

ΠΡΟΗΓΜΕΝΗ ΣΧΕΔΙΑΣΗ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2024-2025

ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ

*Sobel Edge Detector SoC Design on Zynq
7000 AP SoC*

Περίληψη της Εργασίας

Σε αυτή την εργασία θα σχεδιάσετε ένα σύστημα σε τσιπ (System-on-Chip) που θα επιταχύνει τον αλγόριθμο ανίχνευσης ακμών Sobel. Αρχικά θα υλοποιήσετε τον αλγόριθμο ανίχνευσης ακμών με την γλώσσα προγραμματισμού C στο περιβάλλον του PetaLinux και θα τον τρέξετε σε επεξεργαστή αρχιτεκτονικής ARM συλλέγοντας στατιστικά επίδοσης με κατάλληλους μετρητές στο πρόγραμμά σας.

Έπειτα θα σχεδιάσετε με χρήση της VHDL μια μονάδα ανίχνευσης ακμών που υλοποιεί τον Sobel αλγόριθμο (Sobel Edge Detection IP Core) όπου επιταχύνοντας τον παραπάνω αλγόριθμο στο υλικό (fabric) του FPGA. Το IP core θα υποστηρίξει εσωτερικούς μετρητές απόδοσης (performance counters) που θα μετρούν τους κύκλους ρολογιού που χρειάστηκαν για την ολοκλήρωση της επεξεργασίας μιας εικόνας καθώς επίσης και τον αριθμό των εικονοστοιχείων εισόδου και εξόδου.

Το Edge Detection IP Core θα αποτελεί μια αυτόνομη μονάδα που μπορεί να δοκιμαστεί με προσομοίωση (Simulation-based verification) αλλά θα διαθέτει παράλληλα και κατάλληλες διεπαφές (AXI4-Lite, AXI-4 Stream Interfaces) επιτρέποντάς του την διασύνδεση με το επεξεργαστικό σύστημα (Processing System) του Zynq 7000 AP SoC της Xilinx ως υπομονάδα ενός γενικού Sobel Edge Detector SoC. Το όλο σύστημα θα ελέγχεται από πρόγραμμα χρήστη υλοποιημένο σε C και θα τρέχει σε περιβάλλον ενσωματωμένου λειτουργικού συστήματος Linux- το PetaLinux.

Για την επιτυχή ολοκλήρωση της εργασίας απαιτείται

1. Η υλοποίηση και εκτέλεση του αλγορίθμου Sobel Edge Detection σε επεξεργαστή ARM της πλατφόρμας Zedboard Evaluation Board. Το πρόγραμμα θα είναι γραμμένο σε C (ή C++ για όποιον επιθυμεί).
2. Η λήψη στατιστικών επίδοσης του παραπάνω προγράμματος.
3. Η σχεδίαση με VHDL και επαλήθευση με χρήση προσομοίωσης του Sobel Edge Detector IP Core.
4. Η ενσωμάτωση του Sobel Edge Detector IP Core σε ένα SoC και μέτρηση της επιτάχυνσης του αλγορίθμου σε σχέση με την απλή υλοποίηση του ακολουθιακού πρόγραμμα που φτιάξατε προηγουμένως.

Η επίδειξη για γίνει με τη βοήθεια του επιβλέποντος συνεργάτη στο εργαστήριο.

Οδηγίες για την δημιουργία του PetaLinux OS και της εκτέλεσής του στο Zynq 7000 θα σας δοθούν στο e-class. Επιπλέον το e-class θα βρείτε διάφορες εικόνες δοκιμής και όλα τα απαραίτητα αρχεία που θα χρειαστείτε για την δημιουργία του προγράμματος ελεγκτή του Sobel Edge Detector SoC.

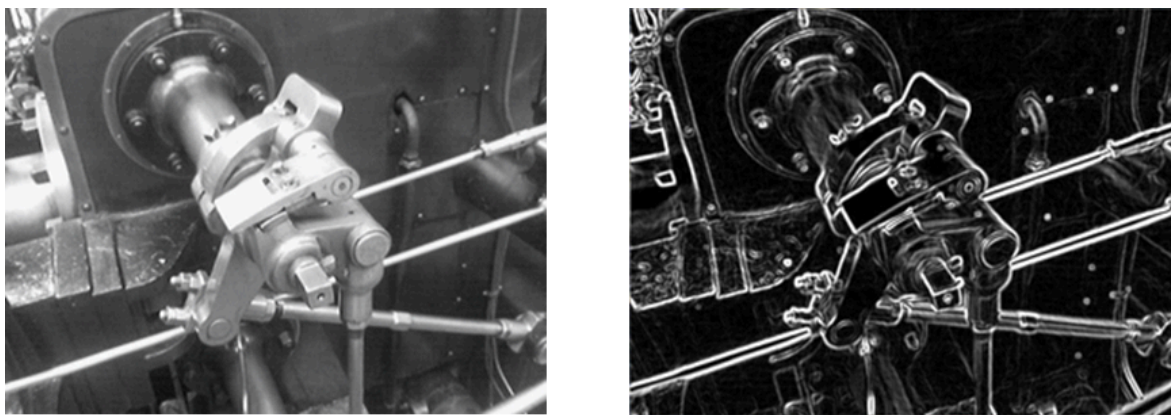
Θα πρέπει να παραδώσετε μια τεχνική έκθεση που θα παρουσιάζεται αναλυτικά τα αποτελέσματα και τα ευρήματά σας. Θα πρέπει να φαίνεται ξεκάθαρα η μεθοδολογία υλοποίησης και

σχεδίασης, τα αποτελέσματα σύγκρισης, οι πόροι υλικού του χρειάστηκαν για το σύστημα, τυχόν προβληματισμοί και σχόλια. Ενθαρρύνεστε να “παίξετε” με τις διάφορες παραμέτρους που επηρεάζουν την επίδοση του Sobel Edge Detector SoC (π.χ. μέγεθος block δεδομένων προς το DMA IP Core, συχνότητες λειτουργίας, μέγεθος FIFO κλπ) και να σχολιάσετε τα αποτελέσματα.

Μαζί με την αναφορά θα επισυνάψετε σε ένα zip όλα τα VHDL αρχεία που φτιάξατε συμπεριλαμβανομένου και των testbenches, τον κώδικα C (ή C++) του προγράμματος αναφοράς σας, το TCL script της τελικής αρχιτεκτονικής σας καθώς και τα boot components του PetaLinux (BOOT.BIN, boot.scr, system.dtb, image.ub, rootfs.tar.gz).

Βασικές Έννοιες Ανίχνευσης Ακμών

Η ανίχνευση ακμών αποτελεί μια ιδιαίτερως σημαντική εφαρμογή επεξεργασίας εικόνας με πληθώρα εφαρμογών όπως η ασφαλείας (κάμερες παρακολούθησης), η όραση υπολογιστών (computer vision), η τεχνητή νοημοσύνη κλπ. Με την ανίχνευση ακμών αναφερόμαστε στον προσδιορισμό των περιοχών μιας εικόνας στις οποίες γίνεται εμφανής μια απότομη αλλαγή στην ένταση. Καθώς αυτές οι περιοχές εμφανίζονται επί το πλείστον στα όρια των αντικειμένων, συνιστούν ακμές και άρα προσδιορίζουν ευδιάκριτα αντικείμενα τα οποία παρουσιάζονται σε μιας σκηνή (εικόνα). Ένα παράδειγμα ανίχνευσης ακμών παρουσιάζεται στην Εικόνα 1.



Εικόνα 1. Ανίχνευση ακμών σε μια εικόνα

Για την εργασία, θα υποθέσουμε μονοχρωματικές εικόνες ανάλυσης 512x512 pixels των 8 bit. Δηλαδή οι τιμές των pixel ερμηνεύονται ως απροσημοί (unsigned) ακέραιοι με εύρος από 0 (μαύρο) έως 255 (λευκό).

Για την ανίχνευση ακμών θα γίνει χρήση ενός σχετικά απλού αλγορίθμου, που καλείται ανιχνευτής ακμών Sobel. Ο αλγόριθμος Sobel υπολογίζει τις παραγώγους του σήματος της έντασης σε κάθε μια από τις κατευθύνσεις x (οριζόντια) και y (κατακόρυφα) και αναζητά για αντίστοιχα μέγιστα και ελάχιστα και προς τις δύο αυτές κατευθύνσεις επάνω σε μια εικόνα. Αυτές είναι οι περιοχές όπου η ένταση αλλάζει ραγδαία και άρα σύμφωνα με τον ορισμό που δόθηκε παραπάνω συνιστούν ακμές αντικειμένων της σκηνής (εικόνας). Ο αλγόριθμος Sobel προσεγγίζει την παράγωγο σε κάθε κατεύθυνση για κάθε pixel με μια διαδικασία που καλείται *συνέλιξη (convolution)*. Η συνέλιξη αφορά την πρόσθεση του κεντρικού pixel μιας ομάδας 3x3 και των οκτώ πιο κοντινών γειτονικών του, το καθένα πολλαπλασιασμένο με ένα συντελεστή. Οι συντελεστές συχνά αναπαρίστανται με μια μάσκα συνέλιξης 3x3. Οι μάσκες συνέλιξης Sobel, G_x και G_y , για τις παραγώγους στις κατευθύνσεις x και y , αντίστοιχα, παρουσιάζονται στην Εικόνα 2. Μπορείτε να βρείτε και να δοκιμάσετε μάσκες με διαφορετικά βάρη, αρκεί να το αναφέρετε ρητά στην αναφορά σας!

G_x	-1	0	+1
	-2	0	+2
	-1	0	+1

G_y	+1	+2	+1
	0	0	0
	-1	-2	-1

Εικόνα 2. Μάσκες συνέλιξης Sobel

Θεωρούμε ότι η παράγωγος υπολογίζεται «κεντράροντας» κάθε μια από τις μάσκες συνέλιξης πάνω σε διαδοχικά pixel στην αρχική εικόνα. Ο συντελεστή σε κάθε μάσκα πολλαπλασιάζεται με την τιμή της έντασης του υποκείμενου pixel και τα επιμέρους εννέα γινόμενα αθροίζονται μαζί σχηματίζοντας τελικά δύο μερικές παραγώγους για την αρχική εικόνα οι οποίες δίνονται ως D_x και D_y και οι οποίες συνιστούν εικόνες παραγώγων ως προς την κατεύθυνση x και y αντίστοιχα. Το μέτρο του pixel της εικόνας της παραγώγου υπολογίζεται ως εξής:

$$|D| = \sqrt{D_x^2 + D_y^2}$$

Καθώς μας ενδιαφέρει απλά η εύρεση των μέγιστων και των ελάχιστων στο μέτρο, μια ικανοποιητική προσέγγιση είναι η ακόλουθη:

$$|D| = |D_x| + |D_y|$$

Αυτή η προσέγγιση ισχύει επειδή οι συναρτήσεις της τετραγωνικής ρίζας και του τετραγώνου είναι και οι δύο μονοτονικές (δηλαδή, αυξάνονται με την αύξηση του τελεστέου και μειώνονται με τη μείωση του τελεστέου). Συνεπώς, τα μέγιστα και τα ελάχιστα στο πραγματικό μέτρο και στο κατά προσέγγιση μέτρο εμφανίζονται στις ίδιες περιοχές στην εικόνα. Η προσέγγιση αυτή επιτρέπει μια πιο απλή υλοποίηση στο υλικό από τον υπολογισμό των συναρτήσεων του τετραγώνου και της τετραγωνικής ρίζας.

Αλγόριθμος 1 - Ο αλγόριθμος ανίχνευσης ακμών Sobel.

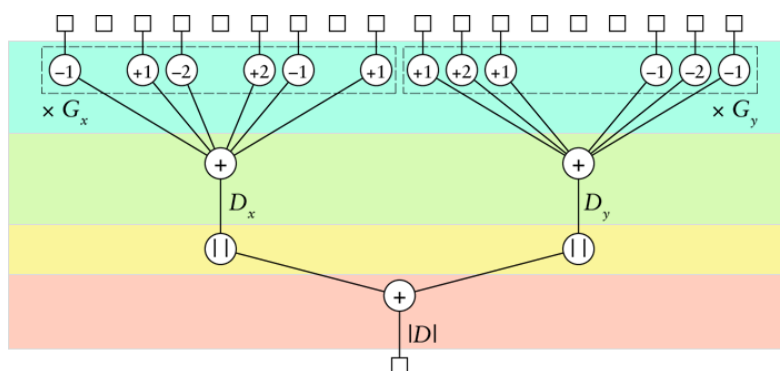
```
for row from 1 to 478 loop
  for col from 1 to 638 loop
    sumx := 0;
    sumy := 0;
    for i from -1 to +1 loop
      for j from -1 to +1 loop
        sumx := sumx + O(row+i, col+j) * Gx(i, j);
        sumy := sumy + O(row+i, col+j) * Gy(i, j);
      end loop
    end loop
    D(row, col) := abs(sumx) + abs(sumy);
  end loop
end loop
```

Θα πρέπει να σημειώσουμε ότι τα pixel στις άκρες της εικόνας δεν διαθέτουν ένα πλήρες σύνολο από γειτονικά pixel, και έτσι χρειάζεται να τα διαχειριστούμε ξεχωριστά. Η απλούστερη προσέγγιση είναι να θέσουμε την τιμή του $|D|$ για τα pixel στις άκρες της εικόνας της παραγώγου στο 0. Μια καλύτερη τεχνική είναι το “γέμισμα” (Padding) της εικόνας, κατά την οποία αυξάνεται το μέγεθος της κατά 2 εικονοστοιχεία οριζοντίως και καθέτως συμπληρώνοντας είτε απλά με μηδενικό είτε αντιγράφοντας την τιμή του αμέσως πλησιέστερου εικονοστοιχείου. Η περιγραφή του αλγορίθμου ανίχνευσης ακμών Sobel χρησιμοποιώντας ψευδοκώδικα παρουσιάζεται στον **Αλγόριθμο 1**. Στον ψευδοκώδικα έχουμε ορίσει $O(row, col)$ τα pixel στην αρχική εικόνα και $D(row, col)$ τα pixel στην εικόνα της παραγώγου. Επίσης, τα $Gx(i, j)$ και $Gy(i, j)$ αντιπροσωπεύουν τις μάσκες συνέλιξης, όπου τα i και j έχουν εύρος από -1 έως $+1$.

Προτεινόμενη Αρχιτεκτονική της Μονάδας Ανίχνευσης Ακμών Sobel

Οι υπολογισμοί που απαιτούνται για όλα τα pixel της παραγώγου είναι μεταξύ τους ανεξάρτητοι, αφού απαιτούν μόνο τιμές από τα pixel της αρχικής εικόνας ευνοώντας έτσι την παράλληλη εκτέλεση. Για τον υπολογισμό του κάθε pixel της παραγώγου, ο γράφος εξάρτησης δεδομένων παρουσιάζεται στην Εικόνα 3. Ο γράφος αποτυπώνει τα δεδομένα που απαιτούνται για κάθε λειτουργία, ξεκινώντας με τα pixel από την αρχική εικόνα στο πάνω μέρος, με τα ενδιάμεσα αποτελέσματα να τροφοδοτούν τις εξαρτώμενες λειτουργίες, και τελικά αποδίδοντας το pixel της παραγώγου στο κάτω μέρος.

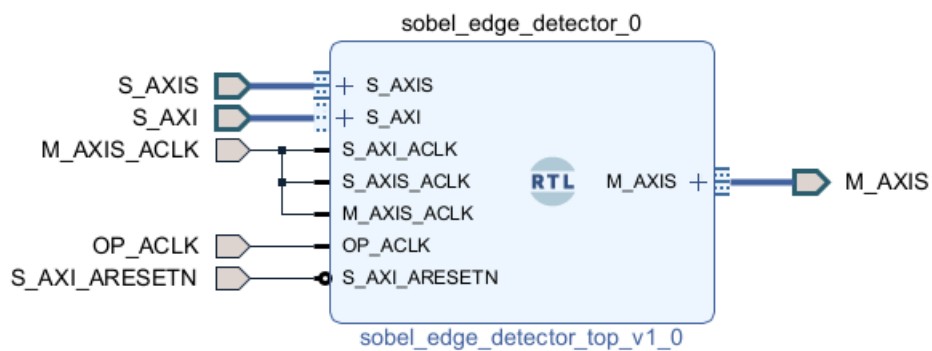
Εσκεμμένα τα μερικά γινόμενα στα οποία ο συντελεστής είναι 0 έχουν παραληφθεί, αφού αυτά δεν συνεισφέρουν στο τελικό αποτέλεσμα όπως είναι προφανές. Από το διάγραμμα είναι ξεκάθαρο ότι είναι δυνατόν να υπολογίσουμε όλα τα μερικά γινόμενα παράλληλα, αφού κάθε μερικό γινόμενο εξαρτάται μόνο από την τιμή ενός αρχικού pixel και από ένα σταθερό συντελεστή. Επίσης οι δύο απόλυτες τιμές μπορούν να υπολογιστούν εξίσου παράλληλα, πριν τις αθροίσουμε για να παράγουμε την τιμή του pixel της ολικής παραγώγου.



Εικόνα 3. Γράφος εξάρτησης δεδομένων για τον υπολογισμό ενός pixel της παραγώγου.

Τα διαφορετικά χρώματα συνιστούν στάδια διοχέτευσης

Η μονάδα ανίχνευσης ακμών θα πρέπει να είναι λειτουργική ως απομονωμένη υπομονάδα την οποία θα ελέγξετε με κατάλληλο testbench. Όμως η μονάδα θα πρέπει να διαθέτει και παράλληλα κατάλληλες διεπαφές για την διασύνδεσή της με το SoC. Η μορφή της μονάδας για να είναι συμβατή με το υπόλοιπο σύστημα θα πρέπει να είναι όπως παρουσιάζεται στην Εικόνα 4. Στο μπλοκ διάγραμμα της μονάδας διακρίνονται τρεις επιμέρους διεπαφές: μια AXI-4 Lite (S_AXI) για τον έλεγχο και πρόσβαση σε μετρητές και δύο AXI-4 Stream (S_AXI, M_AXI) για την είσοδο και έξοδο δεδομένων. Το σύστημα διαθέτει ασύγχρονο reset (S_AXI_ARESETN) καθώς επίσης και τέσσερα clock ports ένα για κάθε interface και ένα για τις εσωτερικές λειτουργίες της μονάδας αντίστοιχα. Τα σήματα clock των διεπαφών προτείνεται να λειτουργούν στα 100 MHz ενώ το εσωτερικό ρολόι στα 200 MHz. Μικρότερη συχνότητα δεν επιτρέπεται καθώς θα παραβιάσει τις προδιαγραφές του συστήματος. Αν μπορείτε να φτιάξετε ένα καλύτερο σύστημα το οποίο να είναι ικανό να λειτουργεί σε συχνότητες μεγαλύτερες από τις προαναφερόμενες χωρίς να παραβιάζει τις προδιαγραφές χρονισμού αυτό είναι βεβαίως θεμιτό.



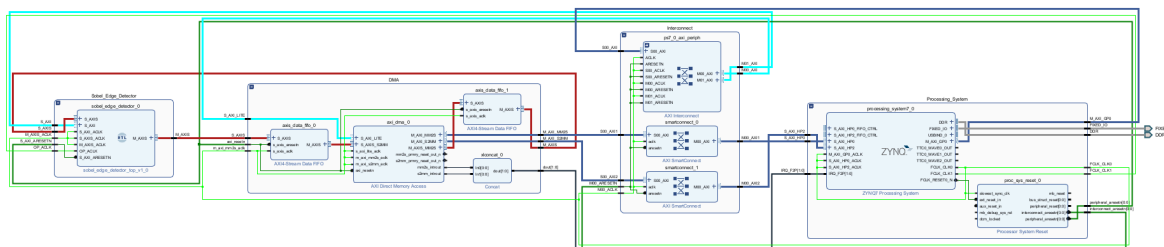
Εικόνα 4. Το μπλοκ διάγραμμα του top level του Sobel Edge Detector IP Core

Αρχιτεκτονική του Συστήματος

Η μονάδα ανίχνευσης ακμών θα πρέπει να είναι λειτουργική και ως απομονωμένο σύστημα εκτελώντας αντίστοιχες προσομοιώσεις. Παρ' όλα αυτά τελικός στόχος της εργασίας είναι η προσάρτηση της μονάδας σε ένα ολοκληρωμένο σύστημα σε τσιπ ελεγχόμενη από επεξεργαστή και προγράμματα χρήστη υψηλού επιπέδου. Η αρχιτεκτονική του Sobel Edge Detector SoC έχει την μορφή που παρουσιάζεται στην Εικόνα 5.

Τα δεδομένα της μονοχρωματικής εικόνας τύπου RAW των 8-bit θα βρίσκονται αποθηκευμένα σε μια SD. Το λειτουργικό σύστημα PetaLinux θα φορτώνει αυτά τα δεδομένα στην DRAM τεχνολογίας DDR3 του συστήματος. Η πρόσβαση στην DRAM θα επιτυγχάνεται μέσω του επεξεργαστικού συστήματος Zynq 7 και τα δεδομένα θα τροφοδοτούνται στον Sobel Edge Detector IP Core με χρήση μονάδας DMA (AXI DMA IP Core). Η μονάδα θα παράγει μια εικόνα με ακμές και μέσω του DMA IP Core θα τις επιστρέφει πάλι πίσω στην DRAM. Ο επεξεργαστής του Zynq 7 θα πρέπει έπειτα να αποθηκεύει την παραγόμενη εικόνα πίσω στην SD.

Το σύστημα αυτό πλην του Sobel Edge Detector IP Core θα σας δοθεί έτοιμο το οποίο μέσω ενός TCL script θα το αναπαράγετε και σε αυτό θα προσαρμόσετε την δική σας μονάδα για να υλοποιήσετε το επιπλέον υλικό.



Εικόνα 5. Το μπλοκ διάγραμμα του Sobel Edge Detector IP Core.

Ζητούμενα και Προδιαγραφές

1. Υλοποίηση και εκτέλεση προγράμματος C (ή C++) για την ανίχνευση ακμών με χρήση του αλγορίθμου Sobel.

- Το πρόγραμμα θα σχεδιαστεί για να τρέχει στους πυρήνες ARM Cortex A9 του Zynq 7000 PS σε λειτουργικό περιβάλλον Linux (PetaLinux) στα 667 MHz. Η χρήση του περιβάλλοντος PetaLinux προσφέρει την δυνατότητα υλοποίησης του προγράμματος σε ένα οποιοδήποτε Linux-based λειτουργικό σύστημα χωρίς να είναι απαραίτητη η άμεση επαφή με την πλατφόρμα. Μετά την επιβεβαίωση ότι το πρόγραμμα λειτουργεί ορθά θα το τρέξετε στο πραγματικό τσιπ μετά από κατάλληλο compilation.
- Το πρόγραμμα θα λαμβάνει ως είσοδο RAW μονοχρωματικές εικόνες των 8-bit, θα εφαρμόζει την διαδικασία της συνέλιξης Sobel και θα αποθηκεύει πίσω την εικόνα που αναπαριστά το μέτρο των ακμών εξίσου σε RAW format των 8-bit.
- Το πρόγραμμα θα πρέπει να είναι σε θέση να μετρά επιδόσεις για την διαδικασία της ανίχνευσης ακμών τις οποίες τιμές θα χρησιμοποιήσετε για να συγκρίνετε με την επιταχυνόμενη στο υλικό έκδοση του αλγορίθμου καθώς και να επιβεβαιώσετε την ορθή λειτουργία του IP core σας.
- Όλες οι εικόνες για την δοκιμή θα σας δίνονται. Να θεωρήσετε ότι οι διαστάσεις της εικόνας είναι δεδομένες και αφορούν ανάλυση 512x512 ως αναφορά. Χρήση επιπλέον εικόνων και άλλων διαστάσεων είναι επιτρεπτή.
- Οδηγίες για την δημιουργία του PetaLinux OS και για την ενσωμάτωση και εκτέλεση των εφαρμογών σας θα σας δοθούν.

2. Σχεδίαση του Sobel Edge Detector IP Core

- Ο ανιχνευτής ακμών θα πρέπει να επιτυγχάνει ρυθμό δεδομένων (throughput) πολύ κοντά στο 200 MSamples/sec για ρολόι 200 MHz (δηλαδή 1 Sample per second).
- Η σχεδίαση του Sobel Edge Detector IP Core θα πρέπει να γίνει έτσι ώστε να μπορεί να ελεγχθεί από κατάλληλο testbench. Εάν κρίνεται απαραίτητο, να εφαρμόσετε deep pipeline στο IP Core του ανιχνευτή ακμών για να φτάσετε τις προδιαγεγραμμένες επιδόσεις.
- Να υλοποιήσετε εσωτερικό μετρητή απόδοσης (performance counter) που να μετράει τους κύκλους ρολογιού που χρειάστηκαν για την ολοκλήρωση της επεξεργασίας της εικόνας. Ο μετρητής ξεκινάει να μετράει κύκλους ρολογιού όταν λάβει από την FIFO εισόδου το πρώτο pixel και τερματίζει όταν αποστείλει το τελευταίο αποτέλεσμα της επεξεργασίας στην FIFO εξόδου. Ο μετρητής θα έχει εύρος 4 Bytes (32bit). Η τιμή του θα είναι προσβάσιμη από την θέση μνήμης 0x43c10004 της κύριας μνήμης του Zynq 7000 από το PS.
- Να θεωρήσετε ότι οι διαστάσεις της εικόνας είναι δεδομένες και αφορούν ανάλυση 512x512 όπως και προαναφέρθηκε (χρησιμοποιήστε κατάλληλους μετρητές και κατάλληλα generics).

- Αφού υπολογίσετε τον αριθμό των bit που απαιτούνται για την ορθή αναπαράσταση των ενδιάμεσων (μερικά γινόμενα, Dx, Dy) και των τελικών τιμών των pixel $|D|$ στην συνέλιξη Sobel, στην τελική τιμή των pixel της εξόδου να υλοποιήσετε κατάλληλη κλιμάκωση πίσω στα 8-bit.
- Η διασύνδεση του IP core στην είσοδο και στην έξοδο θα είναι συμβατή με ένα AXI4-Stream FIFO interface. Χρησιμοποιήστε το FIFO Generator της Xilinx για να παράγετε AXI4-Stream FIFO με δεδομένα των 8-bit (TDATA bytes=1) και το TLAST που όταν είναι ενεργοποιημένο σηματοδοτεί το τέλος ενός πακέτου (τελευταίο pixel τρέχουσας εικόνας), και η επόμενη τιμή στα δεδομένα αφορά την πρώτη τιμή από το επόμενο πακέτο (πρώτο Pixel επόμενης εικόνας).
- Ακόμη θα υλοποιήσετε μια AXI4-Lite διεπαφή με τέσσερις καταχωρητές (memory mapped registers) των 32-bit με σκοπό τον έλεγχο της μονάδας και την πρόσβαση σε μετρητές απόδοσης άμεσα προσπελάσιμους από το PS. Ο ακόλουθος πίνακας περιέχει τις διευθύνσεις μνήμης που θα αντιστοιχούν σε κάθε καταχωρητή καθώς και το περιεχόμενό του.

Πίνακας 1 - AXI4 - Lite register address map

Καταχωρητής	Διεύθυνση	Σκοπός	R/ W
reg_1	0x43c10000	[0:0] → όταν είναι '1' ενεργοποιεί την υπομονάδα ανίχνευσης ακμών. Όταν είναι '0' απενεργοποιεί την μονάδα και την κρατά σε κατάσταση αρχικοποίησης idle.	W/ R
reg_2	0x43c10004	[31:0] → ο αριθμό των κύκλων ρολογιού από την έναρξη της επεξεργασίας έως την λήξη της.	R
reg_3	0x43c10008	[31:0] → ο αριθμός των εικονοστοιχείων που διαβάστηκαν από την μονάδα ανίχνευσης ακμών.	R
reg_4	0x43c1000c	[31:0] → ο αριθμός των εικονοστοιχείων που γράφτηκαν από την μονάδα ανίχνευσης ακμών	R

- Για την επαλήθευση της ορθής σχεδίασης του IP core που υλοποιεί την ανίχνευση ακμών Sobel με προσομοίωση να γράψετε κατάλληλο VHDL testbench. Συγκρίνεται τα αποτελέσματα με το πρόγραμμα που φτιάξατε προηγουμένως.

3. Σύνδεση του Sobel Edge Detector IP Core με το SoC

- Θα σας δοθεί ένα ημιτελές block διάγραμμα για την αρχιτεκτονική του Sobel Edge Detector SoC υπό μορφή TCL script. Σε αυτό το διάγραμμα θα προσαρτήσετε καταλλήλως τον δικό σας Sobel Edge Detector IP Core και θα εξάγετε το γενικό σύστημα του Sobel Edge Detector SoC υπό μορφή

XSA με ενσωματωμένο το bitstream. Οδηγίες για την αναπαραγωγή του συστήματος από TCL script θα σας δοθούν.

- ΠΡΟΣΟΧΗ! Για τον Sobel Edge Detector IP Core να χρησιμοποιήσετε την έξοδο FCLK_CLK1 από τον Zynq 7000 PS η οποία είναι ρυθμισμένη στα 200 MHz. Αν μετά την υλοποίηση δεν ικανοποιεί τις χρονικές απαιτήσεις να ανατρέξετε πίσω στο σχέδιο του Sobel Edge Detector IP Core και να κάνετε κατάλληλες ρυθμίσεις. Είναι υποχρεωτικό το σύστημα να τρέχει τουλάχιστον στα 200 MHz!
- Θα χρησιμοποιήσετε το XSA για την δημιουργία ενός PetaLinux OS στο οποίο και θα τρέξετε την εφαρμογή ελεγκτή του SoC. Η εφαρμογή συλλέγει στατιστικά τα οποία θα χρησιμοποιήσετε για να εκτιμήσετε την απόδοση του design σας.
- Οδηγίες για την δημιουργία του PetaLinux OS θα σας δίνονται καθώς επίσης και όλα τα απαραίτητα αρχεία κώδικα για την σύνθεση της εφαρμογής ελεγκτή του SoC.