Assignmet 3

1. (Chapter 3) Represent X=−6 and A=−7 in 4-bit 2’s complement representation and then use sequential multiplication algorithm to obtain the product X·A. (12 marks)

Ans.) A= (-7)10 = ( 1 0 0 1 )2's complement

X= (-6)10 = ( 1 0 1 0 )2's complement

---------------------------------------------------------------------------

P(0)=0 0 0 0|

x0 =0 => Add 0 = 0 0 0 0 |

-----------------------------|------------------

0 0 0 0 |

Shift to get P(1) = 0 0 0 0 | 0

x1 = 1 => Add A = 1 0 0 1 |

------------------------- |----------------

1 0 0 1 | 0

Shift to get P(2) = 1 1 0 0 | 1 0

x2 = 0 => Add 0 = 0 0 0 0 |

------------------------- |----------------

1 1 0 0 | 1 0

Shift to get P(3) = 1 1 1 0 | 0 1 0

x3 = 1 => Add (-A) = 0 1 1 1 |

Correction term ------------------------ |-----------------

0 1 0 1 | 0 1 0 = 2-3 \* (42)

--------------------------

Hence, ( 1 0 0 1)2's complement \* (1 0 1 0 )2's complement = (0 0 1 0 1 0 1 0)2's complement

**(-7)10 \* (-6)10 = (42)10**

2. (Chapter 3) Let X = (0.110000)2 = 3/4 and D = (0.101)2 = 5/8. Use restoring division

algorithm to obtain the quotient Q = 0.q1 · · · qm and the remainder R = 2−m rm . Note: This

problem is a bit advanced. (12 marks)

Ans) Here, X > D. To perform restoring division, X < D

⇒ X = (0.110000)2 = (0.011000)2 \* 2 = (X / 2) \* 2 = X' \* 2

Let X = Q . D + R

⇒ X/ 2 = (Q / 2) . D + (R / 2) where X / 2 = X' < D

⇒X' = Q' . D + R' ⇒ Q = 2Q' and R = 2R'

Hence, we can perform the restoring division X' / D.

Here, -D in 2's complement = (11.011)2

r0 = X 0. 0 1 1 | 0 0 0

2r0 0 0. 1 1 0 | 0 0

Add -D 1 1. 0 1 1 |

------------------------------------------|----------

r1 = 2r0 - D ≥ 0 0 0. 0 0 1 | 0 0 set q0 = 1

2r1 0 0. 0 1 0 | 0

Add -D 1 1. 0 1 1 |

----------------------------------------- |-----------

2r1 - D < 0 1 1. 1 0 1 | 0 set q1 = 0

r2 = 2r1 0 0. 0 1 0 | 0

2r2 0 0. 1 0 0 |

Add -D 1 1. 0 1 1 |

------------------------------------------|-----------

2r2 - D < 0 1 1. 1 1 1 | set q2 = 0

r3  =2r2 0 0. 1 0 0 |

⇒ Q' = (0.100)2 = 1/2 and r3 = 1/2 ⇒ R' = rm 2-m  = r3 2-3 = (0.0001)2 = 1/16

⇒ Q = 2 Q' = (1.00)2 = 1 and R = 2R' = (0.001)2 = 1/8

3. (Chapter 4) Find the normalized floating-point representations of the number 6400 (16 marks)

i). in the single-precision IEEE format;

ii). in the double-precision IEEE format.

Ans) i) => (6400)10 = (1100100000000)2 = 1.1001 \* 212  = (-1)0 \* (1.1001) \* 2139 - 127

In the single-precision IEEE format, F = (−1)s × 1.f × 2e−127 .

⇒ s = 0

e = 1000 1100

f = 1001 0000 0000 0000 000

ii) => (6400)10 = (1100100000000)2 = 1.10010000 \* 212  = (-1)0 \* (1.1001) \* 21035-1023

In the double-precision IEEE format, F = (-1)s × 1.f × 2e−1023

⇒s = 0

e = 1000 000 1011

f = 1001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

4. (Chapter 4) Find the value for the following IEEE single-precision representation. The final

result should be in the form 1.a × 2b , where 1.a is a decimal number with integer digit of 1

and four fractional digits, and b is a decimal integer. (16 marks)

Ans.) Here, s= 1, e = (1100 0010)2 = , f =

=> Result = (-1)1 \* (1.0111) \* 267

= - 1.0111 \* 267

5. (Chapter 4) Suppose that the input is X = x1x0 .x−1x−2 , and the output is an integer. List

the truth table and then draw a block diagram using adder and/or necessary logic gates for an

implementation of round-to-the-nearest scheme. (16 marks)

Ans.) Truth table for the round to nearest scheme is as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Round-to-nearest scheme with d = 2 | | | | | | | | |
| Input: X = x1x0.x-1x-2 | | | | |  | | Output: Y = round(X) | |
| x1 | x0 | x-1 | x-2 |  | | y1 | | y0 |
| 0 | 0 | 0 | 0 |  | | 0 | | 0. |
| 0 | 0 | 0 | 1 |  | | 0 | | 0. |
| 0 | 0 | 1 | 0 |  | | 0 | | 1. |
| 0 | 0 | 1 | 1 |  | | 0 | | 1. |
| 0 | 1 | 0 | 0 |  | | 0 | | 1. |
| 0 | 1 | 0 | 1 |  | | 0 | | 1. |
| 0 | 1 | 1 | 0 |  | | 1 | | 0. |
| 0 | 1 | 1 | 1 |  | | 1 | | 0. |
| 1 | 0 | 0 | 0 |  | | 1 | | 0. |
| 1 | 0 | 0 | 1 |  | | 1 | | 0. |
| 1 | 0 | 1 | 0 |  | | 1 | | 1. |
| 1 | 0 | 1 | 1 |  | | 1 | | 1. |
| 1 | 1 | 0 | 0 |  | | 1 | | 1. |
| 1 | 1 | 0 | 1 |  | | 1 | | 1. |
| 1 | 1 | 1 | 0 |  | | 0 | | 0. |
| 1 | 1 | 1 | 1 |  | | 0 | | 0. |

Then, x1o and x0o can be obtained as follows using Karnaugh maps:

x-1x-2 x-1 x-2

x1x0 0001 11 10x1 x0 00 01 11 10

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 00 | 0  0 | 0  1 | 0  3 | 0  2 | 00 | 0  0 | 0  1 | 1  3 | 1  2 |
| 01 | 0  4 | 0  5 | 1  7 | 1  6 | 01 | 1  4 | 1  5 | 0  7 | 0  6 |
| 11 | 1  12 | 1  13 | 0  15 | 0  14 | 11 | 1  12 | 1  13 | 0  15 | 0  14 |
| 10 | 1  8 | 1  9 | 1  11 | 1  10 | 10 | 0  8 | 0  9 | 1  11 | 1  10 |

⇒ y1 = x1' . x0. x-1 + x1 . x-1' + x1 . x0' = x1' ( x0 . x-1) + x1 (x-1 ' + x0 ' ) = x1' (x0 . x-1) + x1 ( x-1 . x0 )'

= x1' ⊕ (x-1 . x0)

and y0= x0 . x-1' + x0'. x-1 = x0 ⊕ x-1

x-1

=> Block diagram using adder and/or logic gates is as follows:

x0

x1 x0 x-1 x1 0 x0 x-1 x0 x-1 x1

<=> <=>

+

Full

Adder

Half

Adder

c1

y0

y1 y0 y1 y0 y1

(Cout  is ignored)

6. (Chapter 4) Find the maximal positive and negative errors and the bias for ROM rounding

scheme with l = 3 and d = 2. (12 marks)

Ans.) l = 3 and d=2 => Input (x) = x2x1x0 . x-1 x-2

|  |  |  |  |
| --- | --- | --- | --- |
| ROM scheme with l = 3 input bits | | | |
|  | Input: x | Output: ROM(x) | Error: ROM(x) - x |
| 1 | 000.00 | 000. | 0 |
| 2 | 000.01 | 000. | -1/4 |
| 3 | 000.10 | 001. | +1/2 |
| 4 | 000.11 | 001. | +1/4 |
| 5 | 001.00 | 001. | 0 |
| 6 | 001.01 | 001. | -1/4 |
| 7 | 001.10 | 010. | +1/2 |
| 8 | 001.11 | 010. | +1/4 |
| 9 | 010.00 | 010. | 0 |
| 10 | 010.01 | 010. | -1/4 |
| 11 | 010.10 | 010. | +1/2 |
| 12 | 010.11 | 011. | +1/4 |
| 13 | 011.00 | 011. | 0 |
| 14 | 011.01 | 011. | -1/4 |
| 15 | 011.10 | 100. | +1/2 |
| 16 | 011.11 | 100. | +1/4 |
| 17 | 100.00 | 100. | 0 |
| 18 | 100.01 | 100. | -1/4 |
| 19 | 100.10 | 101. | +1/2 |
| 20 | 100.11 | 101. | +1/4 |
| 21 | 101.00 | 101. | 0 |
| 22 | 101.01 | 101. | -1/4 |
| 23 | 101.10 | 110. | +1/2 |
| 24 | 101.11 | 110. | +1/4 |
| 25 | 110.00 | 110. | 0 |
| 26 | 110.01 | 110. | -1/4 |
| 27 | 110.10 | 111. | +1/2 |
| 28 | 110.11 | 111. | +1/4 |
| 29 | 111.00 | 111. | 0 |
| 30 | 111.01 | 111. | -1/4 |
| 31 | 111.10 | 111. | -1/2 |
| 32 | 111.11 | 111. | -3/4 |

\* It can be seen from the above truth table that the maximal errors e+max = +1/2 and e-max = -3/4

And, bias = 1/32 [ 7 ( 0 + (-1/4) + (1/2) + (1/4)) + (0 + (-1/4) + (-1/2) + (-3/4))]

= (1/32) \* (7/2 -3/2)

=> Bias = 1/16

7. (Chapter 5) Give two different designs for a 12-bit CLA (carry-lookahead adder) using 4-bit

CLA and/or carry generator as building blocks. One has one-level carry lookahead architec-

ture and the other uses two-level carry lookahead. Show a critical path and the critical path

delay for both designs. (16 marks)

Ans.) A 12-bit CLA with one-level carry lookahead architecture can be generated using 4-bit CLA

and carry generator as building blocks as follows:

Diagram 1) 12-bit CLA with one-level carry lookahead

c0

x11 y11 x10 y10  x9 y9 x8 y8 x7 y7 x6 y6  x5 y5 x4 y4 x3 y3 x2 y2  x1 y1 x0 y0

abd

s11 s10 s9 s8 s7 s6 s5 s4 s3 s2 s1 s0

g11 g10 g9 g8 g7 g6 g5 g4 g3 g2 g1 g0

p11 p10 p9 p8 p7 p6 p5 p4 p3 p2 p1 p0

c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1

Carry generator Carry generator Carry generator

c12 s11 s10 s9 s8 s7 s6 s5 s4 s3 s2 s1 s0

Here, gi = xi + yi ; pi = xi ⊕ yi ; .........................(1) => 1ΔG gate delay

ci+1 = gi + ci . pi ..................................................(2)=> 2ΔG gate delay

si = pi  ⊕ ci .........................................................(3) => 1ΔG gate delay

As can be seen from the figure, the critical path will be from x0/y0 /x1/y1 /x2/y2 /x3/y3 to s11/s10/s9/s8

Hence, critical path delay = 1ΔG + 2ΔG + 2ΔG + 2ΔG + 1ΔG  = 8ΔG

A 12-bit CLA with two-level carry lookahead architecture can be generated using 4-bit CLA

and carry generator as building blocks as follows:

Diagram 2) 12-bit CLA with two-level carry lookahead

X11-8 Y11-8 X7-4 Y7-4 X3-0 Y3-0

C0

4b CLA C8 4b CLA C4 4b CLA

C12 G8-11 S2 G4,7 S1 G0,3 S0

P8-11 P4-7 P0-3

Carry generator

Here, Gi,i+3 = gi+3 + gi+2 pi+3 + gi+1 pi+2 pi+3 + gi pi+1 pi+2pi+3 .....................(4) => 2ΔG  gate delay

Pi,i+3 = pi pi+1 pi+2 pi+3

C4 = G0,3 + P0,3 . C0

C8 = G4,7 + P4,7 . C4 = G4,7 + P4,7 . G0,3 + P4,7 .P0,3 .C0

C12 = G8-11 + P8,11 . C8 ( C12 is generated by the 4b CLA instead of Carry generator, hence, it

would depend on C8 and not C0 due to which it forms the critical path)

As can be seen from the figure, critical path will be from X11-8 / Y11-8 / X7-4 / Y7-4 /X3-0 /Y3-0 to C12

Hence, critical path delay = (1ΔG + 2ΔG )G and P from 4 b CLA  + (2ΔG)Carry generator  + (2ΔG ) Carry generator within 4b CLA

= 7ΔG