



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΦΥΣΙΚΗΣ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΦΥΣΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΕΙΔΙΚΕΥΣΗΣ ΣΤΟΝ
ΗΛΕΚΤΡΟΝΙΚΟ ΑΥΤΟΜΑΤΙΣΜΟ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**«Ο επεξεργαστής Microblaze και οι δυνατότητές του για
επεξεργασία δεδομένων και περιφερειακές συνδέσεις»**

ΝΙΚΟΛΑΟΣ ΣΠΥΡΙΔΑΚΗΣ
2017524

ΕΠΙΒΛΕΠΩΝ: ΔΙΟΝΥΣΙΟΣ ΡΕΪΣΗΣ, ΚΑΘΗΓΗΤΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ

ΑΘΗΝΑ, 2020

Περίληψη

Οι απαιτήσεις των εφαρμογών αυξάνονται ραγδαία με την πάροδο του χρόνου. Έτσι ο χρήστης θα πρέπει να αλλάξει τα στοιχεία που χρησιμοποιεί ή να αλλάξει τον τρέχον σχεδιασμό του ώστε να ανταποκρίνεται στις απαιτήσεις που προέκυψαν. Αυτό είναι χρονοβόρο και δαπανηρό, οπότε οι χρήστες ψάχνουν λύσεις που να παρέχουν ευελιξία ως προς τον σχεδιασμό τους.

Μια λύση είναι οι soft processor, όπως ο Microblaze της Xilinx τον οποίο θα χρησιμοποιήσουμε σε αυτή την διπλωματική. Ο Microblaze μπορεί να υλοποιηθεί σε πολλές εναλλακτικές μορφές και μπορεί να χρησιμοποιηθεί ως microcontroller ή ως ενσωματωμένος επεξεργαστής πάνω στα FPGAs. Σε αυτή την εργασία θα δούμε τον Microblaze ως επεξεργαστή, τις συνδέσεις που μπορεί να πραγματοποιήσει με τα περιφερειακά του έτσι ώστε να θεωρηθεί ένα ενσωματωμένο σύστημα ή ένα SoC, καθώς και τις δυνατές διαμορφώσεις για επεξεργασία δεδομένων.

Περιεχόμενα

| | |
|--|----|
| 1 Εισαγωγή | 5 |
| 2 Microblaze | 6 |
| 2.1 Εισαγωγή | 6 |
| 2.2 Endianness..... | 7 |
| 2.3 Εντολές..... | 7 |
| 2.4 Καταχωρητές..... | 8 |
| 2.5 Pipeline..... | 11 |
| 2.6 Αρχιτεκτονική Μνήμης..... | 12 |
| 2.7 Διαχείριση Εικονικής Μνήμης..... | 15 |
| 2.8 Reset, Interrupts, Breaks and Exceptions | 16 |
| 2.8.1 Reset..... | 17 |
| 2.8.2 Hardware Exceptions..... | 17 |
| 2.8.3 Breaks..... | 19 |
| 2.8.4 Interrupt..... | 19 |
| 2.9 Interfaces..... | 20 |
| 3 Interconnections and Interfaces..... | 21 |
| 3.1 AXI Interconnection Interface..... | 22 |
| 3.2 PLB..... | 24 |
| 3.3 LMB..... | 25 |
| 3.4 FSL..... | 26 |
| 3.5 XCL..... | 26 |
| 3.6 Peripherals..... | 26 |

| | |
|---|----|
| 3.6.1 Bus & Bridges..... | 27 |
| 3.6.2 Clock , Reset & Interrupt..... | 27 |
| 3.6.3 UARTs..... | 27 |
| 3.6.4 DMA..... | 28 |
| 3.6.5 Memory & Memory Controllers..... | 29 |
| 3.6.6 GPIO..... | 30 |
| 4 Configuration..... | 31 |
| 4.1 General Configuration..... | 32 |
| 4.2 Minimum Area and Maximum Performance Configuration..... | 33 |

1. Εισαγωγή

Τα FPGAs (Field Programmable Gate Arrays) είναι ολοκληρωμένα κυκλώματα τα οποία μπορούν να παραμετροποιηθούν ώστε να μπορούν να υλοποιήσουν πολλών και διαφορετικών ειδών ψηφιακά κυκλώματα μετά την παρασκευή τους από κάποια εταιρία υλικού. Τα FPGAs είναι ημιαγωγικές διατάξεις που διαθέτουν ένα πίνακα από προγραμματίσιμα λογικά κομμάτια τα οποία υλοποιούν στο υλικό συναρτήσεις και μια ιεραρχία από επαναδιαχειρίσιμες διασυνδέσεις που επιτρέπουν στα επιμέρους κομμάτια να συνδέονται μεταξύ τους, προσφέροντας υψηλά επίπεδα ευελιξίας.

Με την εξέλιξη της τεχνολογίας η υψηλή κλίμακας ολοκλήρωση των ψηφιακών συστημάτων (Very Large Scale of Integration) επέτρεψε στα ολοκληρωμένα κυκλώματα να μπορούν να υποστηρίξουν ακόμα και ολόκληρα συστήματα (Systems on Chip) για μια πληθώρα εφαρμογών σχετιζόμενων με τις τηλεπικοινωνίες ή του αυτομάτου ελέγχου. Στις περισσότερες από αυτές τις περιπτώσεις σε ένα τέτοιο σύστημα υπάρχει απαίτηση για την ύπαρξη ενός επεξεργαστή για τον έλεγχο των λειτουργιών του συστήματος και την ανταπόκριση του σε αλλαγές των εξωτερικών συνθηκών.

Οι πρώτες υλοποιήσεις επεξεργαστών σε FPGA ήταν hard processors, δηλαδή ο επεξεργαστής ήταν υλοποιημένος στο ολοκληρωμένο και ο σχεδιαστής του ψηφιακού κυκλώματος δεν είχε παρά να συνδέσει το υπόλοιπο σύστημα στον επεξεργαστή ώστε να μπορέσει να εκμεταλλευτεί τις δυνατότητες του.

Η σύγχρονη τεχνολογία μας επιτρέπει την υλοποίηση επεξεργαστών στα LUTs (Look Up Tables) των FPGAs. Τέτοιου είδους επεξεργαστές ονομάζονται soft processors και μέσω αυτών δίνεται η δυνατότητα χρήσης όλου του ολοκληρωμένου κυκλώματος από συστήματα που δεν χρειάζονται έναν επεξεργαστή αλλά κυρίως δίνοντας την δυνατότητα στο σχεδιαστή που θα χρησιμοποιήσει έναν soft processor επεξεργαστή να παραμετροποιήσει και να συμπεριλάβει μόνο τα στοιχεία του επεξεργαστή τα οποία είναι απαραίτητα.

Στην παρούσα διπλωματική θα γίνει θα περιγραφεί ο soft processor Microblaze, που προσφέρεται ως IP (Intellectual Property) από την Xilinx για υλοποίηση στα ολοκληρωμένα κυκλωματά της. Θα γίνει αναφορά στα απαραίτητα περιφερειακά που πρέπει να περιέχει ένα σύστημα το οποίο χτίζεται γύρω από τον εν λόγω επεξεργαστή, καθώς και τον τρόπο με τον οποίο μπορεί κανείς να επεκτείνει το σύστημα υλοποιώντας καινούργια περιφερειακά κατάλληλα να επικοινωνούν με τον επεξεργαστή.

Τέλος, θα γίνει κάποια αναφορά στην υλοποίηση εφαρμογών οι οποίες θα εκτελούνται στον επεξεργαστή κάνοντας χρήση κάποιων παρεχόμενων βιβλιοθηκών, προγράμματα οδήγησης για τα περιφερειακά του επεξεργαστή αλλά και την δημιουργία κάποιων προγραμμάτων οδήγησης που μπορεί να υλοποιήσει κανείς για συγκεκριμένο σκοπό.

2. MicroBlaze

2.1 Εισαγωγή

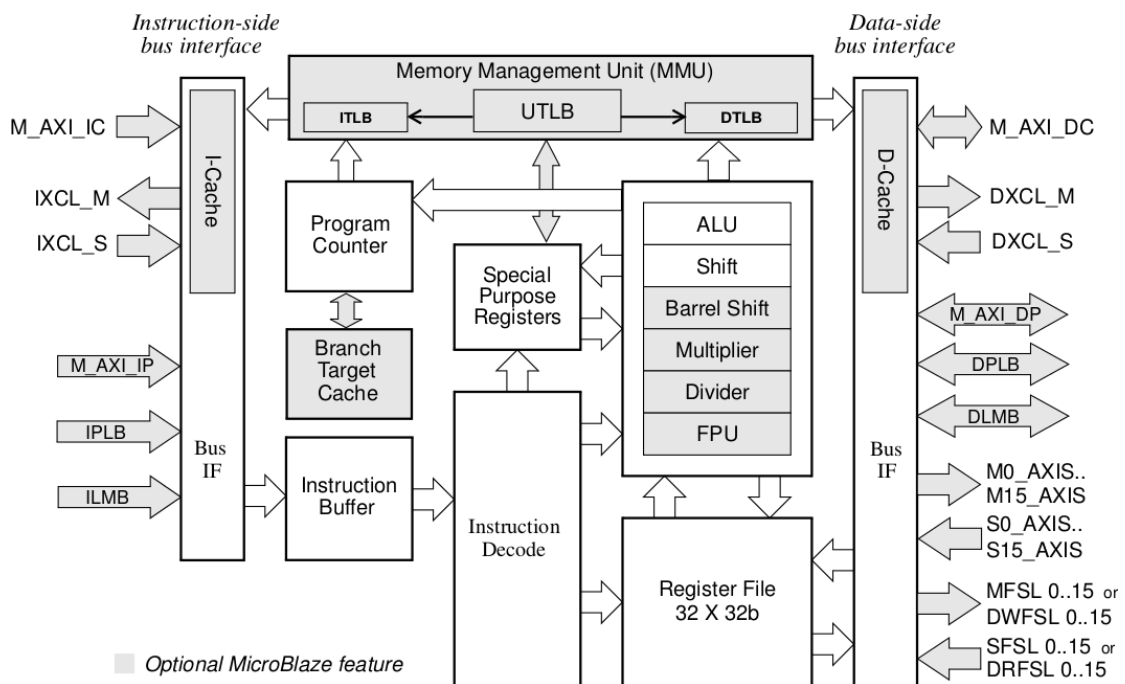
Ο Microblaze είναι ένας επεξεργαστής ο οποίος είναι σχεδιασμένος από την εταιρία υλικού Xilinx και χρησιμοποιείται σαν IP core για χρήση στα FPGAs. Λίγο πιο συγκεκριμένα τα IP cores είναι κωδικώς γραμμένος σε κάποια γλώσσα περιγραφής υλικού (VHDL , Verilog) που υλοποιεί ένα συγκεκριμένο σύστημα και κάποια συγκεκριμένη λειτουργία, χωρίς κάποιος χρήστης να μπορεί να παρέμβει σε αυτόν. Χρησιμοποιείται όπως ακριβώς υπάρχει στις βιβλιοθήκες του εκάστοτε εργαλείο ανάπτυξης και υλοποίησης των εφαρμογών (Vivado , ISE της Xilinx) σε συνδυασμό με το Xilinx SDK υλοποιώντας ένα SoC με αρκετά υψηλή απόδοση

Τα βασικά χαρακτηριστικά του Microblaze είναι:

- Είναι Soft Core, υλοποιείται από πύλες και τα LUTs του FPGA.
- Είναι παραμετροποιήσιμος σε πολύ μεγάλο βαθμό.
- Διαθέτει 32 bit αρχιτεκτονική συνόλου εντολών.

- Ανήκει στην κατηγορία των RISC (Reduced Instruction Set Architecture) επεξεργαστών, διαθέτει μικρό σύνολο εντολών (Instruction Set).
- Έχει τρία ή πέντε στάδια σωλήνωσης (pipeline).

Στην συνέχεια παρατίθεται το block διάγραμμα του επεξεργαστή. Τα στοιχεία που είναι σημειωμένα με γκρί είναι προαιρετικά και μπορούν να αποτελέσουν κομμάτι του επεξεργαστή κάνοντας τις κατάλληλες επιλογές κατά την υλοποίηση του.



2.1 : Block Diagram Microblaze

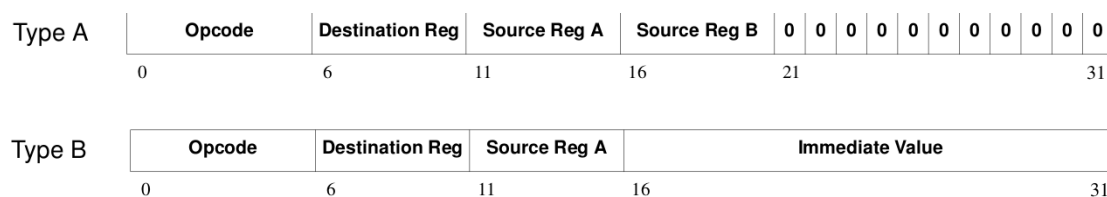
2.2 Endianness

Ο Microblaze υποστηρίζει κατά κύριο λόγο Big-Endian, ο τρόπος με τον οποίο αποθηκεύεται το περισσότερο σημαντικό bit (MSB), αν αποθηκεύεται

πρώτο η τελευταίο. Ανάλογα με το bus το οποίο θα χρησιμοποιήσει ο σχεδιαστής για να συνδέσει τον επεξεργαστή με τα περιφερειακά του, θα πρέπει να χρησιμοποιηθεί η κατάλληλη σειρά στην ανάγνωση των bytes. Η Xilinx χρησιμοποιεί το PLB (Processor Local Bus) που αποτελεί υλοποίηση του Local Bus της IBM και του AXI, καθώς και του AMBA που χρησιμοποιείται περισσότερο στους ARM επεξεργαστές. Η επιλογή ,ανάλογως με το Bus Interface που χρησιμοποιεί ο σχεδιαστής μπορεί να γίνει μέσω της παραμετροποίησης της παραμέτρου C_ENDIANNESS, όταν υλοποιώ τον επεξεργαστή.

2.3 Εντολές

Οι εντολές του Microblaze είναι 32-bit και χωρίζονται σε δυο τύπους, Type A και Type B. Οι Type A εντολές δέχονται ως ορίσματα δυο source registers και έναν destination register και είναι παρόμοιες με τις Type R εντολές της MIPS αρχιτεκτονικής που ορίστηκε από τους Patterson και Hennesey στο πανεπιστήμιο του Berkeley. Αντίθετα οι εντολές Type B δέχονται έναν source register και έναν 16-bit τελεστή, αντίστοιχες με τις Type I εντολές του MIPS. Η δομή των εντολών έχει ως εξής:



2.2 : Instruction Types

2.4 Καταχωρητές

Από τα βασικότερα μέρη μιας αρχιτεκτονικής είναι οι καταχωρητές (registers) και η χρήση τους κατά τα στάδια υλοποίησης μια εντολής είτε αυτή είναι pipelined είτε όχι. Ο Microblaze διαθέτει 32 καταχωρητές γενικού σκοπού (general purpose) των 32-bit και μέχρι 18 καταχωρητές ειδικού σκοπού

(specific purpose) των 32-bit, ανάλογα πάντα με τις τιμές κάποιων παραμέτρων που επιλέγει ο σχεδιαστής κατά το στάδιο της υλοποίησης του επεξεργαστή.

Ακολουθεί ο πίνακας με τους καταχωρητές γενικού σκοπού:

| Bits | Name | Description | Reset Value |
|------|-----------------|---|-------------|
| 0:31 | R0 | Always has a value of zero. Anything written to R0 is discarded | 0x00000000 |
| 0:31 | R1 through R13 | 32-bit general purpose registers | - |
| 0:31 | R14 | 32-bit register used to store return addresses for interrupts. | - |
| 0:31 | R15 | 32-bit general purpose register. Recommended for storing return addresses for user vectors. | - |
| 0:31 | R16 | 32-bit register used to store return addresses for breaks. | - |
| 0:31 | R17 | If MicroBlaze is configured to support hardware exceptions, this register is loaded with the address of the instruction following the instruction causing the HW exception, except for exceptions in delay slots that use BTR instead (see “ Branch Target Register (BTR) ”); if not, it is a general purpose register. | - |
| 0:31 | R18 through R31 | R18 through R31 are 32-bit general purpose registers. | - |

2.3 : General Purpose Registers

Οι καταχωρητές ειδικού σκοπού είναι οι εξής:

- Program Counter (PC): περιέχει την διεύθυνση της επόμενης εντολής που θα εκτελεστεί.
- Machine Status Register (MSR): περιέχει κάποια flags για τον έλεγχο λειτουργίας του επεξεργαστή.
- Exception Address Register (EAR): περιέχει την διεύθυνση προσπέλασης της εντολής που προκάλεσε το exception.
- Exception Status Register (ESR): περιέχει κάποια flags που αφορούν τα exceptions.

- Branch Target Register (BTR): περιέχει την διεύθυνση διακλάδωσης στις εντολές delay που ακολουθούν την εντολή διακλάδωσης.
- Floating Point Status Register (FSR): περιέχει flags σχετιζόμενα με την μονάδα κινητής υποδιαστολής.
- Exception Data Register (EDR): περιέχει τα δεδομένα τα οποία προκάλεσαν εξαίρεση σε συνδέσεις ροών (FSL ή AXIStream).
- Stack Low Register (SLR): περιέχει την μικρότερη διεύθυνση του stack για τον έλεγχο του overflow.
- Stack High Register (SHR): περιέχει την μεγαλύτερη διεύθυνση του stack για τον έλεγχο του underflow.
- Process Identifier Register (PID): χρησιμοποιείται για να αναγνωρίσει μια διεργασία κατά την λειτουργία της MMU.
- Zone Protection Register (ZPR): χρησιμοποιείται για την παράκαμψη της προστασίας της μνήμης που ορίζεται στις TLB εγγραφές της MMU.
- Translation Look-Aside Buffer Low Register (TLBLO): χρησιμοποιείται για την πρόσβαση σε ULTB εγγραφές της MMU.
- Translation Look-Aside Buffer High Register (TLBHI): χρησιμοποιείται για την πρόσβαση σε ULTB εγγραφές της MMU.
- Translation Look-Aside Buffer Index Register (TLBX): χρησιμοποιείται ως δείκτης στον UTLB όταν χρησιμοποιούνται οι καταχωρητές TLBLO και TLBHI.
- Translation Look-Aside Buffer Search Index Register (TLBSX): χρησιμοποιείται κατά την αναζήτηση ενός αριθμού εικονικής σελίδας στον UTLB.
- Processor Version Register (PVR): μια σειρά από καταχωρητές που περιέχουν στοιχεία για τις λειτουργίες που υλοποιεί ο επεξεργαστής.

2.5 Pipeline

Οι εντολές του Microblaze είναι σωληνωμένες (pipelined). Κάθε στάδιο του pipeline έχει διάρκεια ενός κύκλου του ρολογιού. Άμεσο συμπέρασμα αυτού είναι συμπεράνουμε ότι ο συνολικός χρόνος εκτέλεσης των εντολών ισούται με τα στάδια του pipeline και κάθε εντολή ολοκληρώνεται σε κάθε ένα κύκλο του ρολογιού. Μερικές εντολές χρειάζονται παραπάνω κύκλους του ρολογιού.

Το πλήθος των σταδίων του pipeline μπορεί να παραμετροποιηθεί. Ως άμεση συνέπεια αυτού προκύπτει:

1. Το pipeline τριων σταδίων που περιέχει τα στάδια:

- Fetch (IF)
- Decode (OD)
- Execute (EX)

| | cycle1 | cycle2 | cycle3 | cycle4 | cycle5 | cycle6 | cycle7 |
|---------------|--------|--------|---------|---------|---------|---------|---------|
| instruction 1 | Fetch | Decode | Execute | | | | |
| instruction 2 | | Fetch | Decode | Execute | Execute | Execute | |
| instruction 3 | | | Fetch | Decode | Stall | Stall | Execute |

2.4 Three-staged Pipeline

Παρατηρούμε ότι η δεύτερη εντολή, για παράδειγμα, απαιτεί τρεις κύκλους του ρολογιού για την προσπέλαση στην μνήμη, οπότε έχουμε μια καθυστέρηση στην έναρξη της τρίτης εντολής.

2. Το pipeline πέντε σταδίων που περιέχει τα στάδια:

- Fetch (IF)
- Decode (OD)
- Execute (EX)
- Access Memory (MEM)
- Write Back (WB)

| | cycle1 | cycle2 | cycle3 | cycle4 | cycle5 | cycle6 | cycle7 | cycle8 | cycle9 |
|---------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| instruction 1 | IF | OF | EX | MEM | WB | | | | |
| instruction 2 | | IF | OF | EX | MEM | MEM | MEM | WB | |
| instruction 3 | | | IF | OF | EX | Stall | Stall | MEM | WB |

2.5 Five-staged Pipeline

Η δεύτερη εντολή απαιτεί τρεις κύκλους ρολογιού για την προσπέλαση της μνήμης οπότε το αντίστοιχο στάδιο της τρίτης εντολής καθυστερεί για δυο κύκλους, όπως φαίνεται στο σχήμα από πάνω.

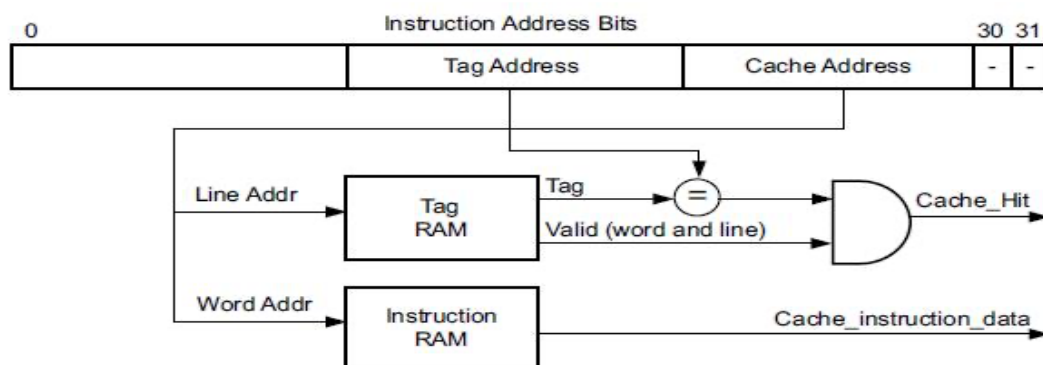
2.6 Αρχιτεκτονική Μνήμης

Ο Microblaze είναι σχεδιασμένος με βάση το μοντέλο αρχιτεκτονικής Harvard, δηλαδή υπάρχει ξεχωριστή μνήμη και για τις εντολές και ξεχωριστή μνήμη για τα δεδομένα με διαφορετικές συνδέσεις του καθενός από αυτά με την CPU. Ο επεξεργαστής μπορεί να εκτελέσει μια εντολή σε ένα κύκλο του ρολογιού με κατάλληλες στρατηγικές pipelining. Σε φυσική υλοποίηση, δηλαδή υλοποίηση στα πλαίσια ενός ολοκληρωμένου κυκλώματος ή ενός SoC (System on Chip), οι δυο περιοχές μνήμης μπορούν να επικαλύπτονται και έτσι να βρίσκονται στο ίδιο φυσικό μέσο αποθήκευσης. Ο επεξεργαστής δεν διαχωρίζει φυσικά την πρόσβαση στην είσοδο και στην έξοδο από την πρόσβαση στην μνήμη και χρησιμοποιείται memory mapped I/O. Έτσι για την πρόσβαση στην μνήμη χρησιμοποιούμε τρία διαφορετικά interfaces:

- AXI4 (Advanced eXtensible Interface) ή XCL (Xilinx CacheLink).
- AXI4 (Advanced eXtensible Interface) ή PLB (Processor Local Bus).
- LMB (Local Memory Bus).

Επίσης δίνει στον χρήστη την δυνατότητα παραμετροποιημένης κρυφής μνήμης εντολών αλλά και δεδομένων. Όταν ο χρήστης ενεργοποιήσει την κρυφή μνήμη εντολών ο χώρος εντολών χωρίζεται σε δύο κομμάτια, την κρυφή

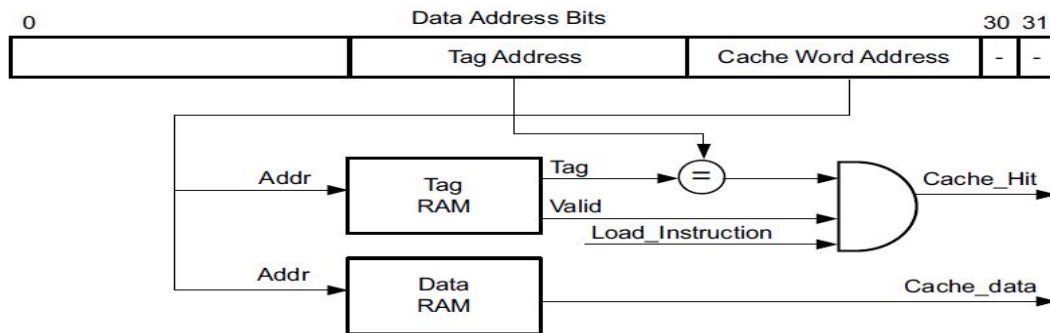
και την κανονική. Το εύρος της κρυφής μνήμης ορίζεται από δυο παραμέτρους, `C_ICACHE_BASEADDR` και `C_ICACHE_HIGHADDR` και πρέπει να είναι αναγκαστικά δύναμη του 2. Η διεύθυνση της κρυφής μνήμης εντολών χωρίζεται σε δύο μέρη : ετικέτα και δεδομένα. Στον Microblaze η κρυφή μνήμη εντολών μπορεί να έχει μέγεθος 64 bytes έως 64 kB, οπότε τα δεδομένα είναι μεταξύ 6 και 16 bits. Για κάθε εντολή που καλείται η κρυφή μνήμη ελέγχει αν η διεύθυνση ανήκει στο τμήμα της κρυφής μνήμης. Αν ανηκεί, γίνεται έλεγχος στην μνήμη tag RAM για να δούμε αν η ζητούμενη διεύθυνση είναι προσωρινά αποθηκευμένη στην κρυφή μνήμη. Υπάρχει επιτυχία αν η έξοδος της μνήμης tag RAM είναι ίδιο με το tag address της διεύθυνσης και το σήμα valid είναι ενεργοποιημένο. Μια πιο σαφή εικόνα της οργάνωσης της κρυφής μνήμης φαίνεται στην παρακάτω εικόνα.



2.6 Instruction Cache Memory Organization

Το ίδιο ισχύει και για την κρυφή μνήμη δεδομένων απλά αλλάζουν οι παράμετροι. Σε αυτήν την περίπτωση είναι οι `C_DCACHE_BASEADDR` and `C_DCACHE_HIGHADDR`. Η παράμετρος `C_DCACHE_USE_WRITEBACK` καθορίζει αν η κρυφή μνήμη θα υλοποιηθεί με το πρωτόκολλο write-back ή write-through. Όταν είναι ενεργοποιημένη, εφαρμόζεται το πρωτόκολλο write-back. Στο πρωτόκολλο write-through τα δεδομένα γράφονται πάντα στη μνήμη και στην κρυφή μνήμη ενώ στο write-back τα δεδομένα γράφονται πάντα στην

κρυφή μνήμη και αντιγράφεται στη μνήμη όταν καθαριστεί από άλλη θέση μνήμης.



2.7 Data Cache Memory Organization

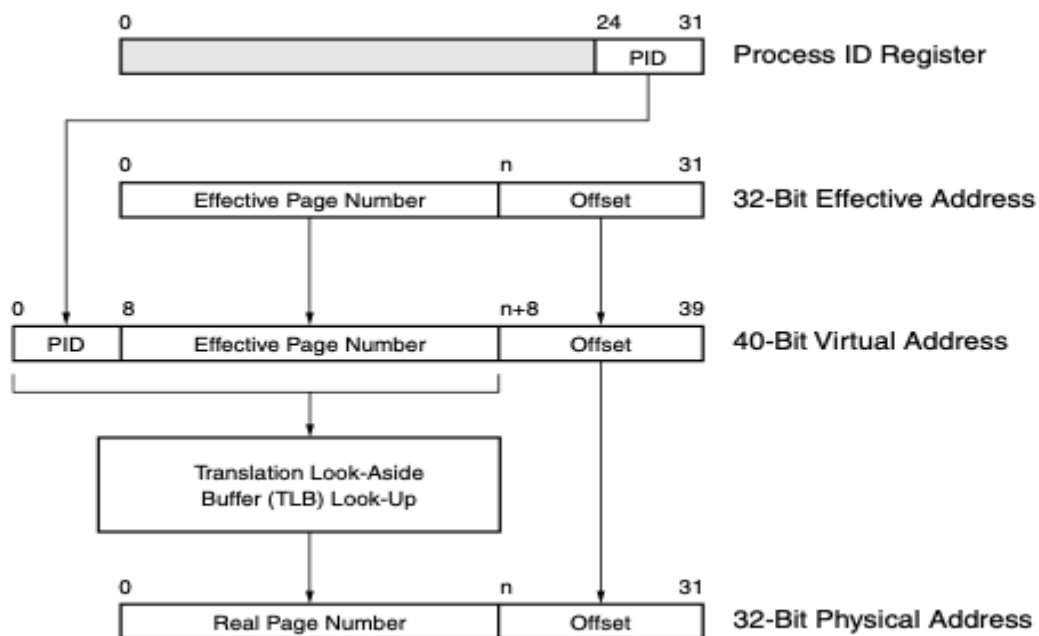
Η μονάδα κινητής υποδιαστολής είναι μονής ακρίβειας (single precision) όταν ο Microblaze είναι ρυθμισμένος στα 32-bit ενώ στα 64-bit είναι διπλής ακρίβειας. Όταν έχουμε μονή ακρίβεια ένα bit χρησιμοποιείται για το πρόσημο, 8 bit για τον εκθέτη και 23 για το κλασματικό μέρος. Στην διπλής ακρίβειας έχουμε ένα bit για το πρόσημο, 11 bit για τον εκθέτη και 52 για το κλασματικό μέρος.

Δίνεται η δυνατότητα στον χρήστη να υλοποιήσει στο ίδιο υλικό δυο ή παραπάνω Microblaze soft processors οι οποίοι εκτελώντας το ίδιο πρόγραμμα μπορεί να ανιχνεύσει προσπάθειες παραβίασης υλικού, παροδικά σφάλματα ή μόνιμα σφάλματα υλικού συγκρίνοντας τις εξόδους. Ένα παραδειγμα που είναι χρήσιμη αυτή η δυνατότητα είναι σε μια εφαρμογή κρυπτογράφησης αλλά και σε εφαρμογές που απαιτείται μικρό ποσοστό σφαλμάτων. Η υλοποίηση των processor είναι λίγο διαφορετική όταν θέλουμε να κάνουμε ανίχνευση λαθών και όταν θέλουμε να ελέγξουμε για τυχόν εξωτερικές παραβιάσεις. Και στις δυο περιπτώσεις η παράμετρος C_LOCKSTEP_SLAVE είναι ενεργοποιημένη (δηλαδή πρέπει να τεθεί στο "1") σε όλους τους slave Microblaze επεξεργαστές εκτός από τον master. Ο master Microblaze επεξεργαστής οδηγεί όλα τα

σήματα εξόδου και είναι υπεύθυνος για το αποσφαλμάτωση (debugging) του συστήματος.

2.7 Διαχείριση Εικονικής Μνήμης (Virtual Memory)

Με κατάλληλη παραμετροποίηση ο επεξεργαστής μπορεί να υποστηρίξει εικονική μνήμη. Η εικονική υλοποιεί μετάφραση του χώρου διευθύνσεων ενός προγράμματος για παράδειγμα σε φυσικές διευθύνσεις. Πρόκειται για μια τεχνική διαχείρισης που χρησιμοποιούν τα λειτουργικά συστήματα προκειμένου να αφήνεται χώρος ελεύθερος στην φυσική μνήμη (RAM), έτσι ώστε να εκτελούνται διεργασίες οι οποίες απαιτούν περισσότερη μνήμη RAM από αυτή που διαθέτει το υπολογιστικό σύστημα. Στον Microblaze ενεργοποιείται η λειτουργία και προστίθεται εσωτερικά του επεξεργαστή ένα Memory Management Unit (MMU).



UG011_37_021302

Με αυτό τον τρόπο ένα σύστημα μπορεί να μεταφέρει δυναμικά τα προγράμματα που είναι φορτωμένα στη μνήμη σε διαφορετική θέση. Επίσης, δεν χρειάζεται ο προγραμματιστής να λαμβάνει υπόψιν ποιες φυσικές διευθύνσεις μνήμης θα πρέπει να αποδίδονται στις υπόλοιπες διεργασίες.

2.8 Reset, Interrupts, Breaks and Exceptions

Ο Microblaze μπορεί να σχεδιαστεί ώστε να ανταποκρίνεται σε ορισμένα events. Τα events αυτά είναι: reset , interrupts, exceptions και break. Η προτεραιότητα εκτέλεσης των αντίστοιχων callbacks διαδικασιών, από την υψηλότερη προς την χαμηλότερη είναι η εξής:

- Reset
- Hardware Exception
- Non-maskable Break
- Break
- Interrupt
- User Vector (Exception)

Για την ανταπόκριση του επεξεργαστή στα events, αυτές οι προτεραιότητες είναι καθορισμένες σε συγκεκριμένες θέσεις μνήμης με σειρά στις διευθύνσεις τους στα οποία εκτελείται εντολή Jump.

| Event | Vector Address | Register File Return Address |
|-----------------------------------|-------------------------|------------------------------|
| Reset | 0x00000000 - 0x00000004 | - |
| User Vector (Exception) | 0x00000008 - 0x0000000C | Rx |
| Interrupt | 0x00000010 - 0x00000014 | R14 |
| Break: Non-maskable hardware | 0x00000018 - 0x0000001C | R16 |
| Break: Hardware | | |
| Break: Software | | |
| Hardware Exception | 0x00000020 - 0x00000024 | R17 or BTR |
| Reserved by Xilinx for future use | 0x00000028 - 0x0000004F | - |

1.6 Vectors and Return Address Register File Location

2.8.1 Reset

Στην περίπτωση που γίνει reset, τότε αδειάζει το pipeline από τις εντολές οι οποίες εκτελούνται την συγκεκριμένη χρονική στιγμή και η εκτέλεση μεταφέρεται στο reset vector, στην θέση 0x00000000 από όπου διαβάζεται η επόμενη εντολή που θα εκτελεστεί.

2.8.2 Hardware Exceptions

Ο Microblaze μπορεί να παραμετροποιηθεί έτσι ώστε να αντιλαμβάνεται και να ενεργεί κατάλληλα όταν συμβαίνουν κάποια συγκεκριμένα σφάλματα. Ορισμένα από αυτά τα exceptions λαμβάνουν χώρα όταν ενεργοποιηθούν συγκεκριμένες μονάδες οι οποίες μπορούν να προκαλέσουν τα σφάλματα αυτά.

Αναλυτικότερα τα σφάλματα είναι τα εξής:

- Μη έγκυρες εντολές (illegal instructions).
- Σφάλματα Εντολών (instruction errors).
- Σφάλματα διαύλου (bus error).
- Ανάγνωση εντολών που δεν είναι ευθυγραμμισμένες (unaligned access).
- Σφάλμα στην διαίρεση τελεστών (divide exception), μόνο σε περίπτωση που ο επεξεργαστής έχει ρυθμιστεί να περιέχει κάποιο διαιρέτη σε επίπεδο hardware.
- Σφάλματα κινητής υποδιαστολής, μόνο στην περίπτωση που στον επεξεργαστή περιλαμβάνεται Floating Point Unit (FPU).
 - underflow
 - overflow
 - float division-by-zero
 - invalid operation
 - denormalized operand error
- Σφάλματα Μονάδας Διαχείρισης Μνήμης (MMU):
 - illegal instruction exception

- data storage exception
- instruction storage exception
- data TLB miss exception
- instruction TLB miss exception

Όταν δημιουργηθεί κάποιο σφάλμα η εκτέλεση των εντολών σταματάει και αδειάζει το pipeline του επεξεργαστή, στη συνέχεια η εκτέλεση μεταφέρεται στο hardware exception vector (0x00000020).

Σε περίπτωση που το error προέρχεται από την MMU, εκτός του illegal instruction exception, η εντολή που προκάλεσε το σφάλμα εκτελείται ξανά μετά την εκτέλεση της συνάρτησης που διαχειρίζεται το σφάλμα. Σε όλες τις άλλες περιπτώσεις, η εκτέλεση μεταφέρεται στην επόμενη εντολή.

Όταν δημιουργηθούν παραπάνω από ένα σφάλμα ταυτόχρονα, ο επεξεργαστής τα διαχειρίζεται με την παρακάτω σειρά, από την υψηλότερη προτεραιότητα προς την χαμηλότερη:

- Instruction Bus Exception
- Instruction TLB Miss Exception
- Instruction Storage Exception
- Illegal Opcode Exception
- Privileged Instruction Exception
- Data TLB Miss Exception
- Data Storage Exception
- Unaligned Exception
- Data Bus Exception
- Divide Exception
- FPU Exception
- Fast Simplex Link Exception

2.8.3 Breaks

Υπάρχουν δυο είδη breaks:

- Hardware (external) breaks: τα hardware breaks επιτυγχάνονται μέσω κάποιων εξωτερικών σημάτων. Όταν συμβαίνει ένα break, η εντολή στο στάδιο της αποκωδικοποίησης αντικαθίσταται από μια διακλάδωση σε ένα break vector (διεύθυνση 0x18).
- Software (internal) breaks: για να επιτευχθεί ένα software break, πρέπει να χρησιμοποιηθεί μια από τις εντολές brk ή brki.

2.8.4 Interrupt

Ο Microblaze μας δίνει την δυνατότητα να υποστηρίξουμε την χρήση ενός μόνο εξωτερικού interrupt. Στην περίπτωση που είναι ενεργοποιημένα τα interrupts στην παραμετροποίηση του επεξεργαστή, το σήμα Interrupt Enable (IE) στον καταχωρητή Machine Status Register έχει την τιμή 1 και εάν ενεργοποιηθεί η είσοδος του interrupt τότε η εκτέλεση μεταφέρεται στο interrupt vector στην διεύθυνση (0x00000010). Ο επεξεργαστής μπορεί να ρυθμιστεί να αναγνωρίζει level και edge interrupt. Χρησιμοποιώντας εξωτερικό περιφερειακό interrupt controller, μπορούν να υποστηριχθούν παραπάνω από ένα interrupts.

2.9 Interfaces

Ο Microblaze μας δίνει την δυνατότητα χρήσης αρκετών διαφορετικών interfaces μέσω του οποίου θα επικοινωνεί με την μνήμη ή με κάποιο περιφερειακό. Τα interfaces τα οποία μπορούμε να βάλουμε στο design είναι τα εξής:

- AXI ή AXI4-Lite
- PLB (Processor Local Bus)
- LMB (Local Memory Bus)

- FSL (Fast Simplex Link)
- XCL (Xilinx CacheLink)

Τα AXI και το PLB interfaces χρησιμοποιούνται για την σύνδεση στα αντίστοιχα interconnects. Το LMB χρησιμοποιείται για την επικοινωνία με BRAM, block μνήμης εσωτερικά του FPGA. Το FSL όπως και το AXIStream χρησιμοποιούνται για point-to-point επικοινωνία με κάποιο περιφερειακό. Τέλος το Xilinx CacheLink κάνει χρήση δύο FSL, προσφέροντας μια γρήγορη λύση για την πρόσβαση σε εξωτερική μνήμη.

3 Interconnections and Interfaces

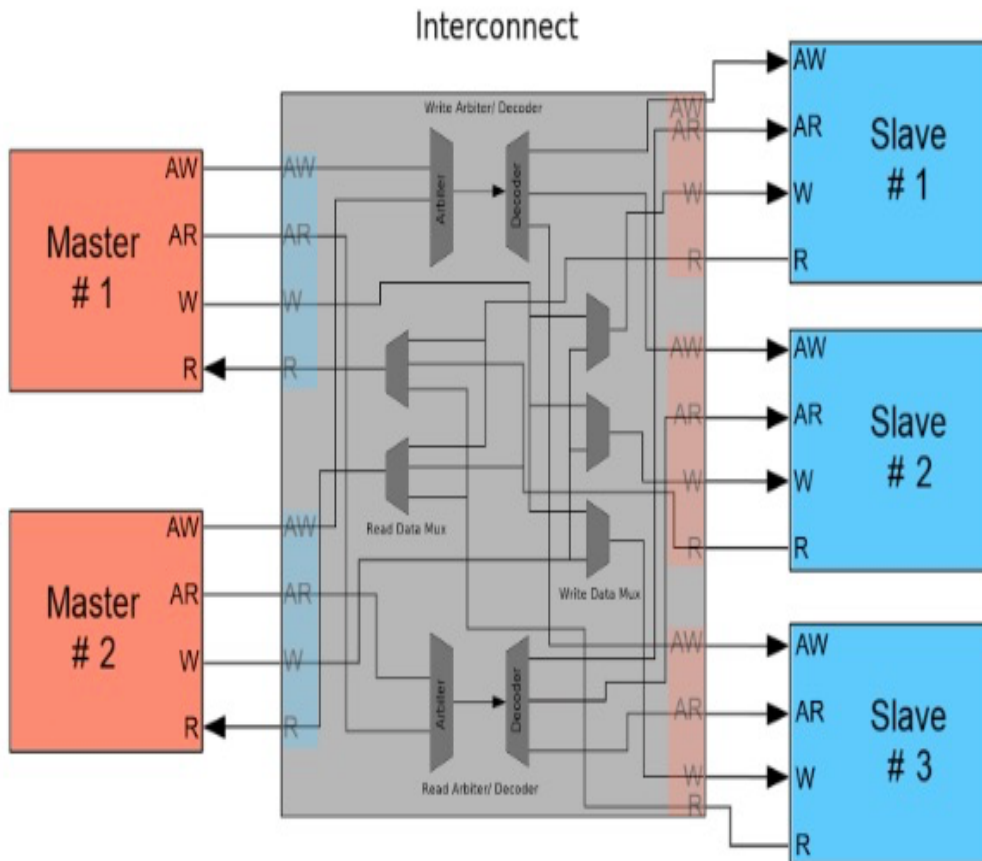
Όπως αναφέρθηκε και προηγουμένως, ο Microblaze διαθέτει την δυνατότητα διασύνδεσης μέσω διαφορετικών interfaces τόσο με την μνήμη όσο και με κάποιο περιφερειακό. Τα interfaces που μπορούν να υλοποιηθούν είτε σαν IPs είτε γράφοντας σε κάποια γλώσσα περιγραφής υλικού (HDL) περιγράφοντας την λειτουργία τους είναι τα εξής:

- AXI ή AXI4-lite (Advanced eXtensive Interface)
- PLB (Processor Local Bus)
- LMB (Local Memory Bus)
- FSL (Fast Simplex Link)
- XCL (Xilinx CacheLink)

Τα AXI και το PLB interfaces χρησιμοποιούνται για την σύνδεση στα αντίστοιχα interconnects. Το LMB χρησιμοποιείται για την επικοινωνία με BRAM, block μνήμης εσωτερικά του FPGA. Το FSL όπως και το AXIStream χρησιμοποιούνται για point-to-point επικοινωνία με κάποιο περιφερειακό. Τέλος το Xilinx CacheLink κάνει χρήση δύο FSL, προσφέροντας μια γρήγορη λύση για την πρόσβαση σε εξωτερική μνήμη.

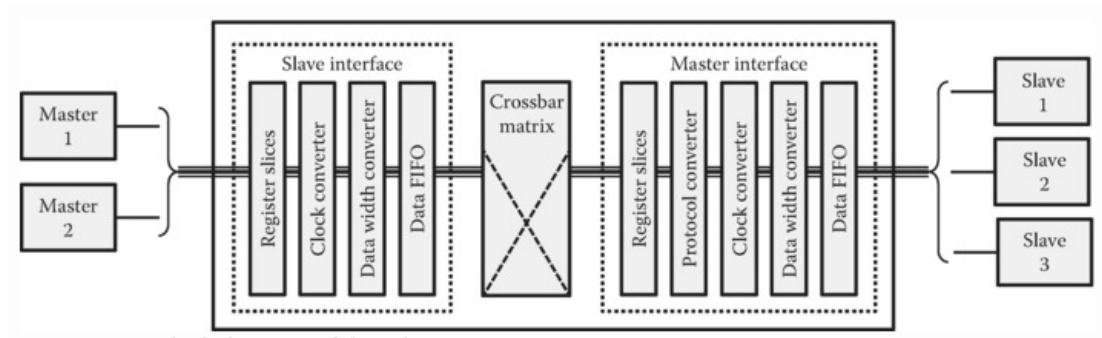
3.1 AXI Interconnection Interface

Το AXI Interconnection Interface χρησιμοποιείται για την διασύνδεση με σκοπό την επικοινωνία μιας ή περισσοτέρων master συσκευών με μια ή περισσότερες slave συσκευές. Το AXI είναι μια υλοποίηση σύγχρονου, υψηλής επίδοσης, παράλληλου και υψηλής συχνότητας interface που σχεδιάστηκε για επικοινωνία εντός του chip. Τα AXI interfaces είναι υλοποίηση της προδιαγραφής AMBA AXI 4 από την ARM.



3.1 AXI interface

Το core αποτελείται από slave interface, το master interface και το crossbar.



3.2 Block Diagram of AXI Interconnect

Όσον αφορά την υλοποίηση στο FPGA διαφέρει ανάλογα με την χρήση που έχει οριστεί, δηλαδή με τον αριθμό των master και slave components καθώς και των μετατροπών ή του pipeline. Η τελική υλοποίηση στο FPGA μπορεί να πάρει τις παρακάτω μορφές:

- Pass through: στην περίπτωση που είναι συνδεδεμένοι μόνο ένα master και ένα slave και δεν υπάρχει κάποια άλλη λειτουργία, τότε η υλοποίηση μετατρέπεται σε απευθείας σύνδεση των συσκευών.
- Conversion only: στην περίπτωση αυτή γίνεται κάποια μετατροπή ή γενικότερα κάποια λειτουργία, οι οποίες μπορεί να είναι μια από τις:
 1. Data width conversion
 2. Clock rate conversion
 3. AXI4-lite slave adaption
 4. Pipelining
- N-to-1 Interconnect: στην περίπτωση αυτή έχουμε πολλές master συσκευές αλλά μόνο μια slave συσκευή τότε το κύκλωμα διαθέτει arbiter για να μπορεί να καθοριστεί η πρόσβαση στην slave συσκευή.
- 1-to-N Interconnect: στην περίπτωση αυτή έχουμε μια master και πολλές memory mapped συσκευές. Σε αυτή περίπτωση δεν χρειάζεται κύκλωμα διαιτησίας, εφόσον έχουμε έναν master, αλλά

κύκλωμα αποκωδικοποίησης και δρομολόγησης για τις διευθύνσεις και τα δεδομένα.

- N-to-M Interconnection (Crossbar mode): στην περίπτωση αυτή έχουμε πολλαπλούς master και slaves σε crossbar mode, το interconnect έχει ένα διαμοιραζόμενο arbiter για εγγραφή και ανάγνωση καθώς και πολλαπλές διαδρομές για τα δεδομένα ανάγνωσης και εγγραφής.
- M-to-N Interconnection (Shared Access Mode): στην περίπτωση αυτή σε κάθε χρονική στιγμή μπορεί να πραγματοποιηθεί μόνο σε ένα transaction.

Τέλος το AXI Interconnect core μπορεί να παραμετροποιηθεί ώστε να πραγματοποιεί ορισμένες λειτουργίες μετατροπής κατά την διάρκεια ενός transaction. Ορισμένες από τις λειτουργίες που μπορεί να επιτελέσει το interface είναι:

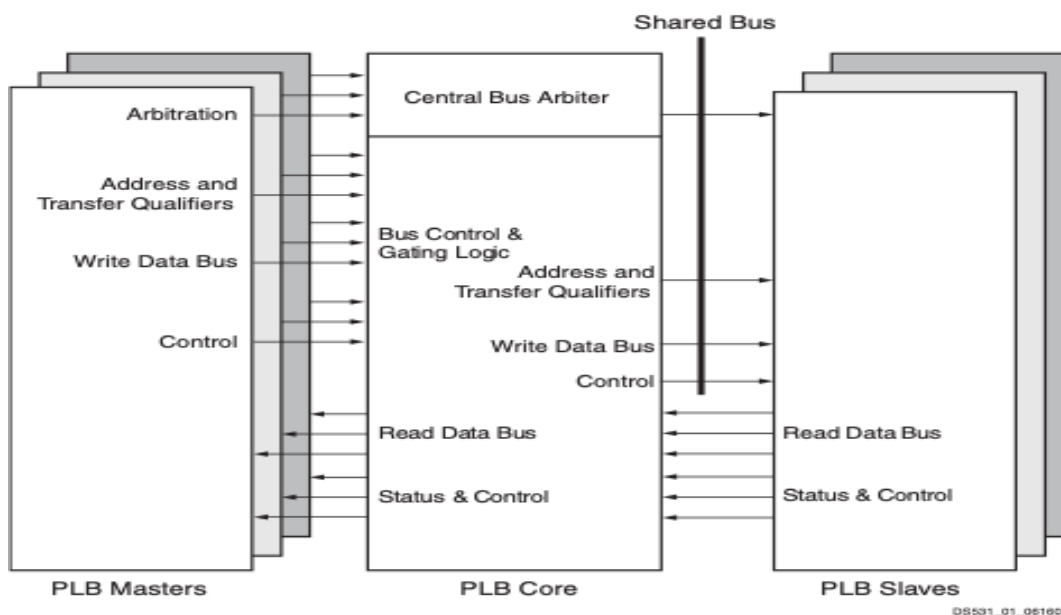
- Width Conversion: η μετατροπή του εύρους της αρτηρίας. Το AXI interconnect έχει εσωτερικά παραμετροποιήσιμο μέγεθος αρτηρίας. Αυτό γίνεται όταν τα δεδομένα έχουν διαφορετικό μέγεθος οπότε θα πρέπει να γίνει μετατροπή στο μήκος των δεδομένων.
- Clock Conversion: μια άλλου είδους μετατροπή που μπορεί να επιτευχθεί είναι η μετατροπή στα ρολόγια των master και slave. Η μετατροπή μπορεί να είναι είτε αύξηση είτε μείωση και στις δυο περιπτώσεις γίνεται μέσω αέρας πύξης, είτε πολλαπλασιασμό για αύξηση είτε διαίρεση για μείωση.
- Peripheral Register Slices: μπορούν να προστεθούν δυο καταχωρητές στα slave και master interfaces καθώς και ένας

buffer εισόδου-εξόδου. Στην πλειονότητα τέτοιων περιπτώσεων ο λόγος είναι η επίτευξη υψηλότερης τιμής ρολογιού ή επίτευξη κάποιας συγκεκριμένης περιόδου, δηλαδή να επιτευχθεί στο με βάση το ρολόι, να έχουμε δηλαδή εφικτά timing results ή για λόγους pipelining.

- Datapath FIFOs: σε κάποιες περιπτώσεις που η φύση του προβλήματος απαιτεί να γίνει κάποιου είδους μετατροπή είτε μήκους αρτηρίας είτε του ρυθμού του ρολογιού η ενδιάμεση αποθήκευση μπορεί να βελτιώσει την απόδοση του συστήματος καθώς και την επίτευξη σωστού timing utilization.

3.2 PLB (Processor Local Bus)

Το PLB της Xilinx αποτελείται από έναν κεντρικό bus arbiter, τον απαραίτητο έλεγχο διαύλου και λογική πύλης και όλες τις απαραίτητες δομές διαύλου OR / MUX. Το Xilinx PLB παρέχει ολόκληρη τη δομή διαύλου PLB και επιτρέπει την άμεση σύνδεση με έναν αριθμό master και slaves. Στο παρακάτω σχήμα φαίνεται ένα τέτοιο component με τρία master τρία slaves.

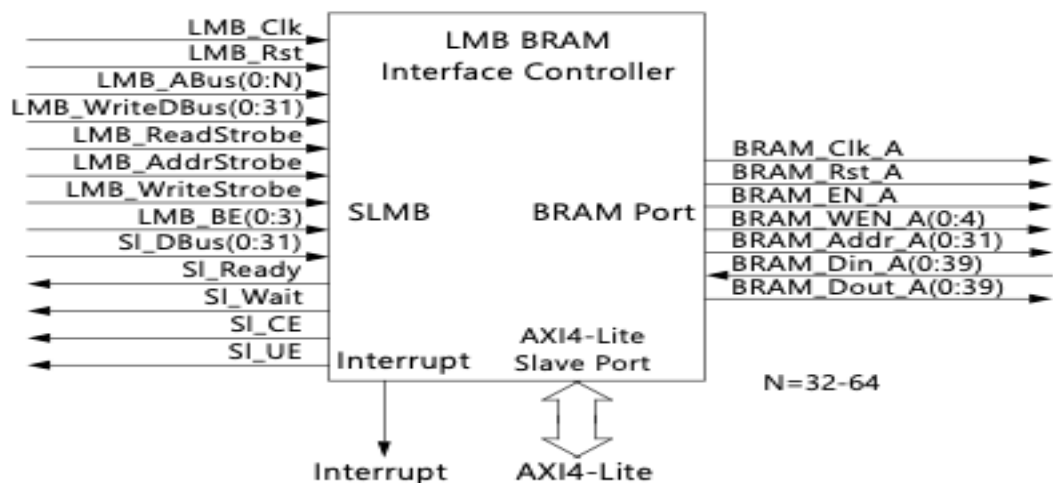


3.3 PLB Interconnect Diagram

Το PLB στον Microblaze χρησιμοποιείται για την διασύνδεση στα αντίστοιχα interconnects. Πιο συγκεκριμένα όπως αναφέρθηκε παραπάνω ο Microblaze είναι παράδειγμα της Harvard αρχιτεκτονικής, οπότε έχει διαχωρισμένα η μνήμη εντολών και η μνήμη δεδομένων και έτσι υπάρχουν PLBs και στις δυο πλευρές δηλαδή υπάρχουν τέτοιου είδους αρτηρίες που επικοινωνούν τόσο με το interface των εντολών όσο και με το interface των δεδομένων, τα IPLB και DPLB.

3.3 LMB (Local Memory Bus)

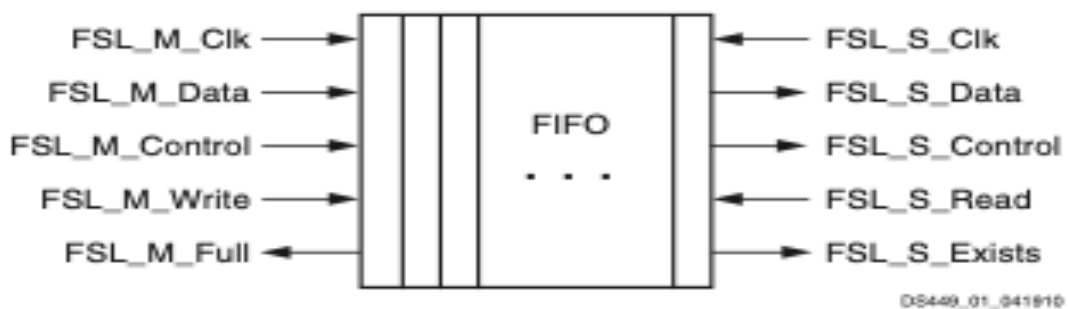
Το LMB (Local Memory Bus) είναι το interface που ενώνει τον επεξεργαστή με το αποτέλεσμα της υλοποίησης από το Block Memory Generator core. Το LMB core χρησιμοποιείται για την διασύνδεση του επεξεργαστή με την BRAM (block RAM) του FPGA. Το LMB είναι ένα γρήγορο, local bus για τη σύνδεση των εντολών του MicroBlaze και των θυρών δεδομένων σε περιφερειακά υψηλής ταχύτητας, κυρίως RAM με μπλοκ chip (BRAM).



3.4 LMB BRAM Interface Controller Core Block Diagram

3.4 FSL (Fast Simplex Link)

Το FSL (Fast Simplex Link) είναι μια αρτηρία μιας κατεύθυνσης για point to point επικοινωνία που χρησιμοποιείται για την γρήγορη επικοινωνία μεταξύ δυο στοιχείων στο FPGA. Το FSL είναι διαθέσιμο στον Microblaze και χρησιμοποιείται για την μεταφορά δεδομένων από και προς το αρχείο καταχωρητών του επεξεργαστή που περιγράφουμε την συμπεριφορά του μέσω μιας γλώσσας περιγραφής υλικού (VHDL, Verilog) στο υλικό του FPGA.



3.5 FSL Bus Block Diagram

3.5 XCL (Xilinx CacheLink)

Το Xilinx CacheLink (XCL) είναι μια διεπαφή που είναι σχεδιασμένη με σκοπό την απευθείας σύνδεση με κάποια μνήμη. Η διεπαφή είναι διαθέσιμη μόνο όταν οι caches του Microblaze είναι ενεργοποιημένες. Οι κρυφές μνήμες μπορούν να σχετίζονται τόσο στην με την μνήμη δεδομένων όσο και με την μνήμη εντολών. Το XSL συνδέεται απευθείας σε έναν memory controller με δυο FSL buffers.

3.6 Peripherals

Για την ολοκληρωμένη υλοποίηση ενός συστήματος στο FPGA, η Xilinx παρέχει έτοιμα περιφερειακά που μπορούν να υλοποιηθούν στο FPGA. Ανάλογα με την λειτουργικότητα τους μπορούν να χωριστούν στις εξής κατηγορίες:

3.6.1 Bus & Bridges

Όπως προαναφέρθηκε στην αρχιτεκτονική, ο Microblaze διαθέτει ένα πλήθος από αρτηρίες για την επικοινωνία με τα περιφερειακά. Ένα είναι το AXI Interconnect, ένα άλλο είναι το PLB το οποίο είναι υλοποίηση του CoreConnect της IBM. Το PLB ήταν η βασική αρτηρία στα ολοκληρωμένα κυκλώματα της Xilinx, αφού χρησιμοποιούνταν και από τον επεξεργαστή PowerPC που ανήκει στην κατηγορία των hard processors και ήταν υλοποιημένος στο ολοκληρωμένο. Υπάρχει το LMB, για την επικοινωνία του επεξεργαστή με BRAMs που είναι υλοποιημένες μέσα στο ολοκληρωμένο. Επιπλέον, υπάρχουν τα cores που υλοποιούν bridges, οι οποίες χρησιμοποιούνται για την διασύνδεση των αρτηριών μεταξύ τους.

3.6.2 Clock , Reset & Interrupt

Γενικά ένας επεξεργαστής χρειάζεται ένα πλήθος από διαφορετικά ρολόγια για τον ίδιο επεξεργαστή αλλά και για τα περιφερειακά που είναι υλοποιημένα μέσα στο ολοκληρωμένο. Για την δημιουργία των κατάλληλων ρολογιών χρησιμοποιώ τον clock manager της Xilinx ή κάποιο custom διαιρέτη περιόδου σε VHDL.

Όσον αφορά τα interrupts υπάρχουν cores τα οποία προσφέρουν τα οποία προσφέρουν λειτουργικότητα που έχει να κάνει με την διαχείριση των interrupts, όπως ο AXI Interrupt.

3.6.3 UARTs

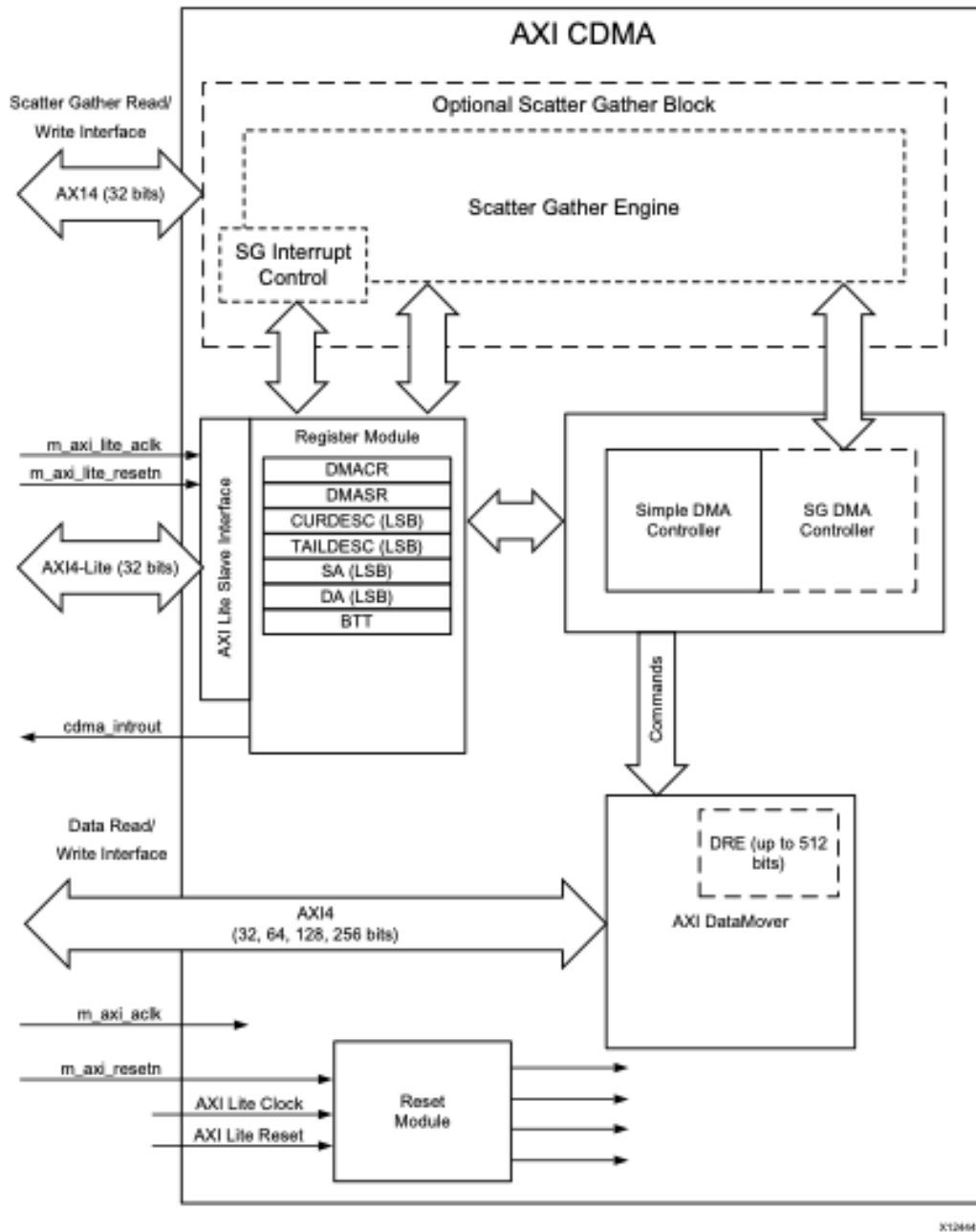
Σε περίπτωση που χρειαστεί να επιτευχθεί κάποιου είδους επικοινωνία με το σύστημα ή με κάποιο εξωτερικό στοιχείο ή διάταξη, θα χρειαστούμε IP

cores των οποίων η υλοποίηση θα ισοδυναμεί με κάποιο πρωτόκολλο επικοινωνίας. Το πιο συνηθισμένο και απλό πρωτόκολλο επικοινωνίας είναι το RS232 που υλοποιείται μέσω του core Uartlite. Σε περίπτωση που είναι συνδεδεμένο στο σύστημα μπορεί αν χρησιμοποιηθεί σαν την κύρια είσοδο/έξοδο του συστήματος.

Ένας άλλος τρόπος επικοινωνίας είναι η σύνδεση στο φυσικό επίπεδο του δικτύου με την χρήση του ethernet. Για τέτοιου είδους επικοινωνία μπορούμε να χρησιμοποιήσουμε το core axi ethernet lite που μπορεί να προσφέρει επικοινωνία 10/100 Mbps. Προφανώς μια τέτοιου είδους επικοινωνία επιτυγχάνεται μέσω κάποιου τερματικού ή του command line στο λειτουργικό σύστημα Linux με τις ssh εντολές.

3.6.4 DMA (Direct Memory Access)

Μια ακόμη είδους σύνδεση που προσφέρεται στο Microblaze είναι το AXI CDMA που προσφέρει DMA λειτουργικότητα μεταξύ δυο memory mapped διευθύνσεων, μιας πηγής και ενός προορισμού, χρησιμοποιώντας το AXI4 πρωτόκολλο. Προαιρετικά μπορεί να υπάρξει η λειτουργία Scatter-Gather, όπου σε επίπεδο χρήστη τρέχουν κάποιες προκαθορισμένες εντολές με σκοπό τις αυτόματες DMA μεταφορές.



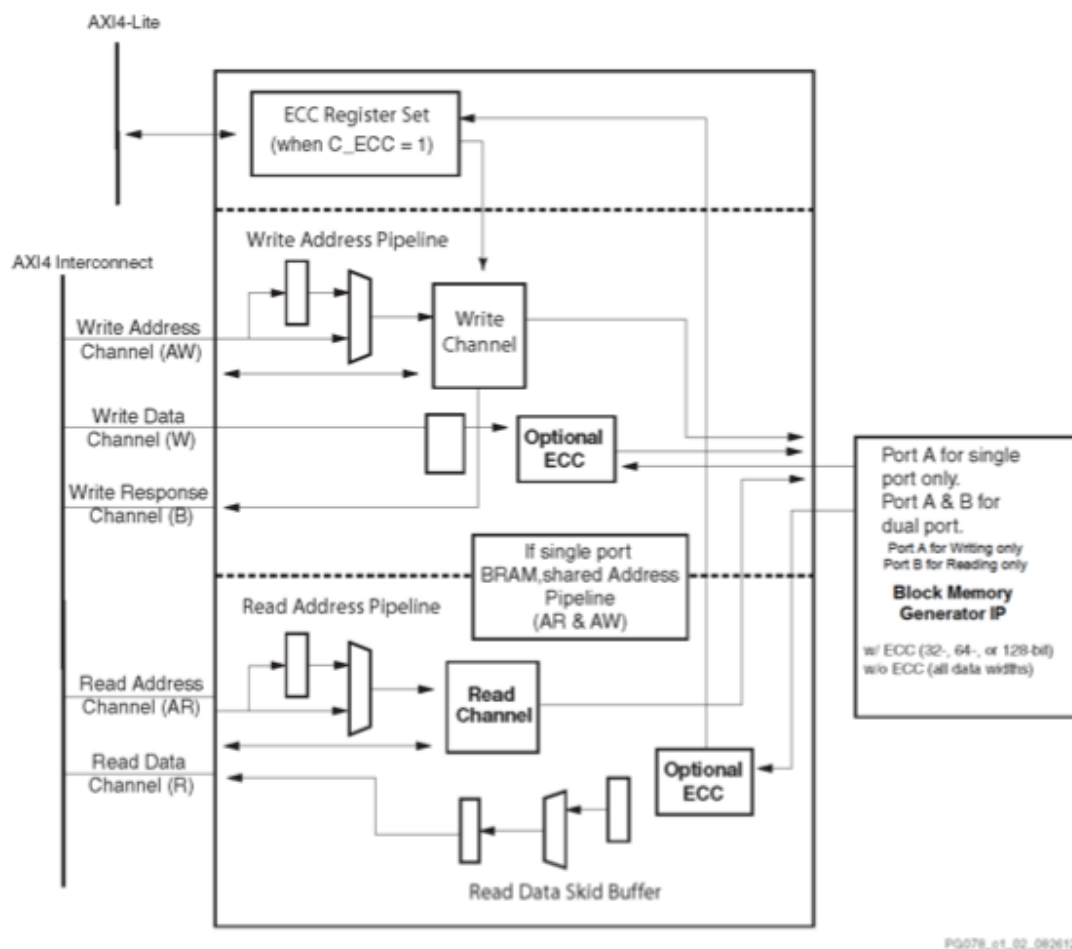
3.7 AXI CDMA Block Diagram

3.6.5 Memory & Memory Controllers

Τα cores που η λειτουργία τους έχει να κάνει με την μνήμη είναι cores που εκμεταλλεύονται την εσωτερική μνήμη του FPGA καθώς και κάποιοι controllers για την επικοινωνία με την μνήμη. Ο σχεδιαστής μπορεί να χρησιμοποιήσει τα resources σε μνήμη που υπάρχουν εσωτερικά του FPGA

είτε μέσω του LMB BRAM Controller είτε μέσω του core AXI BRAM Controller. Όσον αφορά το πρώτο είναι σχεδιασμένο για την γρήγορη πρόσβαση στην BRAM, ενώ το δεύτερο υποστηρίζει το AXI4 ή το AXI4-lite Interconnect.

Για την επικοινωνία με εξωτερική μνήμη ram θα πρέπει να χρησιμοποιηθεί ειδικό core που να υλοποιεί τον controller της μνήμης. Ο controller μπορεί να παραμετροποιηθεί έτσι ώστε να είναι συμβατός με κάθε ξεχωριστή οικογένεια FPGAs της Xilinx.

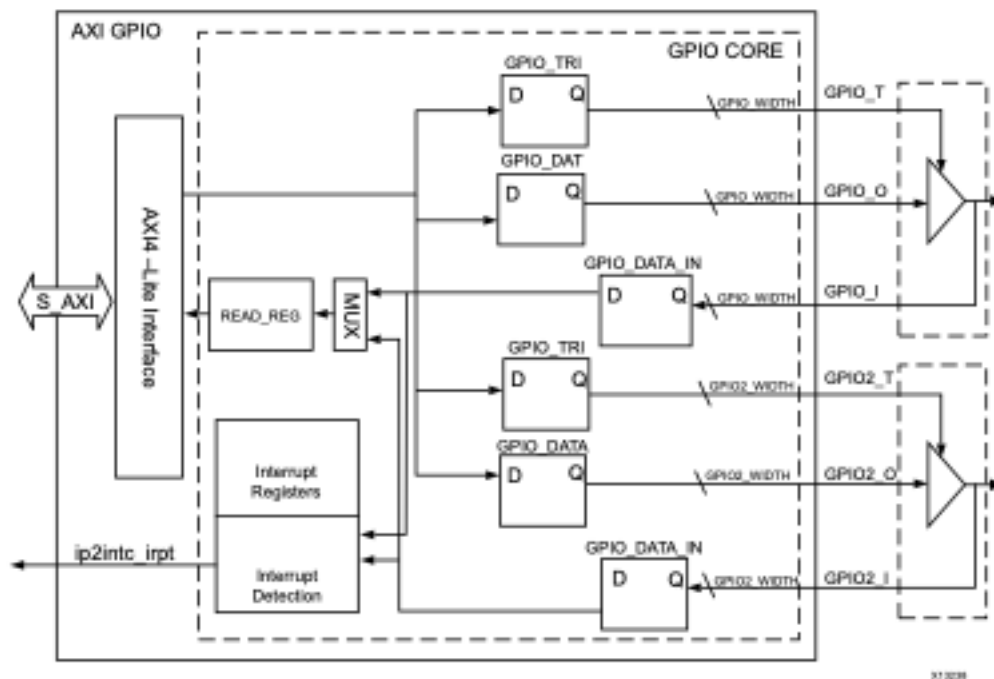


3.8 AXI4 BRAM Controller Block Diagram

3.6.6 GPIO (General Purpose Input Output)

Τέλος, ένα σύστημα θα πρέπει να μπορεί να συνδέεται και να επικοινωνεί με εξωτερικές συσκευές. Σε απλές περιπτώσεις όπου θέλουμε όπου θέλουμε να ελεέγξουμε τις τιμές κάποιων pins στο FPGA ή την σύνδεση

της πλευράς του ολοκληρωμένου για την επικοινωνία μέσω ενός ή δυο καναλιών με το εξωτερικό του chip. Οι θύρες διαμορφώνονται δυναμικά για είσοδο ή έξοδο ενεργοποιώντας ή απενεργοποιώντας το 3-state buffer. Τα κανάλια μπορούν να ρυθμιστούν ώστε να δημιουργούν interrupt όταν πραγματοποιείται μετάβαση σε οποιαδήποτε από τις εισόδους τους.



3.9 AXI GPIO Block Design

4 Configuration

Στο τελευταίο μέρος, θα αναφερθούμε στις δυνατότητες που έχει ο Microblaze για επεξεργασία δεδομένων καθώς στο προηγούμενο αναφερθήκαμε στις δυνατότητες που έχει ο επεξεργαστής ως προς τις περιφερειακές συνδέσεις.

4.1 General Configuration




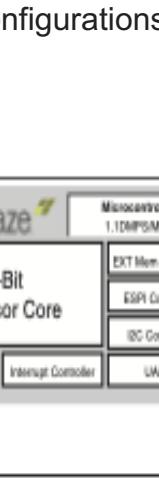
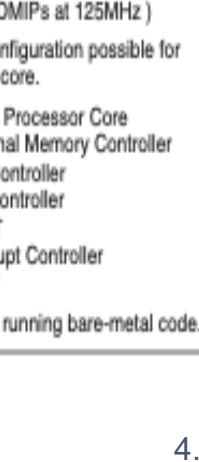
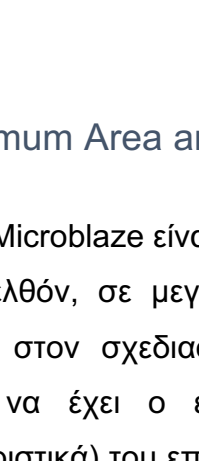
Μια ποικιλία ενσωματωμένων επεξεργαστών που διατίθενται σήμερα στην αγορά στοχεύουν διαφορετικούς τομείς εφαρμογών. Ένα ενσωματωμένο σύστημα είναι ένα σύστημα υλικού υπολογιστή που βασίζεται σε μικροεπεξεργαστή με λογισμικό που έχει σχεδιαστεί για να εκτελεί μια ειδική λειτουργία, είτε ως ανεξάρτητο σύστημα είτε ως μέρος ενός μεγάλου συστήματος. Στον πυρήνα είναι ένα ολοκληρωμένο κύκλωμα σχεδιασμένο να εκτελεί υπολογισμό για λειτουργίες σε πραγματικό χρόνο. Αυτά τα συστήματα διαφέρουν σε μέγεθος, ταχύτητα, απόδοση επεξεργαστή, τύπο υποσυστήματος μνήμης και περιφερειακά. Επιπλέον, το λογισμικό που εκτελείται σε κάθε σύστημα, συμπεριλαμβανομένου του λειτουργικού συστήματος, γίνεται όλο και πιο περίπλοκο.

| Application | Persona | Description |
|---|-----------------|--|
| Basic Control Applications | Microcontroller | Typically uses small, single-task oriented processors, often running bare-metal code. Used in printers, faxes, appliances, etc. |
| Time-Sensitive/Deterministic Processing | Real-Time | Fast, deterministic task-switching processor using an RTOS. Used in control systems, CNC machines, automatic driving assistance systems, etc. |
| High Performance Embedded Systems | Application | Uses a comprehensive operating system (such as Linux) with multiple tasks running. For complex embedded systems such as network switches, routers, video/image processing systems, signal processing, etc. |

4.1 Typical Embedded Systems

Ο επεξεργαστής MicroBlaze προσφέρει στον σχεδιαστή πολλά εξαιρετικά πλεονεκτήματα σε σχέση με άλλους τυπικούς μικροεπεξεργαστές ή μικροελεγκτές. Το Xilinx παρέχει μια τεράστια βιβλιοθήκη περιφερειακών για τη δημιουργία ενσωματωμένων συστημάτων επεξεργασίας για διάφορες εφαρμογές. Η προγραμματιζόμενη φύση των Xilinx FPGAs και SoCs παρέχει απόλυτη ευελιξία στην προσαρμογή του επεξεργαστή MicroBlaze για συγκεκριμένες ανάγκες εφαρμογών.

Οι διαφορετικές διαμορφώσεις του Microblaze ανάλογα με τα δεδομένα που πρέπει να επεξεργαστεί είναι αυτά που φαίνονται στον πίνακα παρακάτω. Αναλόγως με την απαίτηση σε χαρακτηριστικά διαλέγουμε κάποιο από τα configurations.

| MicroBlaze  Microcontroller 1.1 DMIPS/MHz | MicroBlaze  Real-Time Processor 1.3 DMIPS/MHz | MicroBlaze  Application Processor 1.4 DMIPS/MHz |
|--|---|---|
|  |  |  |
| <p>Microcontroller Preset (137 DMIPs at 125MHz)</p> <p>Smallest configuration possible for MicroBlaze core.</p> <ul style="list-style-type: none"> • 32-bit Processor Core • External Memory Controller • SPI Controller • I2C Controller • UART • Interrupt Controller • Timer <p>Suitable for running bare-metal code.</p> | <p>Real-Time Processor Preset (162 DMIPs at 125MHz)</p> <p>Targeted for Deterministic and Real-Time Systems.</p> <ul style="list-style-type: none"> • All Microcontroller Preset blocks • Instruction Cache • Data Cache • DDR Controller <p>MicroBlaze core configured to run Real-Time Operating system like a FreeRTOS.</p> | <p>Application Processor Preset (175 DMIPs at 125MHz)</p> <p>MicroBlaze settings suitable to get high performance when running Linux with Memory Management Unit (MMU).</p> <ul style="list-style-type: none"> • All Real-Time Processor Preset blocks • Memory Management Unit <p>Configured to run comprehensive operating systems, such as Linux.</p> |

WP901_02_041918

4.2 Microblaze Processor Presets

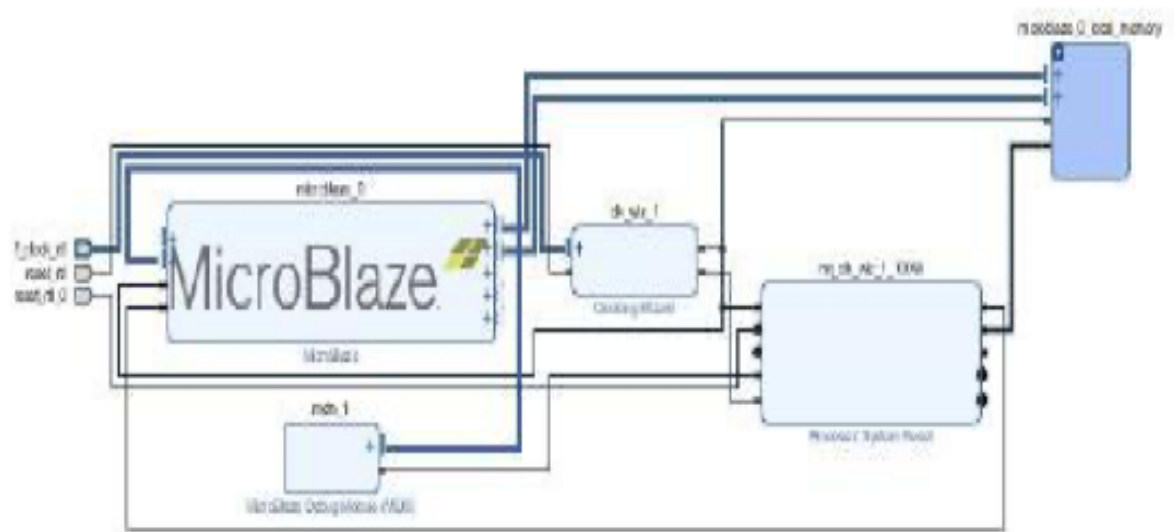
4.2 Minimum Area and Maximum Performance Configuration

Ο Microblaze είναι ένας soft core επεξεργαστής, όπως έχουμε αναφέρει στο παρελθόν, σε μεγάλο βαθμό παραμετροποιήσιμος γεγονός το οποίο επιτρέπει στον σχεδιαστή να επιλέξει μεταξύ των χαρακτηριστικών που επιθυμεί να έχει ο επεξεργαστής. Υπάρχουν δυο σεντ από επιλογές (χαρακτηριστικά) του επεξεργαστή το οποία καθορίζουν την αρχιτεκτονική του core. Υπάρχουν κάποια χαρακτηριστικά που εισέρχονται σε κάθε περίπτωση. Τέτοια είναι το σεντ των 32-bit καταχωρητών γενικού σκοπού, 32-bit μήκος εντολών με τρεις τελεστές και δυο διευθύνσεις, 32-bit αρτηρία

διευθυνσιοδότησης. Ο Microblaze ανάλογα με την παραμετροποίηση του, λοιπόν, μπορεί να πάρει τις εξής μορφές:

- Minimum Area: το μικρότερο δυνατό core του Microblaze. Δεν διαθέτει caches ή debug modes.
- Maximum Performance: η μέγιστη δυνατή απόδοση. Διαθέτει μεγάλες caches και debug mode.

Για το minimum area mode, ο MicroBlaze είναι προγραμματισμένο για την ελάχιστη διαμόρφωση περιοχής και το core IP δημιουργείται για βασική αρχιτεκτονική μικροεπεξεργαστή. Η HDL (Hardware Description Language) δημιουργείται για το ίδιο και η σύνθεση του γίνεται με τη χρήση του Vivado. Το παρακάτω σχήμα δείχνει το MicroBlaze σε ρύθμιση ελάχιστης περιοχής. Το HDL είναι η γλώσσα επιλογής για την περιγραφή ενός σχεδιασμού FPGA. Ένα περιτύλιγμα HDL είναι βασικά μια περιγραφή ανώτερου επιπέδου του συστήματος.

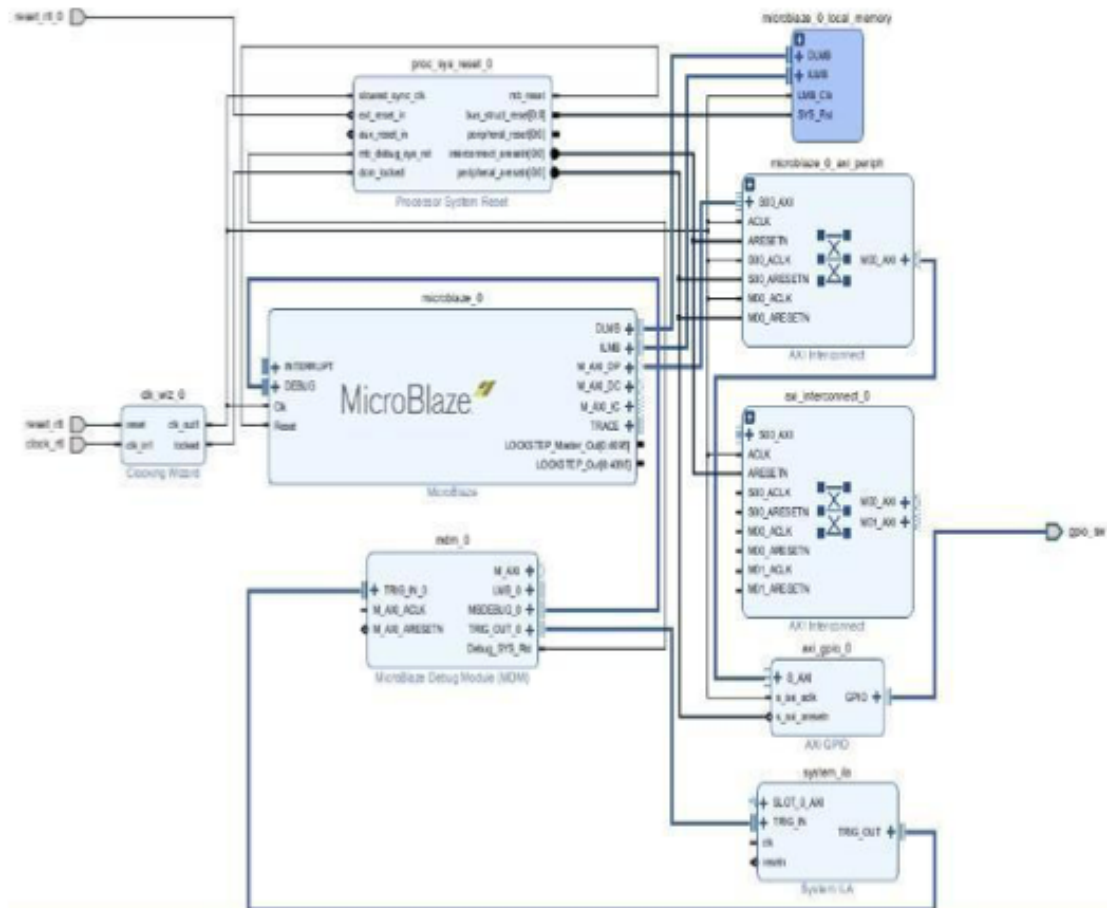


4.3 Minimum Area Configuration

Για το maximum performance mode, η διαμόρφωση μέγιστης απόδοσης του MicroBlaze περιλαμβάνει όλες τις διαθέσιμες δυνατότητες του επεξεργαστή MicroBlaze. Η διαμόρφωση μέγιστης απόδοσης περιλαμβάνει τη μνήμη προσωρινής μνήμης υψηλής ταχύτητας και περιλαμβάνεται η προσωρινή μνήμη δεδομένων υψηλής απόδοσης και στοχεύει για εφαρμογές υψηλής περιοχής και υψηλής απόδοσης. Η υψηλή απόδοση του σχεδιασμού αντισταθμίστηκε με τις απαιτήσεις κατανάλωσης ισχύος και περιοχής του επεξεργαστή.

Η μέγιστη απόδοση του σχεδιασμού χρησιμοποιείται για την επίτευξη υψηλής ταχύτητας, υψηλής απόδοσης επεξεργαστών. Χρησιμοποιείται για υπολογιστικά πολύπλοκους αλγόριθμους. Η λειτουργία μέγιστης απόδοσης της διαμόρφωσης φαίνεται στο παρακάτω σχήμα καθώς και ο πίνακας

| | Min. Area | Max. Performance | Percentage increase in area (%) |
|-----------------|-----------|------------------|---------------------------------|
| LUTs | 985 | 4214 | 327.82 |
| Slice registers | 1334 | 4218 | 216.19 |
| Mux | 33 | 148 | 348.48 |
| Block RAM tile | 2 | 21 | 950 |



4.4 Maximum Performance Configuration

Η σύγκριση των διαφορετικών διαμορφώσεων μας δίνει τη βοήθεια στην επιλογή της συγκεκριμένης διαμόρφωσης για μια στοχευμένη εφαρμογή. Η ίδια διαμόρφωση μπορεί να εφαρμοστεί στο Software Development Kit (SDK). Η εφαρμογή μπορεί να εκτελεστεί πάνω από τον πολυεπεξεργαστή, για να βελτιώσει το πρωτότυπο και τις στοχευμένες απαιτήσεις εφαρμογής. Έτσι, η μελέτη αποκαλύπτει ότι ο πυρήνας Xilinx MicroBlaze είναι κατάλληλος για εφαρμογές χαμηλής έκτασης και υψηλής ταχύτητας.

Βιβλιογραφικές Αναφορές

1. Xilinx Inc, MicroBlaze Processor Reference Guide, https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_2/ug984-vivado-microblaze-ref.pdf, June 21, 2018
2. Xilinx Inc, LogiCORE IP Product Guide AXI Interconnect v2.1, https://www.xilinx.com/support/documentation/ip_documentation/axi_interconnect/v2_1/pg059-axi-interconnect.pdf, December 20,2017
3. Xilinx Inc, LogiCORE IP Product Guide AXI4-Lite IPIF v3.0, https://www.xilinx.com/support/documentation/ip_documentation/axi_lite_ipif/v3_0/pg155-axi-lite-ipif.pdf, April 6,2016
4. Xilinx Inc, The Microblaze Soft Processor: Flexibility and Performance for Cost-Sensitive Embedded Designs, https://www.xilinx.com/support/documentation/white_papers/wp501-microblaze.pdf, April 13, 2017
5. <https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18842560/MicroBlaze>
6. <https://www.xilinx.com/publications/product-briefs/microblaze-product-brief.pdf>
7. <https://www.jblopen.com/microblaze-configuration-part-1/>
8. <https://www.jblopen.com/microblaze-configuration-part-2/>
9. Patterson D. A., and J. L. Hennessey, Computer Organization and Design: The Hardware/Software Approach, 3rd edition, 2004
10. Wakerly J., Digital Design: Principles and Practices, 4th edition, July 2005
11. Mattson D. and Christensson M., Evaluation of Synthesizable CPU cores, Gothenburg 2004
12. Dr. S.Hema Chitra, Sarvesh K, Madhu Preetha BR, Comparison of Different Configurations of MicroBlaze Soft IP Core , 2019