



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΗΜΜΥ

Ψηφιακά Συστήματα VLSI

Ακαδημαϊκό έτος 2020-21

4^η Εργαστηριακή Άσκηση

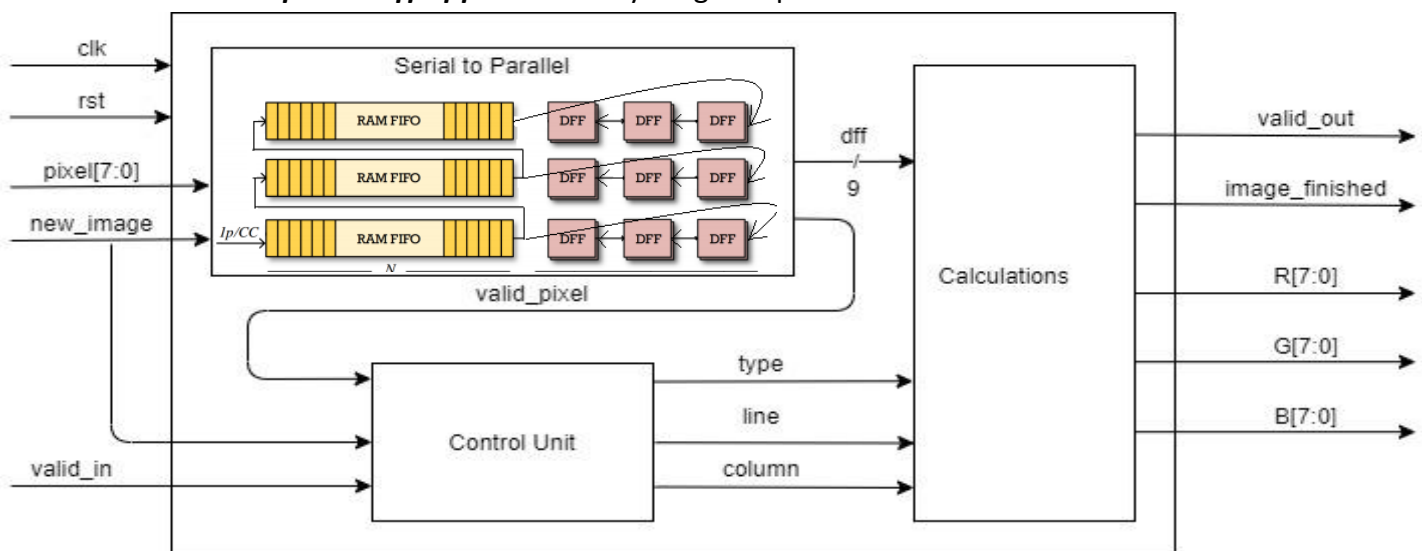
Υλοποίηση Debayering Φίλτρου σε FPGA

Αιμιλία Σιόκουρου – 03117703

Νάταλυ Πεγειώτη – 03117707

**Όλοι οι κώδικες της VHDL παρατίθενται σε ξεχωριστά αρχεία*

Ακολουθεί το **δομικό διάγραμμα** του Debayering Φίλτρου



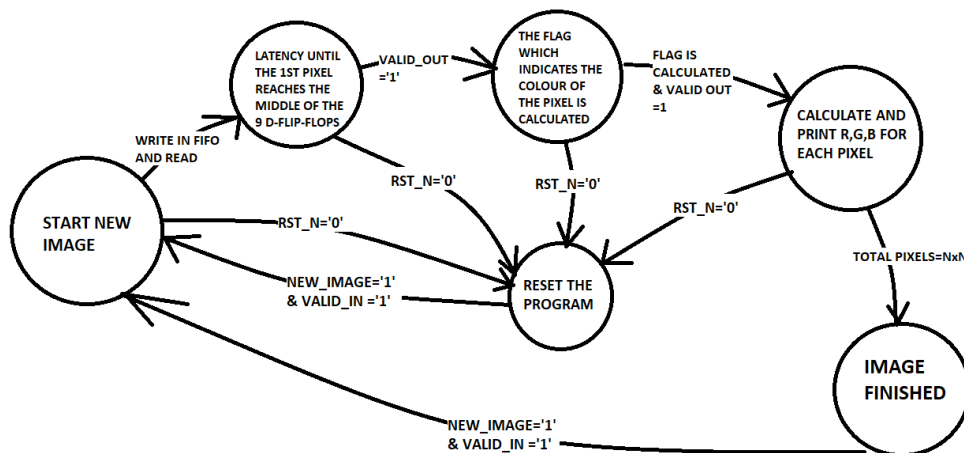
Βασικές Δομές του Κυκλώματος

Control Unit: όταν δοθεί το σήμα `new_image='1'` και παράλληλα το `pixel` που δίνεται είναι `valid` ξεκινάει η μέτρηση των `valid_in` pixels σε κάθε γραμμή καθώς και η μέτρηση των γραμμών της εικόνας. Έχοντας ορίσει ως generic το `N` που αντιπροσωπεύει το μήκος της κάθε γραμμής καθώς και το πλήθος των γραμμών, το `valid_in_counter` αυξάνεται για κάθε `valid pixel` που δίνεται ενώ το `line_counter` αυξάνεται όταν το `valid_in_counter` γίνει ίσο με `N`, και τότε το `valid_in_counter` αρχικοποιείται πάλι στο 1. Δηλαδή για παραδειγμα έχει γεμίσει τα `N` στοιχεία της 1ης γραμμής και ξεκινάει την καταγραφή της δεύτερης και ούτω καθεξής.

Serial_to_parallel: περιλαμβάνονται οι 3 fifo καθώς και τα 9 d-flip-flops όπως φαίνεται και στο πιο πάνω σχημα. Συγκεκριμένα, όταν δοθεί `new_image='1'` ενεργοποιείται η εγγραφή στην 1η fifo (σχηματικά η πιο κάτω) και δίνεται το 1ο pixel. Δίνονται νέα pixel σε κάθε κύκλο ρολογιού. Όταν γεμίσει η πρώτη fifo (δηλαδή `valid_count > N`) ξεκινάει η ανάγνωση των pixel και η τοποθέτηση τους στα d-flip-flop όπως φαίνεται στο παραπάνω σχήμα. Σε κάθε κύκλο ρολογιού από εδώ και στο εξής πέρα από το να εγγράφονται και να διαβάζονται pixel γίνεται και μία αριστερή ολίσθηση στα dff. Μετά από $2*N+2+3$ κύκλους ρολογιού από την στιγμή που έγινε το `new_image='1'` ($2*N$: για να έχουν γεμίσει οι πρώτες 2 fifo, 2: ώστε να έχει ολισθήσει το 1ο pixel στην μεσαία θέση της 2ης σειράς dff και 3: καθυστέρηση έτσι ώστε τα αποτελέσματα από κάθε fifo να βγαίνουν ταυτόχρονα), βγαίνουν στην έξοδο οι τιμές που βρίσκονται σε κάθε dff για να τις λάβει η μονάδα calculations.

Calculations: λαμβάνει σαν εισόδους τις εξόδους των δύο προηγούμενων μονάδων (`valid_count`, `line_count`, `flag` και τα 9 dff) και ανάλογα με τη τιμή των `valid_count`, `line_count` και `flag` υπολογίζει τον μέσο όρο των RGB για κάθε Pixel όπως μας δίνεται στην εκφώνηση. (Αν το κελί είναι πράσινο (περ. i, ii), η κόκκινη (μπλε) χρωματική του συνιστώσα υπολογίζεται ως ο μέσος όρος των δύο κόκκινων (μπλε) γειτονικών pixels. Αν αυτό είναι κόκκινο (περ. iii), η πράσινη (μπλε) χρωματική του συνιστώσα υπολογίζεται ως ο μέσος όρος των τεσσάρων πράσινων (μπλε) γειτονικών pixels. Αν αυτό είναι μπλε (περ. iv), η πράσινη (κόκκινη) χρωματική του συνιστώσα υπολογίζεται ως ο μέσος όρος των τεσσάρων πράσινων (κόκκινων) γειτονικών pixels. Στις γωνίες και στις ακριανές στήλες και γραμμές γίνεται padding με 0.

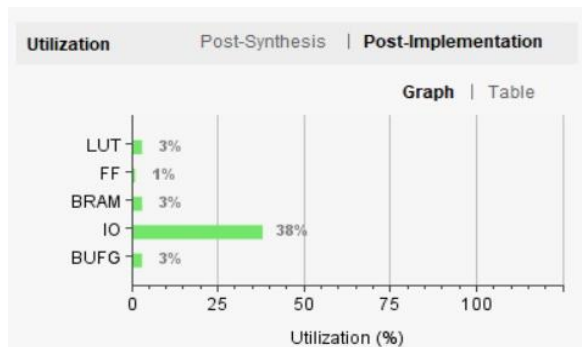
Ακολουθεί το **διάγραμμα μηχανής πεπερασμένων καταστάσεων** του Debayering Φίλτρου



Κατανάλωση Πόρων του FPGA

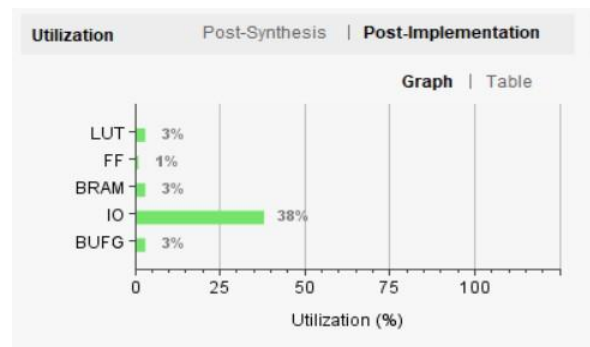
Υπολογίστηκαν οι πόροι του FPGA που καταναλώθηκαν για $N = 64, 128$. Τα αποτελέσματα φαίνονται ακολούθως:

N = 64



Utilization			
		Post-Synthesis	Post-Implementation
Graph Table			
Resource	Utilization	Available	Utilization %
LUT	509	17600	2.89
FF	418	35200	1.19
BRAM	1.50	60	2.50
IO	38	100	38.00
BUFG	1	32	3.13

N=128



Utilization			
		Post-Synthesis	Post-Implementation
Graph Table			
Resource	Utilization	Available	Utilization %
LUT	507	17600	2.88
FF	412	35200	1.17
BRAM	1.50	60	2.50
IO	38	100	38.00
BUFG	1	32	3.13

Παρατηρούμε ότι στις δύο περιπτώσεις οι πόροι που καταναλώνονται είναι σχεδόν ίσοι. Ελάχιστες διαφορές παρατηρούνται στα LUTs και τα Flip-Flops.

Υπολογισμός του Latency

Όπως έχει εξηγηθεί και πιο πάνω η καθυστέρηση έχει υπολογιστεί ως $T_{latency} = 2*N + 2 + 3$ κύκλους. $2*N$: για να έχουν γεμίσει οι πρώτες 2 fifo, 2: ώστε να έχει ολισθήσει το 1ο pixel στην μεσαία θέση της 2ης σειράς dff και 3: καθυστέρηση έτσι ώστε τα αποτελέσματα απο κάθε fifo να βγαίνουν ταυτόχρονα.

Throughput

Η απόδοση υπολογίζεται ως εξής:

throughput = κύκλοι ρολογιού που παράγεται έγκυρη έξοδος/συνολικοί κύκλοι ρολογιού