***King Fahd University of Petroleum and Minerals***

***College of Computer Science and Engineering***

***Computer Engineering Department***

**COE 202: Digital Logic Design (3-0-3)**

**Term 112 (Spring 2012)**

**Major Exam II**

**Thursday April 12, 2012**

**Time: 150 minutes, Total Pages: 13**

**Name:\_\_\_KEY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_\_**

**Notes:**

* Do not open the exam book until instructed
* **Calculators are not allowed** (*basic, advanced, cell phones, etc*.)
* Answer all questions
* All steps must be shown
* Any assumptions made must be clearly stated

|  |  |  |
| --- | --- | --- |
| **Question** | **Maximum Points** | **Your Points** |
| **1** | **23** |  |
| **2** | **22** |  |
| **3** | **10** |  |
| **4** | **20** |  |
| **5** | **15** |  |
| **6** | **15** |  |
| **Total** | **105** |  |

**Question 1. (23 points)**

1. For the following Boolean function shown in the K-map:

# F(A, B, C, D)=m(0, 1, 2, 5, 6, 7, 10, 12, 13, 14, 15)

AB

CD

00

01

11

10

00

01

11

10

1

1

0

1

0

1

1

1

1

1

1

1

0

0

0

1

# Identify all the *prime implicants* and the *essential prime implicants* of F.

Prime Implicants:

Essential Prime Implicants:

# Simplify the Boolean function **F** into a minimal sum-of-products expression.

1. Consider the following Boolean function **F** together with the don`t care conditions **d** shown in the k-map:

## F(A, B, C, D)=m(0, 10, 15), d(A, B, C, D)=m(1, 2, 4, 8, 11, 14)

CD

AB

10

11

01

00

1

X

0

X

00

X

0

0

0

01

0

0

1

X

11

X

0

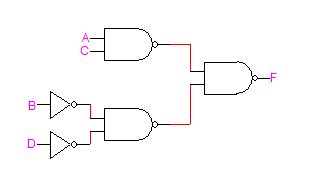
X

1

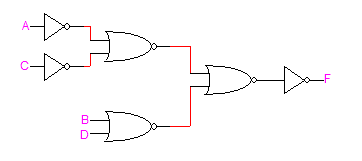
10

# Simplify the Boolean function **F** together with the don`t care conditions **d,** into minimal sum-of-products expression.

# Starting with the sum-of-products expression, implement the function using only **NAND** gates and **Inverters**.



# Starting with the sum-of-products expression, implement the function using only **NOR** gates and **Inverters**.



**Question 2. (22 Points)**

Design a circuit that accepts two 2-bit unsigned numbers A = A1A0 and B = B1B0. The circuit produces A – B when A > B, and produces A + B otherwise. Find the following:

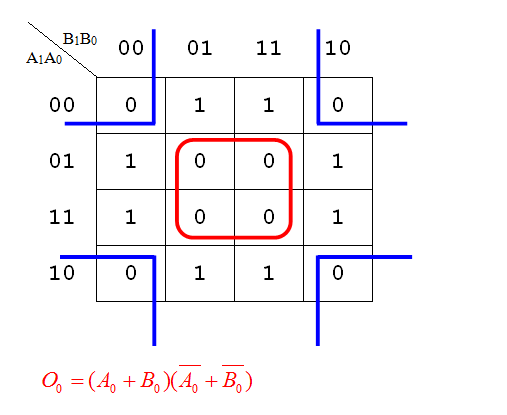
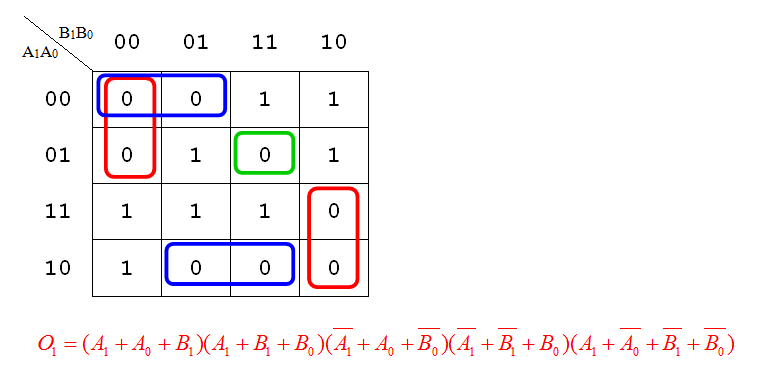
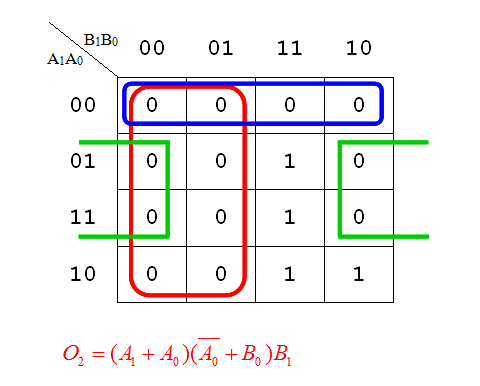
1. The number of outputs produced by the circuit.

*A* – *B* result is at most 2 bits, *A* + *B* result is at most 3 bits ⇒ **# outputs = 3**

1. The truth table of the circuit.

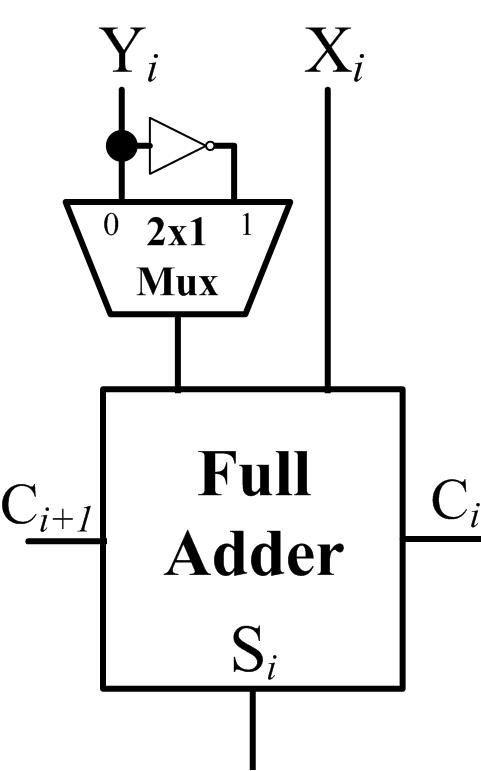
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *A*1 | *A*0 | *B*1 | *B*0 | *O*2 | *O*1 | *O*0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

1. The minimal product-of-sums expression for each output.

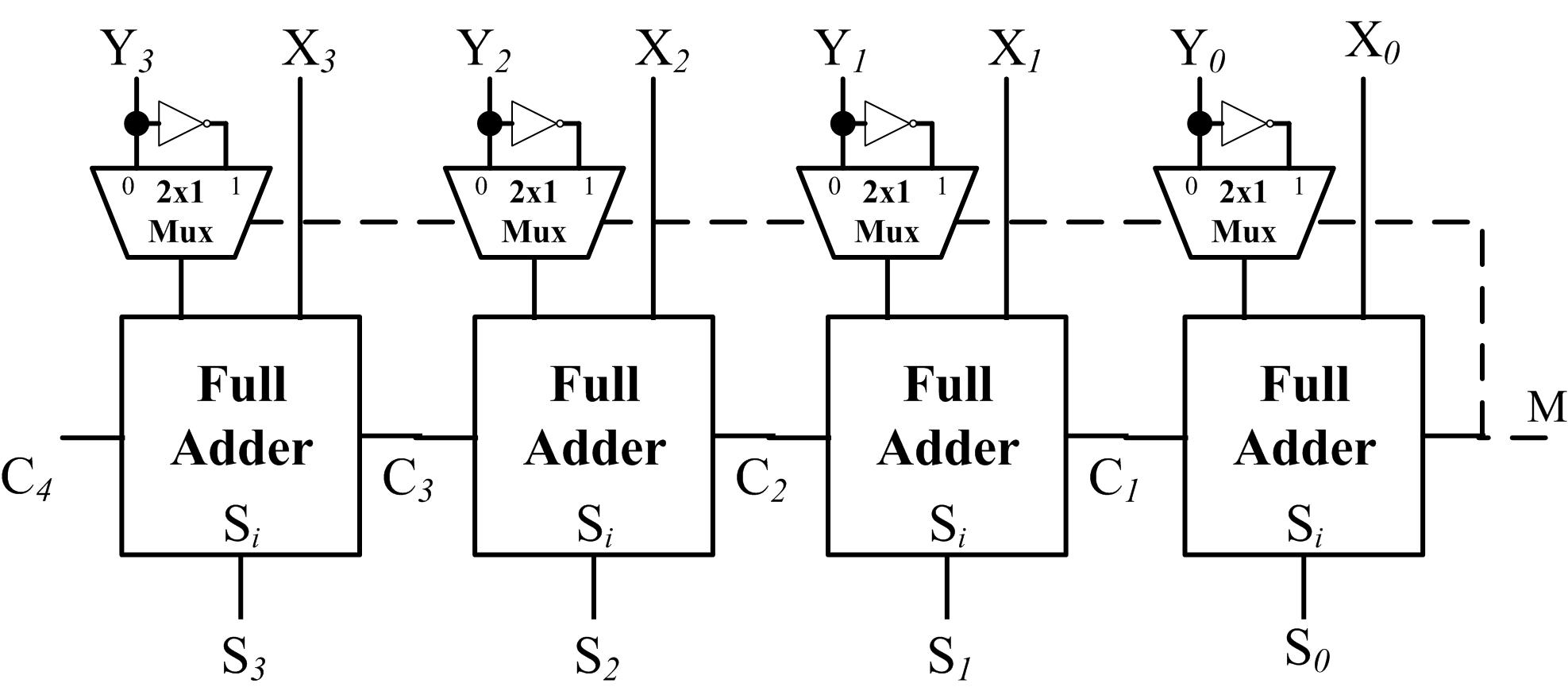
 (TT =4, TT 🡪 K-Map=4, Expressions=8)

**Question 3. (10 Points)**

1. Use the shown circuit on the right to build a 4-bit **adder-subtractor** which can add or subtract two 4-bit numbers X and Y. A mode control input signal **M** is used to define the operation to be performed; if **M=0**, output is (X+Y) while if **M=1**, output is (X-Y).



*CLEARLY label ALL inputs and outputs*



1. The Full-Adder circuit is shown to the right. Given the following gate delays;



|  |  |
| --- | --- |
| **Gate/**  **Circuit** | **Propagation**  **Delay** |
| Inverter | **1 ** |
| AND, OR | **2 ** |
| XOR | **4 ** |
| 2x1 **Mux** | **5 ** |

1. What is the carry propagation delay per Full adder stage?

T(CP)= TAND+TOR = 4****

1. For an ***n*-bit**  Ripple-Carry **Adder-Subtractor** using the circuit of part (a), what is the total delay for the nth sum bit and the (n+1)th carry-out bit?

(*Clearly identify each delay component*)

Delay of the nth Sum output = (1****5**n** = 4n****

Delay of the (n+1)th Carry output = (1****5**n** = 4n****

**Question 4. (20 Points)**

1. If **6-bit registers** are used, show the binary number representation of the decimal numbers (+23), (–23), (+11), and (–11) using the following representation systems:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **+23** | **–23** | **+11** | **–11** |
| Signed magnitude system | **010111** | **110111** | **001011** | **101011** |
| Signed 1’s complement system | **010111** | **101000** | **001011** | **110100** |
| Signed 2’s complement system | **010111** | **101001** | **001011** | **110101** |

1. Provide the decimal equivalent of each of the following **signed 2’s complement** numbers:

|  |  |
| --- | --- |
| **Signed 2’s Complement Number** | **Equivalent Decimal Number** |
| 001101 | **+13** |
| 010011 | **+19** |
| 101101 | **-19** |
| 110011 | **-13** |

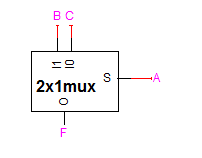
1. If **6-bit registers** are used, perform the following **signed 2’s complement** arithmetic operations on the provided signed 2’s complement numbers. For each case, state whether the result is correct or an **overflow** has occurred.

|  |  |
| --- | --- |
| Signed 2’s Complement Arithmetic Operation | Correct Answer or Overflow? |
| 1. **001101 – 101101**   **001101 ⇒ 001101**  **– 101101 ⇒ + 010011**  **⇒ 100000** | ***Overflow*** |
| 1. **010011 – 001101**   **010011 ⇒ 010011**  **– 001101 ⇒ + 110011**  **⇒ 000110** | ***Correct*** |
| 1. **101101 + 110011**   **101101**  **+ 110011**  **100000** | ***Correct*** |

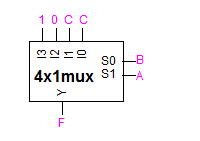
**Question 5. (15 Points)**

Given the function

1. Implement F using a single 2-to-1 MUX with no additional gates. Properly label all input and output lines.

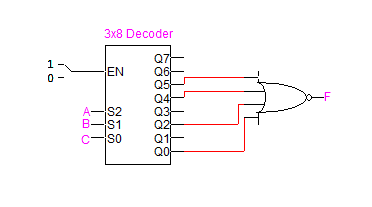


1. Implement F using a 4-to-1 MUX. Properly label all input and output lines.

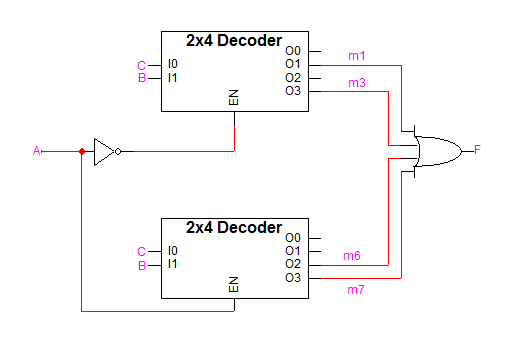


1. Implement F using a single 3-to-8 decoder, and a single NOR gate. Properly label all input and output lines.

F=∑m(1,3,6,7)=ΠM(0,2,4,5)



1. Implement F using two 2-to-4 decoders with enable, one inverter, and one OR gate. Properly label all input and output lines.

**Question 6. (15 points)**

1. Given two 4-bit signed 1’s complement numbers **A** and **B**; for A = 1010 and B = 1101;
2. What are the corresponding decimal values of A and B?

**A=1010 = ( -5 )Decimal B=1101 = ( -2 )Decimal**

1. If these values of A and B are applied to the shown magnitude comparator circuit, what are the values of the resulting outputs?

X

Y

X>Y

X<Y

X=Y

**4-bit**

**Magnitude Comparator**

4

4

**A**

**B**

**(X > Y) = \_\_0\_\_\_\_\_**

**(X = Y) = \_\_0\_\_\_\_\_**

**(X < Y) = \_\_1\_\_\_\_\_**

1. Given two 4-bit signed 1’s complement numbers **A** and **B**, design the following circuits using any number of the following components: XOR gates, decoders, encoders, multiplexers, adders, and/or magnitude comparators:
2. A circuit whose 4-bit output **Z** equals the larger of either **A** or **B** given that both **A** and **B** are positive values.
3. A circuit whose 4-bit output **Z** equals the larger of either **A** or **B** given that both **A** and **B** are negative values (*Hint: use conclusions of part* (a)).
4. A circuit whose 4-bit output **Z** equals the larger value of either **A** or **B** given that **A** and **B** may be +ive or –ive in any possible combination.

(*You* ***must*** *clearly label the MSI parts used together with all inputs and outputs*)

