



Technical
University
of Crete

Αναδιατασσόμενα Ψηφιακά Συστήματα

HPY591

Milestone 3

Σπυριδάκης Χρήστος
Βίττης Βασίλειος

2014030022
2015030164

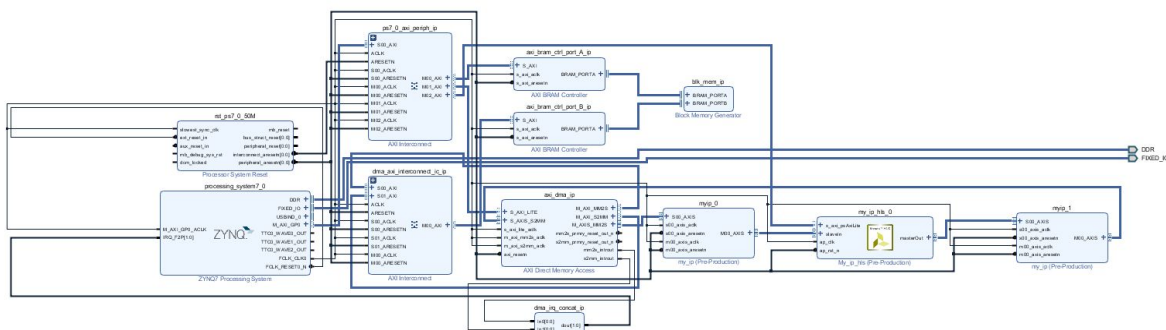
1.Εισαγωγή

Σκοπος του συγκεκριμενου Milestone είναι να χρησιμοποιήσουμε το εργαλείο Vivado προκειμένου να ενωποιήσουμε τα στοιχεία των Milestone 1 & 2 σε ένα, ώστε να λειτουργούν μαζί. Αξίζει να σημειωθεί ότι για την υλοποίηση του συγκεκριμένου Milestone, ακολουθήσαμε τις οδηγίες της εκφώνησης καθώς και τις οδηγίες που δόθηκαν κατά τις διαλέξεις.

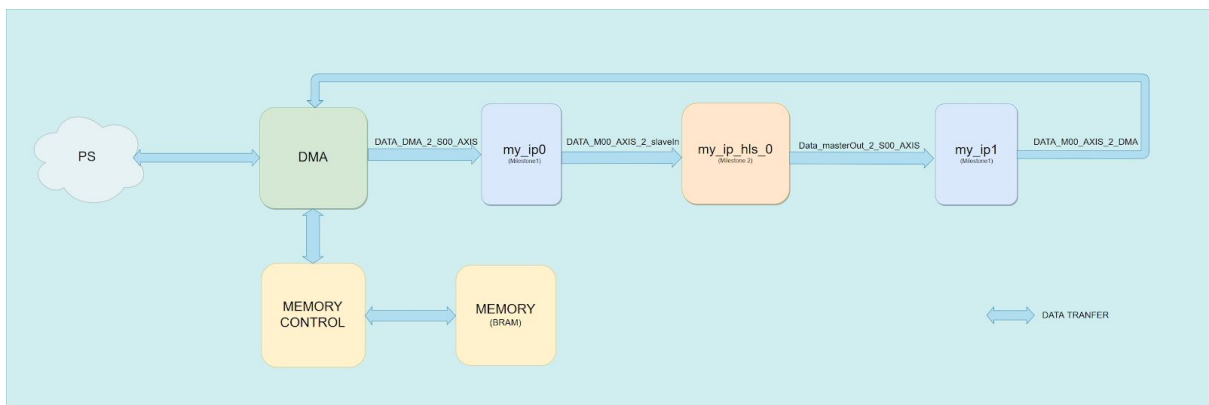
Σημείωση: Από την στιγμή που δεν αναφέρεται ότι χρειάζεται στα παραδοτέα να είναι και η αποστολή του ενοποιημένου project, αυτό δεν δίνεται. Παρόλα αυτά αν για κάποιο λόγο χρειάζεται να επαληθευθεί κάτι από την συνένωση τους ή να γίνει προβολή των εικόνων σε καλύτερη ανάλυση τότε μπορεί να βρεθεί ολόκληρο το project σε αυτό το url: <https://github.com/CSpyridakis/Reconfigurable-Computing>.

2.Υλοποίηση

Έπειτα από τις οδηγίες που ακολουθήσαμε, το Top-Module το οποίο δημιουργήσαμε εμφανίζεται στην εικόνα 2.1.



Εικόνα 2.1 - Top-Module



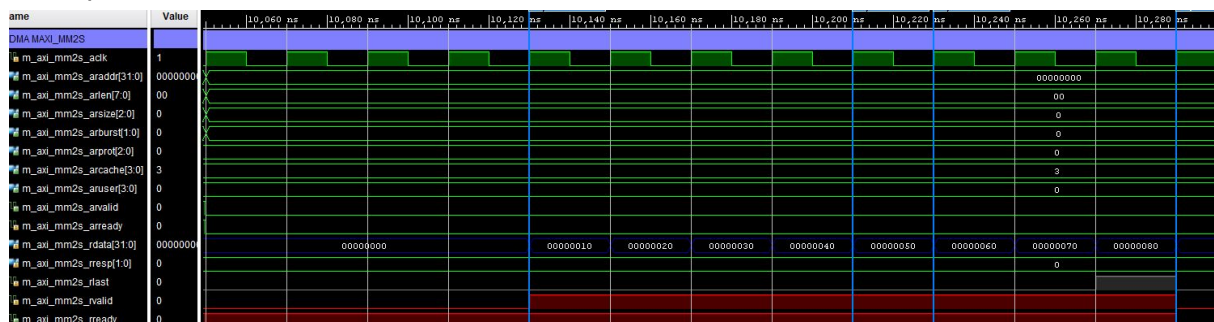
Εικόνα 2.2 - Dataflow

Ενώ στην εικόνα 2.2 βλέπουμε την ροή που έχουν τα δεδομένα από τον DMA engine προς τη λογική μας που δημιουργήσαμε κατά τα Milestone 1 και Milestone 2 όπως επίσης και την σύνδεση του DMA με το PS και την BRAM. Αρχικά, το PS μετά από το configuration του DMA, του δίνει εντολή για να ζητήσει δεδομένα από την BRAM. Έτσι, το DMA ζητάει δεδομένα από την BRAM σε συγκεκριμένες διευθύνσεις μνήμης. Αφού τα λάβει τότε ξεκινάει την διαδικασία αποστολής τους στα modules των προηγούμενων Milestones. Για να επιτευχθεί αυτή η διαδικασία πρώτα τα δεδομένα πρέπει να μεταφερθούν μέσω της FIFO που δημιουργήσαμε στο Milestone 1 (my_ip0). Η εφαρμογή της υλοποίησης, της εν λόγω FIFO είναι περιττή σε αυτό το σημείο αλλά δικαιολογείται καθώς έχουμε ως γνώμονα την ενοποίηση και την ανάδειξη της πλήρους λειτουργικότητας των προηγούμενων Milestones ως σύνολο. Άρα, αφού τα δεδομένα μας γραφτούν στην πρώτη FIFO και διαβαστούν από το module του Milestone 2 (my_ip_hls_0) τότε γίνεται η επεξεργασία τους. Όπως γίνεται εμφανές και παρακάτω στην είσοδο του my_ip_hls_0 έρχονται σε μορφή Stream οι αριθμοί από το 10 έως το 80 με βήμα 10 και περνάνε από το φιλτράρισμα όλοι οι αριθμοί εκτός από το 10, 30, 50 που αντιστοιχίζονται με τους κανόνες. Το αποτέλεσμα αυτού του φιλτραρίσματος μεταφέρεται στην δεύτερη FIFO (my_ip1) που και αυτή δεν είναι απαραίτητη και μέσω αυτής πίσω πάλι στο DMA engine. Τέλος, αφού το DMA engine έχει τα φιλτραρισμένα δεδομένα τα γράφει στις διευθύνσεις μνήμης της BRAM. Πρέπει να αναφερθεί ότι προκειμένου να είναι πιο “πλήρης” η επικοινωνία AXI Stream που γίνεται μεταξύ των IP και του DMA, διαμορφώθηκαν οι FIFOs του Milestone 1 ώστε να δίνουν το σήμα last στο τελευταίο πακέτο.

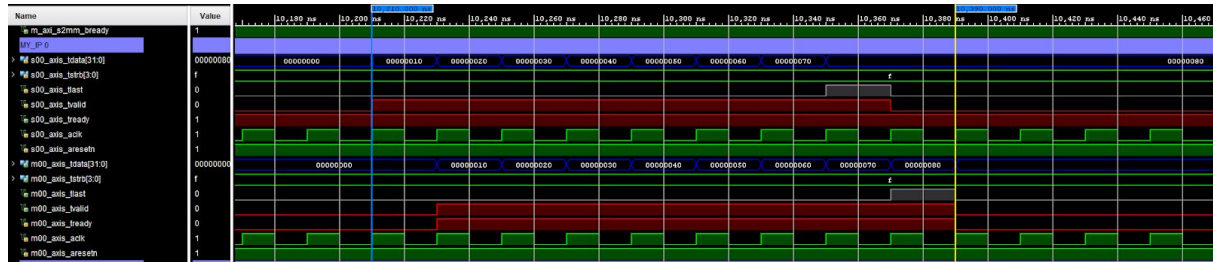
3. Testing

Σημείωση: Σε κάθε κυματομορφή υπάρχει το timeline ώστε να φαίνεται η διαδοχική πορεία των ενεργειών.

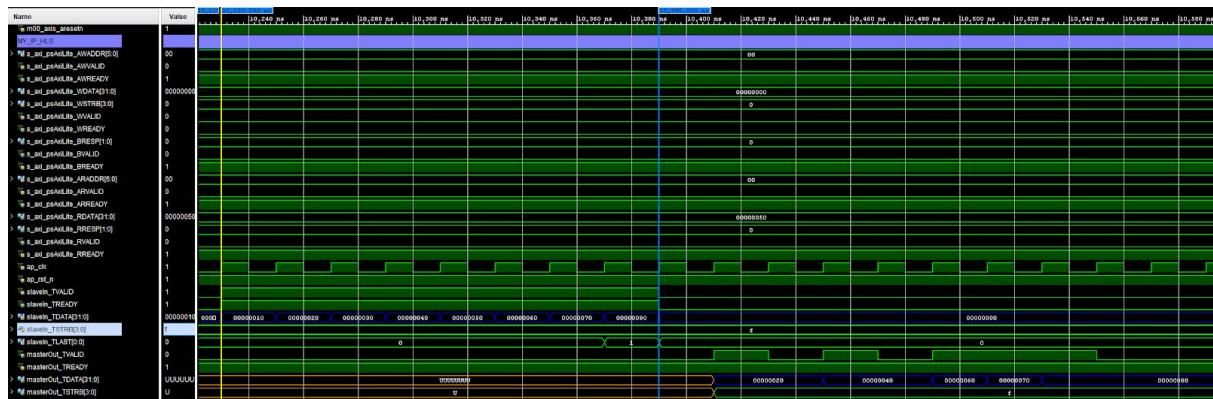
Στην παρακάτω κυματομορφή βλέπουμε την μεταφορά των δεδομένων από το DMA στην πρώτη FIFO.



Παρακάτω βλέπουμε την σωστή λειτουργία της πρώτης FIFO του πρώτου Milestone.



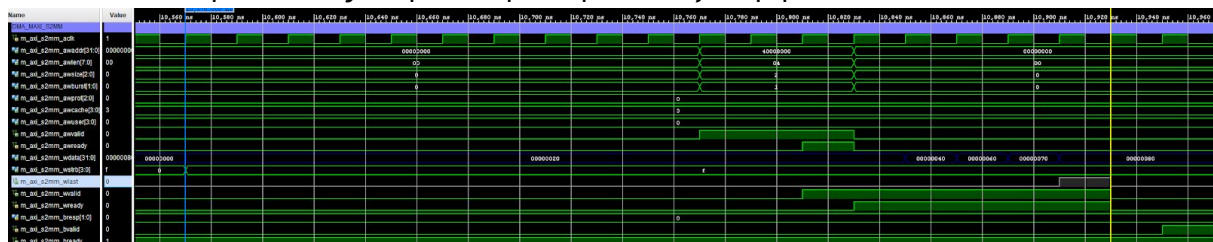
Στην κάτωθι κυματομορφή βλέπουμε ότι σωστά τα δεδομένα που μπαίνουν (10,...,80) φιλτράρονται και προκύπτει η έξοδος (20,40,60,70,80) στο my_ip_hls_0.



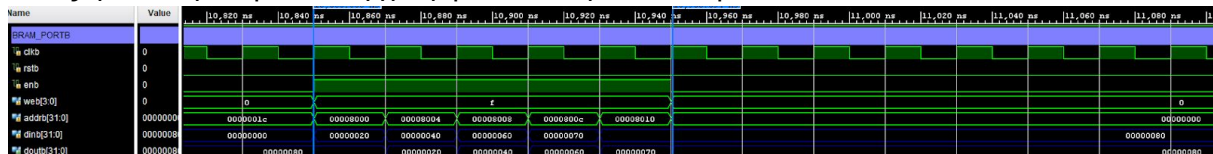
Αντίστοιχα με την πρώτη FIFO έτσι και σε αυτή βλέπουμε να λειτουργεί σωστά η μεταφορά των δεδομένων μέσω αυτής.



Έπειτα, τα δεδομένα όπως και βλέπουμε παρακάτω ξαναγυρνούν στον DMA



Τέλος, βλέπουμε την επανεγγραφή των δεδομένων στη BRAM.



4.Μελλοντική δουλειά

Αν θέλαμε να ασχοληθούμε σε μελλοντικό σημείο με το συγκεκριμένο project ένα από αυτά που θα θέλαμε να κάνουμε είναι να μπορεί το `my_ip_hls` να επεξεργάζεται ροές δεδομένων όπου τυχαίνει το τελευταίο πακέτο να πρέπει να φιλτραριστεί. Ο λόγος που είναι σημαντικό αυτό είναι ότι σε αυτό το σημείο αν το τελευταίο πακέτο αντιστοιχεί σε κάποιο κανόνα τότε η πληροφορία του `last` χάνεται.