ISCAS 2020
2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020
CAS

Towards a Reconfigurable Bit-Serial/Bit-Parallel Vector Accelerator Using In-Situ Processing-in-SRAM

Khalid Al-Hawaj, *Olalekan Afuye, Shady Agwa, Alyssa Apsel, Christopher Batten*

Cornell University
Electrical and Computer Engineering

2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

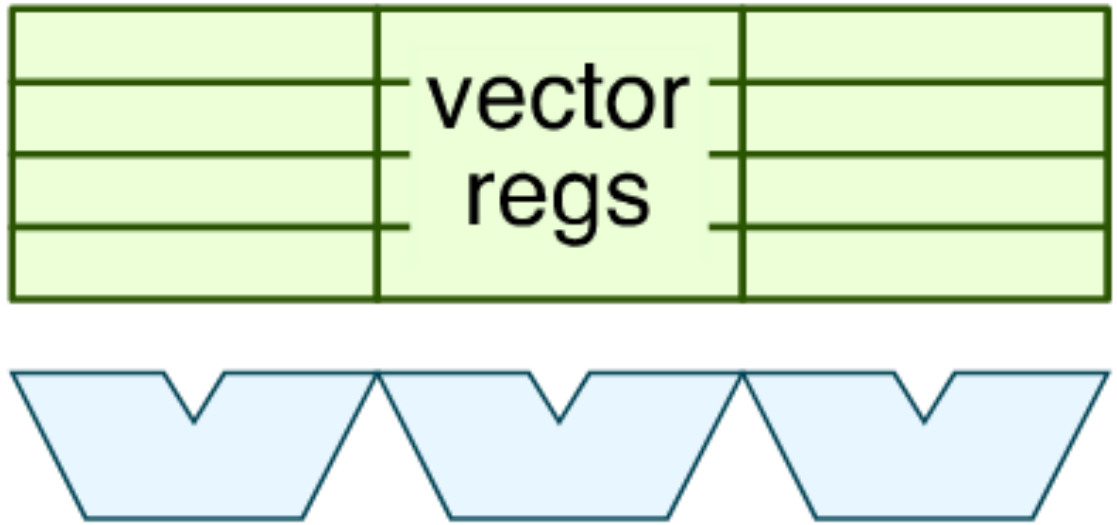
TOWARDS A RECONFIGURABLE BIT-SERIAL/BIT-PARALLEL VECTOR ACCELERATOR USING IN-SITU PROCESSING-IN-SRAM

Khalid Al-Hawaj, Olalekan Afuye, Shady Agwa, Alyssa Apsel, Christopher Batten

Cornell University
Electrical and Computer Engineering

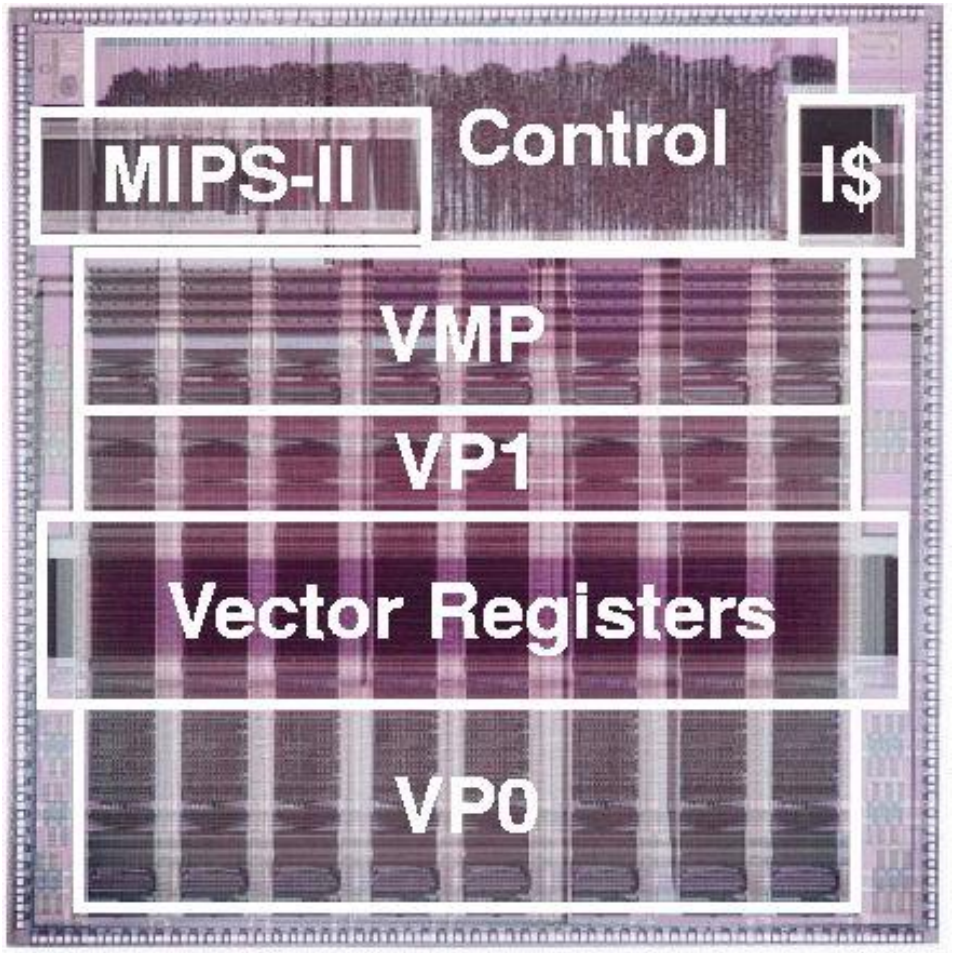
- There is a resurgence of interest in vector abstraction evident by recent ISA extensions (e.g., ARM SVE and RISC-V RVV).

- Vector machines leverage vector abstraction to increase performance in executing data-level parallel (DLP) workloads efficiently by exploiting regularity.



Disadvantages of Vector Machines

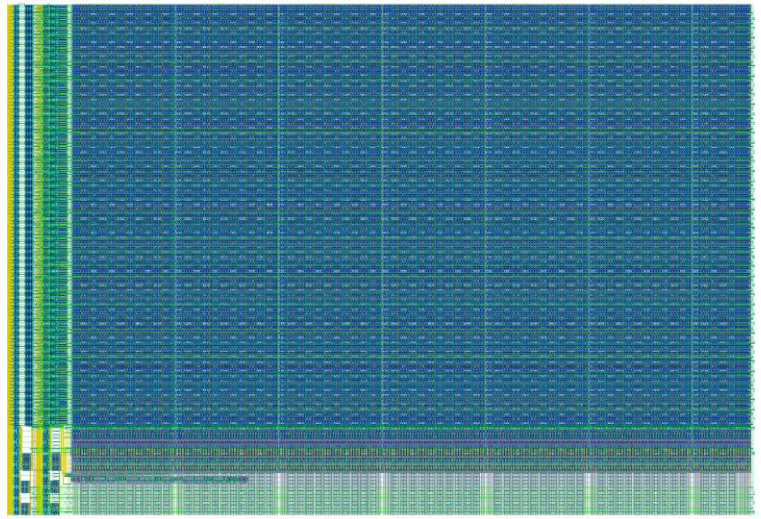
- Vector machines require highly expensive multi-ported state elements (i.e., register files) to feed vector arithmetic and logical unit (ALU).
- Recent work on in-situ processing-in-SRAM shows promise in fusing the vector register file with the ALU to enable efficient vector acceleration using bit-serial execution paradigm.*



Asanovic et al., "The T0 Vector Microprocessor," HotChips '95

* S. Jeloka et al., "A Configurable TCAM/BCAM/SRAM Using 28nm Push-Rule 6T Bit Cell", VLSIC '15.
* J. Wang at al. "A Compute SRAM with Bit-Serial Integer/Floating-Point Operations for Programmable In-Memory Vector Acceleration". ISSCC '19.

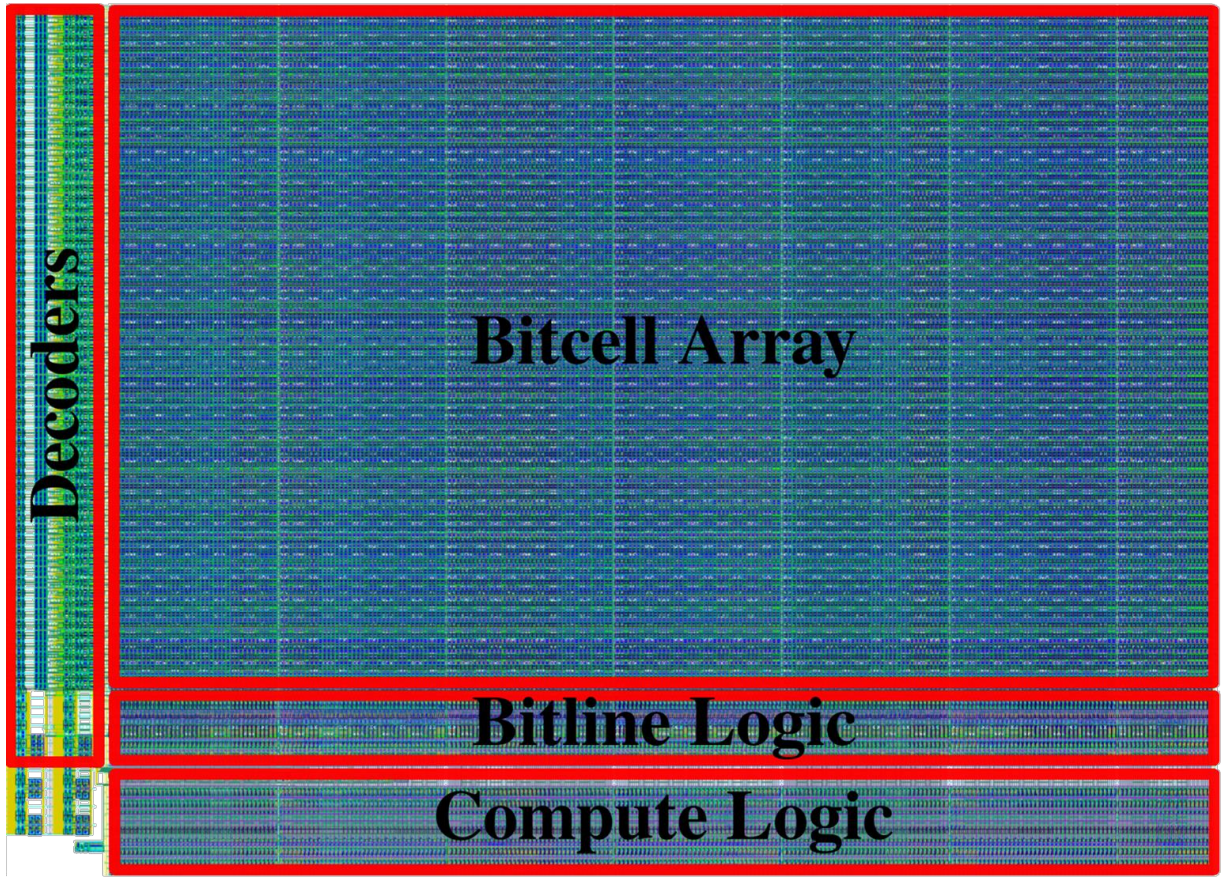
- **We propose vector RAM (VRAM) leveraging in-situ processing-in-SRAM to create vector accelerator in two different flavors: bit-serial vector RAM (BS-VRAM) and bit-parallel vector RAM (BP-VRAM).**



- **Main contributions:**
  1. Detailed circuit-level design of both BS-VRAM and BP-VRAM
  2. Implementation of 17 different macro-operations for BS-VRAM and BP-VRAM using micro-operation abstraction
  3. Detailed study of the trade-offs in area, cycle time, latency, throughput, and energy for BS-VRAM vs. BP-VRAM

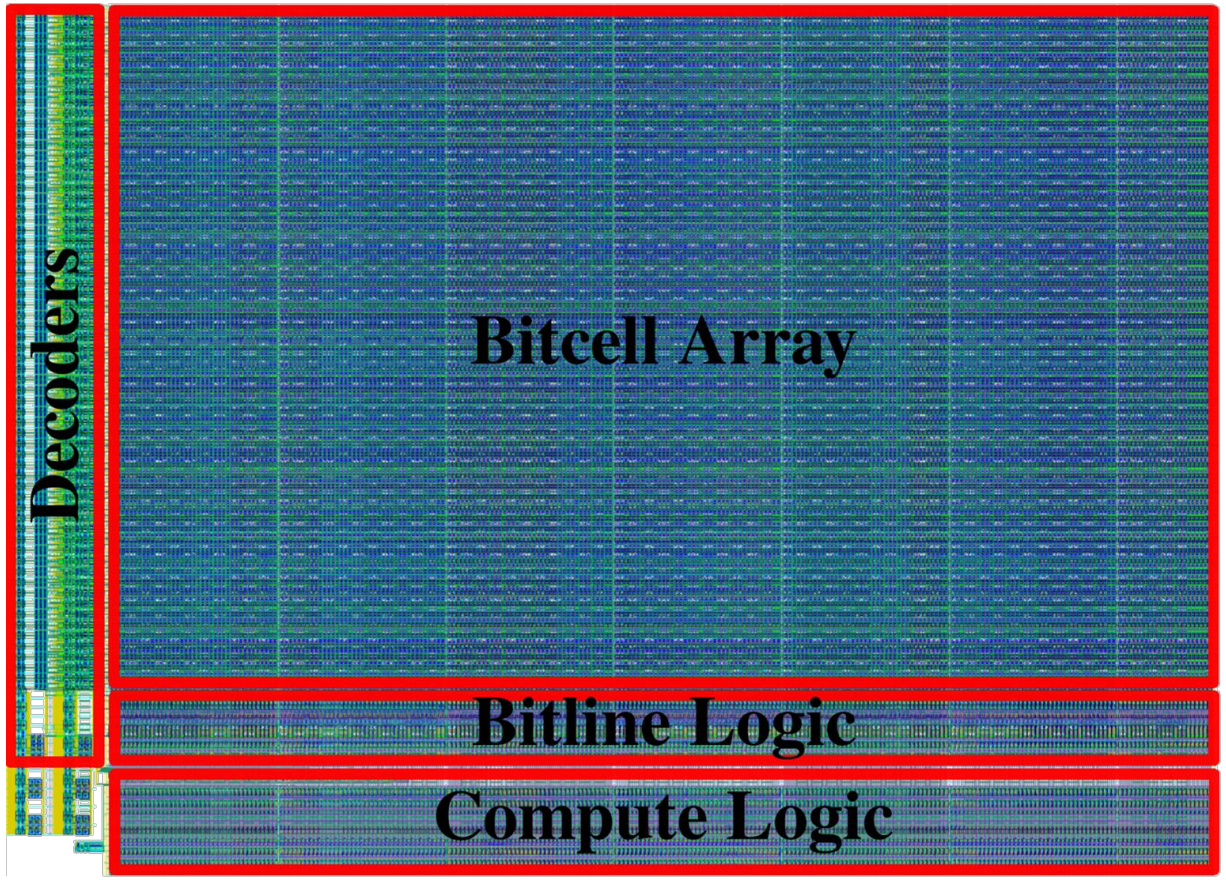
Outline

- **Motivation**
- **Background: Bit-line Compute**
- **Vector RAM**
  - VRAM Circuits
  - VRAM Micro-Programming
  - VRAM Macro-Programming
  - Evaluation
- **Conclusion**

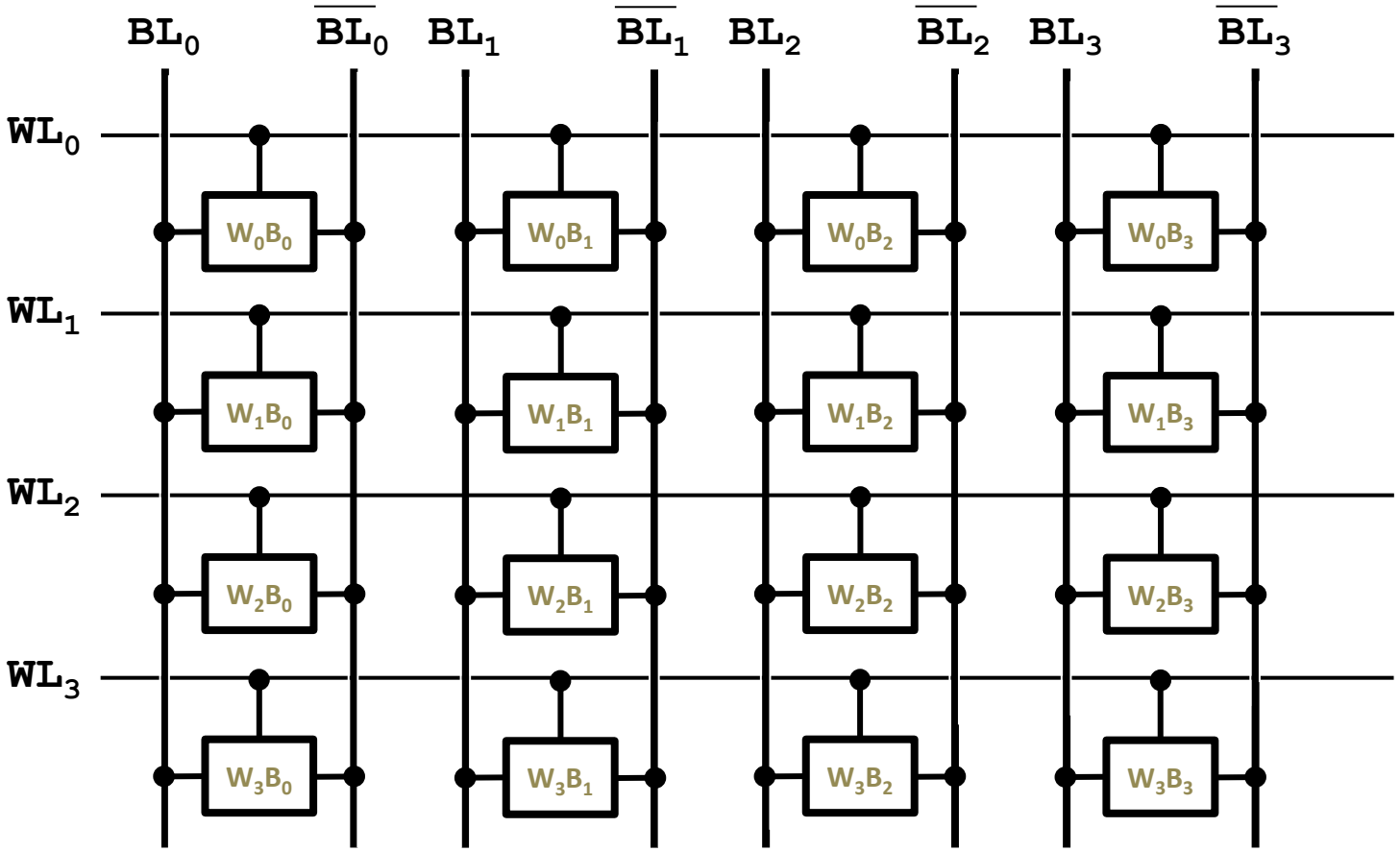


Outline

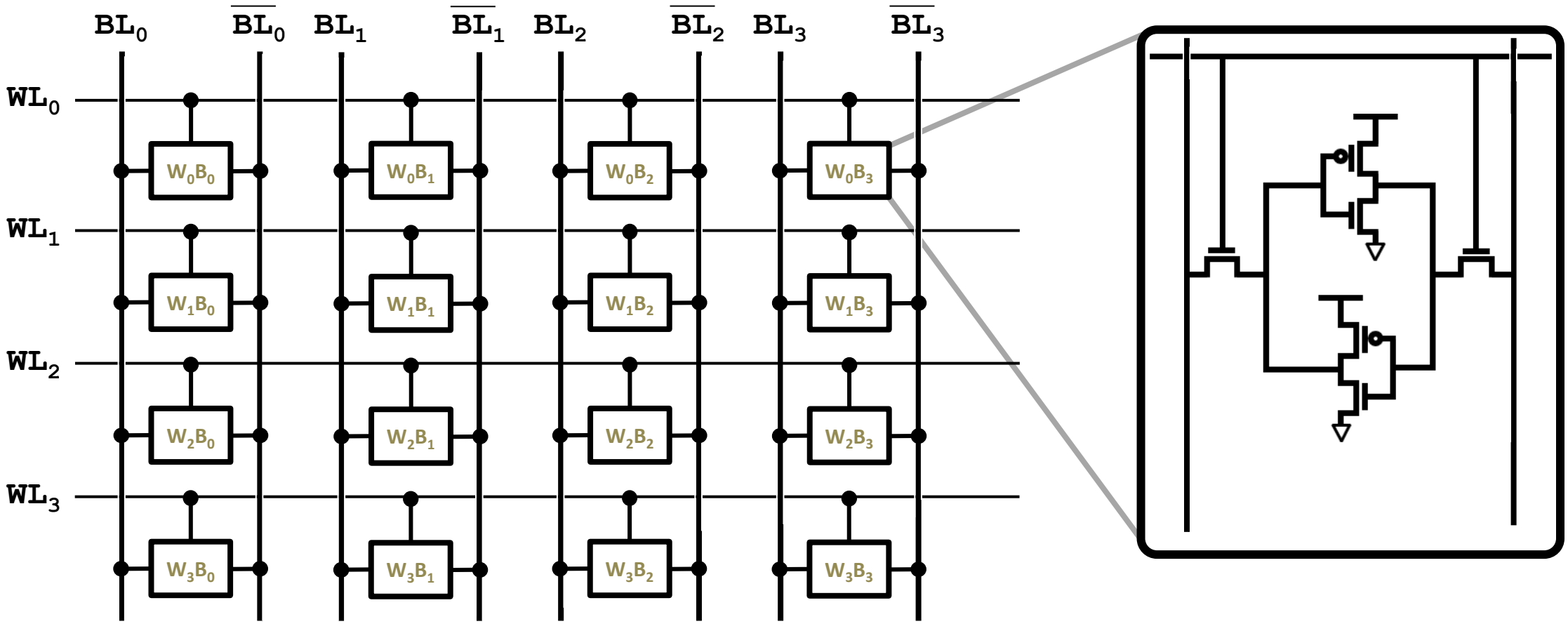
- Motivation
- **Background: Bit-line Compute**
- Vector RAM
  - VRAM Circuits
  - VRAM Micro-Programming
  - VRAM Macro-Programming
  - Evaluation
- Conclusion



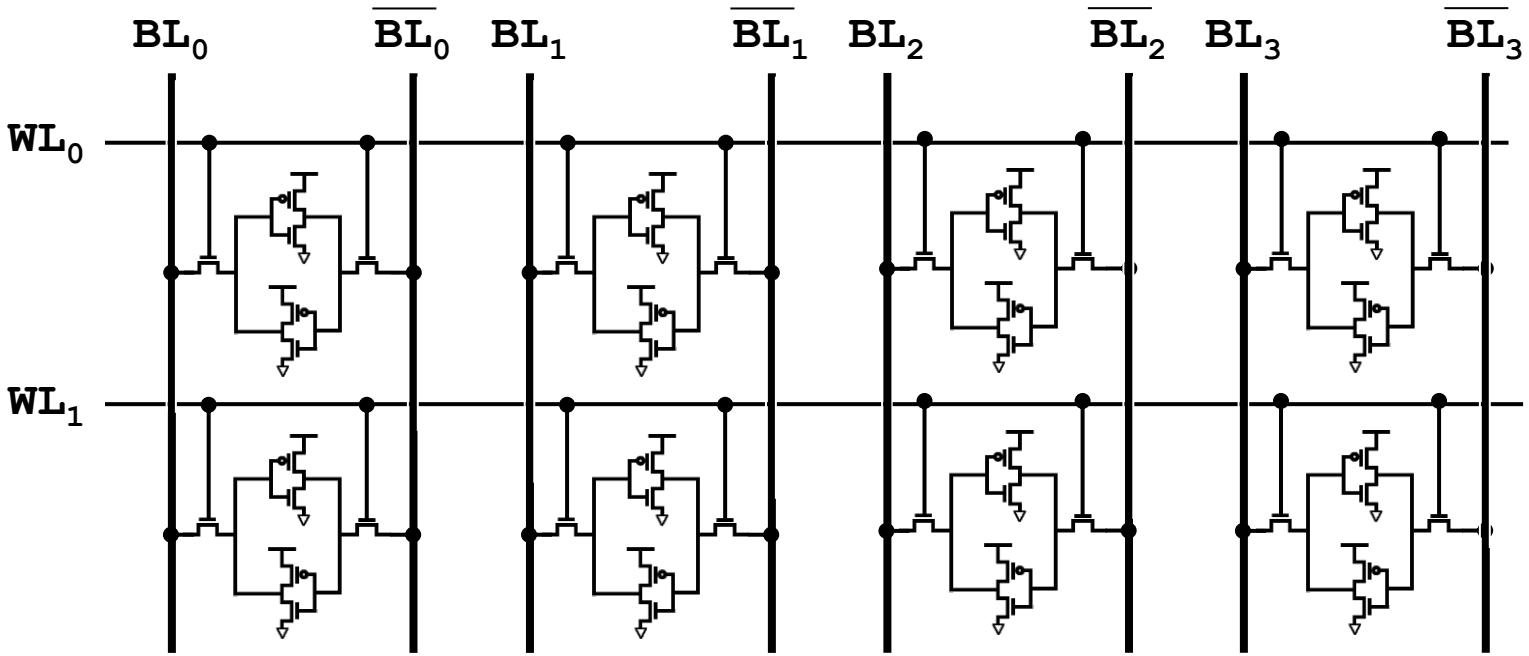
Background: Bit-line Compute



Background: Bit-line Compute

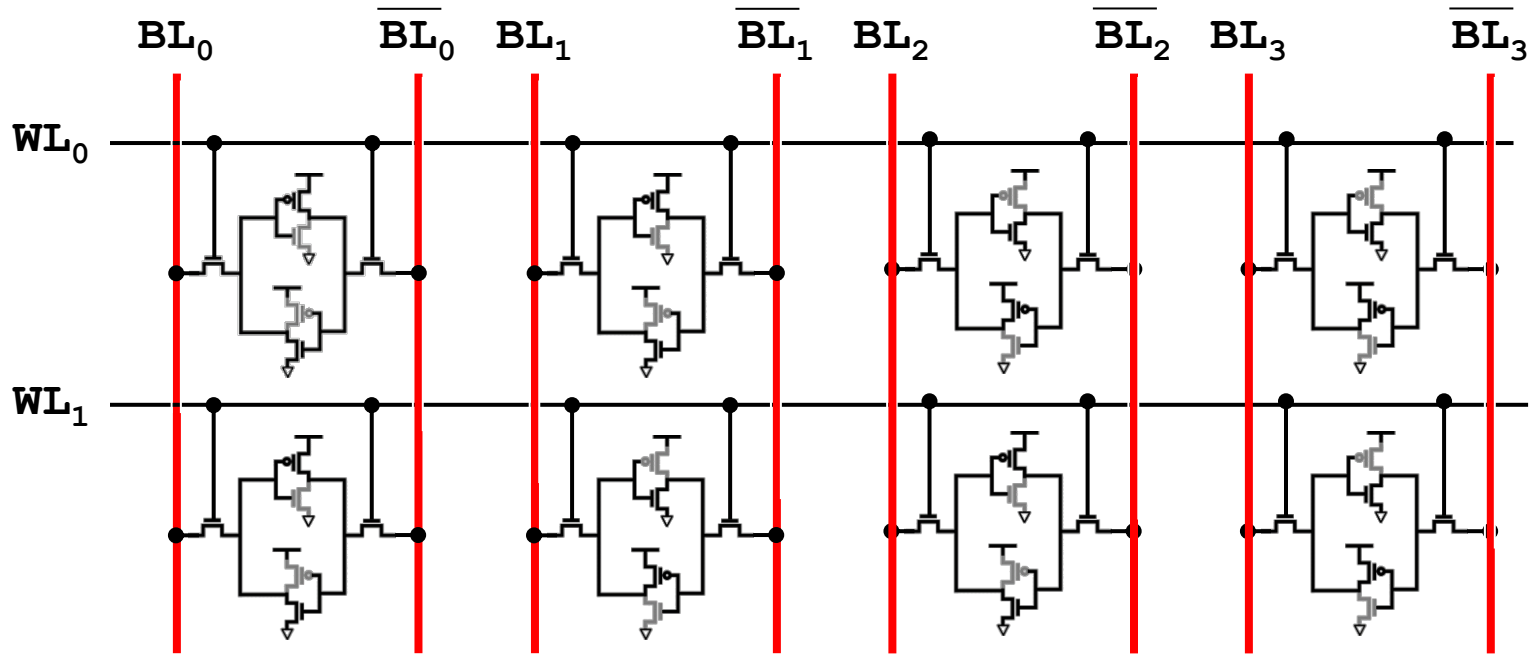


Background: Bit-line Compute



S. Jeloka et al., ”A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory.” IEEE Journal of Solid-State Circuits ‘16.

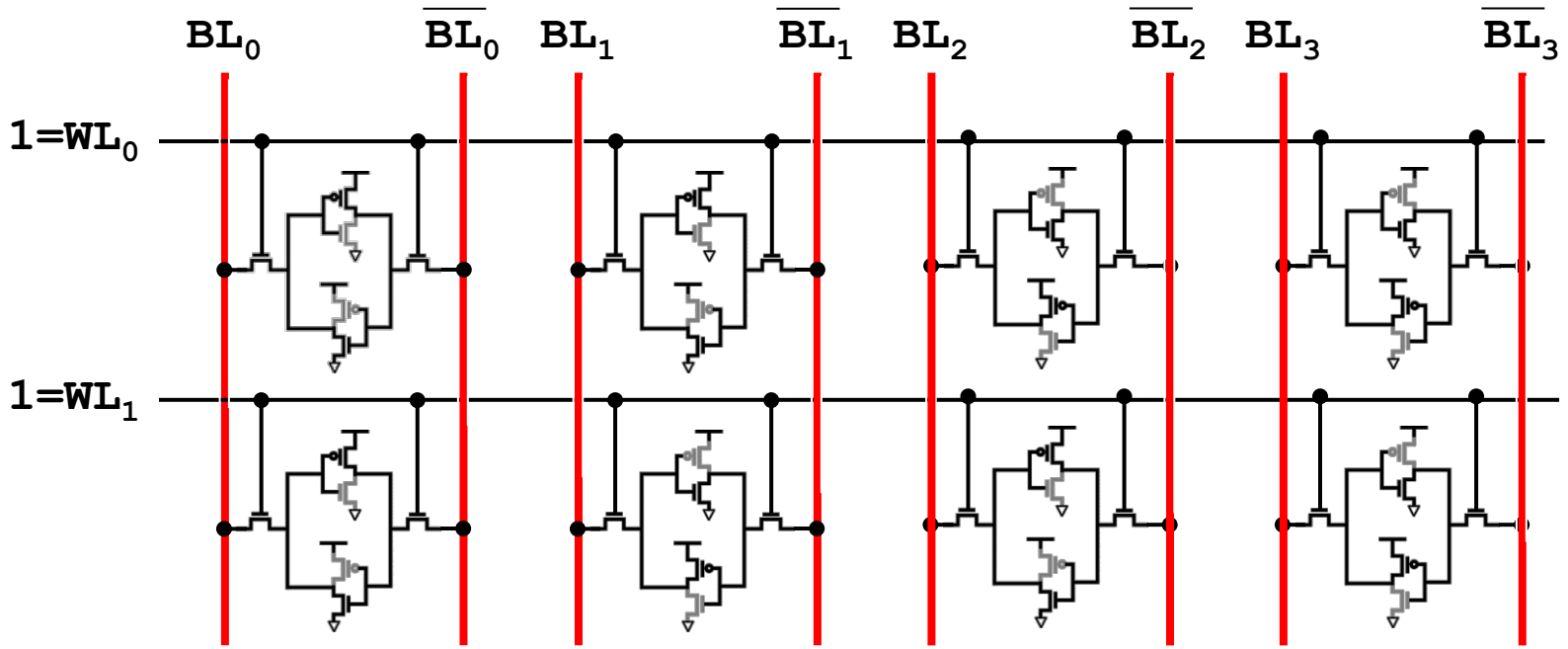
Background: Bit-line Compute



ROW 0	0		0		1		1
ROW 1	0		1		0		1
BL							
BL′							

S. Jeloka et al., ”A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory.” IEEE Journal of Solid-State Circuits ‘16.

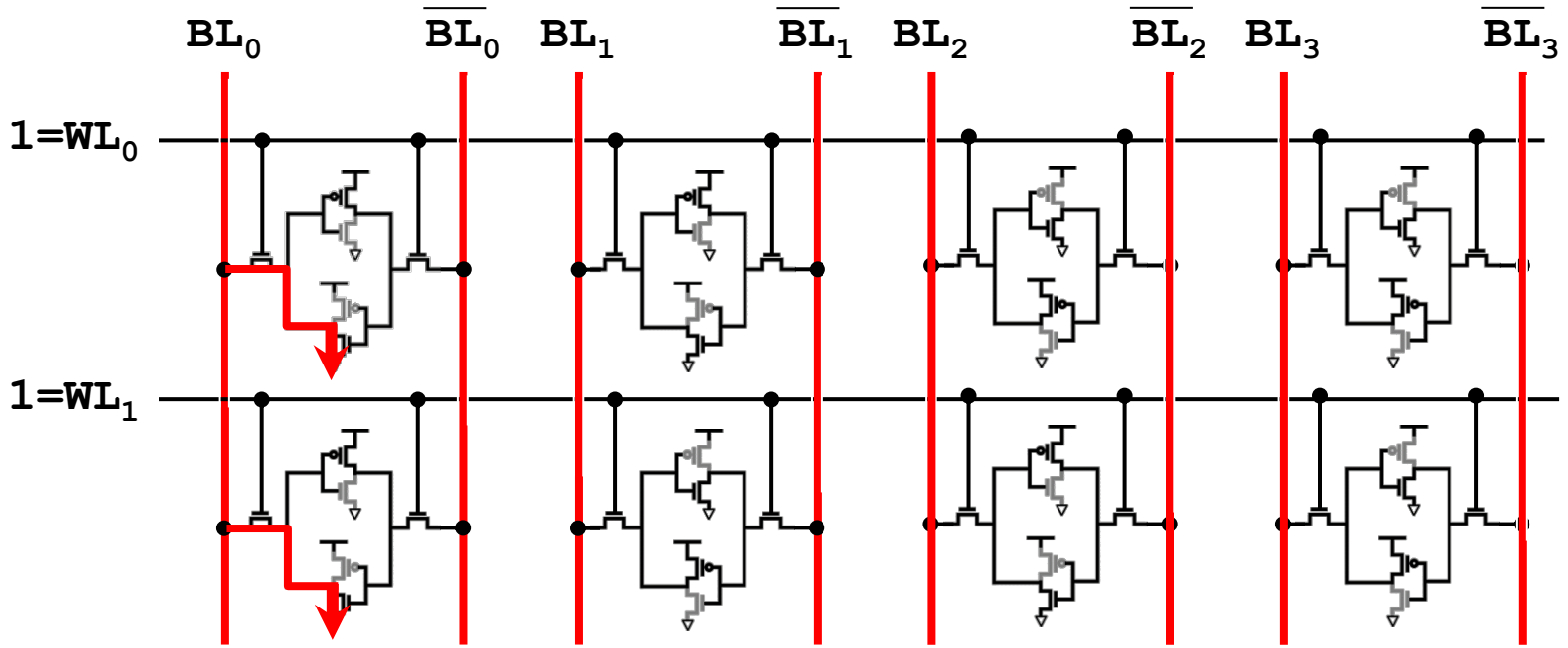
Background: Bit-line Compute



ROW 0	0		0		1		1
ROW 1	0		1		0		1
BL							
BL′							

S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.

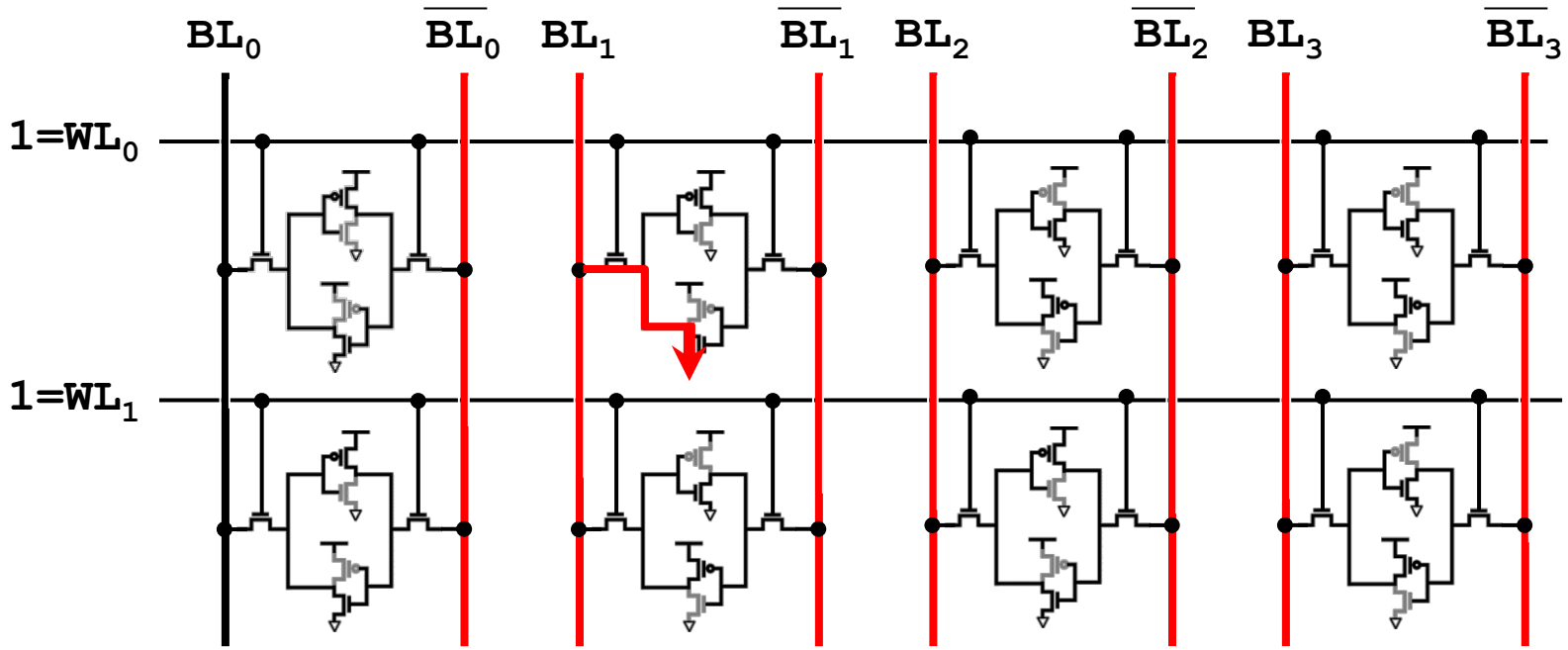
Background: Bit-line Compute



ROW 0	0		0		1		1
ROW 1	0		1		0		1
BL	0						
BL′							

S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.

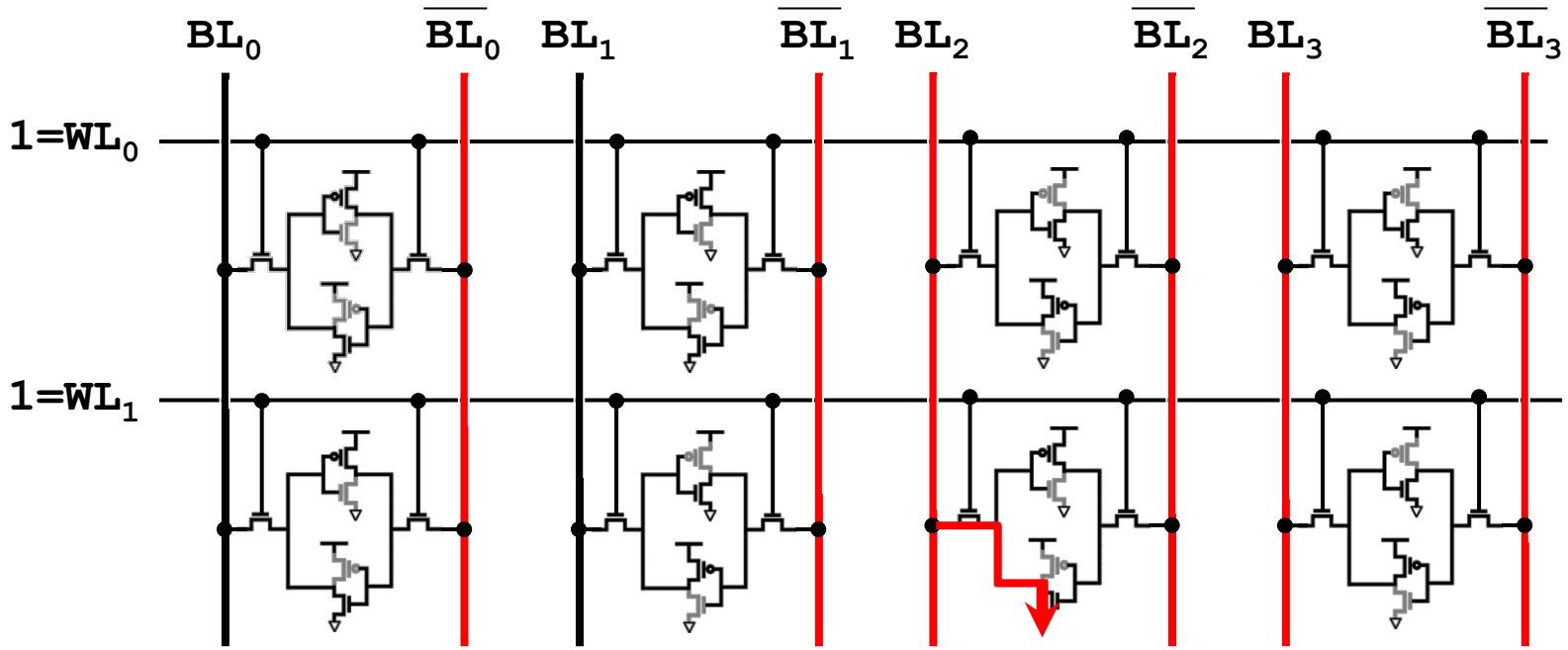
Background: Bit-line Compute



ROW 0	0		0		1		1
ROW 1	0		1		0		1
BL	0		0				
BL′							

S. Jeloka et al., ”A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory.” IEEE Journal of Solid-State Circuits ‘16.

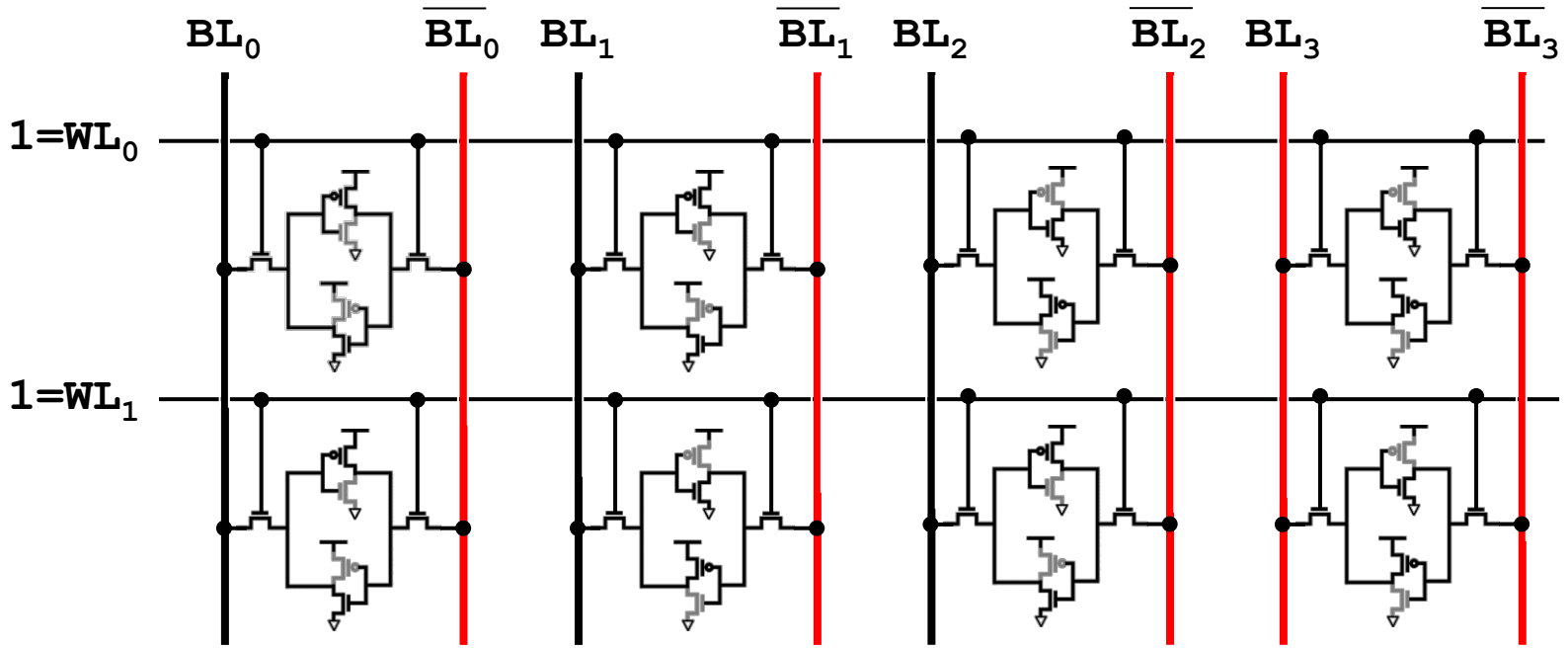
Background: Bit-line Compute



ROW 0	0		0		1	1
ROW 1	0		1		0	1
BL	0		0		0	
BL′						

S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.

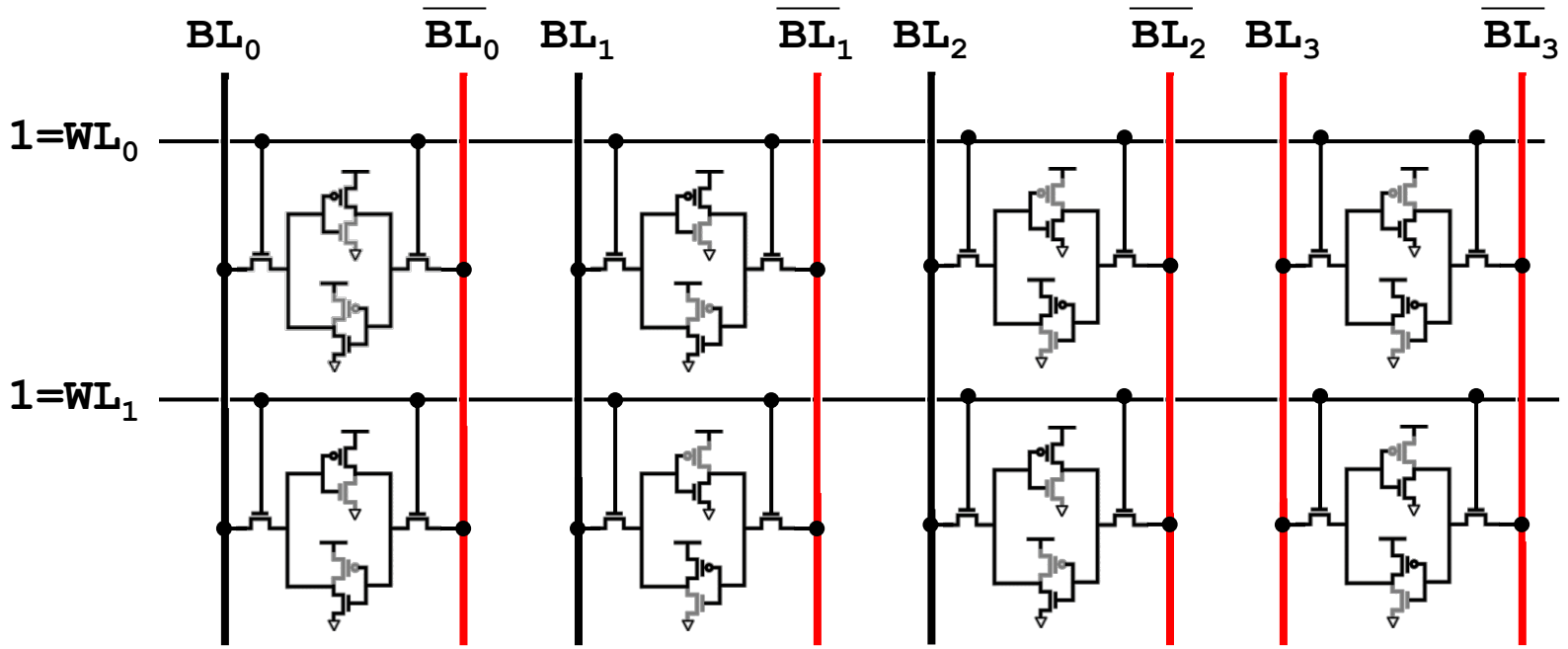
Background: Bit-line Compute



ROW 0	0		0		1	1
ROW 1	0		1		0	1
BL	0		0		0	1
BL′						

S. Jeloka et al., ”A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory.” IEEE Journal of Solid-State Circuits ‘16.

Background: Bit-line Compute

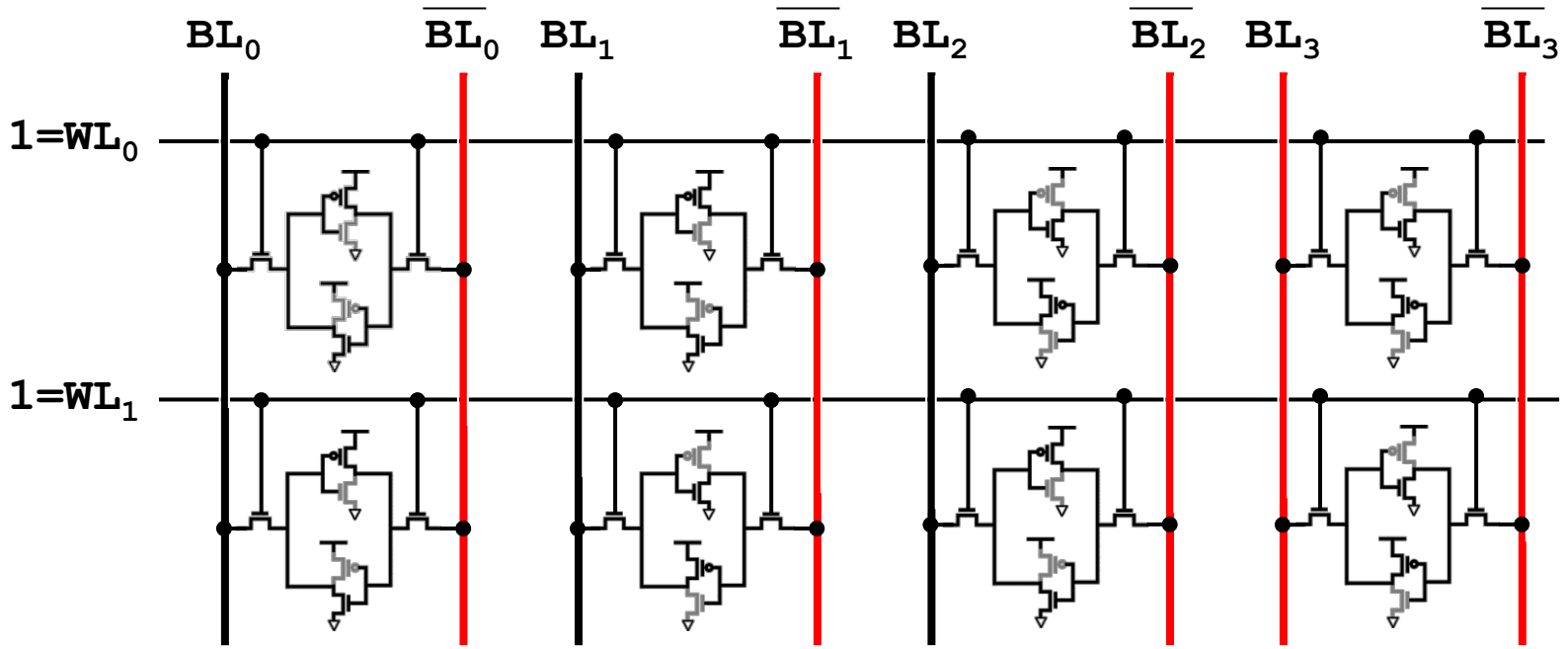


ROW 0	0		0		1		1
ROW 1	0		1		0		1
BL	0		0		0		1
BL′							

AND

S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.

Background: Bit-line Compute

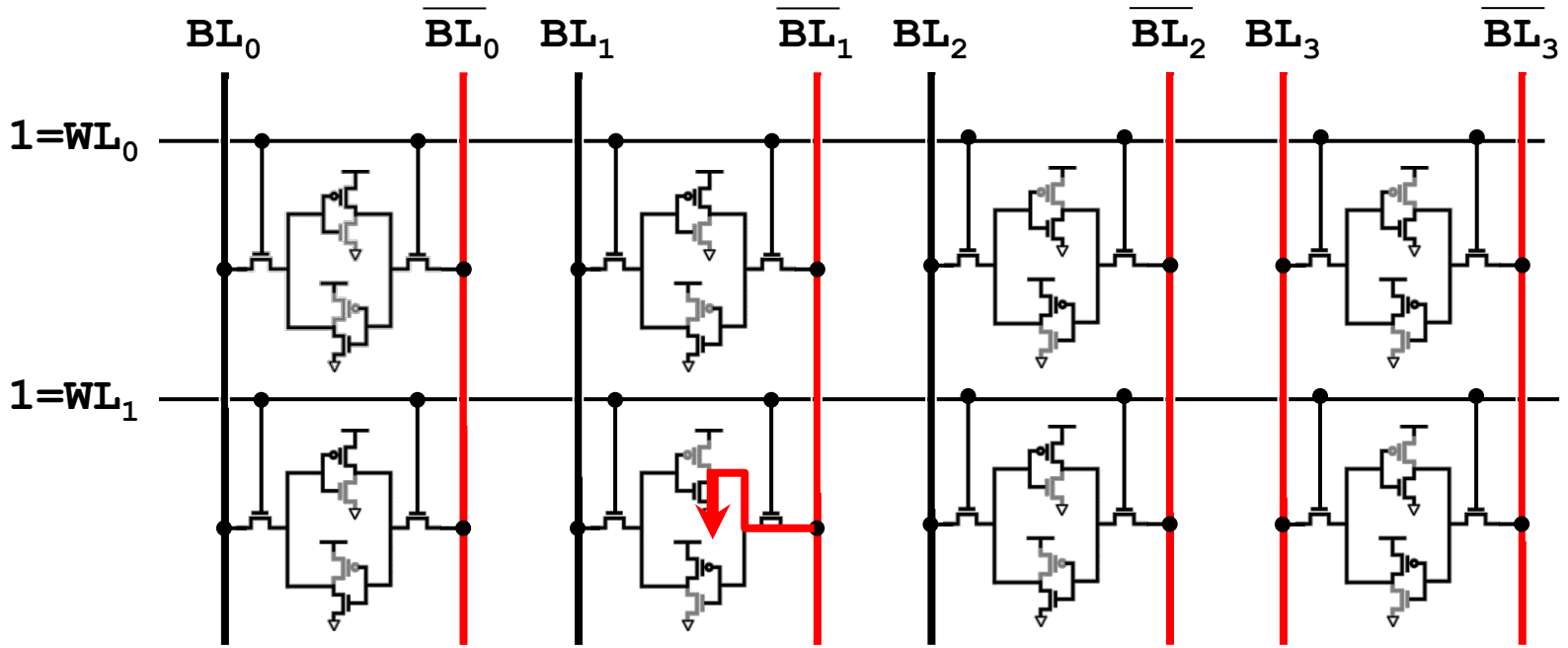


ROW 0	0		0		1	1
ROW 1	0		1		0	1
BL	0		0		0	1
BL′	1					

AND

S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.

Background: Bit-line Compute

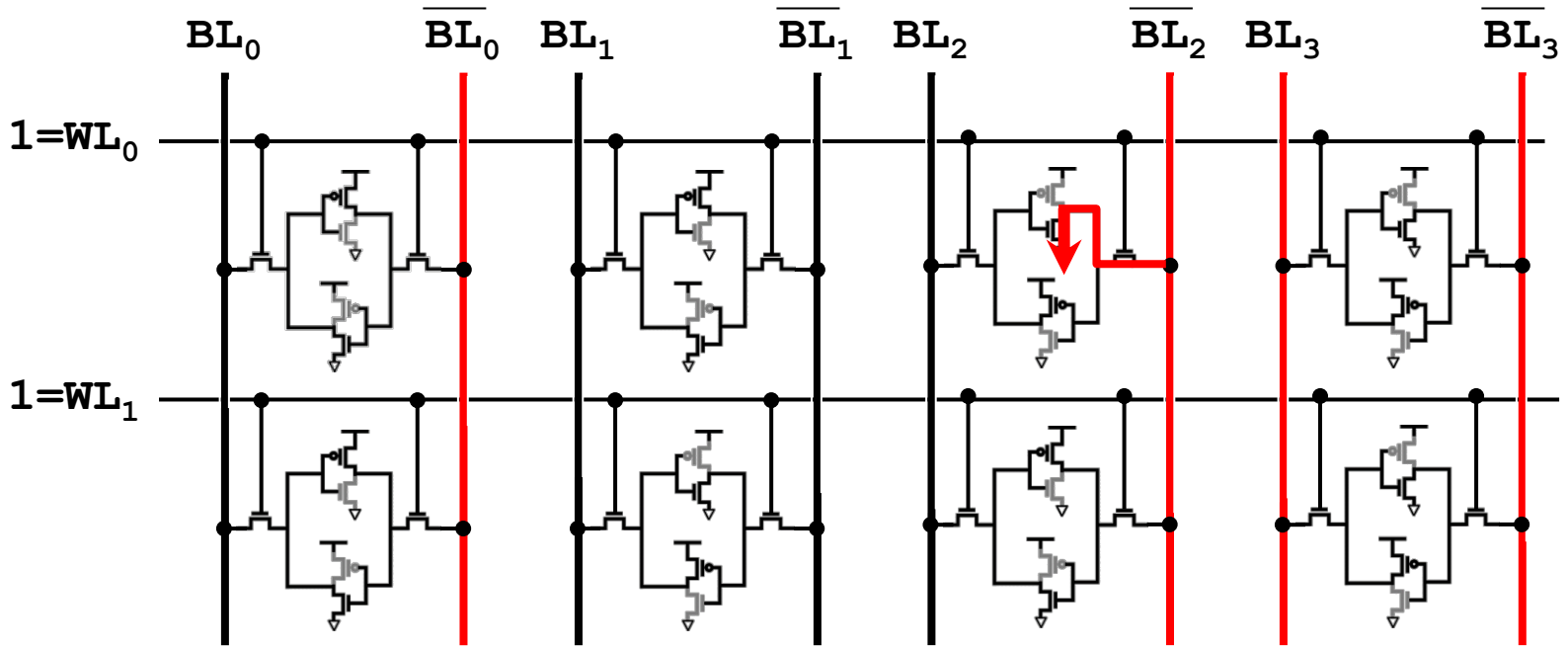


ROW 0	0		0		1	1
ROW 1	0		1		0	1
BL	0		0		0	1
BL′	1		0			

AND

S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.

Background: Bit-line Compute

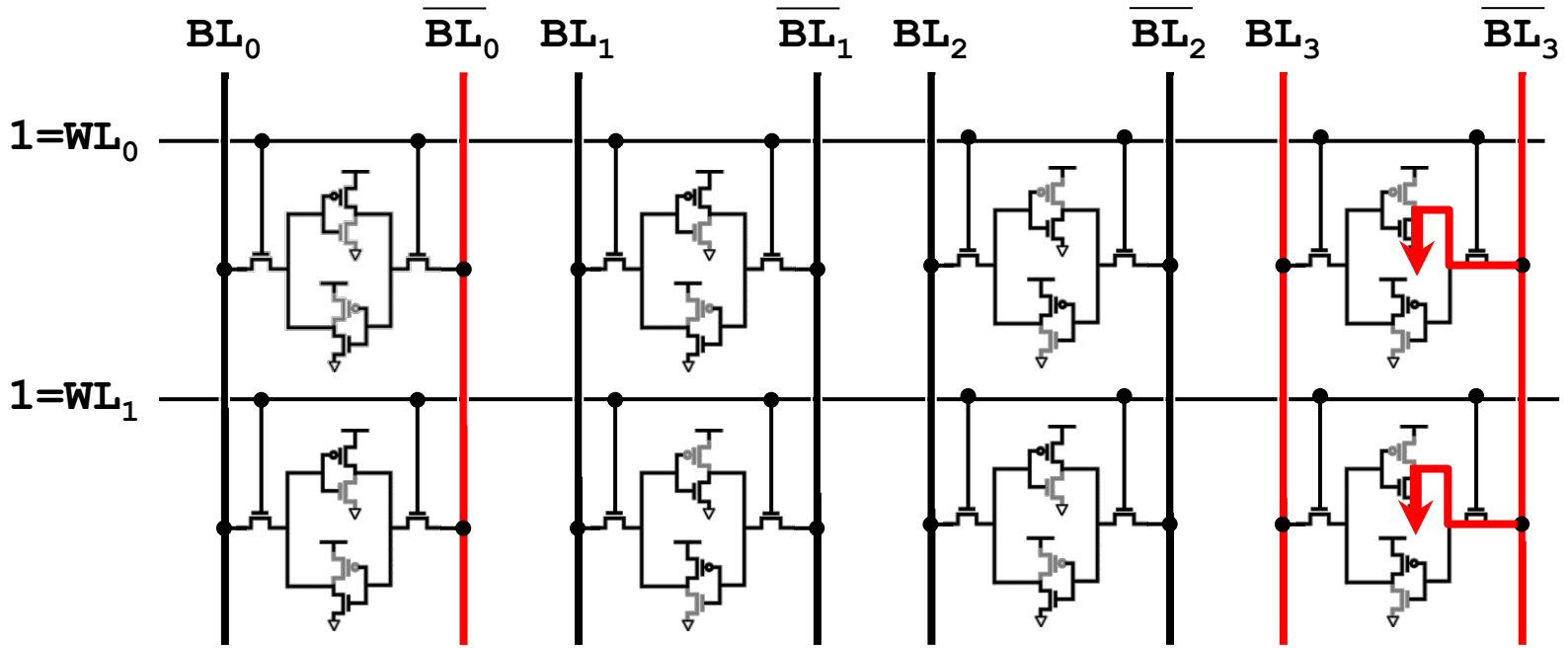


ROW 0	0		0		1	1
ROW 1	0		1		0	1
BL	0		0		0	1
BL′	1		0		0	

AND

S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.

Background: Bit-line Compute

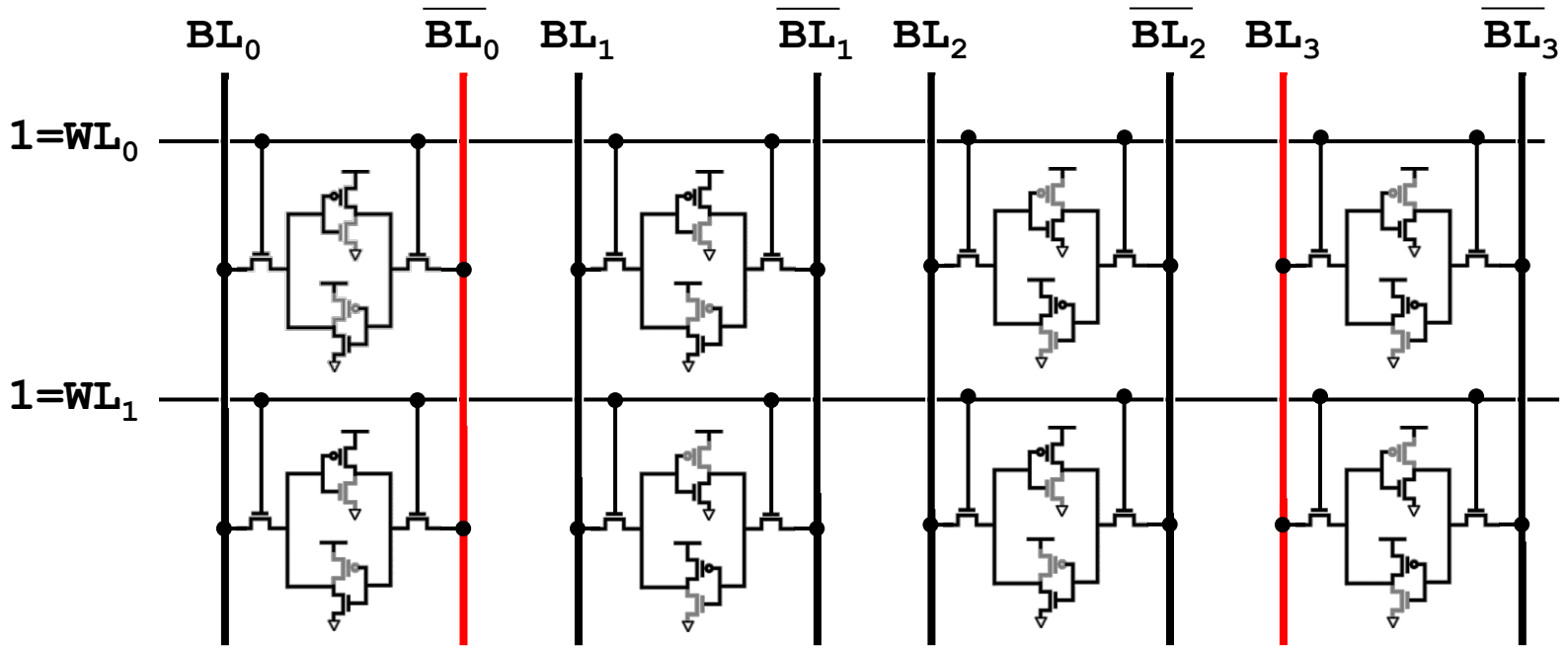


ROW 0	0		0		1	1
ROW 1	0		1		0	1
BL	0		0		0	1
BL′	1		0		0	0

AND

S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.

Background: Bit-line Compute



ROW 0	0		0		1		1
ROW 1	0		1		0		1
BL	0		0		0		1
BL′	1		0		0		0

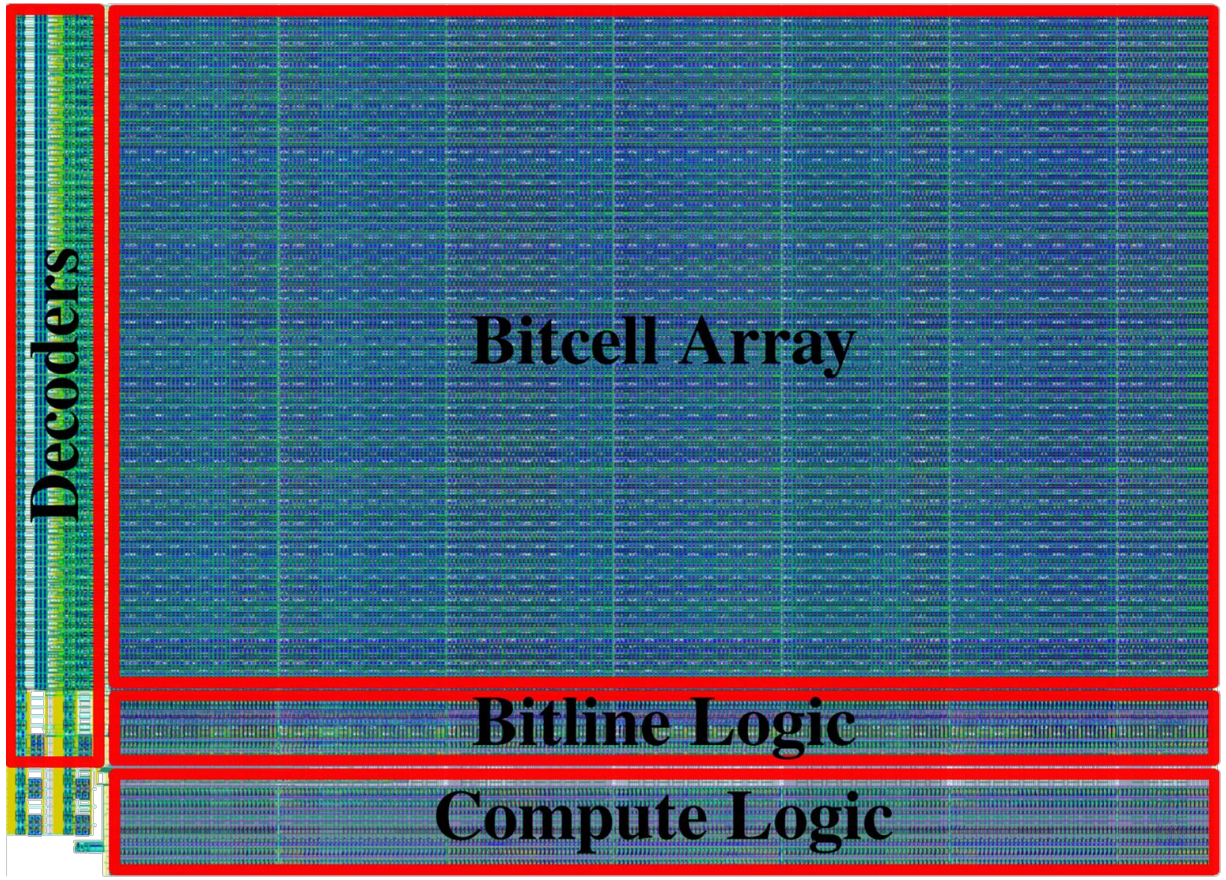
AND

NOR

S. Jeloka et al., ”A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory.” IEEE Journal of Solid-State Circuits ‘16.

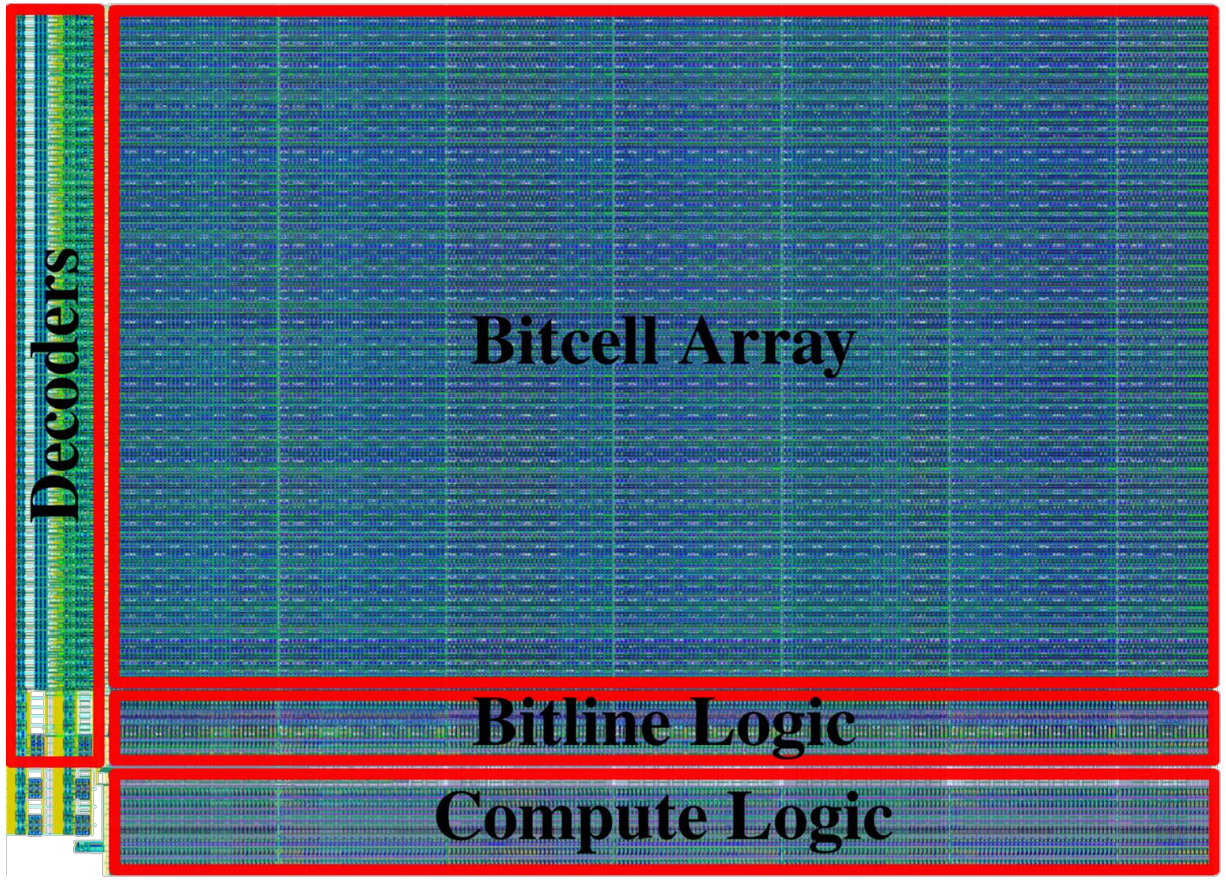
Outline

- Motivation
- Background: Bit-line Compute
- **Vector RAM**
  - VRAM Circuits
  - VRAM Micro-Programming
  - VRAM Macro-Programming
  - Evaluation
- Conclusion

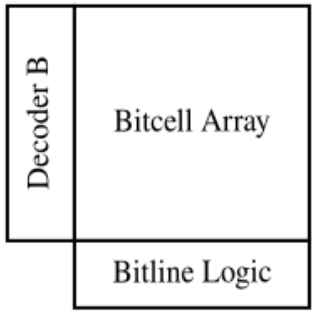


Outline

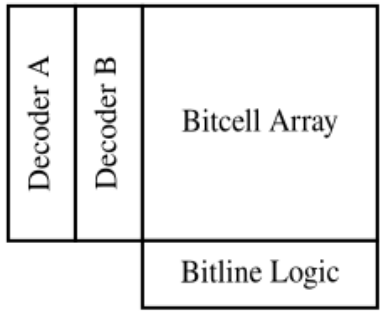
- Motivation
- Background: Bit-line Compute
- **Vector RAM**
  - **VRAM Circuits**
  - VRAM Micro-Programming
  - VRAM Macro-Programming
  - Evaluation
- Conclusion



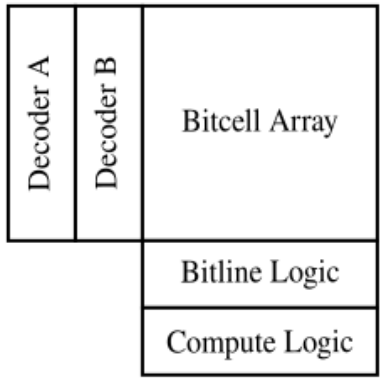
VRAM: Circuits



VRAM: Circuits

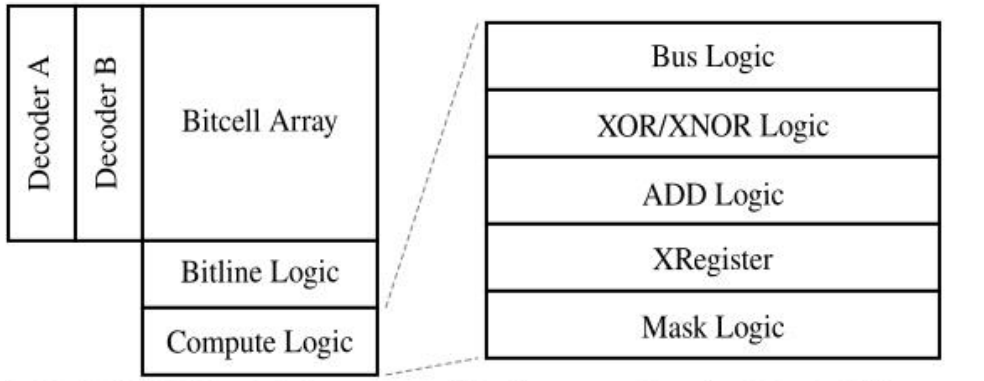


VRAM: Circuits



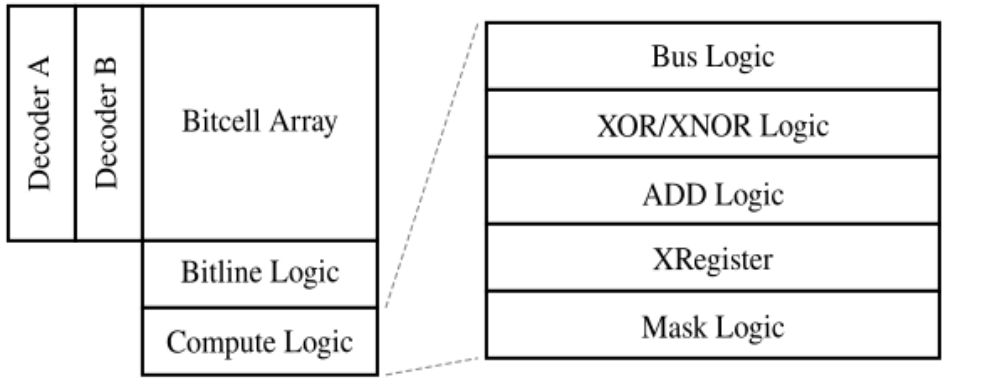
(a) VRAM Block Diagram

VRAM: Circuits



(a) VRAM Block Diagram (b) Compute Logic Block Diagram

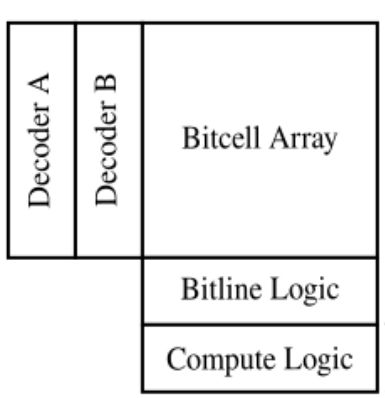
VRAM: Circuits—Bit-Serial Compute Logic



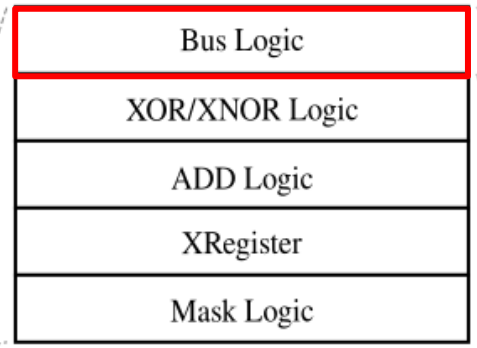
(a) VRAM Block Diagram (b) Compute Logic Block Diagram

(c) One Column of Bit-Serial Compute Logic

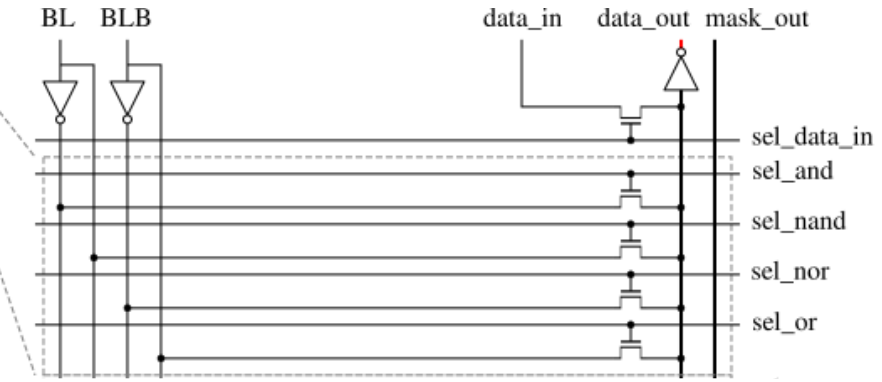
VRAM: Circuits—Bit-Serial Compute Logic



(a) VRAM Block Diagram

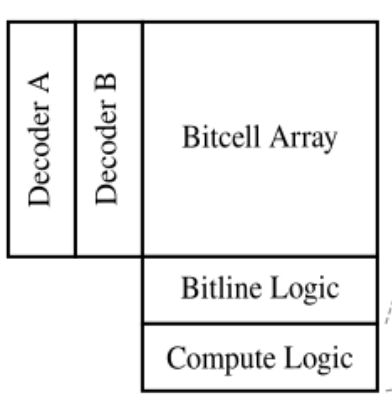


(b) Compute Logic Block Diagram

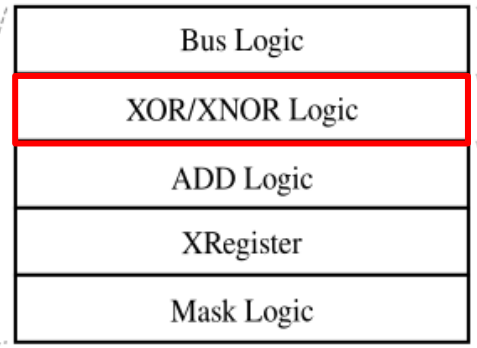


(c) One Column of Bit-Serial Compute Logic

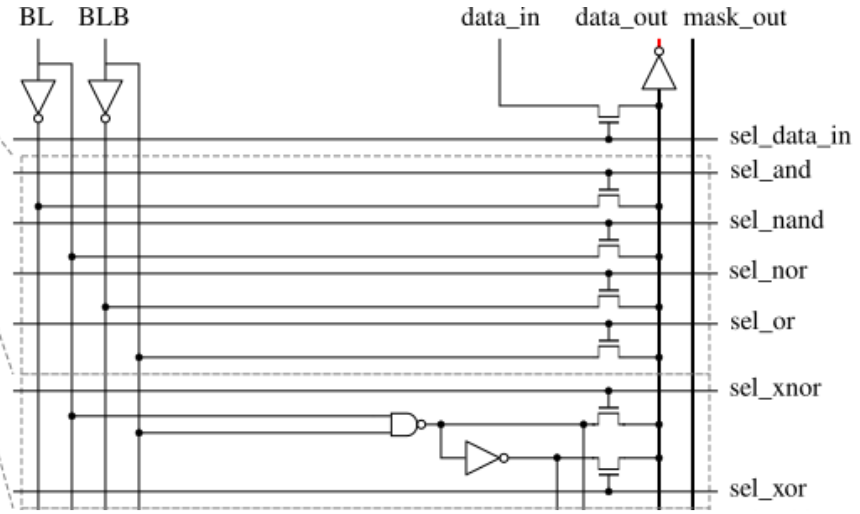
VRAM: Circuits—Bit-Serial Compute Logic



(a) VRAM Block Diagram

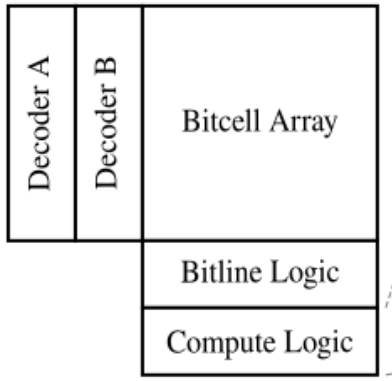


(b) Compute Logic Block Diagram

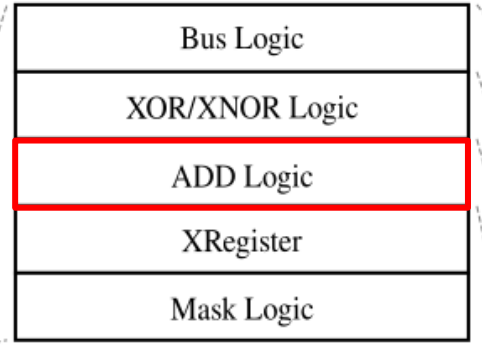


(c) One Column of Bit-Serial Compute Logic

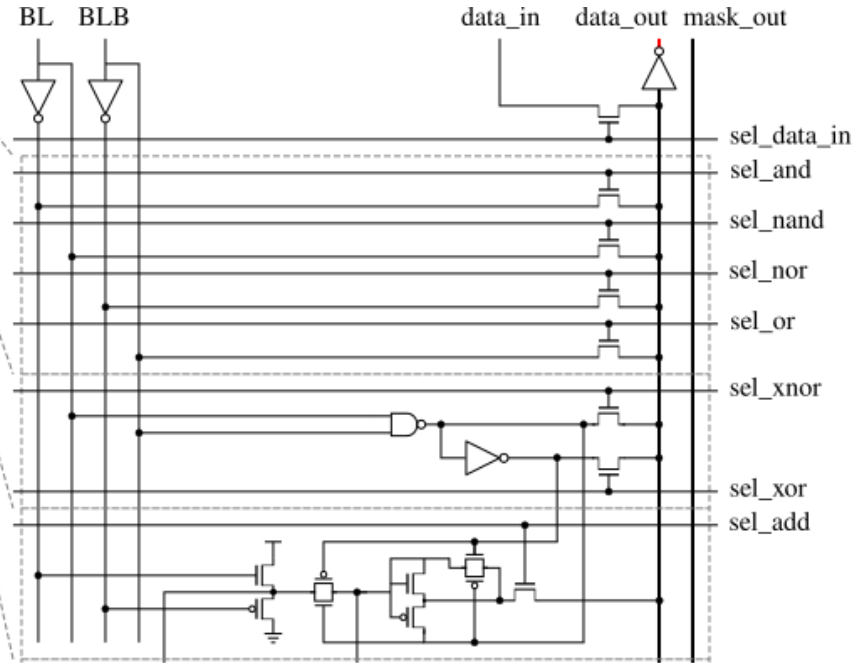
VRAM: Circuits—Bit-Serial Compute Logic



(a) VRAM Block Diagram

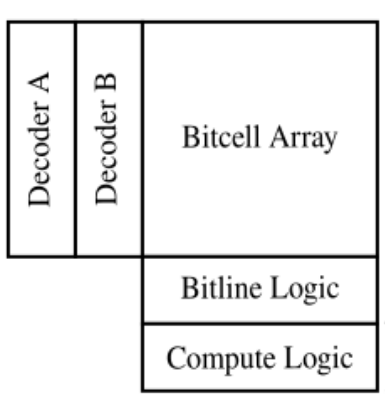


(b) Compute Logic Block Diagram

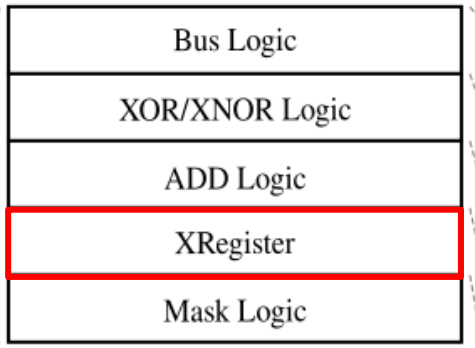


(c) One Column of Bit-Serial Compute Logic

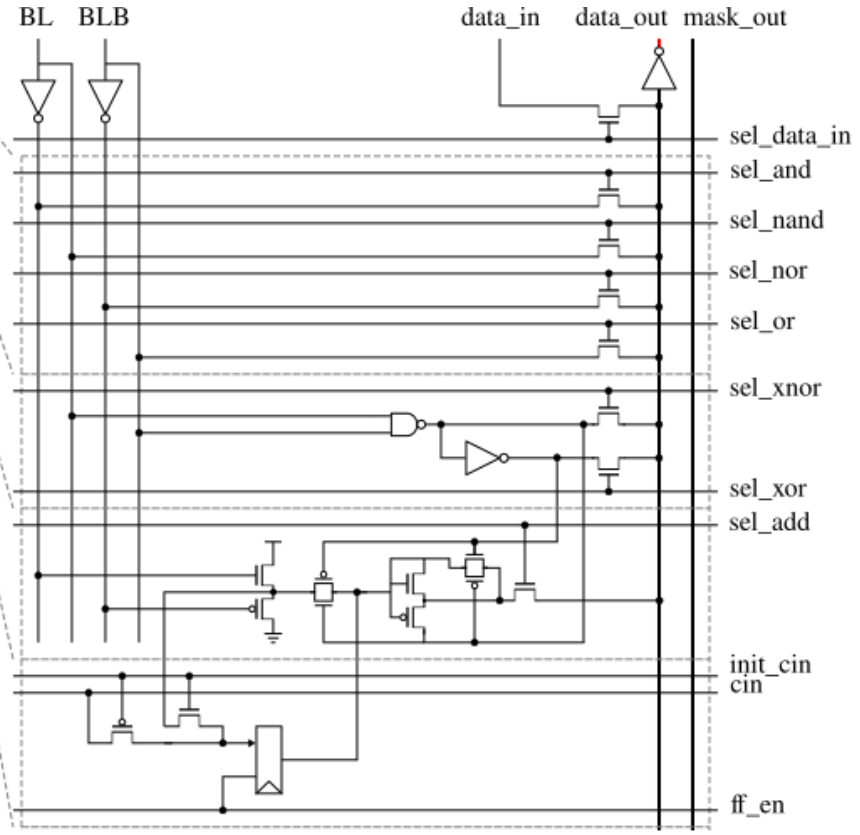
VRAM: Circuits—Bit-Serial Compute Logic



(a) VRAM Block Diagram

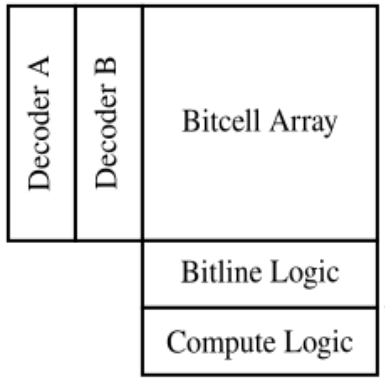


(b) Compute Logic Block Diagram

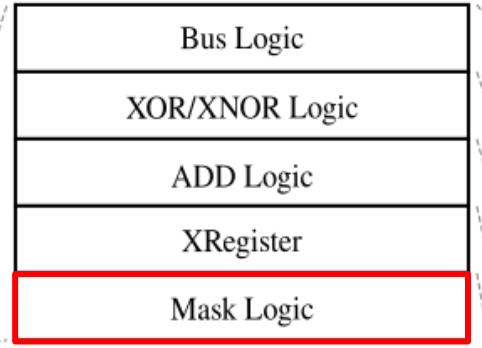


(c) One Column of Bit-Serial Compute Logic

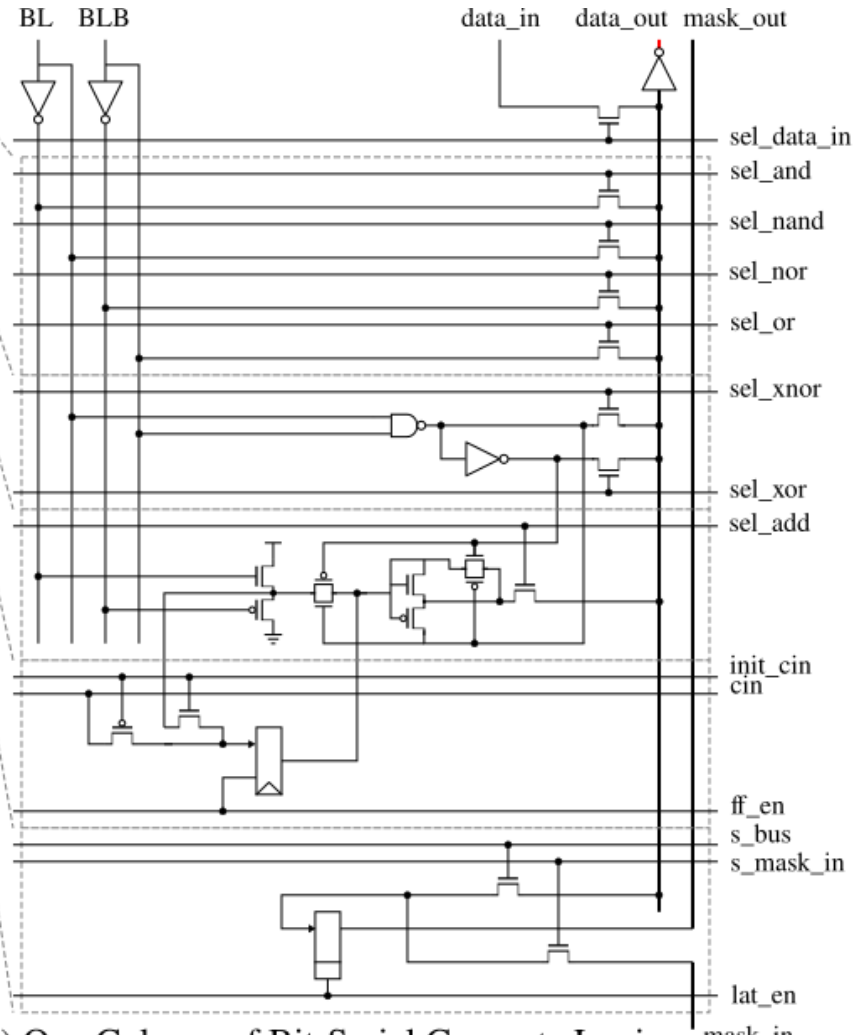
VRAM: Circuits—Bit-Serial Compute Logic



(a) VRAM Block Diagram

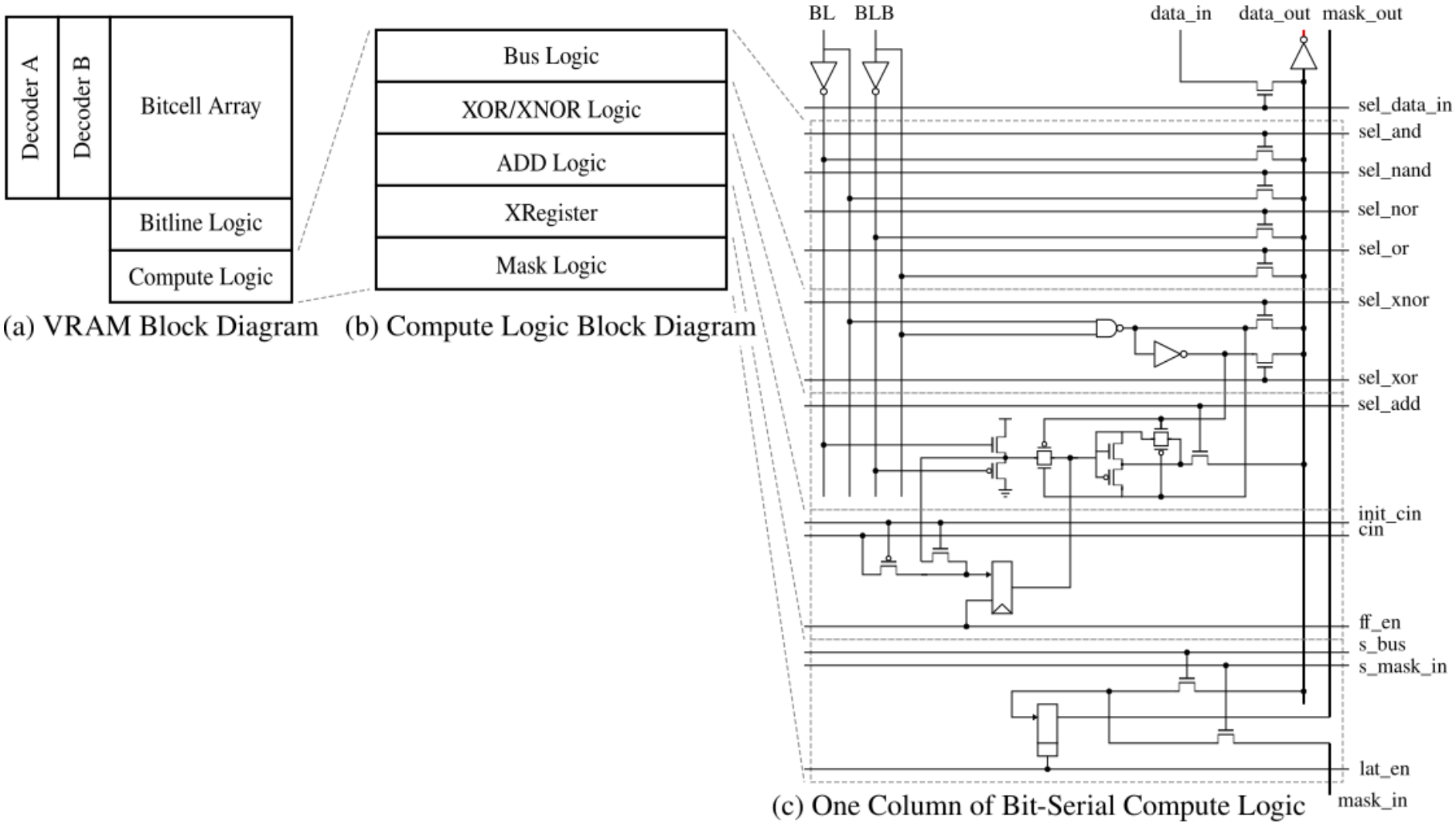


(b) Compute Logic Block Diagram



(c) One Column of Bit-Serial Compute Logic

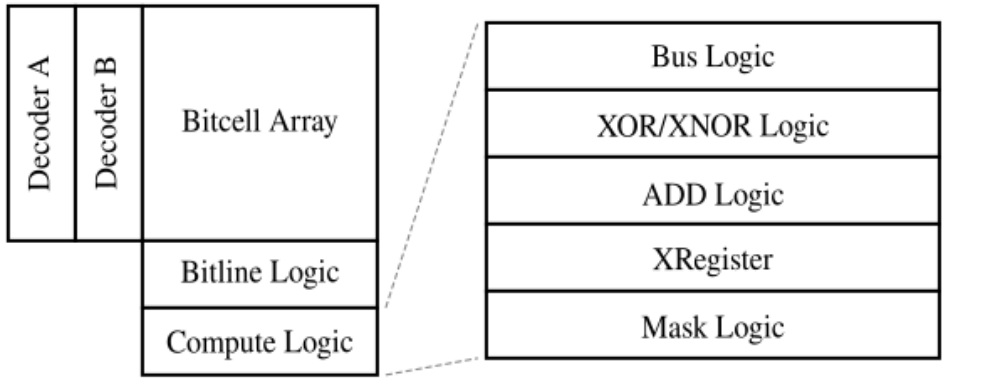
VRAM: Circuits—Bit-Serial Compute Logic



(a) VRAM Block Diagram (b) Compute Logic Block Diagram

(c) One Column of Bit-Serial Compute Logic

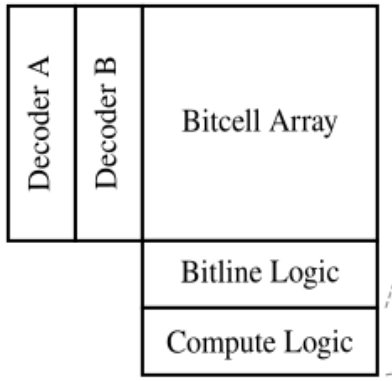
VRAM: Circuits—Bit-Parallel Compute Logic



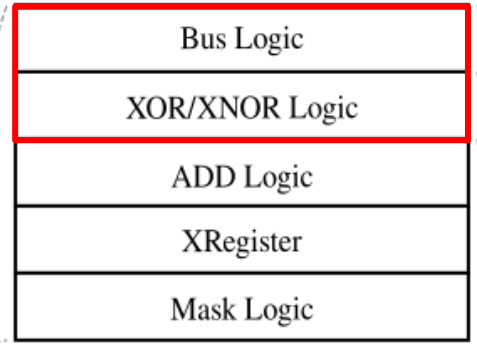
(a) VRAM Block Diagram (b) Compute Logic Block Diagram

(d) Bit-Parallel Add, XRegister, Mask Logic

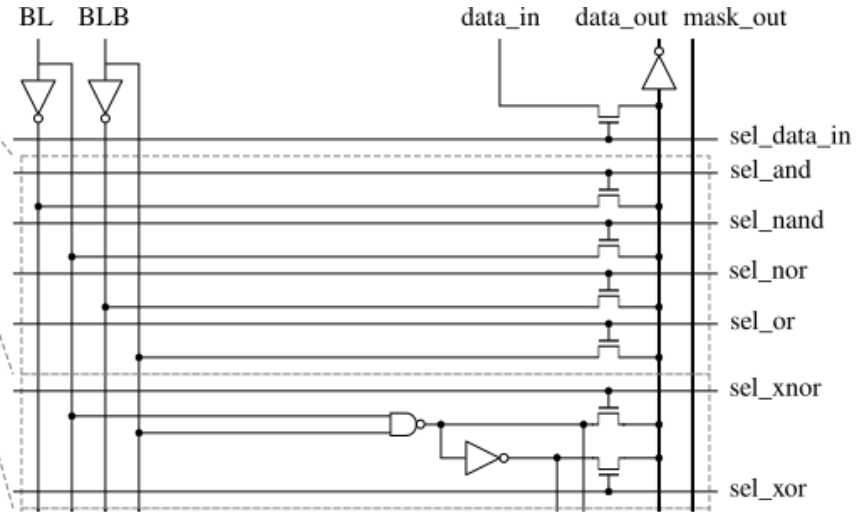
VRAM: Circuits—Bit-Parallel Compute Logic



(a) VRAM Block Diagram

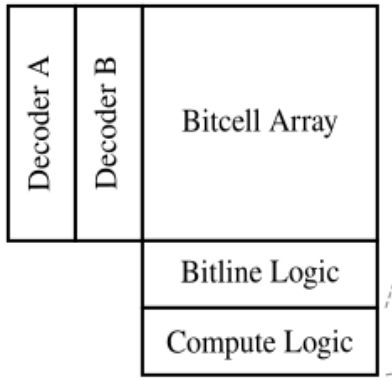


(b) Compute Logic Block Diagram

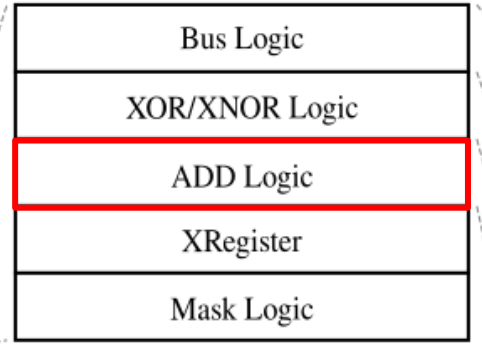


(d) Bit-Parallel Add, XRegister, Mask Logic

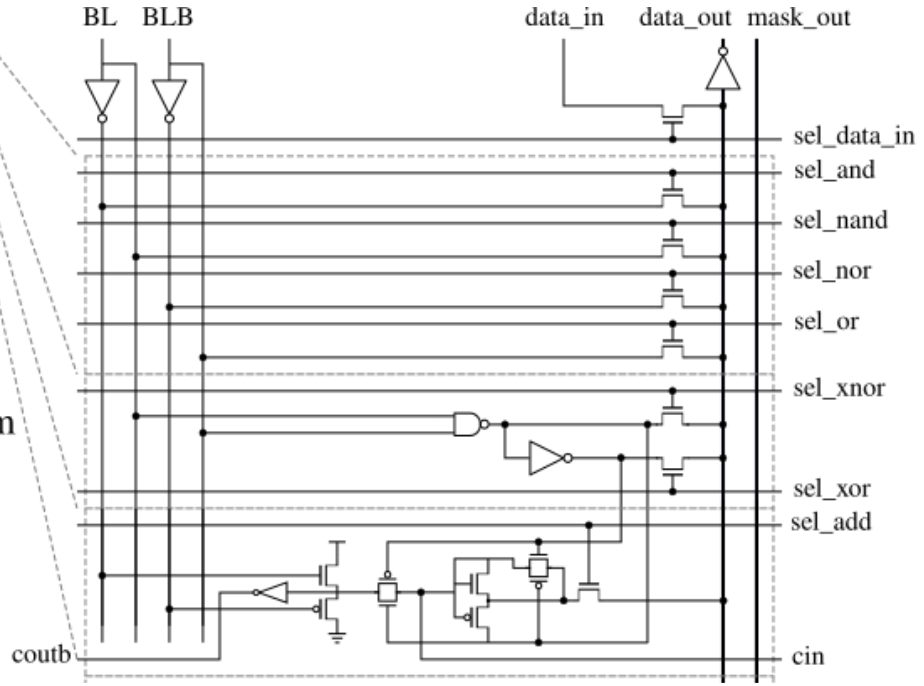
VRAM: Circuits—Bit-Parallel Compute Logic



(a) VRAM Block Diagram

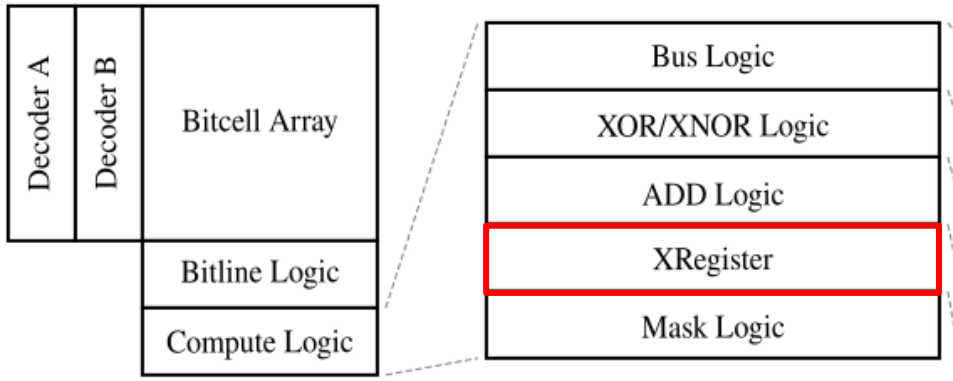


(b) Compute Logic Block Diagram



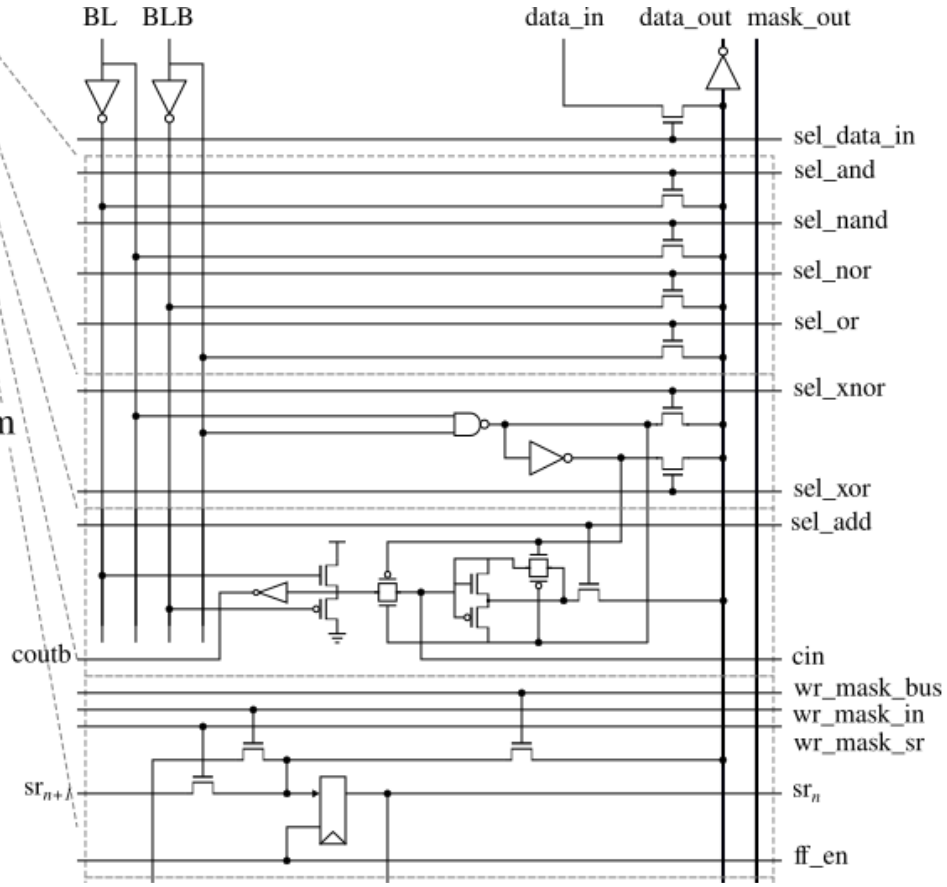
(d) Bit-Parallel Add, XRegister, Mask Logic

VRAM: Circuits—Bit-Parallel Compute Logic



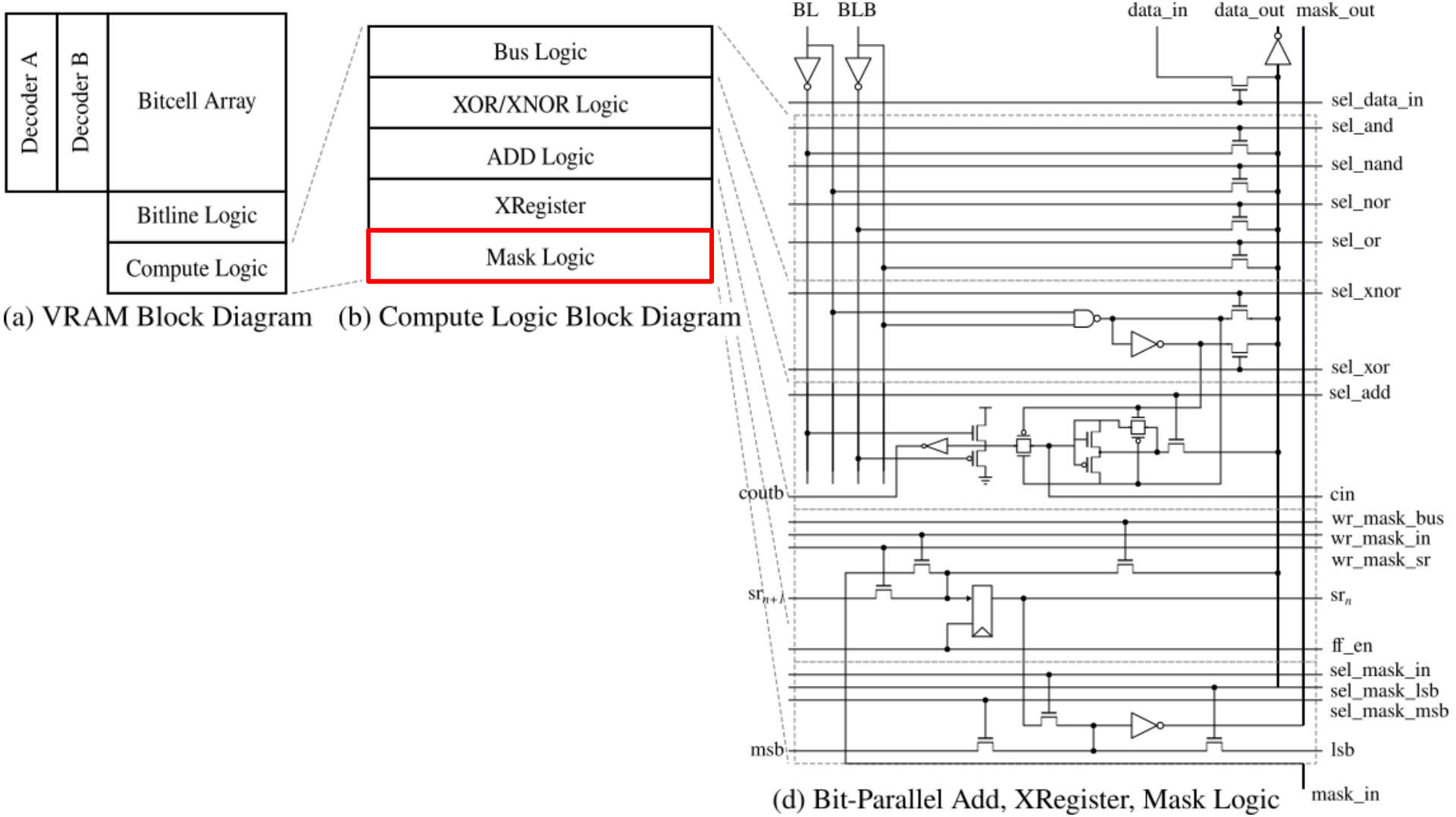
(a) VRAM Block Diagram

(b) Compute Logic Block Diagram



(d) Bit-Parallel Add, XRegister, Mask Logic

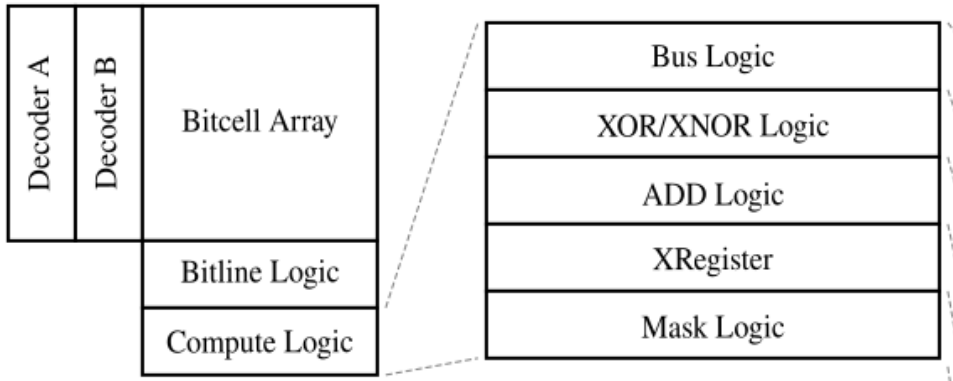
VRAM: Circuits—Bit-Parallel Compute Logic



(a) VRAM Block Diagram (b) Compute Logic Block Diagram

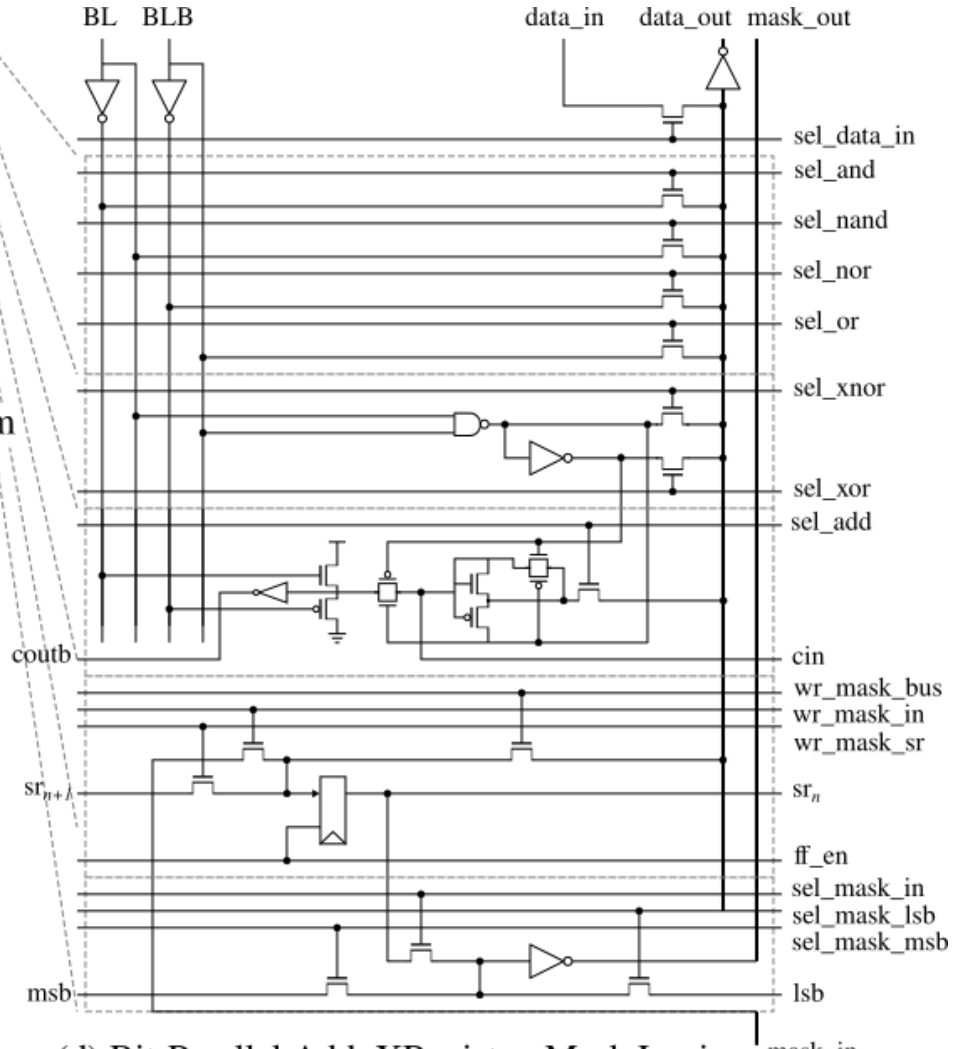
(d) Bit-Parallel Add, XRegister, Mask Logic

VRAM: Circuits—Bit-Parallel Compute Logic



(a) VRAM Block Diagram

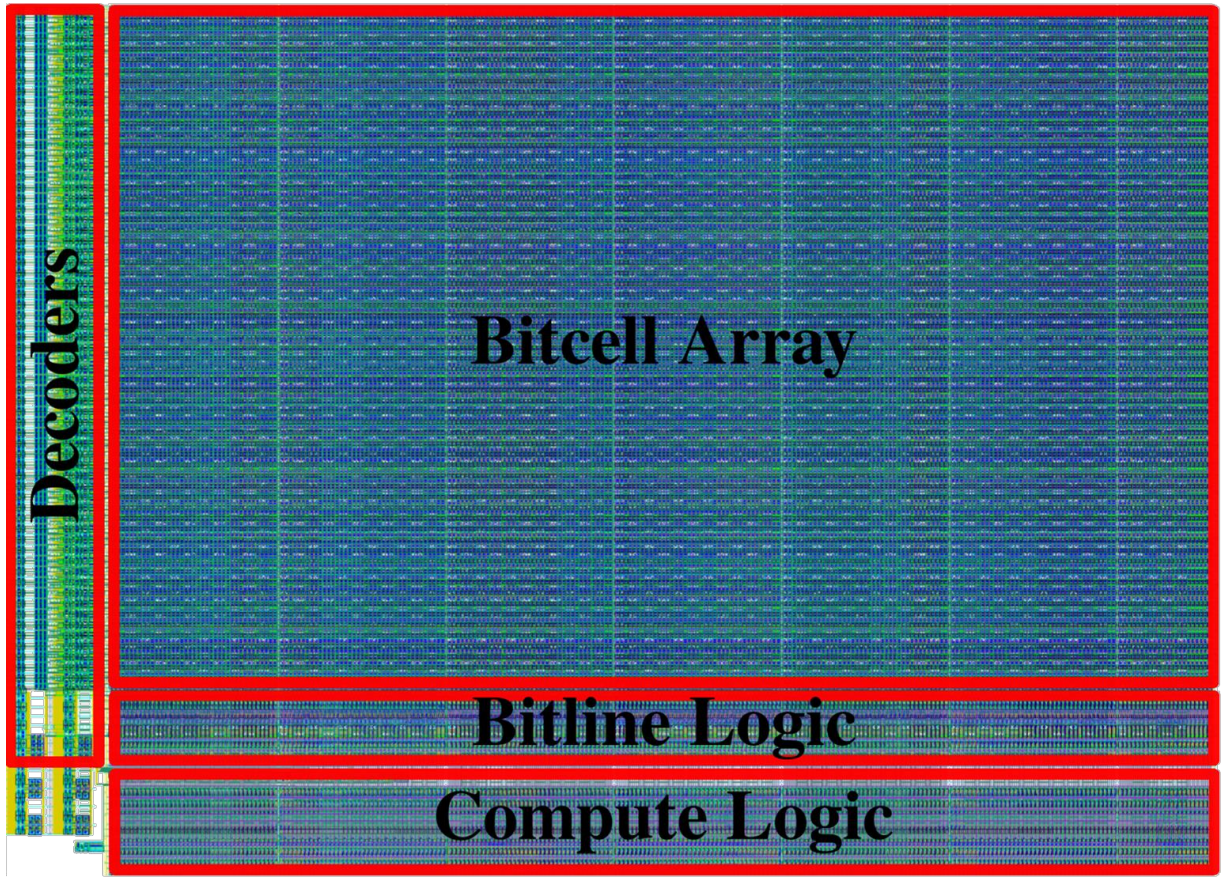
(b) Compute Logic Block Diagram



(d) Bit-Parallel Add, XRegister, Mask Logic

Outline

- Motivation
- Background: Bit-line Compute
- **Vector RAM**
  - VRAM Circuits
  - **VRAM Micro-Programming**
  - VRAM Macro-Programming
  - Evaluation
- Conclusion



Arithmetic μOps:

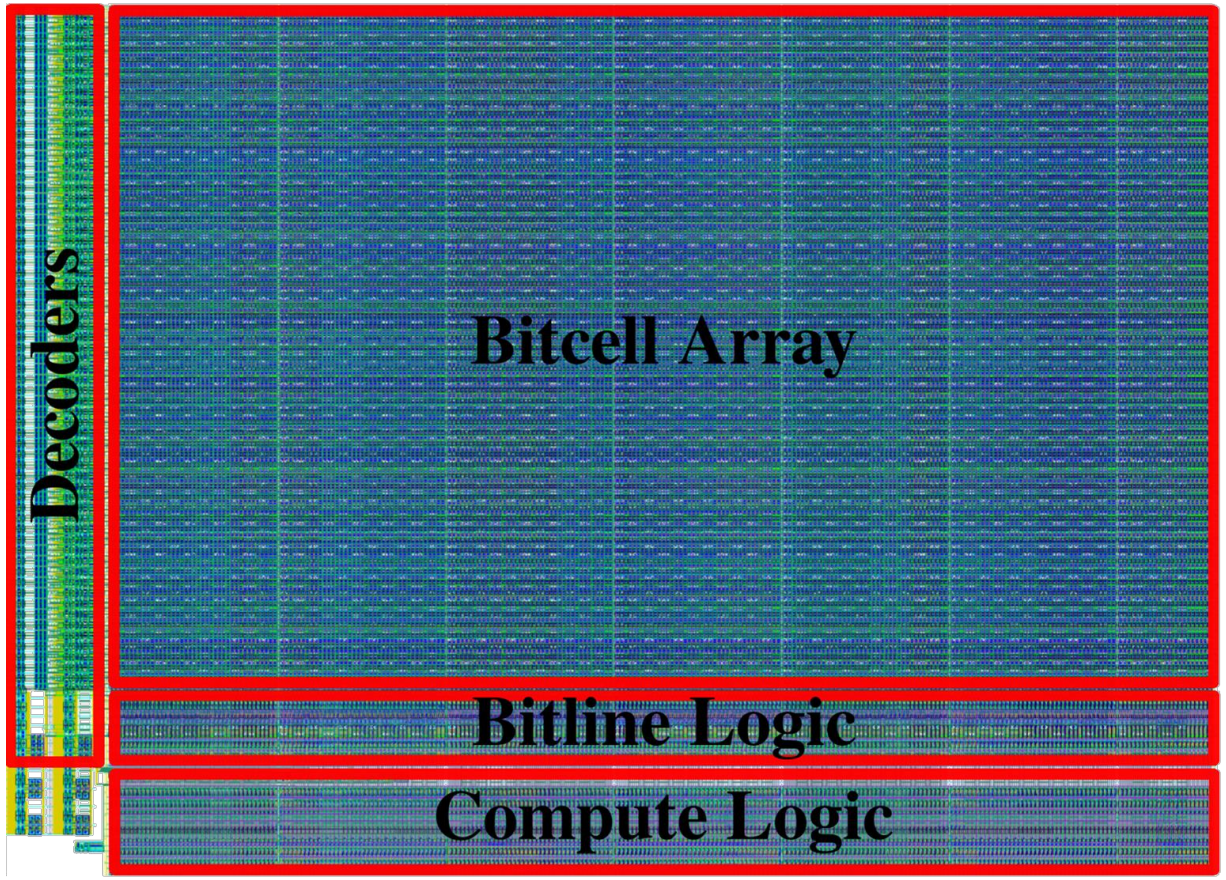
- *Bit-line Compute (blc)*: Perform bit-line compute between two operands.
- *Writeback (cond.wb.src)*: Writeback a specified logic stack to SRAM.
- *Write to Mask (wr_mask.src)*: Writeback a specified logic stack to mask.
- *Shift Right Logical (srl):* Shift the content of the XRegister logic right by one bit.

Control μOps:

- *Jump If not Done (j_n_done_{0, 1}):* Decrement specified counter and jump to label if counter not zero.

Outline

- Motivation
- Background: Bit-line Compute
- **Vector RAM**
  - VRAM Circuits
  - VRAM Micro-Programming
  - **VRAM Macro-Programming**
  - Evaluation
- Conclusion



VRAM Macro-Programming

```
1    set_cin       1
2    wb_mask       <(1)
   init:
3    wr            addr_C <(0)
     ; j_n_done_0 init
   iter:
4    rd            addr_B
5    wr_mask.and
   iter_add:
6    blc           addr_C, addr_A
7    wb.add        addr_C
     ; j_n_done_1 iter_add
8    j_n_done_0    iter
```

(c) mul in BS-VRAM

```
1    wr            addr_c <(0)
2    rd            addr_a
3    wb.and t0
4    rd            addr_b
5    wr_mask.and
   iter:
6    blc           addr_c, t0
7    msb.wr.add    addr_c
     ; srl
8    rd            t0
9    wb.add        t0
     ; j_n_done_0 iter
```

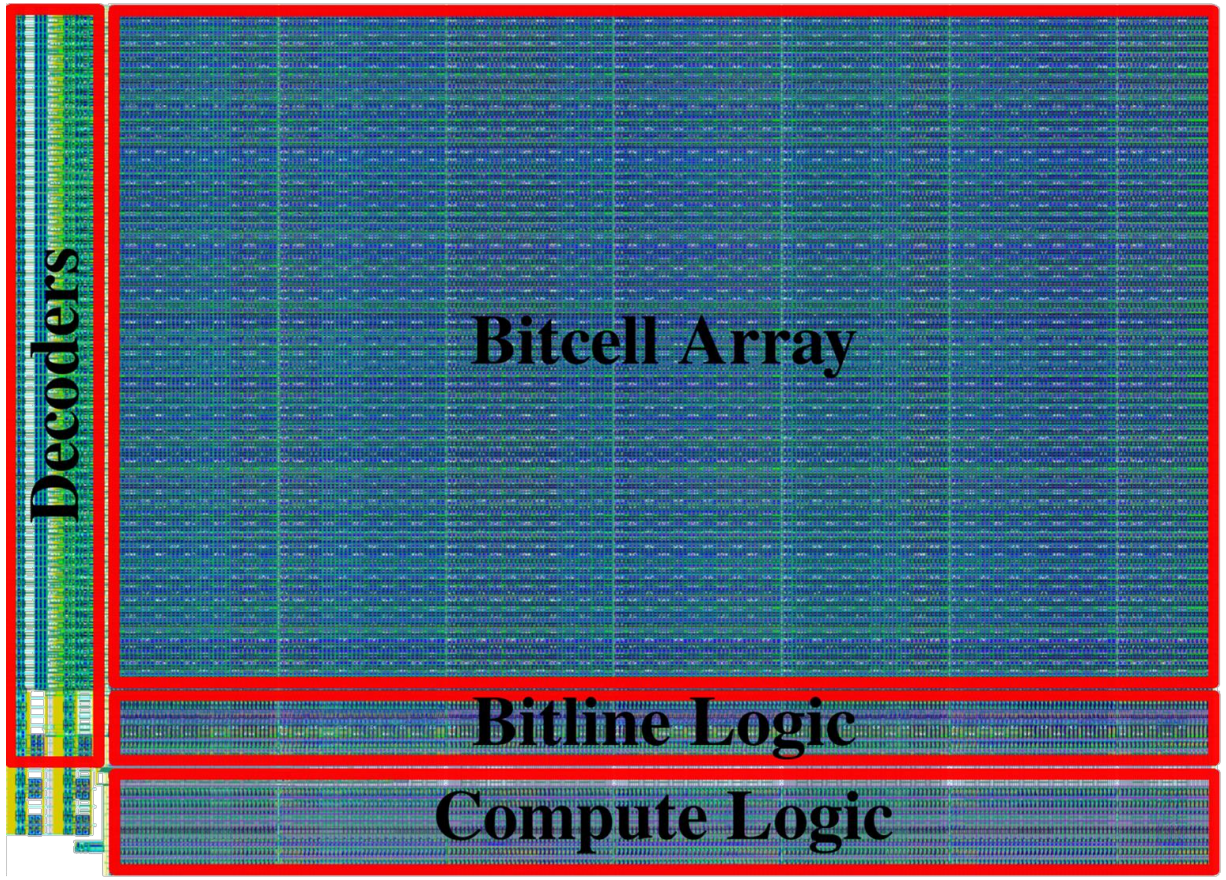
(d) mul in BP-VRAM

TABLE I. SUPPORTED MACRO-OPERATIONS

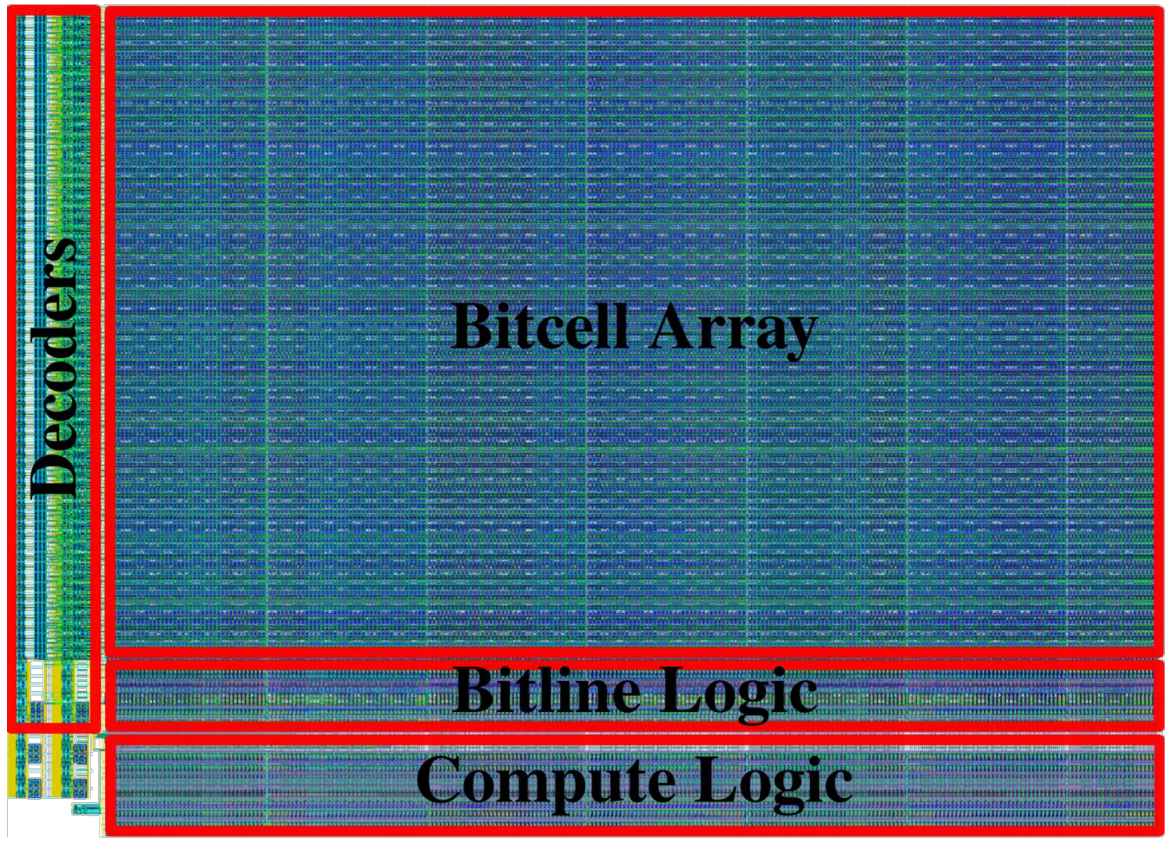
Macro-Operation	Cycle Count		# Temporary Rows	
	BS-VRAM	BP-VRAM	BS-VRAM	BP-VRAM
`add`	64	2	0	0
`sub`	128	4	0	0
`and,nand,or nor,xor,xnor`	64	2	0	0
`mul`	1185	133	0	1
`mac`	1153	132	0	1
`udiv`	1712	519	5	1
`rem`	1680	390	4	2
`slt,sle,sgt,sge`	162	6	1	0
`seq`	96	11	1	1

Outline

- Motivation
- Background: Bit-line Compute
- **Vector RAM**
  - VRAM Circuits
  - VRAM Micro-Programming
  - VRAM Macro-Programming
  - **Evaluation**
- Conclusion



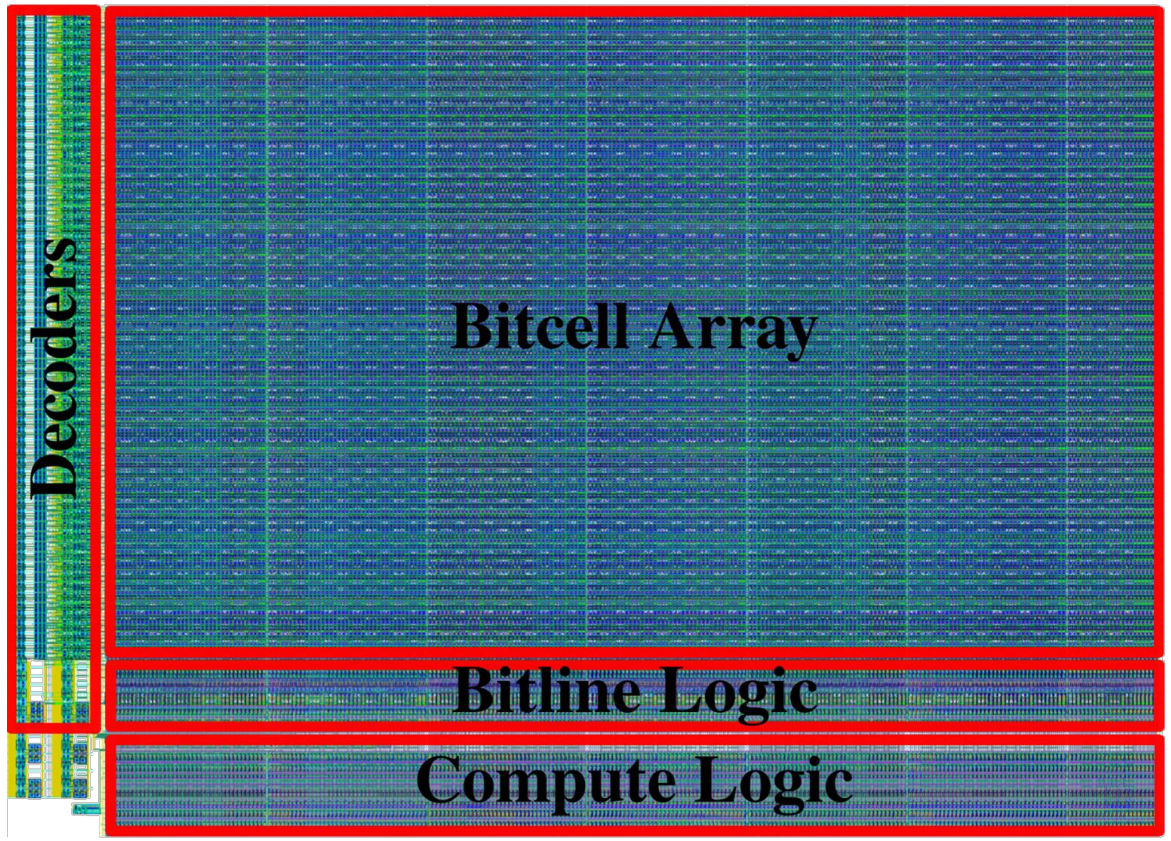
Evaluation—Methodology



- **Using OpenRAM, we generate layout for SRAM, BL-SRAM, BS-VRAM, and BP-VRAM on 28nm.**
- **We evaluate the following metrics:**
  - **Area**: measured from the layout
  - **Frequency**: measured from post-extraction netlist
  - **Throughput**: estimated using the freq. and number of dynamic μOps
  - **Energy**: measured from post-extraction by averaging 1000 random μOps

* M. R. Guthaus et al., "OpenRAM: An Open-Source Memory Compiler." ICCAD'16

Evaluation—Compared to Commercial SRAM



- **Compared to commercial 28nm SRAM generator:**
  - **Area:** Compared to commercial SRAM generator, bitcell is 80% bigger (i.e., taller).
  - **Energy:** Writes are 1.5x higher and reads are 3x.
  - **Frequency:** Operating frequency is 45% slower (1.1GHz vs 2GHz).
- **Lower frequency for BS-VRAM (900MHz) & BP-VRAM (645MHz).**
- **There is room for improvement, but the goal is to evaluate BS vs BP approach.**

VRAM Evaluation—BS-VRAM vs. BP-VRAM

CSL

Table IV. Detailed Comparsion Table Between BS-VRAM and BP-VRAM

		32-bit		8-bit	
		BS	BP	BS	BP
Latency (ns)	add	71.1	**3.1** (23.0×)	17.8	**3.1** (5.8×)
	mul	1316.7	**206.2** (6.4×)	116.7	**57.4** (2.0×)
Throughput (GOPS)	add	**3.6** (1.4×)	2.6	**14.4** (5.5×)	2.6
	mul	**0.2** (5.0×)	0.04	**2.2** (15.7×)	0.14
Energy (pJ/Op)	add	4.7	4.8	1.2	4.8
	mul	112.5	221.3	9.0	58.1

VRAM Evaluation—Compared to Previous Work

- **Comparing vector RAMs with three previous works**
  - **ISSCC '19**
    Based on 8T-SRAM; utilizes in-situ processing-in-SRAM to perform vector operations covering most functionalities similar to VRAM.
  - **JSSC '16**
    Based on 6T-SRAM; utilizes in-situ processing-in-SRAM to perform bit-wise logical operations only.
  - **VLSI '17**
    Based on 10T-SRAM; utilizes in-situ processing-in-SRAM to perform fixed-function cryptography acceleration.

* J. Wang et al.,"A Compute SRAM with Bit-Serial Integer/Floating-Point Operations for Programmable In-Memory Vector Acceleration." Int'l Solid-State Circuits Conf '19
* S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.
* Y. Zhang et al., "A Reconfigurable In-Memory Cryptographic Cortex-M0 Processor for IoT." Symp. on Very Large-Scale Integration Circuits (VLSIC) '17

VRAM Evaluation—Compared to Previous Work

Paper	VRAM		ISSCC'19	JSSC'16	VLSI'17
	BS	BP	[19]	[9]	[21]
Technology	28nm	28nm	28nm	28nm	40nm
Voltage	0.9V	0.9V	0.9V	0.9V	0.9V
SRAM Capacity	128kB	128kB	128kB	128kB	128kB
SRAM Macro	4kB	4kB	16kB	0.5kB	8kB
SRAM Bitcell	6T	6T	8T	6T	10T
Precision	Arb.	32b	Arb.	Arb.	Arb.
Freq (MHz)	900	645	225	594	90
Area (mm^2)*	1.1	1.1	2.7	0.7	1.28
Logic Ops	✓	✓	✓	✓(a)	✓(b)
Basic Int Ops	✓	✓	✓		
Cmplx Int Ops	✓	✓	✓(c)		
Cmp Ops	✓	✓	✓(d)		
Search			✓	✓	
FX Ops	✓	✓			
FP Ops			✓		
8b MAC GOPS	76.0	4.5	4.2	n/a	n/a
8b MAC GOPS/W	115.5	17.2	245.5	n/a	n/a
32b MAC GOPS	6.4	1.2	0.4	n/a	n/a
32b MAC GOPS/W	9.0	4.5	22.5	n/a	n/a

* J. Wang et al.,"A Compute SRAM with Bit-Serial Integer/Floating-Point Operations for Programmable In-Memory Vector Acceleration." Int'l Solid-State Circuits Conf '19
* S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.
* Y. Zhang et al., "A Reconfigurable In-Memory Cryptographic Cortex-M0 Processor for IoT." Symp. on Very Large-Scale Integration Circuits (VLSIC) '17

VRAM Evaluation—Compared to Previous Work

Paper	VRAM		ISSCC'19	JSSC'16	VLSI'17
	BS	BP	[19]	[9]	[21]
Technology	28nm	28nm	28nm	28nm	40nm
Voltage	0.9V	0.9V	0.9V	0.9V	0.9V
SRAM Capacity	128kB	128kB	128kB	128kB	128kB
SRAM Macro	4kB	4kB	16kB	0.5kB	8kB
SRAM Bitcell	6T	6T	8T	6T	10T
Precision	Arb.	32b	Arb.	Arb.	Arb.
Freq (MHz)	900	645	225	594	90
Area (mm^2)*	1.1	1.1	2.7	0.7	1.28
Logic Ops	✓	✓	✓	✓(a)	✓(b)
Basic Int Ops	✓	✓	✓		
Cmplx Int Ops	✓	✓	✓(c)		
Cmp Ops	✓	✓	✓(d)		
Search			✓	✓	
FX Ops	✓	✓			
FP Ops			✓		
8b MAC GOPS	76.0	4.5	4.2	n/a	n/a
8b MAC GOPS/W	115.5	17.2	245.5	n/a	n/a
32b MAC GOPS	6.4	1.2	0.4	n/a	n/a
32b MAC GOPS/W	9.0	4.5	22.5	n/a	n/a

- **BS-VRAM is representative of previous work; it achieves higher throughput (up to 18x).**

* J. Wang et al.,"A Compute SRAM with Bit-Serial Integer/Floating-Point Operations for Programmable In-Memory Vector Acceleration." Int'l Solid-State Circuits Conf '19
* S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.
* Y. Zhang et al., "A Reconfigurable In-Memory Cryptographic Cortex-M0 Processor for IoT." Symp. on Very Large-Scale Integration Circuits (VLSIC) '17

VRAM Evaluation—Compared to Previous Work

Paper	VRAM		ISSCC'19	JSSC'16	VLSI'17
	BS	BP	[19]	[9]	[21]
Technology	28nm	28nm	28nm	28nm	40nm
Voltage	0.9V	0.9V	0.9V	0.9V	0.9V
SRAM Capacity	128kB	128kB	128kB	128kB	128kB
SRAM Macro	4kB	4kB	16kB	0.5kB	8kB
SRAM Bitcell	6T	6T	8T	6T	10T
Precision	Arb.	32b	Arb.	Arb.	Arb.
Freq (MHz)	900	645	225	594	90
Area (mm^2)*	1.1	1.1	2.7	0.7	1.28
Logic Ops	✓	✓	✓	✓(a)	✓(b)
Basic Int Ops	✓	✓	✓		
Cmplx Int Ops	✓	✓	✓(c)		
Cmp Ops	✓	✓	✓(d)		
Search			✓	✓	
FX Ops	✓	✓			
FP Ops			✓		
8b MAC GOPS	76.0	4.5	4.2	n/a	n/a
8b MAC GOPS/W	115.5	17.2	245.5	n/a	n/a
32b MAC GOPS	6.4	1.2	0.4	n/a	n/a
32b MAC GOPS/W	9.0	4.5	22.5	n/a	n/a

- **BS-VRAM is representative of previous work; it achieves higher throughput (up to 18x).**

* J. Wang et al.,"A Compute SRAM with Bit-Serial Integer/Floating-Point Operations for Programmable In-Memory Vector Acceleration." Int'l Solid-State Circuits Conf '19
* S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.
* Y. Zhang et al., "A Reconfigurable In-Memory Cryptographic Cortex-M0 Processor for IoT." Symp. on Very Large-Scale Integration Circuits (VLSIC) '17

VRAM Evaluation—Compared to Previous Work

Paper	VRAM		ISSCC'19	JSSC'16	VLSI'17
	BS	BP	[19]	[9]	[21]
Technology	28nm	28nm	28nm	28nm	40nm
Voltage	0.9V	0.9V	0.9V	0.9V	0.9V
SRAM Capacity	128kB	128kB	128kB	128kB	128kB
SRAM Macro	4kB	4kB	16kB	0.5kB	8kB
SRAM Bitcell	6T	6T	8T	6T	10T
Precision	Arb.	32b	Arb.	Arb.	Arb.
Freq (MHz)	900	645	225	594	90
Area (mm^2)*	1.1	1.1	2.7	0.7	1.28
Logic Ops	✓	✓	✓	✓(a)	✓(b)
Basic Int Ops	✓	✓	✓		
Cmplx Int Ops	✓	✓	✓(c)		
Cmp Ops	✓	✓	✓(d)		
Search			✓	✓	
FX Ops	✓	✓			
FP Ops			✓		
8b MAC GOPS	76.0	4.5	4.2	n/a	n/a
8b MAC GOPS/W	115.5	17.2	245.5	n/a	n/a
32b MAC GOPS	6.4	1.2	0.4	n/a	n/a
32b MAC GOPS/W	9.0	4.5	22.5	n/a	n/a

- **BS-VRAM is representative of previous work; it achieves higher throughput (up to 18x).**
- **BS-VRAM and BP-VRAM occupy small footprint due to the use of 6T-SRAM.**

* J. Wang et al.,"A Compute SRAM with Bit-Serial Integer/Floating-Point Operations for Programmable In-Memory Vector Acceleration." Int'l Solid-State Circuits Conf '19
* S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.
* Y. Zhang et al., "A Reconfigurable In-Memory Cryptographic Cortex-M0 Processor for IoT." Symp. on Very Large-Scale Integration Circuits (VLSIC) '17

VRAM Evaluation—Compared to Previous Work

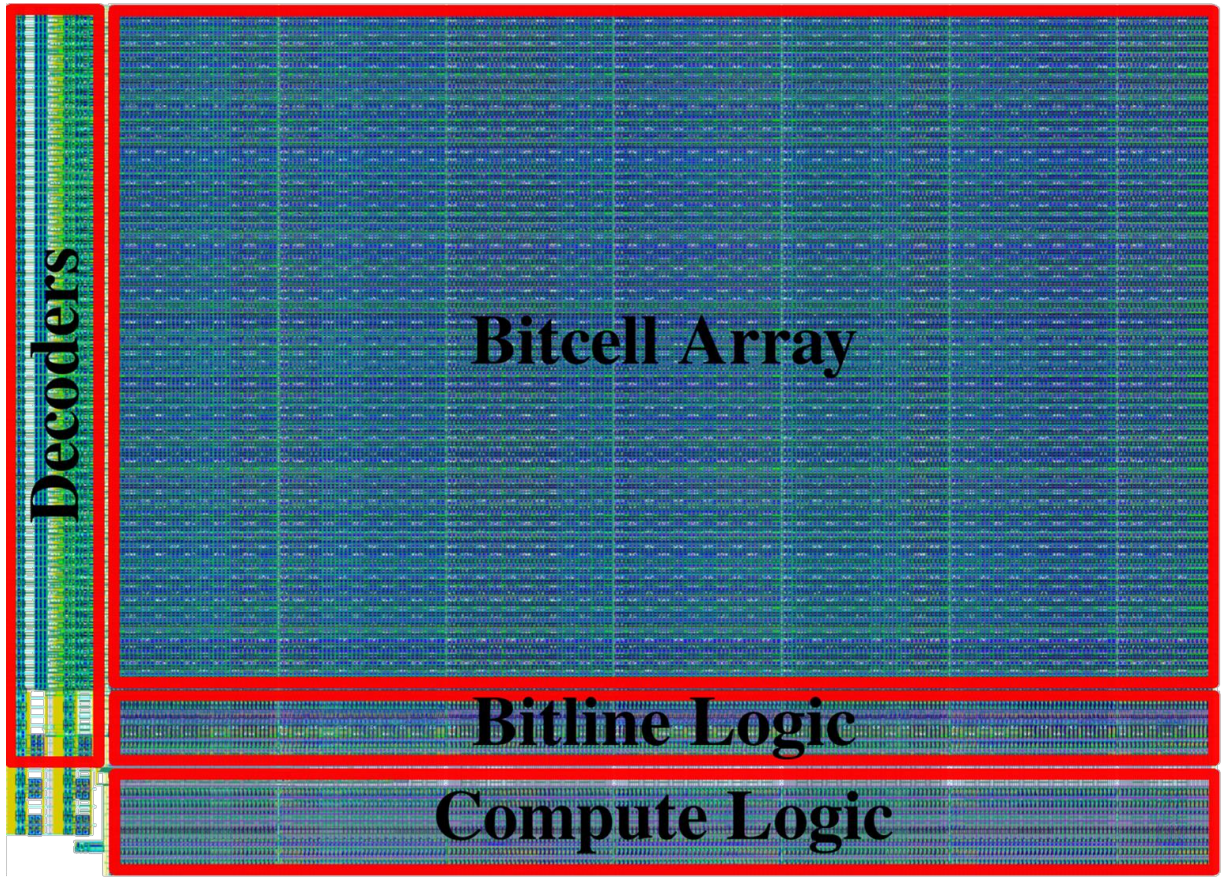
Paper	VRAM		ISSCC'19	JSSC'16	VLSI'17
	BS	BP	[19]	[9]	[21]
Technology	28nm	28nm	28nm	28nm	40nm
Voltage	0.9V	0.9V	0.9V	0.9V	0.9V
SRAM Capacity	128kB	128kB	128kB	128kB	128kB
SRAM Macro	4kB	4kB	16kB	0.5kB	8kB
SRAM Bitcell	6T	6T	8T	6T	10T
Precision	Arb.	32b	Arb.	Arb.	Arb.
Freq (MHz)	900	645	225	594	90
Area (mm^2)*	1.1	1.1	2.7	0.7	1.28
Logic Ops	✓	✓	✓	✓(a)	✓(b)
Basic Int Ops	✓	✓	✓		
Cmplx Int Ops	✓	✓	✓(c)		
Cmp Ops	✓	✓	✓(d)		
Search			✓	✓	
FX Ops	✓	✓			
FP Ops			✓		
8b MAC GOPS	76.0	4.5	4.2	n/a	n/a
8b MAC GOPS/W	115.5	17.2	245.5	n/a	n/a
32b MAC GOPS	6.4	1.2	0.4	n/a	n/a
32b MAC GOPS/W	9.0	4.5	22.5	n/a	n/a

- **BS-VRAM is representative of previous work; it achieves higher throughput (up to 18x).**
- **BS-VRAM and BP-VRAM occupy small footprint due to the use of 6T-SRAM.**
- **BS-VRAM has lower efficiency as there is room for improvement especially in peripheral circuits.**

* J. Wang et al.,"A Compute SRAM with Bit-Serial Integer/Floating-Point Operations for Programmable In-Memory Vector Acceleration." Int'l Solid-State Circuits Conf '19
* S. Jeloka et al., "A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T BitCell Enabling Logic-in-Memory." IEEE Journal of Solid-State Circuits '16.
* Y. Zhang et al., "A Reconfigurable In-Memory Cryptographic Cortex-M0 Processor for IoT." Symp. on Very Large-Scale Integration Circuits (VLSIC) '17

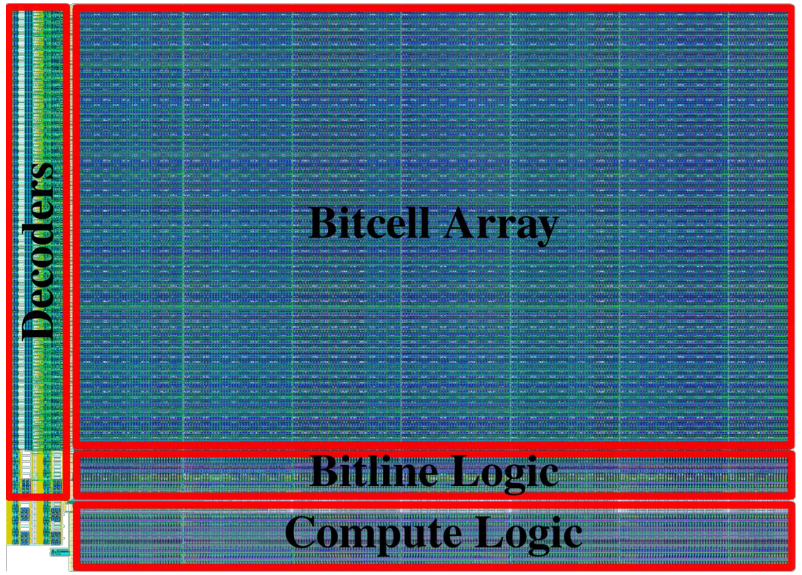
Outline

- Motivation
- Background: Bit-line Compute
- Vector RAM
  - VRAM Circuits
  - VRAM Micro-Programming
  - VRAM Macro-Programming
  - Results
- Conclusion



Conclusion

- Vector RAM (VRAM) is among the first work to explore the implementation of an efficient vector accelerator using bit-serial/bit-parallel execution paradigms leveraging in-situ processing-in-SRAM.



- Bit-serial execution enables BS-VRAM to achieve higher throughput compared to BP-VRAM; whereas, BP-VRAM leverages the low cycle-count for bit-parallel execution to achieve lower latency.

- There is an interesting design-space between bit-serial and bit-parallel flavors trading off latency and throughput.