



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

Ψηφιακά Συστήματα VLSI

Άσκηση στο High Level Synthesis

Γενικές οδηγίες

Οδηγίες χρήσης του Vivado HLS και αναλυτική παρουσίαση των optimization directives είναι διαθέσιμα στα εξής user guide και κεφάλαια:

- https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug871-vivado-high-level-synthesis-tutorial.pdf (Chapter 1,2,3,6,7)
- https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_2/ug902-vivado-high-level-synthesis.pdf

Στα παρακάτω design στο HLS να επιλεχθεί Target clock 10ns.

Στο Ζητούμενο 1 να επιλεχθεί το Part xc7z020clg400-1.

Στο Ζητούμενο 2 να επιλεχθεί το Part xc7z020clg484-1.

Τα αρχεία για το Ζητούμενο 2 βρίσκονται στο HELIOS στο path Υλικό/DVLSI_2022_Lab/HLS_bonus_project.

Για απορίες μπορείτε να στείλετε στο konstantina@microlab.ntua.gr.

Ζητούμενο 1

Δίνεται το fir φίλτρο που υλοποιήσατε στην Εργαστηριακή Άσκηση 4 και το fir φίλτρο που παρουσιάζεται στο Κεφάλαιο 2 του User Guide 871.

- 1) Τροποποιήστε το HLS fir φίλτρο ώστε να έχει τον ίδιο αριθμό taps και τα ίδια coefficient με το φίλτρο που υλοποιήσατε στο Ζητούμενο 1 της Εργαστηριακής Άσκησης 4. Να ελεγχθεί η λειτουργία του με το testcase:

input: 40, 248, 245, 124, 204, 36, 107, 234, 202, 245, 0, 0, 0, 0, 0, 0, 0

output: 40, 328, 861, 1518, 2379, 3276, 4280, 5518, 6598, 6011, 5203,
5239, 4431, 4931, 4756, 3331, 1960

Στην αναφορά να συμπεριλάβετε:

- τις αλλαγές στον δοθέν κώδικα
- screenshot από την εκτέλεση του φίλτρου

- 2) Τροποποιήστε το νέο HLS fir φίλτρο ώστε το input να είναι 8-bit και οι ενδιάμεσοι υπολογισμοί να χρησιμοποιούν τον ελάχιστο αριθμό bit. Να χρησιμοποιήσετε τα arbitrary precision types που παρέχει το HLS (UG871).

Στην αναφορά να συμπεριλάβετε:

- τις αλλαγές στον δοθέν κώδικα
- συγκριτικό πίνακα του target και estimated clock, performance (latency) και resource utilization (%) για τις δύο εκδοχές του κώδικα.

- 3) Συγκρίνετε την υλοποίηση του ερωτήματος 2 με το φίλτρο του Ζητήματος 1 της Εργ. Άσκησης 4.

Στην αναφορά να συμπεριλάβετε:

- το ρολόι που επιτυγχάνεται, το latency και resource utilization σε συγκριτικό πίνακα.

Ζητούμενο 2

Οι Μηχανές Διανυσμάτων Υποστήριξης(Support Vector Machines -SVM) είναι μοντέλα επιβλεπόμενης μάθησης που εκπαιδεύονται με ένα μεγάλο σύνολο δεδομένων και είναι κατάλληλα για την ταξινόμηση νέων εισόδων σε δύο υποψήφιες κλάσεις συμπληρωματικές μεταξύ τους. Το σύνολο εκπαίδευσης (training set) αποτελείται από διανύσματα με συγκεκριμένα χαρακτηριστικά, καθένα από τα οποία διαθέτει μια ετικέτα δηλωτικής της κλάσης στην οποία ανήκει. Ένα σύνολο από άλλα διανύσματα με τα ίδια χαρακτηριστικά και γνωστές τις ετικέτες (testing set) χρησιμοποιείται για να ελεγχθεί η ακρίβεια της πρόβλεψης.

Στα πλαίσια της άσκησης θα χρησιμοποιηθεί ένα trained SVM, που ταξινομεί τις εισόδους στην κλάση που ανήκει καθεμία με βάση την εξίσωση:

$$class = sgn \left(\sum_{i=1}^{N_{sv}} (coef_i * exp(-\gamma * | \underline{x} - \underline{supvector}_i |^2)) - b \right)$$

όπου, \underline{x} το διάνυσμα προς ταξινόμηση

N_{sv} , ο αριθμός των support vector που προκύπτουν στο τέλος του training
 $suprvector_i$, το i -οστό support vector
 γ, b σταθερές που προκύπτουν από το training
 $coef_i$, το i -οστό coefficient που αντιστοιχεί στο i -οστό support vector

Για τις δύο κλάσεις χρησιμοποιείται η κωδικοποίηση $\{-1, 1\}$ αναλόγως αν το πρόσημο του αθροίσματος είναι αρνητικό ή θετικό αντίστοιχα.

Στον ακόλουθο πίνακα δίνονται οι τιμές του μοντέλου ενώ επισυνάπτονται σε ξεχωριστά αρχεία τα support vector, τα coefficient, το testing test και το annotation file το οποίο περιέχει το σωστό output του classification.

Μεταβλητή	Τιμή
αριθμός support vector N_{sv}	1222
διάσταση διανυσμάτων D_{sv}	18
b	2.8180
γ	8
ακρίβεια test set	99.5%

- 1) Να γραφεί κώδικας που υλοποιεί το classification με βάση την παραπάνω εξίσωση, να ελεγχθεί η ακρίβεια του classification χρησιμοποιώντας τα αρχεία testing_set.csv, annotation.csv και να γίνει synthesis του κώδικα.

Η ακρίβεια ορίζεται ως το ποσοστό των προβλέψεων/label που είναι ίδιες με το αντίστοιχο label του annotation.csv.

Στην αναφορά να επισυνάψετε:

- τον κώδικα
- την ακρίβεια των αποτελεσμάτων
- screenshot του synthesis report.

Υπόδειξη: Να χρησιμοποιήσετε ένα διπλά φωλεωμένο loop για να κάνετε iterate στα support vector και στα feature αυτών στο εξωτερικό και εσωτερικό loop αντίστοιχα.

- 2) Βελτιστοποιήστε την υλοποίηση του κώδικα στο fpga εφαρμόζοντας το UNROLL DIRECTIVE στο εσωτερικό loop για unroll factor ίσο με 9.

Εξετάστε το scheduling διάγραμμα που προκύπτει και εντοπίστε πιθανές μη αποδοτικές χρονοδρομολογήσεις πράξεων ανεξάρτητων μεταξύ τους.

Επαναλάβετε τη διαδικασία υλοποιώντας το loop unrolling manually (γράφοντας νέο κώδικα) και βέλτιστα ώστε να εκτελούνται ταυτόχρονα πράξεις χωρίς data dependencies μεταξύ τους.

Στην αναφορά να συμπεριλάβετε:

- εξήγηση (με σχηματικό διάγραμμα κατά προτίμηση) της υλοποίησης του manual unrolling
- συγκριτικό πίνακα για το clock (target & estimated), το performance (latency) και resource utilization % (bram,ff,lut,dsp) για τους δύο τρόπους unrolling
- screenshot του scheduling στο οποίο να φαίνεται η διαφορά των δύο υλοποιήσεων

- 3) Χρησιμοποιήστε την υλοποίηση του προηγούμενου ερωτήματος. Εφαρμόστε επιπλέον το ARRAY_PARTITION directive όπως κρίνετε απαραίτητο στους σχετικούς πίνακες.

Στην αναφορά να συμπεριλάβετε:

- εξήγηση της επιλογής σας για το array partition scheme που υλοποιήσατε
- συγκριτικό πίνακα για το clock (target & estimated), το performance (latency) και resource utilization % (bram,ff,lut,dsp) για τα δύο configuration
- screenshot των δύο διαφορετικών τρόπων scheduling των δύο configuration και εξήγηση της διαφοράς των διαγραμμάτων

- 4) Υλοποιήστε δύο optimized SVM kernel στο FPGA που εκτελούνται παράλληλα*. Ομοίως αυξάνοντας κάθε φορά κατά ένα τον αριθμό των SVM, υλοποιήστε όσα instances SVM χωρά το FPGA.

Θεωρήστε ότι κάθε kernel επεξεργάζεται και ταξινομεί ένα διαφορετικό input feature vector.

Στην αναφορά να συμπεριλάβετε:

- τον κώδικα που υλοποιήσατε για τα 2 instances με τα directive ως #pragma.
- έναν πίνακα με το clock (target & estimated), latency, resource utilization % (bram,ff,lut,dsp) στην περίπτωση που έχει υλοποιηθεί ένας και δύο SVM kernel.
- ένα κοινό διάγραμμα για το latency καθώς αυξάνονται τα instances των SVM

- ένα κοινό διάγραμμα για το resource utilization % (πχ bar plot) καθώς αυξάνονται τα instances των SVM

* Υπόδειξη: Χρησιμοποιείτε το *DATAFLOW DIRECTIVE*.