

Zahtjevi IEEE Simulator Design natjecanja

12. veljače 2011.

1 Zahtjevi od strane IEEE-a

Potpuni opis natjecanja naveden je na stranici:

- <http://www.computer.org/portal/web/competition>

uz dosta koristan FAQ:

- <http://www.computer.org/portal/web/competition/faqs> (obavezno pročitati!).

Zahtjevi su dosta labavi i mogu se sažeti u “*Napravite simulator mikroprocesora za pomoć u nastavi iz Arhitekture računala.*” Zahtjevi po navodima:

- izraditi simulator,
- ISA može biti uzet ili razvijen od nule,
- mora se moći simulirati i CISC i RISC arhitektura (kod CISC-a dužina riječi može biti fiksna – 32b),
- moramo napisati par test primjera (C kod → naš assembler),
- moramo provesti beta testiranje (za ovo moramo zamoliti druge kolege).

Svaki slučaj biranja (npr. odabir skupa instrukcija, odabir veličine instrukcija i sl.) moramo opravdati u dokumentaciji.

1.1 Komponente bodovanja

Boduju se tri komponente:

- kvaliteta programskog rješenja,
- ISA (skup instrukcija),
- mogućnosti (opseg) simulatora.

Težine komponenti slijede gornji raspored. Njihova ideja je da poštujemo pravila dobrog programskog inženjerstva, možda da izradimo inovativan skup instrukcija ili novi asemblerski zapis (dan je primjer, umjesto “ADD R0, R1, R2” iskoristiti zapis “R0 ? R1 + R2”), te da napravimo simulator koji će biti primjenjiv kao pomoć pri nastavi iz predmeta Arhitektura računala. U prvu komponentu, osim same izvedbe ubrajaju se dokumentacija i izvedba evaluacije.

1.2 Izvještaj

Izvještaj, odnosno “*Final report*” se sastoji od sljedećih cjelina:

1. Introduction,
2. Design of the architecture and its fitness for purpose,
3. Design of the Simulator,
4. Use of software engineering in creating the product,
5. Testing the product,
6. Your response to the testing and suggestions for future improvement,
7. Appendices.

NAPUTAK: “The final report may consist of no more than 50 pages ($\frac{3}{2}$ line spacing).”

2 Naši zahtjevi, želje i ideje

- mora moći simulirati,
- prikaz rezultata (snapshot):
 - stanje registara,
 - stanje memorije,
 - stanje pipelinea,
 - stanje vanjskih jedinica.
- mogućnost spajanja vanjskih jedinica,
- mogućnost konfiguriranja vanjskih jedinica,
- mogućnost debugiranja,
- *forward* i *backward* debugiranje,
- lijep GUI :) – SWT; editor, prikaz stanja, itd.
- GUI za vanjske jedinice,

- mogućnost konfiguriranja okoline (stanje status registra, memorije, registra, itd.),
- podešavanje kašnjenja simulacije (usporavanje izvođenja simulacije),
- odabir točki praćenja kod debugiranja – trace points,
- IDEJA: $C \rightarrow \text{LLVM intermediate} \rightarrow \text{naš assembler!}$ Ukratko, napraviti prevoditelj iz LLVM koda u naš assembler (FRISC ili nešto slično), tako da se praktički C kod može simulirati :).