DIGITAL ARITHMETIC
Miloš D. Ercegovac and Tomás Lang
Morgan Kaufmann Publishers, an imprint of Elsevier Science, ©2004
– Updated: December 20, 2003 –

Chapter 11: Solutions to Exercises

Exercise 11.1

Compute $\sin(30^o)$ and $\cos(30^o)$ to a precision of seven bits 7 using the CORDIC algorithm.

The number of iterations performed depends on the datapath width, so that the angle becomes 0 for that width.

a) Datapath width of 7 fractional bits. We perform 7 iterations.

j	$z[j]$	σ_j	α_j	$x[j]$	$y[j]$
0	0.1000011	1	0.1100100	0.1001101	0.0000000
1	-0.0100001	-1	0.0111011	0.1001101	0.1001101
2	0.0011010	1	0.0011111	0.1110011	0.0100111
3	-0.0000101	-1	0.0001111	0.1101010	0.1000011
4	0.0001010	1	0.0000111	0.1110010	0.0110110
5	0.0000011	1	0.0000011	0.1101111	0.0111101
6	0.0000000	1	0.0000001	0.1101110	0.1000000
7				0.1101101	0.1000001

The angle decomposition is in radians. Values given in sign and magnitude.

The errors are $|\cos^o(30) - x[7]| = |0.866 - 0.852| = 0.014$ and $|\sin(30^o) - y[7]| = |0.5 - 0.508| = 0.008$

b) Datapath width of 10 fractional bits:

j	$z[j]$	σ_j	α_j	$x[j]$	$y[j]$
0	0.1000011000	1	0.1100100100	0.1001101101	0.0000000000
1	-0.0100001100	-1	0.0111011010	0.1001101101	0.1001101101
2	0.0011000111	1	0.0011111010	0.1110100011	0.0100110111
3	-0.0000101100	-1	0.0001111111	0.1101010110	0.1000011111
4	0.0001010011	1	0.0000111111	0.1110011001	0.0110110101
5	0.0000010100	1	0.0000011111	0.1101111111	0.0111101111
6	-0.0000001011	-1	0.0000001111	0.1101101111	0.1000001001
7	0.0000000100	1	0.0000000111	0.1101110111	0.0111111100
8	-0.0000000011	-1	0.0000000011	0.1101110100	0.1000000010
9	0.0000000000	1	0.0000000001	0.1101110110	0.0111111111
10				0.1101110101	0.1000000000

The result truncated to 7 fractional bits is

$$x[10] = 0.1101110 = 0.8594 \quad y[10] = 0.1000000 = 0.5$$

The errors are $|\cos(30^o) - x[10]| = |0.866 - 0.859| = 0.007$ and $|\sin(30^o) - y[10]| = |0.5 - 0.5| = 0$

c) we have not found a systematic solution method.

Exercise 11.3

The number of iterations performed depends on the datapath width, so that the last α_i becomes 0 for that width.

a) Datapath width of 7 fractional bits. We perform 7 iterations.

j	$y[j]$	σ_j	α_j	$z[j]$	$x[j]$
0	10.0010000	-1	0.1100100	0.0000000	11.0100000
1	-01.0010000	1	0.0111011	0.1100100	101.0110000
2	01.1001000	-1	0.0011111	0.0101001	101.1111000
3	00.0001010	-1	0.0001111	0.1001000	110.0101010
4	-00.1011011	1	0.0000111	0.1010111	110.0101011
5	-00.0101001	1	0.0000011	0.1010000	110.0110000
6	-00.0010000	1	0.0000001	0.1001101	110.0110001
7	-00.0000100			0.1001100	110.0110001

The angle decomposition is in radians. Values given in sign and magnitude.

The result values are $z[7] = 0.101100 = 0.580$ and $x[7] = 110.0110001 = 6.3828$.The compensated value is $x_R = x[7] \times 1/K[7] = 6.3828 \times 0.6072 = 3.876$.

The errors are $|\tan^{-1}(2.13/3.25) - z[7]| = |0.580 - 0.594| = 0.014$ and $modulus(2.13, 3.25) - x_R = 3.8856 - 3.876 = 0.009$

b) Datapath width of 10 fractional bits. We perform 10 iterations.

j	$y[j]$	σ_j	α_j	$z[j]$	$x[j]$
0	10.0010000101	-1	0.1100100100	0.0000000000	11.0100000000
1	-01.0001111011	1	0.0111011010	0.1100100100	101.0110000101
2	01.1001000111	-1	0.0011111010	0.0101001010	101.1111000010
3	00.0001010111	-1	0.0001111111	0.1001000100	110.0101010011
4	-00.1011010011	1	0.0000111111	0.1011000011	110.0101011101
5	-00.0100111110	1	0.0000011111	0.1010000100	110.0110001010
6	-00.0001110010	1	0.0000001111	0.1001100101	110.0110010011
7	-00.0000001100	1	0.0000000111	0.1001010110	110.0110010100
8	00.0000100111	-1	0.0000000011	0.1001001111	110.0110010100
9	00.0000001110	-1	0.0000000001	0.1001010010	110.0110010100
10	00.0000000010			0.1001010011	110.0110010100

The result truncated to 7 fractional bits is

$$z[10] = 0.1001010 = 0.578 \quad x[10] = 110.0110010 = 6.391$$

We compensate $x_R = x[10] \times 1/K[10] = 6.391 \times 0.6072 = 3.8806$

The errors are $|\tan^{-1}(2.13/3.25 - z[10]| = |0.580 - 0.578| = 0.002$ and $|modulus(2.13, 3.25) - x[10]| = 3.8856 - 3.8806 = 0.005$.

c) we have not found a systematic method to get a solution.

Exercise 11.4

Note that the sequence of α's should be decreasing. That is,

$$\alpha_{i+1} < \alpha_i \leq 2\alpha_{i+1}$$

From the definition of A and the values of s_i, we obtain that the range of A is

$$0 \leq A \leq A_{max} = \sum_{i=0}^{\infty} \alpha_i \tag{1}$$

The recurrent algorithm using $s_i \in \{0, 1\}$ converges iff for all j the residual value

$$W[j] = A - \sum_{i=0}^{j} s_i \alpha_i$$

is bounded by

$$0 \leq W[j] \leq \sum_{i=j+1}^{\infty} \alpha_i \tag{2}$$

From (1) and (2) we see that the algorithm converges while the values of s_i are all 1. Consider therefore the value $i = k$ for which the first $s_i = 0$ is selected. In the iteration

$$W[k+1] = W[k] - s_k \alpha_k$$

to have a non-negative residual $W[k+1]$, we need to make $s_{j+1} = 0$ when $W[k] \leq \alpha_k - ulp$. For the largest value $(\alpha_k - ulp)$ we get $W[k+1] = \alpha_k - ulp$. Moreover, to have convergence, from (2) we have

$$\alpha_k - ulp \leq \sum_{j=k+1}^{\infty} \alpha_j = \alpha_{k+1} + \sum_{j=k+2}^{\infty} \alpha_j \tag{3}$$

Now from the hypothesis $\alpha_i \leq 2\alpha_{i+1}$ we obtain

$$\alpha_i \leq \sum_{j=i+1}^{\infty} \alpha_j \tag{4}$$

This results from the well-known fact that if $a_i = 2a_{i+1}$ then $a_i = \sum_{j=i+1}^{\infty} a_j$. Introducing (4) in (3) we conclude that the algorithm converges.

For $s_i\{-1, 1\}$ we apply the same technique. Now the convergence condition is

$$|W[j]| \leq \sum_{i=j+1}^{\infty} \alpha_i$$

Again, the algorithm converges while $s_j = 1$. We choose $s_j = -1$ when $W[j] < 0$. The most negative value of $W[j]$ occurs when $W[j-1] = 0$. Consequently,

$$W[j] \geq -\alpha_{j-1}$$

So, for convergence,

$$\alpha_{j-1} \leq \sum_{i=j}^{\infty} \alpha_i$$

and the same proof as before follows.

Exercise 11.9

The Taylor series expansion of $\tan^{-1}(2^{-j})$ is

$$\tan^{-1}(2^{-j}) = 2^{-j} - \frac{2^{-3j}}{3} + \frac{2^{-5j}}{5} - ...$$

Consequently,

$$|\tan^{-1}(2^{-j}) - 2^{-j}| = \frac{2^{-3j}}{3} - \frac{2^{-5j}}{5} + ... \leq 2^{-n}$$

results in

$$j \geq J = \frac{n-1}{3}$$

This implies that for $j \geq J$ there is no need to store the value of $\tan^{-1}(2^{-j})$ in the table, since the value 2^{-j} can be used.

Exercise 11.12

According to Table 11.5 we perform hyperbolic CORDIC in vectoring mode with initial conditions $x_{in} = 1.17$, $y_{in} = -0.83$, and $z_{in} = 0$. Performing eight iterations with a datapath width of 8 bits, we obtain

j	$y[j]$	σ_j	α_j	$z[j]$	$x[j]$
1	-0.11010100	1	0.10001100	0.00000000	1.00101011
2	-0.00111111	1	0.01000001	-0.10001100	0.11000001
3	-0.00001111	1	0.00100000	-0.11001101	0.10110010
4	0.00000111	-1	0.00010000	-0.11101101	0.10110001
4	-0.00000100	1	0.00010000	-0.11011101	0.10110001
5	0.00000111	-1	0.00001000	-0.11101101	0.10110001
6	0.00000010	-1	0.00000100	-0.11100101	0.10110001
7	0.00000000	-1	0.00000010	-0.11100001	0.10110001
8	-0.00000001	1	0.00000001	-0.11011111	0.10110001
9	-0.00000001	-	0.00000000	-0.11100000	0.10110001

The result is $2z[10] = -1.11000000 = -1.75$. The error is $|\ln(0.17) - 2z[10]| = |-1.772 + 1.75| = 0.022$