An Adder for a Redundant Digit Arithmetic Unit

Hossam A. H. Fahmy

Electronics and Communications Department

Cairo University, Egypt

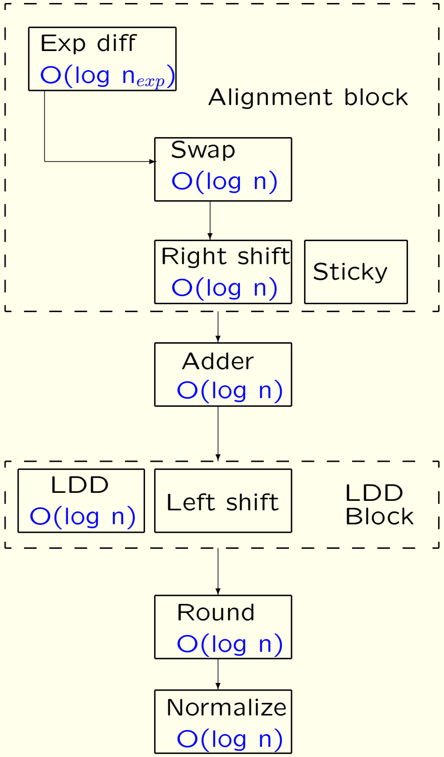
Michael J. Flynn

Computer Systems Laboratory

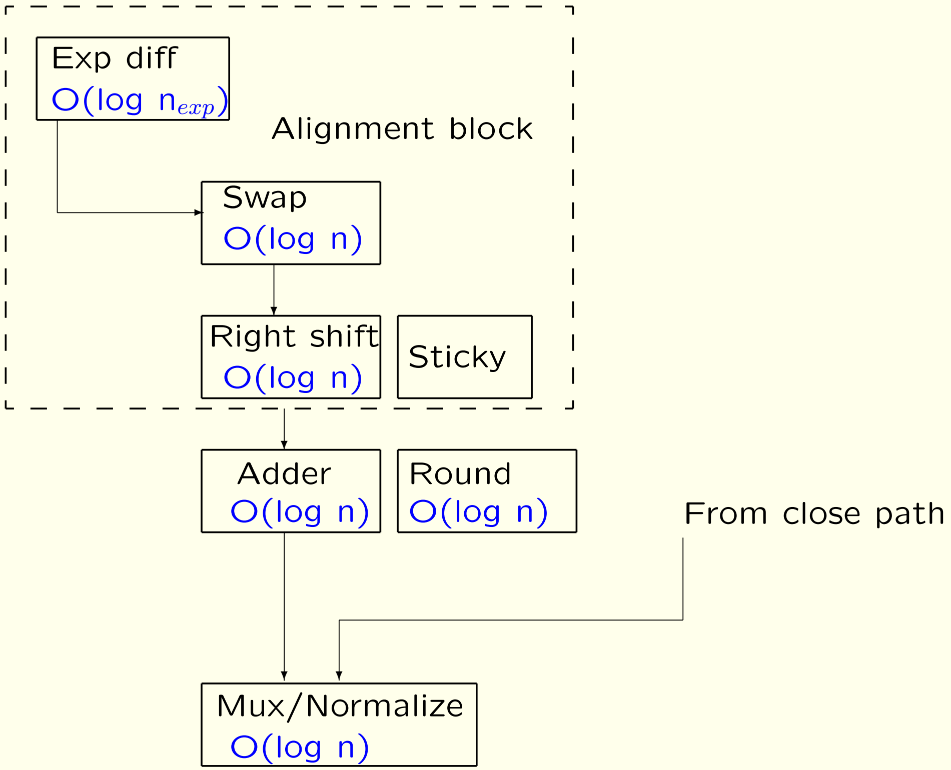
Stanford University, USA

General Outline

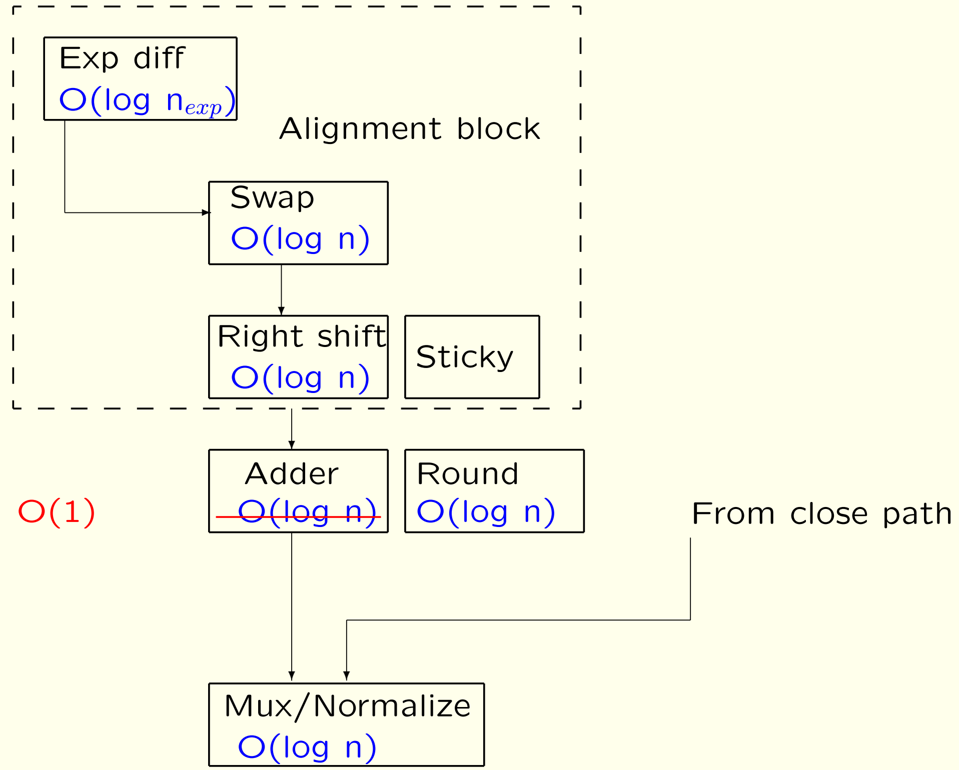
- Time delays in floating point addition
- Proposed system
- Modeling and comparisons
- Simulation results



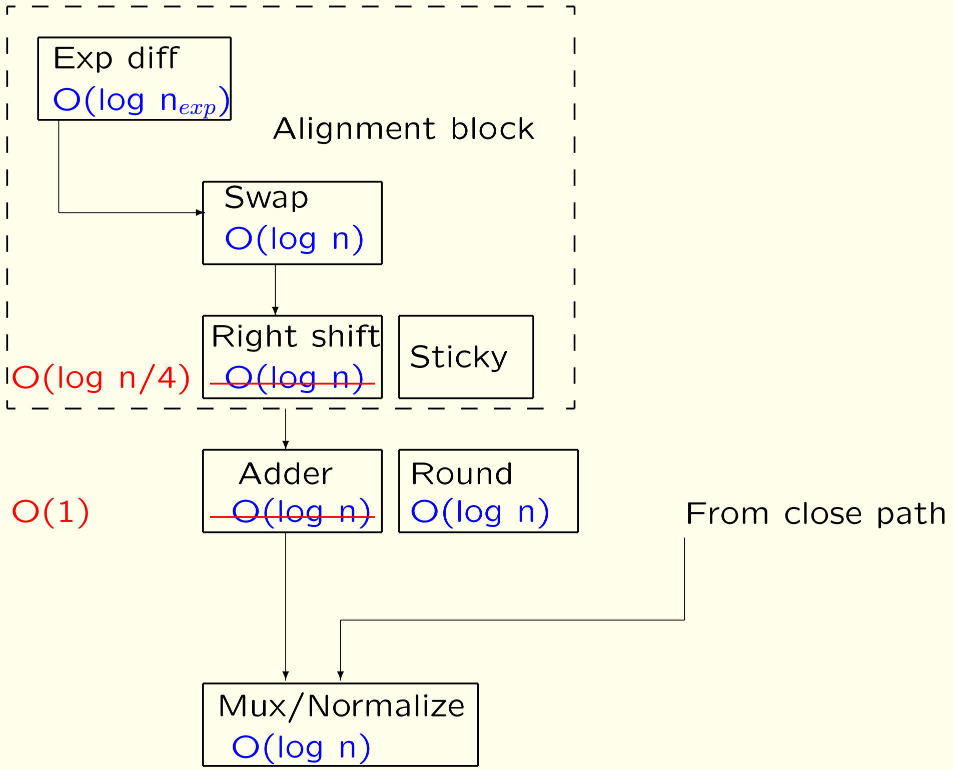
Time delays in the blocks of an adder (one-path algorithm)



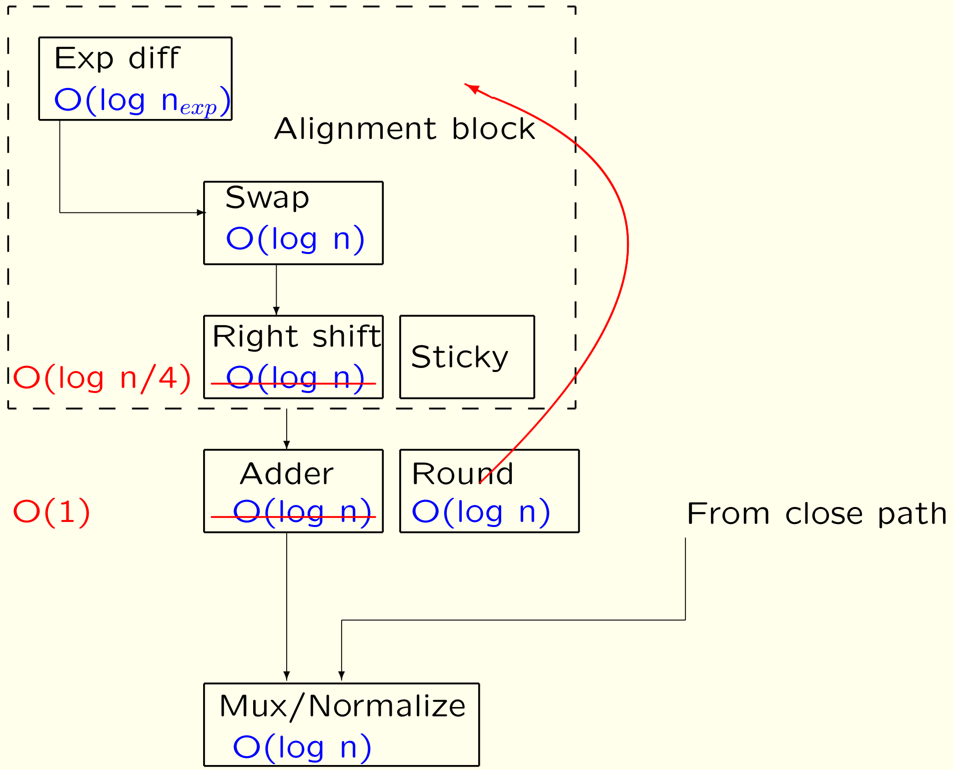
Time delays in the blocks of an adder (two-path algorithm)



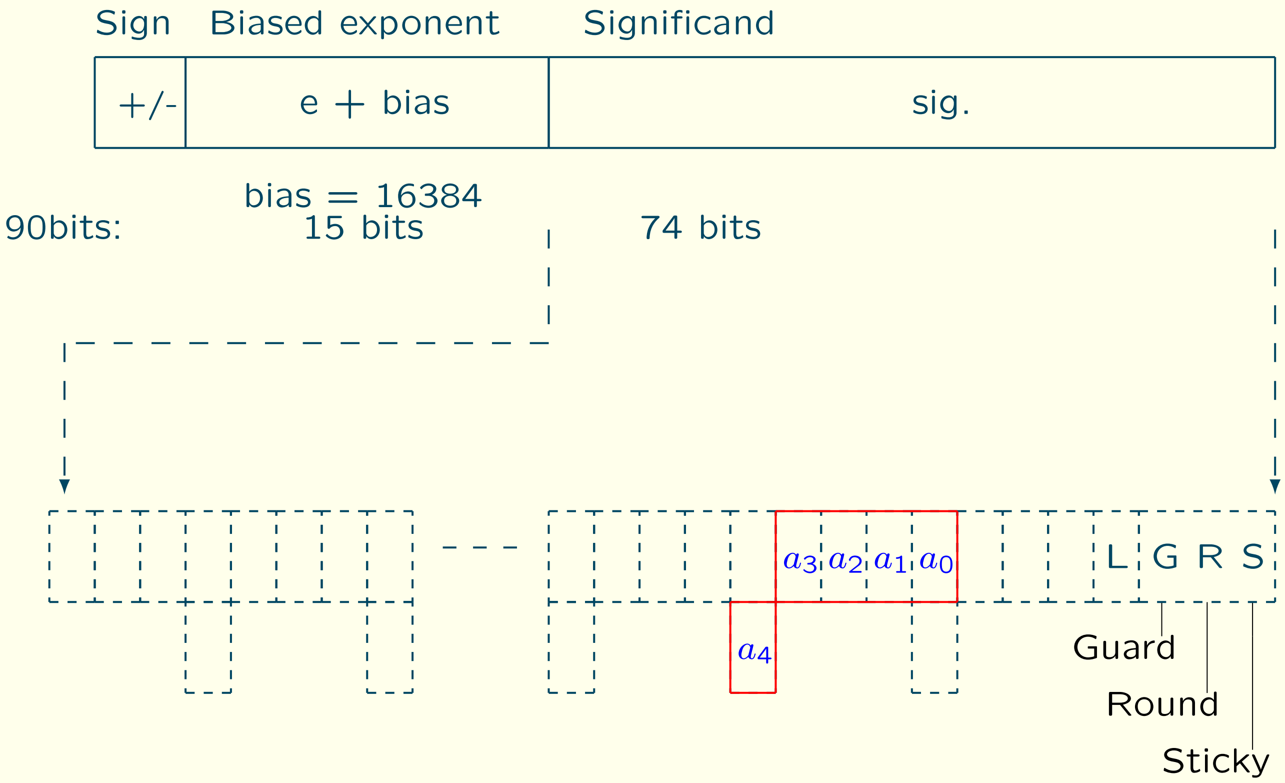
Time delays with redundancy



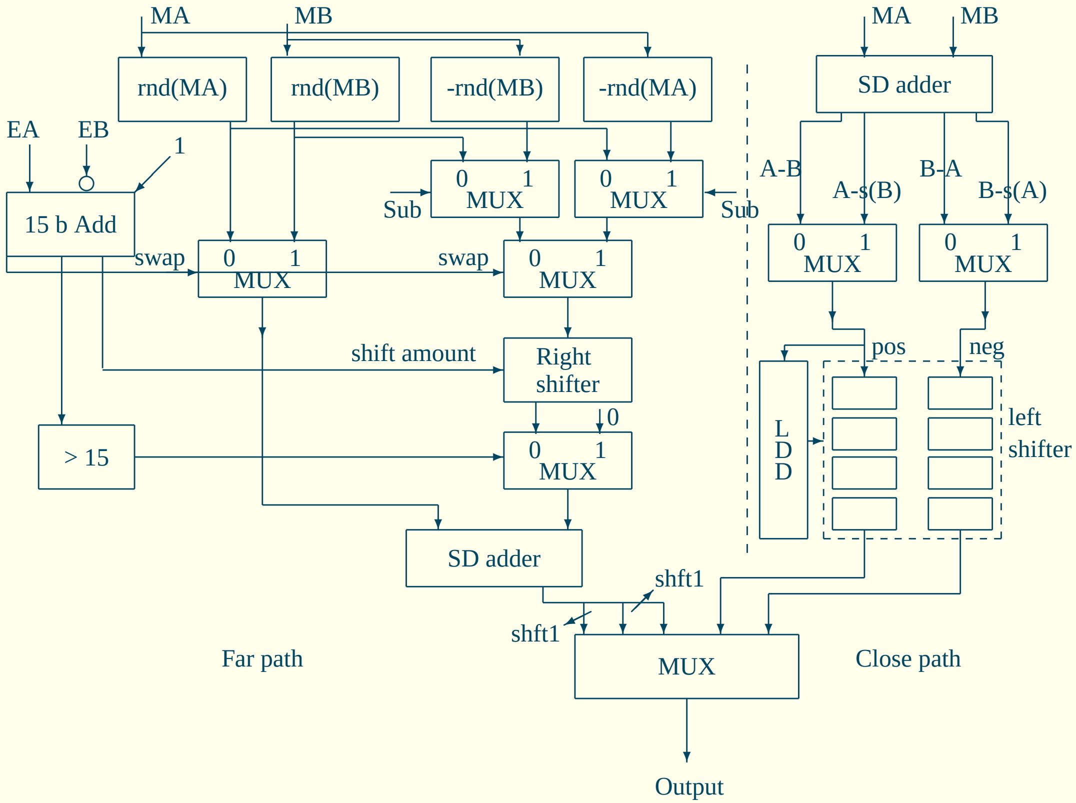
Time delays with redundancy



Time delays with redundancy



The proposed SD format for floating point numbers



General blocks of the proposed adder (two-path algorithm)

Background on SD numbers

Ordinary Signed Digit (SD) numbers represent a number in radix $r > 2$ with digits $x_i \in \{-\alpha, \cdots, -1, 0, 1, \cdots, \alpha\}$ where $\frac{r}{2} < \alpha < r$

For example, in decimal where $r = 10$ we have $1\,\overline{8} = 0\;2$.

Assuming $r = 10$ and $\alpha = 9$ addition goes as follows:

$$
\begin{array}{cccccccccc}
 & 2 & 1 & \overline{8} & & & & 2 & 0 & 2 \\
+ & 7 & 9 & 9 & & & + & 7 & 9 & 9 \\ \cline{1-4}\cline{6-10}
 & 9 & 10 & 1 & \mid p \mid \geq \alpha? & & & 9 & 9 & 11 \\ \cline{1-4}\cline{6-10}
1 & 1 & 0 & & c & & & & & \\
 & \overline{1} & 0 & 1 & w & & & & & \\ \cline{1-4}\cline{6-10}
1 & 0 & 0 & 1 & s & & 1 & 0 & 0 & 1
\end{array}
$$

$$p_i = x_i + y_i \qquad\qquad c_i = \begin{cases} -1 & \text{if } p_i \leq -\alpha \\ 1 & \text{if } p_i \geq \alpha \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} w_i &= x_i + y_i - \beta c_i \\ s_i &= w_i + c_{i-1} \end{aligned}$$

Using as a radix $\beta = 16$ and $\alpha = 15$,

$$\begin{array}{ccc} x_i, y_i, s_i & \in & \{-15, \cdots, 15\} \\ p_i & \in & \{-30, \cdots, 30\} \end{array}$$

In the case of a positive overflow, $x_{i_4} = y_{i_4} = 0$ but $p_{i_4} = 1$. Hence, $w_i = p_i - 16 = p_i - (10000)_2$.

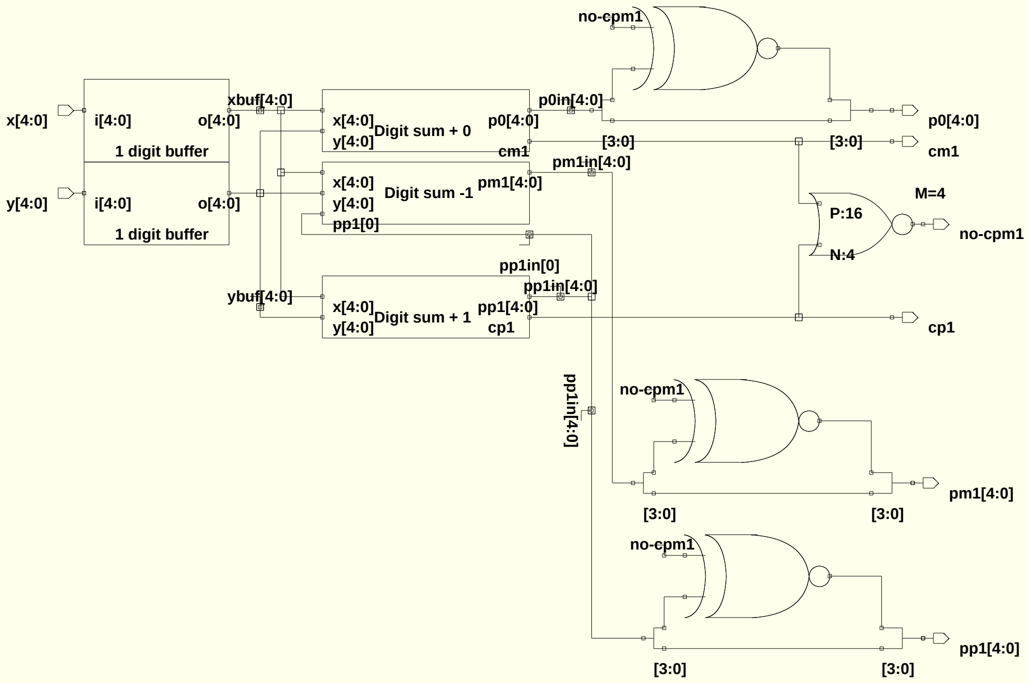
$$\boxed{\text{For } c_i = \pm 1,\ w_{i_4} = \bar{p}_{i_4}}$$

1. $c_{i(1)}$ is set for a positive overflow or a result equal to $+15$. Both of these occur only if $x_{i_4} = y_{i_4} = 0$.

   - The adder calculating $x_i + y_i + 1$ has a positive overflow if $x_i + y_i \geq 15$

2. $c_{i(-1)}$ is set for a negative overflow or a result equal to -15.

   - An output of -15 results if one input is -15 and the other is zero or both inputs are negative and their sum equals -15.
   - The inputs being -16 is a don't care condition.

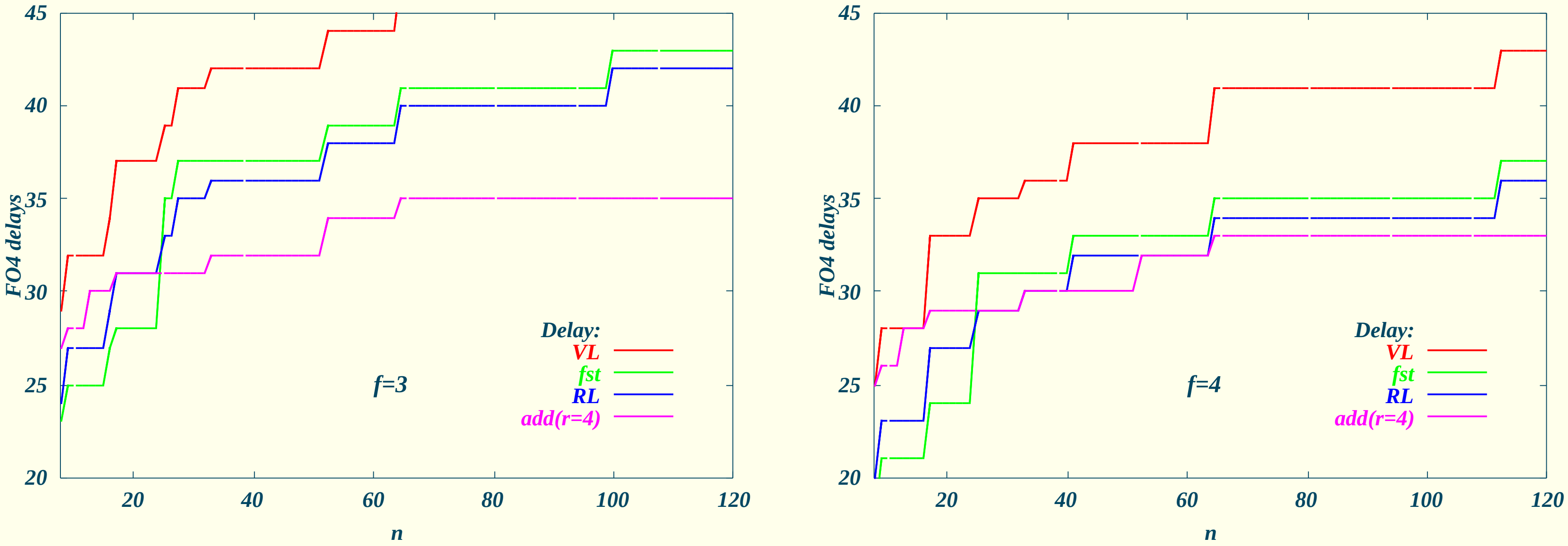


Possible outcomes for one digit

For decimal floating point operations, use $\beta = 10$ and $\alpha = 9$.

$$\begin{array}{ccl} x_i, y_i, s_i & \in & \{-9, \cdots, 9\} \\ p_i & \in & \{-18, \cdots, 18\} \end{array}$$

$$\begin{array}{ll} c_{i(1)} & = \overline{x}_{i_4}\overline{y}_{i_4}(p_{i_4} + p_{i_3}(p_{i_2} + p_{i_1} + p_{i_0})) \\ c_{i(-1)} & = x_{i_4}y_{i_4}\overline{p}_{i_4} + (x_{i_4} + y_{i_4})p_{i_4}\overline{p}_{i_3} \end{array}$$



Complete binary floating point adder comparisons:
Time delay versus significand width for different fan-in

- A scalable CMOS technology was used to design a binary floating point adder and multiplier at the transistor level with $n = 53$, $f = 3$ and $r = 4$.

- Both designs perform all the IEEE rounding modes.

- Both designs were simulated for functionality at the logic level using *verilog* and for speed at the transistor level using *irsim*.

Circuit statistics and simulation results

	Floating point adder
number of nodes	46845
NMOS transistors	63589
PMOS transistors	61649
Model delay	34$FO4$
Sim delay(0.6μm)	14ns (33.35$FO4$)
Sim delay(0.3μm)	6ns (32.40$FO4$)

Cost?

		$n = 60$	$n = 80$	$n = 120$
$r = 4$	width increase(%)	33.3	31.6	29.6
	speed up (%)	10.5	12.5	16.7
$r = 8$	width increase(%)	29.3	21.1	18.5
	speed up (%)	7.9	10	14.3

Floating point adder trade off when $f = 3$.

Conclusions and Contributions

- A new internal format based on SD numbers and the corresponding floating point unit adder are presented
- The SD adder is adapted to perform decimal addition
- Through analytical modeling and transistor simulations, the proposed designs perform better than the conventional ones for the double precision numbers.