ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Σχεδίαση Συστημάτων Υλικού - Λογισμικού

Εργαστήριο 1
Εξοικείωση με το εργαλείο Vivado HLS

Ν. ΤΑΜΠΟΥΡΑΤΖΗΣ - Π. ΜΟΥΣΟΥΛΙΩΤΗΣ

Διδάσκων: Ιωάννης Παπαευσταθίου

Version 0.1

Νοέμβριος 2020

4. Σχεδίαση H/W accelerator χρησιμοποιώντας το Vivado HLS (80%)

Χρησιμοποιώντας το Vivado HLS, σχεδιάστε το hardware accelerator MATRIX_MUL, ο οποίος θα υπολογίζει το γινόμενο 2 πινάκων όπως περιγράφονται στη συνέχεια.

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{pmatrix}, \qquad B = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mp} \end{pmatrix},$$

Όπως φαίνεται στη παραπάνω εικόνα, ο Πίνακας A έχει διάσταση n x m, ενώ ο B έχει διάσταση m x p, όπου αναγκαστικά ο αριθμός των στηλών του A είναι ίδιος με τον αριθμό των γραμμών του B. Το γινόμενο των 2 πινάκων συμβολίζεται με AB και έχει διάσταση n x p όπως φαίνεται στη παρακάτω εικόνα:

$$AB = \begin{pmatrix} (AB)_{11} & (AB)_{12} & \cdots & (AB)_{1p} \\ (AB)_{21} & (AB)_{22} & \cdots & (AB)_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ (AB)_{n1} & (AB)_{n2} & \cdots & (AB)_{np} \end{pmatrix}$$

Όπου AB ορίζεται ως:

$$(AB)_{ij} = \sum_{k=1}^{m} A_{ik} B_{kj}$$

Υποθέσεις:
Κάθε στοιχείο των πινάκων A και B, π.χ. A_{11}, B_{22}, είναι ένας μη προσημασμένος ακέραιος (unsigned integer) 8bit.
Κάθε στοιχείο του πίνακα AB, π.χ. AB_{11}, AB_{22}, είναι ένας μη προσημασμένος ακέραιος (unsigned integer) (16+lm)-bit (για λόγους απλότητας δηλώστε το ως 32bit unsigned integer).

- $m = 2^{lm}$, όπου lm είναι ένας ακέραιος τέτοιος ώστε $1 \le lm \le 8$, π.χ. $m=2^3 = 8$
- $n = 2^{ln}$, όπου ln είναι ένας ακέραιος τέτοιος ώστε $1 \le ln \le 8$, π.χ. $n=2^4 = 16$
- $p = 2^{lp}$, όπου lp είναι ένας ακέραιος τέτοιος ώστε $1 \le lp \le 8$, π.χ. $p=2^5 = 32$

Ερώτημα 1 (30%)

Γράψτε C κώδικα για το MATRIX_MUL στο Vivado HLS σύμφωνα με τους παραπάνω ορισμούς και αποθηκεύστε το στο *source* πεδίο του Vivado HLS. Ορίστε τα lm, ln, lp ως σταθερές χρησιμοποιώντας #define. Ορίστε τις τιμές m, n, p ως σταθερές οι οποίες εξαρτώνται από τις lm, ln, lp χρησιμοποιώντας #define. Υπολογίστε τη δύναμη του 2 χρησιμοποιώντας shift left. **Μην** αρχικοποιείτε αυτές τις τιμές χρησιμοποιώντας scanf().

Επίσης γράψτε ένα testbench σε C/C++ και προσθέστε το στο Vivado HLS, το οποίο:

α) αρχικοποιεί τους πινάκες A και B με ψευδό-τυχαίες τιμές στο εύρος 0-255
β) υπολογίζει το γινόμενο των 2 πινάκων χρησιμοποιώντας τόσο μια S/W μαζί με τη H/W λύση ούτος ώστε να διασφαλίζει τη σωστή εκτέλεση της H/W λειτουργικότητας.
γ) Εκτυπώνει ευανάγνωστα τα αποτελέσματα μαζί με ακόλουθο μήνυμα «Test Passed» σε περίπτωση επιτυχίας.

Ερώτημα 2 (5%)

Κάντε σύνθεση τη παραπάνω σας σχεδίαση (C Synthesis) με default settings και συμπληρώστε τα ακόλουθα:

Estimated clock period:	4.997 ns
Worst case latency:	42.273 ms
Number of DSP48E used:	3
Number of BRAMs used:	0
Number of FFs used:	115
Number of LUTs used:	232

Ερώτημα 3 (5%)

Τρέξτε C/RTL cosimulation και βεβαιωθείτε ότι η σχεδίαση σας περνάει το test επιτυχώς και συμπληρώστε τα ακόλουθα για τις τιμές lm=ln=lp=8:

Total Execution Time:	42.273 ms
Min latency:	42.273 ms
Avg. latency:	42.273 ms
Max latency:	42.273 ms

Ερώτημα 4 (40%)

Εφαρμόστε Vivado HLS directives (π.χ. ARRAY_PARTITION, PIPELINE, UNROLL) έτσι ώστε να βελτιώσετε όσο πιο πολύ μπορείτε το execution time[5]. Πληροφορίες για τα directives μπορείτε να βρείτε εδώ[6]. Πειραματιστείτε με διάφορες διαστάσεις των πινάκων. Τι

[5] Αν χρειαστεί εφαρμόστε το TRIPCOUNT pragma για να ορίσετε τα όριο των επαναλήψεων.
[6] https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_1/ug1270-vivado-hls-opt-methodology-guide.pdf

παρατηρείτε στη περίπτωση όπου lm=7 και μεταβάλετε τις άλλες 2 διαστάσεις? Μόλις φτάσετε στη βέλτιστη λύση παρακαλώ πολύ συμπληρώστε τα ακόλουθα (για lm=ln=lp=8) καθώς και ποια directives χρησιμοποιήσατε για να φτάσετε σε αυτή και γιατί;

Estimated clock period:	8.410 ns
Number of DSP48E used:	384
Number of BRAMs used:	0
Number of FFs used:	648
Number of LUTs used:	6810
Total Execution Time:	0.164 ms
Min latency:	0.164 ms
Avg. latency:	0.164 ms
Max latency:	0.164 ms

Τέλος υπολογίστε την επιτάχυνση (speed-up) της βέλτιστης hardware υλοποίηση σας συγκριτικά με:
α) την αρχική σχεδίαση σας σε hardware
β) την υλοποίηση σας σε software

Ηλιάδης-Αποστολίδης Δημοσθένης 8811
Φραντζέσκος Παναγιώτης 8939