



# 南湖架构 Fudian FPU 设计与实现

---

李乾若、蔺嘉炜、何逸飞

中科院计算所

2022年8月25日

# 南湖FPU: fudian

- 弃用Berkeley HardFloat, 重新实现浮点功能部件 “fudian”
- <https://github.com/OpenXiangShan/fudian/>

运算	延迟
FADD	3
FMUL	3
FMA	5 (从乘法开始)
FDIV	$\leq 11$ (单精度) $\leq 18$ (双精度)
FSQRT	$\leq 17$ (单精度) $\leq 31$ (双精度)
FCVT, FCMP	3

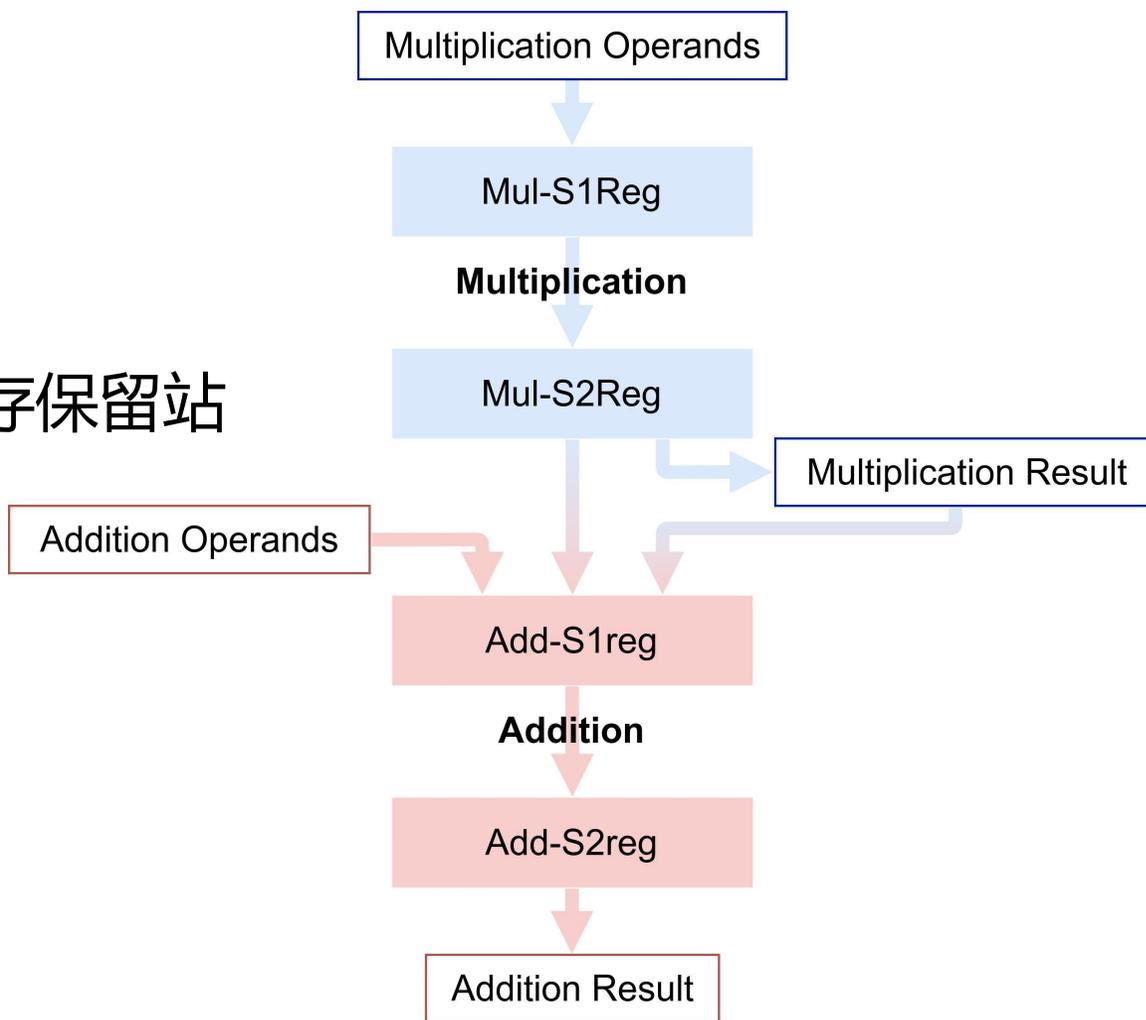
# 和HardFloat对比

南湖 (fudian)	雁栖湖 (Berkeley HardFloat)
<b>使用IEEE754格式</b> <ul style="list-style-type: none"><li>• 和IntBlock, MemBlock交互更友好</li><li>• 便于实现向量扩展</li></ul>	FloatBlock内部使用recoded格式: 每个浮点数指数扩展一位表示, 消除subnormal <ul style="list-style-type: none"><li>• 功能部件内部逻辑简单</li></ul>
<b>级联式FMA</b>	5级FMA
<b>SRT16除法器</b>	SRT4除法器
<b>SPECfp 20.96@2GHz*</b>	SPECfp 7.00@1GHz

\* Checkpoint估计值, DRAMSim3模拟DDR4-2400, O3编译选项

# 级联式FMA

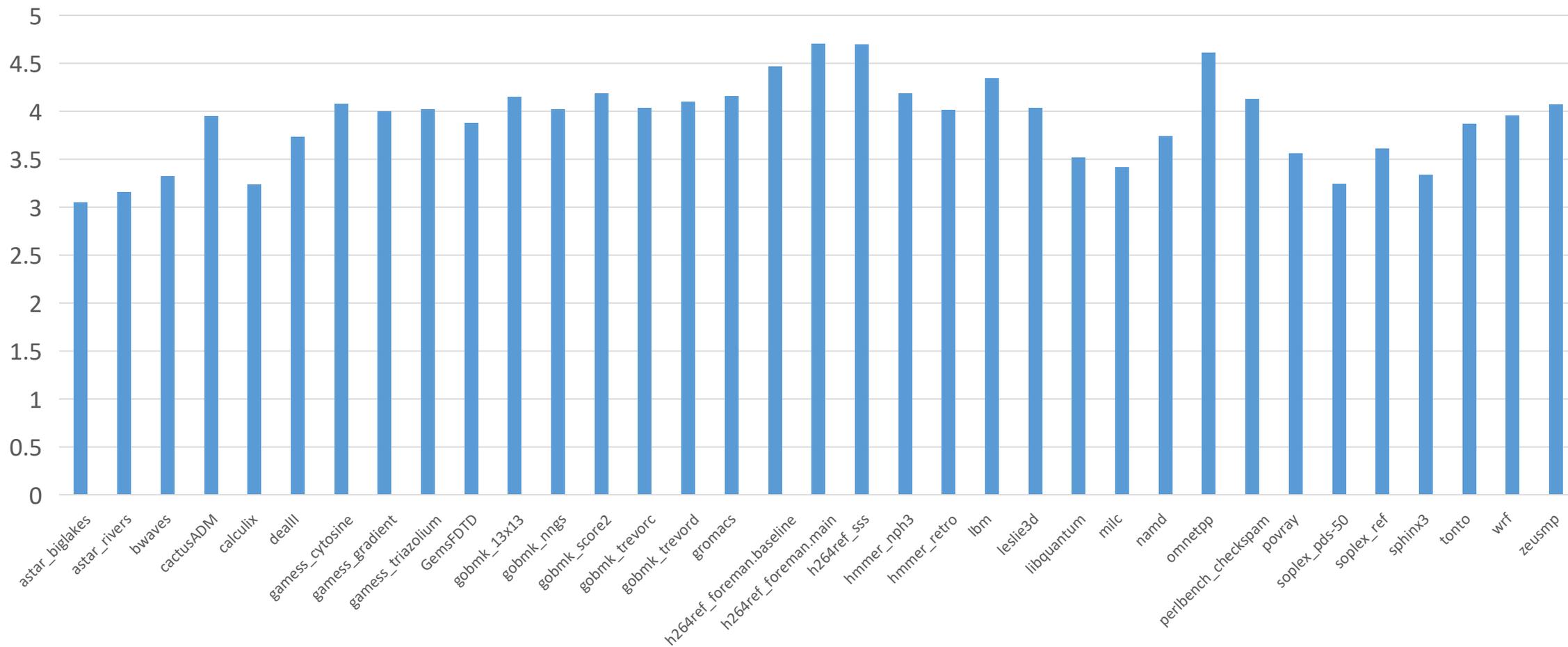
- 乘法、加法不再占用对方流水线
  - 5拍 -> 3拍
- FMA先发射乘法
  - 加法操作数S3进入FMA
  - 加法操作数未准备好，乘法结果暂存保留站
- FMA指令平均执行延迟3.9拍





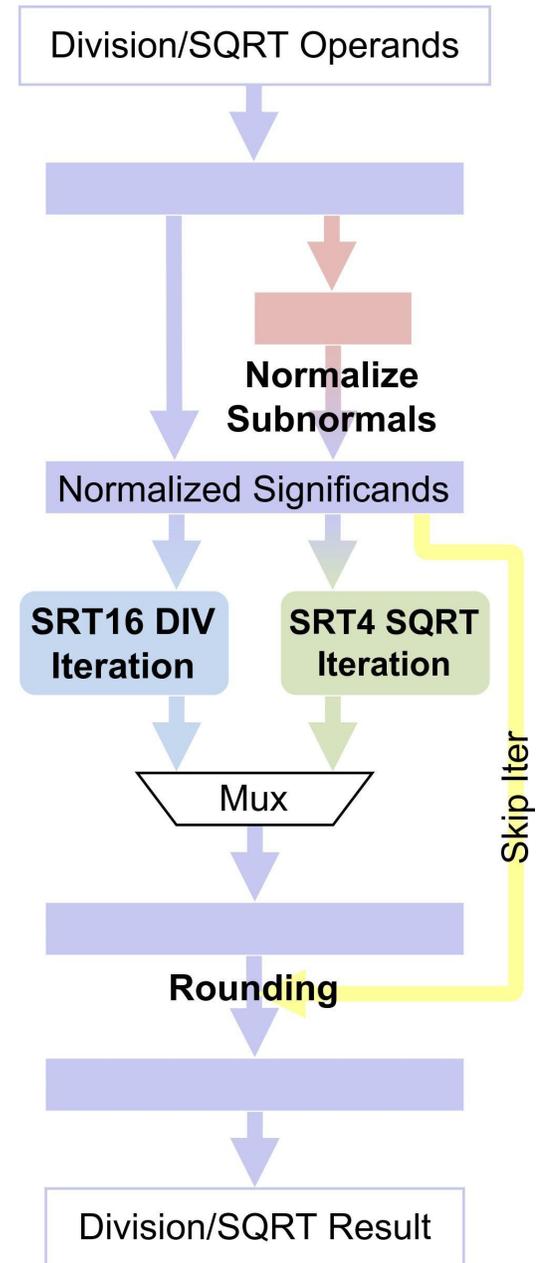
# FMA 平均执行时间

SPEC206 中 FMAC的执行时间 (Cycle)



# 🌟 浮点除法开方

- SRT16除法
  - 叠加两个SRT4商选择
- SRT4开方
- 共用Subnormal处理、fflag处理、修约等
- 独立的迭代模块
- 基于比较的商(q)选择
- Shift Over Zero



# 🌟 SRT16 迭代模块

- 每周期计算4位商( $q$ )
- 每周期进行两次 $q$ 选择
- 提前一轮计算一位 $q$ 来解耦 $q$ 与 $w$  (部分余数) 的计算
- 更窄的Mux, 减少一级CSA

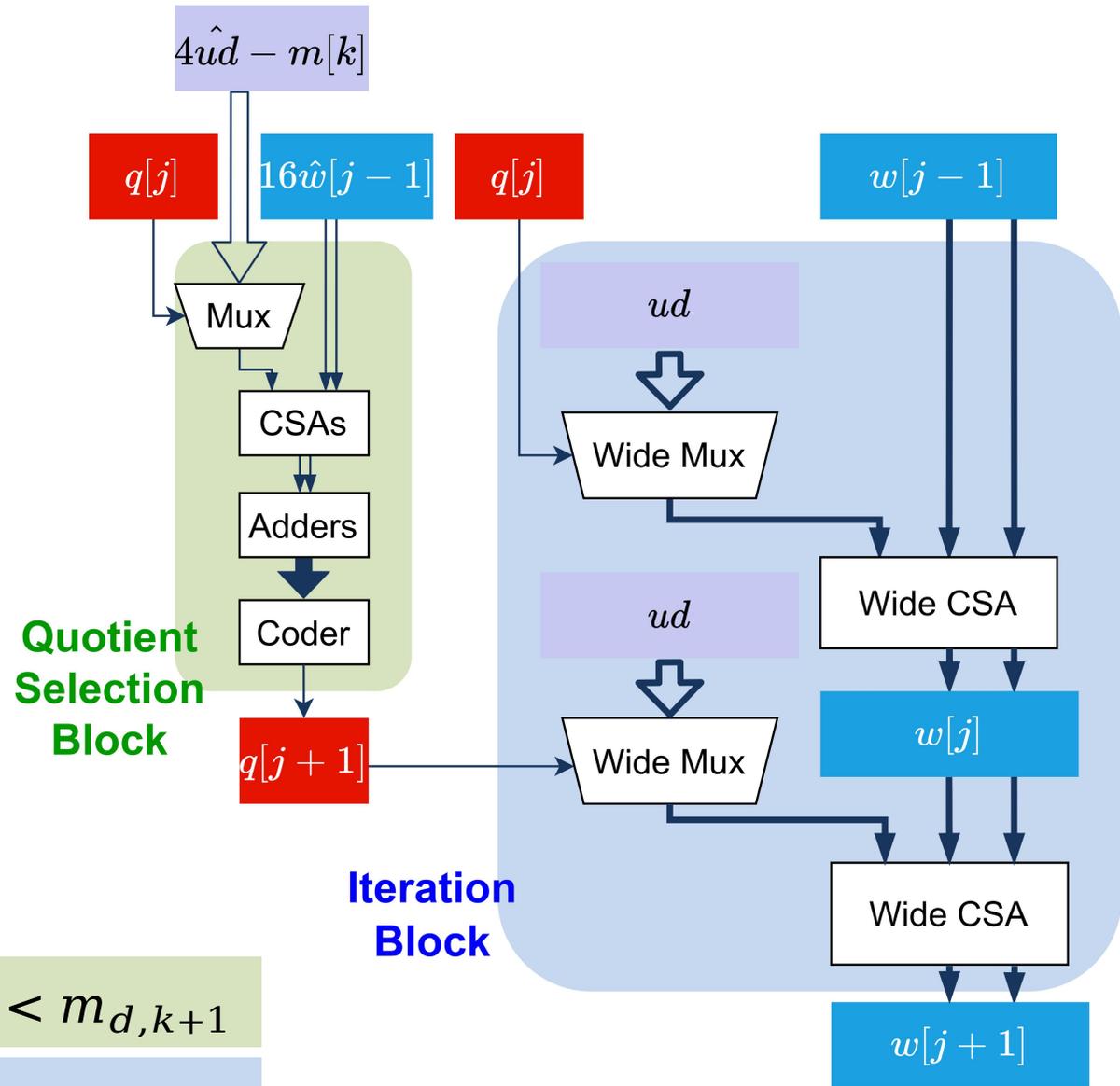
$$q[j+1] = k \text{ if } m_{d,k} \leq r\hat{w}[j] < m_{d,k+1}$$

$$w[j+1] = rw[j] - q[j+1]d$$

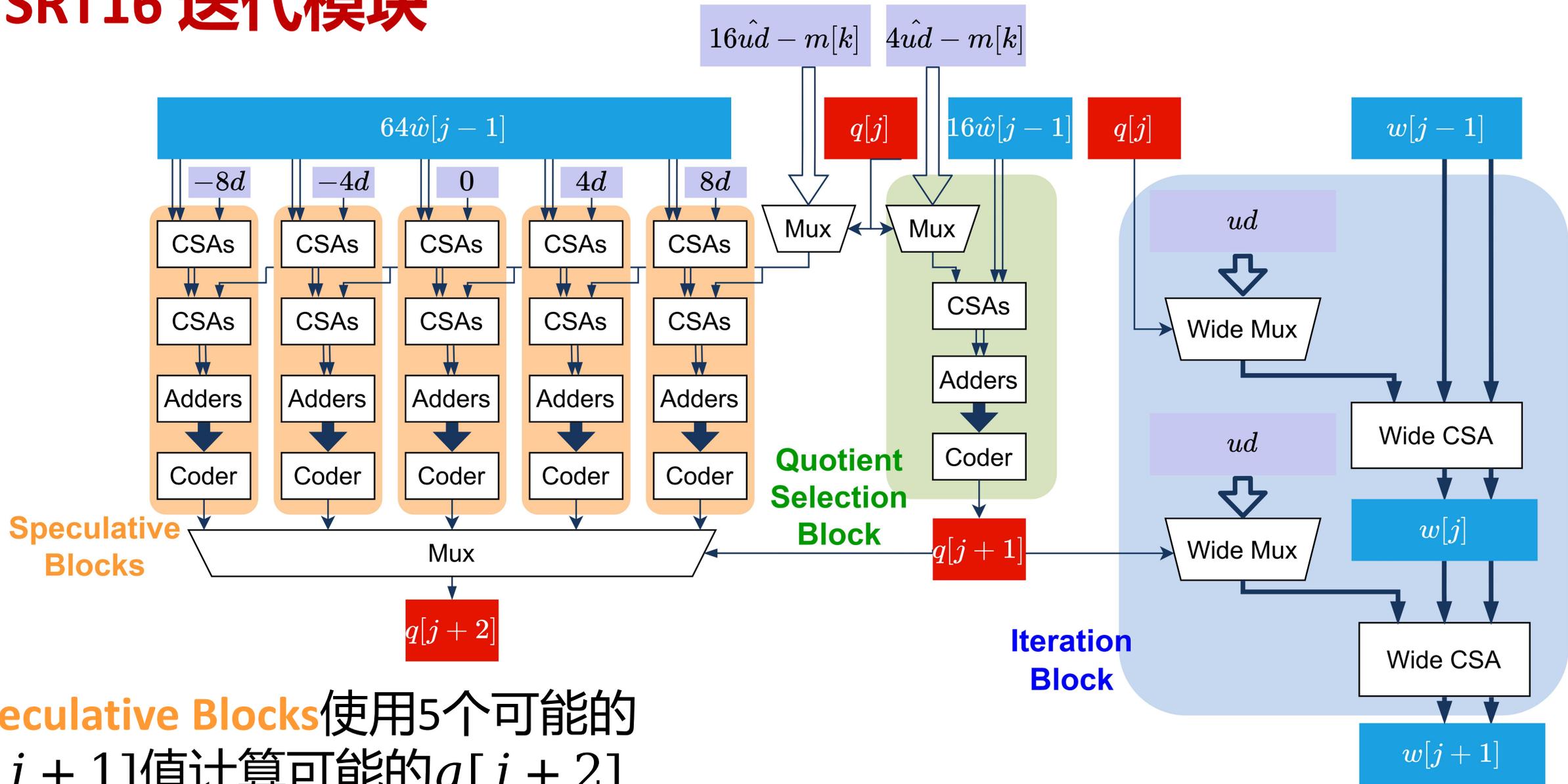


$$q[j+1] = k \text{ if } m_{d,k} \leq r^2\hat{w}[j-1] - rq[j]\hat{d} < m_{d,k+1}$$

$$w[j] = rw[j-1] - q[j]d$$



# SRT16 迭代模块



Speculative Blocks 使用5个可能的  $q[j+1]$  值计算可能的  $q[j+2]$

# 单元测试

- Berkeley Softfloat 和 Testfloat用于生成测试用例
- fudian目录下执行 `make all\_tests`

**敬请批评指正!**