

- Tentin mukana jaetaan taulukko "H8S/2000 CPU Instruction set & - codes". Palauta se tentin jälkeen!!!
- Kirjoita vastauspaperiin, milloin olet tehnyt harjoitustyöt (esim. Kesä 08, Kevät 08...)
- Tentissä on sallittua käyttää funktiolaskinta. Ota tehtäväpaperi mukaan, kun poistut.
- Ohjelmointitehtävissä hahmottele vastauksesi ensin suttupaperille.
- **Kirjoita selkeästi!!!!**

- 1 Kirjoita heksadesimaalisina ne luvut, jotka käännin tuottaa kysymysmerkkien paikalle alla olevan ohjelman riveillä 4, 8, 10, 15 ja 16. (Käskytaulukon takasivulta näet, mikä luku vastaa mitäkin rekisteriä käskyn rekisterikentässä.) (6p)

```

1                               .program koetehtava
2
3 00000000                       .section kayn_vekt,data,locate=0
4 00000000 ???                   .data.l h'FFA200
5
6 00FFA200                       .section koodi,code,locate=h'ffa200
7
8 00FFA200 ???                   mov.l @luvut, ER5
9 00FFA208 6A2800FFA224          mov.b @luvut+4, R0L
10 00FFA20E ???                  mov.l ER5, ER6
11 00FFA210 1035                 shl.l ER5
12 00FFA212 01F06565            xor.l ER6, ER5
13 00FFA216 13F5                 jatka:  rotl.l #2, ER5
14 00FFA218 131E                 rotxr.w E6
15 00FFA21A ???                  dec.b R0L
16 00FFA21C ???                  bne jatka
17 00FFA21E 40FE                 bra $
18
19 00FFA220 800001FF10          luvut: .data.b h'80, h'00, h'01, h'ff, h'10

```

- 2 a) Mikä on rekisterin E6 sisältö (heksadesimaalisena) ohjelman suorituksen päätyttyä? Perustele vastauksesi. (4p)
- b) Mitä ohjelma tekee? (2p)
- 3 Yllä olevassa ohjelmassa rivillä 16 sijaitsevan BNE-käskyn toimintakuvaus on: *if Z=0 then PC <- PC + d, else next*. Selosta kaikki käskyn suoritukseen liittyvät tapahtumat yksitellen (mieluiten ranskalaisin viivoin), kun tehtävän BNE-käsky ensimmäisen kerran suoritetaan. Aloita siitä, kun suoritin asettaa käskyn osoitteen FFA 21C osoiteväylälle. (6p)

KÄÄNNÄ!!!

- 4 Jakolaskukäsky DIVX ei ilmoita käyttäjälle mahdollisesti tapahtuneesta lukualueen ylittymisestä.

Tee aliohjelma, joka suorittaa DIVXU.B-käskyn, jos tulos mahtuu sille varattuun paikkaan. Mikäli lukualue ylittyy (tulee ylittymään) suoritetaan DIVXU.W-käsky. Jaettava välitetään aliohjelmalle rekisterissä R1 ja jakaja R0L. Tulos välitetään pääohjelmalle alla olevan kuvan mukaisissa rekistereissä ja V-lipun arvolla osoitetaan, mistä rekistereistä tulos on luettavissa:

V = 0 → osamäärä ja jakojäännös ovat 8-bittisiä
 V = 1 → osamäärä ja jakojäännös ovat 16-bittisiä.

Tarkoitus on siis tehdä "täydellinen jakolasku", mutta sen ei tarvitse toimia oikein jakajan arvolla nolla, koska tästä DIVX-tyyppinen käsky ilmoittaa Z-lipussa (jakaja = 0 → Z=1 muuten Z=0). Sinun tulee huolehtia, että Z-lipun arvo on tässä mielessä oikea, kun aliohjelmasta poistutaan. Vinkki: jos lukualue ylittyy, toteutuu ehto: $R1 / R0L > 255$. (6p)



- 5 Käytössäsi on muistikomponentteja kooltaan 32, 16, 8, 2 ja 1 Ktavua. Suorittimen dataväylän leveys on 8 bittiä.
- Täytä muistialue H'FFE800–H'FFFFFF annetuilla komponenteilla siten, että käytettyjen komponenttien määrä minimoituu. (4p)
 - Millaiseen ongelmaan törmäät, jos dataväylän leveys olisikin 16 bittiä? (2p)