DIGITAL ARITHMETIC
Miloš D. Ercegovac and Tomás Lang
Morgan Kaufmann Publishers, an imprint of Elsevier Science, ©2004
– Updated: September 9, 2003 –

Chapter 5: Solutions to Selected Exercises

– With contributions by Elisardo Antelo and Fabrizio Lamberti –

Exercise 5.2

In the following, two iterations of the division recurrence using a radix-16 implementation with two overlapped radix-4 stages for $x = 0.1001001110100101$ and $d = 0.110$ are shown.

- First iteration

$$\begin{array}{rl} 4WS[0] = & 000.1001001110100101 \\ 4WC[0] = & 000.0000000000000001^{*} \quad \hat{y}[0] = \frac{9}{16} \quad q_1 = 1 \\ -q_1 d = & 111.0011111111111111 \\ \hline WS[1] = & 111.1010110001011011 \\ WC[1] = & 000.0010011101001010 \end{array}$$

Speculative computations

– *Case a)* $q_1 = 2$

$$\begin{array}{rl} 4^2\widehat{WS}[0] & 010.01001 \\ 4^2\widehat{WC}[0] & 000.00000 \\ 4 \times (-2 \times d) & 001.11111 \\ \hline & 011.1011 \\ & 000.1001 \\ \hline & 100.0100 \quad \hat{y}[1] = -60/16 \quad \widehat{q_2} = -2 \text{ (tentative)} \end{array}$$

– *Case b)* $q_1 = 1$

$$\begin{array}{rl} 4^2\widehat{WS}[0] & 010.01001 \\ 4^2\widehat{WC}[0] & 000.00000 \\ 4 \times (-1 \times d) & 100.11111 \\ \hline & 110.1011 \\ & 000.1001 \\ \hline & 111.0100 \quad \hat{y}[1] = -12/16 \quad \widehat{q_2} = -1 \text{ (tentative)} \end{array}$$

– *Case c)* $q_1 = 0$

$$\begin{array}{rll} 4^2\widehat{WS}[0] & 010.0100 & \\ 4^2\widehat{WC}[0] & 000.0000 & \\ \hline & 010.0100 & \hat{y}[1] = 36/16 \quad \widehat{q}_2 = 2 \text{ (tentative)} \end{array}$$

– *Case d)* $q_1 = -1$

$$\begin{array}{rll} 4^2\widehat{WS}[0] & 010.01001 & \\ 4^2\widehat{WC}[0] & 000.00000 & \\ 4 \times (-1 \times d) & 011.00000 & \\ \hline & 001.0100 & \\ & 100.0000 & \\ \hline & 101.0100 & \hat{y}[1] = -44/16 \quad \widehat{q}_2 = -2 \text{ (tentative)} \end{array}$$

– *Case e)* $q_1 = -2$

$$\begin{array}{rll} 4^2\widehat{WS}[0] & 010.01001 & \\ 4^2\widehat{WC}[0] & 000.00000 & \\ 4 \times (-2 \times d) & 110.00000 & \\ \hline & 100.0100 & \\ & 100.0000 & \\ \hline & 000.0100 & \hat{y}[1] = 4/16 \quad \widehat{q}_2 = 0 \text{ (tentative)} \end{array}$$

Since $q_1 = 1$, we select case b). Therefore we have $q_2 = -1$. We can complete the first iteration as follows:

$$\begin{array}{rl} 4WS[1] = & 110.1011000101101100 \\ 4WC[1] = & 000.1001110100101000 \\ -q_2 d = & 000.1100000000000000 \\ \hline WS[2] = & 110.1110110001000100 \\ WC[2] = & 001.0010001001010000 \end{array}$$

- Second iteration

$$\begin{array}{rll} 4WS[2] = & 011.1011000100010000 & \\ 4WC[2] = & 100.1000100101000000 & \hat{y}[2] = \frac{3}{16} \quad q_3 = 0 \\ -q_3 d = & 000.0000000000000000 & \\ \hline WS[3] = & 111.0011100001010000 & \\ WC[3] = & 001.0000001000000000 & \end{array}$$

Speculative computations

– *Case a)* $q_3 = 2$

$$\begin{array}{rll} 4^2\widehat{WS}[2] & 110.11000 & \\ 4^2\widehat{WC}[2] & 010.00100 & \\ 4 \times (-2 \times d) & 001.11111 & \\ \hline & 101.0001 & \\ & 101.1100 & \\ \hline & 010.1101 & \hat{y}[3] = 45/16 \quad \widehat{q}_4 = 2 \text{ (tentative)} \end{array}$$

– *Case b)* $q_3 = 1$

$4^2\widehat{WS}[2]$	110.11000		
$4^2\widehat{WC}[2]$	010.00100		
$4 \times (-1 \times d)$	100.11111		
	000.0001		
	101.1100		
	101.1101	$\hat{y}[3] = -35/16$	$\widehat{q}_4 = -2$ (tentative)

– *Case c)* $q_3 = 0$

$4^2\widehat{WS}[2]$	110.1100		
$4^2\widehat{WC}[2]$	010.0010		
	000.1110	$\hat{y}[3] = 14/16$	$\widehat{q}_4 = 1$ (tentative)

– *Case d)* $q_3 = -1$

$4^2\widehat{WS}[2]$	110.11000		
$4^2\widehat{WC}[2]$	010.00100		
$4 \times (-1 \times d)$	011.00000		
	111.1110		
	100.0000		
	011.1110	$\hat{y}[3] = 62/16$	$\widehat{q}_4 = 2$ (tentative)

– *Case e)* $q_3 = -2$

$4^2\widehat{WS}[2]$	110.11000		
$4^2\widehat{WC}[2]$	010.00100		
$4 \times (-2 \times d)$	110.00000		
	010.1110		
	100.0000		
	110.1110	$\hat{y}[3] = -18/16$	$\widehat{q}_4 = -1$ (tentative)

Since $q_3 = 0$ we select case c). Therefore we have $q_4 = 1$. We can complete the second iteration as follows:

$$
\begin{array}{rl}
4WS[3] = & 100.1110000101000000 \\
4WC[3] = & 100.0000100000000001^{*} \\
-q_4 d = & 111.0011111111111111 \\
\hline
WS[4] = & 111.1101011010111110 \\
WC[4] = & 000.0101001010000010
\end{array}
$$

The digits of the result are $q_1 = 1$, $q_2 = -1$, $q_3 = 0$ and $q_4 = 1$. Therefore, we have $q = 00110001$.

Exercise 5.5

Let $Q[j]$ be the digit vector of the converted quotient consisting of the j most-significant digits, that is

$$Q[j] = \sum_{i=1}^{j} q_i r^{-i}$$

We have $Q[j+1] = Q[j] + q_{j+1} r^{-(j+1)}$. Since we are considering a radix-2 positive redundant representation with $q_i \in \{0,1,2\}$, we can use the following algorithm for the addition:

$$Q[j+1] = \begin{cases} Q[j] + q_{j+1} 2^{-(j+1)} & \text{if} \quad q_{j+1} \leq 1 \\ Q[j] + 2^{-j} & \text{if} \quad q_{j+1} = 2 \end{cases}$$

This algorithm has the disadvantage that the addition $Q[j] + 2^{-j}$ requires the propagation of a carry and therefore it is slow. To avoid this propagation we define $QP[j]$ with value

$$QP[j] = Q[j] + 2^{-j}$$

Using this second form, the conversion algorithm is

$$Q[j+1] = \begin{cases} Q[j] + q_{j+1} 2^{-(j+1)} & \text{if} \quad q_{j+1} \leq 1 \\ QP[j] & \text{if} \quad q_{j+1} = 2 \end{cases}$$

It is necessary to update also the form $QP[j]$, as follows:

$$QP[j+1] = Q[j+1] + 2^{-(j+1)} = \begin{cases} Q[j] + 2^{-(j+1)} & \text{if} \quad q_{j+1} = 0 \\ Q[j] + (1 + q_{j+1})\, 2^{-(j+1)} & \text{if} \quad q_{j+1} = 1 \\ QP[j] + 2^{-(j+1)} & \text{if} \quad q_{j+1} = 2 \end{cases}$$

Using the definition of $QP[j]$, the expression for $QP[j+1]$ when $q_{j+1} = 1$ can be rewritten as follows:

$$Q[j] + (1 + q_{j+1})\, 2^{-(j+1)} = Q[j] + 2^{-j} = QP[j]$$

Therefore, the expression $QP[j+1]$ when $q_{j+1} = 1$ and $q_{j+1} = 2$ can be condensed as follows:

$$QP[j+1] = QP[j] + (q_{j+1} - 1)\, 2^{-(j+1)} \quad \text{if} \quad q_{j+1} \geq 1$$

In conclusion, the algorithm for $QP[j+1]$ can be rewritten as follows:

$$QP[j+1] = \begin{cases} Q[j] + 2^{-(j+1)} & \text{if} \quad q_{j+1} = 0 \\ QP[j] + (q_{j+1} - 1)\, 2^{-(j+1)} & \text{if} \quad q_{j+1} \geq 1 \end{cases}$$

All the additions are now expressed by means of concatenations and no carry is propagated. In terms of concatenations, the on-the-fly conversion algorithm for a radix-2 positive redundant representation with digit set $\{0,1,2\}$ is

$$Q[j+1] = \begin{cases} (Q[j], q_{j+1}) & \text{if} \quad q_{j+1} \leq 1 \\ (QP[j], 0) & \text{if} \quad q_{j+1} = 2 \end{cases}$$

$$QP[j+1] = \begin{cases} (Q[j], 1) & \text{if} \quad q_{j+1} = 0 \\ (QP[j], q_{j+1} - 1) & \text{if} \quad q_{j+1} \geq 1 \end{cases}$$

with the initial conditions $Q[0] = 0$ and $QP[0] = 1$.

As an example, consider the conversion into conventional representation of the result 10211202.

j	q_j	$Q[j]$	$QP[j]$
0		0	1
1	1	0.1	1.0
2	0	0.10	0.11
3	2	0.110	0.111
4	1	0.1101	0.1110
5	1	0.11011	0.11100
6	2	0.111000	0.111001
7	0	0.1110000	0.1110001
8	2	0.11100010	0.11100011

Exercise 5.7

a) Implementation

An implementation of the retimed digit recurrence division (radix-4 with carry-save adder) is illustrated in Figure E5.7a. Details regarding the size of the most significant slice are presented in Figure E5.7b.

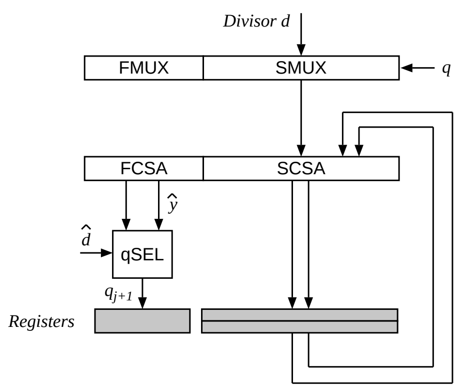

Figure E5.7a: Retimed implementation.

b) Delay analysis

– *Conventional design*

Computing the delay in the critical path we have (from Figure 5.4)

$$t_{cycle} = t_{qsel}(10.8) + t_{buff}(1.8) + t_{mux}(1.8) + t_{HA}(2.2) + t_{reg}(4).$$

Therefore, $t_{cycle} = 21t_{nand2}$. The number of iteration for IEEE double precision operands ($\rho < 1$) is $\lceil \frac{53+1+2}{2} \rceil = 28$. The latency of the conventional implementation can be computed as $(28+1) \times t_{cycle} = 29 \times 21t_{nand2} = 609t_{nand2}$.

– *Retimed version*

Computing the delay in the critical path (fast part) we have

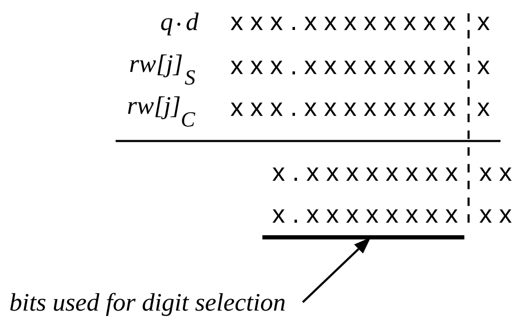

Figure E5.7b: Size of the most significant part of the path (size of FCSA is 7 bits, size of FMUX is 8 bits).

$$t_{cycle} = \quad t_{buff}(1.8) \times \frac{40}{100} + + (t_{mux}(1.8) + t_{HA}(2.2)) \times \frac{80}{100} + t_{qsel}(10.8) + t_{reg}(4)$$

Therefore, $t_{cycle} = 19t_{nand2}$. Computing the latency of the retimed version we get $(28 + 1 + 1) \times t_{cycle} = 30 \times 19t_{nand2} = 570t_{nand2}$.

Exercise 5.10

We normalize d to produce $d^* = 10010000 = 2^m d$ with $m = 4$. We define $d_f = d^* \times 2^{-n}$, where $n = 8$ is the number of bits of the operands. Assuming a redundant quotient digit-set with $q_i \in \{-2, -1, 0, 1, 2\}$, the redundancy factor is $\rho = \frac{a}{r-1} = \frac{2}{3}$. Since $\rho < 1$, we have $v = 2$. In order to obtain a correct remainder, the last digit of the quotient has to be aligned with a radix-4 boundary. For this, it must be $(m + v + s) \bmod k = 0$. Therefore we have $(4 + 2 + s) \bmod 2 = 0$ (with $k = 2$ and $m = 4$) and $s = 0$. We define $x_f = x \times 2^{-n}$ (as for the divisor). To achieve the required alignment, we shift x_f right by $v + s = 2$ bits. The initial condition is therefore

$$w[0] = \frac{x_f}{4} = .0001111000$$

Moreover, since the truncated divisor $\widehat{d} = 0.1001 = \frac{9}{16}$, we can compute $i = 16\widehat{d} = 9$. The corresponding selection constants are given by the following table:

i	8	9	10	11	12	13	14	15
$m_2(i)^+$	12	14	15	16	18	20	20	24
$m_1(i)^+$	4	4	4	4	6	6	8	8
$m_0(i)^+$	-4	-6	-6	-6	-8	-8	-8	-8
$m_{-1}(i)^+$	-13	-15	-16	-18	-20	-20	-22	-24

Finally, we can compute the number of iteration, $N = \lceil \frac{m+v}{k} \rceil$. Here $k = 2$ (since $r = 2^k$ where r is the radix of the quotient digit as produced by the division algorithm) and we get $N = \lceil \frac{4+2}{2} \rceil = 3$.

$4WS[0] =$	000.01111000		
$4WC[0] =$	000.00000001*	$\widehat{y}[0] = 000.0111 = \frac{7}{16}$	$q_1 = 1$
$-d_f =$	111.01101111		
$WS[1] =$	111.00010110		
$WC[1] =$	000.11010010		
$4WS[1] =$	100.01011000		
$4WC[1] =$	011.01001000	$\widehat{y}[1] = 111.1001 = -\frac{7}{16}$	$q_2 = \overline{1}$
$+d_f =$	000.10010000		
$WS[2] =$	111.10000000		
$WC[2] =$	000.10110000		
$4WS[2] =$	110.00000000		
$4WC[2] =$	010.11000001*	$\widehat{y}[2] = 000.1100 = \frac{12}{16}$	$q_3 = 1$
$-d_f =$	111.01101111		
$WS[3] =$	011.10101110		
$WC[3] =$	100.10000010		

Since $w[3] > 0$ the correction step is not needed. The quotient and the remainder are

$$q = 1\overline{1}1 = (13)_{10}$$
$$rem = w[3] \times 2^{n \log_2 2 - m} = w[3] \times 2^4 = 11 = (3)_{10}$$

Exercise 5.12

For signed-digit representation of the residual we get

$$\epsilon_{min} = -2^{-t} + ulp \quad e_{max} = 2^{-t} + ulp$$

and

$$L_k^* = L_k - e_{min} = L_k + 2^{-t} - ulp$$
$$U_k = U_k - e_{max} = U_k - 2^{-t} + ulp$$

resulting in

$$\widehat{U}_{k-1} = \lfloor U_{k-1}^* + 2^{-t} \rfloor_t = \lfloor U_{k-1} \rfloor_t$$
$$\widehat{L}_k = \lceil L_k^* \rceil_t = \lceil L_k + 2^{-t} \rceil_t$$

For a necessary condition on δ and t (for $k > 0$) we get

$$U_{k-1}(d_i) - L_k(d_{i+1} + 2^{-t} \geq 0$$

that is,

$$(k - 1 + \rho)d_i - ((k - \rho)(d_i + 2^{-\delta}) + 2^{-t}) \geq 0$$

The worst case is for $k = a$ and $d_i = 1/2$ resulting in

$$\frac{2\rho - 1}{2} - (a - \rho)2^{-\delta} \geq 2^{-t}$$

which is the same as for carry-save representation of the residual (expression 5.101). For radix 2 ($\rho = a = 1$ we get $t \geq 1$ and it is possible to use the same constant for the whole range of the divisor. We use $t = 1$ and obtain

$$\widehat{U}_0(1/2) = 1/2 \quad \widehat{U}_{-1}(1) = 0$$

$$\widehat{L}_1(1) = 1/2 \qquad \widehat{L}_0(1/2) = 0$$

Consequently, we get $m_1 = 1/2$ and $m_0 = 0$.

The range of the estimate $\hat{y}$ is

$$\lfloor -r\rho - (2^{-t} - ulp)\rfloor_t \leq r\rho + 2^{-t} - ulp\rfloor_t$$

which for $r = 2$ and $\rho = 1$ results in

$$-2 \leq \hat{y} \leq 2$$

The selection function is then

$$q_{j+1} = \begin{cases} 1 & \text{if } 1/2 \leq \hat{y} \leq 2 \\ 0 & \text{if } \;\; \hat{y} = 0 \\ -1 & \text{if} -2 \leq \hat{y} \leq -1/2 \end{cases}$$

The execution for $x = 128 \times 2^{-8}$ and $d = 6 \times 2^{-3}$ is as follows:

$2W[0] =$	0.10000000	$\widehat{y}[0] = 0.5$	$q_1 = 1$
$-q_1 d =$	$0.\bar{1}\bar{1}000000$		
$2W[1] =$	$0.\bar{1}000000$	$\widehat{y}[1] = -0.5$	$q_2 = -1$
$-q_2 d =$	0.11000000		
$2W[2] =$	0.10000000	$\widehat{y}[2] = 0.5$	$q_3 = 1$
$-q_3 d =$	$0.\bar{1}\bar{1}000000$		
$2W[3] =$	$0.\bar{1}0000000$	$\widehat{y}[3] = -0.5$	$q_4 = -1$

Since the pattern is periodic (and final residual is negative) we get

$$q = 2(0.1\bar{1}1\bar{1}1\bar{1}1\bar{1}0 = 0.10101010$$

Exercise 5.14

From expression 5.100, we obtain the lower bound for t and δ by requiring

$$U_{k-1}(d_i) - 2^{-t} - L_k(d_{i+1}) \geq 0$$

Using the definitions of L_k and U_k and considering the worst case condition $d_i = \frac{63}{64}$ (for a range of the divisor restricted to $\left[\frac{63}{64}, 1\right)$) and $k = a = 2$ (since $\rho = \frac{2}{3}$) we get

$$2^{-\delta} \leq \frac{3}{4} \times \left(\frac{21}{64} - 2^{-t}\right)$$

If we try $t = 2$ we get $2^{-\delta} \leq \frac{15}{256}$. We can use $\delta \geq 5$. In this case, if we use $\delta = 5$ we don't have dependence on d in the selection function since the interval of d is of width 2^{-6}.

We compute the selection intervals for $t = 2$. For $k = 2$ we get $\widehat{L}_2 = \lceil L_2 \rceil_2$ and $\widehat{U}_1 = \lfloor (U_1 - 2^{-t}) \rfloor_2$. Since $L_2 = \left(2 - \frac{2}{3}\right) \times 1 = \frac{4}{3}$ and $U_1 = \left(1 + \frac{2}{3}\right) \times \frac{63}{64} = \frac{5}{3} \times \frac{63}{64}$ we get $\widehat{L}_2 = \frac{6}{4}$ and $\widehat{U}_1 = \frac{5}{4}$. Being $\widehat{L}_2 \geq \widehat{U}_1$, $t = 2$ is not a possible solution.

We select $t = 3$. The corresponding selection intervals and selection contants are presented in Table E5.14.

Only one fractional bit of $\widehat{y}$ is necessary for the selection function. A possible implementation is presented in Figure E5.14.

Exercise 5.17

$[d_i, d_{i+1})$	$\left[\frac{63}{64}, 1\right)$
$\widehat{L}_2(d_{i+1}), \widehat{U}_1(d_i)^+$ $m_2(i)$	$11, 12$ 12
$\widehat{L}_1(d_{i+1}), \widehat{U}_0(d_i)^+$ $m_1(i)$	$3, 4$ 4
$\widehat{L}_0(d_{i+1}), \widehat{U}_{-1}(d_i)^+$ $m_0(i)$	$-5, -4$ -4
$\widehat{L}_{-1}(d_{i+1}), \widehat{U}_{-2}(d_i)^+$ $m_{-1}(i)$	$-13, -12$ -12

Table E5.14: Selection interval and m_k constants. Note: $^+$: real value= shown value/8

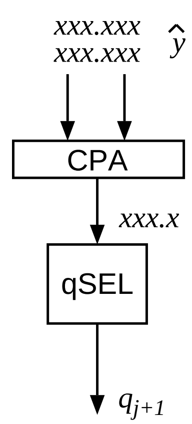

Figure E5.14: Implementation of the digit selection block.

a) Range of the divisor

From expression 5.16 we have

$$w[j+1] = rw[j] - q_{j+1}d = rw[j] - q_{j+1} - q_{j+1}(d-1)$$

Since $|q_{j+1}| \leq a$ we get

$$-a(d-1) \leq -q_{j+1}(d-1) \leq a(d-1)$$

From the expression for quotient digit selection

$$q_{j+1} = integer(rw[j] + 0.5) \leq a$$

we have

$$-\frac{1}{2} < rw[j] - q_{j+1} < \frac{1}{2}$$

From expression 5.15 we have

$$|rw[j]| \leq r\rho d$$

We obtain the following bounds on the shifted residual

$$max\left(-a + \frac{1}{2}, -r\rho d\right) < rw[j] < min\left(a - \frac{1}{2}, r\rho d\right)$$

Since the most critical restriction is the positive bound, we get

$$\frac{1}{2} + a|(1-d)| < min\left(\frac{2a-1}{2r}, \rho d\right)$$

In this case, since $d > 1$, we have

$$\frac{1}{2} + a(d-1) < \frac{2a-1}{2r}$$

Solving for d we get

$$d < 1 + \frac{2a-r-1}{2ar}$$

and therefore for convergence it must be

$$\beta < \frac{1}{r} - \frac{(r+1)}{2ar}$$

b) Possible implementation

An implementation of a high radix digit recurrence division with scaling and selection by rounding for nonredundant residuals is presented in Figure E5.17.

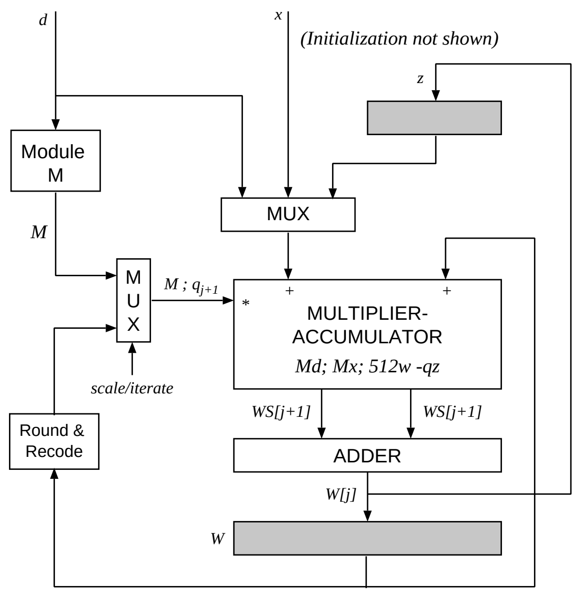

Figure E5.17: Implementation of a high radix division unit with scaling and selection by rounding (nonredundant residuals).

The hardware cost is higher in the high radix unit with respect to other low radix implementations due to the MAC block, additional registers and the module to compute the prescaling factor M. In the high radix unit, the number of cycles is reduced but t_{cycle} is larger. In the proposed implementation the speed-up with respect to other low radix implementations is limited by the nonredundant adder required to handle nonredundant residuals. To achieve a higher speed-up, we should consider a redundant representation of the residuals and a faster adder (see Chapter 5 for a fast implementation of a radix-512 division unit with residuals in carry-save form).

c) Example of execution for $r = 100$, $x = 0.83703960$ and $d = 1.00827040$

In the following we illustrate the method by finding the first three radix-r quotient digits. The recurrence is as follows:

$$w\left[j+1\right] = 100 \times w\left[j\right] - q_{j+1}d$$

The expression for quotient digit selection (for residuals in two's complement form) is

$$q_{j+1} = integer\left(100 \times w\left[j\right] + 0.5\right)$$

From a) we get

$$\beta < \frac{1}{r} - \frac{(r+1)}{2ar}$$

In this case, using $a = r - 1 = 99$ we get $\beta < 0.004899$. For convergence, it must be

$$1 \leq d \leq 1.004899$$

We compute the scaling constant $M = 1/1.005 \approx 0.995$. We scale the divisor thus obtaining $z = M \times d = 1.00322904$. We compute $M \times x$ and initialize $w[0] = M \times x = 0.83285440$.

$$\begin{aligned}
w[0] &= 0.83285440 \rightarrow q_1 = round(83.285440) = 83 \\
w[1] &= 100 \times 0.83285440 - 83 \times 1.00322904 = \\
&= 0.01742968 \rightarrow q_2 = round(1.742968) = 2 \\
w[2] &= 100 \times (0.01742968) - 2 \times 1.00322904 = \\
&= -0.26349008 \rightarrow q_3 = round(-26.349008) = -26
\end{aligned}$$