DIGITAL ARITHMETIC
Miloš D. Ercegovac and Tomás Lang
Morgan Kaufmann Publishers, an imprint of Elsevier Science, ©2004
– Updated: September 23, 2003 –

Chapter 6: Solutions to Exercises

– With contributions by Elisardo Antelo and Fabrizio Lamberti –

Exercise 6.1

a) *Radix-2, $s_j \in \{-1, 0, 1\}$, conventional (nonredundant) residual*

We have $x = 144 \times 2^{-8} = 0.10010000$ and $\rho = 1$. We choose $s_0 = 0$. Therefore the initialization is $w[0] = x - s_0 = 0.10010000$.

We use the result-digit selection function for redundant residual but we consider only 2 integer bits since the range of the residual estimate is smaller than in the redundant case.

$$
\begin{array}{rrll}
2w[0] = & 001.00100000 & \widehat{y} = 1 & s_1 = 1 \\
F_1[0] = & 11.10000000 & F_{-1}[0] = 11.10000000 & \\
\hline
w[1] = & 00.10100000 & & \\
 & & & \\
2w[1] = & 001.01000000 & \widehat{y} = 1 & s_2 = 1 \\
F_1[1] = & 10.11000000 & F_{-1}[1] = 00.11000000 & \\
\hline
w[2] = & 00.00000000 & & \\
 & & & \\
2w[2] = & 000.00000000 & \widehat{y} = 0 & s_3 = 1 \\
F_1[2] = & 10.01100000 & F_{-1}[2] = 01.01100000 & \\
\hline
w[3] = & 10.01100000 & & \\
 & & & \\
2w[3] = & 100.11000000 & \widehat{y} = -4 & s_4 = -1 \\
F_{-1}[3] = & 01.10110000 & F_1[3] = 10.00110000 & \\
\hline
w[4] = & 10.01110000 & & \\
 & & & \\
2w[4] = & 100.11100000 & \widehat{y} = -4 & s_5 = -1 \\
F_{-1}[4] = & 01.10011000 & F_1[4] = 10.01011000 & \\
\hline
w[5] = & 10.01111000 & & \\
 & & & \\
2w[5] = & 100.11110000 & \widehat{y} = -4 & s_6 = -1 \\
F_{-1}[5] = & 01.10001100 & F_1[5] = 01.10001100 & \\
\hline
w[6] = & 1110.01111100 & &
\end{array}
$$

(*continues on next page*)

$$
\begin{array}{rlll}
2w\,[6] = & 100.11111000 & \widehat{y} = -4 & s_7 = -1 \\
F_{-1}\,[6] = & 01.10000110 & F_1\,[6] = 10.01110110 & \\
\hline
w\,[7] = & 10.01111110 & &
\end{array}
$$

$$
\begin{array}{rlll}
2w\,[7] = & 100.11111100 & \widehat{y} = -4 & s_8 = -1 \\
F_{-1}\,[7] = & 01.10000011 & F_1\,[7] = 10.01111011 & \\
\hline
w\,[8] = & 10.01111111 & &
\end{array}
$$

$$
\begin{array}{rlll}
2w\,[8] = & 100.11111110 & \widehat{y} = -4 & s_9 = -1 \\
F_{-1}\,[8] = & 01.10000001 & F_1\,[8] = 10.01111101 & \\
\hline
w\,[9] = & 10.01111111 & &
\end{array}
$$

We perform 9 iterations to compute the additional bit required for rounding. Since $w\,[9] < 0$ the correction step has to be performed. Thus $s_9 = -2$. The result is

$$
s = 0.111\bar{1}\bar{1}\bar{1}\bar{1}\bar{1}\bar{2} = (0.11000000)_2
$$

b) *Radix-2, $s_j \in \{-1, 0, 1\}$, carry-save residual*

$$
\begin{array}{rlll}
2WS\,[0] = & 0001.00100000 & \widehat{y} = 1 & s_1 = 1 \\
2WC\,[0] = & 0000.00000000 & & \\
F_1\,[0] = & 111.10000000 & F_{-1}\,[0] = 111.10000000 & \\
\hline
WS\,[1] = & 110.10100000 & & \\
WC\,[1] = & 010.00000000 & &
\end{array}
$$

$$
\begin{array}{rlll}
2WS\,[1] = & 1101.01000000 & \widehat{y} = 1 & s_2 = 1 \\
2WC\,[1] = & 0100.00000000 & & \\
F_1\,[1] = & 110.11000000 & F_{-1}\,[1] = 000.11000000 & \\
\hline
WS\,[2] = & 111.10000000 & & \\
WC\,[2] = & 000.10000000 & &
\end{array}
$$

$$
\begin{array}{rlll}
2WS\,[2] = & 1111.00000000 & \widehat{y} = 0 & s_3 = 1 \\
2WC\,[2] = & 0001.00000000 & & \\
F_1\,[2] = & 110.01100000 & F_{-1}\,[2] = 001.01100000 & \\
\hline
WS\,[3] = & 000.01100000 & & \\
WC\,[3] = & 110.00000000 & &
\end{array}
$$

$$
\begin{array}{rlll}
2WS\,[3] = & 0000.11000000 & \widehat{y} = -4 & s_4 = -1 \\
2WC\,[3] = & 1100.00000000 & & \\
F_{-1}\,[3] = & 001.10110000 & F_1\,[3] = 110.00110000 & \\
\hline
WS\,[4] = & 101.01110000 & & \\
WC\,[4] = & 001.00000000 & &
\end{array}
$$

(*continues on next page*)

$$
\begin{array}{rlll}
2WS\,[4] = & 1010.11100000 & \widehat{y} = -4 & s_5 = -1 \\
2WC\,[4] = & 0010.00000000 & & \\
F_{-1}\,[4] = & 001.10011000 & F_1\,[4] = 110.01011000 & \\
\hline
WS\,[5] = & 001.01111000 & & \\
WC\,[5] = & 101.00000000 & &
\end{array}
$$

$$
\begin{array}{rlll}
2WS\,[5] = & 0010.11110000 & \widehat{y} = -4 & s_6 = -1 \\
2WC\,[5] = & 1010.00000000 & & \\
F_{-1}\,[5] = & 001.10001100 & F_1\,[5] = 001.10001100 & \\
\hline
WS\,[6] = & 001.01111100 & & \\
WC\,[6] = & 101.00000000 & &
\end{array}
$$

$$
\begin{array}{rlll}
2WS\,[6] = & 0010.11111000 & \widehat{y} = -4 & s_7 = -1 \\
2WC\,[6] = & 1010.00000000 & & \\
F_{-1}\,[6] = & 001.10000110 & F_1\,[6] = 110.01110110 & \\
\hline
WS\,[7] = & 001.01111110 & & \\
WC\,[7] = & 101.00000000 & &
\end{array}
$$

$$
\begin{array}{rlll}
2WS\,[7] = & 0010.11111100 & \widehat{y} = -4 & s_8 = -1 \\
2WC\,[7] = & 1010.00000000 & & \\
F_{-1}\,[7] = & 001.10000011 & F_1\,[7] = 110.01111011 & \\
\hline
WS\,[8] = & 001.01111111 & & \\
WC\,[8] = & 101.00000000 & &
\end{array}
$$

$$
\begin{array}{rlll}
2WS\,[8] = & 0010.11111110 & \widehat{y} = -4 & s_9 = -1 \\
2WC\,[8] = & 1010.00000000 & & \\
F_{-1}\,[8] = & 001.10000001 & F_1\,[8] = 110.01111101 & \\
\hline
WS\,[9] = & 001.01111111 & & \\
WC\,[9] = & 101.00000000 & &
\end{array}
$$

We perform 9 iterations to compute the additional bit required for rounding. Since $w\,[9] < 0$ the correction step has to be performed. Thus $s_9 = -2$. The result is

$$s = 0.111\bar{1}\bar{1}\bar{1}\bar{1}\bar{1}\bar{2} = (0.11000000)_2$$

c) *Radix-4, $s_j \in \{-2, -1, 0, 1, 2\}$, carry-save residual*

Since $\rho = \frac{a}{r-1} = \frac{2}{3} < 1$, s_0 should be 1. Therefore $w\,[0] = 1 - s_0 = 111.10010000$.

$$
\begin{array}{rlll}
4WS[0] = & 1110.01000000 & \widehat{S} = 1.0000 & S[0] = 1 \\
4WC[0] = & 0000.00000000 & \widehat{y} = 1110.010 & s_1 = -1 \\
F_1[0] = & 001.11000000 & & S[1] = 0.11 \\
\hline
WS[1] = & 11.10000000 & & \\
WC[1] = & 00.10000000 & & \\
 & & & \\
4WS[1] = & 1110.00000000 & \widehat{S} = 0.1100 & s_2 = 0 \\
4WC[1] = & 0010.00000000 & \widehat{y} = 0000.000 & S[2] = 0.1100 \\
\hline
WS[2] = & 00.00000000 & & \\
WC[2] = & 00.00000000 & & \\
 & & & \\
4WS[2] = & 0000.00000000 & \widehat{S} = 0.1100 & s_3 = 0 \\
4WC[2] = & 0000.00000000 & \widehat{y} = 0000.000 & S[3] = 0.110000
\end{array}
$$

Since $w = 0$, the rest of the digits of S are 0. We perform 4 iterations to take into account the generation of the additional bit required for rounding. The radix-4 digits of the result are $s_0 = 1$, $s_1 = -1$, $s_2 = 0$, $s_3 = 0$, $s_4 = 0$ and $s_5 = 0$. The result is

$$s = (0.11000000)_2$$

Exercise 6.3

a) Use $S[j]$ in its original signed digit form

In this case it is not necessary the on-the-fly conversion of $S[j]$ for implementing the recurrence. Neverthless the register $K[j]$ is stil necessary. $F[j]$ is computed as

$$-S_{j+1}\left(2S[j]+S_{j+1}r^{-(j+1)}\right)$$

which requires a single concatenation of S_{j+1}, and a digit multiplication by S_{j+1}. Since $F[j]$ is represented in signed-digit form, the adder of the recurrence is more complex, that is, both operands are redundant.

b) Convert $S[j]$ to two's complement representation

The conversion is on-the-fly, and since this conversion is already necessary, it does not introduce additional complexity. The adder is simpler that in *a*) since one operand is in nonredundant form. More specifically the term $-S_{j+1}\left(2S[j]+S_{j+1}r^{-(j+1)}\right)$ is generated in nonredundant form as follows:

- $S_{j+1} \geq 0$

  Concatenate S_{j+1} to $2S[j]$ in position $j+1$. Set the most significant digit to one to have a negative operand (the weight of the most significant digit is negative). Then perform digit multiplication.

- $S_{j+1} < 0$

  In this case

  $$\left(2S[j]+S_{j+1}r^{-(j+1)}\right)=2\left(S[j]-r^{-j}\right)+(2r-S_{j+1})\,r^{-(j+1)}$$

  The term $S[j]-r^{-j}$ is available from the on-the-fly conversion module. The term $2r-S_{j+1}$ is precomputed for every digit and is concatenated to $2\left(S[j]-r^{-j}\right)$ in postion $j+1$. Finally, the digit multiplication is performed.

Exercise 6.5

a) Network for digit selection

Figure E6.5a shows the network for the selection of s_{j+1} and s_{j+2} in a radix-2 square root implementation using two radix-2 overlapped stages.

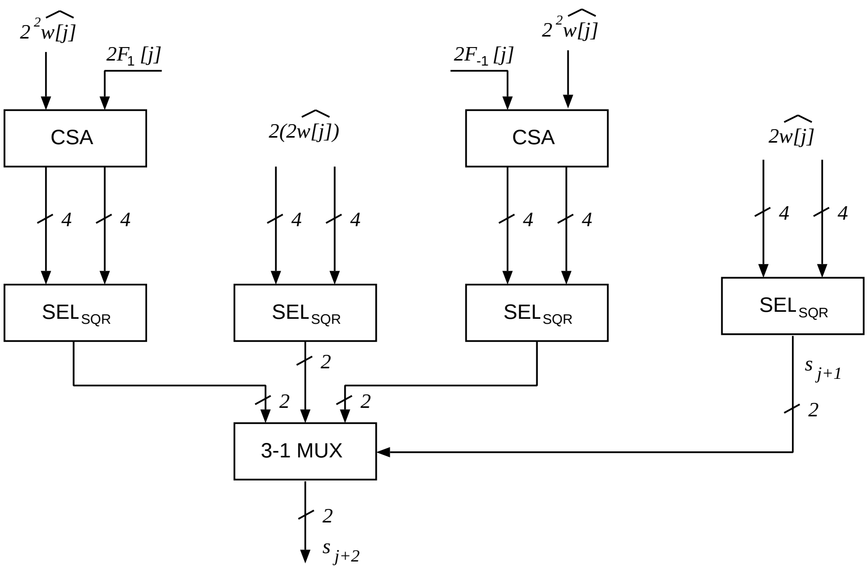

Figure E6.5a: Network for digit selection.

b) Network to produce the next residual

In Figure E6.5b the network producing the next residual is illustrated.

c) Delay analysis

– *Conventional implementation*

Computing the delay in the critical path we have

$$t_{cycle} = t_{SELSQRT}(4) + t_{buff}(1) + t_{mux}(1) + t_{HA}(1) + t_{reg}(2) = 9t_g$$

The latency of the conventional implementation (8 fractional bits) can be computed as $8 \times t_{cycle} = 8 \times 9t_g = 72t_g$.

– *Overlapped implementation*

Computing the delay in the critical path we have that the delay to produce $W[j+1]$ (that is, the delay from $W[j]$ to $W[j+1]$) is

$$t_{SELSQRT}(4) + t_{buff}(1) + t_{mux}(1) + t_{HA}(1) = 7t_g$$

Moreover, the delay to produce s_{j+2} (delay of CSA + delay of selection network + delay of 3-1 multiplexer) is

$$t_{CSA}(2) + t_{SELSQRT}(4) + t_{mux}(1) + t_{buff}(1) = 8t_g$$

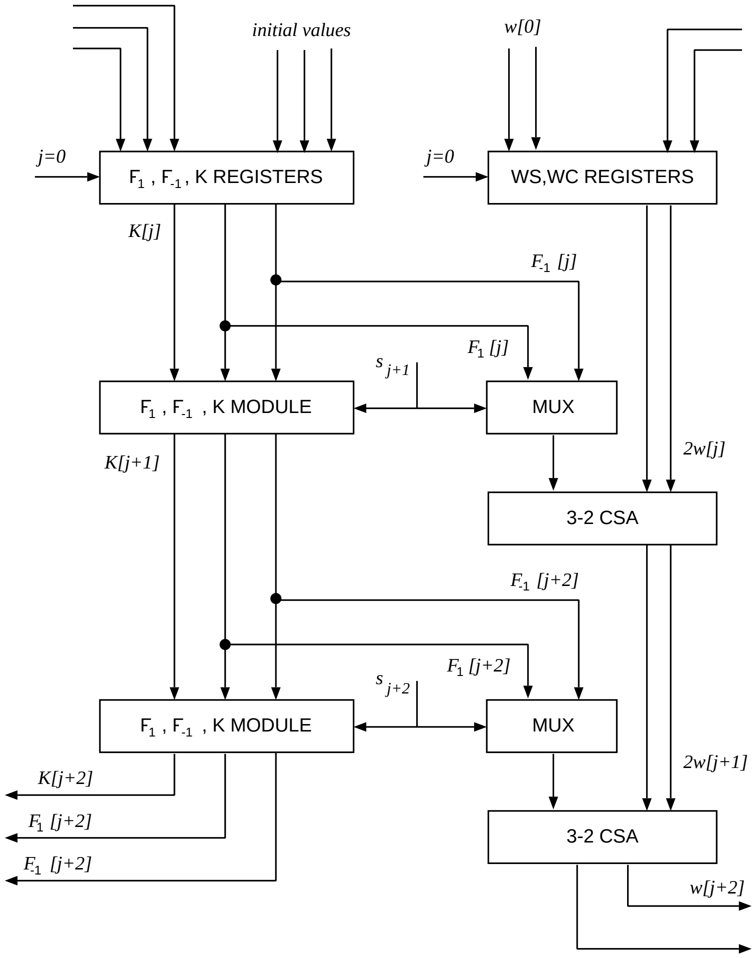

Figure E6.5b: Network to produce the next residual.

Finally, the delay to produce $W[j+2]$ (delay to produce s_{j+2} + delay of buffer + delay of mux + delay of HA) can be computed as

$$8t_g + 1t_g + 1t_g + 1t_g = 12t_g$$

Adding the register delay we get $t_{cycle} = 11t_g + 2t_g = 13t_g$. Computing the latency of the overlapped implementation (8 fractional bits) we get $4 \times t_{cycle} = 4 \times 13t_g = 52t_g$

Exercise 6.8

We compute the radix-4 square root of $x = (53)_{10} = (00110101)_2$. Since $n = 8$, we perform a right-shift of $m = 2$ bits and produce $x^* = .11010100$.

The number of bits of the integer result is $\frac{8-2}{2} = 3$. Consequently, two radix-4 iterations are necessary. We have $S[0] = 1$ and $w[0] = x^* - 1 = 11.11010100$.

Note that no alignment to digit boundary is needed, since the square root algorithm does not require to compute a remainder.

The iterations are as follows:

$$\begin{array}{rl l l l}
4WS[0] = & 1111.01010000 & & & \\
4WC[0] = & 0000.00000000 & \widehat{y} = 1111.0101 & s_1 = -1 & S[1] = 0.11 \\
F_{-1}[0] = & 001.11000000 & & & \\
\hline
WS[1] = & 10.10010000 & & & \\
WC[1] = & 10.10000000 & & & \\
 & & & & \\
4WS[1] = & 1010.01000000 & & & \\
4WC[1] = & 1010.00000000 & \widehat{y} = 0100.0100 & s_2 = 2 & S[2] = 0.1110
\end{array}$$

We do not need to compute $w[2]$. Therefore the result is

$$s = 2^3(0.111) = 111 = (7)_{10}$$

Exercise 6.13

We develop a radix-4 selection function for $J = 3$, $t = 3$ and $\delta = 4$.

- $k > 0$

$$\min\left(U_{k-1}\left(I_i\right)\right) = 2 \times \left(\frac{1}{2} + i \times 2^{-4}\right) \times \left(k - \frac{1}{3}\right)$$

$$\max\left(L_k\left(I_i\right)\right) = 2 \times \left(\frac{1}{2} + (i+1) \times 2^{-4}\right) \times \left(k - \frac{2}{3}\right)$$

- $k \leq 0$

$$\min\left(U_{k-1}\left(I_i\right)\right) = 2 \times \left(\frac{1}{2} + (i+1) \times 2^{-4}\right) \times \left(k - \frac{1}{3}\right)$$

$$\max\left(L_k\left(I_i\right)\right) = 2 \times \left(\frac{1}{2} + i \times 2^{-4}\right) \times \left(k - \frac{2}{3}\right) + \left(k - \frac{2}{3}\right)^2 \times 4^{-4}$$

$$\widehat{L}_k = \max\left(\lceil L_k\left(I_i\right)\rceil_3\right) \leq m_k\left(i\right) \leq \min\left(\lfloor U_{k-1}\left(I_i\right)\right) - 2^{-3}\rfloor_3 = \widehat{U}_{k-1}$$

To improve the presentation of results, we use a bound for $\max\left(L_k\left(I_i\right)\right)$. More specifically, we want an upper bound of the term $\left(k - \frac{2}{3}\right)^2 \times 4^{-4}$. For $k = 0$ we have $\frac{4}{9} \times 4^{-4} = \frac{1}{576} < \frac{1}{512}$. For $k = 1$ we have $\left(-\frac{5}{3}\right)^2 \times 4^{-4} = \frac{25}{2304} < \frac{1}{64}$.

The selection constants are presented in Table E6.13. Note that we give only half of the table (for $\widehat{S}[j] = 8, 9, 10, 11$) since there is an interval $\widehat{U}_{-2} - \widehat{L}_{-1}$ that is negative. Consequently, there is no selection function for $t = 3$ and $\delta = 4$.

$\widehat{S}[j]$	8	9	10	11
$\widehat{L}_2, \widehat{U}_1$	12, 12	14, 14	15, 15	16, 17
m_2	12	14	15	16
$\widehat{L}_1, \widehat{U}_0$	3, 4	4, 5	4, 5	4, 6
m_1	4	4	4	4
$\widehat{L}_0, \widehat{U}_{-1}$	$-5, -4$	$-5, -5$	$-6, -5$	$-7, -5$
m_0	-4	-5	-6	-6
$\widehat{L}_{-1}, \widehat{U}_{-2}$	$-13, -13$	$-14, -15$	$-16, -16$	$-18, -17$
m_{-1}	-13	X	-16	-18

Table E6.13: Selection interval and m_k constants.
$\widehat{S}[j]$: real value= shown value/16.
$\widehat{L}_k$, $\widehat{U}_{k-1}$ and m_k: real value = shown value/8.