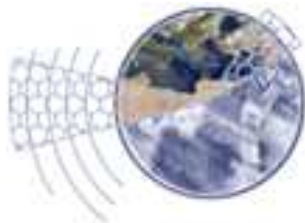


# Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης



## Σχολή Τεχνολογικών Εφαρμογών

### Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



#### Πτυχιακή εργασία

#### Τίτλος:

Παρακολούθηση και δυναμική τροποποίηση συχνότητας, κατανάλωσης ενέργειας και θερμοκρασίας, δικτύου διασύνδεσης ενσωματωμένων συστημάτων σε πραγματικό χρόνο.

**Αλέξανδρος Σπυριδάκης (ΑΜ: 860)**

**Επιβλέπων καθηγητής: Κορνάρος Γιώργος**

**Επιτροπή Αξιολόγησης:**

**Κορνάρος Γιώργος - Γραμματικάκης Μίλτος - Βλησίδης Ανδρέας**

## Abstract

Nowadays, silicon has reached its limits so it is not possible to increase the clock frequency of a processor. We have come to a point where such increases result in energy consumption inefficiencies, create thermal problems, but may even affect the stability of the whole system.

Currently, increasing the computing power can be achieved mainly with two ways.

1. Modification and optimization of the existing system architecture.
2. Increase the number of core processors (hardware) and/or parallelization of applications (software).

The first method requires high cost in time and research, so we see a trend towards more cores in a system for faster processing, or even separate processors for different functions (data processing, decoding video/music, 3D acceleration etc.)

Another issue is power consumption. At a time when thousands of mobile embedded systems are produced, consumers seek products with greater energy independence. This presents a problem where there is a need for fast, yet energy efficient appliances.

This thesis implements a parallel processing network of embedded systems, based on the Xilinx ML405 development board equipped with a Virtex-4 FPGA. The purpose is to monitor individual nodes for various attributes, such as energy consumption, temperature and clock frequency and dynamically alter the frequency of connected nodes in order to reduce the core temperature and total energy consumption.

## Σύνοψη

Το πυρίτιο έχει πλέον φτάσει στα όρια του με αποτέλεσμα να μην είναι δυνατή η αύξηση της συχνότητας χρονισμού ενός επεξεργαστή. Έχουμε έρθει σε ένα σημείο όπου τέτοιες αυξήσεις επιφέρουν μεγάλο κόστος στην κατανάλωση ενέργειας, δημιουργούν προβλήματα έκλυσης θερμότητας, αλλά μπορούν ακόμα και να επηρεάσουν τη σταθερότητα ενός συστήματος.

Η αύξηση της επεξεργαστικής ισχύς μπορεί πλέον να επιτευχθεί κυρίως με δύο τρόπους.

1. Τροποποίηση και βελτίωση της υπάρχουσας αρχιτεκτονικής.
2. Αύξηση των πυρήνων επεξεργασίας (υλισμικό/hardware) και/ή παραλληλοποίηση των εφαρμογών (λογισμικό/software).

Ο πρώτος τρόπος απαιτεί μεγάλο κόστος σε χρόνο και ερεύνα, έτσι διαπιστώνουμε μια τάση για περισσότερους πυρήνες σε ένα σύστημα με σκοπό την ταχύτερη επεξεργασία ή ακόμα και τοδιαχωρισμό επεξεργαστών για διαφορετικές λειτουργίες (επεξεργασία δεδομένων, αποκωδικοποίηση βίντεο/μουσικής, 3D acceleration κλπ).

Άλλο ένα θέμα είναι η κατανάλωση ενέργειας. Σε μια περίοδο όπου παράγονται χιλιάδες κινητά ενσωματωμένα συστήματα, ο καταναλωτής επιζητά συσκευές με μεγαλύτερη ενεργειακή αυτονομία. Έτσι, παρουσιάζεται το πρόβλημα όπου ταυτόχρονα υπάρχει η ανάγκη για γρήγορες, αλλά συνάμα ενεργειακά αποδοτικές συσκευές.

Η πτυχιακή αυτή εργασία περιλαμβάνει την υλοποίηση ενός δικτύου παράλληλης επεξεργασίας από ενσωματωμένα συστήματα βασισμένα στην πλατφόρμα ανάπτυξης ML405 της Xilinx με Virtex-4 FPGA. Σκοπός είναι να παρακολουθούμε τους επιμέρους κόμβους για διάφορα στοιχεία όπως, κατανάλωση ενέργειας, θερμοκρασία και συχνότητα χρονισμού. Με τα δεδομένα αυτά θα πρέπει δυναμικά να μπορούμε να τροποποιήσουμε τη συχνότητα χρονισμού ενός κόμβου και να μεταφέρουμε το φόρτο εργασίας σε έναν άλλο, έτσι ώστε να περιορίσουμε την αύξηση της θερμοκρασίας αλλά και της ενέργειας που καταναλώνεται.

## Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b> .....	1
1.1	Κίνητρο για τη διεξαγωγή της εργασίας .....	1
1.2	State of the Art .....	1
1.3	Σκοπός και στόχοι Εργασίας.....	2
1.4	Δομή εργασίας .....	2
<b>2</b>	<b>Γενική περιγραφή εξοπλισμού και εργαλείων ανάπτυξης</b> .....	3
2.1	Πλατφόρμα ανάπτυξης ML405 της Xilinx .....	3
2.2	Ολοκληρωμένο INA219 της Texas Instruments .....	4
2.3	Σουίτα σχεδιασμού ISE 11.5 της Xilinx .....	5
2.4	To framework Qt της Nokia.....	5
2.5	Προσομοιωτής ModelSim της MentorGraphics.....	6
2.6	Πρόγραμμα σχεδίασης κυκλωμάτων EAGLE της CadSoft .....	6
<b>3</b>	<b>Ανάλυση αρχιτεκτονικής</b> .....	7
3.1	Σύνοψη της αρχιτεκτονικής.....	7
3.2	Κεντρικός επεξεργαστής PowerPC 405 .....	8
3.3	Βοηθητικοί πυρήνες MicroBlaze .....	9
3.4	Δίαυλος PLB.....	10
3.5	Δίαυλος OPB.....	11
3.6	Δίαυλος FSL .....	12
3.7	Δίαυλος I <sup>2</sup> C ως περιφερειακό PLB .....	13
3.8	GPIO και η οθόνη υγρών κρυστάλλων .....	14
3.9	Σειριακά κανάλια επικοινωνίας Aurora .....	15
3.10	Παραγωγή και έλεγχος συχνοτήτων χρονισμού .....	16
3.11	Περιφερειακό σειριακής επικοινωνίας UartLite.....	17
<b>4</b>	<b>Δυναμική τροποποίηση συχνότητας χρονισμού</b> .....	18
4.1	Περιγραφή του προβλήματος.....	18
4.2	Αρχή λειτουργίας .....	19
4.3	Στοιχεία BUFGCTRL.....	19
4.4	Περιφερειακό ελέγχου πυρήνων MicroBlazeCoreControl .....	20
4.5	Περιφερειακό ελέγχου επεξεργαστή PowerPPCControl .....	22
4.6	Αποκοπή συχνοτήτων περιφερειακών μέσω του ClocksControl.....	25
4.7	Δίκτυο και δομή συχνοτήτων χρονισμού στο σύστημα.....	26

<b>5</b>	<b>Κατασκευή και λειτουργία αισθητηρίου ισχύος.....</b>	<b>28</b>
5.1	Η Ισχύς ως μέγεθος κατανάλωσης ενέργειας.....	28
5.2	Μέθοδοι μέτρησης ισχύος.....	28
5.3	Κύκλωμα λειτουργίας INA219 και κατασκευή αισθητηρίου.....	29
5.4	Επικοινωνία μέσω I <sup>2</sup> C.....	32
5.5	Βαθμονόμηση αισθητηρίου.....	33
<b>6</b>	<b>Ανάλυση λογισμικού κόμβου και διεπαφής.....</b>	<b>35</b>
6.1	Έναρξη λειτουργίας κόμβου.....	35
6.2	Έναρξη Xilkernel και περιφερειακών.....	35
6.3	Κύρια εφαρμογή ελέγχου και επεξεργασίας.....	36
6.4	Δίκτυο δακτυλίου.....	36
6.5	Λογισμικό διεπαφής.....	38
6.6	Ανάλυση λειτουργίας διεπαφής.....	39
<b>7</b>	<b>Μετρήσεις και συμπεράσματα.....</b>	<b>40</b>
7.1	Ταχύτητα επεξεργαστών.....	40
7.2	Κατανάλωση ενέργειας.....	40
7.3	Θερμοκρασία συστήματος.....	42
7.4	Συμπεράσματα, μελλοντική εργασία και επεκτάσεις.....	43
	<b>Παράρτημα - Βιβλιογραφία.....</b>	<b>45</b>

## Πίνακας εικόνων

<b>Εικόνα 1:</b> Κάτοψη του αναπτυξιακού ML405.....	3
<b>Εικόνα 2:</b> Σχηματικό INA219 .....	5
<b>Εικόνα 3:</b> Σχηματικό κυκλώματος πάνω σε PCB.....	6
<b>Εικόνα 4:</b> Αρχιτεκτονική κόμβου .....	8
<b>Εικόνα 5:</b> Σχέδιο διασύνδεσης περιφερειακών και επεξεργαστή μέσω PLB .....	11
<b>Εικόνα 6:</b> Σχέδιο διασύνδεσης πυρήνων MicroBlaze - BRAM μνήμης μέσω OPB .....	12
<b>Εικόνα 7:</b> Τυπική τοπολογία και διασύνδεση συσκευών σε ένα δίαυλο I <sup>2</sup> C.....	13
<b>Εικόνα 8:</b> Τρόπος σύνδεσης δύο καναλιών Aurora.....	16
<b>Εικόνα 9:</b> Σχηματικό περιφερειακού UartLite .....	18
<b>Εικόνα 10:</b> Σχηματικό λειτουργίας πολυπλέκτη 2 σε 1 .....	19
<b>Εικόνα 11:</b> Σχηματικό BUFCTRL και κατανομή κεντρικών πόρων στην FPGA.....	20
<b>Εικόνα 12:</b> Σύνδεση στοιχείων BUFCTRL στο περιφερειακό CoreControl .....	21
<b>Εικόνα 13:</b> Τα σήματα του περιφερειακού CoreControl .....	21
<b>Εικόνα 14:</b> Παράδειγμα λειτουργίας CoreControl .....	22
<b>Εικόνα 15:</b> Τα βήματα του μηχανισμού "ResumeonReset" .....	23
<b>Εικόνα 16:</b> Τα σήματα του περιφερειακού PPControl .....	24
<b>Εικόνα 17:</b> Σχηματικό λειτουργίας ClocksControl .....	26
<b>Εικόνα 18:</b> Κατανομή συχνοτήτων σε όλο το σύστημα.....	27
<b>Εικόνα 19:</b> Τρόποι μέτρησης ρεύματος. High-side (αριστερά) και Low-side (δεξιά) .....	28
<b>Εικόνα 20:</b> Περίγραμμα και σήματα του IN219. PackageSOT23-8 και SO-8 .....	29
<b>Εικόνα 21:</b> Τυπικό κύκλωμα λειτουργίας του INA219.....	30
<b>Εικόνα 22:</b> Τελικό σχηματικό αισθητηρίου .....	31
<b>Εικόνα 23:</b> Κάτοψη PCB αισθητηρίου.....	31
<b>Εικόνα 24:</b> Τελική κατασκευή και μέγεθος του αισθητηρίου .....	32
<b>Εικόνα 25:</b> Χρονικό διάγραμμα σημάτων SDL και SDA. Γραφή και ανάγνωση δεδομένων .....	32
<b>Εικόνα 26:</b> Δίκτυο τοπολογίας απλού δακτυλίου.....	37
<b>Εικόνα 27:</b> Δομή πακέτου startofframe.....	38
<b>Εικόνα 28:</b> Έλεγχος των κόμβων του δικτύου μέσω της διεπαφής.....	39
<b>Εικόνα 29:</b> Δίκτυο ενσωματωμένων συστημάτων τριών κόμβων.....	44

## Λίστα Πινάκων

<b>Πίνακας 1:</b> Διεύθυνση INA219 ως προς τη σύνδεση των σημάτων A1 και A0 .....	33
<b>Πίνακας 2:</b> Διάγραμμα MD5 επαναλήψεων - συχνότητας επεξεργαστών .....	40
<b>Πίνακας 3:</b> Διάγραμμα ισχύος - συχνότητας επεξεργαστών.....	41
<b>Πίνακας 4:</b> Διάγραμμα ισχύος - επαναλήψεων.....	41
<b>Πίνακας 5:</b> Κατανάλωση ενέργειας διαφορετικών στοιχείων και καταστάσεων .....	42
<b>Πίνακας 6:</b> Διάγραμμα θερμοκρασίας - συχνότητας αναπτυξιακού ML405 .....	42

# 1 Εισαγωγή

Σε αυτήν την πτυχιακή εργασία υλοποιούμε ένα δίκτυο από ενσωματωμένα συστήματα. Κάθε κόμβος του δικτύου αποτελείται από:

1. Την πλατφόρμα ανάπτυξης ML405 της Xilinx βασισμένη σε FPGA Virtex-4.
2. Ένα αισθητήριο κατανάλωσης ενέργειας βασισμένο στο ολοκληρωμένο INA219 της Texas Instruments.

Στόχος μας είναι η παρακολούθηση αλλά και η δυναμική τροποποίηση των παραμέτρων του εκάστοτε κόμβου σε εφαρμογές παράλληλης επεξεργασίας, για να βελτιστοποιήσουμε την ενεργειακή κατανάλωση και θερμοκρασία του συστήματος.

Πιο συγκεκριμένα, σε κάθε κόμβο έχουμε τη δυνατότητα να παρακολουθούμε τη θερμοκρασία, την κατανάλωση, το φόρτο εργασίας καθώς και τη συχνότητα χρονισμού του κάθε επεξεργαστή που είναι υλοποιημένος στην πλατφόρμα ML405. Σκοπός είναι η απρόσκοπτη αλλαγή της συχνότητας των πυρήνων επεξεργασίας, έτσι ώστε να πετύχουμε το κατάλληλο κόστος θερμοκρασίας/κατανάλωσης για συγκεκριμένα σενάρια λειτουργίας.

## 1.1 **Κίνητρο για τη διεξαγωγή της εργασίας**

Ολοένα και περισσότεροι καταναλωτές ηλεκτρονικών συσκευών επιζητούν προϊόντα με μεγαλύτερη επεξεργαστική δύναμη αλλά και καλύτερη ενεργειακή αυτονομία. Έτσι οι κατασκευαστές αναζητούν διάφορες τεχνικές για να καλύψουν τις αυξημένες ανάγκες. Μεγαλύτερη επεξεργαστική δύναμη μπορούμε να πετύχουμε με τη μέθοδο του υπερχρονισμού (overclocking)<sup>1</sup>, μέχρι πρόσφατα όμως για να γίνει κάτι τέτοιο έπρεπε να σταματήσει η λειτουργία του επεξεργαστή καθιστώντας το ασύμφορο σε κινητά ενσωματωμένα συστήματα.

Επίσης, ένα μεγάλο πρόβλημα είναι η ενεργειακή αυτονομία μίας συσκευής. Η συχνότητα χρονισμού ενός επεξεργαστή είναι ανάλογη με την κατανάλωση ενέργειας, και μαζί με την αύξηση της κατανάλωσης έχουμε αύξηση και της θερμοκρασίας. Ο συνδυασμός αύξησης κατανάλωσης και θερμοκρασίας, κάνουν το όλο σύστημα ενεργειακά μη αυτόνομο και πολλές φορές αναξιόπιστο<sup>2</sup>, άρα η λύση του στατικού υπερχρονισμού δεν είναι η βέλτιστη. Μια λύση στο προαναφερθέν πρόβλημα είναι η δυναμική τροποποίηση της συχνότητας χρονισμού ενός επεξεργαστή. Σε ένα σενάριο όπου ένας επεξεργαστής απαιτείται να τελειώσει γρήγορα μια διεργασία, είναι δυνατό να αυξηθεί η συχνότητα του επεξεργαστή και μόλις τελειώσει η διεργασία να επανέρθει στο αρχικό της επίπεδο. Με αυτήν τη μέθοδο έχουμε μεγαλύτερη κατανάλωση και θερμοκρασία του επεξεργαστή μόνο στις περιπτώσεις όπου είναι αναγκαίο κάτι τέτοιο.

Αν και αυτή η μέθοδος ήδη χρησιμοποιείται σε αρκετές εφαρμογές, δυστυχώς υπάρχει παντελής έλλειψη παραδειγμάτων και υλοποιήσεων της σε ενσωματωμένα συστήματα με FPGA (Field Programmable Gate Arrays). Το κίνητρο της πτυχιακής αυτής είναι να καλύψει αυτό το κενό και να δείξει έναν αξιόπιστο τρόπο που μπορεί να υλοποιηθεί κάτι τέτοιο σε ένα δίκτυο ενσωματωμένων συστημάτων.

## 1.2 **State of the Art**

Η δυναμική τροποποίηση της συχνότητας χρονισμού ενός επεξεργαστή είναι σχετικά καινούργια μέθοδος και μέχρι πρότινος δεν υπήρχε κάποια υλοποίηση της σε υπολογιστές. Εγκαθιδρύθηκε με τη ραγδαία ανάπτυξη των κινητών συσκευών, κυρίως για λόγους εξοικονόμησης ενέργειας και αύξησης της ενεργειακής τους αυτονομίας. Ακόμα και σήμερα πολλές συσκευές δε

διαθέτουν κάποιο μηχανισμό δυναμικής αλλαγής της συχνότητας τους και ειδικά σε desktopυπολογιστές ο στατικός υπέρ-χρονισμός ακόμα και σήμερα θεωρείται κάτι συνηθισμένο.

Η αδυναμία της συνεχής αύξησης των συχνοτήτων χρονισμού στους υπολογιστές (λόγο των ορίων του πυριτίου), ανάγκασε τους κατασκευαστές να στραφούν σε συστήματα με περισσότερους πυρήνες και σε μία τάση μείωσης της συνολικά απαιτούμενης ενέργειας. Νέες αρχιτεκτονικές έφεραν τελικά και στους υπολογιστές τέτοιους μηχανισμούς και σιγά-σιγά οι υπάρχουσες υλοποιήσεις βελτιστοποιήθηκαν με επεξεργαστές όπως οι Intel i7<sup>3</sup> και AMD Phenom II.

Σε FPGA δυστυχώς δεν υπάρχουν πολλά δείγματα για τη χρήση τέτοιων μηχανισμών, καθώς όσες υλοποιήσεις υπάρχουν περιορίζονται σε πειραματικά στάδια. Όσο μεγαλώνει όμως η χωρητικότητα μιας FPGA και υπάρχει δυνατότητα για πολυπλοκότερες αρχιτεκτονικές, τόσο θα αυξάνεται η ανάγκη για νέους τρόπους μείωσης των ενεργειακών απαιτήσεων τους.

### 1.3 Σκοπός και στόχοι Εργασίας

Οι κύριοι στόχοι της πτυχιακής εργασίας είναι:

1. Υλοποίηση και ανάδειξη ενός αξιόπιστου τρόπου δυναμικής τροποποίησης της συχνότητας χρονισμού ενός επεξεργαστή σε FPGA.
2. Κατασκευή υψηλής ακρίβειας ψηφιακού αισθητήριου για τη μέτρηση κατανάλωσης ενέργειας, βασισμένο στο ολοκληρωμένο INA219.
3. Σχεδίαση της αρχιτεκτονικής ενός πολυπύρηνου κόμβου επεξεργασίας.
4. Υλοποίηση ενός δικτύου ενσωματωμένων συστημάτων βασισμένο στο πρωτόκολλο Αυτοαυτοματισμού Xilinx.
5. Προγραμματισμός ειδικού πρωτόκολλου επικοινωνίας των κόμβων του δικτύου.
6. Προγραμματισμός διεπαφής για την παρουσίαση των μετρήσεων στο δίκτυο.
7. Μελέτη και ανάλυση της ενεργειακής κατανάλωσης και θερμοκρασίας των κόμβων.

### 1.4 Δομή εργασίας

Η συγκεκριμένη πτυχιακή εργασία χωρίζεται σε 7 κεφάλαια. Εκτός από το τρέχον κεφάλαιο της εισαγωγής περιλαμβάνει τα ακόλουθα:

- Κεφάλαιο 2<sup>ο</sup>: Γενική περιγραφή και αναφορά όλων των προγραμμάτων/εργαλείων που χρησιμοποιήσαμε καθώς και τον εξοπλισμό που χρειαστήκαμε για να εκπονήσουμε την εργασία.
- Κεφάλαιο 3<sup>ο</sup>: Πλήρης ανάλυση της αρχιτεκτονικής που υλοποιήσαμε στο αναπτυξιακό ML405, περιγράφοντας τις επιλογές και τον τρόπο λειτουργίας κάθε στοιχείου στο σύστημα.
- Κεφάλαιο 4<sup>ο</sup>: Επισκόπηση του μηχανισμού δυναμικής τροποποίησης συχνότητας χρονισμού των επεξεργαστών και λεπτομερής ανάλυση του τρόπου λειτουργίας του.
- Κεφάλαιο 5<sup>ο</sup>: Περιγραφή της μεθοδολογίας μέτρησης ισχύος και κατασκευή αισθητήριου οργάνου με τα διάφορα στάδια υλοποίησης του κυκλώματος.
- Κεφάλαιο 6<sup>ο</sup>: Ανάλυση των επιμέρους εφαρμογών που εκτελούνται στο σύστημα, καλύπτοντας όλες τις λογισμικές (software) λειτουργίες του δικτύου αλλά και τη διεπαφή του τελικού χρήστη σε τερματικό υπολογιστή.
- Κεφάλαιο 7<sup>ο</sup>: Παρουσίαση όλων των μετρήσεων και των αποτελεσμάτων, μαζί με την ανάλυση τους και τη διεξαγωγή συμπερασμάτων για τη χρήση τους σε πιθανές επεκτάσεις του θέματος.



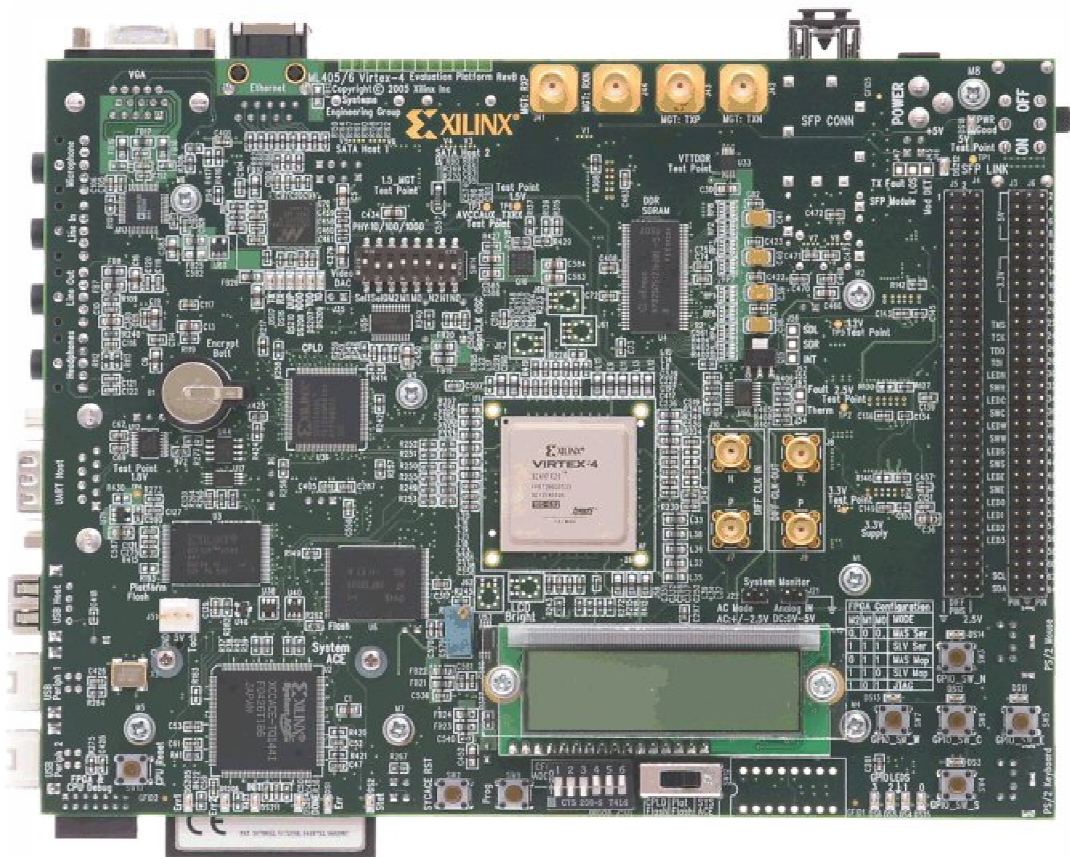
## 2 Γενική περιγραφή εξοπλισμού και εργαλείων ανάπτυξης

Στο κεφάλαιο αυτό θα περιγράψουμε αναλυτικά τα επιμέρους στοιχεία και τον εξοπλισμό που χρειάστηκε για να υλοποιηθεί η πτυχιακή. Οι ανάγκες για το εγχείρημα μας είναι πολλές, είναι λοιπόν σημαντικό να γίνει σωστή επιλογή του εξοπλισμού, τόσο για την πλήρη κάλυψη των αναγκών μας αλλά και για την εύκολη και σωστή υλοποίηση του όλου συστήματος.

### 2.1 Πλατφόρμα ανάπτυξης ML405 της Xilinx

Το αναπτυξιακό (developmentboard) ML405<sup>4</sup> είναι μια ολοκληρωμένη πλατφόρμα ανάπτυξης ενσωματωμένων συστημάτων βασισμένο στην Virtex-4 FPGA. Προσφέρει πολλαπλές δυνατότητες τόσο σε λογισμικό (software) αλλά και υλισμικό (hardware), αφού περιέχει αρκετά περιφερειακά και η FPGA που διαθέτει θα μας επιτρέψει να υλοποιήσουμε ότι παραπάνω δυνατότητες χρειαζομαστε.

Η επιλογή αυτής της πλατφόρμας ανάπτυξης δεν είναι τυχαία. Η εργασία απαιτεί αρκετά περιφερειακά καθώς και τον προγραμματισμό ειδικού hardware, όπου δεν θα ήταν δυνατό να υλοποιηθεί σε ένα απλό αναπτυξιακό με επεξεργαστή τεχνολογίας ASIC. Το ML405 συνδυάζει την απλότητα ενός επεξεργαστή με συνηθισμένα περιφερειακά και την ευέλικτη παραμετροποίηση που προσφέρει μια FPGA. Όπως θα δούμε σε μετέπειτα κεφάλαια θα χρησιμοποιήσουμε στο έπακρο τη χωρητικότητα της FPGA και τις αναπτυξιακές επιλογές που μας δίνει το ML405.



Εικόνα 1: Κάτοψη του αναπτυξιακού ML405

Αναλυτικότερα το ML405 περιέχει μεταξύ άλλων:

- Μια Virtex-4 XC4VFX20 FPGA χωρητικότητας 8.544 slices.
- 128 Megabyte DDR-SDRAM.
- Έξι παραμετροποιήσιμα κουμπιά και LED.
- 64 header pins επέκτασης για διασύνδεση άλλων περιφερειακών.
- Περιφερειακό στερεοφωνικού ήχου 16-bit στα 48kHz δειγματοληψίας.
- RS-232 σειριακή θύρα για επικοινωνία γενικής χρήσης.
- Οθόνη υγρών κρυστάλλων LCD, 16 χαρακτήρων και 2 γραμμών.
- Δίαυλο επικοινωνίας I<sup>2</sup>C για εσωτερικά και εξωτερικά περιφερειακά.
- I<sup>2</sup>C μνήμη EEPROM 4 Kilobyte.
- Έξοδο VGA για τη σύνδεση οθόνης.
- Θύρες PS/2 για πληκτρολόγιο και ποντίκι.
- 1 Megabyte μνήμη SRAM.
- Ελεγκτή SystemACE για την αποθήκευση και εκτέλεση προγραμμάτων (bitstream) από την κάρτα CompactFlash.
- Μνήμη 32 Megabyte Flash.
- Ελεγκτή Ethernet 10/100/1000.
- Ελεγκτή USB και USB host για σύνδεση περιφερειακών.
- Θύρα JTAG για τον προγραμματισμό και αποσφαλμάτωση (debug) του επεξεργαστή.
- Υποδοχείς Serial-ATA.
- Κεντρικό επεξεργαστή PowerPC ενσωματωμένο στην FPGA.
- I<sup>2</sup>C ελεγκτή θερμοκρασίας MAX6653 της Maxim<sup>5</sup>.

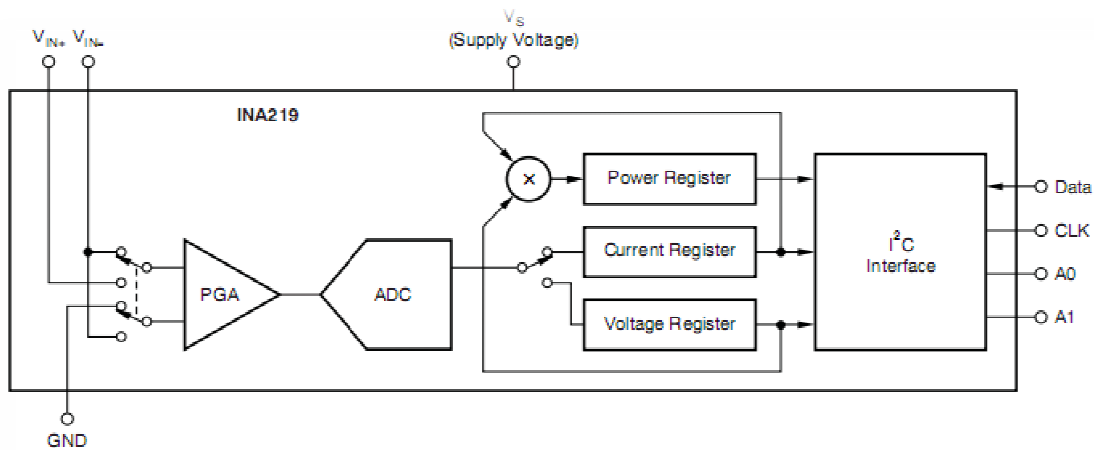
## 2.2 Ολοκληρωμένο INA219 της Texas Instruments

Το ολοκληρωμένο INA219<sup>6</sup> είναι ένα υψηλής ακρίβειας, ψηφιακό αισθητήριο κατανάλωσης ενέργειας, που χρησιμοποιεί το δίαυλο επικοινωνίας I<sup>2</sup>C. Ένας από τους στόχους της εργασίας είναι να γίνει πλήρης παρακολούθηση της ενεργειακής κατανάλωσης κάθε κόμβου και παρουσίασης των αποτελεσμάτων στον τελικό χρήστη. Για να πετύχουμε κάτι τέτοιο είναι απαραίτητο να χρησιμοποιήσουμε ένα ψηφιακό αισθητήριο έτσι ώστε να αποφύγουμε το θόρυβο αναλογικών κυκλωμάτων ή/και την περαιτέρω χρήση άλλων περιφερειακών για την ψηφιοποίηση ενός αναλογικού σήματος.

Το INA219 προσφέρει μεγάλο εύρος τιμών μέτρησης και υψηλή ακρίβεια παρά τις μικρές διαστάσεις που έχει, μόλις 9 mm<sup>2</sup> και είναι τεχνολογίας SMT (surface mount technology). Η αρχή λειτουργίας του βασίζεται στην πτώση τάσης, που εμφανίζεται σε πολύ μικρή ωμική αντίσταση (shunt resistor), που εισάγεται στο υπό μέτρηση κύκλωμα εν σειρά. Οι μετρήσεις μεταφέρονται στον επεξεργαστή με το πρωτόκολλο I<sup>2</sup>C όπου το αναπτυξιακό ML405 ήδη διαθέτει.

Οι δυνατότητες του INA219 είναι:

- Μέτρηση συσκευών με τάση από 0 - 26 Volt.
- Μετρήσεις ρεύματος, τάσης και ισχύος.
- 16 προγραμματιζόμενες I<sup>2</sup>C διευθύνσεις.
- Υψηλής ακρίβειας ADC.
- Επιλογές φιλτραρίσματος μετρήσεων.
- Καταχωρητές βαθμονόμησης.
- SMT τεχνολογία σε διαστάσεις SOT23-8 και SO-8.



Εικόνα 2: Σχηματικό INA219

### 2.3 Σουίτα σχεδιασμού ISE 11.5της Xilinx

Για την ανάπτυξη τόσο του softwareαλλά και του hardwareπάνω στο αναπτυξιακό ML405,χρησιμοποιούμε τη σουίτα σχεδιασμού ISE 11.5 της Xilinx<sup>7</sup>, η οποία παρέχει ένα μεγάλο πλήθος από εργαλεία για τις διάφορες ανάγκες του προγραμματιστή, αλλά και τα διάφορα στάδια υλοποίησης ενός συστήματος. Η σουίτα ISEπεριλαμβάνει εργαλεία για το σχεδιασμό hardware πάνω σε FPGA, ανάλυση και βελτιστοποίηση των ψηφιακών κυκλωμάτων, προγραμματισμό softwareγια ενσωματωμένα συστήματα καθώς και βιβλιοθήκες με drivers και πηγαίο κώδικα για τα διάφορα περιφερειακά που μπορούν να υλοποιηθούν σε FPGAή που υπάρχουν ήδη στο αναπτυξιακό.

Η σουίτα ISEπεριλαμβάνει:

- ISEprojectnavigatorγια τη γενική διαχείριση των σχεδίων μας.
- Το ChipScopePro όπου προσφέρει την παρακολούθηση σημάτων της FPGAκατά την εκτέλεση, επίσης περιέχει εργαλεία όπως “logicκαι systemanalyzers”.
- ΤοPlatformStudio (XPS)για τον πλήρη έλεγχο και σχεδιασμό ενσωματωμένων συστημάτων και εργαλεία για τη διαχείριση του λογισμικού (compilers, linkers)
- Το COREGenerator περιέχει βιβλιοθήκες παραγωγής ειδικών περιφερειακών και στοιχείων hardwareγια το αναπτυξιακό μας.
- PlanAhead για λεπτομερή ανάλυση τουbitstreamτης FPGA, καθώς και εργαλεία “floorplanning” για τη βέλτιστη κατανομή των πόρων.

### 2.4 Το framework Qtτης Nokia

Για την παρουσίαση των μετρήσεων στον τελικό χρήστη χρησιμοποιούμε το Qt 4.6<sup>8</sup>. Το Qtείναι μια πλατφόρμα ανάπτυξης λογισμικού (framework),βασισμένο στην αντικειμενοστραφή γλώσσα προγραμματισμού C++. Ο λόγος που επιλέξαμε το Qtγια τον προγραμματισμό της διεπαφής μας είναι η καθολικότητα (cross-platform) του συγκεκριμένου framework. Το Qtείναι συμβατό με όλα τα γνωστά λειτουργικά συστήματα υπολογιστών όπως Windows, MacOSXκαι Linux και χρησιμοποιείται επίσης σε κινητές ενσωματωμένες συσκευές με λειτουργικό WindowsCE, Symbianκαι Meego. Τέλος, το Qtείναι εντελώς ελεύθερο στην χρήση του και διατίθεται υπό τη διαδεδομένη άδεια ανοικτού κώδικα (opensource) GPLv3.

Το Qtπροσφέρει:

- Πολλές και εύκολες στη χρήση βιβλιοθήκες C++.
- Καθολικότητα κώδικα τόσο σε διαφορετικά λειτουργικά αλλά και συσκευές.
- Εργαλεία σχεδιασμού διεπαφών και υλοποίησης προγραμμάτων.
- Υψηλή απόδοση εκτέλεσης με χαμηλό επεξεργαστικό κόστος.
- Ελεύθερη χρήση υπό την άδειαGPLv3.

## 2.5 ΠροσομοιωτήςModelSim της MentorGraphics

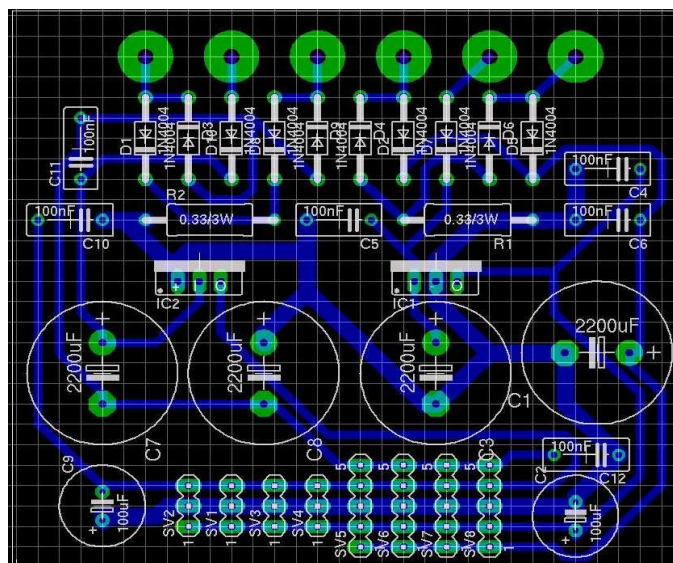
Ο προγραμματισμός hardware σε VHDLγια FPGAείναι μια χρονοβόρα διαδικασία όπου διαφέρει αρκετά από τις «ευκολίες» του software προγραμματισμού. Η σωστή ανάπτυξη και ανάλυση του hardwareαπαιτεί έναν προσομοιωτή (simulator) για εξοικονόμηση χρόνου και πόρων. Με τη μέθοδο της προσομοίωσης μπορούμε να θέσουμε τις συνθήκες που θέλουμε εύκολα και να ελέγξουμε το σύστημα μας με τον ταχύτερο δυνατό τρόπο. Για αυτό το λόγο, για την ανάπτυξη της πτυχιακής εργασίας χρησιμοποιήσαμε το ModelSimSE 6.4c.

Το ModelSim SE περιλαμβάνει:

- Υψηλής ταχύτητα μηχανή προσομοίωσης.
- Υποστήριξη VHDLκαι Verilog.
- Υποστήριξη SystemVerilogκαι SystemC.
- Δια-λειτουργικότητα σε περιβάλλον Windows, Solariskαι Linux.

## 2.6 Πρόγραμμα σχεδίασης κυκλωμάτωνEAGLE της CadSoft

Το EAGLE (EasilyApplicableGraphicalLayoutEditor) χρησιμοποιήθηκε για την υλοποίηση του αισθητηρίου κατανάλωσης ενέργειας. Είναι ένα πρόγραμμα σχεδίασης ηλεκτρονικών κυκλωμάτων και σχηματικών που χρησιμοποιείται για την παραγωγή PCB (printedcircuitboard), αλλά και την παρουσίαση σχηματικών και κυκλωμάτων.Με το EAGLEέγινε εφικτή η βελτίωση του χώρου της πλακέτας του αισθητηρίου καθώς και η αυτοματοποίηση της δρομολόγησης των χαλκό-διαδρόμων πάνω στηνPCB.



Εικόνα 3: Σχηματικό κυκλώματος πάνω σε PCB

### 3 Ανάλυση αρχιτεκτονικής

Σε αυτό το κεφάλαιο θα αναλύσουμε την αρχιτεκτονική που υλοποιήσαμε στο αναπτυξιακό ML405 για τις ανάγκες της εργασίας. Το ML405 διαθέτει πολλά περιφερειακά και μια Virtex-4 FPGA που περιλαμβάνει περίπου 8.500 slices<sup>9</sup>, για τη σχεδίαση hardware, καθώς και έναν επεξεργαστή PowerPC. Όπως θα δούμε παρακάτω, η χρήση της FPGA φτάνει κοντά στο 100% της χωρητικότητας της, κάτι που σημαίνει ότι η αρχιτεκτονική που σχεδιάσαμε είναι αρκετά πολύπλοκη αλλά και βελτιστοποιημένη για το διαθέσιμο χώρο, χωρίς να έχουμε κόστος στην απόδοση ή τη λειτουργικότητα του συστήματος.

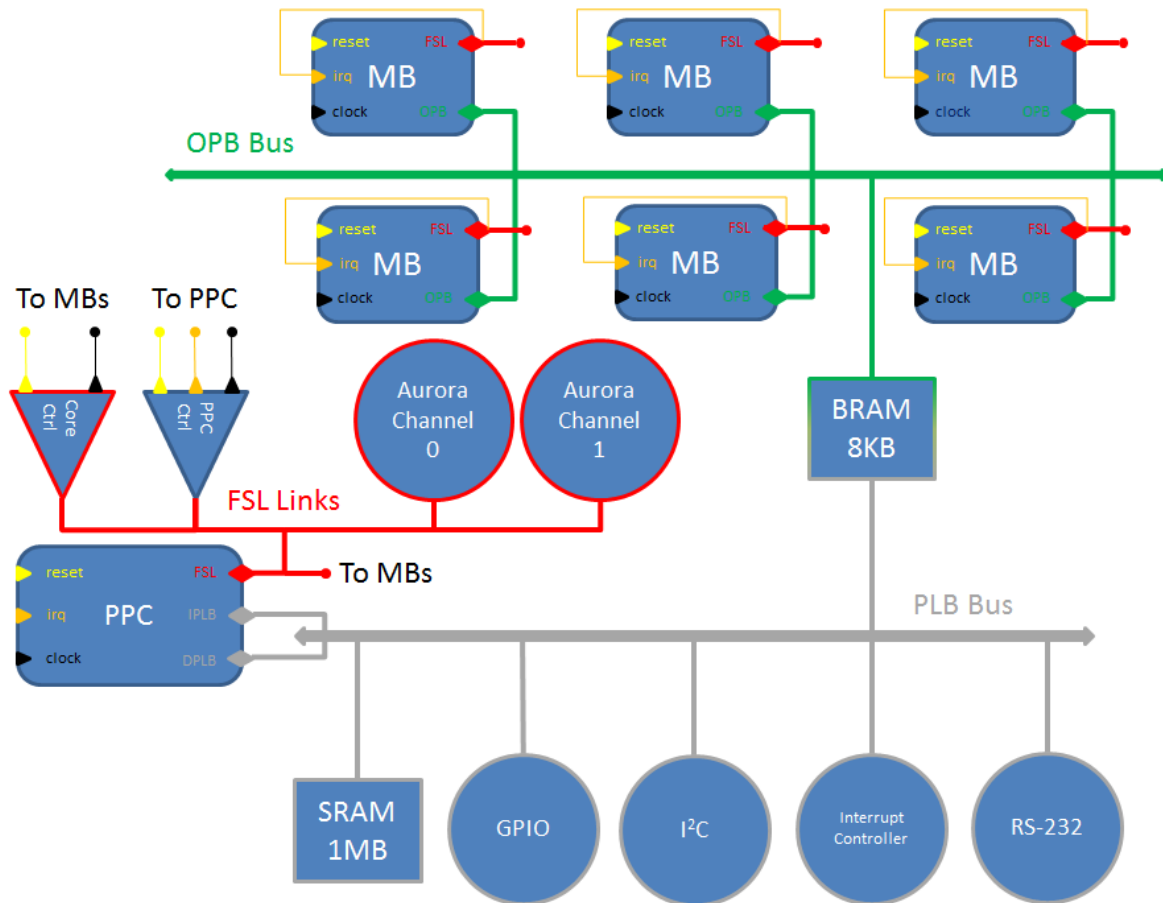
#### 3.1 Σύνοψη της αρχιτεκτονικής

Κάθε κόμβος του δικτύου αποτελείται κυρίως από τα επεξεργαστικά στοιχεία (processing elements) και τα επιμέρους περιφερειακά για τη μεταφορά δεδομένων, τη διασύνδεση των κόμβων, την επικοινωνία με ένα τερματικό υπολογιστή και φυσικά τα αισθητήρια για τις κατάλληλες μετρήσεις.

Πιο συγκεκριμένα κάθε κόμβος αποτελείται από:

- A. Επεξεργαστικά στοιχεία.
  1. Έναν κύριο επεξεργαστή PowerPC405 ενσωματωμένο στην FPGA.
  2. Έξι βοηθητικούς πυρήνες MicroBlaze.
- B. Δίαυλοι επικοινωνίας
  1. PLB (processor local bus) για τον PowerPC και τα περιφερειακά του.
  2. OPB (on-chip peripheral bus) για τη διασύνδεση των έξι MicroBlaze.
  3. Πολλαπλά αμφίδρομα κανάλια FSL (fast simple link) για την επικοινωνία μεταξύ των επεξεργαστών PowerPC–MicroBlaze και του PowerPC με τα περιφερειακά ελέγχου των συχνοτήτων.
  4. Δύο αμφίδρομα σειριακά κανάλια επικοινωνίας των κόμβων, βασισμένα στο πρωτόκολλο Aurora.
  5. I<sup>2</sup>C για την αποστολή και λήψη δεδομένων στα αισθητήρια θερμοκρασίας και κατανάλωσης ενέργειας.
- C. Στοιχεία μνήμης
  1. 1 Megabyte SRAM για τις ανάγκες του PowerPC.
  2. 16 Kilobyte BRAM για κάθε MicroBlaze πυρήνα.
  3. 8 Kilobyte BRAM κοινής μνήμης (shared memory) μεταξύ PowerPC και MicroBlaze.
- D. Περιφερειακά PowerPC
  1. Σειριακή θύρα RS-232 για την επικοινωνία του κόμβου με κάποιο τερματικό υπολογιστή.
  2. Σήματα GPIO (general purpose input-output) για τη λειτουργία της οθόνης υγρών κρυστάλλων (LCD screen).
  3. Κεντρικό ελεγκτή σημάτων interrupt.
  4. Περιφερειακό ελέγχου της συχνότητας χρονισμού του επεξεργαστή PowerPC (PPC control).
  5. Περιφερειακό ελέγχου της συχνότητας χρονισμού των πυρήνων MicroBlaze (Core control).
  6. Περιφερειακό αποκοπής συχνοτήτων για την εξοικονόμηση ενέργειας. (Clock control).





Εικόνα 4: Αρχιτεκτονική κόμβου

### 3.2 Κεντρικός επεξεργαστής PowerPC 405

Ο PowerPC 405 είναι ένας 32-bit, big-endian, RISC επεξεργαστής και η υλοποίηση του είναι βασισμένη στην ομώνυμη αρχιτεκτονική της IBM (*Performance Optimization with Enhanced RISC - Performance Computing*) για ενσωματωμένα συστήματα<sup>10</sup>. Ο PowerPC είναι ενσωματωμένος στην Virtex-4 FPGA του ML405 μαζί με "cache" μνήμη 32 Kilobyte (μοιρασμένη σε μνήμη instruction και data) και δεν επιφέρει κανένα κόστος χωρητικότητας στα προγραμματιζόμενα slicest της FPGA.

Μεταξύ άλλων διαθέτει:

- Δύο ζεύγη διαύλων PLB.
- Τρία προγραμματιζόμενα timers (programmable interval timer, fixed interval timer και watchdog timer).
- MMU (memory management unit) για τη διαχείριση μνήμης.
- Ελεγκτή JTAG για αποσφαλμάτωση (debugging).
- APU (auxiliary processor unit) για τη σύνδεση FSL συσκευών ή του FPU (floating point unit).
- Ελεγκτή interrupt.
- Πέντε επιπέδων pipeline.

Στην αρχιτεκτονική μας έχει τον κύριο επεξεργαστικό ρόλο και είναι υπεύθυνος για να:

- Ελέγχει όλη την κίνηση δεδομένων στο δίκτυο.
- Έχει επικοινωνία και τον πλήρη έλεγχο των βοηθητικών πυρήνων MicroBlaze.
- Ρυθμίζει και παίρνει μετρήσεις από τα αισθητήρια.
- Ανανεώνει και στέλνει τις μετρήσεις στην LCDοθόνη και τη σειριακή θύρα.
- Αναλαμβάνει μέρος του φόρτου εργασίας στην εφαρμογή παράλληλης επεξεργασίας που εκτελείται στο δίκτυο.

Όπως διαπιστώνουμε, το έργο του PowerPCείναι μεγάλο και πολύπλοκο. Για να γίνει δυνατή η εκτέλεση τόσων πολλών και ταυτόχρονων διεργασιών, ο προγραμματισμός του επεξεργαστή έγινε υπό το λειτουργικό πυρήνα (kernel)Xilkernelv4της Xilinx,όπου προσφέρει εργαλεία παράλληλης επεξεργασίας όπως threads, semaphores, softwaretimers, threadscheduling, κτλ.

Μια πολλή σημαντική λειτουργία του PowerPCείναι η χρήση του ελεγκτή APU(auxiliaryprocessorunit). Με τη χρήση του APUμπορούμε να συνδέσουμε στον επεξεργαστή μέχρι και 32 αμφίδρομους δεσμούς FSL (fastsimplexlink). Το πρωτόκολλο FSLμας επιτρέπει την άμεση επικοινωνία με τους βοηθητικούς πυρήνες MicroBlaze, καθώς και ειδικά σχεδιασμένα στοιχεία που έχουμε υλοποιήσει στην FPGA όπου ελέγχουν τη συχνότητα χρονισμού του κάθε επεξεργαστή.

Τέλος, υψίστης σημασίας είναι η χρήση των interruptsγια να αποφευχθεί το επεξεργαστικό κόστος της λειτουργίας «*rolling*». Σε λειτουργία *rolling*, ο επεξεργαστής πρέπει να ελέγχει κάθε συγκεκριμένο χρονικό διάστημα για εισερχόμενα δεδομένα ή άλλα γεγονότα που αφορούν το σύστημα. Ο τρόπος αυτός προσθέτει ένα τεράστιο βάρος στον επεξεργαστή, με αποτέλεσμα να χάνουμε πολύτιμη επεξεργαστική δύναμη σε διεργασίες που δεν την έχουν άμεση ανάγκη. Αντίθετα, με τη χρήση interruptso επεξεργαστής σταματάει την κανονική του λειτουργία και τρέχει μια ειδικευμένη συνάρτηση όπου εκτελεί τον κατάλληλο κώδικα για το συγκεκριμένο γεγονός που προκάλεσε το interrupt<sup>11</sup>. Μετά το τέλος της συνάρτησης, ο επεξεργαστής συνεχίζει κανονικά τη λειτουργία του και δεν χρειάζεται να γίνεται συνεχής παρακολούθηση των κρίσιμων γεγονότων στο σύστημα.

Ο PowerPCδέχεται συνολικά 10 interrupts (με σειρά προτεραιότητας):

1. Έξι interruptγια κάθε βοηθητικό πυρήνα MicroBlazepου στέλνει δεδομένα προς τον PowerPC.
2. Ειδικευμένοinterruptγια την αλλαγή συχνότητας χρονισμού του PowerPC.
3. Interruptεισερχόμενων δεδομένων στη σειριακή θύρα RS-232.
4. Δύο interruptγια κάθε αμφίδρομο σειριακό κανάλι επικοινωνίας Aurora.

### 3.3 Βοηθητικοί πυρήνες MicroBlaze

ΟMicroBlazeείναι ένας 32-bit, big-endian, RISCεπεξεργαστής για FPGAτης Xilinx<sup>12</sup>. Σε αντίθεση με τον PowerPCο MicroBlazeείναι “*softprocessorcore*”που σημαίνει ότι δεν προϋπάρχει στην FPGA, αλλά υλοποιείται σε αυτήν σαν ψηφιακό κύκλωμα καταναλώνοντας slices. Λόγο της απλούστερης αρχιτεκτονικής αλλά και της softcore φύσης του, δεν έχει την ίδια επεξεργαστική ισχύ όπως ο PowerPCκαι δεν μπορεί να φτάσει τόσο υψηλές συχνότητες. Η αδυναμία του στις υψηλές συχνότητες προέρχεται από την αυξημένη καθυστέρηση διάδοσης (*propagationdelay*) που προκύπτει σε μια FPGAσε σχέση με μια ASIC(*application-specificintegratedcircuit*) υλοποίηση ενός κανονικού επεξεργαστή. Σε μετέπειτα κεφάλαιο θα ακολουθήσει λεπτομερής ανάλυση της κατανάλωσης/απόδοσης του MicroBlazeσε σύγκριση με τον PowerPC.

Συνοπτικά ένας πυρήνας MicroBlaze περιλαμβάνει:

- Τριάντα δύο 32-bit καταχωρητές γενικής χρήσης.
- OPB ή PLB δίαυλο επικοινωνίας.
- Δίαυλο LMB για τη χρήση μνήμης.
- Τριών επιπέδων pipeline.
- Υποστήριξη FSL.
- Ενσωματωμένο πολλαπλασιαστή (hardware multiplier).

Η απλούστερη αρχιτεκτονική του MicroBlaze τον κάνει ιδανικό για το βοηθητικό ρόλο που έχει επιλεγεί. Κάθε πυρήνας καταναλώνει περίπου 1.200 slices από την FPGA και 8 RAMB16 blocks για τα 16 kilobyte μνήμης που χρειαζόμαστε. Η χωρητικότητα της FPGA του ML405 μας περιορίζει συνολικά στους έξι πυρήνες MicroBlaze. Υπάρχει δυνατότητα υλοποίησης του επεξεργαστή με μειωμένο χωρητικό κόστος, με αντάλλαγμα ένα ποσοστό της επεξεργαστικής του ισχύς αλλά και χαμηλότερη ενεργειακή κατανάλωση. Στην αρχιτεκτονική που σχεδιάσαμε δεν επιλέξαμε κάτι τέτοιο αφού περισσότεροι πυρήνες MicroBlaze θα σήμαινε περισσότερα RAMB16 blocks όπου η XC4VFX20FPGA δεν διαθέτει.

Ο ρόλος του κάθε πυρήνα είναι ένας, να εκτελεί ένα αλγόριθμο παράλληλης επεξεργασίας και να ενημερώνει τον PowerPC με τα αποτελέσματα. Μέσω FSL ο κάθε πυρήνας μπορεί να λαμβάνει και να στέλνει δεδομένα στον PowerPC. Επίσης, όλοι οι πυρήνες έχουν πρόσβαση σε 8 kilobyte κοινής μνήμης μέσω του διαύλου OPB που μοιράζονται. Η συγκεκριμένη μνήμη είναι προσβάσιμη και από τον PowerPC μέσω του διαύλου PLB. Με αυτότο μηχανισμό, ο κάθε πυρήνας μπορεί να λειτουργήσει εντελώς αυτόνομα (εφόσον επιτραπεί από τον PowerPC) ή σε συνεργασία με τους υπόλοιπους πυρήνες μέσω της κοινής μνήμης.

### 3.4 Δίαυλος PLB

OPLB (processor local bus) είναι ένας δίαυλος επικοινωνίας μεταξύ επεξεργαστών και περιφερειακών της IBM, μέρος της αρχιτεκτονικής «CoreConnect»<sup>13</sup>. Υποστηρίζει μέχρι και 128-bit αρχιτεκτονικές καθώς και πολλαπλές συσκευές σε λειτουργία «master» ή/και «slave» πάνω στον ίδιο δίαυλο.

Τα χαρακτηριστικά του είναι<sup>14</sup>:

- Υποστήριξη διαιτησίας (arbitration) μεταξύ master συσκευών.
- Λειτουργία 128, 64 ή 32 bit για master ή slave συσκευές.
- Pipelining στο επίπεδο διευθύνσεων.
- Αλγοριθμική διαιτησία σταθερής ή «round robin» προτεραιότητας.
- Επιλογή βελτιστοποίησης (χώρου και ταχύτητας) για τοπολογίες point-to-point.
- Watchdog timer για τη σωστή λήξη κίνησης σε περίπτωση σφάλματος.
- Παραγωγή ενιαίων σημάτων reset για όλες της συσκευές στο δίαυλο.

Ο δίαυλος PLB είναι πλέον η καθιερωμένη λύση διασύνδεσης περιφερειακών και επεξεργαστών σε FPGA της Xilinx. Αντικατέστησε τον προκάτοχο του, OPB, με περισσότερα χαρακτηριστικά αλλά με αντάλλαγμα το μεγαλύτερο χωρητικό κόστος. Σχεδόν όλα τα υπάρχοντα OPB περιφερειακά της Xilinx έχουν ενημερωθεί για να είναι συμβατά με το PLB πρωτόκολλο και τις αυξημένες δυνατότητες που παρέχει.

Στην αρχιτεκτονική μας, η σύνδεση των περισσότερων περιφερειακών γίνεται μέσω PLB. Ο PowerPC έχει δύο ζεύγη PLB όπου το ένα είναι ειδικευμένο για point-to-point τοπολογίες, επειδή

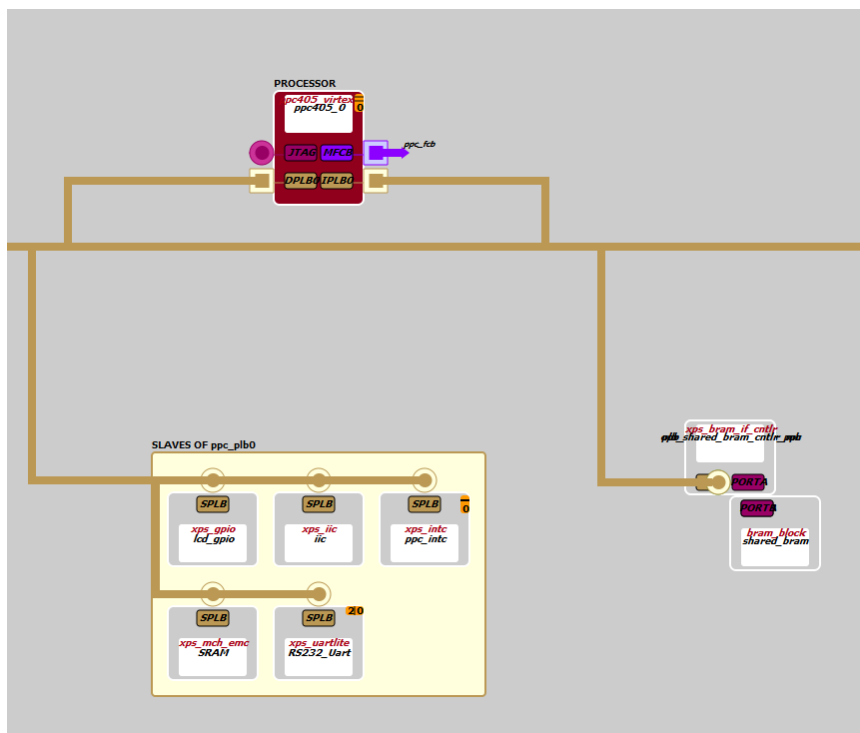


όμως έχουμε περισσότερα από ένα περιφερειακά, χρησιμοποιούμε το ζεύγος IPLB0 και DPLB0 (κάθε ζεύγος είναι για σύνδεση λειτουργίας instruction και data).

Τα περιφερειακά που συνδέονται απευθείας στον PowerPC μέσω του PLB είναι έξι:

1. Η μνήμη SRAM.
2. Η σειριακή θύρα RS-232.
3. Ο διάυλος επικοινωνίας I<sup>2</sup>C.
4. Οι κοινή BRAM μνήμη μεταξύ PowerPC και MicroBlaze.
5. Τα σήματα GPIO για τη λειτουργία της LCD οθόνης.
6. Ο κεντρικός ελεγκτής interrupt.

Η επιλογή μας να μη χρησιμοποιήσουμε το δεύτερο ζεύγος PLB του PowerPC είναι καθαρά χωρητικής φύσης. Ο διάυλος PLB παρά την ταχύτητα επικοινωνίας και τα χαρακτηριστικά που προσφέρει, χρειάζεται ένα μεγάλο μέρος από τα slices της FPGA. Ενδεικτικά αναφέρουμε ότι ένας PLB διάυλος με έξι slave περιφερειακά και δύο master συσκευές (PowerPC instruction και data) χρειάζεται συνολικά 300 slices, τα διπλάσια από ένα αντίστοιχο διάυλο OPB.



Εικόνα 5: Σχέδιο διασύνδεσης περιφερειακών και επεξεργαστή μέσω PLB

### 3.5 Δίαυλος OPB

Επίσης μέρος της αρχιτεκτονικής CoreConnect της IBM, ο διάυλος OPB (on-chip peripheral bus)<sup>15</sup> προορίζεται για τη διασύνδεση συσκευών 32-bit χαμηλότερης ταχύτητας. Όπως και ο PLB, διαθέτει υποστήριξη για πολλαπλές συσκευές σε λειτουργία master ή/και slave.

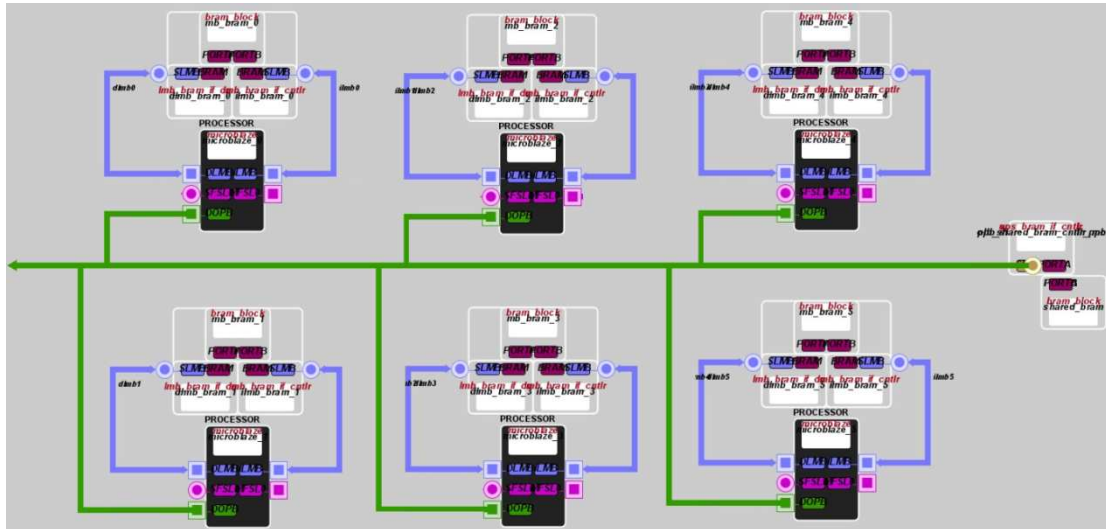
Τα χαρακτηριστικά του είναι:

- Διαιτησία (arbitration) μεταξύ master συσκευών.
- Επιλογή σταθερής ή δυναμικής διαιτησίας.
- Υποστήριξη 8, 16 και 32 bit συσκευών.
- Μέχρι 16 master και 16 slave συσκευές.

- Watchdog timer
- Ενιαία παραγωγή σημάτων reset.

Ο ρόλος του διαύλου OPB στην αρχιτεκτονική μας είναι η σύνδεση της κοινής BRAM μνήμης, μεταξύ των έξι βοηθητικών πυρήνων MicroBlaze.Αν και το OPB βρίσκεται σε περίοδο αντικατάστασής του από τηXilinx, αποφύγαμε τη χρήση του διαύλου PLBγια δύο λόγους.

1. Ο δίαυλος PLBαπαιτεί περισσότερα slices, ειδικά όσο πληθαίνουν οι masterσυσκευές (έξι στο παράδειγμά μας).
2. Λόγο της απλούστερης αρχιτεκτονικής του OPB,είναι εφικτό να γίνει χρήση υψηλότερων συχνοτήτων για τους πυρήνες MicroBlaze.



Εικόνα 6: Σχέδιο διασύνδεσης πυρήνων MicroBlaze - BRAM μνήμης μέσω OPB

### 3.6 Δίαυλος FSL

Ο δίαυλος FSL (fastsimplexlink)<sup>16</sup> είναι ένα 32-bit, αμφίδρομο, point-to-point κανάλι επικοινωνίας της Xilinx, για τη γρήγορη μεταφορά δεδομένων μεταξύ δύο στοιχείων σε FPGA (συνήθως έναν επεξεργαστή και ένα περιφερειακό). Χρησιμοποιείται ευρέως για τη σύνδεση συνεπεξεργαστών (co-processor) συμπληρώνοντας τη λειτουργία του συνδεδεμένου επεξεργαστή ή για την άμεση επικοινωνία με κάποιο ειδικά σχεδιασμένο περιφερειακό.

Χαρακτηριστικά του δίαυλου FSL:

- Υλοποίηση απλού και αμφίδρομου δεσμού επικοινωνίας τοπολογίας point-to-point.
- Λόγο της απλής τοπολογίας δε χρειάζεται μηχανισμούς διαιτησίας.
- Γρήγορη μεταφορά δεδομένων μεταξύ συνδεδεμένων συσκευών master/slave.
- Επιπλέον παροχή ενός biteλέγχου για το διαχωρισμό μεταξύ δεδομένων και δεδομένων ελέγχου.
- Επικοινωνία βασισμένη σε FIFOβάθους μίας λέξης (32-bit word) έως 8 kilobyte.
- Επιλογή υλοποίησης FIFO σε LUTRAM ή BRAM.
- Υποστήριξη συσκευών master/slave διαφορετικής συχνότητας.

Οι δύο τύποι επεξεργαστών που χρησιμοποιεί η αρχιτεκτονική μας (PowerPC και MicroBlaze) διαθέτουν υποστήριξη πολλαπλών FSL. Ο MicroBlaze διαθέτει εγγενώς 8 κανάλια FSL και ο PowerPC μέσω της χρήσης του FCB (fabricco-processor), καθώς και του block «FCBtoFSLbridge» υποστηρίζει μέχρι και 32 κανάλια FSL.

Η χρήση των διαύλων FSL είναι κρίσιμη για τη λειτουργικότητα της αρχιτεκτονικής που έχουμε σχεδιάσει. Το χωρητικό κόστος τους είναι πολύ χαμηλό (μόλις 53 slices) για την ταχύτητα μεταφοράς που διαθέτουν, καθώς και είναι αρκετά παραμετροποιήσιμοι για τη διασύνδεση εντελώς διαφορετικών συσκευών. Η κατασκευή των ειδικευμένων περιφερειακών μας (PPCControl, CoreControl, Aurora) με το πρωτόκολλο PLB ήταν πρακτικά ανέφικτη για δύο κύριους λόγους.

1. Χωρητικά, οι συσκευές με υποστήριξη PLB καταναλώνουν περισσότερα slices και κάθε επιπλέον συσκευή συνδεδεμένη στο δίαυλο μεγαλώνει το μέγεθος του.
2. Το πρωτόκολλο PLB απαιτεί συσκευές με συχνότητα χρονισμού πολλαπλάσια από αυτήν του διαύλου (αναλογία συχνότητας 1:N).

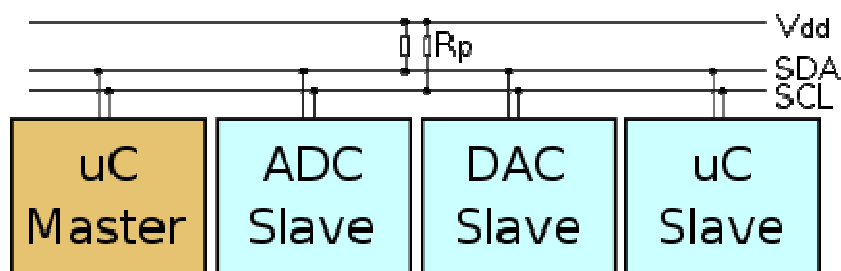
Ο δεύτερος περιορισμός δεν υπάρχει στην υλοποίηση FSL, αφού κάθε συσκευή master/slave μπορεί να έχει εντελώς διαφορετική συχνότητα χρονισμού, κάτι τέτοιο είναι επιτακτικό για τις ανάγκες της αρχιτεκτονικής.

Χρήση των δέκα διαύλων FSL:

- Μεταξύ PowerPC και κάθε πυρήνα MicroBlaze.
- Υλοποίηση των δύο σειριακών καναλιών επικοινωνίας Aurora.
- Δυναμική τροποποίηση συχνότητας χρονισμού και έλεγχος των επεξεργαστών (PPCControl και CoreControl).

### 3.7 Δίαυλος I<sup>2</sup>C ως περιφερειακό PLB

Ο I<sup>2</sup>C (inter-integrated circuit) είναι ένας δίαυλος διασύνδεσης περιφερειακών χαμηλής ταχύτητας σε ενσωματωμένα συστήματα<sup>17</sup>. Εφευρέθηκε από την Philips το 1976 και χρησιμοποιείται ευρέως λόγω των χαμηλών απαιτήσεων υλοποίησής του και των πολλών συσκευών που τον χρησιμοποιούν. Η υλοποίηση του I<sup>2</sup>C είναι πολύ απλή αφού χρειάζεται μόλις δύο σήματα για τη σύνδεση περιφερειακών συσκευών. Συνήθως η λειτουργία του διαύλου βασίζεται σε τοπολογία master/slave, έχοντας δυνατότητες επέκτασης σε εφαρμογές με πολλαπλές master συσκευές.



Εικόνα 7: Τυπική τοπολογία και διασύνδεση συσκευών σε ένα δίαυλο I<sup>2</sup>C

Τα δύο σήματα επικοινωνίας αποτελούνται από:

- SCL (serial clock) όπου συγχρονίζει την επικοινωνία και ελέγχεται από τη master συσκευή.
- SDA (serial data) όπου γίνεται όλη η μεταφορά δεδομένων σειριακά από τις συνδεδεμένες συσκευές.

Η υλοποίηση του διαύλου I<sup>2</sup>C έρχεται από τη Xilinx ως ένα περιφερειακό PLB, αφού οι επεξεργαστές PowerPC και MicroBlaze δεν υποστηρίζουν εγγενώς το πρωτόκολλο I<sup>2</sup>C. Το περιφερειακό I<sup>2</sup>C της Xilinx μεταξύ άλλων περιλαμβάνει<sup>18</sup>:

- Λειτουργία του περιφερειακού ως master ή slave συσκευή.
- Υποστήριξη πολλαπλών master συσκευών.
- Επιλογή ταχύτητας στα 100KHz (standard mode) ή 400KHz (fast mode).
- 7 ή 10 bit διευθυνσιοδότηση.
- FIFO μετάδοσης, βάθους 16 byte.
- Προαιρετική χρήση σημάτων εξόδου πλάτους 1 - 8 bit (generalpurposeoutput).
- Ανίχνευση και παραγωγή σημάτων ελέγχου START/STOP και repeatedSTART.

Σε κάθε κόμβο του δικτύου υπάρχουν τρεις συσκευές όπου η χρήση του I<sup>2</sup>C είναι επιτακτική:

1. Ο ελεγκτής θερμοκρασίας MAX6653 της Maxim.
2. Η μνήμη EEPROM.
3. Το εξωτερικό αισθητήριο κατανάλωσης ενέργειας (βασισμένο στο ολοκληρωμένο INA219) που κατασκευάσαμε.

### 3.8 GPIO και η οθόνη υγρών κρυστάλλων

Η οθόνη υγρών κρυστάλλων χαρακτήρων (characterLCD) του ML405 είναι συμβατή με όλες τις οθόνες τύπου HD44780<sup>19</sup> και παρέχει όλες τις καθιερωμένες εντολές για τον έλεγχο και την εμφάνιση ενός περιορισμένου σετ χαρακτήρων καθώς και 8 θέσεων μνήμης για χαρακτήρες που μπορούν να δημιουργηθούν κατά το χρόνο εκτέλεσης. Η συγκεκριμένη οθόνη είναι ύψους 2 γραμμών και πλάτους 16 χαρακτήρων.

Η οθόνη χρησιμοποιεί συνολικά 11 pin/bit (σήματα) για τη λειτουργία της, 8 σήματα δεδομένων και 3 ελέγχου. Στην περίπτωση του ML405, για εξοικονόμηση πόρων έχουμε μόνο 4 σήματα δεδομένων, άρα για κάθε λειτουργία απαιτούνται δύο εντολές. Σε αυτόν τον τύπο λειτουργίας η οθόνη (αφού αρχικοποιηθεί κατάλληλα) περιμένει δύο εντολές πριν την εκτέλεση κάποιας ενέργειας. Η πρώτη εντολή έχει τα 4 πρώτα bit δεδομένων (upper nibble) και μετά περιμένει άλλα 4 (lower nibble) πριν προχωρήσει στην εντολή που δώσαμε, έτσι ουσιαστικά επιτυγχάνουμε την ίδια λειτουργία με λιγότερα σήματα δεδομένων, με αντίτιμο όμως διπλάσιο χρόνο εκτέλεσης.

Τα 3 bit ελέγχου είναι υπεύθυνα για τις εντολές που παίρνει η οθόνη. Το «E» (enable) bit ορίζει πότε θα δεχθεί η οθόνη κάποια εντολή. Θέτοντας αυτό το bit στην τιμή 1 και αμέσως μετά (μετά από κάποια καθυστέρηση) στην τιμή 0, αναγκάζει την οθόνη να εκτελέσει τις αλλαγές που έχουμε στα υπόλοιπα bit. Έτσι, ουσιαστικά για κάθε εντολή που στέλνουμε, πρέπει να μεταβάλλουμε την κατάσταση αυτού του bit (δύο φορές για κάθε nibble στην περίπτωση λειτουργίας με 4 bit). Το «RS» bit ορίζει αν η εντολή που θα εκτελεστεί θα είναι η εμφάνιση κάποιου χαρακτήρα, ή εάν πρόκειται για εσωτερική εντολή της οθόνης (π.χ. καθαρισμός της οθόνης ή εμφάνιση του κέρσορα). Τέλος, το «R/W» bit επιλέγει αν θα γράψουμε απευθείας στην οθόνη ή στην εσωτερική μνήμη που έχει (8 θέσεις όπου μπορούμε να βάλουμε δικούς μας χαρακτήρες). Ο έλεγχος και η λειτουργία της οθόνης γίνεται με τη μέθοδο «bitbanging» πάνω στα σήματα ελέγχου και δεδομένων, μέσω ενός περιφερειακού GPIO (generalpurposeinputoutput), που είναι συνδεδεμένο στο δίαυλο PLB. Κατά αυτόν τον τρόπο, ο επεξεργαστής PowerPC μπορεί να στείλει ειδικές εντολές στο περιφερειακό για την αλλαγή κατάστασης των 11 bit που ελέγχουν την οθόνη, ή να ενημερωθεί για την κατάσταση τους και να πάρει πληροφορίες από αυτήν.

Τα χαρακτηριστικά του περιφερειακού GPIO είναι<sup>20</sup>:

- Επιλογή χρήσης ενός ή δύο καναλιών GPIO.
- Παραμετροποίηση του πλάτους για κάθε κανάλι από 1 - 32 bit.
- Προγραμματισμός κάθε bit για λειτουργία εισόδου ή εξόδου.

- Προαιρετική παραγωγή σήματος interrupt.

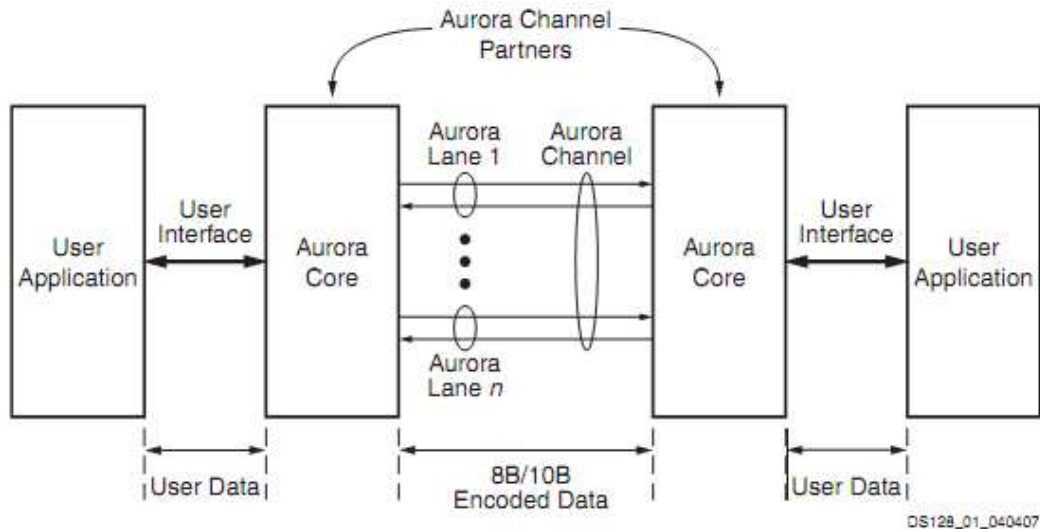
### 3.9 Σειριακά κανάλια επικοινωνίας Aurora

Εκτός από τους εσωτερικούς μηχανισμούς ενδοεπικοινωνίας και σύνδεσης των επεξεργαστών/περιφερειακών στο αναπτυξιακό ML405, χρειαζόμαστε μία πολύ γρήγορα αλλά χωρητικά αποδοτική μέθοδο εξωτερικής επικοινωνίας μεταξύ των κόμβων του δικτύου. Μία προφανής λύση θα ήταν η χρήση του ελεγκτή Ethernet που προσφέρει το αναπτυξιακό ML405. Με αυτόν τον τρόπο θα είχαμε μία καθιερωμένη και σχετικά γρήγορη σύνδεση των κόμβων, δυστυχώς όμως κάτι τέτοιο είναι χωρητικά αδύνατο στην υπάρχουσα αρχιτεκτονική. Ο ελεγκτής Ethernet απαιτεί αρκετά μεγάλο χώρο στην FPGA και σαν αντάλλαγμα θα έπρεπε να αφαιρέσουμε πολύτιμα στοιχεία της αρχιτεκτονικής (λιγότερα επεξεργαστικά στοιχεία, περιφερειακά, μνήμη, κτλ). Επίσης η χρήση του πρωτοκόλλου TCP/IP θα ήταν πολύπλοκη, για τις σχετικά απλές ανάγκες επικοινωνίας του δικτύου. Για τους παραπάνω λόγους στραφήκαμε σε μια πιο ειδικευμένη λύση, που μπορεί να μην είναι τόσο καθιερωμένη όπως το Ethernet, αλλά σχεδιασμένη με τέτοιο τρόπο ώστε να καλύψει πλήρως της ανάγκες του δικτύου, με το μικρότερο χωρητικό κόστος και τη μέγιστη δυνατή ταχύτητα που μπορούμε να πετύχουμε στο αναπτυξιακό ML405.

Το Aurora 8B/10B<sup>21</sup> είναι ένα software core περιφερειακό, που υλοποιεί υψηλής ταχύτητας σειριακή μεταφορά δεδομένων, βασισμένο στο πρωτόκολλο Aurora<sup>22</sup> της Xilinx. Έχει τεράστιες δυνατότητες παραμετροποίησης, κάτι που καθιστά τη χρήση του ιδανική σε εφαρμογές που απαιτούν υψηλή ταχύτητα αλλά και απλή αρχιτεκτονική. Η λειτουργία του βασίζεται στη χρήση μέχρι και 16 Virtex-4 Rocket IOMGT (multi-gigabit transceivers) στοιχείων, δίνοντας του τη δυνατότητα για ταχύτητες από 1,26 έως 100 Gbps. Η αρχική παραγωγή του Aurora 8B/10B γίνεται μέσω του CORE Generator της Xilinx όπου μας δίνει πληθώρα επιλογών για την προσαρμογή του στις ανάγκες μας.

Το Aurora 8B/10B περιλαμβάνει<sup>23</sup>:

- Κανάλια επικοινωνίας γενικευμένης χρήσης με εύρος ζώνης από 1,26 έως 100 Gbps.
- Χρήση μέχρι και 16 MGT σε Virtex-4 FPGA.
- Συμβατότητα με το πρωτόκολλο Aurora 2.0 (κωδικοποίηση 8B/10B).
- Πολύ χαμηλό χωρητικό κόστος.
- Επιλογή λειτουργίας full-duplex ή simplex.
- Δυνατότητα επικοινωνίας τύπου LocalLink (framing) ή streaming.
- Προαιρετική υποστήριξη ελέγχου ροής (flow control).



Εικόνα 8: Τρόπος σύνδεσης δύο καναλιών Aurora

Όπως αναφέραμε, το Aurora8B/10B βασίζεται στη χρήση των RocketIOMGT<sup>24</sup> (πομποδέκτες πολλαπλών gigabit) στοιχείων τηςFPGAγια τη σειριακή μεταφορά δεδομένων. Κάθε στοιχείο MGTμεταξύ άλλων περιέχει τον SERDES (serializer/deserializer), buffersεισόδου/εξόδου, clockgenerator/clockcorrectionγια την κατάλληλη ρύθμιση της συχνότητας χρονισμού της επικοινωνίας, καθώς και τον κωδικοποιητή/αποκωδικοποιητή 8B/10B.Η XC4VFX20 Virtex-4 FPGAπου εξοπλίζει το ML405 διαθέτει συνολικά 8 MGTστοιχεία, από αυτά, 2 είναι αφιερωμένα στους υποδοχείς SATA. Μπορούμε λοιπόν να χρησιμοποιήσουμε τα συγκεκριμένα MGTστοιχεία και σε συνδυασμό με τους υποδοχείς SATA,να υλοποιήσουμε δύο αμφίδρομα σειριακά κανάλια υψηλής ταχύτητας βασισμένα στο πρωτόκολλο Aurora. Φυσικά οι υποδοχείςSATA, χρησιμοποιούνται σαν μέσο μεταφοράς των δεδομένων και η χρήση τους δεν έχει καμία σχέση με το ομώνυμο πρωτόκολλο.

Τα δύο αμφίδρομα κανάλια Auroraείναι προσαρμοσμένα για να αποδώσουν την υψηλότερη ταχύτητα με το μικρότερο χωρητικό κόστος. Χρησιμοποιούν 1 lane(MGT) πλάτους τεσσάρων bytesκαι μέθοδο επικοινωνίας streaming. Η συχνότητα αναφοράς στους υποδοχείς SATAείναι 150 MHz, δίνοντας τελικό εύρος ζώνης 1,5 Gbps. Τα δύο κανάλια Auroraσυνδέονται στον επεξεργαστή PowerPCμέσω FSL, με αυτόν το μηχανισμό ο επεξεργαστής λαμβάνει/στέλνει δεδομένα στους δύο γειτονικούς του κόμβους και μπορεί να σταματήσει εντελώς τη λειτουργία των καναλιών Auroraεξοικονομώντας ενέργεια. Τέλος, να αναφέρουμε ότι με αυτές τις επιλογές, τα δύο κανάλια Auroraμαζί με τη σύνδεση FSLστον επεξεργαστή PowerPC, καταναλώνουν συνολικά 1000slices. Η χρήση ενός μόνο PLBEthernetπεριφερειακού στα 100 Mbps απαιτεί 900. Διαπιστώνουμε λοιπόν ότι η χρήση του Aurora,μας απέδωσε στον ίδιο περίπου αριθμό slicesπολύ μεγαλύτερο εύρος ζώνης (1,5 Gbps), καθώςκαι το διπλάσιο αριθμό καναλιών.

### 3.10 Παραγωγή και έλεγχος συχνότητων χρονισμού

Οι περισσότερες συσκευές στο ML405 (περιφερειακά και επεξεργαστές) χρειάζονται πολλές και διαφορετικές συχνότητες χρονισμού για να λειτουργήσουν. Άλλο ένα μεγάλο ζήτημα, είναι η διαχείριση των συσκευών που έχουμε δυναμική τροποποίηση της συχνότητας τους και η σχέση τους με τα περιφερειακά που έχουν σταθερή συχνότητα. Τα περιφερειακά που οργανώνουν τη σύνθεση όλων των συχνότητων είναι τα: ClockGenerator, PPCControl, CoreControlκαι ClocksControl.

Η γεννήτρια συχνότητωνClockGeneratorείναι το πρωταρχικό στοιχείο παραγωγής συχνότητων χρονισμού για όλα τα περιφερειακά και τους επεξεργαστές. Χρησιμοποιεί μια



συχνότητα αναφοράς από έναν κρύσταλλο (100 MHz) στο αναπτυξιακό ML405 και μέσω των στοιχείων DCM (digital clock managers) της FPGA, γίνεται η σύνθεση όλων των άλλων συχνοτήτων και η διάθεση τους στις συσκευές. Η XC4VFX20 Virtex-4 FPGA, όπου βασίζεται το ML405, διαθέτει συνολικά 4 στοιχεία DCM και βρίσκονται στην κεντρική στήλη πόρων της FPGA. Κατά αυτόν τον τρόπο, μπορούμε να καλύψουμε όλη την έκταση της με αρκετές συχνότητες για όλες τις συσκευές και επιτυγχάνοντας τη μικρότερη δυνατή καθυστέρηση μετάδοσης.

Το Clock Generator προσφέρει<sup>25</sup>:

- Λειτουργία ανάδρασης για τη διόρθωση καθυστερήσεων (clock skew).
- Σύνθεση μέχρι και 16 συχνοτήτων εξόδου.
- Σταθερή τροποποίηση φάσης από 0 έως 359 μοίρες.

Τα περιφερειακά PPC Control, Core Control και Clocks Control είναι υπεύθυνα για τη δυναμική τροποποίηση της συχνότητας χρονισμού των επεξεργαστών, καθώς και την εξοικονόμηση ενέργειας, αποκόπτοντας τις συχνότητες που δε χρησιμοποιούνται όταν σταματάει η λειτουργία κάποιων επεξεργαστών/περιφερειακών. Η λεπτομερής ανάλυση τους θα γίνει στο επόμενο κεφάλαιο.

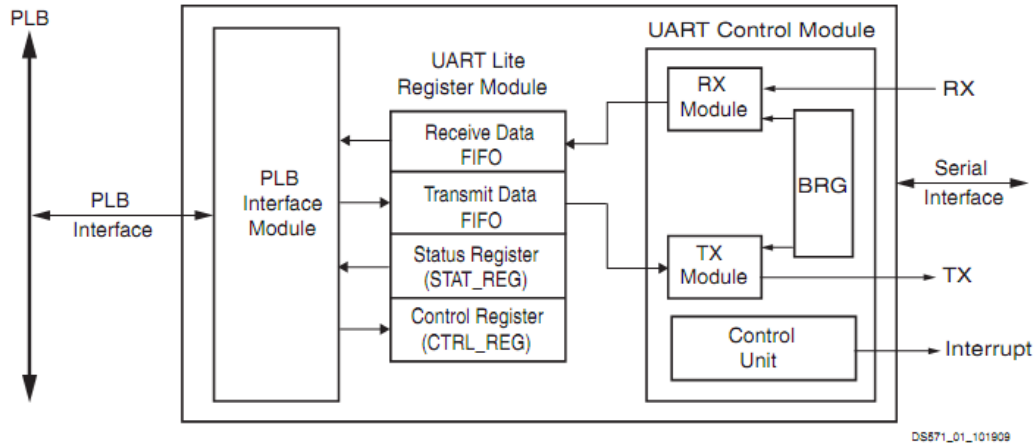
### 3.11 Περιφερειακό σειριακής επικοινωνίας UartLite

Για την επικοινωνία μεταξύ του αναπτυξιακού ML405 και του τερματικού υπολογιστή, κάνουμε χρήση της ενσωματωμένης σειριακής θύρας, με το software core περιφερειακό UartLite της Xilinx<sup>26</sup>. Το πρωτόκολλο επικοινωνίας που χρησιμοποιείται από το UartLite, είναι το ευρέως γνωστό σειριακό πρότυπο RS-232<sup>27</sup> (Recommended Standard 232) της Electronic Industries Alliance.

Το περιφερειακό UartLite προσφέρει:

- Υποστήριξη διαύλου PLB.
- Λειτουργία full duplex.
- Δύο FIFO, μετάδοσης και λήψης, βάθους 16 χαρακτήρων.
- Παραμετροποιήσιμος αριθμός bit ανά χαρακτήρα (5 - 8).
- Επιλογή του bitομοτιμίας (parity) σε μονό ή ζυγό (odd or even).
- Ταχύτητα μετάδοσης μέχρι και 1 Mbps.
- Προαιρετική χρήση σήματος interrupt.

Μέσω του UartLite ο επεξεργαστής PowerPC μπορεί να λαμβάνει και να στέλνει δεδομένα προς τον τελικό χρήστη, μέσω της διεπαφής που εκτελείται στον τερματικό υπολογιστή. Η λήψη δεδομένων από τον PowerPC προκαλεί τη χρήση του προαιρετικού σήματος interrupt, αποτρέποντας την άσκοπη κατανάλωση επεξεργαστικών πόρων παρακολουθώντας συνεχώς (polling) τη σειριακή θύρα. Η ταχύτητα που χρησιμοποιούμε για την εφαρμογή μας είναι 9.600 bps.



Εικόνα 9: Σχηματικό περιφερειακούUartLite

## 4 Δυναμική τροποποίηση συχνότητας χρονισμού

Η αρχιτεκτονική και τα στοιχεία που υλοποιούν το μηχανισμό δυναμικής τροποποίησης της συχνότητας χρονισμού των επεξεργαστών, καθώς και η δομή αλλά και η παραγωγή όλων των συχνοτήτων για τη λειτουργία του συστήματος, αναλύονται σε αυτό το κεφάλαιο. Η περιγραφή που ακολουθεί καλύπτει κυρίως τη σχεδίαση του hardware αλλά και τα προβλήματα που μπορούν να προκύψουν στο software και διάφορες μεθόδους επίλυσης τους.

### 4.1 Περιγραφή του προβλήματος

Όπως έχουμε αναφέρει, τα επεξεργαστικά στοιχεία της αρχιτεκτονικής μας είναι δύο.

1. Ο ενσωματωμένος κεντρικός επεξεργαστής PowerPC.
2. Έξι βοηθητικοί πυρήνες MicroBlaze (softprocessors).

Οι δύο τύποι επεξεργαστών, προσφέρουν αρκετές ρυθμίσεις παραμετροποίησης κατά τη διαδικασία σχεδίασης ενός συστήματος, καθώς και τη δυνατότητα επιλογής της συχνότητας χρονισμού τους. Ο PowerPC έχει τη δυνατότητα να λειτουργήσει σε συχνότητες έως και 300MHz ενώ ένας πυρήνας MicroBlaze προτείνεται για συχνότητες μέχρι 100MHz. Παρ' όλες τις επιλογές προσαρμογής κάθε επεξεργαστή, διαπιστώνουμε μια παντελής έλλειψη ρύθμισης της συχνότητας χρονισμού τους κατά τη λειτουργία τους. Κάτι τέτοιο δεν υποστηρίζεται με τα συγκεκριμένα



εργαλεία που προσφέρει η Xilinx, έτσι η δυνατότητα αυτή επωμίζεται στο σχεδιαστή, όπου θα πρέπει με ειδικά προγραμματιζόμενο hardware να υλοποιηθεί ο συγκεκριμένος μηχανισμός.

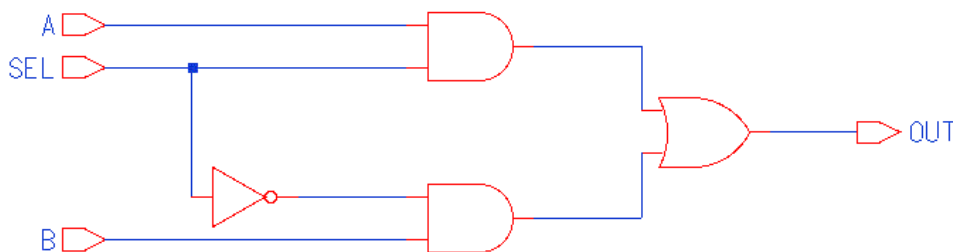
## 4.2 Αρχή λειτουργίας

Η αρχή λειτουργίας της δυναμικής τροποποίησης συχνότητας χρονισμού ενός επεξεργαστή, είναι σχετικά απλή και βασίζεται στις ιδιότητες της πολυπλεξίας (multiplexing). Με τη μέθοδο αυτή, θεωρητικά ένας απλός πολυπλέκτης μπορεί να εκτελέσει τη λειτουργία που επιζητούμε. Ο πολυπλέκτης είναι μια συσκευή που σκοπός της είναι να επιλέγει ένα σήμα μεταξύ πολλών, βάση της εισόδου που δέχεται.

Στην πιο απλή του μορφή, ένας πολυπλέκτης 2 - 1 μπορεί να υλοποιηθεί με 4 πύλες λογικής:

- 2 πύλες λογικού ΚΑΙ (andgate)
- 1 πύλη αναστροφής (or gate)
- 1 πύλη λογικού Ή (notgate)

Η μαθηματική έκφραση ενός πολυπλέκτη 2 σε 1 είναι:  $OUT = (A \cdot SEL) + (B \cdot \overline{SEL})$ .



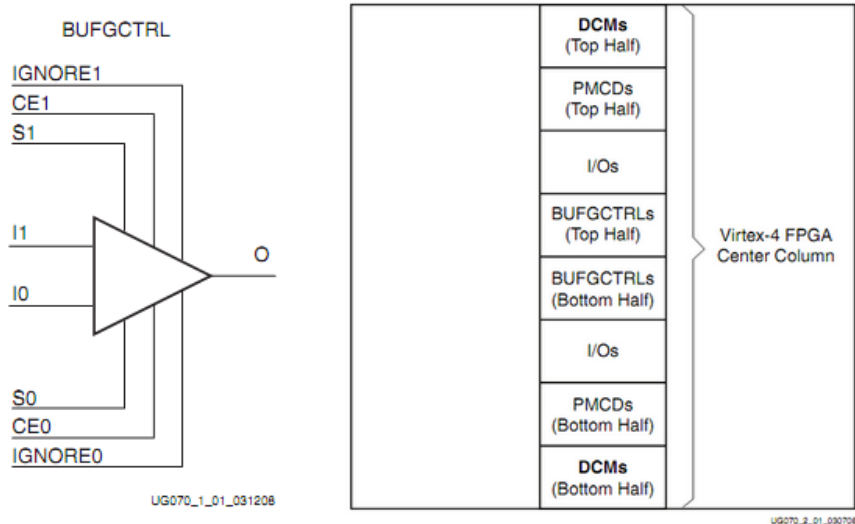
Εικόνα 10: Σχηματικό λειτουργίας πολυπλέκτη 2 σε 1

Στην εικόνα 9 βλέπουμε έναν πολυπλέκτη 2 σε 1 που μπορεί να επιλέξει μεταξύ δύο σημάτων Α και Β βάσει της εισόδου SEL. Όταν το σήμα SEL είναι στο λογικό 0, τότε στην έξοδο OUT θα παρουσιαστεί το σήμα Α, στην αντίθετη περίπτωση το σήμα Β θα εμφανιστεί στην έξοδο. Χρησιμοποιώντας τις εξόδους από δύο πολυπλέκτες στις εισόδους ενός τρίτου, μπορούμε με δύο σήματα επιλογής να επιλέξουμε μεταξύ τεσσάρων σημάτων εισόδου, έτσι συνδυάζοντας απλούς πολυπλέκτες 2 σε 1, μπορούμε να δημιουργήσουμε πολυπλέκτες με μεγαλύτερο εύρος σημάτων εισόδου (4 σε 1, 8 σε 1, κ.ο.κ).

## 4.3 Στοιχεία **BUFCTRL**

Δυστυχώς αν και θεωρητικά ορθό, μια υλοποίηση με απλές πύλες λογικής δεν κάνουν εφικτή τη δημιουργία ενός μηχανισμού δυναμικής τροποποίησης της συχνότητας χρονισμού. Για πολύ χαμηλές συχνότητες (κάτω από 20MHz) κάτι τέτοιο θα μπορούσε ίσως να πραγματοποιηθεί, αλλά οι πύλες μίας FPGA έχουν πολύ μεγάλη καθυστέρηση μετάδοσης για τις συχνότητες που χρειάζεται το σύστημα. Για να κρατήσουμε τις συχνότητες συγχρονισμένες είναι απαραίτητο η μετάδοσή τους να έχει την ελάχιστη καθυστέρηση, ειδικά σε μια αρχιτεκτονική όπου η χρήση της FPGA φτάνει το 100% της χωρητικότητας της και ήδη αρχίζουν να παρουσιάζονται προβλήματα δρομολόγησης.

Το πρόβλημα της αυξημένης καθυστέρησης μπορεί να λυθεί με τα στοιχεία BUFGCTRL<sup>28</sup> που υπάρχουν στη Virtex-4 FPGA. Κάθε στοιχείο BUFGCTRL προσφέρει τη δυνατότητα επιλογής μεταξύ δύο ασύγχρονων συχνοτήτων και είναι σχεδιασμένο κατά τέτοιο τρόπο ώστε να εγγυάται τη μικρότερη καθυστέρηση μετάδοσης, καθώς και να κάνει αδύνατη την απορύθμιση των δύο συχνοτήτων (clockglitch). Τα στοιχεία BUFGCTRL είναι συνολικά 32 και τοποθετημένα στην κεντρική στήλη της FPGA με άλλα ειδικά στοιχεία και πόρους όπως, I/O, PMCD και DCM. Η τοποθεσία τους στην κεντρική στήλη επιτρέπει το μοίρασμα των συγκεκριμένων πόρων και η πρόσβαση σε αυτά να γίνεται με ευκολία από όλη την έκταση της FPGA.



Εικόνα 11: Σχηματικό BUFGCTRL και κατανομή κεντρικών πόρων στην FPGA

Στην εικόνα 10 (δεξιά) βλέπουμε όλα τα σήματα ενός στοιχείου BUFGCTRL. I1 και I2 είναι οι δύο συχνοτήτες εισόδου και «O» η έξοδος της επιλεγμένης συχνότητας. Τα σήματα IGNORE1 και IGNORE0 επιτρέπουν την αλλαγή της συχνότητας ακαριαία, αλλά δεν εγγυώνται ότι η αλλαγή θα γίνει απροβλημάτιστα αν είναι ενεργά. Τέλος τα ζεύγη σημάτων CE1/S1 και CE0/S0 είναι υπεύθυνα για τη συχνότητα που θα μεταδοθεί στην έξοδο, ανάλογα με το ποια έχουν τη λογική τιμή 1.

#### 4.4 Περιφερειακό έλεγχο πυρήνων MicroBlazeCoreControl

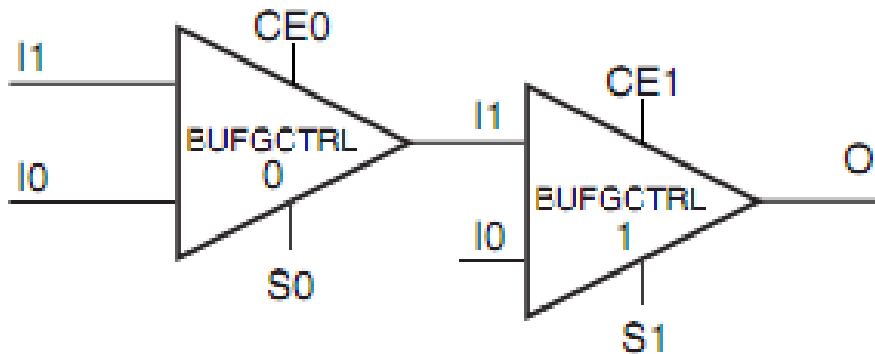
Για τον έλεγχο και τη δυναμική τροποποίηση συχνότητας χρονισμού των έξι πυρήνων MicroBlaze σχεδιάσαμε και υλοποιήσαμε, μέσω της γλώσσας VHDL, το FSL περιφερειακό CoreControl. Το συγκεκριμένο περιφερειακό επικοινωνεί με τον κεντρικό επεξεργαστή PowerPC μέσω FSL και συνολικά μπορεί να ελέγξει 8 διαφορετικούς πυρήνες MicroBlaze.

Συνοπτικά το CoreControl έχει τις παρακάτω δυνατότητες:

- Ελέγχει μέχρι και 8 πυρήνες MicroBlaze.
- Απλό λειτουργικό πρωτόκολλο μέσω FSL.
- Παύση λειτουργίας κάθε πυρήνα (χωρίς reset).
- Επανεκκίνηση κάθε πυρήνα με σήμα reset.
- Επιλογή ταχύτητας πυρήνων μεταξύ τριών συχνοτήτων χρονισμού.

Για την παύση λειτουργίας καθώς και την επιλογή συχνότητας των MicroBlaze, κάνουμε χρήση δύο BUFGCTRL στοιχείων σε σειρά. Κάθε στοιχείο BUFGCTRL έχει τη δυνατότητα επιλογής

μεταξύ δύο σημάτων, έτσι αν συνδέσουμε την έξοδο του πρώτου BUFGCTRL στην πρώτη είσοδο του δεύτερου BUFGCTRL, μπορούμε με δύο σήματα ελέγχου να επιλέξουμε μεταξύ τριών συχνοτήτων.

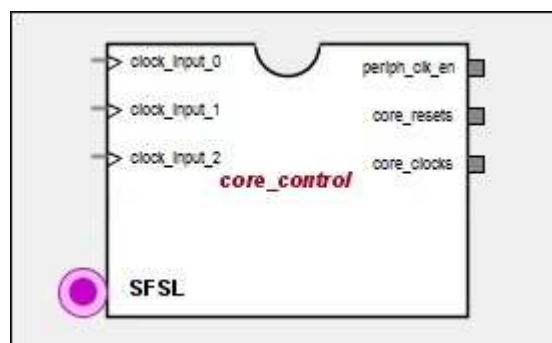


Εικόνα 12: Σύνδεση στοιχείων BUFGCTRL στο περιφερειακό CoreControl

Όπως παρατηρούμε στο σχηματικό της εικόνας 11 η συγκεκριμένη σύνδεση των BUFGCTRL είναι μη συμμετρική. Τα σήματα στις εισόδους I1 και IO του BUFGCTRL0 θα έχουν μεγαλύτερη καθυστέρηση μετάδοσης από ότι το σήμα IO του BUFGCTRL. Εκμεταλλευόμαστε αυτή την ιδιότητα, έτσι ώστε πάντα οι δύο πρώτες συχνότητες εισόδου (που καταλήγουν στο πρώτο BUFGCTRL), να είναι μικρότερες από ότι η τρίτη που συνδέεται κατευθείαν στην είσοδο του δεύτερου BUFGCTRL. Το αποτέλεσμα είναι ότι η μεγαλύτερη συχνότητα (όπου είναι και η πιο ευαίσθητη), έχει πάντα τη μικρότερη καθυστέρηση μετάδοσης μεταξύ των τριών.

Δυστυχώς η χρήση τριών (ή περισσότερων) BUFGCTRL για την επιλογή μεταξύ τεσσάρων συχνοτήτων είναι αδύνατη κυρίως για δύο λόγους:

1. Η (έστω και μικρή) καθυστέρηση μετάδοσης που έχουν τα στοιχεία BUFGCTRL αυξάνεται κρίσιμα, μετά τα δύο επίπεδα (ειδικά για μεγάλες συχνότητες).
2. Τα στοιχεία BUFGCTRL περιορίζονται στα 16 ανά τη μισή FPGA. Λόγο των έξι MicroBlaze θα χρειαζόμασταν σύνολο 18 BUFGCTRL στοιχεία (3 ανά πυρήνα) και ενώ συνολικά τα 32 BUFGCTRL είναι αρκετά, λόγω δρομολόγησης, η καθυστέρηση μετάδοσης αυξάνεται δραματικά.



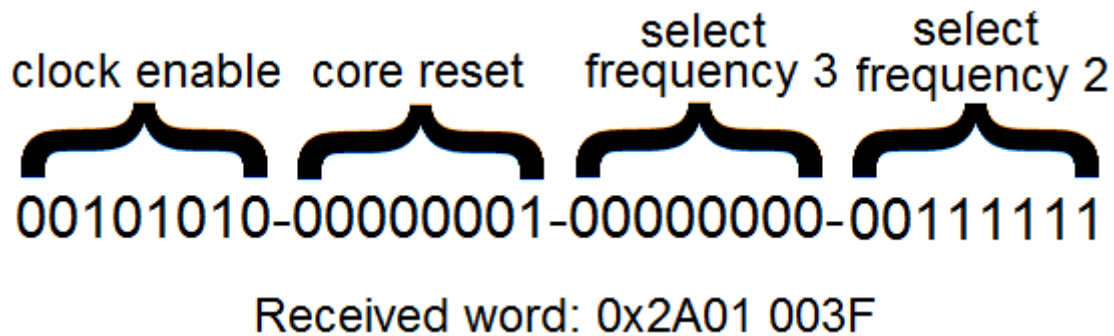
Εικόνα 13: Τα σήματα του περιφερειακού CoreControl

Τα σήματα εισόδου στο περιφερειακό CoreControl (εκτός των σημάτων FSL) είναι οι τρεις συχνότητες όπου θα χρονιστούν οι πυρήνες MicroBlaze. Τα σήματα εξόδου είναι:

1. periph\_clk\_en: Ένα προαιρετικό σήμα που καταλήγει στο ClocksControl για την αποκοπή των συχνοτήτων στα περιφερειακά των MicroBlaze.

2. `core_resets`: Για κάθε πυρήνα MicroBlaze υπάρχει ένα σήμα `core_resets` όπου καταλήγει στην είσοδο `resettw` πυρήνων.
3. `core_clocks`: Η επιλεγμένη συχνότητα χρονισμού για κάθε πυρήνα MicroBlaze.

Η λειτουργία του CoreControl γίνεται με 32-bit λέξεις (word) όπου στέλνονται σε αυτό μέσω ενός διαύλου FSL. Κάθε byte της εισερχόμενης λέξης (4 byte) είναι υπεύθυνο για μια συγκεκριμένη λειτουργία και κάθε bit ενός byte αντιστοιχεί σε συγκεκριμένο πυρήνα MicroBlaze. Η μέθοδος αυτή επιτρέπει να καλύψουμε όλους τους συνδυασμούς λειτουργίας των MicroBlaze. Τα δύο πρώτα byte είναι υπεύθυνα για την επιλογή της συχνότητας ενός πυρήνα, ενώ το τρίτο byte κάνει `resettw` πυρήνες. Τέλος, το τέταρτο byte αποκόπτει τη μετάδοση όλων των συχνοτήτων σταματώντας εντελώς τη λειτουργία του εκάστοτε πυρήνα.



Εικόνα 14: Παράδειγμα λειτουργίας CoreControl

Στην εικόνα 13 το περιφερειακό CoreControl έλαβε μέσω FSL τη λέξη 0x2A01 003F. Αναλύοντας τον αριθμό αυτό στο δυαδικό σύστημα και χωρίζοντας τον σε τέσσερα byte, μπορούμε να δούμε ακριβώς την κατάσταση του κάθε πυρήνα.

Από τα δεδομένα της εικόνας προκύπτει ότι:

- Και οι έξι πυρήνες λαμβάνουν τη δεύτερη συχνότητα εισόδου.
- Ο πυρήνας MicroBlaze 0 είναι σε κατάσταση `reset`
- Οι πυρήνες MicroBlaze 1, 3 και 5 λειτουργούν, ενώ όλοι οι άλλοι είναι σε κατάσταση παύσης.

#### 4.5 Περιφερειακό ελέγχου επεξεργαστή PowerPPCControl

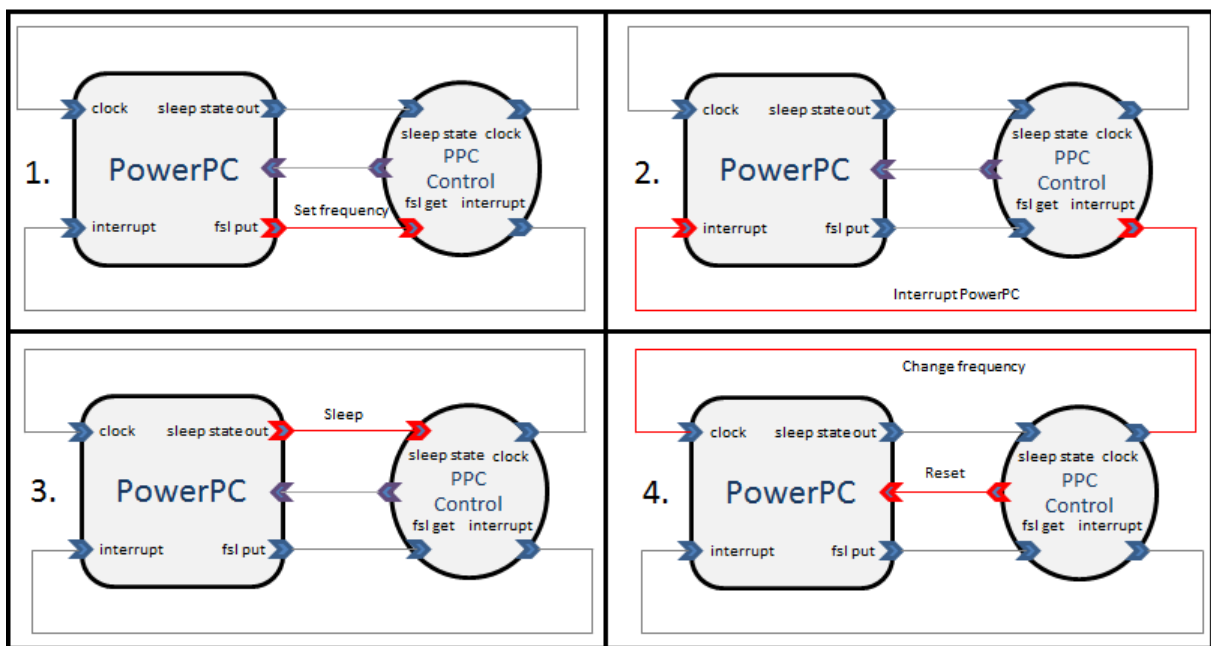
Για τη δυναμική τροποποίηση της συχνότητας χρονισμού του επεξεργαστή PowerPC σχεδιάσαμε και προγραμματίσαμε το περιφερειακό PPCControl. Όπως και στην περίπτωση του CoreControl το περιφερειακό PPCControl δέχεται εντολές από τον ίδιο τον επεξεργαστή μέσω ενός διαύλου FSL και κάνει δυνατή την επιλογή μεταξύ τριών διαφορετικών συχνοτήτων. Η ομοιότητες των δύο περιφερειακών όμως σταματάνε εδώ, αφού στην περίπτωση του CoreControl έχουμε μέχρι και 8 διαφορετικού πυρήνες (έξι στην περίπτωση μας), ενώ το PPCControl υποστηρίζει μόνο έναν επεξεργαστή PowerPC (άλλωστε δεν έχουμε στη διάθεση μας περισσότερους). Όπως θα δούμε παρακάτω το περιφερειακό PPCControl διαφέρει στον τρόπο αλλαγής συχνότητας του επεξεργαστή.

Η αρχιτεκτονική των πυρήνων MicroBlaze είναι αρκετά απλούστερη συγκριτικά με αυτή του κεντρικού επεξεργαστή PowerPC, για αυτό το λόγο η αλλαγή συχνότητας χρονισμού κατά το χρόνο εκτέλεσης γίνεται ακαριαία και χωρίς προβλήματα στους πυρήνες αυτούς. Δυστυχώς κάτι τέτοιο δεν ισχύει για τον επεξεργαστή PowerPC, αφού λόγω του μεγαλύτερου pipeline καθώς και της μνήμης cache που διαθέτει η αλλαγή της συχνότητας χρονισμού του δεν είναι μία τόσο εύκολη διαδικασία. Αν χρησιμοποιήσουμε τον ίδιο μηχανισμό με αυτόν του περιφερειακού CoreControl θα

διαπιστώσουμε ότι πολλές φορές η λειτουργία του επεξεργαστή PowerPC σταματάει απρόσμενα και η αλλαγή της συχνότητας γίνεται επιτυχώς μόνο μετά από την επανεκκίνηση του. Μια τέτοια συμπεριφορά κάνει το σύστημα μας άκρως αναξιόπιστο και μη λειτουργικό.

Ο μηχανισμός που υλοποιήσαμε για τις ανάγκες του PowerPC είναι υβριδικός και ονομάζεται «*resumeonreset*» (συνέχιση κατά την επανεκκίνηση). Βασίζεται τόσο στο hardware αλλά και σε σημαντικό βαθμό στο software και πρέπει να υπάρχει πλήρης συνεργασία μεταξύ των δύο για να γίνει δυνατή η αλλαγή συχνότητας κατά το χρόνο εκτέλεσης. Ο μηχανισμός *resumeonreset* είναι λειτουργικός κυρίως για δύο λόγους:

1. Η αλλαγή συχνότητας πραγματοποιείται επιτυχώς μετά από την επανεκκίνηση (reset) του επεξεργαστή, ακόμα και σε περίπτωση σφάλματος.
2. Τα δεδομένα της μνήμης του επεξεργαστή δε χάνονται μετά από μια επανεκκίνηση του, καθώς η SRAM δεν απενεργοποιείται κατά τη διαδικασία αυτή.



Εικόνα 15: Τα βήματα του μηχανισμού "ResumeonReset"

Στην εικόνα 14 μπορούμε να δούμε τη διαδικασία λειτουργίας του μηχανισμού *resumeonreset*. Συνοπτικά αποτελείται από τέσσερα κύρια βήματα:

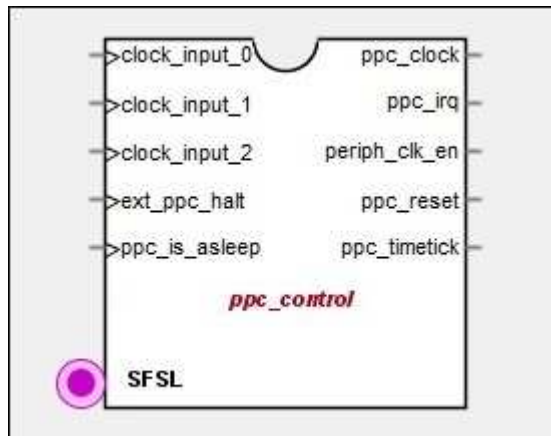
1. Το περιφερειακό PPCControl λαμβάνει εντολή παύσης ή αλλαγής συχνότητας του επεξεργαστή.
2. Το PPCControl απαντάει διακόπτοντας με interrupt την κανονική εκτέλεση του επεξεργαστή. Ο επεξεργαστής εκτελεί συγκεκριμένο κώδικα όπου μεταξύ άλλων, βάζει τον εαυτό του σε κατάσταση αδρανοποίησης (sleepmode).
3. Η κατάσταση αδρανοποίησης αναμένεται από το PPCControl.
4. Η επιλεγμένη συχνότητα αλλάζει και ταυτόχρονα ο επεξεργαστής γίνεται reset.

Φυσικά ο μηχανισμός *resumeonreset* δεν πραγματοποιεί μόνο αυτά τα τέσσερα βήματα αφού δεν έχουμε καλύψει τι γίνεται μετά την επανεκκίνηση του επεξεργαστή. Το σύστημα δεν μπορεί να είναι λειτουργικό ακόμα, καθώς η κατάσταση επανεκκίνησης διακόπτει τη λειτουργία και τη συγκεκριμένη διεργασία που εκτελούσε ο κόμβος.

Για να λύσουμε το πρόβλημα της επανεκκίνησης χρειαζόμαστε ειδικά προγραμματιζόμενο software. Στο στάδιο όπου το περιφερειακό PPCControl διακόπτει τον PowerPC με interrupt, πρέπει να εκτελέσουμε τα παρακάτω βήματα:

1. Να γράψουμε στην προτελευταία διεύθυνση μνήμης SRAM μια συγκεκριμένη τιμή, όπου σηματοδοτεί ότι ο επεξεργαστής θα αλλάξει συχνότητα.
2. Στην τελευταία διεύθυνση μνήμης να αποθηκευθεί η διεύθυνση της εντολής όπου ακολουθεί την εντολή αδρανοποίησης.
3. Ο επεξεργαστής να μπει σε κατάσταση αδρανοποίησης (sleepmode).

Αφού η κατάσταση αδρανοποίησης γίνει αισθητή από το PPCControl (και μόνο τότε) η συχνότητα θα αλλάξει και ο επεξεργαστής θα γίνει reset. Όταν ο επεξεργαστής ξεκινήσει πάλι τη λειτουργία του, τότε εκτελείται ένας ειδικός «bootloader» προγραμματισμένος σε assembly, όπου ελέγχει τις δύο τελευταίες θέσης μνήμης της SRAM και η εκτέλεση του επεξεργαστή συνεχίζει στη διεύθυνση όπου είχε αποθηκευθεί στο στάδιο του interrupt. Με τον bootloader, ο επεξεργαστής και η μνήμη αρχικοποιείται κανονικά όταν ξεκινάει για πρώτη φορά η λειτουργία του κόμβου ή όταν πατήσουμε χειροκίνητα το κουμπί reset. Όταν όμως έχει προηγηθεί αλλαγή της συχνότητας, τότε το interrupt που είχε σταματήσει στην εντολή αδρανοποίησης ολοκληρώνεται και ο επεξεργαστής συνεχίζει στο ακριβές σημείο όπου είχε διακοπεί.



Εικόνα 16: Τα σήματα του περιφερειακού PPCControl

Το περιφερειακό PPCControl έχει συνολικά πέντε σήματα εισόδου και πέντε σήματα εξόδου (χωρίς να περιλαμβάνουμε τα σήματα FSL). Ακολουθεί συνοπτική περιγραφή τους:

1. Τα `clock_input_0` έως `clock_input_2` είναι τα σήματα που φέρουν τις τρεις συχνότητες εισόδου.
2. Το `ext_ppc_halt` είναι ένα προαιρετικό σήμα όπου αποκόπτει τη λειτουργία του PowerPC όσο είναι στο λογικό 1. Χρησιμοποιείται για παύση της λειτουργίας του επεξεργαστή από εξωτερική πηγή.
3. Όταν ο PowerPC μπει σε κατάσταση αδρανοποίησης (sleepmode) το σήμα `ppc_is_asleep` παίρνει την τιμή 1 (από τον PowerPC).
4. Το σήμα `ppc_clock` είναι η επιλεγμένη συχνότητα χρονισμού που μεταδίδεται στον PowerPC.
5. Το σήμα `ppc_irq` γίνεται 1 όταν το περιφερειακό δεχθεί εντολή αλλαγής συχνότητας. Με αυτό το σήμα γίνεται interrupt ο PowerPC.
6. Επίσης προαιρετικό σήμα, το `periph_clk_en` παίρνει την τιμή 0 όταν παύει να λειτουργεί ο επεξεργαστής. Σε συνδυασμό με το στοιχείο `ClocksControl` αποκόπτει τις συχνότητες χρονισμού των περιφερειακών για την εξοικονόμηση ενέργειας.



7. Σε κάθε αλλαγή συχνότητας το σήμα `ppc_reset` γίνεται 1 και ακολουθεί η επανεκκίνηση του επεξεργαστή.
8. Το `ppc_timertick` είναι η συχνότητα προσαύξησης του εσωτερικού timer στον επεξεργαστή PowerPC. Η λειτουργία του αναλύεται στην επόμενη παράγραφο.

Ο κεντρικός επεξεργαστής PowerPC παρέχει έναν προγραμματιζόμενο timer (PIT). Ο προγραμματιζόμενος timer χρησιμοποιείται από το λειτουργικό πυρήνα Xikernel για την αλλαγή των thread (contextswitch), καθώς και για την υλοποίηση των software timer. Ο PowerPC εκτός από το σήμα εισόδου για τη συχνότητα λειτουργίας του, διαθέτει προαιρετικό σήμα εισόδου για τη συχνότητα προσαύξησης των timer που διαθέτει<sup>29</sup>. Ο προγραμματιζόμενος timer αλλάζει τιμή σε κάθε θετικό παλμό των δύο συχνοτήτων και το πρόβλημα που προκύπτει είναι ότι όταν έχουμε αλλαγή της συχνότητας του επεξεργαστή, τότε ο timer αλλάζει τιμή σε διαφορετικό χρονικό διάστημα και έτσι δεν μπορούμε να μετρήσουμε την πάροδο του χρόνου αξιόπιστα. Για αυτό το λόγο κάνουμε χρήση του σήματος `ppc_timertick` του PPCControl. Το `ppc_timertick` είναι μια νέα σταθερή συχνότητα που προκύπτει από το λογικό άθροισμα των τριών συχνοτήτων εισόδου (`clock_input_0 AND clock_input_1 AND clock_input_2`). Επακόλουθο είναι ότι σε οποιαδήποτε συχνότητα λειτουργίας, ο PowerPC είναι σε θέση να μετράει χρονικά διαστήματα αξιόπιστα.

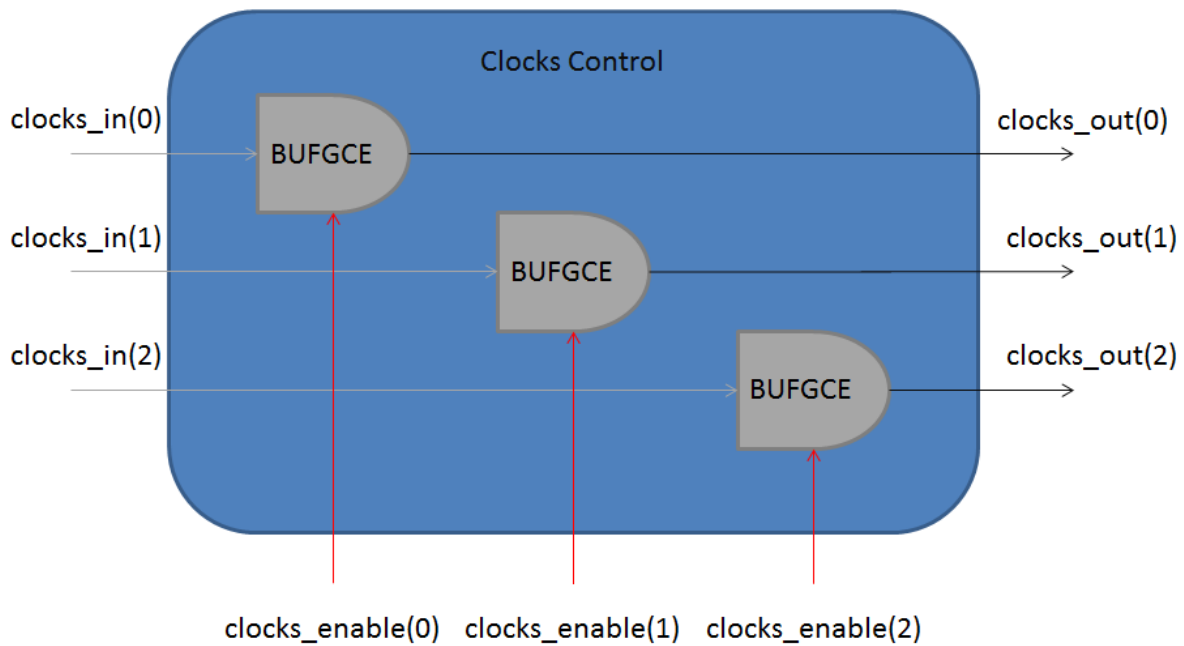
Κλείνοντας, βλέπουμε ότι ο PowerPC έχει ένα αρκετά πιο πολύπλοκο σύστημα δυναμικής τροποποίησης της συχνότητας χρονισμού σε σχέση με τους πυρήνες MicroBlaze. Με κατάλληλα σχεδιασμένο hardware, καθώς και τη χρήση ειδικών προγραμματιζόμενων ρουτινών σε assembly, μπορούμε να αλλάζουμε συχνότητα αξιόπιστα και απρόσκοπτα χωρίς να απαιτείται το σύστημα να ξεκινήσει τη λειτουργία του από την αρχή.

#### **4.6 Αποκοπή συχνοτήτων περιφερειακών μέσω του ClocksControl**

Για την περαιτέρω εξοικονόμηση ενέργειας, σχεδιάσαμε το ClocksControl, όπου βασίζεται στο στοιχείο BUFGCE (υλοποιείται με BUFGCTRL) και λειτουργεί σε συνδυασμό με τα περιφερειακά PPCControl και CoreControl. Το ClocksControl έχει δύο σειρές εισόδων και μια σειρά εξόδων:

1. Εισερχόμενα σήματα `clocks_enable`.
2. Εισερχόμενα σήματα `clocks_in`.
3. Εξερχόμενα σήματα `clocks_out`.

Κάθε συχνότητα εισόδου `clocks_in` μεταδίδεται στην αντίστοιχη έξοδο `clocks_out`, μόνο στην περίπτωση όπου η αντίστοιχη είσοδος `clocks_enable` είναι στη λογική τιμή 1. Η μέθοδος αυτή μας επιτρέπει να αποκόψουμε πλήρως τις συχνότητες χρονισμού πολλών περιφερειακών μαζί, απλά αλλάζοντας κατάσταση σε ένα σήμα `clocks_enable`.



Εικόνα 17: Σχηματικό λειτουργίας ClocksControl

#### 4.7 Δίκτυο και δομή συχνοτήτων χρονισμού στο σύστημα

Κλείνοντας το κεφάλαιο, θα πρέπει να αναφέρουμε όλες τις υπάρχουσες συχνότητες χρονισμού στο σύστημα και να περιγράψουμε τη διανομή τους σε όλα τα περιφερειακά και τους επεξεργαστές. Λόγω του μεγάλου αριθμού συχνοτήτων στο σύστημα καθώς και τη χωρητική χρήση ολόκληρης της FPGA, οι χρονικοί περιορισμοί (timingconstraints) είναι πολύ αυστηροί. Βάση όλων αυτών των χωρητικών και χρονικών περιορισμών, αναγκαστήκαμε να βελτιώσουμε την αρχιτεκτονική στο έπακρο, για να πραγματοποιηθεί η σωστή και αξιόπιστη λειτουργία του συστήματος.

Συνολικά, η Virtex-4 XC4VFX20FPGA, διαθέτει τέσσερα DCM. Οι ανάγκες της αρχιτεκτονικής μας χρειάζονται τουλάχιστον τρία, καθώς είναι απαραίτητη η συνολική σύνθεση οκτώ συχνοτήτων. Η αρχική συχνότητα είναι ένας κρύσταλλος 100MHz της EPSON και η σύνθεση όλων των υπόλοιπων συχνοτήτων γίνεται από το περιφερειακό ClockGenerator της Xilinx, που έχει ως είσοδο το συγκεκριμένο κρύσταλλο. Η χρήση του ClockGenerator προσφέρει μεγάλη παραμετροποίηση και ευκολία στη σύνθεση των συχνοτήτων κατά το στάδιο του σχεδιασμού, καθώς και μεγάλη αξιοπιστία μέσω της ιδιότητας του clockdeskew.

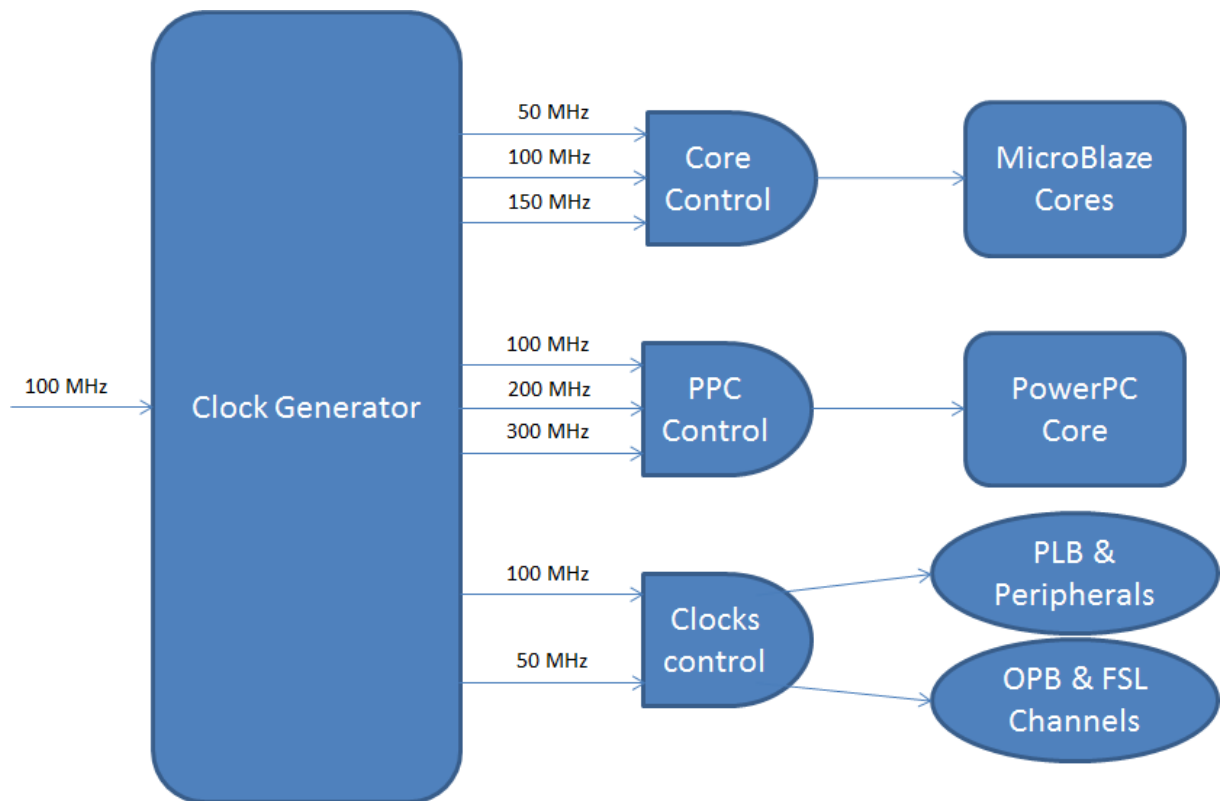
Οι συχνότητες του συστήματος έχουν ως εξής:

- A. Συχνότητες DCM0: Περιφερειακά, δίαυλοι FSL, PLB και OPB
  1. CLKOUT0: 50 MHz
  2. CLKOUT1: 100 MHz
- B. Συχνότητες DCM 1: Πυρήνες MicroBlaze
  1. CLKOUT2: 50 MHz
  2. CLKOUT3: 100 MHz
  3. CLKOUT4: 150 MHz
- C. Συχνότητες DCM 2: Επεξεργαστής PowerPC
  1. CLKOUT5: 100 MHz
  2. CLKOUT6: 200 MHz
  3. CLKOUT7: 300 MHz



Οι τιμές όλων των συχνοτήτων είναι προσεκτικά επιλεγμένες για τη μέγιστη απόδοση ταχύτητας και λειτουργίας. Η προτεινόμενη συχνότητα χρονισμού των πυρήνων MicroBlaze δίνεται στα 100 MHz παρ' όλα αυτά, καταφέραμε να αυξήσουμε τη συχνότητα τους στα 150MHz, βελτιώνοντας τη δρομολόγηση των σημάτων χρονισμού.

Άλλο ένα σημείο άξιο προσοχής, είναι οι συχνότητες των επεξεργαστών σε σχέση με τους διαύλους επικοινωνίας. Οι διάυλοι OPB και PLB υποστηρίζουν συσκευές που είναι χρονισμένες μόνο σε πολλαπλάσιες συχνότητες από αυτές των διαύλων (λειτουργία 1:N). Στην περίπτωση του PowerPC ο διάυλος PLB είναι χρονισμένος στα 100 MHz, έτσι η συχνότητα χρονισμού του PowerPC είναι πολλαπλάσια των 100, δηλαδή 100 (1:1), 200 (1:2) και 300(3:1) MHz. Κάτι αντίστοιχο συμβαίνει μεταξύ των πυρήνων MicroBlaze και του διαύλου OPB, όπου ο διάυλος έχει συχνότητα χρονισμού 50MHz και οι πυρήνες λειτουργούν σε συχνότητες 50(1:1), 100 (1:2) και 150 (1:3) MHz.



Εικόνα 18: Κατανομή συχνοτήτων σε όλο το σύστημα

## 5 Κατασκευή και λειτουργία αισθητηρίου ισχύος

Ένα σημαντικό στόχος της πτυχιακής εργασίας είναι η ακριβής παρακολούθηση της κατανάλωσης ενέργειας (ισχύς) του κάθε κόμβου και συνολικά όλου του δικτύου. Το κεφάλαιο αυτό θα ασχοληθεί κυρίως με τον τρόπο λειτουργίας του αισθητηρίου και τη διαδικασία κατασκευής του.

### 5.1 Η Ισχύς ως μέγεθος κατανάλωσης ενέργειας

Η Ισχύς είναι ο ρυθμός παραγωγής έργου, ή πιο κατανοητά, ο ρυθμός μεταβίβασης ενέργειας. Η ισχύς ενός κυκλώματος ορίζεται ως το γινόμενο της τάσης επί το συνολικό ηλεκτρικό ρεύμα που το διατρέχει και συνήθως ο συγκεκριμένος τύπος εκφράζεται ως  $P = V \cdot I$  (όπου Ρη ισχύςσε Watt, Vη τάση σε Voltκαι Iτο ρεύμα σε Ampere).Αντικαθιστώντας τα μεγέθη τάσης και ρεύματος μπορούμε να επαναδιατυπώσουμε τον τύπο της ισχύος ως (όπου Rη αντίσταση του κυκλώματοςσε Ohm):

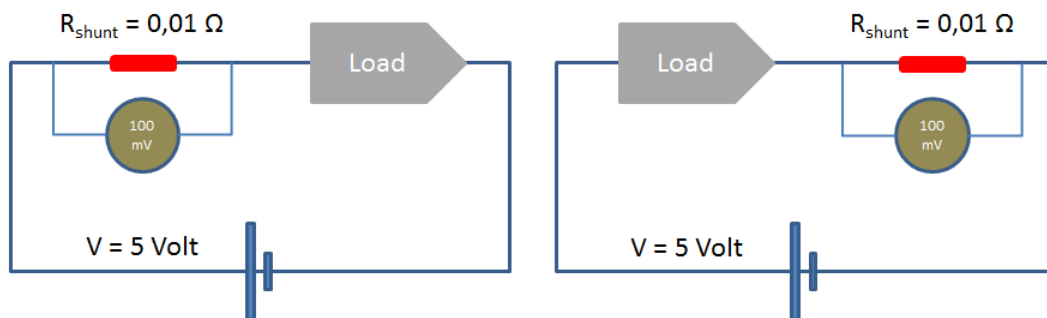
- $P = I^2 \cdot R$
- $P = \frac{V^2}{R}$

Η Σιμονάδα μέτρησης για την ισχύ είναι το Watt, όπου 1 Wattισούται με 1 Jouleanά δευτερόλεπτο. Σε όλες τις μετρήσεις κατανάλωσης ενέργειας σε αυτή την εργασία χρησιμοποιούμε την υποδιαίρεση milliwatt (mW).

### 5.2 Μέθοδοι μέτρησης ισχύος

Για να μετρήσουμε την κατανάλωση ενέργειας ενός κόμβου, όπως είδαμε στην προηγούμενη παράγραφο, είναι αναγκαίο να ξέρουμε την εφαρμοζόμενη σε αυτόν τάση καθώς και το ηλεκτρικό ρεύμα. Την τάση του κόμβου τη γνωρίζουμε, αφού το τροφοδοτικό του αναπτυξιακού ML405 έχει έξοδο 5 Volt(DC), η μέτρηση του ρεύματος όμως είναι προβληματική, αφού πρέπει να επέμβουμε στο κύκλωμα προσθέτοντας ένα πολύμετρο σε σειρά. Δυστυχώς κάτι τέτοιο δεν μπορούμε να το αποφύγουμε αν θέλουμε να έχουμε ακριβείς μετρήσεις ισχύος.

Θεωρητικά η μέτρηση ρεύματος σε ένα κύκλωμα, γίνεται με την εν σειρά προσθήκη μίας (ιδανικής) ωμικής αντίστασης, τόσο μικρή, όπου ουσιαστικά δεν επηρεάζεται η λειτουργία του κυκλώματος (shuntresistor). Βάση του νόμου του Ohm ( $V = I \cdot R$ ),η τάση που εμφανίζεται στους ακροδέκτες της αντίστασης, ισούται με το γινόμενο του ρεύματος που τη διατρέχει επί την τιμή της. Αφού η συγκεκριμένη αντίσταση βρίσκεται σε σειρά με το κύκλωμα, τότε μπορούμε να πούμε ότι το ρεύμα που περνάει από την αντίσταση είναι ακριβώς το ίδιο με το συνολικό ρεύμα του κυκλώματος.



Εικόνα 19: Τρόποι μέτρησης ρεύματος.High-side (αριστερά) και Low-side (δεξιά)

Στην εικόνα 18 βλέπουμε τους τρόπους «High-και Low-side» μέτρησης ρεύματος με αντίσταση σε σειρά<sup>30</sup>. Ο πρώτος, ουσιαστικά εισάγει την αντίσταση shunt μεταξύ πηγής και φορτίου, ενώ ο δεύτερος κάνει το ακριβώς αντίθετο, εισάγοντας την μεταξύ φορτίου και γείωσης. Κάθε τρόπος έχει διαφορετικά πλεονεκτήματα και αδυναμίες και η επιλογή μεταξύ των δύο γίνεται πάντα ανάλογα τους περιορισμούς και τις ανάγκες του κυκλώματος.

Οι κύριοι λόγοι χρήσης του τρόπου low-side είναι:

1. Αποφυγή θορύβου κοινής τάσης στη μέτρηση.
2. Ασφαλέστερη μέτρηση σε κυκλώματα υψηλής τάσης.

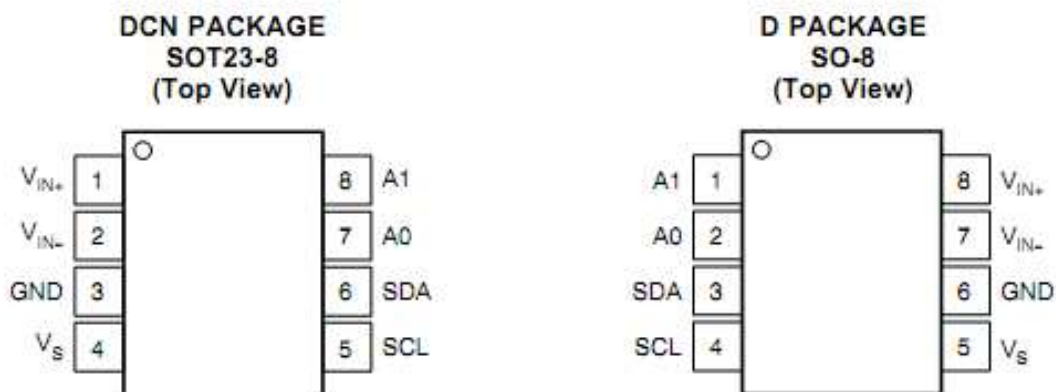
Υπάρχουν όμως και μειονεκτήματα όπως:

1. Η συσκευή υπό μέτρηση χάνει την άμεση επαφή με τη γείωση και αυτό μπορεί να επιφέρει προβλήματα σε επιπρόσθετα κυκλώματα ελέγχου ή την έκκληση θορύβου.
2. Μόνο το ρεύμα που επιστρέφει άμεσα στην πηγή θα μετρηθεί από το αισθητήριο. Τυχόν διαρροές δε λαμβάνονται υπόψη, έτσι υπάρχει πιθανότητα οι μετρήσεις να είναι λάθος.

Στα δύο παραδείγματα της εικόνας 18, η πτώση τάσης των 100 mV που εμφανίζεται στην αντίσταση shunt, μας επιτρέπει να υπολογίσουμε το διερχόμενο ρεύμα σε όλο το κύκλωμα με την εφαρμογή του νόμου του Ohm. Τελικά το συνολικό διερχόμενο ρεύμα είναι: — Ampere.

### 5.3 Κύκλωμα λειτουργίας INA219 και κατασκευή αισθητηρίου

Όπως αναφέραμε στο κεφάλαιο 2.2 το ολοκληρωμένο INA219 της Texas Instruments προσφέρει ένα υψηλής ακρίβειας high-side αισθητήριο ρεύματος, βασισμένο στο δίαυλο επικοινωνίας I<sup>2</sup>C. Βασίζεται σε τεχνολογία SMT πολύ μικρών διαστάσεων SOT23-8 και SO-8. Η διάσταση που χρησιμοποιούμε σε αυτήν την εργασία είναι η SOT23-8.

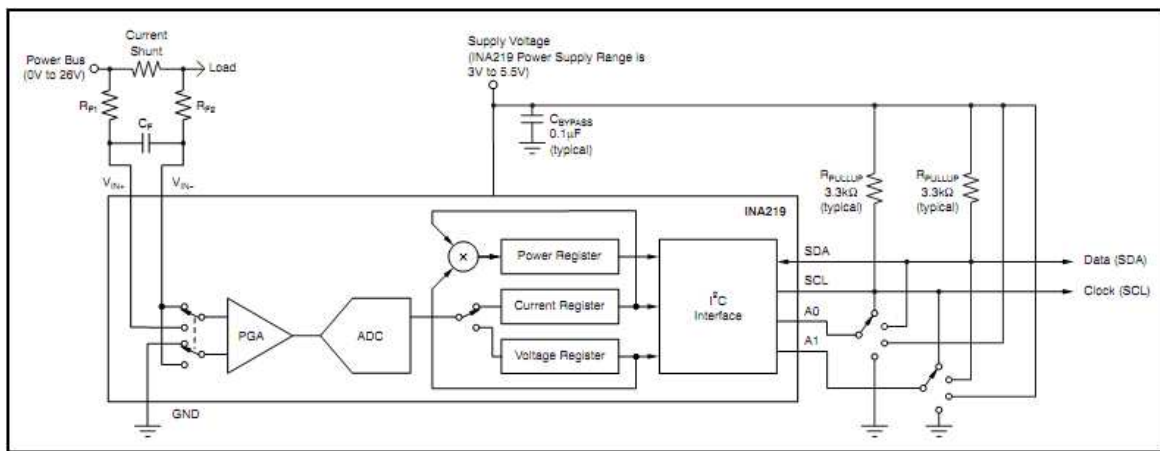


Εικόνα 20: Περιγραφή και σήματα του INA219. Package SOT23-8 και SO-8

Το INA219 περιλαμβάνει μόλις 8 σήματα για τη λειτουργία του:

- A. Σήματα I<sup>2</sup>C:
1. SCL: Χρησιμοποιείται για το συγχρονισμό της επικοινωνίας.
  2. SDA: Τα δεδομένα στέλνονται και λαμβάνονται σειριακά μέσω αυτού του σήματος.

3. A0-A1: Σήματα καθορισμού της διεύθυνσης I<sup>2</sup>C της συσκευής. Ανάλογα με το πώς θα συνδεθούν, δίνουν διαφορετική τιμή διεύθυνσης. Η σύνδεση τους μπορεί να γίνει στα σήματα V<sub>S</sub>, GND, SDAή SCL. Το χαρακτηριστικό αυτό επιτρέπει τη σύνδεση 16 συσκευών INA219 στον ίδιο δίαυλο επικοινωνίας I<sup>2</sup>C.
- B. Σήματα τροφοδοσίας:
1. V<sub>S</sub>: Το σήμα αυτό πρέπει να συνδεθεί στο θετικό πόλο μιας πηγής 3 έως 5 Volt.
  2. GND: Η γείωση του ολοκληρωμένου.
- C. Σήματα μέτρησης τάσης:
1. V<sub>IN+</sub>: Είσοδος μέτρησης πτώσης τάσης στο θετικό shunt (προς την πηγή) άκρο της αντίστασης shunt.
  2. V<sub>IN-</sub>: Είσοδος μέτρησης πτώσης τάσης στο αρνητικό (προς το φορτίο) άκρο της αντίστασης shunt.



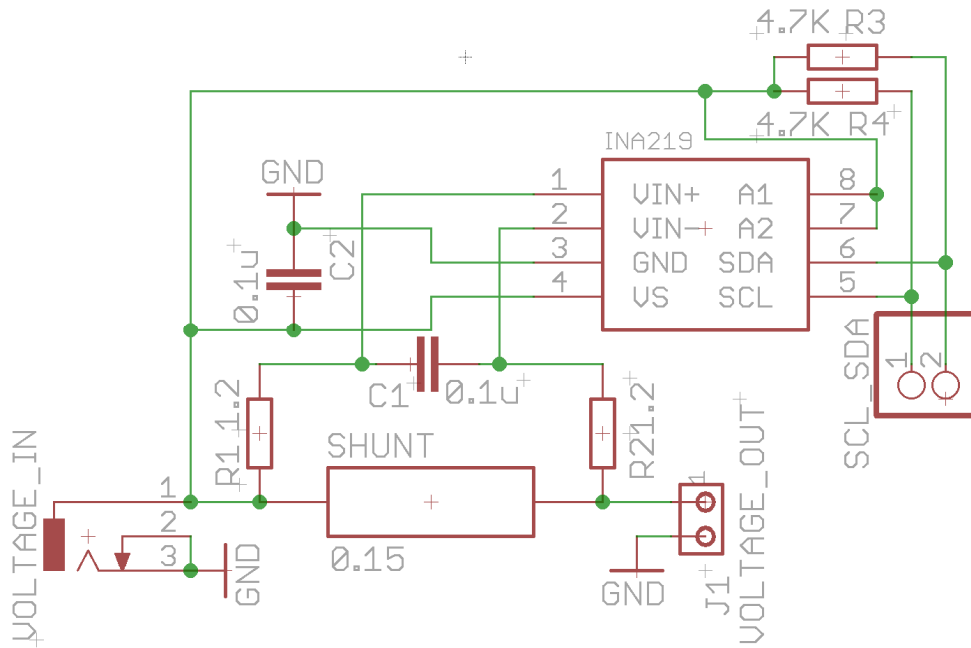
Εικόνα 21: Τυπικό κύκλωμα λειτουργίας του INA219

Για να λειτουργήσει σωστά το INA219 απαιτεί την υλοποίηση ενός συμπληρωματικού κυκλώματος κυρίως για την αξιόπιστη επικοινωνία μέσω I<sup>2</sup>C, καθώς και την αποφυγή θορύβου στις μετρήσεις. Συγκεκριμένα χρειάζονται:

1. Φυσικά μια αντίσταση shunt για να μετρηθεί η πτώση τάσης που ψάχνουμε.
2. Δύο αντιστάσεις «pull-up» 1-10kΩ στα σήματα SDA και SCL για τη λειτουργία του διαύλου I<sup>2</sup>C.
3. Φίλτρο σταθεροποίησης τάσης με πυκνωτή 0,1 µF, μεταξύ V<sub>S</sub> και GND.
4. Φίλτρο θορύβου με αντιστάσεις 0 -10Ω και πυκνωτή 0,1-1µF στα σήματα μέτρησης V<sub>IN+</sub> και V<sub>IN-</sub>.

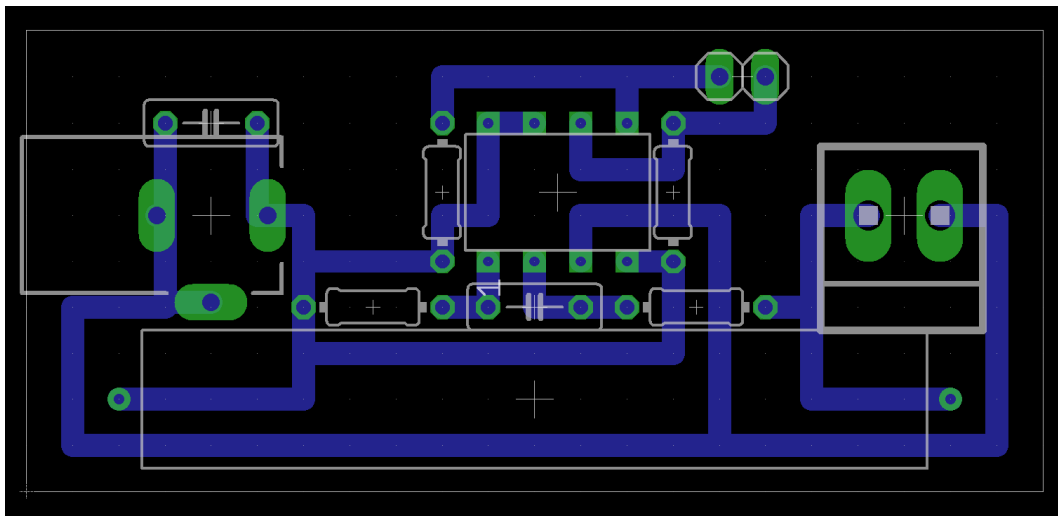
Συνολικά για την κατασκευή του αισθητηρίου χρησιμοποιήσαμε:

- Αντίσταση shunt 0,15 Ω στα 10 Watt.
- Pull-up αντιστάσεις 4,7kΩ για τα σήματα SDA, SCL.
- Δύο πυκνωτές 0,1µF.
- Δύο αντιστάσεις 1,2 Ω για το φίλτρο θορύβου στα σήματα V<sub>IN+</sub> και V<sub>IN-</sub>.
- Βάση SOT23-8 σε PDIP-8 για το ολοκληρωμένο INA219.
- Ομοαξονικό υποδοχέα τροφοδοσίας 2,5 mm.
- Κλέμα τροφοδοσίας και pinheaders.



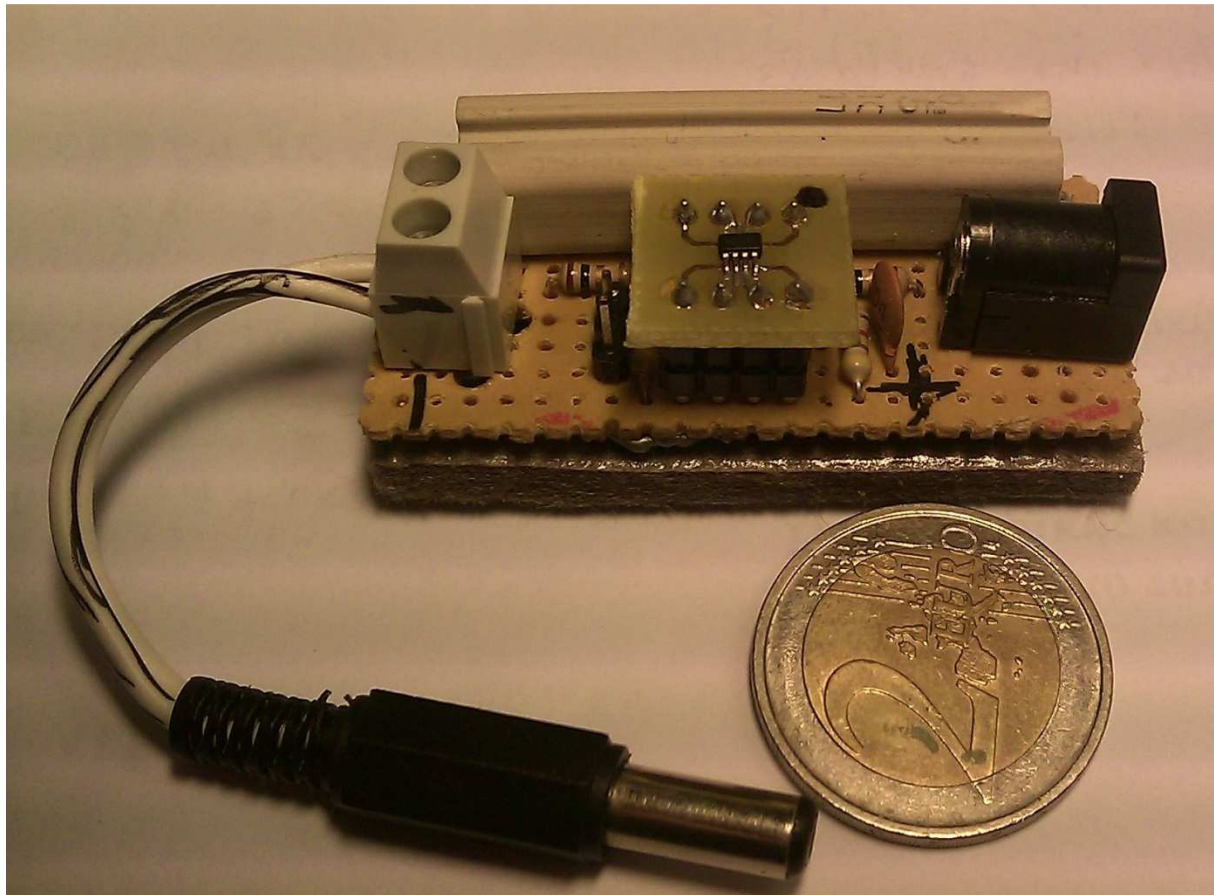
Εικόνα 22: Τελικό σχηματικό αισθητήριου

Για τη σχεδίαση του σχηματικού καθώς και την υλοποίηση του κυκλώματος χρησιμοποιήσαμε το πρόγραμμα EAGLE. Μέσω του εργαλείου «autorouter» και κατάλληλων ρυθμίσεων, έγινε βελτιστοποίηση στις διαστάσεις του αισθητήριου και αυτοματοποίηση της δρομολόγησης των χαλκο-διαδρόμων.



Εικόνα 23: Κάτοψη PCB αισθητήριου

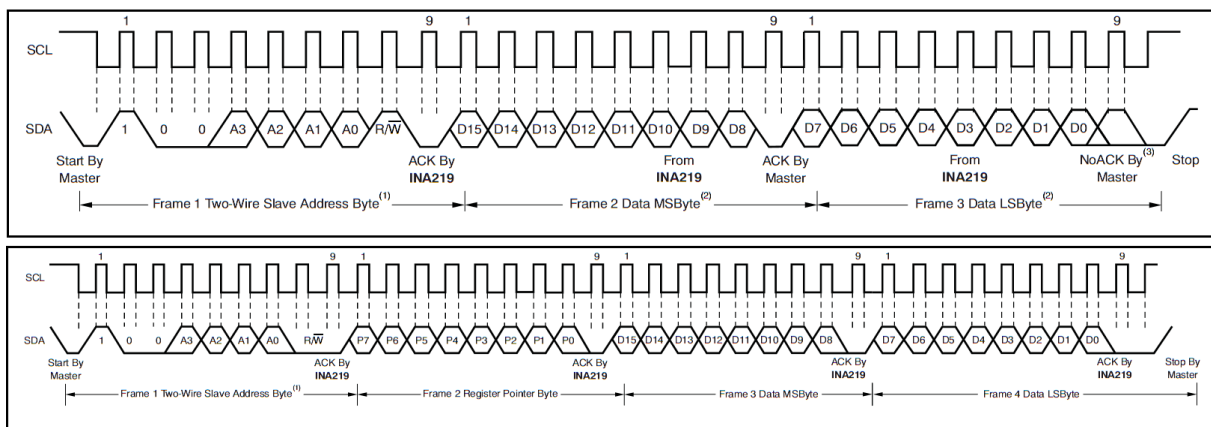
Στην εικόνα 22 παρουσιάζεται το κύκλωμα προς εκτύπωση για την κατασκευή του PCB. Με μπλε χρώμα διακρίνονται οι χαλκο-διάδρομοι, ενώ με πράσινο οι καταλήξεις των σημάτων (pads). Αξίζει να σημειώσουμε ότι το αυξημένο πάχος των χαλκο-διαδρόμων είναι εσκεμμένο διότι το συνολικό ρεύμα που διατρέχει το κύκλωμα μπορεί να είναι μέχρι και 2 Ampere, κάτι που απαιτεί πολύ μεγαλύτερο πάχος από ότι σε συνηθισμένα κυκλώματα SMT<sup>31</sup>. Η απαιτήσεως ρεύματος στο αναπτυξιακό ML405 είναι ανάλογες με το πλήθος των περιφερειακών που χρησιμοποιούνται, τη συνολική χρήση της FPGA αλλά και τη συχνότητα χρονισμού σε κάθε στοιχείο.



Εικόνα 24: Τελική κατασκευή και μέγεθος του αισθητηρίου

## 5.4 Επικοινωνία μέσω I<sup>2</sup>C

Το INA219 επικοινωνεί με τον επεξεργαστή PowerPCσειριακά μέσω του διαύλου I<sup>2</sup>C, χρησιμοποιώντας μόλις δύο σήματα. Το σήμαSCLγια το συγχρονισμό της επικοινωνίας και το SDAγια τη μετάδοση της πληροφορίας. Ουσιαστικά σε κάθε μορφής επικοινωνία,το σήμα SCLμεταβάλλεται μεταξύ λογικού 1 και 0 σε συχνότητα καθορισμένη από τον PowerPC (masterσυσκευή), στην περίπτωση μας 100KHz. Όταν υπάρξει αλλαγή κατάστασης του σήματος SCLη τιμή του SDAαποθηκεύεται, έτσι κάθε 8 εναλλαγές έχουμε μεταφέρει 1 byteπληροφορίας. Όταν δεν υπάρχει μετάδοση πληροφορίας τα σήματα SDAκαι SCLείναι στο λογικό 1, έτσι όλες οι slaveσυσκευές γνωρίζουν τότε ο δίαυλος είναι ελεύθερος.



Εικόνα 25: Χρονικό διάγραμμα σημάτων SCLκαι SDA. Γραφή και ανάγνωση δεδομένων



Το πρωτόκολλο I<sup>2</sup>C διαθέτει τρεις βασικού τύπου επικοινωνίας:

1. Γραφή δεδομένων στη συσκευή slave ενός byte.
2. Ανάγνωση δεδομένων από τη συσκευή slave ενός byte.
3. Πολλαπλή γραφή και ανάγνωση δεδομένων.

Μεταξύ κάθε μορφής επικοινωνίας υπάρχουν ειδικά μηνύματα ελέγχου, αυτά είναι:

- START: Έναρξη της επικοινωνίας
- STOP: Λήξη της επικοινωνίας
- ACK: Αναγνώριση (acknowledge) λήψης/γραφής byte.
- RepeatedSTART: Χρησιμοποιείται για τον τρίτο τύπο επικοινωνίας.

Πίνακας 1: Διεύθυνση INA219 ως προς τη σύνδεση των σημάτων A1 και A0

A1	A0	Διεύθυνση	A1	A0	Διεύθυνση
GND	GND	1000000	SDA	GND	1001000
GND	V <sub>S</sub>	1000001	SDA	V <sub>S</sub>	1001001
GND	SDA	1000010	SDA	SDA	1001010
GND	SCL	1000011	SDA	SCL	1001011
V <sub>S+</sub>	GND	1000100	SCL	GND	1001100
V <sub>S+</sub>	V <sub>S</sub>	1000101	SCL	V <sub>S</sub>	1001101
V <sub>S+</sub>	SDA	1000110	SCL	SDA	1001110
V <sub>S+</sub>	SCL	1000111	SCL	SCL	1001111
SDA	GND	1001000	SDA	GND	1001000
SDA	V <sub>S</sub>	1001001	SDA	V <sub>S</sub>	1001001

Στην εικόνα 23 φαίνονται τα χρονικά διαγράμματα για τα σήματα SDA/SCL, καθώς και ο τρόπος μετάδοσης των μηνυμάτων ελέγχου. Κάθε έναρξη επικοινωνίας ξεκινάει πάντα με τον επεξεργαστή να γράφει στο δίαυλο I<sup>2</sup>C τη διεύθυνση της slave συσκευής όπου θέλει να επικοινωνήσει. Κάθε I<sup>2</sup>C συσκευή έχει μια διεύθυνση 7 bit όπου δίνει συνολικό αριθμό 127 συσκευών σε ένα δίαυλο. Στο INA219 καθορίζεται από τη συνδεσμολογία των σημάτων A0 και A1 (πίνακας 1). Το 8<sup>ο</sup> bit της διεύθυνσης καθορίζει αν το επόμενο byte μετάδοσης ζητάει δεδομένα (λογικό 1) ή γράφει δεδομένα στη slave συσκευή (λογικό 0).

## 5.5 Βαθμονόμηση αισθητηρίου

Το ολοκληρωμένο INA219 διαθέτει συνολικά 6 καταχωρητές:

1. Configuration register: Περιέχει ρυθμίσεις για τη λειτουργία της συσκευής.
2. Shunt voltage: Η μετρούμενη πτώση τάσης στην αντίσταση shunt.
3. Bus voltage: Τάση τροφοδοσίας της συσκευής υπό μέτρηση.
4. Power: Ισχύς που καταναλώνεται, εφόσον έχει γίνει βαθμονόμηση.
5. Current: Ρεύμα που διέρχεται από την αντίσταση shunt (χρειάζεται βαθμονόμηση).
6. Calibration register: Καταχωρητής βαθμονόμησης.

Για τη σωστή αναφορά ισχύος και ρεύματος πρέπει πρώτα να έχει γίνει ακριβής βαθμονόμηση στο αισθητήριο. Η βαθμονόμηση του αισθητηρίου γίνεται σε δύο στάδια<sup>32</sup>:

1. Ακριβής υπολογισμός και ενημέρωση του καταχωρητή βαθμονόμησης.
2. Προαιρετική διόρθωση καταχωρητή βαθμονόμησης, βάση εξωτερικής μέτρησης.

Ο υπολογισμός του καταχωρητή βαθμονόμησης έγινε με την παρακάτω μέθοδο:

- Ορίζουμε τα αρχικά σταθερά μεγέθη του συστήματος
  - Μέγιστη τάση αντίστασης shunt  $V_{ms} = 320 \text{ mV}$ . Η τάση αυτή είναι η μέγιστη για το INA 219, αφού το ADC του δεν μπορεί να μετρήσει υψηλότερη τάση.
  - Τιμή αντίστασης shunt  $S_R = 0,16 \Omega$ . Η αντίσταση που έχουμε επιλέξει είναι κανονικά  $0,15 \Omega$ , αλλά λόγω του μήκους των χαλκο-διαδρόμων (από την αντίσταση στο αισθητήριο), θεωρούμε ότι η συνολική αντίσταση αυξήθηκε.
- Υπολογίζουμε το μέγιστο ρεύμα που μπορεί να υπάρξει, βάση τη μέγιστη τάση αντίστασης και την τιμή της αντίστασης.
  - $C_{tm} = \frac{V_{ms}}{R_s} = 2 \text{ A}$ .
- Εμπειρική επιλογή του μέγιστου ρεύματος στο σύστημα μας. Βάση μετρήσεων πριν την κατασκευή του αισθητηρίου, είδαμε ότι το ρεύμα στην αρχιτεκτονική μας δεν μπορεί να ξεπεράσει τα  $1,9 \text{ Ampere}$ .
  - $C_{sm} \leq C_{tm} = 1,9 \text{ A}$ .
- Υπολογισμός και επιλογή μεταξύ ελάχιστου και μέγιστου, λιγότερο σημαντικού bit ρεύματος (LSB). Το ελάχιστο LSB (least significant bit) ρεύματος είναι  $C_{max_{lsb}} = \frac{C_{sm}}{32767} = 57,98 \mu\text{A}$ , ενώ το μέγιστο είναι  $C_{min_{lsb}} = \frac{C_{sm}}{4096} = 463,86 \mu\text{A}$ . Διαλέγουμε μεταξύ αυτών, το μικρότερο δυνατό στρογγυλοποιημένο αριθμό. Στην περίπτωση μας διαλέξαμε ακρίβεια ρεύματος  $60 \mu\text{A}$ .
  - $C_{lsb} \leq C_{max_{lsb}} \geq C_{min_{lsb}} = 60 \mu\text{A}$ .
- Υπολογισμός του καταχωρητή βαθμονόμησης
  - $Cal = \frac{0,04096}{C_{lsb} \cdot S_r} = 4.267$ .
- Προαιρετικά, αν ξέρουμε συγκεκριμένη τιμή ρεύματος μπορούμε να κάνουμε επιπλέον βαθμονόμηση στο αισθητήριο τροποποιώντας τον καταχωρητή εκ νέου.
  - $Cal_n = \frac{Cal \cdot C_n}{C_{INA}}$  ( $C_n$  αναμενόμενο ρεύμα,  $C_{INA}$  ρεύμα που αναφέρει το INA219)

Με τη συγκεκριμένη μέθοδο βαθμονόμησης, την προσεκτική επιλογή των στοιχείων του κυκλώματος, καθώς και την υλοποίηση φίλτρων για την εξάλειψη θορύβου, καταφέραμε να υλοποιήσουμε ένα αισθητήριο με ακρίβεια ισχύος  $1,2 \text{ mW}$ . Αν αναλογιστούμε ότι το αναπτυξιακό ML405 καταναλώνει  $6.000$  έως  $8.000 \text{ mW}$  στην αρχιτεκτονική μας, τότε μπορούμε να διακρίνουμε την ακρίβεια μέτρησης του INA219.



## 6 Ανάλυση λογισμικού κόμβου και διεπαφής

Αφού συνοψίσαμε και περιγράψαμε τα επιμέρους στοιχεία της αρχιτεκτονικής του συστήματος, μένει να αναλύσουμε το λογισμικό (software) που πλαισιώνει τον τρόπο λειτουργίας του δικτύου, καθώς και την παρουσίαση των δεδομένων σε τερματικό υπολογιστή μέσω διεπαφής. Η ανάλυση του λογισμικού που θα ακολουθήσει θα γίνει περιγραφικά καθώς η λεπτομερής κάλυψη περίπου 4.000 γραμμών πηγαίου κώδικα είναι πέρα από το σκοπό αυτής της εργασίας.

### 6.1 Έναρξη λειτουργίας κόμβου

Κατά την έναρξη έχουμε μια σειρά από διαδικασίες για την αρχικοποίηση των περιφερειακών και των επεξεργαστών του συστήματος. Η κύρια διαδικασία εκκίνησης ονομάζεται «poweronreset» και εμφανίζεται όταν θέσουμε το διακόπτη του αναπτυξιακού ML405 στη θέση ON. Η σωστή αρχικοποίηση όλων των στοιχείων στο σύστημα καλύπτεται από το μηχανισμό poweronreset.

Κατά τη διαδικασία poweronreset συμβαίνουν τα παρακάτω:

- Ενεργειακή αρχικοποίηση του συστήματος, σταθεροποίηση της τάσης στην FPGA και τα περιφερειακά.
- Επιλογή και ανάγνωση του bitstream από την κάρτα CompactFlash.
- Προγραμματισμός της FPGA με το επιλεγμένο bitstream.
- Αναμονή των DCM για την παραγωγή και τη σταθεροποίηση των συχνοτήτων χρονισμού.
- Εκκίνηση του συστήματος hardware/software.

Μόλις προγραμματιστεί η FPGA τότε ξεκινάει η λειτουργία του διαύλου PLB, της μνήμης SRAM καθώς και της κοινής μνήμης BRAM. Με τη λειτουργία αυτών, ο επεξεργαστής PowerPC είναι σε θέση να ξεκινήσει (boot) από την τελευταία θέση μνήμης της BRAM και να συνεχίσει την εκτέλεση κώδικα μέσω της SRAM. Τα υπόλοιπα περιφερειακά και επεξεργαστές του συστήματος, ελέγχονται έμμεσα ή άμεσα από τον PowerPC και η έναρξη της λειτουργίας τους γίνεται από τον ίδιο σε επόμενο στάδιο.

Εφόσον έχει ξεκινήσει ο PowerPC περνάμε στη φάση εκκίνησης λογισμικού σε assembly. Η πρώτη διαδικασία που εκτελείται είναι ο boot loader που έχουμε προγραμματίσει για το μηχανισμό resumeonreset. Σε αυτό το στάδιο ελέγχεται αν έχει προκληθεί επανεκκίνηση του συστήματος από αλλαγή συχνότητας. Φυσικά στην πρώτη εκκίνηση δεν ισχύει κάτι τέτοιο, έτσι περνάμε στη φάση αρχικοποίησης της μνήμης (heap, stack, global variables). Το τελικό στάδιο είναι η έναρξη της συνάρτησης main() και η αρχή εκτέλεσης κώδικα γραμμένο σε γλώσσα C.

### 6.2 Έναρξη Xilkernel και περιφερειακών

Η συνάρτηση main() σηματοδοτεί την εκτέλεση κώδικα C και την αρχική λειτουργία του συστήματος. Πριν όμως αρχίσουμε την επεξεργασία δεδομένων πρέπει πρώτα να κάνουμε χρήση του λειτουργικού πυρήνα Xilkernel<sup>33</sup>. Όπως αναφέραμε στο κεφάλαιο 3.2, ο πυρήνας Xilkernel μας επιτρέπει τη χρήση εργαλείων παραλληλοποίησης όπως threads και software timers. Αργότερα θα δούμε πως αρκετές διαδικασίες στο λογισμικό μας, είναι απαραίτητο να εκτελούνται παράλληλα και χωρίς τη χρήση threads το σύστημα θα ήταν δυσλειτουργικό και μη αποδοτικό.

Βήματα μέχρι την έναρξη της κύριας εφαρμογής επεξεργασίας:

- Ενεργοποίηση του APUγια την επικοινωνία με τα περιφερειακά FSL.
- Αρχικοποίηση του πυρήνα Xilkernel.
- Δημιουργία masterthreadκαι έναρξη του Xilkernel.
- Αρχικοποίηση των σημάτων interrupt.
- Ρύθμιση και βαθμονόμηση του αισθητηρίου INA219.
- Ρύθμιση οθόνης LCDκαι σειριακής θύρας RS-232.
- Δημιουργία threadγια την αποστολή αποτελεσμάτων σε LCD, RS-232 και Aurora.
- Εκκίνηση κύριας εφαρμογής.

Σε αυτό το σημείο, μέσω των interrupts,ο επεξεργαστής είναι σε θέση να λάβει και να στείλει μηνύματα στους γειτονικούς κόμβους του δικτύου και τους πυρήνες MicroBlaze και να επικοινωνεί με τον τερματικό υπολογιστή. Παράλληλα, λαμβάνονται μετρήσεις από τα αισθητήρια ισχύος και θερμοκρασίας και ενημερώνεται ο τερματικός υπολογιστής και η οθόνη LCD.

### 6.3 Κύρια εφαρμογή ελέγχου και επεξεργασίας

Με τη δημιουργία των κατάλληλων διεργασιώνκαθώς και την έναρξη των περιφερειακών, μπορούμε πλέον να εκτελέσουμε το κύριο μέρος της εφαρμογής μας, όπου είναι υπεύθυνο για:

1. Εκτέλεση συνάρτησης επεξεργασίας δεδομένων.
2. Έλεγχος και αλλαγή συχνότητας των επεξεργαστών του συστήματος.
3. Λήψη δεδομένων επεξεργασίας από του πυρήνες MicroBlaze.

Η κύρια εφαρμογή ουσιαστικά είναι μια συνάρτηση κλειστού βρόχου, όπου εκτελείται κάθε 1 δευτερόλεπτο. Σε κάθε επανάληψη ρυθμίζει τη συχνότητα χρονισμού των πυρήνων MicroBlazeκαι του PowerPC, βάση των δεδομένων που έλαβε μέσω τηςRS-232 ή από τα κανάλια Aurora. Έπειτα, σηματοδοτεί τους πυρήνες MicroBlazeκαι PowerPCνα ξεκινήσουν την επεξεργασία δεδομένων και μετά από 1 δευτερόλεπτο επεξεργασίας τα αποτελέσματααποθηκεύονται και αποστέλλονται, είτε στο δίκτυο (λειτουργία slave),είτε στον τερματικό υπολογιστή για την εμφάνιση τους στο χρήστη (λειτουργία master).

Η συνάρτηση επεξεργασίας που εκτελούν όλοι οι πυρήνες, είναι μια δοκιμασία επιδόσεων του συστήματος, βασισμένη στην κρυπτογραφημένη συνάρτησηκατακερματισμούMD5 (messagedigest5)<sup>34</sup>. Το ζητούμενο είναι να δούμε πόσες συνόψεις MD5,μπορεί κάθε επεξεργαστής να υπολογίσει κατά τη διάρκεια ενός δευτερολέπτου. Σε κάθε επανάληψη επαληθεύεται το αποτέλεσμα της σύνοψης και αναφέρονται πιθανών λάθη. Η συνάρτηση επεξεργασίας θα μπορούσε να είναι οποιοσδήποτε αλγόριθμος. Στόχος της πτυχιακής δεν είναι η έρευνα απόδοσης συγκεκριμένου αλγόριθμου, αλλά η υλοποίηση ενός δικτύου ενσωματωμένων συστημάτων και κυρίως η εφαρμογή του μηχανισμού δυναμικής τροποποίησης συχνότητας χρονισμού σε επεξεργαστές. Έτσι η επιλογή του MD5 έχει ως σκοπό να παρουσιάσει την επίτευξη της ιδέας του εγχειρήματος (proofofconcept)και όχι τη μελέτη του.

### 6.4 Δίκτυο δακτυλίου

Κάθε αναπτυξιακό ML405 διαθέτει υποδοχείς SATA, όπου στην περίπτωση μας είναι το μέσο μεταφοράς δεδομένων των αμφίδρομων καναλιών επικοινωνίας Aurora. Έχοντας στη διάθεση μας δύο κανάλια επικοινωνίας μπορούμε να υλοποιήσουμε ένα δίκτυο τοπολογίας δακτυλίου. Η κύρια επιλογή της τοπολογίας αυτής είναι η απόδοση, αφού μια τοπολογία διαύλου (bustopology) βασισμένη μόνο σε ένα κανάλι θα ήταν σημαντικά πιο αργή στη μετάδοση πληροφορίας και θα χρειαζόταν πολυπλοκότερο πρωτόκολλο (arbitration). Η λειτουργία του δικτύου από άποψης λογισμικού είναι απλού δακτυλίου, που σημαίνει ότι τα δεδομένα μεταδίδονται μόνο προς μία κατεύθυνση. Η συγκεκριμένη επιλογή έγινε κυρίως για την απλοποίηση του πρωτόκολλου

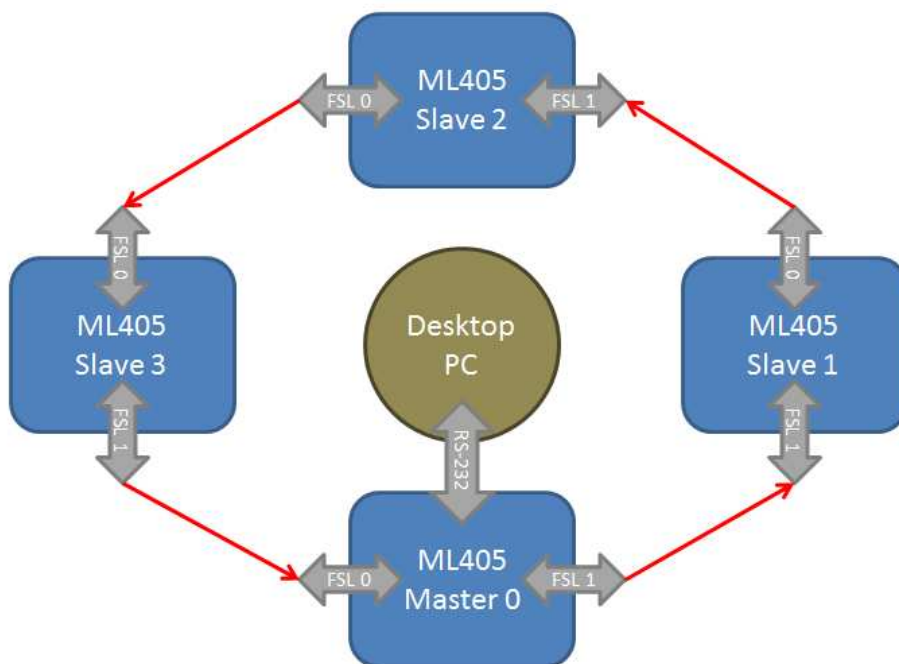
επικοινωνίας, παρ' όλα αυτά η αρχιτεκτονική μας εμπριέχει αμφίδρομα κανάλια Aurora και μέσω λογισμικού μπορεί εύκολα να επεκταθεί σε λειτουργία διπλού δακτυλίου.

Τα πλεονεκτήματα της τοπολογίας δακτυλίου είναι:

- Τοπολογία απλή και ίσης μεταχείρισης κόμβων.
- Αποδίδει καλύτερα από αντίστοιχες τοπολογίες διαύλου.
- Δε χρειάζεται επιπλέον hardware μεταξύ των κόμβων για τη σύνδεση τους.

Τα κύρια μειονεκτήματα είναι:

- Αν κάποιος κόμβος δεν αποκρίνεται, σταματάει όλη η λειτουργία του δικτύου.
- Αλλαγή στον αριθμό των κόμβων ή οι μετακίνηση τους κατά τη λειτουργία του δικτύου, προκαλούν προβλήματα.

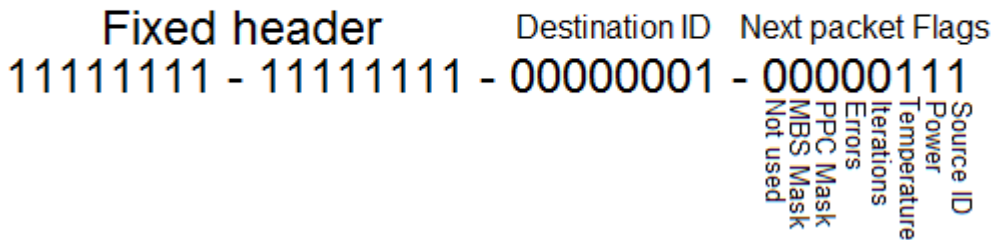


Εικόνα 26: Δίκτυο τοπολογίας απλού δακτυλίου

Οι μετρήσεις στο δίκτυο μεταφέρονται και παρουσιάζονται στον τελικό χρήστη μέσω διεπαφής, που εκτελείται σε ένα τερματικό υπολογιστή. Τα δεδομένα φτάνουν στον υπολογιστή μέσω της σειριακής θύρας RS-232 και όπως είναι λογικό δεν είναι πρακτικό να έχουν όλοι οι κόμβοι άμεση επαφή μαζί του. Αυτή η ιδιομορφία, μας αναγκάζει να θέσουμε έναν κόμβο ως το master της τοπολογίας και οι υπόλοιποι έχουν ρόλο slave. Ο master κόμβος εκτελεί τη διευθυνσιοδότηση του δικτύου και τροποποιεί της συχνότητες χρονισμού όλων των άλλων κόμβων. Οι υπόλοιποι κόμβοι στέλνουν τα δεδομένα τους προς την κατεύθυνση του master προωθούν τα δεδομένα των γειτόνων τους προς αυτόν.

Κάθε κανάλι Aurora έχει ξεχωριστό σήμα interrupt προς τον επεξεργαστή PowerPC, έτσι ο επεξεργαστής ενημερώνεται άμεσα για τα νέα εισερχόμενα δεδομένα και με μόλις μια συνάρτηση μπορεί να γίνει επεξεργασία, ή προώθηση τους στον επόμενο κόμβο. Λόγο της αμεσότητας του δακτυλίου, το πρωτόκολλο επικοινωνίας στο δίκτυο είναι πάρα πολύ απλό, με μόλις δύο μηνύματα ελέγχου, το μήνυμα «startofframe» και το «forceidframe». Το πρώτο μήνυμα ελέγχου καθορίζει τον παραλήπτη καθώς και πόσα πακέτα πληροφορίας ακολουθούν. Το δεύτερο είναι υπεύθυνο για τη διευθυνσιοδότηση του δικτύου.

## Start of Frame Structure Received word: 0xFFFF0107



Εικόνα 27: Δομή πακέτου startofframe

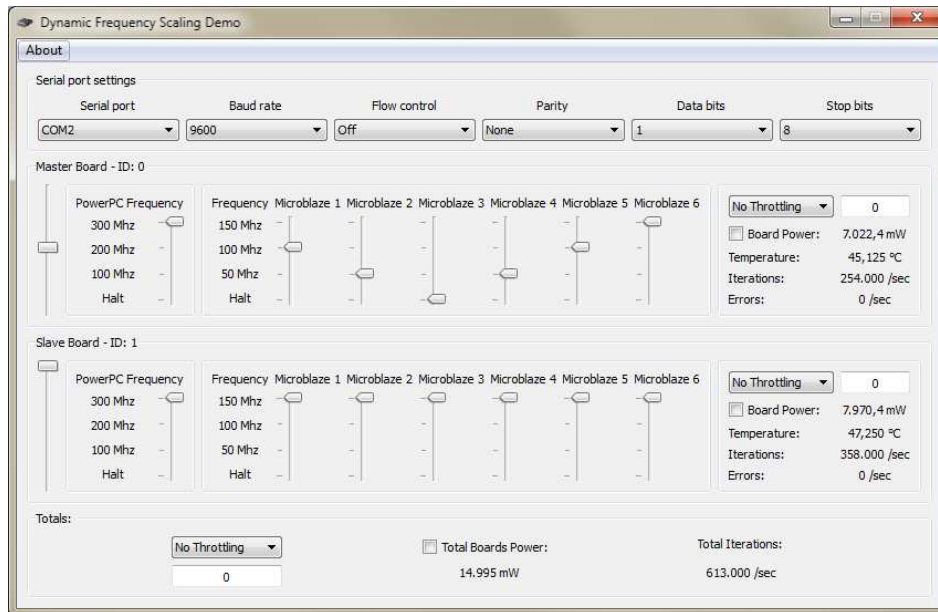
Η εικόνα 26 δείχνει την δομή ενός μηνύματος startofframe αναλύοντας το σε δυαδική μορφή. Τα δύο byte από αριστερά είναι πάντα σταθερά, ενώ το μεσαίο από δεξιά ορίζει τον παραλήπτη του πακέτου. Τέλος, το λιγότερο σημαντικό byte είναι ένα σύνολο από bit που δείχνουν τι είδους πληροφορία παρουσιάζουν και πόσα είναι τα πακέτα που ακολουθούν. Στο συγκεκριμένο παράδειγμα ο παραλήπτης του πακέτου είναι ο κόμβος με τιμή διεύθυνσης 1 και ακολουθούν ακόμα 3 πακέτα, ο αποστολέας, η ενέργεια που καταναλώνει και η θερμοκρασία της FPGA. Ο συγκεκριμένος μηχανισμός επικοινωνίας είναι πολύ αποδοτικός αφού μόλις με ένα μήνυμα εκκίνησης μπορούμε να μεταφέρουμε όλη τη πληροφορία θέλουμε, αξιόπιστα και χωρίς τη χρήση μηνυμάτων αναγνώρισης. Σε περίπτωση σφάλματος αν μεταφερθούν περισσότερα πακέτα από όσα περιμένουμε, τότε αυτά απορρίπτονται, αν λάβουμε λιγότερα τότε γίνεται επανεκκίνηση της επικοινωνίας χωρίς να μεταδίδονται εσφαλμένες μετρήσεις.

### 6.5 Λογισμικό διεπαφής

Όλες οι μετρήσεις και τα στατιστικά του δικτύου καταλήγουν σε ένα τερματικό υπολογιστή, μέσω της σειριακής θύρας RS-232 και τη σύνδεση της σε ένα από τους κόμβους. Εκεί εκτελείται μια διεπαφή εμφάνισης των αποτελεσμάτων στον τελικό χρήστη, όπου παρέχονται τα στοιχεία για τον έλεγχο των κόμβων. Η εφαρμογή είναι προγραμματισμένη στην αντικειμενοστραφή γλώσσα C++, υπό το framework Qt της Nokia κατά τέτοιο τρόπο ώστε να μπορεί να χρησιμοποιηθεί σε πολλά λειτουργικά συστήματα όπως Windows, MacOSX και Linux.

Οι δυνατότητες της διεπαφής επιτρέπουν στο χρήστη τον πλήρη έλεγχο του δικτύου, καθορίζοντας τη συχνότητα χρονισμού κάθε επεξεργαστή στο σύστημα, καθώς και την επίβλεψη των τιμών θερμοκρασίας, κατανάλωσης και επαναλήψεων/λαθών MD5 ανά δευτερόλεπτο. Στην εικόνα 27 μπορούμε να διακρίνουμε τις τρεις κύριες περιοχές της διεπαφής σε ένα δίκτυο δύο κόμβων ML405:

1. Serialport settings: Υπεύθυνη για της ρυθμίσεις της θύρας RS-232 και την έναρξη της επικοινωνίας.
2. Περιοχή Board: Για κάθε κόμβο στο δίκτυο δημιουργείται ένας χώρος με στοιχεία slider για τον έλεγχο των συχνοτήτων χρονισμού, καθώς και στοιχεία label για την εμφάνιση των μετρήσεων.
3. Totals: Η τελευταία περιοχή εμφανίζει το άθροισμα των επαναλήψεων MD5 καθώς και τη συνολική κατανάλωση του δικτύου. Όπως και σε κάθε χώρο board παρέχεται τρόπος αυτόματης ρύθμισης (throttling) κατανάλωσης ή απόδοσης του δικτύου βάση επιλεγμένης τιμής.



Εικόνα 28: Έλεγχος των κόμβων του δικτύου μέσω της διεπαφής

## 6.6 Ανάλυση λειτουργίας διεπαφής

Η πρώτη λειτουργία της εφαρμογής είναι η αρχικοποίηση και η ρύθμιση της επιλεγμένης σειριακής θύρας. Αμέσως γίνεται η παρακολούθηση κίνησης δεδομένων στη σειριακή και αναμένουμε μετρήσεις από κάποιον συνδεδεμένο κόμβο. Σε αυτό το στάδιο το μόνο που εμφανίζεται είναι η περιοχή «Serialportsettings» αφού ακόμα δε γνωρίζουμε πόσοι κόμβοι βρίσκονται στο δίκτυο, καθώς δεν έχει πραγματοποιηθεί η διευθυνσιοδότηση.

Στην έναρξη λειτουργίας του δικτύου όλοι οι κόμβοι δρουν ανεξάρτητα και δεν υπάρχει επικοινωνία μεταξύ τους. Για να εδραιωθεί η λειτουργία του δικτύου, πρέπει κάποιος από τους κόμβους να λάβει ένα πακέτο διευθυνσιοδότησης μέσω της σειριακής θύρας. Το πακέτο διευθυνσιοδότησης δημιουργείται από τη διεπαφή και θέτει το συνδεδεμένο κόμβο σε λειτουργία master. Στο επόμενο στάδιο ο masterκόμβος προωθεί το μήνυμα αυτό μέσω ενός καναλιού Αυτοακαί περιμένει το συνολικό αριθμό κόμβων από το άλλο διαθέσιμο κανάλι. Με την τοπολογία δακτυλίου όλοι οι κόμβοι παραλαμβάνουν το πακέτο διευθυνσιοδότησης και ξεκινάει η λειτουργία του δικτύου.

Αφού γίνει η χαρτογράφηση του δικτύου, η διεπαφή δημιουργεί τις περιοχές Board και Totals και εμφανίζει όλες τις μετρήσεις και τα στατιστικά του κάθε κόμβου. Σε αυτό το σημείο ο χρήστης μπορεί να επέμβει και να τροποποιήσει οποιαδήποτε συχνότητα επεξεργαστή επιλέξει. Μόλις γίνει κάποια αλλαγή η διεπαφή ενημερώνει το masterκόμβο και αυτός με τη σειρά του προωθεί τις επιλεγμένες συχνότητες στους προορισμένους κόμβους. Σχεδόν άμεσα, ο masterκόμβος ενημερώνεται μέσω Αυτοακαί για τις αλλαγές στη θερμοκρασία και την κατανάλωση και με τη σειρά του, προωθεί τα αποτελέσματα στη διεπαφή.

Μέσω της διεπαφής μπορούμε να εισάγουμε μια επιθυμητή τιμή κατανάλωσης ή επαναλήψεων MD5 και να γίνει αυτόματη ρύθμιση στη συχνότητα των επεξεργαστών, γύρω από τη συγκεκριμένη τιμή. Τέλος, υπάρχει η επιλογή εμφάνισης της κατανάλωσης του κάθε κόμβου, σαν διαφορά από την κατανάλωση σε λειτουργία ηρεμίας (idle). Με αυτόν τον τρόπο μπορούμε να δούμε ξεχωριστά την ακριβή κατανάλωση ενέργειας κάθε πυρήνα, αφαιρώντας τη σταθερή επιβάρυνση που έχει όλο το αναπτυσσόμενο ML405.

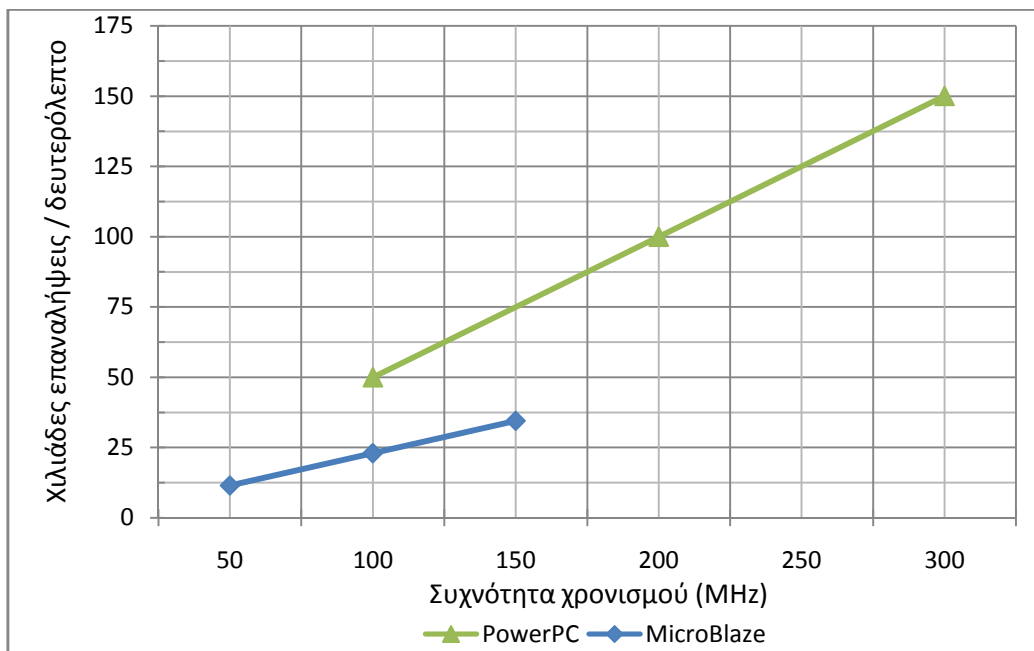
## 7 Μετρήσεις και συμπεράσματα

Ολοκληρώνοντας την πτυχιακή εργασία, σε αυτό το κεφάλαιο παρουσιάζουμε όλα τα μετρήσιμα δεδομένα, καθώς και τις διάφορες σχέσεις που προκύπτουν μεταξύ τους. Ακολουθούν τα συμπεράσματα και τα αποτελέσματα των μετρήσεων, καθώς και πιθανές επεκτάσεις που μπορούν να προκύψουν από περαιτέρω εργασία πάνω στο θέμα.

### 7.1 Ταχύτητα επεξεργαστών

Όπως είχαμε αναφέρει σε προηγούμενο κεφάλαιο, ο επεξεργαστής PowerPC λόγω της αρχιτεκτονικής του, αλλά και της υλοποίησης του σε τεχνολογία ASIC, μπορεί να επεξεργαστεί δεδομένα γρηγορότερα αλλά και να χρονιστεί σε μεγαλύτερη συχνότητα σε σχέση με έναν πυρήνα MicroBlaze. Πιο συγκεκριμένα, η μέγιστη συχνότητα χρονισμού του PowerPC είναι διπλάσια από ότι του MicroBlaze, 300 έναντι 150 MHz αντίστοιχα. Ακόμα όμως και σε καταστάσεις ίδιας συχνότητας χρονισμού, ο PowerPC είναι τουλάχιστον δύο φορές πιο γρήγορος στην εκτέλεση του αλγόριθμου MD5.

Πίνακας 2: Διάγραμμα MD5 επαναλήψεων - συχνότητας επεξεργαστών



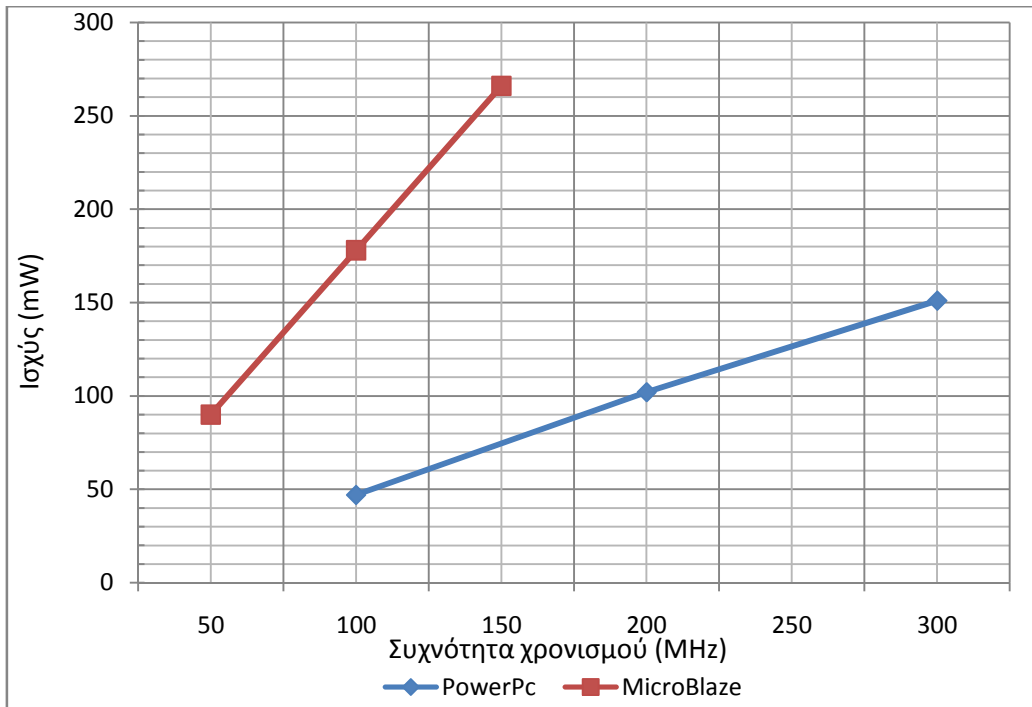
Στον πίνακα 2 βλέπουμε τη σχέση MD5 επαναλήψεων/συχνότητας για του επεξεργαστές PowerPC και MicroBlaze. Μπορεί να παρατηρήσει κανείς ότι και για τα δύο είδη επεξεργαστών, η αύξηση της συχνότητας χρονισμού φέρει ανάλογη αύξηση στις MD5 επαναλήψεις ανά δευτερόλεπτο. Η σχέση τους είναι άκρος γραμμική, και αργότερα θα δούμε ακριβώς την ίδια τάση και για τα υπόλοιπα μετρήσιμα χαρακτηριστικά.

### 7.2 Κατανάλωση ενέργειας

Διαφορετική αρχιτεκτονική και τρόπος κατασκευής, σχεδόν πάντα σημαίνει και διαφορετική κατανάλωση ενέργειας. Ο PowerPC δεν είναι μόνο πιο γρήγορος επεξεργαστής σε σχέση με το MicroBlaze, αλλά είναι και πιο αποδοτικός αφού σε ίδιες συνθήκες λειτουργίας η κατανάλωση του είναι σχεδόν τέσσερις φορές μικρότερη.

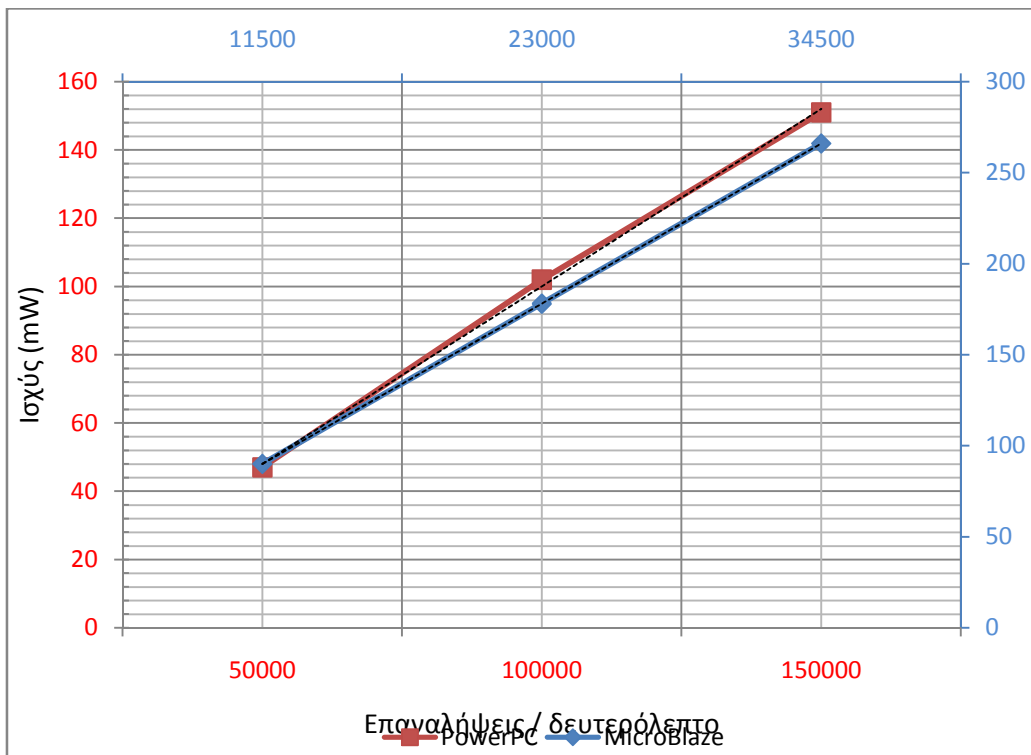


Πίνακας 3: Διάγραμμα ισχύος - συχνότητας επεξεργαστών



Όπως βλέπουμε στον πίνακα 3 η γραμμική σχέση παραμένει ακόμα και για την ισχύ. Κάθε αύξηση στη συχνότητα χρονισμού, ανεβάζει ανάλογα το ενεργειακό κόστος επεξεργασίας. Εφόσον λοιπόν οι σχέσεις συχνότητας - ισχύος και συχνότητας - επαναλήψεων είναι γραμμικές τότε και η σχέση ισχύος - επαναλήψεων θα είναι γραμμική.

Πίνακας 4: Διάγραμμα ισχύος - επαναλήψεων



Αξίζει να σημειώσουμε ότι η γραμμικότητα συνεχίζει να υφίσταται ακόμα και με άθροισμα όμοιων επεξεργαστών. Τρεις πυρήνες MicroBlaze στα 150MHz συμπεριφέρονται σαν ένα θεωρητικό πυρήνα στα 450 MHz, καταλαλώντας την ίδια ενέργεια και αποδίδοντας το ίδιο.

Πίνακας 5: Κατανάλωση ενέργειας διαφορετικών στοιχείων και καταστάσεων

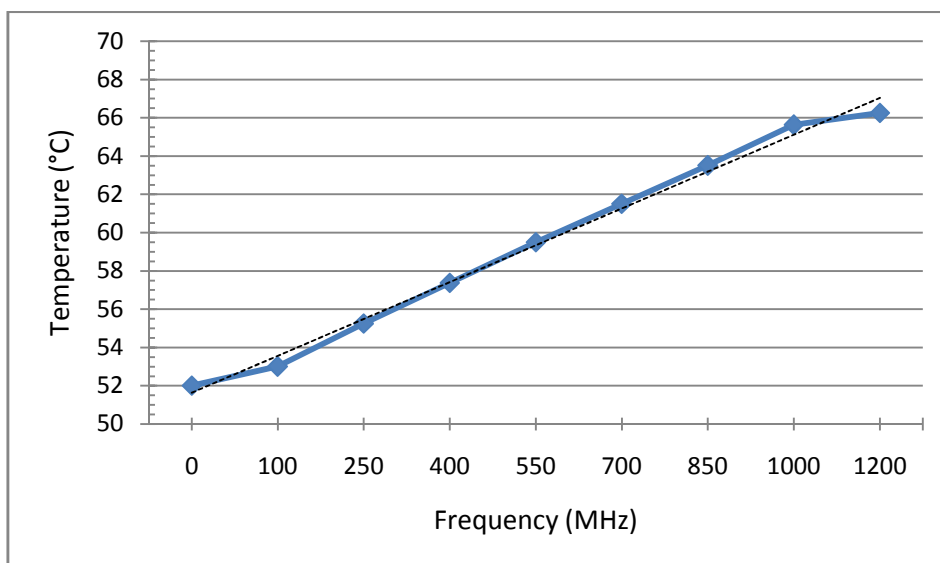
Στοιχείο	Κατανάλωση (mW)	Στοιχείο	Κατανάλωση(mW)
ML405 (blank):	3.400	MicroBlaze (50 MHz):	90
ML405 (reset):	5.300	PLB & Peripherals:	300
ML405 (idle):	6.000	Aurora Channels:	60
ML405 (full throttle):	8.000	OPB Bus:	12
PowerPC (100 MHz):	45		

Τέλος, στον πίνακα 4 βλέπουμε πόση ενέργεια καταναλώνει το κάθε στοιχείο της αρχιτεκτονικής μας. Κάτι τέτοιο γίνεται εφικτό λόγω του αισθητήριου κατανάλωσης (με ακρίβεια έως και 1,2 mW) καθώς και με τα διάφορα περιφερειακά ελέγχου και αποκοπής των συχνοτήτων χρονισμού. Αποκόπτοντας τις συχνότητες σε διαφορετικά μέρη του συστήματος μπορούμε να μετρήσουμε με ακρίβεια τις ενεργειακές ανάγκες των επεξεργαστών, των διαύλων και των περιφερειακών. Συνολικά η κατανάλωση ενέργειας διαφέρει κατά 2.000 mW από κατάσταση ηρεμίας (idle) σε κατάσταση πλήρους επεξεργασίας (fullthrottle).

### 7.3 Θερμοκρασία συστήματος

Άλλος ένας μη επιθυμητός παράγοντας που εμφανίζεται με την αύξηση της συχνότητας είναι η θερμοκρασία. Το αναπτυξιακό ML405 δεν έχει κάποιο ενεργητικό σύστημα ψύξης και η μέγιστη θερμοκρασία που μπορεί να φτάσει η Virtex-4 FPGA χωρίς προβλήματα είναι 85°C. Το ευχάριστο είναι ότι στις δοκιμές μας δεν καταφέραμε να ξεπεράσουμε τους 67°C. Παρ' όλα αυτά είναι σημαντικό να καταγράψουμε την άνοδο της θερμοκρασίας σε σχέση με το φόρτο εργασίας του συστήματος, για να αποφύγουμε προβλήματα σε συστήματα που κάνουν χρήση μεγαλύτερων και γρηγορότερων FPGA, όπου ενδεχόμενος να έχουν περισσότερους πυρήνες επεξεργασίας ή/και μεγαλύτερη συχνότητα χρονισμού.

Πίνακας 6: Διάγραμμα θερμοκρασίας - συχνότητας αναπτυξιακού ML405



Η σχέση θερμοκρασίας - συχνότητας είναι και πάλι γραμμική. Στον πίνακα 6 φαίνεται η αύξηση της θερμοκρασίας σε όλο το εύρος των συχνοτήτων χρονισμού στο σύστημα. Η γραμμικότητα αλλιώνεται στην αρχή και στο τέλος της γραφικής παράστασης λόγω του επεξεργαστή PowerPC, όπου αυξάνει τη θερμοκρασία του συστήματος κατά 2°C στα 300MHz, έναντι των MicroBlazeόπου έχουν την ίδια αύξηση στα 150MHz.

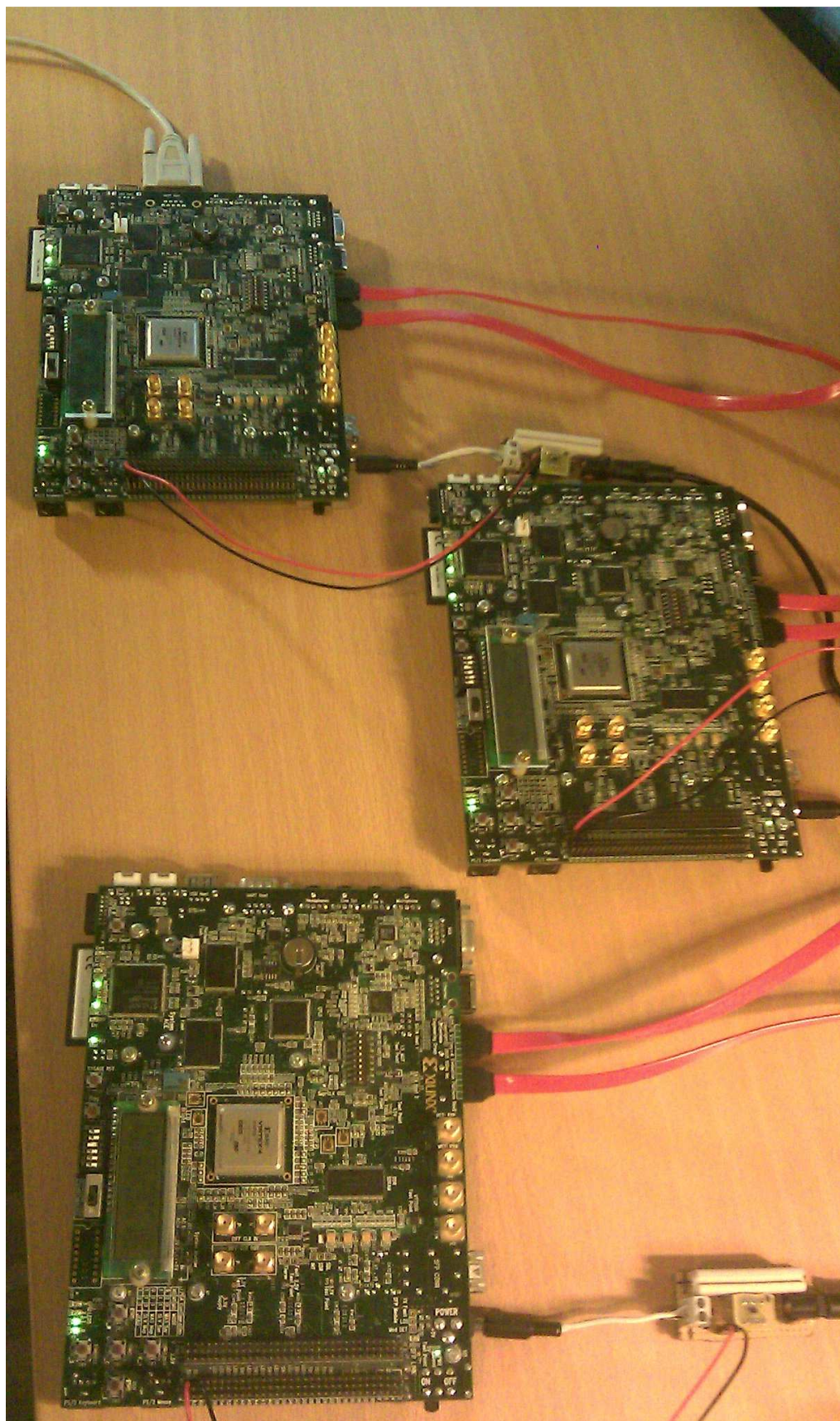
#### **7.4 Συμπεράσματα, μελλοντική εργασία και επεκτάσεις**

Μέσα από αυτήν την πτυχιακή εργασία δείξαμε πως η δυναμική τροποποίηση συχνότητας χρονισμού προσφέρει άμεσα ενεργειακά κέρδη. Με τη μέθοδο αυτή, καταφέραμε να εξοικονομήσουμε μέχρι και 2.000 mW, ποσό που επηρεάζει άμεσα την ενεργειακή αυτονομία ενός κινητού ενσωματωμένου συστήματος. Ταυτόχρονα, όταν η θερμοκρασία φτάσει κρίσιμες τιμές, το σύστημα είναι σε θέση να ελαττώσει τους ρυθμούς επεξεργασίας και να επανέλθει σε φυσιολογικά επίπεδα. Το σημαντικότερο όμως είναι ότι δεν είμαστε περιορισμένοι σε μια συγκεκριμένη συχνότητα και ανάλογα το σενάριο λειτουργίας το σύστημα μπορεί να αυτό-ρυθμιστεί στις απαιτούμενες ενεργειακές ανάγκες.

Η κατασκευή ενός αισθητηρίου ισχύος υψηλής ακρίβειας, μας έδωσε την ικανότητα να ερευνήσουμε και να μετρήσουμε το ενεργειακό προφίλ κάθε στοιχείου πάνω στην FPGA και το αναπτυξιακό ML405. Γνωρίζοντας ακριβώς πόσο επιβαρύνει το σύστημα η κάθε συσκευή, μπορούμε να επιλέξουμε την παύση ορισμένων, για την άμεση εξοικονόμηση ενέργειας. Η παρακολούθηση όλων αυτών των χαρακτηριστικών (θερμοκρασίας, συχνότητας, κατανάλωσης) σε πραγματικό χρόνο, μας επιτρέπει την πλήρη και δυναμική διαχείριση του συστήματος. Τέλος, η ζεύξη πολλών ενσωματωμένων συστημάτων, κάνει το σύστημα άμεσα επεκτάσιμο και διαλειτουργικό, προσφέροντας τη δυνατότητα συγκέντρωσης και διαχείρισης πόρων σε ένα γρήγορο και απλό δίκτυο.

Το θέμα της πτυχιακής εργασίας καλύπτει μεγάλη έκταση, λογικό είναι λοιπόν να υπάρχουν αμέτρητα θέματα επέκτασης της. Σε μελλοντική εργασία θα μπορούσε να επιτευχθεί ακόμα πιο δυναμική μέθοδος τροποποίησης των συχνοτήτων χρονισμού. Με το μηχανισμό «μερικής αναδιάθρωσης» (partialreconfiguration)<sup>35</sup> στην FPGA, θα μπορούσαμε να αποφύγουμε τους περιορισμούς επιλογής ανάμεσα σε μόνο τρεις συχνότητες χρονισμού. Από την άποψη του δικτύου θα μπορούσε να προγραμματιστεί ένα πιο ανεπτυγμένο και πολύπλοκο πρωτόκολλο επικοινωνίας, βελτιστοποιημένο για τη μετάδοση μεγάλου όγκου δεδομένων, χρησιμοποιώντας τοπολογία διπλού δακτυλίου. Στο κομμάτι της επεξεργασίας θα μπορούσαν να υλοποιηθούν άλλοι γνωστοί αλγόριθμοι (AES, DES, κτλ) και να ερευνηθεί η απόδοσή τους μέσα στη αρχιτεκτονική του συστήματος.

Κλείνοντας την εργασία, θα θέλαμε να παρακινήσουμε τον ενδιαφερόμενο αναγνώστη, να συμβουλευτεί ενδελεχώς τον πηγαίο κώδικα και τα σχόλια των εφαρμογών, καθώς και τα ανάλογα φύλλα δεδομένων (datasheets) των κατασκευαστών που επισυνάπτονται, για την άμεση κατανόηση αυτού του μεγάλου σε έκταση θέματος.



Εικόνα 29: Δίκτυο ενσωματωμένων συστημάτων τριών κόμβων



## Παράρτημα - Βιβλιογραφία

- <sup>1</sup>Wainner, Scott; Robert Richmond (2003). [The Book of Overclocking](#). pp. 1-2. ISBN 188641176X.
- <sup>2</sup>Kozierok, Charles M. (2001). [Risks of Overclocking the Processor](#).
- <sup>3</sup>Intel, (2008). [First the Tick, Now the Tock: Next Generation Intel® Microarchitecture \(Nehalem\)](#).
- <sup>4</sup>Xilinx, (2008) [ug210 - ML405 Evaluation Platform](#).
- <sup>5</sup>Maxim, (2003) [MAX6653-MAX6664 - Temperature Monitors and PWM Fan Controllers](#).
- <sup>6</sup>Texas Instruments, (2010). [Zero-Drift, Bi-directional current/power monitor with I<sup>2</sup>C interface](#). pp. 1
- <sup>7</sup>Xilinx, (2010). [ug631 - ISE Design Suite 11.5](#)
- <sup>8</sup>Nokia, (2009). [Qt 4.6 whitepaper](#).
- <sup>9</sup>Xilinx, (2007). [ds112 - Virtex-4 Overview](#).
- <sup>10</sup>Xilinx. (2010). [ug011 - PowerPC Processor Reference Guide](#).
- <sup>11</sup>Intel, (2009). [Reducing Interrupt Latency Through the Use of Message Signaled Interrupts](#).
- <sup>12</sup>Xilinx, (2004). [MicroBlaze Processor Reference Guide](#).
- <sup>13</sup>IBM, (2006). [IBM CoreConnect bus cores](#).
- <sup>14</sup>Xilinx, (2010). [Processor Local Bus v4.6](#).
- <sup>15</sup>Xilinx, (2010). [ds402 - On-Chip Peripheral Bus with OPB Arbiter](#).
- <sup>16</sup>Xilinx, (2010). [ds449 - Fast Simplex Link v2.0 Bus](#).
- <sup>17</sup>Philips, (2010). [The I2C-Bus Specification v2.1](#).
- <sup>18</sup>Xilinx, (2010). [ds606 - XPS IIC Bus Interface](#).
- <sup>19</sup>Hitachi, (1998). [HD44780U Dot Matrix Liquid Crystal Display Controller/Driver](#).
- <sup>20</sup>Xilinx, (2009). [ds486 - PLB General Purpose Input/Output \(GPIO\)](#).
- <sup>21</sup>Xilinx, (2009). [ug061 - Aurora 8B/10B v3.1 for Virtex-4 FPGA User Guide](#).
- <sup>22</sup>Xilinx, (2010). [sp002 - Aurora 8B/10B Protocol Specification](#).
- <sup>23</sup>Xilinx, (2009). [ds128 - Aurora 8B-10B v3.1 for Virtex-4 FPGA](#).
- <sup>24</sup>Xilinx, (2009). [ug076 - Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide](#).
- <sup>25</sup>Xilinx, (2010). [ds614 - Clock Generator](#).
- <sup>26</sup>Xilinx, (2010). [ds571 - XPS UART Lite](#).
- <sup>27</sup>Dallas Semiconductor, (1998). [Application Note 83 - Fundamentals of RS232 Serial Communications](#).
- <sup>28</sup>Xilinx, (2008). [ug070 - Virtex-4 FPGA User Guide](#). pp. 27-37.
- <sup>29</sup>Xilinx, (2008). [ug018 - PowerPC 405 Processor Block Reference Guide](#). pp. 35-39.
- <sup>30</sup>Maxim, (2001). [AN746 - High-Side Current-Sense Measurement: Circuits and Principles](#).
- <sup>31</sup>Miller, Freeman; UltraCAD Design, (1998). [Temperature Rise in PCB Traces](#).
- <sup>32</sup>Texas Instruments, (2010). [Zero-Drift, Bi-directional current/power monitor](#). pp 17-22
- <sup>33</sup>Xilinx, (2006). [Xilkernel](#).
- <sup>34</sup>Network Working Group; R. Rivest, (1992). [The MD5 Message-Digest Algorithm](#).
- <sup>35</sup>Xilinx, (2010). [Partial Reconfiguration User Guide](#).