CDA3100: Computer Organization

Assignment 05

Assigned: 02/26/15 Due: 03/22/15

Problems From the Textbook (Graded)

3.14. (3×8)x4tu =96 time units for the hardware and (4×8)x4tu=128 time units for the software

3.15. 8 bits wide, it requires 7 adders so 7x4tu=21 time units.

3.16. 8 bits wide, it requires 7 adders in 3 levels so 3x4tu=12 time units

Additional Problems (Graded)

1. Use Boolean algebra to find the minimum cost SOP expression for the function below. (2)

𝑓 ( 𝑎 , 𝑏 , 𝑐 , 𝑑 ) = 𝑎 𝑏̅ 𝑐 + 𝑐 𝑑 + 𝑎̅ 𝑑 + 𝑏 𝑑

f = ab`c + cd +a`d+bd

f = acb`+cd+a`d = acb`+a`d+bd #using consensus

f = ab`c+a`bd

2. Use Boolean algebra to find the minimum cost POS expression for the function below. (2) 𝑓(𝑎,𝑏,𝑐,𝑑,𝑒)= ∏𝑀(0,8,9,13,14)

f = (a+b+c+d+e)(a+!b+c+d+e)(a+!b+c+d+!e)(a+!b+!c+d+!e)(a+!b+!b+!d+e)

f = [(a+c+d+e)+b]\*[(a+c+d+e)+!b] [(a+!b+d+!e)+c]\*[(a+!b+d+!e)+!c] (a+!b+!c+!d+e)

f = (a+c+d+e)(a+!b+d+e)(a+!b+!c+!d+e)

3. Answer the following questions using the circuit in Figure 1.

a) Derive the truth table for the circuit in Figure 1. (1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **F** |
| **0** | **0** | **0** | **0** | **1** |
| **0** | **0** | **0** | **1** | **0** |
| **0** | **0** | **1** | **0** | **1** |
| **0** | **0** | **1** | **1** | **1** |
| **0** | **1** | **0** | **0** | **1** |
| **0** | **1** | **0** | **1** | **0** |
| **0** | **1** | **1** | **0** | **1** |
| **0** | **1** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **0** | **1** | **0** |
| **1** | **0** | **1** | **0** | **1** |
| **1** | **0** | **1** | **1** | **0** |
| **1** | **1** | **0** | **0** | **1** |
| **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** | **1** |
| **1** | **1** | **1** | **1** | **0** |
|  |  |  |  |  |
|  |  |  |  |  |

b)  Derive the Boolean expression from the truth table (not simplified). (1)

f = (a+!b)(!cb)+d

c)  Redesign the circuit so that it only uses two-input NAND gates. (4)

4. Show how to implement the function below using a 3:8 decoder and an OR gate. (2) 𝑓(𝑎, 𝑏, 𝑐) = ∑ 𝑚(0,2,3,4,5,7)

5. Assume that the integers are unsigned. You must perform the operation in binary and show your work. Use ten bits of precision and state whether there was an overflow – justify your answer. (2)

188 +371 = 551

188 = 010111100

371 = 101110011

1111 (carry overs)

010111100

+ 101110011

= 1000101111

- No overflow because the most significant bit did not carry over using 10 bits of precision.

6. Assume that the integers are signed, 2’s compliment. You must perform the operation in binary and show your work. Use ten bits of precision and state whether there was an overflow – justify your answer. (2)

188 −371

188 = 10111100

371 = 101110011

10111100

* 101110011

= 1101001001

No overflow because 0 XOR with 1 gives you zero.

1. Assume that the integers are unsigned. You must perform the operation in binary and show your work. Use sixteen bits of precision. (2)

188 x 64

0000 0000 1011 1100

x0000 0000 0100 0000

= 0010 1111 0000 0000

- Simply left shifted the multiplicand by 6 and filled with zeros.

1. Assume that the integers are unsigned. You must use hexadecimal arithmetic. Determine the result in hex, octal, and binary. (2)

6A 94 +5AB1

Hex = C545

Binary of C545 = 1100 0101 0100 0101

Oct = 1 100 010 101 000 101

= 142505

1. Show how to divide 371 by 8 using unsigned binary division. Show the remainder, if there is one. (2)

1000/ 101110011

1000 | 101110011

+1000

00111

+0000

1110

+ 1000

01100

+1000

01001

+1000

Remainder = 11(3)

Answer: xxx101110 R.11

1. Show how to perform the operation (6 \* 9) using the optimized multiplier discussed in class. Assume the operands are 5b instead of 32b (be sure that you adjust all of the registers appropriately). You must show each step, from initialization to the final product. (8)

0110 \* 1001

|  |  |
| --- | --- |
| Product | Mult |
| 00000 01001  00110 01001  00011 00100  00001 1010  00000 10010  00110 11001  00011 01100  00001 10110 | 00110  00110  00110  00110  00110  00110  00110  00110 |

1. Show how to perform the operation (-6 \* 9) using Booth’s algorithm. Assume the operands are 5b instead of 32b (be sure that you adjust all of the registers appropriately). You must show each step, from initialization to the final product. (8)

Product Multicplicand

|  |  |
| --- | --- |
| 00000 01001  00110 01001  00011 00100  11101 00100  11110 10010  11111 01001  11011 0100  11101 10100  10111 10100  11011 11010 | 11010 |