Cross-point Circuits for

Computation, Interconnects, Security and Storage

by

Supreet Jeloka

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in the University of Michigan
2017

Doctoral Committee:

Professor David T. Blaauw, Chair
Professor Jerome P. Lynch
Professor Trevor N. Mudge
Professor Dennis M. Sylvester

To my family and friends

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Technology scaling used to improve circuit performance with every technology node while keeping the power density (power dissipation per unit area) constant. Power density is an important parameter for modern chip design. Power density determines how quickly a system heats up, and has to be maintained within the limits of the thermal cooling capabilities of the system. As technology scaling geometrically shrinks circuit area, cooling becomes harder if power is not reduced proportionately.

For older technology generations, voltage scaling used to provide quadratic power reduction, while smaller capacitance for smaller transistors provided a linear reduction in power. These two factors combined more than offset the area reduction, and allowed a linear increase in the clock frequency of the system, while keeping the power density constant. This increase in clock frequency of the system implied more instructions being processed in the same wall clock time, and hence enhanced performance of the system.

However, in newer technology nodes, the supply voltage scaling has plateaued as seen in Figure 1.1 [56], and hence both energy density and power density are increasing. As the thermal cooling capabilities are bounded, system designers have had to limit the clock frequency. As a result frequency scaling came to an end around 2004, as seen in Figure 1.2. Also, to maintain the power density as area continues to shrink, the maximum power has actually been reducing for processors, as seen in Figure 1.3. This total power consumption curve reiterates the limits created by power density, and consequently thermal cooling capabilities on system performance.

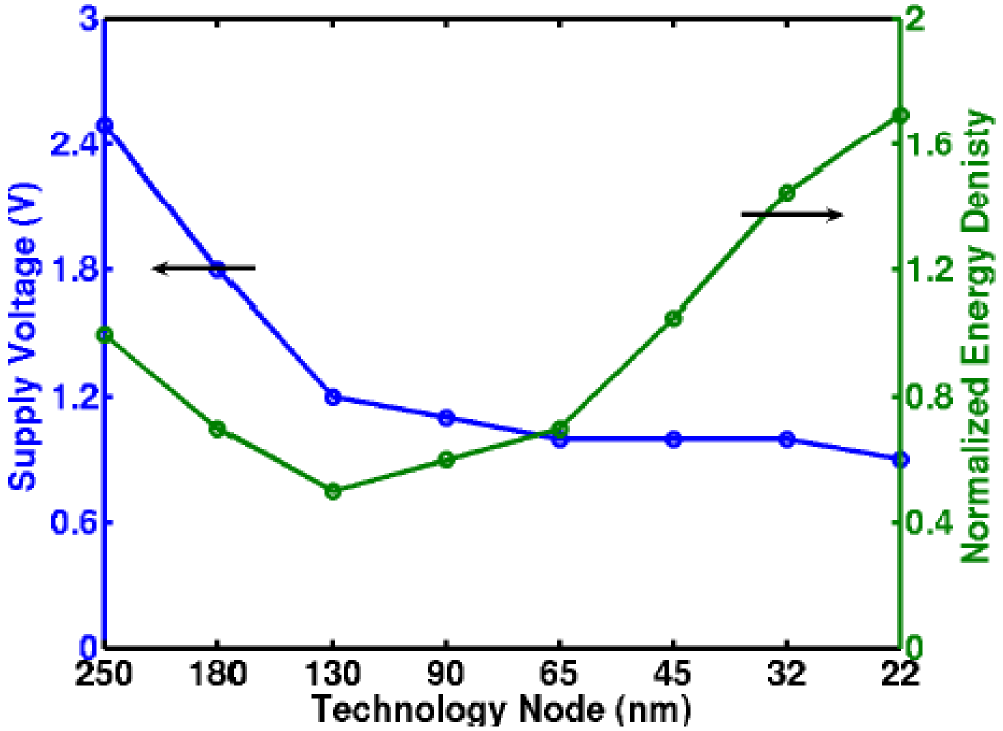


Figure 1.1 Energy density and supply voltage scaling with technology node [56]

As clock frequency stopped increasing, the performance stagnated because the processors can no longer be operated faster. The next logical step was to use this extra available area to add more cores in the system, and hence performing parallel processing to gain performance. Processor designs started using multiple cores, as can be seen from the trends in Figure 1.4, about the same time that clock frequencies stagnated.

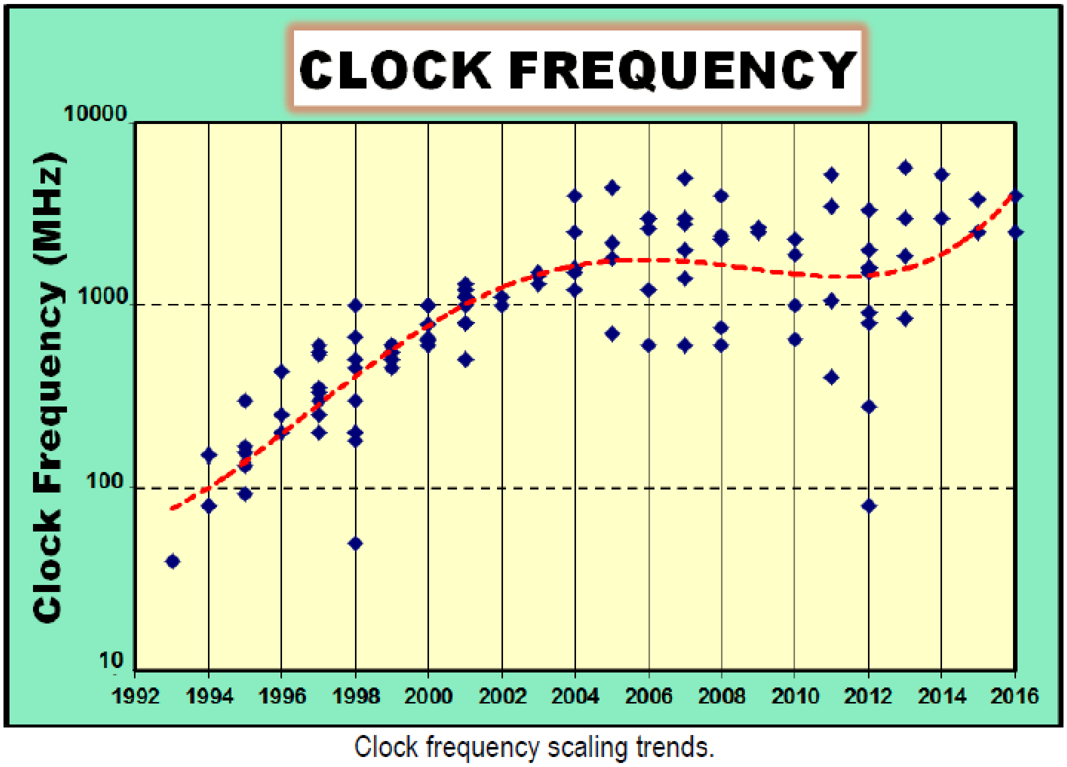


Figure 1.2 Clock frequency trends [58]

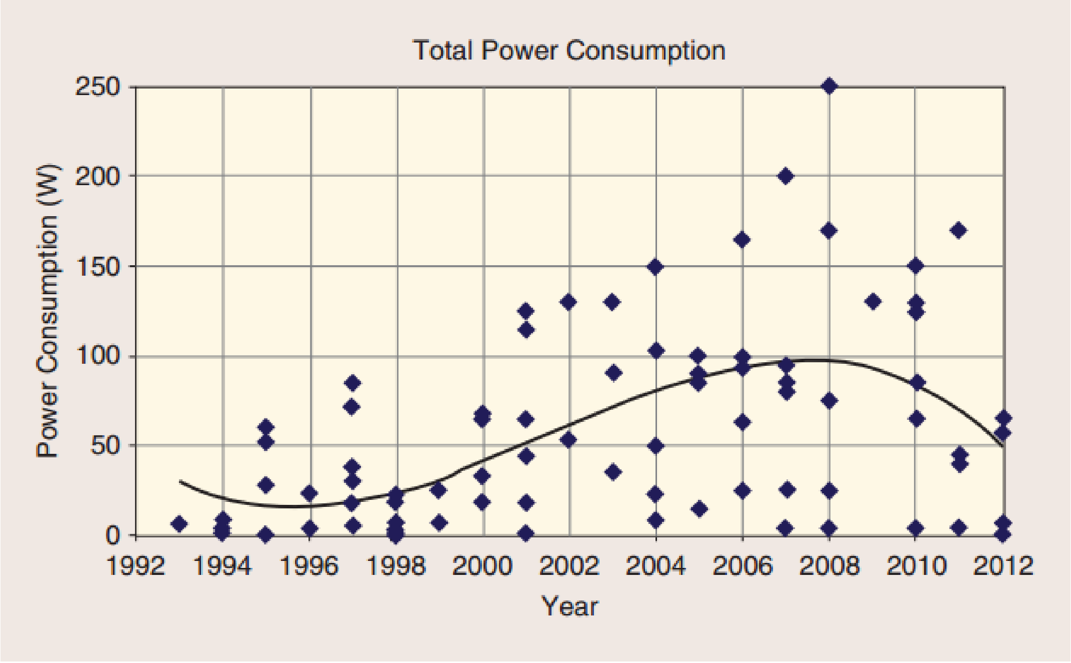
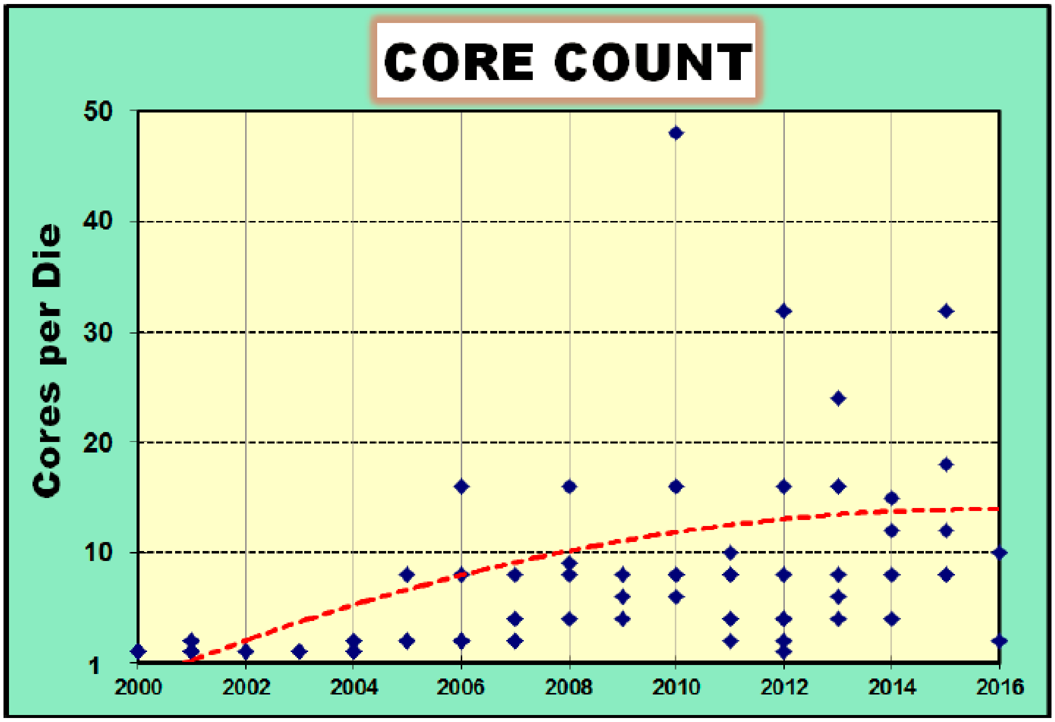


Figure 1.3 Total power consumption trend [57]



Core counts on processors published at ISSCC.

Figure 1.4 Core count trends [58]

Multi-cores increase area linearly, but the slowing down of voltage scaling has a quadratic impact on power dissipation. This implies that multi-core processors only delayed reaching the thermal cooling capability limits, but could not avoid it.

This has led to development of several architectural and circuit-level techniques for low power. The general trend has been to replace power-hungry, high-voltage and high-speed complex cores with a number of more power-efficient simpler cores working in tandem. New techniques like heterogeneous cores use a mix of both simple and complex cores. Other techniques like Near-Threshold Computing (NTC) [56][70] in conjunction with many-core systems, try to reduce the supply voltage to achieve the most optimum energy-performance operating point.

For performance intensive parts, such as servers, the number of cores has steadily increased. The same solution may not work for application specific processor, where the power profile is very different, and the 'all-on' mode power still shows an increasing trend in power, as seen in Figure 1.5. Notice the order of magnitudes lower total power than the power numbers shown in Figure 1.3. These applications still need more performance gains, but most are battery-operated and hence extremely energy sensitive.

Thus, with the current scaling trends, we need to apply different solutions for different application space. In this thesis we divide the design space into three categories, and propose solutions for improved energy-efficiency and the power-density problem for all three.

i. Many core systems – Performance intensive

ii. Mobile – Single / few-cores. Both performance and energy important

iii. Internet of Things (IoT) – Very low activity. Energy most important

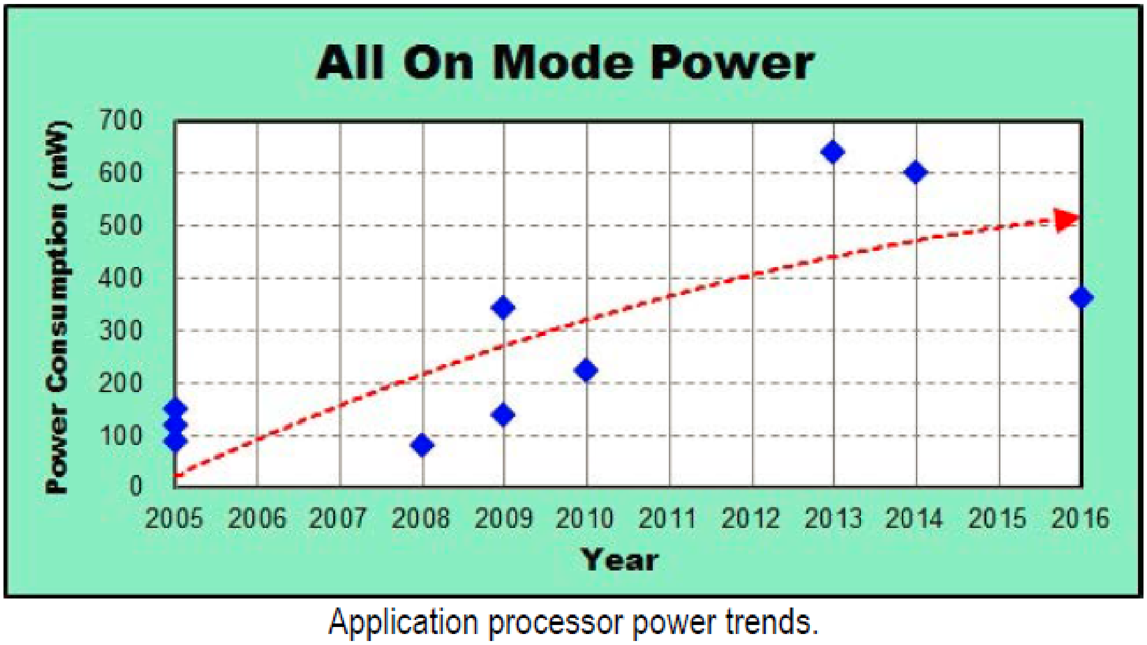


Figure 1.5 All-on mode power for application processors [58]

1.1 Many-core systems

Many-core systems are increasingly using power efficient simple cores. The performance of a single simple core is typically much lower than a power-hungry complex core, but by using a number of simple cores in parallel, and dividing the task into smaller parts, the overall performance of the system can be increased. The major challenge for many-core systems is the need to share data and hand-shaking signals between the cores working in parallel on the same task. As the simpler cores are more power-efficient, a majority of the energy is spent in moving the data from one core to the other, and back and forth from the memory.

As a result, many-core systems require low latency, area-energy efficient interconnect fabrics with extremely high band-width. Conventional interconnect topologies are optimized for few core systems, and hence are constructed out of low-radix switches such as a 2D-Mesh [1][2]. The radix of a switch is defined as the number of ports (nodes) that the switch connects. In a conventional 2D Mesh, each low-radix switch can only connect to four other nodes. For a fully connected many-core system comprising of several tens of cores, the low-radix switches will have to make connections (hops) through several intermediate nodes to transmit data from one core to the other. This leads to both high latency and high power consumption [3]. In addition, many-cores communicating together imply the interconnection fabric has to arbitrate more between competing requests from cores. Therefore, an interconnect fabric with efficiently designed high-radix switch and fair arbitration can be more optimal for many-core systems [4][5].

Concurrently, 3D integration has become an important means of improving performance as process scaling slows down. 3D integration allows the number of cores to be increased by stacking different layers [6], with short vertical connections between the layers. These short connections can be leveraged for speeding up inter-layer communication and building an efficient interconnect.

This thesis proposes a 3D cross-point based interconnect, *Hi-Rise* [67]. Hi-Rise is an efficient 3D high-radix switch. The proposed switch is a true 3D switch which connects inputs and outputs across different silicon layers. Hi-Rise adopts a hierarchical architecture with two internal switches per layer and dedicated layer-to-layer channels, to improve area efficiency, lower delay, and minimize the number of inter-layer vertical connections. Hi-Rise also provides built-in single-cycle arbitration across all inputs and outputs across different silicon layers. This improves both efficiency and scalability.

A new class based arbitration scheme is also proposed that is fully integrated into the switching fabric. This scheme makes the 3D hierarchical switch's fairness comparable to that of a flat 2D switch. The proposed switch extends scalability to radix 96 from that of the 64 radix supported by 2D switches at the same operating frequency, allowing for efficient integration of more cores.

1.2 Mobile systems with performance and energy constraints

Current generation of mobile systems already use aggressive voltage-frequency scaling and other energy reduction techniques, to bring down the power of conventional architectures. As Figure 1.6 shows, the conventional Von Neumann architecture has an inherent issue, where orders of magnitude more energy is spent on data movement between the cores and the memory hierarchy, than on the actual computation in the Arithmetic Logic Unit (ALU). The data movement cost consists of fetching the operands from the cache or the main memory to the register file of the processor, computation in the ALU, and then finally writing back the result in the memory.

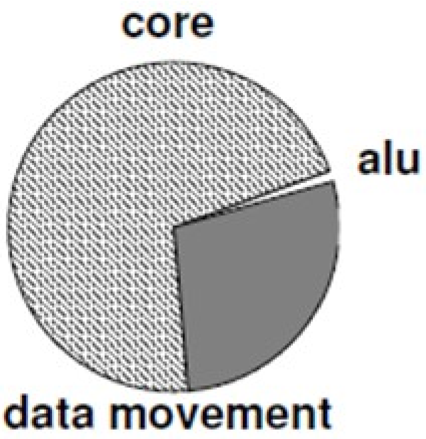


Figure 1.6 Processor dynamic energy

To improve the energy efficiency for such systems, the change required in the architecture is to move computation to where the data is stored. This shift in architecture will require storage that can inherently support computations. If the operands stored in the memory can be computed upon within the memory itself and then the result written back into the same memory, the data movement cost can be greatly reduced.

This thesis proposes a configurable memory that uses conventional SRAM bit cells, but can provide CAM (Content Addressable Memory) [36] capabilities for search applications and can also perform logical operations within the memory array. A Content Addressable Memory compares its search input data with every word stored in the memory, and returns the address location of the matching words. In addition, the configurable memory can perform bit-wise logical operations on two or more words stored within the array.

Thus, the configurable memory with CAM [45] and logical functions capability [66] can be used to off-load specific computational operations to the memory, improving system performance and efficiency. Performing logical operations in memory also frees up the ALU for more involved calculations, and hence boosts performance [46][47][48][49]. Using logic-in-memory improved both performance (~1.9×) and energy (~2.4×) [71] for certain applications like string matching, and bit matrix multiplication. The configurable SRAM can therefore be used in accelerators for application specific design, as well as general purpose processors. The proposed configurable memory chip has been designed and fabricated in a 28nm technology node.

The configurable memory can also be repurposed as a PUF (Physically Unclonable Function), [62] [63] which is like a device fingerprint, typically used for hardware authentication. The proposed PUF is a modified design of the configurable memory discussed above, and has also been fabricated in the 28nm technology node. Unlike existing PUF circuits, where the response depends only on the current challenge (input) vector, the proposed PUF response is dependent on a sequence of inputs. Sequence dependence, combined with a larger number of possible challenge-response pairs makes the response from this PUF hard to predict and hence more secure.

1.3 IoT systems

Unlike the server and mobile systems, IoT (Internet-of-Things) systems have an extremely small energy budget, and very low activity rate. They mainly consist of sensor nodes, which intermittently wake-up, log sensed data and go back to a very low-power or zero-power state.

For such IoT systems, the ability to power-off to save energy is vital. This is the reason for use of non-volatile memory in IoT systems for data logging. But, non-volatile memory may expend significant energy for write operations. To improve energy efficiency in IoT systems, this thesis proposes two non-volatile memory solutions with charge recycling techniques, to significantly reduce the write energy for the proposed non-volatile memory. The two non-volatile memory solutions proposed are low power SONOS (Silicon-Oxide-Nitride-Oxide-Silicon) Flash [64][65] and adiabatic Ferroelectric RAM (FRAM). [59][60][61]

Unlike conventional flash which uses energy-intensive hot carrier injection based program (write) operation, both erase and program in a SONOS flash cell is based on tunneling. This makes SONOS inherently more energy efficient. This thesis proposes a SONOS flash with an ultra-wide 1Kb/program cycle, enabled by low tunneling program power and a dedicated, multi-output transition pump which uses charge sharing and charge recycling. Combined with energy efficient charge pumps, the proposed flash program energy is 122pJ/bit with a 1Mbps throughput. The proposed SONOS flash memory chip has been designed and fabricated in a 130nm technology node, and is able to match the throughput of a conventional floating gate flash with ~10× lower energy.

The second non-volatile memory (NVM) solution proposed in this thesis is an adiabatic (charge recycling) FRAM array. Compared to the more popular flash memory, FRAM has significantly faster write access time, lower write energy, and does not require high voltages to write. Flash on the other hand, has better density than FRAM, and is scaling better with technology. FRAM solutions are ideal for sensor nodes with small write-intensive arrays, as the density advantage of flash is offset by the area penalty for the high voltage circuits. The proposed FRAM chip is designed and fabricated in a 130nm technology node. The proposed adiabatic FRAM design reduces the write energy by ~7× in simulations, compared to the baseline FRAM design.

1.4 Thesis organization

This dissertation proposes to solve the power density problem across systems, varying from very high performance systems to very low energy systems. It uses cross-point circuits for these solutions. Cross-point circuits are array based circuits, where the unit block is small, and can be easily optimized for both area and energy. These unit blocks are then tiled together to make much larger scalable circuits.

Chapter 2 proposes a cross-point based 3D high radix interconnect fabric, Hi-Rise, which is optimized for both throughput and arbitration fairness. At a radix of 64, Hi-Rise achieves an operating frequency of 2.2GHz, consumes only 44pJ of energy per 128-bit transaction and has an area of 0.451mm^2 in 32 nm technology. Hi-rise is therefore a fast, area-energy efficient interconnect for many-core systems.

Chapter 3 proposes a configurable memory that uses conventional SRAM bit cells, but can provide CAM capabilities for search applications and can also perform logical operations within the array between two or more words. Using memory for computation, reduces data movement costs, and greatly improves both energy-efficiency and performance of a processor. Using a standard push-rule 6T 28nm SRAM bit cell, the 64x64 (4kb) BCAM achieves 370 MHz at 1V and consumes 0.6fJ/search/bit. A logical operation between two 64-bit words achieves 787 MHz at 1V.

Chapter 4 proposes a cross-point based security block, based on the SRAM in Chapter 3. The proposed PUF response is dependent on both, the length and the order of the input challenge sequence. Therefore, the sequence dependent PUF can be run in many configurations and has a large challenge-response space.

Chapter 5 and Chapter 6 propose cross-point based non-volatile storage circuits optimized for write-intensive applications in IoT systems.

In Chapter 5, a low power SONOS flash memory is proposed. The proposed memory is a 130nm, 1024x260 SONOS flash which requires only 122pJ/bit program energy. It supports a wide program of 1Kbit per program cycle at a throughput of 1 Mbps. The proposed SONOS flash is ~10× lower program energy than conventional flash.

In Chapter 6, an adiabatic FRAM array is proposed. The proposed 130nm, 512x80 FRAM memory uses adiabatic techniques to reduce the write energy by ~7× compared to the non-adiabatic design in simulations, whereas the read with write back energy reduces by ~4×.

Concluding this dissertation is Chapter 7, which summarizes the contributions of this dissertation and discusses some possible future directions. At the end of Chapter 7, is a list of related publications generated from this work.

CHAPTER 2

Hi-Rise: A high-radix switch for 3D integration

2.1 Introduction

The number of cores on a single chip has seen a steady upward trend due to emerging parallel workloads and the need to meet performance goals within constrained power budgets. These many-core systems require low latency, area-energy efficient interconnects with extremely high bandwidth. Conventional interconnects constructed out of low-radix switches such as a 2D-Mesh [1][2] do not scale well because of the decreased performance resulting from larger hop counts and high power consumption [3]. Therefore, an interconnect fabric with efficiently designed high-radix switches is optimal for future many-core processors [4][5]. Concurrently, 3D integration has become an important means of improving performance as process scaling slows down. This technique allows the number of cores to be increased by stacking different layers [6], with short vertical connections between the layers. These short connections can be leveraged for speeding up inter-layer communication and building an efficient interconnect. Interconnects based on low-radix 3D switches [7][8][9][10] have been proposed in the past for 3D multi-core processors. However, as previously mentioned, low-radix and low bandwidth switches do not provide good scalability for a large number of cores.

3D high-radix switch design entails its own unique challenges. Unlike a flat 2D high-radix switch [11], the inputs and outputs of a 3D switch are spread over multiple layers. A 3D high-radix switch requires both intra-layer connections and inter-layer connections. The inter-layer vertical connections between silicon layers are made using Through-Silicon Vias (TSV). This leads to heterogeneity in the intra-layer and inter-layer connections. Consequently, a simple 3D high-radix [12] switch folded over silicon layers has lower performance than a flat 2D switch. A high-radix 3D switch design thus requires: 1) switch datapath optimized for this connection heterogeneity; 2) composable and fair arbitration scheme across inter-layer and intra-layer connections; 3) reduction in the number of expensive TSVs with minimal impact on the switch performance parameters, i.e., throughput, latency and fairness; and, 4) improved area and energy efficiency to offset the design and manufacturing cost.

This thesis proposes Hi-Rise, a high-radix 3D switch that achieves significant scalability and reduces the required number of TSVs by using a hierarchical architecture with dedicated layer-to-layer channels. The proposed Hi-Rise switch is divided into layers, each layer has two switches, a local switch and an inter-layer switch. The local switch connects local inputs to both intermediate outputs and vertical channels to other layers. The inter-layer switch connects both vertical channels from other layers and the intermediate outputs from the local switch to the final outputs on its layer. When combined, the two switches per layer result in a fully connected switch.

The hierarchical datapath of the switch is optimized for 3D connections. A key issue with the hierarchical switch datapath is that it can lead to unfairness as the arbitration is decomposed into two phases. To address this, a new arbitration scheme, Class-based Least Recently Granted (CLRG), which brings the fairness of a hierarchical 3D switch close to that of a flat 2D switch using LRG priority, is proposed. In this scheme, the inter-layer switch maintains a small counter for each input which signifies that input's output usage, and accordingly bins the requestors into different priority classes. Inputs in the same class use LRG to break ties. In contrast to CLRG, the implementation complexity of prior multi-stage arbiter designs [13][14], make them unattractive for high-radix switches. In addition, these arbiter designs are not optimized for 3D, and lead to high inter-layer traffic, unlike the proposed CLRG scheme. We demonstrate that the proposed class based arbitration allows for single cycle arbitration and full integration within the switch fabric, with no area and negligible performance overheads.

The proposed 3D switch is evaluated for various architectural and physical configurations. This thesis studies the proposed switch design through detailed circuit-level delay analysis, power modeling, and micro-architectural cycle accurate performance simulations. We study various synthetic traffic patterns, and also real application benchmarks. The 3D switch is analyzed for different radices, number of stacked layers, and different TSV technologies. A 64-radix, 128-bit, 4-layer Hi-Rise is evaluated in detail using a 32nm technology. It has a throughput of 10.65 Tbps for uniform random traffic, which marks a 15% improvement over a 2D design along with a 33% area reduction, 20% latency reduction, and 38% energy per transaction reduction. For application workloads evaluated on a 64-core processor, Hi- Rise switch improves overall performance by 8% on average over a 2D switch.

In summary, our key contributions are:

- Hi-Rise, an efficient 3D high-radix switch. The proposed switch is a true 3D switch which connects inputs and outputs across different silicon layers.

- Hi-Rise adopts a hierarchical architecture with two internal switches per layer and dedicated layer-to-layer channels, to improve area efficiency, lower delay, and minimize the number of inter-layer TSVs.
- Hi-Rise provides built-in single-cycle arbitration across all inputs and outputs across different silicon layers. This improves efficiency and scalability.
- This thesis proposes a new class based arbitration scheme that is fully integrated into the switching fabric. This scheme makes the 3D hierarchical switch's fairness comparable to that of a flat 2D switch.
- At a radix of 64, Hi-Rise achieves an operating frequency of 2.2GHz, consumes 44pJ of energy per 128-bit transaction and has an area of 0.451mm^2 in 32 nm technology. The proposed switch extends scalability to radix 96 from that of the 64 radix supported by 2D switches at the same operating frequency.

2.2 Background: 2D switch

2.2.1 The 2D swizzle-switch

This section provides a brief background of a high-radix 2D Swizzle-Switch [11][15]. As discussed earlier, unlike a 2D flat switch, a high-radix 3D switch design connects inputs and outputs across multiple layers with both intra-layer and expensive vertical TSV inter-layer connections. The proposed Hi-Rise switch solves the design challenges of a 3D switch, while using the basic concepts of a 2D Swizzle-Switch for its internal switch structures.

A 2D Swizzle-Switch is a matrix type crossbar, with built-in arbitration, optimized for high radix switches. The input and outputs of the switch are placed in a grid fashion. The intersection of the horizontal input bus, with the vertical output bus is termed as a cross-point. A cross-point contains a connectivity bit, which if set, connects its input and output bus. The connectivity bit is set during the arbitration phase. The cross-point also stores a priority vector, containing priority information of its input with respect to all other inputs, for this output. The priority vector is updated based on LRG priority at the end of the arbitration phase.

The arbitration phase begins with each input requesting the outputs with which it wants to communicate. The input data lines are reused to index the outputs during arbitration. The output data lines are also reused as a priority bus during the arbitration phase. One advantage of reusing the output bus for priority lines during arbitration is that the same hardware used for data transfer (pre-charge, pull-down drivers and sense-amps) are reused during arbitration. This allows arbitration to be incorporated into the switch fabric without additional area overhead (since space underneath the cross-point is otherwise largely unused), and guarantees that the arbitration delay is identical to the datapath delay.

Thus, by embedding the logic-dominated arbitration into the wire-dominated crossbar, the 2D Swizzle-Switch allows a compact design and scaling of matrix crossbars to high radices.

2.2.2 Baseline 3D switch: A folded 2D switch

A natural extension of the 2D switch to a 3D stacked implementation is to fold the 2D switch over multiple silicon layers. For this, the inputs and outputs will be redistributed across the layers.

A 64 × 64 2D switch evenly folded across four layers, will result in 16 inputs and 16 outputs on each layer. Since each input still needs to be able to communicate with all outputs, each layer will have a 16 × 64 switch, with 16 outputs connected locally as shown in Figure 2.1. Note that there are still 64 output buses running from layer 1 through layer 4, and that each layer has a cross-point for all 64 outputs. Vertical TSVs go down from each layer to connect the 64 outputs lines between the layers. Essentially, the switch is a single 64 × 64 switch that is folded in the y-dimension across the layers.

In this basic 3D switch design each layer in itself has some benefit in terms of compactness, as instead of 64 nodes each layer has only 16 nodes. This compactness is an inherent benefit of 3D stacking. However, the delay of the switch itself is increased, as shown in Table 2.1, because the wire and device capacitance in the switch remains the same after folding, while the addition of TSVs add to the total capacitance. Also, the number of TSVs required is very high as every output bus wire has to reach every layer.

Overall the folded switch configuration has higher implementation costs in terms of silicon area, delay, and energy over the 2D switch. The folded configuration was proposed and evaluated by Sewell et al. [12], however, their calculations incorrectly identified the number of TSVs required, and switch delay. Table 2.1 reflects the correct calculations. The folded baseline is still a single, radix-64 switch, which requires 64x64 cross-points, unlike the much leaner proposed Hi-Rise switch with hierarchical datapath. The arbitration in a 3D folded switch is identical to that of a 2D Swizzle-Switch, whereas the proposed Hi-Rise switch has a two phase class-based arbitration which composes fairly over inter-layer and intra-layer connections. Our goal is to realize a 3D switch with significantly improved efficiency that takes advantage of the potential that 3D integration affords.

2.3 Hi-Rise 3D switch architecture: Datapath & arbitration

First we focus on the datapath of the proposed Hi-Rise switch followed by details of the arbitration mechanisms in Section 2.3.2.

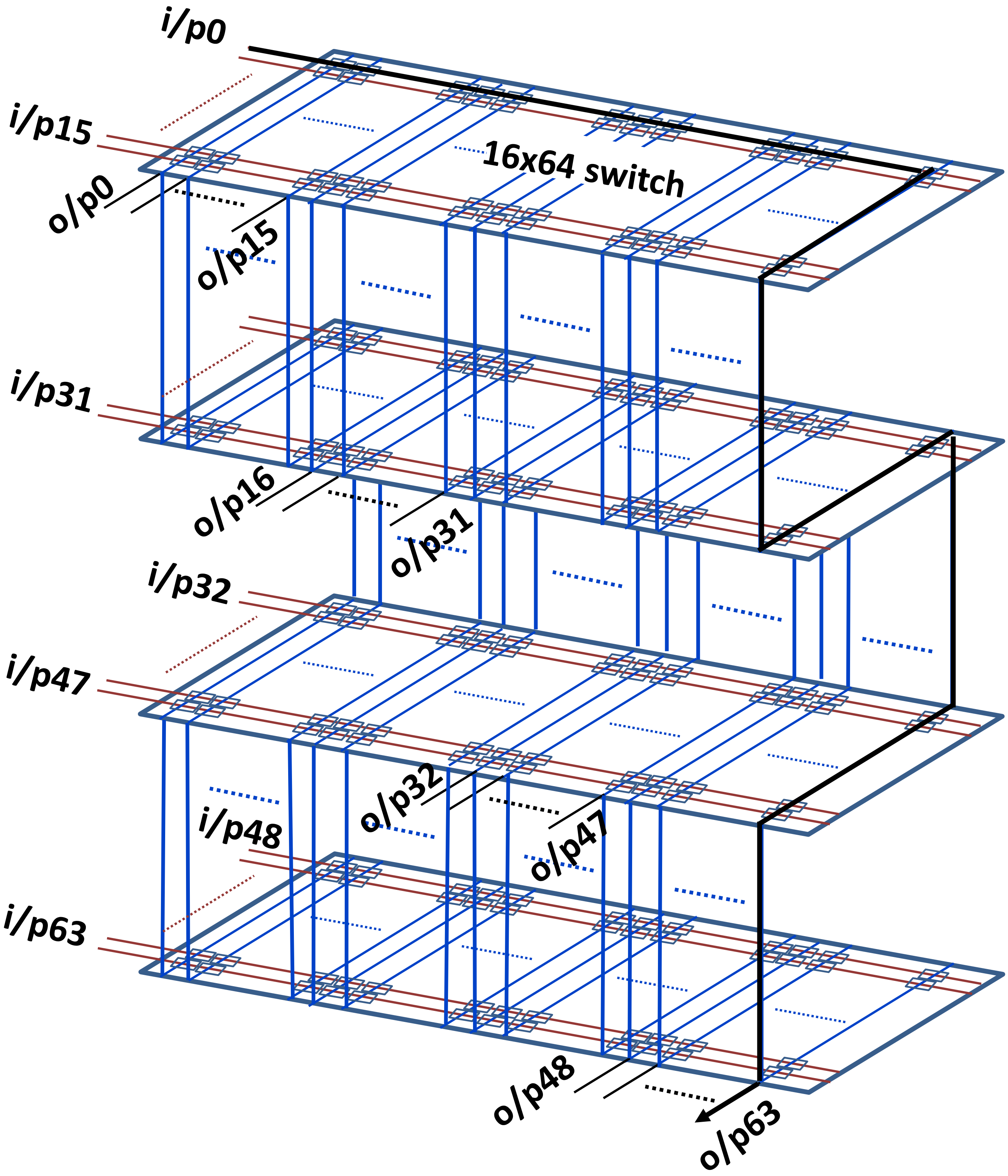


Figure 2.1 A 64-radix 3D folded switch

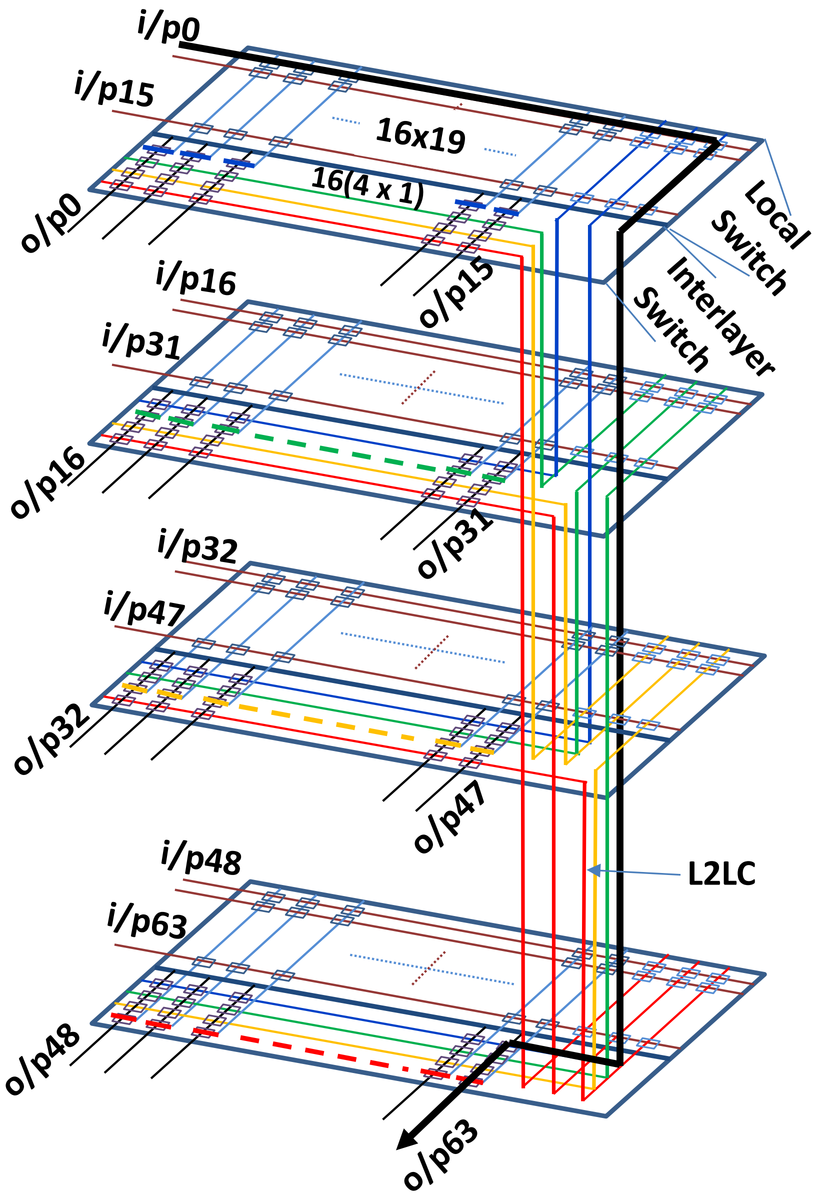


Figure 2.2 Conceptual view of *Hi-Rise* switch

Table 2.1 Implementation cost of 2D versus 3D folded for 64-radix switch

Design	Configuration	Area (mm^2)	Frequency (*GHz*)	Energy/transaction (*pJ/trans*)	Throughput (*Tbps*)	#TSVs
2D	64x64	0.672	**1.69**	71	9.24	0
3DFolded	[16x64]x4	0.705	**1.58**	73	8.86	8192

2.3.1 3D switch datapath

For a switch with radix N, the Hi-Rise switch divides the N inputs and N outputs equally amongst the L layers of stacking. Therefore, at each layer we have N/L inputs and N/L outputs. It provides one or more dedicated vertical layer-to-layer channels (L2LC) from each layer to all other L − 1 layers, as shown in Figure 2.2.

To create a fully connected switch, each input on any layer must be able to arbitrate for, and transmit data to all outputs i.e. outputs on the same layer, and also outputs on every other layer. For this, each layer has two blocks, as shown in Figure 2.3. The first block on a layer, referred to as the local switch, allows inputs to arbitrate for both local intermediate outputs on its layer, and outgoing vertical L2LCs to reach other layers. The second block, the inter-layer switch, is made up of several sub-blocks. Each inter-layer switch sub-block arbitrates between one particular incoming local intermediate output, and all the incoming vertical L2LCs from other layers, and forms the connection to one final output.

We define channel multiplicity as the number of L2LCs between any two layers, which we denote by the variable 'c'. The local switch, has N/L inputs, and both intermediate outputs (N/L) and vertical L2LC outputs (c·(L− 1)). The local switch handles requests from all N/L inputs on a layer, and routes them to the desired layer, which may also be the current layer. The inter-layer switch on a layer has N/L sub-blocks. Each sub-block can connect a unique output to either one of the (c·(L − 1)) vertical channels coming in from other layers or to the unique intermediate output from the local switch on its own layer.

As an example, a 64-radix switch spread across 4 layers of silicon, will have 16 inputs and 16 outputs on each layer for the proposed 3D configuration. If c = 1, i.e. there is only one L2LC between any two layers, as shown in Figure 2.2, then the local switch is a 16×19 switch and the inter-layer switch has 16 sub-blocks of 4 × 1. In the inter-layer switch each final output can choose from 4 inputs—the three incoming L2LCs from the other three layers or the intermediate output bus from the local switch.

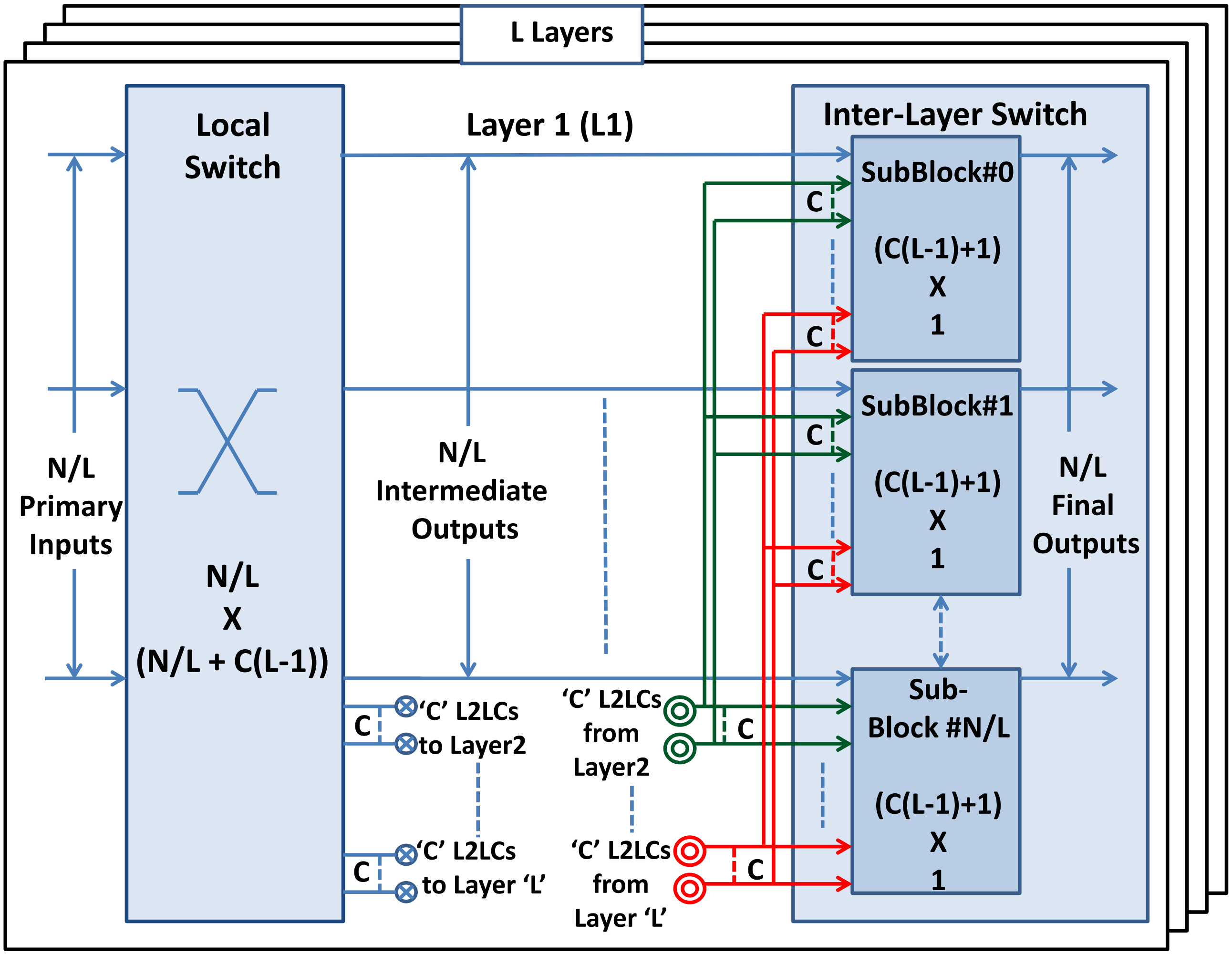


Figure 2.3 One Layer of a generic NxN L-layered *Hi-Rise* with Channel Multiplicity of 'c'

Suppose input 0 from layer 1 (L1) wants to communicate to output 63 on layer 4 (L4). Input 0 has to first win the dedicated L2LC from L1 to L4, by competing against all inputs from its own layer (L1) that want to communicate to L4. This arbitration happens on the local 16 × 19 switch on L1 . Upon winning this L2LC, input 0 gets access to the inter-layer switch on L4. It has to then contend against similar winners from L2 and L3 wanting to communicate with output 63, and also the local contender from within L4 on intermediate output 63. Once the connection has been setup, input 0 can transmit flits across the layer to output 63, using the L2LC.

In the previous example configuration, there was only one vertical L2LC between any two layers. This L2LC is required to service any request from the inputs on L1 to the outputs on L4. In the absence of a strong spatial locality the vertical L2LCs can limit inter-layer traffic, and become bottlenecks. This problem can be solved by increasing the channel multiplicity 'c'. However, the addition of more L2LCs, leads to increased size of both the local switch and the inter-layer switch. The outputs on the local switch grow by $L-1$ for every additional channel. The number of inputs on the inter-layer switch also grows similarly by $L - 1$. For the previous 64-radix example, a switch with $c = 4$ will have a 16×28 local switch, and 16 sub-blocks of 13×1 on the inter-layer switch. For channel multiplicity greater than one, rules are needed to allocate inputs to L2LCs. We discuss below a few possible channel allocation policies.

- **Input binned:** The inputs on a layer are given a fixed, uniform allocation to the L2LCs. In this case, in N radix with L layers and a channel multiplicity of c, each L2LC will service request from $N/(L\times c)$ pre-assigned inputs. These inputs are selected in an interleaved fashion to reduce spatial locality dependence.

- **Output binned:** Output binned is similar to input binning, except it is based on the output.

- **Priority based:** The above two methods of channel allocation may lead to under utilization of the critical vertical L2LCs under certain adversarial traffic as the assignments are fixed. A more efficient utilization can be done by using a priority mux to choose between all N/L inputs. However this method incurs higher delay because arbitration across L2LCs is now serialized.

2.3.2 3D switch arbitration

In this section we discuss the arbitration mechanisms employed for the proposed Hi-Rise switch architecture. Both the local switch and the sub-blocks of the inter-layer switch can have different arbitration schemes that trade overall throughput and design complexity for fairness. The motivation of these schemes is to get as close as possible to the fairness of a 2D flat switch using a Least Recently Granted (LRG) scheme.

1) **Baseline Layer-to-Layer (L-2-L) priority:** This approach applies a simple, independent LRG policy on both the switches on a layer. For a 64-radix, 4-layered switch, the local switch has 16 inputs. The local switch thus maintains a 16-bit LRG priority vector at each cross-point, to arbitrate only between the local inputs to win a local intermediate output channel or an L2LC. Each sub-block on the inter-layer switch will get as inputs, the L2LCs from each of the other three layers as well as an intermediate output on its layer; hence it only needs a small priority vector. For a 4 layer switch with channel multiplicity of one, a 4:1 LRG arbitration is required on each of the inter-layer sub-blocks. The inter-layer switch follows the standard procedure; its priority is updated after every arbitration cycle. The priorities are updated at the local-switch only if it wins the final output. The local switch priority update is triggered by the winner at the inter-layer switch, and is back-propagated to the winner's local switch. This ensures that an input request always gets serviced, as its priority will rise on the inter-layer switch in subsequent arbitrations while remaining at the same priority on the local switch, thus, avoiding the possibility of starvation.

2) **Unfairness with baseline L-2-L LRG priority:** The baseline arbitration performs well for uniform random traffic. But, as the traffic requests to a particular output becomes more biased from a particular layer, the latency to service requests can become long.

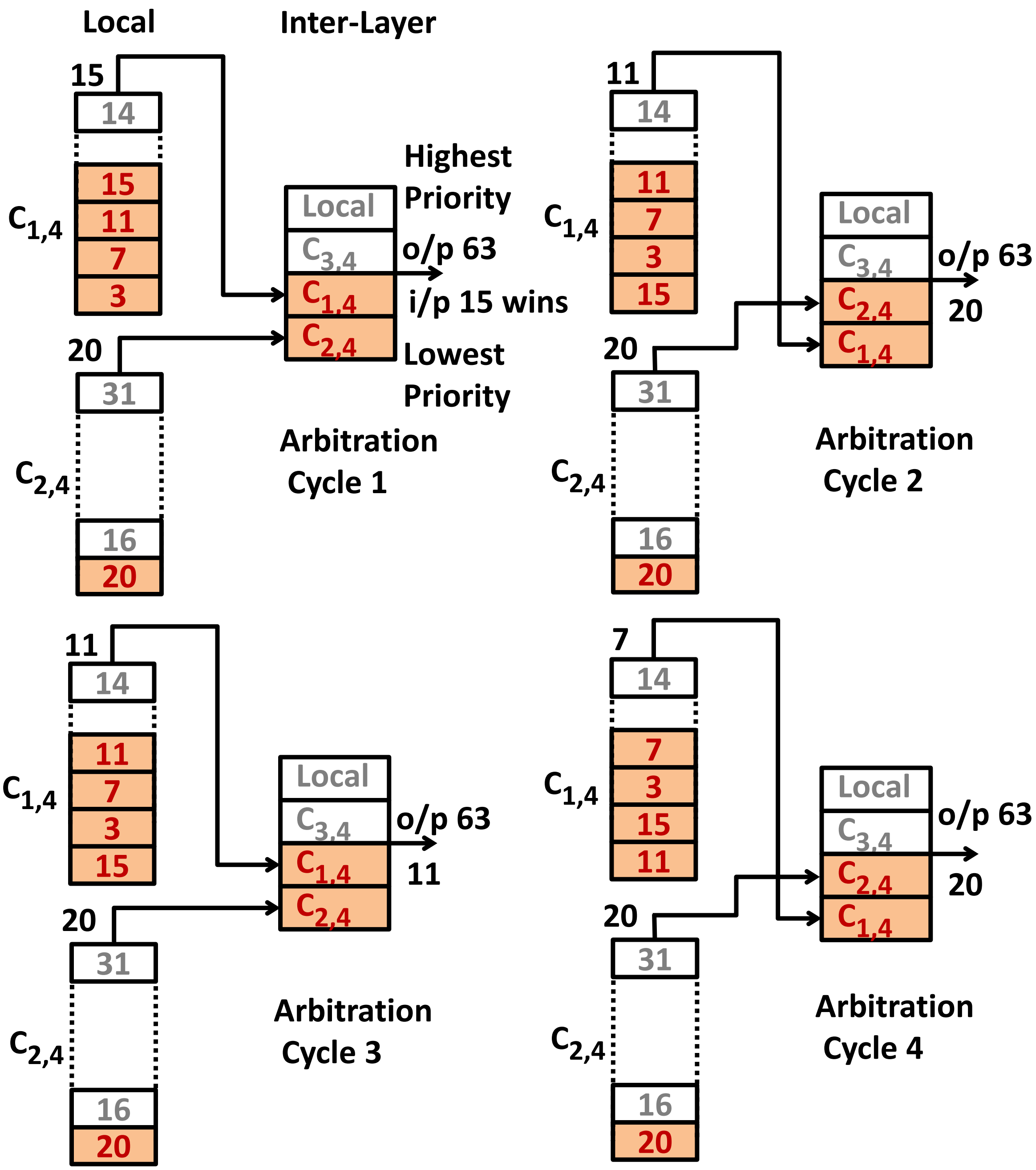


Figure 2.4 Baseline Layer-to-Layer (L-2-L LRG) example.

We will illustrate this with the following example: a 1-channel 4-layer 64-radix configuration where 4 inputs, {3, 7, 11, 15} from the first layer (L1), and one input, {20} from the second layer (L2), are all requesting output 63 on layer four (L4). As shown in Figure 2.4, all four inputs requesting from L1 use the same L2LC going from L1 to L4, denoted as $C_{1,4}$. The four inputs compete with each other during arbitration at L1's local switch. On the other hand, input 20 is the only requester from L2, thus it always wins the local arbitration on its layer for $C_{2,4}$. At the sub-block of the inter-layer switch on L4 belonging to output 63, the winners of the local switch arbitrations compete for the final output.

Figure 2.4 walks through four cycles of arbitration for this example. The LRG priorities decrease from top to bottom. In arbitration cycle 1, input 15 wins $C_{1,4}$ on the L1 local switch. Input 20, which is the lone contender for $C_{2,4}$, wins on the L2 local switch. The two local winners contend on the inter-layer switch of L4 where output 63 is being arbitrated. Input 15 wins as $C_{1,4}$ has higher priority than $C_{2,4}$. This is followed by an LRG update at both the sub-block of the inter-layer switch L4, and at the local switch on L1.

In the subsequent arbitration cycle, input 11 and input 20 contend but input 20 wins, as $C_{2,4}$ now has higher priority than $C_{1,4}$. The LRG of $C_{1,4}$ remains unchanged as it did not win at the inter-layer switch. Thus, in the next arbitration cycle input 11 again gets to contend and wins against input20. The pattern continues, with one input amongst the four contenders on L1 winning, followed by the only contender from L2. The connection formed over time at output 63 is {15, 20, 11, 20, 7, 20, 3, 20, 15, 20 ...}. This pattern shows that the layer with the fewest number of contenders is able to access the output more frequently and the arbitration is unfair. In a 2D flat switch with LRG the output patter would be {20, 15, 11, 7, 3, 20, 15 ...}. The observation is that the baseline L-2-L LRG arbitration will be unfair, whenever multiple L2LCs contending for a single output have disparate number of requestors.

3) Weighted LRG priority: To resolve the unfairness problem, the arbitration policy in the sub-blocks of the inter-layer switch needs to be modified. Weighted LRG (WLRG) arbitration scheme is a possible solution and is based on the intuition that L2LCs with higher traffic need to have higher priority. This can be achieved by freezing the LRG priorities for multiple cycles on the inter-layer switch sub-block when an L2LC has more than one requestor. The proportion of arbitration cycles for which the LRG is held, the weights, is determined by the number of requestors the L2LC represents.

Weights are generated by the local switch by counting the number of requestors. Weight information is then transmitted from the local switch to the inter-layer switch along with the request vector, and stored in a counter.

Calculating the number of requestors, involves counting the number of parallel requestors for an L2LC, which is hard to implement in hardware in a single-cycle. It makes the arbitration phase much longer, and hence slows down the cycle time for WLRG considerably. Furthermore, for a 3D switch the WLRG scheme becomes prohibitive due to the large amount of information (weights) that needs to be transmitted from the local switch to the interlayer switch over the L2LC.

4) Class-based Least Recently Granted (CLRG) priority: To improve the fairness over the baseline L-2-L LRG without having to compute and transmit weights to the inter-layer switch, CLRG priority scheme is proposed.

In this scheme, at the inter-layer switch a counter is maintained for each input-output pair. By keeping track of all inputs (across all layers) at the inter-layer sub-block cross-points, the switch can be made fair. This counter tracks the number of times the specific primary input won the arbitration for a particular final output. The arbitration scheme at the inter-layer switch uses this count as a coarse priority, dividing the inputs into different subsets called classes. A bigger count value for an input signifies that the input has had a larger share of the bandwidth for this output, and it is relegated to a lower priority class. The inter-layer switch thus allows the contender with the least count to win. However, if input contenders belong to the same class, CLRG uses layer-to-layer LRG for tie-breaking.

To keep the counter based arbitration logic small, and to avoid cases where bursty traffic penalizes an input for a long time after the burst, the counter is kept short. The number of classes (counter length) required is a heuristic that needs to be tuned. Whenever any counter saturates in a sub-block on the inter-layer switch, all 64 input counters for that sub-block are divided by 2. This maintains the relative class ordering between inputs.

Revisiting the 1-channel example, the counters of all the inputs will be initialized to 0, placing them in the highest priority class 'P0' as shown in Figure 2.5. In arbitration cycle one, input 20 is the only contender from L2's local switch, and hence wins $C_{2,4}$ (the L2LC to layer L4). The LRG at L1's local switch has input 15 as the highest priority requesting input, and hence input 15 wins the arbitration for$C_{1,4}$. As both input 20 and input 15 are in priority class P0, LRG is used to tie-break. Input 20 wins, as $C_{2,4}$ has higher LRG priority than $C_{1,4}$. On winning the arbitration, input 20 increments its counter and moves to the lower priority classP1. In arbitration cycle 2 input 15 again contends against input 20. This time input 15 has class P0 and input 20 has class P1, therefore the switch employs class-based priority to make input 15 the winner. Even though LRG is not used for this arbitration cycle, it is still updated.

In arbitration cycle three, input 11 and input 20 contend and input 11 wins by virtue of its class, even though input 20 has a higher LRG priority. This is followed by input 7and input 3 winning against input 20 as they are in class P0, while 20 is in P1. Now all requesting inputs have a count of 1, i.e., all are in class P1. In arbitration cycle 6, input 20 wins again on the basis of LRG tie-breaking. The sequence of winners for the class based arbitration will be {20, 15, 11, 7, 3, 20, 15, 11, 7, 3, 20 ...}.

This is similar to the pattern that will be followed in a single flat 2D LRG switch. Therefore, this scheme is able to resolve the fairness issue of the baseline scheme and also has an efficient single cycle hardware implementation, as we will see in the next section.

2.4 Implementation

To design a high-radix switch with area-energy efficiency, the local and inter-layer sub-blocks are designed similar to a Swizzle-Switch. Swizzle-Switch is implemented by tiling together cross-points. This section details the cross-point design for both the local switch, and the inter-layer sub-blocks. It also details how we integrate the proposed CLRG logic within a cross-point.

2.4.1 Basic cross-point design

Recall, a cross-point connects an input to an output port and each cross-point contains both the connectivity and the arbitration logic. A 2D switch of radix N contains N × N cross-points, where each cross-point has a N-bit priority vector. The proposed Hi-Rise switch has three types of cross-points, intermediate output cross-points on the local switch, L2LC output cross-points on the local switch, and cross-points on the inter-layer sub-blocks.

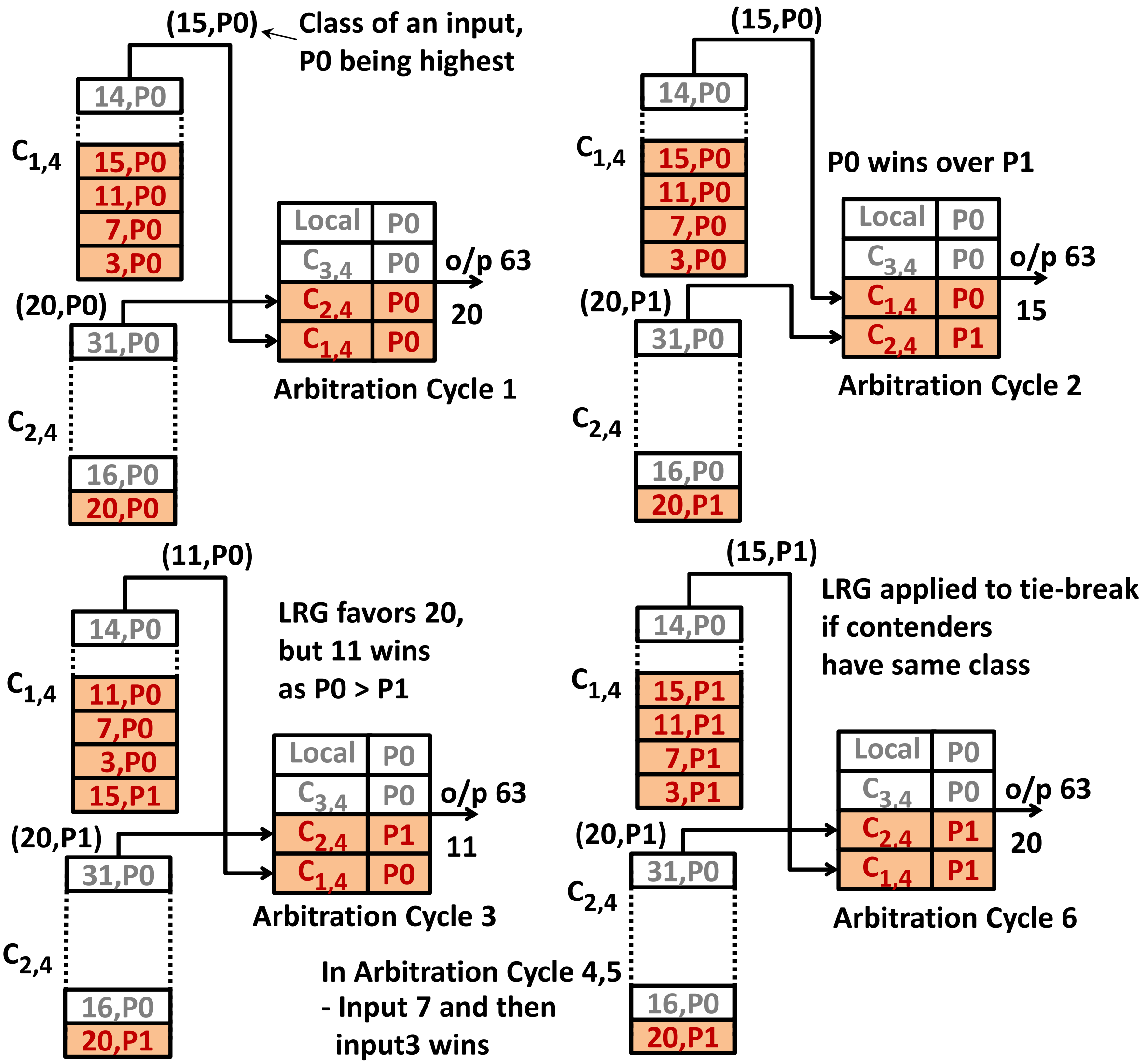


Figure 2.5 Class-based Least Recently Granted (CLRG) arbitration example.

The underlying circuit for all three types of cross-points is similar to that of a 2Dswitch. The circuit schematic for the first type of cross-point, the intermediate output cross-point, is shown in Figure 2.6. The only difference from a 2D switch cross-point is that it has a N/L-bit priority vector, and two extra output bit-lines to transmit the request and the release signals to the inter-layer switch.

The second type of cross-point, L2LC output cross-point, builds upon the intermediate output cross-point. The L2LC cross-point differs from the intermediate output cross-point in two respects. First, the priority vector size is N/L in the intermediate output cross-point, whereas, it is only (N/(L ∗c)) for the input binned L2LC cross-point. Second, the L2LCcross-point transmits the request vector of the input to the inter-layer switch during the arbitration phase, by setting high the L2LC wire with the requested output's index. This is because the L2LC output can request any of the N/L outputs on the destination inter-layer switch. On the other hand, the local intermediate output is dedicated to a single final output on the inter-layer switch.

The third type of cross-point, inter-layer sub-block cross-point, is discussed in the next section.

2.4.2 Arbitration specific cross-point design

The inter-layer sub-block cross-point structure is dependent on the arbitration scheme employed. For the baseline L-2-L LRG, the inter-layer cross-point is very similar to the basic 2D cross-point with only the priority vector size changed. Below we discuss the implementation of the Class-based LRG cross-point.

CLRG cross-point design: The CLRG technique does not require any additional logic in the local switch cross-points. However, the inter-layer cross-point is modified to enable the class-based scheme.

Figure 2.7 shows a single cross-point within an inter-layer switch for an input binned 4-channel 4-layer 64-radix configuration. This configuration has 13 cross-points (corresponding to 12 L2LCs and a local input) in a sub-block.

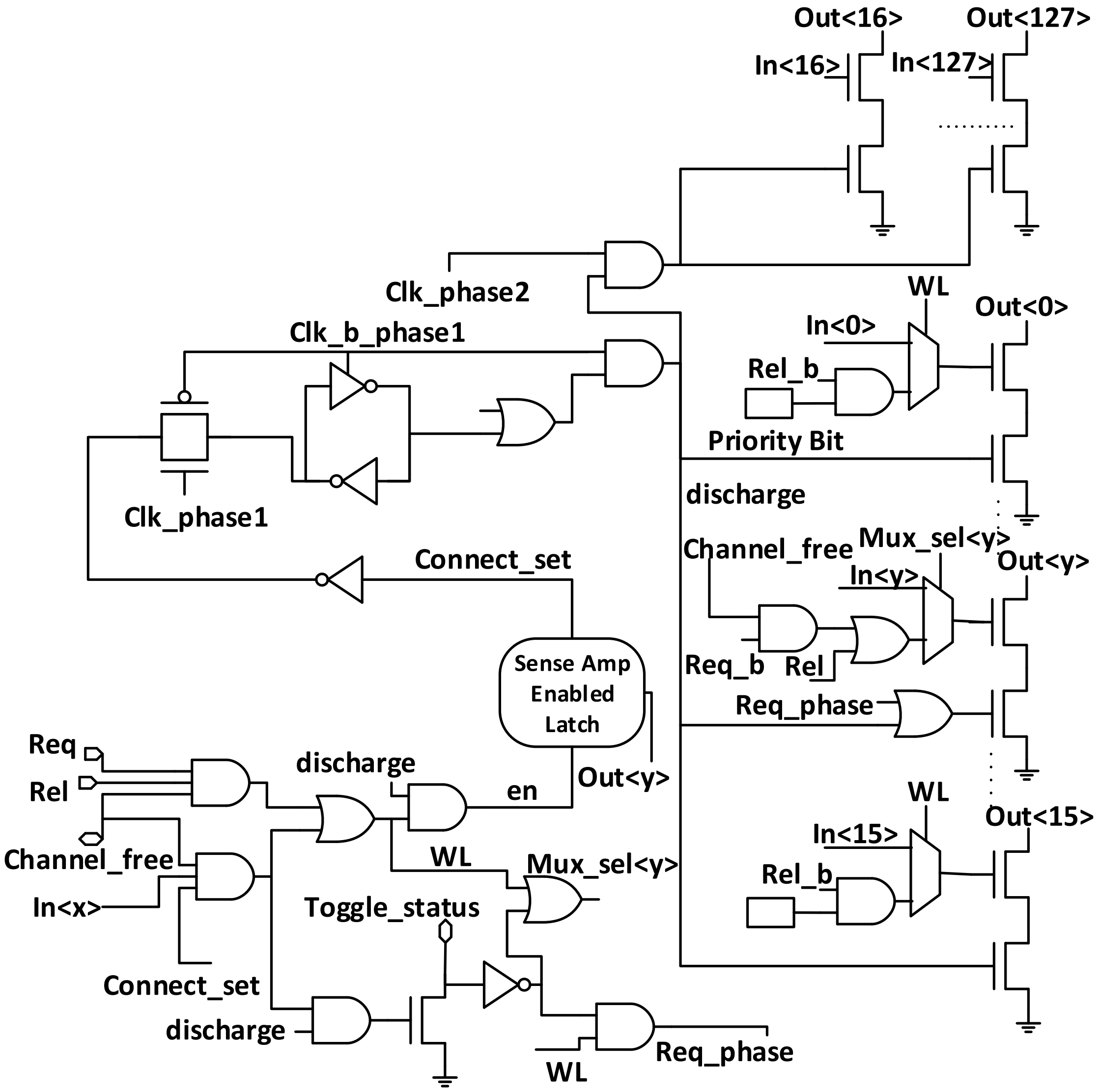


Figure 2.6 Circuit schematic of local intermediate output cross-point

Each cross-point provides connectivity for an L2LC, which in turn is associated with four primary inputs. For each of these four primary inputs, a thermometer class counter is placed within the inter-layer switch cross-point. We find empirically that three classes provide reasonable fairness for a 64-radix Hi-Rise switch, hence we use a thermometer counter with the following sequence {00, 01, 11}.

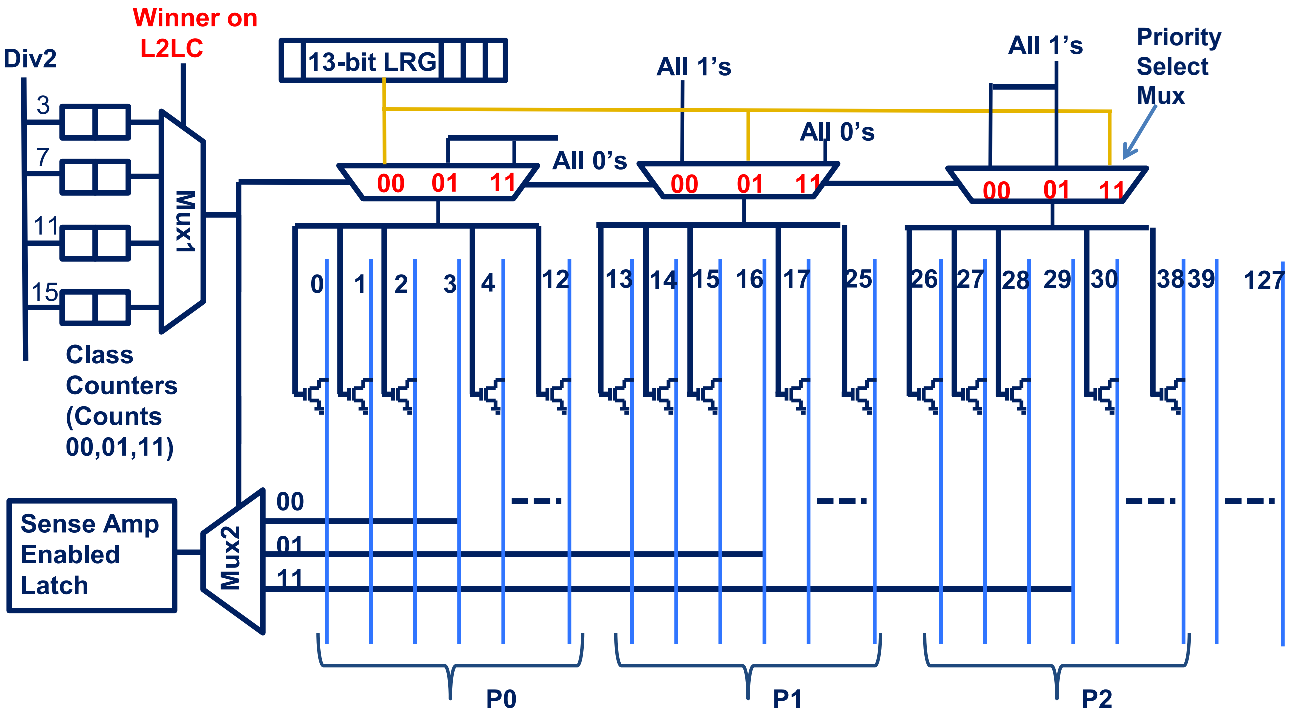


Figure 2.7 Conceptual view of a cross-point at inter-layer sub-block for CLRG arbitration

The counter for a primary input, keeps track of how many times that input won the arbitration for the final output. During the arbitration phase the counter value of the primary input which wins the L2LC, is chosen using a multiplexer (Mux1). This counter value is used for setting up other multiplexer as shown in Figure 2.7.

The arbitration circuit shown in the figure enables class based arbitration along with LRG tie breaking in a single cycle. The output lines are reused as priority lines during the arbitration phase. The priority lines are grouped class-wise, where each group has priority lines for each of the 13 L2LCs. Priority Class '00' uses wires 0-12, priority class '01' uses wires 13-25, and priority class '11' uses wires26-38 as shown in the figure.

Each cross-point has three Priority Select Multiplexers (PSMs) as shown in Figure 2.7. The PSMs apply '1' to all priority lines belonging to a lower priority class, so that any request from a lower-priority class is inhibited. The PSMs apply ‘0’ to all priority lines belonging to a higher priority class, so that it does not affect the arbitration for the higher priority class. The PSM applies the LRG priority vector to the priority lines belonging to its own class.

The arbitration circuit thus allows L2LC with a higher priority class to pull-down the priority lines being polled by the L2LCs with a lower priority class. The L2LC in the highest priority class thus wins. However, if multiple L2LCs in the highest priority class are requesting, then they pull-down priority lines of L2LCs with lower LRG priority in their own class.

Each L2LC polls one priority line in each of the three priority class. The multiplexer (M ux2) as shown in Figure 2.7 is used to select one of the three lines based on class counter. The polled value goes to a sense-amplifier enabled latch, which is the connectivity bit. Once this bit is set, the cross-point connects the input data to the output lines. The winning primary input also increments its corresponding counter.

2.4.3 Clocking of Hi-Rise switch

As shown in Figure 2.8, the Hi-Rise switch uses two-phase clocking. In the first phase the local switch evaluates and transmits the outputs to the inter-layer’s inputs. The inter-layer switch stays pre-charged until the outputs of all the local switches have been evaluated and stabilized. In the second phase, the inter-layer switch evaluates and generates the final output. Intermediate outputs are not latched at the local switch outputs.

2.4.4 Physical implementation

The cross-point layout for the circuit schematic shown in Figure 2.6, has horizontal input bus metal wires, and vertical output bus metal wires, with logic underneath. The 3D local switch cross-points and inter-layer switch cross-points for baseline have fewer priority bits than in a 2D cross-point, so the logic area is significantly lower. The area is thus wire limited and the switch can be extended to higher radices before the logic becomes dominant. For CLRG arbitration, due to the additional counters in each cross-point, the inter-layer cross-point's gate count is comparable to the 2D cross-point. To reduce switch area, wires are stacked using two metal layers in each direction. To reduce coupling between wires, double pitch spacing is used.

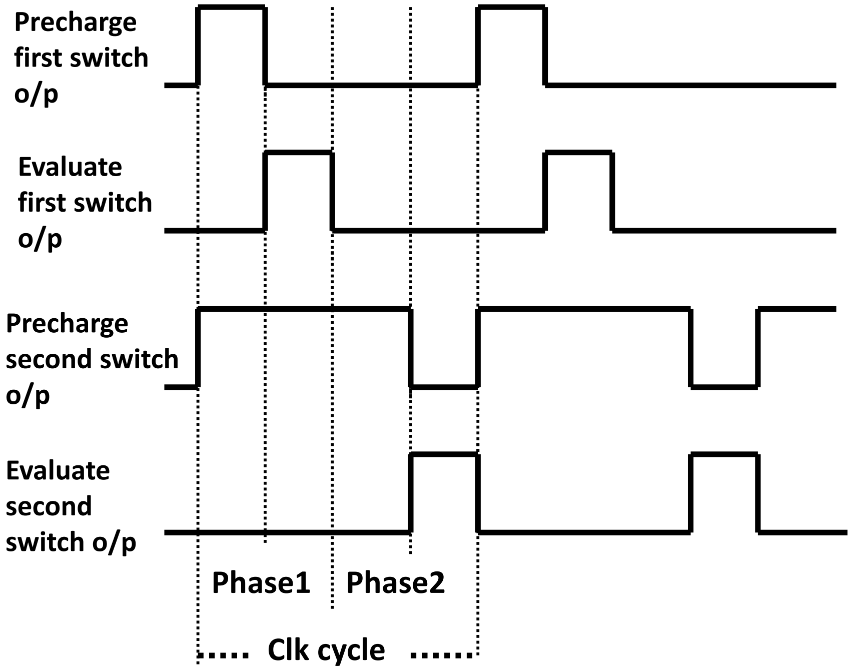

Precharge first switch o/p
Evaluate first switch o/p
Precharge second switch o/p
Evaluate second switch o/p
Phase1
Phase2
..... Clk cycle ......

Figure 2.8 Phase-wise clocking of *Hi-Rise* switch.

The TSV used for evaluation has a 0.8μ minimum pitch with 0.2fF feed-through capacitance and 1.5ohm resistance [16]. Section 2.6.3 studies TSV parameters and the impact of TSV size on the performance and area of the proposed Hi-Rise switch. The TSVs are all located in the local switch. The 4-channel input binned configuration has only four cross-points out of a column of 16, leaving plenty of empty space to place the TSVs without any area increase.

2.5 Methodology

To evaluate the various interconnect performance characteristics, a cycle accurate network simulator is used. To accurately model circuit implementation, C models are written for each of the different switch configurations and arbitration schemes described in Sections 2.3.1 and 2.3.2. The baseline design is a 2D 64 × 64 switch.

The synthetic traffic patterns used to evaluate the various switch configurations are uniform random, hot spot and bursty. Custom synthetic traffic patterns are also used to evaluate specific corner cases and adversarial cases for the proposed switch configurations. For synthetic patterns, the simulator uses 4 virtual channels at each port with a buffer depth of 4 flits per virtual channel. Each flit has a size of 128 bits to match the databus width, and 4 flit packets have been used for simulations.

Table 2.2 Spice condition and TSV parameters

Spice Conditions	Process = Typical	Temperature = 27C	Voltage = 1V
TSV Parameters	Pitch = 0.8 *μm*	Feed-Through Cap. = 0.2 *fF*	Resistance = 1.5 *ohm*

Spice models for both the baseline arbitration scheme and the CLRG arbitration scheme are created based on their cross-point implementation. The spice models for the switch are verified against the 2D Swizzle-Switch silicon results. The spice models are in a commercial 32 nm SOI technology. These models are then used to determine the area, speed and energy for the entire switch. The spice netlist accurately models the effect of wire routing, with appropriate length wire models of the correct metal layers used. The spacing between the wires is double-pitched to avoid capacitive coupling. Physical implementation details, like using multiple metal layers stacking for input and output routing to reduce wire lengths are also considered. The spice model accounts for the capacitive loading of the TSVs, and also the routing to and from the TSV. The TSV parameters and the Spice PVT conditions used for evaluations are as shown in the Table 2.2.

To run real application workloads, a trace-driven, cycle-accurate many-core simulator [17] is integrated with a system built out of a single Hi-Rise switch, cores, caches and memory controller models. The system parameters used for application workloads is shown in Table 2.3. A front-end functional simulator based on Pin [18] is used to collect instruction traces from applications, which are then fed into the cycle-level simulator. We study a diverse set of benchmarks, including SPEC CPU2006 [19] benchmarks, and four commercial workload traces (sap, tpcw, sjbb, sjas).

Table 2.3 Processor configuration for application workloads

Cores	64 cores , 2-way out-of-order, 2 GHz frequency
L1 Caches	32 KB per-core, private, 4-way set associative, 64B blocks, 2-cycle latency, split I/D caches, 32 MSHRs
L2 Caches	64 banks, 256KB per bank, shared, 16-way set associative, 64B block size, 6-cycle latency, 32 MSHRs
Main Memory	8 on-chip memory controllers, 4 DDR channels each @16GB/s, up to 16 outstanding requests per core, 80ns access latency

2.6 Results

The proposed 3D switch has several design parameters that can be tuned to gain better performance, increased scalability, and reduced implementation cost. In Section 2.4.1, we first study the datapath of the proposed 3D switch and find the optimal point of operation with respect to both the network characteristics and the implementation cost. In Section 2.4.2, we evaluate the different built-in arbitration schemes. Section 2.4.3 analyzes the sensitivity to the TSV technology. Finally, we present the results for the application workloads in Section 2.4.4.

2.6.1 Analysis of 3D switch datapath parameters

This section will first discuss how we optimize Hi-Rise switch for speed. We then compare network characteristics for uniform random (UR) traffic. The goal is to find a configuration with high saturation throughput, low latency and high speed of operation.

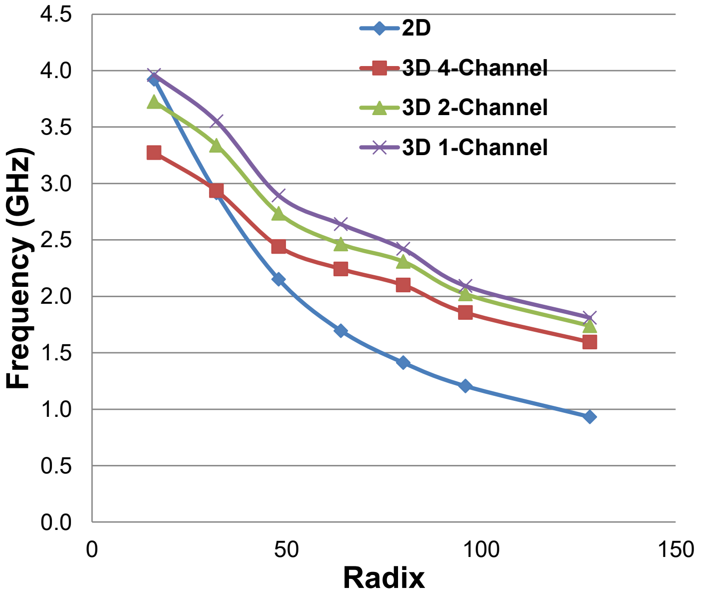


Figure 2.9 Hi-Rise switch frequency vs radix

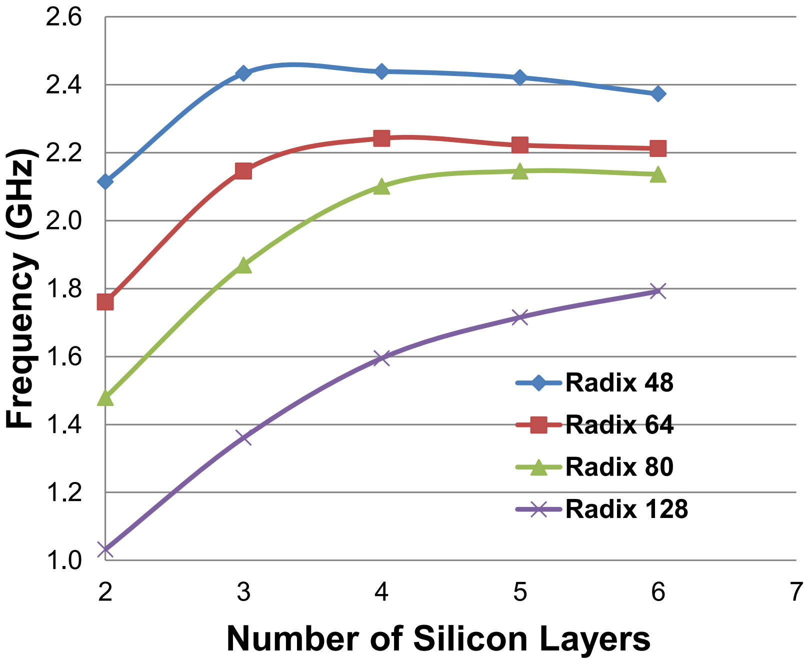


Figure 2.10 Frequency vs number of silicon layers stacked

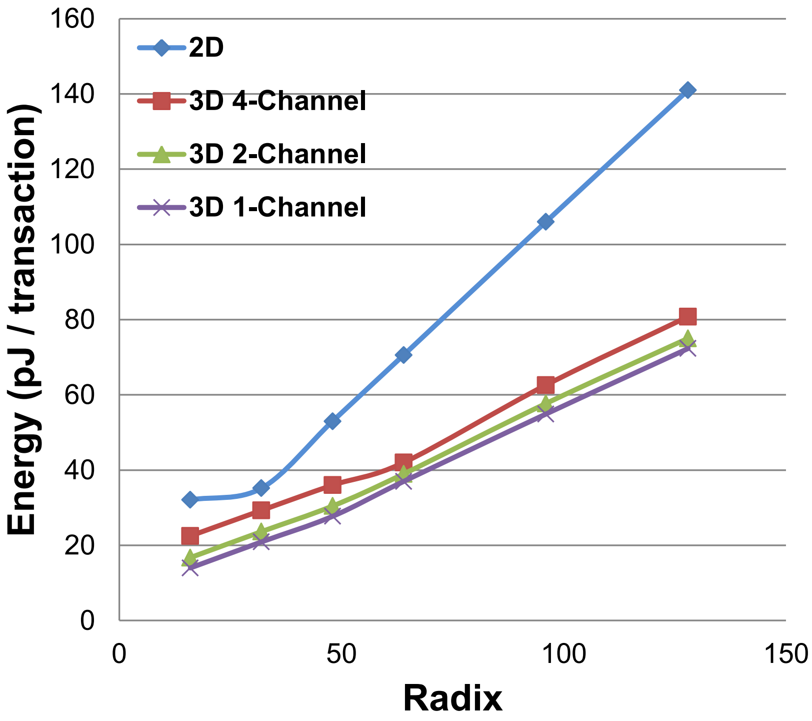


Figure 2.11 Energy per transaction (128-bit) for 2D and 3D switch.

In the proposed 3D switch, frequency is a function of the radix of the switch and the number of layers stacked. The L2LC multiplicity ‘c’ is also a factor, as increased ‘c’ causes both the local switches and inter-layer switches to grow in size. Figure 2.9 shows the frequency for different radices of a 4-layered 3D switch, and the 2D Swizzle-Switch. The 2D switch has a better frequency at low radix, as the overheads incurred by the hierarchical architecture makes the 3D switch slow. Beyond radix 32, all 3D configurations have a better speed than 2D. As the radix increases the frequency gap widens, making the 3D switch more favorable. As radix increases, the channel multiplicity also becomes less of a factor, as can be seen from the converging 1, 2 and 4 channel frequency plots.

The number of silicon layers stacked is another factor that changes the frequency significantly. At a low number of stacked layers, the switches on each layer are still large, so the frequency will be low. However, if we have too many layers, the numbers of L2LCs increase, and become the dominating factor. Therefore, the number of stacked layers in a switch has an optimal point. As seen in Figure 2.10, for a 64-radix 3D switch the frequency is maximum in the range of 3 to 5 layers and then decreases on either side. At small radices, the optimum number of layers required is lower, whereas for higher radices the optimum point shifts towards higher number of stacked layers. We use a 3D 64-radix switch as a fair comparison, because the 2D Swizzle-Switch scales well until 64-radix.

As seen in Figure 2.10, for 64-radix the optimal number of stacked layers is 4. The 4-layer 64-radix 3D switch can still have different channel multiplicity numbers. Lower channel multiplicity will have lower overhead, but may not provide sufficient throughput. The network throughput for channel multiplicity of 1, 2 and 4 are listed in Table 2.4. The latency curves are shown in Figure 2.12. The 3D one-channel configuration performs poorly and saturates at very low injection rates. The configuration with channel multiplicity of 2 is only 19% worse than a 2D flat switch's throughput, while the configuration with channel multiplicity of 4 has 18% better throughput than the 2D switch. The proposed 4-channel 4-layer 64-radix 3D switch has a saturation throughput of 21.42 packets/ns or 10.97 Tbps for uniform random traffic. Also, the zero-load latency for proposed 3D configurations is about 20% better than 2D. So even at low injection rates the 3D will outperform the 2D switch. The naive folded implementation, on the other hand, has 7% less saturation throughput than a 2D flat switch.

The energy consumed per 128-bit transaction is also an important metric for switch performance. The compactness of the Hi-Rise switch makes it more energy efficient. Figure 2.11 shows the energy per transaction as the radix increases. The 3D switch energy increases at a more gradual slope as compared to a 2D switch, allowing it to have a significantly higher radix switch for iso-energy.

Table 2.4 Implementation cost of different 64-radix switch implementations

Design	Configuration	Area (*mm*2)	Frequency (*GHz*)	Energy/ transaction (*pJ/trans*)	Throughput (*T bps*)	#TSVs
2D	64×64	0.672	1.69	71	9.24	0
3D Folded	[16×64]×4	0.705	1.58	73	8.86	8192
3D 4-Channel	[(16×28), 16·(13×1)]×4	0.451	2.24	42	10.97	6144
3D 2-Channel	[(16×22), 16·(7×1)]×4	0.315	2.46	39	7.65	3072
3D 1-Channel	[(16×19), 16·(4×1)]×4	0.247	2.64	37	4.27	1536

The implementation cost for the various channel configurations is shown in Table 2.4. The proposed 3D switch's hierarchical structure leads to much smaller switches, and hence the large implementation cost benefits over a 2D Swizzle-Switch. These switches are smaller not just in their dimension but also in the gate count. The 4-Channel 3D switch has a 40% lower energy requirement than the 2D switch, and occupies 33% less area. Also, the implementation cost of the 4-Channel 3D switch is not significantly higher than the 2-Channel 3D switch. From both implementation cost and channel multiplicity traffic study, we choose the 4-channel 4-layer 64-radix Hi-Rise switch as the optimal configuration for all further analysis.

2.6.2 Analysis of 3D switch arbitration schemes

In Section 2.3.2 three arbitration schemes were discussed: baseline layer-to-layer LRG (L2L LRG); Weighted LRG (WLRG); and, Class-based LRG (CLRG). The goal of the proposed arbitration schemes is to make the switch fair. In this section we present the results of analyzing fairness for the various traffic patterns. We also present the implementation cost for these arbitration schemes in hardware.

Hotspot traffic helps bring out the fairness issue in the baseline L2L LRG. Hotspot traffic involves all inputs requesting the same output. The pattern used in this experiment involves all inputs from layers 1, 2, 3 and 4, requesting for output 63. Figure 2.13 shows the average latency for inputs 0 to 63 in cycles. The load rate used in this experiment is 80% of the saturation load rate for hotspot traffic.

In 3D 4-channel 4-layer configuration, any inter-layer sub-block has one connection for local intermediate output and twelve L2LC connections. This causes the 3D L2L LRG arbitration to show a wide deviation between the latency for local inputs, and the latency for other layer inputs. Since, all requests are for the same output in hotspot traffic, the local intermediate output is arbitrated for by 16 primary inputs, while each L2LC is arbitrated for by 4 primary inputs only. Thus, L2L LRG effectively allots only 1/4th of the bandwidth to the local intermediate output as compared to other layers. This is evident from the high latency for the local inputs {48 to 63} in the Figure 2.13.

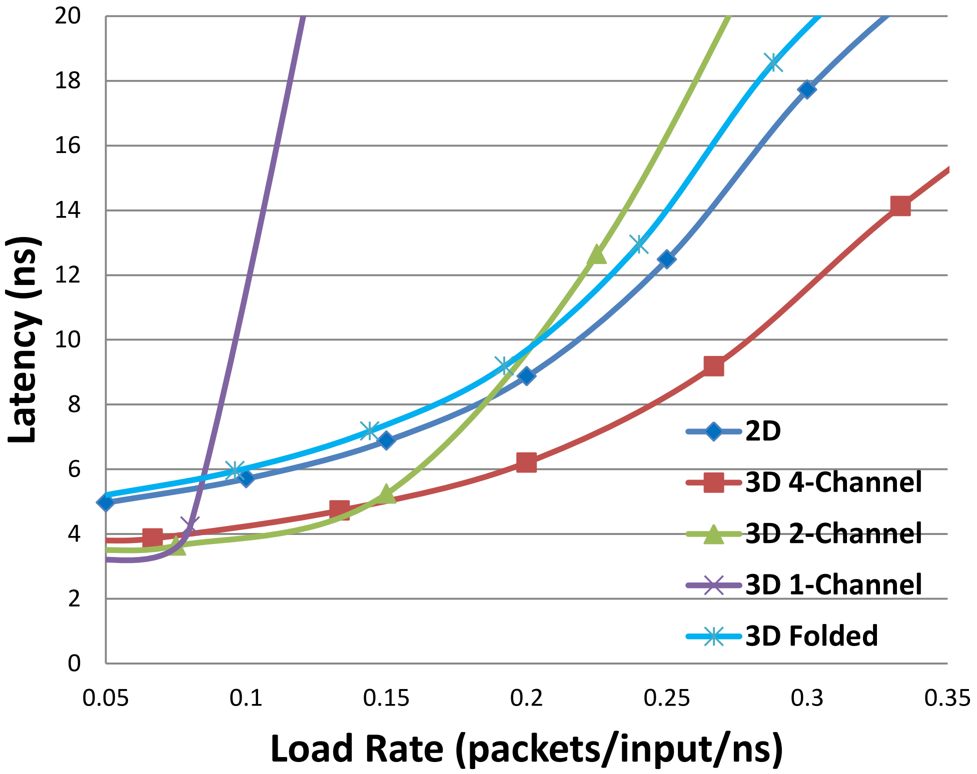


Figure 2.12 Latency of 2D and 3D multi-channel configurations for UR traffic

In the CLRG scheme the unfairness is resolved. Initially all inputs have a count of 0, i.e., the highest priority class. The other layer inputs get through faster initially, as only 4 primary inputs contend for a L2LC. But as they keep winning, they are relegated to a lower priority class, thus elevating the priority of the non-serviced requests from the local layer. In the case of hotspot traffic, every primary input will reach a count of 1 before anyone gets to transmit again. Hence, class-based arbitration behaves similar to flat LRG. Comparison of the average throughput for different arbitration schemes with uniform random traffic is shown in Figure 2.14. For uniform random traffic, even the 3D L2L LRG arbitration scheme behaves in an unbiased manner. Hence, the performance for uniform random traffic mainly depends on the frequency of operation for the switches. The 3D L2L LRG, due to its design simplicity, is able to run marginally faster than the CLRG arbitration. Also, all the 3D arbitration schemes are still considerably faster than a 2D flat switch, as seen from the frequency numbers in Table 2.5. Thus, the throughput of CLRG is slightly lower than the 3D L2L LRG, but compared to a 2D switch, they have 15% better throughput. The latency of all the three schemes is very similar, and all the schemes have a zero-load latency that is about 20% better than a 2D Swizzle-Switch.

We study the latency and throughput for the adversarial traffic pattern, which was used as the example in Section 2.3.2. The pattern consists of five requesting inputs, four inputs {3, 7, 11, 15} from L1, and input {20} from L2. All five inputs are requesting output 63 on L4. For this pattern, the L2L LRG shows a wide disparity between throughputs of input 20 versus the throughput of the other four inputs. This is shown in Figure 2.15. Both the WLRG and the CLRG arbitration schemes are able to resolve this bias as explained in Section 2.3.2.

A pathological case for the 3D switch is when we have only inter-layer traffic, but no within-layer traffic. In this case, the throughput is limited by the bandwidth available through the L2LCs between any two layers. The throughput for such traffic is not improved by the different arbitration schemes. The worst case scenario is, all the four inputs using the same L2LC, request for different outputs on another layer. In this corner case, the throughput of the 3D switch can get limited up to 1/4th of the flat 2D switch.

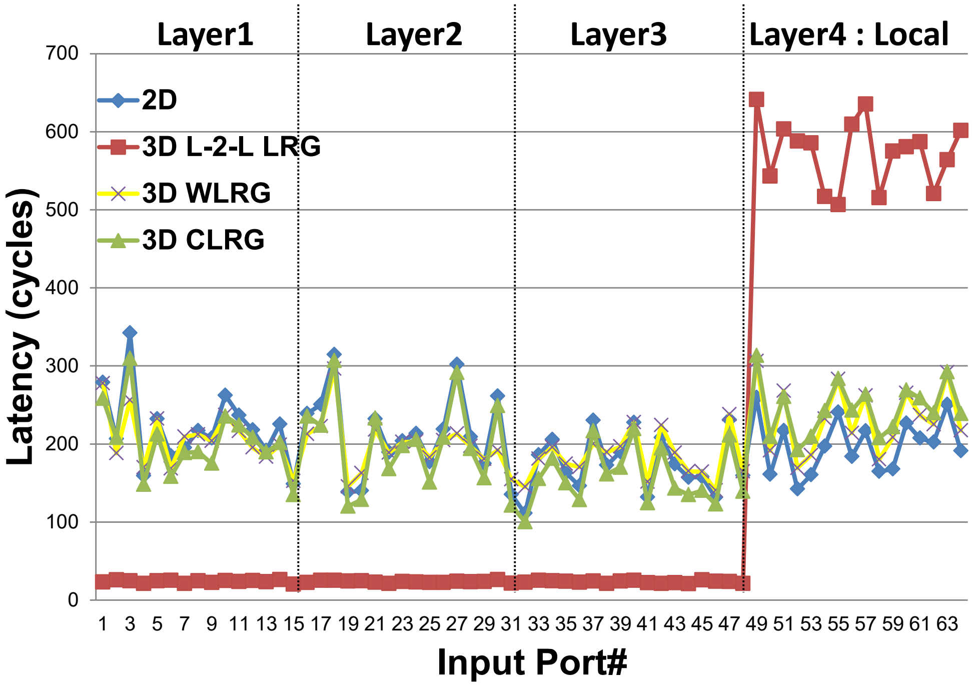


Figure 2.13 Latency of each input for hotspot traffic

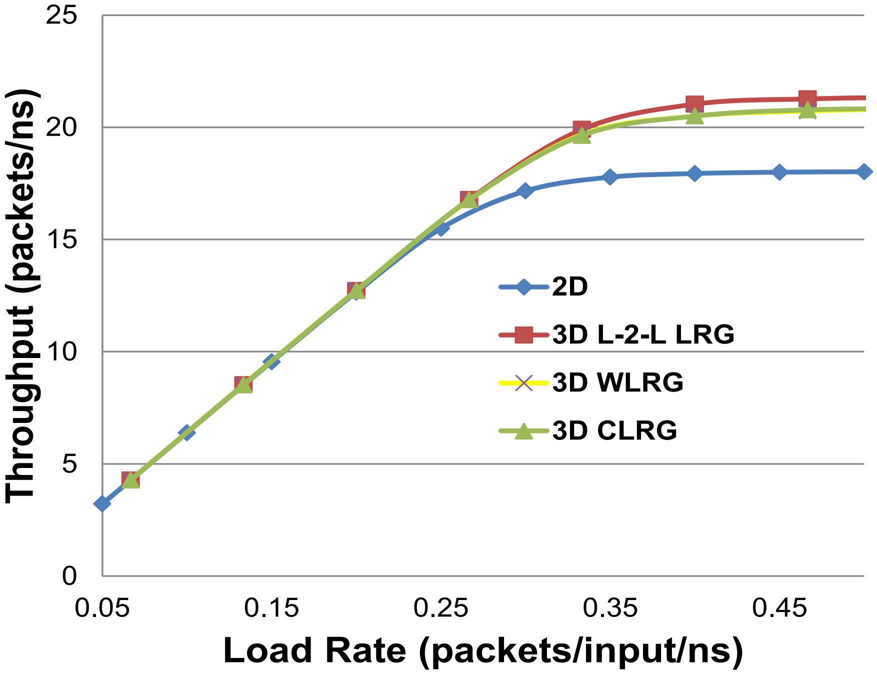


Figure 2.14 Throughput of arbitration schemes for UR traffic.

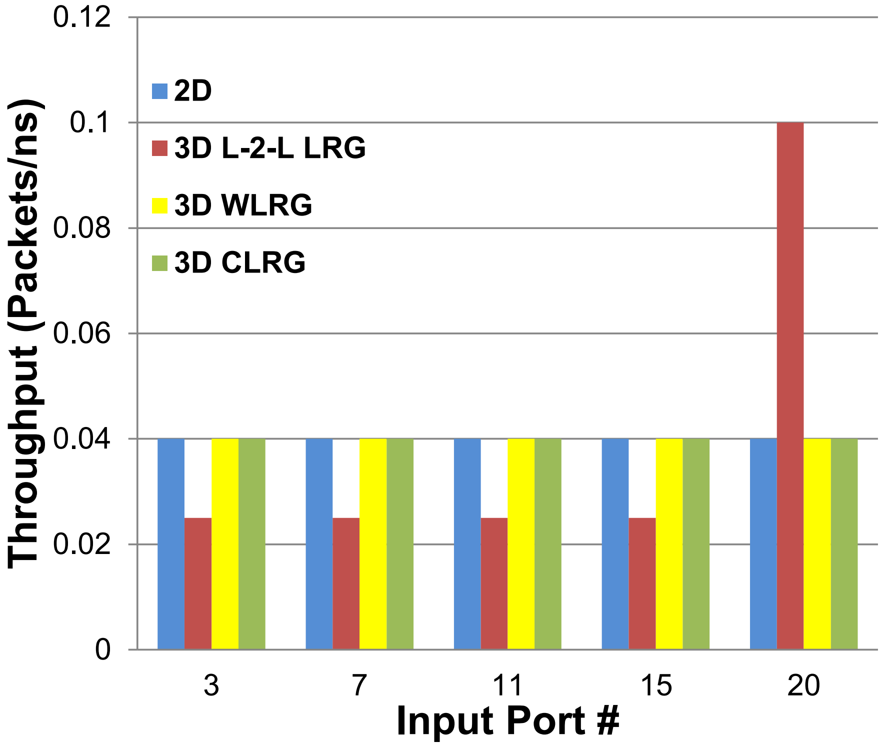


Figure 2.15 Throughput of requesting inputs for baseline's adversarial traffic.

2.6.3 TSV technology parameters

In Section 2.4.1 effects of design parameters like the number of stacked layers, and number of L2L channels were discussed. Another important consideration that can affect both the implementation cost and the performance is the TSV technology being used. The TSV technology used in this switch is a high-end 0.8um pitch TSV. The area and switch delay for a less advanced TSV technology will be more, because of the bigger pitch and higher wire parasitics. However, TSV pitch has constantly been going down as 3D integration evolves. The advancement of technology will lead the 3D switch to become effective even at low radices.

Figure 2.16 shows the area increase with TSV pitch. This area increase is attributed to the fact that a TSV punches through the silicon layer, rendering that area useless. The area increase also factors in the routing to and from the TSVs. The increase in area and capacitive loading for large pitched TSVs in less advanced technologies causes the delay to increase. Even with an additional 25% pitch, Hi-Rise area increases by only 1.67%, and frequency falls by 1.8%. Additionally, Tezzaron [16] uses Tungsten TSVs instead of copper, which has a matching expansion coefficient with silicon, hence keep-out-zone is negligible. Another feature of Hi-Rise topology is that it can use clustered TSVs for the Layer-to-Layer Channels, amortizing the effect of keep-out-zones, for other TSV technologies.

2.6.4 Application results

In this section, we evaluate the proposed Hi-Rise switch for real application traffic. For this, a 64-core system using a single switch as the interconnect fabric is created as discussed under the methodology section. The two systems used for comparison are identical, except that one has a 2D flat switch as the interconnect, while the other uses the Hi-Rise 4-channel 4-layer switch with CLRG built into it.

To evaluate the effect on performance, eight different multi-programmed workloads are simulated. Each workload consists of six applications, with multiple application instances used to construct the workload as shown in the Table 2.6. The applications' allocation is done randomly, and is oblivious of the layer-to-layer dependencies in the switch.

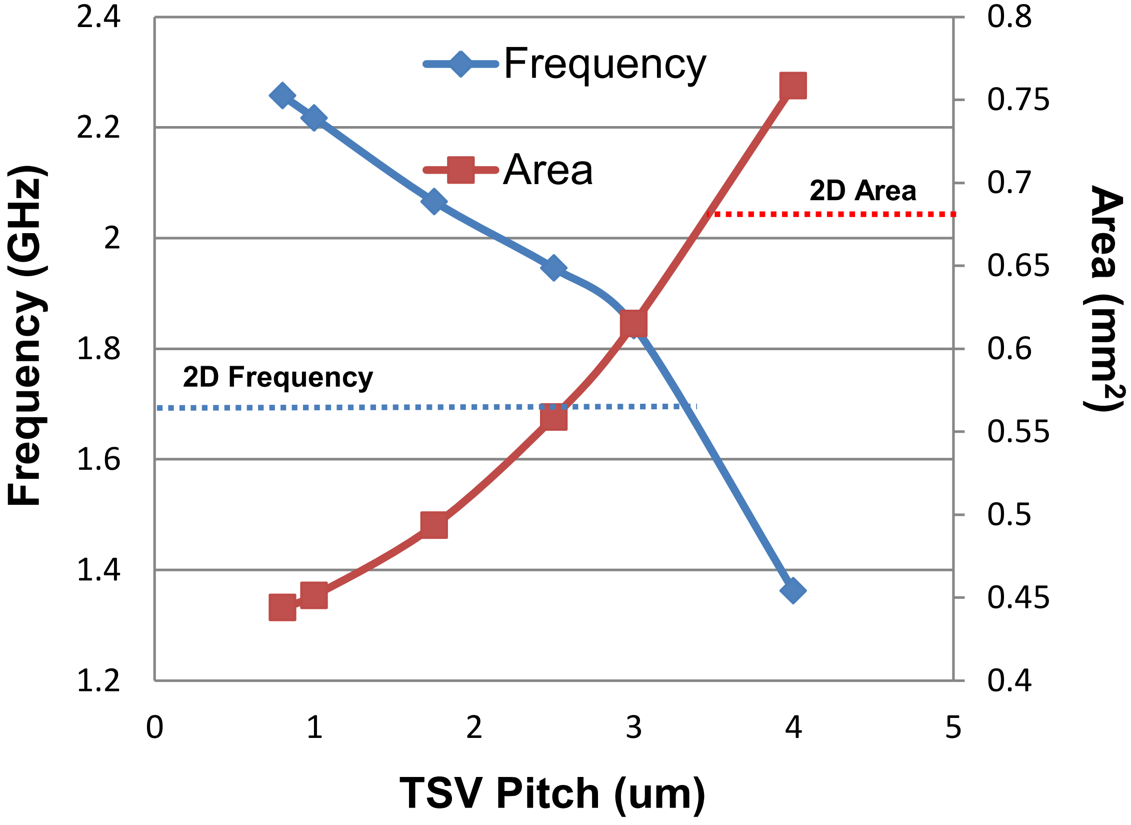


Figure 2.16 Sensitivity of frequency and area to TSV pitch for *Hi-Rise*

Table 2.5 Implementation cost of different switch arbitration variants for 64-Radix

Design	Configuration	Area (*mm*2)	Frequency (*GHz*)	Energy/ transaction (*pJ/trans*)	Throughput (*T bps*)	#TSVs
2D	64×64	0.672	1.69	71	9.24	0
3D L-2-L LRG	[(16×28), 16·(13×1)]×4	0.451	2.24	42	10.97	6144
3D CLRG	[(16×28), 16·(13×1)]×4	0.451	2.2	44	10.65	6144

The numbers in parenthesis in Table 2.6 indicates the number of application instances used to construct the workload. The average Misses-Per-Kilo-Instruction (MPKI) per core is the sum of the L1-MPKI and L2-MPKI, which corresponds to the network load. The last column of Table 2.6 shows the normalized system speedup of the proposed 3D switch over 2D Swizzle-Switch. The 3D switch outperforms the 2D switch by 8% on an average. The 3D switch provides better speedup for workloads with higher cache miss rates. For Mix8, which has the largest MPKI amongst all workloads, the proposed 3D switch shows a 15% performance advantage.

Table 2.6 Benchmark speedup results for proposed switch for a 64-core processor

Mix							**avg.MPKI**	**Speedup**
Mix1	milc(11)	applu (11)	astar (10)	sjeng(11)	tonto(11)	hmmer(10)	**15.0**	**1.02**
Mix2	sjas(11)	gcc (11)	sjbb (11)	gromacs(11)	sjeng(10)	xalan(10)	**21.3**	**1.04**
Mix3	milc(11)	libquantum (10)	astar (11)	barnes(11)	tpcw(11)	povray(10)	**33.3**	**1.06**
Mix4	astar(11)	swim(11)	leslie (10)	omnet(10)	sjas(11)	art(11)	**38.4**	**1.06**
Mix5	mcf(11)	ocean(10)	gromacs (10)	lbm(11)	deal(11)	sap(11)	**52.2**	**1.08**
Mix6	mcf(10)	namd(11)	Hmmer (11)	tpcw(11)	omnet(10)	swim(11)	**58.4**	**1.09**
Mix7	Gems(10)	sjbb(11)	Sjas (11)	mcf(10)	xalan(11)	sap(10)	**66.9**	**1.16**
Mix8	milc(11)	tpcw(10)	Gems (11)	mcf(11)	sjas(11)	soplex(10)	**76**	**1.15**

2.6.5 Discussion

2D Swizzle-Switch [12] has been compared to other topologies like mesh and Swizzle-Switch enhanced flattened butterfly. We chose the 2D Swizzle-Switch for comparison, as its power is 33% better than mesh and 28% better than flattened butterfly. Hi-Rise further improves over the 2D Swizzle-Switch power by about 38%, giving us about 58% power savings over flattened butterfly. The system speedup of Hi-Rise over flattened butterfly is approximately 13%.

In this section, we also briefly discuss composing Hi-Rise switches to form larger topologies with 1000 cores (kilo-core). Future kilo-core systems cannot use existing low-radix networks due to scalability issues. To get around this, prior designs have proposed high-radix topologies with concentration [4][5]. This helps reduce the number of routers in the network in addition to reducing the average hop count. Hi-Rise, or other true 3D switch designs, can also be used to make NoC topologies for 3D chips like the one shown in Figure 2.17. The topology is a 2D mesh of 3D switches. This allows routing algorithms to be XY dimensionally ordered, while the 3D switch can provide the adaptable Z dimension routing, leading to optimal utilization of the L2LC. Layer-aware routing algorithms that minimize the traversal of traffic in the vertical direction will also help alleviate the L2LC bottleneck problems within the switch.

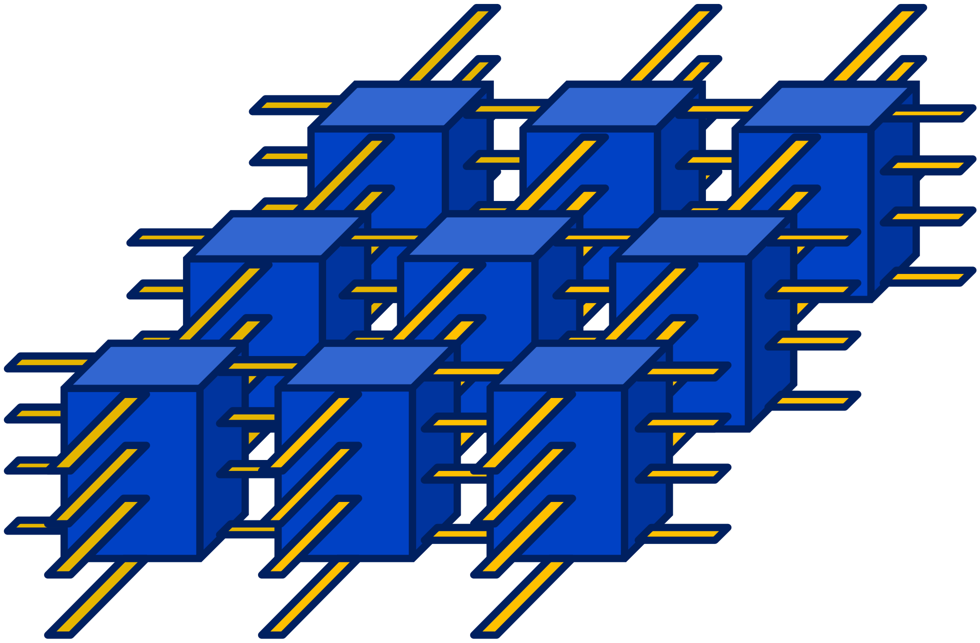


Figure 2.17 A 2D mesh NoC topology composed of 3D *Hi-Rise* switches for 3D chips

2.7 Related work

Prior works have shown that high-radix switches can be used to construct low latency networks [4][5][20][21][22][23][69]. Kim et al. [22] proposed several optimizations to improve the scalability of switches. The optimizations include breaking down the arbitration into multiple local and global stages, and hierarchical crossbars with intermediate buffering. Our proposed Hi-Rise is a 3D high-radix switch composed of two switches that does not have intermediate buffering and has deterministic datapath connections, allowing it to either arbitrate or transmit data in a single-cycle.

As 3D systems become more main stream, 3D integration becomes extremely important in interconnect fabrics. However, most of the 3D switches proposed in recent years have been low radix switches [7][8][9][10]. Xu et al. [10] explored 4 × 4 × 4 and 4 × 4 × 5 3D switch to reduce implementation cost. Kim et al. [9] proposed a low-radix 3D switch which was customized for dimension ordered routing in mesh topologies.

Lewis et al. [24] proposes a folded 3D crossbar and 3D Multistage Interconnect Networks (MINs). It optimizes the folded 3D crossbar by adding switches at each output of different stack layers to cut down the critical path. Unlike [24], in Hi-Rise datapath, only the local layer has a unique bus to each sub-block, while all inter-layer routing is shared among the sub-blocks. This considerably eases the routing and TSV requirements, especially for a wide data-bus. [24] partitions MIN networks effectively to reduce both wiring density and wiring length in 3D. But, MIN networks made up of 2x2 switches, require many stages for high-radix, and have contentions. Hi-Rise datapath has only two heterogeneous stages, each of which is composed of an efficient, contention-free switch.

Several arbitration policies are possible for a switch arbiter [14]. Policies like Longest Queue First (LQF) [25] and Oldest Cell First (OCF) [25] have been used. WLRG is similar to LQF, which is also based on higher priority for higher number of requestors. However, as discussed, the implementation cost of WLRG makes it infeasible. Similarly, [26] uses distance based weights. OCF chooses oldest request based on timestamps, which requires a prohibitively expensive comparison, especially for on-chip high-radix switch with single cycle arbitration. [27] Also uses an age-based arbitration. For high-radix switches, Ping-Pong Arbiters (PPA) [28] have been used which combine small arbiters in a comparison tree. These are also difficult to integrate within the datapath. The CLRG arbitration fits into the datapath itself, reuses the output lines, utilizes embedded self-updating priorities and inhibit logic for arbitration. This reduces the implementation cost significantly, and allows scalability to high-radices with single-cycle arbitration.

Allocators [13][29][30][31][68] utilize multi-stage arbiters to maximize the output bandwidth utilization by matching the requests from the virtual channel buffers to the switch outputs. Allocation schemes like hierarchical switches, also try to compose multi-level arbitration schemes. Allocation policies utilize different combinations of round-robin arbitration [13][30][31]. For example, an iteration of iSLIP [31] updates the round-robin priority at the pre-final stage in a multi-stage arbitration, only if the input wins at the final stage. A single iteration of iSLIP is similar to the baseline L-2-L LRG we discussed before and does not solve the fairness issues. In Backlog Weighted Round-Robin (BWRR), proposed for hierarchical switches [32], a backlog signal is passed from the first stage to the second stage. The second stage does not update its priority if the backlog signal for the winner is high. This technique incurs similar overheads as WLRG. The CLRG arbitration is also a multi-stage arbitration scheme, which trades-off implementation complexity and fairness. Our proposed class based division of primary inputs, allows maintaining a coarse-grained LRG at the inter-layer switch for each primary input.

2.8 Conclusion

The processor industry is moving towards 3D integration and more cores per chip. This is creating the need for interconnects with features to exploit the potential of 3D integration.

This thesis presents Hi-Rise, a fast, high-bandwidth, area-energy efficient, high-radix, 3D switch, with single-cycle built-in arbitration. The proposed Hi-Rise switch adopts a hierarchical architecture with two internal switches per layer and dedicated layer-to-layer channels. The inter-layer switch on each layer makes the proposed solution a true 3D switch which connects inputs and outputs across different silicon layers. The thesis proposes an integrated arbitration scheme to resolve unfairness in a hierarchical 3D switch. The proposed Class-based Least Recently Granted (CLRG) scheme is able to provide fairness comparable to that of a flat 2D switch with Least Recently Granted arbitration.

This is the first work which presents an efficient 3D high-radix switch design. The proposed 3D switch is evaluated for different radices, number of stacked layers and different TSV technology parameters. A 64-radix, 128-bit width, 4-layer Hi-Rise evaluated on a 32nm technology has a throughput of 10.65 Tbps for uniform random traffic, which marks a 15% improvement in throughput over a 2D design, along with a 33% area reduction, 20% latency reduction, and 38% reduction in energy per transaction.

CHAPTER 3

Configurable memory (TCAM/ BCAM / SRAM / logic-in-memory)

3.1 Introduction

A Content Addressable Memory (CAM) compares its search input data with every word stored in the memory, and returns the address location of matching words. A BCAM or binary CAM looks for an exact match, while a TCAM or ternary CAM can have "don't care" bits in the memory, and therefore TCAM words can match multiple search strings.

CAMs are very useful wherever a lookup table is involved. CAMs can perform a parallel search operation across multiple data and consequently boost system performance. This parallel multi-data search makes CAM an indispensable component for high-associativity caches, translation look-aside buffers [33], and register-renaming [34]. Look-up tables are also the main function of IP router tables, as shown in Figure 3.1, and therefore CAMs are the major component of many router chips [35][36].

Despite CAM being an important building block, it tends to use large bit cells. The main reason is that foundries typically focus on density and power of SRAM arrays and only make push rule bit cells for SRAMs. In addition, CAMs require highly specialized bit cells with 10 transistors for a BCAM [36][37], or even 16 transistors for a TCAM [36], as shown in Figure 3.2. Hence, in practice, non-push-rule CAM bit cells are several times larger [37][38][39][40] than dense push rule 6T SRAM [41][42] and this results in large CAM arrays.

The main motivation for the proposed solution is to improve CAM density [43][44]. For this, a new CAM structure is proposed that uses a traditional push rule 6T SRAM bit cell, which results in as much as 4× improvement [45] in array density over conventional CAMs. In this way, chip area and overall capacitance can be reduced, leading to higher energy efficiency for search operations.

In addition to CAM functionality, the configurable memory also provides the ability to perform bit-wise logical operations between two or more data words stored in the memory. By performing the operation within the memory array, a system using the proposed solution will be more energy efficient due to reduced data movement. Performing logical operations in memory also frees up the ALU for more involved calculations, and hence boosts performance [46][47][48][49]. The configurable SRAM with both CAM and logic functions can therefore be used in accelerators in both ASICs and general purpose design.

3.2 Conventional CAM design

A conventional CAM is organized to have its words stored row-wise. The search string is applied in the vertical direction, which is same as the bit-lines, whereas the match lines run horizontally like the word-lines, as shown in Figure 3.3. The match-line sense amplifiers at the end of the match-lines provides the match or mismatch result for each row.

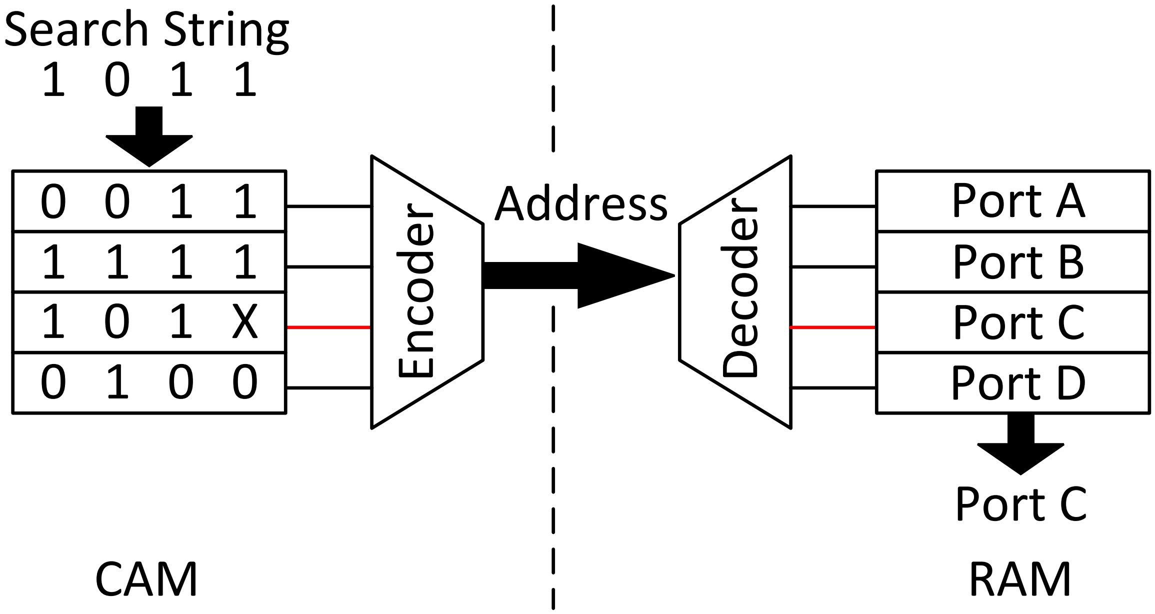


Figure 3.1 CAM – A major component of IP router tables [4]

A word is said to match the search string if each bit of the word matches every bit of the search string. To accomplish the bit-wise comparison, each bit cell has a storage part and a dynamic xnor part. The bit-wise xnors are wire ANDed on the match lines, and the match result is obtained at the output of the sense amplifiers. In many lookup applications multiple matches are required, but if a single address is required the results can also be priority encoded.

As shown in Figure 3.2, a conventional 10-transistor (10T) BCAM bit cell is composed of a 6T SRAM-like storage component, and a 4T XOR component to determine the bit-wise match. A Ternary CAM can store 0, 1, or X, where ‘X’ implies that it matches with both a ‘0’ and a ‘1’ of the search key. As such, it requires double the storage, resulting in a 16T cell. The high transistor count of BCAM /TCAM cells, coupled with the fact that foundries do not typically support “push-rule” CAM cells, results in a CAM array with 2–5× larger area than a corresponding SRAM; this significantly impacts chip area as well as power and performance. Certain TCAM cells are built-up using push-rule [35][50] 8T bit cells. From the layout shown in [50], the 16T TCAM bit cell uses two 8T cells, which is estimated to be 1.35× the size of the proposed TCAM composed of two 6T cells.

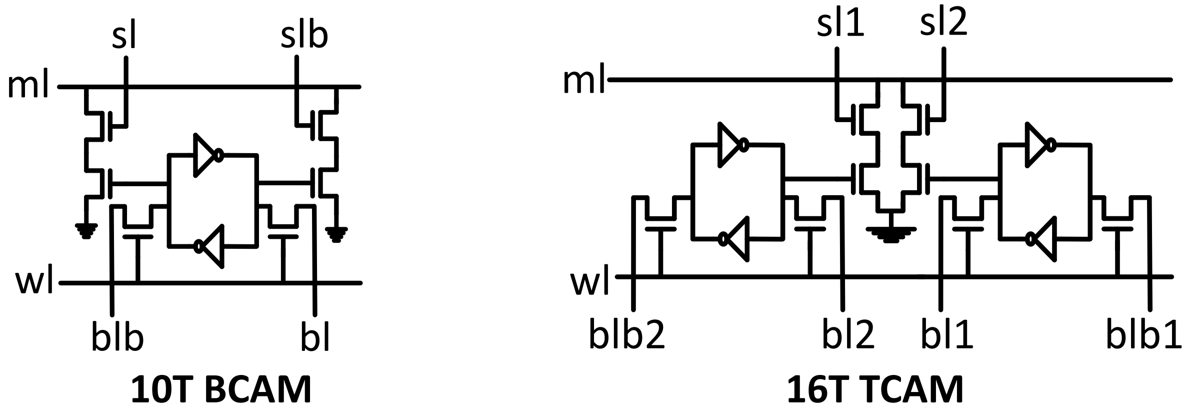


Figure 3.2 Conventional bit cell design for BCAM and TCAM respectively

3.3 Overview of proposed configurable memory circuit

A reconfigurable CAM circuit based on a conventional, push-rule 6T SRAM bit cell that improves array density by as much as 4× is proposed. The approach hinges on storing the words column-wise and using the standard bit-lines to perform a matching operation. Figure 3.4 shows the configurable memory. The word-lines are re-used to apply the search string in the horizontal direction, and the bit-lines are also re-used to read-out the match result.

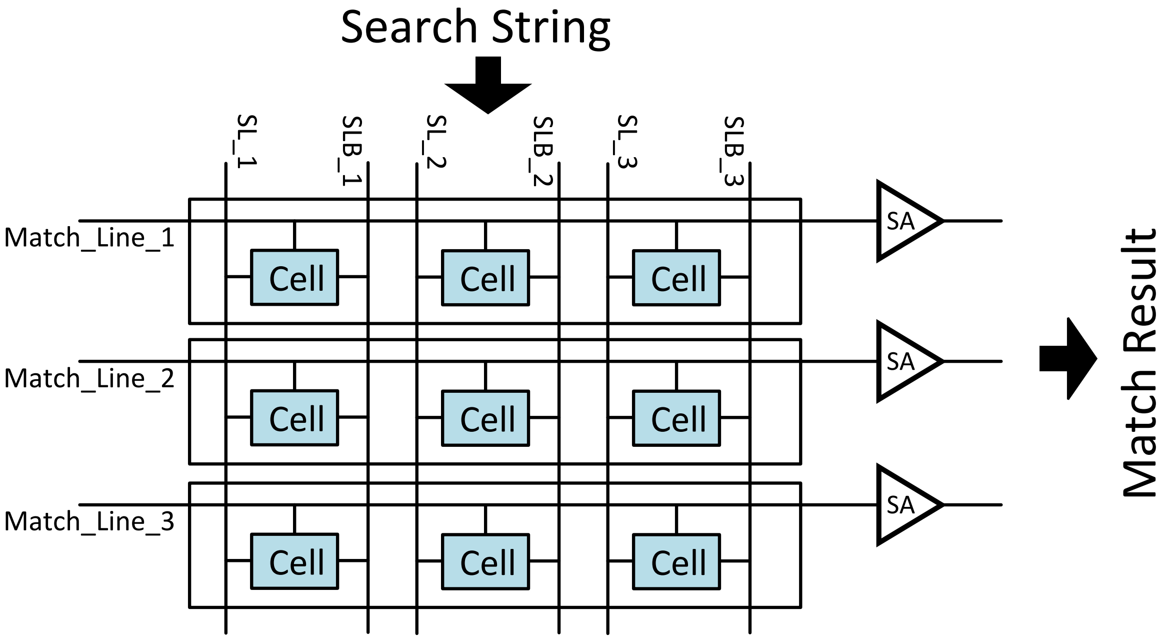


Figure 3.3 Conventional CAM array organization.

A configurability feature allows on-the-fly mode switching among BCAM, TCAM, and SRAM operation. In this way, an SRAM memory can be re-configured to a CAM upon demand to accelerate parallel search-like applications. SRAM mode is still used conventionally with address on word-lines, words stored row-wise, and Data-out on the bit-lines. As a result of using standard push-rule 6T cells, the bit density for the proposed memory array is about 4 times higher than other conventional BCAMs after normalizing for technology.

The configurable memory can also perform bit-wise logical operations: 'AND' and 'NOR' on two or more words stored row-wise within the array. Thus, the configurable memory with CAM functionality and logical function capability can be used to off-load specific computational operations to the memory in order to improve system performance and energy efficiency.

The proposed memory is energy efficient and configurable. Using a 6T 28nm FDSOI SRAM bit cell, the 64×64 (4kb) BCAM achieves 370 MHz at 1V and consumes 0.6fJ/search/bit, as shown in Figure 3.18(a), energy minimum point, while the TCAM achieves the same performance at 0.74fJ/search/bit, as shown in Figure 3.19(a).A logical operation between two 64-bit words achieves 787 MHz at 1V. Some tradeoffs are required to be made for the proposed memory configurability. The proposed memory sacrifices speed in cam mode compared to an SRAM, for area and energy improvement over a conventional CAM.

Also, the re-configurability overhead causes this solution to be 7% larger than a conventional SRAM due to the additional peripherals.

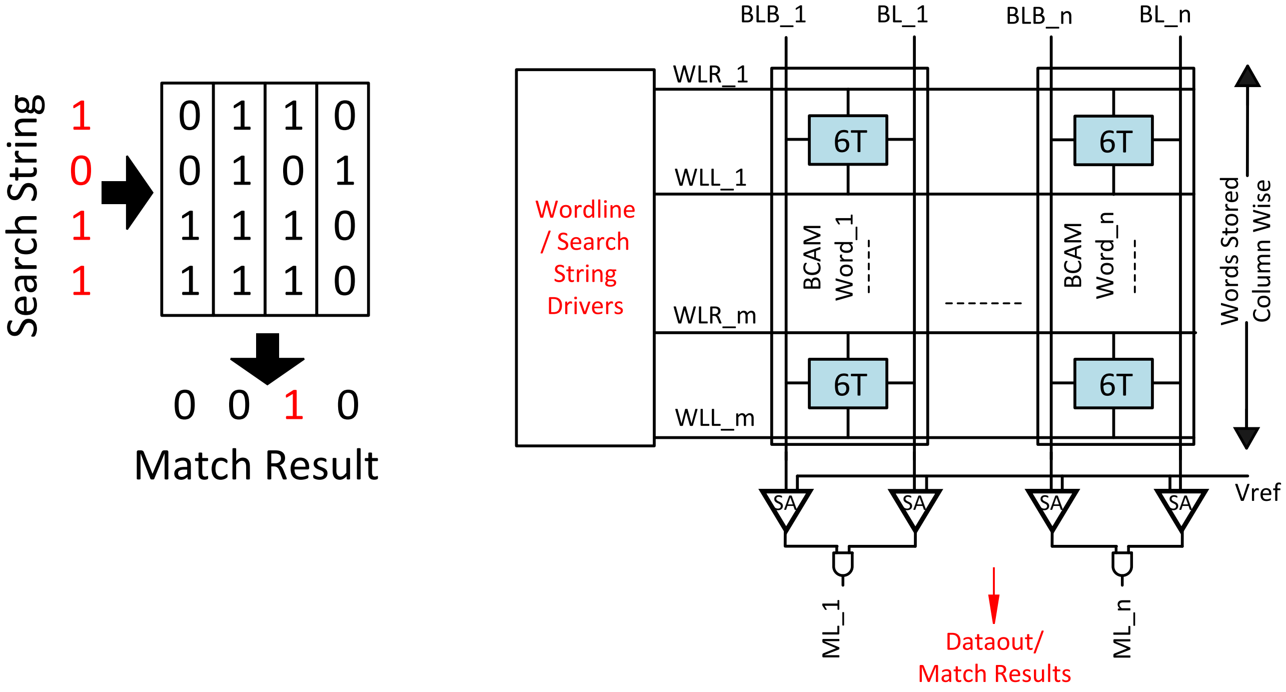


Figure 3.4 Proposed CAM array organization.

3.4 Configurable memory: CAM circuit implementation

This section describes in detail how to obtain CAM operation and logic operations with SRAM bit cells. The section first describes the proposed bit cell and builds up from there. Although the proposed bit cell is 6T push-rule, to obtain the CAM operation the word-line is separated into WLR (Word-Line-Right) and WLL (Word-Line-Left) (Figure 3.4). This creates two independent access transistors but incurs no area penalty since the push-rule layers are kept intact (i.e., only DRC-compliant metallization changes are made).

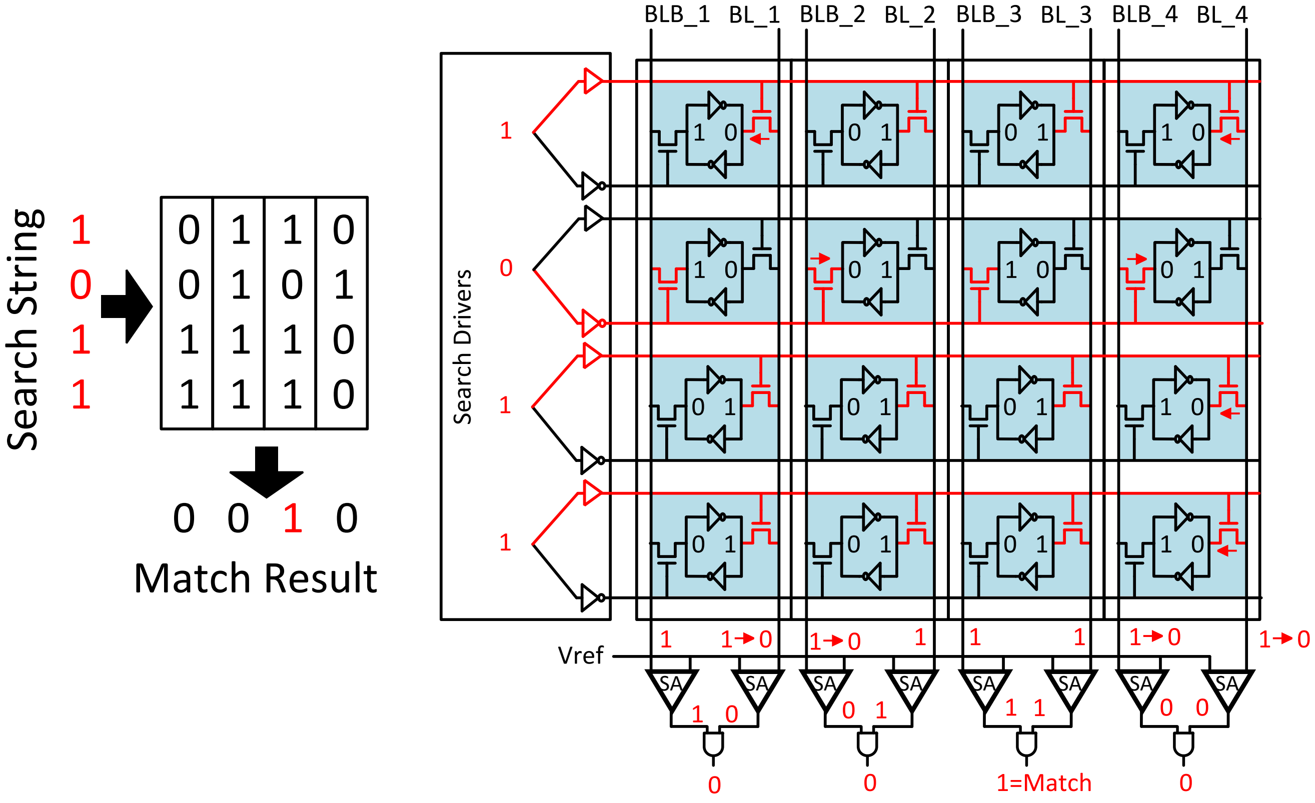


Figure 3.5 BCAM search example. Only column 3 is a match.

The key to performing a parallel search with this bit-cell is to store words column-wise (vertically) while placing the search data on the word-lines rather than the bit-lines as in a conventional BCAM.

3.4.1 BCAM search operation

This subsection explains BCAM search with an example on a simplified 4×4 array. In Figure 3.5 the search-data is applied to WLRs (the bit-line side access transistors) and search-data-bar to WLLs (the bit-line-bar side access transistors).In the match case, both BL and BLB stay at the pre-charged high value. If there is a mismatch, BL, BLB, or both discharge. To detect this, BL and BLB are sensed separately using two single-ended sense amplifiers that are logically ANDed to indicate a match in the column.

The CAM operation will happen in parallel for all the columns of the array. The first column has a 0 in place of the 1 in the search string; therefore, it has a mismatch. As indicated by the red arrow in Figure 3.5, the '0' on the top bit will start pulling the pre-charged bit-line down. This will make the bit-line sense amplifier (SA) to read a '0'. Hence, the AND of the two sense amplifiers outputs a 0 indicating a mismatch, as expected. The timing waveform for BCAM search operation is shown in Figure 3.6. The second cycle shows BL getting discharged, and hence OUT senses 0, indicating a mismatch. In the match case both OUT and OUTB stay high.

The second column in Figure 3.5 has a 1 in place of the 0 in the search string. The 0 on the second bit will start pulling the bit-line-bar down. The ANDing is a 0, indicating a mismatch.

Notice that the proposed memory always indicates the mismatch for a 1 in the search string, on the bit-line sense amplifier, whereas it indicates a mismatch for a 0 on the bit-line-bar sense amplifier. The third column is a match, as all bits are the same in the search string and the stored word. As seen in Figure 3.5, all the access transistors that are enabled have a 1 on both source and drain. Therefore, both bit-line and bit-line-bar stay high and the output at the AND gate is a 1, implying a match.

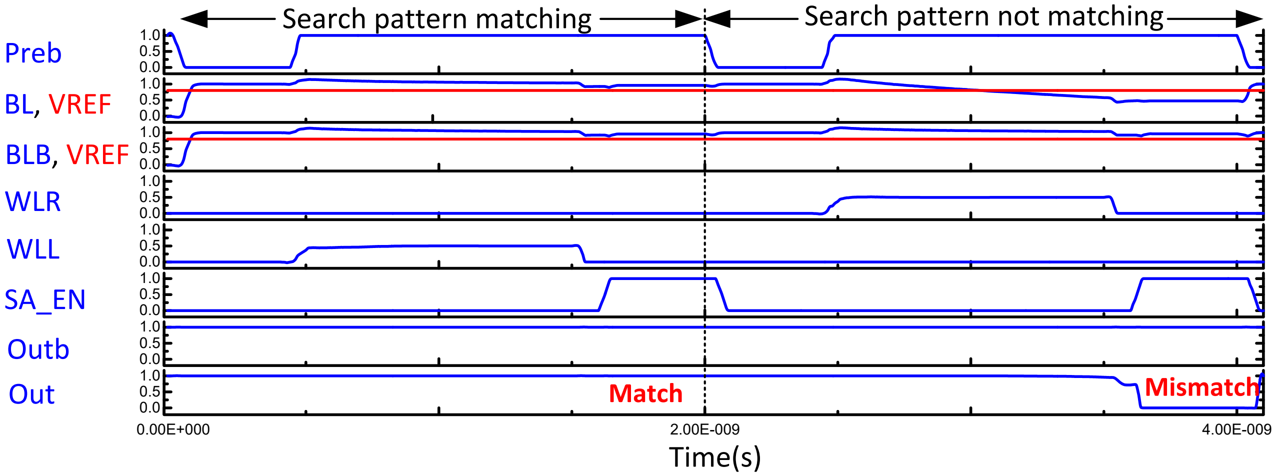


Figure 3.6 Timing waveforms for BCAM search.

The array thus performs a similar operation as a conventional BCAM. The bit-wise XNOR of the data is performed at the access transistors and they are then wire-ANDed at the bit-line sense amplifiers.

The next section describes the unconventional two sense amplifiers per column which is actually designed as a single, reconfigurable amplifier.

3.4.2 Reconfigurable sense amplifier design

The cross-couple of a conventional voltage differential sense amplifier is split into two parallel cross-couples, as shown in Figure 3.7.During the CAM mode, it is required to individually sense both bit-line and bit-line-bar. The upper cross-couple compares bit-line-bar against a reference voltage, vref, while the lower one compares bit-line against vref. During the SRAM mode, the faster differential mode is used between the bit-line and the bit-line-bar. In SRAM mode, both the cross-couples are tied together in parallel, effectively leading to the same strength differential sense amplifier that had been split. Hence, the two sense amplifiers per column obtained for the CAM operation are designed using the same area as that of a standard amplifier for SRAM. Figure 3.8 shows the SPICE simulation waveform for the reconfigurable sense amplifier in the single-ended mode. In the figure, BLB falls below vref; therefore OUTB senses a 0 when ‘SA_EN’ is asserted. This reference voltage vref, used for single-ended sensing mode, is brought in as an additional supply for this chip.

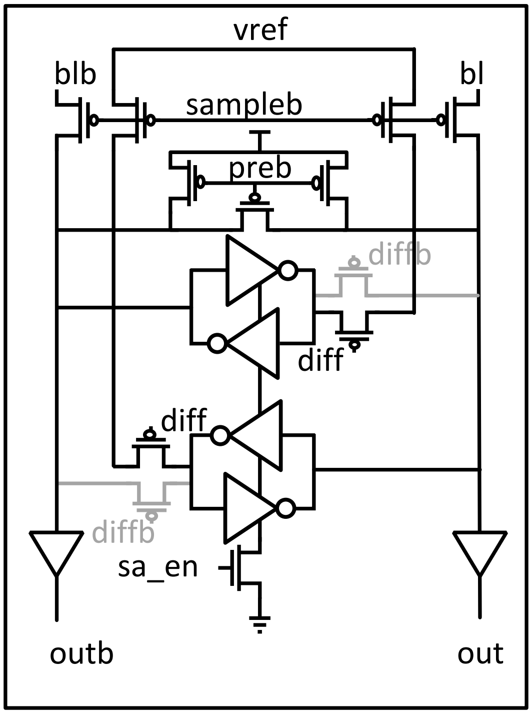


Figure 3.7 Reconfigurable sense amplifier: 2-Single-ended amplifiers in CAM and logic modes

3.4.3 BCAM write operation

One way to write the CAM is to use the SRAM mode and write the transpose of the required CAM data row-wise. But this implies doing a bulk write of cam data, which might be acceptable for applications where the look-up table has static data while the search string changes. However, for a general CAM based lookup it is required to update specific data elements. To write data column-wise into the CAM, as required for parallel search, a two-cycle write scheme is proposed for BCAM mode. A column-decoder is added to select the column to be written.

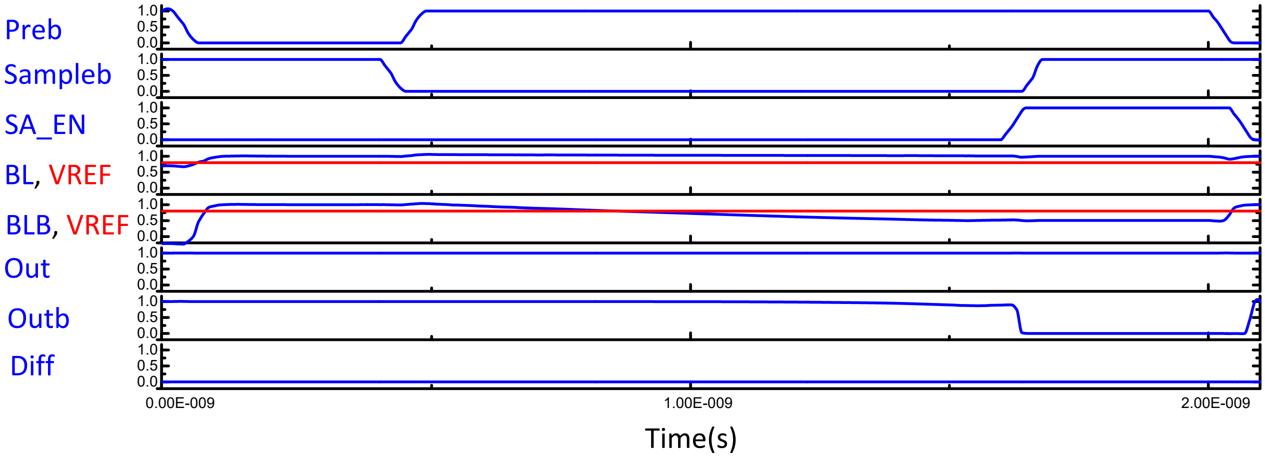
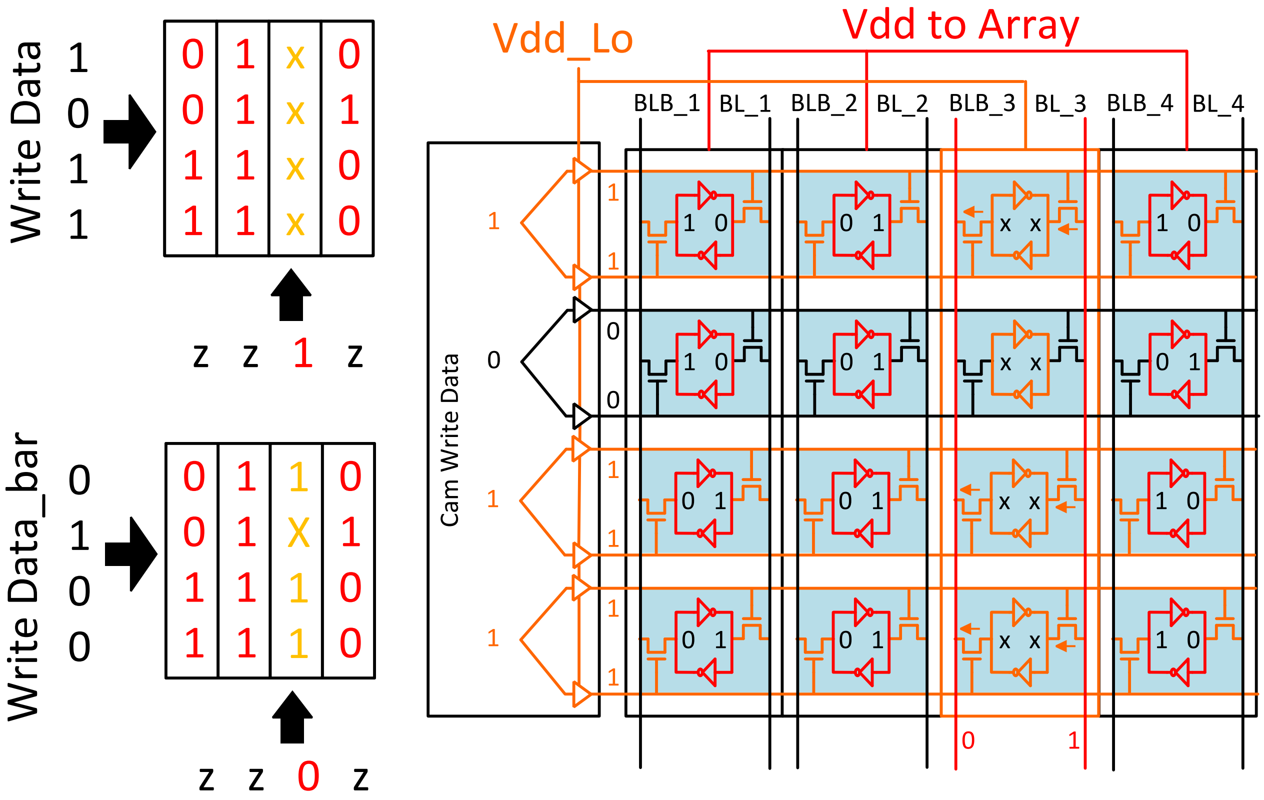


Figure 3.8 Spice waveforms for reconfigurable sense amplifier in single-ended mode

To write column-wise, the data is applied to the word-lines instead of the conventional bit-lines, as shown in Figure 3.9. Column 3, marked in orange, is the column-under-write. The column-wise write takes 2 cycles, wherein all the‘1’s are written in cycle-1 and all the‘0’s are written in cycle 2.In cycle 1, only the word-lines for those bit positions are enabled where a ‘1’ has to be written. The word-lines are under-driven, and additionally, the cross-couple voltage of the column-under-write is also lowered to Vdd_Lo as seen in Figure 3.9 by the orange cross-couples in column 3. This allows the third column to be written even with low word-line voltages. The other columns are protected from data corruption, by keeping their cross-couple voltage high. Also the bit-line & bit-line-bar are driven strongly only for the column-under-write. Thus, the first cycle only writes all the 1s in the column-under-write.



Write Data: 1 0 1 1

0	1	x	0
0	1	x	1
1	1	x	0
1	1	x	0

z z 1 z

Write Data_bar: 0 1 0 0

0	1	1	0
0	1	X	1
1	1	1	0
1	1	1	0

z z 0 z

Figure 3.9 BCAM column-wise write. In this example column 3 is being written

Similarly, in the 2nd cycle the 0s are written. For this, data-bar is applied on the word-lines. When writing a 0, the 1s already written in the column should not be corrupted. Therefore, the Vdd_Lo should not go below the retention voltage. The constraint for Vdd_Lo is thus two sided – It should be less than the Vdd_disturb and more than the retention voltage. The timing waveform for BCAM write operation is shown in Figure 3.10. The first cycle shows Bit_x (bit at row index 'x'in the column-under-write) being written with a '1' followed by Bit_y in the same column being written with a '0' in the second cycle. While Bit_y is being written, Bit_x holds its data at Vdd_Lo.

In addition, if data is written in 'bulk', the extra write cycle can be avoided by first writing zero into the entire array in one cycle and then only writing the '1' bits in the data to each of the columns.

3.4.4 BCAM search robustness

The robustness and the probability of data corruption in a BCAM is discussed in this section. Unlike the SRAM, multiple word-lines are enabled in the array for the CAM operation.

The bit cell encircled in Figure 3.11matches the search string but the data in the column as a whole does not, and hence the bit-line will discharge. As a result, this matching bit cell has a write-like condition, a pseudo write, where the BL is falling, and the access transistor is on. But this disturb is not very strong because of two reasons. Firstly, the search disturb is only single-ended as just one access transistor is on for the cell. Secondly, the falling bit-line is well above 0.

However, the bit-line voltage is data-dependent. A column with multiple mismatches with '1's on the search string can have BL closer to 0. Thus, the data in the bit cell might still flip under sufficient process variation.

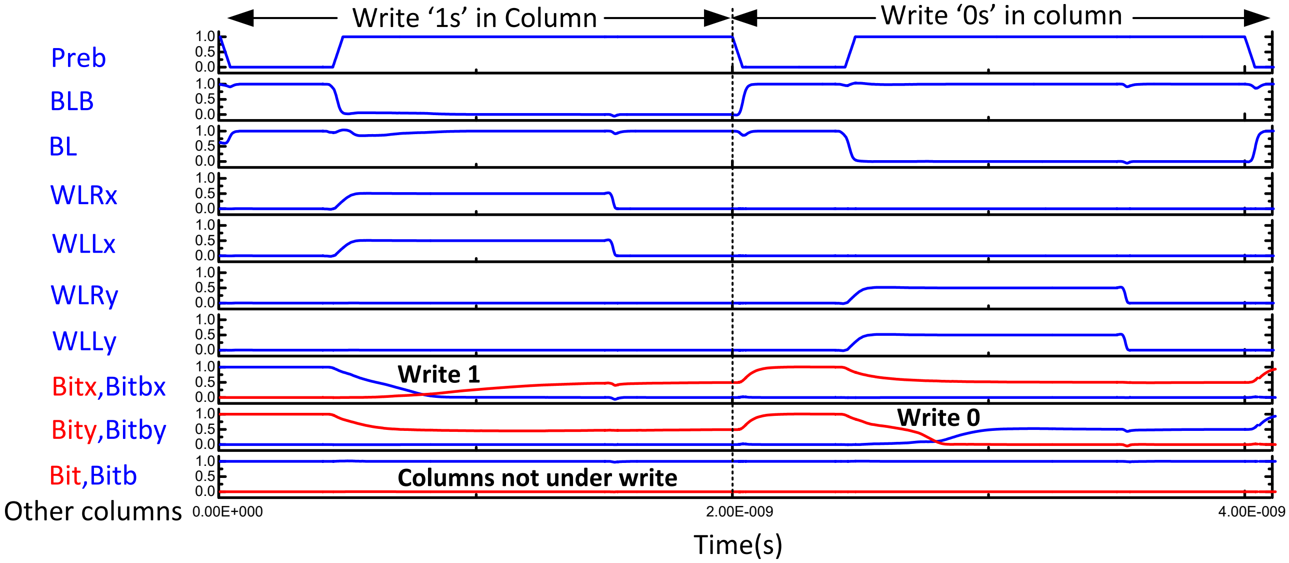


Figure 3.10 Timing waveforms for 2-cycle BCAM write

To solve this search disturb, it is required to weaken the access transistors, and make the storage cross-couple stronger. But for this, the layout cannot be changed, as the SRAM mode and the push-rule cell should not be affected. Therefore, a different voltage on the word-line drivers is used as an assist technique. The word-lines are under-driven, while the power lines supplying the cross-couple in the columns are kept high at Vdd.

The word-line under drive and cell boosting prevents data corruption during the search and write operations. By using Vdd_Lo for both write and search assist, only one additional supply voltage is needed for the configurable memory. Figure 3.12 shows a Monte-Carlo analysis of the write and search disturbs. The disturb margin in write is smaller than search, as both access transistors are on during write, but with the assist techniques used we still get a mean noise margin of 263mV at 27°C.

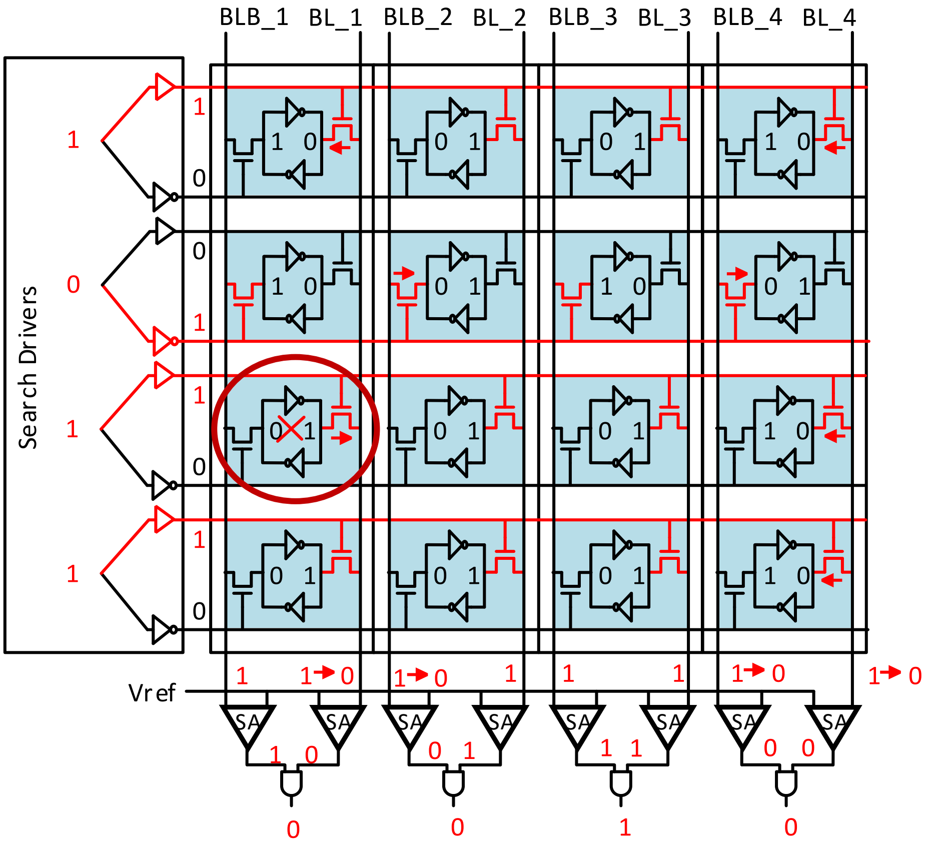


Figure 3.11 BCAM Search disturb: Pseudo write condition on encircled bit cell

3.4.5 TCAM mode operation

TCAM mode will be covered in brief in this section, as it is very similar to BCAM mode in its operation. As the TCAM needs 0, 1 & don't care to be represented, it needs two bits per cell. Consequently two columns have to be used for each word, as shown in Figure 3.13, and hence the capacity is half. To represent X, '01' is used, whereas 0 and 1 are simply 00 and 11respectively.

In TCAM read, the only difference with BCAM mode is the sense amplifiers being observed, as each word spans two columns, as can be seen in Figure 3.13. In this mode two of the four sense amplifier outputs that span the two columns constituting a word are ANDed together. A mask bit ‘X’ will not discharge either sensed bit-line or bit-line-bar as it stores a ‘1’ in both positions. Hence it matches with both 0 and 1 of the search data. In the example in Figure 3.13, the top right bit enclosed in the red box is masked; hence the second word matches ‘1011’. By virtue of the mask bit, the second word would also have matched the search string ‘0011’.

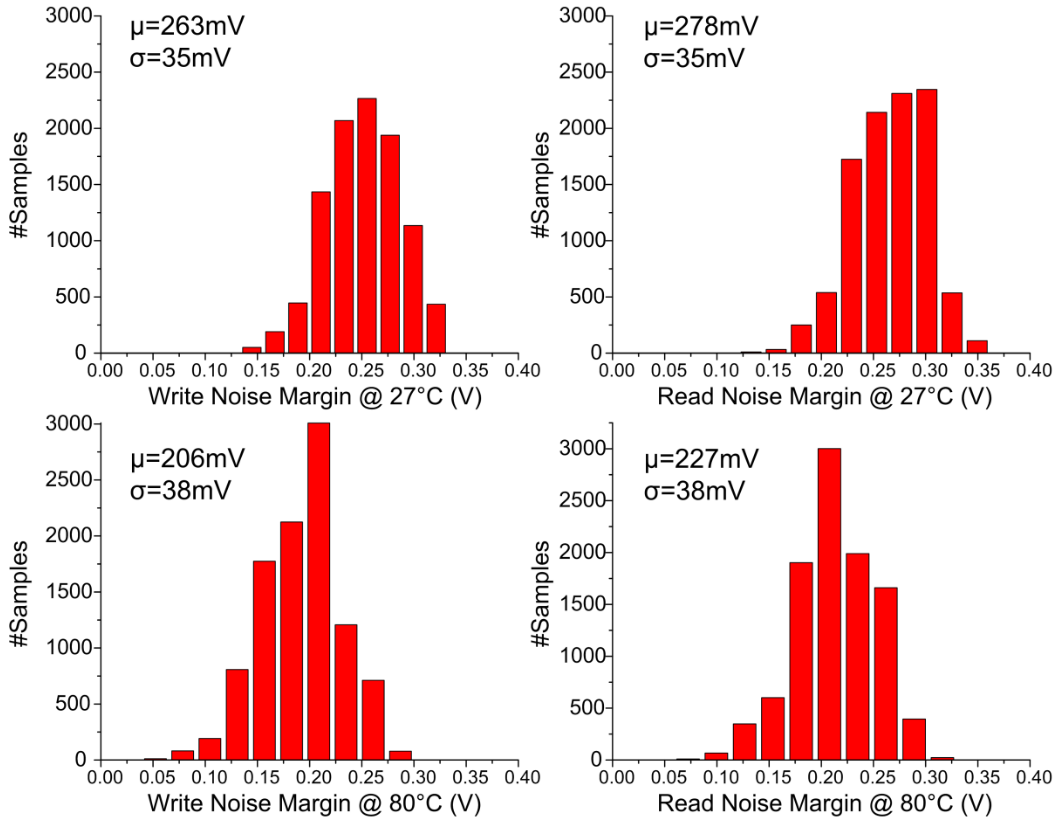


Figure 3.12 Monte-carlo simulations for write and search (read) robustness in CAM modes

TCAM write is similar to BCAM but takes three cycles. The first two cycles are similar to BCAM, as first ‘11’is written and then ‘00’ is written. The mask bits ‘01’ are then written in the third cycle by only enabling the word-lines of rows which need to be masked. The adjacent cells are written with 01, by applying the appropriate voltage levels at the bit-lines. This has been shown conceptually in Figure 3.13. In Figure 3.14 we show the TCAM write operation's timing waveform. Bit$_x$ and Bit$_{x+1}$ is written with '11', whereas Bit$_y$ and Bit$_{y+1}$ is written with '00'. To write mask in column-word$_i$ in row$_z$, Bit$_z$ is written as '0' and Bit$_{z+1}$ as '1'as shown in the figure. Since write is less common in many CAM applications than search, the additional cycles pose less overhead.

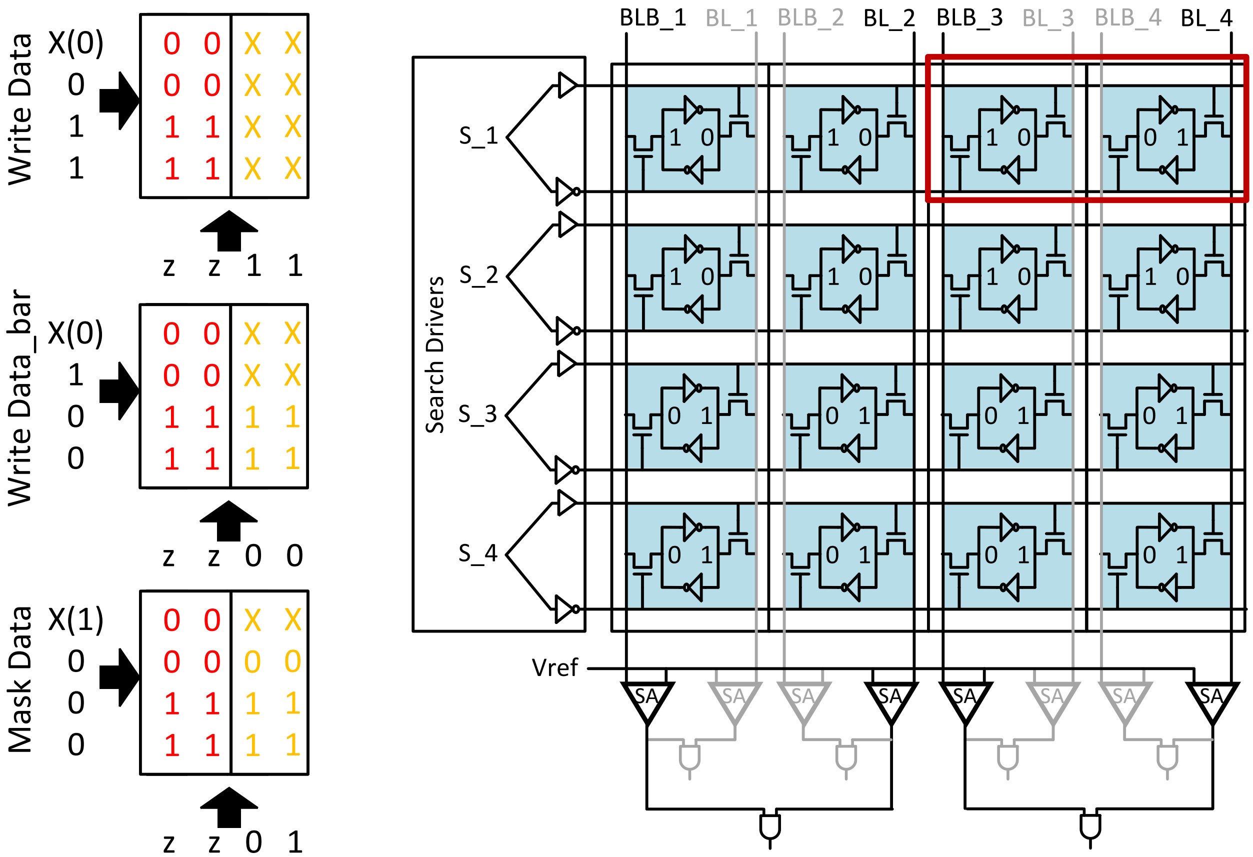


Figure 3.13 TCAM mode organization. Two columns comprise one TCAM word

3.5 SRAM mode and logic in memory

3.5.1 SRAM mode operation

In SRAM mode, the configurable memory works conventionally with both WLR and WLL driven from the address-decoder output. In SRAM mode, reads and writes proceed row-wise using conventional differential signaling and the performance impact from re-configurability is found to be negligible. By reconfiguring the two single-ended sense amplifiers in CAM mode into a single differential sense amplifier in SRAM mode, total reconfiguration area overhead is limited to only 7% for the added column decoder.

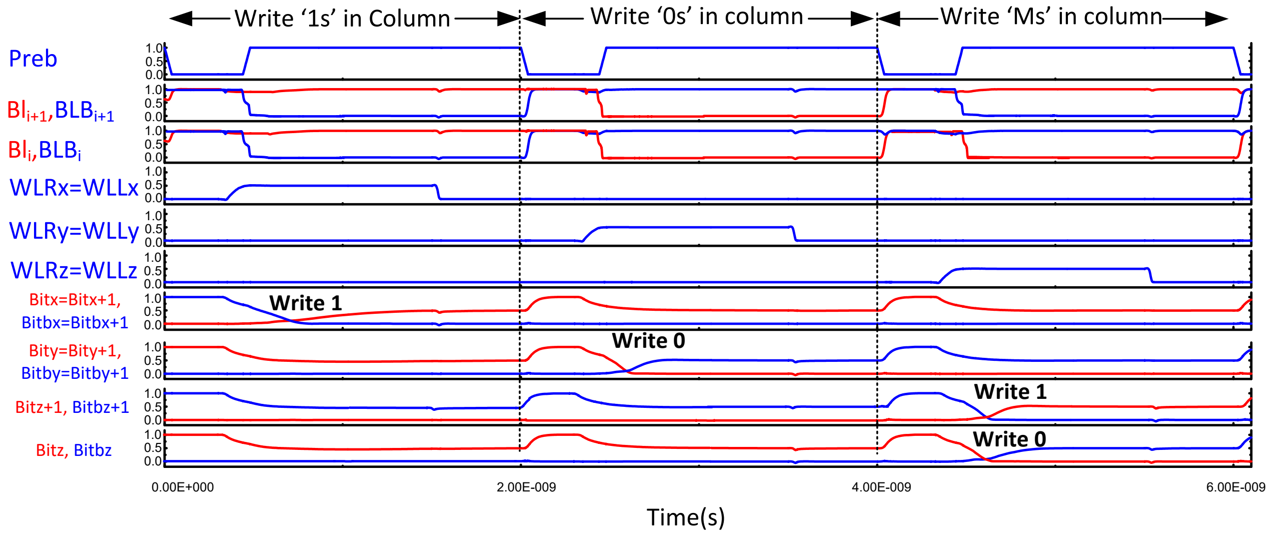


Figure 3.14 Timing waveforms for 3-cycle TCAM write.

3.5.2 Logic operations in memory

The configurable memory can be used to perform certain logical operations between the row-wise stored SRAM words. These logic operations are enabled by reutilizing the circuits used in the CAM modes.

Logic-in-memory here is defined as - the feature of performing logical operations within the memory sub-array itself, without having to read-out or sense the individual words being operated upon. The term logic-in-memory has been used before in other contexts, such as using memory technology other than CMOS to realize logic on a dedicated memory layer [51], or dedicated logic layer in 3D DRAMs [52], or logic in the main memory [47], but not within the sub-array. The main difference lies in not sensing the individual operands to perform the logic.

Figure 3.15 shows an example of a bit-wise 'AND' operation between two rows in the array. To perform the 'AND' operation, the memory is put in the BCAM search mode.

In the BCAM search mode an input bit of the search string can be masked (denoted by 'M' in Figure 3.15) by applying a '0' to both the WLR and the WLL. This feature allows the word-lines of two or more rows to be enabled.

Table 3.1 Logic operations supported in configurable memory

Search String	**Logic in Memory**	**Multi-Word Operation Possible**
11 (Other Rows Masked)	A AND B	Yes (A&B&C…)
00 (Other Rows Masked)	A NOR B	Yes (NOT(A|B|C…))
01 (Other Rows Masked)	A_Bar AND B	No
10 (Other Rows Masked)	A AND B_BAR	No

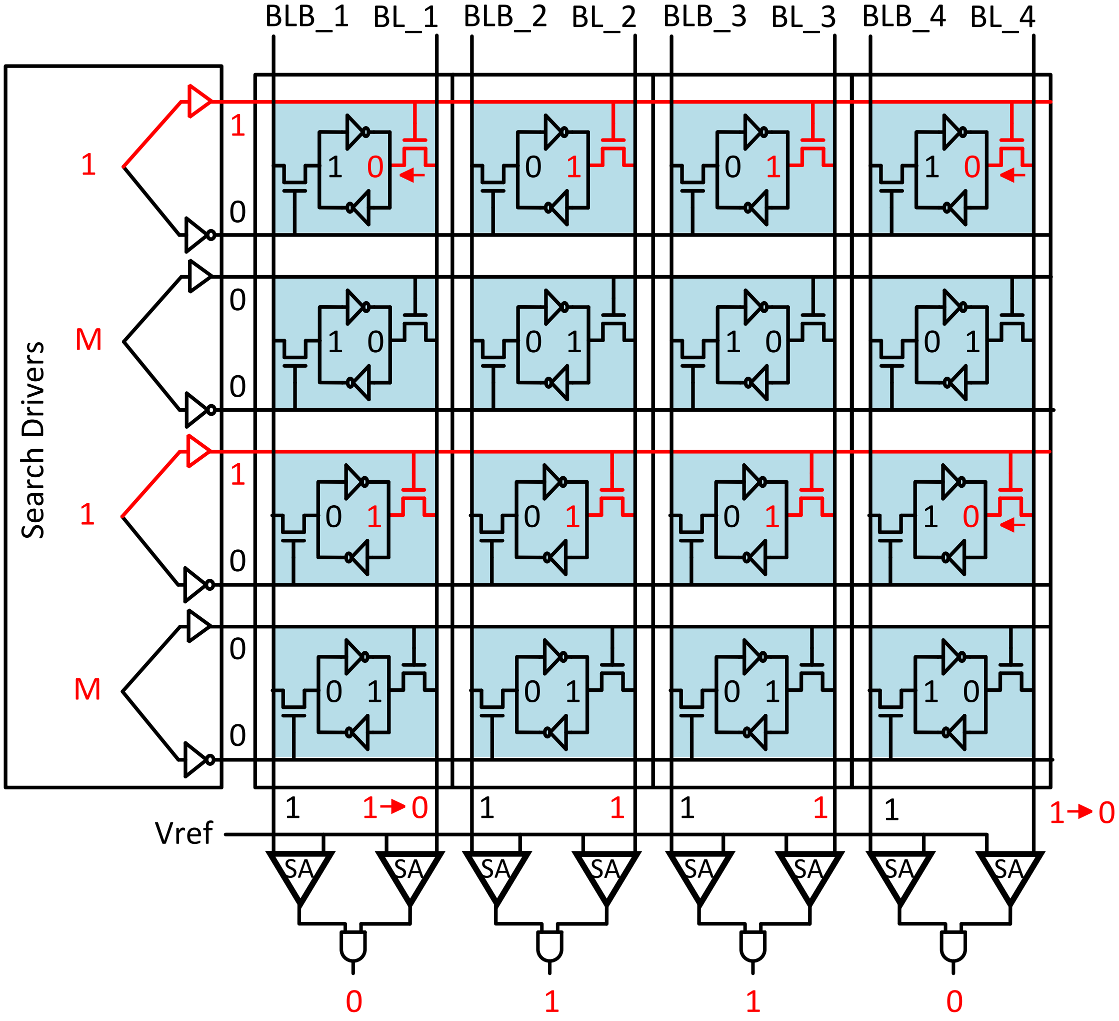


Figure 3.15 Logic operations in memory - 'AND' example

In the example shown in Figure 3.15, the search string (1, M, 1, M) is applied, which only activates WLR for rows 1 and 3. If any bit in row 1 or row 3 is '0', it will pull-down the pre-charged bit-line. As all WLL transistors are disabled, all bit-line-bar lines stay high. Hence, the bit-wise AND of rows 1 and 3 is obtained at the memory output.

More than two words can also be activated by putting more 1s in the search string. The bit-wise ‘AND’ operation can thus be executed for two or more than two words. Table 3.1 shows all the logic operations supported by the proposed configurable memory. Similar to the ‘AND’ operation, a NOR operation can be performed by only activating the WLL access transistor and by applying 0s at the search string. A ‘01’ combination activates WLL for row A and WLR for row B, hence it senses the complement of the data in row A on the bit-line-bar sense amplifiers, and simultaneously senses the data in row B on the bit-line sense amplifiers. These two are then ANDed to produce the result. ‘10’ has the same operation as ‘01’, but changes the location of the rows activated. A ‘01’ like operation allows two rows to be read out simultaneously, as the configurable memory has two single-ended sense amplifiers. Thus, this feature can also be used as a dual read port, where ‘A_bar’ is read on the bit-line-bar sense amplifiers and ‘B’ is read on the bit-line sense amplifiers.

Similar to the BCAM search robustness issue discussed in Section 3.4.4 above, logic operations also activate multiple word lines. The BCAM search activates all the word lines, and hence the probability of data corruption is higher. To prevent data corruption in BCAM mode, Vdd_Lo has to be reduced significantly. During a logic operation on two words, only two bits are fighting in any column. This allows Vdd_Lo to rise significantly; hence the logic mode for two words can run much faster than the BCAM mode. For multi-word logic operations, the Vdd_Lo reduces with the increase in the number of words that are simultaneously operated upon, and consequently the frequency of operation reduces. If an ‘AND’ or ‘NOR’ operation is performed on all the rows in the array, operation approaches the BCAM frequency and Vdd_Lo value.

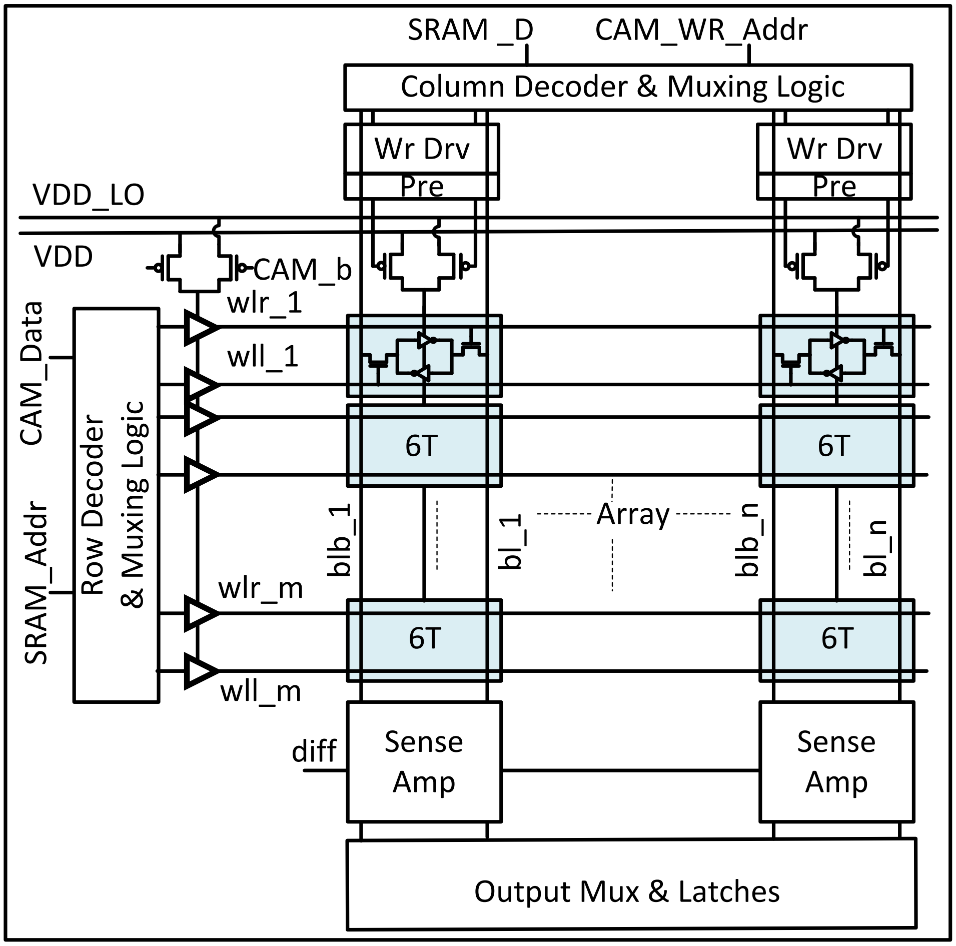
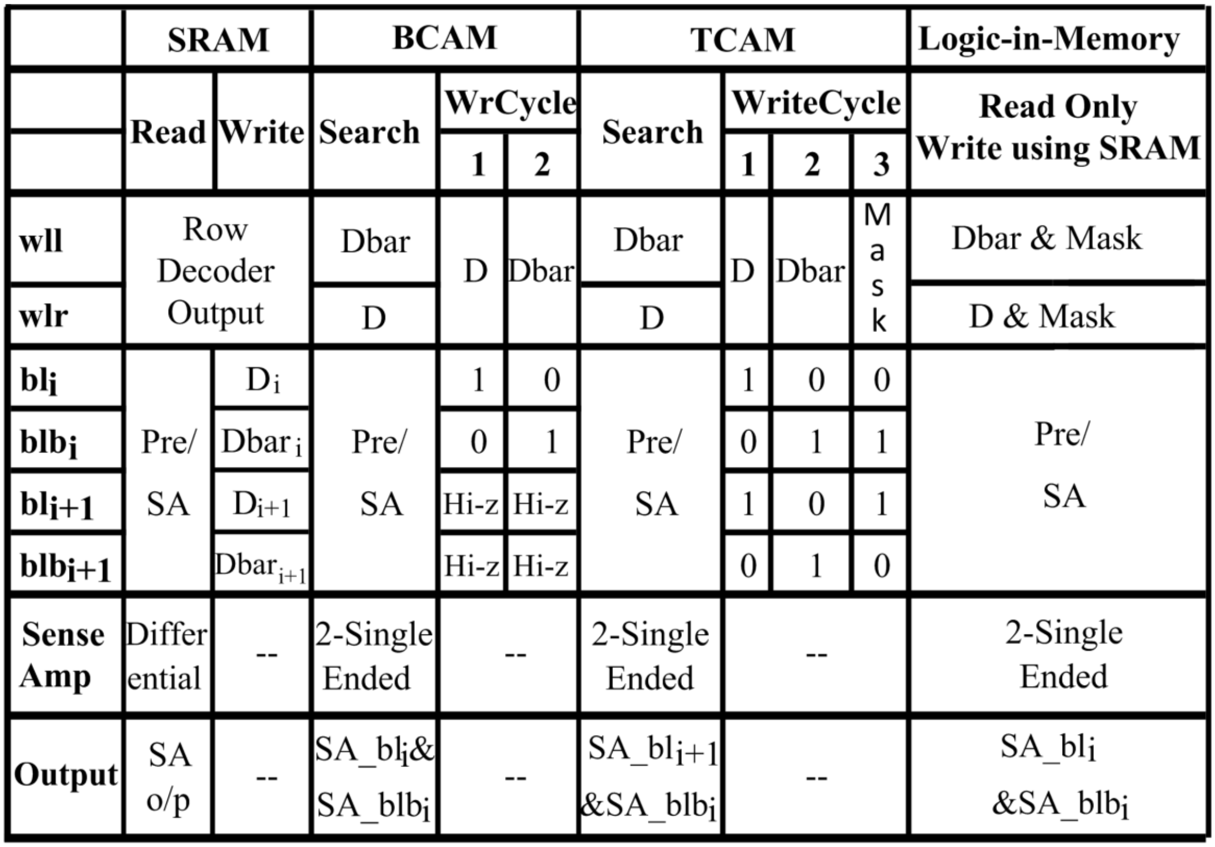


Figure 3.16 Configurable memory organization

Figure 3.16 is the overall block diagram of the reconfigurable memory. Notice the additional column decoder at the top. This ensures that only the column-under-write is supplied by low Vdd. The column decoder output also controls the enable of the write drivers. A common header switch is placed for word-line drivers to switch between Vdd and Vdd_Lo, as shown in Figure 3.16. Also, most 6T SRAMs at advanced technology nodes, need some type of read/write assist techniques. One common technique used is WL under-drive/over-drive. If this were the case, these assist switches are reutilized for CAM WL under-drive. Table 3.2 summarizes the memory driver and sense amplifier configuration during different memory modes.

Table 3.2 Configurable memory – Mode configuration table.



	SRAM		BCAM			TCAM				Logic-in-Memory
	Read	Write	Search	WrCycle		Search	WriteCycle			Read Only Write using SRAM
				1	2		1	2	3	
wll	Row Decoder Output		Dbar	D	Dbar	Dbar	D	Dbar	Mask	Dbar & Mask
wlr			D			D				D & Mask
bl_i	Pre/ SA	D_i	Pre/ SA	1	0	Pre/ SA	1	0	0	Pre/ SA
blb_i		$Dbar_i$		0	1		0	1	1	
bl_{i+1}		D_{i+1}		Hi-z	Hi-z		1	0	1	
blb_{i+1}		$Dbar_{i+1}$		Hi-z	Hi-z		0	1	0	
Sense Amp	Differential	--	2-Single Ended	--		2-Single Ended	--			2-Single Ended
Output	SA o/p	--	SA_bl_i& SA_blb_i	--		SA_bl_{i+1} &SA_blb_i	--			SA_bl_i &SA_blb_i

3.6 Test harness

The configurable memory is validated in a 28nm FDSOI CMOS test chip. An on-chip Built-In Self-Test (BIST) is used to test the different modes. March test was applied to test the bit cells for fault models like stuck-at, transition, coupling and neighborhood pattern sensitive faults. As 6T bit cells are being used even for CAM, the SRAM tests are able to cover most of the fault models.

For testing the CAM search functionality, walk mode search patterns are used. The most critical search pattern is differentiating between match case and a single mismatch case. The walk mode uses a search pattern that negates one bit position every cycle. This pattern is able to test a 1-bit mismatch condition for every bit in every column for at-speed functionality. In addition read disturb faults in CAM modes are tested by creating the worst case column data and corresponding search patterns. The worst disturb scenario is when all bits mismatch except one bit, on the same bit-line. Again we use walk modes to test each bit for worst disturb.

Interleaved write and search operations are used to test the correct at-speed column-wise write for CAM modes. The BIST is able to modify the expected data pattern in a walking mode. In addition, checkerboard patterns are run on the 4kb array with 64-bit words to check for column-wise write and then read back using SRAM mode reads. Arbitrary data can be searched at-speed using an on-chip FIFO buffer.

3.7 Measured results

The configurable memory has been designed in a 28nm FDSOI CMOS process. Figure 3.17 shows the die photo. The dimension of the memory array is 64×64, to make a 4kb array. The BCAM read and write disturbs are sensitive to column length, and were verified on this chip for a contiguous column length of 64 bits. The word length can be extended further by having multiple banks with an AND tree of match results. With an array area of 724um2 for 4kb, the bit density of array is ~5.4Mb/mm2.

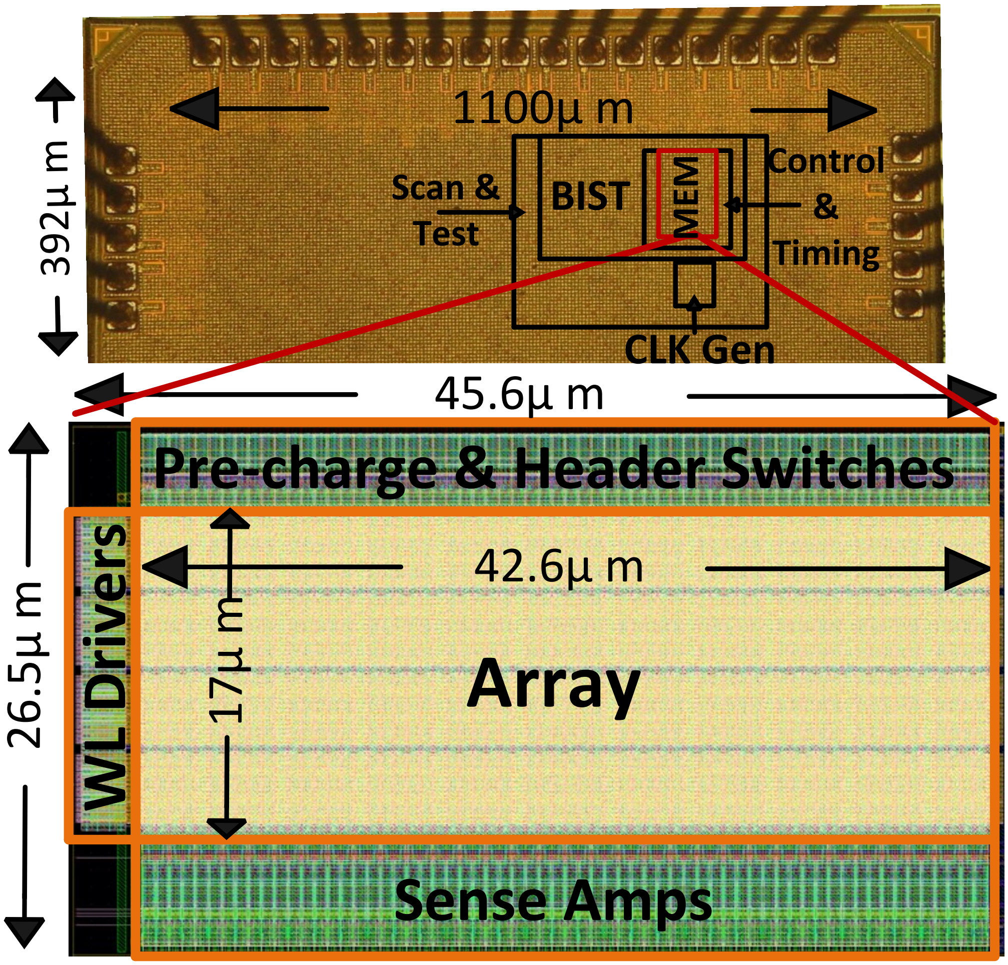


Figure 3.17 Die photo and memory layout

For BCAM, both the measured frequency and energy are a function of Vdd & Vdd_Lo.It is found that Vdd_Lo close to Vdd-divided-by-2 works well. Vdd_Lo is brought in as an additional supply for this chip.On the x-axis of the graph in Figure 3.18(b), Vdd and Vdd_lo are swept, keeping a ratio of 0.5 between them. The black curve shows that the frequency increases with voltage, as expected. At Vdd = 1V, a maximum frequency of 400MHz is achieved. The blue energy curve is a bit more complex, as it also depends on the frequency.

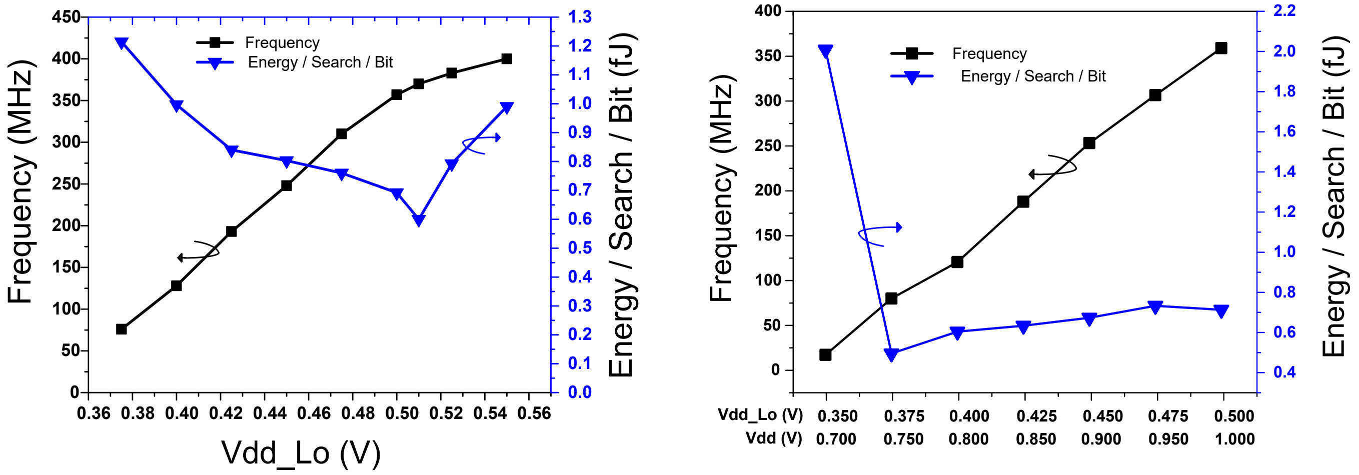


Figure 3.18 (a) Measured frequency and energy in BCAM against Vdd_Lo, with Vdd = 1V. (b) Measured frequency and energy in BCAM against Vdd, with Vdd_Lo = 0.5*Vdd

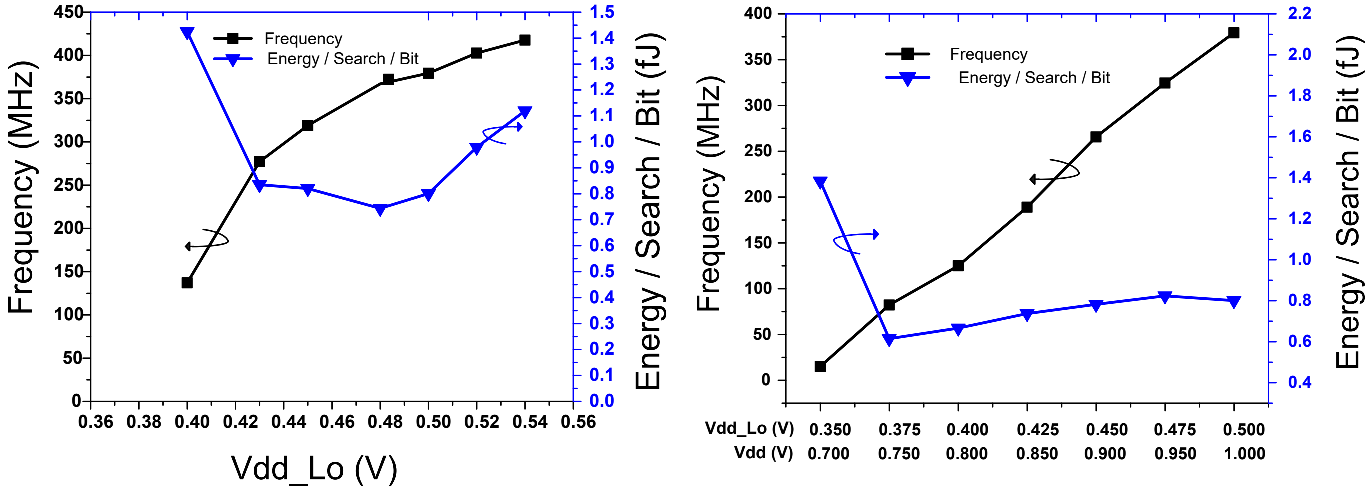
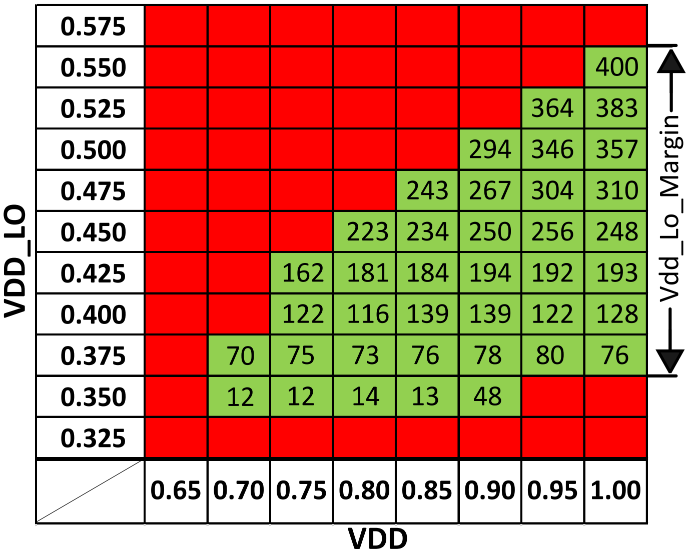


Figure 3.19 (a) Measured frequency and energy in TCAM against Vdd_Lo, with Vdd = 1V. (b) Measured frequency and energy in TCAM against Vdd, with Vdd_Lo = 0.5*Vdd

To gain more insight into this, Vdd is kept fixed to 1V nominal and only Vdd_Lo is swept, as shown in Figure 3.18(a).As can be seen from the figure, the frequency of BCAM operation is a strong function of Vdd_Lo. Also the energy has a sweet spot. It decreases with voltage up to a certain point, before starting to increase again due to the frequency falling, which incurs higher leakage energy. This energy optimum for BCAM at Vdd=1V is measured to be 0.6fJ per search per bit. The minimum energy point is 0.41fJ, with a frequency of 70MHz at Vdd=0.7V and Vdd_Lo=0.375V.

The TCAM frequency and energy have a similar trend as BCAM. In TCAM mode, the maximum frequency is 417MHz and the optimum energy is measured to be 0.74fJ per search per bit, as seen in Figure 3.19(a). The energy consumption per bit in TCAM is higher as the total number of bits is half, but in TCAM only half the sense-amplifiers and output latches are used. The minimum energy point is 0.61fJ, with a frequency of 116MHz at Vdd=0.75V and Vdd_Lo=0.375V, as seen in Figure 3.19(b).



0.575	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
0.550	🟥	🟥	🟥	🟥	🟥	🟥	🟥	400
0.525	🟥	🟥	🟥	🟥	🟥	🟥	364	383
0.500	🟥	🟥	🟥	🟥	🟥	294	346	357
0.475	🟥	🟥	🟥	🟥	243	267	304	310
0.450	🟥	🟥	🟥	223	234	250	256	248
0.425	🟥	🟥	162	181	184	194	192	193
0.400	🟥	🟥	122	116	139	139	122	128
0.375	🟥	70	75	73	76	78	80	76
0.350	🟥	12	12	14	13	48	🟥	🟥
0.325	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00

Figure 3.20 Measured shmoo plot of Vdd_Lo vs VDD for BCAM. Numbers in box are frequency in MHz

In Figure 3.20, a shmoo plot is shown with Vdd on the x-axis and Vdd_Lo on the y-axis. As discussed earlier Vdd_Lo has a two-sided constraint. The red tiles are voltage pairs where the BCAM fails, whereas the numbers in the passing green tiles is the operating frequency. If Vdd_Lo is high, speed is better, but as the access transistor become stronger, the probability of disturb goes up, and hence failures start to be seen in the upper left triangle. The failures below 0.325 Vdd_Lo are due to the sense amplifier read resolution, i.e., the design cannot reliably resolve the single-mismatch case for every column.

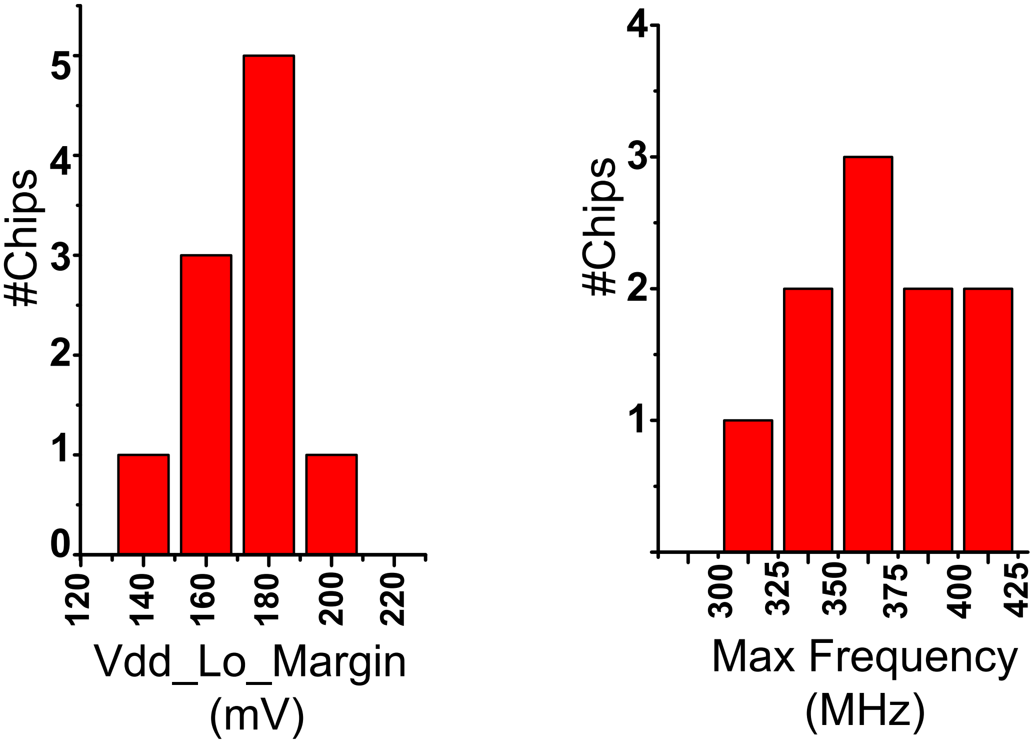


Figure 3.21 Measured Vdd_Lo margin and max frequency across 10 chips.

Figure 3.21 shows the Vdd_Lo operational voltage margin distribution across multiple chips. Vdd_Lo_margin is the voltage range of Vdd_Lo over which the CAM is functional. At nominal Vdd, the Vdd_Lo_margin across 10 chips has a mean of 180mV as shown in the bar graph on the left. Thus, a reasonable margin for CAM operations is available. The mean of the max frequency in BCAM mode across chips is about 365MHz as shown on the right.

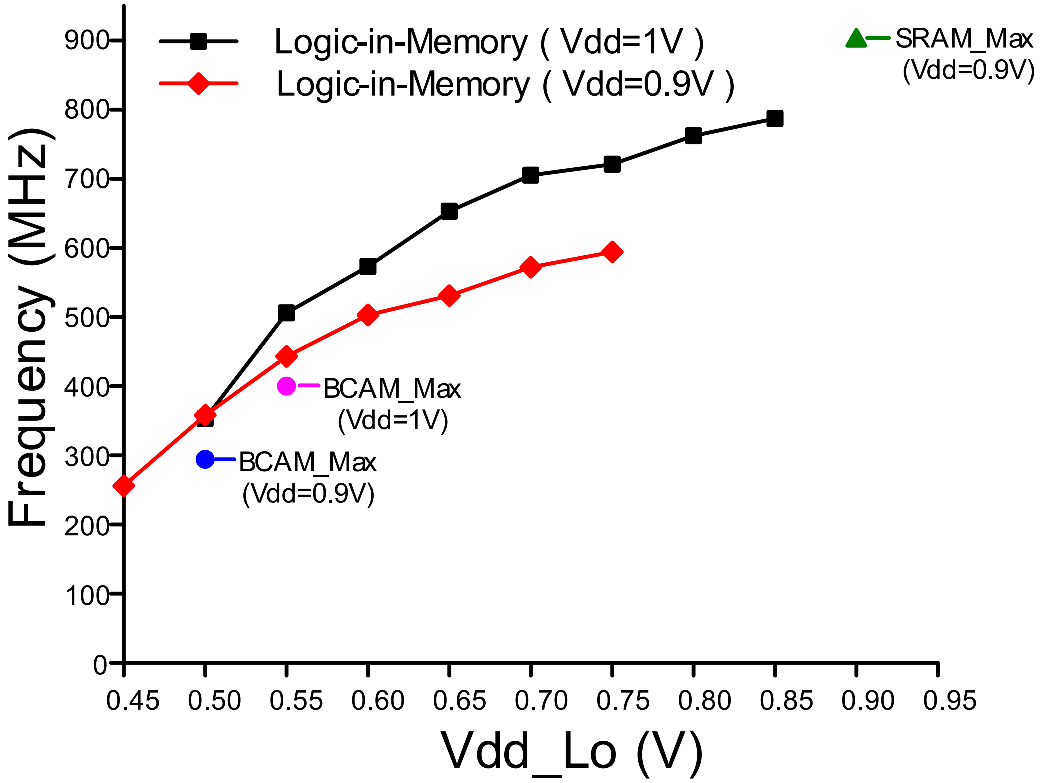


Figure 3.22 Measured frequency for logic operation between two words in memory against Vdd_Lo, with fixed Vdd at room temperature.

The max frequency in SRAM mode is about 900MHz at 0.9V, as it is not affected by Vdd_Lo because the word-lines are driven to nominal voltage in SRAM mode. Figure 3.22 shows the operational frequency for logic operation between two words stored row-wise in the memory. The logic in memory mode is similar to BCAM search operation and hence its frequency is also a function of both Vdd & Vdd_Lo. On the x-axis of the graph in Figure 3.22, Vdd_Lo is swept, keeping Vdd fixed. As explained in Section 3.5.2 above, the search disturb is less pronounced in the logic mode than in the BCAM mode where all rows are activated. For two-word logic mode at Vdd=1V,the Vdd_Lo can be increased to 0.85V, allowing it to achieve a maximum frequency of 787MHz as compared to the BCAM's 400MHz. Compared to the 900MHz for SRAM mode at Vdd=Vdd_Lo=0.9V, the logic mode achieves a maximum frequency of 594MHz at Vdd=0.9V and Vdd_Lo=0.75V. The frequency loss in logic mode compared to the SRAM mode is because of, first, lower word-line voltage and, second, slower sense amplifier as logic mode uses single-ended sensing mode.

Table 3.3 Comparison with previous BCAM works

	This Work	**Agarwal [33]**	**Do [37]**	**Wang [38]**	**Yang [39]**
Technology	**28nm FDSOI**	32nm	65nm	0.13μm	0.18μm
Transistor/Cell	**6T**	11T	10T	9T+Read	13T,14T
Area/Cell[μm^2](F^2)	**0.152 (194)**	n.a.	3.3 (780)*	20 (1200)	30 (926)*
Energy/Search /bit [fJ]	**0.6 (1V)** **0.41 (0.75V)**	1.07 (1V) 0.3 (0.5V)	0.77 (1.2V)	1.87 (1V)	2.82 (1.8V)
Frequency [MHz]	**370**	n.a.	500	250	210
Array Size	**64x64**	(64x64) x4Arrays	128x128	128x32	128x 34x4Block
Match-line Tecnique	**2-Single Ended SA**	Wide AND	NOR	Differential	NAND-NOR
Memory Modes	**BCAM / TCAM / SRAM / Logic**	BCAM	BCAM	BCAM	TCAM/ BCAM

*** From die-photo**

In Table 3.3, our design is compared against other more conventional BCAMs, while in Table 3.4, we compare against other conventional TCAMs. All the conventional BCAMs have a higher transistor count in their bit cells. If the area normalized for technology in F^2(F being feature size) is compared, the gain is by more than 4×. Even the push-rule TCAM bit-cell is 1.35× larger than the proposed TCAM bit cell. The energy efficiency achieved is good at 0.41 fJ / search / bit at 0.75V for BCAM. Also, configurability is possible between different operating modes.

Table 3.4 Comparison with previous TCAM works.

	This Work	**Arsovski [53]**	**Nii [35]**	**Hayashi [50]**	**Huang [54]**
Technology	**28nm FDSOI**	32nm SOI	28nm	65nm	65nm
Transistor/Cell	**12T**	16T	16T	16T	14T
Area/Cell[μm^2](F^2)	**0.304 (388)**	n.a.	0.625 (797)*	1.69 (400)**	7.05 (1669)
Energy/Search /bit [fJ]	**0.74 (1V) 0.61 (0.75V)**	0.58	n.a.	1.98	0.34-0.16 (based on % don't care)
Frequency [MHz]	**370**	1000	400	250	400
Array Size	**32x64 (2columns/word)**	128x128(max) variable	(4k x 80) x64 x4	(2k x 72) x32 x4	256x144
Match-line Tecnique	**2-Single Ended SA**	Early Predict Late Correct	Differential+ Valid bit	Differential Low voltage	Butterfly match-line
Memory Modes	**BCAM / TCAM / SRAM / Logic**	TCAM	TCAM	TCAM	TCAM

* Bit cell area calculated from density, assuming array efficiency of 40%. Reference [35] cites [50] as its previous work. The array efficiency for [50] is 43%. From details in [35] and [50], we conservatively estimate [35] to have an array efficiency of ~ 40%.

** Scaling trend of push-rule SRAM according to ISSCC trends [55] – 124f^2 at 65nm but at 28nm it is 162f^2. Also, from the layout figure in [50] the bit cell uses two 6T cells plus the additional 4 transistors. From the layout figure shown, this bit cell is estimated to be 1.35x the size of the proposed TCAM bit cell (two 6T cells).

3.8 Conclusion

A configurable memory with CAM functionality using standard push-rule SRAM 6T bit cells is presented. This memory can be used as an area-energy efficient CAM in search-based applications. It also has lower instantaneous power because of low voltage word-line drive. The memory can also be used to perform certain logic operations between two or more rows. This can be used to off-load computations to the memory, improving system performance.

The proposed configurable memory with logic-in-memory has an energy efficiency of 0.6fJ/search/bit at 1V with an array bit density of ~5.4Mb/mm2which is a 4× improvement in array density over conventional BCAMs. This is achieved with only 7% area overhead for configurability over a conventional SRAM. The logic-in-memory operations between two 64-bit words in the configurable memory, achieves 787 MHz at 1V.

CHAPTER 4

Sequence dependent PUF

4.1 Introduction

Electronic devices have become more ubiquitous than ever before, from banking infrastructure to critical communication links. This creates enhanced security risks of setting up trustworthy communication links and data privacy. Tampering at hardware level, can defeat the higher, software abstraction level, authentication methods. This has led to the development of hardware level security features, like tamper detection, true random number generator to generate secure keys and chip ID for authentication.

4.1.1 PUF concept

Physically Unclonable Functions (PUFs) have become a popular hardware technique for generating chip ID and authenticating. [72] [73] Physically Unclonable Function is like a device fingerprint. It is unique to each chip due to the manufacturing variations. PUF circuits amplify these variations to reliably get the same output (response) from a given chip for a particular input (challenge), and a different response for the same challenge from another chip.

The strength of a PUF is measured by how many challenge-response pairs the PUF supports, and how hard is it to model the PUF based on the observed challenge-response subset. PUF can be based on delay variations, or bi-stable element resolving to one state, or measuring other physical attributes on-chip.

Existing PUFs are linear, implying the response depends on the current challenge only [62] [63]. This makes it easier to model and learn. The motivation for the proposed PUF design is to make it sequence dependent, which will make the response depend not only on the current challenge, but a sequence of challenges, while simultaneously making the challenge response space larger.

4.2 Proposed PUF design

Conventional SRAM PUFs are usually based on start-up value on power up. The start-up value is decided by the relative strength of the two inverters in the cross-couple. The challenge response space of conventional SRAM PUF is equal to the number of bits in the memory, and it is easier to machine-learn. Also, the bit-error rate is high for power-on PUF.

The proposed PUF is an SRAM based design, which is sequence dependent, independent of power-on, and has a much larger challenge response space. The basic concept behind the proposed PUF is to connect any two words in the SRAM by simultaneously asserting their word-lines. Based on the manufacturing strengths of the two cross-couples of the selected words in each column, the stronger bit cell is able to over-write the weaker bit cell as shown in Figure 4.1.

In Figure 4.1, the first two rows are connected together. Any column which has the bit cells in opposite states (as shown in the ellipse in Figure 4.1), will fight over the bit lines, and one cell will over write the value of the other. The value they resolve to depend on the relative strength of all the 12 transistors of the two bit-cells in the column. More transistors involved in resolving the bi-stable, implies harder modeling. Additionally, the challenge response space is also increased, as we can choose any two rows from the array. For N-rows, we have ~N2 choices of pair-wise row selection. The Vbit voltage is controlled in relation to Vwl voltage to allow bit overwrites to happen.

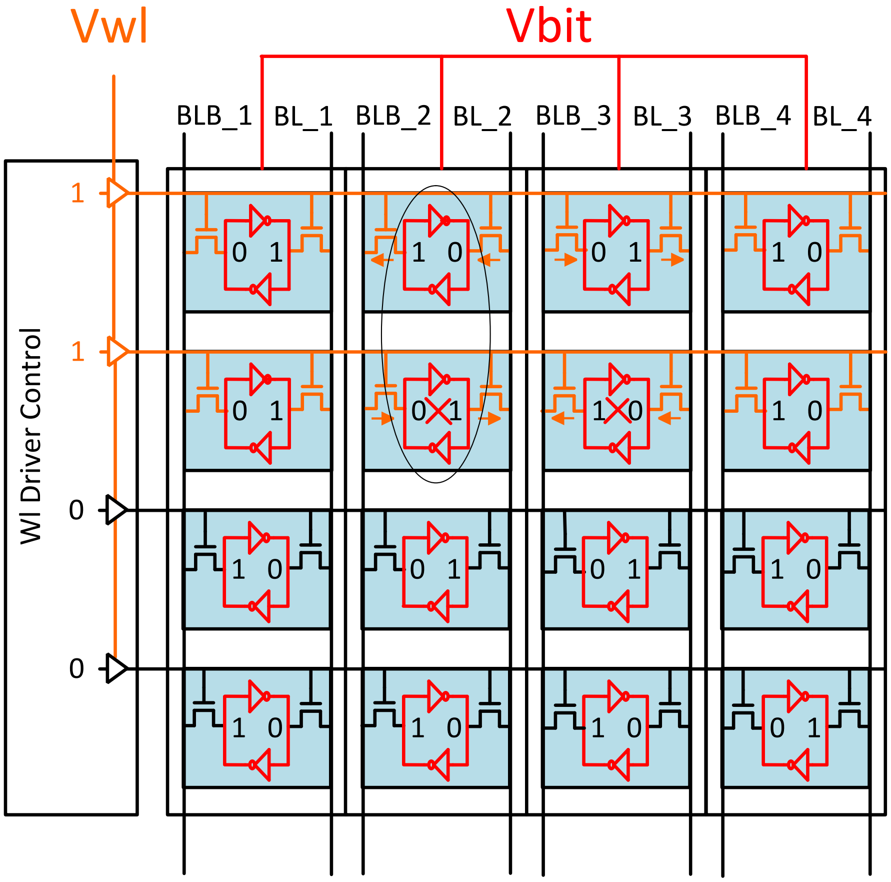


Figure 4.1 Sequence based PUF with two word-lines enabled

Figure 4.2 shows an example of two sequences, where the order of the sequence {(1,2); (2,3); (3,4)} versus {(4,3); (3,2);(2,1)} changes the final state being read from the memory. The notation (a,b) implies that row ‘a’ and row ‘b’ have their word-line asserted simultaneously. In Figure 4.2, the memory is initialized to the same starting value for both sequences. The bits in red are stronger, and overwrite the bits in black. This example is simplified, as two strong bits never fight. In the actual circuit, the bits fight based on their relative strengths.

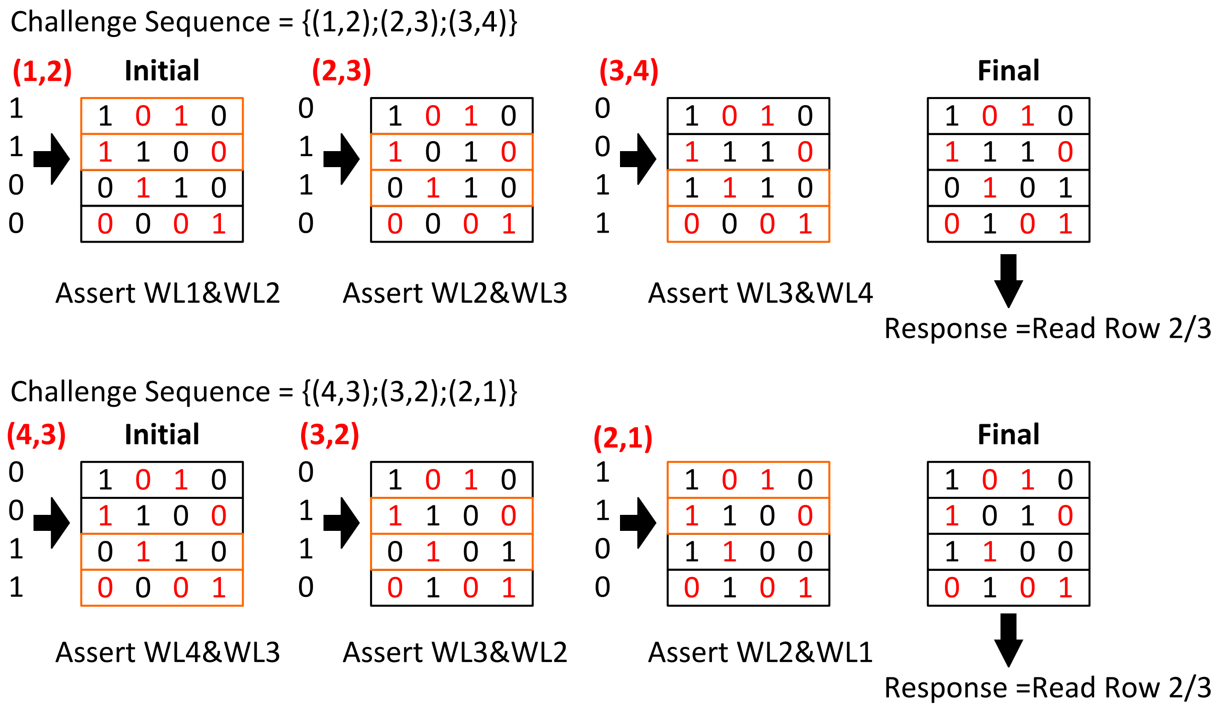


Figure 4.2 Example of sequence dependence of proposed PUF

Even in this simplified example using globally strong bits, at the end of the sequence of three pair-wise word-line assertions, the rows 2 and 3 end up in different states, based on the order in which the rows were connected.

In addition to challenge response space, the other major parameters of a PUF design are the hamming distance (number of bits different) between PUFs and the bit-error rate (BER). When we assert two word-lines together in the proposed PUF, systematic mismatch could lead to reduced hamming distance. This can be caused by one word-line asserting before another, or the slew rates on the word-lines being different, either of which will bias the bit flipping towards one row. Also transition coupling on the bit-lines when word-lines assert, can lead to increase in BER. To avoid the two issues, we extend the pre-charge phase of the SRAM beyond the word-line assertion.

Figure 4.3 shows a column of the array with two bit-cells. The 'preb' signal is typically also used to equalize the two bit-lines in a conventional SRAM. Here the 'preb' signal is split from the 'eqb' signal, to avoid any bias on BL, BLB due to the pre-charge circuit variation. Figure 4.4 shows the waveform. The dashed lines for 'preb' and 'eqb' denote the tunable range of these two signals. We can assert 'eqb' longer than 'preb', hence providing a DC like condition for the two bit cells to fight, without having any row or column based bias.

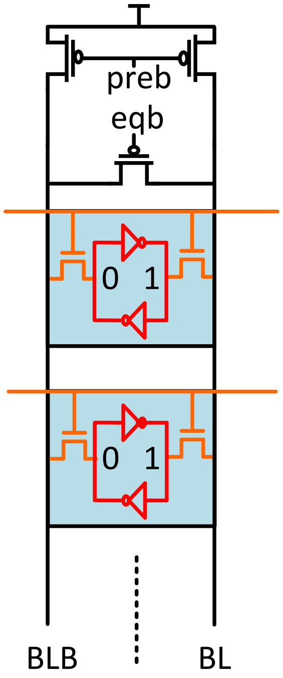


Figure 4.3 Split equalizer signal in SRAM array column

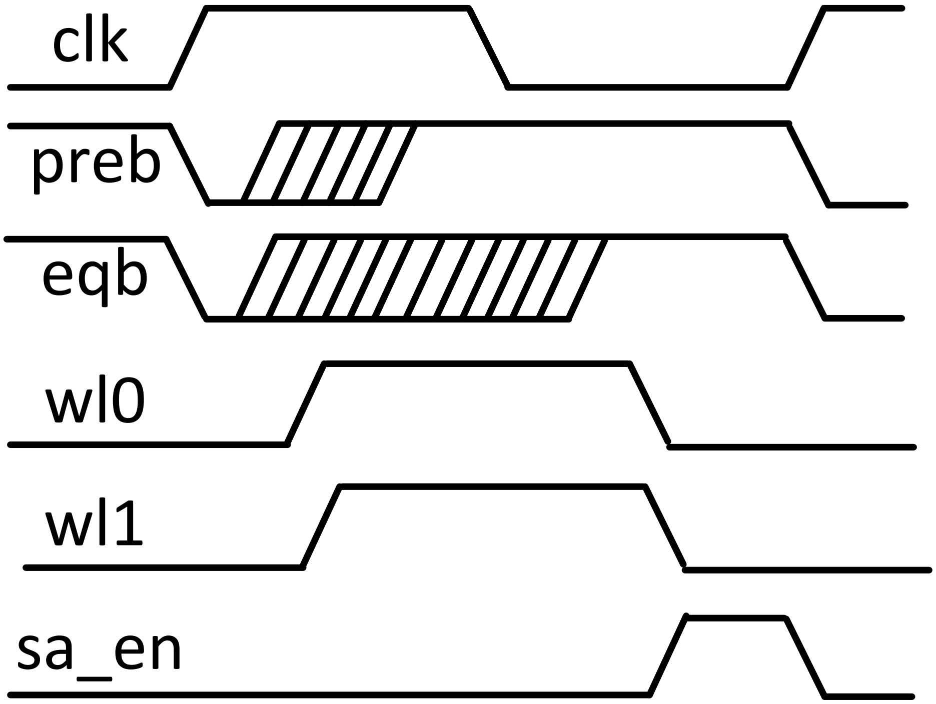


Figure 4.4 Timing waveform for proposed PUF

The response read out at the end for the proposed PUF is dependent on the following:

i. SRAM initialization value

ii. Sequence length

iii. Sequence order

As the response depends on both the sequence length and sequence order, learning such a PUF, or its challenge response state is difficult. In the next section we discuss the measured results and also the sensitivity to the above three factors.

4.3 Measured results

The configurable memory has been designed in a 28nm FDSOI CMOS process. The dimension of the memory array is 64×64, to make a 4kb array. All PUF numbers are shown for 64 bit readout.

We first measure the basic 2-row PUF. This basic PUF is not sequence dependent as it is only (a,b); where ‘a’ and ‘b’ are the rows, whose word-lines are asserted simultaneously. We initialize ‘a’ and ‘b’ with 64-bit complementary (inverse) data of each other. For example if ‘a’ is initialized to all ones, ‘b’ is initialized to all zeroes.

Table 4.1 shows the results of 2-row experiment performed over 1024 different PUF per chip. Bias denotes the number of 1s appearing on an average in every 64 bits read. BER (bit-error rate) is measured by repeating the experiment several times, and measuring how many bits out of 64 changes. Ideal bias is 32 (50% of 64) and ideal BER is 0. The measured bias is acceptable as it is close to 50%, and the BER is low and can be further improved by masking. Inter-chip HD (Hamming Distance) is measured across chips for the same experiments, and has a mean of 30.8.

Table 4.1 Two-row simple PUF measurement results.

Parameter	**Value**
Bias	32.8
BER	3.16
Inter-chip HD mean	30.8
Inter-chip HD Sigma	4.3

To find out the sensitivity of final response to initialization value, we perform the 2-row experiment with data background (BG) initialization and flipped data background (FBG) initialization. For example if row ‘a’ had all ones and row ‘b’ all zeroes in data BG, then in FBG row ‘a’ has all zeroes and row ‘b’ has all ones at initialization. Table 4.2 summarizes the results. The hamming distance between the two different initializations (Inter-init HD) is significant.

Table 4.2 Two-row PUF: Initialization sensitivity measurements.

Parameter	Data Init	Flipped Data Init
Bias	32.9	32.6
BER	3.17	3.16
Inter-init HD mean	24.1	
Inter-init HD Sigma	4.4	

The other two major factors that change the final response are sequence length and sequence order. For this we run three different sequence types as explained below, and also measure the inter-sequence hamming distance, to show the sequence dependent nature of the proposed PUF circuit. The inter-sequence hamming distance is the bit difference between the responses for same rows interaction on a chip, but in a different order.

Sequence based experiments:

i. 3-Row cyclic is of the form: {(a,b); (b,c); (c,a);}.

ii. 3-Row straight is of the form: {(a,b); (b,c);}.

iii. 4-Row straight is of the form: {(a,b); (b,c);(c,d)}.

The 3-row sequences have 6 different sequence permutations for interaction between the same set of three rows, whereas the 4-row sequence has 24 different sequence permutations.

In Table 4.3, we show the measured results for the different sequence based experiments. Figure 4.5 shows the inter-sequence hamming distance distribution for 3-row cyclic. Figure 4.6 and Figure 4.7 shows the inter-sequence hamming distance distribution for 3-row straight and 4-row straight respectively.

Table 4.3 Sequence based PUF measurement results.

Parameter	**3-Row Cyclic**	**3-Row Straight**	**4-Row Straight**
Bias	35.9	37.7	35.7
BER	3.3	3.26	3.33
Inter-sequence HD mean	4.9	12.3	16.9
Inter-sequence HD Sigma	2.7	5.1	5.9
Inter-chip HD mean	30.8	28.4	28.3
Inter-chip HD Sigma	4.3	4.1	4.4

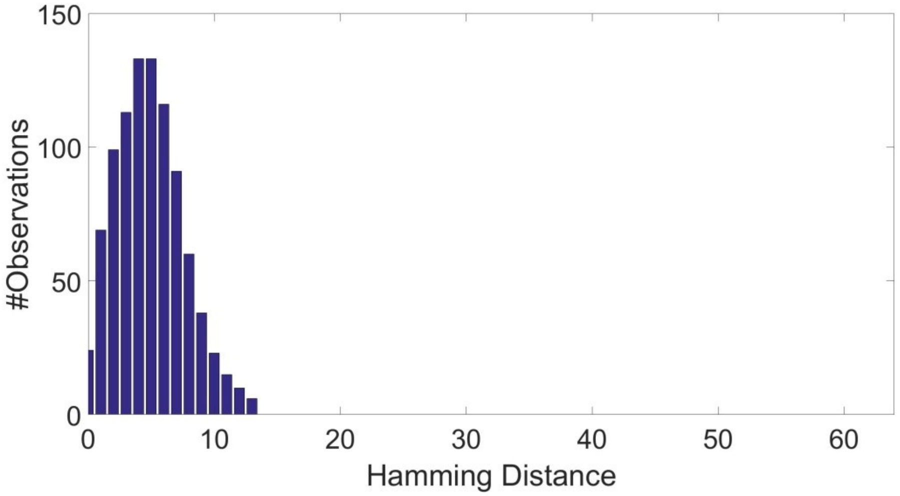


Figure 4.5 Inter-sequence hamming distance distribution for 3-row cyclic

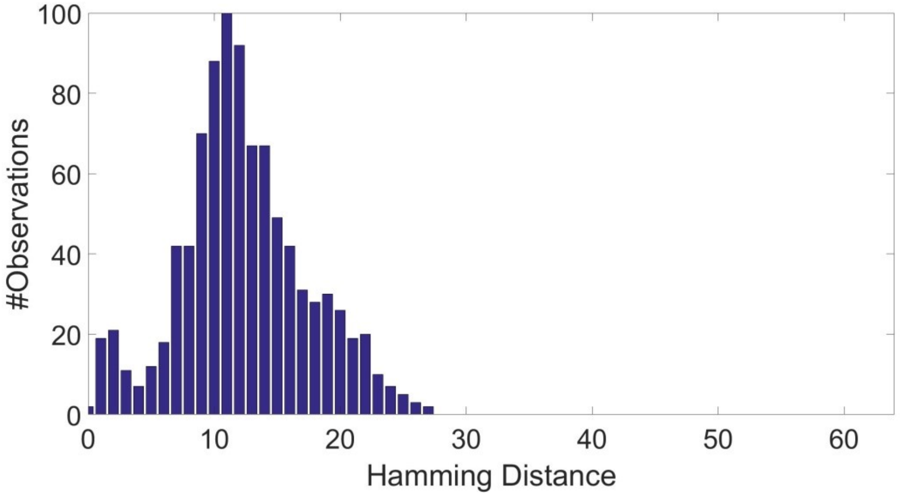


Figure 4.6 Inter-sequence hamming distance distribution for 3-row straight

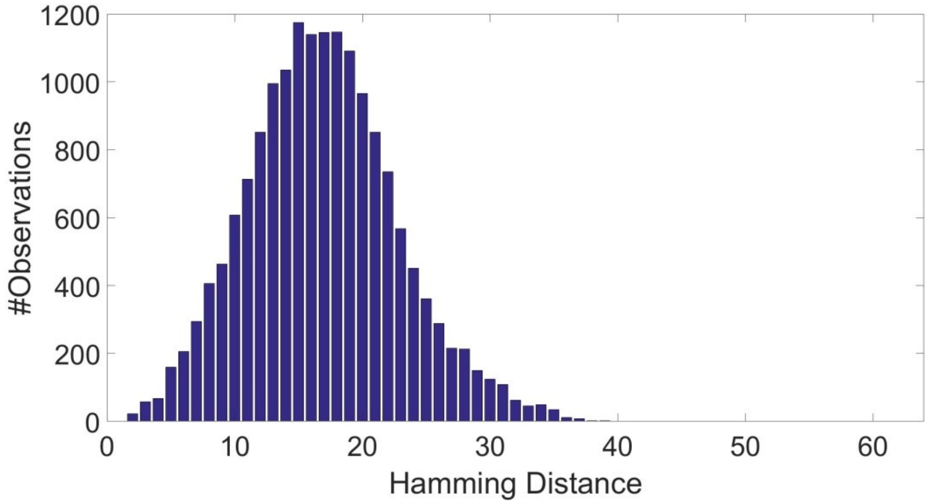


Figure 4.7 Inter-sequence hamming distance distribution for 4-row straight

4.4 Conclusion

Cyclic sequences reduce inter-sequence hamming distance, but even short straight sequences are useful. We observe that as sequence length increases, the inter-sequence hamming distance increases. Also the inter-chip hamming distance is good for all sequence based experiments.

The proposed PUF is the first sequence dependent PUF. The PUF has a large challenge response space, and can be run in many configurations based on the initialization and sequence length used. These features make it extremely hard to predict the response from this PUF, hence making it more secure.

CHAPTER 5

Low power SONOS flash memory

5.1 Introduction

Low power sensing systems have limited energy budgets and typically operate with low activity rates. A common use case involves intermittent wake-up, logging of sensed data, followed by a return to a very low-power or near-zero-power state. For such systems, the ability to completely power down to save energy is vital; hence they typically save data in a non-volatile memory such as embedded flash.

However, conventional flash technologies consume significant energy in write-intensive applications [74]. Unlike floating-gate flash, which uses a high current hot carrier injection (HCI) based program [75], SONOS (Silicon-Oxide-Nitride-Oxide-Silicon) flash cells perform both erase and program using Fowler Nordheim (FN) tunneling. The ONO acts as the trap layer for this transistor. Using FN tunneling makes SONOS inherently low power [76], however FN tunneling programming is $\sim 10^3\times$ slower and hence programming energy is typically similar at several nJ/bit [77].

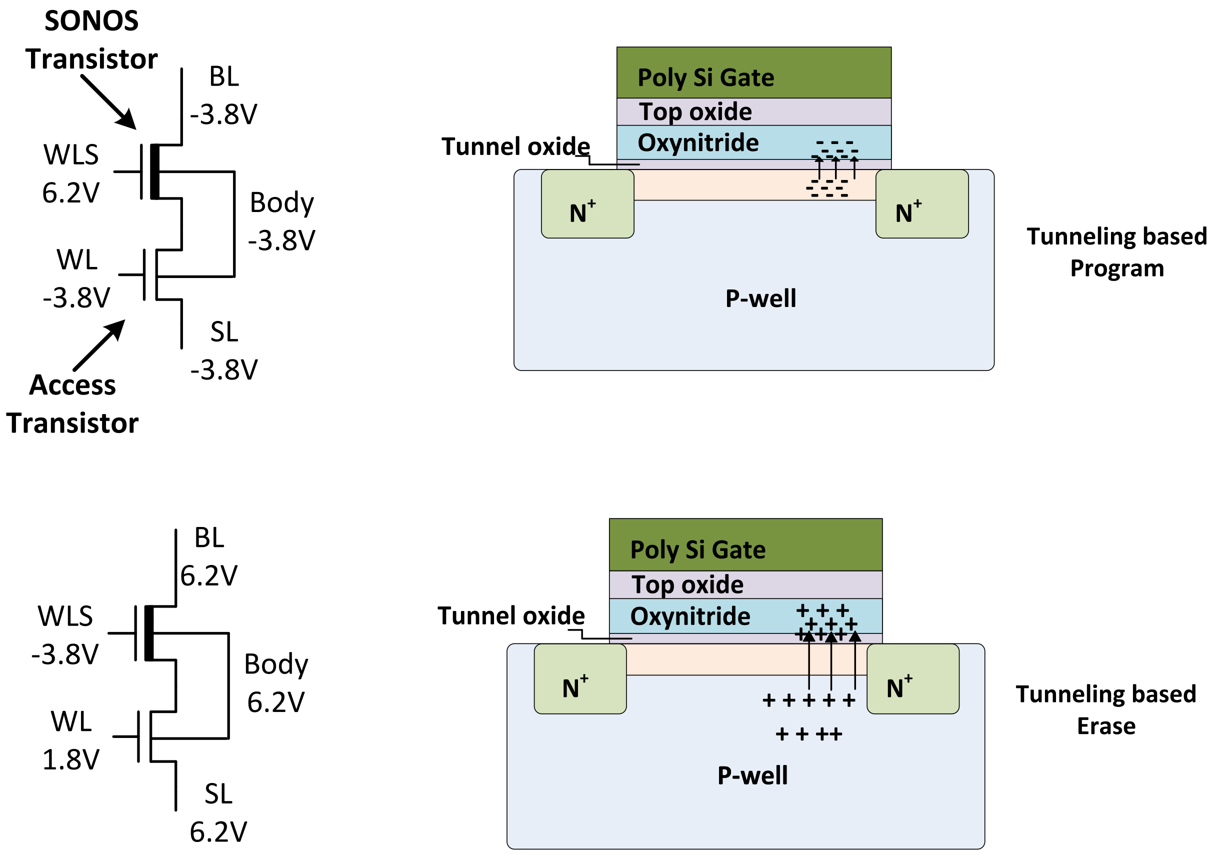


Figure 5.1 Erase & program voltage for SONOS (Silicon-Oxide-Nitride-Oxide-Silicon) using FN tunneling.

5.2 Proposed SONOS design

5.2.1 Overview of proposed design

This work re-architects a SONOS flash array to achieve very low programming energy by: 1) supporting an ultra-wide 1Kb per program cycle, enabled by low FN tunneling program power and a dedicated ‘transition pump’ to support high current draw when entering program phase; 2) charge sharing and charge recycling using a multi-output transition pump, allowing for the transition of 1K bit-lines and source-lines within strict power budgets consistent with sensor node platforms; 3) energy efficient charge pumps to support the low DC tunneling currents; 4) a program and erase reference cell in each 1Kb row, to track inhibit and bit-line disturb over the lifetime of the flash row, improving read margin. Combined, these approaches reduce the SONOS program energy to 122pJ/bit, which is ~10× lower than conventional floating-gate Flash, and maintains average programming power at 125μW. Through wide programming the memory achieves 1Mbps throughput, which is comparable to hot carrier injection based programming and enables logging of large data sets (e.g., audio) during short wake-up periods.

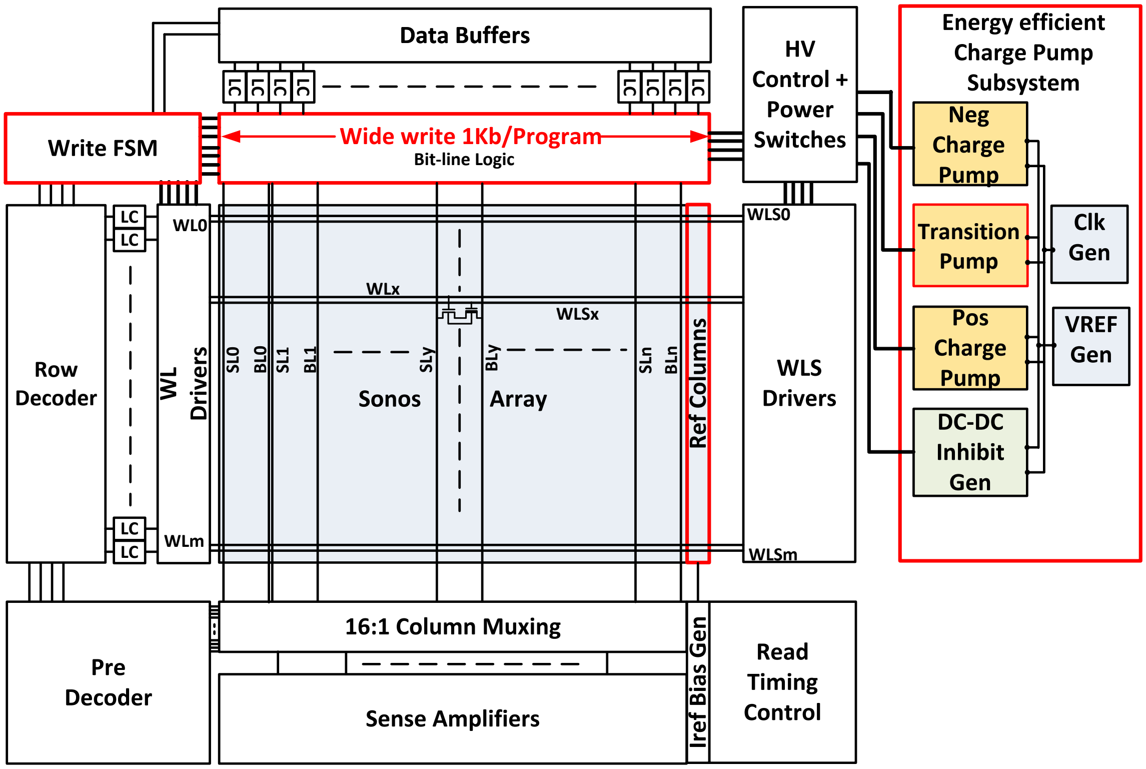


Figure 5.2 Block diagram of proposed wide-write low power SONOS flash with transition pump for high current.

5.2.2 Ultra-wide program

Figure 5.1 depicts the program and erases processes in a 130nm SONOS 2T flash cell, highlighting the required voltages for each step. Programming sets both the bit-line (BL) and source-line (SL) to -3.8V for selected cells and 1V for half-selected cells. Once the program voltages are set, tunneling based program operation takes ~1ms, during which the current is very small. However, as shown in Figure 5.3, there is a large transition current at the beginning of a program cycle due to initial charging/discharging of BLs and SLs. This transition current limits the number of cells that can be programmed simultaneously.

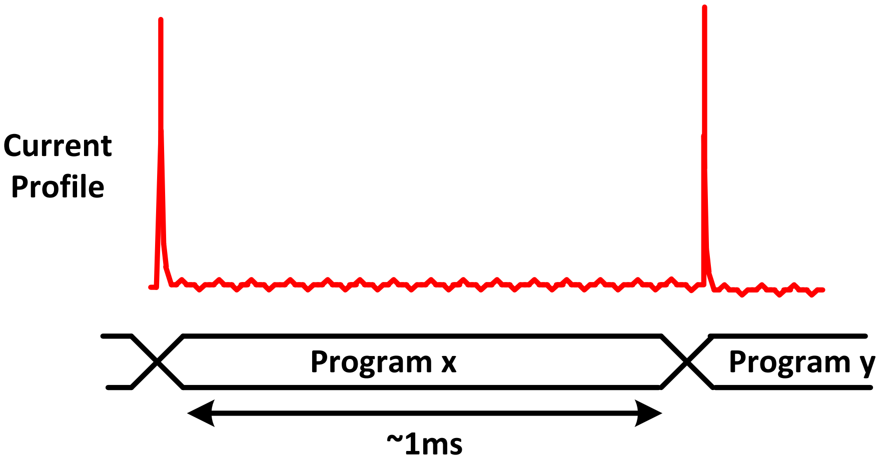


Figure 5.3 Current profile for FN tunneling based program. Large transition current at beginning of each program cycle.

An ultra-wide write can improve energy efficiency and throughput, and hence we propose a high current 'transition charge pump' to support the large transient current when entering program mode. Figure 5.4 shows the transition pump and its connection to the BL selection logic.

5.2.3 Transition pump: Charge sharing and charge recycling

The BL logic uses the current and previous write's data values to selectively connect a bit-line to one of four voltage rails: stable -3.8 V, stable 1V, 'Rising Rail', or 'Falling Rail'. Data input lines that are the same as the previous write value remain static. Bit-lines that need to be charged up from -3.8V to 1V are switched to the 'Rising Rail', which is increased in voltage in several steps by the transition pump, and then switched to the stable 1V rail. Falling bit-lines are similarly transitioned from 1V to -3.8V. The write state machine monitors the comparator output from the charge pump loops to determine voltage stability, ensuring safe entry to/from the transition phase and between transition steps. By not transitioning bit-lines needlessly, the proposed approach improves peak power and energy efficiency.

In addition, changing the transition rail voltage in 4 smaller steps instead of one large step reduces conduction loss in the charge pump. Also, step 3 (Figure 5.4, bottom) employs charge sharing by shorting the rising and falling transition rails, consuming no energy. Finally, by charging and discharging the transition rails simultaneously out of the same pump (step 4), charge is recycled between the opposite transitioning rails, instead of sourcing or sinking the entire charge from Vdd or Gnd, further improving energy efficiency. The charge sharing and charge recycling transitions are 3.3× more energy efficient than a single step transition of BL and SL. Furthermore, stepped transitions using a dedicated transition pump reduces peak power by 14.7× (simulated).

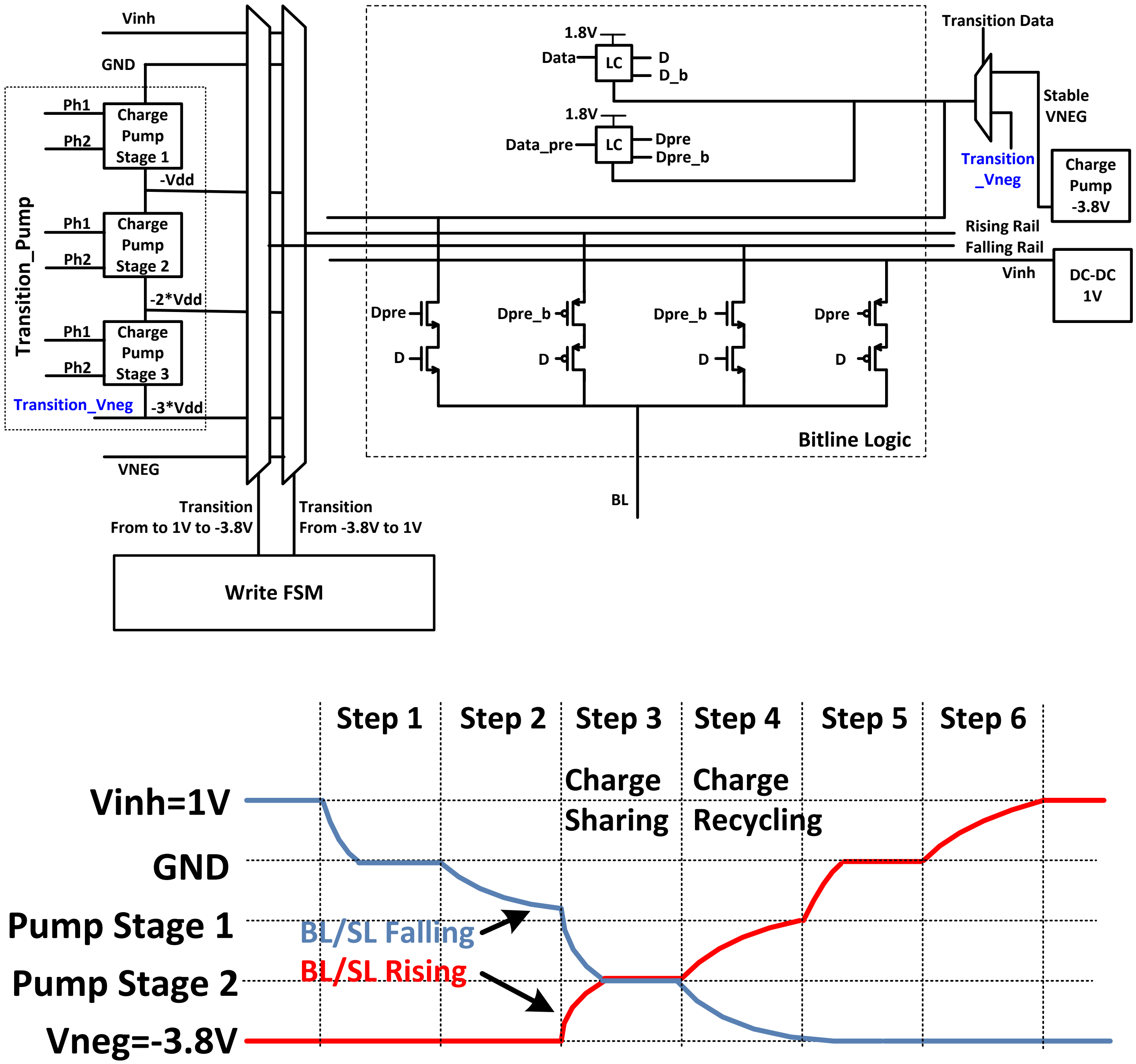


Figure 5.4 Stepped-transition of 1K BL & SL using transition pump to reduce peak power, enabling wide write.

This gain enables the concurrent programming of 1Kb without collapsing the charge pump voltages, which further reduces programming energy per bit by both amortizing fixed programming energy overhead over many bits as well as increasing throughput. After the transition phase, the transition pumps are disconnected and the low programming DC currents are supplied by high efficiency closed-loop charge pumps. Figure 5.6 shows a flowchart for the proposed SONOS program operation.

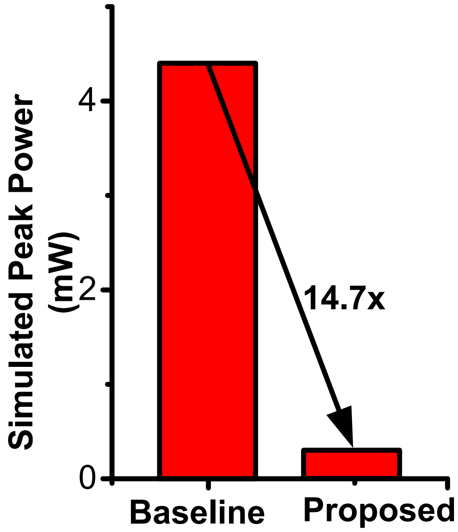


Figure 5.5 Transition pump reduces peak power, enabling wide write.

5.2.4 Energy efficient charge pump

Figure 5.7 shows the negative charge pump design for the proposed flash array. The negative charge pump consists of three Dickson stages with its clock controlled by an output voltage monitoring loop. To generate a positive intermediate voltage for comparison from the negative charge pump's output, a diode stack divider is placed between a 1.2V reference voltage VREF [78] and the negative pump output. In Figure 5.7, as VNEG falls, the voltage divider output falls, allowing the use of a clocked comparator with a positive reference voltage to gate the clock and control VNEG. The clock generator and on-chip voltage reference generator are shared between all voltage generation blocks (positive, negative, and transition pumps as well as the 1V DC-DC converter). The clock frequency is trimmed on-chip to reduce charge pump control loop power. Figure 5.8 shows simulated efficiency over load current for the voltage generation blocks.

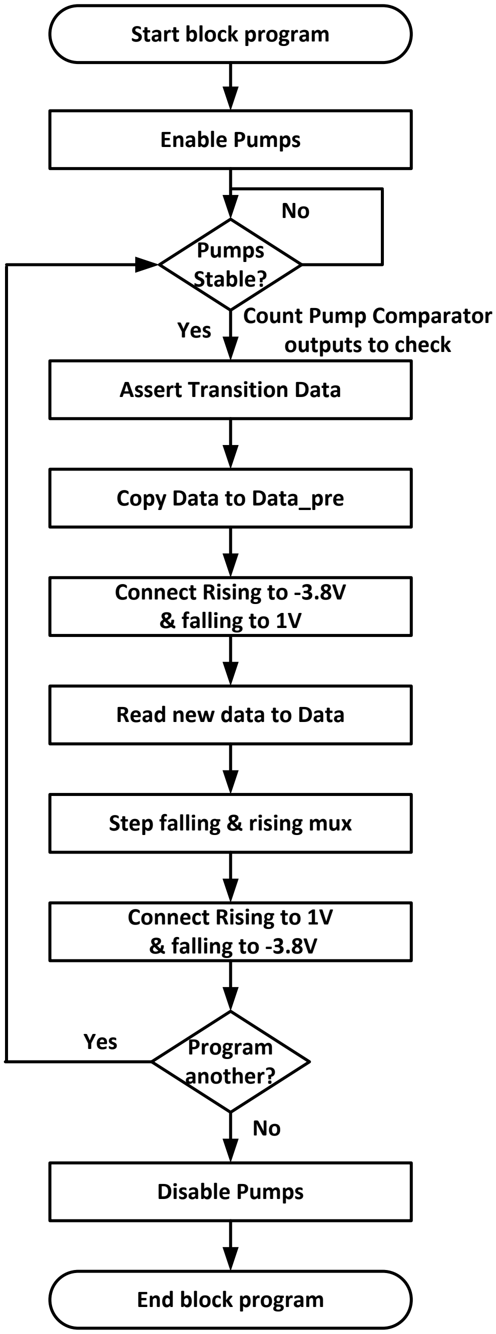


Figure 5.6 Write flow chart implemented in the proposed SONOS program controller

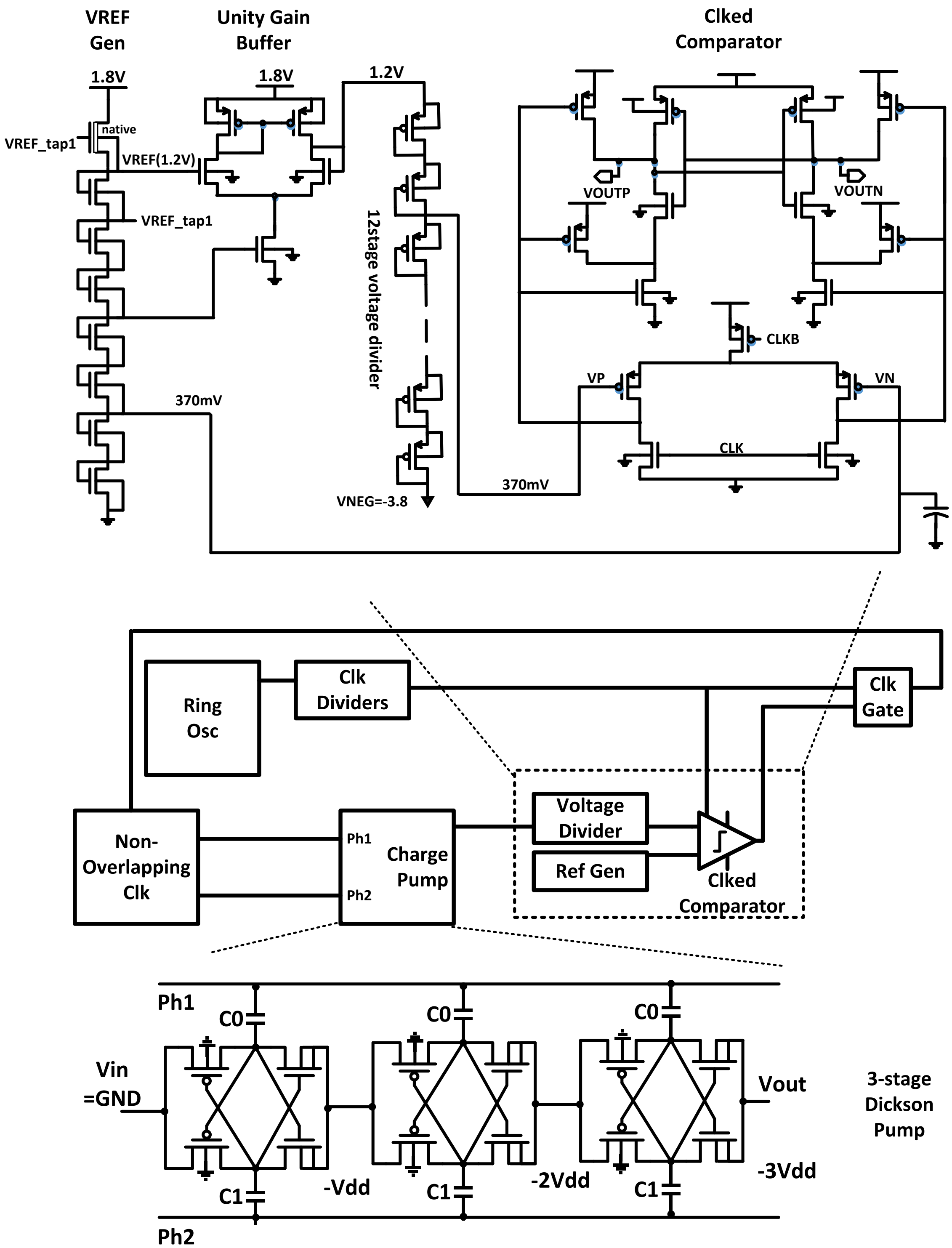


Figure 5.7 Negative charge pump design with control loop.

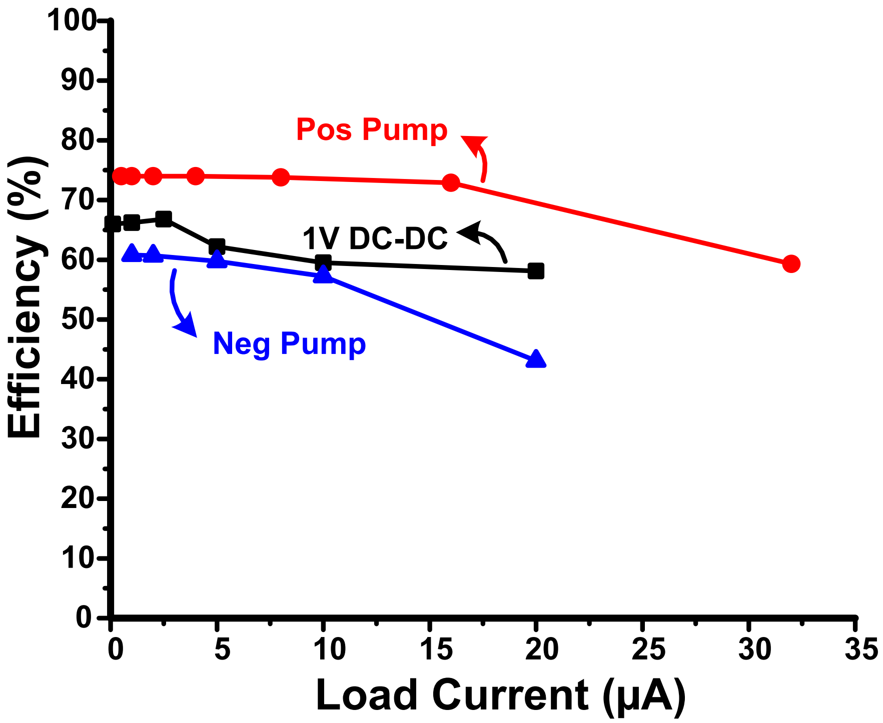


Figure 5.8 Simulated efficiency over load current for negative and positive pumps and 1V DC-DC generator.

5.2.5 SONOS read

Figure 5.9 shows the read circuit of the proposed array. SONOS flash has two main disturb mechanisms during program. First, an erased bit-cell in a selected row can become soft programmed, despite the applied inhibit voltages. Second, an already programmed cell in an unselected row can be soft erased when inhibit voltages are applied to its BL & SL. The erase and program currents in a particular row will therefore shift with the number of past write cycles during which the row was selected or not selected.

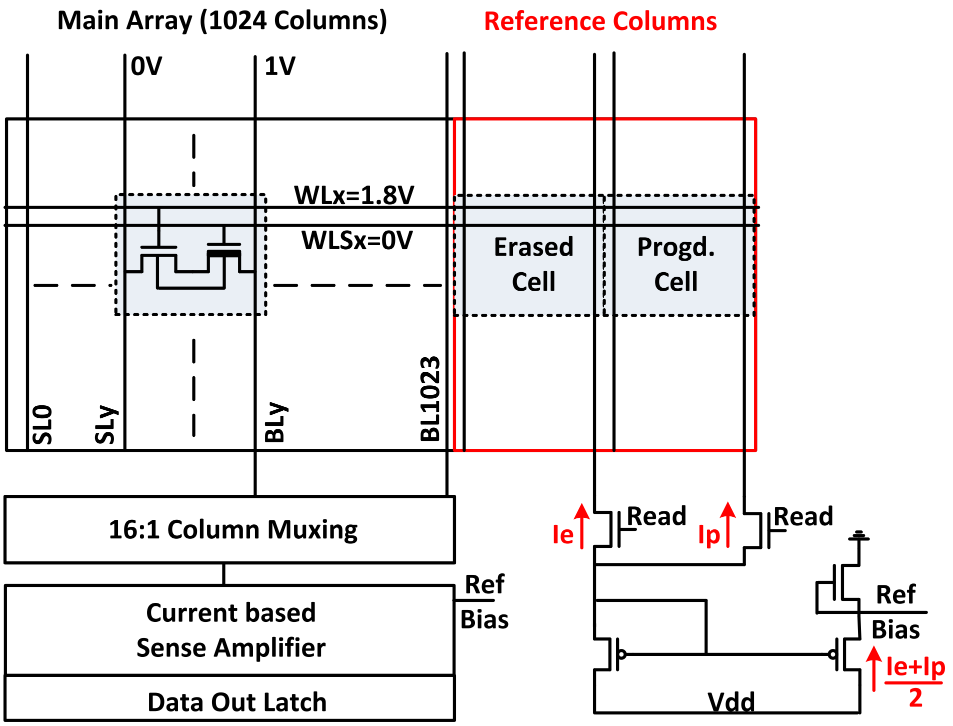


Figure 5.9 Read reference current generation for proposed flash improves read margin by 23%

To compensate for these shifts during read operation, we use two reference cells for each row (1024 bit-cells); the WL & WLS voltages are the same across the row including the reference cells. One reference cell is initially programmed, while the other is erased; as the array is accessed through normal use these cells track the extent of soft erase and soft program, respectively. A current mirror generates (Ierase + Iprogram) / 2 as the reference current for reading the row and the resulting reference bias is supplied to the current sense amplifiers (CSAs). During erase verify, we still use Ierase / 2, as all cells in the row are in erased state in that phase. The array performs 64b reads and hence a 16:1 column mux is used before the CSAs. Using the two reference voltages improves read margin by 23% in simulation with end of life model.

5.3 Measured results

The low power ultra-wide write SONOS flash design was fabricated in a 130nm technology. Figure 5.10 shows the measured read results; maximum read frequency is 40MHz at nominal Vdd of 1.8V, with a corresponding read energy of 2.23pJ/bit. Read operates down to 1.2V at 7MHz, at which point read energy is minimized at 1.2pJ/bit. Program energy is also measured across 0° to 85°C, showing 20% increase at high temperature, as shown in Figure 5.11. The measured distribution of program energy is shown for 21 chips, with a mean of 122pJ/bit.

Table 5.1 compares the proposed flash to other tunneling and HCI based NOR flashes memories. The measured programming energy for the proposed 1024x260 bit flash is 122pJ/bit with an average power of 125µW for a 1kb program. Block erase consumes 29pJ/bit. By enabling wide-write FN programming, the program throughput is similar to hot-carrier injection based flash, with >10× better programming energy efficiency.

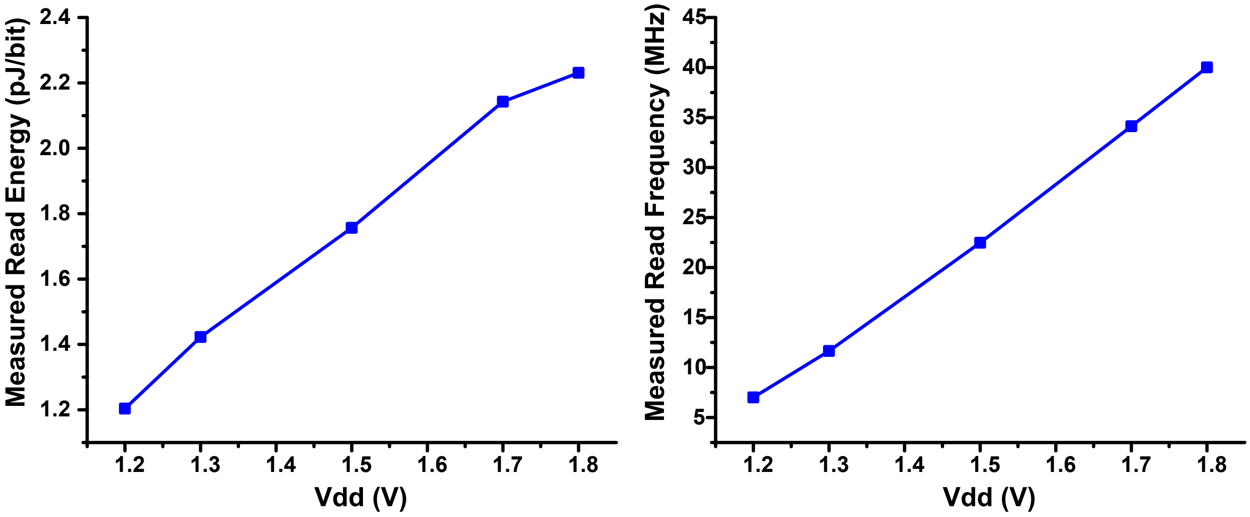


Figure 5.10 Measured read energy and frequency over Vdd. Read Vddmin is 1.2V.

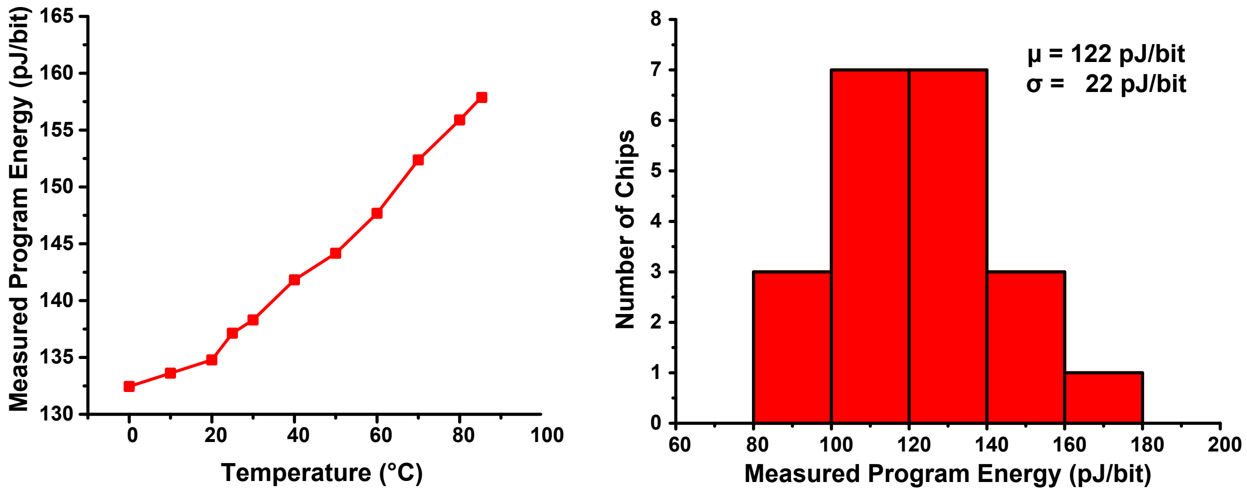
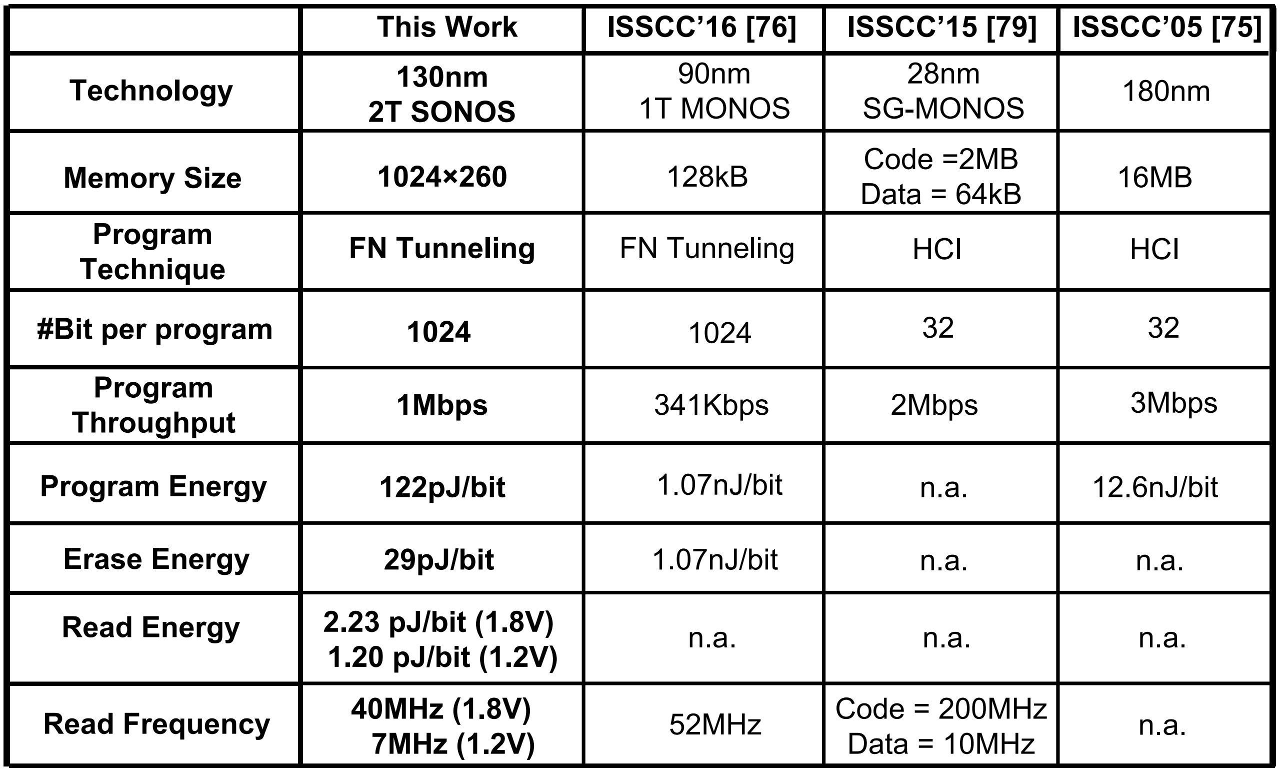


Figure 5.11 Measured program energy across temperature (left). Program energy distribution is plotted for 21 chips (right).

Table 5.1 Comparison with previous flash works.



	This Work	**ISSCC'16 [76]**	**ISSCC'15 [79]**	**ISSCC'05 [75]**
Technology	**130nm 2T SONOS**	90nm 1T MONOS	28nm SG-MONOS	180nm

Memory Size	**1024×260**	128kB	Code =2MB
			Data = 64kB
				16MB

Program Technique	**FN Tunneling**	FN Tunneling	HCI	HCI
#Bit per program	**1024**	1024	32	32
Program Throughput	**1Mbps**	341Kbps	2Mbps	3Mbps
Program Energy	**122pJ/bit**	1.07nJ/bit	n.a.	12.6nJ/bit
Erase Energy	**29pJ/bit**	1.07nJ/bit	n.a.	n.a.
Read Energy	**2.23 pJ/bit (1.8V) 1.20 pJ/bit (1.2V)**	n.a.	n.a.	n.a.

Read Frequency	**40MHz (1.8V)
	7MHz (1.2V)**
		52MHz	Code = 200MHz
			Data = 10MHz
				n.a.

5.4 Conclusion

Figure 5.12 shows a die picture of the proposed flash. The proposed flash memory achieves very low programming energy by supporting ultra-wide access. By using transition pump to support initial high power requirement, and then providing DC tunneling currents with energy efficient charge pumps, the proposed flash is able to achieve program energy of 122pJ/bit. The proposed SONOS flash is able to match the throughput of a floating gate flash, with ~10× lower energy.

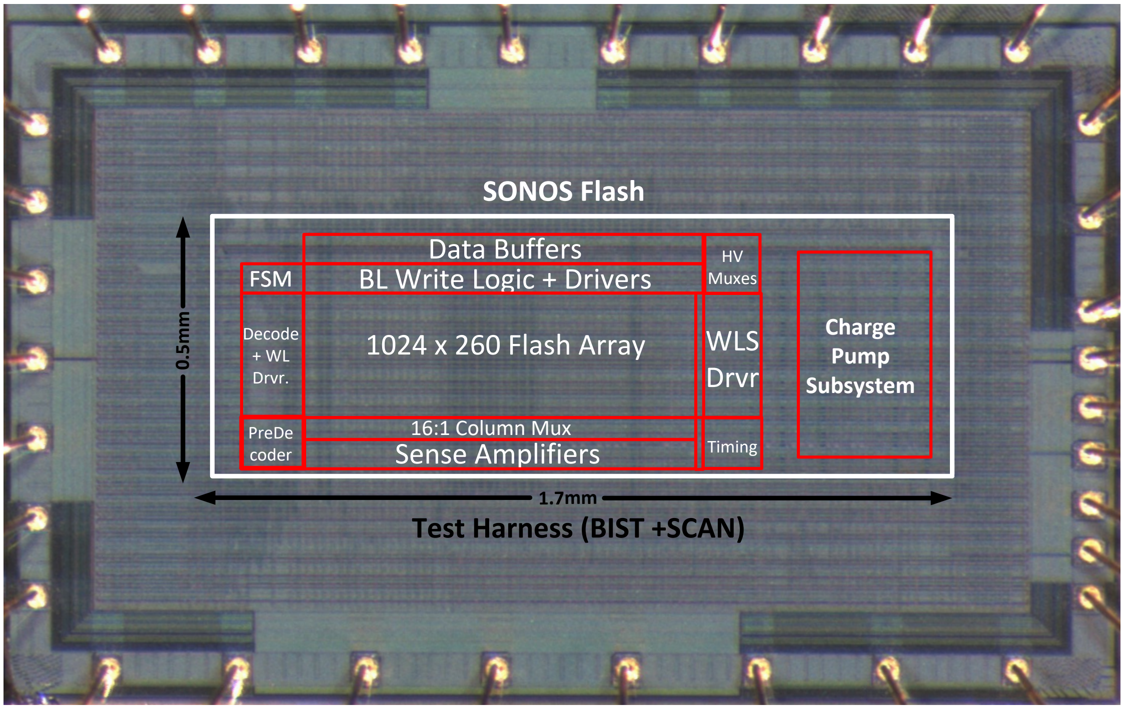


Figure 5.12 Proposed SONOS flash die photo

CHAPTER 6

Adiabatic FRAM

6.1 Introduction

In Chapter 5, we discuss the design for low write energy SONOS flash for data-logging in sensing systems. In this chapter we discuss an alternate solution based on FRAM (Ferroelectric RAM). Compared to flash memory, FRAM has significantly faster write access time, lower write energy, and orders of magnitude better endurance.

On the other hand, Flash has better density than FRAM, and is scaling better. Flash requires high voltages to write, which incurs high area penalty for the additional peripherals (like charge pumps). Flash is therefore more area efficient for larger array sizes. FRAM is more suitable for smaller write intensive memory in sensor nodes.

6.1.1 FRAM concept

FRAM bit cell structure is very similar to DRAM, except it has a non-linear capacitor, Ferroelectric capacitor (FeCap), making it non-volatile. The FeCaps are different from regular capacitors, as they use ferroelectric material instead of regular dielectric one. A widely used ferroelectric material is lead zirconate titanate (PZT), as shown in Figure 6.1. It has two stable states which have opposite polarization. Initially, the polarization of each unit cell is random. The polarization of all cells aligns on providing an appropriate external electric field. The polarization direction determines the two stable states '0' or '1'. When the external electric field is removed, the FeCap still retains some polarization alignment, making it non-volatile.

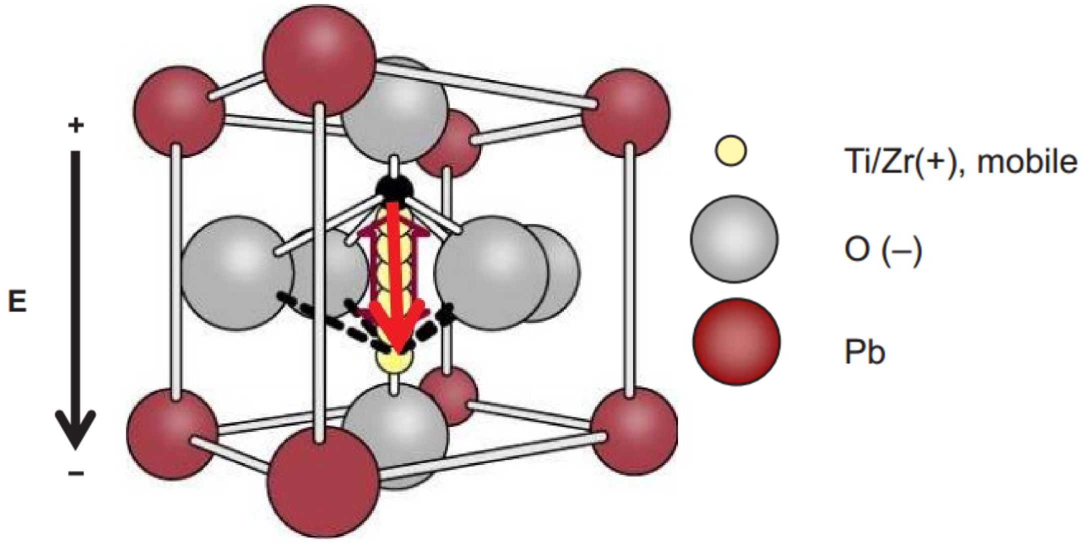


Figure 6.1 FeCap based on Lead Zirconate Titanate

Figure 6.2 displays the hysteresis loop measured for a ferroelectric capacitor. The charge on the capacitor reaches saturation charge (Qs) when the voltage across the capacitor is Vdd. When the voltage is 0V, the capacitor stays in one state, either '0' or '1', depending on the remnant charge Qr or –Qr. The charge difference of these two states can be used for detection of the data stored in the capacitor.

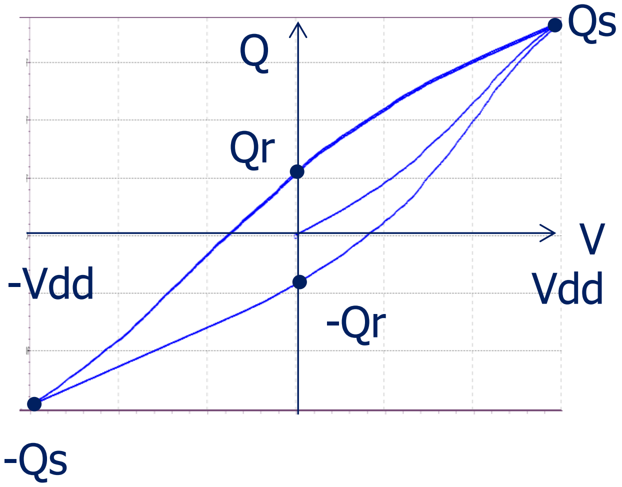


Figure 6.2 FeCap hysteresis loop

This charge difference determines the read margin. The read margin can be affected by corners, but is a large function of voltage, as shown in Figure 6.3. This margin erosion at low voltage makes read sensing hard, and less reliable.

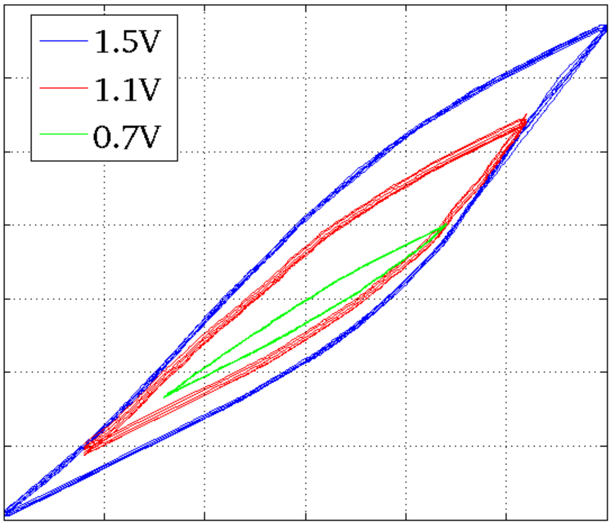


Figure 6.3 FeCap hysteresis loop for different supply voltages

6.2 Conventional FRAM design

The 1T1C FRAM cell [59][60][61] consists of an access transistor in series with the ferroelectric capacitor. In addition to bit-line (BL), and word-line (WL), the other end of the FeCap is connected to a Plate Line (PL) as shown in Figure 6.4.

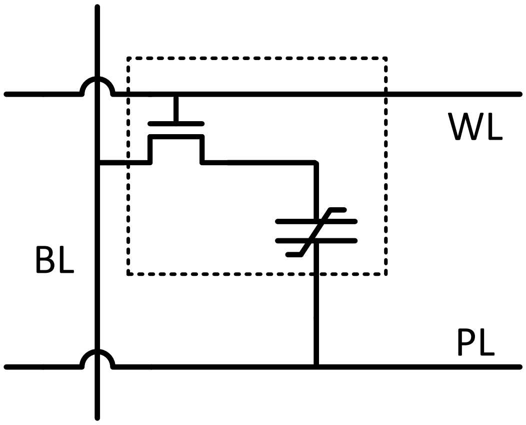


Figure 6.4 A 1T1C FRAM bit cell

Figure 6.5 shows a write '1' sequence, where PL is kept at 0, while Bit Line BL is asserted. For write 0, PL is asserted, while BL is kept at 0. Read operation is similar to writing a '0', except that BL is left floating after pre-discharging it to ground. When the access transistor is enabled, the ferroelectric capacitor will share its charge with BL parasitic capacitor, which causes the BL voltage to rise. Depending on the previous state (0 or 1) of the bit cell, the remnant charge stored in the ferroelectric capacitor is different which leads to different BL voltage rise. The difference between the BL voltages developed for the two states is the read margin, 'ΔV'. A larger margin will lead to a faster and more robust read.

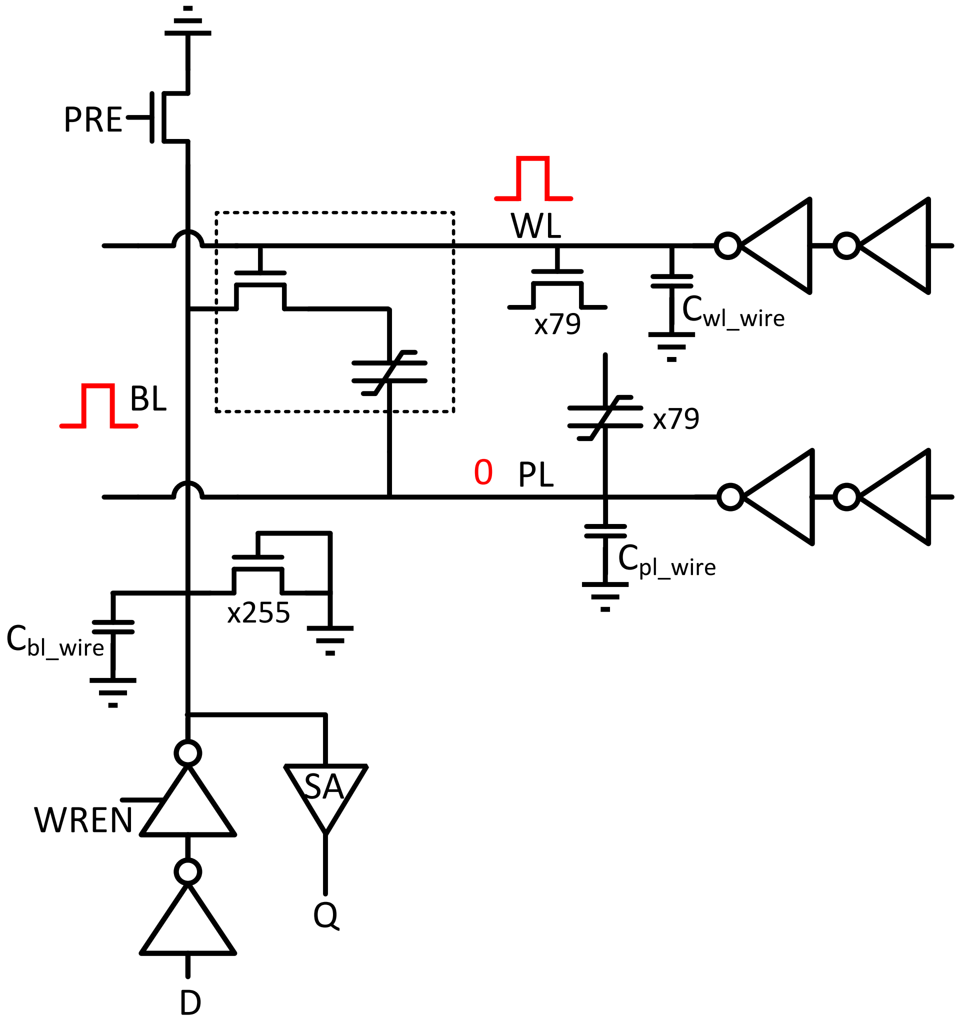


Figure 6.5 Conventional 256x80 FRAM array model: Write ‘1’ example

6.3 Proposed FRAM adiabatic design

In the base line FRAM array, the access energy can be divided into the following: transitioning WL, PL and bit-lines, as well as the actual charging of the FeCaps. The bit-line transition energy dominates the energy/bit consumption, as an entire BL needs to transition for each bit in the array, whereas the other components are amortized over the number of bits in the row. The transition energy required for the bit-line can be reduced if we step the bit-line in multiple steps. To reduce the bit-line energy component to be negligible, we can make it transition in a sinusoidal form.

The proposed adiabatic FRAM is based on such a sinusoidal waveform. The PL is oscillated between 0 and Vdd, using an off-chip LC tank. The basic adiabatic concept of the proposed idea is that, whenever a bit-line has to transition up or down, it connects to PL to transition, and then clamps to the high or low value it has moved to.

Figure 6.6 shows the FRAM array design with adiabatic write capability. Figure 6.7 shows the corresponding waveforms. The LC tank shown in the dashed box is off-chip. The capacitance for resonance is composed of the external cap, along with the cap of the selected word's FeCaps and BL wire capacitance.

To start the resonance, the LC tank capacitor is stepped to VDD/2, which leads PL to resonate between 0 and VDD. At the peak of the resonating sine wave we generate a pull-up (PU) pulse, and a pull-down (PD) pulse at the trough. The PU & PD pulse are used to restore the amplitude of the resonance waveform. The PU and PD pulse are also used to generate the timing information for the write-enable (WREN) and PL-enable (PLEN) transmission gate mux controls. The BLs are transitioned adiabatically, by connecting BLs to resonating PL by asserting PLEN. The BLs are clamped to the write data by de-asserting PLEN, and asserting WREN. The starting point and ending point of BL for writing '0' is 0 of PL and for writing '1' is VDD of PL.

Each row needs one and a half cycle of resonance of PL to write a new data, with its word-line asserted. The resonance frequency is therefore 1.5x the write frequency. Therefore, for every alternate row write we switch the order of writing ones and zeroes.

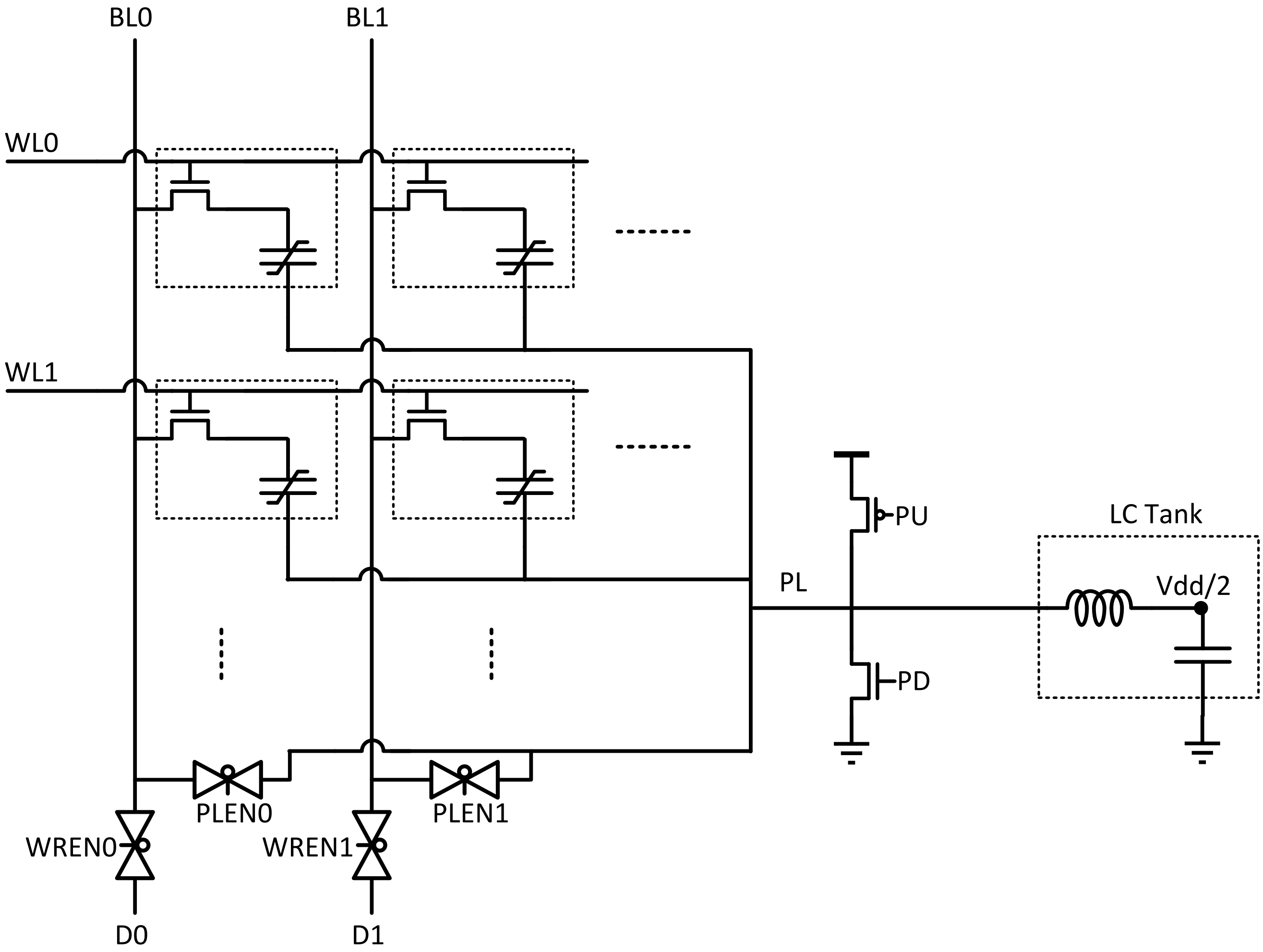


Figure 6.6 Proposed FRAM adiabatic design

6.3.1 PU/PD signal generation: Continuous comparator

The PU and PD pulse generation is critical to both timing and reducing energy. We use a continuous comparator with reference near Vdd for PU generation and another continuous comparator with reference near ground for PD generation. The circuit shown in Figure 6.8 is the continuous comparator for PU. The output pulse width of this comparator is wide and also is temporally not at the exact peak. The comparator is followed by a pulse shaping and pulse delay module. The pulse shaping can make the pulse thin and the delay module helps center it at the peaks and troughs.

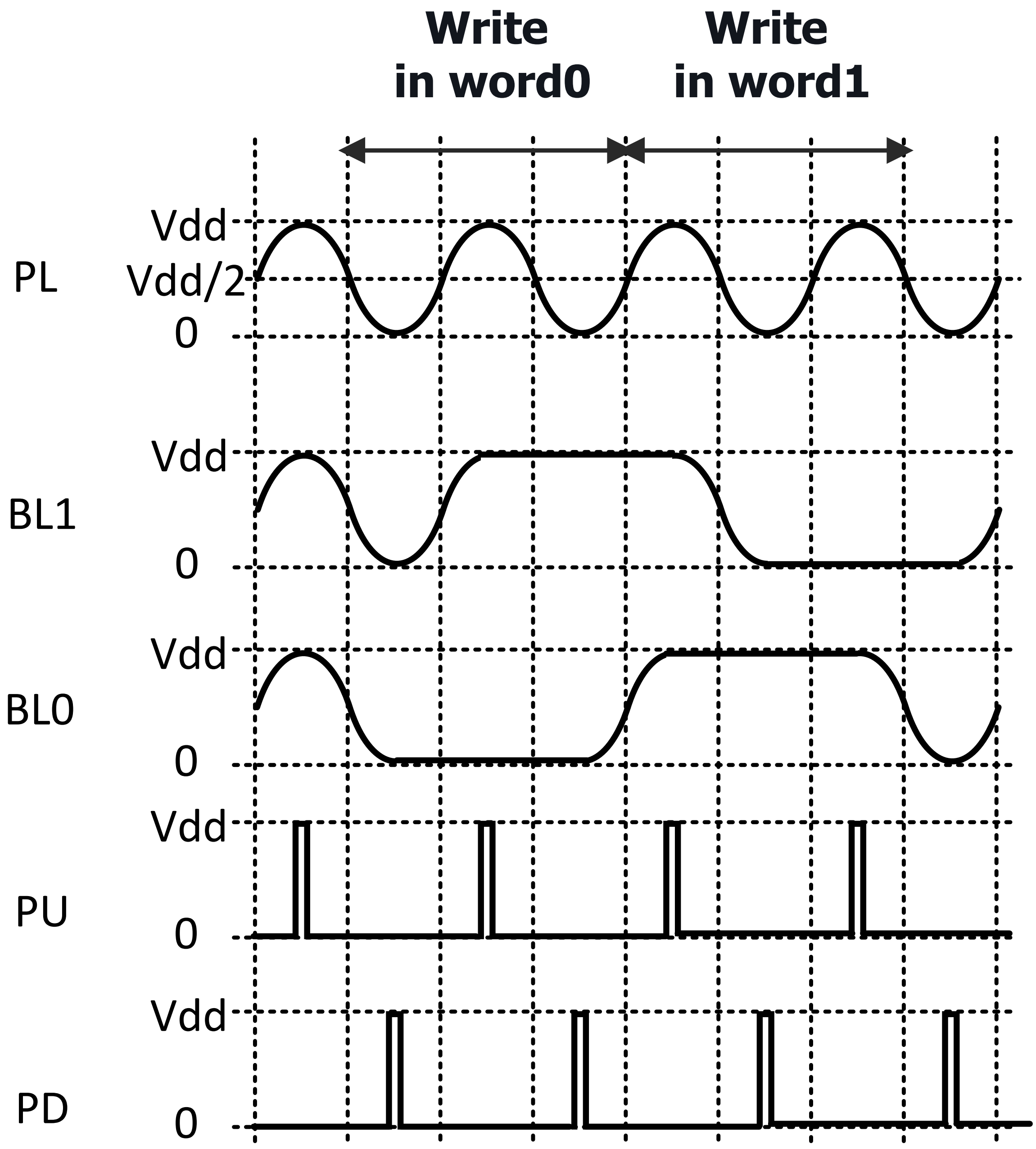


Figure 6.7 Timing waveform for proposed FRAM adiabatic design

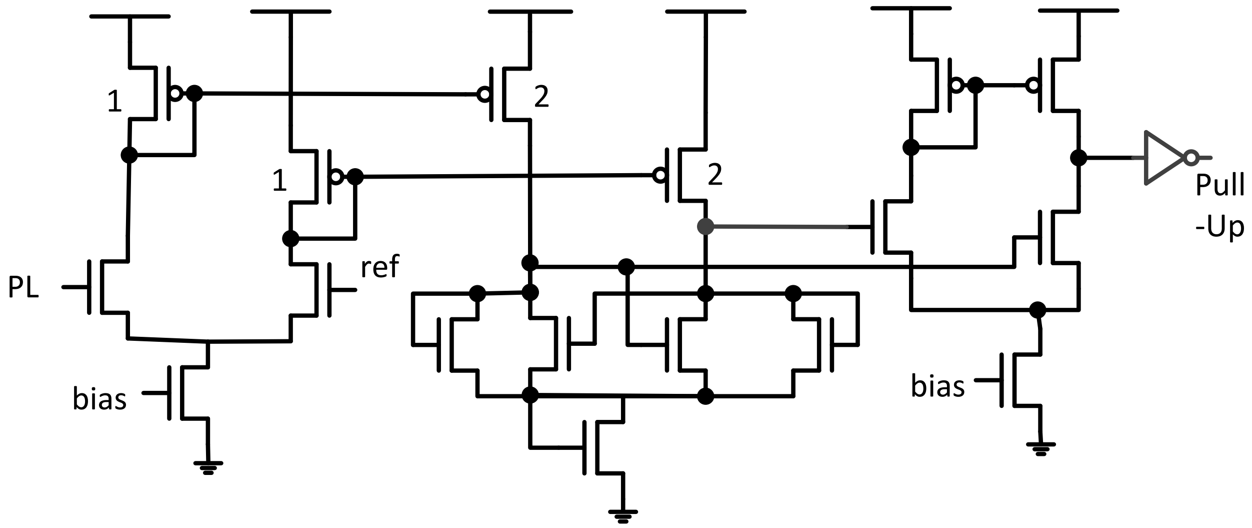


Figure 6.8 Continuous comparator design for PU (Pull-up) signal

6.3.2 Reconfigurable to 2T-2C cell

In the normal 1T-1C mode the array has full capacity, with a 2:1 column muxing. In this mode the sense-amp reads BL against vref. The margin in 1T-1C can be small, and for more robust operation we can reconfigure to a 2T-2C bit cell mode, as shown in Figure 6.9, where we store complementary data in the two capacitances. In this mode the array capacity reduces to half capacity with no column multiplexing and the sense amplifier is used in a differential mode.

6.3.3 Other peripherals

Additionally, we have a decoder followed by Word Line (WL) drivers which runs at 2.3 V. The high voltage ensures, there is no Vt drop across the nmos access transistor of the FRAM bit cell. The read and write finite state machine (FSM) generates the control signals for BL write and sense amplifiers, respectively. They also generate the signals for latching new address, new input data and latching the output data. These FSMs work of PU and PD pulses. Figure 6.10 shows the overall block diagram of the proposed adiabatic FRAM design.

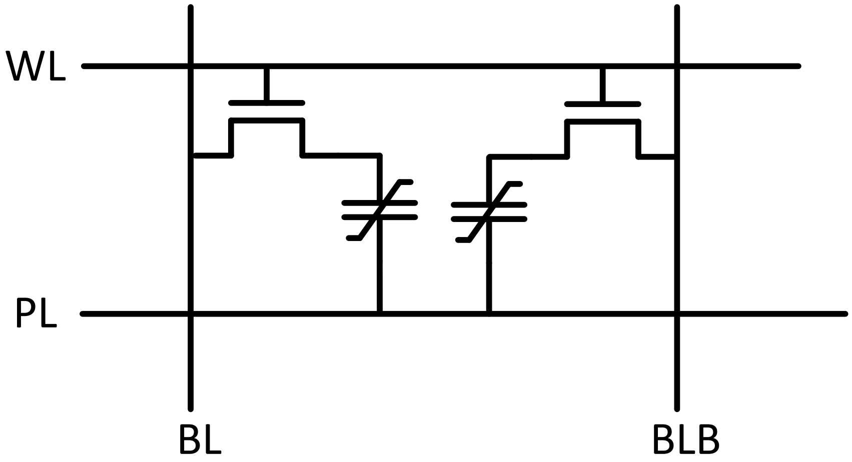


Figure 6.9 A 2T-2C bit cell configuration

Figure 6.11 shows the layout of the FRAM prototype chip in a 130nm process node. It has an array size of 512x80 in 1T-1C mode, and 256x80 in the 2T-2C mode. As the memory is asynchronous, the BIST to test the memory is also using PU and PD signals for generating inputs. The configuration and calibration bits are set using scan.

6.3.3 Simulation results

The FRAM array size changes the energy per bit. As the BL energy component is largely reduced by adiabatic transitions, the number of rows does not affect the energy per bit significantly.

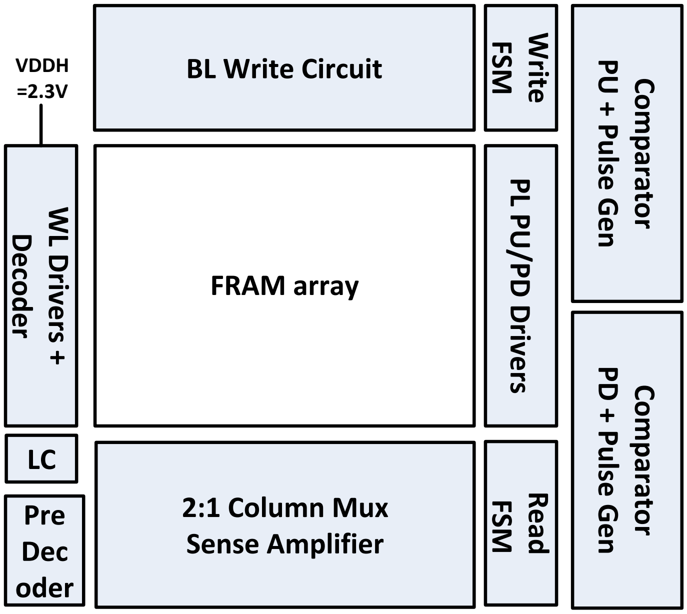


Figure 6.10 Proposed FRAM block diagram

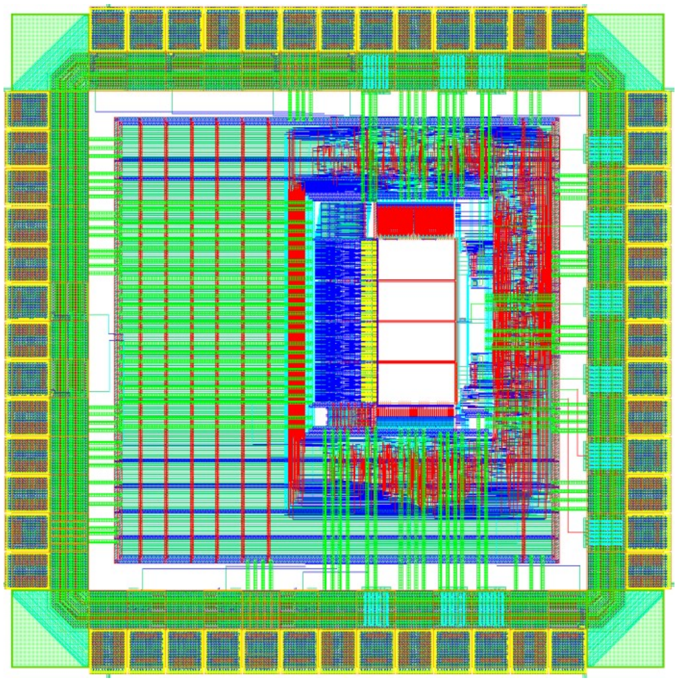


Figure 6.11 FRAM prototype chip layout

The write-energy per bit decreases for longer words in an array before stabilizing, as shown in Figure 6.12. This is the reason behind choosing an array size of 512x80 with 2:1 column multiplexing. Also, block write (writing multiple rows in series) decreases the write-energy per bit. Therefore, writing the whole array gives best energy numbers.

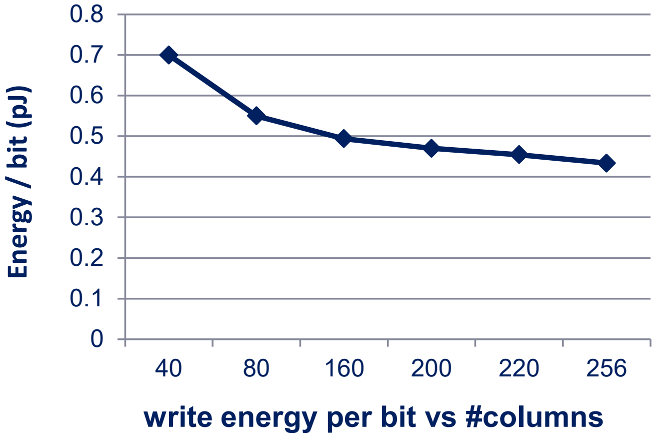


Figure 6.12 Write energy per bit for proposed FRAM versus number of columns in array

6.4 Conclusion

The proposed FRAM design uses adiabatic techniques to reduce the write energy by ~7× compared to the non-adiabatic design in simulations, whereas the read with write back energy reduces by ~4×. In read, the major saving is in the write back component, as bit-lines are floating during read. Thus, the proposed adiabatic FRAM design can be used as the non-volatile memory solution for write-intensive sensor applications.

CHAPTER 7

Conclusion and future directions

This chapter summarizes the contributions of this dissertation and highlights the future research directions derived from this work. A list of related publications generated as a result of this work, is provided in the last section of this chapter.

7.1 Contributions of this work

With the current scaling trends, power-density problems are limiting system performance. This thesis divides the application space into three categories. The divisions are performance intensive many-core systems, performance and energy constrained mobile systems, and low-activity energy critical IoT systems. This dissertation proposes cross-point circuit based solutions for energy-efficiency and power-density in each of the three design spaces.

Chapter 2 explains the need for an optimized high-radix switch for many-core systems. The proposed Hi-Rise switch leverages 3D integration, and is a true 3D switch which connects inputs and outputs across different silicon layers. Hi-Rise has hierarchical data-path architecture and provides built-in single-cycle arbitration using a new class based arbitration scheme. This arbitration scheme makes the 3D hierarchical switch's fairness comparable to that of a flat 2D switch, and can also be used for other 2D hierarchical switch architecture. A 64-radix, 128-bit width, 4-layer Hi-Rise evaluated on a 32nm technology has a throughput of 10.65 Tbps for uniform random traffic, which is a 15% improvement in throughput over a 2D design, along with a 33% area reduction, 20% latency reduction, and 38% reduction in energy per transaction.

In Chapter 3, a new BCAM/TCAM is proposed that can operate with standard push-rule 6T SRAM cells, reducing array area by 2-5× and allowing reconfiguration of the SRAM as a CAM. In this way, chip area and overall capacitance can be reduced, leading to higher energy efficiency for search operations. In addition, the configurable memory can perform bit-wise logical operations on two or more words stored within the array. The proposed configurable memory designed in 28nm FDSOI, with logic-in-memory has an energy efficiency of 0.6fJ/search/bit at 1V with an array bit density of ~5.4Mb/mm2. The area overhead for configurability over a conventional SRAM is only 7%. Thus, the storage memory can be reconfigured to perform compute operations as well, enabling new architecture and accelerator designs.

In Chapter 4, a new sequence dependent PUF is proposed. It has a large challenge-response space, as two or more rows of the SRAM can be allowed to interact, with different initializations. Also, it can be run in many configurations depending on the sequence lengths. These features make it hard to predict the PUF response. The SRAM PUF is implemented in 28nm FDSOI, and for a 4-row sequence the inter-sequence hamming distance is 26% while the inter-chip hamming distance is 45%.

In Chapter 5, a 130nm, 1024x260 embedded flash for low power sensing systems is presented. The embedded flash for IoT systems need to have very low write energy while meeting the peak power requirement. The proposed SONOS flash design supports an ultra-wide 1Kb/program cycle, enabled by low FN tunneling program power and a dedicated, transition pump. The proposed transition pump is tapped at multiple pump stage outputs to allow for charge sharing and charge recycling between transitioning high voltage bit-lines. This combined with the energy efficient charge pumps, brings down the program energy to 122pJ/bit with a 1Mbps throughput, and an average program power of 125 μW. The proposed SONOS flash is able to match the throughput of a floating gate flash, with ~10× lower energy.

Chapter 6 extends the concept of adiabatic logic design to memories. The major energy component in FRAM memory access is bit-line transition energy, which can be significantly reduced by transitioning the bit-lines using a resonating sine wave. The proposed FRAM design uses adiabatic techniques to reduce the write energy by ~7× compared to the non-adiabatic design in simulations, whereas the read with write back energy reduces by ~4×.

7.2 Future directions

Many future research directions open up based on the work presented here. This section discusses a few of the possible directions for each chapter.

The new class based arbitration scheme in Chapter 2 can be extended to other hierarchical data-paths. Hi-Rise can also be used to make new topologies for 3D chips like the one shown in Figure 2.17, which is a 2D mesh of 3D switches. This opens up the possibility of many new routing algorithms, for example, dimensionally ordered XY with 3D switch controlling the adaptable Z dimension routing, or physical-aware routing algorithms to utilize TSV routes more optimally.

Chapter 3 introduced a configurable SRAM memory with CAM and logic-in-memory features. This can be used as building block for new architecture and accelerators. Logic-in-memory features can be used to accelerate bitwise-logic dominated algorithms, like encryption and sequencing. CAM feature can be used to build associative processors or accelerate search. In addition to logical operations, arithmetic capabilities can be added to have full ALU functionality in the compute-memory, allowing it to be used as non Von-Neumann architecture. Also, new error correction algorithms may be developed for such in-memory operations, where only the result of the operation is read.

Chapter 4 introduced a sequence dependent PUF. It can be used to develop more stringent security protocols based on variable length challenges. Also, the PUF structure can be used in memory testing to make a map of relative bit cell strengths, and possibly replace weak cells, which have a higher probability of read disturb failures.

Chapter 5 introduced low power SONOS flash. For sensor node write applications, which update only one or few words every time it wakes up, the write operation is expensive as it comes with the penalty of erasing the whole block and rewriting the old data along with the updated word. By using a multi-level program, i.e. increasing the threshold voltage for program in steps, we can overwrite the data a few times on each word before incurring the penalty for erasing the entire block. For this we need two reference cells in each word, one with old program threshold and one with new program threshold value. As the read margin will be reduced for this scheme, it will need better sensing.

Finally, Chapter 6 introduces adiabatic FRAM. As the memory is asynchronous, and is running on the resonating sine wave, the work can be extended to frequency and phase lock the resonance waveform to the system clock. This will make the adiabatic FRAM to behave as a synchronous memory within the system.

7.3 Related publications

- S. Jeloka, R. Das, R. G. Dreslinski, T. Mudge and D. Blaauw, "Hi-Rise: A High-Radix Switch for 3D Integration with Single-Cycle Arbitration," *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, Cambridge, 2014, pp. 471-483.
- S. Jeloka, N. Akesh, D. Sylvester and D. Blaauw, "A configurable TCAM/BCAM/SRAM using 28nm push-rule 6T bit cell," *2015 Symposium on VLSI Circuits (VLSI Circuits)*, Kyoto, 2015, pp. C272-C273.
- S. Jeloka, N. B. Akesh, D. Sylvester and D. Blaauw, "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T Bit Cell Enabling Logic-in-Memory," in *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 1009-1021, April 2016.
- S.Aga, S.Jeloka, A.Subramaniyan, S. Narayanswamy, D. Blaauw and R.Das, "Compute Caches for Efficient Very Large Vector Processing," in *23rd IEEE Symposium on High Performance Computer Architecture (HPCA)*,Austin, February 2017 (to appear).
- S. Jeloka, J. Lee, Z. Li, J. Shah, Q. Dong, K. Yang, D. Sylvester and D. Blaauw, "A 122pJ/bit, Tunneling Based Embedded Non-Volatile Memory using Ultra-Wide Access and Charge Recycling". Pending Submission

BIBLIOGRAPHY

[1] J. Howard *et al.*, “A 48-Core IA-32 message-passing processor with DVFS in 45nm CMOS,” *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, San Francisco, CA, 2010, pp. 108-109.

[2] D. Wentzlaff *et al.*, “On-Chip Interconnection Architecture of the Tile Processor,” in *IEEE Micro*, vol. 27, no. 5, pp. 15-31, Sept.-Oct. 2007.

[3] S. Borkar, “Thousand core chips: a technology perspective,” in *Proceedings of the 44th annual Design Automation Conference* (DAC '07), 2007, pp. 746-749.

[4] J. Kim, J. Balfour and W. J. Dally, “Flattened Butterfly Topology for On-Chip Networks,” in *IEEE Computer Architecture Letters*, vol. 6, no. 2, pp. 37-40, Feb. 2007.

[5] N. Abeyratne *et al.*, “Scaling towards kilo-core processors with asymmetric high-radix topologies,” *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, Shenzhen, 2013, pp. 496-507.

[6] D. Fick *et al.*, “Centip3De: A 3930DMIPS/W configurable near-threshold 3D stacked system with 64 ARM Cortex-M3 cores,” *2012 IEEE International Solid-State Circuits Conference*, San Francisco, CA, 2012, pp. 190-192.

[7] D. Park *et al.*, “MIRA: A Multi-layered On-Chip Interconnect Router Architecture,” *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, Beijing, 2008, pp. 251-261.

[8] Feihui Li, C. Nicopoulos, T. Richardson, Yuan Xie, V. Narayanan and M. Kandemir, “Design and Management of 3D Chip Multiprocessors Using Network-in-Memory,” *33rd International Symposium on Computer Architecture (ISCA'06)*, Boston, MA, 2006, pp. 130-141.

[9] J. Kim, C. Nicopoulos et al., “A novel dimensionally- decomposed router for on-chip communication in 3D archi tectures,” In *Proceedings of the 34th annual international symposium on Computer architecture* (ISCA '07), 2007, pp. 138-149.

[10] Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, and J. Yang, “A low-radix and low-diameter 3D interconnection network design,” *2009 IEEE 15th International Symposium on High Performance Computer Architecture*, Raleigh, NC, 2009, pp. 30-42.

[11] S. Satpathy *et al.*, “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,” *2012 IEEE International Solid-State Circuits Conference*, San Francisco, CA, 2012, pp. 478-480.

[12] K. Sewell, R. Dreslinski et al., “Swizzle-switch networks for many-core systems,” in *IEEE Journal on emerging and Selected topics in circuits and systems* , vol. 2, June 2012

[13] N. W. McKeown, Scheduling algorithms for input-queued cell switches, PhD Thesis, 1992.

[14] W. Dally and B. Towles, Principles and Practices of Inter- connection Networks. Morgan Kaufmann, 2003.

[15] S. Satpathy, R. Das, R. Dreslinski, D. Sylvester, T. Mudge, and D. Blaauw, "High radix self-arbitrating switch fabric with multiple arbitration schemes and quality of service," in *Proceedings of the 49th annual Design Automation Conference* (DAC '12), 2012, pp. 746-749.

[16] "3D Integration: New Opportunities for Speed, Power and Performance," www.tezzaron.com/about/papers/Tezzaron- Presentation-CASS-020712-dist-2.pdf.

[17] R. Das, O. Mutlu, T. Moscibroda and C. R. Das, "Application-aware prioritization mechanisms for on-chip networks," *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, New York, NY, 2009, pp. 280-291.

[18] H. Patil, R. Cohn, M. Charney, R. Kapoor, A. Sun and A. Karunanidhi, "Pinpointing Representative Portions of Large Intel ® Itanium ® Programs with Dynamic Instrumentation," *Microarchitecture, 2004. MICRO-37 2004. 37th International Symposium on*, Portland, OR, USA, 2004, pp. 81-92.

[19] J. L. Henning, "Spec cpu2006 benchmark descriptions," in *ACM SIGARCH Computer Architecture News 34.4*, 2006.

[20] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," in *ACM SIGARCH Computer Architecture News 35.2*, 2006

[21] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology- Driven, Highly-Scalable Dragonfly Topology," in *35th International Symposium on Computer Architecture (ISCA)*, 2008.

[22] J. Kim, W. J. Dally, B. Towles, and A. K. Gupta, "Microarchitecture of a high-radix router," in *32nd International Symposium on Computer Architecture (ISCA)*, 2005.

[23] S. Scott, D. Abts, J. Kim, and W. J. Dally, "The black widow high-radix clos network," in *33rd International Symposium on Computer Architecture (ISCA)*, 2006.

[24] D. L. Lewis, S. Yalamanchili and H. H. S. Lee, "High Performance Non-blocking Switch Design in 3D Die-Stacking Technology," *2009 IEEE Computer Society Annual Symposium on VLSI,*Tampa,FL,2009,pp.25-30.

[25] N. McKeown, A. Mekkittikul, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch," in *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1260-1267, Aug 1999.

[26] M. M. Lee, J. Kim, D. Abts, M. Marty and J. W. Lee, "Probabilistic Distance-Based Arbitration: Providing Equality of Service for Many-Core CMPs," *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, Atlanta, GA, 2010, pp. 509-519.

[27] D. Abts and D. Weisser, "Age-based packet arbitration in large-radix k-ary n-cubes," *Supercomputing, 2007. SC '07. Proceedings of the 2007 ACM/IEEE Conference on*, Reno, NV, USA, 2007, pp. 1-11

[28] H. J. Chao, C. H. Lam and X. Guo, "A fast arbitration scheme for terabit packet switches," *Global Telecommunications Conference, 1999. GLOBECOM '99*, Rio de Janeireo, 1999, pp. 1236-1243 vol.2.

[29] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High-speed switch scheduling for local-area networks," in *ACM Transactions on Computer Systems (TOCS)*, 1993, pp. 319-352

[30] D. N. Serpanos and P. I. Antoniadis, "FIRM: a class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues," *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Tel Aviv, 2000, pp. 548-555 vol.2.

[31] N. McKeown, "The islip scheduling algorithm for input-queued switches," *Networking, IEEE/ACM Trans. on*, 1999, pp. 188-201 vol.2.

[32] J. A. Jum, S. H. Byun, B. J. Ahn, S. Y. Nam, and D. K. Sung, "A two-dimensional scalable crossbar matrix switch architec ture," in *IEEE International Conf. on Communications*, 2003.

[33] A. Agarwal, S. Hsu, S. Mathew, M. Anders, H. Kaul, F. Sheikh, and R. Krishnamurthy, "A 128x128b high-speed wide-and match-line content addressable memory in 32nm CMOS," in *Proc. 2011 ESSCIRC*, 2011, pp. 83-86.

[34] G. Burda, Y. Kolla, J. Dieffenderfer, and F. Hamdan, "A 45nm CMOS 13-port 64-word 41b fully associative content-addressable register file," in *IEEE ISSCC Dig. Tech. Papers*, 2010, pp. 286-287.

[35] K. Nii, T. Amano, N. Watanabe, M. Yamawaki, K. Yoshinaga, M. Wada, and I. Hayashi, "A 28nm 400MHz 4-parallel 1.6 Gsearch/s 80Mb ternary CAM," in *IEEE ISSCC Dig. Tech. Papers*, 2014, pp. 240-241.

[36] K. Pagiamtzis and A. Sheikholeslami, "Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712-727, Mar. 2006

[37] A. T. Do, C. Yin, K. S. Yeo, and T. T. H. Kim, "Design of a power-efficient CAM using automated background checking scheme for small match line swing," in *Proc. 2013 ESSCIRC*, 2013, pp. 209-212.

[38] C.-C. Wang, C.-H. Hsu, C.-C. Huang, and J.-H. Wu, "A self-disabled sensing technique for content-addressable memories," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 57, no. 1, pp. 31–35, Jan. 2010.

[39] B.-D. Yang, Y.-K. Lee, S.-W. Sung, J.-J. Min, J.-M. Oh, and H.-J. Kang, "A Low Power Content Addressable Memory Using Low Swing Search Lines," *IEEE Trans. on Circuits and Systems I: Regular Papers*, vol. 58, no. 12, pp. 2849-2858, Dec. 2011.

[40] C.-C. Wang, J.-S. Wang, and C. Yeh, “High-speed and low-power design techniques for TCAM macros,” *IEEE J. Solid State Circuits*, vol. 43, no. 2, pp. 530–540, Feb. 2008.

[41] R. Ranica, N. Planes, O. Weber, O. Thomas, S. Haendler, D. Noblet, D. Croain, C. Gardin, and F. Arnaud, "FDSOI process/design full solutions for ultra low leakage, high speed and low voltage SRAMs," in *2013 Symp. VLSI Technology (VLSIT)*, 2013, pp. T210-T211.

[42] E. Karl, Z. Guo, J. W. Conary, J. L. Miller, Y.-G. Ng, S. Nalam, D. Kim, J. Keane, U. Bhattacharya, and K. Zhang, "A 0.6 V 1.5 GHz 84Mb SRAM design in 14nm FinFET CMOS technology," in *IEEE ISSCC Dig. Tech. Papers*, 2015, pp. 1-3.

[43] J. Li, R. K. Montoye, M. Ishii, and L. Chang, “1 Mb 0.41 µm2 2T-2R cell nonvolatile TCAM with two-bit encoding and clocked self-referenced sensing,” *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 896–907, Apr. 2014.

[44] I. Arsovski, T. Chandler, and A. Sheikholeslami, “A ternary content addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme,” *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 155–158, Jan. 2003.

[45] S. Jeloka, N. Akesh, D. Sylvester, and D. Blaauw, "A configurable TCAM/BCAM/SRAM using 28nm push-rule 6T bit cell," in *2015 Symp. VLSI Circuits (VLSIC)*, 2015, pp. C272-C273.

[46] P. Jain, G. E. Suh, and S. Devadas, "Embedded intelligent SRAM," in *Proc. 40th annual Design Automation Conference*, 2003, pp. 869-874.

[47] D. PattersonT, N. C. Anderson, R. Fromm, K. Keeton, C. Kozyrakis, R. Tomas, and K. Yelick, "A Case for Intelligent DRAM: IRAM," *IEEE Micro*, pp. 33-44, April 1997.

[48] K. Mai, T. Paaske, N. Jayasena, R. Ho, W. J. Dally, and M. Horowitz, “Smart memories: A modular reconfigurable architecture,” in *Proc. 27th Int. Symp. Computer Architecture(ISCA)*, 2000, pp. 161–171.

[49] K. Mai, R. Ho, E. Alon, D. Liu, Y. Kim, D. Patil, and M. Horowitz, "Architecture and circuit techniques for a reconfigurable memory block," in *IEEE ISSCC Dig. Tech. Papers*, 2004, pp. 500-501.

[50] I. Hayashi, T. Amano, N. Watanabe, Y. Yano, Y. Kuroda, M. Shirata, K. Dosaka, K. Nii, H. Noda, and H. Kawai, “A 250-MHz 18-Mb full ternary CAM with low-voltage matchline sensing scheme in 65-nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2671–2680, Nov. 2013

[51] S. Matsunaga, J. Hayakawa, S. Ikeda, K. Miura, T. Endoh, H. Ohno, and T. Hanyu, "Mtj-based nonvolatile logic-in-memory circuit, future prospects and issues," in *Proc. Conference on Design, Automation and Test in Europe(DATE)*, 2009, pp. 433-435.

[52] Q. Zhu, B. Akin, H. E. Sumbul, F. Sadi, J. C. Hoe, L. Pileggi, and F. Franchetti, "A 3D-stacked logic-in-memory accelerator for application-specific data intensive computing," in *IEEE 3D Systems Integration Conference (3DIC)*, 2013, pp. 1-7.

[53] I. Arsovski, T. Hebig, D. Dobson, and R. Wisort, "A 32 nm 0.58-fJ/ Bit/Search 1-GHz ternary content addressable memory compiler using silicon-aware early-predict late-correct sensing with embedded deep-trench capacitor noise mitigation," *IEEE J. Solid-State Circuits*, vol. 48, no. 4, pp. 932–939, Apr. 2013.

[54] P.-T. Huang and W. Hwang, "A 65 nm 0.165 fJ/Bit/search 256 144 TCAM macro design for IPv6 lookup tables," *IEEE J. Solid-State Circuits*, vol. 46, no. 2, pp. 507–519, Feb. 2011.

[55] S. G. Narendra, L. C. Fujino, and K. Smith, "Through the Looking Glass? The 2015 Edition: Trends in Solid-State Circuits from ISSCC," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 1, pp. 14-24,Winter 2015.

[56] N. Pinckney *et al.*, "Assessing the performance limits of parallelized near-threshold computing," *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, San Francisco, CA, 2012, pp. 1143-1148.

[57] "ISSCC 2016 Trends," http://isscc.org/doc/2016/ISSCC2016_TechTrends.pdf

[58] K. C. Smith, A. Wang and L. C. Fujino, "Through the Looking Glass: Trend Tracking for ISSCC 2012," in *IEEE Solid-State Circuits Magazine*, vol. 4, no. 1, pp. 4-20, winter 2012.

[59] Qazi, M. Clinton, S. Bartling, and A. P. Chandrakasan, "A low voltage 1 Mb FRAM in 0.13 um CMOS featuring time-to-digital sensing for expanded operating margin," *IEEE J. Solid-State Circuits*, vol. 47, no. 1, pp. 141–150, Jan. 2012.

[60] A. Sheikholeslami and P. G. Gilak, "A survey of circuit innovations in ferroelectric random-access memories," *Proc. IEEE*, vol. 88, pp. 667–689, 2000.

[61] J. Rodriguez et al., "Reliability of Ferroelectric Random Access memory embedded within 130nm CMOS," *Reliability Physics Symposium (IRPS)*, 2010 IEEE International , vol., no., pp.750-758, 2-6 May 2010

[62] S. K. Mathew, S. K. Satpathy, M. A. Anders, H. Kaul, S. K. Hsu, A. Agarwal, G. K. Chen, R. J. Parker, R. K. Krishnamurthy, and V. De, "A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm CMOS," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014 IEEE International, 2014, pp. 278–279.

[63] R. Pappu, "Physical One-Way Functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, Sep. 2002

[64] Myoung-Kyu Seo *et al.*, "A 130-nm 0.9-V 66-MHz 8-Mb (256K × 32) local SONOS embedded flash EEPROM," in *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 877-883, April 2005.

[65] N. Derhacobian, S. C. Hollmer, N. Gilbert and M. N. Kozicki, "Power and Energy Perspectives of Nonvolatile Memory Technologies," in *Proceedings of the IEEE*, vol. 98, no. 2, pp. 283-298, Feb. 2010.

[66] S. Jeloka, N. B. Akesh, D. Sylvester and D. Blaauw, "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T Bit Cell Enabling Logic-in-Memory," in *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 1009-1021, April 2016.

[67] S. Jeloka, R. Das, R. G. Dreslinski, T. Mudge and D. Blaauw, "Hi-Rise: A High-Radix Switch for 3D Integration with Single-Cycle Arbitration," *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, Cambridge, 2014, pp. 471-483.

[68] S. Rao, S. Jeloka, R. Das, D. Blaauw, R. Dreslinski and T. Mudge, "VIX: Virtual Input Crossbar for efficient switch allocation," *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, CA, 2014, pp. 1-6.

[69] N. Abeyratne, S. Jeloka, Y. Kang, D. Blaauw, R. Dreslinski, R. Das and T. Mudge, "Quality-of-Service for a High-Radix Switch," *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, CA, 2014, pp. 1-6.

[70] N. Pinckney, L. Shifren, B. Cline, S. Sinha, S. Jeloka, R. Dreslinski, T. Mudge, D. Sylvester, and D. Blaauw, "Near-Threshold Computing in FinFET Technologies: Opportunities for Improved Voltage Scalability," *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, TX, 2016, pp. 1-6.

[71] S.Aga, S.Jeloka, A.Subramaniyan, S. Narayanswamy, D. Blaauw, R.Das, "Compute Caches for Efficient Very Large Vector Processing", *High Performance Computer Architecture,* 2017 (to appear).

[72] S. Rosenblatt et al., "Field Tolerant Dynamic Intrinsic Chip ID Using 32 nm High-K/Metal Gate SOI Embedded DRAM," in *IEEE Journal of Solid-State Circuits*, vol. 48, no. 4, pp. 940-947, April 2013.

[73] K. Yang, Q. Dong, D. Blaauw and D. Sylvester, "14.2 A physically unclonable function with BER $<10^{-8}$ for robust chip authentication using oscillator collapse in 40nm CMOS," *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, San Francisco, CA, 2015, pp. 1-3.

[74] M. F. Chang and S. J. Shen, "A Process Variation Tolerant Embedded Split-Gate Flash Memory Using Pre-Stable Current Sensing Scheme," in *IEEE Journal of Solid-State Circuits*, vol. 44, no. 3, pp. 987-994, March 2009.

[75] R. Sundaram *et al.*, "A 128 Mb NOR flash memory with 3 MB/s program time and low-power write performance by using in-package inductor charge-pump," *ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference, 2005.*, San Francisco, CA, 2005, pp. 50-584 Vol. 1.

[76] H. Mitani *et al.*, "7.6 A 90nm embedded 1T-MONOS flash macro for automotive applications with 0.07mJ/8kB rewrite energy and endurance over 100M cycles under Tj of 175°C," *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2016, pp. 140-141.

[77] K. Ramkumar *et al.*, "A scalable, low voltage, low cost SONOS memory technology for embedded NVM applications," *2013 5th IEEE International Memory Workshop*, Monterey, CA, 2013, pp. 199-202.

[78] M. Seok, G. Kim, D. Blaauw and D. Sylvester, "A Portable 2-Transistor Picowatt Temperature-Compensated Voltage Reference Operating at 0.5 V," in *IEEE Journal of Solid-State Circuits*, vol. 47, no. 10, pp. 2534-2545, Oct. 2012.

[79] Y. Taito *et al.*, "7.3 A 28nm embedded SG-MONOS flash macro for automotive achieving 200MHz read operation and 2.0MB/S write throughput at Ti, of 170°C," *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, San Francisco, CA, 2015, pp. 1-3.