



**Архитектура рачунара**  
- јунски испитни рок -

1. (5) Контролер излазне периферије и излазна периферија за међусобно повезивање користе паралелни интерфејс који се састоји од линија података, једне статусне линије и једне управљачке линије. Статусном линијом контролер излазне периферије даје индикацију периферији да ли се у његовом регистру податка већ налази податак који је доступан на линијама података, док периферија управљачком линијом обезбеђује читање податка са линија података у свој интерни регистар.

а) Посматра се синхронизација између процесора и контролера периферије. Навести по ком редоследу и како се размењују информације (не обавезно сигнали) између контролера и процесора како би се обавио трансфер.

Редослед	Информација	Значење	Ко поставља

б) Након завршеног трансфера податка од процесора до контролера, која линија између контролера и периферије мења вредност, која је била стара вредност и које је значење нове вредности?

Линија	Стара вредност	Нова вредност	Значење

2. (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 6 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 тих 6 улазно/излазних уређаја треба повезати помоћу та 2 пара линија на процесор.

б) Објаснити како тих 6 улазно/излазних уређаја шаљу захтеве за прекид и добијају дозволе, и како се реализује скок на одговарајућу прекидну рутину.

3. (5) Написати оптималну секвенцу инструкција која одговара следећој стандардној библиотечкој С функцији која копира низ *src* у низ *dest*:

```
char *strcpy(char *dest, const char *src);
```

На располагању је процесор код кога аритметичке, логичке и померајке 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, mem* где је првим операндом дат одредишни регистар (*reg*), а другим извориште. Инструкција *STORE* има формат: *STORE reg, mem* где је првим операндом дат изворишни регистар (*reg*), а другим одредиште. На располагању стоји 8 регистара опште намене. Претпоставити да су сви подаци и адресе исте дужине која је једнака адресибилној јединици. На располагању стоје и сложене инструкције.

**4. (25)** У рачунарском систему се налази једноадресни процесор, меморија и периферије PER0, PER1 са придруженим контролером периферије DMA1, као и одвојени DMA контролер. 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 и меморијски адресни простор су раздвојени. Адресе релевантних регистара су:

PER0_CONTROL	FF00h	DMA1_CONTROL	FF20h	DMA_CONTROL	FF30h
PER0_STATUS	FF01h	DMA1_STATUS	FF21h	DMA_STATUS	FF31h
PER0_DATA	FF02h	DMA1_DATA	FF22h	DMA_DATA	FF32h
PER1_CONTROL	FF10h	DMA1_ADDR	FF23h	DMA_ADDR_SRC	FF33h
PER1_STATUS	FF11h	DMA1_CNT	FF24h	DMA_ADDR_DST	FF34h
PER1_DATA	FF12h			DMA_CNT	FF35h

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 4 управљачких регистра контролера DMA1 и DMA задаје режим рада (0-блоковски (*burst*), 1-циклус по циклус (*cycle stealing*)). Бит 5 управљачког регистра DMA контролера представља бит *Same* (1 - при преносу података се не мења изворишна адреса, 0 - при преносу података се мења изворишна адреса).

Написати главни програм, одговарајуће потпрограме и одговарајуће прекидну рутине којима се обавља следећи пренос. Периферија PER0 шаље низ А који се смешта у меморију почев од адресе 1000h. Пријем података од периферије PER0 престаје када периферија пошаље елемент низа са вредношћу *STOP*. Након примљеног комплетног низа, потребно је обрадити низ А позивом потпрограма **void Obrada()**. Овај потпрограм треба реализовати тако да замењује све елементе неопдајућих поднизова низа А са минималним елементом тог поднизва (нпр. за низ А са елементима 1, 3, 3, 5, 4, 8, 6, 4 обрађен низ би био 1, 1, 1, 1, 4, 4, 6, 4). Након завршене обраде низа потребно је цео низ послати периферији PER1. Након завршеног преноса обрађеног низа А на периферију PER1, потребно је обрисати цео низ уписивањем вредности нула свим елементима низа А. За брисање низа потребно је имплементирати и искористити потпрограм **void memset (unsigned short \*ptr, short value, unsigned short num)**, који уписује вредност *value*, почев од адресе *ptr* у наредних *num* локација.

Примање података са периферије PER0 реализовати испитивањем бита спремности. Слање података на периферију PER1 реализовати користећи DMA1 контролера. За реализацију потпрограма **memset** искористити DMA контролер. Потребно је обезбедити да газде магистралу држе заузетом што краће, по цену вишеструког тражења магистрале за коришћење.

Процесор не поседује регистре опште намене. Дозвољено је користи додатне променљиве, али њихове називе треба писати описно и семантички исправно. **Обавезно је писање концизних коментара над семантичким целинама.**

**Напомене:** На испиту нису дозвољена никаква помоћна средства, ни калкулатори, ни литература. Испит траје 90 минута. **Студент је дужан да пише читко и уредно.**