HY330 – Ψηφιακά Συστήματα VLSI

ΣΕΤ ΑΣΚΗΣΕΩΝ 4

8/1/2020

Βαγενάς Αναστάσης

ΑΕΜ : 2496

Άσκηση 1

Στην άσκηση αυτή θέλουμε να υλοποιήσουμε ένα αποκωδικοποιητή (αρχικά 1 bit ,ύστερα 3 σε 8):

Μέρος (α)

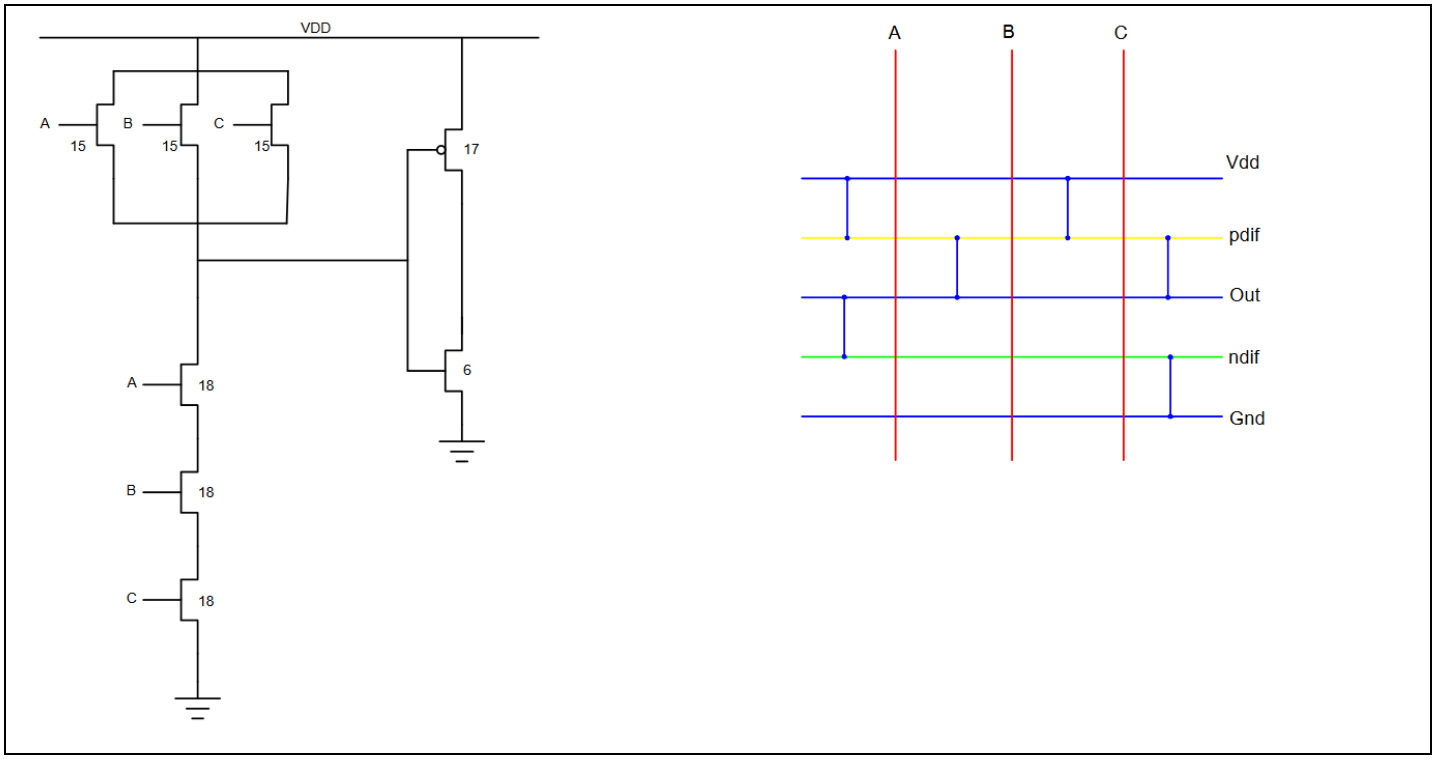
Πίνακας αληθείας (Πχ για το 1^{ο} bit)

A	B	C	out
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Η έκφραση για το 1^{ο} bit ουσιαστικά είναι το out = A'B'C' .

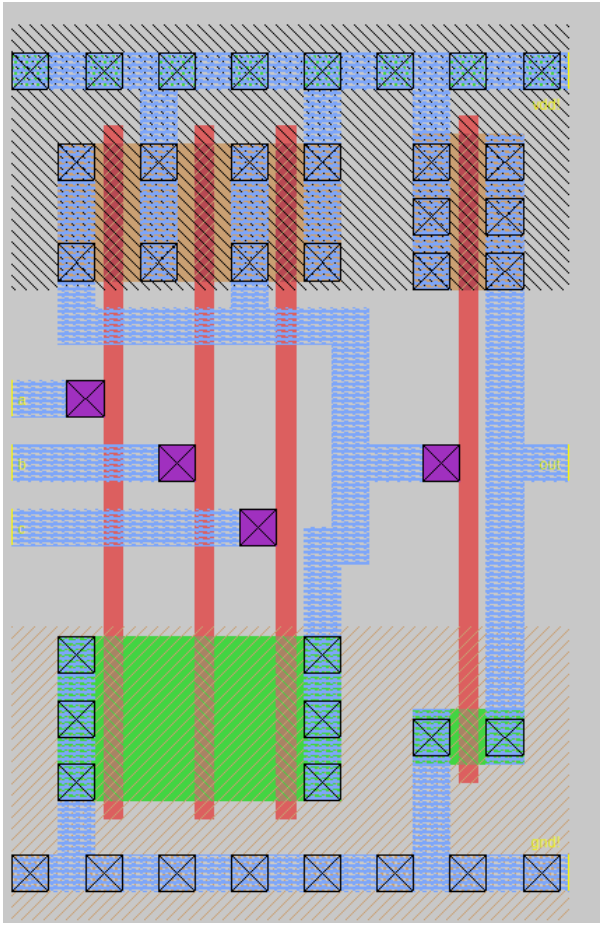
Όμοια έχουμε και για τις υπόλοιπες περιπτώσεις όπου για ένα συγκεκριμένο διάνυσμα εισόδων , έχουμε το output να γίνεται 1 ενώ σε όλα τα υπόλοιπα 0.Άρα η πύλη που πρέπει να σχεδιαστεί είναι μια AND πύλη.

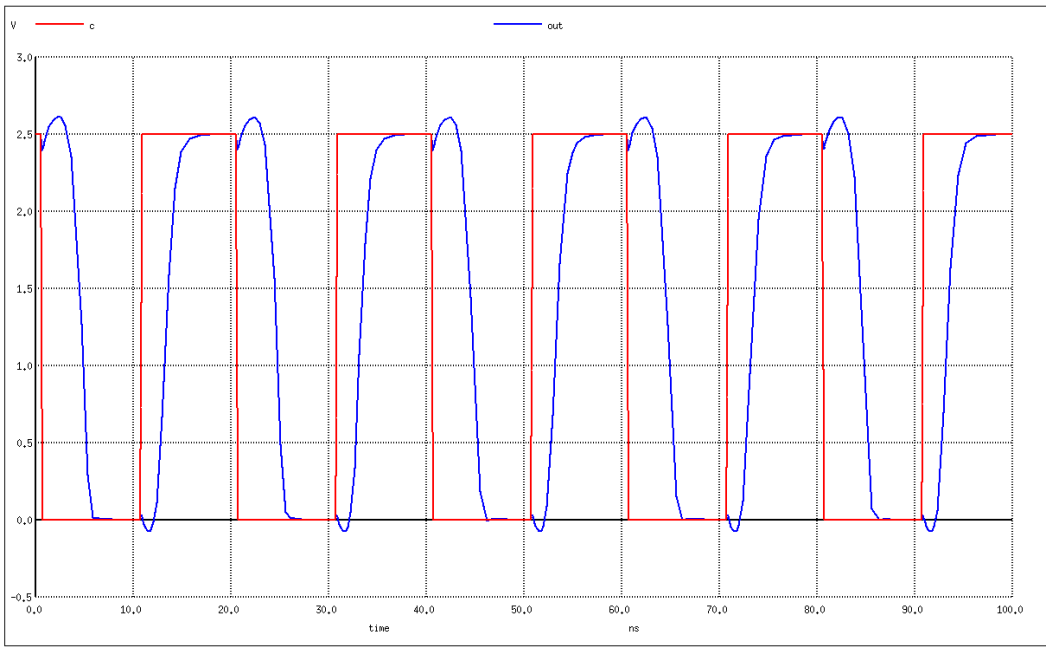
Παρακάτω το σχηματικό:



Τα μεγέθη είναι ορισμένα με βάση το μοντέλο 13kΩ και 31kΩ στο NMOS και PMOS αντίστοιχα,με στόχο τα 6.5kΩ στο καθένα.

Σχεδιασμός στο MAGIC/προσομείωση στο NGSPICE 1bit





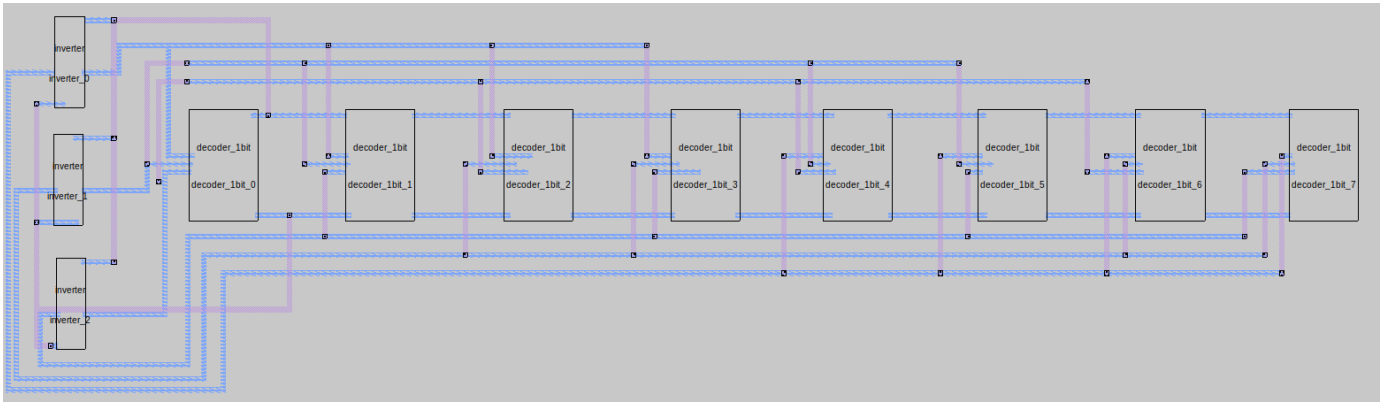
Στη προσομείωση κρατάμε τα A και B στα 2.5V σταθερά και δίνουμε παλμό στο C (από 0 στα 2.5V).

Όπως φαίνεται η πύλη λειτουργεί κανονικά με την έξοδο να γίνεται 1 όταν το C πηγαίνει στα 2.5V.

Οι χρόνοι ανόδου/καθόδου για την πύλη είναι 2.01ns και 1.78ns (αυτοί είναι οι πιο κοντινές τιμές που μπόρεσαν να φτάσουν μεταξύ τους).

Σχεδιασμός 3 σε 8 αποκωδικοποιητή

Παρακάτω το σχέδιο στο MAGIC:

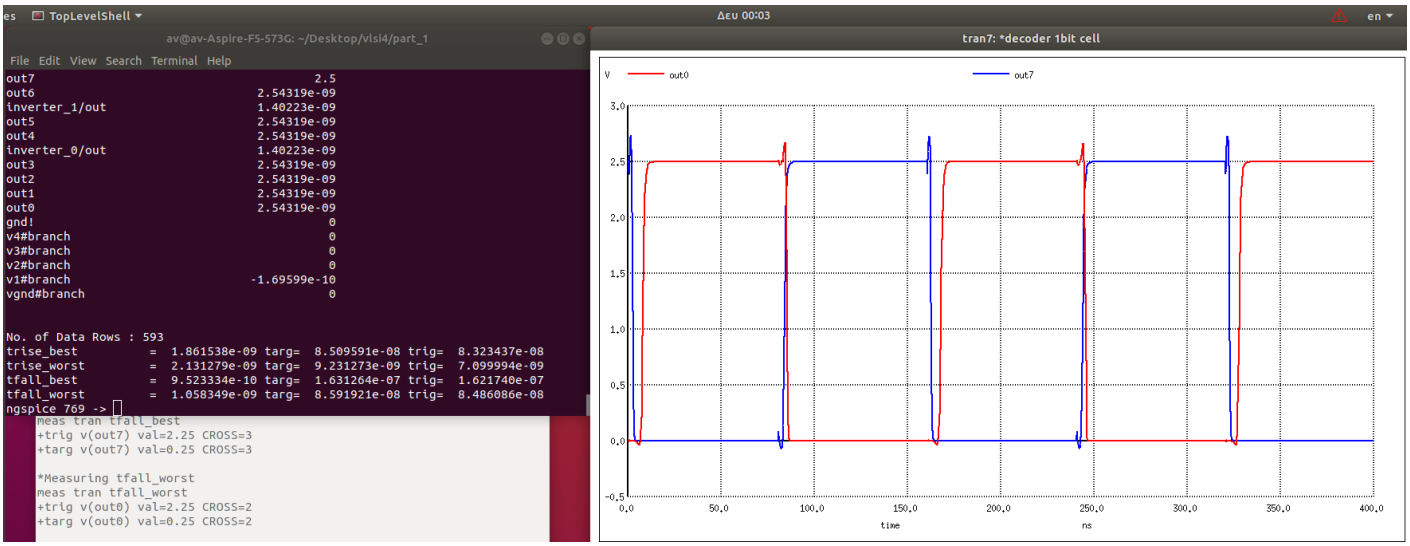


Όπως φαίνεται παραπάνω ,αρχικά έχουμε τους αντιστροφείς οι οποιοί μας δίνουν τα αντιστραμένα σήματα εισόδου.Τα σήματα αυτά θα είναι από την πάνω πλευρά των αποκωδικοποιητών ενώ τα αντίθετα από κάτω.Με αυτόν τον τρόπο δεν έχουμε καλύτερη οργάνωση ,συμμετρία αλλά και δεν χρειάζεται να χρησιμοποιήσουμε πέρα από μέταλλο 2.

Τέλος, οι αντιστροφείς είναι το υποκύκλωμα από την εργασία 1.

Πιο πολλά μπορουν να φανούν ,φυσικά,στο .mag αρχείο .

Προσομείωση του αποκωδικοποιητή στο NGSPICE



Για την προσομείωση δίνουμε ως διανύσματα εισόδου το ‘000’ και το ‘111’.Αυτό το πετυχαίνουμε με να έχουμε παλμού που αλλάζουν ταυτόχρονα κάθε είσοδο (τα A-B-C) από 0->1 και 1->0.Έτσι έχουμε αυτές τις τιμές για τον έλεγχο.

Επιλέχθηκαν αυτές γιατι αποτελούν και τις ακραίες περιπτώσεις του αποκωδικοποιητή.Η είσοδος ‘111’ είναι η καλύτερη περίπτωση μιας και τα σήματα για να ενεργοποιήσουν το 8^{ο} bit του αποκωδικοποιητή έρχονται κατευθείαν από το περιβάλλον δοκιμής ,δηλαδή έχουμε μόνο την καθυστέρηση της AND πύλης.Αντίθετα, με την είσοδο ‘000’ όλες οι είσοδοι πρέπει να περάσουν από τους αντιστροφείς με αποτέλεσμα να έχουμε και την καθυστέρηση των αντιστροφέων μαζί με την πύλη AND.

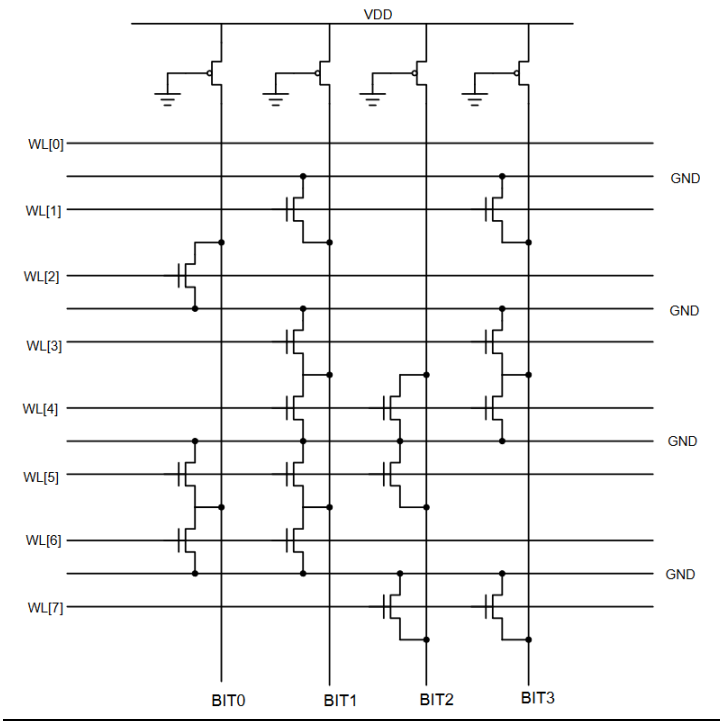
Στη δεξιά εικόνα βλέπουμε τις εναλλαγές των εξόδων (όλες οι άλλες παραμένουν στο 0) .Η out0 για είσοδο ‘000’ και η out7 για είσοδο ‘111’.

Στα αριστερά επιβεβαιώνεται η παραπάνω υπόθεση μας.Για είσοδο ‘000’ ο χρόνος ανόδου/καθόδου είναι 2.13ns/1.05ns αντίστοιχα .Ενώ για είσοδο ‘111’ ο χρόνος ανόδου/καθόδου είναι 1.86ns/0.95ns .

Άσκηση 2

Στη $2^{η}$ άσκηση έχουμε να σχεδιάσουμε μια μνήμη ROM ,8 θέσεων ,4-bit και μορφής NOR.

Σχηματικό μνήμης ROM



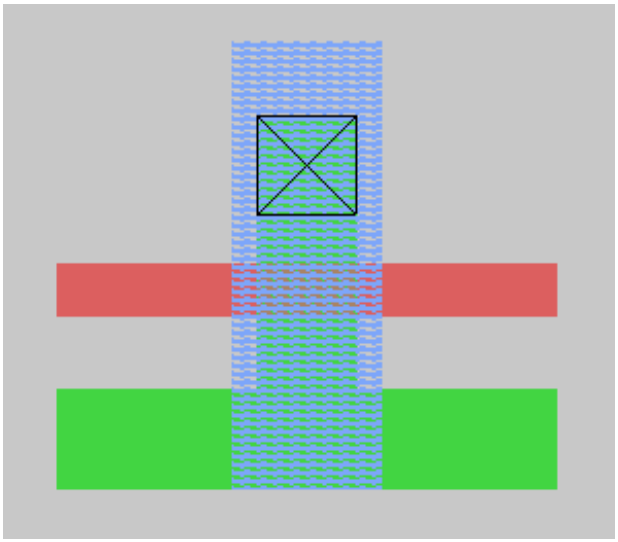
Όπως φαίνεται έχουμε τη διάταξη της μνήμης που ζητείται .Στο πάνω μέρος έχουμε το δίκτυο ανέλκυσης από τα PMOS τρανζίστορ.Στο εσωτερικό της μνήμης στο σημεία που έχουμε transistor έχουμε αποθηκευμένο ‘0’ ενώ απουσία του είναι το λογικό ‘1’.Επιπλέον το MSB είναι το δεξίοτερο bit και όσο κατευθυνόμαστε προς τα αριστερά έχουμε τα LSB της αποθηκευμένης λέξης.

Σχεδιασμός 1bit cell

Αρχικά σχεδιάζουμε το 1-bit μνήμης και σχεδιάζουμε 2 μορφές ,την μορφή που έχει αποθηκευμένο 1 (απουσία transistor-diffusion) και την μορφή που έχει αποθηκευμένο το 0 (παρουσία transistor-diffusion).

Αυτό γίνεται γιατί αν περάσουμε diffusion απευθείας πάνω από το cell θα έχουμε error ,μιας και δεν μπορεί το diffusion να βρίσκεται πάνω από τον πολυπυρίτιο ,το μέταλλο άλλα πρέπει να βρίσκεται στο ίδιο επίπεδο με το diffusion του cell.

Παρακάτω το σχηματικό του 1-bit cell (της μορφής που έχει αποθηκευμένο 0):

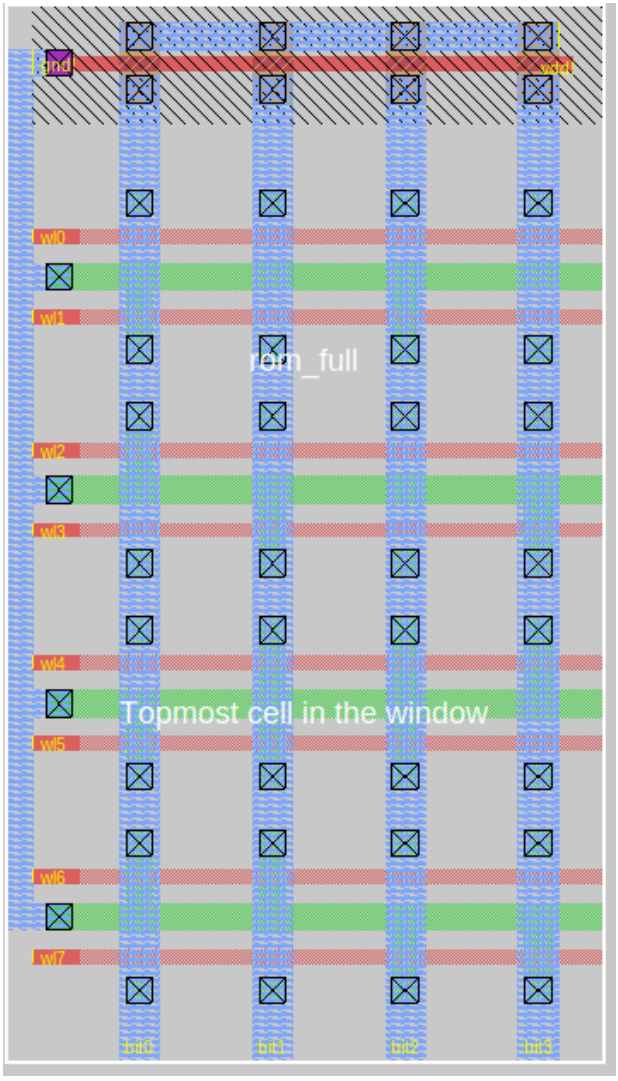


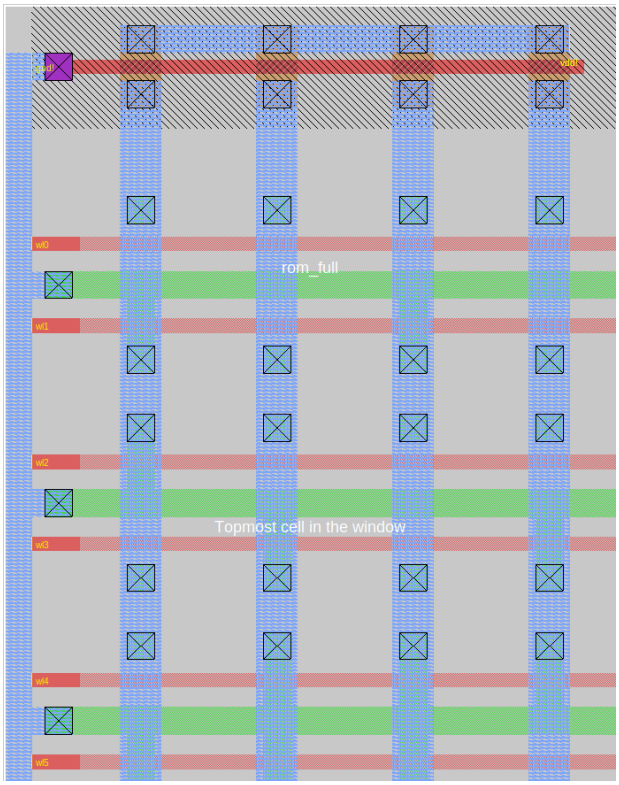
Δεν τοποθετούμε labels μιας (το magic δίνει στα labels των cells ονόματα της μορφής cell/inA ,τα οποία δεν δέχεται το spice όταν ζητάμε να κάνουμε plot).

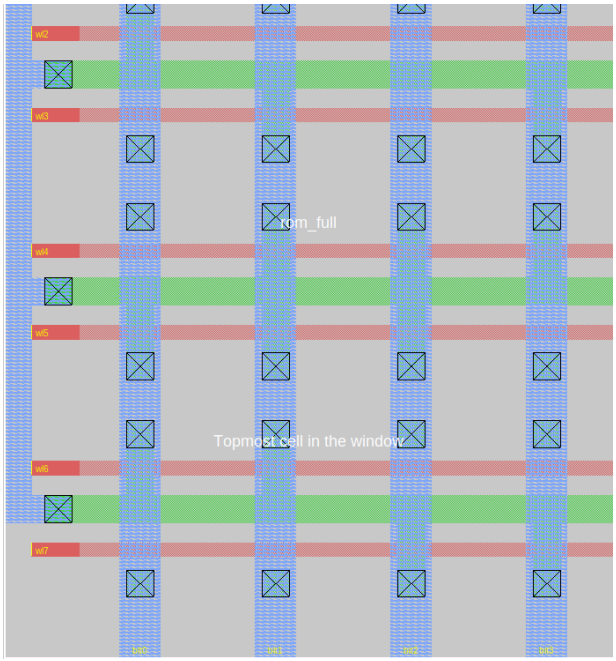
Αντίστοιχα ο άλλος τύπος (που έχει αποθηκευμένο το 1) έχει μόνο diffusion για την γραμμή του ground.

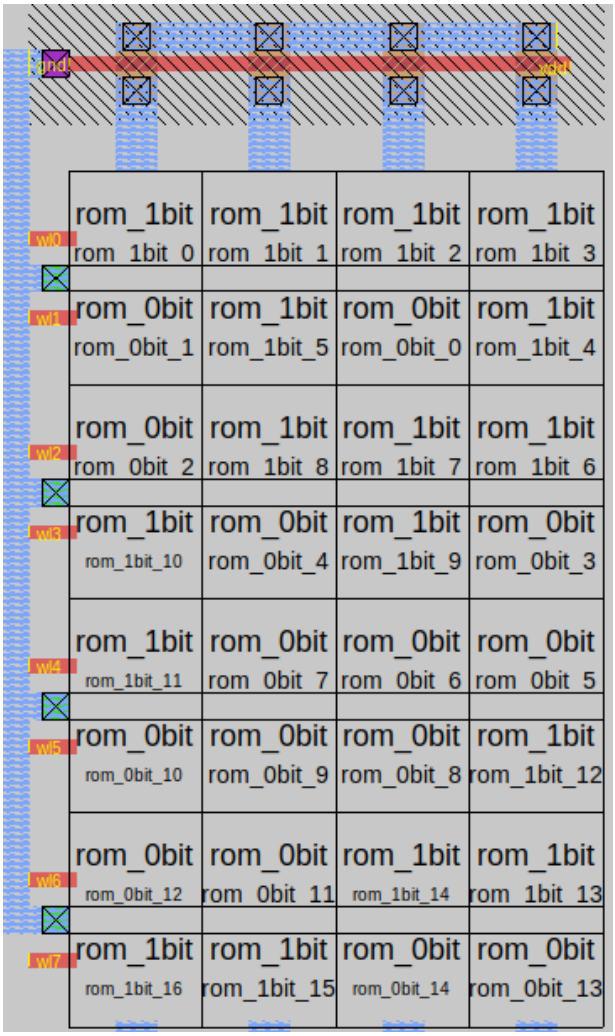
Σχεδιασμός του ζητούμενου κυκλώματος

Παρακάτω εικόνες από τη μνήμη:




rom_full
Topmost cell in the window


rom_full
Topmost cell in the window



Στην τελευταία εικόνα είναι εύκολο να δούμε τι έχουμε αποθηκευμένο σε κάθε bit (rom_0bit = 0 ,rom_1bit = 1) ,επιπλέον η τέρμα δεξιά στήλη τη μνήμης είναι το MSB και όσο κινούμασε προς τα αριστερά πηγαίνουμε προς το LSB.

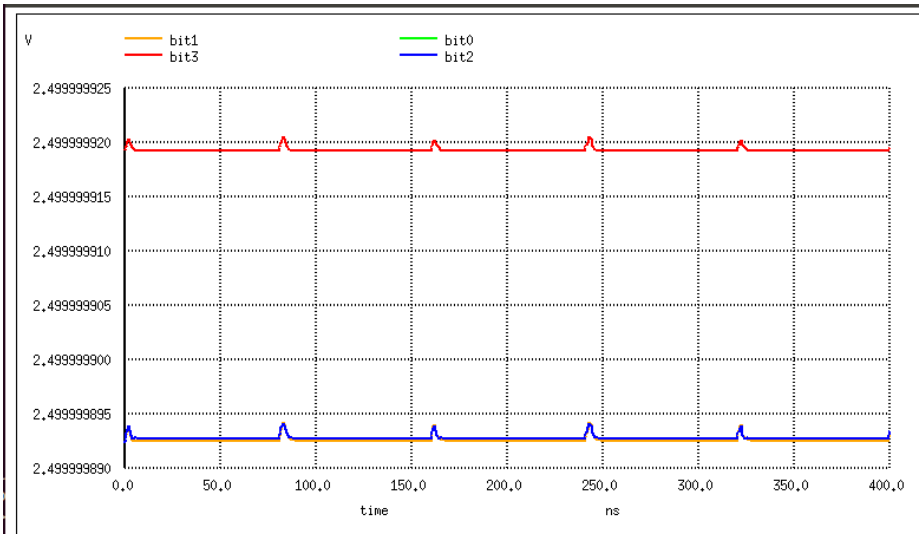
Πάνω από την μνήμη έχουμε ένα δίκτυο ανέλκυσης από PMOS (συνεχώς ενεργά) και στα αριστερά της μνήμης έχουμε τις συνδέσεις για diffusion προς το ground και τα σήματα wl[0-7] για να διαβάζουμε από τη μνήμη.

Επιπλέον άξιο αναφοράς είναι πως τα cells που βρίσκονται στη $2^{η}$,$4^{η}$,$6^{η}$,$8^{η}$ σειρά είναι περιστραμένα κατά 180 μοίρες έτσι ώστε να είναι σωστά τοποθετημένα στο πλέγμα.

<u>Προσομείωση στο NGSPICE</u>

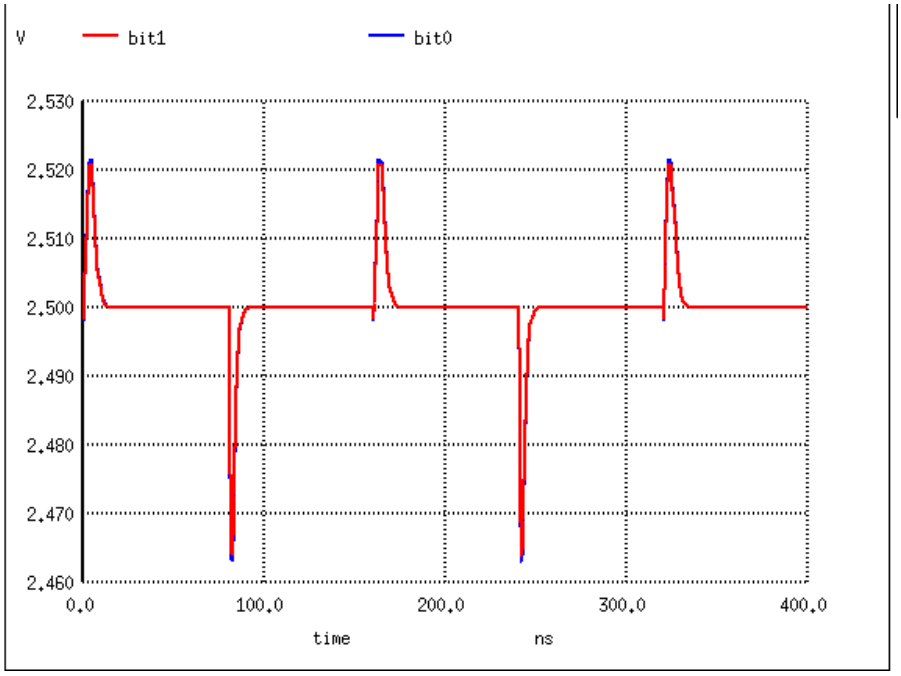
Στο NGSPICE δίνουμε 3 διαφορετικές εισόδους.Κάθε είσοδος ,το αντίστοιχο wordline, θα του δίνουμε ένα παλμό έτσι ώστε να πηγαίνουμε από ενεργή σε ανένεργη κατάσταση.Επιπλέον γνωρίζουμε ότι σε ανένεργη κατάσταση η ROM έχει ως έξοδο το ‘1111’.

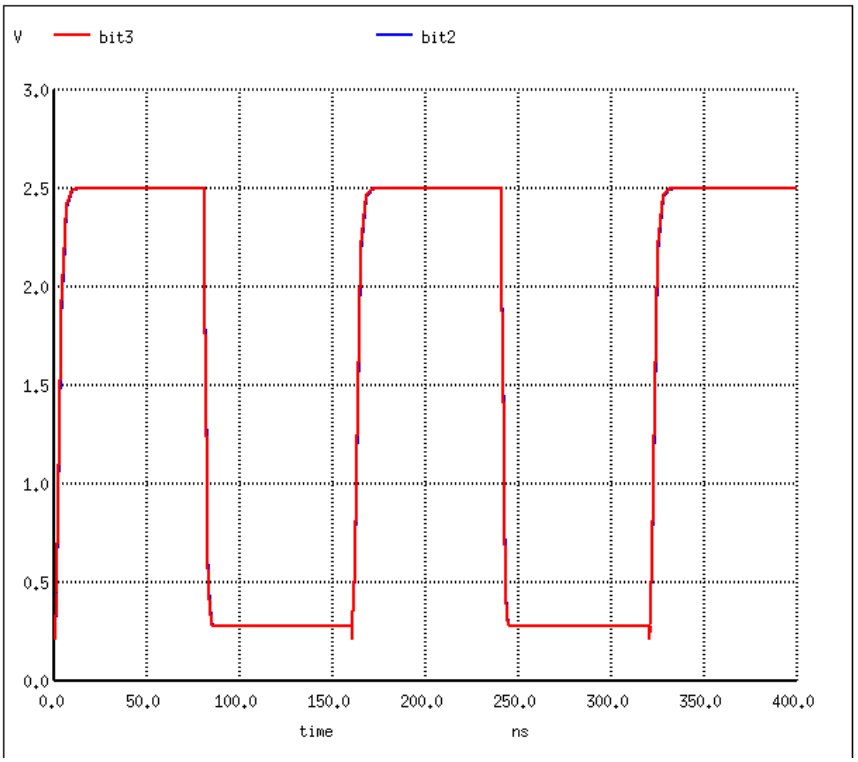
Για διεύθυνση 000 (ενεργοποιούμε wl0):



Οι έξοδοι παραμένουν στο ‘1111’ το οποίο είναι και σωστό.

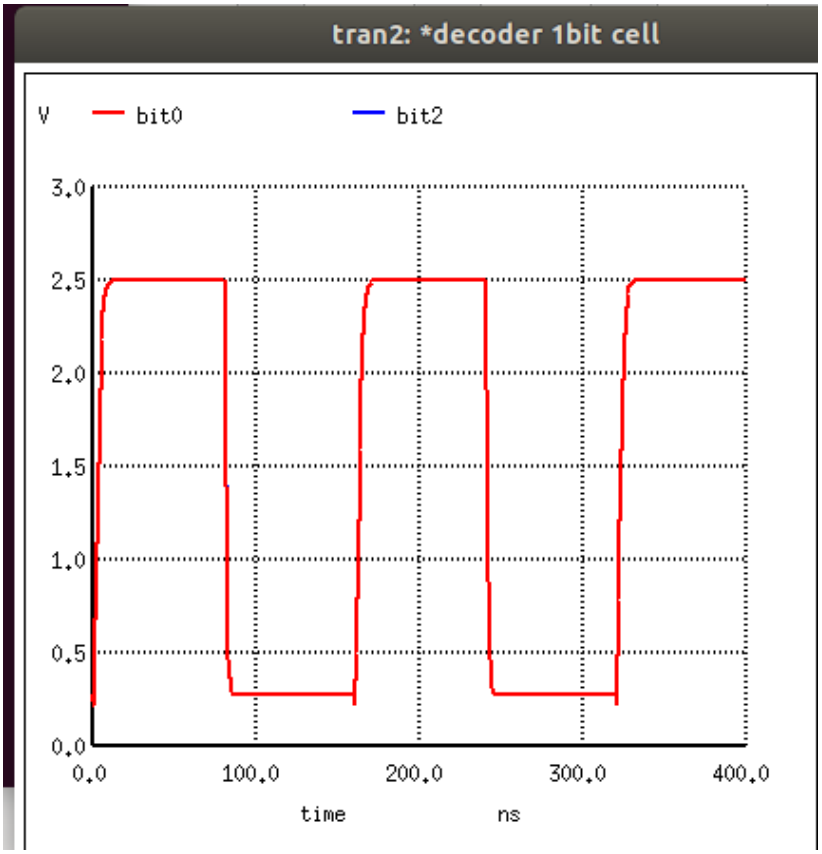
Για διεύθυνση 111 (ενεργοποιούμε wl7):





Εδώ βλέπουμε την έξοδο να γίνεται ‘0011’ .Στα σημεία όπου το wl7 είναι 1 τότε και έχουμε πτώση των bit3,bit2 στο 0.

Για διεύθυνση 001(ενεργοποιούμε το wl1):



Εδώ βλέπουμε την έξοδο να γίνεται ‘1010’ .Στα σημεία όπου το wl7 είναι 1 τότε και έχουμε πτώση των bit2,bit0 στο 0.

Μπορούμε και βλέπουμε σε κάθε προσομείωση ότι το λογικό 0 μας δεν είναι ποτέ στα 0V (η έστω κοντά σε αυτό).Αντιθέτως ,παραμένει κοντά στα 0.27V σε κάθε περίπτωση διαβάσματος με διαφορετική είσοδο.Αυτήν η τάση είναι και το V_T του NMOS.