

Projektovanje FPGA mikroprocesorskih sistema korišćenjem PowerPC mikroprocesora i *Embedded Linux* operativnog sistema

Bugarski Milan i Ivan Mezei

Sadržaj — Namenski (*embedded*) mikroprocesorski sistemi mogu se realizovati na FPGA komponentama. FPGA komponente pružaju veliku slobodu u konfigurisanju svih delova sistema i pored toga omogućuju veoma brz razvoj *embedded* sistema. Rad prikazuje mogućnosti projektovanja namenskih mikroprocesorskih sistema na Xilinx-ovoj razvojnoj ploči ML405. Razrađeni su postupci i načini korišćenja softverskih alata za projektovanje sistema. Rad obuhvata nekoliko primera realizacije sistema koji izvršavaju određene samostalne korisničke aplikacije, kao i primer koji opisuje konfigurisanje i generisanje *embedded Linux* operativnog sistema i korišćenje tako podešenog Linux-a na razvojnoj ploči ML405 sa PowerPC mikroprocesorom.

Ključne reči — Mikroprocesorski sistemi, PowerPC, Virtex4 FPGA, Xilinx ML405, *Embedded Linux*.

I. UVOD

FPGA (Field Programmable Gate Array) su poluprovodničke komponente visokog stepena integracije, koje se sastoje od velikog broja programabilnih logičkih blokova i programabilnih konekcija između tih blokova. Korisnik, programiranjem FPGA komponente dobija željeni hardverski sklop, a pošto su ove komponente reprogramabilne, moguće je jednostavno ponovno programiranje, izmena ili dodavanje različitih funkcionalnosti hardvera.

Povećane performanse modernih FPGA kola omogućuju projektovanje celokupnih mikroracunarskih sistema na jednom FPGA čipu. Kompanija Xilinx je jedan od vodećih svetskih proizvođača FPGA komponenti. Konkretno, u ovom radu, korišćena je Xilinx ML405 razvojna ploča, sa Virtex4 čipom. Virtex4 (XC4VFX20-FF672-10) komponenta [1] sadrži tridesetdvobitni mikroprocesor zasnovan na IBM PowerPC arhitekturi [2] koji je realizovan na određenom mestu na silicijumskoj osnovi kao ASIC komponenta i povezan sa programabilnim delom čipa. Periferije ovakvih sistema realizovane su u programabilnoj logici FPGA uređaja. Postoje gotove standardne periferije u vidu IP jezgara koje su optimizovane i testirane ali uglavnom zahtevaju

posedovanje licence. Korisnik može dizajnirati svoje periferije, za šta postoje korisni alati koji u mnogome olakšavaju ove poslove. Najveća prednost *hard-core* mikroprocesora je to što se nalazi u okruženju gde se periferije mogu jednostavno i brzo povezivati i uklanjati sa magistrale, a takođe se može i njihova funkcionalnost brzo menjati i prilagođavati trenutnim potrebama i zahtevima.

Linux operativni sistem se standardno koristi na *embedded* mikroracunarskim sistemima [3]. Implementacijom Linux operativnog sistema, koji je takođe konfigurabilan, omogućena je još veća fleksibilnost namenskih sistema, a razvoj ovih sistema postaje veoma brz i jeftin.

II. PROJEKTOVANJE HARDVERA I SOFTVERA

Za projektovanje sistema korišćena je evaluaciona verzija Xilinx EDK 11.1 (Embedded Development Kit) softverskog paketa [4]. Xilinx EDK je skup alata i IP (Intellectual Property) jezgara koji omogućava dizajniranje celokupnog *embedded* mikroprocesorskog sistema za implementaciju na Xilinx-ovim FPGA komponentama. EDK je deo Xilinx-ovog *Integrated Software Environment (ISE®)* softverskog paketa, koji je neophodan i služi za sintezu i implementaciju dizajna na FPGA uređaj, odnosno generisanje i spuštanje *bitstream* fajla na FPGA [5].

Embedded sistemi mogu da budu veoma kompleksni. Pojedini hardverski i softverski delovi sistema su već projekti sami za sebe, a kad se na to doda i povezivanje delova tako da funkcionišu kao jedinstven sistem i FPGA projektni dizajn, projektovanje celokupnog sistema može da postane vrlo komplikovano. EDK softverski paket omogućava da ovaj proces bude mnogo jednostavniji. Komponente EDK paketa u velikoj meri automatizuju proces projektovanja *embedded* sistema.

Xilinx Platform Studio (XPS) je komponenta EDK paketa koja služi za projektovanje hardverskog sklopa *embedded* mikroprocesorskih sistema baziranih na PowerPC ili MicroBlaze procesorima. Pri pokretanju XPS komponente automatski se pokreće *Base System Builder (BSB) wizard*, grafički interfejs u kojem se na jednostavan način konfigurišu parametri vezani za hardver, biraju potrebne periferije i ostale stvari vezane za budući mikroprocesorski sistem. Neki sistemi mogu biti kompletirani korišćenjem samo ovog *Wizard*-a. Ova početna podešavanja se mogu kasnije izmeniti u glavnom projektnom prozoru XPS-a.

Još jedan veoma koristan alat je *Create and Import*

Istraživanje u okviru ovog rada delimično je finansirano od strane Ministarstva za nauku i tehnološki razvoj u okviru projekta „Razvoj sistema i instrumenata za istraživanje vode, nafte i gasa“, Ev.Br. 11006.

Milan Bugarski, Fakultet Tehničkih Nauka u Novom Sadu, Srbija (telefon: 381-64-0151103; e-mail: bugarskimilan@gmail.com).

Ivan Mezei, Fakultet tehničkih nauka, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija; (e-mail: imezei@uns.ac.rs).

Peripheral (CIP) Wizard, koji pomaže korisniku pri kreiranju sopstvenih periferija i pri integrisanju tih periferija u projekat. Ovaj alat potpuno automatizuje proces kreiranja periferije, i omogućava korisniku da se bavi samo konkretnom funkcionalnošću projektovane komponente bez potrebe za detaljnim informacijama o interfejsu sa magistralom i drugim potrebnim detaljima.

Za projektovanje softvera koristi se Software Development Kit (SDK). SDK generiše softversku platformu koja predstavlja skup biblioteka, drajvera i po potrebi operativnih sistema i kernela, na čijem temelju korisnik implementira svoju softversku aplikaciju. Ovako generisana softverska platforma sadrži samo neophodne biblioteke koje koristi korisnička aplikacija. Softverski deo *embedded* sistema može biti realizovan kao *standalone* aplikacija koja se izvršava na mikroprocesoru, a može biti zasnovan i na *embedded* Linux operativnom sistemu gde se aplikacije izvršavaju preko korisničkog interfejsa tog operativnog sistema. U ovom radu su prikazane obe varijante projektovanja sistema.

U sklopu probne verzije EDK su takođe i brojna IP jezgra, softverske biblioteke, drajveri, kao i C/C++ cross-compiler za programe koji treba da se izvršavaju na PowePC arhitekturi.

EDK obezbeđuje i alate potrebne za simulaciju i verifikaciju celokupnog mikroprocesorskog sistema. Simulacija sistema je efektivan metod za proveru kompatibilnosti softvera sa hardverskom platformom. EDK projekti se simuliraju korišćenjem ModelSim simulatora. Prednost ovakve simulacije je mogućnost praćenja rada sistema (uključujući i praćenje unutrašnjih signala), bez potrebe za vezom sa hardverom. Proces verifikacije je ubrzan zbog toga što nije potrebno ponovo pokretati proces implementacije (kao u slučaju verifikacije na hardveru), da bi stekli uvid u efekte promene koda. Stimulacione signale za simulaciju obezbeđuje C kod koji se izvršava na procesoru. Ovaj kod može biti izvršavan iz BRAM ili DDR memorije. Simulacija koda iz DDR memorije je dosta kompleksnija ali dozvoljava veće programe. Simulacija kompletnog sistema je rađena za prvi primer od tri primera sistema sa *standalone* softverom, a kao simulator korišćena je probna verzija ModelSim 6.5 SE.

Osnovni tok projektovanja *embedded* sistema uz pomoć alata iz EDK paketa prikazan je na slici 1.

III. PRIMERI PROJEKTOVANJA SISTEMA SA *STANDALLONE* APLIKACIJAMA

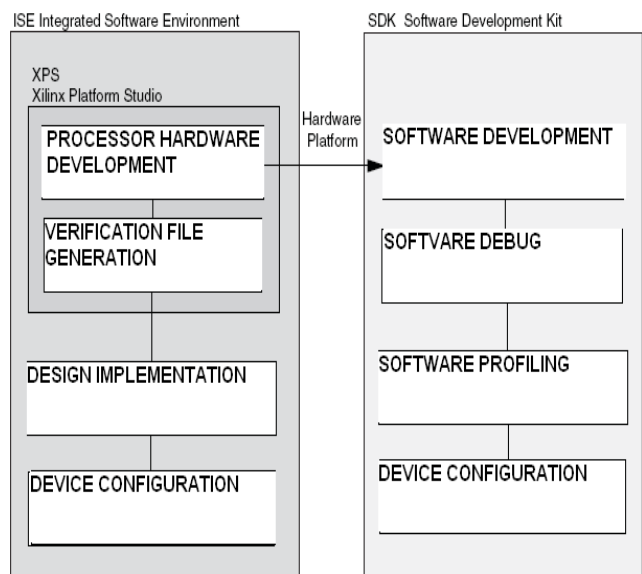
Za urađene primere korišćene su minimalne potrebne hardverske konfiguracije, tako da se za različite primere hardverski sistemi razlikuju po korišćenim periferijama. Zajedničko za sve je da se koristi PowerPC 405 mikroprocesor na koji se periferijske komponente vezane preko Processor Local Buss (PLB) magistrale.

Prvi primer demonstrira korišćenje GPIO (General Purpose Input Output) periferije na koju su povezani eksterni tasteri i led diode sa razvojne ploče. GPIO je 32-

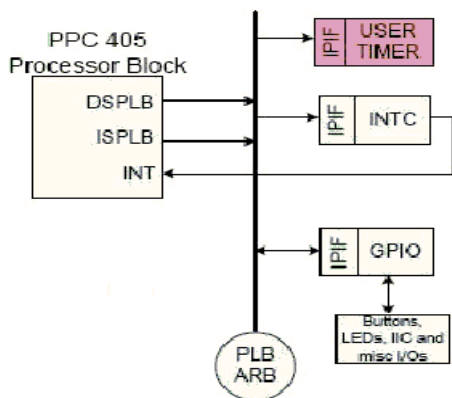
bitna periferija koja se vezuje na PLB preko IPIF (*intellectual property interface*) interfejsa. Može se konfigurisati kao jednokanalna ili dvokanalna periferija i svaki bit se može dinamički podesiti kao ulazni ili kao izlazni signal. Blok šema hardverskog sklopa sistema potrebnog za ovaj primer, prikazana je na slici 2, s tim što nisu korišćene tajmerska i prekidna periferija.

Drugi primer opisuje načine korišćenja kontrolera prekida(XPS INTC), konfigurisanje i povezivanje eksternog signala prekida sa procesorom. Kao periferija je korišćena standardna GPIO komponenta, povezana na spoljašnji taster na ploči, sa mogućnošću generisanja prekidnog signala na svaku promenu stanja tastera. Taj prekidni signal se dovodi na ulaz XPS INTC periferije koja obrađuje ovaj signal i šalje zahtev za prekidom direktno procesoru. Uloga kontrolera prekida je da sakuplja prekidne signale iz različitih izvora, da ih onda uređuje i analizira njihove prioritete i da na izlazu generiše jedinstveni prekidni signal koji šalje procesoru. XPS INTC periferija omogućava povećanje broja prekidnih signala ka procesoru i po potrebi obezbeđuje sistem prioriteta za prekidne signale. Blok šema za ovaj primer je na slici 2. ali bez tajmerske periferije.

Treći primer predstavlja demonstraciju rada i način korišćenja brojača odnosno tajmera. Tajmer se uključuje u sistem u vidu standardne periferije *XPS Timer*. Ovo je 32-bitni tajmer koji se vezuje na PLB magistralu. Brojač je povezan i na XPS INTC periferiju i u korisničkom programu konfigurisan za generisanje prekida i za *auto-reload* način rada. Ovo znači da će tajmer po isteku zadatog vremenskog intervala, generisati prekidni signal prema kontroleru prekida i da će istovremeno učitati u odgovarajući registar zadatu vrednost i ponovo početi da odbrojava. U ovom i u prethodnom primeru se koriste korisničke led diode na ploči kao jednostavna signalizacija da se prekid desio i da se prekidna rutina u korisničkoj aplikaciji izvršila. Ilustracija sistema je na slici 2.



Sl.1. Osnovni tok projektovanja sistema



Sl.2. Blok šema sistema korišćenog u primerima.

IV. PROJEKTOVANJE LINUX EMBEDDED SISTEMA

Počev od svog prvog javnog predstavljanja 1991. godine, Linux operativni sistem je široko primenjivan u raznim sistemima. Inicijalno ograničen na slabo povezane grupe projekatana i entuzijasta na internetu, Linuks je vremenom sazreo u solidan Unix-bazirani operativni sistem za radne stanice, servere itd. Zbog svoje fleksibilnosti, robusnosti i cene, Linux je u poziciji da postane veoma korišćen u *embeded* mikror računarskim sistemima.

Embeded Linux se obično odnosi na konfigurisani Linux kernel koji je specijalno podešen za odgovarajući hardver, mada se *embeded* sistemi mogu projektovati i sa nekom zvaničnom verzijom kernela.

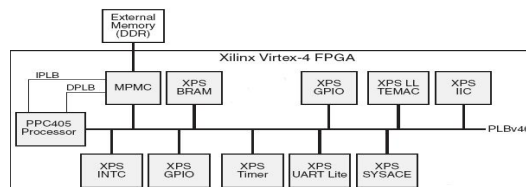
Hardverska konfiguracija sistema na kojoj je pokrenut Linux, sastoji se od standardnih komponenti i perifera na ML405 razvojnoj ploči. Hardverska platforma sadrži PowerPC 405 mikroprocesor, 128MB DDR_SDRAM memorije, IIC EEPROM, TriMode_MAC_GMII, SYSACE_CompactFlash, XPS 16550 UART, xps_timer, xps_intc_0, xps_bram_if_cntlr_1, Push_Buttons_Position i LEDs_4Bit [6].

PLBv4.6 magistrala i DDR memorija rade na 100MHz, a procesor je podešen na 300MHz. Pristup DDR memoriji je omogućen preko Multi-Port Memory Controller (MPMC) perifera vezane na PLB.

XPS_LL_TEMAC periferija obezbeđuje dodatnu funkcionalnost i olakšava upotrebu Hard TEMAC komponente koja je ugrađena u Virtex4 čip i još u neke druge FPGA uređaje, a istovremeno obezbeđuje soft Ethernet MAC opciju svim ostalim FPGA uređajima. TEMAC je akronim za *Tri-Mode Ethernet Media Access Controller*, i obezbeđuje sistemu Ethernet interfejs sa okruženjem.

XPS IIC jezgro je uključeno u projekat jer se u EEPROM memoriji čuva jedinstvena Ethernet MAC adresa, a FPGA pristupa ovoj memoriji preko IIC magistrale. Ovim je onda omogućeno Linux-u da pročita Ethernet MAC adresu.

Blok dijagram za projektovani PowerPC referentni sistem prikazan je na slici 3.



Sl.3. Blok dijagram PowerPC referentnog sistema.

XPS 16550 UART periferija obavlja paralelno-serijsku konverziju karaktera primljenih od procesora i serijsko-paralelnu konverziju karaktera primljenih sa modema ili neke perifera. UART je skraćeno od Universal Asynchronous Receiver/Transmitter. U projektu omogućava komunikaciju *embeded* sistema sa računarom preko hiperterminala.

V. GENERISANJE I IMPLEMENTACIJA LINUX OPERATIVNOG SISTEMA

Najzahtevniji deo posla pri projektovanju *embeded* Linux sistema je generisanje i prevođenje Linux operativnog sistema koji će na kraju biti pokrenut na datom *target* uređaju.

Pravljenje izvršnog (elf) fajla sa Linuxom može biti dosta naporan posao jer ne postoji tačno definisan i razrađen postupak a takođe se uputstva i objašnjenja različitih autora ne poklapaju.

Prepoznavši potencijal Linux sistema u *embeded* uređajima kao i sve veću prisutnost ovih uređaja na tržištu, mnoge kompanije su predstavile komercijalne alate i aplikacije za razvoj *embeded* Linux operativnog sistema. Ovi alati automatizuju proces projektovanja. Najpoznatiji komercijalni razvojni paket je *Monta Vista Linux*, koji pored alata za brz razvoj sistema sadrži i gotove *embeded* Linux distribucije za određene hardverske platforme.

Postoji i veliki broj *open source* [7] rešenja koji omogućavaju razvoj Linux-a na određenim hardverskim platformama.

U radu, kao alat za razvoj *embeded* Linux-a, korišćen je besplatni *Denx ELDK* paket za PowerPC 4xx procesore. Generisanje izvršnog fajla sa Linux-om izvedeno je u nekoliko koraka opisanih u daljem tekstu.

Prvo je potrebno u XPS-u generisati DTS (*device tree*) fajl koji sadrži informacije o hardverskoj platformi, odnosno koje su perifera i na koji način upotrebljene, koje su biblioteke i drajveri kompatibilni sa hardverom itd. Za ovo je potrebno kopirati *device tree generator* [8] u projektni folder XPS-a i posle podešavanja softverske platforme izabrati opciju „Generate Libraries and BSPs“ iz menija.

Operativni sistem i aplikacije za projektovani hardverski, *target*, sistem na razvojnoj ploči, razvijaju se na *host* PC računaru koji je baziran na intel i386 arhitekturi sa Ubuntu 9.04 Linux distribucijom kao operativnim sistemom. Ploča je povezana JTAG vezom na *host* mašinu, uz pomoć Xilinx-ovog *Platform Cable USB* uređaja. Pošto su arhitekture *host* i *target* sistema različite, da bi se softver razvijen na PC računaru mogao izvršavati na *target* uređaju potrebno je taj kod prevesti u odgovarajući oblik prepoznatljiv mikroprocesoru *target*

uređaja. To se postiže pomoću *cross-compiler* alata koji se nalazi u sklopu Denx ELDK [9] paketa.

Korišćen je Xilinx-ov Linux kernel [8] koji je posebno podešen tako da pruža podršku za veliki broj Xilinx-ovih IP jezgara, odnosno za njihove drajvere a takođe i za mnoge referentne razvojne sisteme. Ova verzija kernela (2.6.30) omogućuje pristup i upravljanje uređajima u FPGA čipu, kao što su na primer Ethernet kontroler i sve ostale korisničke periferije.

Sledeći korak je podešavanje kernela, odnosno definisanje odgovarajućeg konfiguracionog fajla koji kompajler poziva pri prevođenju kernela. U sastavu Xilinx Linux kernela postoje gotovi *.conf* fajlovi za neke referentne uređaje. Ovi fajlovi se mogu koristiti u projektovanju i mogu biti dobra osnova za konkretno i detaljno podešavanje kernela koje je uvek različito za različite hardverske platforme.

Da bi Linux operativni sistem mogao da se pokrene i da funkcioniše na datom hardveru, između ostalog potrebno je uključiti i *root filesystem* koji sadrži fajlove i funkcije neophodne za pokretanje operativnog sistema tako da se korisnik može ulogovati i koristiti njegove pogodnosti. Fajl sistem za *embedded* Linux sisteme obično je zasnovan na *BusyBox* [10] aplikaciji koja obezbeđuje veliki broj standardnih funkcija Linux-a. Fajl sistem može biti implementiran pomoću reznih pomoćnih softverskih alata kao što je na primer *BuildRoot* [11] koji pretstavlja skup fajlova i zakrpa koji olakšavaju generisanje fajl sistema. Za ovaj kotrak potrebno je dosta informacija o samom Linux operativnom sistemu da bi se mogla izvršiti razna potrebna podešavanja.

Za potrebe ovog projekta korišćen je funkcionalni fajl sistem u vidu ramdiska koji se nalazi u ELDK paketu. Ovaj konkretan ramdisk je zasnovan na *BusyBox* aplikaciji i naknadno je podešen. Izmene su načinjene na konfigurisanju ftp serverske aplikacije, u cilju omogućavanja povezivanja *host* računara na *embedded* Linux sistem *target* uređaja. Ovakva konfiguracija je praktična jer se onda korisničke aplikacije, razvijene i kompajlirane na PC-u, mogu jednostavno prebaciti na *target* fajl sistem preko ftp protokola i potom na njemu izvršavati.

Sledeće što treba uraditi je smeštanje prethodno generisanog DTS fajla i ramdiska u odgovarajuće direktorijume unutar Linux kernela, i pokretanje procesa kompajliranja kernela. Ali pre pokretanja samog procesa potrebno je još podesiti okruženje i učitati konfiguracioni fajl da bi *cross-compiler* imao pristup svim potrebnim fajlovima i funkcijama za postupak prevođenja. Kao konačni rezultat dobijen je izvršni *elf* fajl čijim se izvršavanjem na projektovanoj hardverskoj platformi pokreće *embedded* Linux operativni sistem i učitava inicijalni ramdisk fajl sistem.

Kao primer aplikacija koje se izvršavaju iz korisničkog okruženja ovako generisanog Linux-a, napisana su dva programa čija je funkcija pristupanje u upravljanje GPIO periferijama uz pomoć prisutnih Linux drajvera za ove periferije.

Eksperimenti sa *embedded* Linux-om dati su u [12], gde

je takođe realizovan jedan namenski FPGA SoC sistem na sličnoj hardverskoj platformi. Nekoliko primera u vezi konfigurisanja *embedded* Linux operativnog sistema, mogu se naći i u [13] i [14].

ZAKLJUČAK

Rezultati rada prikazuju i definišu postupke i procedure kao i uobčajene tokove projektovanja namenskih mikroročunarskih sistema zasnovanih na PowerPC arhitekturi i implementiranih na FPGA uređajima. Rezultati mogu biti korisni studentima koji nameravaju da se bave razvojem ovakvih sistema, jer urađeni praktični primeri demonstriraju i sistematizuju standardne tehnike projektovanja mikroprocesorskih sistema na Xilinx ML405 razvojnoj ploči zajedno sa postupcima podizanja *embedded* Linux-a na referentnom hardveru. Prema tome, osnovna svrha rada je da se na jednom mestu definišu postupci za realizaciju FPGA SoC mikroročunarskih sistema i može se smatrati da je edukativnog karaktera.

LITERATURA

- [1] ML405 Evaluation Platform, *User Guide*, Xilinx UG210 (v1.5.1) March 10, 2008.
- [2] PowerPC Processor Reference Guide, Xilinx UG011 (v1.2) January 19, 2007.
- [3] Karim Yaghmour, "Building Embedded Linux Systems", O'Reilly April 2003.
- [4] Embedded System Tools Reference Manual *Embedded Development Kit, EDK 11.1*, Xilinx 2009.
- [5] EDK Concepts, Tools, and Techniques A Hands-On Guide to Effective Embedded System Design, Xilinx 2009.
- [6] Srikanth Vemula, "Getting Started with EDK and Linux 2.6", Xilinx XAPP969 (v1.1) February 23, 2007.
- [7] Xilinx Open Source Wiki, <http://xilinx.wikidot.com/>
- [8] Xilinx Git site, <http://git.xilinx.com/cgi-bin/gitweb.cgi>, September 2009.
- [9] Denx ftp directory, http://ftp.denx.de/pub/eldk/4.2/ppc-linux-x86/iso/ppc-2008-04-01_iso, September 2009.
- [10] *BusyBox*, www.busybox.net
- [11] Buildroot, <http://buildroot.uclibc.org/>
- [12] Dragomir El Mezeni, Milan Novaković, "Embedded FPGA Linux sistemi", 16. Telekomunikacioni forum TELFOR 2008.
- [13] Linux for the Xilinx Virtex4/5 FPGAs, V Pucknell; STFC; Daresbury Lab, Jan 15. 2009. <http://npg.dl.ac.uk/MIDAS/DataAcq/EmbeddedLinux.html>
- [14] Wolfgang Klingauf and Uwe Klingauf "Virtex2Pro & Linux" <http://www.klingauf.de/v2p/index.phtml>

ABSTRACT

The wide application of embedded systems becomes a trend in the post-PC era. Embedded systems have been deployed in numerous fields which have different requirements of embedded systems architecture. Modern FPGA devices provide very flexible and rapid development flows for such microprocessor systems. This project describes, through several examples, ways to build PowerPC based systems on Xilinx ML405 evaluation board.

DESIGNING FPGA MICROPROCESSOR SYSTEMS BASED ON POWERPC MICROPROCESSOR AND LINUX OPERATING SYSTEM

Bugarski Milan i Ivan Mezei