

FPGA IMPLEMENTACIJA HIBRIDNOG ON-LINE NADZORA PROCESA ZA PC-BAZIRANE REAL-TIME SISTEME

Bojan Jovanović, *Elektronski fakultet u Nišu*, bojan@elfak.ni.ac.rs
Milun Jevtić, *Elektronski fakultet u Nišu*, milun.jevtic@elfak.ni.ac.rs

Nagrađeni rad mladog istraživača

Sadržaj– U radu je opisan jedan način FPGA implementacije hibridnog (baziranog na hardveru i softveru) sistema za on-line nadzor procesa u PC-baziranim Real-Time sistemima. Najpre su opisani razlozi uvođenja nadzora Real-Time sistema. Nakon toga dat je pregled različitih pristupa u izvođenju nadzora, zajedno sa njihovim prednostima i nedostacima. Na kraju je dat detaljan opis jednog hibridnog on-line nadzornog sistema implementiranog na FPGA čip i testiranog uz pomoć Alterine DE2 razvojne ploče. Prikazani su i rezultati FPGA implementacije, a pomenute su i moguće primene ovog nadzornog sistema.

1. UVOD

Sistem nadzora je proces ili set mogućih distribuiranih procesa čija je funkcija dinamičko prikupljanje, interpretacija i učešće u informacijama koje se tiču aplikacije dok se ta aplikacija izvršava [1]. Zbog toga se može reći da sistemi nadzora povećavaju vitalnost, bezbednost, toleranciju greške, i adaptabilnost Real-Time sistema (RTS-a). Da bi RTS ispravno funkcionisao, pored toga što na svojim izlazima treba da da ispravne rezultate, potrebno je još i da se ti rezultati pojave u tačno definisanim vremenskim intervalima. Ovo je naročito bitno za Hard Real-Time sisteme (HRTS) kod kojih izvršenje zadataka van zadatih vremenskih parametara može dovesti do katastrofe. Praćenjem toka događaja u RTS-u za vreme njegovog rada može se utvrditi da li RTS ispunjava ili ne potrebna vremenska ograničenja. Može se, prema tome, s dobrim razlogom reći da je on-line nadzor (nadzor sistema za vreme njegovog rada) procesa i događaja u HRTS-u od velikog značaja jer nam omogućava da predvidimo njegovo ponašanje. Implementiranjem on-line nadzora može se, za svaki proces/zadatak ili segment programskog koda, utvrditi da li se izvršava u prethodno utvrđenim vremenskim granicama (minimalno i maksimalno vreme izvršenja).

2. STRATEGIJE NADZORA RTS-A

Nadzorni sistem je nametljiv ako zahteva korišćenje resursa RTS-a koji nadzire (CPU vreme, I/O uređaje, komunikacione kanale itd.). Nadzorni sistemi su uglavnom manje ili više nametljivi. Potpuno nenametljiv nadzorni sistem koristi hardver specijalno dizajniran za potrebe nadzora. Postoje tri osnovna pristupa u implementaciji nadzornih modula RTS-a: Softverski, Hardverski i Hibridni pristup.

Softverski implementiran nadzor RTS-a je fleksibilan ali i u velikoj meri nametljiv. Zbog toga, ovako implementiran nadzorni modul značajno pogoršava vremenske karakteristike RTS-a.

Nadzorni modul baziran na hardveru je nenametljiv ali zahteva specijalizovani hardver za realizaciju nadzora. RTS koji se nadzire mora da podržava mogućnost njegove ugradnje. O tome treba voditi računa još u fazi projektovanja RTS-a.

Hibridni pristup u implementaciji nadzora RTS-a ima nenametljivu prirodu hardverskog pristupa i fleksibilnost softverskog pristupa. Zbog svojih osobina ovaj pristup predstavlja neku vrstu kompromisa između čisto hardverskog i čisto softverskog pristupa u implementaciji nadzora RTS-a.

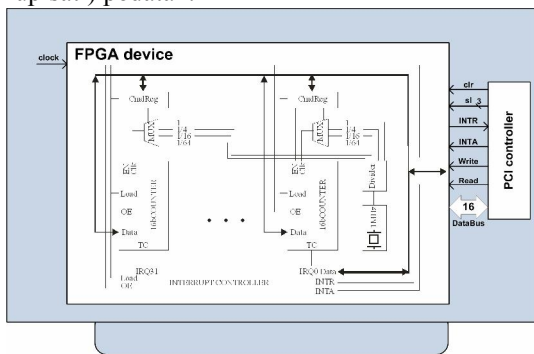
3. MODUL ZA HIBRIDNI ON-LINE NADZOR

Nadzorni modul predstavljen u ovom radu implementiran je na FPGA čip i primenljiv na svaki RTS koji treba da ispuni stroge vremenske zahteve nametnute od strane procesa koji se odvijaju u realnom vremenu. Ne zahteva značajno CPU vreme RTS-a koji nadzire kao ni glomazan dodatni hardver. Baziran je na FPGA programabilnoj komponenti i koristi hibridni pristup za realizaciju on-line nadzora RTS-a. Primenljiv je na RTS-e koji su bazirani na industrijskim PC-jevima i Linux operativnom sistemu kao široko prihvaćenom i dostupnom otvorenom (open source) operativnom sistemu. Implementirani modul nadzire do 32 procesa tj. RT zadatka i događaja koji se paralelno izvršavaju. Broj procesa koji se nadzire je relativno mali, ali treba reći da ih u HRTS-ima u industrijskim aplikacijama realno toliko i ima. Budući da implementirani modul za 32 procesa zahteva samo 23% FPGA resursa, kao što će se kasnije videti, broj procesa koji se nadzire može se lako proširiti. Sistem je baziran na dodatnom hardverskom modulu sa 32 programirajuća brojača-tajmera i interapt logikom [2]. Svakom procesu koji se nadzire pridružuje se jedan brojač-tajmer. Brojači-tajmeri koriste se kao uređaji za definisanje vremenskih trenutaka pojavljivanja događaja i kao watchdog tj. nadzorni tajmeri za proveru korektnog vremenskog izvršavanja zadataka. Sa ciljem da bude što manje nametljiv i koristi CPU vreme RTS-a, nadzorni modul koristi PCI slot PC-ja, kao što je prikazano na Sl. 1. Sa slike se može videti da nadzorni modul sa RTS-om komunicira preko sledećih signala: Data Bus, Read, Write, INTR, INTA, sl i clr.

Data Bus je 16-bitna bidirekciona magistrala. Preko nje se prenose podaci od RTS-a do nadzornog modula, i obrnuto. RTS aktivira Read (Write) signal svaki put kad treba da pročita podatak od nadzornog module (upiše podatak u nadzorni modul).

Nadzorni modul postavlja INTR (Interrupt Request) signal svaki put kada se neki od zadataka koji se trenutno izvršavaju ne izvršava pravilno ili se izvrši izvan zadatih vremenskih parametara. Kao odgovor, RTS čita poruku sa Data Bus magistrale i postavlja INTA (Interrupt Acknowledge) marker. Poruka sadrži informacije o prirodni prekida, kao i identifikaciju procesa koji je uzročnik prekida. Koje će se dalje akcije preduzeti zavisi isključivo od planera RTS-a. Kada vidi da je INTA marker postavljen, nadzorni modul resetuje INTR signal i nastavlja sa nadzorom RTS-a.

sl signal je 3-bitni select signal koji se koristi od strane RTS-a prilikom selekcije registra iz koga se želi pročitati (u koji se želi upisati) podatak.



Sl. 1. PCI kartica sa modulom za hibridni on-line nadzor

clr signal koristi se od strane RTS-a za resetovanje nadzornog modula.

RTS preko softverskih primitive upravlja radom nadzornog modula. Funkcije modula su sledeće:

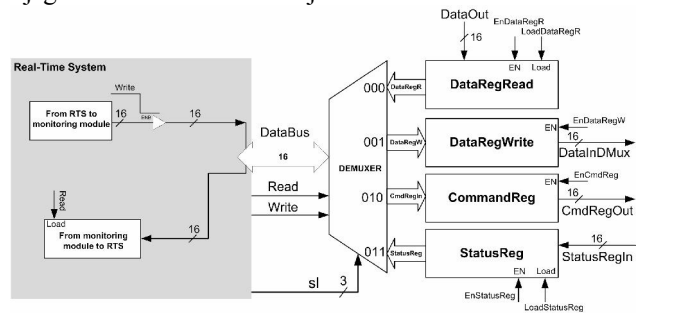
- Postavljanje moda (načina) rada brojača-tajmera
- Postavljanje vrednosti vremenskih ograničenja
- Dozvola rada brojača-tajmera
- Zabrana rada brojača-tajmera
- Čitanje vrednosti brojača-tajmera
- Obrada interapta od brojača-tajmera
- Poređenje vrednosti brojača-tajmera sa vremenskim ograničenjem.

Nadzorni sistem ima dva radna režima. Prvi režim rada odnosi se na analizu sistema koja se izvodi da bi se videlo koliko koji zadatak traje, odnosno koliko je vreme izvršenja svakog zadatka i da bi se u cilju pravilnog funkcionisanja RTS-a dobijene informacije mogle iskoristiti za buduću kontrolu sistema. U drugom režimu rada nadzorna kartica ima funkciju ugrađenog samotestiranja baziranog na potpunoj funkciji Watchdog-a. Proverava donju i gornju granicu vremena na nivou zadataka i periodičnih ili kvaziperiodičnih događaja. Svako aktiviranje izvršenja zadatka inicira proceduru startovanja nadzornog tajmera. Nadzorni tajmer se setuje na prethodno definisano maksimalno vreme izvršenja zadatka i startuje njegovo odbrojavanje. Ako dođe do prekoračenja vremenskog intervala, setuje se zahtev za interapt. Ukoliko se zadatak izvrši pre nego dođe do generisanja zahteva za interapt, okončanjem zadatka aktivira se procedura zaustavljanja tajmer-brojača, čita se njegovo stanje i proverava da li je zadatak izvršen pre vremena (nekorektno izvršen). Ukoliko je zadatak izvršen u regularnom vremenskom intervalu, rad RTS-a se nastavlja, u suprotnom se startuje predviđena procedura za oporavak sistema od detektovane greške. Na taj način se obezbeđuje predvidivo ponašanje HRTS-a.

Arhitektura nadzornog modula sastoji se od staze podataka, upravljačke jedinice i komunikacionog interfejsa između RTS-a i modula. Sa Sl. 2 može se videti da nadzorni modul komunicira sa RTS-om preko četiri različita 16-bitna registra: DataRegRead, DataRegWrite, CommandReg i StatusReg. Svi registri imaju signal dozvole (EN) preko koga se aktiviraju. DataRegRead i DataRegWrite su registri za pamćenje podataka. Pomoću DataRegRead registra RTS čita podatke sa nadzornog modula, dok uz pomoć DataRegWrite registra šalje podatke modulu.

Za slanje komande nadzornom modulu koristi se CommandReg registar. Status nadzornog modula smešten je

u StatusReg registru. Za pristup ovim registrima RTS koristi Read, Write i sl signale (Tabela 1). Na Sl. 3 prikazana je struktura na nivou bita komandnog i status registra. 10 MSB bitova StatusReg registra se ne koriste. StatusReg(5) čuva TCmin bit. Ovaj bit se postavlja na 1 svaki put kada se RT zadatak izvrši brže od minimalnog potrebnog vremena za njegovo korektno izvršavanje.



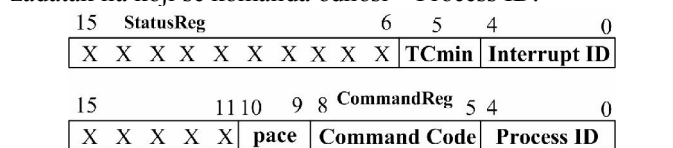
Sl. 2. Komunikacioni interfejs RTS - nadzornog modul

Budući da modul nadzire do 32 procesa, neophodno je minimalno 5 bitova za identifikaciju svakog od njih. Zato, 5 LSB bitova StatusReg registra sadrže identifikaciju RT zadatka (processa) koji je doveo do interapta – Interrupt ID.

Tabela 1 komunikacija RTS – nadzorni modul

sl	Read	Write	Selektovani reg	Akcija
000	1	0	DataRegRead	iz Reg u DataBus
001	0	1	DataRegWrite	iz DataBus u Reg
010	0	1	CommandReg	iz DataBus u Reg
011	1	0	StatusReg	iz Reg u DataBus

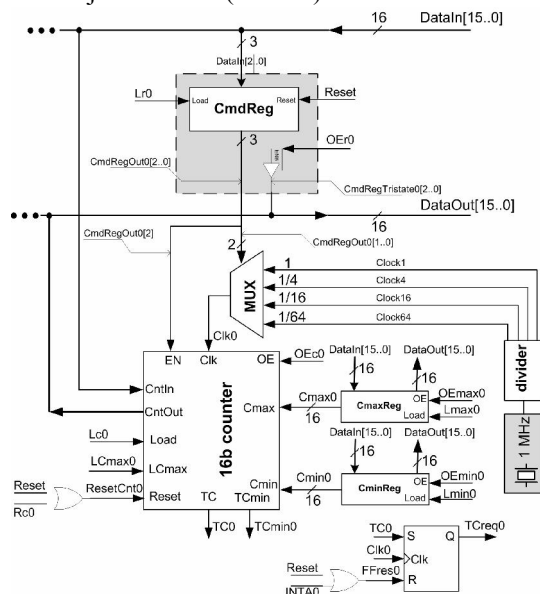
Što se tiče CommandReg registra, 5 MSB bitova se ne koriste. CommandReg(10:9) bitovi sadrže informaciju o vremenskom kvantu koji se koristi prilikom merenja različitih vremenskih intervala. CommandReg(8:5) bitovi sadrže informaciju o kodu komande koja se primenjuje na nadzorni modul, dok 5 LSB bitova ovog registra adresiraju zadatak na koji se komanda odnosi – Process ID.



Sl. 3. Struktura na nivou bita StatusReg i CommandReg

Staza podataka nadzornog modula: Na Sl. 4 prikazan je deo staze podataka potreban za nadzor jednog RT zadatka. Svaki od 32 zadatka ima istu arhitekturu. Staze podataka svih zadataka povezane su na 16-bitne DataIn i DataOut magistrale u cilju komunikacije sa RTS-om. Takti oscilator frekvencije 1MHz i delitelj frekvencije zajednički su za staze podataka svih procesa. Staza podataka za nadzor jednog RT procesa sastoji se od jednog 3-bitnog CmdReg registra, jednog 4u1 multipleksera, jednog 16-bitnog brojača koji broji unazad, dva 16-bitna registra za pamćenje C_{max} i C_{min} konstanti i jednog RS flip-flopa. MSB bit CmdReg registra koristi se za dozvolu/zabranu rada brojača, dok se njegova 2 LSB bita vode na selektorski ulaz multipleksera i time određuju frekvenciju taktovanja brojača (pace bitovi). Setovanjem Lr ulaza podaci sa DataIn magistrale upisuju se u CmdReg registar. Slično, setovanjem OEr ulaza sadržaj ovog registra dostupan je preko DataOut magistrale. Nakon reseta CmdReg registar je u stanju 000. Upotrebom taktnog oscilatora frekvencije 1MHz vremenski kvant za merenje vremena može biti 1, 4, 16 ili 64 μs . Shodno tome,

maksimalno vreme za izvršenje zadatka može biti 65.2, 262, 1048 ili 4194 ms. Treba reći da se promenom frekvencije taktnog oscilatora mogu se dobiti drugačiji vremenski kvantovi i maksimalna vremena izvršenja zadataka. CmaxReg i CminReg su 16-bitni registri namenjeni za smeštanje konstanti koje definišu maksimalno i minimalno vreme izvršenja RT zadatka, respektivno. I oni su povezani na DataIn i DataOut magistrale pa se u njih može upisivati setovanjem Lmax (Lmin) ulaza, ili se njihov sadržaj može čitati setovanjem OEmax (OEmin) ulaza.



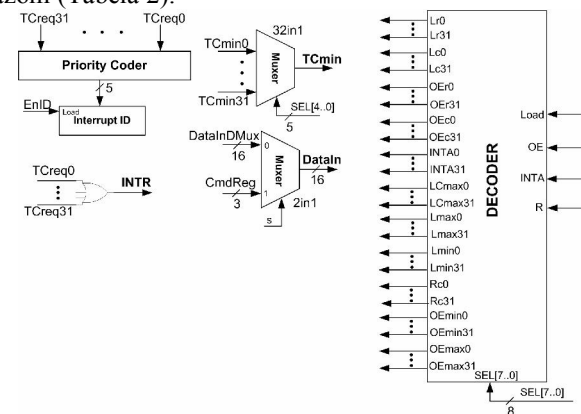
Sl. 4. Deo staze podataka potreban za jedan RT zadatak

Ovi registri su takođe povezani i sa brojačem. Brojač je 16-bitni i broji unazad. Početna vrednost brojača može se postaviti bilo preko DataIn magistrale (setovanjem Lc ulaza) ili preko CmaxReg registra (setovanjem LCmax ulaza). Stanje brojača može se pročitati preko DataOut magistrale (setovanjem OEc ulaza). TC izlaz brojača je u stanju logičke 1 kada brojač, brojeći unazad, odbroji do nule. U tom slučaju nadzorni modul zna da je maksimalno dozvoljeno vreme za izvršenje zadatka isteklo. U svim ostalim stanjima brojača TC izlaz je u stanju logičke 0. Budući da TC izlaz brojača predstavlja S ulaz RS flip-flopa, TC bit kontroliše TCreq izlaz flip-flopa. TCmin izlaz brojača daje informaciju o tome da li je isteklo ili ne minimalno potrebno vreme za korektno izvršenje RT zadatka. Pošto brojač može da vidi sadržaje CmaxReg i CminReg registara, moguće je meriti vreme koje je proteklo od početka izvršavanja zadatka (1) i porediti ga sa minimalnim potrebnim vremenom za njegovo korektno izvršenje.

$$\text{expired_time} = \text{CmaxReg} - \text{counter_state} \quad (1)$$

Sve dok je proteklo vreme manje od minimalno potrebnog vremena za korektno izvršenje zadatka ($\text{expired_time} < \text{CminReg}$), TCmin izlaz brojača je u stanju logičke 1, inače je u stanju logičke 0. Deo staze podataka koji je zajednički za sve RT zadatke koji se nadziru prikazan je na Sl. 5. 8-bitni selektorski ulaz dekodera (SEL) određuje način na koji su ulazi (Load, OE, INTA i R) povezani na izlaze. Load ulazni signal može biti povezan na jedan od 32 različita Lr, Lc, LCmax, Lmax ili Lmin izlaza. OE ulaz se, takođe, može povezati na jedan od 32 različita OEr, OEc, OEmax ili OEmin izlaza. INTA ulazni signal može se povezati na jedan od 32 različita INTAx izlaza, dok je R

ulaz povezan na jedan od 32 različita Rc izlaza. 5 LSB bitova selektorskog SEL ulaza određuju koji će se od 32 različita izlaza povezati sa ulazom. U slučaju Load i OE ulaza, budući da oni mogu biti povezani na nekoliko različitih tipova izlaza, 3 MSB bita SEL signala koriste se da definišu koji će se od različitih tipova izlaza povezati sa ulazom (Tabela 2).



Sl. 5. Deo staze podataka zajednički za sve RT zadatke

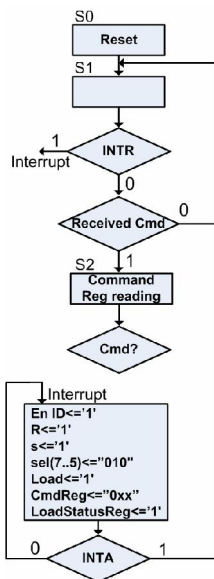
Tabela 2 Način povezivanja Load i OE signala

SEL(7:5) bitovi	Load povezan sa:	OE povezan sa:
000	Lc	OEc
001	LCmax	
010	Lr	OEr
011	Lmax	OEmax
100	Lmin	OEmin

5 LSB bitova SEL signala povezana su takođe i na selektorske ulaze multipleksera 32u1 određujući tako koji će se od 32 različita TCmin ulaza povezati na jedinstveni TCmin izlaz. 32 različita TCreq izlaza iz RS flip-flova povezana su na ulaze prioritet koder. Ovaj koder na svom izlazu daje 5-bitnu identifikaciju zadatka koji je uzrokovao interapt. U slučaju da dva ili više zadatka zahtevaju interapt, prioritet koder će identifikovati zadatak najvišeg prioriteta. 32 TCreq izlaza takođe su povezani na 32 ulaza OR logičkog kola sa INTR izlazom. Tako, ako istekne maksimalno dozvoljeno vreme za korektno izvršenje RT zadatka, TCreq bit se postavlja u stanje logičke 1. Kao posledica toga, INTR izlaz se postavlja u stanje logičke 1 a prioritet koder na svom izlazu daje 5-bitnu identifikaciju RT zadatka koji je uzrokovao interapt. Ovih 5 bitova smeštaju se u Interrupt ID registar.

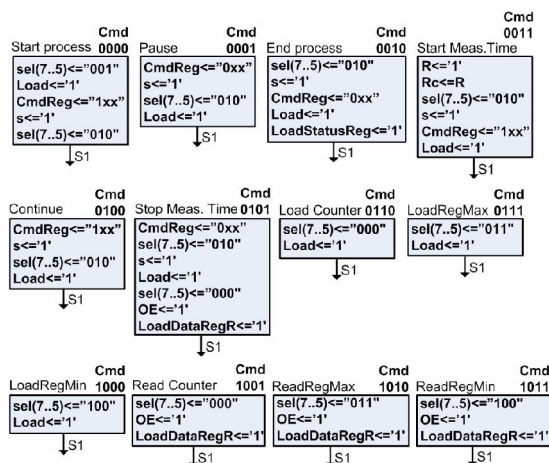
Upravljačka jedinica nadzornog modula: Za upravljačku jedinicu nadzornog modula koristi se automat sa konačnim brojem stanja (Finite State Machine – FSM). Izvor takta FSM automata je 50MHz. Prema tome, FSM automat prelazi iz tekućeg u naredno stanje na svakih 20ns. Prvi deo ASM dijagrama FSM automata prikazan je na Sl. 6. Nakon resetu u s0 stanju, sledeće stanje u koje automat prelazi je s1. U ovom stanju automat ispituje da li se desio interapt. Ako jeste (INTR=1), automat prelazi u Interrupt stanje u kome se vrši obrada interapta. Ako nema interapta (INTR=0), automat čeka na primanje komande. Kad se komanda primi prelazi se u s2 stanje. U ovom stanju automat čita kod komande i, u zavisnosti od primljenog koda, prelazi u neko od narednih stanja. U Interrupt stanju automat resetuje brojač i vrši upis u InterruptID i StatusReg registre. Takođe, upisom 0xx u CmdReg registar, brojaču se zabranjuje brojanje. Iz Interrupt stanja se, nakon detektovanja INTA markera koji postavlja RTS, prelazi u s1 stanje. U zavisnosti od primljene komande, automat prelazi iz stanja s2 u jedno od stanja prikazanih na Sl. 7. U slučaju da je komandni kod 0000, automat prelazi u StartProcess stanje. U brojač se

učitava vrednost iz Cmax registra i brojaču se dozvoljava odbrojavanje. Za komandni kod 0001 automat prelazi u Pause stanje. Brojaču se zabranjuje odbrojavanje upisom 0xx u CmdReg registar. U EndProcess stanju (komandni kod 0010) brojaču se zabranjuje rad a vrednost TCmin bita se smešta u StatusReg registar. Ako je TCmin bit u stanju logičke 1 RTS zna da je zadatak izvršen brže od minimalno potrebnog vremena za njegovo korektno izvršenje. U StartMeasuringTime stanju (0011) automat resetuje brojač i dozvoljava brojaču da počne da odbrojava. Continue stanje je suprotno Pause stanju. U Continue stanju automat brojaču dozvoljava odbrojavanje koje mu je prethodno bilo zabranjeno.



Sl. 6. Prvi deo ASM dijagrama FSM automata

Slično, StopMeasuringTime (0101) stanje je suprotno StartMeasuringTime stanju. U ovom stanju se brojaču zabranjuje odbrojavanje a njegovo trenutno stanje se preko DataOut magistrale smešta u DataRegRead registar. Preko ovog registra ono je na raspolaganju i RTS-u.



Sl. 7. Ostatak ASM dijagrama

U LoadCounter stanju (0110) podaci se preko DataIn magistrale učitavaju u brojač, u LoadRegMax (0111) stanju preko DataIn magistrale u CmaxReg registar, dok se u LoadRegMin stanju (1000) podaci preko DataIn magistrale učitavaju u CminReg registar. Automat u ReadCounter stanju (1001) preko DataOut magistrale smešta stanje brojača u DataRegRead registar. U stanjima ReadRegMax (1010) i ReadRegMin (1011) automat preko DataOut magistrale smešta podatke iz CmaxReg/CminReg registra u DataRegRead registar.

4. FPGA IMPLEMENTACIJA

Svaki deo komunikacionog interfejsa kao i staze podataka i upravljačke jedinice nadzornog modula opisan je

u VHDL programskom jeziku i implementiran na EP2C35F672C6N FPGA čip Alterine DE2 ploče [3]. Rezultati implementacije prikazani su u tabeli 3. Iz tabele se može videti je staza podataka najkritičnije deo FPGA implementacije. Ona zahteva najviše FPGA resursa a takođe i određuje maksimalnu radnu frekvenciju nadzornog modula.

Tabela 3 Rezultati FPGA implementacije

	Total logic elem.	% FPGA	Clock setup
Interface	165/33216	< 1%	241.25MHz
Data-path	7169/33216	22%	78.47MHz
FSM	172/33216	< 1%	280.50MHz
Σ	7506/33216	23%	

Nakon implementacije nadzorni modul je testiran na DE2 razvojnoj ploči. Modulu su komande zadavana sa razvojne ploče a odziv modula praćen je nadgledanjem stanja registara komunikacionog interfejsa. Pošto je nadzorni modul uspešno testiran potrebno je implementirati ga na neku od razvojnih ploča koje na sebi imaju ugrađeni PCI komunikacioni interfejs. Kao jeftino rešenje autori predlažu neku od Ragedstone1 PCI razvojnih ploča [4]. PCI jezgro (core) za komunikaciju može se dodatno naručiti ili naći kao gotov open core modul na [5].

5. PRIMENE NADZORNOG MODULA

Neke moguće primene nadzornog modula do sada nisu pominjane. To je zato što nadzorni modul nije razvijan za neku konkretnu primenu. Namera autora bila je da ga učini pogodnim, sa manje ili više izmena, za nadzor što je više moguće različiti RTS-a. Implementacijom većeg broja brojača-tajmera (do 1000) u FPGA integrisano kolo nadzorni modul postaje pogodan za nadzor veoma velikog broja procesa (na primer internet veza jednog računara-servera).

6. ZAKLJUČAK

Potreba za efikasnim nadzorom RTS-a je očigledna, naročito kod HRTS-a. Sa ciljem da bude što je manje moguće nametljiv i što je više moguće fleksibilan, u radu je predložen jedan hibridni pristup u realizaciji nadzora RTS-a. Budući da je namenjen za PC bazirane RTS-e ovaj nadzorni modul priključuje se na PCI priključak RTS-a koji se nadzire. Nadzorni modul detaljno je opisan, a dati su i rezultati FPGA implementacije.

LITERATURA

- [1] Jane W., Liu S.: "Real-Time Systems", Prentice Hall, 2000.
- [2] Jovanovic B., Jevtic M.: "Module for run-time monitoring in PC hardware based real-time system", Int. Scientific Conference - Unitech, Gabrovo, Bulgaria, 2009.
- [3] Altera DE2 user's manual: http://ftp.altera.com/up/pub/Webdocs/DE2_UserManual.pdf
- [4] Ragedstone1 user's manual: <http://www.enterpoint.co.uk/moelbryn/ragedstone1.html>
- [5] Opencores PCI core: http://www.opencores.org/project.pci32tlite_oc.overview

Abstract – This paper presents one way of FPGA implementation of hybrid (hardware-software based) on-line process monitoring in Real-Time systems (RTS). The reasons for RTS monitoring are presented at the beginning. The summary of different RTS monitoring approaches along with its advantages and drawbacks are also exposed. At the end, monitoring module is described in details. Also, FPGA implementation results and some useful monitoring system applications are mentioned.

FPGA IMPLEMENTATION OF A HYBRID ON-LINE PROCESS MONITORING IN PC BASED REAL-TIME SYSTEMS

Bojan Jovanović, Milun Jevtić