Supplementary Material for DNA Pattern Matching Acceleration with Analog CAMs

Supplementary Note S1: Block Selector Implementation

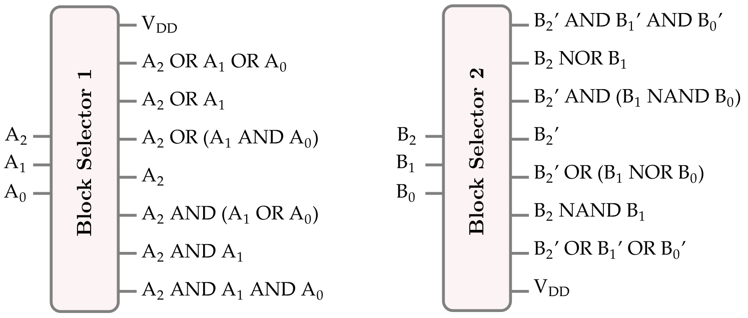
The first block selector takes an input i and selects the i^{th} block and all the blocks above it. On the other hand, block selector 2 with an input equal to i selects the i^{th} block and all the blocks below it, as illustrated in the truth Tables S1 and S2 of block selectors 1 and 2, respectively. Figure S1 shows the implementation of the two block selectors.

Table S1. Truth Table for Block Selector 1

Input			Outputs							
A_2	A_1	A_0	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0
0	1	0	1	1	1	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0
1	0	0	1	1	1	1	1	0	0	0
1	0	1	1	1	1	1	1	1	0	0
1	1	0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1

Table S2. Truth Table for Block Selector 2

Input			Outputs							
B_2	B_1	B_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	1	1	1
0	1	1	0	0	0	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1
1	0	1	0	0	0	0	0	1	1	1
1	1	0	0	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	0	0	1



A_2
A_1
A_0
Block Selector 1
V_{DD}
A_2 OR A_1 OR A_0
A_2 OR A_1
A_2 OR (A_1 AND A_0)
A_2
A_2 AND (A_1 OR A_0)
A_2 AND A_1
A_2 AND A_1 AND A_0
B_2
B_1
B_0
Block Selector 2
B_2' AND B_1' AND B_0'
B_2 NOR B_1
B_2' AND (B_1 NAND B_0)
B_2'
B_2' OR (B_1 NOR B_0)
B_2 NAND B_1
B_2' OR B_1' OR B_0'
V_{DD}

Figure S1. Block Selectors Implementation.

Supplementary Note S2: Pattern Detector

Figure S2 shows the implementation of the block for pointer i, which consists of a counter, comparator, and maximum register to store the maximum value of the pattern occurrences traversed by pointer i.

Figure S3 illustrates the implementation of the comparison logic block diagram, where the maximum values of pattern occurrences corresponding to pointers 1 and 2 are first compared. The maximum among them is then compared with the maximum corresponding to pointer 3, and finally, the global maximum is stored in the global max register.

The state-transition table for the finite state machine is shown in Table S3. The next state and outputs are determined depending on the current state and input xD.

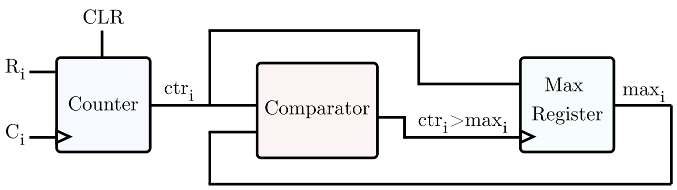


Figure S2. Block for Pointer i.

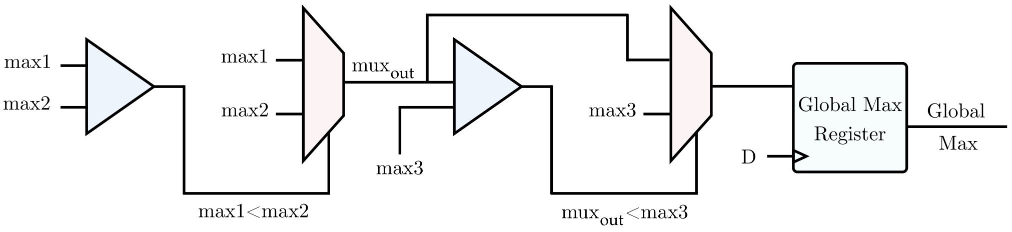


Figure S3. Comparison Logic Block Diagram.

Table S3. State-Transition Table for the proposed finite state machine

	Next State				Outputs						
Current State	xD=00	xD=01	xD=10	xD=11	C1	C2	C3	R1	R2	R3	CLR
Initial	S1	Exit	S2	Exit	0	0	0	0	0	0	1
S1	S3	Exit	S4	Exit	0	0	0	1	0	0	0
S2	S3	Exit	S4	Exit	1	0	0	0	0	0	0
S3	S5	Exit	S6	Exit	0	0	0	0	1	0	0
S4	S5	Exit	S6	Exit	0	1	0	0	0	0	0
S5	S1	Exit	S2	Exit	0	0	0	0	0	1	0
S6	S1	Exit	S2	Exit	0	0	1	0	0	0	0
Exit	Exit	Exit	Exit	Exit	0	0	0	0	0	0	1

Supplementary Note S3: Memory Operation Modes

In the 1T1R memory array, the values of the signals at each node (SL, X, and Y) will vary depending on the write, read, or reset mode of the memory as shown in table S4.

Table S4. Signal Configuration for different memory operation modes

	Write	Read	Reset
SL	V_{set}	0	0
X	0/1	0/1	1
Y	0	open	V_{reset}
Read-ckt	open	closed	open