Ναπολεων Παπουτσάκης ΑΕΜ:9170
Γιωργος – Εφραίμ Παππάς ΑΕΜ: 9124

Εργαστηριο 2ο

2. Εισαγωγή στο Vitis

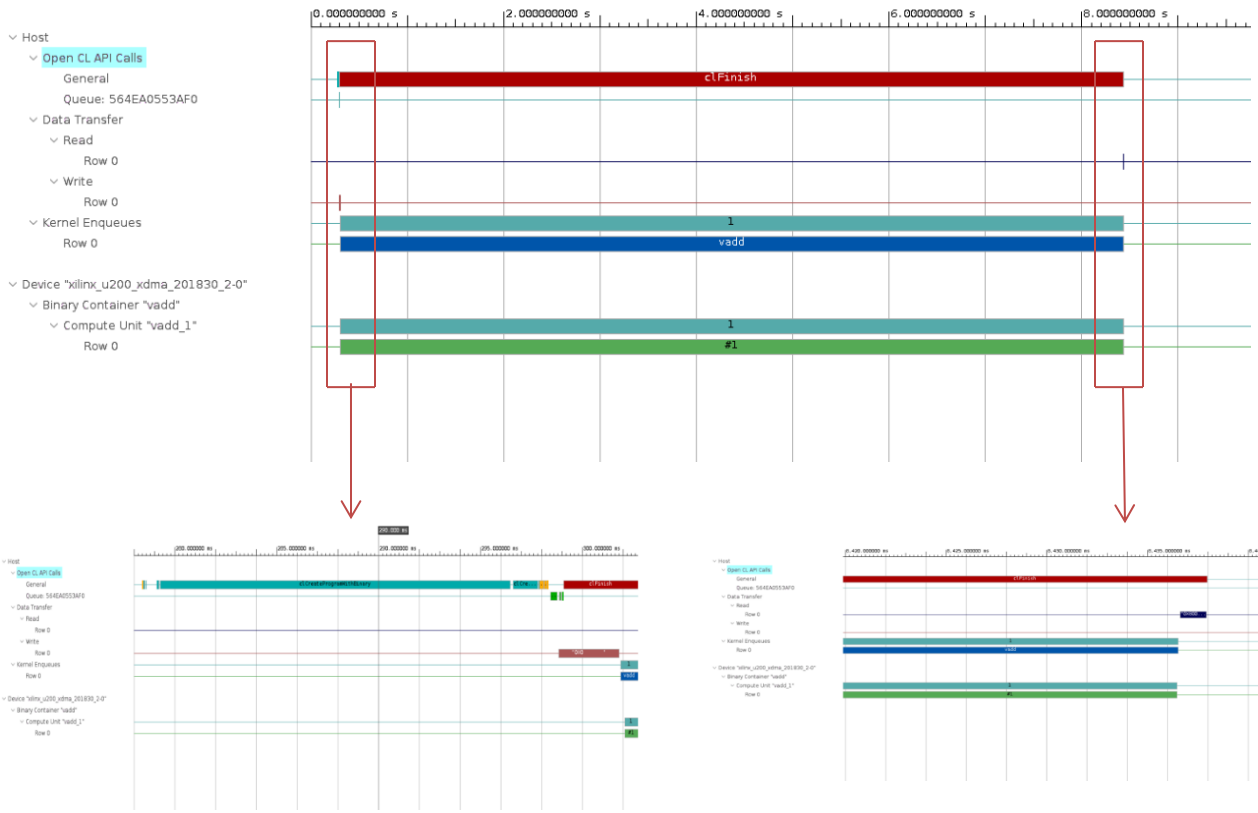
Αρχικά, προκειμένου να εξοικειωθούμε με το περιβάλλον του Vitis ακολουθήσαμε τα βήματα του «introduction to Vitis» εκτελώντας Software Emulation και Hardware Emulation για το παράδειγμα της πρόσθεσης 2 διανυσμάτων (vadd).Το παράδειγμα του vadd αποτελείται απο αρχεία. Στο πρώτο αρχείο, το host.cpp, που θα τρέξει στο host επεξεργαστή ειναι γραμμένος σε openCL , δημιουργούνται τα διανύσματα που θα προστεθούν και αρχικοποιώντας τα με τυχαίες τιμές. Έπειτα,εκτελεί το άθροισμα των διανυσμάτων στο software ,αποθηκεύει το αποτέλεσμα σε ένα άλλο διάνυσμα, ενω περνάει τα διανύσματα που θα προστεθούν στην καθολική μνήμη του συστήματος προκειμένου ο kernel που θα εκτελεστεί στην FPGA να μπορεί να τα διαβάσει και να εκτελέσει την πρόσθεση σε Hardware. Τέλος, γίνεται μια σύγκριση των αποτελεσμάτων που προέκυψαν απο το software και το hardware και περίπτωση που τα αποτελέσματα ταυτίζονται εκτυπώνεται το μήνυμα “TEST PASSED” . Στο δεύτερο αρχείο, το vadd.cpp, αφού φτιαχνονται τα interfaces (axilite,axi), αντιγράφονται τα δεδομένα απο την DRAM στη BRAM για τα 2 διανύσματα, πραγματοποιεί την πρόσθεση και αποθηκεύει τα αποτελέσματα πάλι πίσω στην DRAM.

3. Υλοποίηση του accelerator που δημιουργήσατε στο 1ο εργαστήριο στο Vitis

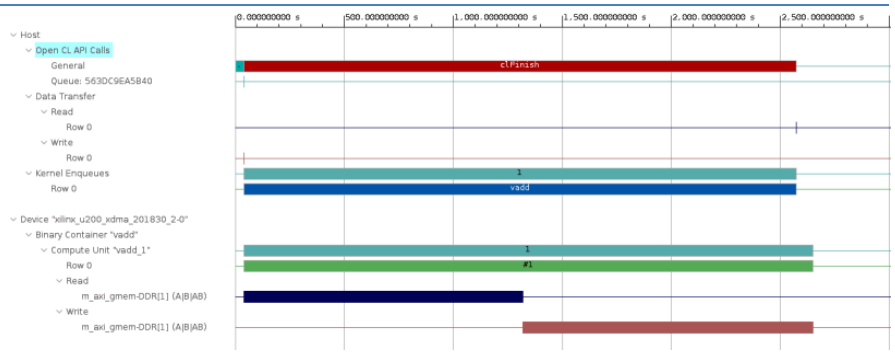
Αφού εκτελέστηκαν σωστά τα βήματα του παραδείγματος vadd, τροποποιήσαμε τον κώδικα του host.cpp έτσι ώστε να υλοποιεί πλέον πολλαπλασιασμό σε software (και όχι πρόσθεση) και να καλεί τον kernel που υλοποιήσαμε στο πρώτο εργαστήριο, για την εκτέλεση του πολλαπλασιασμού σε hardware.
Η γενικότερη λειτουργικότητα του host παραμένει ίδια με το παράδειγμα, καθώς πάλι αρχικοποιούνται δυο vectors, αυτη τη φορά με μέγεθος όσο και ο αριθμός των στοιχείων των πινάκων A, B αντίστοιχα, δηλαδή n*m και m*p. Πλέον, τα δύο αυτά vectors, με κατάλληλη προσπέλαση αντικαθιστούν του πινακες A, B(πχ vectorA[i*m+j] = A[i][j]).Τα διανύσματα αυτά είναι που θα πολλαπλασιαστούν στο software με τον τρόπο που δείξαμε, με το αποτέλεσμα να συγκρίνεται με αυτό που θα παραχθεί απο το hardware, αφου ο host περάσει τα αποτελέσματα στην καθολική μνήμη και καλέσει τον kernel μας. Ομοια με το παράδειγμα, σε περίπτωση που τα αποτελέσματα του software και του hardware ταυτίζονται, εκτυπώνεται το μήνυμα “TEST PASSED”
Στο αρχείο του kernel, αφού οριστούν τα απαραίτητα interfaces όμοια με το παράδειγμα vadd, αντιγράφουμε τους δύο πίνακες στη bram(όπως και στο πρώτο εργαστήριο) και υλοποιούμε τον πολλαπλασιασμό μεταξύ τους, εφαρμόζοντας διάφορα pragmas για την επιτάχυνση του κώδικα(array partition, pipeline, loop unrolling) αποθηκεύοντας τελικά το αποτέλεσμα πίσω στην καθολική μνήμη.

Παρακάτω παρατίθενται τα αποτελέσματα απο το **Software Simulation**:

Ναπολεων Παπουτσάκης AEM:9170
Γιωργος – Εφραίμ Παππάς AEM: 9124



Παρατίθενται τα αποτελέσματα του **Hardware Simulation:**



Στη δεξιά εικόνα, φαίνεται στο κοκκινο πλαίσιο ο χρόνος εκτέλεσης του kernel μας στην Hardware emulation ο οποιος ειναι 2.129ms. Αυτή η πληροφορία είναι χρήσιμη καθώς πλέον γνωρίζουμε οτι ο χρόνος που θα χρειστεί ο kernel να τρέξει στον alveo board θα είναι σίγουρα λιγότερος απο τα 2.129ms, μιας και ο χρόνος του emulator ειναι αυτος που θα χρειαστεί ο kernel στη χειροτερη περίπτωση.

Top Data Transfer: Kernels to Global Memory

Device	Compute Unit	Number Of Transfers	Average Bytes per Transfer	Transfer Efficiency (%)	Total Data Transfer (MB)	Total Write (MB)	Total Read (MB)	Total Transfer Rate (MB/s)
xilinx_u200_xdma_201830_2-0	vadd_1	196608	4.000	0.098	0.786	0.262	0.524	370.069

Top Kernel Execution

Kernel Instance Address	Kernel	Context ID	Command Queue ID	Device	Start Time (ms)	Duration (ms)	Global Work Size	Local Work Size
0x563dc9f032d0	vadd	0	0	xilinx_u200_xdma_201830_2-0	0.022	2.129	1:1:1	1:1:1

Top Memory Writes: Host to Global Memory

Buffer Address	Context ID	Command Queue ID	Start Time (ms)	Duration (ms)	Buffer Size (KB)	Writing Rate (MB/s)
0x400000000	0	0	38607.800	N/A	524.288	N/A

Top Memory Reads: Host to Global Memory

Buffer Address	Context ID	Command Queue ID	Start Time (ms)	Duration (ms)	Buffer Size (KB)	Reading Rate (MB/s)
0x400080000	0	0	2573850.000	N/A	262.144	N/A

Ναπολεων Παπουτσάκης ΑΕΜ:9170
Γιωργος – Εφραίμ Παππάς ΑΕΜ: 9124