

Logiikka-analysaattori ja väyläliikenne

Sisältö

Työssä perehdytään mikroprosessorin väylien toimintaan mittaamalla näiden signaaleita ja analysoimalla saatuja tuloksia.

Tavoite

Tavoitteena on oppia ymmärtämään väylien sekä tyypillisen sulautettujen järjestelmien kontrollerin toimintaa ja käyttämään väyläsignaalien mittaamiseen soveltuvia laitteita.

Laitteet

Prossessorikortti EVB2144F ja logiikka-analysaattori.

Yleistä

Alla seuraavan johdannon yhteydessä on esitetty aiheeseen liittyviä esiselostuskysymyksiä. Nämä tulee esittää assistentille ennen varsinaisen työn tekemistä. Parhaiten tämä onnistuu päivystysaikoina, jotka löytyvät kurssin verkkosivuilta. Palautuksissa tulee olla vastaukset kysymyksiin perusteluiden ja pohdintojen kera. Etusivulle kirjoitetaan

- kurssin nimi ja numero
- ryhmän numero
- ryhmän jäsenten nimet ja opiskelijanumerot
- päivämäärä.

Kun esiselustus on tarkastettu, suoritetaan mittaukset laboratoriossa TC 321. Lopulliseen työselostukseen tulee liittää tarkastettu esiselustus, tämän mahdollinen korjausliite, mittauspöytäkirja sekä jälkiselustus. Lopullinen työselustus palautetaan assistentille niin ikään päivystysaikoina.

Johdanto

Väylien tehtävänä on toimia rinnakkaismuotoisen datan siirtotienä suorittimelta tai integroidulta piiriltä jollekin toiselle komponentille. Tällaisia komponentteja voivat olla esimerkiksi muistit, erilaiset laskurit ja rinnakkais- ja sarjamuotoiset liittynät. Väylien varrella voi toki sijaita, ja yleensä sijaitseekin, useampi komponentti. On hyvin ymmärrettävää, että mitä enemmän komponentteja sijoitetaan väylän varrelle sitä monimutkaisemmaksi tulevat säännöt, joiden mukaan väylällä liikennöidään. Lisäksi väyliä esiintyy suorittimen sisällä. Näitä kutsutaan yllättäen sisäisiksi väyliksi. Tässä työssä perehdymme yksinomaan tarkastelemaan prosessorin ja muistin välistä liikennettä. Eli osoiteväylä, dataväylä sekä näihin liittyvät yksittäiset

signaalit ovat tarkastelun kohteena. Tämä pitäisi olla tuttua asiaa. Jos kuitenkin tunnet jo tässä vaiheessa epävarmuutta puhuttuja asioita kohtaan, ota mikroprosessorit-kurssin luentomoniste (Uotila) esiin ja kertaakaan sieltä luku 10. Ja jos sinulla ei ole kyseistä monistetta, työpaikalta löydät kansion, jossa luku 10 on kopioituna.

E1 Selitä mitä tarkoitetaan ulkoisella ja sisäisellä dekodauksella muistikomponenteista puhuttaessa.

E2 Miten muistikomponentit sijoitetaan 16-bittiseen dataväylään. (Muistipaikan koko 8 bittiä. Kuva voisi olla paikallaan.)

Työn aiheena on siis väylätoimintojen mittaaminen. Testipiirinä toimii mikroprosessorit kurssilta tutuksi tullut evaluaatiokortti EVB2144F. Prosessori suorittaa ohjelmaa, joka on sijoitettu muistikomponenteille, joiden koko on 128 kilotavua. Ohjelma on siis ulkoisessa muistissa, joten prosessori hakee sitä ulkoisen dataväylän kautta. Miksi sitten ohjelman tulee sijaita ulkoisessa muistissa? Prosessorin sisäisiin väyliin on vaikea päästä käsiksi. Sen sijaan ulkoisia väyläsignaaleita voimme tutkia vaikkapa kytkemällä mitta-anturit muistikomponenttien jalkoihin. Seuraavaksi herääkin kysymys, miten ja millä väyläsignaaleja mitataan.

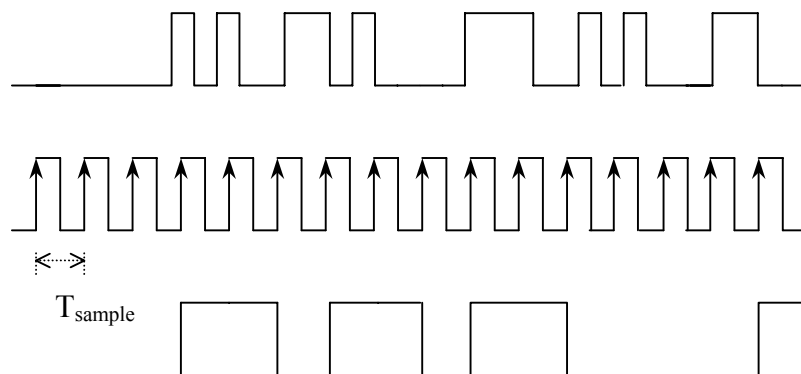
Ensimmäiseksi mieleen tulee käyttää oskilloskooppia. Kun väylien kellotaajuus tunnetaan, oskilloskoopin ruudulta olisi helppo lukea signaalin tila, joka on joko ykkönen tai nolla. Kun asiaa miettii pidemmälle, on helppo huomata, että mittaustapa on hieman ongelmallinen. Väylillä kulkevat bitit ovat satunnaisia. Oskilloskooppi on verraton mittalaitte jaksollisia signaaleita mitattaessa, mutta jaksottomiin se ei pysty tahdistumaan. Edellä kuvatussa tilanteessa oskilloskoopin ruudulla näkyisi pelkkää kohinaa, ellei väylän johtimella sattuisi kulkemaan ykköstä ja nollaa vuoronperään (tai näiden symmetrisiä yhdistelmiä) koko mittaustapahtuman ajan. Oskilloskoopilla on mahdollista mitata maksimissaan neljää signaalia yhtä aikaa. Nyt kuitenkin jo dataväylällä 16:ta signaalia pitäisi päästä mittaamaan samaan aikaan ja vielä osoiteväylän bitit tulisi saada tietoon kunkin muutoksen yhteydessä. Digitaalisignaaleita tarkasteltaessa ollaan yleisesti kiinnostuneita ainoastaan kahdesta jännitetasosta, ei signaalin muodosta. Väylämittausiimme riittää varsin hyvin laite, joka kertoo kulkeeko väylän johtimella ykkös- vaiko nollabitti. Ja koska binääriluvut ilmaistaan yleisesti heksalukuina, säästyisi vaivaa, jos mittalaitteemme näyttäisi väyliä bitit suoraan tällaisina lukuina. Näin säästyisi tilaa ruudullakin.

Kuten todettiin, väyläsignaalit ovat yleisesti jaksottomia ja muutosnopeudeltaan sellaisia, että ihmissilmä ei ehdi lukemaan arvoja reaaliajassa. Tämä tarkoittaa sitä, että mittalaitteemme on tallennettava väylätapahtumat ensin muistiin, josta voimme sitten poimia ne ruudulle muutama tapahtuma kerrallaan. Toisaalta, koska muisti on rajallinen, laite pitää pystyä asettamaan niin, että mittaus aloitetaan vasta jostakin halutusta kohdasta. Tyypillinen aloituskohta on jokin tietty osoite- tai dataväylän arvo. Toisin sanoen jokin muistiosoite tai sen sisältö. Laite, joka tekee kaiken tämän, on nimeltään logiikka-analysaattori.

- E3** Edellä todettiin, että logiikka-analysaattorin tarvitsee ilmaista ainoastaan, onko väylän johtimen looginen tila ykkönen vai nolla. Kerro yksinkertainen periaate, miten tämä toteutetaan.

Logiikka-analysaattori

Edellä kehitit periaatteen yksinkertaiselle näytteistämiseksi (sampling). Näytteistuksen yhteydessä puhutaan myös näytteistystaajuudesta. Tämä tarkoittaa, että signaalista otetaan näytteitä tietyn väliajoin. Logiikka-analysaattori voidaan asettaa näytteistämään signaalia sen oman kellon määräämään tahtiin. Signaali ei siis riipu mitenkään analysaattorin kellosta. Jos asetettu näytteistystaajuus on liian pieni, on selvää, että jotain voi jäädä huomaamatta. Kuva 1 selkeyttää asiaa.



Kuva 1. Näytteistys. Ylhäällä näytteistettävä signaali, keskellä kellosignaali ja alhaalla näytteistuksen tulos

Kuvassa nuolet merkitsevät, että signaalia näytteistetään kellosignaalin nousevalla reunalla. Analysaattorin ruudulla näkyvä signaali poikkeaa huomattavasti oikeasta. Näytteistys ei ehdi mukaan signaalissa tapahtuviin nopeisiin muutoksiin. Näin käy, jos siis näytteistystaajuus on liian pieni. Nyt voisit hahmotella mieleesi/paperille kuvan yllä olevasta signaalista, kun sen taajuutta lähdetään pienentämään. Näytteistystaajuus pysyy samana, mutta itse mitattava signaali levenee. Yritä löytää signaalin taajuus, jolloin se pystytään mittaamaan oikein.

- E4** Mikä on mitattavan signaalin taajuuden suhde näytteistystaajuuteen, jotta mitattava signaali tulee näytteistettyä oikein?

Logiikka-analysaattoriin voidaan tuoda näytteistuksen kellosignaali myös ulkoa. Tämä on järkevää silloin, jos voimme tuoda mitattavia signaaleita ohjaavan kellosignaalin myös näytteistykselle. Tällöin analysaattorimme näyttää

automaattisesti väylien todelliset arvot, ja meidän ei tarvitse huolehtia riittävän suuresta näytteistystaajuudesta sen enempää.

Yleisesti mittaus aloitetaan, kun jokin tietty väylä tapahtuma ilmenee. Esimerkiksi voimme olla kiinnostuneita tapahtumista, jotka seuraavat, kun prosessori käsittelee muistipaikkaa h'xxxxxx. Tai mikä on se osoite, johon ohjelman suoritus siirtyy, kun käsky yyy suoritetaan. Voimme myös asettaa useampia ehtoja mittauksen alkamiselle. Esimerkiksi mittaus aloitetaan vain silloin, kun jokin määrätty luku kirjoitetaan tietty muistipaikkaan. Tai silloin, kun dataväylällä on arvo yyy ja osoite h'xxxxxx on esiintynyt tätä ennen kolmesti. Nyt olemme puhuneet mittauksen tahdistumisesta mitattavaan signaaliin, mistä usein käytetään sanontaa triggeraus (triggering).

Prossessorikortin väylätapahtumien mittaaminen

Olet mahdollisesti tutustunut Hitachin prosessorikorttiin jo Mikroprosessorit-kurssilla. Nyt näet kortin toisen puolen, jossa sijaitsee itse suoritin ja muistit. Tässä työssä mittaamme kyseisen kortin väylätapahtumia tarkoituksenamme selvittää muun muassa prosessorin suorittama ohjelma. Kun prosessori käsittelee sisäistä muistiaan, tapahtumat eivät näy ulkoisilla väylillä, ja taas sisäisiin väyliin emme pääse käsiksi. Toisaalta ulkopuolinen muisti on RAM-muistia, jossa ohjelma säilyy ainoastaan niin pitkään, kun kortilla on virta päällä. Jos kiinnostaa, tässä vaiheessa voisi keskeyttää tekstin lukemisen hetkeksi ja miettiä, miten ratkaista ongelma. Yksi tapa ratkaista tämä on tehdä ohjelmamme sisäiselle flash-muistille (ROM) josta se ensin siirtää itsensä (suoritettavan osan) ulkoiselle RAM muistille ja kun tämä on tehty ohjelman suoritus hyppää siirretyn osan alkuun. Alla on esitetty se osa ohjelmaamme, joka siirtää varsinaisen ohjelman.

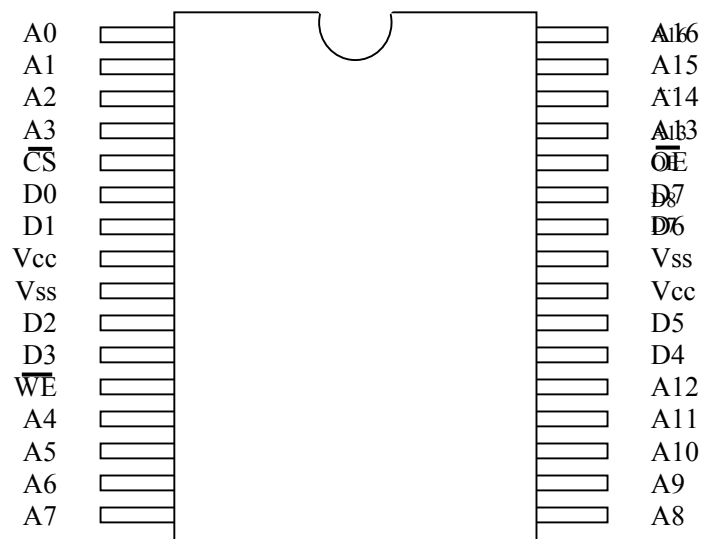
```
00001000 7A000000*siirto:    mov.l    #alku, ER0        ; tasta lahtien siirretaan
00001006 7A010002*              mov.l    #h'20100, ER1    ; ohjelman osoite ulkona
0000100C 6902    next_16:      mov.w    @ER0, R2        ; otetaan sisältä...
0000100E 6992              mov.w    R2, @ER1        ; ... ja siirretään ulos
00001010 7A200000*              cmp.l    #loppu, ER0     ; onko kaikki siirretty
00001016 4706              beq     alustus         ; jos on tehdään alustus
00001018 0BF0              inc.l    #2, ER0        ; jos ei, siirretään...
0000101A 0BF1              inc.l    #2, ER1        ; .. seuraavat 16 bittiiä
0000101C 40EE              bra     next_16
```

Kuva 2. Ohjelma, joka siirtää flash-muistilla olevan, suoritettavan ohjelman ulkoiselle RAM-muistille.

Ohjelma, jota mittaamme, alkaa osoitteesta `alku` sisäisessä muistissa. Osoite `loppu` merkitsee ohjelman päättymistä eli myös ohjelman siirtäminen loppuu, kun tämä viimeinen rivi on tullut vietyä ulkoiselle muistille.

Kun mittaamme ulkoista väylää, helpoimmalla pääsemme, kun kytkemme ns. testiklipsit muistikomponenttien pinneihin. Mittalaitteelta tulevat johdot on sitten helppo kytkeä näihin klipseihin. Tarvitsemme kaksi klipsiä, koska toisella muistikomponentilla on parilliset osoitteet ja toisella parittomat. Tästä on kuva Mikroprosessorit-kurssin monisteessa luvussa 10. Kuvassa 3 on esitetty muistikomponenttien pinnijärjestys.

Tämä järjestely ei ole täysin ongelmaton. Testiklipsit ovat erittäin heiveröisesti kiinni muistikomponenteissa ja siis irtoavat jo pienestäkin heiluttelusta. Jos klipsit siirtyvät paikaltaan, ne helposti yhdistävät kaksi pinniä, ja riippuen pinnien loogisista tiloista, seurauksena saattaa olla oikosulku. Vältä kehikon tönimistä työpaikalla!



Kuva 3. Ulkoisen RAM-muistikomponentin pinnijärjestys.

Logiikka-analysaattorimme pystyy mittaamaan 32:n signaalin loogisia tiloja yhtäaikaaisesti. Tämä on hienoinen ongelma, koska emme täten saa näkyviin kaikkia osoite- ja dataväylän bittejä yhtäaikaisesti. Lisäksi luku- ja kirjoitussignaaleiden tutkiminen on usein yhtä tärkeää.

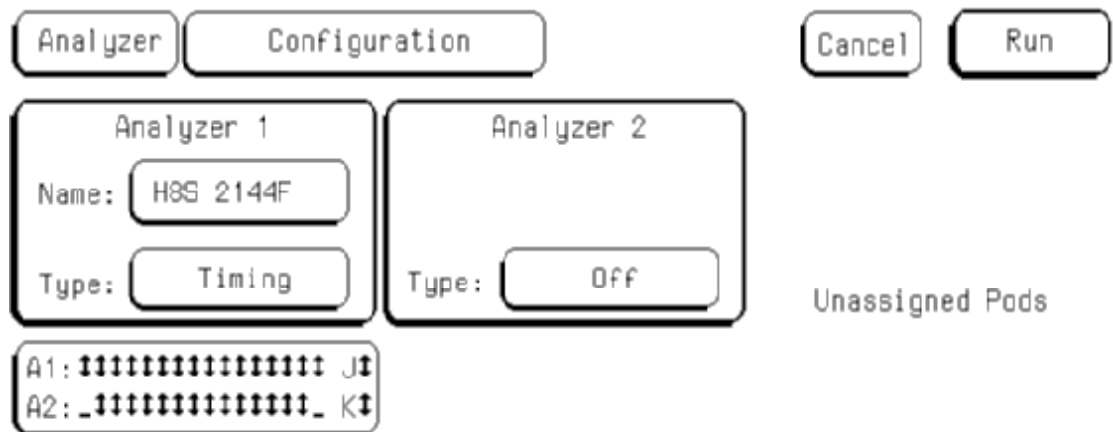
E5 Kuinka monta datakanavaa logiikka-analysaattorissa tulisi olla, jotta voisimme mitata yhtäaikaisesti dataväylän, osoiteväylän, kirjoitus- ja lukusignaalien kaikki bitit?

Kuten kuvan 2 koodista voimme tulkita, ohjelmamme alkaa ulkoisen muistin osoitteesta h'20100. Koska ohjelmamme sijaitsee muistiosoitteiden h'20100 ja h'20FFF välissä (*, voimme unohtaa osoiteväylä bitit A23 - A12, koska nämä pysyvät muuttumattomina. Riittää siis, että ainoastaan 12 kanavaa on käytetty osoiteväylän mittaamiseen. On selvää, että dataväylä tulee kytkeä kokonaisuudessaan mittalaitteeseen. Nyt olemme käyttäneet 28 kanavaa. Lukusignaali vie yhden ja kirjoitussignaalit kaksi kanavaa, joten paljon ei jää ylimääräistä.

Logiikka-analysaattorilla mittaaminen

Tässä vaiheessa kannattaa siirtyä työpaikalle analysaattorin pariin. Tällöin tekstistä saa hieman paremmin selvää.

Kun kytket logiikka-analysaattorin päälle, kuvaruudun pitäisi näyttää suurin piirtein kuvan 4 mukaiselta.

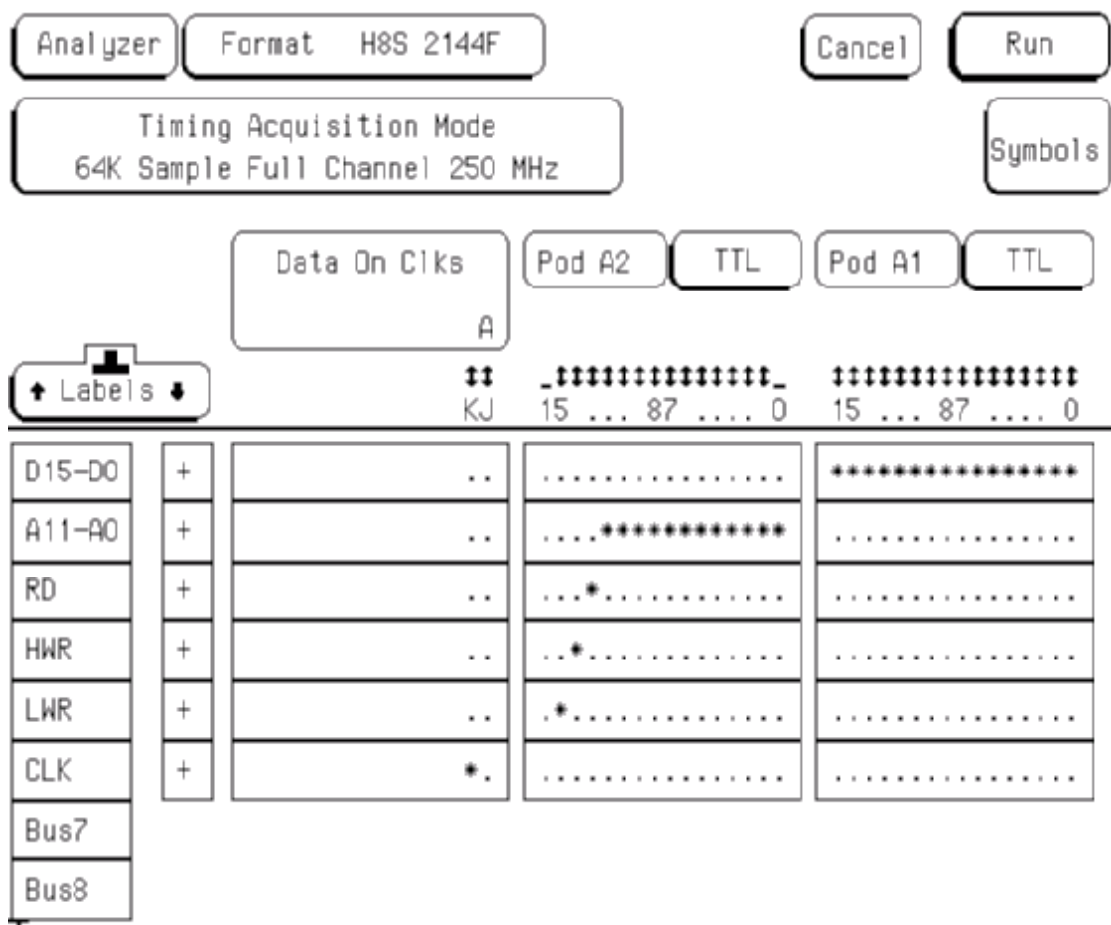


+

Kuva 4. Configuration-ikkuna. Tässä ikkunassa voimme ottaa käyttöön kovalevylle talletettuja mittaasetuksia.

(* tätä ei pitäisikään pystyä tekstin perusteella päättelemään.

Vasemmalla ylhäällä olevasta laatikosta valitaan moodi ja tämän vieressä olevasta laatikosta löytyy moodin valikko. Analysaattorimme siis avautuu oletusarvoisesti *Analyzer*-moodissa ja tämän valikosta *Configuration*-ikkuna ilmestyy ruudulle. Mittaukset suoritetaan tässä moodissa. Avautuneessa valikon ikkunassa voimme ottaa käyttöön itse tehtyjä ja muistiin talletettuja mittausasetuksia. Nyt käytössä on mittausasetukset nimeltään *H8S 2144F* (laatikko name). Tämän alapuolella olevassa valinnassa näkyy, mikä mittaustyyppi on käytössä. Mittauksissamme käytössä on ajoitusmittaus. Muita vaihtoehtoja olisi tila-analyysi, tilavertailu sekä järjestelmän suorituskykyanalyysi (SPA). Ikkunassa pääset liikkumaan nuolinäppäimillä (tai hiirellä) ja valinnan sisälle pääset näppäimellä *Select*. Kokeile. Valikossa, *Configuration*-ikkunan alapuolella, löytyy ikkuna *Format H8S 2144F*. Kun siirryt sinne, ikkuna näyttää kuvan 5 mukaiselta.



Kuva 5. *Format*-ikkuna. Tässä ikkunassa määritellään mittapäille halutut mittauskohteet sekä nimetään signaalit tai signaaliryhmät.

Logiikka-analysaattorissamme on siis 32 kanavaa, joka tarkoittaa, että voimme mitata yhtäaikaisesti 32 eri signaalia. Ikkunassa *Format H8S 2144F* pääsemme kokoamaan yksittäiset kanavat yhteen. Esimerkiksi 16 kanavaa olemme varanneet dataväylälle, 12 osoiteväylälle jne. Lisäksi voimme tuoda analysaattorille kaksi ulkopuolista kellosignaalia *J* ja *K*. Vaikka prosessorin kellosignaali on tuotu mainittujen kanavien kautta, käytämme analysaattorin sisäistä kelloa näytteistykseen. Kanavien johtimet on jaettu 16 kappaleen ryhmiksi. Ikkunasta voidaan määrittää

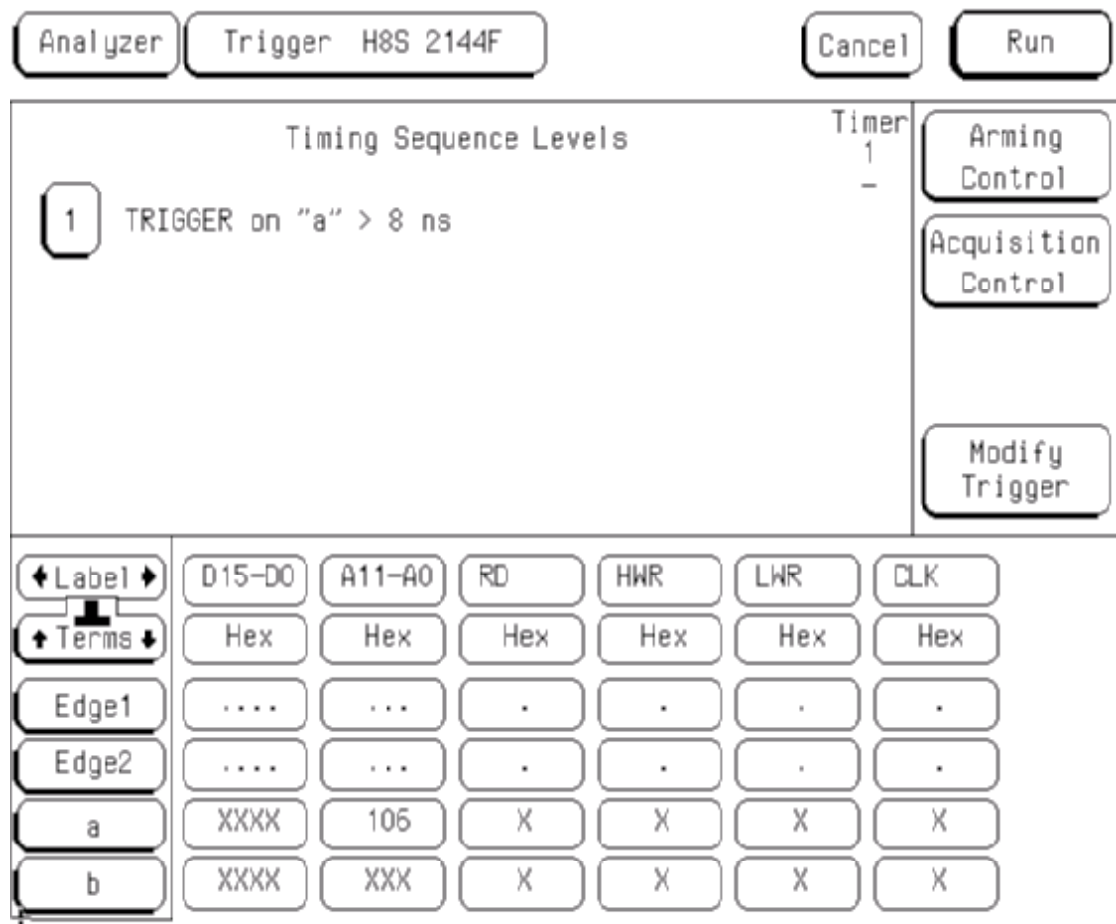
kunkin johtoryhmän (Pod:in) signaalitasot. Nyt olemme valinneet TTL-tasot. Huomaat, että Pod A1 on kokonaan valittu dataväylälle. Nyt kun signaalit on yhdistetty, voimme kuvata niiden signaalitiloja lukuina. Siirtymällä ikkunaan Symbols voimme määrittellä kullekin nimikkeelle (labels) symboleita siten, että logiikka-analysaattori ilmoittaa määritellyn symbolin sitä vastaavan luvun ilmaantuessa väylälle.

E6 Esitä muutama esimerkki, milloin saattaisi olla kannattavaa määrittellä symboli jollekin osoite- tai dataväylän arvolle?

Tällä kertaa jätämme symbolien määrittelyn pois ja tyydymme pelkkiin lukuarvoihin. Liiku ikkunassa ja tutustu valintojen takana oleviin toimintoihin.

Jos vahingossa muutat jotakin arvoa ja et muista alkuperäistä, kannattaa analysaattori buutata. Muutoksesi eivät talletu konfiguraatitiedostoon.

Siirtykäämme valikossa eteenpäin. Seuraavana on vuorossa *Trigger H8S 2144F* -ikkuna, joka näyttää seuraavalta.



Kuva 6. *Trigger H8S 2144F* -ikkuna, jossa määritellään laitteen triggausehdot.

Tässä ikkunassa voidaan asettaa mittaukselle alkamisajankohtia eli niin sanottuja triggausepisteitä. *Timing Sequence Levels* kertoo, että konfiguraatitiedostossa on määritelty ainoastaan yksi triggausehto. Trikkkaus tapahtuu, mikäli *Terms*-valikon a-

rivillä asetetut määrittelyt toteutuvat. Nyt siis mittaus tapahtuu, mikäli osoiteväylällä on arvo h' 20106. Triggausehtoja voi asettaa toki useampia valitsemalla *Modify trigger* ja sieltä valikon *Add Sequence Level*. Tutustu tähän. Niin ikään triggaustermien määrittelyitä voi olla enemmän kuin yksi. Määrittelyitä voit muuttaa vaikka viemällä hiiren kohdistimen valitsemasi ikkunan kohdalle, klikkaamalla ja asettamalla haluamasi luvun. Merkintä X tarkoittaa niin sanottua don't care -tilaa.

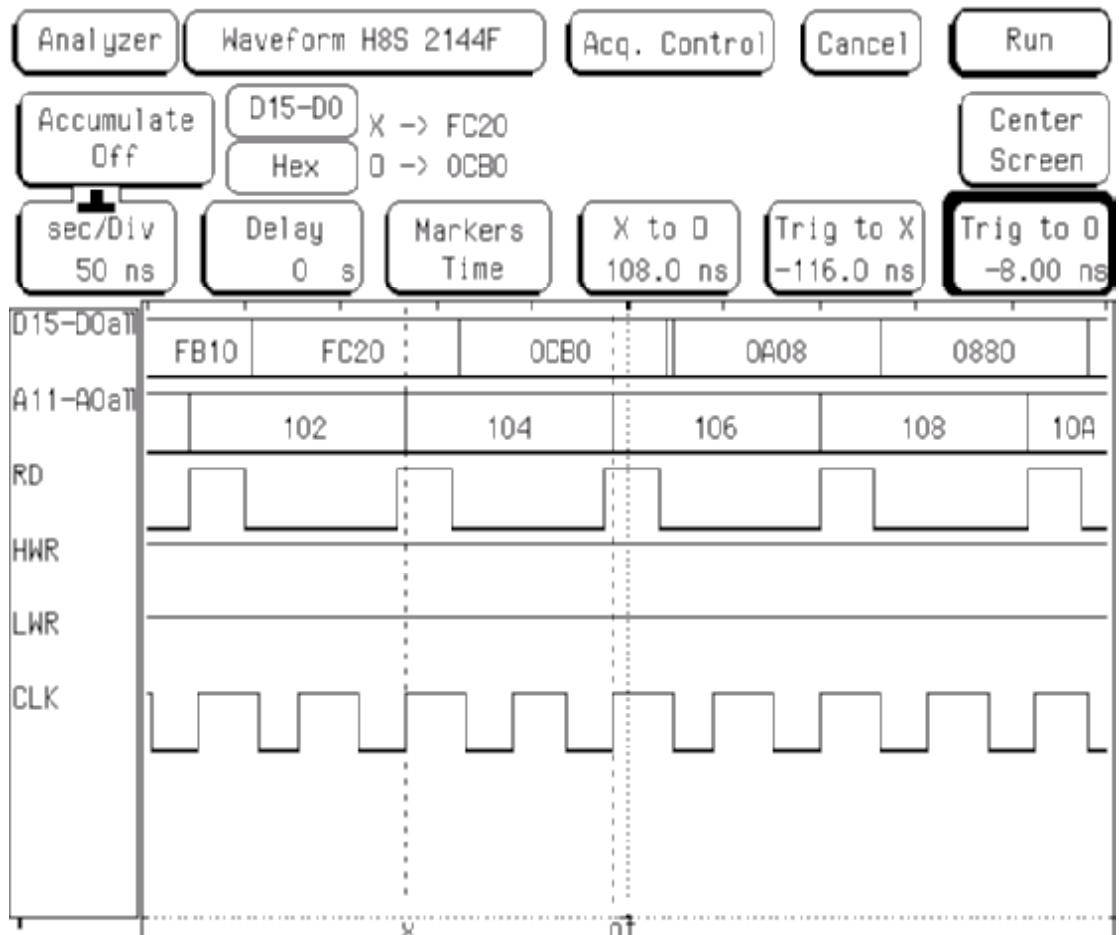
Tässä työssä riittää, että käytämme yhtä, jo ruudulla olevaa, triggausehtoa "a". Sen sijaan termin "a" määrittelyitä joudumme muuttamaan.

Seuraavaksi tarkastelemme ikkunan *Acquisition Control* sisältöä. Huomaat, että analysaattori tallentaa yhtä monta signaalin arvoa ennen triggausta kuin sen jälkeen, kaiken kaikkiaan 65536 tavua dataa. Toinen vaihtoehto olisi aloittaa datan tallennus triggaukskohdasta. Tämä on luonnollisesti hyödyllisempi tapa silloin, kun triggaukskohtaa edeltäneet arvot eivät ole kiinnostavia. Tässä työssä tarkastelemme väylien tiloja ennen ja jälkeen triggauksen, joten asetukset saavat jäädä ennalleen. *Sample Period* kertoo näytteistykseen jakson ajan. *Configuraatiotiedostossa* se on määriteltä 4 ns:ksi. Eli analysaattori tutkii signaaleiden tiloja 4 ns:n välein. Tämä on suhteellisen lyhyt aika, ja tarkoittaa, että analysaattorimme muisti täyttyy nopeasti. Asetus saa kuitenkin jäädä ennalleen. Asiasta lisää mittausosiossa.

Analyzer-valikon alimmainen ikkuna on nimeltään *Waveform H8F 2144F*. Siirry sinne. Nyt kannattaa suorittaa koemittaus painamalla Run. Ruudulle pitäisi ilmestyä kuta kuinkin kuvan 7 mukainen näkymä.

Nyt olemme vihdoinkin päässeet valikon ikkunaan, jossa tarkastelemme mittauksuloksia. Vasemmalla on listattu väylien tai signaalien nimet ja näitä seuraavat mittauksulokset. Väyläarvot näytetään heksälukuina, ja signaalit niiden tasojen mukaan.

Vasemmalla oleva *Accumulate*-valinnan tila on tulostaa jokaisen mittauksen edellisen päälle. Tätä emme halua, joten valinta saa olla tilassa Off. Oikealla, Run-painikkeen alapuolella oleva valinta kertoo, että mittauksuloksia on talletettu yhtä paljon triggaukskohtaa ennen kuin sen jälkeen. Kun siirrät kursorin valinnan *sec/Div* päälle ja painat näppäintä select, voit zoomata mittauksuloksiasi käyttämällä valkoista pyörrintä analysaattorin etupaneelissa. Kun siirryt kohtaan *Delay*, vastaavat toimenpiteet siirtävät aika-akselia ja pääset katselemaan mittauksuloksia ennen tai jälkeen triggauksen. Valkoinen "hattu" laatikon päällä kertoo mihin toimintoon pyörrin kulloinkin vaikuttaa. *Select*-napin painaminen siirtää hattua. Enää on jäljellä markkerit, jotka näkyvät ruudulla katkoviivoina. Näitä pääset käyttämään valinnan *Markers* kautta. Tutustu tähän toimintoon. Työssämme mittaamme ainoastaan aika-arvoja (*Pattern*-toimintoa emme tarvitse). Triggauks tapahtuu markkerin t kohdalla, joka on keskellä ruutua silloin kun *Delay* on nolla. Markkereita X ja 0 pääset liikuttelemaan aktiivisella kohdat *Trig to X* ja *Trig to 0*. Kohdassa *X to 0* näkyy näiden erotus, eli siis markkereiden "välinen aika".



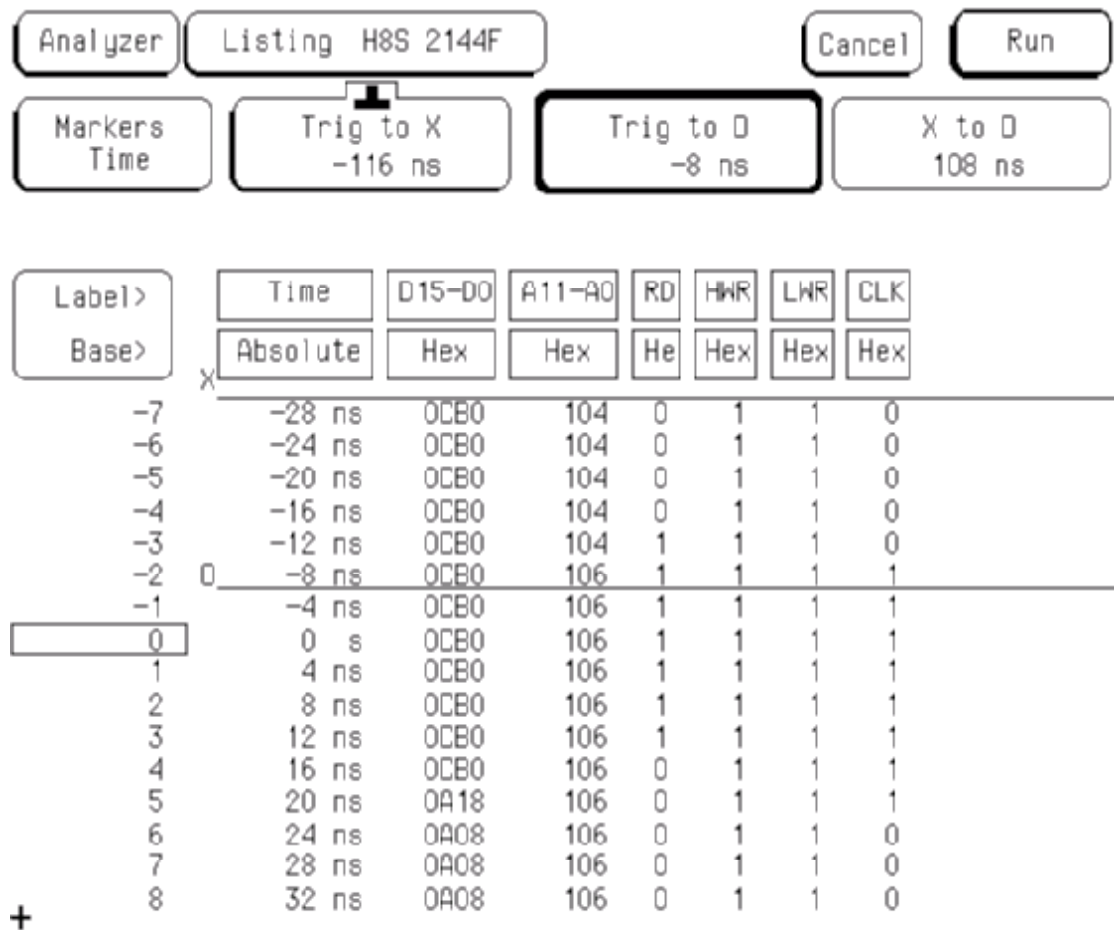
Kuva 7. Waveform H8F 2144F -ikkunassa voidaan tarkastella mittaustuloksia.

Väylillä saattaa esiintyä hyvin lyhytkestoisia tapahtumia. Jotta tämä ei häiritse työskentelyämme, kannattaa aika-akseli (zoomaus) asettaa kuvan 7 mukaiseksi (50 ns /Div) tai tätä suuremmaksi.

Analyzer-moodissa on vielä yksi ikkuna jäljellä. Tämä on nimeltään *Listing H8F-2144F* ja se on esitelty kuvassa 8. Tässä ikkunassa näkyvät samat mittaustulokset, joita tarkastelimme jo edellisessä kohdassa, nyt jokainen näytteistys listattuna allekkain. Se, onko tämä tarkastelumuoto parempi kuin waveform-ikkunan tarjoama, riippuu lähinnä mitä tarkastelemme. Muutama selvä etu *Listing*-ikkunalla kuitenkin on. Jos olemme määritelleet symboleita signaaleiden arvoille, nämä näkyvät myös täällä. Esimerkiksi jos määrittelemme symbolin INC.B dataväylän arvolle 0A09, niin käskyn tunnus ilmestyy vastaavan luvun kohdalle, kun prosessori hakee muistista käskyn operaatiokoodin. Näin voimme saada mittaustulokset assembly-koodin muodossa. Tosin, jos nyt mainittu luku luetaan muistista MOV-käskyllä, mittaustuloksemme näyttävät virheelliseltä.

Lisäksi on tilanteita, joissa aaltomuodot eivät kerro meille mitään hyödyllistä vieden ainoastaan tilaa tulosteissa. Tällöin *Listing*-ikkuna on järkevämpi dokumentaatiomenetelmä.

Palaamme tähän ikkunaan tehtävässä M4, mutta tutustu ikkunaan jo tässä vaiheessa. *Listing*-ikkuna on esitetty kuvassa 8.



Kuva 8. *Listing H8F-2144F* -ikkuna näyttää mittaustulokset listauksen muodossa

Lisäksi analysaattoristamme löytyy mainitsemisenarvoinen moodi. Tämä on nimeltään *System*. Siellä voimme huolehtia mittalaitteen liitynnöistä sekä muun muassa käyttää laitteen kiintolevyä. Tutustu tähän moodiin kuitenkin muuttamatta asetuksia. Ennen kaikkea sinun ei tule muuttaa kovalevyn tilaa.

Mittaustulokset (kuvaruutunäkymä) talletetaan korpulle, josta kuvat liitetään selostukseen. Ota mukaasi tyhjä korppu, kun tulet tekemään mittaussosiotia. Tallennus tapahtuu seuraavasti. Kun olet *Waveform*- tai *Listing*-ikkunassa ja mittauksen tulokset ovat ruudulla, paina *Print*-valintaa. Kohteeksi valitse *Print Disk*. Kuvaruudulle aukeaa näkymä, jossa tiedostolle annetaan nimi ja valitaan tulosteen muoto (*Output Form*) sekä kohde (*Output Disk*). Muoto saa olla *B/W Tif*, mutta kohteeksi valitaan *Flexible Disk*.

Nyt olemme saaneet käsityksen, miten mittaamme väyläsignaaleita ja kuinka oivallinen laite logiikka-analysaattori on digitaalisignaaleiden yhteydessä, mutta ainakin eräs asia on pysynyt piilossa. Nimittäin minkä vuoksi väyläliikennettä tarvitsee mitata? Yleisesti tarve ilmaantuu toiminnaltaan tuntemattomien laitteiden yhteydessä. Voimme selvittää prosessorin suorittaman ohjelman, mikäli tunnemme

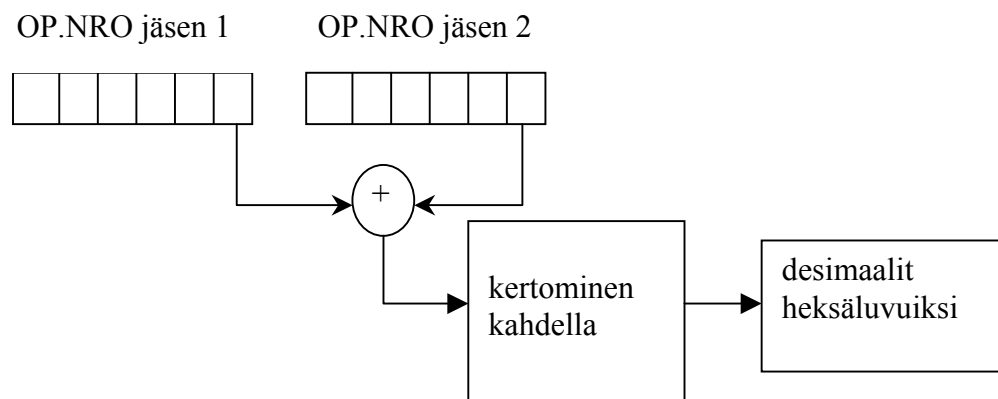
laitteen käskykannan. Voimme tarkastella mitä muistipaikkoja ja I/O-toimintoja prosessori käyttää. Lisäksi mittaustuloksista saamme selville tietoja prosessorin järjestelmätason toteutuksesta (esim. miten prosessori suorittaa tietyn käskyn).

Näitä asioita yritämme selvittää seuraavaksi. Työpaikalla olevasta kansioista löytyy lisää vastauksia tähän kysymykseen. Lue teksti läpi ennen kuin alat tekemään mittauksia.

Mittaukset

Asetukset, joita käytämme mittauksissamme, on määritelty H8S 2144F:n konfiguraatitiedostossa. Kun käynnistät analysaattorin, asetukset ovat automaattisesti oikeat. Mittauksissa sinun tarvitsee käyttää kuvan 7 mukaista ikkunaa mittaustuloksien analysointiin sekä kuvan 6 mukaista ikkunaa triggausehdon määrittelyihin. Lisäksi tarkastelemme hieman *Listing*-ikkunaa (kuva 8).

- M1** Etsi lähin ADD.B-käsky, joka seuraa osoitetta h' 201XX. Osoitteessa oleva XX saadaan, kun ryhmän jäsenten op.numeroista summataan viimeiset numerot ja summa kerrotaan kahdella. Lopuksi saatu luku muutetaan heksäluvuksi. Kuva 9 selvittää tätä vaikeaselkoista asiaa.



Kuva 9. Osoitteen muodostaminen.

Vinkki: tällä muistialueella kaikki add-käskyt ovat muotoa ADD.B Rs, Rd, ja kaikkien käskyjen objektikoodit ovat pituudeltaan 16-bittisiä. Tällaisten ADD-käskyjen operaatiokoodin enitenmerkitsevä tavu on h'08, joka sinun tarvitsee siis löytää dataväylältä. Triggau tapahtuu osoiteväylän arvoon h'1XX. Kun olet hakemasi löytänyt, zoomaa näyttöä niin (mikäli tarvitsee), että näet myös ADD-käskyä seuraavan ja edellisen käskyn.. Tulosta tämä kuvaruutunäkymä ja anna tiedostolle nimi M1.

- M2** Mittaa edellisen tehtävän kaikkien kolmen käskyn hakemiseen kulunut aika. Ota kolme tulostetta, joissa jokaisessa on nähtävissä eri käskyjen dataväylällä kuluttama aika. Tiedostojen nimet voisivat olla vaikka M21, M22 ja M23.
- M3** Ohjelmassa kirjoitetaan osoitteeseen h'21A20 jokin tavun mittainen luku. Etsi tämä kohta, ja aseta mittaustulokset ruudulle niin, että siinä näkyy viisi muistiosoitusta (osoiteväylän arvoa) ennen kirjoitustapahtumaa sekä yksi sen

jälkeen (zoomaus 100 ns/Div voisi olla hyvä). Tulosta kuvaruutunäkymä ja kirjoita tiedostolle nimeksi M3

M4 Jossakin kohtaa ohjelmaamme sijaitsee käsky ADD.B R2L, R0H. Tämän objektikoodi on h' 08A0. Etsi kyseinen käsky asettamalla ensin triggausehto. Siirry sitten *Waveform*-ikkunan laatikkoon *Acq. Control*. Aseta *Sample Period* 96 ns:ksi. Tämän jälkeen siirry *Listing*-ikkunaan vaikka painamalla näppäintä List analysaattorin etupaneelista. Nyt voit käynnistää mittauksen. Tallenna kuvaruutunäkymä, jossa osoite h' 20124 ja sen sisältö näkyvät ylimmäisenä. Käytä kohdistamiseen valkoista pyörrintä.

M5 Aseta *Sample Period* ennalleen. Ohjelmamme viimeinen käsky on BRA (branch always). Tästä käskystä ohjelman suoritus hyppää ohjelman alkuun. Käskyn pituus on 32 bittiä ja sen operaatiokoodi on h' 5800. Etsi tämä. Tulosta kuvaruutunäkymä, jossa operaatiokoodin lisäksi on nähtävillä kaksi muistiosoitusta.

Nyt olemme mittaukset tehneet ja on aika siirtyä jälkiselostuksen pariin. Tulosteita tarvitset seuraavassa osiossa. Sammuta laitteet!

Jälkiselostus

Alla olevien tehtävien kanssa tarvitset prosessorin käskykantaan. Assistentilta saat tällaisen kopiointia varten.

- J1** Esitä kohdan M1 mittaustulokset. Kaikki ruudulla näkyvät dataväylän arvot ovat konekielisiä käskyjä. Käännä käskyt assembly-kielelle käyttämällä käskykantaan apunasi.
- J2** Esitä kohdan M2 mittaustulokset, joissa näkyvät kunkin käskyn hakemiseen kulunut aika. Tee tuloksista lyhyt yhteenveto.
- J3** Esitä kohdan M3 mittaustulokset. Mikä 8 bittinen luku kirjoitetaan osoitteeseen h'21A20 (tarkkaile kirjoitussignaaleita)? Käännä mittaustuloksissa olevat dataväylän arvot assembly-kielelle ja analysoi tämän perusteella MOV-käskyn hakua ja suoritusta.
- J4** Kuvitellaan että olisit määritellyt seuraavat symbolit tehtävässä M4:

h' 0CB0	=	mov.b	R3L, R0H
h' 0A0A	=	inc.b	R2L
h' 08A0	=	add.b	R2L, R0H
h' 100A	=	shll.b	R2L
h' 0CC1	=	mov.b	R4L, R1H

Miltä M4:n tulosteen viisi ensimmäistä riviä näyttäisivät, jos olisit tehnyt symbolimäärittelyt ennen mittausta? Voit hahmotella tämän vaikkalyijykynällä.

- J5** BRA-käskey on siis 32 bittinen. Operaatiokoodia seuraava 16 bittinen osa (operandi) kertoo hypyn etäisyyden. Tämä operandi on kahden komplementti muodossa ja kun se lisätään BRA-käskyn jälkeiseen osoitteeseen, saadaan muistiosoite, josta seuraava suoritettava käskey haetaan. Esitä kohdan M5 mittaustulokset. Kuten tuloksistasi huomaat, ohjelmamme BRA-käskey sijaitsee osoitteissa h' 201CC – h' 201CF, jolloin käskyn jälkeinen osoite on h' 201D0. Suorita tämän muistiosoitteen ja operandin summaus. Vertaa tulosta mittaustuloksiin. Huomioi, että ennen summausta käskyn operandi tulee venyttää osoitteen pituiseksi siten, että etumerkki säilyy.