



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ψηφιακά Συστήματα VLSI

4η Εργαστηριακή Άσκηση

Υλοποίηση FIR Φίλτρου

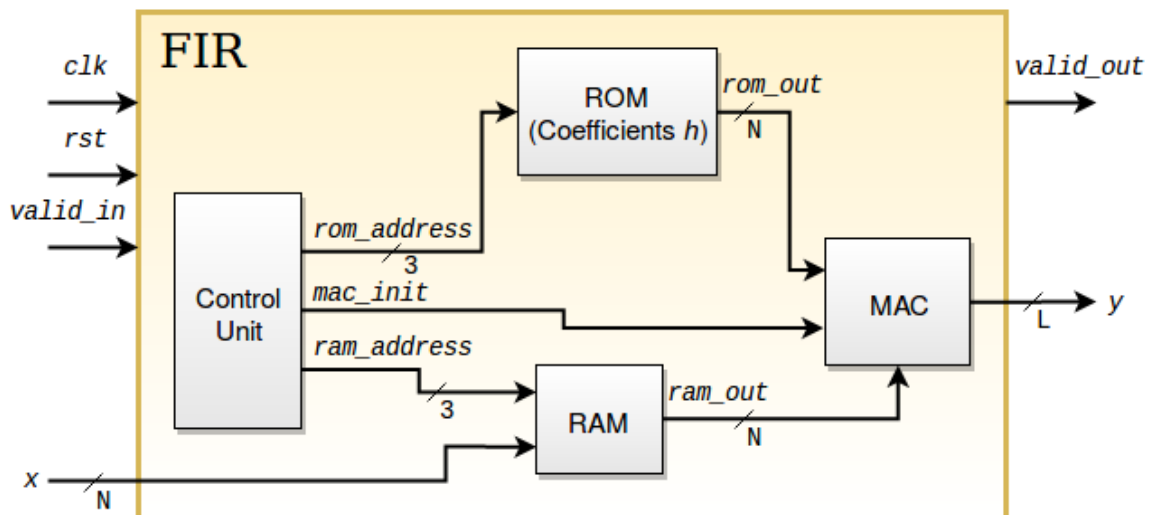
Στη γενική περίπτωση, η σχέση εισόδου - εξόδου ενός FIR φίλτρου είναι η ακόλουθη:

$$y[n] = \sum_{k=0}^M h[k] x[n-k] = h[0] x[n] + h[1] x[n-1] + \dots + h[M] x[n-M] \quad (1)$$

όπου

- M είναι η **τάξη του φίλτρου**
- $y[n]$ είναι η **εξοδος του φίλτρου** τη διακριτή χρονική στιγμή n
- $h[k]$ είναι ο **k -οστός συντελεστής του φίλτρου**, με $k = 0, 1, 2, \dots, M$
- $x[n]$ είναι η **τιμή του σήματος εισόδου** τη διακριτή χρονική στιγμή n

Στα πλαίσια αυτής της εργαστηριακής άσκησης θα υλοποιήσετε ένα **8-tap FIR φίλτρο**. Μία προτεινόμενη αρχιτεκτονική αυτού του φίλτρου, σύμφωνα με την εξίσωση (1), φαίνεται στο ακόλουθο σχήμα όπου το εύρος των δεδομένων είναι N bits. Στην άσκηση η υλοποίηση να γίνει για **$N=8$ bits** εύρος δεδομένων x .



Τα βασικά modules της αρχιτεκτονικής του φίλτρου είναι τα ακόλουθα:

1. **MAC - Μονάδα Πολλαπλασιασμού με Συσσώρευση (Multiplier Accumulator Unit):** υπολογίζει την έξοδο του φίλτρου y , εκτελώντας την ακόλουθη πράξη:

$$a \leftarrow a + b \times c$$

Δέχεται ως είσοδο τους **σταθερούς** συντελεστές του φίλτρου (rom_out), τις τιμές του σήματος εισόδου (ram_out), καθώς και ένα σήμα αρχικοποίησης (mac_init) που υποδηλώνει την αρχικοποίηση της συσσώρευσης πριν τον υπολογισμό κάθε νέας τιμής του y .

Η υλοποίηση αυτής της μονάδας να γίνει σε **behavioral** περιγραφή και με τη χρήση των τελεστών '+' και '*' που υποστηρίζονται από τη βιβλιοθήκη **std_logic_unsigned**.

2. **ROM:** έχει αποθηκευμένους του **σταθερούς** συντελεστές του φίλτρου h .

Δέχεται ως είσοδο μία διεύθυνση ($rom_address$) και δίνει στην έξοδο τον αντίστοιχο συντελεστή που είναι αποθηκευμένος σε αυτή (rom_out).

Η υλοποίηση αυτής της μονάδας (αρχικοποιημένη με τους συντελεστές) σας παρέχεται έτοιμη και πρέπει να την ενσωματώσετε κατάλληλα στη συνολική αρχιτεκτονική του φίλτρου.

3. **RAM:** αποθηκεύει την παρούσα τιμή του σήματος εισόδου x , καθώς και τις 7 προηγούμενες, που είναι απαραίτητες για τον υπολογισμό της εξόδου y σύμφωνα με την εξίσωση (1).

Δέχεται ως είσοδο μία διεύθυνση ($ram_address$) και δίνει στην έξοδο την αντίστοιχη τιμή του σήματος εισόδου που είναι αποθηκευμένη σε αυτή (ram_out).

Κατά τη διάρκεια της λειτουργίας εγγραφής, εγγράφεται η νέα τιμή του σήματος εισόδου x στην πρώτη θέση της μνήμης και οι ήδη αποθηκευμένες 8 τιμές ολισθαίνουν κατά μία θέση με αποτέλεσμα η παλαιότερη χρονικά να διαγράφεται.

Για την υλοποίηση αυτής της μονάδας σας παρέχεται έτοιμος ο κώδικας μίας write-first μνήμης (τα νέα δεδομένα που εγγράφονται σε συγκεκριμένη διεύθυνση βγαίνουν και στην έξοδο της μνήμης στον ίδιο κύκλο ρολογιού). Βασιζόμενοι σε αυτή την μονάδα, θα πρέπει να προσθέσετε ένα μικρό κομμάτι κώδικα το οποίο θα υλοποιεί την μετατόπιση των δεδομένων της μνήμης κατά τη διάρκεια της εγγραφής σύμφωνα με την παραπάνω περιγραφή. Αρχικά όλες οι θέσεις της μνήμης είναι αρχικοποιημένες στο '0' μιας και δεν έχει γίνει καμία εγγραφή.

4. **Control Unit - Μονάδα Ελέγχου:** αποτελεί τη μονάδα που ελέγχει και καθορίζει τη λειτουργία του φίλτρου:

- Παράγει το σήμα αρχικοποίησης του MAC (mac_init).
- Διευθυνσιοδοτεί ταυτόχρονα τις μνήμες ROM και RAM για τη λειτουργία ανάγνωσης των αντίστοιχων τιμών μέσω των σημάτων $rom_address$ και $ram_address$ αντίστοιχα. Σε κάθε κύκλο ρολογιού δίνει την επόμενη διεύθυνση των μνημών για ανάγνωση.

Η υλοποίηση αυτής της μονάδας να βασιστεί σε έναν μετρητή (up-counter) ο οποίος θα χρησιμοποιηθεί για τη διευθυνσιοδότηση των μνημών ROM και RAM. Η υλοποίηση της μονάδας αυτής να γίνει σε **behavioral** περιγραφή και να χρησιμοποιηθεί η βιβλιοθήκη **std_logic_unsigned** για την υλοποίηση του μετρητή (χρησιμοποιώντας τον τελεστή '+').

Η λειτουργία του φίλτρου θα είναι η ακόλουθη:

- Δέχεται μία νέα μη προσημασμένη (unsigned) τιμή εισόδου x κάθε 8 κύκλους ρολογιού.

- Κάθε νέα τιμή της εξόδου y παράγεται με 8 κύκλους ρολογιού διαφορά από την προηγούμενη.
- Για κάθε νέα είσοδο x δέχεται ένα σήμα εγκυρότητας εισόδου (*valid_in*) το οποίο υποδηλώνει την εγκυρότητα της συγκεκριμένης εισόδου.
- Για κάθε έγκυρη τιμή της εξόδου y ενεργοποιείται ένα σήμα εγκυρότητας εξόδου (*valid_out*).
- Περιλαμβάνει ασύγχρονο reset το οποίο αρχικοποιεί τη μονάδα Control Unit, τη μνήμη RAM με '0' καθώς και το σήμα εγκυρότητας εξόδου *valid_out*.

Υπενθύμιση: Για την σωστή αναπαράσταση του αποτελέσματος του πολλαπλασιασμού 2 αριθμών των X bits, χρειάζονται $2X$ bits, ενώ για την πρόσθεση/αφαίρεση $(X+1)$ bits.

Σημείωση: Το πλήθος των L bits της εξόδου y του FIR θα πρέπει να λαμβάνει υπόψη όλα τα απαραίτητα bits για την αναπαράσταση του ορθού αποτελέσματος, συμπεριλαμβανομένων και των περιπτώσεων υπερχείλισης.

Προσοχή: Κατά την ενσωμάτωση όλων των ξεχωριστών μονάδων για τη δημιουργία της συνολικής αρχιτεκτονικής του φίλτρου θα πρέπει να ληφθεί υπόψη ο κατάλληλος συγχρονισμός μεταξύ τους.

ΖΗΤΟΥΜΕΝΑ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

- 1) Να υλοποιήσετε το παραπάνω φίλτρο λαμβάνοντας υπόψη όλες τις ζητούμενες λειτουργίες του.
- 2) Να δημιουργήσετε testbench για τον έλεγχο της ορθής λειτουργίας του φίλτρου για $N=8$. Το testbench να περιλαμβάνει τον έλεγχο όλων των λειτουργιών του φίλτρου.
- 3) Να καταγράψετε τους πόρους (resources) που χρησιμοποιήθηκαν για την υλοποίηση του φίλτρου καθώς και τη συχνότητα λειτουργίας του FPGA ελέγχοντας το κρίσιμο μονοπάτι.