



ISSCC2023: Background

工作对比:

22年: 12T SRAM cell+或非门乘法器的方案

23年: 更紧凑的8T SRAM cell+OAI的方案

Bitcell + Bit-flexibility:

- 总管子数量: 从28T减少到了24T
- 垂直方向上信号数量: 从9根减少到了6根
- ✓ 面积和走线上的提升

比特位数以4b为单位, 可组织成8b/12b/16b等更高的位数。

INT类型选择:

配合硬件上的量化/反量化, 在ImageNet数据集上, MobileNet_v2和ResNet-50上INT16可以展现出**非常接近FP16的精度**, 而INT8的精度掉点在1%-2%, 但是能效可以做到INT16的四倍。

- ✓ 相对折中的方案: **INT12**
- ∴这篇工作只做了8b和12b权重

软件: 浮点数

硬件: INT

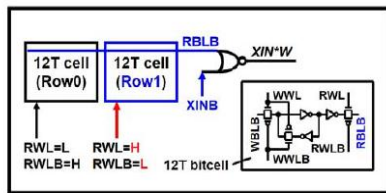
量化: 浮点——INT (四舍五入 (量化间隔? gauss分布 (概率分布密集时提高精度

硬件量化: INT8权重 + 输入 位宽膨胀 恢复

Bitcell selection for Digital CIM

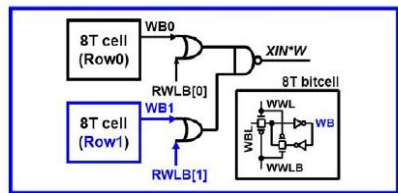
Latch based 12T bitcell + OR multiplier

- 28 Transistors
- 9 signals in vertical

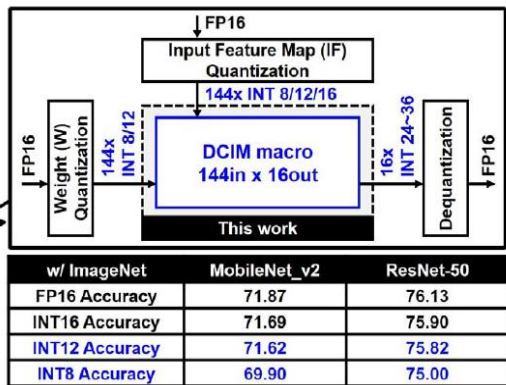
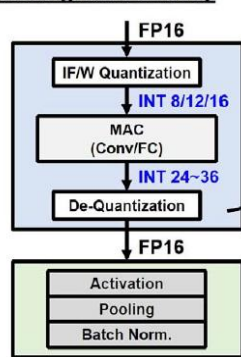


Latch based 8T bitcell + OAI (This work)

- 24 Transistors
- 6 signals in vertical



Supporting bit-flexibility



w/ ImageNet	MobileNet_v2	ResNet-50
FP16 Accuracy	71.87	76.13
INT16 Accuracy	71.69	75.90
INT12 Accuracy	71.62	75.82
INT8 Accuracy	69.90	75.00

硬件量化? 如何理解



ISSCC2023:整体结构

一个SRAM sub-array连接外部乘加电路的近存结构（与22年类似）

CIM Array

规模: $16 \times 144 \times 12 \times 2 = 54\text{Kb}$

共16个SRAM sub-array, 每个sub-array有144个bank/12b cell (三组4b cell) /2行

乘加电路

- **3个Local-MAC**: 出线时是 144×12 , 分到每个Local-MAC上的是 144×4
- **MAC内乘法器**: $1\text{b} \times 4\text{b}$ 结构, $144 \times 1 \times 2$ 根RWLB上将144个4b的XIN并转串后输入。Local-MAC有144个 $1\text{b} \times 4\text{b}$ 乘法器, 后面的加法树共 $\text{ceil}(\log_2(144)) = 8$ 级
- **Global IO**: 包含一套加法树, 处理三个Local MAC的求和 + 移位累加。

位宽配置

- **XIN**: 8b/12b/16b三种可选的位宽模式
- **权重**: 8b/12b两种位宽模式

8b权重:

两个Local-MAC进行组合, 第三个Local-MAC被disable以节省功耗 (一个NOR实现)
有符号8b时, 第二个Local-MAC的4b weight是Signed

12b权重:

使用三个Local-MAC组合, 有符号12b时, 第二个Local-MAC的4b weight是unsigned, 第三个Local-MAC的4b weight是signed

4个cycle算完4b MAC, 8b/12b/16b对应8 cycle/12 cycle/16 cycle。

为什么是2rows?

$RWLB = XINB$, bitcell本身含2roll

粗粒度流水线

第一级D触发器的作用?

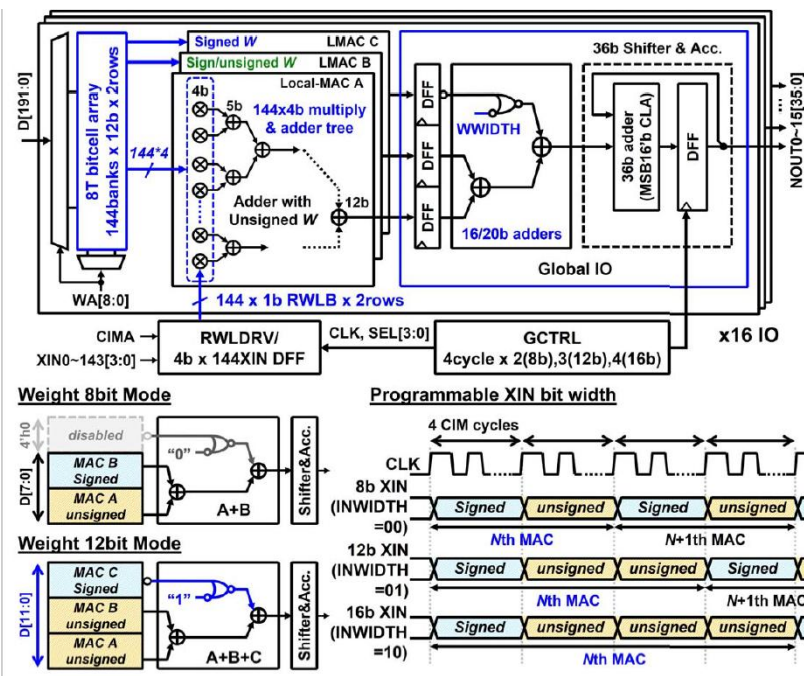
直接移位的操作 12b/16b/20b(4b一个bank)

提高吞吐量(流水线, MAC可以导入新的计算, stage1+stage2)

GCTRL的作用?

4cycle 和pipeline特性有关

根据位宽 改变累加的周期8c/12c/16c





- **管数**: 少掉读出的NOT和TG构成的4T, 只保留了写入时的8T
- **线数**: 控制线减少了3根, 只剩下WBL, WWL和WWLB
- 存储反相数据的WB则直接连到OAI上

OAI实现或非门乘法器 (21/22年)

RWLB: input输入线, 两row的两个cell每个配一根, 进行计算时一次只会用一个cell的数据, 另一个的RWLB始终拉高, 从而使OR的输出始终为1, 这时NAND门由于一个输入被固定为1, 功能相当于是**一个NOT**, 再和前面的**OR**组合, 实际上仍是一个**NOR**

36b有符号累加器

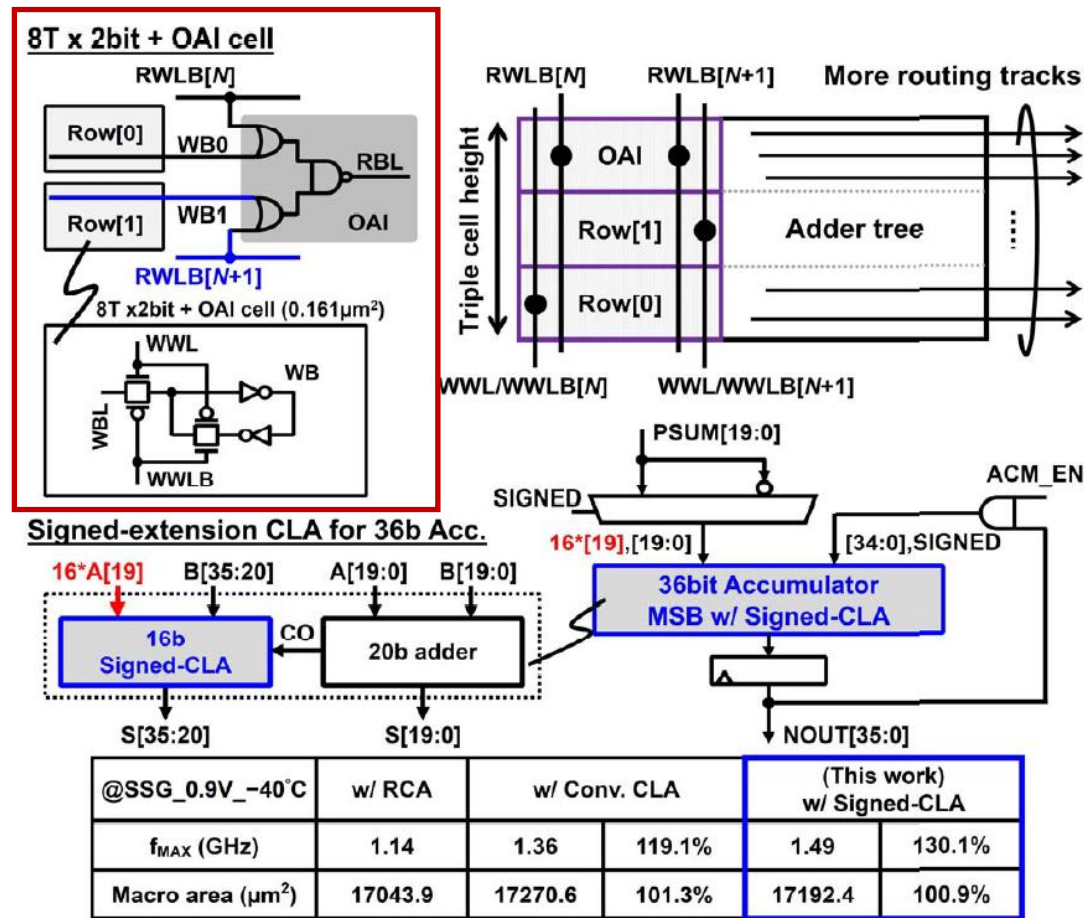
20b无符号adder + 16b有符号超前进位加法器的组合 (减少时延)

最终的累加器在每个循环中累加两个部分和：

- 一个部分和取自加法树(PSUM[19:0], 符号位扩展至36位)
- 一个部分和取自上一个循环后的一位移位操作(有符号NOUT[34:0])

减少水平方向上的走线负担:Triple cell height

把ROW[0], ROW[1]的Bit cell和OAI垂直的垒起来, 水平上有了更多的走线空间, 从而减小阻塞的问题



更快的速度和更小的面积
(Area~行波进位加法器, 30.1%速度提升)



ISSCC2023:补充

Floorplan: (延续22年)

- SRAM array和Local adder耦合在一起形成一个块
- 两个块share一个semi-global-adder
- 四个块share Global adder和移位累加器

Mix-Vt设计: 混用高Vt管和低Vt管来平衡速度和漏电

高Vt管漏电小但速度慢, 低Vt管漏电大但速度快

✓ 比纯低Vt管子可以减少38.6%的漏电

分配上, 在Stage1中高Vt和低Vt产生漏电是49.5%与50.5%, 基本对半分;

造成的延迟是36.7%与63.3%, 在延迟链上更多使用低Vt管来提高速度

总器件数量: 高Vt管76.2%, 低Vt管23.8%

Pipeline结构: 提高吞吐率

为了使得pipeline之间的延迟近似相等:

- Stage1: mix-Vt方案和使用长线 (线延迟更大), 减少了延迟链上的门数量, 从而获得更小的门延迟
- Stage2: 全用低Vt管子, 更短的线, 但有更多的门来增大门延迟

静态时序分析STA曲线: 两个Stage的延迟在不同电压域下保持基本相同, 最高在0.9V下可到1.5GHz频率。

