28.1 A 4.5Tb/s 3.4Tb/s/W 64×64 Switch Fabric With Self-Updating Least-Recently-Granted Priority and Quality-of-Service Arbitration in 45nm CMOS

Sudhir Satpathy, Korey Sewell, Thomas Manville, Yen-Po Chen, Ronald Dreslinski, Dennis Sylvester, Trevor Mudge, David Blaauw

University of Michigan, Ann Arbor, MI

High speed and low power routers form the basic building blocks of on-die interconnect fabrics that are critical to overall throughput and energy efficiency of high performance systems [1,2]. Conventional routers use distinct logic blocks for routing data and handling arbitration [3,4]. At higher radices, connections between these blocks become a bottleneck, limiting router scalability and degrading performance. Recently two switch topologies [5,6] merged the data routing fabric with arbitration control, avoiding this bottleneck. However, [6] relies on centralized control for channel allocation, limiting performance, while [5] is restricted to a small set of fixed priorities, rendering input ports prone to starvation. In addition, ever larger CMPs will require continued increases in bandwidth over previous designs. To address these issues, we present a 64×64 single stage swizzle-switch network (SSN) with 128b data buses (8192 input/output wires). SSN can connect any input to any output, including multicast. It has a peak measured throughput of 4.5Tb/s at 1.1V in 45nm SOI CMOS at 25ºC. SSN's key features are: 1) A novel, single cycle least recently granted (LRG) priority arbitration technique that re-uses the already present input and output data buses and their drivers and sense amps. 2) An additional 4-level message-based priority arbitration for quality of service (QoS) with 2% logic and 3% wiring overhead. 3) A new bidirectional bit-line repeater that allows the router to scale to >8000 wires. These features result in a compact fabric (4.06mm^2) with throughput gain of 2.1× over [5] at 3.4Tb/s/W efficiency which improves to 7.4Tb/s/W at 600mV.

Conventional LRG implementations use controllable delay elements to resolve conflicts [4], which are tuned to change priorities. Large routers require many such delay elements, incurring overhead and probability of meta-stability failures. In contrast, SSN uses a fully static circuit technique that is completely embedded in the data routing fabric by re-using the data-routing bit-lines as "*priority lines*" for arbitration. The LRG and QoS priorities and the switch configuration are all stored locally at each crosspoint using a novel encoding. Since the data routing fabric is routing limited, this additional logic imposes zero area overhead over a simple switch. Furthermore, as the arbitration logic re-uses the data bit-lines and peripherals, arbitration has the same latency as data transfer and the two scale in tandem with the switch radix.

SSN is a matrix-type fabric (Fig. 28.1.1) with input buses running horizontally and output buses vertically. When data is routed, the input and output buses transfer data traffic. During arbitration the input bus routes a multi-hot code indicating which output channel(s) are requested by that input, and the output bus is used for conflict detection and arbitration. Each crosspoint stores a connectivity status bit indicating whether the input bus was granted access to the output channel. A 63-bit priority vector is also stored to represent the priority of the input bus with respect to all other inputs for that output bus. Fig. 28.1.1 shows the priority vector at each crosspoint in a blow-up of a *single* output channel. Each input bus is assigned a unique bit line from the channel as its *priority line* which, if high, indicates it as the winner in a particular arbitration cycle. Similarly, each bit in the priority vector at a crosspoint corresponds with a *priority line* (bit-line) and indicates whether the input bus at that crosspoint has higher or lower priority than the input bus associated with the priority line. For instance, in Fig. 28.1.1 priority line *m* corresponds to input bus *m* while the *m*-th priority bit of bus *n* is a 1, indicating that *n* has higher priority than *m*. When input *n* requests the output channel this high bit results in the discharge of priority line *m*, suppressing access by input *m*. In contrast, input *l* stores a 0 at its *m*-th priority bit and hence does not suppress an access request from input *m*, meaning that *l* has lower priority that input *m*. Priority vectors need to be set consistently. In Fig. 28.1.1, the 0 at bit *m* of input *l* must be mirrored with a 1 at bit *l of* input *m*. Furthermore, the priority bits need to be correctly updated after each arbitration cycle to implement LRG policy. We propose a new, simple mechanism to accomplish this. In Fig. 28.1.1, inputs *l* and *m* request the output channel in an arbitration cycle. Input *m* wins owing to its higher priority and its connectivity status bit is set to 1. After data transfer, input *m* releases its channel during a channel release cycle. In this cycle, input *m* first *resets* all its priority bits. This guarantees that *m* now has lowest priority, as required by the LRG algorithm. At the same time, input *m* also lowers its *priority line m*, which is a signal to other crosspoints in the output channel to *set* their *m*-th priority bit. This ensures that all other input buses now have higher priority than *m*. Input buses with higher priority than *m* remain unchanged and o*nly* inputs with lower priority than input *m* are increased in their priority by exactly one level. This simple and fast update mechanism provably guarantees both consistency of all priority vectors and correct implementation of the LRG arbitration scheme, which enables efficient and deadlock-free routing [7].

Fig. 28.1.2 shows an SSN crosspoint circuit and the priority storage latch. During a request/release cycle channels are indexed using the lower 64 bits from the input bus. Crosspoints send acknowledgement over the upper 64 bits. SSN also features a 4-level message-based QoS arbitration technique that allows only input buses with the highest message priority to arbitrate for the channel (Fig. 28.1.3.) A 2-bit *message priority* is decoded into a 4-bit thermometer code at the crosspoint, which is used to selectively discharge priority bit-lines comprising the *QoS priority bus*. A multiplexer samples one of those priority bit-lines using its own *message priority* and the input bus progresses to the LRG arbitration cycle if the monitored priority bit is not discharged. Using separate wires for QoS arbitration incurs 3% area overhead. However, the additional QoS arbitration cycle can be overlapped with the prior routing operation for the output bus, avoiding a latency penalty. The SSN features 8448 word-lines and 8576 bit-lines spread across 4096 crosspoints. The integration of the LRG and QoS control within this fabric with low overhead greatly improves SSN scalability to realize a fabric of large size. In addition, new bi-directional repeaters (Fig. 28.1.3) are used for bit-lines that use a regenerative sensing element to improve delay despite high slew rates on long bit-lines. The regeneration reduces bit-line delay by 32% and allows for a 50% smaller bit-line driver compared to a conventional repeater (Fig. 28.1.4, simulated). Simulated fabric latency shows 1.6× performance benefit from repeated bit-lines (Fig. 28.1.4) and near-linear latency increase with radix size. Fine grain clock gating reduces clock power by 94% at each crosspoint with 2.3% added delay. A crosspoint is clocked only if its connectivity status is ON, a request is asserted, or an LRG priority update occurs. Adjacent input ports are driven from opposite directions, reducing routing congestion and local Ldi/dt drop when repeaters on the 2.5mm long input bus switch.

The SSN achieves 4.5Tb/s at 1.1V with an efficiency of 3.4Tb/s/W (Fig. 28.1.5), which is 3.7× higher than [4] at similar bandwidth. [4] uses an 8×8 mesh topology based on 5×5 routers at each node to connect 64 units whereas SSN uses a 64×64 single stage fabric. SSN is fully functional down to 550mV with a measured peak efficiency of 7.4Tb/s/W at 0.6V. Architectural simulations show that the worst-case cache access latency for conflicting requests improves by 1.8× for an SSN-enabled 64-core system due to the implemented LRG algorithm (Fig. 28.1.6). A routing study shows that only one metal layer in each direction (NS/EW) is needed, requiring 12% of routing tracks in these layers to connect 64 cores and caches with the SSN.

Acknowledgement: We are thankful to NSF and ARM Ltd. for funding this research.

References

[1] S. Bell *et al*., "Tile64 Processor: A 64-Core SoC with Mesh Interconnnect," *ISSCC,* pp. 88-89, 2008.

[2] S. Rusu *et al*., "A 45 nm 8-Core Enterprise Xeon® Processor," *JSSCC,* pp. 7-13, Vol. 45, No. 1, Jan 2010.

[3] P. Salihundam *et al*., "A 2Tb/s 6x4 Mesh Network with DVFS and 2.3Tb/s/W router in 45nm CMOS," *SoVC,* pp. 79-80, 2010.

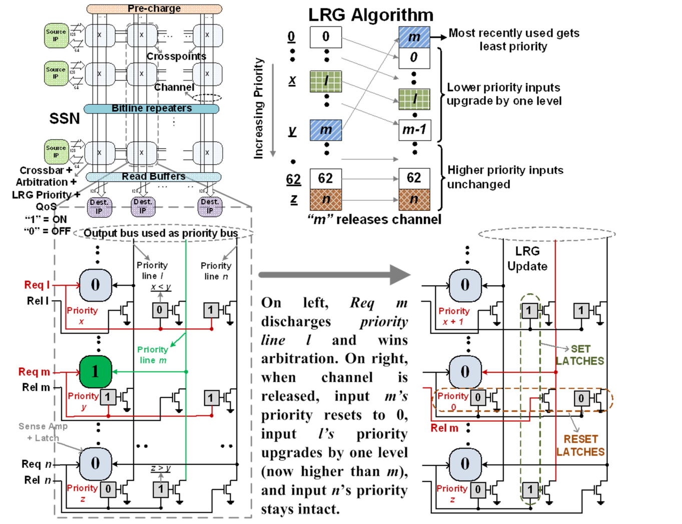
[4] M. Anders *et al*., "A 4.1 Tb/s Bisection-Bandwidth 560Gb/s/W Streaming Circuit-Switched 8x8 Mesh Network-on-Chip in 45nm CMOS," *ISSCC,* pp. 110-111, 2010.

[5] S. Satpathy *et al*., "SWIFT: A 2.1Tb/s 32×32 Self-Arbitrating Manycore Interconnect Fabric," *SoVC,* pp. 180-181, 2011.

[6] S. Satpathy *et al*., "A 1.07 Tb/s 128x128 Swizzle network for SIMD Processors," *SoVC,* pp. 81-82, 2010.

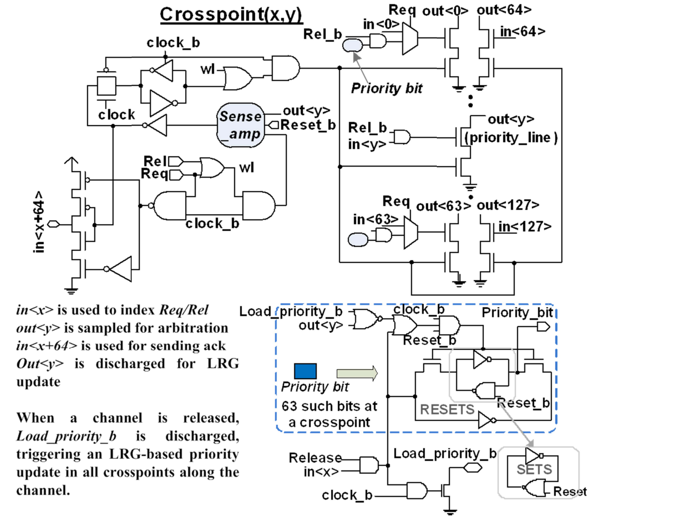
[7] M. Lee *et al*., "Probabilistic Distance-based Arbitration: Providing Equality of Service for Many-core CMPs", IEEE *MICRO43,* 2010.

[8] S. Vangal *et al*., "A 5.1GHz 0.34mm2Router for Network-on-Chip Applications", *SoVC,* pp. 42-43, 2007.



On left, *Req m* discharges *priority line l* and wins arbitration. On right, when channel is released, input *m's* priority resets to 0, input *l's* priority upgrades by one level (now higher than *m*), and input *n's* priority stays intact.

Figure 28.1.1: SSN architecture with LRG priority update



in<x> is used to index *Req/Rel* *out<y>* is sampled for arbitration *in<x+64>* is used for sending ack *Out<y>* is discharged for LRG update

When a channel is released, *Load_priority_b* is discharged, triggering an LRG-based priority update in all crosspoints along the channel.

Figure 28.1.2: Crosspoint circuit and priority storage latch

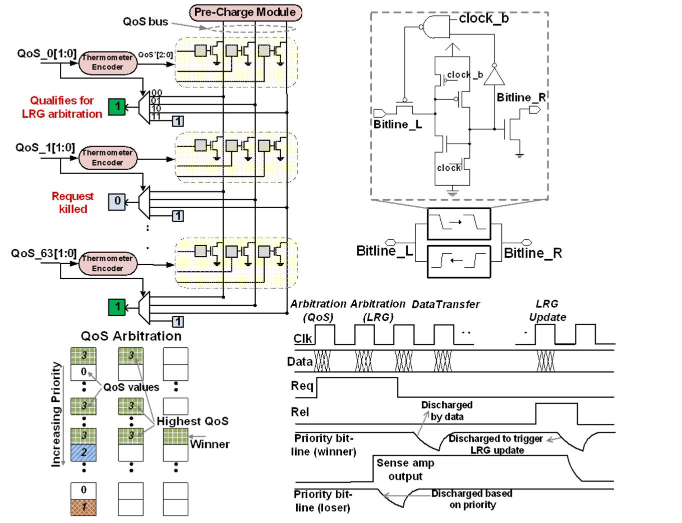
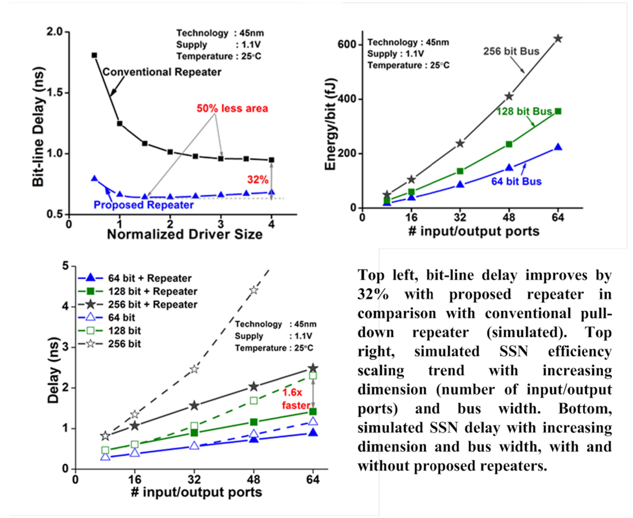


Figure 28.1.3: QoS arbitration technique and bit-line repeater



Top left, bit-line delay improves by 32% with proposed repeater in comparison with conventional pull-down repeater (simulated). Top right, simulated SSN efficiency scaling trend with increasing dimension (number of input/output ports) and bus width. Bottom, simulated SSN delay with increasing dimension and bus width, with and without proposed repeaters.

Figure 28.1.4: Simulated SSN delay and efficiency scaling

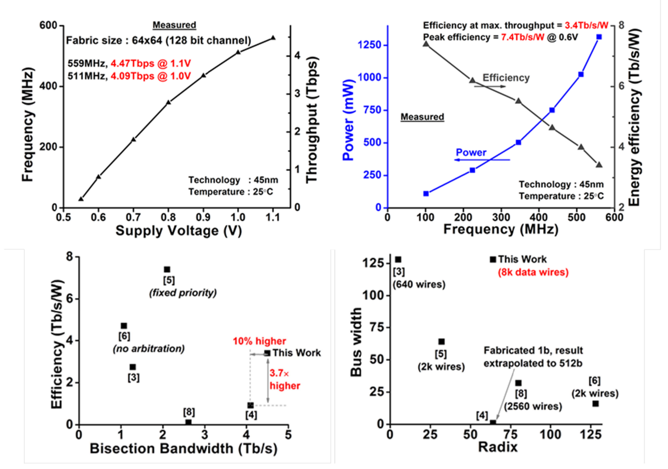
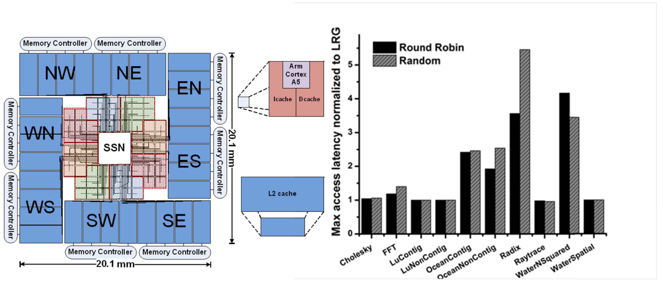


Figure 28.1.5: Measured SSN performance and power and comparison with prior switch fabrics



LRG reduces worst-case cache access latency by 1.83× and 2.03× on average over round robin and random arbitration schemes, respectively. Left, floorplan study to determine routability of a 64-core CMP with SSN. Cores are placed closer to SSN to reduce average interconnect length while caches are placed at periphery. Routing between SSN and cores/caches was performed using only 1 metal layer in each direction (NS/EW) and used only 12% of the interconnect resources in those two global routing layers.

Figure 28.1.6: Architectural evaluation of SSN enabled 64 core system with SPLASH 2 benchmarks