Terabit Burst Switching
Progress Report (12/98–6/99)

Jonathan S. Turner
jst@cs.wustl.edu

WUCS-99-21

August 8, 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 (12/98–6/99)

Jonathan S. Turner
jst@cs.wustl.edu

This report summarizes progress on the Terabit Burst Switching Project at Washington University for the period from December 15, 1998 through June 30, 1999.

1. Prototype Burst Switch Progress

The following paragraphs summarize status and progress on the various components being developed for the protytpe 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 provides an aggregate bandwidth of 256 Gb/s, which is obtained using a bit-sliced organization with eight parallel planes for carrying the data. The crossbar is to be constructed from individual chips that implement a 256×128 crossbar. Pairs of such chips are combined to produce a 256 × 256 bit-slice. An asynchronous protocol has been designed for sending data to and from the crossbar. The protocol allows the receiver circuits in the crossbar to automatically adjust for clock and data skew. Receiving data is organized as a sequence of 16 bits preceded by two start bits (one high, one low) and followed by a stop bit (low). Each receiver circuit (there are 256 on the chip) uses the start bits to determine which of 3 available clock phases provides the best sampling point. To compensate for asymmetries in rise and fall times, the circuit makes separate decisions for data bits following rising edges and falling edges. The subsequent data bits in each word are then sampled using the selected clock phases. The frequency of the received data stream is 150 MHz (peak data rate is just over 125 Mb/s accounting for the overhead of the start and stop bits).

  Our initial plan was for the crossbar to be implemented in a Laser Programmable Gate Array process (LPGA). Unfortunately, the supplier (ChipExpress) ultimately proved unable to provide the technology in a timely fashion. Consequently, the design has been retargeted for an ASIC implementation in .35 micron CMOS. The circuit design has been specified in Verilog, synthesized and simulated. Floorplanning for the layout has been completed, and an initial layout is now in progress. The core of the chip is expected to consume less than 70 mm^2, although pin constraints will force the use of a substantially larger die.

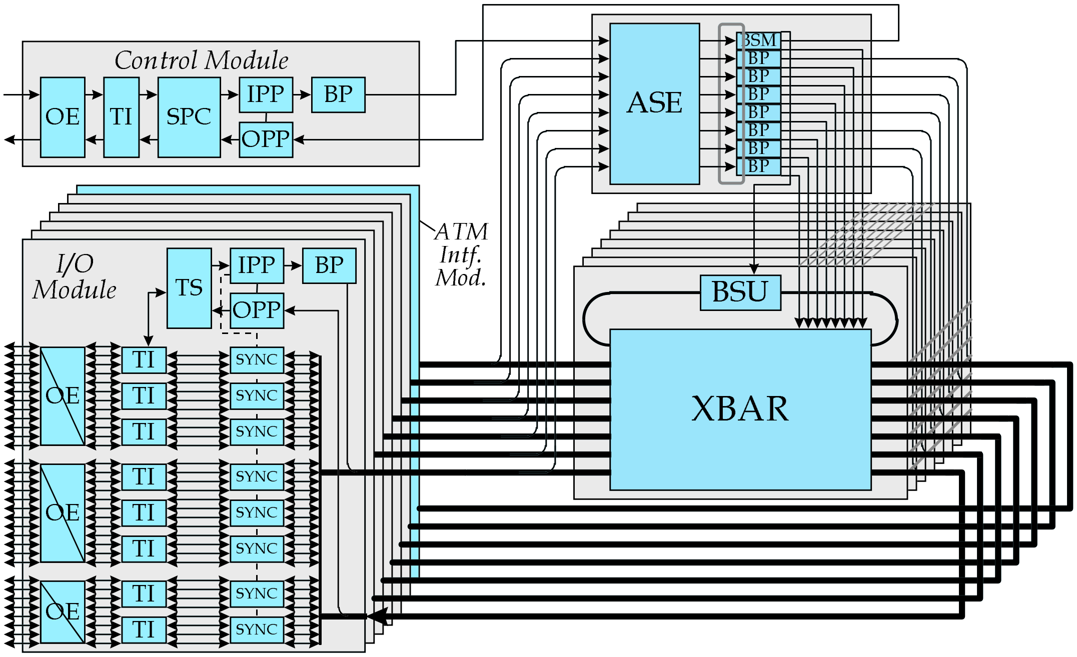

Figure 1: Prototype Burst Switch

- *Synchronization Chip* (SYNC). The SYNC chip inserts a delay into the data path for data coming from the external link. Each chip handles four channels of data (at 1 Gb/s per channel) and provides up to 50 μs of delay (the amount of delay can be configured). Since the SYNC chip sends data to and receives data from crossbar chips, it implements the same protocol as the crossbar and has 32 serial receiver circuits that implement the protocol. In addition, it connects to the transmission interface chips that convert the incoming serial bitstream to parallel form.

  Like the crossbar, the SYNC chip was to be implemented in an LPGA but has now been retargeted to an ASIC process. The logic has been defined and simulated, but layout has not yet begun.

- *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 a fifth memory chip which will hold linked list pointers to enable it to manage the memory in a flexible fashion. It is to be implemented using an FPGA, to minimize risk and provide flexibility for alternative implementations. The BSU is in the design specification phase. It is not needed for the first phase of the prototype implementation.

- *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.

The BP is being implemented using a pair of FPGAs. Each FPGA has an external memory that can be used for either data storage or for control information. In the first phase of the project, the BP logic will include a horizon link scheduler and a crossbar controller for managing the creation and removal of connections in the crossbar at the appropriate times. In later phases of the project, we plan to replace this logic with more sophisticated versions, designed to provide higher levels of performance. 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, leaving sufficient space for the more complex logic planned for later phases.

- *Burst Storage Manager* (BSM). The BSM schedules the storage of bursts in the BSU. This component is not required in phase 1, but has been the subject of much of the architectural studies to date. For phase 2, we plan a relatively simple horizon scheduler with partitioned memory areas for each BP. In phase 3, we expect to implement a more complex scheduler, capable of higher performance under heavy loading conditions.

- *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.

  The TS chip is to be 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. The architectural definition of this card is now in progress, and is described in Section 2.

- *ATM Switch Element* (ASE). An ATM Switch Element is used within the Burst Switch Element to route Burst Header Cells to the appropriate Burst Processor. This chip is a revised version of a chip that was developed in an earlier project. The new chip implements four priority classes, allowing Burst Header Cells to be given high priority treatment relative to normal ATM data cells that can also be carried through the system. The new chip also doubles the cell buffering of the previous chip and corrects timing flaws that limited the operational frequency of the original chip.

The ASE is targeted for implementation in a .35 micron ASIC process. The logic has been fully specified in VHDL and the chip has been simulated extensively. Layout will take place in the third and fourth quarters of this year.

- *ATM Input Port Processor* (IPP). The IPP is a modified version of a component first developed for an ATM switch developed in an earlier project. It is being applied in the burst switch to perform input processing (primarily route selection) for Burst Header Cells and ordinary ATM cells.

  The IPP is targeted for implementation in a .35 micron ASIC process. The logic has been fully specified in VHDL, the chip has been extensively simulated and the layout is nearly complete. It is to be fabricated in the third quarter of this year.

- *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. For the burst switch project, 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. We expect the repackaging to be completed in the third quarter of this year.

- *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 and four gigabit receivers. The chips encode the data for transmission using a 4B/5B 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 most of the PC boards required for the burst switch has been completed (IO Board, BSE Datapath Board, BSE Control Board, Timing Generation Board and Backplane) and a plan for the physical packaging design has been developed. Extensive simulations have been carried out to evaluate signal integrity issues for signals passing between boards through the backplane. Particular concern centers on the control signals from the BSE Control Board to the nine BSE Datapath boards. To keep pincount from becoming excessive, it will be necessary to fanout control signals from the Control Board to up to five Datapath Boards on the backplane. Maintaining adequate signal integrity on these lines is challenging, but a design has been developed and simulated that should work well. We expect to complete schematic definition of all circuit boards and pin definitions for all chips in the third quarter. Board layout should begin in the fourth quarter.

2. ATM Interface Module

Burst switching systems are most likely to be deployed in the backbone of existing networks. This makes it is important for burst switches to interface directly to links from existing network

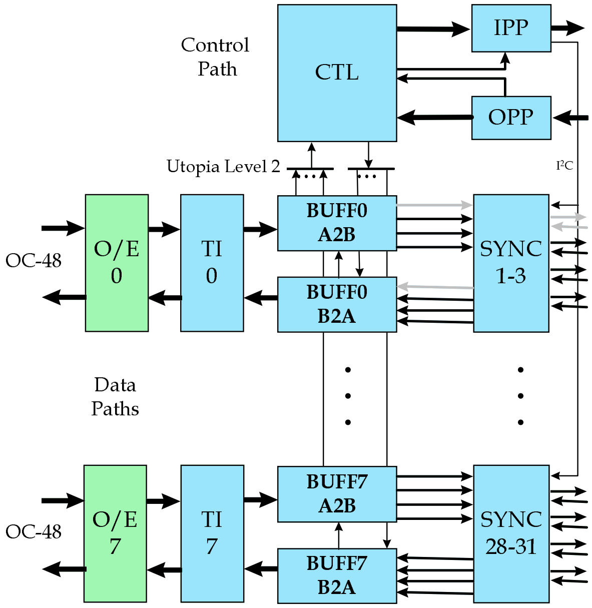

Figure 2: ATM Interface Module

technologies, such as ATM, IP or Ethernet. In the project, we plan to implement an IO module that connects directly to ATM links operating at 2.4 Gb/s. In particular, the IO module will terminate eight such links allowing ATM data to be forwarded transparently through a prototype burst switch (see Figure 2).

Data received on an ATM virtual circuit will be stored in the IO module at the entry to the burst network until either (1) a pre-specified number of cells has been received, or (2) a pre-specified time period has elapsed since the first cell was received. When either event occurs, the cells received at that point are forwarded to the burst switch. That is, the interface formats and sends a Burst Header Cell, followed by the burst itself on one of four available transmission channels. After the burst is forwarded through the network, it arrives at an exit interface where it is to be forwarded to the destination ATM network. At this point, the burst is buffered within the IO interface and forwarded to the ATM network. The forwarded cells are labeled with the appropriate outgoing VCI (typically not the same as the VCI value they had when they left the source ATM network). Cells sent from the exit point can be paced so that the rate of cell forwarding from the exit point matches the rate at which cells were received at the entry point. This keeps the burst network from artificially increasing the burstiness of the data stream.

The ATM Interface Module will be implemented as a circuit board that occupies a slot in the same chassis as the other circuit boards making up the burst switch. It will terminate eight OC48 links and will have the associated transmission interface circuits. In addition, for each of the eight interfaces, there will be a pair of FPGAs, one for the entry processing, one for the exit processing. These will each have sufficient external memory to buffer data on entry and exit. In addition, the board will have a control FPGA which will forward BHCs and normal ATM cells to the control channel of the burst switch. The board will also include a set of SYNC chips and the ATM IPP and

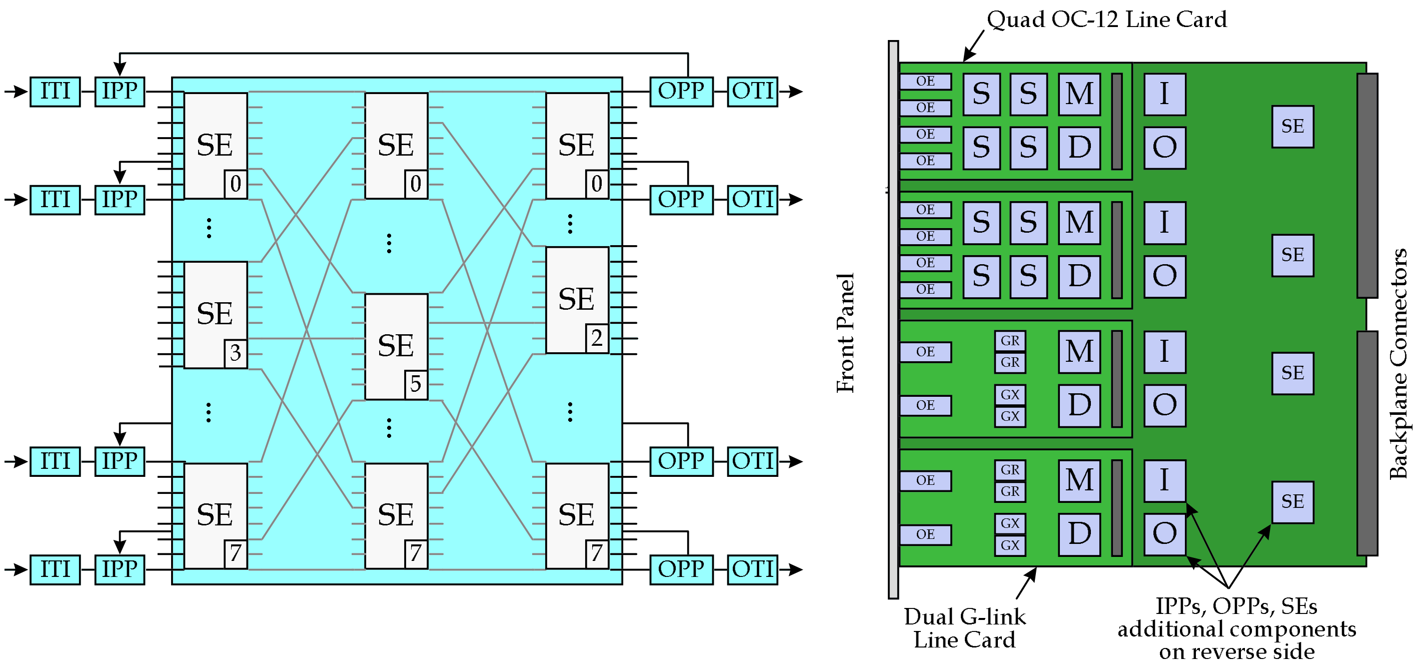

Figure 3: 160 Gb/s ATM Switch

OPP chips required to provide compatibility with the standard IO modules.

3. 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 1 shows the overall structure of the prototype and details the location of each component in the overall architecture.

- *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 targeted for implementation in a .35 micron ASIC process. The logic has been fully specified in VHDL and the chip has been simulated extensively. Layout will take place in the third and fourth quarters of this year.

- *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 targeted for implementation in a .35 micron ASIC process. The logic has been fully specified in VHDL, the chip has been extensively simulated and the layout is nearly complete. It is to be fabricated in the third quarter of this year.

- *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. We expect the repackaging to be completed in the third quarter of this year.

- *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. Due to recent changes in the available OC-12 framer chips, the controller for the board requires some minor modifications which are now being made. We expect to implement a prototype of this board for test/debug purposes in the third quarter and will implement a final version by the end of the year.

- *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. We plan to complete evaluation of the feasibility of the OC-48 card this quarter.

- *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, Timing Generation Board and Backplane) and the physical packaging design has been completed (including a full set of mechanical drawings). Extensive simulations have been carried out to evaluate signal integrity issues for signals passing between boards through the backplane. The pinouts for the modified chips have all been fully specified, to allow layout to proceed. We expect to complete all board-level schematics this quarter, with layout proceeding through the fourth quarter.

4. Architectural Studies

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

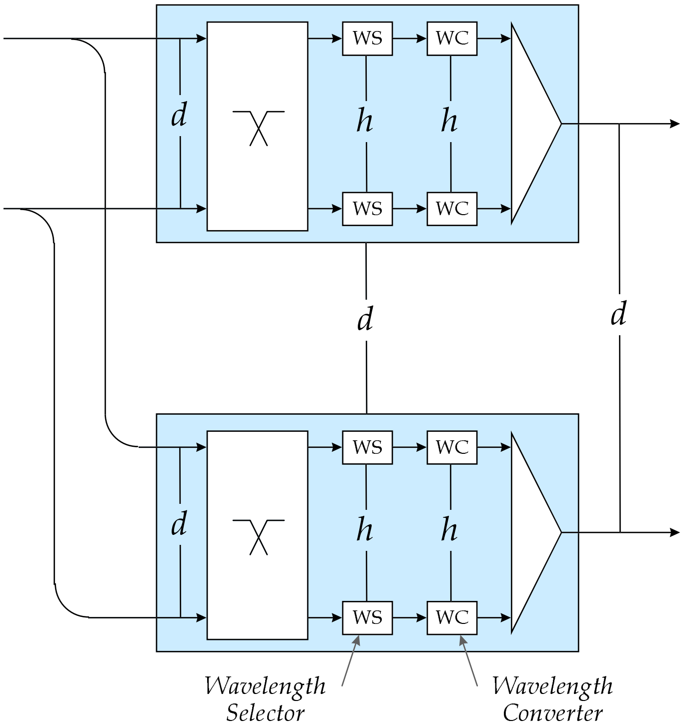

Figure 4: Optical Crossbar for Burst Switch Element

Optical Datapath Complexity Analysis

The original motivation of burst switching was to enable a large portion of packet switching systems to be implemented entirely in optics. We expect that ultimately, optical technology will be able to implement ultra high capacity routers in a more cost-effective way than can be done with electronics. We have done a detailed complexity analysis of an optical datapath design, in order to obtain a better understanding of what factors drive the complexity of burst switches with optical data paths. This, together with comparable data on large-scale electronic routers, allows us to formulate cost targets for key optical technology components.

Previous reports have described a multistage interconnection network for large-scale burst switching systems. The key component in this architecture is the Burst Switch Element (BSE), which has d inputs and d outputs, each with h WDM channels. The datapath of a BSE consists of a $d \times d$ space-wavelength crossbar, capable of switching individual wavelength channels from input links to output links. The switching operation includes wavelength conversion, allowing an input signal to leave using a different wavelength than the one on which it entered. An implementation of such a crossbar is shown in Figure 4. It consists of d separate sections, one for each output. Each section includes a $d \times h$ optical space switch, followed by h wavelength selectors and h wavelength converters. The outputs of the h wavelength converters are then combined with a passive coupler. The wavelength selectors pass a specified input wavelength while rejecting all others. The wavelength converters, take a signal on any one of h possible input wavelengths and produce a corresponding output signal on a fixed output wavelength.

A system supporting $n = d^k$ external links (each with h channels), requires $2k - 1$ stages and each stage has n/d BSEs. Consequently, such a system requires $n(2k - 1)$ of the optical space switches, $nh(2k - 1)$ wavelength selectors and $nh(2k - 1)$ wavelength converters. The wavelength selectors and converters are expected to be the dominant cost components. For each optical output channel, the system requires $2k - 1$ of each of these components. A system with $n = 4096$, $h = 128$, $d = 16$ will have $k = 3$ and so will require 5 wavelength selectors and converters for each output channel. If each channel operates at 10 Gb/s, the system as a whole will have a capacity of about 5 Pb/s. If the combined cost of a wavelength selector and converter is C, then the cost per Gb/s of total system bandwidth is $C/2$.

A similar analysis of the costs of a large electronic router show that the parts cost, using current generation technology (.25 micron CMOS) is in the neighborhood of $1,000 per Gb/s of system throughput (with the optical transmission components accounting for close to half of the system cost). If the wavelength selectors and converters account for half of the cost of the burst switching system, then we need $C \leq$$1,000 in order for optical technology to be cost-competitive with electronics. To gain a significant advantage over electronics requires that C be no more than a few hundred dollars.

At what point is this cost objective likely to be achieved? The wavelength selector appears to be the biggest challenge. The most straightforward implementation consists of a wavelength demultiplexor followed be a rank of h optical gates, one of which is enabled at any one time, allowing the selected wavelength to pass to the output. Semiconductor optical amplifiers can be used to implement the required optical gates, but they are currently expensive and cannot be integrated in large quantities on a single substrate along with the wavelength demultiplexor. Substantial progress is needed on optical component integration to allow a device capable of selecting one of 128 channels to be mass-produced at a cost of a few hundred dollars, but there do not seem to be any fundamental obstacles to prevent it.

Deployment Strategies for Burst Switching Systems

Burst switching systems will have to be deployed within existing networks. The prototype being constructed will include an interface to an ATM network for this purpose. Burst switches can also be deployed within IP networks. Currently, large ISPs implement their networks with routers at the edges and ATM switches for interconnection among the routers. Burst switches could be used to replace or augment ATM switches in this architecture.

To implement this strategy, edge routers can select packets to be sent through the burst network based either on knowledge of end-to-end flow characteristics or on the basis of packet length, with long packets going to the burst network and short packets going through an ATM or router-based network. Since large packets typically belong to larger data transfers, this strategy would allow the interface to the burst network to group packets into bursts for efficient transmission through the burst network.

Crossbar Control in Burst Switch Elements

Burst switching systems use lookahead resource management to schedule the use of channels in WDM links and the use of memory. This requires that crossbar operations (making and breaking connections) be scheduled in advance by a crossbar controller. Since each crossbar operation takes time to complete, we need to ensure that each required operation will take place in a timely fashion – not too early and not too late. It's important that we be able to determine, at the time an operation is scheduled, if it can be performed within a prescribed time interval or not. If it cannot be, the operation may have to be rejected.

The prototype burst switch uses a simple but not entirely general approach to the problem of crossbar scheduling. A separate queue of requests is maintained for each outgoing channel and these requests are timestamped. Because the phase 1 system uses horizon scheduling, crossbar operations are placed in the queue in timestamp order, allowing for relatively simple processing. The crossbar controller is designed to be fast enough to allow a worst-case sequence of operations to be performed, so long as consecutive bursts in a channel are separated by a prescribed minimum time and so long as burst durations satisfy a prescribed minimum duration.

A more general solution to the crossbar control problem involves the use of a *differential search tree* [5] which is used to project the number of pending crossbar operations at all future times. When scheduling a new request, we first determine if the addition of this operation to the crossbar's operation schedule would cause the projected backlog to ever exceed some specified threshold level, D. This threshold is chosen as the maximum delay we can tolerate in the performance of a scheduled crossbar operation.

To implement this solution, the differential search tree of [5] must be extended. The crossbar controller must be able to quickly determine the first time following a specified time t when the backlog will be zero, since the addition of a new operation at time t will cause the backlog to increase by one for all times between t and the time the backlog first becomes zero. To provide this capability, the entries in the differential search tree are extended to include a Δ*min* field, used to determine the minimum backlog within a given subtree, in the same way that the existing Δ*max* field is used to determine the maximum backlog within a given subtree.

Scheduling a crossbar operation within an interval $[t_1, t_2]$ involves the following steps.

- Let t be the earliest time following $t_2 - D$ when the backlog will be zero.
- If the backlog is never as large as D during the time interval from $t_2 - D$ to t, then schedule the operation for $t_2 - D$ and add one to the backlog throughout this time interval.
- If the backlog is equal to D at some point during the time interval from $t_2 - D$ to D, then determine the latest time τ prior to $t_2 - D$ when the backlog is zero. If $\tau \geq t_1$, then schedule the operation to take place at time τ and add one to the backlog at time τ.

Note that $t_2 - t_1$ must be at least D. Also note that each of the three steps can be performed in $O(\log n)$ time using the differential search tree representation, where n is the number of scheduled crossbar operations. Finally, note that the procedure outlined above schedules the operation as close to the end of the interval as possible. There is a similar procedure that can be used to schedule the operation as close to the start of the interval as possible. Each of these procedures is preferred in different situations.

Name	Role	ARL start	% effort
Alex Chandra	logic design	1/99	100%
Tom Chaney	logic design & project coordination	1/93	50%
Yuhua Chen	logic design	1/99	100%
Maynard Engebretson	physical design & PC board layout	1/94	50%
John Lockwood	logic designer	6/99	70%
Tom McLaughlin	logic design	11/98	100%
Naji Naufel	logic design	5/99	100%
Dave Richard	transmission interface design	6/91	30%
Mike Richards	PC board layout	6/91	50%
Randy Richards	ASIC layout	6/91	100%
Fred Rosenberger	logic design	6/95	30%
Wen-Jing Tang	physical design & signal integrity	2/99	100%
Jonathan Turner	principal investigator	6/91	35%

Figure 5: Development Staff for Burst Switch Project

5. Project Staffing

The burst switch project has lagged behind the original schedule, in large part due to unexpected difficulties in getting the project fully staffed. This problem has now been remedied as can be seen from the table in Figure 5, which summarizes the current staffing situation. We now have over nine full-time equivalents working on this project. As can be seen from the table, six new staff members have been added since late last year. We expect to be able to make up much of the lost time, now that the project is fully staffed. However, we anticipate that the original schedule may have to be stretched by up to six months in order to complete all the project deliverables.

References

[1] Turner, Jonathan S. “Terabit Burst Switching,” Washington University Technical Report, WUCS-98-17, 1998.

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

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

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

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

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

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