

УНИВЕРЗИТЕТ У БЕОГРАДУ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ



**ИМПЛЕМЕНТАЦИЈА RISC-V ПРОЦЕСОРА СА
ПОДРШКОМ ЗА ЕКСТЕРНИ УВИД У СТАЊЕ
ПРОЦЕСОРА**

Дипломски рад

Ментор:

проф. др Захарије Радивојевић,
ванредни професор

Кандидат:

Лазар Премовић
2019/0091

Београд, Септембар 2023.

Садржај

1	Увод	1
2	Преглед спецификација и коришћених технологија	3
2.1	<i>RISC-V</i> инструкцијски сет	3
2.2	Подршка за екстерно дебаговање	4
2.3	Поређење са <i>Si Five Freedom E310-G002</i>	4
2.4	Преглед коришћених технологија и алата	6
	Литература	8
	Списак скраћеница	10
	Списак слика	11
	Списак табела	12

1 Увод

Рачунари су од свог настанка па до данашњег дана нашли примену у најразличитијим областима живота. Свака од тих примена придаје различиту важност одређеним карактеристикама рачунара (нпр. перформансе, величина, цена), што је довело до настанка различитих класа рачунара. Прелазак тржишта са мејнфрејм и персоналних рачунара на данас популарне кластере великих размера и преносиве уређаје, као и експлозија јефтиних уређаја повезаних на мрежу (тзв. интернет ствари), од дизајнера хардвера захтева да преиспитају одлуке и метрике којима су се до сада водили. Ове нове класе рачунара, уместо на сирове перформансе, акценат стављају на енергетску ефикасност, тј. однос утрошене енергије и количине обрађених података. Ова промена захтева представља одличну шансу за примену алтернативних архитектура рачунара. Једну од таквих архитектура представљају *Reduced Instruction Set Computer (RISC)* процесори. Иако *RISC* архитектуре нису нова појава (први *RISC* процесори су направљени 1970-их), тренутно тржиште је веома погодно за развој оваквих процесора.

Други фактор који има значајан утицај на развој модерних процесора је смањење брзине којом се повећава број транзистора на једном чипу (тзв. *Moore*-ов закон). Последица тог смањења је да перформансе и енергетска ефикасност више не могу да се побољшавају повећањем броја транзистора и смањењем њихових димензија. Већ сада добици у перформансама примарно долазе из унапређења микро-архитектуре. Ово доводи до додатног ојачања већ монополистичких позиција које велике компаније у овом простору имају и тиме негативно утиче на иновативност у том пољу.

Истраживачи на Универзитету Калифорнија, Беркли су увидели ове проблеме те су дизајнирали и објавили *RISC* процесорску архитектуру отвореног кода под именом *RISC-V* [1] (изговара се *RISC five*). *RISC-V* архитектура је дизајнирана тако да буде довољно једноставна за примену у едукацији а такође и довољно моћна за примену у истраживачким[2] и комерцијалним[3][4] пројектима. То је постигнуто модуларним приступом где постоји основна архитектура и мноштво опционих екстензија. Зато је *RISC-V* архитектура одабрана за имплементацију на процесору који је тема овог рада.

Проблем којим се овај рад директно бави се тиче отклањања грешака у софтверу који се извршава на *RISC-V* процесорима. Временом софтвер постаје све комплекснији а самим тиме и подложнији грешкама. Један од најчешћих алата које инжењери користе при проналажењу грешака је дебагер који омогућава контролу извршавања програма и увид у његово стање. Уобичајено је да подршку за дебаговање пружа оперативни систем, међутим већина уграђених (енг. *embedded*) система имају јако просте (или уопште немају) оперативне системе. Код таквих система се подршка за дебаговање реализује директно у хардверу, тако што се инжењеру обезбеђује посебан интерфејс преко којег се (уз помоћ посебног адаптера) циљни систем повезује на софтвер који управља дебаговањем а извршава се на десктоп рачунару инжењера.

Како је подршка за екстерно дебаговање неопходна за било коју озбиљну комерцијалну имплементацију, *RISC-V* организација је дефинисала спецификацију за екстерно дебаговање[5], коју је већина комерцијалних имплементација усвојила.

Циљ рада је имплементација *RISC-V* процесора са подршком за екстерно дебаговање која поштује званичну спецификацију[5] ради бољег разумевања спецификације, проналажења потенцијалних унапређења спецификације и процене комплексности имплементирања спецификације у другим истраживачким пројектима.

Имплементација је реализована на *Field Programmable Gate Array (FPGA)* чипу, те су поред симулација вршени тестови и на конфигурисаном хардверу. Сама имплементација се састоји од самог *RISC-V* процесорског језгра (у даљем тексту само језгро), меморије за програм и податке, модула који обезбеђује подршку за екстерно дебаговање (у даљем тексту *Debug Module (DM)*), модула који омогућава повезивање са рачунаром који управља дебаговањем (у даљем тексту *Debug Transport Module (DTM)*) и неколико периферија које доприносе живописнијој демонстрацији система. Ове компоненте су међусобно повезане коришћењем три магистрале, прва повезује језгро, меморију и периферије, друга повезује језгро и *DM* и трећа повезује *DM* и *DTM*.

У 2. поглављу је дат кратак опис *RISC-V* спецификације[6] и спецификације за екстерно дебаговање[5], поређење са једном комерцијалном имплементацијом, као и опис технологија и алата коришћених при изради рада. Поглавље 3 даје нешто детаљнији преглед целог система и његових мање битних компоненти. Поглавља 4, 5 и 6 се фокусирају на детаље имплементације језгра, *DM*-а и *DTM*-а, док се поглавље 7 односи на софтверску страну екстерног дебаговања и њену конфигурацију. На крају се у поглављу 8 приказује методика и резултати тестирања.

2 Преглед спецификација и коришћених технологија

За читаоце који нису упућени у *RISC-V* екосистем, у овом поглављу је дат кратак преглед најбитнијих делова спецификације. Такође је приказано и поређење имплементације дате у овом раду са *Si Five Freedom E310-G002* процесором коришћеном на *Si Five HiFive RevB*[7] развојној плочи. Тај процесор је изабран због своје релативне популарности и зато што је дизајниран за примену у уграђеним системима и као такав је по перформансама и способностима најближи имплементираним процесору. На крају је дат и кратак опис технологија и алата коришћених у изради рада.

2.1 *RISC-V* инструкцијски сет

Како би постигао примењивост у широком опсегу имплементација са различитим циљевима, *RISC-V* користи модуларан приступ, спецификација[6] прописује неколико основних инструкцијских сетова, као и велики број опционих екстензија. Основни инструкцијски сет прописује око 40 обавезних инструкција које обухватају аритметичке и логичке инструкције, инструкције за приступ меморији и инструкције контроле тока. Тренутно су ратификована два основна инструкцијска сета: *RV32I* и *RV64I*. Оба сета прописују 32 регистра опште намене а разликују се у ширини регистара (самим тиме и у величини меморијског простора) која је 32 и 64 бита респективно. Такође су предложена још два основна сета, један који смањује број регистра на 16 и један који повећава ширину регистра и меморијског простора на 128 бита.

Поред основног инструкцијског сета, *RISC-V* организација је ратификовала и 8 опционих екстензија које проширују способности процесора.

То су:

- **M** - Целобројно множење, дељење и остатак при дељењу
- **A** - Атомичне операције над меморијом
- **F** - Операције над бројевима у покретном зарезу једноструке прецизности
- **D** - Операције над бројевима у покретном зарезу двоструке прецизности
- **Q** - Операције над бројевима у покретном зарезу четвороструке прецизности
- **C** - Компримоване (16 битне) инструкције
- **Zicsr** - Читање и писање контролних и статусних регистра
- **Zifencei** - Синхронизација уписа у програмску меморију

RISC-V такође предлага и привилеговану архитектуру[8] која садржи 3 нивоа извршавања: машински (*M*), супервизорски (*S*) и кориснички (*U*). Предложена привилегована архитектура такође подржава виртуализацију и садржи опис процедуре обраде прекида.

2.2 Подршка за екстерно дебаговање

Спецификација подршке за екстерно дебаговање[5] се састоји из 4 дела чији су мањи или већи делови опциони. То су: *DM*, *DTM*, посебан ниво извршавања у коме се процесор налази док је заустављен од стране дебагера (енг. *debug (D)* мод) и хардверски окидачи.

Debug Module (*DM*) прима команде које долазе од софтвера који управља дебаговањем (које долазе преко *DTM*-а и *Debug Module Interface (DMI)*-а) и на основу њих управља једним или више језгара. Већина делова *DM*-а су опциони али је неопходно да имплементирани делови омогућавају све потребне операције за успешно дебаговање, сама спецификација предлага два подскупа спецификације који испуњавају овај услов. *DM* мора имплементирати контролу ресет сигнала језгра, механизам за покретање и заустављање језгра и приступ регистрима језгра. *DM* опционо може подржати приступ меморији из погледа језгра или коришћењем додатног гатче на магистралама, извршавање произвољних инструкција и приступ контролним и статусним регистрима.

Debug Transport Module (*DTM*) прима команде које долазе од софтвера који управља дебаговањем одабраним протоколом и преводи их у приступе *DMI* магистралама. *DTM* може користити било који протокол али у спецификацији постоји само опис *DTM*-а који користи *Joint Test Action Group (JTAG)* протокол. Као такав, *DTM* је неопходан али није неопходно да имплементирани *DTM* користи *JTAG* протокол.

Имплементација *D* мода је обавезна и састоји се од малих промена понашања језгра у односу на *M* мод извршавања и неколико додатних контролних и статусних регистара доступних само *DM*-у.

Хардверски окидачи су опциони али могу бити имплементирани и независно од остатка спецификације јер могу бити корисни и када је подршка за дебаговање имплементирана у софтверу. Спецификација омогућава произвољан број хардверских окидача као и произвољан избор које функционалности окидача су имплементиране. У основи постоје 4 типа окидача који унутар себе имају велики број функционалности које су опције и у које се неће улазити сада.

- Окидач на адресу или податак учитан из меморије (ово обухвата читање и упис у меморију као и дохватање и извршавање инструкције)
- Окидач на број извршених инструкција (ово је један од начина за имплементацију проласка кроз програм инструкцију по инструкцију)
- Окидач на обраду изузетка
- Окидач на обраду прекида

Када се окидач окине, у зависности од конфигурације, језгро може генерисати изузетак или прећи у *D* мод и стати са извршавањем.

2.3 Поређење са *Si Five Freedom E310-G002*

На следећој табели за налази поређење имплементираног процесора и подршке за дебаговање са *Si Five Freedom E310-G002*[9] по неким од параметара изнетих у претходне две секције.

Табела 2.1: Поређење *Si Five Freedom E310-G002* са имплементираним процесором

	<i>Si Five Freedom E310-G002</i>	Имплементирани процесор
Основни инструкцијски сет	RV32I	RV32I
Подржане екстензије	M, A, C, Zicsr	Zicsr
Подржани модови извршавања	M, U	M
Организација	Проточна обрада са 5 корака	Вишециклична
Максимални ИПЦ ¹	1	1
Фреквенција сигнала такта	до 384 MHz	35 MHz
Механизам за покретање, ресетовање и заустављање језгра	Да	Да
Приступ регистрима опште намене	Да	Да
Приступ контролним и статусним регистрима	Не	Да
Приступ регистрима без заустављања језгра	Не	Не
Извршавање произвољних инструкција	Да	Да
Величина бафера за произвољне инструкције	16 меморијских речи	16 меморијских речи
Приступ меморији из погледа језгра	Не	Да
Приступ меморији из погледа језгра без заустављања језгра	Не	Не
Број помоћних регистара ²	1 + 1	2 + 12
Приступ меморији коришћењем додатног газде на магистрали	Не	Да
Број хардверских окидача	8	4
Окидач на адресу или податак подржан	Да	Да ³
Окидач на број извршених инструкција подржан	Не	Не
Окидач на обраду изузетка подржан	Не	Не
Окидач на обраду прекида подржан	Не	Не
<i>DTM</i> протокол	<i>JTAG</i>	<i>JTAG</i>
Препоручена фреквенција <i>JTAG</i> интерфејса	4MHz	1MHz
Интегрисан <i>JTAG</i> адаптер	Да (J-Link OB)	Не

¹ Инструкције По Циклусу (ИПЦ)² Мапираних као контролни и статусни регистри + меморијски мапираних³ Окидач не подржава комплетан сет опционих функционалности

Као што се може видети на табели изнад, *E310-G002* је знатно софистициранији процесор, што је и разумљиво, јер је у питању комерцијална имплементација. Међутим подршка за дебаговање на *E310-G002* обухвата један од препоручених минималних подскупа функционалности. Имплементирани подскуп функционалности се базира на извршавању произвољних инструкција (функционалности које нису директно подржане се могу емулирати извршавањем произвољног кода који користи помоћне регистре).

Како је фокус рада примарно на подршци за дебаговање, нешто једноставнији процесор са опширнијом подршком за дебаговање представља логичан избор.

2.4 Преглед коришћених технологија и алата

2.4.1 *FPGA*

Field Programmable Gate Array (FPGA) су интегрисана кола чија се функционалност може мењати по потреби. За разлику од процесора који су такође програмабилни, *FPGA* не извршава код већ директно имплементира тражени дизајн на нивоу дигиталне логике. Дизајн за *FPGA* је репрезентован битским током (енг. *bitstream*) који посебни алати за синтезу генеришу користећи дизајн написан у неком од језика за опис хардвера.

FPGA чипови су интерно реализовани коришћењем логичких елемената (енг. *Logic Element (LE)*) који се састоје од лукап табеле (која може да репрезентује произвољну комбинациону логику) и Д флип-флопа. Велики број логичких елемената (неколико десетина хиљада) је међусобно повезано конфигурабилним везама тако да се произвољни улази и излази логичких елемената могу повезати. *FPGA* чипови поред ове конфигурабилне логике често имају и додатне компоненте које олакшавају имплементацију одређених решења, то су обично интегрисане меморије, сабирачи, множачи, фазно закључане петље (енг. *Phase Locked Loop (PLL)*) итд.

За имплементацију процесора приказаног у овом раду коришћен је *Altera Cyclone V 5CSXFC6D6F31C6N*[10] *FPGA* чип. *5CSXFC6D6F31C6N* садржи 110 хиљада логичких елемената, 5761 килобита интегрисане меморије, 6 фазно закључаних петљи, 2 интегрисана меморијска контролера и двојезгарни *ARM* микропроцесор (меморијски контролери и *ARM* микропроцесор нису коришћени у овом раду).

2.4.2 *System Verilog*

System Verilog[11] је језик за опис хардвера, настао као проширење на *Verilog*. Језици за опис хардвера омогућавају дизајнерима хардвера да коришћењем програмског кода, на једноставан начин опишу структуру и понашање жељеног хардверског дизајна. Дизајн описан у језику за опис хардвера се након креирања може симулирати, претворити у шаблон за производњу интегрисаних кола или битски ток за конфигурацију *FPGA* чипа.

Једна од предности *System Verilog*-а у односу на *Verilog* је постојање интерфејса који представљају именовану групу сигнала, што их чини идеалним за репрезентовање магистрала и представља примарни разлог његовог избора за имплементацију процесора у овом раду.

2.4.3 *Quartus* и *Questa*

За синтезу дизајна написаног у језику за опис хардвера коришћен је *Intel Quartus Prime Lite*[12] верзија 22.1std.1, док је за симулацију и прелиминарно тестирање дизајна коришћена *Questa Intel Starter FPGA Edition-64*[13] верзија 2012.2.

2.4.4 *Eclipse Embedded CDT*

Eclipse је лако прошириво интегрисано развојно окружење (енг. *Integrated Development Environment (IDE)*) које подржава велики број програмских језика, а уз *Embedded C/C++ Development Tools (CDT)*[14] сет екстензија је специјализовано за развој уграђених система. *Embedded CDT* сет екстензија између осталог садржи подршку за *Make* систем за превођење кода, као и директну подршку за дебаговање коришћењем *OpenOCD*-а или *J-Link* софтвера. Из тих разлога је одлучено да се за сав развој и дебаговање софтвера који се извршава на имплементираном процесору користи *Eclipse Embedded CDT* верзија 2023-06.

2.4.5 *GCC* и *OpenOCD*

GNU Compiler Collection (GCC)[15] је добро познати пакет програмских преводилаца и пратећих алата, који обухвата и укрштени преводилац (енг. *cross-compiler*) за *RISC-V* архитектуру. За превођење софтвера писаног у *C* програмском језику за имплементирани процесор коришћена је верзија 13.2.0 *GCC*-а која је део пакета *xPack GNU RISC-V Embedded GCC x86_64* који је препоручен од стране *Eclipse Embedded CDT* развојног окружења.

Open On-Chip Debugger (OCD)[16] је пројекат отвореног кода који за циљ има да премости јаз између унификованог интерфејса који *GNU Debugger (GDB)* пружа програмерима и интерфејса који пружа сам сиљни процесор. Ово је реализовано кроз три нивоа апстракције. На самом дну имамо ниво адаптера за дебаговање, који комуницира са адаптером за дебаговање и користећи њега извршава просте операције комуникационог протокола који циљни процесор користи. Изнад тога се налази ниво архитектуре процесора који операције као што су читање регистара или заустављање процесора преводи у секвенцу операције комуникационог протокола који се користи. На врху се налази ниво конкретних система где се архитектури процесора додају информације о броју језгара, меморијској мапи и повезаним периферијама.

Како *OpenOCD* већ има уграђену подршку за верзију спецификације[5] коришћену у овом раду, као и *J-Link* (коришћени адаптер за дебаговање), потребно је само дефинисати конфигурациони фајл за ниво система. У раду је коришћена верзија 0.12.0, такође део пакета *xPack OpenOCD* препорученог од стране *Eclipse Embedded CDT*.

2.4.6 *JTAG*

Joint Test Action Group (JTAG)[17] је комуникациони протокол, оригинално дизајниран за тестирање интегрисаних кола и штампаних плоча, али је убрзо усвојен и од стране произвођача микропроцесора као протокол за комуникацију са подршком за екстерно дебаговање. Протокол ће бити детаљније објашњен у поглављу 6 које се бави имплементацијом *DTM*-а.

2.4.7 *J-Link*

J-Link[18] је један од најпопуларнијих адаптера за дебаговање који подржава *JTAG* протокол. *J-Link* се може у овом случају користити на два начина, директно са пратећим софтвером или коришћењем *OpenOCD*-а. Уколико се користи са пратећим софтвером, тај софтвер игра сличну улогу као *OpenOCD* али због бољег познавања хардвера може боље да га искористи што резултује у бољем времену одзива при дебаговању. Уколико се користи *OpenOCD*, *J-Link* се понаша као прост адаптер и *OpenOCD* мора да управља њиме на нижем нивоу. Иако су предности првог приступа очигледне, у духу пројекта отвореног кода у поглављу 7 ће бити приказана оба метода.

Литература

- [1] RISC-V International. *RISC-V International. RISC-V: The Open Standard RISC Instruction Set Architecture*. Септ. 13, 2023. URL: <https://riscv.org/>.
- [2] Krste Asanović и др. *The Rocket Chip Generator*. Техн. изв. UCB/EECS-2016-17. EECS Department, University of California, Berkeley, 2016. URL: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html>.
- [3] SiFive Inc. *SiFive. Leading the RISC-V Revolution*. Септ. 13, 2023. URL: <https://www.sifive.com/>.
- [4] Tenstorrent Inc. *Tenstorrent. Scalable and Efficient Hardware for Deep Learning*. Септ. 13, 2023. URL: <https://tenstorrent.com/>.
- [5] *RISC-V External Debug Support*. Уп. Tim Newsome. Уп. Megan Wachs. Верзија 0.13.2. RISC-V Foundation, Мар. 22, 2019.
- [6] *The RISC-V Instruction Set Manual, Volume I: User-Level ISA*. Уп. Andrew Waterman. Уп. Krste Asanović. Верзија 20191213. RISC-V Foundation, Дец. 13, 2019.
- [7] SiFive Inc. *HiFive1 Rev B. SiFive*. Септ. 14, 2019. URL: <https://www.sifive.com/boards/hifive1-rev-b>.
- [8] *The RISC-V Instruction Set Manual, Volume II: Privileged Architecture*. Уп. Andrew Waterman. Уп. Krste Asanović. Уп. John Hauser. Верзија 20211203. RISC-V International, Дец. 4, 2021.
- [9] SiFive Inc. *SiFive FE310-G002 Manual*. Верзија v1p5. Септ. 22, 2022.
- [10] Altera Corporation. *Cyclone V Device Handbook, Volume 1: Device Interfaces and Integration*. Верзија CV-5V2. Јул 5, 2022.
- [11] „IEEE Standard for SystemVerilog--Unified Hardware Design, Specification, and Verification Language”. У: *IEEE Std 1800-2017 (Revision of IEEE Std 1800-2012)* (2018), стр. 1–1315. DOI: 10.1109/IEEESTD.2018.8299595.
- [12] Intel Corporation. *Intel Quartus Prime Standard Edition User Guides - Combined*. Дец. 16, 2019.
- [13] Intel Corporation. *Questa Intel FPGA Edition Simulation User Guide*. Јун 7, 2023.
- [14] Liviu Ionescu и др. *Eclipse Embedded CDT. Eclipse Embedded CDT (C/C++ Development Tools)*. Септ. 15, 2023. URL: <https://eclipse-embed-cdt.github.io/>.
- [15] Free Software Foundation Inc. *GCC online documentation. GNU Project*. Септ. 15, 2023. URL: <https://gcc.gnu.org/onlinedocs/>.
- [16] The OpenOCD Project. *OpenOCD User's Guide. Top*. Септ. 15, 2023. URL: <https://openocd.org/doc/html/index.html>.

- [17] „IEEE Standard for Test Access Port and Boundary-Scan Architecture”. Y: *IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001)* (2013), стр. 1–444. DOI: 10.1109/IEEESTD.2013.6515989.
- [18] Segger Microcontroller. *Debug Probes - J-Link & J-Trace. SEGGER Wiki*. Сент. 15, 2023. URL: https://wiki.segger.com/Debug_Probes_-_J-Link_%26_J-Trace.

Списак скраћеница

DM Debug Module.

DMI Debug Module Interface.

DTM Debug Transport Module.

FPGA Field Programmable Gate Array.

JTAG Joint Test Action Group.

LE Logic Element.

PLL Phase Locked Loop.

RISC Reduced Instruction Set Computer.

ИПЦ Инструкције По Циклусу.

Списак слика

Списак табела

2.1	Поређење <i>Si Five Freedom E310-G002</i> са имплементираним процесором	5
-----	---	---