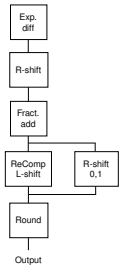
EE 486 lecture 14: Floating Point Considerations (ADD)

M. J. Flynn

A1: Baseline FADD

- Form exp difference,Δ
- Shift smaller char by Δ
- Characteristic add/sub
- Complement the result
- Post shift to renormalize
- round



A1: Baseline details

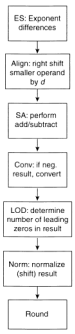
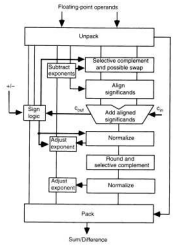


Figure 2.1: Floating point addition: algorithm A1.

Baseline FADD

- Needs 3 Adds
- And 2 long shifts
- Plus some selection and short shift overhead.



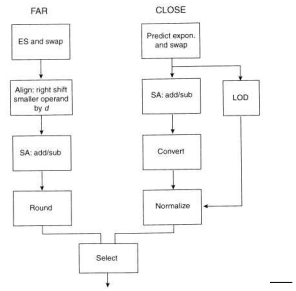
FP ADD improvements

- A2: the 2-path FP adder
- A3: the integrated round using compound adders
- Variable latency add

A2: the 2 path FP Adder

- The two long shifts can't occur with the same operands
- A long preshift occurs when one operand is much smaller than the other.
- A long postshift occurs on subtract when the operand value are close.
- So make two paths, a "far path" and a "close path".
- (M. Farmwald '81)

A2: The two path adder



A2: Two Path and rounding

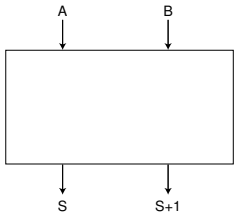
- Note that rounding can only occur in the far path or with no shift or one shift in the close path.
- In the close path, when a large postshift occurs then the result needs no round, as LGS = 000.

A3: Integrated round

- In the far path (or close and no big postshift) we can use a compound adder to produce sum and sum + 1. Then select the true result.
- Need to avoid the delay due to the rounding step.

A3: Compound Adders

- Produces sum and sum +1
- Uses same basic adder at the bit level (s,c,p,g) but needs a new look ahead structure.



A3: ADD/SUB rounding

Add/Sub – shifting

(A) Before N' L' G R S

(B) After L G S

(C) Final L – –

Problem: create action table based on (A) and corrections based on shift amount and sign bits that result in (C) as if it were determined by (B).

A3: Subtract/Add post shifts

- **Subtract Cases:**
- I. $|e_A - e_B| > 1$
- (a) no shift
- (b) left shift by 1.
- II. $|e_A - e_B| =< 1$
- (a) no shift: positive result
- (b) no shift: negative result
- (c) left shift by >= 1

Add Cases:

I. All $|e_A - e_B|$

(a) no shift

(b) one right shift

Rounding Cases, Subtract

- Rounding is not need when we have subtract in the close path and
  - the result is negative
  - the result is positive and a left shift is required
- An unadjusted (no shift) round occurs in cases I (a) and II (a)
- In case II (b) we have rounded in the L' rather than G' bit but it has the same effect.

Rounding cases, Add

- In case I (a), the unadjusted (no shift) rounds occurs.
- In case I (b), we have one right shift. Here we must look at the individual LGs combinations to select the correct outcome.

A3: actions Add, RN

Table 2.1: RTN action: Addition with one right shift; circled actions (C'_m) indicate critis for correct selection.

Before one right shift					After				
N'	L'	G'	$R'+s'$	Action'	L	G	s	Correct action (C_m)	
0	0	0	0	0	0	0	0	0	
0	0	0	1	0	0	0	1	0	
0	0	1	0	0	0	0	1	0	
(0	0	1	1)	1	0	0	1	0)	late select
0	1	0	0	0	0	1	0	0	
(0	1	0	1)	0	0	1	1	1)	late select
0	1	1	0	1	0	1	1	1	
0	1	1	1	1	0	1	1	1	
1	0	0	0	0	1	0	0	0	
1	0	0	1	0	1	0	1	0	
1	0	1	0	0	1	0	1	0	
(1	0	1	1)	1	1	0	1	0)	late select
(1	1	0	0)	0	1	1	0	1)	late select
(1	1	0	1)	0	1	1	1	1)	late select
1	1	1	0	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	

Action based on unnormalized result

RP and RM modes

- These modes require an S+2 as well as S and S+1.
- Needs another row of half adders, or an extra cycle when using these modes.

A3: Two path with integrated round

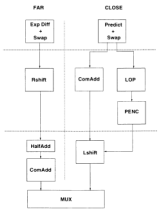


Figure 2.6: A version of algorithm A3.

A4: Variable Latency FADD

- Depending on operand values and what actions are needed we can provide a result in either one or two or three cycles.
- Value of this approach is limited by
  - Frequency of 1 and 2 ~ results
  - Scheduling difficulties in the instruction issuing mechanism.
  - Collisions in pipelined implementations.

VL Add: 2& 3 ~

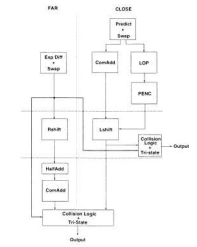


Figure 2.7: Two or Three Cycle Variable Latency Adder

VL Add: 2 & 3 cycle

- The far path requires the full exponent add, making it the slower path.
- So start the close path early (speculatively) and complete in 2~. If the true path was the far path, we'll know this at the end of the first cycle.

VL ADD: 1,2 or 3~

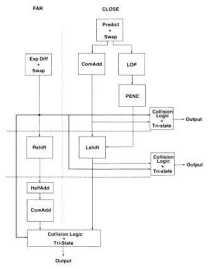


Figure 2.8: One, Two or Three Cycle Variable Latency Add

VL Add: 1,2 & 3 cycle

- Some operations may require neither a preshift nor a round.
- Even some one bit shifts might be included in a single cycle ADD/SUB
- Performance results depends on how aggressive we are in allocating actions to the first cycle.

Exponent differences and shifts

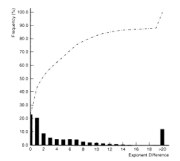


Figure 2.10: Histogram of Exponent Difference

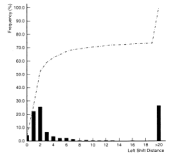


Figure 2.11: Histogram of Normalizing Shift Distanc

VL Performance

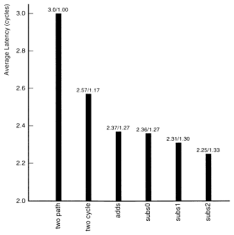


Figure 2.12: Performance Summary of Proposed Techniques