



```
1 module Prob_5_29 [output reg y_out, input x_in, clock, reset_b];  
2 parameter a0 = 3'b000, a1 = 3'b001, a2 = 3'b010, a3 = 3'b011, a4 = 3'b100;  
3 reg [2:0] state, next_state;  
4 always @ (posedge clock, negedge reset_b)  
5  
6 if (reset_b == 1) state <= a0;  
7 else state <= next_state;  
8 always @ (state, x_in) begin  
9 y_out = 0;  
10 next_state = a0;  
11 case (state)  
12 a0: if (x_in) begin next_state = a4; y_out = 1; end else begin next_state = a3; y_out = 0; end  
13 a1: if (x_in) begin next_state = a4; y_out = 1; end else begin next_state = a1; y_out = 0; end  
14 a2: if (x_in) begin next_state = a0; y_out = 1; end else begin next_state = a2; y_out = 0; end  
15 a3: if (x_in) begin next_state = a0; y_out = 1; end else begin next_state = a1; y_out = 0; end  
16 a4: if (x_in) begin next_state = a3; y_out = 0; end else begin next_state = a2; y_out = 0; end  
17 default: next_state = 3'bxxx;  
18 endcase  
19 end  
20 endmodule
```

ECE119 Ψηφιακή Σχεδίαση

Εργαστηριακές ασκήσεις, Multisim - Verilog

Lab 5: Binary Conversion and Adders

Required Tools and Technology	1
Lab 5: Binary Conversion and Adders.....	2
Learning Objectives.....	2
Expected Deliverables.....	2
5.1 Theory and Background	3
5.2 Simulate: Building a Half-Adder Circuit.....	6
5.3 Simulate: Building a Full-Adder Circuit	8
5.4 Conclusion.....	10
5.5 Exercise: HDL – Verilog, Half adder.....	12
5.6 Exercise: HDL – Verilog, Full adder.....	12
5.7 Exercise: HDL – Verilog, Ripple Carry 4-bit, Add-Sub.....	13
5.8 Exercise: HDL - Verilog - Test Circuit	14

Required Tools and Technology

Software: NI Multisim 14.0 or newer

- ✓ **Install Multisim:**
http://www.ni.com/gate/gb/GB_ACADEMICEVALMULTISIM/US
- ✓ **View Help:**
<http://www.ni.com/multisim/technical-resources/>

MultisimLive

- ✓ <https://www.multisim.com/>

Lab 5: Binary Conversion and Adders

In the first lab, you explored truth tables with two inputs and learned how to design their corresponding circuits. Two binary inputs, namely, 1 and 0 are the simplest of circuits. More complex circuits have more combinations of binary numbers. This makes it impractical to create truth tables for all of the possible combinations and permutations. Instead of creating truth tables, you will look to a system that converts your binary numbers to a *binary-coded decimal (BCD)*. Depending on the sequence and each number's position in it, they are assigned a value. Adding up these values gives us the BCD.

Learning Objectives

In this lab, students will:

1. Construct half and full adders with logic gates and create truth tables from them
2. Confirm the truth table for a full adder.

Expected Deliverables

In this lab, you will collect the following deliverables:

- Long answer questions regarding adders
- 3 Truth Tables
- Conclusion questions

5.1 Theory and Background

Binary-Coded Decimal

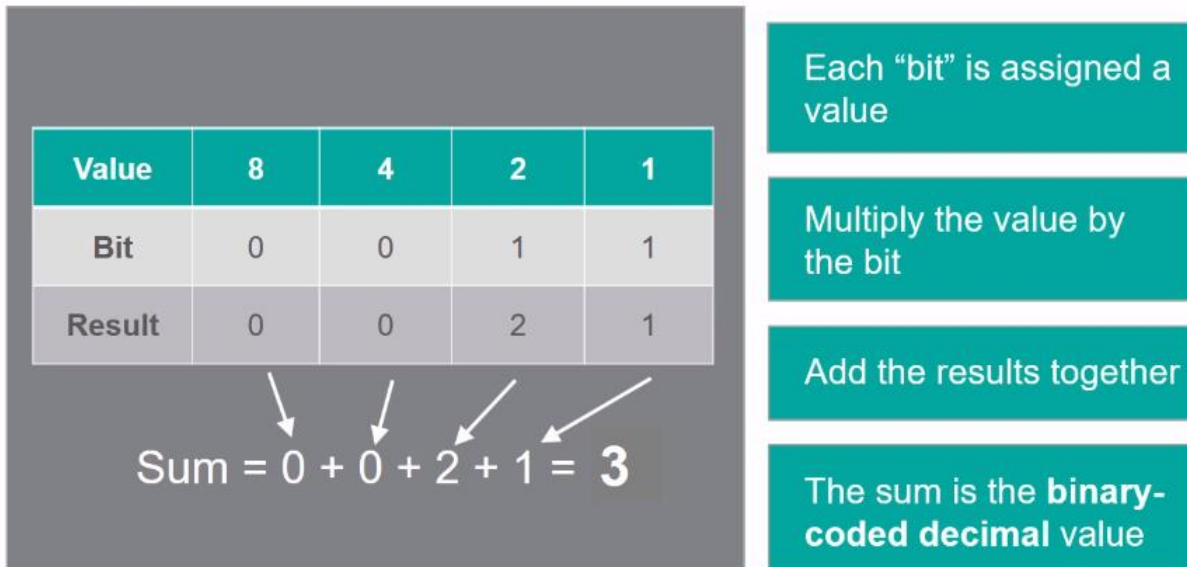


Figure 5-1 Video. View the video here: <https://youtu.be/YYGAPRracIY>



Video Summary

- In larger circuits it is not practical to make truth tables so binary is converted into a binary-coded decimal (BDC)
- Half adders can be represented in Multisim by a single component or by creating them using an AND and an XOR gate
- Half adders have two inputs and two outputs
- Full adders can be represented in Multisim by a single component or two AND gates, two XOR gates and one OR gate.
- Full adders have three inputs and two outputs

Binary-Coded Decimals

Let's look at the following example of a 4-bit binary number (four binary numbers code to one decimal number):

0101

The number furthest to the right is given the value of 1 (similar to the ones column in regular addition). The position immediately to its left is given the value of 2 (similar to a

tens column). The value to the left of this is 4 (similar to the hundreds column). The pattern you will notice is that the values assigned to the number on the left increase by twice the value of the number before it.

In our example, from left to right, the BCD value for the four-bit binary code above, adding up the values is:

$$0 + 4 + 0 + 1 = 5$$

Conversions that result in a single digit number (0-9) are already in BCD format. In order to output the decimal value of a binary number greater than 9 in circuitry, a Binary-to-BCD converter such as a Shift-Add-3 algorithm must be used.

Adders

Half-Adders:

- A half-adder does binary addition on two inputs (A and B).
- The two outputs are labeled sum (S) and carry (C).
- Half adders can be built with:
 - an XOR gate and an AND gate (shown on the left).
 - a component in Multisim (shown on the right).

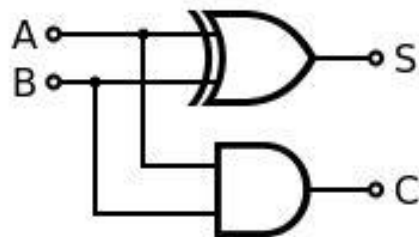


Figure 5-2 Half Adder from Gates

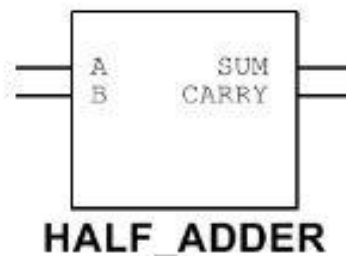


Figure 5-3 Half Adder Multisim

Full-Adders:

- A full-adder does binary addition on three inputs: A, B, and C_{in} .
- Full adders usually work in a cascade fashion where they are used to add binary numbers with an increasing number of bits.
- The two outputs are sum (S) and carry (C_{out}).
- You will notice that full adders can also use logic gates or a component.

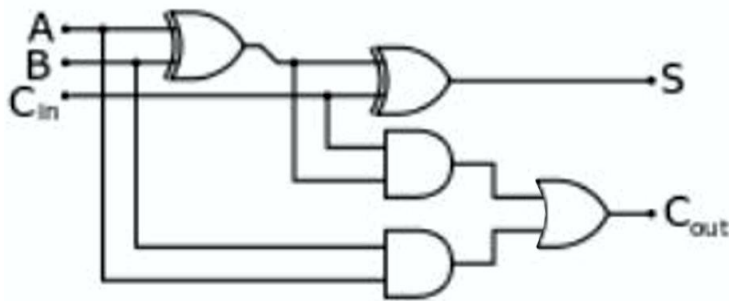


Figure 5-4 Full Adder from Gates

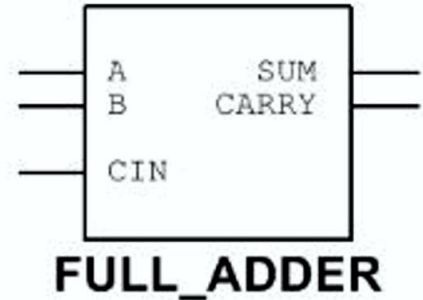


Figure 5-5 Full Adder Multisim

5-1 What are the similarities and differences between half adders and full adders?
Select the right answers (multiple choices)

- Τόσο ο half adder όσο και ο full adder είναι συνδυαστικά κυκλώματα. Και οι δύο προσθέτουν δύο ψηφία ενός bit.
- Τόσο ο half adder όσο και ο full adder είναι συνδυαστικά κυκλώματα. Ωστόσο, ο πρώτος προσθέτει δύο ψηφία ενός bit, ενώ ο δεύτερος προσθέτει τρία ψηφία.
- Ο half adder έχει δύο τιμές εισόδου a και b που αντιπροσωπεύουν τα bits δεδομένων. Ο full adder έχει μια επιπλέον είσοδο, το κρατούμενο από την κατώτερη βαθμίδα (C_{in}).
- Τόσο ο half adder όσο και ο full adder έχουν δύο τιμές εισόδου a και b που αντιπροσωπεύουν τα bits δεδομένων.
- Ο half adder έχει δύο εξόδους. Η μία είναι το άθροισμα της εισόδου (S) και η άλλη είναι το κρατούμενο του αθροίσματος (C). Ο full adder έχει επίσης δύο εξόδους (S; C_{out}).
- Ο half adder έχει δύο εξόδους. Η μία είναι το άθροισμα της εισόδου (S) και η άλλη είναι το κρατούμενο του αθροίσματος (C). Ο full adder έχει τρεις εξόδους (S; C_1 , C_2).
- Στην περίπτωση του half adder, το κρατούμενο από την κατώτερη βαθμίδα προστίθεται. Στην περίπτωση του full adder, το προηγούμενο κρατούμενο δεν αθροίζεται.
- Στην περίπτωση του half adder, το κρατούμενο από την κατώτερη βαθμίδα δεν προστίθεται. Στην περίπτωση του full adder, το προηγούμενο κρατούμενο μεταφέρεται και αθροίζεται.
- Ο half adder αποτελείται από μία πύλη XOR και μία πύλη AND. Ο full adder αποτελείται πρακτικά από 2 half adders συνδυασμένους. Αποτελείται από δύο πύλες XOR, δύο πύλες AND και μία πύλη OR.

5.2 Simulate: Building a Half-Adder Circuit

Half-Adder Circuits

Half-adder circuits can be built using a combination of logic gates.

- Launch Multisim.
- Open a new circuit.
 - Select **File>>New**.
 - In the menu that appears, select **Blank** and click **Create**.
- Connect the following circuit:
 - Place an **XOR** gate and an **AND** gate from the **Misc Digital** group.
 - Place two **INTERACTIVE_DIGITAL_CONSTANTS** from the **Sources** group.
 - Place two **PROBE_DIG_REDS** from the **Indicators** group.
 - Wire them as shown:

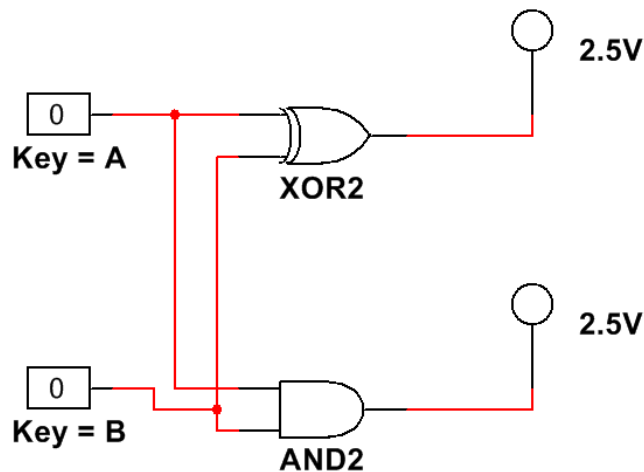


Figure 5-6 A Half Adder Circuit



- | | |
|--|--|
| - Multisim: | Όνομα αρχείου " 5_half_adder.ms14 ".
Προσθήκη στο zip file με όνομα "Lab05_ονοματεπώνυμο_AM.zip" |
| ή | |
| - MultisimLive:
Schematic image | Όνομα αρχείου " 5_half_adder.png ".
Προσθήκη στο zip file με όνομα "Lab05_ονοματεπώνυμο_AM.zip" |

- Click the **Run** button to begin simulating the circuit.



Figure 5-7 Run Button

- Using the **A** and **B** keys, vary the inputs into the circuit.

5-2 Fill out the truth table below.

A	B	XOR (SUM)	AND (CARRY)
0	0		
0	1		
1	0		
1	1		

- Stop the simulation by clicking the **Stop** button.



Figure 5-8 Stop Button

Can you see how the XOR and AND gates represent the sum and carry of the numbers A and B added together?

- $0 + 0 = 0$
- $1 + 0$ or $0 + 1 = 1$, with no carry.
- $1 + 1 = 2$, but 2 is not a binary number. In binary, 2 is represented as 10. The 1 is the carry and the 0 is the sum.

5.3 Simulate: Building a Full-Adder Circuit

Full-Adder Circuits

Connect the following circuit:

- Place two **XOR** gates, two **AND** gates and one **OR** gate from the **Misc Digital** group.
- Wire them as shown:

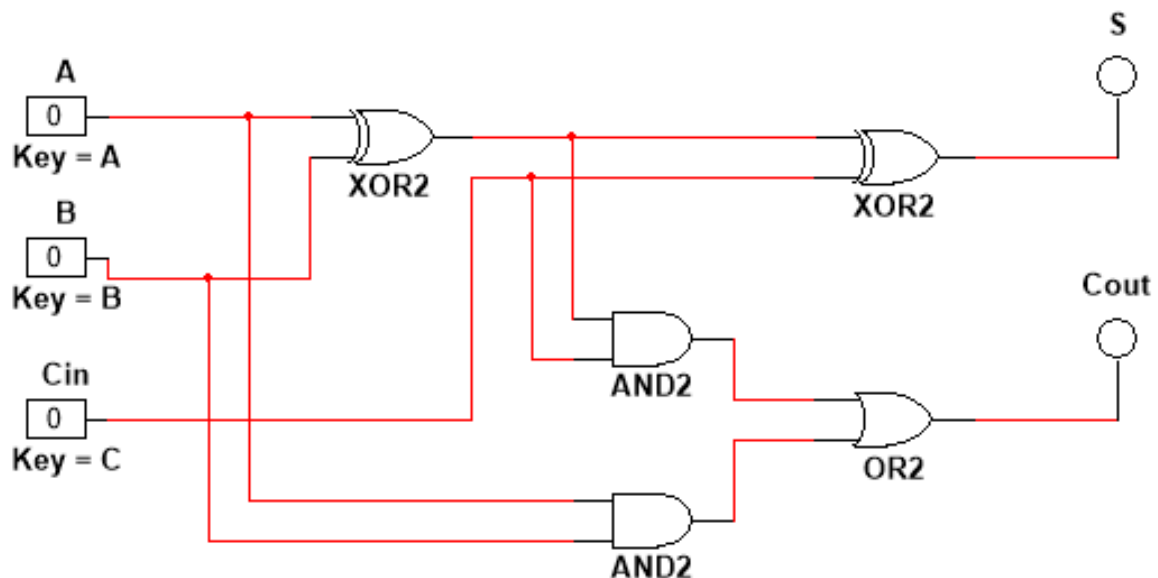


Figure 5-9 A Full Adder Circuit



- **Multisim:** Όνομα αρχείου "**5_full_adder.ms14**".
Προσθήκη στο zip file με όνομα "Lab05_ονοματεπώνυμο_AM.zip"
- ή
- **MultisimLive:** Όνομα αρχείου "**5_full_adder.png**".
Schematic image Προσθήκη στο zip file με όνομα "Lab05_ονοματεπώνυμο_AM.zip"

- Click the **Run** button to begin simulating the circuit.
- Using the **A**, **B** and **C** keys, vary the inputs into the circuit.

5-3 Fill out the truth table below.

A	B	C _{in}	SUM	C _{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Again, can you see how the truth table represents the sum and carry of the numbers A, B, and C_{IN} added together?

- $0 + 0 + 0 = 0$
- $0 + 0 + 1 = 1$, with no carry.
- $0 + 1 + 1 = 2$, but 2 is not a binary number. In binary, 2 is represented as 10. The 1 is the carry and the 0 is the sum.
- $1 + 1 + 1 = 3$. In binary, 3 is represented as 11. The first 1 is the carry and the second 1 is the sum.

5.4 Conclusion

5-4 Under what conditions would you use a half adder? A full adder?

Select the right answers (multiple choices)

- a) Half adder: Μπορεί να προσθέσει δύο αριθμούς του ενός bit, λαμβάνοντας υπόψη την ύπαρξη προηγούμενου κρατουμένου
- b) Half adder: Μπορεί να προσθέσει δύο αριθμούς του ενός bit, χωρίς όμως να λαμβάνει υπόψη την ύπαρξη προηγούμενου κρατουμένου.
- c) Full adder: Μπορεί να προσθέσει δύο αριθμούς του ενός bit, λαμβάνοντας υπόψη την ύπαρξη προηγούμενου κρατουμένου.
- d) Full adder: Μπορεί να προσθέσει δύο αριθμούς του ενός bit, χωρίς όμως να λαμβάνει υπόψη την ύπαρξη προηγούμενου κρατουμένου.
- e) Half adder: Προσθέτει δύο αριθμούς του ενός bit
- f) Half adder: Προσθέτει τρεις αριθμούς του ενός bit
- g) Full adder: Προσθέτει δύο αριθμούς του ενός bit
- h) Full adder: Προσθέτει τρεις αριθμούς του ενός bit

5-5 When converting binary numbers to BCD:

- A. Numbers are given specific values depending on their position in the sequence
- B. The value of the number on the left increases by twice the value of the number on the right
- C. Single digit numbers are already in BCD format
- D. All of the above

5-6 Half-adders do binary addition on two inputs using two logic gates. What is the correct combination of these logic gates?

- A. XOR and OR
- B. XOR and AND
- C. XNOR and AND
- D. NOT and OR

5-7 The outputs of a half adder are:

- A. Carry and C_{in}
- B. Sum and Carry
- C. Sum and C_{in}

D. Carry and C_{out}

5-8 Which is a feature of full-adders?

- A. They have two inputs
- B. They have three outputs
- C. They work in a cascading fashion
- D. All of the above

5-9 When connecting full-adders together in Multisim, the Carry of one is connect to _____ input of the next.

- A. A
- B. B
- C. C_{in}
- D. None of the above

5.5 Exercise: HDL – Verilog, Half adder

Ο Ημιαθροιστής (Half Adder – H/A) είναι ένα λογικό κύκλωμα το οποίο στην είσοδό του δέχεται 2 δυαδικά ψηφία A και B και στην έξοδό του δίνει το άθροισμα τους Z και το κρατούμενο K της πρόσθεσής τους.

Γράψτε την περιγραφή HDL σε επίπεδο πυλών του παραπάνω κυκλώματος.

Ονομάστε το module: “half_adder”.



- Verilog:

Όνομα αρχείου “**half_adder.v**”.

Προσθήκη στο zip file με όνομα “Lab05_ονοματεπώνυμο_AM.zip”

5.6 Exercise: HDL – Verilog, Full adder

Κατόπιν, σκεφτείτε πως μπορείτε να υλοποιήσετε έναν Πλήρη Αθροιστή (Full Adder – Π/Α) ο οποίος σχηματίζει το αριθμητικό άθροισμα 3 bits εισόδου. Δηλαδή, έχει 3 εισόδους και 2 εξόδους. Δύο από τα bits εισόδου είναι τα 2 bits τα οποία προσθέτουμε. Η τρίτη είσοδος αναπαριστά το κρατούμενο της αμέσως προηγούμενης λιγότερο σημαντικής θέσης. Συμβολίστε με S το άθροισμα και C το κρατούμενο.

Έχετε ήδη γράψει τον πίνακα αληθείας και σχεδιάσατε την τελική του μορφή χρησιμοποιώντας 2 H/A και μια πύλη OR στο Multisim.

Γράψτε τώρα την περιγραφή HDL σε επίπεδο πυλών του παραπάνω κυκλώματος.



Θα πρέπει να χρησιμοποιήσετε 2 στιγμιότυπα από τον ημιαθροιστή της άσκησης 5.5

Θα πρέπει να δώσετε την οδηγία (directive): ``include "half_adder.v"`



Θα πρέπει να την γράψετε ακριβώς πριν από το module (έξω από αυτό).

Με αυτόν τον τρόπο συμπεριλαμβάνετε το περιεχόμενο ενός άλλου αρχείου, στο σημείο όπου βρίσκετε η οδηγία ``include`.

Ονομάστε το module: “full_adder”.



- Verilog:

Όνομα αρχείου “**full_adder.v**”.

Προσθήκη στο zip file με όνομα “Lab05_ονοματεπώνυμο_AM.zip”

5.7 Exercise: HDL – Verilog, Ripple Carry 4-bit, Add-Sub

Γράψτε την περιγραφή HDL σε επίπεδο πυλών ενός 4-bit αθροιστή-αφαιρέτη που δέχεται ως εισόδους μη προσημασμένους αριθμούς. Το κύκλωμα πρέπει να είναι παρόμοιο με αυτό του σχήματος 4.13 (βιβλίο Morris Mano), αλλά χωρίς την έξοδο V.



Θα πρέπει να χρησιμοποιήσετε 4 στιγμιότυπα του πλήρους αθροιστή της άσκησης 5.6

Θα πρέπει να δώσετε την οδηγία (directive): ``include "full_adder.v"`



Θα πρέπει να την γράψετε ακριβώς πριν από το module (έξω από αυτό).

Με αυτόν τον τρόπο συμπεριλαμβάνετε το περιεχόμενο ενός άλλου αρχείου, στο σημείο όπου βρίσκετε η οδηγία ``include`.

Ονομάστε το module: `"add_sub_4_bit"`.



- Verilog:

Όνομα αρχείου `"add_sub_4_bit.v"`.

Προσθήκη στο zip file με όνομα "Lab05_ονοματεπώνυμο_AM.zip"

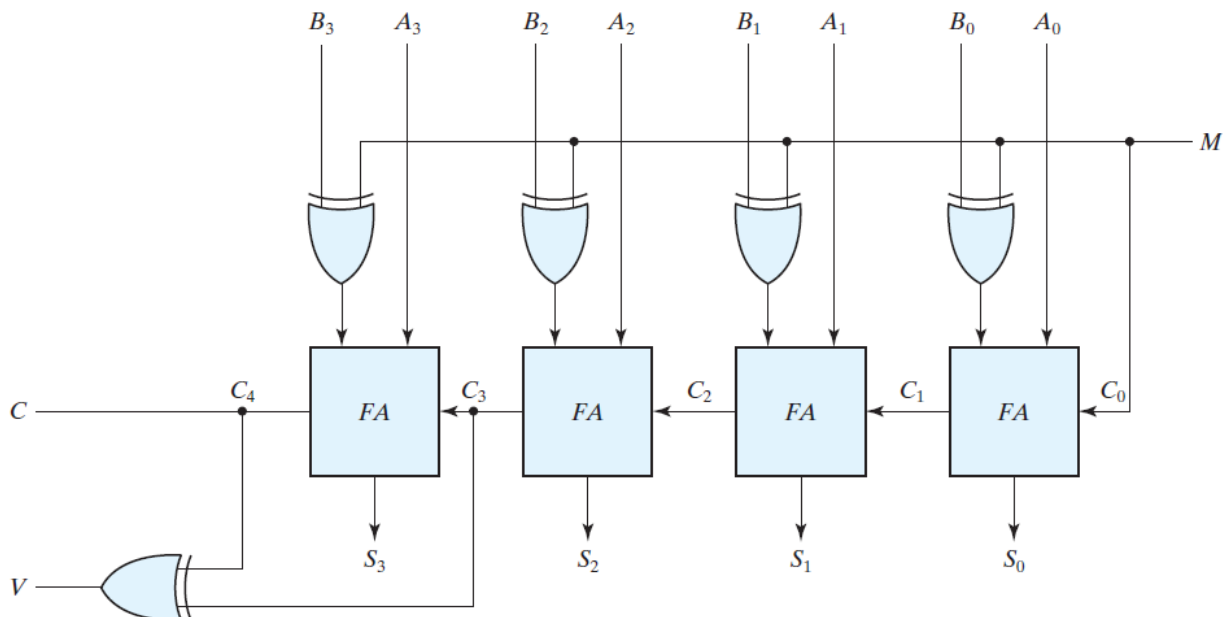


FIGURE 4.13

Four-bit adder-subtractor (with overflow detection)

5.8 Exercise: HDL - Verilog - Test Circuit

Δημιουργήστε μια υπομονάδα διέγερσης (test bench) για την ανάλυση του κυκλώματος που κατασκευάσατε στην Exercise 5.7 (Module: add_sub_4_bit).

Για να δοκιμάσουμε το κύκλωμα, στην αρχή θα ζητήσουμε να μας κάνει μία πρόσθεση δύο 4-bit αριθμών π.χ. 8+4 και θα ελέγξουμε ότι μας δίνει έξοδο την τιμή 12.

Έπειτα θα ζητήσουμε μία αφαίρεση π.χ. 12-8 και θα ελέγξουμε ότι μας δίνει έξοδο την τιμή 4.

Πιο συγκεκριμένα:

- Στα 10 nsec: $A = 1000, B = 0100, M = 0$ (για την πράξη 8+4)
- Στα 50 nsec: $A = 1100, B = 1000, M = 1$ (για την πράξη 12-8)

Ελέγξτε τις κυματομορφές για την σωστή τιμή εξόδου.



Να χρησιμοποιήσετε την εντολή **\$monitor** για να ελέγξετε τις εισόδους - εξόδους.



Να χρησιμοποιήσετε τις εντολές **\$dumpfile** και **\$dumpvars** ώστε να παραχθεί το αρχείο "vcd" με τις κυματομορφές των σημάτων. Έπειτα να ανοίξετε αυτό το αρχείο με το VSCode για να δείτε τις κυματομορφές και να ελέγξετε οπτικά τον τρόπο που μεταβάλλεται η έξοδος στους συνδυασμούς της εισόδου.



Θα πρέπει να δώσετε την οδηγία (directive): ``include "add_sub_4_bit.v"`

Θα πρέπει να την γράψετε ακριβώς πριν από το module (έξω από αυτό).

Με αυτόν τον τρόπο συμπεριλαμβάνετε το περιεχόμενο ενός άλλου αρχείου, στο σημείο όπου βρίσκετε η οδηγία ``include`.

Ονομάστε το module: "t_add_sub_4_bit".



- Verilog:

Όνομα αρχείου "**t_add_sub_4_bit.v**".

Προσθήκη στο zip file με όνομα "Lab05_ονοματεπώνυμο_AM.zip"