Σύνθεση υψηλού επιπέδου για τη σχεδίαση ψηφιακών ολοκληρωμένων κυκλωμάτων

4ο σετ ασκήσεων

'Ασκηση 1η

Η συνάρτηση compute_row_sum υπολογίζει το άθροισμα κάθε γραμμής του πίνακα a.

```
void compute_row_sum (short a[N][M], short row_sum[N]) {
  for (int i=0; i < N; i++) {
    row_sum[i]=0;
    for (int j=0; j < M; j++) {
      row_sum[i] += a[i][j];
    }
  }
}
```

Θα θέλαμε να υλοποιήσουμε αυτή τη συνάρτηση μέσω HLS σε υλικό με τις εξής προϋποθέσεις:

1. Η διεπαφή της υπομονάδας θα περιλαμβάνει **δύο χωριστές διεπαφές μνήμης** των 16bit η καθεμία. Μία για την προσπέλαση των στοιχείων του πίνακα a (ο πίνακας είναι αποθηκευμένος μονοδιάστατα με τα στοιχεία του να είναι διαδοχικά τοποθετημένα σε μία μνήμη - αυτό είναι η τυπική επιλογή) και μία για την προσπέλαση των στοιχείων του πίνακα row_sum.
2. Το κύκλωμα μας στοχεύει στα 500MHz στην τεχνολογία των 45nm
3. Το κύκλωμα μας να περνάει επιτυχημένα από τη προσομοίωση της RTL μέσα από το Catapult (αυτόματη σύγκριση του testbench σε C++ με την RTL). Θα σας δοθεί έτοιμο ενδεικτικό παράδειγμα.
4. Να πετυχαίνει τη μικρότερη δυνατή καθυστέρηση (latency) με τη χρήση loop pipeline. Ποιο είναι το initiation interval (II) που πετυχαίνετε και γιατί; Τι περιορίζει τις επιλογές σας;
5. Μπορείτε να μειώσετε τις προσπελάσεις της μνήμης χρησιμοποιώντας έξυπνα τοπικές μεταβλητές. Παράλληλα αυτό θα σας επιτρέψει να βελτιώσετε το loop pipeline πετυχαίνοντας II = 1. Γιατί μπορεί να συμβεί αυτό; Δώστε τον αναμορφωμένο κώδικα σας.

Πριν έρθετε στο εργαστήριο θα έχετε τον κώδικα σας έτοιμο στο server. Θα έχετε ετοιμάσει 2 slides που θα δείχνετε με screenshots τις απαντήσεις σας στα ερωτήματα 4 και 5 (ή αντίστοιχα δικά σας ευανάγνωστα σχήματα).

Τέλος σιγουρευτείτε ότι έχετε λειτουργικό μικρόφωνο καθώς στο εργαστήριο θα ελεγχθεί η δουλειά του καθενός από την κάθε ομάδα χωριστά.