

Elektrotehnički fakultet
Univerzitet u Beogradu

SIMULATOR PROCESORSKOG SISTEMA ORIJENTISAN NA OPTIMIZACIJU POTROŠNJE I PERFORMANSI

Projekat: **Razvoj i modelovanje energetske efikasne, adaptibilne, višeprocorskih i višesenzorskih elektronskih sistema male snage**

Oznaka projekta: **TR32043**
Rukovodilac projekta: **Goran Dimić**

Vrsta dokumenta: **Tehnička dokumentacija projekta**
Stepen poverljivosti: **poverljivo – interno**

Odgovorno lice:

Strahinja Janković
Elektrotehnički fakultet Univerziteta u Beogradu
Tel:
E-mail: jankovics@el.etf.rs

Realizator:

Strahinja Janković, Dragomir El Mezeni, Ivan Popović, Lazar Saranovac

SADRŽAJ:

1. KRATAK OPIS TEHNIČKOG REŠENJA.....	3
2. STANJE U OBLASTI.....	4
3. DETALJNO TEHNIČKO REŠENJE	5
3.1. Uvod	5
3.2. Model procesora.....	5
3.2.1. Režimi rada procesora	5
3.2.2. Opterećenje procesora	6
3.2.3. Model potrošnje	6
3.3. Model programskih poslova	7
3.3.1. Raspoređivanje.....	7
3.3.2. Sinhronizacija	8
3.4. Procena potrošnje.....	8
4. VERIFIKACIJA.....	9
4.1. Analiza rada simulatora.....	9
4.2. Procena potrošnje modelovanog procesora	10

1. KRATAK OPIS TEHNIČKOG REŠENJA

Primena od (dd.mm.gggg): 25.11.2014. godine
Godina* : 2014
Odgovorno lice : Strahinja Janković

Opis: Implementiran je simulator procesorskog sistema za potrebe analize potrošnje i mogućnosti primene algoritama za optimizaciju potrošnje i performansi. Simulator je implementiran u Python-u korišćenjem SimPy modula za simulaciju diskretnih događaja. Procesorski sistem je vođen događajima. Realizovan je model procesora koji radi na zadatoj učestanosti i podržava više radnih režima sa određenim profilima potrošnje. Podržan je prioritetni algoritam raspoređivanja. Moguće je podešavati učestanost procesora u toku rada, što utiče i na potrošnju i na performanse procesora. U zavisnosti od seta taskova koji se izvršavaju menja se i opterećenje procesora. Na osnovu aktivnosti procesora i definisanih modela potrošnje za dati procesor moguće je odrediti ukupnu potrošnju, kao i potrošnju u segmentima od značaja. Definisanje novih modela procesora i novih modela potrošnje procesora je jednostavno, čime se može proširiti primena realizovanog simulatora. Verifikovana je funkcionalnost simulatora na modelu procesora koji podržava više radnih učestanosti i režima smanjene potrošnje. Korišćenjem realizovanog simulatora omogućen je razvoj algoritama za energetski efikasno raspoređivanje posla na procesorskim sistemima.

Realizatori : Strahinja Janković, Dragomir El Mezeni, Ivan Popović, Lazar Saranovac

Korisnici : Laboratorija za Integrisane računarske sisteme na Elektrotehničkom fakultetu u Beogradu

Podtip rešenja : Softver (M85)

2. STANJE U OBLASTI

Performanse digitalnih sistema direktno zavise od učestanosti na kojoj taj sistem radi. Međutim, povećanjem učestanosti raste i potrošnja sistema. Da bi se optimizovala potrošnja procesora, potrebno je meriti potrošnju prilikom izvršavanja različitih programskih poslova. U zavisnosti od karakteristika procesora kao i željenih karakteristika merenja, merenje potrošnje procesora može biti zahtevan posao. Takođe, za analizu određenih algoritama nije bitna tačna vrednost potrošnje, već trend. Na osnovu određenih merenja moguće je kreirati pojednostavljeni model potrošnje procesora, koji se može dalje koristiti u simuliranom okruženju. Tako se može ubrzati inicijalni razvoj određenih optimizacionih algoritama, jer se fokus stavlja na problem koji se rešava.

Razvijeno je više algoritama za raspoređivanje poslova u procesorskim sistemima. Međutim, cilj optimizacije tih algoritama je povećanje performansi sistema. Sa druge strane, u nekim situacijama je važno maksimalno iskoristiti dostupnu energiju, pogotovu kada se radi o mobilnim uređajima. Problem koji se javlja je kako proceniti potrošnju sistema tako da to može da se koristi kao parametar prilikom optimizacije.

Dostupno je više simulatora procesorskih sistema. Postojeći simulatori su uglavnom bazirani ili na što tačnijoj simulaciji instrukcijskog seta procesora, ili na implementaciji dovoljno funkcionalnosti tako da je moguće testiranje i razvoj algoritama za optimizaciju performansi. Podatak o potrošnji procesorskog sistema prilikom izvršavanja nedostaje u obe varijante.

Realizovani simulator omogućava simulaciju procesorskog sistema sa posebnim fokusom na potrošnju i performanse. Na taj način, omogućen je jednostavan razvoj i testiranje algoritama za optimizaciju.

3. DETALJNO TEHNIČKO REŠENJE

3.1. Uvod

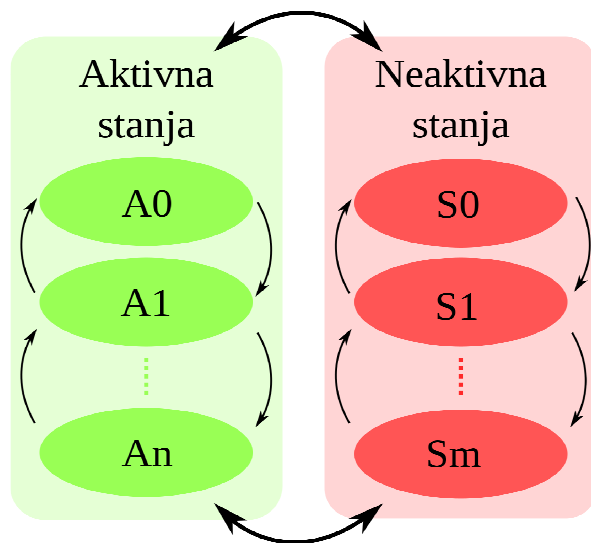
U ovom rešenju predstavljen je simulator procesorskog sistema. Modelovan je procesor sa setom programskih poslova koje obavlja. Procesorski sistem je vođen događajima. Simulator je realizovan u programskom jeziku Python pomoću modula Simpy za simulaciju diskretnih događaja.

3.2. Model procesora

Modelovan je procesor sa setom parametara. Procesor podržava više režima rada kao i režime smanjene potrošnje. Za svaki od režima rada definisani su modeli potrošnje, na osnovu kojih se računa ukupna potrošnja. U različitim režimima rada različita je radna učestanost procesora. U zavisnosti od učestanosti menjaju se performanse i potrošnja. Na osnovu programskih poslova koji se izvršavaju na procesoru menja se i trenutno opterećenje procesora.

3.2.1. Režimi rada procesora

Modelovani procesor podržava više režima rada. Podržani režimi (stanja) prikazani su na slici 1.



Slika 1 – Stanja modelovanog procesora

Modelovani procesor može da se nalazi u jednom od aktivnih ili neaktivnih stanja. Aktivna stanja su stanja $A_0..A_n$, dok su neaktivna stanja $S_0..S_n$.

Radna učestanost procesora je vezana za režim rada (stanje). Stanje A_0 je stanje u kojem procesor ima najbolje performanse (najveću učestanost), ali i najveću potrošnju. Stanje A_1 je stanje u kojem procesor ima lošije performanse (nižu učestanost) od stanja A_0 ali bolje performanse veću učestanost) od A_2 (samim tim i manju potrošnju od A_0 , ali veću od A_2). Stanje A_n je stanje u kojem procesor ima lošije performanse (nižu učestanost), ali i manju potrošnju od stanja A_{n-1} .

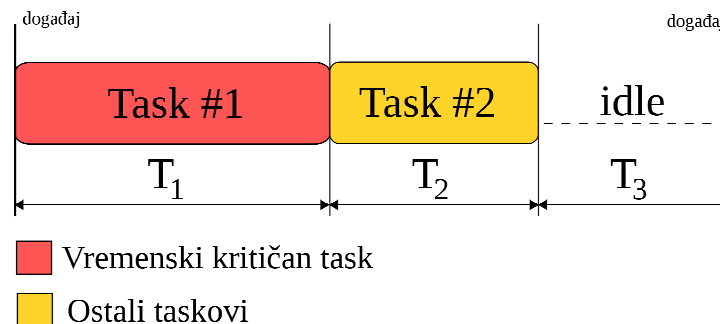
Za svako od neaktivnih stanja (režima smanjene potrošnje), pored potrošnje u tom režimu definisano je i vreme potrebno da procesor pređe iz tog režima u neki od aktivnih režima. Stanje S_0 je stanje u kojem procesor ima najveću potrošnju od svih neaktivnih stanja, ali i najkraće vreme buđenja. Stanje S_1 je stanje u kojem procesor ima manju potrošnju i veće vreme buđenja od stanja S_0 , odnosno veću potrošnju i manje vreme buđenja od stanja S_2 . Stanje S_m je stanje u kojem procesor ima manju potrošnju, ali i veće vreme buđenja od stanja S_{m-1} .

3.2.2. Opterećenje procesora

Na opterećenje procesora utiču taskovi koji se izvršavaju kao i vreme boravka u režimu smanjene potrošnje. Moguće je definisati taskove od interesa i onda računati opterećenje u odnosu na njih. Ako se posmatra programski posao u sistemu vođenom događajima dat na slici 2, onda se dobija da je opterećenje procesora u odnosu na kritičan posao dato formulom

$$DR = \frac{T_1}{T_1 + T_2 + T_3}$$

Dakle, na osnovu trajanja vremenski kritičnog posla i ukupnog vremena između dva uzastopna događaja moguće je definisati opterećenje procesora u odnosu na kritičan posao, što se može koristiti kao parametar za dalju optimizaciju.



Slika 2 – Izgled programskog posla u sistemu vođenom događajima

3.2.3. Model potrošnje

U zavisnosti od stanja u kojem se nalazi procesor, definisana je određena potrošnja. Taj podatak se može koristiti kao parametar za optimizaciju, a moguće je i grafički prikazati uticaj određenih parametara na potrošnju. Potrošnja u određenom stanju zadaje se kao srednja snaga koja se utroši u datom stanju. Za neaktivna stanja, pored potrošnje u tom stanju potrebno je zadati i vreme potrebno za buđenje procesora iz datog stanja.

Model potrošnje se zadaje u formatu prikazanom na slici 3. Odabran je JSON format, zbog jednostavnosti.

```
[
  {
    "state":
      [
        {
          "id": "A0",
          "name": "active0",
          "power": 100
        },
        {
          "id": "A1",
          "name": "active1",
          "power": 70
        },
        ...
        {
          "id": "S0",
          "name": "sleep0",
          "power": 5,
          "latency": 1
        },
        {
          "id": "S1",
          "name": "sleep1",
          "power": 1,
          "latency": 10
        }
      ]
  }
]
```

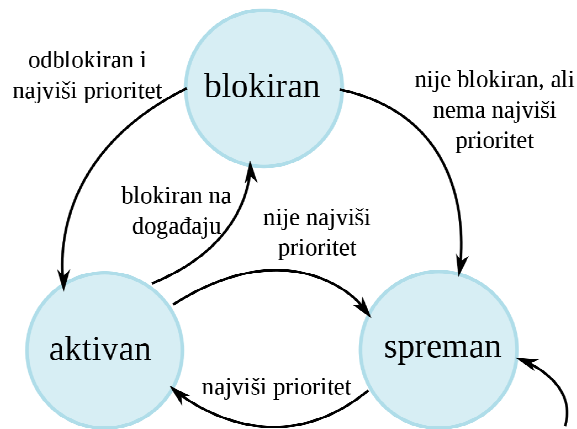
Slika 3 – Primer modela potrošnje procesora

3.3. Model programskih poslova

Na modelovanom procesoru izvršavaju se određeni programski poslovi. Implementiran je algoritam prioritnog raspoređivanja. Omogućena je sinhronizacija programskih poslova preko liste događaja. Programski poslovi su modelovani kao intervali vremena u kojima je procesor zauzet i nalazi se u nekom od aktivnih stanja.

3.3.1. Raspoređivanje

Implementirano je prioritno raspoređivanje programskih poslova. Taskovi imaju definisane prioritete od 0 do 4, gde je 0 najviši a 4 najniži prioritet. Ukoliko u toku izvršavanja nekog taska, task sa višim prioritetom postane spreman za izvršavanje, task višeg prioriteta prelazi u stanje izvršavanja, a task koji se do tada izvršavao prelazi u stanje čekanja i čeka da se procesor oslobodi. U slučaju da su taskovi istog prioriteta, novi task čeka završetak izvršavanja prethodnog taska. Mašina stanja taskova prikazana je na slici 4.



Slika 4 – Mašina stanja simuliranih taskova

3.3.2. Sinhronizacija

Da bi se omogućila sinhronizacija između taskova, implementirana je lista događaja. Taskovi mogu da se blokiraju na nekom od događaja. Svi događaji su dostupni svim taskovima.

Nakon što se pojavi događaj, task koji je čekao pojavu tog događaja postaje spreman za izvršavanje. U zavisnosti od prioriteta taska odlučuje se da li odmah kreće da se izvršava, ili čeka da procesor postane slobodan.

3.4. Procena potrošnje

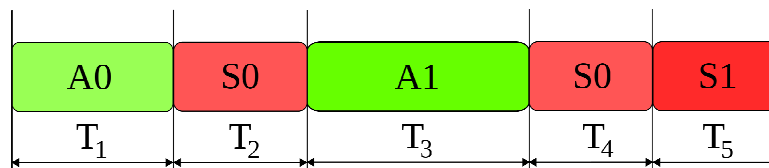
Procena potrošnje se vrši na osnovu modela potrošnje i aktivnosti procesora. Na osnovu srednje potrošnje koja se zadaje kroz model potrošnje i vremena provedenog u datom stanju moguće je odrediti ukupnu utrošenu energiju, kao i srednju potrošnju snage u intervalu od interesa.

Dakle, za aktivnost procesora prikazanu na slici 5, ukupna potrošena energija bi bila

$$E_{tot} = P_{A0}T_1 + P_{S0}T_2 + P_{A1}T_3 + P_{S0}T_4 + P_{S1}T_5$$

odnosno ukupna potrošena snaga

$$P_{tot} = \frac{E_{tot}}{\sum_{i=1}^5 T_i}$$



Slika 5 – Primer aktivnosti procesora

4. VERIFIKACIJA

Simulator je testiran na primeru procesora koji izvršava određeni set taskova. Procesor ima 3 aktivna stanja i 1 neaktivno stanje. Potrošnja u pojedinim stanjima prikazana je u tabeli 1.

Tabela 1 – Stanja modelovanog procesora

Aktivna stanja			Neaktivna stanja		
Stanje	Učestanost [MHz]	Potrošnja [mW]	Stanje	Potrošnja [mW]	Vreme buđenja [μ s]
A0	3	100	S0	5	0
A1	2	70			
A2	1	30			

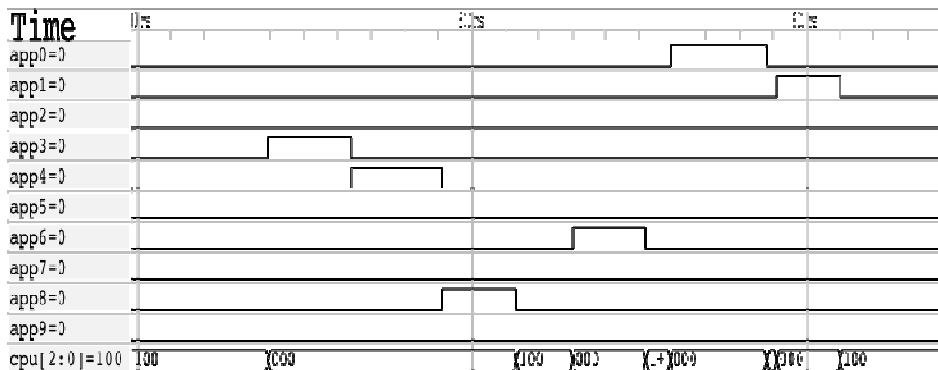
Pokrenuto je 10 taskova. Vreme aktiviranja i trajanja taskova je nasumično odabrano. Taskovi 0 i 1 imaju prioritet 0, taskovi 2 i 3 imaju prioritet 1, itd.

Podušeno je da je aktivno stanje procesora prve 3 sekunde A0, naredne 3 sekunde A1 i poslednje 3 sekunde A2.

Grafički prikazi su generisani pomoću programa GTKWave.

4.1. Analiza rada simulatora

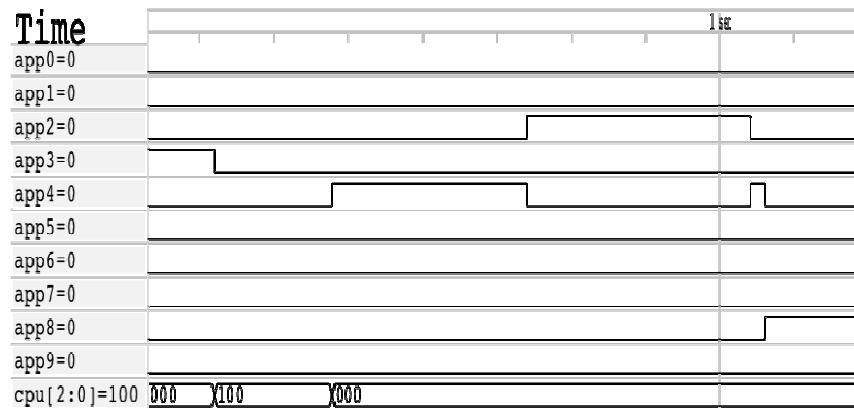
Na slici 6 prikazane su aktivnosti pojedinačnih taskova i ukupna aktivnost procesora.



Slika 6 – Aktivnosti pojedinačnih taskova i ukupna aktivnost procesora

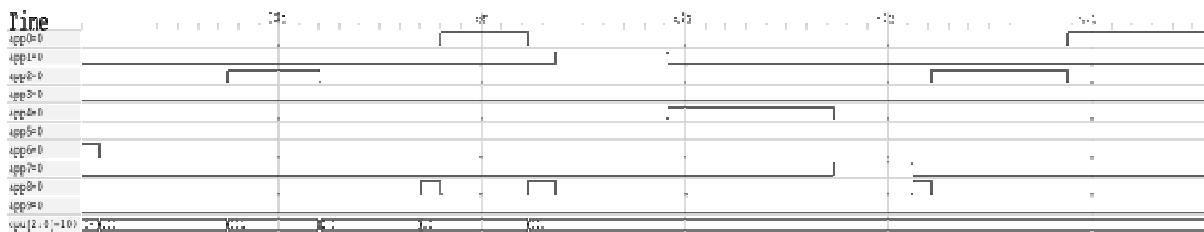
Stanja procesora su kodovana tako da bit 2 ukazuje na to da li je aktivno (bit 2 je 0) ili neaktivno stanje (bit 2 je 1), dok biti 1 i 0 pokazuju koje je stanje.

U trenutku $t=974$ ms task 2 sa prioritetom 1 prekida izvršavanje taska 4 (prioritet 2) i počinje da se izvršava, što je prikazano na slici 7.



Slika 7 – Prioritetno raspoređivanje taskova

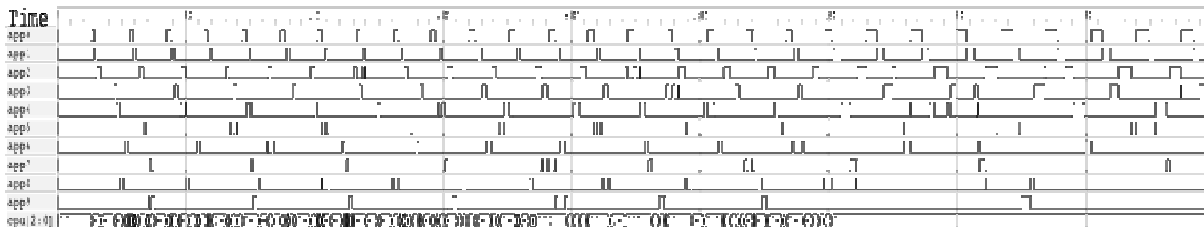
Sa promenom aktivnog stanja procesora, menja se i učestanost, što utiče na trajanje izvršavanja taskova. Na slici 8 se vidi da se vreme izvršavanja taska 2 i taska 8 povećalo nakon smanjenja učestanosti.



Slika 8 – Uticaj promene učestanosti na izvršavanje taskova

4.2. Procena potrošnje modelovanog procesora

Na osnovu aktivnosti procesora prikazanoj na slici 9 dobijaju se podaci o potrošnji prikazani u tabeli 2.



Slika 9 – Primer aktivnosti procesora

Tabela 2 – Procenjena potrošnja simuliranog procesora sa opterećenjem prikazanim na slici 9

E_{active}	382mJ
E_{idle}	12mJ
P_{avg}	43.9mW

Korišćenjem datih podataka moguće je odrediti uticaj parametara procesora na potrošnju, kao i analizirati kvalitet algoritama za optimizaciju.