Phi Luu ECE_271_001_F2018 11/14/2018

EXAMPLE PROBLEMS

Example 5.1: Ripple-carry adder and carry-lookahead adder delay (page 243)

32-bit ripple-carry adder, 32-bit carry-lookahead adder with 4-bit blocks. $t_{pg} = 100ps$, $t_{FA} = 300ps$, $N = 32$

$$t_{ripple} = Nt_{FA} = 32 \cdot 300ps = 9600ps = 9.6ns$$

$$t_{CLA} = t_{pg} + t_{pg\_block} + \left(\frac{N}{k} - 1\right)t_{AND\_OR} + kt_{FA}$$

$$= 100ps + (6 \cdot 100ps) + \left(\frac{32}{4} - 1\right)(2 \cdot 100ps) + (4 \cdot 300ps)$$

$$= 3.3ns$$

Thus, the carry-lookahead adder is almost 3 times faster than the ripple-carry adder.

Variation: $N = 64$, $k = 8$, $t_{pg} = 50ps$, $t_{FA} = 200ps$

$$t_{ripple} = Nt_{FA} = 64 \cdot 200ps = 12.8ns$$

$$t_{CLA} = t_{pg} + t_{pg\_block} + \left(\frac{N}{k} - 1\right)t_{AND\_OR} + kt_{FA}$$

$$= 50ps + (6 \cdot 50ps) + \left(\frac{64}{8} - 1\right)(2 \cdot 50ps) + (4 \cdot 200ps)$$

$$= 1.85ns$$

Example 5.2: Prefix adder delay

$N = 32$, $t_{pg} = 100ps$, $t_{pg\_prefix} = 200ps$

$$t_{PA} = t_{pg} + \log_2 N(t_{pg\_prefix}) + t_{XOR}$$

$$= 100ps + \log_2(32) \cdot 200ps + 100ps = 1.2ns$$

Variation $N = 64$

$$t_{PA} = t_{pg} + \log_2 N(t_{pg\_prefix}) + t_{XOR}$$

$$= 100ps + \log_2(64) \cdot 200ps + 100ps = 1.4ns$$

Example 5.3. Set less than.

$N = 32$, $A = 25_{10}$, $B = 32_{10}$.

Since $A < B$, $Y_{expected} = 1$. For SLT, $F_{2:0} = 111$.

$S = A - B = 25_{10} - 32_{10} = -7_{10} = \underbrace{1111 \ldots 1001_2}_{32\text{ bits}}$.

With $F_{1:0} = 11$, the final multiplexer sets $Y = S_{31} = 1$

Variation: $N = 64$, $A = 96_{10}$, $B = 69_{10}$

Since $A > B$, $Y_{expected} = 0$. For SLT, $F_{2:0} = 111$

$S = A - B = 96_{10} - 69_{10} = 27_{10} = \underbrace{0000 \ldots 11011_2}_{32\text{ bits}}$

With $F_{1:0} = 11$, the final multiplexer sets $Y = S_{31} = 0$

Example 5.4. Arithmetic with fixed-point numbers.

Compute $0.75 + -0.625$ using fixed-point numbers

$0.625 \xrightarrow[-2^{-1}]{} 0.125 \xrightarrow[-2^{-3}]{} 0 \Rightarrow 0.625_{10} = 0000.1010_2$

$$\begin{array}{r} \neg 1111.0101_2 \\ + \quad 1_2 \\ \hline 1111.0110_2 \end{array}$$

$\Rightarrow -0.625_{10} = 1111.0110_2$

$0.75 \xrightarrow[-2^{-1}]{} 0.25 \xrightarrow[-2^{-2}]{} 0 \rightarrow 0.75_{10} = 0000.1100_2$

Thus, $0.75_{10} + -0.625_{10} = \begin{array}{r} 0000.1100_2 \\ + 1111.0110_2 \\ \hline \cancel{1}0000.0010_2 \end{array} = 0.125_{10}$

Variation: $1.5 + 7.5 = ?$

$1.5 \xrightarrow[-2^{0}]{} 0.5 \xrightarrow[-2^{-1}]{} 0 \Rightarrow 1.5_{10} = 0001.1000_2$

$7.5 \xrightarrow[-2^{2}]{} 3.5 \xrightarrow[-2^{1}]{} 1.5 \xrightarrow[-2^{0}]{} 0.5 \xrightarrow[-2^{-1}]{} 0$

$\Rightarrow 7.5_{10} = 0111.1000_2$

Thus, $1.5_{10} + 7.5_{10} = \begin{array}{r} 0001.1000_2 \\ + 0111.1000_2 \\ \hline 01001.0000_2 \end{array} = 01001.0000_2 = 9_{10}$

Phi Luu ECE_271_001_F2018 11/14/2018

Example 5.5: 32-bit floating point numbers

Show the floating-point representation of the decimal number 228

$228_{10} = 11100100_2 = 1.11001_2 \cdot 2^7$

1 bit	8 bits	23 bits
0	0000 0111	111 0010 0000 0000 0000 0000
Sign	Exponent	Mantissa

Variation: $6969_{10} = ?$

$6969_{10} = 1\ 1011\ 0011\ 1001_2 = 1.\ 1011\ 0011\ 1001_2 \cdot 2^{12}$

1 bit	8 bits	23 bits
0	0 000 1100	11011001110010000000000
Sign	Exponent	Mantissa