Kravspecifikation

Navn:	SCU301 kravspecifikation
Dato:	94-02-17
Forfatter:	aaj/ap
Version:	7
Dokument ID:	1-0105
Status:	Godkendt

Indhold **side**

1 REVISIONSHISTORIE

Dette er den syvende udgave af dokumentet.

- I version 2 er CPU typen defineret, og SCU'en har nu adgang til den globale bus via agenten på BAIO istedet for via BAIO's VSB interface. Desuden forskellige mindre rettelser af andre afsnit.

- I version 3 er SCU'en et standard submodul til BAIO, BAIO har direkte adgang til visse funktioner på SCU'en og der benyttes en simpel microcontroller på SCU'en til styring af display, modem m.m. Der er ikke længere separat back-up port.

- I version 4 er display'et ændret, modemswitchen er flyttet til fronten af maskinen, ID er placeret på busprint og læses af SCU, prod/service har fået yderligere testmuligheder mht. blæsere og strømforsyninger, ligesom der er foretaget mindre rettelser af andre afsnit.

- I version 5 er processoren ændret til en Hitachi H8, system ID er ændret og der er ikke længere mulighed for sekundær SCU. Der er reduceret mulighed for justering af forsyningsspænding. Ur funktionen har reduceret nøjagtighed.

- I version 6 er der ændret ved reset mulighederne, beskrivelse af ID læsning og forsyningsspændingsændring er fjernet fra SCU kravspec., idet disse ting ikke længere involverer SCU'en, der er tilføjet en ekstern alarmindgang samt en HW debug tast, og der er foretaget mindre, redaktionelle rettelser i forskellige afsnit.

- I version 7 er tilføjet udgange til styring af lysdioder på diske i diskarray. Der er nu også mulighed for at detektere, at der på bagpanelet er monteret et specielt teststik.

2 REFERENCER

1. (ks.4): SPC/3 Kravspecifikation
2. 1-0102: BAIO301, kravspecifikation
3. 1-0104: SPC/3 kabinetter, kravspecifikation
4. 1-0201: SPC/3 System Bus Specification
5. 1-0204: BAIO301, design document
6. 1-0207: BAIO301 Programmers Guide
7. H8/532 92/01 10742: HD6475328 Hardware Manual, Users Manual.
8. 1-0209: SMK302 Kabinet

3 INTRODUKTION

SCU301 (Service Computer Unit) er et sevicecomputer modul til SPC/3 datamatserien. Modulet udfører forskellige system overvågnings- og hjælpefunktioner.
SCU301 udformes som et standard submodul, der monteres på et BAIO301 (basal I/O) kort (VSB interface).

SCU301 er en selvstændig enhed, der i så vid udstrækning som muligt skal fungere uafhængigt af BAIO301, og SPC/3 maskinen skal kunne køre videre, hvis SCU'en fejler.

Nogle funktioner på SCU kortet accesses dog direkte fra BAIO. SCU'en er et slave modul set fra BAIO.

4 PRODUKTBESKRIVELSE

SCU301 skal stille de nedenfor nævnte funktioner til rådighed for SPC/3 systemet.
Nogle accesses direkte fra BAIO (via submodul interface), mens andre funktioner udføres vha. kommunikation via dual port memory'en på SCU'en.

4.1 BAIO hovedmodul interface

SCU'en er forbundet til BAIO hovedmodulet via et dedikeret submodul bus stik, ref. 5, med standard submodul forbindelserne plus enkelte ekstra signaler. Ligeledes af hensyn til forbindelser til diverse sensorer o. lign. anvendes en dedikeret submodul position på BAIO.
Kommunikation mellem SCU'en og BAIO foregår via læsning og skrivning i et dual port lager mellem submodul bussen og SCU CPU bussen.
Funktioner som Real Time Clock, NV RAM og læsning af key switch udføres direkte ved læsning/skrivning fra BAIO.
Key switch kan læses af såvel SCU som BAIO.
Power-down relæet anbringes på power backplane og manøvreres via SCU.
NVRAM skal kunne læses fra SCU; kan evt. læses fra BAIO. Dette vil være opfyldt, hvis dual port memory'en implementeres som RAM med backup.
Det statusregister, der kan læses fra BAIO, indeholder også statusbits til indikering af power-on start, start efter reset af BAIO eller efter total reset. Desuden findes bit for status af SCU watchdog samt bit for indikering af specielt teststik på SCU bagpanel.

4.2 SCU-CPU

SCU'en implementeres med en Hitachi 6475328 8-bit embedded processor, som styrer/overvåger display, taster, blæsere, temperatur, LED's, UPS og modem.

4.3 Service port

SCU301 skal være forsynet med en V.24/RS232 port for tilslutning af en debug konsol. Desuden skal den have et indbygget opkaldsmodem (min. 2400 baud) til brug for fjerndiagnosticering. Tilgang til SCU'en via modeminterfacet skal kunne enables/disables via switch på fronten af maskinen. Funktionen kan kun aktiveres med key switch i stilling SYSTEM. Aktivering indikeres med lys i tast.

4.4 Ur funktion

En realtidsklokke med batteri back-up. Klokken leverer tidspunktet enten på formatet "år måned dag timer min sec" som på nuværende SM servicecomputer eller som en 48 bit tæller. Real time clock'ens tidsangivelse skal have en opløsning på mindst 0,01 sek.
Nøjagtigheden af klokken skal være bedre end 25 ppm.

4.5 SCU identifikation og revisionsniveau

SCU'en er forsynet med 2 PROM'er til lagring af hhv. type/serie nummer og revisionsniveau. Denne information kan læses fra BAIO.

4.6 Service panel

Interface til SPC/3 kabinetternes service panel, der indeholder:

- En 3 stillings omskifter med nøgle. Omskifteren har flg. positioner: OFF, RUN, SYS
- En trykknap mærket: START
- En trykknap mærket: SHUT-DOWN
- En trykknap (med lys) mærket: MODEM
- Et LCD display, der som minimum har samme funktionalitet som displayet på Supermax (Alfanumerisk, 2 x 24 karakterer).

Det skal være muligt at sætte power på maskinen uafhængigt af SCU'en.

4.7 NVM RAM

En NVM RAM på minimum 64 kbyte til logning af status information i forbindelse med system fejl. Kan evt. udgøres af en del af dual port RAM'en, hvis denne har power back-up, så informationen kan læses af SCU'en, også hvis BAIO fejler.

4.8 Aktivitetsindikator

Styring af aktivitetsindikator på kabinet. Aktivitetsindikatoren er en (evt. flere) lysdiode(r), hvor lys angiver aktivitet. Aktivitetsindikatoren skal være synlig, selv om lågen er lukket.

4.9 Blæserstyring og overvågning

Styring af kabinettets ventilatorer, således at hastigheden nedsættes (reduktion af støjniveau), hvis temperaturen tillader det. I tilfælde af fejl på SCU'en skal en sikkerhedsanordning sørge for at ventilatorerne kører med fuld hastighed.
Alle ventilatorer overvåges og ved udfald gives en advarsel til systemet.

Der er mulighed for forceret blæserstyring, d.v.s. remote styring af blæserhastighed i forbindelse med test i varmerum i prod.

4.10 Temperaturovervågning

Temperaturen overvåges på strategiske steder i kabinettet.
Temperaturen måles ved analog måling og omsættes med en opløsning på mindst 8 bit.
Overvågningen har 2 niveauer:

1. Forhøjet temperatur. D.v.s. at temperaturstigningen er så stor at der er fare for overtemperatur indenfor et kortere tidsrum.

2. Overtemperatur. D.v.s. at der er fare for fejlfunktion eller ødelæggelse af komponenter.

Ved forhøjet temperatur sendes en advarsel til systemet. Ved overtemperatur gives yderliger een advarsel (så BAIO kan nedlukke systemet).

4.11 Disk LED styring

Der skal være udgange til styring af lysdioderne (3 stk) på hver af diskene i disksystemet, se ref 8.

4.12 Remote Power ON/OFF

Der skal være 2 seperate udgange til styring af 'Remote Power ON/OFF' på power supplies.

- Een udgang til styring af kabinettets power supplies.
- Een udgang til styring af externe disk kabinetter.

Remote power ON/OFF relæet manøvreres via SCU'en og indgår i trådningen til service panelets key switch. Relæet er anbragt på power backplane.

4.13 Power supply overvågning

For hver af de individuelle power supplies (max. 8) i kabinettet overvåges signalerne:

- Power Fail, der angiver, at en af power supply'ens udgangsspændinger er udenfor de fastsatte tolerancer.

- Temperature Warning, der angiver for høj temperatur i power supply'en.

Endvidere overvåges og måles spændingerne udvalgte steder i maskinen (målinger med minimum 8 bits opløsning), og der sendes advarsel til BAIO i tilfælde af at spændingen kommer udenfor tolerancerne.

4.14 UPS interface

Indgang for advarselssignal fra UPS i tilfælde af netudfald. Indgangen skal have høj indgangsimpedans, når der ikke er spænding på SCU'en.
Der findes i UPS interface stikket mulighed for at afgøre om stikket er monteret.
Desuden findes en ekstern alarm indgang (digital indgang, hvis tilstand kan aflæses af SCU).

4.15 Reset og interrupt

Da BAIO aktiverer systembus reset efter power-up og efter programmeret reset request, giver SCU'en ikke system reset, men SCU'en skal kunne give reset separat til BAIO (separat signal).
Der findes også bits i status registeret for power-on start, reset BAIO start, og start efter total reset.
SCU'en må ikke resettes, hvis andre enheder giver reset på global bussen, og resettes selv kun i tilfælde af power on.

SCU'en får debug interrupt fra system bussen via BAIO.
Remote debugging via modem initieres via interrupt til BAIO.
Der findes yderligere en HW debug tast på fronten af SCU.

SCU'en giver endvidere et power fail signal til bussen (via BAIO) indtil forsyningsspændingerne har været inden for tolerancerne i mindst 100 mS.

SCU'en giver interrupt til BAIO i tilfælde af alarmtilstande, eller når BAIO's medvirken kræves.
BAIO kan aflæse interruptstatus via dual port memory.

4.16 Fejlhåndtering

SCU301 er forsynet med en watch-dog timer til overvågning af dens egen funktion. Ved udløb af timeren foretages flg.

- Fejllampen på fronten tændes.
- Blæserstyringen sætter ventilatorerne på fuld hastighed.
- Information om time-out gemmes
- SCU'en resetter sig selv og genstartes
- Ved opstart efter time-out skrives information om time-out på servicepanelets display.
- Der sendes et interrupt til systemet.

5 INSTALLATION og TEST

SCU301 monteres som et submodul på et BAIO301 kort i den øverste submodul position (pga interface ledninger). BAIO med SCU installeres i et SPC/3 kabinet i den reserverede system position.

Ved opstart udføres en selvtest ved start af et PROM baseret testprogram. Der startes med tændt rød LED, og denne slukkes ved slut af test OK.

6 SERVICE

Service af SCU modulet er baseret på udskiftning af defekte moduler, der repareres i DDE's reperationscenter.

Eneste forebyggende vedligeholdelse er skift af back-up batteri efter foreskrevne tidsinterval.

BAIO med SCU skal kunne trækkes ud af kortskuffen uden at afmontere andre kabler end de der er monteret på forkanten af BAIO/SCU.

Modulet er forsynet med mærkning, der viser modulbetegnelse og revisionsniveau. Ved rettelser på grundlag af FCN'er opdateres revisionsniveauet. Modulets type, serienummer og revisionsniveau kan aflæses af software og vises på servicepanelets display.

7 ANDRE KRAV

7.1 Miljø

Når SCU modulet er monteret i et SPC/3 kabinet opfylder det miljøkravene (omgivelsestemperatur, EMC, etc.) til SPC/3 moduler som beskrevet i ref.1.

7.2 Strømforbrug

max. 2 Amp @ 5V (incl. display).

8 DOKUMENTATION

Dokumentation udarbejdes i overensstemmelse med proceduren 'Hardware/Software Udvikling' (QM3100) og de dertil relaterede instruktioner.

9 ØKONOMI

max 2500 kr incl. display.

Kravspecifikation

Navn:	SCU301 kravspecifikation
Dato:	94-04-14
Forfatter:	aaj/ap
Version:	6.1
Dokument ID:	1-0105
Status:	Foreløbig

Indhold **side**

1 REVISIONSHISTORIE

- I version 2 er CPU typen defineret, og SCU'en har nu adgang til den globale bus via agenten på BAIO istedet for via BAIO's VSB interface. Desuden forskellige mindre rettelser af andre afsnit.

- I version 3 er SCU'en et standard submodul til BAIO, BAIO har direkte adgang til visse funktioner på SCU'en og der benyttes en simpel microcontroller på SCU'en til styring af display, modem m.m. Der er ikke længere separat back-up port.

- I version 4 er display'et ændret, modemswitchen er flyttet til fronten af maskinen, ID er placeret på busprint og læses af SCU, prod/service har fået yderligere testmuligheder mht. blæsere og strømforsyninger, ligesom der er foretaget mindre rettelser af andre afsnit.

- I version 5 er processoren ændret til en Hitachi H8, system ID er ændret og der er ikke længere mulighed for sekundær SCU. Der er reduceret mulighed for justering af forsyningsspænding. Ur funktionen har reduceret nøjagtighed.

- I version 6 er der ændret ved reset mulighederne, beskrivelse af ID læsning og forsyningsspændingsændring er fjernet fra SCU kravspec., idet disse ting ikke længere involverer SCU'en, der er tilføjet en ekstern alarmindgang samt en HW debug tast, og der er foretaget mindre, redaktionelle rettelser i forskellige afsnit.

- I version 6.1 er styring af lamper (LED) på hot-plug disk modul tilføjet. Et afsnit (3.1), der specificerer en 2 faset implementering af SCU301, er tilføjet.

2 REFERENCER

1. (ks.4): SPC/3 Kravspecifikation
2. 1-0102: BAIO301, kravspecifikation
3. 1-0104: SPC/3 kabinetter, kravspecifikation
4. 1-0201: SPC/3 System Bus Specification
5. 1-0204: BAIO301, design document
6. 1-0207: BAIO301 Programmers Guide
7. 1-0211: Submodule Programming Interface
8. H8/532 92/01 10742: HD6475328 Hardware Manual, Users Manual.

3 INTRODUKTION

SCU301 (Service Computer Unit) er et sevicecomputer modul til SPC/3 datamatserien. Modulet udfører forskellige system overvågnings- og hjælpefunktioner.
SCU301 udformes som et standard submodul, der monteres på et BAI0301 (basal I/O) kort (VSB interface).

SCU301 er en selvstændig enhed, der i så vid udstrækning som muligt skal fungere uafhængigt af BAI0301, og SPC/3 maskinen skal kunne køre videre, hvis SCU'en fejler.

Nogle funktioner på SCU kortet accesses dog direkte fra BAIO. SCU'en er et slave modul set fra BAIO.

3.1 Implementering

For at fremskynde tidspunktet for tilgængeligheden af SCU'en for softwareindkøring, deles implementeringen af SCU301 op i 2 faser. I første fase implementeres et SCU modul (SCU302) med et begrænset antal funktioner, der er et subset af SCU301 funktionerne. I anden fase implementeres den fulde funktionalitet. Af hensyn til styring af projektet får det SCU modul, der implementeres i fase 1, en selvstændig modulbetegnelse, SCU302. SCU302 er tænkt som en midlertidig løsning, der erstattes af SCU301.

I fase 1 implementeres følgende funktionalitet:

- Standard submodul interface
- Ur funktion
- Nøgle og trykknapper
- Power-down kontrol
- Hot-plug disk indikator kontrol
- Power supply overvågning???
- Serviceport excl. modem???

4 PRODUKTBESKRIVELSE

SCU301 skal stille de nedenfor nævnte funktioner til rådighed for SPC/3 systemet.
Nogle accesses direkte fra BAIO (via submodul interface), mens andre funktioner udføres vha. kommunikation via dual port memory'en på SCU'en.

4.1 BAIO hovedmodul interface

SCU'en er forbundet til BAIO hovedmodulet via et dedikeret submodul bus stik, ref. 5, med standard submodul forbindelserne plus enkelte ekstra signaler. Ligeledes af hensyn til forbindelser til diverse sensorer o. lign. anvendes en dedikeret submodul position på BAIO.
Kommunikation mellem SCU'en og BAIO foregår via læsning og skrivning i et dual port lager mellem submodul bussen og SCU CPU bussen.
Funktioner som Real Time Clock, NV RAM og læsning af key switch udføres direkte ved læsning/skrivning fra BAIO.
Key switch kan læses af såvel SCU som BAIO.
Power-down relæet anbringes på power backplane og manøvreres via SCU.
NVRAM skal kunne læses fra SCU; kan evt. læses fra BAIO. Dette vil være opfyldt, hvis dual port memory'en implementeres som RAM med backup.
Det statusregister, der kan læses fra BAIO, indeholder også statusbits til indikering af power-on start, start efter reset af BAIO eller efter total reset. Desuden findes bit for status af SCU watchdog.

4.2 SCU-CPU

SCU'en implementeres med en Hitachi 6475328 8-bit embedded processor, som styrer/overvåger display, taster, blæsere, temperatur, LED's, UPS og modem.

4.3 Standard submodul interface

SCU'en har et standard submodul interface som beskrevet i reference 7.

4.4 Service port

SCU301 skal være forsynet med en V.24/RS232 port for tilslutning af en debug konsol. Desuden skal den have et indbygget opkaldsmodem (min. 2400 baud) til brug for fjerndiagnosticering.
Tilgang til SCU'en via modeminterfacet skal kunne enables/disables via switch på fronten af maskinen. Funktionen kan kun aktiveres med key switch i stilling SYSTEM. Aktivering indikeres med lys i tast.

4.5 Ur funktion

En realtidsklokke med batteri back-up. Klokken leverer tidspunktet enten på formatet "år måned dag timer min sec" som på nuværende SM servicecomputer eller som en 48 bit tæller. Real time clock'ens tidsangivelse skal have en opløsning på mindst 0,01 sek.
Nøjagtigheden af klokken skal være bedre end 25 ppm.

4.6 Switch panel

Interface til SPC/3 kabinetternes switch panel, der indeholder:

- En 3 stillings omskifter med nøgle. Omskifteren har flg. positioner: OFF, RUN, SYS
- En trykknap mærket: START
- En trykknap mærket: SHUT-DOWN
- En trykknap (med lys) mærket: MODEM

4.7 LCD display

Interface til et LCD display, der som minimum har samme funktionalitet som displayet på Supermax (Alfanumerisk, 2 x 24 karakterer).

4.8 Aktivitetsindikator

Styring af aktivitetsindikator på kabinet. Aktivitetsindikatoren er en (evt. flere) lysdiode(r), hvor lys angiver aktivitet. Aktivitetsindikatoren skal være synlig, selv om lågen er lukket.

4.9 NVM RAM

En NVM RAM på minimum 64 kbyte til logning af status information i forbindelse med system fejl. Kan evt. udgøres af en del af dual port RAM'en, hvis denne har power back-up, så informationen kan læses af SCU'en, også hvis BAIO fejler.

4.10 Blæserstyring og overvågning

Styring af kabinettets ventilatorer, således at hastigheden nedsættes (reduktion af støjniveau), hvis temperaturen tillader det. I tilfælde af fejl på SCU'en skal en sikkerhedsanordning sørge for at ventilatorerne kører med fuld hastighed.
Alle ventilatorer overvåges og ved udfald gives en advarsel til systemet.

Der er mulighed for forceret blæserstyring, d.v.s. remote styring af blæserhastighed i forbindelse med test i varmerum i prod.

4.11 Temperaturovervågning

Temperaturen overvåges på strategiske steder i kabinettet.
Temperaturen måles ved analog måling og omsættes med en opløsning på mindst 8 bit.
Overvågningen har 2 niveauer:

1. Forhøjet temperatur. D.v.s. at temperaturstigningen er så stor at der er fare for overtemperatur indenfor et kortere tidsrum.

2. Overtemperatur. D.v.s. at der er fare for fejlfunktion eller ødelæggelse af komponenter.

Ved forhøjet temperatur sendes en advarsel til systemet. Ved overtemperatur gives yderliger een advarsel (så BAIO kan nedlukke systemet).

4.12 DC power ON/OFF

Der skal være 2 seperate udgange til styring af 'DC power ON/OFF' på power supplies.

- Een udgang til styring af kabinettets power supplies.
- Een udgang til styring af externe disk kabinetter.

DC power ON/OFF relæet styres af SCU og switch panelets nøgle og start trykknap, således at det er muligt at sætte DC power på maskinen uden at SCU'en er monteret. Programmeret DC power OFF styres af SCU'en.

4.13 Power supply overvågning

For hver af de individuelle power supplies (max. 8) i kabinettet overvåges signalerne:

- Power Fail, der angiver, at en af power supply'ens udgangsspændinger er udenfor de fastsatte tolerancer.

- Temperature Warning, der angiver for høj temperatur i power supply'en.

Endvidere overvåges og måles spændingerne udvalgte steder i maskinen (målinger med minimum 8 bits opløsning), og der sendes advarsel til BAIO i tilfælde af at spændingen kommer udenfor tolerancerne.

4.14 UPS interface

Indgang for advarselssignal fra UPS i tilfælde af netudfald.
Indgangen skal have høj indgangsimpedans, når der ikke er spænding på SCU'en.
Der findes i UPS interface stikket mulighed for at afgøre om stikket er monteret.
Desuden findes en ekstern alarm indgang (digital indgang, hvis tilstand kan aflæses af SCU).

4.15 Hot-plug disk indikator kontrol

SCU'en er forsynet med udgange til styring af LED indikatorerne på hot-plug disk modulerne.

4.16 Reset og interrupt

Da BAIO aktiverer systembus reset efter power-up og efter programmeret reset request, giver SCU'en ikke system reset, men SCU'en skal kunne give reset separat til BAIO (separat signal).
Der findes også bits i status registeret for power-on start, reset BAIO start, og start efter total reset.
SCU'en må ikke resettes, hvis andre enheder giver reset på global bussen, og resettes selv kun i tilfælde af power on.

SCU'en får debug interrupt fra system bussen via BAIO.
Remote debugging via modem initieres via interrupt til BAIO.
Der findes yderligere en HW debug tast på fronten af SCU.

SCU'en giver endvidere et power fail signal til bussen (via BAIO) indtil forsyningsspændingerne har været inden for tolerancerne i mindst 100 mS.

SCU'en giver interrupt til BAIO i tilfælde af alarmtilstande, eller når BAIO's medvirken kræves.
BAIO kan aflæse interruptstatus via dual port memory.

4.17 Fejlhåndtering

SCU301 er forsynet med en watch-dog timer til overvågning af dens egen funktion. Ved udløb af timeren foretages flg.

- Fejllampen på fronten tændes.
- Blæserstyringen sætter ventilatorerne på fuld hastighed.
- Information om time-out gemmes
- SCU'en resetter sig selv og genstartes
- Ved opstart efter time-out skrives information om time-out på servicepanelets display.
- Der sendes et interrupt til systemet.

5 INSTALLATION og TEST

SCU301 monteres som et submodul på et BAIO301 kort i den øverste submodul position (pga interface ledninger). BAIO med SCU installeres i et SPC/3 kabinet i den reserverede system position.

Ved opstart udføres en selvtest ved start af et PROM baseret testprogram. Der startes med tændt rød LED, og denne slukkes ved slut af test OK.

6 SERVICE

Service af SCU modulet er baseret på udskiftning af defekte moduler, der repareres i DDE's reperationscenter.

Eneste forebyggende vedligeholdelse er skift af back-up batteri efter foreskrevne tidsinterval.

BAIO med SCU skal kunne trækkes ud af kortskuffen uden at afmontere andre kabler end de der er monteret på forkanten af BAIO/SCU.

Modulet er forsynet med mærkning, der viser modulbetegnelse og revisionsniveau. Ved rettelser på grundlag af FCN'er opdateres revisionsniveauet. Modulets type, serienummer og revisionsniveau kan aflæses af software og vises på servicepanelets display.

7 ANDRE KRAV

7.1 Miljø

Når SCU modulet er monteret i et SPC/3 kabinet opfylder det miljøkravene (omgivelsestemperatur, EMC, etc.) til SPC/3 moduler som beskrevet i ref.1.

7.2 Strømforbrug

max. 2 Amp @ 5V (incl. display).

8 DOKUMENTATION

Dokumentation udarbejdes i overensstemmelse med proceduren 'Hardware/Software Udvikling' (QM3100) og de dertil relaterede instruktioner.

9 ØKONOMI

max 2500 kr incl. display.

DESIGN DOCUMENT for SCU302 Servicecomp.

Title:	SCU302 Programmers Guide
Date:	94.05.10
Author:	kan
Version:	1
Id:	xxxxxx
Status:	Preliminary

1. Revisions:

This is the first version of the document.

2. References:

2.1 1-0105 SCU301 Kravspecifikation.
2.2 1-0212 SCU301 Programmers Guide
2.3 Dallas Semiconductor 1992-1993 Product Data Book
2.4 ADC0848 National Linear Databook 2.
2.5 Spec. No. MG1216B1N000-002, Seiko Instruments Inc.
2.6 1-0210 BAIO-SMI Design Document.
2.7 1-0211 BAIO-SMPI Design Document.
2.8 1-0209 SMK302 Kabinet Design Document.

3. Introduction:

The SCU303 is a simplified version of SCU301 described in ref. 2.1 and ref. 2.2. SCU303 implements various check and controlfunctions in the SPC/3 computer system.

The module is designed as a standard submodule to be mounted in the dedicated VSB submodule port on the BAIO module.

The SCU303 is a simple slave VSB module seen from the BAIO.

The sensor inputs and outputs for the SCU303 are connected via a 220 pin connector at the rear of the SCU module.

The submodule is designed around the following main components:

Dallas DS1486 NVRAM/RTC chip
Seiko Instruments G1216B1N000 LCD display (128x64 dots)

A description of the registers and functions can be found in the document listed in ref. 2.3 and ref. 2.5.

4. Description:

Submodule in position 1 of BAIO: Base addr. = 0x08000000.

The following table shows the addresses that the SCU303 submodule will respond to when an access is made from the BAIO module.

The following addresses are all offsets from the base address:

Addr: (BAIOBIG=1)	Use:
0x00003 to 0x0007F	Submodule ID PROM.
0x00083 to 0x000FF	Submodule Revision PROM.
0x00103	Submodule status register.
0x00183	Submodule control register.
0x00203	Keyswitch register.
0x00283	Power control register.
0x00303	PS Config register.
0x00383	PS Temp. check register.
0x00403	Power check register.
0x00483	Lamp control register.
0x00503	Disk control register 0.
0x00583	Disk control register 1.
0x00603	Activity LED control register 0.
0x00683	Activity LED control register 1.
0x00703	Analog/digital converter 0.
0x00783	Analog/digital converter 1.
0x00803	Fan control register A.
0x00883	Fan control register B.
0x00903	Alarm register.
0x00a03	Display.
0x20000 to 0x2003C	Real Time Clock registers.
0x20003 to 0x2003F	Real Time Clock registers.
0x20043 to 0x9FFFF	Non volatile RAM, 128k bytes.

Table of all addresses used on the submodule (for BAIOBIG=1).

The following addresses are all offsets from the base address:

Addr: (BAIOBIG=0)	Use:
0x00000 to 0x0007C	Read submodule ID PROM.
0x00080 to 0x000FC	Read submodule Revision PROM.
0x00100	Submodule status register.
0x00180	Submodule control register.
0x00200	Keyswitch register.
0x00280	Power control register.
0x00300	PS Config register.
0x00380	PS Temp. check register.
0x00400	Power check register.
0x00480	Lamp control register.
0x00500	Disk control register 0.
0x00580	Disk control register 1.
0x00600	Activity LED control register 0.
0x00680	Activity LED control register 1.
0x00700	Analog/digital converter 0.
0x00780	Analog/digital converter 1.
0x00800	Fan control register A.
0x00880	Fan control register B.
0x00900	Alarm register.
0x00a00	Display.
0x20000 to 0x2003C	Real Time Clock registers.
0x20040 to 0x9FFFC	Non volatile RAM, 128k bytes.

Table of all addresses used on the submodule (for BAIOBIG=0).

All register accesses to the submodule are physically 8-bit accesses, and the data lines are connected to DA07-DA00. The BAIO must address the registers according to the chosen BAIOBIG. The SCU checks/supplies the correct parity for the addressed byte.

4.1. Description of registers:

Submodule ID PROM.	
BIG=1; Addr:	0x003 to 0x07F, 32 byte addresses.
BIG=0; Addr:	0x000 to 0x07C, 32 byte addresses.
Access types:	Byte. Read only.
32 Bytes indicating submodule TYPE and Serial number. Refer to BAIO SMPI Design Document.	

Submodule Revision PROM.		
BIG=1; Addr:	0x083 to 0x0FF, 32 byte addresses.	
BIG=0; Addr:	0x080 to 0x0FC, 32 byte addresses.	
Access types:	Byte. Read only.	
32 Bytes indicating submodule revision level.		
Refer to BAIO SMPI Design Document.		

STATUS register.		
BIG=1; Addr:	0x103	
BIG=0; Addr:	0x100	
Access types:	Byte. Read/write.	
STATUS0 (07)	Interrupt req.	Active high signal indicating interrupt request from one or more sources.
STATUS0 (06)	Pending Intr.	Pending Interrupt Request.
STATUS0 (05)	Pending PERR	*Pending Parity Fault Interrupt
STATUS0 (04)	Not used	Not used.
STATUS0 (03)	Type bit 3	*Parity Fault in byte 3
STATUS0 (02)	Type bit 2	*Parity Fault in byte 2
STATUS0 (01)	Type bit 1	*Parity Fault in byte 1
STATUS0 (00)	Type bit 0	*Parity Fault in byte 0

*) These bits are cleared by a write to the register address.

CONTROL register.		
BIG=1; Addr:	0x183	
BIG=0; Addr:	0x180	
Access types:	Byte. Read/write.	
CONTROL (07)	INTRENA	*Enable interrupts from SCU.
CONTROL (06)	PFINTENA	*Enable parity fault intr.
CONTROL (05)	ODD	*Odd parity.
CONTROL (04)	T.B.D.	*Not used.
CONTROL (03)	T.B.D.	*Not used.
CONTROL (02)	T.B.D.	*Not used.
CONTROL (01)	T.B.D.	*Not used.
CONTROL (00)	T.B.D.	*Not used.

*) Low on power-up or after SMI reset.

Keyswitch register.		
BIG=1; Addr:	0x203	
BIG=0; Addr:	0x200	
Access types:	Byte. Read only.	
KEYSTAT (07)	SYSTEM	Key in pos. SYST.
KEYSTAT (06)	START	START activated.
KEYSTAT (05)	SHUTDOWN	Shutdown activated.
KEYSTAT (04)	MODEM	Modem activated.
KEYSTAT (03)	T.B.D.	Not used.
KEYSTAT (02)	LAMP1	LAMP1 is ON.
KEYSTAT (01)	LAMP2	LAMP2 is ON.
KEYSTAT (00)	LAMP3	LAMP3 is ON.

Register: Power Control Register.	
BIG=1; Addr:	0x283
BIG=0; Addr:	0x280
Access types:	Write only.

Bit:	Use:
D7	Not used.
D6	PSHIGH, activate high voltage setting on PS
D5	PSLOW, activate low voltage setting on PS
D4	Power-down Control, internal PS and external disks
D3	Not used.
D2	Not used.
D1	Not used.
D0	Not used.

Register: PS Config Register.	
BIG=1; Addr:	0x303
BIG=0; Addr:	0x300
Access types:	Read only.

Bit:	Use:
D7	PS unmounted signal, from Power Supply in slot no. 7
D6	PS unmounted signal, from Power Supply in slot no. 6
D5	PS unmounted signal, from Power Supply in slot no. 5
D4	PS unmounted signal, from Power Supply in slot no. 4
D3	PS unmounted signal, from Power Supply in slot no. 3
D2	PS unmounted signal, from Power Supply in slot no. 2
D1	PS unmounted signal, from Power Supply in slot no. 1
D0	PS unmounted signal, from Power Supply in slot no. 0

Register: Temp. Check Register (Temp check signal from PS)	
BIG=1; Addr:	0x383
BIG=0; Addr:	0x380
Access types:	Read only.

Bit:	Use:
D7	Temperature Check OK from Power Supply in slot no. 7
D6	Temperature Check OK from Power Supply in slot no. 6
D5	Temperature Check OK from Power Supply in slot no. 5
D4	Temperature Check OK from Power Supply in slot no. 4
D3	Temperature Check OK from Power Supply in slot no. 3
D2	Temperature Check OK from Power Supply in slot no. 2
D1	Temperature Check OK from Power Supply in slot no. 1
D0	Temperature Check OK from Power Supply in slot no. 0

Register: Power Check Register (from Power Supplies).	
BIG=1; Addr:	0x403
BIG=0; Addr:	0x400
Access types:	Read only.

Bit:	Use:
D7	Power Check OK from Power Supply in slot no. 7
D6	Power Check OK from Power Supply in slot no. 6
D5	Power Check OK from Power Supply in slot no. 5
D4	Power Check OK from Power Supply in slot no. 4
D3	Power Check OK from Power Supply in slot no. 3
D2	Power Check OK from Power Supply in slot no. 2
D1	Power Check OK from Power Supply in slot no. 1
D0	Power Check OK from Power Supply in slot no. 0

Register: LAMP control Register (Keyswitch lamps)	
BIG=1; Addr:	0x483
BIG=0; Addr:	0x480
Access types:	Write only.

Bit:	Use:
D7	Error LED control (Green)
D6	Error LED control (Red)
D5	None
D4	None
D3	None
D2	Lamp3 control (Modem)
D1	Lamp2 control (Shutdown)
D0	Lamp1 control (Start)

Register: Disk Control Register 0.	
BIG=1; Addr:	0x503
BIG=0; Addr:	0x500
Access types:	Write only.

Bit:	Use:
D7	None
D6	SCSI channel bit 2.
D5	SCSI channel bit 1
D4	SCSI channel bit 0.
D3	SCSI ID bit 3
D2	SCSI ID bit 2
D1	SCSI ID bit 1
D0	SCSI ID bit 0

Register: Disk Control Register 1.	
BIG=1; Addr:	0x583
BIG=0; Addr:	0x580
Access types:	Write only.

Bit:	Use:
D7	None
D6	None
D5	LED address bit 1
D4	LED address bit 0
D3	Strobe
D2	None
D1	None
D0	LED data

Register: Activity LED CONTROL REGISTER 0	
BIG=1; Addr:	0x603
BIG=0; Addr:	0x600
Access types:	Write only.

Bit:	Use:
D7	Led7
D6	Led6
D5	Led5
D4	Led4
D3	Led3
D2	Led2
D1	Led1
D0	Led0

Register: Activity LED CONTROL REGISTER 1.	
BIG=1; Addr:	0x683
BIG=0; Addr:	0x680
Access types:	Write only.

Bit:	Use:
D7	Led15
D6	Led14
D5	Led13
D4	Led12
D3	Led11
D2	Led10
D1	Led9
D0	Led8

Register: Analog/digital converter 0.		
BIG=1; Addr:	0x703	
BIG=0; Addr:	0x700	
Access types:	Read/write.	
Bit:	Use: Read Converted Data	Write Channel Select
D7	dat7	not used
D6	dat6	not used
D5	dat5	not used
D4	dat4	MA4
D3	dat3	MA3
D2	dat2	MA2
D1	dat1	MA1
D0	dat0	MA0

See ref. x.x.

Register: Analog/digital converter 1.		
BIG=1; Addr:	0x783	
BIG=0; Addr:	0x780	
Access types:	Read/write.	
Bit:	Use: Read Converted Data	Write Channel Select
D7	dat7	not used
D6	dat6	not used
D5	dat5	not used
D4	dat4	MA4
D3	dat3	MA3
D2	dat2	MA2
D1	dat1	MA1
D0	dat0	MA0

See ref x.x.

Register: Fan Control 0	
BIG=1; Addr:	0x803
BIG=0; Addr:	0x800
Access types:	Write

Bit:	Use:
D7	not used
D6	not used
D5	fs5
D4	fs4
D3	fs3
D2	fs2
D1	fs1
D0	fs0

63 levels fan speed control.
Full speed: All ones and all zeros

Register: Fan Control 1	
BIG=1; Addr:	0x803
BIG=0; Addr:	0x800
Access types:	Write

Bit:	Use:
D7	not used
D6	not used
D5	fs5
D4	fs4
D3	fs3
D2	fs2
D1	fs1
D0	fs0

63 levels fan speed control.
Full speed: All ones and all zeros

Register: Fan Tacho Err and UPS/Alarm register	
BIG=1; Addr:	0x903
BIG=0; Addr:	0x900
Access types:	Read

Bit:	Use:
D7	UPS Off Line
D6	UPS not connected
D5	External Alarm
D4	Alarm not connected
D3	not used
D2	not used
D1	Tacho Error from Fan Control Unit 1
D0	Tacho Error from Fan control Unit 0

Register: Display Control Registers. 4 registers	
BIG=1; Addr:	0xa03, 0xa83, 0xb03, 0xb83
BIG=0; Addr:	0xa00, 0xa80, 0xb00, 0xb80
Access types:	Write/Read

Reg:	Use:
0xa00	Chip select 1. Data Write/Read
0xa80	Chip select 1. Instruction Write/Read
0xb00	Chip select 2. Data Write/Read
0xb80	Chip select 2. Instruction Write/Read

Register: Display Contrast Setting	
BIG=1; Addr:	0x983
BIG=0; Addr:	0x980
Access types:	Write

Bit:	Use:
D7	Select
D6	Up/down
D5	not used
D4	not used
D3	not used
D2	not used
D1	not used
0xb00	Chip select 2. Data Write/Read
D0	Strobe
0xb80	Chip select 2. Instruction Write/Read

Base address for non volatile memory on submodule	
BIG=1; Addr:	0x20003 to 0x9FFFF (128k addresses).
BIG=0; Addr:	0x20000 to 0x9FFFC (128k addresses).
Access types:	Byte. Read/write.

Lower 16 addresses are registers for the Real Time Clock.
See ref. 2.3.

5. Connections to SCU via SCU front panel:

- One HW Debug switch.

- Two Error indicators (LED).

The HW Debug switch activates the DEBUG-S signal to the BAIO. The SCU will then receive an interrupt via DEBUG-R from BAIO. The DEBUG-S signal can also be activated by the SCU.

The Error LEDs can be switched on and off by the program running on BAIO.

6. SCU Backpanel Connections:

The following is a description of all connections from the Service Computer Backplane to the SPC/3 computersystem. Not all connections are used by SCU303.

6.1. Connections from SCU to Activity LED's:

Pin:	Module:	Signal:
1	Activity	+5V
2	Activity	+5V
3	Activity	ALED15
4	Activity	ALED14
5	Activity	ALED13
6	Activity	ALED12
7	Activity	ALED11
8	Activity	ALED10
9	Activity	ALED9
10	Activity	ALED8
11	Activity	ALED7
12	Activity	ALED6
13	Activity	ALED5
14	Activity	ALED4
15	Activity	ALED3
16	Activity	ALED2
17	Activity	ALED1
18	Activity	ALED0
19	Activity	+5V
20	Activity	+5V

6.2. Connections from SCU to Keyswitch:

Pin:	Module:	Signal:
1	Keyswitch	+5V
2	Keyswitch	SYSTEM
3	Keyswitch	KEYRES
4	Keyswitch	GND
5	Keyswitch	SHUTDWN
6	Keyswitch	MODEM
7	Keyswitch	GND
8	Keyswitch	POWDWN
9	Keyswitch	POWUP
10	Keyswitch	OFF1
11	Keyswitch	GND
12	Keyswitch	LAMP1
13	Keyswitch	+12V
14	Keyswitch	LAMP2
15	Keyswitch	+12V
16	Keyswitch	LAMP3

6.3. Connections from SCU to Display:

Pin:	Module:	Signal:	
1	Display	+5V to display	
2	Display	+5V to display	
3	Display	Vlc	Contrast reg.
4	Display	DB0	
5	Display	DB1	
6	Display	DB2	
7	Display	DB3	
8	Display	DB4	
9	Display	DB5	
10	Display	DB6	
11	Display	DB7	
12	Display	CS1	Select1, active low
13	Display	CS2	Select2, active low
14	Display	RST	Reset, active low
15	Display	R/W	Read/Write
16	Display	D/I	Data/Instr. Register select
17	Display	GND	
18	Display	GND	
19	Display	E	Strobe, active high
20	Display	GND	
21	Display	LEDA	
22	Display	LEDC	

6.4. Connections from SCU to Disk Backplane:

Pin:	Module:	Signal:
1	DBP	GND
2	DBP	IDCH2 SCSI Channel ID, bit 2
3	DBP	IDCH1 SCSI Channel ID, bit 1
4	DBP	IDCH0 SCSI Channel ID, bit 0
5	DBP	GND
6	DBP	GND
7	DBP	IDAD3 SCSI Address, bit 3
8	DBP	IDAD2 SCSI Address, bit 2
9	DBP	IDAD1 SCSI Address, bit 1
10	DBP	IDAD0 SCSI Address, bit 0
11	DBP	GND
12	DBP	GND
13	DBP	LADR1 LED Address, bit 1
14	DBP	LADR0 LED Address, bit 0
15	DBP	GND
16	DBP	LEDDA LED Data
17	DBP	GND
18	DBP	GND
19	DBP	LSTRB Strobe
20	DBP	GND

6.5. Connections from SCU to temperature sensors, 8 blocks:

Pin:	Module:	Signal:	
1	SENSOR	+5V	
2	SENSOR	Vtemp	Signal from sensor
3	SENSOR	AGND	
4	SENSOR	GND	

6.6. Connections from SCU to Fan Control Board, 4 blocks:

Only two blocks used by SCU303

Pin:	Module:	Signal:	
1	FANCTRL	FANSPEED	PWM control signal from SCU
2	FANCTRL	GND	
3	FANCTRL	FERR	Error signal from controller
4	FANCTRL	GND	

6.7. Connections from SCU to UPS/Alarm:

Pin:	Module:	Signal:	
1	UPS	UPSSG	UPS status signal (input)
2	UPS	GND	
3	UPS	UPSCNN	UPS not connected.
4	UPS	ALSG	External Alarm input
5	UPS	GND	
6	UPS	ALCNN	External Alarm not connected
7	UPS	DSKOFF	Disk cabinet control output
8	UPS	DSKGND	Disk Cabinet GND

6.5. Connections from SCU to temperature sensors, 8 blocks:

Pin:	Module:	Signal:	
1	SENSOR	+5V	
2	SENSOR	Vtemp	Signal from sensor
3	SENSOR	AGND	
4	SENSOR	GND	

6.6. Connections from SCU to Fan Control Board, 4 blocks:

Only two blocks used by SCU303

Pin:	Module:	Signal:	
1	FANCTRL	FANSPEED	PWM control signal from SCU
2	FANCTRL	GND	
3	FANCTRL	FERR	Error signal from controller
4	FANCTRL	GND	

6.7. Connections from SCU to UPS/Alarm:

Pin:	Module:	Signal:	
1	UPS	UPSSG	UPS status signal (input)
2	UPS	GND	
3	UPS	UPSCNN	UPS not connected.
4	UPS	ALSG	External Alarm input
5	UPS	GND	
6	UPS	ALCNN	External Alarm not connected
7	UPS	DSKOFF	Disk cabinet control output
8	UPS	DSKGND	Disk Cabinet GND

6.8. Connections from SCU to Power Backplane:

Pin:	Module:	Signal:
1	PDB	Power fail signal from PS1
2	PDB	Unmounted from PS1
3	PDB	GND
4	PDB	Temp OK signal from PS1
5	PDB	Power fail signal from PS2
6	PDB	Unmounted from PS2
7	PDB	GND
8	PDB	Temp OK signal from PS2
9	PDB	Power fail signal from PS3
10	PDB	Unmounted from PS3
11	PDB	GND
12	PDB	Temp OK signal from PS3
13	PDB	Power fail signal from PS4
14	PDB	Unmounted from PS4
15	PDB	GND
16	PDB	Temp OK signal from PS4
17	PDB	Power fail signal from PS5
18	PDB	Unmounted from PS5
19	PDB	GND
20	PDB	Temp OK signal from PS5
21	PDB	Power fail signal from PS6
22	PDB	Unmounted from PS6
23	PDB	GND
24	PDB	Temp OK signal from PS6
25	PDB	Power fail signal from PS7
26	PDB	Unmounted from PS7
27	PDB	GND
28	PDB	Temp OK signal from PS7
29	PDB	Power fail signal from PS7
30	PDB	Unmounted from PS7
31	PDB	SETLOW
32	PDB	Temp OK signal from PS7
33	PDB	SETHIGH
34	PDB	REMOTE

6.9. Connections from SCU to Bus Backplane:

Not used by SCU302.

Pin:	Module:	Signal:
1	BPL	Check +3.3V (sense +3.3V)
2	BPL	Check +3.3V (return +3.3V)
3	BPL	Check +5V (sense +5V)
4	BPL	Check +5V (return +5V)
5	BPL	Check +12V (sense +12V)
6	BPL	Check +12V (return +12V)
7	BPL	Check Reserved Power (sense)
8	BPL	Check Reserved Power (return)
9	BPL	Check +2.1V (sense +2.1V left)
10	BPL	Check +2.1V (return +2.1V left)
11	BPL	Check +2.1V (sense +2.1V right)
12	BPL	Check +2.1V (return +2.1V right)
13	BPL	GND
14	BPL	GND
15	BPL	GND
16	BPL	GND
17	BPL	GND
18	BPL	GND
19	BPL	KEYRES
20	BPL	GND

7. Connections to BAIO, Submodule Interface:

See ref. 2.6

mudv2/aaj/spc3/scu301/scu301dd.v2.0

DESIGN DOCUMENT for SCU301 Servicecomp.

Title:	SCU301 Programmers Guide
Date:	94.02.16
Author:	Allan Petersen
Version:	2
Id:	1-0212
Status:	Preliminary

1. Revisions:

This is the second version of the document.

V1->V2: Access type of SMI registers is byte read. Addresses for all registers and also for BAIOBIG=0 now specified. Bits in SMI Status register changed. SMI Control regster can also be read. Bit definition changed in Keyswitch register. Timeout periode for watchdog specified. Strap check register removed. Connections for PSLOW and PSHIGH added to SCU backpanel connections. Various editorial corrections.

2. References:

2.1 1-0105 SCU301 Kravspecifikation.
2.2 H8/532 92/01 10742, H8/532 User's Manual
2.3 Sierra Semiconductor Data Book, August 1990
2.4 Dallas Semiconductor 1992-1993 Product Data Book
2.5 Spec. No. MG1216B1N000-002, Seiko Instruments Inc.
2.6 1-0210 BAIO-SMI Design Document.
2.7 1-0211 BAIO-SMPI Design Document.
2.8 1-0209 SMK302 Kabinet Design Document.

3. Introduction:

The SCU301 (Service Computer Unit) is a service computer module for the SPC/3 computer. The module implements various check and control functions.

The module is designed as a standard submodule to be mounted in the dedicated VSB submodule port on the BAIO module.
The SCU301 is an independent processor module that will communicate with the host BAIO and any errors in the SCU does not put the SPC/3 out of service.

The SCU is a slave VSB module as seen from the BAIO. Normally interaction between the BAIO and the SCU takes place via the dual port memory. However, some registers on the SCU module can be accessed directly by the BAIO without participation from the SCU processor.

The sensor inputs and outputs for the SCU are connected via a 220 pin connector at the rear of the SCU module.

The submodule is designed around the following main components:

Hitachi HD6475328 Single-chip microcontroller
Sierra SC11011CV and SC11044CN modem chipset
Dallas DS1486 NVRAM/RTC chip
Seiko Instruments G1216B1N000 LCD display (128x64 dots)

A description of the registers and functions of these components can be found in the documents listed in ref. 2.2, 2.3, 2.4, and 2.5.

4. Description:

Submodule in position 1 of BAIO: Base addr. = 0x08000000.

The following table shows the addresses that the SCU301 submodule will respond to when an access is made from the BAIO module.

The following addresses are all offsets from the base address:

Addr: (BAIOBIG=1)	Use:
0x00003 to 0x0007F	Read submodule ID PROM
0x00083 to 0x000FF	Read submodule Revision PROM
0x00103 and 0x00107	Submodule status register
0x00183	Submodule control register
0x00203	Interrupt to SCU processor
0x00283	Activate Powerdown Relay
0x00303	Read keyswitch settings
0x20003 to 0x2003F	Real Time Clock registers.
0x20043 to 0x9FFFF	Dual port memory, 128k bytes.

Table of all addresses used on the submodule (for BAIOBIG=1).

Addr: (BAIOBIG=0)	Use:
0x00000 to 0x0007C	Read submodule ID PROM
0x00080 to 0x000FC	Read submodule Revision PROM
0x00100 and 0x00104	Submodule status register
0x00180	Submodule control register
0x00200	Interrupt to SCU processor
0x00280	Activate Powerdown Relay
0x00300	Read keyswitch settings
0x20000 to 0x2003C	Real Time Clock registers.
0x20040 to 0x9FFFC	Dual port memory, 128k bytes.

Table of all addresses used on the submodule (for BAIOBIG=0).

All register accesses to the submodule are physically 8-bit accesses, and the data lines are connected to DA07-DA00. The BAIO must address the registers according to the chosen BAIOBIG. The SCU checks/supplies the correct parity for the addressed byte.

Communication between the BAIO and the SCU is accomplished through the dual port memory, which is used for command, handshaking, and status report, as well as for data transfer.

4.1 SMI register description:

Submodule ID register. RPID 0x000	
BIG=1; Addr:	0x003 to 0x07F, 32 byte addresses.
BIG=0; Addr:	0x000 to 0x07C, 32 byte addresses.
Access types:	Byte. Read only.
32 Bytes indicating submodule TYPE and Serial number.	
Refer to BAIO SMPI Design Document.	

Submodule revision register. RPVERS 0x080	
BIG=1; Addr:	0x083 to 0x0FF, 32 byte addresses.
BIG=0; Addr:	0x080 to 0x0FC, 32 byte addresses.
Access types:	Byte. Read only.
32 Bytes indicating submodule revision level.	
Refer to BAIO SMPI Design Document.	

STATUS register 0.		0x100
BIG=1; Addr:	0x103	
BIG=0; Addr:	0x100	
Access types:	Byte. Read/write.	
STATUS0 (07)	Interrupt req.	Active high signal indicating interrupt request from one or more sources.
STATUS0 (06)	Pending Intr.	Pending Interrupt Request.
STATUS0 (05)	Pending PERR	*Pending Parity Fault Interrupt
STATUS0 (04)	Test.	Test plug connected.
STATUS0 (03)	Type bit 3	*Parity Fault in byte 3
STATUS0 (02)	Type bit 2	*Parity Fault in byte 2
STATUS0 (01)	Type bit 1	*Parity Fault in byte 1
STATUS0 (00)	Type bit 0	*Parity Fault in byte 0

*) These bits are cleared by a write to the register address.

not used

STATUS register 1.		
BIG=1; Addr:	0x107	
BIG=0; Addr:	0x104	adress 0x380
Access types:	Byte. Read/write.	
STATUS1 (07)	IRSTAW	*INT fra Watchdog
STATUS1 (06)	IRSTAS	*INT fra SCU.
STATUS1 (05)	T.B.D	Not used.
STATUS1 (04)	T.B.D.	Not used.
STATUS1 (03)	PO Reset stat.	*High after power-on.
STATUS1 (02)	BA Reset stat.	*High after BAIO reset. by SCU
STATUS1 (01)	GL Reset stat.	*High after global reset.
STATUS1 (00)	T.B.D.	Not used.

*) These bits are cleared by a write to the register address.

CONTROL register.		0x180
BIG=1; Addr:	0x183	
BIG=0; Addr:	0x180	
Access types:	Byte. Read/write.	
CONTROL (07)	INTRENA	*Enable interrupts from SCU.
CONTROL (06)	PFINTENA	*Enable parity fault intr.
CONTROL (05)	ODD	*Odd parity.
CONTROL (04)	T.B.D.	*Not used.
CONTROL (03)	T.B.D.	*Not used.
CONTROL (02)	T.B.D.	*Not used.
CONTROL (01)	T.B.D.	*Not used.
CONTROL (00)	T.B.D.	*Not used.

*) Low on power-up or after SMI reset.

ATTENTION register.		
BIG=1; Addr:	0x203	
BIG=0; Addr:	0x200	
Access types:	Byte. Write only.	
ATTN (07-00)	Data	Data is "don't care" in this operation. The write access itself will signal attention to the SCU processor. Correct parity must be used.

Powerdown register.	erstellt als Control Reg C0008 h	
BIG=1; Addr:	0x283	
BIG=0; Addr:	0x280	
Access types:	Byte. Write only.	
PWRDOWN (07)	POWERDOWN	*Activate PowerDown relay.
PWRDOWN (06)	T.B.D.	Not used.
PWRDOWN (05)	T.B.D.	Not used.
PWRDOWN (04)	T.B.D.	Not used.
PWRDOWN (03)	T.B.D.	Not used.
PWRDOWN (02)	T.B.D.	Not used.
PWRDOWN (01)	T.B.D.	Not used.
PWRDOWN (00)	T.B.D.	Not used.

*) Low on power-up or after SMI reset.

Keyswitch register. RPKEY 0x200		
BIG=1; Addr:	0x303	
BIG=0; Addr:	0x300	
Access types:	Byte. Read only.	
KEYSTAT (07)	SYSTEM	Key in pos. SYST.
KEYSTAT (06)	START	START activated.
KEYSTAT (05)	SHUTDOWN	Shutdown activated.
KEYSTAT (04)	MODEM	Modem activated.
KEYSTAT (03)	T.B.D.	Not used.
KEYSTAT (02)	LAMP1	LAMP1 is ON.
KEYSTAT (01)	LAMP2	LAMP2 is ON.
KEYSTAT (00)	LAMP3	LAMP3 is ON.

Base address for dual port memory on submodule.	
BIG=1; Addr:	0x20003 to 0x9FFFF (128k addresses).
BIG=0; Addr:	0x20000 to 0x9FFFC (128k addresses).
Access types:	Byte. Read/write.
Lower 16 addresses are registers for the Real Time Clock. See ref. 2.4.	

4.2 SCU Processor:

The SCU uses a Hitachi H8/532 single-chip microcontroller (HD6475328F) with a 20 MHz clock crystal.

The controller has a 32kbytes internal PROM memory, which is used for internal test, flash prom download and initializing programs. The SCU application program resides in a 128k bytes flash memory external to the controller chip. Additionally the controller has a 1k bytes internal fast RAM memory and 128k bytes external RAM memory.

The internal registers are described in ref. 2.2.

The processor has one on-chip USART, an 8 channel A/D converter (10 bit resolution), an interrupt controller, and various timers/PWM generators, see ref. 2.2.

The BAIO and the SCU communicates through a 128kbytes dual port memory. The dual port memory also contains a real time clock function (lower 16 bytes of memory), thus accessible from both the BAIO and the SCU.

External registers are memory mapped.

Memory map: (Mode=4)

Address:	Use:
0x00000 - 0x07FFF	On-chip PROM (32kbytes)
0x08000 - 0x0FB7F	Not used
0x0FB80 - 0x0FF7F	On-chip RAM (when enabled) (1kbytes)
0x0FF80 - 0x0FFFF	On-chip registers *)
0x10000 - 0x2FFFF	Extern PROM (Flash EPROM, 128kbytes)
0x30000 - 0x3FFFF	Not used
0x40000 - 0x5FFFF	Extern RAM (128kbytes)
0x60000 - 0x7FFFF	Not used
0x80000 - 0x9FFFF	Dual port RAM (128kbytes, incl. RTC)
0xA0000 - 0xBFFFF	Not used
0xC0000 - 0xC0090	External registers **)
0xC0091 - 0xCFFFF	Not used
0xD0000 - 0xD003F	ID/Type registers
0xD0040 - 0xFFFFF	Not used

All writable registers are cleared on power-on.
*) See ref. 2.2
**) See description 4.6.

4.3 External registers:

Address:			Use:
0xC0000	R	2	Status Register
0xC0008	W	1	Control Register
0xC0010	R	1	Interrupt status for IR1
0xC0018	W	1	Interrupt mask for IR1
0xC0020	R/W	1	RS232 handshake
0xC0028	R/W	8	Modem Registers
0xC0030	W	1	Watchdog trig
0xC0038	R	1	PS config status
0xC0040	W	1	Temp/voltage mux set
0xC0048	R	1	Temp check from PS
0xC0050	R	1	Power check OK from PS
0xC0058	W	1	Activity LED's
0xC0060	W	1	LED control (SCU front)
0xC0068	W	1	Lamp control (keyswitch panel)
0xC0070	W	4	Fan PWM Control registers
0xC0078	R	1	Fan Tacho Error/ UPS check/ Ext.Alarm
0xC0080	W	2	Disk control
0xC0088	R/W	4	Display Control
0xC0090	W	1	Display contrast setting
0xD0000	R	32	Type/serial no. *)
0xD0020	R	32	Revision *)

*) See ref. 2.7 for description of type/revision registers.

4.4 SCU processor interrupts:

System interrupt is generated from internal timer. This timer interrupt has highest priority.

Interrupt:

NMI: Not used

IR0: Interrupt from modem

IR1: Interrupt from:
- BAIO not sc0302
- RTC1
- RTC2
- UPS Alarm not sc0302
- Ext. Alarm
- Debug from system bus
- Keyswitch intr.
- Modem enable switch not sc0302

The IR1 interrupt source can be read via the external IR1 status register and each interrupt may be masked via IR1 mask register. IR1 status reg. is cleared by reading.

4.5 Connections to SCU via SCU front panel:

- On the front panel there are two telephone RJ11 connectors. One is connected to the telephone wall plug, the other one is used as a loop-through connection for an external telephone, when the modem is not active.
  RJ11 Modem/Phone Connector:

PIN	SIGNAL
1	n.c.
2	n.c.
3	Modem1/PH1
4	Modem2/PH2
5	n.c.
6	n.c.

- One RS232 port with a (shielded) RJ45 connector:

PIN	SIGNAL	KABELPAIR	
1	Data terminal ready	pair 1	DTR, output
2	Transmit data	pair 1	TXD, output
3	Receive data	pair 2	RXD, input
4	Request to send	pair 3	RTS, output
5	Clear to send	pair 3	CTS, input
6	Data set ready	pair 2	DSR, input
7	Signal ground	pair 4	GND
8	Carrier detect	pair 4	DCD, input

- One HW Debug switch.

- One Error indicator (LED).

The HW Debug switch activates the DEBUG-S signal to the BAIO. The SCU will then receive an interrupt via DEBUG-R from BAIO. The DEBUG-S signal can also be activated by the SCU.

The Error LED is activated by power-on reset. The internal SCU test program that is started immediately after power-on is responsible for switching the error LED off, when the test is completed without errors.
The Error LED is activated by the watchdog or the SCU during error conditions.

4.6 Description of external registers:

Register: Status Register. RPKEY 0x200	
Address: 0xC0000	Access: Read Only.

Bit:	Use:
D7	Watchdog timeout.
D6	Keyswitch: SYSTEM
D5	Keyswitch: START
D4	Keyswitch: SHUTDOWN
D3	Keyswitch: MODEM
D2	Lamp3 (Modem)
D1	Lamp2 (Shutdown)
D0	Lamp1 (Start)

Register: CONTROL REGISTER. WPCONT 0x280	
Address: 0xC0008	Access: Write only.

Bit:	Use:
D7	HW Debug
D6	PSHIGH, activate high voltage setting on PS
D5	PSLOW, activate low voltage setting on PS
D4	Power-down Control, internal PS and external disks
D3	BAIO interrupt
D2	None
D1	Reset BAIO
D0	Reset Display

Register: Interrupt Status for IR1.	
Address: 0xC0010	Access: Read Only.

Bit:	Use:
D7	Interrupt from BAIO
D6	Interrupt from RTC1
D5	Interrupt from RTC2
D4	Interrupt from UPS Fault
D3	Interrupt from External Alarm
D2	Interrupt from Debug from system bus
D1	Interrupt from Keyswitch
D0	Interrupt from Modem Enable switch

Register: Interrupt Mask for IR1.	
Address: 0xC0018	Access: Write only.

Bit:	Use:
D7	Mask interrupt from BAIO
D6	Mask interrupt from RTC1
D5	Mask interrupt from RTC2
D4	Mask interrupt from UPS Fault
D3	Mask interrupt from External Alarm
D2	Mask interrupt from Debug from system bus
D1	Mask interrupt from Keyswitch
D0	Mask interrupt from Modem Enable switch

Register: RS232 handshake.	
Address: 0xC0020	Access: Read/Write

Bit:	Use:	
D7	Read: CTS	Write: RTS
D6	Read: DSR	Write: DTR
D5	Read: Low	Write: Don't care
D4	Read: Low	Write: Don't care
D3	Read: Low	Write: Don't care
D2	Read: Low	Write: Don't care
D1	Read: Low	Write: Don't care
D0	Read: DCD	Write: Don't care

Register: Modem registers. 8 registers.	
Address: 0xC0028 to 0xC002F	Access: Read/Write

Reg:	Use:
C002F	STR, Scratch Pad Register.
C002E	MSR, Modem Status Register.
C002D	LSR, Line Status Register.
C002C	MCR, Modem Control Register.
C002B	LCR, Line Control Register.
C002A	Read: Intr. ID Write: None
C0029	Read: Intr. enable/DLM Write: Intr. enable /DLM
C0028	Read: Receive Data/DLL Write: Transmit Data/DLL

See also ref. 2.3.

Register: Watchdog trig register.	
Address: 0xC0030	Access: Write only.

Bit:	Use:
D7	Data is "don't care" in this
D6	operation. The write access
D5	itself will signal a restart
D4	to the watchdog circuitry.
D3	The software must access this
D2	address with a regular frequency
D1	to avoid reset of the processor.
D0	The watchdog is held reset by power-on reset.

Watchdog timeout: 200 mS.

Register: PS Config Status Register. RPCONF 0x300	
Address: 0xC0038	Access: Read Only.

Bit:	Use:
D7	PS unmounted signal, from Power Supply in slot no. 7
D6	PS unmounted signal, from Power Supply in slot no. 6
D5	PS unmounted signal, from Power Supply in slot no. 5
D4	PS unmounted signal, from Power Supply in slot no. 4
D3	PS unmounted signal, from Power Supply in slot no. 3
D2	PS unmounted signal, from Power Supply in slot no. 2
D1	PS unmounted signal, from Power Supply in slot no. 1
D0	PS unmounted signal, from Power Supply in slot no. 0

Register: Temp/Voltage MUX control.	
Address: 0xC0040	Access: Write only.

Bit:	Use:
D7	None
D6	None
D5	None
D4	None
D3	MUX3
D2	MUX2
D1	MUX1
D0	MUX0

Register: Temp. Check Register (Temp check signal from PS)	
Address: 0xC0048	Access: Read Only. RPTEMP 0x380

Bit:	Use:
D7	Temperature Check OK from Power Supply in slot no. 7
D6	Temperature Check OK from Power Supply in slot no. 6
D5	Temperature Check OK from Power Supply in slot no. 5
D4	Temperature Check OK from Power Supply in slot no. 4
D3	Temperature Check OK from Power Supply in slot no. 3
D2	Temperature Check OK from Power Supply in slot no. 2
D1	Temperature Check OK from Power Supply in slot no. 1
D0	Temperature Check OK from Power Supply in slot no. 0

Register: Power Check Register (from Power Supplies).	
Address: 0xC0050	Access: Read Only. RPCHECK 0x400

Bit:	Use:
D7	Power Check OK from Power Supply in slot no. 7
D6	Power Check OK from Power Supply in slot no. 6
D5	Power Check OK from Power Supply in slot no. 5
D4	Power Check OK from Power Supply in slot no. 4
D3	Power Check OK from Power Supply in slot no. 3
D2	Power Check OK from Power Supply in slot no. 2
D1	Power Check OK from Power Supply in slot no. 1
D0	Power Check OK from Power Supply in slot no. 0

Register: Activity LED CONTROL REGISTER.	
Address: 0xC0058	Access: Write only.

Bit:	Use:
D7	POWER LED ON
D6	None
D5	None
D4	ACT (AIO ON)
D3	AC3
D2	AC2
D1	AC1
D0	AC0

Register: LED control Register.	
Address: 0xC0060	Access: Write only. WPLAMP 0x480

Bit:	Use:
D7	None
D6	None
D5	None
D4	None
D3	None
D2	None
D1	Error LED control (Green)
D0	Error LED control (Red)

Register: LAMP control Register (Keyswitch lamps) WPLAMP 0x4B0	
Address: 0xC0068	Access: Write only.

Bit:	Use:
D7	None LEDG
D6	None LEDR
D5	None
D4	None ~~LEDG~~
D3	None
D2	Lamp3 control (Modem)
D1	Lamp2 control (Shutdown)
D0	Lamp1 control (Start)

Register: Fan PWM Control Register 0 to 3.	
Address: 0xC0070 to 0xC0073	Access: Write only.
Bit:	**Use:**
D7	One byte written to this address instructs the
D6	PWM counter to generate a PWM signal with a
D5	duty cycle corresponding to the hex value of
D4	the byte. 01H corresponds to a duty cycle of 1/256.
D3	FEH corresponds to a duty cycle of 254/256.
D2	00H or FFH generates a constantly low or high signal.
D1	This will make the FAN controller generate an error
D0	and fully activate the fans.

Register: Fan Tacho Err and UPS/Alarm Register.	
Address: 0xC0078	Access: Read Only.
Bit:	**Use:**
D7	UPS Off Line.
D6	UPS not connected.
D5	External Alarm.
D4	Alarm not connected.
D3	Tacho Error from Fan Control Unit 3
D2	Tacho Error from Fan Control Unit 2
D1	Tacho Error from Fan Control Unit 1
D0	Tacho Error from Fan Control Unit 0

Register: Disk Control Register 0.	
Address: 0xC0080	Access: Write only.
Bit:	**Use:**
D7	None
D6	SCSI channel bit 2.
D5	SCSI channel bit 1
D4	SCSI channel bit 0.
D3	SCSI ID bit 3
D2	SCSI ID bit 2
D1	SCSI ID bit 1
D0	SCSI ID bit 0

Register: Disk Control Register 1.		XPDISK1 0x580
Address: 0xC0081	Access: Write only.	

Bit:	Use:
D7	None
D6	None
D5	LED address bit 1
D4	LED address bit 0
D3	Strobe
D2	None
D1	None
D0	LED data

Register: Display control registers. 4 registers.	
Address: 0xC0088 to 0xC008B	Access: Read/Write

Reg:	Use:	
0xC008B	Read: Instr.Read Chip2	Write: Instr.Write Chip2
0xC008A	Read: Data Read Chip2	Write: Data Write Chip2
0xC0089	Read: Instr.Read Chip1	Write: Instr.Write Chip1
0xC0088	Read: Data Read Chip1	Write: Data Write Chip1

Instruction format for Chip1/Chip2: See ref. 2.5.

Register: Display Contrast Setting.	
Address: 0xC0090	Access: Write only.

Bit:	Use:
D7	Select
D6	Up/down
D5	None
D4	None
D3	None
D2	None
D1	None
D0	Strobe

5. SCU Backpanel Connections:

5.1 Connections from SCU to Activity LED's:

Pin:	Module:	Signal:
Tba	Activity	Power LED ON
Tba	Activity	GND
Tba	Activity	Act. LED no.1 ON
Tba	Activity	GND
Tba	Activity	Act3
Tba	Activity	Act2
Tba	Activity	Act1
Tba	Activity	Act0
Tba	Activity	GND
Tba	Activity	GND
Tba	Activity	+12V from SCBP
Tba	Activity	+12V from SCBP

5.2 Connections from SCU to Keyswitch:

Pin:	Module:	Signal:
Tba	Keyswitch	SYSTEM
Tba	Keyswitch	GND
Tba	Keyswitch	START
Tba	Keyswitch	GND
Tba	Keyswitch	SHUTDOWN
Tba	Keyswitch	MODEM
Tba	Keyswitch	GND
Tba	Keyswitch	LAMP1
Tba	Keyswitch	LAMP2
Tba	Keyswitch	LAMP3
Tba	Keyswitch	GND
Tba	Keyswitch	+12V from SCBP
Tba	Keyswitch	+12V from SCBP
Tba	Keyswitch	+12V from SCBP

5.3 Connections from SCU to Display:

Pin:	Module:	Signal:	
Tba	Display	Vdd	+5V til display
Tba	Display	Vss	GND
Tba	Display	Vlc	Contrast reg.
Tba	Display	DB0	
Tba	Display	DB1	
Tba	Display	DB2	
Tba	Display	DB3	
Tba	Display	DB4	
Tba	Display	DB5	
Tba	Display	DB6	
Tba	Display	DB7	
Tba	Display	CS1	Select1, active low
Tba	Display	CS2	Select2, active low
Tba	Display	RST	Reset, active low
Tba	Display	R/W	Read/Write
Tba	Display	D/I	Data/Instr. Register select
Tba	Display	E	Strobe, active high
Tba	Display	FGND	GND
Tba	Display	LEDA	LED Anode, +5V via resistor
Tba	Display	LEDC	LED Cathode, GND

20

5.4 Connections from SCU to Disk Backplane:

Pin:	Module:	Signal:	
Tba	DBP	IDCH2	SCSI Channel ID, bit 2
Tba	DBP	IDCH1	SCSI Channel ID, bit 1
Tba	DBP	IDCH0	SCSI Channel ID, bit 0
Tba	DBP	GND	
Tba	DBP	IDAD3	SCSI Address, bit 3
Tba	DBP	IDAD2	SCSI Address, bit 2
Tba	DBP	IDAD1	SCSI Address, bit 1
Tba	DBP	IDAD0	SCSI Address, bit 0
Tba	DBP	GND	
Tba	DBP	LADR1	LED Address, bit 1
Tba	DBP	LADR0	LED Address, bit 0
Tba	DBP	GND	
Tba	DBP	LEDDA	LED Data
Tba	DBP	GND	
Tba	DBP	LSTRB	Strobe
Tba	DBP	GND	

16

5.5 Connections from SCU to temperature sensors, 8 blocks:

Pin:	Module:	Signal:	
Tba	SENSOR 1	+5V	
Tba	SENSOR 2	Vtemp	Signal from sensor
Tba	SENSOR 3	GND	
Tba	SENSOR 4	Shield	

5.6 Connections from SCU to Fan Control Board, 4 blocks:

Pin:	Module:	Signal:	
Tba	FANCTRL	+12V	from SCBP
Tba	FANCTRL	FANSPEED	PWM control signal from SCU
Tba	FANCTRL	GND	
Tba	FANCTRL	FERR	Error signal from controller
Tba	FANCTRL	GND	
Tba	FANCTRL	+12V	from SCBP
Tba	FANCTRL	+12V	from SCBP

5.7 Connections from SCU to UPS/Alarm:

Pin:	Module:	Signal:	
Tba	UPS	UPSSG	UPS status signal (input)
Tba	UPS	UPSCNN	UPS not connected.
Tba	UPS	UPSGND	UPS GND
Tba	UPS	DSKOFF	Disk cabinet control output
Tba	UPS	DSKGND	Disk cabinet GND
Tba	UPS	ALSG	External Alarm input
Tba	UPS	ALCNN	External Alarm not connected
Tba	UPS	ALGND	External Alarm GND

5.8 Connections from SCU to Power Backplane:

Pin:	Module:	Signal:
Tba	PDB	Power fail signal from PS 8
Tba	PDB	GND from PS 8
Tba	PDB	Power fail signal from PS 7
Tba	PDB	GND from PS 7
Tba	PDB	Power fail signal from PS 6
Tba	PDB	GND from PS 6
Tba	PDB	Power fail signal from PS 5
Tba	PDB	GND from PS 5
Tba	PDB	Power fail signal from PS 4
Tba	PDB	GND from PS 4
Tba	PDB	Power fail signal from PS 3
Tba	PDB	GND from PS 3
Tba	PDB	Power fail signal from PS 2
Tba	PDB	GND from PS 2
Tba	PDB	Power fail signal from PS 1
Tba	PDB	GND from PS 1
Tba	PDB	Power fail signal from PS 0
Tba	PDB	GND from PS 0
Tba	PDB	Temp OK signal from PS 8
Tba	PDB	GND from PS 8
Tba	PDB	Temp OK signal from PS 7
Tba	PDB	GND from PS 7
Tba	PDB	Temp OK signal from PS 6
Tba	PDB	GND from PS 6
Tba	PDB	Temp OK signal from PS 5
Tba	PDB	GND from PS 5
Tba	PDB	Temp OK signal from PS 4
Tba	PDB	GND from PS 4
Tba	PDB	Temp OK signal from PS 3
Tba	PDB	GND from PS 3
Tba	PDB	Temp OK signal from PS 2
Tba	PDB	GND from PS 2
Tba	PDB	Temp OK signal from PS 1
Tba	PDB	GND from PS 1
Tba	PDB	Temp OK signal from PS 0
Tba	PDB	GND from PS 0
Tba	PDB	PS7 unmounted
Tba	PDB	PS6 unmounted
Tba	PDB	PS5 unmounted
Tba	PDB	PS4 unmounted
Tba	PDB	PS3 unmounted
Tba	PDB	PS2 unmounted
Tba	PDB	PS1 unmounted
Tba	PDB	PS0 unmounted
Tba	PDB	GND
Tba	PDB	PSLOW
Tba	PDB	GND
Tba	PDB	PSHIGH

5.9 Connections from SCU to Bus Backplane:

Pin:	Module:	Signal:
Tba	BPL	Check +12V (sense +12V)
Tba	BPL	Check +12V (return +12V)
Tba	BPL	GND
Tba	BPL	Check +5V (sense +5V)
Tba	BPL	Check +5V (return +5V)
Tba	BPL	GND
Tba	BPL	Check +3.3V (sense +3.3V)
Tba	BPL	Check +3.3V (return +3.3V)
Tba	BPL	GND
Tba	BPL	Check +2.1V (sense +2.1V right)
Tba	BPL	Check +2.1V (return +2.1V right)
Tba	BPL	GND
Tba	BPL	Check +2.1V (sense +2.1V left)
Tba	BPL	Check +2.1V (return +2.1V left)

14

5.10 Connections from SCU to SCBPL:

Pin:	Module:	Signal:	
Tba	SCBPL	SCBPD	Activate power down relay
Tba	SCBPL	GND	GND
Tba	SCBPL	GND	GND
Tba	SCBPL	GND	GND
Tba	SCBPL	+12V	+12V
Tba	SCBPL	+12V	+12V
Tba	SCBPL	+12V	+12V
Tba	SCBPL	Spare	
Tba	SCBPL	Spare	
Tba	SCBPL	Spare	

10

6. Connections to BAIO, Submodule Interface:

See ref. 2.6

```
"external controller for servicecomputer 3 submodule for baio
"kan 26/10-1994

module scu200;

declarations;

"counter and register for fan control reduced to 6 bits.

"register reduced to 5 bits. Bit zero not used.

"control signals for latch to display added 25/10-1994. fitted
"name changed from cntscux. 1/2-1995

scu200 device 'mach210a';

x, z, c = .X., .Z., .C.;


"#########################################################################
"                        input signals
"#########################################################################


ad19            pin 14  ;"address bit 19
                        ;"when 0: access to DIV
                        ;"when 1: access to TPM

ad11            pin 15  ;"address bit 11
ad10            pin 16  ;"address bit 10
ad9             pin 17  ;"address bit  9
ad8             pin 20  ;"address bit  8
ad7             pin 18  ;"address bit  7. these bits ared decoded to give
                        ;"these bits are  decoded to give various
                        ;"chip select.

dat5            pin 8   ;"data bit 5 input
dat4            pin 11  ;"data bit 4 input
dat3            pin 19  ;"data bit 3 input
dat2            pin 10  ;"data bit 2 input
dat1            pin 32  ;"data bit 1 input
dat0            pin 7   ;"data bit 0 input

"input signals from SCU module.

!rp             pin 21  ;"read pulse

!wp             pin 9   ;"write pulse

clk             pin 35  ;"clock input pin connected to 20 MHz
clks            pin 13  ;"clock input pin connected to slow clock

!resi           pin 33  ;"reset input


"#########################################################################
```

```
"                          nodes
"############################################################################

ra5     node 18 istype'reg_t,buffer';"most significant bit of register A
ra4     node 60 istype'reg_t,buffer';
ra3     node 59 istype'reg_t,buffer';
ra2     node 74 istype'reg_t,buffer';
ra1     node 73 istype'reg_t,buffer';

dummy   node 67 istype'com';"to make dat0 an input

rb5     node 64 istype'reg_t,buffer';"most significant bit of register B
rb4     node 63 istype'reg_t,buffer';
rb3     node 51 istype'reg_t,buffer';
rb2     node 50 istype'reg_t,buffer';
rb1     node 49 istype'reg_t,buffer';

cn5     node 69 istype'reg_t,buffer';"most significant bit of counter
cn4     node 71 istype'reg_t,buffer';
cn3     node 72 istype'reg_t,buffer';
cn2     node 75 istype'reg_t,buffer';
cn1     node 65 istype'reg_t,buffer';
cn0     node 66 istype'reg_t,buffer';"least significant bit of counter
nul     node 70 istype'com'  ;"detects all zeros in counter

alla    node 58 istype'com'  ;"detects all ones in register A
toma    node 57 istype'com'  ;"detects all zeros in register A
ca      node 54 istype'com'  ;"compares cn(5:0) to ra(5:0)

allb    node 56 istype'com'  ;"detects all ones in register B
tomb    node 55 istype'com'  ;"detects all zeros in register B
cb      node 53 istype'com'  ;"compares cn(5:0) to rb(5:0)

csra    node 61 istype'com'  ;"chip select to register A
csrb    node 62 istype'com'  ;"chip select to register B

csdisp  node 52 istype'com,buffer'  ;"chip select to display


"############################################################################
"                          output pins
"############################################################################

pulsea  pin 42  istype'reg_t,invert';"fan control pulse for register A
pulseb  pin 43  istype'reg_t,invert';"fan control pulse for register B

csadc0  pin 24  istype'com,invert'  ;"chip select to A/D-converter 0.
csadc1  pin 25  istype'com,invert'  ;"chip select to A/D-converter 1.
csadc   pin 26  istype'com,invert'  ;"node to state machine

rpalarm pin 31  istype'com,invert'  ;"read pulse to alarm register.
wpcont  pin 27  istype'com,invert'  ;"write pulse to contrast control.

csdisp1 pin 29  istype'com,invert'  ;"chip select 1 to display.
csdisp2 pin 28  istype'com,invert'  ;"chip select 2 to display.
```

```
dispinst pin 30 istype'com,buffer'  ;"instruction/data to display.

strobe   pin 4  istype'reg_t,buffer';"strobe pulse to display.

dispre   pin 6  istype'com,invert'  ;"output enable read data from disp
                                    ;"read command to display.
dispwr   pin 5  istype'com,invert'  ;"output enable write data to disp
ledispd  pin 2  istype'reg_t,buffer';"latch enable read data from disp

wait     pin 3  istype'reg_t,buffer';"wait signal to smiscu3

"###################################################################
"             equations for registers and counter
"###################################################################
equations;

ra5.c    = clk;
ra4.c    = clk;
ra3.c    = clk;
ra2.c    = clk;
ra1.c    = clk;

rb5.c    = clk;
rb4.c    = clk;
rb3.c    = clk;
rb2.c    = clk;
rb1.c    = clk;

cn5.c    = clks;
cn4.c    = clks;
cn3.c    = clks;
cn2.c    = clks;
cn1.c    = clks;
cn0.c    = clks;

csra     = wp & !ad19 &  ad11 & !ad10 & !ad9 & !ad8 & !ad7;"address 0x800

ra5.t    = csra & ( dat5 & !ra5.q # !dat5 &  ra5.q);
ra4.t    = csra & ( dat4 & !ra4.q # !dat4 &  ra4.q);
ra3.t    = csra & ( dat3 & !ra3.q # !dat3 &  ra3.q);
ra2.t    = csra & ( dat2 & !ra2.q # !dat2 &  ra2.q);
ra1.t    = csra & ( dat1 & !ra1.q # !dat1 &  ra1.q);

dummy    = dat0;


csrb     = wp & !ad19 &  ad11 & !ad10 & !ad9 & !ad8 &  ad7;"address 0x880

rb5.t    = csrb & ( dat5 & !rb5.q # !dat5 &  rb5.q);
rb4.t    = csrb & ( dat4 & !rb4.q # !dat4 &  rb4.q);
rb3.t    = csrb & ( dat3 & !rb3.q # !dat3 &  rb3.q);
rb2.t    = csrb & ( dat2 & !rb2.q # !dat2 &  rb2.q);
rb1.t    = csrb & ( dat1 & !rb1.q # !dat1 &  rb1.q);

nul      =        !cn5.q &!cn4.q &!cn3.q &!cn2.q &!cn1.q &!cn0.q;
cn5.t    =                 cn4.q & cn3.q & cn2.q & cn1.q & cn0.q;
```

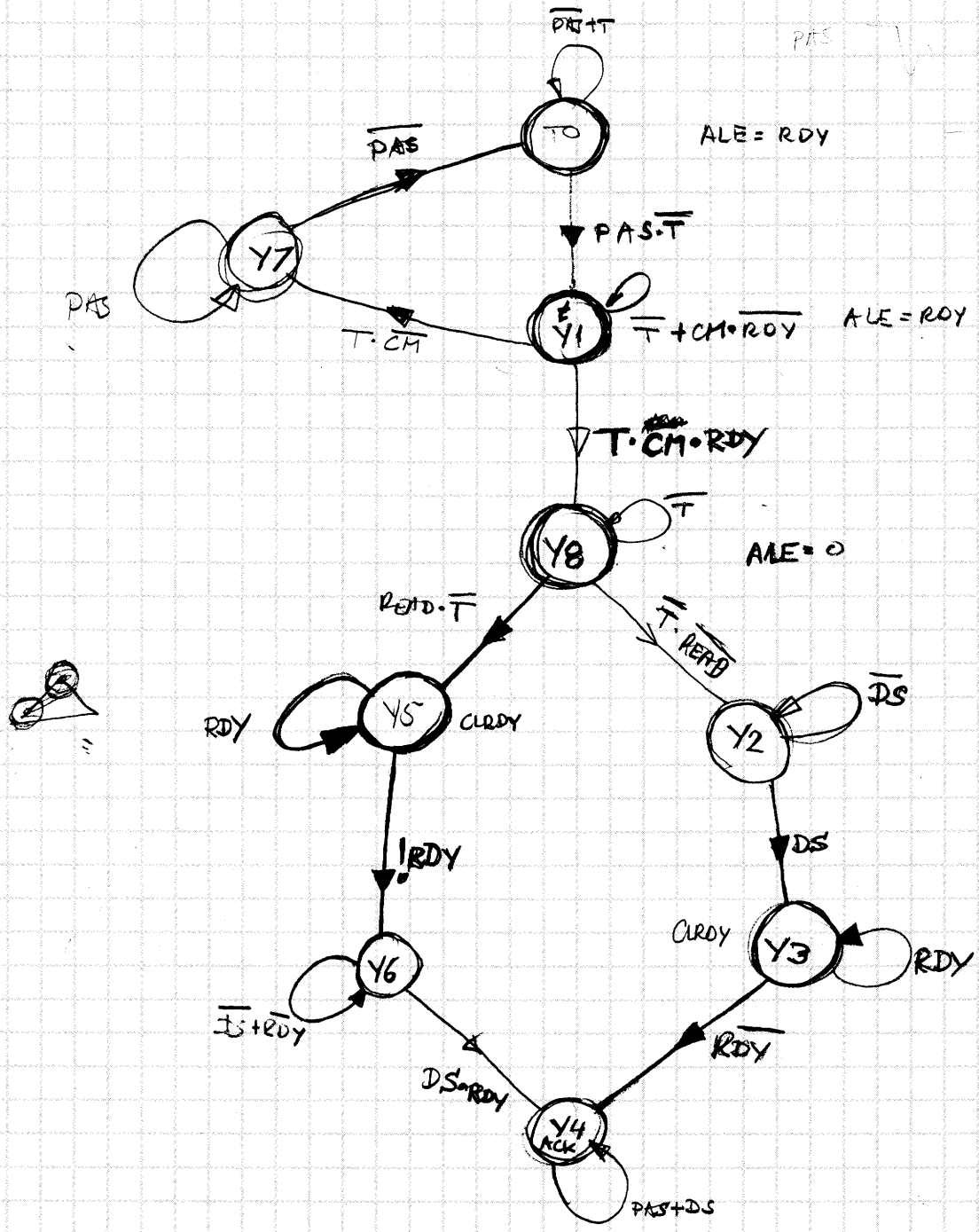
```
cn4.t   =                              cn3.q & cn2.q & cn1.q & cn0.q;
cn3.t   =                                      cn2.q & cn1.q & cn0.q;
cn2.t   =                                              cn1.q & cn0.q;
cn1.t   =                                                      cn0.q;
cn0.t   =                                                           1;

"#########################################################################
"                    equations for comparators
"#########################################################################
"The comparators compares the content of the registers with the content of
"the counter. The output from the comparator is 1, when the content og the
"content of the register is greater than the content of the counter.

"A comparator is implemented as a iterative circuit consisting of four
" elements. An element has five inputs: two bits from the register, two
"bits from the counter, an the output from the more significant comparator
"element. The register content: 'all ones' is a special case. In this case
"the output from the comparator must be a steady one.

alla =                  ra5.q & ra4.q & ra3.q & ra2.q & ra1.q;
toma =                 !ra5.q &!ra4.q &!ra3.q &!ra2.q &!ra1.q;

ca    = !ra5.q &  cn5.q #  ra5.q & !cn5.q
      # !ra4.q &  cn4.q #  ra4.q & !cn4.q
      # !ra3.q &  cn3.q #  ra3.q & !cn3.q
      # !ra2.q &  cn2.q #  ra2.q & !cn2.q
      # !ra1.q &  cn1.q #  ra1.q & !cn1.q;

pulsea.t = !pulsea.q & nul & !toma                  "clear condition
         #  pulsea.q & !ca & !alla; "set condition

pulsea.c = clks;

allb =                  rb5.q & rb4.q & rb3.q & rb2.q & rb1.q;
tomb =                 !rb5.q &!rb4.q &!rb3.q &!rb2.q &!rb1.q;


cb    = !rb5.q &  cn5.q #  rb5.q & !cn5.q
      # !rb4.q &  cn4.q #  rb4.q & !cn4.q
      # !rb3.q &  cn3.q #  rb3.q & !cn3.q
      # !rb2.q &  cn2.q #  rb2.q & !cn2.q
      # !rb1.q &  cn1.q #  rb1.q & !cn1.q;


pulseb.t = !pulseb.q & nul & !tomb                  "clear condition
         #  pulseb.q & !cb & !allb;                 "set condition

pulseb.c = clks;

"#########################################################################
"                          other outputs
"#########################################################################

"chip select to AD converter

!csadc0  =  ( rp # wp ) &
```

```
                !ad19 & !ad11 &  ad10 &  ad9 &  ad8 & !ad7;"address 0x700

!csadc1  =  ( rp # wp ) &
                !ad19 & !ad11 &  ad10 &  ad9 &  ad8 &  ad7;"address 0x780

!csadc   =  ( rp # wp ) &
                !ad19 & !ad11 &  ad10 &  ad9 &  ad8       ;"0x700 + 0x780

"read pulse to Alarm register

!rpalarm = rp & !ad19 &  ad11 & !ad10 & !ad9 &  ad8 & !ad7;"address 0x900

"write pulse to contrast control

!wpcont  = wp & !ad19 &  ad11 & !ad10 & !ad9 &  ad8 &  ad7;"address 0x980

"chip-selects to display

!csdisp1 = (wp # rp) &
                !ad19 &  ad11 & !ad10 &  ad9 & !ad8       ;"address 0xa00
                                                           " and    0xa80
!csdisp2 = (wp # rp) &
                !ad19 &  ad11 & !ad10 &  ad9 &  ad8       ;"address 0xb00
                                                           " and    0xb80

csdisp   = !csdisp1 # !csdisp2;                            "node to state m
ac
                                                           ;"address 0xb00
                                                           " and    0xb80
dispinst = (wp # rp) &
                !ad19 &  ad11 & !ad10 &  ad9 &  ad7       ;"

"instruction data command to display


!dispre  = rp & (!csdisp1 # !csdisp2); "output enable read data from
                                       "display. Inverted: read to disp.
!dispwr  = wp & (!csdisp1 # !csdisp2); "oe write data to display

"#########################################################################
"                        state-machine wait generation
"#########################################################################
declarations

n3      node 47 istype'reg_t,buffer';
n2      node 46 istype'reg_t,buffer';
n1      node 45 istype'reg_t,buffer';
n0      node 48 istype'reg_t,buffer';

t0  =   ^b1111;
w14 =   ^b1110;
w13 =   ^b1101;
w12 =   ^b1100;
w11 =   ^b1011;
w10 =   ^b1010;
w9  =   ^b1001;
w8  =   ^b1000;
```

```
w7  =   ^b0111;
w6  =   ^b0110;
w5  =   ^b0101;
w4  =   ^b0100;
w3  =   ^b0011;
w2  =   ^b0010;
w1  =   ^b0001;
w0  =   ^b0000;

ina = [rp, wp,!csadc,csdisp,!wpcont];

TILST = [n3, n2, n1, n0];

state_diagram TILST;

state t0:

wait.t = !wait.q & (!csadc # csdisp);
ledispd.t = !ledispd.q & !wait.q & csdisp;

case(ina == [x, x, 0, 0, 0]) : t0;
    (ina == [x, x, 1, x, x]) : w1;
    (ina == [x, x, x, 1, x]) : w14;
    (ina == [x, x, x, x, 1]) : w0;
endcase;
state w14: goto w13;
state w13: goto w12;

state w12:
strobe.t = !strobe.q & csdisp;
goto w11;

state w11: goto w10;
state w10: goto w9 ;
state w9 : goto w8 ;
state w8 : goto w7 ;
state w7 : goto w6 ;
state w6 : goto w5 ;
state w5 : goto w4 ;
state w4 : goto w3 ;
state w3 : goto w2 ;

state w2:
ledispd.t = ledispd.q;
goto w1;

state w1:
wait.t = wait.q;
strobe.t =  strobe.q;
goto w0;

state w0:  goto t0;


equations;
n3.c = clk;
```

```
n2.c = clk;
n1.c = clk;
n0.c = clk;
wait.c = clk;
strobe.c = clk;
ledispd.c =clk;

"###############################################################################
####
"                              reset equations
"###############################################################################
####

n3.ar       = resi;
n2.ar       = resi;
n1.ar       = resi;
n0.ar       = resi;
wait.ar     = resi;
strobe.ar   = resi;

ra5.ar      = resi;
ra4.ar      = resi;
ra3.ar      = resi;
ra2.ar      = resi;
ra1.ar      = resi;

rb5.ar      = resi;
rb4.ar      = resi;
rb3.ar      = resi;
rb2.ar      = resi;
rb1.ar      = resi;

cn5.ar      = resi;
cn4.ar      = resi;
cn3.ar      = resi;
cn2.ar      = resi;
cn1.ar      = resi;
cn0.ar      = resi;

pulsea.ar   = resi;
pulseb.ar   = resi;
ledispd.ar  = resi;

declarations;

ADR = [ad19, ad11, ad10, ad9, ad8, ad7];

DAT = [dat5, dat4, dat3, dat2, dat1];

RA  = [ra5, ra4, ra3, ra2, ra1];

RB  = [rb5, rb4, rb3, rb2, rb1];

CN  = [cn5, cn4, cn3, cn2, cn1, cn0];

@include'cntst1.abl'; "DAT=^h00
@include'cntst2.abl'; "DAT=^h01
@include'cntst3.abl'; "DAT=^h02
```

```
@include'cntst4.abl'; "DAT=^h04
@include'cntst5.abl'; "DAT=^h08
@include'cntst6.abl'; "DAT=^h10
@include'cntst7.abl'; "DAT=^h1f
@include'cntstx8.abl';"test of read and write pulses
@include'cntstx9.abl';"test of wait and strobe

end scu200;
```



$T \cdot \overline{CM} + T \cdot CM \cdot RDY$

$T(\overline{CM} + CM \cdot RDY$

$T \cdot (\overline{CM} + RDY)$

$\overline{T} + CM \cdot \overline{RDY}$

```
"controller for servicecomputer 3 submodule for baio
"kan  9/9-1994

module scu100;

"wait input signal added.       d.9/5-1994
"state Y8 added between Y1 and Y2/Y5     d. 9/9-1994
"name changed from smiscu4 d. 1/2-1995

declarations;

scu100 device 'mach220a';

x, z, c = .X., .Z., .C.;


"#########################################################################
"                        input signals
"#########################################################################

"input signals from vsb

!pas            pin 14  ;"physical address strobe from vsb
 read           pin 20  ;"read signal from vsb. Latched with address
!ds             pin 51  ;"data strobe from vsb

ad27            pin 22  ;"address bit 27
ad26            pin  9  ;"address bit 26
ad25            pin 11  ;"address bit 25

"There is an access to SCU when ad(27:25)= 100

ad19            pin  7  ;"address bit 19
                        ;"when 0: access to DIV
                        ;"when 1: access to TPM

ad11            pin 24  ;"address bit 11
ad10            pin 16  ;"address bit 10
ad9             pin 49  ;"address bit  9
ad8             pin 50  ;"address bit  8
ad7             pin 17  ;"address bit  7. these bits ared decoded to give
                        ;"these bits are  decoded to give various
                        ;"chip select.

"input signals from SCU module.

T               pin 23  ;"input signal from delay line.
                         "the delay line is driven by the output signa
                         "t from this PAL.

clk             pin 15  ;"clock input pin connected to 30 MHz

!resi           pin 54  ;"reset input

t               pin 43  ;"output signal to delay line;
```

```
test            pin 12  ;"test input signal. Is read in status bit 4

wait            pin 37  ;"wait input

"#########################################################################
"                         nodes
"#########################################################################

y8              node    ;"state variable
y7              node    ;"state variable
y6              node    ;"state variable
y5              node    ;"state variable
y4              node    ;"state variable
y3              node    ;"state variable
y2              node    ;"state variable
y1              node    ;"state variable

cm              node    ;"match signal. ad(27:25)= 100

rdy             node    ;"internal combined ready signal
rdyi            node    istype'reg_t,buffer';"internal ready signal
rdys            node    istype'reg_t,buffer';"rdy synchronized
reqi            node    istype'com,buffer';
led             node    istype'reg_t,buffer';

n2              node    istype'reg_d,buffer';
n1              node    istype'reg_d,buffer';
n0              node    istype'reg_d,buffer';

"the state variables defines the following states:

"T0 = !y8&!y7&!y6&!y5&!y4&!y3&!y2&!y1;
"Y1 = !y8&!y7&!y6&!y5&!y4&!y3&!y2& y1;      address handshake
                        address handshake
"Y18=  y8&!y7&!y6&!y5&!y4&!y3&!y2& y1;
"Y8 =  y8&!y7&!y6&!y5&!y4&!y3&!y2&!y1;
"Y82=  y8&!y7&!y6&!y5&!y4&!y3& y2&!y1;
"Y2 = !y8&!y7&!y6&!y5&!y4&!y3& y2&!y1;      write cycle wait for data
"Y23= !y8&!y7&!y6&!y5&!y4& y3& y2&!y1;
"Y3 = !y8&!y7&!y6&!y5&!y4& y3&!y2&!y1;      start write cycle; set request
"Y34= !y8&!y7&!y6&!y5& y4& y3&!y2&!y1;
"Y4 = !y8&!y7&!y6&!y5& y4&!y3&!y2&!y1;      data handshake to SMI
"Y85=  y8&!y7&!y6& y5&!y4&!y3&!y2&!y1;
"Y5 = !y8&!y7&!y6& y5&!y4&!y3&!y2&!y1;      start read cycle; set request
"Y56= !y8&!y7& y6& y5&!y4&!y3&!y2&!y1;
"Y6 = !y8&!y7& y6&!y5&!y4&!y3&!y2&!y1;      wait for read data
"Y17= !y8& y7& y6&!y5&!y4&!y3&!y2& y1;
"Y7 = !y8& y7&!y6&!y5&!y4&!y3&!y2&!y1;      no match; wait for end of cycl
e

resn    node    istype'reg_t,buffer';   "programmed reset.
                                        "control register bit 0

inten   node    istype'reg_t,buffer';   "interrupt enable
                                        "control register bit 1

cnt2    node    istype'reg_t,buffer';   "control register bit 2
cnt3    node    istype'reg_t,buffer';   "control register bit 3
```

```
cnt4    node    istype'reg_t,buffer';   "control register bit 4

actpm   node    istype'com,buffer';     "access to TPM
csc     node    istype'com,buffer';     "select control register
css     node    istype'com,buffer';     "select status register
oed     node    istype'com,buffer';     "output enable to PAL
clrdy   node    istype'com,buffer';     "clear rdyi

rpx     node    istype'com,buffer';     "used to generate read pulses
wpx     node    istype'com,buffer';     "used to generate write pulses

"#########################################################################
"                          output signals
"#########################################################################

"output signals to SMI;

ac      pin 36 istype'com,invert';      "address complete
asack   pin 10 istype'com,buffer';      "address acknowledge
ack     pin  6 istype'com,buffer';      "data handshake.
int     pin 59 istype'com,buffer';      "interrupt request


"output signals registers.

lrd     pin 64  istype'reg_d,buffer';   "latch read data
                                        "generated from sync part
rp      pin 67  istype'reg_d,invert';   "read pulse sync part
wp      pin 66  istype'reg_d,invert';   "write pulse sync part
csm     pin 41  istype'invert';         "chip select to memory

"data i/o pins for control and status to SMI

dat7    pin 45;         "data bit 7
dat6    pin  5;         "data bit 6
dat5    pin 46;         "data bit 5
dat4    pin 47;         "data bit 4
dat3    pin  4;         "data bit 3
dat2    pin  3;         "data bit 2
dat1    pin  2;         "data bit 1
dat0    pin 48;         "data bit 0

ale     pin 60  istype'com,buffer'; "address latch enable. High signal
                                    "opens address latch.

enrd    pin 63  istype'invert';     "output enable read data to SMI
                                    "generated from async part
enwd    pin 65  istype'invert';     "output enable write data from SMI
                                    "generated from sync part
lwd     pin 62  istype'com,buffer'; "latch write data.          async part

"#########################################################################
"                          pins for parity check/generation
"#########################################################################

oddi    pin 13;         "input from parity generator 74x280
```

```
par     pin 44;          "parity bit for data. I/O pin

oddp    pin 21  istype'reg_t,buffer'; "use odd parity;

flten   node    istype'reg_t,buffer'; "enable parity fault interrupt

"error in databit( 7: 0)
err0    node    istype'reg_t,buffer';

        "once set the error flags is kept until they are
        "until cleared by a write cycle to the status register

"#########################################################################
"                          pins read and write pulses
"#########################################################################

rpid    pin 55  istype'reg_d,invert';"read pulse to ID-PROM
rpvers  pin 56  istype'reg_d,invert';"read pulse to VERS-PROM
rpkey   pin 29  istype'reg_d,invert';"read pulse to KEY register
wpcont  pin 30  istype'reg_d,invert';"write pulse to PS control
rpconf  pin 28  istype'reg_d,invert';"read pulse to PS config register
rptemp  pin 57  istype'reg_d,invert';"read pulse to PS temp register
rpcheck pin 58  istype'reg_d,invert';"read pulse to PS check register
wplamp  pin 33  istype'reg_d,invert';"write pulse to lamp register
wpdisk0 pin 31  istype'reg_d,invert';"write pulse to disk control register
0
wpdisk1 pin 32  istype'reg_d,invert';"write pulse to disk control register
1
wpal0   pin 26  istype'reg_d,invert';"write pulse to activity register 0
wpal1   pin 25  istype'reg_d,invert';"write pulse to activity register 1

"#########################################################################
"                          equations for state variables
"#########################################################################

equations;

y1  = !resi & (
          !y8&!y7&!y6&!y5&!y4&!y3&!y2     & pas & !T "(T0+Y1) & pas & !T
    #     !y8&!y7&!y6&!y5&!y4&!y3&!y2& y1);          "Y1

y2  = !resi & (
           y8&!y7&!y6&!y5&!y4&!y3    &!y1 & !read & !T "(Y8+Y82)
    #         !y7&!y6&!y5&!y4&!y3& y2&!y1 & !read & !T "(Y82+Y2)
    #     !y8&!y7&!y6&!y5&!y4&!y3& y2&!y1);            "Y2

y3  = !resi & (
          !y8&!y7&!y6&!y5&!y4    & y2&!y1 & ds        "(Y2+Y23)
    #     !y8&!y7&!y6&!y5&!y4& y3    &!y1 & ds        "(Y23+Y3)
    #     !y8&!y7&!y6&!y5&!y4& y3&!y2&!y1);           "Y3

y4  = !resi & (
          !y8&!y7&!y6&!y5    & y3&!y2&!y1 & !rdy      "(Y3+Y34)
    #     !y8&!y7&!y6&!y5& y4    &!y2&!y1 & !rdy & pas"(Y34+Y4)
    #     !y8&!y7&!y6&!y5& y4    &!y2&!y1 & !rdy & ds "(Y34+Y4)
    #     !y8&!y7& y6&!y5    &!y3&!y2&!y1 &  rdy & ds "(Y6+Y64)
    #     !y8&!y7&    !y5& y4&!y3&!y2&!y1 &  rdy & ds "(Y64+Y4)
```

```
    #     !y8&!y7&!y6&!y5& y4&!y3&!y2&!y1 & pas       "Y4
    #     !y8&!y7&!y6&!y5& y4&!y3&!y2&!y1 & ds );     "Y4

y5  = !resi & (
           y8&!y7&!y6    &!y4&!y3&!y2&!y1 & read & !T"(Y8+Y85)
    #         !y7&!y6& y5&!y4&!y3&!y2&!y1 & read & !T"(Y85+Y5)
    #     !y8&!y7&!y6& y5&!y4&!y3&!y2&!y1);           "Y5

y6  = !resi & (
          !y8&!y7    & y5&!y4&!y3&!y2&!y1 & !rdy      "(Y5+Y56)
    #     !y8&!y7& y6    &!y4&!y3&!y2&!y1 & !rdy      "(Y56+Y6)
    #     !y8&!y7& y6&!y5&!y4&!y3&!y2&!y1);           "Y6

y7  = !resi & (
          !y8    &!y6&!y5&!y4&!y3&!y2& y1 & T & !cm     "(Y1+Y17)
    #     !y8& y7&!y6&!y5&!y4&!y3&!y2& pas & T & !cm    "(Y17+Y7)
    #     !y8& y7&!y6&!y5&!y4&!y3&!y2&!y1 & pas);       "Y7

y8  = !resi & (
             !y7&!y6&!y5&!y4&!y3&!y2& y1 & T & cm & rdy    "(Y1+Y18)
    #      y8&!y7&!y6&!y5&!y4&!y3&!y2     & T & cm & rdy    "(Y18+Y8)
    #      y8&!y7&!y6&!y5&!y4&!y3&!y2&!y1);                 "Y8

"#########################################################################
"                        equations for node variables
"#########################################################################

rdy = rdyi.q;                  "rdyi is an internal ready signal

cm  = ad27 & !ad26 & !ad25;

"#########################################################################
"                        equations for output to vsb
"#########################################################################

ack =   y4 & ds;                                    "Y4

asack = !y1 & y8 # y2 # y3 # y5 # y6 # y4 & pas;

!ac =   y7 & pas # y2 # y8 # y3 # y5 # y6 # y4 & pas;

"#########################################################################
"                    equations for output to SCU module
"#########################################################################

"output to delay line

t =     y1;

ale = !asack & rdy;     "latch open when high. Address latch is closed
                        "when asack is activated or rdy deactivated.

lwd  =  y2 # y3;   "latch open when high

!enrd   = ds & read & (y6 # y4);        "oe read data to SMI
```

```
reqi   =  (y3 # y5);"internal request

"#########################################################################
"       equations  for internal registers
"#########################################################################

actpm   =  ad19;                                        "access to NVM
csc     = !ad19 & !ad11 & !ad10 & !ad9 &  ad8 &  ad7; "control register
css     = !ad19 & !ad11 & !ad10 & !ad9 &  ad8 & !ad7; "status register

oed     = !ad19 & !ad11 & !ad10 & !ad9 &  ad8 &  ad7  "control register
        # !ad19 & !ad11 & !ad10 & !ad9 &  ad8 & !ad7; "status register

dat7.oe = oed & rp.q;
dat6.oe = oed & rp.q;
dat5.oe = oed & rp.q;
dat4.oe = oed & rp.q;
dat3.oe = oed & rp.q;
dat2.oe = oed & rp.q;
dat1.oe = oed & rp.q;
dat0.oe = oed & rp.q;


int  =  inten.q & 0              "no interrupt sources on scu302
     # flten.q & err0.q;

dat7 =  css   & int
     #  csc   & inten.q;

dat6 =  css   & 0
     #  csc   & flten.q;

dat5 =  css   & err0.q
     #  csc   & oddp.q;

dat4 =  css   & test
     #  csc   & cnt4.q;

dat3 =  css   & 0
     #  csc   & cnt3.q;

dat2 =  css   & 0
     #  csc   & cnt2.q;

dat1 =  css   & 0
     #  csc   & led.q;

dat0 =  css   & err0.q
     #  csc   & resn.q;

"#########################################################################
"                        equations  for other outputs
"#########################################################################
```

```
"#####################################################################
"                          equations  for parity check/generation
"#####################################################################

err0.c = clk;   "the error flag is clocked in all read an write cycles

"parity bits are output in all read cycles to the module

par.oe = rp.q;              "all read cycles
                            bør fjernes a.h.t. data hold time.
par    = oddi & rp.q;       "The parity bit from the parity generator is
                            "output during all read cycles.

err0.t = !err0.q & (oddi $ par.pin) & wp.q      "check parity
       #  err0.q & css & wp.q;                  "clear error flag

"#####################################################################
"                          equations  for control register
"#####################################################################
resn.c  = clk;
led.c   = clk;
cnt2.c  = clk;
cnt3.c  = clk;
cnt4.c  = clk;
oddp.c  = clk;
flten.c = clk;
inten.c = clk;

resn.t  = csc & wp.q & ( dat0.pin & !resn.q  # !dat0.pin & resn.q );

led.t   = csc & wp.q & ( dat1.pin & !led.q   # !dat1.pin & led.q  );

cnt2.t  = csc & wp.q & ( dat2.pin & !cnt2.q  # !dat2.pin & cnt2.q );

cnt3.t  = csc & wp.q & ( dat3.pin & !cnt3.q  # !dat3.pin & cnt3.q );

cnt4.t  = csc & wp.q & ( dat4.pin & !cnt4.q  # !dat4.pin & cnt4.q );

oddp.t  = csc & wp.q & ( dat5.pin & !oddp.q  # !dat5.pin & oddp.q );

flten.t = csc & wp.q & ( dat6.pin & !flten.q # !dat6.pin & flten.q);

inten.t = csc & wp.q & ( dat7.pin & !inten.q # !dat7.pin & inten.q);

resn.ar  = resi;
led.ar   = resi;
cnt2.ar  = resi;
cnt3.ar  = resi;
cnt4.ar  = resi;
oddp.ar  = resi;
flten.ar = resi;
inten.ar = resi;
wpal0.ar = resi;
wpal1.ar = resi;
```

```
"#########################################################################
"                        synchronous part
"#########################################################################

n2.ar = resi; "state variable
n1.ar = resi; "state variable
n0.ar = resi; "state variable

rdys.ap   = resi; "synchronized version of rdy
rdyi.ap   = resi; "rdyi is set by resi.

n2.c = clk; "state variable
n1.c = clk; "state variable
n0.c = clk; "state variable

rdyi.c   = clk;          "internal ready signal
rdys.c   = clk;          "synchronized version og rdy

clrdy  =  reqi;          "rdyi is cleared in all internal cycles
                         "and in read cycles to TPM

rdyi.ar  = clrdy;        "rdyi is cleared by clrdy.

wp.c   = clk;            "write pulse
rp.c   = clk;            "read pulse
lrd.c  = clk;            "latch read data. Latch open when 1.

rdys.t = rdys.q & !rdy; "clear rdys

declarations;

s0      = ^b000; "idle state;
s1      = ^b001;
s2      = ^b011;
s3      = ^b010;
s4      = ^b110;
s5      = ^b100;
s6      = ^b101;
s7      = ^b000;

ina  = [rdys.q, actpm, read, wait];

state_diagram([n2,n1,n0]);

state s0:
"idle state

!enwd= !read & !rdys.q;
rp.d =  read & !rdys.q;
wp.d = !read & !rdys.q;

rpx  =  read & !rdys.q;
wpx  = !read & !rdys.q;
!csm  =  ad19 & !rdys.q;                    "chip select to NVM
```

```
case"          ys ex r  w
    (ina == [ 1, x, x, x]): s0;
    (ina == [ 0, x, x, x]): s1;
endcase;


state s1:

!enwd= !read;
rp.d =  read;
wp.d = !read;

rpx  =  read;
wpx  = !read;
!csm  =  ad19;                                  "chip select to NVM

goto s2;

state s2:

!enwd= !read;
rp.d =  read;
wp.d = !read;
lrd.d = read;
rpx  =  read;
wpx  = !read;
!csm  =  ad19;                                  "chip select to NVM

goto s3;

state s3:

!enwd= !read;
!csm  =  ad19;                                  "chip select to NVM

rp.d =  read & wait;
wp.d = !read & wait;
lrd.d = read & wait;
rpx  =  read & wait;
wpx  = !read & wait;

case "        ys ex  r  w
    (ina == [ x, x, x, 1]): s3;
    (ina == [ x, x, x, 0]): s4;
endcase;

state s4:

!enwd= !read;
!csm  =  ad19;                                  "chip select to NVM
rdyi.t = !rdyi.q; "set rdyi
rdys.t = !rdys.q; "set rdys

goto s0;

"#####################################################################
```

```
"                  equations for read and write pulses
"#################################################################
equations;

rpid.c          = clk;"read pulse to ID-PROM
rpvers.c        = clk;"read pulse to VERS-PROM
rpkey.c         = clk;"read pulse to KEY register
wpcont.c        = clk;"write pulse to PS control
rpconf.c        = clk;"read pulse to PS config register
rptemp.c        = clk;"read pulse to PS temp register
rpcheck.c       = clk;"read pulse to PS check register
wplamp.c        = clk;"write pulse to lamp register
wpdisk0.c       = clk;"write pulse to disk control register 0
wpdisk1.c       = clk;"write pulse to disk control register 1
wpal1.c         = clk;"write pulse to activity register 1
wpal0.c         = clk;"write pulse to activity register 0


rpid.d          = rpx & !ad19 & !ad11 & !ad10 & !ad9 & !ad8 & !ad7;
rpvers.d        = rpx & !ad19 & !ad11 & !ad10 & !ad9 & !ad8 &  ad7;
rpkey.d         = rpx & !ad19 & !ad11 & !ad10 &  ad9 & !ad8 & !ad7;
wpcont.d        = wpx & !ad19 & !ad11 & !ad10 &  ad9 & !ad8 &  ad7;
rpconf.d        = rpx & !ad19 & !ad11 & !ad10 &  ad9 &  ad8 & !ad7;
rptemp.d        = rpx & !ad19 & !ad11 & !ad10 &  ad9 &  ad8 &  ad7;
rpcheck.d       = rpx & !ad19 & !ad11 &  ad10 & !ad9 & !ad8 & !ad7;
wplamp.d        = wpx & !ad19 & !ad11 &  ad10 & !ad9 & !ad8 &  ad7;
wpdisk0.d       = wpx & !ad19 & !ad11 &  ad10 & !ad9 &  ad8 & !ad7;
wpdisk1.d       = wpx & !ad19 & !ad11 &  ad10 & !ad9 &  ad8 &  ad7;

wpal0.d         = wpx & !ad19 & !ad11 &  ad10 &  ad9 & !ad8 & !ad7;
wpal1.d         = wpx & !ad19 & !ad11 &  ad10 &  ad9 & !ad8 &  ad7;

declarations;

MA = [ad27, ad26, ad25];

AD =  [ad19, ad11, ad10, ad9, ad8, ad7];

DAT = [dat7, dat6, dat5, dat4, dat3, dat2, dat1, dat0];

TIL   = [y8, y7, y6, y5, y4, y3, y2, y1];

T0 = ^b00000000;
Y1 = ^b00000001;
Y2 = ^b00000010;
Y3 = ^b00000100;
Y4 = ^b00001000;
Y5 = ^b00010000;
Y6 = ^b00100000;
Y7 = ^b01000000;
Y8 = ^b10000000;
CNT = [inten, flten, oddp, cnt4, cnt3, cnt2, led, resn];

PULS = [!rpid,!rpvers,!rpkey,!rpconf,!rptemp,!rpcheck,
        !wpcont,!wplamp,!wpdisk0,!wpdisk1,!wpal0,!wpal1];
```

```
@const tst = ^h0;
@include 'scutst1.abl';"test vectors for no access
@const tst = ^h1;
@include 'scutst1.abl';"test vectors for no access
@const tst = ^h2;
@include 'scutst1.abl';"test vectors for no access

@const p   = ^b0;
@const tst = ^h00;
@include 'scutst2.abl';"test vectors for control register value 0
@const tst = ^h01;
@include 'scutst2.abl';"test vectors for control register bit 0
@const tst = ^h02;
@include 'scutst2.abl';"test vectors for control register bit 1
@const tst = ^h04;
@include 'scutst2.abl';"test vectors for control register bit 2
@const tst = ^h08;
@include 'scutst2.abl';"test vectors for control register bit 3
@const tst = ^h10;
@include 'scutst2.abl';"test vectors for control register bit 4
@const p   = ^b1;
@const tst = ^h20;
@include 'scutst2.abl';"test vectors for control register bit 5
@const p   = ^b0;
@const tst = ^h40;
@include 'scutst2.abl';"test vectors for control register bit 6
@const tst = ^h80;
@include 'scutst2.abl';"test vectors for control register bit 7

"testvectors for generation of write pulses
@const tsta = ^h05;
@const tstr = ^b000000100000; "wpcont
@include 'scutst3.abl';"test vectors generation of write pulse
@const tsta = ^h09;
@const tstr = ^b000000010000; "wplamp
@include 'scutst3.abl';"test vectors generation of write pulse
@const tsta = ^h0a;
@const tstr = ^b000000001000; "wpdisk0
@include 'scutst3.abl';"test vectors generation of write pulse
@const tsta = ^h0b;
@const tstr = ^b000000000100; "wpdisk1
@include 'scutst3.abl';"test vectors generation of write pulse
@const tsta = ^h0c;
@const tstr = ^b000000000010; "wpal0
@include 'scutst3.abl';"test vectors generation of write pulse
@const tsta = ^h0d;
@const tstr = ^b000000000001; "wpal1
@include 'scutst3.abl';"test vectors generation of write pulse

@const tsta = ^h00;
@const tstr = ^b100000000000; "rpid
@include 'scutst4.abl';"test vectors generation of read pulse
@const tsta = ^h01;
@const tstr = ^b010000000000; "rpvers
@include 'scutst4.abl';"test vectors generation of read pulse
@const tsta = ^h04;
```

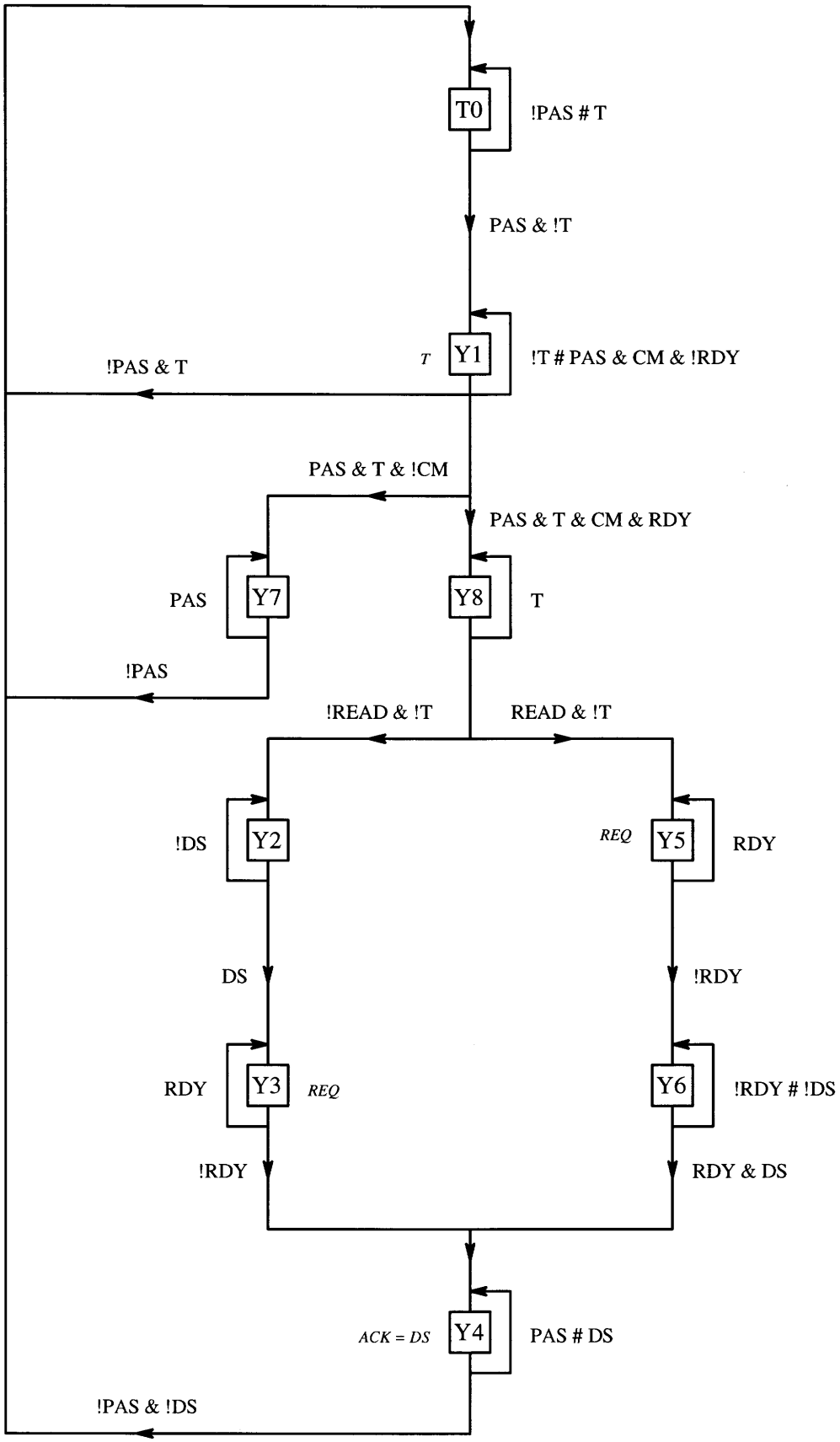
```
@const tstr = ^b001000000000; "rpkey
@include 'scutst4.abl';"test vectors generation of read pulse
@const tsta = ^h06;
@const tstr = ^b000100000000; "rpconf
@include 'scutst4.abl';"test vectors generation of read pulse
@const tsta = ^h07;
@const tstr = ^b000010000000; "rptemp
@include 'scutst4.abl';"test vectors generation of read pulse
@const tsta = ^h08;
@const tstr = ^b000001000000; "rpid
@include 'scutst4.abl';"test vectors generation of read pulse

"test status register and interrupt

"clear control register
@const tsta = ^h03; "address
@const tst  = ^h00; "data
@const i    = ^b0;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstw.abl';"write

"clear status register
@const tsta = ^h02; "address
@const tst  = ^h00; "data
@const i    = ^b0;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstw.abl';"write

"clear control register, generate parity error
@const tsta = ^h03; "address
@const tst  = ^h00; "data
@const i    = ^b0;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b1;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstw.abl';"write

"read  status register
@const tsta = ^h02; "address
@const tst  = ^h21; "data
@const i    = ^b0;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstr.abl';"read

"enable interrupt, control register write
@const tsta = ^h03; "address
@const tst  = ^h40; "data
@const i    = ^b1;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b1;  "parity input
```

```
@const o    = ^b1;  "input from parity checker
@include 'scutstw.abl';"write

"read  status register
@const tsta = ^h02; "address
@const tst  = ^ha1; "data
@const i    = ^b1;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstr.abl';"read

"clear status register
@const tsta = ^h02; "address
@const tst  = ^h00; "data
@const i    = ^b0;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstw.abl';"write

"read  status register
@const tsta = ^h02; "address
@const tst  = ^h00; "data
@const i    = ^b0;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstr.abl';"read

"enable interrupt, control register write, generate parity error
@const tsta = ^h03; "address
@const tst  = ^h40; "data
@const i    = ^b1;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b1;  "input from parity checker
@include 'scutstw.abl';"write

"read  status register
@const tsta = ^h02; "address
@const tst  = ^hb1; "data
@const i    = ^b1;  "interrupt
@const e    = ^b1;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstr.abl';"read

"disable interrupt, control register write.
@const tsta = ^h03; "address
@const tst  = ^h00; "data
@const i    = ^b0;  "interrupt
@const e    = ^b0;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstw.abl';"write
```

```
"read  status register
@const tsta = ^h02; "address
@const tst  = ^h31; "data
@const i    = ^b0;  "interrupt
@const e    = ^b1;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstr.abl';"read

"clear status register. Status register write
@const tsta = ^h02; "address
@const tst  = ^hff; "data
@const i    = ^b0;  "interrupt
@const e    = ^b1;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstw.abl';"read

"read  status register
@const tsta = ^h02; "address
@const tst  = ^h10; "data
@const i    = ^b0;  "interrupt
@const e    = ^b1;  "test input pin
@const p    = ^b0;  "parity input
@const o    = ^b0;  "input from parity checker
@include 'scutstr.abl';"read
@include 'scutstm.abl';"test access to memory


end scu100;
```

SCU101	KAN	Page 1
	960415	SCU302

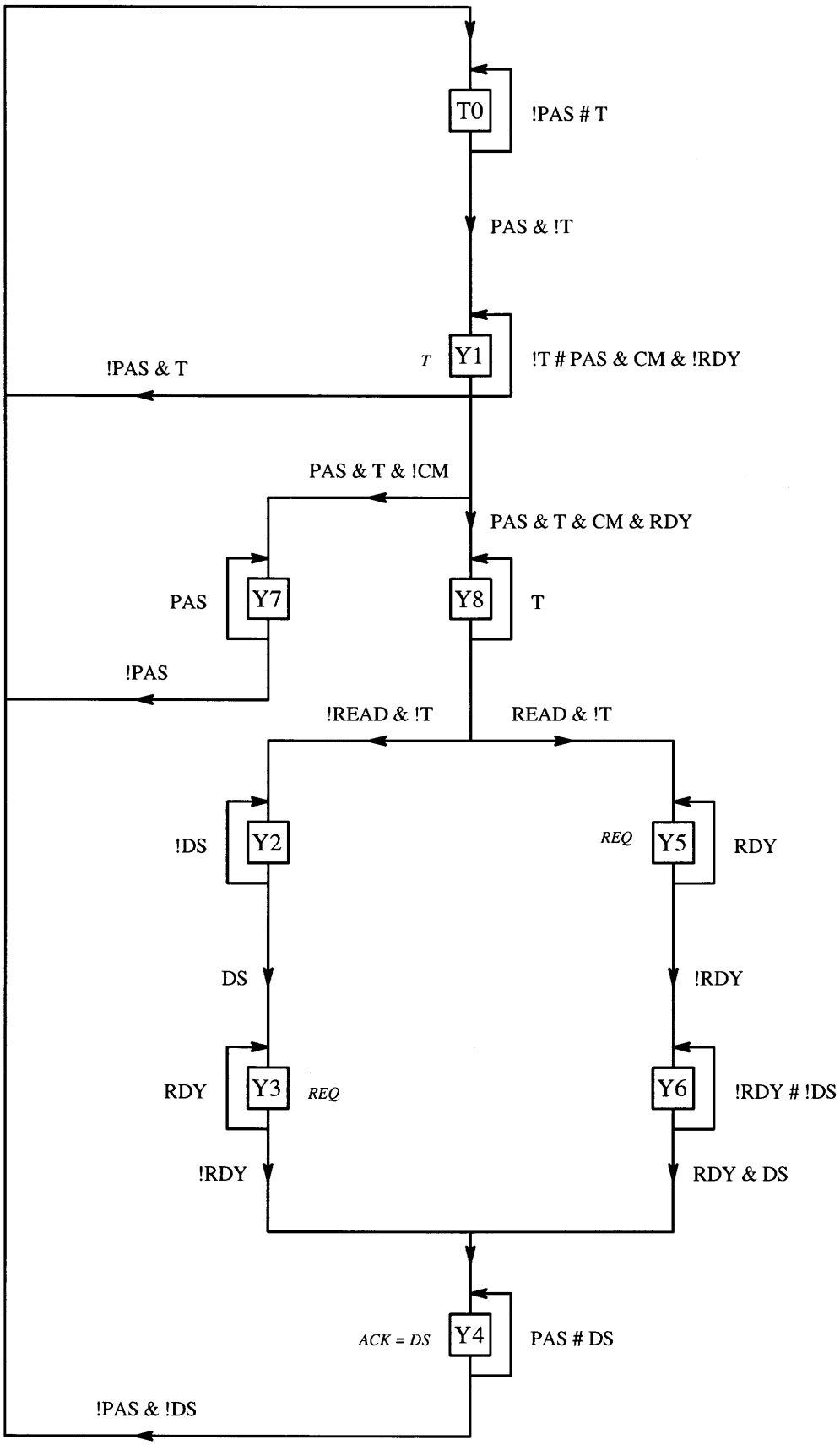
Asynchronous State Machine

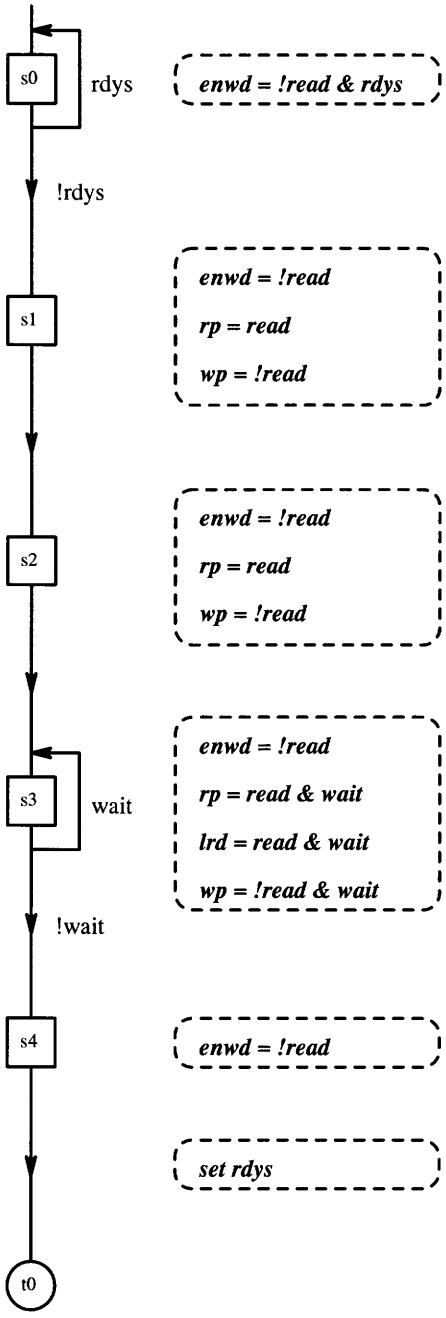


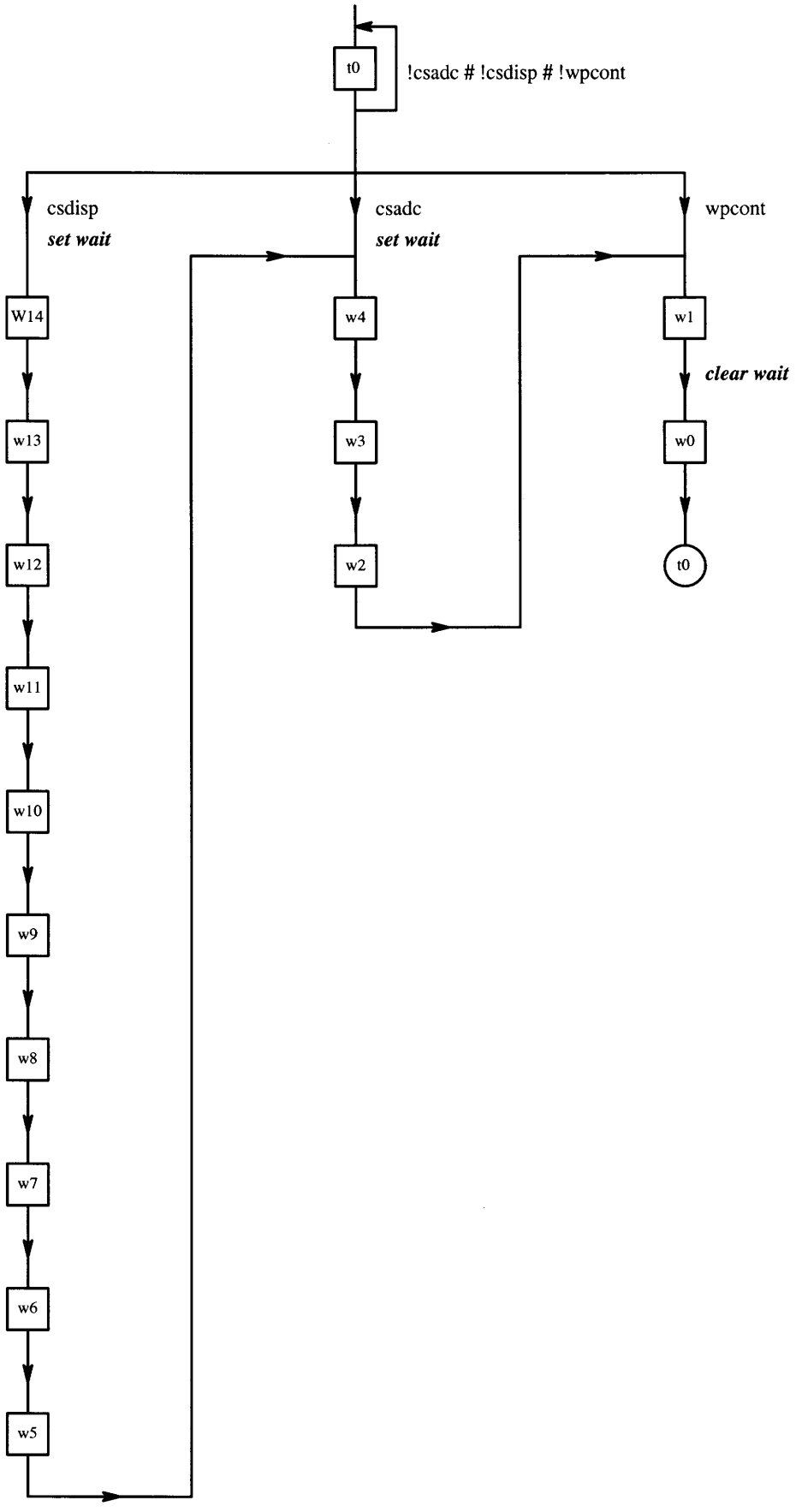
Dansk Data Elektronik A/S

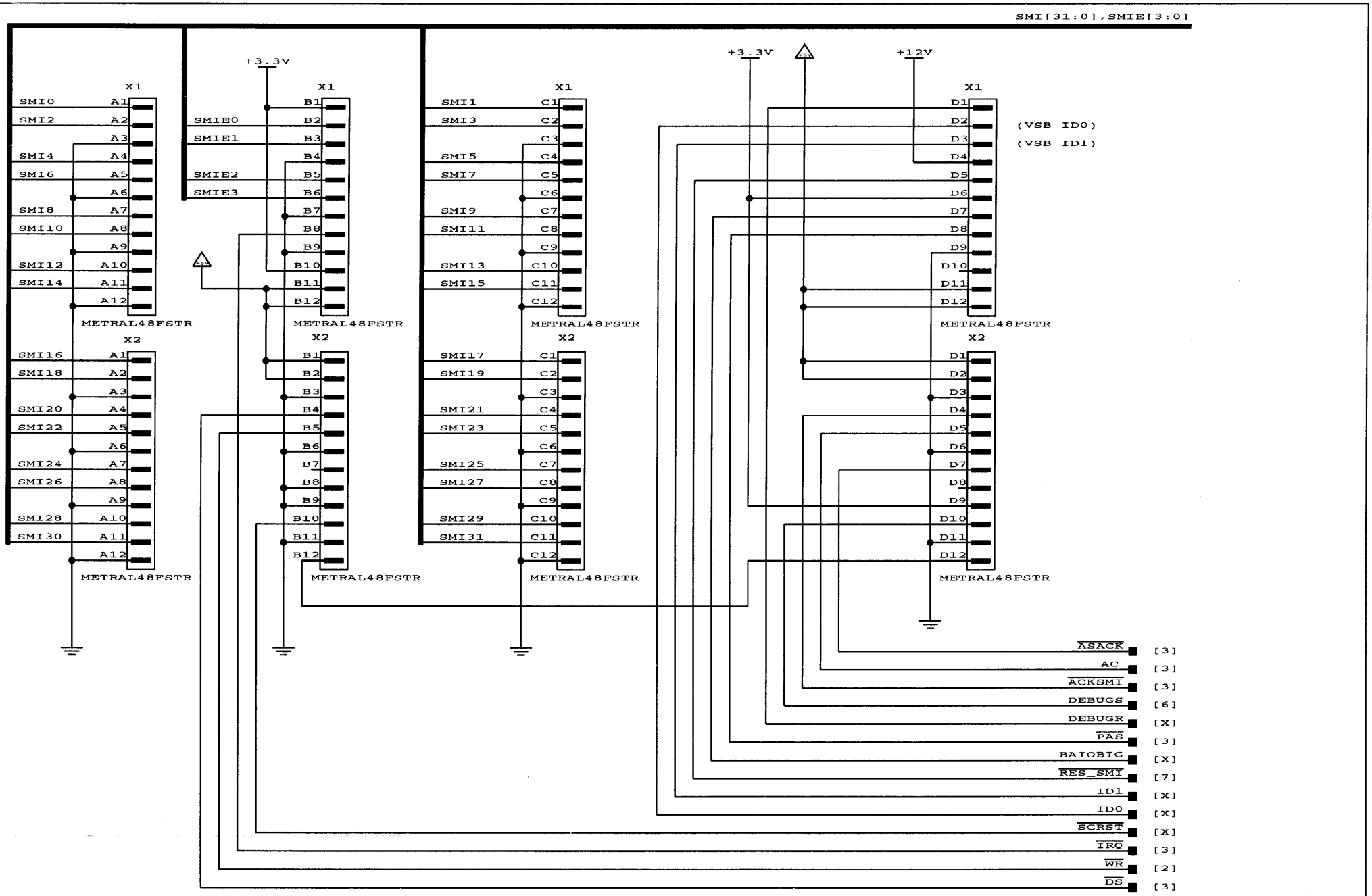
SCU101	KAN	Page 1
	960415	SCU302

Asynchronous State Machine

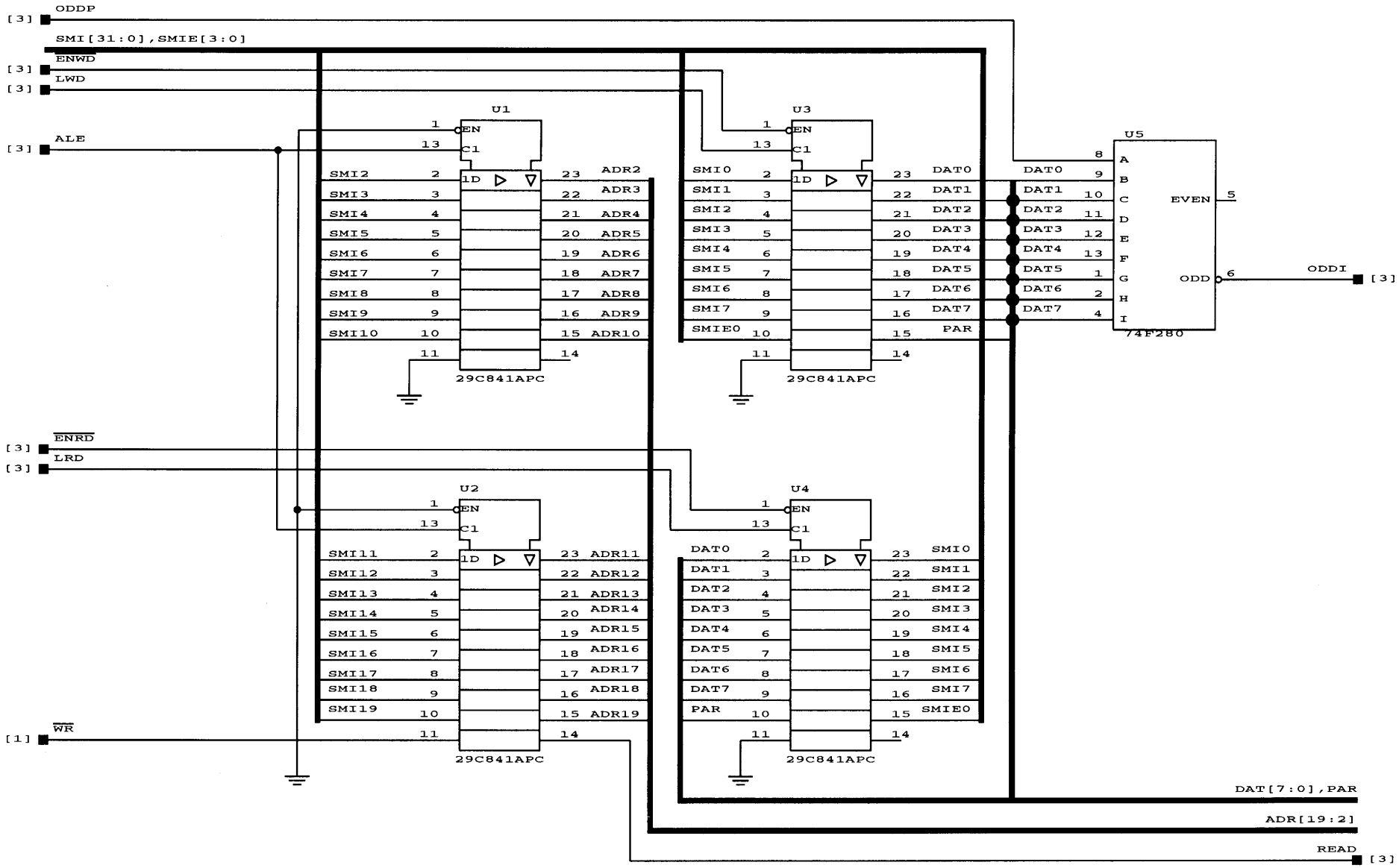



s0
rdys
enwd = !read & rdys
!rdys
s1
enwd = !read
rp = read
wp = !read
s2
enwd = !read
rp = read
wp = !read
s3
wait
enwd = !read
rp = read & wait
lrd = read & wait
wp = !read & wait
!wait
s4
enwd = !read
set rdys
t0

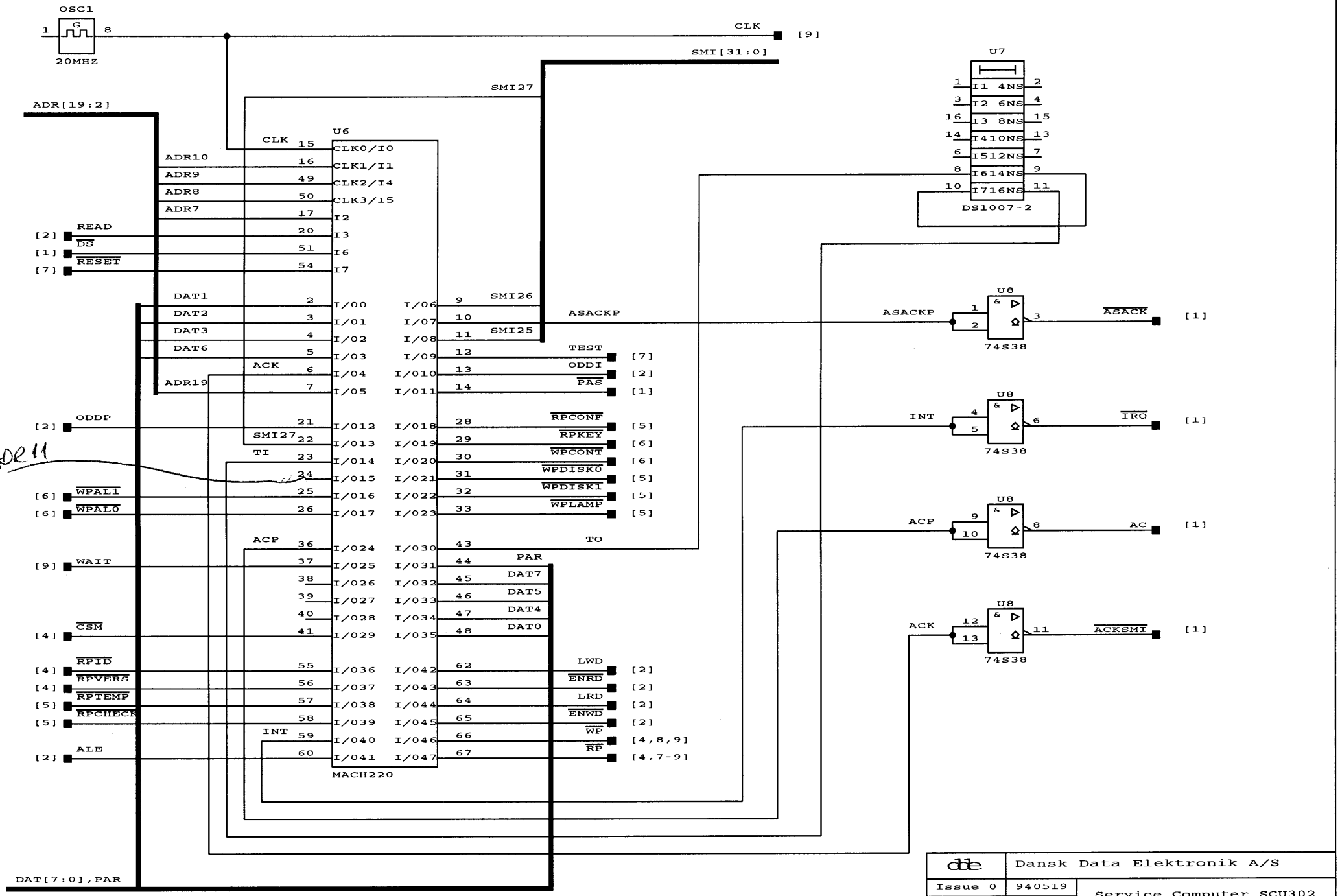

t0
!csadc # !csdisp # !wpcont
csdisp
set wait
csadc
set wait
wpcont
W14
w4
w1
clear wait
w13
w3
w0
w12
w2
t0
w11
w10
w9
w8
w7
w6
w5



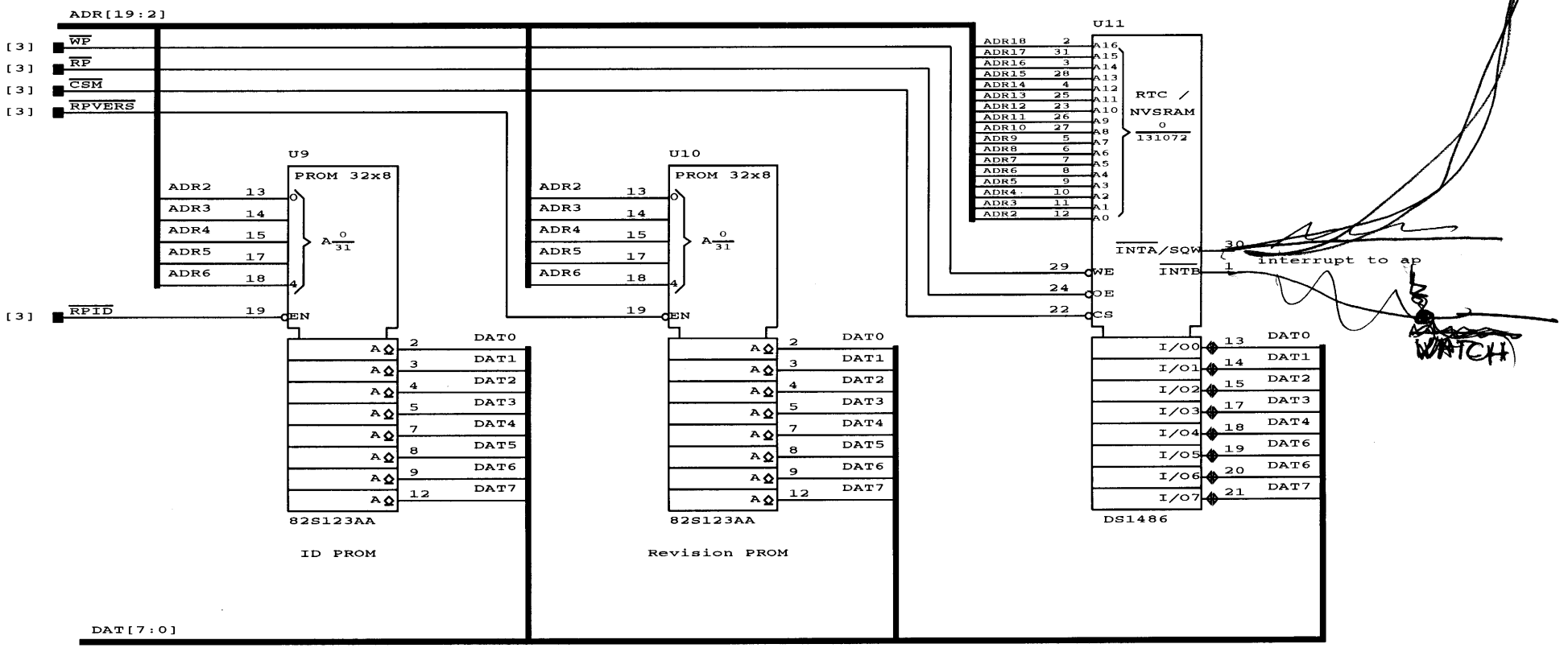
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 SMI interface	
Issue 1			
Issue 2			
Issue 3		File: scu	Page: 1 of 12



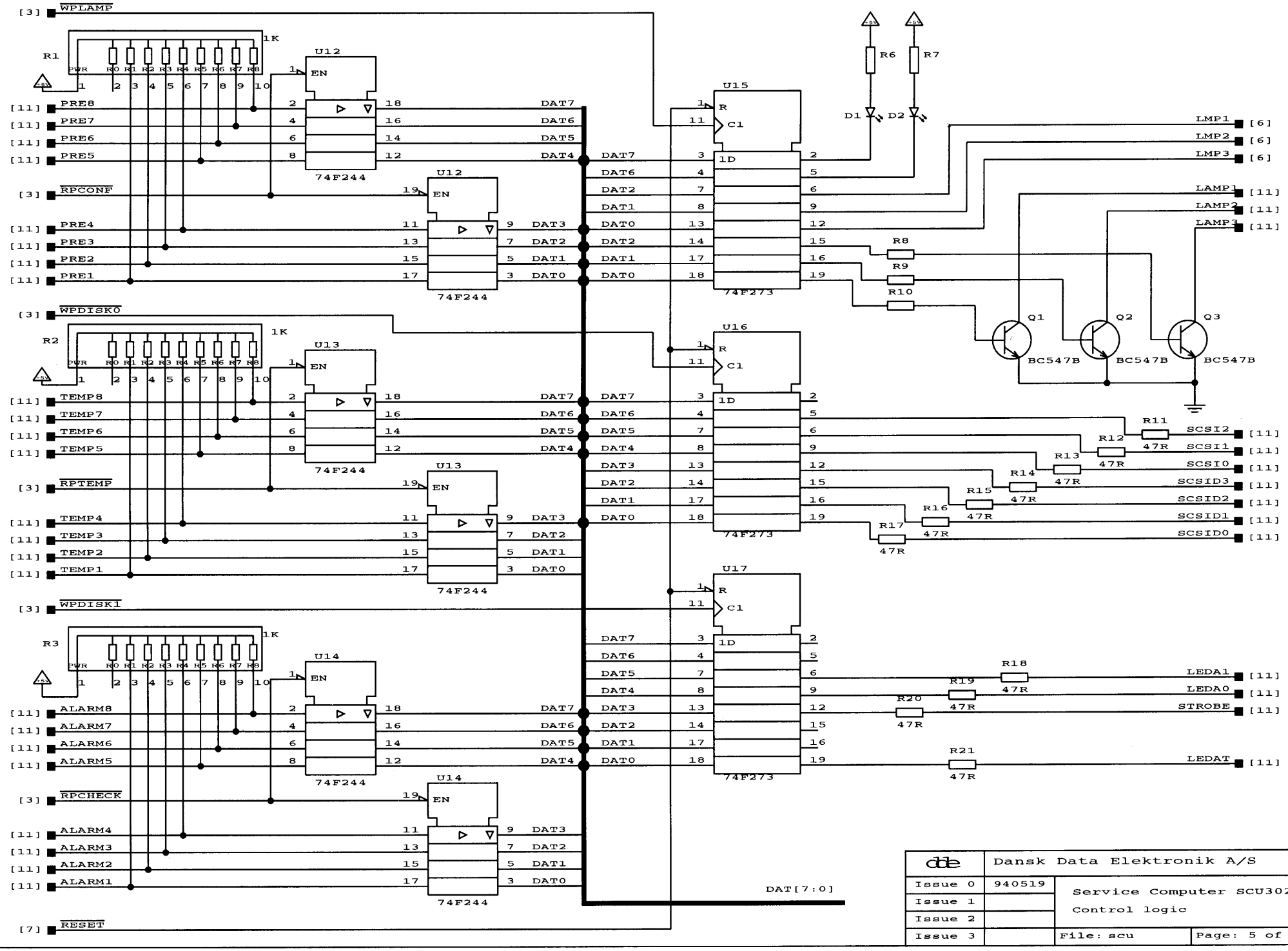
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 Address/Data Register	
Issue 1			
Issue 2			
Issue 3		File: scu	Page: 2 of 12

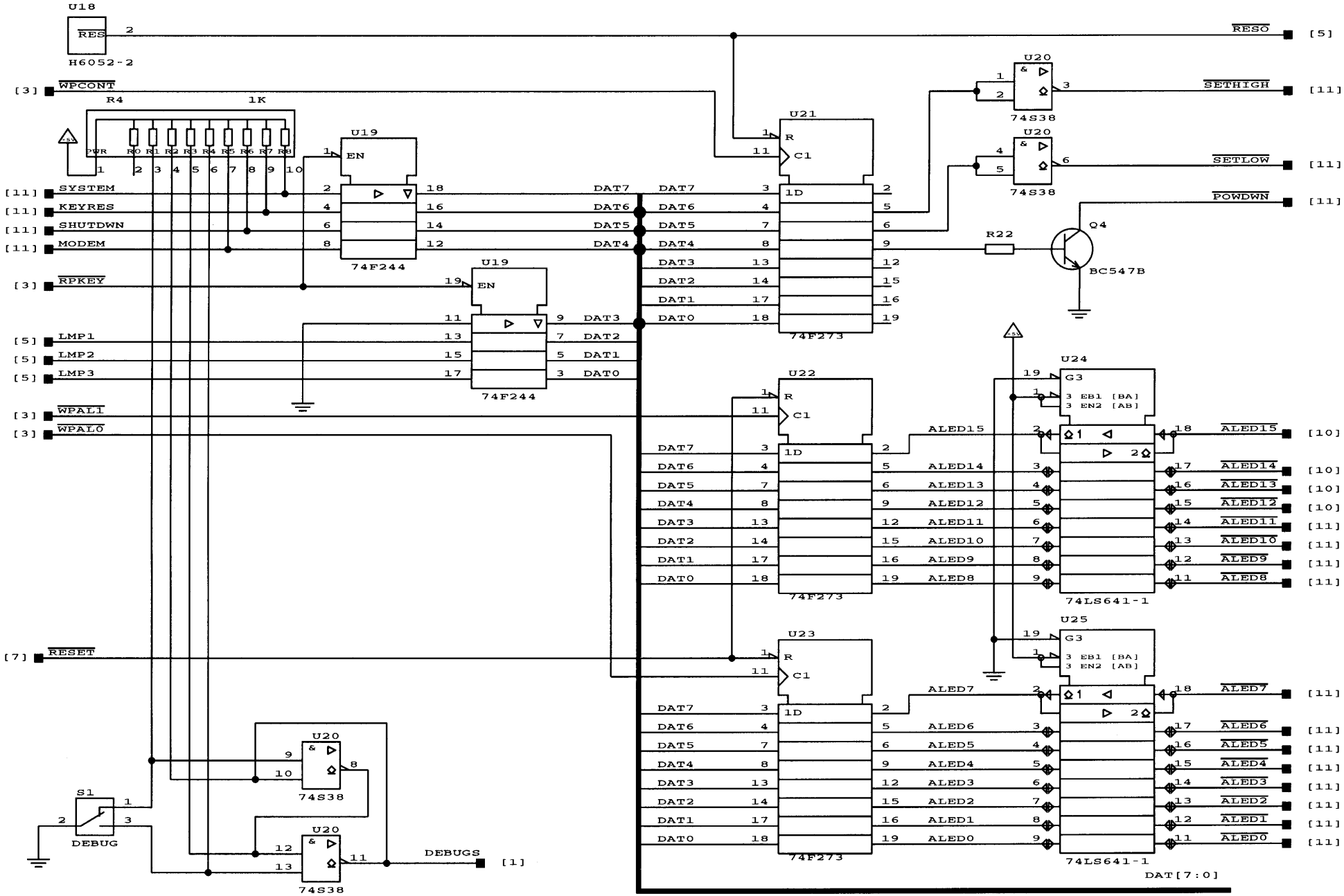


dde	Dansk Data Elektronik A/S	
Issue 0	940519	Service Computer SCU302 SMI control
Issue 1		
Issue 2		
Issue 3		File: scu — Page: 3 of 12

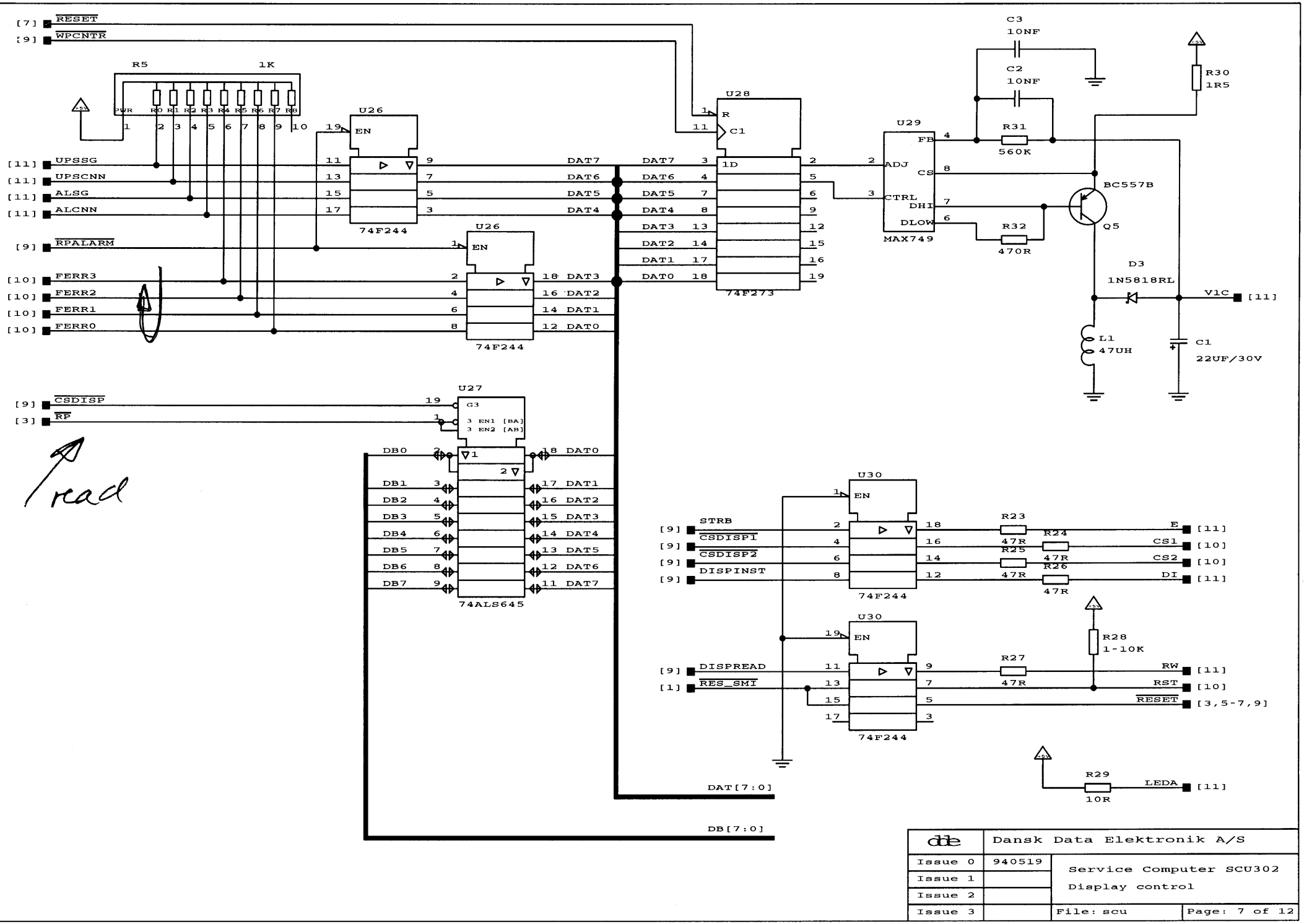


dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302	
Issue 1		ID and Revision PROMs	
Issue 2			
Issue 3		File: scu	Page: 4 of 12

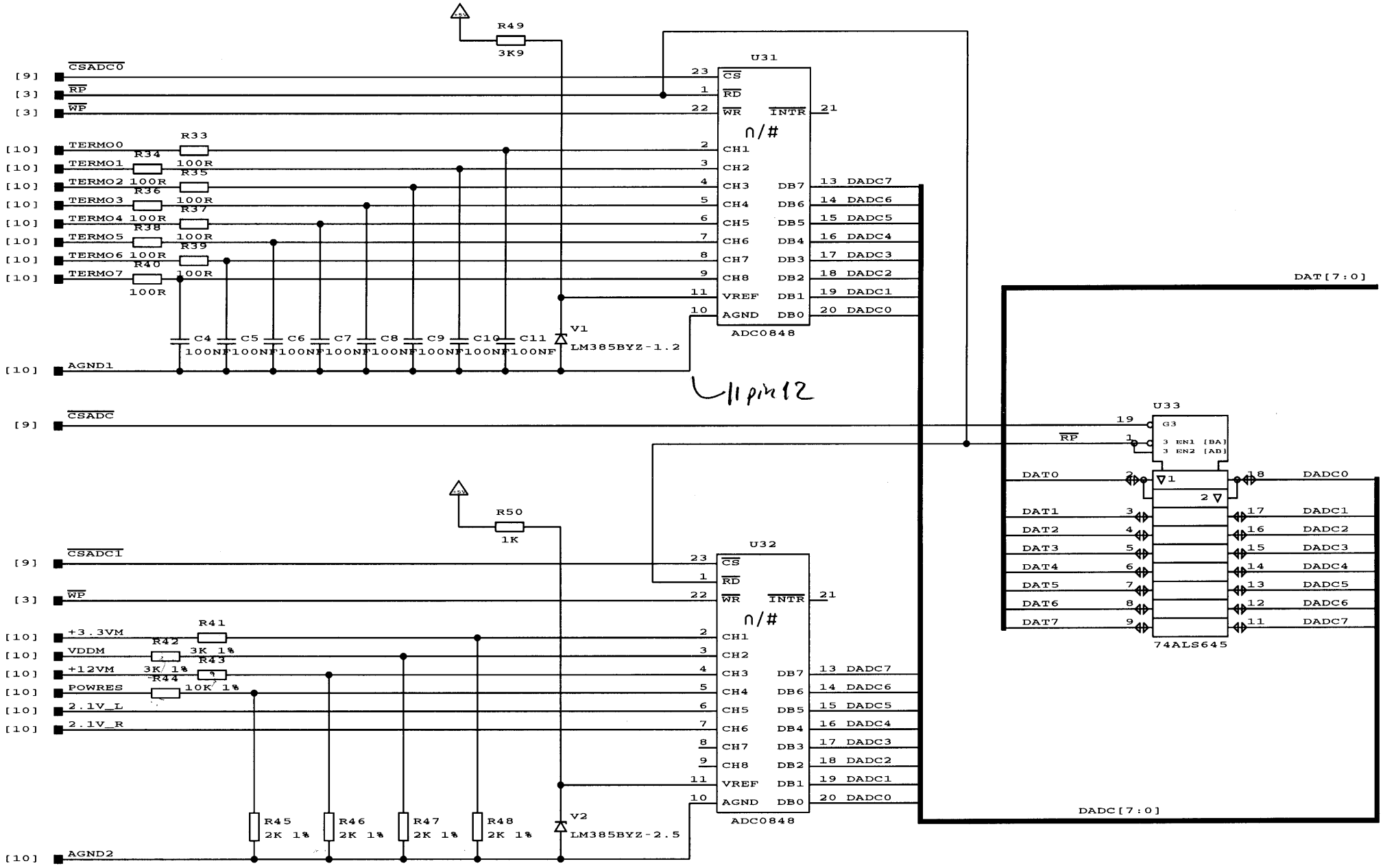

[3] WPLAMP
R1
1K
PWR R0 R1 R2 R3 R4 R5 R6 R7 R8
U12
EN
[11] PRE8
[11] PRE7
[11] PRE6
[11] PRE5
74F244
[3] RPCONF
[11] PRE4
[11] PRE3
[11] PRE2
[11] PRE1
DAT7
DAT6
DAT5
DAT4
DAT3
DAT2
DAT1
DAT0
U15
R
C1
1D
74F273
D1
D2
R6
R7
LMP1 [6]
LMP2 [6]
LMP3 [6]
LAMP1 [11]
LAMP2 [11]
LAMP3 [11]
R8
R9
R10
Q1
Q2
Q3
BC547B
[3] WPDISK0
R2
U13
[11] TEMP8
[11] TEMP7
[11] TEMP6
[11] TEMP5
[3] RPTEMP
[11] TEMP4
[11] TEMP3
[11] TEMP2
[11] TEMP1
U16
R11
R12
R13
R14
R15
R16
R17
47R
SCSI2 [11]
SCSI1 [11]
SCSI0 [11]
SCSID3 [11]
SCSID2 [11]
SCSID1 [11]
SCSID0 [11]
[3] WPDISK1
R3
U14
[11] ALARM8
[11] ALARM7
[11] ALARM6
[11] ALARM5
[3] RPCHECK
[11] ALARM4
[11] ALARM3
[11] ALARM2
[11] ALARM1
U17
R18
R19
R20
R21
LEDA1 [11]
LEDA0 [11]
STROBE [11]
LEDAT [11]
DAT[7:0]
[7] RESET
Dansk Data Elektronik A/S
Issue 0 940519
Issue 1
Issue 2
Issue 3
Service Computer SCU302
Control logic
File: scu
Page: 5 of 12



dde	Dansk Data Elektronik A/S	
Issue 0	940519	Service Computer SCU302
Issue 1		Control logic
Issue 2		
Issue 3		File: scu — Page: 6 of 12



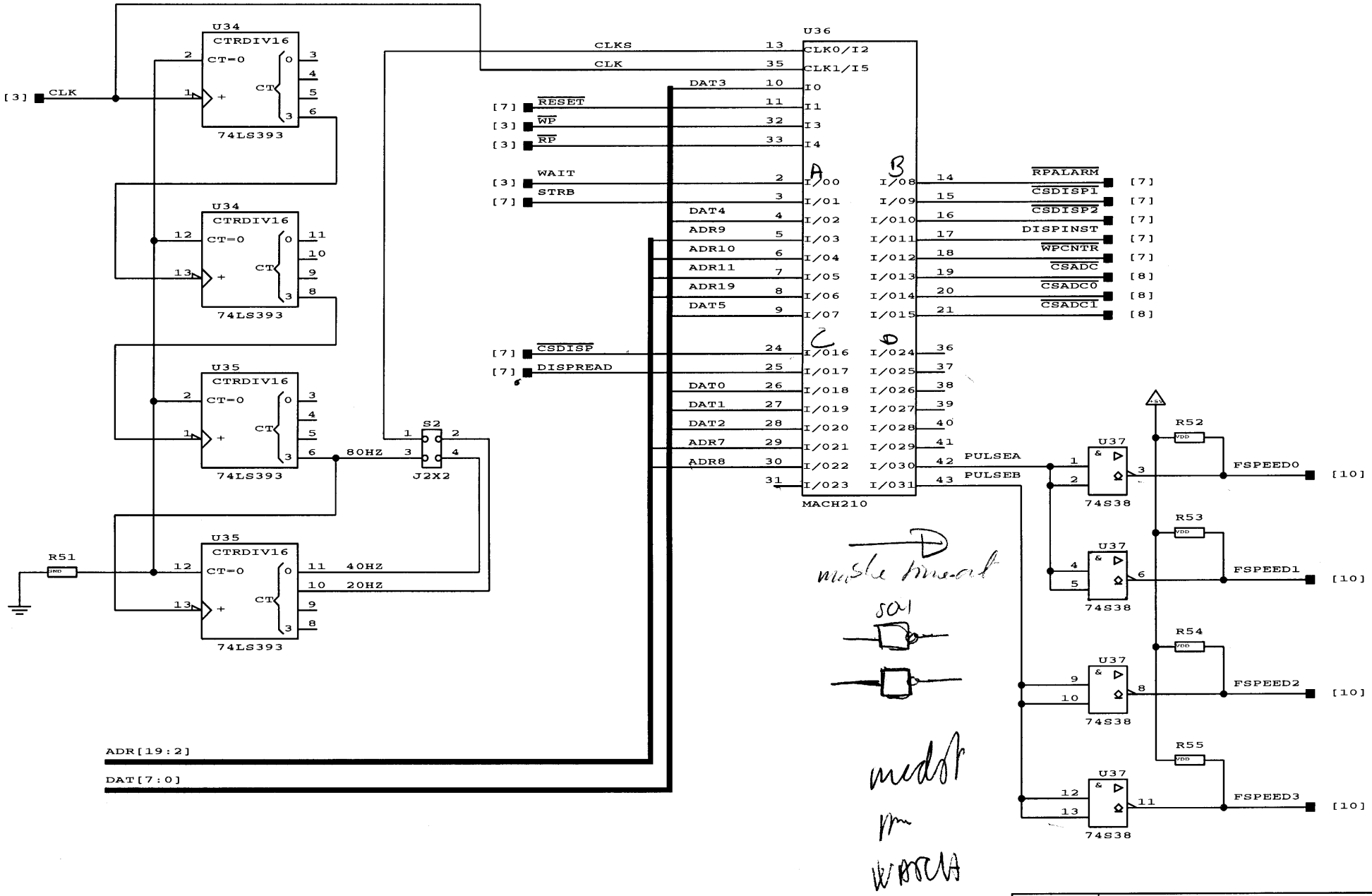
dde	Dansk Data Elektronik A/S	
Issue 0	940519	Service Computer SCU302 Display control
Issue 1		
Issue 2		
Issue 3		File: scu Page: 7 of 12



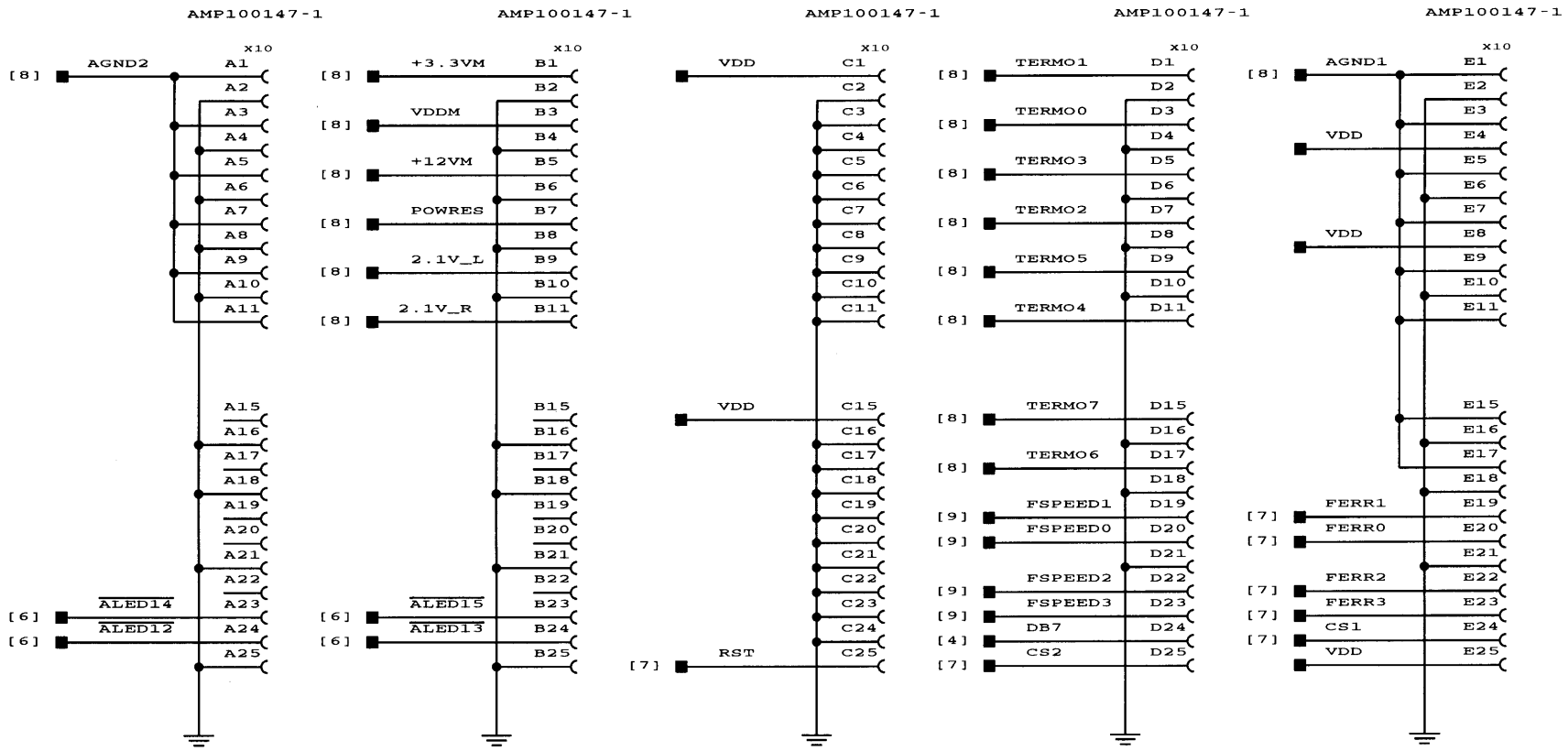
11 pin 12

11 pin 12

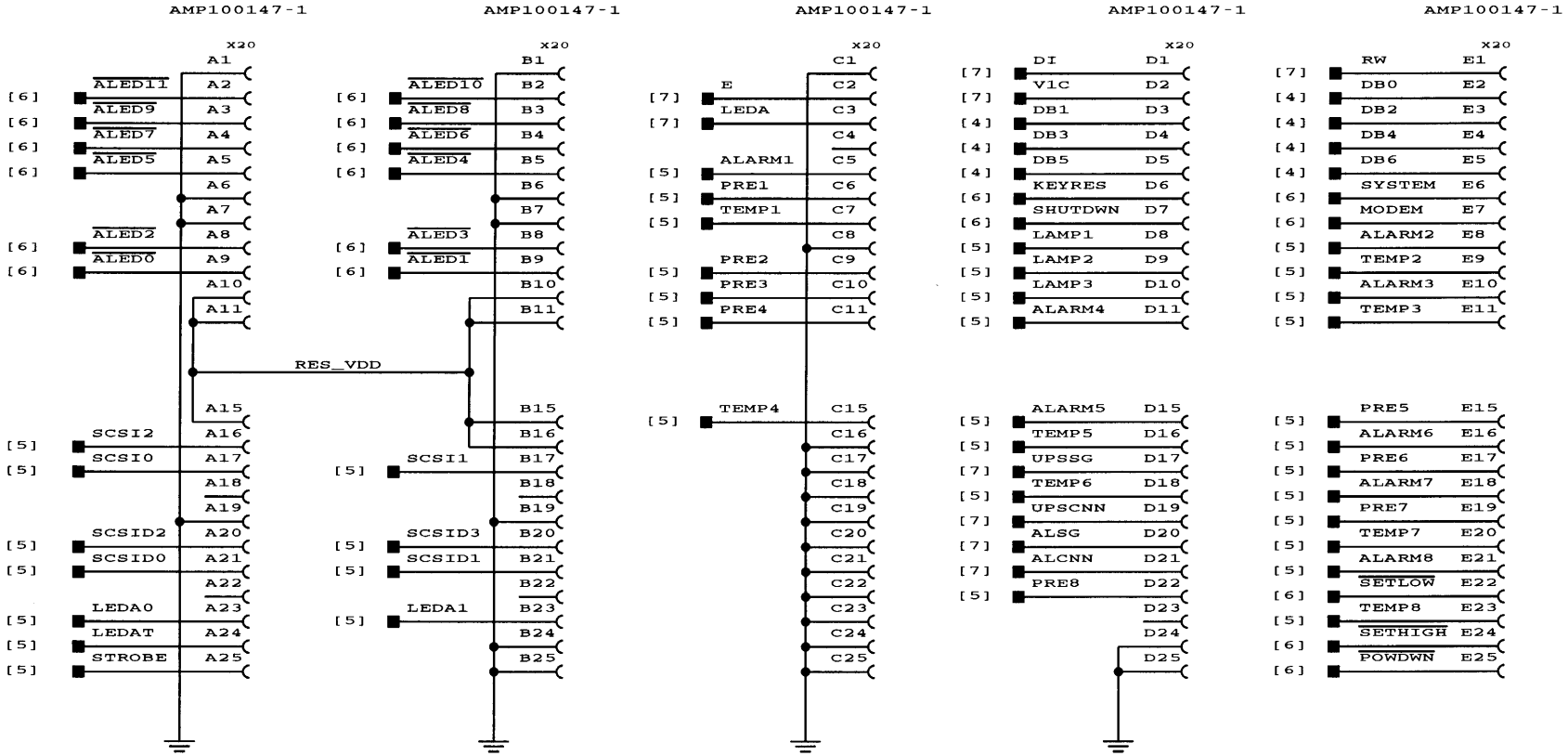
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 A/D converters	
Issue 1			
Issue 2			
Issue 3		File: scu	Page: 8 of 12



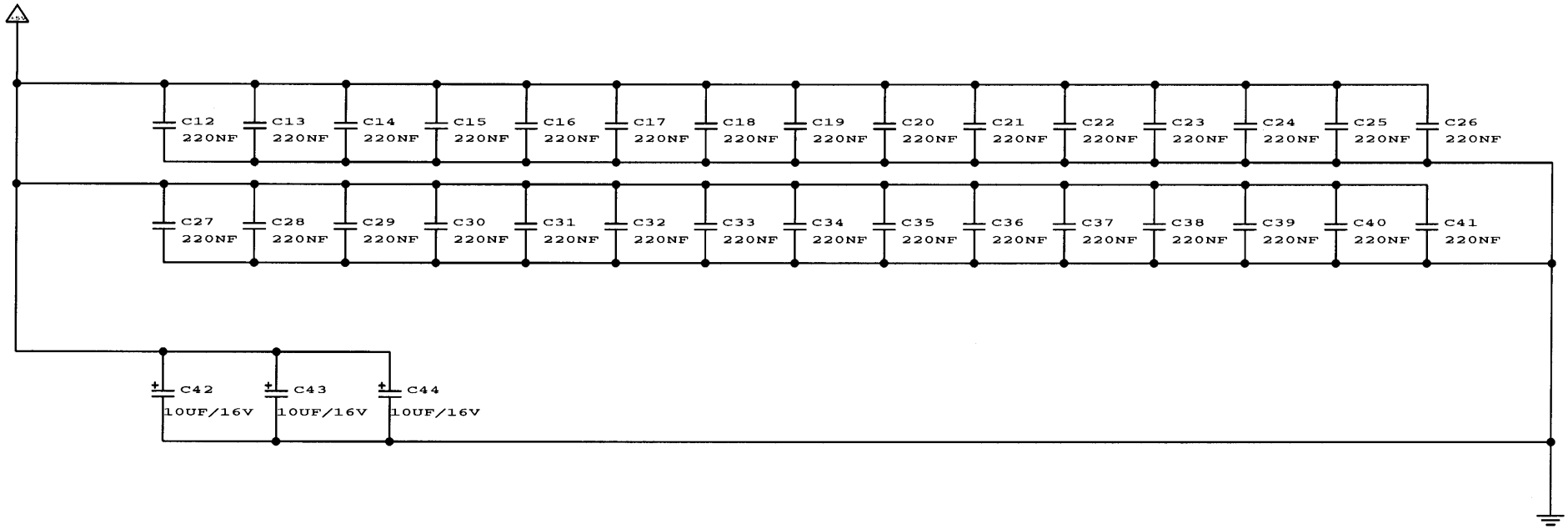
dde	Dansk Data Elektronik A/S	
Issue 0	940519	Service Computer SCU302
Issue 1		Control logic
Issue 2		
Issue 3		File: scu — Page: 9 of 12



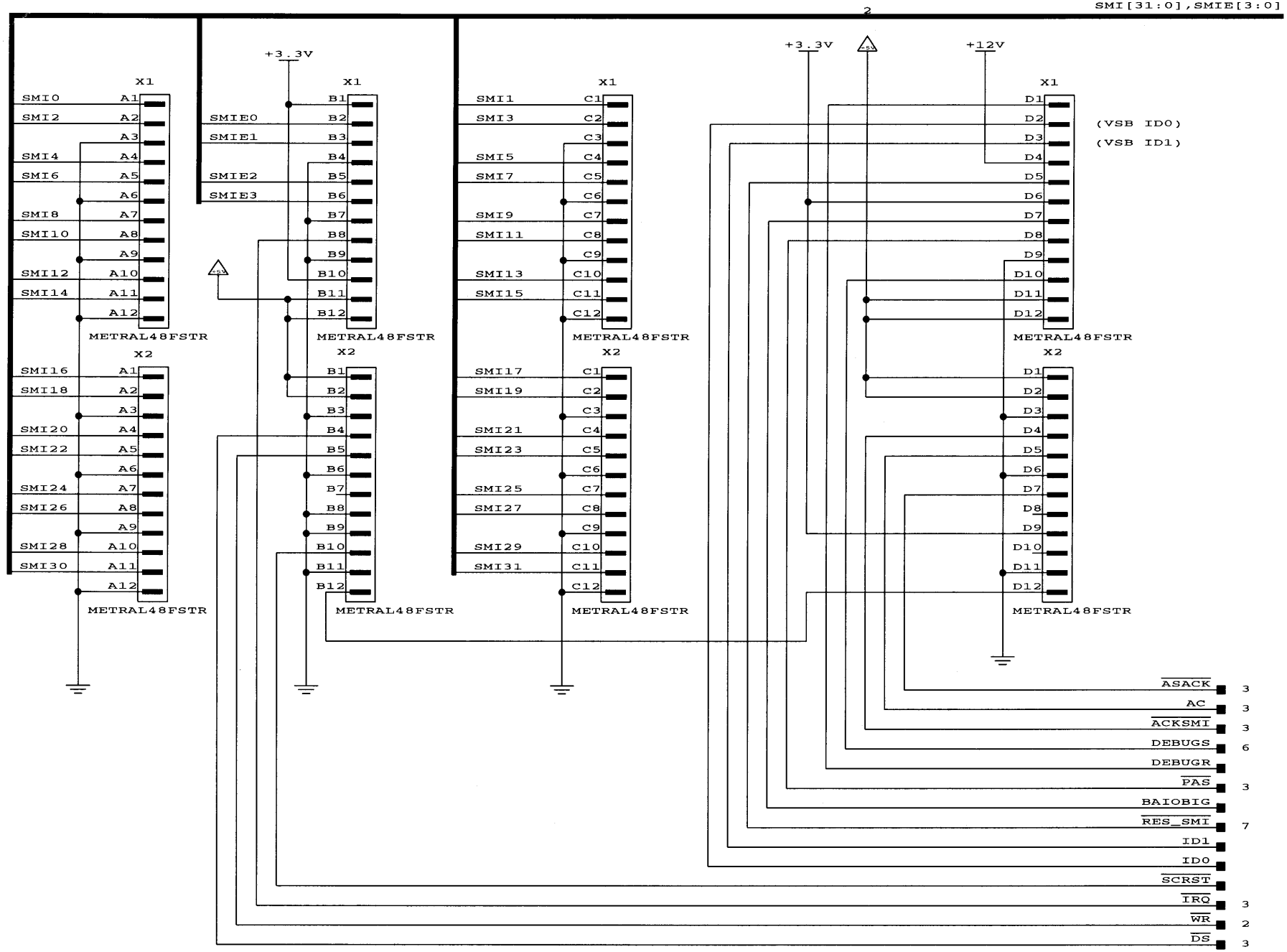
dde	Dansk Data Elektronik A/S	
Issue 0	940519	Service Computer SCU302 Back panel connectors
Issue 1		
Issue 2		
Issue 3		File: scu — Page:10 of 12



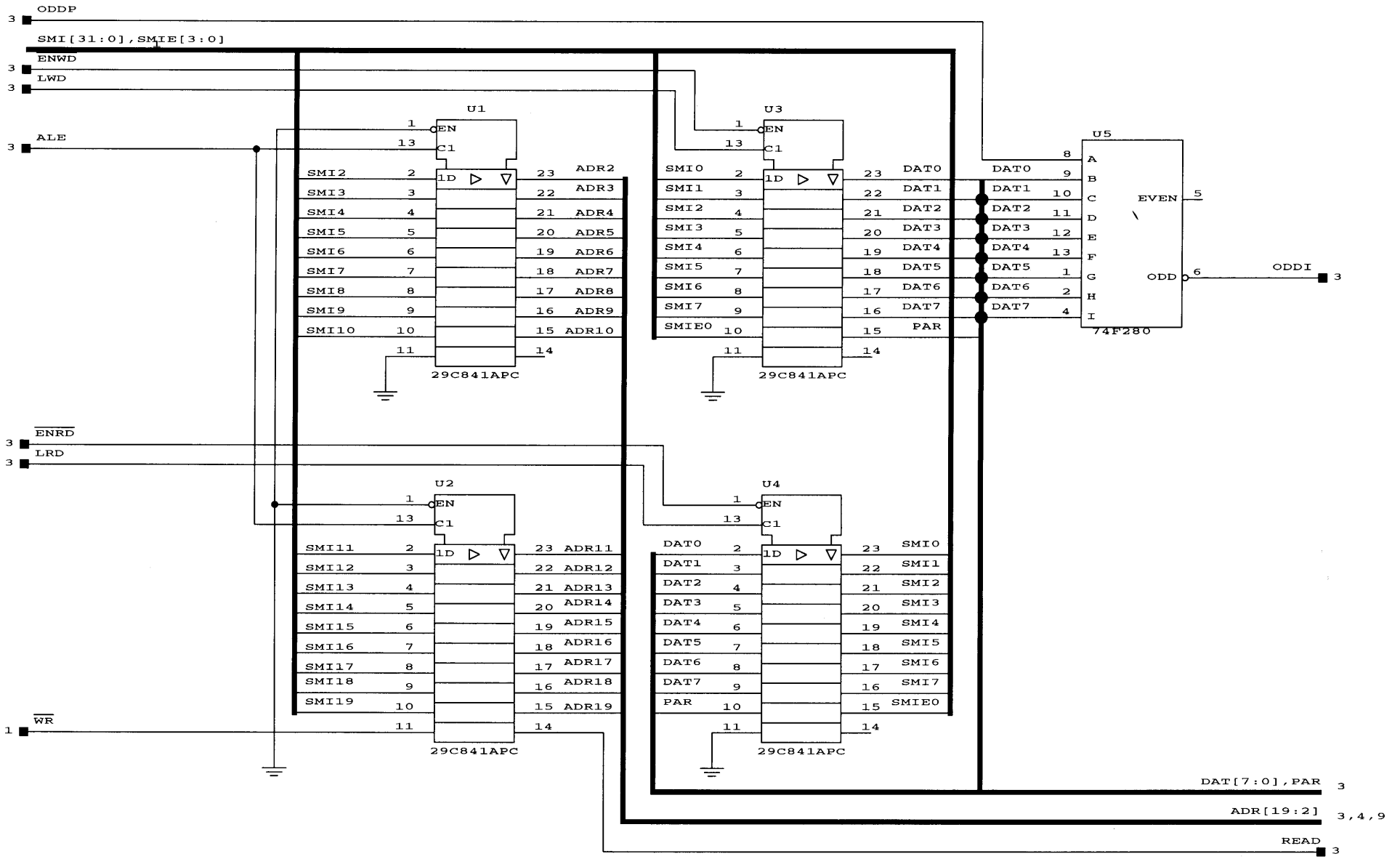
dde	Dansk Data Elektronik A/S	
Issue 0	940519	Service Computer SCU302
Issue 1		Back panel connectors
Issue 2		
Issue 3		File: scu — Page:11 of 12



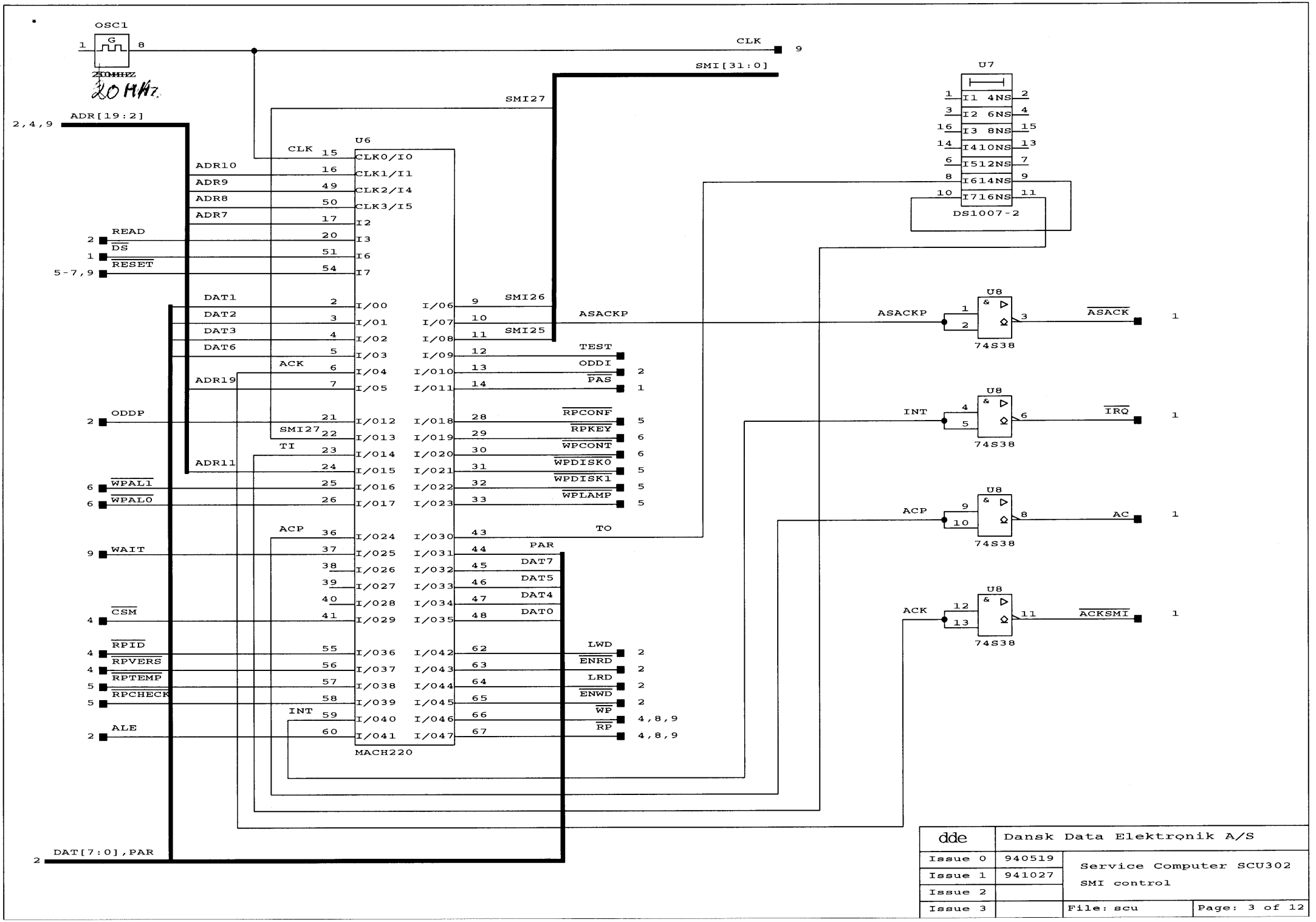
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 Decoupling capacitors	
Issue 1			
Issue 2			
Issue 3		File: scu	Page:12 of 12

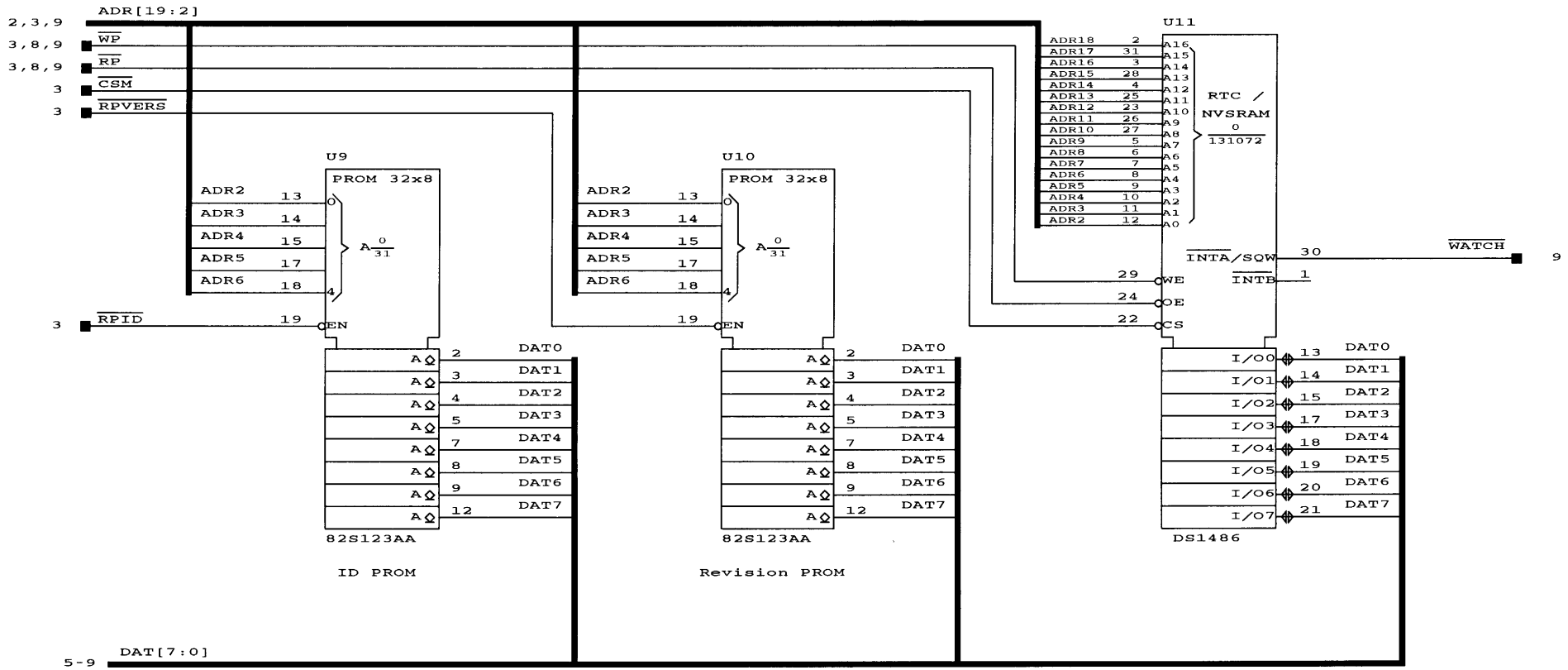


dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 SMI interface	
Issue 1	941027		
Issue 2			
Issue 3		File: scu	Page: 1 of 12

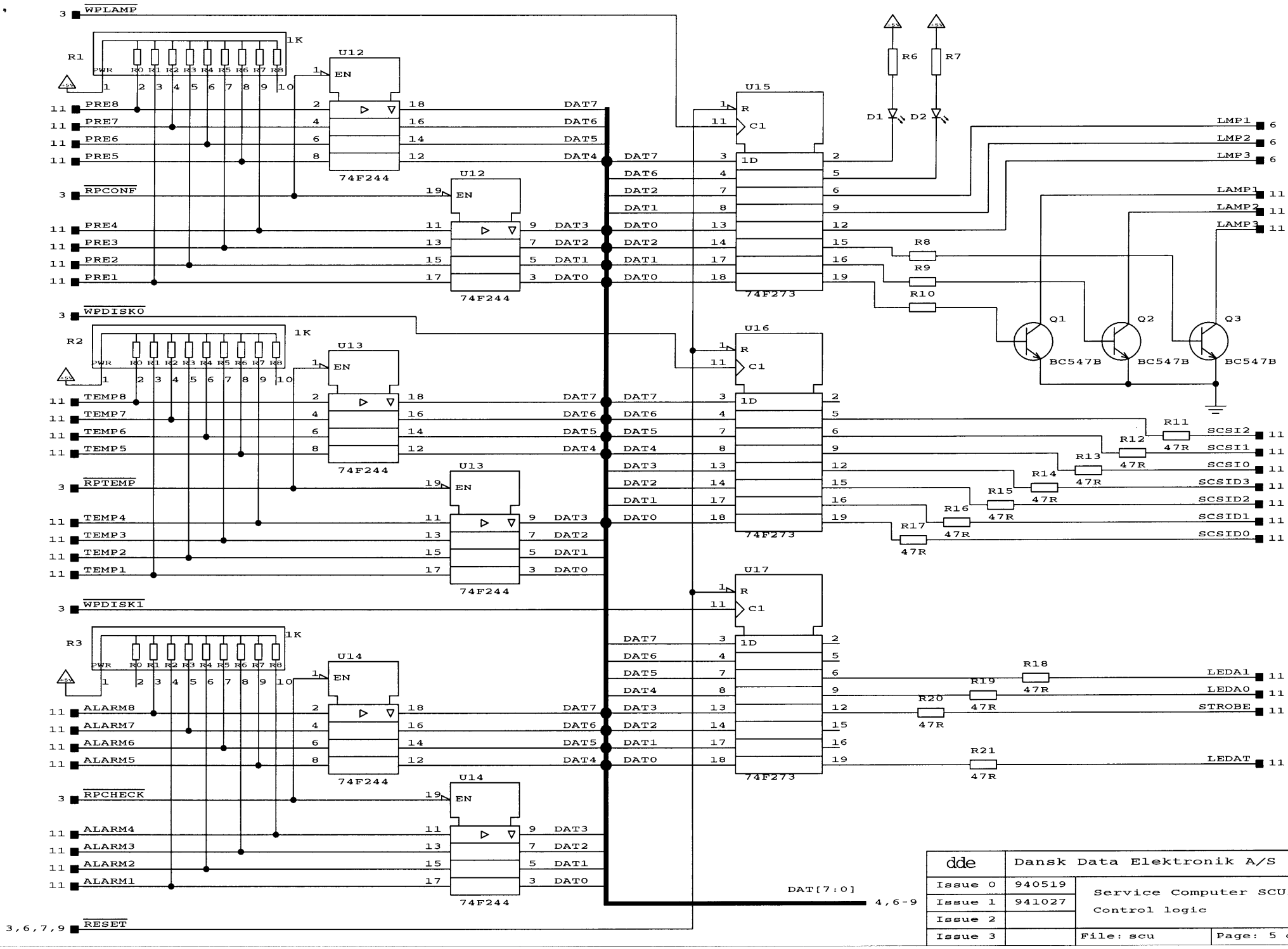


dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302	
Issue 1	941027	Address/Data Register	
Issue 2			
Issue 3		File: scu	Page: 2 of 12

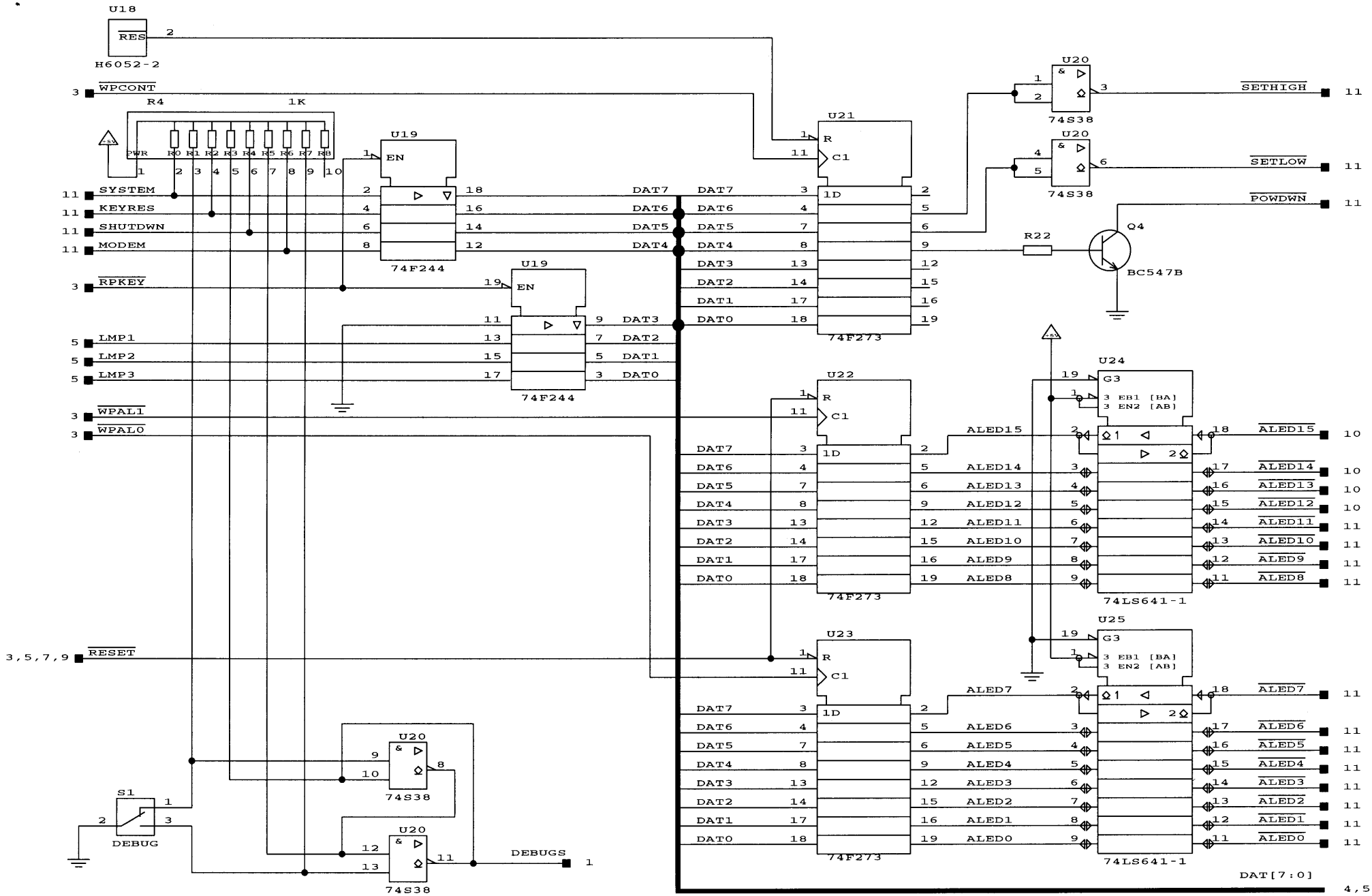

OSC1
G
20MHZ
20 MHz
CLK
SMI[31:0]
SMI27
ADR[19:2]
2,4,9
U6
CLK
ADR10
ADR9
ADR8
ADR7
READ
DS
RESET
5-7,9
DAT1
DAT2
DAT3
DAT6
ACK
ADR19
ADR11
ODDP
SMI27
TI
WPAL1
WPAL0
WAIT
CSM
RPID
RPVERS
RPTEMP
RPCHECK
ALE
INT
ACP
CLK0/I0
CLK1/I1
CLK2/I4
CLK3/I5
I2
I3
I6
I7
I/00
I/01
I/02
I/03
I/04
I/05
I/012
I/013
I/014
I/015
I/016
I/017
I/024
I/025
I/026
I/027
I/028
I/029
I/036
I/037
I/038
I/039
I/040
I/041
I/06
I/07
I/08
I/09
I/010
I/011
I/018
I/019
I/020
I/021
I/022
I/023
I/030
I/031
I/032
I/033
I/034
I/035
I/042
I/043
I/044
I/045
I/046
I/047
MACH220
SMI26
SMI25
TEST
ODDI
PAS
RPCONF
RPKEY
WPCONT
WPDISK0
WPDISK1
WPLAMP
TO
PAR
DAT7
DAT5
DAT4
DAT0
LWD
ENRD
LRD
ENWD
WP
RP
4,8,9
4,8,9
U7
I1 4NS
I2 6NS
I3 8NS
I410NS
I512NS
I614NS
I716NS
DS1007-2
U8
ASACKP
ASACK
74S38
INT
IRQ
74S38
ACP
AC
74S38
ACK
ACKSMI
74S38
DAT[7:0],PAR
2
dde
Dansk Data Elektronik A/S
Issue 0
940519
Issue 1
941027
Issue 2
Issue 3
Service Computer SCU302
SMI control
File: scu
Page: 3 of 12



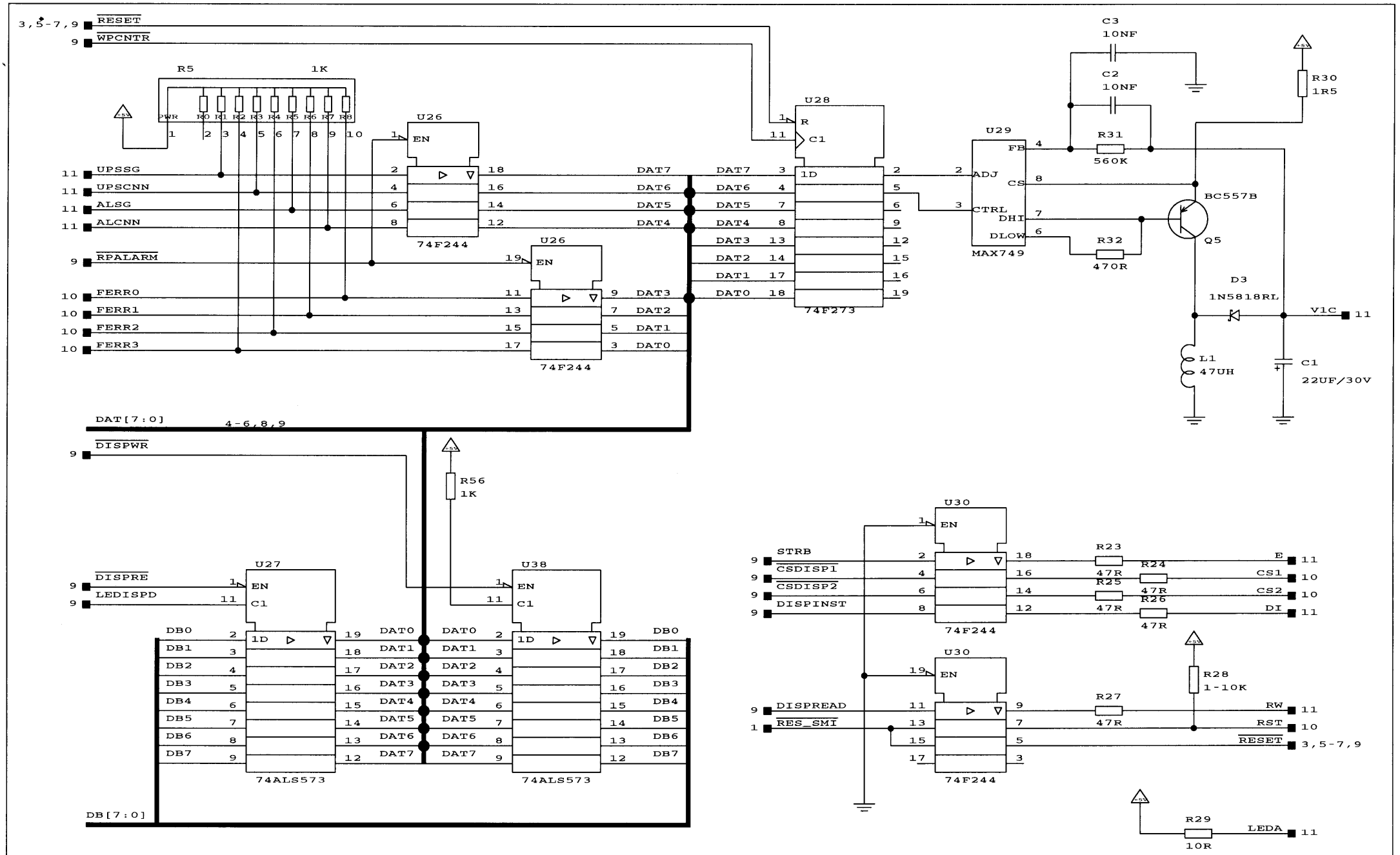
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 ID and Revision PROMs	
Issue 1	941027		
Issue 2			
Issue 3		File: scu	Page: 4 of 12



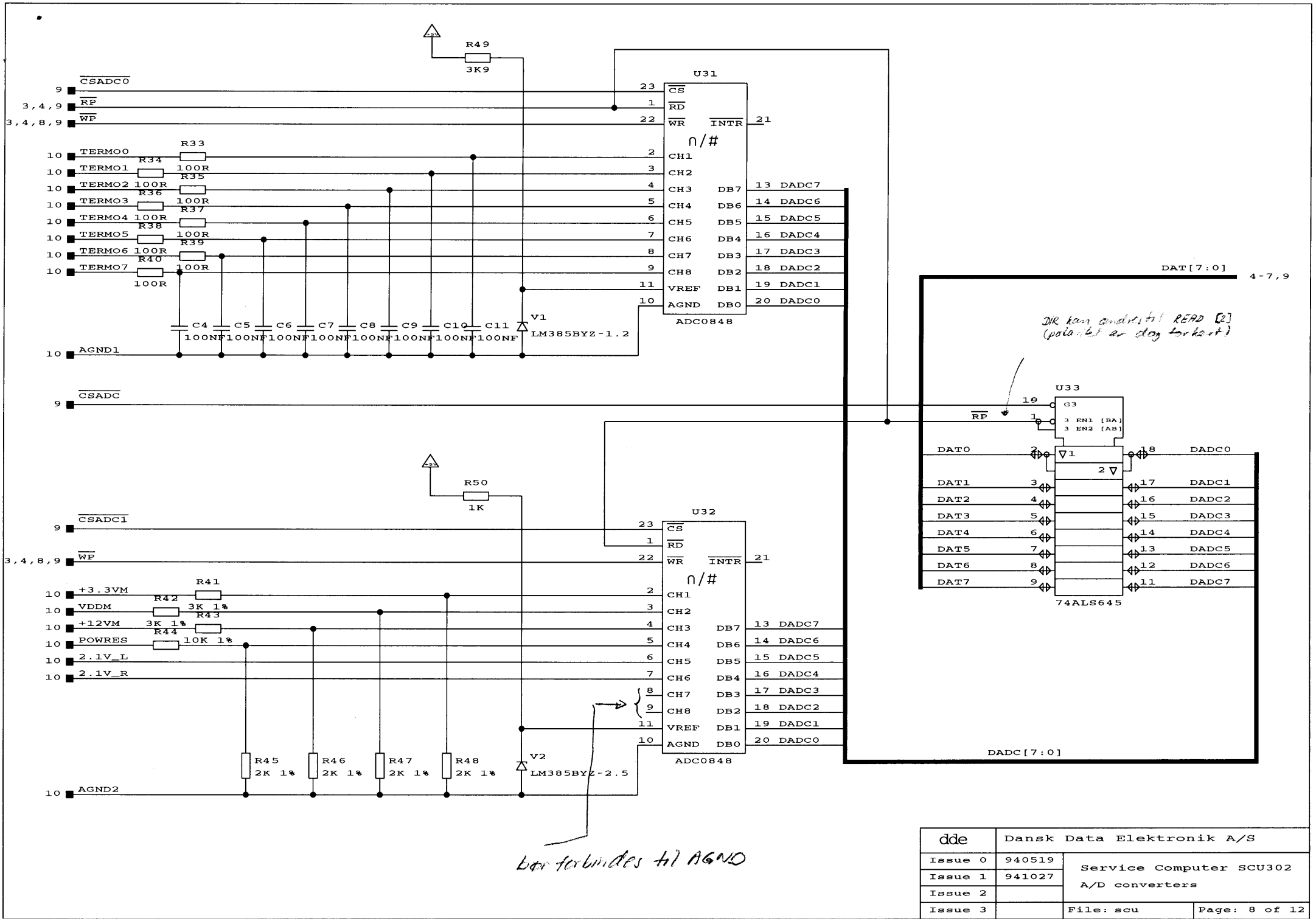
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 Control logic	
Issue 1	941027		
Issue 2			
Issue 3		File: scu	Page: 5 of 12



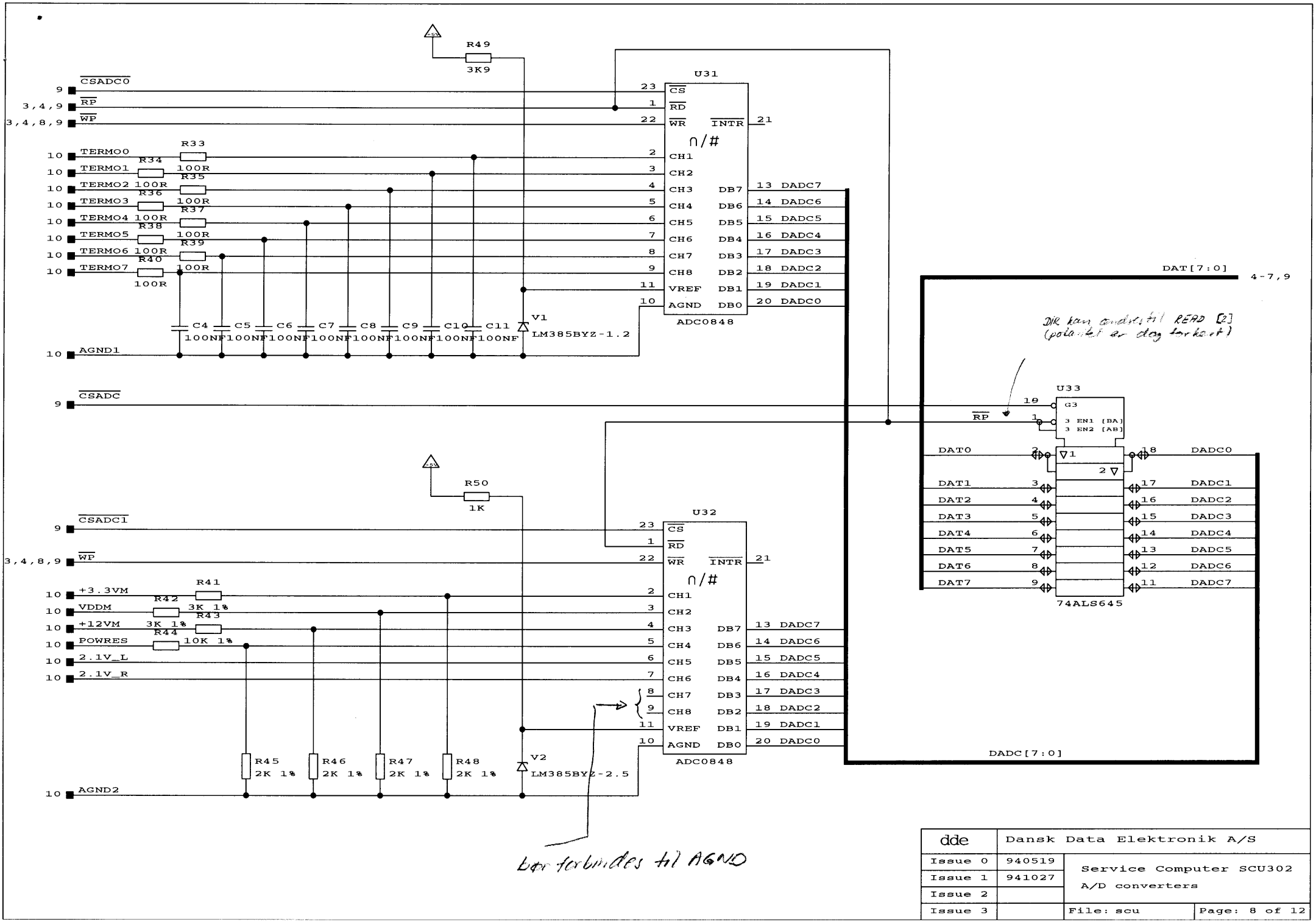
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302	
Issue 1	941027	Control logic	
Issue 2			
Issue 3		File: scu	Page: 6 of 12



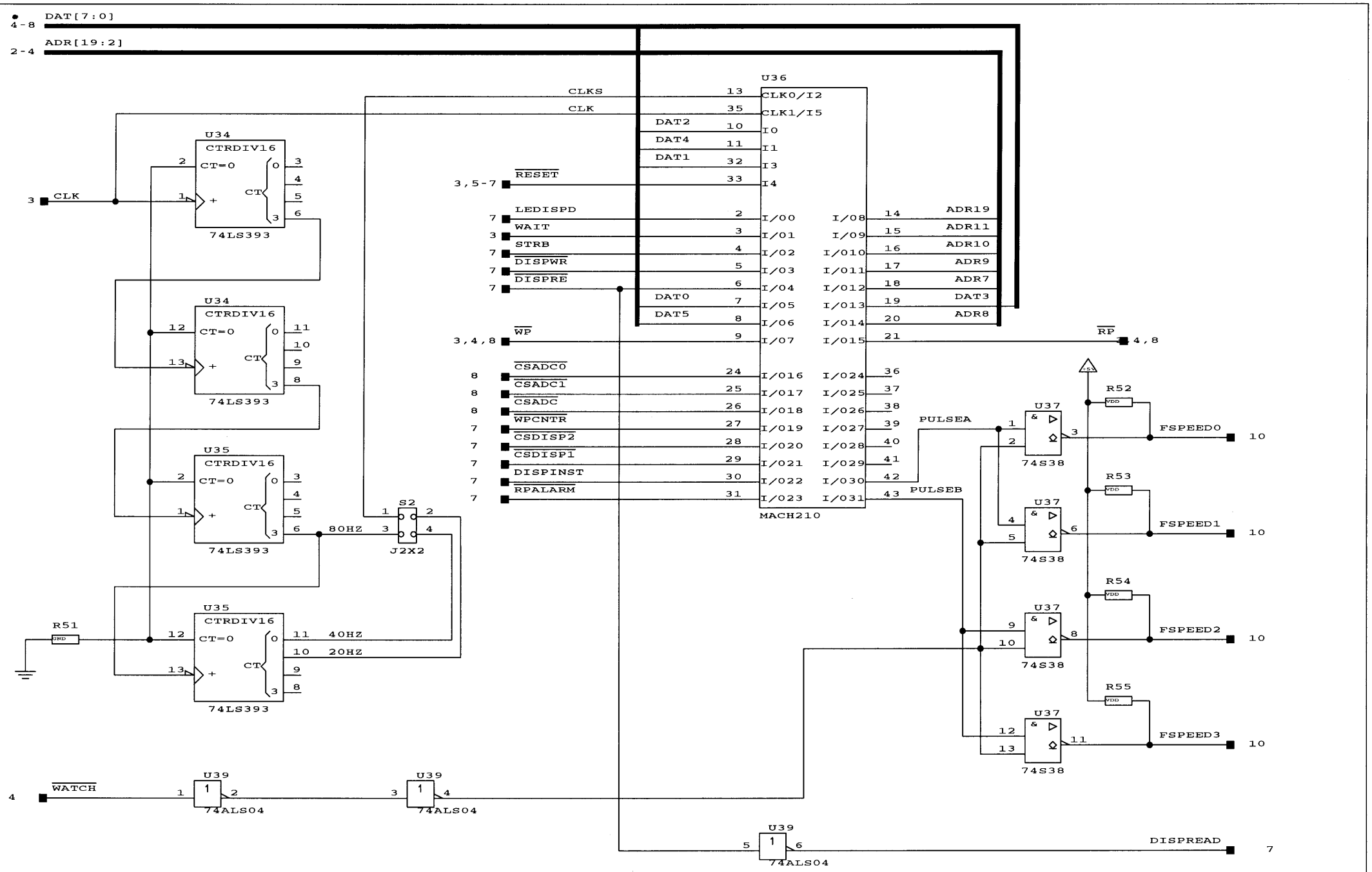
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302	
Issue 1	941027	Display control	
Issue 2			
Issue 3		File: scu	Page: 7 of 12



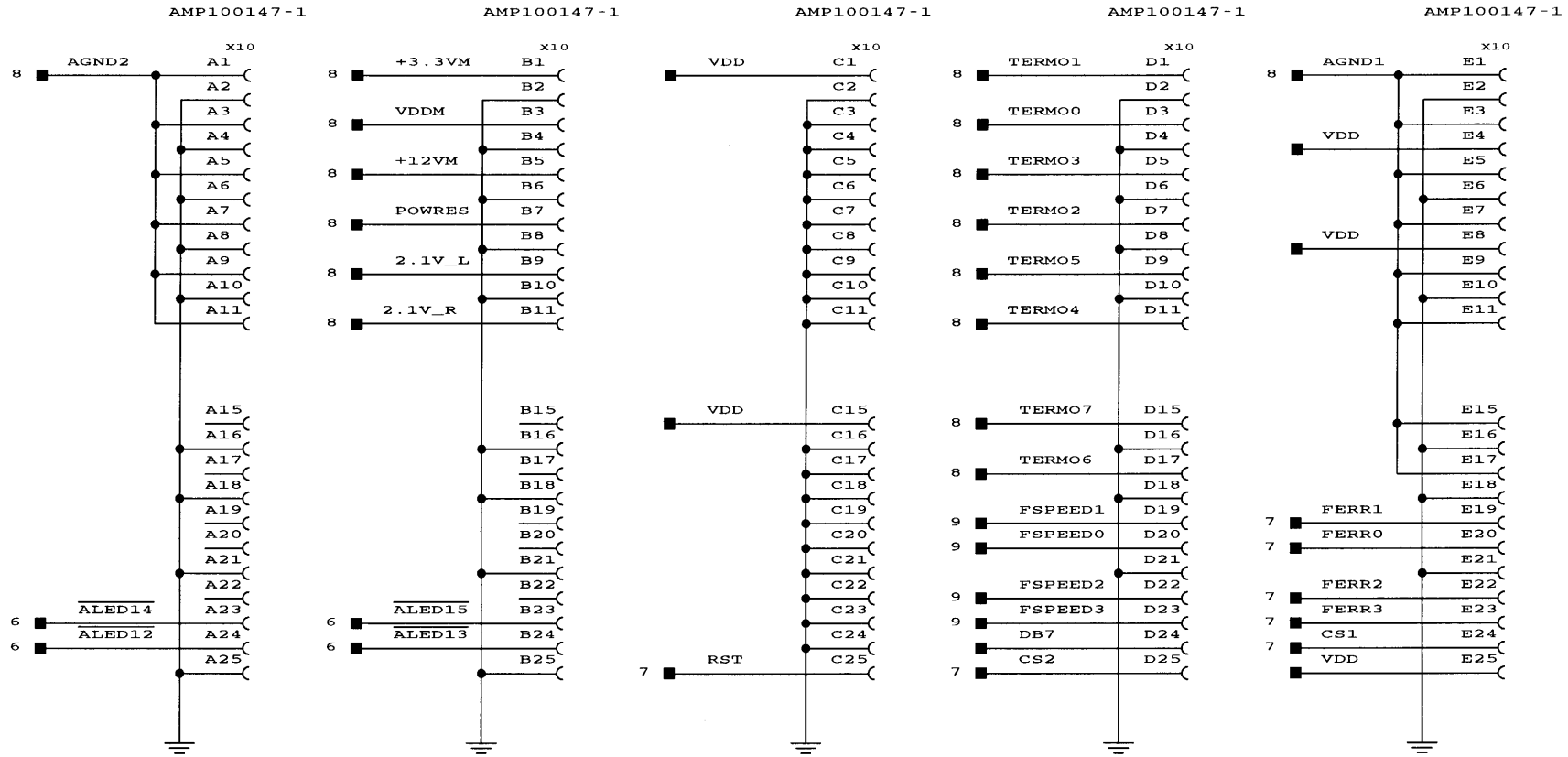
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302	
Issue 1	941027	A/D converters	
Issue 2			
Issue 3		File: scu	Page: 8 of 12



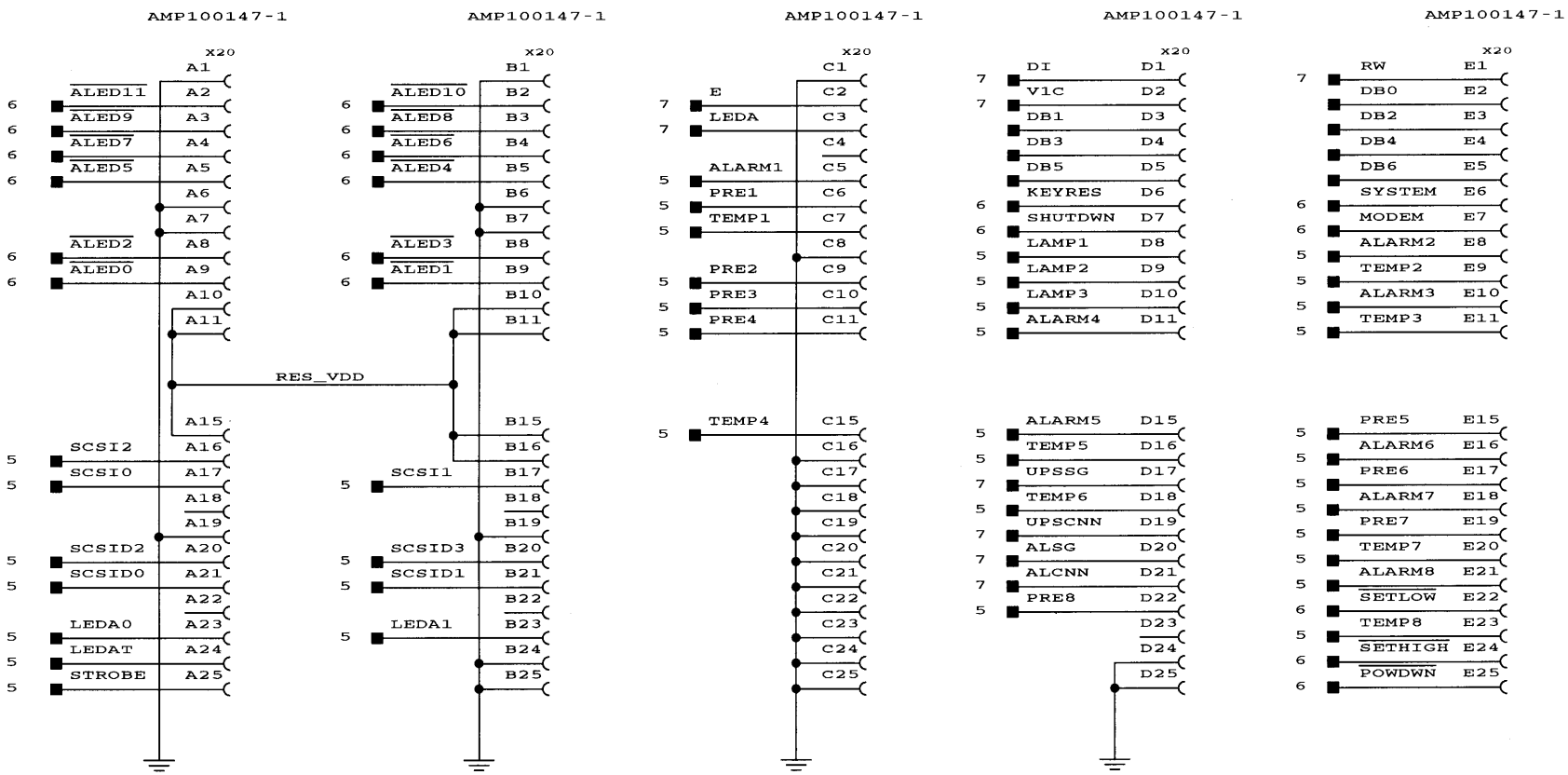
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 A/D converters	
Issue 1	941027		
Issue 2			
Issue 3		File: scu	Page: 8 of 12



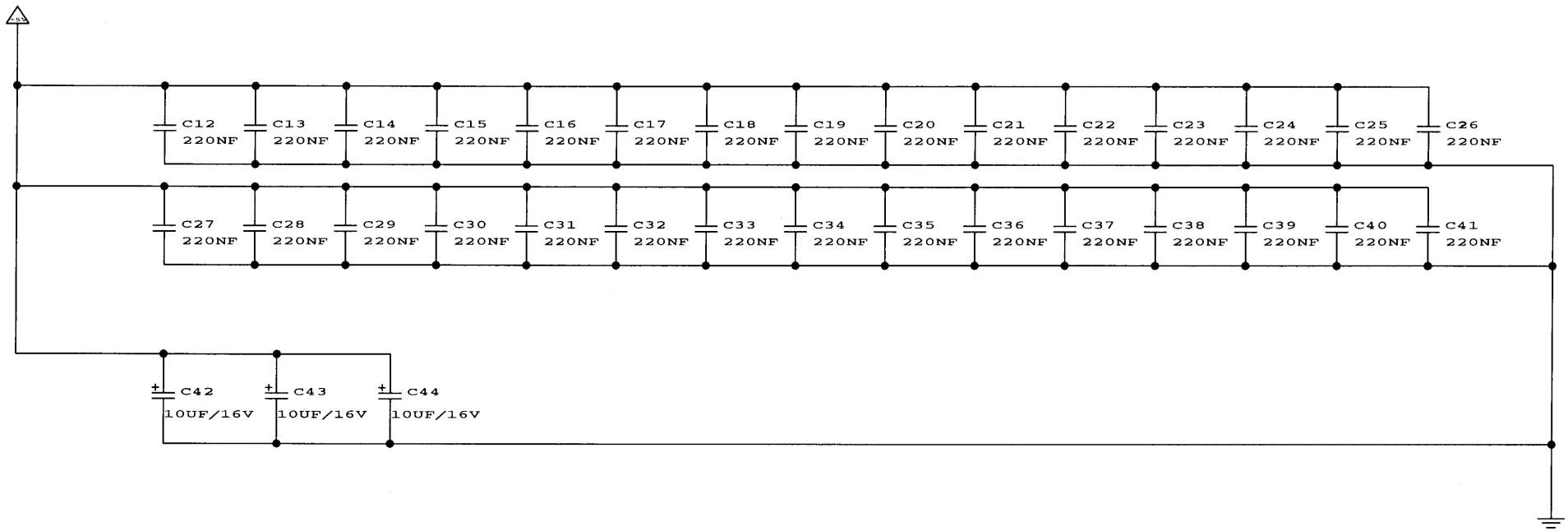
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 Control logic	
Issue 1	941027		
Issue 2			
Issue 3		File: scu	Page: 9 of 12



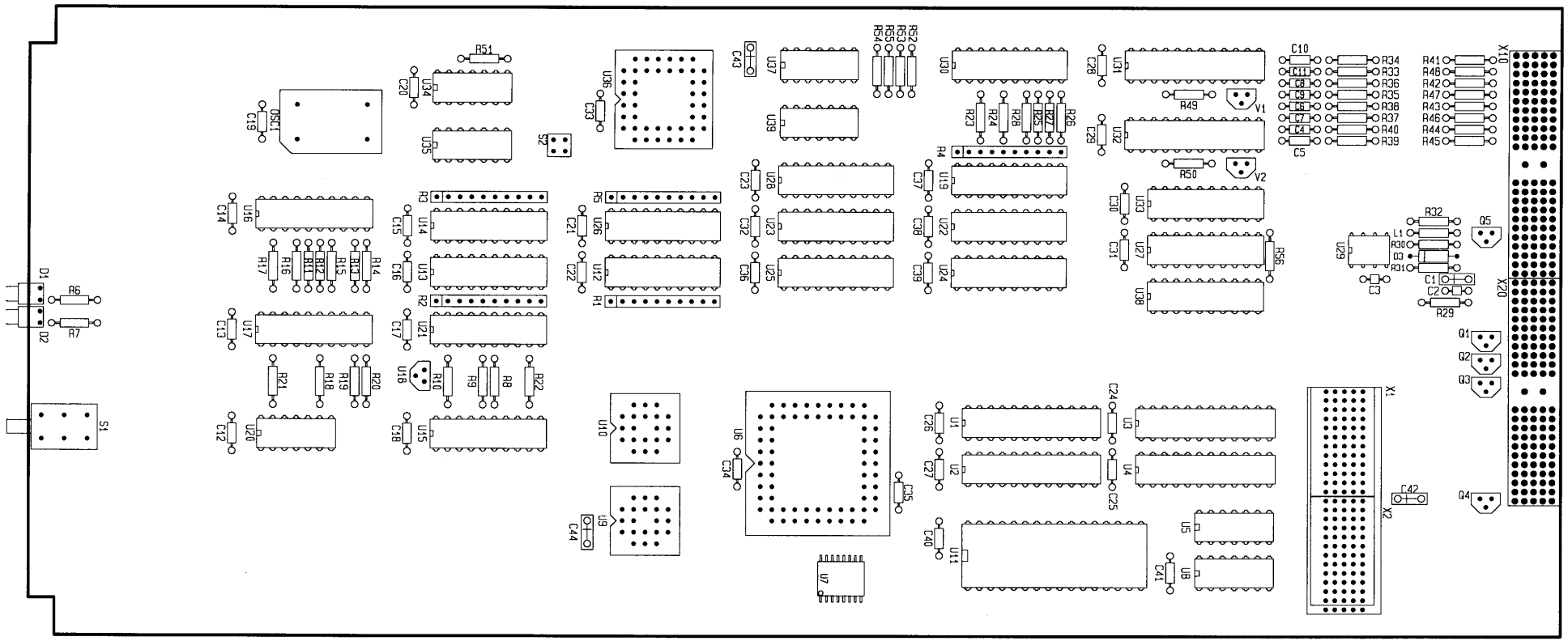
dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 Back panel connectors	
Issue 1	941027		
Issue 2			
Issue 3		File: scu	Page:10 of 12



dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 Back panel connectors	
Issue 1	941027		
Issue 2			
Issue 3		File: scu	Page:11 of 12



dde	Dansk Data Elektronik A/S		
Issue 0	940519	Service Computer SCU302 Decoupling capacitors	
Issue 1	941027		
Issue 2			
Issue 3		File: scu	Page:12 of 12



Assembly (Top View)				
PCB: SCU302	Issue: 1	Date: 94-11-18	dansk data elektronik a/s	herlev hovedgade 199, 2730 herlev, tlf. 42-84 50 11

Name	Type	Pos
C1	22uF/30V	N4
C2	10nF	N4
C3	10nF	M4
C4	100nF	M5
C5	100nF	M5
C6	100nF	M5
C7	100nF	M5
C8	100nF	M5
C9	100nF	M5
C10	100nF	M6
C11	100nF	M5
C12	220nF	B2
C13	220nF	B3
C14	220nF	B4
C15	220nF	D4
C16	220nF	D4
C17	220nF	D3
C18	220nF	D2
C19	220nF	C5
C20	220nF	D5
C21	220nF	E4
C22	220nF	E4
C23	220nF	G5
C24	220nF	K3
C25	220nF	K2
C26	220nF	H3
C27	220nF	H2
C28	220nF	K6
C29	220nF	K5
C30	220nF	K4
C31	220nF	K4
C32	220nF	G4
C33	220nF	E5
C34	220nF	G2
C35	220nF	H2
C36	220nF	G4
C37	220nF	H5
C38	220nF	H4
C39	220nF	H4
C40	220nF	H2
C41	220nF	K1
C42	10uF/16V	M2
C43	10uF/16V	G6
C44	10uF/16V	E2
D1	LED	A4
D2	LED	A3
D3	1N5818RL	M4
L1	47uH	N4
OSC1	20MHz	C5
Q1	BC547B	N3
Q2	BC547B	N3
Q3	BC547B	N3
Q4	BC547B	N2
Q5	BC557B	N4
R1	1K	F3
R2	1K	D3
R3	1K	D4
R4	1K	J5
R5	1K	F4
R6	100R	A3
R7	100R	A3
R8	100R	E3
R9	100R	D3
R10	100R	D3
R11	47R	C4
R12	47R	C4
R13	47R	C4
R14	47R	C4
R15	47R	C4
R16	47R	C4
R17	47R	C4
R18	47R	C3
R19	47R	C3
R20	47R	C3
R21	47R	C3
R22	100R	E3
R23	47R	J5
R24	47R	J5

Name	Type	Pos
R25	47R	J5
R26	47R	J5
R27	47R	J5
R28	1-10K	J5
R29	10R	N3
R30	1R5	M4
R31	560K	M4
R32	470R	M4
R33	100R	M5
R34	100R	M6
R35	100R	M5
R36	100R	M5
R37	100R	M5
R38	100R	M5
R39	100R	M5
R40	100R	M5
R41	3K 1%	N6
R42	3K 1%	N5
R43	10K 1%	N5
R44	100R	N5
R45	2K 1%	N5
R46	2K 1%	N5
R47	2K 1%	N5
R48	2K 1%	N5
R49	3K9	K5
R50	1K	K5
R51	100R	E6
R52	100R	H6
R53	100R	H6
R54	100R	H5
R55	100R	H5
R56	1K	L4
S1	DEBUG	A2
S2	J2X2	E5
U1	29C841APC	J2
U2	29C841APC	J2
U3	29C841APC	K2
U4	29C841APC	K2
U5	74F280	L1
U6	MACH220	G2
U7	DS1007-2	H1
U8	74S38	L1
U9	82S123AA	F2
U10	82S123AA	F2
U11	DS1486	J1
U12	74F244	F4
U13	74F244	D4
U14	74F244	D4
U15	74F273	D2
U16	74F273	C4
U17	74F273	C3
U18	H6052-2	D3
U19	74F244	J4
U20	74S38	C2
U21	74F273	D3
U22	74F273	J4
U23	74F273	G4
U24	74LS641-1	J4
U25	74LS641-1	G4
U26	74F244	F4
U27	74ALS645	K4
U28	74F273	G4
U29	MAX749	M4
U30	74F244	J5
U31	ADC0848	K5
U32	ADC0848	K5
U33	74ALS645	K4
U34	74LS393	D5
U35	74LS393	D5
U36	MACH210	F5
U37	74S38	G5
U38	74ALS573	K3
U39	74ALS04	G5
V1	LM385BYZ-1.2	L5
V2	LM385BYZ-2.5	L5
X1	METRAL48FSTR	M3
X2	METRAL48FSTR	M2
X10	AMP100147-1	N6

Name	Type	Pos
X20	AMP100147-1	N4

Component locations

PCB: SCU302 | Issue: 1 | Date: 94-11-18

dansk data elektronik a/s
herlev hovedgade 199, 2730 herlev, tlf. 42-84 50 11

PARTS LIST

Module: scu302 Issue: 1 Date: 950131 Page: 1/1

Part no	Device	Qty	Comp
99.021.001	22uF/16V	1	C1
99.020.817	10nF	2	C2-3
99.020.805	220nF	8	C4-11
99.020.805	220nF	30	C12-41
99.020.950	10uF/16V	3	C42-44
99.021.412	LED	2	D1-2
99.021.206	1N5818RL	1	D3
*SCU302 VI1	47uH	1	L1
99.022.010	20MHz	1	OSC1
99.021.604	BC547B	4	Q1-4
99.021.606	BC557B	1	Q5
99.022.214	1K	5	R1-5
99.020.002	100R	20	R6-10,R22,R33-40,R44,R51-55
99.020.038	47R	16	R11-21,R23-27
99.020.022	5K6	1	R28
99.020.047	10R	1	R29
99.020.065	1R5	1	R30 JEG MENER DET !!
*SCU302 VI2	560K	1	R31
99.020.010	470R	1	R32
99.020.233	3K 1%	2	R41-42
99.020.227	10K 1%	1	R43
99.020.232	2K 1%	4	R45-48
99.020.020	3K9	1	R49
99.020.013	1K	2	R50,R56
99.022.809	DEBUG	1	S1
	J2X2	1	S2 MONTERES IKKE !
99.012.028	29C841APC	4	U1-4
99.000.817	74AS280	1	U5
99.010.908	MACH220-15	1	U6
99.025.013	DS1007-2	1	U7
99.000.211	74S38	3	U8,U20,U37
99.012.095	82S123AA	2	U9-10
*SCU302 VI3	DS1486	1	U11
99.001.016	74ALS244	6	U12-14,U19,U26,U30
99.000.448	74LS273	7	U15-17,U21-23,U28
99.012.086	H6052-2	1	U18
99.000.464	74LS641-1	2	U24-25
	74ALS573	2	U27,U38
*SCU302 VI4	MAX749	1	U29
99.012.008	ADC0848	2	U31-32
99.001.029	74ALS645	1	U33
99.000.457	74LS393	2	U34-35
99.010.905	MACH210	1	U36
	74ALS04	1	U39
99.005.023	LM385BYZ-1.2	1	V1
99.005.024	LM385BYZ-2.5	1	V2
	AMP188508-1	2	X1-2
	AMP106138-2	2	X1-2
99.040.111	AMP100147-1	2	X10,X20
99.023.011	IC-SOCKET 32P 6M	1	U11
99.023.219	PLCC-SOCKET 68P	1	U6
99.023.225	PLCC-SOCKET 44P	1	U36
	PLCC-SOCKET 20P	2	U9-10

Frigivelse
af
Programmerbare Komponenter

Modulnavn: SCU302-1

Dato: 950201

Initialer: kan

Filnavn	Checksum	Label	Type	Råvarenummer
scu100.jed	00046D79	SCU100	MACH220-15	99.010.908
scu200.jed	0003563E	SCU200	MACH210-15	99.010.905

T.....L.....T.....T.....T.....T.....T.....T.....T....R

kan 950201

Ved indkøring af SCU302-1 er der fundet følgende fejl/problemer:

Første eksemplar af SCU302 ISS 1 er modtaget.

U39 er ikke monteret, da den mangler i stykliste.

Stykliste er opdateret findes i scu.stk dateret 950131.

Kortet monteres med pal mach PAL'er SCU100 og SCU200.

Lysdioder er igen vendt forkert i printet. Lysdioder skal derfor vendes i huset, inden de monters.

Der er fejl i kabel til Alarm, UPS, Power Off. Beskrivelsen er rigtig. Ivan er opmærksom på problemet.

Tilføjet d. 2/2-1995.

Fejl i blæserstyring fundet dd.

Fejlen rettes på følgende måde:

Forbindelsen mellem U11 pin 30 og U39 pin 1 afbrydes på printets underside. U11 er RAM-uret.

I stedet forbindes U11 pin 1 til U39 pin 1 med en ledning på printets underside.

På printets overside placeres en modstand på 10 Kohm mellem U39 pin 1 og U39 pin 14. Pull-up til det nye signal.

Af hensyn til testprogrammerne jordes signalet "test" ved at forbinde U9 pin 12 (test) til U9 pin 8 (gnd).

Også sendt til paxmax.

19-0143; Rev 0; 5/93

Evaluation Kit Manual Follows Data Sheet

MAXIM

Digitally Adjustable LCD Bias Supply

MAX749

General Description

The MAX749 generates negative LCD-bias contrast voltages from 2V to 6V inputs. Full-scale output voltage can be scaled to -100V or greater, and is digitally adjustable in 64 equal steps by an internal digital-to-analog converter (DAC). Only seven small surface-mount components are required to build a complete supply. The output voltage can also be adjusted using a PWM signal or a potentiometer.

Unique current-limited control scheme reduces supply current and maximizes efficiency, while a high switching frequency (up to 500kHz) minimizes the size of external components. Quiescent current is only 60µA max and is reduced to under 15µA in shutdown mode. While shut down, the MAX749 retains the voltage set point, simplifying software control. The MAX749 drives either an external P-channel MOSFET or a PNP transistor.

Features

- ♦ **+2.0V to +6.0V Input Voltage Range**
- ♦ **Flexible Control of Output Voltage:**
  - **Digital Control**
  - **Potentiometer Adjustment**
  - **PWM Control**
- ♦ **Output Voltage Range Set by One Resistor**
- ♦ **Low, 60µA Max Quiescent Current**
- ♦ **15µA Max Shutdown Mode**
- ♦ **Small Size – 8-Pin SO and Plastic DIP Packages**

Applications

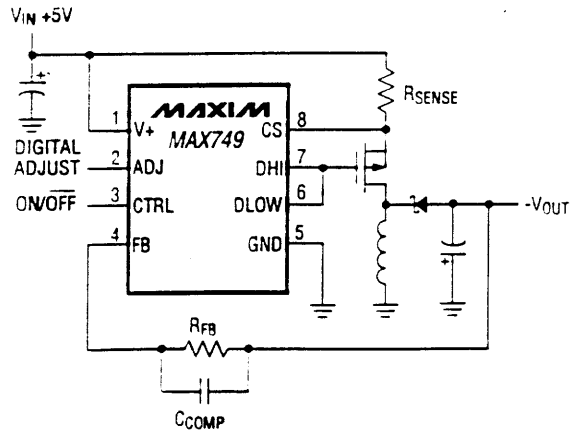
Notebook Computers
Laptop Computers
Palmtop Computers
Personal Digital Assistants
Communicating Computers
Portable Data-Collection Terminals

Ordering Information

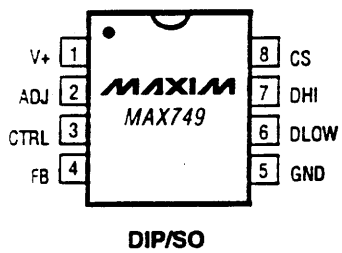
PART	TEMP. RANGE	PIN-PACKAGE
MAX749CPA	0°C to +70°C	8 Plastic DIP
MAX749CSA	0°C to +70°C	8 SO
MAX749C/D	0°C to +70°C	Dice*
MAX749EPA	-40°C to +85°C	8 Plastic DIP
MAX749ESA	-40°C to +85°C	8 SO

** Contact factory for dice specifications.*

Typical Operating Circuit



Pin Configuration



DIP/SO

Digitally Adjustable LCD Bias Supply

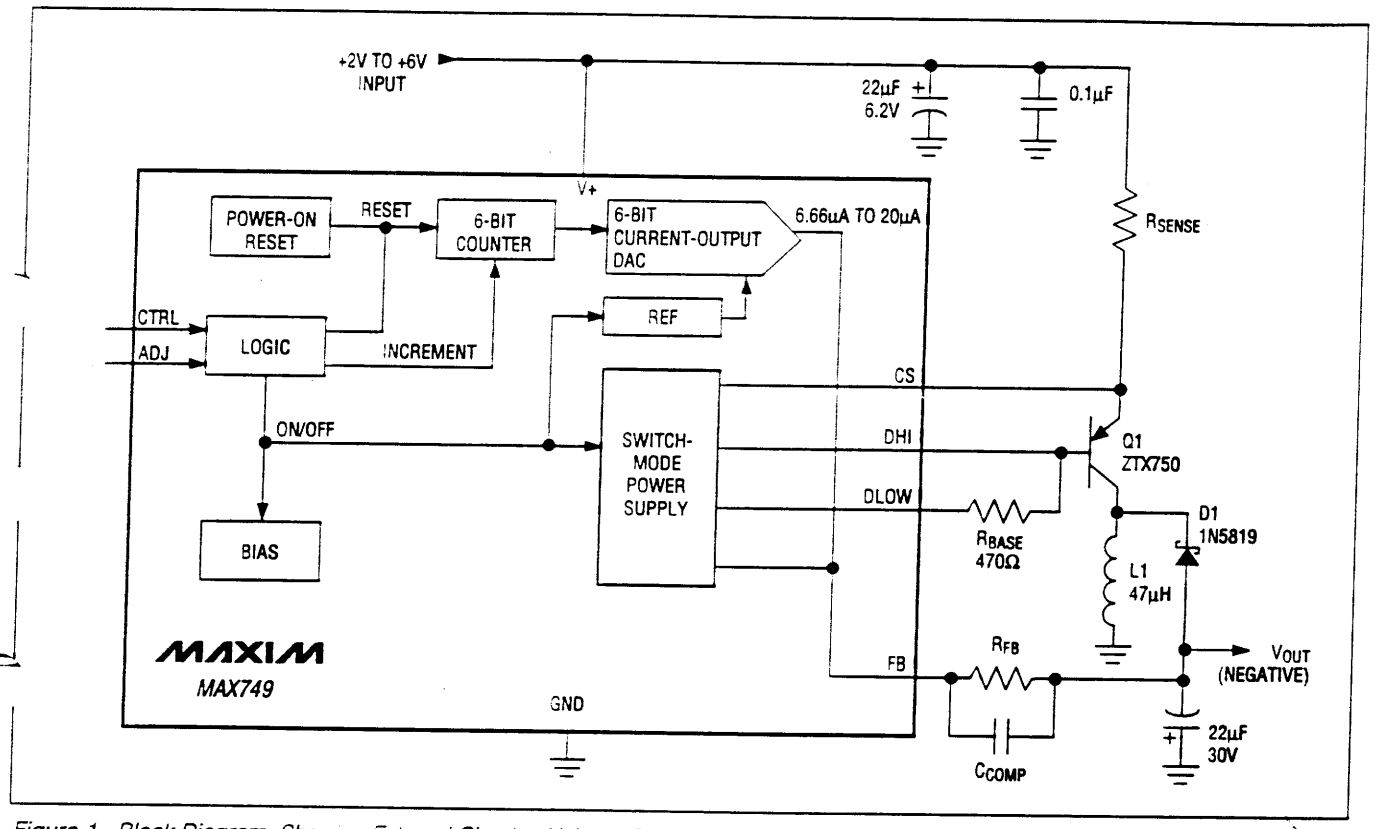


Figure 1. Block Diagram, Showing External Circuitry Using a PNP Transistor

Detailed Description

The MAX749 is a negative-output inverting power controller that can drive an external PNP transistor or P-channel MOSFET. An external resistor and an internal DAC control the output voltage (Figure 1).

The MAX749 is designed to operate from 2V to 6V inputs, ideal for operation from low-voltage batteries. In systems with higher-voltage batteries, such as notebook computers, the MAX749 may also be operated from the regulated +5V supply. A high-efficiency +5V regulator, such as the MAX782, is an ideal source for the MAX749. In this example, the MAX749 efficiency (80%) is compounded with the MAX782 efficiency (95%): 80% x 95% = 76%, which is still high.

Operating Principle

The MAX749 and the external components shown in the *Typical Operating Circuit* form a flyback converter. When the external transistor is on, current flows through the current-sense resistor, the transistor, and the coil. Energy is stored in the core of the coil during this phase, and the diode does not conduct. When the transistor turns off, current flows from the output through the diode and the coil, driving the output negative. Feedback control adjusts the external transistor's timing to provide a regulated negative output voltage.

The MAX749's unique control scheme combines the ultra-low supply current of pulse-skipping, pulse-frequency modulation (PFM) converters with the high full-load efficiency characteristic of pulse-width modulation (PWM) converters. This control scheme allows the device to achieve high efficiency over a wide range of loads. The current-sense function and high operating frequency allow the use of tiny external components.

Switching control is accomplished through the combination of a current limit in the switch plus on- and off-time limits (Figure 2).

Once turned on, the transistor stays on until either:

- the maximum on-time one-shot turns it off (8µs later), or
- the switch current reaches its limit (as determined by the current-sense resistor and the current comparator).

Digitally Adjustable LCD Bias Supply

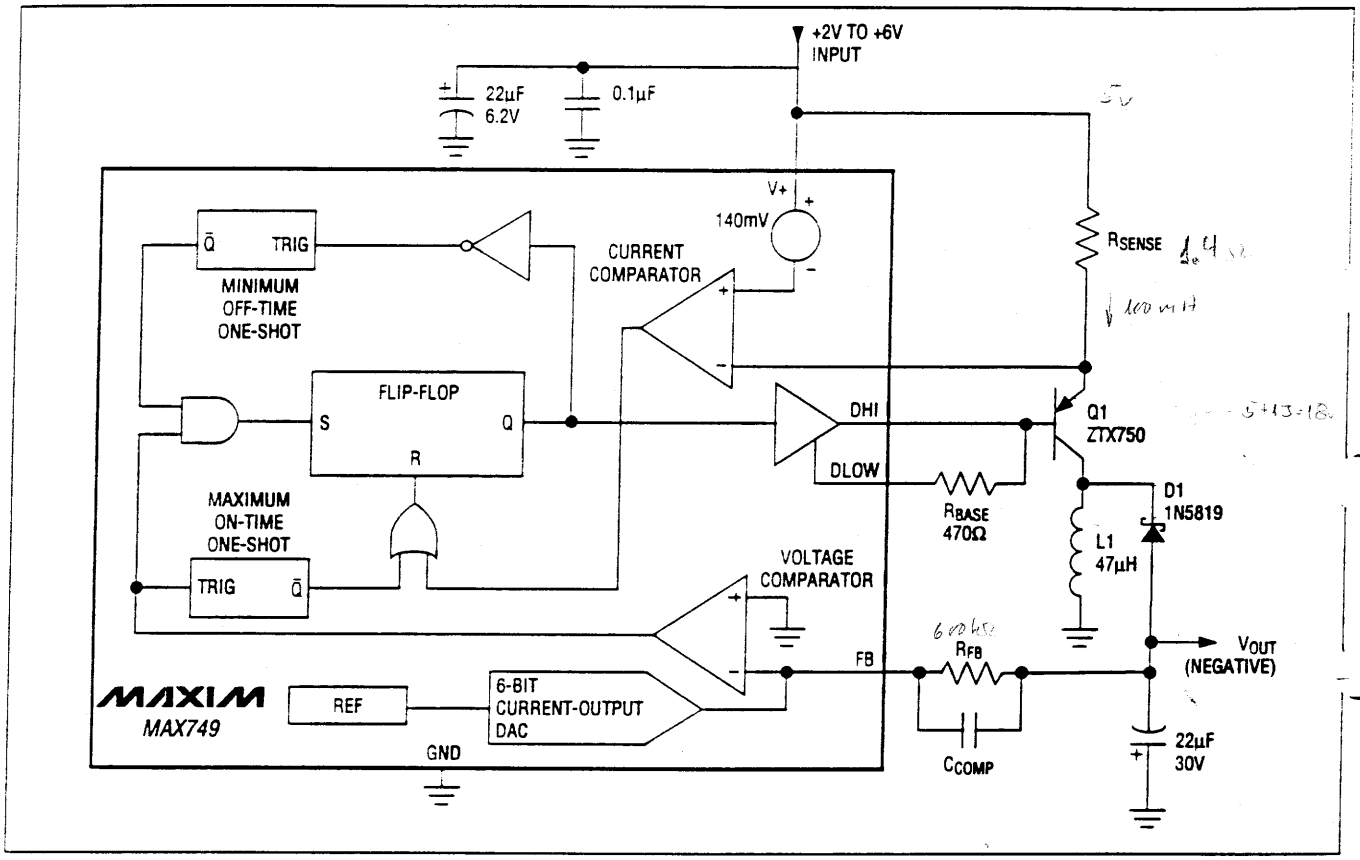


Figure 2. Switch-Mode Power-Supply Section Block Diagram

Once turned off, a one-shot holds the switch off for a minimum of 1µs, and the switch either stays off (if the output is in regulation), or turns on again (if the output is out of regulation).

With light loads, the transistor switches for one or more cycles and then turns off, much like a traditional PFM converter. With heavy loads, the transistor stays on until the switch current reaches the current limit; it then shuts off for 1µs, and immediately turns on again until the next time the switch current reaches its limit. This cycle repeats until the output is in regulation.

Output Voltage Control

The output voltage is set using a single external resistor and the internal current-output DAC (Figure 1). The full-scale output voltage is set by selecting the feedback resistor, R_{FB}. The output voltage is controlled from 33% to 100% of the full-scale output by an internal 64-step DAC/counter.

On power-up or after a reset, the counter sets the DAC output to mid-range. Each rising edge of ADJ increments the DAC output. When incremented beyond full scale, the counter rolls over and sets the DAC to the minimum value. In this way, a single pulse applied to ADJ increases the DAC set point by one step, and 63 pulses decrease the set point by one step.

Table 1 is the logic table for the CTRL and ADJ inputs, which control the internal DAC and counter. Figures 3-7 show various timing specifications and different ways of incrementing and resetting the DAC, and of placing it in the low-power standby mode. As long as the timing specifications for ADJ and CTRL are observed, any sequence of operations can be implemented.

Table 1. Input Truth Table

ADJ	CTRL	RESULT
Low	Low	Shut down
High	Low	Reset counter to mid-range. The device is not shut down.
X	High	On
	High	Increment the counter

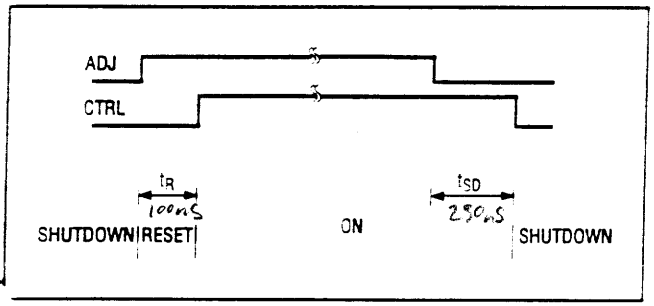


Figure 3. Shutdown-Reset-On-Shutdown Sequence of Operation. The device is not shut down during reset.

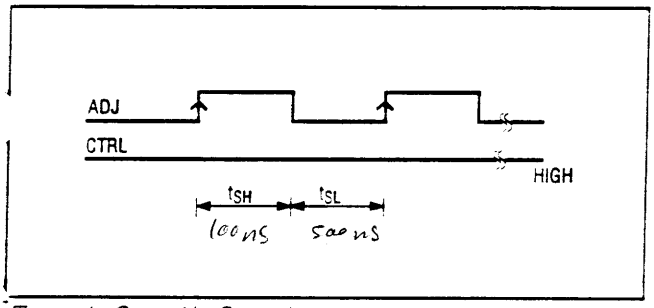


Figure 4. Count-Up Operation

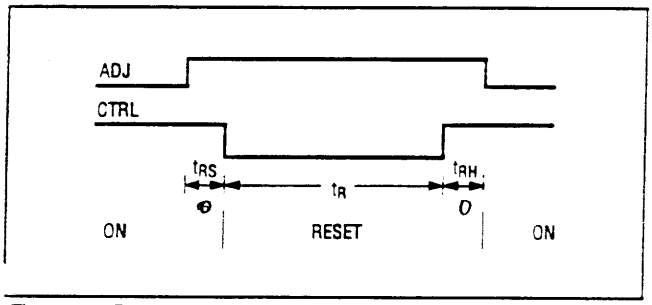


Figure 5. Reset Sequence without Shutdown. The device is not shut down during reset.

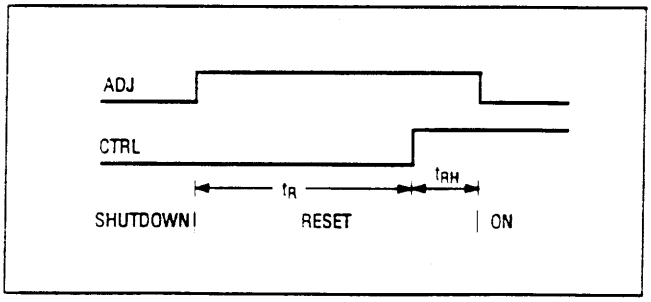


Figure 6. Reset Sequence with Shutdown

In Figure 3, the MAX749 is reset when it is taken out of shutdown, which sets the output at mid-scale. Figure 4 shows how to increment the counter. Figure 5 illustrates a reset without shutting the device down.

Figure 7 provides an example of a sequence of operations: Starting from shutdown, the device is turned on, incremented, reset to mid-scale without being shut down, incremented again, and finally shut down.

Shutdown Mode

When CTRL and ADJ are both low, the MAX749 is shut down (Table 1): The internal reference and biasing circuitry turn off, the output voltage drops to zero, and the supply current drops to 15µA. The MAX749 retains its DAC setting, simplifying software control.

Reset Mode

If ADJ is high when CTRL is low, the DAC set point is reset to mid-scale and the MAX749 is not shut down. Mid-scale is 32 steps from the minimum, 31 steps from the maximum.

Design Procedure and Component Selection

Setting the Output Voltage

The MAX749's output voltage is set using an external resistor and the internal current-output DAC. The full-scale output voltage is set by selecting the feedback resistor R_{FB} according to the formula:

$$-V_{OUT(MAX)} = R_{FB} \times 20\mu A \text{ (Figure 1).}$$

The device is in regulation when $V_{FB} = 0V$.

DAC Adjustment

On power-up or after a reset, the counter sets the DAC output to mid-range, and $-V_{OUT} = R_{FB} \times 13.33\mu A$. Each rising edge of ADJ increments the counter (and therefore the DAC output) in the direction of $-V_{OUT(MAX)}$ by one count. When incremented beyond $-V_{OUT(MAX)}$, the

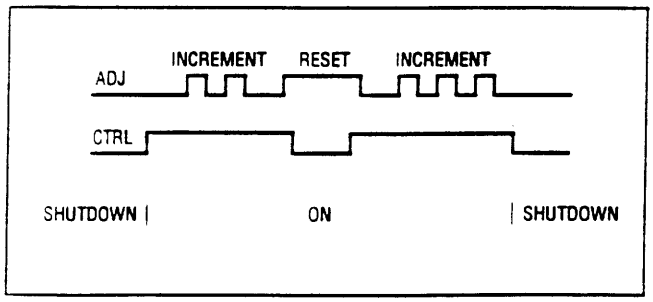


Figure 7. Control Sequence Example (see Output Voltage Control section)

QM 5012-1 Vareidentifikation

1. **Dato:** 12-01-1995

2. **Rekvirent:** plk

3. **Varebetegnelse:** AMP106138-2

4. **Indgår varen i produktionsgrundlag (ja/nej):** Ja
   Får varen salgsvarenummer (ja/nej): Nej

5. **Råvarenummer:**

6. **Varekategori:** Mærkevare

   Dokumentreference:

7. **Beskrivelse:** Connector Z-Pack 125 pos. male shroud

8. **Leverandører:**

   Leverandør 1: AMP Danmark

   Fabrikat: AMP Anslået pris: 5 kr
   Bestillingsnummer: 106138-2

   Leverandør 2:

   Fabrikat: Anslået pris:
   Bestillingsnummer:

   Leverandør 3:

   Fabrikat: Anslået pris:
   Bestillingsnummer:

9. **Godkendelsespligtig komponent i henhold til typegodkendende myndighed (ja/nej):** Nej

   Norm/standard:

10. **Bemærkninger:**

QM 5012-1 Vareidentifikation

1. **Dato:** 12-01-1995

2. **Rekvirent:** plk

3. **Varebetegnelse:** AMP188508-1

4. **Indgår varen i produktionsgrundlag (ja/nej):** Ja
   Får varen salgsvarenummer (ja/nej): Nej

5. **Råvarenummer:**

6. **Varekategori:** Mærkevare

   Dokumentreference:

7. **Beskrivelse:** Connector Z-Pack 125 pos. male feedthrough, level 3.

8. **Leverandører:**

   Leverandør 1: AMP Danmark

   Fabrikat: AMP Anslået pris: 45 kr
   Bestillingsnummer: 188508-1

   Leverandør 2:

   Fabrikat: Anslået pris:
   Bestillingsnummer:

   Leverandør 3:

   Fabrikat: Anslået pris:
   Bestillingsnummer:

9. **Godkendelsespligtig komponent i henhold til typegodkendende myndighed (ja/nej):** Nej

   Norm/standard:

10. **Bemærkninger:** 16 ugers lev. tid, min 198 stk.

QM 5012-1 Vareidentifikation

1. **Dato:** 30.08.94

2. **Rekvirent:** ap

3. **Varebetegnelse:** Graphic LCD display f. SPC/3

4. **Indgår varen i produktionsgrundlag (ja/nej):** ja
   Får varen salgsvarenummer (ja/nej): nej

5. **Råvarenummer:** 99030701

6. **Varekategori:** Mærkevare

   Dokumentreference:

7. **Beskrivelse:** LCD display, 128 x 64 dot, graphic
   Seiko Instruments.

8. **Leverandører:**

   Leverandør 1: E.V.J. Elektronik A/S
   31 839022

   Fabrikat: Seiko Instruments Anslået pris:
   Bestillingsnummer: Seiko G1216B1N000

   Leverandør 2:

   Fabrikat: Anslået pris:
   Bestillingsnummer:

   Leverandør 3:

   Fabrikat: Anslået pris:
   Bestillingsnummer:

9. **Godkendelsespligtig komponent i henhold til typegodkendende myndighed (ja/nej):** nej

   Norm/standard:

10. **Bemærkninger:**

1/9.94 MLO

Kopi

el sendt til mig

QM 5012-1 Vareidentifikation

1. **Dato:** 30.08.94

2. **Rekvirent:** ap

3. **Varebetegnelse:** Graphic LCD display f. SPC/3

4. **Indgår varen i produktionsgrundlag (ja/nej):** ja
   Får varen salgsvarenummer (ja/nej): nej

5. **Råvarenummer:**

6. **Varekategori:** Mærkevare

   Dokumentreference:

7. **Beskrivelse:** LCD display, 128 x 64 dot, graphic
   Seiko Instruments.

8. **Leverandører:**

   Leverandør 1: E.V.J. Elektronik A/S
   31 839022

   Fabrikat: Seiko Instruments Anslået pris:
   Bestillingsnummer: Seiko G1216B1N000

   Leverandør 2:

   Fabrikat: Anslået pris:
   Bestillingsnummer:

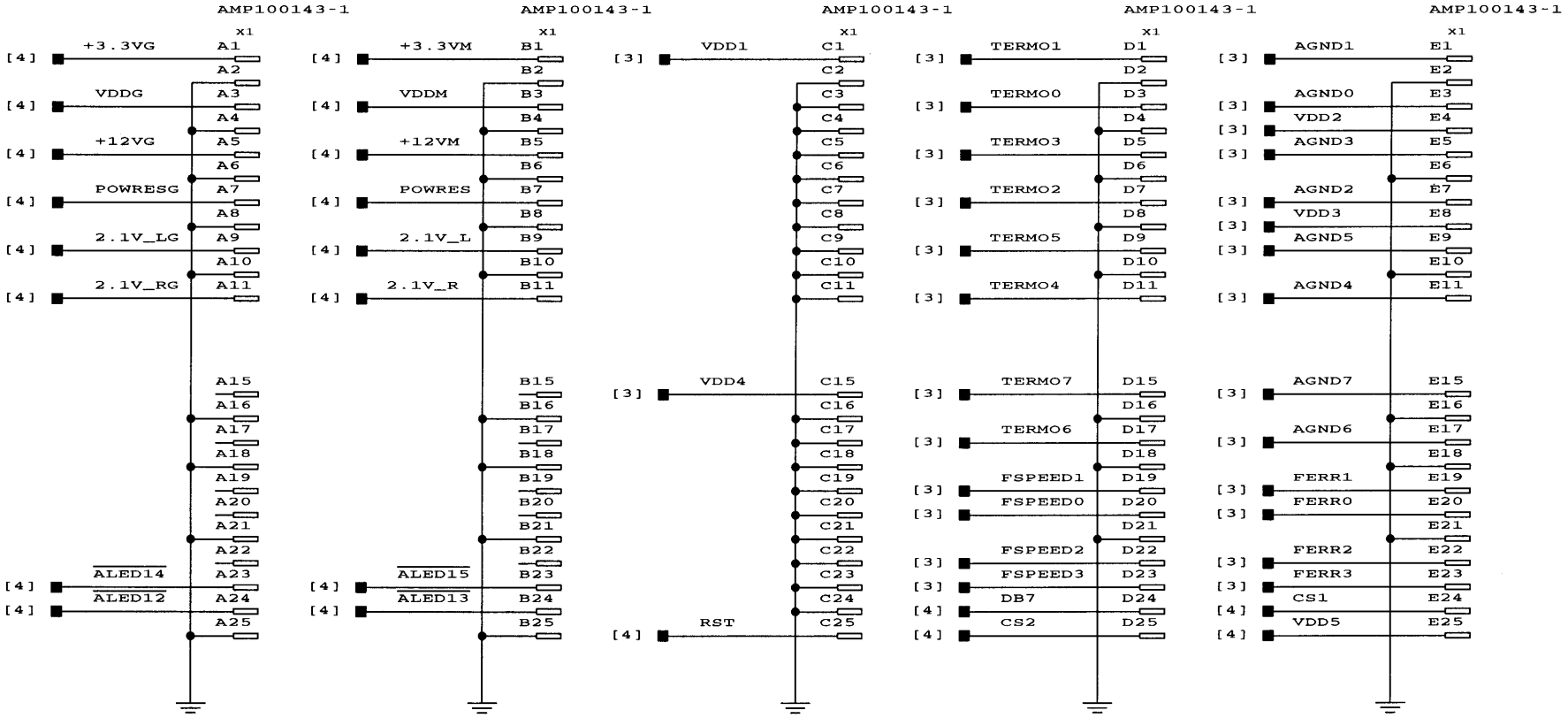
   Leverandør 3:

   Fabrikat: Anslået pris:
   Bestillingsnummer:

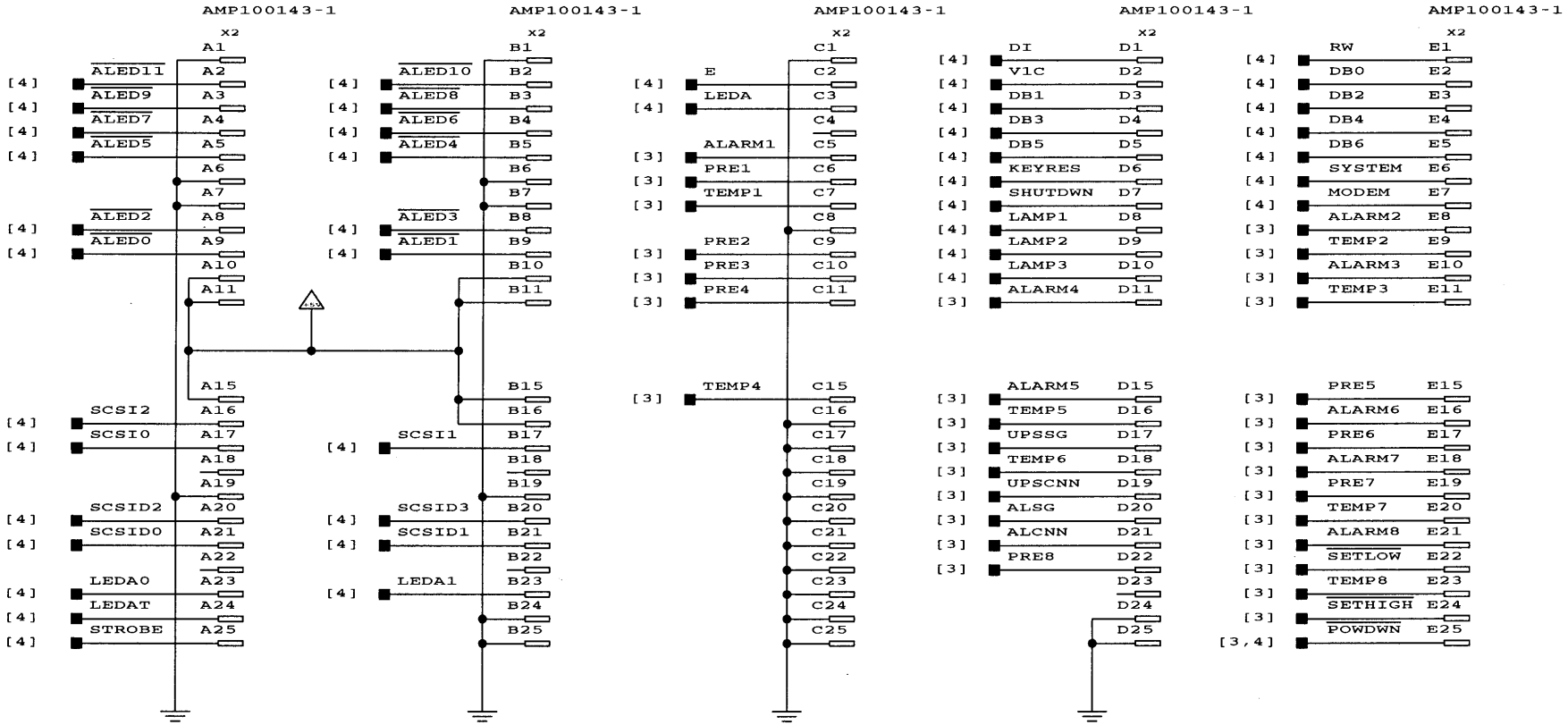
9. **Godkendelsespligtig komponent i henhold til typegodkendende myndighed (ja/nej):** nej

   Norm/standard:

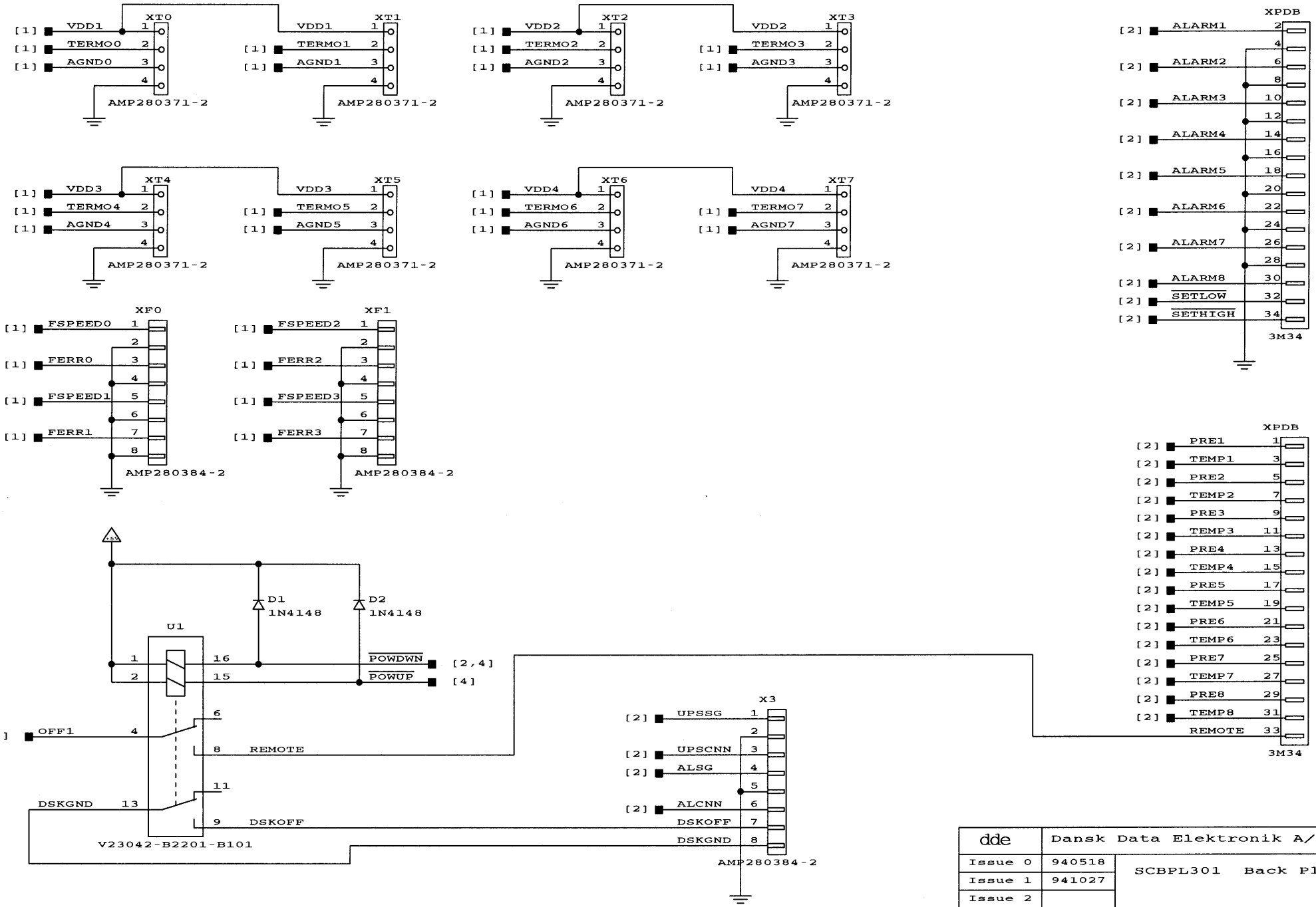
10. **Bemærkninger:**



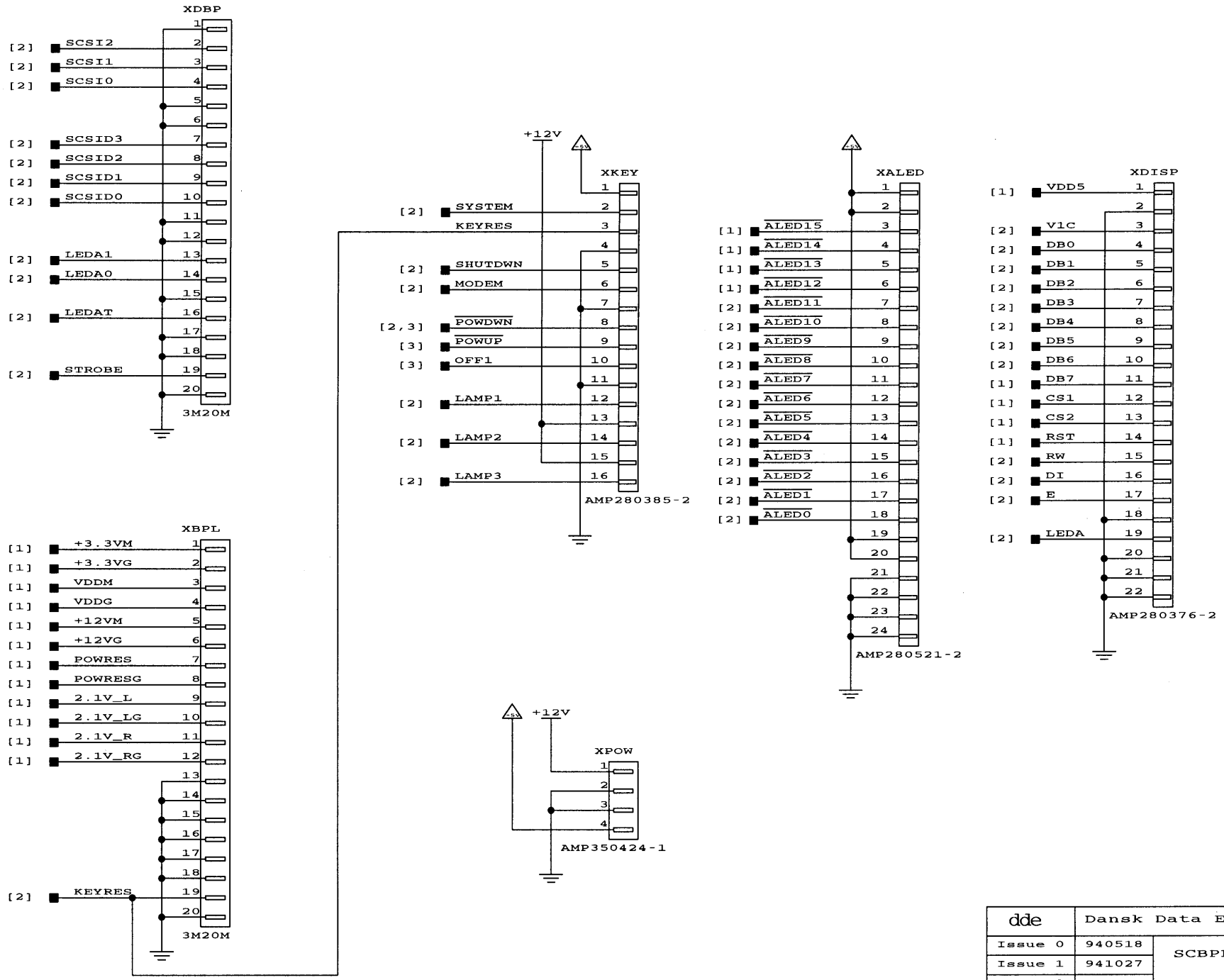
dde	Dansk Data Elektronik A/S		
Issue 0	940518	SCBPL301 Back Plane	
Issue 1	941027		
Issue 2			
Issue 3		File: scbpl	Page: 1 of 4



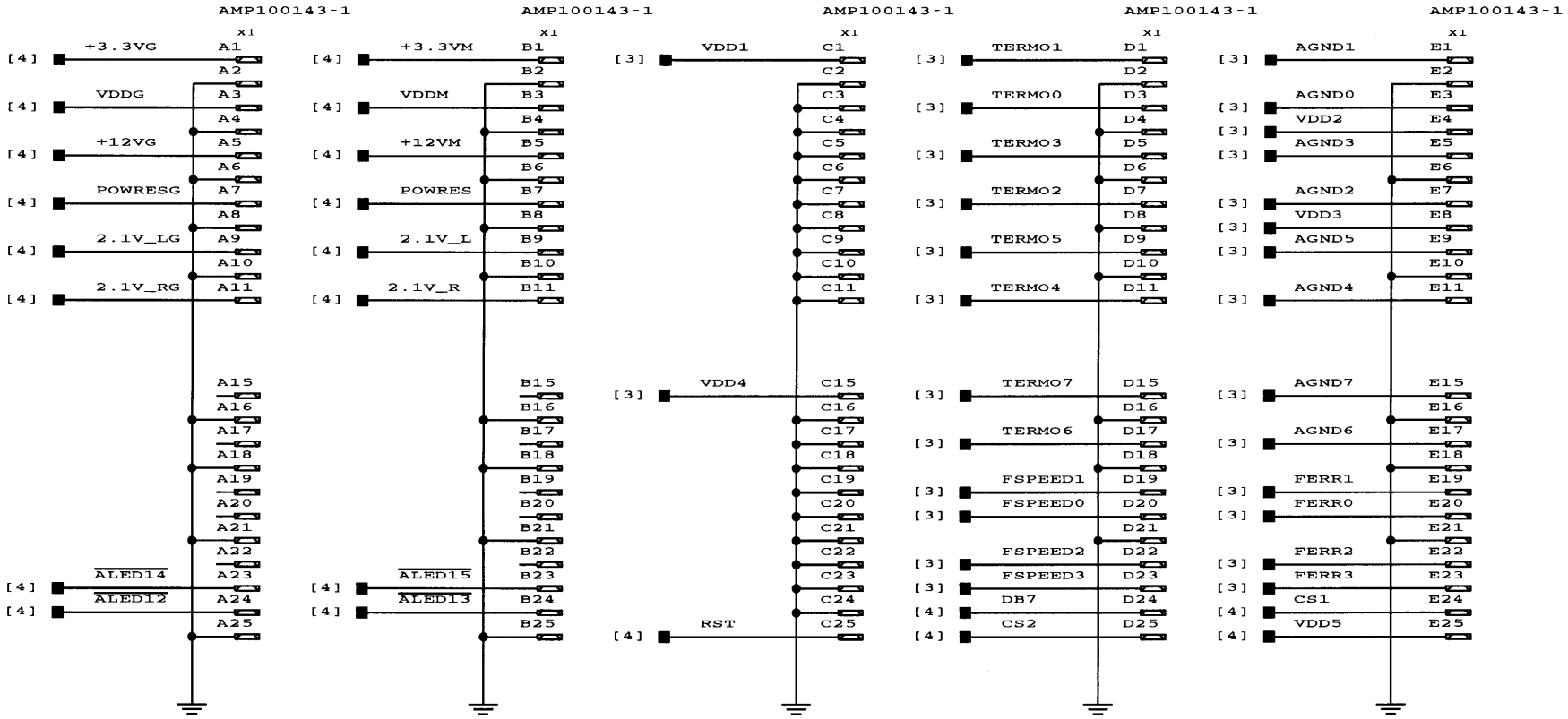
dde	Dansk Data Elektronik A/S		
Issue 0	940518	SCBPL301 Back Plane	
Issue 1	941027		
Issue 2			
Issue 3		File: scbpl	Page: 2 of 4



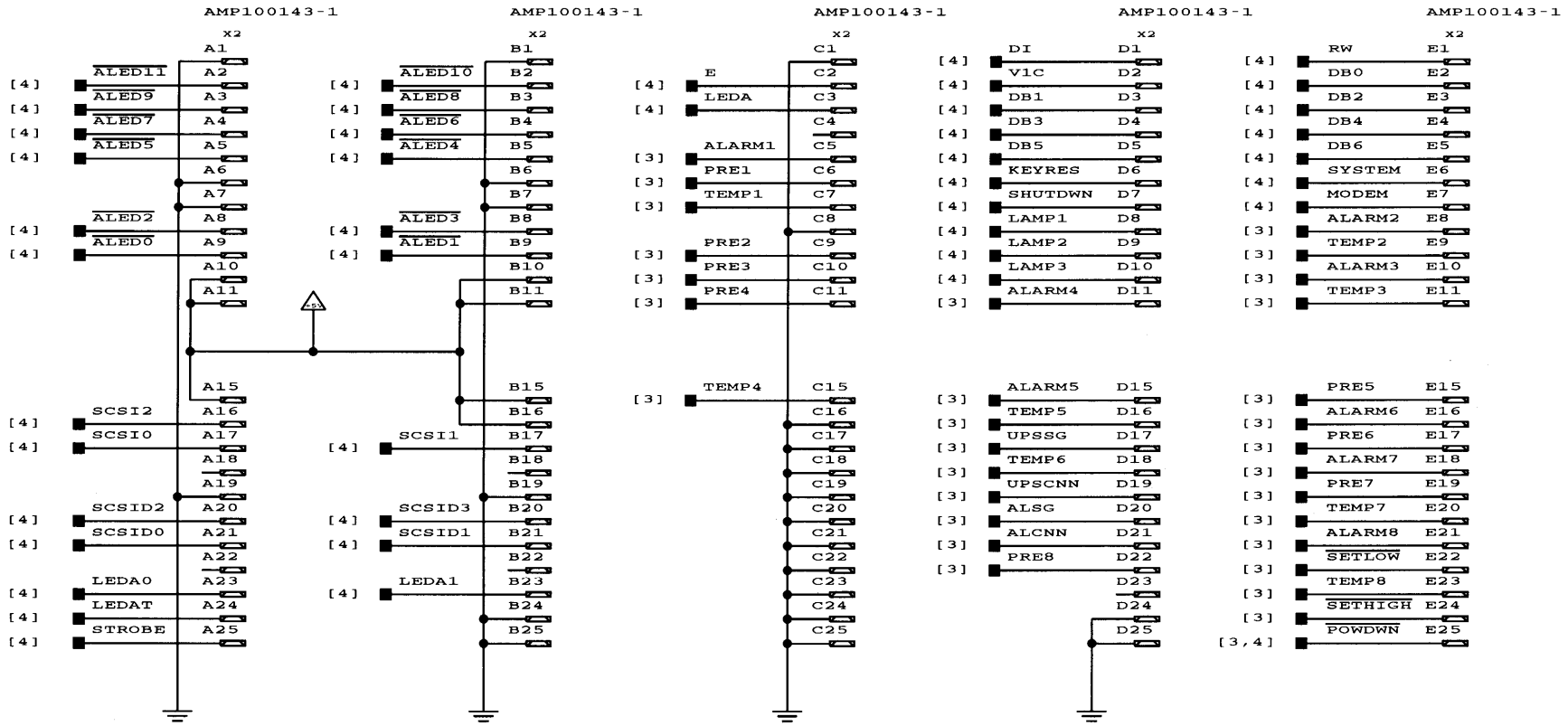
dde	Dansk Data Elektronik A/S		
Issue 0	940518	SCBPL301 Back Plane	
Issue 1	941027		
Issue 2			
Issue 3		File: scbpl	Page: 3 of 4



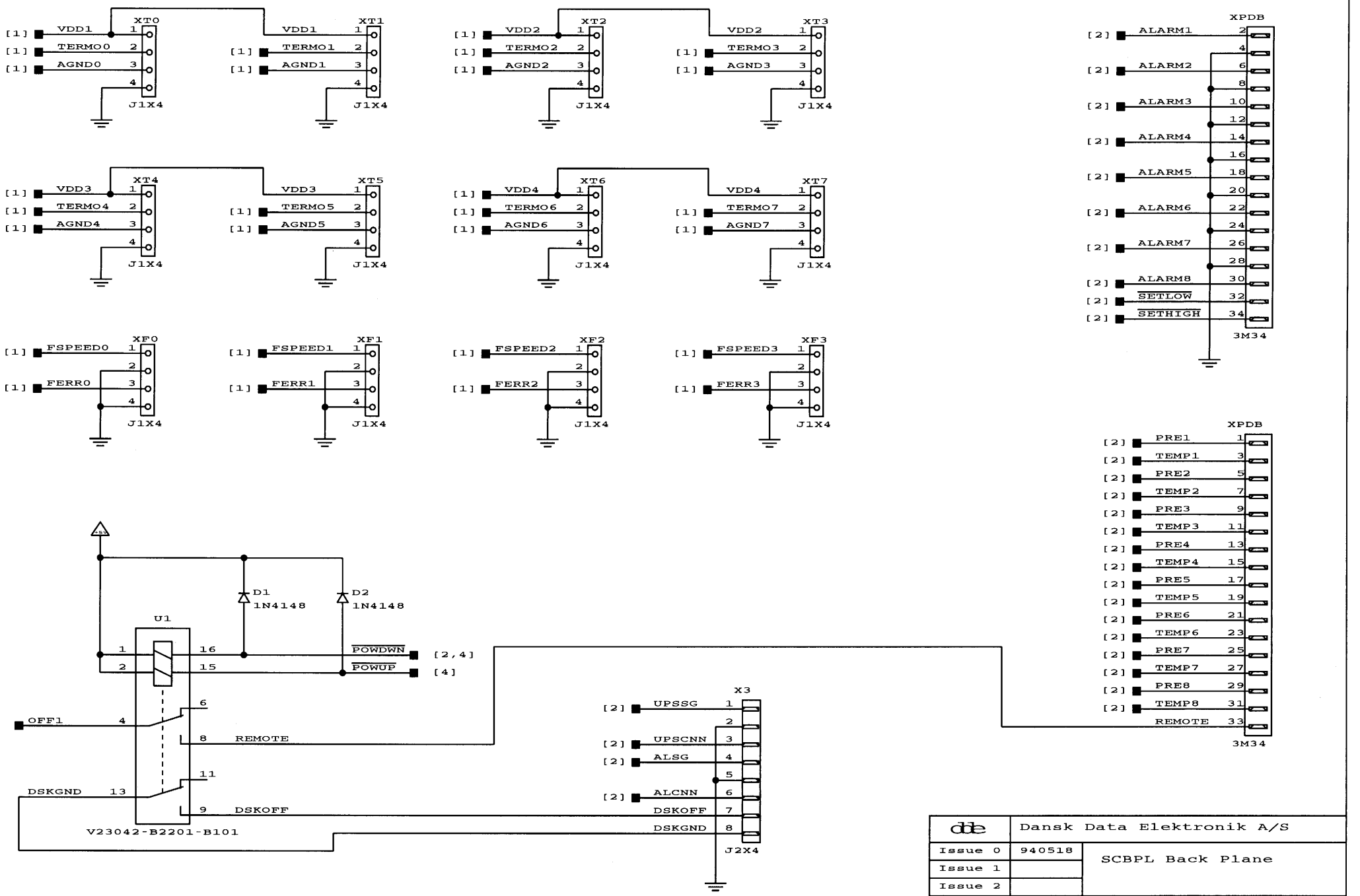
dde	Dansk Data Elektronik A/S		
Issue 0	940518	SCBPL301 Back Plane	
Issue 1	941027		
Issue 2			
Issue 3		File: scbpl	Page: 4 of 4



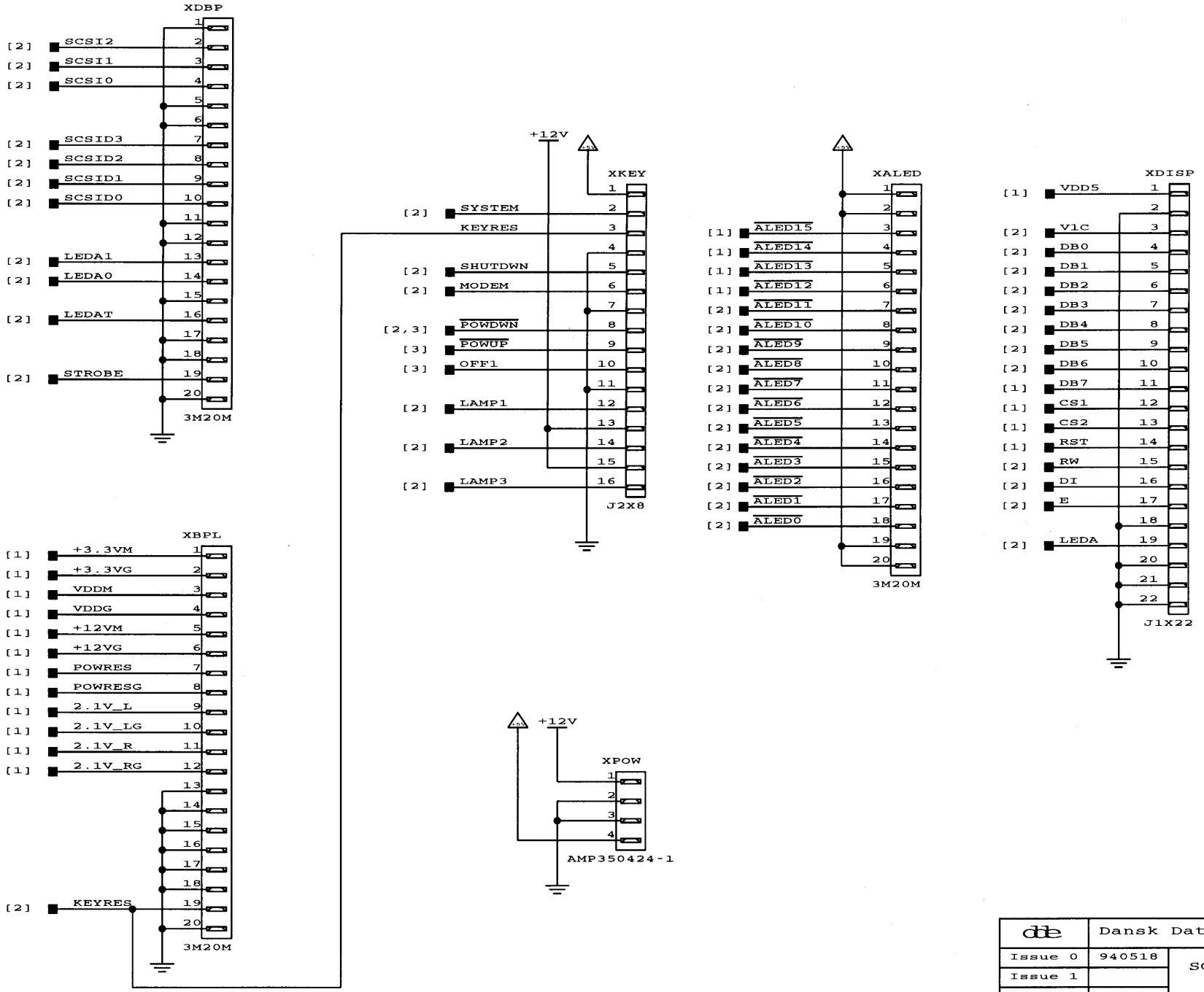
dde	Dansk Data Elektronik A/S		
Issue 0	940518	SCBPL Back Plane	
Issue 1			
Issue 2			
Issue 3		File: scbpl	Page: 1 of 4



dde	Dansk Data Elektronik A/S		
Issue 0	940518	SCBPL Back Plane	
Issue 1			
Issue 2			
Issue 3		File: scbpl	Page: 2 of 4



dde	Dansk Data Elektronik A/S		
Issue 0	940518	SCBPL Back Plane	
Issue 1			
Issue 2			
Issue 3		File: scbpl	Page: 3 of 4



dde	Dansk Data Elektronik A/S		
Issue 0	940518	SCBPL Back Plane	
Issue 1			
Issue 2			
Issue 3		File: scbpl	Page: 4 of 4

PARTS LIST

Module: scbpl301 **Issue:** 0 **Date:** 940531

Part no	Device	Qty	Comp
99.021.208	1N4148	2	D1-2
99.041.405	V23042-B2201-B101	1	U1
99.040.228	3M20M	3	XALED,XBPL,XDBP
VI	J1X22	1	XDISP, AMP MODUII 0-280376-2
99.040.620	J1X4	12	XF0-3,XT0-7
99.040.622	J2X8	1	XKEY
99.040.229	3M34	1	XPDB
99.040.118	AMP350424-1	1	XPOW
99.040.109	AMP100143-1	2	X1-2
99.040.621	J2X4	1	X3

6. Pin list.

First connector.

A1 : VSBAD0	B1 : 3.3 V	C1 : VSBAD1	D1 : DEBUG_R <
A2 : VSBAD2	B2 : BE/ADP0	C2 : VSBAD3	D2 : *GA0
A3 : GND	B3 : BE/ADP1	C3 : GND	D3 : *GA1
A4 : VSBAD4	B4 : GND	C4 : VSBAD5	D4 : 12 V
A5 : VSBAD6	B5 : BE/ADP2	C5 : VSBAD7	D5 : *RESET
A6 : GND	B6 : BE/ADP3	C6 : GND	D6 : 3.3 V
A7 : VSBAD8	B7 : GND	C7 : VSBAD9	D7 : BAIOBIG
A8 : VSBAD10	B8 : *IRQ	C8 : VSBAD11	D8 : *PAS
A9 : GND	B9 : GND	C9 : GND	D9 : GND
A10: VSBAD12	B10: 3.3 V	C10: VSBAD13	D10: DEBUG_S <
A11: VSBAD14	B11: VCC	C11: VSBAD15	D11: VCC
A12: GND	B12: VCC	C12: GND	D12: VCC

Second connector.

A1 : VSBAD16	B1 : VCC	C1 : VSBAD17	D1 : VCC
A2 : VSBAD18	B2 : VCC	C2 : VSBAD19	D2 : VCC
A3 : GND	B3 : GND	C2 : GND	D3 : GND
A4 : VSBAD20	B4 : *DS	C4 : VSBAD21	D4 : *ACK
A5 : BSVAD22	B5 : *WR	C5 : VSBAD23	D5 : AC
A6 : GND	B6 : GND	C6 : GND	D6 : GND
A7 : VSBAD24	B7 : *BREQ	C7 : VSBAD25	D7 : *ASACK
A8 : VSBAD26	B8 : GND	C8 : VSBAD27	D8 : *BUSY
A9 : GND	B9 : *SCU <	C9 : GND	D9 : 3.3 V
A10: VSBAD28	B10: *SCRST<	C10: VSBAD29	D10: reserved <
A11: VSBAD30	B11: GND	C11: VSBAD31	D11: GND
A12: GND	B12: *BGIN	C12: GND	D12: *BGOUT

Geographic addresses:	GA1	GA0	Notice: active high signals.
Submodule Connector 1:	low	high	
Submodule Connector 2:	high	low	
Submodule Connector 3:	high	high	

940725

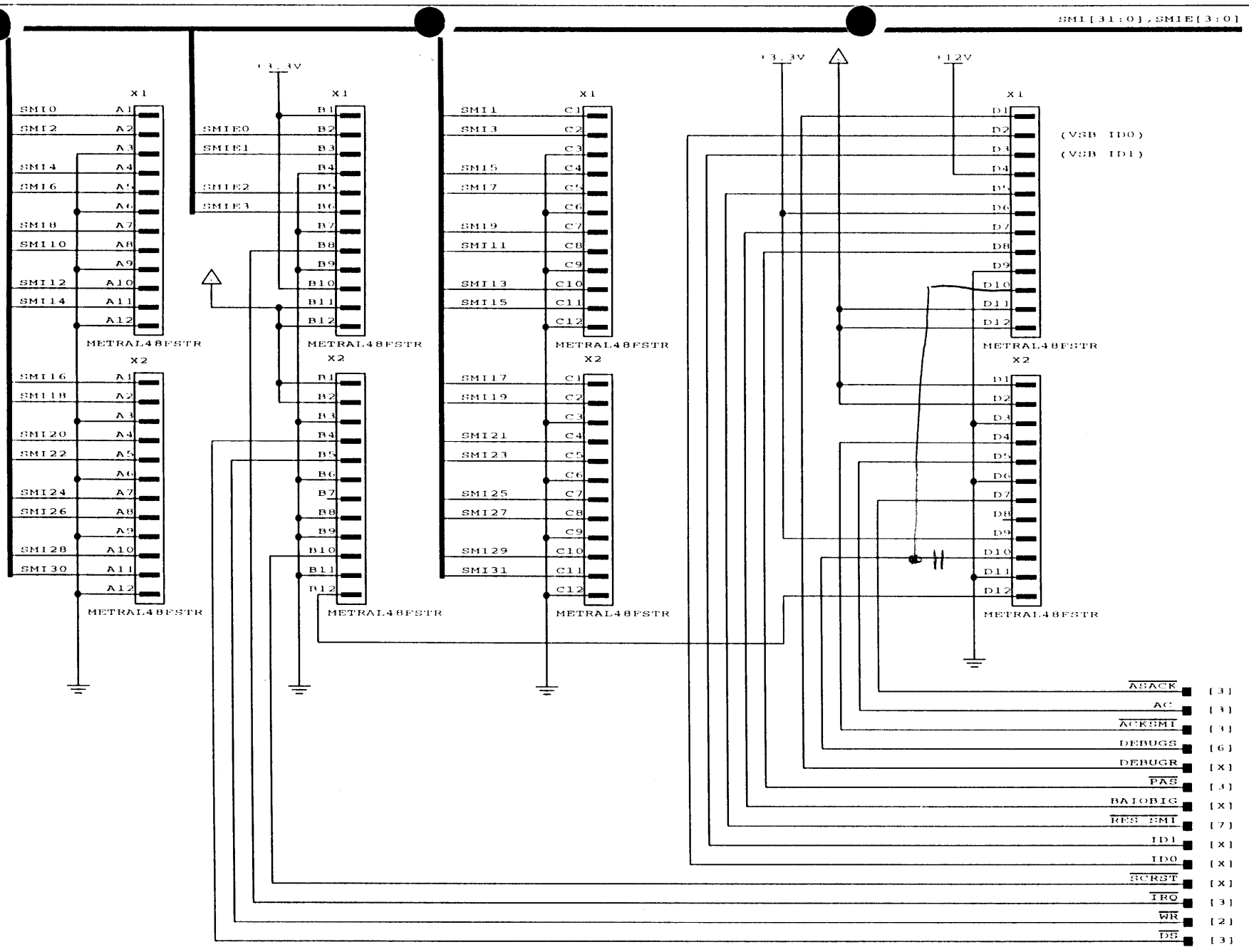
Hej Knud Arne

OK havde desværre ret m.h.t. Debs

Det kommer ikke ud på bussen. Se venligst vedlagte

mvh

PLK



dde	Dansk Data Elektronik A/S	
Issue 0	940519	Service Computer SCU302
Issue 1		SMI Interface
Issue 2		
Issue 3		File: scu Page: 1 of 12

kan 940711

Ved indkøring af SCU303 er der fundet følgneden fejl/problemer;

1. Polariteten af DIR signal til 74645 mellem DAT og DB, databus til display er forkert. !rp er erstattet af read.

2. Der mangler forbindelse mellem analog og digital jord. Denne er etableret på AD-converter.

3. 1.5 ohms modstand i DC-DC converter er erstattet med 1.5 kohm. Produktionsfejl.

4. Blæserstyring skal sikkert resettes af time-out fra watch-dog.

5. Det skal sikres at DC-DC converter maksimalt kan give -12 volt. Zener-diode ?

√6. Data bit 5 mangler i NVRAM.

√7. DEBUGS skal forbindes til X1 D10.

Frigivelse
af
Programmerbare Komponenter

Modulnavn: SCU302-1. FCN154

Dato: 951002

Initialer: kan

Filnavn	Checksum	Label	Type	Råvarenummer
scu201.jed	000347EA	SCU201	MACH210-15	99.010.905

```
mailx -s"FCN154 til SCU302-1" kkj <send1
```

Til: Produktionen
Fra: Knud Arne Nielsen
Dato: 951002
Ang.: FCN154 til SCU302-1

Sendt elektronisk via tftp

```
tftp -i paxmax put ./cntsty8.abl /tmp/cntsty8.abl      Testvektorer
tftp -i paxmax put ./cntsty9.abl /tmp/cntsty9.abl      Testvektorer
tftp -i paxmax put ./scu201.abl /tmp/scu201.abl        Kildefil
tftp -i paxmax put ./scu201.jed /tmp/scu201.jed        JEDEC
tftp -i paxmax put ./scu302-1.fri1 /tmp/scu302-1.fri1 Frigivelse
```

FCN154 følger på papir. Jens Jørgen er orienteret.

Supermax Field Change Notice no. 160

Module:	SCU302 Service Computer submodule.
Date:	96-04-11

Category:

- Production change.
- In the field: When error occurs or when a TPE301 submodule is installed.

Corrects the error:

Caused by a glitch on the trailing edge of the signal PAS, SCU302 interprets data on the bus as a new address and starts a new cycle. This faulty cycle may cause the submodule bus to lock up or may cause a parity error in a following read cycle to SCU302.
Symptoms: The submodule interface stops. System crash caused by time out on BAIO30x. Parity error.

Needed tools:

68 pin PLCC extraction tool.

Supermax FCN kit 160, stock number 95101600, consisting of:

1. One MACH labeled "**SCU101**".
2. One PROM labeled "**FCN160**".

Description:

- All previous FCNs for this module must be made.
- Replace MACH in position U6 labeled "**SCU100**" with MACH labeled "**SCU101**".
- Replace FCN PROM in position U10 labeled "**FCN154**" with new FCN PROM labeled "**FCN160**".
- Do not update the revision field.

  The field should be **REV:ABCDEFGHIJK**

Circuits involved:

Name	Type
U6	MACH220-15JC
U10	AM27519AJC

Previous FCN:

FCN154.

Frigivelse
af
Programmerbare Komponenter

Modulnavn: SCU302-1. FCN160

Dato: 960411

Initialer: kan

Filnavn	Checksum	Label	Pos.	Type	Råvarenummer
scu101.jed	0004726E	SCU101	U6	MACH220-15	99010908
scu201.jed	000347EA	SCU201	U36	MACH210-15	99010905

```
mailx -s"FCN160 til SCU302-1" kkj <send2

Til: Produktionen
Fra:Knud Arne Nielsen
Dato:960411
Ang.:FCN160 til SCU302-1
-------------------------------------------------------------------
Sendt elektronisk via tftp

tftp -i paxmax put ./scu101.abl /tmp/scu101.abl       Kildefil
tftp -i paxmax put ./scu101.jed /tmp/scu101.jed       JEDEC
tftp -i paxmax put ./scu302-1.fri2 /tmp/scu302-1.fri2 Frigivelse


FCN160 følger på papir.
```

Tolerance på spændingsmålinger i SCU302

spændingsdeler 1% modstande : ± 2%
zenerdiode (2.4570 - 2.5431) : ± 1.5%
konvertering ± 1 bit : ± 0.5%
± 4%

stabilitet af zenerdiode 5mV ~ 2‰

vedmåling af 2.1V anvendes ikke
spændingsdeler : tol ± 2% ~ ± 50mV

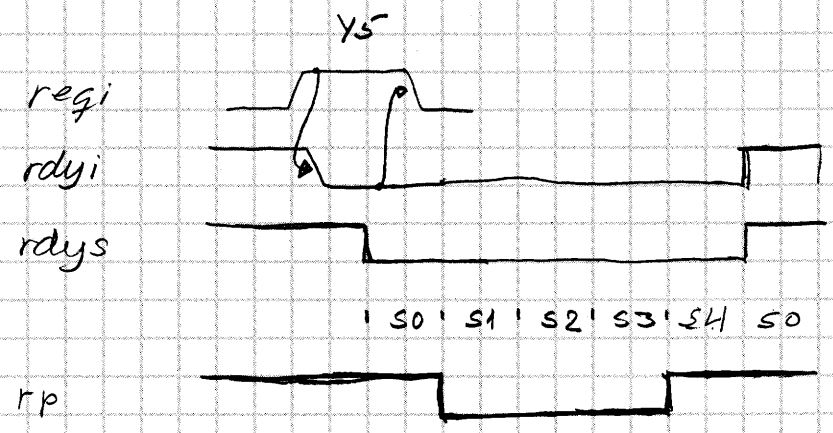
Hvis programmet justerer A/D konverter
reduceres tol. til ca. ± 0.5% svarende
til konvertering.

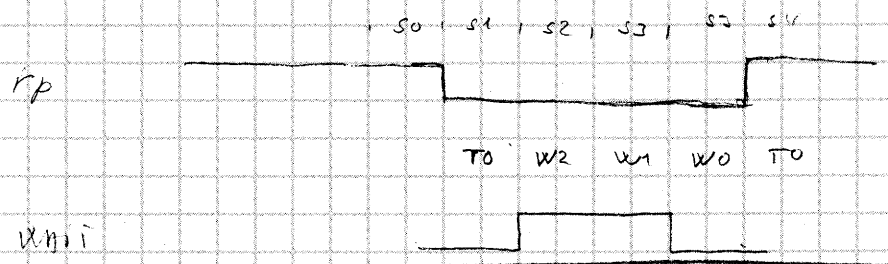
Zenerdiode LM385BYZ-2.5

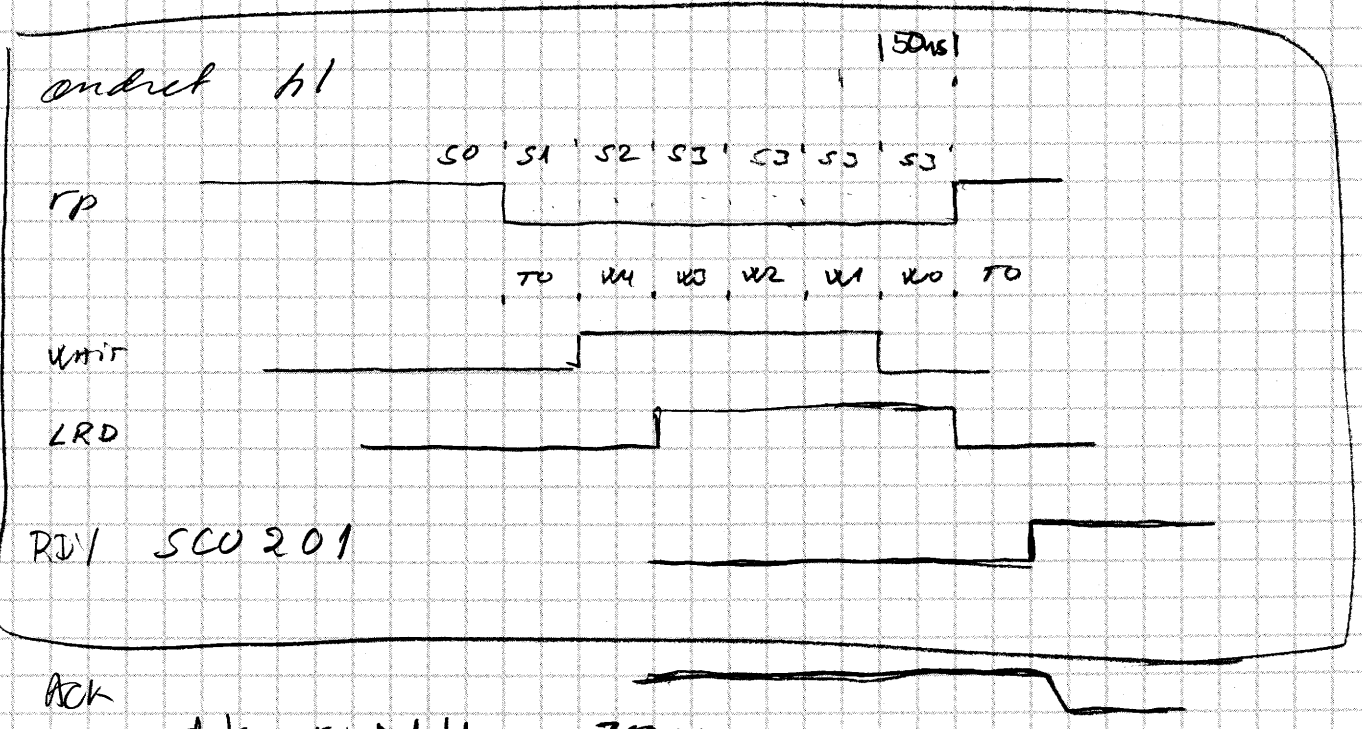
dde Dansk Data Elektronik A/S

SCU 201	Initials HAN	Page 1
	Date 25/4-95	Project SCU302-1

Punktskjved laning af AVC på service computer.
SCU200 HARDWARE:
Chip select til AD converter er gated med RP+WP
og har derfor negativ set up tid!
Håndtes usandt, men skal i modstrid med
datablad.
Endret i SCU201



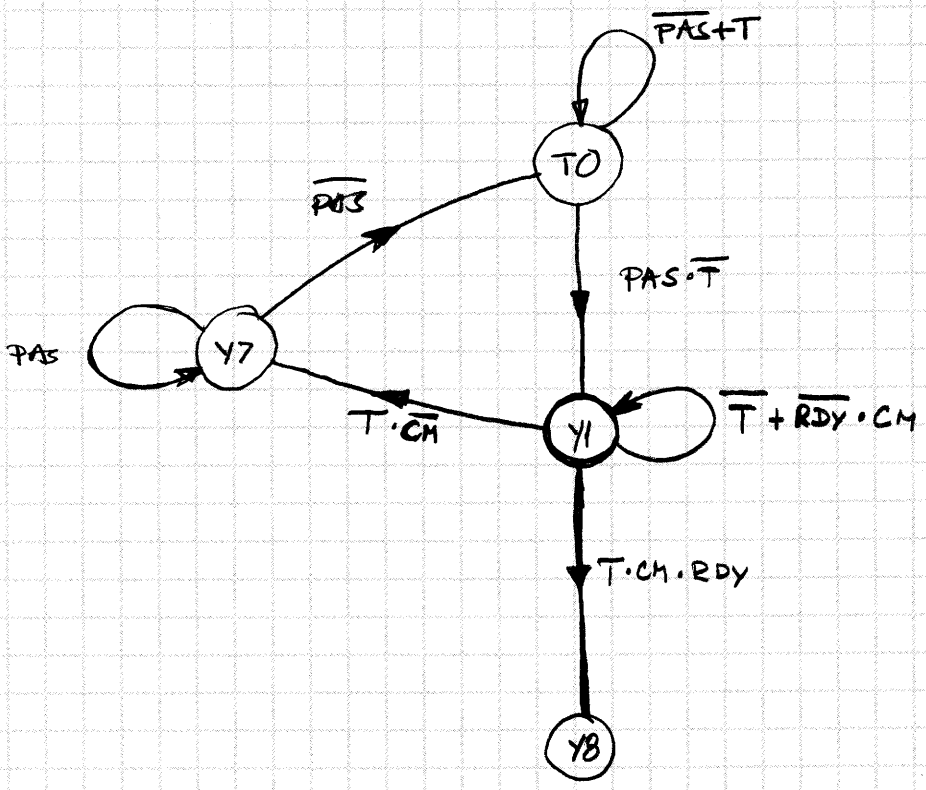




delay rp → latch : 300 ns

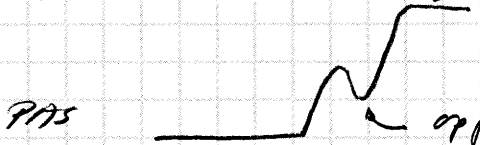
ADC	225	
F280	17	
PAL	15	257 ~~skuffed min 43~~

SCU100



Ved indkøring af TPE301 er der den 28/3-96 fundet følgende problem:

TPE301 laver en access til 3010. Da° bagkanten af denne access, når PAS gaˊ inaktiv, tiler service computer sig adressen og sætter ASACK hvorved det hele gaˊ i stå. Problemet skyldes bureaufarmen paˊ bagkanten af PAS:



opfattes som ny PAS af servicecomputer

Under accessen fra TPE301 befinder SCU100 sig i tilstand Y7. Når PAS forsvinder hopper SCU100 til T0. Her ser SCU100 PAS lav p.g.a refleksion og hopper til Y1. SCU100 venter her til T=1 hvorefter den hopper til Y7 eller Y8 styret af CH uafhængigt af PAS. Hvis CH=1 hænger systemet op. Problemet er lagt i SCU101: SCU101 venter i Y1 til T=1 dernæst hoppes til T0, hvis PAS=0. Hvis PAS=1 hoppes til Y7/Y8 styret af CH.

Kan problemet i SCU100 forklare paritetsfejl ved læsning fra SCU302?

Ved access til SCU302 venter SCU100 i tilstand Y4 til PAS=0, hvorefter den hopper til T0. Situationen er ækvivalent til det tidligere beskrevne hvor SCU100 hopper fra Y7 til T0. Der er altså mulighed for at SCU100 starter igen og hopper til Y1. Hvis CH=1 og RPO=1 vil SCU100 lave en læsning og vente i Y6 på DS. I denne situation vil SCU100 holde BACK aktiveret.

Hvorfor hænger BUS ikke op?

BA10 vil opføre sig på følgende måde:
START vil generere PAS medmindre ASACK.
er aktivt vil BA10 ikke sætte
adresse ud, men gå videre til datafasen
når YH bliver aktiveret.

BA10 vil altså ikke hænge op, men kun
en cycle på VSB-bussen uden at adressen
kommer ud på bussen.

SCU100 venter i Y6 på DS. Når DS kommer
afslutter SCU100 cyclen. med hvilke data.
Adressen er hjælpelig, men stabil, da latch
er lukket. Der udføres en læsning
på SCU302 med læsepuls. Data latches
og leveres til BA10. Hvis der læses et
ulovligt register på SCU302 er panikten
på. Hvis de læses ud i det blå
er de gode muligheder for panikstejl!
Bemærk SCU302 driver kun data(7:0). Bussens indhold
vil derfor aldrig matche adressen for SCU302. Herfra
kræves access til et modul som driver data