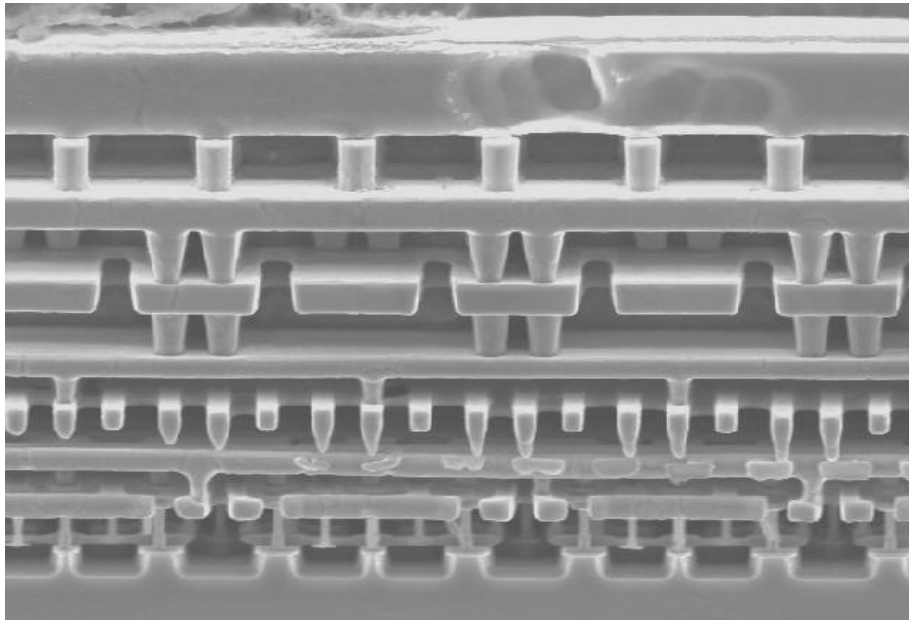
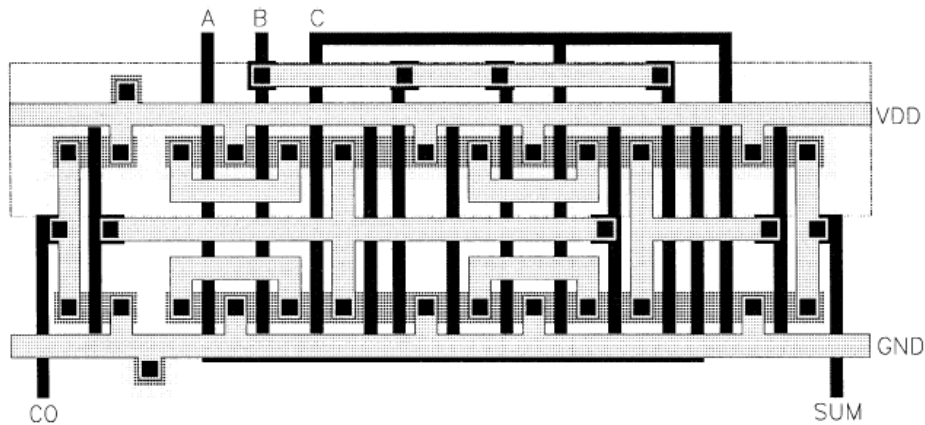




ΕΛΜΕΠΙΑ / ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Εργαστηριακές Ασκήσεις του μαθήματος:  
**«Μικροηλεκτρονική και VLSI»**  
+ Εγχειρίδιο περιβάλλοντος **Microwind**



Νικόλαος Πετράκης, Καθηγητής Εφαρμογών  
Ηλίας Μαλαμάς, Εργαστηριακός Συνεργάτης

Χανιά 2020



## Εισαγωγή

Το αντικείμενο των εργαστηριακών ασκήσεων του μαθήματος «Μικροηλεκτρονική και VLSI» είναι η μελέτη της λειτουργίας των MOSFET στοιχείων και ο φυσικός σχεδιασμός (full-custom design) καθώς και η εξομοίωση (simulation) ψηφιακών κυκλωμάτων τεχνολογίας CMOS (Complementary Metal-Oxide-Semiconductor).

Το λογισμικό πακέτο που θα χρησιμοποιηθεί ονομάζεται **Microwind** το οποίο αναπτύχθηκε από τον **Etienne Sicard** στην Τουλούζη με αρχικό στόχο μόνο την εκπαιδευτική χρήση, ενώ στην συνέχεια απέκτησε και εμπορικό κομμάτι. Στην ιστοσελίδα: <http://www.microwind.org/> μπορείτε να βρείτε λεπτομέρειες καθώς και να κατεβάσετε, χωρίς επιβάρυνση, μια λιτή (lite) έκδοση με τα βασικά στοιχεία του πακέτου. Το MICROWIND είναι ένα φιλικό εργαλείο που τρέχει σε προσωπικούς υπολογιστές συμβατούς με IBM PC με λειτουργικό σύστημα WINDOWS (95, 98, NT, XP) για τον σχεδιασμό και την εξομοίωση μικροηλεκτρονικών κυκλωμάτων στο επίπεδο της φυσικής σχεδίασης (layout).

Το πρώτο μάθημα θα είναι μια σύντομη γνωριμία με το περιβάλλον σχεδίασης και των δυνατοτήτων που μας παρέχει ενώ από το δεύτερο θα αρχίσουμε την σχεδίαση απλών λογικών πυλών και προοδευτικά θα φτάσουμε να σχεδιάζουμε περισσότερο πολύπλοκα συνδυαστικά και ακολουθιακά κυκλώματα, πάντα σε επίπεδο layout. Κάθε κύκλωμα (ξεκινώντας από τον αντιστροφέα και τις βασικές πύλες δύο ή περισσότερων εισόδων και συνεχίζοντας με πολυπλέκτες, αθροιστές, δισταθή (Latches, Flip-Flops), κύτταρα μνήμης κλπ) θα μελετάται στο χαρτί, θα σχεδιάζεται στον Η/Υ και κατόπιν θα εξομοιώνεται ώστε να γίνει ο χρονικός χαρακτηρισμός του (μέτρηση καθυστέρησης απόκρισης, ευαισθησία καθυστέρησης απόκρισης ανάλογα με το χωρητικό φορτίο που οδηγεί κτλ.).

Μετά από κάθε ολοκληρωμένη – ελεγμένη σχεδίαση μπορεί να χρησιμοποιηθεί η επιλογή 3D (Process View in 3D), η οποία δίνει τη δυνατότητα τρισδιάστατης άποψης των κυκλωμάτων που σχεδιάστηκαν και περιγράφει σταδιακά την διαδικασία της κατασκευής του τρέχοντος κυκλώματος με την επιλεγμένη τεχνολογία. Η επιλογή αυτή είναι αρκετά χρήσιμη διότι βοηθάει στην πλήρη κατανόηση των περιγραφών της διαδικασίας κατασκευής ολοκληρωμένων κυκλωμάτων, παρουσιάζοντάς τις με πολύ παραστατικό τρόπο.

## Πίνακας περιεχομένων

Εισαγωγή .....	ii
Πίνακας περιεχομένων.....	iii
1. Εξοικείωση με το περιβάλλον εργασίας του Microwind.....	1
2. Η λογική CMOS και τα τρανζίστορ nMOS και pMOS. ....	9
3. Κατασκευή και εξομοίωση CMOS αντιστροφέα.....	14
4. Σχεδιασμός και εξομοίωση απλών πυλών σε CMOS λογική.....	20
5. Σχεδιασμός πολυπλέκτη από 2 σε 1 με απλές πύλες. ....	26
6. Σχεδίαση και υλοποίηση σύνθετων πυλών CMOS.....	29
7. Σχεδίαση και υλοποίηση της πύλης XOR. ....	33
8. Σχεδίαση και υλοποίηση του ημι-αθροιστή και του πλήρους αθροιστή..	37
9. Μελέτη και υλοποίηση ακολουθιακών κυκλωμάτων.....	43
10. CMOS Μανδαλωτές και Flip-Flops.....	52
11. Σχεδιασμός και εξομοίωση στοιχείων μνήμης. ....	58
12. Σχεδιασμός και εξομοίωση εφαρμογών της CMOS τεχνολογίας.....	61
<b>Παραρτήματα:</b>	
A. Χειρισμός & Εντολές Προγράμματος Microwind2. ....	67
B. Οδηγός γρήγορης αναφοράς του Microwind.....	91
<b>Βιβλιογραφία .....</b>	<b>96</b>

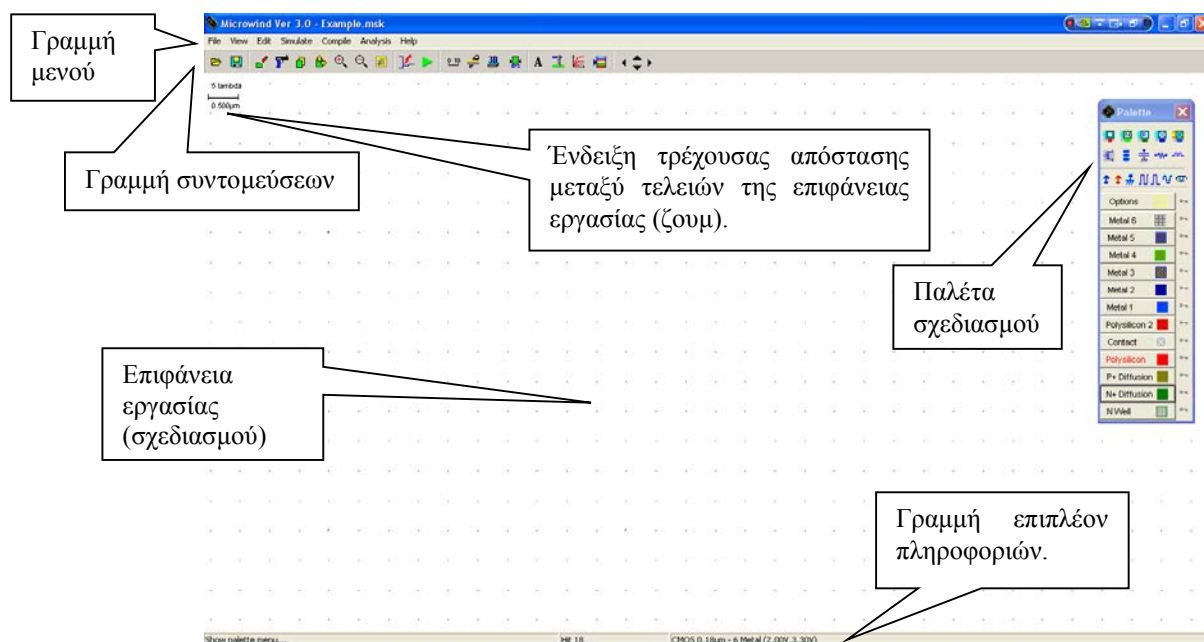
## 1. Εξοικείωση με το περιβάλλον εργασίας του Microwind.

**Σκοπός:** Σε αυτή την άσκηση θα γίνει μια πρώτη γνωριμία με το περιβάλλον Microwind2, το οποίο είναι εργαλείο σχεδιασμού CMOS κυκλωμάτων και εξοικείωση με τις βασικές του λειτουργίες.

### Θεωρητικό υπόβαθρο

Το εργαλείο σχεδιασμού κυκλωμάτων Microwind παρέχει τη δυνατότητα φυσικού σχεδιασμού (layout) τόσο ψηφιακών όσο και αναλογικών – RF – κυκλωμάτων, αλλά και εξομοίωσής τους για επαλήθευση λειτουργίας και χρονικό χαρακτηρισμό των υπό σχεδίαση κυκλωμάτων. Ο σχεδιασμός με το πρόγραμμα αυτό βασίζεται στη γραφική αποτύπωση σε δυσδιάστατη επιφάνεια εργασίας περιοχών με διαφορετικές ηλεκτρικές ιδιότητες (ημιαγωγοί, αγωγοί, μονωτές) οι οποίες ονομάζονται περιοχές. Συνεπώς, ο χρήστης του προγράμματος βλέπει το σχέδιό του σε κάτοψη.

Με την εκτέλεση του εικονιδίου του στην επιφάνεια εργασίας ή από το μενού "Εναρξη" -> "Όλα τα προγράμματα" -> "Microwind" των Windows εμφανίζεται το αρχικό παράθυρο του προγράμματος το οποίο χρησιμοποιείται και ως επιφάνεια εργασίας στο σχεδιασμό.



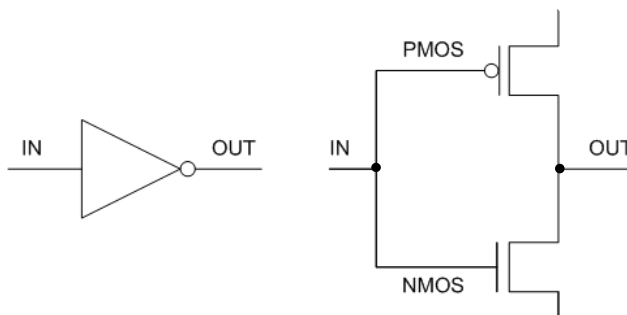
Σχήμα 1.1: Αρχικό παράθυρο του προγράμματος Microwind.

Λεπτομέρειες για τη χρήση κάθε ενός από τα παραπάνω βασικά μέρη του προγράμματος μπορείτε να βρείτε στον οδηγό χρήσης του προγράμματος. Παρακάτω θα δούμε μερικές από τις βασικές δυνατότητες του εργαλείου, μέσα από ορισμένα απλά παραδείγματα.

### Πρακτική διαδικασία

Σχεδιασμός και εξομοίωση ενός CMOS αντιστροφέα.

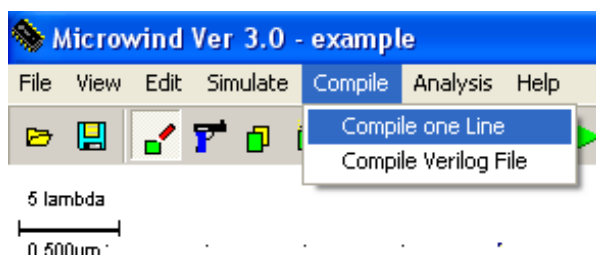
Το σύμβολο ενός τυπικού αντιστροφέα φαίνεται παρακάτω, μαζί με το κύκλωμα τρανζίστορ.



Σχήμα 1.2: Σύμβολο και κύκλωμα ενός CMOS αντιστροφέα.

Για το σχεδιασμό του στο Microwind κάνουμε τα εξής:

1. Εκτελέστε το πρόγραμμα για να εμφανιστεί η βασική του οθόνη σχεδιασμού, όπως στο Σχήμα 1.1.
2. Επιλέξτε το μενού "File"-> "Select Foundry"-> "cmos018.rul" για να χρησιμοποιήσετε την τεχνολογία CMOS 0.18μm 6Metal, την οποία υποστηρίζει το Microwind, μεταξύ άλλων τεχνολογιών σχεδιασμού.
3. Επιλέξτε το μενού "Compile"-> "Compile one line":

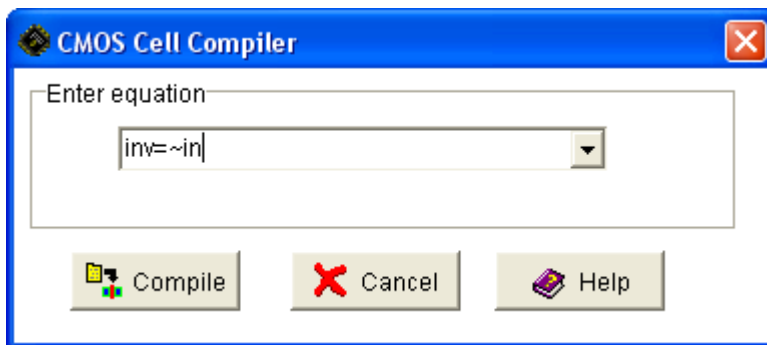


Σχήμα 1.3: Η λειτουργία "Compile".

Το παρακάτω παράθυρο εμφανίζεται το οποίο επιτρέπει την επιλογή της έκφρασης της λογικής συνάρτησης προς υλοποίηση. Τα παρακάτω σύμβολα χρησιμοποιούνται για το συμβολισμό των βασικών λογικών πράξεων:

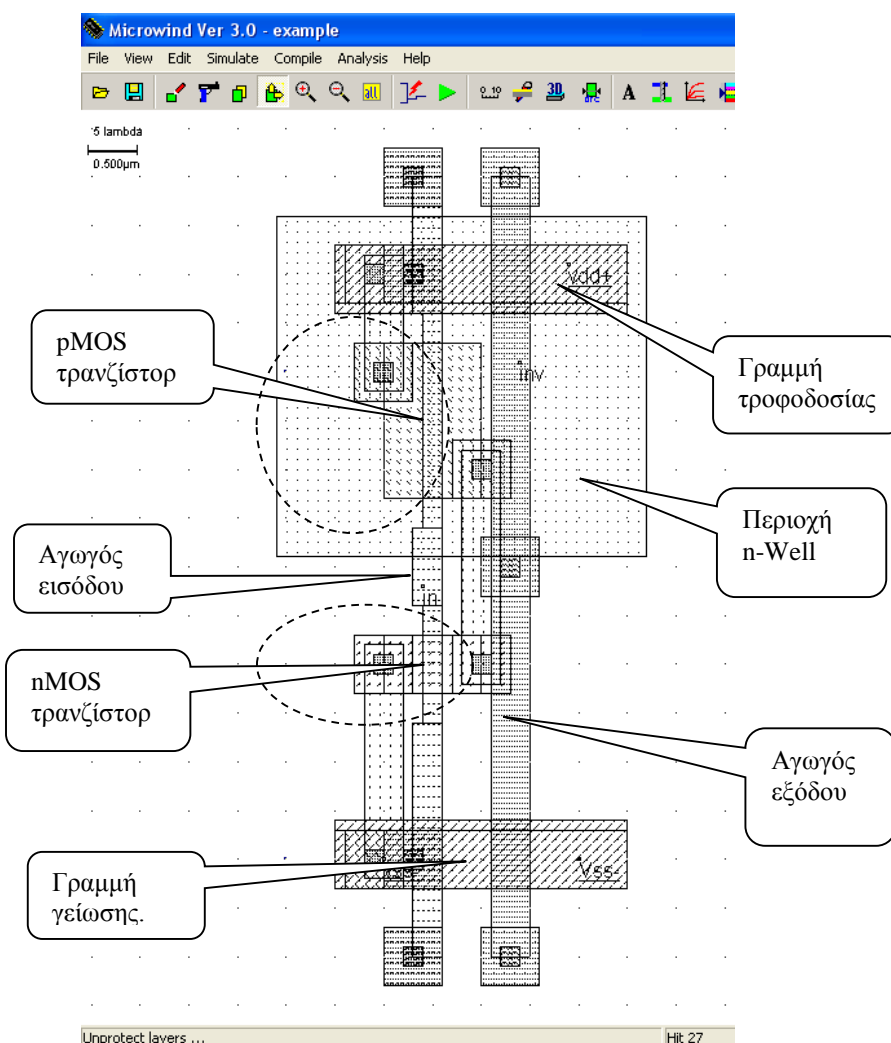
~	NOT
	OR
&	AND
^	XOR
~^	XNOR

Κατεβάζοντας τη λίστα εμφανίζονται ορισμένες έτοιμες εκφράσεις τις οποίες μπορεί να μεταγλωττίσει το πρόγραμμα σε φυσικό σχέδιο. Στο παράθυρο αυτό μπορείτε να εισάγετε δικές σας συναρτήσεις με βάση τον παραπάνω πίνακα προκειμένου να τις μεταγλωττίσει το πρόγραμμα.



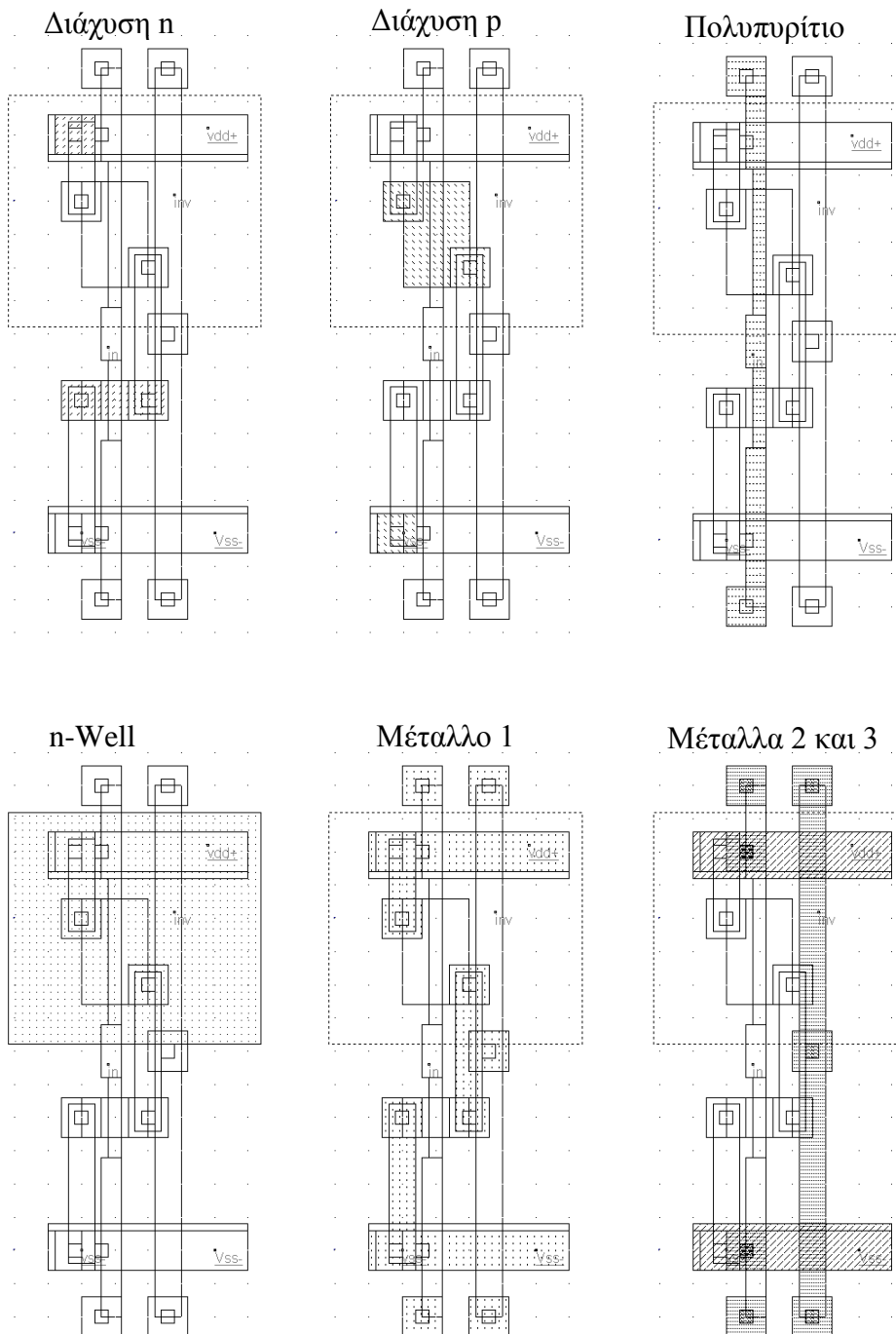
Σχήμα 1.4: Το παράθυρο του μεταγλωττιστή για εισαγωγή συναρτήσεων.

4. Επιλέξτε την προτεινόμενη έκφραση (λογική περιγραφή ενός αντιστροφέα) και πατήστε "Compile". Το παρακάτω σχέδιο εμφανίζεται στην επιφάνεια εργασίας:




Σχήμα 1.5: Βασικά μέρη του φυσικού σχεδίου ενός αντιστροφέα.

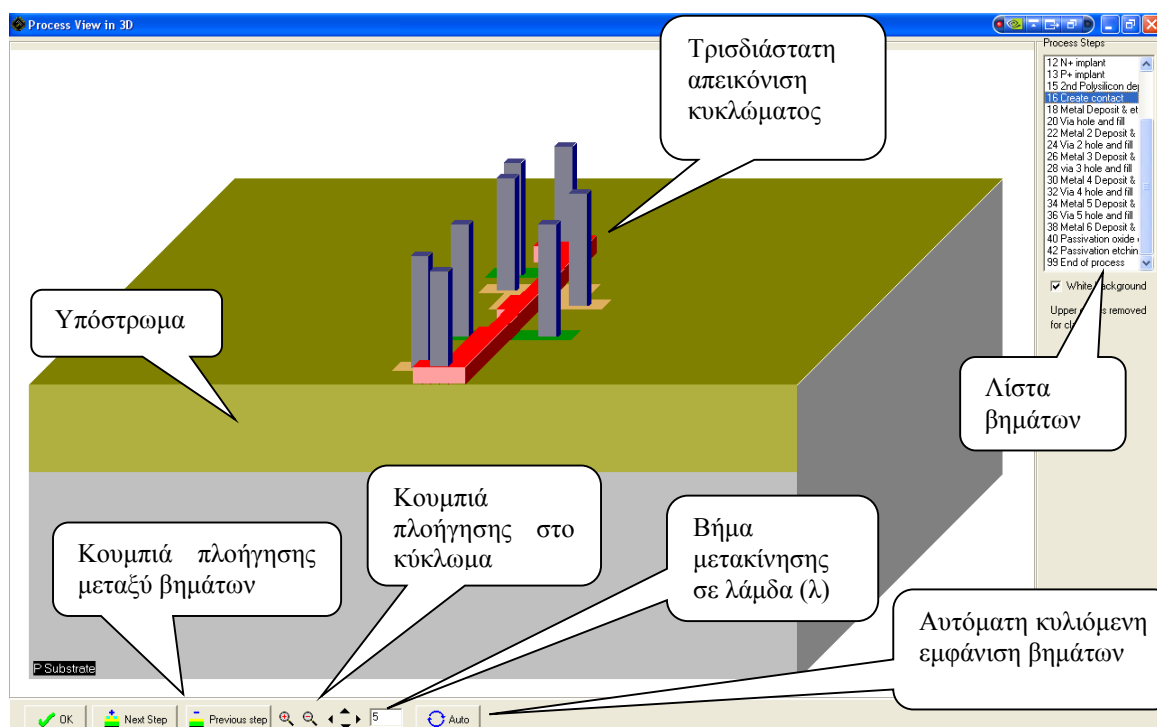
Παρακάτω φαίνονται οι βασικές περιοχές (υλικά) που εμπλέκονται στην κατασκευή του αντιστροφέα:



Σχήμα 1.6: Οι μάσκες που εμπλέκονται στην παραγωγή ενός CMOS αντιστροφέα.


5. Για την εξομοίωση της διαδικασίας της ολοκλήρωσης, δηλαδή, της χρήσης των βασικών μασκών που εμπλέκονται στην κατασκευή των περιοχών του αντιστροφέα, υπάρχει ένα εργαλείο τρισδιάστατης απεικόνισης, το οποίο μπορείτε να καλέσετε κάνοντας κλικ στο

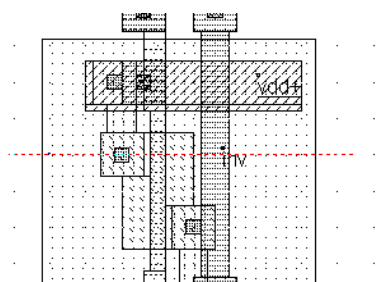
εικονίδιο . Έτσι έχετε την δυνατότητα να παρακολουθήσετε με πολύ παραστατικό τρόπο τα βήματα που γίνονται στην διαδικασία παραγωγής των ολοκληρωμένων κυκλωμάτων. Το Σχήμα 1.7 παρουσιάζει ένα τέτοιο ενδιαμέσο βήμα.



Σχήμα 1.7: Η εφαρμογή τρισδιάστατης αναπαράστασης της επίστρωσης των περιοχών-υλικών.

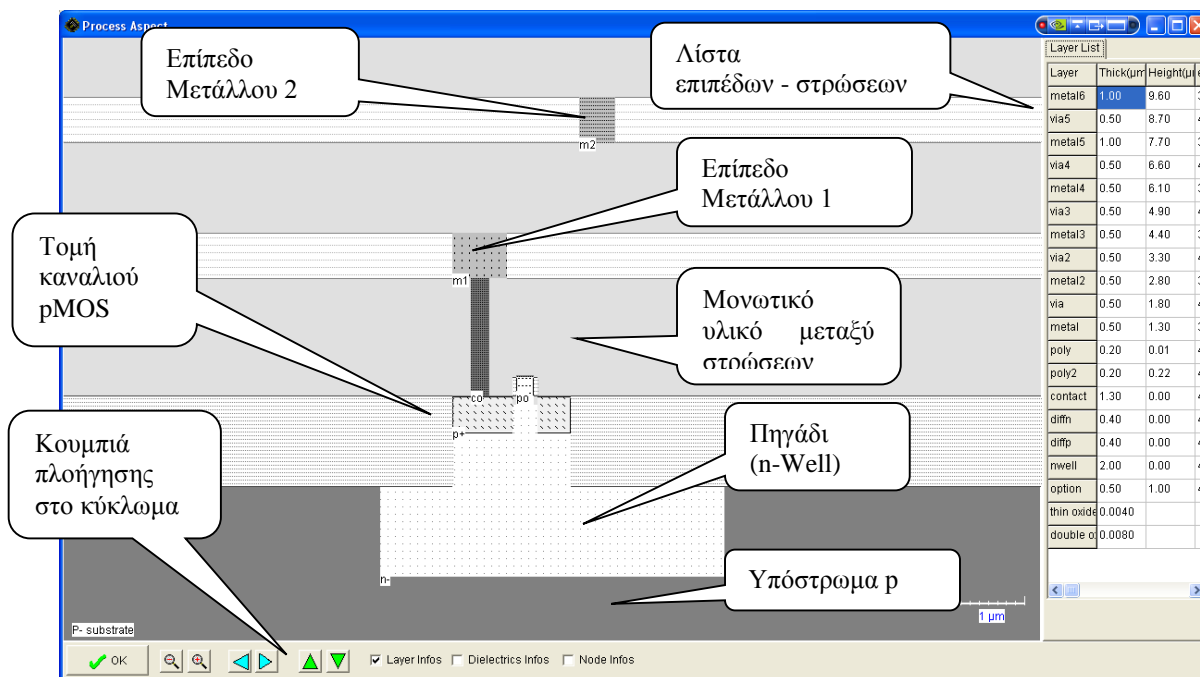
Στην εφαρμογή 3D υπάρχει η δυνατότητα γραφικής αναπαράστασης της διαδικασίας παραγωγής, είτε χειροκίνητα (βήμα-βήμα), είτε αυτόματα (auto) με την κυλιόμενη εμφάνιση των διαδοχικών βημάτων. Υπάρχει η δυνατότητα μεγέθυνσης/σμίκρυνσης και μετακίνησης προς τις τέσσερις βασικές κατευθύνσεις. Το μέγεθος της μετακίνησης σε κάθε κλικ ρυθμίζεται με το πλαίσιο κειμένου αριστερά από το κουμπί "Auto". Στο δεξιό πάνω άκρο της εφαρμογής υπάρχει σύντομη επεξήγηση του κάθε βήματος.

6. Για τη δισδιάστατη απεικόνιση τμημάτων του κυκλώματος, μπορεί να χρησιμοποιηθεί το εργαλείο τομής . Χαράσσοντας μια γραμμή σε οποιοδήποτε σημείο του κυκλώματος όπως εδώ στην περιοχή του pMOS τρανζίστορ:



Σχήμα 1.8: Η γραμμή που χαράσσουμε για τη δημιουργία μιας τομής.

Με αυτόν τον τρόπο θα λάβουμε μια απεικόνιση της τομής παρόμοια με αυτήν που παρουσιάζεται στο Σχήμα 1.9 που ακολουθεί.



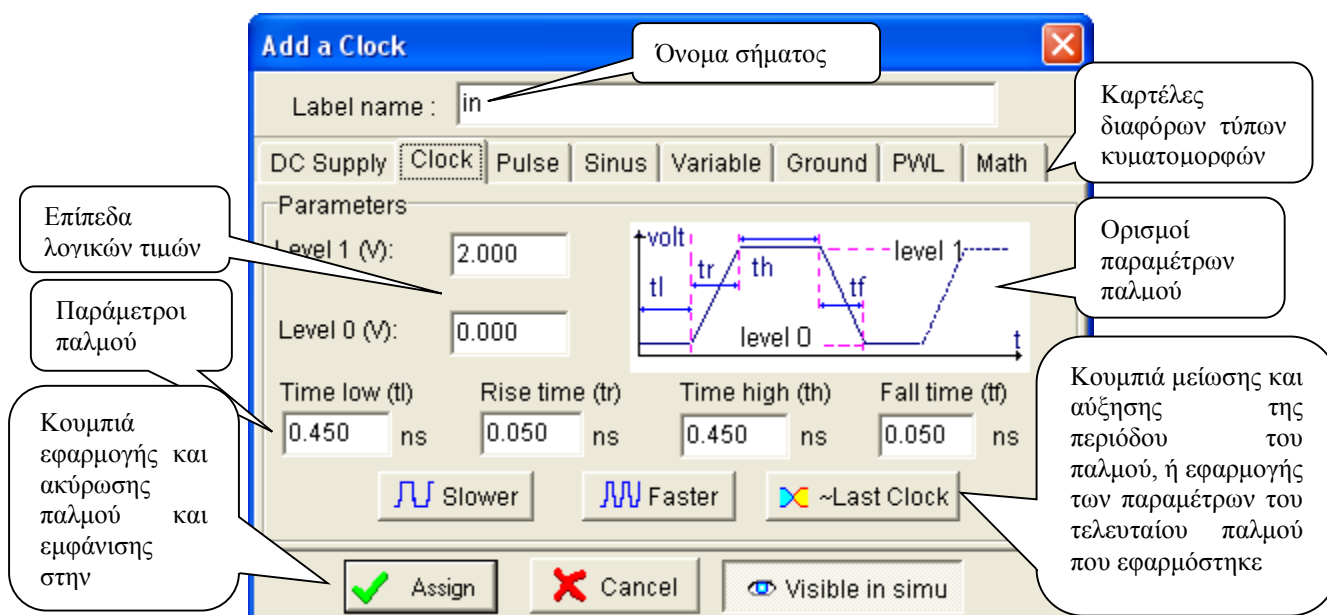
Σχήμα 1.9: Η εγκάρσια τομή στο pMOS τρανζίστορ του αντιστροφέα.

Οι περιοχές που τέμνονται από τη γραμμή που χαραμάσαμε, εμφανίζονται στην εικόνα σε στρώματα, όπως είναι χαραγμένα πάνω στο δισκίο πυριτίου του ολοκληρωμένου που σχεδιάζουμε. Έτσι για παράδειγμα, στο κάτω μέρος φαίνεται η τομή του p-τρανζίστορ του αντιστροφέα, η τομή του πρώτου επιπέδου μετάλλου (m1) και του δεύτερου επιπέδου μετάλλου (m2) και ενδιάμεσα μονωτικό υλικό που εξασφαλίζει την ηλεκτρική απομόνωση των παραπάνω στοιχείων μεταξύ τους.

7. Για την εξομοίωση της λειτουργίας του αντιστροφέα και την επαλήθευση της ορθής του συμπεριφοράς, χρησιμοποιούμε τη λειτουργία του μενού "Simulate". Πριν από αυτό εφαρμόζουμε μια γεννήτρια παλμοσειράς με συγκεκριμένη συχνότητα, επιλέγοντας το




εικονίδιο (τέταρτο από αριστερά στην τρίτη σειρά εργαλείων της παλέτας σχεδιασμού). Κατόπιν κάνουμε κλικ στην κουκίδα του ονόματος του σήματος εισόδου (εδώ: **in**) που βρίσκεται στη μέση της γραμμής πολυπυριτίου. Η ακόλουθη εικόνα εμφανίζεται:




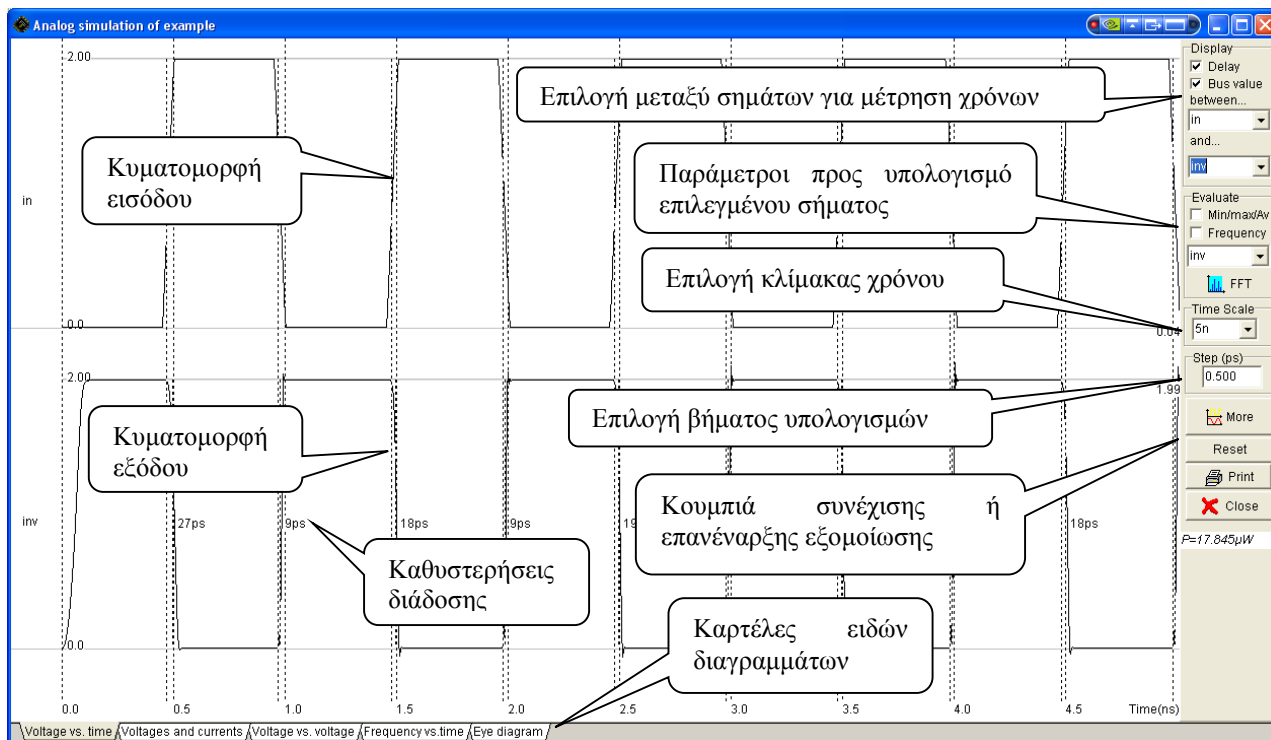
Σχήμα 1.10: Το παράθυρο εισαγωγής κυματομορφών.

Πατάμε "Assign" για την εφαρμογή της γεννήτριας στο επιλεγμένο σημείο και παρατηρούμε ότι το όνομα του σήματος εισόδου έχει αποκτήσει πλαγιαστή μορφοποίηση (*italics*). Αυτό σημαίνει ότι η συμπεριφορά της εισόδου θα είναι ορατή στο παράθυρο της εξομοίωσης.

8. Για να είναι ορατή και η συμπεριφορά της εξόδου **inv** στην εξομοίωση, επιλέγουμε το

εικονίδιο  (πρώτο από δεξιά στην τρίτη σειρά της παλέτας σχεδιασμού) και κατόπιν κάνουμε κλικ στην κουκίδα του ονόματος του σήματος εξόδου **inv**. Ομοίως θα παρατηρήσουμε ότι η γραμματοσειρά θα αλλάξει σε πλάγια γραφή.

9. Τέλος, πατάμε το κουμπί εξομοίωσης  (ευρίσκεται στην μέση της γραμμής εργαλείων) και παίρνουμε μια εικόνα όπως η ακόλουθη:



Σχήμα 1.11: Το παράθυρο της εξομοίωσης – απόκριση του υπό εξέταση αντιστροφέα.

Ο αντιστροφέας είναι έτοιμος!

10. Για την αποθήκευση του σχεδίου σας στον υπολογιστή χρησιμοποιείτε την επιλογή "File" > "Save as", οπότε και εμφανίζεται το τυπικό παράθυρο αποθήκευσης των Windows, στο οποίο εισάγετε το επιθυμητό όνομα του σχεδίου σας. Σε αυτό προσκολλάται αυτόματα η κατάληξη ".msk". Χρειάζεται προσοχή ώστε η αποθήκευση να γίνεται σε οποιοδήποτε σημείο εκτός της επιφάνειας εργασίας των Windows, καθώς σε αυτή την περίπτωση το πρόγραμμα όχι μόνο δεν αποθηκεύει το σχέδιο, αλλά ούτε και εμφανίζει κάποια σχετική ειδοποίηση (warning) για να ενημερωθεί ο χρήστης. Άρα, θα πρέπει να σώζετε συχνά τα σχέδιά σας για αποφυγή απώλειας της δουλειάς σας.

## Ασκήσεις

Σε κάθε μια από τις παρακάτω ασκήσεις αναγνωρίστε τα p και n τρανζίστορ των κυκλωμάτων που προκύπτουν.

- 1.1 Να φτιαχτεί με το μεταγλωττιστή και να εξομοιωθεί μια NAND πύλη 2 εισόδων και μια NOR πύλη δύο εισόδων. Να φτιαχτούν οι αντίστοιχες πύλες AND και OR 2 εισόδων. Τι παρατηρείτε συγκρίνοντας τα φυσικά σχέδια των αντίστοιχων πυλών όσον αφορά στις διαφορές τους;
- 1.2 Να φτιαχτεί με το μεταγλωττιστή και να εξομοιωθεί μια NAND πύλη 3 εισόδων και μια NOR πύλη 4 εισόδων.
- 1.3 Να φτιαχτεί με το μεταγλωττιστή και να εξομοιωθεί η λογική συνάρτηση:

$$F=(A+B)*(A+C)$$

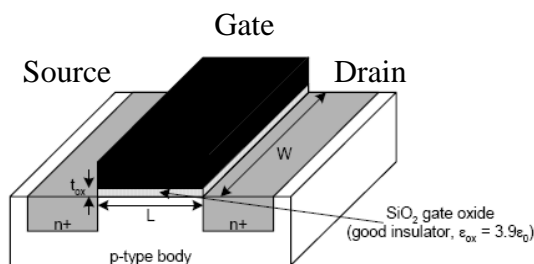
## 2. Η λογική CMOS και τα τρανζίστορ nMOS και pMOS.

**Σκοπός:** Στην άσκηση αυτή γίνεται μια εισαγωγή στις έννοιες τις λογικής CMOS και των τρανζίστορ που θα χρησιμοποιηθούν στην κατασκευή των κυκλωμάτων του Εργαστηρίου Μικροηλεκτρονικής και VLSI.

### Θεωρητικό υπόβαθρο

Η λογική CMOS (Complementary Metal-Oxide-Semiconductor) χρησιμοποιείται σήμερα ευρύτατα στην κατασκευή Ολοκληρωμένων Κυκλωμάτων. Η βάση της κατασκευής οποιουδήποτε κυκλώματος σε CMOS λογική είναι τα τρανζίστορ nMOS και pMOS. Εδώ θα αναφέρουμε περιληπτικά τις κυριότερες ιδιότητες αυτών των τρανζίστορ, ώστε να μπορούμε να τα χρησιμοποιήσουμε με άνεση στις υπόλοιπες εργαστηριακές ασκήσεις για την κατασκευή χρήσιμων κυκλωμάτων.

Παρακάτω φαίνεται η εικόνα ενός MOS τρανζίστορ και οι παράμετροι που κατά κύριο λόγο επηρεάζουν τη λειτουργία του. Η εικόνα αυτή φαίνεται από την κάθετη τομή στο δισκίο ενός Ολοκληρωμένου Κυκλώματος, δηλαδή αν νοητά κόψουμε το υλικό πάνω στο οποίο έχει κατασκευαστεί το τρανζίστορ και φέρουμε μπροστά μας την τομή.



Σχήμα 2.1: Η τρισδιάστατη δομή ενός MOS τρανζίστορ όπως φαίνεται σε τομή.

Το MOS τρανζίστορ έχει τρεις ακροδέκτες με τα δυναμικά των οποίων ελέγχεται η κατάσταση λειτουργίας του (υπάρχει και τέταρτος ακροδέκτης, αυτός του υποστρώματος-substrate- αλλά επειδή το δυναμικό του είναι σταθερά συνδεδεμένο είτε στη γείωση είτε στην τροφοδοσία δεν τον λαμβάνουμε υπόψη). Ο ακροδέκτης Πύλης (Gate) δέχεται σήματα εισόδου και με το δυναμικό του εξασφαλίζεται η ύπαρξη ή όχι καναλιού για τη διέλευση φορέων μεταξύ των ακροδεκτών Πηγής (Source) και Απαγωγού (Drain). Με τις κατάλληλες διαφορές δυναμικού, φορείς απάγονται από την Πηγή και έλκονται στον Απαγωγό, ώστε να εξασφαλίζεται ροή φορέων και επομένως ρεύμα μεταξύ αυτών των ακροδεκτών. Σε αυτή την περίπτωση το τρανζίστορ άγει, η δε ένταση του ρεύματος αγωγής εξαρτάται από τη διαφορά δυναμικού μεταξύ Πηγής και Απαγωγού. Να σημειωθεί εδώ ότι **η βασική διαφορά** μεταξύ nMOS και pMOS είναι ότι το ρεύμα στο πρώτο οφείλεται σε ροή ηλεκτρονίων ενώ στο δεύτερο σε ροή οπών. Συνεπώς, οι περιοχές διάχυσης Πηγής και Απαγωγού στο μεν πρώτο είναι αρνητικά φορτισμένες στο δε δεύτερο θετικά.

Μερικά από τα βασικά χαρακτηριστικά ενός MOS τρανζίστορ είναι το πλάτος  $W$  του καναλιού αγωγής και το μήκος του καναλιού  $L$ , το πάχος  $t_{ox}$  του μονωτή, καθώς και το δυναμικό κατωφλίου  $V_t$  το οποίο προσδιορίζει το όριο τάσης για τη δημιουργία καναλιού. Τα γεωμετρικά χαρακτηριστικά μετρώνται σε μικρόμετρα ( $10^{-6}$  μέτρα), ενώ το κατώφλι  $V_t$  σε Volt.

	nMOS	pMOS
i. Σύμβολα		
ii. Φυσικά σχέδια (layouts)		
iii. Συνθήκες λειτουργίας	<p>Αποκοπή: <math>V_{GS} &lt; V_t</math>  <math>I_{DS}=0</math></p> <p>Γραμμική περιοχή: <math>V_{GS} &gt; V_t</math> &amp; <math>V_{DS} &lt; V_{GS}-V_t</math>  <math>I_{DS}=\beta_n[(V_{GS}-V_t)-V_{DS}/2]*V_{DS}</math></p> <p>Περιοχή Κόρου: <math>V_{GS} &gt; V_t</math> &amp; <math>V_{DS} &gt; V_{GS}-V_t</math>  <math>I_{DS}=\beta_n/2*(V_{GS}-V_t)^2</math></p>	<p>Αποκοπή: <math> V_{GS}  &lt;  V_t </math>  <math>I_{DS}=0</math></p> <p>Γραμμική περιοχή: <math> V_{GS}  &gt;  V_t </math> &amp; <math> V_{DS}  &lt;  V_{GS}-V_t </math>  <math>I_{DS}=-\beta_p[(V_{GS}-V_t)-V_{DS}/2]*V_{DS}</math></p> <p>Περιοχή Κόρου: <math> V_{GS}  &gt;  V_t </math> &amp; <math> V_{DS}  &gt;  V_{GS}-V_t </math>  <math>I_{DS}=-\beta_p/2*(V_{GS}-V_t)^2</math></p>
iv. Καμπύλες λειτουργίας	<p>NMOS transistor, 0.25μm, <math>L_d = 0.25\mu\text{m}</math>, W/L = 1.5, <math>V_{DD} = 2.5\text{V}</math>, <math>V_T = 0.4\text{V}</math></p>	<p>PMOS transistor, 0.25μm, <math>L_d = 0.25\mu\text{m}</math>, W/L = 1.5, <math>V_{DD} = 2.5\text{V}</math>, <math>V_T = -0.4\text{V}</math></p>
v. Διακοπτική λειτουργία	<p><math>g = 0</math></p> <p>OFF</p> <p><math>g = 1</math></p> <p>ON</p>	<p>ON</p> <p>OFF</p>
vi. Χαρακτηριστικά διελεύσεως	<p><math>g = 0</math></p> <p>Input <math>g = 1</math> Output</p> <p><math>0 \rightarrow 0</math> - strong 0</p> <p><math>g = 1</math></p> <p><math>1 \rightarrow 0</math> - degraded 1</p>	<p><math>g = 0</math></p> <p>Input <math>g = 0</math> Output</p> <p><math>0 \rightarrow 0</math> - degraded 0</p> <p><math>g = 1</math></p> <p><math>1 \rightarrow 0</math> - strong 1</p>

Σχήμα 2.2: Συγκεντρωτικός Πίνακας περιγραφής των nMOS και pMOS τρανζίστορ.

Τα σύμβολα των nMOS και pMOS τρανζίστορ καθώς και οι ακροδέκτες τους φαίνονται στο Σχήμα 2.2.i. Το φυσικό σχέδιο των τρανζίστορ φαίνεται στο Σχήμα 2.2.ii έτσι όπως παρουσιάζεται στο εργαλείο σχεδιασμού Microwind. Το σημαντικό εδώ είναι ότι κάθε διασταύρωση περιοχής πολυπυριτίου (poly) – οριζόντια ράβδος με όνομα Gate – με μια περιοχή διάχυσης n ή p – περιοχή κάθετη στο Gate – δημιουργεί τρανζίστορ. Έτσι, όταν εξετάζουμε ένα φυσικό σχέδιο πχ στο Microwind, μπορούμε να καταλάβουμε την ύπαρξη τρανζίστορ εντοπίζοντας τα μέρη του σχεδίου στα οποία υπάρχει **διασταύρωση πολυπυριτίου με κάποια περιοχή διάχυσης n ή p**.

Οι συνθήκες λειτουργίας και τα αντίστοιχα ρεύματα που διαρρέουν το κανάλι φαίνονται στο Σχήμα 2.2.iii, οι δε γραφικές παραστάσεις λειτουργίας τρανζίστορ με όμοια χαρακτηριστικά (γεωμετρικές διαστάσεις και δυναμικό κατωφλίου) φαίνονται στο Σχήμα 2.2.iv. Παρατηρείστε ότι τα ρεύματα που παρέχει ένα pMOS τρανζίστορ υπολείπονται αυτών ενός nMOS τρανζίστορ κατά ένα παράγοντα 2~3. Αυτό οφείλεται στο ότι το κέρδος  $\beta$  του pMOS είναι μικρότερο κατά τον ίδιο παράγοντα 2~3 από αυτό του nMOS.

Στο Σχήμα 2.2.v παρουσιάζονται τα διακοπτικά χαρακτηριστικά των δύο ειδών τρανζίστορ. Η **συμπληρωματική συμπεριφορά** μεταξύ τους δικαιολογεί και το όνομα της λογικής Complementary MOS: το nMOS άγει όταν η πύλη του G είναι στο λογικό "1", ενώ το pMOS άγει όταν η πύλη του G είναι στο λογικό "0". Βασιζόμενοι σε αυτές τις ιδιότητες μπορούμε να κατασκευάσουμε ψηφιακά κυκλώματα οποιασδήποτε πολυπλοκότητας.

Από τα παραπάνω μπορούμε να αποκτήσουμε μια εισαγωγική αντίληψη στην έννοια της τεχνολογίας στο σχεδιασμό Ολοκληρωμένων Κυκλωμάτων (Ο.Κ.): Η **Τεχνολογία στο Σχεδιασμό VLSI** είναι ένα σύνολο γεωμετρικών κανόνων σχεδιασμού των περιοχών του φυσικού σχεδίου (layout) ενός Ο.Κ., καθώς και των ηλεκτρικών τους χαρακτηριστικών (πχ αντίσταση ανά μονάδα επιφάνειας, χωρητικότητα κλπ) τους οποίους πρέπει να ακολουθήσει ο σχεδιαστής προκειμένου να εξασφαλίσει ότι το κύκλωμα που θα σχεδιάσει θα λειτουργεί σύμφωνα με τις προδιαγραφές. Στο εργαλείο σχεδιασμού Microwind οι υποστηριζόμενες τεχνολογίες περιγράφονται σε αρχεία με προέκταση ".rul" τα οποία είναι αποθηκευμένα στο φάκελο εγκατάστασης του προγράμματος. Έτσι, πχ η τεχνολογία CMOS 0.18 $\mu$ m περιγράφεται στο αρχείο cmos018.rul, το δε 0.18 $\mu$ m εκφράζει το ελάχιστο μήκος του καναλιού  $L_{min}$ . Συνηθίζεται οι διαστάσεις των φυσικών σχεδίων να εκφράζονται με την παράμετρο  $\lambda=L_{min}/2$ .

Τέλος, στο Σχήμα 2.2.vi παρατίθενται τα χαρακτηριστικά διέλευσης των CMOS τρανζίστορ. Εκεί φαίνεται ότι όταν το nMOS άγει ( $G="1"$ ) μπορεί να μεταφέρει αναλλοίωτο το λογικό "0", αλλά το λογικό "1" μεταφέρεται υποβαθμισμένο κατά το δυναμικό κατωφλίου του. Ομοίως, το pMOS όταν άγει ( $G="0"$ ), μπορεί να μεταφέρει αναλλοίωτο το "1", αλλά το "0" υποβαθμίζεται κατά το δυναμικό κατωφλίου του.

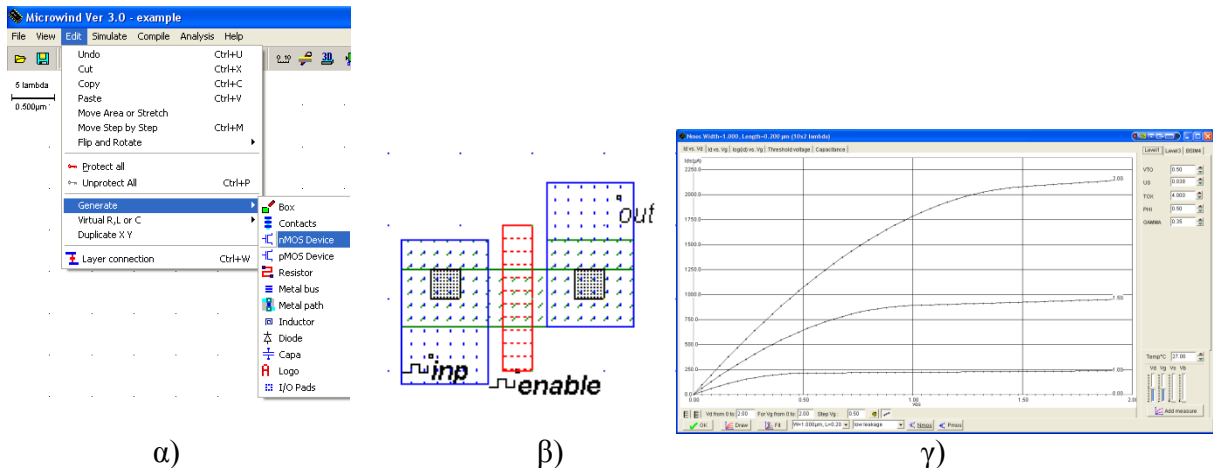
### Πρακτική διαδικασία<sup>{Bl. [2]}</sup>

1. Ο εξομοιωτής του MICROWIND. Επιλέγοντας το menu simulate από τη γραμμή των pull-down menus του MICROWIND μπορείτε να καθορίσετε τις παραμέτρους της εξομοίωσης. ΠΡΟΣΟΧΗ: Να επιλέγετε πάντα εξομοίωση με χρήση του μοντέλου BSIM4 για τα τρανζίστορ. Επιλογή από το menu Simulate -> Using model -> BSIM4(advanced)

2. Εξομοίωση ενός n-mos και ενός p-mos transistor με διαστάσεις  $W=0.36\mu\text{m}$  και  $L=0.18\mu\text{m}$  σε τεχνολογία CMOS 0.18 $\mu\text{m}$ .

Για το σχεδιασμό ενός τρανζίστορ επιλέγετε πρώτα την τεχνολογία CMOS 0.18 $\mu\text{m}$ : File -> Select Foundry -> "cmos018.rul".

Κατόπιν σχεδιάζετε το τρανζίστορ με την ευκολία αυτόματου σχεδιασμού του Microwind: Edit -> Generate -> nMOS Device.



α) Γεννήτρια αυτόματης παραγωγής ηλεκτρονικών στοιχείων, β) Το τρανζίστορ MOS, γ) Το παράθυρο εξομοίωσης τρανζίστορ.

a. Παραγωγή των χαρακτηριστικών  $I_d$  vs  $V_d$  για διαφορετικές τάσεις πύλης  $V_g$

Επιλέξτε Simulate -> MOS Characteristics και κατόπιν της προτροπής του Microwind κάντε κλικ στο τρανζίστορ που έχετε σχεδιάσει για να σας παρουσιαστεί το παράθυρο της εξομοίωσης τρανζίστορ.

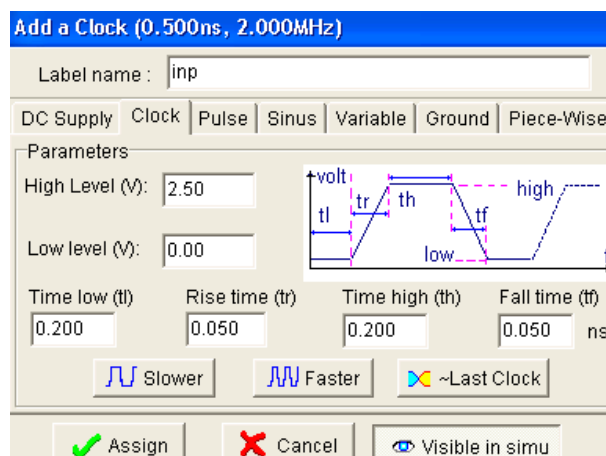
b. Σύγκριση n-mos και p-mos transistors

c. Παραμετρική ανάλυση των χαρακτηριστικών ως προς το πλάτος καναλιού του transistor

3. Μελέτη της απόκρισης των δύο τύπων transistor ως διακόπτες με είσοδο τετραγωνικό παλμό χωρίς φορτίο και με φορτίο 0.2pF.

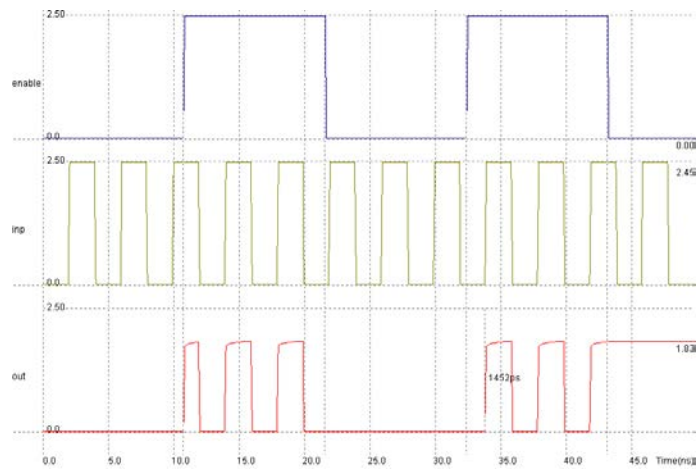
a. Σχεδιάστε ένα n-mos τρανζίστορ με διαστάσεις  $W=0.36\mu\text{m}$  και  $L=0.18\mu\text{m}$  χρήση της γεννήτριας στοιχείων.

β. Για να γίνει εξομοίωση στο πεδίο του χρόνου και να εμφανιστούν οι σχετικές κυματοφορφές πρέπει να δώσετε ονόματα στους κόμβους του κυκλώματος και να ορίσετε τις τροφοδοσίες ( $V_{dd}$  και  $V_{ss}$ ). Τα εργαλεία για την εκτέλεση των διαδικασιών αυτών βρίσκονται στην τρίτη γραμμή εργαλείων της παλέτας. Επιλέγοντας, για παράδειγμα, το add clock εμφανίζεται το παρακάτω παράθυρο που σας παρέχει τη δυνατότητα να ορίσετε το όνομα του κόμβου και να επιλέξετε τους χρόνους ανόδου και καθόδου και το εύρος των παλμών και το αν η κυματομορφή αυτή θα εμφανίζεται στην εξομοίωση

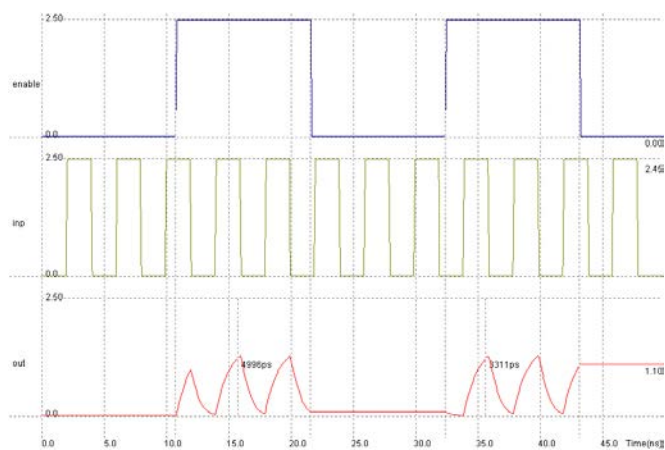


Δώστε στο drain του τρανζίστορ το όνομα inp και ένα παλμό με διάρκεια low και high 1.95nsec και χρόνο ανόδου και καθόδου 50psec, στο gate του τρανζίστορ το όνομα enable και παλμό με διάρκεια low και high 10.8 nsec και χρόνο ανόδου και καθόδου 50psec, και στο source το όνομα out. Στο Σχήμα

2.3β φαίνεται το υπό μελέτη κύκλωμα. Εκτελώντας την εξομοίωση θα πάρετε τις παρακάτω κυματομορφές:



γ. Τοποθετήστε έναν πυκνωτή 0.2pF στην έξοδο χρησιμοποιώντας πάλι την αυτόματη γεννήτρια ηλεκτρονικών στοιχείων. Εκτελώντας την εξομοίωση με πυκνωτή θα πάρετε τις παρακάτω κυματομορφές. Τι πρέπει να κάνετε για να γίνουν και πάλι τετραγωνικοί οι παλμοί στην έξοδο;



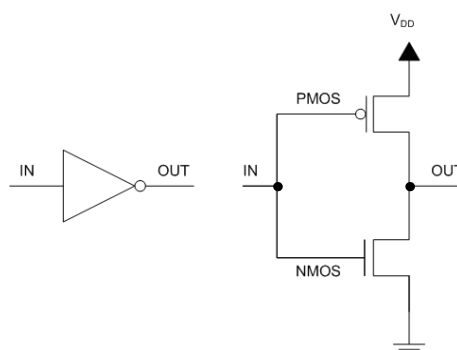
4. Επαναλάβετε τα παραπάνω βήματα 3α- 3γ για ένα p-mos τρανζίστορ με διαστάσεις  $W=1.08 \mu\text{m}$  και  $L=0.18 \mu\text{m}$ . ΠΡΟΣΟΧΗ: Για να γίνει σωστή εξομοίωση το πηγάδι τύπου n πρέπει να πολωθεί στην τάση τροφοδοσίας  $V_{DD}$ .

### 3. Κατασκευή και εξομοίωση CMOS αντιστροφέα.

**Σκοπός:** Σε αυτή την άσκηση θα κατασκευαστεί ένας στατικός CMOS αντιστροφέας και θα εξομοιωθεί προκειμένου να μελετηθεί η συμπεριφορά του.

#### Θεωρητικό υπόβαθρο

Ένας στατικός CMOS αντιστροφέας αποτελείται από ένα pMOS τρανζίστορ και ένα nMOS τρανζίστορ συνδεδεμένα με κοινή πύλη (gate) και κοινό απαγωγό (drain), όπως δείχνει και το Σχήμα 3.1. Η κοινή πύλη συνιστά την είσοδο του αντιστροφέα ενώ ο κοινός απαγωγός συνιστά την έξοδο του αντιστροφέα.




Σχήμα 3.1: Το σύμβολο και το κύκλωμα τρανζίστορ ενός στατικού CMOS αντιστροφέα.

Η πηγή (source) του pMOS συνδέεται στην τροφοδοσία ενώ η πηγή του nMOS συνδέεται στη γείωση. Αυτό εξασφαλίζει ότι ανά πάσα χρονική στιγμή το ένα από τα δύο τρανζίστορ θα λειτουργεί, το μεν nMOS παρέχοντας το λογικό "0" στην έξοδο (επειδή είναι συνδεδεμένο στη γείωση  $V_{SS}$ ), το δε pMOS παρέχοντας το λογικό "1" στην έξοδο (επειδή είναι συνδεδεμένο στην τροφοδοσία  $V_{DD}$ ).

Σημειώνεται ότι η σειρά με την οποία παρουσιάζονται τα στάδια κατασκευής του αντιστροφέα δεν ακολουθεί τη σειρά με την οποία επιστρώνονται στην πράξη οι περιοχές των υλικών, αλλά ακολουθεί μια τυποποιημένη σειρά που ακολουθείται στο σχεδιασμό CMOS κυκλωμάτων. Αυτό βέβαια δε δημιουργεί κάποιο πρόβλημα, καθώς το πρόγραμμα Microwind αναλαμβάνει να μεταγλωττίσει τις σχεδιασμένες περιοχές σε στοιχεία κυκλώματος προς εξομοίωση. Αυτό είναι και η πεμπουσία ενός σχεδιαστικού προγράμματος: να απομονώνει το σχεδιαστή από τις λεπτομέρειες της υλοποίησης.



#### Πρακτική διαδικασία

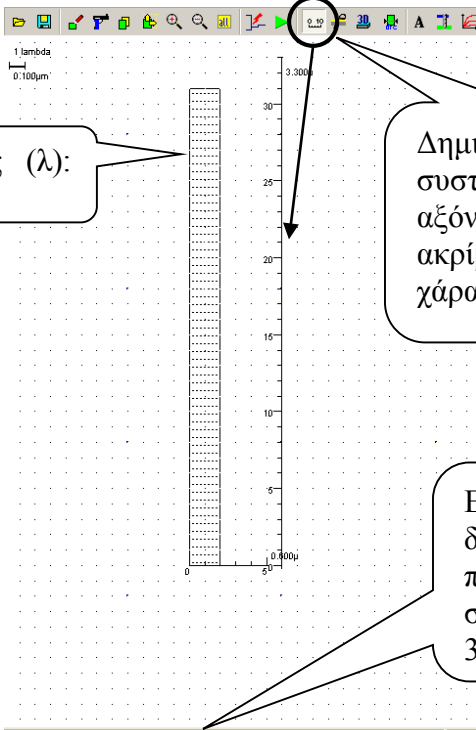
Για να σχεδιάσουμε έναν αντιστροφέα ακολουθούμε την εξής διαδικασία:

1. Εκτελούμε το πρόγραμμα Microwind,
2. Επιλέγουμε από το μενού "File" -> "Select Foundry" -> "cmos018.rul" το αρχείο της τεχνολογίας με την οποία θα δουλέψουμε,
3. Χρησιμοποιώντας τα κουμπιά μεγέθυνσης/σμίκρυνσης  στη γραμμή εργαλείων, επιτυγχάνουμε κλίμακα  $1\lambda/0.100\mu\text{m}$ , όπως φαίνεται στο αριστερό πάνω μέρος της

επιφάνειας εργασίας



- Εμφανίζουμε την παλέτα σχεδιασμού στην επιφάνεια εργασίας του προγράμματος, πατώντας το  στη γραμμή εργαλείων (1<sup>ο</sup> από δεξιά). Η παλέτα εμφανίζεται στο δεξιό τμήμα της οθόνης.
- Σχεδιάζουμε την είσοδο του αντιστροφέα επιλέγοντας το  για την ιδιότητα σχεδιασμού στη γραμμή εργαλείων (3<sup>ο</sup> από αριστερά) και χρησιμοποιώντας πολυπυρίτιο (Polysilicon) από την παλέτα σχεδιασμού.



**Διαστάσεις (λ):**  
2x28

Δημιουργία συστήματος αξόνων για ακρίβεια στη χάραξη περιοχών

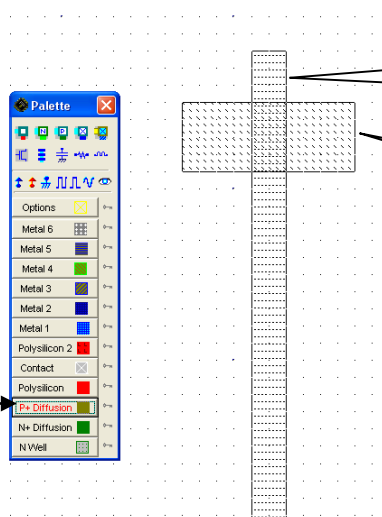
Κανόνες τεχνολογίας:  
☞ Ελάχιστη διάσταση πολυπυριτίου 2λ.

Επεξηγηματικό σχόλιο για τις διαστάσεις σε λ και μη της περιοχής πολυπυριτίου που σχεδιάστηκε.  
3λ

Options  
Metal 6  
Metal 5  
Metal 4  
Metal 3  
Metal 2  
Metal 1  
Polysilicon 2  
Contact  
Polysilicon  
P+ Diffusion  
N+ Diffusion  
NWell

dx=6 lambda (0.600um), dy=33 lambda (3.300um) (ratio 5.50) (198 lambda2, 1.98 um2) Hit: 37

- Σχεδιασμός του pMOS από τη διασταύρωση περιοχής πολυπυριτίου και διάχυσης p (P+ Diffusion):



Το πολυπυρίτιο εξέρχει της διάχυσης κατά 3λ

Διαστάσεις (λ):  
10x4

Κανόνες τεχνολογίας:  
☞ 4λ ελάχιστη διάσταση διάχυσης p και n  
☞ 3λ ελάχιστη εξοχή πολυπυριτίου από διάχυση p και n

Options  
Metal 6  
Metal 5  
Metal 4  
Metal 3  
Metal 2  
Metal 1  
Polysilicon 2  
Contact  
Polysilicon  
P+ Diffusion  
N+ Diffusion  
NWell

- Σχεδιασμός του nMOS από τη διασταύρωση περιοχής πολυπυριτίου και διάχυσης n (N+ Diffusion):

Ελάχιστη απόσταση  $p$  και  $n$  Diffusion  $12\lambda$ , προκειμένου το  $n$ Well να απέχει  $6\lambda$  από κάθε διάχυση.

Διαστάσεις και εξοχές περιοχής  $n$  Diffusion όπως και  $p$  Diffusion παραπάνω.

**Κανόνες τεχνολογίας:**  
 6λ ελάχιστη απόσταση διάχυσης  $p$  και  $n$  από  $n$ Well.

8. Επίστρωση περιοχών μετάλλου επιπέδου 1 (Metal 1) για τροφοδοσία, γείωση και έξοδο:

Οι περιοχές μετάλλων δε χρειάζεται να επικαλύπτονται με τις περιοχές διάχυσης.

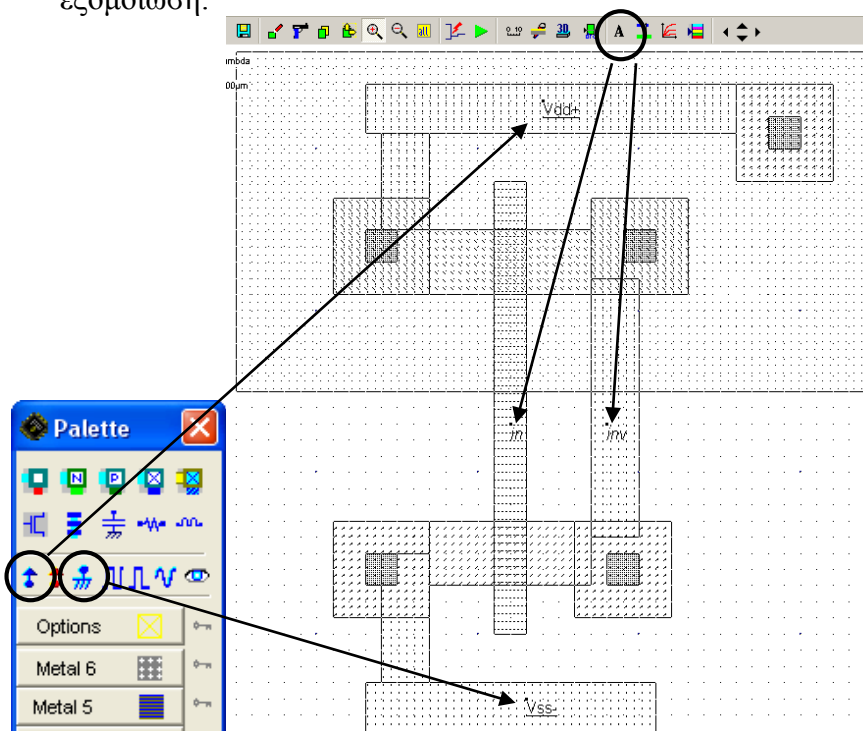
**Κανόνες τεχνολογίας:**  
 3λ ελάχιστη διάσταση περιοχών μετάλλου 1  
 4λ ελάχιστη απόσταση μεταξύ μετάλλων

9. Ηλεκτρική ένωση περιοχών διάχυσης και μετάλλων με χρήση επαφών (Contacts)

Επαφή  $n$ Diff/Metal 1 πόλωσης  $n$ Well σε τάση  $V_{DD}$

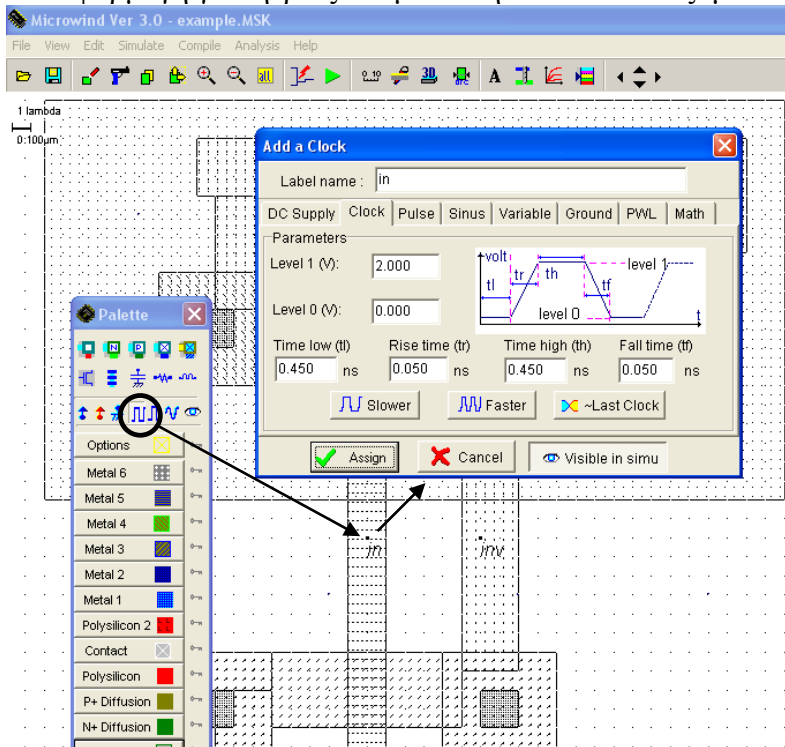
10. Επίστρωση περιοχής  $n$ Well στο  $p$ MOS τρανζίστορ και ονομασία σημάτων τροφοδοσίας, γείωσης, εισόδου και εξόδου. Φροντίζουμε ώστε στα παράθυρα ονοματοδοσίας των σημάτων εισόδου και εξόδου να πατήσουμε το κουμπί "Not in simu" ώστε να γίνει


"Visible in simu" δηλαδή το αντίστοιχο σήμα να είναι ορατό (να συμπεριληφθεί) στην εξομοίωση:

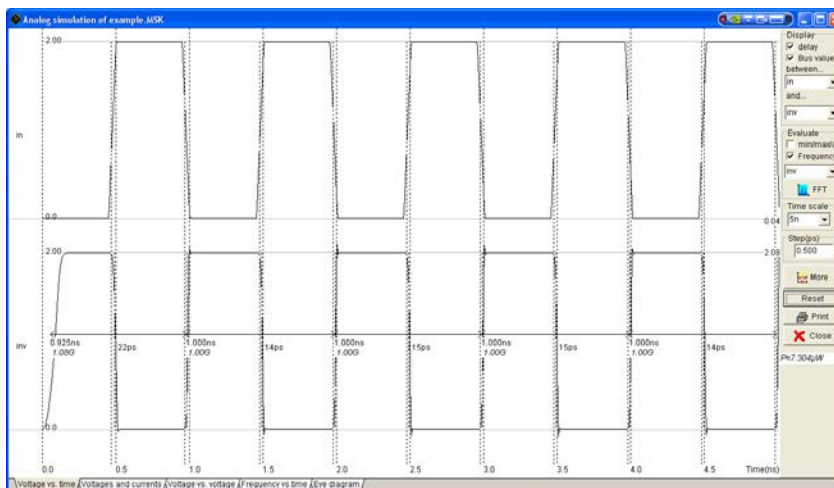


**Κανόνες τεχνολογίας:**  
 6λx6λ  
 ελάχιστες διαστάσεις επαφών  
 4λ ελάχιστη απόσταση μεταξύ μετάλλων

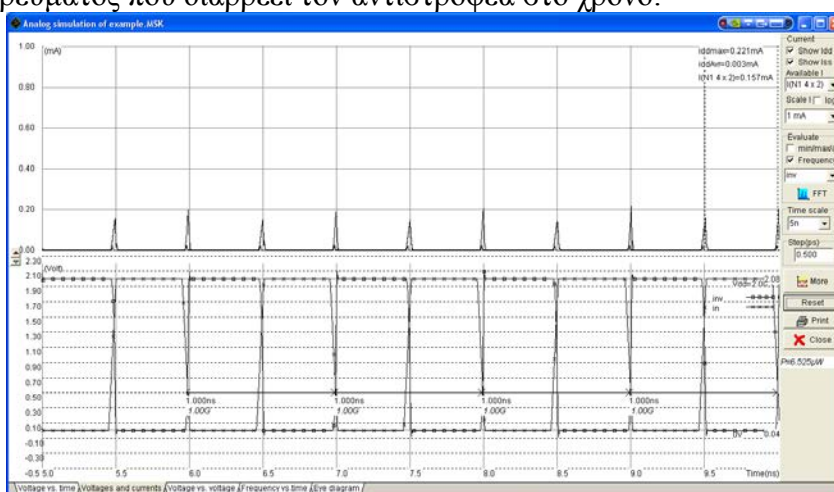
11. Εφαρμογή γεννήτριας παλμών στην είσοδο και εξομοίωση για επαλήθευση λειτουργίας:



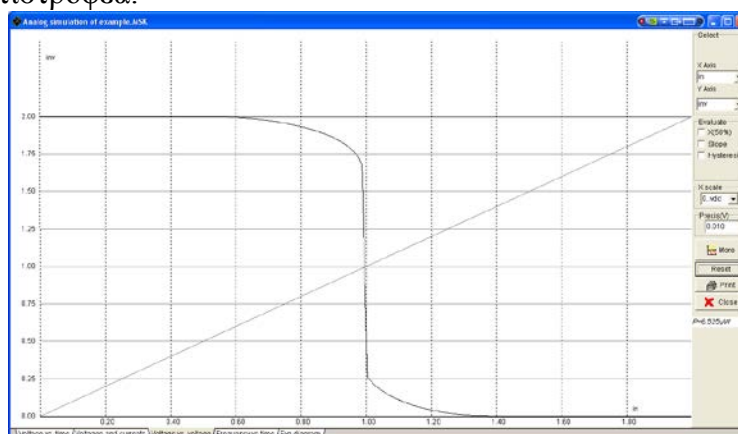
12. Πατώντας το κουμπί  στη γραμμή εργαλείων ή επιλέγοντας "Simulate" -> "Run simulation..." -> "Voltage vs Time (Default) Ctrl + S" παίρνουμε τη χρονική συμπεριφορά των ορατών σημάτων, δηλαδή της εισόδου και της εξόδου του αντιστροφέα:



Επιλέγοντας την καρτέλα "Voltages and Currents" παίρνουμε τη γραφική αναπαράσταση των μεταβολών του ρεύματος που διαρρέει τον αντιστροφέα στο χρόνο:



Επιλέγοντας την καρτέλα "Voltage vs. Voltage" βλέπουμε τη γραφική αναπαράσταση της αλλαγής της εξόδου ως προς την είσοδο, δηλαδή τη χαρακτηριστική μεταφοράς τάσης του αντιστροφέα. Η γραμμή  $V_{in}=V_{out}$  είναι επίσης ζωγραφισμένη και η τομή της με τη VTC παρέχει το σημείο αναστροφής του αντιστροφέα:

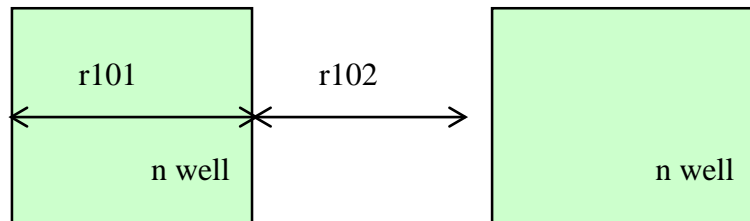


**Χρήσιμες παρατηρήσεις**


1. Η τεχνολογία CMOS 0.18μm όπως και κάθε τεχνολογία, συνίσταται από ένα σύνολο κανόνων σχεδιασμού οι οποίοι αφορούν σε ελάχιστες αποστάσεις μεταξύ περιοχών των υλικών και ελάχιστες διαστάσεις των περιοχών των υλικών, προκειμένου να εξασφαλιστεί ότι το τελικό κύκλωμα παραγωγής θα λειτουργεί χωρίς σφάλματα. Η τιμή 0.18μm είναι η διάσταση του

μικρότερου τρανζίστορ που μπορεί να κατασκευαστεί με αυτήν την τεχνολογία και αντιστοιχεί σε 2λ. Οι κανόνες είναι αποτυπωμένοι στο αρχείο cmos018.rul του Microwind, από το οποίο παραθέτουμε ένα μικρό μέρος που αφορά στους κανόνες του nWell:

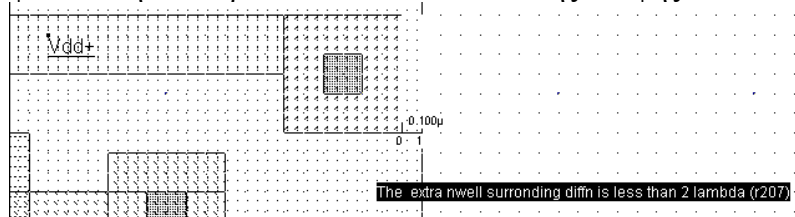
\* Well (Gds2 level 1)  
 r101 = 10 (well width)  
 r102 = 11 (well spacing)




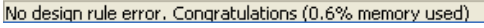
2. Σε **κάθε βήμα σχεδιασμού** πρέπει να ελέγχουμε την ικανοποίηση (τήρηση) των κανόνων σχεδιασμού της τεχνολογίας ώστε να το διορθώνουμε αμέσως, ειδάλλως ενδέχεται οι απαιτούμενες αλλαγές για την εξάλειψη της παραβίασης των κανόνων να απαιτεί δραματικές και χρονοβόρες αλλαγές στο σχέδιο. Η διαδικασία ελέγχου των κανόνων λέγεται "Design Rule Check" και το

αντίστοιχο εργαλείο στο Microwind είναι το  που βρίσκεται 5<sup>ο</sup> από δεξιά στη γραμμή εργαλείων. Εναλλακτικά, το καλούμε επιλέγοντας το μενού "Analysis" -> "Design rule checker Ctrl + D".

3. Σε περίπτωση παραβίασης κάποιου κανόνα, τότε κατά τον έλεγχο DRC ένα συνοπτικό μήνυμα εμφανίζεται στο σημείο παράβασης το οποίο ειδοποιεί το χρήστη για την παράβαση πχ εδώ φαίνεται η ανεπάρκεια του nWell έναντι της επαφής nDiff/metal1:



Σε αυτές τις περιπτώσεις χρησιμοποιούμε τα κουμπιά αλλαγών "Draw", "Delete", "Copy", και

"Stretch, Move" αντίστοιχα  για τη διόρθωση των παραβατικών περιοχών και ξανακάνουμε έλεγχο μέχρι να μην υπάρχει καμία παραβίαση. Τότε το Microwind παρέχει το μήνυμα στο κάτω μέρος της οθόνης: 

## ΑΣΚΗΣΕΙΣ

3.1 Μεταβάλλετε τα πλάτη των περιοχών διάχυσης nDiff και pDiff του αντιστροφέα, έτσι ώστε οι χρόνοι διάδοσης  $t_{prop-low} = t_{prop-high}$  (Υπόδειξη: Πρέπει  $W_p = 2-3W_n$ ).

3.2 Φτιάξτε τρεις αντιστροφείς με  $W_p = W_n = 4\lambda$  ο πρώτος,  $W_p = 16\lambda$  και  $W_n = 4\lambda$  ο δεύτερος και  $W_p = 4\lambda$  και  $W_n = 16\lambda$  ο τρίτος. Τροφοδοτείστε τους με την ίδια γεννήτρια και εξομοιώστε τους παράγοντας τις γραφικές παραστάσεις "Voltage vs Time" "Voltage vs Voltage". Τι παρατηρείτε όσον αφορά στους χρόνους διάδοσης και στις μορφές των VTC; Υπολογίστε τα σημεία αναστροφής των τριών αντιστροφέων. Ποια μεταβολή υφίσταται η VTC ανάλογα με τις διαστάσεις των τρανζίστορ; (Υπόδειξη: φτιάξτε ένα αντιστροφέα και αντιγράψτε τον άλλες δύο φορές φροντίζοντας να τοποθετηθούν **ο ένας δίπλα στον άλλο**, και μετά μεταβάλλετε τις διαστάσεις των τρανζίστορ όπως ορίζει η εκφώνηση).

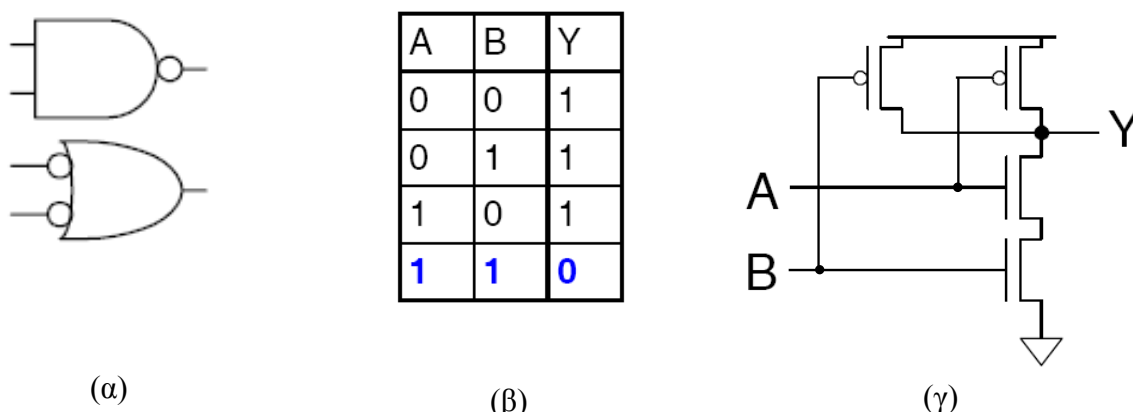
#### 4. Σχεδιασμός και εξομοίωση απλών πυλών σε CMOS λογική.

**ΣΚΟΠΟΣ:** Σε αυτή την άσκηση θα μελετηθούν, θα σχεδιαστούν και θα εξομοιωθούν οι απλές πύλες NAND και NOR σε CMOS λογική.

##### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Η σημασία απλών πυλών αρνητικής λογικής όπως NOT, NAND, και NOR στα ψηφιακά κυκλώματα σε λογική CMOS είναι μεγάλη, καθώς λόγω της φύσης των τρανζίστορ nMOS και pMOS που τα αποτελούν, η κατασκευή αυτών των πυλών είναι άμεση. Η κατασκευή απλών πυλών θετικής λογικής όπως buffers, AND, και OR προκύπτει από τις πρώτες με σύνδεση στην έξοδό τους ενός αντιστροφέα (πύλη NOT).

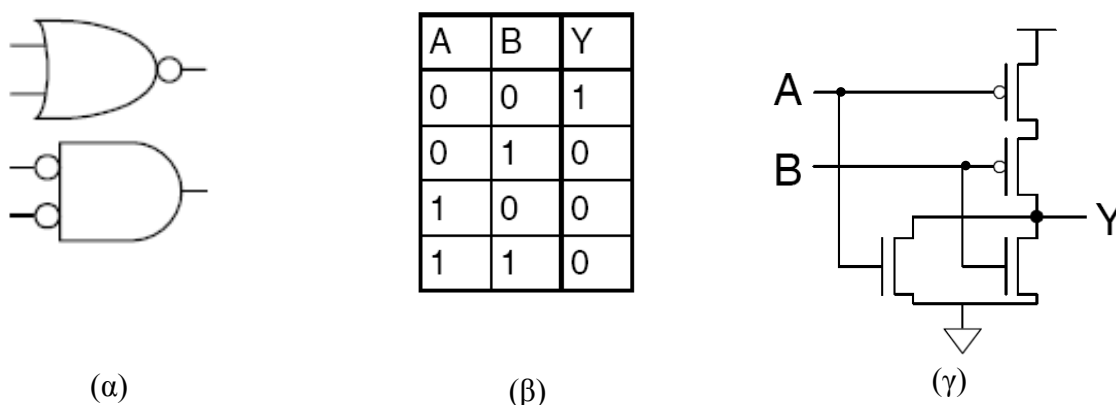
Τα σύμβολα, ο πίνακας αληθείας και το κύκλωμα MOS τρανζίστορ μιας πύλης NAND δύο εισόδων φαίνονται στο παρακάτω Σχήμα 4.1:



Σχήμα 4.1: (α) Σύμβολα, (β) πίνακας αληθείας και (γ) κύκλωμα τρανζίστορ μιας πύλης NAND-2in.

Από το Σχήμα 4.1(γ) παρατηρούμε ότι το κόστος σε τρανζίστορ για μια πύλη NAND 2 εισόδων είναι 4, δηλαδή 2 pMOS και 2 nMOS.

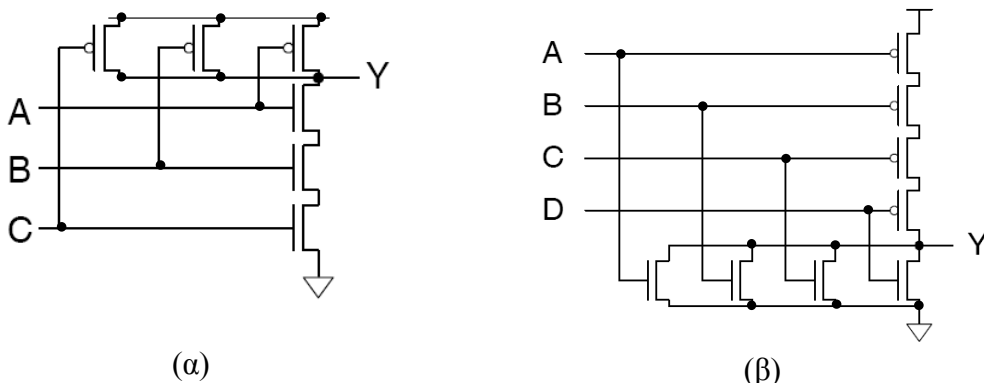
Στο Σχήμα 4.2 φαίνονται τα αντίστοιχα στοιχεία για μια πύλη NOR 2 εισόδων σχεδιασμένη σε CMOS λογική:



Σχήμα 4.2: (α) Σύμβολα, (β) πίνακας αληθείας και (γ) κύκλωμα τρανζίστορ μιας πύλης NOR-2in.

Από το Σχήμα 4.2(γ) παρατηρούμε ότι το κόστος κατασκευής μιας πύλης NOR 2 εισόδων σε CMOS λογική είναι 4 τρανζίστορ, όσο δηλαδή και για μια NAND. Επίσης, από τη σύγκριση των Σχημάτων 3.1(γ) και 3.2(γ), παρατηρούμε ότι τα δύο κυκλώματα είναι δυϊκά μεταξύ τους, καθώς η εν παραλλήλω σύνδεση pMOS τρανζίστορ στη NAND έχει γίνει εν σειρά στη NOR, ενώ η εν σειρά σύνδεση nMOS τρανζίστορ στη NAND, έχει γίνει εν παραλλήλω στη NOR.

Στο Σχήμα 4.3(α) και 3.3(β) απεικονίζονται τα κυκλώματα απλών πυλών NAND 3 εισόδων και NOR τεσσάρων εισόδων αντίστοιχα.

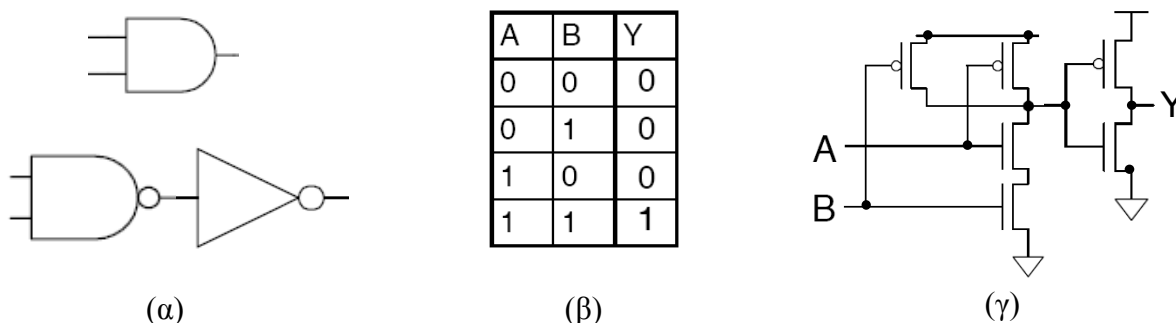


Σχήμα 4.3: (α) κύκλωμα πύλης NAND 3in, (β) κύκλωμα πύλης NOR 4in.

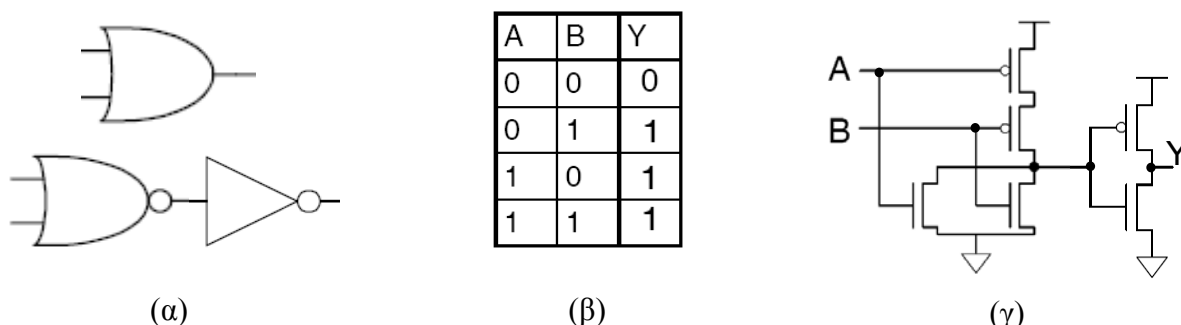
Από τα κυκλώματα των Σχημάτων 4.1-4.3 παρατηρούμε ότι το κόστος κατασκευής πυλών NAND και NOR σε CMOS λογική είναι:

$$\text{Κόστος(NAND/NOR)} = 2 * \langle \text{Αριθμός εισόδων πύλης} \rangle$$

Τέλος, για την κατασκευή πυλών AND και OR σε στατική CMOS λογική, όπως προαναφέρθηκε χρησιμοποιούνται πύλες NAND και NOR με συνδεδεμένη πύλη NOT στην έξοδό τους. Αυτά τα κυκλώματα παρουσιάζονται στα Σχήματα 4.4 και 4.5 αντίστοιχα.



Σχήμα 4.4: (α) Σύμβολα, (β) πίνακας αληθείας, και (γ) κύκλωμα μιας πύλης AND 2in σε CMOS.



Σχήμα 4.5: (α) Σύμβολα, (β) πίνακας αληθείας, και (γ) κύκλωμα μιας πύλης OR 2in σε CMOS.

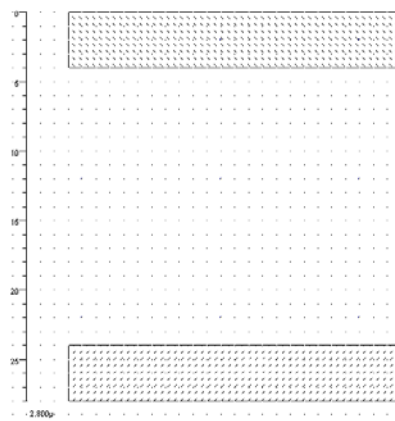
Από τα Σχήματα 4.4(γ) και 4.5(γ) παρατηρούμε ότι οι πύλες AND και OR απαιτούν 6 τρανζίστορ για να υλοποιηθούν σε στατική CMOS λογική, δηλαδή 2 παραπάνω από τις αντίστοιχες NAND NOR λόγω του αντιστροφέα. Γενικότερα επομένως, το κόστος αυτών των πυλών θα είναι:

$$\text{Κόστος(AND/OR)} = 2 * \langle \text{Αριθμός εισόδων πύλης} \rangle + 2$$

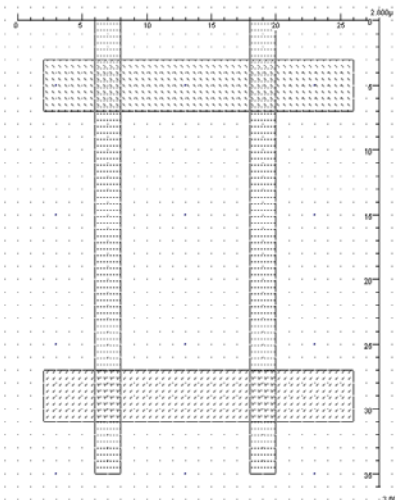
**ΠΡΑΚΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ**

Για το σχεδιασμό μιας πύλης NAND 2 εισόδων ακολουθούμε τη διαδικασία που περιγράφεται στην Άσκηση 3 για το CMOS αντιστροφέα. Η διαδικασία αυτή συνοψίζεται παρακάτω:

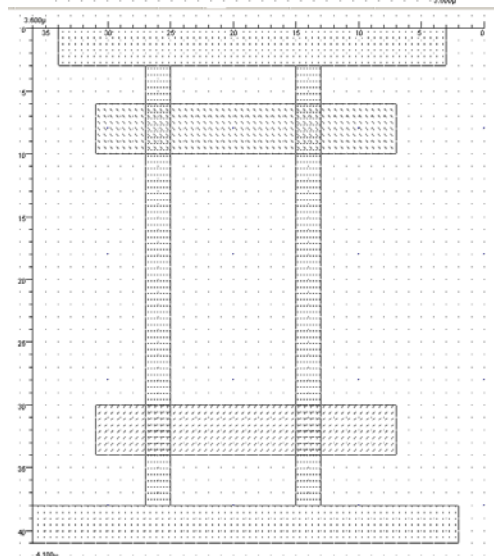
Σχήμα 4.6.1. Σχεδιασμός δύο παράλληλων περιοχών διάχυσης n και p πλάτους 4λ σε απόσταση τουλάχιστον 12λ (20λ προτείνεται):



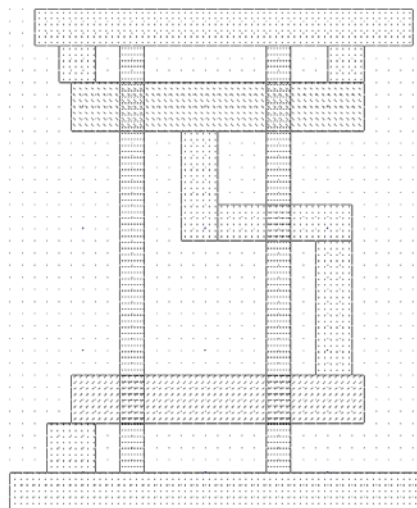
Σχήμα 4.6.2. Σχεδιασμός δύο παράλληλων περιοχών πολυπυριτίου πλάτους 2λ **κάθετα** στις διαχύσεις και σε απόσταση τουλάχιστον 6λ (10λ προτείνεται):



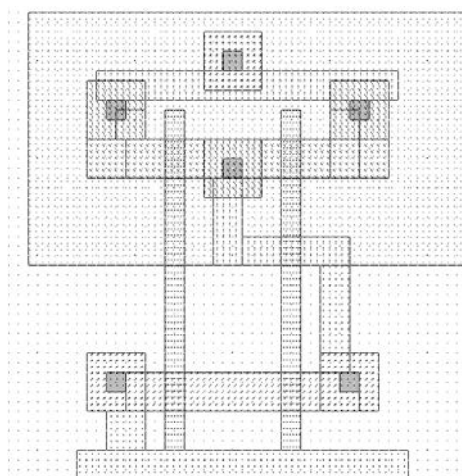
Σχήμα 4.6.3. Σχεδιασμός δύο παράλληλων περιοχών μετάλλου επιπέδου 1 (m1) εκατέρωθεν των περιοχών διάχυσης πλάτους 3λ, η μεν πάνω θα δεχτεί την τροφοδοσία η δε κάτω τη γείωση:



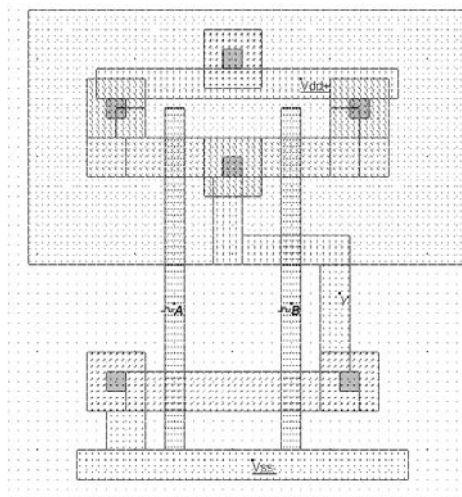
Σχήμα 4.6.4. Σχεδιασμός περιοχών μετάλλων για τις διασυνδέσεις μεταξύ των άκρων των τρανζίστορ:



Σχήμα 4.6.5. Σχεδιασμός περιοχής nWell γύρω από τα pMOS τρανζίστορ σε απόσταση 6λ από κάθε διάχυση και τοποθέτηση επαφών:



Σχήμα 4.6.6. Τοποθέτηση σημάτων τροφοδοσίας, γείωσης, γεννήτριας παλμών στις εισόδους και σημείου παρατήρησης στη έξοδο:

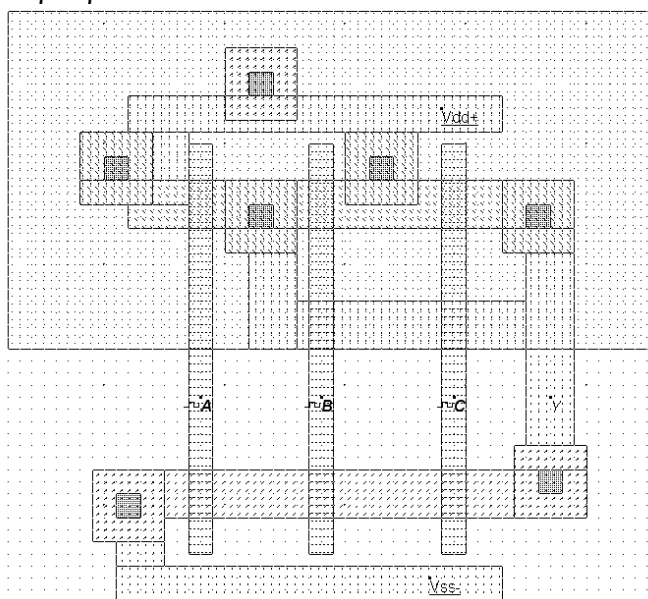


Σχήμα 4.6.7. Εξομοίωση για επαλήθευση λειτουργίας:



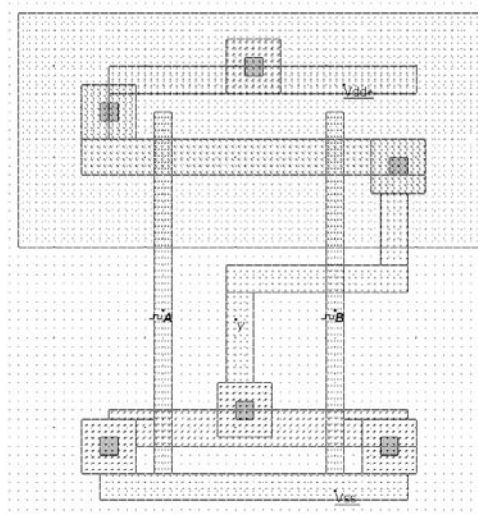
Στα παρακάτω σχήματα φαίνονται τα φυσικά σχέδια διαφόρων πυλών των οποίων η λειτουργία παρουσιάστηκε στην ενότητα "Θεωρητικό Υπόβαθρο".

Σχήμα 4.7: Το φυσικό σχέδιο (layout) μιας πύλης NAND 3 εισόδων.

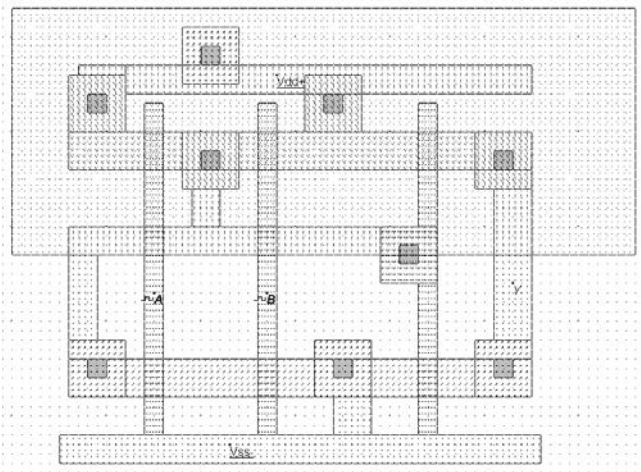


**Παρατήρηση:** Παρατηρήστε στο Σχήμα 4.7 τον τρόπο με τον οποίο σχεδιάζεται το δικτύωμα pMOS μιας πύλης NAND με παραπάνω από δύο εισόδους: οι κοινές περιοχές των τρανζίστορ συνδέονται στην τροφοδοσία και την έξοδο με εναλλασσόμενο τρόπο. Η ίδια δομή ισχύει και για τα nMOS τρανζίστορ μιας NOR πύλης με πάνω από δύο εισόδους.

Σχήμα 4.8: Το φυσικό σχέδιο (layout) μιας πύλης NOR 2 εισόδων.



Σχήμα 4.9: Το φυσικό σχέδιο (layout) μιας πύλης AND 2 εισόδων.



**Παρατήρηση:** Παρατηρείστε στο κύκλωμα του Σχήματος 4.9 ότι ο αντιστροφέας που είναι συνδεδεμένος στην έξοδο της πύλης NAND, "μοιράζεται" κοινές περιοχές διάχυσης p (που τροφοδοτείται με  $V_{DD}$ ), καθώς και διάχυσης n (που συνδέεται με  $V_{SS}$ ) με την πύλη NAND. Αυτός ο τρόπος υλοποίησης εξοικονομεί χώρο και οδηγεί σε πιο συμπυκνωμένα φυσικά σχέδια.

**Χρήσιμες Συμβουλές:** Α. Έλεγχος ικανοποίησης κανόνων σχεδιασμού της τεχνολογίας πρέπει να γίνεται τακτικότητα, σε κάθε βήμα της διαδικασίας σχεδιασμού.

Β. Το Microwind έχει τη δυνατότητα να διπλασιάζει την περίοδο της παλμοσειράς, κάθε φορά που εισάγουμε μια νέα γεννήτρια παλμών σε κάποια είσοδο του κυκλώματος.

Γ. Να αποθηκεύετε το σχέδιό σας σε αρχείο στο δίσκο και να επαναλαμβάνετε την αποθήκευση σε τακτά χρονικά διαστήματα.

## ΑΣΚΗΣΕΙΣ

4.1 Να φτιαχτεί το φυσικό σχέδιο μιας πύλης NOR τεσσάρων εισόδων και να επιβεβαιωθεί η ορθότητα λειτουργίας με εξομοίωση.

4.2 Να φτιαχτεί το φυσικό σχέδιο μιας πύλης OR τριών εισόδων και να επιβεβαιωθεί η ορθότητα λειτουργίας με εξομοίωση.

## 5. Σχεδιασμός πολυπλέκτη από 2 σε 1 με απλές πύλες.

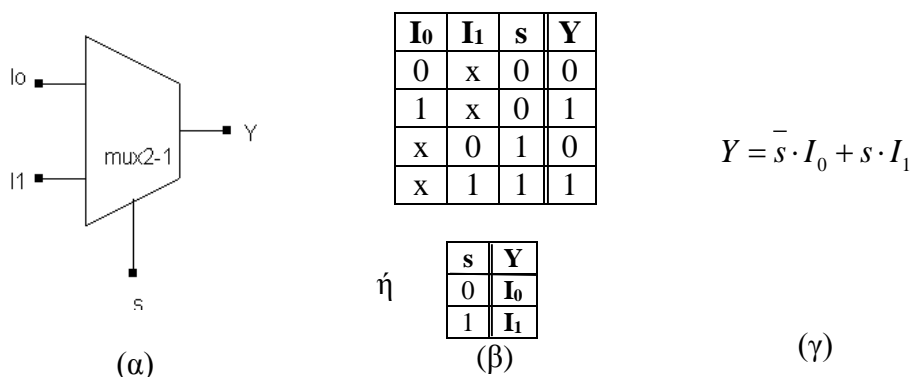
**ΣΚΟΠΟΣ:** Σε αυτή την άσκηση θα μελετηθεί, θα σχεδιαστεί και θα εξομοιωθεί ένας πολυπλέκτης 2-1, με χρήση απλών πυλών (NOT, NAND, NOR) με κριτήριο το κόστος της υλοποίησης σε αριθμό τρανζίστορ.

### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Το κόστος της υλοποίησης ενός ψηφιακού κυκλώματος ειδικά στο πεδίο των Ολοκληρωμένων Κυκλωμάτων, είναι μια από τις σημαντικότερες παραμέτρους (μαζί με την ταχύτητα και τη χαμηλή κατανάλωση) που επηρεάζουν το σχεδιασμό. Ο αριθμός των χρησιμοποιούμενων τρανζίστορ επηρεάζει την επιφάνεια του τελικού προϊόντος, την ευκολία παρασκευής του και τέλος το κόστος διάθεσής του στην αγορά.

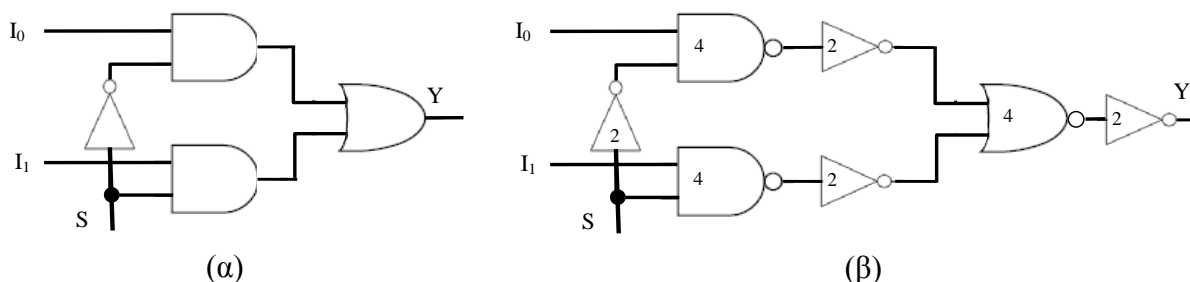
Η βέλτιστη υλοποίηση από πλευράς κόστους σε τρανζίστορ επιτυγχάνεται με χρήση διαφόρων τεχνικών. Εδώ θα δούμε μια από τις δημοφιλέστερες, η οποία στοχεύει στο να μετατρέψει την έκφραση του υπό εξέταση κυκλώματος με την εκτεταμένη χρήση κανόνων της Άλγεβρας Boole, σε έκφραση που χρησιμοποιεί μόνο απλές πύλες NOT, NAND, και NOR των οποίων η υλοποίηση σε CMOS λογική είναι φθηνή.

Ένας πολυπλέκτης 2-1 έχει ένα σήμα επιλογής  $s$  το οποίο δίνει τη δυνατότητα να επιλεγεί ένα από δύο σήματα εισόδου  $I_0$  και  $I_1$  το οποίο και θα οδηγηθεί στην έξοδο του πολυπλέκτη. Το σύμβολό του φαίνεται στο Σχήμα 5.1(α).



Σχήμα 5.1: (α) Σύμβολο, (β) πίνακας αληθείας, (γ) λογική έκφραση

Για την υλοποίηση σε CMOS λογική ενός πολυπλέκτη 2-1, ξεκινάμε από τον ορισμό της λειτουργίας του πολυπλέκτη, δηλαδή τον πίνακα αληθείας του Σχήματος 5.1(β), ο οποίος δίνει τη συνάρτηση του Σχήματος 5.1(γ) ως λογική έκφραση της λειτουργίας του πολυπλέκτη. Η απευθείας (direct) υλοποίηση του πολυπλέκτη με απλές πύλες σύμφωνα με την 4.1(γ) φαίνεται στο Σχήμα 5.2(α).



Σχήμα 5.2: (α) Απευθείας υλοποίηση του πολυπλέκτη 2-1 με απλές πύλες, (β) ανάλυση κόστους.

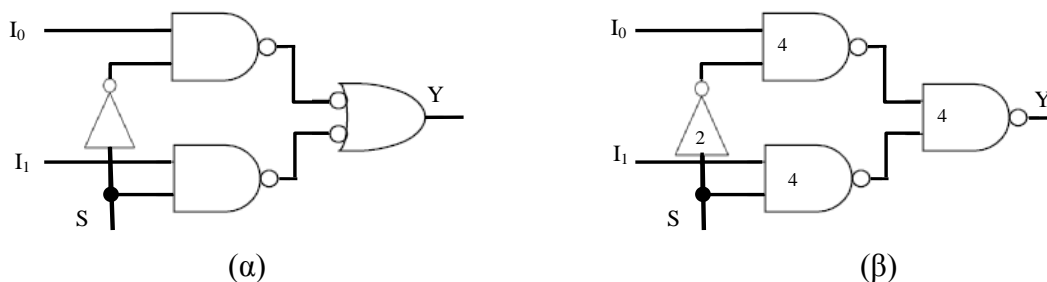
Η ανάλυση κόστους για τον πολυπλέκτη για απευθείας υλοποίηση με χρήση απλών πυλών σε CMOS λογική, φαίνεται στο Σχήμα 5.2(β) και είναι συνολικά:

$$\text{Κόστος}_1 = 4 * \text{NOT} + 2 * \text{NAND} + 1 * \text{NOR} = 4 * 2 + 2 * 4 + 1 * 4 = 20$$

Με εφαρμογή του κανόνα DeMorgan, η έκφραση 4.1(γ) μπορεί να δώσει καλλίτερη υλοποίηση:

$$\overline{Y} = \overline{s \cdot I_0 + s \cdot I_1} = \overline{s \cdot I_0} \cdot \overline{s \cdot I_1}$$

Η αντίστοιχη μέθοδος γραφικής αναπαράστασης της εφαρμογής του DeMorgan, λέγεται "εισαγωγή φυσαλίδων" (bubble insertion), αναφέρεται στην εισαγωγή ζευγών αντιστροφών ώστε να μην αλλοιωθεί η λογική λειτουργία του κυκλώματος και φαίνεται στο Σχήμα 5.3.



Σχήμα 5.3: (α) Εισαγωγή φυσαλίδων (αντιστροφείς), (β) τελικό κύκλωμα με ανάλυση κόστους.

Το τελικό αποτέλεσμα είναι ένα κύκλωμα NAND-NAND όπως φαίνεται στο Σχήμα 5.3(β), το οποίο υλοποιείται άμεσα σε CMOS λογική. Το κόστος αυτής της υλοποίησης είναι:

$$\text{Κόστος}_2 = 1 * \text{NOT} + 3 * \text{NAND} = 1 * 2 + 3 * 4 = 14$$

Παρατηρούμε ότι το κόστος<sub>2</sub> είναι αρκετά μικρότερο από το κόστος<sub>1</sub>, πράγμα που δικαιολογεί τη χρήση της μεθόδου.

## ΠΡΑΚΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Για το σχεδιασμό του πολυπλέκτη 2-1 του Σχήματος 5.3(β), ακολουθούμε την εξής διαδικασία:

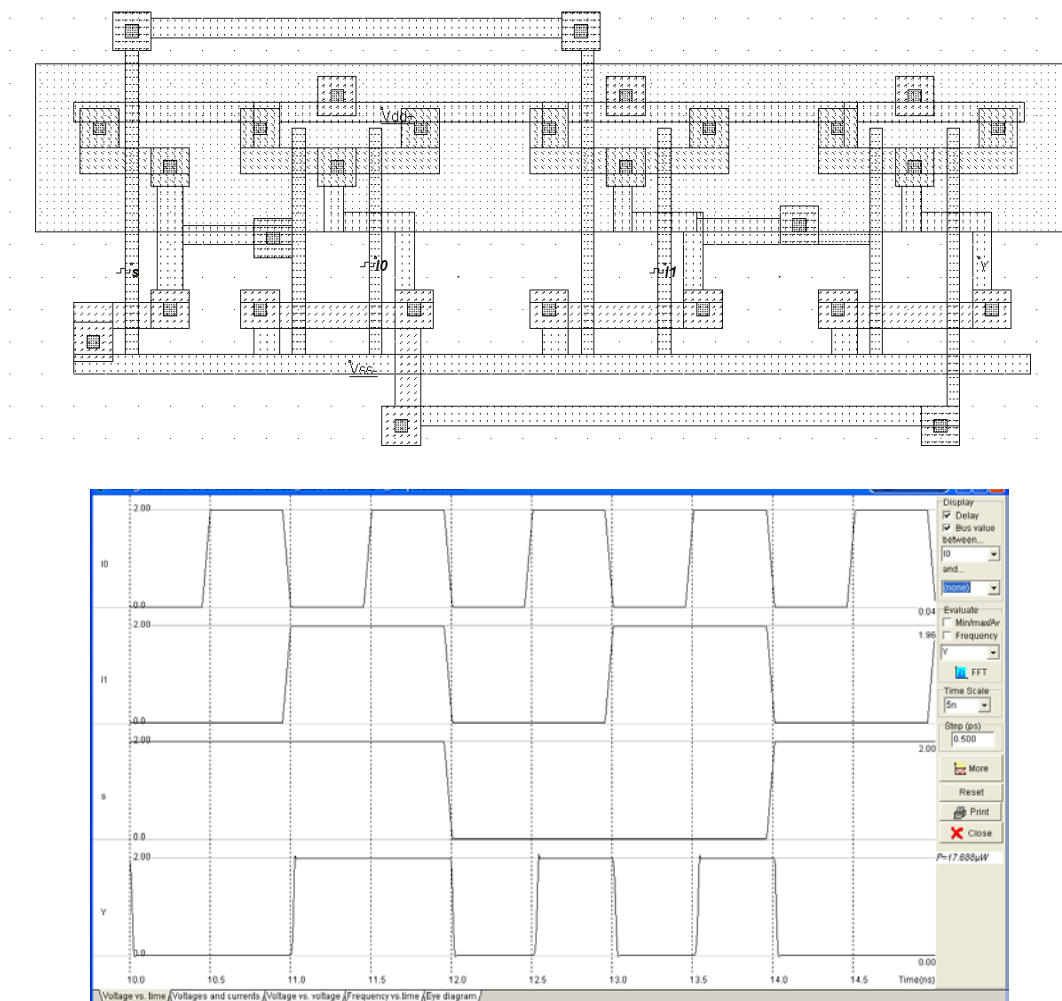
1. Σχεδιασμός ενός αντιστροφέα με τρανζίστορ ελάχιστων διαστάσεων.
2. Σχεδιασμός στην ίδια επιφάνεια σχεδιασμού (ίδιο αρχείο) μιας πύλης NAND 2 εισόδων (βλέπε Εργαστηριακή Άσκηση 3) με τρανζίστορ ελάχιστων διαστάσεων.  
ΠΡΟΣΟΧΗ: Τα δύο σχέδια πρέπει να έχουν το ίδιο ύψος σε λ, δηλαδή ίδια απόσταση μεταξύ γραμμών V<sub>DD</sub> και V<sub>SS</sub>.
3. Με το εργαλείο αντιγραφής επιλεγμένο σχηματίστε ένα πλαίσιο που να περιλαμβάνει όλο το κελί της NAND πύλης, και αντιγράψτε τη δύο φορές. Εναλλακτικά, μπορείτε να χρησιμοποιήσετε την ευκολία μαζικής αντιγραφής του Microwind, επιλέγοντας "Edit" -> "Duplicate XY", ώστε να γίνει η αντιγραφή των τριών NAND σε ένα βήμα.
4. Τοποθετήστε τις τρεις NAND και τη NOT ευθυγραμμισμένα έτσι ώστε να μπορείτε να ενώσετε τις γραμμές V<sub>DD</sub> V<sub>SS</sub> και τις περιοχές nWell (η ευθυγράμμιση των NAND γίνεται αυτόματα αν κάνετε Duplicate XY).
5. Επιλέξτε το ρόλο που θα παίζει η κάθε πύλη NAND στο κύκλωμα, αντιστοιχίζοντάς τις με τις πύλες του Σχήματος 5.3(β).
6. Ενώστε τις εξόδους των οδηγών πυλών με τις εισόδους των οδηγούμενων πυλών κατάλληλα. Υπάρχουν δύο εναλλακτικές λύσεις: α) Χρησιμοποιήστε μέταλλο επιπέδου 2 ή β) προεκτείνετε τα πολυπυρίτια των εισόδων ώστε να τελειώνουν πάνω από το μέταλλο τροφοδοσίας V<sub>DD</sub> ή κάτω από το μέταλλο γειώσεως V<sub>SS</sub>, και ενώστε τα επιμέρους πολυπυρίτια με οριζόντιες γραμμές μετάλλου 1. Χρησιμοποιήστε τις κατάλληλες επαφές metal1/metal2 και Poly/metal1 αντίστοιχα από την παλέτα σχεδιασμού.

7. Ονομάστε τις εισόδους  $I_0$ ,  $I_1$ , και  $s$ , καθώς και την έξοδο  $Y$ . Εφαρμόστε γεννήτριες παλμοσειρών στις εισόδους έτσι ώστε το σήμα επιλογής  $s$  να είναι το πιο αργά μεταβαλλόμενο (μεγαλύτερη περίοδος) από τα σήματα των εισόδων  $I_0$  και  $I_1$ . Προτείνονται τα εξής:

$$T_{I_0} < T_{I_1} < T_s \text{ με } T_{I_0} = 1\text{nsec}, T_{I_1} = 2\text{nsec}, T_s = 4\text{nsec}.$$

όπου  $T_i$  η περίοδος του αντίστοιχου σήματος  $i$ . Η συγκεκριμένη σχέση μεταξύ περιόδων επιλέχθηκε ώστε να επιτραπεί στην εκάστοτε επιλεγμένη είσοδο του πολυπλέκτη να ολοκληρώσει τουλάχιστον μια πλήρη περίοδο στην έξοδο  $Y$ .

ΠΑΡΑΤΗΡΗΣΗ: Προσοχή χρειάζεται στην ονοματοδοσία των εισόδων  $I_0$  και  $I_1$ , ώστε  $I_0$  να ονομαστεί η είσοδος της NAND η οποία στη δεύτερη είσοδό της δέχεται σήμα  $\bar{s}$  (έξοδος αντιστροφέα), ενώ  $I_1$  να ονομαστεί η είσοδος της NAND η οποία στην άλλη της είσοδο δέχεται σήμα  $s$  (είσοδος αντιστροφέα). Το Σχήμα 5.4 παρουσιάζει το layout και την εξομοίωση του πολυπλέκτη 2-1.



Σχήμα 5.4: Layout και εξομοίωση ενός πολυπλέκτη 2-1.

## ΑΣΚΗΣΕΙΣ

- 5.1 Να μελετηθεί και να σχεδιαστεί ένας πολυπλέκτης 4-1 χρησιμοποιώντας μόνο πολυπλέκτες 2-1. Πιο είναι το κόστος της υλοποίησης σε σχέση με την απευθείας υλοποίηση;
- 5.2 Να μελετηθεί και να σχεδιαστεί ένας πολυπλέκτης 2-1 των 4bit.

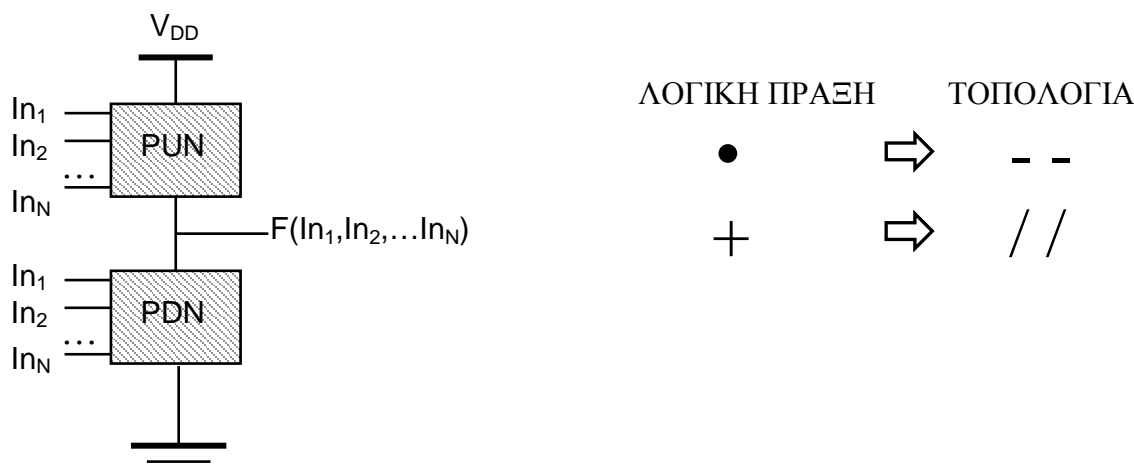
## 6. Σχεδίαση και υλοποίηση σύνθετων πυλών CMOS.

**ΣΚΟΠΟΣ:** Σε αυτή την άσκηση θα μελετηθεί ο σχεδιασμός και η υλοποίηση σύνθετων πυλών σε λογική CMOS.

### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Ο σχεδιασμός κυκλωμάτων με σύνθετες πύλες στοχεύει στην ελαχιστοποίηση της επιφάνειας υλοποίησης ορισμένων συνδυαστικών κυκλωμάτων με δύο τρόπους: ελαχιστοποίηση του αριθμού των τρανζίστορ της υλοποίησης και ελαχιστοποίηση της επιφάνειας των διαχύσεων  $n$  και  $p$  σχεδιάζοντας τις ενιαίες (ακατάτμητες).

Η γενική θεώρηση ενός κυκλώματος CMOS περιλαμβάνει ένα δικτύωμα αποτελούμενο από nMOS διακόπτες/τρανζίστορ συνδεδεμένο με τη γείωση και την έξοδο, ώστε να παρέχει τα λογικά '0' του ψηφιακού κυκλώματος (PDN: Pull-Down Network) και το δυϊκό του από pMOS διακόπτες συνδεδεμένο στην τροφοδοσία και την έξοδο, ώστε να παρέχει τα λογικά '1' (PUN: Pull-Up Network). Οι είσοδοι του κυκλώματος τροφοδοτούν και τα δύο δικτυώματα. Αυτό σχηματικά αναπαρίσταται στο Σχήμα 6.1(A).



Σχήμα 6.1: (α) Η γενική θεώρηση ενός κυκλώματος σε CMOS λογική, (β) κανόνες διασύνδεσης τρανζίστορ.

Μια σύνθετη πύλη δημιουργείται από τον συνδυασμό δομών διακοπών σε σειρά και παράλληλα για κάθε τμήμα (τόσο για το τμήμα τύπου  $n$  και για το τμήμα τύπου  $p$ ) της πύλης. Αν η συνάρτηση Boole που θέλουμε να υλοποιήσουμε με την σύνθετη πύλη έχει την μορφή του συμπληρώματος μια αλγεβρικής έκφρασης Boole (π.χ.  $F = ((A \cdot B) + C) \cdot D$ )' ή  $F = ((A + B + C) \cdot D)$ ' ή  $F = ((A \cdot B) + (C \cdot D))$ ' κ.λ.π.) τότε ξεκινάμε τη σχεδίαση από το τμήμα τύπου  $n$  χρησιμοποιώντας την  $F'$ , δηλαδή τη μη αντεστραμμένη έκφραση. Έτσι, όπου έχουμε τελεστές AND χρησιμοποιούμε συνδέσεις σε σειρά, ενώ όπου έχουμε πράξεις OR χρησιμοποιούμε παράλληλη σύνδεση. Στο Σχήμα 6.1(β) οι κανόνες αυτοί έχουν πινακοποιηθεί για εύκολη αναφορά.

Στη συνέχεια, για το τμήμα τύπου  $p$  χρησιμοποιούμε το δυϊκό της έκφρασης Boole που χρησιμοποιήσαμε στο τμήμα  $n$ , με τον ίδιο τρόπο. Υπενθυμίζουμε ότι το δυϊκό μιας έκφρασης παράγεται αντικαθιστώντας όλους τους τελεστές AND με τελεστές OR και τους OR με AND. Από την στιγμή που έχουμε σχεδιάσει τα δικτυώματα των τμημάτων  $n$  και  $p$  μένει να γίνει το τελευταίο στάδιο της σχεδίασης που είναι η σύνδεση του ενός ακροδέκτη του δικτυώματος τύπου  $n$  στο  $V_{SS}$  και του άλλου στην έξοδο (της σύνθετης πύλης) και ομοίως ο ένας ακροδέκτης του δικτυώματος τύπου  $p$  συνδέεται στο  $V_{DD}$  ενώ ο δεύτερος επίσης στην έξοδο.

Αν, σε αντίθεση με την προηγούμενη περίπτωση, η συνάρτηση Boole που θέλουμε να υλοποιήσουμε με την σύνθετη πύλη δεν έχει την μορφή του συμπληρώματος μια αλγεβρικής έκφρασης Boole, τότε χρησιμοποιούμε την ίδια την  $F$  για τη σχεδίαση του τμήματος  $n$  και το δυϊκό της  $F$  για το τμήμα  $p$ . Η σχεδίαση των τμημάτων γίνεται με τους ίδιους κανόνες καθώς και η σύνδεσή τους με τις τροφοδοσίες και την έξοδο. Η διαφορά είναι ότι σ' αυτήν την περίπτωση η έξοδος της σύνθετης πύλης θα παράγει την συνάρτηση  $F'$  αντί την  $F$ . Αυτό διορθώνεται αν προσθέσουμε έναν αντιστροφέα (inverter) στην έξοδο της σύνθετης πύλης. Η ολοκληρωμένη διαδικασία σχεδιασμού μιας σύνθετης πύλης  $F$  με χρήση μονοπατιών Euler έχει ως εξής:

▪ **Βήμα 1: Κατασκευή των κυκλωματικών διαγραμμάτων.**

Κατασκευάζονται τα κυκλωματικά διαγράμματα των  $n$  και  $p$  τρανζίστορ από τη λογική συνάρτηση  $F$  με βάση τους κανόνες του Σχήματος 6.1(β).

▪ **Βήμα 2: Εξαγωγή  $n$  και  $p$  γράφων.**

Εξαγωγή των γράφων  $n$  και  $p$  για τα αντίστοιχα κυκλώματα του Βήματος 1.

Οι κανόνες που ακολουθούνται είναι οι εξής:

α) Οι ακμές του κάθε γράφου γίνονται από τα τρανζίστορ των κυκλωμάτων και αποκτούν το όνομα της εισόδου του αντίστοιχου τρανζίστορ, και

β) Οι κορυφές του κάθε γράφου γίνονται από τις ενώσεις source/drain του κυκλώματος, δηλαδή τις ενώσεις μεταξύ των τρανζίστορ.

▪ **Βήμα 3: Καταγραφή των μονοπατιών Euler.**

Καταγραφή των μονοπατιών Euler και για τους δύο γράφους  $n$  και  $p$ .

Οι κανόνες που ακολουθούνται είναι οι εξής:

α) Το κάθε μονοπάτι πρέπει να περιλαμβάνει όλα τα τρανζίστορ.

β) Το κάθε τρανζίστορ πρέπει να εμφανίζεται μία φορά σε κάθε μονοπάτι.

γ) Η φορά κάλυψης του γράφου δεν παίζει ρόλο. Σε κάθε μονοπάτι που έχει καταγραφεί με μια συγκεκριμένη φορά κάλυψης του γράφου, αντιστοιχεί και ένα με την ανάποδη φορά (μονοπάτι-είδωλο).

**Συμπέρασμα:** ο αριθμός των μονοπατιών είναι πάντα ζυγός.

δ) Πρέπει να καταγραφούν όλα τα δυνατά μονοπάτια.

▪ **Βήμα 4: Σύγκριση ένα-προς-ένα των μονοπατιών.**

Σύγκριση ένα-προς-ένα των μονοπατιών που βρέθηκαν για το γράφο  $n$  με αυτά που βρέθηκαν για το γράφο  $p$  για την εύρεση κοινών μονοπατιών. Καταγραφή των κοινών μονοπατιών και επιλογή ενός κοινού μονοπατιού.

▪ **Βήμα 5: Τοποθέτηση πολυπυριτίων των εισόδων.**

Ο σχεδιασμός του φυσικού σχεδίου της συνάρτησης  $F$  ξεκινά με το σχεδιασμό ενιαίων περιοχών διάχυσης  $n$  και  $p$ , μετάλλων τροφοδοσίας παράλληλων στις διαχύσεις και την τοποθέτηση των ράβδων πολυπυριτίου (polysilicon) με τη σειρά που ορίζει το μονοπάτι που επιλέχθηκε στο Βήμα 4 ή την ανάστροφή του (γιατί:). Υπενθυμίζεται ότι κάθε γράμμα του κοινού μονοπατιού αντιστοιχεί σε μια είσοδο της συνάρτησης  $F$  (βλ. Βήμα 2.α)), η οποία στο φυσικό σχεδιασμό αναπαρίσταται με πολυπυρίτιο.

▪ **Βήμα 6: Σχεδιασμός διασυνδέσεων.**

Αποπεράτωση του φυσικού σχεδιασμού με το σχεδιασμό των υπόλοιπων διασυνδέσεων του κυκλώματος μεταξύ τρανζίστορ, με χρήση περιοχών μετάλλου. Σχεδιασμός και πόλωση περιοχής  $n$ Well.

⇒ **ΠΑΡΑΤΗΡΗΣΕΙΣ**

α) Σε περίπτωση που δε βρεθούν ίδια  $n$  και  $p$  μονοπάτια στο Βήμα 4, η συνάρτηση  $F$  πρέπει να «σπάσει» σε επιμέρους λογικές συναρτήσεις και να επαναληφθεί η διαδικασία για την καινούργια μορφή της  $F$ . Σε αυτήν την περίπτωση η προσπάθεια εντοπίζεται στην εύρεση των μεγαλύτερων δυνατών ενιαίων περιοχών διάχυσης για τις επιμέρους λογικές συναρτήσεις που συνθέτουν την  $F$ .

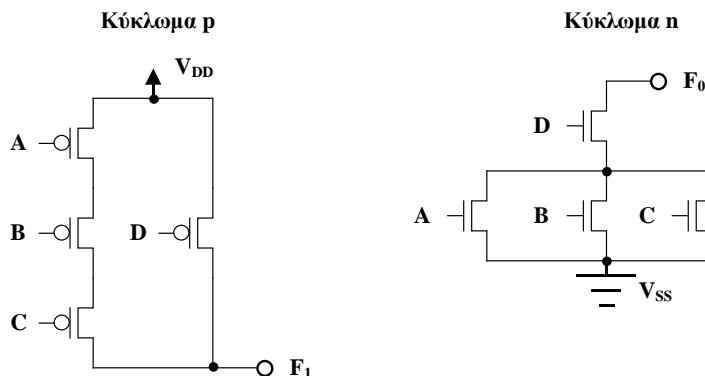
β) Τα ανάστροφης φοράς είδωλα των μονοπατιών υπονοούνται και η καταγραφή τους δεν είναι απαραίτητη, παρά μόνο για περίπλοκους γράφους. Π.χ. αν στο  $n$  γράφο μιας συνάρτησης τεσσάρων εισόδων υπήρχε το

μονοπάτι BCAD, τότε το ανάστροφης φοράς μονοπάτι DACB υπονοείται. Έτσι, αν στον  $p$  γράφο της ίδιας συνάρτησης υπήρχε το μονοπάτι DACB, τότε το κοινό μονοπάτι μεταξύ των δύο γράφων θα ήταν είτε το BCAD είτε το DACB.

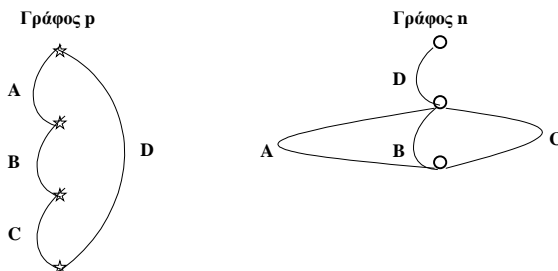
**ΠΡΑΚΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ**

Να φτιαχτεί το φυσικό σχέδιο της λογικής συνάρτησης  $f(A,B,C,D) = (A + B + C) \bullet D$

**Βήμα 1: Κατασκευή των κυκλωματικών διαγραμμάτων.**



▪ **Βήμα 2: Εξαγωγή n και p γράφων.**



▪ **Βήμα 3: Καταγραφή των μονοπατιών Euler.**

**Μονοπάτια p** (Μονοπάτια-είδωλα)

- ABCD (DCBA)
- BCDA (ADCB)
- CDAB (BADC)
- DABC (CBAD)

**Μονοπάτια n** (Μονοπάτια-είδωλα)

- DABC (CBAD)
- DACB (BCAD)
- DBCA (ACBD)
- DBAC (CABD)
- DCBA (ABCD)
- DCAB (BACD)

▪ **Βήμα 4: Σύγκριση ένα-προς-ένα των μονοπατιών.**

**Μονοπάτια p** (Μονοπάτια-είδωλα)

- ABCD (DCBA)
- ~~BCDA (ADCB)~~
- ~~CDAB (BADC)~~
- DABC (CBAD)

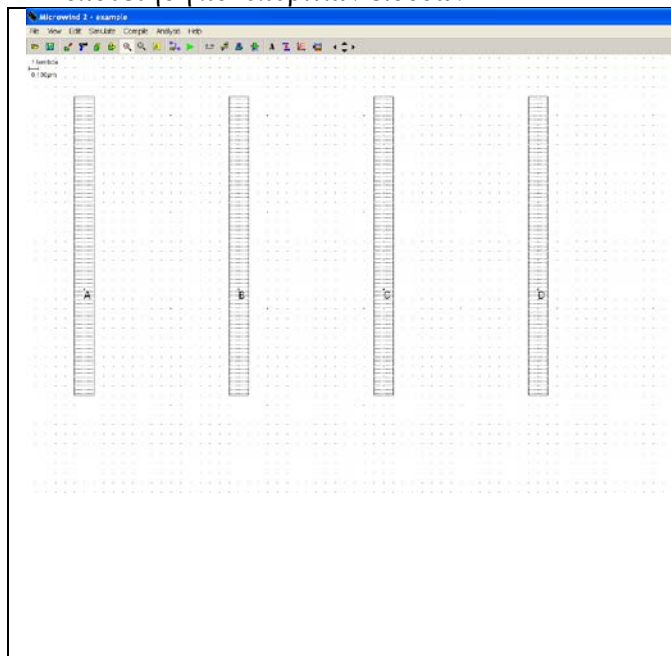
**Μονοπάτια n** (Μονοπάτια-είδωλα)

- DABC (CBAD)
- ~~DACB (BCAD)~~
- ~~DBCA (ACBD)~~
- ~~DBAC (CABD)~~
- DCBA (ABCD)
- ~~DCAB (BACD)~~

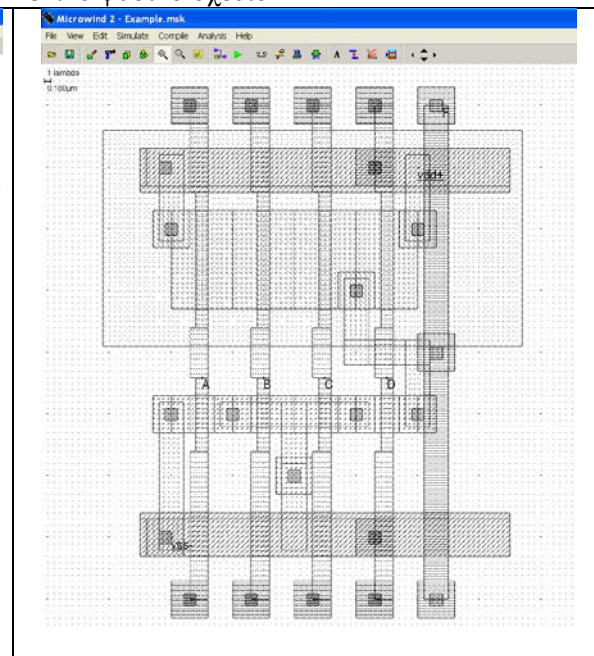
Επιλογή του μονοπατιού ABCD.

▪ **Βήματα 5 και 6: Τοποθέτηση πολυπυρτιών των εισόδων και λοιπών διασυνδέσεων.**

Τοποθέτηση πολυπυρτιών εισόδων



Τελικό φυσικό σχέδιο



**ΑΣΚΗΣΕΙΣ**

6.1 Να σχεδιαστούν σύνθετες πύλες CMOS για κάθε μία από τις παρακάτω συναρτήσεις:

$F1 = (((A \cdot B) + C) \cdot D)'$

$F2 = ((A \cdot B) + (C \cdot D))'$

$F3 = (A + B) + (C \cdot D)$

$F4 = A + B + C$

$F5 = A \cdot B \cdot C$

$F6 = (A \cdot B) + (C \cdot D)$

Ακολούθως, να υλοποιηθούν στο MicroWind και να επαληθευτεί η λειτουργία τους.

## 7. Σχεδίαση και υλοποίηση της πύλης XOR.

**ΣΚΟΠΟΣ:** Σε αυτή την άσκηση θα μελετηθεί ο σχεδιασμός και η υλοποίηση της XOR πύλης με χρήση μεθοδολογίας σύνθετων πυλών σε λογική CMOS και θα γίνει σύγκριση με άλλες σχεδιάσεις-υλοποιήσεις.

### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Η XOR πύλη είναι μια σημαντική πύλη που απαντάται σε πολλά ψηφιακά κυκλώματα και υλοποιεί τη λογική συνάρτηση "Αποκλειστικό ή". Γι' αυτό και θα μελετήσουμε την υλοποίησή της με διάφορους τρόπους, αναζητώντας τον πιο αποτελεσματικό από πλευράς κόστους σε τρανζίστορ. Ο πίνακας αληθείας μιας XOR πύλης φαίνεται στο Σχήμα 7.1(α), ενώ το σχηματικό της σύμβολο στο Σχήμα 7.1(β).

a	b	$F = a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

(α)



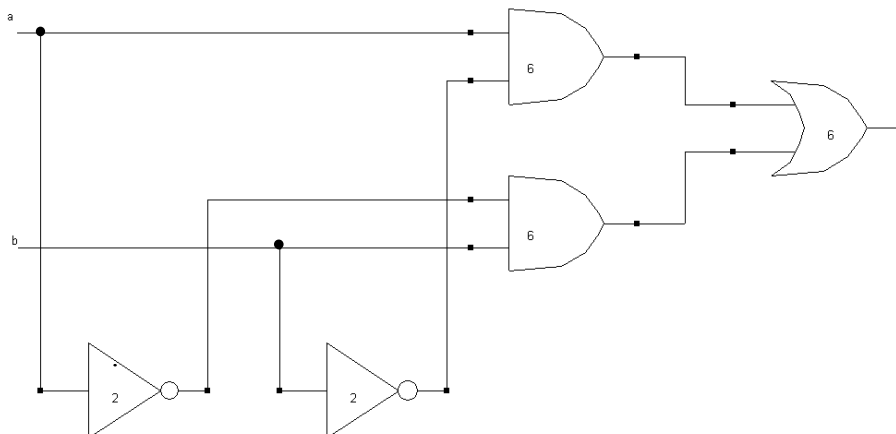
(β)

Σχήμα 7.1: (α) Πίνακας αληθείας XOR πύλης (β) Σχηματικό σύμβολο XOR πύλης.

Η λογική συνάρτηση που περιγράφει τη λειτουργία της πύλης είναι ως εξής:

$$F = a \oplus b = a \cdot \bar{b} + \bar{a} \cdot b$$

Η σχέση αυτή αν υλοποιηθεί απευθείας με απλές πύλες CMOS λογικής, θα δώσει το κύκλωμα του Σχήματος 7.2



Σχήμα 7.2: Απευθείας υλοποίηση της XOR πύλης με απλές πύλες συμπληρωματικής λογικής.

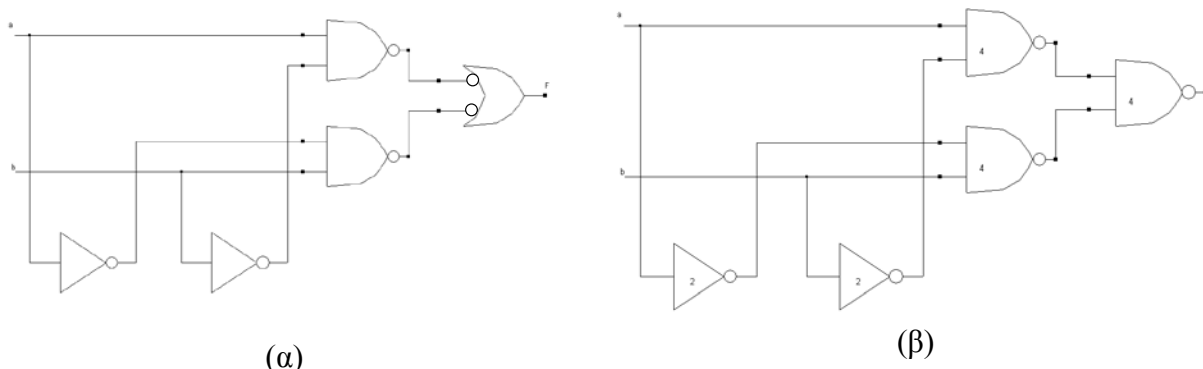
Από το Σχήμα 7.2 μπορούμε να υπολογίσουμε το κόστος σε τρανζίστορ αυτής της υλοποίησης:

$$\text{Κόστος} = 2 \cdot \text{AND} + 1 \cdot \text{OR} + 2 \cdot \text{NOT} = 2 \cdot 6 + 1 \cdot 6 + 2 \cdot 2 = 22 \text{ τρανζίστορ}$$

Με εφαρμογή του κανόνα DeMorgan μπορούμε να πάρουμε μια πιο καλή υλοποίηση:

$$F = a \oplus b = a \cdot \bar{b} + \bar{a} \cdot b \Rightarrow \bar{F} = \overline{a \cdot \bar{b} + \bar{a} \cdot b} = (\overline{a \cdot \bar{b}}) \cdot (\overline{\bar{a} \cdot b})$$

Η γραφική αναπαράσταση της εφαρμογής του DeMorgan φαίνεται στο Σχήμα 7.3 ως "εισαγωγή φουσαλίδων".



Σχήμα 7.3: (α) Υλοποίηση με εισαγωγή φυσαλίδων της XOR πύλης (β) τελικό κύκλωμα με NAND-NAND σύνδεση.

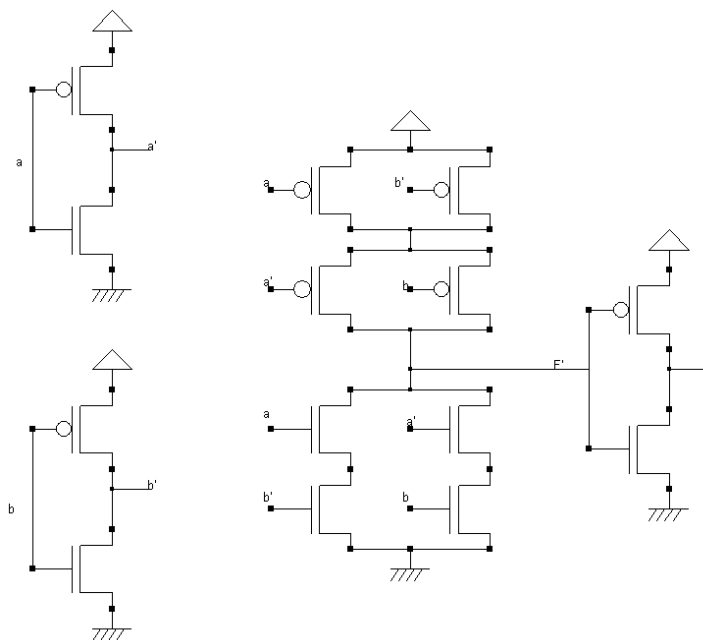
Το κόστος της υλοποίησης αυτής είναι:

$$\text{Κόστος} = 3 \cdot \text{NAND} + 2 \cdot \text{NOT} = 3 \cdot 4 + 2 \cdot 2 = 16 \text{ τρανζίστορ}$$

Μολονότι η δεύτερη υλοποίηση είναι αρκετά καλύτερη από την πρώτη, μπορούμε με τη χρήση μεθοδολογίας σύνθετων πυλών να επιτύχουμε ακόμα μεγαλύτερη οικονομία σε τρανζίστορ. Πράγματι, παίρνοντας το συμπλήρωμα της λογικής έκφρασης της XOR πύλης, καταλήγουμε στην παρακάτω μορφή:

$$F = a \oplus b = a \cdot \bar{b} + \bar{a} \cdot b \Rightarrow \bar{F} = \overline{a \cdot \bar{b} + \bar{a} \cdot b}$$

Σε αυτήν την περίπτωση, το κύκλωμα που θα υλοποιεί την XOR πύλη θα είναι όπως αυτό του Σχήματος 7.4.



Σχήμα 7.4: Το κύκλωμα σύνθετης πύλης της XOR πύλης με τους σχετικούς αντιστροφείς.

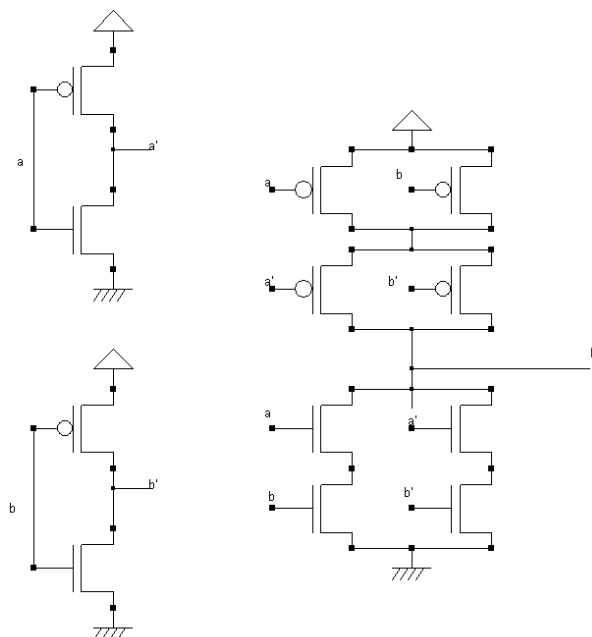
Από την ανάλυση του κυκλώματος του Σχήματος 7.4, υπολογίζουμε το κόστος ως εξής:

$$\text{Κόστος} = 3 \cdot \text{NOT} + \langle \text{Σύνθετη πύλη} \rangle = 3 \cdot 2 + 8 = 14 \text{ τρανζίστορ}$$

Με μια πιο προσεκτική μελέτη της XOR πύλης, μπορούμε να εξαλείψουμε τον αντιστροφέα της εξόδου και να πάρουμε ακόμα οικονομικότερη υλοποίηση. Ξεκινώντας από την XNOR πύλη έχουμε:

$$Y = a \otimes b = a \cdot b + \bar{a} \cdot \bar{b} \Rightarrow \bar{Y} = F = \overline{a \cdot b + \bar{a} \cdot \bar{b}}$$

Παρατηρούμε ότι η τελευταία έκφραση που προκύπτει από την αντιστροφή της XNOR πύλης δίνει την XOR πύλη, χωρίς να χρειάζεται αντιστροφέας στην έξοδο της σύνθετης πύλης. Πράγματι, το κύκλωμα τρανζίστορ του Σχήματος 7.5 προκύπτει από την υλοποίηση της τελευταίας έκφρασης της XOR πύλης.



Σχήμα 7.5: Το κύκλωμα τρανζίστορ της σύνθετης πύλης XOR από την αντιστροφή της XNOR.

Το κόστος της τελευταίας υλοποίησης όπως προκύπτει από το Σχήμα 7.5, είναι:

$$\text{Κόστος}_4 = 2 * \text{NOT} + \langle \text{Σύνθετη πύλη} \rangle = 2 * 2 + 8 = 12 \text{ τρανζίστορ}$$

Με μια απλή σύγκριση βλέπουμε ότι το Κόστος<sub>4</sub> είναι σχεδόν το μισό από το Κόστος<sub>1</sub>.

**Παρατήρηση:** Από τη σύγκριση των Σχημάτων 7.4 και 7.5 φαίνεται ότι από πλευράς τοπολογίας (διασύνδεση τρανζίστορ) οι σύνθετες πύλες είναι πανομοιότυπες. Αυτό που τις διαχωρίζει είναι τα σήματα που οδηγούν τα πολυπυρίτια των διασυνδεδεμένων τρανζίστορ.

### ΠΡΑΚΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Για το σχεδιασμό του φυσικού σχεδίου της XOR πύλης, ακολουθούμε τη διαδικασία που παρουσιάστηκε στην άσκηση 5 για τις σύνθετες πύλες και τα μονοπάτια Euler:

Βήμα 1: Κατασκευή κυκλωματικών διαγραμμάτων.

Ξεκινώντας από την τελευταία έκφραση της XOR πύλης:

$$Y = a \otimes b = a \bullet b + \bar{a} \bullet \bar{b} \Rightarrow \bar{Y} = F = \overline{a \bullet b + \bar{a} \bullet \bar{b}}$$

παίρνουμε τις λογικές εκφράσεις για τα n και p δικτυώματα:

$$n: a \bullet b + \bar{a} \bullet \bar{b}$$

⇓ (Δυϊκό)

$$p: (a + b) \bullet (\bar{a} + \bar{b})$$

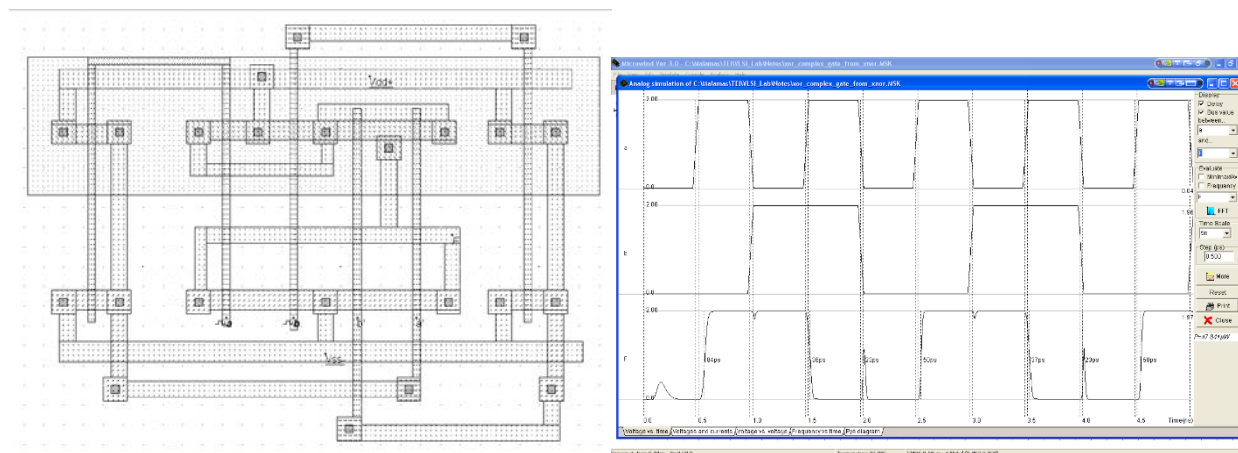
Εφαρμόζοντας τους κανόνες που μετατρέπουν τις λογικές πράξεις AND και OR σε εν σειρά και παράλληλο συνδυασμό τρανζίστορ αντίστοιχα, παίρνουμε το κύκλωμα του Σχήματος 7.5.

Βήμα 2: Εξαγωγή p και n γράφων της σύνθετης πύλης.



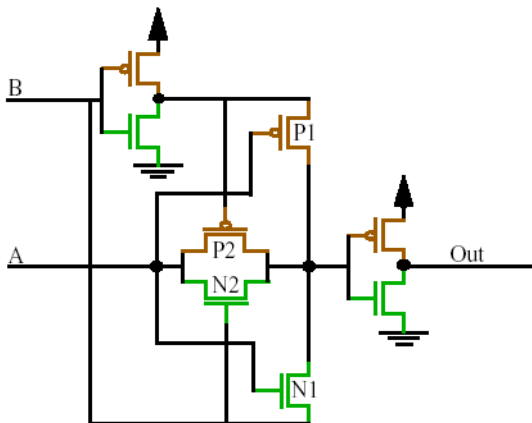
Βήμα 3 και 4: Καταγραφή μονοπατιών Euler και εύρεση κοινού μονοπατιού n, p γράφων. Ένα κοινό μονοπάτι που μπορεί να βρεθεί μεταξύ των δύο γράφων είναι το  $abb'a'$ .

Βήμα 5 και 6: Σχεδιασμός πολυπυριτίων εισόδων και υπόλοιπου φυσικού σχεδίου με βάση το μονοπάτι Euler που βρέθηκε.



**ΑΣΚΗΣΕΙΣ**

- 7.1 Να σχεδιαστεί και να εξομοιωθεί η XOR πύλη τριών εισόδων  $F = a \oplus b \oplus c$ .
- 7.2 Να σχεδιαστεί και να εξομοιωθεί το παρακάτω κύκλωμα. Τι λογική λειτουργία επιτελεί;



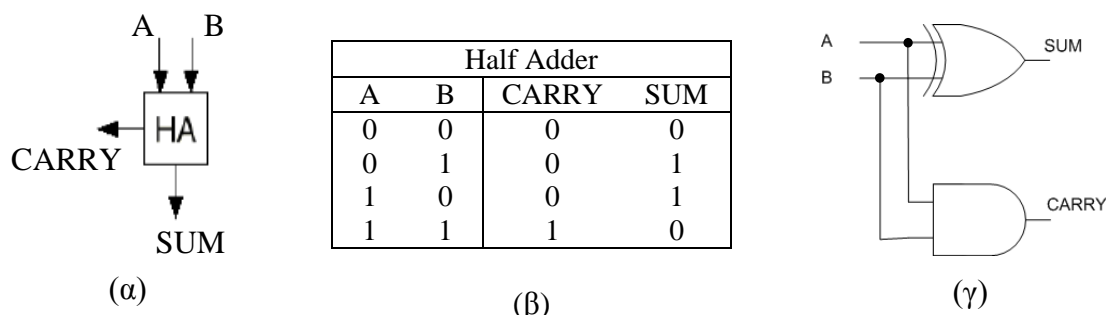
## 8. Σχεδίαση και υλοποίηση του ημι-αθροιστή και του πλήρους αθροιστή.

**ΣΚΟΠΟΣ:** Η άσκηση αυτή σκοπό έχει να εισάγει στη μελέτη της λειτουργίας του ημιαθροιστή (Half-Adder) και του πλήρους αθροιστή (Full-Adder), καθώς και στην υλοποίηση των αριθμητικών αυτών κυκλωμάτων σε επίπεδο φυσικού σχεδίου.

### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Η πρόσθεση αποτελεί τη βάση για πολλές λειτουργίες, από τη μέτρηση και τον πολλαπλασιασμό, έως και το φιλτράρισμα. Ως αποτέλεσμα, τα κυκλώματα αθροιστών που προσθέτουν δύο δυαδικούς αριθμούς είναι αντικείμενο μεγάλης σημασίας για τους σχεδιαστές ψηφιακών συστημάτων. Μια ευρεία ποικιλία από υλοποιήσεις αθροιστών είναι διαθέσιμη για την εξυπηρέτηση διαφορετικών απαιτήσεων ταχύτητας, κόστους και κατανάλωσης.

Οι πίνακες αλήθειας και τα κυκλωματικά διαγράμματα του ημιαθροιστή και του πλήρους αθροιστή δίνονται στα Σχήματα 8.1 και 8.2 αντίστοιχα. Στις υλοποιήσεις αυτές γίνεται μια προσπάθεια ώστε να χρησιμοποιηθεί ο μικρότερος αριθμός λογικών πυλών για την παραγωγή των λογικών συναρτήσεων του ημιαθροιστή (HA) και του πλήρους αθροιστή (ΠΑ).



Σχήμα 8.1: (α) Σύμβολο, (β) πίνακας αληθείας και (γ) απ' ευθείας υλοποίηση ενός ημιαθροιστή.

Από τον πίνακα αληθείας του ημιαθροιστή του Σχήματος 8.1(β), παίρνουμε τη συνάρτηση που περιγράφει τη λειτουργία του κυκλώματος αυτού:

$$\begin{aligned} SUM &= A \cdot \bar{B} + \bar{A} \cdot B = A \oplus B \\ CARRY &= A \cdot B \end{aligned}$$

για το ημιάθροισμα και για το κρατούμενο αντίστοιχα.

Μιλώντας ειδικά για το ημιάθροισμα, παρατηρούμε ότι η λογική συνάρτηση που το περιγράφει είναι μια πύλη XOR. Στην προηγούμενη Εργαστηριακή Άσκηση 6 είδαμε ένα πλήθος δυνατοτήτων για την υλοποίηση της πύλης αυτής, κάθε μια με το δικό της κόστος υλοποίησης. Λαμβάνοντας υπόψιν την τελευταία υλοποίηση με τη σύνθετη πύλη όπως εξήχθη από την πύλη XNOR, παίρνουμε την παρακάτω έκφραση για το ημιάθροισμα:

$$SUM = \overline{A \cdot B + \bar{A} \cdot \bar{B}}$$

οπότε η υλοποίηση του ημιαθροιστή με σύνθετη πύλη για το ημιάθροισμα δίνει το εξής κόστος:

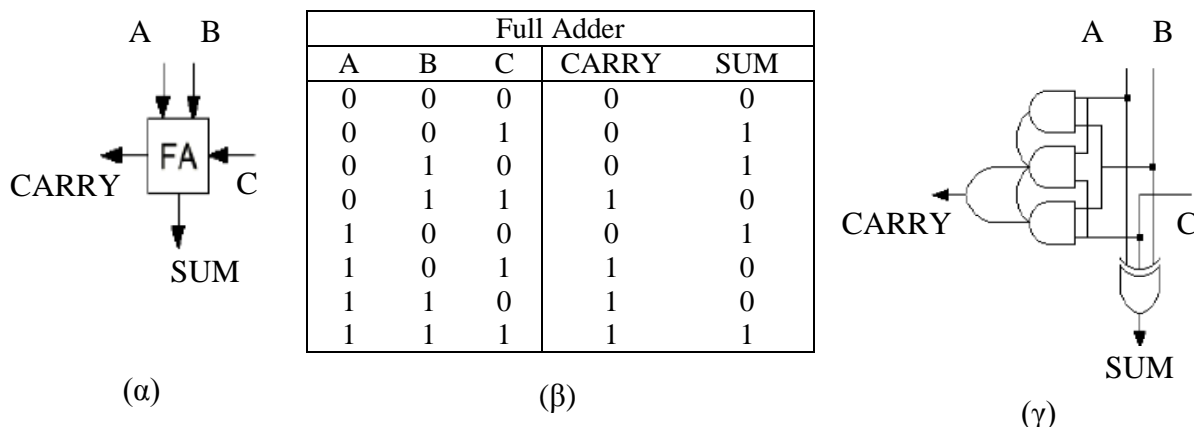
$$\text{Κόστος Ημιαθροιστή} = \langle \text{SUM} \rangle + \langle \text{CARRY} \rangle = 12 + 1 \cdot \text{AND} = 12 + 6 = 18 \text{ τρανζίστορ}$$

Περαιτέρω μείωση μπορεί να επιτευχθεί αν το κρατούμενο γραφτεί συναρτήσει των συμπληρωματικών εισόδων A και B, καθώς οι απαιτούμενοι αντιστροφείς χρησιμοποιούνται ήδη για την παραγωγή του αθροίσματος (βλέπε την αμέσως προηγούμενη συνάρτηση SUM παραπάνω). Έτσι, έχουμε:

$$CARRY = A \cdot B = \overline{\overline{A + B}}$$

Οπότε, το συνολικό κόστος του ημιαθροιστή θα είναι:

$$\text{Κόστος Ημιαθροιστή} = \langle \text{SUM} \rangle + \langle \text{CARRY} \rangle = 12 + 1 \cdot \text{NOR} = 12 + 4 = 16 \text{ τρανζίστορ}$$



Σχήμα 8.2: (α) Σύμβολο, (β) πίνακας αληθείας και (γ) κύκλωμα απλών πυλών ενός πλήρους αθροιστή.

Όσον αφορά στον πλήρη αθροιστή, υπάρχει ένα πλήθος δυνατοτήτων για την υλοποίησή του και στη συνέχεια θα αναλύσουμε μερικές από αυτές, τις πιο χαρακτηριστικές.

Από τον πίνακα αληθείας του Σχήματος 8.2(β), παίρνουμε τις παρακάτω εκφράσεις για το πλήρες άθροισμα και το κρατούμενο αντίστοιχα:

$$SUM = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot C = (\overline{A} \cdot \overline{B} + A \cdot B) \cdot C + (\overline{A} \cdot B + A \cdot \overline{B}) \cdot \overline{C} =$$

$$\overline{(\overline{A} \cdot B + A \cdot \overline{B})} \cdot C + (\overline{A} \cdot B + A \cdot \overline{B}) \cdot \overline{C} \Rightarrow SUM = A \oplus B \oplus C$$

$$CARRY = A \cdot B + B \cdot C + A \cdot C \tag{8.1}$$

Υποθέτοντας απ' ευθείας υλοποίηση του Σχήματος 8.2(γ) με χρήση πυλών CMOS λογικής, τότε θα έχουμε (βλέπε Σχήμα 7.2 Εργ. Άσκ. 7 για απ' ευθείας υλοποίηση της XOR πύλης):

$$\text{Κόστος Αθροιστή}_1 = \langle \text{SUM} \rangle + \langle \text{CARRY} \rangle = 2 \cdot \text{XOR} + 1 \cdot \text{OR}_{3\text{input}} + 3 \cdot \text{AND}_{2\text{input}} = 2 \cdot 22 + 1 \cdot 8 + 3 \cdot 6 = 44 + 8 + 18 \Rightarrow \text{Κόστος Αθροιστή}_1 = 70 \text{ τρανζίστορ}$$

Μετασχηματίζοντας το κύκλωμα με πύλες συμπληρωματικής λογικής μόνο με χρήση του κανόνα De Morgan, παίρνουμε για το κρατούμενο του πλήρους αθροιστή:

$$CARRY = A \cdot B + B \cdot C + A \cdot C = \overline{\overline{A \cdot B + B \cdot C + A \cdot C}} = \overline{\overline{A \cdot B} \cdot \overline{B \cdot C} \cdot \overline{A \cdot C}}$$

Η αντίστοιχη ανάλυση εισαγωγής φυσαλίδων για τις XOR πύλες έγινε στην Άσκηση 7. Συνεπώς το καινούργιο κόστος για τον πλήρη αθροιστή θα είναι:

$$\text{Κόστος Αθροιστή}_2 = \langle \text{SUM} \rangle + \langle \text{CARRY} \rangle = 2 \cdot \text{XOR} + 1 \cdot \text{NAND}_{3\text{input}} + 3 \cdot \text{NAND}_{2\text{input}} = 2 \cdot 16 + 1 \cdot 6 + 3 \cdot 4 = 32 + 6 + 12 \Rightarrow \text{Κόστος Αθροιστή}_2 = 50 \text{ τρανζίστορ}$$

Η χρήση σύνθετων πυλών για την υλοποίηση των πυλών XOR του αθροίσματος SUM και της έκφρασης του κρατουμένου CARRY από τη σχέση 7.1, μπορεί να μειώσει ακόμα περισσότερο το κόστος της υλοποίησης. Αν στη σχέση 7.1 ομαδοποιηθούν ορισμένοι όροι τότε αυτή θα γίνει:

$$CARRY = A \cdot B + B \cdot C + A \cdot C = (A + C) \cdot B + A \cdot C$$

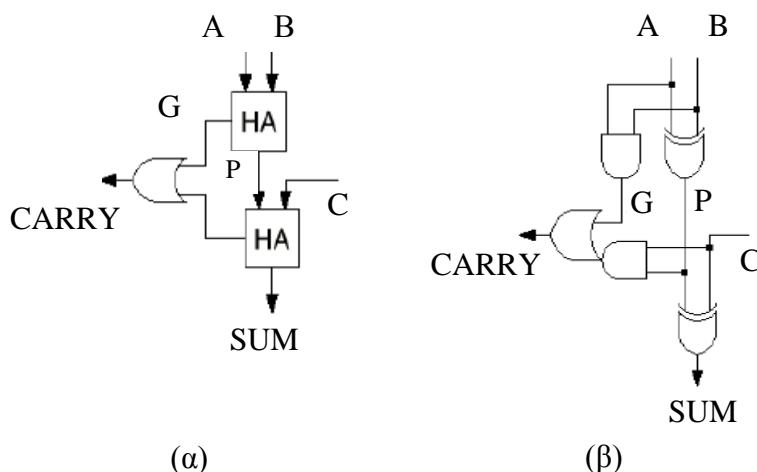
και επομένως το κόστος για αυτή την υλοποίηση θα είναι:

$$\text{Κόστος Αθροιστή 3} = \langle \text{SUM} \rangle + \langle \text{CARRY} \rangle = 2 * \text{XOR} + \langle \text{CARRY} \rangle = 2 * 12 + \langle 10 + 2 \rangle = 24 + 12 = 36$$

τρανζίστορ

όπου τα 2 τρανζίστορ στο κόστος του CARRY αντιστοιχούν στον αντιστροφέα που πρέπει να προστεθεί στην έξοδο της σύνθετης πύλης για την ορθότητα της λογικής της λειτουργίας.

Μια εναλλακτική υλοποίηση με χρήση ημιαθροιστών μπορεί να δώσει εξίσου οικονομικό κόστος για τον πλήρη αθροιστή. Το κύκλωμα με χρήση ημιαθροιστών παρουσιάζεται στο Σχήμα 8.3 και βασίζεται στην πρόβλεψη κρατουμένου (Carry Lookahead Adder - CLA).



Σχήμα 8.3: (α) Δομικό διάγραμμα και (β) κύκλωμα με πύλες ενός πλήρους αθροιστή από ημιαθροιστές.

Τα σήματα που εμφανίζονται στα σχήματα 8.3α και β παρέχονται από τις ακόλουθες σχέσεις:

$$P = A \oplus B \quad G = A \cdot B \quad \text{SUM} = P \oplus C \quad \text{CARRY} = G + P \cdot C$$

Με βάση την ανάλυση κόστους που έγινε για τον ημιαθροιστή για υλοποίηση με σύνθετες πύλες, θα έχουμε για το κόστος:

$$\text{Κόστος Αθροιστή 4} = 2 * \text{Κόστος Ημιαθροιστή} + 1 * \text{OR} = 2 * 18 + 1 * 6 = 36 + 6 = 42 \text{ τρανζίστορ}$$

Μολονότι τα δύο τελευταία κόστη είναι συγκρίσιμα και αρκετά μικρά, υπάρχει η δυνατότητα με κατάλληλο χειρισμό των λογικών συναρτήσεων του SUM και του CARRY να καταλήξουμε σε ακόμα πιο οικονομική υλοποίηση. Πράγματι, μπορούμε να δώσουμε την παρακάτω μορφή στη συνάρτηση αθροίσματος (η επιβεβαίωση της ορθότητας αφήνεται ως άσκηση για το σπίτι):

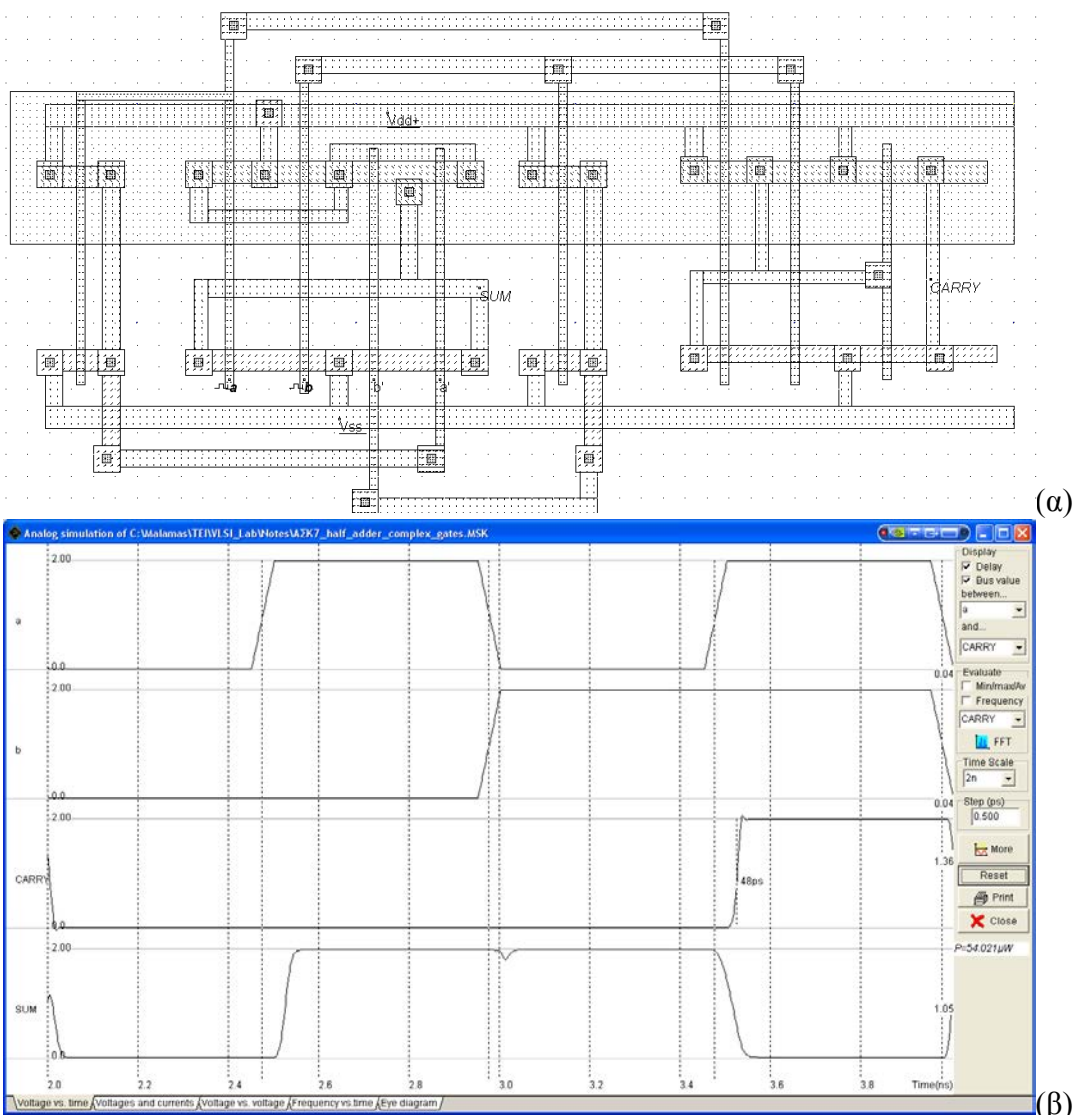
$$\text{SUM} = A \cdot B \cdot C + (A + B + C) \cdot \overline{\text{CARRY}} \quad 8.2$$

Η υλοποίηση τόσο του CARRY όσο και του SUM με βάση τη σχέση 8.2 και χρήση μεθοδολογίας σύνθετων πυλών, θα δώσει το παρακάτω κόστος:

$$\text{Κόστος Αθροιστή 5} = \langle \text{SUM} \rangle + \langle \text{CARRY} \rangle = \langle 14 + 2 \rangle + \langle 10 + 2 \rangle = 28 \text{ τρανζίστορ}$$

όπου τα 2 τρανζίστορ στο κόστος των SUM και CARRY αντιστοιχούν στον αντιστροφέα που πρέπει να συνδεθεί στην έξοδο των αντίστοιχων σύνθετων πυλών για την ορθότητα της λογικής τους λειτουργίας.

Στο επόμενο σχήμα δίνεται ο φυσικός σχεδιασμός του ημιαθροιστή. Ο σχεδιασμός έχει γίνει χρησιμοποιώντας τις βασικές αρχές σχεδιασμού των σύνθετων πυλών που δόθηκαν αναλυτικά στην προηγούμενη άσκηση για την πύλη XOR. Είναι καλύτερα να χρησιμοποιηθούν γραμμές πολυκρυσταλλικού πυριτίου μόνο για μικρές σε μήκος διασυνδέσεις λόγω της μεγάλης ειδικής αντίστασης και της μεγάλης χωρητικότητας που παρουσιάζει το υλικό αυτό. Στις υπόλοιπες διασυνδέσεις χρησιμοποιούμε γραμμές μετάλλου και επαφές μετάλλου/πολυκρυσταλλικού πυριτίου και μετάλλου/διάχυσης.



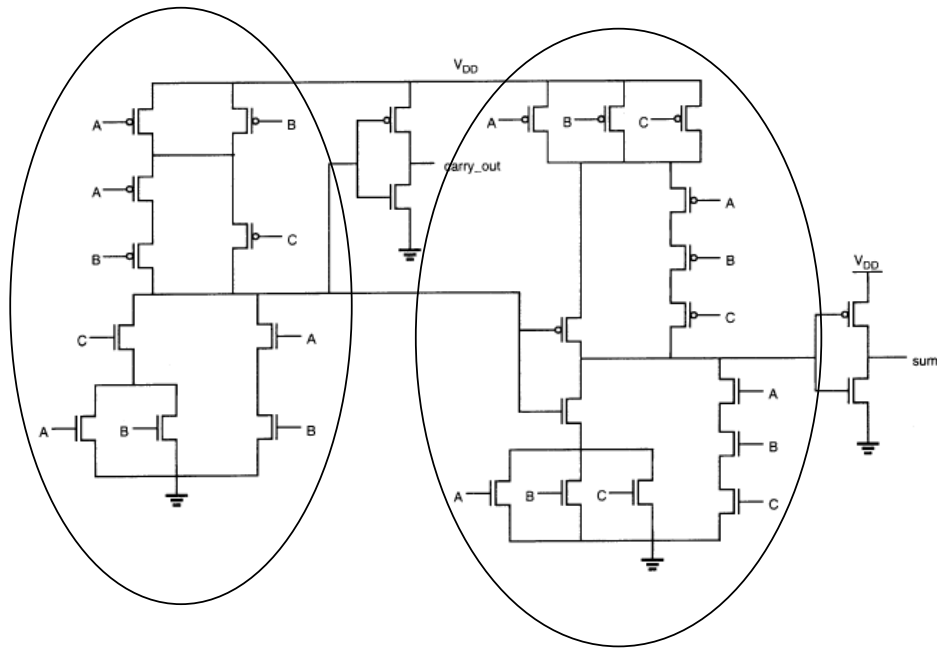
Σχήμα 8.4: (α) το layout και (β) η εξομοίωση του ημιαθροιστή.

Πραγματοποιώντας την εξομοίωση του κυκλώματος, μπορεί να επαληθευτεί ο πίνακας αλήθειας του ημιαθροιστή. Αποτελέσματα εξομοίωσης για ένα συνδυασμό εισόδων φαίνονται στο Σχήμα 8.4(β). Επίσης μπορούμε να μετρήσουμε και τις χωρητικότητες εισόδων και εξόδων του ημιαθροιστή, πιέζοντας δύο φορές το αριστερό πλήκτρο του “ποντικιού” στα αντίστοιχα σημεία του κυκλώματος.

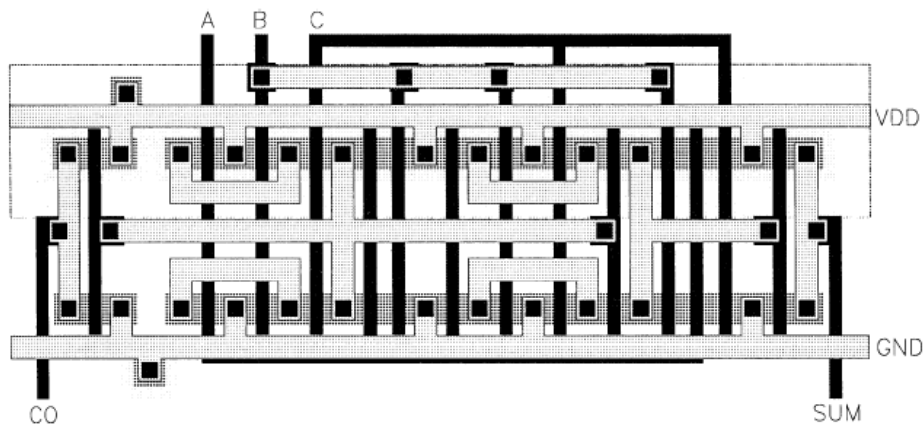
Το κύκλωμα τρανζίστορ του πλήρους αθροιστή με βάση τη σχέση 8.2 για το SUM, φαίνεται στο Σχήμα 8.5(α). Ο φυσικός σχεδιασμός του πλήρους αθροιστή δίνεται στο Σχήμα 8.5(β) και η εξομοίωση της λειτουργίας του στο Σχήμα 8.5(γ). Μπορούμε να παρατηρήσουμε ότι το κύκλωμα της σύνθετης πύλης για το κρατούμενο CARRY βρίσκεται στο αριστερό μέρος του κυκλώματος, ενώ το αντίστοιχο κύκλωμα για τη σύνθετη πύλη του SUM στο δεξιό μέρος. Ενδιάμεσα φαίνονται οι αντιστροφείς που χρειάζονται για την ορθή λογική λειτουργία των σύνθετων πυλών.

ΣΥΝΘΕΤΗ ΠΥΛΗ CARRY

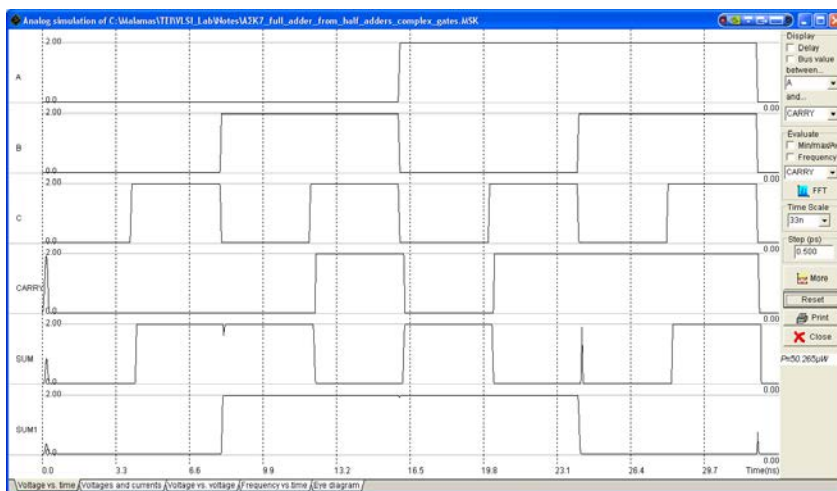
ΣΥΝΘΕΤΗ ΠΥΛΗ SUM



(α)



(β)



(γ)

Σχήμα 8.5: (α) Κύκλωμα τρανζίστορ, (β) layout, και (γ) εξομοίωση ενός πλήρους αθροιστή.

### ΠΡΑΚΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Σχεδιάστε το φυσικό σχέδιο του ημιαθροιστή του Σχήματος 8.1 με χρήση συμπληρωματικών πυλών. Εξομοιώστε το σχέδιό σας και επαληθεύστε τον πίνακα αληθείας του Σχήματος 8.1(β).

Κατόπιν σχεδιάστε το φυσικό σχέδιο του πλήρους αθροιστή του Σχήματος 8.2(α) χρησιμοποιώντας τη σχέση 8.2 για το SUM:

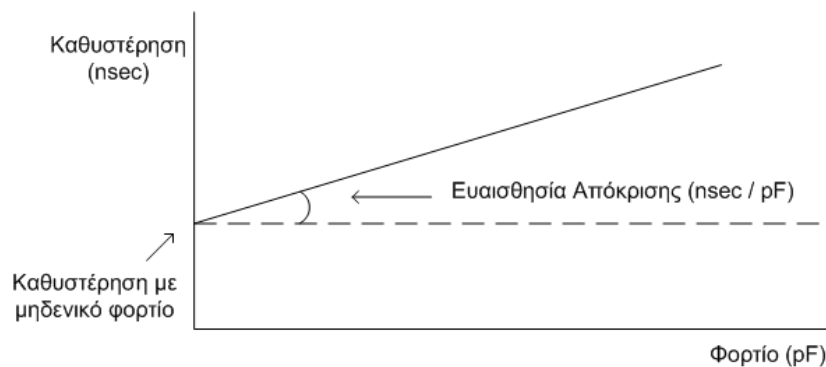
$$SUM = A \cdot B \cdot C + (A + B + C) \cdot \overline{CARRY}$$

και την τυπική έκφραση για το CARRY:


$$CARRY = A \cdot B + B \cdot C + A \cdot C = (A + B) \cdot C + A \cdot B$$

Χρησιμοποιήστε συμπληρωματικές πύλες για την υλοποίηση του SUM και σύνθετη πύλη για την υλοποίηση του CARRY. Εξομοιώστε το σχέδιό σας και επαληθεύστε τον πίνακα αληθείας του Σχήματος 8.2(β).

Το επόμενο βήμα της άσκησης είναι η εξομοίωση και ο πλήρης χρονικός χαρακτηρισμός του αθροιστή. Μπορούμε εύκολα να μετρήσουμε την χρονική καθυστέρηση από κάθε είσοδο σε κάθε έξοδο, πχ μετρώντας την καθυστέρηση ανόδου και καθόδου από την είσοδο έως την έξοδο Sum. Η μέτρηση αυτή όμως γίνεται χωρίς την ύπαρξη κάποιου φορτιού στην έξοδο του αθροιστή. Για να ολοκληρώσουμε τον χρονικό χαρακτηρισμό του αθροιστή πρέπει να μετρήσουμε την καθυστέρηση για διαφορετικά φορτία σε κάθε έξοδο. Με αυτόν τον τρόπο μετράμε την ευαισθησία απόκρισης για κάθε έξοδο του αθροιστή, η οποία είναι η κλίση της ευθείας του παρακάτω σχήματος.



### ΑΣΚΗΣΕΙΣ

- 8.1 Να γίνει ο χρονικός χαρακτηρισμός του πλήρους αθροιστή που σχεδιάστηκε στην άσκηση, με μηδενικό φορτίο στην αρχή και στη συνέχεια συνδέοντας στις εξόδους του φορτία  $n \cdot 1\text{pF}$ , με  $n=1, 2, 4, 8, 16$ , χρησιμοποιώντας τη λειτουργία της παλέτας  για την προσθήκη εικονικού χωρητικού φορτίου στις εξόδους SUM και CARRY του αθροιστή.
- 8.2 Να σχεδιαστεί ένας 4-bit αθροιστής διάδοσης κρατουμένου με χρήση του πλήρους αθροιστή που σχεδιάστηκε στην άσκηση.

## 9. Μελέτη και υλοποίηση ακολουθιακών κυκλωμάτων.

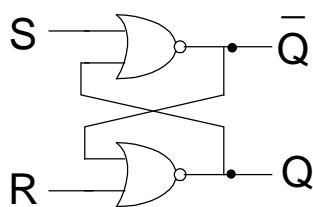
**ΣΚΟΠΟΣ:** Σκοπός αυτής της άσκησης είναι η μελέτη και ο φυσικός σχεδιασμός απλών ακολουθιακών κυκλωμάτων.

### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

#### Δισταθή στοιχεία.

Τα ακολουθιακά κυκλώματα διαφοροποιούνται από τα συνδυαστικά στο ότι διαθέτουν μνήμη, δηλαδή η τρέχουσα κατάσταση των εξόδων τους προσδιορίζεται από την κατάσταση των εισόδων τους και την προηγούμενη κατάσταση των εξόδων τους. Η βάση των ακολουθιακών κυκλωμάτων είναι τα δισταθή στοιχεία, δηλαδή κυκλώματα των οποίων οι εξοδοι μπορούν να ηρεμούν μεταξύ δύο καταστάσεων, η δε μετάβαση από τη μία κατάσταση στην άλλη εξασφαλίζεται με κατάλληλη διέγερση των εισόδων. Ένα δισταθές στοιχείο μπορεί να κατασκευαστεί με τη διασύζευξη (cross-coupling) κατάλληλων διακοπτικών στοιχείων.

Αν τα διακοπτικά αυτά στοιχεία είναι πύλες NOR 2-εισόδων, τότε το αποτέλεσμα είναι ο SR μανταλωτής (latch) του Σχήματος 9.1.



(α)

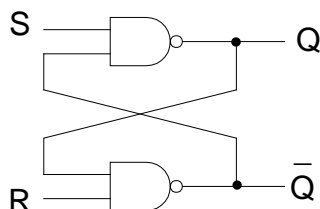
S	R	$Q_{n+1}$	$\overline{Q_{n+1}}$	Λειτουργία
0	0	$Q_n$	$\overline{Q_n}$	Κράτηση (Αμετάβλητο)
0	1	0	1	Reset (Επαναφορά στο 0)
1	0	1	0	Set (Θέση στο 1)
1	1	0	0	Απαγορευμένη (Απροσδιόριστη)

(β)

Σχήμα 9.1: (α) Λογικό διάγραμμα και (β) Πίνακας αληθείας του μανταλωτή SR με πύλες NOR με εισόδους ενεργές στο 1.

Ο συνδυασμός  $S=0, R=1$  προκαλεί την έξοδο να μηδενιστεί ( $Q=0$ ), ενώ ο συνδυασμός  $S=1, R=0$  προκαλεί την έξοδο να τεθεί στο 1 ( $Q=1$ ). Ο συνδυασμός  $S=R=0$  αφήνει την έξοδο αναλλοίωτη, ενώ ο συνδυασμός  $S=R=1$  προκαλεί τις δύο εξόδους να λάβουν την ίδια τιμή, πράγμα που εξ' ορισμού δεν επιτρέπεται, και επομένως ο συνδυασμός αυτός θα πρέπει να μην εμφανίζεται σε αυτόν το μανταλωτή.

Αν αντί για πύλες NOR χρησιμοποιηθούν πύλες NAND, τότε ο SR μανταλωτής θα γίνει όπως φαίνεται στο Σχήμα 9.2.



(α)

S	R	$Q_{n+1}$	$\overline{Q_{n+1}}$	Λειτουργία
0	0	1	1	Απαγορευμένη
0	1	1	0	Set
1	0	0	1	Reset
1	1	$Q_n$	$\overline{Q_n}$	Κράτηση

(β)

Σχήμα 9.2: (α) Λογικό διάγραμμα και (β) Πίνακας αληθείας του μανταλωτή SR με πύλες NAND και εισόδους ενεργές στο 0.

Από το Σχήμα 9.2(β) παρατηρούμε ότι για το μανταλωτή με πύλες NAND υπάρχουν καταστάσεις αντίστοιχες με το μανταλωτή με πύλες NOR, μόνο που η απαγορευμένη κατάσταση εμφανίζεται για  $S=R=0$ , ενώ η κατάσταση κράτησης για  $S=R=1$ .

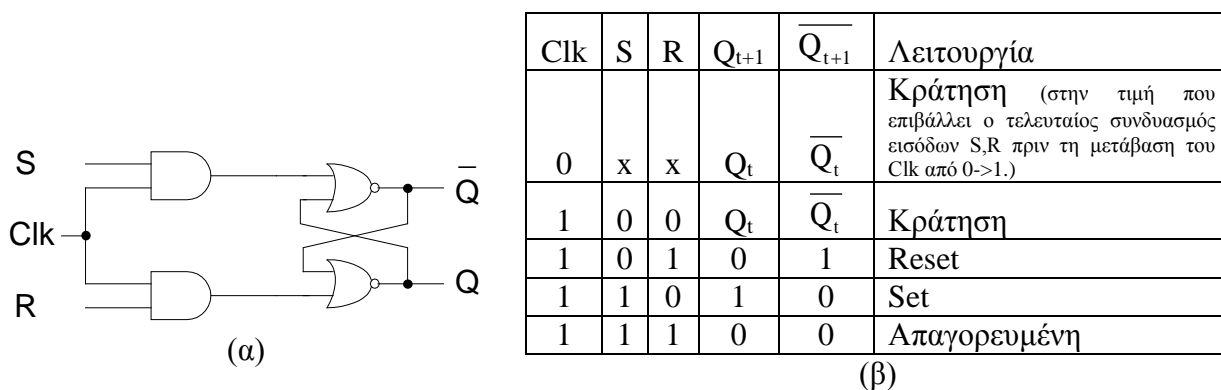
ΠΑΡΑΤΗΡΗΣΕΙΣ: α) Από την εξέταση των πινάκων καταστάσεων των δύο μανταλωτών παρατηρούμε ότι η μεν κατάσταση κράτησης εμφανίζεται για εισόδους ίσες με τις λογικές τιμές επίτρεψης των αντιστοιχών πυλών (0 για NOR, 1 για NAND), ενώ η απαγορευμένη κατάσταση εμφανίζεται για εισόδους ίσες με τις ελέγχουσες τιμές των αντίστοιχων πυλών (1 για NOR, 0 για NAND).

β) Έτσι όπως έχουν οριστεί οι εισόδοι, ο μεν μανταλωτής με NOR έχει εισόδους ενεργές στο 1, ο δε μανταλωτής με NAND έχει εισόδους ενεργές στο 0. Τα ίδια κυκλώματα αλλά με διαφορετική ανάθεση ονομάτων στις εισόδους μπορούν να χρησιμοποιηθούν και για τη συμπληρωματική λειτουργία, δηλαδή μανταλωτής με NOR και εισόδους ενεργές στο 0 και μανταλωτής με NAND και εισόδους ενεργές στο 1 (Άσκηση για το σπίτι).

**Σύγχρονοι μανταλωτές.**

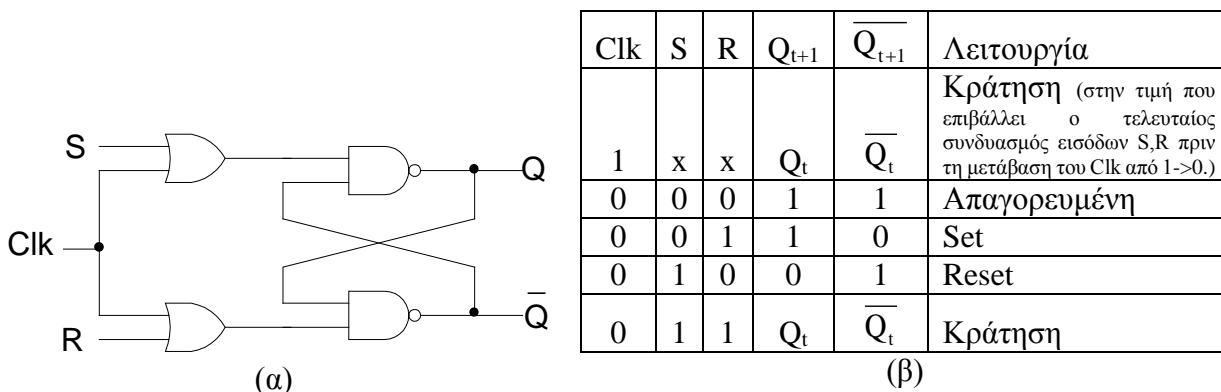
Η λειτουργία των παραπάνω κυκλωμάτων είναι ασύγχρονη, καθώς οι έξοδοι αλλάζουν τιμή οποιαδήποτε στιγμή αλλάζουν τιμή οι εισόδοι (κατόπιν βεβαίως παρελεύσεως της καθυστέρησης που προκαλείται από τις συστατικές πύλες). Για να υπάρχει ένας ελεγχόμενος τρόπος αλλαγής των εξόδων κατά τη διάρκεια ενός συγκεκριμένου χρονικού διαστήματος (σύγχρονος τρόπος απόκρισης του κυκλώματος), απαιτείται η χρήση ενός σήματος χρονισμού (ρολόι) και κάποιων πυλών για την εφαρμογή του σήματος αυτού στο μανταλωτή.

Στο Σχήμα 9.3 παρουσιάζεται η σύγχρονη εκδοχή του μανταλωτή SR με πύλες NOR.



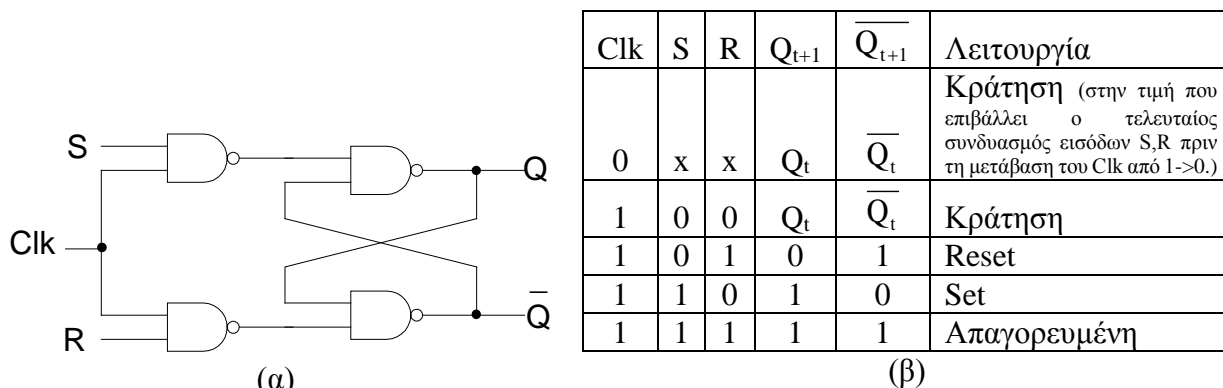
Σχήμα 9.3: (α) Λογικό κύκλωμα και (β) πίνακας αληθείας του σύγχρονου μανταλωτή με πύλες NOR και εισόδους ενεργές στο 1.

Η χρήση πυλών AND για την εφαρμογή του ρολογιού επιτρέπει στις εξωτερικές εισόδους S και R να διατηρήσουν την ίδια λογική (ενεργές στο 1) με την ασύγχρονη εκδοχή του μανταλωτή. Με παρόμοιο τρόπο κατασκευάζεται και ο σύγχρονος μανταλωτής με πύλες NAND, όπως φαίνεται στο Σχήμα 9.4.



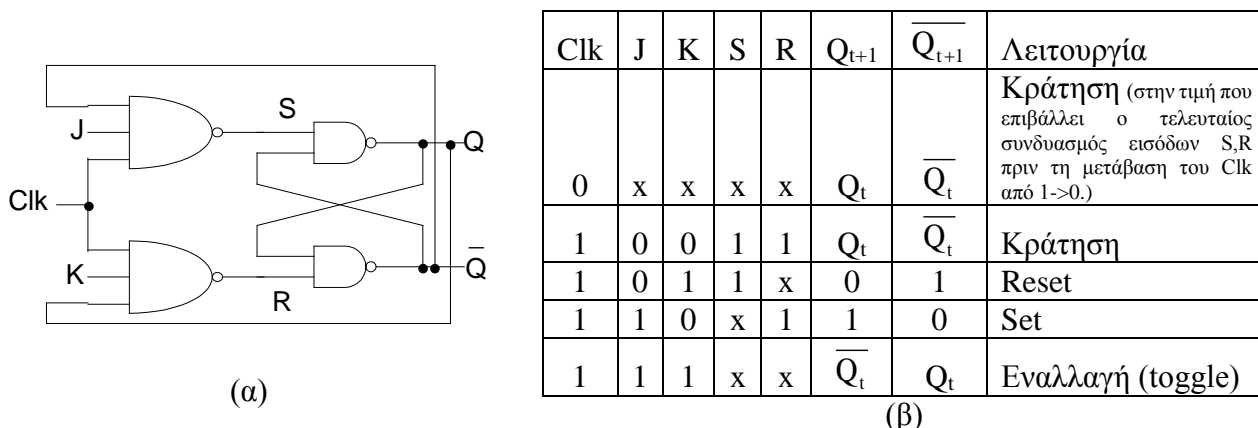
Σχήμα 9.4: (α) Λογικό διάγραμμα και (β) πίνακας αληθείας σύγχρονου μανταλωτή SR με πύλες NAND και εισόδους ενεργές στο 0.

Προκειμένου να αλλάξει η λογική των εισόδων του παραπάνω κυκλώματος και αυτές να γίνουν ενεργές στο 1, οι πύλες OR μπορούν να αντικατασταθούν με πύλες NAND, οπότε θα έχουμε μια πλήρως NAND υλοποίηση του μανταλωτή SR. Αυτή η υλοποίηση φαίνεται στο Σχήμα 9.5



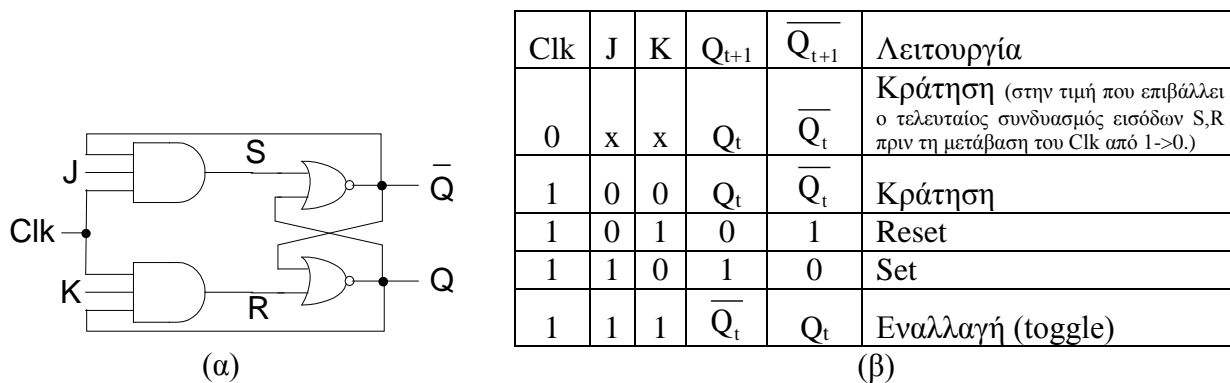
Σχήμα 9.5: (α) Λογικό διάγραμμα και (β) πίνακας αληθείας του σύγχρονου μανταλωτή SR με πύλες NAND μόνο και εισόδους ενεργές στο 1.

Μολονότι οι μανταλωτές των Σχημάτων 9.3, 9.4 και 9.5 διαθέτουν σύγχρονη λειτουργία με τη χρήση σήματος ρολογιού, συνεχίζουν να υποφέρουν από τις ίδιες απροσδιοριστίες με τις ασύγχρονες εκδοχές τους, όπως φαίνεται και από τους πίνακες αληθείας. Για να εξαιρεθούν αυτές οι απροσδιοριστίες κατασκευάστηκε ο μανταλωτής JK του οποίου τη σύγχρονη εκδοχή βλέπουμε στο Σχήμα 9.6. Ο μανταλωτής αυτός προέρχεται από τη NAND-μόνο υλοποίηση του Σχήματος 9.5, αλλά θα μπορούσε να κατασκευαστεί από οποιαδήποτε άλλη υλοποίηση που παρουσιάστηκε μέχρι τώρα, με ανάδραση των τελικών εξόδων Q και  $\bar{Q}$  στις βασικές εισόδους.



Σχήμα 9.6: (α) Λογικό διάγραμμα και (β) πίνακας αληθείας του σύγχρονου μανταλωτή JK με πύλες NAND-μόνο και εισόδους ενεργές στο 1.

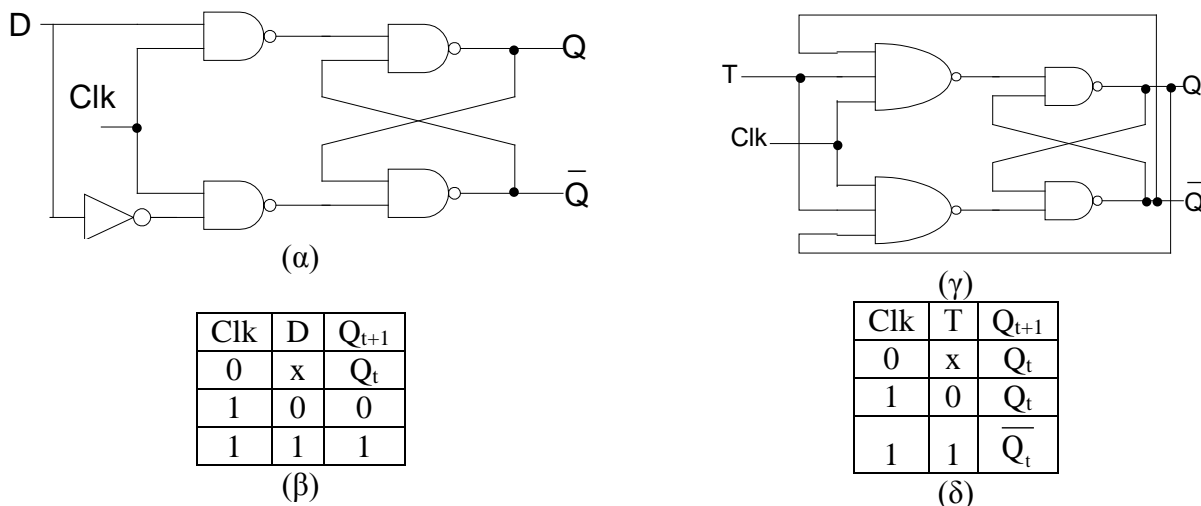
Στο κύκλωμα αυτό η απροσδιοριστία των μανταλωτών SR έχει εξαιρεθεί, καθώς για είσοδο  $J=K=1$  η απροσδιοριστία έχει αντικατασταθεί από μια κατάσταση εναλλαγής, όπου οι έξοδοι παίρνουν τιμή συμπληρωματική σε σχέση με αυτή που είχαν κατά τον προηγούμενο κύκλο ρολογιού. Για πληρότητα, στο Σχήμα 9.7 δείχνουμε τον αντίστοιχο σύγχρονο μανταλωτή JK από το μανταλωτή SR με πύλες NOR.



Σχήμα 9.7: (α) Λογικό διάγραμμα και (β) συνεπτυγμένος πίνακας αληθείας του σύγχρονου μανταλωτή JK από μανταλωτή SR με NOR πύλες

Η εναλλαγή που υφίσταται η έξοδος των μανταλωτών στο Σχήμα 9.6 (και στο Σχήμα 9.7) για το συνδυασμό  $J=K=1$  μπορεί να δημιουργήσει αστάθεια στο μανταλωτή JK, καθώς η έξοδος του κυκλώματος θα ταλαντευθεί μέχρις ότου κάποια από τις εισόδους γίνει 0 ή το ρολόι γίνει 0. Για να αποφευχθεί αυτό το ανεπιθύμητο πρόβλημα χρονισμού, η διάρκεια παραμονής του ρολογιού στο 1 πρέπει να είναι μικρότερη από την καθυστέρηση διάδοσης από την είσοδο στην έξοδο του μανταλωτή JK. Αυτός ο περιορισμός επιβάλλει στο σήμα ρολογιού να κατέβει στο 0 πριν η τιμή εξόδου έχει την ευκαιρία να αλλάξει πάλι, οπότε έτσι αποτρέπεται η μη ελεγχόμενη ταλάντωση της εξόδου. Πρακτικά όμως είναι δύσκολο να επιτευχθεί αυτός ο περιορισμός οπότε και ο μανταλωτής JK χρησιμοποιείται σπάνια.

Υπάρχουν όμως ορισμένες πρακτικά χρήσιμες παραλλαγές των σύγχρονων μανταλωτών που παρουσιάστηκαν παραπάνω, όπως είναι ο μανταλωτής D και ο μανταλωτής T, οι οποίοι και παρουσιάζονται στο Σχήμα 9.8.



Σχήμα 9.8: (α) Λογικό διάγραμμα, (β) πίνακας αληθείας μανταλωτή D. (γ) Λογικό διάγραμμα και (δ) πίνακας αληθείας μανταλωτή T.

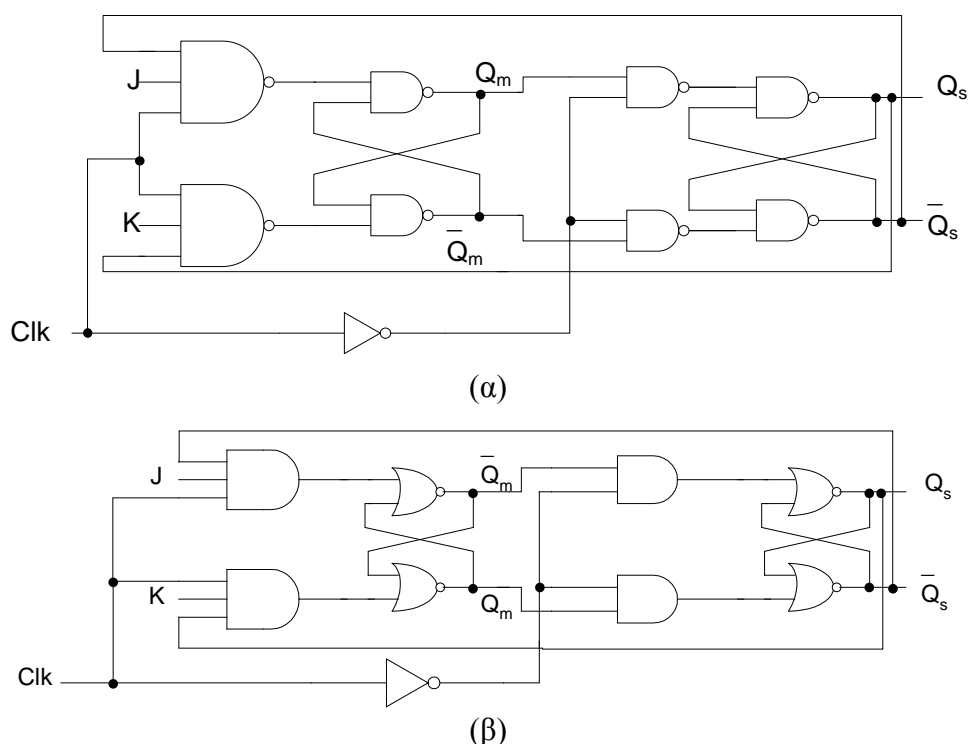
Οι μανταλωτές του Σχήματος 9.8 χρησιμοποιούνται κυρίως σε καταχωρητές ή ως στοιχεία καθυστέρησης (D) και σε μετρητές (T).

**Flip-flop τύπου Αφέντη-Σκλάβου (master-slave flip-flops).**

Οι μανταλωτές που παρουσιάστηκαν τελευταία (Σχήμα 9.8) είναι κυκλώματα που χρησιμοποιούνται ως στοιχεία μνήμης, όταν οι απαιτήσεις για μειωμένο υλικό είναι επιτακτικές ή ακόμα και σε υψηλής ταχύτητας κυκλώματα, καθώς η λειτουργία τους ολοκληρώνεται σε μισή

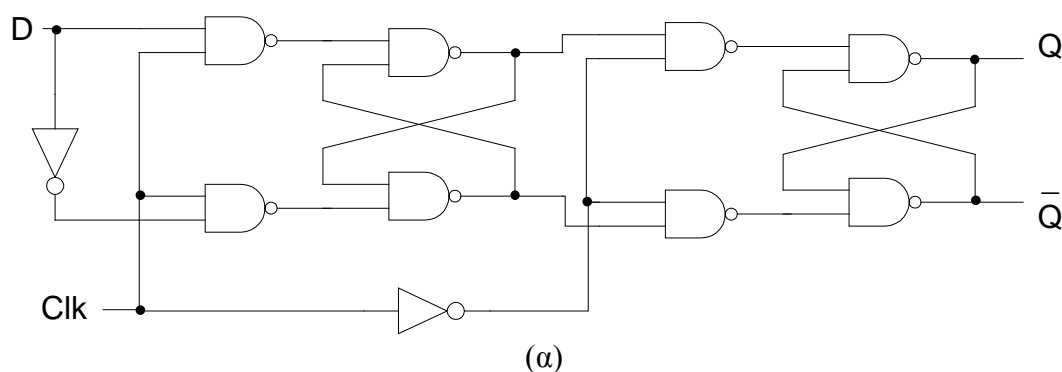
περίοδο (διέγερση εισόδου, αποθήκευση πληροφορίας, εμφάνιση πληροφορίας στην έξοδο όσο το ρολόι είναι ενεργό). Όμως δεν μπορούν να χρησιμοποιηθούν σε κυκλώματα με ανάδραση (πχ κυκλώματα pipeline) καθώς υπάρχει ο κίνδυνος χρονικής αστάθειας όταν στην διάρκεια του παλμού ρολογιού (παραμονή στο 1) η είσοδος του μανταλωτή αλλάξει, λόγω εφαρμογής της τρέχουσας εξόδου του μανταλωτή στο κύκλωμα ανάδρασης που τροφοδοτεί την είσοδό του.

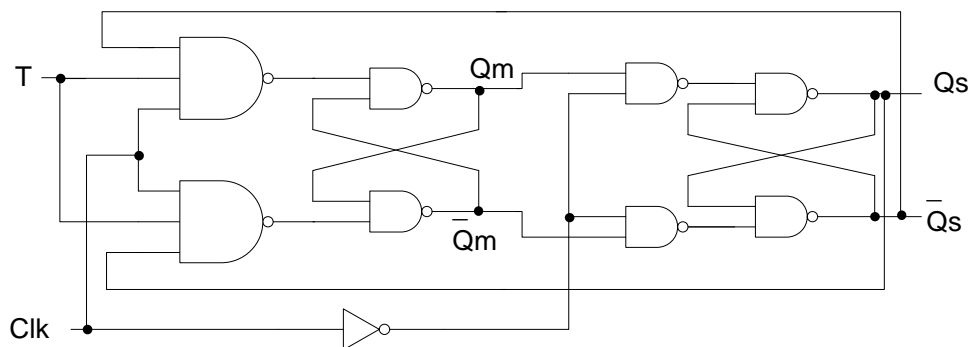
Αυτό το πρόβλημα χρονισμού μπορεί να εξλειφθεί αν χρησιμοποιηθούν δύο όμοια κυκλώματα σύγχρονων μανταλωτών σε σειρά, έτσι ώστε οι εξοδοί του πρώτου να οδηγούν τις εισόδους του δεύτερου. Οι δύο μανταλωτές συγχρονίζονται από ρολόγια με ανάστροφες φάσεις, ώστε να αποθηκεύουν καινούργια πληροφορία με διαδοχικό τρόπο. Αυτή η διάταξη καλείται flip-flop αφέντη-σκλάβου (Master-Slave FF). Στο Σχήμα 9.9 παρουσιάζεται το JK MS FF, υλοποιημένο με δύο τρόπους.



Σχήμα 9.9: (α) Υλοποίηση βασισμένη σε πύλες NAND και (β) υλοποίηση βασισμένη σε πύλες NOR του JK MS FF.

Τέλος, στο Σχήμα 9.10 παρουσιάζονται οι υλοποιήσεις των D και T MS FF με πύλες NAND-μόνο (με εξαίρεση τους αντιστροφείς των σημάτων οι οποίοι μπορούν επίσης να αντικατασταθούν από NAND πύλες).






(β)

Σχήμα 9.10: (α) Υλοποίηση του D MS FF. (β) Υλοποίηση του T MS FF.

**ΠΑΡΑΤΗΡΗΣΗ:** Η χρήση αντιστροφέα για τη δημιουργία ρολογιών ανάστροφης φάσης μπορεί να αποφευχθεί αν ο αφέντης μανταλωτής έχει εισόδους ενεργές στο 1 και ο σκλάβος μανταλωτής έχει εισόδους ενεργές στο 0. Σε αυτή την περίπτωση οι δύο μανταλωτές μπορούν να οδηγούνται από το ίδιο ρολόι (μια φάση) και έτσι να μη χρησιμοποιηθεί ο αντιστροφέας, μειώνοντας έτσι το κόστος της υλοποίησης κατά δύο τρανζίστορ.

## ΠΡΑΚΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Για τη μελέτη των ακολουθιακών κυκλωμάτων, μπορείτε να ξεκινήσετε επιλέγοντας ένα είδος μανταλωτή (με πύλες NAND ή με πύλες NOR) και πάνω σε αυτόν να "κτίσετε" τα υπόλοιπα κυκλώματα της άσκησης. Για την εξομοίωση των κυκλωμάτων στο Microwind εξυπηρετεί περισσότερο η χρήση γεννήτριας συγκεκριμένων παλμών, παρά η χρήση γεννήτριας ρολογιού. Την εισαγωγή παλμών την κάνουμε είτε από το εικονίδιο  της παλέτας του Microwind (τρίτη σειρά, πέμπτο από αριστερά), είτε κάνοντας διπλό κλικ στο στίγμα (χαρακτηριστικό τετραγωνάκι) δίπλα από το όνομα του σήματος εισόδου και κατόπιν επιλέγοντας την ετικέτα "PWL" (ετικέτα "Piece-Wise" σε παλαιότερες εκδόσεις του Microwind) στο παράθυρο χαρακτηρισμού σημάτων.

Επίπεδα λογικού 0 και λογικού 1, βασισμένα στην επιλεγμένη τεχνολογία.

Πεδίο κειμένου εισαγωγής αλληλουχίας λογικών επιπέδων της γεννήτριας

Επεξεργάσιμος πίνακας χρονικής περιγραφής της γεννήτριας παλμών.

Κουμπιά εισαγωγής παλμών στον πίνακα, σβησίματος πίνακα και αναστροφής δεδομένων πίνακα.

Time(ns)	0/1/2bdr
0.0	0
1.0	1
2.0	0

Σχήμα 9.11: Χρήση της ετικέτας εισαγωγής γεννήτριας παλμών.

## I. Ασύγχρονοι μανταλωτές.

Προκειμένου να εξομοιωθεί ο ασύγχρονος μανταλωτής SR του Σχήματος 9.2 (8.1) κάνουμε την εξής διαδικασία:

1. Σχεδιάζουμε στο Microwind το κύκλωμα,
2. Ονομάζουμε τις εξόδους και τις κάνουμε ορατές στην εξομοίωση,
3. Ονομάζουμε τις εισόδους S και R (ΠΡΟΣΟΧΗ στην αναγνώριση του σωστού ακροδέκτη), και επιλέγοντας την ετικέτα "PWL" (βλ. Σχήμα 9.11) εισάγουμε τις παρακάτω γεννήτρίες παλμών στο αντίστοιχο πεδίο ως εξής:

- α. Πατάμε "Clear" για τον καθαρισμό του πίνακα δεδομένων,
- β. Εισάγουμε τις λογικές τιμές του παρακάτω πίνακα στο πεδίο "Seq:" για το κύκλωμα του Σχήματος 9.2:

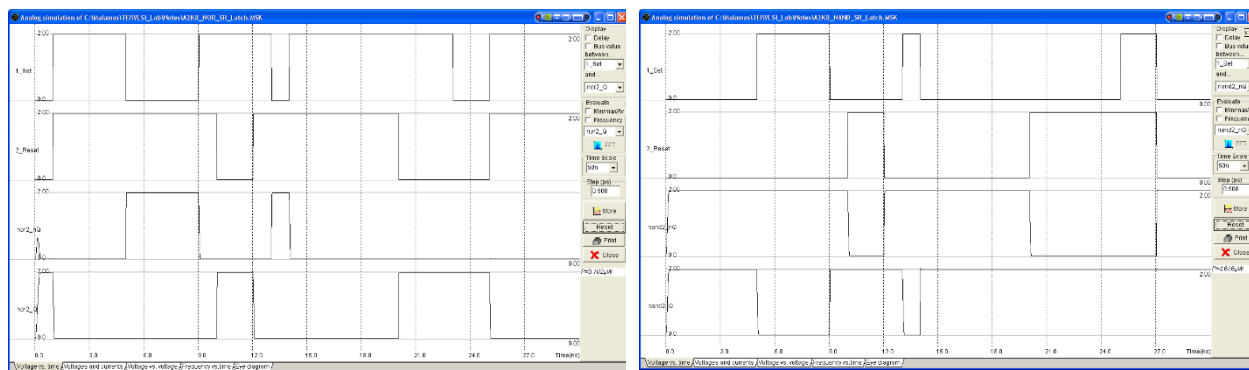
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
S	1	1	1	1	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1
R	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1

ή του παρακάτω πίνακα (συμπληρωματικός του πάνω πίνακα) για το κύκλωμα του Σχήματος 9.1:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
S	0	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0

- γ. Πατάμε "Insert" για την εισαγωγή των λογικών τιμών στον πίνακα δεδομένων,
- δ. Πατάμε "Visible in simu" για την εμφάνιση της αντίστοιχης εισόδου στην εξομοίωση,
- ε. Πατάμε "Assign" για την εφαρμογή της παλμο-γεννήτριας στην αντίστοιχη είσοδο.

Από την παραπάνω διαδικασία παίρνουμε τις εξομοιώσεις του Σχήματος 9.12 για τα αντίστοιχα κυκλώματα. Από αυτές τις γραφικές παραστάσεις μπορούμε να επιβεβαιώσουμε τον πίνακα αληθείας του αντίστοιχου κυκλώματος.



(α) (β)

Σχήμα 9.12: (α) Εξομοίωση ασύγχρονου μανταλωτή NAND SR με εισόδους ενεργές στο 0. (β) Εξομοίωση ασύγχρονου μανταλωτή NOR SR με εισόδους ενεργές στο 1.

**II. Σύγχρονοι μανταλωτές.**

Για την εξομοίωση των σύγχρονων μανταλωτών ακολουθούμε την ίδια διαδικασία όπως και για τους ασύγχρονους. Για την εισαγωγή του ρολογιού συγχρονισμού μπορούμε να εισάγουμε γεννήτρια ρολογιού με τα παρακάτω στοιχεία (ετικέτα "Clock"):

Όνομα σήματος: Clk, Χαρακτηριστικά:  $t_{low}/t_{rise}/t_{high}/t_{fall}$ : 3.950ns/0.05ns/3.950ns/0.05ns

Εναλλακτικά μπορούμε να εισάγουμε το ρολόι σαν γεννήτρια παλμών όπως και για τις εισόδους S και R, όπως παρακάτω (ετικέτα "PWL") για το μανταλωτή με εισόδους ενεργές στο 0:

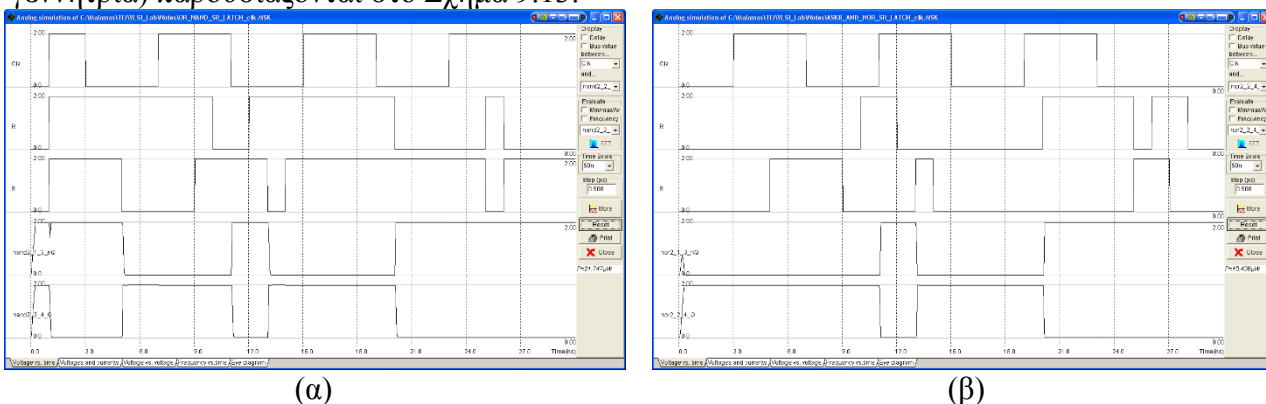
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Clk	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0

ή τον παρακάτω πίνακα για το μανταλωτή με εισόδους ενεργές στο 1:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Clk	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0

Clk	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Οι εξομοιώσεις που λαμβάνονται με την παραπάνω διαδικασία (με το ρολόι Clk ως παλμο-γεννήτρια) παρουσιάζονται στο Σχήμα 9.13.



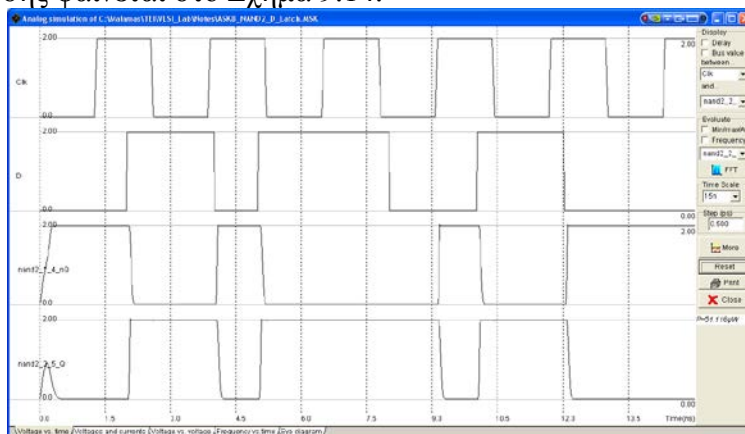
Σχήμα 9.13: (α) Εξομοίωση σύγχρονου μανταλωτή SR (NAND) με εισόδους ενεργές στο 0. (β) Εξομοίωση σύγχρονου μανταλωτή SR (NOR) με εισόδους ενεργές στο 1.

**III. Σύγχρονος μανταλωτής D.**

Με παρόμοιο τρόπο μπορούμε να εξομοιώσουμε τη λειτουργία του σύγχρονου μανταλωτή D, χρησιμοποιώντας τον παρακάτω πίνακα λογικών τιμών για την παλμο-γεννήτρια που θα εφαρμοστεί στην είσοδο D:

D	0	1	1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Μέρος της εξομοίωσης φαίνεται στο Σχήμα 9.14.



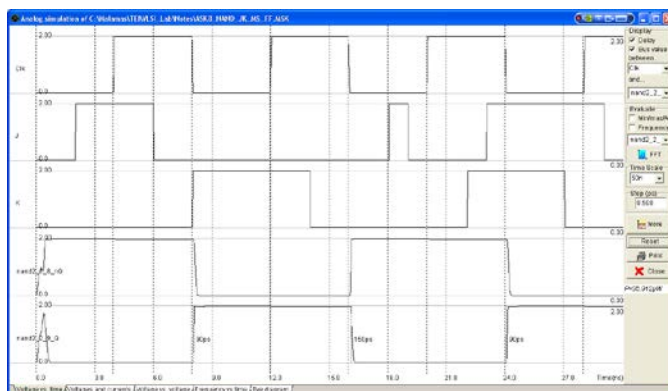
Σχήμα 9.14: Εξομοίωση σύγχρονου μανταλωτή D με εισόδους ενεργές στο 1.

**IV. Σύγχρονα κυκλώματα αφέντη-σκλάβου (MS FFs).**

Ανάλογα με ποιον μανταλωτή SR ξεκινήσατε, μπορείτε να σχεδιάσετε είτε το JK του Σχήματος 9.9(α) είτε του Σχήματος 9.9(β). Για την εξομοίωση του σύγχρονου JK MS FF, ακολουθούμε τη διαδικασία που παρουσιάστηκε παραπάνω, εφαρμόζοντας στις εισόδους του FF τις παρακάτω λογικές τιμές υπό τη μορφή παλμο-γεννήτριας:

	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	3			
J	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
K	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0

Για το σήμα ρολογιού CLK μπορεί να εφαρμοστεί γεννήτρια ρολογιού (ετικέτα "Clock") με τα εξής χαρακτηριστικά:  $t_{low}/t_{rise}/t_{high}/t_{fall}$ : 3.950ns/0.05ns/3.950ns/0.05ns. Η εξομοίωση παρουσιάζεται στο Σχήμα 9.15.

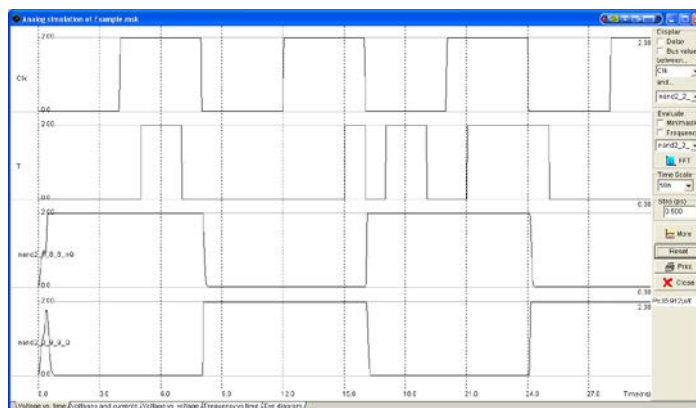


Σχήμα 9.15: Εξομοίωση JK master-slave flip-flop.

Τέλος, για τη μελέτη του T MS FF, χρησιμοποιήστε το κύκλωμα του JK MS FF που σχεδιάσατε προηγουμένως, ενώνοντας τις δύο εισόδους J και K μεταξύ τους και ονομάζοντας 'T' την κοινή είσοδο που θα προκύψει. Κατόπιν εξομοιώστε το κύκλωμα, χρησιμοποιώντας το ίδιο ρολόι με αυτό του JK για την είσοδο ρολογιού CLK και εφαρμόζοντας τις παρακάτω λογικές τιμές στην είσοδο T:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
T	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1	1	1	1

Η εξομοίωση παρουσιάζεται στο γράφημα του Σχήματος 9.16.



Σχήμα 9.16: Εξομοίωση του T MS FF.

**ΑΣΚΗΣΕΙΣ**

- 9.1 Να σχεδιαστεί και να εξομοιωθεί ένας σύγχρονος μανταλωτής JK. Δημιουργήστε συνθήκες προβληματικού χρονισμού δίνοντας στο ρολόι αρκετά μεγάλη περίοδο, ώστε να επιτραπεί στις εξόδους του μανταλωτή να αλλάξουν κατά τη διάρκεια που το ρολόι είναι ενεργό. Αν το ρολόι πρέπει να έχει συνολική περίοδο 8nsec, τι χαρακτηριστικά ( $t_{high}$ ,  $t_{low}$ ) πρέπει να έχει, ώστε ο μανταλωτής να λειτουργεί χωρίς πρόβλημα χρονισμού;  
 Υπόδειξη: ο χρόνος παραμονής του ρολογιού στην ενεργό κατάσταση πρέπει να είναι μικρότερος από την καθυστέρηση διάδοσης του μανταλωτή JK.
- 9.2 Να σχεδιαστεί και να εξομοιωθεί ένας σύγχρονος μανταλωτής T, χρησιμοποιώντας το μανταλωτή JK της Άσκησης 9.1.
- 9.3 Να σχεδιαστεί ένας σύγχρονος μετρητής 3-bit χρησιμοποιώντας μανταλωτές T.

## 10. CMOS Μανδαλωτές και Flip-Flops.

**ΣΚΟΠΟΣ:** Στην άσκηση αυτή θα γίνει μελέτη της υλοποίησης μανταλωτών (latches) και ακμοπυροδοτούμενων (edge-triggered) ακολουθιακών κυκλωμάτων σε CMOS λογική.

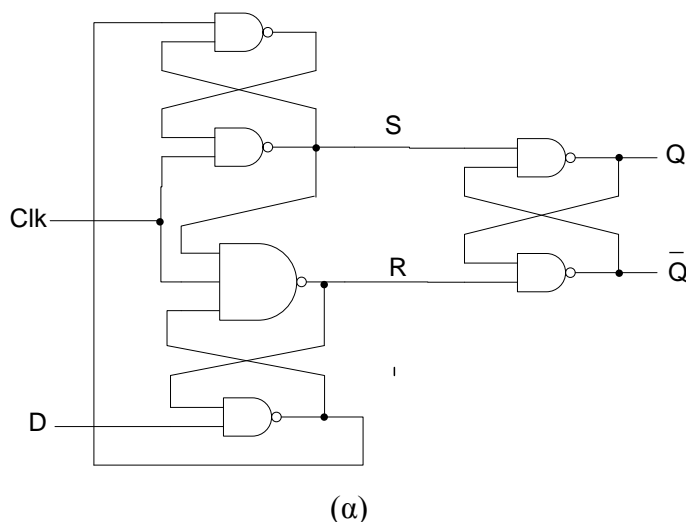
### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

#### Ακμοπυροδοτούμενα FFs.

Η χρησιμότητα των Master-Slave Flip-Flops (όπως είδαμε και στην Εργαστηριακή Άσκηση 9) είναι μεγάλη και αυτό φαίνεται από την ευρεία εφαρμογή τους στα ψηφιακά κυκλώματα. Όμως παρουσιάζουν και αυτά ένα σημαντικό περιορισμό: οι είσοδοί τους είναι "εκτεθειμένες" σε αλλαγές της λογικής τους στάθμης καθ' όλη τη διάρκεια παραμονής του ρολογιού σε ενεργό κατάσταση (πχ παραμονή στο 1). Αυτό μπορεί να δημιουργήσει συνθήκες απροσδιοριστίας πχ σε περίπτωση που κάποια είσοδος υποστεί μια στιγμιαία μεταβολή λόγω θορύβου (είτε αιχμή είτε βύθιση μικρής διάρκειας) στην τιμή της λίγο πριν την αλλαγή της κατάστασης του ρολογιού. Η στιγμιαία αυτή μεταβολή θα "συλληφθεί" από το μανταλωτή αφέντη και θα μεταδοθεί στον μανταλωτή σκλάβο την επόμενη ημιπερίοδο, προκαλώντας την αποθήκευση και μετάδοση λάθος πληροφορίας από το flip-flop.

Για να αποφευχθούν τέτοιες καταστάσεις κατασκευάστηκαν τα ακμοπυροδοτούμενα FFs, τα οποία αποθηκεύουν πληροφορία μόνο κατά τη διάρκεια της μετάβασης του ρολογιού και όχι κατά τη διάρκεια παραμονής του ρολογιού στην ενεργό κατάσταση. Επειδή όμως ο παλμός του ρολογιού έχει ανιούσα (low-to-high) και κατιούσα (high-to-low) παρυφή, υπάρχουν δύο ειδών ακμοπυροδοτούμενα ffs: τα θετικά και τα αρνητικά ακμοπυροδοτούμενα.

Μια υλοποίηση θετικά ακμοπυροδοτούμενου D ff παρουσιάζεται στο Σχήμα 10.1 μαζί με τον πίνακα αληθείας του.

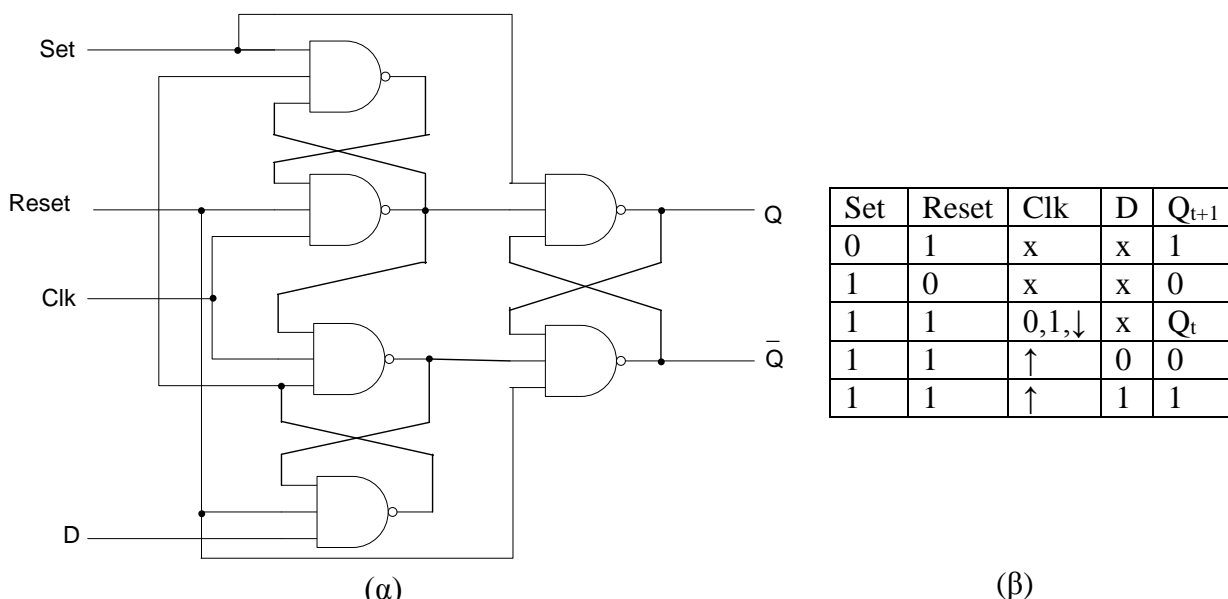


Clk	D	$Q_{t+1}$
0, 1, ↓	x	$Q_t$
↑	0	0
↑	1	1

Σχήμα 10.1: (α) Λογικό διάγραμμα και (β) πίνακας αληθείας του θετικά ακμοπυροδοτούμενου D FF.

Η επιθυμία να υπάρχει άμεσος έλεγχος στα περιεχόμενα των ακολουθιακών κυκλωμάτων, οδήγησε στην ενσωμάτωση δύο επιπλέον ασύγχρονων σημάτων Set και Reset (ή αλλιώς Clear) τα οποία ονομάζονται άμεσες είσοδοι (Direct Inputs). Το όνομα αυτό προκύπτει από τον τρόπο επενέργειας των σημάτων στα περιεχόμενα των στοιχείων μνήμης, ο οποίος είναι άμεσος, ανεξάρτητα από την

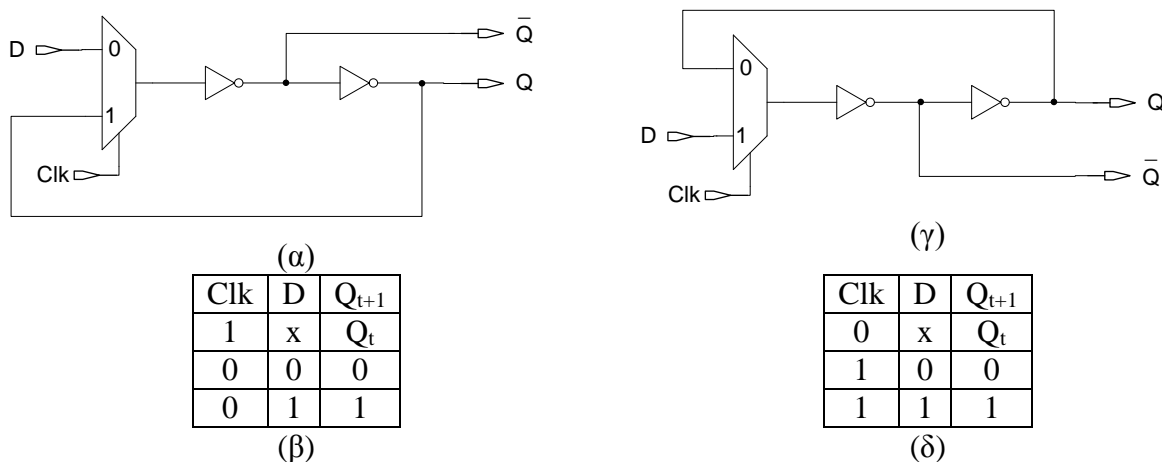
κατάσταση των σύγχρονων εισόδων και του ρολογιού. Η ενσωμάτωση αυτών των εισόδων στο ακμοπυροδοτούμενο D FF οδηγεί στην υλοποίηση του Σχήματος 10.2 η οποία βασίζεται μόνο σε πύλες NAND 3-εισόδων.



Σχήμα 10.2: (α) Λογικό διάγραμμα και (β) πίνακας αληθείας του D ακμοπυροδοτούμενου FF με ασύγχρονες εισόδους Set και Reset.

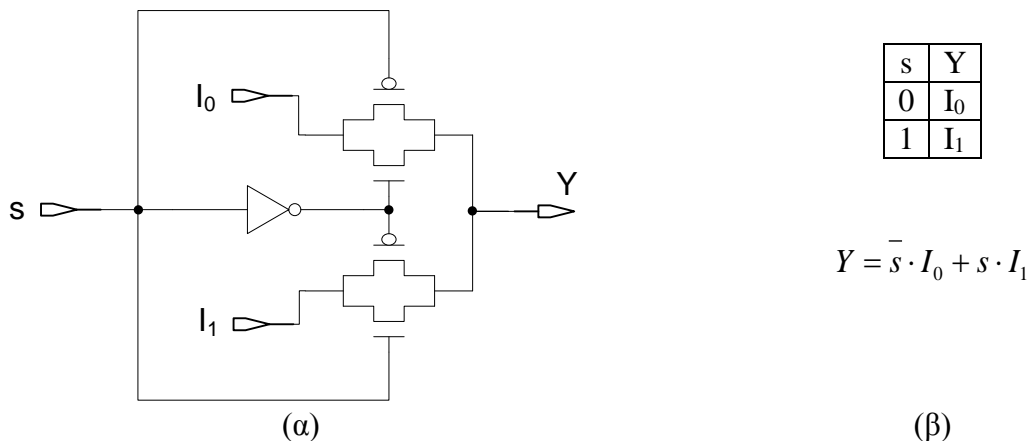
**CMOS σύγχρονοι μανταλωτές και ακμοπυροδοτούμενα FFs.**

Η κατασκευή στοιχείων μνήμης σε CMOS λογική πλεονεκτεί σε σχέση με άλλες υλοποιήσεις καθώς οι απαιτήσεις σε υλικό (αριθμός τρανζίστορ) είναι μειωμένες. Ένας σύγχρονος μανταλωτής D φτιάχνεται όπως φαίνεται στο Σχήμα 10.3, χρησιμοποιώντας ένα πολυπλέκτη 2-1 για την εναλλαγή μεταξύ εισόδου και ανάδρασης. Ανάλογα με το που θα συνδεθούν στον πολυπλέκτη η είσοδος D και η ανάδραση, κατασκευάζονται μανταλωτές ενεργοί στο 1 και ενεργοί στο 0.



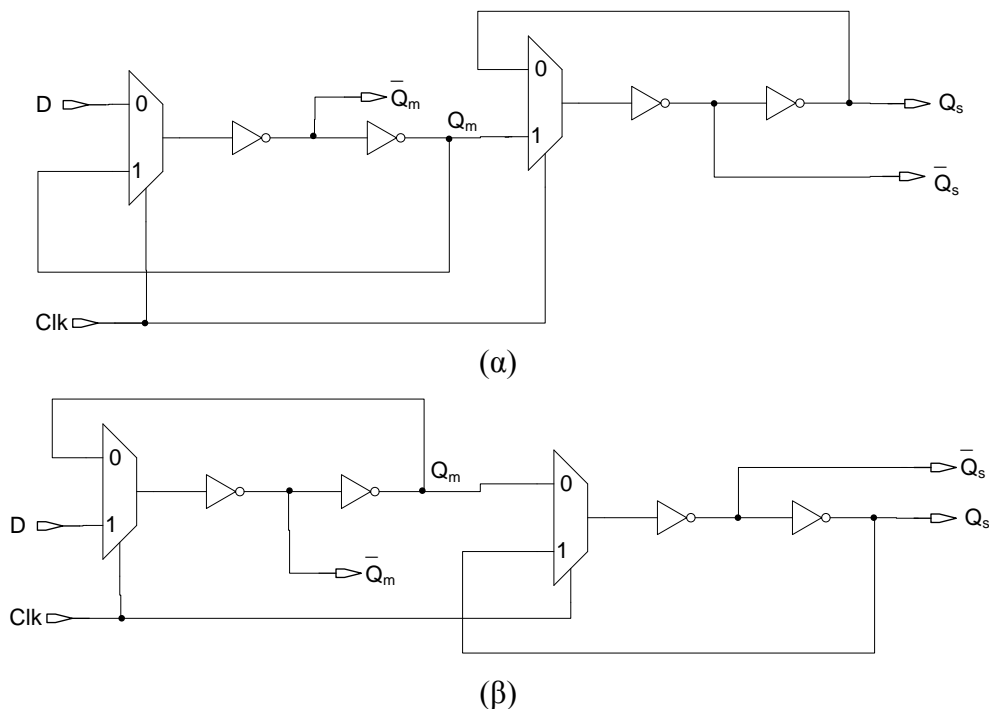
Σχήμα 10.3: (α) Λογικό διάγραμμα και (β) πίνακας αληθείας μανταλωτή D αρνητικής στάθμης. (γ) Λογικό διάγραμμα και (δ) πίνακας αληθείας μανταλωτή D θετικής στάθμης.

Οι πολυπλέκτες του Σχήματος 10.3 μπορούν να υλοποιηθούν με ένα πλήθος τρόπων, όμως η CMOS λογική και ειδικότερα η λογική επίτρευσης (pass transistor logic) παρέχει μια οικονομική υλοποίηση βασισμένη σε πύλες μετάδοσης, η οποία παρουσιάζεται στο Σχήμα 10.4.



Σχήμα 10.4: (α) Λογικό διάγραμμα και (β) πίνακας αληθείας και λογική συνάρτηση του πολυπλέκτη 2-1 σε CMOS λογική.

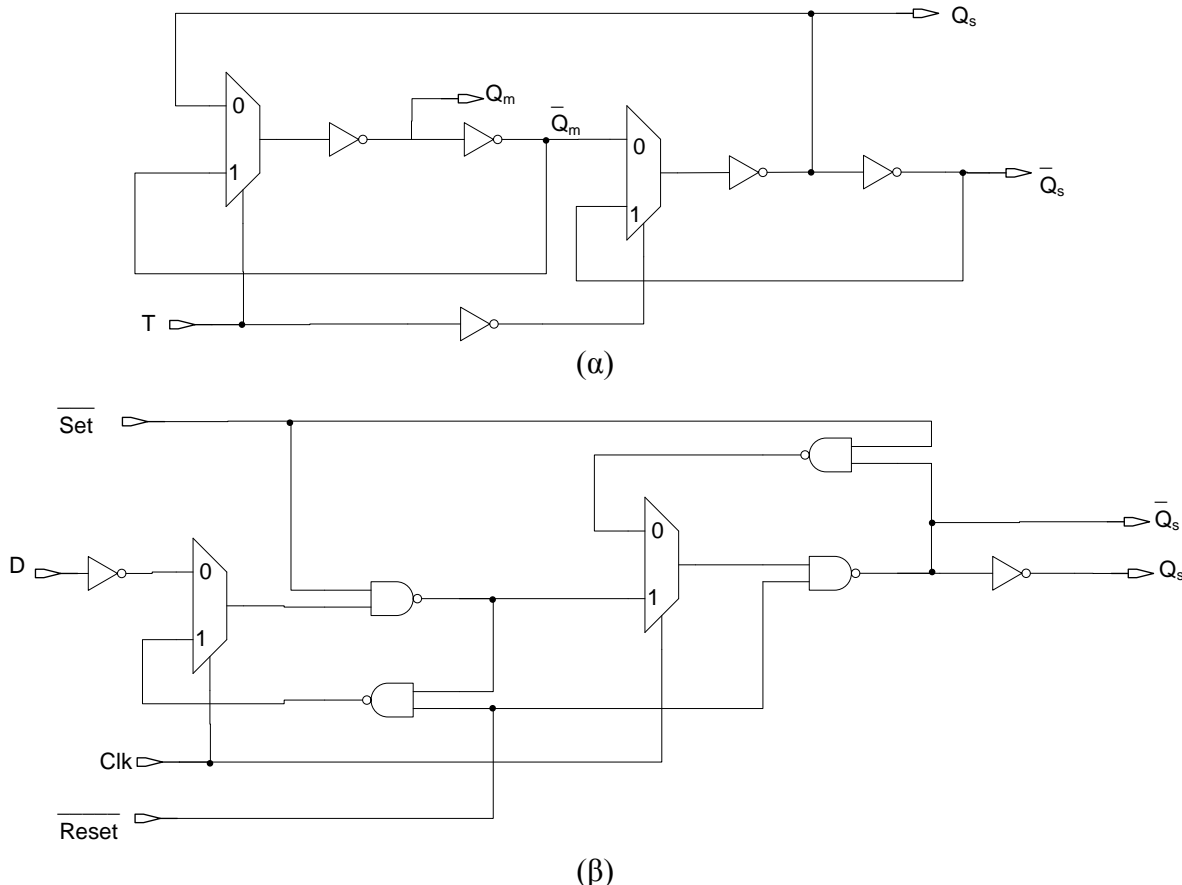
Με βάση τους μανταλωτές που παρουσιάστηκαν προηγουμένως μπορούμε να υλοποιήσουμε τα ακμοπυροδοτούμενα ffs τύπου αφέντη-σκλάβου που φαίνονται στο Σχήμα 10.5.



Σχήμα 10.5: (α) Θετικά (ανιούσα παρυφή ρολογιού) και (β) αρνητικά (κατιούσα παρυφή ρολογιού) ακμοπυροδοτούμενο MS FF.

ΠΑΡΑΤΗΡΗΣΕΙΣ: 1. Ακμοπυροδοτούμενα MS FFs μπορούν να κατασκευαστούν και με τη χρήση όμοιας στάθμης μανταλωτών και για τον αφέντη και για το σκλάβο. Σε αυτή την περίπτωση όμως θα πρέπει οι δύο μανταλωτές να συγχρονίζονται από ρολόγια ανάστροφης φάσης, με τη χρήση ενός αντιστροφέα, ο οποίος θα αναστρέφει το ρολόι πριν αυτό τροφοδοτήσει το σκλάβο μανταλωτή.  
 2. Για την ορθή λειτουργία των ακμοπυροδοτούμενων FFs, πρωταρχικής σημασίας είναι η τήρηση των χρόνων αποκατάστασης (χρόνος πριν την παρυφή του ρολογιού κατά την οποία τα δεδομένα πρέπει να έχουν σταθεροποιηθεί στην είσοδο D) και συγκράτησης (χρόνος μετά την παρυφή του ρολογιού κατά την οποία τα δεδομένα πρέπει να παραμείνουν σταθερά στην είσοδο D). Κάθε παραβίαση ενός εκ των χρόνων αυτών, συνεπάγεται αποθήκευση λανθασμένων δεδομένων στο FF.

Δύο ειδικές περιπτώσεις ακμοπυροδοτούμενων MS FFs είναι το T MS FF και το D MS FF με ασύγχρονα σήματα Set/Reset, τα οποία παρουσιάζονται στο Σχήμα 10.6.



Σχήμα 10.6: (α) CMOS ακμοπυροδοτούμενο T MS FF, (β) CMOS ακμοπυροδοτούμενο D MS FF με άμεσες εισόδους Set και Reset ενεργές στο 0.

Το T χρησιμοποιείται σε μετρητές καθώς σε κάθε περίοδο της εισόδου προκαλείται μια εναλλαγή (toggle) στην έξοδο, διαιρώντας έτσι τη συχνότητα της εισόδου δια δύο. Τέλος, το D MS FF χρησιμοποιείται ευρύτατα ως προσωρινό στοιχείο αποθήκευσης δεδομένων και ως στοιχείο καθυστέρησης.

**ΠΡΑΚΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ**

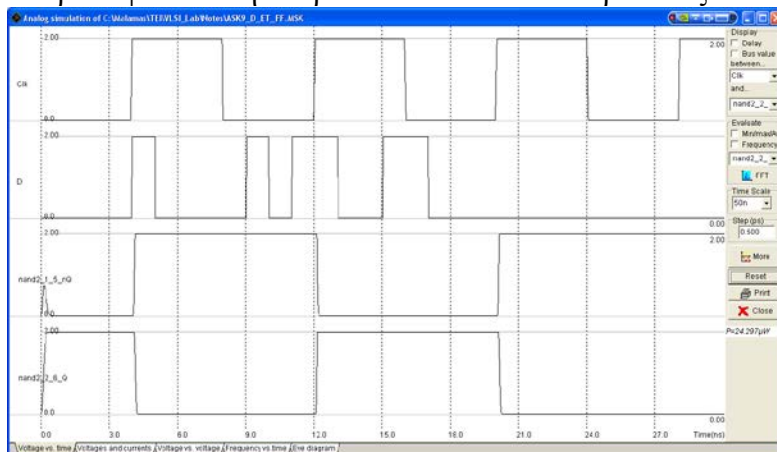
**Ακμοπυροδοτούμενο D FF.**

Σχεδιάστε το κύκλωμα του Σχήματος 10.1 στο Microwind και εξομοιώστε το για την επιβεβαίωση της λειτουργίας του. Για την εξομοίωση ακολουθείστε διαδικασία παρόμοια με την εξομοίωση των κυκλωμάτων της Εργαστηριακής Άσκησης 9, χρησιμοποιώντας γεννήτρια ρολογιού για την είσοδο Clk με χαρακτηριστικά: t<sub>low</sub>/t<sub>rise</sub>/t<sub>high</sub>/t<sub>fall</sub>: 3.950ns/0.05ns/3.950ns/0.05ns.

Για την είσοδο D χρησιμοποιήστε παλμο-γεννήτρια, εισάγοντας τις λογικές τιμές του παρακάτω πίνακα:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
D	0	0	0	1	0	0	0	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0

Η εξομοίωση που θα προκύψει από την παραπάνω διαδικασία παρουσιάζεται στο Σχήμα 10.7.



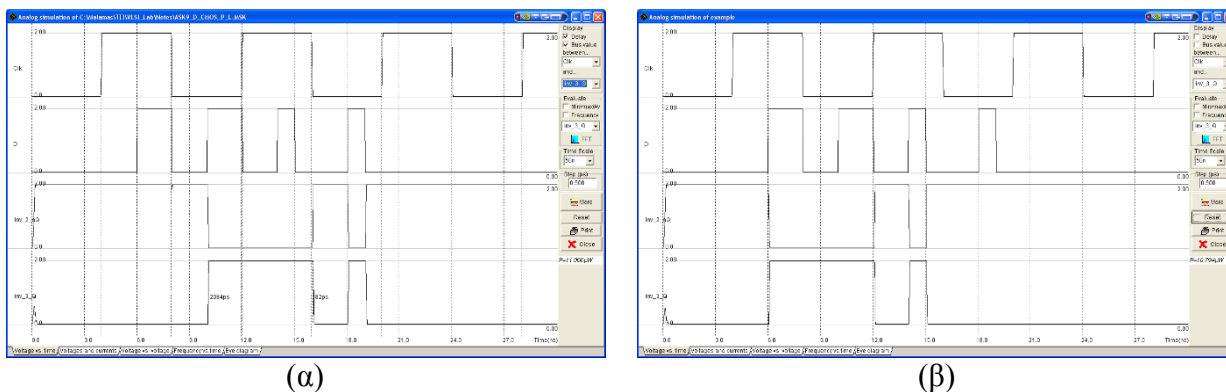
Σχήμα 10.7: Εξομοίωση του θετικά ακμοπυροδοτούμενου D FF με μανταλωτές SR.

**CMOS σύγχρονοι μανταλωτές.**

Να σχεδιαστεί στο Microwind ένας μανταλωτής θετικής στάθμης και ένας μανταλωτής αρνητικής στάθμης. Για την εξομοίωση και των δύο κυκλωμάτων χρησιμοποιήστε γεννήτρια ρολογιού για το σήμα Clk με χαρακτηριστικά:  $t_{low}/t_{rise}/t_{high}/t_{fall}$ : 3.950ns/0.05ns/3.950ns/0.05ns, ενώ για το σήμα εισόδου D χρησιμοποιήστε τον παρακάτω πίνακα λογικών τιμών:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
D	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0

Οι εξομοιώσεις παρουσιάζονται στο Σχήμα 10.8.



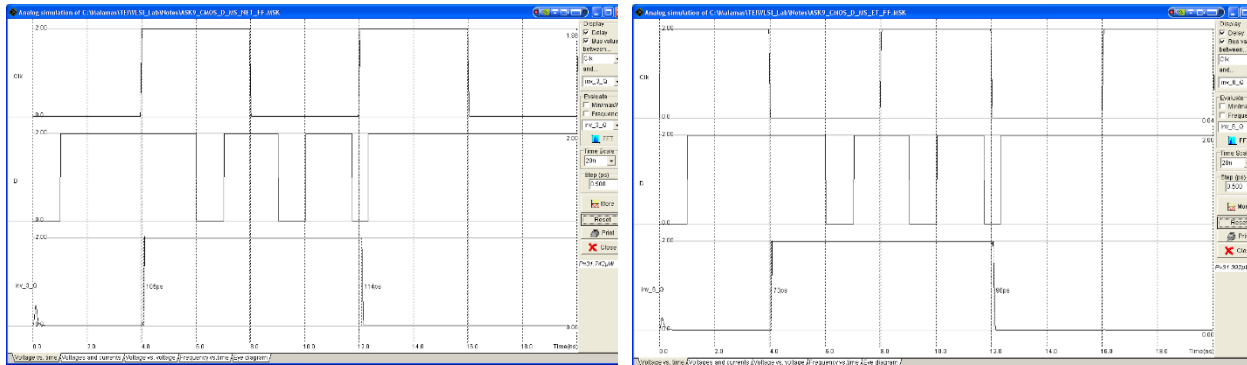
Σχήμα 10.8: (α) Εξομοίωση CMOS μανταλωτή D αρνητικής στάθμης, (β) εξομοίωση CMOS μανταλωτή D θετικής στάθμης.

**Ακμοπυροδοτούμενα MS FFs.**

Για την κατασκευή ενός θετικά ακμοπυροδοτούμενου D MS FF χρησιμοποιήστε ένα μανταλωτή αρνητικής στάθμης για αφέντη και ένα μανταλωτή θετικής στάθμης για σκλάβο. Για αρνητικά ακμοπυροδοτούμενο D MS FF, υλοποιήστε την ανάστροφη διασύνδεση (θετικής στάθμης αφέντης, αρνητικής στάθμης σκλάβος). Για την εξομοίωση του θετικά ακμοπυροδοτούμενου D FF μπορείτε να χρησιμοποιήσετε γεννήτρια ρολογιού για το σήμα Clk, με τα εξής χαρακτηριστικά:  $t_{low}/t_{rise}/t_{high}/t_{fall}$ : 3.950ns/0.05ns/3.950ns/0.05ns, ενώ για το σήμα D παλμο-γεννήτρια με λογικές τιμές που φαίνονται στον παρακάτω πίνακα:

	1	2	3	4	5	6	7	8	9	10	11.7	12.3	13	14	15	16	17	18	19	20
D	1	1	1	1	1	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1

Για την εξομοίωση του αρνητικά ακμοπυροδοτούμενου FF, μπορείτε να χρησιμοποιήσετε για το σήμα Clk γεννήτρια ρολογιού με τα ίδια χαρακτηριστικά αλλά με ανεστραμμένη φάση (κάντε το Level 0 (V) ίσο με 2 και το Level 1 (V) ίσο με 0 στην ετικέτα χαρακτηριστικών γεννήτριας ρολογιού "Add a Clock"). Για το σήμα εισόδου D χρησιμοποιήστε τον παραπάνω πίνακα. Οι εξομοιώσεις που θα προκύψουν θα είναι παρόμοιες με αυτές του Σχήματος 10.9.



(α)

(β)

Σχήμα 10.9: (α) Εξομοίωση θετικά και (β) αρνητικά ακμοπυροδοτούμενου CMOS D MS FF.

## ΑΣΚΗΣΕΙΣ

10.1 Να σχεδιαστεί, να υλοποιηθεί και να εξομοιωθεί στο Microwind ένα ακμοπυροδοτούμενο T MS FF σε CMOS λογική.

10.2 Να σχεδιαστεί, να υλοποιηθεί και να εξομοιωθεί στο Microwind ένας καταχωρητής ολίσθησης 4-bit χρησιμοποιώντας θετικά ακμοπυροδοτούμενα D ffs.

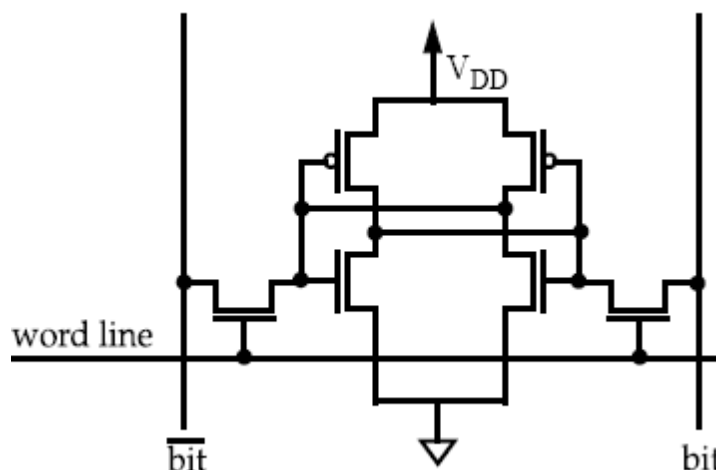
10.3 Να μετρηθούν οι χρόνοι  $t_{set-up}$ ,  $t_{hold}$ , και  $t_{CK1-to-Q}$  σε ένα ακμοπυροδοτούμενο D Flip-Flop (βλ. διαδικασία στο Εργαστήριο).

## 11. Σχεδιασμός και εξομοίωση στοιχείων μνήμης.

**Σκοπός:** Αντικείμενο της άσκησης αυτής είναι ο σχεδιασμός και η εξομοίωση ενός βασικού στοιχείου μνήμης τυχαίας προσπέλασης (Random Access Memory).

### Θεωρητικό υπόβαθρο

Στο παρακάτω σχήμα (Σχήμα 11.1.) δίνεται το κυκλωματικό διάγραμμα ενός στοιχείου μνήμης τυχαίας προσπέλασης (RAM Cell). Στο στοιχείο αυτό χρησιμοποιείται το κύκλωμα του αντιστροφέα διασταυρούμενης σύζευξης (Cross-Coupled). Για την υλοποίηση μιας μνήμης τυχαίας προσπέλασης απαιτείται η οργάνωση σε γραμμές και στήλες πολλών βασικών στοιχείων. Η γραμμή επιλογής ‘Select’ περνά από όλα τα στοιχεία μιας γραμμής, ενώ οι γραμμή ‘Data’ και η συμπληρωματική της περνούν από όλα τα στοιχεία μιας στήλης της μνήμης.



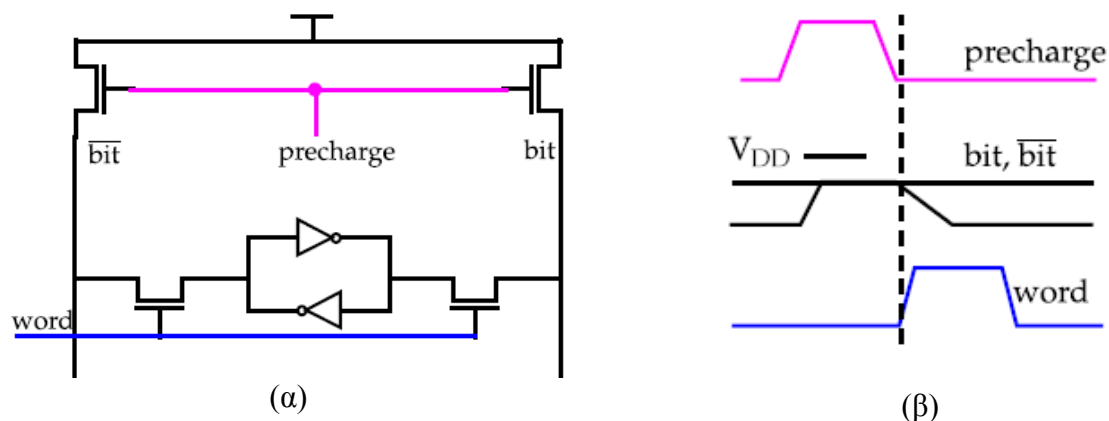
*Σχήμα 11.1.* Κυκλωματικό διάγραμμα ενός στατικού στοιχείου μνήμης.

Οι λειτουργίες ενός βασικού στοιχείου μνήμης είναι η ανάγνωση και η εγγραφή. Για να διαβάσουμε το περιεχόμενο ενός στοιχείου, θέτουμε τη γραμμή επιλογής (word line) στη λογική στάθμη ‘1’. Τότε η τιμή που περιέχεται στο στοιχείο διαδίδεται στη γραμμή bit, και η συμπληρωματική της στη γραμμή  $\overline{\text{bit}}$ . Η διαδικασία της εγγραφής έχει ως εξής: Ας υποθέσουμε ότι στο στοιχείο μνήμης είναι αποθηκευμένο ένα ‘1’ και εμείς θέλουμε να θέσουμε τα περιεχόμενά του σε λογική στάθμη ‘0’, δηλ. να ‘γράψουμε’ στη μνήμη ένα ‘0’. Θέτουμε στη γραμμή bit το ‘0’ και στη συμπληρωματική της το ‘1’. Τότε θέτουμε στη γραμμή επιλογής (word line) την τιμή ‘1’ και μέσω του αντιστροφέα διασταυρούμενης σύζευξης η έξοδος του αριστερού αντιστροφέα παίρνει την τιμή ‘0’, που θέλαμε να ‘γράψουμε’ στη μνήμη. Τέλος, η γραμμή επιλογής απενεργοποιείται.

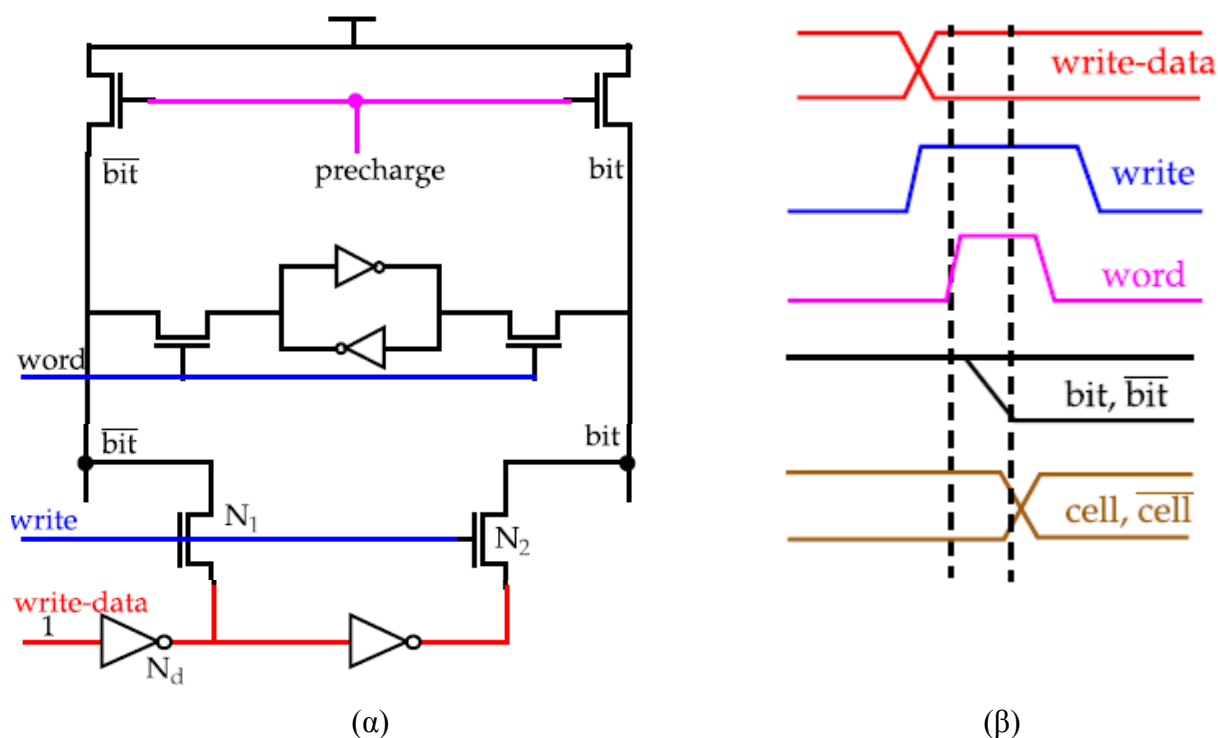
Οι στήλες των μνημών (κοινά σήματα bit και  $\overline{\text{bit}}$ ) αναπτύσσονται με προσθήκη επιπλέον στοιχείων μνήμης κατά την κάθετη έννοια στο Σχήμα 11.1, ενώ οι γραμμές (κοινό σήμα επιλογής word line) αναπτύσσονται κατά την οριζόντια έννοια, με την προσθήκη επιπλέον στοιχείων μνήμης συνδεδεμένα στη γραμμή word line.

### Πρακτική διαδικασία

Η διαδικασία εγγραφής και ανάγνωσης στη μνήμη του Σχήματος 11.1 εξασφαλίζεται με την προσθήκη κυκλώματος προφόρτισης των γραμμών δεδομένων, με χρήση pMOS τρανζίστορ, καθώς τα pMOS τρανζίστορ είναι πιο αργά για ίδιων διαστάσεων κανάλι. Συνεπώς, το κύκλωμα για ανάγνωση φαίνεται στο Σχήμα 11.2, ενώ το πλήρες κύκλωμα του στοιχείου μνήμης το οποίο εξασφαλίζει εγγραφή και ανάγνωση φαίνεται στο Σχήμα 11.3, μαζί με ένα παράδειγμα εγγραφής λογικού "1" όταν η προηγούμενη αποθηκευμένη τιμή είναι "0".



Σχήμα 11.2: (α) Το τμήμα του RAM κυκλώματος για ανάγνωση και (β) αλληλουχία σημάτων για την ανάγνωση.



Σχήμα 11.3: (α) Πλήρες κύκλωμα εγγραφής-ανάγνωσης του στατικού στοιχείου μνήμης, και (β) αλληλουχία σημάτων για εγγραφή.

Για το σχεδιασμό του RAM cell θα χρειαστούν δύο αντιστροφείς συνδεδεμένοι σε ανάδραση (η είσοδος του ενός να τροφοδοτεί την έξοδο του άλλου). Επίσης, θα χρειαστούν δύο αντιστροφείς για την οδήγηση των δεδομένων στο κύτταρο μνήμης. Τέλος, θα χρειαστούν τέσσερα (4) nMOS τρανζίστορ διέλευσης για τον έλεγχο της ροής των δεδομένων από και προς το κύτταρο, καθώς και της κατάστασης λειτουργίας του. **ΠΡΟΣΟΧΗ:** οι διαχύσεις των τεσσάρων nMOS τρανζίστορ

διέλευσης να είναι τουλάχιστον **10λ σε πλάτος** (πλάτος καναλιού), προκειμένου τα τρανζίστορ αυτά να μπορούν να οδηγήσουν σωστά τα κυκλώματα με τα οποία είναι συνδεδεμένα.

Μετά την ολοκλήρωση του φυσικού σχεδίου (layout) του κυττάρου μνήμης στον H/Y, το επόμενο βήμα είναι η εξομοίωση της λειτουργίας του και η πλήρης δοκιμή του ώστε να επαληθευτεί ότι μπορεί να γραφτεί είτε 0 είτε 1 σε αυτό και ότι μπορεί να διαβαστεί η τιμή που έχει αποθηκευτεί. Για την εξομοίωση ακολουθούμε τα χρονοδιαγράμματα των Σχημάτων 11.2 και 11.1 για την εγγραφή και την ανάγνωση αντίστοιχα και κατασκευάζουμε τους ακόλουθους πίνακες λογικών τιμών. Οι πίνακες αυτοί μπορούν να εφαρμοστούν με τη μορφή παλμο-γεννητριών στα αντίστοιχα σήματα εισόδου του κυττάρου μνήμης.

Πίνακας 1: Σήματα και οι τιμές τους για την εγγραφή και ανάγνωση "1" στο RAM cell.

	Εγγραφή "1"							Ανάγνωση "1"						
Χρονική Στιγμή	1	2	3	4	5	6	7	8	9	10	11	12	13	14
word	0	0	0	1	1	0	0	0	0	0	1	1	0	0
write	0	0	1	1	1	0	0	0	0	0	0	0	0	0
write-data	0	0	1	1	1	1	0	0	0	0	0	0	0	0
precharge	0	1	1	0	0	0	0	0	1	1	0	0	0	0

Για την εγγραφή λογικού "0" μπορεί να χρησιμοποιηθεί ο παραπάνω Πίνακας 1 αντικαθιστώντας τα λογικά "1" του σήματος write-data με λογικά "0". Έτσι προκύπτει ο Πίνακας 2.

Πίνακας 2: Σήματα και οι τιμές τους για την εγγραφή και ανάγνωση "0" στο RAM cell.

	Εγγραφή "0"							Ανάγνωση "0"						
Χρονική Στιγμή	1	2	3	4	5	6	7	8	9	10	11	12	13	14
word	0	0	0	1	1	0	0	0	0	0	1	1	0	0
write	0	0	1	1	1	0	0	0	0	0	0	0	0	0
write-data	0	0	0	0	0	0	0	0	0	0	0	0	0	0
precharge	0	1	1	0	0	0	0	0	1	1	0	0	0	0

Να μετρηθεί το χρονικό διάστημα που απαιτείται από τη στιγμή της ενεργοποίησης του σήματος επιλογής (word) μέχρι την εμφάνιση του περιεχομένου της μνήμης στις γραμμές bit και  $\bar{\text{bit}}$ .

### Άσκησης

11.1 Να σχεδιαστεί και να υλοποιηθεί στο Microwind μια στατική μνήμη 3X3 και να ελεγχθεί για την ορθή της λειτουργία ως εξής:

- Γράψτε την ακολουθία "111" στη μεσαία γραμμή.
- Διαβάστε την ακολουθία "111" από τη μεσαία γραμμή.
- Γράψτε την ακολουθία "001" στη δεξιά στήλη.
- Διαβάστε την ακολουθία "001" από τη δεξιά στήλη.

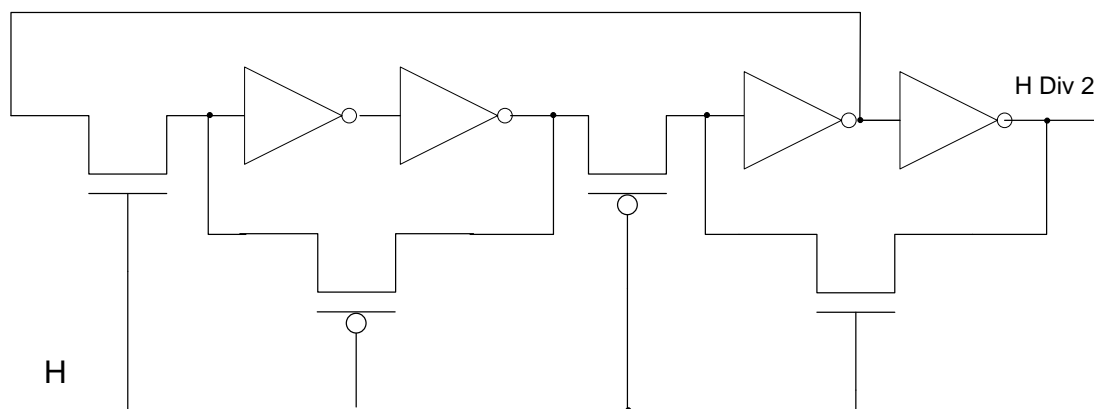
## 12. Σχεδιασμός και εξομοίωση εφαρμογών της CMOS τεχνολογίας.

**ΣΚΟΠΟΣ:** Σχεδιασμός και εξομοίωση τριών εφαρμογών της CMOS τεχνολογίας στα ψηφιακά κυκλώματα, ήτοι του διαιρέτη συχνότητας, της αναγεννητικής ιδιότητας του CMOS αντιστροφέα και τέλος του CMOS ταλαντωτή.

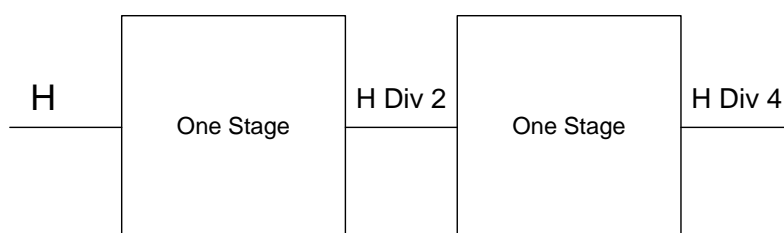
### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

#### Διαιρέτης Συχνότητας.

Η διαίρεση της συχνότητας είναι μια λειτουργία πολύ χρήσιμη στα ψηφιακά συστήματα. Ο διαιρέτης συχνότητας έχει τη δυνατότητα να πολλαπλασιάζει την περίοδο του ρολογιού που εμφανίζεται στην είσοδο του. Στα πλαίσια της άσκησης αυτής θα ασχοληθούμε με το σχεδιασμό ενός διαιρέτη συχνότητας με δύο βαθμίδες. Η κάθε βαθμίδα διαιρεί τη συχνότητα του ρολογιού που εφαρμόζεται στην είσοδο της με το συντελεστή 2. Το κυκλωματικό διάγραμμα μιας βαθμίδας του διαιρέτη δίνεται στο Σχήμα 12.1. Στο Σχήμα 12.2 φαίνεται η χρήση του κυκλώματος του διαιρέτη συχνότητας σε ένα σύστημα με δύο πανομοιότυπες βαθμίδες. Η διασύνδεση των δύο βαθμίδων γίνεται έτσι ώστε το ρολόι εισόδου της δεύτερης βαθμίδας να είναι το σήμα εξόδου της πρώτης, όπως φαίνεται και στο Σχήμα 12.2.



Σχήμα 12.1: Το κύκλωμα του διαιρέτη συχνότητας.



Σχήμα 12.2: Ένας διαιρέτης συχνότητας δύο βαθμίδων.

Από την παρατήρηση του κυκλώματος του Σχήματος 12.1 συμπεραίνουμε ότι η δομή και η λειτουργία του διαιρέτη συχνότητας, βασίζεται στο T Master-Slave flip-flop του Σχήματος 10.6(α), το οποίο επίσης έχει τη δυνατότητα να διαιρεί τη συχνότητα του σήματος εισόδου διά 2. Όσον αφορά στο διαιρέτη του Σχήματος 12.1, παρατηρούμε ότι όταν  $H=0$  τα nMOS τρανζίστορ δεν άγουν, ενώ τα pMOS άγουν, εγκλωβίζοντας έτσι την τρέχουσα τιμή στην ανάδραση του αριστερού συγκροτήματος αντιστροφέων ("master"). Το δεξιό συγκρότημα αντιστροφέων μεταδίδει στην έξοδο "H Div 2" την τρέχουσα λογική τιμή, ταυτόχρονα όμως προετοιμάζει την αλλαγή στη λογική

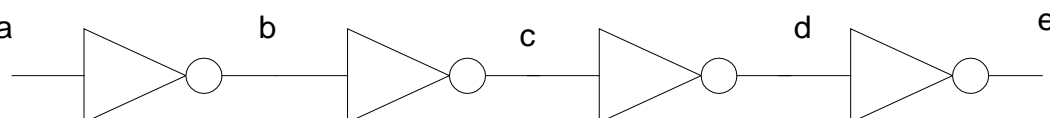
στάθμη του διαιρέτη συχνότητας (λειτουργία toggle), με τη μετάδοση της αντίστροφης λογικής τιμής πίσω στο αριστερό συγκρότημα αντιστροφέων, μέσω της ανάδρασης.

Έτσι, όταν  $H="1"$ , η τρέχουσα λογική τιμή που βρίσκεται στην έξοδο του δεξιού συγκροτήματος συνεχίζει να μεταδίδεται για δεύτερη ημιπερίοδο του σήματος  $H$ , αλλά η αντίστροφη λογική τιμή εισέρχεται στο αριστερό συγκρότημα αντιστροφέων, αλλάζοντας έτσι την αποθηκευμένη τιμή. Όταν γίνει  $H="0"$ , τότε η νέα αυτή τιμή θα εγκλωβιστεί στο αριστερό συγκρότημα αντιστροφέων και θα μεταδοθεί στην έξοδο " $H \text{ Div } 2$ " για δύο ημιπεριόδους του  $H$ . Το αποτέλεσμα αυτής της διαδικασίας είναι ο διπλασιασμός της περιόδου του σήματος εισόδου, ή αλλιώς η διαίρεση δια δύο της συχνότητάς του. Με το κύκλωμα του Σχήματος 12.2 επιτυγχάνεται ο τετραπλασιασμός της περιόδου του σήματος εισόδου, αφού η κάθε βαθμίδα διπλασιάζει την περίοδο του σήματος στην είσοδό του.

### Αναγέννηση σήματος.

Η αναγέννηση σήματος είναι απαραίτητη στα ψηφιακά κυκλώματα στα οποία υπάρχει επίδραση  $H/M$  θορύβου, πχ σε ολοκληρωμένα κυκλώματα συνύπαρξης αναλογικών και ψηφιακών σημάτων (Mixed signal Integrated Circuits) όπως γίνεται στους μετατροπείς Αναλογικό-Σε-Ψηφιακό ή Ψηφιακό-Σε-Αναλογικό. Αναγέννηση επίσης απαιτείται και στις περιπτώσεις κατά τις οποίες χρησιμοποιούνται κυκλώματα λογικής επίτρεψης (pass transistor logic), καθώς τα  $nMOS$  μεταδίδουν φτωχούς άσσους, τα  $pMOS$  μεταδίδουν φτωχά μηδενικά. Επομένως, η αναγέννηση είναι απαραίτητη στην έξοδο τέτοιων κυκλωμάτων, προκειμένου η μετάδοση των λογικών τιμών να γίνει απρόσκοπτα. Ένα τέτοιο παράδειγμα βλέπουμε στο Σχήμα 12.1 όπου τα τρανζίστορ επίτρεψης παρεμβάλλονται στο μονοπάτι δεδομένων (datapath), αλλά τα ζεύγη αντιστροφέων αναγεννούν τα παραγόμενα σήματα.

Η δυνατότητα που έχουν οι CMOS αντιστροφείς να παρέχουν στην έξοδό τους λογικό "1" με  $V_{OHmin}=V_{DD}$  και  $V_{OLmax}=V_{SS}$ , τους δίνει ένα σημαντικό πλεονέκτημα για την αναγέννηση σημάτων στην είσοδό τους. Για την επίτευξη της αναγέννησης συνδέονται ζεύγη αντιστροφέων σε σειρά (προκειμένου να μην αλλοιωθεί η φάση του αρχικού σήματος), όπως φαίνεται και στο Σχήμα 12.3.



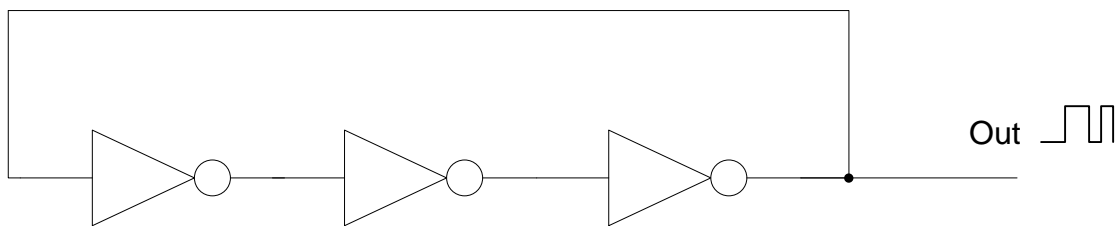
Σχήμα 12.3: Κύκλωμα διασυνδεδεμένων εν σειρά αντιστροφέων για την αναγέννηση του σήματος εισόδου  $a$ .

### CMOS Ταλαντωτές.

Η παραγωγή σημάτων για τη διέγερση ψηφιακών κυκλωμάτων μπορεί να επιτευχθεί με τη χρήση CMOS ταλαντωτών οι οποίοι υλοποιούνται με την εν σειρά σύνδεση μονού αριθμού αντιστροφέων και τη σύνδεση της εξόδου του τελευταίου αντιστροφέα με την είσοδο του πρώτου μέσω ανάδρασης. Η συχνότητα της παραγόμενης κυματομορφής εξαρτάται:

- από την καθυστέρηση διάδοσης του κάθε αντιστροφέα που συμμετέχει στο κύκλωμα (δηλαδή της χρησιμοποιούμενης τεχνολογίας και των διαστάσεων του καναλιού) καθώς επίσης και
- από τον αριθμό των συνδεδεμένων αντιστροφέων.

Στο Σχήμα 12.4 παρουσιάζεται μια τέτοια διάταξη ταλαντωτή με χρήση τριών αντιστροφέων. Παρατηρείστε την παντελή έλλειψη εισόδων στο κύκλωμα, πέραν βέβαια της τροφοδοσίας και της γείωσης. Το εύρος της παραγόμενης κυματομορφής προσδιορίζεται από την τάση τροφοδοσίας των αντιστροφέων  $V_{DD}$ .

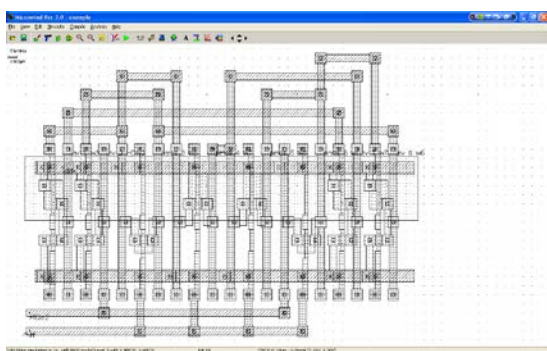


Σχήμα 12.4: Διάταξη ταλαντωτή με CMOS αντιστροφείς.

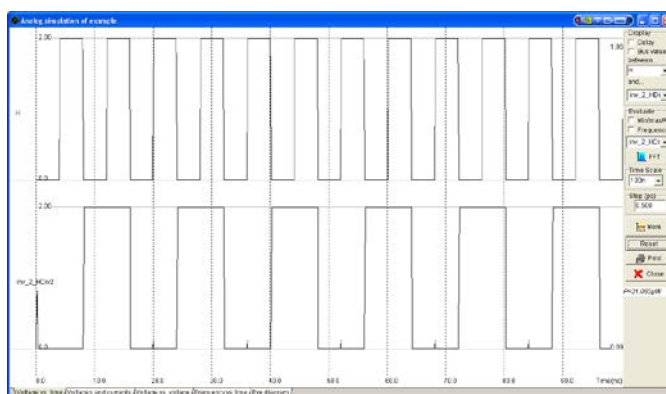
## ΠΡΑΚΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

### Διαιρέτης Συχνότητας.

Σχεδιάστε στο Microwind το κύκλωμα του Σχήματος 12.1. Σχεδιάστε πρώτα έναν αντιστροφέα και κατόπιν αντιγράψτε τον τέσσερις φορές, προσέχοντας ώστε οι γραμμές τροφοδοσίας και γείωσης να είναι ευθυγραμμισμένες. Για το σκοπό αυτό μπορείτε να χρησιμοποιήσετε τη λειτουργία "Edit" -> "Duplicate X Y" του Microwind. Προσέξτε ώστε το μεγαλύτερο μέρος των διασυνδέσεων να υλοποιείται με γραμμές μετάλλου, λόγω της μικρής του ειδικής αντίστασης και χωρητικότητας. Εφαρμόζοντας μια γεννήτρια με χαρακτηριστικά  $t_{low}/t_{rise}/t_{high}/t_{fall}$ : 3.950ns/0.05ns/3.950ns/0.05ns, παίρνετε στην έξοδο την κυματομορφή του Σχήματος 12.5.



(α)

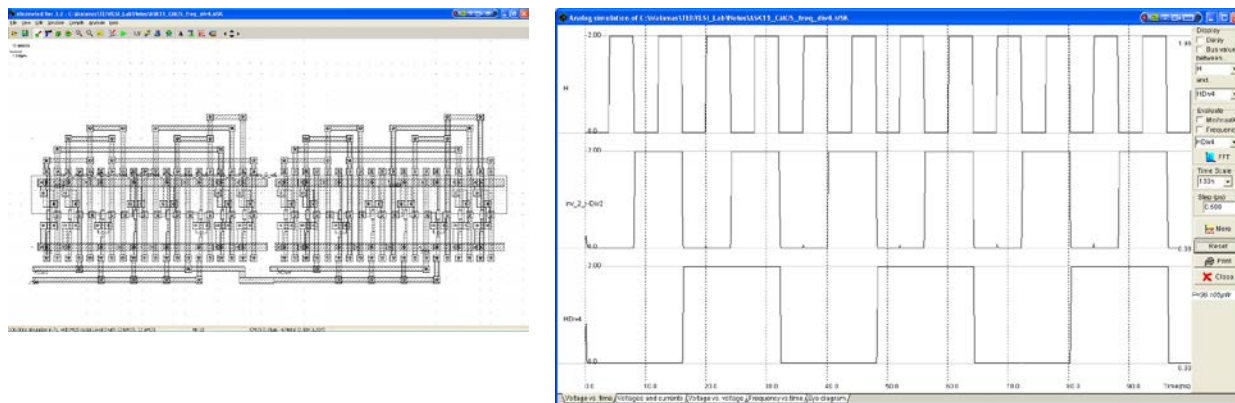


(β)

Σχήμα 12.5: (α) Φυσικό σχέδιο και (β) εξομοίωση του διαιρέτη συχνότητας διά 2.

Για την ολοκλήρωση του σχεδιασμού του διαιρέτη συχνότητας δια 4, αρκεί να υλοποιήσουμε αρχικά τη μια βαθμίδα και έπειτα με τη δυνατότητα αντιγραφής που μας παρέχει το εργαλείο να τοποθετήσουμε τη δεύτερη βαθμίδα, έτσι ώστε η είσοδος της να είναι η έξοδος της προηγούμενης.

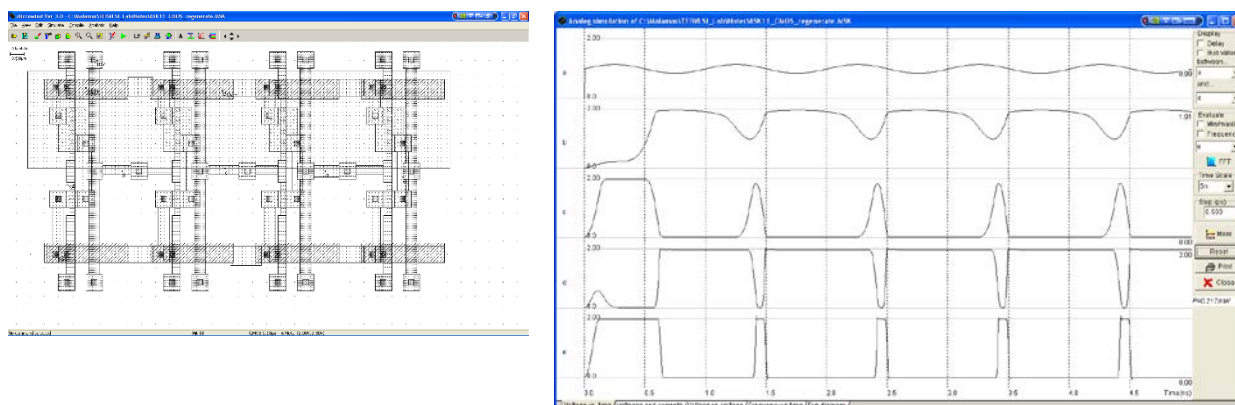
Εφαρμόζοντας ένα συνεχόμενο παλμό ρολογιού στην είσοδο της πρώτης βαθμίδας του διαιρέτη, με τη βοήθεια του μενού εξομοίωσης του εργαλείου παρατηρούμε στα χρονικά διαγράμματα του Σχήματος 12.6 τη διαίρεση συχνότητας που επιτυγχάνεται στην έξοδο κάθε βαθμίδας του κυκλώματος.



(α) Φυσικό σχέδιο και (β) εξομοίωση του διαιρέτη συχνότητας δύο βαθμίδων.

**Αναγέννηση σήματος.**

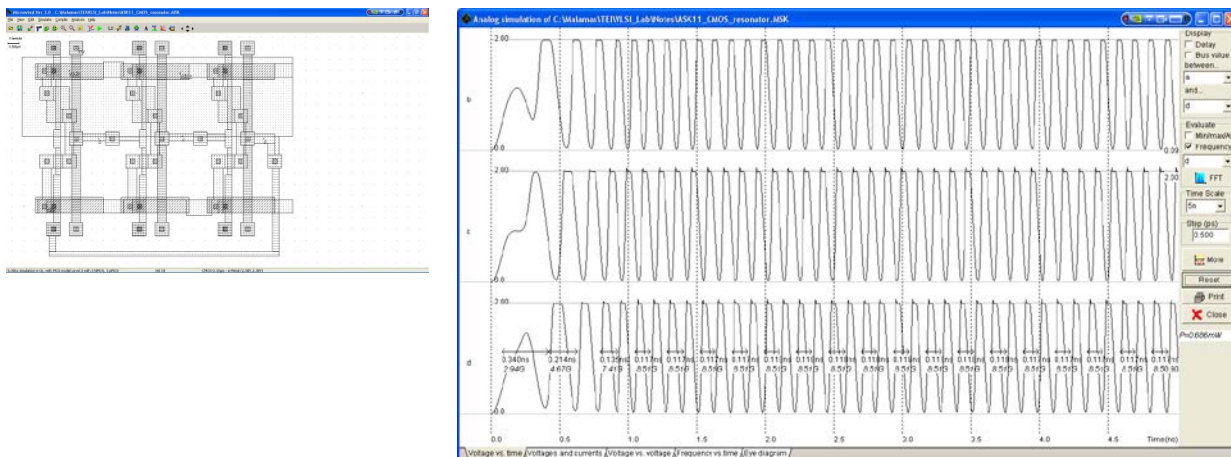
Σχεδιάστε τη διάταξη του Σχήματος 12.3, σχεδιάζοντας ένα αντιστροφέα και εν συνεχεία αντιγράφοντάς τον τρεις φορές (λειτουργία "Duplicate X Y" με X=4, Y=1). Εφαρμόστε γεννήτρια ημιτόνου στην είσοδο a, επιλέγοντας την ταμπέλα "Sinus" με στοιχεία: Amplitude: 0.500 mV, Frequency: 1000.000 MHz, Offset: 1.000V και Increase f: 0.000. Αρχίστε να μειώνετε την παράμετρο Amplitude (πχ δώστε τιμές 400, 250, 150, και 50) και παρατηρήστε τις εξόδους b, c, d, και e στην εξομοίωση. Στο Σχήμα 12.6 παρουσιάζεται το φυσικό σχέδιο και η εξομοίωση για Amplitude 150 του κυκλώματος αναγέννησης.



(α) Φυσικό σχέδιο και (β) Εξομοίωση του κυκλώματος αναγέννησης με 4 αντιστροφείς.

**CMOS ταλαντωτές.**

Σχεδιάστε το φυσικό σχέδιο της διάταξης του Σχήματος 12.4 με αντιστροφείς σε τεχνολογία CMOS 0.18um. Εξομοιώστε το κύκλωμα και μετρήστε τη συχνότητα της παραγόμενης κυματομορφής.



(α)

(β)

Σχήμα 12.7: (α) Φυσικό σχέδιο και (β) εξομοίωση του ταλαντωτή 3 βαθμίδων.

Την παραγόμενη συχνότητα μπορεί να υπολογίσει αυτόματα το Microwind, αν έχετε επιλέξει το Check box "Frequency" στην εξομοίωση. Προσθέστε άλλα δύο στάδια στον ταλαντωτή και υπολογίστε τη νέα συχνότητα που θα παραχθεί. Ποια είναι η σχέση μεταξύ αριθμού συνδεδεμένων βαθμίδων και παραγόμενης συχνότητας;

**ΑΣΚΗΣΕΙΣ**

12.1 Για το κύκλωμα αναγέννησης του Σχήματος 12.3 υπολογίστε το ελάχιστο δυναμικό λογικού "1" που μπορεί να αναγεννηθεί, εφαρμόζοντας γεννήτρια παλμών και μειώνοντας σταδιακά την παράμετρο "Level 1 (V)" της ταμπέλας "Clock". Υπολογίστε το μέγιστο δυναμικό λογικού "0" που μπορεί να αναγεννηθεί, αυξάνοντας σταδιακά την παράμετρο "Level 0 (V)". Η ζητούμενη τιμή είναι αυτή για την οποία το κύκλωμα δε μπορεί να αναγνωρίσει την αντίστοιχη λογική τιμή και η έξοδος της δε μεταβάλλεται.

12.2 Για το κύκλωμα ταλάντωσης του Σχήματος 12.4 χρησιμοποιήστε τεχνολογία CMOS 0.12um και ξανασχεδιάστε το φυσικό σχέδιο του ταλαντωτή στη νέα τεχνολογία. Μετρήστε τη συχνότητα για διάταξη 3 βαθμίδων και διάταξη 5 βαθμίδων.



# Παραρτήματα

## A. Χειρισμός & Εντολές Προγράμματος Microwind2.

### 1. Ξεκινώντας

Για να ξεκινήσετε το πρόγραμμα MICROWIND2 από το CD, χρησιμοποιήστε την ακόλουθη διαδικασία:

- 1 Εισάγετε το συνοδευτικό CD στον οδηγό
- 2 Στα Windows 98/NT/XP, κάντε διπλό κλικ στο [index.html](#)
- 3 Κάντε κλικ στο [“Download Microwind2”](#). Αποσυμπιέστε το αρχείο mw2.zip στον επιθυμητό κατάλογο.
- 4 Κάντε διπλό κλικ στην εικόνα του Microwind2 για να εκκινήσετε το λογισμικό.

Το λογισμικό τρέχει στα λειτουργικά συστήματα Windows XP, 7, 8.x, 10.

#### Παράμετροι γραμμής εντολών

Η γραμμή εντολών μπορεί να περιλάβει δύο παραμέτρους:

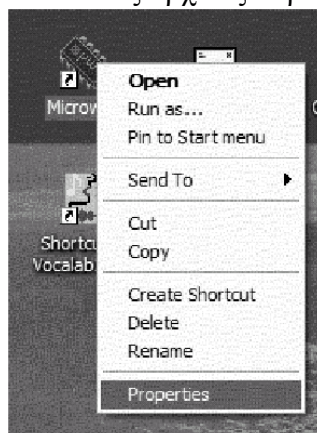
Η Πρώτη παράμετρος είναι το αρχείο της αρχικής μάσκας που φορτώνεται στην αρχικοποίηση.

Η Δεύτερη παράμετρος είναι το αρχείο των κανόνων σχεδιασμού που φορτώνεται κατά την αρχικοποίηση.

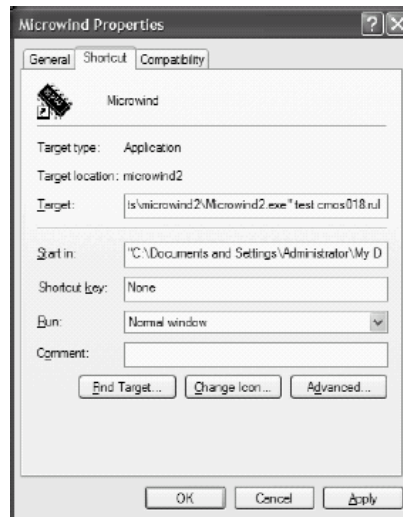
Για παράδειγμα, η εντολή `microwind2 test.MSK cmos018.rul` εκτελεί το MICROWIND2 με αρχείο αρχικής μάσκας το `“test.MSK”` και αρχείο κανόνων το `“cmos018.rul”`.

### Διαμόρφωση της εικόνας του Microwind2

Μπορείτε να προγραμματίσετε το εικονίδιο (την συντόμευση που ευρίσκεται στην επιφάνεια εργασίας) του Microwind2 με το πάτημα του δεξιού κουμπιού, και μετά `“properties”`. Ο αρχικός στόχος δεν περιέχει καμιά παράμετρο. Απλά προσθέστε το αρχικό όνομα αρχείου για το layout και το αρχικό αρχείο κανόνων σχεδιασμού. Στο παρακάτω παράδειγμα, το Microwind2 χρησιμοποιεί το αρχείο `“TEST.MSK”` και το αρχείο κανόνων σχεδιασμού `“CMOS018.RUL”` ως αρχικές παραμέτρους.



Εικόνα 1: Πρόσβαση στις ιδιότητες του εικονιδίου του Microwind2



Εικόνα 2: Διαμορφώνοντας το Microwind2 με αρχικό όνομα αρχείου το “test.MSK” και αρχική τεχνολογία τη “cmos018.RUL”

## 2. Κατάλογος εντολών στο Microwind2

### About Microwind2

Πληροφορίες για την έκδοση του λογισμικού και για τον τρόπο επαφής για υποστήριξη.

#### Add Text to Layout



Χρησιμοποιείστε αυτό το εικονίδιο για να τοποθετήσετε κείμενο σε ένα κουτί ή μια τοποθεσία στο σχέδιο. Εκείνο το κείμενο αναδεικνύει το layout και πρέπει να χρησιμοποιείται όσο το δυνατόν περισσότερο για κάθε σημαντικό κόμβο όπως οι εισοδοί και οι εξοδοί. Για να προσθέσετε κείμενο σε ένα συγκεκριμένο μέρος, προχωρήστε ως ακολούθως:

- ❶ Κάντε κλικ στο εικονίδιο
- ❷ Ορίστε τη θέση του κειμένου με το ποντίκι. Ένα κουτί διαλόγου εμφανίζεται
- ❸ Εισαγάγετε το κείμενο μπροστά από το “Label name:” και πατήστε “Assign”. Το κείμενο τοποθετείται στο σχέδιο.

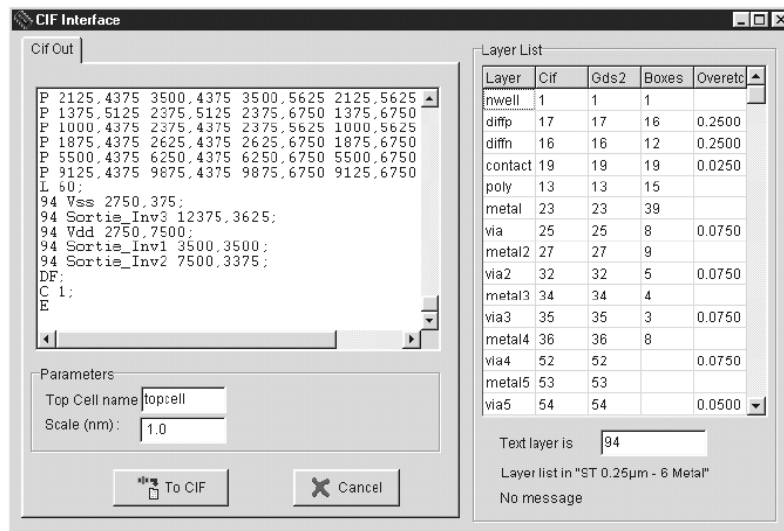
Ένα κείμενο μπορεί να τροποποιηθεί ως εξής: κάντε κλικ στο εικονίδιο, κάντε κλικ μέσα στο υπάρχον κείμενο. Το αρχικό κείμενο εμφανίζεται. Τροποποιήστε το και κάντε κλικ στο “Assign”. Μπορείτε να προσθέσετε ένα ρολόι, έναν παλμό, μια πηγή τάσης VDD ή VSS στο κείμενο.

### Convert into

Το MICROWIND2 μετατρέπει το MSK layout σε CIF χρησιμοποιώντας μια ειδική διεπαφή, η οποία καλείται με “File -> Make CIF file”. Το CIF αρχείο μπορεί να εξαχθεί σε λογισμικό CAD VLSI. Ο δεξιός πίνακας της οθόνης (Εικόνα 3) δίνει την αντιστοιχία μεταξύ επιπέδων MICROWIND2 και επιπέδων CIF, τον αριθμό των κουτιών στο layout και το αντίστοιχο over-etch. Το over-etch χρησιμοποιείται για να τροποποιήσει το τελικό μέγεθος των κουτιών CIF προκειμένου να χωρέσουν στους ακριβείς κανόνες σχεδιασμού.

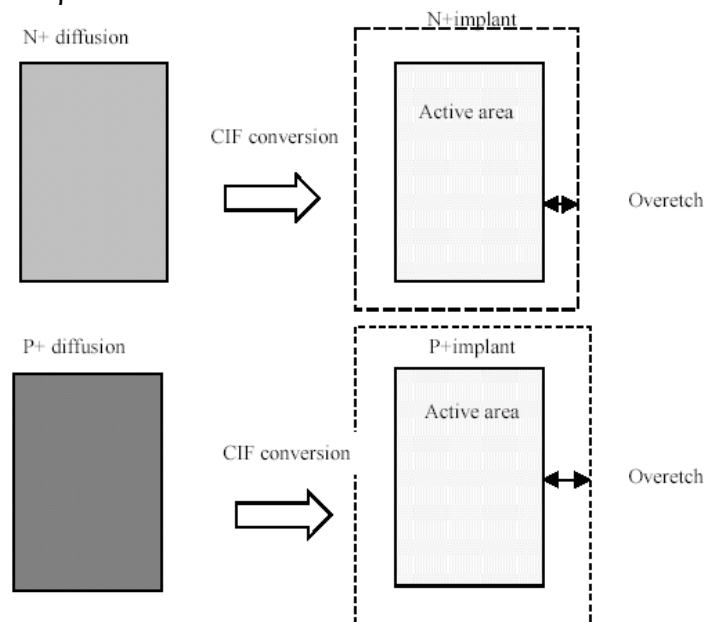
Κάντε κλικ στο “Convert to CIF” για να ξεκινήσετε τη μετατροπή. Ορισμένα μέρη του αποτελέσματος εμφανίζονται στο αριστερό παράθυρο. Η κύρια μονάδα είναι το 1nm. Μπορείτε να την αλλάξετε για να ικανοποιηθούν οι απαιτήσεις του στοχευόμενου εργαλείου CAD.

Για αρχείο κανόνων CMOS 0.25μm (cmos025.RUL), παρατηρείστε το over-etch που εφαρμόζεται στην επαφή και το via. Αυτό το over-etch είναι επιτακτικό να υπακούει στους τελικούς κανόνες σχεδιασμού, ενώ ικανοποιείται ο φιλικός-στο-χρήστη και ο φορητός βασιζόμενος-στο-λάμδα σχεδιασμός.



Εικόνα 3: Η οθόνη παραγωγής του CIF

Όσον αφορά στις διαχύσεις, παρατηρείστε ότι η γεννήτρια CIF παράγει ενεργές περιοχές και εμφυτεύσεις. Το Microwind2 χρησιμοποιεί απλή n+διάχυση και p+διάχυση, ενώ τα περισσότερα βιομηχανικά εργαλεία σχεδιασμού layout χρησιμοποιούν την ιδέα της ενεργούς περιοχής περιστοιχισμένης από εμφυτεύσεις, είτε n+ είτε p+, όπως φαίνεται παρακάτω.



Εικόνα 4: Η μετατροπή CIF παράγει ενεργές περιοχές και εμφυτεύσεις, ώστε να είναι συμβατές με βιομηχανική διεργασία

Αυτό σημαίνει ότι κάθε κουτί n+ διάχυσης που σχεδιάζεται στο Microwind μετατρέπεται σε δύο κουτιά, ένα συνδεδεμένο με “Active Area” (Ενεργό Περιοχή), με ένα κωδικό όνομα δηλωμένο στο αρχείο κανόνων σχεδιασμού, ένα δεύτερο κουτί συνδεδεμένο με το “diffn”, με ένα δοσμένο overetch. Κάθε κουτί p+

διάχυσης μετατρέπεται σε δύο κουτιά επίσης. Ένα είναι συνδεδεμένο με την ίδια “Active Area”, και ένα δεύτερο κουτί είναι συνδεδεμένο με το “diffp” με ένα δοσμένο overetch.

## Colors

- Μετατροπή σε μονόχρωμο: το φυσικό σχέδιο (layout) σχεδιάζεται σε μαύρο και άσπρο. Αυτός ο τύπος σχεδιασμού είναι βολικός για την κατασκευή μονόχρωμης τεκμηρίωσης. Πατήστε “Alt”+ “Print Screen” για να αντιγράψετε την οθόνη στο clipboard. Μετά, ανοίξτε “Word”, κάντε κλικ “Edit->Paste”. Η οθόνη εισάγεται στο έγγραφο.
- Άσπρο παρασκήνιο. Τα επίπεδα εμφανίζονται με μια παλέτα χρωμάτων πάνω σε άσπρο παρασκήνιο.

### Copy



Κάντε κλικ στο εικονίδιο Copy. Μετακινήστε τον δρομέα (cursor) στο παράθυρο σχεδιασμού και οριοθετήστε την ενεργό περιοχή με το ποντίκι. Ακολούθως, όλα τα γραφικά που περιλαμβάνονται σε αυτήν την περιοχή αντιγράφονται. Το εξωτερικό περίγραμμα των αντιγραμμένων αντικειμένων εμφανίζεται. Τοποθετήστε εκείνα τα αντιγραμμένα αντικείμενα στην επιθυμητή θέση με ένα κλικ στο ποντίκι.

Κάντε κλικ στο Undo για να ακυρώσετε την εντολή αντιγραφής.

**ΠΡΟΣΟΧΗ!!** Η διαδικασία της αναίρεσης στο MICROWIND λειτουργεί δυστυχώς **μόνο** για την τελευταία ενέργεια.

### Cut

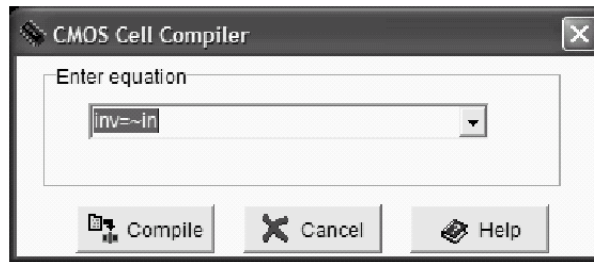


Κάντε κλικ στο εικονίδιο Cut. Μετακινήστε τον δρομέα (cursor) στο παράθυρο σχεδιασμού και περιορίστε την ενεργό περιοχή με το ποντίκι. Ακολούθως, όλα τα γραφικά που περιλαμβάνονται σε αυτή την περιοχή διαγράφονται. Κάντε κλικ στο Undo αν επιθυμείτε να τα επαναφέρετε πίσω στο σχέδιο.

- Ένα επίπεδο προστατεύεται από σβήσιμο αν αφαιρέσετε το tik στην παλέτα δύο φορές. Στην παλέτα ένα άδειο τετράγωνο στα δεξιά του επιπέδου υποδεικνύει ένα προστατευμένο επίπεδο.
- Ένα επίπεδο είναι απροστάτευτο στο σβήσιμο αν επιλέξετε ξανά στην παλέτα. Ένα tik στο τετράγωνο στα δεξιά του επιπέδου υποδεικνύει ένα απροστάτευτο επίπεδο.
- Ένα κουτί μόνο μπορεί να σβηστεί με ένα κλικ μέσα σε εκείνο το κουτί όταν η εντολή αποκοπής είναι ενεργή. Το κουτί τότε σβήνεται.

## Compile one Line

Ο μεταγλωττιστής κελιών είναι ένα συγκεκριμένο εργαλείο σχεδιασμένο για την αυτόματη δημιουργία κελιών CMOS από μια λογική περιγραφή. Κάντε κλικ στο Compile -> Compile One Line. Το παρακάτω μενού εμφανίζεται (Εικόνα 5). Η αρχική εξίσωση αντιστοιχεί σε μια πύλη NOR τριών εισόδων. Αν απαιτείται, μπορεί κανείς να χρησιμοποιήσει το πληκτρολόγιο προκειμένου να τροποποιήσει την εξίσωση και τότε να κάνει κλικ στο Compile. Η πύλη μεταγλωττίζεται και το αντίστοιχο layout παράγεται.



Εικόνα 5: Το παράθυρο του μεταγωγτιστή κελιών.

- Το πρώτο μέρος της σύνταξης μιας-γραμμής αντιστοιχεί στο όνομα της εξόδου.
- Το τελευταίο ακολουθείται από το σύμβολο “=”, από τον κατάλογο των ονομάτων εισόδου διαχωρισμένα από τελεστές AND ‘&’, OR ‘|’, XOR ‘^’, NOT ‘~’, XNOR ‘~^’. Αν χρειάζεται, μπορούν να προστεθούν παρενθέσεις.
- Τα ονόματα των εισόδων και των εξόδων είναι φράσεις των 8 χαρακτήρων κατά μέγιστο.

Cell	Formula
Inverter	out=/in
NAND gate	n=/(a.b)
3 Input OR	s=a+b+c
3 Input NAND	out=/(a.b.c)
AND-OR Gate	cgate=a.(b+c)
CARRY Cell	cout=(a.b)+(cin.(a+b))

Πίνακας 1: Παραδείγματα περιγραφών λογικών κελιών

Τα τρανζίστορ p-καναλιού τοποθετούνται στο πάνω μέρος του δικτύου των τρανζίστορ n-καναλιού. Αν κάποιο layout υπάρχει ήδη κοντά σε εκείνα τα εικονίδια, η αρχή του κελιού μετακινείται προς τα δεξιά μέχρι να βρεθεί αρκετός ελεύθερος χώρος. Αν ο τελεστής NOT (Σύμβολο ‘~’) δεν έχει προσδιοριστεί μετά το σύμβολο ‘=’, ένας αντιστροφέας (INVERTER) προστίθεται στο δεξιό μέρος του μεταγωγτισμένου κελιού. Αυτός είναι ο λόγος για τον οποίο μια πύλη AND μεταγωγτίζεται ως μια πύλη NAND ακολουθούμενη από έναν αντιστροφέα.

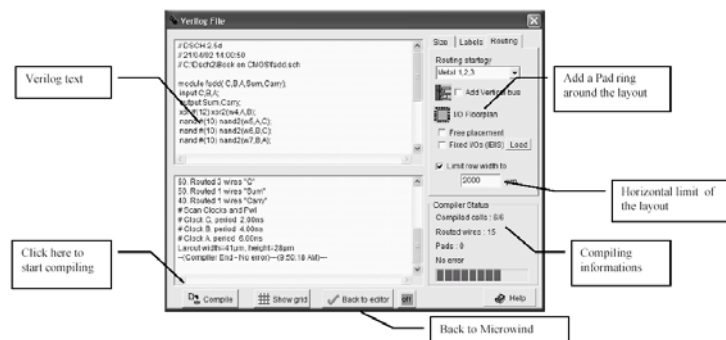
## Compile VERILOG file

Ο μεταγωγτιστής κελιών μπορεί να χειριστεί αυτόματα την παραγωγή φυσικού σχεδίου (layout) από ένα αρχέγονα-βασισμένο κείμενο περιγραφής σε VERILOG. Κάντε κλικ στο Compile -> Compile Verilog File. Επιλέξτε ένα αρχείο κειμένου VERILOG και κάντε κλικ στο “Generate”. Για παράδειγμα, ο κατάλογος του microwind2 περιέχει το αρχείο “FADD.TXT”, το οποίο αντιστοιχεί στην περιγραφή ενός πλήρους αθροιστή (Full Adder).

```
// DSCH 2.5d
// 21/04/02 14:00:50
// C:\Dsch2\Book on CMOS\fadd.sch

module fadd( C,B,A,Sum,Carry);
input C,B,A;
output Sum,Carry;
xor #(12) xor2(w4,A,B);
nand #(10) nand2(w5,A,C);
nand #(10) nand2(w6,B,C);
nand #(10) nand2(w7,B,A);
xor #(12) xor2(Sum,w4,C);
nand #(10) nand3(Carry,w7,w6,w5);
endmodule

// Simulation parameters
// C CLK 10 10
// B CLK 20 20
// A CLK 30 30
```

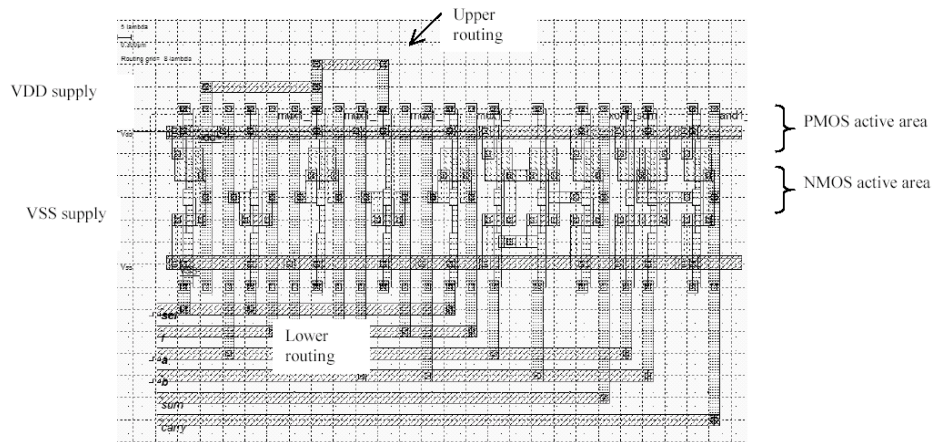


Εικόνα 6: Το παράθυρο του μεταγωγτιστή VERILOG

PRIMITIVE	NODES	EXAMPLE
dreg	Inputs : Data, RESET, CLOCK Outputs: Q, nQ	dreg reg1(d, rst, h, q, nq);
Inv, not	Inputs : IN Outputs: OUT	inv inv1(s,e); // both 'inv' and 'not' not inv1(s,e); // can be used
and	Inputs : 2 to 4 Outputs: S	and and1(s,a,b,c,d); // limit inputs to 4
nand	Inputs : 2 to 4 Outputs: S	nand nand1(s,a,b,c,d);
or	Inputs : 2 to 4 Outputs: S	or or3(s,a,b,c);
nor	Inputs : 2 to 4 Outputs: S	nor my_nor4(s,a,b,c,d);
xor	Inputs : a,b Outputs: S	xor xor_gate(xor_out,d0,d1);
Nmos	Inputs: gate, source Outputs: drain	nmos nmos1(d,s,g);

Εικόνα 7: Τα αρχέγονα κελιά σε VERILOG που υποστηρίζονται από τον μεταγλωττιστή CMOS

Οι κόμβοι εισόδου/εξόδου δρομολογούνται στο πάνω και στο κάτω μέρος των ενεργών μερών, με ένα κανονικό διάκενο για να διευκολυνθεί η αυτόματη δρομολόγηση καναλιών μεταξύ κελιών. Κάντε κλικ σε **Compile -> Show grid** για να υπερθέσετε το πλέγμα (grid) δρομολόγησης πάνω στο layout. Στην εικόνα 8 παρουσιάζεται η δρομολόγηση εισόδων/εξόδων μεταξύ βασικών κελιών. Παρατηρείστε ότι η δρομολόγηση πραγματοποιείται είτε στο πάνω μέρος είτε στο κάτω μέρος των ενεργών μερών.

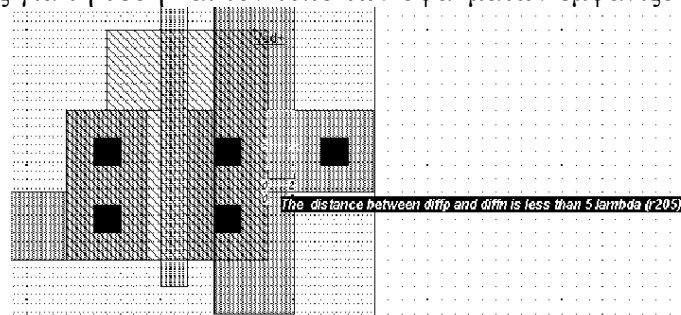


Εικόνα 8: Το πλέγμα (grid) του μεταγλωττιστή κελιών (CMOS cell Compiler)

## Design Rule Checker



Ο ελεγκτής κανόνων σχεδιασμού (Design Rule Checker-DRC) σαρώνει όλο το σχέδιο και εντοπίζει τα σημεία του φυσικού σχεδίου (layout) στα οποία παραβιάζονται βασικοί κανόνες σχεδιασμού. Κάντε κλικ στο παραπάνω εικονίδιο ή στο Analysis -> Design Rule Checker για να τρέξετε το DRC. Τα σφάλματα επισημαίνονται στο παράθυρο παρουσίασης, με ένα κατάλληλο μήνυμα που περιγράφει τη φύση του σφάλματος. Λεπτομέρειες για τη θέση και τον τύπο των σφαλμάτων εμφανίζονται στην οθόνη.



Εικόνα 9: Παράδειγμα σφάλματος κανόνων σχεδιασμού

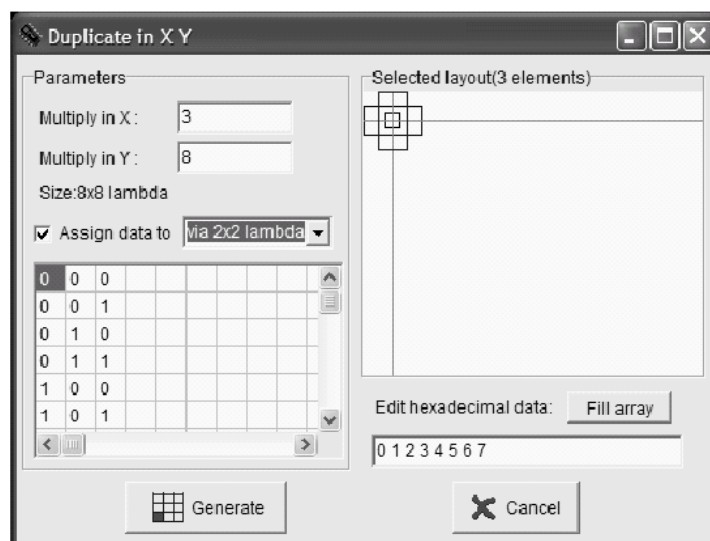
**Draw a Box**

Το εικονίδιο “Draw Box” είναι το αρχικό εικονίδιο (που επιλέγεται ως ιδιότητα στη χρήση του ποντικιού). Δημιουργεί ένα κουτί στο επιλεγμένο επίπεδο. Το αρχικό επίπεδο είναι πολύ-πυρίτιο. Αν το εικονίδιο “Draw Box” δεν είναι επιλεγμένο, κάντε κλικ πάνω του. Μετά, μετακινήστε τον δρομέα στο παράθυρο παρουσίασης και τοποθετήστε την πρώτη ακμή του κουτιού με το πάτημα του ποντικιού. Κρατείστε πατημένο και τραβήξτε το ποντίκι στην απέναντι ακμή του κουτιού. Ελευθερώστε το ποντίκι και δείτε πώς δημιουργείται το κουτί.

- Το ενεργό επίπεδο επιλέγεται στην παλέτα.
- Το κόκκινο χρώμα υποδεικνύει το ενεργό επίπεδο.
- Το γκρι κλειδί στα δεξιά του κουμπιού του επιπέδου προσδιορίζει ότι όλα τα κουτιά που χρησιμοποιούν το επίπεδο αυτό μπορούν να σβηστούν, επιμηκυνθούν, ή να αντιγραφούν.
- Ένα κλικ στο γκρι κλειδί μετατρέπει το χρώμα του κλειδιού σε κόκκινο. Ένα κόκκινο κλειδί προστατεύει το επίπεδο.

**Duplicate XY**

Η εντολή “Duplicate XY” είναι πολύ χρήσιμη για την παραγωγή μιας διάταξης πανομοιότυπων κελιών όπως κελιά RAM για παράδειγμα. Κάντε κλικ στο Edit -> Duplicate XY, περιλάβετε τα στοιχεία για διπλασιασμό σε μια περιοχή που ορίζεται από το ποντίκι και εμφανίζεται η οθόνη που φαίνεται στην εικόνα 10. Τόσο στο X όσο και στο Y, ο αρχικός παράγων πολλαπλασιασμού είναι x2. Μπορείτε να προσαρμόσετε το κενό μεταξύ των κελιών. Εξ αρχής, τα κελιά βρίσκονται σε επαφή μεταξύ τους. Τα επιλεγμένα κουτιά εμφανίζονται στο δεξιό παράθυρο και επίσης με κίτρινο στο κύριο παράθυρο layout.

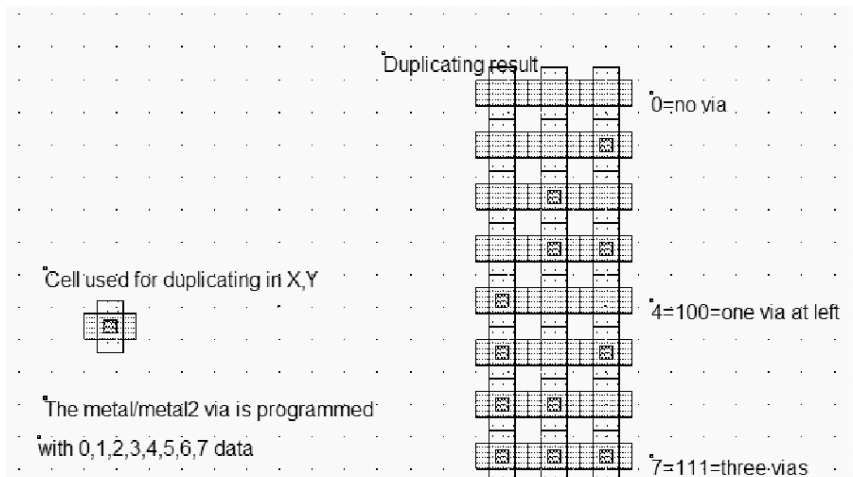


Εικόνα 10: Το μενού Duplicate X, Y, χρησιμοποιείται για τη διασταύρωση metal1/metal2

Η επιλογή data (Assign data, Edit Hexadecimal data, Fill Array) είναι πολύ χρήσιμη για την παραγωγή μασκών ROM ή διατάξεων αποκωδικοποιητών. Ένα παράδειγμα μιας διάταξης αποκωδικοποιητή δίδεται στην εικόνα 10. Ο προγραμματισμός επηρεάζει το via μεταξύ metal1 και metal2, σύμφωνα με τη λίστα των δεκαεξαδικών τιμών που δίδονται στη λίστα επεξεργασίας.

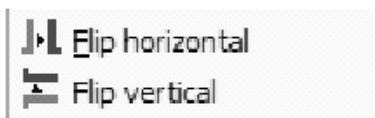
- Κάντε κλικ στα επιθυμητά δεδομένα στην περιοχή επεξεργασίας “Edit hexadecimal data”. Οι τιμές πρέπει να διαχωρίζονται με ένα κενό.

- Κάντε κλικ στο “Fill array” για να μετατρέψετε αυτά τα δεδομένα σε Boolean τιμές, σύμφωνα με το μέγεθος της διάταξης X, Y.
- Επιλέξτε το κατάλληλο κουτί για προγραμματισμό. Σε αυτήν την περίπτωση, το 0/1 δημιουργεί ή όχι ένα via σε κάθε θέση X, Y.
- Κάντε κλικ στο “Generate”. Το ακόλουθο αποτέλεσμα εμφανίζεται.



Εικόνα 11: Παράδειγμα διπλασιασμού ενός προτύπου με προγραμματιζόμενο via (DuplicateXYExample.MSK)

## Flip



Για να εφαρμόσετε μια περιστροφή ή αντιστροφή σε ένα μέρος του σχεδίου, κάντε κλικ στο “Edit -> Flip and Rotate”, και επιλέξτε την κατάλληλη εντολή Flip (Horizontal ή Vertical). Περιορίστε την ενεργό περιοχή των κουτιών στο layout που θα τροποποιηθεί.

## Help

Παρέχει μια on-line βοήθεια στη χρήση του Microwind2. Περιλαμβάνει μια περίληψη των εντολών και μερικές λεπτομέρειες για τους κανόνες σχεδιασμού, όπως φαίνεται στην εικόνα 12.

Layer	Width	Spacing	Surface	Surf capa	Lin capa	Clk capa	Res	Thickn	Height	Permitt
	lambda	lambda	lambda2	af/µm2	al/µm	af/µm	ohm	µm	µm	
nitride	0	0	0							
passiv	1330	1330	0							
metal6	8	8	144	100.00		50.00	0.05/sq	0.70	6.60	3.10
via5	5	5	0				1.00/via	0.50	6.10	4.00
metal5	8	8	100	120.00		50.00	0.05/sq	0.70	5.40	3.10
via4	2	4	0				1.00/via	0.50	4.70	4.00
metal4	3	4	16	140.00		50.00	0.06/sq	0.50	4.20	3.10
via3	2	4	0				2.00/via	0.50	3.70	4.00
metal3	3	4	16	160.00		50.00	0.06/sq	0.50	3.20	3.10
via2	2	4	0				2.00/via	0.50	2.70	4.00
metal2	3	4	16	180.00		50.00	0.06/sq	0.50	2.20	3.10
via	2	4	0				2.00/via	0.50	1.70	4.00
metal	3	4	16	200.00		30.00	0.06/sq	0.50	1.20	3.10
poly	2	3	16	400.00			4.00/sq	0.20	0.01	4.00

Εικόνα 12: Κανόνες σχεδιασμού και ηλεκτρικοί κανόνες που προτείνεται στο μενού βοήθειας

## Insert Layout

Η εντολή “File -> Leave Microwind2” (ή CTRL+Q) στο βασικό μενού. Αν έχετε κάνει ένα σχέδιο ή αν έχετε τροποποιήσει κάποια δεδομένα, θα ερωτηθείτε να το αποθηκεύσετε. Μετά την επιβεβαίωση, μπορείτε να επιστρέψετε στα Windows.

## Generate



Η γεννήτρια layout περιλαμβάνει ένα σύνολο από προκαθορισμένες μακροεντολές layout όπως κουτί, επαφές, συσκευές n-MOS και p-MOS, αντιστάτες, γραμμές μετάλλου, μονοπάτι μετάλλου, επαγωγό, δίοδο, πυκνωτή, κείμενο και I/O pads. Εκείνα τα κελιά είναι δομημένα σύμφωνα με τους κανόνες σχεδιασμού και παραμέτρους μεγέθους χρήστη.

## Generate Box

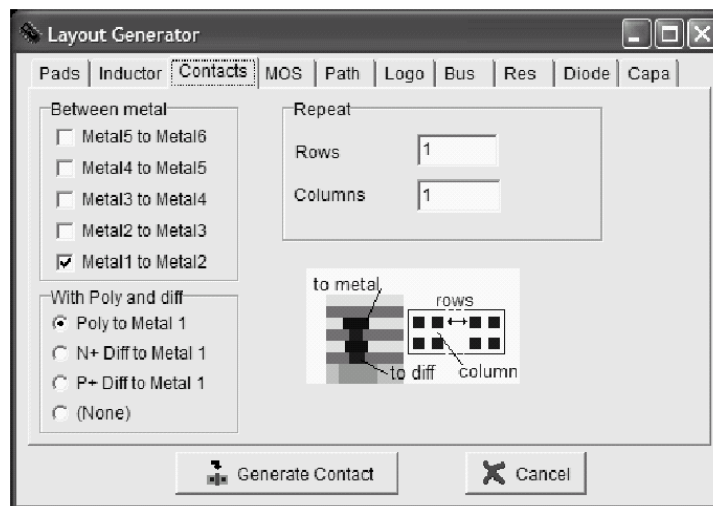


Όπως και η εντολή “Draw Box” που περιγράφηκε σε προηγούμενες σελίδες.

## Generate Contacts



Αυτή η μακροεντολή παράγει επαφές όπως polysilicon/metal, n-diffusion/metal, p-diffusion/metal και metal1/metal2/metal3/metal4, ή σωριασμένες επαφές που μπορούν να ληφθούν εδώ. Μπορείτε επίσης να κάντε κλικ στο παραπάνω εικονίδιο στην παλέτα. Πολλαπλές επαφές μπορούν να παραχθούν όταν εισάγετε αριθμό επαφών στο X και στο Y μεγαλύτερο από ένα.

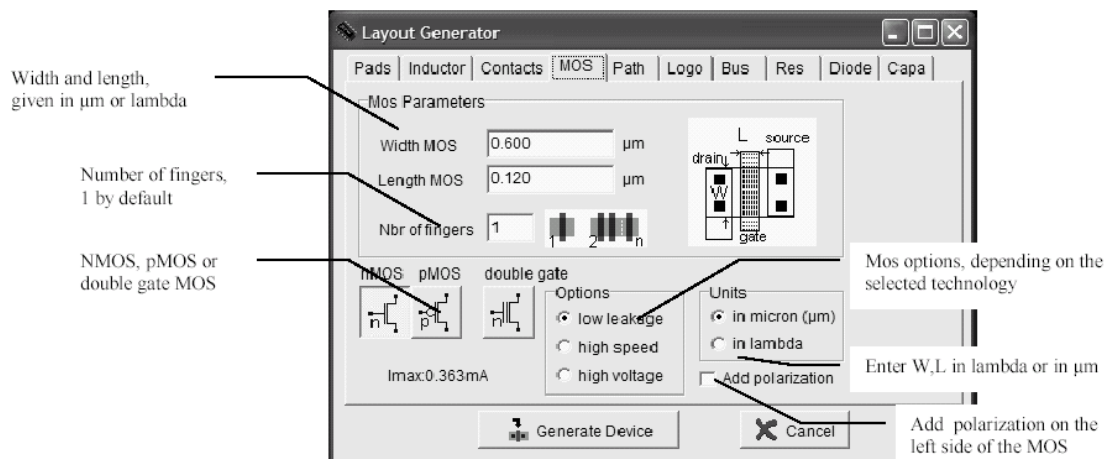


Εικόνα 13: Το μενού επαφών

## Generate nMOS, pMOS devices



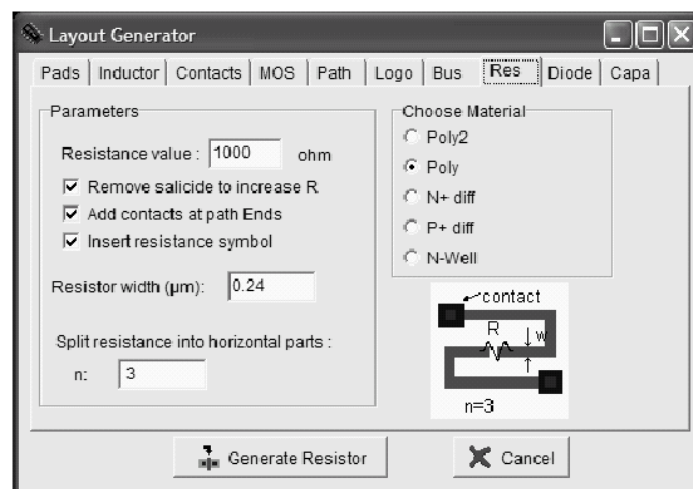
Αυτή η μακροεντολή παράγει είτε τρανζίστορ n-καναλιού είτε p-καναλιού. Η διπλή πύλη MOS είναι επίσης διαθέσιμη σε ορισμένες τεχνολογίες CMOS, για τη δόμηση μνήμης EEPROM. Οι παράμετροι του κελιού είναι το μήκος καναλιού (αρχική τιμή δίδεται από τους κανόνες σχεδιασμού), το πλάτος του και ο αριθμός των πυλών. Όταν αυτές οι παράμετροι οριστούν, εμφανίζεται το περίγραμμα της συσκευής. Κάντε κλικ στο ποντίκι για να την τοποθετήσετε στο κατάλληλο μέρος.



Εικόνα 14: Το μενού παραγωγής MOS

## Generate Resistor

Αυτή η εντολή παράγει ένα τρανζίστορ σε n-well, polysilicon ή poly2, N+ ή P+ διάχυση. Η αρχική άποψη του τρανζίστορ είναι ένα Z με τρεις μπάρες (Παράμετρος n στο μενού). Ένα εικονικό σύμβολο αντιστάτη μπορεί να εισαχθεί στο layout του αντιστάτη, για να βεβαιώσει το χειρισμό του φαινομένου αντίστασης κατά τη διάρκεια της εξομοίωσης. Εξ' αρχής, ένα επίπεδο κατ' επιλογήν που διαμορφώνεται για να αφαιρέσει τη salicidation<sup>1</sup> προστίθεται στο layout της αντίστασης. Εξ' αρχής, όλα τα πολυπυρίτια και οι διαχύσεις έχουν μια salicide επιφάνεια επιμετάλλωσης για να μειωθεί κατά ένα παράγοντα περί το 10 η αντίσταση του φύλλου. Η unsalicide επιλογή συνίσταται για υψηλή τιμή αντίστασης σε μικρή περιοχή. Τελικά, οι επαφές προστίθενται εξ' αρχής στις εγγύς και απομακρυσμένες άκρες της αντίστασης για να διευκολυνθεί η περαιτέρω διασύνδεση.

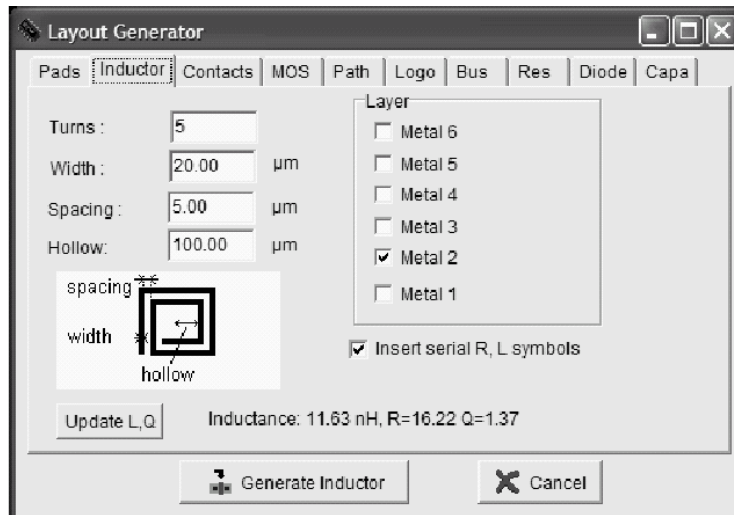


Εικόνα 15: Το μενού γεννήτριας αντιστάσεων

## Generate Inductor

Αυτή η εντολή παράγει μια σπείρα φτιαγμένη από επίπεδα μετάλλου ορισμένα-από-το-χρήστη. Αυτή η συσκευή χρησιμοποιείται για ταλαντωτές πολύ υψηλής συχνότητας. Αυτός ο αγωγός θεωρείται ως μια επαγωγή χάριν στο εικονικό σύμβολο επαγωγού που εισάγεται στο layout. Μια εκτίμηση της επαγωγής προτείνεται με το πάτημα του κουμπιού **Update L,Q**. Ο επαγωγός μπορεί να είναι ένας σωρός από διάφορα επίπεδα, χάριν στο μενού επιπέδων.

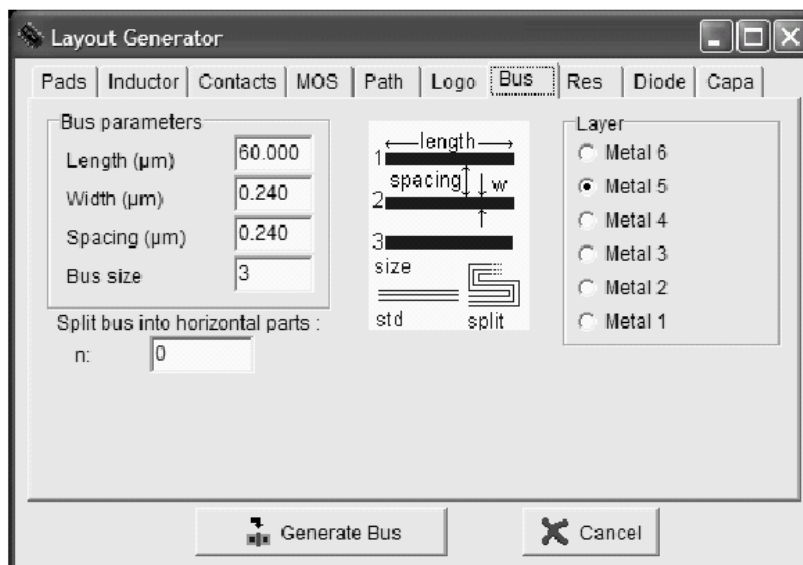
<sup>1</sup> Σημείωση: salicidation (self-aligned silicide) είναι μια διαδικασία στην τεχνολογία παραγωγής ολοκληρωμένων κυκλωμάτων, η οποία βασίζεται στη χρήση πυριτίου (silicon) και tantalίου (tantalum) ως μείγμα με όνομα silicide. Η διαδικασία salicide χρησιμοποιείται για την κατασκευή μασκών πολυπυριτίου για τις πύλες των pMOS και nMOS τρανζίστορ. Η τεχνική αυτή εξασφαλίζει πολυπυρίτιο χαμηλής σχετικά αντίστασης, καθιστώντας το κατάλληλο για διασυνδέσεις μεταξύ των τρανζίστορ, αντί για τη χρήση μετάλλου. Έτσι χρησιμοποιείται πολυπυρίτιο και για τις πύλες των τρανζίστορ και για τη διασύνδεση των πυλών τους, απλοποιώντας τις μάσκες που χρησιμοποιούνται στην παραγωγή των ολοκληρωμένων κυκλωμάτων.



Εικόνα 16: Το μενού γεννήτριας επαγωγών

## Generate Bus

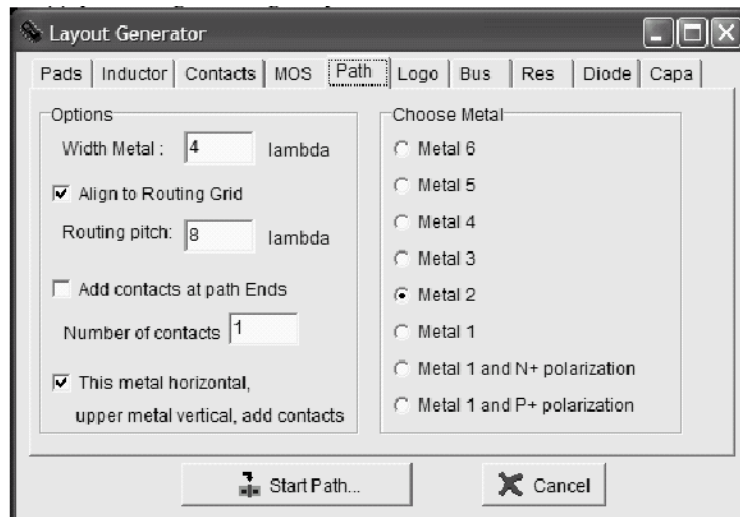
Αυτή η εντολή παράγει ένα σύνολο από παράλληλες γραμμές με επίπεδα, πλάτος και διάκενο ορισμένα-από-το-χρήστη. Αυτή η εντολή είναι χρήσιμη για τη δόμηση συνεζευγμένων διασυνδέσεων, ή μονοπατιών διαύλων που χρησιμοποιούνται στην τελική δρομολόγηση του τσιπ.



Εικόνα 17: Το μενού γεννήτριας διαύλων

## Generate Path

Αυτή η εντολή παράγει ένα μονοπάτι από διασυνδέσεις χρησιμοποιώντας ένα και μοναδικό επίπεδο. Το πλάτος του μονοπατιού μπορεί να αλλάξει, καθώς επίσης και η ευθυγράμμιση στο δίκτυο δρομολόγησης. Ένα σύνολο επαφών μπορεί επίσης να τοποθετηθεί και στα δύο άκρα του μονοπατιού. Αυτή η εντολή είναι πολύ χρήσιμη για σχεδιασμό τροφοδοσίας VDD και VSS και για διασυνδέσεις ενός επιπέδου.



Εικόνα 18: Το μενού διαμόρφωσης μονοπατιών

## Generate I/O pads

Είναι δυνατόν να προστεθούν διάφορα αντικείμενα όπως μονό pad (συνήθως 80x80μm), ή ακόμα και ένα σύνολο από pads παντού και στο layout χρησιμοποιώντας δακτυλίους τροφοδοσίας VDD και VSS. Στην τελευταία περίπτωση (προσθέτοντας περισσότερα από ένα pad), δώστε τον αριθμό των pads για κάθε πλευρά του τσιπ και αν απαιτείται τροποποιήστε το πλάτος των γραμμών VDD και VSS, καθώς και τον αριθμό των ζευγών pad VDD/VSS.

## Generate Diode

Μπορείτε να παράγετε επίσης μια σφραγίδα πόλωσης γύρω από την επαφή για τη δημιουργία μιας προστασίας με pad δίοδο για παράδειγμα.

## Make Spice File

Κάντε κλικ στο File -> Make Spice File για να μεταφράσετε το σχέδιό σας σε μια περιγραφή συμβατή με SPICE. Ο εξαγωγέας κυκλωμάτων που περιλαμβάνεται στο λογισμικό παράγει το ισοδύναμο κυκλωματικό διάγραμμα του layout και μια netlist συμβατή με SPICE έτοιμη για εξομοίωση. Μπορείτε να επιλέξετε ένα μοντέλο που θα χρησιμοποιήσετε για την εξομοίωση. Η επιλογή κείται μεταξύ model 1, model 3 και model 9.

- ◆ Η περιγραφή SPICE περιλαμβάνει τη λίστα από τρανζίστορ n-καναλιού και p-καναλιού και το συσχετισμένο με αυτά πλάτος και μήκος εξαχθέν από το layout.
- ◆ Το αρχείο κειμένου επίσης αναφέρει λεπτομερώς τα ονόματα κόμβων, τις παρασιτικές χωρητικότητες, και τα μοντέλα συσκευών.
- ◆ Το όνομα αρχείου του SPICE αντιστοιχεί στο τρέχον όνομα αρχείου με την κατάληξη .CIR

## Measure

### distance



Ο χάρακας δίνει τις οριζόντιες και κάθετες μετρήσεις (dx και dy) μεταξύ δύο σημείων, απευθείας στην οθόνη σε λάμδα ( $\lambda$ ) και μικρά ( $\mu\text{m}$ ). Η αλγεβρική απόσταση (d) δίδεται επίσης σε  $\mu\text{m}$ . Ο χάρακας απλά σβήνεται από την εντολή “View -> Refresh the screen” ή πατώντας <ESC>.

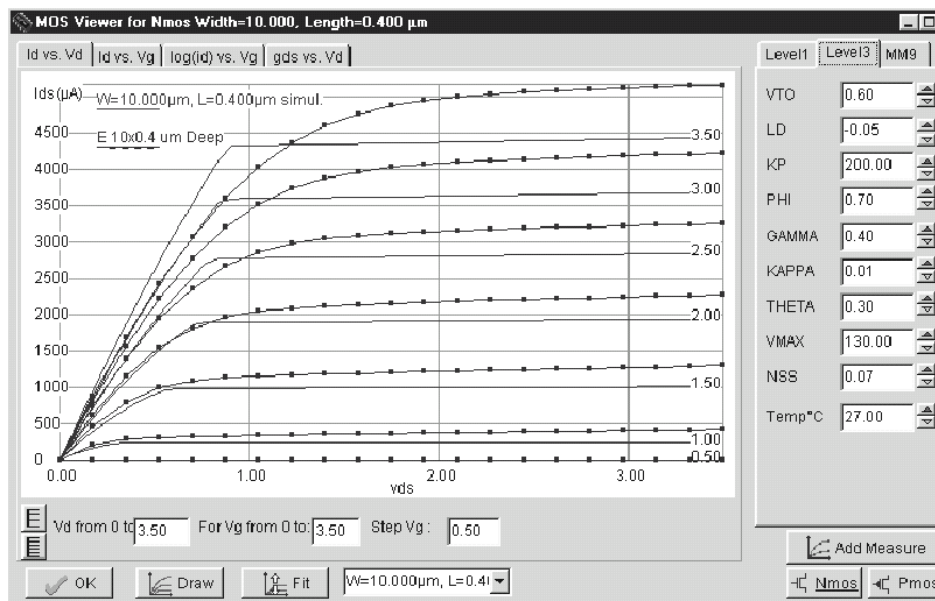
### MOS

### Characteristics



Κάντε κλικ στο εικονίδιο. Η καμπύλη  $I_d/V_d$  του αρχικού MOS ( $W=20\mu\text{m}$ ,  $L=L$  minimum) εμφανίζεται.

- ♦ Τα αποτελέσματα της αλλαγής των παραμέτρων του μοντέλου μπορούν να φανούν άμεσα στην οθόνη με το κλικ στα μικρά βέλη (Up/Down), τα οποία αλλάζουν τις τιμές των παραμέτρων.
- ♦ Κάντε κλικ στο “Id vs Vg” για να επισημάνετε την τάση κατωφλίου
- ♦ Κάντε κλικ στο “Id(log) vs. Vg” για να δείτε τη συμπεριφορά υπό-κατωφλίου.
- ♦ Προσθέστε μετρήσεις επιλέγοντας ένα αρχείο “.MES”
- ♦ Μεταβείτε από τη συσκευή NMOS στην PMOS κάνοντας κλικ στο αντίστοιχο κουμπί
- ♦ Επιλέξτε ένα μέγεθος για τη συσκευή στο κατώτερο μενού λίστας



Τρία μοντέλα μπορούν να χρησιμοποιηθούν:

- MOS Model 1 (Berkeley Spice level 1) για συσκευές μακριού καναλιού. Αυτό το μοντέλο είναι ξεπερασμένο αλλά παραμένει ενδιαφέρον για σύγκριση με προηγμένα μοντέλα.
- MOS Model 3 (απλουστευμένη μορφή του Berkeley Spice level 3). Παραμένει σε χρήση για πρώτης τάξης εκτίμηση της απόδοσης των κυκλωμάτων.

- BSIM4 (απλουστευμένη μορφή του Berkeley MOS BSIM4). Το μοντέρνο μοντέλο για μοντελοποίηση συσκευών στη βαθιά υπό-μικρόν περιοχή.

## MOS List

Κάντε κλικ στο “Edit -> MOS List” για να πάρετε τη λίστα των συσκευών MOS n-καναλιού και p-καναλιού που υπάρχουν την εκάστοτε στιγμή στο layout. Η λίστα MOS παρουσιάζεται στο παράθυρο πλοήγησης. Κάντε κλικ στο επιθυμητό MOS στη λίστα για να μεγεθύνετε στην αντίστοιχη τοποθεσία στο layout.

Move,

Stretch a

Box



Για να μετακινήσετε ένα κουτί, κάντε κλικ στο παραπάνω εικονίδιο. Χρησιμοποιώντας το ποντίκι, δημιουργήστε μια περιοχή που να περικλείει το κουτί. Μετά, τραβήξτε το ποντίκι στη νέα θέση και απελευθερώστε το ποντίκι. Σαν αποτέλεσμα, το κουτί έχει μετακινηθεί στη νέα θέση. Επαναλάβετε το ίδιο προκειμένου να μετακινήσετε ένα σύνολο από κουτιά.

- ♦ Για να προστατέψετε ένα επίπεδο από το να μετακινηθεί, κάντε κλικ στο τετράγωνο της παλέτας η οποία βρίσκεται στη δεξιά πλευρά του επιπέδου. Αυτό θα μετακινήσει το tik.
- ♦ Για να τεντώσετε ένα κουτί, κάντε κλικ στη μία πλευρά του κουτιού που θέλετε να τεντώσετε. Το περίγραμμα του κουτιού εμφανίζεται. Τραβήξτε το ποντίκι στη νέα θέση και απελευθερώστε το ποντίκι. Το κουτί τεντώθηκε.

Συμβουλή: Για να πιάσετε το επιθυμητό όριο του κουτιού, τραβήξτε μια γραμμή κάθετη στο σύνορο, μπαίνοντας στο κουτί

Move

Step by

Step

Για να μετακινήσετε ένα κουτί λάμδα-λάμδα, κάντε κλικ στο “Edit -> Move Step by Step”. Χρησιμοποιώντας το ποντίκι, δημιουργήστε μια περιοχή που να περιλαμβάνει τα κουτιά. Η επιλογή εμφανίζεται κίτρινη. Μετά, κάντε κλικ στο βέλος μέχρι η επιλογή να μετακινηθεί στη νέα θέση. Η τιμή του βήματος μετακίνησης (σε λάμδα) δηλώνεται στη γραμμή επεξεργασίας.



## New

Κάντε κλικ στο “File -> New” προκειμένου να ξαναξεκινήσετε το λογισμικό με μια άδεια οθόνη. Το τρέχον σχέδιο πρέπει να σωθεί πριν τη χρήση αυτής της εντολής, καθώς όλη η γραφική πληροφορία θα σβηστεί από τη μνήμη του υπολογιστή. Δεν υπάρχει Undo για να αναιρέσετε την εντολή New.

## Open

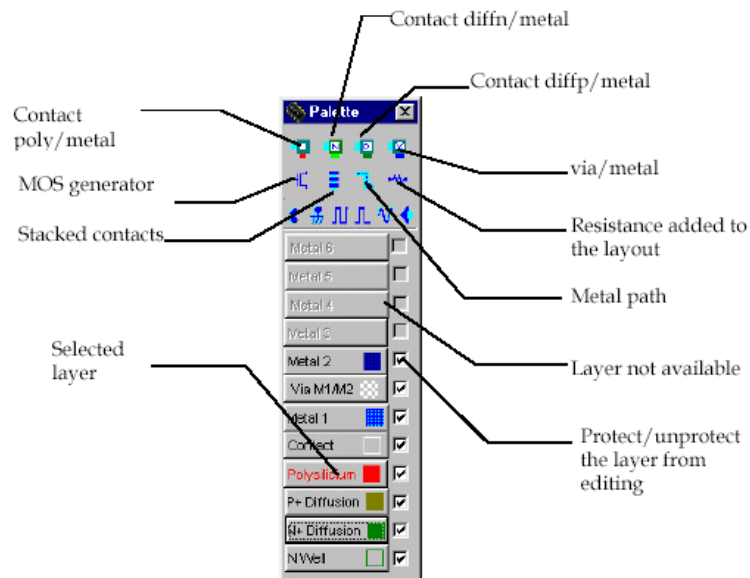


Κάντε κλικ στο παραπάνω εικονίδιο. Στη λίστα, κάντε διπλό κλικ στο αρχείο που θα φορτωθεί. Η αρχική επέκταση είναι “.MSK” η οποία αντιστοιχεί στα αρχεία layout. Τα αρχεία CIF (“.CIF”) μπορούν να φορτωθούν με τον ίδιο τρόπο. Το κατάλληλο πρόγραμμα μετατροπής μετασχηματίζει την είσοδο CIF σε μορφή MSK.

## Palette



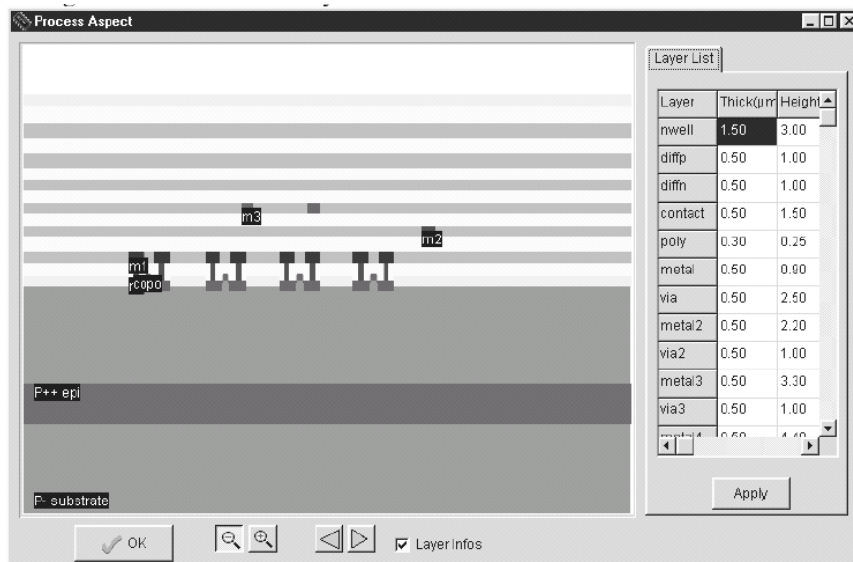
Η παλέτα βρίσκεται στη δεξιά πλευρά της οθόνης. Ένα μικρό τικ υποδεικνύει το τρέχον επίπεδο. Το τρέχον επίπεδο εξ' αρχής είναι το πολυπυρίτιο (polysilicon-PO). Η λίστα των επιπέδων παρουσιάζεται παρακάτω.



- ◆ Αν αφαιρέσετε το τικ στο δεξιό μέρος του επιπέδου, το επίπεδο γυρνάει σε κατάσταση προστασίας. Οι εντολές Cut, Stretch, και Copy δεν μπορούν πλέον να επηρεάσουν το επίπεδο.
- ◆ Χρησιμοποιήστε το “View -> Protect all” για να προστατέψετε όλα τα επίπεδα. Τα τικς σβήνονται.
- ◆ Χρησιμοποιήστε το “View -> Unprotect all” για να αναιρέσετε την προστασία. Όλα τα επίπεδα μπορούν να επεξεργαστούν.

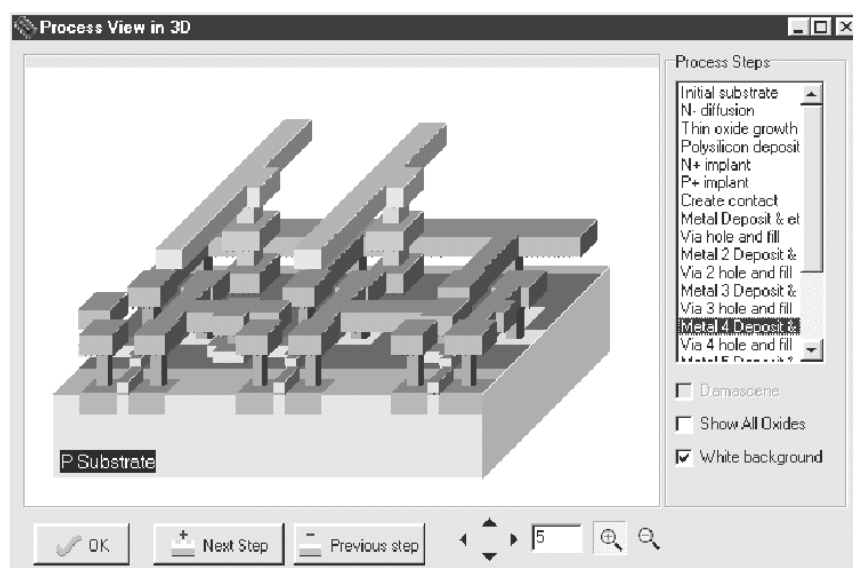
## Process Section in 2D

Κάντε κλικ στο παραπάνω εικονίδιο για να προβείτε στην εξομοίωση διεργασίας. Μια γραμμή χειριζόμενη-με-ποντίκι παρέχεται και ενσωματώνει την τομή. Η παρακάτω οθόνη εμφανίζεται. Τα βέλη μπορούν να χρησιμοποιηθούν για να μετακινηθεί η τομή προς τα δεξιά ή προς τα αριστερά του άξονα X, και μπροστά ή πίσω στον Y άξονα. Μεγεθύνσεις ή σμικρύνσεις είναι διαθέσιμες. Αφαιρέστε τα ονόματα επιπέδων, αφαιρώντας το τικ μπροστά από το “Layer infos”.



## Process Section in 3D

Κάντε κλικ στο “Simulation -> Process Steps in 3D”. Κάντε κλικ στο “Next step” για να παρακολουθήσετε τον τρόπο με τον οποίο το τρέχον επεξεργαζόμενο layout στην οθόνη, θα παρασκευαστεί χρησιμοποιώντας την επιλεγμένη τεχνολογία. Χρησιμοποιήστε το βέλος για να ολισθήσετε το εμφανιζόμενο μέρος. Μεγεθύνσεις και σμικρύνσεις είναι διαθέσιμες.



## Protect All

Κάντε κλικ στο “View -> Protect All” για να προστατέψετε όλα τα επίπεδα από σκοπό επεξεργασίας. Όλα τα τικς στην παλέτα θα αφαιρεθούν.

## Print Layout

Κάντε κλικ στο “File -> Print Layout” για να μεταφέρετε τα γραφικά περιεχόμενα της οθόνης στον εκτυπωτή. Εναλλακτικά, μπορείτε να κάνετε ένα αντίγραφο στο παράθυρο στο clipboard προκειμένου να εισάγετε την οθόνη στον επιθυμητό επεξεργαστή κειμένου πατώντας <Alt> + <Print Screen>. Στον επεξεργαστή κειμένου ή στον επεξεργαστή γραφικών, απλά κάντε κλικ στο “Edit -> Paste”. Συνιστούμε να αλλάξετε σε μονόχρωμη παρουσίαση πρώτα καλώντας τη λειτουργία “File -> Colors -> Switch to Monochrom”. Σε αυτήν την περίπτωση το layout θα ζωγραφιστεί σε ένα άσπρο παρασκήνιο χρησιμοποιώντας επίπεδα του γκρι και διαγραμμίσεις.

## Rotate

Για να εφαρμόσετε την περιστροφή σε ένα κομμάτι του σχεδίου, κάντε κλικ στο “Edit -> Rotate”. Περιορίστε την ενεργό περιοχή των κουτιών στο layout έτσι ώστε να μπορεί να τροποποιηθεί χρησιμοποιώντας το ποντίκι.

## Save



Κάντε κλικ στο “File -> Save” για να σώσετε το layout με το τρέχον όνομά του. Το αρχικό όνομα είναι “EXAMPLE.MSK”.

## Save As

Ένα νέο παράθυρο εμφανίζεται, στο οποίο μπορείτε να εισάγετε το όνομα του σχεδίου. Χρησιμοποιήστε το πληκτρολόγιο και γράψτε το επιθυμητό όνομα αρχείου. Πατήστε “Save”. Το σχέδιο καταχωρείται με το επίθεμα .MSK.

## Search Text

Ο πιο βολικός τρόπος για να βρείτε ένα κείμενο στο layout είναι καλώντας το “Edit -> Search Text”. Η λίστα των ετικετών κειμένου εμφανίζεται στο μενού πλοήγησης. Αν κάντε κλικ στο επιθυμητό κείμενο, η οθόνη ξανασχεδιάζεται έτσι ώστε η επόμενη ετικέτα να είναι στο κέντρο του παραθύρου, με δύο γραμμές να σχεδιάζουν ένα σταυρό στη θέση του κειμένου. Οι ιδιότητες εμφανίζονται στο μενού πλοήγησης.

- ◆ Κάντε κλικ στο “Hide” για να κλείσετε το παράθυρο πλοήγησης.
- ◆ Κάντε κλικ στο “Extract” για να προσθέσετε τις ηλεκτρικές ιδιότητες του επιλεγμένου κειμένου αν το layout δεν έχει εξαχθεί προηγουμένως.

- ♦ Στην περίπτωση μιας πολύ μεγάλης λίστας κειμένου, επιλέξτε το πρώτο γράμμα του κειμένου ανά χείρας, πατήστε αυτό το γράμμα στο πληκτρολόγιο. Αυτό θα έχει αυτόματα αποτέλεσμα και θα ερευνήσει αλφαβητικά και ο επιλογέας θα μετακινηθεί στην πρώτη ετικέτα που αρχίζει από το επιλεγμένο γράμμα.

## Select Foundry

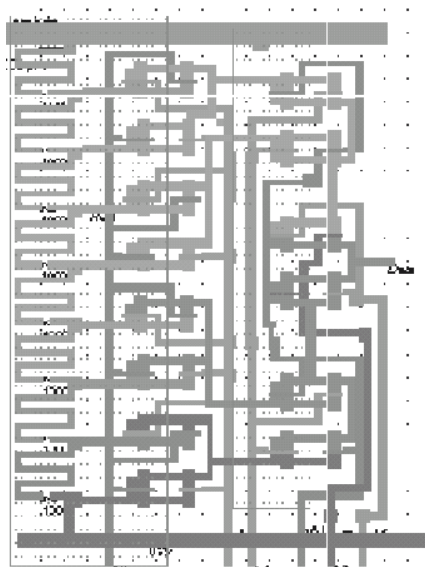
Κάντε κλικ στο “File -> Select Foundry”. Η λίστα με τις διαθέσιμες διεργασίες εμφανίζεται. Το αρχικό αρχείο κανόνων σχεδιασμού είναι γραμμένο με έντονους χαρακτήρες. Διάφορες τεχνολογίες είναι διαθέσιμες από 1.2 έως και 0.12 μm. Κάντε κλικ στο όνομα αρχείου κανόνων και το λογισμικό αναδιαμορφώνει τον εαυτό του προκειμένου να προσαρμοστεί στη νέα διεργασία.

## Simulation Parameters

- ♦ Η αρχική εξαγωγή περιλαμβάνει την αφαίρεση πλεοναζόντων κουτιών (Purge) και την αφαίρεση των επικαλύψεων (Merge). Η γρήγορη εξαγωγή δεν χειρίζεται λειτουργίες Purge ή Merge.
- ♦ Το επίπεδο MOS μπορεί να επιλεγεί μεταξύ επιπέδου 1, 3 και 9. Δείτε το κεφάλαιο 2 για περισσότερες λεπτομέρειες για τα μοντέλα.
- ♦ Άλλες επιλογές αφορούν στον υπολογισμό της πλευρικής (lateral) χωρητικότητας και της χωρητικότητας κάθετου crosstalk.

## Simulation on layout

Η εξομοίωση γίνεται άμεσα στο layout με μια παλέτα χρωμάτων. Τα πιο ενδιαφέροντα αρχεία layout που μπορούν να εξομοιωθούν σε αυτή τη διαμόρφωση είναι τα αναλογικά τμήματα όπως ο DAC.

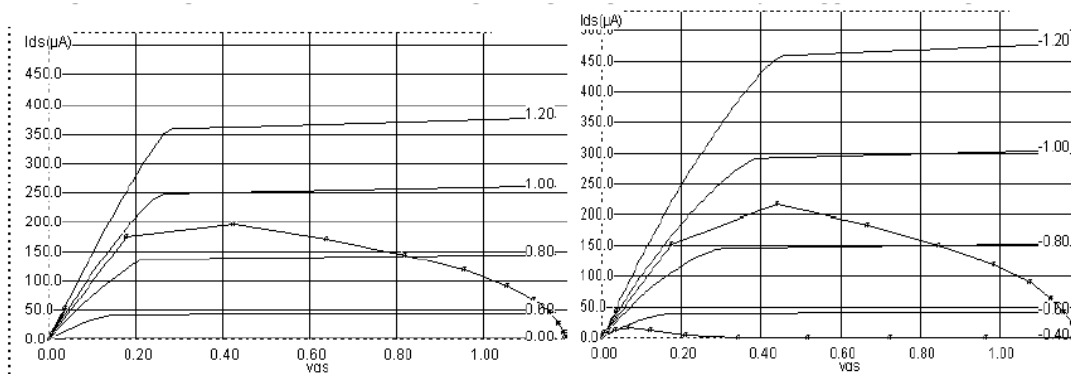


Κατά τη διάρκεια του ανοιγοκλείματος, οι συσκευές MOS του αντιστροφέα συμπεριφέρονται εναλλακτικά ως κλειστοί και ανοικτοί διακόπτες, όπως παρουσιάστηκε προηγουμένως. Σε μια πρώτης τάξης προσέγγιση, το ισοδύναμο μοντέλο του διακόπτη είναι μια αντίσταση. Στην παρουσίαση της εικόνας 4-51, παρέχεται το γράφημα του σημείου λειτουργίας στις χαρακτηριστικές  $I_d/V_d$ , και για τις nMOS και για τις

pMOS συσκευές. Έτσι, η αντίσταση δεν είναι μια απλή συνάρτηση, αλλά περισσότερο μια μεταβαλλόμενη αντίσταση μεταξύ συγκεκριμένων τιμών.

Η εικόνα 4-51 έχει ληφθεί χρησιμοποιώντας μια συγκεκριμένη λειτουργία εξομοίωσης που λέγεται **Simulation on Layout** στο μενού εξομοίωσης. Οι χαρακτηριστικές I/V των επιλεγμένων συσκευών ενημερώνονται κατά τη διάρκεια της εξομοίωσης για να εντοπίσουν το σημείο λειτουργίας.

Κατά τη διάρκεια της αναλογικής εξομοίωσης, η τάση κόμβου υπερθέτεται στο layout και εμφανίζεται με μια παλέτα χρωμάτων:



Εικόνα 4-51: Οι συσκευές nMOS και pMOS κατά τη διάρκεια του ανοιγοκλειμάτος (Inv.MSK)

### Start Simulation



Το παραπάνω εικονίδιο ή η εντολή “Simulate -> Start Simulation” μπορεί να δώσει πρόσβαση στην αυτόματη εξαγωγή και την αναλογική εξομοίωση του layout.

- Κάντε κλικ στο “Voltage vs Time” για να πάρετε τη μεταβατική ανάλυση όλων των ορατών σημάτων. Η καθυστέρηση μεταξύ του επιλεγμένου κόμβου εκκίνησης και του επιλεγμένου κόμβου τερματισμού υπολογίζεται στο  $VDD/2$ . Μπορείτε να αλλάξετε τον επιλεγμένο κόμβο εκκίνησης στη λίστα κόμβων, στο δεξιό πάνω μενού του παραθύρου. Μπορείτε να κάνετε το ίδιο για τον επιλεγμένο κόμβο τερματισμού.
- Κάντε κλικ στο “Voltage and Currents” ώστε να εμφανιστούν οι καμπύλες τάσης στο κατώτερο παράθυρο, και τα VDD, VSS και τα επιθυμητά ρεύματα MOS εμφανίζονται στο ανώτερο παράθυρο. Σε αυτή τη διαμόρφωση, η καταναλισκόμενη ισχύς στην εξομοίωση επιδεικνύεται επίσης.
- Κάντε κλικ στο “Voltage vs Voltage” για να πάρετε χαρακτηριστικές μεταφοράς μεταξύ του επιλεγμένου κόμβου στον X άξονα και του αντίστοιχου στον Y άξονα. Αρχικά ο κόμβος εκκίνησης είναι το πρώτο ρολόι ή παλμός στη λίστα κόμβων, και ο κόμβος τερματισμού είναι ο πρώτος μεταβαλλόμενος κόμβος. Αυτή η διαμόρφωση είναι χρήσιμη για τον υπολογισμό των χαρακτηριστικών Αντιστροφεία (σημείο μετάβασης), τη DC απόκριση του τελεστικού ενισχυτή, ή για το σκανδαλισμό Schmitt για να φανεί το φαινόμενο υστέρησης. Η πρώτη εξομοίωση υπολογίζει την τιμή του κόμβου τερματισμού και μεταβολή του κόμβου εκκίνησης από 0 σε VDD. Αυτό το γνώρισμα είναι ενδιαφέρον για κύκλωμα με φαινόμενα μνήμης (Schmitt trigger). Σημειώστε ότι οι καμπύλες μπορεί να μην είναι ακριβώς οι ίδιες. Μπορείτε να αυξήσετε την ακρίβεια μειώνοντας το βήμα υπολογισμού “Precision”, που παρέχεται στο μενού και εκφράζεται σε mV.

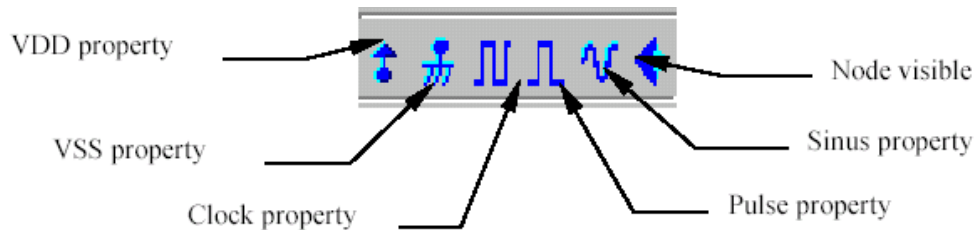
**ΣΗΜΕΙΩΣΗ:** Μπορείτε να τροποποιήσετε το ελάχιστο βήμα εξομοίωσης  $\Delta t$ , αλλά μπορεί να είναι επικίνδυνο. Αν αυξήσετε το  $\Delta t$  η ταχύτητα εξομοίωσης βελτιώνεται αλλά το αριθμητικό σφάλμα μπορεί

να οδηγήσει σε ασταθείς εξομοιώσεις. Αν μειώσετε το  $\Delta t$ , η ταχύτητα εξομοίωσης μειώνεται επίσης, αλλά η αριθμητική ακρίβεια βελτιώνεται. Το ρίσκο της υπολογιστικής απόκλισης μειώνεται.

## Simulation

### Simulation Icons

Τα εικονίδια εξομοίωσης προσθέτουν ιδιότητες στους κόμβους. Ιδιότητες εφαρμόζονται στους ηλεκτρικούς κόμβους του κυκλώματος προκειμένου να εξυπηρετήσουν ως οδηγοί εξομοίωσης. Πρέπει να προσδιορίσετε ποιους κόμβους ανατίθεται σε ποια τάση πριν ξεκινήσετε την αναλογική εξομοίωση.



**VDD & VSS.** Ο κόμβος αποκτά τάση τροφοδοσίας με το εικονίδιο Vdd και τίθεται στα 0V με το εικονίδιο Vss.

**CLOCK.** Όταν ένας κόμβος γίνεται ρολόι, οι παράμετροι του τελευταίου διαιρούνται ως ακολούθως: χρόνος ανόδου, επίπεδο ένα, χρόνος καθόδου, επίπεδο μηδέν. Όλες οι τιμές εκφράζονται σε nano-seconds (ns). Αν ζητήσετε ένα δεύτερο ρολόι, η περίοδος θα πολλαπλασιαστεί επί δύο.

- ◆ Μπορείτε να ανταλλάξετε το επίπεδο 0 με το επίπεδο 1 εισάγοντας μια νέα τιμή με το πληκτρολόγιο.
- ◆ Για να παράγετε ένα ρολόι που ξεκινά από VDD αντί για VSS, κάντε κλικ στο “Invert L/H”.
- ◆ Χρησιμοποιήστε το “Period \* 2” για να πολλαπλασιάσετε την περίοδο του ρολογιού επί δύο.
- ◆ Χρησιμοποιήστε το “Period \* 2” για να διαιρέσετε την περίοδο διά του δύο.

**PULSE.** Ο παλμός ανοιγοκλείνει μεταξύ “Level 0” (0 εξ’ αρχής) και “Level 1” (VDD εξ’ αρχής) ανάλογα με τον οριζόμενο-από-το-χρήστη χρονο-πίνακα.

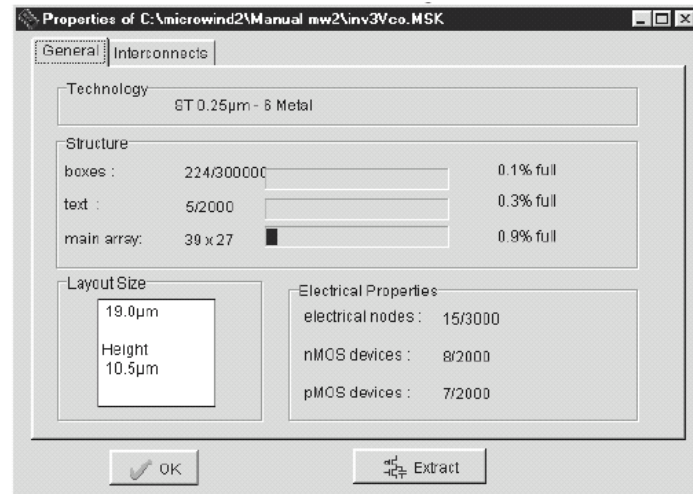
- ◆ Εισάγετε τη φράση «0101100» και πατήστε “Insert”. Ο χρονο-πίνακας ενημερώνεται
- ◆ Κάντε κλικ στο “Erase”: όλες οι γραμμές που βρίσκονται μετά το επιλεγμένο στοιχείο του χρονο-πίνακα σβήνονται.

**SINUS.** Οι παράμετροι της ημιτονοειδούς κυματομορφής είναι το πλάτος, το offset, η συχνότητα και η φάση.

**VISIBLE NODE.** Κάντε κλικ στο “μάτι” και κάντε κλικ στο υπάρχον κείμενο στο layout για να παράγετε χρονογράμματα του κόμβου. Αρχικά, όλοι οι κόμβοι είναι ορατοί, αλλά τα ρολόγια και οι κόμβοι διέγερσης γίνονται ορατοί ακολούθως.

## Statistics

Η εντολή “File -> Statistics” παρέχει κάποια πληροφορία για την τρέχουσα τεχνολογία, το ποσοστό της μνήμης που χρησιμοποιείται από το layout και το μέγεθος του layout συν τα λεπτομερή περιεχόμενά του. Αν το layout έχει εξαχθεί από πριν ή αν κάνετε κλικ στο “extract now”, ο αριθμός των συσκευών και των κόμβων θα ενημερωθεί.



## Undo

Η εντολή Undo (Edit -> Undo) είναι χρήσιμη ώστε να μη ληφθεί υπ' όψιν η τελευταία εντολή επεξεργασίας. Είναι δυνατόν να κάνετε Undo στις εντολές Cut, Paste, Copy, Move, Stretch, Edit και Compile.

## Unprotect

### All

Κάντε κλικ στο “View -> Unprotect All” για να επιλέξετε όλα τα επίπεδα για λόγους επεξεργασίας. Όλα τα τικς στην παλέτα διεγείρονται.

## Unselect

### All

Κάντε κλικ στο “View -> Unselect All” (ή στο <ESC>) για να αποεπιλέξετε το layout. Αυτή η εντολή είναι χρήσιμη για το σχεδιασμό του layout με τα αρχικά χρώματα μετά από εντολές όπως η View Interconnect ή η View Node, οι οποίες επισημαίνουν ένα και μόνο κόμβο.

## View All



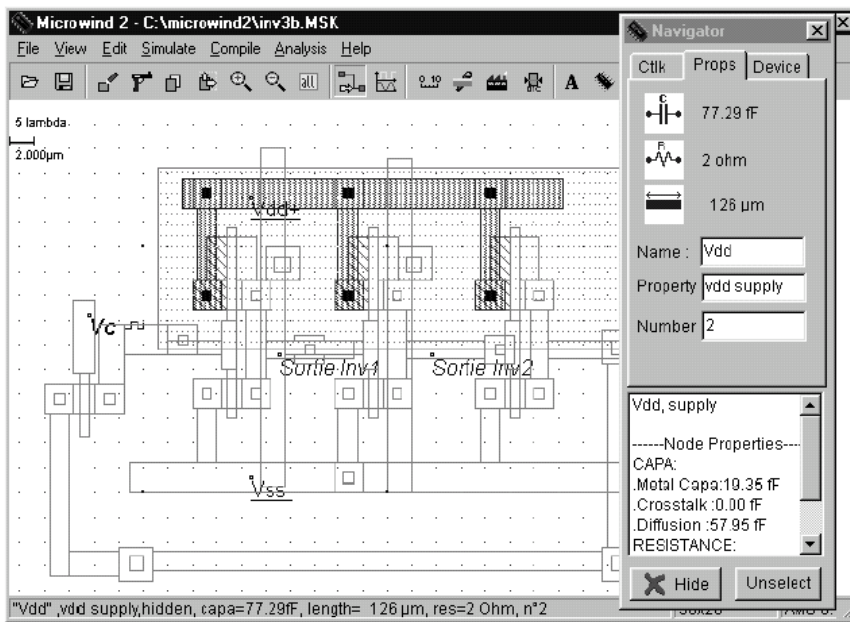
Κάντε κλικ στο “View -> View All” για να χωρέσετε την εικόνα με όλα τα τρέχοντα γραφικά στοιχεία στην οθόνη.

## View

## Node



Κάντε κλικ στο παραπάνω εικονίδιο ή στο “View -> View Node”. Μετά, κάντε κλικ στο επιθυμητό κουτί στο layout. Μετά από την ολοκλήρωση μιας διαδικασίας εξαγωγής, θα δείτε όλα τα κουτιά που είναι συνδεδεμένα σε αυτόν τον κόμβο. Στην περίπτωση ενός μεγάλου layout, η εντολή μπορεί να πάρει χρόνο. Η συσχετιζόμενη παρασιτική χωρητικότητα, η λίστα των ετικετών κειμένου που έχει προστεθεί στα επιλεγμένα κουτιά, και οι ιδιότητες κόμβου θα φανούν επίσης σε ένα ξεχωριστό παράθυρο πλοήγησης. Κάντε κλικ στο “Unselect”, “Hide”, <Escape> “View -> Unselect All” για να αποεπιλέξετε το layout.



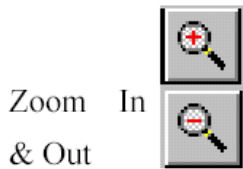
## View

## Interconn

ect

Η εντολή “View -> View Interconnect” εκτελεί μια ηλεκτρική εξαγωγή των κουτιών μετάλλου και πολυπυριτίου που είναι συνδεδεμένα με το επιθυμητό σημείο. Σε σύγκριση με το View Node, αυτή η εντολή δουλεύει ταχύτερα αλλά δεν λαμβάνει υπ’ όψιν διαχυμένα επίπεδα που μπορούν να προεκτείνουν τον δίκτυο κόμβων διασύνδεσης. Η εντολή δίνει μια λίστα συνδεδεμένων ετικετών κειμένου.

Κάντε κλικ στο <Escape> ή στο View -> Unselect All για να αποεπιλέξετε το layout.

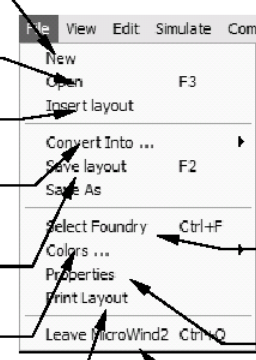
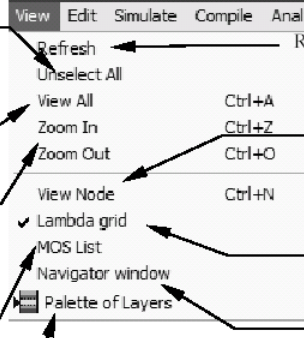


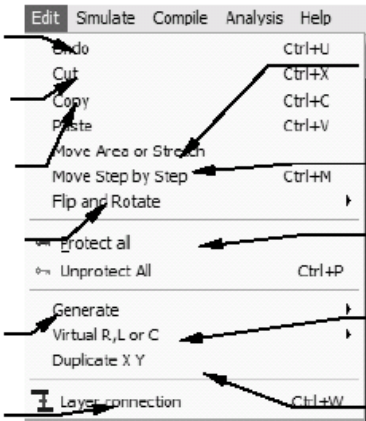
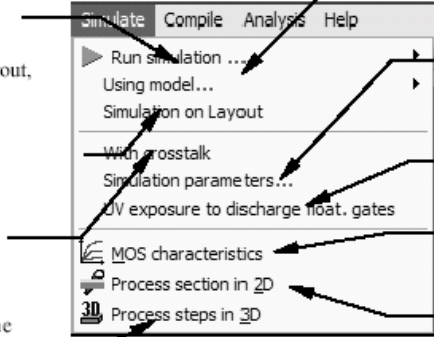
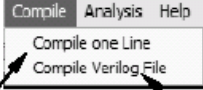
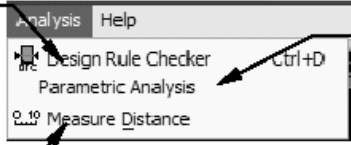
Τα παραπάνω εικονίδια εκτελούν Μεγέθυνση ή Σμίκρυνση. Όταν μεγεθύνετε, η περιοχή που προσδιορίστηκε από το ποντίκι θα μεγαλώσει για να χωρέσει στο παράθυρο παρουσίασης. Όταν σμικρύνετε, η περιοχή που προσδιορίστηκε με το ποντίκι θα περιέχει το παράθυρο παρουσίασης.

- ◆ Αν κάνετε κλικ μία φορά, μια μεγέθυνση θα εκτελεστεί στην επιθυμητή περιοχή.
- ◆ Πατήστε CTRL+A για να κάνετε “View All”, και CTRL+ο για να κάνετε σμίκρυνση.


## B. Οδηγός γρήγορης αναφοράς του Microwind.

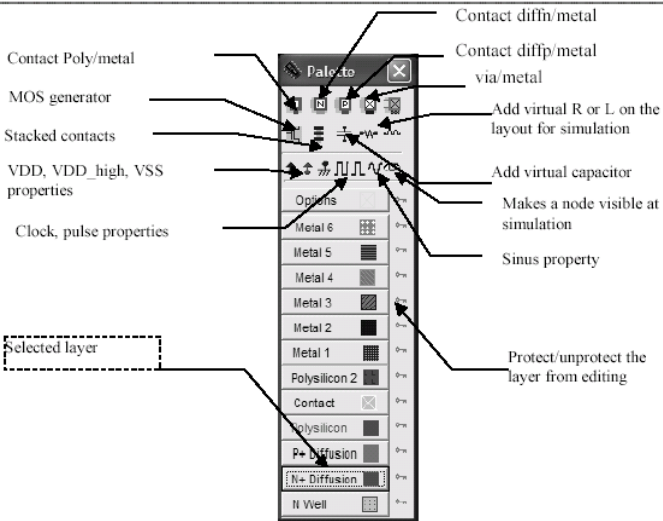
### 1. Microwind2 Menus

<p><b>FILE MENU</b></p>	<p>Reset the program and starts with a clean screen</p> <p>Read a layout data file</p> <p>Insert a layout in the current layout</p> <p>Translates the layout into CIF, SPICE</p> <p>Save the current layout into the current filename</p> <p>Switch to monochrom/Color mode</p> <p>Print the layout</p>  <p>Configure Microwind2 to a foundry</p> <p>Layout properties : number of box, devices, size, etc...</p> <p>Quit Microwind2 and returns to Windows</p>
<p><b>VIEW MENU</b></p>	<p>Unselect all layers and redraw the layout</p> <p>Fit the window with all the edited layout</p> <p>Zoom In, Zoom out the layout window</p> <p>Give the list of nMOS and pMOS devices</p> <p>Show the palette of layers, the layout macro and the simulation properties</p>  <p>Redraw the screen</p> <p>Extract the electrical node starting at the cursor location</p> <p>Show/Hide the lambda grid</p> <p>Show the navigator window to display the node properties</p>

<p><b>EDIT MENU</b></p>	<p>Cancel last editing command</p> <p>Cut elements included in an area</p> <p>Duplicate elements included in an area</p> <p>Flip or rotate elements included in an area</p> <p>Generate MOS, contacts, pads, diodes, resistors, capacitors, etc...</p> <p>Connect layers at a desired location</p>  <p>Move elements included in an area or stretch the selected box border</p> <p>Move step by step a selection of elements</p> <p>Protect and unprotect layers from copying, moving, erasing</p> <p>Add a virtual R,L,C for simulation purpose</p> <p>Duplicate in X and Y a selection of elements</p>
<p><b>SIMULATE MENU</b></p>	<p>Run the simulation and choose the appropriate mode V(t), I(t), V/V, F(t), etc</p> <p>Simulate directly on the layout, with a palette of colors representing voltage</p> <p>Include crosstalk effects in simulation</p> <p>View the process steps of the layout fabrication in 3D</p>  <p>Select model 1, model 3 or BSIM4</p> <p>Access to the SPICE model and some simulation options : VDD value, temperature, simulation step</p> <p>Discharge floating gates</p> <p>Access to static characteristics of the MOS devices</p> <p>2D view of the circuit at the desired location</p>
<p><b>COMPILE MENU</b></p>	<p>Compile one single line (on-line)</p>  <p>Compile a Verilog file generated by DSCH2</p>
<p><b>ANALYSIS MENU</b></p>	<p>Verifies the layout and highlight the design rule violations</p> <p>Measure the distance in the layout window, in <math>\mu\text{m}</math> and lambda</p>  <p>Computes the influence of one parameter such as VDD, <math>t^{\circ}</math>, capacitance, on a set of parameters: delay, frequency, etc...</p>

### PALETTE

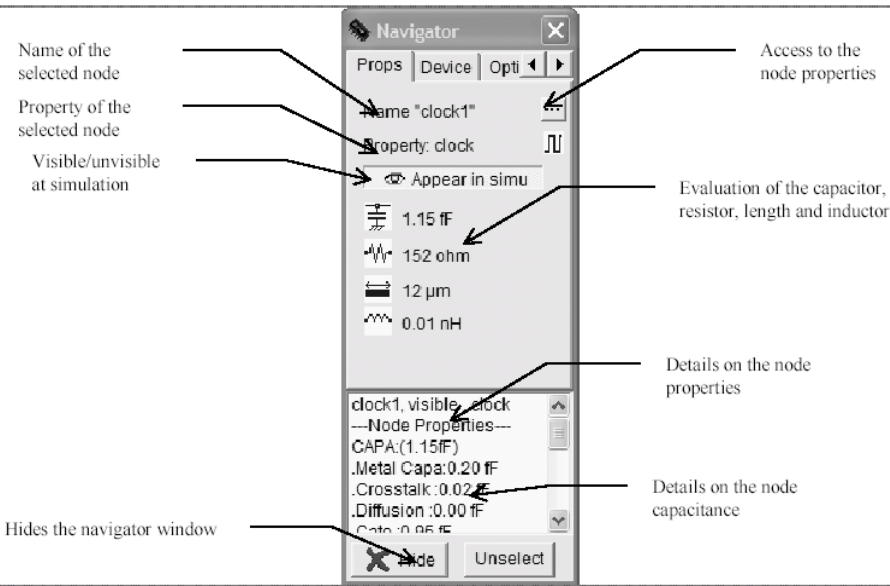




Annotations for Palette window:

- Contact Poly/metal
- MOS generator
- Stacked contacts
- VDD, VDD\_high, VSS properties
- Clock, pulse properties
- Contact diffn/metal
- Contact diffp/metal
- via/metal
- Add virtual R or L on the layout for simulation
- Add virtual capacitor
- Makes a node visible at simulation
- Sinus property
- Protect/unprotect the layer from editing
- Selected layer





















### NAVIGATOR WINDOW



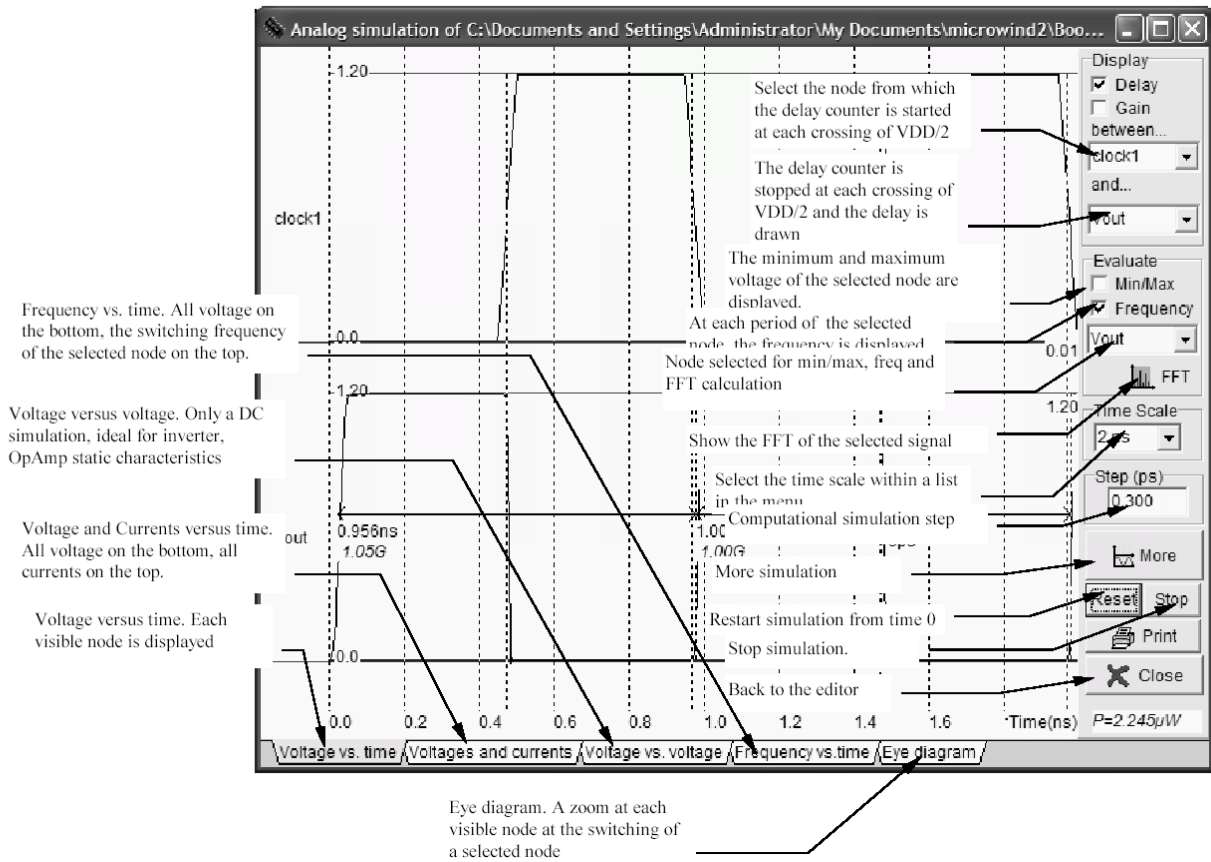
Annotations for Navigator window:

- Name of the selected node
- Property of the selected node
- Visible/unvisible at simulation
- Access to the node properties
- Evaluation of the capacitor, resistor, length and inductor
- Details on the node properties
- Details on the node capacitance
- Hides the navigator window

## LIST OF ICONS

	Open a layout file (MSK format)		Extract and simulate the circuit
	Save the layout file in MSK format		Measure the distance in lambda and micron between two points
	Draw a box using the selected layer of the palette		2D vertical aspect of the device
	Delete boxes or text.		Step by step fabrication of the layout in 3D
	Copy boxes or text		Design rule checking of the circuit. Errors are notified in the layout
	Stretch or move elements		Add a text to the layout. The text may include simulation properties.
	Zoom In		Connect the lower to the upper layers at the desired location using appropriate contacts.
	Zoom Out		Static MOS characteristics
	View all the drawing		View the palette
	Extract and view the electrical node pointed by the cursor		Move the layout up, left, right, down

## 2. Microwind2 Simulation menu



## Βιβλιογραφία

- [1] “Σχεδίαση ολοκληρωμένων κυκλωμάτων CMOS VLSI”, N.H.WESTE, K. ESHRAGHIAN, Μετάφραση – Επιμέλεια Κ. Πεκμετζή, Δ. Σούντρης, Κ. Γκούτης, εκδ. ΠΑΠΑΣΩΤΗΡΙΟΥ, Αθήνα 1996.
- [2] "Σχεδίαση Ολοκληρωμένων Κυκλωμάτων VLSI - Θεωρία", Ε. Κυριάκης-Μπιτζάρος, ΤΕΙ Πειραιά, Τμ. Ηλεκτρονικής.
- [3] A book on deep submicron CMOS design using Microwind (Appendix B) by **Etienne Sicard** and **Sonia Bendhia**, <http://inrage.insa-toulouse.fr/~etienne/microwind/book.html>