



Πανεπιστημίο Πατρών

Μηχανικών Η/Υ και Πληροφορικής

**Σχεδιασμός System-on-Chip για επεξεργασία εικόνας
και υλοποίηση με FPGA.**

Author:
Χαράλαμπος
Καλάργαρης

Supervisor:
Θεμιστοκλής
Χανιωτάκης

10 Απριλίου 2012

Περιεχόμενα

1	Εισαγωγή	4
2	OpenRISC 1200 IP Core	5
2.1	Εισαγωγή	5
2.1.1	OpenRISC Οικογένεια	5
2.1.2	OpenRISC 1200	5
2.2	Αρχιτεκτονική	8
2.2.1	CPU/DSP	8
2.2.2	Κρυφή Μνήμη Δεδομένων	11
2.2.3	Κρυφή Μνήμη Εντολών	14
2.2.4	Διαχείριση Μνήμης (MMU) Δεδομένων	16
2.2.5	Διαχείριση Μνήμης (MMU) Εντολών	17
2.2.6	Προγραμματιζόμενος Ελεγκτής Διακοπών	19
2.2.7	Tick Timer	20
2.2.8	Διαχείριση Ενεργειακών Απαιτήσεων	20
2.2.9	Μονάδα Αποσφαλμάτωσης	21
2.2.10	Σήματα Χρονισμού και Επανεκκίνησης	21
2.2.11	Δίαυλος επικοινωνίας Wishbone	22
3	OR1200 Simulation	23
3.1	Simulation Enviroment	23
3.2	Package Structure	23
3.3	Simulation commands	25
3.3.1	Η βασική διαδικασία	25
3.3.2	Εκτέλεση ενός συγκεκριμένου test	25
3.3.3	Εκτέλεση συγκεκριμένων tests μαζί	25
3.3.4	Παρέχοντας μια προσαρμοσμένη VMEM εικόνα (image)	25
3.3.5	Παρέχοντας ένα "precompiled" εκτελέσιμο .ELF αρχείο	26
3.3.6	Κυματομορφές	26
3.3.7	Επιπρόσθετα επιλογές στις εντολες	26

Κατάλογος σχημάτων

2.1.1 Σχηματική αναπαράσταση OpenRISC αρχιτεκτονικής.	5
2.2.1 Core's Architecture	8
2.2.2 CPU/DSP block diagram.	9
2.2.3 Οργάνωση κρυφής μνήμης δεδομένων.	13
2.2.4 Οργάνωση κρυφής μνήμης εντολών	15
2.2.5 Οργάνωση MMU δεδομένων.	17
2.2.6 Οργάνωση MMU εντολών.	19
2.2.7 Μπλοκ διάγραμμα του ελεγκτή διακοπών.	20
2.2.8 Μπλόκ διάγραμμα μονάδας αποσφαλμάτωσης.	22
3.1.1 Συνολική ροή εξομοίωσης.	24

Κατάλογος πινάκων

2.1.1 Overview of OR1200 specifications	7
2.2.1 Ρυθμίσεις μεγέθους κρυφής μνήμης δεδομένων	12
2.2.2 Ρυθμίσεις μεγέθους κρυφής μνήμης εντολών	14
2.2.3 Ρυθμίσεις Δεδομένων TLB (translation lookaside buffer)	16
2.2.4 Ρυθμίσεις Εντολών TLB (translation lookaside buffer)	18
2.2.5 Καταστάσεις βελτιστοποίησης κατανάλωσης ενέργειας.	21

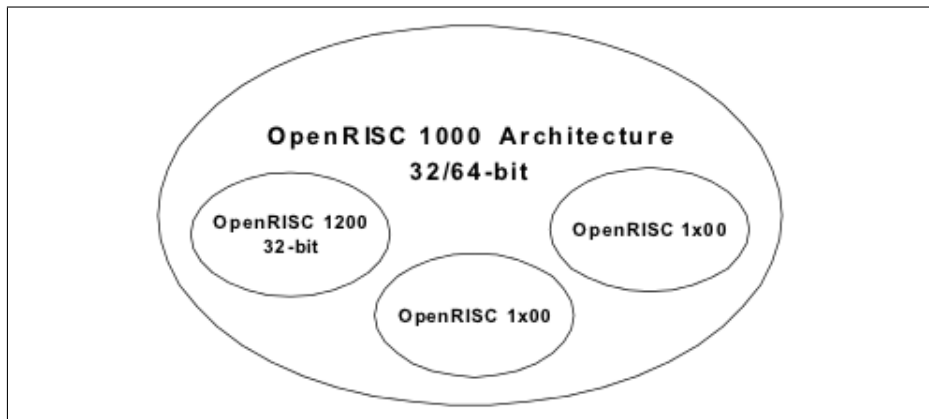
1 Εισαγωγή

2 OpenRISC 1200 IP Core

2.1 Εισαγωγή

2.1.1 OpenRISC Οικογένεια

Η OpenRISC 1000 οικογένεια επεξεργαστών αναφέρεται σε μια ελεύθερη, ανοιχτού λογισμικού RISC αρχιτεκτονική κεντρικών μονάδων επεξεργασίας. Σχετικά με την αρχιτεκτονική, η OpenRISC 1000 οικογένεια στοχεύει σε ένα φάσμα υλοποιήσεων που ποικίλουν ως προς την τιμή/απόδοση και το είδος της εφαρμογής. Είναι μια 32/64-bit φόρτωσης και αποθήκευσης (load and store) RISC αρχιτεκτονική που σχεδιάστηκε με έμφαση στην απόδοση, στην απλότητα, στην χαμηλή ενεργειακή κατανάλωση, στην επεκτασιμότητα και στην ευελιξία. Η OpenRISC αρχιτεκτονική στοχεύει σε μεσαία και υψηλή απόδοση δικτύωσης (networking), σε ενσωματωμένα, αυτοκινητοβιομηχανικά και φορητά υπολογιστικά περιβάλλοντα.



Σχήμα 2.1.1: Σχηματική αναπαράσταση OpenRISC αρχιτεκτονικής.

Όλες οι OpenRISC υλοποιήσεις που το πρώτο ψηφίο στον αριθμό ταυτότητας (identification number) είναι '1' ανήκουν στην OpenRISC 1000 οικογένεια. Το δεύτερο ψηφίο ορίζει ποια χαρακτηριστικά της OpenRISC 1000 αρχιτεκτονικής είναι υλοποιημένα και με ποιο τρόπο είναι υλοποιημένα. Τα δύο τελευταία ψηφία αναφέρονται στο πως μια υλοποίηση ήταν παραμετροποιημένη πριν χρησιμοποιηθεί σε πραγματική εφαρμογή.

2.1.2 OpenRISC 1200

Ο OR1200 είναι ένας 32-bit βαθμωτός RISC επεξεργαστής με Harvard μικρο-αρχιτεκτονική, 5 stage integer pipeline, υποστήριξη εικονικής μνήμης (MMU) και βασικές δυνατότητες DSP.

Οι προκαθορισμένες κρυφές μνήμες είναι:

- 1-way direct-mapped 8KB κρυφή μνήμη δεδομένων.
- 1-way direct-mapped 8KB κρυφή μνήμη εντόλων.

- Κάθε κρυφή μνήμη έχει γραμμή μεγέθους 16-byte.
- Και οι δύο κρύφες μνήμες είναι physically tagged(todo).

Η προκαθορισμένη MMU αποτελείται από:

- 64-entry hash based 1-way direct-mapped data TLB.
- 64-entry hash based 1-way direct-mapped instruction TLB.

Μερικές άλλες επιπρόσθετες λειτουργίες που παρέχει ο OpenRISC 1200 είναι η μονάδα απασφαλμάτωσης πραγματικού χρόνου (real-time debug unit), υψηλής ανάλυσης χρονιστή, προγραμματιζόμενο ελεκτή διακοπών (programmable interrupt controller) και μονάδα ρύθμισης των ενεργειακών απαιτήσεων.

Ο OR1200 ουσιαστικά προορίζεται για εφαρμογές σε ενσωματωμένα, φορητά και δικτύωσης συστήματα. Μπορεί να ανταγωνιστεί τους τελευταίους βαθμωτούς 32-bit RISC επεξεργαστές της κλάσης του και να υποστηρίξει αποδοτικά οποιοδήποτε μοντέρνο λειτουργικό σύστημα. Ανταγωνιστές του θεωρούνται οι ARM10, ARC και Tensilica RISC επεξεργαστές.

Συνοπτικά παρουσιάζονται παρακάτω τα χαρακτηριστικά του OR1200:

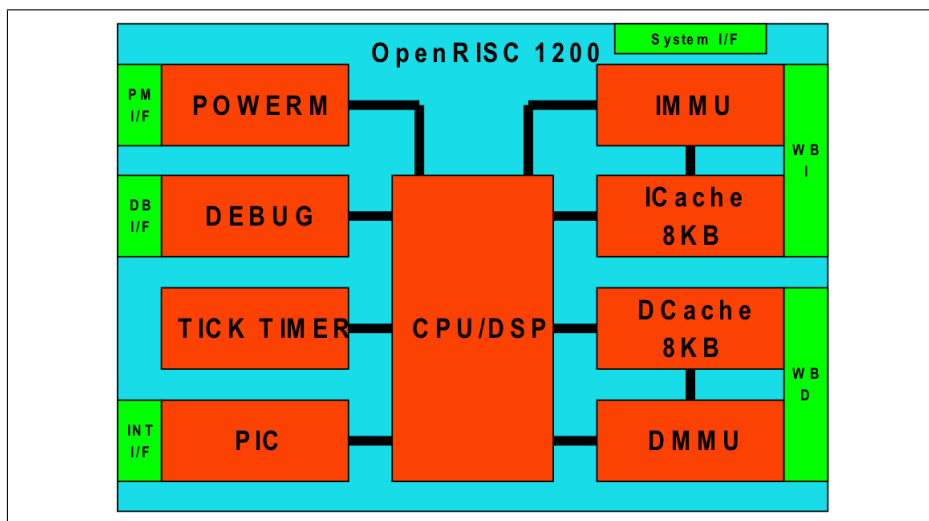
OR 1200	
License	<i>GNU LGPL</i>
Platform	<i>FPGA, ASIC</i>
Distributed file format	<i>Verilog</i>
General	
Architecture	<i>32-bit RISC</i>
Byte Ordering	<i>Big endian</i>
Pipeline depth	<i>5</i>
Issue type	<i>Single</i>
Register file	
Organization	<i>Flat</i>
# of global registers	<i>32</i>
Total # of GPRs	<i>32</i>
ISA	
Type	<i>ORBIS32</i>
Addressing modes	<i>Immediate, displacement, pcrelative</i>
MAC	<i>32x32-bit, 48-bit Acc</i>
Custom instructions	<i>Yes</i>
Custom coprocessor	<i>Yes</i>
Software floating-point support	<i>IEEE-754 Single and double precision</i>
Cache	
Hierarchy	<i>Harvard</i>
Instruction cache size	<i>512 byte-8 Kbyte</i>
Data cache size	<i>4-8 Kbyte</i>
Line size	<i>8-16 byte</i>
Placement scheme	<i>Direct-mapped</i>
Valid bits	<i>One per cache line</i>
Line-locking	<i>Set basis</i>
System Interface	<i>Wishbone SoC rev. B32-bit</i>
Power Management	<i>Slow and idle mode, sleep mode, doze mode</i>
Memory	
On-chip RAM	<i>Configurable</i>
Operating system support	<i>Linux, uClinux, OAR RTEMS RTOS</i>

Πίνακας 2.1.1: Overview of OR1200 specifications

2.2 Αρχιτεκτονική

Στο παρακάτω σχήμα βρίσκεται η σχηματική αναπαράσταση του OR1200 επεξεργαστή όπου αποτελείται από τις εξής υπομονάδες:

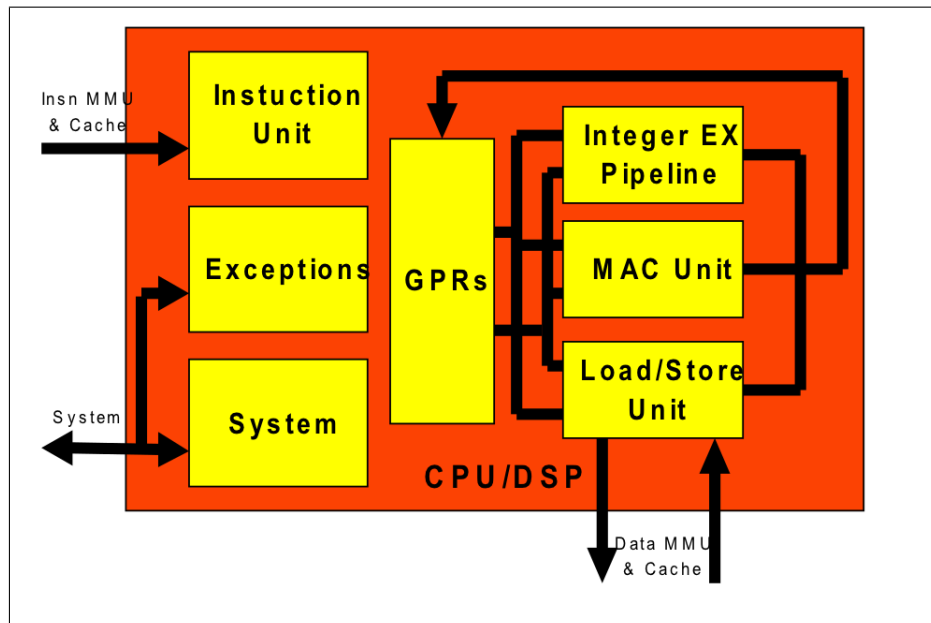
- CPU/FPU/DSP central block.
- Direct-mapped data cache.
- Direct-mapped instruction cache.
- Data MMU based on hash based DTLB.
- Instruction MMU based on hash based ITLB.
- Power management unit and power management interface.
- Tick timer.
- Debug unit and development interface.
- Interrupt controller and interrupt interface.
- Instruction and Data WISHBONE host interfaces.



Σχήμα 2.2.1: Core's Architecture

2.2.1 CPU/DSP

Το μπλόκ CPU/FPU/DSP είναι ένα κεντρικό κομμάτι του επεξεργαστή OR1200 RISC. Το σχήμα 2.3 δείχνει το βασικό μπλόκ διάγραμμα του CPU/DSP (δεν απεικονίζεται η μονάδα FPU). Η μονάδα του OR1200 CPU/FPU/DSP υλοποιεί μόνο τις ORBIS32 και ORFPX32 ομάδες εντολών. Δεν υπάρχει ακόμα υλοποίηση για τις ομάδες εντολών ORBIS64, ORFBX64 και ORVDX64.



Σχήμα 2.2.2: CPU/DSP block diagram.

Μονάδα Εντολών

Η μονάδα εντολών (instruction unit) υλοποιεί την βασική pipeline διαδικασία, φέρνει τις εντολές από το υποσύστημα της μνήμης, τις αποστέλλει στις διαθέσιμες μονάδες εκτέλεσης (execution unit) και διατηρεί ένα ιστορικό καταστάσεων ώστε να εξασφαλίσει ένα ακριβές μοντέλο εξαιρέσεων (exception model) για τον σωστό τερματισμό των εργασιών. Επίσης εκτελεί τις υπό όρους και άνευ όρων εντολές διακλάδωσης.

Παράλληλα μπορεί να στείλει διαδοχικές εντολές σε κάθε ρολόι αν η κατάλληλη μονάδα εκτέλεσης είναι διαθέσιμη. Η μονάδα εκτέλεσης είναι αυτή που πρέπει να διακρίνει αν τα δεδομένα που χρειάζεται η εντολή είναι διαθέσιμα και να διασφαλίσει ότι καμία άλλη εντολή δεν χρησιμοποιεί εκείνη την στιγμή τον ίδιο καταχωρητή.

Τέλος η μονάδα εντολών χειρίζεται μόνο τις εντολές της κλάσης ORBIS32 και κατά επιλογή ένα υποσύνολο των εντολών ORFPX32. Μερικές εντολές της κλάσης ORFPX32 και όλες οι εντολές των ORFPX3264 και ORVDX64 δεν υποστηρίζονται ακόμα από τον OR1200.

Καταχωρητές Γενικού Σκοπού

Ο OpenRISC 1200 χρησιμοποιεί 32 καταχωρητές γενικού σκοπού (general-purpose registers) των 32 δυαδικών ψηφίων. Η αρχιτεκτονική του επίσης υποστηρίζει σκιώδη αντίγραφα (shadow copies) του αρχείου καταχωρητών (register file) ώστε να υλοποιεί γρήγορη εναλλαγή μεταξύ των πλαισίων εργασίας (working contexts), όμως αυτή η λειτουργία δεν υποστηρίζεται από την παρούσα υλοποίηση του OR1200.

Ο OR1200 υλοποιεί το αρχείο καταχωρητών γενικού σκοπού σαν δυο σύγχρονες μνήμες διπλής θύρας με χωρητικότητα 32 λέξεων των 32 bit ανά λέξη.

Μονάδα Load/Store

Η μονάδα Load/Store (LSU) μεταφέρει όλα τα δεδομένα μεταξύ των καταχωρητών γενικού σκοπού (GPRs) και τον επεξεργαστή διαμέσου του εσωτερικού διαύλου επικοινωνίας. Έχει υλοποιηθεί σαν μια ανεξάρτητη μονάδα εκτέλεσης ώστε οι καθυστερήσεις στο υποσύστημα μνήμης λόγω εξαρτήσεων μεταξύ δεδομένων να επηρεάζει μόνο την pipeline διαδικασία.

Τα κυριότερα χαρακτηριστικά της μονάδας Load/Store παρουσιάζονται παρακάτω:

- όλες οι εντολές Load/Store είναι υλοποιημένες σε επίπεδο hardware (atomic instructions include)
- pipeline λειτουργία
- ευθυγραμμισμένες προσπελάσεις στην μνήμη
- address entry buffer

Όταν κάποια εντολή Load ή Store προορίζεται για εκτέλεση τότε η μονάδα Load/Store (LSU) διερευνά αν όλα τα τελούμενα είναι διαθέσιμα. Τα τελούμενα που ελέγχει είναι τα παρακάτω:

- Οι διευθύνσεις των καταχωρητών που θα χρησιμοποιηθούν.
- Τα δεδομένα στους καταχωρητές που προορίζονται για εκτέλεση (store εντολή).
- Τα δεδομένα στους καταχωρητές που προορίζονται για αποθήκευση (load εντολή).

Πράξεις ακεραίων

Ο πυρήνας του OR1200 υποστηρίζει τα παρακάτω είδη εντολών για πράξεις μεταξύ 32 bit ακεραίων (integer execution pipeline):

- Αριθμητικές εντολές.
- Εντολές σύγκρισης.
- Λογικές εντολές.
- Εντολές ολίσθησης και περιστροφής.

Οι περισσότερες εντολές μπορούν να εκτελεστούν σε ένα κύκλο(TODO).

Μονάδα MAC

Η μονάδα MAC εκτελεί λειτουργίες DSP MAC. Οι MAC λειτουργίες είναι 32x32 με 48 bit συσσωρευτή. Η μονάδα MAC είναι υλοποιημένη με την τεχνική pipeline και μπορεί να δεχτεί καινούργιες MAC εργασίες σε κάθε καινούργιο κύκλο.

Μονάδα Κινητής Υποδιαστολής

Η υλοποίηση της μονάδας Κινητής Υποδιαστολής(FPU) είναι βασισμένη σε δύο άλλες FPUs που παρέχονται από το OpenCores.org. Για τις συναρτήσεις σύγκρισης και μετατροπής κομμάτια υλοποίησης έχουν παρθεί από την FPU που έχει σχεδιαστεί από τον Rudolf Usselman, ενώ για τις αριθμητικές λειτουργίες από το πρότζεκτ fpu100 όπου ο Jidan Al-Eryani τις μετέτρεψε σε γλώσσα Verilog HDL.

Όλες οι εντολές ORFPX32 εκτός από τις *lf.madd.s* και *lf.rem.s* υποστηρίζονται από την FPU όταν αυτή έχει ενεργοποιηθεί από τις ρυθμίσεις του OR1200.

Μονάδα Συστήματος

Η μονάδα συστήματος συνδέει όλα τα σήματα της CPU/FPU/DSP μονάδας που δεν είναι συνδεδεμένα με τις διεπαφές (interfaces) των εντολών και των δεδομένων. Παράλληλα υλοποιεί όλους τους καταχωρητές ειδικού σκοπού (π.χ. supervisor registers).

Μονάδα Εξαιρέσεων

Οι εξαιρέσεις του πυρήνα μπορούν να δημιουργηθούν όταν μια συνθήκη εξαίρεσης παρουσιάζεται. Παρακάτω φαίνονται ποιες είναι αυτές οι συνθήκες για τον OR1200.

- Εξωτερικά αιτήματα διακοπών (external interrupt request).
- Ορισμένες συνθήκες πρόσβασης στην μνήμη.
- Εσωτερικά λάθι όπως η προσπάθεια εκτέλεσης μη υλοποιημένης εντολής (undefined opcode).
- Εσωτερικές εξαιρέσεις όπως σημεία διακοπής (breakpoints).

Η μονάδα διαχείρισης εξαιρέσεων είναι αόρατη από το λογισμικό του χρήστη και χρησιμοποιεί τον ίδιο μηχανισμό για να εξυπηρετεί όλα τα είδη των εξαιρέσεων. Όταν μία εξαίρεση παρουσιάζεται τότε ο έλεγχος μεταφέρεται στον διαχειριστή εξαιρέσεων με μία προκαθορισμένη μετατόπιση ανάλογα με το είδος της εξαίρεσης που παρουσιάστηκε. Οι εξαιρέσεις διαχειρίζονται από το supervisor mode.

2.2.2 Κρυφή Μνήμη Δεδομένων

Η προκαθορισμένες ρυθμίσεις για την κρυφή μνήμη δεδομένων για τον OR1200 είναι 8-KByte, 1-τρόπων πλήρους απεικόνισης (1-way direct-mapped), οι οποίες επιτρέπουν στον πυρήνα ταχεία προσπέλαση των δεδομένων. Παρόλα αυτά η κρυφή μνήμη μπορεί να παραμετροποιηθεί σύμφωνα με τον Πίνακα (TODO).

	Direct mapped
16B/line, 256 lines, 1 way	4KB
16B/line, 512 lines, 1 way	8KB (default)
16B/line, 1024 lines, 1 way	16KB
32B/line, 1024 lines, 1 way	32KB

Πίνακας 2.2.1: Ρυθμίσεις μεγέθους κρυφής μνήμης δεδομένων

Ο OR1200 παρέχει την δυνατότητα η κρυφή μνήμη δεδομένων να χρησιμοποιεί είτε την write-through είτε την write-back στρατηγική, όμως η write-back στρατηγική είναι ακόμα σε πειραματικό στάδιο.

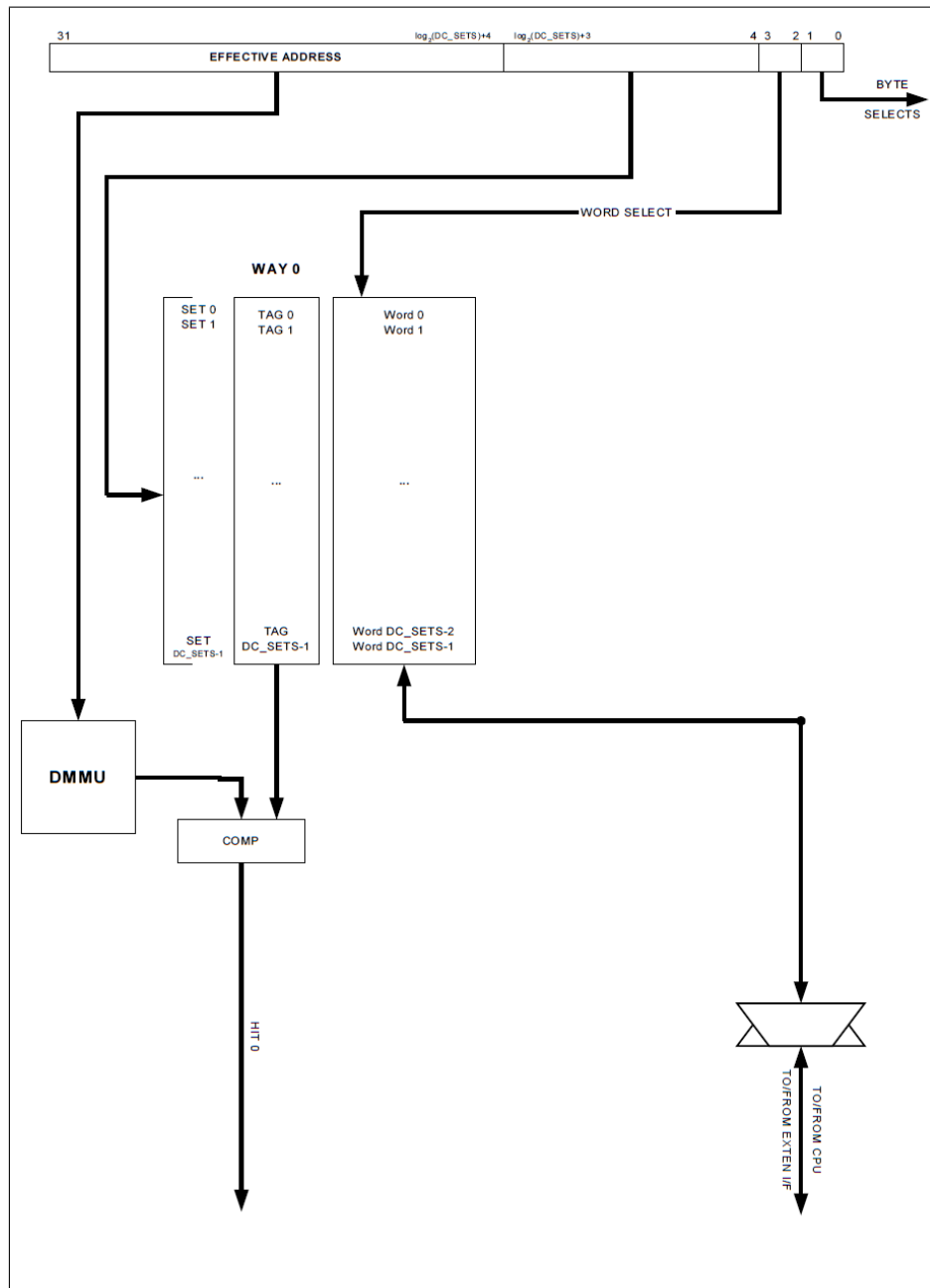
Χαρακτηριστικά:

- Η κρυφή μνήμη δεδομένων είναι ξεχωριστά υλοποιημένη από την κρυφή μνήμη εντολών (Harvard αρχιτεκτονική)
- Η κρυφή μνήμη δεδομένων χρησιμοποιεί τον αλγόριθμο "Λιγότερο Πρόσφατα Χρησιμοποιημένο" για την αντικατάσταση των πλαισίων.
- Ο κατάλογος της κρυφής μνήμης είναι φυσικά διευθυνσιοδοτημένος (physically addressed). Η ετικέτα της φυσικής διεύθυνσης είναι αποθηκευμένη στον κατάλογο της κρυφής μνήμης.
- Write-through ή write-back στρατηγική.
- Ολόκληρη η κρυφή μνήμη μπορεί να απενεργοποιηθεί, γραμμές να ακυρωθούν, να καθαριστούν ή να γραφτούν πίσω, γράφοντας στους καταχωρητές ειδικού σκοπού της κρυφής μνήμης.

Κατά την αστοχία της κρυφής μνήμης και με τις κατάλληλες συνθήκες, η γραμμή της κρυφής μνήμης γεμίζει ή αδειάζει (written back στρατηγική) με ριπές (burst) των 16-byte. Οι ριπές πραγματοποιούνται σαν κρίσιμη-λέξη-πρώτα (critical-word-first) λειτουργία, η κρίσιμη γράφεται στην κρυφή μνήμη και ταυτόχρονα προωθείται στην μονάδα αιτήσεων (requesting unit), έτσι μειώνονται οι καθυστερήσεις που προκαλούνται από το γέμισμα της γραμμής. Η κρυφή μνήμη παρέχει αποθηκευτικό χώρο για τις ετικέτες και εκτελεί συναρτήσεις αντικατάστασης γραμμών.

Η κρυφή μνήμη δεδομένων είναι στενά συνδεδεμένη με μια εξωτερική διεπαφή ώστε να επιτρέπει την αποδοτική πρόσβαση στον ελεγκτή μνήμης.

Παράλληλα η κρυφή μνήμη δεδομένων προμηθεύει τα δεδομένα στον καταχωρητές γενικού σκοπού διαμέσου της διεπαφής των 32-bit της μονάδας Load/Store. Η μονάδα Load/Store παρέχει όλη την απαραίτητη λογική για τον υπολογισμό της αποτελεσματικής διεύθυνσης (effective addresses), την κατάλληλη αλληλουχία για τις λειτουργίες load/store και χειρίζεται την ευθυγράμμιση των δεδομένων από και προς την κρυφή μνήμη δεδομένων. Οι λειτουργίες εγγραφής στην κρυφή μνήμη δεδομένων μπορούν να γίνουν με βάση το 1 byte, την μισή-λέξη (half-word) ή μια λέξη (word).



Σχήμα 2.2.3: Οργάνωση κρυφής μνήμης δεδομένων.

Κάθε γραμμή περιέχει τέσσερις συνεχόμενες λέξεις από τη μνήμη που έχουν φορτωθεί από μια οριοθετημένη γραμμή ως προς το μέγεθός της. Ως αποτέλεσμα, οι γραμμές της κρυφής μνήμης είναι ευθυγραμμισμένες με τα όρια της σελίδας.

2.2.3 Κρυφή Μνήμη Εντολών

Η προκαθορισμένες ρυθμίσεις για την κρυφή μνήμη εντολών για τον OR1200 είναι 8-KByte, 1-τρόπων πλήρους απεικόνισης (1-way direct-mapped), οι οποίες επιτρέπουν στον πυρήνα ταχεία προσπέλαση στις εντολές. Παρόλα αυτά η κρυφή μνήμη μπορεί να παραμετροποιηθεί σύμφωνα με τον Πίνακα (TODO).

	Direct mapped
16B/line, 32 lines, 1 way	512B
16B/line, 256 lines, 1 way	4KB
16B/line, 512 lines, 1 way	8KB (default)
16B/line, 1024 lines, 1 way	16KB
32B/line, 1024 lines, 1 way	32KB

Πίνακας 2.2.2: Ρυθμίσεις μεγέθους κρυφής μνήμης εντολών

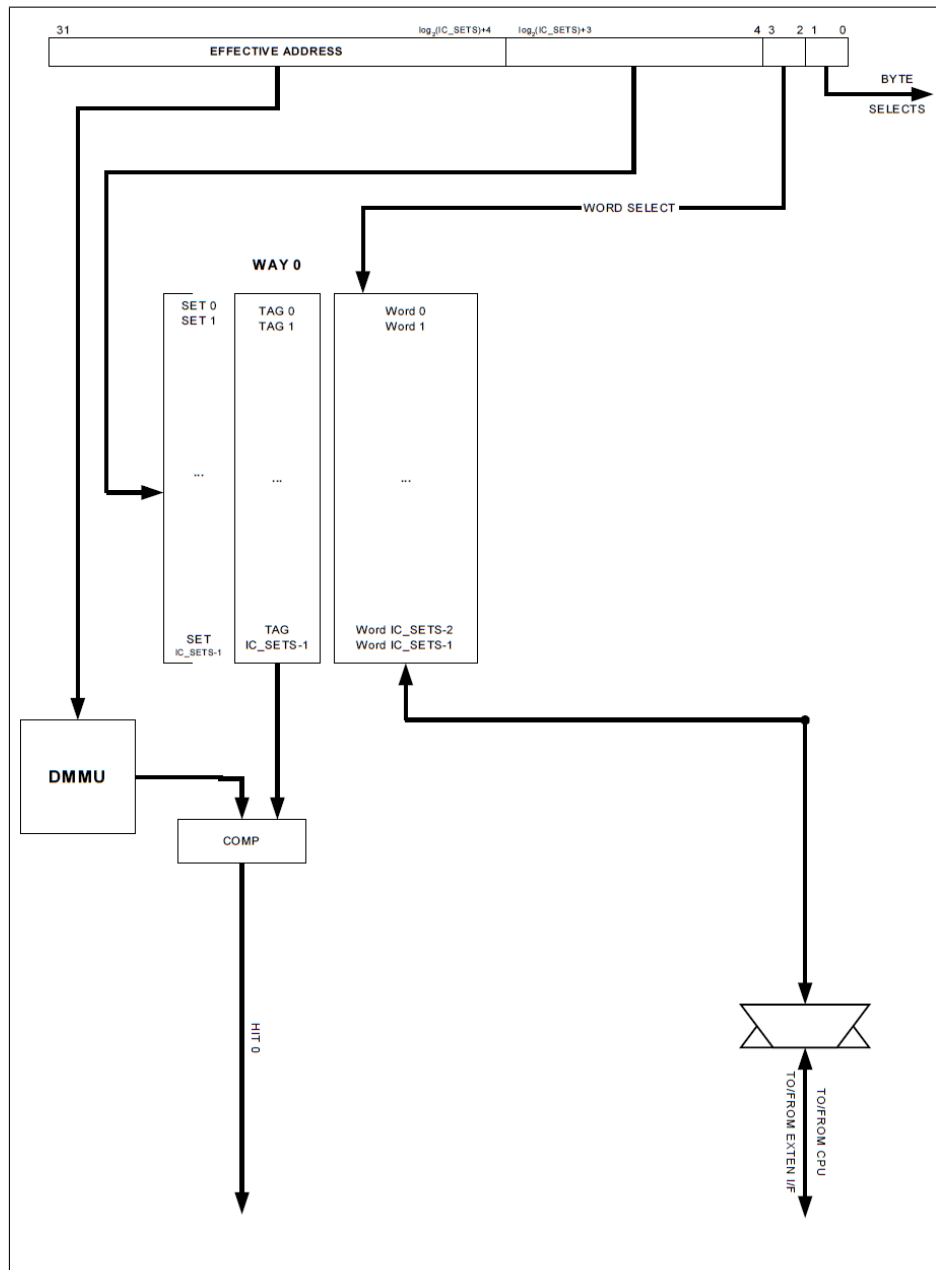
Χαρακτηριστικά:

- Η κρυφή μνήμη εντολών είναι ξεχωριστά υλοποιημένη από την κρυφή μνήμη δεδομένων (Harvard αρχιτεκτονική).
- Η κρυφή μνήμη εντολών χρησιμοποιεί τον αλγόριθμο "Λιγότερο Πρόσφατα Χρησιμοποιημένο" για την αντικατάσταση των πλαισίων.
- Ο κατάλογος της κρυφής μνήμης είναι φυσικά διευθυνσιοδοτημένος (physically addressed). Η ετικέτα της φυσικής διεύθυνσης είναι αποθηκευμένη στον κατάλογο της κρυφής μνήμης.
- Μπορεί να απενεργοποιηθεί ή να ακυρωθεί/ γράφοντας στους καταχωρητές ειδικού σκοπού της κρυφής μνήμης.

Κατά την αστοχία της κρυφής μνήμης η γραμμή της κρυφής μνήμης γεμίζει ή με ριπές (burst) των 16-byte. Οι ριπές πραγματοποιούνται σαν κρίσιμη-λέξη-πρώτα (critical-word-first) λειτουργία: η κρίσιμη γράφεται στην κρυφή μνήμη και ταυτόχρονα προωθείται στην μονάδα αιτήσεων (requesting unit), έτσι μειώνονται οι καθυστερήσεις που προκαλούνται από το γέμισμα της γραμμής. Η κρυφή μνήμη εντολών παρέχει αποθηκευτικό χώρο για τις ετικέτες και εκτελεί συναρτήσεις αντικατάστασης γραμμών.

Η κρυφή μνήμη εντολών είναι στενά συνδεδεμένη με μια εξωτερική διεπαφή ώστε να επιτρέπει την αποδοτική πρόσβαση στον ελεγκτή μνήμης.

Παράλληλα η κρυφή μνήμη εντολών προμηθεύει τις εντολές προς εκτέλεση διαμέσου της διεπαφής των 32-bit της υπομονάδας προσκόμισης εντολών. Η υπομονάδα προσκόμισης εντολών παρέχει όλη την απαραίτητη λογική για τον υπολογισμό της αποτελεσματικής διεύθυνσης (effective addresses).



Σχήμα 2.2.4: Οργάνωση κρυφής μνήμης εντολών

Κάθε γραμμή περιέχει τέσσερις συνεχόμενες λέξεις από τη μνήμη που έχουν φορτωθεί από μια οριοθετημένη γραμμή ως προς το μέγεθός της. Ως αποτέλεσμα, οι γραμμές της κρυφής μνήμης είναι ευθυγραμμισμένες με τα όρια της σελίδας.

2.2.4 Διαχείριση Μνήμης (MMU) Δεδομένων

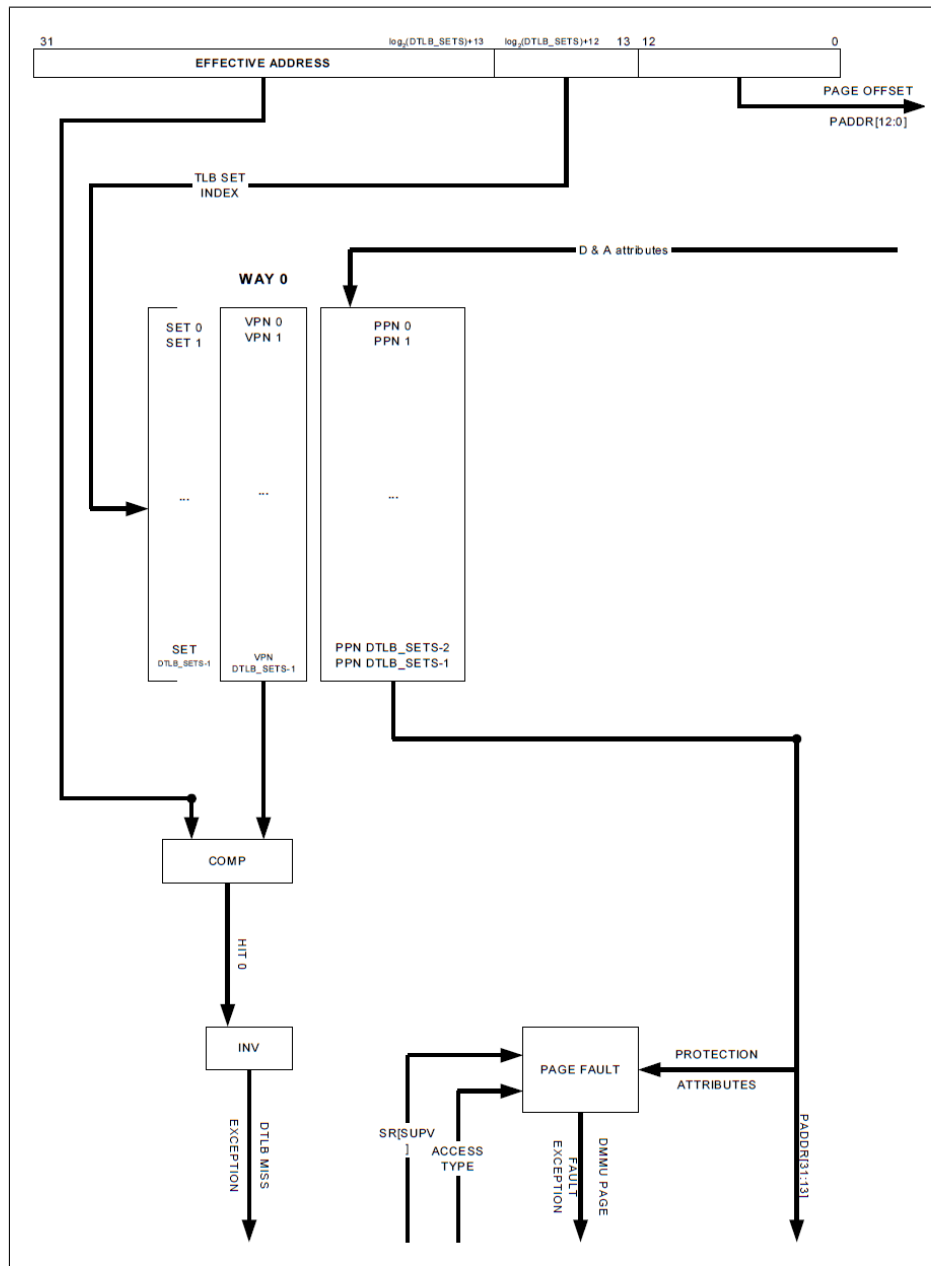
Η OR1200 υλοποιεί ένα εικονικό σύστημα διαχείρισης μνήμης (memory management unit - MMU) που παρέχει προστασία κατά την πρόσβαση στην μνήμη και αποτελεσματική μετάφραση σε φυσικές διευθύνσεις. Η διακριτότητα της προστασίας είναι όπως ορίζεται από την αρχιτεκτονική OpenRISC 1000 8-Kbyte και 16-Mbyte σελίδες.

	Direct mapped
16 entries per way	16 DTLB entries
32 entries per way	32 DTLB entries
64 entries per way	64 DTLB entries (default)
128 entries per way	128 DTLB entries

Πίνακας 2.2.3: Ρυθμίσεις Δεδομένων TLB (translation lookaside buffer)

Χαρακτηριστικά:

- Η MMU δεδομένων είναι ξεχωριστή από την MMU εντόλων.
- Το μέγεθός της σελίδας είναι 8-KByte.
- Ολοκληρωμένο σύστημα προστασίας σελίδας.
- Πλήρης απεικόνιση κατακερματισμού βασισμένο στον translation lookaside buffer (DTLB) με προκαθορισμένο 1-τροπο συσχέτισης και τα παρακάτω χαρακτηριστικά:
 - Παροχή εξαιρέσεων στην περίπτωση εξαιρέσεων και λαθών.
 - Software tablewalk.
 - Υψηλή απόδοση λόγω του κατακερματισμού.
 - Τροποποιήσιμο αριθμό καταχωρήσεων στο DLTB με προκαθορισμένες τις 64 καταχωρήσεις ανά τρόπο.



Σχήμα 2.2.5: Οργάνωση MMU δεδομένων.

Η υλοποίηση της MMU σε υλικό υποστηρίζει δύο-επιπέδων software tablewalk.

2.2.5 Διαχείριση Μνήμης (MMU) Εντολών

Η OR1200 υλοποιεί ένα εικονικό σύστημα διαχείρισης μνήμης (memory management unit - MMU) που παρέχει προστασία κατά την πρόσβαση στην μνήμη και αποτελεσματική μετάφραση σε φυσικές διευθύνσεις. Η διακριτότητα της προστασίας είναι

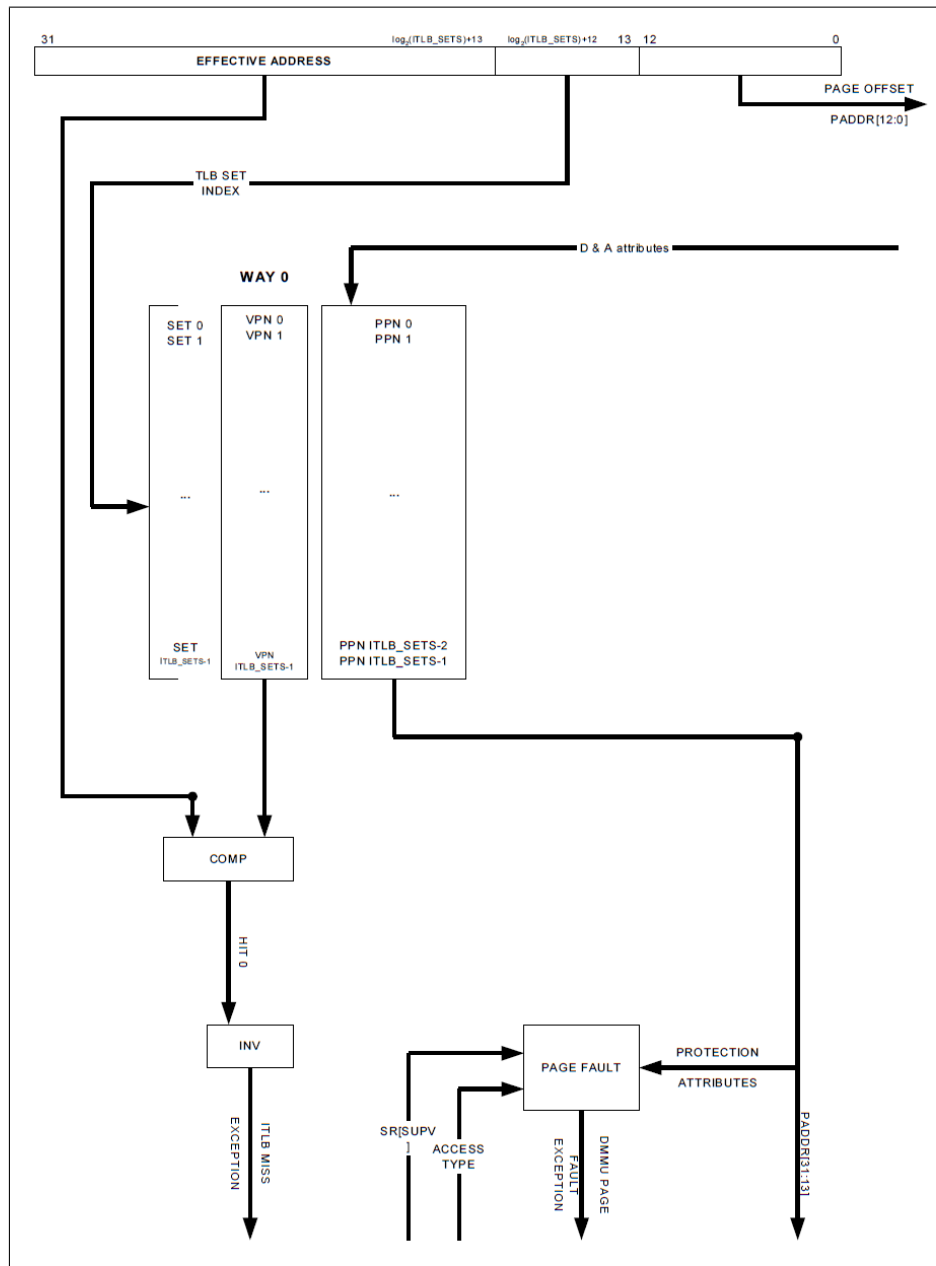
όπως ορίζεται από την αρχιτεκτονική OpenRISC 1000 8-Kbyte και 16-Mbyte σελίδες.

	Direct mapped
16 entries per way	16 ITLB entries
32 entries per way	32 ITLB entries
64 entries per way	64 ITLB entries (default)
128 entries per way	128 ITLB entries

Πίνακας 2.2.4: Ρυθμίσεις Εντολών TLB (translation lookaside buffer)

Χαρακτηριστικά:

- Η MMU εντολών είναι ξεχωριστή από την MMU δεδομένων.
- Το μέγεθός της σελίδας είναι 8-KByte.
- Ολοκληρωμένο σύστημα προστασίας σελίδας.
- Πλήρης απεικόνιση κατακερματισμού βασισμένο στον translation lookaside buffer (ITLB) με προκαθορισμένο 1-τροπο συσχέτισης και τα παρακάτω χαρακτηριστικά:
 - Παροχή εξαιρέσεων στην περίπτωση εξαιρέσεων και λαθών.
 - Software tablewalk.
 - Υψηλή απόδοση λόγω του κατακερματισμού.
 - Τροποποιήσιμο αριθμό καταχωρήσεων στο ILTB με προκαθορισμένες τις 64 καταχωρήσεις ανά τρόπο.

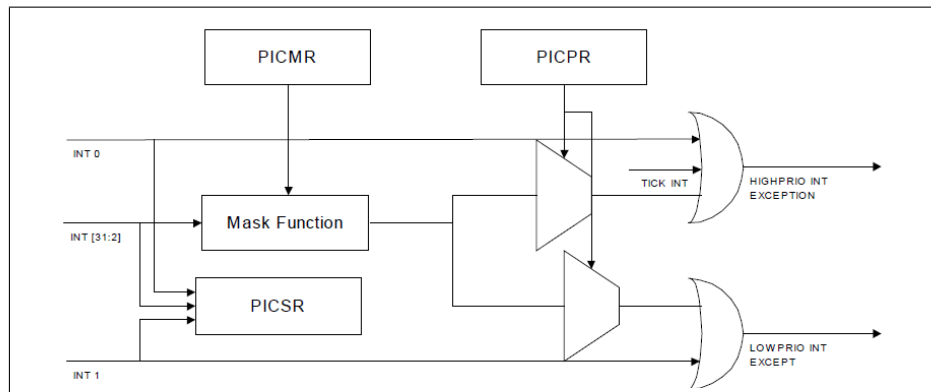


Σχήμα 2.2.6: Οργάνωση MMU εντολών.

Η υλοποίηση της MMU σε υλικό υποστηρίζει δύο-επιπέδων software tablewalk.

2.2.6 Προγραμματιζόμενος Ελεγκτής Διακοπών

Ο ελεγκτής διακοπών (Interrupt controller) λαμβάνει διακοπές από εξωτερικές πηγές και τις προωθεί ανάλογα με την προτεραιότητά τους (χαμηλή-υψηλή) σαν εξαιρέσεις (exception) στον πυρήνα του επεξεργαστή.



Σχήμα 2.2.7: Μπλοκ διάγραμμα του ελεγκτή διακοπών.

Ο προγραμματιζόμενος ελεγκτής διακοπών έχει τρεις καταχωρητές ειδικού σκοπού και 32 σήματα εισόδου διακοπών. Τα σήματα εισόδου διακοπών 0 και 1 είναι πάντα ενεργοποιημένα και συνδεδεμένα με την υψηλή και την χαμηλή προτεραιότητα εισόδου αντίστοιχα, όπως φαίνεται και στο σχήμα (TODO).

Υπάρχουν 30 ακόμα σήματα εισόδου που μπορούν να φιλτράριστούν και να ανατεθεί προτεραιότητά διαμέσου του προγραμματισμού των καταχωρητών ειδικού σκοπού.

2.2.7 Tick Timer

Ο επεξεργαστής OR1200 έχει υλοποιημένη λειτουργία χρονιστή (tick timer). Βασικά αυτός είναι ένας χρονοδιακόπτης που χρονίζεται από το ρολόι του επεξεργαστή και χρησιμοποιείται από το λειτουργικό σύστημα για ακριβείς μετρήσεις χρόνου και για τον χρονοπρογραμματισμό των διεργασιών τους συστήματος.

Ο OR1200 ακολουθεί αυστηρά τον αρχιτεκτονικό ορισμό του χρονιστή όπου:

- Μέγιστες μετρήσεις του χρονιστή 2^{32} κύκλους ρολογιού.
- Μέγιστη χρονική περίοδο, 2^{28} κύκλους ρολογιού μεταξύ διακοπών.
- Φιλτραρισμένο σήμα διακοπής χρονιστή (Maskable tick timer interrupt).
- Δυνατότητα επανεκκίνησης, μονής και συνεχούς μέτρησης.

2.2.8 Διαχείριση Ενεργειακών Απαιτήσεων

Για την βελτιστοποίηση της κατανάλωσης ενέργειας, ο OR1200 παρέχει καταστάσεις (mode) χαμηλής ενεργειακής κατανάλωσης που μπορούν να χρησιμοποιηθούν ώστε δυναμικά να ενεργοποιούνται και να απενεργοποιούνται ορισμένες εσωτερικές ενότητες (module).

Ο OR1200 έχει τρεις κύριες λειτουργίες για την ελαχιστοποίηση της κατανάλωσης ενέργειας:

- Slow και Idle καταστάσεις (SW controlled clock freq reduction)
- Doze και Sleep καταστάσεις (interrupt wake-up)

Power Minimization Feature	Approx Power Consumption Reduction
Slow and Idle mode	2x – 10x
Doze mode	100x
Sleep mode	200x
Dynamic clock gating	N/A

Πίνακας 2.2.5: Καταστάσεις βελτιστοποίησης κατανάλωσης ενέργειας.

Η κατάσταση Slow Down εκμεταλλεύεται τους low-power διαιρέτες από το εξωτερικό κύκλωμα παραγωγής ρολογιού ώστε να επιτυγχάνει πλήρης λειτουργικότητα αλλά σε χαμηλότερη συχνότητα, εξοικονομώντας έτσι ενέργεια.

Όταν το λογισμικό εκκινεί την κατάσταση Doze, τότε τα προγράμματα που τρέχουν στον επεξεργαστή αναστέλλονται. Τα ρολόγια στις εσωτερικές ενότητες (modules) του επεξεργαστή απενεργοποιούνται εκτός του χρονιστή (tick timer). Ωστόσο οποιοδήποτε άλλο μπλόκ πάνω στο ολοκληρωμένο κύκλωμα συνεχίζει να λειτουργεί κανονικά. Ο επεξεργαστής OR1200 θα φύγει από την κατάσταση doze και θα επανέλθει σε κανονική κατάσταση όταν κάποια διακοπή (interrupt) παρουσιαστεί.

Στην κατάσταση Sleep, όλες οι εσωτερικές μονάδες του OR1200 απενεργοποιούνται και τα ρολόγια οριοθετούνται (clocks gated). Προαιρετικά (ανάλογα με την υλοποίηση) μπορεί να χαμηλωθεί η παροχή της τάσης στον πυρήνα του OR1200. Ο επεξεργαστής OR1200 θα φύγει από την κατάσταση doze και θα επανέλθει σε κανονική κατάσταση όταν κάποια διακοπή (interrupt) παρουσιαστεί.

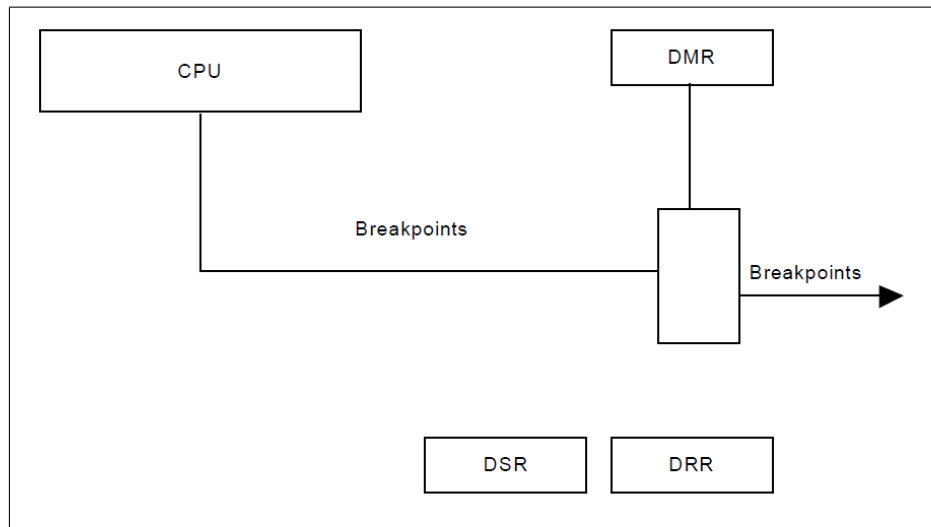
Η λειτουργία Dynamic Clock gating δεν υποστηρίζεται προς το παρόν από τον OR1200.

2.2.9 Μονάδα Αποσφαλμάτωσης

Η μονάδα αποσφαλμάτωσης βοηθάει τους προγραμματιστές λογισμικού διορθώσουν λάθοι στο σύστημα τους. Παρέχει την βασική βοήθεια για αποσφαλμάτωση, χωρίς να παρέχει όμως προηγμένη τεχνολογία σύμφωνα με την αρχιτεκτονική του OpenRISC 1000 όπως watchpoints, breakpoints και πρόγραμμα ελέγχου της ροής των καταχωρητών ελέγχου.

2.2.10 Σήματα Χρονισμού και Επανεκκίνησης

(TODO) Ο πυρήνας του OR1200 χρησιμοποιεί ένα ρολόι για τον χρονισμό κάθε μιας διεπαφής του διαύλου επικοινωνίας Wishbone τόσο για τα δεδομένα όσο και για τις εντολές. Το σήμα ρολογιού *clk_cpu* χρονίζει οτιδήποτε βρίσκεται μέσα στην



Σχήμα 2.2.8: Μπλόκ διάγραμμα μονάδας αποσφαλμάτωσης.

διεπαφή του διαύλου Wishbone. Η διεπαφή διαύλου δεδομένων Wishbone χρονίζεται από το σήμα *dwd_clk_i*, ενώ των εντολών με το σήμα *iwd_clk_i*.

Ο επεξεργαστής OR1200 παρέχει ένα ασύγχρονο σήμα επανεκκίνησης του συστήματος (asynchronous reset signal), *rst*. Όταν αυτό σήμα είναι ενεργοποιημένο τότε αμέσως επαναφέρει (resets) όλα τα flip-flops μέσα στον OR1200. Όταν είναι απενεργοποιημένο τότε ο επεξεργαστής λειτουργεί κανονικά.

2.2.11 Διάυλος επικοινωνίας Wishbone

Δύο διεπαφές Wishbone του επεξεργαστή OR1200 συνδέουν τον πυρήνα με τα περιφερειακά και το εξωτερικό υποσύστημα μνήμης. Αυτές είναι συμβατές το WISHBONE SoC Interconnection specification Rev. B3. Η υλοποίηση παρέχει ένα δίαυλο επικοινωνίας πλάτους 32 bit χωρίς να υποστηρίζει άλλα πλάτη.

(TODO)

3 OR1200 Simulation

3.1 Simulation Enviroment

Υπάρχουν δύο περιβάλλοντα με τα οποία γίνεται η εξομοίωση του OR1200 επεξεργαστή. Το πρώτο χρησιμοποιεί τον OpenRISC αρχιτεκτονικό εξομοιωτή *or1ksim* και το δεύτερο χρησιμοποιεί τον *NC-Verilog* εξομοιωτή που κάνει εξομοίωση με βάση το υλικό (hardware based simulation). Στο πρώτο περιβάλλον γίνεται η επαλήθευση της λειτουργικότητας των benchmarks και στο δεύτερο προσομοιώνεται η ουσιαστική λειτουργία του OR1200 σε επίπεδο υλικού με βάση το benchmark που εκτελέσαμε.

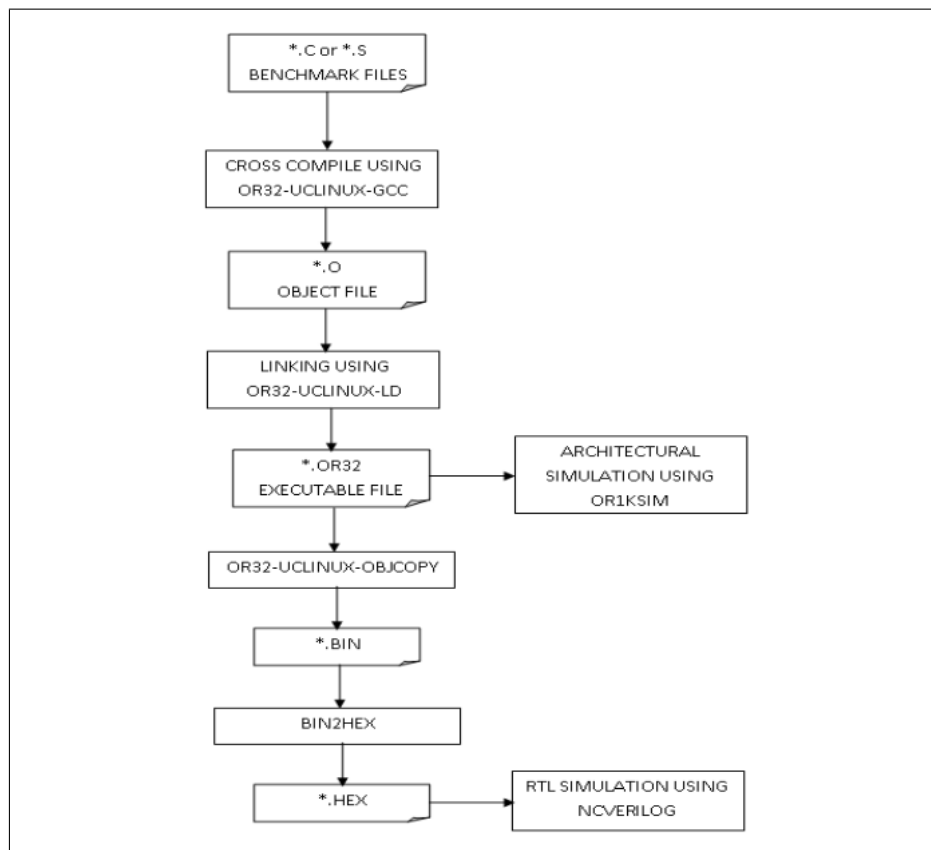
Η συνολική ροή της εξομοίωσης παρουσιάζεται στο Σχήμα 4. Τα benchmarks είναι είτε .C αρχεία είτε .S αρχεία. Αυτά τα αρχεία, αρχικά γίνονται cross-compiled χρησιμοποιώντας την εντολή `or32-uclinux-gcc` και παράγουν ένα .O object αρχείο. Το object αρχείο μετά μετατρέπεται σε ένα .OR32 εκτελέσιμο αρχείο χρησιμοποιώντας την συνδυαστική εντολή `or32-uclinux-ld`. Αυτό το εκτελέσιμο αρχείο χρησιμοποιείται από τον `or1k` αρχιτεκτονικό εξομοιωτή. Περαιτέρω το αρχείο .OR32 μετατρέπεται σε δυαδικό (binary) αρχείο χρησιμοποιώντας την εντολή `or32-uclinux-objcopy`. Στο τέλος δημιουργείται ένα .HEX αρχείο χρησιμοποιώντας τον `binary to hex` μετατροπέα `bin2hex`. Το παραγόμενο αρχείο .HEX φορτώνεται στην flash μνήμη του RTL κώδικα του OR1200 επεξεργαστή και μετά γίνεται η εξομοίωση με βάση το υλικό.

3.2 Package Structure

Σε αυτό το μέρος θα παρουσιάσουμε την δομή και τα περιεχόμενα του φακέλου που αποτελεί τον επεξεργαστή OR1200.

Το συγκεκριμένο πακέτο περιέχει:

- **Rtl** : Εδώ βρίσκεται ο κώδικας verilog που περιγράφει σε υλικό τον επεξεργαστή OR 1200.
- **Boards** : Περιέχει κατάλληλα αρχεία για να περάσεις τον OR1200 σε συγκεκριμένες πλακέτες FPGA.
- **Sim** : Σε αυτό τον φάκελο δημιουργούνται τα αποτελέσματα της εξομοίωσης σύμφωνα με το Makefile που δημιουργήσαμε για τους σκοπούς του εργαστηρίου.
 1. *sim/bin* : Εδώ βρίσκεται το Makefile που δημιουργήσαμε για τους σκοπούς του εργαστηρίου και χειρίζεται την λειτουργία του επεξεργαστή.
 2. *sim/run* : Εδώ εκτελούνται όλες οι εντολές που σχετίζονται με την εξομοίωση του επεξεργαστή.
 3. *sim/out* : Εδώ τοποθετούνται όλα τα αρχεία που παράγονται μετά την εξομοίωση. Να σημειωθεί ότι τα waveforms τοποθετούνται στο *sim/run*.
- **Sw** : Εδώ βρίσκονται τα βασικότερα αρχεία που είναι απαραίτητα για το cross-compilation και την σωστή λειτουργία του OpenRISC επεξεργαστή.
 1. *sw/drivers* : Εδώ βρίσκονται οι drivers και τα εργαλεία για τροποποίηση του hardware.



Σχήμα 3.1.1: Συνολική ροή εξομοίωσης.

2. *sw/lib* :Εδώ βρίσκεται μια απλή βιβλιοθήκη που σε συνδυασμο με τους drivers κατά την διάρκεια του compile δημιουργούν την βιβλιοθήκη liborpsoc που τοποθετείται στο *sw/lib*.
 3. *sw/lib/include* :Εδώ βρίσκεται το αρχείο *cpu-utils.h* που περιέχει όλες τις συναρτήσεις σχετικές με την CPU του OpenRISC.
 4. *sw/tests* :Εδώ βρίσκεται το λογισμικό που χρησιμοποιείται (.C και .S αρχεία) για να δοκιμαστεί η σωστή λειτουργία του επεξεργαστή (testing) σε υπομονάδες όπως *ethmac*, *or1200*, *sdram*, *spi* και *uart*. Στον φάκελο κάθε υπομονάδας (πχ *sw/test/sdram*) υπάρχουν δύο υποφάκελοι *board* και *sim* (πχ *sw/test/sdram/board* και *sw/test/sdram/sim*). Στον *sim* φάκελο υπάρχουν τα tests που εκτελούνται κατά την εξομοίωση του επεξεργαστή σε ένα PC και στον φάκελο *board* υπάρχουν τα tests που εκτελούνται κατά την λειτουργία του επεξεργαστή.
- **Doc** : Εδώ βρίσκεται documentation που χρειαζόμαστε για να καταλάβουμε την φύση των testbenches και οι οδηγίες για το simulation σύμφωνα με το Makefile που δημιουργήσαμε για τους σκοπούς του εργαστηρίου.

3.3 Simulation commands

3.3.1 Η βασική διαδικασία

Η διαδικασία με την οποία γίνεται η εξομοίωση του OR1200 είναι η εξής:

1. Το Makefile που ελέγχει το simulation βρίσκεται στο `/sim/bin/` και είναι προσαρμοσμένο και από το `/sim/run/`.
2. Στον φάκελο `/sim/run/` εκτελώντας την εντολή `make rtl-tests` κάνει compile τον rtl (verilog) κωδικά του OR1200 και εκτελεί όλα τα testbenches (assembly) που βρίσκονται στο `sw/tests/or1200`.
3. Τα αποτελέσματα του παραπάνω βήματος τοποθετούνται στο `/sim/out/`. Αυτά είναι (στα αγγλικά για καλύτερη κατανόηση):
 - `test-name-executed.log` : A trace of the processor after each executed instruction
 - `test-name-sprs.log` : A list of processor special purpose registers (SPR) accesses is created
 - `test-name-lookup.log` : A list of when each instruction was executed is generated
 - `test-name-general.log` : The use of the processor's report mechanism is commonplace in the test software, as it allows for the checking of intermediate values after simulation.

3.3.2 Εκτέλεση ενός συγκεκριμένου test

Η εξομοίωση ενός συγκεκριμένου test γίνεται με την εντολή `make rtl-test TEST=test-name`¹. Πρέπει το αρχείο `test-name.c` (ή `test-name.s`) να είναι τοποθετημένο στο `sw/tests/module/sim/` όπου `module` είναι η υπομονάδα που θέλουμε να ελέξουμε με κάποιο από τα tests που μας παρέχει (πχ `sw/tests/sdram/sim/sdram-rows.c`).

3.3.3 Εκτέλεση συγκεκριμένων tests μαζί

Η εξομοίωση πολλών συγκεκριμένων tests γίνεται με την εντολή `make rtl-test TEST="test-name1 test-name2 ..."` (πχ `make rtl-tests TESTS="sdram-rows uart-simple or1200-mmio or1200-fp"`)

3.3.4 Παρέχοντας μια προσαρμοσμένη VMEM εικόνα (image)

Είναι δυνατό να καθορίσουμε το μονοπάτι μιας ήδη υπάρχουσας VMEM εικόνας που θα την χρησιμοποιήσουμε αντί να κάνουμε πάλι από την αρχή test το software. Χρησιμοποιώντας την μεταβλητή `USER_VMEM` μπορούμε να καθορίσουμε το μονοπάτι της VMEM εικόνας που θέλουμε να τρέξουμε. Για παράδειγμα `make rtl-test USER_VMEM=/path/to/myapp.vmem`. Αυτή η εικόνα θα αντιγραφεί στο φάκελο στον οποίο εργαζόμαστε και θα μετονομαστεί σύμφωνα με το τι η μνήμη στην εξομοίωση απαιτεί.

¹ test-name: Είναι η ονομασία του test που θέλουμε να εκτελέσουμε.

3.3.5 Παρέχοντας ένα "precompiled" εκτελέσιμο .ELF αρχείο

Είναι δυνατό να καθορίσουμε το μονοπάτι ενός OR32 ELF εκτελέσιμου αρχείου που θα το χρησιμοποιήσουμε αντί να κάνουμε πάλι από την αρχή test το software. Χρησιμοποιώντας την μεταβλητή *USER_ELF* μπορούμε να καθορίσουμε το μονοπάτι στο οποίο βρίσκεται αυτό το αρχείο. Για παράδειγμα *make rtl-test USER_ELF=/path/to/myapp.elf*. Το ELF αρχείο θα μετατραπεί σε δυαδική μορφή και μετά σε VMEM και θα φορτωθεί στο μοντέλο για να εκτελεστεί.

3.3.6 Κυματομορφές

Παράλληλα με το simulation ενός ή περισσότερων testbench παράγονται και οι κυματομορφές των εξομοιώσεων που μας βοηθάνε στην καλύτερη κατανόηση τους. Τα αρχεία που παράγονται βρίσκονται σε φακέλους μέσα στο */sim/run/* που έχουν την ονομασία *test-name.shm*. Για να εμφανίσεις τις κυματομορφές αυτές πρέπει μέσω κονσόλας να οδηγηθείς στο */sim/run/* και μετά να εκτελέσεις την εντολή *simvision test-name.shm*.

3.3.7 Επιπρόσθετα επιλογές στις εντολές

Παρακάτω παρουσιάζονται μερικές μεταβλητές που μας βοηθούν να επιλέξουμε καποιές συγκεκριμένες λειτουργίες.

- *END_TIME* : Αναγκάζει την εξομοίωση να τερματίσει (\$finish). Πχ *make rtl-test TEST="or1200-mul" END_TIME=100* όπου είναι ίδιο με το *#100 \$finish* σε αρχείο verilog.
- *DISABLE_PROCESSOR_LOGS* : Απενεργοποιεί την οθόνη παρακολούθησης του επεξεργαστή που συλλέγει πληροφορίες κατά την εκτέλεση μιας εξομοίωσης. Αυτό βοηθάει στην επιτάχυνση της εξομοίωσης αφού απαιτείται λιγότερος χρόνος στην εγγραφή αρχείων και αποτρέπει την δημιουργία πολύ μεγάλων αρχείων σε χρονοβόρες εξομοιώσεις.
- *SIMULATOR* : Επιλέγουμε τον εξομοιωτή υλικού που θέλουμε να χρησιμοποιήσουμε. Προκαθορισμένος εξομοιωτής είναι ο NC-Verilog. Άλλη επιλογή είναι το ICARUS.