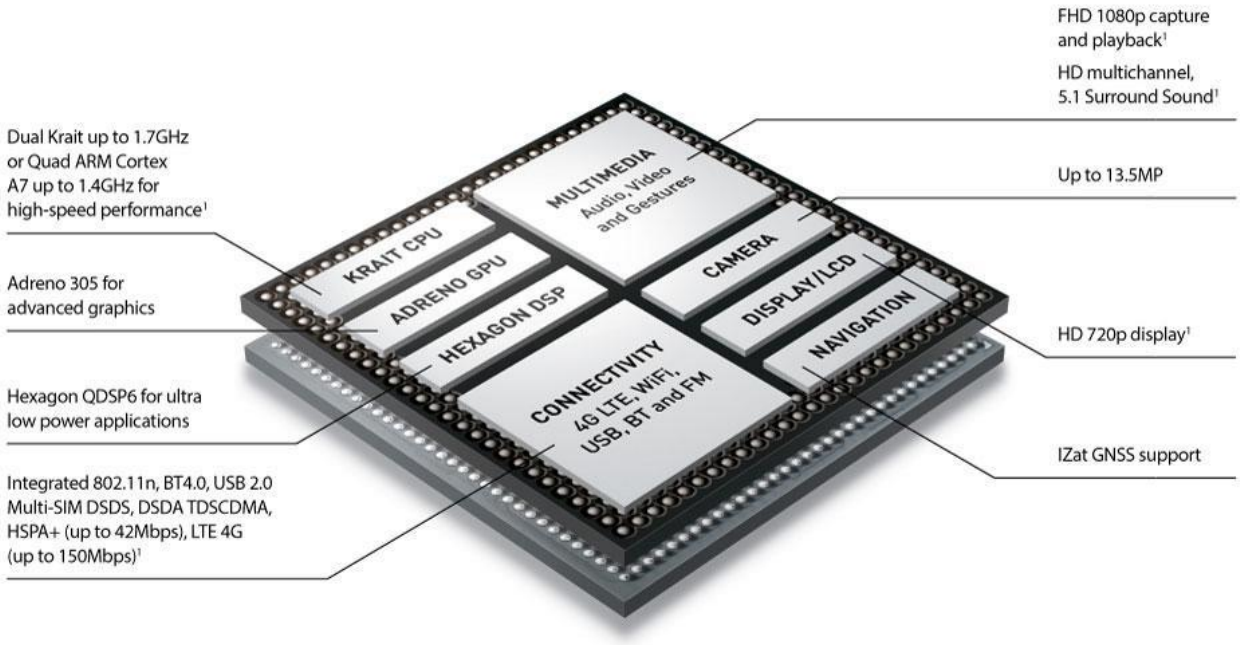
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας

CE333
ΕΡΓΑΣΤΗΡΙΟ ΣΧΕΔΙΑΣΗΣ SoC ΜΕ ΕΡΓΑΛΕΙΑ CAD



REPORT 1: ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ

ΠΑΤΣΙΑΝΩΤΑΚΗΣ ΧΑΡΑΛΑΜΠΟΣ
ΚΟΞΕΝΟΓΛΟΥ ΝΙΚΟΛΑΟΣ

Εισαγωγή

Στην Τεχνική Έκθεση αυτή αναλύεται με ποιο SoC έχει σκοπό η ομάδα να ασχοληθεί. Προτείνονται δύο ιδέες, η μία ως η κύρια ιδέα που επιθυμούμε να υλοποιήσουμε και έπειτα μια δεύτερη ως εφεδρική σε περίπτωση που δεν μπορεί να δουλέψει η πρώτη. Και στις δύο περιπτώσεις αναφέρουμε συνοπτικά την δομή, την μέθοδο συναρμολόγησης που θα ακολουθήσουμε και τα προβλήματα που μπορούν να εμφανισθούν.

Nisiotes:

Πατσιανωτάκης Χαράλαμπος 2116

Κοξένογλου Νικόλαος 1711

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Παρουσίαση του Συστήματος προς Υλοποίηση

Δύο είναι τα συστήματα που έχουμε σκεφθεί να κατασκευάσουμε κατά την διάρκεια του εξαμήνου. Αρχικά σκεφθήκαμε ένα είδος παρεμφερές σε σύστημα Μαζικής Επεξεργασίας Δεδομένων (GPU like), βασισμένο στην SIMD κλάση παράλληλων υπολογιστών. Έπειτα, σε περίπτωση που αυτό δεν δουλέψει θα κατευθυνθούμε στην εναλλακτική λύση, η οποία είναι υλοποίηση μίας Αριθμητικής Λογικής Μονάδας, η οποία γίνεται για να δωθεί έμφαση στην ιδέα του stalling.

Α. ΣΥΣΤΗΜΑ ΑΠΛΗΣ ΕΝΤΟΛΗΣ ΠΟΛΛΑΠΛΩΝ ΔΕΔΟΜΕΝΩΝ - SIMD

Γενικά η SIMD (Single Instruction Multiple Data) είναι ένα σύστημα το οποίο για κάθε εντολή που δέχεται, διαιρεί το σύνολο των δεδομένων της εντολής και εκτελεί την εντολή για κάθε υποσύνολο δεδομένων ξεχωριστά και παράλληλα, την ίδια χρονική στιγμή. Σκοπός του συστήματος είναι η αποφυγή σειριακής μετατροπής των τμημάτων δεδομένων, επιταχύνοντας την διαδικασία, όταν δεν υπάρχουν αλληλοεξαρτήσεις κατά την διάρκεια των πράξεων. Σε επίπεδο εφαρμογής χρησιμοποιείται για προσαρμογή αντίθεσης σε ψηφιακές εικόνες και προσαρμογή έντασης ψηφιακού ήχου.

Οπότε προφανώς θα αποτελείται από πολλαπλές υπολογιστικές μονάδες, παράλληλα και ομοιόμορφα τοποθετημένες μεταξύ τους και μνήμη για την διατήρηση των δεδομένων ως το πέρας της επεξεργασίας.

Στην προκειμένη περίπτωση θα απλουστεύσουμε το σύστημα, τοποθετώντας δύο μόνο υπολογιστικές μονάδες. Περισσότερη ανάλυση θα γίνει παρακάτω.

Β. ΑΡΙΘΜΗΤΙΚΗ ΛΟΓΙΚΗ ΜΟΝΑΔΑ - ALU

Μια ALU (Arithmetic Logic Unit) είναι ένα σύστημα που δέχεται στην είσοδο της δεδομένα και κάνει συγκεκριμένα (έχουν ορισθεί με την κατασκευή του hardware) αριθμητικές πράξεις μεταξύ τους. Σε επίπεδο εφαρμογής εντάσσεται σε άλλα συστήματα, όπου πραγματοποιεί τις όποιες πράξεις εκείνα χρειάζονται.

Οπότε υποθέτουμε βασικές τεχνικές παράμετροι της θα είναι διάφορα modules που το καθένα εκτελεί από μία πράξη.

ΜΕΡΟΣ Α: ΣΥΝΘΕΣΗ SIMD

1.Προτεινόμενη Δομή Υλοποίησης του Συστήματος

Το σύστημα αποτελείται από τις ακόλουθες μονάδες:

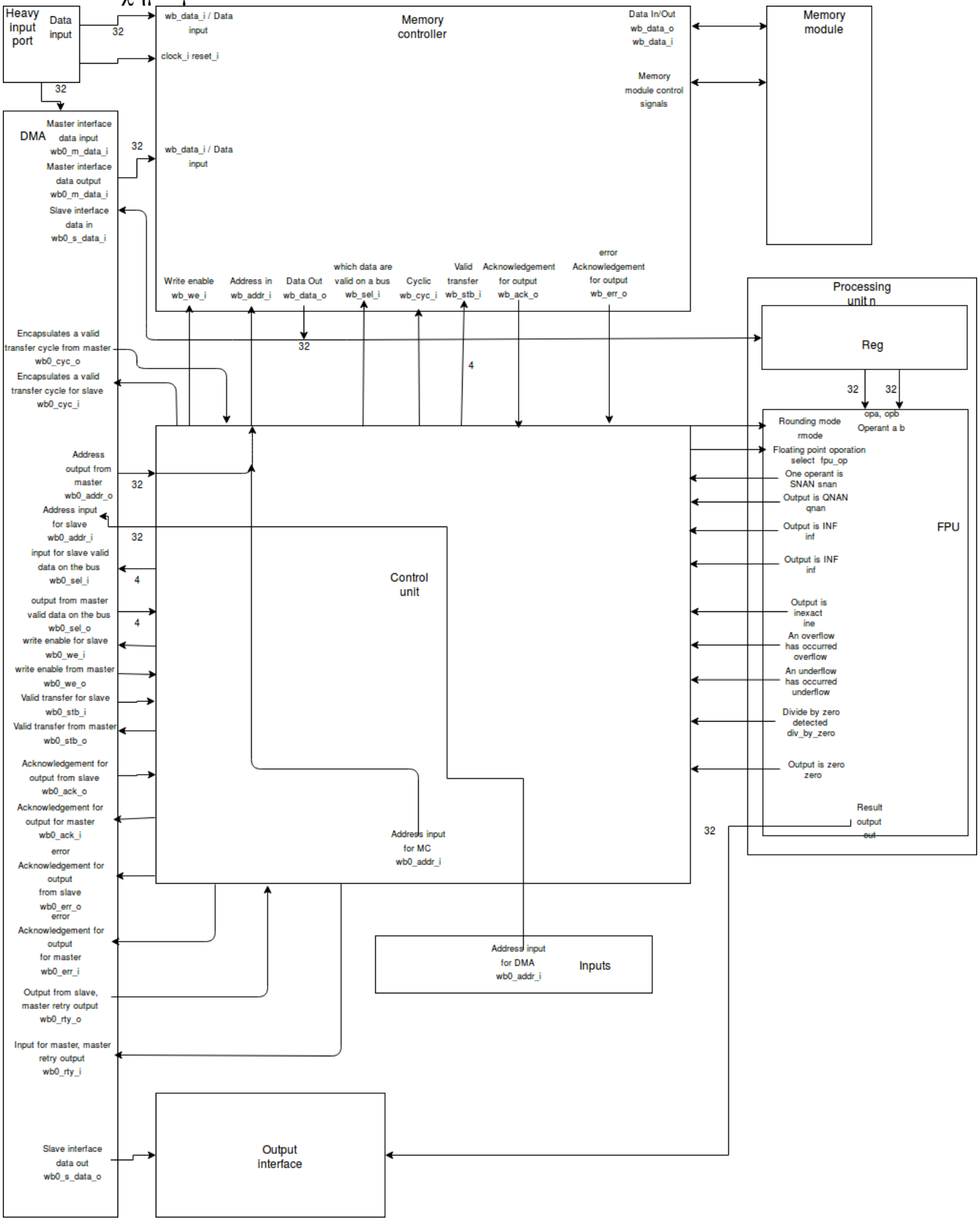
- Ένα Control Unit
- Ένα Memory Controller Unit (MCU)
- Δύο Processing Units, αποτελούμενα καθένα από:
  - Ένα module από καταχωρητές.
  - Και δύο FPU.
- Ένα Output Interface
- Ένα σύνολο Εισόδων (Heavy Input Port HIP, Inputs)

Περιγραφή συνδεσμολογίας και λειτουργίας συστήματος:

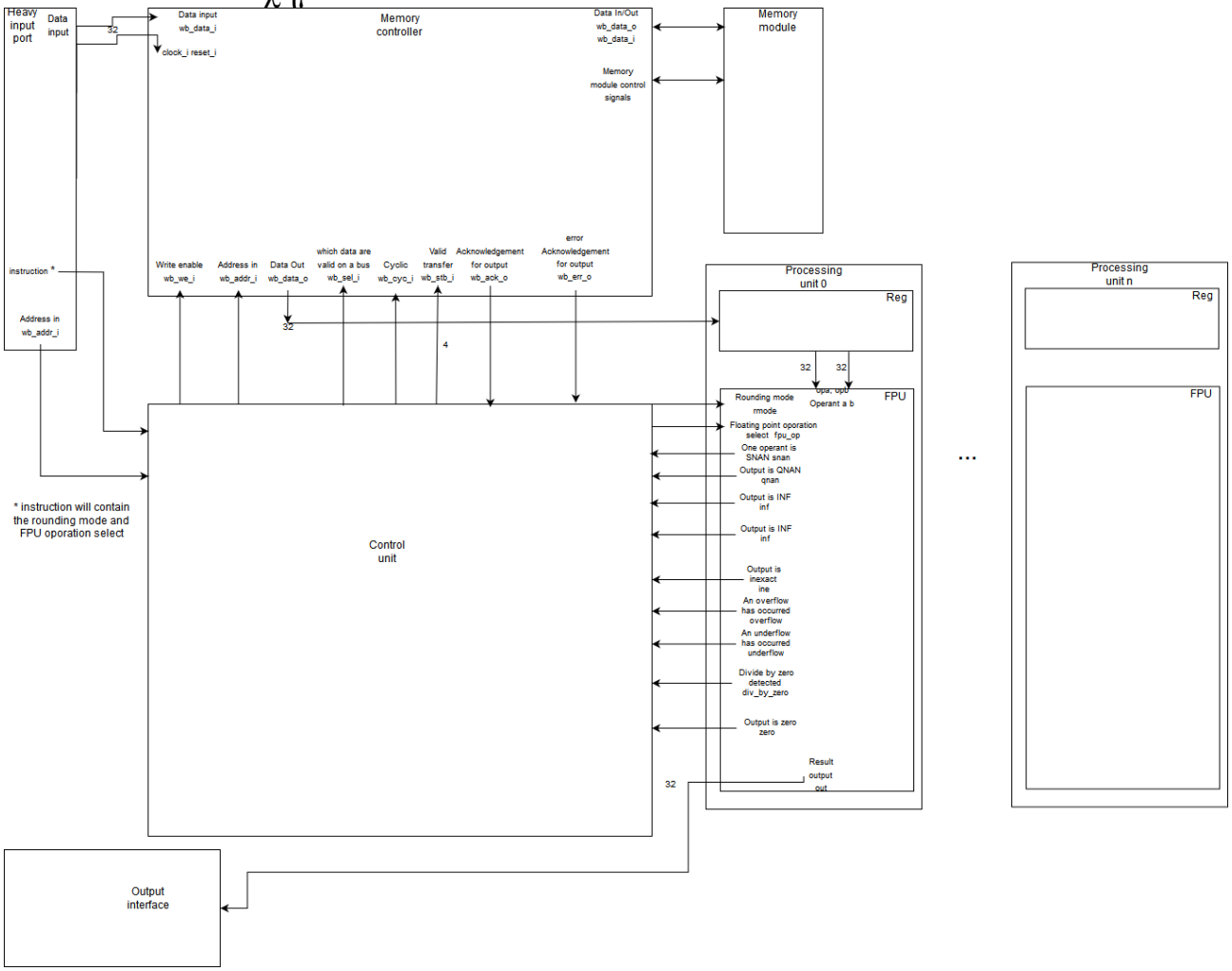
- ➢ Το Control Unit είναι υπεύθυνο για τα όποια control signals απαιτούνται στο σύστημα γενικά.
- ➢ Η είσοδος από το HIP αποθήκευεται (μέσω MCU) στην μνήμη του συστήματος.
- ➢ Τα 2 Processing Units παίρνουν τα δεδομένα που χρειάζονται από την μνήμη (μέσω MCU)
- ➢ Τα αποτελέσματα από τα Processing Units καταλήγουν στο Output Interface για να βγούν στην έξοδο.

Σημ: Αρχικά είχαμε σκεφθεί να εισάγουμε και έναν DMA Controller, ώστε να βοηθήσουμε την επικοινωνία του εξωτερικού κόσμου με την μνήμη. Έπειτα από περισσότερη ανάλυση, αποφασίσθηκε πως αυτό είναι περιττό για την απλή λειτουργία του συστήματος και η ιδέα απορρίφθηκε.

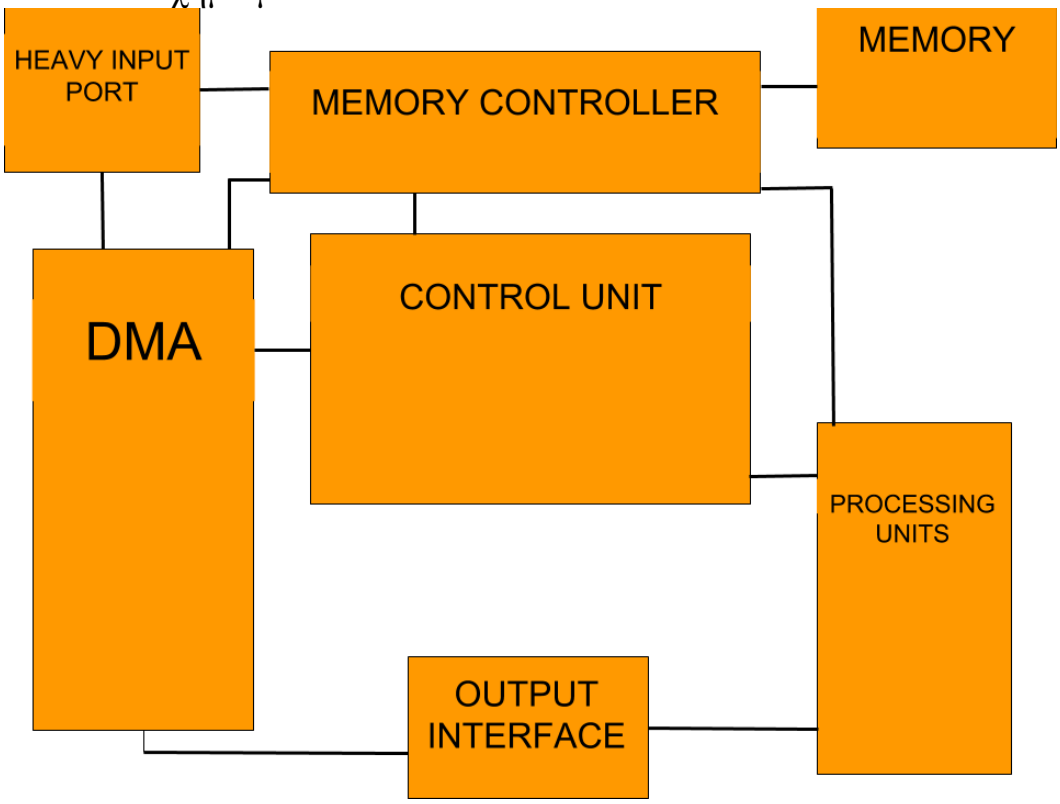
- Το σχήμα με το DMA αναλυτικά:



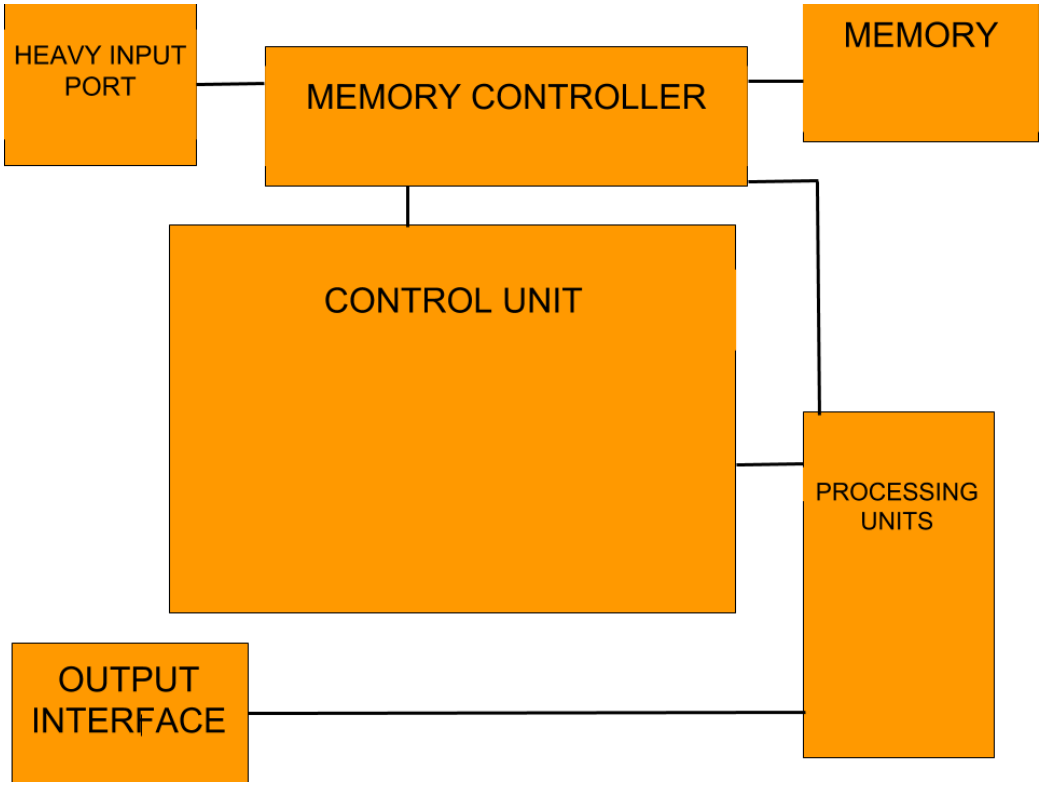
- Το τελικό σχήμα αναλυτικά:



- Το σχήμα με DMA συνοπτικά:



- Το τελικό σχήμα συνοπτικά:



Σε περίπτωση που εμφανισθεί κάποιο πρόβλημα* στην θέση της μονάδας μνήμης θα τοποθετηθεί ένας πίνακας από καταχωρητές ή γενικά απλούστερες μονάδες μνήμης, οι οποίες δεν χρειάζονται οδήγηση από πολύπλοκες μονάδες Memory Controller.

Βασικό πλεονέκτημα της προτεινόμενης υλοποίησης είναι ότι αποτελεί μια μικρογραφία ενός processing unit μιας GPU. Επίσης άλλο πλεονέκτημα είναι το κοινό πρωτόκολλο που ακολουθούν οι δύο βασικές μονάδες πράγμα που μειώνει πάρα πολύ τις πιθανότητες αποτυχημένης επικοινωνίας. Μειονέκτημα είναι η μεγάλη εσωτερική πολυπλοκότητα των μονάδων η οποία μπορεί να προκαλέσει πρόβλημα στο area που θα καταλαμβάνει το SoC και την αύξηση χρόνου που απαιτείται από τα εργαλεία για επεξεργασία του Design.

*Θα αναλυθούν στο αντίστοιχο πεδίο

2. Τεχνικές Προδιαγραφές του Συστήματος

Δεδομένα που γνωρίζουμε για τις μονάδες που επιλέξαμε:

- ➢ FPU:
  - Ειναι fully IEEE754 complient
  - Μπορεί να εκτελέσει μια πράξη floating point σε κάθε κύκλο.
  - Κάθε αποτέλεσμα χρειάζεται 4 κύκλους για να βγει.
- ➢ MCU:
  - Υποστηρίζει SDRAM, SSRAM, FLASH, ROM και άλλους τύπους memory modules
  - Έχει ευέλικτο timing για να μπορεί να υποστηρίξει μια πληθώρα από memory modules
- ➢ DMA Controller(αφαιρέθηκε εν τέλη) :
  - Υποστηρίζει hardware handshaking
  - Χρησιμοποιεί το DMA request and aknowledge μέσω hardware handshaking.

Προδιαγραφές που απαιτήσαμε από τα components:

- ➢ Εύκολη και κατανοητή συνδεσιμότητα για MCU και DMA (WishBone complient). (με την αφαίρεση του DMA Controller δεν ήταν πλεόν αναγκαίο)
  - Για ομοιόμορφο σχεδιασμό Control Unit
  - Για μεγαλύτερη πιθανότητα επιτυχούς επικοινωνίας μεταξύ των μονάδων.
- ➢ Το Processing Unit να είναι όσο απλούστερο γίνεται.
  - Εφόσον ένα Manycore Accelerator αποτελείται από πολλαπλά μικρά Execution Units, έπρεπε (εφόσον το σύστημα είναι GPU like) να ακολουθήσουμε αυτήν την πολιτική.

Ρίσκο υπάρχει εννοείται από τις τεχνικές προδιαγραφές, όπως:

- ➢ Να μην υπάρξει δυνατότητα μείωσης critical path, ώστε να ικανοποιηθούν οι χρόνοι για τις διαφορετικές συχνότητες.
- ➢ Να προκύψουν bugs στον έτοιμο κώδικα Verilog που δανειστήκαμε.

Σε περίπτωση που τα προβλήματα αυτά αποδειχθούν σοβαρά και μη επιλύσιμα σε σύντομο χρονικό διάστημα, θα μεταβούμε στο σύστημα που αναγράφεται στο Μέρος Β.

3.Στρατηγική Συναρμολόγησης, Υλοποίησης και Επαλήθευσης

Σε περίπτωση που δεν υπάρξουν προβλήματα με τις έτοιμες μονάδες, απαιτείται η υλοποίηση μόνο του Control Unit. Αυτή θα λαμβάνει τα σήματα ελέγχου που εκπέμπουν οι διάφορες μονάδες και θα διαχειρίζεται την κατάσταση εσωτερικά του συστήματος αλλά και την επικοινωνία του με τον έξω κόσμο.

Για την επαλήθευση του Control Unit, θα χρησιμοποιήσουμε set εισόδων που θα προσομοιώνουν διάφορα πιθανά αιτήματα από εξωτερικές οντότητες προς την μονάδα, ελέγχοντας έτσι αν τα σήματα που στέλνονται από την μονάδα είναι τα σωστά για την ορθή λειτουργία του SoC

Για την επαλήθευση των έτοιμων μονάδων θα χρησιμοποιηθούν διάφορα σενάρια εισόδων, και θα εξετάσουμε πόσο οι εξόδοι ανταποκρίνονται σωστά σε αυτές. Έπειτα η κάθε έτοιμη μονάδα θα συνδεθεί μεμονωμένα με το Control Unit και θα ελεγχθεί η συμπεριφορά του. Τέλος θα γίνει μια συνολική προσομοίωση επαλήθευσης με όλα τα τμήματα διασυνδεδεμένα, εισάγοντας ένα σύνολο πιθανών εισόδων.

4.Ανάλυση Ρίσκου

Προβλήματα που μπορούν να παρουσιαστούν κατά την διάρκεια της εργασίας είναι:

- ➢ Αδυναμία ορθής επικοινωνίας Control Unit με τις διάφορες μονάδες
  - Έλεγχος Control Unit και αύξηση πολυπλοκότητας όπου είναι ανάγκη
- ➢ Εμφάνιση bugs στις έτοιμες μονάδες
  - Προσπάθεια επίλυσης τους, και αν δεν είναι αυτό εφικτό, μετάβαση στο πρότζεκτ Μέρους Β’.
- ➢ Μεγάλο design που αναγκάζει τα εργαλεία να δημιουργεί και επεξεργάζει τα designs σε αρκετά μεγάλα χρονικά διαστήματα
  - Απλούστευση των μονάδων (Με αυτήν την λογική αφαιρέθηκε ο DMA Controller) ή
  - εναλλαγή σε προτζεκτ Μέρους Β’.
- ➢ Incompatible adressing μεταξύ των δομών διευθύνσεων που θα δέχεται το MCU και αυτών που θα υπάρχουν στις έτοιμες βιβλιοθήκες.
  - Αλλαγή MCU, αναζήτηση διαφορετικών μνημών ή μετάβαση στο project του ALU.
- ➢ Εμφάνιση bugs στα εργαλεία CAD
  - Αναζήτηση στα manuals και παράκαμψη ή διαφορετικη πορεία εαν αυτό είναι εφικτό.

ΜΕΡΟΣ Β: ΣΥΝΘΕΣΗ ΑΡΙΘΜΗΤΙΚΗΣ ΛΟΓΙΚΗΣ ΜΟΝΑΔΑΣ

1.Προτεινόμενη Δομή Υλοποίησης του Συστήματος και αξιολογηση

Το σύστημα αποτελείται από τις ακόλουθες μονάδες:

- Ένα Control Unit
- Έναν αθροιστή
- Έναν αφαιρέτη
- Εναν multiplier
- Έναν divider
- Έναν shifter
- Έναν πολυπλέκτη για επιλογή έγκυρου αποτελέσματος
- Δύο σειρές από flip flop για input-output

Η είσοδος στο σύστημα θα γίνεται μέσω μιας σειράς από flip flop. Η είσοδος έχει την εξής μορφή:

opcode	**Data 1**	**Data 2**
3 bits	16 bits	16 bits
35 bits		

Στο πεδίο opcode γίνεται η δήλωση της πράξης που επιθυμούμε να γίνει μεταξύ Data 1 και Data 2. Η κωδικοποίηση των εντολών έχει ως εξής:

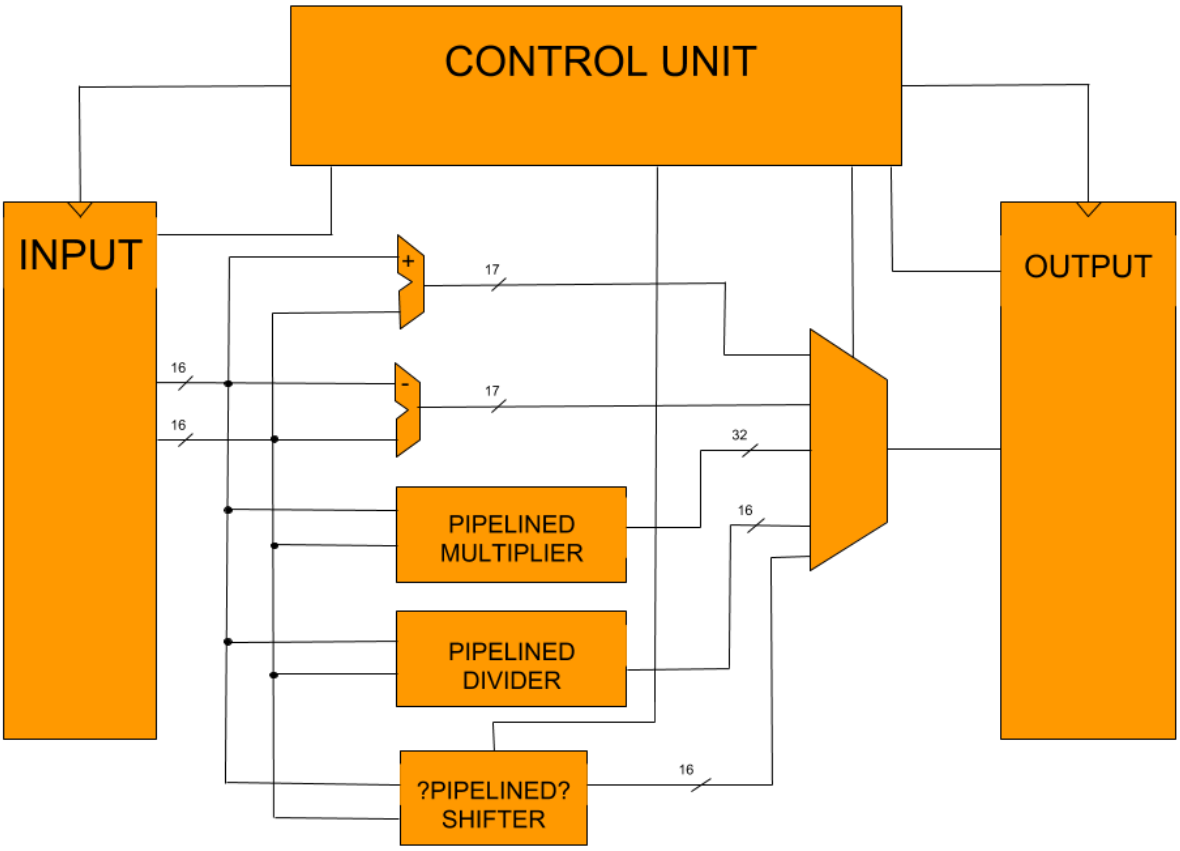
opcode	**function**
000	NO_FUNCTION
001	ADDITION
010	SUBSTITUTION
011	MULTIPLICATION
100	DIVISION
101	SHIFT RIGHT
110	SHIFT LEFT
111	NO_FUNCTION

Έπειτα αφού δοθεί έγκυρο opcode από την είσοδο τα δεδομένα εκτελούνται από την κάθε πράξη παράλληλα. Έπειτα το σύνολο των αποτελεσμάτων καταλήγει σε έναν πολυπλέκτη όπου ανάλογα την πράξη που επιθυμούμε να εκτελέσουμε, προωθείται και το αντίστοιχο αποτέλεσμα. Τέλος τα αποτελέσματα καταγράφονται στο τελικό σύνολο από flip flops που αντιστοιχεί στην έξοδο.

Δεδομένου όμως της διαφορετικής καθυστέρησης που έχει η κάθε πράξη, θα εφαρμοστεί η μέθοδος του Stalling. Δηλαδή σε περίπτωση που μία χρονοβόρα πράξη ακολουθεί μια σύντομη, κατά την διάρκεια που έχει τελειώσει η σύντομη και εκτελείται η χρονοβόρα, η έξοδος θα διατηρείται σταθερή και τιμές εισόδου δεν θα λαμβάνονται υπόψιν.

Πλεονέκτημα της συγκεκριμένης δομής είναι η αποφυγή καθυστέρησης όταν δεν χρειάζεται (π.χ.αποφεύγεται το γεγονός που το αποτέλεσμα μιας πρόσθεσης να εμφανισθεί στην έξοδο με την ίδια καθυστέρηση που έχει το αποτέλεσμα ενός πολ/σμου που έχει μεγαλύτερη πολυπλοκότητα και μεγαλύτερα μονοπάτια). Όμως σε περίπτωση που παγώσει η είσοδος και η έξοδος, τυχόν δεδομένα από πράξεις που θα ήθελε να στείλει ένα άλλο component χάνονται, εφόσον καμία είσοδος δεν γίνεται δεκτή.

Το σχήμα συνοπτικά:


CONTROL UNIT
INPUT
OUTPUT
+
-
17
17
16
16
32
16
16
PIPELINED
MULTIPLIER
PIPELINED
DIVIDER
?PIPELINED?
SHIFTER

2.Τεχνικές Προδιαγραφές του Συστήματος

Για κάθε μονάδα αναγκαστικά θα απορροφήσουμε στοιχεία του, τα οποία και θα χρησιμοποιήσουμε:

- Το Control Unit το κατασκευάζουμε εμείς ανάλογα τα υπόλοιπα στοιχεία του κυκλώματος

- Στον αθροιστή:
    - Αρκούν οι είσοδοι - έξοδοι δεδομένων για την πράξη πρόσθεσης

- Στον αφαιρέτη 16 bit:
    - Αρκούν οι είσοδοι - έξοδοι δεδομένων για την πράξη αφαίρεσης

- Στον multiplier 16 bit:
    - Οι είσοδοι - έξοδοι δεδομένων για την πράξη πολλαπλασιασμού
    - Στην εσωτερική δομή το σύνολο των pipeline stages ώστε να γνωρίζουμε την επιπλέον καθυστέρηση

- Στον divider 16 bit:
    - Οι είσοδοι - έξοδοι δεδομένων για την πράξη διαίρεσης
    - Στην εσωτερική δομή το σύνολο των pipeline stages ώστε να γνωρίζουμε την επιπλέον καθυστέρηση

- Στον shifter:
    - Οι είσοδοι - έξοδοι δεδομένων για την πράξη ολίσθησης.
    - Τις εισόδους δήλωσης αριστερής και δεξιάς ολίσθησης.
    - Στην εσωτερική δομή το σύνολο των pipeline stages (αν υπάρχει) ώστε να γνωρίζουμε την επιπλέον καθυστέρηση.

Προδιαγραφές που απαιτήσαμε από τα components:

➢ Κατανοήσιμη εσωτερική υλοποίηση.
    - Για μετατροπή εαν χρειάζεται στα pipeline stages στο εσωτερικό τους.
    - Για να ελέγξουμε εάν απαιτείται επιπλέον έλεγχος σε αυτά μέσω Control Unit.

Σε περίπτωση που υπάρξουν σφάλματα μέσα στις αρχικές υλοποιήσεις, θα γίνει ανάλυση του κώδικά τους και προσπάθεια διόρθωσης τους.

3.Στρατηγική Συναρμολόγησης, Υλοποίησης και Επαλήθευσης

Θα απαιτηθεί να κατασκευάσουμε το Control Unit εξ αρχής, καθώς και να προσθέσουμε τα Pipeline Stages στα modules που θα παραλάβουμε έτοιμα από την σελίδα: http://www.ellab.physics.upatras.gr/~bakalis/Eudoxus/ .
Οπότε θα ελέγξουμε και αν και πόσα stages χρειάζεται το κάθε component.

Αρχικά θα ελέγξουμε την κάθε μονάδα ξεχωριστά, βάζοντας τις κατάλληλες αριθμητικές εισόδους μέσω testbench και ελέγχοντας το αριθμητικό αποτέλεσμα στην έξοδο και τους κύκλους που κάνει. Έπειτα θα ελέγξουμε αν το Control Unit δίνει τα σωστά σήματα εξόδου, ανάλογα τις εισόδους του (θα δίνει σωστές εντολές, ανάλογα αυτά που βλέπει). Τέλος θα γίνει μια επαλήθευση για όλο το σύστημα, με όλες τις δυνατές εισόδους (Όλο το σύνολο δεδομένων θα ελεγχθεί για κάθε πιθανή σειρά εντολών).

4.Ανάλυση Ρίσκου

Προβλήματα που μπορούν να παρουσιαστούν κατά την διάρκεια της εργασίας είναι:

- ➢ Αδυναμία ορθής επικοινωνίας Control Unit με τις διάφορες μονάδες
  - ○ Έλεγχος Control Unit και αύξηση πολυπλοκότητας όπου είναι ανάγκη
- ➢ Εμφάνιση bugs στις έτοιμες μονάδες
  - ○ Προσπάθεια επίλυσης τους με debugging.
- ➢ Εμφάνιση bugs στα εργαλεία CAD
  - ○ Αναζήτηση στα manuals και παράκαμψη ή διαφορετικη πορεία εαν αυτό είναι εφικτό.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η κύρια μονάδα που επιθυμούμε να υλοποιήσουμε είναι μία σύνδεση διαφόρων έτοιμων μονάδων δημιουργώντας ένα απλουστευμένο SoC. Ο λόγος που καταλήγουμε στην συγκεκριμένη απλοποιημένη μονάδα , είναι η εστίαση στο ASIC flow και όχι στο Digital Design. Θεωρούμε πως θα έχει αρκετό ενδιαφέρον αφού θα μας βοηθήσει να καταλάβουμε καλύτερα ποια είναι η λειτουργία των GPUlike Systems και των SIMD μοντέλων παραλληλισμού καθώς έχουν ευρεία χρήση όπως αναφέρθηκε πιο πάνω.

Σε περίπτωση όμως που δεν δουλέψει αυτό το εγχείρημα, αποφασίσαμε να μεταβούμε σε ένα σύστημα που θα είναι πιο απλό και θα είναι ευκολότερο να πειράξουμε τον κώδικα του όπου χρειάζεται. Βέβαια αυτόνομα αυτό δεν υπάρχει στην αγορά, παρά μόνο για να βοηθήσει άλλα συστήματα. Αλλά θα έχει αρκετό ενδιαφέρον γιατί θα δουλέψουμε και θα εξοικειωθούμε με την έννοια του pipeline stall.