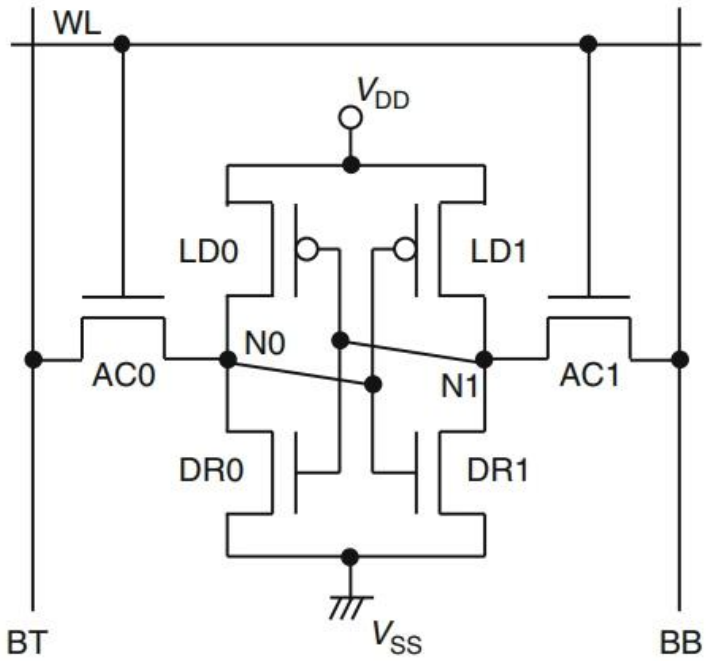
C3SRAM: An In-Memory-Computing SRAM Macro Based on Robust Capacitive Coupling Computing Mechanism

Zhewei Jiang, *Student Member, IEEE*, Shihui Yin, *Student Member, IEEE*, Jae-sun Seo, *Senior Member, IEEE*, and Mingoo Seok, *Senior Member, IEEE*

SRAM的优势

- ✓SRAM array operates fast as logic circuits operate, and consumes a little power at standby mode.
- ✓SRAM cell is fabricated by same process as logic, so that it does not need extra process cost.

6T SRAM Cell



LD0, LD1: Load MOSFET
AC0, AC1: Access MOSFET
DR0, DR1: Driver MOSFET
WL: Word line
BT, BB: Bit line
N0, N1: Cell node

6T SRAM单元由两个反相器(load MOSFET (LD0)-driver MOSFET(DR0), LD1-DR1), 分别连接到互补的bit line(BT,BB)的两个access MOSFETs(AC0, AC1)组成。同时为了实现选通, AC0和AC1的gate连接到WL。为了实现双稳态存储, 两个反相器的输入和输出连接到一起。

Abstract

该论文实现了基于SRAM的存内计算架构C3SRAM，能够实现input和weight都是二值的神经网络加速。在该结构中，每一列对应一个ADC，所以能够并行的计算整个macro。(需要注意为什么ADC个数这么多，ADC带来的energy/area overhead却不是特别严重)。65-nm的C3SRAM芯片原型的能效为672 TOPS/W，算力为1638GOPS(20.2TOPS/mm^2)。计算结果：MNIST的accuracy为98.3%，CIFAR-10的accuraycy为85.5%。

该计算结构虽然采用的是模拟域实现计算，但是在计算过程中，仍然没有常开的回路，只涉及到C_C的充/放电，动态功耗与数字电路相似。

Multi-Bit Weights in IMC Designs

传统的神经网络输入和权重都是多比特的数值，但是memory bitcell的物理拓扑是相互独立的，IMC中多比特权重的表达是在电路级实现而不是架构级去实现（这句话的理解不够深，需要去看参考文献。现在的理解是多比特权重映射到memory上，需要有固定的硬件电路去实现多比特结果的聚合，而不能通过系统指令去进行控制任意两个结果相加。在ReRAM PIM中，我们能够在电路级实现多比特的关联，但是如果一层不能映射到一个macro上，就不能在电路级实现了。）实现多比特权重神经网络的工作有参考文献[6-9]。例如在参考文献[6]中，通过多个SRAM cell来存储多比特权值，比特位之间通过晶体管的宽度比联系在一起。(这个比较有创意)
对于新型存储器如PCM，RRAM等，由于工艺不成熟存在器件特性不稳定和非线性的限制。

Binary Weights in IMC Design

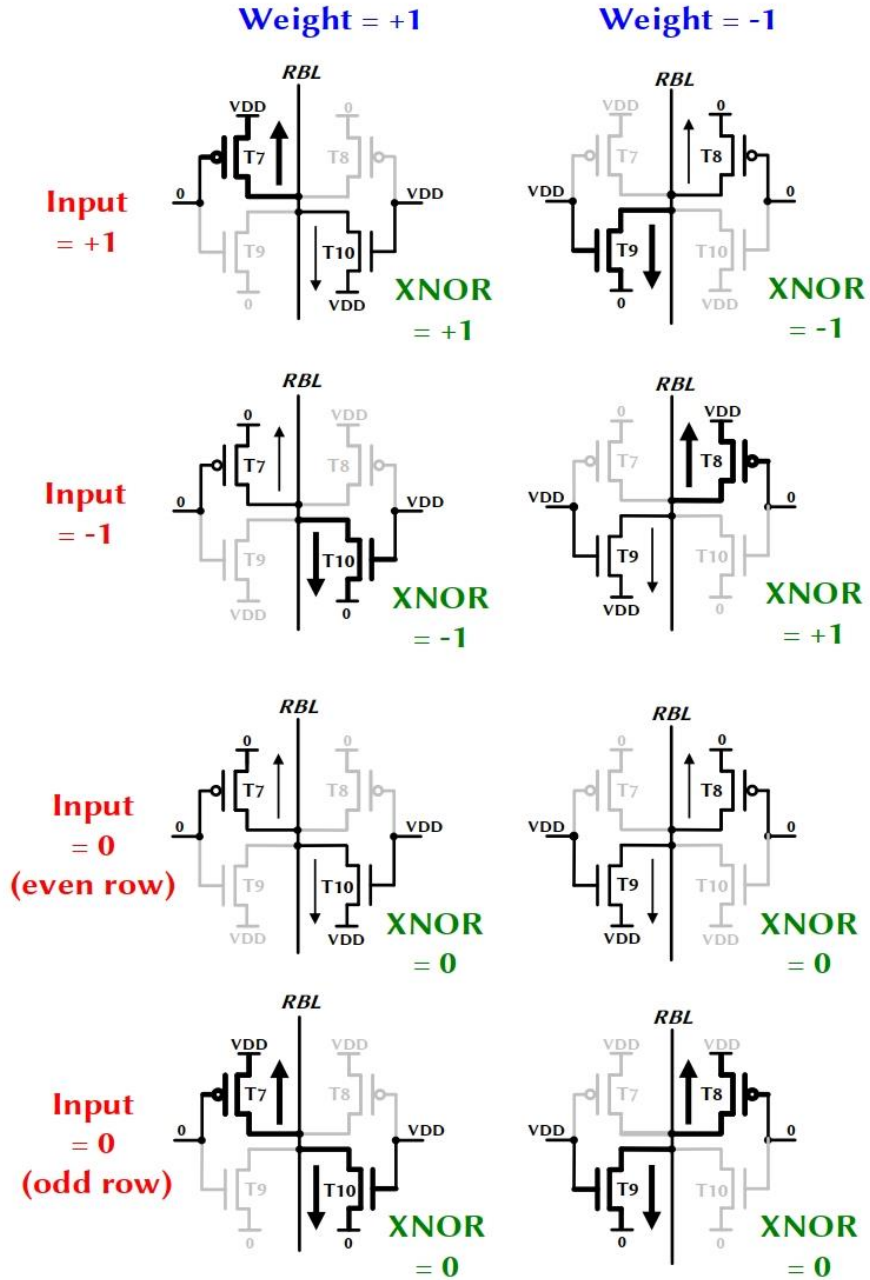
解决多位权重IMC设计难题的主要算法进步之一是二进制权重网络，其中网络权重被二值化，但是输入和输出激活可以保持多位。权重二值化放宽了存储限制，并使存储权重变得简单。BWN的子集称为二进制神经网络（BNN），将权重和激活值都二进制化为+1和-1，从而可以通过简单的XNOR运算来表示乘法。

Multi-Bit Activations in IMC Designs

Multi-bit activations可以在数字域或模拟域实现。数字域实现方式是通过multi-cycle操作，然后在数字域实现部分和的累加了，这种方式的代价是高延时和高能耗。模拟域的实现需要DAC (Digital-to-analog conversion)，而DAC又分为电压输出和电流输出。基于电压的DAC通常有精度限制的问题，基于电流的DAC电路原理是：首先通过脉宽调制(PWM)将input activation转化成脉冲信号，通过该脉冲信号将充电电流加到电容上。基于电流的DAC有两个主要的设计挑战：(1) the PWM needs to be linear; (2) the current source needs to be constant.

Compute in Current/Charge Domain

模拟MAC大致可分为两类：1）基于电阻分压器，放电速率等的电流域计算，以及2）基于电荷共享，电容分压器等的电荷域计算。参考文献[10]是基于电阻分压，如下图所示。该结构的代价是高的电流和器件不稳定性。参考文献[4]通过对WL冲/放电实现MAC操作，但是电流源是一个简单的transistor实现，具有非线性。为了在不同的情况下调节电流，需要额外的晶体管。


Weight = +1
Weight = -1
Input = +1
Input = -1
Input = 0 (even row)
Input = 0 (odd row)
RBL
VDD
T7
T8
T9
T10
XNOR = +1
XNOR = -1
XNOR = 0

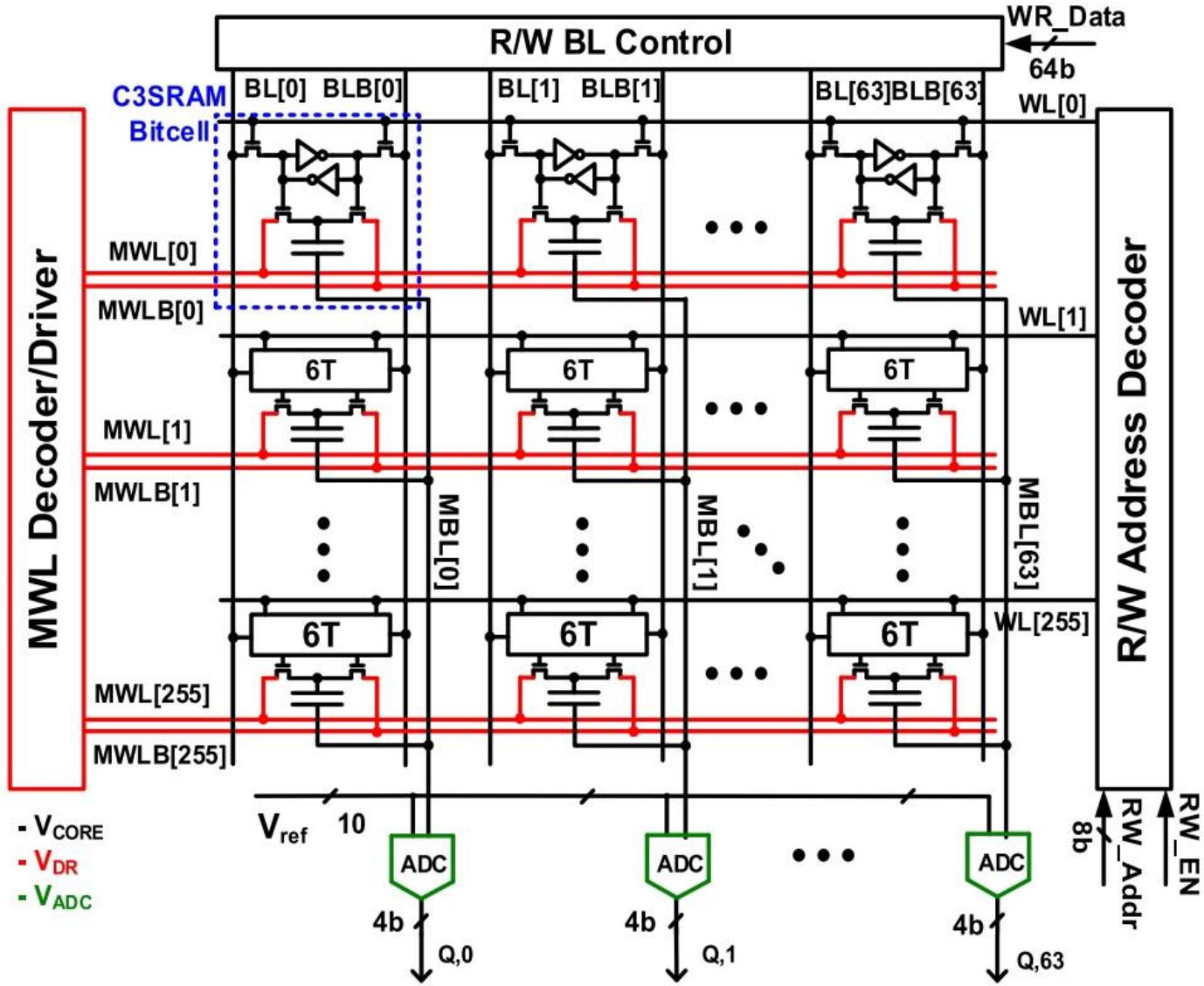
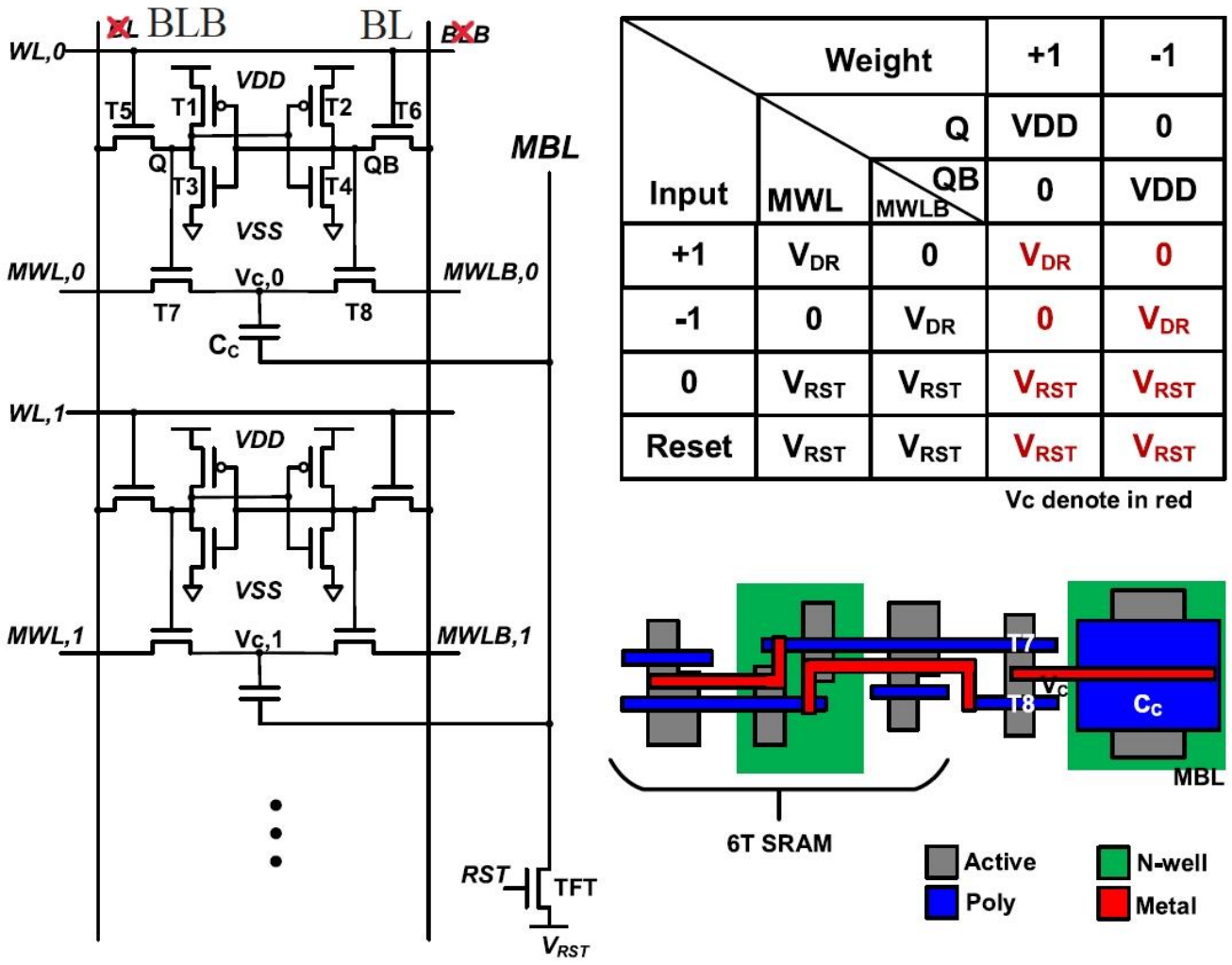


Fig. 1. Architecture of C3SRAM IMC macro.

8T1C比特单元的layout如Fig. 2所示，该比特单元比传统的6T SRAM的面积大80%。电容采用MOMCAP可以减小面积的overhead，但是本文为了实现更高的电容密度，采用的是MOSCAP，如果采用MOMCAP放置在C3SRAM的上方，电容值会减小80%。(本文的第一个瓶颈在这个地方，MOSCAP占整体面积的27%，如果能够通过更小的MOMCAP就可以实现功能，那么芯片的面积可以得到很大的优化。第二个瓶颈在与后面ADC电容限制了工作频率只能到50MHz。)



Input	MWL	MWLB	Weight +1	Weight -1
		Q	VDD	0
		QB	0	VDD
+1	V_{DR}	0	V_{DR}	0
-1	0	V_{DR}	0	V_{DR}
0	V_{RST}	V_{RST}	V_{RST}	V_{RST}
Reset	V_{RST}	V_{RST}	V_{RST}	V_{RST}

Vc denote in red

Fig. 2. C3SRAM bitcell design and in-cell bMAC operand table.

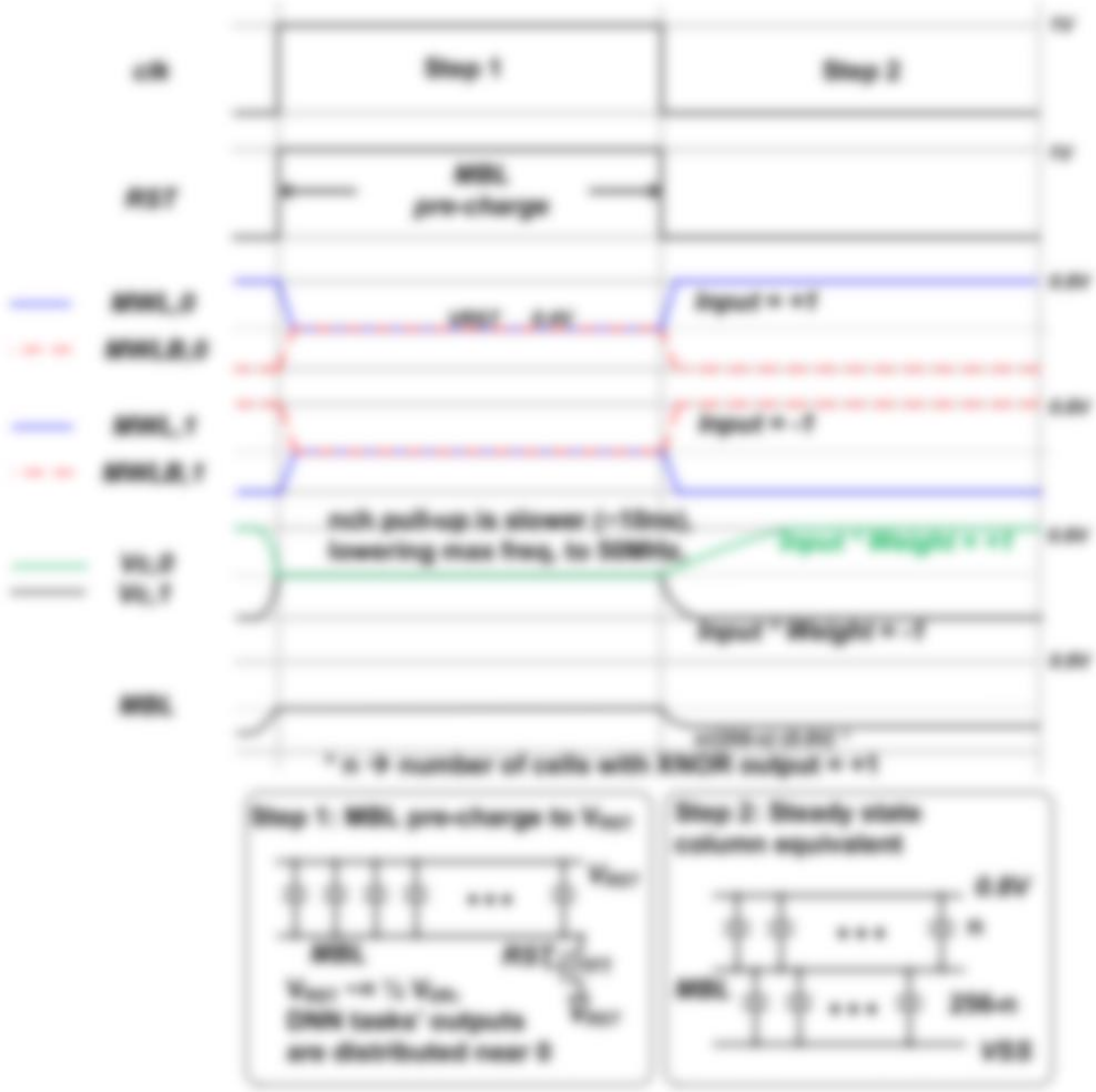
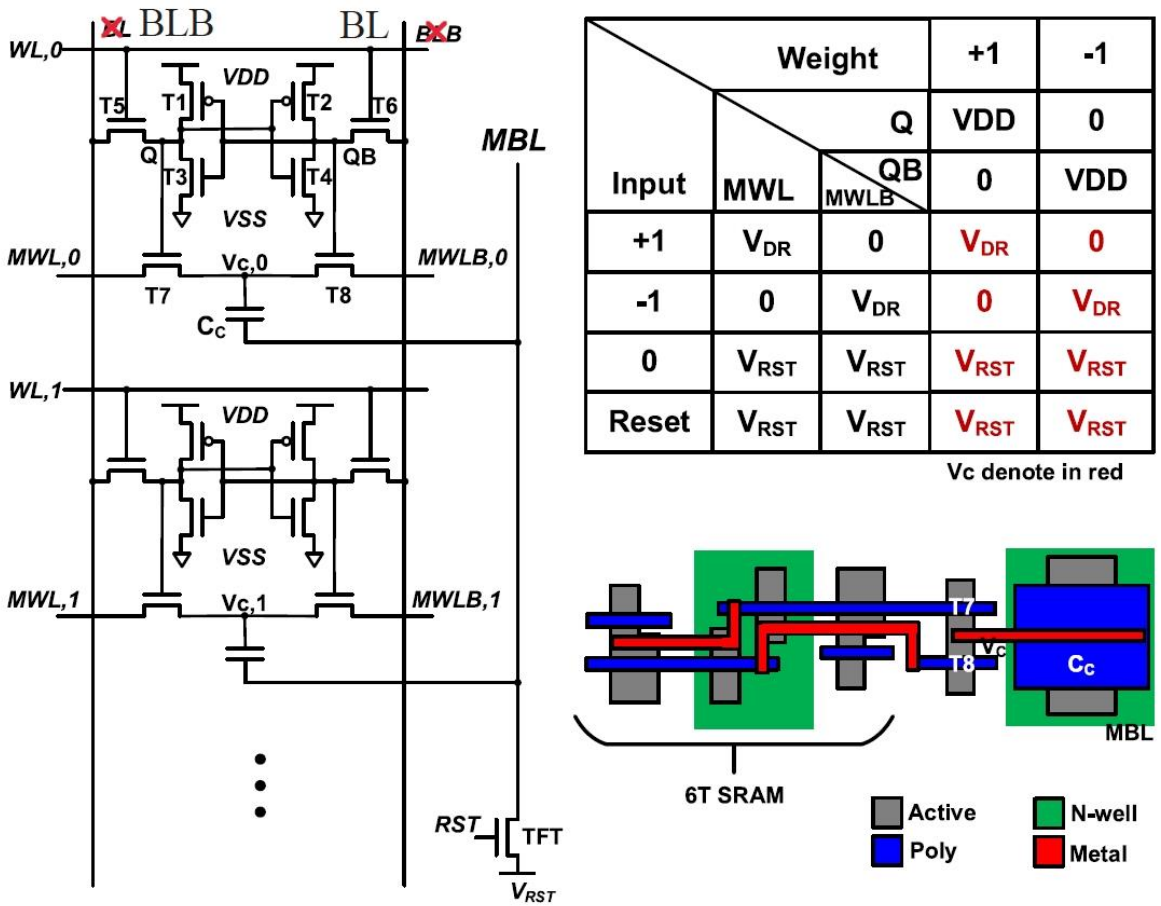


Fig. 4. Capacitive coupling based in-memory computation of bMAC.

C3SRAM cell实现点积的原理如下：电容通过pass transistors向MAC wordlins(MWL/MWLB)充放电，而PGs是通过SRAM的值进行门控。由于PGs都是NMOS，所以如果MAC wordlines和SRAM具有相同的电压源，T7和T8上就会存在V_{th}的压降。为了避免阈值电压variation的问题，该论文的PGs采用的是LVT器件，同时MWLs上的V_{DR}(0.8V)比V_{CORE}(1V)低200mV。通过Monte Carlo仿真的结果如Fig. 3所示。



Input	MWL	MWLB	Weight +1	Weight -1
		Q	VDD	0
		QB	0	VDD
+1	V_{DR}	0	V_{DR}	0
-1	0	V_{DR}	0	V_{DR}
0	V_{RST}	V_{RST}	V_{RST}	V_{RST}
Reset	V_{RST}	V_{RST}	V_{RST}	V_{RST}

Vc denote in red

Fig. 2. C3SRAM bitcell design and in-cell bMAC operand table.

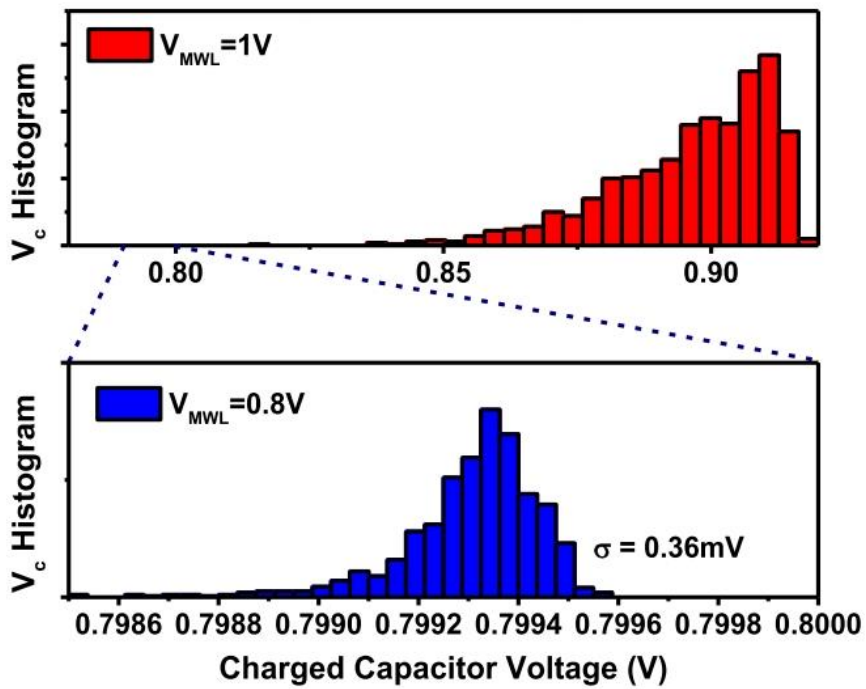
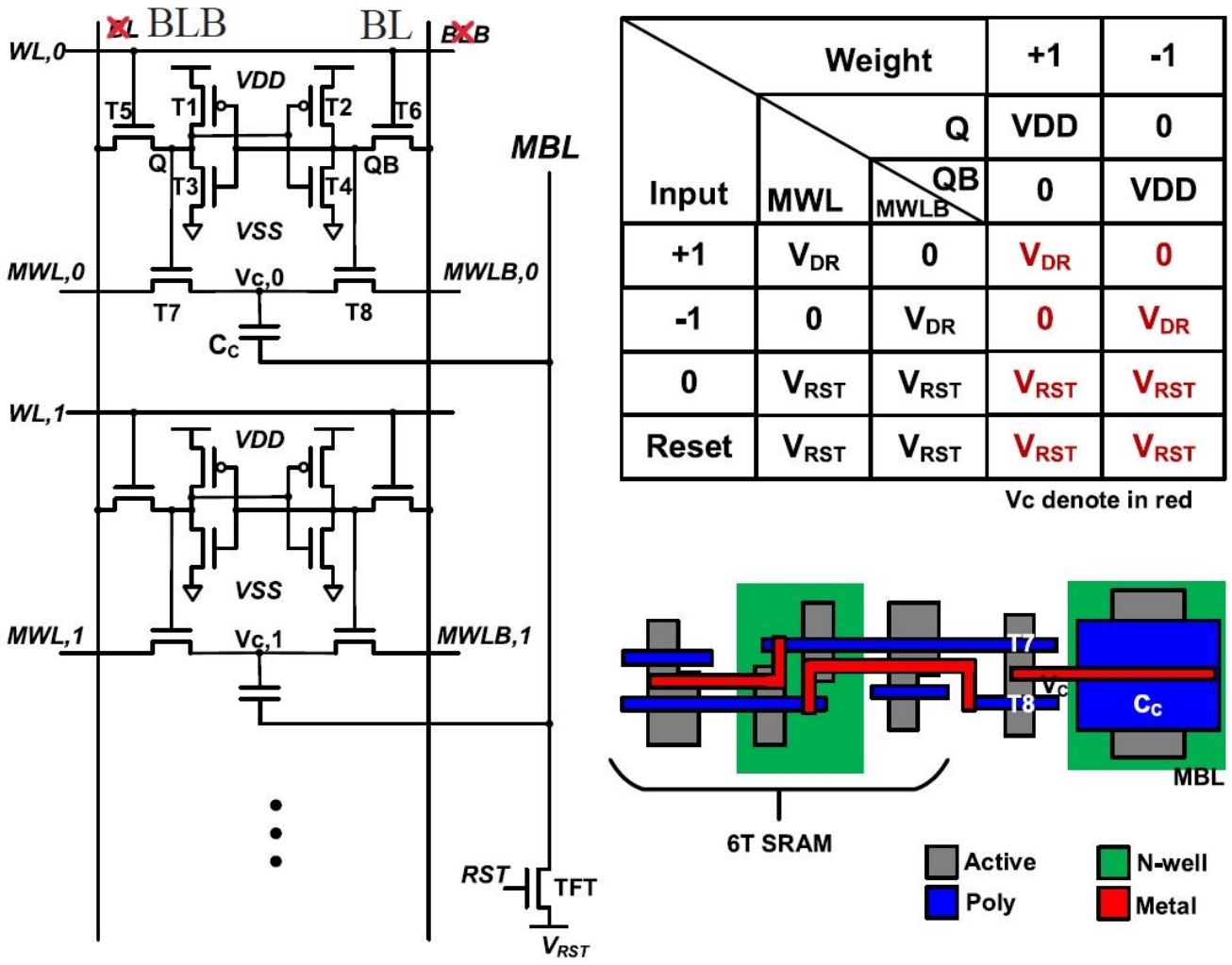


Fig. 3. Threshold voltage variability effects on charged capacitor voltage.



Input	MWL	MWLB	Weight: Q = VDD, QB = 0	Weight: Q = 0, QB = VDD
+1	V_{DR}	0	V_{DR}	0
-1	0	V_{DR}	0	V_{DR}
0	V_{RST}	V_{RST}	V_{RST}	V_{RST}
Reset	V_{RST}	V_{RST}	V_{RST}	V_{RST}

Vc denote in red

Fig. 2. C3SRAM bitcell design and in-cell bMAC operand table.

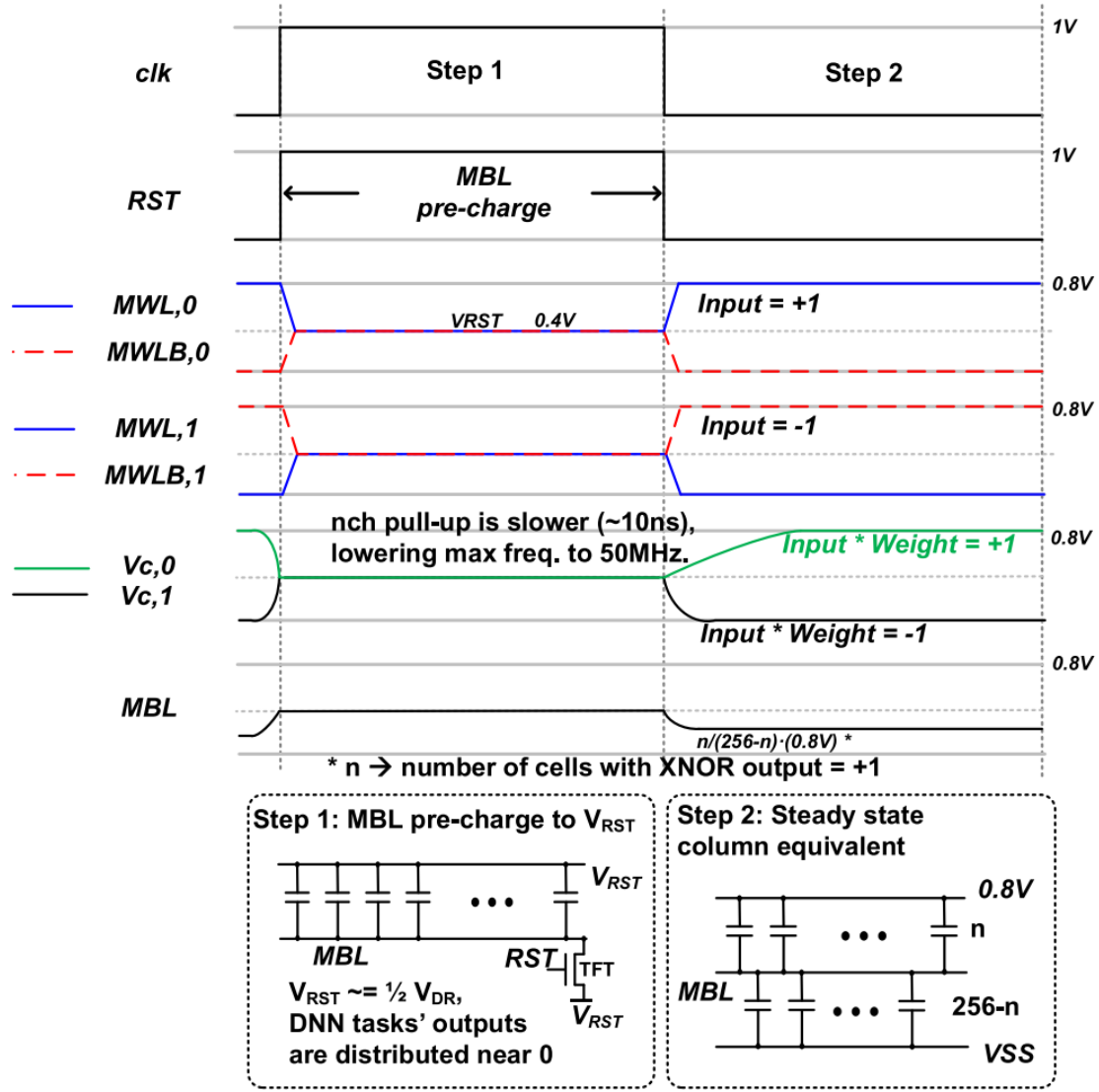
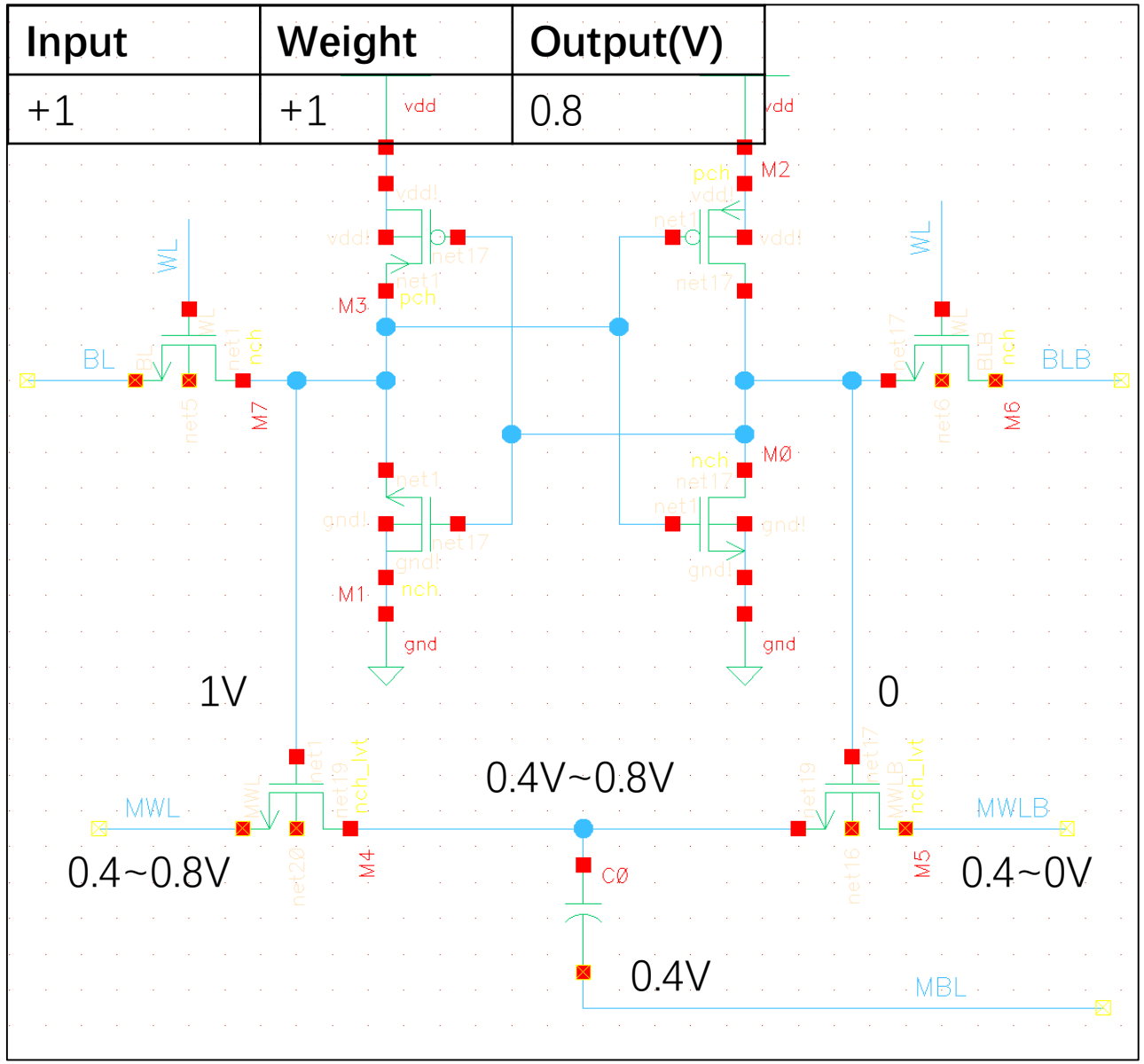
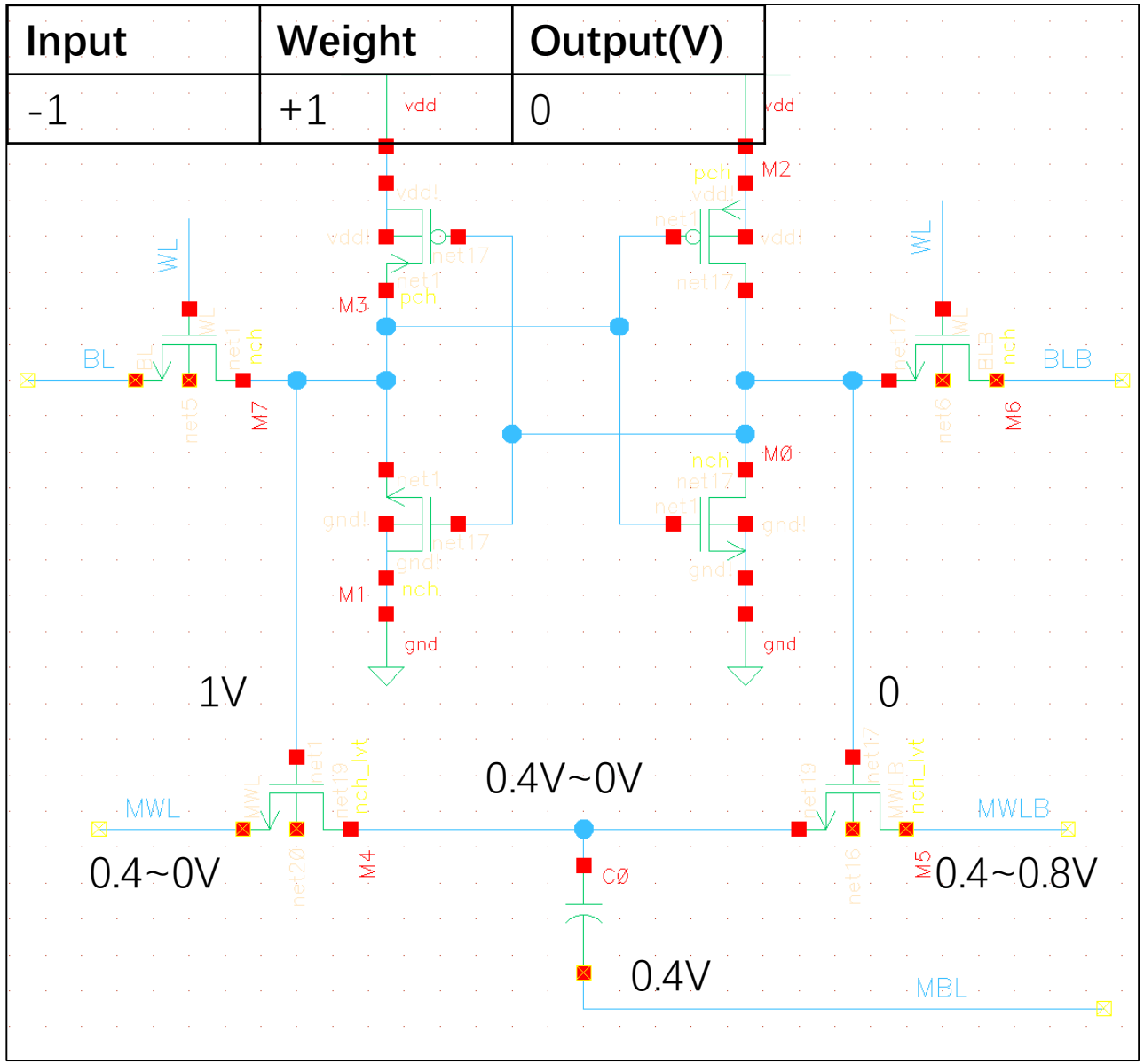


Fig. 4. Capacitive coupling based in-memory computation of bMAC.

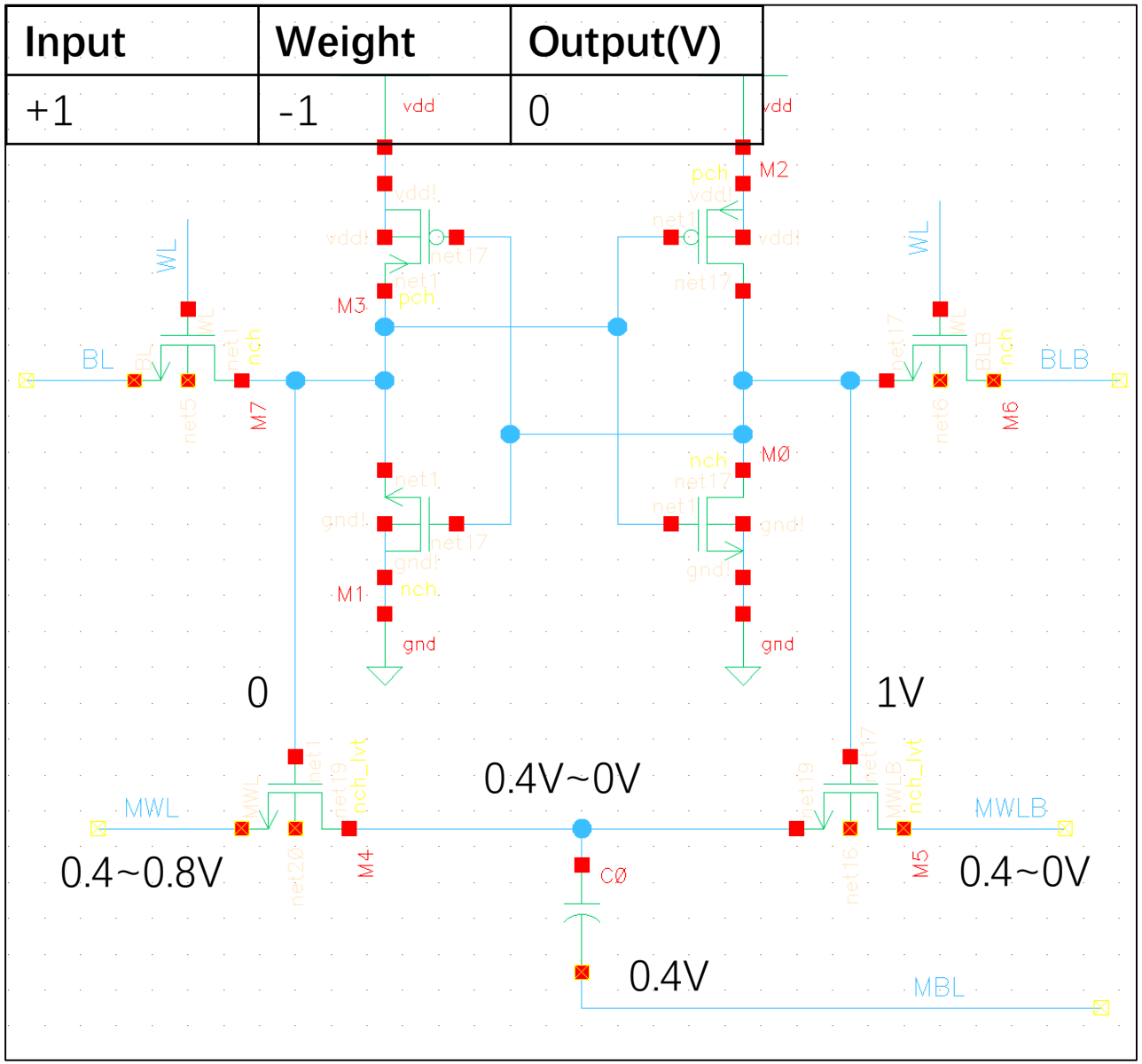
Input	Weight	Output(V)
+1	+1	0.8



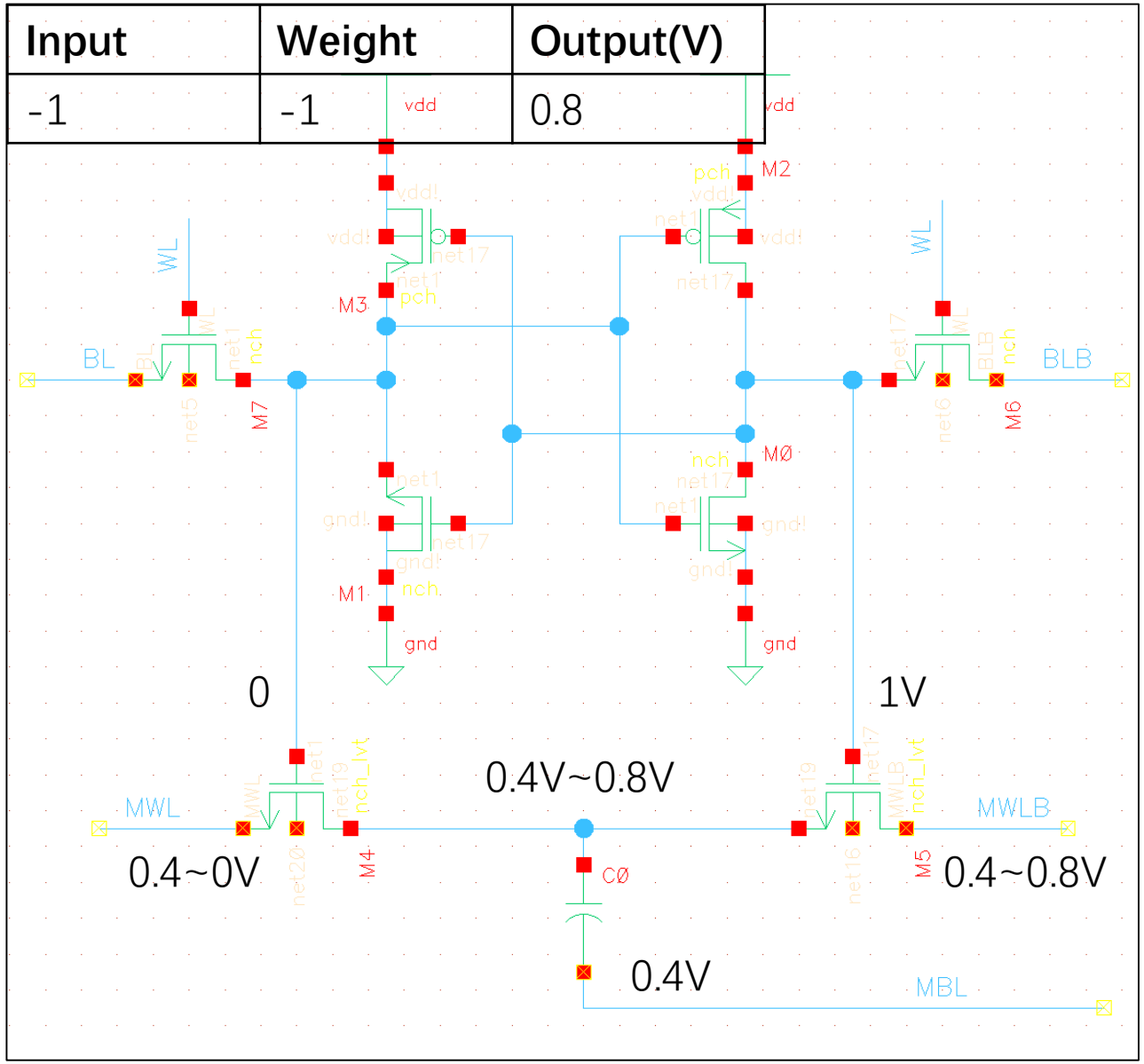
Input	Weight	Output(V)
-1	+1	0

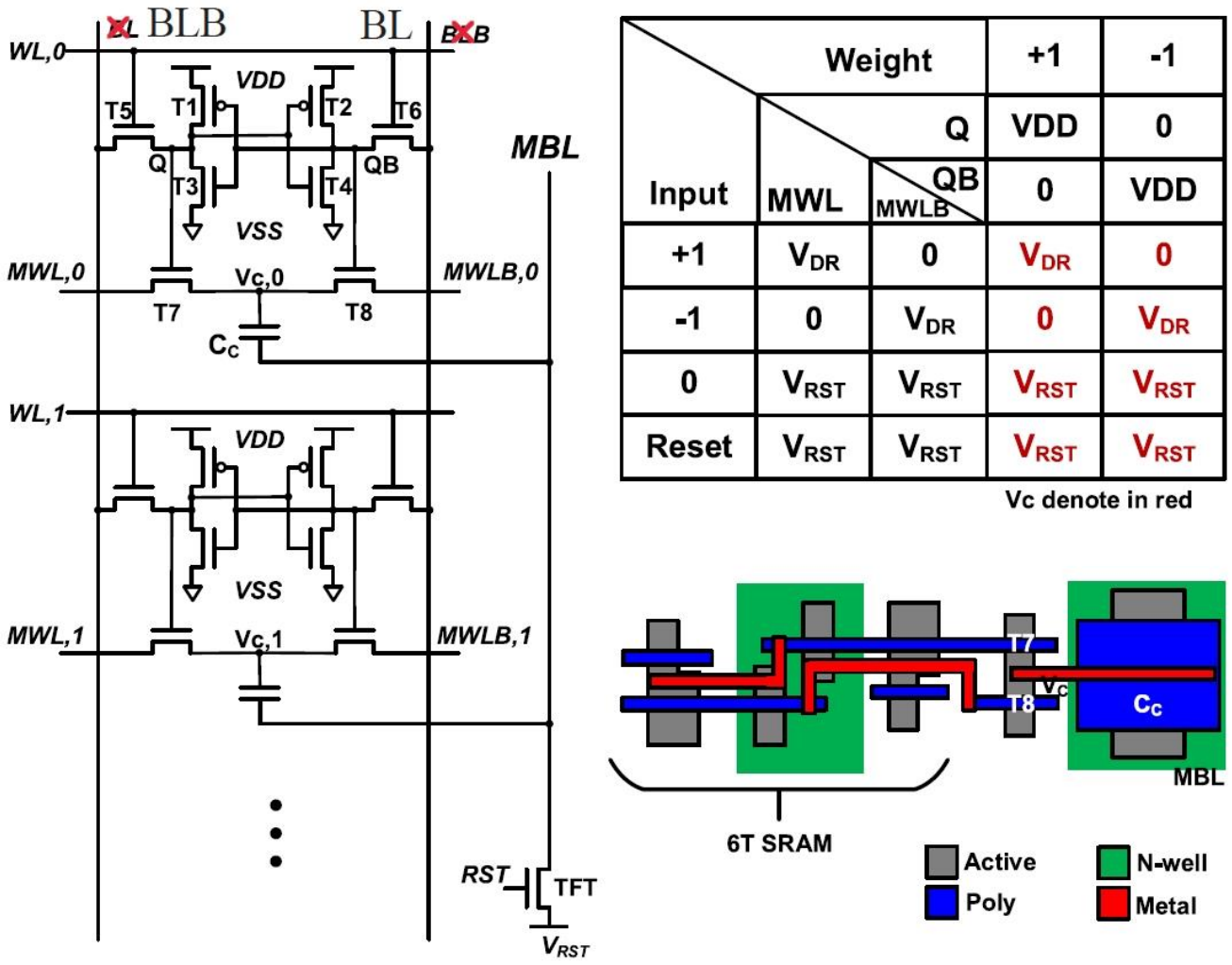


Input	Weight	Output(V)
+1	-1	0



Input	Weight	Output(V)
-1	-1	0.8





Input	MWL	MWLB	Weight: Q = VDD, QB = 0	Weight: Q = 0, QB = VDD
+1	V_{DR}	0	V_{DR}	0
-1	0	V_{DR}	0	V_{DR}
0	V_{RST}	V_{RST}	V_{RST}	V_{RST}
Reset	V_{RST}	V_{RST}	V_{RST}	V_{RST}

Vc denote in red

Fig. 2. C3SRAM bitcell design and in-cell bMAC operand table.

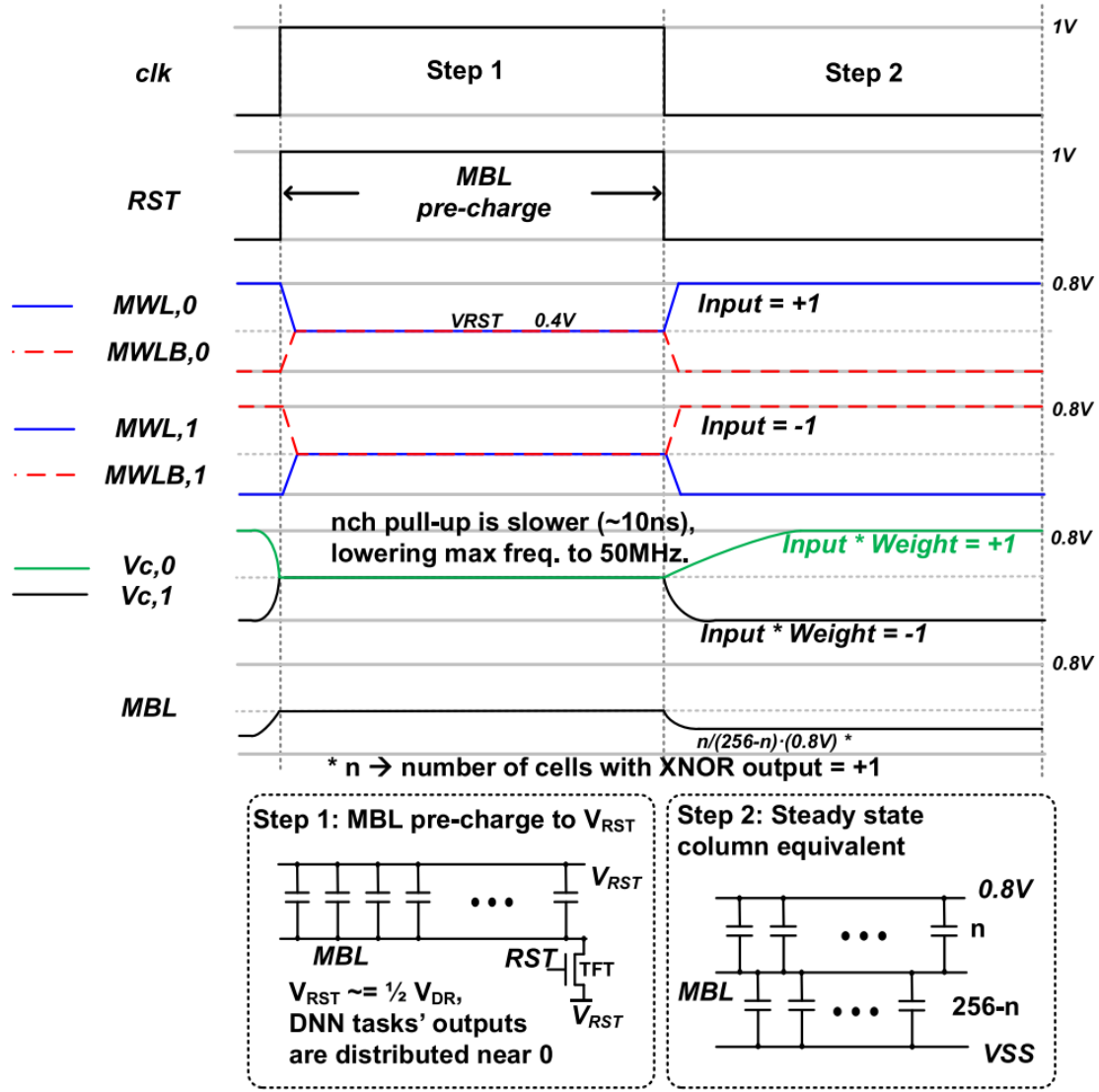


Fig. 4. Capacitive coupling based in-memory computation of bMAC.

- Step 1：每一列的MBL通过TFT预充到$V_{RST} = 0.5 * V_{DR}$，V_{RST}设置为接近bMAC输出为0的电压值，这么做的原因是在BNN中bMAC的输出分布在0附近很窄的区域，这样可以最小化MBL节点的电压摆幅。同时在这一步骤中，每一行的MWL和MWLB也接到V_{RST}的电压，这样bitcell的电容两端就没有电压差。此步骤结束，同一行的512个电容的情况如Fig. 4左下角所示。
- Step 2: TFT晶体管关闭，256根MWLs/MWLBs的电压根据对应的输入连接到V_{DR}或VSS，此时，对应Vc电压变化取决于input和weight如下图所示。电容两端的电压变化导致电容通过一个位移电流

$$I_C = C_C * \frac{dV_{Vc}}{dt}$$

(原文中Vc处是MWL(B)，这两者是等效的，Vc的电压变化等于两者之一的变化)，由此，bitcell到MBL的转移电荷为:

$$Q_{Ci} = \int_0^{t1} I_C dt = \frac{1}{2} C_C V_{DR}$$

其中t_1为V_{MWL}到达V_{DR}的时间，最终MBL的电压为：

$$V_{MBL} = C_C V_D R \sum_1^{256} \frac{(XNOR_i)}{(256C_C + C_p)}$$

其中XOR_i表示第i个XNOR输出的值，C_p表示MBL的寄生电容值加上ADC的输入电容值。在这个状态下，每列可以看成是两部分电容的串联，如Fig. 4的右下角所示。

从V_{MBL}的公式可以看出，C_C与C_p的比例愈大，bMAC的转移曲线的范围越宽，因此，MOSCAP的高电容密度特性能够实现一个wider full-scale range (FSR)的bMAC transfer curve。(从这边也可以看出降低ADC的输入电容对C_C的电容值要求也会变小，这样会降低电容带来的面积和功耗，而bMAC计算消耗的能量也就是电容的充放电。但是将ADC的输入电容减小对ADC的速度和精度不知道影响如何？)

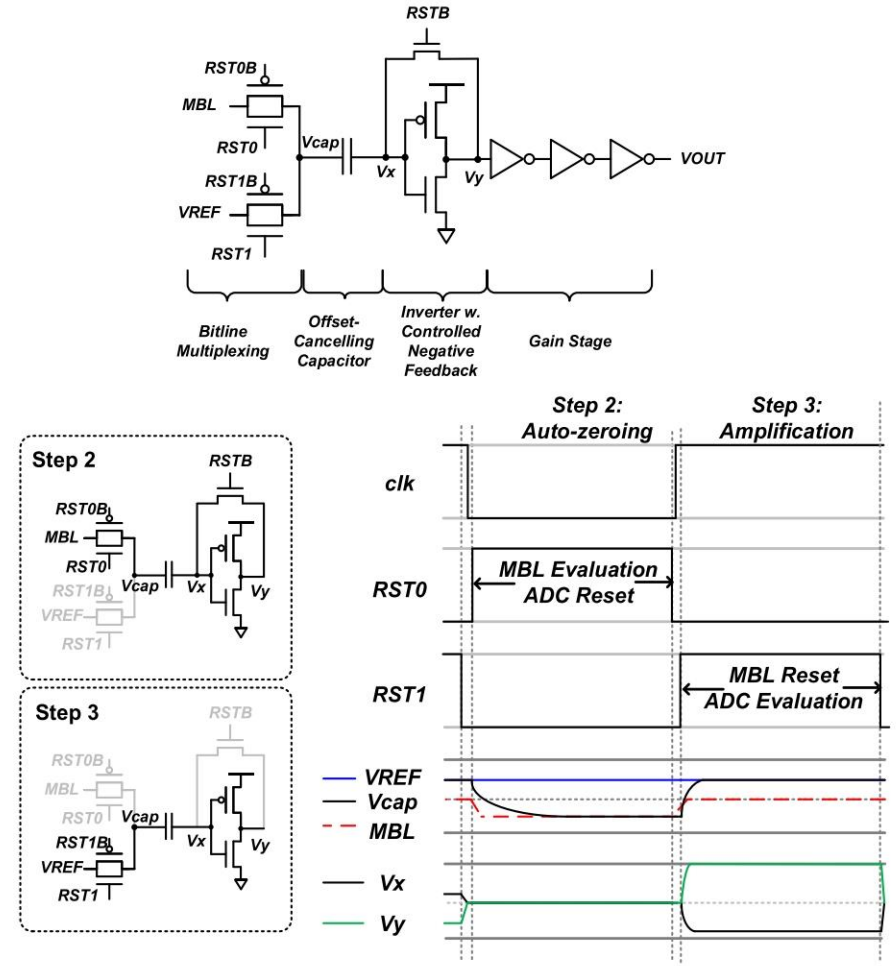


Fig. 6. Operation of the double-sampling self-calibrating single-ended comparator.

ADC的操作如Fig. 5所示，每一列对应一个11-level flash ADC。(每一列都有一个ADC，最终ADC的面积和功耗有没有很大的overhead，没搞懂为什么？为什么ADC的工作速度只做到了50MHz吗？但是flash ADC的速度不应该很快吗？)

Activation Bit Precision

对于工作在同一个IMC硬件上，BNN的推断精度损失要比BWN with multi-bit activation低。这源于两方面的原因：(1)多比特的模拟表示需要使用ADC；(2)使用高精度训练的网络模型对AMS error更敏感。对于第二个原因，该论文进行了实验验证，结果如Fig. 8所示。

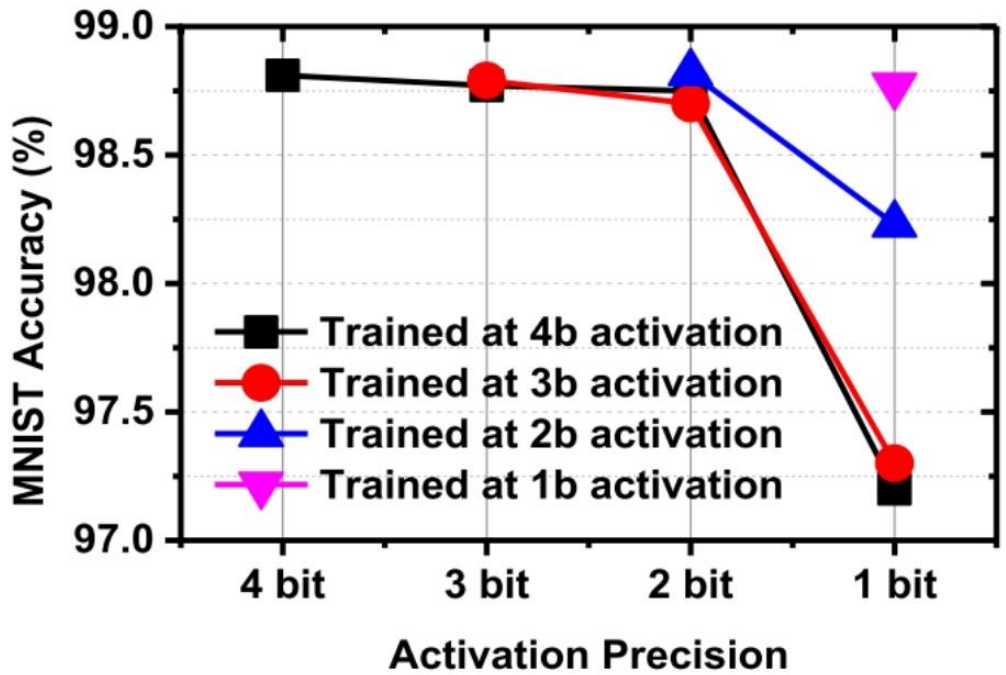


Fig. 8. MLP on the MNIST data set inference accuracy losses at various levels of activation precisions.

在实现5-bit精度的ADC的时候，作者还用了一个trick，由于权值集中在0附近，所以实现ADC的时候只实现了11-level，权值比较高的时候统一量化成2个level，这样$\Delta V_{ref} = 30mV$，对应0~0.8的范围差不多实现了5-bit精度。(如果这样做没有问题的话，CAM-ADC也可以这么做。)

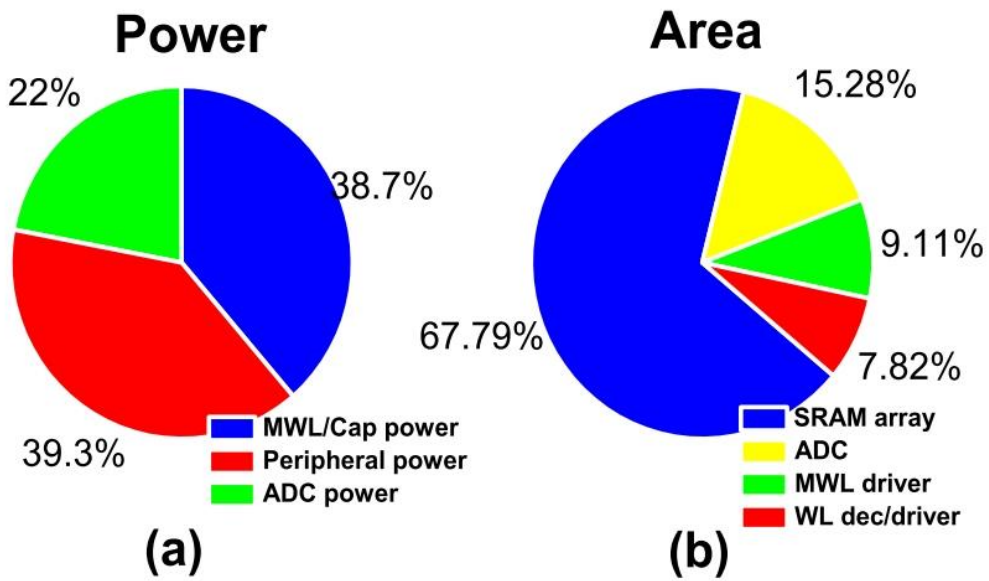


Fig. 13. (a) Measured power consumption breakdown between the three supplies powering bMAC compute (blue), partial sum accumulation (red), and ADC (green). (b) Area breakdown of the C3SRAM module.

Measurements and Analyses

该架构在65-nm CMOS工艺下实现，macro的容量为16kb(256 rows * 64 columns)，面积为0.081 mm^2。

Energy and Throughput

受ADC输入电容的限制，macro的最高工作频率为50MHz(这里是否可以改进).

$$Throughput = 2 \times 256 \times 64 \div 20ns = 1638GOPS$$

$$compute\ density = 1.638TOPS \div 0.081mm^2 = 20.2TOPS/mm^2$$

在不考虑input-output data movement的情况下，

$$energyefficiency = 2 \times 256 \times 64OPs \div 49pJ = 668.7TOPS/W$$

具体的能量和面积分布如Fig. 13所示。(为什么ADC的占比这么小？)

TABLE I

COMPARISON TO PRIOR IMC WORKS

	Biswas [22]	Si [25]	Valavi [26]	Jiang [30]	Si [31]	Gonugondla [45]	**This work**
Technology	65nm	65nm	65nm	65nm	55nm	65nm	**65nm**
Cell Type	10T	Split-6T	10T1C	12T	Twin-8T	6T	**8T1C**
Operating Voltage	1.2V (DAC)/ 0.8V (Array)/ 1V (rest)	1V	1V	0.6-1V	1V	1V	**1V (Array)/ 0.8V (Driver)/ 0.6V (ADC)**
Memory Capacity	2 kB	512 B	32 kB	2 kB	480 B	16 kB	**2 kB**
Input Precision	6	1	1	1	1-4	8	**1**
Weight Precision	1	1	1	1	2-5	8	**1**
Output Precision	6	1	1	5	3-7	4	**5** [1]
Efficiency (TOPS/W) [2]	40.3	30.49-55.8	658	403	18.37-72.03	6.25	**671.5**
Throughput (GOPS) [3]	8	1,112.8	589.9	665	84.8-269.6	8.26	**1,638**

[1] Margin equivalent to 11-level mid-range of 5-bit resolution.

[2] One MAC is counted as two operations (multiplication and addition).

[3] Consider an array size of 256x64 as unit capacity.

TABLE II

ACCURACY COMPARISON

	MNIST	CIFAR-10
Neural Network	MLP	VGG-like CNN
Network Topology[a]	784FC-512FC-512FC-512FC-10FC	128C3-128C3-MP2-256C3-256C3-MP2-512C3-512C3-MP2-1024FC-1024FC-10FC
Baseline Accuracy	98.7%	88.6%
Test Chip Accuracy	98.3%	85.5%

[a] *n*C*k* – *k*x*k* kernel convolutional layer with *n* filters, *m*FC – *m*-neuron FC layer, MP*p* – max-pooling layer with *p*x*p* pooling size.