

16-bitni RISC procesor sa fon Nojmanovom arhitekturom

Nebojša Z. Čirović, Stevan D. Marinković, Dragoljub S. Petrović

Mentor: prof. dr Lazar Saranovac, prof. dr Jelena Popović

Sadržaj — Namenski računarski sistemi se u velikoj meri baziraju na jednostavnim i pouzdanim procesorima. U radu je opisan projekat 16-bitnog RISC procesora sa fon Nojmanovom arhitekturom, koji je pogodan za namenske sisteme. Opisane su operaciona i upravljačka jedinica projektovanog procesora, kao i procesorski interfejs, a predstavljene su i najznačajnije osobine: skup instrukcija, načini adresiranja, mapiranje adresnog prostora, trajanje izvršavanja instrukcija, banke registara, generator konstante i mogućnost rada sa 16-bitnim i 8-bitnim operandima.

Ključne reči — Fon Nojmanova arhitektura, instrukcije, načini adresiranja, operaciona jedinica, RISC procesor, upravljačka jedinica.

I. UVOD

DANAŠNJI veoma brz razvoj hardvera značajno je pomognut postojanjem FPGA čipova. Međutim, da bi se u potpunosti iskoristila mogućnost projektovanja raznovrsnog hardvera, na tržištu postoje IP procesori koji se na jednostavan način mogu koristiti za upravljanje i testiranje projektovanih uređaja. IP procesori koji su dostupni studentima uglavnom su previše jednostavni, a njihova funkcionalnost je značajno ograničena. Ovo je bio razlog da se projektuje IP procesor sa dobrim hardverskim i softverskim resursima, a da bude pogodan za implementaciju na FPGA čipovima.

Namenski računarski sistemi zahtevaju pored procesora i raznovrsne periferijske uređaje, pa se kao logičan izbor za realizaciju ovakvih sistema nameću mikrokontroleri. Iz tog razloga jedan od osnovnih ciljeva pri projektovanju procesora je kasnija nadogradnja implementacijom različitih periferija. Radi jednostavnosti i pouzdanosti, za procesor je izabrana 16-bitna RISC, fon Nojmanova arhitektura.

Detaljan projekat IP procesora, opisan je u diplomskom radu Nebojše Čirovića [1], dok se detalji VHDL implementacije mogu pronaći u diplomskom radu Stevana Marinkovića [2], a opis realizacije asemblerskog prevodioca u diplomskom radu Dragoljuba Petrovića [3].

Nebojša Z. Čirović, Elektrotehnički fakultet u Beogradu, Srbija (telefon: +381-64-2173130; e-mail: nebojsachiro@yahoo.com).

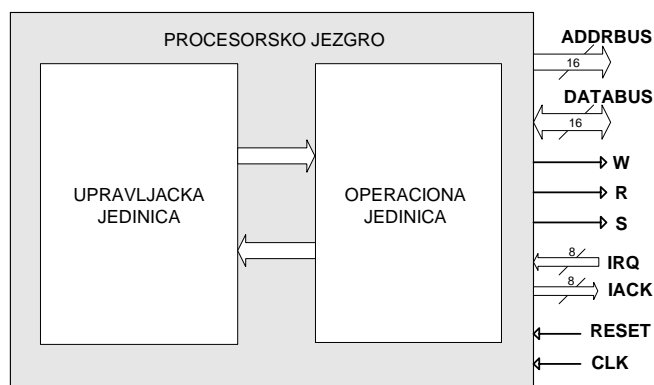
Stevan D. Marinković, Elektrotehnički fakultet u Beogradu, Srbija (telefon: +381-64-1852528; e-mail: stevankg2004@yahoo.com).

Dragoljub S. Petrović, Elektrotehnički fakultet u Beogradu, Srbija (telefon: +381-11-602100; e-mail: srbapet@gmail.com).

II. PREGLED OSOBINA PROJEKTOVANOG PROCESORA

A. Interfejs procesora

Struktura procesora prikazana je na slici 1 gde su predstavljeni i signali koji pripadaju procesorskom interfejsu. Operaciona i upravljačka jedinica biće opisane u narednim poglavljima.



Sl. 1. Struktura i interfejs procesorskog jezgra

Funkcija signala procesorskog interfejsa je sledeća. ADDRBUS_{15..0} je 16-bitna adresna magistrala. DATABUS_{15..0} je 16-bitna magistrala podataka. Signal W označava magistralni ciklus upisa, a signal R označava magistralni ciklus čitanja, pri čemu je aktivan nivo logičke jedinice. Signal S određuje da li je magistralni ciklus sa 8-bitnim ('1') ili 16-bitnim ('0') podatkom. IRQ_{7..0} predstavlja 8 signala prekida sa aktivnom usponskom ivicom različitog prioriteta. IACK_{7..0} predstavlja signale prihvatanja odgovarajućih prekidnih zahteva. Signal RESET predstavlja asinhroni reset procesora sa aktivnim nivoom logičke jedinice. CLK je signal takta.

B. Najznačajnije osobine procesora

Procesor je 16-bitni i projektovan je korišćenjem fon Nojmanove arhitekture, tj. sa jedinstvenim adresnim prostorom za pristup instrukcijama i podacima. Ipak podržan je rad sa instrukcijama koje se izvršavaju i nad 16-bitnim i nad 8-bitnim podacima.

Skup instrukcija se sastoji od 27 instrukcija sa 7 podržanih načina adresiranja pri čemu je poštovana ortogonalna arhitektura, tj. mogućnost korišćenja svake instrukcije sa bilo kojim načinom adresiranja. Omogućen je i transfer iz memorije u memoriju, bez međukoraka sa prihvatanjem operanda u neki od registara opšte namene.

Omogućen je potpuni pristup svim registrima procesora, uključujući programski brojač (PC), pokazivač steka (SP) i statusni registar (PSW). Osim nabrojanih registara postoje i 4 banke, sa po dvanaest 16-bitnih „shadow“ registara opšte namene. Veliki broj registara omogućava brzo izvršavanje programa s obzirom da je trajanje instrukcija sa registarskim načinom adresiranja minimalano, jedan takti ciklus, zahvaljujući „pipeline“ strukturi dubine jedan.

Procesor, zahvaljujući generatoru konstanti (CG), ima mogućnost formiranja šest najčešće korišćenih neposrednih vrednosti, što umanjuje veličinu programa.

C. Adresni prostor i mapiranje memorije

Adresni prostor procesora je zajednički za programsku memoriju, memoriju za podatke i za periferije. Ukupno je moguće adresirati 64 KB memorije. Raspodela memorije nije fiksna, već zavisi od sistema u kom se procesor implementira.

Na vrhu memorijskog prostora nalazi se programska memorija, pri čemu je najviša adresa 0FFFFEh rezervisana za adresu početka programa. Po resetu, procesor počinje izvršavanje programa od instrukcije koja se nalazi na adresi koja je upisana u lokaciju 0FFFFEh.

Na osam 16-bitnih lokacija, na adresama od 0FFFECh do 0FFFCh, nalaze se prekidni vektori dodeljeni odgovarajućim prekidnim signalima. Najnižoj adresi u tabeli prekidnih vektora dodeljen je prekid izazvan signalom IRQ₀, dok je najvišoj adresi dodeljen IRQ₇. Ostatak programske memorije koristi se za smeštanje programa i podataka. Pristup programskoj memoriji može biti i 16-bitni i 8-bitni kada su u pitanju podaci, dok je pristup instrukcijama isključivo 16-bitni.

U ostatku memorijskog prostora moguće je postaviti RAM memoriju kao i registre potrebnih periferija u sistemu. Pristup može biti i 16-bitni i 8-bitni. Nema dinamičkog menjanja širine magistrale, tako da 16-bitne reči moraju biti na parnim adresama, dok 8-bitne reči mogu biti i na parnim i na neparnim adresama, mada su uvek na nižih 8-bitna magistrale podataka.

III. SKUP INSTRUKCIJA I NAČINI ADRESIRANJA

A. Skup instrukcija

U procesoru je implementirano 27 instrukcija. Ugrađene instrukcije omogućavaju formiranje većeg broja asemblerski emuliranih instrukcija.

Sve instrukcije se sastoje od 16-bitnih reči i mogu biti dužine od jedne do tri reči, pri čemu dodatne dve reči mogu biti ili neposredni podaci ili delovi adrese i ne sadrže kod instrukcije.

Postoje tri formata instrukcija:

- instrukcije skoka,
- jednoadresne instrukcije,
- dvoadresne instrukcije.

Skup instrukcija i formati instrukcija formirani su po uzoru na procesore kompanije *Texas Instruments* iz serije MSP430, pa se detaljan opis može pronaći u okviru korisničkog uputstva navedenog u literaturi [3].

B. Načini adresiranja

Postoji sedam načina adresiranja: registarsko, indeksno, apsolutno, simboličko, indirektno registarsko, indirektno registarsko sa autoinkrementom i neposredno.

U slučaju jednoadresnih instrukcija operand se može adresirati na bilo koji od navedenih načina. U slučaju dvoadresnih instrukcija izvorišni operand se može adresirati na svih sedam načina, dok se odredišni operand može adresirati samo na prva četiri načina.

Svi načini adresiranja su po uzoru na procesore kompanije *TI* iz serije MSP430 [3].

IV. OPERACIONA JEDINICA

A. Struktura operacione jedinice

Procesor je podeljen na dva bloka, operacionu i upravljačku jedinicu, kao što je prikazano na slici 1.

Operaciona jedinica predstavlja skup kombinacionih i sekvencijalnih logičkih kola koja omogućavaju procesoru prihvatanje instrukcija i podataka iz „spoljnog sveta“ kao i za slanje podataka van procesora. U operacionoj jedinici se izvršavaju instrukcije na osnovu signala koje generiše upravljačka jedinica.

Procesor je projektovan za VHDL implementaciju u FPGA čipu pomoću programskog paketa *Xilinx ISE 8.1*, tako da je cela sekvencijalna logika u procesoru, pa samim tim i u operacionoj jedinici, sinhronizovana na usponsku ivicu taktnog signala. Navedeni nedostatak je posledica ograničenja samog programskog paketa, koji ne podržava osetljivost VHDL procesa na obe ivice takta, a prevaziđen je implementacijom „pipeline“-a dubine jedan.

Najvažniji delovi operacione jedinice su 16-bitni registri koji se „ne vide“ van procesora: MAR, MBR, PIR, IR i PR, zatim aritmetičko-logička jedinica (ALU), blok za prihvatanje prekida i 16-bitni registri koji su dostupni programeru PC, SP, PSW, CG i 4 banke opštenamenskih registara. Operaciona jedinica je projektovana tako da svi delovi budu međusobno povezani, pri čemu aktivne veze odeduju multiplekseri na osnovu upravljačkih signala. Međutim, zbog velikog broja registara opšte namene, postoje i dve interne magistrale koje omogućavaju brzo izvršavanje instrukcija nad operandima smeštenim u registrima.

Registar MAR služi za postavljanje adresne magistrale. Registar MBR omogućava postavljanje i prihvatanje podataka preko magistrale podataka. Registar PR predstavlja pomoćni registar u slučaju prihvatanja dva operanda iz memorije.

Registar PIR je pomoćni instrukcijski registar i omogućava „pipeline“ dubine jedan. Ukoliko ne dođe do narušavanja „pipeline“-a, zbog instrukcije skoka ili neke druge instrukcije koja ne dovodi do inkrementiranja programskog brojača, u registru PIR se nalazi sadržaj sa adrese na koju pokazuje PC. Sadržaj registra PIR može biti instrukcija, neposredni podatak, adresa ili deo adrese u zavisnosti od načina adresiranja. Registar IR je instrukcijski registar u kome se nalazi instrukcija čije je izvršavanje u toku.

B. Aritmetičko-logička jedinica

ALU predstavlja kombinacionu mrežu koja na osnovu upravljačkih signala izvršava aritmetičke i logičke operacije. U skladu sa ugrađenim skupom instrukcija u procesoru, ALU mora da podrži rad sa 8-bitnim i sa 16-bitnim operandima, zatim da generiše statusne bite N, Z, V i C i da omogući izračunavanje rezultata operacija datih u tabeli 1.

TABELA 1: OPERACIJE ARITMETIČKO-LOGIČKE JEDINICE.

Naziv	Opis
RRA	Aritmetičko rotiranje operanda A udesno
RRC	Aritmetičko rotiranje operanda A udesno kroz C.
SWPB	Menja mesta višem i nižem bajtu operanda A
SXT	Vrši ekstenziju znaka nižeg bajta u viši.
MOV	Prosleđuje operand B kao rezultat operacije.
ADD	Sabiranje operanda A i B
ADDC	Sabiranje operanda A i B i bita prenosa Cin
SUBC	Oduzimanje op. B od op. A sa pozajmicom
SUB	Oduzimanje operanda B od operanda A
DADD	Sabiranje operanda A i B kao BCD brojeva.
BIC	operand A \wedge operand B
BIS	operand A \vee operand B
XOR	operand A $\underline{\vee}$ operand B
AND	operand A \wedge operand B

Sve operacije iz table 1 generišu statusne bite, ali pošto se ALU koristi i za interna izračunavanja, onda se generisani statusni biti upisuju u statusni registar samo na komandu upravljačke jedinice.

C. Registri dostupni programeru

Važna karakteristika procesora je postojanje velikog broja registara dostupnih programeru, ukupno 52 registra, od kojih su 48 „shadow“ registri opšte namene, podeljeni u 4 banke. Postojanje „shadow“ registara predstavlja najznačajniju razliku u odnosu na procesore iz serije MSP430. „Shadow“ registri su pogodni jer povećanje ukupnog broja registara ne utiče na formate instrukcija jer su podeljeni u banke, a brzina izvršavanja registarskih instrukcija dolazi do punog izražaja.

Dakle, istovremeno, na raspolaganju je 16 registara, pri čemu su prva četiri registra sa konkretnom funkcijom: programski brojač (PC), pokazivač steka (SP), statusni registar (PSW) i generator konstante (CG). Preostalih 12 registara, R4 do R15, su registri koji pripadaju aktivnoj banci registara. Aktivna banka se menja softverski, menjanjem bita B_i u statusnom registru (tabela 2), tako da programer može značajno ubrzati izvršavanje programa ako, na primer, pri ulasku u potprograme menja aktivnu banku čime dobija 12 praznih registara, a da se vrednosti do tada korišćenih registara čuvaju u neaktivnoj banci.

Svi registri imaju mogućnost izlaska na obe interne magistrale, što omogućava povećanje brzine izvršavanja registarskih instrukcija.

Na početku programa neophodna je inicijalizacija registra SP, i to parnom vrednosti. Resgistar SP pokazuje na poslednju zauzetu lokaciju na steku, koji raste na dole.

Statusni registar čuva statusne bite na osnovu kojih

upravljačka i operaciona jedinica generišu svoje signale. Značenje statusnih bita procesora prikazano je u tabeli 2.

TABELA 2: Značenje bita u statusnom registru.

Naziv	Opis
N	Bit znaka.
Z	Bit nule.
V	Bit prekoračenja.
C	Bit prenosa.
IE _{2..0}	Bitovi za podešavanje nivoa prioriteta dozvoljenog prekida.
B _{1..0}	Bitovi za podešavanje aktivne banke registara.

Registar CG, zapravo i nije registar, već kombinaciona mreža koja generiše šest 16-bitnih konstanti: 0, 1, 2, 4, 8 i -1. Generisanje željene konstante postiže se kombinacijom odgovarajućeg načina adresiranja sa registrima PSW i CG.

D. Blok za prihvatanje prekida

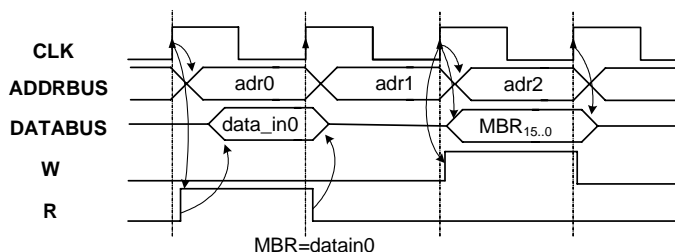
U interfejsu procesora navedeno je da procesor ima mogućnost prijema prekidnog zahteva po 8 linija, IRQ_{7..0}. Prekidni zahtev se javlja kad se na nekoj od linija IRQ_{7..0} pojavi usponska ivica. Da li će zahtev za prekid biti prihvaćen ili ne zavisi od vrednosti bita IE_{2..0}, u statusnom registru, kojim se prekidi po linijama IRQ_{7..1} mogu maskirati, a zavisi i od toga da li je istovremeno stigao zahtev za prekid višeg prioriteta.

Prioritet prekidnih zahteva je definisan hardverski tako da je linija IRQ₀ najvišeg prioriteta i predstavlja nemaskirajući prekid, a linija IRQ₇ je najnižeg prioriteta.

Svakoj liniji za prekid odgovara 16-bitna lokacija u memoriji u kojoj se nalazi adresa odgovarajuće prekidne rutine. Kada procesor prihvati prekid, generiše se signal IACK_i u trajanju jedne periode takta, koji potvrđuje da je prekidni zahtev po liniji IRQ_i prihvaćen.

E. Vremenski dijagrami magistralnih ciklusa

Magistralni ciklus čitanja predstavljen je na slici 2. Svi signali su sinhronizovani na usponsku ivicu signala takta. Procesor postavlja adresu na adresnoj magistrali i signal čitanja R. Na sledećoj usponskoj ivici takta, procesor smešta sadržaj sa magistrale podataka u registar MBR i obara vrednost signala R na logičku nulu, čime je proces čitanja završen.



Sl. 2. Vremenski dijagram magistralnih ciklusa.

Magistralni ciklus upisa predstavljen je na slici 2. Na usponsku ivicu takta, procesor postavlja adresu na adresnoj magistrali, podatak iz registra MBR na magistralu podataka i signal za upis W. Na sledećoj usponskoj ivici magistrala podataka se vraća u stanje visoke impedanse, a signal W se obara na nivo logičke nule, čime je ciklus upisa završen.

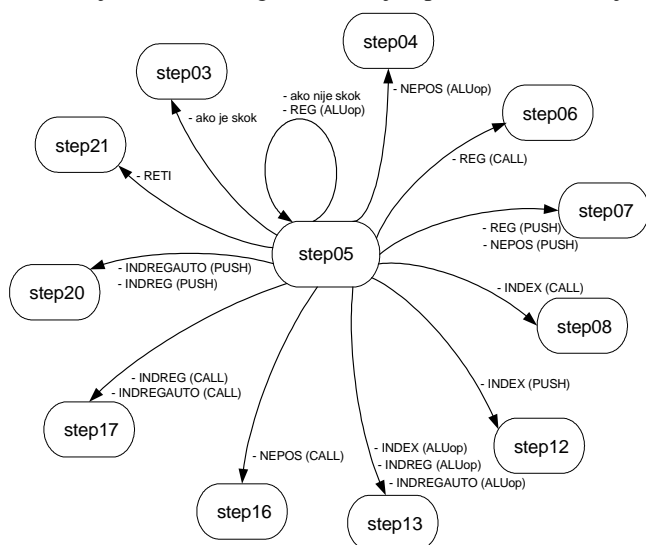
V. UPRAVLJAČKA JEDINICA

A. Mašina stanja

Upravljačka jedinica je blok koji upravlja radom operacione jedinice. Realizovana je kao mašina stanja, sa ukupno 37 stanja: $step_{00}$, $step_{01}$... $step_{36}$. Mašina stanja je formirana na osnovu toka izvršavanja instrukcija. Najznačajnije faze toka izvršavanja instrukcija su sledeće:

1. Po resetu u registar PC se upisuje sadržaj memorijske lokacije 0FFFFh, tj. adresa prve instrukcije.
2. Uspostavlja se „pipeline“, punjenjem registra PIR.
3. Smeštanjem svake nove instrukcije u registar IR počinje dekodovanje operacije i načina adresiranja.
4. Dovođenje adresiranih operanda, izvršavanje operacije i smeštanje rezultata.
5. Provera prekidnih zahteva. U slučaju da nema prekidnog zahteva, održava se „pipeline“ uz povratak na fazu 3. Ako prekidni zahtev postoji, prelazi se na fazu 6.
6. Sadržaj registara PC i PSW se smešta na stek, a zatim sledi povratak na fazu 2.

Cela mašina stanja detaljno je opisana pomoću pseudokoda na osnovu kog se može praviti VHDL kod. Jezgro mašine stanja predstavlja stanje $step_{05}$, jer se u njemu tumači tekuća instrukcija i načini adresiranja. Svaka instrukcija, u toku svog izvršavanja, prolazi kroz stanje



Sl. 3. Dijagram dela mašine stanja za opis prelaska u sledeće stanje iz stanja $step_{05}$.

$step_{05}$, što je demonstrirano na slici 3, gde je opisan prelazak u sledeće stanje iz stanja $step_{05}$ u slučaju jednoadresnih instrukcija i instrukcija skoka. Uz strelice su navedeni uslovi koji uzrokuju odgovarajući prelaz.

B. Trajanje instrukcija

Broj stanja kroz koje prolazi upravljačka jedinica u toku izvršavanja instrukcije određuje trajanje instrukcije, koje se ogleda u broju taktih ciklusa. Pošto je u procesoru preslikan skup instrukcija procesora iz serije MSP430, napravljen je uporedni pregled trajanja instrukcija.

Sve instrukcije skoka imaju istu brzinu izvršavanja. Ukoliko je ispunjen uslov za skok, instrukcija se izvršava u 3 taktih ciklusa, a ako nije onda se instrukcija izvršava u

jednom taktih ciklusu. U procesorima iz serije MSP430 instrukcije skoka traju 2 taktih ciklusa bez obzira da li dolazi do skoka ili ne.

Trajanje jednoadresnih i dvoadresnih instrukcija kreće se u opsegu od 1 do 7 taktih ciklusa u zavisnosti od načina adresiranja. U odnosu na procesore iz serije MSP430, razlika je ± 1 taktih ciklus.

Trajanje ulaska u prekidnu rutinu iznosi 7 taktih ciklusa, a povratak iz prekidne rutine traje 6 taktih ciklusa, što je za po jedan taktih ciklus duže nego kod procesora iz serije MSP430.

VI. ZAKLJUČAK

Projektovani procesor je realizovan u VHDL-u, implementiran u FPGA čipu *Spartan3E* serije, kompanije *Xilinx*, posle čega su izvršena testiranja na razvojnoj ploči.

Testovi su pokazali da procesor pouzdano radi na učestanosti 12,5 MHz. Ipak, treba naglasiti da to nije maksimalna učestanost rada, jer nije napravljena potpuna analiza te vrste, mada su neki manji testovi radili ispravno i na učestanosti 25 MHz. S obzirom da je procesor projektovan po ugledu na procesore iz serije MSP430, čija je maksimalna učestanost takta 8 MHz, a da je broj taktih ciklusa potrebnih za izvršavanje instrukcija veoma sličan, sledi da projektovani procesor poseduje veću brzinu rada čime je ispunjena i željena brzina rada koja nije bila prioritet pri projektovanju.

Ideja o stvaranju procesorskog jezgra pogodnog za projektovanje različitih namenskih sistema u FPGA čipu, opisana u uvodu, potpuno je ostvarena. U budućnosti treba očekivati dalji razvoj, kako procesora tako i periferija.

LITERATURA

- [1] N. Ćirović, „Projekat 16-bitnog RISC procesorskog jezgra sa fon Nojmanovom arhitekturom”, diplomski rad, Elektrotehnički fakultet, Beograd 2007.
- [2] S. Marinković, „FPGA implementacija 16-bitnog RISC procesora sa fon Nojmanovom arhitekturom”, diplomski rad, Elektrotehnički fakultet, Beograd 2007.
- [3] D. Petrović, „Asemblerski prevodilac za 16-bitni RISC procesor sa fon Nojmanovom arhitekturom”, diplomski rad, Elektrotehnički fakultet, Beograd 2007.
- [4] Texas Instruments „MSP430x44xx Family User's Guide”, Vol 3. pp. 41-115, 2006.

ABSTRACT

This paper explains organization of 16-bit RISC processor with von Neumann architecture suitable for FPGA implementation. It contains descriptions of main functional blocks, operational and control unit. Also it presents the processor interface and the most important features: instruction set, addressing modes, map of address space, banks of shadow registers, constant generator, interrupt handling and bus cycles. Described processor is successfully realized in VHDL and downloaded into the FPGA chip where all projected features are confirmed.

16-BIT RISC PROCESSOR WITH VON NEUMANN ARCHITECTURE

Nebojša Ćirović, Stevan Marinković, Dragoljub Petrović