ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ

ΗΡΥ 519
ΑΝΑΔΙΑΤΑΣΣΟΜΕΝΑ ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2019-20

Υλοποίηση Αλγόριθμου σε Hardware με τη χρήση τεχνικών high-level synthesis και του περιβάλλοντος SDSoC

Χειμερινό εξάμηνο 2019-20
Ν. Αλαχιώτης

Περιγραφή

Στην άσκηση αυτή καλείστε να υλοποιήσετε έναν αλγόριθμο σε αναδιατασσόμενη λογική με τη χρήση τεχνικών high-level synthesis και του περιβάλλοντος SDSoC.

Προετοιμασία (**Παραδοτέο 1**)

Μελετήστε τον C κώδικα που σας δόθηκε και κατανοήστε τη λειτουργία του. Σκοπός της άσκησης είναι η δημιουργία αρχιτεκτονικής για την επιτάχυνση της συνάρτησης **myFunc**, καθώς και η βέλτιστη δυνατή χρήση των πόρων του ZedBoard development board με σκοπό την επίτευξη της καλύτερης δυνατής απόδοσης για τη συγκεκριμένη συνάρτηση όταν καλείται από το processing system του συγκεκριμένου Zynq-7000 AP-SoC (all-programmable SoC).

Αρχικά επεκτείνετε κατάλληλα τον C κώδικα ώστε να διευκολύνει τον έλεγχο ορθότητας κατά τη σχεδίαση της αρχιτεκτονικής του επιταχυντή, καθώς και την αξιολόγηση απόδοσης κατά την εκτέλεση στο ZedBoard, με σημείο αναφοράς για ορθότητα και απόδοση την εκτέλεση της συνάρτησης myFunc στον επεξεργαστή του Zynq-7000. Η συνάρτηση μέσω της οποίας ο επεξεργαστής του Zynq-7000 θα χρησιμοποιεί την αναδιατασσόμενη λογική θα είναι η *myFuncAccel()*.

Σχεδίαση αρχιτεκτονικής επιταχυντή (**Παραδοτέα 2Α, 2Β**)

Αφού εκτελέσετε τον τροποποιημένο κώδικα του Παραδοτέου 1 μέσω του εργαλείου Xilinx Vivado HLS και παρατηρήσετε ορθή εκτέλεση μέσω C simulation, προχωρήστε στον σχεδιασμό της αρχιτεκτονικής του επιταχυντή.

Μπορεί να χρειαστεί να τροποποιήσετε συγκεκριμένα κομμάτια του κώδικα ώστε να αποτυπωθούν πιο αποτελεσματικά σε αναδιατασσόμενη λογική. Δικαιολογήστε αναλυτικά στην αναφορά σας κάθε τροποποίηση του C κώδικα που θα κάνετε, καθώς και την λειτουργία και τον λόγο χρήσης της κάθε hls ντιρεκτίβας που θα επιλέξετε να χρησιμοποιήσετε.

Θεωρήστε ότι σε ένα πραγματικό σενάριο κλήσης της συνάρτησης myFunc, η παράμετρος εισόδου size μπορεί να λάβει οποιαδήποτε θετική τιμή (ο μόνος περιορισμός είναι η διαθέσιμη

μνήμη), ενώ η παράμετρος εισόδου dim λαμβάνει αυστηρά και μόνο τις τιμές 4 ή 16. Βάση αυτού, αν κρίνετε ότι εξυπηρετεί τον σχεδιασμό της αρχιτεκτονικής και εξηγήσετε πλήρως τον σχετικό λόγο στην αναφορά σας, μπορείτε να δημιουργήσετε δύο επιταχυντές (ίδια αρχιτεκτονική), έναν για dim = 4 και έναν για dim = 16. Μπορείτε δηλαδή να θεωρήσετε το dim ως σταθερά για την υλοποίηση του επιταχυντή αν και μόνο αν δικαιολογήσετε αναλυτικά την απόφαση σας στην αναφορά.

Οι ελάχιστες απαιτήσεις για τον σχεδιασμό του επιταχυντή με HLS ντιρεκτίβες είναι να είναι pipelined με το ελάχιστο δυνατό initiation interval, να εκτελεί παράλληλα όσες πράξεις μπορούν να γίνουν παράλληλα και να λειτουργεί σε συχνότητα τουλάχιστον 100 MHz.

Δείξτε την ορθότητα της τελικής σχεδίασης σας μέσω C/RTL co-simulation και αναφέρετε στην αναφορά σας τα αποτελέσματα του synthesis σχετικά με latency, initiation interval, και πόρους αναδιατασσόμενης λογικής.

Κλήση της *myFuncAccel* σε περιβάλλον Linux μέσω Xilinx SDSoC (Παραδοτέο 3)

Αφού έχετε σχεδιάσει τον επιταχυντή σας (αν δημιουργήσατε δύο επιταχυντές λόγω του dim προχωρήστε με έναν από τους δύο μόνο), χρησιμοποιήστε το Xilinx SDSoC για να δημιουργήσετε ένα πραγματικό περιβάλλον εκτέλεσης και αξιολόγησης της σχεδίασης σας. Σε αυτό το σημείο καλείστε να τροποποιήσετε κατάλληλα τον κώδικα σας από το προηγούμενο βήμα με sds ντιρεκτίβες και τις κατάλληλες συναρτήσεις της βιβλιοθήκης sds_lib.h, ώστε το processing system να δεσμεύει χώρο μνήμης για τα δεδομένα εισόδου και εξόδου του επιταχυντή σε shared address space και να κατευθύνετε το εργαλείο SDSoC να τοποθετήσει τους κατάλληλους data movers στο αναδιατασσόμενο μέρος του Zynq-7000.

Δημιουργήστε και καλέστε δύο αντίγραφα του επιταχυντή για περισσότερο παραλληλισμό αν είναι εφικτό. Αν όχι, δείξτε και εξηγήστε τον λόγο.

Απαιτούμενο για το Παραδοτέο 3 είναι να γίνεται επιτυχώς build ο κώδικας σας στο SDSoC για συχνότητας λειτουργίας 100MHz.

Τελική Αναφορά (Παραδοτέο 4)

Στην τελική αναφορά θα πρέπει να συμπεριλάβετε με διακριτούς τίτλους τουλάχιστον τα παρακάτω, καθώς και ότι άλλο κρίνετε απαραίτητο για την λεπτομερή περιγραφή της λογικής που ακολουθήσατε για τον σχεδιασμό του επιταχυντή και την υλοποίηση της συγκεκριμένης εργασίας:

1. Περιγραφή της λειτουργίας της συνάρτησης myFunc
2. Περιγραφή της αρχιτεκτονικής του επιταχυντή
3. Δικαιολόγηση σχεδιαστικών αποφάσεων και τροποποιήσεων του κώδικα
4. Σύντομη περιγραφή της αρχιτεκτονικής του Zynq-7000 AP-SoC
5. Αναφορά των HLS ντιρεκτίβων που χρησιμοποιήθηκαν, περιγραφή της λειτουργίας τους, και δικαιολόγηση της χρήσης τους
6. Αναφορά των SDS ντιρεκτίβων και των συναρτήσεων του sds_lib API που χρησιμοποιήθηκαν, περιγραφή της λειτουργίας τους, και δικαιολόγηση της χρήσης τους
7. Αξιολόγηση απόδοσης
8. Συμπεράσματα

Παραδοτέα

Παραδοτέο 1. Ο τροποποιημένος κώδικας σας (πριν το σχεδιασμό του επιταχυντή) με τις κατάλληλες προσθήκες για έλεγχο ορθότητας και απόδοσης.

Παραδοτέο 2A. Ο τροποποιημένος κώδικας του Παραδοτέου 1 με σκοπό την καλύτερη απεικόνιση σε αναδιατασσόμενη λογική (αν το κρίνετε απαραίτητο).

Παραδοτέο 2B. Ο τελικός κώδικας του επιταχυντή και του test environment για C/RTL cosimulation.

Παραδοτέο 3. Ο τελικός κώδικας του συνολικού συστήματος για χρήση μέσω του SDSoC (επιτυχές build) και εκτέλεση στο ZedBoard.

Παραδοτέο 4. Τελική αναφορά σε PDF.

Υποβολή και Βαθμολόγηση

Παραδοτέο 1:	3 Νοεμβρίου	(5%)
Παραδοτέο 2A :	3 Νοεμβρίου	(5%)
Παραδοτέο 2B:	24 Νοεμβρίου	(35%)
Παραδοτέο 3:	19 Δεκεμβρίου	(20%)
Παραδοτέο 4:	19 Δεκεμβρίου	(35%)

Πληροφορίες για HLS pragmas:

- https://www.xilinx.com/html_docs/xilinx2017_4/sdaccel_doc/okr1504034364623-2.html

Πληροφορίες για το SDSoC pragmas:

- https://www.xilinx.com/html_docs/xilinx2017_4/sdaccel_doc/nmc1504034362475.html

Πληροφορίες για το SDSoC environment API (sds_lib):

- https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_2/sdsoc_doc/topics/api/reference_sdsoc_api.html