PARTS LIST

Part no: ? **Module:** CPU305-1 **Pcb no:** ?
Status: PRELIMINARY **Date:** 96-07-30 **Page:** 1/2

Part no	Device		Qty	Comp
99020850	22NF		315	C1,C10-313,C322-331
99020851	4.7UF/25V		1	C2
99020849	33UF/16V		8	C314-321
99021413	LXH1032ID		5	D1,D5,D9,D13,D17
99021412	LXH1032GD		12	D2-4,D6-8,D10-12,D14-16
99000227	74S1050D		14	D18-31
99022032/27	1.000MHZ		1	OSC1
99020305 2	100R		48	R5,R5A,R6-9,R15,R17,R19,R21,R23 R25,R27,R29,R31-32,R35,R38-39 R41,R45,R60,R65,R68,R108-109 R112-113,R116-117,R120-121 R124-125,R128-129,R132-133 R136-137,R139,R141,R143,R145 R147,R149,R151,R153
99020307	82R		2	R10,R12
99020308	130R		2	R11,R13
99020306	220R		33	R14,R16,R18,R20,R22,R24,R26,R28 R30,R106-107,R110-111,R114-115 R118-119,R122-123,R126-127 R130-131,R134-135,R138,R140 R142,R144,R146,R148,R150,R152
99020303	4K7		7	R33,R42,R66-67,R69-70,R92
99020327	3K		2	R34,R43
99022235	NET4K7		1	R36
99020301	1K		20	R37,R40,R44,R49-57,R62-64 R86-90
99020309	560R		17	R46-48,R72-84,R91
99022234	NET1K		1	R93
99020322	0R		12	R94-105
1K20603.vi	1K2		288	R200-487
99011702	CA303		8	U1-8
99002411	74F1808D		2	U9,U11
99002412	74F1832D		2	U10,U12
99000831	74AS20D		3	U13-14,U16
99002413	74F377D		2	U17,U25
99010406	PALCE22V10H-7JC	S	2	U18-19
99010902	MACH110-20JC	S	4	U20,U78,U85,U94
99010909	MACH210A-10JC	S	2	U22,U24
99002222	74FCT16823BTPV		8	U26-27,U48-49,U81-84
99011223	MCM6706AJ10		20	U28-29,U33-34,U38-39,U43-44 U50-51,U55-56,U60-61,U65-66 U140,U142,U145,U147
99002223	74FCT16244ATPV		14	U30,U35,U40,U45,U52,U57,U62,U67 U77,U89-93
99002221	74FCT521BTSO		18	U31-32,U36-37,U41-42,U46-47 U53-54,U58-59,U63-64,U68-70,U88
99010908	MACH220-15JC	S	1	U71
99012095	82S123AA	S	2	U72-73
99002219	74FCT16501ATPV		1	U74
99002408	74F280BD		3	U75-76,U80
99012114	AM29C040-120JC		1	U79
99010604	PALCE16V8H-7JC	S	2	U86,U95
99010604	PALCE16V8H-7JC_8	S	1	U87
99011218	CY7B991-7JC		1	U96
99005033	MC100H641FN		1	U97
99002500	FB2031B		20	U98-107,U111-120
99002501	FB2033B		6	U108-110,U121-123
99001040	74ALS240D		2	U124-125

Part no	Device		Qty	Comp
99000472	74LS393D		3	U127-129
99002405	74F74D		2	U130-131
99000832	74AS30D		3	U132-134
99010022	PAL16R6-5JC	S	2	U136-137
99010904	MACH210-12JC	S	6	U138-139,U141,U143-144,U146
99002407	74F244D		1	U148
99000830	74AS1004D		1	U149
99040151	3M955515		16	X00-03,X10-13,X20-23,X30-33
99040111	AMP100147-1		3	X101-102,X104
99040110	AMP100145-1		1	X103
99023228	20P SOKKEL		7	U72-73,U86,U95,U87, U136-137
99023229	28P SOKKEL		2	U18-19
99023230	32P SOKKEL		1	U79
99023231	44P SOKKEL		12	U20,U78,U85,U94,U22 U24,U138-139,U141 U143-144,U146
99023232	68P SOKKEL		1	U71
99030332	PRINT CPU301 ISS. 0		1	
990680XX	FORPLADE CPU 305		1	

PLD list

Part no: ???????? **Module:** CPU305-0 **Pcb no:**
Status: Preliminary **Date:** 96-04-17 **Page:** 1 / 1
Init: AAJ

Label	Pos.	Part no.	Type	X-pgm.	File	Checksum
c31171	U18	990	PALCE22V10H-7JC		c31171.jed	9610
c31171	U19	990	PALCE22V10H-7JC		c31171.jed	9610
c31161	U136	99010022	PAL16R6-5JC		c31161.jed	2260
c31161	U137	99010022	PAL16R6-5JC		c31161.jed	2260
c31030	U20	99010902	MACH110-20JC		c31030.jed	B930
c31151	U22	990	MACH210A-10JC		c31151.jed	6FA3
c31151	U24	990	MACH210A-10JC		c31151.jed	6FA3
c31062	U138	99010904	MACH210-12JC		c31062.jed	8C39
c31062	U143	99010904	MACH210-12JC		c31062.jed	8C39
c31071	U139	99010904	MACH210-12JC		c31071.jed	C79D
c31071	U141	99010904	MACH210-12JC		c31071.jed	C79D
c31071	U144	99010904	MACH210-12JC		c31071.jed	C79D
c31071	U146	99010904	MACH210-12JC		c31071.jed	C79D
c31081	U71	99010908	MACH220-15JC		c31081.jed	FE9C
c31091	U78	99010902	MACH110-20JC		c31091.jed	E1C6
c31101	U85	99010902	MACH110-20JC		c31101.jed	5A30
c31101	U94	99010902	MACH110-20JC		c31101.jed	5A30
c31110	U86	99010604	PALCE16V8H-7JC	R6	c31110.jed	234F
c31110	U95	99010604	PALCE16V8H-7JC	R6	c31110.jed	234F
c31120	U87	99010604	PALCE16V8H-7JC	R8	c31120.jed	2470

Directory: nessie: /wd/mudv3/pal/cpu305-0/jed

PLD list

Part no: ???????? **Module:** CPU305-0 **Pcb no:**
Status: Preliminary **Date:** 96-03-14 **Page:** 1 / 1
Init: AAJ

Label	Pos.	Part no.	Type	X-pgm.	File	Checksum
c31170	U18	990	PALCE22V10H-7JC		c31170.jed	745E
c31170	U19	990	PALCE22V10H-7JC		c31170.jed	745E
c31160	U136	99010022	PAL16R6-5JC		c31160.jed	227F
c31160	U137	99010022	PAL16R6-5JC		c31160.jed	227F
c31030	U20	99010902	MACH110-20JC		c31030.jed	B930
c31150	U22	990	MACH210A-10JC		c31150.jed	
c31150	U24	990	MACH210A-10JC		c31150.jed	
c31061	U138	99010904	MACH210-12JC		c31061.jed	807D
c31061	U143	99010904	MACH210-12JC		c31061.jed	807D
c31071	U139	99010904	MACH210-12JC		c31071.jed	C79D
c31071	U141	99010904	MACH210-12JC		c31071.jed	C79D
c31071	U144	99010904	MACH210-12JC		c31071.jed	C79D
c31071	U146	99010904	MACH210-12JC		c31071.jed	C79D
c31080	U71	99010908	MACH220-15JC		c31080.jed	004F
c31090	U78	99010902	MACH110-20JC		c31090.jed	E876
c31100	U85	99010902	MACH110-20JC		c31100.jed	604D
c31100	U94	99010902	MACH110-20JC		c31100.jed	604D
c31110	U86	99010604	PALCE16V8H-7JC	R6	c31110.jed	234F
c31110	U95	99010604	PALCE16V8H-7JC	R6	c31110.jed	234F
c31120	U87	99010604	PALCE16V8H-7JC	R8	c31120.jed	2470

CPU 304 : 1 MByte Cache - Global Bus
CPU 305 : 1 MByte Cache - Global + Local Bus
CPU 306 : 4 - - - Global Bus
CPU 307 : 4 - - - Global + Local Bus

Til: kkj
Fra: aaj
Dato: 96-04-17

Emne: Prototype nr. 2 af CPU305-0

Hermed en opdateret PAL-liste. Alle ændrede PAL'er har fået nye numre.

Der mangler desværre to ledninger på kortet. Dem bliver i nødt til at lægge for at kunne teste det.

Fra U148-42 til U27-30
Fra U143-42 til U49-30

Med venlig hilsen
Aage Jørgensen

PARTS LIST

Part no: 99750041 **Module:** CPU304-1 **Version:** 2
Status: PRELIMINARY **Date:** 96-08-14

Part no	Device		Qty	Comp
99020850	22NF		315	C1,C10-313,C322-331
99020851	4.7UF/25V		1	C2
99020849	33UF/16V		8	C314-321
99021413	LXH1032ID		5	D1,D5,D9,D13,D17
99021412	LXH1032GD		4	D2,D6,D10,D14
99000227	74S1050D		7	D18-24
99022032	1.000MHZ		1	OSC1
99020305	100R		48	R5,R5A,R6-9,R15,R17,R19,R21,R23 R25,R27,R29,R31-32,R35,R38-39 R41,R45,R60,R65,R68,R108-109 R112-113,R116-117,R120-121 R124-125,R128-129,R132-133 R136-137,R139,R141,R143,R145 R147,R149,R151,R153
99020307	82R		2	R10,R12
99020308	130R		2	R11,R13
99020306	220R		33	R14,R16,R18,R20,R22,R24,R26,R28 R30,R106-107,R110-111,R114-115 R118-119,R122-123,R126-127 R130-131,R134-135,R138,R140 R142,R144,R146,R148,R150,R152
99020303	4K7		7	R33,R42,R66-67,R69-70,R92
99020327	3K		2	R34,R43
99022235	NET4K7		1	R36
99020301	1K		20	R37,R40,R44,R49-57,R62-64 R86-90
99020309	560R		17	R46-48,R72-84,R91
99022234	NET1K		1	R93
99020322	0R		2	R95,R98
99020331	1K2		288	R200-487
99011702	CA303		4	U1,U3,U5,U7
99002411	74F1808D		2	U9,U11
99002412	74F1832D		2	U10,U12
99000831	74AS20D		1	U13
99002413	74F377D		1	U17
99010406	PALCE22V10H-7JC	S	1	U18
99010902	MACH110-20JC	S	3	U20,U78,U85
99010909	MACH210A-10JC	S	1	U22
99002222	74FCT16823BTPV		6	U26-27,U81-84
99011223	MCM6706AJ10		10	U28-29,U33-34,U38-39,U43-44 U140,U142
99002223	74FCT16244ATPV		5	U30,U35,U40,U45,U77
99002221	74FCT521BTSO		9	U31-32,U36-37,U41-42,U46-47 U70
99010908	MACH220-15JC	S	1	U71
99012095	82S123AA	S	2	U72-73
99002219	74FCT16501ATPV		1	U74
99002408	74F280BD		3	U75-76,U80
99012114	AM29C040-120JC		1	U79
99010604	PALCE16V8H-7JC	S	1	U86
99010604	PALCE16V8H-7JC_8	S	1	U87
99011218	CY7B991-7JC		1	U96
99005033	MC100H641FN		1	U97
99002500	FB2031B		10	U98-107
99002501	FB2033B		4	U108-110,U123
99001040	74ALS240D		2	U124-125
99000472	74LS393D		3	U127-129
99002405	74F74D		2	U130-131

PARTS LIST

Part no: 99750041 **Module:** CPU304-1 **Version:** 2
Status: PRELIMINARY **Date:** 96-08-14 **Page:** 2/2

Part no	Device		Qty	Comp
99000832	74AS30D		2	U132-133
99010022	PAL16R6-5JC	S	1	U136
99010904	MACH210-12JC	S	3	U138-139,U141
99002407	74F244D		1	U148
99000830	74AS1004D		1	U149
99040151	3M955515		16	X00-03,X10-13,X20-23,X30-33
99040111	AMP100147-1		3	X101-102,X104
99040110	AMP100145-1		1	X103
99023228	20P SOKKEL		5	U72-73,U86,U87
				U136
99023229	28P SOKKEL		1	U18
99023230	32P SOKKEL		1	U79
99023231	44P SOKKEL		7	U20,U78,U85,U22
				U138-139,U141
99023232	68P SOKKEL		1	U71
99030332	PRINT CPU305 ISS. 1		1	
990680XX	FORPLADE CPU306		1	

PLD list

Part no: 99750041 **Module:** CPU304-1 **Pcb no:** CPU305 ISS 1
Status: FINAL **Date:** 971010 **Page:** 1 / 1

Label	Pos.	Part no.	Type	X-pgm.	File	Checksum
C31171	U18	99010406	PALCE22V10H-7JC		c31171.jed	9610
C31161	U136	99010022	PAL16R6-5JC		c31161.jed	2260
C31030	U20	99010902	MACH110-20JC		c31030.jed	B930
C31151	U22	99010909	MACH210A-10JC		c31151.jed	6FA3
C31062	U138	99010904	MACH210-12JC		c31062.jed	8C39
C31071	U139	99010904	MACH210-12JC		c31071.jed	C79D
C31071	U141	99010904	MACH210-12JC		c31071.jed	C79D
C31081	U71	99010908	MACH220-15JC		c31081.jed	FE9C
C31091	U78	99010902	MACH110-20JC		c31091.jed	E1C6
C31101	U85	99010902	MACH110-20JC		c31101.jed	5A30
C31110	U86	99010604	PALCE16V8H-7JC	R6	c31110.jed	234F
C31120	U87	99010604	PALCE16V8H-7JC	R8	c31120.jed	2470
IDXXXX	U72	99012095	82S123AA/27S19			
FCN	U73	99012095	82S123AA/27S19			

Archive: nessie: /wd/mudv3/pal/cpu305-0/jed

This file: nessie:/wd/mudv3/proddok/cpu304-1/pallist

PARTS LIST

Part no: 99750043 **Module:** CPU306-1 **Version:** 2
Status: PRELIMINARY **Date:** 96-08-14

Part no	Device		Qty	Comp
99020850	22NF		315	C1,C10-313,C322-331
99020851	4.7UF/25V		1	C2
99020849	33UF/16V		8	C314-321
99021413	LXH1032ID		5	D1,D5,D9,D13,D17
99021412	LXH1032GD		4	D2,D6,D10,D14
99000227	74S1050D		7	D18-24
99022032	1.000MHZ		1	OSC1
99020305	100R		48	R5,R5A,R6-9,R15,R17,R19,R21,R23 R25,R27,R29,R31-32,R35,R38-39 R41,R45,R60,R65,R68,R108-109 R112-113,R116-117,R120-121 R124-125,R128-129,R132-133 R136-137,R139,R141,R143,R145 R147,R149,R151,R153
99020307	82R		2	R10,R12
99020308	130R		2	R11,R13
99020306	220R		33	R14,R16,R18,R20,R22,R24,R26,R28 R30,R106-107,R110-111,R114-115 R118-119,R122-123,R126-127 R130-131,R134-135,R138,R140 R142,R144,R146,R148,R150,R152
99020303	4K7		7	R33,R42,R66-67,R69-70,R92
99020327	3K		2	R34,R43
99022235	NET4K7		1	R36
99020301	1K		20	R37,R40,R44,R49-57,R62-64 R86-90
99020309	560R		17	R46-48,R72-84,R91
99022234	NET1K		1	R93
99020322	0R		2	R96,R99
99020331	1K2		288	R200-487
99011702	CA303		4	U1,U3,U5,U7
99002411	74F1808D		2	U9,U11
99002412	74F1832D		2	U10,U12
99000831	74AS20D		1	U13
99002413	74F377D		1	U17
99010406	PALCE22V10H-7JC	S	1	U18
99010902	MACH110-20JC	S	3	U20,U78,U85
99010909	MACH210A-10JC	S	1	U22
99002222	74FCT16823BTPV		6	U26-27,U81-84
99011223	MCM6706AJ10		10	U28-29,U33-34,U38-39,U43-44 U140,U142
99002223	74FCT16244ATPV		5	U30,U35,U40,U45,U77
99002221	74FCT521BTSO		9	U31-32,U36-37,U41-42,U46-47 U70
99010908	MACH220-15JC	S	1	U71
99012095	82S123AA	S	2	U72-73
99002219	74FCT16501ATPV		1	U74
99002408	74F280BD		3	U75-76,U80
99012114	AM29C040-120JC		1	U79
99010604	PALCE16V8H-7JC	S	1	U86
99010604	PALCE16V8H-7JC_8	S	1	U87
99011218	CY7B991-7JC		1	U96
99005033	MC100H641FN		1	U97
99002500	FB2031B		10	U98-107
99002501	FB2033B		4	U108-110,U123
99001040	74ALS240D		2	U124-125
99000472	74LS393D		3	U127-129
99002405	74F74D		2	U130-131

PARTS LIST

Part no: 99750043 **Module:** CPU306-1 **Version:** 2
Status: PRELIMINARY **Date:** 96-08-14 **Page:** 2/2

Part no	Device		Qty	Comp
99000832	74AS30D		2	U132-133
99010022	PAL16R6-5JC	S	1	U136
99010904	MACH210-12JC	S	3	U138-139,U141
99002407	74F244D		1	U148
99000830	74AS1004D		1	U149
99040151	3M955515		16	X00-03,X10-13,X20-23,X30-33
99040111	AMP100147-1		3	X101-102,X104
99040110	AMP100145-1		1	X103
99023228	20P SOKKEL		5	U72-73,U86,U87
				U136
99023229	28P SOKKEL		1	U18
99023230	32P SOKKEL		1	U79
99023231	44P SOKKEL		7	U20,U78,U85,U22
				U138-139,U141
99023232	68P SOKKEL		1	U71
99030332	PRINT CPU305 ISS. 1		1	
990680XX	FORPLADE CPU306		1	

PLD list

Part no: 99750043 **Module:** CPU306-1 **Pcb no:** CPU305 ISS 1
Status: FINAL **Date:** 971010 **Page:** 1 / 1

Label	Pos.	Part no.	Type	X-pgm.	File	Checksum
C31171	U18	99010406	PALCE22V10H-7JC		c31171.jed	9610
C31161	U136	99010022	PAL16R6-5JC		c31161.jed	2260
C31030	U20	99010902	MACH110-20JC		c31030.jed	B930
C31151	U22	99010909	MACH210A-10JC		c31151.jed	6FA3
C31062	U138	99010904	MACH210-12JC		c31062.jed	8C39
C31071	U139	99010904	MACH210-12JC		c31071.jed	C79D
C31071	U141	99010904	MACH210-12JC		c31071.jed	C79D
C31081	U71	99010908	MACH220-15JC		c31081.jed	FE9C
C31091	U78	99010902	MACH110-20JC		c31091.jed	E1C6
C31101	U85	99010902	MACH110-20JC		c31101.jed	5A30
C31110	U86	99010604	PALCE16V8H-7JC	R6	c31110.jed	234F
C31120	U87	99010604	PALCE16V8H-7JC	R8	c31120.jed	2470
IDXXXX	U72	99012095	82S123AA/27S19			
FCN	U73	99012095	82S123AA/27S19			

Archive: nessie: /wd/mudv3/pal/cpu305-0/jed

This file: nessie:/wd/mudv3/proddok/cpu306-1/pallist

AAJ / 1997-01-24

CPU304/305/306/307

Cache størrelse og cache linie længde strappes v.h.a. 0 ohms modstandene R94, R95, R96, R97, R98 og R99. De lovlige kombinationer er vist i nedenstående tabel.

	R94	R95	R96	R97	R98	R99
CPU304, 1 MByte/32 words		X			X	
CPU305, 1 MByte/32 words		X			X	
CPU306, 4 MByte/32 words			X			X
CPU306, 4 MByte/32 words			X			X

CPU304-307 Diagrams

CPU304: 1 MB snooper. Only global bus components mounted.
CPU305: 1 MB snooper.
CPU306: 4 MB snooper. Only global bus components mounted.
CPU307: 4 MB snooper.

Processor/agent interface

1 Connectors 0
2 Global agent 0
3 Local agent 0
4 Connectors 1
5 Global agent 1
6 Local agent 1
7 Connectors 2
8 Global agent 2
9 Local agent 2
10 Connectors 3
11 Global agent 3
12 Local agent 3
13 Programmable LEDs
14 Reset counters
15 Gating towards processors
16 Gating towards system bus

Bus control

17 Global and local output enable control
18 Pull-down for address and data valid
19 Global and local bus request decoder
20 Global bus arbitration
21 Local bus arbitration

Snooper control

22 Global snooper control
23 Global super snooper 0-1
24 Global super snooper 2-3
25 Local snooper control
26 Local super snooper 0-1
27 Local super snooper 2-3

Address and command registers and snoopers

28 Global and local command register
29 Global address register
30 Global snooper 0
31 Global snooper 1
32 Global snooper 2
33 Global snooper 3
34 Local address register
35 Local snooper 0
36 Local snooper 1
37 Local snooper 2
38 Local snooper 3

Global control space, boot, and dummy block generator

39 Global control space decode
40 Global control space control
41 Module ID and FCN PROMs
42 Processor module ID register
43 Global control space transceiver
44 Boot control
45 Boot PROM
46 Boot register
47 Global dummy block generator
48 Global data identifier output register

Local control space and dummy block generator

49 Local control space decode and target acknowledge
50 Local dummy block generator
51 Local dummy data
52 Local data identifier output register

Clock

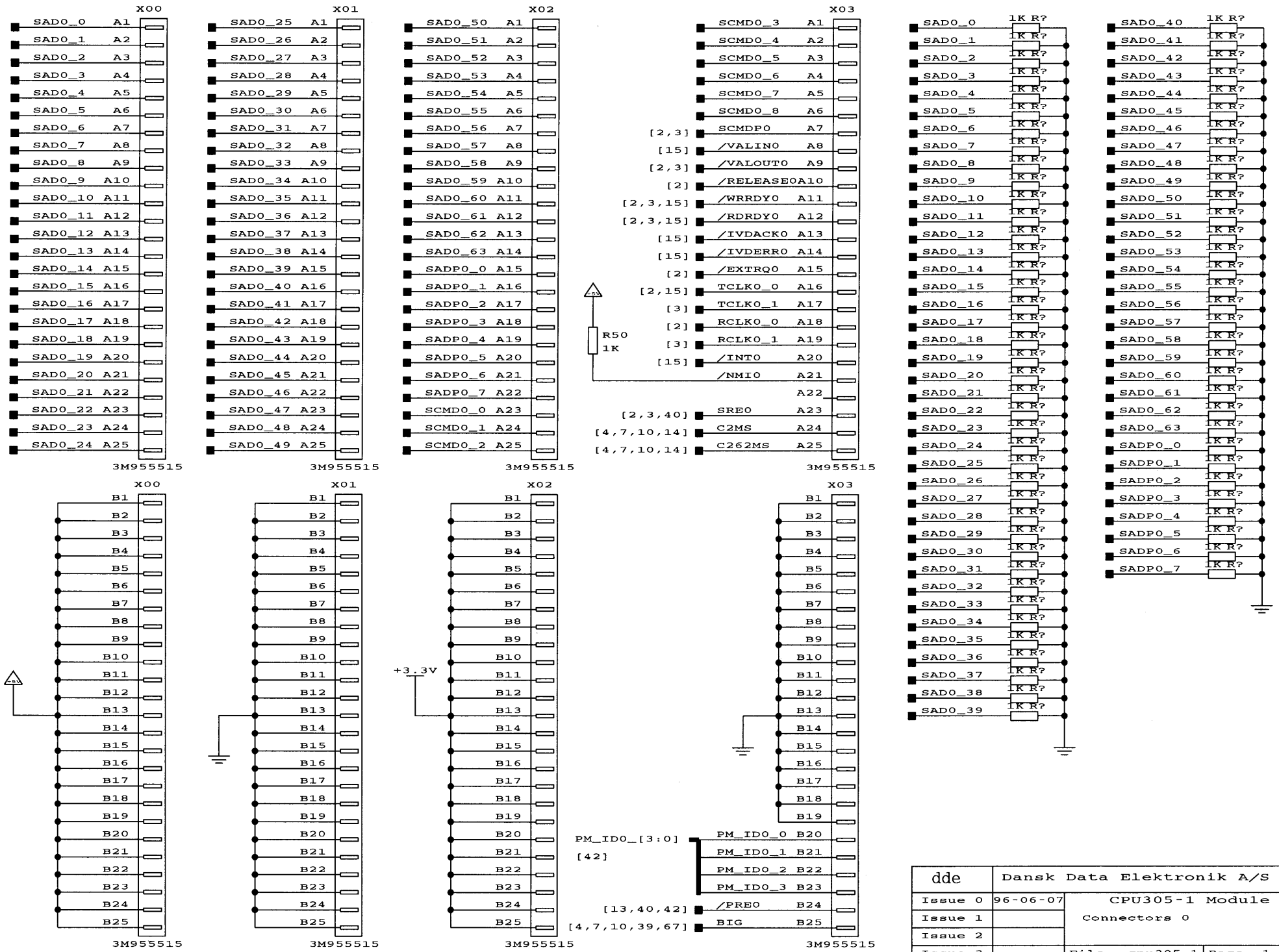
53 Clock distribution

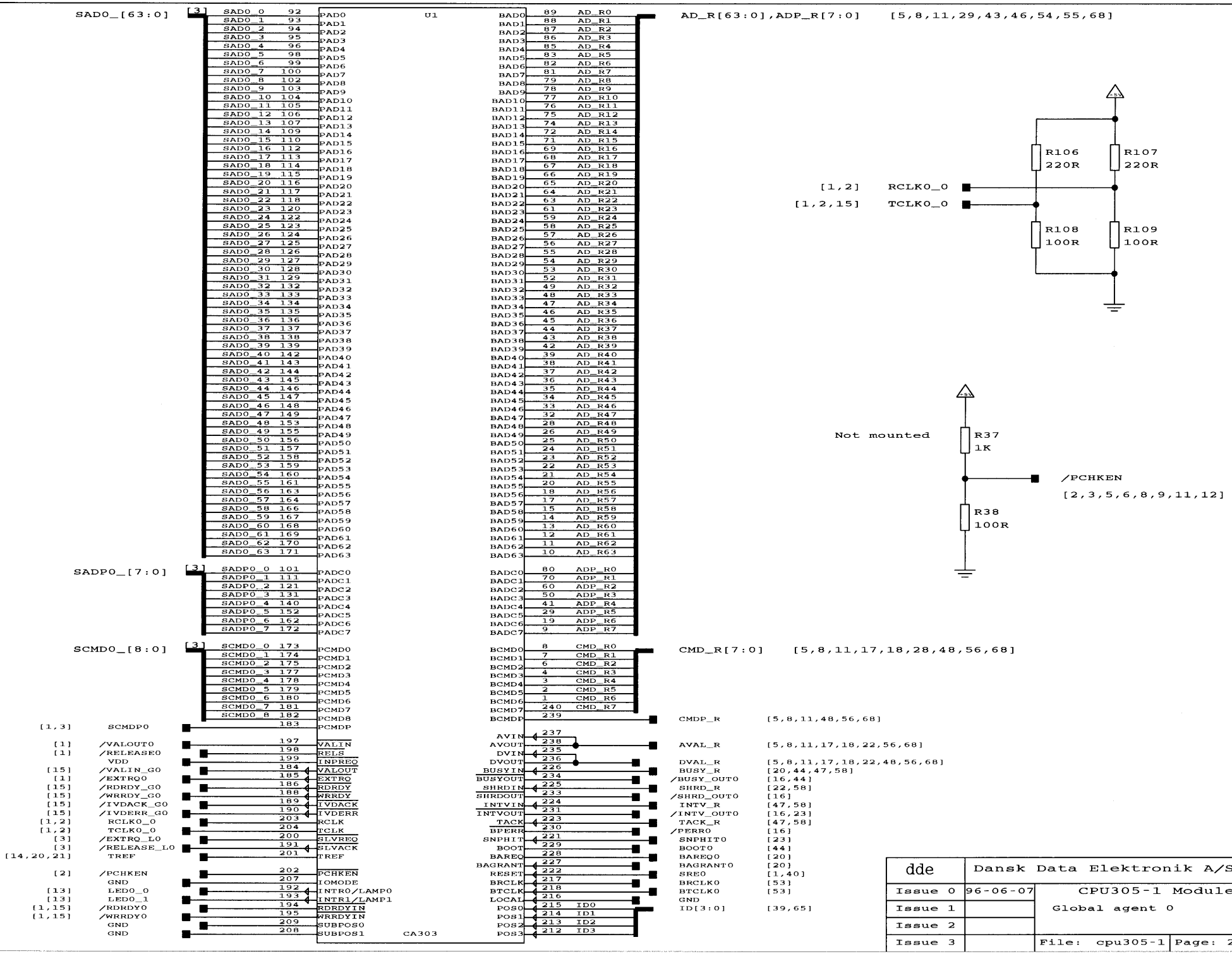
Bus interface

54 Global address/data transceiver
55 Global address/data transceiver
56 Global command and valid transceiver
57 Global bus request transceiver
58 Global control transceiver
59 Local address/data transceiver
60 Local address/data transceiver
61 Local command and valid transceiver
62 Local bus request transceiver
63 Local control transceiver
64 Connector (row 1-25)
65 Connector (row 26-50)
66 Connector (row 51-75)
67 Connector (row 76-100)

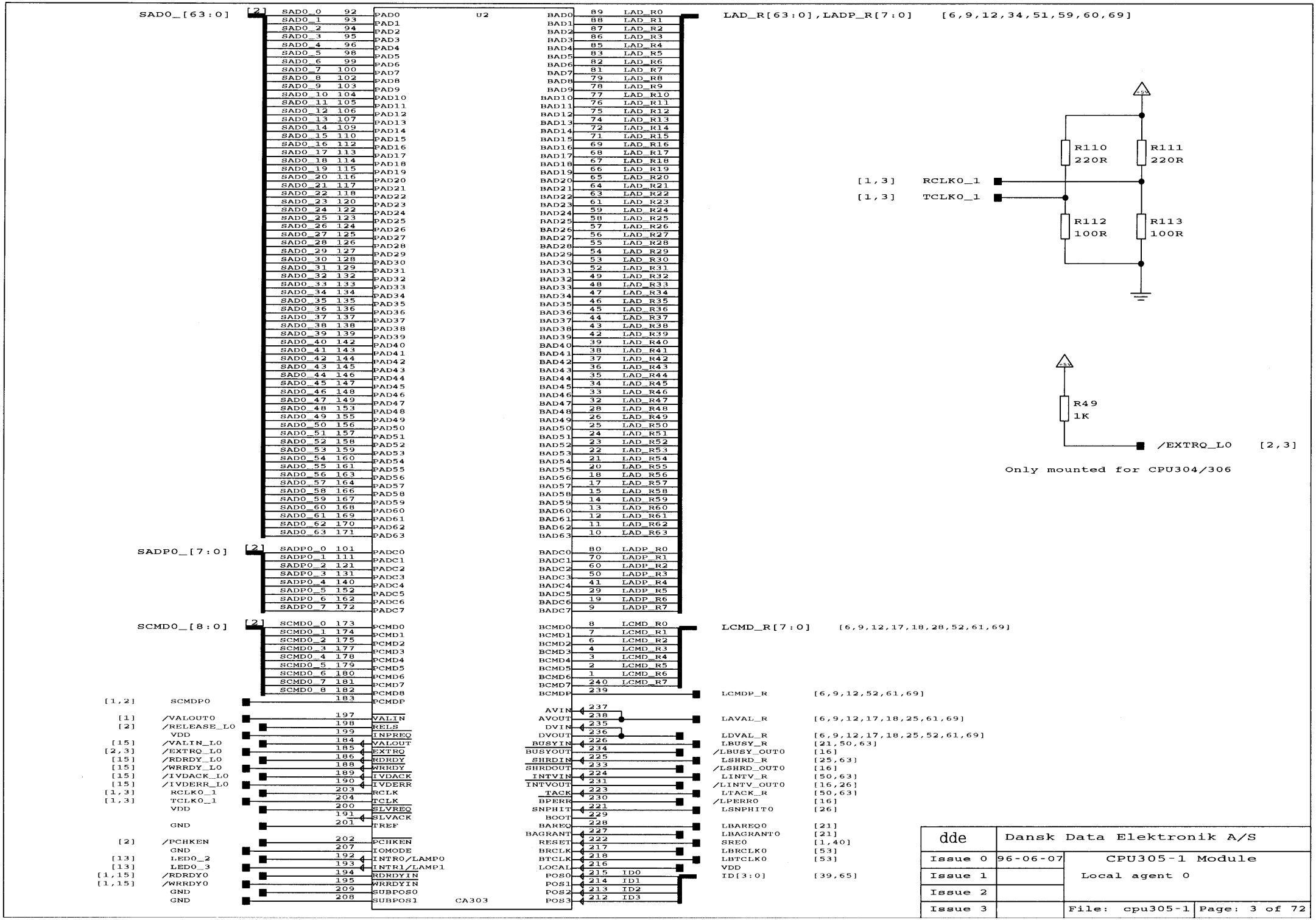
Termination and decoupling

68 Internal global bus termination
69 Internal local bus termination
70 Decoupling capacitors
71 Decoupling capacitors
72 Decoupling capacitors

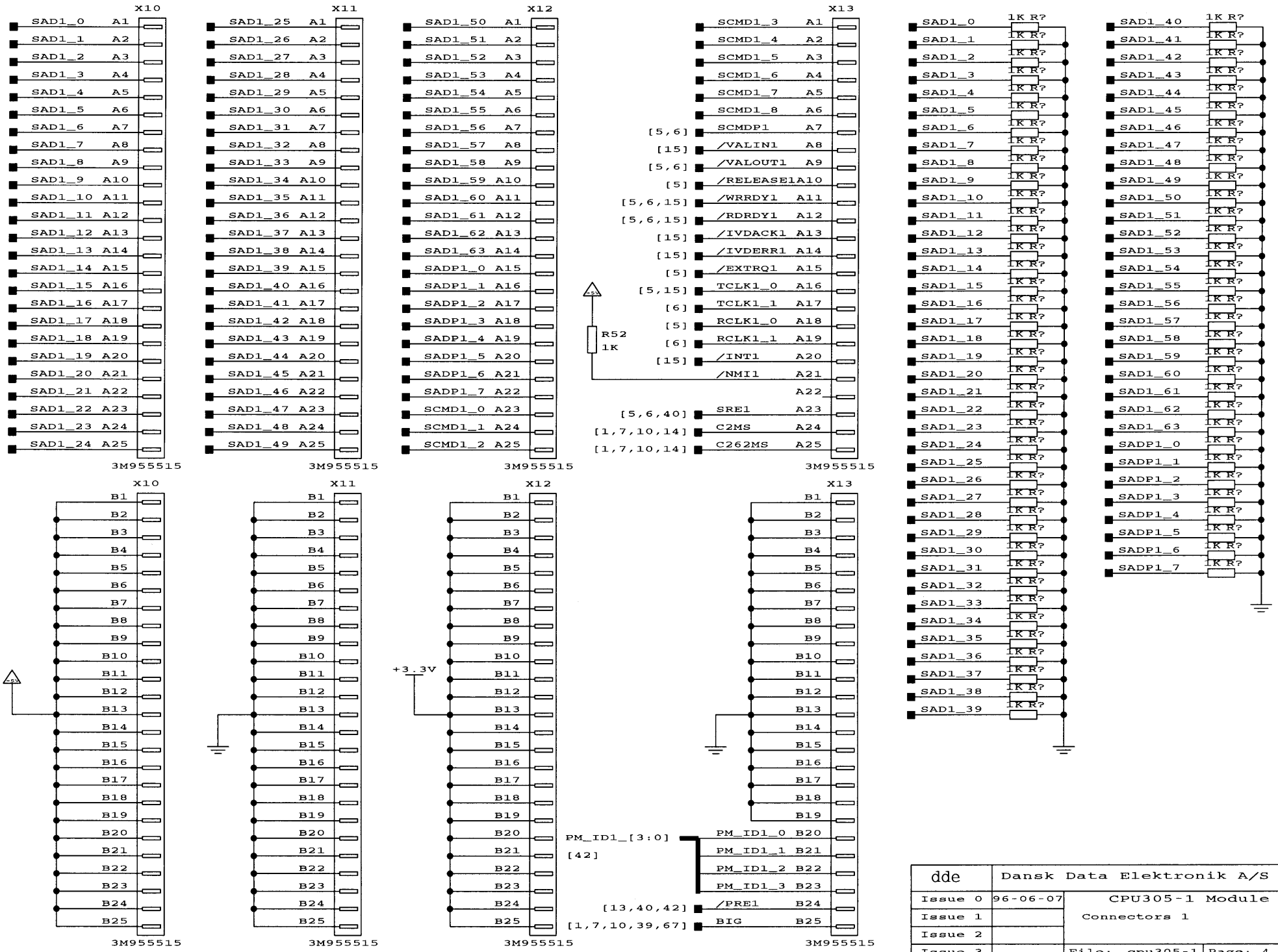

X00
SAD0_0 A1
SAD0_1 A2
SAD0_2 A3
SAD0_3 A4
SAD0_4 A5
SAD0_5 A6
SAD0_6 A7
SAD0_7 A8
SAD0_8 A9
SAD0_9 A10
SAD0_10 A11
SAD0_11 A12
SAD0_12 A13
SAD0_13 A14
SAD0_14 A15
SAD0_15 A16
SAD0_16 A17
SAD0_17 A18
SAD0_18 A19
SAD0_19 A20
SAD0_20 A21
SAD0_21 A22
SAD0_22 A23
SAD0_23 A24
SAD0_24 A25
3M955515
X01
SAD0_25 A1
SAD0_26 A2
SAD0_27 A3
SAD0_28 A4
SAD0_29 A5
SAD0_30 A6
SAD0_31 A7
SAD0_32 A8
SAD0_33 A9
SAD0_34 A10
SAD0_35 A11
SAD0_36 A12
SAD0_37 A13
SAD0_38 A14
SAD0_39 A15
SAD0_40 A16
SAD0_41 A17
SAD0_42 A18
SAD0_43 A19
SAD0_44 A20
SAD0_45 A21
SAD0_46 A22
SAD0_47 A23
SAD0_48 A24
SAD0_49 A25
3M955515
X02
SAD0_50 A1
SAD0_51 A2
SAD0_52 A3
SAD0_53 A4
SAD0_54 A5
SAD0_55 A6
SAD0_56 A7
SAD0_57 A8
SAD0_58 A9
SAD0_59 A10
SAD0_60 A11
SAD0_61 A12
SAD0_62 A13
SAD0_63 A14
SADP0_0 A15
SADP0_1 A16
SADP0_2 A17
SADP0_3 A18
SADP0_4 A19
SADP0_5 A20
SADP0_6 A21
SADP0_7 A22
SCMD0_0 A23
SCMD0_1 A24
SCMD0_2 A25
3M955515
X03
SCMD0_3 A1
SCMD0_4 A2
SCMD0_5 A3
SCMD0_6 A4
SCMD0_7 A5
SCMD0_8 A6
[2,3] SCMDP0 A7
[15] /VALIN0 A8
[2,3] /VALOUT0 A9
[2] /RELEASE0 A10
[2,3,15] /WRRDY0 A11
[2,3,15] /RDRDY0 A12
[15] /IVDACK0 A13
[15] /IVDERR0 A14
[2] /EXTRQ0 A15
[2,15] TCLK0_0 A16
[3] TCLK0_1 A17
[2] RCLK0_0 A18
[3] RCLK0_1 A19
[15] /INT0 A20
/NMI0 A21
A22
[2,3,40] SRE0 A23
[4,7,10,14] C2MS A24
[4,7,10,14] C262MS A25
3M955515
+5V
R50
1K
1K R?
SAD0_0
SAD0_1
SAD0_2
SAD0_3
SAD0_4
SAD0_5
SAD0_6
SAD0_7
SAD0_8
SAD0_9
SAD0_10
SAD0_11
SAD0_12
SAD0_13
SAD0_14
SAD0_15
SAD0_16
SAD0_17
SAD0_18
SAD0_19
SAD0_20
SAD0_21
SAD0_22
SAD0_23
SAD0_24
SAD0_25
SAD0_26
SAD0_27
SAD0_28
SAD0_29
SAD0_30
SAD0_31
SAD0_32
SAD0_33
SAD0_34
SAD0_35
SAD0_36
SAD0_37
SAD0_38
SAD0_39
SAD0_40
SAD0_41
SAD0_42
SAD0_43
SAD0_44
SAD0_45
SAD0_46
SAD0_47
SAD0_48
SAD0_49
SAD0_50
SAD0_51
SAD0_52
SAD0_53
SAD0_54
SAD0_55
SAD0_56
SAD0_57
SAD0_58
SAD0_59
SAD0_60
SAD0_61
SAD0_62
SAD0_63
SADP0_0
SADP0_1
SADP0_2
SADP0_3
SADP0_4
SADP0_5
SADP0_6
SADP0_7
X00
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17 B18 B19 B20 B21 B22 B23 B24 B25
3M955515
X01
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17 B18 B19 B20 B21 B22 B23 B24 B25
3M955515
X02
+3.3V
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17 B18 B19 B20 B21 B22 B23 B24 B25
3M955515
X03
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17 B18 B19
PM_IDO_[3:0] [42]
PM_IDO_0 B20
PM_IDO_1 B21
PM_IDO_2 B22
PM_IDO_3 B23
[13,40,42] /PRE0 B24
[4,7,10,39,67] BIG B25
3M955515
dde
Dansk Data Elektronik A/S
Issue 0 96-06-07
Issue 1
Issue 2
Issue 3
CPU305-1 Module
Connectors 0
File: cpu305-1
Page: 1 of 72

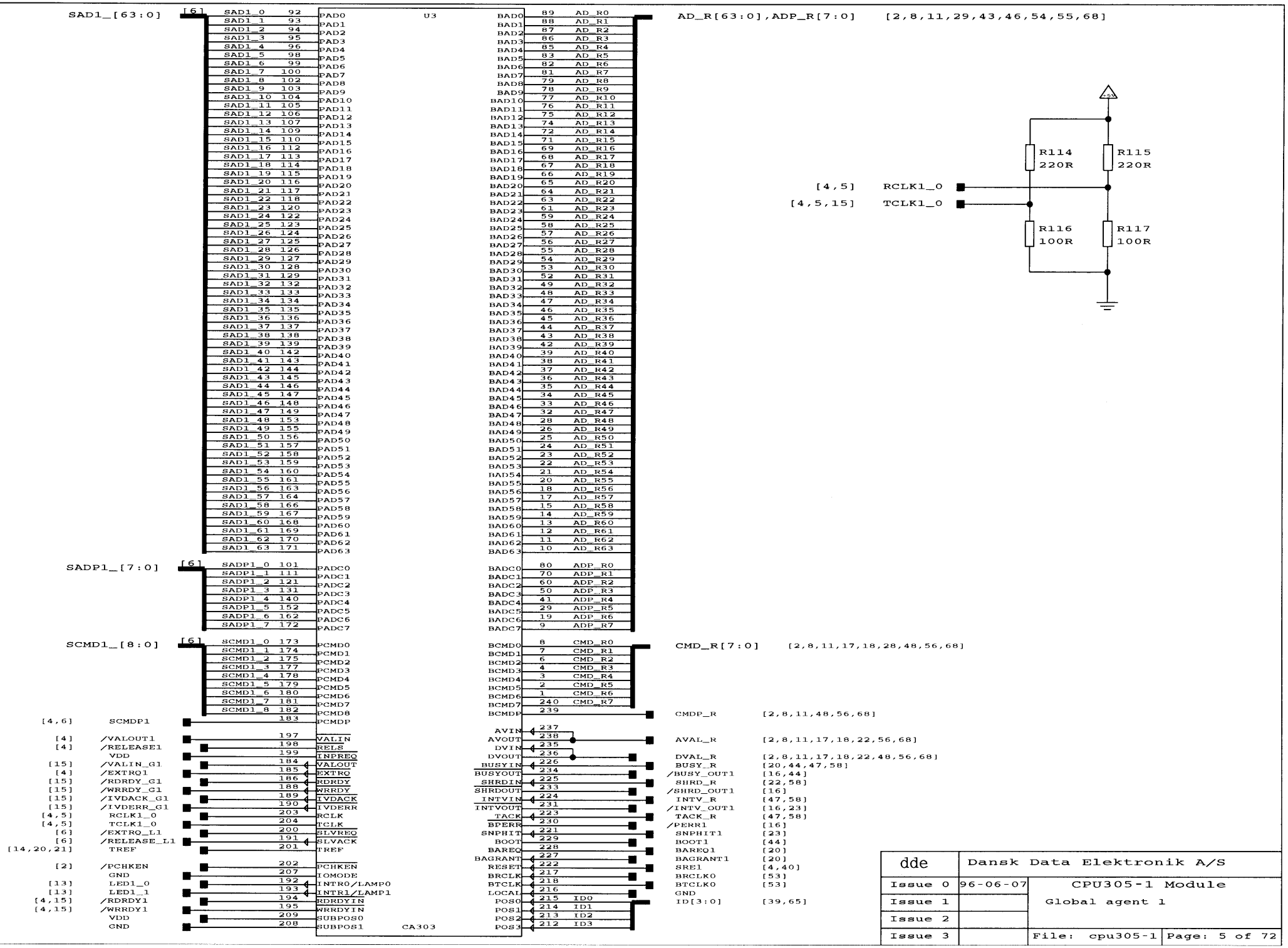


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global agent 0	
Issue 2			
Issue 3		File: cpu305-1	Page: 2 of 72

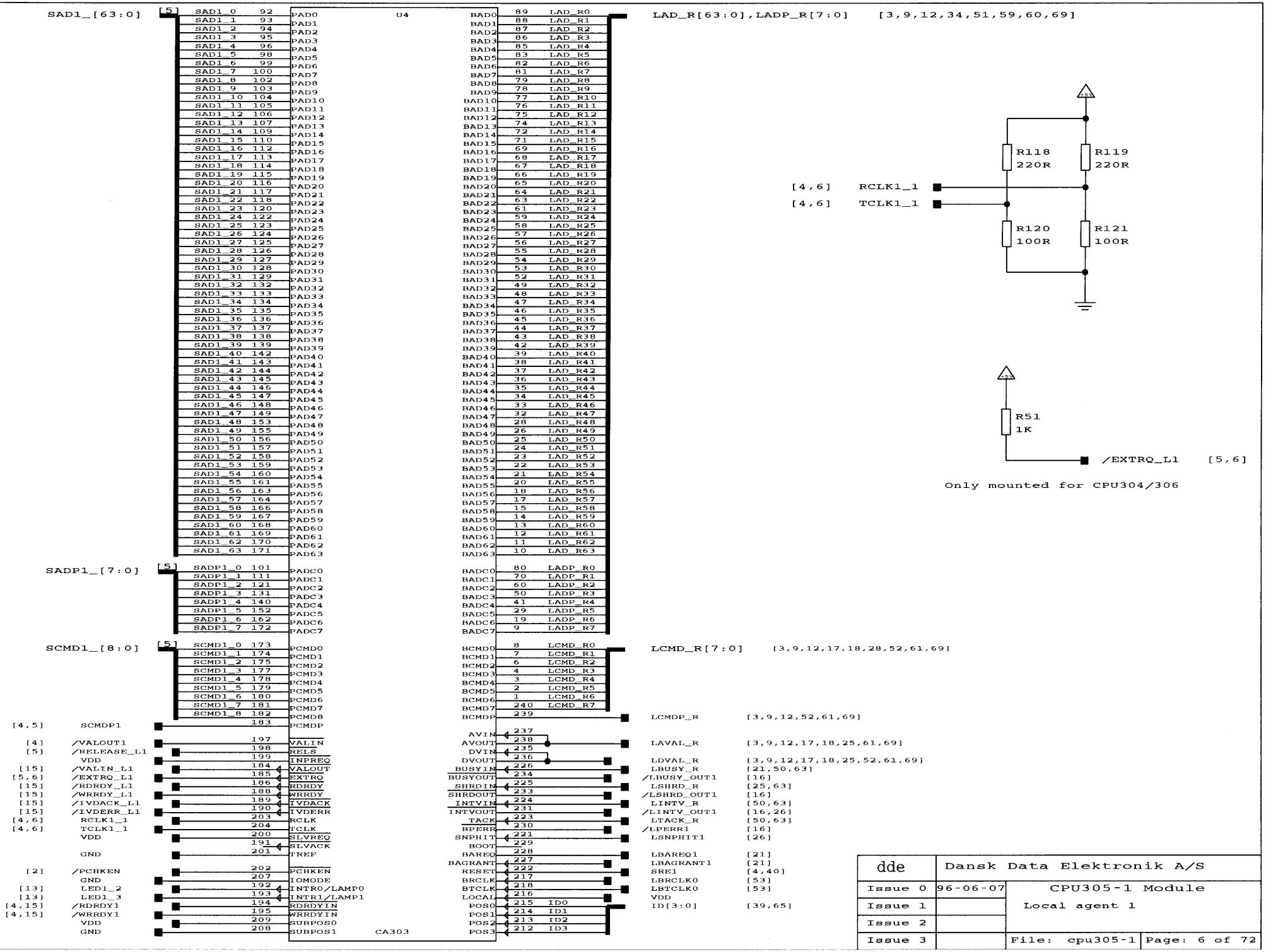


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local agent 0	
Issue 2			
Issue 3		File: cpu305-1	Page: 3 of 72

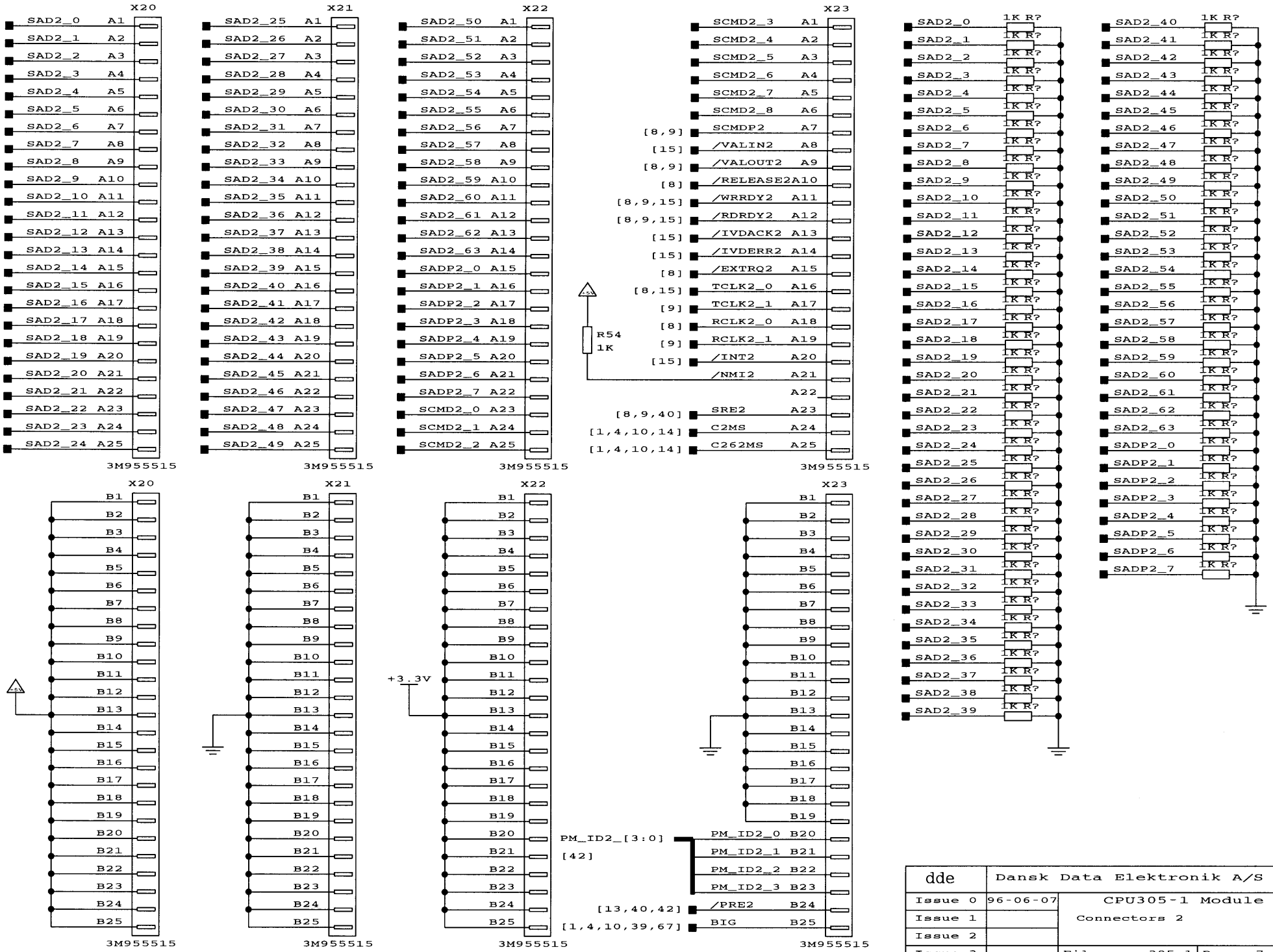

X10
SAD1_0 A1
SAD1_1 A2
SAD1_2 A3
SAD1_3 A4
SAD1_4 A5
SAD1_5 A6
SAD1_6 A7
SAD1_7 A8
SAD1_8 A9
SAD1_9 A10
SAD1_10 A11
SAD1_11 A12
SAD1_12 A13
SAD1_13 A14
SAD1_14 A15
SAD1_15 A16
SAD1_16 A17
SAD1_17 A18
SAD1_18 A19
SAD1_19 A20
SAD1_20 A21
SAD1_21 A22
SAD1_22 A23
SAD1_23 A24
SAD1_24 A25
3M955515
X11
SAD1_25 A1
SAD1_26 A2
SAD1_27 A3
SAD1_28 A4
SAD1_29 A5
SAD1_30 A6
SAD1_31 A7
SAD1_32 A8
SAD1_33 A9
SAD1_34 A10
SAD1_35 A11
SAD1_36 A12
SAD1_37 A13
SAD1_38 A14
SAD1_39 A15
SAD1_40 A16
SAD1_41 A17
SAD1_42 A18
SAD1_43 A19
SAD1_44 A20
SAD1_45 A21
SAD1_46 A22
SAD1_47 A23
SAD1_48 A24
SAD1_49 A25
3M955515
X12
SAD1_50 A1
SAD1_51 A2
SAD1_52 A3
SAD1_53 A4
SAD1_54 A5
SAD1_55 A6
SAD1_56 A7
SAD1_57 A8
SAD1_58 A9
SAD1_59 A10
SAD1_60 A11
SAD1_61 A12
SAD1_62 A13
SAD1_63 A14
SADP1_0 A15
SADP1_1 A16
SADP1_2 A17
SADP1_3 A18
SADP1_4 A19
SADP1_5 A20
SADP1_6 A21
SADP1_7 A22
SCMD1_0 A23
SCMD1_1 A24
SCMD1_2 A25
3M955515
X13
SCMD1_3 A1
SCMD1_4 A2
SCMD1_5 A3
SCMD1_6 A4
SCMD1_7 A5
SCMD1_8 A6
[5,6] SCMDP1 A7
[15] /VALIN1 A8
[5,6] /VALOUT1 A9
[5] /RELEASE1 A10
[5,6,15] /WRRDY1 A11
[5,6,15] /RDRDY1 A12
[15] /IVDACK1 A13
[15] /IVDERR1 A14
[5] /EXTRQ1 A15
[5,15] TCLK1_0 A16
[6] TCLK1_1 A17
[5] RCLK1_0 A18
[6] RCLK1_1 A19
[15] /INT1 A20
/NMI1 A21
A22
[5,6,40] SRE1 A23
[1,7,10,14] C2MS A24
[1,7,10,14] C262MS A25
3M955515
+5V
R52
1K
SAD1_0 1K R?
SAD1_1 1K R?
SAD1_2 1K R?
SAD1_3 1K R?
SAD1_4 1K R?
SAD1_5 1K R?
SAD1_6 1K R?
SAD1_7 1K R?
SAD1_8 1K R?
SAD1_9 1K R?
SAD1_10 1K R?
SAD1_11 1K R?
SAD1_12 1K R?
SAD1_13 1K R?
SAD1_14 1K R?
SAD1_15 1K R?
SAD1_16 1K R?
SAD1_17 1K R?
SAD1_18 1K R?
SAD1_19 1K R?
SAD1_20 1K R?
SAD1_21 1K R?
SAD1_22 1K R?
SAD1_23 1K R?
SAD1_24 1K R?
SAD1_25 1K R?
SAD1_26 1K R?
SAD1_27 1K R?
SAD1_28 1K R?
SAD1_29 1K R?
SAD1_30 1K R?
SAD1_31 1K R?
SAD1_32 1K R?
SAD1_33 1K R?
SAD1_34 1K R?
SAD1_35 1K R?
SAD1_36 1K R?
SAD1_37 1K R?
SAD1_38 1K R?
SAD1_39 1K R?
SAD1_40 1K R?
SAD1_41 1K R?
SAD1_42 1K R?
SAD1_43 1K R?
SAD1_44 1K R?
SAD1_45 1K R?
SAD1_46 1K R?
SAD1_47 1K R?
SAD1_48 1K R?
SAD1_49 1K R?
SAD1_50 1K R?
SAD1_51 1K R?
SAD1_52 1K R?
SAD1_53 1K R?
SAD1_54 1K R?
SAD1_55 1K R?
SAD1_56 1K R?
SAD1_57 1K R?
SAD1_58 1K R?
SAD1_59 1K R?
SAD1_60 1K R?
SAD1_61 1K R?
SAD1_62 1K R?
SAD1_63 1K R?
SADP1_0 1K R?
SADP1_1 1K R?
SADP1_2 1K R?
SADP1_3 1K R?
SADP1_4 1K R?
SADP1_5 1K R?
SADP1_6 1K R?
SADP1_7 1K R?
X10
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17 B18 B19 B20 B21 B22 B23 B24 B25
3M955515
X11
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17 B18 B19 B20 B21 B22 B23 B24 B25
3M955515
X12
+3.3V
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17 B18 B19 B20 B21 B22 B23 B24 B25
3M955515
X13
PM_ID1_[3:0]
[42]
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17 B18 B19
PM_ID1_0 B20
PM_ID1_1 B21
PM_ID1_2 B22
PM_ID1_3 B23
[13,40,42] /PRE1 B24
[1,7,10,39,67] BIG B25
3M955515
dde
Dansk Data Elektronik A/S
Issue 0 96-06-07
Issue 1
Issue 2
Issue 3
CPU305-1 Module
Connectors 1
File: cpu305-1
Page: 4 of 72

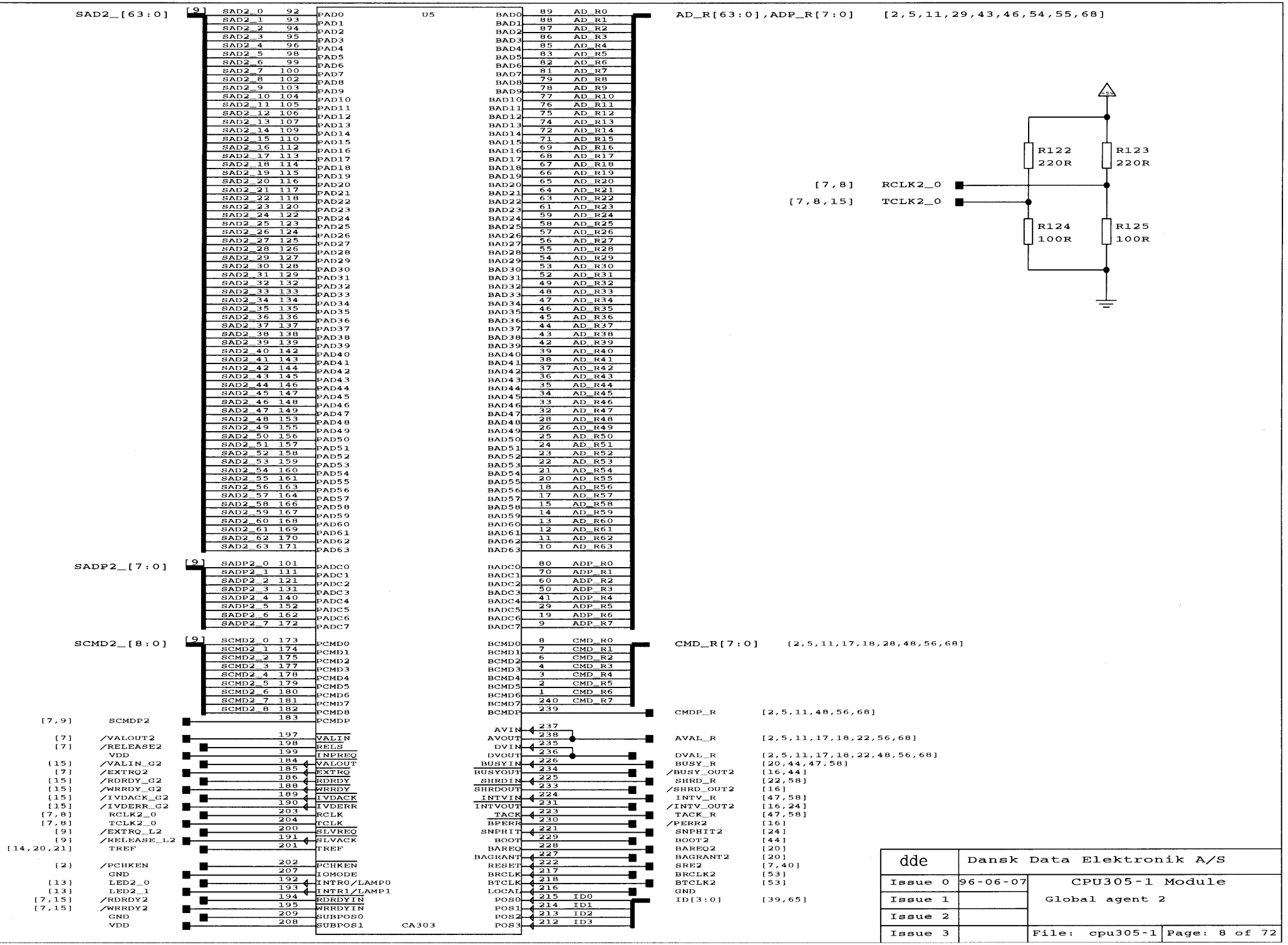


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global agent 1	
Issue 2			
Issue 3		File: cpu305-1	Page: 5 of 72

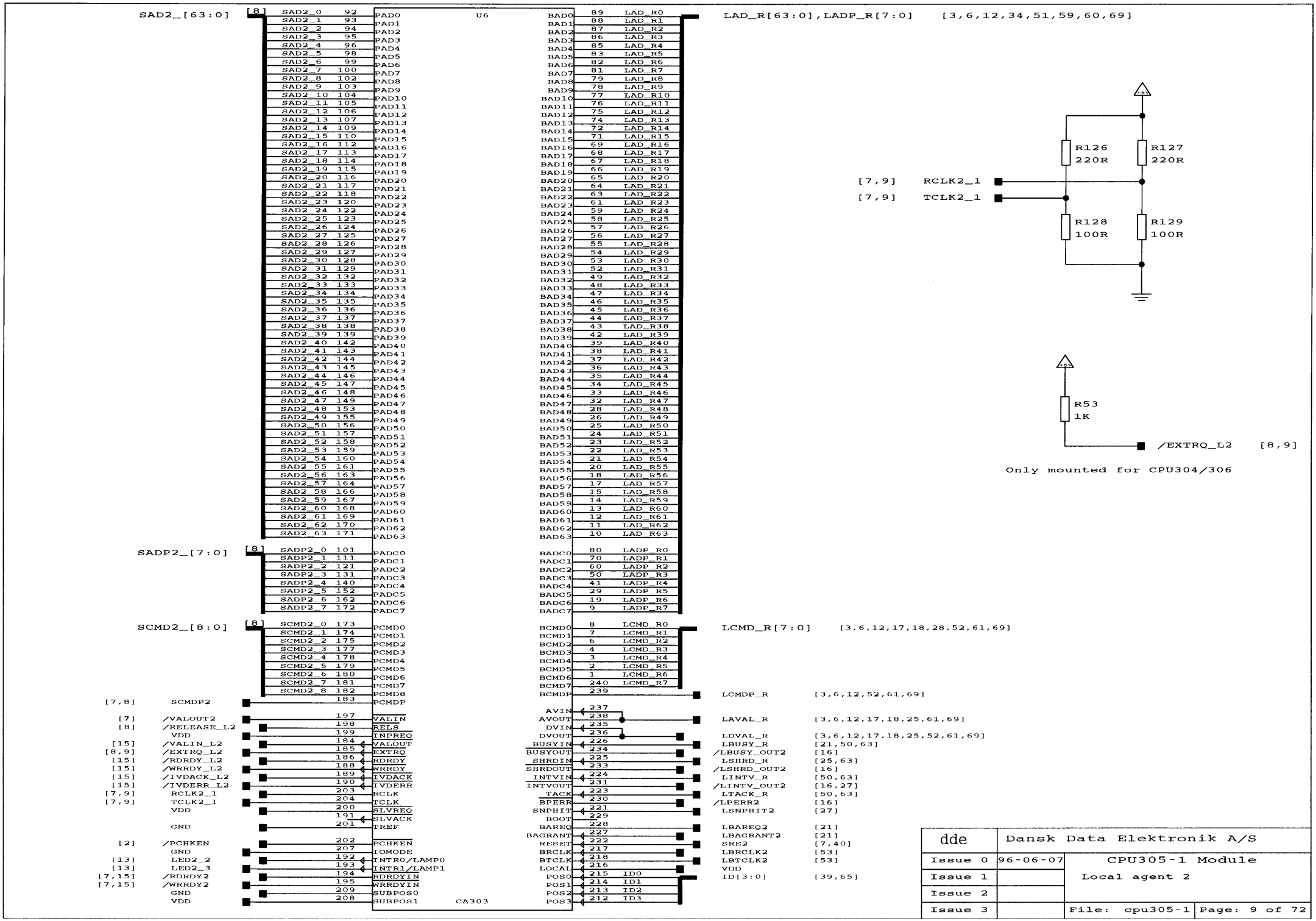


dde	Dansk Data Elektronik A/S	
Issue 0	96-06-07	CPU305-1 Module
Issue 1		Local agent 1
Issue 2		
Issue 3		File: cpu305-1 Page: 6 of 72

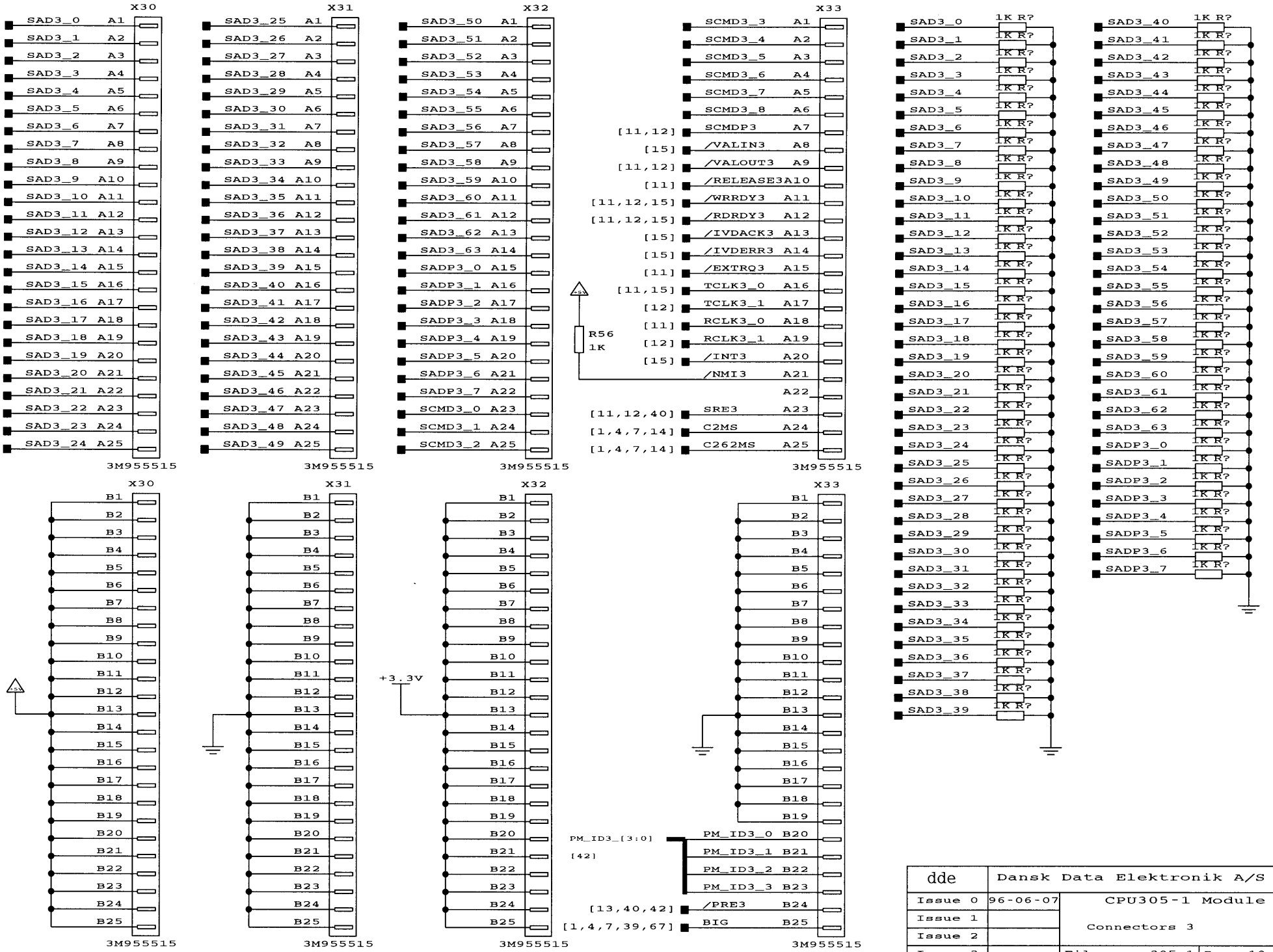

X20
SAD2_0 A1
SAD2_1 A2
SAD2_2 A3
SAD2_3 A4
SAD2_4 A5
SAD2_5 A6
SAD2_6 A7
SAD2_7 A8
SAD2_8 A9
SAD2_9 A10
SAD2_10 A11
SAD2_11 A12
SAD2_12 A13
SAD2_13 A14
SAD2_14 A15
SAD2_15 A16
SAD2_16 A17
SAD2_17 A18
SAD2_18 A19
SAD2_19 A20
SAD2_20 A21
SAD2_21 A22
SAD2_22 A23
SAD2_23 A24
SAD2_24 A25
3M955515
X21
SAD2_25 A1
SAD2_26 A2
SAD2_27 A3
SAD2_28 A4
SAD2_29 A5
SAD2_30 A6
SAD2_31 A7
SAD2_32 A8
SAD2_33 A9
SAD2_34 A10
SAD2_35 A11
SAD2_36 A12
SAD2_37 A13
SAD2_38 A14
SAD2_39 A15
SAD2_40 A16
SAD2_41 A17
SAD2_42 A18
SAD2_43 A19
SAD2_44 A20
SAD2_45 A21
SAD2_46 A22
SAD2_47 A23
SAD2_48 A24
SAD2_49 A25
3M955515
X22
SAD2_50 A1
SAD2_51 A2
SAD2_52 A3
SAD2_53 A4
SAD2_54 A5
SAD2_55 A6
SAD2_56 A7
SAD2_57 A8
SAD2_58 A9
SAD2_59 A10
SAD2_60 A11
SAD2_61 A12
SAD2_62 A13
SAD2_63 A14
SADP2_0 A15
SADP2_1 A16
SADP2_2 A17
SADP2_3 A18
SADP2_4 A19
SADP2_5 A20
SADP2_6 A21
SADP2_7 A22
SCMD2_0 A23
SCMD2_1 A24
SCMD2_2 A25
3M955515
X23
SCMD2_3 A1
SCMD2_4 A2
SCMD2_5 A3
SCMD2_6 A4
SCMD2_7 A5
SCMD2_8 A6
[8,9] SCMDP2 A7
[15] /VALIN2 A8
[8,9] /VALOUT2 A9
[8] /RELEASE2 A10
[8,9,15] /WRRDY2 A11
[8,9,15] /RDRDY2 A12
[15] /IVDACK2 A13
[15] /IVDERR2 A14
[8] /EXTRQ2 A15
[8,15] TCLK2_0 A16
[9] TCLK2_1 A17
[8] RCLK2_0 A18
[9] RCLK2_1 A19
[15] /INT2 A20
/NMI2 A21
A22
[8,9,40] SRE2 A23
[1,4,10,14] C2MS A24
[1,4,10,14] C262MS A25
3M955515
R54
1K
1K R?
SAD2_0
SAD2_1
SAD2_2
SAD2_3
SAD2_4
SAD2_5
SAD2_6
SAD2_7
SAD2_8
SAD2_9
SAD2_10
SAD2_11
SAD2_12
SAD2_13
SAD2_14
SAD2_15
SAD2_16
SAD2_17
SAD2_18
SAD2_19
SAD2_20
SAD2_21
SAD2_22
SAD2_23
SAD2_24
SAD2_25
SAD2_26
SAD2_27
SAD2_28
SAD2_29
SAD2_30
SAD2_31
SAD2_32
SAD2_33
SAD2_34
SAD2_35
SAD2_36
SAD2_37
SAD2_38
SAD2_39
SAD2_40
SAD2_41
SAD2_42
SAD2_43
SAD2_44
SAD2_45
SAD2_46
SAD2_47
SAD2_48
SAD2_49
SAD2_50
SAD2_51
SAD2_52
SAD2_53
SAD2_54
SAD2_55
SAD2_56
SAD2_57
SAD2_58
SAD2_59
SAD2_60
SAD2_61
SAD2_62
SAD2_63
SADP2_0
SADP2_1
SADP2_2
SADP2_3
SADP2_4
SADP2_5
SADP2_6
SADP2_7
X20
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
3M955515
X21
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
3M955515
X22
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
3M955515
+3.3V
X23
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
PM_ID2_[3:0]
[42]
PM_ID2_0 B20
PM_ID2_1 B21
PM_ID2_2 B22
PM_ID2_3 B23
[13,40,42] /PRE2 B24
[1,4,10,39,67] BIG B25
3M955515
dde
Dansk Data Elektronik A/S
Issue 0 96-06-07
Issue 1
Issue 2
Issue 3
CPU305-1 Module
Connectors 2
File: cpu305-1
Page: 7 of 72

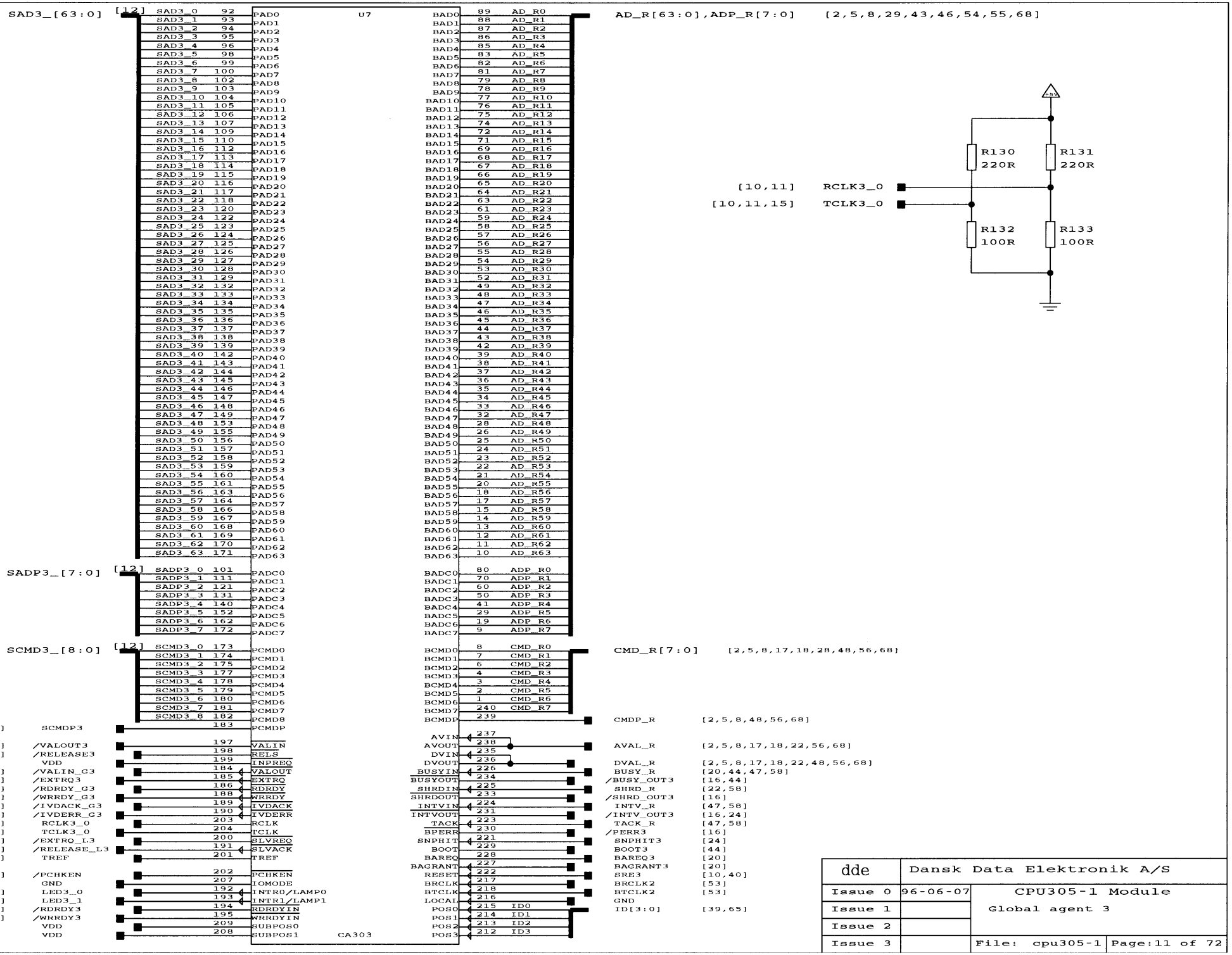


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global agent 2	
Issue 2			
Issue 3		File: cpu305-1	Page: 8 of 72

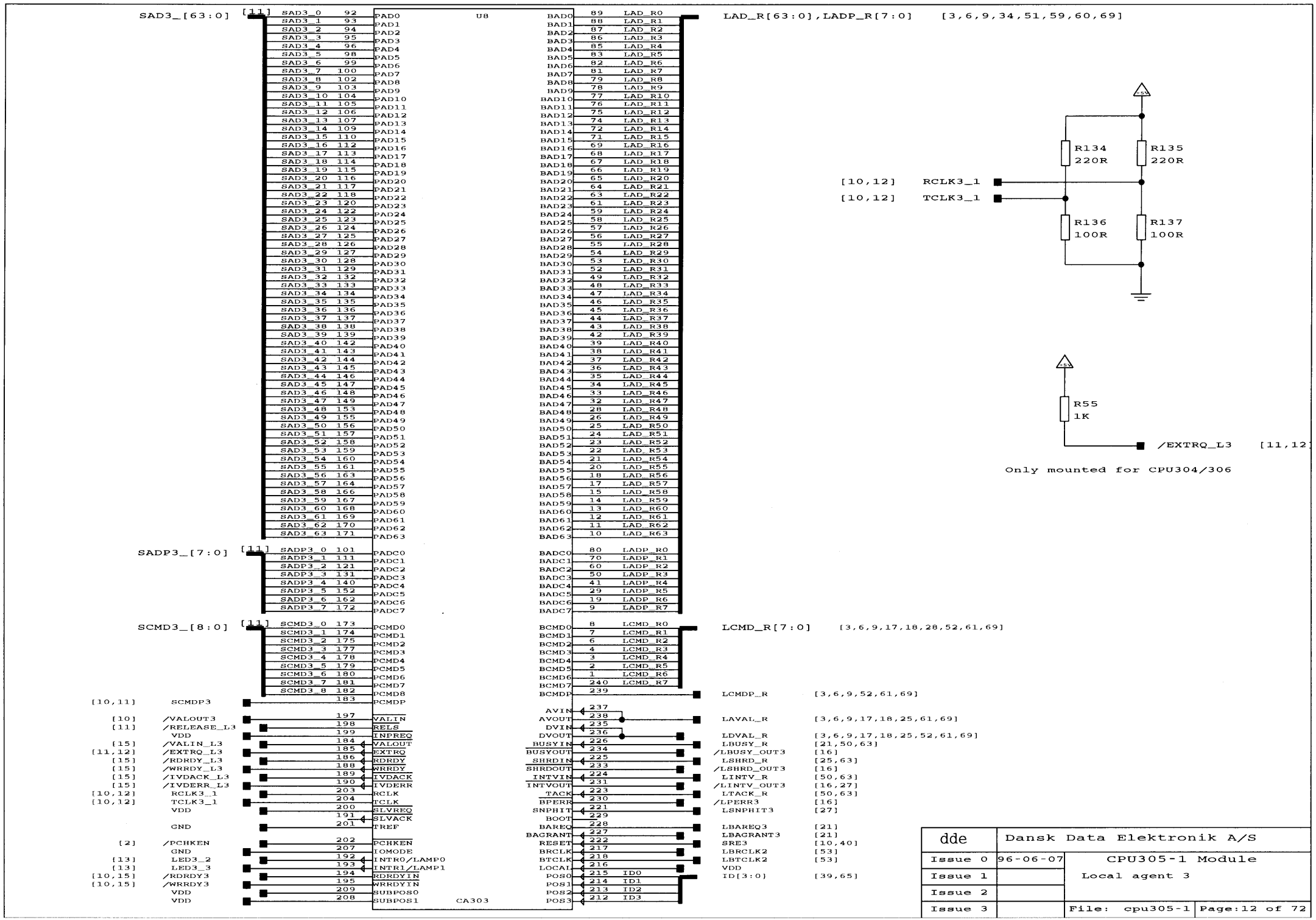


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local agent 2	
Issue 2			
Issue 3		File: cpu305-1	Page: 9 of 72

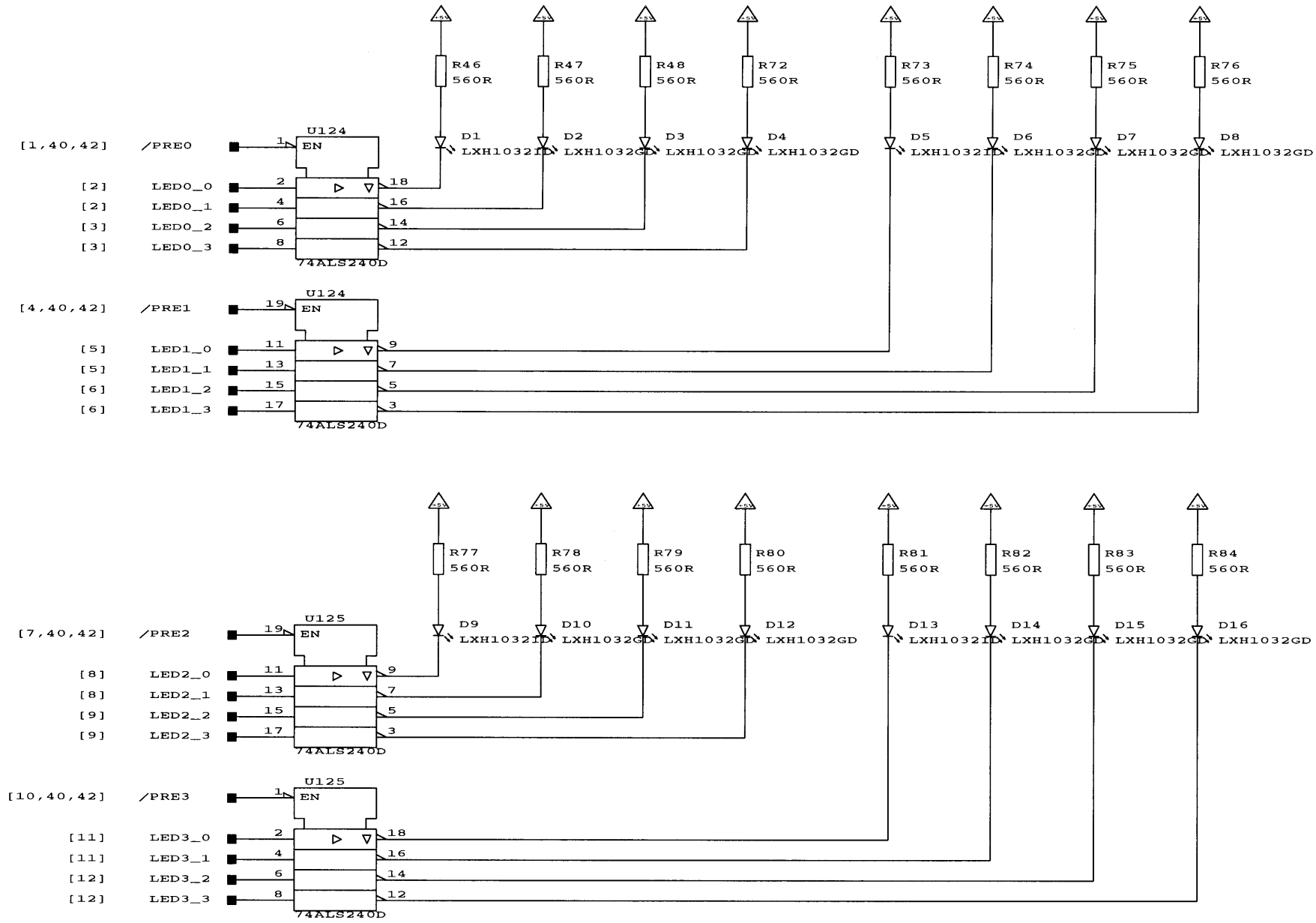

X30
SAD3_0 A1
SAD3_1 A2
SAD3_2 A3
SAD3_3 A4
SAD3_4 A5
SAD3_5 A6
SAD3_6 A7
SAD3_7 A8
SAD3_8 A9
SAD3_9 A10
SAD3_10 A11
SAD3_11 A12
SAD3_12 A13
SAD3_13 A14
SAD3_14 A15
SAD3_15 A16
SAD3_16 A17
SAD3_17 A18
SAD3_18 A19
SAD3_19 A20
SAD3_20 A21
SAD3_21 A22
SAD3_22 A23
SAD3_23 A24
SAD3_24 A25
3M955515
X31
SAD3_25 A1
SAD3_26 A2
SAD3_27 A3
SAD3_28 A4
SAD3_29 A5
SAD3_30 A6
SAD3_31 A7
SAD3_32 A8
SAD3_33 A9
SAD3_34 A10
SAD3_35 A11
SAD3_36 A12
SAD3_37 A13
SAD3_38 A14
SAD3_39 A15
SAD3_40 A16
SAD3_41 A17
SAD3_42 A18
SAD3_43 A19
SAD3_44 A20
SAD3_45 A21
SAD3_46 A22
SAD3_47 A23
SAD3_48 A24
SAD3_49 A25
3M955515
X32
SAD3_50 A1
SAD3_51 A2
SAD3_52 A3
SAD3_53 A4
SAD3_54 A5
SAD3_55 A6
SAD3_56 A7
SAD3_57 A8
SAD3_58 A9
SAD3_59 A10
SAD3_60 A11
SAD3_61 A12
SAD3_62 A13
SAD3_63 A14
SADP3_0 A15
SADP3_1 A16
SADP3_2 A17
SADP3_3 A18
SADP3_4 A19
SADP3_5 A20
SADP3_6 A21
SADP3_7 A22
SCMD3_0 A23
SCMD3_1 A24
SCMD3_2 A25
3M955515
X33
SCMD3_3 A1
SCMD3_4 A2
SCMD3_5 A3
SCMD3_6 A4
SCMD3_7 A5
SCMD3_8 A6
[11,12] SCMDP3 A7
[15] /VALIN3 A8
[11,12] /VALOUT3 A9
[11] /RELEASE3 A10
[11,12,15] /WRRDY3 A11
[11,12,15] /RDRDY3 A12
[15] /IVDACK3 A13
[15] /IVDERR3 A14
[11] /EXTRQ3 A15
[11,15] TCLK3_0 A16
[12] TCLK3_1 A17
[11] RCLK3_0 A18
[12] RCLK3_1 A19
[15] /INT3 A20
/NMI3 A21
A22
[11,12,40] SRE3 A23
[1,4,7,14] C2MS A24
[1,4,7,14] C262MS A25
3M955515
R56
1K
SAD3_0 1K R?
SAD3_1 1K R?
SAD3_2 1K R?
SAD3_3 1K R?
SAD3_4 1K R?
SAD3_5 1K R?
SAD3_6 1K R?
SAD3_7 1K R?
SAD3_8 1K R?
SAD3_9 1K R?
SAD3_10 1K R?
SAD3_11 1K R?
SAD3_12 1K R?
SAD3_13 1K R?
SAD3_14 1K R?
SAD3_15 1K R?
SAD3_16 1K R?
SAD3_17 1K R?
SAD3_18 1K R?
SAD3_19 1K R?
SAD3_20 1K R?
SAD3_21 1K R?
SAD3_22 1K R?
SAD3_23 1K R?
SAD3_24 1K R?
SAD3_25 1K R?
SAD3_26 1K R?
SAD3_27 1K R?
SAD3_28 1K R?
SAD3_29 1K R?
SAD3_30 1K R?
SAD3_31 1K R?
SAD3_32 1K R?
SAD3_33 1K R?
SAD3_34 1K R?
SAD3_35 1K R?
SAD3_36 1K R?
SAD3_37 1K R?
SAD3_38 1K R?
SAD3_39 1K R?
SAD3_40 1K R?
SAD3_41 1K R?
SAD3_42 1K R?
SAD3_43 1K R?
SAD3_44 1K R?
SAD3_45 1K R?
SAD3_46 1K R?
SAD3_47 1K R?
SAD3_48 1K R?
SAD3_49 1K R?
SAD3_50 1K R?
SAD3_51 1K R?
SAD3_52 1K R?
SAD3_53 1K R?
SAD3_54 1K R?
SAD3_55 1K R?
SAD3_56 1K R?
SAD3_57 1K R?
SAD3_58 1K R?
SAD3_59 1K R?
SAD3_60 1K R?
SAD3_61 1K R?
SAD3_62 1K R?
SAD3_63 1K R?
SADP3_0 1K R?
SADP3_1 1K R?
SADP3_2 1K R?
SADP3_3 1K R?
SADP3_4 1K R?
SADP3_5 1K R?
SADP3_6 1K R?
SADP3_7 1K R?
X30
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
3M955515
X31
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
3M955515
X32
+3.3V
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
3M955515
X33
PM_ID3_[3:0]
[42]
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
PM_ID3_0 B20
PM_ID3_1 B21
PM_ID3_2 B22
PM_ID3_3 B23
[13,40,42] /PRE3 B24
[1,4,7,39,67] BIG B25
3M955515
dde
Dansk Data Elektronik A/S
Issue 0 96-06-07
Issue 1
Issue 2
Issue 3
CPU305-1 Module
Connectors 3
File: cpu305-1
Page:10 of 72



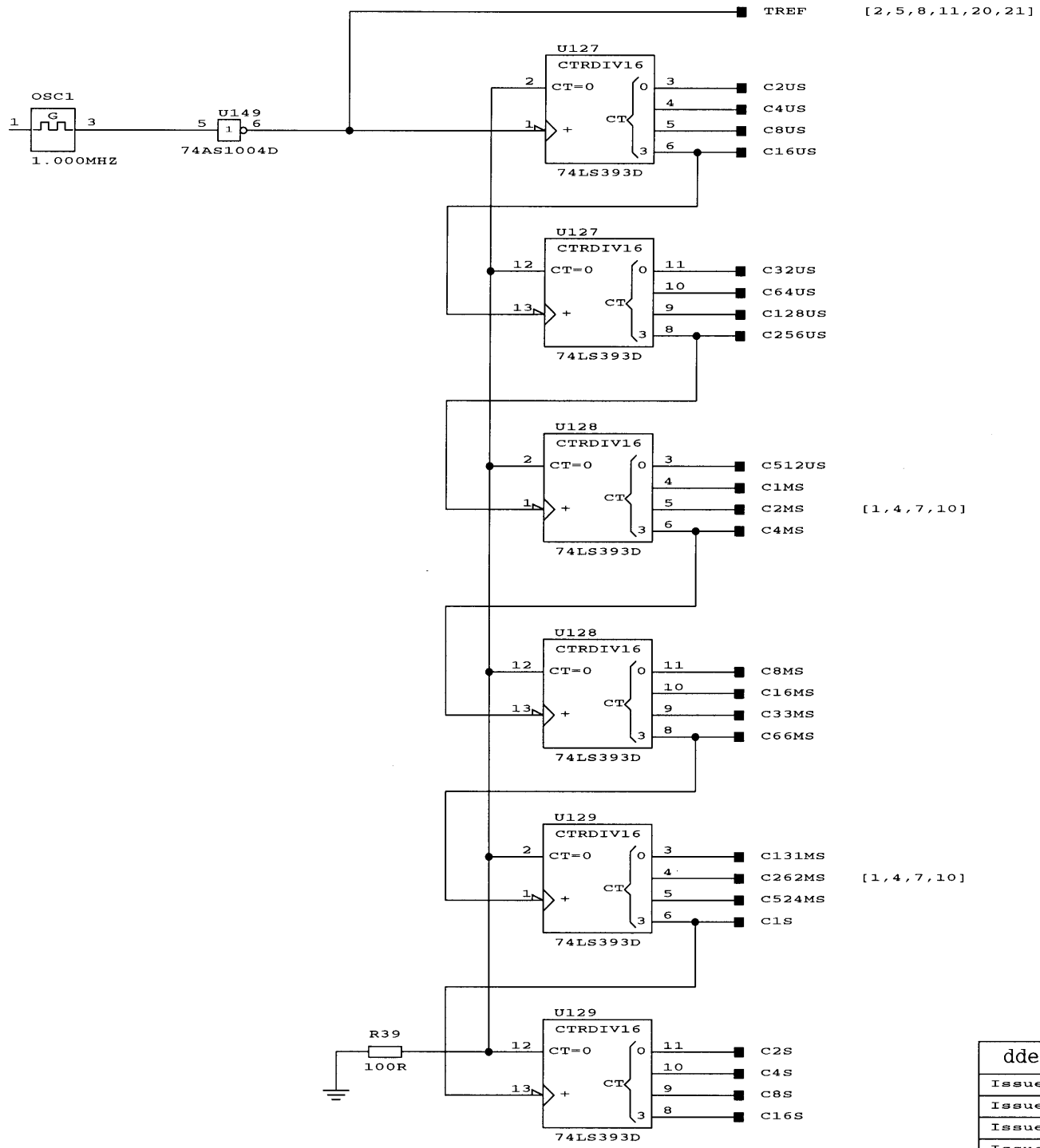
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global agent 3	
Issue 2			
Issue 3		File: cpu305-1	Page:11 of 72



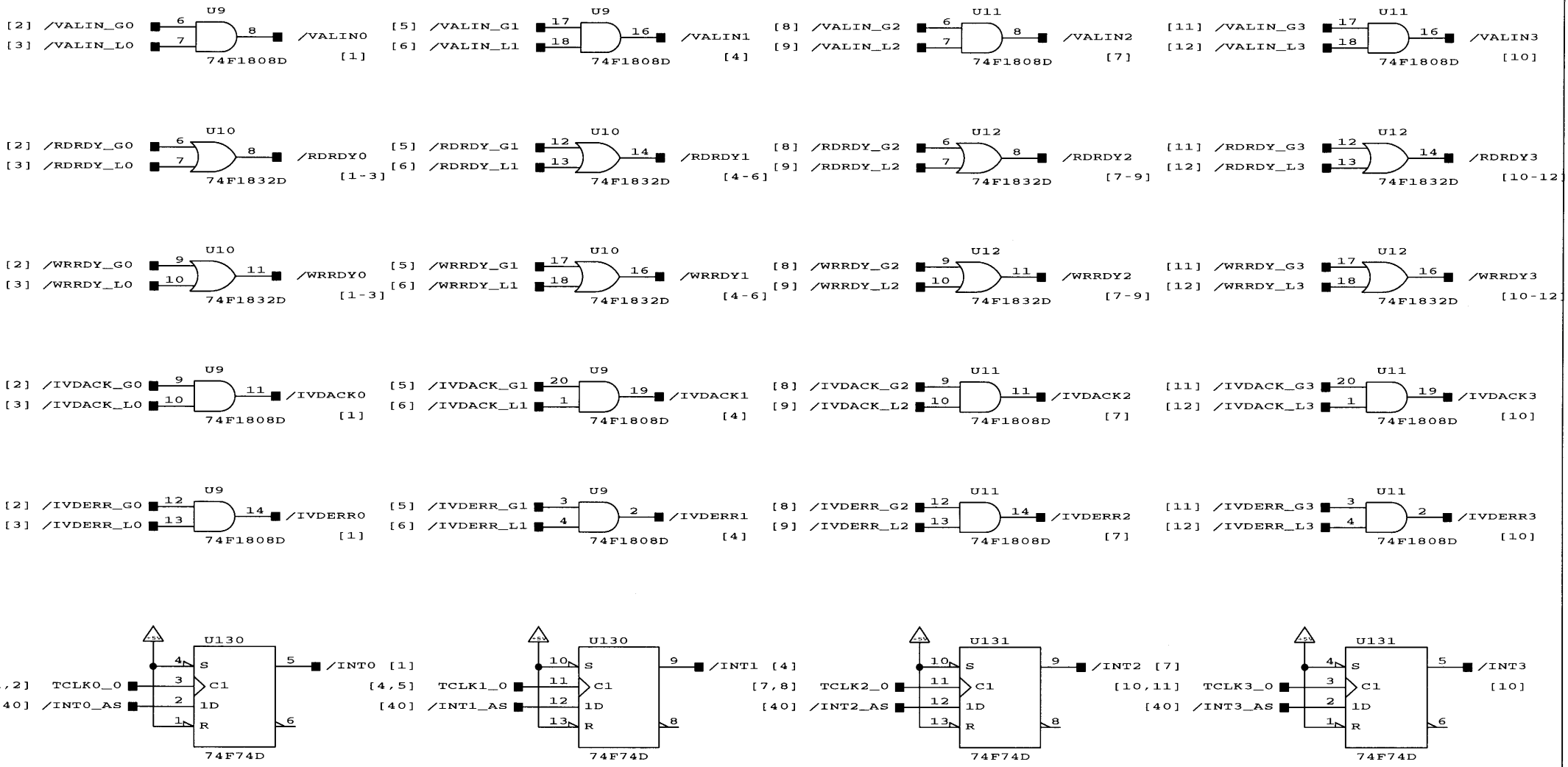
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local agent 3	
Issue 2			
Issue 3		File: cpu305-1	Page:12 of 72



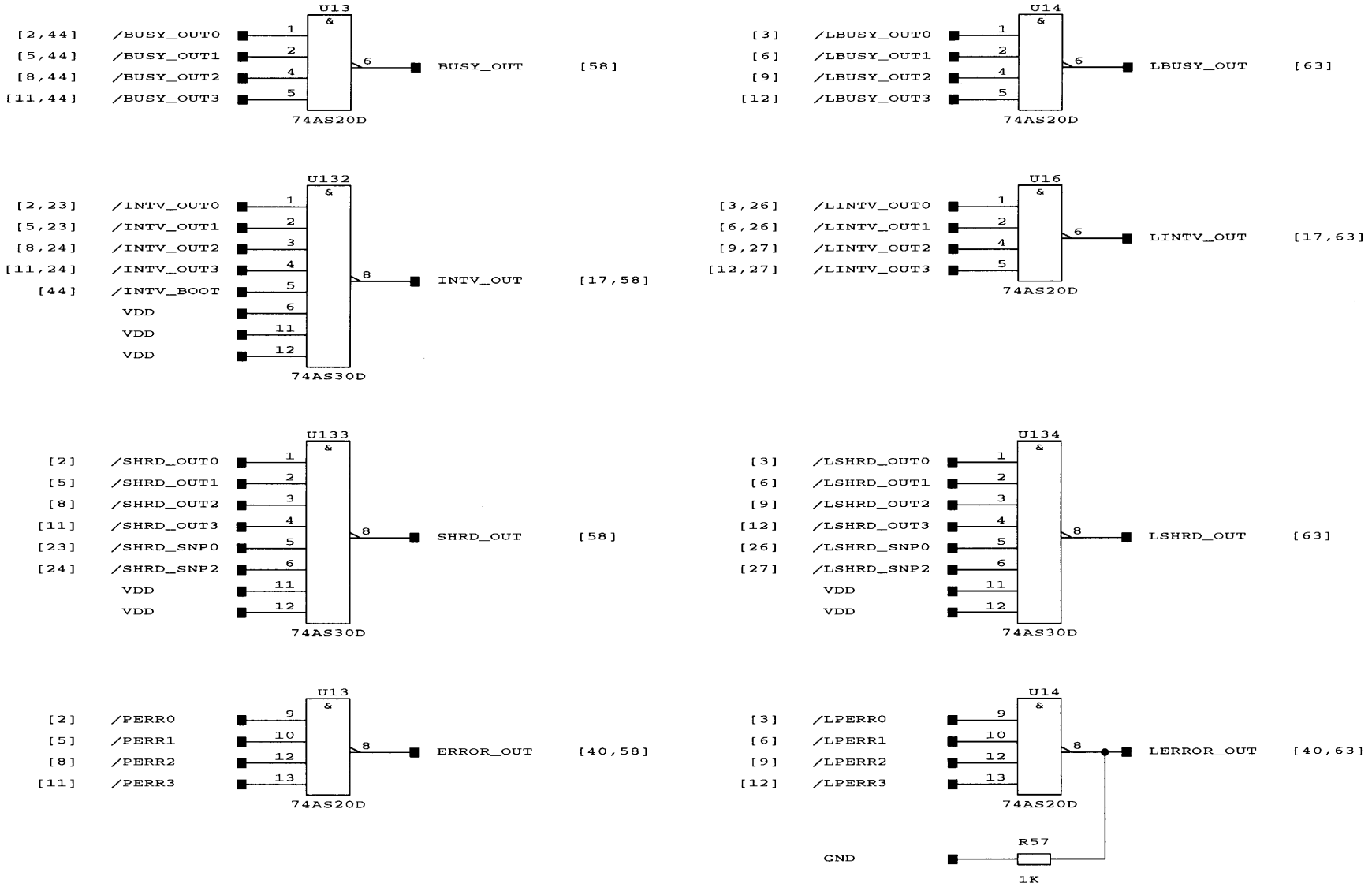
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Programmable LEDs	
Issue 2			
Issue 3		File: cpu305-1	Page:13 of 72



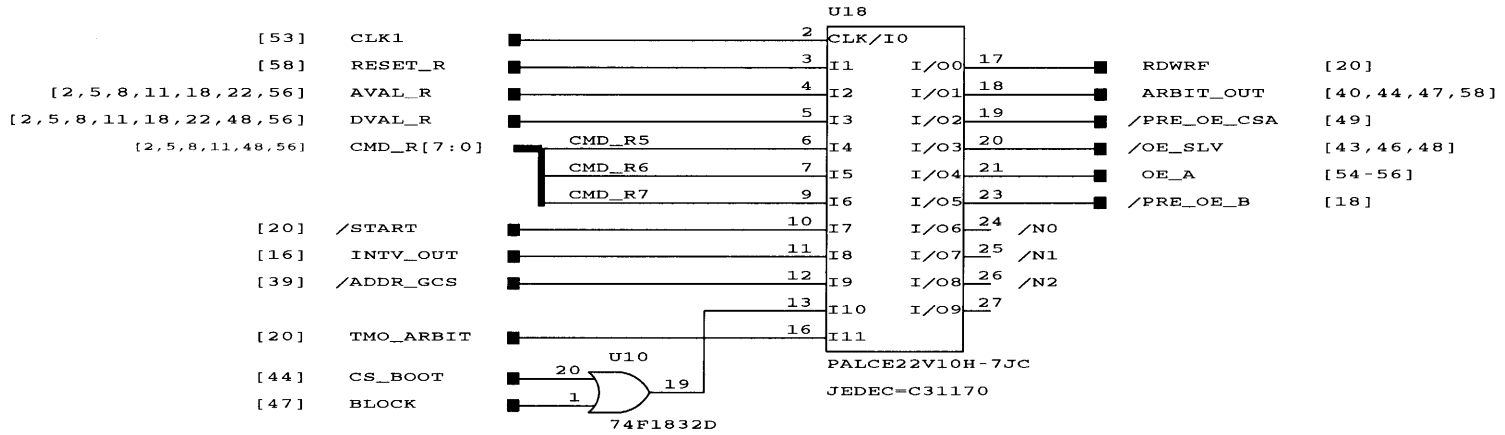
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Reset counters	
Issue 2			
Issue 3		File: cpu305-1	Page:14 of 72

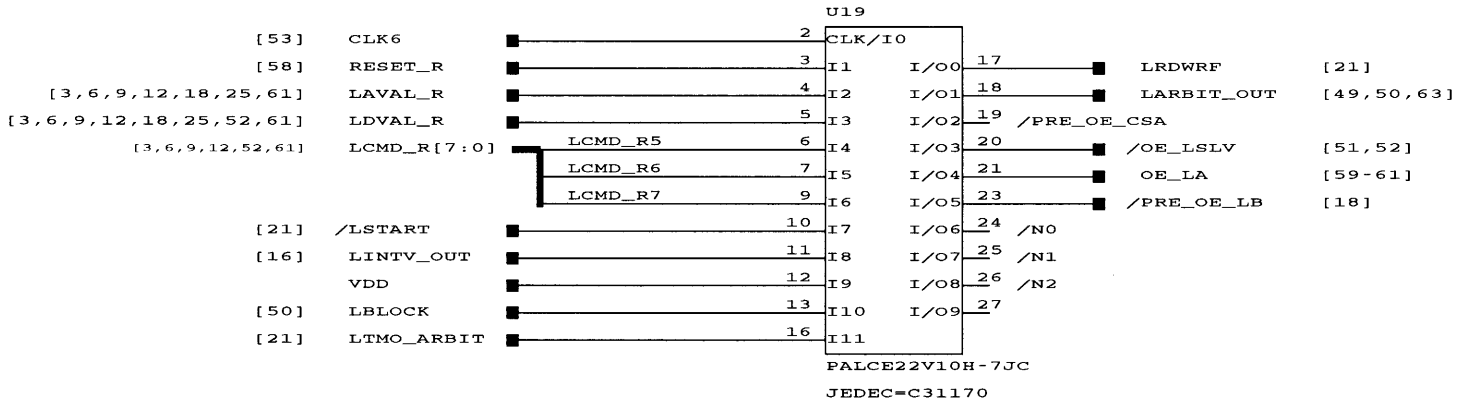


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Gating towards processors	
Issue 2			
Issue 3		File: cpu305-1	Page:15 of 72

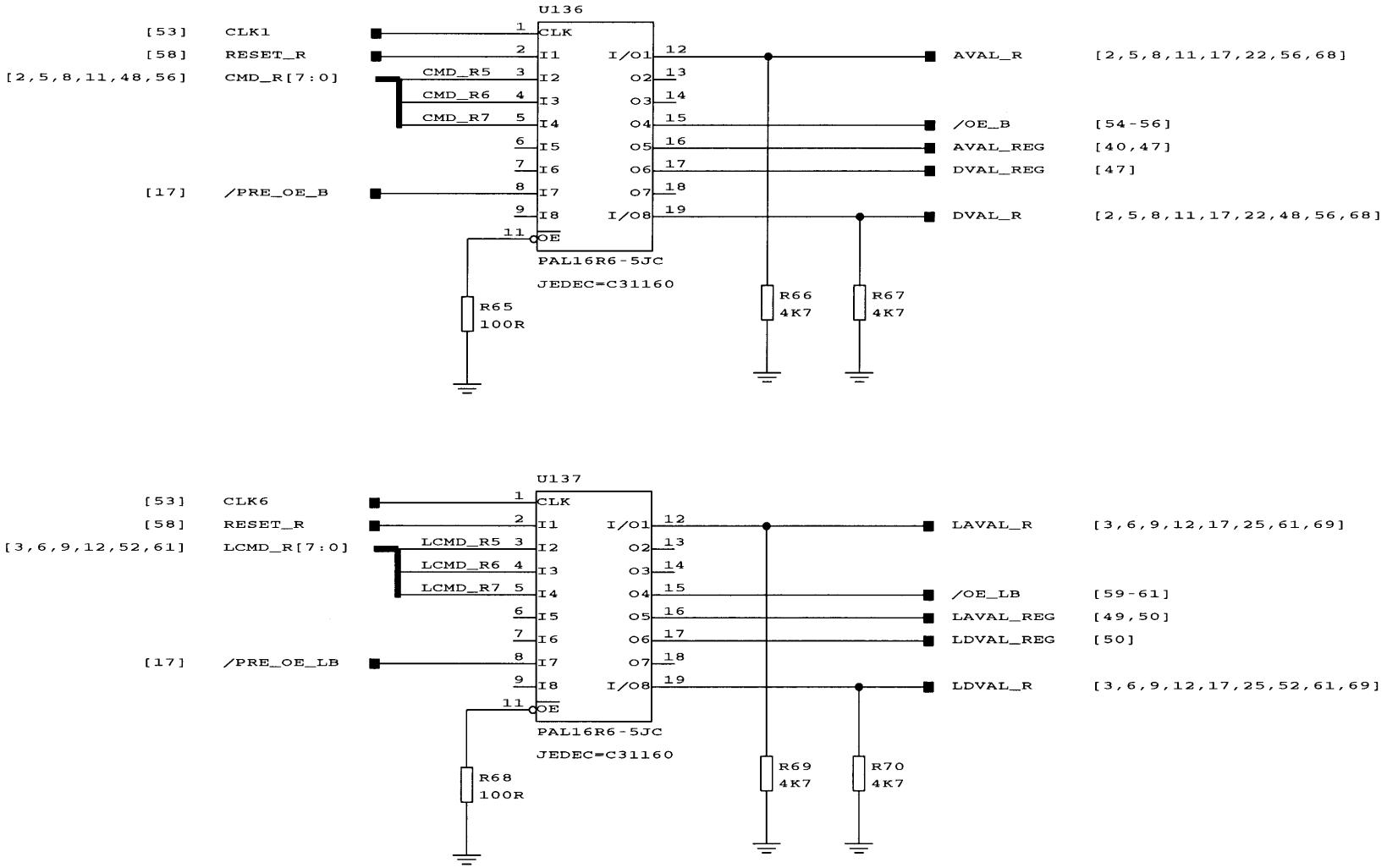


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Gating towards system bus	
Issue 2			
Issue 3		File: cpu305-1	Page:16 of 72

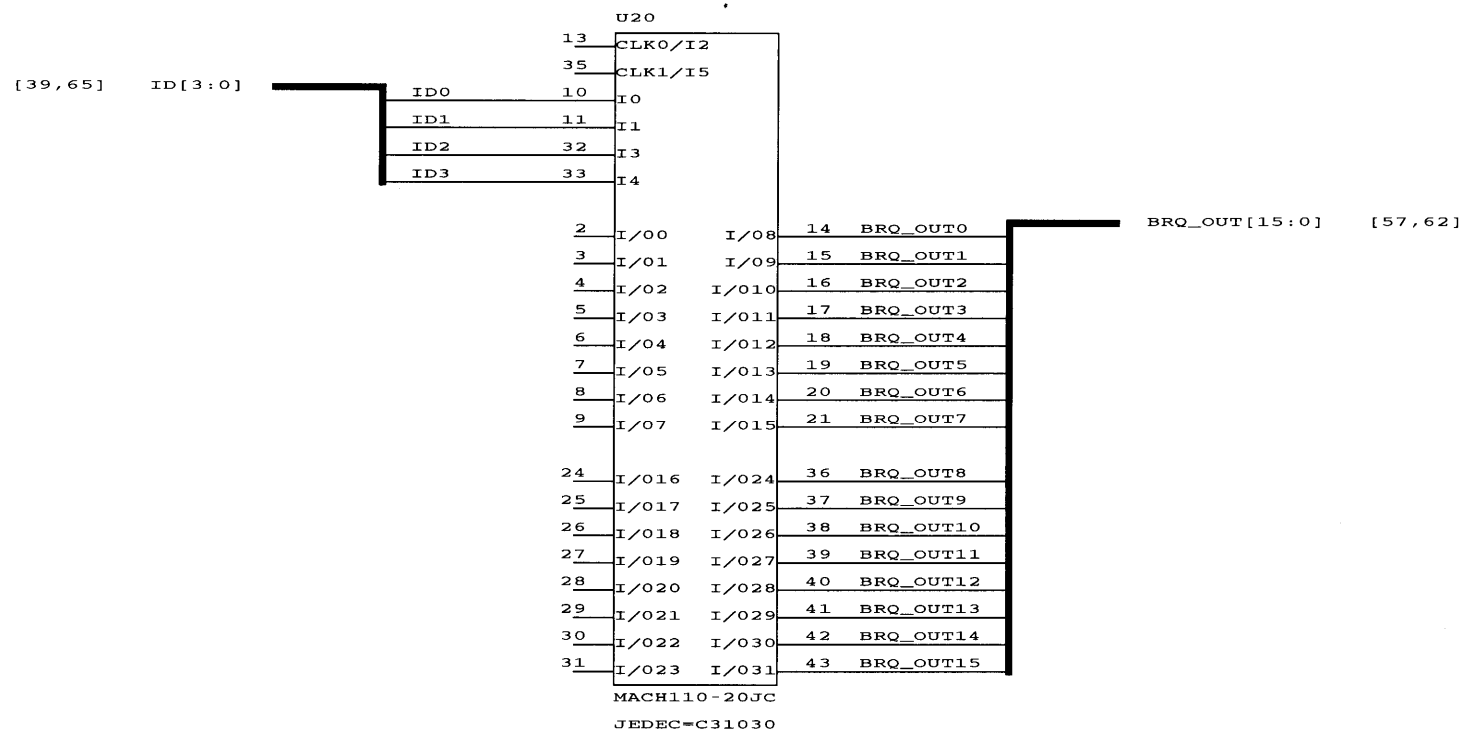




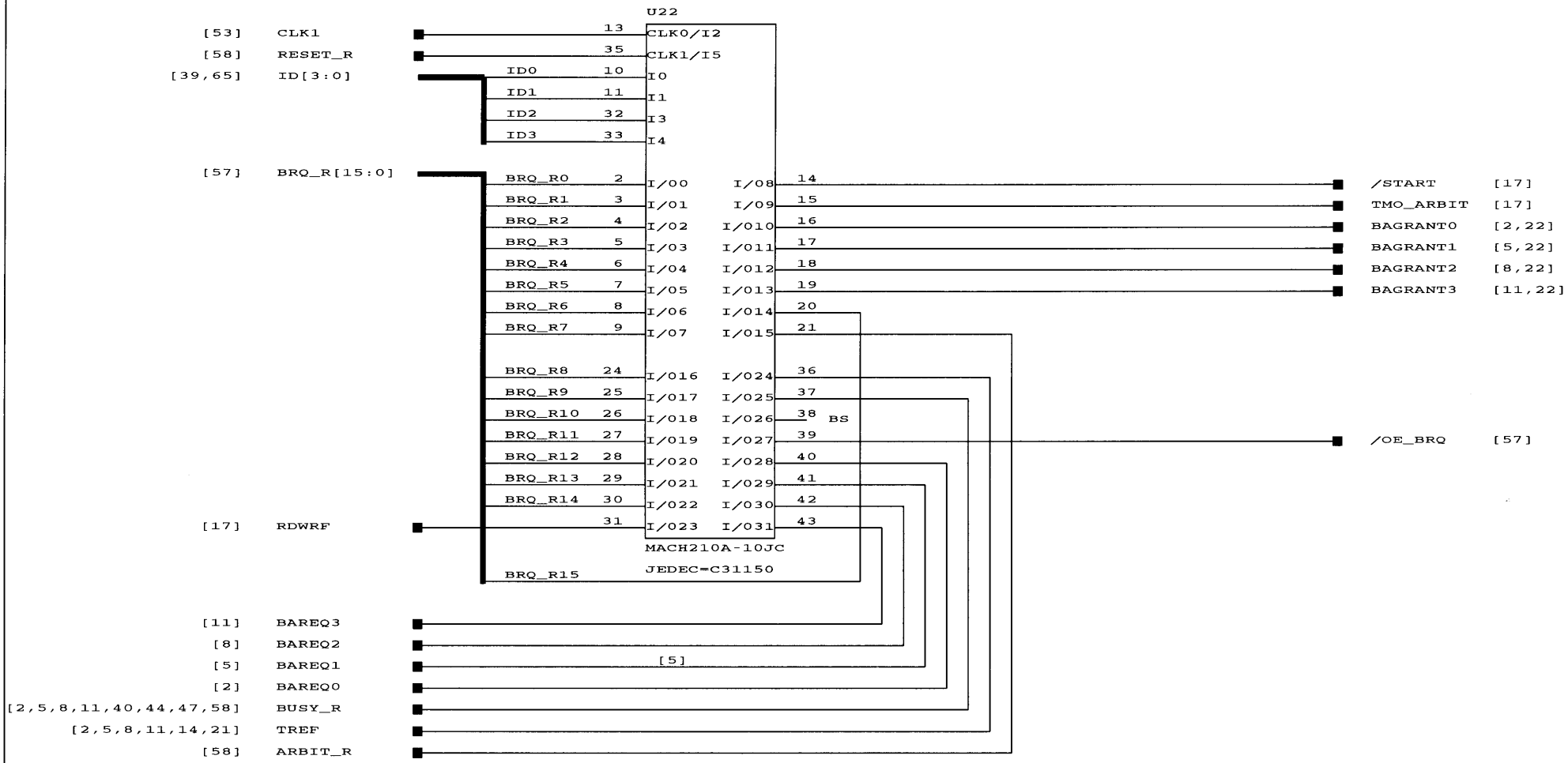
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global and local	
Issue 2		output enable control	
Issue 3		File: cpu305-1	Page:17 of 72



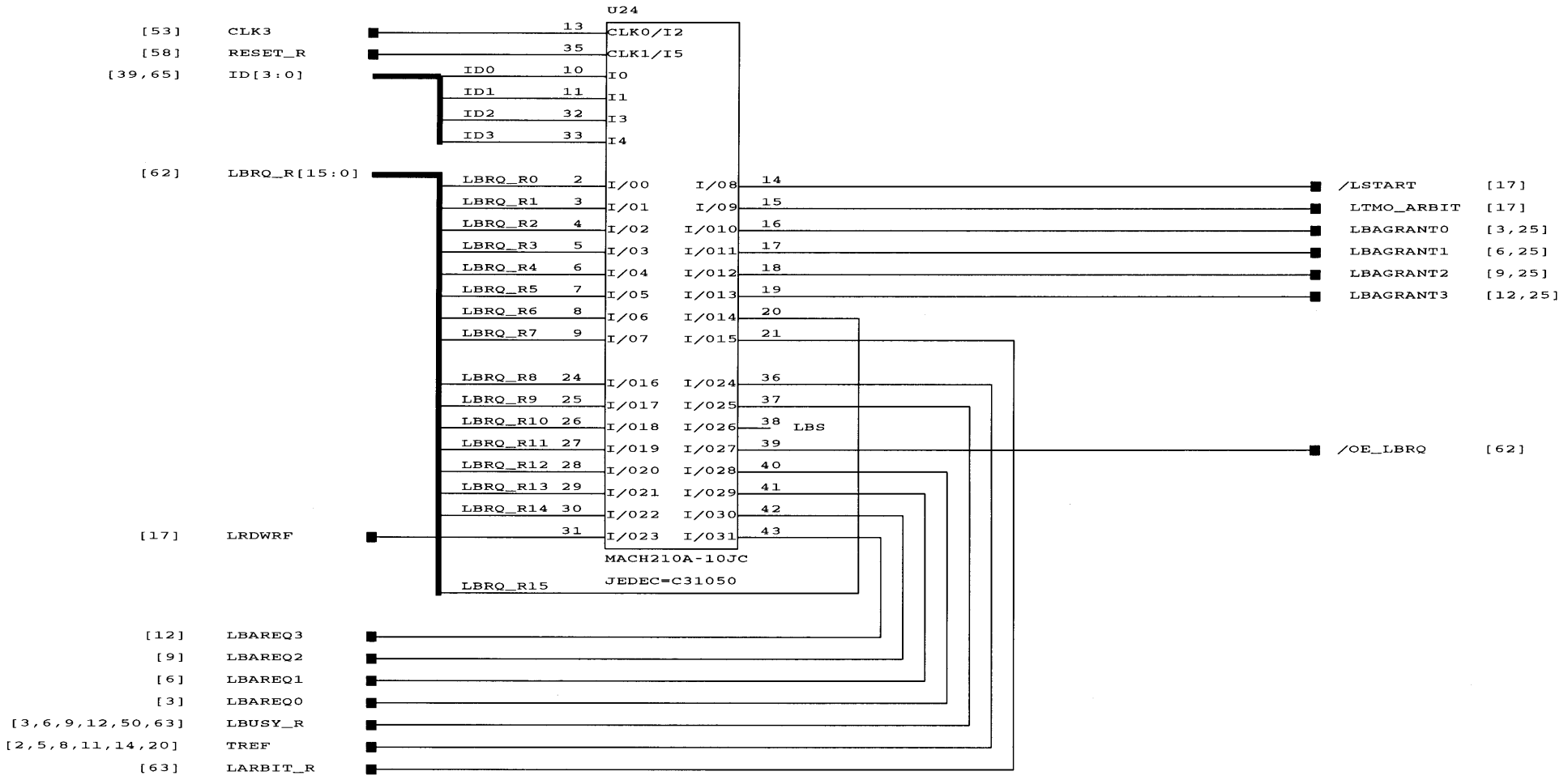
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Pull-down for	
Issue 2		address and data valid	
Issue 3		File: cpu305-1	Page:18 of 72



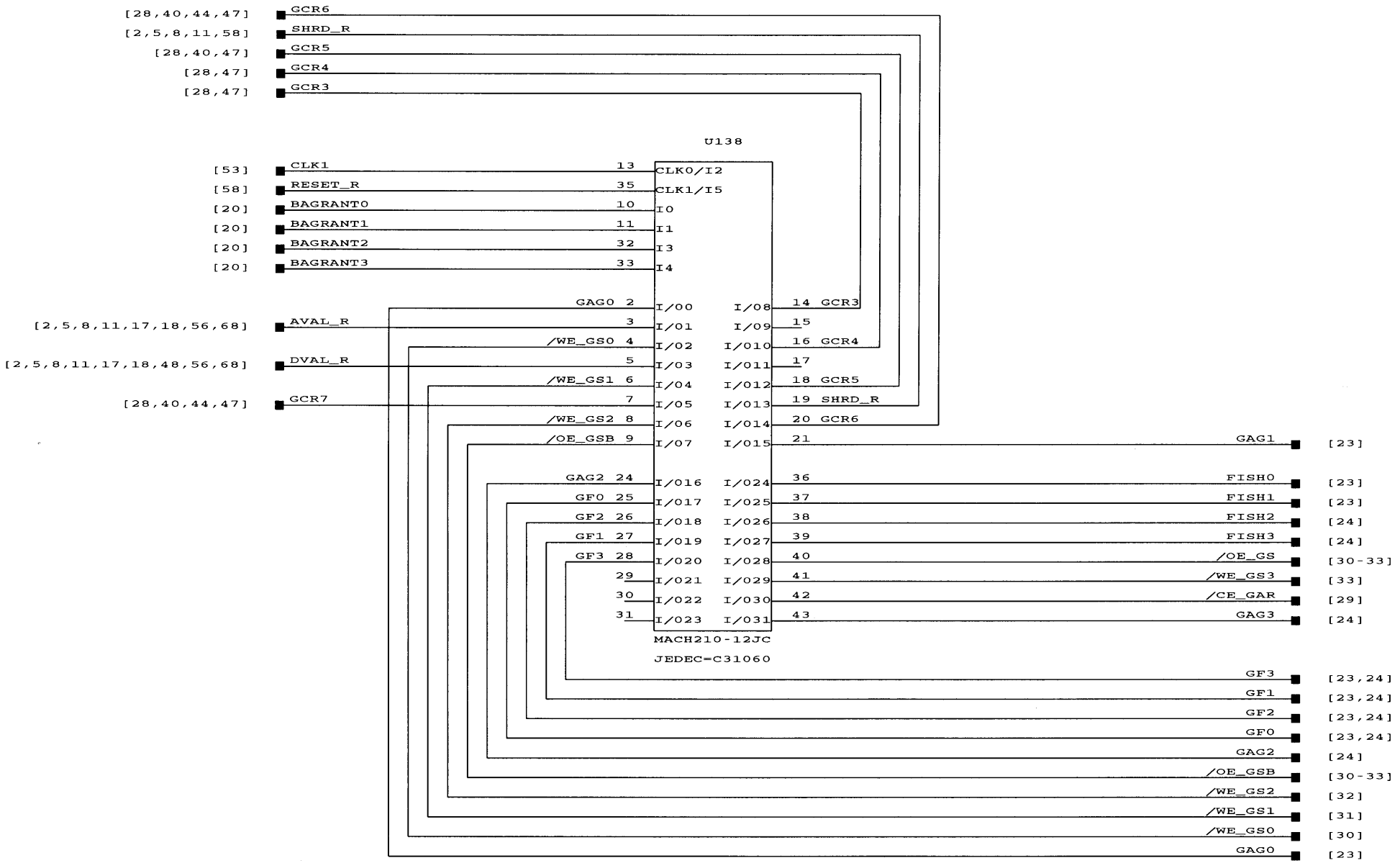
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global and local	
Issue 2		bus request decoder	
Issue 3		File: cpu305-1	Page:19 of 72



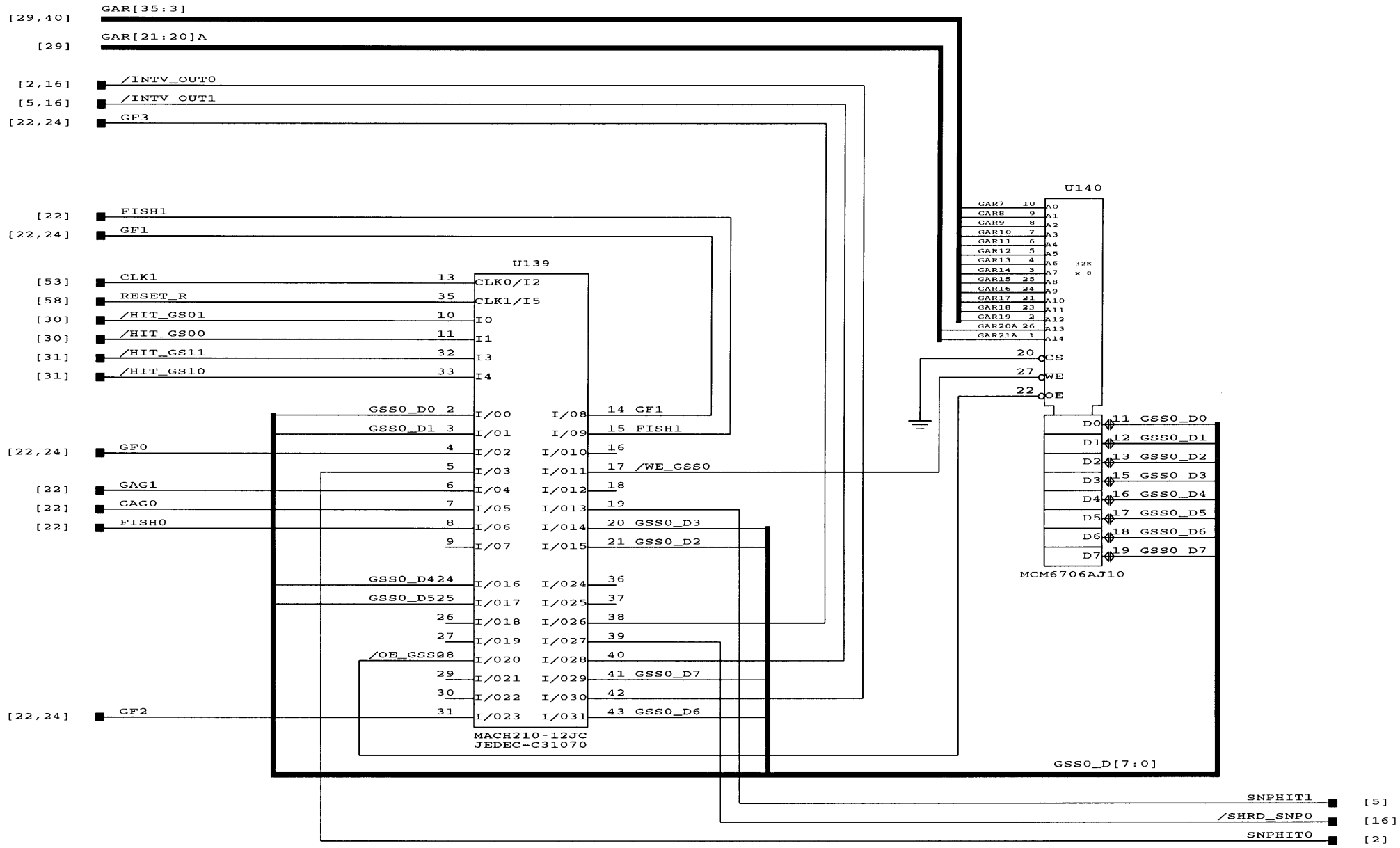
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global bus arbitration	
Issue 2			
Issue 3		File: cpu305-1	Page:20 of 72



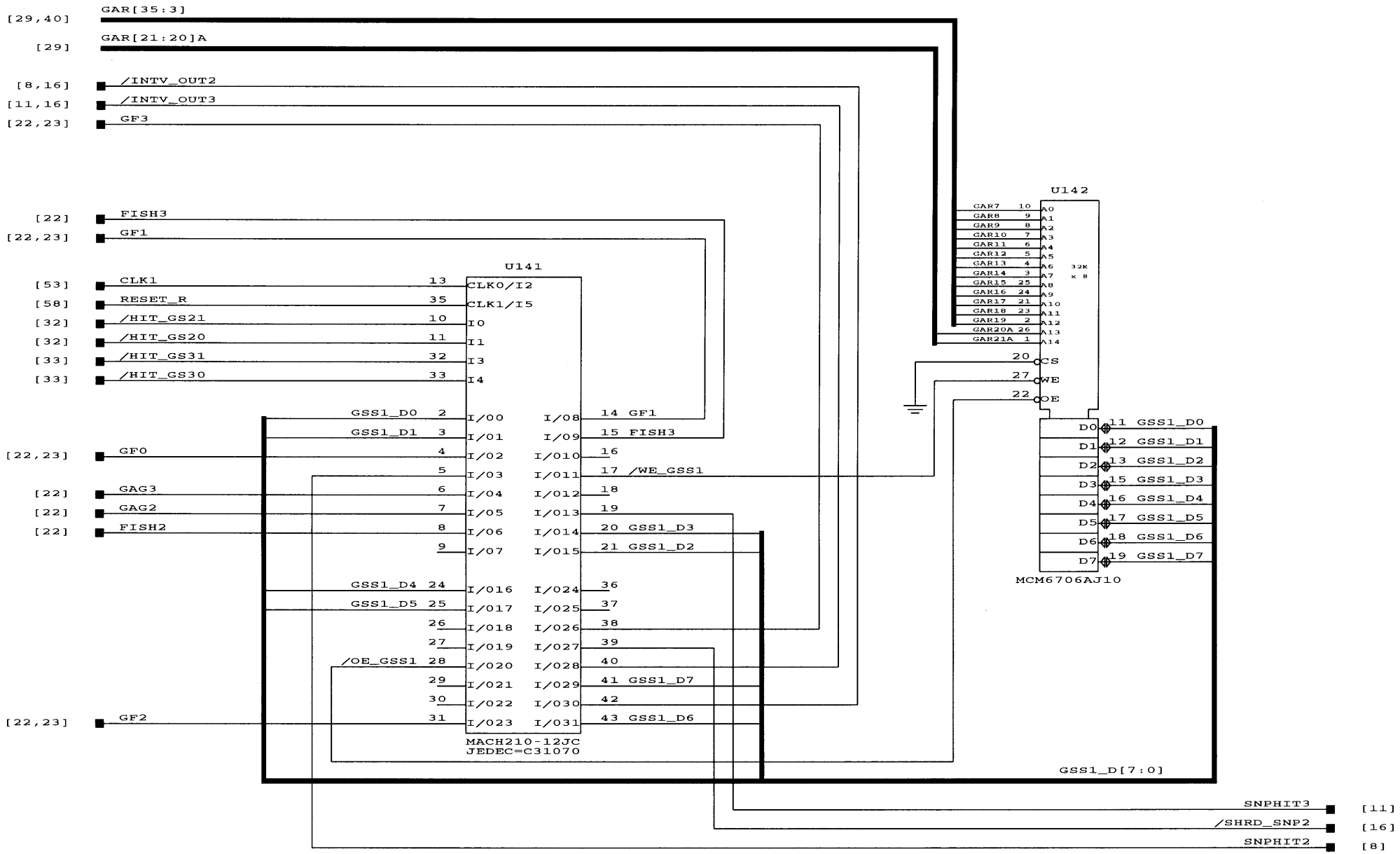
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local bus arbitration	
Issue 2			
Issue 3		File: cpu305-1	Page:21 of 72



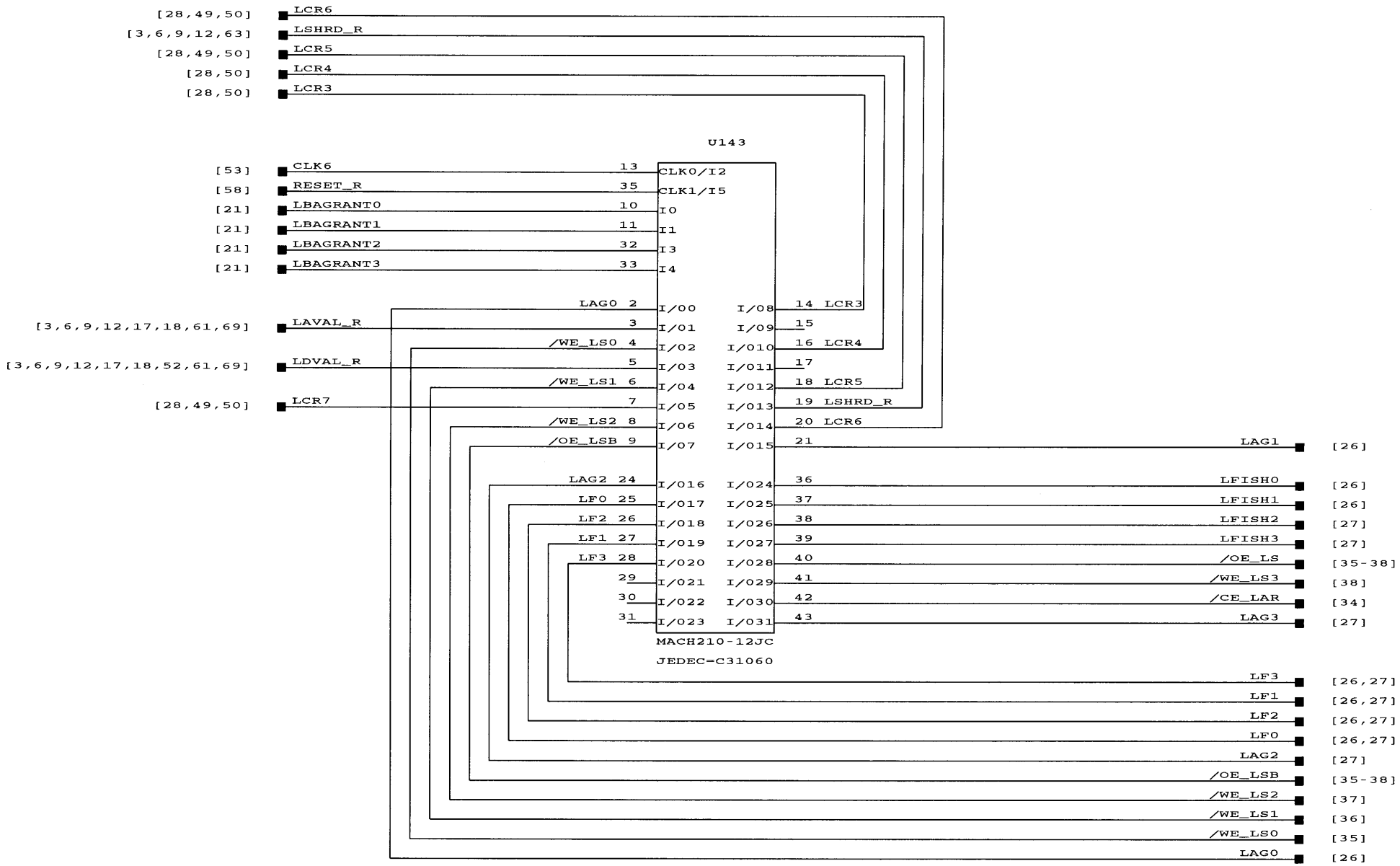
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global snooper control	
Issue 2			
Issue 3		File: cpu305-1	Page:22 of 72



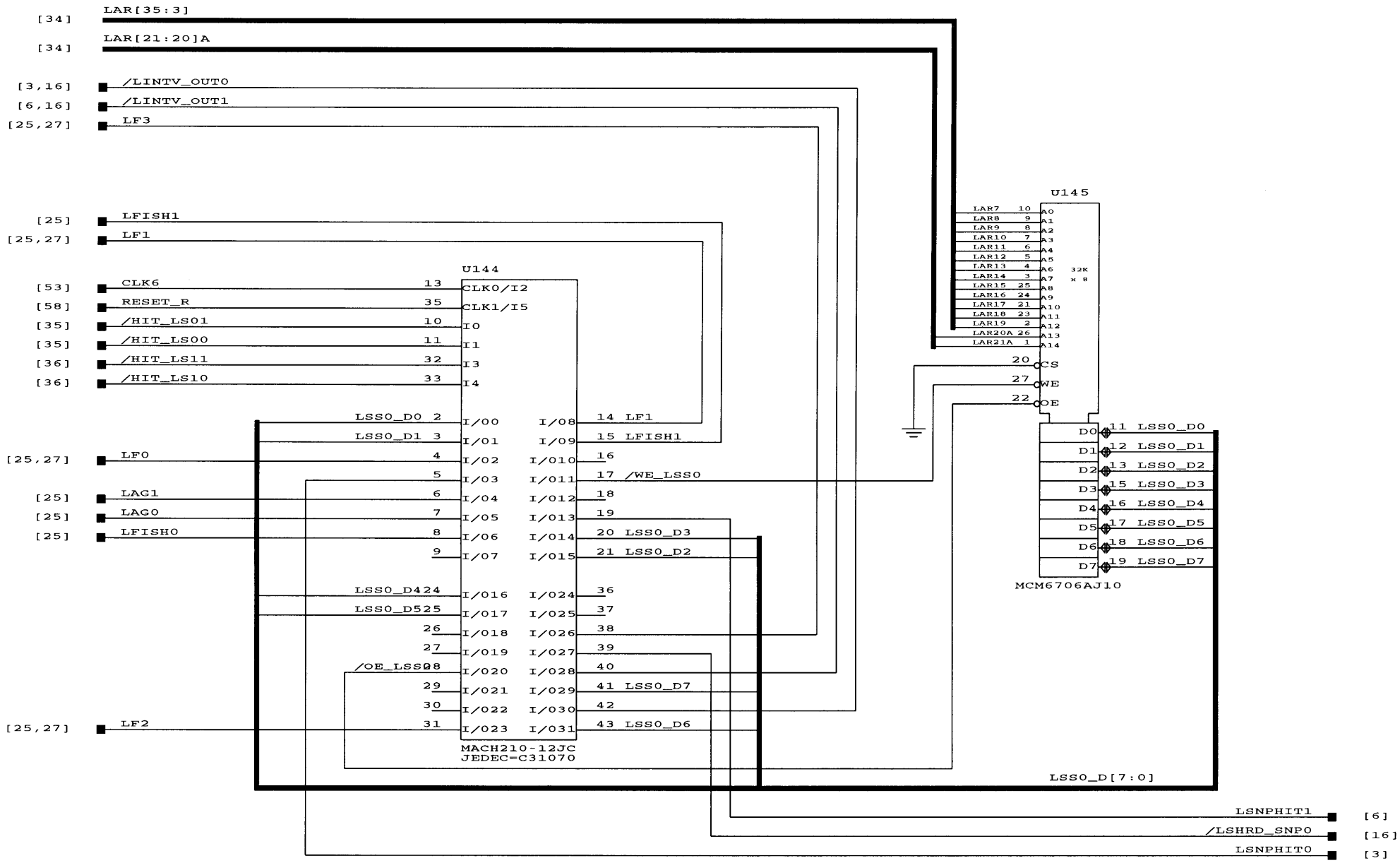
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global Super Snooper 0-1	
Issue 2			
Issue 3		File: cpu305-1	Page:23 of 72



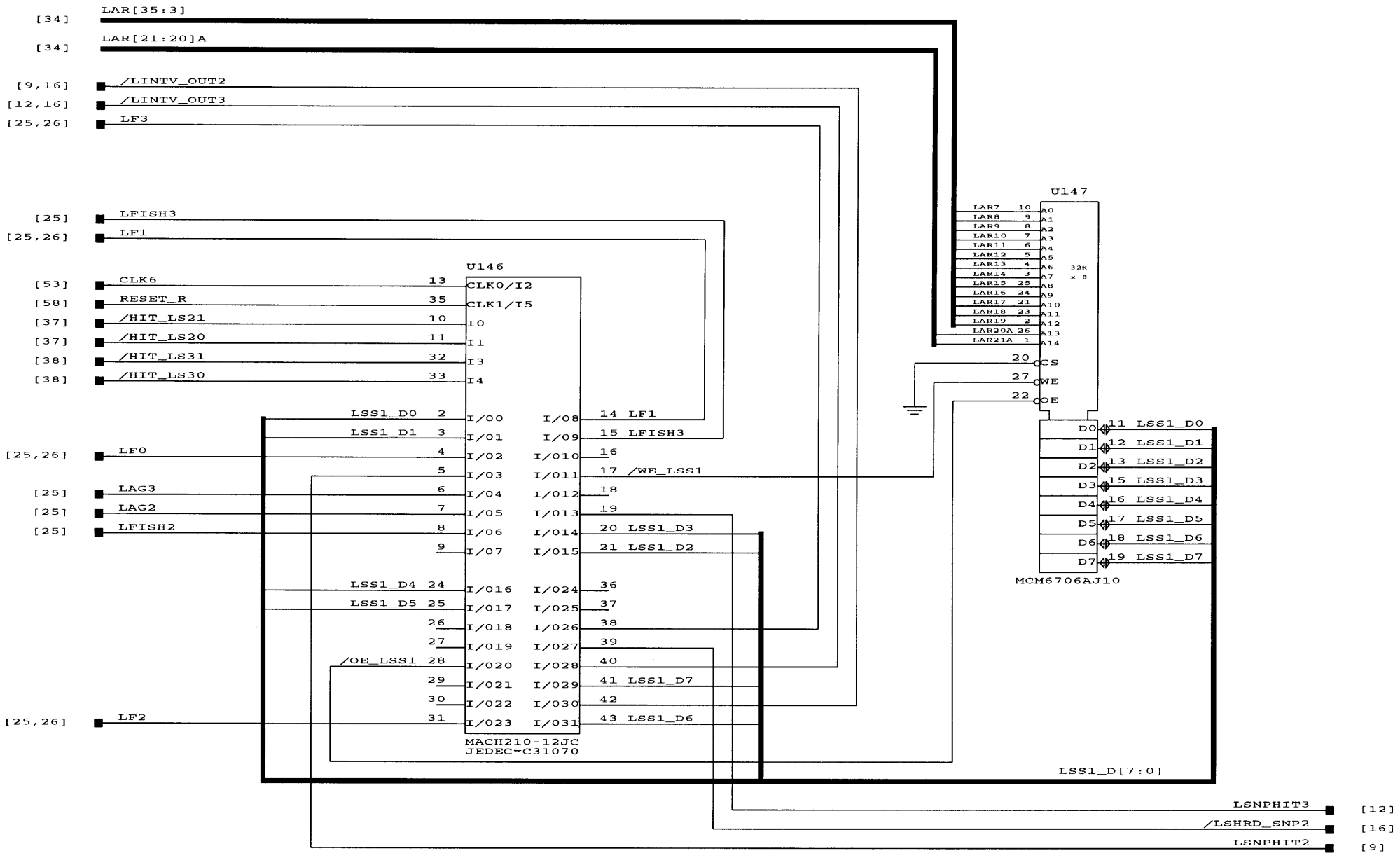
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global Super Snooper 2-3	
Issue 2			
Issue 3		File: cpu305-1	Page:24 of 72



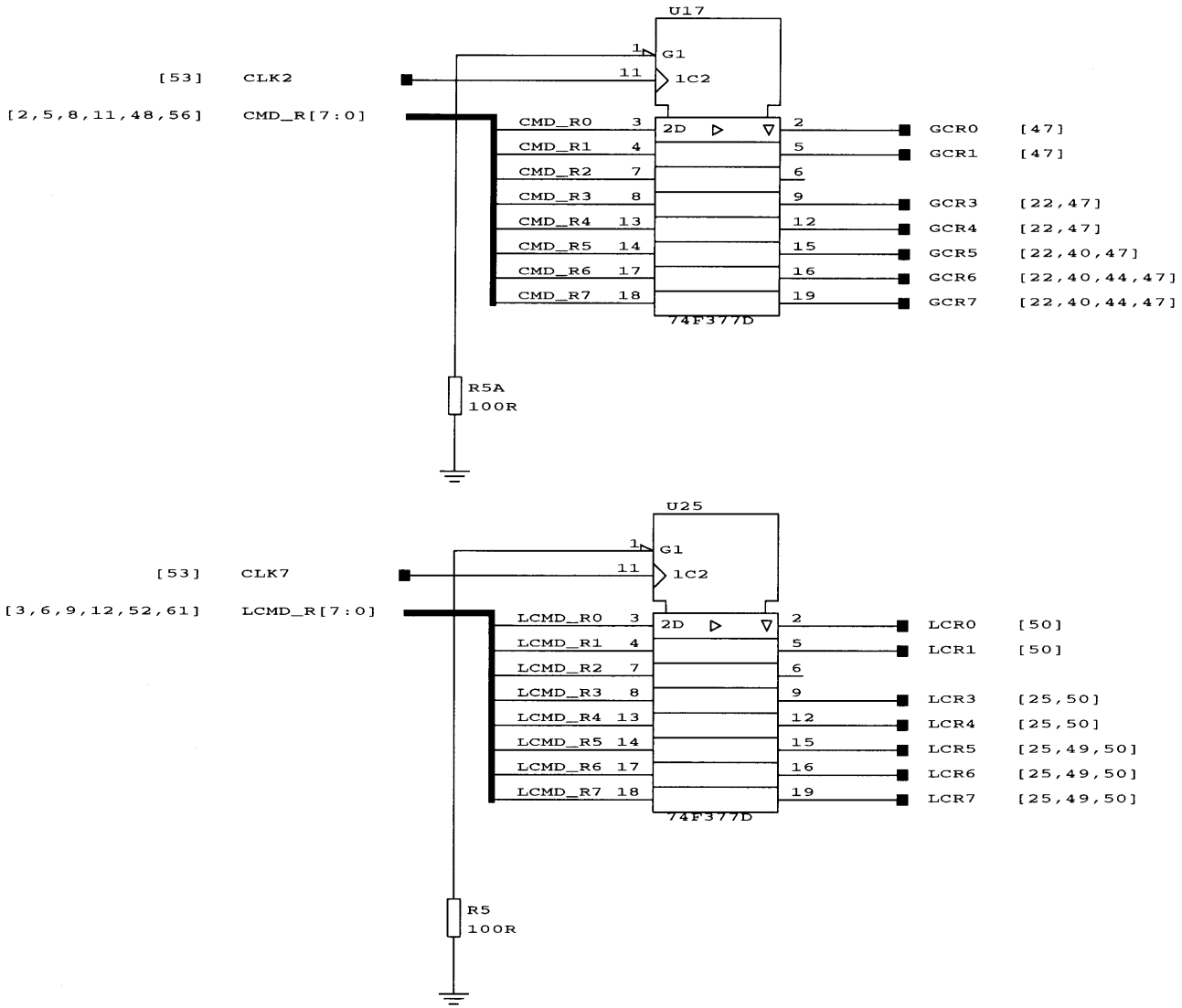
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local snooper control	
Issue 2			
Issue 3		File: cpu305-1	Page:25 of 72



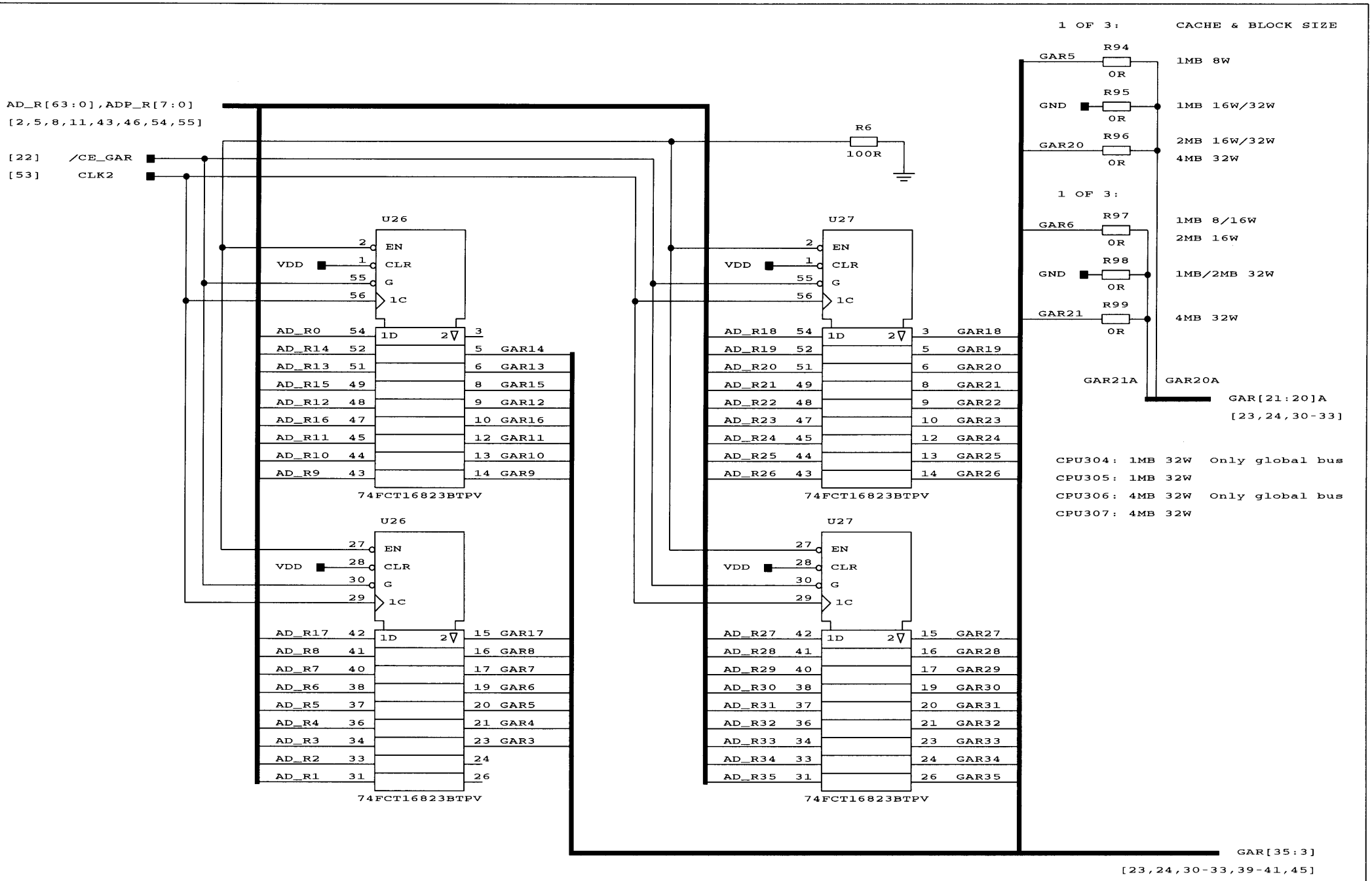
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local Super Snooper 0-1	
Issue 2			
Issue 3		File: cpu305-1	Page:26 of 72



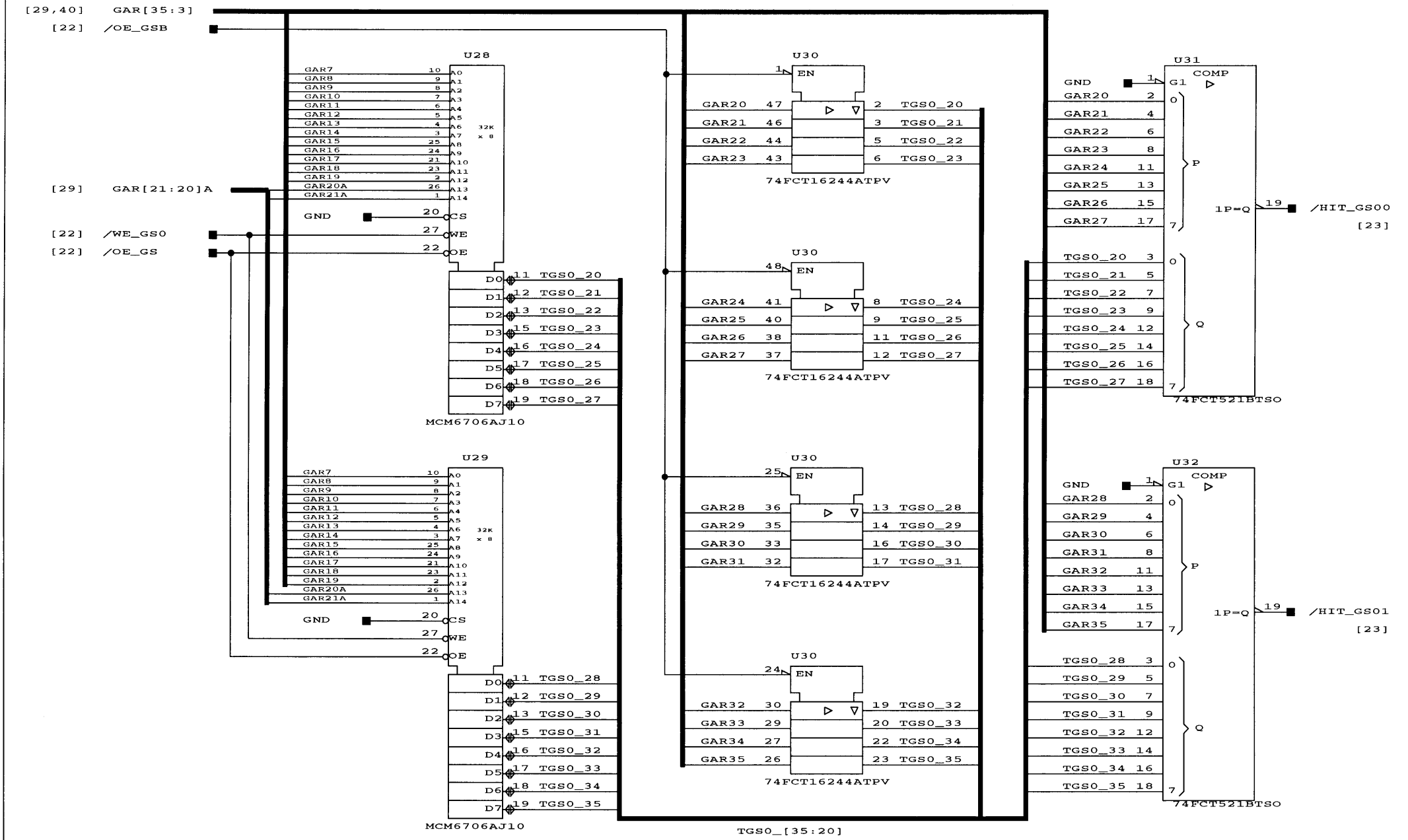
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local Super Snooper 2-3	
Issue 2			
Issue 3		File: cpu305-1	Page:27 of 72



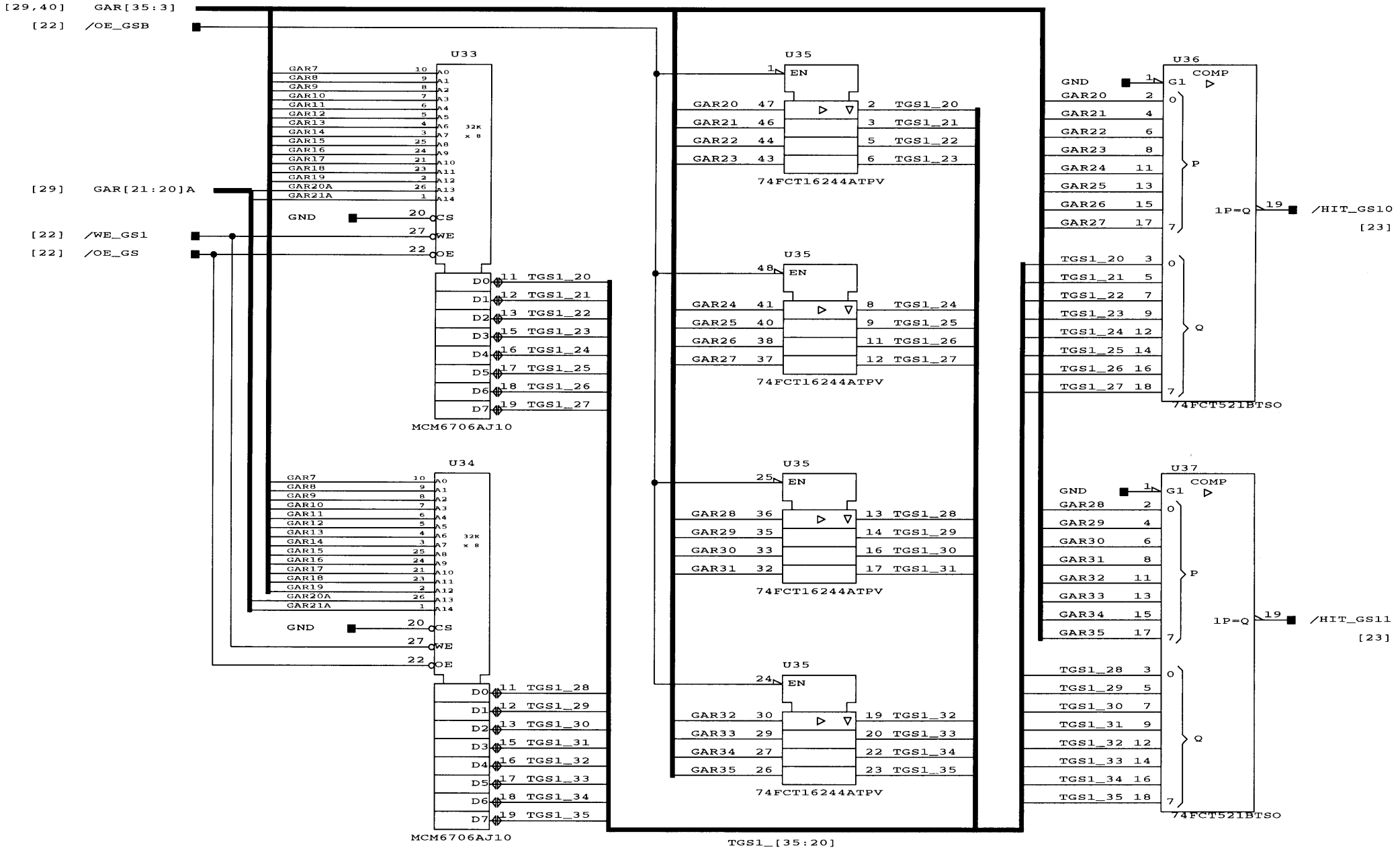
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global and local	
Issue 2		command register	
Issue 3		File: cpu305-1	Page:28 of 72



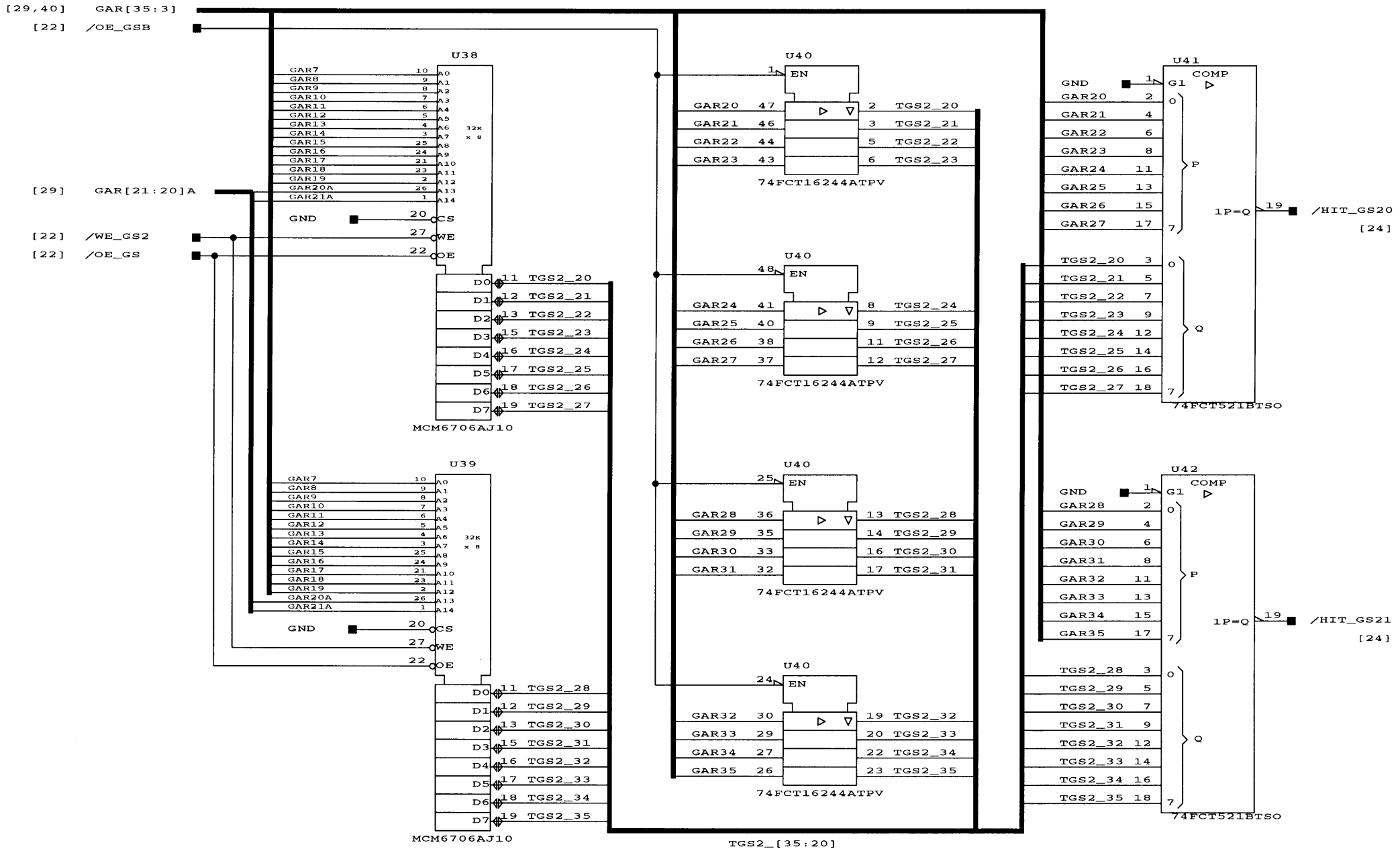
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global address register	
Issue 2			
Issue 3		File: cpu305-1	Page:29 of 72



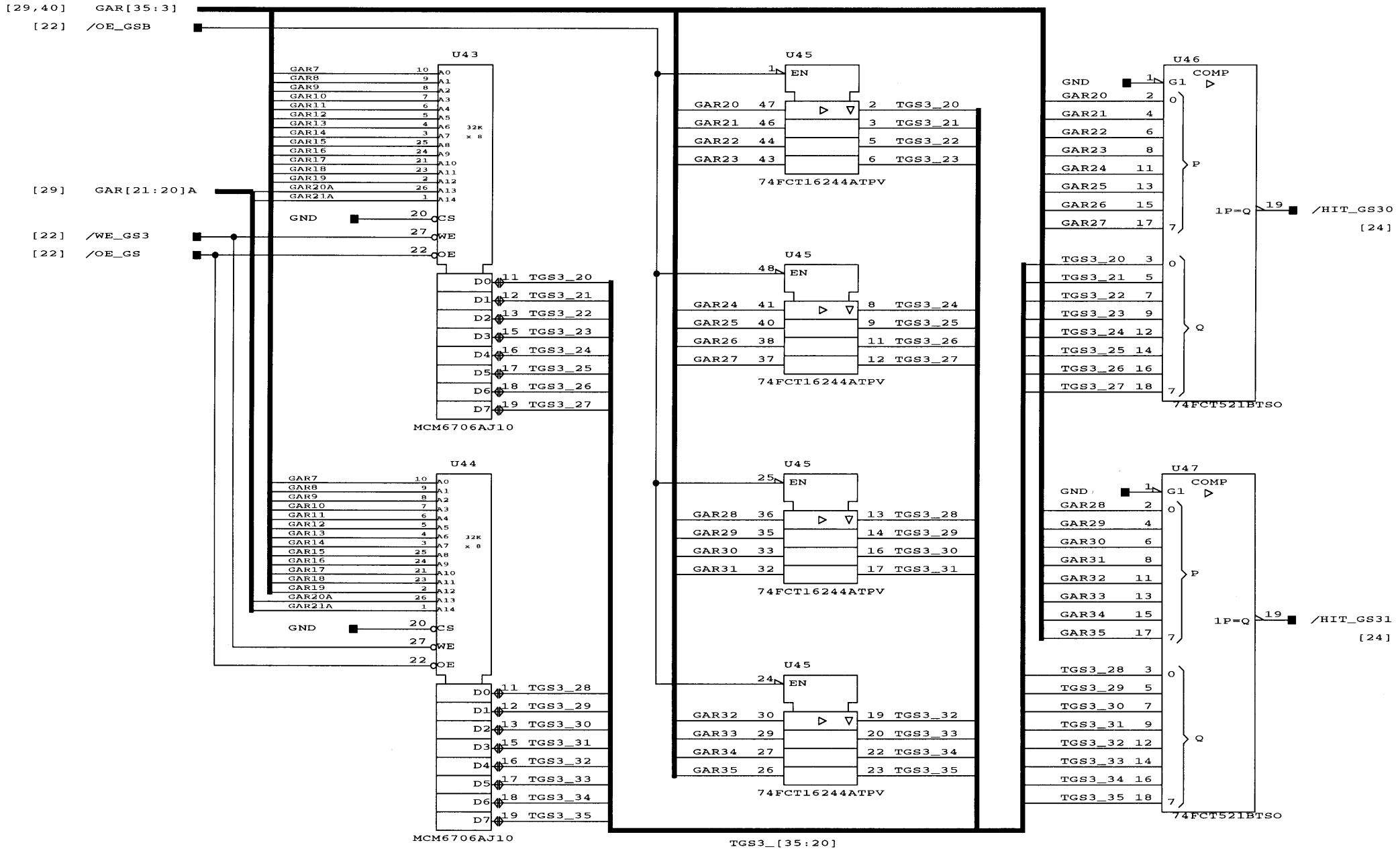
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global snooper 0	
Issue 2			
Issue 3		File: cpu305-1	Page:30 of 72



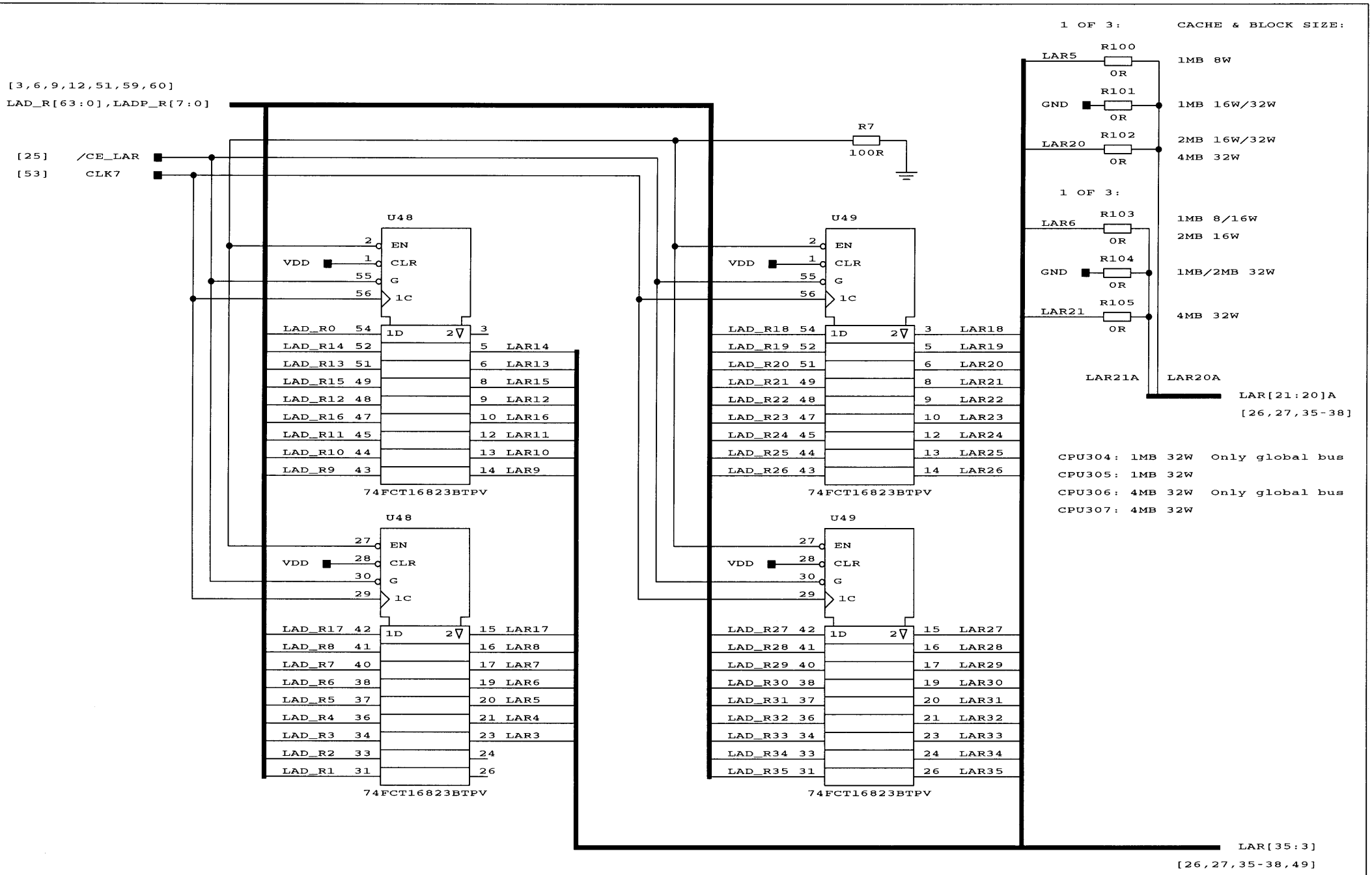
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global snooper 1	
Issue 2			
Issue 3		File: cpu305-1	Page:31 of 72



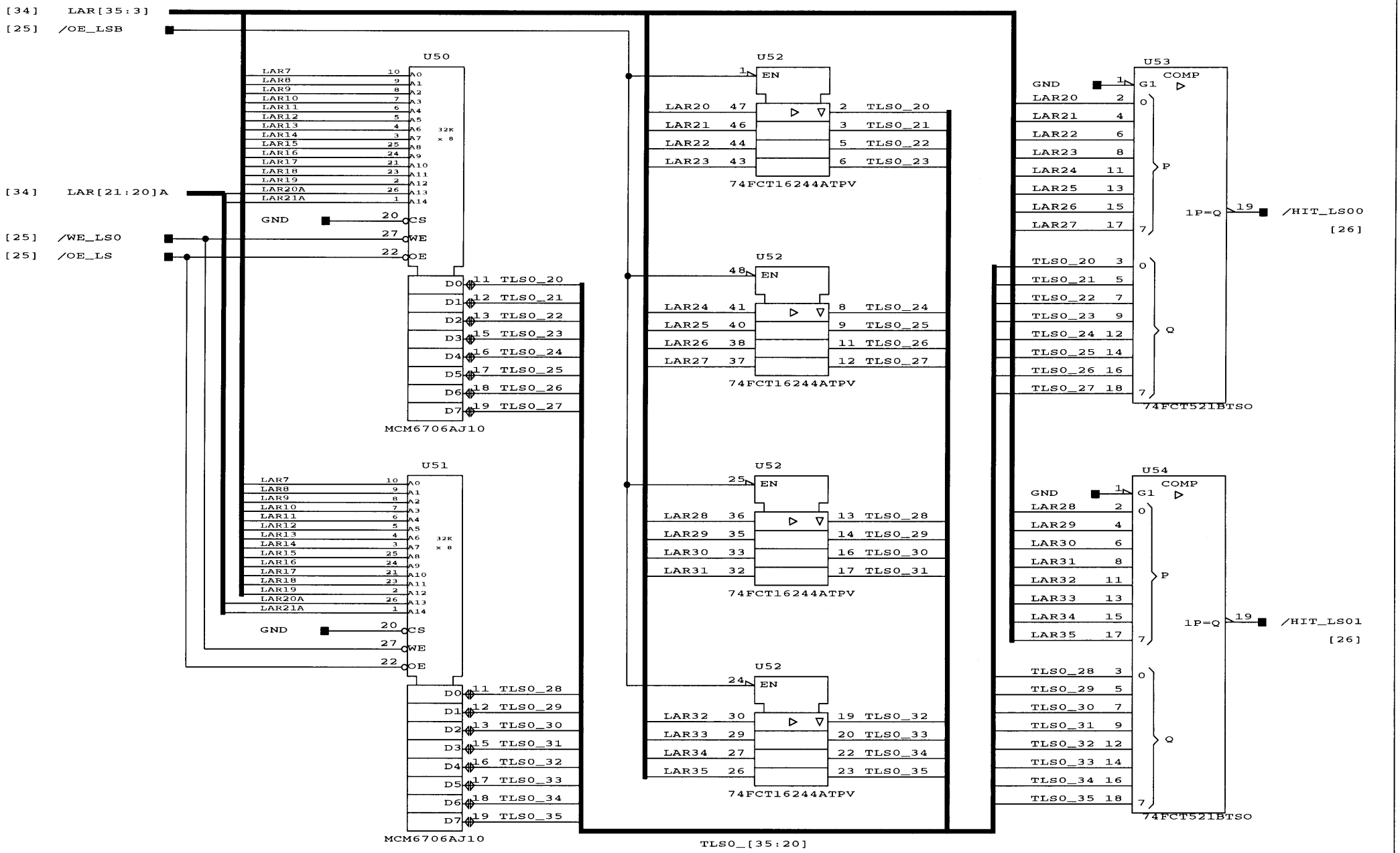
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global snooper 2	
Issue 2			
Issue 3		File: cpu305-1	Page:32 of 72



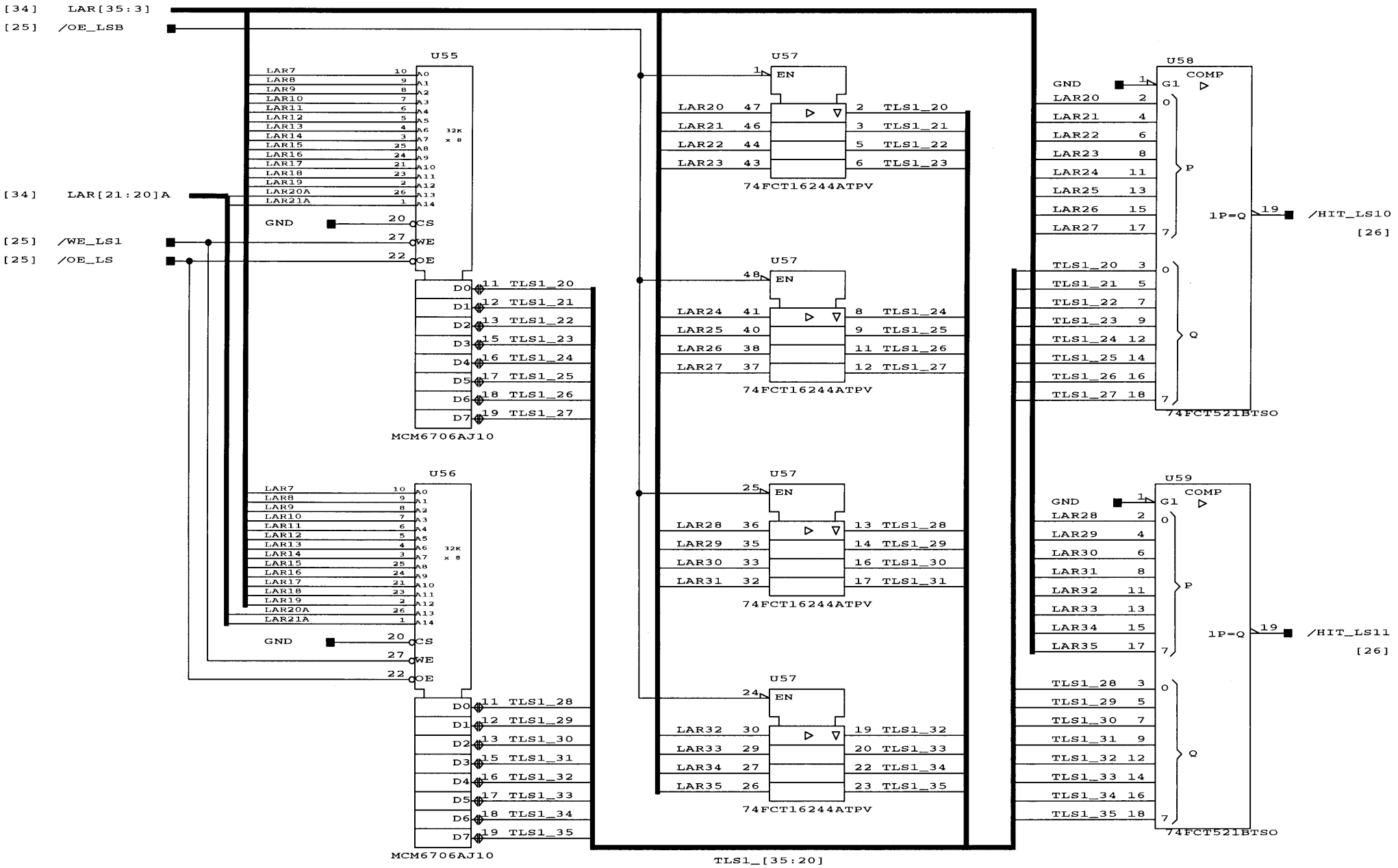
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global snooper 3	
Issue 2			
Issue 3		File: cpu305-1	Page:33 of 72



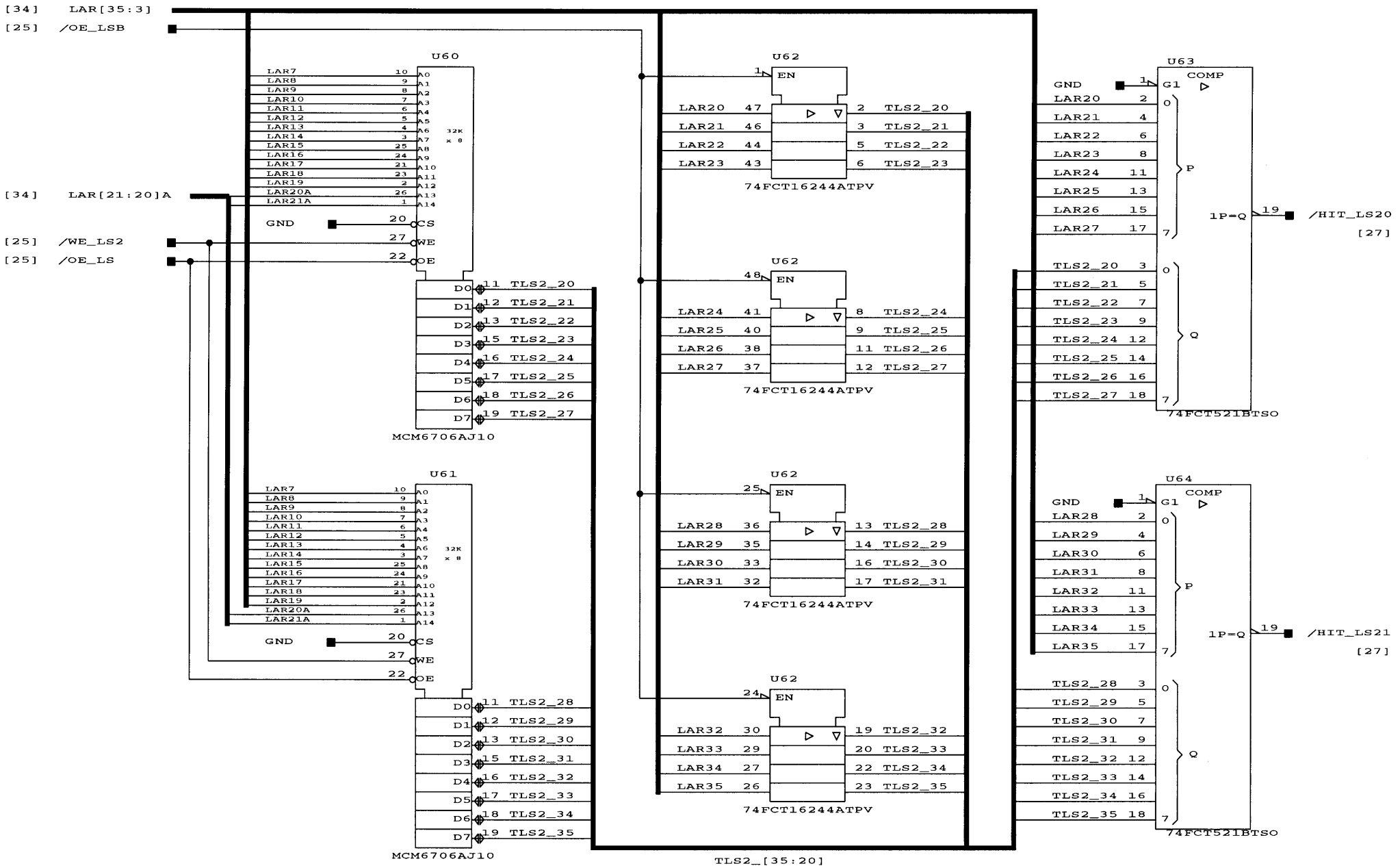
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local address register	
Issue 2			
Issue 3		File: cpu305-1	Page:34 of 72



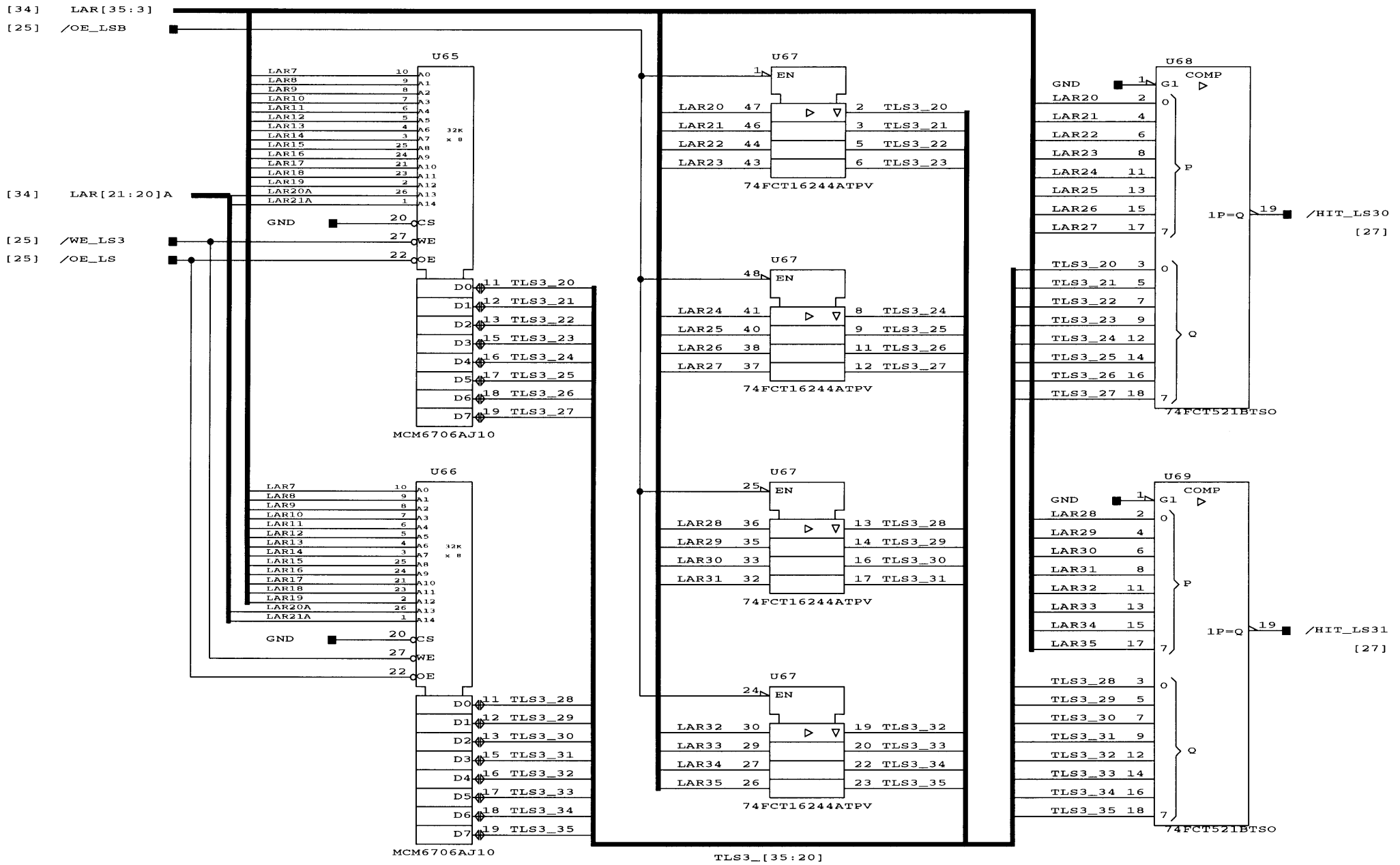
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local snooper 0	
Issue 2			
Issue 3		File: cpu305-1	Page:35 of 72



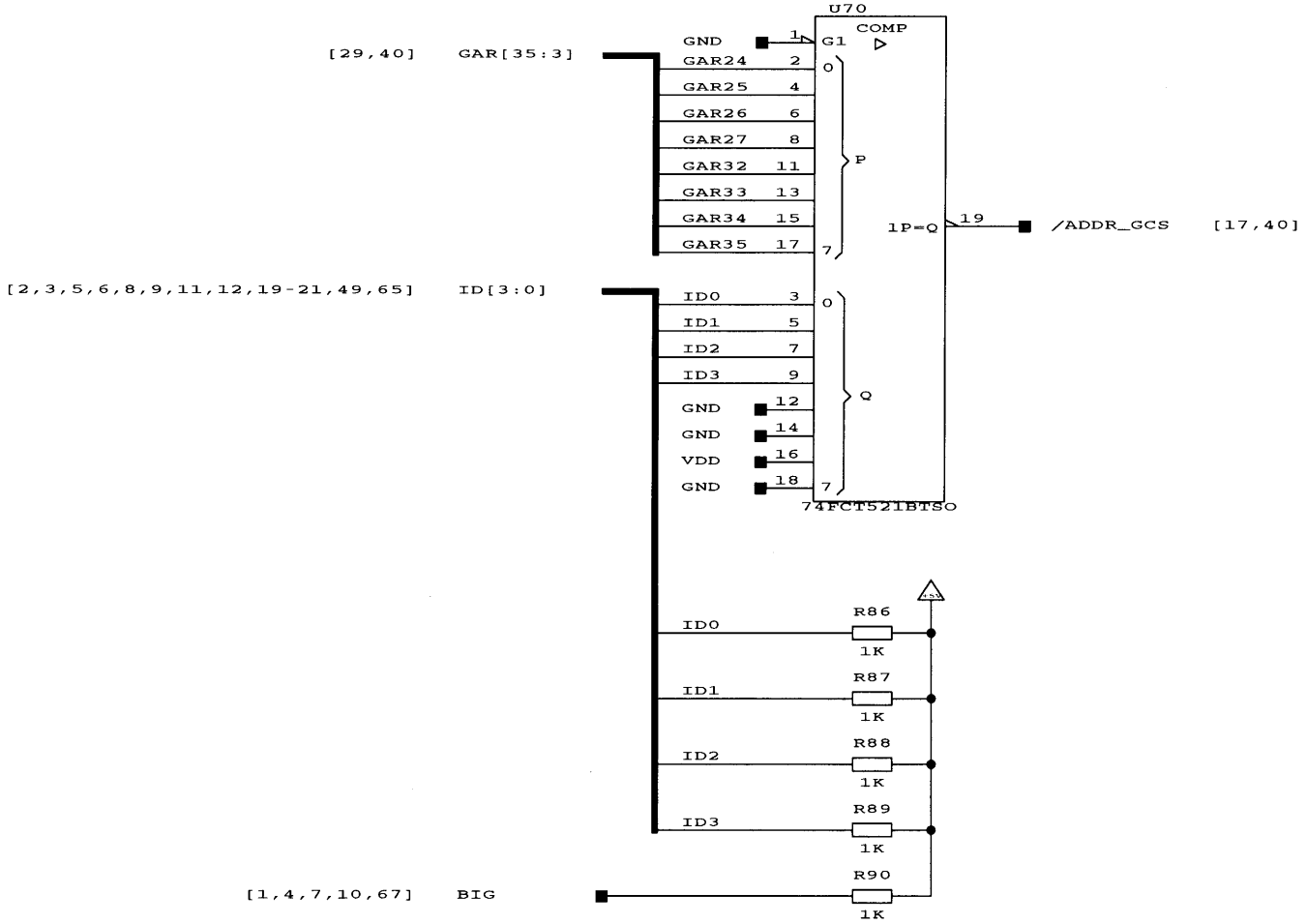
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local snooper 1	
Issue 2			
Issue 3		File: cpu305-1	Page:36 of 72



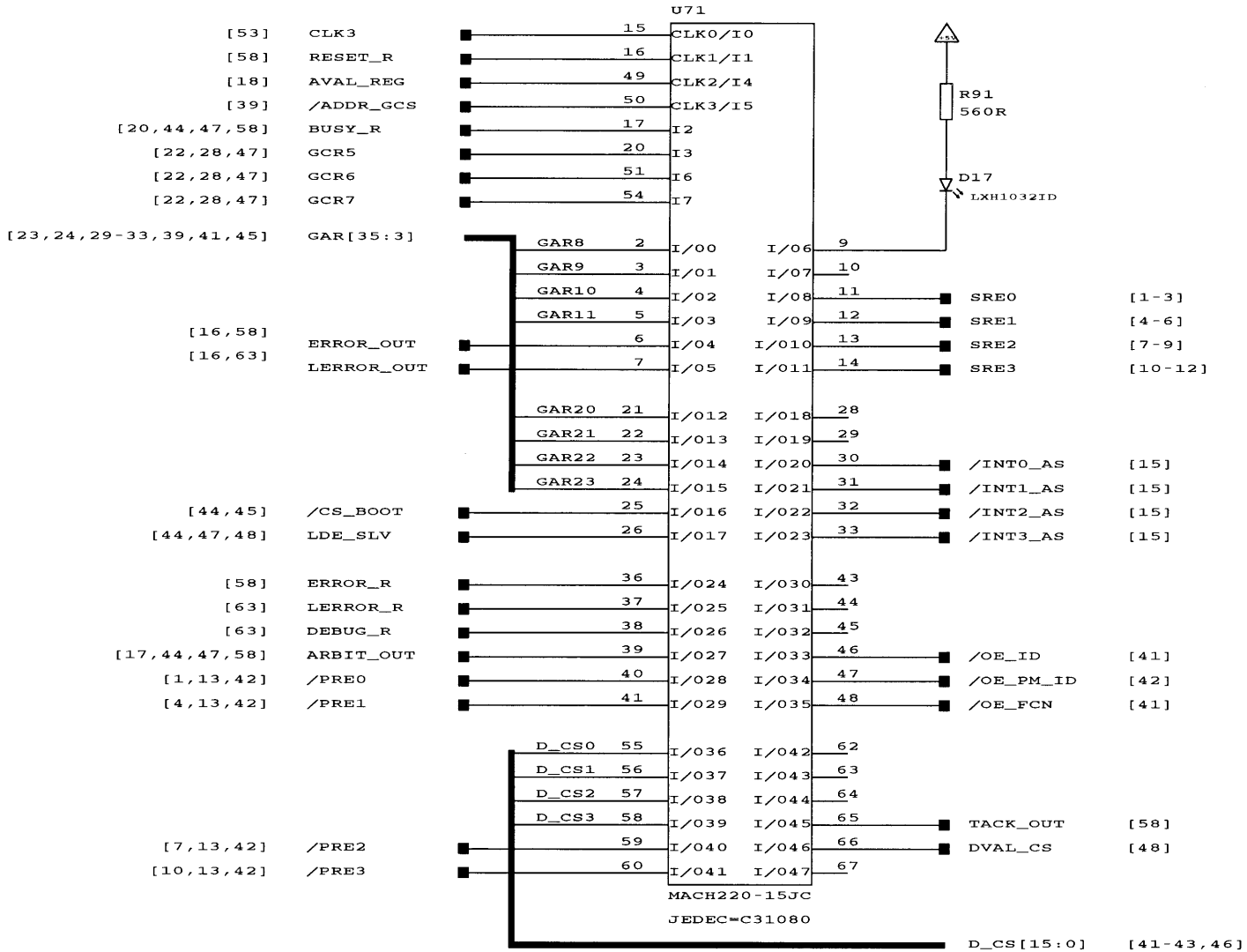
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local snooper 2	
Issue 2			
Issue 3		File: cpu305-1	Page:37 of 72



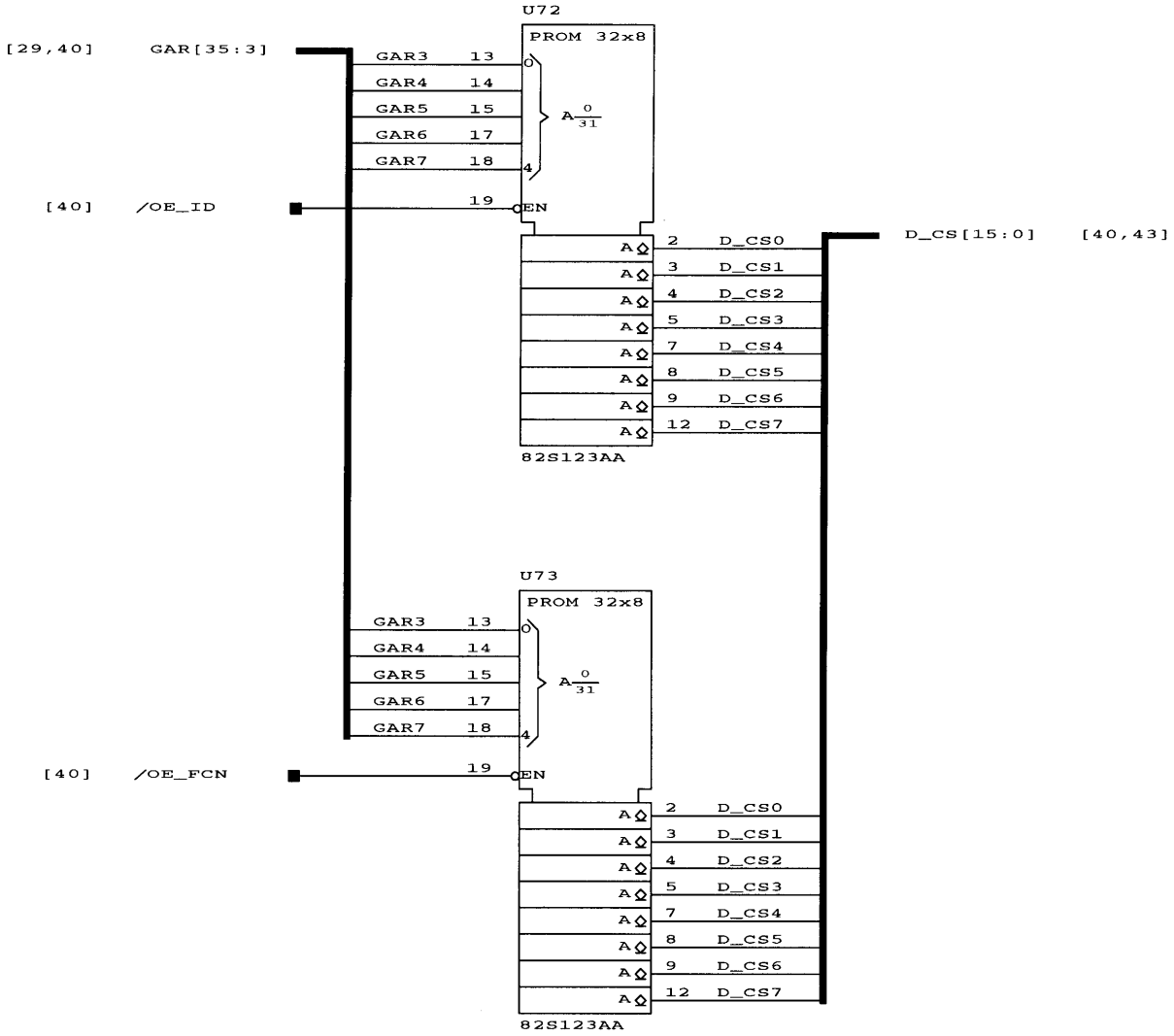
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local snooper 3	
Issue 2			
Issue 3		File: cpu305-1	Page:38 of 72



dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global control	
Issue 2		space decode	
Issue 3		File: cpu305-1	Page:39 of 72

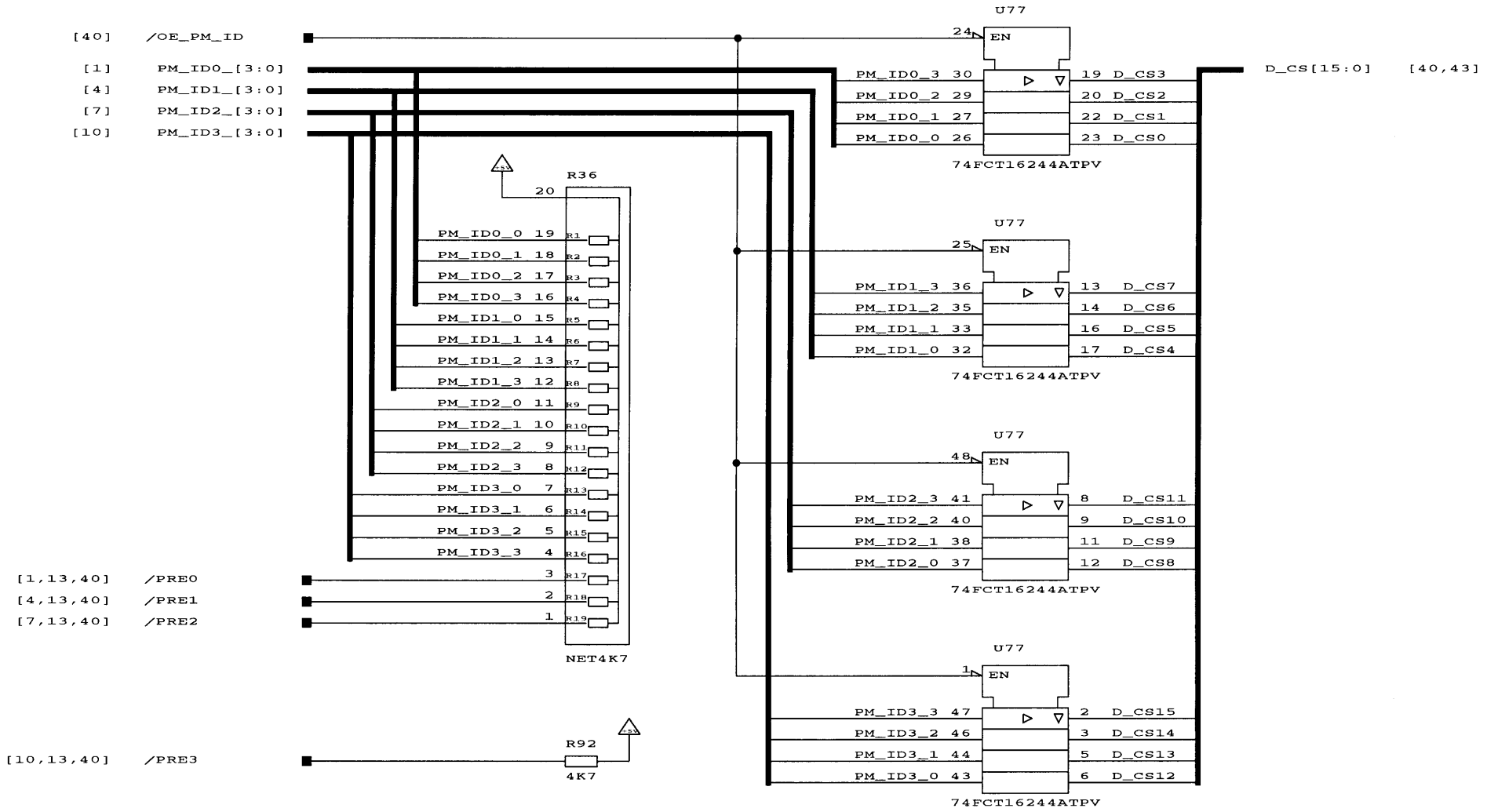


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global control space	
Issue 2		control	
Issue 3		File: cpu305-1	Page:40 of 72

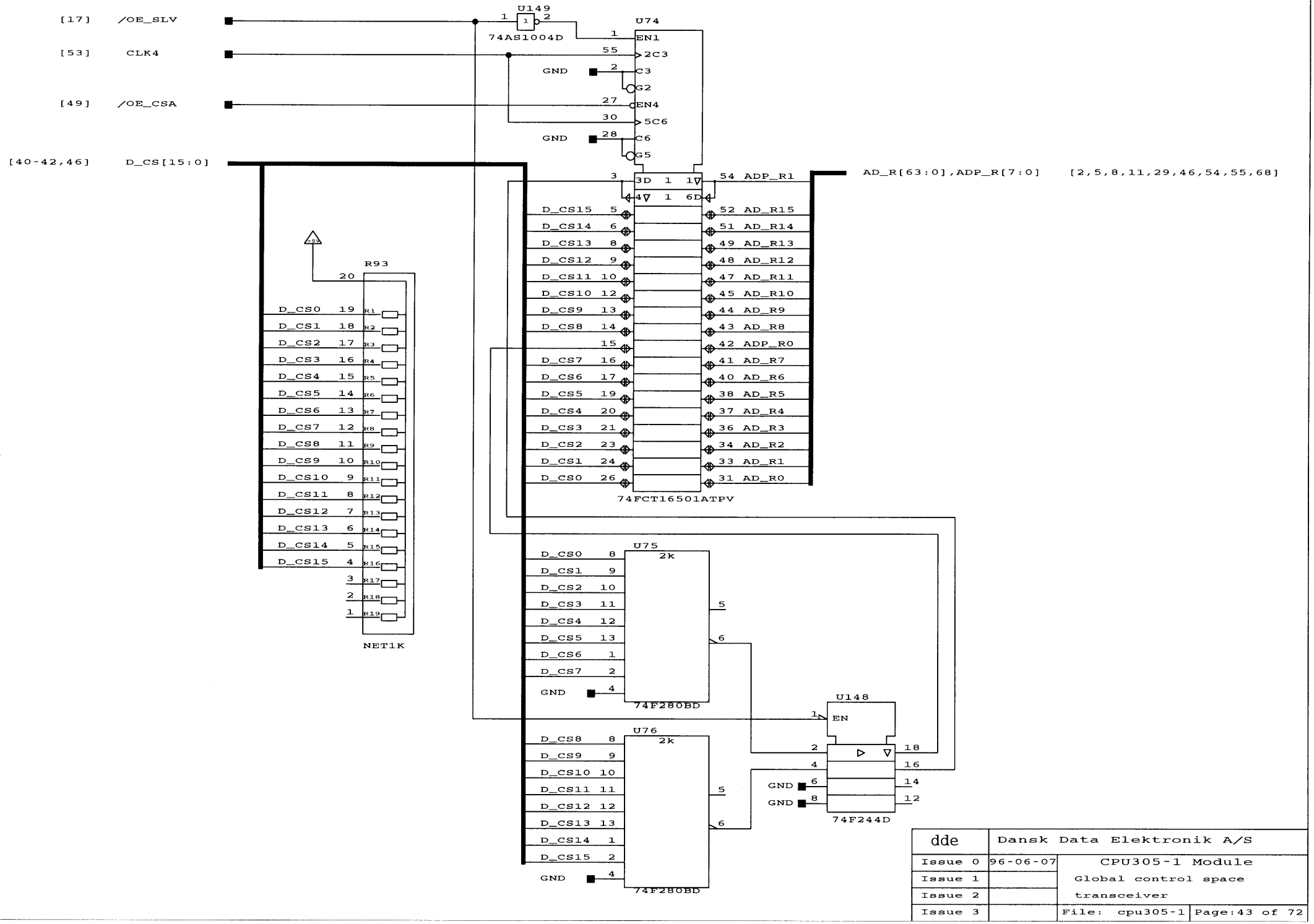


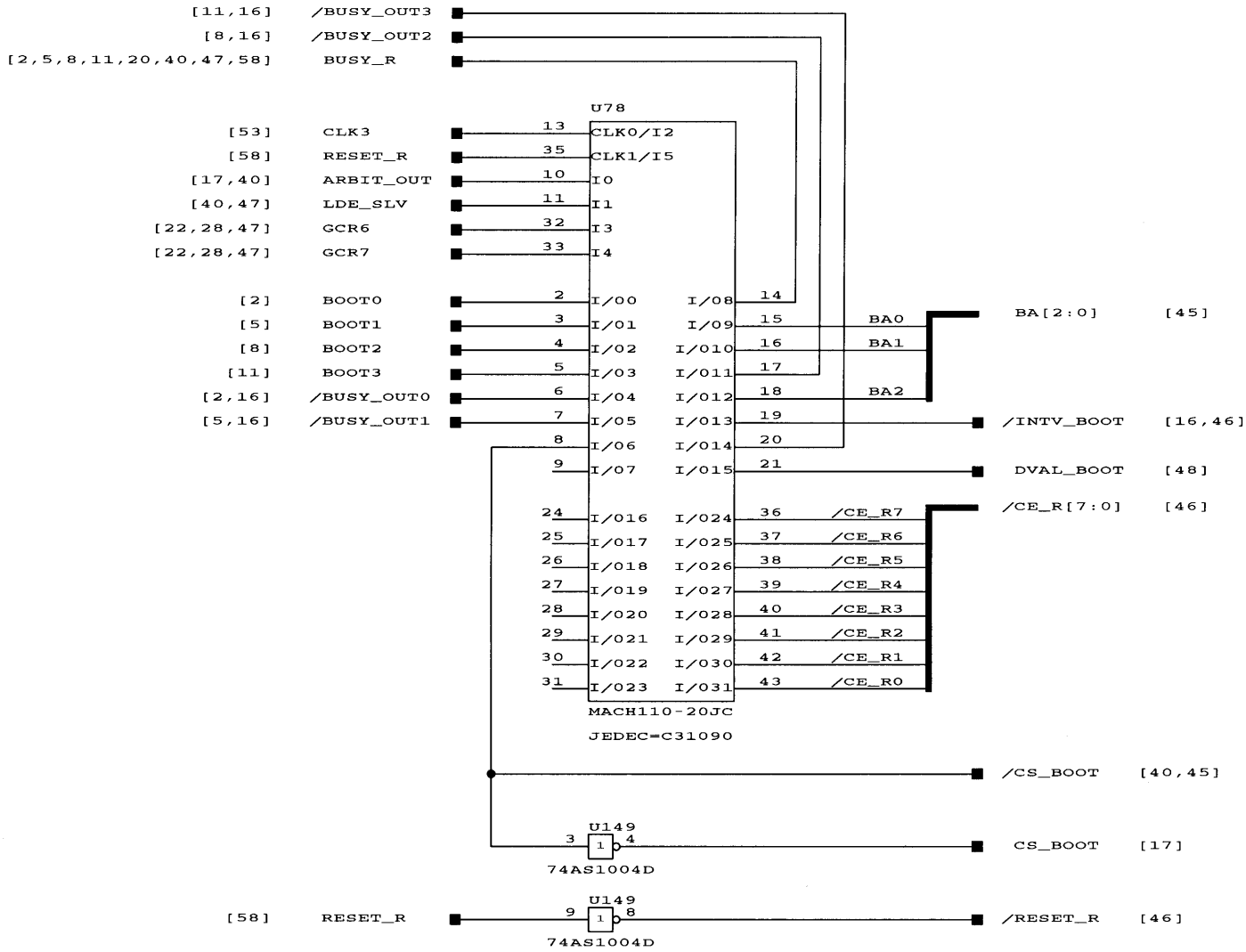
FCN PROM mounted in socket.

dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Module ID and FCN PROMs.	
Issue 2			
Issue 3		File: cpu305-1	Page:41 of 72

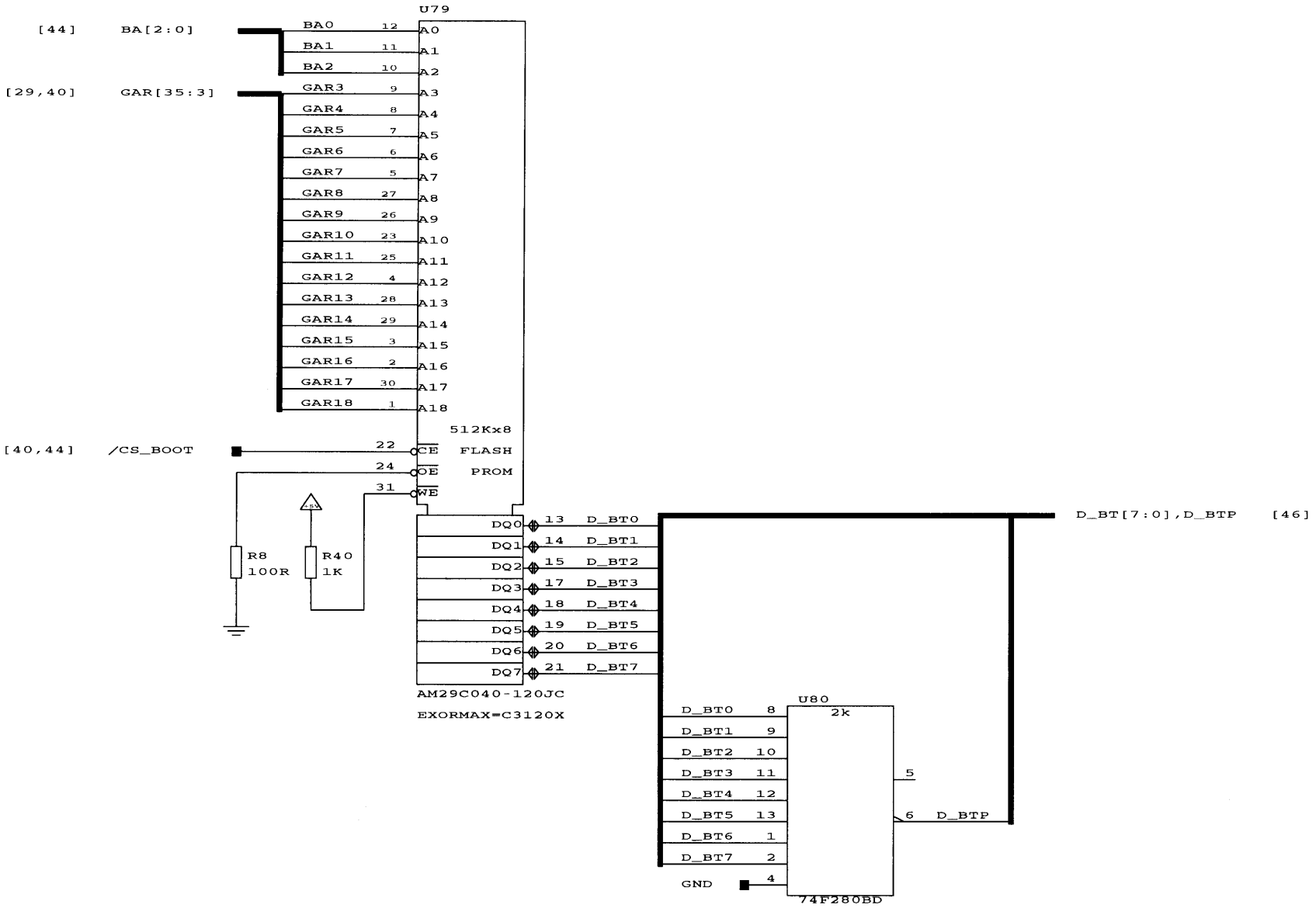


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Processor module	
Issue 2		ID register	
Issue 3		File: cpu305-1	Page:42 of 72

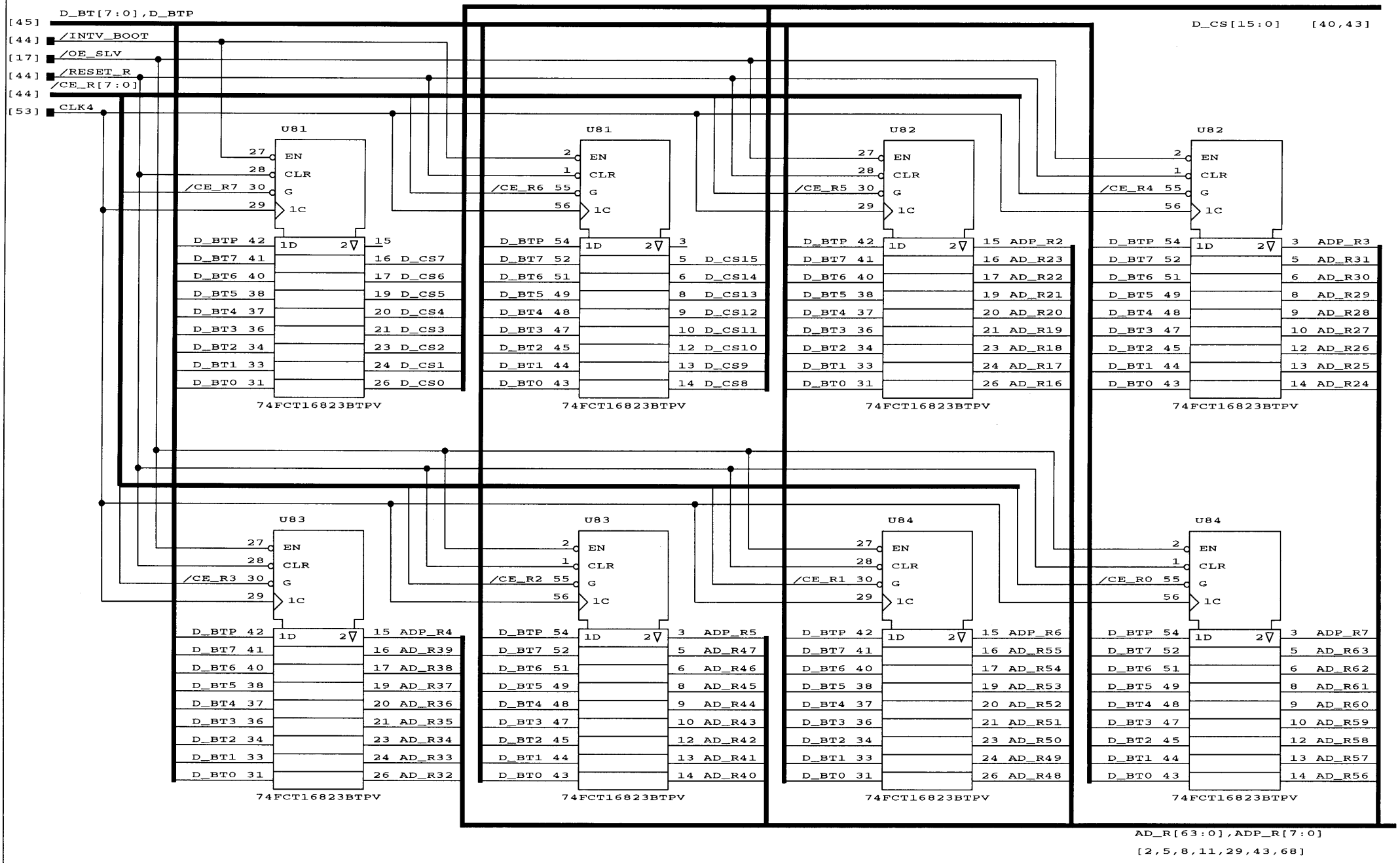

[17] /OE_SLV
[53] CLK4
[49] /OE_CSA
[40-42,46] D_CS[15:0]
U149
74AS1004D
U74
EN1
2C3
C3
G2
EN4
5C6
C6
G5
GND
AD_R[63:0],ADP_R[7:0] [2,5,8,11,29,46,54,55,68]
74FCT16501ATPV
R93
NET1K
U75
U76
74F280BD
U148
74F244D
dde
Dansk Data Elektronik A/S
Issue 0 96-06-07
Issue 1
Issue 2
Issue 3
CPU305-1 Module
Global control space
transceiver
File: cpu305-1
Page:43 of 72



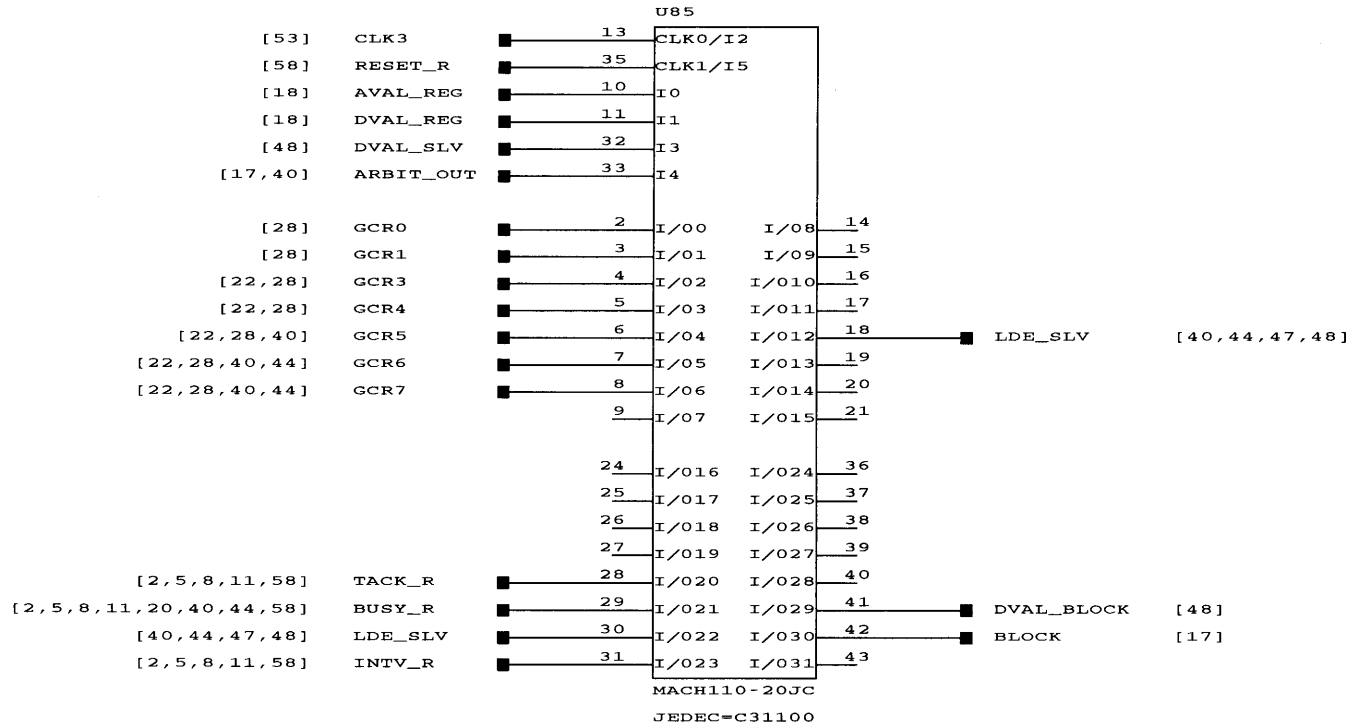
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Boot control	
Issue 2			
Issue 3		File: cpu305-1	Page:44 of 72



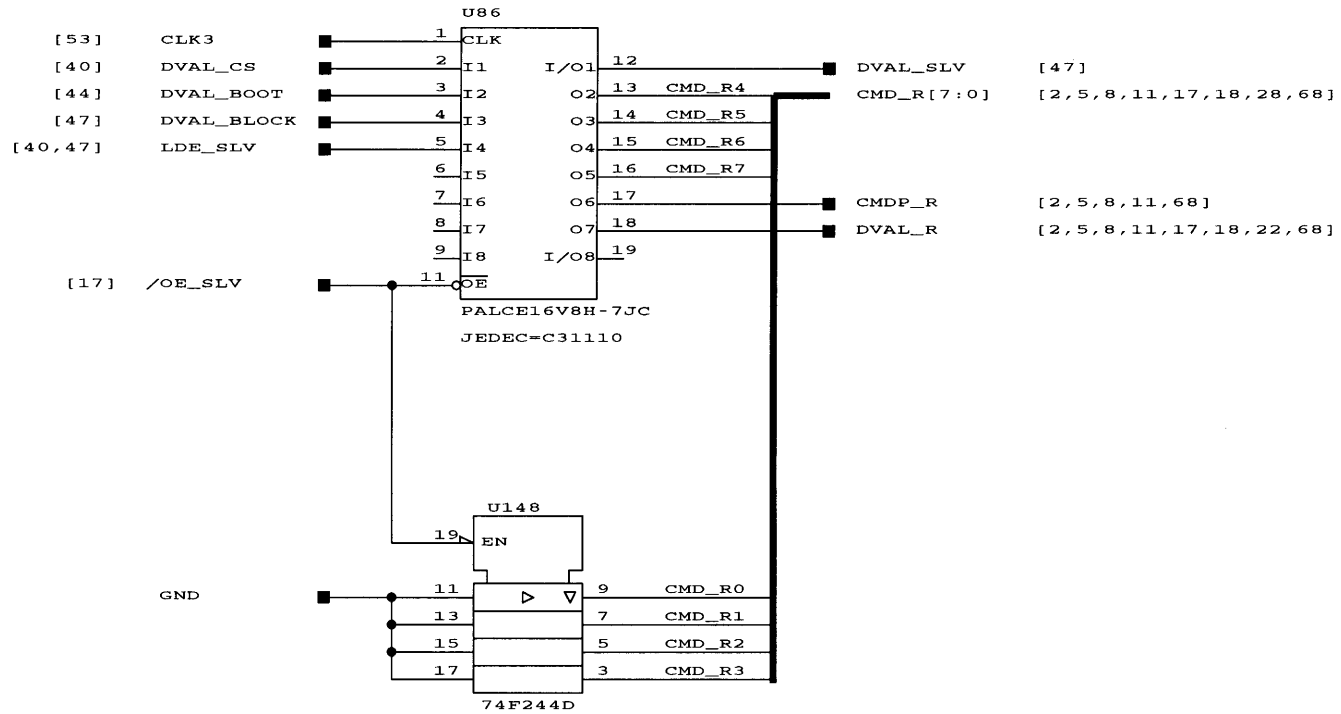
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Boot PROM	
Issue 2			
Issue 3		File: cpu305-1	Page:45 of 72



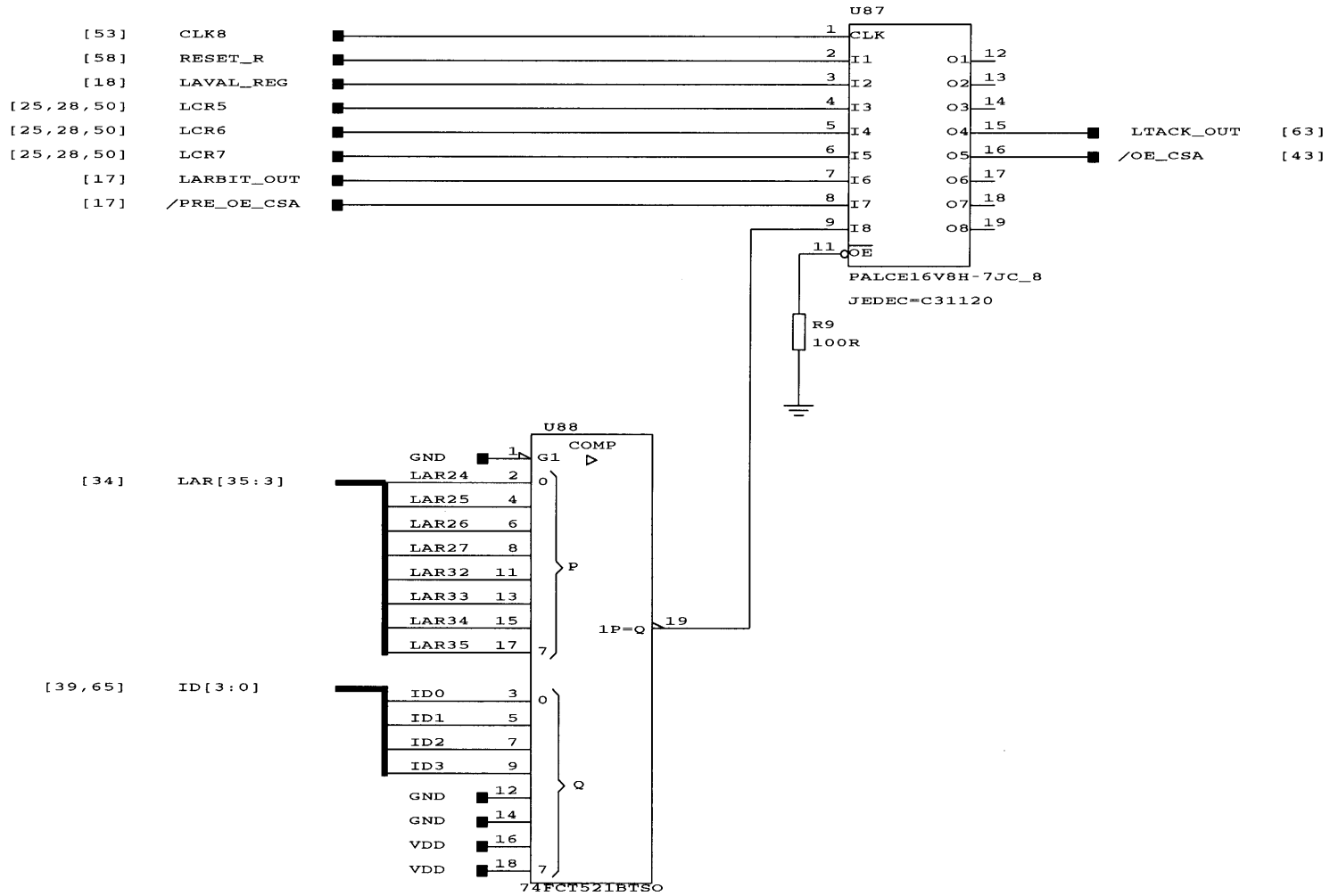
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Boot register	
Issue 2			
Issue 3		File: cpu305-1	Page:46 of 72



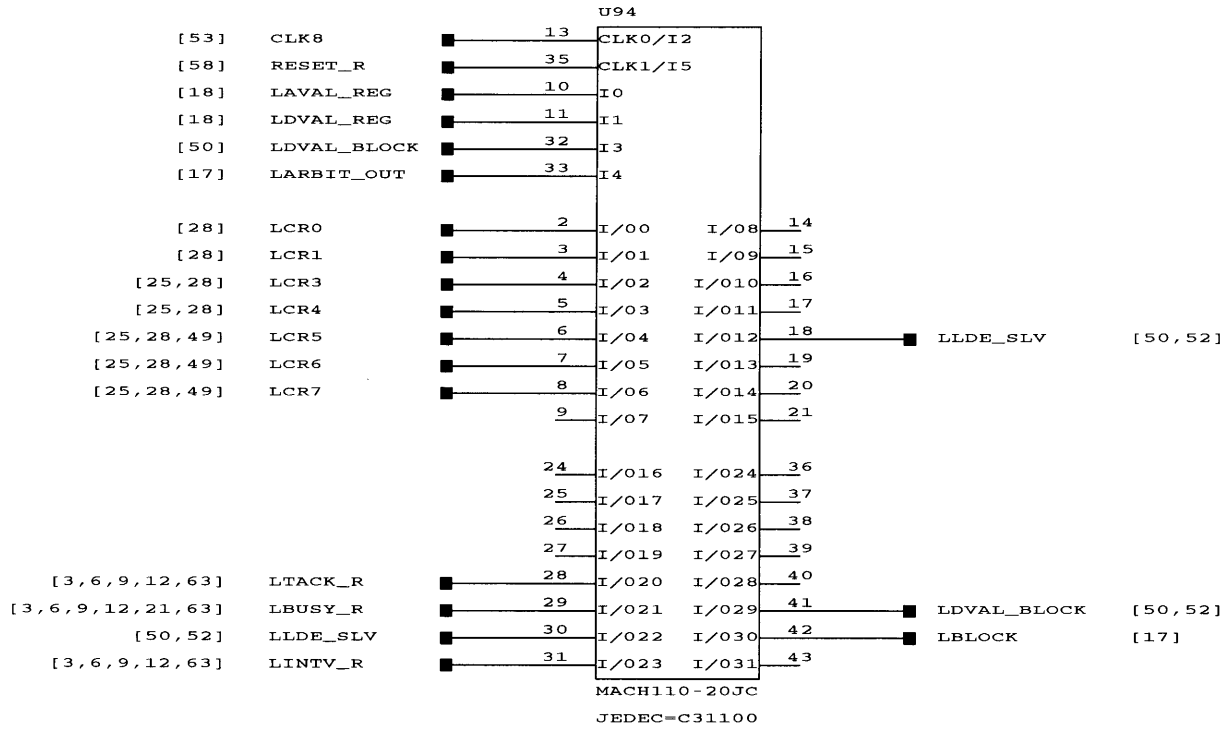
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global dummy	
Issue 2		block generator	
Issue 3		File: cpu305-1	Page:47 of 72



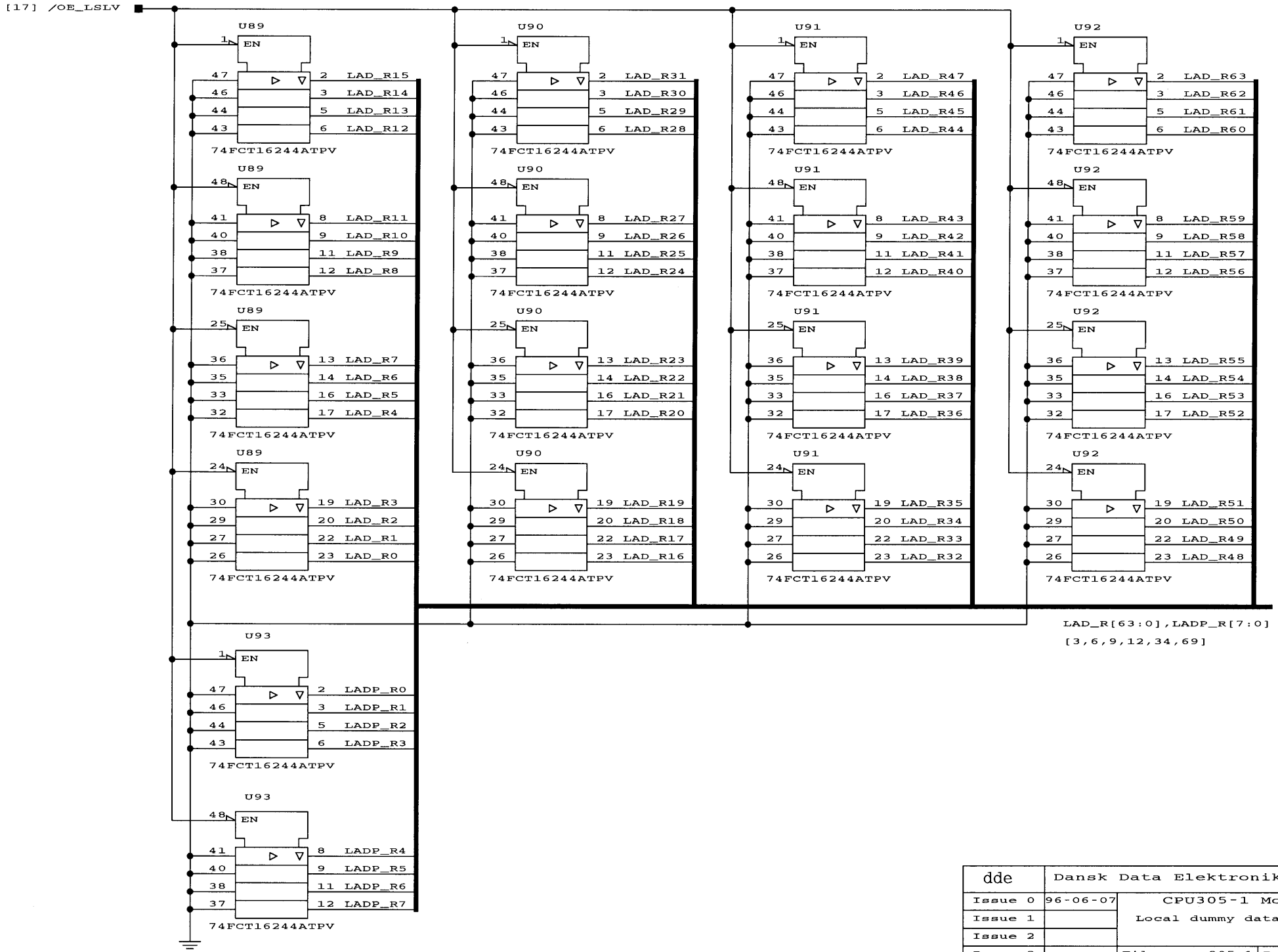
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global data identifier	
Issue 2		output register	
Issue 3		File: cpu305-1	Page:48 of 72



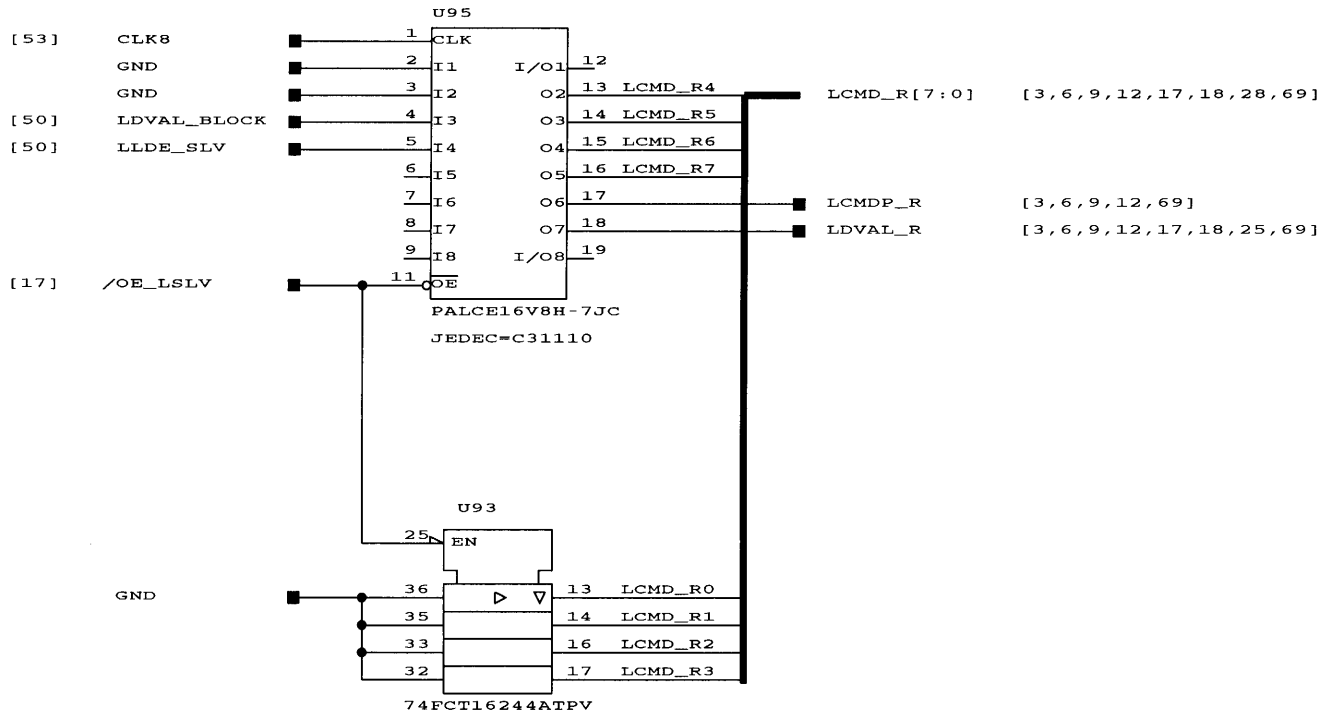
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local control space decode	
Issue 2		and target acknowledge	
Issue 3		File: cpu305-1	Page:49 of 72



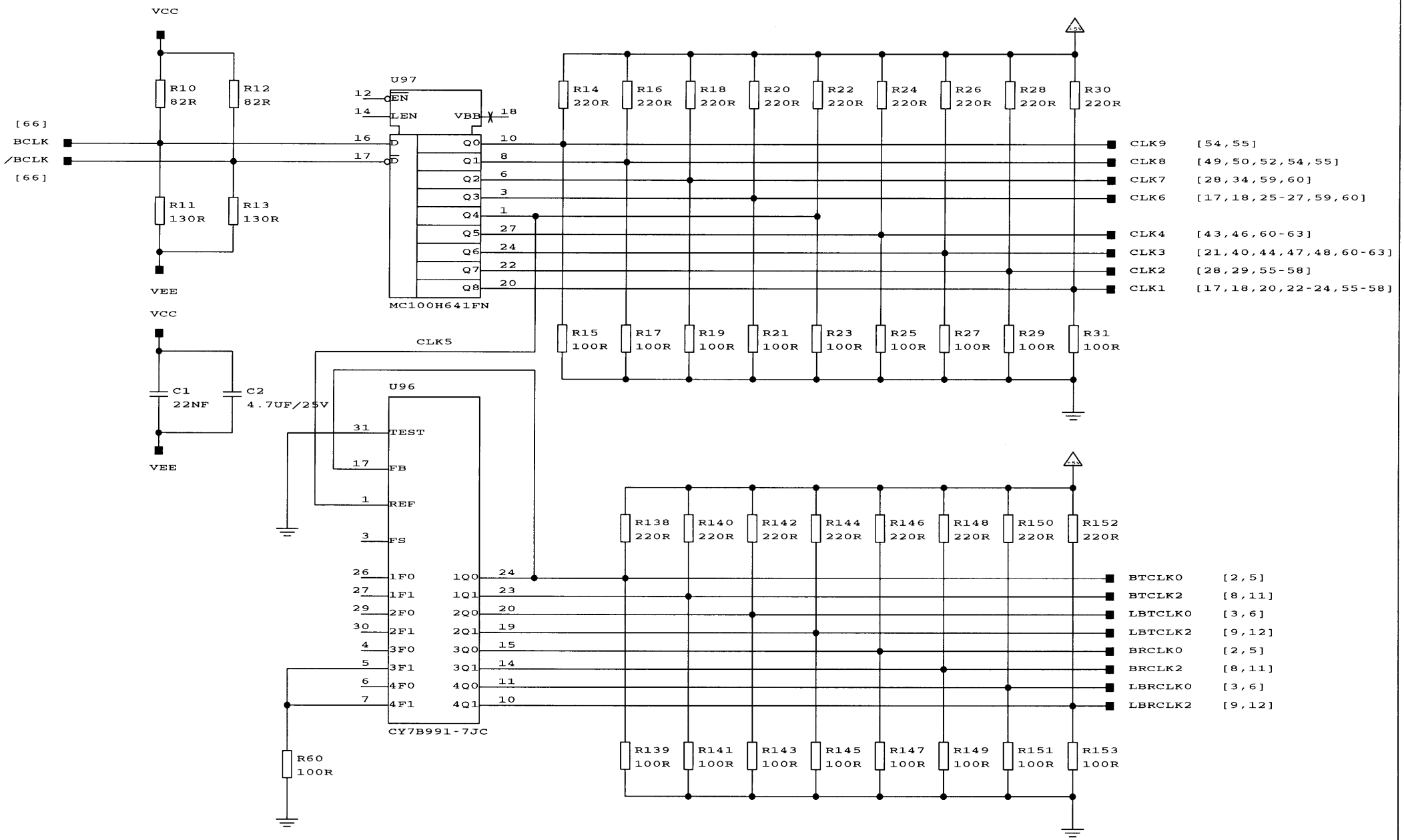
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local dummy block generator	
Issue 2			
Issue 3		File: cpu305-1	Page:50 of 72



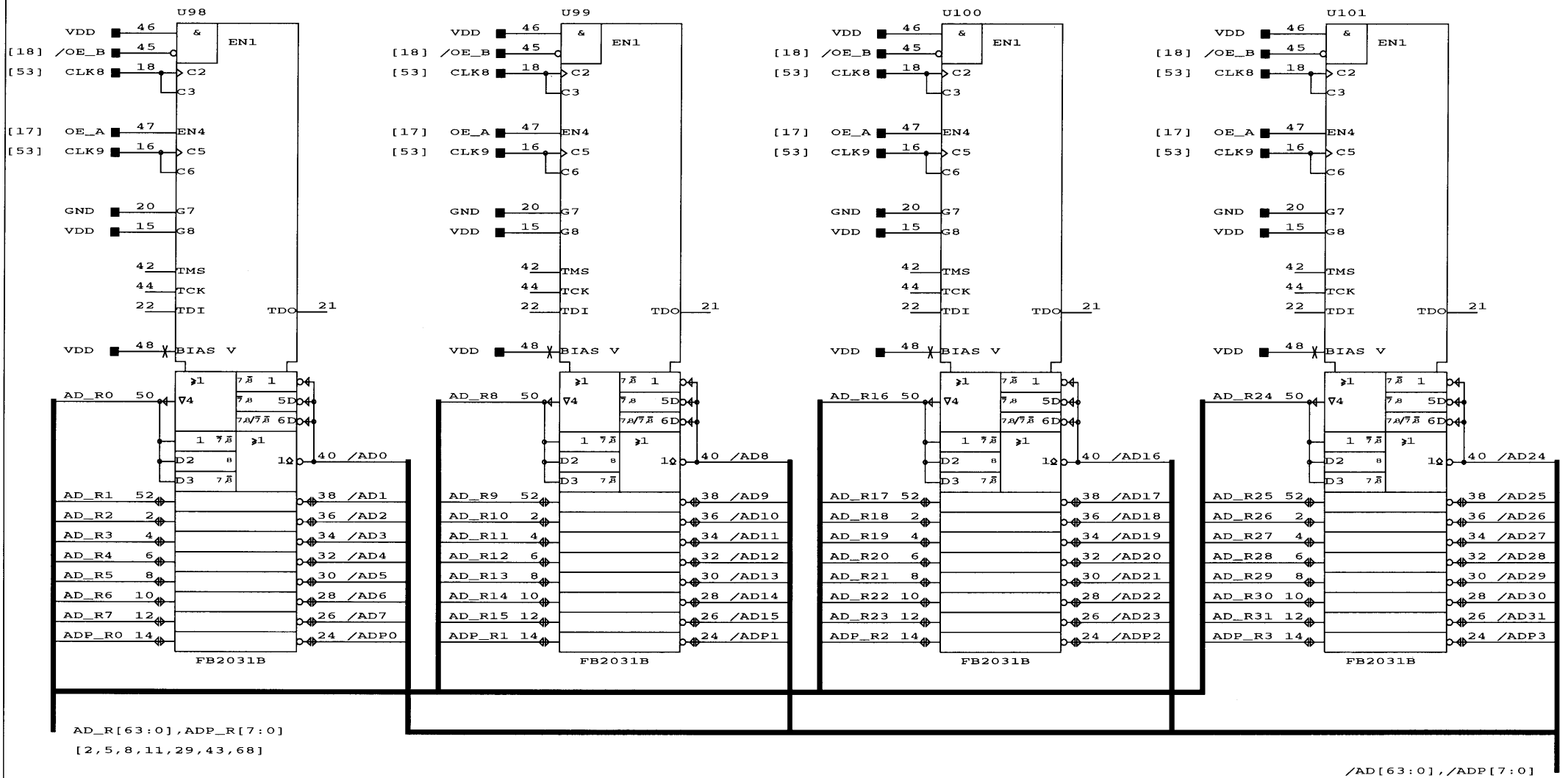
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local dummy data	
Issue 2			
Issue 3		File: cpu305-1	Page:51 of 72



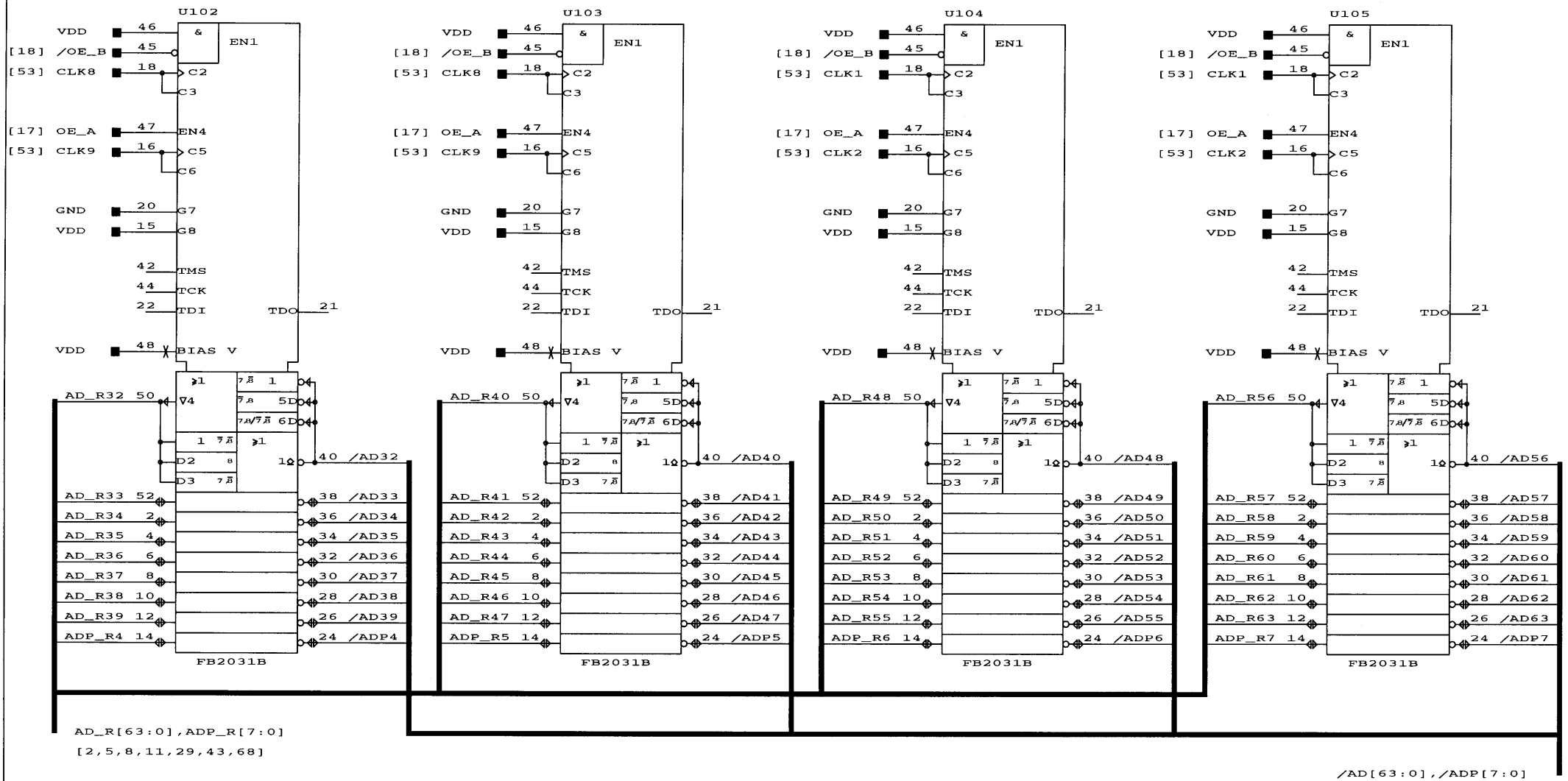
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local data identifier	
Issue 2		output register	
Issue 3		File: cpu305-1	Page:52 of 72



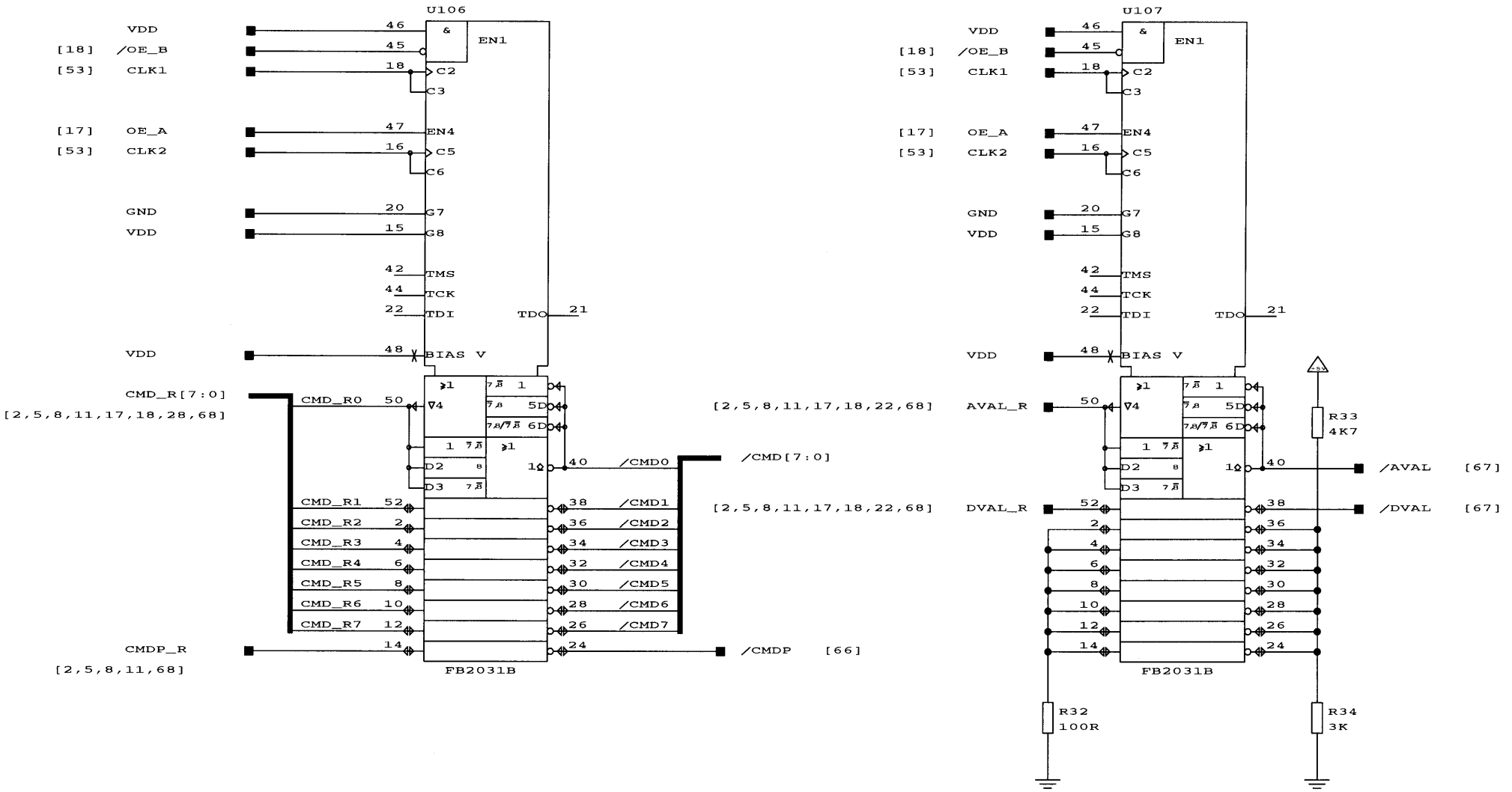
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Clock distribution	
Issue 2			
Issue 3		File: cpu305-1	Page:53 of 72



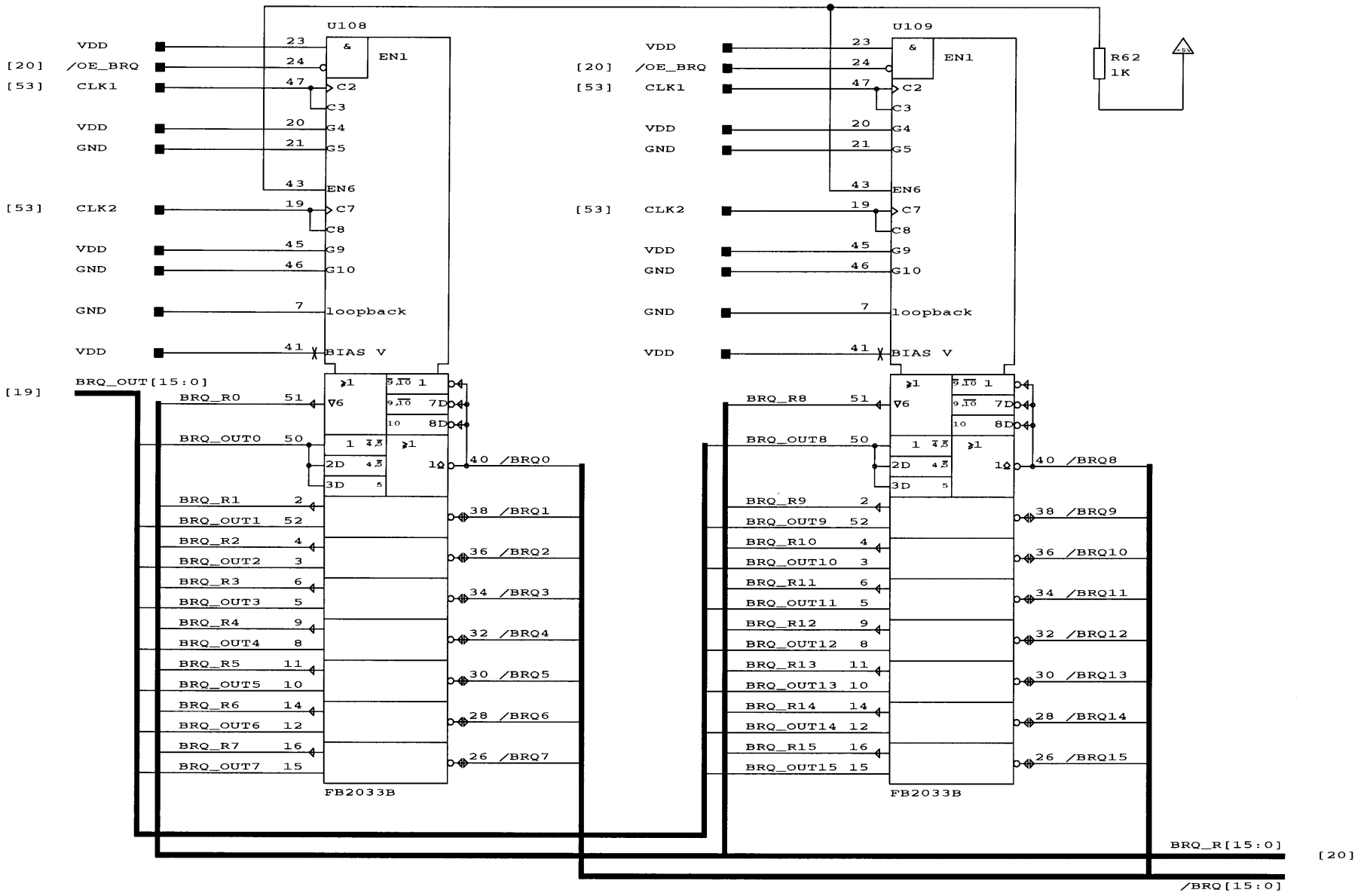
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global address/data	
Issue 2		transceiver	
Issue 3		File: cpu305-1	Page:54 of 72



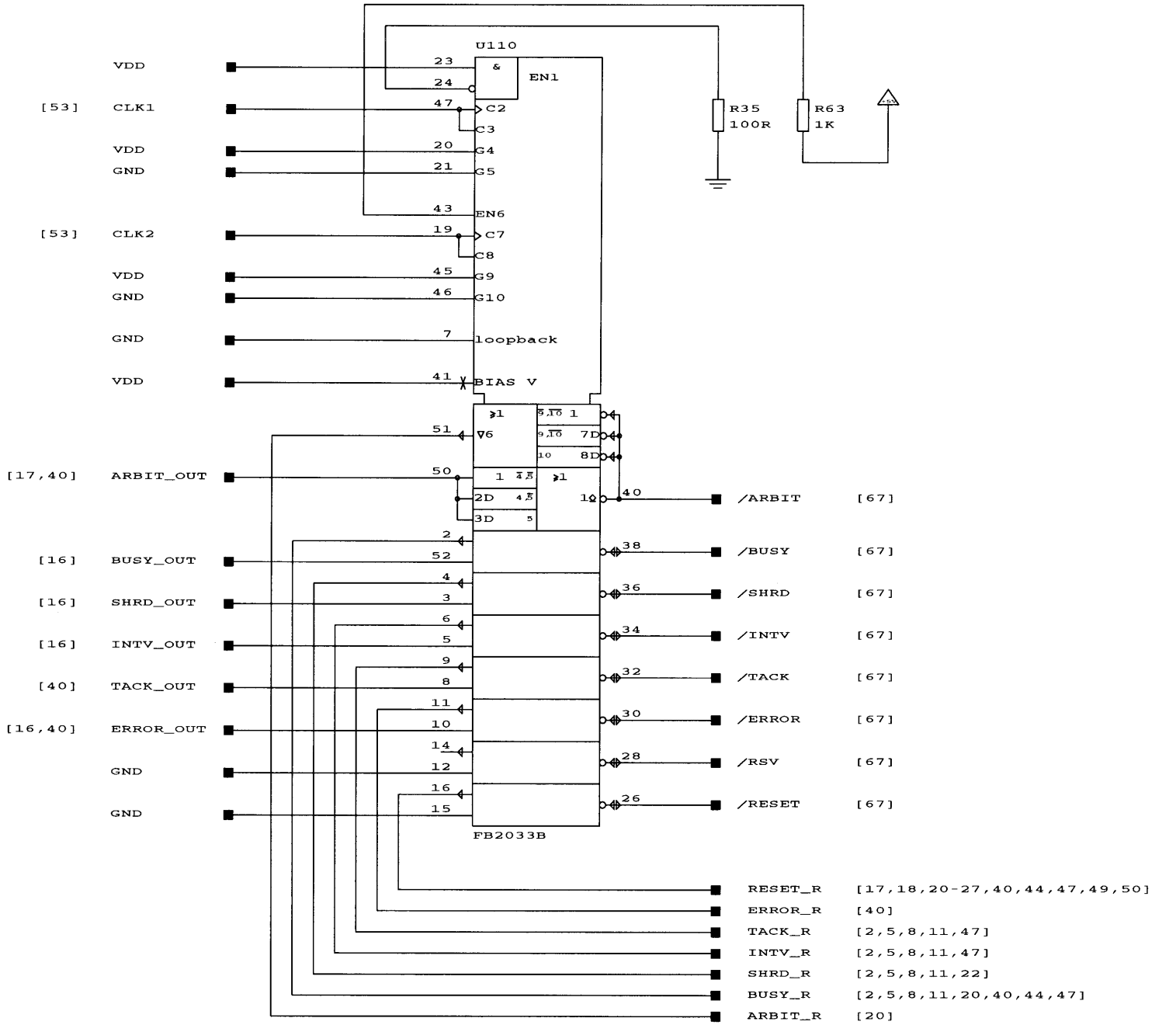
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global address/data	
Issue 2		transceiver	
Issue 3		File: cpu305-1	Page:55 of 72



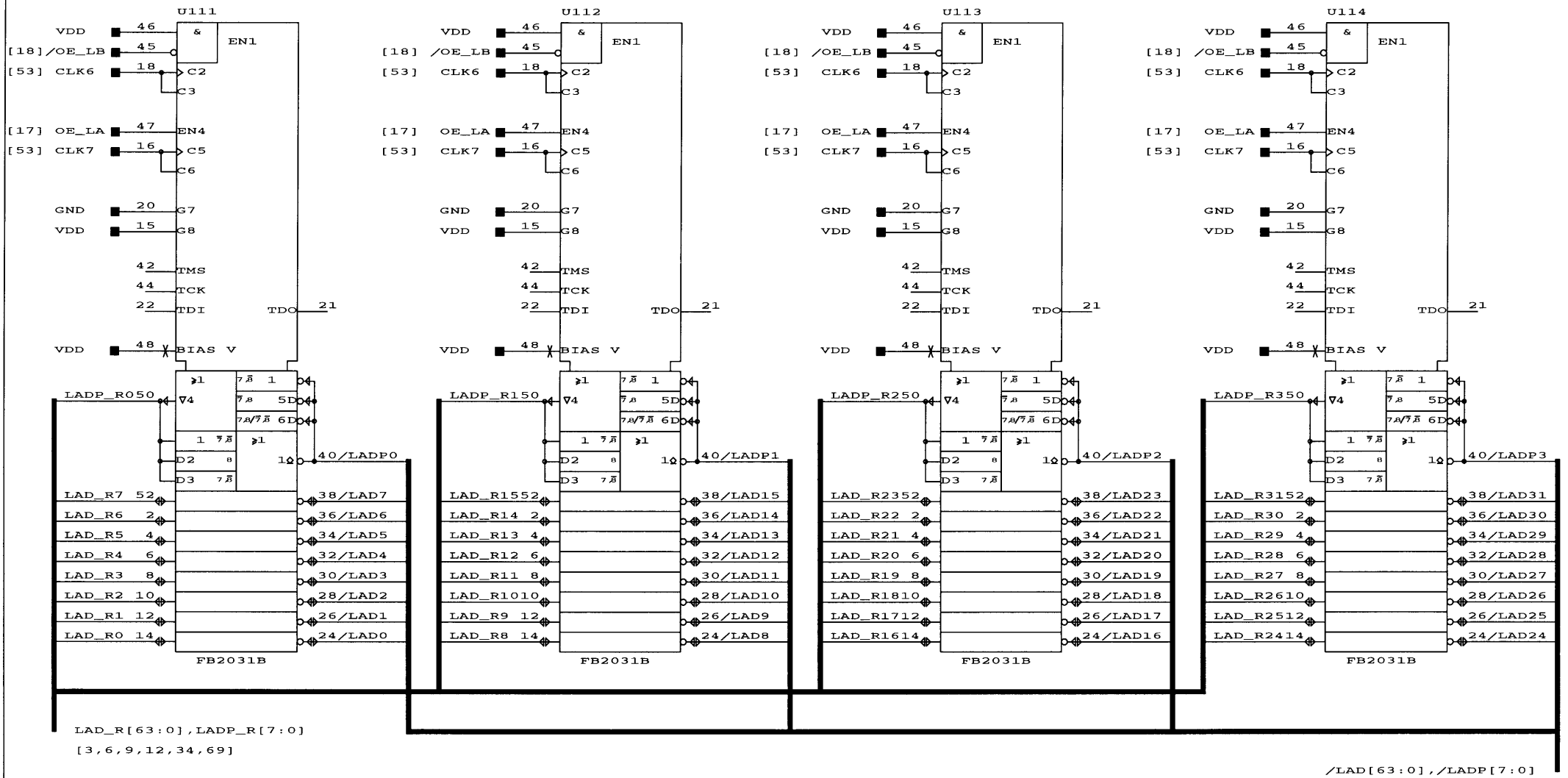
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global command and valid transceiver	
Issue 2			
Issue 3		File: cpu305-1	Page:56 of 72



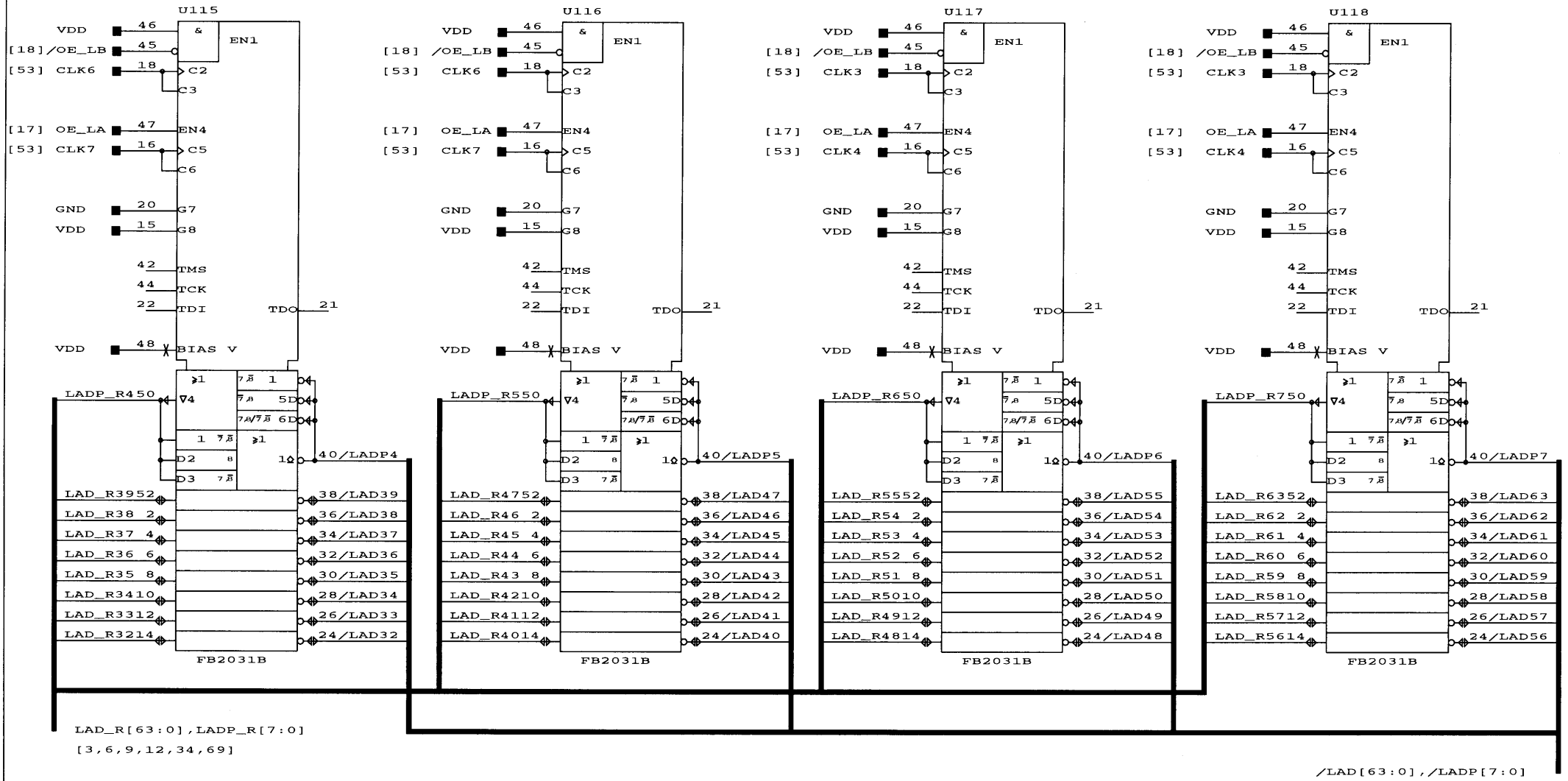
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global bus request	
Issue 2		transceiver	
Issue 3		File: cpu305-1	Page:57 of 72



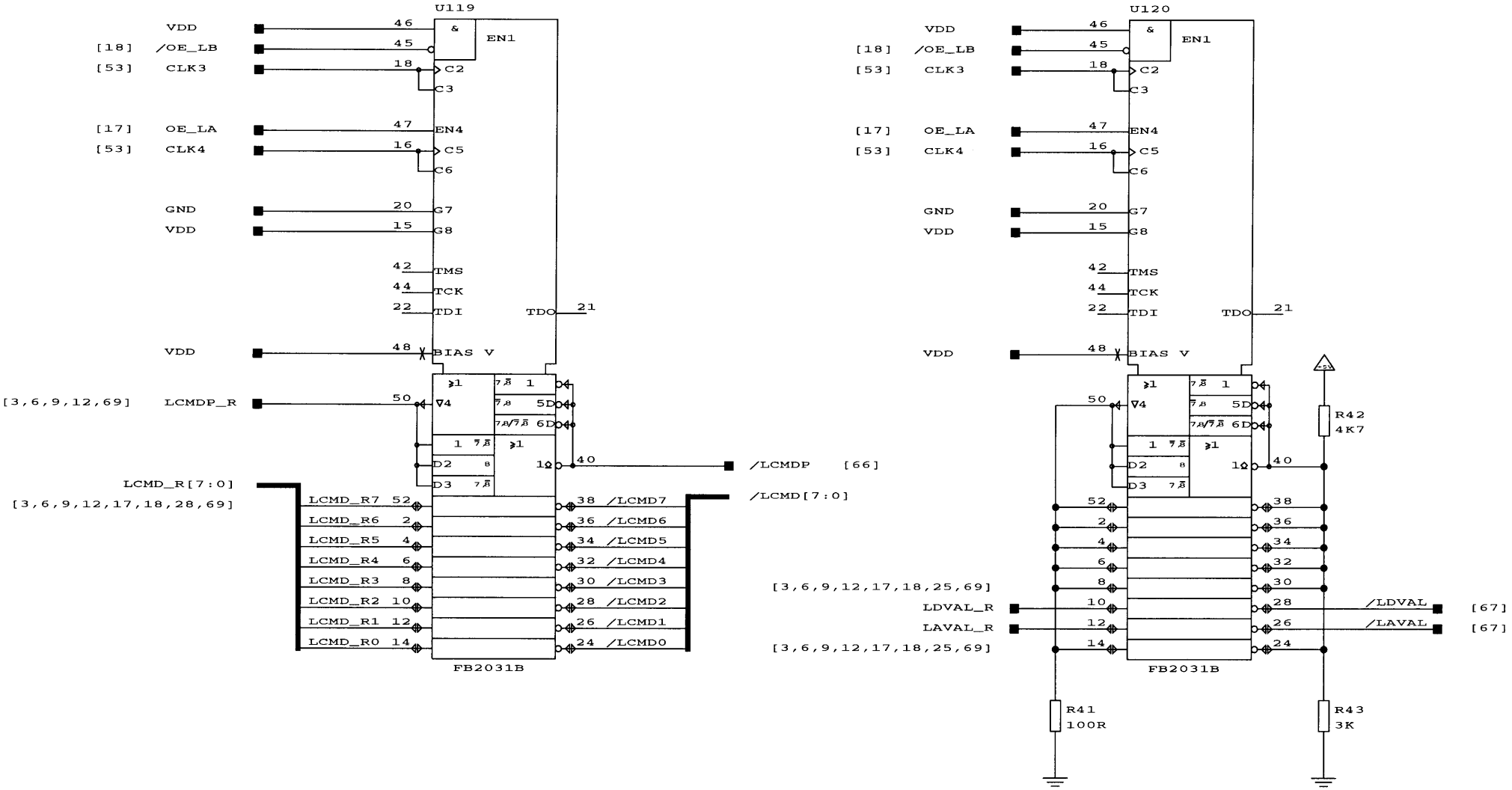
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Global control transceiver	
Issue 2			
Issue 3		File: cpu305-1	Page:58 of 72



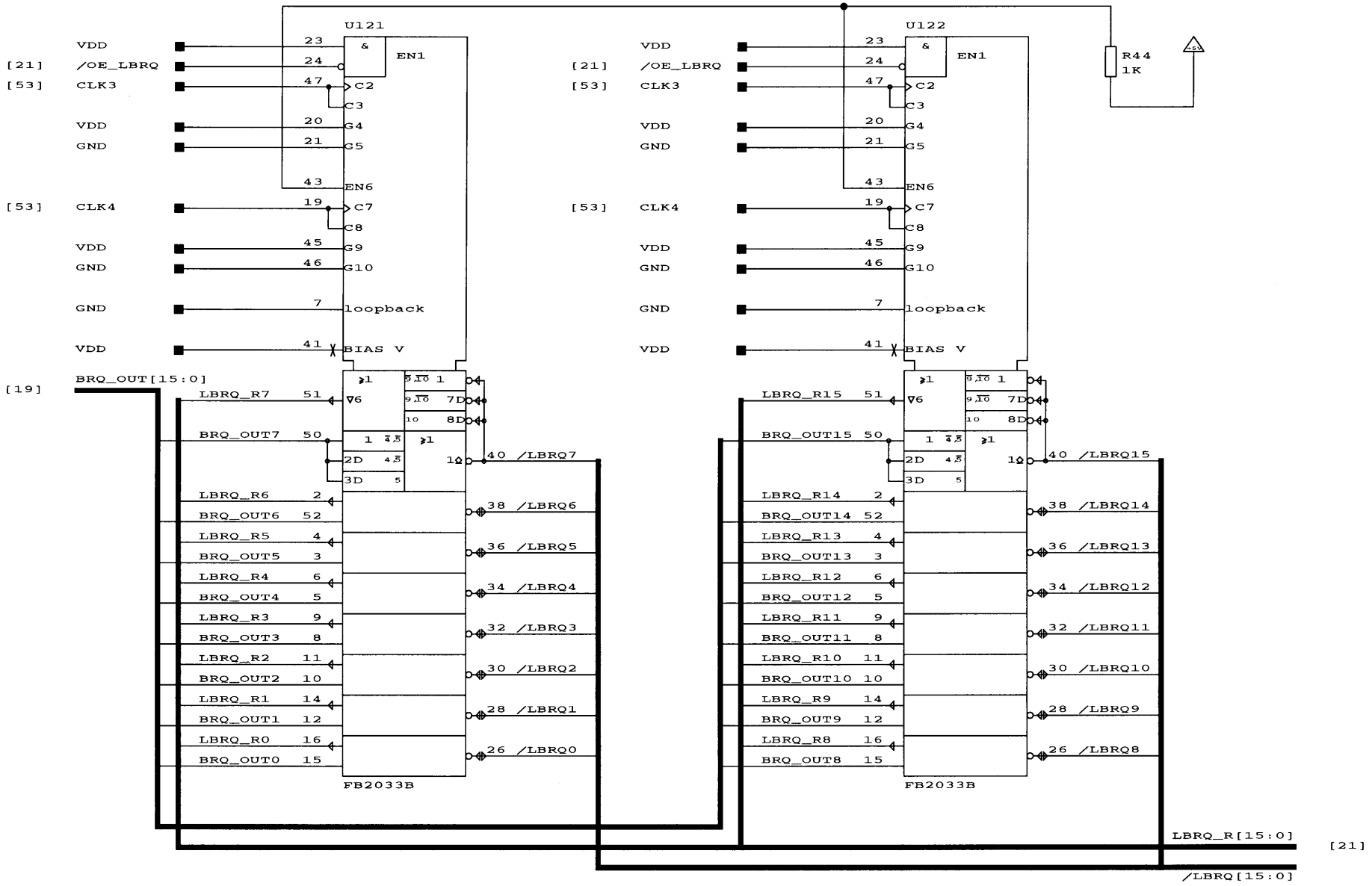
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local address/data	
Issue 2		transceiver	
Issue 3		File: cpu305-1	Page:59 of 72



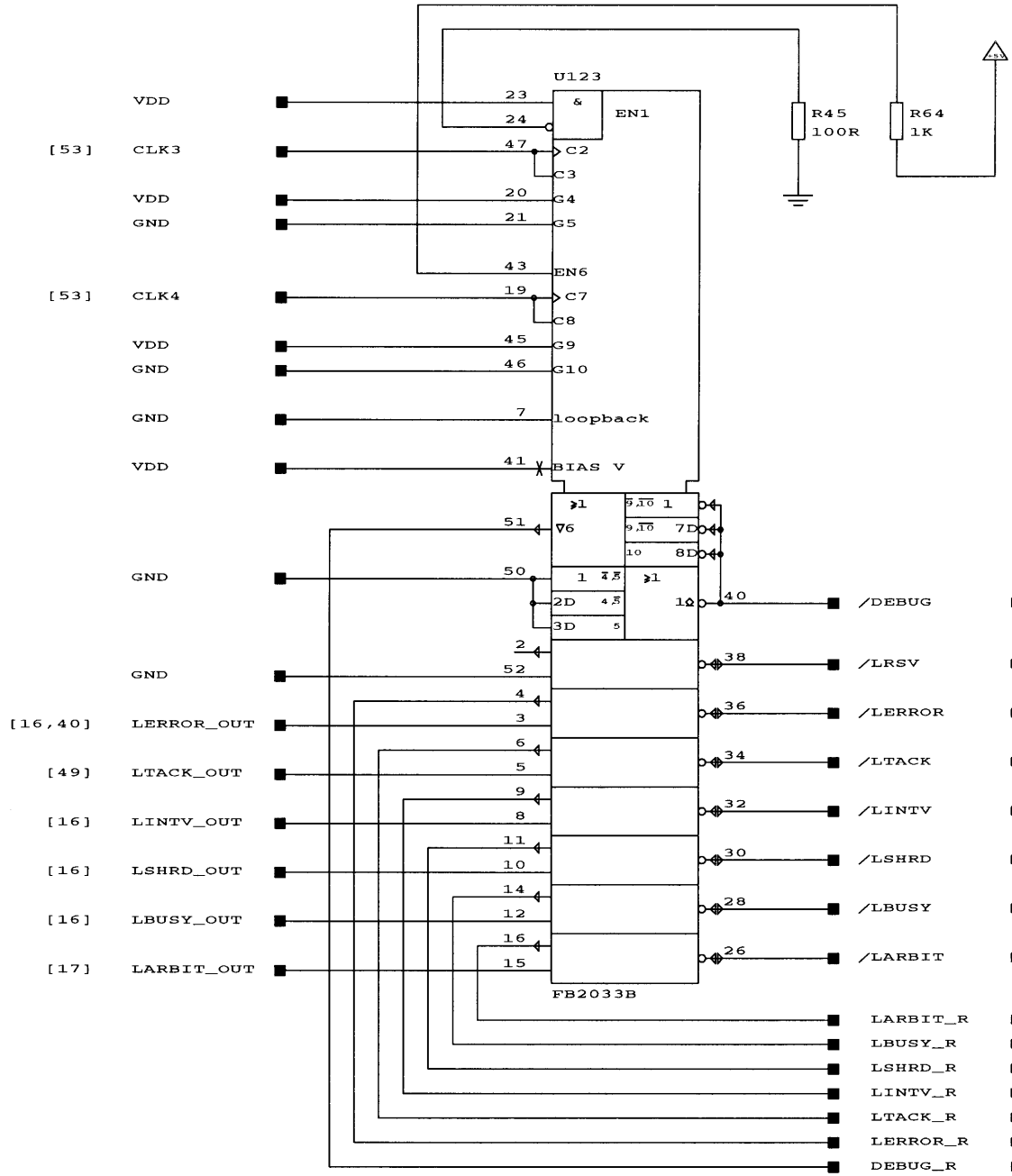
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local address/data	
Issue 2		transceiver	
Issue 3		File: cpu305-1	Page:60 of 72



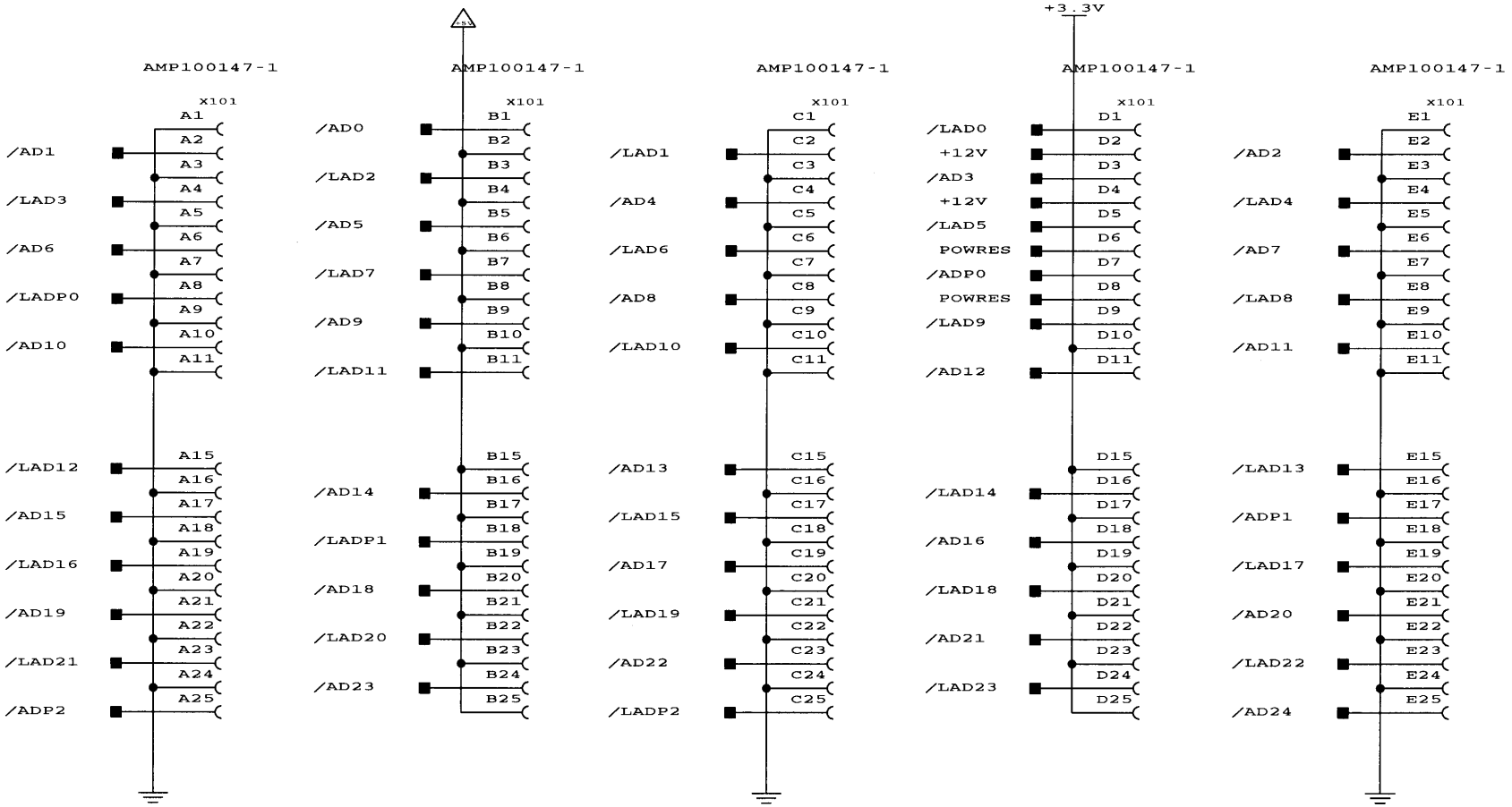
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local command and valid transceiver	
Issue 2			
Issue 3		File: cpu305-1	Page:61 of 72



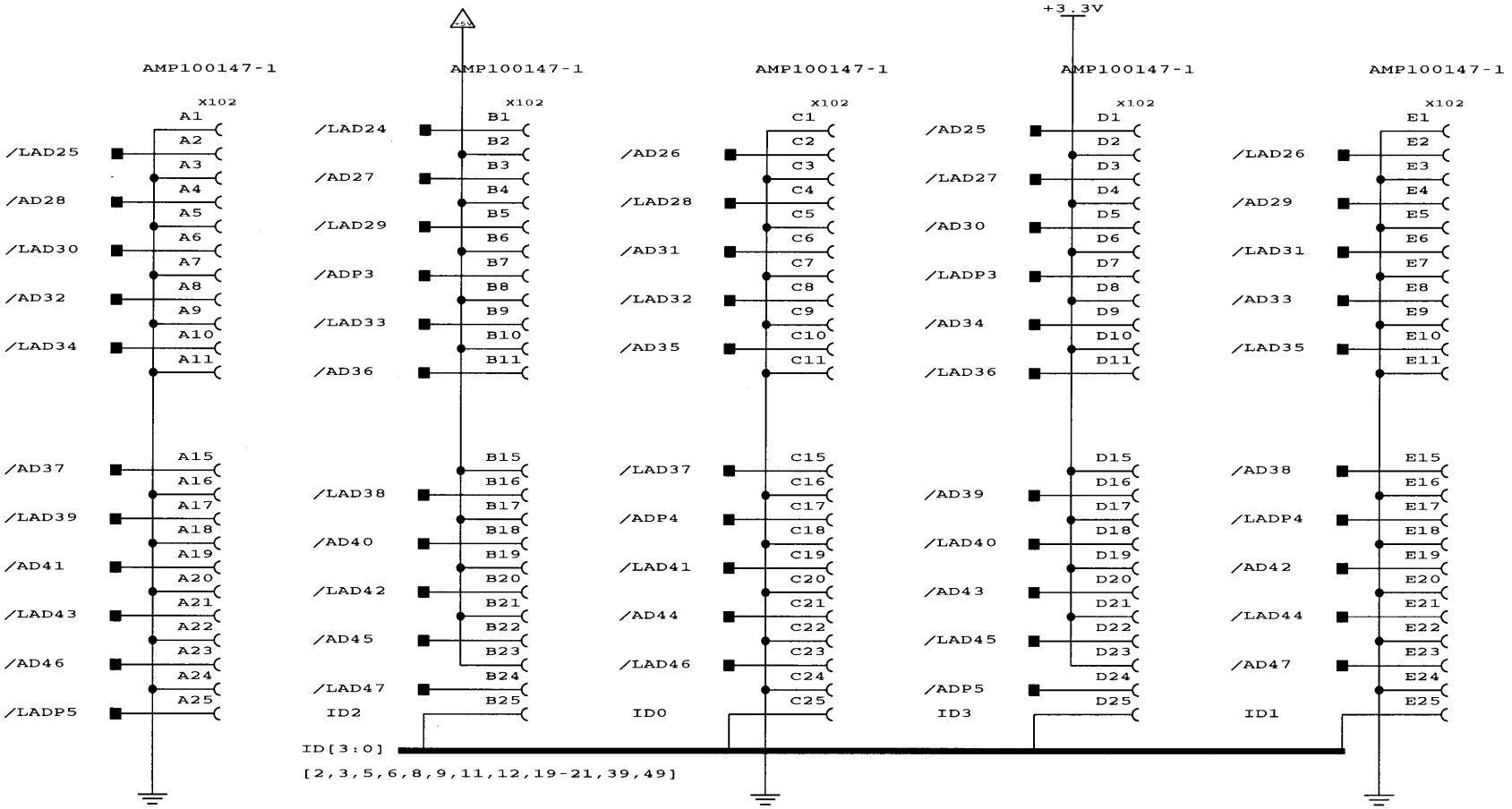
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local bus request	
Issue 2		transceiver	
Issue 3		File: cpu305-1	Page:62 of 72



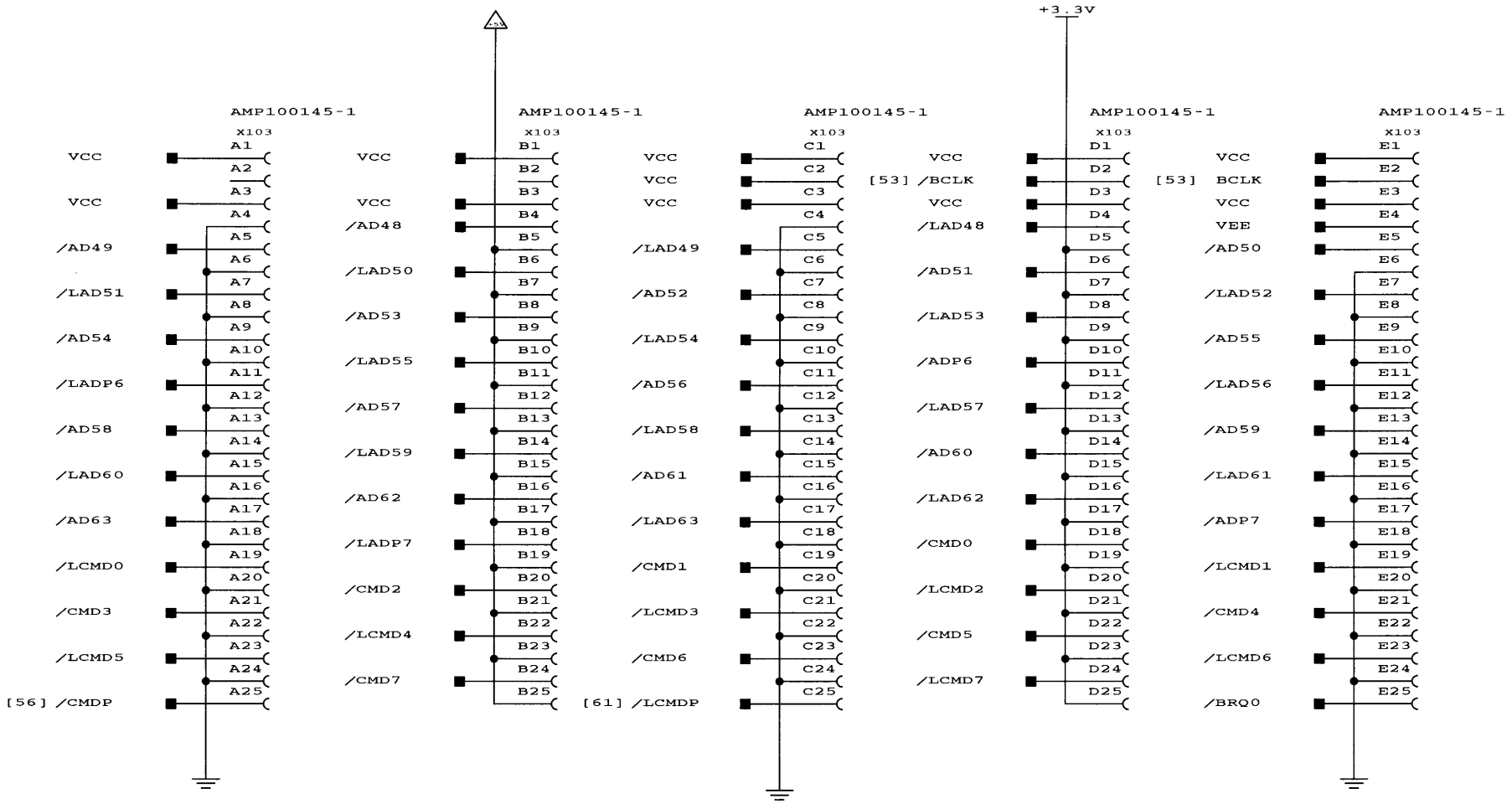
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Local control transceiver	
Issue 2			
Issue 3		File: cpu305-1	Page:63 of 72



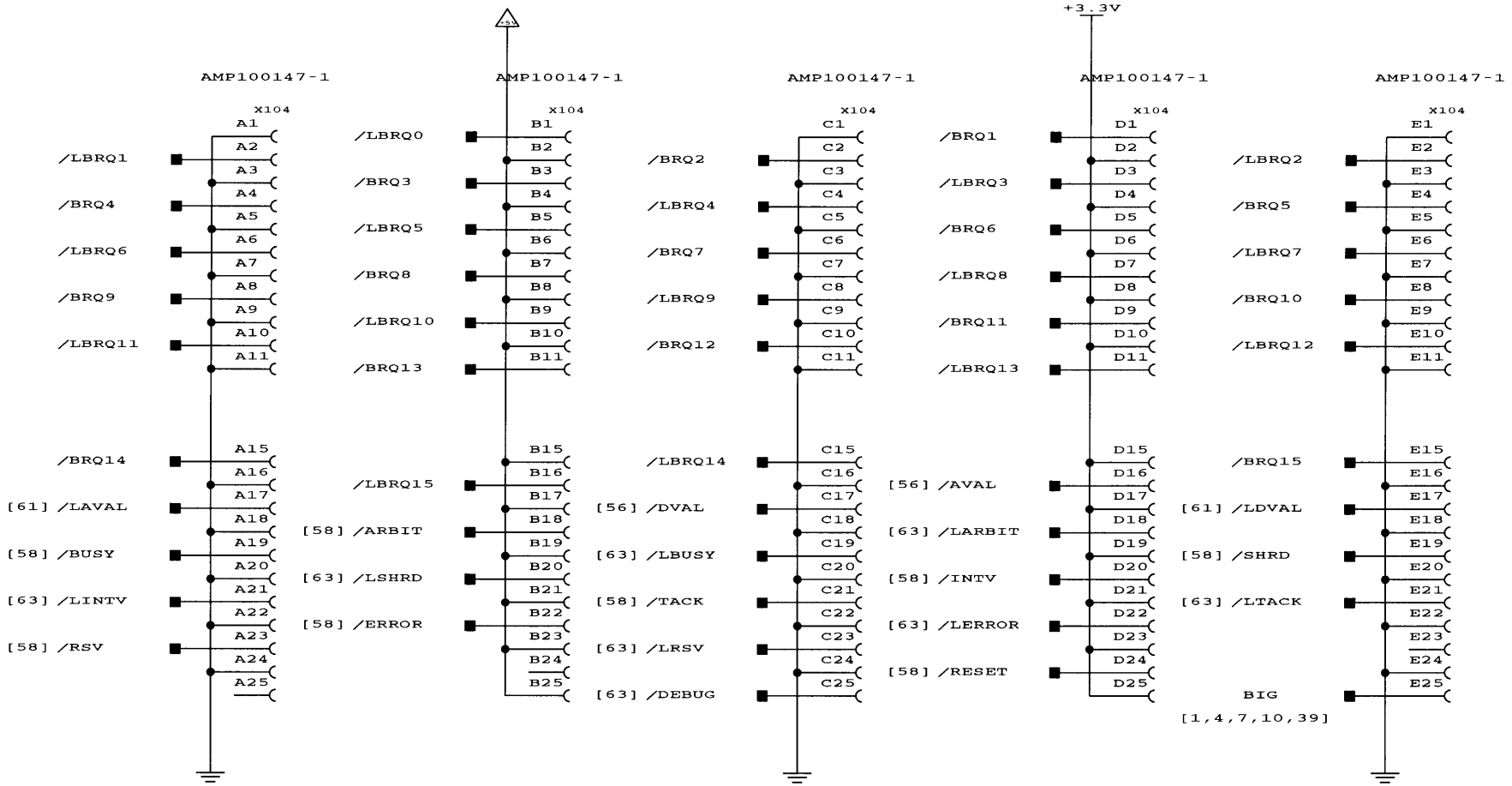
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Connector (row 1-25)	
Issue 2			
Issue 3		File: cpu305-1	Page:64 of 72



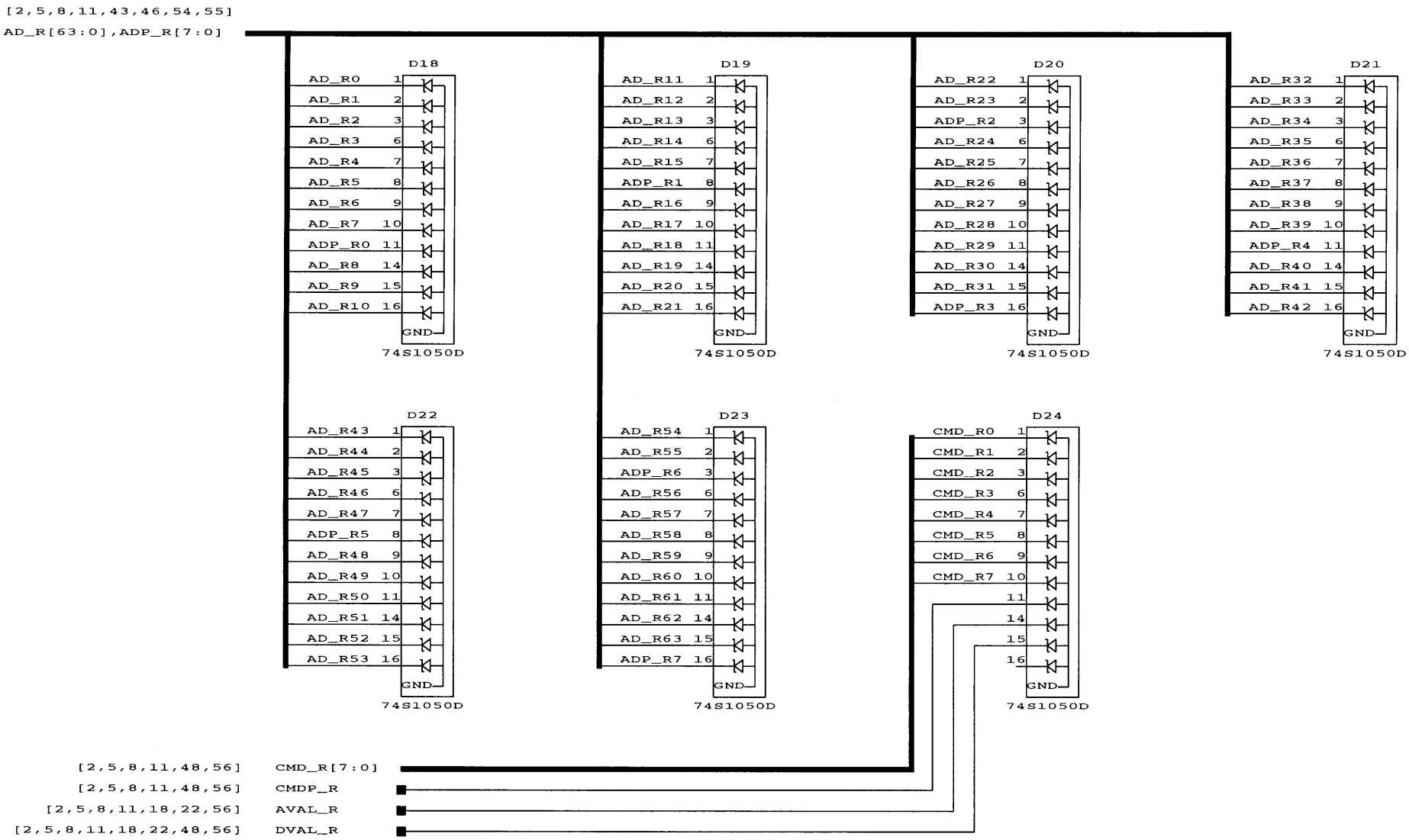
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Connector (row 26-50)	
Issue 2			
Issue 3		File: cpu305-1	Page:65 of 72



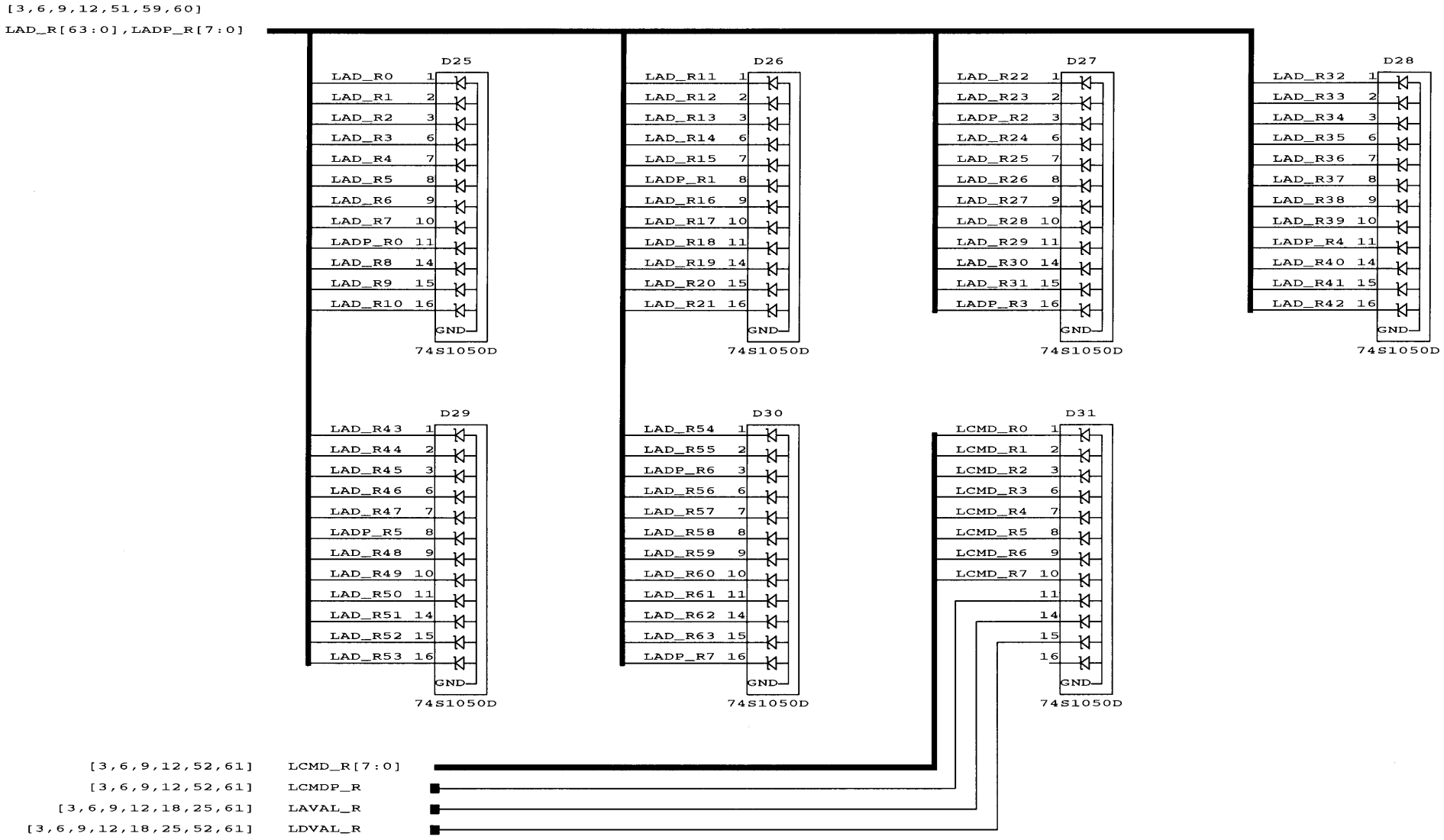
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Connector (row 51-75)	
Issue 2			
Issue 3		File: cpu305-1	Page:66 of 72



dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Connector (row 76-100)	
Issue 2			
Issue 3		File: cpu305-1	Page:67 of 72

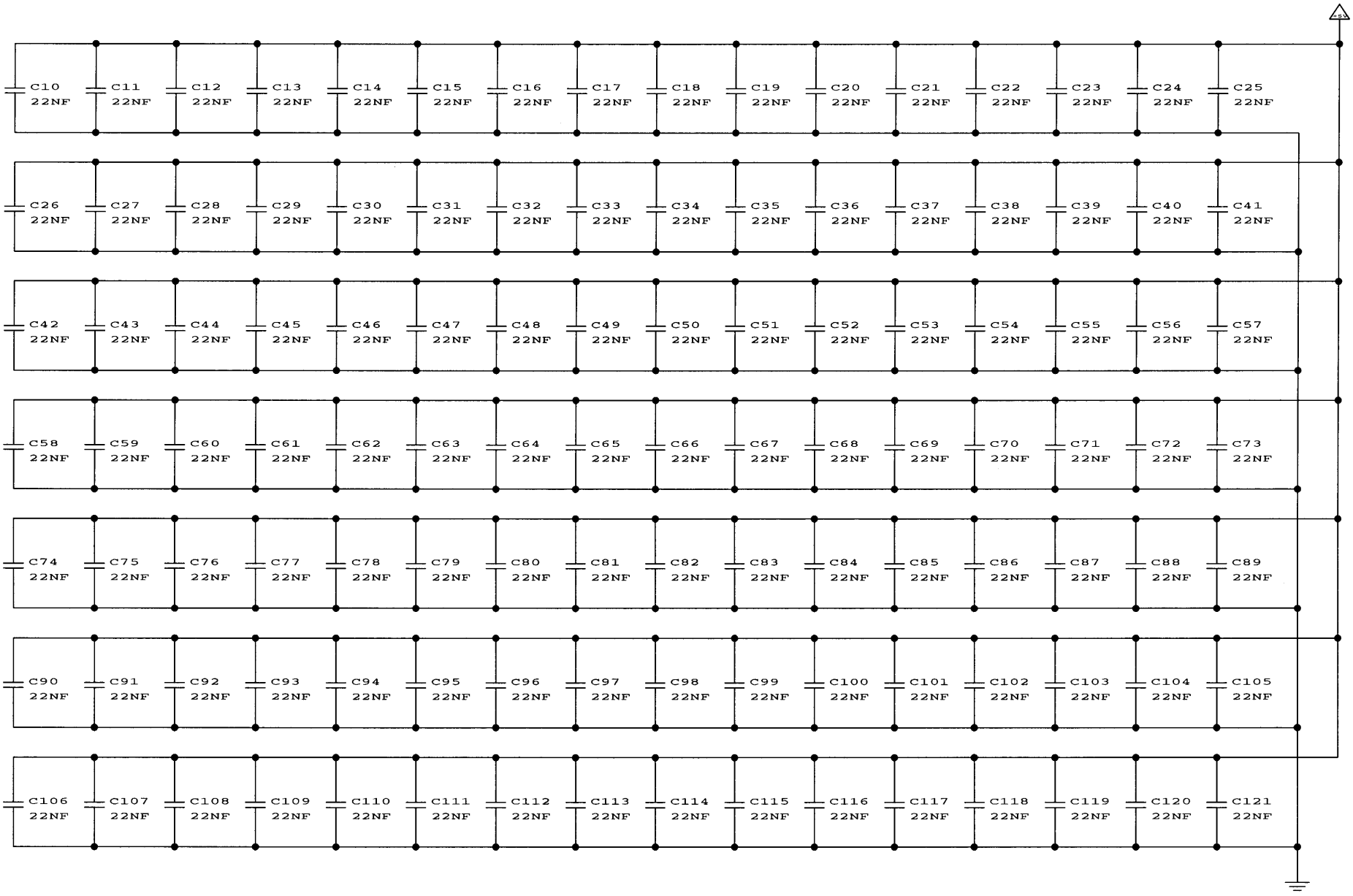


dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Internal global bus term.	
Issue 2			
Issue 3		File: cpu305-1	Page:68 of 72



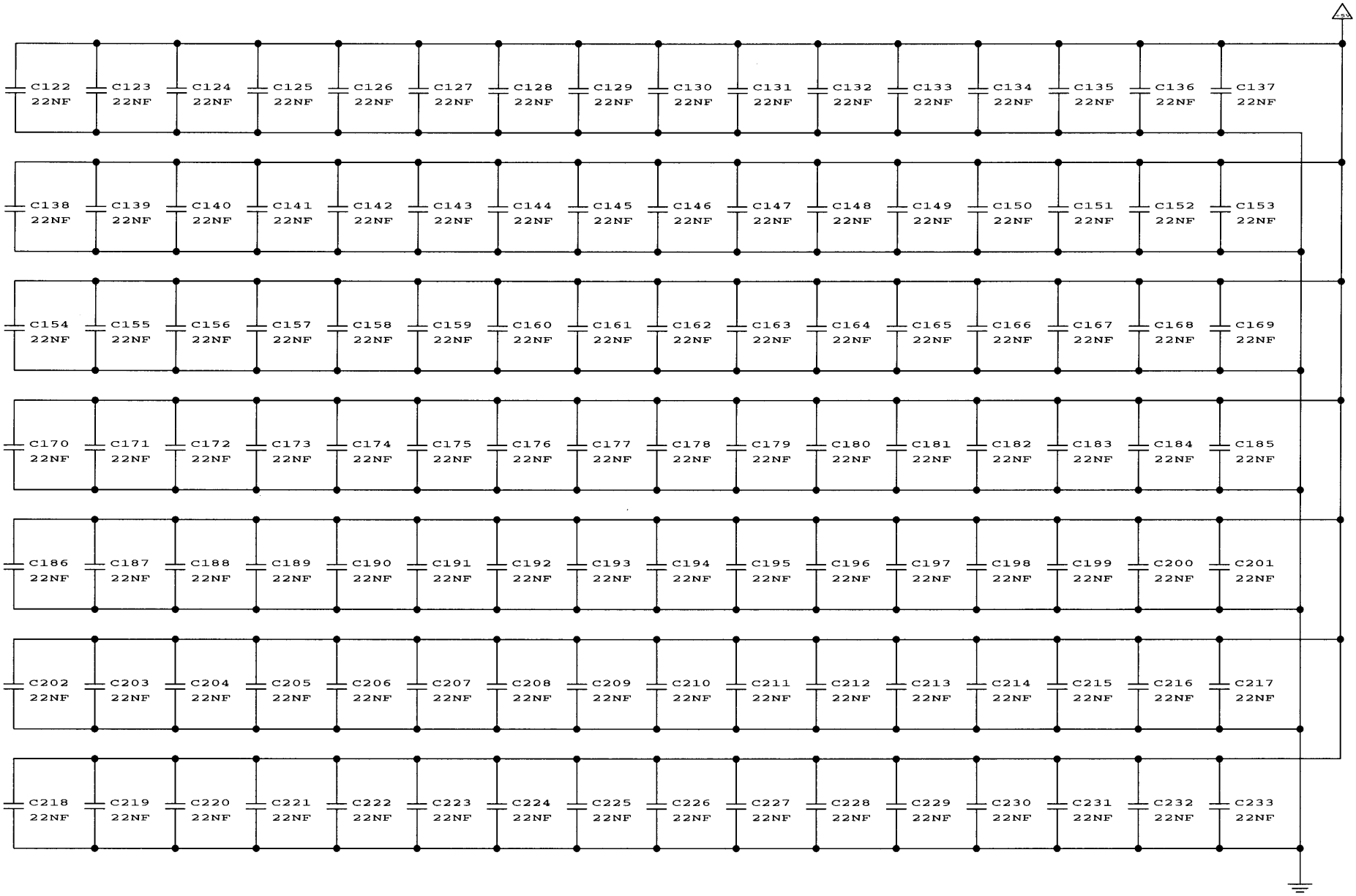
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Internal local bus term.	
Issue 2			
Issue 3		File: cpu305-1	Page:69 of 72

Chip level decoupling: 22nF/chip, 2*22nF/MACH chip, and 8*22nF/agent chip.



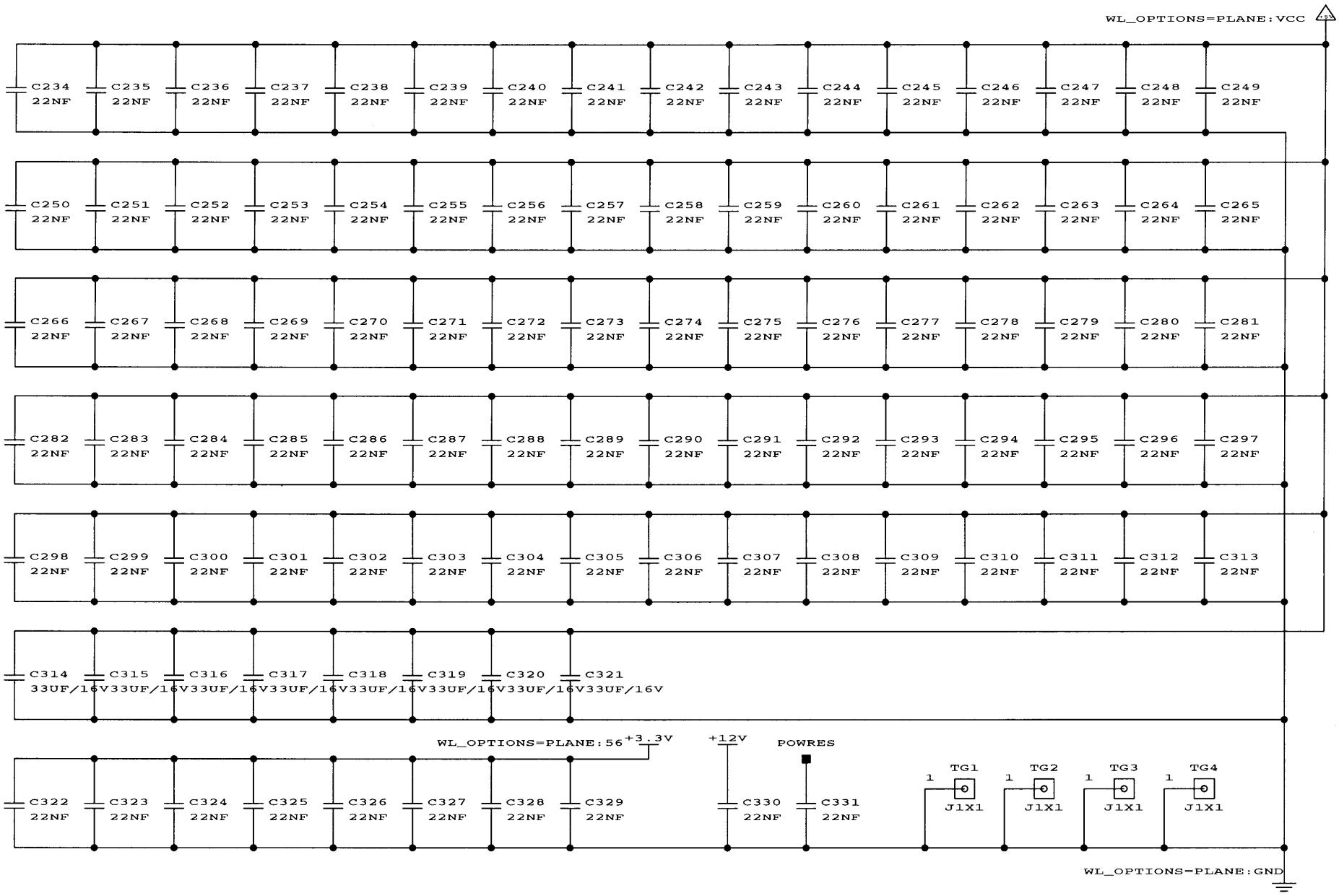
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Decoupling capacitors	
Issue 2			
Issue 3		File: cpu305-1	Page:70 of 72

Chip level decoupling: 22nF/chip, 2*22nF/MACH chip, and 8*22nF/agent chip.



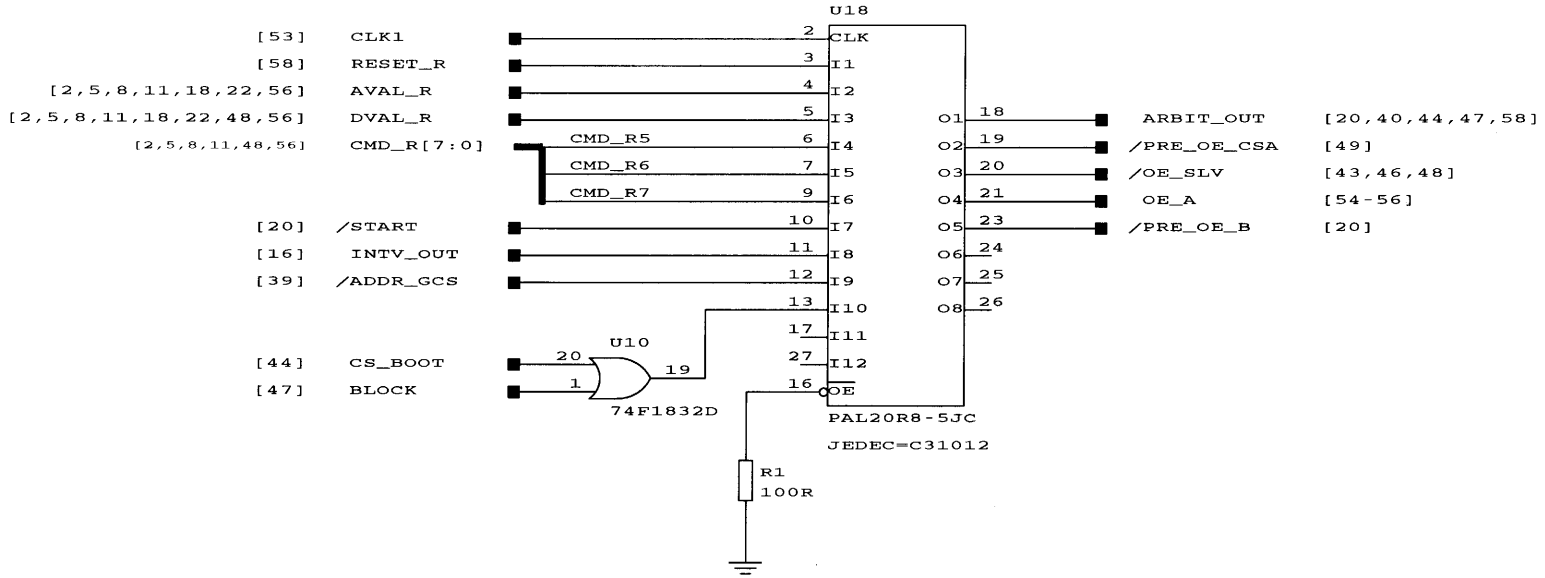
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Decoupling capacitors	
Issue 2			
Issue 3		File: cpu305-1	Page:71 of 72

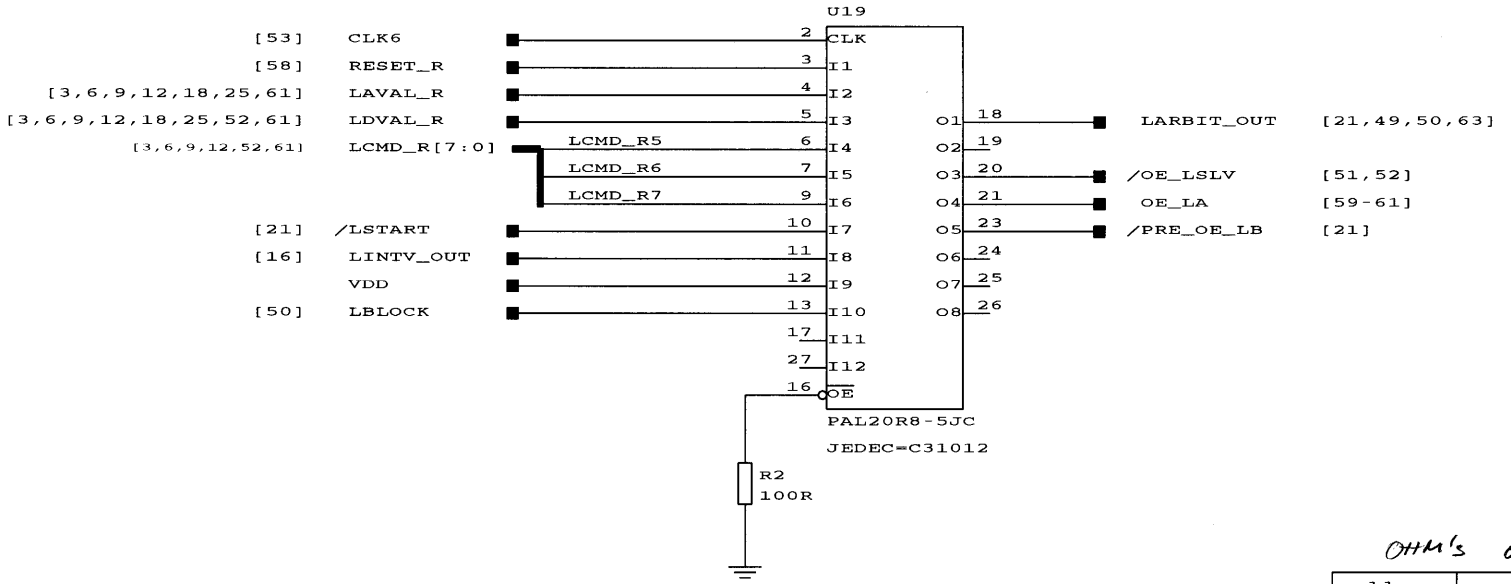
Chip level decoupling: 22nF/chip, 2*22nF/MACH chip, and 8*22nF/agent chip.



Decoupling at bus connector: 22nF for 3.3 V, 12 V, POWRES and 33uF for 5 V.

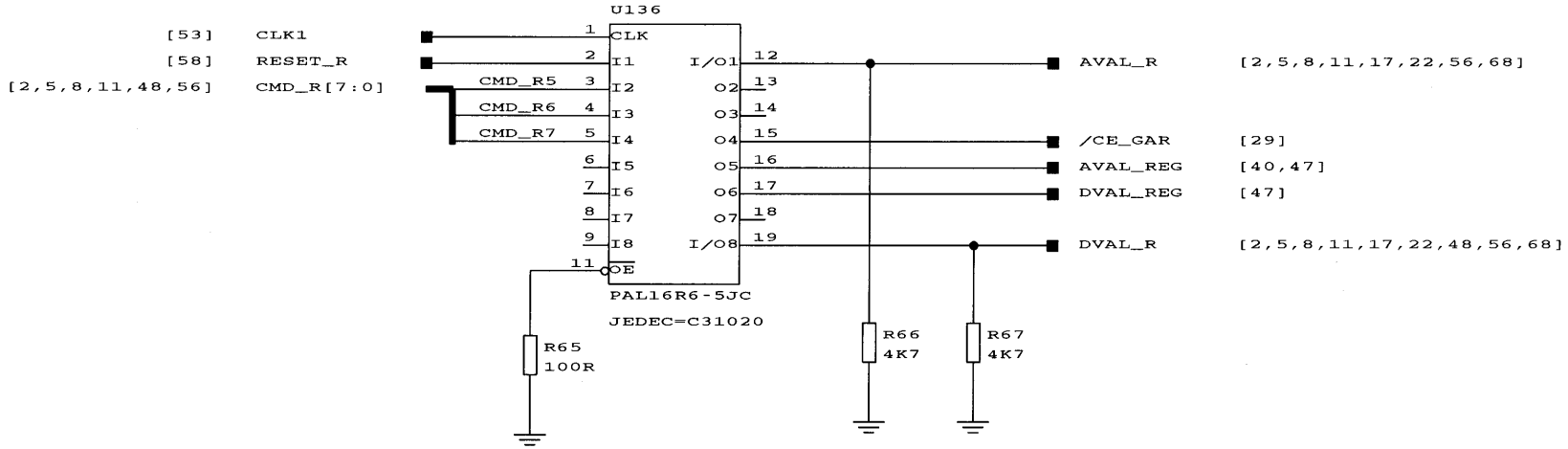
dde	Dansk Data Elektronik A/S		
Issue 0	96-06-07	CPU305-1 Module	
Issue 1		Decoupling capacitors	
Issue 2			
Issue 3		File: cpu305-1	Page:72 of 72

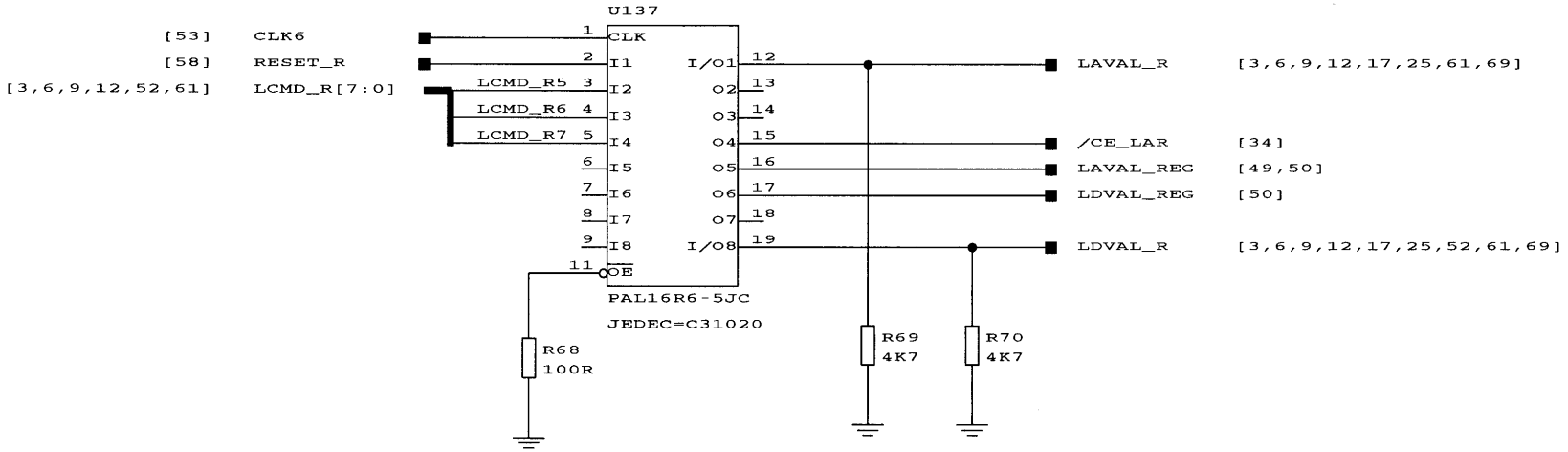




OHM's original

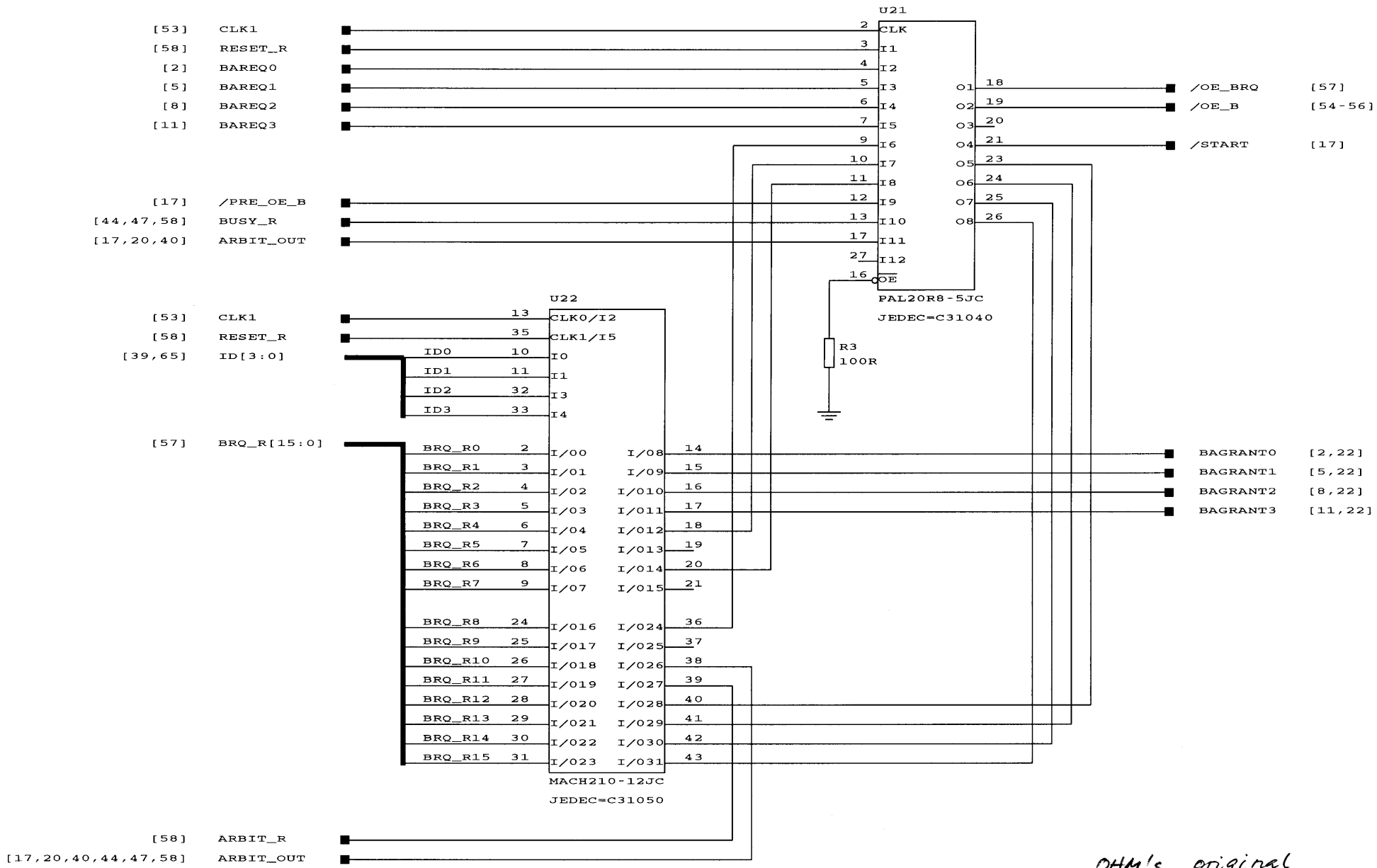
dde	Dansk Data Elektronik A/S		
Issue 0	940825	CPU305 Module	
Issue 1	950131	Global and local	
Issue 2	951129	output enable control	
Issue 3		File: cpu305-0	Page:17 of 72





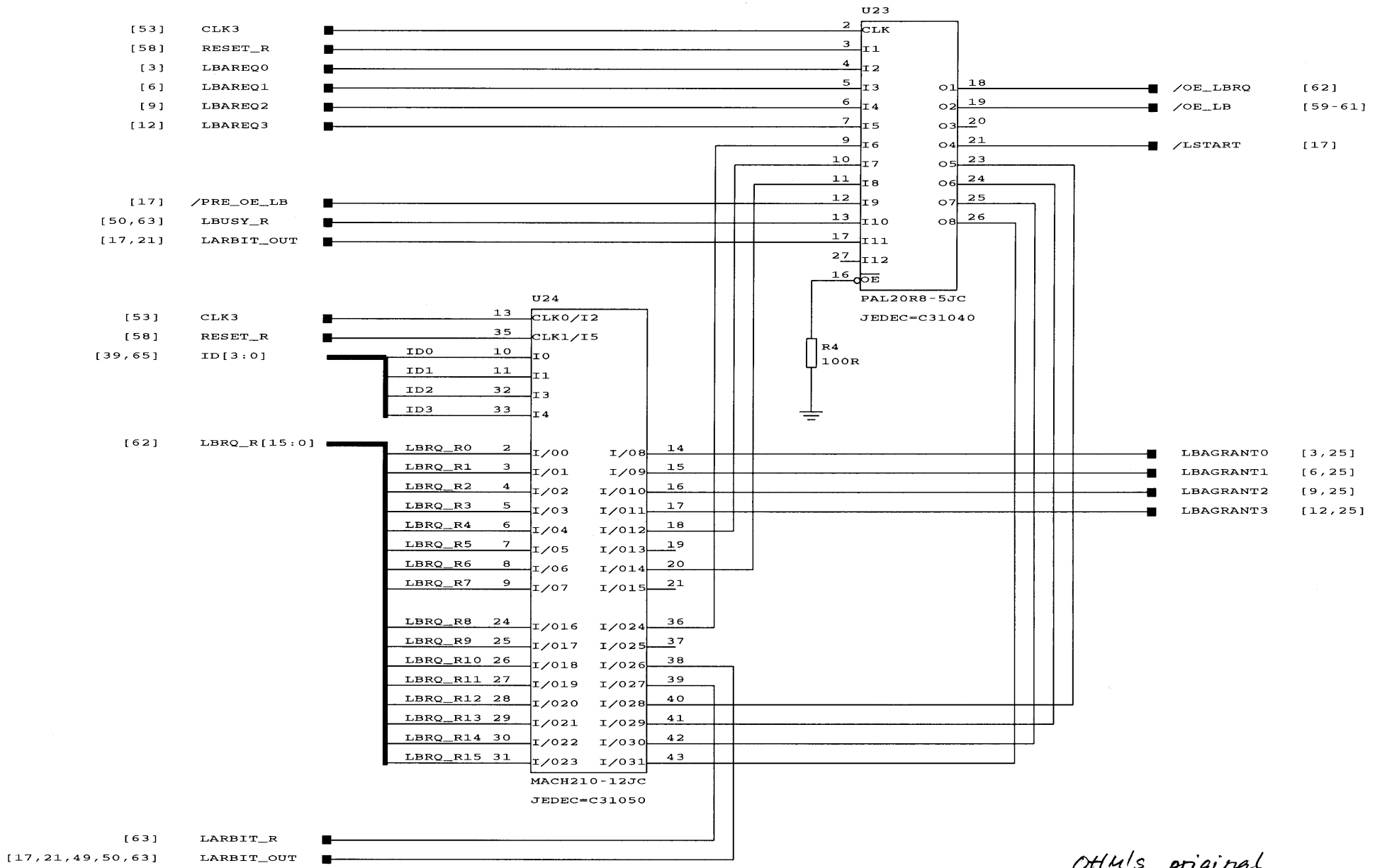
OHM's original

dde	Dansk Data Elektronik A/S	
Issue 0	940825	CPU305 Module
Issue 1	950131	Pull-down for
Issue 2	951129	address and data valid
Issue 3		File: cpu305-0 Page:18 of 72



OHM's original

dde	Dansk Data Elektronik A/S		
Issue 0	940825	CPU305 Module	
Issue 1	950131	Global bus arbitration	
Issue 2	951129		
Issue 3		File: cpu305-0	Page:20 of 72



OH4's original

dde	Dansk Data Elektronik A/S		
Issue 0	940825	CPU305 Module	
Issue 1	950131	Local bus arbitration	
Issue 2	951129		
Issue 3		File: cpu305-0	Page:21 of 72

```
module c31151
title 'Bus master arbitration controller for CPU304/5/6/7
       Dansk Data Elektronik A/S
       AAJ/96-04-12';

c31151 device 'mach210a';

"Inputs:
CLK         pin 13;
RESET       pin 35;

ID0         pin 10; "Bus position ID
ID1         pin 11;
ID2         pin 32;
ID3         pin 33;

BARQ0       pin 40; "Bus access requests from CA0-3
BARQ1       pin 41;
BARQ2       pin 42;
BARQ3       pin 43;

BRQ_R0      pin 2;  "Received bus requests
BRQ_R1      pin 3;
BRQ_R2      pin 4;
BRQ_R3      pin 5;
BRQ_R4      pin 6;
BRQ_R5      pin 7;
BRQ_R6      pin 8;
BRQ_R7      pin 9;
BRQ_R8      pin 24;
BRQ_R9      pin 25;
BRQ_R10     pin 26;
BRQ_R11     pin 27;
BRQ_R12     pin 28;
BRQ_R13     pin 29;
BRQ_R14     pin 30;
BRQ_R15     pin 20;

ARBIT_R     pin 21;"Received bus arbitration signal
BUSY_R      pin 37;"Received bus BUSY signal
RDWRF       pin 31;"Indicates Read w Wrt Forthcoming
TIMEIN      pin 36;"1 microsec timer input

"Outputs:
BS          pin 38 istype 'reg, buffer';"System bus state
BAGR0       pin 16 istype 'reg, buffer';"Grant to CA0
BAGR1       pin 17 istype 'reg, buffer';"         CA1
BAGR2       pin 18 istype 'reg, buffer';"         CA2
BAGR3       pin 19 istype 'reg, buffer';"         CA3
TMOARBIT    pin 15 istype 'reg, buffer';"Time-out ARBIT
!START      pin 14 istype 'reg, invert';"Bus cycle starts on next clk
!OE_BRQ     pin 39 istype 'reg, invert';"Outp enable for bus request


"Internal nodes:

WIN         node   istype 'com, buffer'; "Priority win
NOBRQ       node   istype 'com, buffer'; "No bus requests

P0          node   istype 'reg, buffer'; "Priority register
P1          node   istype 'reg, buffer';

PSTART      node   istype 'reg, buffer';"Waiting for !BUSY_R

BRC         node   istype 'reg, buffer';"Bus request control

BAC0        node   istype 'reg, buffer'; "Bus arbitration control
BAC1        node   istype 'reg, buffer';

TIME        node 76 istype 'reg, buffer'; "Sync. 1 microsec input
DTIME       node   istype 'reg, buffer';    "TIME delayed

TIM0        node   istype 'reg_t buffer'; "time out counter
TIM1        node   istype 'reg_t buffer';
TIM2        node   istype 'reg_t buffer';
TIM3        node   istype 'reg_t buffer';

"Constants:
ID      = [ID3..ID0];
BRQ_R   = [BRQ_R15..BRQ_R0];
ANYBRQ  = !NOBRQ;

BARQ    = [BARQ3..BARQ0];
BAGR    = [BAGR3..BAGR0];
GRANT   = BAGR!=0;

GP      = [P1..P0];
BARQP   =   BARQ0 & (GP==0)
          # BARQ1 & (GP==1)
          # BARQ2 & (GP==2)
          # BARQ3 & (GP==3);

CNTEN   =   TIME & !DTIME & BAC0.fb; "time out count enable
TIMEOUT=    TIM3.q & !TIM2.q & !TIM1.q & TIM0.q; "8 microsec time out

x       = .X.;
c       = .C.;

"Bus states:
bus     = [BS];
idle    = 0;
ms_sel  = 1;

"Bus request control states:
req_contr = [OE_BRQ, BRC];
```

```
brcs0     = ^b00; "waiting for BARQ and No BRQ_R condition
brcs1     = ^b10; "enable bus request
brcs2     = ^b01; "waiting for selected CA to release BARQ

"Bus arbitration control states:
control   = [TMOARBIT, BAC1, BAC0];
bacs0     = ^b000; "idle waiting for RDWRF
bacs1     = ^b001; "RDWRF issued, wait for bus req from same CA
bacs2     = ^b010; "grant write req
bacs3     = ^b100; "time out


equations

[BAGR0,BAGR1,BAGR2,BAGR3,BS,P0,P1,START,PSTART,OE_BRQ,BRC,
TMOARBIT,BAC1,BAC0,TIME,DTIME,TIM3,TIM2,TIM1,TIM0].clk = CLK;

NOBRQ=     BRQ_R==0;

WIN = (ID==0)  &  BRQ_R0
    # (ID==1)  & !BRQ_R0 &   BRQ_R1
    # (ID==2)  & !BRQ_R0 &  !BRQ_R1 &   BRQ_R2
    # (ID==3)  & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &   BRQ_R3
    # (ID==4)  & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 &  BRQ_R4
    # (ID==5)  & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 BRQ_R5
    # (ID==6)  & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &   BRQ_R6
    # (ID==7)  & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &  !BRQ_R6 &   BRQ_R7
    # (ID==8)  & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &  !BRQ_R6 &  !BRQ_R7 &   BRQ_R8
    # (ID==9)  & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &  !BRQ_R6 &  !BRQ_R7 &  !BRQ_R8 &  BRQ_R9
    # (ID==10) & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &  !BRQ_R6 &  !BRQ_R7 &  !BRQ_R8 & !BRQ_R9 &
                  BRQ_R10
    # (ID==11) & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &  !BRQ_R6 &  !BRQ_R7 &  !BRQ_R8 & !BRQ_R9 &
                 !BRQ_R10 &  BRQ_R11
    # (ID==12) & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &  !BRQ_R6 &  !BRQ_R7 &  !BRQ_R8 & !BRQ_R9 &
                 !BRQ_R10 & !BRQ_R11 &  BRQ_R12
    # (ID==13) & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &  !BRQ_R6 &  !BRQ_R7 &  !BRQ_R8 & !BRQ_R9 &
                 !BRQ_R10 & !BRQ_R11 & !BRQ_R12 &  BRQ_R13
    # (ID==14) & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &  !BRQ_R6 &  !BRQ_R7 &  !BRQ_R8 & !BRQ_R9 &
                 !BRQ_R10 & !BRQ_R11 & !BRQ_R12 & !BRQ_R13 &
                  BRQ_R14
    # (ID==15) & !BRQ_R0 &  !BRQ_R1 &  !BRQ_R2 &  !BRQ_R3 & !BRQ_R4 &
                 !BRQ_R5 &  !BRQ_R6 &  !BRQ_R7 &  !BRQ_R8 & !BRQ_R9 &
                 !BRQ_R10 & !BRQ_R11 & !BRQ_R12 & !BRQ_R13 &
                 !BRQ_R14 &  BRQ_R15;

"Grant pointer is loaded with current GRANT

P0       := (BAGR1 # BAGR3 ) & !RESET
          # P0 & !GRANT & !RESET;

P1       := (BAGR2 # BAGR3 ) & !RESET
          # P1 & !GRANT & !RESET;

BAGR0  := ((GP==3) & BARQ0
          # (GP==0) & BARQ0 & !BARQ1 & !BARQ2 & !BARQ3
          # (GP==1) & BARQ0 & !BARQ2 & !BARQ3
          # (GP==2) & BARQ0 & !BARQ3 ) & !BS.fb & WIN & !RESET
          # BAC1.fb & BARQ0 & (GP==0) & !RESET;

BAGR1  := ((GP==0) & BARQ1
          # (GP==1) & BARQ1 & !BARQ2 & !BARQ3 & !BARQ0
          # (GP==2) & BARQ1 & !BARQ3 & !BARQ0
          # (GP==3) & BARQ1 & !BARQ0 ) & !BS.fb & WIN & !RESET
          # BAC1.fb & BARQ1 & (GP==1) & !RESET;

BAGR2  := ((GP==1) & BARQ2
          # (GP==2) & BARQ2 & !BARQ3 & !BARQ0 & !BARQ1
          # (GP==3) & BARQ2 & !BARQ0 & !BARQ1
          # (GP==0) & BARQ2 & !BARQ1 ) & !BS.fb & WIN & !RESET
          # BAC1.fb & BARQ2 & (GP==2) & !RESET;

BAGR3  := ((GP==2) & BARQ3
          # (GP==3) & BARQ3 & !BARQ0 & !BARQ1 & !BARQ2
          # (GP==0) & BARQ3 & !BARQ1 & !BARQ2
          # (GP==1) & BARQ3 & !BARQ2 ) & !BS.fb & WIN & !RESET
          # BAC1.fb & BARQ3 & (GP==3) & !RESET;


"START control. START indicates that this board will start a new
"bus operation in the next clock cycle.

PSTART :=  !BS.fb & WIN & !RESET
          # BAC1.fb & !RESET
          # PSTART.fb & BUSY_R & !RESET;

START  :=  PSTART.fb & !BUSY_R & !RESET;


"Time out counter

TIME     := TIMEIN;
DTIME    := TIME;

TIM0.t   = TIM0.q & !BAC0.fb
          # CNTEN;
```

```
TIM1.t    = TIM1.q & ! BAC0.fb
          # TIM0.q & CNTEN;

TIM2.t    = TIM2.q & ! BAC0.fb
          # TIM1.q & TIM0.q & CNTEN;

TIM3.t    = TIM3.q & ! BAC0.fb
          # TIM2.q & TIM1.q & TIM0.q & CNTEN;


state_diagram bus

state idle:      if RESET then idle else
                   if ANYBRQ then ms_sel else idle;

state ms_sel:    if RESET then idle else
                   if ARBIT_R then idle else ms_sel;


state_diagram req_contr

state brcs0:     if RESET then brcs0 else
                   if ((BARQ!=0) & NOBRQ) then brcs1 else brcs0;

state brcs1:     if RESET then brcs0 else
                   if GRANT then brcs2 else brcs1;

state brcs2:     if RESET then brcs0 else
                   if !BARQP then brcs0 else brcs2;


state_diagram control

state bacs0:     if RESET then bacs0 else
                   if RDWRF then bacs1 else bacs0;

state bacs1:     if RESET then bacs0 else
                   if BARQP then bacs2 else
                   if TIMEOUT then bacs3 else bacs1;

state bacs2:     goto bacs0;

state bacs3:     goto bacs0;


test_vectors 'bus priority'

([RESET, ID, BRQ_R, BARQ, ARBIT_R, CLK] -> [BAGR0])

[1,  0,^b0000000000000000, ^b0000, 0, c] -> [0];"reset
[0,  0,^b0000000000000000, ^b0001, 0, c] -> [0];

[0,  0,^b1111111111111111, ^b0001, 0, c] -> [1];"GRANT
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  1,^b1111111111111111, ^b0001, 0, c] -> [0];"BS:= ms_sel
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle


[0,  0,^b1111111111111110, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  1,^b1111111111111110, ^b0001, 0, c] -> [1];"GRANT
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  2,^b1111111111111110, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0,  1,^b1111111111111100, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  2,^b1111111111111100, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  3,^b1111111111111100, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0,  2,^b1111111111111000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  3,^b1111111111111000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  4,^b1111111111111000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0,  3,^b1111111111110000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  4,^b1111111111110000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  5,^b1111111111110000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0,  4,^b1111111111100000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  5,^b1111111111100000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  6,^b1111111111100000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0,  5,^b1111111111000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  6,^b1111111111000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  7,^b1111111111000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0,  6,^b1111111110000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  7,^b1111111110000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
```

```
[0,  8,^b1111111110000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0,  7,^b1111111100000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  8,^b1111111100000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  9,^b1111111100000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0,  8,^b1111111000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0,  9,^b1111111000000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 10,^b1111111000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0, 9, ^b1111110000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 10,^b1111110000000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 11,^b1111110000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0, 10,^b1111100000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 11,^b1111100000000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 12,^b1111100000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0, 11,^b1111000000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 12,^b1111000000000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 13,^b1111000000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0, 12,^b1110000000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 13,^b1110000000000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 14,^b1110000000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0, 13,^b1100000000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 14,^b1100000000000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 15,^b1100000000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle

[0, 14,^b1000000000000000, ^b0001, 0, c] -> [0];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle
[0, 15,^b1000000000000000, ^b0001, 0, c] -> [1];
[0,  0,^b0000000000000000, ^b0001, 1, c] -> [0];"BS:= idle


test_vectors 'bus state'

([RESET, BRQ_R, ARBIT_R, CLK] -> [BS])
[1, ^b0000000000000000, 0, c] -> [0]; "reset - bus idle
[0, ^b0000000000000000, 0, c] -> [0]; "bus idle
[0, ^b0000000000011000, 0, c] -> [1]; "bus master selected
[0, ^b0000000000011000, 0, c] -> [1]; "bus master selected
[0, ^b0000000000010000, 0, c] -> [1]; "bus master selected
[0, ^b0000000000010000, 1, c] -> [0]; "ARBIT - bus idle
[0, ^b0000000000010000, 0, c] -> [1]; "bus master selected
[0, ^b0000000000010000, 0, c] -> [1]; "bus master selected
[0, ^b0000000000000000, 1, c] -> [0]; "ARBIT - bus idle
[0, ^b0000000000000000, 0, c] -> [0]; "bus idle


test_vectors 'bus request and start control'

([RESET,RDWRF, ID,BRQ_R, BARQ,BUSY_R, ARBIT_R, CLK] ->
                                    [!OE_BRQ, BAGR0, !START])

[1,0,  0,^b0000000000000000, ^b0000,0, 0, c] -> [1, 0, 1];"reset

[0,0,  0,^b0000000000000000, ^b0001,0, 0, c] -> [0, 0, 1];"OE_BRQ
[0,0,  0,^b0000000000000000, ^b0001,0, 0, c] -> [0, 0, 1];
[0,0,  0,^b0000000000000001, ^b0001,0, 0, c] -> [0, 1, 1];"GRANT
[0,0,  0,^b0000000000000001, ^b0001,0, 0, c] -> [1, 0, 0];"START
[0,0,  0,^b0000000000000000, ^b0000,0, 0, c] -> [1, 0, 1];
[0,0,  0,^b0000000000000000, ^b0000,0, 1, c] -> [1, 0, 1];

"Wait for !BUSY
[0,0,  0,^b0000000000000000, ^b0001,1, 0, c] -> [0, 0, 1];"OE_BRQ
[0,0,  0,^b0000000000000000, ^b0001,1, 0, c] -> [0, 0, 1];
[0,0,  0,^b0000000000000001, ^b0001,1, 0, c] -> [0, 1, 1];"GRANT
[0,0,  0,^b0000000000000001, ^b0001,1, 0, c] -> [1, 0, 1];
[0,0,  0,^b0000000000000001, ^b0001,1, 0, c] -> [1, 0, 1];
[0,0,  0,^b0000000000000001, ^b0001,0, 0, c] -> [1, 0, 0];"START
[0,0,  0,^b0000000000000000, ^b0000,0, 0, c] -> [1, 0, 1];
[0,0,  0,^b0000000000000000, ^b0000,0, 1, c] -> [1, 0, 1];

"Wait for NOBRQ
[0,0,  0,^b0000000000000010, ^b0001,0, 0, c] -> [1, 0, 1];"!NOBRQ
[0,0,  0,^b0000000000000010, ^b0001,0, 0, c] -> [1, 0, 1];"!NOBRQ
[0,0,  0,^b0000000000000010, ^b0001,0, 0, c] -> [1, 0, 1];"!NOBRQ
[0,0,  0,^b0000000000000000, ^b0001,0, 1, c] -> [0, 0, 1];"OE_BRQ
[0,0,  0,^b0000000000000000, ^b0001,0, 1, c] -> [0, 0, 1];
[0,0,  0,^b0000000000000000, ^b0001,0, 0, c] -> [0, 0, 1];
[0,0,  0,^b0000000000000001, ^b0001,0, 0, c] -> [0, 1, 1];"GRANT
```

```
[0,0,  0,^b0000000000000001, ^b0001,0, 0, c] -> [1, 0, 0];"START
[0,0,  0,^b0000000000000000, ^b0000,0, 0, c] -> [1, 0, 1];
[0,0,  0,^b0000000000000000, ^b0000,0, 1, c] -> [1, 0, 1];


test_vectors 'bus arbitration control'

([RESET, TIMEIN, RDWRF, ID, BRQ_R, ARBIT_R, BARQ, CLK] ->
[BAGR, TMOARBIT])

"Check fairness priority scheme for simultaneously BARQ's
[1, 0,0, 0,^b0000000000000000, 0,^b0000, c] -> [^b0000,0];"reset

[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b0010,0];"BAGR1
[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000000, 1,^b1111, c] -> [^b0000,0];

[0, 0,0, 0,^b0000000000000001, 0,^b0010, c] -> [^b0010,0];"BAGR1
[0, 0,0, 0,^b0000000000000001, 0,^b0010, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000000, 1,^b0010, c] -> [^b0000,0];

[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b0100,0];"BAGR2
[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000000, 1,^b1111, c] -> [^b0000,0];

[0, 0,0, 0,^b0000000000000001, 0,^b0100, c] -> [^b0100,0];"BAGR2
[0, 0,0, 0,^b0000000000000001, 0,^b0100, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000000, 1,^b0100, c] -> [^b0000,0];

[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b1000,0];"BAGR3
[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000000, 1,^b1111, c] -> [^b0000,0];

[0, 0,0, 0,^b0000000000000001, 0,^b1000, c] -> [^b1000,0];"BAGR3
[0, 0,0, 0,^b0000000000000001, 0,^b1000, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000000, 1,^b1000, c] -> [^b0000,0];

[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b0001,0];"BAGR0
[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000000, 1,^b1111, c] -> [^b0000,0];

[0, 0,0, 0,^b0000000000000001, 0,^b0001, c] -> [^b0001,0];"BAGR0
[0, 0,0, 0,^b0000000000000001, 0,^b0001, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000000, 1,^b0001, c] -> [^b0000,0];

"Check that RdwWr and the subsequent Wr are indivisible
[0, 0,0, 0,^b0000000000000001, 0,^b0001, c] -> [^b0001,0];"GP=0
[0, 0,0, 0,^b0000000000000000, 0,^b0000, c] -> [^b0000,0];
[0, 0,1, 0,^b0000000000000000, 0,^b0001, c] -> [^b0000,0];"bacs1
[0, 0,0, 0,^b0000000000000001, 0,^b1110, c] -> [^b0000,0];"bacs1
[0, 0,0, 0,^b0000000000000001, 0,^b1110, c] -> [^b0000,0];"bacs1
[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b0000,0];"bacs2
[0, 0,0, 0,^b0000000000000001, 0,^b1111, c] -> [^b0001,0];"BAGR0
[0, 0,0, 0,^b0000000000000001, 1,^b1111, c] -> [^b0000,0];

"Time out due to missing BARQ from agent that issued the RDWRF
[0, 0,0, 0,^b0000000000000001, 0,^b0010, c] -> [^b0010,0];"BAGR1
[0, 0,0, 0,^b0000000000000000, 0,^b0000, c] -> [^b0000,0];
[0, 1,1, 0,^b0000000000000000, 0,^b1101, c] -> [^b0000,0];"bacs1
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];"tim=1
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];"tim=2
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];"tim=3
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];"tim=4
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];"tim=5
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];"tim=6
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];"tim=7
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];"tim=8
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];"tim=9
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,1];"TMOARBIT
[0, 0,0, 0,^b0000000000000001, 1,^b1101, c] -> [^b0000,0];"ARBIT
[0, 1,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0100,0];"BAGR2
[0, 0,0, 0,^b0000000000000001, 0,^b1101, c] -> [^b0000,0];
[0, 0,0, 0,^b0000000000000000, 1,^b0000, c] -> [^b0000,0];

end c31151;
```

```
module c31161

title 'Active pull-down for AVAL and DVAL of CPU304/5/6/7
       Dansk Data Elektronik.
       AAJ/OHM/96-04-10';

c31161 device 'P16R6';

"Inputs:

RESET_R          pin 2;
CMD_R5           pin 3;
CMD_R6           pin 4;
CMD_R7           pin 5;
!PRE_OEB         pin 8;

"Combinational input/outputs:

AVAL_R           pin 12;
DVAL_R           pin 19;

"Register outputs:

!OE_ZAVAL        pin 13;
!OE_ZDVAL        pin 14;
!OEB             pin 15;
AVAL_REG         pin 16;
DVAL_REG         pin 17;

"Control:

CLK              pin 1;
!OE              pin 11;

"Definitions:

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

lde_id           = DVAL_R & !CMD_R7;

command          = [AVAL_R, DVAL_R, CMD_R7..CMD_R5];

none             = ^b00000;
read             = ^b10000;
rdwrf            = ^b10001;
write            = ^b10010;
null             = ^b10011;
inval            = ^b10100;
update           = ^b10101;
wr_inv           = ^b10110;
reserved         = ^b10111;
nlde             = ^b01100;
lde              = ^b01000;
undef            = [x,x,x,x,x];

reg_vector       = [AVAL_REG, DVAL_REG, OEB];


equations

reg_vector.clk = CLK;
reg_vector.oe = OE;
[OE_ZAVAL, OE_ZDVAL].clk= CLK;
[OE_ZAVAL, OE_ZDVAL].oe=  OE;

AVAL_R.oe = OE_ZAVAL;
DVAL_R.oe = OE_ZDVAL;

AVAL_R = 0;
DVAL_R = 0;

OE_ZAVAL:=         AVAL_R & !RESET_R;
OE_ZDVAL:=         lde_id & !RESET_R;

AVAL_REG:=         AVAL_R;
DVAL_REG:=         DVAL_R;

OEB:=              PRE_OEB & !RESET_R;


test_vectors 'Pull-down and delayed valid signals'

([RESET_R, command, CLK] ->
 [!OE_ZAVAL,!OE_ZDVAL, AVAL_R,DVAL_R, AVAL_REG,DVAL_REG])

[1, undef, c] -> [1,1, Z,Z, x,x];
[0, none,  c] -> [1,1, Z,Z, 0,0];

"Block read
[0, read,  c] -> [0,1, 0,Z, 1,0];
[0, none,  c] -> [1,1, Z,Z, 0,0];
[0, nlde,  c] -> [1,1, Z,Z, 0,1];
[0, lde,   c] -> [1,0, Z,0, 0,1];
[0, none,  c] -> [1,1, Z,Z, 0,0];

"Simple write
[0, write, c] -> [0,1, 0,Z, 1,0];
[0, lde,   c] -> [1,0, Z,0, 0,1];
[0, none,  c] -> [1,1, Z,Z, 0,0];


test_vectors 'System bus output enable'
```

```
([RESET_R, !PRE_OEB, CLK] -> !OEB)
[1, x, c] -> 1;
[0, 1, c] -> 1;
[0, 0, c] -> 0;

end c31161;
```

```
module c31171

title 'Output enable control for CPU304/5/6/7
       Dansk Data Elektronik.
       AAJ/OHM 96-04-11';

c31171 device 'P22V10c';

"Inputs:

CLK             pin 2;
RESET_R         pin 3;
AVAL_R          pin 4;
DVAL_R          pin 5;
CMD_R5          pin 6;
CMD_R6          pin 7;
CMD_R7          pin 9;
!START          pin 10;"GRANT delayed by BUSY_R
INTV_OUT        pin 11;
!ADDR_CS        pin 12;"Control space addr
BOOT_BLOCK      pin 13;
TMO_ARBIT       pin 16;"Arbit after rdwrf cmd time-out

"Register outputs:

 RDWRF          pin 17 istype 'reg, invert';"Rd w Wr Forthcoming cmd
 ARBIT_OUT      pin 18 istype 'reg, invert';"arbit to system bus
!PRE_OE_CSA     pin 19 istype 'reg, invert';"Pre OE data to contr space
!OE_SLV         pin 20 istype 'reg, invert';"OE data from contr space
 OE_A           pin 21 istype 'reg, invert';"OE A-side of BTL
!PRE_OE_B       pin 23 istype 'reg, invert';"Pre OE Bus side of BTL
!N0             pin 24 istype 'reg, invert';"State control
!N1             pin 25 istype 'reg, invert';
!N2             pin 26 istype 'reg, invert';
 P_START        pin 27 istype 'reg, invert';"Pending START


"Definitions:

c                = .C.;
x                = .X.;

"Write command includes all non-read commands
"for the purpose of output enable control.

rdwrf_cmd        = AVAL_R & !CMD_R7 & !CMD_R6 & CMD_R5;
not_rdwrf_cmd    = AVAL_R & !(!CMD_R7 & !CMD_R6 & CMD_R5);
read_cmd         = AVAL_R & !(CMD_R7 # CMD_R6);
write_cmd        = AVAL_R & (CMD_R7 # CMD_R6);
lde_id           = DVAL_R & !CMD_R7;

command          = [ AVAL_R, DVAL_R, CMD_R7..CMD_R5 ];

none             = ^b00000;
read             = ^b10000;
rdwrf            = ^b10001;
write            = ^b10010;
null             = ^b10011;
inval            = ^b10100;
update           = ^b10101;
wr_inv           = ^b10110;
reserved         = ^b10111;
nlde             = ^b01100;
lde              = ^b01000;


"The two idle states prevent outgoing data from being echoed by the
"registered BTL transceiver. Similarly, the states afa_rd and
"afa_rd_m0 prevent an outgoing address from being echoed.

state_vector = [OE_A, PRE_OE_B, OE_SLV, PRE_OE_CSA, N2,N1,N0 ];

idle1         = ^b0000000; "First idle cycle after BTL send.
idle2         = ^b0000010; "Second idle cycle after BTL send.
listen        = ^b1001000; "Listen to bus.
addr          = ^b0101000; "Agent emits address.
afa_rd        = ^b0000001; "Decide between memory and ctrl.sp.
afa_rd_m0     = ^b0000011; "Do not echo read address from agent.
afa_rd_m      = ^b1000110; "Agent or foreign agent reads memory.
afa_rd_cs     = ^b0110110; "Agent or foreign agent reads ctrl.sp.
a_wr_mcs      = ^b0101100; "Agent writes memory or control space.
fa_wr_mcs     = ^b1001100; "Foreign agent writes memory or ctrl.sp.
intv          = ^b0101101; "Intervention.
                           "(Memory includes foreign control space).
undef_st      = ^b1000000; "Undefined state

addr_state    = PRE_OE_B.fb & PRE_OE_CSA.fb & !N2.fb;


equations

state_vector.clk =                CLK;

[ARBIT_OUT, RDWRF, P_START].clk = CLK;


ARBIT_OUT:=     addr_state & not_rdwrf_cmd & !RESET_R
              # TMO_ARBIT & !RESET_R;

RDWRF:=         addr_state & rdwrf_cmd & !RESET_R;

P_START:=       START & !RESET_R
              # P_START & !addr_state & !RESET_R;
```

```
@dcset

state_diagram state_vector

state idle1:    if RESET_R then listen else
                  if (START # P_START) then addr else idle2;

state idle2:    if RESET_R then listen else
                  if (START # P_START) then addr else listen;

state listen:   if RESET_R then listen else
                  if read_cmd then afa_rd else
                    if write_cmd then fa_wr_mcs else
                      if (START # P_START) then addr else listen;

state addr:     if RESET_R then listen else
                  case
                    read_cmd  : afa_rd;
                    write_cmd : a_wr_mcs;
                  endcase;

state afa_rd:   if RESET_R then listen else
                  if !ADDR_CS then afa_rd_m0 else afa_rd_cs;

state afa_rd_m0:if RESET_R then listen else afa_rd_m;

state afa_rd_m: if RESET_R then listen else
                  if BOOT_BLOCK then afa_rd_cs else
                    if INTV_OUT then intv else
                      if lde_id then listen else afa_rd_m;

state afa_rd_cs:if RESET_R then listen else
                  if lde_id then idle1 else afa_rd_cs;

state a_wr_mcs: if RESET_R then listen else
                  if lde_id then idle1 else a_wr_mcs;

state fa_wr_mcs:if RESET_R then listen else
                  if lde_id then listen else fa_wr_mcs;

state intv:     if RESET_R then listen else
                  if lde_id then idle1 else intv;

state undef_st: goto listen;


test_vectors 'state machine'

([RESET_R, command, START,
   ADDR_CS, INTV_OUT, BOOT_BLOCK, CLK] -> state_vector)

"Agent or foreign agent reads from memory.

[1, x,      x, x,x,x, c] -> listen;
[1, x,      x, x,x,x, c] -> listen;
[0, 0,      0, x,x,x, c] -> listen;
[0, 0,      1, x,x,x, c] -> addr;
[0, read,   0, x,x,x, c] -> afa_rd;
[0, 0,      0, 0,x,x, c] -> afa_rd_m0;
[0, 0,      0, x,0,0, c] -> afa_rd_m;
[0, nlde,   0, x,0,0, c] -> afa_rd_m;
[0, lde,    0, x,0,0, c] -> listen;

"Agent or foreign agent reads from control space.

[0, 0,      1, x,x,x, c] -> addr;
[0, read,   0, x,x,x, c] -> afa_rd;
[0, 0,      0, 1,x,x, c] -> afa_rd_cs;
[0, nlde,   0, x,x,x, c] -> afa_rd_cs;
[0, lde,    0, x,x,x, c] -> idle1;
[0, x,      0, x,x,x, c] -> idle2;

"Agent writes to memory or control space.

[0, 0,      1, x,x,x, c] -> addr;
[0, write,  0, x,x,x, c] -> a_wr_mcs;
[0, nlde,   0, x,x,x, c] -> a_wr_mcs;
[0, lde,    0, x,x,x, c] -> idle1;
[0, x,      0, x,x,x, c] -> idle2;

"Foreign agent writes to memory or control space.

[0, 0,      0, x,x,x, c] -> listen;
[0, write,  0, x,x,x, c] -> fa_wr_mcs;
[0, nlde,   0, x,x,x, c] -> fa_wr_mcs;
[0, lde,    0, x,x,x, c] -> listen;

"Agent memory read with intervention.

[0, 0,      1, x,x,x, c] -> addr;
[0, read,   0, x,x,x, c] -> afa_rd;
[0, 0,      0, 0,x,x, c] -> afa_rd_m0;
[0, 0,      0, x,1,0, c] -> afa_rd_m;
[0, 0,      0, x,1,0, c] -> intv;
[0, 0,      0, x,x,x, c] -> intv;
[0, nlde,   0, x,x,x, c] -> intv;
[0, lde,    0, x,x,x, c] -> idle1;

"Agent reads boot PROM or dummy block.

[0, x,      1, x,x,x, c] -> addr;
[0, read,   0, x,x,x, c] -> afa_rd;
[0, 0,      0, 0,x,x, c] -> afa_rd_m0;
[0, 0,      0, x,x,x, c] -> afa_rd_m;
[0, nlde,   0, x,x,1, c] -> afa_rd_cs;
```

```
[0, lde,    0, x,x,x, c] -> idle1;
[0, x,      0, x,x,x, c] -> idle2;

"Start from idle2.

[0, x,      1, x,x,x, c] -> addr;


test_vectors 'Arbitration output'

([RESET_R, command, START, TMO_ARBIT, CLK] ->
[ARBIT_OUT, RDWRF])

[1, none,    0,0, c] -> [0, 0];"reset
[0, none,    1,0, c] -> [0, 0];"start
[0, read,    0,0, c] -> [1, 0];"command

[1, none,    0,0, c] -> [0, 0];"reset
[0, none,    1,0, c] -> [0, 0];"start
[0, rdwrf,   0,0, c] -> [0, 1];"no arbit out, RDWRF

[1, none,    0,0, c] -> [0, 0];"reset
[0, none,    1,0, c] -> [0, 0];"start
[0, write,   0,0, c] -> [1, 0];

[1, none,    0,0, c] -> [0, 0];"reset
[0, none,    1,0, c] -> [0, 0];"start
[0, null,    0,0, c] -> [1, 0];

[1, none,    0,0, c] -> [0, 0];"reset
[0, none,    1,0, c] -> [0, 0];"start
[0, inval,   0,0, c] -> [1, 0];

[1, none,    0,0, c] -> [0, 0];"reset
[0, none,    1,0, c] -> [0, 0];"start
[0, update,  0,0, c] -> [1, 0];

[1, none,    0,0, c] -> [0, 0];"reset
[0, none,    1,0, c] -> [0, 0];"start
[0, wr_inv,  0,0, c] -> [1, 0];

[1, none,    0,0, c] -> [0, 0];"reset
[0, none,    0,0, c] -> [0, 0];"start
[0, none,    0,0, c] -> [0, 0];
[0, none,    0,1, c] -> [1, 0];"tmo_arbit
[0, none,    0,0, c] -> [0, 0];


end c31171;
```

```
module cspace

title
'Control space for CPU304/5/6/7
Dansk Data Elektronik.
Ole Moller.
September 18, 1993.';
"January 11, 1994. Selective reset asserted during reset.
"January 18, 1994. Boot target acknowledge only for read.
"January 19, 1994. Selective reset asserted after reset.
"October  6, 1994. ADDR_GCS always qualified by AVAL_REG.
"AAJ/96-04-12       ARBIT_OUT inp. delayed 1 clock.

c31081 device 'mach220a';

"Inputs:

CLK             pin 15;
RESET_R         pin 16;

AVAL_REG        pin 49;
!ADDR_GCS       pin 50;
BUSY_R          pin 17;
LDE_SLV         pin 26;

GCR5            pin 20;
GCR6            pin 51;
GCR7            pin 54;

GAR8            pin  2;
GAR9            pin  3;
GAR10           pin  4;
GAR11           pin  5;

GAR20           pin 21;
GAR21           pin 22;
GAR22           pin 23;
GAR23           pin 24;

ERROR_OUT       pin  6;
LERROR_OUT      pin  7;

ERROR_R         pin 36;
LERROR_R        pin 37;
DEBUG_R         pin 38;

ARBIT_OUT       pin 39;
!CS_BOOT        pin 25;

!PRE0           pin 40;
!PRE1           pin 41;
!PRE2           pin 59;
!PRE3           pin 60;

"Input/output:

D_CS0           pin 55 istype 'com, buffer';
D_CS1           pin 56 istype 'com, buffer';
D_CS2           pin 57 istype 'com, buffer';
D_CS3           pin 58 istype 'com, buffer';

"Register outputs:

ERROR_OUT_REG   node 73 istype 'reg, buffer'; "Input register
LERROR_OUT_REG  node 74 istype 'reg, buffer'; "Input register

SUBPOS0         node istype 'reg, buffer';
SUBPOS1         node istype 'reg, buffer';
SUBPOS2         node istype 'reg, buffer';
SUBPOS3         node istype 'reg, buffer';

GBE             node istype 'reg, buffer';
LBE             node istype 'reg, buffer';

SRE0            pin 11 istype 'reg, buffer';
SRE1            pin 12 istype 'reg, buffer';
SRE2            pin 13 istype 'reg, buffer';
SRE3            pin 14 istype 'reg, buffer';

GPI0            node istype 'reg, buffer';
GPI1            node istype 'reg, buffer';
GPI2            node istype 'reg, buffer';
GPI3            node istype 'reg, buffer';

LPI0            node istype 'reg, buffer';
LPI1            node istype 'reg, buffer';
LPI2            node istype 'reg, buffer';
LPI3            node istype 'reg, buffer';

DBI0            node istype 'reg, buffer';
DBI1            node istype 'reg, buffer';
DBI2            node istype 'reg, buffer';
DBI3            node istype 'reg, buffer';

!INT0           pin 30 istype 'reg, invert';
!INT1           pin 31 istype 'reg, invert';
!INT2           pin 32 istype 'reg, invert';
!INT3           pin 33 istype 'reg, invert';

OE_ST_ERR       node istype 'reg, buffer';
!OE_ID          pin 46 istype 'reg, invert';
!OE_FCN         pin 48 istype 'reg, invert';
!OE_PM_ID       pin 47 istype 'reg, invert';
SEL_ERROR       node istype 'reg, buffer';
OE_DUMMY        node istype 'reg, buffer';
```

```
CLR_STATUS      node istype 'reg, buffer';
CE_CTRL         node istype 'reg, buffer';
CLR_ERROR       node istype 'reg, buffer';
DVAL_GCS        pin 66 istype 'reg, buffer';
N0              pin  istype 'reg, buffer';
N1              pin  istype 'reg, buffer';

ADDR_GCS_REG    node istype 'reg, buffer';
DELAY2          node istype 'reg, buffer';
DELAY3          node istype 'reg, buffer';
TACK_OUT        pin 65 istype 'reg, buffer';

!ERROR_LED      pin  9 istype 'com, invert';

AMDMACH property 'GROUP C SUBPOS0 SUBPOS1 SUBPOS2 SUBPOS3'; "nodes
AMDMACH property 'GROUP B GBE LBE';                         "nodes
AMDMACH property 'GROUP A GPI0 GPI1 GPI2 GPI3';             "nodes
AMDMACH property 'GROUP E LPI0 LPI1 LPI2 LPI3';             "nodes
AMDMACH property 'GROUP G DBI0 DBI1 DBI2 DBI3';             "nodes
AMDMACH property 'GROUP F OE_ST_ERR SEL_ERROR OE_DUMMY';    "nodes
AMDMACH property 'GROUP F CLR_STATUS CE_CTRL CLR_ERROR';    "nodes
AMDMACH property 'GROUP H N0 N1';
AMDMACH property 'GROUP H DELAY2 DELAY3';                   "nodes

"Definitions:

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

cmd             = [ AVAL_REG, ADDR_GCS, GCR7..GCR5 ];

u               = [ x,x, x,x,x ]; "undefined"
n               = [ 0,0, x,x,x ]; "no command"
read            = [ 1,1, 0,0,0 ];
read1           = [ 1,1, 0,0,1 ];
write           = [ 1,1, 0,1,0 ];
null            = [ 1,1, 0,1,1 ];
invalidate      = [ 1,1, 1,0,0 ];
update          = [ 1,1, 1,0,1 ];
write_inv       = [ 1,1, 1,1,0 ];
reserved        = [ 1,1, 1,1,1 ];

read_cmd        = (cmd==read) # (cmd==read1);
write_cmd       = (cmd==write) # (cmd==write_inv);

offset          = [ GAR11..GAR8, x,x,x,x,x,x,x,x ];
subpos          = [ GAR23..GAR20 ];

D_CS            = [ D_CS3..D_CS0 ];

input_register  = [ ERROR_OUT_REG, LERROR_OUT_REG ];

subpos_register = [ SUBPOS3..SUBPOS0 ];

status_register = [ LBE, GBE ];
ctrl_register0  = SRE0;
ctrl_register1  = SRE1;
ctrl_register2  = SRE2;
ctrl_register3  = SRE3;
error_register0 = [ DBI0, LPI0, GPI0 ];
error_register1 = [ DBI1, LPI1, GPI1 ];
error_register2 = [ DBI2, LPI2, GPI2 ];
error_register3 = [ DBI3, LPI3, GPI3 ];

interrupts      = [ INT3, INT2, INT1, INT0 ];

tack_vector     = [ ADDR_GCS_REG, DELAY2, DELAY3, TACK_OUT ];

state_vector    = [ OE_ST_ERR, OE_ID, OE_FCN, OE_PM_ID, SEL_ERROR,
                    OE_DUMMY,
                    CLR_STATUS, CE_CTRL, CLR_ERROR,
                    DVAL_GCS, N1, N0 ];

idle            = [ 0,0,0,0,0, 0, 0,0,0, 0,0,0 ];
status          = [ 1,0,0,0,0, 0, 0,0,0, 0,0,0 ];
status2         = [ 1,0,0,0,0, 0, 0,0,0, 0,0,1 ];
status3         = [ 1,0,0,0,0, 0, 0,0,0, 0,1,0 ];
status4         = [ 1,0,0,0,0, 0, 0,0,0, 1,0,0 ];
id              = [ 0,1,0,0,0, 0, 0,0,0, 0,0,0 ];
id2             = [ 0,1,0,0,0, 0, 0,0,0, 0,0,1 ];
id3             = [ 0,1,0,0,0, 0, 0,0,0, 0,1,0 ];
id4             = [ 0,1,0,0,0, 0, 0,0,0, 1,0,0 ];
fcn             = [ 0,0,1,0,0, 0, 0,0,0, 0,0,0 ];
fcn2            = [ 0,0,1,0,0, 0, 0,0,0, 0,0,1 ];
fcn3            = [ 0,0,1,0,0, 0, 0,0,0, 0,1,0 ];
fcn4            = [ 0,0,1,0,0, 0, 0,0,0, 1,0,0 ];
pm_id           = [ 0,0,0,1,0, 0, 0,0,0, 0,0,0 ];
pm_id2          = [ 0,0,0,1,0, 0, 0,0,0, 0,0,1 ];
pm_id3          = [ 0,0,0,1,0, 0, 0,0,0, 0,1,0 ];
pm_id4          = [ 0,0,0,1,0, 0, 0,0,0, 1,0,0 ];
error           = [ 1,0,0,0,1, 0, 0,0,0, 0,0,0 ];
error2          = [ 1,0,0,0,1, 0, 0,0,0, 0,0,1 ];
error3          = [ 1,0,0,0,1, 0, 0,0,0, 0,1,0 ];
error4          = [ 1,0,0,0,1, 0, 0,0,0, 1,0,0 ];
dummy           = [ 0,0,0,0,0, 1, 0,0,0, 0,0,0 ];
dummy2          = [ 0,0,0,0,0, 1, 0,0,0, 0,0,1 ];
dummy3          = [ 0,0,0,0,0, 1, 0,0,0, 0,1,0 ];
dummy4          = [ 0,0,0,0,0, 1, 0,0,0, 1,0,0 ];
clr_stat        = [ 0,0,0,0,0, 0, 1,0,0, 0,0,0 ];
control         = [ 0,0,0,0,0, 0, 0,1,0, 0,0,0 ];
clr_error       = [ 0,0,0,0,0, 0, 0,0,1, 0,0,0 ];

ce_ctrl0        = CE_CTRL.fb & SUBPOS0;
ce_ctrl1        = CE_CTRL.fb & SUBPOS1;
```

```
ce_ctrl2        = CE_CTRL.fb & SUBPOS2;
ce_ctrl3        = CE_CTRL.fb & SUBPOS3;

clr_error0      = CLR_ERROR.fb & SUBPOS0;
clr_error1      = CLR_ERROR.fb & SUBPOS1;
clr_error2      = CLR_ERROR.fb & SUBPOS2;
clr_error3      = CLR_ERROR.fb & SUBPOS3;

equations

D_CS.oe = OE_ST_ERR.fb;
input_register.clk = CLK;
subpos_register.clk = CLK;
status_register.clk = CLK;
ctrl_register0.clk = CLK;
ctrl_register1.clk = CLK;
ctrl_register2.clk = CLK;
ctrl_register3.clk = CLK;
error_register0.clk = CLK;
error_register1.clk = CLK;
error_register2.clk = CLK;
error_register3.clk = CLK;
interrupts.clk = CLK;
tack_vector.clk = CLK;
state_vector.clk = CLK;

ERROR_OUT_REG  := ERROR_OUT;
LERROR_OUT_REG := LERROR_OUT;

SUBPOS0 := (subpos==0) & AVAL_REG # SUBPOS0 & !AVAL_REG;
SUBPOS1 := (subpos==1) & AVAL_REG # SUBPOS1 & !AVAL_REG;
SUBPOS2 := (subpos==2) & AVAL_REG # SUBPOS2 & !AVAL_REG;
SUBPOS3 := (subpos==3) & AVAL_REG # SUBPOS3 & !AVAL_REG;

D_CS0     = GBE  & !SEL_ERROR.fb #
            GPI0 &  SEL_ERROR.fb & SUBPOS0 #
            GPI1 &  SEL_ERROR.fb & SUBPOS1 #
            GPI2 &  SEL_ERROR.fb & SUBPOS2 #
            GPI3 &  SEL_ERROR.fb & SUBPOS3;

D_CS1     = LBE  & !SEL_ERROR.fb #
            LPI0 &  SEL_ERROR.fb & SUBPOS0 #
            LPI1 &  SEL_ERROR.fb & SUBPOS1 #
            LPI2 &  SEL_ERROR.fb & SUBPOS2 #
            LPI3 &  SEL_ERROR.fb & SUBPOS3;

D_CS2     = DBI0 &  SEL_ERROR.fb & SUBPOS0 #
            DBI1 &  SEL_ERROR.fb & SUBPOS1 #
            DBI2 &  SEL_ERROR.fb & SUBPOS2 #
            DBI3 &  SEL_ERROR.fb & SUBPOS3;

D_CS3     = 0;

GBE  := (ERROR_OUT_REG  # GBE) & !(CLR_STATUS.fb & D_CS0) & !RESET_R;
LBE  := (LERROR_OUT_REG # LBE) & !(CLR_STATUS.fb & D_CS1) & !RESET_R;

SRE0 := (D_CS0 & ce_ctrl0 # SRE0 & !ce_ctrl0) # RESET_R # !PRE0;
SRE1 := (D_CS0 & ce_ctrl1 # SRE1 & !ce_ctrl1) # RESET_R # !PRE1;
SRE2 := (D_CS0 & ce_ctrl2 # SRE2 & !ce_ctrl2) # RESET_R # !PRE2;
SRE3 := (D_CS0 & ce_ctrl3 # SRE3 & !ce_ctrl3) # RESET_R # !PRE3;

GPI0 := (ERROR_R  # GPI0) & !(clr_error0 & D_CS0) & !RESET_R;
GPI1 := (ERROR_R  # GPI1) & !(clr_error1 & D_CS0) & !RESET_R;
GPI2 := (ERROR_R  # GPI2) & !(clr_error2 & D_CS0) & !RESET_R;
GPI3 := (ERROR_R  # GPI3) & !(clr_error3 & D_CS0) & !RESET_R;

LPI0 := (LERROR_R # LPI0) & !(clr_error0 & D_CS1) & !RESET_R;
LPI1 := (LERROR_R # LPI1) & !(clr_error1 & D_CS1) & !RESET_R;
LPI2 := (LERROR_R # LPI2) & !(clr_error2 & D_CS1) & !RESET_R;
LPI3 := (LERROR_R # LPI3) & !(clr_error3 & D_CS1) & !RESET_R;

DBI0 := (DEBUG_R  # DBI0) & !(clr_error0 & D_CS2) & !RESET_R;
DBI1 := (DEBUG_R  # DBI1) & !(clr_error1 & D_CS2) & !RESET_R;
DBI2 := (DEBUG_R  # DBI2) & !(clr_error2 & D_CS2) & !RESET_R;
DBI3 := (DEBUG_R  # DBI3) & !(clr_error3 & D_CS2) & !RESET_R;

INT0 := GPI0 # LPI0 # DBI0;
INT1 := GPI1 # LPI1 # DBI1;
INT2 := GPI2 # LPI2 # DBI2;
INT3 := GPI3 # LPI3 # DBI3;

ADDR_GCS_REG := AVAL_REG & ADDR_GCS & !RESET_R;

DELAY2   :=  ARBIT_OUT & !RESET_R;
DELAY3   :=  DELAY2 & (ADDR_GCS_REG # CS_BOOT) & !RESET_R;
TACK_OUT := (DELAY3 # !ARBIT_OUT & AVAL_REG & ADDR_GCS) &
            !RESET_R;

ERROR_LED = GBE # LBE;

@dcset

state_diagram state_vector

state idle:     if RESET_R then idle else
                  if read_cmd & (offset==^h000) then status else
                  if read_cmd & (offset==^h200) then id else
                  if read_cmd & (offset==^h300) then fcn else
                  if read_cmd & (offset==^h600) then pm_id else
                  if read_cmd & (offset==^h700) then error else
                  if read_cmd                  then dummy else
                  if write_cmd & (offset==^h000) then clr_stat else
                  if write_cmd & (offset==^h100) then control else
                  if write_cmd & (offset==^h700) then clr_error else
                    idle;
```

```
state status:   if RESET_R then idle else status2;

state status2:  if RESET_R then idle else status3;

state status3:  if RESET_R then idle else
                  if BUSY_R then status3 else status4;

state status4:  if RESET_R then idle else
                  if LDE_SLV then idle else status4;

state id:       if RESET_R then idle else id2;

state id2:      if RESET_R then idle else id3;

state id3:      if RESET_R then idle else
                  if BUSY_R then id3 else id4;

state id4:      if RESET_R then idle else
                  if LDE_SLV then idle else id4;

state fcn:      if RESET_R then idle else fcn2;

state fcn2:     if RESET_R then idle else fcn3;

state fcn3:     if RESET_R then idle else
                  if BUSY_R then fcn3 else fcn4;

state fcn4:     if RESET_R then idle else
                  if LDE_SLV then idle else fcn4;

state pm_id:    if RESET_R then idle else pm_id2;

state pm_id2:   if RESET_R then idle else pm_id3;

state pm_id3:   if RESET_R then idle else
                  if BUSY_R then pm_id3 else pm_id4;

state pm_id4:   if RESET_R then idle else
                  if LDE_SLV then idle else pm_id4;

state error:    if RESET_R then idle else error2;

state error2:   if RESET_R then idle else error3;

state error3:   if RESET_R then idle else
                  if BUSY_R then error3 else error4;

state error4:   if RESET_R then idle else
                  if LDE_SLV then idle else error4;

state dummy:    if RESET_R then idle else dummy2;

state dummy2:   if RESET_R then idle else dummy3;

state dummy3:   if RESET_R then idle else
                  if BUSY_R then dummy3 else dummy4;

state dummy4:   if RESET_R then idle else
                  if LDE_SLV then idle else dummy4;

state clr_stat: goto idle;

state control:  goto idle;

state clr_error: goto idle;

test_vectors 'Set Status register'

([ RESET_R,  cmd, offset,  ERROR_OUT, LERROR_OUT,  CLK ] ->
 [ state_vector,  LBE, GBE ])

[ 1, u,     x,    0,0, c ] -> [ idle,     0,0 ];

[ 0, n,     x,    1,0, c ] -> [ idle,     0,0 ];
[ 0, n,     x,    0,1, c ] -> [ idle,     0,1 ];
[ 0, n,     x,    x,x, c ] -> [ idle,     1,1 ];

test_vectors 'Read Status register'

([ RESET_R,  cmd, offset,  BUSY_R, LDE_SLV,  CLK ] ->
 [ state_vector,  LBE, GBE,  D_CS1, D_CS0 ])

[ 0, read,  000,  x,x, c ] -> [ status,   1,1, 1,1 ];
[ 0, u,     x,    x,x, c ] -> [ status2,  1,1, 1,1 ];
[ 0, u,     x,    x,x, c ] -> [ status3,  1,1, 1,1 ];
[ 0, u,     x,    1,x, c ] -> [ status3,  1,1, 1,1 ];
[ 0, u,     x,    0,x, c ] -> [ status4,  1,1, 1,1 ];
[ 0, u,     x,    x,0, c ] -> [ status4,  1,1, 1,1 ];
[ 0, u,     x,    x,1, c ] -> [ idle,     1,1, Z,Z ];

test_vectors 'Clear Status register'

([ RESET_R,  cmd, offset,  BUSY_R, LDE_SLV,  D_CS1, D_CS0,  CLK ] ->
 [ state_vector,  LBE, GBE ])

[ 0, write, 000,  x,x, x,x, c ] -> [ clr_stat, 1,1 ];
[ 0, u,     x,    x,x, 0,1, c ] -> [ idle,     1,0 ];

[ 0, write, 000,  x,x, x,x, c ] -> [ clr_stat, 1,0 ];
[ 0, u,     x,    x,x, 1,0, c ] -> [ idle,     0,0 ];

test_vectors 'Read Status register again'

([ RESET_R,  cmd, offset,  BUSY_R, LDE_SLV,  CLK ] ->
```

```
 [ state_vector,  LBE, GBE,  D_CS1, D_CS0 ])

[ 0, read1, 000,  x,x, c ] -> [ status,   0,0, 0,0 ];
[ 0, u,     x,    x,x, c ] -> [ status2,  0,0, 0,0 ];
[ 0, u,     x,    x,x, c ] -> [ status3,  0,0, 0,0 ];
[ 0, u,     x,    0,x, c ] -> [ status4,  0,0, 0,0 ];
[ 0, u,     x,    x,1, c ] -> [ idle,     0,0, Z,Z ];

test_vectors 'Control register'

([ RESET_R,  cmd, subpos, offset,  D_CS0,  PRE0, PRE1, PRE2, PRE3,
   CLK ] ->
 [ state_vector,  SRE3,SRE2,SRE1,SRE0 ])

[ 1, u,     x,x,     x, 1,1,1,1, c ] -> [ idle,    1,1,1,1 ];
[ 0, n,     x,x,     x, 1,1,1,1, c ] -> [ idle,    1,1,1,1 ];

[ 0, write, 0,^h100, x, 1,1,1,1, c ] -> [ control, 1,1,1,1 ];
[ 0, u,     x,x,     0, 1,1,1,1, c ] -> [ idle,    1,1,1,0 ];
[ 0, write, 0,^h100, x, 1,1,1,1, c ] -> [ control, 1,1,1,0 ];
[ 0, u,     x,x,     1, 1,1,1,1, c ] -> [ idle,    1,1,1,1 ];

[ 0, write, 1,^h100, x, 1,1,1,1, c ] -> [ control, 1,1,1,1 ];
[ 0, u,     x,x,     0, 1,1,1,1, c ] -> [ idle,    1,1,0,1 ];
[ 0, write, 1,^h100, x, 1,1,1,1, c ] -> [ control, 1,1,0,1 ];
[ 0, u,     x,x,     1, 1,1,1,1, c ] -> [ idle,    1,1,1,1 ];

[ 0, write, 2,^h100, x, 1,1,1,1, c ] -> [ control, 1,1,1,1 ];
[ 0, u,     x,x,     0, 1,1,1,1, c ] -> [ idle,    1,0,1,1 ];
[ 0, write, 2,^h100, x, 1,1,1,1, c ] -> [ control, 1,0,1,1 ];
[ 0, u,     x,x,     1, 1,1,1,1, c ] -> [ idle,    1,1,1,1 ];

[ 0, write, 3,^h100, x, 1,1,1,1, c ] -> [ control, 1,1,1,1 ];
[ 0, u,     x,x,     0, 1,1,1,1, c ] -> [ idle,    0,1,1,1 ];
[ 0, write, 3,^h100, x, 1,1,1,1, c ] -> [ control, 0,1,1,1 ];
[ 0, u,     x,x,     1, 1,1,1,1, c ] -> [ idle,    1,1,1,1 ];

test_vectors 'Set, read, and clear Error register 0'

([ RESET_R,  cmd, subpos, offset,  BUSY_R, LDE_SLV,
   ERROR_R, LERROR_R, DEBUG_R,  D_CS2, D_CS1, D_CS0,  CLK ] ->
 [ state_vector,  DBI0, LPI0, GPI0,  INT0,  D_CS2, D_CS1, D_CS0 ])

[1, u,    0,x,    x,x, x,x,x, Z,Z,Z, c] -> [idle,    0,0,0, 0, Z,Z,Z];

[0, n,    0,x,    x,x, 0,0,0, Z,Z,Z, c] -> [idle,    0,0,0, 0, Z,Z,Z];
[0, n,    0,x,    x,x, 1,0,0, Z,Z,Z, c] -> [idle,    0,0,1, 0, Z,Z,Z];
[0, n,    0,x,    x,x, 0,1,0, Z,Z,Z, c] -> [idle,    0,1,1, 1, Z,Z,Z];
[0, n,    0,x,    x,x, 0,0,1, Z,Z,Z, c] -> [idle,    1,1,1, 1, Z,Z,Z];

[0, read, 0,^h700,x,x, 0,0,0, x,x,x, c] -> [error,   1,1,1, 1, 1,1,1];
[0, u,    0,x,    x,x, 0,0,0, x,x,x, c] -> [error2,  1,1,1, 1, 1,1,1];
[0, u,    0,x,    x,x, 0,0,0, x,x,x, c] -> [error3,  1,1,1, 1, 1,1,1];
[0, u,    0,x,    0,x, 0,0,0, x,x,x, c] -> [error4,  1,1,1, 1, 1,1,1];
[0, u,    0,x,    x,1, 0,0,0, Z,Z,Z, c] -> [idle,    1,1,1, 1, Z,Z,Z];

[0, write,0,^h700,0,0, 0,0,0, Z,Z,Z, c] ->[clr_error,1,1,1, 1, Z,Z,Z];
[0, u,    0,x,    0,0, 0,0,0, 1,0,0, c] -> [idle,    0,1,1, 1, x,x,x];

[0, write,0,^h700,0,0, 0,0,0, Z,Z,Z, c] ->[clr_error,0,1,1, 1, Z,Z,Z];
[0, u,    0,x,    0,0, 0,0,0, 0,1,0, c] -> [idle,    0,0,1, 1, x,x,x];

[0, write,0,^h700,0,0, 0,0,0, Z,Z,Z, c] ->[clr_error,0,0,1, 1, Z,Z,Z];
[0, u,    0,x,    0,0, 0,0,0, 0,0,1, c] -> [idle,    0,0,0, 1, x,x,x];

[0, read ,0,^h700,0,0, 0,0,0, x,x,x, c] -> [error,   0,0,0, 0, 0,0,0];
[0, u,    0,x,    0,0, 0,0,0, x,x,x, c] -> [error2,  0,0,0, 0, 0,0,0];
[0, u,    0,x,    0,0, 0,0,0, x,x,x, c] -> [error3,  0,0,0, 0, 0,0,0];
[0, u,    0,x,    0,0, 0,0,0, x,x,x, c] -> [error4,  0,0,0, 0, 0,0,0];
[0, u,    0,x,    0,1, 0,0,0, Z,Z,Z, c] -> [idle,    0,0,0, 0, Z,Z,Z];

test_vectors 'Read ID and FCN PROMs, dummy, and other commands'

([ RESET_R,  cmd, offset,  BUSY_R, LDE_SLV,  CLK ] -> state_vector )

[ 1, u,     x,     x,x, c ] -> idle;

[ 0, read,  ^h200, x,x, c ] -> id;
[ 0, n,     x,     x,x, c ] -> id2;
[ 0, n,     x,     x,x, c ] -> id3;
[ 0, n,     x,     1,x, c ] -> id3;
[ 0, n,     x,     0,x, c ] -> id4;
[ 0, n,     x,     x,0, c ] -> id4;
[ 0, n,     x,     x,1, c ] -> idle;

[ 0, read,  ^h300, x,x, c ] -> fcn;
[ 0, n,     x,     x,x, c ] -> fcn2;
[ 0, n,     x,     x,x, c ] -> fcn3;
[ 0, n,     x,     1,x, c ] -> fcn3;
[ 0, n,     x,     0,x, c ] -> fcn4;
[ 0, n,     x,     x,0, c ] -> fcn4;
[ 0, n,     x,     x,1, c ] -> idle;

[ 0, read,  ^h800, x,x, c ] -> dummy;
[ 0, n,     x,     x,x, c ] -> dummy2;
[ 0, n,     x,     x,x, c ] -> dummy3;
[ 0, n,     x,     1,x, c ] -> dummy3;
[ 0, n,     x,     0,x, c ] -> dummy4;
[ 0, n,     x,     x,0, c ] -> dummy4;
[ 0, n,     x,     x,1, c ] -> idle;

[ 0, null,       x, x,x, c ] -> idle;
[ 0, invalidate,x, x,x, c ] -> idle;
[ 0, update,     x, x,x, c ] -> idle;
[ 0, write_inv, x, x,x, c ] -> clr_stat;
```

```
[ 0, u,          x, x,x, c ] -> idle;
[ 0, reserved,   x, x,x, c ] -> idle;

test_vectors 'Target acknowledge'

([ RESET_R,  ARBIT_OUT, AVAL_REG, ADDR_GCS, CS_BOOT,  CLK ] ->
   TACK_OUT );

[ 1, x,x,x,x, c ] -> 0;

"Memory access by internal master."

[ 0, 1,1,0,0, c ] -> 0;
[ 0, 0,0,0,0, c ] -> 0;
[ 0, 0,0,0,0, c ] -> 0;

"Control space access by internal master."

[ 0, 1,1,1,0, c ] -> 0;
[ 0, 0,0,0,0, c ] -> 0;
[ 0, 0,0,0,0, c ] -> 1;

"Boot access by internal master."

[ 0, 1,0,0,0, c ] -> 0;
[ 0, 0,0,0,1, c ] -> 0;
[ 0, 0,0,0,0, c ] -> 1;

"Memory access by external master."

[ 0, 0,0,0,0, c ] -> 0;
[ 0, 0,1,0,0, c ] -> 0;
[ 0, 0,0,0,0, c ] -> 0;
[ 0, 0,0,0,0, c ] -> 0;

"Control space access by external master."

[ 0, 0,0,0,0, c ] -> 0;
[ 0, 0,0,0,0, c ] -> 0;
[ 0, 0,0,0,0, c ] -> 0;
[ 0, 0,1,1,0, c ] -> 1;

end cspace;
```

```
module boot

title
'Boot control for CPU301/302.
Dansk Data Elektronik.
Ole Moller.
September 9, 1993.';
"January 14, 1994. Big endian.
"January 17, 1994. Boot signals not impeccable.
"February 9, 1994. Boot signals qualified by Busy signals.
"October  6, 1994. Flash PROM requires only 5 wait states.
"April 12,   1996. ARBIT_OUT inp. is delayed 1 cloc,
"                  byte_state test is removed

c31091 device 'mach110a';

"Inputs:

CLK             pin 13;
RESET_R         pin 35;
ARBIT_OUT       pin 10;
LDE_SLV         pin 11;
BUSY_R          pin 14;

GCR6            pin 32;
GCR7            pin 33;

BOOT0           pin  2;
BOOT1           pin  3;
BOOT2           pin  4;
BOOT3           pin  5;

!BUSY_OUT0      pin  6;
!BUSY_OUT1      pin  7;
!BUSY_OUT2      pin 17;
!BUSY_OUT3      pin 20;

"Register outputs:

BOOT_REG        pin    istype 'reg, buffer';

!CS_BOOT        pin  8 istype 'reg, invert';
!INTV_BOOT      pin 19 istype 'reg, invert';
DVAL_BOOT       pin 21 istype 'reg, buffer';
BA0             pin 15 istype 'reg, buffer';
BA1             pin 16 istype 'reg, buffer';
BA2             pin 18 istype 'reg, buffer';
N0              pin    istype 'reg, buffer';
N1              pin    istype 'reg, buffer';
N2              pin    istype 'reg, buffer';

"Combinational outputs:

!CE_R0          pin 43 istype 'com, invert';
!CE_R1          pin 42 istype 'com, invert';
!CE_R2          pin 41 istype 'com, invert';
!CE_R3          pin 40 istype 'com, invert';
!CE_R4          pin 39 istype 'com, invert';
!CE_R5          pin 38 istype 'com, invert';
!CE_R6          pin 37 istype 'com, invert';
!CE_R7          pin 36 istype 'com, invert';

AMDMACH property 'GROUP B N0 N1 N2';
AMDMACH property 'GROUP B BOOT_REG';

"Definitions:

c               = .C.;
x               = .X.;

read            = !GCR7 & !GCR6;

CE_R            = [ CE_R7..CE_R0 ];

byte_state      = [ CS_BOOT, INTV_BOOT, DVAL_BOOT, BA2..BA0 ];

idle            = ^b000000;
byte0           = ^b110000;
byte1           = ^b110001;
byte2           = ^b110010;
byte3           = ^b110011;
byte4           = ^b110100;
byte5           = ^b110101;
byte6           = ^b110110;
byte7           = ^b110111;
wait            = ^b010000;
finito          = ^b011000;

timing_state    = [ N2..N0 ];

step0           = ^b000;
step1           = ^b001;
step2           = ^b010;
step3           = ^b011;
step4           = ^b100;

equations

BOOT_REG.clk = CLK;
byte_state.clk = CLK;
timing_state.clk = CLK;

BOOT_REG        := BOOT0 & BUSY_OUT0 # BOOT1 & BUSY_OUT1 #
                   BOOT2 & BUSY_OUT2 # BOOT3 & BUSY_OUT3;
```

```
CE_R0           = (byte_state==byte0);
CE_R1           = (byte_state==byte1);
CE_R2           = (byte_state==byte2);
CE_R3           = (byte_state==byte3);
CE_R4           = (byte_state==byte4);
CE_R5           = (byte_state==byte5);
CE_R6           = (byte_state==byte6);
CE_R7           = (byte_state==byte7);

@dcset

state_diagram byte_state

state idle:     if RESET_R then idle else
                  if ARBIT_OUT & BOOT_REG & read then byte0
                  else idle;

state byte0:    if RESET_R then idle else
                  if (timing_state==step4) then byte1 else byte0;

state byte1:    if RESET_R then idle else
                  if (timing_state==step4) then byte2 else byte1;

state byte2:    if RESET_R then idle else
                  if (timing_state==step4) then byte3 else byte2;

state byte3:    if RESET_R then idle else
                  if (timing_state==step4) then byte4 else byte3;

state byte4:    if RESET_R then idle else
                  if (timing_state==step4) then byte5 else byte4;

state byte5:    if RESET_R then idle else
                  if (timing_state==step4) then byte6 else byte5;

state byte6:    if RESET_R then idle else
                  if (timing_state==step4) then byte7 else byte6;

state byte7:    if RESET_R then idle else
                  if !(timing_state==step4) then byte7 else
                    if BUSY_R then wait else finito;

state wait:     if RESET_R then idle else
                  if BUSY_R then wait else finito;

state finito:   if RESET_R then idle else
                  if LDE_SLV then idle else finito;


state_diagram timing_state

state step0:    if RESET_R then step0 else
                  if (byte_state==byte0) #
                     (byte_state==byte1) #
                     (byte_state==byte2) #
                     (byte_state==byte3) #
                     (byte_state==byte4) #
                     (byte_state==byte5) #
                     (byte_state==byte6) #
                     (byte_state==byte7) then step1 else step0;

state step1:    if RESET_R then step0 else step2;

state step2:    if RESET_R then step0 else step3;

state step3:    if RESET_R then step0 else step4;

state step4:    goto step0;


test_vectors

([ RESET_R,  ARBIT_OUT,  BOOT0, BOOT1, BOOT2, BOOT3,
   BUSY_OUT0, BUSY_OUT1, BUSY_OUT2, BUSY_OUT3,
   GCR7, GCR6,  BUSY_R, LDE_SLV,  CLK ] ->
 [ byte_state, timing_state ])

[ 1, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 0, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,0,0, 1,1,1,1, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 1,0,0,0, 0,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,1,0,0, 0,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,1,0, 0,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,0,1, 0,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 1,0,0,0, 1,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 1,1, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,1,0,0, 0,1,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 1,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,1,0, 0,0,1,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,1, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,0,1, 0,0,0,1, x,x, x,x, c ] -> [ idle,  step0 ];
```

```
CE_R0           = (byte_state==byte0);
CE_R1           = (byte_state==byte1);
CE_R2           = (byte_state==byte2);
CE_R3           = (byte_state==byte3);
CE_R4           = (byte_state==byte4);
CE_R5           = (byte_state==byte5);
CE_R6           = (byte_state==byte6);
CE_R7           = (byte_state==byte7);

@dcset

state_diagram byte_state

state idle:     if RESET_R then idle else
                  if ARBIT_OUT & BOOT_REG & read then byte0
                  else idle;

state byte0:    if RESET_R then idle else
                  if (timing_state==step4) then byte1 else byte0;

state byte1:    if RESET_R then idle else
                  if (timing_state==step4) then byte2 else byte1;

state byte2:    if RESET_R then idle else
                  if (timing_state==step4) then byte3 else byte2;

state byte3:    if RESET_R then idle else
                  if (timing_state==step4) then byte4 else byte3;

state byte4:    if RESET_R then idle else
                  if (timing_state==step4) then byte5 else byte4;

state byte5:    if RESET_R then idle else
                  if (timing_state==step4) then byte6 else byte5;

state byte6:    if RESET_R then idle else
                  if (timing_state==step4) then byte7 else byte6;

state byte7:    if RESET_R then idle else
                  if !(timing_state==step4) then byte7 else
                    if BUSY_R then wait else finito;

state wait:     if RESET_R then idle else
                  if BUSY_R then wait else finito;

state finito:   if RESET_R then idle else
                  if LDE_SLV then idle else finito;


state_diagram timing_state

state step0:    if RESET_R then step0 else
                  if (byte_state==byte0) #
                     (byte_state==byte1) #
                     (byte_state==byte2) #
                     (byte_state==byte3) #
                     (byte_state==byte4) #
                     (byte_state==byte5) #
                     (byte_state==byte6) #
                     (byte_state==byte7) then step1 else step0;

state step1:    if RESET_R then step0 else step2;

state step2:    if RESET_R then step0 else step3;

state step3:    if RESET_R then step0 else step4;

state step4:    goto step0;


test_vectors

([ RESET_R,  ARBIT_OUT,  BOOT0, BOOT1, BOOT2, BOOT3,
   BUSY_OUT0, BUSY_OUT1, BUSY_OUT2, BUSY_OUT3,
   GCR7, GCR6,  BUSY_R, LDE_SLV,  CLK ] ->
 [ byte_state, timing_state ])

[ 1, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 0, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,0,0, 1,1,1,1, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 1,0,0,0, 0,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,1,0,0, 0,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,1,0, 0,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,0,1, 0,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 1,0,0,0, 1,0,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 1,1, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,1,0,0, 0,1,0,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 1,0, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,1,0, 0,0,1,0, x,x, x,x, c ] -> [ idle,  step0 ];
[ 0, 1, x,x,x,x, x,x,x,x, 0,1, x,x, c ] -> [ idle,  step0 ];

[ 0, 0, 0,0,0,1, 0,0,0,1, x,x, x,x, c ] -> [ idle,  step0 ];
```

```
[ 0, 1, x,x,x,x, x,x,x,x, 0,0, x,x, c ] -> [ byte0, step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte0, step1 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte0, step2 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte0, step3 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte0, step4 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte1, step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte1, step1 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte1, step2 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte1, step3 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte1, step4 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte2, step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte2, step1 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte2, step2 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte2, step3 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte2, step4 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte3, step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte3, step1 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte3, step2 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte3, step3 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte3, step4 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte4, step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte4, step1 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte4, step2 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte4, step3 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte4, step4 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte5, step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte5, step1 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte5, step2 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte5, step3 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte5, step4 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte6, step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte6, step1 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte6, step2 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte6, step3 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte6, step4 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte7, step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte7, step1 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte7, step2 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte7, step3 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,x, c ] -> [ byte7, step4 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, 1,x, c ] -> [ wait,  step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, 1,x, c ] -> [ wait,  step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, 0,x, c ] -> [ finito,step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,0, c ] -> [ finito,step0 ];
[ 0, x, x,x,x,x, x,x,x,x, x,x, x,1, c ] -> [ idle,  step0 ];

end boot;
```

```
module block

title
'Dummy block generator and last data element counter for CPU301/302.
Dansk Data Elektronik.
Ole Moller.
September 9, 1993.';
"AAJ/96-04-14: ARBIT_OUT inp delayed 1 clock

c31101 device 'mach110a';

"Inputs:

CLK             pin 13;
RESET_R         pin 35;

AVAL_REG        pin 10;
DVAL_REG        pin 11;
DVAL_SLV        pin 32;
ARBIT_OUT       pin 33;

CR0             pin  2;
CR1             pin  3;
CR3             pin  4;
CR4             pin  5;
CR5             pin  6;
CR6             pin  7;
CR7             pin  8;

TACK_R          pin 28;
BUSY_R          pin 29;
LDE_SLV_IN      pin 30;
INTV_R          pin 31;

"Register outputs:

LDE_SLV         pin 18 istype 'reg, buffer';
!N0             pin    istype 'reg, invert';
!N1             pin    istype 'reg, invert';
!N2             pin    istype 'reg, invert';
!N3             pin    istype 'reg, invert';
EMPTY_CYCLE     pin    istype 'reg, buffer';

BLOCK           pin 42 istype 'reg, buffer';
DVAL_BLOCK      pin 41 istype 'reg, buffer';
!M0             pin    istype 'reg, invert';
!M1             pin    istype 'reg, invert';
!M2             pin    istype 'reg, invert';

AMDMACH property 'GROUP A N0 N1 N2 N3 EMPTY_CYCLE';
AMDMACH property 'GROUP B M0 M1 M2';

"Definitions:

c               = .C.;
x               = .X.;

cmd             = [ AVAL_REG, DVAL_REG, CR7..CR5 ];

none            = [ 0, 0, x,x,x ];
read            = [ 1, 0, 0,0,0 ];
read1           = [ 1, 0, 0,0,1 ];
write           = [ 1, 0, 0,1,0 ];
null            = [ 1, 0, 0,1,1 ];
invalidate      = [ 1, 0, 1,0,0 ];
update          = [ 1, 0, 1,0,1 ];
write_inv       = [ 1, 0, 1,1,0 ];
reserved        = [ 1, 0, 1,1,1 ];
nlde_id         = [ 0, 1, 1,x,x ];
lde_id          = [ 0, 1, 0,x,x ];
xc              = [ x, x, x,x,x ];

attr            = [ CR4..CR3 ];

c_block         = 0;
xc_block        = 1;
nc_block        = 2;
simple          = 3;
xa              = [ x,x ];

format          = [ CR1..CR0 ];

byte            = 0;
word4           = 0;
word8           = 1;
word16          = 2;
word32          = 3;
xf              = [ x,x ];

read_cmd        = ((cmd==read) # (cmd==read1));

counter_state   = [ LDE_SLV, N3, N2, N1, N0 ];

dw15            = ^b01111;
dw14            = ^b01110;
dw13            = ^b01101;
dw12            = ^b01100;
dw11            = ^b01011;
dw10            = ^b01010;
dw9             = ^b01001;
dw8             = ^b01000;
dw7             = ^b00111;
dw6             = ^b00110;
dw5             = ^b00101;
dw4             = ^b00100;
```

```
dw3             = ^b00011;
dw2             = ^b00010;
dw1             = ^b00001;
dw0             = ^b10000;

block_state     = [ BLOCK, DVAL_BLOCK, M2, M1, M0 ];

idle            = ^b00000;
step1           = ^b00001;
step2           = ^b00010;
step3           = ^b00011;
step4           = ^b00100;
step5           = ^b00101;
step6           = ^b00110;
bingo           = ^b10000;
d_block         = ^b11000;

equations

counter_state.clk = CLK;
block_state.clk = CLK;
EMPTY_CYCLE.clk = CLK;

EMPTY_CYCLE     := cmd==lde_id;

@dcset

state_diagram counter_state

state dw15:     if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw14 else dw15;

state dw14:     if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw13 else dw14;

state dw13:     if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw12 else dw13;

state dw12:     if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw11 else dw12;

state dw11:     if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw10 else dw11;

state dw10:     if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw9 else dw10;

state dw9:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw8 else dw9;

state dw8:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw7 else dw8;

state dw7:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw6 else dw7;

state dw6:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw5 else dw6;

state dw5:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw4 else dw5;

state dw4:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw3 else dw4;

state dw3:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw2 else dw3;

state dw2:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw1 else dw2;

state dw1:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if DVAL_SLV then dw0 else dw1;

state dw0:      if RESET_R # EMPTY_CYCLE then dw0 else
                  if !read_cmd then dw0 else
                    if (attr==simple) then dw0 else
                      case
                        format==word4  : dw1;
                        format==word8  : dw3;
                        format==word16 : dw7;
                        format==word32 : dw15;
                      endcase;


state_diagram block_state

state idle:     if RESET_R # INTV_R then idle else
                  if ARBIT_OUT & read_cmd then step2 else idle;

state step2:    if RESET_R # INTV_R then idle else step3;

state step3:    if RESET_R # INTV_R then idle else step4;

state step4:    if RESET_R # INTV_R then idle else step5;

state step5:    if RESET_R # INTV_R then idle else step6;

state step6:    if RESET_R # INTV_R then idle else
                  if TACK_R then idle else bingo;

state bingo:    if RESET_R # INTV_R then idle else
                  if BUSY_R then bingo else d_block;

state d_block:  if RESET_R # INTV_R then idle else
```

```
                  if LDE_SLV_IN then idle else d_block;

test_vectors 'Last data element counter'

([ RESET_R,  cmd, attr, format,  DVAL_SLV,  CLK ] -> counter_state )

[ 1, xc,    xa,       xf,      x, c ] -> dw0;
[ 0, none,  xa,       xf,      x, c ] -> dw0;
[ 0, write, xa,       xf,      x, c ] -> dw0;

"Different read formats.

[ 0, read,  simple,  byte,    x, c ] -> dw0;

[ 0, read,  c_block, word4,   x, c ] -> dw1;
[ 0, xc,    xa,       xf,      1, c ] -> dw0;

[ 0, read,  c_block, word8,   x, c ] -> dw3;
[ 0, xc,    xa,       xf,      1, c ] -> dw2;
[ 0, xc,    xa,       xf,      1, c ] -> dw1;
[ 0, xc,    xa,       xf,      1, c ] -> dw0;

[ 0, read,  c_block, word16,  x, c ] -> dw7;
[ 0, xc,    xa,       xf,      1, c ] -> dw6;
[ 0, xc,    xa,       xf,      1, c ] -> dw5;
[ 0, xc,    xa,       xf,      1, c ] -> dw4;
[ 0, xc,    xa,       xf,      1, c ] -> dw3;
[ 0, xc,    xa,       xf,      1, c ] -> dw2;
[ 0, xc,    xa,       xf,      1, c ] -> dw1;
[ 0, xc,    xa,       xf,      1, c ] -> dw0;

[ 0, read,  c_block, word32,  x, c ] -> dw15;
[ 0, xc,    xa,       xf,      1, c ] -> dw14;
[ 0, xc,    xa,       xf,      1, c ] -> dw13;
[ 0, xc,    xa,       xf,      1, c ] -> dw12;
[ 0, xc,    xa,       xf,      1, c ] -> dw11;
[ 0, xc,    xa,       xf,      1, c ] -> dw10;
[ 0, xc,    xa,       xf,      1, c ] -> dw9;
[ 0, xc,    xa,       xf,      1, c ] -> dw8;
[ 0, xc,    xa,       xf,      1, c ] -> dw7;
[ 0, xc,    xa,       xf,      1, c ] -> dw6;
[ 0, xc,    xa,       xf,      1, c ] -> dw5;
[ 0, xc,    xa,       xf,      1, c ] -> dw4;
[ 0, xc,    xa,       xf,      1, c ] -> dw3;
[ 0, xc,    xa,       xf,      1, c ] -> dw2;
[ 0, xc,    xa,       xf,      1, c ] -> dw1;
[ 0, xc,    xa,       xf,      1, c ] -> dw0;

"Different read formats with stalls.

[ 0, read,  c_block, word32,  x, c ] -> dw15;
[ 0, xc,    xa,       xf,      0, c ] -> dw15;
[ 0, xc,    xa,       xf,      1, c ] -> dw14;
[ 0, xc,    xa,       xf,      0, c ] -> dw14;
[ 0, xc,    xa,       xf,      1, c ] -> dw13;
[ 0, xc,    xa,       xf,      0, c ] -> dw13;
[ 0, xc,    xa,       xf,      1, c ] -> dw12;
[ 0, xc,    xa,       xf,      0, c ] -> dw12;
[ 0, xc,    xa,       xf,      1, c ] -> dw11;
[ 0, xc,    xa,       xf,      0, c ] -> dw11;
[ 0, xc,    xa,       xf,      1, c ] -> dw10;
[ 0, xc,    xa,       xf,      0, c ] -> dw10;
[ 0, xc,    xa,       xf,      1, c ] -> dw9;
[ 0, xc,    xa,       xf,      0, c ] -> dw9;
[ 0, xc,    xa,       xf,      1, c ] -> dw8;
[ 0, xc,    xa,       xf,      0, c ] -> dw8;
[ 0, xc,    xa,       xf,      1, c ] -> dw7;
[ 0, xc,    xa,       xf,      0, c ] -> dw7;
[ 0, xc,    xa,       xf,      1, c ] -> dw6;
[ 0, xc,    xa,       xf,      0, c ] -> dw6;
[ 0, xc,    xa,       xf,      1, c ] -> dw5;
[ 0, xc,    xa,       xf,      0, c ] -> dw5;
[ 0, xc,    xa,       xf,      1, c ] -> dw4;
[ 0, xc,    xa,       xf,      0, c ] -> dw4;
[ 0, xc,    xa,       xf,      1, c ] -> dw3;
[ 0, xc,    xa,       xf,      0, c ] -> dw3;
[ 0, xc,    xa,       xf,      1, c ] -> dw2;
[ 0, xc,    xa,       xf,      0, c ] -> dw2;
[ 0, xc,    xa,       xf,      1, c ] -> dw1;
[ 0, xc,    xa,       xf,      0, c ] -> dw1;
[ 0, xc,    xa,       xf,      1, c ] -> dw0;

"Early termination by last data element.

[ 0, read,  c_block, word4,   x, c ] -> dw1;
[ 0, lde_id,xa,       xf,      0, c ] -> dw1;
[ 0, xc,    xa,       xf,      0, c ] -> dw0;

[ 0, read,  c_block, word8,   x, c ] -> dw3;
[ 0, lde_id,xa,       xf,      0, c ] -> dw3;
[ 0, xc,    xa,       xf,      0, c ] -> dw0;

[ 0, read,  c_block, word16,  x, c ] -> dw7;
[ 0, lde_id,xa,       xf,      0, c ] -> dw7;
[ 0, xc,    xa,       xf,      0, c ] -> dw0;

[ 0, read,  c_block, word32,  x, c ] -> dw15;
[ 0, lde_id,xa,       xf,      0, c ] -> dw15;
[ 0, xc,    xa,       xf,      0, c ] -> dw0;


test_vectors 'Dummy block generator'

([ RESET_R,  cmd,  ARBIT_OUT,  TACK_R, BUSY_R, LDE_SLV_IN,  INTV_R,
```

```
   CLK ] -> block_state )

[ 1, xc,     x, x,x,x, 0, c ] -> idle;
[ 0, none,   0, x,x,x, 0, c ] -> idle;

"Write.

[ 0, write, 1, x,x,x, 0, c ] -> idle;

"Read with target acknowledge.

[ 0, read,  1, x,x,x, 0, c ] -> step2;
[ 0, xc,    x, x,x,x, 0, c ] -> step3;
[ 0, xc,    x, x,x,x, 0, c ] -> step4;
[ 0, xc,    x, x,x,x, 0, c ] -> step5;
[ 0, xc,    x, x,x,x, 0, c ] -> step6;
[ 0, xc,    x, 1,x,x, 0, c ] -> idle;

"Read with cache intervention.

[ 0, read,  1, x,x,x, 0, c ] -> step2;
[ 0, xc,    x, x,x,x, 0, c ] -> step3;
[ 0, xc,    x, x,x,x, 0, c ] -> step4;
[ 0, xc,    x, x,x,x, 0, c ] -> step5;
[ 0, xc,    x, x,x,x, 0, c ] -> step6;
[ 0, xc,    1, 0,x,x, 1, c ] -> idle;

"Read with missing target acknowledge and
"no cache intervention produces dummy block.

[ 0, read,  1, x,x,x, 0, c ] -> step2;
[ 0, xc,    x, x,x,x, 0, c ] -> step3;
[ 0, xc,    x, x,x,x, 0, c ] -> step4;
[ 0, xc,    x, x,x,x, 0, c ] -> step5;
[ 0, xc,    x, x,x,x, 0, c ] -> step6;
[ 0, xc,    x, 0,x,x, 0, c ] -> bingo;
[ 0, xc,    x, x,1,x, 0, c ] -> bingo;
[ 0, xc,    x, x,0,x, 0, c ] -> d_block;
[ 0, xc,    x, x,x,0, 0, c ] -> d_block;
[ 0, xc,    x, x,x,1, 0, c ] -> idle;

end block;
```

QM 5012-1 Vareidentifikation

1. **Dato:** 1996-03-14

2. **Rekvirent:** aaj

3. **Varebetegnelse:** MACH210A-10JC

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

5. **Råvarenummer:**

6. **Varekategori:** Mærkevare

   Dokumentreference:

7. **Beskrivelse:** MACH210A-10 44 pin PLCC

8. **Leverandører:**

   Leverandør 1: arrow-exatec

   Fabrikat: AMD Anslået pris:
   Bestillingsnummer: MACH210A-10JC

   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:** 1996-03-14

2. **Rekvirent:** aaj

3. **Varebetegnelse:** PALCE22V10H-7JC

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

5. **Råvarenummer:**

6. **Varekategori:** Mærkevare

   Dokumentreference:

7. **Beskrivelse:** CMOS EE PAL22V10 half power 7 ns
   28-pin PLCC

8. **Leverandører:**

   Leverandør 1: arrow-exatec

   Fabrikat: AMD Anslået pris:
   Bestillingsnummer: PALCE22V10H-7JC/5

   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:**

Supermax Technical Data Sheet

Module:	CPU300 and CPU302
Data sheet no:	39
Revision no:	1
Date:	95-10-17

1. General description

The CPU300 and CPU302 are CPU base modules for the Supermax Enterprise Server series. They both interface 1-4 processor modules to the global bus by means of four agents (gate arrays), four bus snoopers, a control space, a boot PROM, and a bus interface.

The CPU300 and CPU302 support processor modules with 1 MB and 4 MB secondary cache, respectively. Both big and little endian byte ordering are supported.

2. Specifications

Bus clock frequency	33.3 MHz
Bus data width	64 bits
Bus bandwidth, burst	267 MB/s
- - , read (32-word)	178 MB/s
- - , write (32-word)	213 MB/s
Max. sec. cache size, CPU300	1 MB
- - - - , CPU302	4 MB
Secondary cache line size	128 bytes

3. Physical dimensions

Standard module form factor.

Length	450 mm
Width	350 mm
Height	30 mm
Weight	1.1 kg

4. Power requirements

Typical values excluding requirements of processor modules.

3.3V	0 A
5.0V	5 A
12.0V	0 A

5. Environment

See the Supermax Technical Data Sheet no. 61.

6. Interfaces

The module interfaces to the global system bus using the signals below.

/AD(0-63)
/ADP(0-7)
/CMD(0-7)
/CMDP
/BRQ(0-15)
/AVAL
/DVAL
/ARBIT
/BUSY
/SHRD
/INTV
/TACK
/ERROR
/RSV
/RESET
/DEBUG
BCLK
/BCLK
ID(0-3)
BIG

All signals above are BTL, except BCLK and /BCLK, which are differential positive ECL, and ID(0-3) and BIG, which are TTL. See the data sheet of the backplane for the pin assignment.

Each of the four processor modules (n) interfaces to the base module by means of four connectors Xn0,Xn1,Xn2,Xn3.

Pin	Xn0	Xn1	Xn2	Xn3
A1	SADn_0	SADn_25	SADn_50	SCMDn_3
A2	SADn_1	SADn_26	SADn_51	SCMDn_4
A3	SADn_2	SADn_27	SADn_52	SCMDn_5
A4	SADn_3	SADn_28	SADn_53	SCMDn_6
A5	SADn_4	SADn_29	SADn_54	SCMDn_7
A6	SADn_5	SADn_30	SADn_55	SCMDn_8
A7	SADn_6	SADn_31	SADn_56	SCMDPn
A8	SADn_7	SADn_32	SADn_57	/VALINnn
A9	SADn_8	SADn_33	SADn_58	/VALOUTn
A10	SADn_9	SADn_34	SADn_59	/RELEASEn
A11	SADn_10	SADn_35	SADn_60	/WRRDYn
A12	SADn_11	SADn_36	SADn_61	/RDDRYn
A13	SADn_12	SADn_37	SADn_62	/IVDACKn
A14	SADn_13	SADn_38	SADn_63	/IVDERRn
A15	SADn_14	SADn_39	SADPn_0	/EXTRQn
A16	SADn_15	SADn_40	SADPn_1	TCLKn_0
A17	SADn_16	SADn_41	SADPn_2	TCLKn_1
A18	SADn_17	SADn_42	SADPn_3	RCLKn_0
A19	SADn_18	SADn_43	SADPn_4	RCLKn_1
A20	SADn_19	SADn_44	SADPn_5	/INTn
A21	SADn_20	SADn_45	SADPn_6	/NMIn
A22	SADn_21	SADn_46	SADPn_7	n.c.
A23	SADn_22	SADn_47	SCMDn_0	SREn
A24	SADn_23	SADn_48	SCMDn_1	C2MS
A25	SADn_24	SADn_49	SCMDn_2	C262MS
B1	5 V	GND	3.3 V	GND
B2	5 V	GND	3.3 V	GND
B3	5 V	GND	3.3 V	GND
B4	5 V	GND	3.3 V	GND
B5	5 V	GND	3.3 V	GND
B6	5 V	GND	3.3 V	GND
B7	5 V	GND	3.3 V	GND
B8	5 V	GND	3.3 V	GND
B9	5 V	GND	3.3 V	GND
B10	5 V	GND	3.3 V	GND
B11	5 V	GND	3.3 V	GND
B12	5 V	GND	3.3 V	GND
B13	5 V	GND	3.3 V	GND
B14	5 V	GND	3.3 V	GND
B15	5 V	GND	3.3 V	GND
B16	5 V	GND	3.3 V	GND
B17	5 V	GND	3.3 V	GND
B18	5 V	GND	3.3 V	GND
B19	5 V	GND	3.3 V	GND
B20	5 V	GND	3.3 V	PM_IDn_0
B21	5 V	GND	3.3 V	PM_IDn_1
B22	5 V	GND	3.3 V	PM_IDn_2
B23	5 V	GND	3.3 V	PM_IDn_3
B24	5 V	GND	3.3 V	/PREn
B25	5 V	GND	3.3 V	BIG

7. Installation

The module may be placed in any position within the crate. Slide the module in carefully such that the connectors of the module and the backplane meet. Then depress the handles firmly such that the connectors engage fully and the front panel travels the full distance. Finally, secure the module with the pozi-drive screws in the handles. Removal takes place in reverse order of fitting.

8. Transport

Precautions against electrostatic shock and mechanical damage must be taken when packing the module for transportation.