

# Εισαγωγή στη Σχεδίαση Συστημάτων VLSI

## 6<sup>η</sup> εργαστηριακή άσκηση

### Α' Ενότητα : Ολοκληρωμένα περιβάλλοντα σχεδίασης

Σε αυτή την ενότητα θα γίνει η σχεδίαση ενός σύνθετου κυκλώματος με χρήση ολοκληρωμένων εργαλείων που προσφέρουν τη δυνατότητα να εκφραστεί ένα κύκλωμα με διαφορετικές μορφές. Συγκεκριμένα, η ροή σχεδίασης ενός κυκλώματος που θα ακολουθηθεί είναι:

1. Κατασκευή σχηματικού διαγράμματος στο πρόγραμμα DSCH
2. Εξαγωγή κώδικα Verilog, όπως προκύπτει από την μετατροπή σε αρχείο.v του DSCH
3. Εισαγωγή κώδικα Verilog στο MICROWIND για την υλοποίηση του layout.

#### ΠΑΡΑΤΗΡΗΣΗ:

Στο πρόγραμμα DSCH λόγω κάποιου σφάλματος στο πρόγραμμα στην μεταφορά των πολυπλεκτών 2:1 η είσοδος ελέγχου είναι σαν να έχει αναστραφεί, οπότε η σειρά που θα εκτελεστούν οι πράξεις δεν θα είναι αυτή που αντιστοιχίζεται στην εκφώνηση, αλλά θα είναι πρόσθεση, αφαίρεση, AND και XOR

#### Ερώτηση 1:

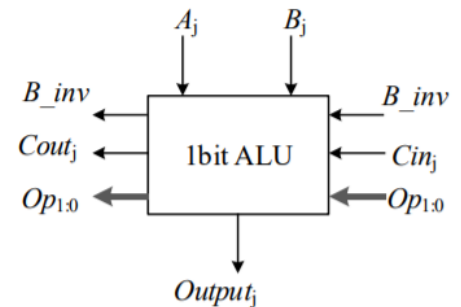
Σε αυτό το ερώτημα θα υλοποιηθεί η αριθμητική και λογική μονάδα των τριών bit που εκτελεί τις πράξεις λογικό XOR, λογικό AND, αφαίρεση και πρόσθεση.

Η μονάδα αυτή επιλέγει κάθε φορά ποια πράξη θα εκτελεστεί με βάση τον 2-bit κωδικό Op[1:0] με βάση την παρακάτω αντιστοίχιση πράξεων-Op :

Operation	Op <sub>0</sub>	Op <sub>1</sub>
XOR	0	0
AND	0	1
Αφαίρεση	1	0
Πρόσθεση	1	1

Αρχικά, θα σχεδιαστεί η μονάδα στο πρόγραμμα DSCH. Για να γίνει αυτό χρειάστηκε η κατασκευή μια μονάδα-κύτταρο που θα χρησιμοποιηθεί για την κατασκευή της 3-bit μονάδας. Η σχηματική απεικόνιση αυτής της μονάδας φαίνεται παρακάτω.

Στο διπλανό σχήμα, όλα τα σήματα είναι 1-bit εκτός από το  $Op[1:0]$ , που είναι δύο bit. Σαν εισόδους δέχεται τα  $A_j$ ,  $B_j$  και παράγει το  $output_j$  και έχει ως εισόδους ελέγχου το κρατούμενο εισόδου  $Cin_j$ , με την αντίστοιχη έξοδο του  $Cout_j$ , και τα  $B\_invertse$  και  $Op$  που βγαίνουν στην έξοδο χωρίς καμία αλλοίωση.

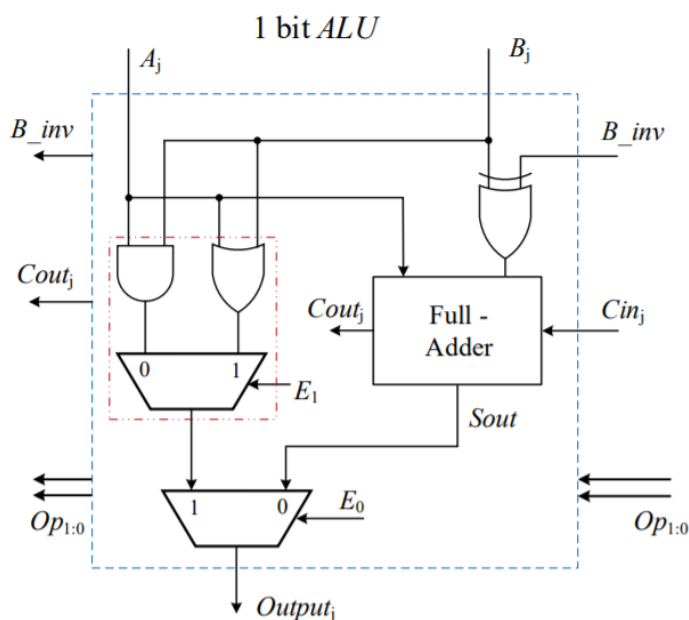


Στο παρακάτω σχήμα, έχουμε μια πύλη AND και μια πύλη XOR, που εισέρχονται σε έναν πολυπλέκτη, ο οποίος αποφασίζει αν θα λειτουργήσει η πράξη AND ή XOR. Η πρόσθεση και η αφαίρεση υλοποιείται από τον Full Adder και μια πύλη XOR, με την οποία επιτρέπεται η ελεγχόμενη αντιστροφή της εισόδου  $B$ .

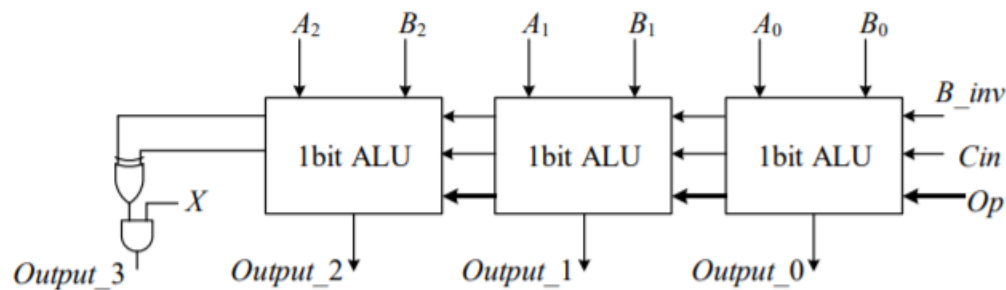
- Όταν έχω  $A+B$ , το  $B$  περνάει αυτούσιο στον F-A.
- Όταν έχω  $A-B$ , το  $B$  αναστρέφεται και μετά εισέρχεται στον F-A.

Στην κυκλωματική υλοποίηση, χρησιμοποιείται και ένας ακόμη πολυπλέκτης, ο οποίος επιλέγει αν θα πάρει τις λογικές πράξεις ή τις αριθμητικές πράξεις, με βάση το  $Op[1:0]$  που αντιστοιχίζει τις πράξεις σε τιμές δυαδικού συστήματος. Συγκεκριμένα,

- Όταν το  $Op$  είναι 1 τότε επιλέγει να ενεργοποιηθεί το σήμα  $E_0$ , δηλαδή θα επιλεχθούν οι λογικές πράξεις.
- Όταν το  $Op$  είναι 0 τότε επιλέγει να ενεργοποιηθεί το σήμα  $E_1$ , δηλαδή θα επιλεχθεί είτε η πράξη AND είτε η πράξη XOR.

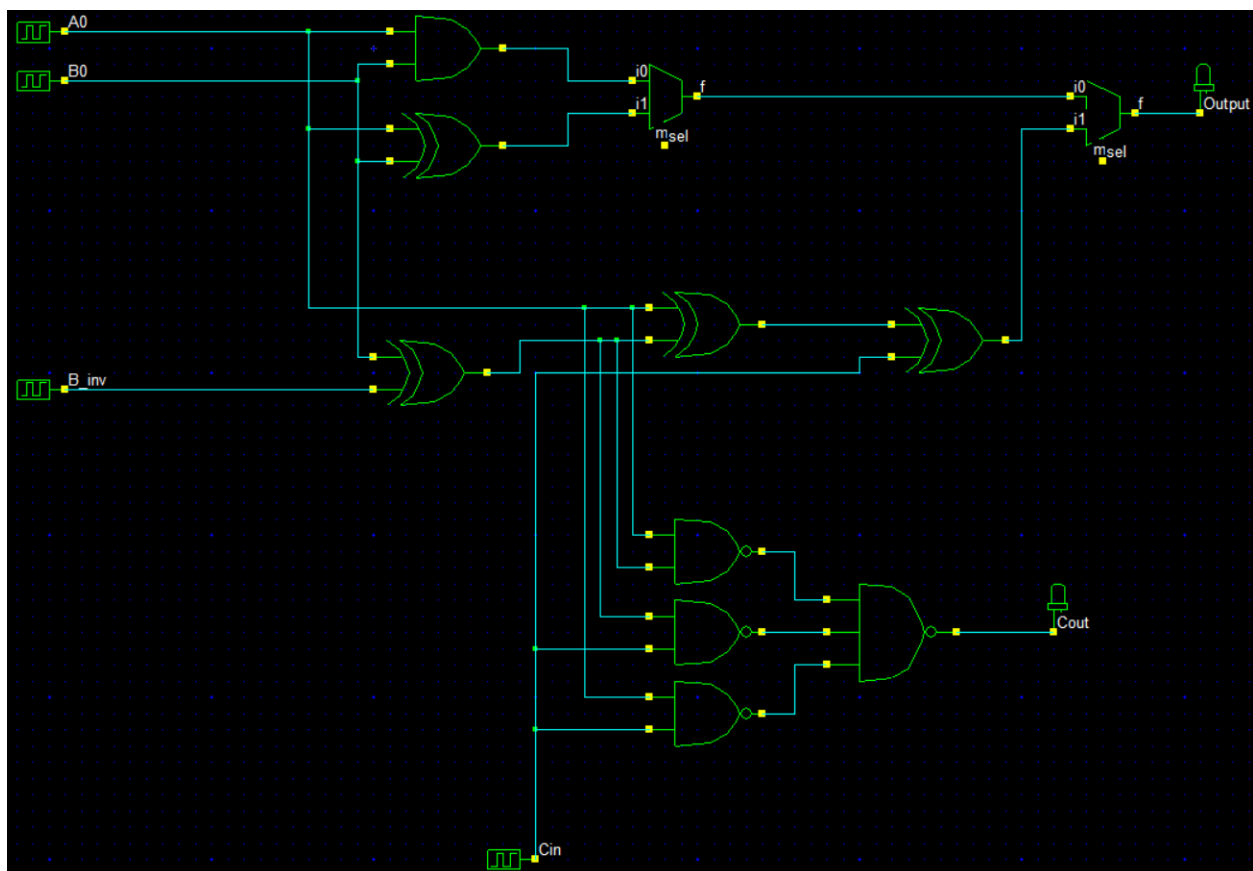


Για την κατασκευή της 3-bit ALU μονάδας, όπως φαίνεται στο παρακάτω κύκλωμα, έχουν χρησιμοποιηθεί 3 μονάδες-κύτταρα 1-bit ALU, με το B\_inverse να εξαρτάται από το Op και οι εξόδοι της μιας μονάδας να αποτελούν είσοδοι της επόμενης.

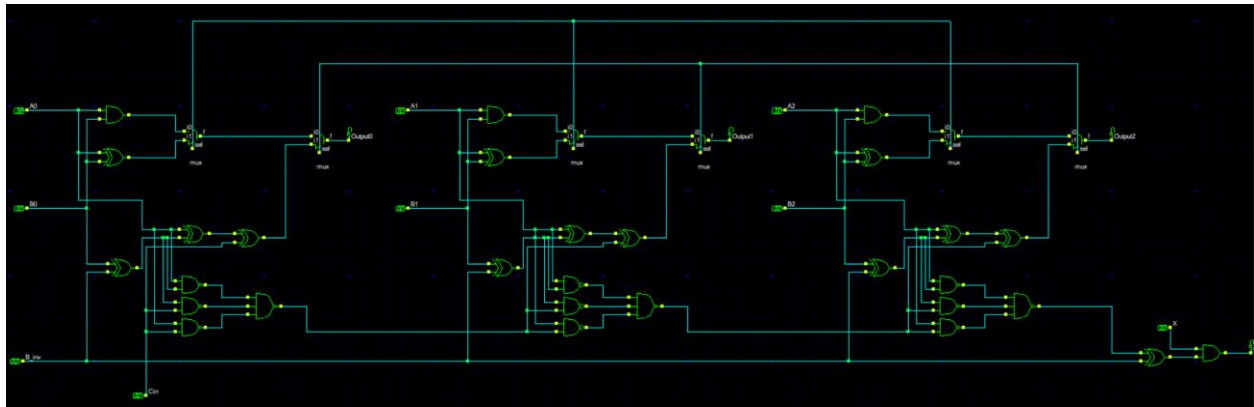


### Κυκλωματική Διάταξη :

Ακολουθώντας την παραπάνω διαδικασία, πρώτα κατασκευάστηκε η 1-bit ALU, όπως φαίνεται στην ακόλουθη κυκλωματική διάταξη στο DSCH.



Αυτή η υλοποίηση της 1-bit ALU, χρησιμοποιήθηκε σαν κύτταρο για την υλοποίηση της 3-bit ALU. Συγκεκριμένα, συνδέθηκαν τρεις 1-bit ALU στη σειρά με τις εξόδους της πρώτης να αποτελούν της εισόδους της δεύτερης κτλ. Με αυτό τον τρόπο προέκυψε η ακόλουθη κυκλωματική διάταξη της 3-bit ALU στο DSCH.



Από αυτή την κυκλωματική διάταξη, βγήκε ο παρακάτω κώδικας Verilog με την εντολή Make Verilog File.

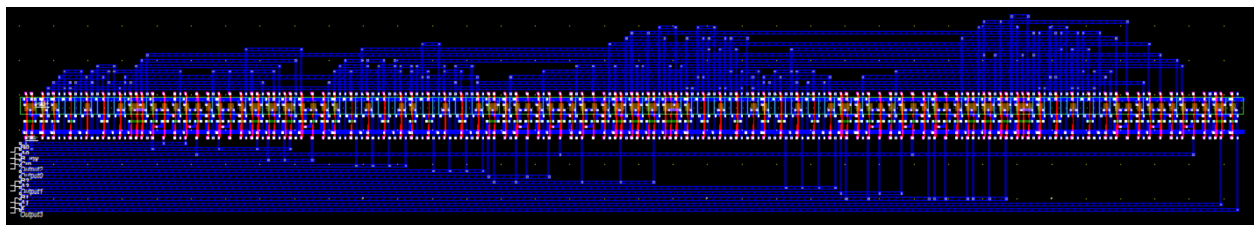
```

Verilog, Hierarchy and Netlist
Verilog | Hierarchy | Netlist | Critical path | Information
// DSCH 3.5
// 20/1/2021 7:28:11 μμ
// D:\NTUA\MAΘHΜATA\7ο Εξάμηνο\VLSI\εργαστηριακές ασκήσ

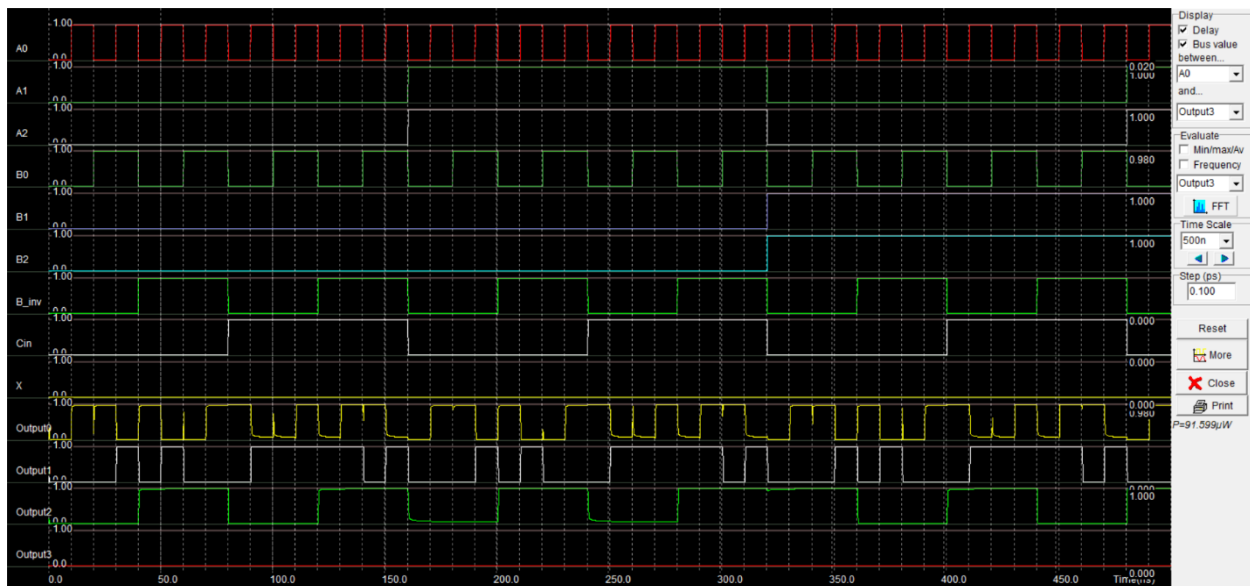
module 3bitALU( A0,B0,B_inv,Cin,A2,B2,A1,B1,
X,Output0,Output1,Output2,Output3);
input A0,B0,B_inv,Cin,A2,B2,A1,B1;
input X;
output Output0,Output1,Output2,Output3;
wire w4,w5,w6,w7,w8,w9,w10,w11;
wire w13,w14,w16,w17,w18,w19,w24,w25;
wire w26,w27,w28,w29,w30,w31,w32,w33;
wire w34,w38,w39,w40,w41,w42,w43,w44;
wire ;
and # (3) and2_1(w4,B0,A0);
mux # (1) mux_2(w7,w4,w5,w6);
nand # (3) nand3_3(w11,w8,w9,w10);
xor # (3) xor2_4(w5,A0,B0);
xor # (4) xor2_5(w13,B0,B_inv);
xor # (3) xor2_6(w14,A0,w13);
nand # (2) nand2_7(w6,w13,A0);
nand # (2) nand2_8(w9,Cin,w13);
nand # (2) nand2_9(w10,Cin,A0);
xor # (3) xor2_10(w16,w14,Cin);
mux # (1) mux_11(Output2,w17,w18,w19);
mux # (1) mux_12(Output0,w7,w16,w19);
and # (3) and2_13(w24,B2,A2);
xor # (3) xor2_14(w25,A2,B2);
xor # (4) xor2_15(w26,B2,B_inv);
xor # (3) xor2_16(w27,A2,w26);
nand # (2) nand2_17(w28,w26,A2);
nand # (2) nand2_18(w30,w29,w26);
nand # (2) nand2_19(w31,w29,A2);
nand # (2) nand3_20(w32,w28,w30,w31);

```

Αυτό το αρχείο κώδικα χρησιμοποιήθηκε στο MICROWIND, όπου και έδωσε την ακόλουθη κυκλωματική διάταξη για την 3-bit ALU μονάδα.



Γραφική Παράσταση:

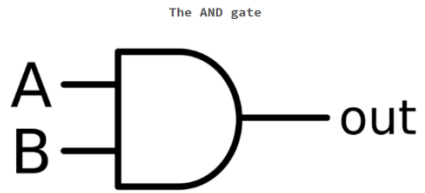


Από την παραπάνω γραφική παράσταση φαίνεται η σωστή λειτουργία της μονάδας αφού για  $Op=[11]$  η μονάδα λειτουργεί σαν πύλη XOR, όπου η έξοδος είναι 1 όταν και οι δύο εισοδοί είναι διαφορετικοί, για  $Op=[10]$  η μονάδα λειτουργεί σαν πύλη AND, όπου η έξοδος είναι 1 μόνο όταν και οι δύο ισούνται με τη μονάδα, για  $Op=[01]$  η μονάδα λειτουργεί σαν πρόσθεση και για  $Op=[00]$  η μονάδα λειτουργεί σαν αφαίρεση.

## B' Ενότητα : Μελέτη Κατανάλωσης Ισχύος σε κυκλώματα CMOS - VLSI

Σε αυτό το μέρος της άσκησης θα εργαστούμε πάνω στην πύλη AND και θα μελετηθούν κάποιες μεταβολές που υφίσταται η ισχύς ανάλογα με διάφορες μεταβολές στις κυματομορφές εισόδου και ανάλογα με διάφορες τιμές χωρητικότητας φορτίου εξόδου.

Η πύλη AND αθροίζει λογικά τις εισόδους της και στην έξοδο βγάζει 1 μόνο όταν και οι δύο κυματομορφές εισόδου είναι 1.



Πίνακας αληθείας AND:

A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1

### **Ερώτηση 1:**

Για την κατασκευή της πύλης AND στο MICROWIND χρησιμοποιήθηκε το ακόλουθο αρχείο Verilog κώδικα.

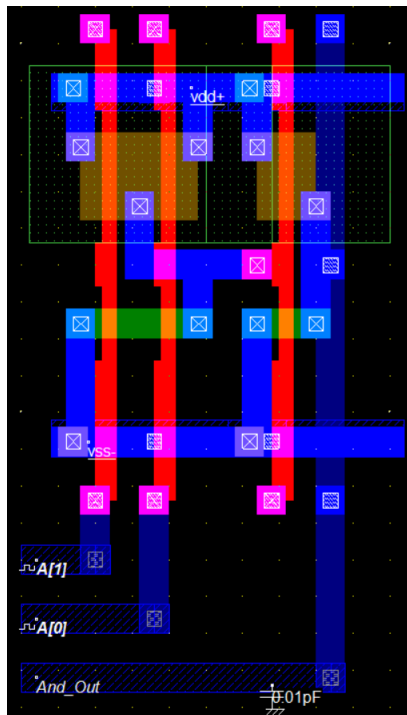
```

module And(A[1],A[0],And_Out);
input A[1],A[0];
output And_Out;
wire;
and #(18) and2_1(And_Out,A[1],A[0]);
endmodule

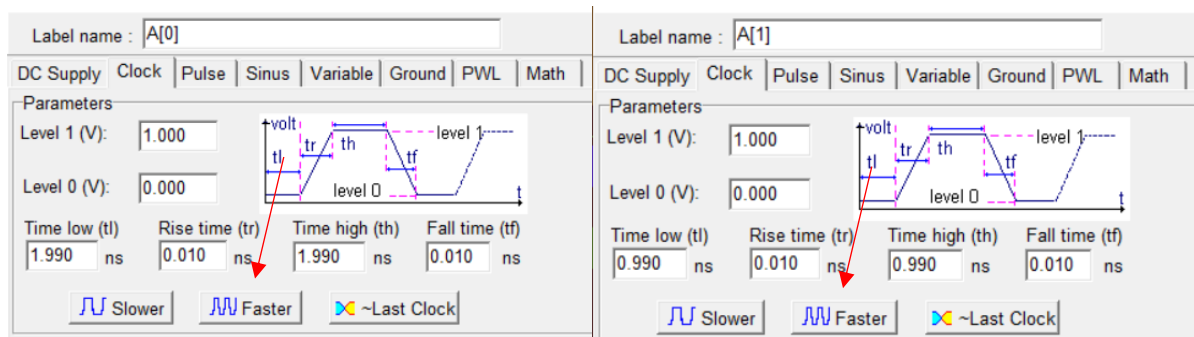
//Simulation parameters in Verilog Format
always
#200 A[1]=~A[1];
#400 A[0]=~A[0]
  
```

Με compile του παραπάνω κώδικα προέκυψε η παρακάτω κυκλωματική διάταξη στην οποία προστέθηκε ένα φορτίο χωρητικότητας  $C = 0,01\text{pF}$ .

Κυκλωματική διάταξη:



Η πρώτη προσομοίωση έτρεξε για την ονομαστική περίοδο εισόδων T ( $A[0]=4\text{nsec}$ ,  $A[1]=2\text{nsec}$ ) για διάρκεια (timescale) ίση με  $20\text{nsec}$ .



Στις επόμενες προσομοιώσεις αυξήθηκε η συχνότητα των παλμοσειρών με χρήση της εντολής faster, που διπλασιάζει τη συχνότητα.

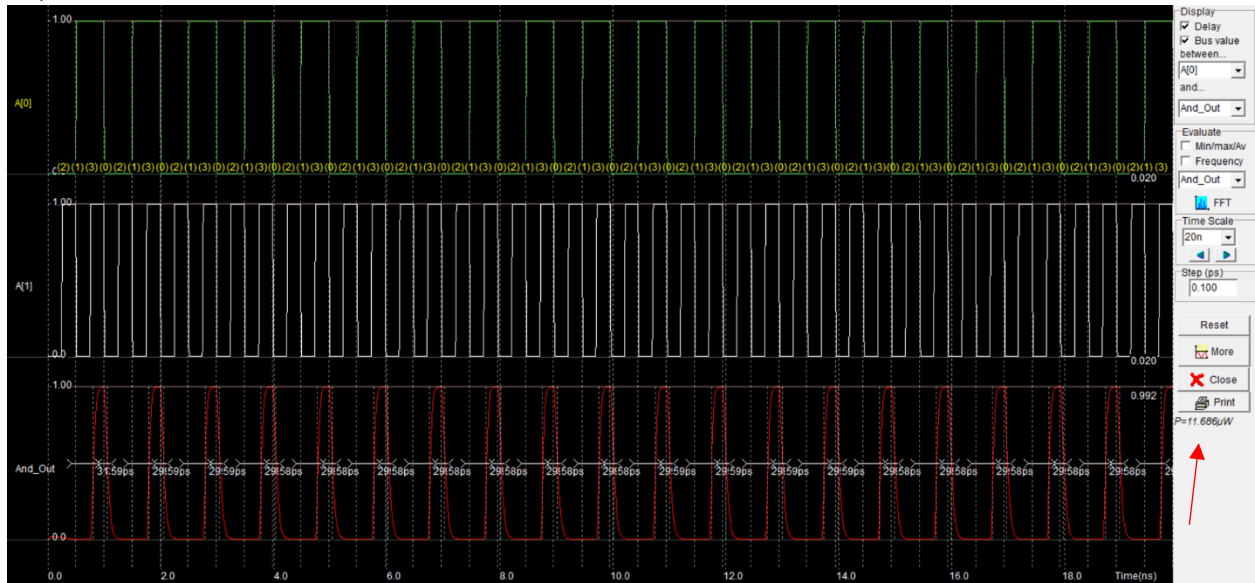
Η κατανάλωση δυναμικής ισχύος βρέθηκε από τις επιμέρους προσομοιώσεις εκεί που δείχνει το βελάκι στις παρακάτω εικόνες.

[illegible]

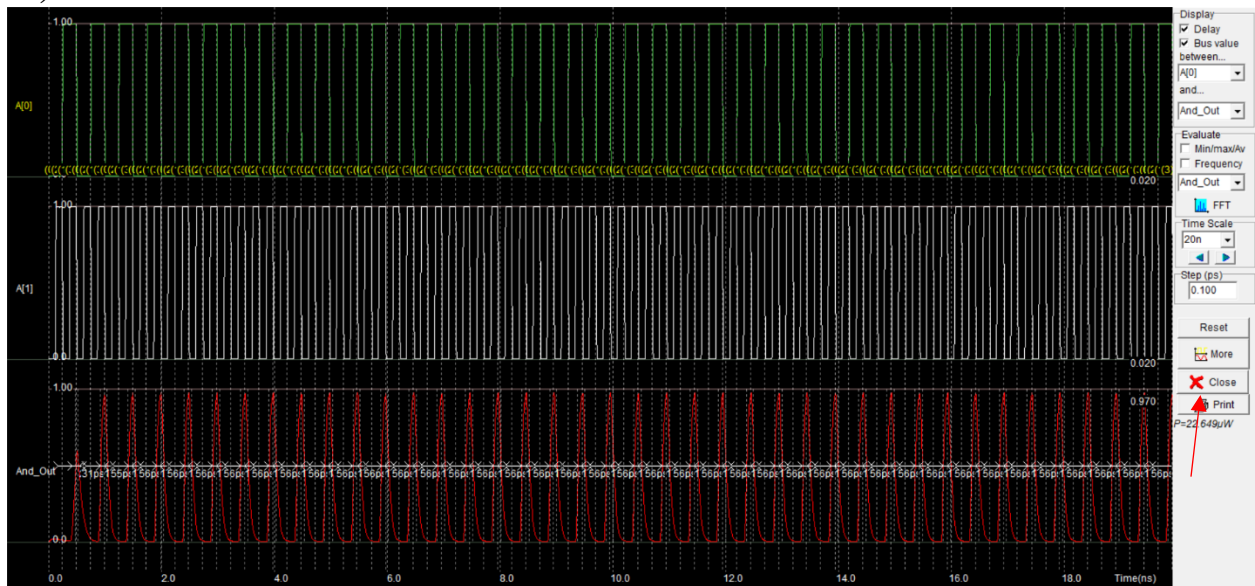
The screenshot displays a logic analyzer interface with three waveforms: A[0], A[1], and And\_Out. The top waveform, A[0], is a high-frequency square wave. The middle waveform, A[1], is a lower-frequency square wave. The bottom waveform, And\_Out, is the logical AND of A[0] and A[1], showing high values only when both A[0] and A[1] are high. The interface includes a control panel on the right with options for Delay, Bus value, and evaluation methods (Min/max/Av, Frequency). The bottom of the screen shows a time scale in nanoseconds (ns) from 0.0 to 18.0.



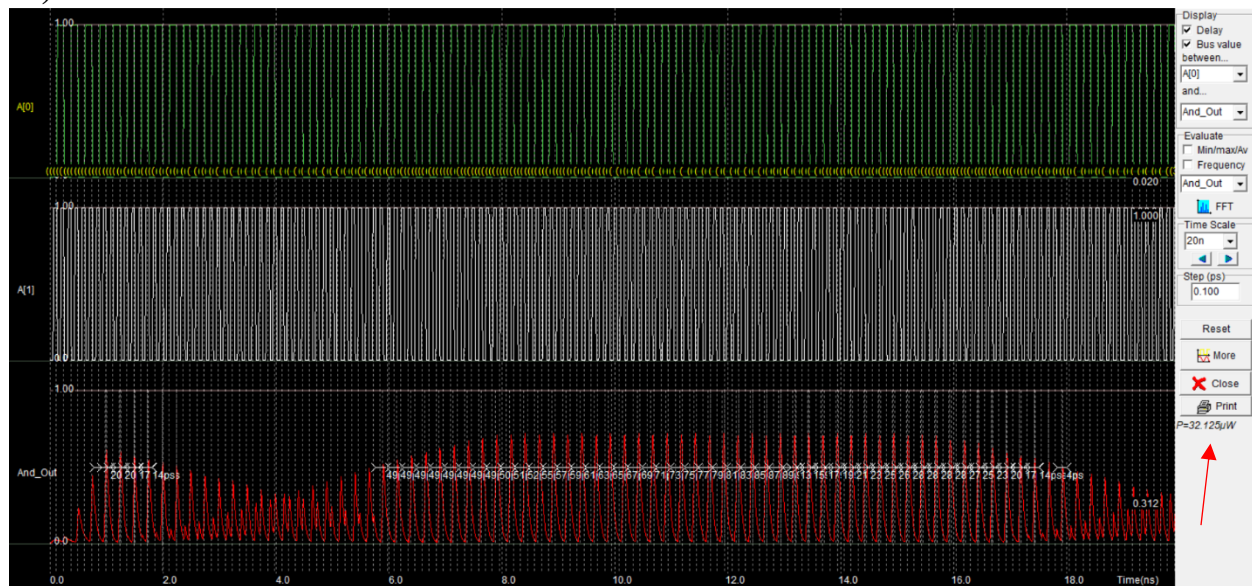
**3<sup>n</sup>)**



4<sup>n</sup>)



5<sup>η</sup>)



Συγκεντρώνοντας τα παραπάνω αποτελέσματα προκύπτει ο ακόλουθος πίνακας μεταβολής της ισχύος σε σχέση με την συχνότητα.

Προσομοιώσεις	Κατανάλωση δυναμικής ισχύος (μW)
1	2,965
2	5,873
3	11,686
4	22,649
5	32,125

Από τα παραπάνω αποτελέσματα φαίνεται ότι όσο αυξάνεται η συχνότητα αυξάνεται και η κατανάλωση δυναμικής ισχύος. Αυτό επαληθεύεται και από τον ακόλουθο τύπο, στον οποίο φαίνεται η αναλογική σχέση μεταξύ  $P_{dynamic} - f$ :

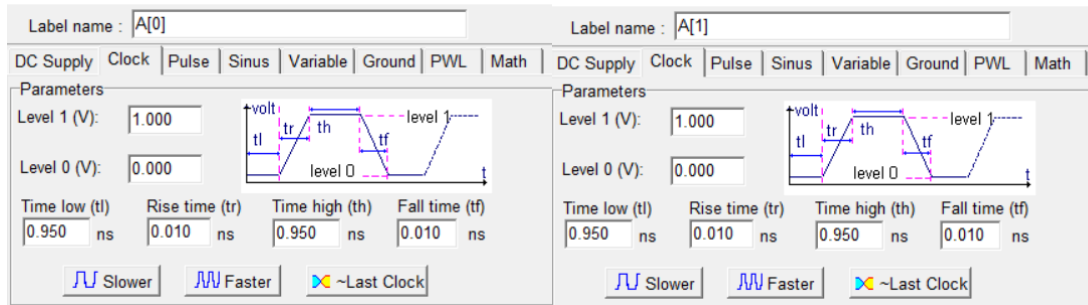
$$P_{dynamic} = P_{switching} + P_{short-circuit} , \text{ όπου}$$

$$P_{dynamic} = \alpha * C * V_{DD}^2 * f$$

## Ερώτηση 2:

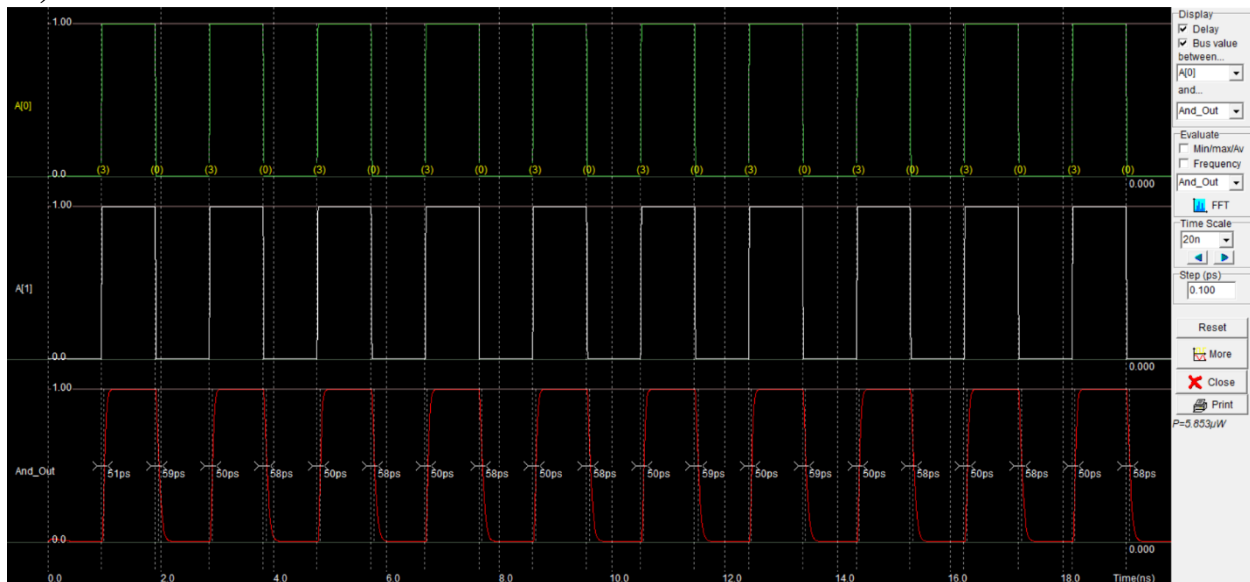
Σε αυτό το ερώτημα θα μελετηθεί η μεταβολή της κατανάλωσης δυναμικής ισχύος όταν μεταβάλλεται το rise time και το fall time κατά 0,01.

Η πρώτη προσομοίωση έτρεξε για τους χρόνους  $t_l = 0.950\text{nsec}$  και  $t_h = 0,950\text{nsec}$  για διάρκεια timescale ίση με  $20\text{nsec}$ .

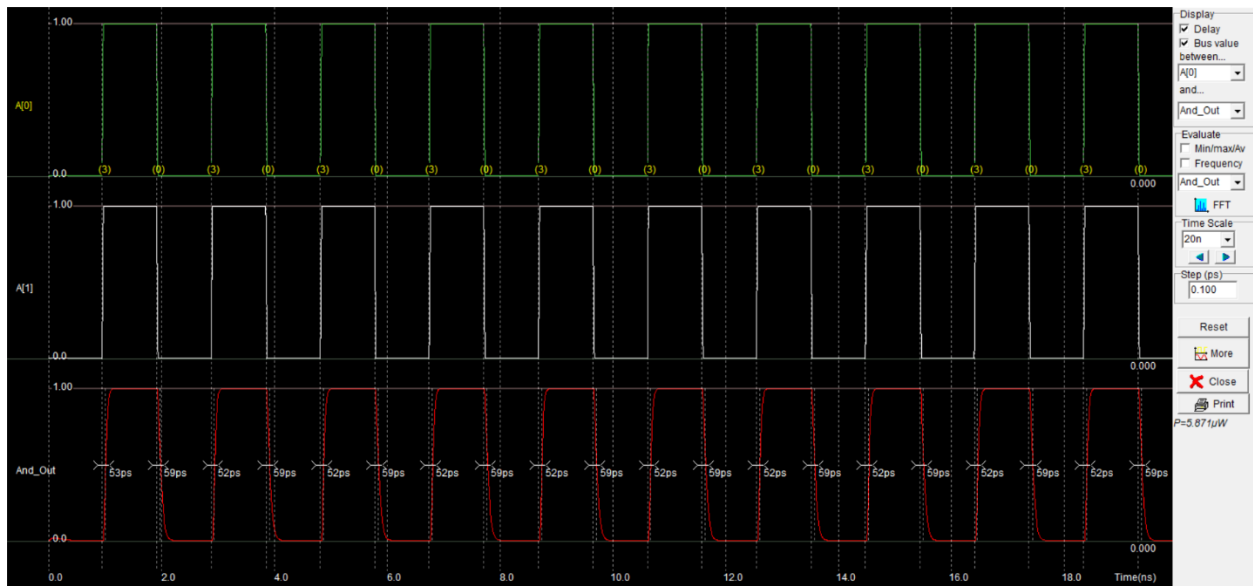


Στις επόμενες προσομοιώσεις αυξήθηκαν το rise time και το fall time κατά 0,01 και η κατανάλωση δυναμικής ισχύος βρέθηκε από τις επιμέρους προσομοιώσεις, στο γνωστό σημείο.

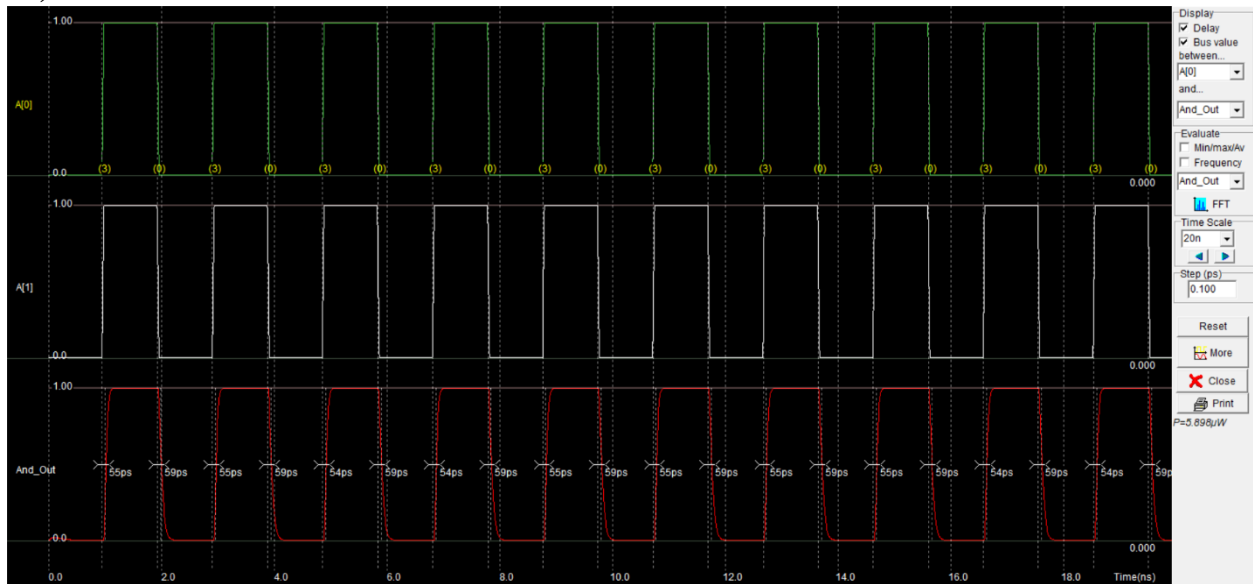
1<sup>η</sup>)



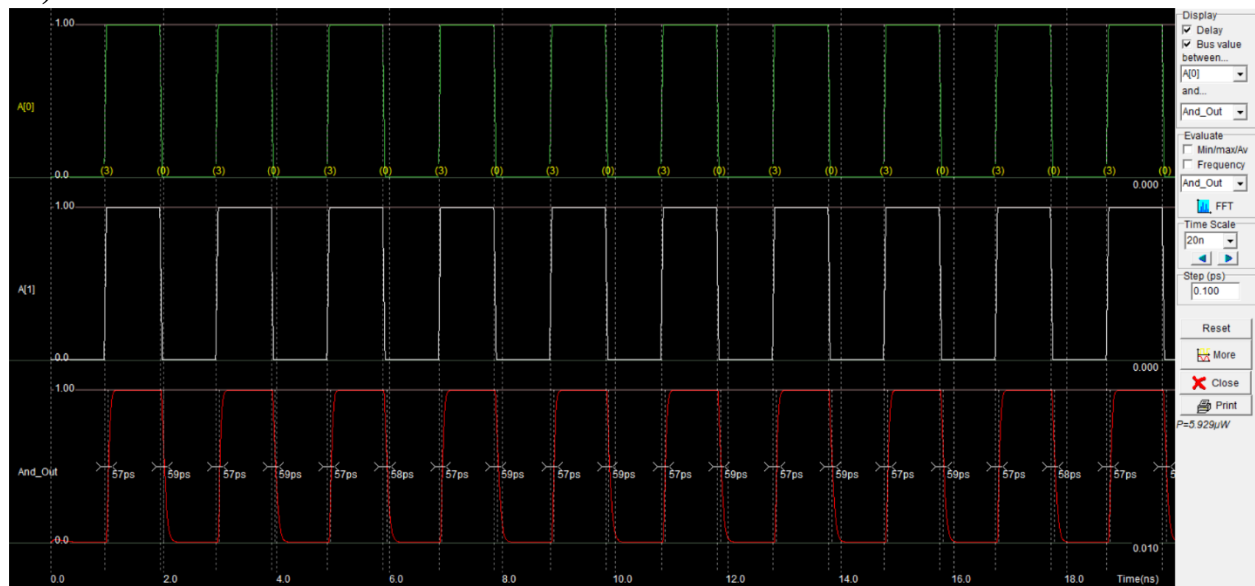
2<sup>η</sup>)



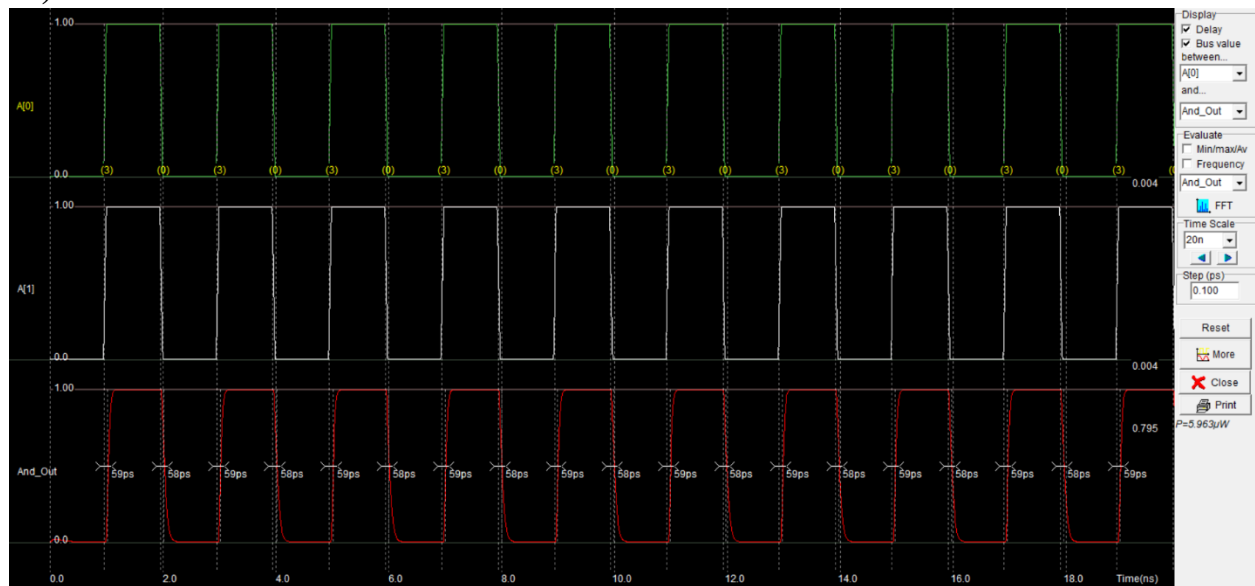
3<sup>n</sup>)



4<sup>η</sup>)



5<sup>η</sup>)



Συγκεντρώνοντας τα παραπάνω αποτελέσματα προκύπτει ο ακόλουθος πίνακας μεταβολής της ισχύος σε σχέση με την συχνότητα.

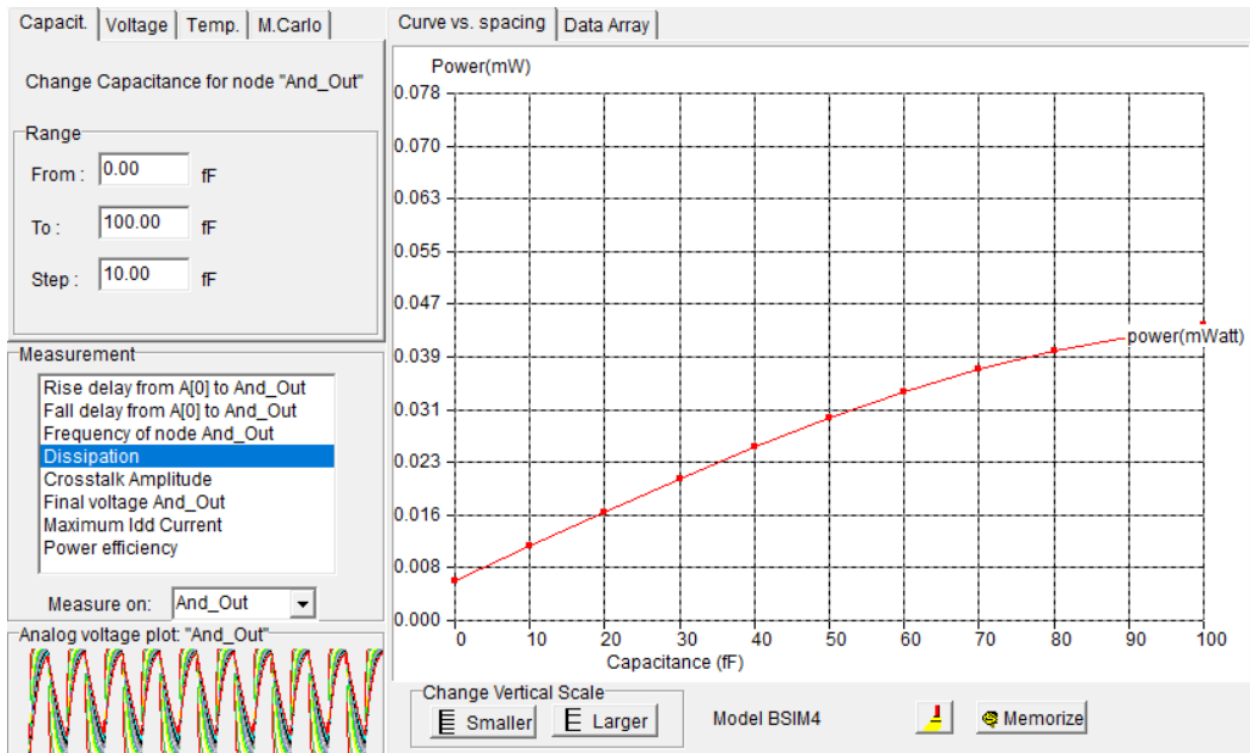
Προσομοιώσεις	Κατανάλωση δυναμικής ισχύος (μW)
1	5,853
2	5,871
3	5,898
4	5,929

Από τα παραπάνω αποτελέσματα φαίνεται ότι όσο αυξάνονται τα rise time και τα fall time κατά 0.01 παρατηρείται ότι η συνολική δυναμική ισχύς αυξάνεται. Αυτό οφείλεται στο γεγονός ότι έχουμε ομαλή μετάβαση από high σε low και από low σε high, δηλαδή κάθε φορά θα έχουμε μεγαλύτερο διάστημα το γινόμενο  $I \cdot V$ . Ένας ακόμη λόγος που οφείλεται για αυτή τη μικρή αύξηση είναι γιατί κάθε φορά η ισχύ που παράγεται από το ρεύμα βραχυκύκλωσης, όταν τα nMOS και pMOS άγουν μερικώς, «προστίθεται» σε αυτούς του χρόνους μετάβασης με αποτέλεσμα να αυξάνει τα high low.

### Ερώτηση 3:

Σε αυτό το ερώτημα θα μετρηθεί η κατανάλωση ισχύος για διάφορες τιμές του πυκνωτή κάνοντας μια «sweep analysis» στο Parametric Analysis του MICROWIND. Από τη θεωρία γνωρίζουμε ότι η κατανάλωση ισχύος σε ένα κύκλωμα οφείλεται στη δυναμική και στη στατική κατανάλωση. Όμως, σε αυτή την περίπτωση η στατική ισχύς δεν θα υπολογιστεί, επειδή τα ρεύματα διαρροής είναι αμελητέα και έτσι έχουν μικρή συνεισφορά στην κατανάλωση. Επιπλέον, η δυναμική κατανάλωση επηρεάζεται από την φόρτιση και την εκφόρτιση των πυκνωτών.

Το αποτέλεσμα αυτής της ανάλυσης είναι το ακόλουθο:





Από το παραπάνω διάγραμμα φαίνεται ότι όσο αυξάνεται η χωρητικότητα του πυκνωτή αυξάνεται και η αντίστοιχη κατανάλωση με γραμμικό τρόπο. Αυτό επαληθεύεται και από τον ακόλουθο τύπο :

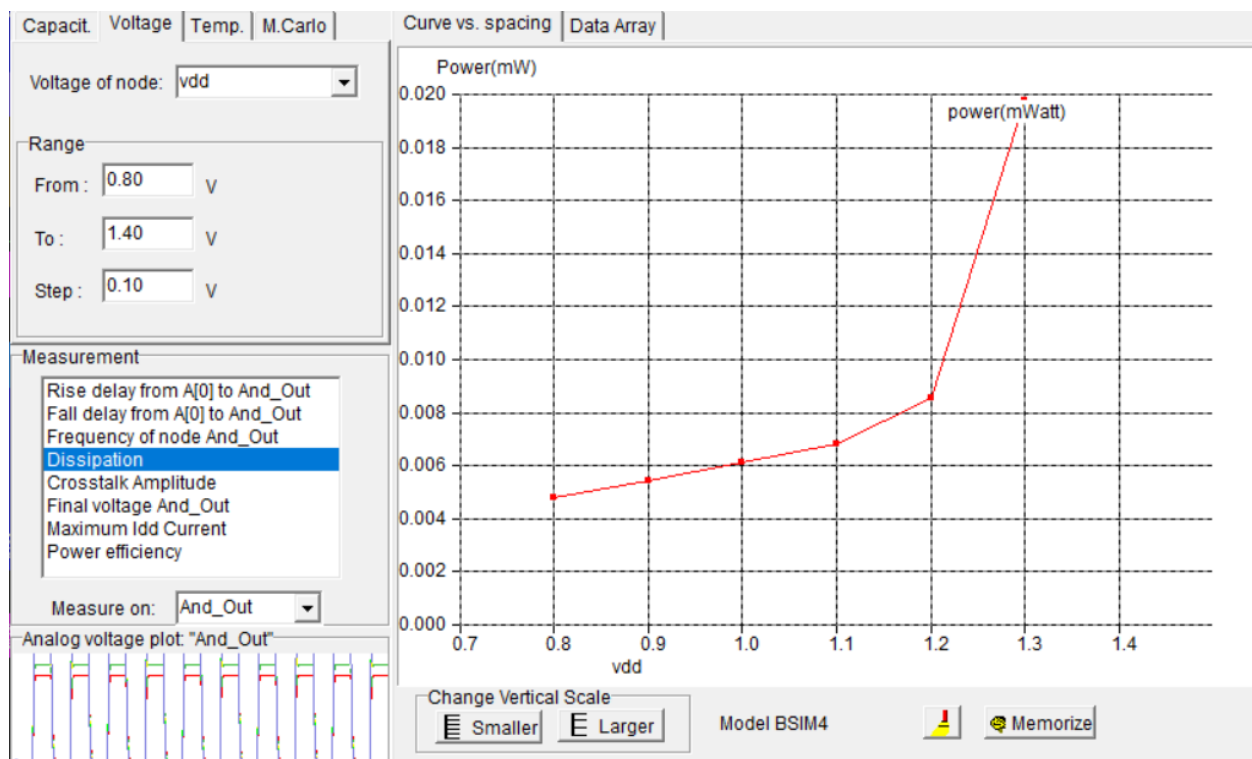
$$P_{\text{switching}} = \alpha \cdot C \cdot V_{DD}^2 \cdot f$$

Όπου

- $P_{\text{dynamic}} = P_{\text{switching}} + P_{\text{short-circuit}}$  , με  $P_{\text{short-circuit}}$  σταθερό και άρα αμελητέο για τη μελέτη
- $V_{DD}$  : τάση τροφοδοσίας
- $f$  : συχνότητα
- $\alpha$  : παράγοντας δραστηριότητας μεταγωγής
- $C$  : συνολική χωρητικότητα κυκλώματος

#### Ερώτηση 4:

Σε αυτό το ερώτημα θα μετρηθεί η κατανάλωση ισχύος για διάφορες τιμές της τροφοδοσίας κάνοντας μια «sweep analysis» στο Parametric Analysis του MICROWIND. Το αποτέλεσμα αυτής της ανάλυσης είναι το ακόλουθο:



Από το παραπάνω διάγραμμα φαίνεται ότι καθώς αυξάνεται η τάση τροφοδοσίας  $V_{DD}$  αυξάνεται και η  $P_{\text{switching}}$  ανάλογα με το  $V_{DD}^2$ , κάτι που γίνεται περισσότερο εμφανές όσο αυξάνεται η τιμή της τάσης τροφοδοσίας.