ARH1 - Jesenski ispitni rok 2016-2017 (13.9.2017.)

1. zadatak – TEORIJA (ukupno 23 boda):

1. a (7 bodova) 4-bitna ALU **oduzima** brojeve 0110-1101 (binarno). **Rezultat** oduzimanja je _____ (binarno), a stanja **zastavica** će biti: prijenos=___, ništica=___, predznak=___, preljev=___. Ako **operande i rezultat** gornjeg oduzimanja promotrimo kao **4-bitne brojeve u formatu 2'k**, njihovi su **iznosi**: prvi operand 0110 je _____, drugi operand 1101 je _____ i rezultat je _____. **ZA SVE PRETHODNE ODGOVORE MORA SE VIDJETI POSTUPAK IZRAČUNA.**

4-bitni format 2'k može prikazati brojeve u opsegu od _____ do _____. Prilikom gornjeg oduzimanja _____ došlo do prekoračenja opsega (**dopišite ''je'' ili ''nije''**). Kod **oduzimanja** u formatu **2'k**, **prekoračenje opsega se raspoznaje** kada je **zastavica** _____ u stanju _____.

1. b (1 bod) Ukoliko **poziv makronaredbe** smije **prethoditi njenoj definiciji**, potreban je barem _____ asembler, a u suprotnom slučaju dovoljan je i _____ asembler.

1.c (3 boda) Napišite **smjerove** FRISC-ovih priključaka: ADR je _____ , DATA je _____ , READ je _____ , WRITE je _____ , WAIT je _____ , INT je _____.

1.d (5,5 bodova) Na prazne crte upišite korake koje FRISC obavlja prilikom izvođenja naredbe LOAD R5, (R1+40)

Razina dohvata:

Rastući brid CLOCK-a:

_____

Padajući brid CLOCK-a:

_____

_____

_____

_____

_____

Razina izvođenja:

Rastući brid CLOCK-a:

_____

Padajući brid CLOCK-a:

_____

_____

_____

1.e (4 boda) Nakon uključenja **ARM7** izvodi se ovaj programski odsječak. Uz svako naredbu napišite koliko ciklusa traje pojedino izvođenje (npr. 5x1C+2x2C znači da naredba pčet puta traje po jedan ciklus i jednom traje dva ciklusa).

```
      ORG   0          _____
      MOV   R0, #3     _____
      LDR   R1, REZULT _____
LAB   SUBS  R0, R0, #1 _____
      BHS   LAB        _____
      STR   R0, [R1]   _____
```

Ukupno trajanje odsječka je _____ ciklusa

1.f (2 boda) Za procesor ARM upišite u registar R0 navedene heksadekadske brojeve naredbom MOV. Koristite sintaksu ATLAS-a kao na predavanjima (tj. rotaciju u lijevo). Ako se pojedini broj ne može upisati, napišite ''NE'':

$0020\ 4000_{16}$ _____ $0400\ 0002_{16}$ _____

$0000\ 0408_{16}$ _____ $0000\ 0222_{16}$ _____

1.g (0,5 boda) Specifičnost podatkovne sabirnice kod ARM-a je da _____

_____.

2. FRISC (14 bodova) Za procesor **FRISC** napisati **potprogram DIJELI** koji prima **dva parametra** preko **stoga**. Potprogram mora **cjelobrojno podijeliti** primljene **parametre** metodom **uzastopnog oduzimanja**. **Parametri i rezultat moraju biti u 32-bitnom formatu 2'k**. Dijeljenje s nulom zanemarite, tj. **drugi operand nikada neće biti nula**. **Rezultat** dijeljenja treba vratiti preko **R0**.

U memoriju se **na adresi 1000_{16}** nalazi **blok** sa 300_{16} **16-bitnih podataka s bitom za predznak**. Glavni program mora, pomoću potprograma DIJELI, **svaki od brojeva iz bloka** podijeliti s brojem -78_{16}, arezultate dijeljenja treba spremiti u memoriju kao **32-bitne podatke u formatu 2'k** u **blok ua adresi 2000_{16}**.

3. FRISC (17 bodova) na **FRISC** je spojena bezuvjetna vanjska jedinica BJ, i sklopovi DMA, CT i GPIO (adrese odaberite sami). **CT i GPIO** zahtijevaju prekide **INT**, ne mogu se međusobno prekidati, a CT ima **prioritet**.

Potrebno je sa GPIO (spojen na INT) **čitati 8-bitne NBC podatke i spremati** ih kao **bajtove** u memorijski **blok** MEMBL koji počinje na adresi 1000_{16}.

Svake 3 sekunde potrebno je **pomoću DMA-sklopa prenijeti** sve podatke **iz bloka MEMBL na bezuvjetnu jedinicu BJ. Nakon što dma-prijenos završi**, daljnji podatci koji se primaju od GPIO-a **ponovno se pune od početne adrese** 1000_{16}. DMA treba raditi **zaustavljanjem procesora.**

Kašnjenje od 3 sekunde ostvarite pomoću **CT**-a (spojen na INT), na čiji ulaz je spojen signal od **100 Hz**.

Pretpostavite da je DMA dovoljno brz da završi znatno prije nego što isteknu 3 sekunde. Također pretpostavite da unutar 3 sekunde GPIO neće prepuniti MEMBL.

Glavni program vrti beskonačnu petlju.

4. ARM (14 bodova) Za **ARM** treba napisati **potprogram ABS** koji preko registra **R1** prima **32-bitni broj u formatu s bitom za predznak**. Potporgram ABS računa **apsolutni iznos broja** i vraća ga registrom **R0**. (Zbog jednostavnosti, potprogram ABS treba pretvarati ''negativnu nulu'' u ''pozitivnu nulu''.)

Napisati **potprogram BRISI** koji u **bloku 32-bitnih brojeva u formatu s bitom za predznak briše** (tj. zamjenjuje nulom) sve brojeve čija je apsolutna vrijednost **strogo veća od 50_{16}**. Blok podataka zadan je početnom adresom i brojem podataka u bloku. **Početna adresa** prenosi se u potprogram preko **stoga**, a **broj podataka** prenosi se registrom **R0**. Također, potprogram treba **prebrojati obrisane** podatke, te njihov broj vratiti pozivatelju preko registra **R0**.

U glavnom programu treba pomoću potprograma **BRISI** obraditi blok od **100_{16} brojeva na adresi 1000_{16}**. Broj obrisanih podataka treba pohraniti na **lokaciju BROJ_OBRISANIH**.

5. ARM (17 bodova) Na **procesor ARM** spojeni su sklopovi GPIO i RTC (adrese im odaberite sami). Na ulaz sklopa RTC spojen je signal frekvencije **10 kHZ**, a RTC je spojen na **IRQ**.

Na **vrata A** sklopa GPIO spojen je **temperaturni sklop** kao na predavanjima (*podsjetnik: bitovi 0-5 su iznos temperature, bit 6 je ulazni za dojavu valjanog očitanja, bit 7 je izlazni za dojavu da je temperatura pročitana*).

Na **vrata B** sklopa GPIO spojen je **LCD-prikaznik** kao na predavanjima (*podsjetnik: bitovi 0-6 služe za slanje znaka, a bit 7 za slanje sinkronizacijskog impulsa, 04 prikazuje interno stanje, 0D briše interno stanje*).

Napišite program koji svakih **5 sekundi (kašnjenje ostvarite RTC-om i prekidima IRQ)** obavlja **provjeru** temperature i **ispisuje** rezultat provjere na LCD-u. **Provjera** se odvija tako da se očita **trenutačna** temperatura **Tt** i usporedi sa **željenom** temperaturom **Tž** koja je upisana u **memorijsku lokaciju ZELJ_TEMP**. Temperature su **NBC** brojevi.

Ispis na LCD-u ovisi o **odnosu Tt i Tž**. Ako Tt==Tž, treba ispisati ''='' (ASCII-kôd **3D**). Ako je Tt>Tž, treba ispisati ''+'' (ASCII-kôd **2B**). Ako je Tt<Tž, treba ispisati ''-'' (ASCII-kôd **2D**).

Za ispis pojedinog znaka na LCD **potrpogram** LCDWR (kao na predavanjima) LCDWR prima **ASCII-znak** registrom **R0**, a **adresu** GPIO-a registrom **R1**. LCDWR šalje znak na LCD, a LCD je spojen na vrata B.

Glavni program izvodi **beskonačnu** petlju.