IEEE Symposium on VLSI Circuits. June, 2010. Honolulu, Hawaii. pp. 81-82.

A 1.07 Tbit/s 128×128 Swizzle Network for SIMD Processors

Sudhir Satpathy, Zhiyoong Foo, Bharan Giridhar, Ronald Dreslinski, Dennis Sylvester, Trevor Mudge, David Blaauw
University of Michigan, Ann Arbor

Abstract

A novel circuit switched swizzle network called XRAM is presented. XRAM uses an SRAM-based approach producing a compact footprint that scales well with network dimensions while supporting all permutations and multicasts. Capable of storing multiple shuffle configurations and aided by a novel sense-amp for robust bit-line evaluation, a 128×128 XRAM fabricated in 65nm achieves a bandwidth exceeding 1Tbit/s, enabling a 64-lane SIMD engine operating at 0.72V to save 46.8% energy over an iso-throughput conventional 16-lane implementation at 1.1V.

Motivation and Proposed Approach

Single instruction multiple data (SIMD) engines are becoming common in modern processors to handle computationally intensive applications like video/image processing [1]. Such processors require swizzle networks to permute data between compute stages. Existing circuit topologies [2,4] for such networks do not scale well due to significant area and energy overhead imposed by a rapidly growing number of control signals (Fig. 4), limiting the number of processing units in SIMD engines. Worsening interconnect delays in scaled technologies aggravate the problem. To mitigate this we propose a new interconnect topology, called XRAM, that re-uses output buses for programming, and stores shuffle configurations at cross points in SRAM cells, significantly reducing routing congestion, lowering area/power, and improving performance.

Fig. 1 provides a top-level diagram of the XRAM approach. The input buses span the width while the output buses run perpendicular to them, creating an array of cross points. For each configuration, the state of an SRAM bitcell at a cross point determines whether or not input data is passed onto the output bus at the cross point. Along a column only one bitcell is programmed to store a logic high and create a connection to an input. To minimize control overhead, output buses are used to program the bitcells, reducing the number of wires by 40% in a 128×128 16-bit bus implementation. Each bitcell in a column is programmed by a unique bit from the output bus. As buses get wider, the additional silicon area at a cross point is used to store multiple configurations. In the 65nm implementation presented here, a 16-bit bus width allows six configurations to be stored without incurring any area penalty. Any one of these configurations can be individually programmed and used to shuffle data. Since configurations are never read out, access transistors in the SRAM cells are upsized to achieve write robustness at the cost of read stability.

In *programming mode*, the controller sends a one-hot signal onto each output bus. A global wordline is then raised high to program the XRAM. With 16-bit buses, a 16×16 XRAM can be programmed in a single clock cycle. Larger XRAMs are divided into multiple sections with independent wordlines and one section is programmed at a time. In *transmission mode*, the wordline stays low and incoming data is passed onto the output bus at the cross point storing a 1 using a precharge followed by conditional discharge technique. Input data can be multicast by storing multiple 1's in a row.

Fig. 2 shows the cross point circuitry. One of the bits (out0<0>) from the output bus is used to program the cell. The one-hot signal Config[0:5] is used to select a configuration. The output bus is precharged concurrently while incoming data settles on the input bus. Thereafter, the output lines are discharged if the cross point bitcell stores a 1 and the incoming data is 1. Widths of 480nm for the top transistor and 540nm for the bottom transistor were chosen in the discharge stack to optimize overall XRAM delay. To avoid repeated discharge for static high inputs, incoming data is transition encoded.

Output lines are evaluated by a bank of sense amplifiers. Conventional sense amps are prone to device mismatch, rely on accurate timing, and consume power in precharging internal nodes every cycle, rendering them unusable in XRAM. We propose a novel single-ended thyristor-based sense amplifier shown in Fig. 3. The sense amp is initially precharged to a low gain leakage state. During evaluation, internal node *A* is immediately coupled to the output line through a PFET access transistor. If the voltage on the output line drops, the thyristor pair in the sense amp is triggered and a logic low is evaluated. The precharge transistors are sized large enough to compensate for leakage through the thyristor pair. The PFET access transistor is sufficiently weak to prevent the output line from fully discharging to save power. Given improved mismatch tolerance, devices can be downsized, resulting in only 3% to 20% increase in XRAM area in comparison with 12% to 52% for a conventional sense amp across dimensions ranging from 128×128 to 16×16. The proposed sense amp can be fired as soon as the output line starts discharging. This simplifies timing generation since *SE* and *discharge* can be generated off the same edge. The internal nodes switch only if the input discharges, reducing precharge power.

The number of wires and hence the dimension of XRAM grows as O(*n*), where *n* is the number of inputs/outputs, compared to O(*n*(log*n*)) in conventional switches. Simulation results in Fig. 4 compare a conventional crossbar to XRAM with increasing dimensions, showing the XRAM is 3.3× smaller, 37% faster, and saves 75% energy per bit transfer for large networks. It does not require clustering of unique bits from different buses at input, thereby avoiding routing congestion. Programming power and latency can be further saved by caching frequently used shuffle patterns in the XRAM. Data transfer through XRAM takes a single cycle, while programming a channel requires one (if cross points to be programmed lie in the same section) or two cycles. An entire XRAM can be programmed in N cycles where N = Number of inputs/Bus width. Programming latency can be improved to ($\log_2$(Number of inputs)/Bus width) cycles by locally decoding control words at cross points. The XRAM can switch configurations in consecutive cycles without additional delay.

Prototype Implementation

We explored the low voltage scalability of SIMD engines in the context of XRAM [5]. As shown in Fig. 5, an XRAM-based system scales better than a conventional system when voltage/frequency are lowered, and SIMD lanes are widened to maintain iso-throughput. The optimal SIMD width was found to be 64, employing a 128×128 XRAM to shuffle two operands, and this design was fabricated in 65nm CMOS. The register files (Fig. 6) are equipped with LFSRs and the processing units use signature analyzers to independently test the XRAM. The processor was verified by running a 64-point FFT. Shuffle patterns for the butterfly stages were programmed once into the XRAM and no further programming was needed during runtime.

Fig. 7 shows measured results for XRAM, which can achieve a bandwidth of 1.07 Tbit/s at 1.1V while consuming 227 mW. At higher voltage, the input delay (max of input data and control) and output delay (discharge + sense amp) are matched. In this case a 12% speed improvement is achieved when the input delay is optimized by launching XRAM control signals early. At lower voltages, the output delay dominates and input delay optimization results in no significant speed improvement. Measured results showing the impact of transition encoding are shown in Fig. 8. With transition encoding disabled the XRAM power increases linearly with the fraction of 1's in the input data. The minima at either extreme are due to zero switching activity in the input/output buses. The power breakdown in Fig. 9 reveals that output lines dominate XRAM power at low voltage, making transition encoding more effective than at higher voltage. Table 1 compares XRAM to a relevant prior design [3]. Measured results for a 64-lane SIMD (64 multipliers) processor using a 128×128 XRAM and a 16-lane SIMD (16 multipliers) processor using a conventional 32×32 crossbar at iso-throughput are summarized in Fig. 10 including die micrographs. Energy savings of 29.5% and 46.8% are demonstrated for FFT-like (light load) and matrix inversion (heavy load) applications, respectively.

References

[1] Y. Lin *et al*, IEEE MICRO, pp. 114-123, 2007.

[2] K.Lee *et al*, ISSCC 2004

[3] M. Borgatti *et al*, ISSCC, 2003

[4] K. Chang *et al*, Symposium on VLSI Circuits, 1999

[5] M.Woh *et al*, ICASSP 2008, pp. 5388-5391

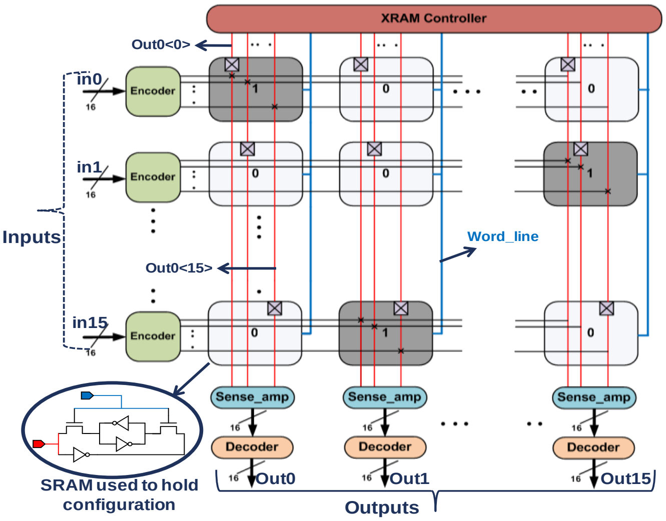
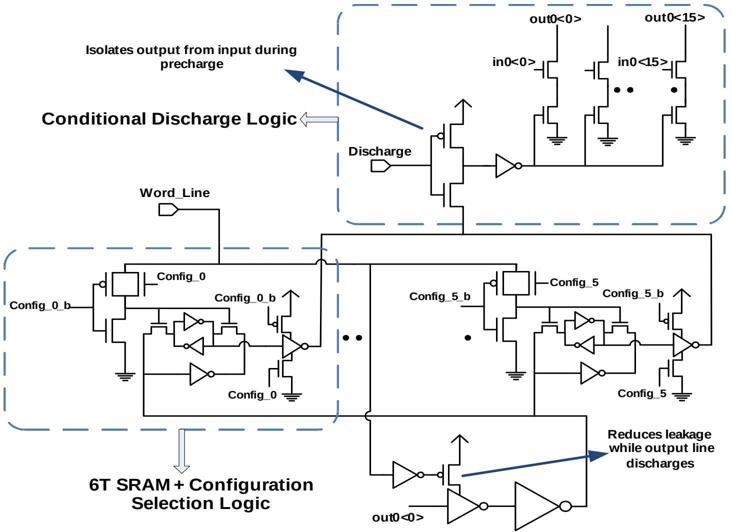


Figure 1. Top level diagram of XRAM showing output buses being used to program SRAM bit cells at cross points



out0<0>
out0<15>
in0<0>
in0<15>
Isolates output from input during precharge
Conditional Discharge Logic
Discharge
Word_Line
Config_0
Config_0_b
Config_5
Config_5_b
6T SRAM + Configuration Selection Logic
Reduces leakage while output line discharges
out0<0>

Figure 2. Cross point circuitry. Out0<0> is used to program the cross point

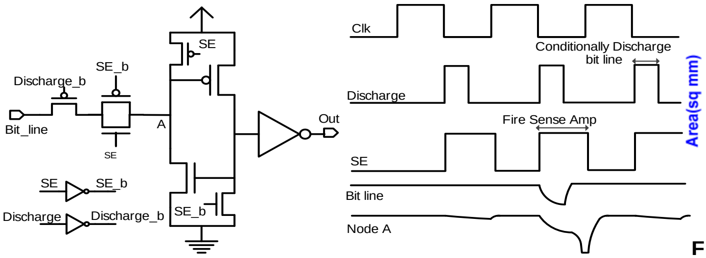


Figure 3. Thyristor based sense amp with timing diagram

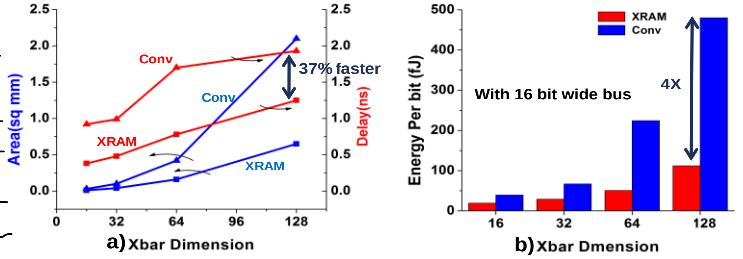


Figure 4. Comparison of XRAM and conventional (conv) crossbar across different sizes (16 bit buses) based on simulation

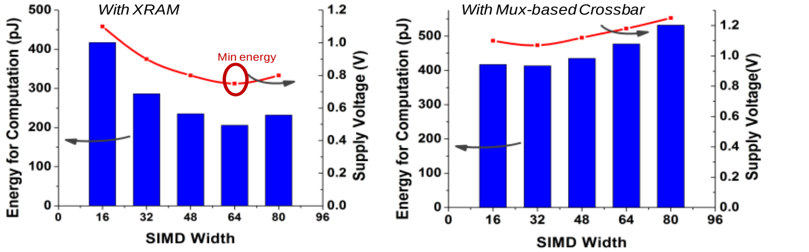


Figure 5. Simulation results showing XRAM equipped SIMD consumes 48% less energy than conventional SIMD at iso-throughput (6.4 GOPs per second)

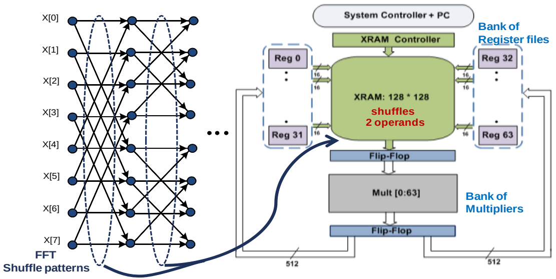


Figure 6. 64 wide SIMD data path with 128×128 XRAM

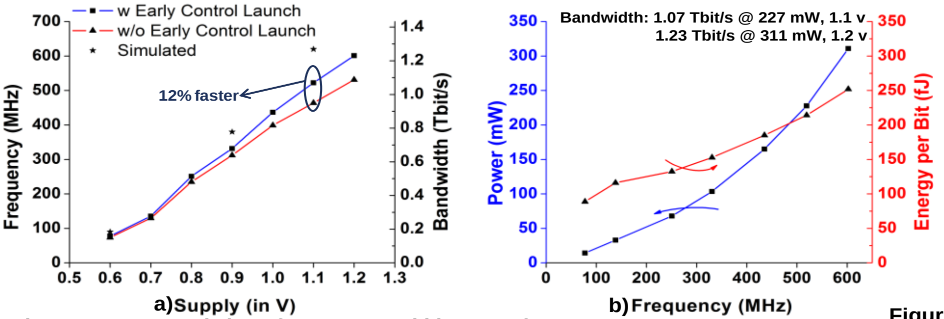


Figure 7. Measured a) performance and b) power for 128×128 XRAM

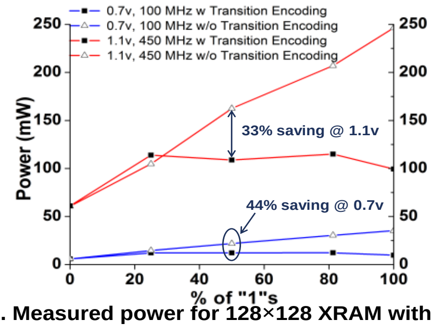


Figure 8. Measured power for 128×128 XRAM with and without transition encoding

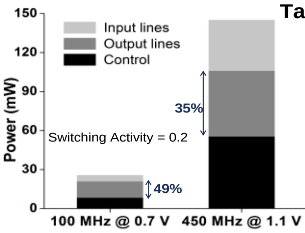


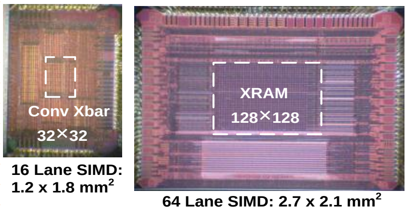
Figure 9. Measured XRAM power breakdown

Table 1. Comparison with ref[3]

Metric	XRAM	Ref [3]
I/O Ports	128+128	8+8
Total Bits	2048	512
Configs	6	8
Storage	SRAM	FPT*
Tech.	65nm	180nm
Area(mm²)	0.87	1.38
Freq.(MHz)	523	100
Latency(ns)	1.89	8.0/22.0**
Supply(v)	1.1	1.6/2.0***

*Flash Programmable Pass Transistor
**worst case
***Programming voltage

METRIC		Light Load		Heavy Load	
		16 Lane SIMD	64 Lane SIMD	16 Lane SIMD	64 Lane SIMD
Power (mW)	Ex	13.20	7.20	50.60	20.16
	Reg. File	27.50	12.96	40.70	18.72
	Xbar	11.00	18.00	47.30	33.84
	Control	5.50	2.16	13.20	8.64
	Total	57.20	40.32	151.80	81.36
Frequency(MHz)		400	100	400	100
Supply (V)		1.10	0.72	1.10	0.72



16 Lane SIMD: 1.2 x 1.8 mm^2

64 Lane SIMD: 2.7 x 2.1 mm^2

Figure 10. Measured data for 16 lane and 64 lane SIMD processors at iso-throughput and die micrographs. 46.8% power is saved for computation intensive work loads