

SALORA MANAGER KOTITIETOKONE

TEKNINEN KÄSIKIRJA

SISÄLLYSLUETTELO

1. LUKU JOHDANTO

1) Yleistä	5
2) Kirjassa käytetyt termit ja lyhenteet	6

2. LUKU JÄRJESTELMÄ

1) Keskuslaite	8
2) Oheislaitteet	8
3) Prosessori-osajärjestelmä	8
4) Lukumuisti-osajärjestelmä	10
5) Suorasaanti-osajärjestelmä	11
6) Syöttö/tulostus-osajärjestelmä	13
7) Muistikartta	15
8) Monitori	16

3. LUKU MIKROPROSESSORI MSC6502A

1) Yleiskuvaus	20
2) Prosessorin nastatoiminnot	20
3) Aakkosellinen luettelo konekielisistä käskyistä	23
4) Osoitusmuodot	25
5) Käskyt	27
Monitor-Assembler Salora Managerille	38
Ohjelmallisuus	41

4. LUKU PIA-PIIRI MC6821

1) Yleistä	46
2) PIA-piirin liitäntäsignaalit mikroprosessorille	48
3) PIA-piirin oheislaitteväylät	49
4) Piirin toiminta	50

5. LUKU VIDEOPROSESSORI TMS9929A

1) Johdanto	54
2) Liitännät, rekisterit ja näyttötilat	55
2.1. Videoprosessorin liitäntä mikroprosessoriin	55
2.2. Videoprosessori/VRAM-liitäntä	58
2.3. Videoprosessori/TV-liitäntä	60
2.4. Videoprosessorin rekisterit	60
2.5. Tilarekisteri	66
2.6. Näyttötilat	66
3) VRAM-muistiosoitteet	86
4) TMS9929A:n nastat	90
5) Sprite-demo-ohjelma	92
Ohjelmallisuus	94

6. LUKU ÄÄNIGENERAATTORI SN76489AN

1) Äänigeneraattorit	96
2) Kohinageneraattori	97
3) Äänisummeri/lähtöpuskuri	97
4) CPU/SN76489AN-liitäntäpiiri	97
5) Ohjausrekisterit	98
6) Tiedon muodot	99
7) Tiedon siirto	99
8) Piirin nastajärjestys	100

7. LUKU LEVYKEASEMA

1) Toiminta	102
2) Ennalta ehkäisevä huolto	103
3) EXEC-komento	104

8. LUKU LIITTIMET, KYTKENNÄT JA LISÄLAITTEET

1) Keskusyksikön liittimet	106
2) Managerin kytkentä	111
3) Datakasettinauhuri	118
4) Levykeasema	120

LUKU 1. JOHDANTO

1. YLEISTÄ

Tämä 8 lukua käsittävä tekninen käsikirja on tarkoitettu täydentämään Manager-tietokoneen BASIC-käsikirjaa. Kirja poikkeaa teknisen sisältönsä vuoksi melkoisesti BASIC-käsikirjasta eikä sisällä samassa suhteessa ohjelmaesimerkkejä. Pääpaino on 6502A-mikroprosessorijärjestelmän sekä 9929A-videoprosessorin toimintojen kuvaamisessa. Nämä muodostavatkin valtaosan kirjan sisällöstä. Kumpaankin lukuun liittyy ohjelmalistaukset, joiden avulla Managerin konekielistä ohjelmointia sekä sprite-kuvien muodostusta on pyritty selventämään. Kirja on jaettu siten, että jokainen luku muodostaa oman kokonaisuutensa ja lukija voi käydä kirjan läpi missä järjestyksessä tahansa.

Konekielisillä ohjelmanosilla voidaan Managerin keskusprosessorin toimintaa entisestään nopeuttaa. Sivulla 38 esitetty Monitor-Assembler-ohjelma on oiva työkalu niin konekielisten ohjelmien tekoa kuin analysointiakin varten.

Sprite-kuvilla grafiikkaohjelmista saadaan eläviä ja vauhdikkaita. Grafiikkahahmojen muodostus kuva kuvalta on esitetty sivulla 81 ja sprite-demo-ohjelma löytyy sivulta 92. Videoprosessoria käsittelevä luku sisältää lisäksi runsaasti teoreettista tietoa piirien välisistä liitännöistä, rekistereistä sekä eri näyttötiloista.

Kirja on nimestään huolimatta ohjelmoijan käsikirja eikä suinkaan huolto-ohje. Vaikka kirjan lukuun 8 onkin kerätty Managerin sen lisälaitteiden kytkentäkaaviot, emme suosittele lukijaa muuttamaan laitteiden kytkentää emmekä edes yrittämään laitteiden korjausta tai huoltoa. Tällaisissa tapauksissa on varminta kääntyä valtuutetun Salora-huollon puoleen. Kytkentäkaavioilla on pyritty havainnollistamaan tietokoneen toimintaa, jotta esim. liitännävyliä voisi soveltaa käyttäjän omiin tarkoituksiin.

2. LYHYT KATSAUS TÄMÄN KIRJAN TEKSTISSÄ, TAULUKOISSA JA KUVISSA KÄYTETYISTÄ TERMEISTÄ

6502A

6502A on tietokoneen keskusprosessori, jota tekstissä olemme nimittäneet vain mikroprosessoriksi. Lyhenne CPU (Central Processing Unit, keskusprosessori) esiintyy myös useissa taulukoissa, signaaliteissä ja kuvissa. Nimityksillä mikroprosessori, prosessori tai CPU tarkoitetaan siis 6502A-piiriä.

6821

6821 on PIA-piiri (Peripheral Interface Adaptor, oheislaiteliitännöjen sovitin tai lisäpiirien liitännäadapteri). PIA-piiri, PIA, 6821 tai MC6821 tarkoittavat tätä lisäpiirien/oheislaitteiden sovitinta.

9929/9918

Kuvanäyttöprosessori (VDP, Video Display Processor), josta lohkokaavioissa on käytetty myös nimeä Video Data Processor. Tekstissä puhumme yleensä videoprosessorista, mutta taulukoissa ja kuvissa sekä signaaliteiden selostuksissa lyhennettä VDP on käytetty viitattaessa näihin piireihin.

76489

76489AN on ohjelmoitava äänigeneraattori (Programmable Sound Generator). Tekstissä olemme yleensä käyttäneet nimeä äänigeneraattori, mutta esimerkiksi joissakin lohkokaavioissa oleva lyhenne Sound tai Sound Generator tarkoittaa tätä 76489-piiriä.

RAM

RAM-piirit (RAM, Random Access Memory; suorasaantimuisti) muodostavat käyttäjän käytössä olevan tietokoneen muistialueen. Keskuslaitteen RAM-muisti on 32 K, mutta siitä osa muodostaa video-RAM:in (ks. selostus alla). Käyttäjän RAM:in suuruus on 16 K, 16 K-lisämuistilla yhteensä 32 K. Käyttäjän suorasaantimuistipiirejä olemme nimittäneet RAM-piireiksi.

Video-RAM

Kuvaruutunäyttömuisti, josta olemme yleensä käyttäneet nimitystä video-RAM. Taulukoissa ja kuvissa sekä signaaliteissä esiintyvä lyhenne VRAM tarkoittaa myös tätä kuvaruutunäyttömuistia. Sekä RAM- että VRAM-piirien sisältämä data häviää, kun tietokoneesta katkaistaan virta.

ROM

ROM-piirit (ROM, Read Only Memory; lukumuisti) sisältävät tietokoneen toimintaohjeet sekä BASICin ja monitorin. Näihin piireihin käyttäjä ei pääse käsiksi ja niissä oleva tieto säilyy myös silloin, kun tietokonetta ei ole kytketty sähköverkkoon.

LUKU 2. JÄRJESTELMÄ

- 1) Keskuslaite
- 2) Oheislaitteet
- 3) Prosessori-osajärjestelmä
- 4) Lukumuisti-osajärjestelmä
- 5) Suorasaanti-osajärjestelmä
- 6) Syöttö/tulostusosajärjestelmä
- 7) Muistikartta
- 8) Monitori

JÄRJESTELMÄ

1. KESKUSLAITE

Tietokone on rakennettu 6502A-mikroprosessorin ympärille. Piirilevyyn, joka ympäröi tätä prosessoria, on yhdistetty useita I-piirejä. Näitä ovat PIA-piiri 6821, joka ohjaa syöttö-/tulostuslaitteita, 7648PA, ohjelmoitava äänigeneraattori, jossa on neljä musiikkikanavaa, 9929A (videoprosessori) on piiri, joka hoitaa kaikki näyttötoiminnot. Lisäksi piirilevyillä on vielä muistipiirejä, RAM-, ROM- ja muita pienempiä piirejä.

2. OHEISLAITTEET

I Levyasemaohjain

Jotta voit lisätä laitekoonpanoosi levyaseman, sinulla pitää olla levyasemaohjain. Se liitetään keskuslaitteeseen suoraan koneen takana olevaan liittimeen.

Kun levyasemaohjain on paikoillaan, se voi ohjata kahta levyasemaa, joissa käytetään 5 1/4 tuuman levykkeitä. Kumpikin levyasema voi tallettaa 130 Ktavua yhdelle levykeelle.

II 16 K muistinlaajennus

Jos tietokoneeseen asennetaan 16 K muistinlaajennus, on muistialue \$8000 - \$BFFF käyttäjän käytettävissä. Siten käyttäjän muistialue kasvaa 16 Ktavulla.

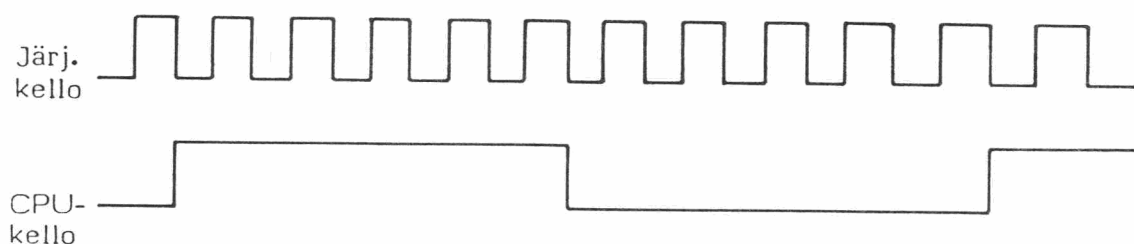
III Coleco-sovitin

Coleco-sovitin avulla voit käyttää Colecon pelimoduuleja Salora Managerissa.

3. PROSESSORI-OSAJÄRJESTELMÄ

Keskuslaitteen sydämenä on 8-bittinen 6502A-mikroprosessori. Se käyttää 16 bitin osoiteavaruutta (64 Ktavun muisti), muistiin syöttö/tulostusta ja toimii joko 1 MHz:n tai 2 MHz:n taajuudella. Salora Managerissa 6502A toimii 1,74437 MHz:n taajuudella. Kellotaajuus on johdettu oskillaattorista, jonka taajuus on jaettu kymmenellä.

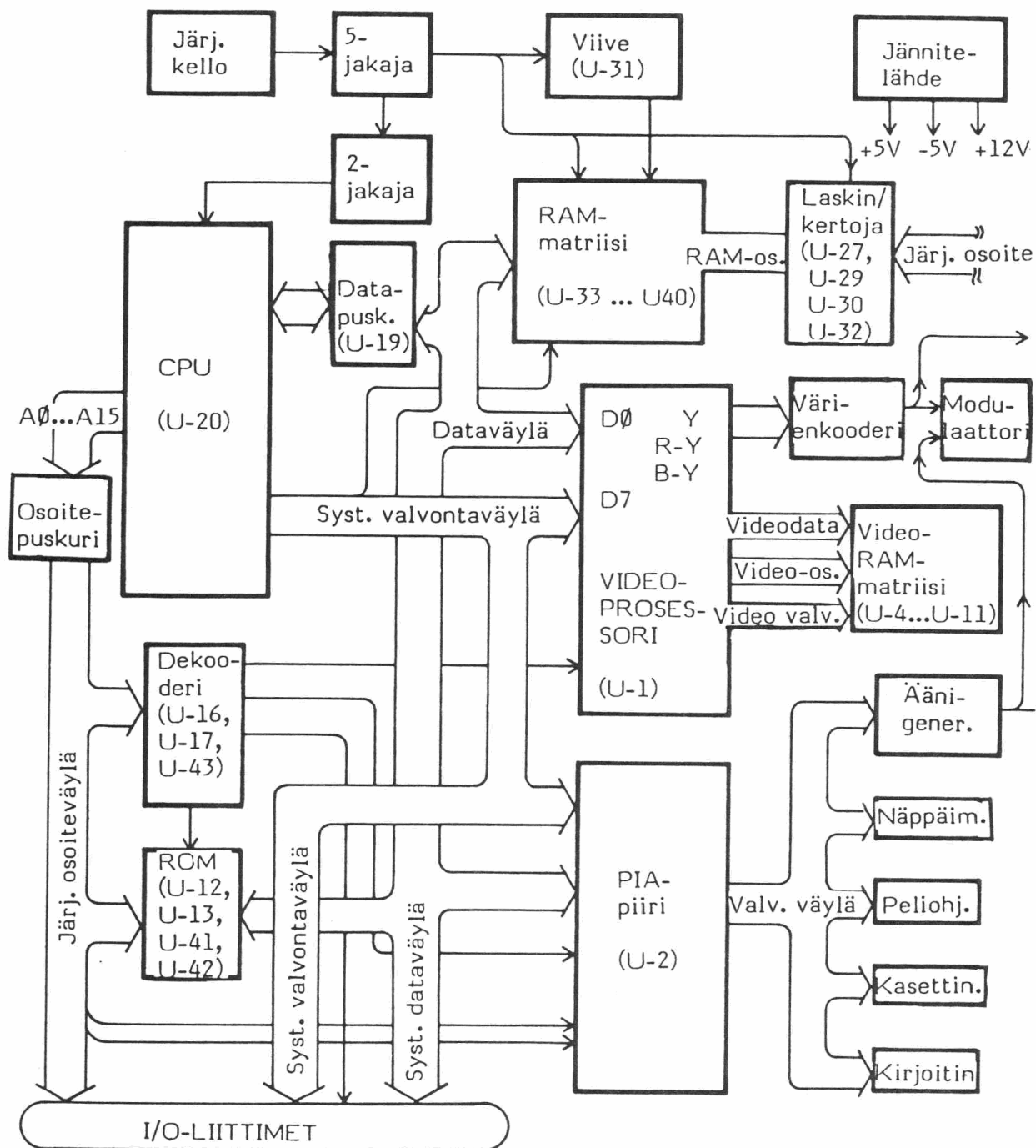
Kuva 2.1. kuvaa järjestelmän kellon ja CPU-kellon ajastuksen välistä suhdetta.



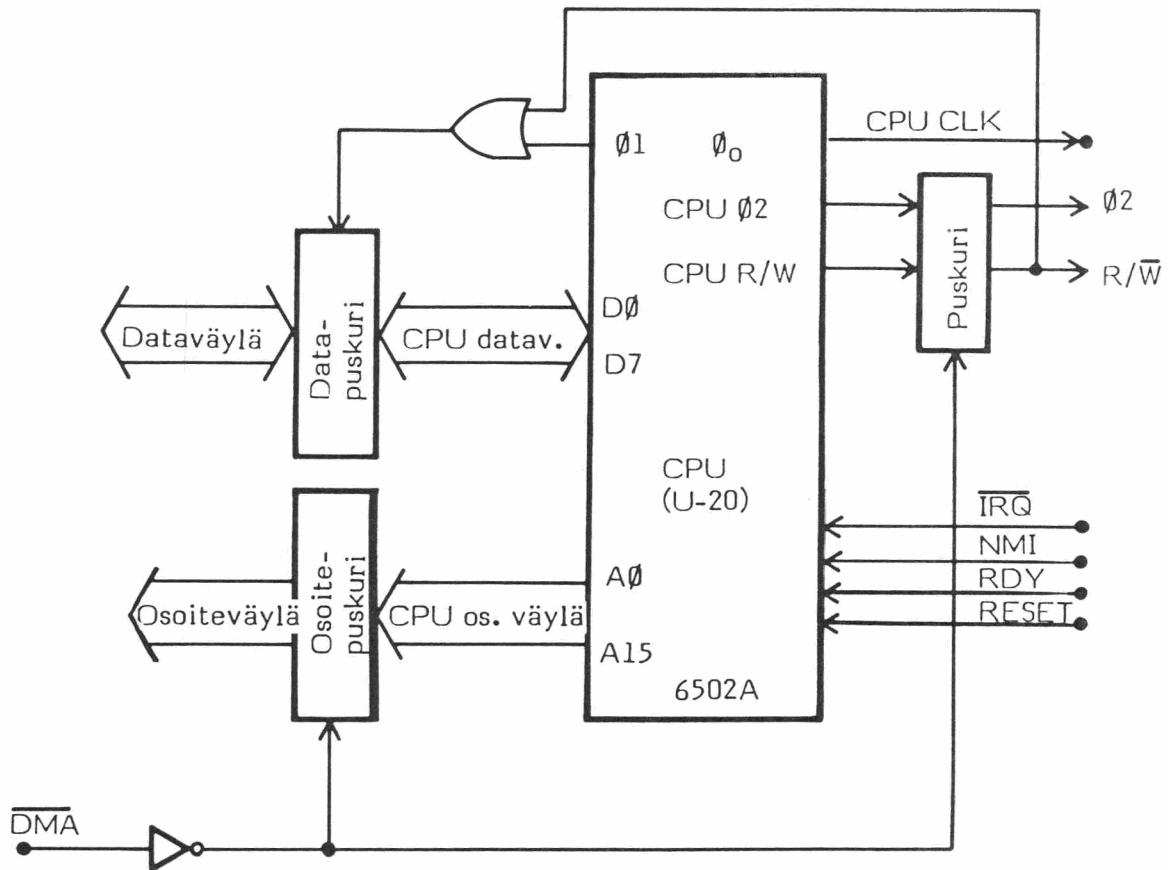
Kuva 2.1.

Osoitepuskurit 74LS244 (U18, U21, U22) ja datapuskuri 74LS245 (U19) on lisätty kytkentään CPU:n ohjaustoimintojen varmistamiseksi.

Kuva 2.2. on kaaviokuva järjestelmästä.



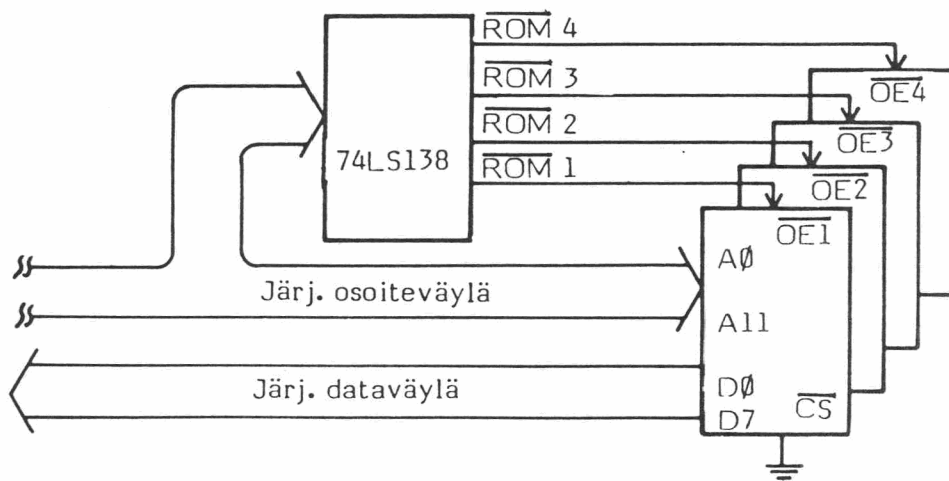
Kuva 2.2.



Kuva 2.3. Prosessorin lohkokaavio

4. LUKUMUISTI-OSAJÄRJESTELMÄ (ROM)

ROM-osajärjestelmään on sijoitettu 16 Ktavun Microsoft Basic sekä monitori. Se on jaettu neljään 4 Ktavun ROM:iin.



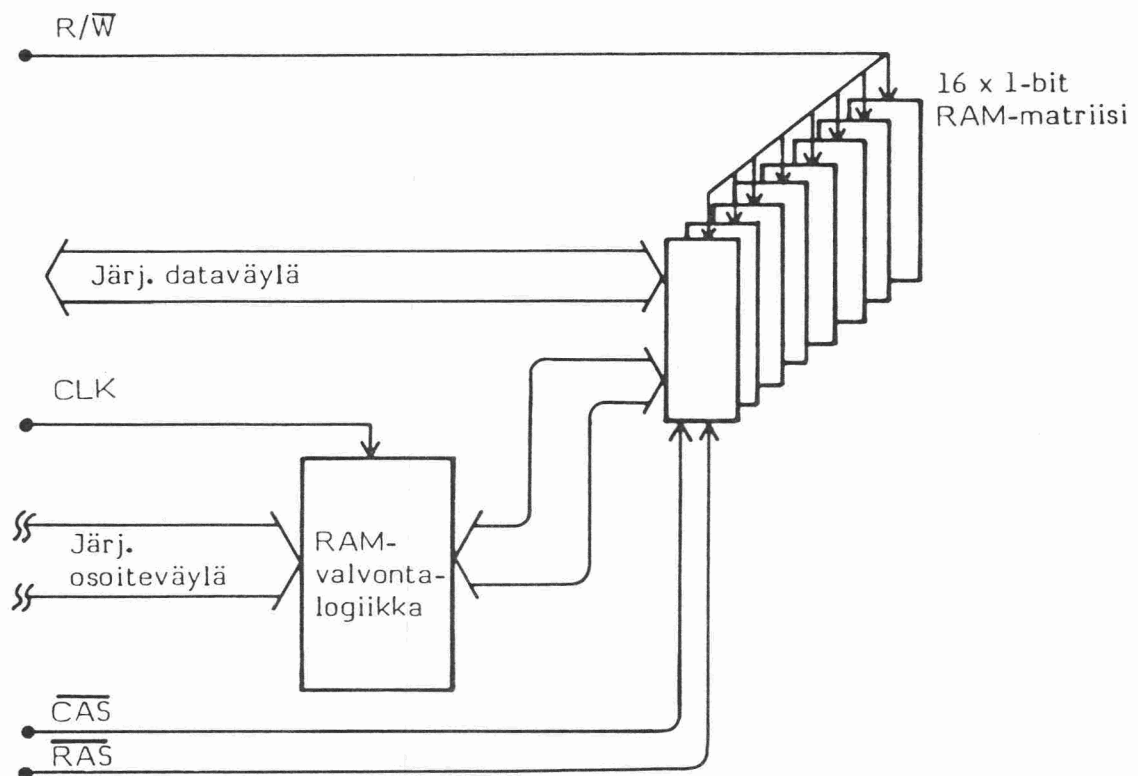
Kuva 2.4. Lukumuistiosajärjestelmä

5. SUORASAANTI-OSAJÄRJESTELMÄ (RAM)

RAM-osajärjestelmä on jaettu kahteen osaan, video- ja keskusyksikkö -RAM-osiin. Kumpikin näistä kahdesta osasta muodostuu kahdeksasta 16 K * 1 bitin dynaamisista RAM-piireistä.

I) Keskusyksikkö -RAM (CPU-RAM)

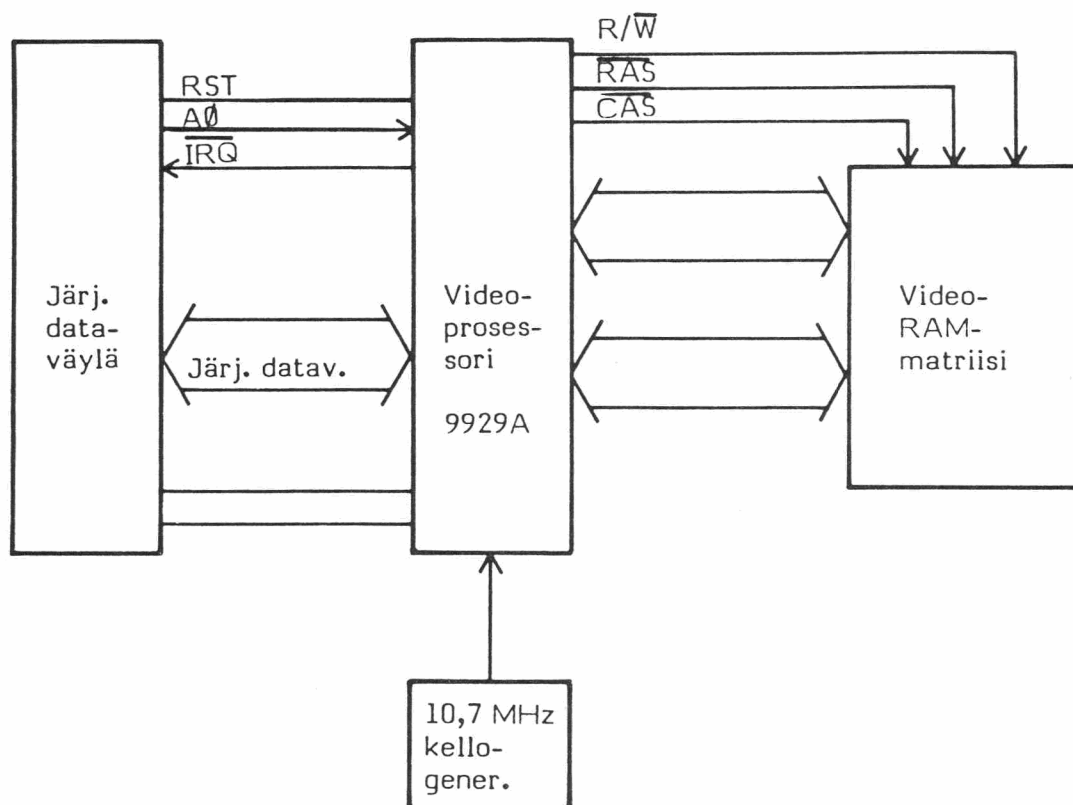
CPU-RAM -osaa varten 16 Ktavun muisti sijaitsee osoitteessa \$4000 - \$7FFF 64 Ktavun muistikartalla. RAM:in osoiteväylään on liitetty RAM:in ohjaus, joka valitsee joko virkistysosoitteen virkistysjakson aikana tai järjestelmän osoitteen järjestelmän luku-/kirjoitusjakson aikana. Looginen laskuri (74LS393) generoi virkistysosoitteen. Kuvassa 2.5. on RAM-osajärjestelmän CPU-osa.



Kuva 2.5.

II Video-RAM-osajärjestelmä (VRAM)

Video-RAM-osan muistikoko on 16 Ktavua, joka ei vie tilaa järjestelmän muistialueesta, joten se ei ole osa käyttäjän muistialueesta. Tämän muistin tarkoituksena on tallettaa ruudulla näkyvä videodata. Kaikki videodata lähetetään keskusyksiköstä videoprosessorin (Video Display Processor VDP, 9929A) kautta. Koska tämä muistialue ei ole käyttäjän muistia, on käytettävä videoprosessorin luku-/kirjoituskytkintä (vrt. kohta 4. "Muistikartta"). Kuva 2.6. havainnollistaa video-RAM:in ja videoprosessorin välistä toimintaa.



Kuva 2.6.

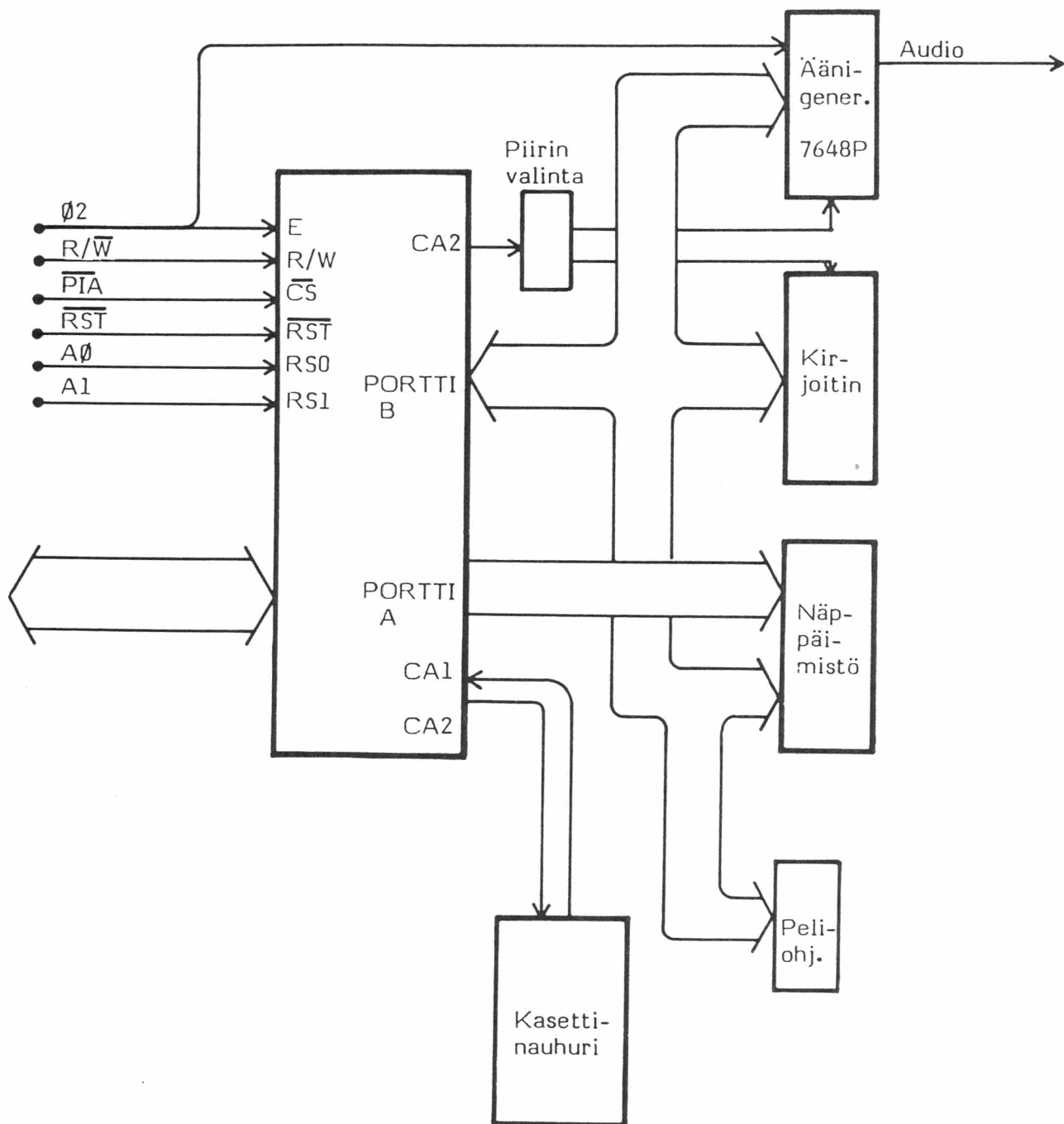
Video-RAM:in muistikartta on seuraavalla sivulla olevan taulukon mukainen.

\$3FFF	Sprite- nimitaulukko
\$3F00	Hahmo- nimitaulukko
\$3C00	Teksti- nimitaulukko
\$3800	Väritaulukko
\$2000	Sprite- generaattori
\$1800	Hahmo- generaattori
\$0000	

6. SYÖTTÖ/TULOSTUS-OSAJÄRJESTELMÄ

Syöttö/tulostus-osajärjestelmä on hyvin tärkeä Salora Managerissa. Koska tämä osa ohjaa kaikkea syöttö/tulostustoimintaa, kuten näppäimistöä, kirjoitinliitettä, äänigeneraattoria, kasettiasemaliitettä sekä peliohjainliitettä. Tämän avulla voit käyttää Salora Manageriasi.

Tämän osan perustana on 6821 (PIA, Peripheral Interface Adaptor, oheislaitesovitin). 6821 on 16-porttinen PIA-piiri ja kaikki syöttö/tulostusportit ovat ohjelmoitavissa (katso lisätietoja PIA-piirin toimintoja kuvaavilta sivuilta). Kuvassa 2.7. esitetään, kuinka oheislaitteet liitetään.



Kuva 2.7.

7. MUISTIKARTTA

\$FFFF	Manager-monitori
\$F800	Managerin BASIC-tulkki
\$C000	RAM-lisämuisti
\$8000	Käyttäjän muisti
\$4000	
\$3100	
\$3001	VDP-rekist. kirj.osoite
\$3000	Video-RAM kirj.osoite
\$2001	VDP-rekist. lukuosoite
\$2000	Video-RAM lukuosoite
\$1003	PIA-kontrollirekist. B
\$1002	PIA-datarekisteri B
\$1001	PIA-kontrollirekist. A
\$1000	PIA-datarekisteri A
\$0200	
\$0000	Järjestelmä-RAM

Keskusmuistista 16 Ktavun ROM (\$C000 - \$FFFF) on varattu Manager-monitorille ja BASIC-tulkille, 256 tavua (\$1000 - \$10FF) ovat erikoistuneet syöttö/tulostuksen hoitoon. Käyttäjän muistialue sijaitsee osoitteessa \$4000 - \$7FFF, kaikkiaan 16 Ktavua. Lisäksi järjestelmä tarvitsee pienen osan tästä käyttäjän muistista (1 Ktavu \$4000 - \$43FF, joka on samanlainen kuin \$0000 - \$03FF). Tämän alueen käytön suhteen tulee olla hyvin varovainen ja sen käyttöä pitäisi välttää, jollei koko järjestelmä ei ole käyttäjälle täysin tuttu.

Kun liität laitteeseen 16 K:n lisämuistin, se sijaitsee automaattisesti osoitteessa \$8000 - \$BFFF. Käyttäjän muistialue laajenee tällöin tällä 16 Ktavulla ollen \$4000 - \$BFFF.

8. JÄRJESTELMÄN YDIN (MONITORI)

Monitori on järjestelmän sielu, jota ilman ei voi tehdä mitään.

Itseasiassa monitori on eräänlainen ohjelmisto, joka hoitaa kaikki järjestelmän toiminnot kuten alustuksen, luvun näppäimistöltä sekä kuvaruutunäytön.

Seuraavalla sivulla on lueteltu osoitteet, joita BASIC-tulkki ja monitori käyttävät.

SIJAINTI		TOIMINTA
Heksadesimaalisena	Desimaalisena	
\$0-\$5	0-5	BASICin alkuvektori
\$A-\$17	10-23	BASICin käyttämät yleiset laskurit/liput
\$20-\$4F	32-79	I/O-monitori ja työmuisti
\$45	69	A-rekisteri
\$46	70	X-rekisteri
\$47	71	Y-rekisteri
\$48	72	Tilarekisteri (status)
\$49	73	Pinorekisteri (stack)
\$50-\$61	80-97	BASICin kirjoittimet
\$62-\$66	98-102	Liukulukutulos
\$67-\$68	103-104	Ohjelman alku
\$69-\$6A	105-106	Muuttujien alun kirjoitin
\$6B-\$6C	107-108	Sääntöiden alun kirjoitin
\$6F-\$70	111-112	Merkkijonomuistin alun kirjoitus
\$71-\$72	113-114	Ei käytössä
\$73-\$74	114-115	Käyttäjän muistin kirjoitus
\$75-\$76	116-117	Suorituksessa oleva rivinumero
\$77-\$78	117-118	Aiempi rivinumero ennen STOP- tai BREAK-käskyä
\$79-\$7A	119-120	Aiempi tekstikirjoitin
\$7B-\$7C	121-122	Viimeisimmän dataauseen rivinumero
\$7D-\$80	123-128	READ- ja INPUT-käskyjen suorittaja
\$81-\$82	129-130	Muuttujanimi
\$83-\$84	131-132	Muuttujan arvo
\$85-\$9C	133-156	Yleinen käyttö
\$9D-\$A3	127-163	Pääliukulukurekisteri
\$A4	164	Päärutiinien yleiskäytössä
\$A5-\$AB	165-171	Toinen liukulukurekisteri
\$AC-\$AE	172-174	Yleinen käyttö
\$AF-\$B0	175-176	Ohjelman loppu
\$B1-\$C8	177-200	Merkkilukurutiini
\$B8-\$B9	184-185	Tekstikirjoitin
\$C9-\$CD	201-205	Ei käytössä
\$D0-\$D5	208-213	Työmuisti
\$D8-\$DF	216-223	Virhetyömuisti

SIJAINTI		TOIMINTA
Heksadesimaalisena	Desimaalisena	
\$E0-\$EF	224-239	Grafiikkakäyttö
\$F0-\$F3	340-242	Ei käytössä
\$F4-\$F8	244-248	Virhekirjoitin
\$100-\$1FF	256-511	Merkkipuskuri
\$200-\$2FF	512-769	Rivipuskuri

LUKU 3. MIKROPROSESSORI MSC6502A

1. Yleiskuvaus
2. Nastatoiminnat
3. Aakkosellinen luettelo konekielisistä käskyistä
4. Osoitusmuodot
5. Käskyt
6. Monitor-assembler Salora Managerille

MIKROPROSESSORI MSC6502

- Yksipuolinen +5 V:n käyttöjännite
- N-kanavainen, pii-hilainen, depletion - load teknologia
- 8-bittinen rinnakkaiskäsitely
- 56 käskyä
- Desimaalinen ja binäärinen aritmetiikka
- 13 osoitustilaa
- Ohjelmoitava pino-osoitin ja vaihtuvamittainen pino
- 1 MHz ja 2MHz toimintataajuus
- Toimintojen limitus

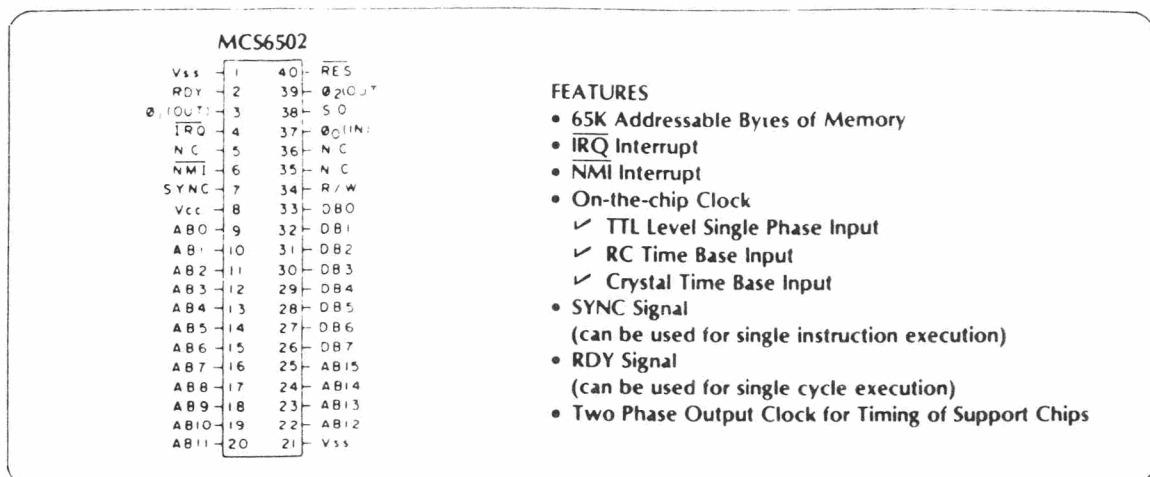
1. YLEISKUVAUS

MCS6500-sarjan mikroprosessorit edustavat täysin ohjelmistosovitteista mikroprosessoriperhettä. Tämä perhe sisältää joukon mikroprosessoreita, jotka tarjoavat valikoiman osoitteellisia muistialueita, keskeytystulo-optioita ja piirin sisäiset kello-oskillaattorit ja ohjaimet. Kaikki MCS6500-ryhmän mikroprosessorit ovat ohjelmistosovitteisia ryhmän sisällä ja väyläsovitteisia M6800-tuotteiden kanssa.

Perhe sisältää viisi mikroprosessoria, joissa kello-oskillaattorit ja ohjaimet sijaitsevat piirilevyillä ja neljä mikroprosessoria, joita ohjaavat ulkoiset kellot. Versioissa, joissa on piirin sisäinen kello, kuten Managerissa käytetty MCS6502A, on yksivaiheiset tulot, kide- ja RC-tulot, jotka huolehtivat ajoituksista. Versiot, joissa on ulkoinen kello, ovat tarkoitettut moniprosessorijärjestelmien sovellutuksiin, joissa maksimijointus on välttämätöntä.

2. NASTATOIMINNAT

MCS6502A:n nastajärjestys on seuraava:



Kuva 1.

2.1. Kellot ($\Phi 1$ ja $\Phi 2$)

MCS6502:n kello on varustettu sisäisellä kellogeneraattorilla. Kellon taajuutta ohjataan ulkoisesti. Yksityiskohtaisemmin tätä käsitellään myöhemmin.

2.2. Osoiteväylä ($A\Phi$ - A15)

Nämä ulostulot ovat TTL-yhteensopivia ja pystyvät ohjaamaan yhtä TTL-kuormaa ja 130 pF kapasitiivista kuormaa.

2.3. Dataväylä ($D\Phi$ - D7)

Dataväylää varten on kahdeksan nastaa. Tämä on kaksisuuntainen väylä, joka siirtää dataa laitteelta oheislaitteelle ja päinvastoin. Ulostulot ovat kolmetilapuskureita ja ne pystyvät ohjaamaan yhtä TTL-kuormaa ja 130 pF kapasitiivista kuormaa.

2.4. DBE-kolmetilaohjaus (Data Bus Enable)

TTL-yhteensopiva tulo sallii kolmetilaisen ulostulopuskurin ulkoisen ohjauksen ja aktivoi 1-tilassa mikroprosessorin väyläohjaimen. Normaalissa toiminnassa DBE-tulo ohjaa vaiheen kaksi ($\Phi 2$) kelloa ja mahdollistaa siten datan tulon mikroprosessorilta ainoastaan vaiheen kaksi aikana. Lukujakson aikana ovat tietoväyläohjaimet sisäisesti estotilassa muodostaen avoimen piirin. Jotta tietoväyläohjaimia voitaisiin ohjata ulkoisesti, DBE:n tulee olla Φ -tilassa.

2.5. Ready (RDY)

Tämä signaali sallii käyttäjän ohjata mikroprosessoria yksi bitti kerrallaan kaikkien muiden jaksojen paitsi kirjoitusjakson aikana. Siirtyminen negatiiviseen tilaan vaiheen yksi aikana pysäyttää ulostulon osoitelinjat ja aiheuttaa vuorossa olevan osoitteen noutamisen. Ehto on voimassa seuraavan vaiheen kaksi ajan, jolloin Ready signaali on Φ . Tämä ominaisuus tekee mahdolliseksi mikroprosessorin liittämisen hitaisiin PROM-muisteihin kuten myös nopeisiin (maks. 2 jaksoa) DMA-muisteihin. Jos Ready-signaali on Φ -tilassa kirjoitusjakson aikana, sitä ei oteta huomioon ennenkuin seuraavan lukutoiminnan aikana.

2.6. Keskeytyspyyntö (IRQ)

Tämä TTL-tasoinen signaali aiheuttaa mikroprosessorissa keskeytyssekvenssin. Mikroprosessori suorittaa loppuun käsittelyssä olevan käskyn ennen keskeytyspyynnön tunnistamista. Tänä aikana tarkastetaan tilakoodirekisterissä oleva keskeytysmaskibitti. Jos keskeytysmaskilippua ei ole asetettu, mikroprosessori aloittaa keskeytyssekvenssin.

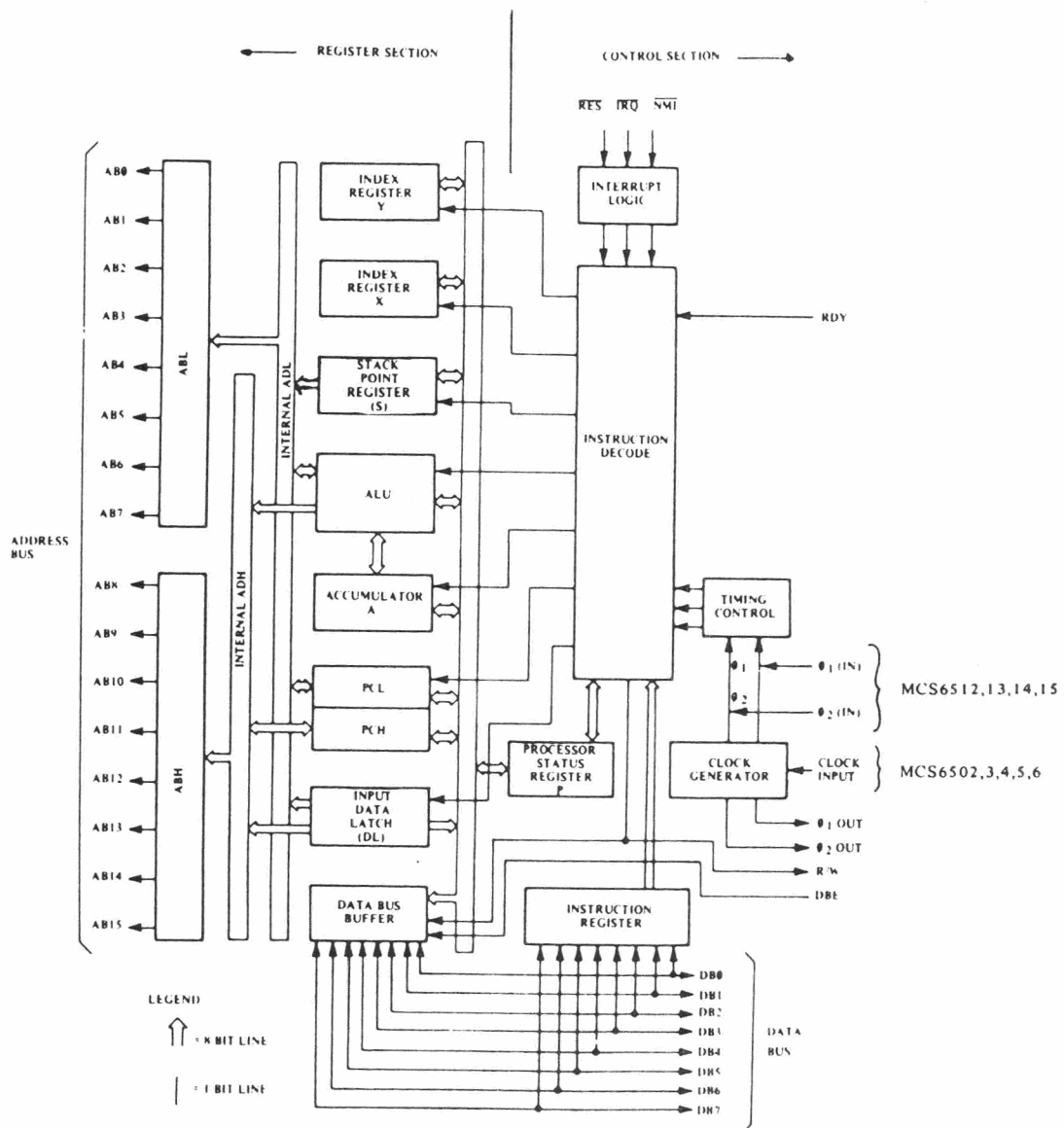
Ohjelmanaskuri ja prosessorin tilarekisteri on tallennettu pinoon. Mikroprosessori asettaa nyt keskeytysmaskilipun 1-tilaan niin ettei lisäkeskeytyksiä tapahdu. Jakson lopussa ohjelmanaskurin Φ -tila ladataan muistipaikasta FFFE ja ohjelmanaskurin 1-tila paikasta FFFF. Nyt ohjelman ohjaus siirtyy muistivektorille, joka sijaitsee näissä osoitteissa.

RDY-signaalin tulee olla 1-tilassa, jotta keskeytys tunnistettaisiin. 3 kohmin ulkoista vastusta on käytettävä, jotta saataisiin oikeat langalliset OR-toiminnot.

2.7. Maskaamaton keskeytys (NMI)

Tämän tulon negatiivinen reuna pyytää maskaamattoman keskeytyssekvenssin muodostamista mikroprosessorissa.

NMI on ehdoton keskeytys. Kun vuorossa oleva käsky on toteutettu, suoritetaan IRQ:lle määritelty toimintasekvenssi riippumatta keskeytysmaskilipun tilasta. Ohjelmalaskuriin ladattu vektori osoite, 0 ja 1, sijaitsevat vastaavasti paikoissa FFFA ja FFFB ja ohjelman ohjaus siirtyy näissä osoitteissa sijaisevalle muistivektorille. Näihin paikkoihin ladatut käskyt aiheuttavat mikroprosessorin haaraantumisen maskaamattomaan keskeytysrutiiniin muistissa. NMI vaatii myös ulkoisen 3 kohmin vastuksen kytkemistä V_{CC} -jännitteeseen, jotta saadaan langalliset OR-toiminnot.



Kuva 2. Mikroprosessorin MCS6502A lohkokaavio

IRQ- ja NMI- tulot ovat laitteiston keskeytyslinjoja, joita näytetään vaiheen kaksi aikana ja jotka aloittavat asiaankuuluvat keskeytysrutiinit vaiheen yksi aikana sen jälkeen kun vuorossa oleva käsky on suoritettu.

2.8. Ylivuotolipun asettaminen (S.O.)

Tulon negatiivinen reuna asettaa ylivuotobitin tilakoodirekisteriin. Signaalia luetaan vaiheen yksi pulssin takareunalla.

2.9. SYNC

Tämän lähtölinjan avulla todetaan ne jaksot, joiden aikana mikroprosessori noutaa operaatiokoodia. SYNC-linja menee 1-tilaan, kun operaatiokoodia noudetaan vaiheen yksi aikana, ja pysyy siinä jakson loppuun asti. Jos RDY-linja on vedetty 0-tilaan kellopulssin vaiheen yksi aikana, jolloin SYNC-linja siirtyi 1-tilaan, pysähtyy prosessori senhetkiseen tilaan ja jää siihen kunnes RDY-linja menee 1-tilaan. Tällä tavalla käytetään SYNC-signaalia ohjaamaan RDY-signaalia siten, että RDY saa aikaan käskyjen toteutuksen yksi kerrallaan.

2.10. Reset

Tuloa käytetään mikroprosessorin käynnistämiseen tai pysäyttämiseen. Kun linja on 0-tilassa, on kirjoittaminen mikroprosessoriin tai sieltä ulos estettynä. Kun tulossa havaitaan positiivinen reuna, mikroprosessori aloittaa välittömästi nollaussekvenssin. Systeemin initialisointi kestää kuusi kellojaksoa ja sen jälkeen asetetaan maskikeskeytyslippu ja mikroprosessori lataa ohjelmalaskurin muistivektoripaikoista FFFC ja FFFD. Tämän jälkeen voi ohjelman ohjaus alkaa.

Kun V_{CC} -jännite saavuttaa arvon 4,75 V käynnistysrutiinissa, reset-linja on pidettävä 0-tilassa ainakin kahden kellojakson ajan. Tämän ajan R/W- ja (SYNC)-signaali pysyvät vakaina.

Kun reset-signaali menee 1-tilaan näiden kahden kellopulssin jälkeen, mikroprosessori jatkaa normaalia edelläkuvattua reset-toimintaa.

3. AAKKOSELLINEN LUETTELO KONEKIELISISTÄ KÄSKYISTÄ

ADC	Add Memory to Accumulator with Carry
AND	"AND" Memory with Accumulator
ASL	Shift Left One Bit (Memory or Accumulator)
BCC	Branch on Carry Clear
BCS	Branch on Carry Set
BEQ	Branch on Result Zero
BIT	Test Bits in Memory with Accumulator
BMI	Branch on Result Minus
BNE	Branch on Result Not Equal
BPL	Branch on Result Plus
BRK	Force Break
BVC	Branch on Overflow Clear
BVS	Branch on Overflow Set

CLC	Clear Carry Flag
CLD	Clear Decimal Mode
CLI	Clear Interrupt Disable Bit
CLV	Clear Overflow Flag
CMP	Compare Memory and Accumulator
CPX	Compare Memory and Index X
CPY	Compare Memory and Index Y
DEC	Decrement Memory by One
DEX	Decrement Index X by One
DEY	Decrement Index Y by One
EOR	Exclusive-OR Memory with Accumulator
INC	Increment Memory by One
INX	Increment Index by One
INY	Increment Index Y by One
JMP	Jump to New Location
JSR	Jump to New Location Saving Return Address
LDA	Load Accumulator with Memory
LDX	Load Index X with Memory
LDY	Load Index Y with Memory
LSR	Shift One Bit Right (Memory or Accumulator)
NOP	No Operation
ORA	OR Memory with Accumulator
PHA	Push Accumulator on Stack
PHP	Push Processor Status on Stack
PLA	Pull Accumulator from Stack
PLP	Pull Processor Status from Stack
ROL	Rotate One Bit Left (Memory or Accumulator)
ROR	Rotate One Bit Right (Memory or Accumulator)
RTI	Return from Interrupt
RTS	Return from Subroutine
SBC	Subtract Memory from Accumulator with Borrow
SEC	Set Carry Flag
SED	Set Decimal Mode
SEI	Set Interrupt Disable Status
STA	Store Accumulator in Memory
STX	Store Index X in Memory
STY	Store Index Y in Memory
TAX	Transfer Accumulator to Index X
TAY	Transfer Accumulator to Index Y
TSX	Transfer Stack Pointer to Index X
TXA	Transfer Index X to Accumulator
TXS	Transfer Index X to Stack Pointer
TYA	Transfer Index Y to Accumulator

Konekielisistä käskyistä on seikkaperäinen luettelo tämän luvun kohdassa 5 sivulla 27 sekä Monitor-Assembler-ohjelma (sivulla 38), jolla voit tehokkaasti ohjelmoida konekielisesti.

4. OSOITUSMUODOT

4.1. Akkuosoitus

Tämä osoitusmuoto edustaa yksitavuista käskyä, joka sisältää akkuun kohdistuvan operaation.

4.2. Välitön osoitus

Välittömässä osoituksessa operandi sisältyy käskyn toiseen tavuun eikä muita muistiin-osoituksia tarvita.

4.3. Absoluuttinen osoitus

Absoluuttisessa osoituksessa käskyn toinen tavu määrittelee osoitteen oikeanpuolisen tavun kolmas vasemmanpuolisen tavun. Siksi absoluuttinen osoitustila mahdollistaa haun koko 65 ktavun osoitteellisesta muistista.

4.4. Nollasivuosoitus

Nollasivukäskyt tekevät mahdolliseksi lyhyen koodi- ja suoritusajat, sillä ne hakevat vain käskyn toisen tavun ja olettamalla yläosoitetavun olevan nolla. Nollasivun huolellisella käytöllä saadaan huomattava koodihyötysuhteen lisäys.

4.5. Indeksoitu nollasivuosoitus (X- ja Y-indeksit)

Tätä osoitusmuotoa käytetään indeksirekisterin yhteydessä, esim. nollasivu X ja nollasivu Y. Todellinen osoite lasketaan lisäämällä toinen tavu indeksirekisterin sisältöön. Koska tämä on eräs nollasivuosoituksen muoto, viittaa toisen tavun sisältö paikkaan sivulla nolla. Muistibittiä ei lisätä muistin kahdeksaan vasemmanpuoliseen bittiin ja sivurajojen ylityksiä ei tapahdu.

4.6. Indeksoitu absoluuttinen osoitus (X- ja Y-indeksit)

Tätä osoitusmuotoa käytetään yhdessä X- ja Y-indeksirekisterien kanssa, esim. absoluuttinen X ja absoluuttinen Y. Todellinen osoite muodostetaan lisäämällä X:n ja Y:n sisällöt osoitteeseen, joka on käskyn toisessa ja kolmannessa tavussa. Tässä osoitustilassa indeksirekisteri voi sisältää indeksin tai laskenta-arvon ja käsky sisältää perusosoitteen. Tällainen indeksointi mahdollistaa viittauksen mihin tahansa paikkaan ja indeksiä voidaan muuntaa monikenttäiseksi ja vähentää siten koodaus- ja suoritusajoja.

4.7. Viitattu osoitus

Tässä osoitusmuodossa on osoite, joka sisältää operandin, sijoitettu epäsuorasti käskyn operaatiokoodiin.

4.8. Suhteellinen osoitus

Suhteellista osoitusta käytetään vain haarautumiskäskyjen yhteydessä ja se asettaa

ehtohaaralle paikan. Käsken toinen tavu on operandi, johon on lisätty sen poikkeama ohjelmanaskurin kahdeksan alemman bitin sisällöstä, kun askuri on asetettu seuraavalle käskenlle. Poikkeaman suuruus on -128 ... +127 tavua seuraavasta käskenstä lähtien.

4.9. Indeksoitu epäsuora osoitus

Tässä osoituksessa käsken toinen tavu lisää X-indeksirekisterin sisältöön ja muistibittiä ei oteta huomioon. Tämän lisäyksen jälkeen kohdistuu osoitus sille sivun nolla muistipaikalle, jonka sisältö on todellisen osoitteen kahdeksan oikeanpuolisinta bittiä. Seuraava nollasivun muistipaikka sisältää todellisen osoitteen kahdeksan vasemmanpuolisinta bittiä. Molempien muistipaikkojen, jotka määräävät osoitteen oikean- ja vasemmanpuolisimmat tavut, on oltava sivulla nolla.

4.10. Epäsuora indeksoitu osoitus

Tässä osoituksessa käsken toinen tavu osoittaa sivulla nolla olevaan muistipaikkaan. Tähän muistipaikan sisältöön lisätään Y-indeksirekisterin sisältö ja tuloksena saadaan todellisen osoitteen kahdeksan oikeanpuolisinta bittiä. Tämän lisäyksen muistibitti lisää seuraavan sivun nolla muistipaikan sisältöön ja tuloksena saadaan todellisen osoitteen kahdeksan vasemmanpuolisinta bittiä.

4.11. Absoluuttinen epäsuora osoitus

Käsken toinen tavu sisältää muistipaikan kahdeksan oikeanpuolisinta bittiä. Muistipaikan kahdeksan vasemmanpuolisinta bittiä sisältyvät käsken kolmanteen tavuun. Tarkoin määritellyn muistipaikan sisältö on todellisen oikeanpuolisin tavu. Seuraava muistipaikka sisältää todellisen osoitteen vasemmanpuolisimman tavun, joka on ladattu 16-bittiseen ohjelmanaskuriin.

5. KÄSKYT

ADC Add memory to accumulator with carry
Lisää muistin sisältö akkuun muistinumeroa käyttäen

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	ADC \nrightarrow Oper	69	2	2
Nollasivu	ADC Oper	65	2	3
Nollasivu, X	ADC Oper, X	75	2	4
Absoluuttinen	ADC Oper	6D	3	4
Absoluuttinen, X	ADC Oper, X	7D	3	4*
Absoluuttinen, Y	ADC Oper, Y	79	3	4*
(Epäsuora, X)	ADC (Oper, X)	61	2	6
(Epäsuora), Y	ADC (Oper), Y	71	2	5*

AND "AND" memory with accumulator
Suorita "AND" toiminta muistin ja akun kesken

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	AND \nrightarrow Oper	29	2	2
Nollasivu	AND Oper	25	2	3
Nollasivu, X	AND Oper, X	35	2	4
Absoluuttinen	AND Oper	2D	3	4
Absoluuttinen, X	AND Oper, X	3D	3	4*
Absoluuttinen, Y	AND Oper, Y	39	3	4*
(Epäsuora, X)	AND (Oper, X)	21	2	6
(Epäsuora), Y	AND (Oper), Y	31	2	5*

ASL Shift Left One Bit (Memory or Accumulator)
Siirry yksi bitti vasemmalle (muisti tai akku)

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Akku	ASL A	0A	1	2
Nollasivu	ASL Oper	06	2	5
Nollasivu, X	ASL Oper, X	16	2	6
Absoluuttinen	ASL Oper	0E	3	6
Absoluuttinen, X	ASL Oper, X	1E	3	7

BCC Branch on Carry Clear
Haarautu muistinumeron nollauksesta

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Suhteellinen	BCC Oper	90	2	2*

BCS **Branch on carry set**
Haarautu muistinumeron asetuksesta

Osoitus- muoto	Assembler-kielinen muoto		OP koodi	Tavujen lukum.	Jaksojen lukum.
Suhteellinen	BCS	Oper	F0	2	2*

BIT **Test bits in memory with accumulator**
Testaa muistin bitit akun kanssa

Osoitus- muoto	Assembler-kielinen muoto		OP koodi	Tavujen lukum.	Jaksojen lukum.
Nollasivu Absoluuttinen	BIT	Oper	24	2	3
	BIT	Oper	2C	3	4

BMI **Branch on result minus**
Haarautu miinustuloksella

Osoitus- muoto	Assembler-kielinen muoto		OP koodi	Tavujen lukum.	Jaksojen lukum.
Suhteellinen	BMI	Oper	30	2	2*

BNE **Branch on result not zero**
Haarautu kun tulos ei ole nolla

Osoitus- muoto	Assembler-kielinen muoto		OP koodi	Tavujen lukum.	Jaksojen lukum.
Suhteellinen	BNE	Oper	D0	2	2*

BPL **Branch on result plus**
Haarautu plustuloksella

Osoitus- muoto	Assembler-kielinen muoto		OP koodi	Tavujen lukum.	Jaksojen lukum.
Suhteellinen	BPL	Oper	10	2	2*

BRK **Force Break**
Pakotettu keskeytys

Osoitus- muoto	Assembler-kielinen muoto		OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	BRK		00	1	7

BVC **Branch on overflow clear**
Haarautu ylivuodon nollauksella

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Suhteellinen	BVC Oper	50	2	2*

BVS **Branch on overflow set**
Haarautu ylivuodon asetuksella

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Suhteellinen	BVS Oper	70	2	2*

CLC **Clear carry flag**
Nollaa muistinumerolippu

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	CLC	18	1	2

CLD **Clear decimal mode**
Nollaa desimaalitila

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	CLD	D8	1	2

CLI **Clear interrupt disable bit**
Nollaa keskeytyksen estobitti

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	CLI	58	1	2

CLV **Clear overflow flag**
Nollaa ylivuotolippu

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	CLV	B8	1	2

CMP Compare memory and accumulator
Vertaa muistia ja akkua

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	CMP ∇ Oper	C9	2	2
Nollasivu	CMP Oper	C5	2	3
Nollasivu, X	CMP Oper, X	D5	2	4
Absoluuttinen	CMP Oper	CD	3	4
Absoluuttinen, X	CMP Oper, X	DD	3	4*
Absoluuttinen, Y	CMP Oper, Y	D9	3	4*
(Epäsuora, X)	CMP (Oper, X)	C1	2	6
(Epäsuora), Y	CMP (Oper), Y	D1	2	5*

CPX Compare Memory and index X
Vertaa muistia ja indeksiä X

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	CPX ∇ Oper	E0	2	2
Nollasivu	CPX Oper	E4	2	3
Absoluuttinen	CPX Oper	EC	3	4

CPY Compare memory and index Y
Vertaa muistia ja indeksiä Y

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	CPY ∇ Oper	C0	2	2
Nollasivu	CPY Oper	C4	2	3
Absoluuttinen	CPY Oper	CC	3	4

DEC Decrement memory by one
Vähennä muistin arvosta yksi

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Nollasivu	DEC Oper	C6	2	5
Nollasivu, X	DEC Oper, X	D6	2	6
Absoluuttinen	DEC Oper	CE	3	6
Absoluuttinen, X	DEC Oper, X	DE	3	7

DEX Decrement index X by one
Vähennä indeksin X arvosta yksi

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	DEX	CA	1	2

DEY Decrement index Y by one
Vähennä indeksin Y arvosta yksi

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	DEY	88	1	2

EOR "Exclusive - Or" memory with accumulator
"Ex-Or" muisti akun yhteydessä

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	EOR H Oper	49	2	2
Nollasivu	EOR Oper	45	2	3
Nollasivu, X	EOR Oper, X	55	2	4
Absoluuttinen	EOR Oper	4D	3	4
Absoluuttinen, X	EOR Oper, X	5D	3	4*
Absoluuttinen, Y	EOR Oper, Y	59	3	4*
(Epäsuora, X)	EOR (Oper, X)	41	2	6
(Epäsuora), Y	EOR (Oper), Y	51	2	5*

INC Increment memory by one
Lisää muistin arvoon yksi

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Nollasivu	INC Oper	E6	2	5
Nollasivu, X	INC Oper, X	F6	2	6
Absoluuttinen	INC Oper	EE	3	6
Absoluuttinen, X	INC Oper, X	FE	3	7

INX Increment Index X by one
Lisää indeksin X arvoon yksi

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	INX	E8	1	2

INY Increment Index Y by one
Lisää indeksin Y arvoon yksi

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	INY	C8	1	2

JMP **Jump to new location**
Hyppää uuteen paikkaan

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Absoluuttinen	JMP Oper	4C	3	3
Indirect	JMP (Oper)	6C	3	5

JSR **Jump to new location saving return address**
Hyppää uuteen paikkaan säilyttäen paluusoite

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Absoluuttinen	JSR Oper	20	3	6

LDA **Load accumulator with memory**
Lataa muisti akkuun

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	LDA \nrightarrow Oper	A9	2	2
Nollasivu	LDA Oper	A5	2	3
Nollasivu, X	LDA Oper, X	B5	2	4
Absoluuttinen	LDA Oper	AD	3	4
Absoluuttinen, X	LDA Oper, X	BD	3	4*
Absoluuttinen, Y	LDA Oper, Y	B9	3	4*
(Epäsuora, X)	LDA (Oper, X)	A1	2	6
(Epäsuora), Y	LDA (Oper), Y	B1	2	5*

LDX **Load index X with memory**
Lataa muisti indeksiin X

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	LDX \nrightarrow Oper	A2	2	2
Nollasivu	LDX Oper	A6	2	3
Nollasivu, Y	LDX Oper, Y	B6	2	4
Absoluuttinen	LDX Oper	AE	3	4
Absoluuttinen, Y	LDX Oper, Y	BE	3	4*

LDY Load index Y with memory
Lataa muisti indeksiin Y

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	LDY $\#$ Oper	A0	2	2
Nollasivu	LDY Oper	A4	2	3
Nollasivu, X	LDY Oper, X	B4	2	4
Absoluuttinen	LDY Oper	AC	3	4
Absoluuttinen, X	LDY Oper, X	BC	3	4*

LSR Shift right one bit (memory or accumulator)
Siirry yksi bitti oikealle (muisti tai akku)

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Akku	LSR A	4A	1	2
Nollasivu	LSR Oper	A6	2	5
Nollasivu, X	LSR Oper, X	56	2	6
Absoluuttinen	LSR Oper	4E	3	6
Absoluuttinen, X	LSR Oper, X	5E	3	7

NOP No operation
Ei toimintaa

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	NOP	EA	1	2

ORA "OR" memory with accumulator
"OR" -toiminta akun ja muistin välillä

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	ORA $\#$ Oper	09	2	2
Nollasivu	ORA Oper	05	2	3
Nollasivu, X	ORA Oper, X	15	2	4
Absoluuttinen	ORA Oper	0D	3	4
Absoluuttinen, X	ORA Oper, X	1D	3	4*
Absoluuttinen, Y	ORA Oper, Y	19	3	4*
(Epäsuora, X)	ORA (Oper, X)	01	2	6
(Epäsuora), Y	ORA (Oper), Y	11	2	5*

PHA Push accumulator on stack
Siirrä akun tila pinoon

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	PHA	48	1	3

PHP Push processor status on stack
Siirrä prosessorin tila pinoon

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	PHP	08	1	3

PLA Pull accumulator from stack
Ota akun sisältö pinosta

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	PLA	68	1	4

PLP Pull processor status from stack
Ota prosessorin tila pinosta

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	PLP	28	1	4

ROL Rotate one bit left (memory or accumulator)
Pyöritä yksi bitti vasemmalle (muisti tai akku)

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Akku	ROL A	2A	1	2
Nollasivu	ROL Oper	26	2	5
Nollasivu, X	ROL Oper, X	36	2	6
Absoluuttinen	ROL Oper	2E	3	6
Absoluuttinen, X	ROL Oper, X	3E	3	7

ROR Rotate one bit right (memory or accumulator)
Pyöritä yksi bitti oikealle (muisti tai akku)

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Akku	ROR A	6A	1	2
Nollasivu	ROR Oper	66	2	5
Nollasivu, X	ROR Oper, X	76	2	6
Absoluuttinen	ROR Oper	6E	3	6
Absoluuttinen, X	ROR Oper, X	7E	3	7

RTI Return from interrupt
Paluu keskeytyksestä

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	RTI	40	1	6

RTS Return from subroutine
Paluu alirutiinista

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	RTS	60	1	6

SBC Subtract memory from accumulator with borrow
Vähennä muistin arvo akusta lainausta käyttäen

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Välitön	SBC \nrightarrow Oper	E9	2	2
Nollasivu	SBC Oper	E5	2	3
Nollasivu, X	SBC Oper, X	F5	2	4
Absoluuttinen	SBC Oper	ED	3	4
Absoluuttinen, X	SBC Oper, X	FD	3	4*
Absoluuttinen, Y	SBC Oper, Y	F9	3	4*
(Epäsuora, X)	SBC (Oper, X)	E1	2	6
(Epäsuora), Y	SBC (Oper), Y	F1	2	5*

SEC Set carry flag
Aseta muistinumerolippu

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	SEC	38	1	2

SED Set decimal mode
Aseta desimaalitila

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	SED	F8	1	2

SEI **Set interrupt disable status**
Aseta keskeytyksen estotila

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	SEI	78	1	2

STA **Store accumulator in memory**
Tallenna akku muistiin

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Nollasivu	STA Oper	85	2	3
Nollasivu, X	STA Oper, X	95	2	4
Absoluuttinen	STA Oper	8D	3	4
Absoluuttinen, X	STA Oper, X	9D	3	5
Absoluuttinen, Y	STA Oper, Y	99	3	5
(Epäsuora, X)	STA (Oper, X)	81	2	6
(Epäsuora), Y	STA (Oper), Y	91	2	6

STX **Store index X in memory**
Tallenna indeksi X muistiin

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Nollasivu	STX Oper	86	2	3
Nollasivu, Y	STX Oper, Y	96	2	4
Absoluuttinen	STX Oper	8E	3	4

STY **Store index Y in memory**
Tallenna indeksi Y muistiin

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Nollasivu	STY Oper	84	2	3
Nollasivu, X	STA Oper, X	94	2	4
Absoluuttinen	STY Oper	8C	3	4

TAX **Transfer accumulator to index X**
Siirrä akku indeksiin X

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	TAX	AA	1	2

TAY Transfer accumulator to index Y
Siirrä akku indeksiin Y

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	TAY	A8	1	2

TYA Transfer index T to accumulator
Siirrä indeksi T akkuun

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	TYA	98	1	2

TSX Transfer stack pointer to index X
Siirrä pino-osoitin indeksiin X

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	TSX	BA	1	2

TXA Transfer index X to accumulator
Siirrä indeksi X akkuun

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	TXA	8A	1	2

TXS Transfer index X to stack pointer
Siirrä indeksi X pino-osoittimeen

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Viitattu	TXS	9A	1	2

BEQ Branch on result zero
Haarautu nollatuloksella

Osoitus- muoto	Assembler-kielinen muoto	OP koodi	Tavujen lukum.	Jaksojen lukum.
Suhteellinen	BEQ Oper	F0	2	2*

MONITOR-ASSEMBLER SALORA MANAGERILLE

Monitor-Assembler on tehokas ohjelmapaketti assembler-ohjelmointiin. Ohjelma on kahtena Basic-ohjelmana, jotka syöttävät konekielisen ohjelman muistiin. Näitten kahden Basic-ohjelman jälkeen konekielinen ohjelma sijoittuu muistiin alueelle 6000_H - 6A00_H (24576 - 24832), josta se voidaan siirtää konekielisenä ohjelmana nauhurille seuraavaalla käskyllä

BSAVE \$A6000, \$LA000

Levyllä tallennettaessa käsky on

MSAVE MONITOR, A\$6000, L\$A000

Konekielinen ohjelma käynnistetään seuraavalla kutsulla **CALL 24576** tai suoraan nauhalta käskyllä **BRUN** tai levyltä käskyllä **MRUN MONITOR**.

Managerin keskusyksikön lisäksi tarvitaan levyasema tai kasettinahuri.

Kun ohjelma on ladattu ja sen suoritus alkaa, ollaan MONITORI-tilassa, minkä voi havaita merkistä **Dr**ivin alussa.

YLEISTÄ OHJELMAN KÄYTÖSTÄ

Monitor-ohjelmassa voidaan käyttää ainoastaan heksadesimaalilukuja eli kunkin luvun arvo on välillä 0 ... F. Nelinumeroinen osoite ja kaksinumeroinen luku erotetaan toisistaan kaksoispisteellä. Jos annetaan pidempiä lukuja, ohjelma käyttää niistä vain neljä viimeistä tai jos annetaan kuutta merkkiä lyhyempi luku, ohjelma täyttää neljä paikkaa vasemmalta lukien nolilla.

Yhdellä rivillä voidaan antaa kuinka monta komentoa tahansa, niin kauan kun 255 merkkiä riittää. Komennot suoritetaan peräkkäin. Kun löytyy virheellinen merkki, monitoriohjelma antaa "piip"-merkkiä. Kukin komentorivi täytyy lopettaa painamalla RETURN näppäintä. Seuraavissa komennossa osoite XXXX voidaan jättää pois. Silloin käytetään viimeksi hyödynnettyä arvoa.

KOMENNOT

Q:

Paluu BASIC:iin. Tällöin kaikki merkit poistetaan ja voidaan ajaa kaikkia käskyjä ja ohjelmia läpi Monitorin tuhoutumatta. Käskyllä & kutsutaan Monitor uudelleen. Jos se ei onnistu, on toinen ohjelma sen poistanut ja Monitor on ladattava uudelleen.

Muistipaikkojen osoittaminen

1) XXXX.YYYY

Muistipaikkojen XXXX:stä lähtien YYYY:hyn asti tulostetaan heksadesimaalilukuna siten, että rivillä on kahdeksan arvoa.

2) aa<XXXX.YYYYYZ

Muistipaikkojen sisältö XXXX:stä YYYYY:hyn tulostetaan aa-merkkiä kullakin rivillä (aa on heksadesimaalinen luku). Tyypiksi Z ilmoitetaan jokin seuraavista:

- a ASCII
- b binäärinen
- d desimaalinen
- h heksadesimaalinen

Muistipaikkojen muuttaminen

XXXX:aa bb cc ...

Käskyn osoittama muistipaikan arvo muutetaan aa:ksi ja seuraavia bb:ksi, cc:ksi jne.

Disassembler

XXXXXL

Muistipaikasta XXXX alkaen tulostetaan muistipaikkojen sisältö heksadesimaalisena ja käännetään disassemblerilla.

Laskeminen

XX+YY tai XX-YY

Laskee heksadesimaalisia lukuja yhteen tai vähentää.

Muistitilan siirtäminen

ZZZZ<XXXX.YYYYM

Muistialue, jonka alkuosoite on XXXX ja loppuosoite YYYY, siirretään muistiosoitteeseen ZZZZ:sta alkaen.

Jos alkuosoite ZZZZ, johon muistialue siirretään, sijaitsee siirrettävän muistialueen sisällä, niin XXXX ja ZZZZ välillä oleva muistialue kertaantuu koko siirrettävälle muistialueelle. Näin voidaan esimerkiksi täyttää muistialue halutuilla merkeillä (XXXX:n ja ZZZZ:n välillä olevilla).

Muistitilan vertailu

ZZZZ<XXXX.YYYYV

Vertaillaan muistialuetta XXXX:stä YYYYY:hyn ZZZZ:sta alkavaan muistialueeseen. Eroavuudet tulostetaan.

Rekisterien osittaminen ja muuttaminen

X

Keskusyksikön 6502:n rekisterit tulostetaan näytölle. Näitä rekistereitä voidaan muuttaa antamalla heti X-komennon jälkeen: AA XX YY SS PP, missä AA jne ovat rekisterien uusia arvoja.

Konekielisen ohjelman suorittaminen

XXXXG

Ohjelman suoritus aloitetaan XXXX:stä lähtien sen hetkisillä rekisterin arvoilla.

U

U-kirjaimella voidaan suorittaa ohjelma, joka alkaa 03F8:sta.

Debugger-toiminta

XXXXS

S-kirjaimella suoritetaan konekielinen ohjelma alkaen XXXX:stä käsky kerrallaan. Käsky esitetään assembly-muodossa ja rekisterien tilat esitetään operaation jälkeen.

XXXXT

T aiheuttaa jatkuvan S-toiminnan. Tämä voidaan keskeyttää ainoastaan ohjelmassa olevalla BRK-käskyllä.

Molempien toimintojen kohdalla BRK-käsky johtaa MONITORI-rutiiniin, johon BRK-käskyn osoite ja rekisterien sisältö osoittavat.

Myös normaalissa konekielisessä ohjelman suorituksessa oleva BRK aiheuttaa hypyn tähän rutiiniin. Tällä voidaan siis asettaa pysähtymiskohdat ohjelmaan.

Tulostus kirjoittimelle

Kaikien toimintojen tulostus voidaan ohjata myös kirjoittimelle kirjoittamalla käskyn eteen ja jälkeen P-kirjain. Tällöin kirjoittimen on oltava toimintavalmis, muuten voidaan jatkaa vain painamalla RESET-näppäintä.

Mikäli komentoa ei ole päätetty johonkin kirjaimeen, on P erotettava välilyönnillä käskystä virhetoiminnan välttämiseksi.

Assembler

Assembleria kutsutaan Monitorilta huutomerkillä (!) ja siitä poistutaan /-merkillä. Tämä rivi-assembleri tuntee kaikki 6502:n komennot.

XXXX:YYY

XXXX:

Voidaan ohittaa välilyönnillä, jolloin käytetään seuraavana olevaa osoitetta.

Käsky ja arvo annetaan siinä muodossa kuin L-käskyllä disassembloituna on saatu aikaan. Mikäli ei tapahdu virheitä, korvaantuu osoitettu rivi disassembler-rivillä ja saatua tulosta voidaan kokeilla. Mikäli syötössä on jokin virhe, tulostuu riville ! ja kuullaan piip-äänimerkki. Osoite ei ole tällöin muuttunut ja käsky voidaan syöttää uudelleen. On mahdollista \$-merkin avulla toteuttaa kaikki monitorikäskyt assemblerista. '!' -merkillä päästään assembler-kääntäjästä takaisin Monitor-ohjelmaan.

```

10 GOTO 300
100 READ AX
105 IF AX = "END" THEN RETURN
110 HX = LEFT$(AX,1):QX = RIGHT$(AX,1)
120 H = ASC (HX):Q = ASC (QX)
130 IF H > = 65 THEN H = H - 55: GOTO 150
140 H = H - 48
150 IF Q > = 65 THEN Q = Q - 55: GOTO 170
160 Q = Q - 48
170 A = H * 16 + Q
180 POKE I,A
190 I = I + 1: GOTO 100
300 I = 25455
310 GOSUB 100
1000 DATA 04,20,54,30,0D,80,04,90,03,22,54,33,0D,80,04,"90"
1010 DATA 04,20,54,33,0D,80,04,90,04,20,54,3B,0D,80,04,"90"
1020 DATA 00,22,44,33,0D,C8,44,00,11,22,44,33,0D,C8,44,"A9"
1030 DATA 01,22,44,33,0D,80,04,90,01,22,44,33,0D,80,04,"90"
1040 DATA 26,31,87,9A,00,21,81,82,00,00,59,4D,91,92,86,"4A"
1050 DATA 85,9D,AC,A9,AC,A3,A8,A4,D9,00,D8,A4,A4,00,1C,"8A"
1060 DATA 1C,23,5D,8B,1B,A1,9D,8A,1D,23,9D,8B,1D,A1,00,"29"
1070 DATA 19,AE,69,A8,19,23,24,53,1B,23,24,53,19,A1,00,"1A"
1080 DATA 5B,5B,A5,69,24,24,AE,AE,A8,AD,29,00,7C,00,15,"9C"
1100 DATA 6D,9C,A5,69,29,53,84,13,34,11,A5,69,23,A0,D8,"62"
1110 DATA 5A,48,26,62,94,88,54,44,C8,54,68,44,E8,94,00,"B4"
1120 DATA 08,84,74,B4,28,6E,74,F4,CC,4A,72,F2,A4,8A,00,"AA"
1130 DATA A2,A2,74,74,74,72,44,68,B2,32,B2,00,22,00,1A,"1A"
1140 DATA 26,26,72,72,88,C8,C4,CA,26,48,44,"44"
2000 DATA A2,C8,20,DB,05,68,85,2C,68,85,2D,A2,08,BD,33,"08"
2010 DATA 95,3C,CA,D0,F8,A1,3A,F0,42,A4,2F,C9,20,F0,59,"C9"
2020 DATA 60,F0,45,C9,4C,F0,5C,C9,6C,F0,59,C9,40,F0,35,"29"
2030 DATA 1F,49,14,C9,04,F0,02,B1,3A,99,3C,00,88,10,F8,"20"
2040 DATA FE,09,4C,3C,00,85,45,68,48,0A,0A,0A,30,03,6C,"FE"
2050 DATA 03,28,20,08,0A,68,85,3A,68,85,3B,20,90,05,20,"E4"
2060 DATA 07,4C,52,0A,18,68,85,48,68,85,3A,68,85,3B,A5,"2F"
2070 DATA 20,63,06,84,3B,18,90,14,18,20,61,06,AA,98,48,"8A"
2080 DATA 48,A0,02,18,B1,3A,AA,88,B1,3A,86,3B,85,3A,80,"F3"
2090 DATA A5,2D,48,A5,2C,48,20,5E,08,A9,45,85,40,A9,00,"85"
2100 DATA 41,A2,FB,A9,A0,20,BD,08,BD,41,07,20,BD,08,A9,"BD"
2110 DATA 20,BD,08,B5,4A,20,AA,08,E8,30,E8,A2,02,20,55,"06"
2120 DATA A4,48,98,0A,A8,A9,AD,90,03,BD,41,08,20,BD,08,"E8"
2130 DATA E0,08,D0,EE,60,18,A0,01,B1,3A,20,63,06,85,3A,"98"
2140 DATA 38,B0,89,20,09,0A,38,B0,85,EA,EA,4C,2E,08,4C,"20"
2145 DATA 08,C1,D8,D9,D0,D3,CE,D6,A0,C2,C4,C9,DA,C3,E6,"42"
2150 DATA D0,02,E6,43,A5,3C,C5,3E,A5,3D,E5,3F,E6,3C,D0,"02"
2160 DATA E6,3D,60,A9,8D,D0,5B,A4,3D,A6,3C,20,5E,08,20,"4B"
2170 DATA 06,A0,00,A9,BA,4C,BD,08,A5,3C,09,07,85,3E,A5,"3D"
2180 DATA 85,3F,A5,3C,29,07,D0,03,20,62,08,A9,A0,20,BD,"08"
2190 DATA B1,3C,20,AA,08,20,4F,08,90,E8,60,4A,90,EA,4A,"4A"
2200 DATA A5,3E,90,02,49,FF,65,3C,48,A9,BD,20,BD,08,68,"48"
2210 DATA 4A,4A,4A,4A,20,B5,08,68,29,0F,09,B0,C9,BA,90,"02"
2220 DATA 69,06,4C,D6,CB,C6,34,F0,AF,CA,D0,16,C9,BA,D0,"CB"
2230 DATA 85,31,A5,3E,91,40,E6,40,D0,02,E6,41,60,A4,34,"B9"
2240 DATA FF,01,85,31,60,A2,01,B5,3E,95,42,95,44,CA,10,"F7"
2250 DATA 60,B1,3C,91,42,20,49,08,90,F7,60,B1,3C,D1,42,"F0"
2260 DATA 1C,20,62,08,B1,3C,20,AA,08,A9,A0,20,BD,08,A9,"A8"
2270 DATA 20,BD,08,B1,42,20,AA,08,A9,A9,20,BD,08,20,49,"08"
2280 DATA 90,D9,60,20,35,09,A9,14,48,20,DB,05,20,5E,06,"85"
2290 DATA 3A,84,3B,68,3B,E9,01,D0,EF,60,8A,F0,07,85,3C,"95"
2300 DATA 3A,CA,10,F9,60,4C,F1,C4,20,35,09,20,FE,09,6C,"3A"
2310 DATA 00,4C,E1,07,C6,34,20,35,09,4C,4D,07,4C,F8,03,"A9"
2320 DATA 02,D0,0A,A9,00,F0,06,A9,80,30,02,A9,01,85,41,"A5"
2330 DATA 42,85,40,20,62,08,A9,A0,20,BD,08,A0,00,B1,3C,"85"
2340 DATA 43,A5,41,30,38,F0,13,4A,B0,22,A5,43,09,80,C9,"A0"
2350 DATA B0,02,A9,AE,20,BD,08,38,B0,17,A2,08,A9,B0,06,"43"
2360 DATA 90,02,A9,B1,20,BD,08,CA,D0,F2,F0,05,A5,43,20,"AA"
2370 DATA 08,20,4F,08,B0,06,C6,40,D0,BC,F0,B3,60,A0,02,"A2"
2380 DATA 00,A5,43,D9,D6,09,90,08,F9,D6,09,85,43,E8,D0,"F1"
2390 DATA 8A,09,B0,20,BD,08,88,10,E6,30,D6,01,0A,64,B9,"00"
2400 DATA 02,09,80,99,00,02,C8,D0,F5,60,AD,00,03,C9,A5,"F0"
2410 DATA 08,A9,A5,8D,00,03,4C,0C,CC,4C,7D,CC,20,C0,08,"68"
2420 DATA 68,D0,54,A5,48,48,A5,45,A6,46,A4,47,28,60,85,"45"
2430 DATA 86,46,84,47,08,68,85,48,BA,86,49,D8,60,A9,80,"20"

```

```

2440 DATA 77,FE,A9,04,20,77,FE,A9,90,20,77,FE,A0,00,E8,"D0"
2450 DATA FD,C8,D0,FA,20,54,FE,60,20,09,0A,A9,4C,8D,F5,"03"
2460 DATA A9,33,8D,F6,03,A9,0A,8D,F7,03,A9,90,8D,FE,BF,"A9"
2470 DATA 07,8D,FF,BF,20,18,0A,20,5E,08,A9,A3,20,8D,08,"20"
2480 DATA 44,ED,20,B9,0A,20,D9,09,C8,20,98,0A,84,34,A0,"16"
2490 DATA 88,30,E1,D9,BE,0A,D0,F8,20,AF,0A,A4,34,4C,64,"0A"
2500 DATA A2,03,0A,0A,0A,0A,0A,26,3E,26,3F,CA,10,F8,A5,"31"
2510 DATA D0,06,B5,3F,95,3D,95,41,E8,F0,F3,D0,06,A2,00,"86"
2520 DATA 3E,86,3F,B9,00,02,C8,49,B0,C9,0A,90,D3,69,88,"C9"
2530 DATA FA,B0,CD,60,B9,EA,0A,48,B9,D4,0A,48,A5,31,A0,"00"
2540 DATA 84,31,60,EA,EE,F1,ED,EF,EC,E9,A6,A4,06,95,05,"00"
2550 DATA 9A,93,A7,C6,99,DA,DB,DD,E1,3F,56,4B,4E,F5,50,"E4"
2560 DATA D7,D7,EB,DF,1D,42,C9,D7,D7,F6,C3,59,5D,61,65,"09"
2570 DATA 09,09,09,08,09,09,08,08,08,08,09,09,04,08,08,"09"
2580 DATA 08,09,09,09,"09"
3000 DATA "END"

```



```

10 GOTO 300
100 READ AX
105 IF AX = "END" THEN RETURN
110 HX = LEFT$(AX,1):QX = RIGHT$(AX,1)
120 H = ASC (HX):Q = ASC (QX)
130 IF H > = 65 THEN H = H - 55: GOTO 150
140 H = H - 48
150 IF Q > = 65 THEN Q = Q - 55: GOTO 170
160 Q = Q - 48
170 A = H * 16 + Q
180 POKE I,A
190 I = I + 1: GOTO 100
300 I = 24576
310 GOSUB 100
350 I = 24870
360 GOSUB 100
370 I = 24786: GOSUB 100
1000 DATA A9,00,A2,DC,20,6F,E9,A9,00,85,E0,A9,68,85,E1,"A0"
1004 DATA 00,A2,07,B1,E0,20,63,E9,C8,D0,F8,E6,E1,CA,10,"F3"
1008 DATA EA,A9,00,A2,F8,20,6F,E9,A9,00,85,E0,A9,6C,85,"E1"
1010 DATA A0,00,A2,03,B1,E0,20,63,E9,EA,EA,EA,EA,CA,"10"
1015 DATA F3,EA,A9,22,8D,3A,03,A9,F8,8D,3B,03,A9,00,85,"E0"
1020 DATA 85,E2,A9,61,85,E1,A9,04,85,E3,A0,26,A2,06,B1,"E0"
1025 DATA 91,E2,C8,D0,F9,E6,E1,E6,E3,CA,10,F2,4C,26,04,"00"
1030 DATA "END"
1040 DATA 4C,33,0A,E9,81,4A,D0,14,A4,3F,A6,3E,D0,01,88,"CA"
1050 DATA 8A,18,E5,3A,85,3E,10,01,C8,98,E5,3B,D0,79,A4,"2F"
1060 DATA B9,3D,00,91,3A,88,10,F8,38,AD,3A,03,E9,50,8D,"3A"
1065 DATA 03,AD,3B,03,E9,00,8D,3B,03,20,DB,05,20,5E,06,"84"
1068 DATA 3B,85,3A,4C,CD,04,20,AF,0A,A4,34,20,9B,0A,84,"34"
1070 DATA A0,16,88,30,4F,D9,BE,0A,D0,F8,C0,10,D0,E8,A5,"31"
1073 DATA A0,00,C6,34,20,C0,08,4C,CD,04,A5,3D,20,9C,05,"AA"
1075 DATA BD,0D,07,C5,42,D0,16,BD,CD,06,C5,43,D0,0F,A5,"44"
1080 DATA A4,2E,C0,9D,D0,03,4C,29,04,C5,2E,F0,91,C6,3D,"D0"
1085 DATA D9,E6,44,C6,35,F0,D3,A4,34,88,98,AA,20,55,06,"A9"
1087 DATA A1,20,BD,08,20,18,0A,20,5E,08,A9,A1,20,BD,08,"20"
1090 DATA 44,ED,20,B9,0A,20,D9,09,C8,AD,01,02,C9,A0,F0,"1D"
1093 DATA C8,C9,A4,D0,03,4C,71,04,C9,A3,D0,03,4C,33,0A,"88"
1095 DATA 20,98,0A,C9,93,D0,C2,8A,F0,BF,20,38,09,A9,03,"85"
1100 DATA 3D,20,7E,05,0A,E9,BE,C9,C2,90,AE,0A,0A,A2,04,"0A"
1105 DATA 26,42,26,43,CA,10,F8,C6,3D,F0,F4,10,E4,A2,05,"20"
1110 DATA 7E,05,84,34,DD,C1,06,D0,13,20,7E,05,DD,C7,06,"F0"
1115 DATA 0D,BD,C7,06,F0,07,C9,A4,F0,03,A4,34,18,88,26,"44"
1120 DATA E0,03,D0,0D,20,98,0A,A5,3F,F0,01,E8,86,35,A2,"03"
1123 DATA 88,86,3D,CA,10,C9,A5,44,0A,0A,05,35,C9,20,80,"06"
1125 DATA A6,35,F0,02,09,80,85,44,84,34,B9,00,02,C9,BB,"F0"
1128 DATA 04,C9,8D,D0,80,4C,90,04,B9,00,02,C8,C9,A0,F0,"F8"
1130 DATA 60,90,04,4A,4A,4A,4A,29,0F,60,A6,3A,A4,3B,20,"66"
1132 DATA 08,20,53,06,A1,3A,A8,4A,90,09,6A,B0,10,C9,A2,"F0"
1134 DATA 0C,29,87,4A,AA,BD,6F,06,20,87,05,D0,04,A0,80,"A9"
1136 DATA 00,AA,BD,B3,06,85,2E,29,03,85,2F,98,29,8F,AA,"98"
1138 DATA A0,03,E0,8A,F0,0B,4A,90,08,4A,4A,09,20,88,D0,"FA"
1140 DATA C8,88,D0,F2,60,20,90,05,48,B1,3A,20,AA,08,A2,"01"
1145 DATA 20,55,06,C4,2F,C8,90,F1,A2,03,C0,04,90,F2,68,"A8"
1150 DATA B9,CD,06,85,2C,B9,0D,07,85,2D,A9,00,A0,05,06,"2D"
1155 DATA 26,2C,2A,88,D0,F8,69,BF,20,BD,08,CA,D0,EC,20,"53"
1158 DATA 06,A4,2F,A2,06,E0,03,F0,1C,06,2E,90,0E,BD,C0,"06"
1160 DATA 20,BD,08,BD,C6,06,F0,03,20,BD,08,CA,D0,E7,60,"88"
1165 DATA 30,E7,20,AA,08,A5,2E,C9,E8,B1,3A,90,F2,20,63,"06"
1170 DATA AA,E8,D0,01,C8,98,20,AA,08,8A,4C,AA,08,A2,03,"A9"
1180 DATA A0,20,BD,08,CA,D0,F8,60,38,A5,2F,A5,2F,A4,3B,"AA"
1190 DATA 10,01,88,65,3A,90,01,C8,"60"
1200 DATA "END"
1300 DATA EE,5E,5E,5E,5E,76,6F,6E,5E,52,2E,53,74,6F,72,"68"
1310 DATA 5E,5E,5E,5E,"EF"
1400 DATA "END"

```


LUKU 4. PIA-PIIRI MC6821, LISÄPIIRIEN LIITÄNTÄADAPTERI

- 1. Yleistä**
- 2. PIA-piirin liitäntäsignaalit mikroprosessorille**
- 3. PIA-piirin oheislaitteväylät**
- 4. Toiminta**

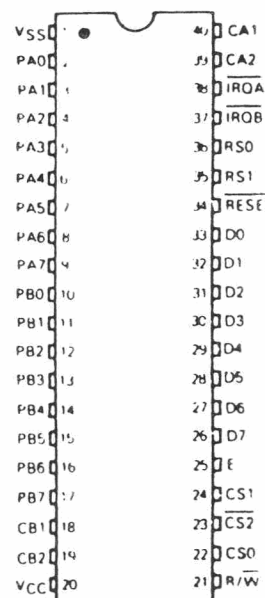
PIA-PIIRI MC6821 (LISÄPIIRIEN LIITÄNTÄADAPTERI)

1. YLEISTÄ

PIA-piirin MC6821 avulla lisäpiirit liitetään mikroprosessoriin kahden 8-bittisen data-väylän avulla. PIA-piirin toimintoja ohjataan mikroprosessorin avulla. Kaikki liitäntäväylät voidaan ohjelmoida toimimaan joko tulo- tai lähtöväylinä ja kaikki valvonta-/keskeytysväylät voidaan ohjelmoida toimimaan yhdessä tai useammassa valvontatilassa. Piirin toimintoja ovat:

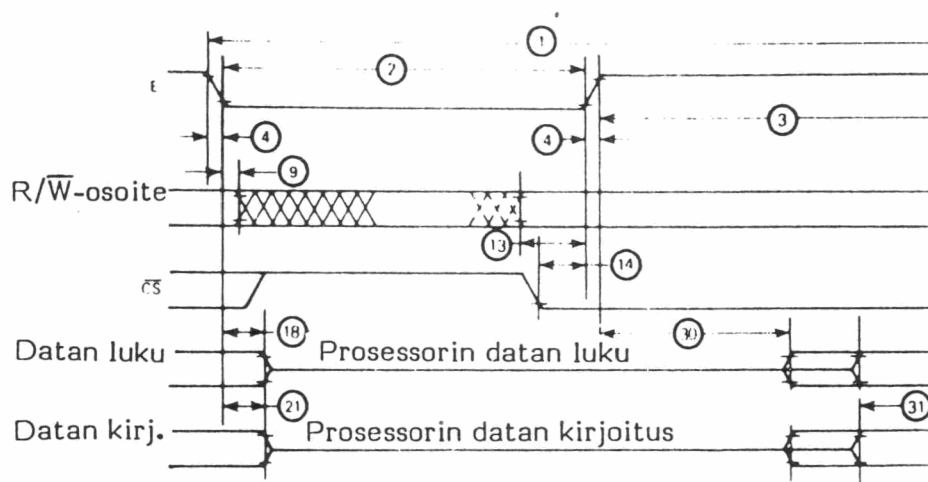
- 8-bittinen kaksisuuntainen dataväylä mikroprosessorin kanssa tapahtuvaa kommunikointia varten
- Kaksi 8-bittistä kaksisuuntaista väylää lisäpiirien liitäntöjä varten
- Kaksi ohjelmoitavaa valvontarekisteriä
- Neljä erikseen ohjattavaa keskeytystuloväylää, joista kahta voidaan käyttää liitäntäpiirien valvontalähtönä
- "Kättely" (kuittauslogiikka) liitäntäpiirien tulo-/lähtötoimintojen kanssa
- Suurimpedanssinen 3-tila ja suorat transistorien ohjauslinjat
- Ohjelmallisesti valvottava keskeytys ja keskeytyksen perustoiminnot
- CMOS-ohjausmahdollisuus A-puolen liitäntälinjoilla
- Kaksi TTL-ohjausmahdollisuutta kaikilla A- ja B-puolen puskureilla
- TTL-yhteensopiva
- Staattinen toiminta

PIA-piirin nastajärjestys on seuraava:

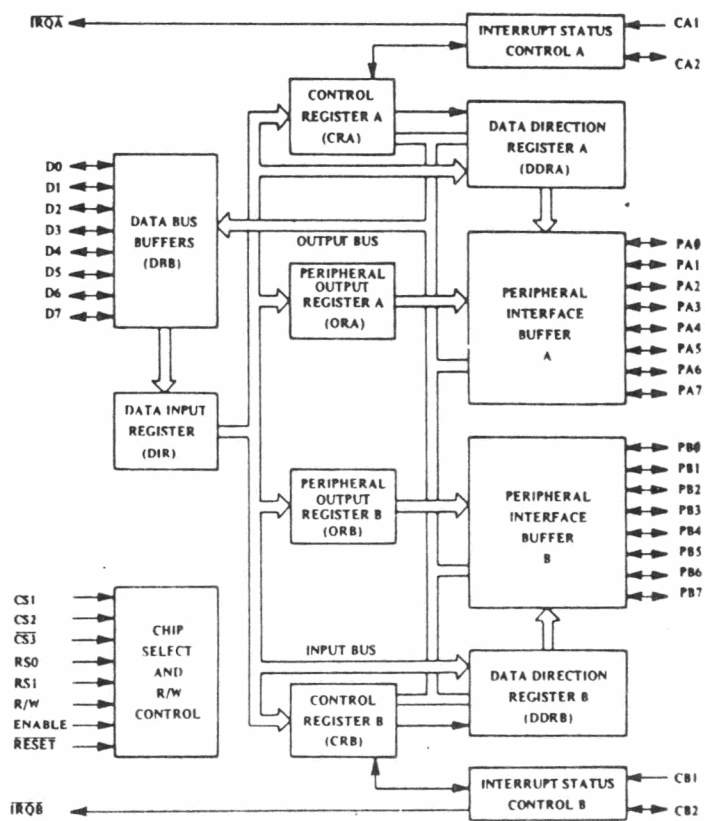


Kuva 1

Väylän ajoituskaavio on esitetty kuvassa 2 ja lohkoakaavio kuvassa 3.



Kuva 2



Kuva 3

2. PIA-PIIRIN LIITÄNTÄSIGNAALIT MIKROPROSESSORILLE

PIA liittyy mikroprosessoriin 8-bittisellä kaksisuuntaisella dataväylällä, kolmella piirinvalintalinjalla, kahdella rekisterinvalintalinjalla, kahdella keskeytyslinjalla, luku-/kirjoituslinjalla sekä enable- ja reset-linjoilla.

2.1. Kaksisuuntainen data (D_0 - D_7)

Kaksisuuntaisten datalinjojen D_0 - D_7 avulla siirretään dataa mikroprosessorin ja PIA-piirin välillä. Dataväylän lähdöt ovat 3-tilalähtöjä, jotka ovat suurimpedanssisessa tilassa (OFF) muulloin paitsi mikroprosessorin suorittaessa PIA-piiriltä lukutoiminnon. Luku-/kirjoituslinja on lukutilassa (ylhäällä), kun PIA on valittu lukutoimintoon.

2.2. Enable (E)

Enable-pulssi (E) on ainoastaan PIA-piirin ajoituspulssi. Kaikkien muiden signaalien ajoituksia verrataan E-pulssin nouseviin ja laskeviin reunoihin.

2.3. Luku/kirjoitus (R/\overline{W})

Tämä signaali muodostetaan mikroprosessorissa ja sen tarkoitus on valvoa dataväylällä tapahtuvaa datan siirron suuntaa. PIA-piirin luku/kirjoituslinjan ollessa alhaalla data siirtyy mikroprosessorilta PIA:n datapuskureihin E-signaalin mukaisesti. Luku/kirjoituslinjan ollessa ylhäällä PIA siirtää dataa väylälle. PIA:n lähtöpuskurit ovat toiminnassa kun oikea osoite sekä E-pulssi ovat käytössä.

2.4. \overline{RESET}

Aktiivisena alhaalla olevaa \overline{RESET} -linjaa käytetään kaikkien PIA-piirin rekisteribittien resetointiin, ts. asettamiseen loogiseksi nollaksi (alas). Linjaa voidaan käyttää virran kytkemishetkellä tapahtuvaan resetointiin sekä resetointiin järjestelmän toiminnan aikana.

2.5. Piirin valinnat (CS_0 , CS_1 ja $\overline{CS_2}$)

Näitä kolmea tulosignaalia käytetään PIA:n valintaan. CS_0 :n ja CS_1 :n täytyy olla ylhäällä ja $\overline{CS_2}$:n alhaalla laitetta valittaessa. Datan siirto suoritetaan tämän jälkeen Enable- ja luku/kirjoitussignaalien valvonnassa. Piirinvalintalinjan täytyy pysyä muuttumattomana E-pulssin keston ajan. Laitteen valinta peruuntuu, jos mikä tahansa piirinvalinnoista on ei-aktiivisessa tilassa.

2.6. Rekisterien valinnat (RS_0 ja RS_1)

Näitä kahta rekisterien valintalinjaa käytetään PIA-piirissä olevien lukuisten rekisterien valintaan. Rekisterien valintalinjoja käytetään yhdessä sisäisen valintarekisterin kanssa tietyn rekisterin - joko kirjoitettavan tai luettavan - valintaan. Rekisterin ja piirin valintalinjan tulee pysyä muuttumattomana E-pulssin keston ajan sekä luku- että kirjoitusjaksossa.

2.7. Keskeytyksen kysely (\overline{IRQA} ja \overline{IRQB})

Aktiivitilassa alhaalla olevat keskeytyksen kyselylinjat \overline{IRQA} ja \overline{IRQB} toimivat mikroprosessorin keskeyttämiseksi joko suoraan tai keskeytysten priorisointipiirin välityksellä. Linjat ovat "open-drain"-nastoja (ei kuormaa). Kummallakin keskeytyksen kysely -linjalla on kaksi sisäistä lippubittiä, jotka voivat aiheuttaa linjan menemisen alas. Kukin lippubitti on yhteydessä tietyn lisälaittekeskeytyslinjan kanssa. PIA on myös varustettu neljällä keskeytyksen mahdollistavalla bitillä, joita voidaan käyttää tiettyjen lisälaitteilta tulevien keskeytysten estoon.

3. PIA-PIIRIN OHEISLAITEVÄYLÄT

PIA-piirillä on kaksi 8-bittistä dataväylää ja neljä keskeytys-/valvontalinjaa jotka toimivat oheislaitteiden liitännöissä.

3.1. Alueen A oheislaitedata (PA \emptyset - PA7)

Kukin oheislaitteen datalinja voidaan ohjelmoida toimimaan joko tulona tai lähtönä. Tämä toteutetaan asettamalla "1"-bitti vastaavaan datan suunnan rekisteriin (lohko-kaaviossa Data Direction Register) niille linjoille, joiden tulee olla lähtöjä. "0"-bitti datan suunnan rekisterissä saa datalinjan toimimaan tulona. Mikroprosessorin oheislaitteen datan lukutoiminnon aikana tuloiksi ohjelmoituilla linjoilla oleva data siirtyy välittömästi mikroprosessorin dataväylälle. Tulotilassa näillä linjoilla oleva sisäinen ylösvetovastus toimii maksimissaan 1,5 kertaisena TTL-kuormana.

Lähtörekisterissä A oleva data ilmestyy datalinjoille, jotka on ohjelmoitu tuloiksi. Rekisteriin kirjoitettu looginen "1" aikaansaa vastaavan datalinjan nousemisen ylös, kun "0" puolestaan vetää linjan alas. Mikroprosessori voi lukea lähtörekisterissä olevan datan, kun vastaavat linjat on ohjelmoitu lähdöiksi. Jännite oheislaitelinjalla on yli 2 V loogiselle "1"-lähdölle ja alle 0,8 V loogiselle "0":lle.

3.2. Alueen B oheislaitedata (PB \emptyset - PB7)

Alueen B I/O-portit toimivat monessa suhteessa samoin kuin alueen A portit. Linjojen ohjelmointi toimimaan joko lähtönä tai tulona on edellä esitetyn kaltainen. Niillä on 3-tilaominaisuus, joten linjat ovat suurimpedanssisessa tilassa toimiessaan tuloina. Lähtöinä nämä linjat ovat TTL-yhteensopivia ja voivat toimia Darlington-transistorikytkimien ohjauksena. Täten voidaan toteuttaa esimerkiksi releiden, lamppujen jne yksinkertainen ohjaus.

3.3. Keskeytystulo/oheislaitteiden valvontalinjat (CA1, CA2, CB1, CB2)

Neljällä keskeytystulo/oheislaittevalvontalinjalla voidaan toteuttaa lukuisia oheislaitteiden valvontatoimintoja ja huomattavasti lisätä oheislaitteporttien PA \emptyset - PA7, PB \emptyset - PB7 tehoa.

3.3.1. Keskeytystulot (CA1 ja CB1)

Oheislaitetulot CA1 ja CB1 ovat ainoastaan keskeytystuloja, jotka asettavat valvontarekisterien keskeytysliput. CA1 asettaa valvontarekisterin CRA bitin 7 loogiseksi "1":ksi ja CB1 vastaavasti CRB:n bitin 7.

3.3.2. Oheislaitteen valvontalinja (CA2)

Oheislaittevalvontalinja CA2 voidaan ohjelmoida toimimaan keskeytystulona tai oheislaitteen valvontalähtönä. Toimiessaan lähtönä tämä linja on TTL-standarditasoa ja sisäisen ylösvetovastuksen vuoksi tulona toimiessaan edustaa 1,5 kertaista TTL-kuormaa. Linjan toiminta ohjelmoidaan valvontarekisterillä A (CRA).

3.3.3. Oheislaitteen valvontalinja (CB2)

Oheislaittevalvontalinja CB2 voidaan myös ohjelmoida toimimaan joko keskeytystulona tai oheislaittevalvontalähtönä. Lähtönä linja edustaa standardia TTL-tasoa ja sitä voidaan käyttää myös transistorikytkimien ohjauksena. Linja ohjelmoidaan valvontarekisterillä B (CRB).

4. TOIMINTA

Järjestelmän alustusrutiinin aikana, joka suoritetaan prosessorin RESET-signaalin toimesta, mikroprosessori kirjoittaa joukon ykkösiä ja nollia datan suunnan rekistereihin. Nämä ykköset ja nollat määrittävät linjat, jotka toimivat tuloina ja linjat, jotka toimivat lähtöinä. Tämä ykkösten ja nollien joukko on yleensä kiinteästi sidottu järjestelmän toimintaan. Sen vuoksi seuraavan vaiheen on asetettava eri toimintatilat, datasiirrot jne, jotka suoritetaan valvontarekisterin kontrolloimana. Samaan aikaan datan suunnan rekisterin valvontatulobitti voidaan asettaa ykköseksi, jotta prosessori puolestaan voisi valvoa oheislaitteportteja järjestelmän toiminnan aikana.

Toiminnan aikana mikroprosessori tutkii oheislaitteen kytkimiä, tunnistimia jne luke-malla dataa oheislaitteen tulolinjoilta. Binäärinen tai desimaalinen data voidaan siirtää mikroprosessorille myös edellä esitetyllä tavalla. Samaan aikaan valvotaan esim. valoja, moottoreita, solenoideja yms. oheislaitteita kirjoittamalla dataa kyseisiin oheislaitteiden lähtörekisterien bitteihin.

4.1. Valvontarekisterit (CRA ja CRB)

Kahden valvontarekisterin CRA ja CRB avulla mikroprosessori voi kontrolloida neljää oheislaittevalvontalinjaa CA1, CA2, CB1 ja CB2. Lisäksi ne antavat mikroprosessorille mahdollisuuden tarkistaa keskeytyslinjat sekä monitoroida keskeytyslippujen tilat. Mikroprosessori voi kirjoittaa tai lukea näiden kahden rekisterin bitit 0 ... 5 kun piirivalinta- ja rekisterinvalintasisignaalit ovat kunnossa. Rekisterien bitit 6 ja 7 voidaan ainoastaan lukea ja niiden tila määräytyy ulkoisten keskeytysten mukaan, jotka esiintyvät linjoilla CA1, CA2, CB1 tai CB2. Valvontasanan muoto on esitetty kohdassa 4.3.

4.2. Datan suunnan valvontatulobitti (CRA-2 ja CRB-2)

Kummankin valvontarekisterin CRA ja CRB bitti 2 määrittelee joko oheislaitteen lähtörekisterin valinnan tai vastaavan datan suunnan rekisterin valinnan kun rekisterin valintasisignaalit saadaan nastoihin RS0 ja RS1.

Bitin 2 ollessa "1" valitaan oheislaitteen lähtörekisteri ja kun se on "0", osoitetaan datan suunnan rekisteri.

4.2.1. Keskeytysliput (CRA-6, CRA-7, CRB-6 ja CRB-7)

Nämä neljä keskeytyslippua asetetaan keskeytys- ja oheislaitteen valvontalinjojen aktiivimuutoksilla linjojen ollessa ohjelmoidut toimimaan tuloina. Näitä bittejä ei voi asettaa suoraan mikroprosessorin dataväylältä ja niiden resetointi tapahtuu epäsuorasti oheislaitteen datan lukutoiminnon toimesta.

4.2.2. Oheislaitteen valvontalinjojen CA2 ja CB2 kontrolli (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4 ja CRB5)

Valvontarekisterien bittejä 3, 4 ja 5 käytetään oheislaitteen valvontalinjojen CA2 ja CB2 kontrollointiin. Nämä bitit määräävät toimiiko valvontalinjat keskeytystuloina vai valvontasignaaliähtöinä. Jos bitti CRA-5 (CRB-5) on alhaalla, CA2 (CB2) on samanlainen keskeytystulo kuin CA1 (CB1). Kun CRA-5 (CRB-5) on ylhäällä, CA2 (CB2) antaa lähtösignaalin jota voidaan käyttää oheislaitteen datasiirron valvontaan. Toimiessaan ähtöinä CA2:lla ja CB2:lla on jonkin verran erilaiset kuormitusominaisuudet.

4.2.3. Keskeytyslinjojen CA1 ja CB1 valvonta (CRA-0, CRB-0, CRA-1 ja CRB-1)

Valvontarekisterien kahta alinta bittiä käytetään keskeytyslinjojen CA1 ja CB1 kontrollointiin. Bitit CRA-0 ja CRB-0 mahdollistavat vastaavat mikroprosessorin keskeytysignaalit IRQA ja IRQB. Bitit CRA-1 ja CRB-1 määrittelevät keskeytystulosignaalien CA1 ja CB1 aktiivisen muutoksen.

4.3. Valvontasanan muoto

b7	b6	b5	b4	b3	b2	b1	b0
IRQA1 IRQB1 -lippu	IRQA2 IRQB2 -lippu	CA2 (CB2) valvonta			Datan suunta	CA1 (CB1) valvonta	

Bitti 7 - Keskeytyslippu

Asetus CA2:n (CB2:n) aktiivisella muutoksella, resetointi automaattisesti mikroprosessorin lukiessa lähtörekisterin A (B). Voidaan tyhjentää myös laitteistosta tapahtuvalla resetoinnilla.

Bitti 6 - Keskeytyslippu

Kun CA2 (CB2) on tulo, IRQA(B) nousee ylös CA2:n (CB2:n) aktiivisella muutoksella. Resetointi automaattisesti lähtörekisterin A (B) lukemisen tapahtuessa.

Bitti 5 - CA2 (CB2) toimintamuodon valinta

Katso b3.

Bitti 4 - Aktiivisen CA2 (CB2) muutoksen määrittäminen keskeytyslipun IRQA(B) asettamista varten

b4=0 IRQA(B)2 asetettu CA2:n (CB2:n) laskevalle muutokselle.
b4=1 IRQA(B)2 asetettu CA2:n (CB2:n) nousevalle muutokselle.

Bitti 3 - CA2 (CB2) valvonta

CA2 (CB2). Toimivat lähtöinä, kun b5=1

Huom! CA2:n ja CB2:n toiminnat lähtöinä eivät ole samanlaiset.

CA2 b3=0

CA2 menee alas ensimmäisellä E-pulssin laskevalla reunalla mikroprosessorin suoritettua lähtörekisterin A lukemisen. Nousee takaisin ylös seuraavalla aktiivilla CA1-pulssilla b1:n määrittelemällä tavalla.

b3=1

CA2 menee alas ensimmäisellä E-pulssin laskevalla reunalla mikroprosessorin suoritettua lähtörekisterin A lukemisen. Nousee takaisin ylös seuraavalla E-pulssin nousevalla reunalla.

CB2 b3=0

CB2 menee alas ensimmäisellä E-pulssin nousevalla reunalla mikroprosessorin luettua lähtörekisterin B. Paluu ylös seuraavalla aktiivisella CB1-muutoksella bitin 1 määrittelemällä tavalla. CRB-b7:n täytyy ensin tulla nollatuksi datan luvun yhteydessä.

b3=1

CB2 menee alas ensimmäisellä E-pulssin nousevalla reunalla mikroprosessorin luettua lähtörekisterin B. Paluu ylös seuraavalla E-pulssin nousevalla reunalla.

CA2:n (CB2:n) asettaminen

CA2 (CB2) menee alas, kun mikroprosessori kirjoittaa valvontarekisteriin b3=0.
CA2 (CB2) nousee ylös, kun mikroprosessori kirjoittaa valv. rekisteriin b3=1.

CA2 (CB2). Toimivat tuloina, kun b5=0

CA2 (CB2)

b3=0 Estää IRQA(B) keskeytyksen CA2:n (CB2:n) aktiivisella muutoksella.

b3=1 Mahdollistaa IRQA(B) keskeytyksen CA2:n (CB2:n) aktiivimuutoksella.

Bitti 2 - Datan suunnan rekisterin tai lähtörekisterin osoittamisen määrittely

b2=0 Valitaan datan suunnan rekisteri.

b2=1 Valitaan lähtörekisteri.

Bitti 1 - Aktiivisen muutoksen määrittäminen keskeytyslipun IRQA(B)1 asettamiseksi

b1=0 IRQA(B)1 on ylhäällä CA1:n (CB1:n) mennessä alas.

b1=1 IRQA(B)1 on alhaalla CA1:n (CB1:n) noustessa ylös.

Bitti 0 - CA1 (CB1) keskeytyksen mahdollisuus/esto

b0=0 Estää IRQA(B) keskeytyksen CA1:n (CB1:n) aktiivisella muutoksella.

b0=1 Mahdollistaa IRQA(B) keskeytyksen CA1:n (CB1:n) aktiivimuutoksella.

LUKU 5. VIDEOPROSESSORI TMS9929A

1. Johdanto

2. Liitännät, rekisterit ja näyttötilat

- 2.1. Videoprosessorin liitäntä mikroprosessoriin
- 2.2. Videoprosessori/VRAM-liitäntä
- 2.3. Videoprosessori/TV-liitäntä
- 2.4. Videoprosessorin rekisterit
- 2.5. Tilarekisterit
- 2.6. Näyttötilat

3. VRAM-muistiosoitteet

4. TMS9928A/9929A nastat

5. Demo-ohjelma

VIDEOPROSESSORI TMS9929A

1. JOHDANTO

1.1. Yleistä

TMS9929A-videoprosessori (VDP, Video Display Processor) on N-kanava MOS LSI-piiri, jota käytetään yleisesti eri videolaitteissa joissa halutaan datanäyttö joko tavallisen väri-TV-vastaanottimen tai värimonitorin kuvaruudulle. TMS9929A-piiri muodostaa kaikki tarpeelliset video-, kontrolli- ja synkronointisignaalit ja valvoo myös näyttödatan talletusta, lukua ja virkistämistä dynaamisessa kuvaruutumuistissa, video-RAM:issa. Piiri tarvitsee minimaalisen vähän ulkoista elektroniikkaa liitääntään mikroprosessorin, video-RAM:in tai TV-vastaanottimen kanssa.

Videoprosessorilla on neljä videonäyttötilaa; Grafiikka I-, Grafiikka II-, Moniväri- tai Tekstitila. Tekstitilassa kuvaruudulle saadaan 24 riviä, joilla on 40 merkkiä rivillä, ja käytettävissä on kaksi väriä. Näin pystytään hyödyntämään tavallisen TV-vastaanottimen maksimaalinen erottelukyky, kun kuvaruudulla halutaan näyttää alfanumeerisia merkkejä. Moniväritilassa voidaan muodostaa rajoittamaton määrä 64 x 48 värialkion näyttöjä käyttäen 15 väriä sekä läpinäkyvää.

Grafiikkatilassa I saadaan 256 x 192 pisteen näyttö hahmografiikkaa varten ja käytettäviä värejä on niinkään 15 sekä läpinäkyvä. Grafiikkatila II on I-tilan laajennus, jossa voidaan muodostaa monimutkaisempia väri- ja hahmonäyttöjä.

Videonäyttö muodostuu 35 tasosta, ulkoisesta videosta (ei käytössä Managerissa), taustaverhosta, hahmotasosta ja 32 sprite*-tasosta. Tasot on pinottu pystysuuntaisesti yhdessä ulkoisen videon kanssa, joka on sisimmällä tason pohjalla. Taustaverho on seuraavana ja sen jälkeen hahmotaso joka sisältää Grafiikka I- ja Grafiikka II-hahmot sekä 32 sprite*-tasoa päällimmäisenä.

* Sprite tai suomenkielessä myös esiintyvä "spraitti" on animaatiohahmo tai grafiikka-merkki/-kuvio jota voidaan liikuttaa nopeasti ja pehmeästi pitkin kuvaruutua.

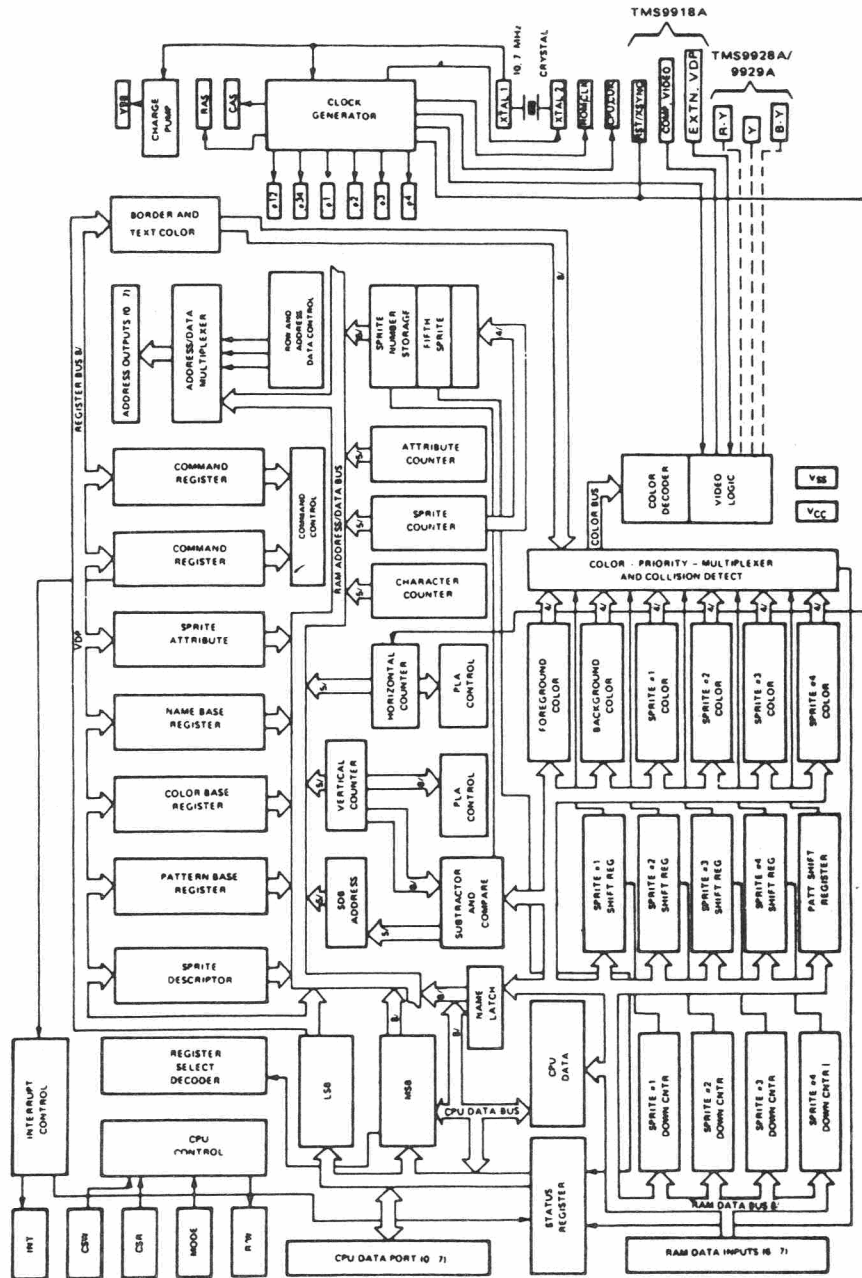
TMS9929A-videoprosessori käyttää näyttöparametrien talletukseen joko 4 K, 8 K tai 16 K dynaamista muistia. Salora Managerissa dynaamisen video-RAM:in koko on 16 K.

1.2. Piirin ominaisuudet

- Yhden piirin liitäntä väri-TV:hen (ei sisällä RAM:ia eikä RF-modulaattoria)
- 256 x 192 pisteen erottelukyky TV-kuvaruudulla
- 15 erilaista väriä sekä läpinäkyvä
- 8-bittinen kaksisuuntainen liitäntä mikroprosessoriin
- Dynaamisen RAM-piirin automaattinen ja havaitsematon virkistys
- Ulkoisen videon syöttömahdollisuus
- Erilainen tasokuvaus 30 simulaatiolle
- Standardi 40-nastainen kotelo

Kuvassa 1 on TMS9929A:n lohkokaavio.

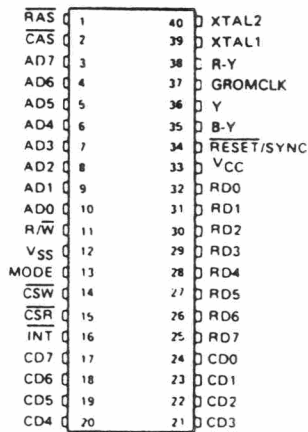
Lohkojen toiminnat esitetään myöhemmin tässä luvussa.



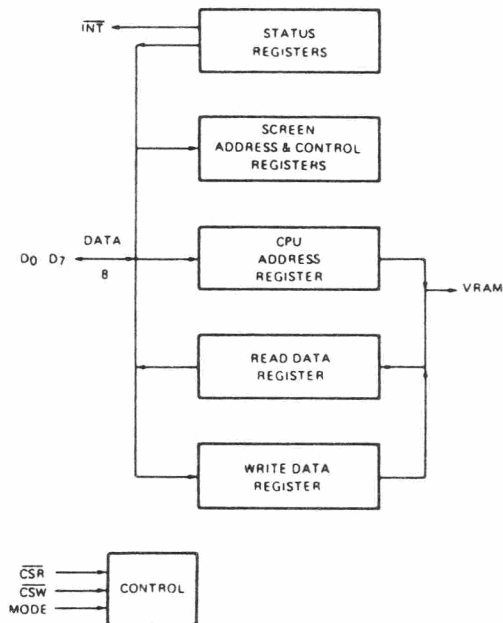
lukutoiminta videoprosessorilta CPU:lle. Kun se on aktiivitulassa (myös alhaalla), videoprosessorilta saadaan 8 bittiä D0 - D7 CPU:lle.

$\overline{\text{CSW}}$:n ja $\overline{\text{CSR}}$:n ei tulisi koskaan olla samanaikaisesti alhaalla. Jos näin tapahtuu, videoprosessori lähettää dataa D0 - D7 -linjalle ja lukee virheellistä tietoa sisään.

MODE määrittelee luku- tai kirjoitustapahtuman lähteen tai määräpaikan.



Kuva 2-a. Videoprosessorin nastajärjestys.



Kuva 2-b. Videoprosessorin liitäntä mikroprosessoriin.

2.1.2. CPU:n kirjoitus videoprosessorin rekisteriin

Videoprosessorissa on kahdeksan kirjoitettavaa rekisteriä (write-only) sekä yksi luettava rekisteri (read-only). Kirjoitusrekisterit valvovat videoprosessorin toimintaa ja määräävät väylän josta video-RAM syöttää dataa. Tilarekisteri käsittää keskeytys-, sprite-kuvien samanaikaisuus- ja viidennen spriten tilaliput.

Kukin videoprosessorin kahdeksan kirjoitusrekisteriä voidaan ladata CPU:lta käyttäen kahta 8-bittistä datasiirtoa. Taulukossa 1 on esitetty tarvittavan kahden tavun muoto. Ensimmäinen siirrettävä tavu on datatavu ja toinen tavu valvoo määräpaikkaa. Toisen tavun eniten merkitsevän bitin täytyy olla "1". Seuraavat neljä bittiä ovat "0" ja kolme alinta bittiä muodostavat määräpaikan rekisterin numeron.

MODE-tulo on ylhäällä molempien tavujen siirron ajan.

Jotta data voitaisiin uudelleenkirjoittaa sisäiseen rekisteriin sen jälkeen kun datatavu on jo ladattu, tilarekisteri täytyy lukea, jonka jälkeen piirin sisäinen logiikka on valmis hyväksymään tavun datana eikä rekisterin määräpaikkana. Tämä tilanne voi esiintyä keskeytysohjatuissa ohjelmanosissa. Aina kun videoprosessorin kirjoitusparametrien tila on kyseessä, tätä toimintoa tulisi käyttää. On myös huomioitava, että CPU:n osoite tuhotaan, kun videoprosessorin rekisteriin kirjoitetaan.

Toiminta	Bitti								CSW	CSR	MODE
	0	1	2	3	4	5	6	7			
Kirj. videoprosessorin rekisteriin											
Tavu 1 datan kirjoitus	D0	D1	D2	D3	D4	D5	D6	D7	0	1	1
Tavu 2 rekisterin valinta	1	0	0	0	0	RS0	RS1	RS2	0	1	1
Kirj. video-RAM:iin											
Tavu 1 osoite	A6	A7	A8	A9	A10	A11	A12	A13	0	1	1
Tavu 2 osoite	0	1	A0	A1	A2	A3	A4	A5	0	1	1
Tavu 3 datan kirjoitus	D0	D1	D2	D3	D4	D5	D6	D7	0	1	0
Luku videoprosessorin rekisteriltä											
Tavu 1 datan luku	D0	D1	D2	D3	D4	D5	D6	D7	1	0	1
Luku video-RAM:ilta											
Tavu 1 osoite	A6	A7	A8	A9	A10	A11	A12	A13	0	1	1
Tavu 2 osoite	0	0	A0	A1	A2	A3	A4	A5	0	1	1
Tavu 3 datan luku	D0	D1	D2	D3	D4	D5	D6	D7	1	0	0

Taulukko 1. CPU/videoprosessorin datan siirrot

2.1.3. CPU:n kirjoitus video-RAM:iin

CPU siirtää dataa video-RAM:iin videoprosessorin välityksellä käyttäen 14-bittistä autoinkrementoitavaa osoiterekisteriä. Osoiterekisterin asettamiseen tarvitaan kahden tavun siirto. Sen jälkeen vaaditaan yhden tavun siirto datan kirjoittamiseksi osoitettuun video-RAM-paikkaan. Tämän jälkeen osoiterekisteriä inkrementoidaan (lisätään yhdellä) automaattisesti. Perättäiset video-RAM:in lukemiset vaativat ainoastaan yhden tavun siirron, koska osoiterekisteri on aina asetettu. Osoiterekisterin asettamisen aikana toisen osoitetavun kahden eniten merkitsevän bitin ensimmäisen täytyy olla "0" ja toisen "1". MODE on ylhäällä molempien osoitesiiirtojen aikana ja alhaalla datasiirron tapahtuessa. \overline{CSW} :tä käytetään kaikissa siirroissa syötettäessä 8 bittiä videoprosessorille (ks. Taulukko 1).

2.1.4. CPU:n lukeminen videoprosessorin tilarekisteriltä (VDP Status Register)

CPU voi lukea tilarekisterin yhden tavun siirrolla. MODE on ylhäällä siirron ajan. Videoprosessorille ilmoitetaan lukutoiminnosta $\overline{\text{CSR}}$ -signaalilla.

2.1.5. CPU:n lukeminen video-RAM:ilta (VRAM)

CPU lukee dataa VRAM:ilta videoprosessorin välityksellä käyttäen automaattisesti inkrementoitavaa (lisäävää) osoiterekisteriä. Osoitteellisen VRAM-tavun datan lukuun tarvitaan yhden tavun siirto, jonka jälkeen osoiterekisteriä inkrementoidaan automaattisesti. Peräkkäisen VRAM-datan lukeminen vaatii ainoastaan yhden tavun siirron, koska osoiterekisteri on jo asetettu. Osoiterekisterin asettamisen ajan toisen osoitetavun kahden eniten merkitsevän bitin tulee olla nollia. Asettamalla osoiterekisteri tällä tavoin aloitetaan VRAM:in lukujakso ja ensimmäisen siirron luettava data saadaan CPU:lta (vrt. Taulukko 1). MODE on ylhäällä osoitetavun siirron ajan ja alhaalla datan siirron ajan. Videoprosessori tarvitsee noin 8 us datan siirron jälkeisen VRAM-tavun hakemiseen ja 3 ms seuraavan osoitteen asettamiseen.

2.1.6. Videoprosessorin keskeytys

Videoprosessorin $\overline{\text{INT}}$ -nastaa käytetään keskeytyksen generointiin jokaisen aktiivinäytön pyyhkäisyn lopussa, mikä tapahtuu noin joka 1/60 sekunti. $\overline{\text{INT}}$ -lähtö on aktiivinen, kun keskeytyksen Enable-bitti (IE, Interrupt Enable) videoprosessorin rekisterissä 1 on "1"-tilassa ja F-bitti tilarekisterissä on "1"-tilassa. Keskeytys peruutetaan, kun tilarekisteri on luettu.

2.1.7. Videoprosessorin alustus

Videoprosessori alustetaan ulkoisesti aina kun $\overline{\text{RESET}}$ -tulo on aktiivitasossa (alhaalla) ja pidetään alhaalla vähintään 3 ms ajan. Ulkoinen nollaus synkronoi kaikki kellot pulssin laskevalla reunalla, asettaa horisontaali- ja vertikaalilaskimet tiettyihin tiloihin ja tyhjentää videoprosessorin rekisterit 0 ja 1. Kuvaruutunäyttöön tulee automaattisesti blanco-kuva (tyhjä ruutu), koska videoprosessorin rekisterissä 1 oleva BLANK-bitti menee "0"-ksi. Videoprosessori jatkaa kuitenkin VRAM:in virkistämistä, vaikka kuvaruudulle saadaankin blanco-kuva. Kun $\overline{\text{RESET}}$ -linja on aktiivitasossa, videoprosessori ei virkistä VRAM:ia.

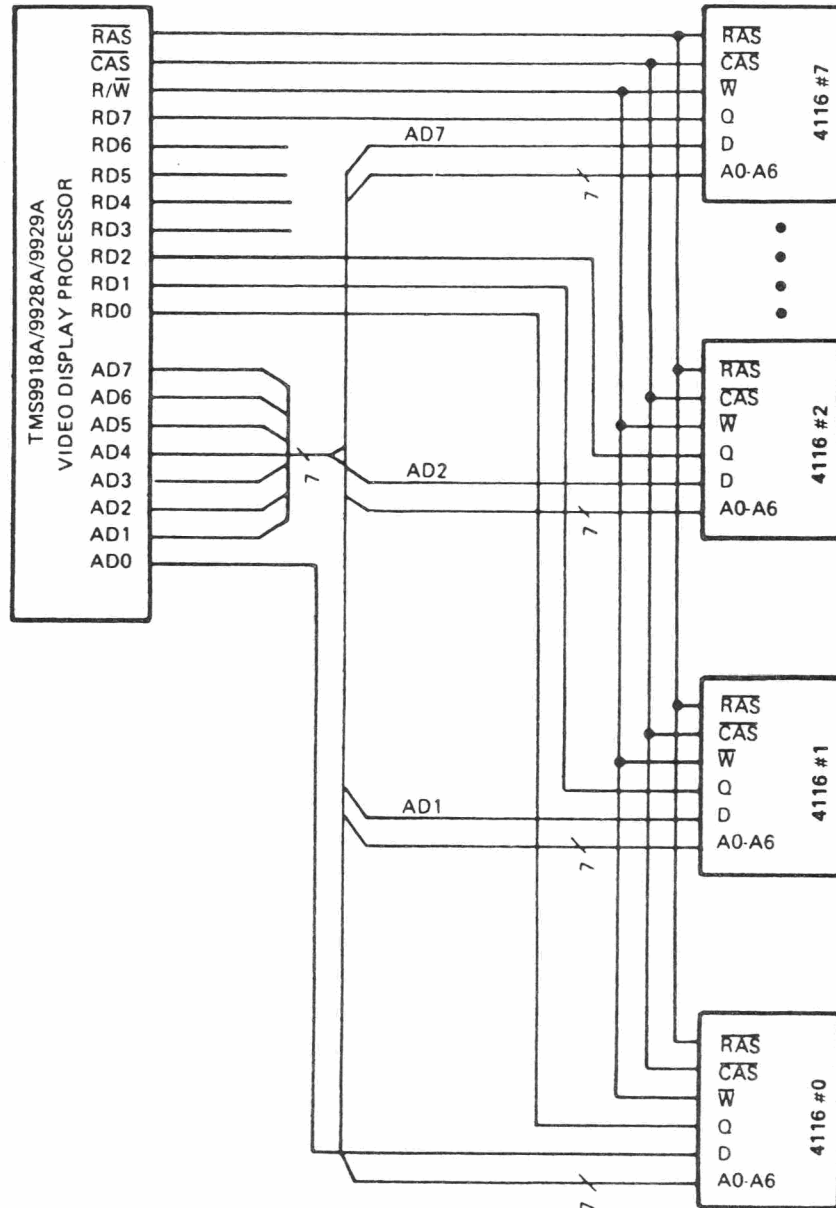
2.2. VIDEOPROSESSORI/VRAM-LIITÄNTÄ

Videoprosessori voi osoittaa VRAM:ista jopa 16 384 tavua käyttäen 14-bittistä VRAM-osoitetta. Videoprosessori hakee dataa VRAM:ista muodostaakseen kuvia kuten myöhemmin tässä luvussa esitetään. Se tallettaa myös dataa ja lukee sitä VRAM:ista CPU:n ja VRAM:in välisen datan siirron aikana sekä virkistää myös automaattisesti VRAM-muistia.

2.2.1. VRAM-liitännän valvontasignaalit

Videoprosessorin ja VRAM:in välinen liitäntä muodostuu kahdesta yksisuuntaisesta 8-bittisestä dataväylästä ja kolmesta valvontalinjasta kuvan 3 mukaisesti. VRAM syöttää

dataa videoprosessorille VRAM-lukuväylällä RD0 - RD7 (RD, Read Data; datan luku). Videoprosessori puolestaan syöttää sekä osoitteen että dataa VRAM:ille tämän osoite/-dataväylällä AD0 - AD7 (AD, Address/Data; osoite/data). VRAM:in riviosoite on lähdössä $\overline{\text{RAS}}$:n ollessa aktiivitisassa (alhaalla). Sarakeosoite saadaan vuorostaan $\overline{\text{CAS}}$:n ollessa aktiivinen (myös alhaalla). Lähdössä on dataa VRAM:ille, kun $\overline{\text{RW}}$ on aktiivitisassa (niinikään alhaalla).

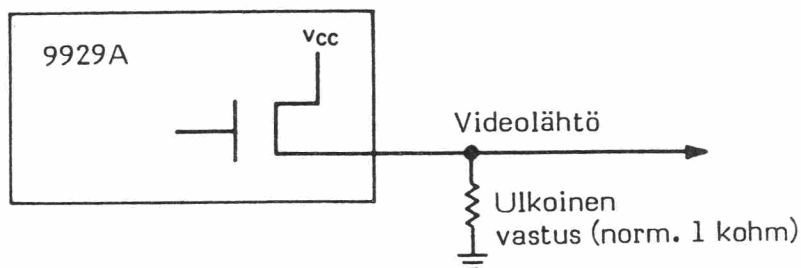


Kuva 3. VRAM-liitäntä

2.3. VIDEOPROSESSORI/TV-LIITÄNTÄ

Yhdistetty videosignaali, joka saadaan videoprosessorilta, voidaan kytkeä MONITOR-liittimen kautta monitoreihin/TV-vastaanottimiin joissa on videoliitäntä. Signaali sisältää kaikki tarvittavat horisontaali- ja vertikaalipulssit sekä luminanssi- ja krominanssiosat.

Tavallisia TV-vastaanottimia varten käytetään RF-modulaattoria videoprosessorin lähdössä, jolloin signaali voidaan syöttää antenniliittimen kautta TV:lle.



Kuva 4.

2.4. VIDEOPROSESSORIN REKISTERIT

Kuvassa 5 esitetään videoprosessorin kahdeksan kirjoitusrekisteriä (Write Only Registers). CPU lataa rekisterit kuten kohdassa 2.1.2. selostettiin. Rekisterit 0 ja 1 sisältävät liput, joilla mahdollistetaan tai peruutetaan videoprosessorin lukuisat eri ominaisuudet ja tilat. Rekisterit 2 ... 6 sisältävät arvot, joilla määritetään video-RAM:in eri alalohkojen aloituspaikat. Nämä selostetaan tarkemmin luvussa 2.5. Rekisteriä 7 käytetään taustaverhon ja tekstivärin määrittelyyn. Seuraavassa esitetään eri rekisterien lyhyt kuvaus.

2.4.1. Rekisteri 0

Rekisteri 0 sisältää kaksi videoprosessorin valvontabittiä. Muiden bittien sisältö on 0.

Bitti 6 M3 (Mode Bit 3; tilabitti 3). Katso kohta 2.4.2.

Bitti 7 Ulkoisen videon käyttö
"0" ulkoisen videon käyttö
"1" ei ulkoista videota (käytössä Managerissa)

2.4.2. Rekisteri 1

Rekisteri 1 sisältää 8 videoprosessorin valvontabittiä.

Bitti 0 4K/16K-valinta
"0" valitsee 4027-RAM-toiminnan
"1" valitsee 4108, 4116-RAM-toiminnan

Managerissa on grafiikkatilassa seuraavat rekisterien arvot:

```

Rekisteri 0 0000 0010
1 1110 0000
2 0000 1111
3 1111 1111
4 0000 0011
5 0111 1110
6 0000 0011
7 0000 0000

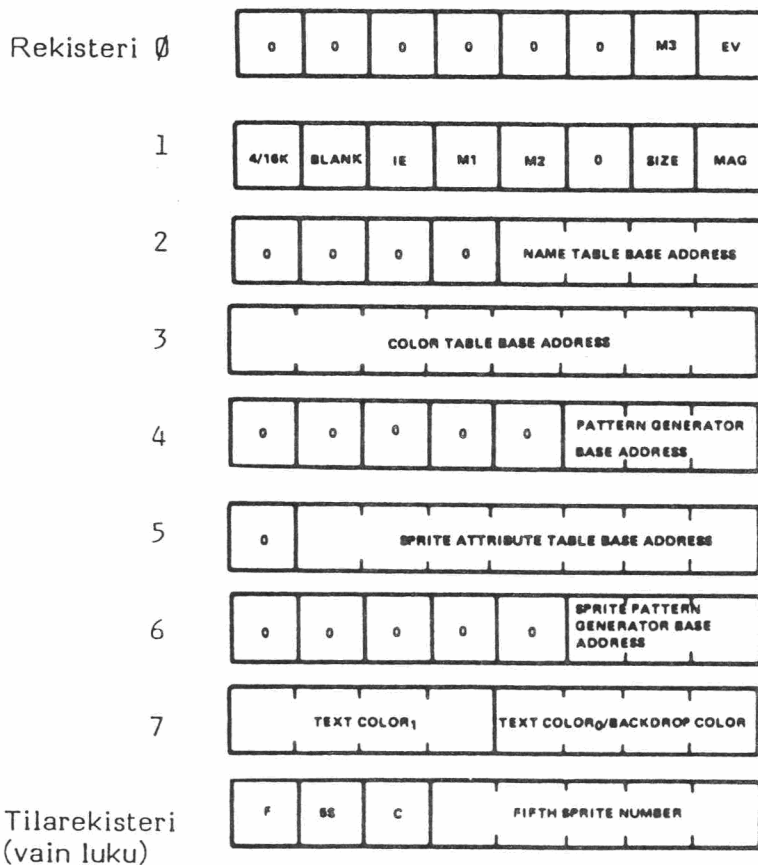
```

Tekstitilassa rekisterien arvot ovat:

```

Rekisteri 0 0000 0000
1 1111 0000
2 0000 1110
3 1111 1111
4 0000 0011
5 0111 1110
6 0000 0011
7 0010 0000

```



Kuva 5. Videoprosessorin rekisterit

- Bitti 1 BLANK-ohjaus
 "0" tyhjentää aktiivisen näyttöalueen (blanco-kuva)
 "1" mahdollistaa kuvaruutunäytön
- Bitti 2 IE (Interrupt Enable, mahdollisuus keskeytyksiin)
 "0" videoprosessorin keskeytykset ei mahdollisia
 "1" keskeytykset mahdollisia
- Bitit 3, 4 M1, M2 (Mode Bit 1, 2 ; tilabitit 1 ja 2)
 M1, M2 ja M3 määrittelevät videoprosessorin toimintatilan

M1	M2	M3	Toimintatila
0	0	0	Grafiikka I
0	0	1	Grafiikka II
0	1	0	Moniväritila
1	0	0	Tekstitila

- Bitti 5 Varattu
- Bitti 6 Koko (spritien koon valinta)
 "0" valitsee 0-koon spritet (8 x 8 bittiä)
 "1" valitsee 1-koon spritet (16 x 16 bittiä)
- Bitti 7 MAG (Magnification, spriten suurennus)
 "0" valitsee MAG0-sprite (1 x)
 "1" valitsee MAG1-sprite (2 x)

2.4.3. Rekisteri 2

Rekisteri 2 määrää nimitaulukko-alalohkon perusosoitteen. Sen sisältö on 0 ... 15. Rekisterin sisältö muodostaa 4 ylintä bittiä 14-bittisistä nimitaulukon osoitteista. Nimitaulukon perusosoite on kuitenkin (rekisteri 2) • 400 (heksadesimaalisena).

2.4.4. Rekisteri 3

Rekisteri 3 määrittelee väritaulukko-alalohkon perusosoitteen. Sen sisältö on alueella 0... 255. Rekisterin sisältö muodostaa 8 ylintä bittiä 14-bittisistä väritaulukon osoitteista. Väritaulukon perusosoite on kuitenkin (rekisteri 3) • 40 (heksadesimaalisena).

2.4.5. Rekisteri 4

Rekisteri 4 määrittelee Hahmo-, Teksti- tai Moniväri-alalohkon perusosoitteen. Sen sisältö on välillä 0 ... 7. Sisältö muodostaa 3 ylintä bittiä 14-bittisestä osoitteesta. Perusosoite on kuitenkin (rekisteri 4) • 800 (heksadesimaalisena).

2.4.6. Rekisteri 5

Rekisteri 5 määrittelee sprite-attribuuttitaulukon perusosoitteen. Sen sisältö on 0 ... 127 ja se muodostaa 7 ylintä bittiä 14-bittisistä sprite-attribuuttitaulukon osoitteista. Perusosoite on (rekisteri 5) • 80 (heksadesimaalisena).

2.4.7. Rekisteri 6

Rekisteri 6 määrittelee sprite-hahmotaulukon perusosoitteen ja sen sisältö voi olla 0 ... 7. Tämä muodostaa 3 ylintä bittiä 14-bittisistä sprite-hahmotaulukko-osoitteista; perusosoite on kuitenkin (rekisteri 6) • 800 (heksadesimaalisena).

2.4.8. Rekisteri 7

Rekisterin 7 neljä ylintä bittiä käsittävät värin 1 värikoodin Tekstitilassa. Neljä alinta

bittiiä sisältävät värin Ø värikoodin Tekstitilassa ja taustavärin kaikissa tiloissa. Värikoodit on esitelty taulukossa 3.

Videoprosessorin rekisteri 3 sisältää väritaulukon alkuosoitteen.

$$(R3) \cdot 4\emptyset_{(16)} = \text{ALKUOSOITE}$$

R3	ALKU-OSOITE	R3	ALKU-OSOITE	R3	ALKU-OSOITE
00	0000	28	0A00	50	1400
01	0040	29	0A40	51	1440
02	0080	2A	0AB0	52	1480
03	00C0	2B	0AC0	53	14C0
04	0100	2C	0B00	54	1500
05	0140	2D	0B40	55	1540
06	0180	2E	0BB0	56	1580
07	01C0	2F	0BC0	57	15C0
08	0200	30	0C00	58	1600
09	0240	31	0C40	59	1640
0A	0280	32	0CB0	5A	1680
0B	02C0	33	0CC0	5B	16C0
0C	0300	34	0D00	5C	1700
0D	0340	35	0D40	5D	1740
0E	0380	36	0D80	5E	1780
0F	03C0	37	0DC0	5F	17C0
10	0400	38	0E00	60	1800
11	0440	39	0E40	61	1840
12	0480	3A	0EB0	62	1880
13	04C0	3B	0EC0	63	18C0
14	0500	3C	0F00	64	1900
15	0540	3D	0F40	65	1940
16	0580	3E	0F80	66	1980
17	05C0	3F	0FC0	67	19C0
18	0600	40	1000	68	1A00
19	0640	41	1040	69	1A40
1A	0680	42	1080	6A	1A80
1B	06C0	43	10C0	6B	1AC0
1C	0700	44	1100	6C	1B00
1D	0740	45	1140	6D	1B40
1E	0780	46	1180	6E	1B80
1F	07C0	47	11C0	6F	1BC0
20	0800	48	1200	70	1C00
21	0840	49	1240	71	1C40
22	0880	4A	1280	72	1C80
23	08C0	4B	12C0	73	1CC0
24	0900	4C	1300	74	1D00
25	0940	4D	1340	75	1D40
26	0980	4E	1380	76	1D80
27	09C0	4F	13C0	77	1DC0

Taulukko jatkuu seuraavalla sivulla.

R3	ALKU-OSOITE	R3	ALKU-OSOITE	R3	ALKU-OSOITE
78	1E00	A6	2980	D3	34C0
79	1E40	A7	29C0	D4	3500
7A	1E80	A8	2A00	D5	3540
7B	1EC0	A9	2A40	D6	3580
7C	1F00	AA	2A80	D7	36C0
7D	1F40	AB	2AC0	D8	3600
7E	1F80	AC	2B00	D9	3640
7F	1FC0	AD	2B40	DA	3680
80	2000	AE	2880	DB	36C0
81	2040	AF	2BC0	DC	3700
82	2080	B0	2C00	DD	3740
83	20C0	B1	2C40	DE	3780
84	2100	B2	2C80	DF	37C0
85	2140	B3	2CC0	E0	2800
86	2180	B4	2D00	E1	3840
87	21C0	B5	2D40	E2	3880
88	2200	B6	2D80	E3	38C0
89	2240	B7	2DC0	E4	3900
8A	2280	B8	2E00	E5	3940
8B	22C0	B9	2E40	E6	3980
8C	2300	BA	2E80	E7	39C0
8D	2340	BB	2EC0	E8	3A00
8E	2380	BC	2F00	E9	3A40
8F	23C0	BD	2F40	EA	3A80
90	2400	BE	2FB0	EB	3AC0
91	2440	BF	2FC0	EC	3B00
92	2480	C0	3000	ED	3B40
93	24C0	C1	3040	EE	3B80
94	2500	C2	3080	EF	3BC0
95	2540	C3	30C0	F0	3C00
96	2580	C4	3100	F1	3C40
97	25C0	C5	3140	F2	3C80
98	2600	C6	3180	F3	3CC0
99	2640	C7	31C0	F4	2D00
9A	2680	C8	3200	F5	3D40
9B	26C0	C9	3240	F6	3D80
9C	2700	CA	3280	F7	3DC0
9D	2740	CB	32C0	F8	3E00
9E	2780	CC	3300	F9	3E40
9F	27C0	CD	3340	FA	3E80
A0	2800	CE	3380	FB	3EC0
A1	2840	CF	33C0	FC	3F00
A2	2880	D0	3400	FD	3F40
A3	28C0	D1	3440	FE	3F80
A4	2900	D2	3480	FF	3FC0
A5	2940				

Videoprosessorin rekisteri 4 käsittää hahmotaulukon alkuosoitteen.

$$(R4) \cdot 800_{(16)} = \text{ALKUOSOITE}$$

R4	ALKU-OSOITE	R4	ALKU-OSOITE
00	0000	04	2000
01	0800	05	2800
02	1000	06	3000
03	1800	07	3800

Videoprosessorin rekisteri 5 sisältää sprite-attribuuttitaulukon alkuosoitteen.

$$R5 \cdot 80_{(16)} = \text{ALKUOSOITE}$$

R5	ALKU-OSOITE	R5	ALKU-OSOITE	R5	ALKU-OSOITE	R5	ALKU-OSOITE
00	0000	21	1080	40	2000	60	3000
01	0080	22	1100	41	2080	61	3080
02	0400	23	1180	42	2100	62	3100
03	0180	24	1200	43	2180	63	3180
04	0200	25	1280	44	2200	64	3200
05	0280	26	1300	45	2280	65	3280
06	0300	27	1380	46	2300	66	3300
07	0380	28	1400	47	2380	67	3380
08	0400	29	1480	48	2400	68	3400
09	0480	2A	1500	49	2480	69	3480
0A	0500	2B	1580	4A	2500	6A	3500
0B	0580	2C	1600	4B	2580	6B	3580
0C	0600	2D	1680	4C	2600	6C	3600
0D	0680	2E	1700	4D	2680	6D	3680
0F	0780	2F	1780	4E	2700	6E	3700
10	0800	30	1800	4F	2780	6F	3780
11	0880	31	1880	50	2800	70	3800
12	0900	32	1900	51	2880	71	3880
13	0980	33	1980	52	2900	72	3900
14	0A00	34	1A00	53	2980	73	3980
15	0AB0	35	1A80	54	2A00	74	3A00
16	0B00	36	1B00	55	2AB0	75	3AB0
17	0BB0	37	1BB0	56	2B00	76	3B00
18	0C00	38	1C00	57	2BB0	77	3BB0
19	0CB0	39	1CB0	58	2C00	78	3C00
1A	0D00	3A	1D00	59	2CB0	79	3CB0
1B	0DB0	3B	1DB0	5A	2D00	7A	3D00
1C	0E00	3C	1E00	5B	2DB0	7B	3DB0
1D	0EB0	3D	1EB0	5C	2ED0	7C	3E00
1E	0F00	3E	1F00	5D	2EB0	7D	3EB0
1F	0FB0	3F	1FB0	5E	2F00	7E	3F00
20	1000			5F	2FB0	7F	3FB0

Rekisteri 6 sisältää sprite-hahmotaulukon alkuosoitteen.

$$\text{ALKUOSOITE} = R6 \cdot <800$$

R4	ALKU-OSOITE	R4	ALKU-OSOITE
00	0000	04	2000
01	0800	05	2800
02	1000	06	3000
03	1800	07	3800

2.5. TILAREKISTERI

Videoprosessorissa on yksi 8-bittinen tilarekisteri jota CPU voi käyttää. Tilarekisteri sisältää keskeytyksen keston lipun, spriten esiintymisen lipun, viidennen spriten lipun ja viidennen spriten numeron, mikäli sellainen on olemassa. Tilarekisterin muoto on esitetty kuvassa 5.

Tilarekisteri voidaan lukea milloin tahansa ja F-, C- ja 5S-bitit testata. Tilarekisterin lukeminen tyhjentää keskeytyslipun F, koska asynkroniset lukemiset aikaansaavat F-lipun bittien nollautumiset. Tästä syystä tilarekisteri pitäisi lukea ainoastaan videoprosessorin keskeytyksen voimassaoloaikana.

2.5.1. Keskeytyslippu (F)

F-lippu asetetaan tilarekisterissä "1":ksi aktiivisen näytön viimeisen juovan pyyhkäisyn jälkeen. Se nollataan kun tilarekisteri on luettu tai kun videoprosessori nollataan ulkoisesti. Jos videoprosessorin rekisterin 1 Interrupt Enable on aktiivitilassa ("1"), keskeytyslähde \overline{INT} tulee aktiiviseksi (menee alas) aina kun tilalippu F on "1".

2.5.2. Pällekkäisyyslippu (C)

C-lippu asetetaan tilarekisterissä "1":ksi jos samanaikaisesti esiintyy päällekkäin kaksi tai useampia sprite-kuvia. Pällekkäisyys tarkoittaa sitä, että spriteilla on yksi tai useampia yhteisiä pisteitä. Läpinäkyvät sprite-kuviot, samoin kuin ne, jotka ovat osittain tai kokonaan kuvaruudun ulkopuolella, voivat olla myös päällekkäin. C-lippu asetetaan "0":ksi kun tilarekisteri on luettu tai kun videoprosessori nollataan ulkoisesti.

2.5.3. Viidennen spriten lippu (5S) ja numero

5S-lippu asetetaan tilarekisterissä "1":ksi jos vaakalinjoilla (riveillä 0 ... 192) on viisi tai useampi sprite-kuvioita ja kun Frame-lippu on "0". 5S-tilalippu asetetaan "0":ksi kun tilarekisteri on luettu tai kun videoprosessori nollataan ulkoisesti. Viidennen spriten numero asetetaan tilarekisterin 5:lle alimmalle bitille 5S-lipun asettamisen jälkeen ja on voimassa aina kun 5S-lippu on "1". Viidennen spriten lipun asettaminen ei aiheuta keskeytyksiä.

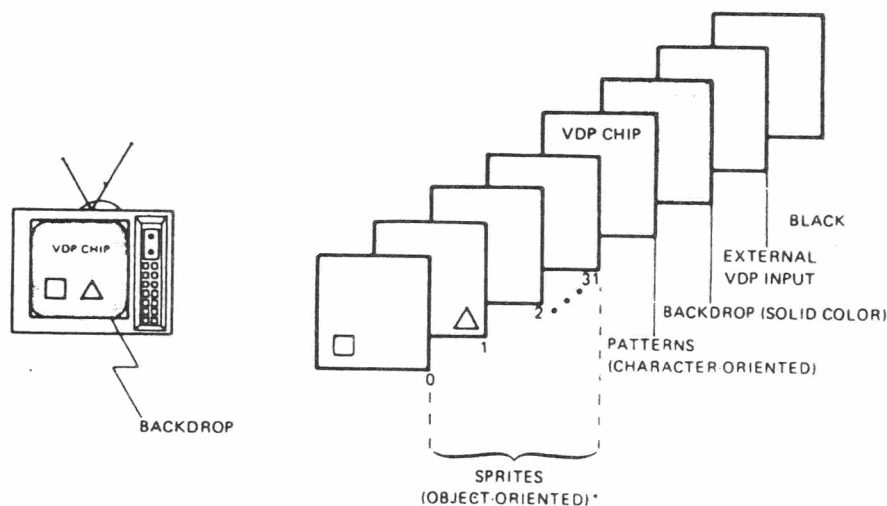
2.6. NÄYTTÖTILAT

Videoprosessori muodostaa kuvion kuvaruudulle siten, että eri näyttötasoja kerrostetaan päällekkäin. Kuvassa 6 a nähdään eri tasot. Lähinnä katsojaa olevilla tasoilla sijaitsevilla kohteilla on suurempi prioriteetti. Tilanteissa, joissa kahden eri tason kuviot esiintyvät kuvaruudun samassa pisteessä, suuremman prioriteetin omaavan tason kuvio esitetään kyseisessä pisteessä. Jotta tietyn tason kuvio näkyisi läpi, kaikkien tämän tason edessä olevien tasojen tulee olla läpinäkyviä tällä kohtaa. 32 ensimmäisellä tasolla (kuvan 6 b tasoilla 0 ... 31) voi olla yksittäinen sprite (spritet tai "spraitit" ovat hahmokuviota joiden sijainti kuvaruudulla määrätään horisontaali- ja vertikaalikoordinaatein, ts. vaaka- ja pystysuorassa olevien rivien ja sarakkeiden leikkauspisteillä) video-RAM:issa. Sprite-tasojen alueet, jotka ovat spriten itsensä ulkopuolella, ovat läpinäkyviä. Koska spriten koordinaatit ilmoitetaan pisteinä, sprite voidaan sijoittaa ja sitä voidaan liikutella hyvin tarkasti. Spritejä voidaan muodostaa kolmessa eri koossa; 8 x 8 pisteen, 16 x 16 pisteen ja 32 x 32 pisteen. Sprite-tason takana on hahmotaso. Hahmotasoa käytetään tekstin tai grafiikkakuvien yhteydessä, jotka generoidaan Teksti-, Grafiikka I-, Grafiikka II- tai

Moniväritilassa. Hahmotason takana on taustaverho, joka on laajempi alue kuin muut tasot ja joka siten muodostaa rajan muiden tasojen ympärille. Viimeinen ja alimman prioriteetin omaava taso on ulkoisen videon taso. Sen kuvio määritellään ulkoisen videon syöttönastalla (ei käytössä Managerissa). Taustaverho käsittää yksittäisen värin, jota käytetään rajojen näyttöön ja oletusvärinä aktiivisella näyttöalueella. Oletusvärin arvo on talletettu videoprosessorin rekisteriin 7. Kun taustaverhon värin koodi rekisterissä on läpinäkyvä, taustaverho oletetaan automaattisesti mustaksi jos ulkoista videota ei ole valittu.

32 sprite-tasoa käytetään 32 sprite-kuvion muodostukseen Moniväri- ja Grafiikkatiloissa. Niitä ei käytetä Tekstitilassa ja ne ovat tällöin automaattisesti läpinäkyviä. Kukin sprite-kuva voi käsittää 8 x 8, 16 x 16 tai 32 x 32 pistettä kyseisellä tasolla. Jokainen tasolla oleva alue, jota sprite ei peitä, on läpinäkyvä. Myös koko sprite-kuvio tai sen osa voi olla läpinäkyvä. Sprite 0 on uloimmalla tai korkeimmalla tasolla ja sprite 31 on rajakkain hahmotason kanssa. Aina kun sprite-tasolla oleva piste on läpinäkyvä, seuraavan tason väri voidaan nähdä läpi tämän pisteen kohdalta. Jos kuitenkin sprite-kuvan piste ei ole läpinäkyvä, alempien tasojen värit korvataan automaattisesti spriten värillä.

Myös linjalla olevien sprite-kuvien määrä on rajoitettu. Vain neljä sprite-kuvaa voi olla aktiivisena kullakin vaakasuoralla linjalla. Muut kyseisellä linjalla olevat spritet tehdään automaattisesti läpinäkyviksi. Vain nuo näytössä olevat aktiiviset sprite-kuvat aikaansaavat kyseisen lipun asettamisen. Videoprosessorin tilarekisteri muodostaa lippubitin ja viidennen spriten numeron aina kun sellainen esiintyy. Hahmotasoa käytetään Teksti-, Moniväri- ja Grafiikkatiloissa merkkien grafiikkahahmon näyttöön. Aina kun hahmotasolla oleva piste on läpinäkymätön, taustaverhon väri korvataan automaattisesti hahmotason värillä. Kun hahmotasolla oleva piste on läpinäkyvä, taustaverhon väri nähdään hahmotason läpi.

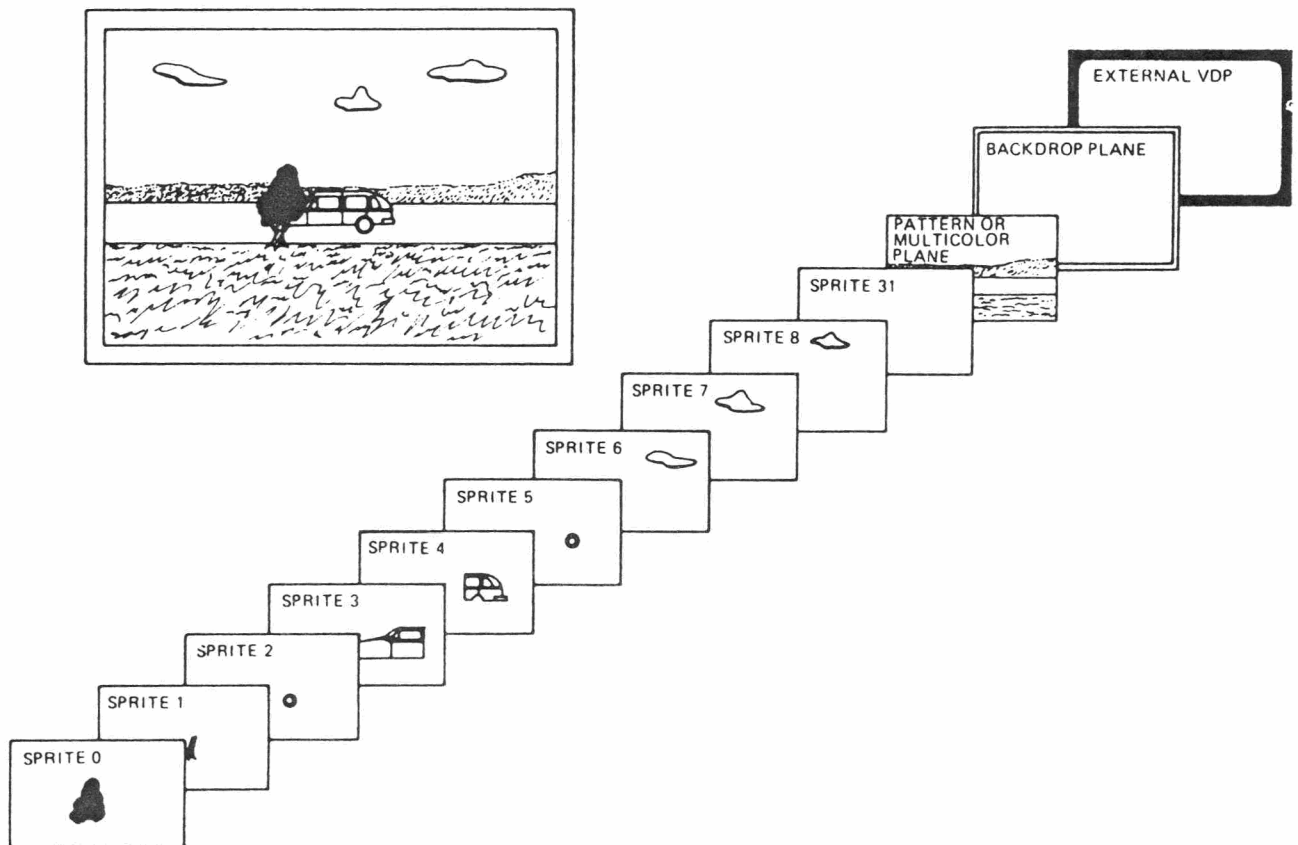


Kuva 6 a. Videoprosessorin näyttötasot

Videoprosessorilla on neljä värinäyttötilaa jotka esiintyvät hahmotasolla Grafiikka I-, Grafiikka II-, Teksti- ja Moniväritilassa. Grafiikka I- ja Grafiikka II -tilat jakavat hahmotason 8 x 8 pisteen ryhmiin, joita kutsutaan hahmopaikoiksi. Koska koko kuva on 256 x 192 pistettä, kuvaruudulla on grafiikkatiloissa 32 x 24 hahmopaikkaa.

Grafiikka I -tilassa 256 mahdollista hahmoa voidaan määrittellä 768 hahmopaikkaan ja kullekin hahmomääritelmälle voidaan antaa kaksi eri väriä. Grafiikka II -tilassa voidaan erityisen kartoituskaavion avulla määrittellä 768 hahmoa 768 hahmopaikkaan. Myös Grafiikka II -tilassa voidaan käyttää kahta eri väriä kunkin rivin hahmojen määrittelyssä. Siten voidaan käyttää kaikkia 15 väriä sekä läpinäkyvää yksittäisessä hahmopaikassa Tekstitilassa. Hahmotaso on jaettu 6 x 8 pisteen ryhmiin, joita kutsutaan tekstipaikoiksi. Tässä tilassa kuvaruudulla on 40 x 24 tekstipaikkaa. Tekstitilassa näyttöruudulle ei saada sprite-kuvioita ja koko kuvaruutu käsittää vain kaksi eri väriä. Moniväritilassa ruutu on jaettu 64 x 48 paikkaan ja kukin paikka muodostuu 4 x 4 pisteestä ja vain yhtä väriä voidaan käyttää kutakin paikkaa kohti.

Videoprosessorin rekisterit määrittelevät perusosoitteet video-RAM:issa oleville alalohkoille. Nämä alalohkot muodostavat taulukkoja, joita käytetään halutun kuvan tekemiseen kuvaruudulle. Sprite-kuvien muodostuksessa käytetään nimitaulukkoa, hahmotaulukkoa sekä sprite-taulukkoa ja näiden sisällön muodostaa mikroprosessori. Animaatio (kuvien liikkuvuus) saadaan aikaan vaihtamalla reaaliajassa video-RAM:in sisältöä.



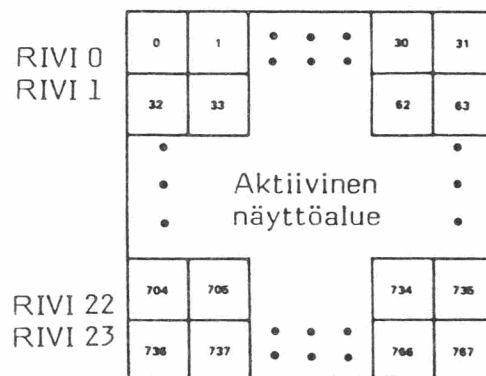
Kuva 6-b. VDP:n näyttötasot (32 ylintä tasoa).

Videoprosessorilla saadaan aikaan 15 väriä, kuten taulukossa 3 esitetään, ja se voi muodostaa myös kahdeksan eri harmaasävyä mustavalkoista TV-kuvaruutua varten. Taulukossa olevat luminanssiarvot (valoisuuden arvot) esittävät näitä tasoja; 0,00 edustaa mustaa ja 1,00 valkoista. Aina kun annetussa pisteessä kaikki tasot ovat läpinäkyviä, pisteessä näkyvä väri on musta.

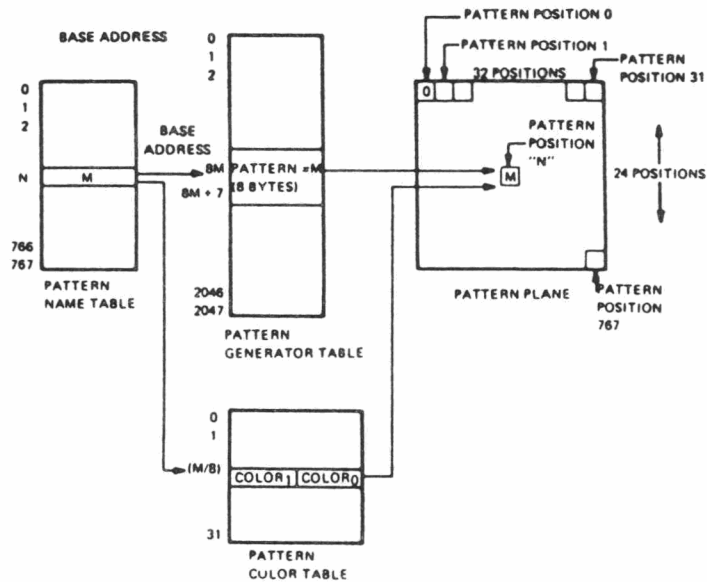
VÄRI (heksadesim.)	VÄRI	LUMINANSSI (DC-arvo)	KROMINANSSI (AC-arvo)
0	Läpinäkyvä	0,00	-
1	Musta	0,00	-
2	Keskivihreä	0,60	0,60
3	Vaal. vihreä	0,80	0,53
4	Tumman sininen	0,47	0,60
5	Vaal. sininen	0,67	0,60
6	Tumman punainen	0,53	0,53
7	Syaani	0,80	0,73
8	Keskipunainen	0,67	0,73
9	Vaal. punainen	0,80	0,73
A	Tumman keltainen	0,87	0,53
B	Vaal. keltainen	1,00	0,40
C	Tumman vihreä	0,47	0,60
D	Magenta	0,60	0,47
E	Harmaa	0,80	-
F	Valkoinen	1,00	-
-	Mustantaso	0,00	-
-	Väribursti	0,00	0,40
-	Synk. taso	0,40	-

2.6.1. Grafiikka I -tila

Videoprosessori (VDP) on Grafiikka I -tilassa, kun bitit M1, M2 ja M3 videoprosessorirekistereissä 0 ja 1 ovat nollia. Tässä tilassa hahmotaso on jaettu 32 x 24-kokoiseksi hahmopaikkamatriisiksi. Jokainen hahmopaikka muodostuu 8 x 8 kuva-alkiosta. VRAM:issa sijaitsevat taulukot, joita käytetään hahmotason luomiseen. Ne ovat hahmo-, nimi- ja väritaulukot. Ne vaativat muistitilaa 2848 VRAM-tavua. Kuvassa 7 on esitetty näiden taulukkojen sijoittuminen hahmotasolle. Muistitilaa säästyy, jos ei tarvita kaikkia 256 mahdollista hahmonmäärittelyä. Taulukot voivat peittää toisiaan, jolloin saadaan pienennettyä kuvion generointiin tarvittavaa VRAM-tilaa.



Kuva 7. Hahmografiikan nimitaulukon kartoitus



Kuva 8. Grafiikka I -tilan kartoitus

Hahmotaulukko sisältää kirjaston, jonka hahmoja voidaan näyttää hahmopaikoissa. Sen pituus on 2048 tavua ja se on järjestetty 256 hahmoon, joista jokaisen pituus on kahdeksan 8 x 8 -bittistä tavua. Kaikki "1"-bitit 8-tavuisessa hahmossa voivat määrätä yhden värin (väri 1), kun taas kaikki 0-bitit voivat määrätä toisen värin (väri 0).

Koko 8-bittistä hahmonimeä käytetään kun valitaan yksi hahmomäärittely hahmotaulukosta. Taulukko on 2048-tavuinen lohko VRAM:ssa. Taulukon alkuosoitteen määrittelee sen perusosoite videoprosessorin rekisterissä 4. Perusosoite muodostaa jokaiselle hahmotaulukon määrittelylle 14-bittisen VRAM-osoitteen kolme eniten merkitsevää bittiä. Seuraavat kahdeksan bittiä ilmaisevat valitun hahmomäärittelyn 8-bittisen nimen. VRAM-osoitteen kolme alinta bittiä ilmaisevat rivinumeron hahmomäärittelyssä.

Rivi/tavu	Sarake (hahmo)						Bitti (hahmomäärittely)							
	0	1	2	3	4	5	0	1	2	3	4	5	6	7
0	C	C	C	C	C	C	0	1	1	1	1	1	0	0
1						C	0	0	0	0	0	1	0	0
2						C	0	0	0	0	0	1	0	0
3			C	C	C	C	0	0	1	1	1	1	0	0
4						C	0	0	0	0	0	1	0	0
5						C	0	0	0	0	0	1	0	0
6		C	C	C	C	C	0	1	1	1	1	1	0	0
7							0	0	0	0	0	0	0	0

Kuva 9. Hahmonäytön kartoitus

Jokainen 256:sta mahdollisesta 8 x 8 -matriisiin hahmomäärittelystä tarvitsee 8 tavua. Ensimmäinen tavu määrittelee hahmon ensimmäisen rivin ja toinen tavu vastaavasti toisen rivin. Jokaisen 8 tavun ensimmäinen bitti määrittelee kuvion ensimmäisen sarakkeen. Muut rivit ja sarakkeet määritellään vastaavalla tavalla. Jokainen bitinsyöttö hahmomäärittelyssä valitsee hahmolle toisen kahdesta värimahdollisuudesta. Bitti "1" valitsee värikoodin (väri 1), joka sisältyy neljään eniten merkitsevään bittiin vastaavassa väritaulukon tavussa. Bitti "0" valitsee toisen värikoodin (väri 0). Kuvassa 9 on esimerkki hahmomäärittelyn kartoituksesta.

Nollien ja ykkösten värin määrittelee väritaulukko, joka sisältää 32 tuloa ja joista jokainen on yhden tavun mittainen. Jokainen tulo määrittelee kaksi väriä; tulon neljä eniten merkitsevää bittiä määrittelevät värin "1":ille ja neljä vähiten merkitsevää bittiä "0":ille. Väritaulukon ensimmäinen tulo määrittelee värit hahmoille nolasta seitsemään (0 - 7) ja seuraava hahmoille kahdeksasta viiteentoista (8 - 15) jne (katso taulukkoa 4). Tällä tavoin voidaan näyttää samanaikaisesti 32 eri väriä.

Nimitaulukko sijaitsee viereisessä 768-tavun VRAM-lohkossa alkaen 1 Ktavun rajalta. Videoprosessorin rekisterin 2 4-bittisen nimitaulukon perusosoitekenttä määrittää nimitaulukon alkuosoitteen. Perusosoite muodostaa 14-bittisen VRAM-osoitteen neljä ylintä bittiä. Alemmat 10 bittiä saadaan rivi- ja sarakelaskureista.

Tavu n:o	Hahmo n:o	Tavu n:o	Hahmo n:o
0	0... 7	16	126...135
1	8... 15	17	136...143
2	16... 23	18	144...151
3	24... 31	19	152...159
4	32... 39	20	160...167
5	40... 47	21	166...175
6	48... 55	22	176...183
7	56... 63	23	184...191
8	64... 71	24	192...199
9	72... 79	25	200...207
10	80... 87	26	208...215
11	88... 95	27	216...223
12	96...103	28	224...231
13	104...111	29	232...239
14	112...119	30	240...247
15	120...127	31	248...256

Taulukko 4. Grafiikka I-tilan väritaulukko

Jokainen tavusyöttö nimitaulukossa on joko hahmon määrittelyn nimihahmotaulukossa tai viittaus tähän. 8-bittisen nimen viisi ylintä bittiä yksilöivät hahmon värijoukon. Kahdeksan hahmon joukkoja on 32. Kaikissa ryhmän kahdeksassa hahmossa käytetään samaa kahta väriä; värikoodit on tallennettu VDP:n väritaulukkoon. Väritaulukko sijaitsee VRAM:issa 32-tavuisessa kellossa alkaen 64-tavun rajalta. Taulukon alkuosoitteen määrää 8-bittinen väritaulukon perusosoite, joka on videoprosessorin rekisterissä 3. Tämä perusosoite muodostaa 14-bittisen väritaulukkosyötön VRAM-osoitteen kahdeksan ylintä bittiä. Seuraava bitti on 0 ja alimmat viisi bittiä ovat samat kuin vastaavan nimitaulukkosyötön viisi ylintä bittiä.

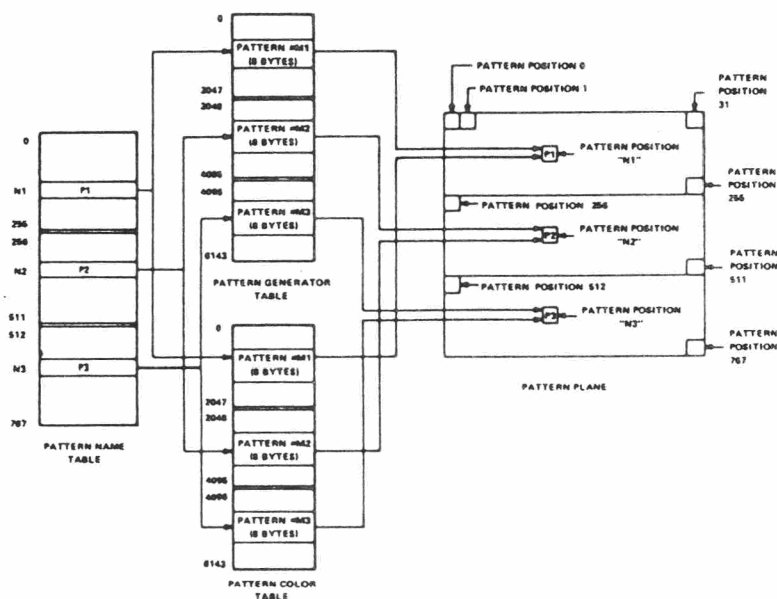
Koska videoprosessorirekisterit määrittelevät VRAM:issa olevien taulukoiden perusosoitteet, voidaan taulukon arvot muuttaa yksinkertaisesti muuttamalla arvot VDP-rekistereissä. Tämä on erittäin käyttökelpoista silloin kun halutaan jakaa aikaa kahden tai useamman grafiikkanäytön välillä.

Kun hahmotaulukkoon on ladattu hahmojoukko, voi nimitaulukon sisällön käsittely muuttaa kuvaruudun tulostusta. Hahmojen dynaaminen muuttaminen grafiikkajakson kuluessa on helposti toteutettavissa, koska kaikki taulukot ovat VRAM:issa. Kaikki VRAM:in 2848 tavua tarvitaan hahmo-, nimi-, väri- ja generointitaulukoille. Muistitilaa tarvitaan vähemmän jos kaikkia 256 mahdollista hahmomäärittelyä ei tarvita; taulukot voivat peittää toisiaan jolloin hahmon generointiin tarvittava VRAM-määrä pienenee.

2.6.2. Grafiikka II-tila

Videoprosessori on grafiikka II-tilassa kun bitit M1 = 0, M2 = 0 ja M3 = 1. Tähän tilaan päästään GR-käskyllä. Grafiikka II-tila vastaa muuten grafiikka I-tilaa paitsi että edellinen mahdollistaa suuremman hahmokitastoin, että kuvaruudun jokaiselle 768:lla (32 x 24) hahmopaikalla voi olla oma hahmosyöttö. Lisäksi jokaiseen 8 x 8-grafiikkahahmoon sisältyy enemmän väri-informaatiota. Näin voidaan jokaiselle 8 x 8-hahmon tavulle määrätä kaksi väriä. Jotta koko grafiikka II-tila olisi käytössä, tarvitaan 12 Ktavua VRAM:ia.

Myös grafiikka II-tilan nimitaulukko sisältää 768 syöttöä, jotka vastaavat 768 hahmopaikkaa kuvaruudulla. Koska grafiikka I-tilan nimet ovat vain 8-bitin mittaisia, voidaan osoittaa enintään 256 hahmonmäärittelyä. Grafiikka II-tila jakaa kuvaruudun kolmeen yhtäsuureen 256 hahmopaikkaa sisältävään osaan ja jakaa myös hahmotaulukon kolmeen yhtäsuureen lohkoksi, joissa jokaisessa on 2048 tavua. Hahmomäärittelyt kuvaruudun ensimmäisessä kolmanneksessa vastaavat ylemmän kolmanneksen hahmopaikkoja. Samalla tavoin hahmonmäärittelyt hahmotaulukon toisessa ja kolmannessa lohossa vastaavat hahmotason toista ja kolmatta aluetta.



Kuva 10. Grafiikka II-tilan kartoitus

Nimitaulukko on myös jaettu kolmeen 256 nimeä käsittävään lohkoon siten että nimet, jotka sijaitsevat hahmonmäärittelyjen ylemmässä kolmannessa, löytyvät hahmotaulukon ylemmästä 2048-tavun lohkoista. Vastaavasti toinen ja kolmas lohko viittaavat hahmonmäärittelyihin toisessa ja kolmannessa 2048-tavun lohkoissa. Siten jos 768 hahmoa ovat yksityiskohtaisesti määritellyt, käytetään 8-bittistä hahmonimeä kolme kertaa, ts. kerran jokaisessa nimitaulukon lohkoissa. Hahmotaulukko mahtuu 8 Ktavuun ja se voidaan sijoittaa 16K:n muistin ylä- tai alapuoliskoon riippuen hahmotaulukon perusosoitteen eniten merkitsevistä bitistä videoprosessorin rekisterissä 4. Vähiten merkitsevät bitit on asetettava kaikille "1":ksi.

Väritaulukon pituus on 6144 tavua ja se on jaettu kolmeen yhtäsuureen 2048-tavun lohkoon. Jokainen syöttö väritaulukossa on 8-tavuinen, ja niiden avulla on mahdollista yksityiskohtaisesti määrätä värit 1 ja 0 jokaiselle vastaavalle hahmonmäärittelyn kahdeksalle tavulle. Osoitustapa on samanlainen kuin hahmotaulukolle lukuunottamatta taulukon sijaintia VRAM:issa. Tätä voidaan ohjata väritaulukon perusosoitteen eniten merkitsevän bitin latauksella prosessorin rekisterissä 3. Vähiten merkitsevät bitit on asetettava kaikille "1":ksi.

Kuva 10 esittää grafiikka II-tilan karttaa. Hahmonimet P1, P2 ja P3 vastaavat hahmo-syöttöjä hahmotaulukon kolmeen lohkoon. Huomaa myös miten nämä kolme nimeä sijoittuvat näytössä. Kuvassa 11 on esimerkki hahmo- ja väritaulukkojen syötöistä.

Rivi\0	0	1	0	0	0	0	0	1	8	1	8	8	8	8	8	1	0	3	4	7	o Rivi			
1	0	0	1	0	0	0	1	0	8	8	7	8	8	8	7	8	1	1 (BLACK)	8 (LT. YELLOW)	1				
2	0	0	0	1	0	1	0	0	8	8	8	C	8	C	8	8	2	7 (CYAN)	8 (LT. YELLOW)	2				
3	0	0	0	0	1	0	0	0	8	8	8	8	E	8	8	8	3	C (GREEN)	8 (LT. YELLOW)	3				
4	0	0	0	0	1	0	0	0	8	8	8	8	8	8	8	8	4	E (GRAY)	8 (LT. YELLOW)	4				
5	0	0	0	0	1	0	0	0	8	8	8	8	8	5	8	8	5	8 (MED. RED)	8 (LT. YELLOW)	5				
6	0	0	0	0	1	0	0	0	8	8	8	8	8	6	8	8	6	5 (LT. BLUE)	8 (LT. YELLOW)	6				
7	0	0	0	0	1	0	0	0	8	8	8	8	8	D	8	8	7	6 (DK. RED)	8 (LT. YELLOW)	7				
																		D (MAGENTA)	8 (LT. YELLOW)					
Hahmotaulukko																	Hahmo				Väritaulukko			

Hahmotaulukko

Hahmo

Väritaulukko

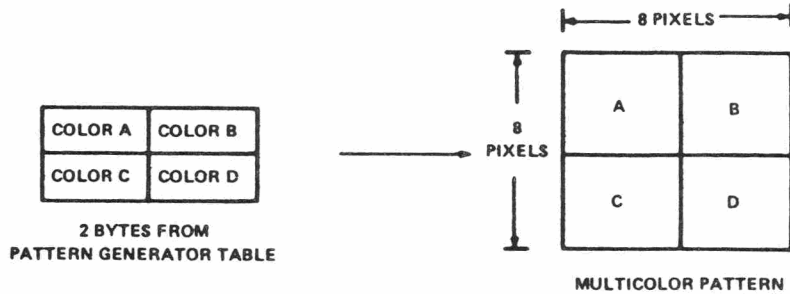
Kuva 11. Hahmonäytön kartoitus

2.6.3. Moniväritila

Videoprosessori on moniväritilassa, kun tilan bitit ovat M1 = 0, M2 = 1 ja M3 = 0. Moniväritila tarjoaa rajoittamattoman 64 x 48-värineliönäytön. Jokainen värineliö koostuu 4 x 4-lohkoisesta kuva-alkiosta. Jokaisen neliön väri voi olla mikä tahansa video-näytön 15 väristä tai läpinäkyvä. Kaikkia värejä voidaan käyttää samanaikaisesti moniväritilassa. Taustaverho ja sprite-tasot ovat edelleen aktiivisia moniväritilassa.

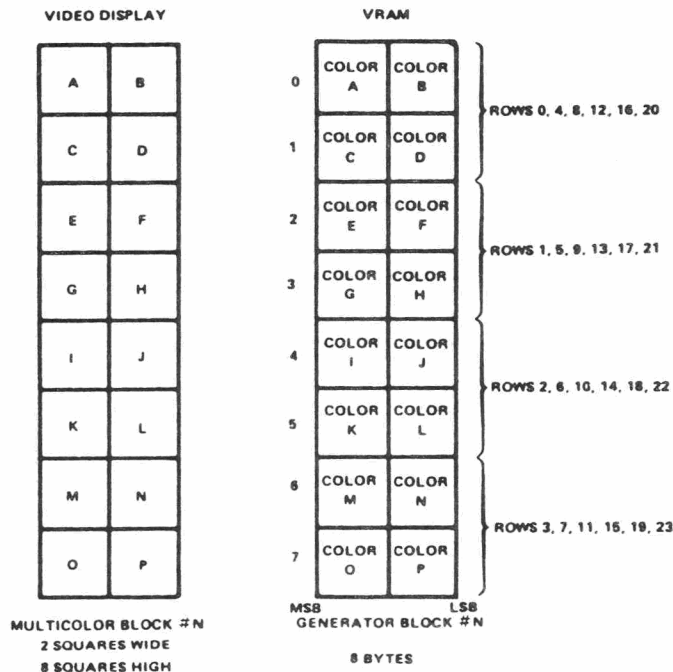
Moniväritilan nimitaulukko on sama kuin grafiikkatilassa koostuen 768 nimisyötöstä, vaikka nimi ei enää osoita värilistaan. Värin määrittely saadaan nyt hahmotaulukosta. Nimi osoittaa 8-tavuiseen VRAM-lohkoon hahmotaulukossa.

Vain kahta tavua 8-tavuisesta lohkoista käytetään kuvaruutunäytön määrittelemiseen. Nämä kaksi tavua määräävät ne neljä väriä, jotka miehittävät 4 x 4-kuva-alkion alan. Ensimmäisen tavun neljä eniten merkitsevää bittä osoittavat värin monivärihahmon ylemmälle vasemmanpuoleiselle neljännekselle, ja vähemmän merkitsevät bitit osoittavat värin ylemmälle oikeanpuoleiselle neljännekselle. Vastaavasti toinen tavu määrittelee monivärihahmon alempien neljännesten värit. Kahden tavun avulla voidaan näin kartoittaa 8 x 8-kuva-alkion monivärinen hahmo, kuva 12.



Kuva 12. Monivärihahmon kartoitus

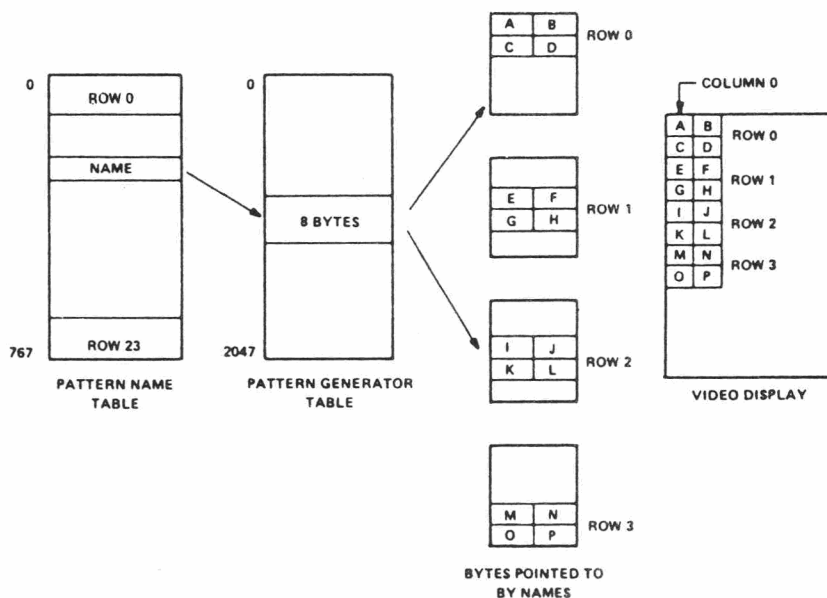
8-tavun lohkon sisältyvien kahden tavun sijainti riippuu nimen paikasta kuvaruudulla, sillä nimi osoittaa tavuihin. Ylimmässä rivissä oleville nimille (nimet 0 - 31) nämä kaksi tavua ovat 8-tavun lohkon ensimmäiset kaksi tavua. Seuraavan rivin nimet (32 - 63) käyttävät 8-tavun lohkon seuraavia kaksi tavua, seuraava rivi tavuja 5 ja 6 sekä viimeinen rivi tavuja 7 ja 8. Tämä sarja toistuu kuvaruudun muille osille.



Kuva 13. Monivärinen näyttö

Esimerkiksi kuvassa 13 jos nimitaulukon syöttö on \emptyset (hahmopaikka \emptyset) ja monivärilohko on $\#N$ (nimi = N), monivärinen hahmonäyttö koostuu 8 x 8-kuva-alkiosta, joiden värit ovat A, B, C ja D. Nämä värit sisältävät moniväritaulukon kaksi ensimmäistä tavua. Jos kuitenkin nimi $\#N$ sijaitsee nimitaulukon tulossa 33 (hahmopaikka 33), esiintyvät näytössä värit E, F, G ja H kuten monivärilohkon tavut 3 ja 4 määrittelevät. Nimi viittaa näihin tavuihin.

Samoin riveillä 2 ja 3 olevat hahmopaikat aiheuttavat vastaavasti värit I, J, K ja L ja värit M, N, O ja P tulostukseen. Siten moniväritaulukon lohkon muodostama väri riippuu hahmopaikasta näytöllä. Kuva 14 esittää moniväritilan kartoitusta.



Kuva 14. Moniväritilan kartoitus

VRAM:n sisällön kartoitusta kuvaruudulle on yksinkertaistettu käyttämällä kaksinkertaisia nimiä nimitaulukossa, koska 8-tavun lohkoissa käytetyt tavusarjat määrittelevät kuvaruudulle 2 x 8-kokoisen värineliöhahmon suorana siirtana 8-tavun VRAM-lohkosta, johon osoitetaan yleisellä nimellä.

Tällä tavalla voidaan 768 tavua käyttää vielä nimitaulukoon ja 1536 tavua käytetään väri-informaatioon hahmotaulukossa (24 riviä x 32 saraketta x 8 tavua/ hahmopaikka). Siten kaikki VRAM:in 1728 tavua tarvitaan. Huomattavaa on, että taulukot alkavat 1 K:n ja 2 K:n rajoilta eivätkä siten ole lähekkäin.

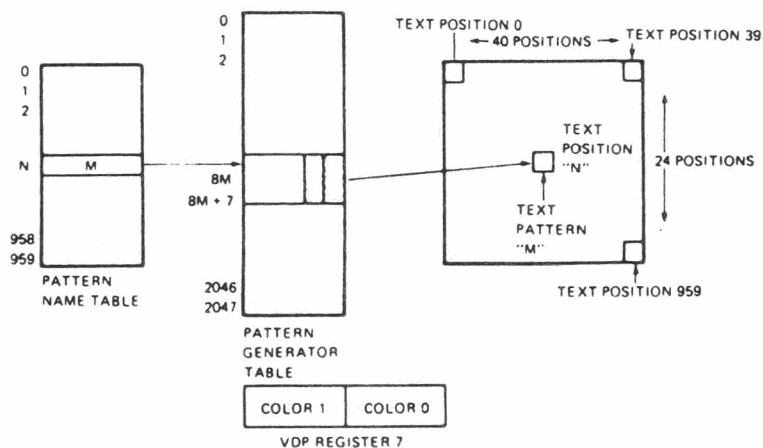
2.6.4. Tekstitila

Videoprosessori on tekstitilassa kun tilan bitit $M1 = 1$, $M2 = 0$ ja $M3 = 0$. Tässä tilassa on kuvaruutu jaettu 40×24 -tekstipaikkaan (kuva 15). Jokainen tekstipaikka sisältää 6 kuva-alkiosaraketta ja 8 riviä. Ne taulukot joiden avulla muodostetaan hahmotaso, ovat nimitaulukko ja hahmotaulukko. Joka hetki voi olla määriteltynä 256 erillistä hahmoa. Hahmomäärittelyt on tallennettu hahmotaulukkoon VRAM:issa ja niitä voidaan muuttaa dynaamisesti. VRAM sisältää hahmotaulukon joka kartoittaa hahmomäärittelyt jokaiseen 960:een alkioon hahmotasolla, kuva 16. Sprite-hahmoja ei voida muodostaa tekstitilassa.



Kuva 15.

Kuten grafiikkatilassakin sisältää hahmotaulukko tekstihahmokirjaston, jonka merkkejä voidaan näyttää tekstipaikoilla. Kirjasto on 2048 tavun mittainen ja siihen on järjestetty 256 tekstihahmoa joiden pituus on 8 tavua. Koska jokainen tekstipaikka kuvaruudulla on 6 kuva-alkion levyinen, niin jokaisen tekstihahmon kaksi vähiten merkitsevää bittiä jätetään ottamatta huomioon. Silloin saadaan 6×8 -bittiä jokaiseen tekstihahmoon. Jokainen 8-tavuinen lohko määrittelee tekstihahmon, jonka kaikki bitit "1" määräävät yhden värin kuvaruutunäytössä ja bitit "0" puolestaan toisen värin. Nämä värit valitaan lataamalla videoprosessorin rekisteriin 7 värit 0 ja 1 vastaavasti oikean- ja vasemmanpuoliseen lohkoon (vrt. kappale 2.4. sivulla 61).



Kuva 16. VRAM:in kartoitus hahmotasoihin tekstitilassa

Tekstitilassa nimitaulukko määrittelee tekstihahmon paikan kuvaruudulla, kuva 16. Nimitaulukossa on 960 tuloa ja jokaisen pituus on 1 tavu. Kuvaruudun tekstihahmo-paikkojen ja nimitaulukon ($40 \times 24 = 960$) syöttöjen välillä on vastaavuus. Ensimmäiset 40 tuloa vastaavat kuvaruudun tekstihahmojen ylintä riviä, seuraavat 40 toista riviä jne. Syötön arvo nimitaulukossa osoittaa mikä 256:sta tekstihahmosta tulee sijoittaa tähän kohtaan hahmotasolla.

Nimitaulukko on sijoitettu viereiseen 960-tavun lohkon VRAM:issa ja se alkaa 1 Ktavun rajalta. Nimitaulukon alkuosoitteen määrää 4-bittinen nimitaulukon perusosoite videoprosessorin rekisterissä 2. Perusosoite muodostuu 14-bittisen VRAM-osoitteen neljästä ylimmästä bitistä. VRAM-osoitteen 10 alinta bittiä osoittavat yhteensä 960:stä hahmo-paikasta. Rivit järjestävät nimitaulukon. Kappaleessa 4 on annettu esimerkki nimitaulukon osoitustavasta.

Jokainen tavusyöttö nimitaulukossa on osoitin hahmotaulukon määrittelyyn. Samaa kahta väriä käytetään kaikille 256:lle hahmolle. Värikoodit on tallennettu videoprosessorin rekisteriin 7.

Kuten nimikin viittaa, on tekstitila tarkoitettu pääasiassa tekstisovellutuksiin ja etenkin sellaisiin, joissa grafiikkatilan 32 hahmoa rivillä on riittämätön. Tekstitilan etuna on että siinä riville saadaan mahtumaan 8 hahmoa lisää ja haittana ettei siinä voida käyttää sprite-kuvia. Koko kuvaruudulle voidaan saada vain kaksi väriä.

Tekstitilassa käytettyä tekstihahmoa on mahdollista käyttää myös Grafiikka I-tilassa. Tämä voidaan tehdä jos varmistetaan siitä, että kaikkien merkkihahmojen kaksi vähiten merkitsevää bittiä ovat "0". Siten siirtyminen tekstitilasta hahmotilaan saa aikaan merkkien välisen tyhjän tilan levenemisen ja vähentää rivillä olevien merkkien määrän 40:stä 32:een. Kun merkkijono on määritelty ja sijoitettu hahmotaulukkaan, voidaan päivittämällä nimitaulukko tuottaa ja käsitellä tekstiä kuvaruudulla.

Kokonaista 8-bittistä hahmonimeä käytetään valittaessa yksi 256:sta hahmotaulukon määrittelystä. Taulukko on 2048-tavuinen lohko VRAM:issa alkaen 2 Ktavun rajalta. Taulukon alkuosoitteen määrää hahmotaulukon perusosoite videoprosessorin rekisterissä 4. Perusosoite muodostaa 14-bittisen VRAM-osoitteen 3 eniten merkitsevää bittiä jokaiselle hahmotaulukon syötölle. Seuraavat 8-bittiä ovat valitun hahmomäärittelyn 8-bittinen nimi. VRAM-osoitteen 3 alinta bittiä ovat hahmomäärittelyn rivinnumero.

Jokaiselle 256 mahdolliselle 6×8 -kokoiselle hahmonäytölle tarvitaan 8 tavua. Ensimmäinen tavu määrittelee hahmon ensimmäisen rivin ja vastaavasti toinen tavu toisen rivin. Kummankaan tavun kahta vähiten merkitsevää bittiä ei käytetä. Suositellaan kuitenkin että nämä bitit asetettaisiin "0":ksi. Jokaisen bitin syöttö hahmomäärittelyyn valitsee toisen hahmon kahdesta väristä.

Nimitaulukko vaatii kaikki 3008 VRAM-tavua. Jos kaikkia 256 mahdollista hahmomäärittelyä ei tarvita, säästetään muistitilaa. Taulukot voivat olla päällekkäin jolloin hahmojen muodostamiseen tarvittava VRAM-määrä pienenee.

2.6.5. Sprite-hahmot

Videonäytöllä voi olla enintään 32 spritea korkeimman prioriteetin omaavilla videotasoilla. Spritet ovat erityisiä animaatiohahmoja joita voidaan liikuttaa pehmeästi kuvaruudulla ja sijoittaa useita päällekkäin. Spriten sijainnin määrää sprite hahmon vasen yläkulma. Sprite-hahmoja voidaan helposti liikuttaa kuva-alkio kerrallaan määrittämällä uudelleen spriten alkupiste. Sprite-hahmot eivät ole aktiivisia tekstiilassa. 32 sprite-tasoa ovat täysin läpinäkyviä itse sprite-hahmon ulkopuolella.

VRAM:issa sijaitsevat alalohkot, jotka ovat sprite-attribuuttitaulukko ja sprite-hahmotaulukko, määrittelevät spritet. Nämä taulukot ovat samanlaisia kuin vastaavat hahmotaulukot. Sprite-attribuuttitaulukko määrittelee sprite-hahmon liikkeen kuvaruudulla kun taas sprite-hahmotaulukko kuvaa sprite-hahmon ulkonäköä. Sprite-hahmojen muodot on esitetty taulukossa 5.

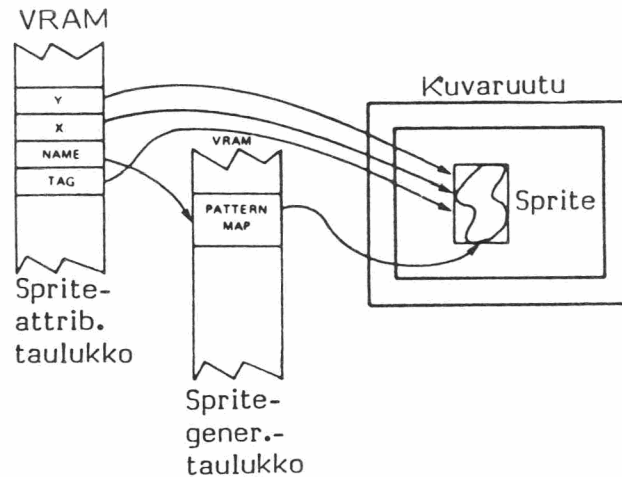


Kuva 17. Sprite-attribuuttitaulukon syöttö

KOKO	MAG	ALUE	EROTTELUKYKY	TAVUT/HAHMO
Ø	Ø	8 x 8	yksi kuva-alkio	8
1	Ø	16 x 16	yksi kuva-alkio	32
Ø	1	16 x 16	2 x 2 kuva-alkiota	8
1	1	32 x 32	2 x 2 kuva-alkiota	32

Taulukko 5. Sprite-hahmon muodot

7-bittisen attribuuttitaulukon perusosoite VDP:n rekisterissä 5 määrittelee hahmotaulukon alkuosoitteen. Perusosoite määrittelee 14-bittisen VRAM-osoitteen 7 ylintä bittiä. VRAM-osoitteen seuraavat viisi bittiä ovat samat kuin sprite-numerossa. Kaksi alinta bittiä valitsevat jokaiselle spritelle yhden neljästä tavusta sprite-2-attribuuttitaulukkosyötössä. Taulukon jokainen syöttö sisältää neljä tavua jotka määrittävät sprite-paikan, sprite-hahmon nimen ja värin kuten kuvassa 17 on esitetty.



Kuva 18. Sprite-kartoitus

Tämä kuva esittää tapaa jolla VRAM-taulukot kartoittavat spritet näytöllä. Koska näytölle voidaan saada 32 spritea, on sprite-attribuuttitaulukossa 32 tuloa. Jokainen tulo muodostuu neljästä tavusta. Tulot on järjestetty niin että ensimmäinen tulo vastaa spritea tasolla 0, seuraava spritea tasolla 1 jne. Sprite-attribuuttitaulukon koko on $4 \times 32 = 128$ ja se sijaitsee viereisessä 128-tavun lohossa VRAM:issa alkaen 128-tavun rajalta.

Sprite-attribuuttitaulukon jokaisen tulon kaksi ensimmäistä tavua määrittelevät spriten paikan kuvaruudulla. Ensimmäinen tavu ilmaisee pystysuunnassa spriten etäisyyden kuvaruudun yläreunasta. Etäisyyden yksikkönä on kuva-alkio. Tämä etäisyys määrittää spriten että arvo -1 asettaa kuvaruudun yläreunassa olevan spriten kosketukseen taustaverhoalueen kanssa. Toinen tavu kuvaa spriten siirtymää vaakasuunnassa kuvaruudun vasemmasta reunasta lukien. Arvo 0 sijoittaa spriten taustaverhon vasenta kulmaa vasten. Huomaa että kaikki mittaukset luetaan ylimmästä vasemmanpuolisesta spritekuva-alkiosta lähtien.

Kun tulon kaksi ensimmäistä tavua sijoittavat sprite-hahmon siten että se ulottuu yli taustaverhon, niin se osa spritea joka on taustaverhon alueella, näytetään normaalisti. Taustaverhon ylimenevä sprite-osa jää pois näkyvistä. Tämä antaa käyttäjälle mahdollisuuden siirtää sprite-kuvia kuvaruudulle taustaverhon takaa.

Siirtyminen on osittain riippuvainen etumerkistä ensimmäisessä tavussa. Jos arvot pystysuoralle siirtymälle ovat välillä -31 ... 0 (E116 ... 0), liikkuu sprite-hahmo taustaverhon yläkulmasta lähtien. Vastaavasti vaakasuoran siirtymän arvot arvon 255 läheisyydessä sallivat spriten liikkeen kuvaruudun oikeasta laidasta. Jos halutaan sprite-hahmojen liikkuvan taustaverhon vasemmasta kulmasta lähtien, on käytettävä erityisbittiä sprite-attribuuttitaulukon syötön kolmannessa tavussa.

Sprite-attribuuttitaulukon syötön kolmas tavu sisältää osoittimen sprite-hahmotaulukkoon joka määrittelee, millaiselta sprite-hahmon tulee näyttää. Tämä on 8-bittinen osoitin sprite-hahmomäärittelyyn. Sprite-nimi on samanlainen kuin nimi grafiikkatilassa.

Sprite-attribuuttitaulukon syötön neljäs tavu sisältää neljässä alimmassa bitissään sprite-hahmon värin (katso taulukkoa 3 värimäärittelyille). Eniten merkitsevä bitti on EC-bitti (Early Clock, klo-bitti kuvassa 17). Jos se asetetaan "0":ksi, tämä bitti ei aiheuta mitään. Kun se asetetaan "1":ksi, spriten vaakatasopaikka siirtyy 32 kuva-alkiota vasemmalle. Tämä mahdollistaa spriten liikkeen taustaverhon vasemmasta kulmasta lähtien. Vaakasuoran liikkeen arvot (toinen tavu syötössä) väliltä 0 - 32 aiheuttavat sen että sprite peittää taustaverhon vasemman rajan.

Sprite-hahmotaulukon pituus on enintään 2048 tavua ja se alkaa 2 Ktavun rajalta. Taulukko on jaettu 256 lohkoksi, joissa kaikissa on 8 tavua. Sprite-attribuuttitaulukon tulo kolmas tavu määrittelee, mitä 8-tavuista lohkoa tulee käyttää määrittäessä sprite-hahmoja. Sprite-hahmotaulukossa "1":set aiheuttavat sprite-hahmon määrittelyn kyseisessä pisteessä kun taas "0":t saavat aikaan läpinäkyvän pisteen. Taulukon alkuosoitteen määrää sprite-hahmotaulukon perusosoite VDP:n rekisterissä 6. Perusosoite antaa 14-bittisen VRAM-osoitteen kolme eniten merkitsevää bittiä. Osoitteen seuraavat kahdeksan bittiä muodostavat sprite-nimen ja kolme viimeistä bittiä antavat sprite-hahmon rivinumeron. Osoitteenmuodostus poikkeaa vain vähän SIZE₁-spriteista (size = koko, vrt. taulukko 5 sivulla 78).

Yhdellä vaakarivillä voidaan esittää enintään neljä sprite-hahmoa. Jos tätä sääntöä ei noudateta, tulostuu näytössä normaalisti neljä suurimman prioriteetin omaavaa spritea. Viides ja sitä seuraavat spritet jätetään tulostamatta tällä rivillä. Edelleen, viidenneksi bitiksi videoprosessorin tilarekisteriin asetetaan bitti "1" ja viidennen spriten numero ladataan tilarekisteriin (katso kappale 2.5.).

Jos halutaan, voidaan käyttää suurempia sprite-hahmoja kuin ovat 8 x 8-kuva-alkiolliset. VDP:n rekisterin 1 MAG- ja SIZE-bittejä käytetään valittaessa haluttuja toimintoja.

MAG = 0, SIZE = 0: Valintoja ei ole tehty.

MAG = 1, SIZE = 0: Sprite-hahmotaulukko käyttää sprite-kuvaukseen 8 tavua. Jokainen bitti taulukossa sijoittuu kuvaruudulle 2 x 2-kuva-alkioon aiheuttaen sprite-koon kaksinkertaistumisen 16 x 16-kokoiseksi.

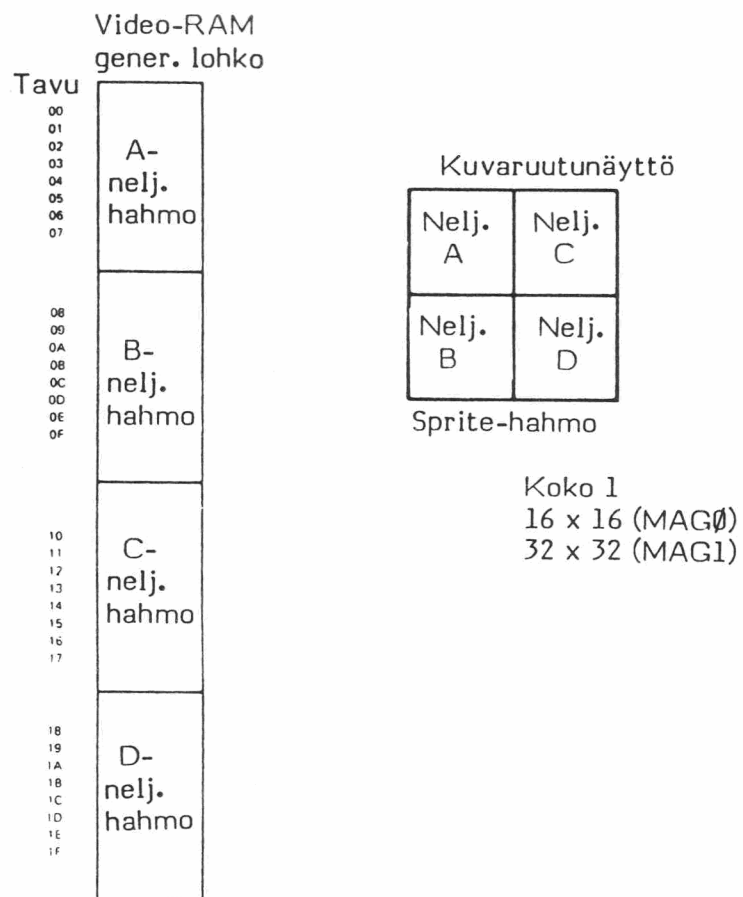
MAG = 0; SIZE = 1: Sprite hahmotaulukko käyttää 31 tavua määrittelemään sprite-hahmoa ja tuloksena on 16 x 16-kuva-alkiomainen sprite. 32 tavun sijoittumista sprite-hahmoon on esitetty kuvassa 19. Kartoituksessa yksi bitti vastaa yhtä kuva-alkiota.

MAG = 1, SIZE = 1: Sama kuin valinta MAG = 0 ja SIZE = 1 paitsi että jokainen bitti kartoittaa 2 x 2-kuva-alkion alaa, jolloin sprite-koko on 32 x 32.

Videoprosessorin avulla voidaan tarkistaa sprite-hahmojen päällekkäisyyttä. Päällekkäisilalippu VDP:n tilarekisterissä asetetaan "1":ksi joka kerta kun kahdella aktiivisella spritellä on "1"-bitti samassa paikassa kuvaruudulla.

Sprite-toiminta päättyy, jos VDP löytää arvon 208 (D016) mistä tahansa syötöstä pystykenttään sprite-attribuuttitaulukossa. Tämä sallii sprite-attribuuttitaulukon lyhentämisen vaadittavaan minimikokoon ja sallii myös käyttäjän sulkea pois osan tai kaikki spritet yksinkertaisesti muuttamalla yhtä tavua VRAM:issa.

Tarvitaan kaikkiaan 2178 VRAM-tavua sprite-nimi- ja hahmotaulukoihin. Vaaditaan huomattavasti vähemmän muistitilaa jos kaikkia 256 mahdollista sprite-hahmomäärittelyä ei tarvita. Sprite-attribuuttitaulukkoa voidaan myös lyhentää kuten edellisessä kappaleessa kuvattiin. Taulukot voivat osin peittää toisiaan jolloin sprite-hahmojen generointiin vaadittava VRAM-määrä pienenee.

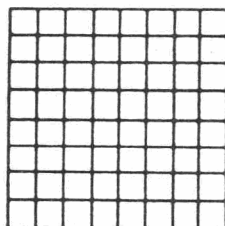


Kuva 19. SIZE 1 sprite-kartoitus

2.6.6. Hahmot ja spritet kuva-alkio kerrallaan

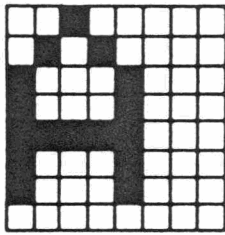
HAHMOT

1. Käytetään kuvan A esittämää 8 x 8-kokoista mallia. Jokainen mallin ruutu vastaa yhtä kuva-alkiota kuvaruudulla.

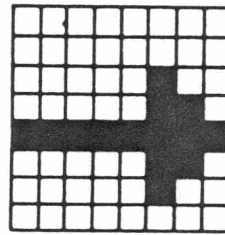


Kuva A.

2. Tummennetaan teksti- tai grafiikkahahmoihin tarvittavat ruudut. Kuvissa B ja C on esimerkkihahmoina kirjain A ja nuolikuvio.



Kuva B

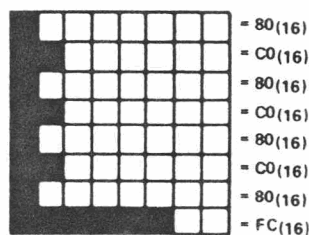
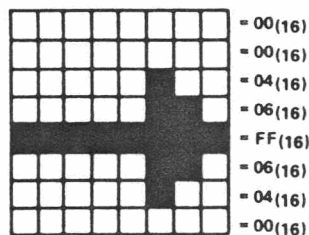
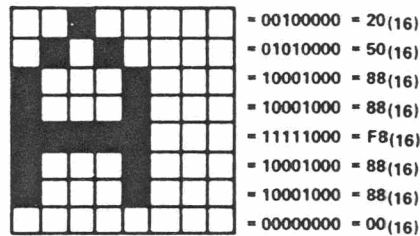


Kuva C

Huom.

Jos näitä hahmoja käytetään tekstitilassa (40 hahmoa rivillä), niin hahmojen tulisi olla vasemmalle tasattuna 6 x 8-ruutuisessa lohkoissa, kuva B. Jos kaikki tekstihahmot ovat 6 x 8-kokoisia, niitä voidaan käyttää sekä Teksti- että Grafiikka I- ja Grafiikka II-tilassa.

3. Sijoitetaan "1" kuvaamaan täytettyä ruutua ja "0" kuvaamaan tyhjää ruutua. Sen jälkeen muutetaan kaikki bitit vastaaviksi heksadesimaaliluvuiksi, kuva D.



Kuva D

4. Seuraavaksi sijoitetaan hahmon määrittelevät 8 tavua hahmotaulukkoon. Oletetaan että hahmotaulukon alalohko on sijoitettu osoitteeseen 800_{16} ja nuolikuviolle on määritetty nimi 00_{16} . Kahdeksan hahmotasoa asetetaan seuraavasti:

800			
801	00		
802	04		
803	06		
804	FF		
805	06		
806	04		
807	00		
808			
809			
80A			
80B			
80C			
80D			
80E			
80F			
810			
900	00		
901	00		
902	00		
903	00		
904	00		
905	00		
906	00		
907	00		
908			
A08	20		
A09	50		
A0A	88		
A0B	88		
A0C	F8		
A0D	88		
A0E	88		

Hahmo-
nimi 00

Hahmo-
nimi 01

Hahmo-
nimi 20

Hahmo-
nimi 41

Huom.

Kun tulostussovellutuksessa käytetään tekstiä, voidaan tekstimerkin kahdeksan tavua sijoittaa sen ASCII-numeropaikkoihin.

Esimerkki:

ASCII välilyönti = 20_{16}

? = $3F_{16}$

A = 41_{16}

B = 42_{16}

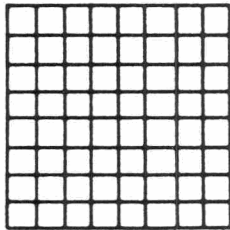
C = 43_{16}

jne.

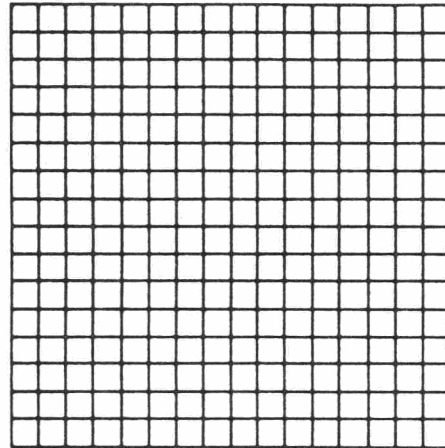
Tämä yksinkertaistaa tekstin kirjoittamista kuvaruudulle. Kirjoitetaan vain suoraan ASCII-nimi hahmotaulukkaan. Välilyöntimerkki on hahmotaulukossa paikassa 20 ja kirjain A hahmonimessä 41.

SPRITET

1. Määritellään, käytetäänkö 8 x 8- tai 16 x 16- sprite-hahmoa. Sen jälkeen käytetään sopivaa ruudukkomallia, kuvat E ja F.

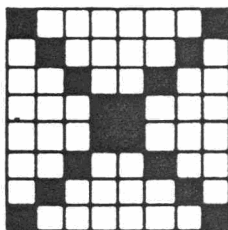


Kuva E

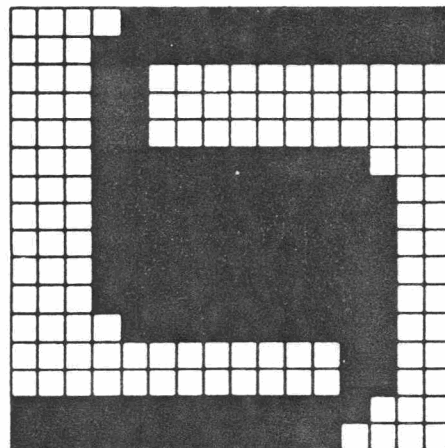


Kuva F

2. Ruutuja täyttämällä luodaan sprite-hahmoja, esimerkkejä kuvissa G ja H.

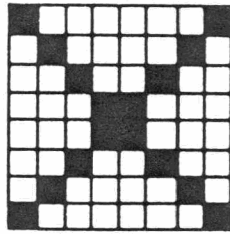


Kuva G

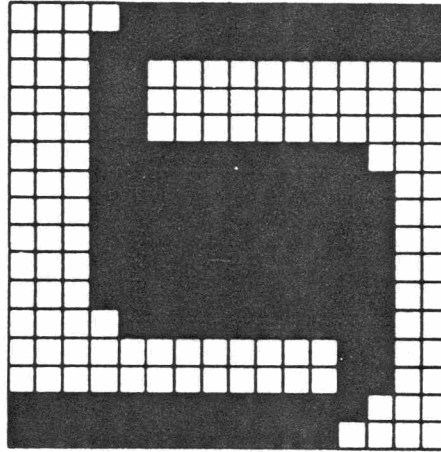


Kuva H

3. Sen jälkeen koodataan sprite-hahmot kuten edellisessä kappaleessakin. 8 x 8-sprite-hahmon koodaus on aivan kuten 8 x 8-hahmoilla muutoinkin, mutta 16 x 16-spritet koodataan kuten kuva J osoittaa, katso myös seur. sivu.



Kuva I



Kuva J

	8 X 8	
000	81	Sprite-nimi 00
001	42	
002	24	
003	18	
004	18	
005	24	
006	42	
007	81	
008		Sprite-nimi 01
009		
00A		
00B		
00C		
00D		
00E		
00F		
010		

Kuva K

	16 X 16	
000	0F	Vasen ylä-kulma
001	1F	
002	30	
003	30	
004	3F	
005	3F	
006	3F	
007	3F	Vasen ala-kulma
008	3F	
009	3F	
00A	1F	
00B	00	
00C	00	
00D	00	
00E	FF	Oikea ylä-kulma
00F	FF	
010	FF	
011	FF	
012	00	
013	00	
014	F0	
015	F8	Oikea ala-kulma
016	F8	
017	F8	
018	F8	
01A	F8	
01B	18	
01C	18	
01D	18	
01E	F8	Sprite-nimi 04
01F	F0	
020	XX	

Kuva L

16 x 16- lohkoiset hahmot jaetaan neljään 8 x 8- kokoisiksi hahmoiksi. Ensiksi koodataan 8 x 8-hahmot edeten vasemmasta yläkulmasta lähtien. Seuraavaksi koodataan vasemmanpuoleinen alempi hahmo ja sen jälkeen oikeanpuoleiset hahmot vastaavassa järjestyksessä.

4. Seuraavaksi sijoitetaan 8 x 8- sprite-hahmojen kahdeksan tavua tai 16 x 16-sprite-hahmojen 32 tavua sprite-hahmotaulukkoon. Oletetaan että sprite-hahmotaulukko sijaitsee osoitteessa 0000. Kuvat K ja L edellisellä sivulla osoittavat miltä taulukot näyttävät 8 x 8- ja 16 x 16- spriteille.

16 x 16-sprite-hahmot alkavat taulukossa ylemmän vasemmanpuoleisen kulman tavulla. Seuraavaksi aloitetaan oikealta ylhäältä ja edetään siitä alaspäin.

3. VRAM-MUISTIOSOITTEET

Seuraavissa taulukoissa esitetään VRAM-osoitteet kaikille videoprosessorin toimintamuodoille. Bittejä kuvaavien linjojen alle on lukemisen helpottamiseksi merkitty nämä kyseistä toimintoa vastaavat bitit.

Grafiikka I -tilan osoitteet

Osoitetyyppi	0	1	2	3	4	5	6	7	8	9	10	11	12	13	Kommentti
1) Hahmon nimi osoite	<div>Nimitaulukko (0-3)</div> <div>Rivi (4-8)</div> <div>Sarake (9-13)</div>														H-nimitaul.(VDP-r2) Hahmon sijainti
2) Hahmon väri osoite	<div>Väritaulukko (0-7)</div> <div>0 8</div> <div>Nimi (0-4) (9-13)</div>														H-väritaul.(VDP-r3) Bitti 8 on aina "0" Nimen 5 eniten merkitsevää bittiä
3) Hahmon gener. osoite	<div>Gener. (0-2)</div> <div>Nimi (3-10)</div> <div>XXX (11-13)</div>														H-gener.(VDP-r4) Nimen kaikki 8 bit 3 alinta bittiä ovat rivisijainnille

Grafiikka II -tilan osoitteet

Osoitetyyppi	0 1 2 3 4 5 6 7 8 9 10 11 12 13	Kommentti
1) Hahmon nimi osoite		H-nimitaul.(VDP-r2) Hahmon rivisijainti Hahmon sarake-sijainti
2) Hahmon väri osoite		H-väritaulukon enit. merk. bitti (VDP-r3) 2 en. merk. bittiä vertikaalilaskurilta Nimen kaikki bitit Väritaulukon tavu/ linja
3) Hahmon gener. osoite		H-nt:n MSB (VDP-r4) Vert.laskurin MSB:t Nimen 8 bittiä Hahmogeneraattorin tavu/linjanumero

Tekstitilan osoitteet

Osoitetyyppi	0 1 2 3 4 5 6 7 8 9 10 11 12 13	Kommentti
Tekstitila nimi osoite		H-nimitaul.(VDP-r2) Tekstin rivisij. x 40 sekä sarakenumero
Tekstitila hahmo osoite		H-gener.(VDP-r4) Nimi Tavu/linjanumero

Sprite-osoitteet

Osoitetyyppi	0 1 2 3 4 5 6 7 8 9 10 11 12 13	Kommentti
Sprite attribuutti osoite		Attr. taul. (VDP-r5) Spriten numero Attrib. numero 00 = vertik. sijainti 01 = horis. sijainti 10 = nimi 11 = EC ja väri
SIZE=0 Sprite-hahmo generaattori		Spr.h-gen.(VDP-r4) Spriten nimiattrib. 3 alinta bittiä tavu/ linjanumerolle
SIZE=1 Sprite-hahmo generaattori		Hahmogen.(VDP-r4) Nimen 6 MSB:tä Size=1 sprite-tavu

Moniväritilan osoitteet

Osoitetyyppi	0 1 2 3 4 5 6 7 8 9 10 11 12 13	Kommentti
4) Moniväri nimi osoite		Nimitaul.(VDP-rek.2) Hahmon rivisijainti Hahmon sarake-sijainti
5) Moniväri nimi osoite		Hahmogen.(VDP-r4) Nimi 3 väh. merk. bittiä tavu/nelj. rivi

3.1. Esimerkki VRAM-osoitteiden määrittelystä

Yleensä käytössä on 256 eri 8x8-hahmoa joilla ei ole hahmoa kohden kahta useampaa väriä ja lisäksi käytetään jopa 32 8x8-sprite-kuvaa. Nämä olosuhteet määräävät mitä videoprosessorin tilaa on käytettävä. Sprite-kuvat ja 8x8-hahmolohkot sulkevat pois Teksti- ja Moniväritilan. Näin ollen valinnan on tapahduttava Grafiikka I- ja Grafiikka II-tilan kesken. Jos ainoana vaatimuksena on kahden värin käyttö kutakin lohkoa kohti, kannattaa valita Grafiikka I-tila sen helppokäyttöisyyden vuoksi. Kuvassa 20 on muistikartta, jossa nämä toiminnot on sijoitettu 4k-muistialueelle.

Kuvan 20 rekisterien arvot ovat seuraavat:

Rekisteri 0 = 00	Ulkoisen videon esto, M3=0.
Rekisteri 1 = C0	Valitaan 16k VRAM, Blank=1, Grafiikkatila I, SIZE=0, MAG=0.
Rekisteri 2 = 01	Nimitaulukon alkuosoite $a > 400$.
Rekisteri 3 = 08	Väritaulukon alkuosoite $a > 0200$.
Rekisteri 4 = 01	Hahmotaulukon alkuosoite $a > 800$.
Rekisteri 5 = 02	Sprite-attribuuttitaulukon alkuosoite $a > 1000$.
Rekisteri 6 = 00	Sprite-hahmogeneraattorin alkuosoite $a > 0000$.
Rekisteri 7 = XX	Käyttäjän määriteltävissä.

0000	Sprite-generaattoritaulukko	32 8x8-hahmoa = 256 tavua
00FF 0100	Sprite-attribuuttitaulukko	32 spriteä x 4 tavua = 128 tavua
017F 0180	Ei käytössä	
01FF 0200	Väri- taulukko	32 tavua
021F 0220	Ei käytössä	
03FF 0400	Hahmo- nimitaulukko	24 riviä x 32 merkkiä = 768 tavua
06FF 0700	Ei käytössä	
07FF 0800	Hahmo- generaattori alalohko	256 hahmoa x 8 tavua/hahmo = 2048 tavua
0FFF		Kuva 20-a.

Jos samaan sovellukseen tarvitaan 16x16-bitin sprite-kuvia, muistikartta voidaan asettaa seuraavasti:

0000	Sprite- generaattori- taulukko	32 16x16-hahmoa 32 spriteä x 32 tavua = 1024 tavua
03FF 0400	Hahmo- nimitaulukko	24 riviä x 32 merkkiä = 768 tavua
06FF 0700	Sprite- attribuuttitaulukko	32 spriteä x 4 tavua = 128 tavua
073F 0740	Väri- taulukko	32 tavua
075F 0760	Ei käytössä	
0800	Hahmo- generaattori alalohko	256 hahmoa x 8 tavua/hahmo = 2048 tavua
0FFF		

Kuva 20-b.

4. TMS9928A/9929A NASTAT

TOIMINTA	NASTA	I/O	KUVAUS
XTAL1 XTAL2	40,39	I	10,7 +MHz kidetulo.*
R-Y	38	O	VDP:n väripurskeen kellotaajuus. TMS9928A/9929A-tyypeille tämä on värierolähtö.
GROMCLK	37	O	VDP kellolähtö = XTAL/24 Ei yleensä käytetä.
Y	36	O	Yhdistetty videolähtö, TMS9928A/9929A-tyypeille tämä on Y (musta/valko-luminanssi ja yhdistetty synkr.) ulostulo.

TOIMINTA	NASTA	I/O	KUVAUS
B-Y	35	I/O	Ulkoinen VDP-tulo, TMS9928A/9929A-tyypeille tämä on B-Y värierolähtö.
RESET/SYNC	34	I	RESET-nasta on kolmitasoinen tulonasta. Kun se on alle 0,8 V, RESET alustaa VDP:n. Kun se on yli 9 V, RESET on synkronointitulo ulkoiselle videolle.
V _{cc}	33	I	+5 V:n jännite
RD \emptyset MSB	32	I	Tietoväylä luettaessa VRAM:iin
RD1	31	I	
RD2	30	I	
RD3	29	I	
RD4	28	I	
RD5	27	I	
RD6	26	I	
RD7	25	I	
CD \emptyset MSB	24	I/O	CPU tietoväylä; (CD \emptyset) on eniten merkitsevä bitti.
CD1	23	I/O	
CD2	22	I/O	
CD3	21	I/O	
CD4	20	I/O	
CD5	19	I/O	
CD6	18	I/O	
CD7 LSB	17	I/O	
INT	16	O	CPU-keskeytyksen ulostulo
CSR	15	I	CPU-VDP lukupulssi
CSW	14	I	CPU-VDP tulostuspulssi
MODE	13	I	CPU-liitäntäpiirin valinta; tavallisesti prosessorin osoitelinja.
V _{ss}	12	I	Maa
R/W	11	O	VRAM tulostuspulssi
AD \emptyset MSB	10	O	VRAM osoite/tietoväylä (multipleksoidut ylimmät ja alimmat VRAM-osoi- ja tulostustavut) AD \emptyset on eniten merkitsevä bitti ja käytetään vain datalle eikä osoitteenmuodostukseen **
AD1	9	O	
AD2	8	O	
AD3	7	O	
AD4	6	O	
AD5	5	O	
AD6	4	O	
AD7	3	O	
CAS	2	O	VRAM-sarakeosoitteen näytteytys
RAS	1	O	VRAM-riviosoitteen näytteytys

* Kun ohjataan ulkoisesti, molempia tuloja on ohjattava.

** Vähiten merkitsevä osoitebitti (AD7) on langoitettu dynaamisen RAM:in A \emptyset :aan.
Vastaavasti AD6 on kytketty RAM:in A1:een.

5. SPRITE-DEMO-OHJELMA

Tässä esimerkissä käymme läpi muutaman sprite-kuvion muodostamisen. Ohjelmasta on listaus tämän kappaleen lopussa sivulla 94.

Tämä ohjelma on perusohjelma, josta voit rakentaa oman grafiikkaa ja sprite-kuvia käyttävän peliohjelman. Ohjelmassa käytetään Grafiikka II-tilaa ja se muodostaa neljä 16x16-spriteä. Spritejä voidaan ohjata kuvaruudulla peliohjaimilla.

Ohjelma käsittää kone- ja Basic-kielisen osan. Konekielinen osa alkuasettaa video-prosessorin ja lataa sprite-kuviot videomuistiin sekä alkuasetusarvot. Basic-kielinen osa lukee konekielisen osan ja syöttää sen muistiin. Samalla se muuttaa heksadesimaaliluvut desimaaliseksi. Basic-kielisellä osalla ohjataan myös sprite-kuvien liikkeitä peliohjaimilla. Laukaisunäppäimellä valitaan liikuteltava sprite. Kun laukaisunäppäimiä ei paineta, liikkuu Salora-"ankka", ja kun vasenta laukaisunäppäintä painetaan, liikkuu oikealla ylhäällä oleva sprite. Oikealla laukaisunäppäimellä liikutellaan vasemmalla alhaalla olevaa spriteä ja painettaessa kumpaakin laukaisunäppäintä liikkuu oikealla alhaalla oleva sprite.

Ohjelman perusteellinen tutkiminen voidaan suorittaa sivulla 38 esitetyn Monitor-Assembler-ohjelman avulla.

Annamme seuraavat alkuasetusarvot videoprosessorille (listauksen rivit 1110 - 1120):

Osoite			
70C6	Rekisteri	0	02
	"	1	C2
	"	2	01
	Rekisteri	3	82

Osoite

Rekisteri	4	01
"	5	0E
"	6	00
70CA Rekisteri	7	01

Spraittien alkuasetusarvot:

(listauksen rivit 1125 - 1130)

Osoite			X	Y	Nimi	Väri
70CE	SPRITE	0	40	60	00	04
70D2	SPRITE	1	60	60	04	07
70D6	SPRITE	2	40	80	08	0B
70DA	SPRITE	3	60	80	0C	0F

SPRAITTIEN DATAT:

[illegible]

SPRITE 1 (rivot 1290 - 1310)

70FE	FF	X X X X X X X X	X X X X X X X X	FF	710E
80	X			X	01
87	X		X X X	X X X X	F1
9F	X		X X X X X	X X X X X	F9
98	X		X X	X X	19
80	X			X X	19
80	X			X X	31
83	X		X X	X X X X	F1
83	X		X X	X X X X	F1
80	X			X X X	39
80	X			X X	19
98	X		X X	X X	19
9F	X		X X X X X	X X X X X	F9
8F	X		X X X X	X X X X	F1
80	X			X	01
710D	FF	X X X X X X X X	X X X X X X X X	FF	711D

SPRITE 2 (rivot 1390 - 1410)

711E	AA	X X	X X	X X	X X	AA	712E
55	X X	X	X	X	X	X	55
AA	X X	X X		X X	X X		AA
55	X	X	X	X	X	X	55
AA	X X	X X		X X	X X		AA
55	X	X	X	X	X	X	55
AA	X X	X X		X X	X X		AA
55	X	X	X	X	X	X	55
AA	X X	X X		X X	X X		AA
55	X	X	X	X	X	X	55
AA	X X	X X		X X	X X		AA
55	X	X	X	X	X	X	55
AA	X X	X X		X X	X X		AA
55	X	X	X	X	X	X	55
711D	55	X	X	X	X	X	713D

SPRITE 3 (rivot 1490 - 1510)

713E	AA	X X	X X	X X	X X	AA	714E
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
AA	X X	X X		X X	X X		AA
714D	AA	X X	X X	X X	X X	AA	715D

```

10 GOTO 780
50 REM LATAUSOHJELMA KONEKIELISELLE
60 REM KOODILLE JA HEKSA/DESIMAALI-
70 REM MUUTOS.
90 I = 28672
100 READ A$
110 IF A$ = "END" THEN RETURN
120 H$ = LEFT$(A$,1):Q$ = RIGHT$(A$,1)
130 H = ASC (H$):Q = ASC (Q$)
140 IF H > = 65 THEN H = H - 55: GOTO 160
150 H = H - 48
160 IF Q > = 65 THEN Q = Q - 55: GOTO 180
170 Q = Q - 48
180 A = H * 16 + Q
200 POKE I,A
210 I = I + 1
220 GOTO 100
230 REM ISOJEN TAI PIENIEN SPRAITTIEN
240 REM VALINTA
250 INPUT "ISOJA K/E":A$
260 IF A$ = "K" THEN POKE 28871,195
270 RETURN
300 CALL 28672
310 REM PELIOHJAJIMMEN SUUNAN VALINTA
350 IF SU = 4 OR SU = 132 THEN Y1 = PEEK (OSOIT): POKE OSOIT,Y1 + 1
360 IF SU = 32 OR SU = 160 THEN Y2 = PEEK (OSOIT): POKE OSOIT,Y2 - 1
370 IF SU = 8 OR SU = 136 THEN X1 = PEEK (OSOIT - 1): POKE OSOIT - 1,X1 - 1
380 IF SU = 2 OR SU = 130 THEN X2 = PEEK (OSOIT - 1): POKE OSOIT - 1,X2 + 1
390 CALL 28706
400 GOTO 805
780 GOSUB 90
790 I = 28831: GOSUB 100
793 I = 28870: GOSUB 100
795 GOSUB 250
800 CALL 28672
803 REM PELIOHJAJIMEN LUKURUTIINI
805 CALL - 1536
810 LIP = PEEK (21)
815 SU = PEEK (24)
818 REM SPRAITIN VALINTA
820 IF LIP = 17 THEN OSOIT = 28883: GOTO 350
830 IF LIP = 34 THEN OSOIT = 28887: GOTO 350
840 IF LIP = 51 THEN OSOIT = 28891: GOTO 350
860 OSOIT = 28879: GOTO 350
900 REM KONEKIELINENOHJELMA
1000 DATA A0,87,A2,07,BD,C6,70,20,9F,70,88,CA,D0,F6,A0,"40"
1010 DATA A9,00,20,9F,70,A2,C0,A0,00,8D,00,30,C8,D0,FA,"E8"
1020 DATA D0,F5,A0,47,A9,00,20,9F,70,A2,00,BD,CE,70,8D,"00"
1030 DATA 30,E8,8A,C9,10,D0,F4,A0,40,A9,00,20,9F,70,A2,"00"
1040 DATA BD,DE,70,8D,00,30,E8,8A,C9,80,D0,F4,60,"END"
1050 REM VIDEOMUISTIIN KIRJOITUS
1100 DATA 8D,01,30,8C,01,30,60,"END"
1110 REM VIDEOPROSESSORIN ALKUARVOT
1120 DATA 02,C2,01,82,01,0E,00,"01"
1125 REM SPRAITTIEN ALKUARVOT
1130 DATA 40,60,00,04,60,60,04,07,40,80,08,0B,60,80,0C,"0F"
1190 REM SPRAITTI 0 DATAT
1200 DATA 00,00,0F,1F,3F,3C,38,3C,1F,0F,FF,7F,3F,1F,0F,"07"
1210 DATA 00,20,F0,F8,FC,00,00,00,FC,FE,FF,FF,FF,FE,"F8"
1290 REM SPRAITTI 1 DATAT
1300 DATA FF,80,87,9F,98,80,80,83,83,80,80,98,9F,8F,80,"FF"
1310 DATA FF,01,F1,F9,19,19,31,F1,F1,39,19,19,F9,F1,01,"FF"
1390 REM SPRAITTI 2 DATAT
1400 DATA AA,55,AA,55,AA,55,AA,55,AA,55,AA,55,AA,55,AA,"55"
1410 DATA AA,55,AA,55,AA,55,AA,55,AA,55,AA,55,AA,55,AA,"55"
1490 REM SPRAITTI 3 DATAT
1500 DATA AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,"AA"
1510 DATA AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,"AA"
1600 DATA END

```

LUKU 6. ÄÄNIGENERAATTORI SN76489AN

- 1) Äänigeneraattorit
- 2) Kohinageneraattori
- 3) Äänisummeri/lähtöpuskuri
- 4) CPU/SN76489AN-liitäntäpiiri
- 5) Ohjausrekisterit
- 6) Tiedon muodot
- 7) Tiedon siirto
- 8) Piirin nastajärjestys

KUVAUS

Digitaalinen SN76489AN äänigeneraattori on toteutettu I²L/bipolaarisella IC-tekniikalla ja tarjoaa edullisen mahdollisuuden tuottaa ääniä mikroprosessorisysteemissä. SN76489AN on tietoväyläsovitteinen I/O-oheislaite.

OMINAISUUKSIA

- 3 ohjelmoitavaa äänigeneraattoria
- Ohjelmoitava generaattori valkoiselle kohinalle
- Ohjelmoitava vaimennus
- Samanaikaiset äänet
- TTL-sovitteinen
- Kellotaajuuudet 4 MHz:iin saakka
- Äänisummeri

TOIMINTA

1. Äänigeneraattorit

Jokainen äänigeneraattori muodostuu taajuussynteesiosasta ja vaimennusosasta. Taajuussynteesiosa vaatii 10-bittistä tietoa (F₀ - F₉) määrittelemään halutun taajuuden puoli-jakson (n). F₀ on eniten merkitsevä bitti ja F₉ vähiten merkitsevä bitti. Tämä tieto on tallennettu 10-tilaiseen äänilaskuriin jonka arvoa pienennetään nopeudella N/16. N on tulon kellotaajuus. Kun äänilaskurin arvoksi on saatu nolla, tuotetaan lainaussignaali. Tämä lainaussignaali muuttaa jatkuvasti taajuuskiikun tilaa ja lataa uudelleen äänilaskurin. Siten halutun taajuuden jakso on jaksorekisterin arvo kaksinkertaisena.

Taajuus voidaan laskea kaavalla:

$$f = N/32n$$

missä N = referenssitaajuus (Hz)
n = 10-bittinen binääriluku.

Taajuuskiikun lähtö syöttää nelitilaista vaimenninta. Vaimennusarvot sekä niiden bittipaikat sanassa on esitetty taulukossa 1. Moninkertaisen vaimennuksen ohjausbitit voivat olla tosia samanaikaisesti. Silloin on maksimivaimennus 28 dB.

BITTIPAIKAT				
A ₀	A ₁	A ₂	A ₃	PAINO
0	0	0	1	2 dB
0	0	1	0	4 dB
0	1	0	0	8 dB
1	0	0	0	16 dB
1	1	1	1	OFF

Taulukko 1. Vaimennuksen ohjaus

2. Kohinageneraattori

Kohinageneraattori muodostuu kohinalähteestä ja vaimentimesta. Kohinalähde on siirtorekisteri jossa on exclusive-OR takaisinkytkentäpiiri. Takaisinkytkentäpiirin tarkoituksena on suojella siirtorekisteriä lukkiutumasta nollatilaan (FB = Feed Back, takaisinkytkentä).

FB	TOIMINTA
0	"Jaksottainen" kohina
1	"Valkoinen" kohina

Taulukko 2. Kohinan takaisinkytkentäohjaus

Aina kun kohinanohjausrekisteriä muutetaan, tyhjennetään siirtorekisteri. Siirtorekisteri saa yhden neljästä nopeudesta, jotka määrätään kahdella NF-bitillä. Kiinteät siirtonepeudet määrää kello-tulo.

BITIT		
NF0	NF1	SIIRTONOPEUS
0	0	N/512
0	1	N/1024
1	0	N/2048
1	1	Äänigeneraattorin 43 lähtö

Taulukko 3. Kohinageneraattorin taajuuden ohjaus

Kohinalähteen ulostulo on yhdistetty ohjelmoitavaan vaimentimeen.

3. Äänisummeri/lähtöpuskuri

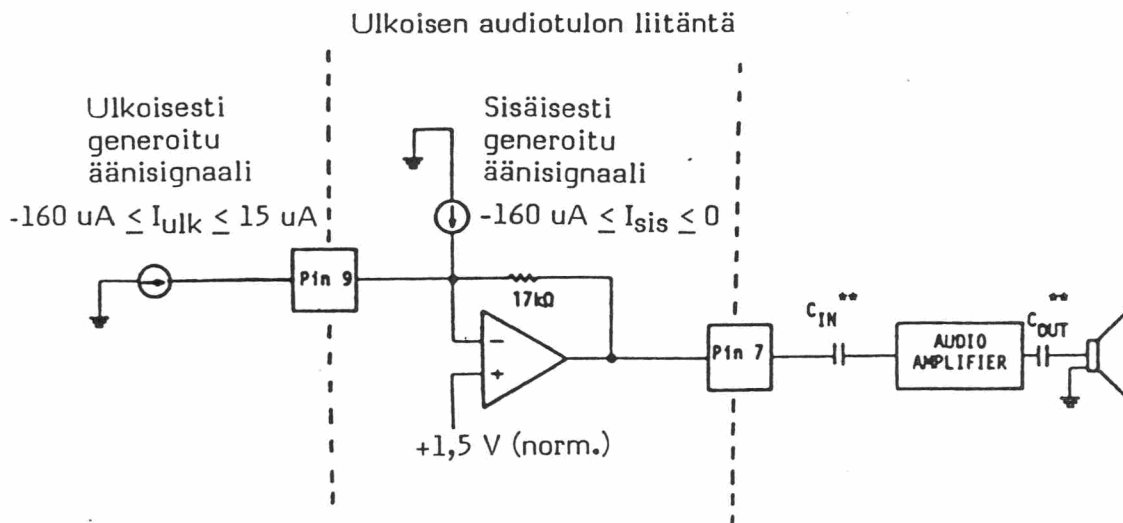
Summeri on tavanomainen operaatiovahvistimella toteutettu summauspiiri. Se yhdistää kolme äänigeneraattorin ulostuloa, kohinageneraattorin ulostulon ja ulkoisen äänilähteen, kuva 1.

4. CPU/SN76489AN-liitäntäpiiri

Mikroprosessori voidaan liittää SN76489AN:n kanssa yhteen kahdeksalla tietoväylällä ja kolmella ohjauslinjalla (WE, CE ja READY). Jokainen äänigeneraattori tarvitsee 10-bittisen syötön taajuuden valitsemiseen ja 4-bittisen syötön vaimennuksen valintaan. Taajuuden päivitykseen tarvitaan kaksi tavunsiirtoa kun taas vaimennuksen päivitykseen tarvitaan vain yksi tavunsiirto.

Jos piirillä ei ole muita ohjausrekistereitä, äänigeneraattori on päivitettävissä nopeasti lähettämällä aluksi molemmat taajuus- ja rekisteritiedon tavut ja sen jälkeen ainoastaan tietojen toinen tavu seuraaville arvoille. Jos rekisteriosoite on lukittu piiriin, niin tieto

kulkee samaan rekisteriin. Tämä sallii kuuden eniten merkitsevän bitin muuntamisen nopeasti taajuuspyyhkäisyä varten.



Kuva 1.

5. Ohjausrekisterit

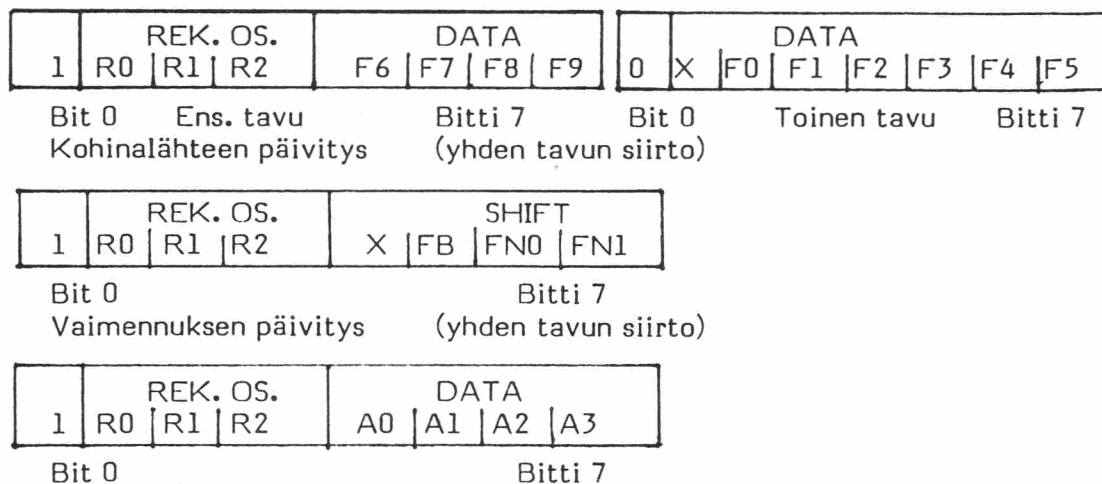
SN76489AN:ssä on kahdeksan ulkoista rekisteriä, joita käytetään ohjaamaan kolmea äänigeneraattoria ja kohinalähdettä. Kaikessa SN76489AN:ään tapahtuvassa tiedonsiirrossa sisältää ensimmäinen tavu kolmen bitin kentän, joka määrittää ohjausrekisterin määräpaikan. Rekisteriosoitteen koodit ovat taulukossa 4.

R0	R1	R2	Määräpaikan ohjausrekisteri
0	0	0	Äänen 1 taajuus
0	0	1	Äänen 1 vaimennus
0	1	0	Äänen 2 taajuus
0	1	1	Äänen 2 vaimennus
1	0	0	Äänen 3 taajuus
1	0	1	Äänen 3 vaimennus
1	1	0	Kohinan ohjaus
1	1	1	Kohinan vaimennus

Taulukko 4. Rekisteriosoittekenttä

6. Tiedon muodot

Tiedon siirtoon vaadittavat datamuodot ovat seuraavat:



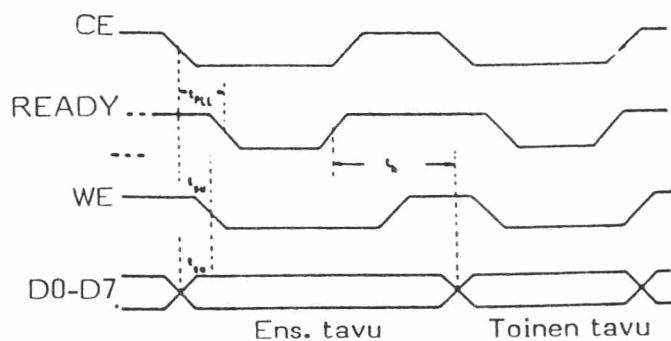
Kuva 2.

7. Tiedon siirto

Mikroprosessori valitsee äänigeneraattorin SN76489AN siirtämällä CE-signaalin alas. Jollei CE ole alhaalla, ei tietoa voida lukea. Kun CE on alhaalla, WE-signaali poimii tietoväylän sisällön sopivaan ohjausrekisteriin. Tietoväylän sisällön täytyy olla tänä aikana vakaa.

SN76489AN vaatii noin 32 kellojaksoa tiedon lataamiseen ohjausrekisteriin. Avoin-kollektorilähtöä READY käytetään tahdistamaan mikroprosessori tähän siirtoon ja se vedetään alas välittömästi CE-signaalin etureunan jälkeen. Se vapautetaan menemään ylös (ulkoisen ylösveto), kun tiedonsiirto on suoritettu.

Tiedonsiirron ajoitus on esitetty kuvassa 3.



Kuva 3. Tiedonsiirron ajoitus

SYÖTTÖ		TULOSTUS
CE	WE	READY
L	L	L
L	H	L
H	L	H
H	H	H

* Tämä taulukko pitää paikkansa, kun laite
(1) ei ole kello-ohjattu ja
(2) on alustettu vetämällä WE ja CE ylös.

Taulukko 5. Toimintataulukko *

8. Nastajärjestys

Allaolevassa listassa esitetään SN76489AN:n nastojen nimet ja kuvataan niiden toiminnot.

TOIMINTA NASTA I/O KUVAUS

CE	6	I	Piirin aktivointi – kun piiri on aktiivinen (alhaalla), voidaan tietoa siirtää CPU:sta SN76489AN:ään.
D0 MSB	3	I	Tietoväylät, joiden kautta ohjaustieto syötetään.
D1	2	I	
D2	1	I	
D3	15	I	
D4	13	I	
D5	12	I	
D6	11	I	
D7	10	I	
V _{CC}	16	I	+5 V:n jännite
GND	8	O	Maa
CLOCK	14	I	Kellotulo
WE	5	I	Tulostuksen aktivointi – aktivoituna (alhaalla) WE ilmaisee, että tietoa saadaan CPU:sta SN76489AN:ään.
READY	4	O	Aktivoituna (ylhäällä) READY ilmaisee, että tieto on luettu. Kun READY on alhaalla, mikroprosessori odottaa kunnes se on ylhäällä.
AIN	9	I	Audiosignaali sisään.
AOUT	7	O	Äänen ulostulo.

LUKU 7. LEVYKEASEMA

1) Toiminta

2) Ennalta ehkäisevä huolto

3) EXEC-komento

1. TOIMINTA

Yleistä

FD100A-levykeasema käsittää luku/kirjoitus-, ohjaus- ja moottorielektroniikan sekä vetomekanismin, luku/kirjoituspään ja raidan kohdistusmekanismin. Nämä osat suorittavat seuraavat toiminnot:

- a) Ottavat vastaan ja tuottavat ohjaussignaaleita.
- b) Sijoittavat luku/kirjoituspään oikealle raidalle.
- c) Lukevat ja kirjoittavat tietoa.
- d) Ohjaavat vetomoottorin nopeutta.

1.1. Luku/kirjoitus ja ohjauselektroniikka

Levykeaseman piirilevyt sisältävät

- a) askelmoottoriohjauksen,
- b) kirjoitusvahvistimen,
- c) lukuvahvistimen ja ohjauspiirit,
- d) tiedostosuojauksen ilmaisimen,
- e) ohjauksen sallintapiirin,
- f) raidan 00 ilmaisimen ja
- g) vetomoottorin ohjauspiirin.

1.2. Vetomekanismi

Vetomoottori pyörittää hihnavedolla levy-yksikköä nopeudella 300 kierrosta/min. Moottorin nopeutta ohjaa servopiiri. Keskiölukitus, joka liikkuu oven lukitusmekanismin yhteydessä, keskittää ja lukitsee levykkeen levykeaseman keskukseen.

1.3. Luku/kirjoituspään asetusmekanismi

Luku/kirjoituspää sijoitetaan halutulle raidalle lähettämällä ohjaussignaaleja askelmoottorille. Luku/kirjoituspään kuljetin ja askelmoottori on kytketty toisiinsa teräsnauhalla. Askelmoottori pyörii kaksi askelta raitaa kohti.

1.4. Luku/kirjoituspää

FD100A-aseman luku/kirjoituspäätä käytetään luettaessa tietoa levyltä tai kirjoitettaessa sille. Luku/kirjoituspää on kiinni kuljettimessa, joka liikkuu kiskoilla ja jota askelmoottori siirtää. Levyä pidetään tasossa, joka on kohtisuorassa luku/kirjoituspäätä vastaan.

1.5. Raidan 00 ilmaismekanismi

Virran kytkemisen ja raidan paikallistamisvirheen jälkeen on luku/kirjoituspään sijainti määrittelämätön. Jotta luku/kirjoituspään sopiva paikka voitaisiin varmistaa, mekanismin ilmaisin lukitsee pään raidalle 00.

1.6. Aseman valinta

Aktivomalla BENBL-linja valitaan levyasema. Valinnan jälkeen vetomoottori käynnistyy ja levyaseman etupaneelin LED-merkkivalo syttyy.

1.7. Tiedoston suojausmekanismi

Tämä mekanismi muodostuu valodiodista ja fototransistorista, joka tunnistaa levyvaipan kirjoitukselle aktivoidun uran. Kun tällainen levy otetaan käyttöön ja jos tunnistamisessa sattuu häiriö, ei luku/kirjoituspään läpi kulje virtaa joka aktivoisi poisto- tai kirjoitus-toiminnan. Levyllä oleva tieto on suojattu kirjoituskomennon virheellistä syöttöä vastaan.

2. Ennaltaehkäisevä huolto

2.1. Magneettipään puhdistaminen puhdistuslevyllä

Kun levyasemaa käytetään pölyisessä ympäristössä, suositellaan että magneettipäiden pinta puhdistetaan säännöllisesti kaupanolevilla puhdistuslevyllä.

A. Työvälineet ja laitteet

- a) Puhdistuslevy
- b) Manager-tietokone

B. Puhdistusmenetelmä

- a) Aseta puhdistuslevy puhdistettavaan asemaan ja sulje ovi.
- b) Puhdista luku/kirjoituspäätä noin 10 ... 30 sekuntia.

2.2. Suora puhdistus

Huom: Tämä menetelmä edellyttää levykeaseman avaamista, joten suosittelemme mieluummin puhdistuslevyn käyttöä.

A. Työvälineet

- a) Absoluuttinen alkoholi (etanoli)
- b) Vanupuikko

B. Puhdistusmenetelmä

- a) Kostuta varovasti vanupuikko alkoholiin.
- b) Nosta ylös lukupään pehmeä alusta ja puhdista magneettipään pinta varovasti vanupuikolla.
Huom. Älä koske alustan pintaan.
- c) Pyyhi magneettipään pinta puhtaalla ja kuivalla kankaalla alkoholin haihduttua pois.
- d) Kun olet varmistunut, että magneettipäät ovat puhtaat eikä niiden pintaan ole jäänyt nukkaa, paina alusta jälleen varovasti alas.

3. EXEC-komento

Manager-DOS tuntee myös käskyn EXEC, jota DOS-käsikirjassa ei ole mainittu.

MITÄ SE TEKEE?

Se ottaa tietokoneen hallintaansa, kunnes kaikki ohjelmarivit on suoritettu. Käyttäjä ei voi tehdä paljoakaan, kun käsky on annettu tietokoneelle. Se on tavallaan Super-RUN-komento, jota ei voi keskeyttää.

Jos EXEC-komento avaa tiedoston, ei CLOSE sulje sitä. EXEC jatkaa suoritustaan loppuun asti, jolloin se itse sulkee tiedoston.

MITEN KIRJOITAT SEN?

EXEC TIEDOSTONIMI, R\$XXXX

tai

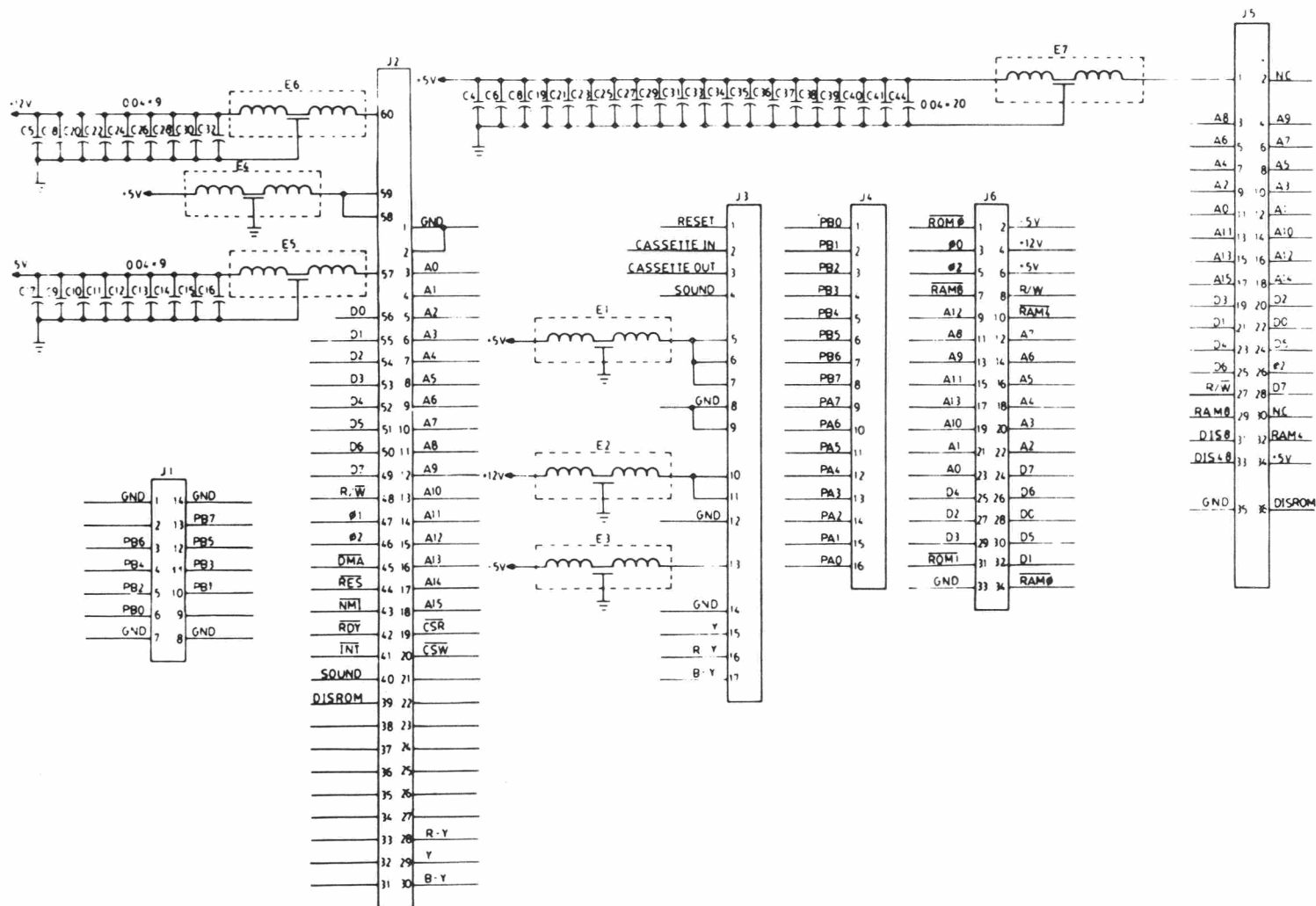
10 DISK "EXEC TIEDOSTONIMI, R\$XXXX"

jossa R\$XXXX-lausekkeessa R:ää seuraa tiedoston rivinumero, jonka jälkeen haluat EXEC-käskyn toimivan.

LUKU 8. LIITTIMET, KYTKENNÄT JA LISÄLAITTEET

- 1) Keskusyksikön liittimet
- 2) Managerin kytkentä
- 3) Datakasettinauhuri
- 4) Levykeasema

1. KESKUSYKSIKÖN LIITTIMET



Kuva 1. Liittimet

a) Lisälaiteväylä J2 (tietokoneen takana)

Nasta n:o	Nastan nimi	Toiminta
1	GND	Järjestelmän maa
2	GND	"
3	A0	Osoiteväylä, vähiten merkitsevä bitti
4	A1	"
5	A2	"
6	A3	"
7	A4	"
8	A5	"
9	A6	"
10	A7	"

Lisälaiteväylä (J2), jatkoa

Nasta n:o	Nastan nimi	Toiminta
11	A8	"
12	A9	"
13	A10	"
14	A11	"
15	A12	"
16	A13	"
17	A14	"
18	A15	Osoiteväylä, eniten merkitsevä bitti
19	\overline{CSR}	Videoprosessorin lukemisen valinta
20	\overline{CSW}	Videoprosessorin kirjoituksen valinta
21-27	N.C.	Ei kytketty
28	R-Y	Videosignaali
29	Y	"
30	B-Y	"
31-38	N.C.	Ei kytketty
39	DISROM	On Board ROM Disable
40	SOUND	Äänisignaali
41	\overline{INT}	CPU:n keskeytys
42	\overline{RDY}	CPU valmis -signaali
43	\overline{NMI}	Non-Maskable Interrupt (maskaamaton keskeytys)
44	\overline{RES}	Järjestelmän nollaus
45	\overline{DMA}	Suora muistin osoitus, yleensä ylhäällä
46	$\emptyset 2$	Järjestelmän kello, vaihe 2
47	$\emptyset 1$	" 1
48	R/ \overline{W}	Luku/kirjoitus
49	D7	Dataväylä, eniten merkitsevä bitti
50	D6	"
51	D5	"
52	D4	"
53	D3	"
54	D2	"
55	D1	"
56	D0	"
57	-5V	Käyttöjännite, -5V
58	+5V	" +5V
59	+5V	" +5V
60	+12V	" +12V

b) Pelimoduuli- ja lisämuistiliitin J5 (tietokoneen oikeassa sivussa)

Nasta n:o	Nastan nimi	Toiminta
1	+5V	Käyttöjännite, +5V
2	N.C.	Ei kytketty
3	A8	Osoiteväylä
4	A9	"
5	A6	"
6	A7	"
7	A4	"
8	A5	"
9	A2	"
10	A3	"
11	A0	"
12	A1	"
13	A11	"
14	A10	"
15	A13	"
16	A12	"
17	A15	"
18	A14	"
19	D3	Dataväylä
20	D2	"
21	D1	"
22	D0	"
23	D4	"
24	D5	"
25	D6	"
26	Ø2	Järj. kello, vaihe 2
27	R/ \bar{W}	Luku/kirjoitus
28	D7	Dataväylä
29	RAM8	RAM-muistin laajennusmahd. (\$4000-\$BFFF)
30	N.C.	Ei kytketty
31	DIS8	Disable RAM8
32	RAM4	RAM-muistinlaajennusmahd. (\$4000-\$7FFF)
33	DIS48	Disable RAM4, RAM8
34	+5V	Käyttöjännite +5V
35	GND	Järjestelmän maa
36	DISROM	Disable On Board ROM

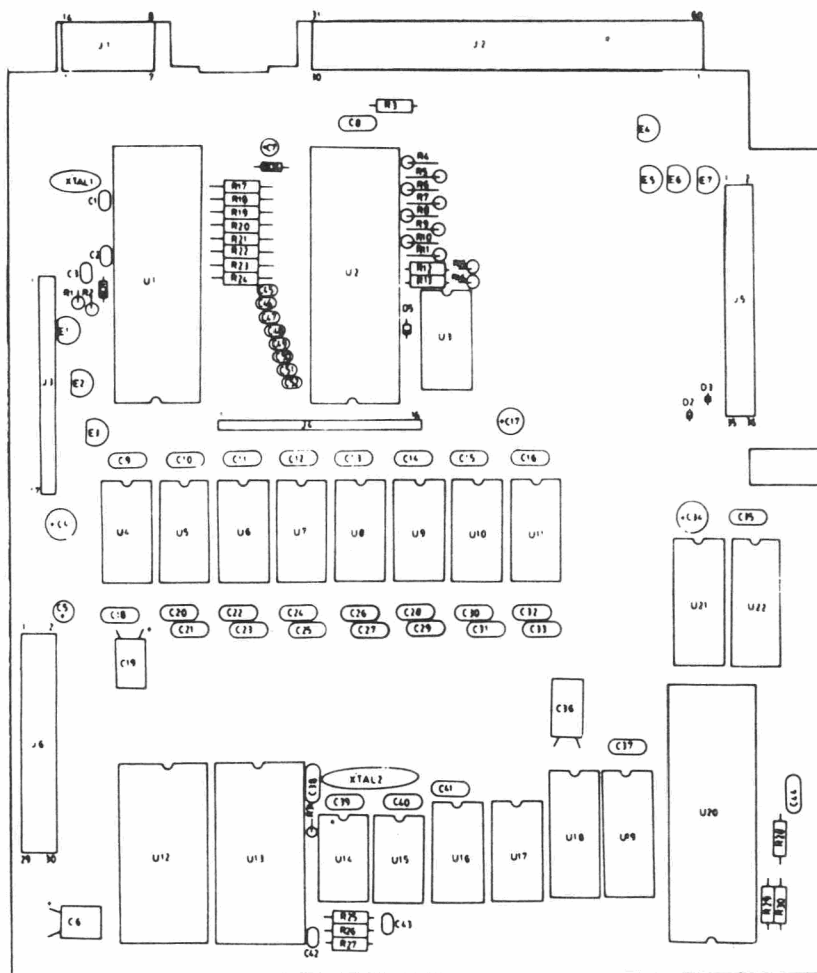
c) Nauhuri- ja monitoriliitännät (piirilevyn liitin J3, ei ulkoista liitännää)

Nasta n:o	Nastan nimi	Toiminta
1	RESET	Järjestelmän nollaus
2	CASSETTE IN	Datan lähetys kasettinauhurille
3	CASSETTE OUT	Datan luku kasettinauhurilta
4	SOUND	Audiosignaali äänigeneraattorilta
5	+5V	Käyttöjännite +5V
6	+5V	" +5V
7	+5V	" +5V
8	GND	Järjestelmän maa
9	GND	"
10	+12V	Käyttöjännite +12V
11	+12V	"
12	GND	Järjestelmän maa
13	-5V	Käyttöjännite -5V
14	GND	Järjestelmän maa
15	Y	Videosignaali VDP:ltä
16	R-Y	"
17	B-Y	"

d) Kirjoitinliitäntä J1 (tietokoneen takana)

Nasta n:o	Nastan nimi	Toiminta
1	GND	Järjestelmän maa
2	\overline{ACK}	Kirjoittimen Acknowledge
3	PB6	Kirjoittimen data 6
4	PB4	" 4
5	PB2	" 2
6	PB0	" 0
7	GND	Järjestelmän maa
8	GND	"
9	\overline{STROBE}	STROBE-signaali
10	PB1	Kirjoittimen data 1
11	PB3	" 3
12	PB5	" 5
13	PB7	" 7
14	GND	Järjestelmän maa

Liittimet J4 (peruslevy - näppäimistö) ja J6 (peruslevy - RAM & ROM-levy) ovat tietokoneen sisällä olevia piirilevyjen välisiä liitännöitä, joita ei ole kytketty ulkoisiin liittämiin.



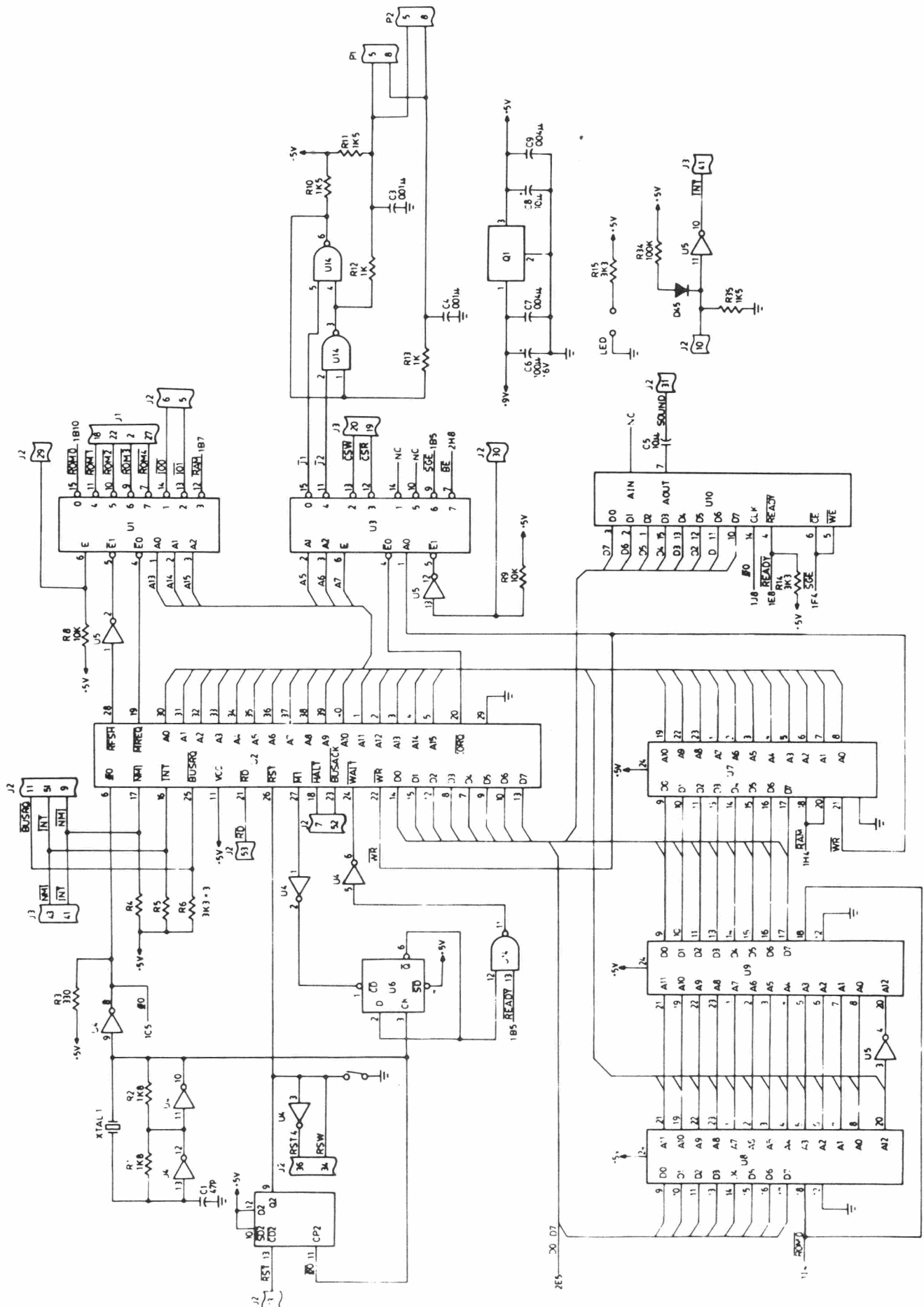
Kuva 2. Osien sijainti peruslevyllä

2. MANAGERIN KYTKENTÄ

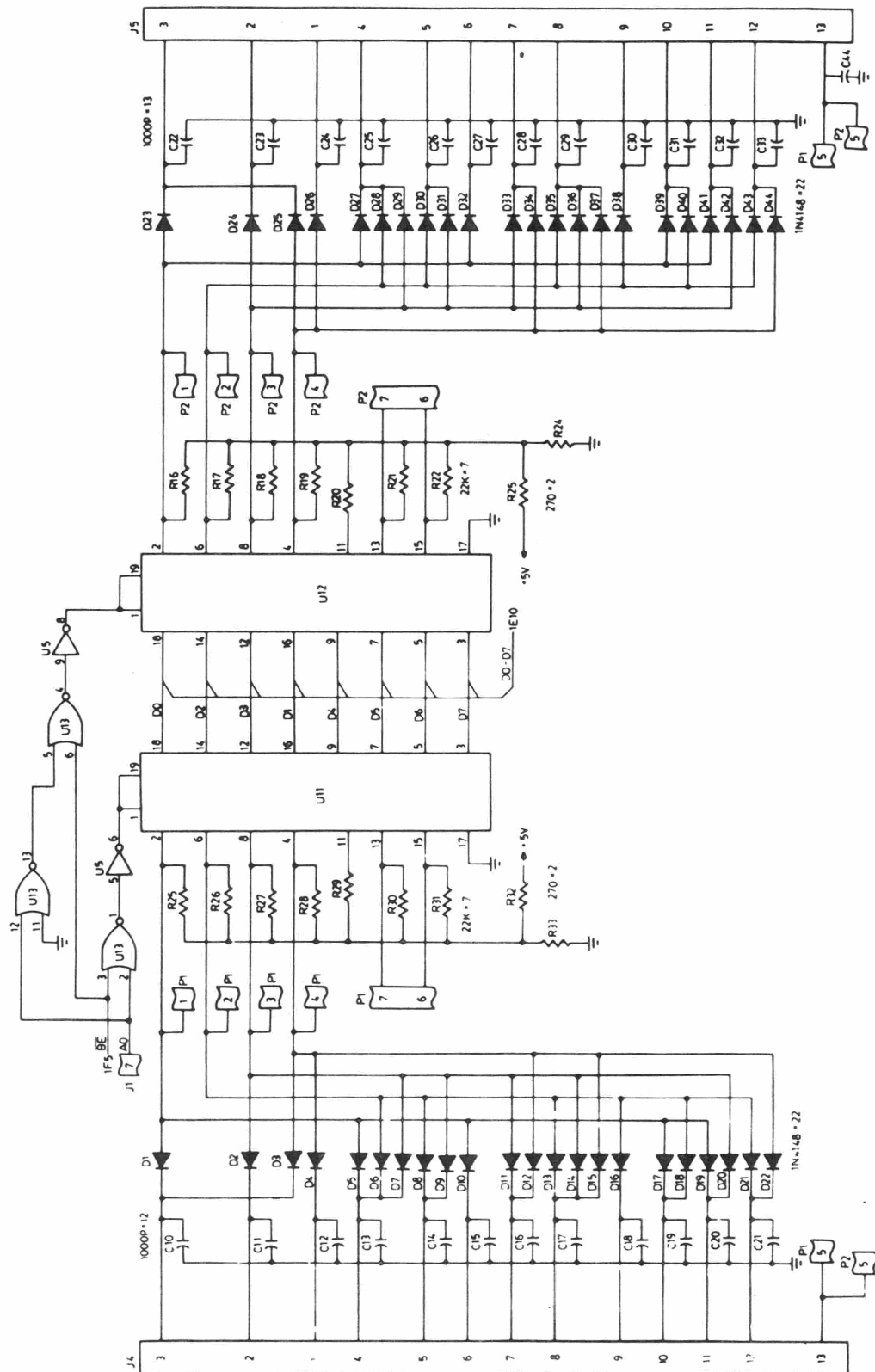
Perusyksikön kytkentäkaaviot on esitetty kuvissa 3 ... 8 sivuilla 111 - 116. ROM & RAM-levyn osasijoittelukuva on sivulla 117, datakasettinauhuri sivuilla 118 - 119 ja levykeaseman kuvat löytyvät sivulta 120 alkaen.



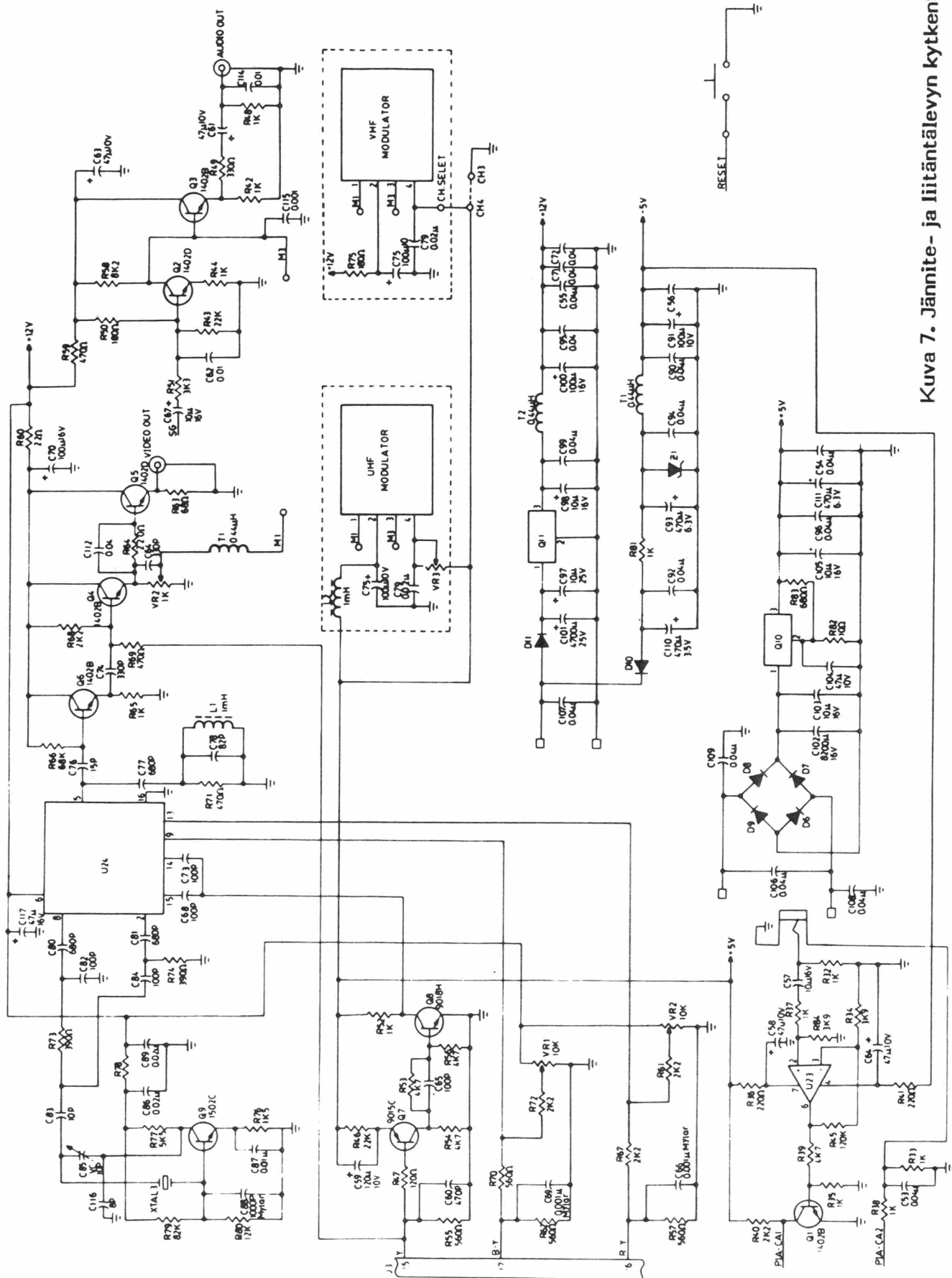
Kuva 3. Peruslevyn osa 1. Lisälaiteväylän kytkentä, CPU (U20) osoituskureineen (U18, U21, U22) sekä videoprosessori (U1), video-RAM-matriisi (U4 ... U11) ja ROM-piirit (U13, U14).



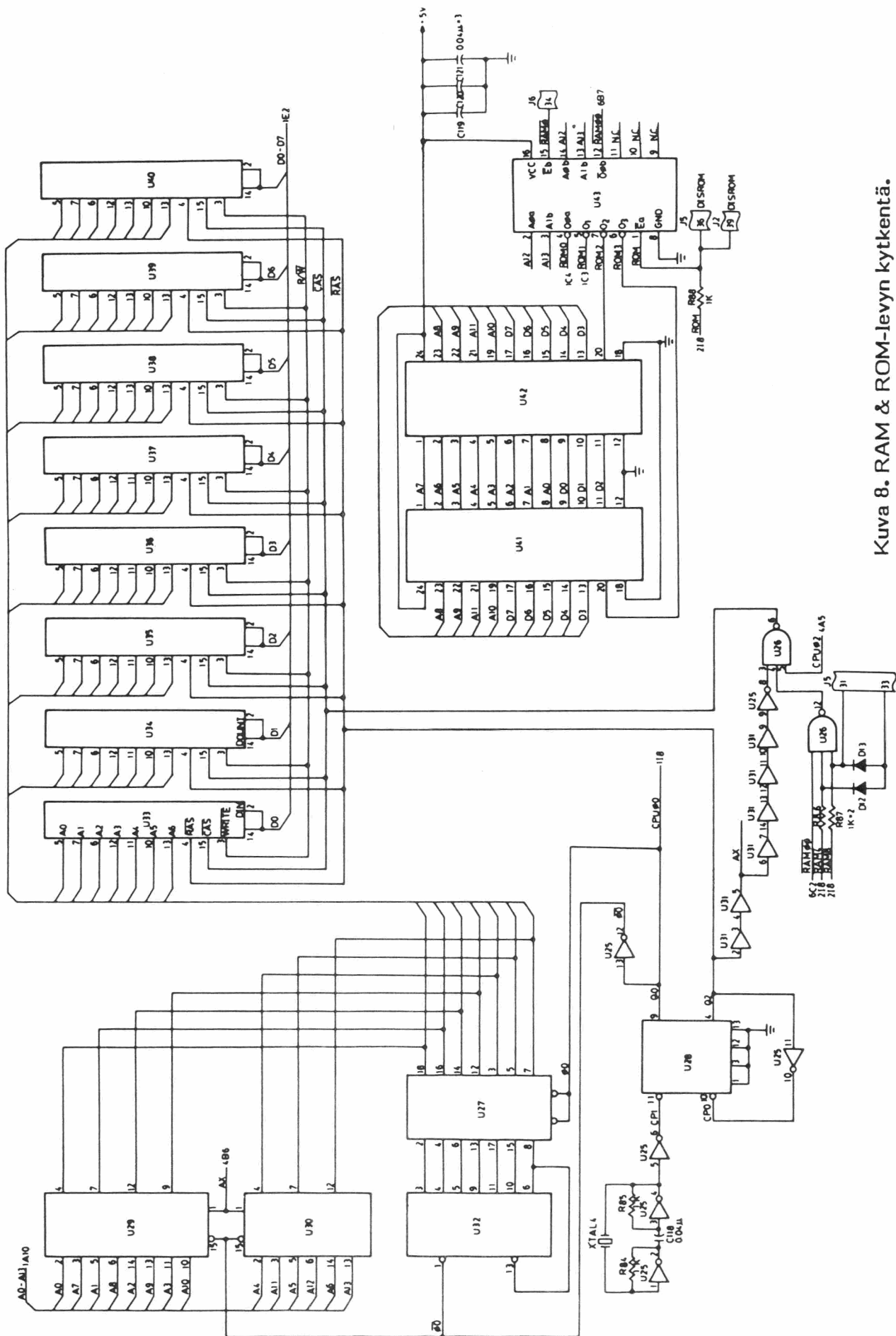
Kuva 4. Peruslevyn osa 2. PIA-piirin (U2) liitanta järjestelmään.



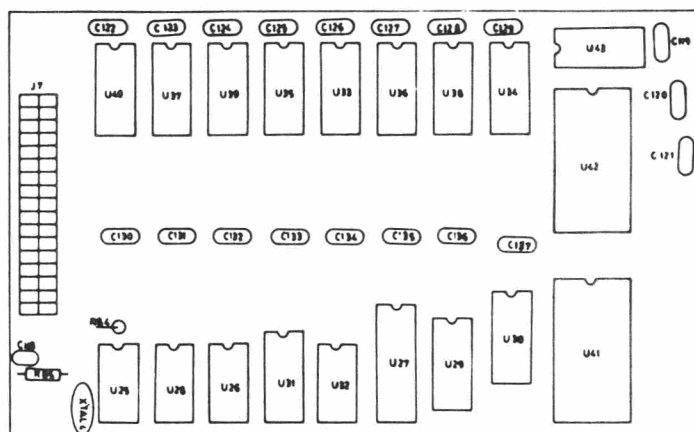
Kuva 5. Peruslevyn osa 3. Pelimoduuli- ja lisämuistimoduuliliittimen (J5) kytkentä.



Kuva 7. Jännite- ja liitäntälevyn kytkentä.

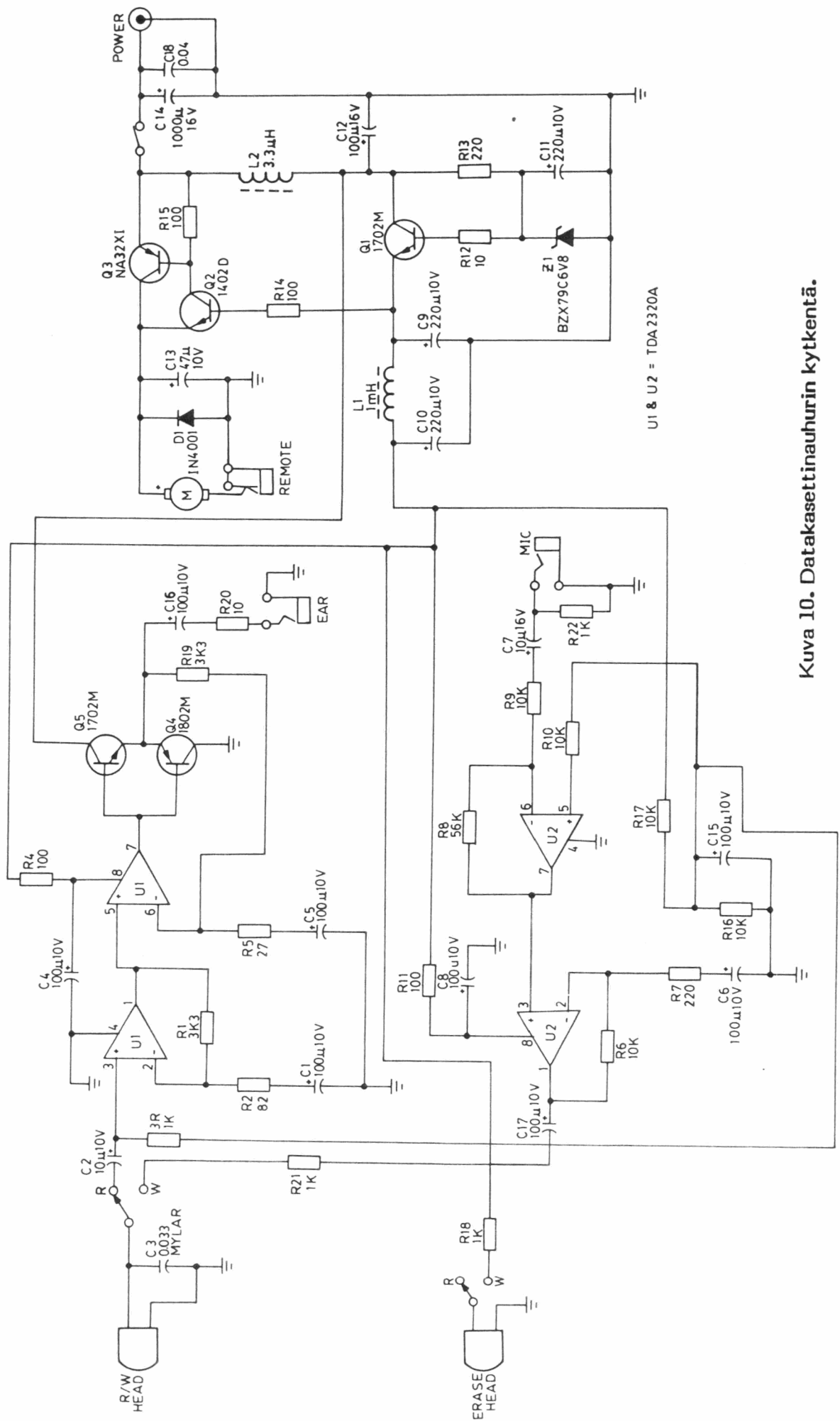


Kuva 8. RAM & ROM-levyn kytkentä.

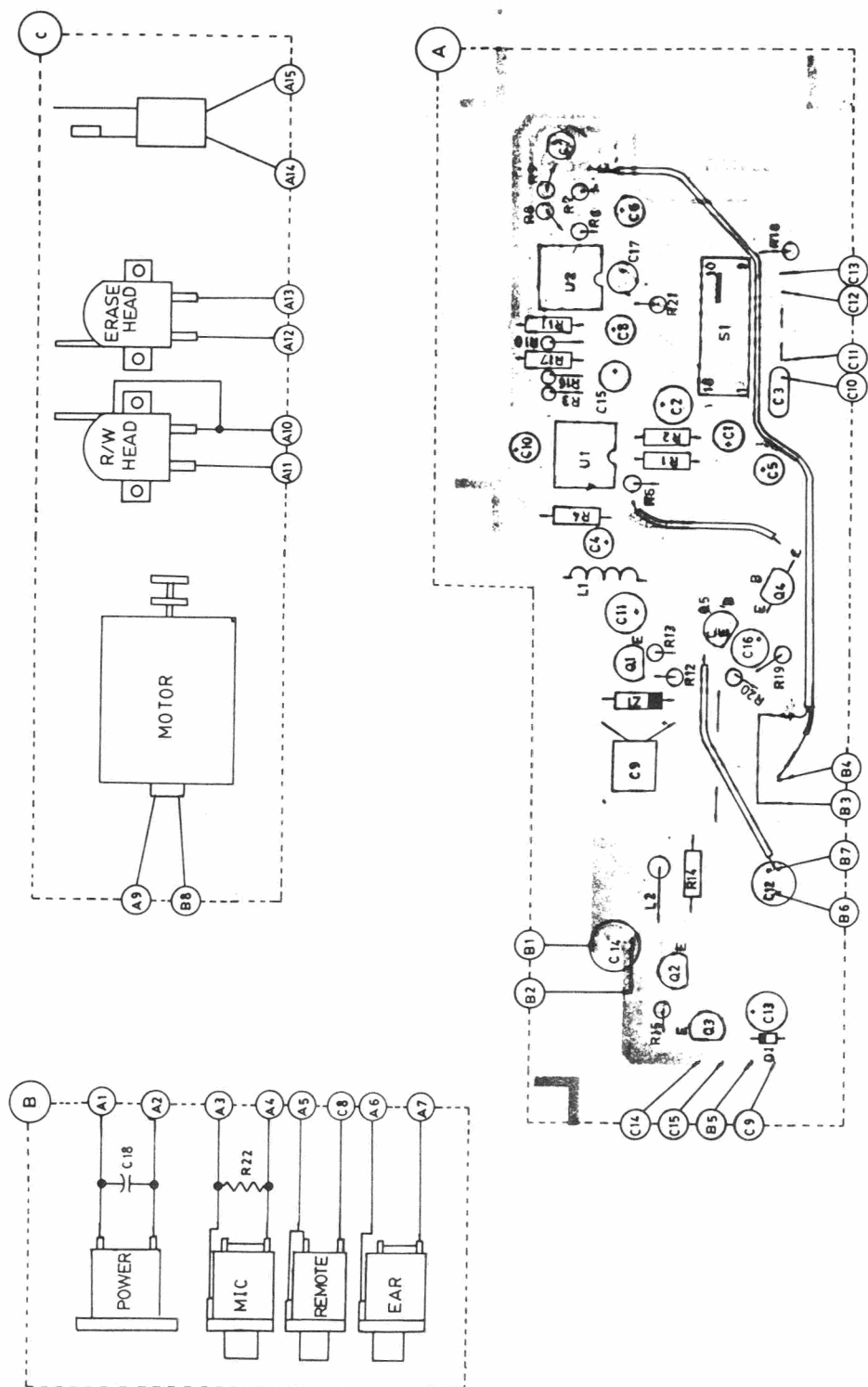


Kuva 9. RAM & ROM-levyn osasijoittelu.

3. DATAKASSETTINAUHURI

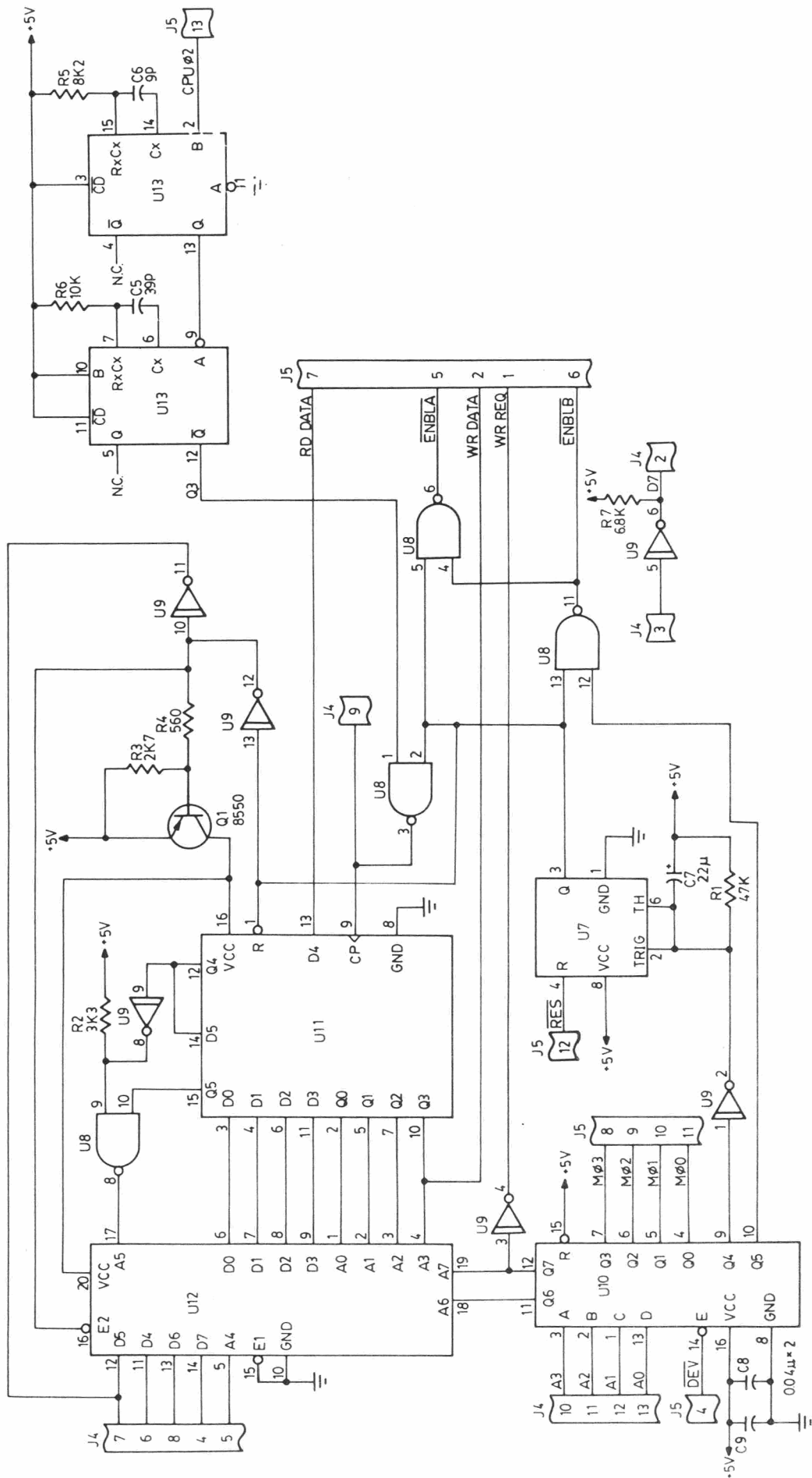


Kuva 10. Datakasettinauhurin kytkentä.

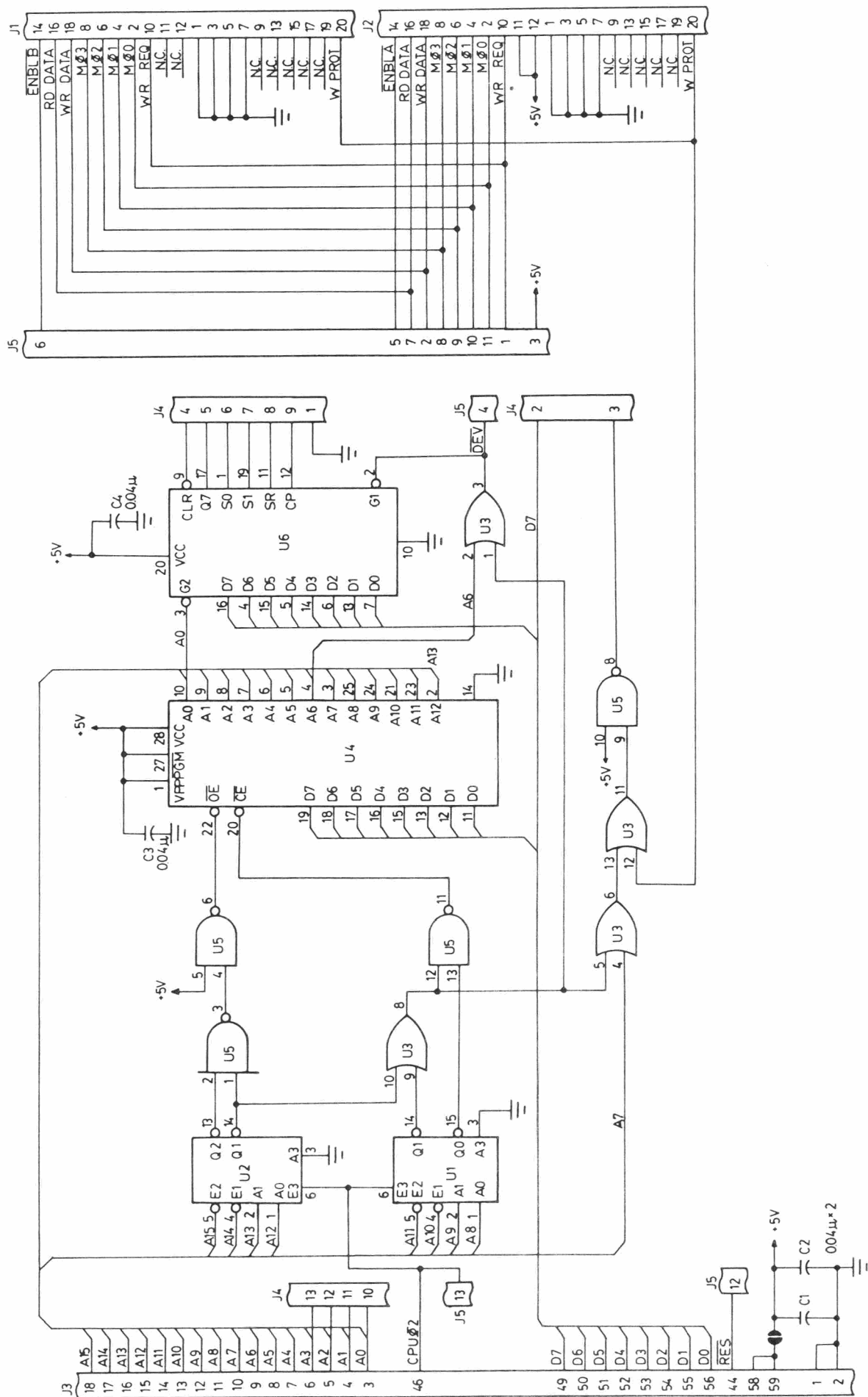


Kuva 11. Nauhurin osasijoittelu.

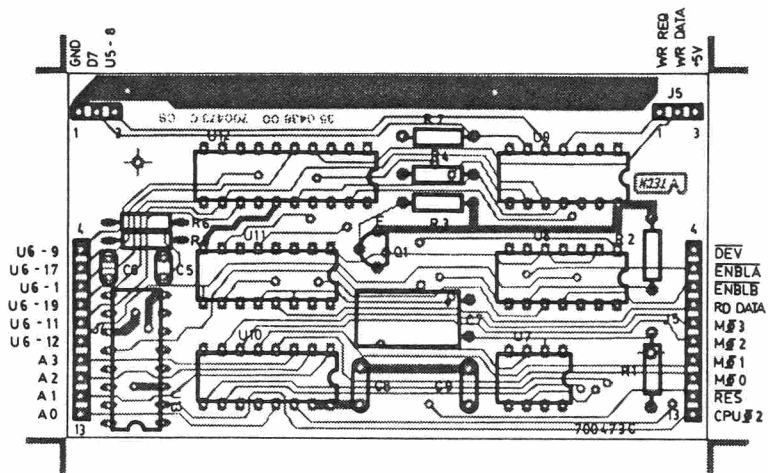
4. LEVYKEASEMA



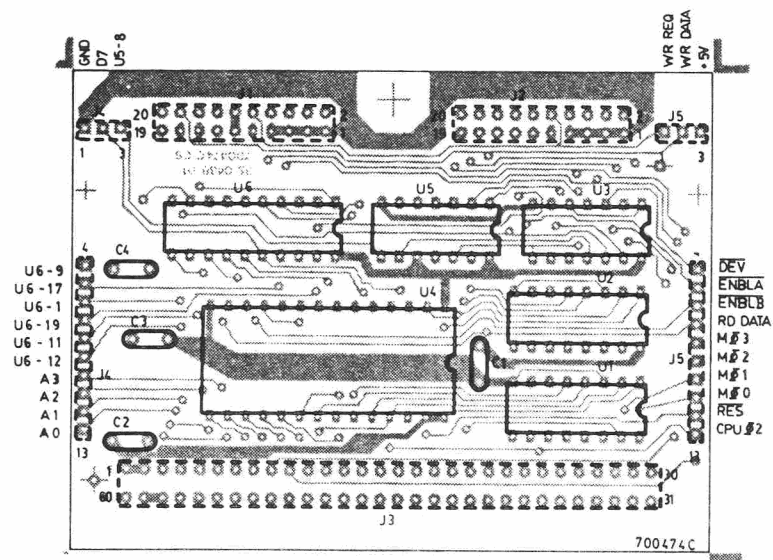
Kuva 12. Levyaseman ohjain, levy 1.



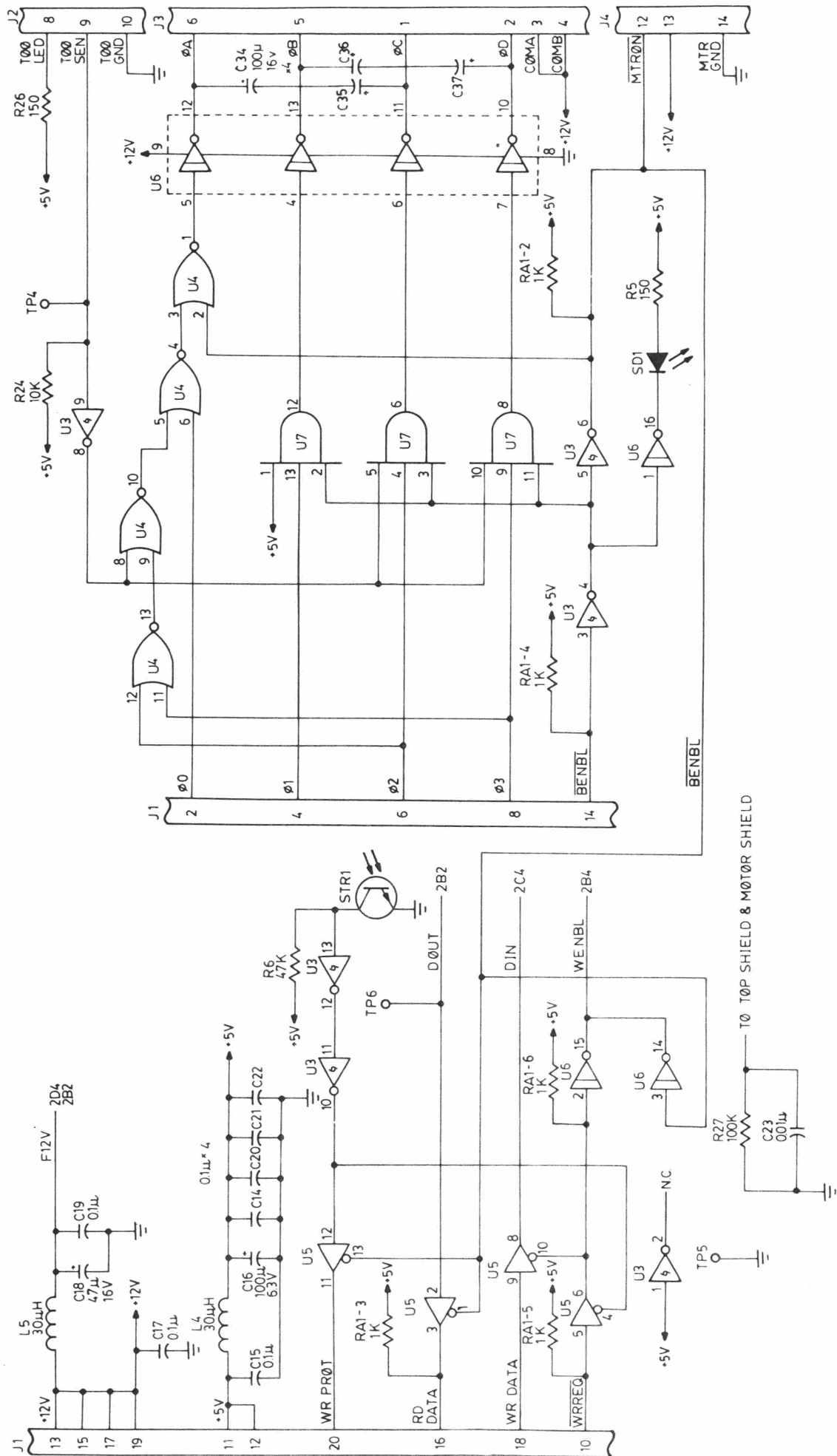
Kuva 13. Levyaseman ohjain, levy 2.



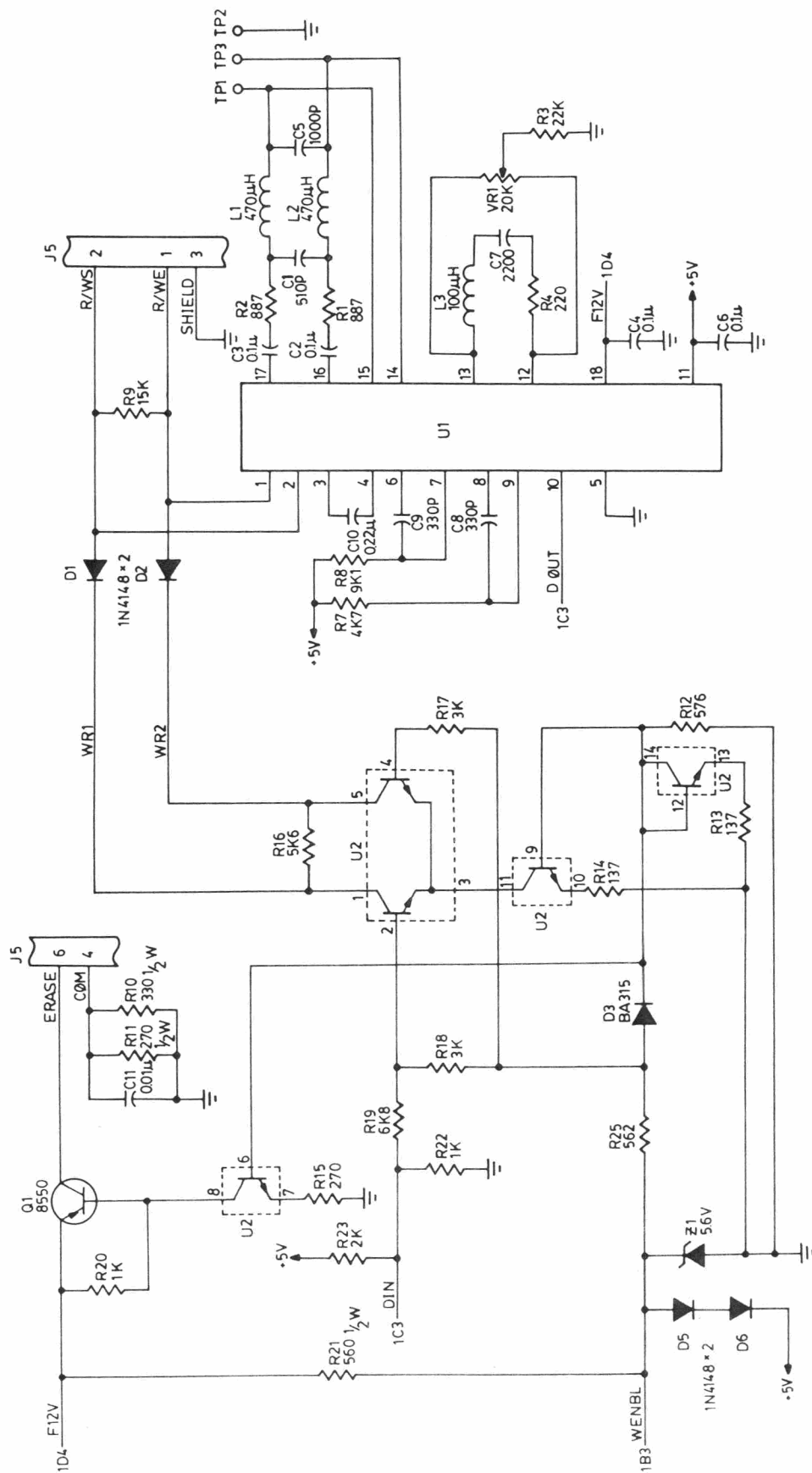
Kuva 14-a. Levyaseman ohjaimen levyn 1 osasijoittelu (ylempi levy).



Kuva 14-b. Levyaseman ohjaimen levyn 2 osasijoittelu (alempi levy).

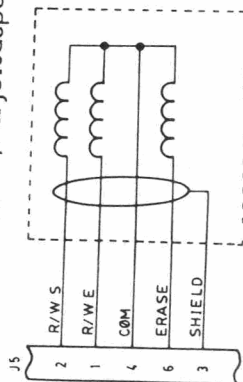


Kuva 15. Levykeasema, osa 1.

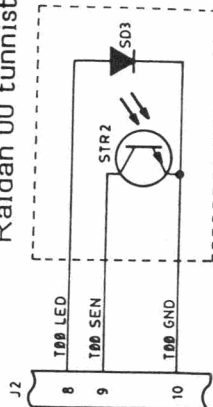


Kuva 16. Levykeasema, osa 2.

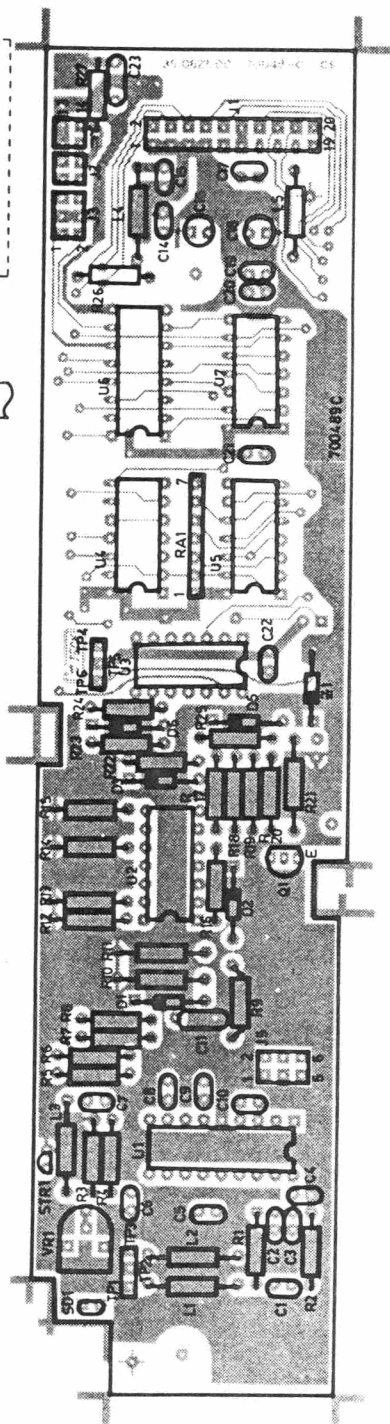
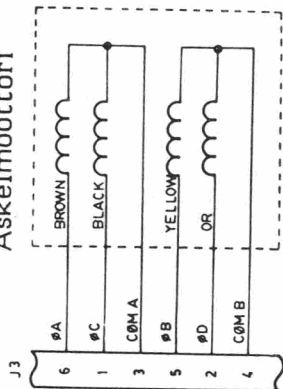
Luku/kirjoituspää



Raidan 00 tunnistin

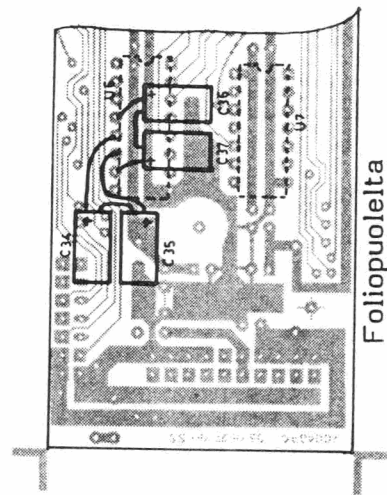


Askelmoottori

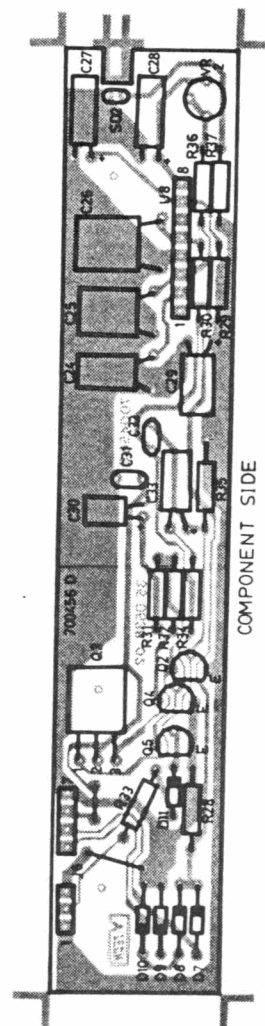
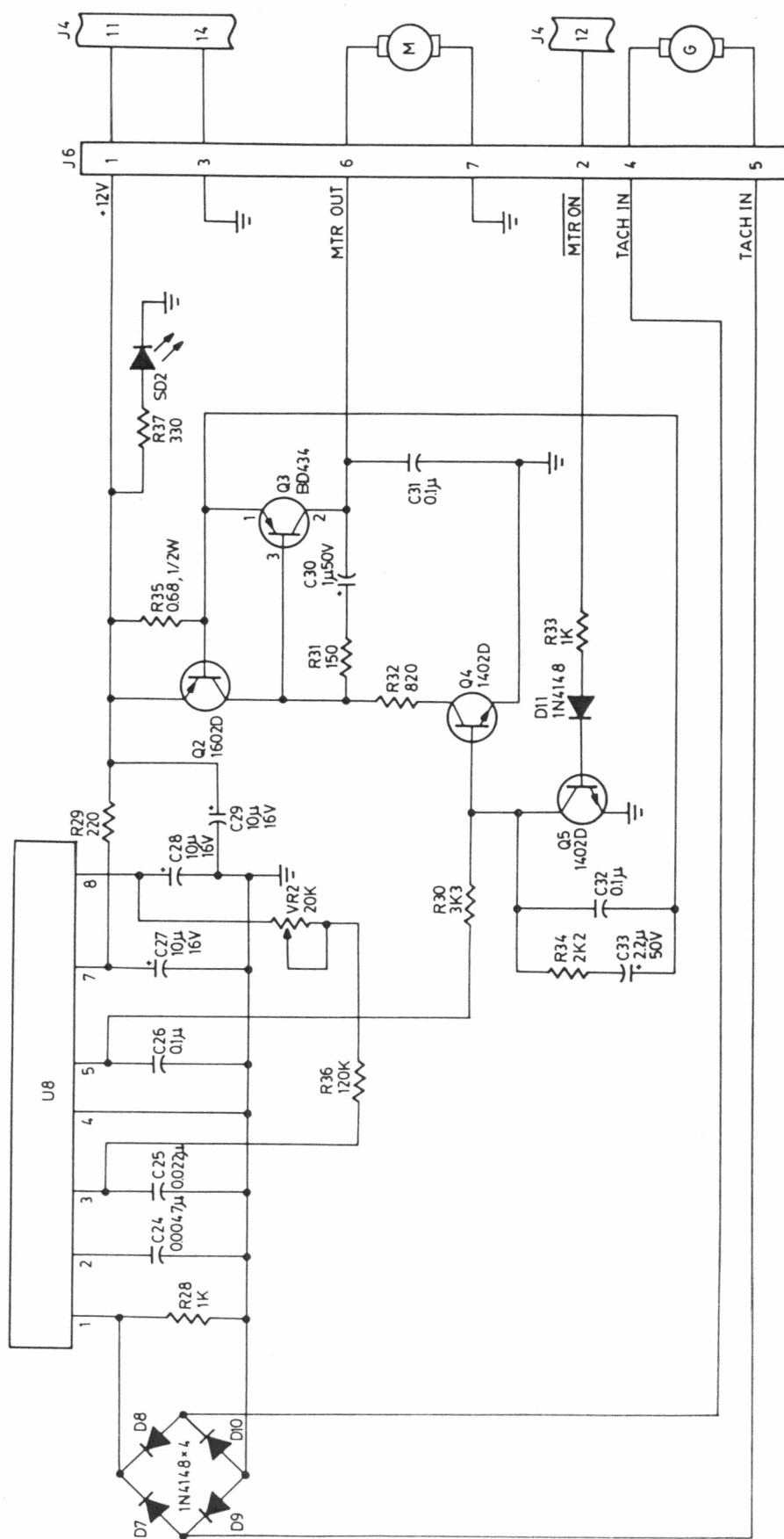


COMPONENT SIDE

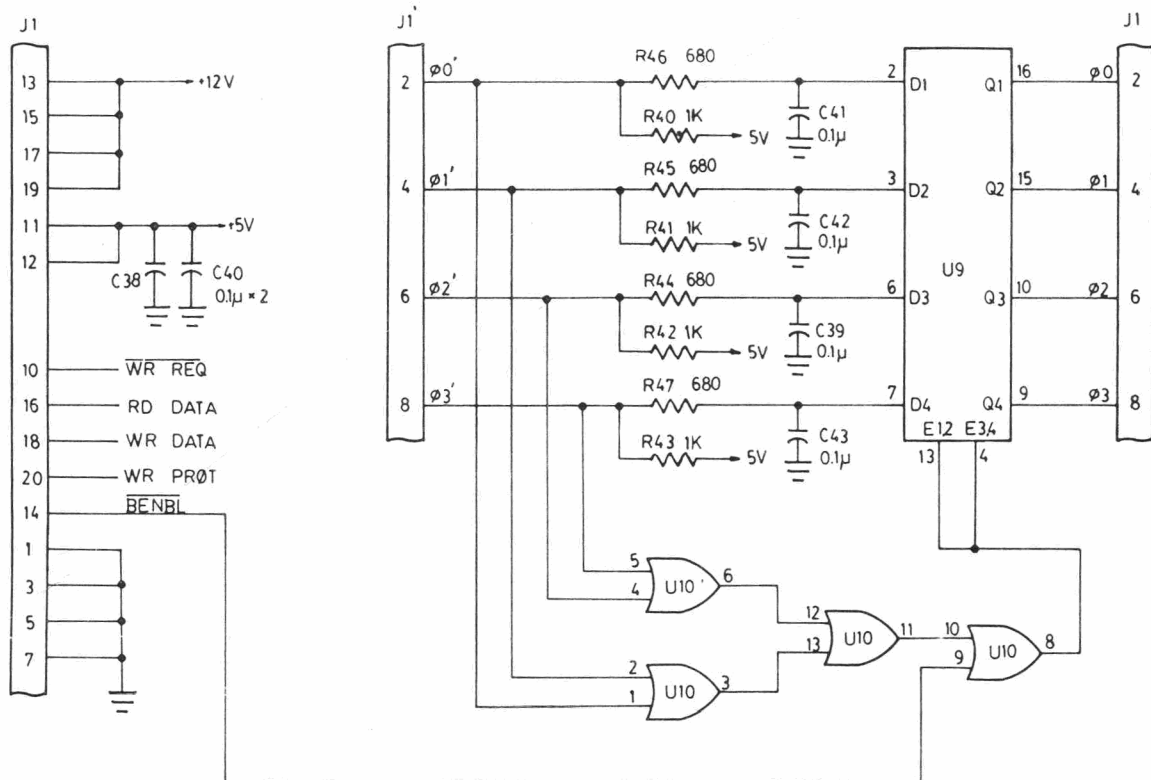
Kuva 17. Levykeaseman osasijoittelu.



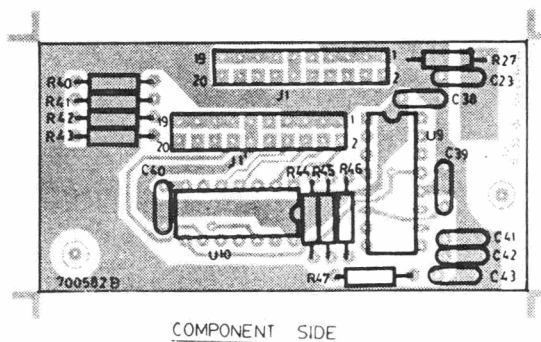
Foliopuolelta



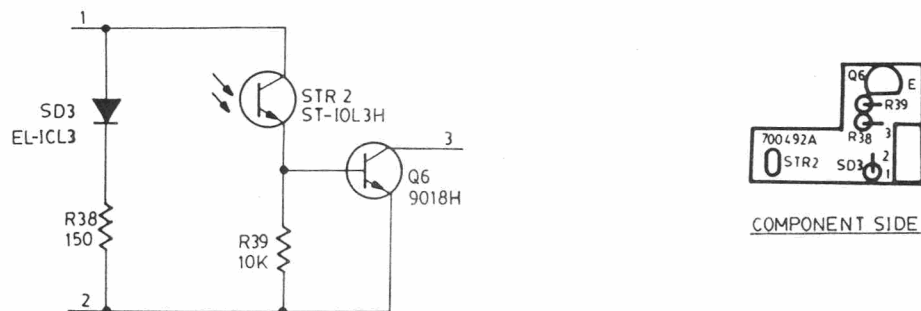
Kuva 18.
Moottorinohjauslevy.



Kuva 19-a. Askelmoottorilevyn kytkentä.



Kuva 19-b. Askelmoottorilevy.



Kuva 20. Raidan 00 tunnistin.

