Conditional Speculative Decimal Addition*

Alvaro Vazquez and Elisardo Antelo
Dep. of Electronic and Computer Engineering
Univ. of Santiago de Compostela, Spain

*This work was supported in part by Xunta de Galicia under grant PGIDT03TIC10502PR.

Conditional Speculative Decimal Addition

Contents

- Introduction
  - Demand of High-Performance Decimal Arithmetic.
  - Revision of the IEEE-754 Standard for Floating Point.
- Previous Work on Integer Decimal Addition.
  - Basic Decimal Addition.
  - Direct Decimal Addition.
  - Speculative Decimal Addition.
- Proposed Method: Conditional Speculative Decimal Addition.
  - Algorithm.
  - Implementations: Parallel Prefix Adders.
    - Binary Carry Tree: Kogge-Stone and Ladner-Fischer.
    - Quaternary Carry Tree.
- Delay-Area Estimations and Comparison.
  - Delay-area model for Static CMOS gates based on Logical Effort.
- Conclusions.

Conditional Speculative Decimal Addition

Contents

- Introduction
  - Demand of High-Performance Decimal Arithmetic.
  - Revision of the IEEE-754 Standard for Floating Point.
- Previous Work on Integer Decimal Addition.
  - Basic Decimal Addition.
  - Direct Decimal Addition.
  - Speculative Decimal Addition.
- Proposed Method: Conditional Speculative Decimal Addition.
  - Algorithm.
  - Implementations: Parallel Prefix Adders.
    - Binary Carry Tree: Kogge-Stone and Ladner-Fischer.
    - Quaternary Carry Tree.
- Delay-Area Estimations and Comparison.
  - Delay-area model for Static CMOS gates based on Logical Effort.
- Conclusions.

Introduction

Demand of High-Performance Decimal Arithmetic.

- Need of hardware support: financial and e-commerce applications.
- **Binary floating-point** units introduce **inaccurate results**.
- Decimal **software** implementations do **not satisfy** performance **demands**.
- Reduced hardware support (IBM S/390 series:G4,G5,G6,z900, z990).
- Only usual decimal integer operations improved in hardware.
- Does not exist hardware implementations of decimal floating-point .

Introduction

Revision of the IEEE-754 Standard for Floating Point.

- Current draft revision of IEEE-754 incorporates specifications for decimal arithmetic.
- Decimal formats: 32-bit, 64-bit and 128-bit.
- Sign, exponent and significand (DPD encoding ➔ BCD).
- Rounding modes and exception handling for binary and decimal.
- Conversions between integer and floating-point formats.
- Operations defined: Add, subtract, multiply, fused multiply-add, divide, square root….
- Software or/and Hardware implementations.

Conditional Speculative Decimal Addition

Contents

- Introduction
  - Demand of High-Performance Decimal Arithmetic.
  - Revision of the IEEE-754 Standard for Floating Point.
- Previous Work on Integer Decimal Addition.
  - Basic Decimal Addition.
  - Direct Decimal Addition.
  - Speculative Decimal Addition.
- Proposed Method: Conditional Speculative Decimal Addition.
  - Algorithm.
  - Implementations: Parallel Prefix Adders.
    - Binary Carry Tree: Kogge-Stone and Ladner-Fischer.
    - Quaternary Carry Tree.
- Delay-Area Estimations and Comparison.
  - Delay-area model for Static CMOS gates based on Logical Effort.
- Conclusions.

Previous Work on Integer Decimal Addition

Basic Integer Decimal Addition (High-level)

- Inputs:

$$A = \sum_{i=0}^{d-1} A_i \bullet 10^i \qquad B = \sum_{i=0}^{d-1} B_i \bullet 10^i \qquad C_0 = C_{in}$$

- Output:

$$S = \sum_{i=0}^{d-1} S_i \bullet 10^i \qquad S_i = \mathrm{mod}_{10}\left(A_i \pm B_i + C_i\right)$$

- Basic decimal carry propagate recurrence:

$$C_{i+1} = \left\lfloor A_i \pm B_i + C_i \,/ 10 \right\rfloor$$

- Subtraction ➔10's complement of subtrahend:

$$-B = 10^d + \sum_{i=0}^{d-1} \left(9 - B_i\right) \bullet 10^i + 1$$

Previous Work on Integer Decimal Addition

Basic Integer Decimal Addition (BCD)

- Digits represented in BCD-8421 (4-bits/digit).
- Subtraction :

$$(9 - B_i) = 15 - (B_i + 6) = \overline{B_i + 6}$$

- Basic Addition/Subtraction in BCD:

$$B_i^* = \begin{cases} \overline{B_i + 6} & if\ (op == sub) \\ B_i & else \end{cases} \qquad S_i^* = \mathrm{mod}_{16}\left(A_i + B_i^* + C_i\right)$$

$$C_{i+1} = \left\lfloor (S_i^* + 6)/16 \right\rfloor \qquad S_i = \begin{cases} \mathrm{mod}_{16}\left(S_i^* + 6\right) & if\ (C_{i+1} == 1) \\ S_i^* & else \end{cases}$$

- **Problem**: carry chain ➔ Improve delay of carry propagate recurrence

Previous Work on Integer Decimal Addition

Direct Decimal Addition

- Uses the following decimal carry recurrence:

$$C_{i+1} = G_i + \overline{K_i} \bullet C_i$$

G_i → decimal carry generate.

K_i → decimal carry kill.

G_i **true** when $A_i+B_i>10$
K_i **true** when $A_i+B_i<9$

- Can be evaluated using conventional parallel carry evaluation techniques:
  - Carry lookahead.
  - Parallel Prefix.
- Quaternary carry tree: 1 decimal carry per 4-bits.

Previous Work on Integer Decimal Addition

Direct Decimal Addition

- G_i and K_i can be expressed in terms of binary $g_i[j]$ and $k_i[j]$:

$$G_i = G_i^* + \overline{K_i^*} \bullet g_i[0] \qquad \overline{K_i} = \overline{K_i^*} \bullet \overline{k_i[0]} \qquad \begin{matrix} i = 0,\ldots,d-1 \\ j = 0,1,2,3 \end{matrix}$$

$$G_i^* = g_i[3] + g_i[2] \bullet g_i[1] + \overline{k_i[3]} \bullet \left(\overline{k_i[2]} + \overline{k_i[1]}\right)$$

$$K_i^* = \overline{k_i[3]} + g_i[2] + \overline{k_i[2]} \bullet g_i[1]$$

- $g_i[j]$ and $k_i[j]$ are the inputs of the quaternary carry tree.

- G_i and K_i are evaluated using specific logic and a logic level of the quaternary carry tree.

- Implemented in the FXU of the G4, G5 and G6 IBM S/390 series.

Previous Work on Integer Decimal Addition

Implementation of Direct Decimal Addition

Performs binary and direct decimal additions/subtractions.

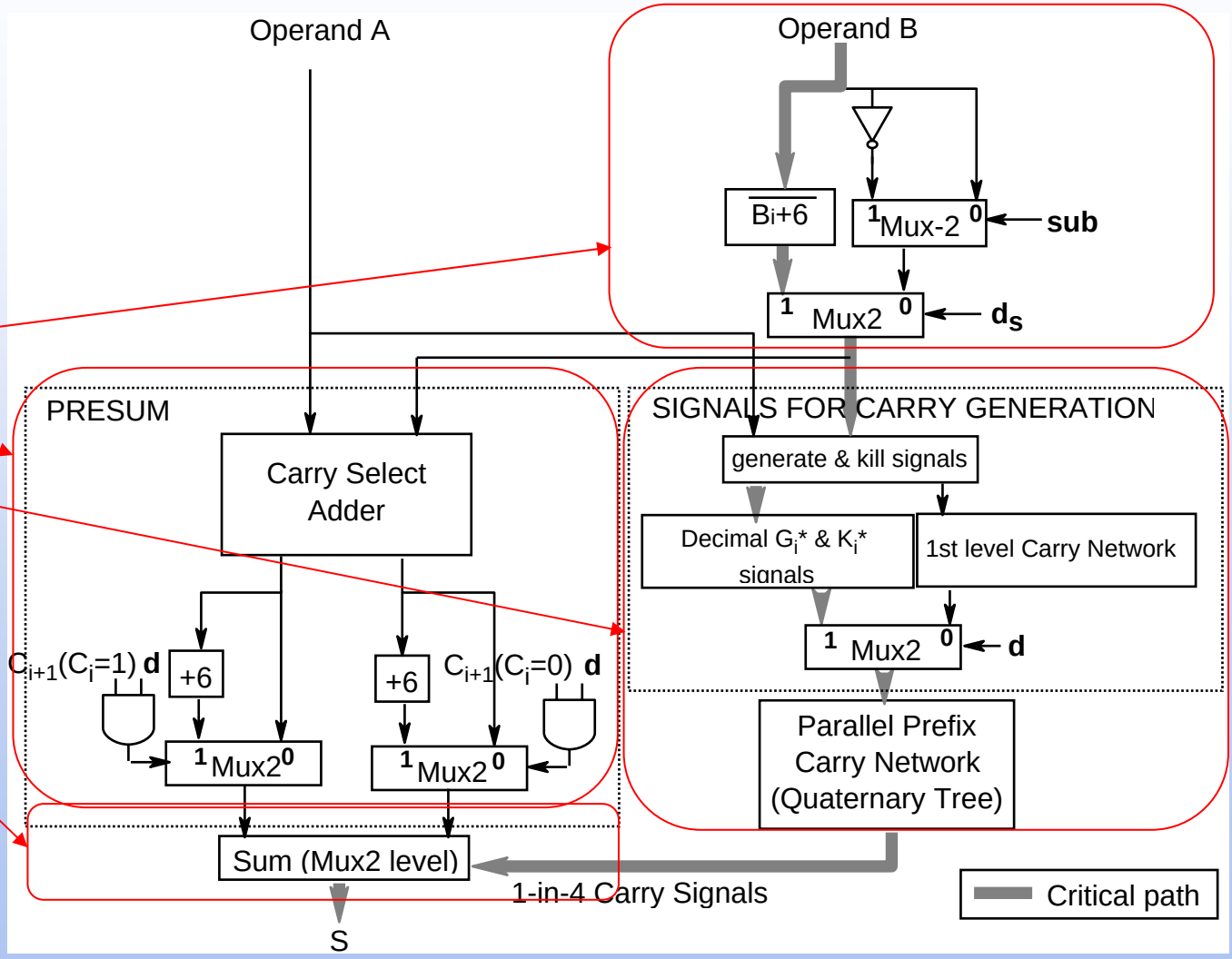
STAGES

- •Operand setup.
- •Pre-sum
- •Carry-evaluation (pre-carry and carry tree).
- •Sum.

Quaternary carry tree (sparse tree, 1-in-4 carries).

Evaluation of decimal carry-generate and carry-kill signals.

Sum performed using 4-bit carry-select adders plus a digit addition of 6.



Previous Work on Integer Decimal Addition

Speculative Decimal Addition

Speculative Addition/Subtraction characteristics:

- Initial (unconditional) sum of input digit+6 (without carry propagation).

$$S_i^* = \mathrm{mod}_{16}\left(A_i + B_i^* + 6 + C_i\right)$$

- Binary carries of S_i^* at decimal positions = decimal carries ➔ allows binary parallel carry evaluation techniques.

$$C_{i+1} = \left\lfloor S_i^* / 16 \right\rfloor$$

- Final correction of S_i^* -6 (without carry propagation).

$$S_i = \begin{cases} \mathrm{mod}_{16}\left(S_i^* - 6\right) & if\ (C_{i+1} == 0) \\ S_i^* & else \end{cases}$$

Previous Work on Integer Decimal Addition

Speculative Decimal Addition

- Two possibilities for the evaluation of S_i^*:
  1. Using a parallel prefix carry tree.
     - XOR operation + post-correction (after carry evaluation).
  2. Using a quaternary carry tree (sparse).
     - 4-bit carry select adders + correction (in parallel with carry evaluation).

- Several choices for initial sum +6➔ similar implementations.

$$\left(A_i + B_i^* + 6\right) = \begin{cases} (A_i + 6) + B_i^* \\ A_i + (B_i^* + 6) \\ (A_i + 3) + (B_i^* + 3) \end{cases}$$

- Implemented in the FXU of the IBM z900 and z990.

Previous Work on Integer Decimal Addition

Implementations of Speculative Decimal Addition

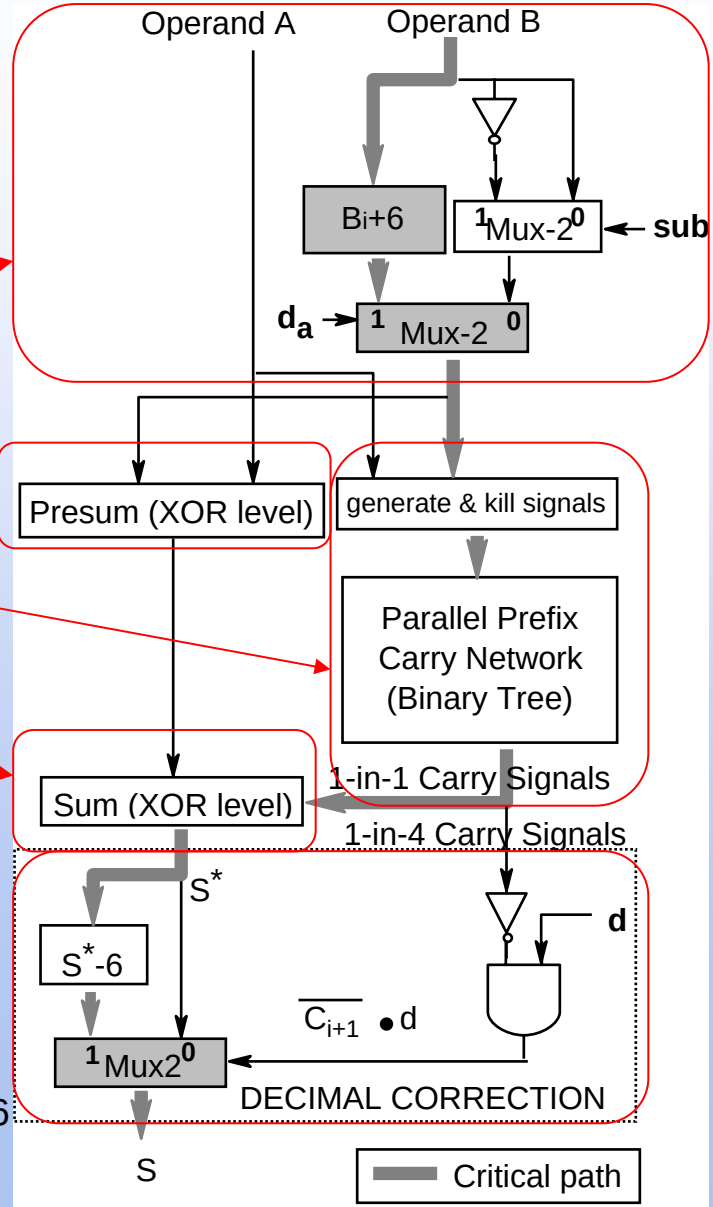
Performs binary and speculative decimal additions/subtractions.

STAGES

- •Operand setup.
- •Pre-sum
- •Carry-evaluation (pre-carry and carry tree).
- •Sum.
- •Post-correction.

Binary carry tree (Kogge-Stone, Ladner-Fischer, etc…)

Needs post-correction in the critical path.



Previous Work on Integer Decimal Addition

Implementations of Speculative Decimal Addition

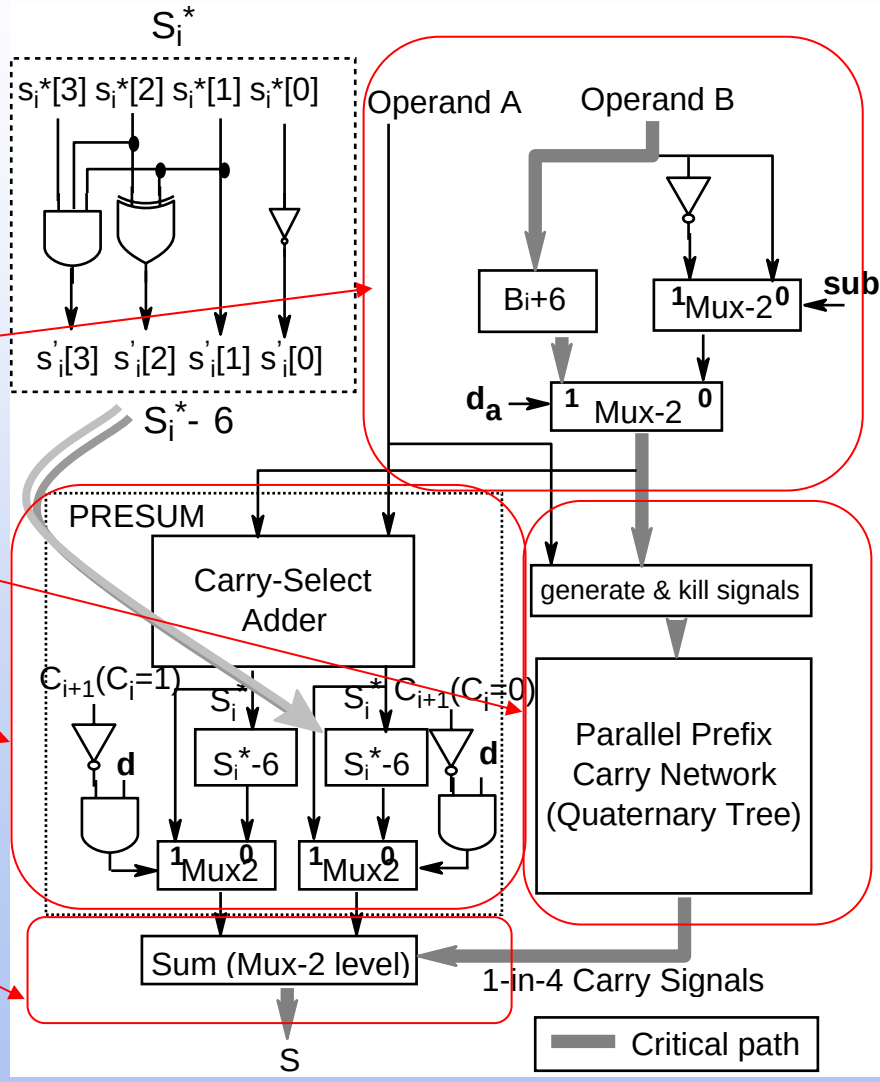
Performs binary and speculative decimal additions/subtractions.

STAGES

- Operand setup.
- Pre-sum
- Carry-evaluation (pre-carry and carry tree).
- Sum.

Quaternary carry tree (sparse tree, 1-in-4 carries).

4-bit carry-select adders. Sum correction performed along with carry evaluation.



Conditional Speculative Decimal Addition

Contents

- Introduction
  - Demand of High-performance Decimal Arithmetic.
  - Revision of the IEEE-754 Standard for Floating Point.
- Previous Work on Integer Decimal Addition.
  - Basic Decimal Addition.
  - Direct Decimal Addition.
  - Speculative Decimal Addition.
- Proposed Method: Conditional Speculative Decimal Addition.
  - Algorithm.
  - Implementations: Parallel Prefix Adders.
    - Binary Carry Tree: Kogge-Stone and Ladner-Fischer.
    - Quaternary Carry Tree.
- Delay-Area Estimations and Comparison.
  - Delay-area model for Static CMOS gates based on Logical Effort.
- Conclusions.

Proposed Method: Conditional Speculative Decimal Addition Algorithm

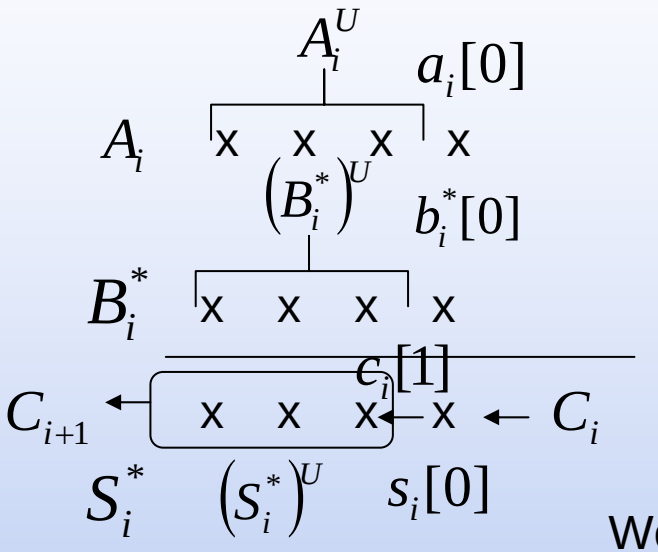
Motivation: improve unconditional speculation.

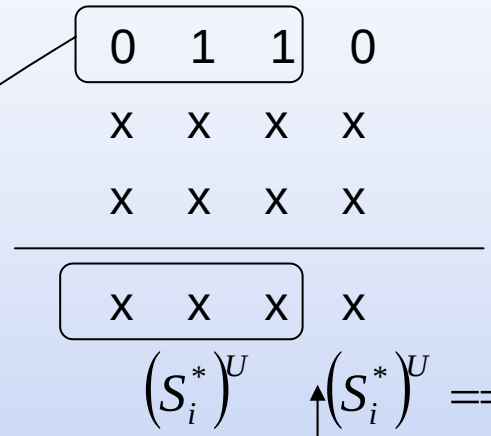
- **Reducing** the **complexity** of sum digits correction.
- **Removing post-correction** from critical path delay.

Solution: Find a **simple condition** to reduce the values for which the speculation fails ➔ simple scheme for sum digits correction.

Proposed Method: Conditional Speculative Decimal Addition Algorithm

- Division of input digits in upper (3 left bits) and lower (right bit) parts:





$(S_i^*)^U == 14 \rightarrow 8$

Correction (14 ➔ 8)

We add +6 in this case.

- Condition for speculation : $A_i^U + (B_i^*)^U \geq 8$ ➔ $C_{i+1} = 1$ ← supposition

real ↓

- Wrong speculation : $A_i^U + (B_i^*)^U + 6 = 14$ and $c_i[1] = 0$ ➔ $C_{i+1} = 0$

Proposed Method: Conditional Speculative Decimal Addition Algorithm

- Signals for conditional speculation (detection of $A_i^U + (B_i^*)^U \geq 8$)

$$r_i = \overline{k_i[3]} + g_i[2] + \overline{k_i[1]} \cdot g_i[1]$$ For addition (d_a ==1)

$$t_i = a_i[3] + \overline{k_i[3]} \cdot (g_i[2] + \overline{k_i[2]} \cdot \overline{k_i[1]})$$ For subtraction (d_s ==0)

- Conditional speculation:

$$\left(S_i^*\right)^U = \begin{cases} Add\ \ 6 & if\ (r_i == 1) \\ - & if\ (r_i == 0) \end{cases}$$ For addition (d_a ==1)

$$\left(S_i^*\right)^U = \begin{cases} - & if\ (t_i == 1) \\ Add\ \ 6 & if\ (t_i == 0) \end{cases}$$ For subtraction (d_s ==0)

Proposed Method: Conditional Speculative Decimal Addition Implementations

- **Goal**: simplify the sum correction of the speculative methods.

  1. Full binary parallel prefix carry tree configurations ➔ **Improve delay** eliminating post-correction from critical path.
  2. Quaternary carry tree configurations ➔ **Improve area** simplifying correction.

- Lower dependency on the carry tree topology ➔ **More flexibility** to choose the adder architecture and area/latency trade-offs.

- Combined binary/decimal implementations ➔ Efficient implementation using any existing binary parallel prefix adder.

Proposed Method: Conditional Speculative Decimal Addition Implementations

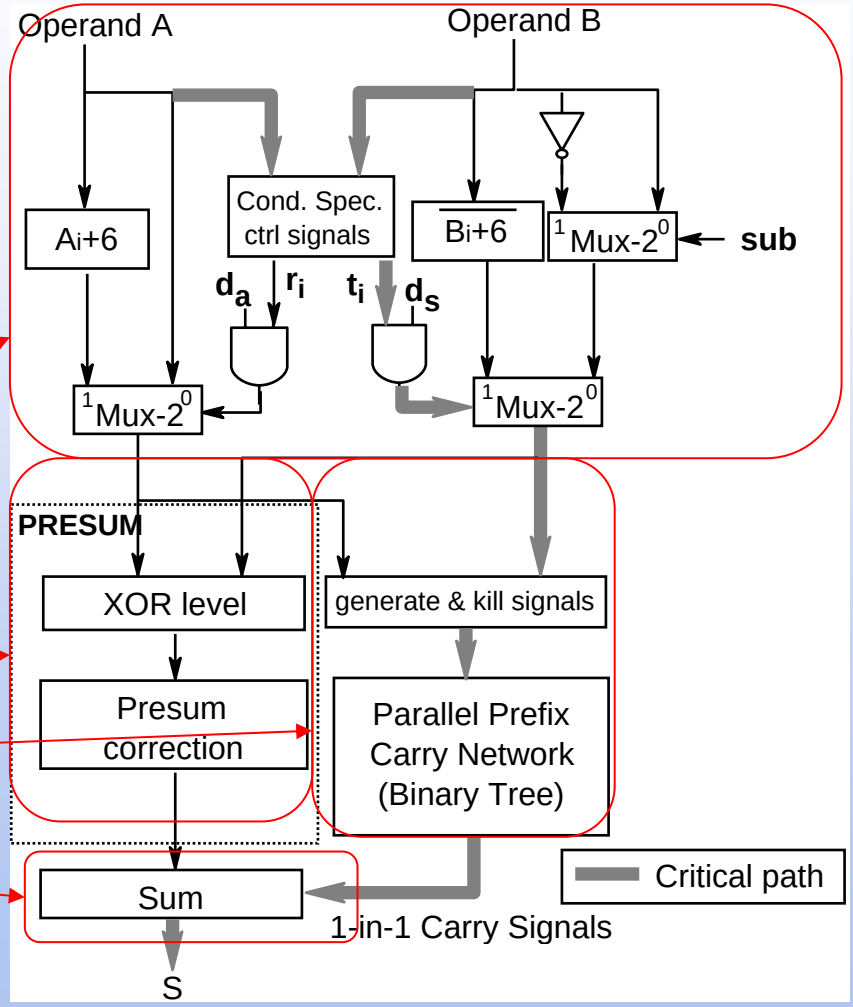
Performs binary and conditional speculative decimal additions/subtractions.

Binary carry tree (Kogge-Stone, Ladner-Fischer, etc…)

Avoids post-correction in the critical path.

STAGES

- •Operand setup.
- •Pre-sum
- •Carry-evaluation (pre-carry and carry tree).
- •Sum.



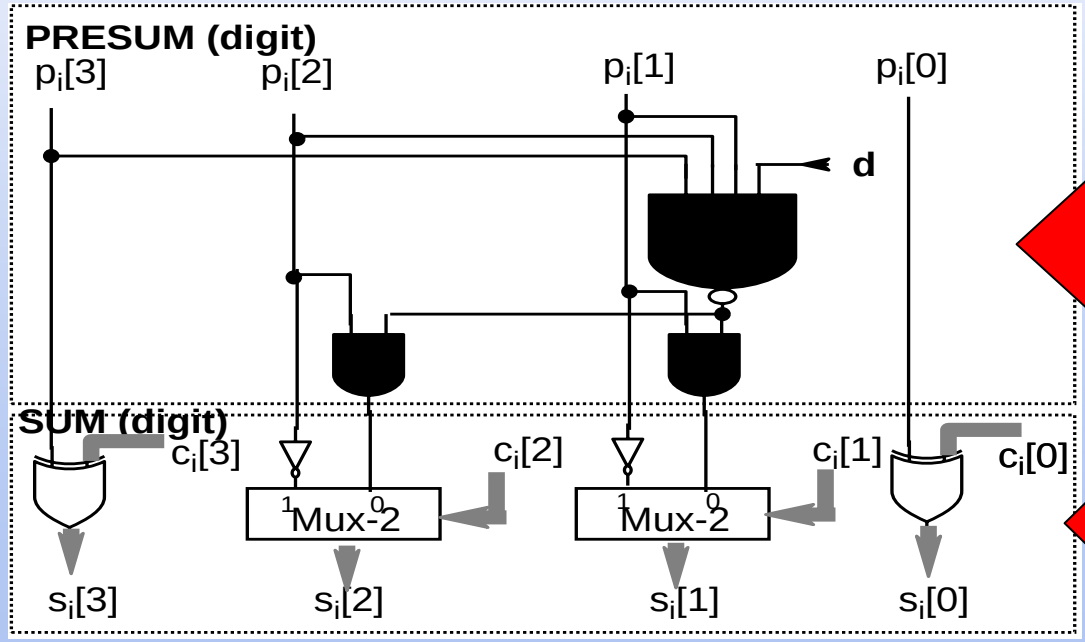
Proposed Method: Conditional Speculative Decimal Addition Implementations

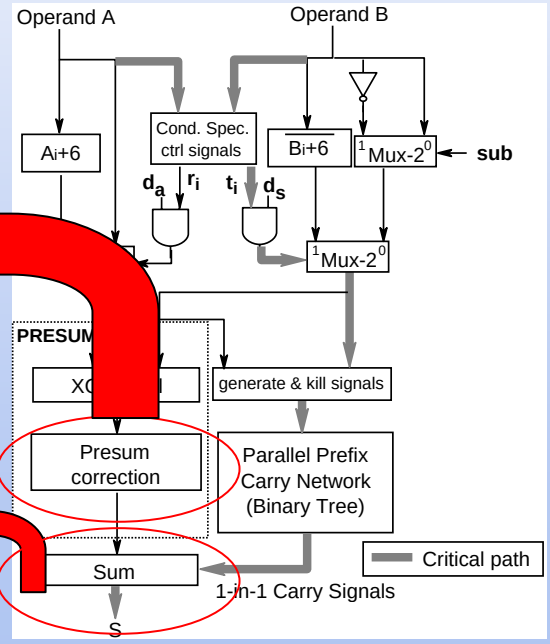
- Simple correction: **(d == 1)**

  Black gates replace 111- by 100- when $\left(S_i^*\right)^U == 14$

  That is, $A_i^U + (B_i^*)^U + 6 = 14$ and $c_i[1] = 0$

- Additional gates (black) not in the critical path (grey).





Proposed Method: Conditional Speculative Decimal Addition Implementations

Performs binary and conditional speculative decimal additions/subtractions.

Quaternary carry tree (sparse tree, 1-in-4 carries).

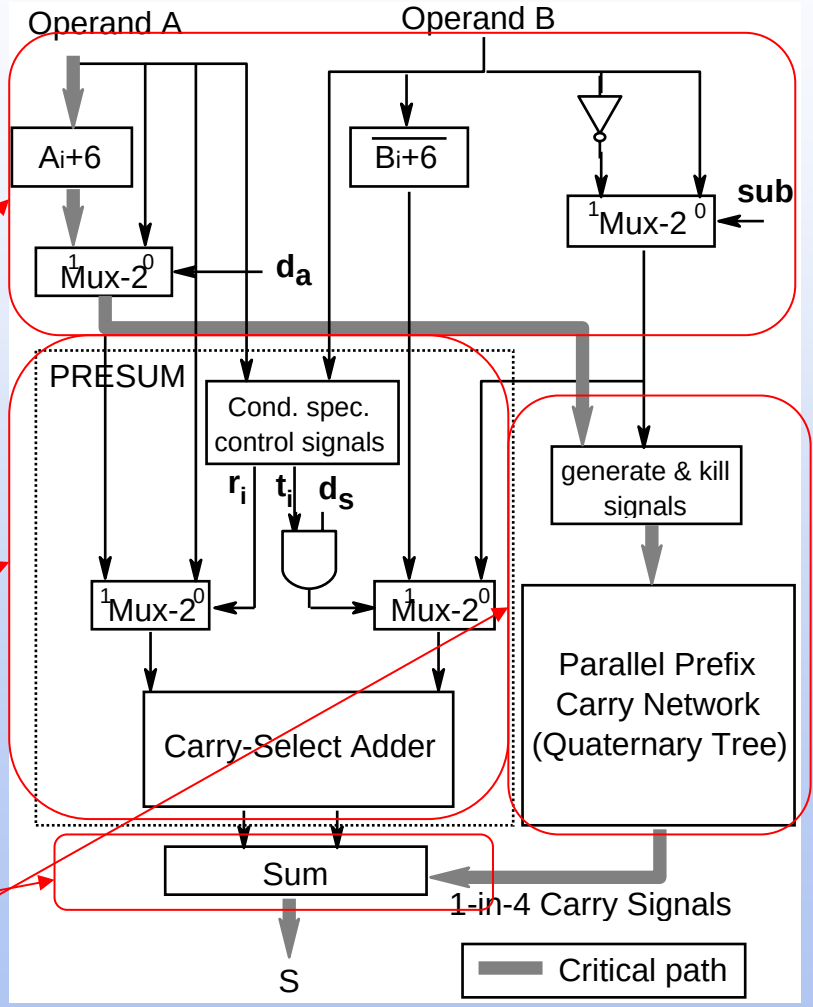
4-bit carry-select adders. Sum correction performed in carry-select adders.

Simplified selection function (only for sum digits correction):

Decimal carries do not depend on condition for speculation

STAGES

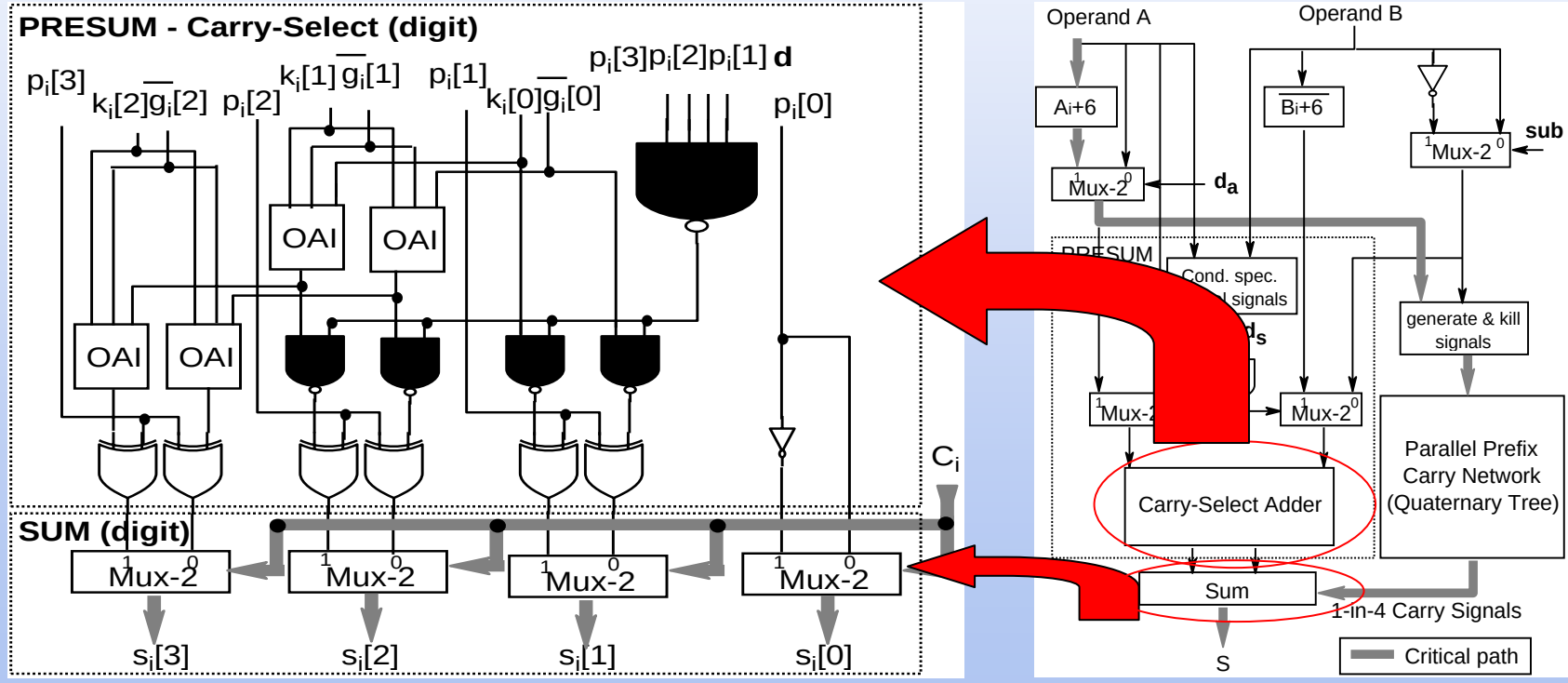
- •Operand setup.
- •Pre-sum
- •Carry-evaluation (pre-carry and carry tree).
- •Sum.



Proposed Method: Conditional Speculative Decimal Addition Implementations

Modified 4-bit carry select adder.

- Simple correction: Replace 111- by 100- when $\left(S_i^*\right)^U == 14$ **(d == 1)**
- Performed along with carry computation
- Additional gates (black) not in the critical path (grey).



Conditional Speculative Decimal Addition

Contents

- Introduction
  - Demand of High-performance Decimal Arithmetic.
  - Revision of the IEEE-754 Standard for Floating Point.
- Previous Work on Integer Decimal Addition.
  - Basic Decimal Addition.
  - Direct Decimal Addition.
  - Speculative Decimal Addition.
- Proposed Method: Conditional Speculative Decimal Addition.
  - Algorithm.
  - Implementations: Parallel Prefix Adders.
    - Binary Carry Tree: Kogge-Stone and Ladner-Fischer.
    - Quaternary Carry Tree.
- Delay-Area Estimations and Comparison.
  - Delay-area model for Static CMOS gates based on Logical Effort.
- Conclusions.

Delay-Area Estimations and Comparison

Delay-Area of Static CMOS Gates

- **Delay model** for static CMOS gates based on **Logical Effort**.
- Delay values given in **FO4 units** (1x inverter with fanout 4 1x inv).
- Area values given in 1x **Nand2 gate units**.
- Rough model valid for comparison among architectures but not for obtaining precise absolute evaluation results.
- We take into account loads but neither interconnections nor gate sizing optimizations (we assume gates with the drive strength of min. sized inv. and introduce buffers when necessary).

Delay-Area Estimations and Comparison

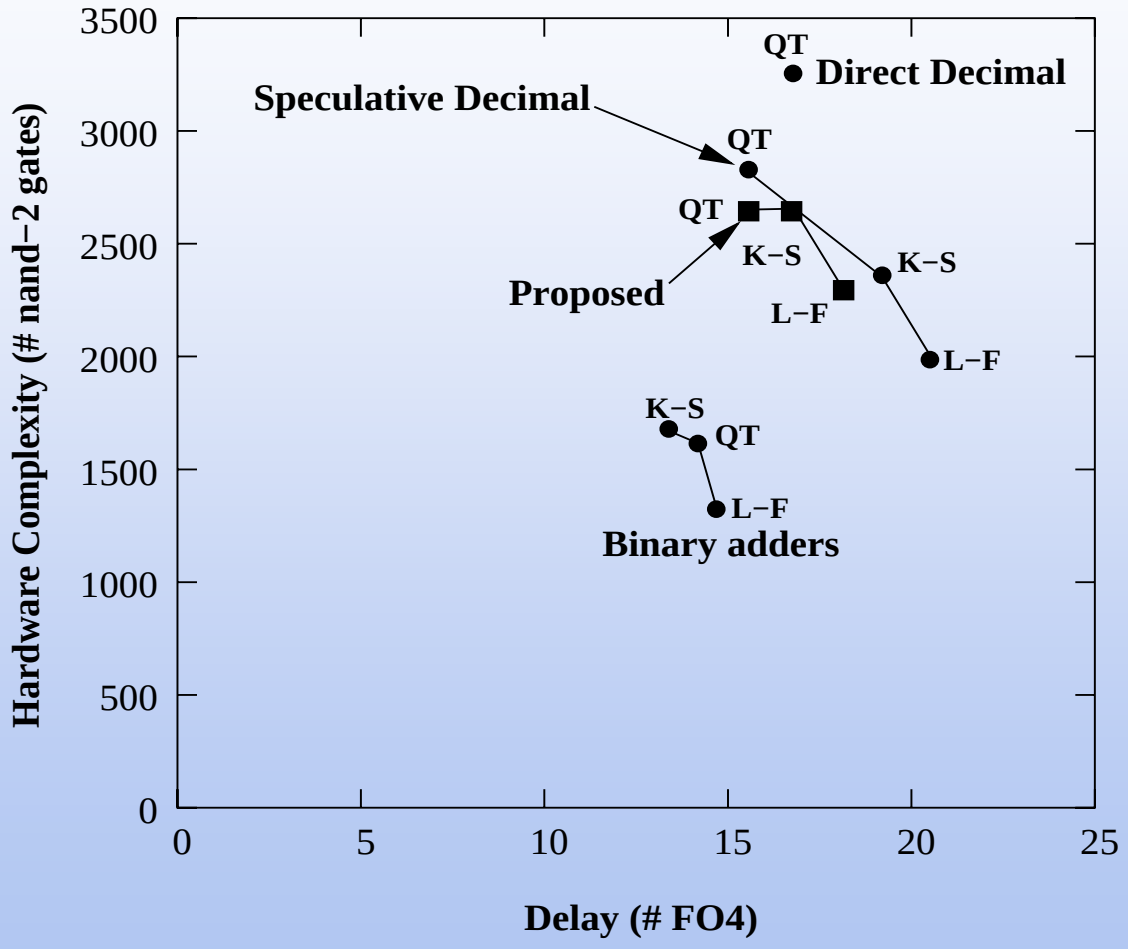
Area-Delay Evaluation Results

64-bit combined binary/decimal adders

Prefix tree	**Direct Decimal**		**Speculative**		**Proposed**	
	Delay (t_{fo4})	Area (Nand2)	Delay (t_{fo4})	Area (Nand2)	Delay (t_{fo4})	Area (Nand2)
K-S	**----**	**----**	**19.25 (1.14x)**	**2360 (1x)**	**16.85 (1x)**	**2660 (1.13x)**
L-F	**----**	**----**	**20.65 (1.13x)**	**1985 (1x)**	**18.25 (1x)**	**2290 (1.15x)**
Q-T	**16.85 (1.08x)**	**3251 (1.22x)**	**15.55 (1x)**	**2825 (1.06x)**	**15.55 (1x)**	**2655 (1x)**

In brackets the relative ratios for each parallel prefix configuration.

Area-Delay Space of Analyzed Adders



Binary/Decimal Combined Adders:

- Direct Decimal no apparent advantage respect speculative methods.
- For low latency Q-T best choice (our proposal requires less hardware).
- For low hardware cost and area-latency trade-off, L-F schemes are the best alternatives.
- Proposed combined Q-T adder only 1.10 slower than binary Q-T although 1.65x more complex.

Conditional Speculative Decimal Addition

Contents

- Introduction
  - Demand of High-performance Decimal Arithmetic.
  - Revision of the IEEE-754 Standard for Floating Point.
- Previous Work on Integer Decimal Addition.
  - Basic Decimal Addition.
  - Direct Decimal Addition.
  - Speculative Decimal Addition.
- Proposed Method: Conditional Speculative Decimal Addition.
  - Algorithm.
  - Implementations: Parallel Prefix Adders.
    - Binary Carry Tree: Kogge-Stone and Ladner-Fischer.
    - Quaternary Carry Tree.
- Delay-Area Estimations and Comparison.
  - Delay-area model for Static CMOS gates based on Logical Effort.
- Conclusions.

Conditional Speculative Decimal Addition

Conclusions

- New high-performance algorithm for decimal integer addition/subtraction.
- Avoid the penalty delay of post-correction schemes.
- Efficient implementation using parallel prefix adders: both binary and quaternary carry tree configurations.
- Evaluation results show very competitive area-delay figures respect to commercial and patented implementations.