

SOLUTIONS TO SELECTED PROBLEMS

Chapter 1:

- 1.1. (a) $(051.2)_8$, $(726.6)_8$, $(726.5)_8$;
(b) $(00101001.01)_2$, $(11010110.11)_2$, $(11010110.10)_2$.
- 1.3. $3.3n$.
- 1.4. (b) e .
- 1.6. For radix complement 0 or -1 , 0. For diminished-radix complement -0 , -0 .
- 1.7. $c_{in} \oplus c_{out}$.
- 1.9. The formula can not be extended but may be modified.
- 1.12. No.

Chapter 2:

- 2.1. (a) $(1\bar{1}\bar{2}1.10)_4$, $(\bar{1}1\bar{2}\bar{1}.\bar{1}0)_4$; (b) $(162.85)_{-10}$, $(059.35)_{-10}$.
- 2.2. Out of the 7 possible representations $(0\bar{1}0010)$ is minimal.
- 2.7. 3 and 4.
- 2.8. The additive inverse of $(08019)_{-10}$ is $(12001)_{-10}$.
- 2.9. Yes.
- 2.10. No.

Chapter 3:

- 3.2. This can be done but with an increase in the length of intermediate results.
- 3.3. $0\bar{1}100000\bar{1}$.
- 3.10. Not recommended.
- 3.12. $Q = 0.11\bar{1}11\bar{1} = 0.101101$.

Chapter 4:

4.1. $F_{max}^+ = (1 - 2^{-29}) 2^{511}$; $F_{min}^+ = 2^{-513}$
 The total number of different values is $2^{39} + 1$.

4.2. (a)

0	1001000	00100...00
---	---------	------------

;

0	10011110	00000...00
---	----------	------------

0	10011110	00000...00
---	----------	------------

(b)

1	1000000	0011111110...0
---	---------	----------------

;

1	01111110	1111110000...00
---	----------	-----------------

1	01111110	1111110000...00
---	----------	-----------------

4.3. (b) 1.5 times.

4.4. There are $2^{24} - 2$ denormalized numbers in the format. This is nearly equal to the total number of positive and negative normalized values with exponent 1.

4.5. All three are satisfied only in systems with denormalized numbers.

4.7. The average bias of the round-to-nearest even scheme is zero for positive and negative numbers. The average bias for the round-towards-zero scheme is

$$-\frac{1}{2^d} \left(\sum_{i=0}^{2^d-1} i(2^{-d}) \right) = -\left(\frac{2^d - 1}{2^{d+1}} \right) \quad \text{and} \quad \left(\frac{2^d - 1}{2^{d+1}} \right)$$

for positive and negative numbers, respectively. The average bias for the round-towards-infinity and the round-towards $-\infty$ schemes are similarly derived.

4.8.

	s	exponent	fraction	G	notes
Original number	0	00011111	11111111111111111111111111111111	1	
Round-towards-nearest-even	0	00100000	00000000000000000000000000000000	-	
Round-towards - 0	0	00011111	11111111111111111111111111111111	-	
Round-towards $+\infty$	0	00100000	00000000000000000000000000000000	-	
Round-towards $-\infty$	0	00011111	11111111111111111111111111111111	-	
Original number	0	11111110	11111111111111111111111111111111	1	
Round-towards-nearest-even	0	11111111	00000000000000000000000000000000	-	$+\infty$
Round-towards - 0	0	11111110	11111111111111111111111111111111	-	
Round-towards $+\infty$	0	11111111	00000000000000000000000000000000	-	$+\infty$
Round-towards $-\infty$	0	11111110	11111111111111111111111111111111	-	
Original number	1	11111110	11111111111111111111111111111111	1	
Round-towards-nearest-even	1	11111111	00000000000000000000000000000000	-	$-\infty$
Round-towards - 0	1	11111110	11111111111111111111111111111111	-	
Round-towards $+\infty$	1	11111110	11111111111111111111111111111111	-	
Round-towards $-\infty$	1	11111111	00000000000000000000000000000000	-	$-\infty$

4.10. No guard bits are necessary for addition or multiplication. For subtraction we need a guard bit, a round bit and a sticky bit. One guard bit is sufficient for division.

4.11. (a) The final result is 3F80 0000 for all the rounding schemes except the round towards $+\infty$ for which it is 3F80 0001.

(b) The result is 3800 0000 for all four rounding schemes

(d) The round-to-nearest-even, and round-towards-infinity yield 3F80 0000 while the round-towards $-\infty$ and truncation result in 3F7F FFFF.

4.12. No. For example, if $E_A - E_B > 24$.

4.13. The distance between a number and its successor is always the same.

4.14. (a) Both methods result in a relative error having the same order of magnitude.

(b) Parallel addition results in smaller relative error as compared to serial addition. We can achieve better error performance if the operands are in some order but the upper bound remains unchanged.

Chapter 5:

5.4. $T_{carry} = 7t_r$.

5.5. The estimated addition times for an 80-bit adder with 0, 1, 2 and 3 levels of a carry-look-ahead circuit (74182) are 265, 109, 64 and 50.75 nsec, respectively.

5.7. (a) A 16-bit conditional-sum adder can be designed using 16 (74183) ICs and 24 (74157 - with four 2-1 MUXs per chip) ICs. A 24-bit conditional-sum adder can be designed using 24 (74183) ICs and 39 (74157) ICs.

(b) The addition time estimates for 16-bit and a 24-bit conditional-sum adders are 80.5 and 97.0 ns, respectively. These are based on typical delays of 14.5 ns for 74183 and 16.5 ns for 74157. The corresponding estimates for carry-look-ahead adders (with the maximum number of carry-look-ahead levels) are 31.0 and 44.25 ns.

5.12. (a) (7,7,7,6).

5.13. 11.

5.15. The n -bit slices which can be designed include 11-bit and 15-bit slices.

Chapter 6:

6.5. (b) 18 (74261) ICs. Figure 6.5.1 depicts the generation of a partial product while Figure 6.5.2 shows the alignment of the six partial products and the additional bits for proper sign extension.

6.7. Either (5,3) or (5,5,4) counters can be used for the central $2n$ -bit portion.

6.14. The alternate multiplier requires 16 CASS cells and its execution time is $10\Delta_{CASS}$.

6.15. Twenty one cells of the type shown below (Figure 6.15.1) are required.

Chapter 7:

7.2. The algorithms in (a) and (c) require three add/subtract operations while the algorithm in (b) requires only two operations.

7.4. (a) For (i) the two methods require the same number of add/subtract operations. For (ii) the original algorithm requires fewer add/subtract operations. For (iii) the pro-

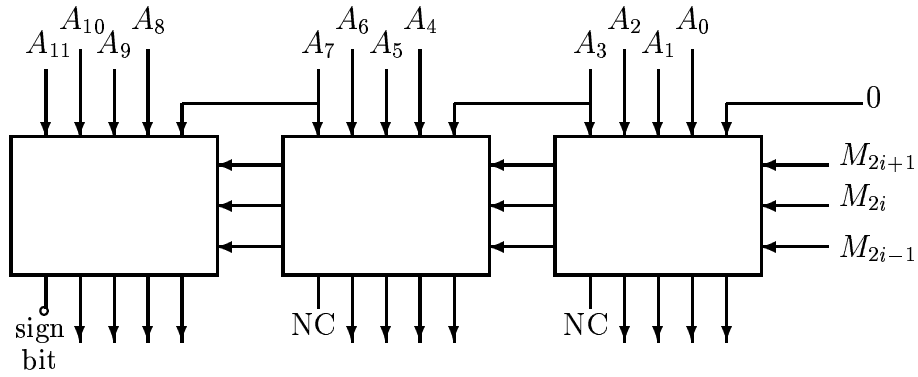


Figure 6.5.1 Partial product generation.

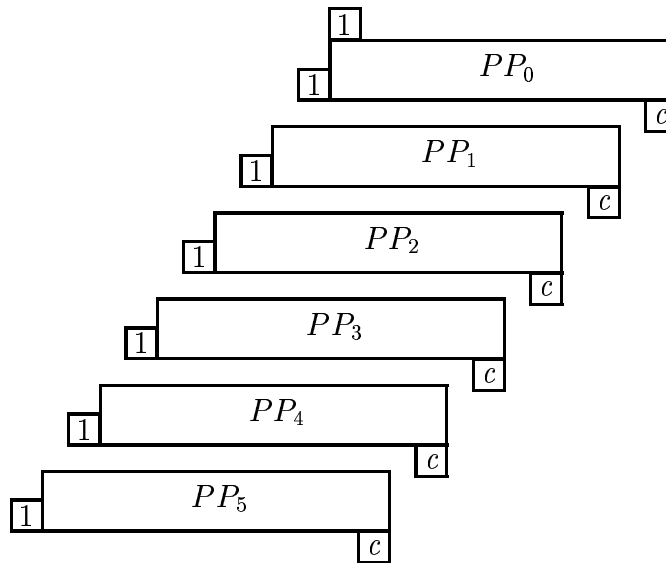


Figure 6.5.2 Alignment of partial products in a 12×12 multiplier.

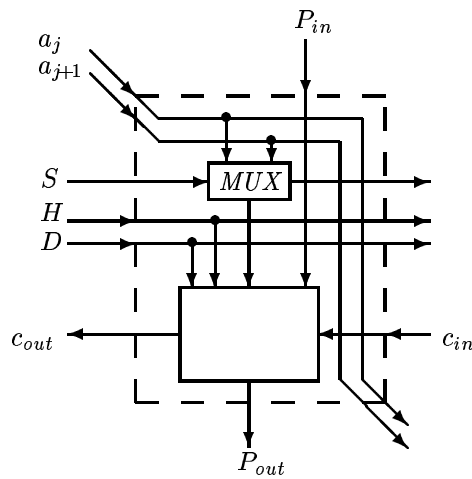


Figure 6.15.1 Controlled add/subtract/shift (CASS) cell.

posed method requires fewer add/subtract operations.

7.5. The smallest comparison constant is $K_1 = 3/8$ and the corresponding divisor subregion is $[0.100, 0.101]$.

7.8. The required divisor precision is 2^{-2} . The required remainder precision is 2^{-1} .

7.10. In only one case a higher divisor precision of 2^{-3} is required.