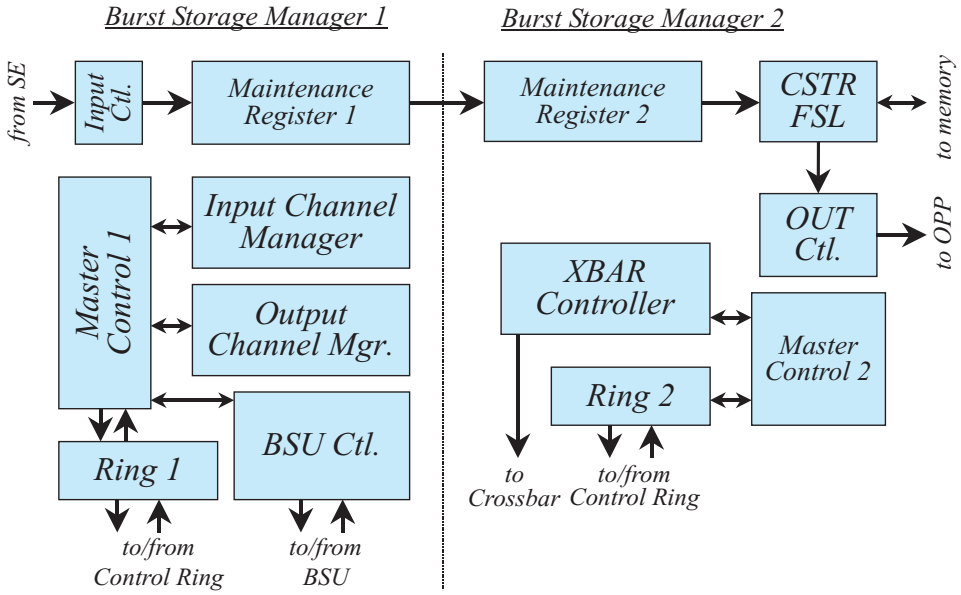

Figure 5: Phase 2 Burst Processor

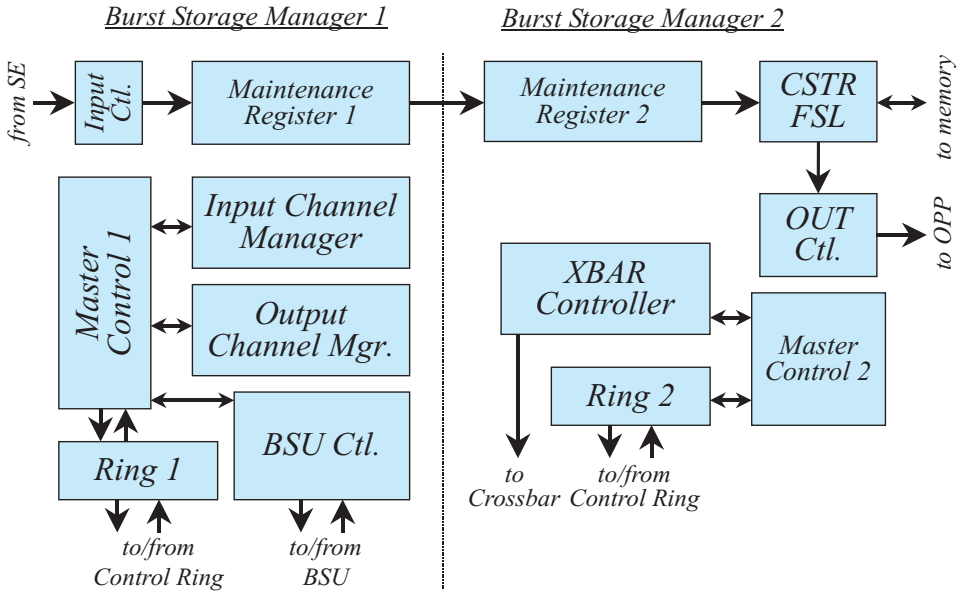

Figure 6: Burst Storage Manager

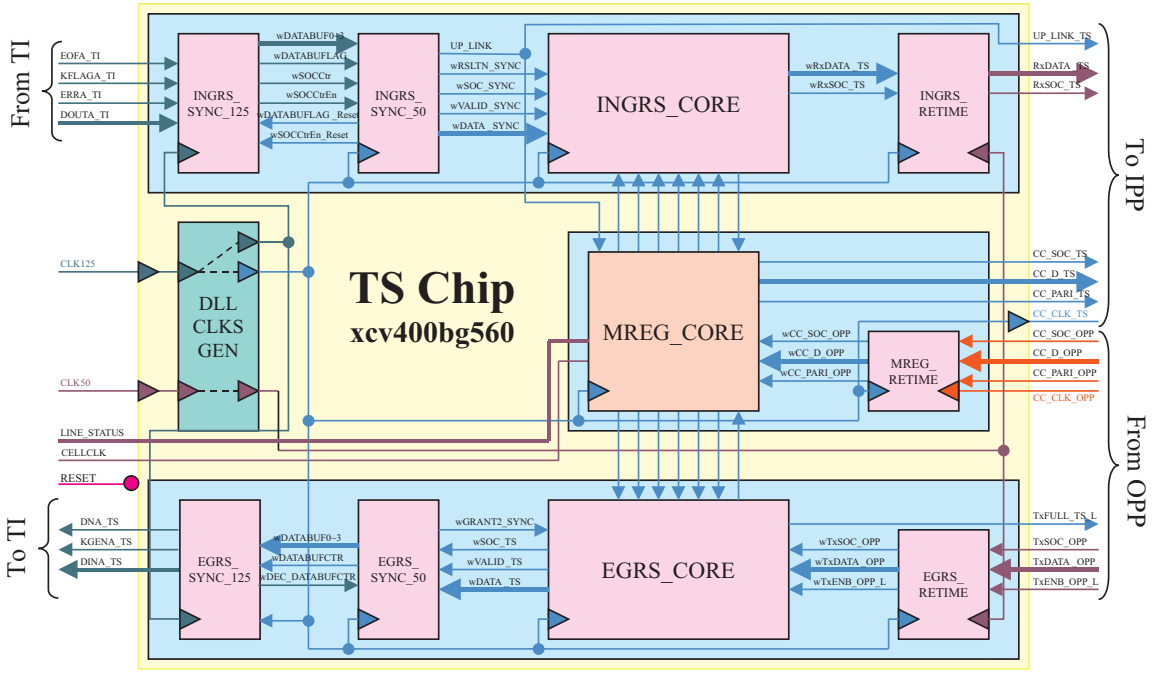

Figure 7: Timestamp Chip

and four gigabit receivers. The chips encode the data for transmission using a 8B/10B line code, decode it on reception and recover clock from the received bit stream. Each IO module will have eight of these components. Samples of these components have been obtained and evaluated in a test fixture.

- *Optoelectronics* (OE). The optical interfaces will be implemented using VCSEL array devices that handle 12 serial data channels at data rates of 1.25 Gb/s and distances up to 500 meters. The specific devices that we plan to use are the Siemens Parallel Optical Link components (PAROLI). Samples of these components have been obtained and evaluated in a test fixture.

- *PC Boards and Physical Design.* The high level design of the PC boards required for the burst switch has been completed (IO Board, BSE Datapath Board, BSE Control Board, Miscellaneous Board and Backplane) and a plan for the physical packaging design has been developed. Figure 8 shows the planned physical packaging for the prototype. It will consist of a single chassis with 18 printed circuit boards. Extensive simulations have been carried out to evaluate signal integrity issues for signals passing between boards through the backplane.

2. 160 Gb/s ATM Switch

The following paragraphs summarize status and progress on the various components being developed for the 160 Gb/s ATM switch being constructed as part of this project. Several of these components are common with the burst switch. Figure 9 shows the overall structure of the prototype and details the location of each component in the overall architecture.

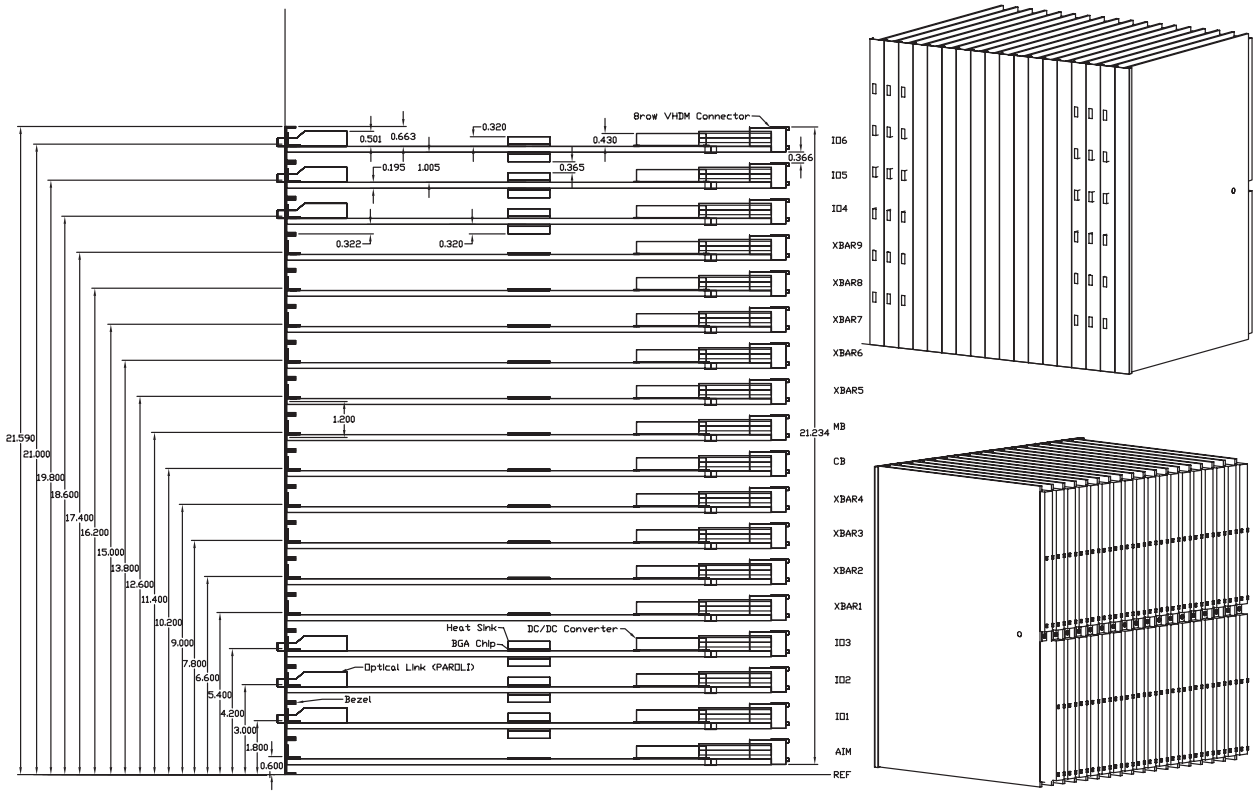

Figure 8: Packaging for Burst Switch Prototype

- *ATM Switch Element* (ASE). This chip is a revised version of a chip that was developed in an earlier project. The new chip implements four priority classes, doubles the cell buffering of the previous chip and corrects timing flaws that limited the operational frequency of the original chip. The ASE is being implemented in a .35 micron ASIC process. The chip is now in the final stages of layout and will be fabricated in the first quarter of 2000. Figure 10 shows the layout, as displayed by the CAD tools.

- *ATM Input Port Processor* (IPP). The IPP is a modified version of a component developed for an earlier project. The new chip provides a larger VPI/VCI lookup table (4096 entries instead of 1024) and allocates those entries more flexibly. It also implements features for reliable multicast and provides more extensive support for traffic monitoring. The IPP is being implemented in a .35 micron ASIC process. The chip is in the final stages of layout and will be fabricated in the first quarter of 2000. Figure 10 shows the layout of the core of the circuit (excluding pads), as displayed by the CAD tools. The shaded boxes are SRAM memory blocks.

- *ATM Output Port Processor* (OPP). This chip was developed in an earlier project. The required die (fabricated in a .7 micron ASIC process) are on hand. These chips are being packaged in a ball grid array (rather than a pin grid array) to make them compatible with other components in the system.

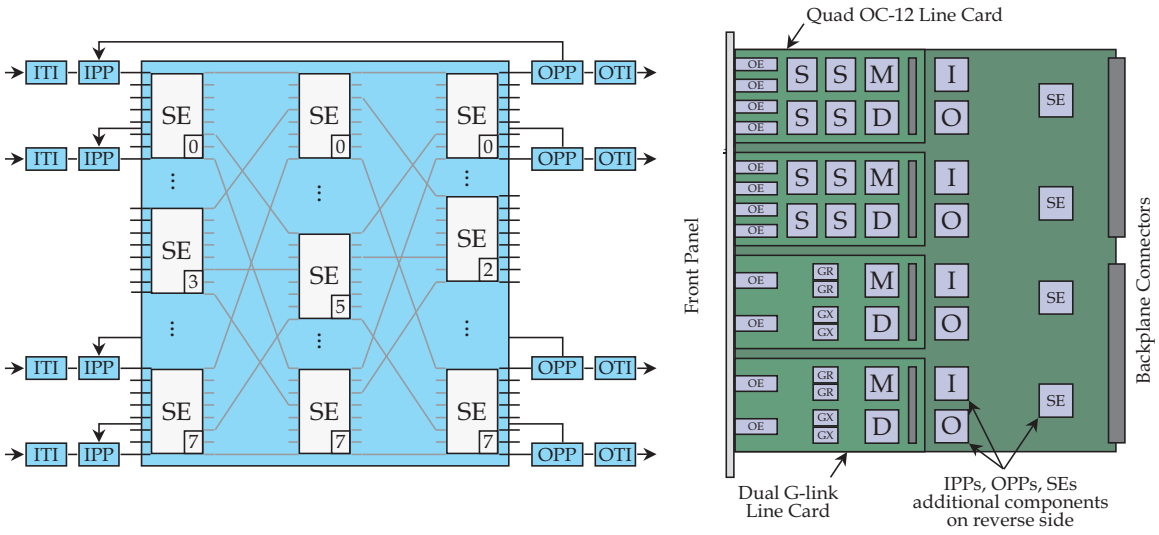

Figure 9: 160 Gb/s ATM Switch

- *Dual G-link Line Card.* This card multiplexes a pair of 1 Gb/s links onto a single core switch port, using an FPGA to perform the input-side multiplexing and output-side demultiplexing. A prototype of this card has been implemented and tested in an existing switch. The design needs to be modified slightly to match the physical packaging of the new switch, but otherwise it is complete.

- *Quad OC-12 Line Card.* This card multiplexes four OC-12 links onto one switch board. The FPGAs to do the required multiplexing and demultiplexing functions have been designed and simulated. The PC board layout is now in progress. We expect to fabricate the board in the first quarter of 2000.

- *OC-48 Line Card.* This card terminates a single OC-48 link. Although not part of the original project plan, we believe it will be feasible to include it in the project, given the recent availability of integrated OC-48 framer components from AMCC.

- *PC Boards and Physical Design.* The high level design of all the PC boards required for the system has been completed (IO Board, Center Stage Board and Backplane). We have eliminated the separate timing generation board, opting instead to place the timing circuitry on each of the center stage boards, but only connect these circuits to the backplane in one slot. The physical packaging design is completed. Figure 12 shows the overall physical arrangement, including two 3-D views. The most complex element of the design is the IO board, which is shown in Figure 13. Each IO board has eight switch element chips, eight Input Port Processors and eight Output Port Processors. In addition, each hosts up to eight line cards that carry the transmission components. These are mounted using mezzanine connectors, on opposite sides of the main board and connect to external links through the front panel.

  Figure 14 shows a close-up of a portion of the layout of the IO board. The four chips at the top are four of the eight Switch Element chips. The IPP and OPP chips are near the center of the figure and a portion of two of the eight line cards can be seen at the bottom. Several

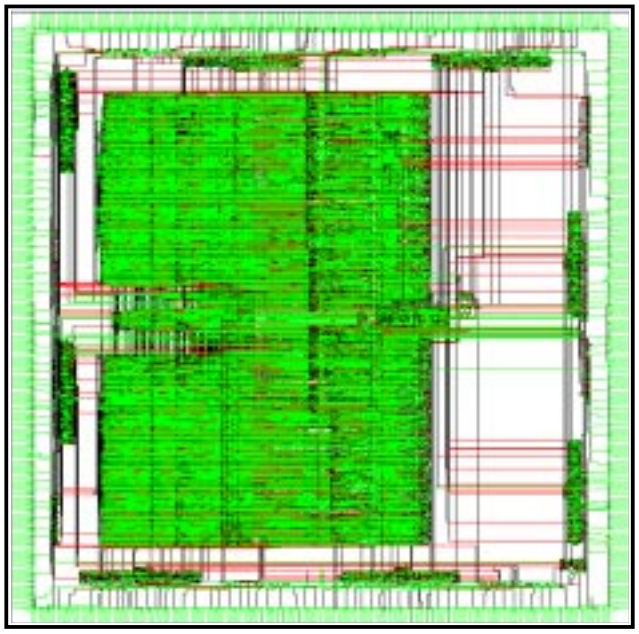

Figure 10: ATM Switch Element

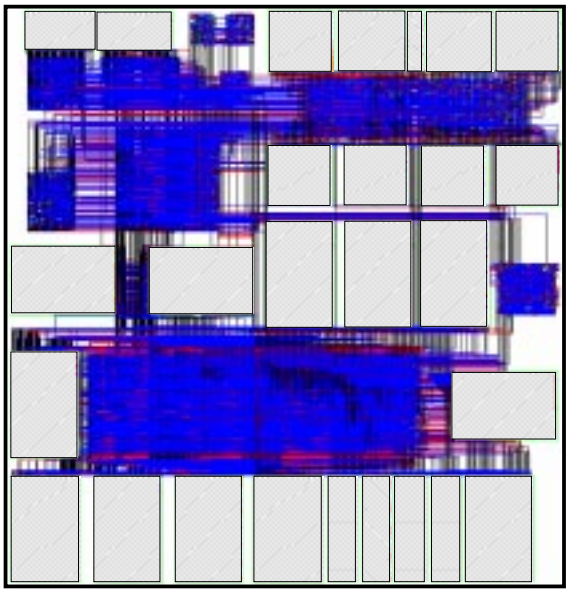

Figure 11: ATM Input Port Processor

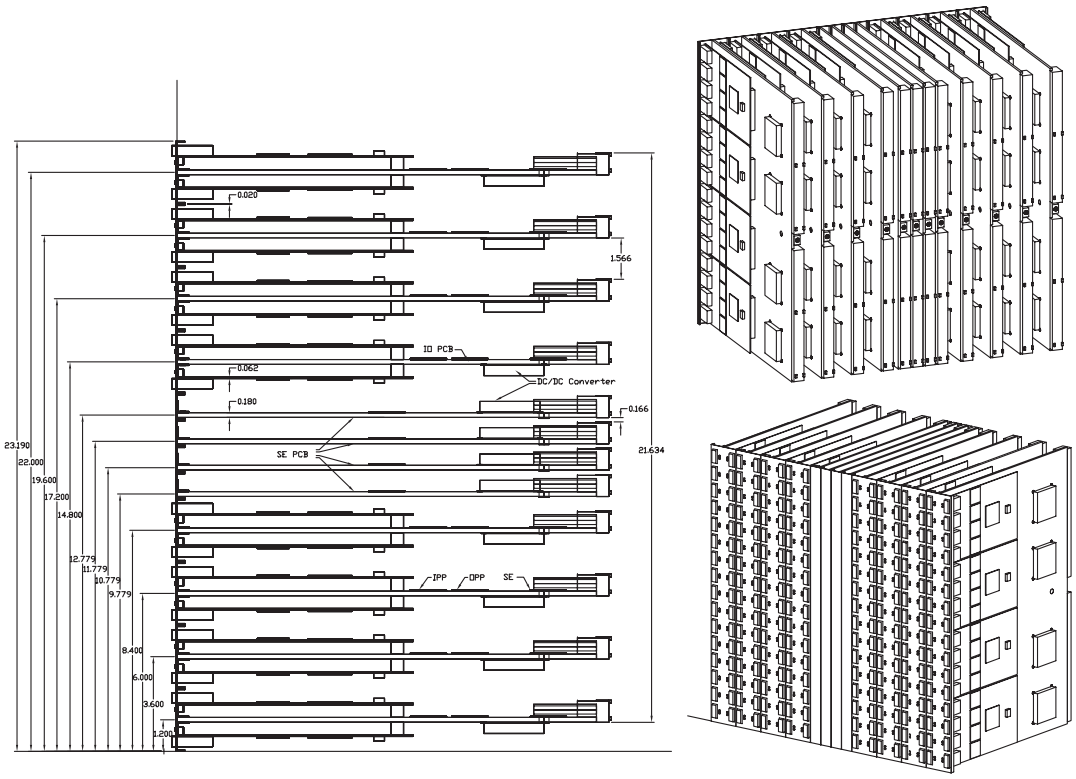

Figure 12: WUGS-160 Physical Packaging

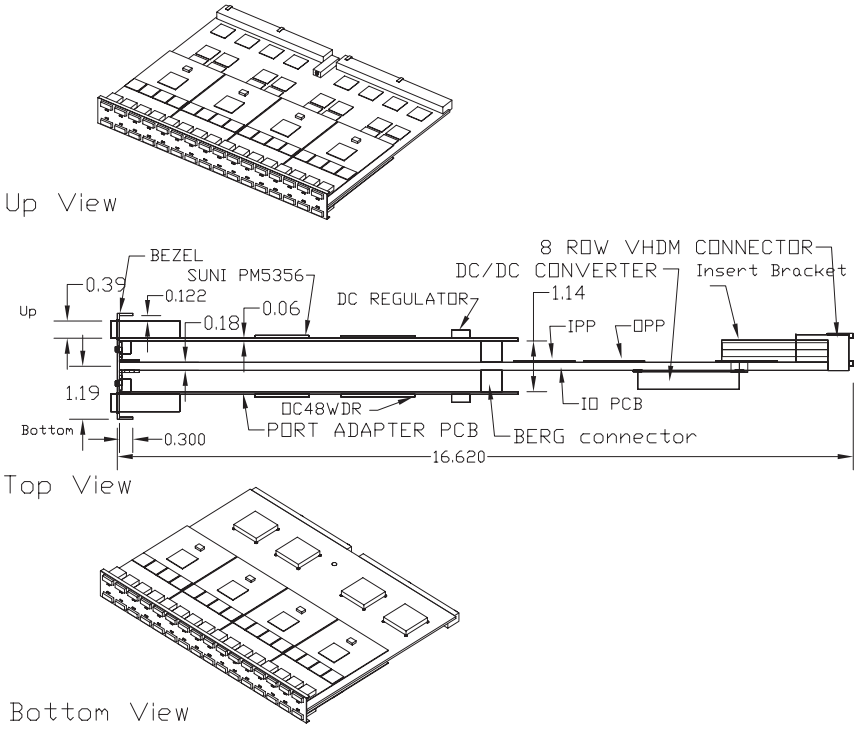

Figure 13: WUGS-160 IO Board Packaging

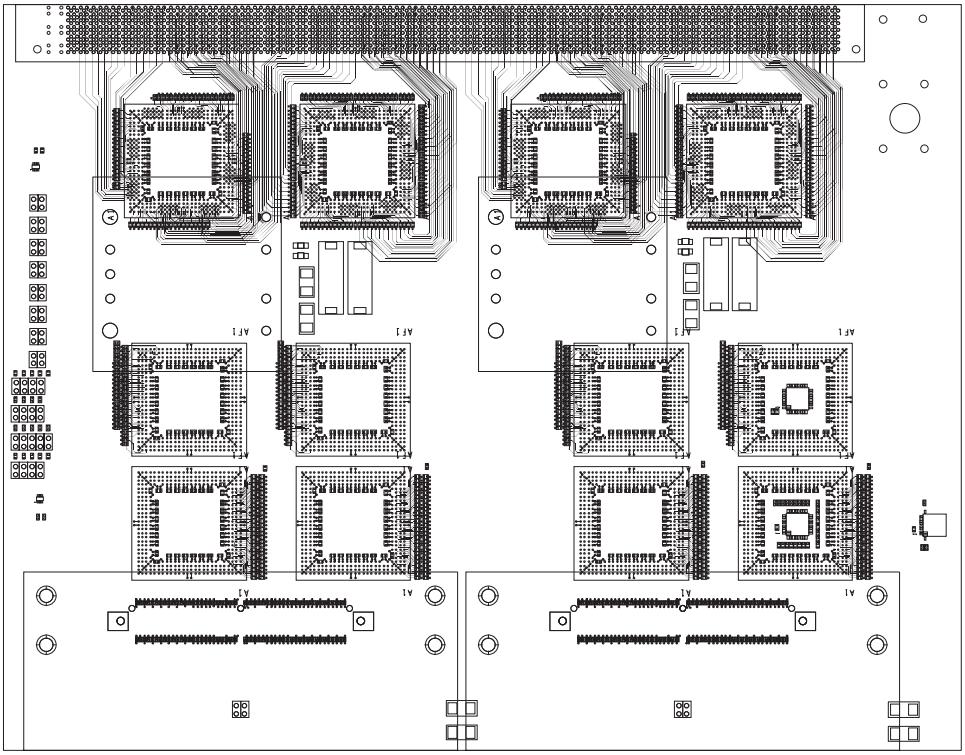

Figure 14: WUGS-160 IO Board Partial Layout

power converters are mounted on the under side of the board. These convert the 48 volt power distributed through the backplane to the voltages required on the board. The power modules were chosen to accommodate any combination of line cards, some of which have different requirements for different voltage levels. Only a portion of the routing has been shown, to allow the various parts to be seen clearly.

3. Architectural Studies

There are a number of architectural issues that we are continuing to study. This section summarizes the current activities.

3.1. Simplified Data Structure for Burst Storage Management

In previous reports [6, 7], we have introduced a general data structure that is useful for managing storage in a burst switching system. This data structure, called a *differential search tree*, keeps track of planned future memory usage and allows one to quickly check if a soon-to-arrive burst can fit within the available memory. It also supports rapid updates to the data structure, once a burst has been accepted. The time required to perform each operation is logarithmic in the number of stored

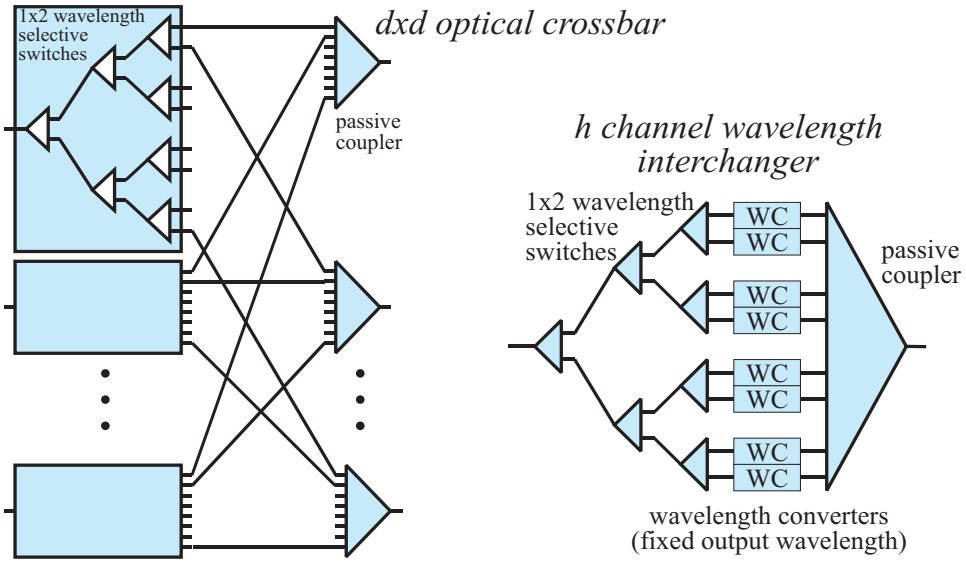

Figure 15: Building Larger Optical Switch Components from Wavelength Selective Switches

bursts. By using tree nodes with a large fanout, we can keep the depth of the tree small. However, this requires a wide interface to memory, making it most effective when the memory to store the data structure can all fit within in the BP chip.

In the initial burst switch prototype, we attempt to keep the variation in offsets between BHCs and bursts relatively small. Ideally, this variation is smaller than the length of the smallest burst we expect to process. In this situation, the horizon link scheduling algorithm (described in previous reports) works very effectively. By exploiting this property in the Burst Storage Manager, we can significantly simplify the differential search tree.

The general differential search tree allows memory to be allocated to an arriving burst from the time the burst is *scheduled* to arrive until the time it is scheduled to leave. When the variation in offsets is small, we can allocate memory from the time the *BHC* arrives until the time the burst is scheduled to leave, without any loss of efficiency. The reason for this is that with small offset variation, later arriving bursts can not use the "pre-burst gap" preceding a previously scheduled burst that has not yet arrived.

If storage is allocated in this way, then the time schedule for the burst store is monotonically non-increasing. Thus, the current time is always the time of maximum scheduled memory utilization. As a result, we can test if an arriving burst will fit, merely by considering the memory usage at the current time. This allows us to eliminate the Δmax field from the data structure. Insertion of a burst is also simplified, since only one probe of the differential search tree is required.

3.2. Using Wavelength Selective Switching for Burst Switching

As reported earlier [11], the design of optical datapaths for burst switching systems is currently limited by the cost and complexity of wavelength selection and conversion. While appropriate

components have been demonstrated in the laboratory, they are complex, expensive and don't seem to be well-suited to the high levels of integration needed to make them cost-effective in switching applications, where large numbers of components are needed.

Recently, we have been considering an alternative approach, using wavelength-selective switching components. A wavelength selective switch allows WDM signals on each of a number of inputs to be routed to each of a number of outputs. Moreover, each wavelength carried on an input fiber can be routed independently of other wavelengths. While wavelength-selective switches can be constructed by demultiplexing signals, switching them in space, then re-multiplexing, the resulting components are complex and expensive. Liquid crystal switching devices [2] allow a more elegant solution, since the basic element is a 2×2 device that can switch different wavelengths independently. This enables the construction of more complex building blocks. For example, a $d \times d$ crossbar can be constructed using $d(d-1)$ 1×2 switches together with d passive optical couplers (d inputs, 1 output). One can also construct an h channel all-optical wavelength-interchanger, using $h - 1$ 1×2 switches, h interferometric wavelength converters and a passive coupler (see Figure 15. These two building blocks correspond directly to the time-multiplexed-switches and time-slot-interchangers used as building blocks in electronic TDM switching systems. Thus, using these components, one can construct large optical WDM circuit switches using architectures directly analogous to those used in TDM systems and having well-understood performance characteristics. For example, a fully-nonblocking optical circuit switch with 64 fibers and 64 channels per fiber can be constructed using wavelength interchangers on the input and output side, and a central Clos network constructed from wavelength-selective switches (8×16, 8×8, 16×8). Such a system requires just 4 1×2 wavelength-selective switching components and two wavelength converters for each output wavelength. If each wavelength carries OC-192 signals, such a system will have a capacity of 40 Tb/s and could possibly be packaged in fewer than 10 equipment racks.

The intrinsic advantage of the liquid crystal switching devices is that they handle switching in the wavelength domain within the confines of the basic switching component. This allows the number of wavelengths to be increased by expanding the basic device capabilities, something that is relatively straightforward and inexpensive. The alternative of demultiplexing signals, switching them in space and re-multiplexing requires more discrete components, as the number of wavelengths grows. This makes it poorly matched to the trends in WDM systems that are moving into the hundreds of wavelengths per fiber.

In burst switching systems, wavelength-selective components can provide similar advantages. In the optical datapath design described in [11], the use of wavelength-selective switches has the potential for drastically reducing the number of discrete components needed. In particular, the design in [11] requires a large number of wavelength selectors, with a single input fiber carrying h wavelengths, a control input for selecting one of these wavelengths and a single output. Each such wavelength selector requires an AWGN for demultiplexing the input signals and h SOAs, acting as gates. Currently, the SOAs must be implemented as discrete devices and there seems no near-term prospect for integrating large numbers of them together. A system using wavelength-selective switches requires approximately the same number of 1×2 switches as the design in [11] requires wavelength selectors. Thus, we can conveniently compare these alternative approaches by examining the relative cost and complexity of these two components. In the wavelength-selective switch, we have a single component, no matter how many wavelengths are used. While switches with many wavelengths must be physically larger and require more control electronics, there are

no intrinsic limits on our ability to integrate the required components. The net result is a large decrease in the number of discrete components needed to build a large system (perhaps as much as two orders of magnitude) and a related decrease in cost.

Unfortunately, liquid-crystal switches are currently far too slow to enable their use in burst switching systems handling short data bursts. While there are prospects for substantial improvements in switching speeds (possibly down to as little as 100 ns), there have been no laboratory demonstrations of such devices. It appears likely that new materials will be needed to enable wavelength-selective switching at the speeds needed for practical burst switching systems. Nevertheless, the possibilities raised by the current generation of devices points the way toward alternative datapath architectures that could offer substantially better cost-performance. Indeed, for circuit switching applications, even the current devices offer intriguing possibilities for very high capacity systems.

3.3. Improving Synchronization

One of the key performance issues for burst switching systems is controlling the timing skew between a Burst Header Cell and its corresponding burst. There are two primary sources of skew. One is the variation in the speed of light, as a function of wavelength. Most of this variation can be compensated if the physical length of the link is known. However, the compensation is imperfect and requires that independent adjustments be made for each incoming link. We can improve the quality of compensation and eliminate the manual adjustments required by providing two extra wavelengths on each fiber that carry periodic timing pulses. If the timing channels are carried on the shortest and longest wavelengths, they will enable automatic determination of how much compensation is required for each of the other wavelengths (the timing pulses must be far enough apart in time to prevent errors in interpretation – 100 μs is sufficient). In fact, the timing channels can also be used to carry burst header cells, so the added bandwidth required for the timing information is negligible. We expect that this approach should enable timing errors to be reduced to just a few nanoseconds per hop.

3.4. IP Routing in Burst Switching Networks

The current prototype burst switch uses ATM-style routing. That is, arriving bursts are switched to an output based on a Virtual Circuit Identifier carried in the Burst Header Cell. To enable more effective support of IP traffic within burst networks, it is desirable to include support for IP-style routing lookups. The fundamental operation in IP routing is a *best matching prefix* lookup. A routing table is essentially a list of (prefix, next hop) pairs and the address lookup process involves finding the prefix that matches the most bits of an arriving packet's address (or an arriving burst's address), and then returning the corresponding next hop.

A number of fast IP lookup algorithms have been proposed in recent years. As part of another project (on Active IP Routing), we are currently prototyping a particular fast lookup algorithm in an FPGA [1]. We believe this algorithm can be implemented to allow it to process over 10 million lookups per second, making it fast enough to process a lookup for every BHC that can be received on a control channel of our prototype burst switch. We anticipate that it may be possible

to demonstrate this capability by substituting a larger FPGA in place of the TS chip that is part of the current architecture. We are currently evaluating whether this is feasible and whether the incremental cost of the larger FPGA is worth the added capability that will be provided.

3.5. Towards Higher Performance Burst Processing

Our prototype burst switch uses a pair of FPGAs to implement the Burst Processor. These chips provide a total of about 200K gates and 250 Kbits of SRAM and can operate at clock frequencies of 50 MHz. While these numbers show the impressive strides that FPGAs have made in recent years, it's important, when projecting to commercial implementations to keep in mind the substantially greater capabilities of Application Specific ICs. Using a .18 μm ASIC process, one can implement chips with over 2 million gates and 2 Mb of SRAM with clock rates of 200 MHz or more. A single such chip could replace the pair of FPGAs used in the current design and could handle both more channels and higher data rates per channel. By extrapolating from our current design, it appears likely that with current ASIC processes, one could implement systems with 64 wavelengths per link and 10 Gb/s channels. If the chip processes one BHC every 40 ns, it can support average burst lengths as small as 3.2 KB. Architectural improvements could likely improve this by another factor of two. Similar improvements can be projected for the Burst Storage Manager. As electronics moves to .12 μm geometries in the next few years, clock rates will increase to at least 300 MHz and achievable gate complexities and SRAM densities will double. These improvements will make it possible to handle average burst lengths of less than 1 KB.

Improvements in electronics technology will also support larger switching components in the control sections of burst switches. Our prototype uses an eight port switch chip with a raw throughput of about 10 Gb/s. These chips were originally designed in a .7 μm ASIC process. Using current .18 μm technology, one can construct switch components with a throughput of over 50 Gb/s and this is expected to double by 2002, as ASIC processes continue to migrate to smaller geometries, and as more sophisticated signalling techniques are used for inter-chip communication. Such improvements would allow the control section of a Burst Switch Element to scale up to 32 inputs and outputs. Assuming that the corresponding datapath could be constructed, the resulting BSE would have an aggregate throughput of 20 Tb/s (assuming 64 OC-192 channels per fiber).

References

[1] Eatherton, Will. *Hardware-Based Internet Protocol Prefix Lookups.* Washington University Electrical Engineering Department, MS thesis, 5/99.

[2] Patel, Jay and Y Sliberberg. "Frequence Selective Optical Switch," U.S. Patent 5414540. See also `http://www.phys.psu.edu/faculty/PatelJ`.

[3] Turner, Jonathan S. "Terabit Burst Switching," Washington University Technical Report, WUCS-98-17, 1998.

[4] Turner, Jonathan S. "Terabit Burst Switching Progress Report (12/97-3/98)," Washington University Technical Report, WUCS-98-16, 1998.

[5] Turner, Jonathan S. “Terabit Burst Switching Progress Report (3/98-6/98),” Washington University Technical Report, WUCS-98-22, 1998.

[6] Turner, Jonathan S. “Terabit Burst Switching Progress Report (6/98-9/98),” Washington University Technical Report, WUCS-98-30, 1998.

[7] Turner, Jonathan S. “Terabit Burst Switching Progress Report (9/98-12/98),” Washington University Technical Report, WUCS-98-31, 1998.

[8] Turner, Jonathan S. “Terabit Burst Switching,” *Journal of High Speed Networks*, vol. 8, no. 1, 1999.

[9] Turner, Jonathan S. “WDM Burst Switching,” *Proceedings of INET,* San Jose, CA, 6/99.

[10] Turner, Jonathan S. “WDM Burst Switching for Petabit Capacity Routers,” *Proceedings of Milcom,* Atlantic City, NJ, 11/99.

[11] Turner, Jonathan S. “Terabit Burst Switching Progress Report (1/99-6/99),” Washington University Technical Report, WUCS-99-21, 8/99.