



Архитектура рачунара
- јунски испитни рок -

1 (15)

1.1. (5) Написати оптималну секвенцу инструкција неопходних за срачунавање израза

```
int a, *b, c, d, e, f;  
...  
f = (a + b[c])/((a*4 + d) * (1 - e));  
...
```

процесор код кога аритметичке, логичке и померачке инструкције имају формат: ОС reg, reg, reg/imm где је ОС код операције, одредишни операнд и први операнд морају бити у регистру (reg), док други може бити или у регистру или дат непосредно (reg/imm). Инструкција LOAD има формат: LOAD reg, тем где је првим операндом дат одредишни регистар (reg), а другим извориште. Инструкција STORE има формат: STORE reg, тем где је првим операндом дат изворишни регистар (reg), а другим одредиште; а остале (A, B, C, D, E и F су глобалне променљиве које одговарају симболичким ознакама адреса меморијских локација у којима се налазе операнди. Садржај меморијских локација означених адресама A, B, C, D и E треба да остане непромењен, садржај одговарајућих регистара је дозвољено мењати. На располагању стоји 8 регистара опште намене. Сви подаци и адресе су исте дужине која је једнака адресибилној јединици.

1.2. (5) Написати оптималну секвенцу инструкција која одговара следећој C функцији која претвара у велика слова првих n знакова (велика слова, мала слова, бројеви, знаци, ...) низа src у низ dst:

```
char * toupper(char *src, char *dst, int n);
```

Формати инструкција и података су као у задатку 1.1. На располагању стоје и сложене инструкције. Функција враћа одредишни низ.

1.3. (5) У посматраном рачунарском систему процесор има 2 пара линија по којима улазно/излазни уређаји могу процесору да шаљу захтеве за прекид и од процесора добијају сигнале потврда. У датом систему има 4 улазно/излазних уређаја које треба некако повезати помоћу та 2 пара линија на процесор и омогућити за сваки улазно/излазни уређај скок на одговарајућу прекидну рутину векторисаним механизмом прекида.

а) Нацртати како та 4 улазно/излазна уређаја треба повезати помоћу та 2 пара линија на процесор.

б) Објаснити у којим се ситуацијама понашање овог система разликује од понашања система са 4 пара линија за слање захтева за прекид и добијање сигнале потврда.

2. (15) У рачунарском систему се налази једноадресни процесор, меморија и периферије PER0, PER1 и PER2. Све компоненте рачунара су повезане системском магистралом са 16 битном адресном и 16 битном магистралом података. Адресирање је на нивоу 16 битних речи. Улазно-излазни адресни простор је меморијски пресликан. Адресе релевантних регистара су:

PER0_CONTROL	FF00h	PER1_CONTROL	FF10h	PER2_CONTROL	FF20h
PER0_STATUS	FF01h	PER1_STATUS	FF11h	PER2_STATUS	FF21h
PER0_DATA0	FF02h	PER1_DATA0	FF12h	PER2_DATA	FF22h
PER0_DATA1	FF03h	PER1_DATA1	FF13h		
DMA_PER2_CONTROL	FF30h	DMA_PER2_DATA	FF32h	DMA_PER2_COUNT	FF34h
DMA_PER2_STATUS	FF31h	DMA_PER2_ADDR	FF33h		

У управљачким регистрима бит 15 је *Start* којим се дозвољава почетак операције, бит 0 одређују тип преноса података (1 - улаз (*input*), 0 – излаз (*output*)), бит 1 је *Enable* којим се дозвољава прекид, а у статусним регистрима бит 0 је *Ready* који сигнализира спремност контролера периферије. Бит 2 управљачког регистра DMA контролера задаје режим рада (0-блоковски (*burst*), 1-циклус по циклус (*cycle stealing*)). Периферије PER0 и PER1 су искључиво улазне, па немају бит којим се одређује тип преноса података.

Написати главни програм и одговарајућу прекидну рутину којима се обавља следећи пренос. Са периферије PER0 се прихвата један, а са периферије PER1 други велики целобројни, неозначени број (*BigUInt*) и смештају се почев од адреса 1000h и 2000h, респективно. Претпоставити да оба броја могу да се сместе у меморију почев од задате адресе. Велики целобројни неозначени број карактерише различита величина, па се трансфер броја са периферије врши тако што се цео број подели на више 16 битних речи. Речи се преносе од нижих ка вишим речима коришћењем DATA0 регистра периферија PER0 и PER1. Регистар DATA1 служи као индикатор да ли се у регистру DATA0 налази највиша реч броја *BigUInt*-а. Ако се у регистру DATA1 налази вредност нула, онда то значи да се у DATA0 налази највиша реч, а ако се у регистру DATA1 налази вредност различита од нуле, онда то значи да се у DATA0 не налази највиша реч. Ready бит статусних регистара периферија PER0 и PER1 добија активну вредност само када се и у DATA0 и у DATA1 учита нова вредност из периферијске јединице, док добија неактивну вредност ако се прочитају вредности и из DATA0 и из DATA1. Када се заврши прихватање оба броја, потребно их је сабрати и резултат сместити почев од адресе 3000h. У случају када су величине ова два броја различите, при сабирању сматрати да је број мање величине проширен нулама. Сабирање и смештање резултата је потребно реализовати у потпрограму **void** AddBigUInt(). Након завршеног потпрограма, збир је потребно послати на периферију PER2. Програм треба да ради циклично.

Прихватање *BigUInt*-а са периферије PER0 реализовати коришћењем механизма прекида, док примање *BigUInt*-а са периферије PER1 реализовати испитивањем бита спремности. Слање великог броја на периферију PER2 реализовати коришћењем придруженог DMA контролера. Процесор не поседује регистре опште намене. Дозвољено је користи додатне променљиве, али њихове називе треба писати описно и семантички исправно.

Напомене: На испиту нису дозвољена никаква помоћна средства, ни калкулатори, ни литература. Испит траје 150 минута. Студент је дужан да пише читко и уредно.