



Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
ΕΙΣΑΓΩΓΗ ΣΤΗ ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΩΝ VLSI

**5η**

# ***Εργαστηριακή άσκηση***

**Σχεδίαση VLSI με τη γλώσσα περιγραφής υλικού Verilog  
HDL**

**Εξέταση – Επίδειξη: Τρίτη 22/12/2020**

**Έκθεση: Κυριακή 27/12/2020**

**ΑΘΗΝΑ 22/12/2020**

## 1. Εισαγωγή

Η γλώσσα Verilog **HDL** (**H**ardware **D**escription **L**anguage), όπως και η VHDL, ονομάζονται *Γλώσσες Περιγραφής Υλικού (ΓΠΥ)* και χρησιμοποιούνται ευρέως στη βιομηχανία για την περιγραφή ψηφιακών συστημάτων. Προτάθηκαν για να χρησιμεύσουν στην περιγραφή κυκλωμάτων που βασίζονται σε τυποποιημένα κύτταρα-πύλες ώστε να μπορούν να ανταλλάσσονται μεταξύ διαφορετικών σχεδιαστικών ομάδων, χωρίς να περιορίζονται από συγκεκριμένα εργαλεία ή τεχνολογίες υλοποίησης. Αυτό σημαίνει ότι διαθέτουν υψηλό βαθμό αφάιρεσης. Κάτι αντίστοιχο είναι εξαιρετικά δύσκολο όταν χρησιμοποιούνται, για την περιγραφή ψηφιακών συστημάτων, σχηματικά διαγράμματα. Συγκεκριμένα, οι περιγραφές με ΓΠΥ μπορεί να δίνουν διάφορες λεπτομέρειες σχετικά με τη λειτουργία και τη δομή ενός κυκλώματος, ανάλογα με το *αφαιρετικό επίπεδο* στο οποίο περιγράφεται και διαθέτουν ποικιλία εκφραστικών δομών. Το περισσότερο αφαιρετικό επίπεδο είναι το επίπεδο *συμπεριφοράς*, που περιγράφει τη λειτουργία ενός κυκλώματος με τη μορφή πολύπλοκων τελεστών, κυρίως αριθμητικών, που τοποθετούνται σε εντολές ακολουθιακής εκτέλεσης, όπως μια γλώσσα προγραμματισμού. Το δεύτερο σε βαθμό αφαιρετικό επίπεδο είναι το επίπεδο *ροής δεδομένων*, που περιγράφει συνήθως πιο απλές σχέσεις (λογικές πράξεις) μεταξύ σημάτων, που θεωρούνται ότι εκτελούνται ταυτόχρονα (όπως και πρέπει να είναι η λειτουργία ενός κυκλώματος). Το λιγότερο αφαιρετικό επίπεδο είναι το επίπεδο *δομής*, που περιγράφει ιεραρχικά τη δομή του κυκλώματος χρησιμοποιώντας δομικές μονάδες και ορίζοντας σήματα (καλώδια) για τις συνδέσεις μεταξύ τους.

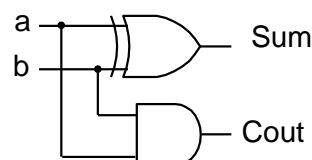
Το πρόγραμμα MICROWIND υποστηρίζει περιορισμένες δυνατότητες σχεδίασης με τη γλώσσα Verilog HDL. Περιλαμβάνει τον compiler μιας γραμμής και τον compiler αρχείου. Τα δύο εργαλεία είναι προσβάσιμα από το menu Compile (Compile one line και Compile Verilog File αντίστοιχα).

Η χρήση του compiler μιας γραμμής είναι απλή. Ο χρήστης έχει τη δυνατότητα να επιλέξει μονάδες μέσα από μια λίστα (drop-down list), που περιλαμβάνει εκφράσεις που αντιστοιχούν σε περιγραφές ροής δεδομένων. Ακόμα μπορεί να γράψει νέες εκφράσεις, οι οποίες αυτόματα εισάγονται στη λίστα. Οι τελεστές που χρησιμοποιούνται στις εκφράσεις είναι οι εξής: & (λογικό γινόμενο - and), | (λογικό άθροισμα - or), ~ (λογική αναστροφή - not) και ^ (λογικό αποκλειστικό άθροισμα - xor). Η λογική αναστροφή επιτρέπεται να εφαρμοστεί μόνο στο σύνολο της έκφρασης που σχεδιάζουμε. Με την επιλογή μιας έκφρασης που περιλαμβάνει μια συγκεκριμένη κυκλωματική μονάδα (π.χ. πύλη xor), το πρόγραμμα σχεδιάζει αυτόματα το αντίστοιχο layout. Με διαδοχικές επιλογές εκφράσεων, τα αντίστοιχα layout εμφανίζονται το ένα δίπλα στο άλλο. Αυτό που απαιτείται για την κατασκευή ενός πολύπλοκου κυκλώματος είναι η σωστή σύνδεση των εισόδων και των εξόδων από το χρήστη. Ο compiler μιας γραμμής είναι τόσο απλός στη χρήση, ώστε ουσιαστικά να μην απαιτεί γνώσεις Verilog HDL για το χειρισμό του.

Ο compiler αρχείου μπορεί να φτιάξει αυτόματα το layout περισσότερο πολύπλοκων κυκλωμάτων. Οι περιγραφές που δέχεται πρέπει να μην είναι ιεραρχικές, δηλαδή να μην χρησιμοποιούνται δομικές μονάδες που ορίζονται από το χρήστη αλλά μόνο πρωτογενείς δομικές μονάδες (primitives) της Verilog HDL. Οι μονάδες αυτές μπορεί να είναι είτε λογικές πύλες είτε τρανζίστορ. Οι περιορισμοί αυτοί ορίζουν ένα μικρό μόνο υποσύνολο της Verilog HDL που μπορεί να χρησιμοποιηθεί στο πρόγραμμα MICROWIND. Παρόλο όμως το μικρό του μέγεθος, το συγκεκριμένο υποσύνολο μπορεί να διευκολύνει σε μεγάλο βαθμό τη διαδικασία σχεδίασης και να χρησιμοποιηθεί σταδιακά, με περισσότερα από ένα αρχεία εισόδου, ή και σε συνδυασμό με τις άλλες μεθόδους σχεδίασης (compiler μιας γραμμής, σχεδιαστικά εργαλεία) στο ίδιο layout.

Οι περιγραφές δομής μπορεί να είναι είτε με λογικές πύλες είτε με τρανζίστορ. Για παράδειγμα, δίνονται παρακάτω περιγραφές με λογικές πύλες ενός ημιαθροιστή, ενός πλήρη αθροιστή και ενός μανδαλωτή RS, μαζί με τα αντίστοιχα σχηματικά διαγράμματα.

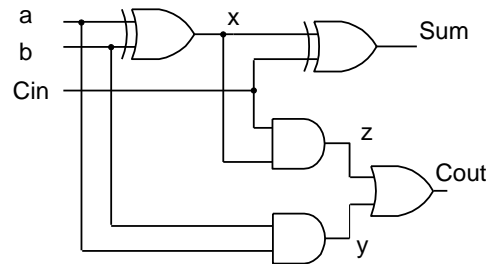
```
module had (Sum,Cout,a,b);
  input a,b;
  output Sum,Cout;
  xor (Sum,a,b);
  and (Cout,a,b);
```



```

endmodule
module fadd (Sum,Cout,a,b,Cin);
    input a,b,Cin;
    output Sum,Cout;
    wire x,y,z;
    xor (x,a,b);
    xor (Sum,x,Cin);
    and (y,a,b);
    and (z,Cin,x);
    or (Cout,y,z);
endmodule

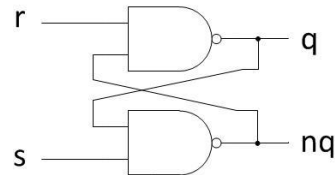
```



```

module rslatch (q,nq,r,s)
    input r,s;
    output q,nq;
    nand nand1(q,r,nq);
    nand nand2(nq,q,s);
endmodule

```



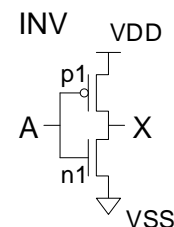
Γράφοντας και αποθηκεύοντας τις παραπάνω περιγραφές (ή και άλλες που μπορείτε να βρείτε ή να κατασκευάσετε) σε αρχείο (μία περιγραφή σε κάθε αρχείο), μπορείτε να πάρετε αυτόματα το layout πολύπλοκων συστημάτων επιλέγοντας παραμέτρους όπως το μήκος και το πλάτος των καναλιών των τρανζίστορ που θα κατασκευαστούν (αφού πρώτα επιλέξετε κατάλληλη τεχνολογία) ή οι ετικέτες (labels) που θα φαίνονται στο layout. Σημειώστε ότι η γλώσσα Verilog HDL περιλαμβάνει, εκτός από αυτές που εμφανίζονται στα παραδείγματα, και τις λογικές πύλες nor, not ενώ κατά σύμβαση, στις πρωτογενείς δομικές μονάδες αναφέρονται πρώτα οι έξοδοι και στη συνέχεια οι εισόδους.

Οι περιγραφές σε επίπεδο τρανζίστορ που δίνονται στα επόμενα παραδείγματα είναι: ενός αναστροφέα, μιας πύλης NOR και μιας πύλης NAND δυο εισόδων.

```

module inverter (Y,A);
    input A;
    output X;
    pmos p1 (X,VDD,A);
    nmos n1 (X,VSS,A);
endmodule

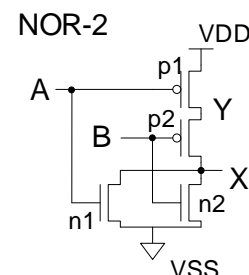
```



```

module nor (X,A,B);
    input A,B;
    output X;
    wire Y;
    pmos p1 (Y,VDD,A);
    pmos p2 (X,Y,B);
    nmos n1 (X,VSS,A);
    nmos n2 (X,VSS,B);
endmodule

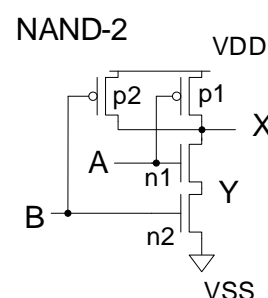
```



```

module nand (X,A,B);
    input A,B;
    output X;
    wire Y;
    pmos p1 (X,VDD,A);
    pmos p2 (X,VDD,B);
    nmos n1 (X,Y,A);
    nmos n2 (Y,VSS,B);
endmodule

```

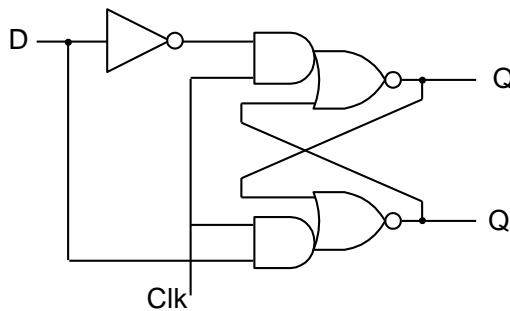


## 2. Τα ζητούμενα της εργαστηριακής άσκησης (όλα σε τεχνολογία cmos90n)

- 1 Να σχεδιασθούν, με την βοήθεια του compiler μιας γραμμής ή/και του compiler αρχείου, (σε τεχνολογία cmos90n) τα layout ενός ημιαθροιστή (H-A), ενός ημιαφαιρέτη (H-S), ενός πλήρη αθροιστή (F-A) και ενός αθροιστή δυο αριθμών των δυο bit που παρέχει στην έξοδό του τις μεταβλητές  $S_0$ ,  $S_1$  και  $S_2$ . Στη συνέχεια ελέγξτε αν τα κυκλώματα (layout) που προκύπτουν από τους compilers είναι σωστά και να γίνει έλεγχος της ορθής λειτουργίας τους μέσω της προσομοίωσης του προγράμματος MICROWIND.

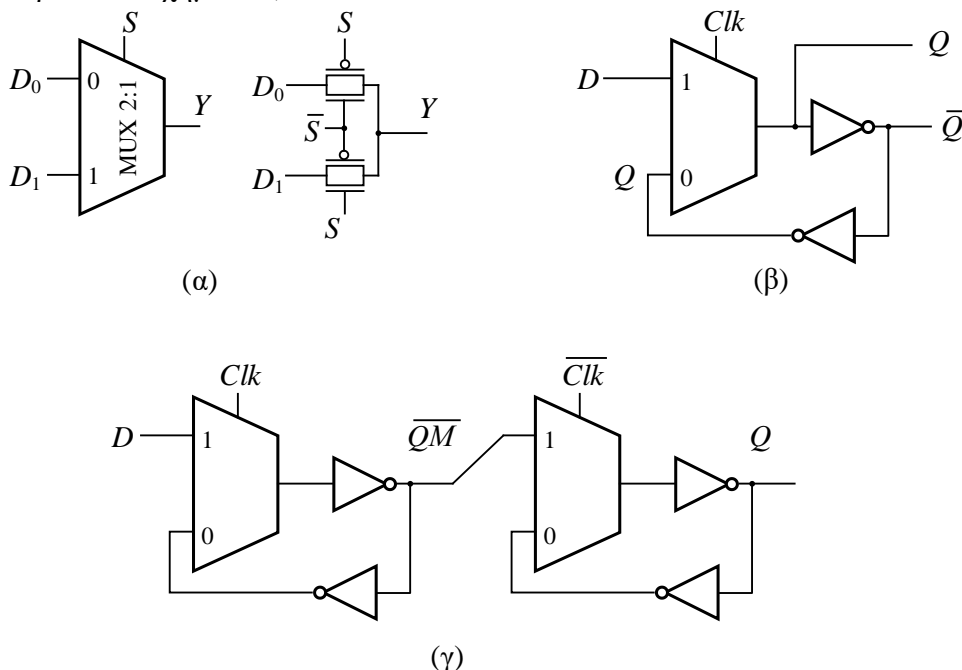
Να κάνετε χρήση των σχεδιάσεων που δίνει το MICROWIND και αυτών που θα βρείτε στον φάκελο Verilog Designs. Εξηγήστε τις περιγραφές και σχολιάστε τα αποτελέσματα των προσομοιώσεων.

- 2 Να σχεδιαστούν είτε με τους compiler Verilog είτε με τον layout editor οι τρεις παρακάτω τύποι καταχωρητών και να γίνει έλεγχος της ορθής λειτουργίας τους μέσω της προσομοίωσης του προγράμματος MICROWIND:
  - Ένας μανδαλωτής D με βάση το παρακάτω Σχήμα 5.1 (προσοχή στην αναγνώριση των σύνθετων πυλών της μορφής  $(AB+C)$  που εμφανίζονται ως AND-NOR συνενωμένες και στην υλοποίησή τους ως σύνθετων πυλών).



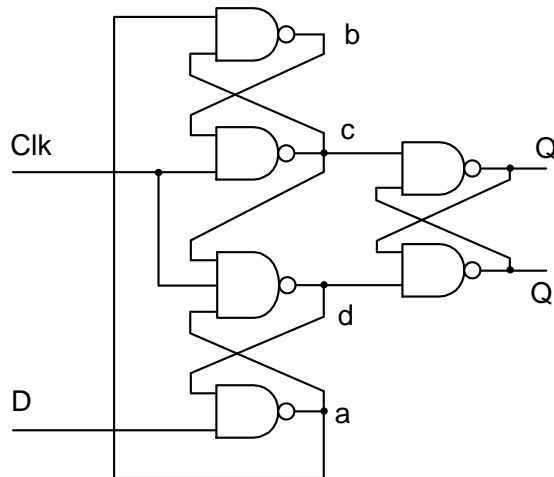
Σχήμα 5.1 Μανδαλωτής

- Ένας ακμοπυροδότητος καταχωρητής με βάση το Σχήμα 1.32 του βιβλίου (βλ. και το παρακάτω Σχήμα 5.2).



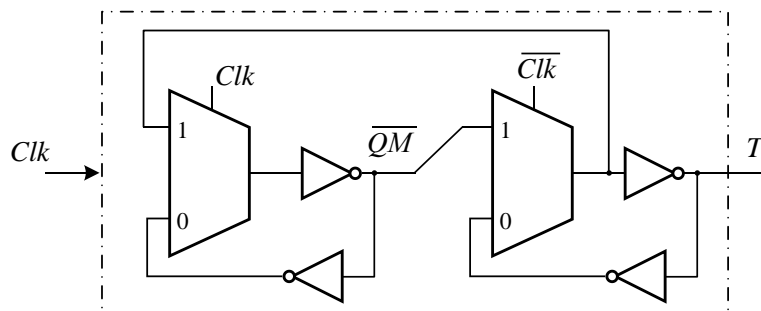
Σχήμα 5.2 (α) Διακόπτης ή Πολυπλέκτης 2 σε 1 (β) Μανδαλωτής (γ) ακμοπυροδότητος καταχωρητής (αλλάζει στον επόμενο κύκλο ρολογιού)

- Ένας θετικά ακμοπυροδότητος καταχωρητής με βάση το Σχήμα 5.3. Στον καταχωρητή αυτό αλλάζει η τιμή εξόδου πάνω στο θετικό μέτωπο του τρέχοντος κύκλου ρολογιού και όχι στο επόμενο όπως στο προηγούμενο σχήμα.



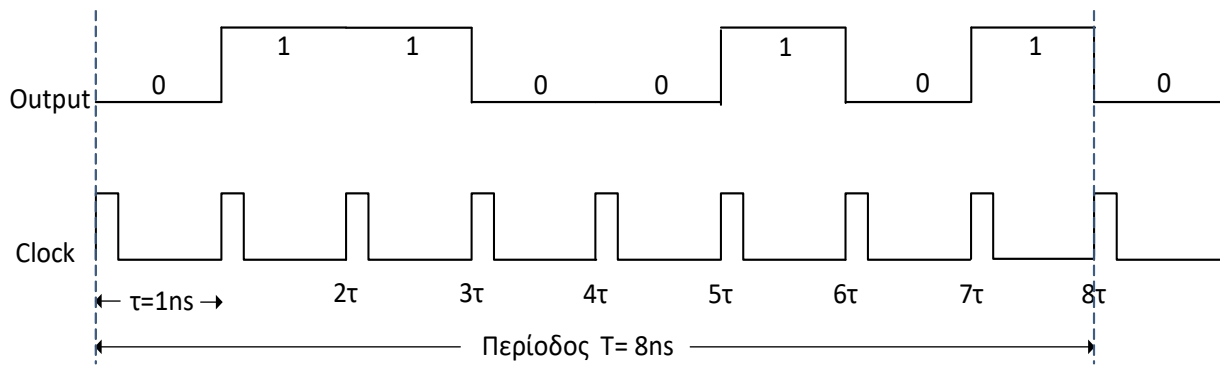
Σχήμα 5.3 Θετικά ακμοπυροδότητος καταχωρητής (αλλάζει στον τρέχοντα κύκλο ρολογιού)

- 3 Να σχεδιάσετε είτε με τους compiler Verilog είτε με τον layout editor ένα T flip-flop χρησιμοποιώντας το κύκλωμα που δίνεται παρακάτω (Σχ. 5.4) αφού εξηγήσετε τη λειτουργία του. Στη συνέχεια υλοποιείτε με την βοήθεια αυτού του κυττάρου έναν απαριθμητή των τριών bit και ελέγξτε τη λειτουργία του. Ποιά είναι η ανώτερη συχνότητα παλμών που μπορεί να απαριθμηθεί με το κύκλωμα αυτό των T flip-flop;



Σχήμα 5.4 Κύκλωμα T flip-flop

- 4 Να σχεδιασθεί είτε με τους compiler Verilog είτε με τον layout editor ένα κύκλωμα που να παράγει την παρακάτω κυματομορφή (Σχ. 5.5). Θεωρούμε ότι μια είσοδος **start/stop** ενεργοποιεί κάθε φορά την έναρξη και τον τερματισμό της περιόδου της. Η έναρξη συγχρονίζεται με το πρώτο θετικό μέτωπο του ρολογιού του συστήματος. Ο τερματισμός θέτει την έξοδο (κυματομορφή) στο λογικό 0 μετά την ολοκλήρωση της περιόδου. Ο έλεγχος του σήματος **start/stop** γίνεται μόνο στα θετικά μέτωπα του ρολογιού. Όταν **start/stop=1** έχουμε συνεχή επανάληψη της περιοδικής κυματομορφής. Όταν **start/stop=0** ολοκληρώνεται η τρέχουσα περίοδος (των 8 κύκλων ρολογιού) και η έξοδος (κυματομορφή) τίθεται στο λογικό 0.



**Σχήμα 5.5** Κυματομορφή Μετρητή

Η σχεδίαση να βασιστεί στον προηγούμενο μετρητή (3<sup>ο</sup> ζητούμενο) των τριών bit και η έξοδος (κυματομορφή) να προκύπτει ως λογική συνάρτηση των εξόδων των τριών T flip-flop. Ο έλεγχος για το σταμάτημα και το ξεκίνημα της κυματομορφής να γίνεται με την παρεμπόδιση ή όχι του ρολογιού είτε με άλλη δική σας τεχνική. Η συνθήκη γι' αυτό μπορεί να βασίζεται στο σήμα **start/stop** και στις εξόδους των τριών T flip-flop. Η πληροφορία για την κατάσταση που βρίσκεται το κύκλωμα (εντολή επανάληψης ή σταματήματος) μπορείτε να την ενταμιεύσετε σε έναν καταχωρητή (Σχήμα 5.2 ή Σχήμα 5.3 όποιο κρίνετε κατάλληλο).