ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Διπλωματική Εργασία

Μια ενδεικτική Υλοποίηση RISC-V επεξεργαστή και ενός υποστηρικτικού Assembler

Βάιος Λασκαρέλιας
A.M.: 1054432

Επιβλέπων
Χαρίδημος Βέργος, Καθηγητής

Μέλη Εξεταστικής Επιτροπής
Δημήτριος Νικολός, Καθηγητής
Νικόλαος Σκλάβος, Αναπληρωτής Καθηγητής

Πάτρα, Οκτώβριος 2021

Η παρούσα διπλωματική εργασία υλοποιεί:

- Ένα λειτουργικό μοντέλο RISC-V RV32I επεξεργαστή σε επίπεδο εξομοίωσης

- Έναν συνοδευτικό RISC-V RV32I Assembler συμβατό με κάθε RV32I επεξεργαστή

Αρχιτεκτονική συνόλου εντολών
Instruction Set Architecture (ISA)

- Ένα σύνολο καλά ορισμένων εντολών
- Ορίζονται οι δυαδικές τους κωδικοποιήσεις
- Κάθε εντολή έχει ορισμένη συνέπεια εκτέλεσης
- Οι εντολές “γεφυρώνουν” το υλικό με το λογισμικό
- Οι συμβατοί επεξεργαστές εκτελούν όλες τις εντολές του συνόλου με τις ορισμένες συνέπειες

Πηγή: [HP11]

Είδη Αρχιτεκτονικών Συνόλων Εντολών

- RISC (Reduced Instruction Set Computer)
  - Λίγες εντολές
  - Εκτελούν γενικές, απλές λειτουργίες
  - Ξεχωριστές προσπελάσεις μνήμης και επεξεργασία δεδομένων
- CISC (Complex Instruction Set Computing)
  - Πολλές εντολές
  - Εκτελούν εξειδικευμένες, περίπλοκες λειτουργίες
  - Ταυτόχρονη προσπέλαση μνήμης και επεξεργασία δεδομένων

Πηγή: [HP11]

RISC-V ISA

- Αρχιτεκτονική συνόλου εντολών τύπου RISC
- Σκοποί
  - Ακαδημαϊκοί
  - Εκπαιδευτικοί
- Χαρακτηριστικά:
  - Απλότητα
  - Παραμετροποίηση
  - Επεκτασιμότητα
  - Αποφυγή σχεδιαστικών περιορισμών των υλοποιήσεων

Πηγή: [Wat+16]

Ακαδημαικές Υλοποιήσεις

- **BOOM,** University of California, Berkeley
  - RV64GC
  - Synthesizable, Out-of-order εκτέλεση
  - Υποστήριξη Linux
- **Riscy Processors – RiscyOO,** MIT
  - Πολλά υποσύνολα RISC-V, οικογένεια επεξεργαστών
  - Out-of-order εκτέλεση, Βασισμένα σε προηγούμενες MIPS σχεδιάσεις
  - Υποστήριξη Linux
- **Lizard,** Cornell University
  - RV64IM, Παραμετροποιήσιμος
  - Synthesizable, Out-of-order εκτέλεση, Register Renaming
  - Υποστήριξη για static C προγράμματα

Πηγή: [Int, CPA16, CSAa, CSAb, JG]

Εμπορικές Υλοποιήσεις

- **SiFive P550,** SiFive, Inc.
  - RV64GC
  - Out-of-order εκτέλεση, 13-stage pipeline, triple instruction issuing
  - Υποστήριξη Linux, ο γρηγορότερος επεξεργαστής γενικής χρήσης
- **SweRV Core EH1,** Western Digital
  - RV32IMC
  - 9-stage pipeline, dual instruction issuing
  - Ελεγκτής αποθηκευτικών μέσων
- **NEOX Graphics,** Think Silicon
  - RV64GC
  - Multithreading, 4-64 core, παραμετροποίηση cache και οργάνωση πυρήνων
  - Επεξεργαστής γραφικών

Πηγή: [SiFb, Dig, Sil]

Επεκτάσεις RISC-V

- RV32I – RV64I – RV128I
  - Πράξεις ακέραιων
  - Αναγνώσεις / εγγραφές μνήμης
  - Εντολές διακλάδωσης
- Επέκταση M
  - Πολλαπλασιασμοί – Διαιρέσεις Ακεραίων
- Επέκταση F, D, Q
  - Αριθμητική κινητής υποδιαστολής μονής, διπλής, τετραπλής ακρίβειας

Πηγή: [Wat+16]

Επεκτάσεις RISC-V (συνέχεια)

- Επέκταση Ziscr
  - Καταχωρητές ελέγχου και κατάστασης
- Επέκταση Zifencei
  - Ατομικές προσπελάσεις μνήμης εντολών
- Επέκταση G
  - Συνδυασμός των προηγούμενων
  - Χρήση σε επεξεργαστές γενικού σκοπού

Πηγή: [Wat+16]

RV32I - Ανάλυση

- Εντολές Αριθμητικών – Λογικών Πράξεων
  - Πράξεις μεταξύ Καταχωρητή – Καταχωρητή
  - Πράξεις μεταξύ Καταχωρητή – Άμεσου δεδομένου
- Εντολές Προσπέλασης Μνήμης
  - Ανάγνωση Μνήμης
  - Εγγραφή Μνήμης
- Εντολές Φόρτωσης Ειδικού Δεδομένου
- Εντολές Διακλάδωσης
  - Υπό Συνθήκη
  - Άλματος
- Εντολές Διαχείρισης
  - Διαχείριση Κλήσεων Λογισμικού
  - Διαχείριση Προσπελάσεων Μνήμης

Εντολές Αριθμητικών – Λογικών Πράξεων

- Πράξεις Καταχωρητή – Καταχωρητή
  - Εντολές `add, sub, sll, slt, sltu, xor, srl, sra, or, and`
  - Ανάγνωση 2 καταχωρητών
  - Εκτέλεση πράξης
  - Αποθήκευση σε τρίτο καταχωρητή

- Πράξεις Καταχωρητή – Άμεσου Δεδομένου
  - Εντολές `addi, subi, slli, xori, ...`
  - Ανάγνωση ενός καταχωρητή
  - Αποκωδικοποίηση άμεσου δεδομένου
  - Εκτέλεση πράξης
  - Αποθήκευση σε τρίτο καταχωρητή

Πηγή: [Wat+16]

Εντολές Προσπέλασης Μνήμης

- Εγγραφή Δεδομένου στην Μνήμη Δεδομένων
  - Εντολές `sb, sh, sw`
  - Ανάγνωση καταχωρητή με το δεδομένο προς εγγραφή
  - Ανάγνωση καταχωρητή και άμεσου δεδομένου για διευθυνσιοδότηση
  - Ορισμός μεγέθους εγγραφής
  - Εγγραφή δεδομένου στην υπολογισμένη διεύθυνση

- Ανάγνωση Δεδομένου από την Μνήμη Δεδομένων
  - Εντολές `lb, lh, lw, lbu, lhu`
  - Ανάγνωση καταχωρητή και άμεσου δεδομένου για διευθυνσιοδότηση
  - Ορισμός μεγέθους ανάγνωσης και επέκτασης προσήμου
  - Αποθήκευση δεδομένου ανάγνωσης σε καταχωρητή

Πηγή: [Wat+16]

Εντολές Φόρτωσης Ειδικού Δεδομένου

- Εντολή `lui`
  - Load Upper Immediate
  - Αποκωδικοποίηση Άμεσου Δεδομένου
  - Εγγραφή στα πιο σημαντικά ψηφία του καταχωρητή

- Εντολή `auipc`
  - `Add Upper Immediate to Program Counter`
  - Αποκωδικοποίηση άμεσου δεδομένου
  - Πρόσθεση μετρητή προγράμματος και άμεσου δεδομένου
  - Εγγραφή αποτελέσματος σε καταχωρητή

Πηγή: [Wat+16]

Εντολές Διακλάδωσης

- Εντολές Διακλάδωση Υπό Συνθήκη
  - Εντολές `beq, bne, blt, bge, bltu, bgeu`
  - Ανάγνωση 2 καταχωρητών με το δεδομένο προς εγγραφή
  - Το άμεσο δεδομένο κωδικοποιεί την σχετική μετατόπιση
  - Εκτέλεση σύγκρισης για απόφαση διακλάδωσης
  - Πρόσθεση της μετατόπισης στον Μετρητή Προγράμματος εφόσον αποφασιστεί διακλάδωση

- Εντολές Άλματος
  - Εντολές `jal, jalr`
  - Αποθήκευση της διεύθυνσης της επόμενης εντολής σε καταχωρητή
  - Πρόσθεση της μετατόπισης στον Μετρητή Προγράμματος ανεξαρτήτως συνθήκης

Πηγή: [Wat+16]

Εντολές Διαχείρισης

- Εντολές Διαχείρισης Κλήσεων Λογισμικού
  - Εντολές `ecall, ebreak`
  - Κλήση ειδικής ρουτίνας O.S. ή Debugger

- Εντολή Διαχείρισης Προσπελάσεων Μνήμης
  - Εντολή `fence`
  - Barrier μνήμης για πολυπύρηνες υλοποιήσεις

Πηγή: [Wat+16]

Δυαδικές Κωδικοποιήσεις

- Σταθερή θέση των πεδίων κωδικοποίησης
- 6 κατηγορίες – R, I, S, B, U, J

31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R - type
imm[11:0]		rs1	funct3	rd	opcode	I - type
imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	S - type
imm[12|10:5]	rs2	rs1	funct3	imm[4:1|11]	opcode	B - type
imm[31:12]				rd	opcode	U - type
imm[20|10:1|11|19:12]				rd	opcode	J - type

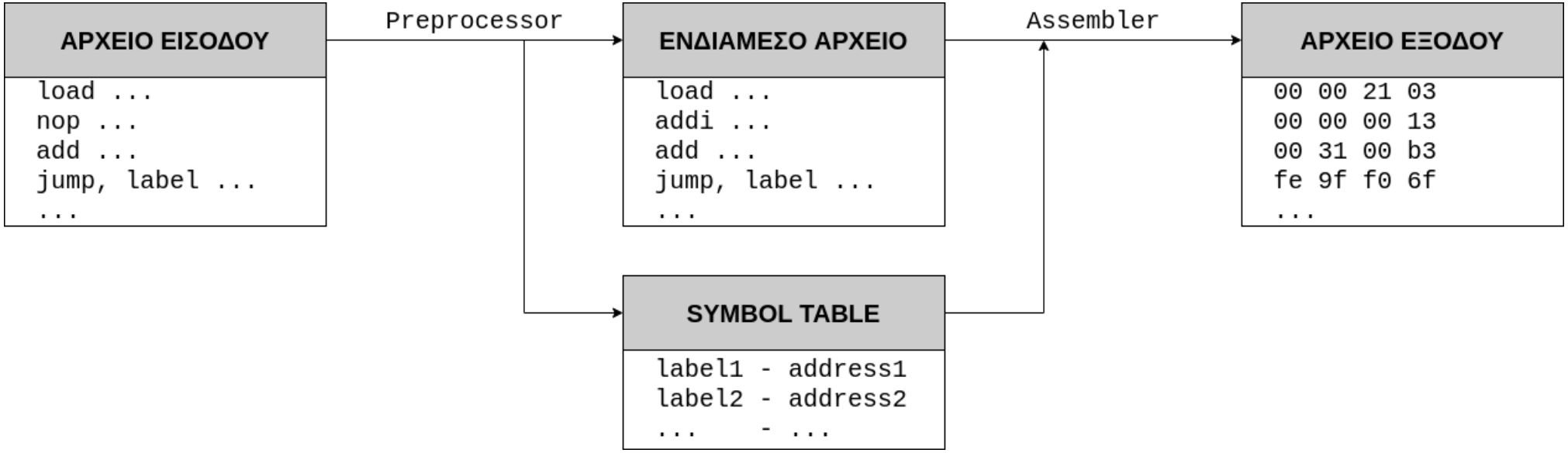
Πίνακας από [Wat+16]

Πηγή: [Wat+16]

Σχεδίαση Assembler

- Μετατροπή κώδικα Assembly σε εκτελέσιμου
- Βασισμένο στις δυαδικές κωδικοποιήσεις
- Preprocessor + Assembler

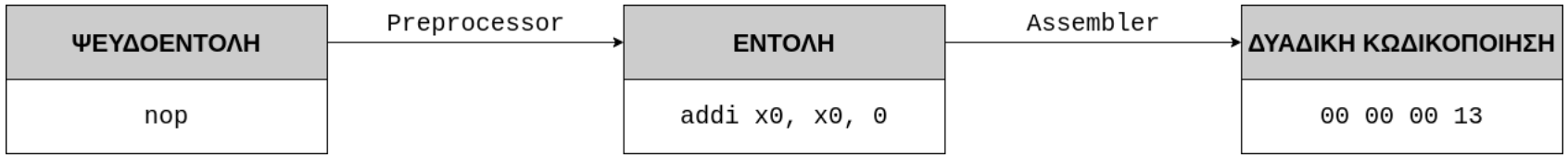
Διαδικασία Μετάφρασης:



Preprocessor + Assembler

- Preprocessor
  - Συντακτικός έλεγχος μορφών εντολών
  - Καταγραφή θέσεων μνήμης των Label
  - Παραγωγή Symbol Table
  - Μετατροπή ψευδοεντολών σε εντολές
  - Παραγωγή ενδιάμεσου αρχείου
- Assembler
  - Έλεγχος πεδίων εντολών
  - Υπολογισμός μετατοπίσεων με βάση το Symbol Table
  - Ανάγνωση ενδιάμεσου αρχείου
  - Μετατροπή εντολών σε δυαδικές κωδικοποιήσεις (Bytecode)

Μετατροπή Ψευδοεντολής



Symbol Table

Όνομα Label	Θέση μνήμης
__start	0x8000
overflow_handler	0x8040
stack_pop	0x8100
...	...

Μορφές εντολών

- Συντακτικό
  - Instruction Register, Immediate(Register)
  - Instruction Register, Register, Register
  - Instruction Register, Register, Immediate
  - Intruction Register, Register, Label
  - Instruction Register, Immediate
  - Instruction Register, Label
- Παραδείγματα
  - `sw s0, 0(s1)`
  - `add s0, ra, sp`
  - `add s1, sp, 1`
  - `beq s0, zero, overflow_handler`
  - `lui t1, 0xff`
  - `jal ra, overflow_handler`

Παραγωγή Διανυσμάτων Εντολών

- Χρήση unsigned int – 32 bit
- Αξιοποίηση ολισθήσεων και προσθέσεων
- Τα λιγότερο σημαντικά ψηφία παραμένουν αμετάβλητα

Παράδειγμα: add x3, x1, x2

xxxxxxx	xxxxx	xxxxx	xxx	xxxxx	xxxxxxx	Αρχικά
xxxxxxx	xxxxx	xxxxx	xxx	xxxxx	**0110011**	1 - opcode
xxxxxxx	xxxxx	xxxxx	xxx	**00011**	0110011	2 - rd
xxxxxxx	xxxxx	xxxxx	**000**	00011	0110011	3 - funct3
xxxxxxx	xxxxx	**00001**	000	00011	0110011	4 - rs1
xxxxxxx	**00010**	00001	000	00011	0110011	5 - rs2
0000000	00010	00001	000	00011	0110011	6 - funct7

Σχεδίαση Επεξεργαστή

- Αξιοποίηση Διασωλήνωσης
- Ανεξάρτητα στάδια εκτέλεσης
- Κλασσική διασωλήνωση RISC 5 σταδίων
  - Ανάκτηση Εντολής (IF)
  - Αποκωδικοποίηση εντολής (ID)
  - Εκτέλεση (EX)
  - Προσπέλαση Μνήμης (MEM)
  - Επανεγγραφή (WB)

- Όλες οι μονάδες απαιτούν 1 κύκλο ρολογιού
- Θετικά ακμοπυροδότητες, σύγχρονες λειτουργίες
- Συγχώνευση σταδίων MEM + WB
- Προσθήκη σταδίου επίλυσης κινδύνων (HAZ)

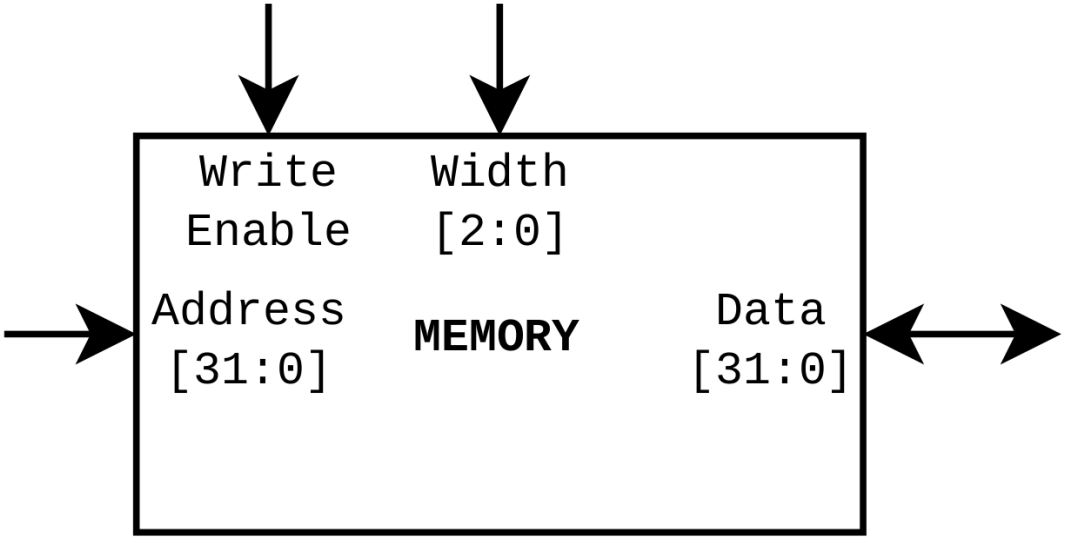
Πηγή: [Nik17]

Μονάδα Μνήμης

- Αποθήκευση Εντολών ή Δεδομένων
- Αφορά τις εντολές Εγγραφής και Ανάγνωσης

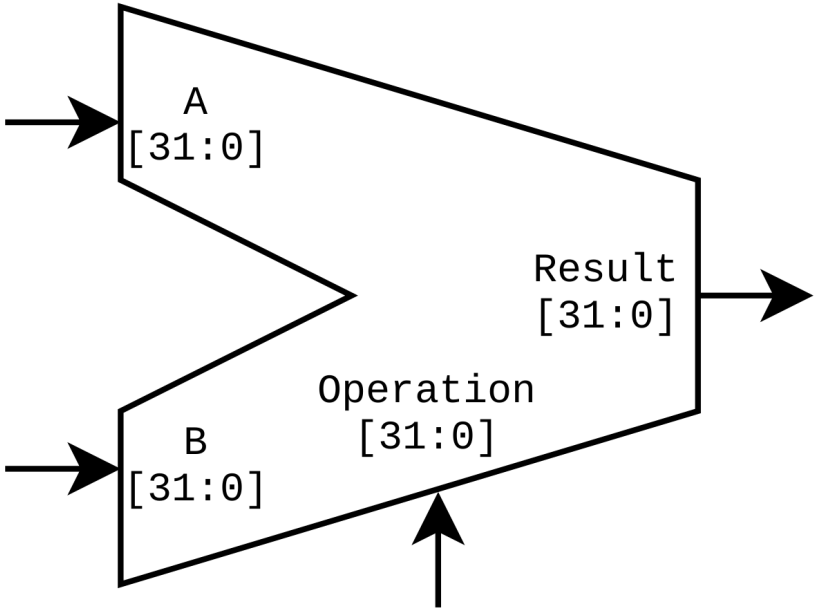
Ενσωματωμένη επέκταση προσήμου για αναγνώσεις

- Μία θύρα δεδομένων



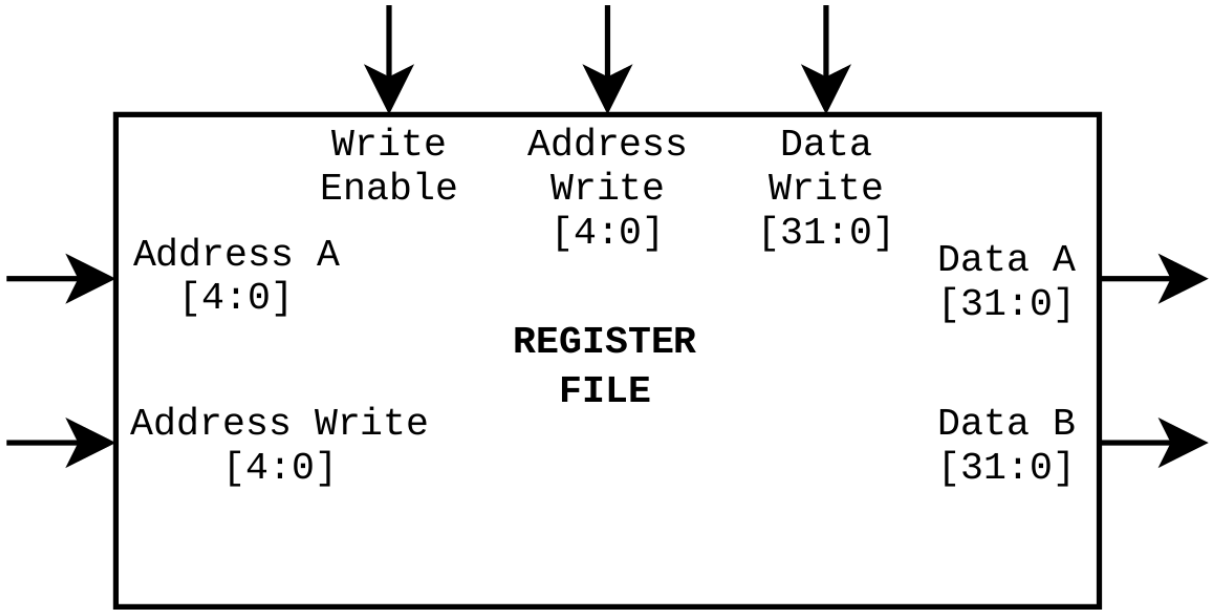
Αριθμητική – Λογική Μονάδα

- Εκτέλεση Αριθμητικών - Λογικών Πράξεων
- Μία είσοδος επιλογής πράξης
- Δύο είσοδοι δεδομένων
- Πολυπλεξία εισόδων για εκτέλεση πολλών τύπων εντολών



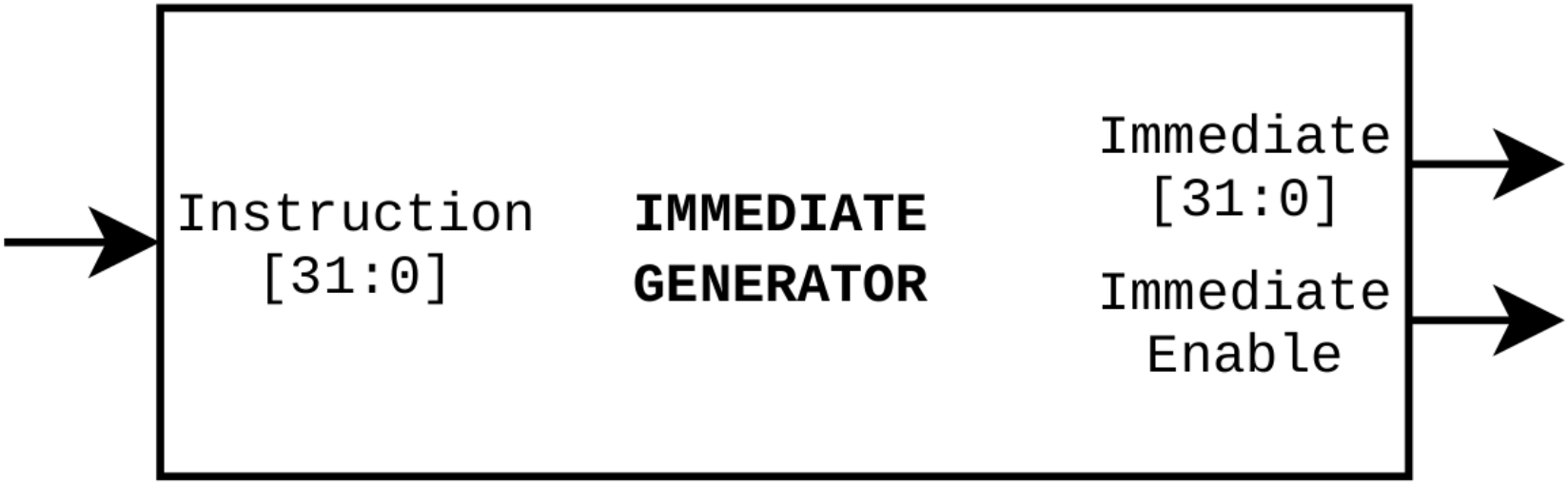
Αρχείο Καταχωρητών

- 2 καταχωρητές προς ανάγνωση
- 1 καταχωρητής προς εγγραφή
- Ταυτόχρονη ανάγνωση και εγγραφή καταχωρητή
- Λειτουργίες στον ίδιο κύκλο



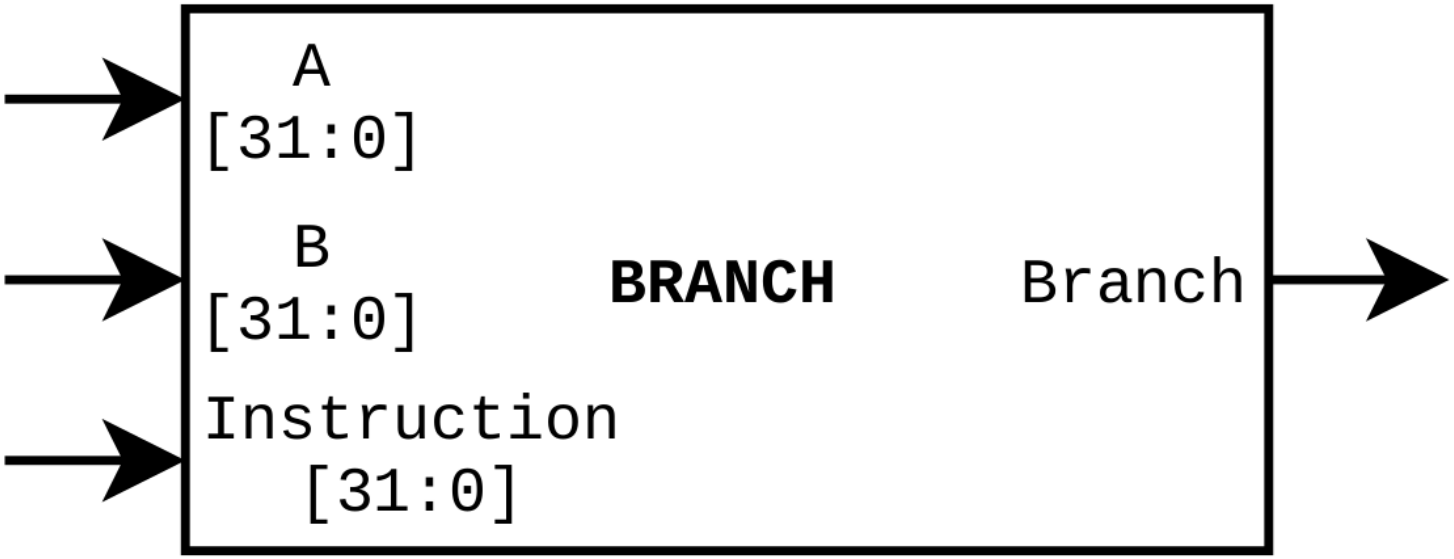
Μονάδα Διαχείρισης Άμεσων Δεδομένων

- Αποκωδικοποίηση άμεσου δεδομένου
- Είσοδος εντολής με άμεσο δεδομένο
- Σήμα ενεργοποίησης για πολυπλεξία εισόδων αριθμητικής – λογικής μονάδας



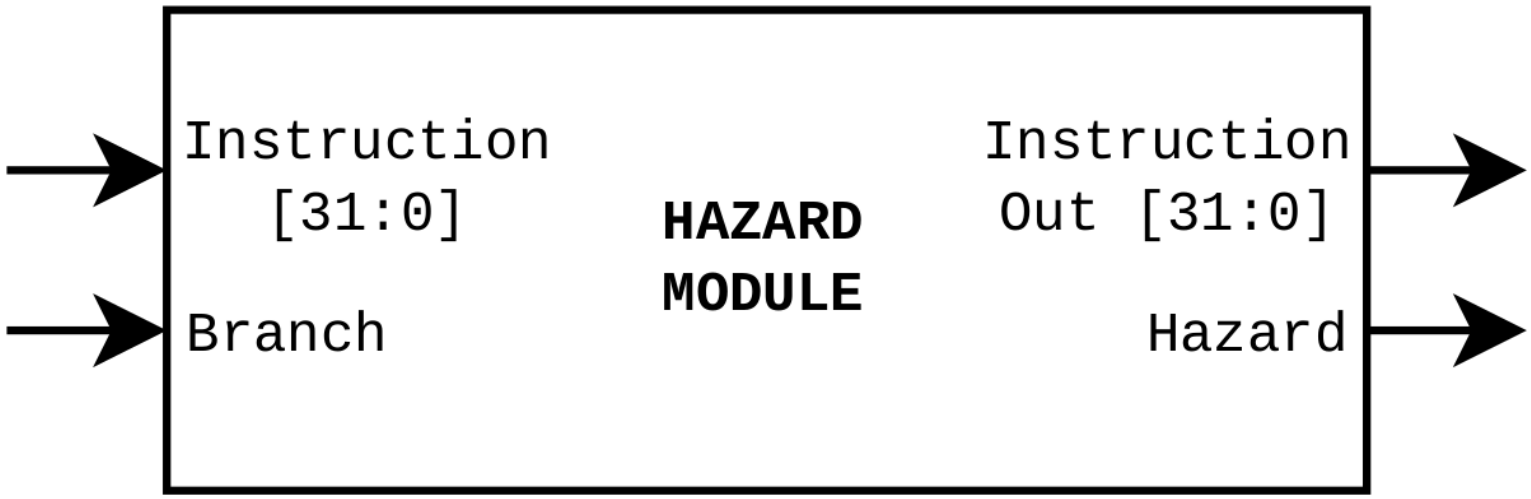
Μονάδα Διαχείρισης Διακλαδώσεων

- Εκτέλεση συγκρίσεων
- Μία είσοδος εντολής σύγκρισης
- 2 είσοδοι δεδομένων
- Σήμα ενεργοποίησης διακλάδωσης

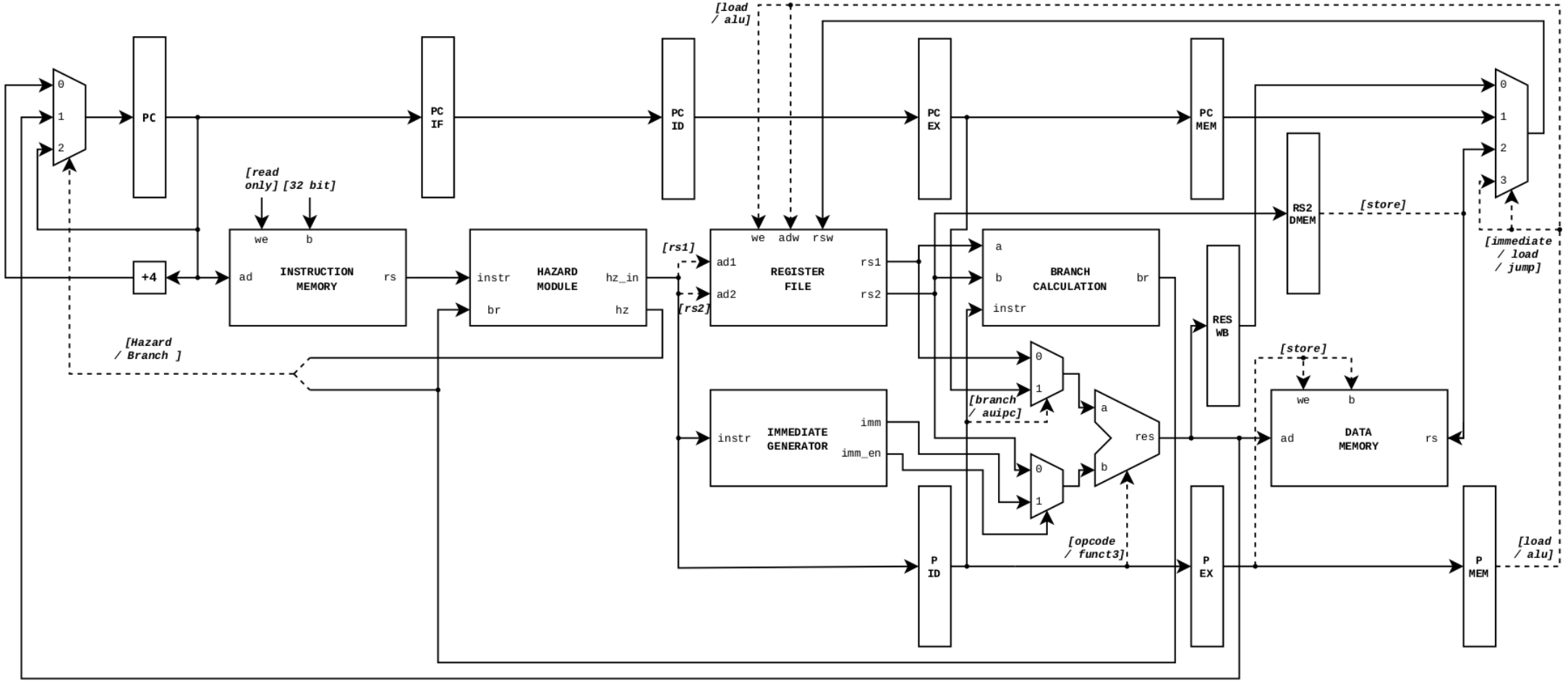


Μονάδα Διαχείρισης Κινδύνων

- Έλεγχος Ροής Εντολών
- Αντιμετώπιση ανάγνωσης καταχωρητή προς εγγραφή
- Αντιμετώπιση κινδύνων δεδομένων
- Σήμα ύπαρξης κινδύνου για παύση της εκτέλεσης



Μικροαρχιτεκτονική Επεξεργαστή



Ενδεικτική Σύνθεση

- Συνθέσιμο σε FPGA
- Αναμενόμενη χρήση λογικών μπλοκ
- 148 MHz clock
- Υλοποιήσιμο σε υλικό εφόσον γίνουν περαιτέρω έλεγχοι ορθής λειτουργίας

Εκτέλεση Προγράμματος

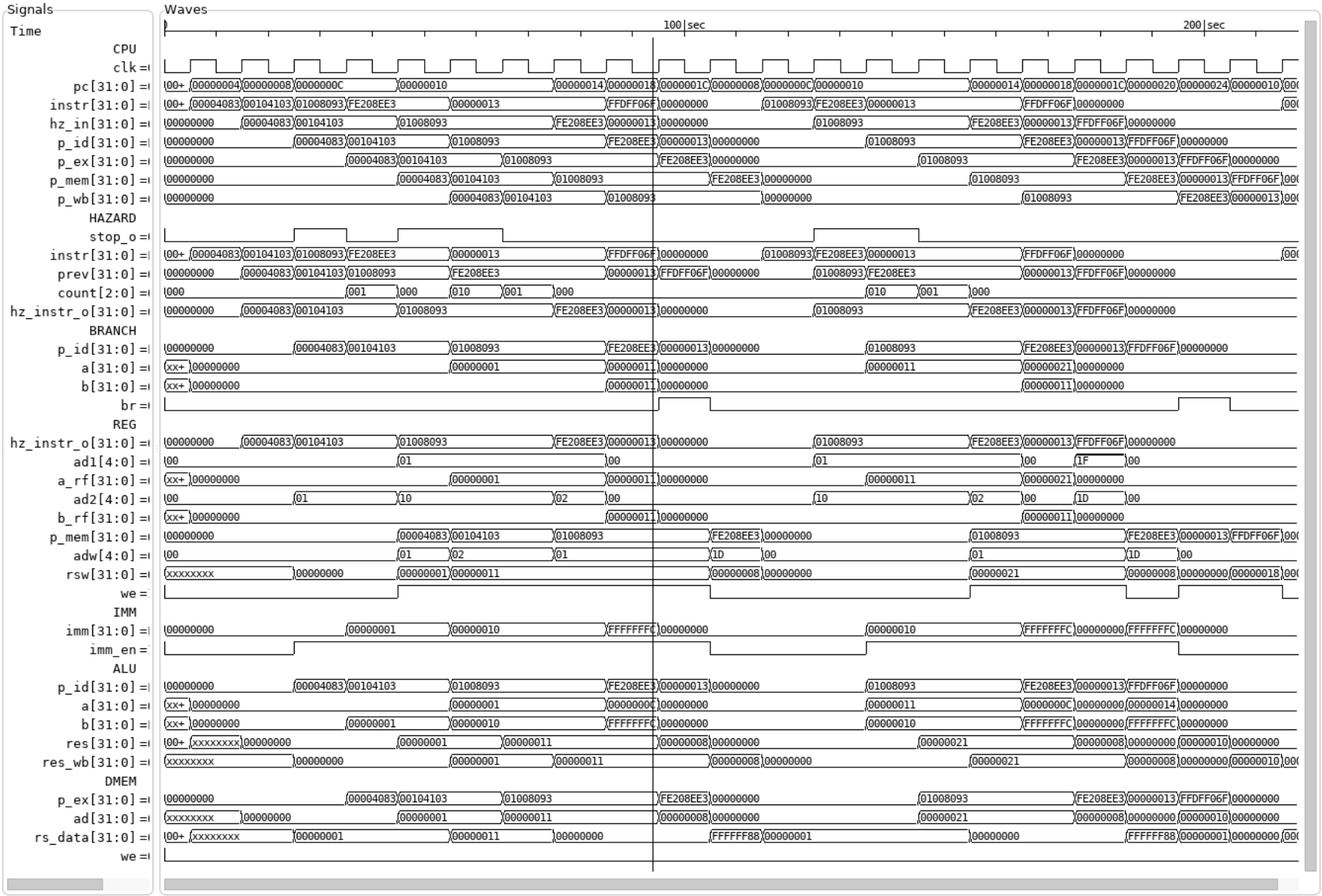
- Παράδειγμα χρήσης ως ολοκληρωμένου εξομοιωτή RV32I
- Αντιμετώπιση όλων των τύπων κινδύνων

Μετάφραση προγράμματος προς εκτέλεση σε δυαδική κωδικοποίηση

Πρόγραμμα σε Assembly	Περιεχόμενο Μνήμης Εντολών*
1 lbu x1, 0(zero) 2 lbu x2, 1(zero) 3 loop: 4 addi x1, x1, 0x010 5 beq x1, x2, loop 6 halt: 7 nop 8 jal x0, halt	1 @0 2 83 40 00 00 3 03 41 10 00 4 93 80 00 01 5 e3 8e 20 fe 6 13 00 00 00 7 6f f0 df ff 8 // SYMBOL TABLE 9 // [0x00000008] loop 10 // [0x00000010] halt

Εκτέλεση Προγράμματος (συνέχεια)

Κυματομορφή εκτέλεσης προγράμματος



Μελλοντικές Βελτιστοποιήσεις

- Επεκτάσεις RISC-V
- Instruction Window – out-of-order
- Register Renaming
- Branch Prediction
- Υλοποίηση σε υλικό
- Υλοποίηση Compiler
- Υποστήριξη συσκευών

Ευχαριστώ για την προσοχή σας

Πηγές - Βιβλιογραφία

- [HP11] John L Hennessy and David A Patterson. Computer architecture: a quantitative approach. Elsevier, 2011.
- [For14] Behrouz Forouzan. Foundations of Computer Science. Andover: Cengage Learning, 2014. isbn: 9781408044117.
- [CPA16] Christopher Celio, David Patterson, and Krste Asanovic. “The Berkeley Out- of-Order Machine (BOOM) Design Specification”. In: University of California, Berkeley (2016).
- [Wat+16] Andrew Waterman et al. “The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.1”. In: (2016).
- [Nik17] Dimitrios Nikolos. Computer Architecture. Pan. Papakonstantinou, 2017. isbn: 978-618-83197-0-7.
- [CSAa] MIT CSAIL. Riscy Processors - Open-Sourced RISC-V Processors. url: https://github.com/csail-csg/riscy. (accessed: 11/10/2021).
- [CSAb] MIT CSAIL. RiscyOO: RISC-V Out-of-Order Processors. url: https://github.com/csail-csg/riscy-OOO. (accessed: 11/10/2021).
- [Dig] Western Digital. RISC-V — Western Digital. url: https://www.westerndigital.com/company/innovations/risc-v. (accessed: 11/10/2021).
- [Int] RISC-V International. Members - RISC-V International. url: https://riscv.org/members/. (accessed: 11/10/2021).
- [JG] Aaron Wisner Jacob Glueck. The Lizard Core. url: https://github.com/cornell-brg/lizard. (accessed: 11/10/2021).
- [kva] kvakil.me. venus. url: https://www.kvakil.me/venus/. (accessed: 11/10/2021).
- [SiFa] SiFive. RISC-V Core IP - SiFive. url: https://www.sifive.com/risc-v-core-ip. (accessed: 11/10/2021). [SiFb] SiFive. SiFive Performance P550 Core Sets New Standard as Highest Performance RISC-V Processor IP - SiFive. url: https://www.sifive.com/risc- v-core-ip. (accessed: 11/10/2021).
- [Sil] Think Silicon. NEOX GPUs - Think Silicon. url: https://www.think-silicon.com/neox-graphics. (accessed: 11/10/2021).