Dansk Data Elektronik A/S
8509 DMA & INT & BANKSWITCH

26 NOVEMBER 1982

for the

SPC-1 MICROCOMPUTER SYSTEM

Auther: Thue Winther-Jørgensen

PREFACE.

The 8509 is a combined module, a combination which consists of three parts, one DMA (Direct Memory Acces) part, one priority interrupt part, and one register file for new bank switch. The three parts, since they are more or less independent, will be described in seperate sections. Section one will deal with the DMA part, section two with the bank register array, and section three will deal with the priority interrupt part.

1.0. GENERAL DESCRIPTION DMA PART.

The 8509 module uses three AMD9517 DMA controllers, which are described in detail in appendix 1. Appendix 2 which is the logic schematic of the module, need not consern the programmer, and is consequently left out in most manuals.

The 9517 DMA controller has four DMA channels, two of these chips are used to service the eight I/O dma channels. Two of the channels in the third 9517 is used with a on board register to perform memory move (channel zero to read from memory to the register, and channel one to read from the register to the memory).
Hardware arbitration give the two first 9517 chips a rotating priority and the third a low priority.

Addressing.

The dma part uses 48 (decimal) addresses (16 for eatch dma chip), ADR(3:0) selects different registers or control functions in the DMA controller chip according to appendix one, the base addres for the dma part is normaly 20 (hex).

1.2 DMA CONTROL SIGNALS.

DMAREQ(7:0)

The signals are open collecter active low. The 8509 module has the pull-up resistors for the signals and converts them to active high before they are connected to the 9517 dma controller.

INDDMA, OUTDMA.

These signals, that are open collector active low with pull-up, control reading and writing respectively in a DMA io unit during a DMA cycle.

$\overline{\text{EOPIN}}$, $\overline{\text{EOPOUT}}$.

These signals are input and output to the EOP signal from the DMA controller. the EOP signal is descriped in appendix one. The signals are not inverted on the 8509 module. Eopout, is open collector active low with pull-up resistor .

Programming considerations.

Se appendix one page six: command register.
The controller should be initialized for normal timing (bit 3) and late write (bit 5).
Dreq should be active high (bit 6) and dack should be active low (bit 7).

2.0. GENERAL DESCRIPTION FOR BANK SWITCH.

The 8509 module contains the bank select register array, witch are used in bank switch implementation 2, that is in systems in which the memory address bus is expanded with four bits from ADR(15:0) to ADR(19:0).

The register array consist of 16 registers of four bits eatch. The selected register drives the address bus extension ADR(19:16).

2.1. WRITING IN THE REGISTER ARRAY.

The address of the register array is x´FB´. The cpu writes in the register array by executing an out instruction with the address FB The four least significant bits in the data word, D(3:0), are written in the register selected by D(7:4).

The program:

```
        MVI     A,XY
        OUT     0FB
```

writes the hexadecimal value Y in the register with the hexadecimal number x.

2.2. READING IN THE REGISTER ARRAY.

Issue the register number at ioaddress x´FD read the contents at address x´FB (note that the two operations must be done indevisibel, that is with a disabled interrupt system).

2.3 DRIVING THE ADDRESS LINES ADR(19:16).

When the cpu is running that is when the cpu has not issued a HOLD acknowledge register number zero, R(0), drives ADR(19:16). When the cpu has issued a HOLD ACKNOWLEDGE and the dma part has issued DMAACK(N) register R(N+1) drives the address lines ADR(19:16).

that is

R(0)	cpu running	
R(1)	DMA chip one (address 20) active on channel 0	DMAACK(0) active
R(2)	DMA chip one (address 20) active on channel 1	DMAACK(1) active
R(3)	DMA chip one (address 20) active on channel 2	DMAACK(2) active
R(4)	DMA chip one (address 20) active on channel 3	DMAACK(3) active
R(5)	DMA chip two (address 30) active on channel 0	DMAACK(4) active
R(6)	DMA chip two (address 30) active on channel 1	DMAACK(5) active
R(7)	DMA chip two (address 30) active on channel 2	DMAACK(6) active
R(8)	DMA chip two (address 30) active on channel 3	DMAACK(7) active
R(9)	DMA chip tree(address 40) active on channel 0	
R(10)	DMA chip tree(address 40) active on channel 1	
R(11)	CURRENTLY NOT USED	
R(12)	CURRENTLY NOT USED	
R(13)	CURRENTLY NOT USED	
R(14)	CURRENTLY NOT USED	
R(15)	CURRENTLY NOT USED	

2.4. PROGRAMMING.

The cpu enables a given bank by writing the bank address in R(0). Before enabling DMA channel N the program must initialize R(n+1) with the bank address of the bank to/from which data is to be transferred.

The content of the registers are unknown after power up, and are not affected by reset.

3.0. GENERAL DESCRIPTION INTERRUPT PART.

The interrupt priority part of the 8509 module is used in configurations where the I/O modules must be served by means of interrupts. The 8509 module is able to deal with incoming interrupt request lines IR(7:0) at eight priority levels. The module sends an interrupt request to the cpu module (with the bus signal INT) if at least one enabled interrupt request line is activated. The cpu module responds to this by executing an interrupt acknowledge michine cycle. In this cycle the 8509 generates a restart instruction (RST 0,1,...,7) corresponding to the interrupt request with highest priority. When the cpu responds to the interrupt request, the internal interrupt enable flip-flop of the 8085 is cleared to prevent more interrupts from beeing served. Acknowledgement of the interrupting I/O unit and reenabling of the interrupt system is under program control.

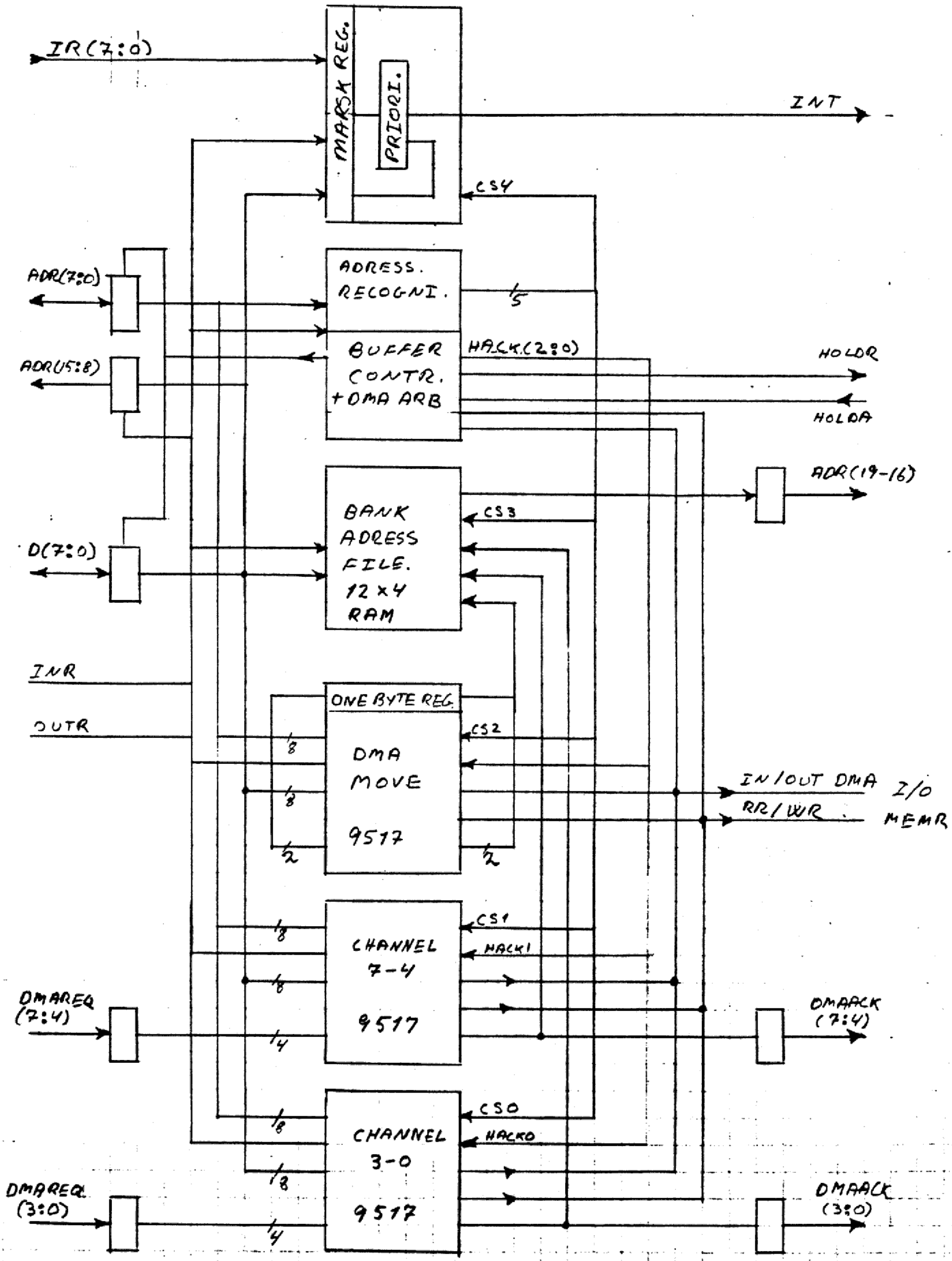
The 8509 module contains an interrupt mask register for selective enabling/disabling of the different priority levels.

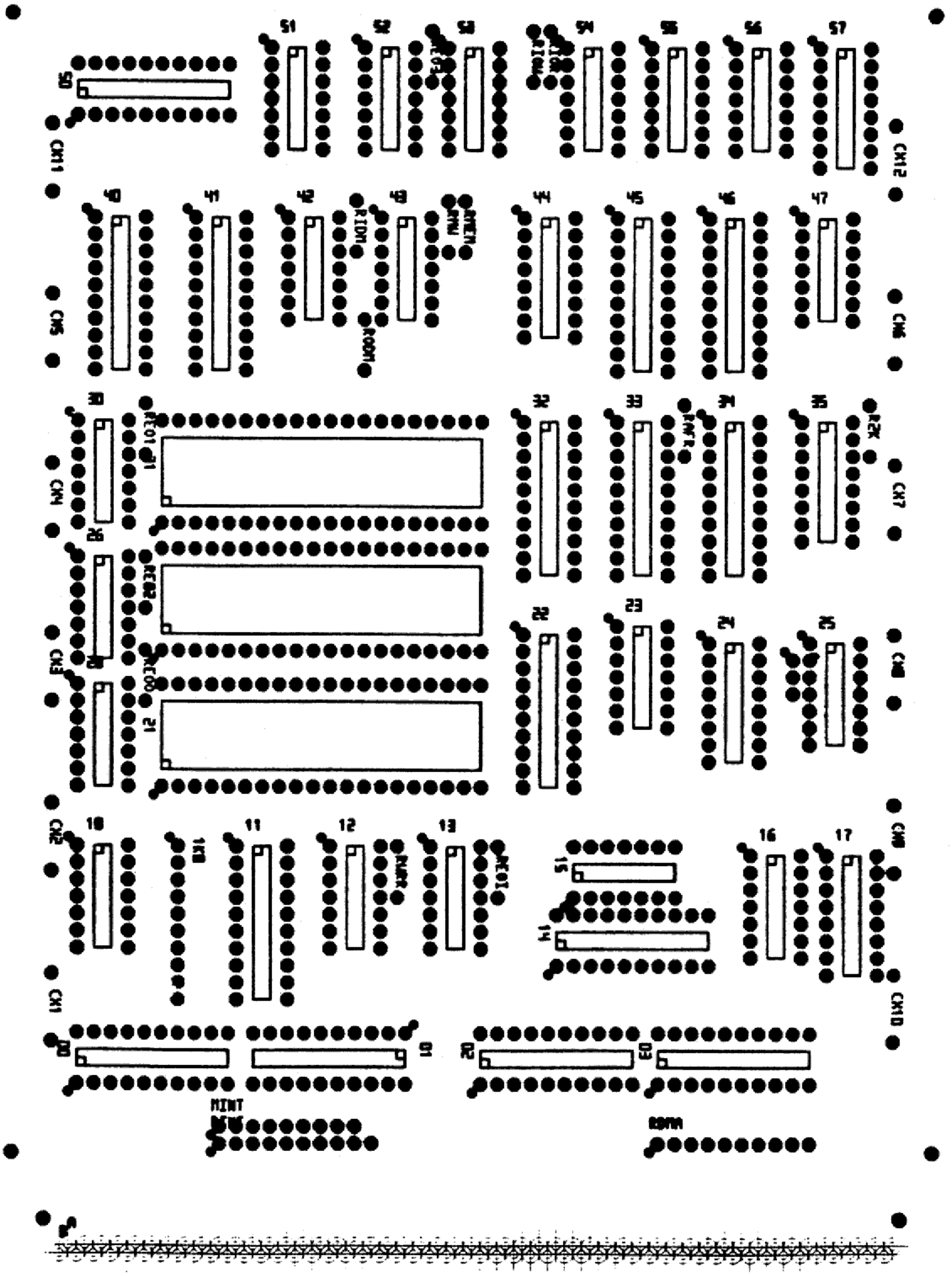
ADDRESSING.

The interrupt mask register is situated at i/o addres x´FE, and can only be written into.

dde dansk data elektronik aps

8509 DMA	Initialer/dato	Side 1
	Revideret	Projekt




50
51
52
RE03
53
RIO4
RIOR
54
55
56
57
CN11
CN12
40
41
42
RIDM
43
RMEM
RMW
44
45
46
47
CN5
CN6
ROOM
30
RE01
31
32
33
RMFR
34
35
R2K
CN4
CN7
26
RE02
22
23
24
25
CN3
CN8
20
RE00
21
CN2
CN9
10
1K8
11
12
DMER
13
RE01
15
14
16
17
CN1
CN10
D0
D1
D2
D3
MINT
RDMA

8509-010183

MEALY MACHINE FOR DMA ARBITER.

INPUT:

R0	HOLD REQUEST FROM FIRST DMA CHIP.
R1	HOLD REQUEST FROM SECOND DMA CHIP.
R2	HOLD REQUEST FROM THIRD DMA CHIP.
HA	HOLD ACKNOWLEDGE FROM CPU
HAS	HOLD ACKNOWLEDGE FROM CPU SYNCRONIZED TO MACHINE CLOCK

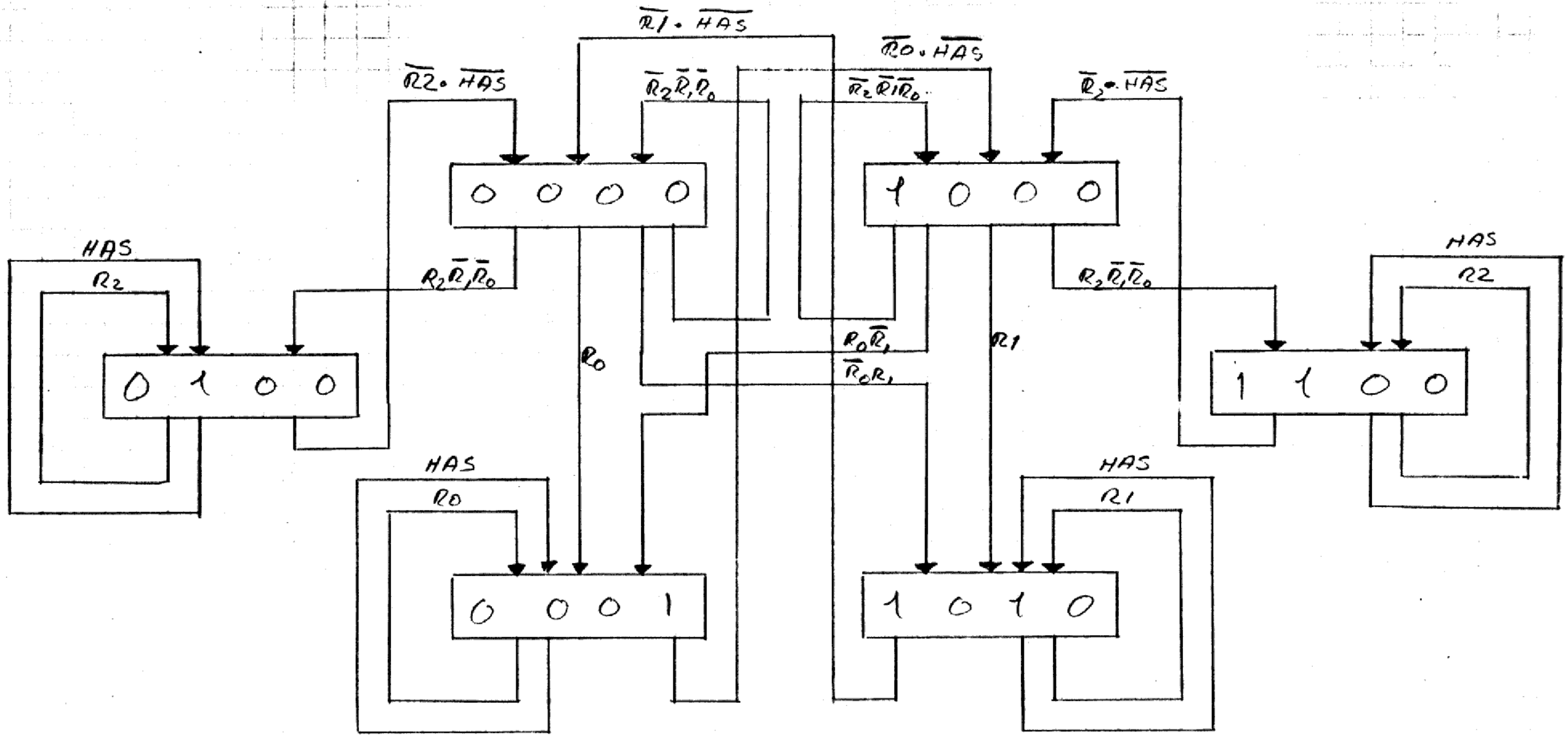
OUTPUT:

A0	HOLD ACKNOWLEDGE TO FIRST DMA CHIP.
A1	HOLD ACKNOWLEDGE TO SECOND DMA CHIP.
A2	HOLD ACKNOWLEDGE TO THIRD DMA CHIP.
HR	HOLD REQUEST TO CPU

S0	STATE VARIABLE
S1	STATE VARIABLE
S2	STATE VARIABLE
S3	STATE VARIABLE

```
S0 = /S3*/S2*/S1*/S0* R0+ S3*/S2*/S1*/S0* R0*/R1+
     /S3*/S2*/S1* S0* R0+/S3*/S2*/S1* S0* HAS

S1 = /S3*/S2*/S1*/S0*/R0* R1+ S3*/S2*/S1*/S0* R1+
      S3*/S2* S1*/S0* R1+ S3*/S2* S1*/S0* HAS

S2 = /S3*/S2*/S1*/S0*/R0*/R1* R2+/S3* S2*/S1*/S0* R2+
     /S3* S2*/S1*/S0* HAS+ S3*/S2*/S1*/S0*/R0*/R1* R2+
      S3* S2*/S1*/S0* HAS

S3 =  S3*/S2*/S1*/S0*/R0*/R1*/R2+ S3*/S2*/S1*/S0* R1+
      S3*/S2*/S1*/S0*/R0*/R1* R2+/S3*/S2*/S1*/S0*/R0* R1+
     /S3*/S2*/S1* S0*/HAS*/R0+ S3*/S2*/S1*/S0* R1+
      S3*/S2* S1*/S0* HAS

A0 = /S3*/S2*/S1* S0* HA
A1 =  S3*/S2* S1*/S0* HA
A2 = /S3* S2*/S1*/S0* HA + S3* S2*/S1*/S0* HA
```

```
HR = /S3*/S2*/S1*/S0* R0+/S3*/S2*/S1*/S0* R1+
     /S3*/S2*/S1*/S0* R2+ S3*/S2*/S1*/S0* R0+
      S3*/S2*/S1*/S0* R1+ S3*/S2*/S1*/S0* R2+
     /S3*/S2*/S1* S0* R0+ S3*/S2* S1*/S0* R1+
     /S3* S2*/S1*/S0* R2+ S3* S2*/S1*/S0* R2
   = /S2*/S1*/S0* R0+/S2*/S1*/S0* R1+/S2*/S1*/S0* R2+
     /S3*/S2*/S1* S0* R0+ S3*/S2* S1*/S0* R1+ S2*/S1*/S0* R2

---------------------------------------------------------------
```


$\overline{R1} \cdot \overline{HAS}$
$\overline{R2} \cdot \overline{HAS}$
$\overline{R_2}\overline{R_1}R_0$
$\overline{R_0} \cdot \overline{HAS}$
$\overline{R_2}\overline{R1}\overline{R_0}$
$\overline{R_2} \cdot \overline{HAS}$
0 0 0 0
1 0 0 0
HAS
R_2
$R_2\overline{R_1}\overline{R_0}$
$R_2\overline{R_1}\overline{R_0}$
HAS
R2
0 1 0 0
1 1 0 0
R_0
$R_0\overline{R_1}$
$\overline{R_0}R_1$
R1
HAS
$R0$
HAS
R1
0 0 0 1
1 0 1 0

$S_0 = \overline{S_3}\,\overline{S_2}\,\overline{S_1}\,\overline{S_0}\,R0 + S_3\overline{S_2}\,\overline{S_1}S_0\,R_0\overline{R_1} + \overline{S_3}\,\overline{S_2}\,\overline{S_1}S_0R_0 + \overline{S_3}\,\overline{S_2}\,\overline{S_1}S_0 \cdot HAS$

$S_1 = \overline{S_3}\,\overline{S_2}\,\overline{S_1}\,\overline{S_0}\,\overline{R_0}R_1 + S_3\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot R_1 + S_3\overline{S_2}S_1\overline{S_0} \cdot R_1 + S_3\overline{S_2}S_1\overline{S_0} \cdot HAS$

$S_2 = \overline{S_3}\,\overline{S_2}\,\overline{S_1}\,\overline{S_0}\,\overline{R_0}\,\overline{R_1}R_2 + \overline{S_3}S_2\overline{S_1}\,\overline{S_0} \cdot R_2 + \overline{S_3}S_2\overline{S_1}\,\overline{S_0} \cdot HAS + S_3\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot \overline{R_0}\,\overline{R_1}R_2 + S_3S_2\overline{S_1}\,\overline{S_0}R_2 + S_3S_2\overline{S_1}\,\overline{S_0} \cdot HAS$

$S_3 = S_3\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot \overline{R_0}\,\overline{R_1}\,\overline{R_2} + S_3\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot R_1 + S_3\overline{S_2}S_1S_0 \cdot \overline{R_0}R_1R_2 + \overline{S_3}\,\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot \overline{R_0}R_1 + \overline{S_3}S_2S_1S_0 \cdot \overline{HAS} \cdot \overline{R_0} + S_3\overline{S_2}S_1\overline{S_0} \cdot R1 + \overline{S_3}S_2S_1\overline{S_0} \cdot HAS$
$+ S_3S_2\overline{S_1}\,\overline{S_0}$

$A0 = \overline{S_3}\,\overline{S_2}\,\overline{S_1}S_0 \cdot HA$, $A1 = S_3\overline{S_2}S_1\overline{S_0} \cdot HA$, $A2 = \overline{S_3}S_2\overline{S_1}\,\overline{S_0} \cdot HA + S_3S_2\overline{S_1}\,\overline{S_0} \cdot HA$

$BR = \overline{S_3}\,\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot R0 + \overline{S_3}\,\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot R1 + \overline{S_3}\,\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot R2 + S_3\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot R0 + S_3\overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot R1 + \overline{S_3}\,\overline{S_2}\,\overline{S_1}\,\overline{S_0}R_2 + \overline{S_3}\,\overline{S_2}\,\overline{S_1}S_0R_0 + S_3\overline{S_2}S_1\overline{S_0}R_1 +$
$\overline{S_3}S_2\overline{S_1}\,\overline{S_0} \cdot R2 + S_3S_2\overline{S_1}\,\overline{S_0}R_2 = \overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot R_0 + \overline{S_2}\,\overline{S_1}\,\overline{S_0}R_1 + \overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot R_2 + \overline{S_3}\,\overline{S_2}\,\overline{S_1}S_0 \cdot R_0 + S_3\overline{S_2}S_1\overline{S_0} \cdot R_1 + S_2\overline{S_1}\,\overline{S_0} \cdot R2$

```
PAL16L8
P8509011 14121982
ADRESSEDEKODNING FOR KORT
TWJ DMA82
AI0  AI1    AI2  AI3  AI4  AI5   AI6   AI7   VEND  GND
HA3  /CSALL MEL  ICS5 MCS4 /MCS3 /DCS2 /DCS1 /DCS0 VCC

IF (VCC) DCS0 = /AI7*/AI6*AI5*/AI4*/VEND
IF (VCC) DCS1 = /AI7*/AI6*AI5*AI4*/VEND
IF (VCC) DCS2 = /AI7*AI6*/AI5*/AI4*/VEND
IF (VCC) MCS3 = AI7*AI6*AI5*AI4*AI3*/AI2*AI1*AI0*/VEND
IF (VCC) MCS4 = MEL+AI2*AI1+/AI2*/AI1+/AI0+VEND
IF (VCC) ICS5 = MEL+/AI2+/AI1+AI0+VEND
IF (VCC) CSALL = /AI7*/AI6*AI5*/VEND +
               /AI7*AI6*/AI5*/AI4*/VEND+
                AI7*AI6*AI5*AI4*AI3*/AI2*AI1*AI0*/VEND +
                AI7*AI6*AI5*AI4*AI3*AI2*/AI1*AI0*/VEND+
                AI7*AI6*AI5*AI4*AI3*AI2*AI1*/AI0*/VEND

IF (VCC) MEL = AI7*AI6*AI5*AI4*AI3

DESCRIPTION
;KHJGHFKV
```

```
PAL16R4
P8509022 14121982
ARBITRERING FOR DMA KREDSE
TWJ DMA82
CL3 R0    R1    R2     GND   HAS  HA    NC   NC   GND
GND  HA2  HOLDR  /S0   /S1   /S2  /S3  HA1  HA0 VCC

S0 := /S3*/S2*/S1*/S0* R0             + S3*/S2*/S1*/S0* R0*/R1         +
      /S3*/S2*/S1* S0* R0             +/S3*/S2*/S1* S0*             HAS+
      /S3*/S2* S1* S0* R0

S1 := /S3*/S2*/S1*/S0*/R0*R1          + S3*/S2*/S1*/S0*     R1         +
       S3*/S2* S1*/S0*     R1         + S3*/S2* S1*/S0*             HAS


S2 := /S3*/S2*/S1*/S0*/R0*/R1*R2     +/S3* S2*/S1*/S0* R2             +
      /S3* S2*/S1*/S0*             HAS+ S3*/S2*/S1*/S0*/R0*/R1* R2     +
       S3* S2*/S1*/S0*         R2     + S3* S2*/S1*/S0*             HAS

S3 :=  S3*/S2*/S1*/S0*/R0*/R1*/R2     + S3*/S2*/S1*/S0*     R1         +
       S3*/S2*/S1*/S0*/R0*/R1* R2     +/S3*/S2*/S1*/S0*/R0* R1         +
      /S3*/S2*/S1* S0*/R0*        /HAS+ S3*/S2* S1*/S0*     R1         +
       S3*/S2* S1*/S0*             HAS+ S3* S2*/S1*/S0

IF (VCC) /HA0 =
                 S3+ S2+ S1+/S0+/HA
IF (VCC) /HA1 =
                /S3+ S2+/S1+ S0+/HA
IF (VCC) /HA2 =
                /S2+ S1+ S0+/HA

IF (VCC) /HOLDR =
          /R0*/R1*/R2+S0*/R0+S1*/R1+S2*/R2+S2*S1+S3*S0+S2*S0


FUNCTION TABLE:
CL3  R0 R1 R2 HAS HA  HA2 S0 S1 S2 S3 HA1 HA0
```

```
---------------------------------------------
C    L  L  L   L  L    L  L  L  L  L  L   L       FRA 0 TIL 0
C    H  X  X   L  L    L  H  L  L  L  L   L       FRA 0 TIL 1
C    H  X  X   L  H    L  H  L  L  L  L   H       FRA 1 TIL 1
C    L  X  X   H  H    L  H  L  L  L  L   H       FRA 1 TIL 1
C    L  X  X   H  L    L  H  L  L  L  L   L       FRA 1 TIL 8
C    L  X  X   L  L    L  L  L  L  H  L   L       FRA 8 TIL A   *
C    X  H  X   L  L    L  L  H  L  H  L   L       FRA A TIL A
C    X  H  X   L  H    L  L  H  L  H  H   L       FRA A     A
C    X  L  X   H  H    L  L  H  L  H  H   L           A     A
C    X  L  X   H  L    L  L  H  L  H  L   L
C    X  L  X   L  L    L  L  L  L  L  L   L       FRA A TIL 0
C    L  L  H   L  L    L  L  L  H  L  L   L       FRA 0 TIL 4
C    X  X  H   X  H    H  L  L  H  L  L   L       FRA 4 TIL 4
C    X  X  L   H  L    L  L  L  H  L  L   L       FRA 4 TIL 4
C    X  X  L   L  H    L  L  L  L  L  L   L       FRA 4 TIL 0
----------------------------------------------
```

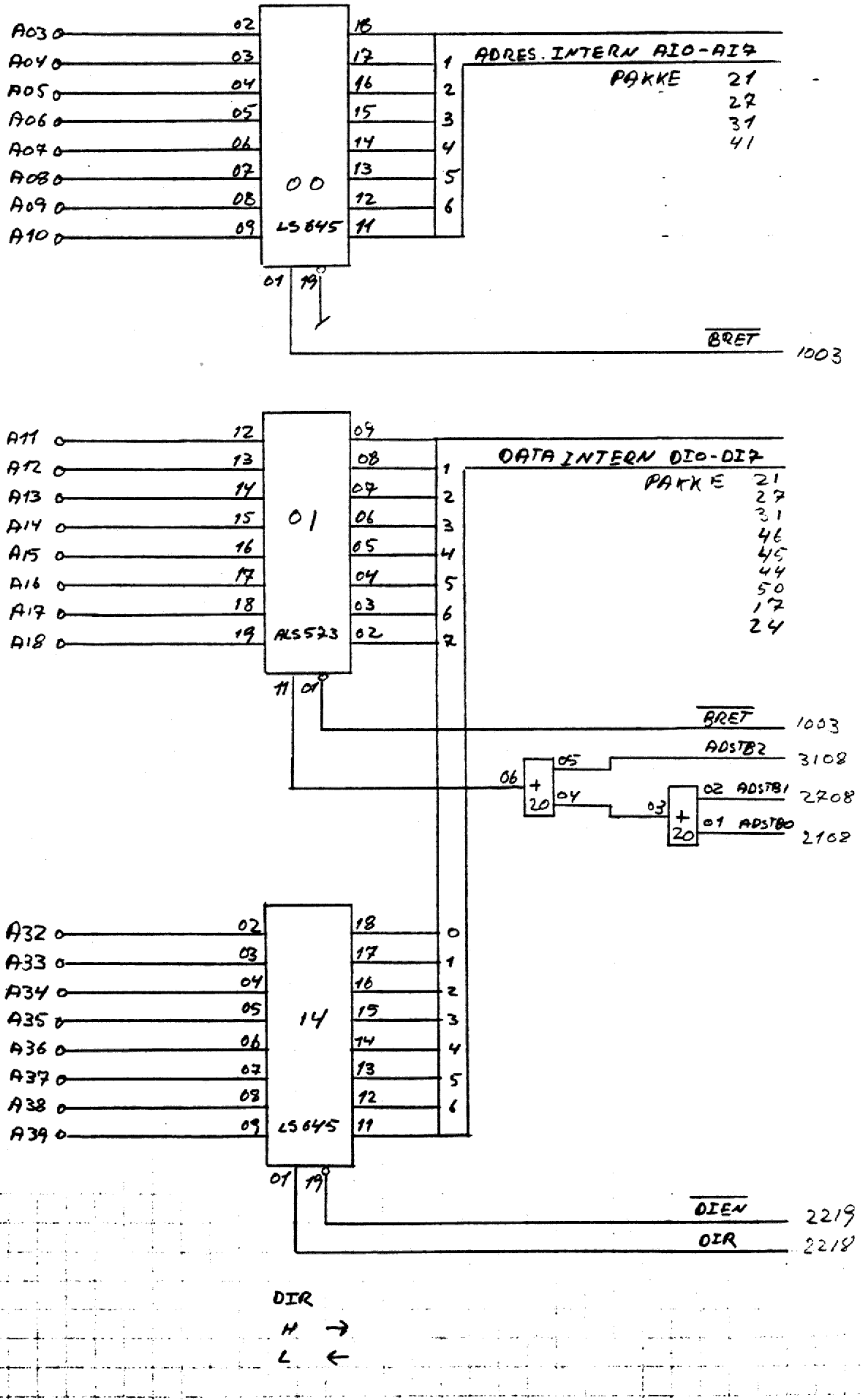
DESCRIPTION

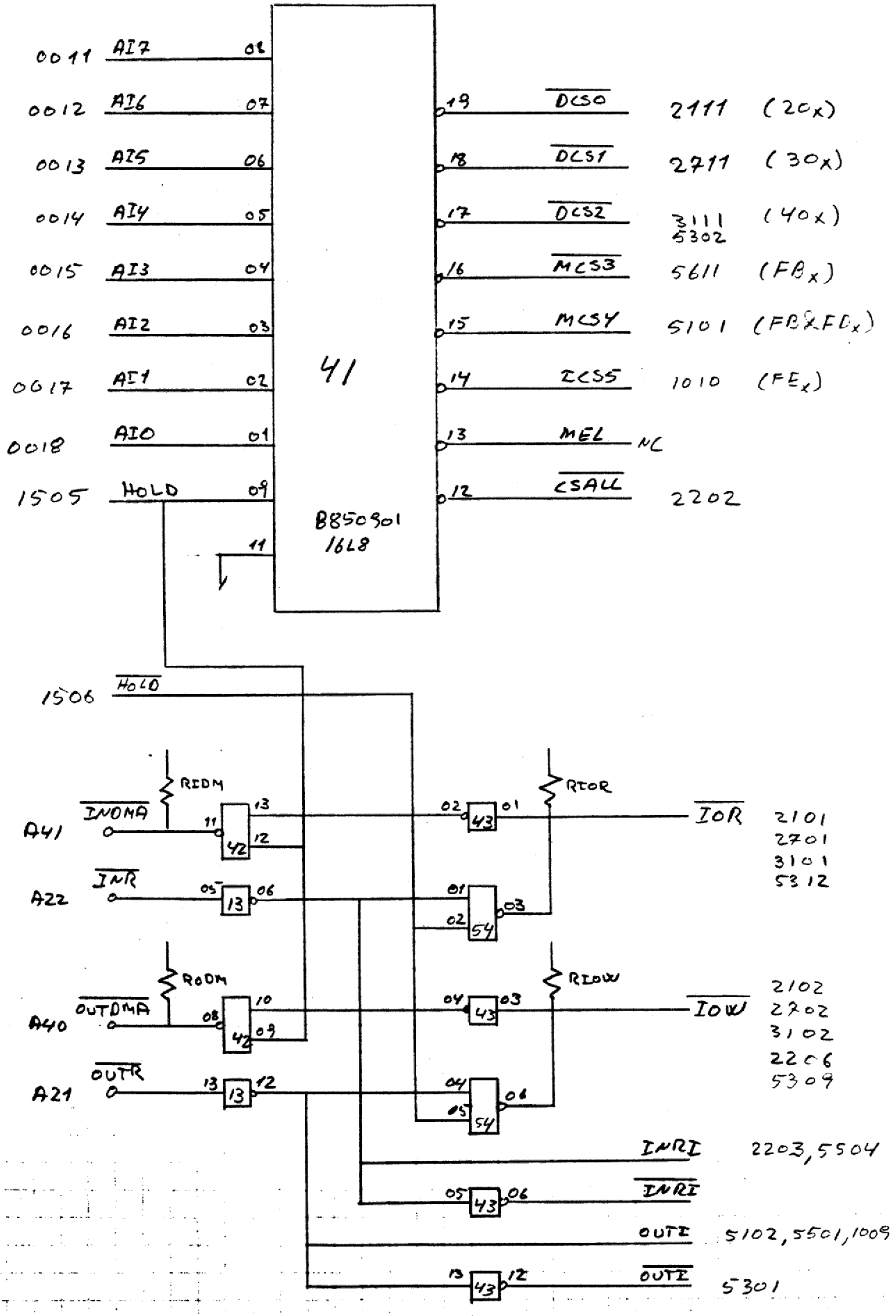
```
;ARBITRERINGEN HAR TRE  REQUEST OG DO TRE  AKNOWLEDGES
;EN OMARBITRERING FORDRER AT HOLDA OG DEN SIDSTE HREQ HAR
;VÆRET INAKTIVE
```

```
PAL16L8
P8509031 14121982
MAPPER ADRESSE AFH. AF DMAACK RETTET 11071982
TWJ DMA82
/ACK8 /ACK7  /ACK6  /ACK5  /ACK4 /ACK3 /ACK2 /ACK1 /ACK0 GND
/ACK9 MA3    /ACK11 /ACK10 MA2   MA1   HOLD  OC    MA0   VCC

IF (OC) MA0 = /ACK0*/ACK2*/ACK4*/ACK6*/ACK8*/ACK10 + /HOLD

IF (OC) MA1 = /ACK1*/ACK2*/ACK5*/ACK6*/ACK9*/ACK10 + /HOLD

IF (OC) MA2 = /ACK3*/ACK4*/ACK5*/ACK6*/ACK11       + /HOLD

IF (OC) MA3 = /ACK7*/ACK8*/ACK9*/ACK10*/ACK11      + /HOLD

DESCRIPTION

;MA0-MA3 TIL MAPPER BEMÆRK AT DER KAN 3-STATES
```

```
PAL16R4
P8509041
TWJ DMA82

CL3 /CSALL  INRI  MEMR  MEMW /IOW   HOLD /RDATA /VENT GND
GND  READY /AFBR /MELO /MEL1 /MEL2 /MEL3 /DIRI /DIEN  VCC

IF (VCC) READY = /MELO + VENT

         MELO := MEMR*/VENT + MEMW*/VENT

IF (VCC) DIEN= /HOLD* INRI*CSALL+/HOLD*IOW*CSALL +HOLD*MEMR +
               /HOLD*AFBR +  HOLD*RDATA + MEMW*MEL3*HOLD +
               DIEN*MEMW*HOLD

IF (VCC) DIRI= /HOLD*INRI*CSALL +/HOLD*AFBR +RDATA* HOLD+
               MEMW*MEL3*HOLD+ DIRI*MEMW*HOLD

         MEL3:= MEMR*HOLD*/IOW + MEL3*HOLD*/IOW*/MEMW

DESCRIPTION

STYRING AF DATABUFFERE OGSÆ UNDER MEMORY BLOCK MOVE
DER SKAL DOG VÆRE OPHOLD MELLEM BLOCK MOVE OG ANDRE
TRANSAKTIONER DVS VEND SKAL GÆ LAV
```

Initialer/dato	Side
TW-3	3.
Revideret	Projekt

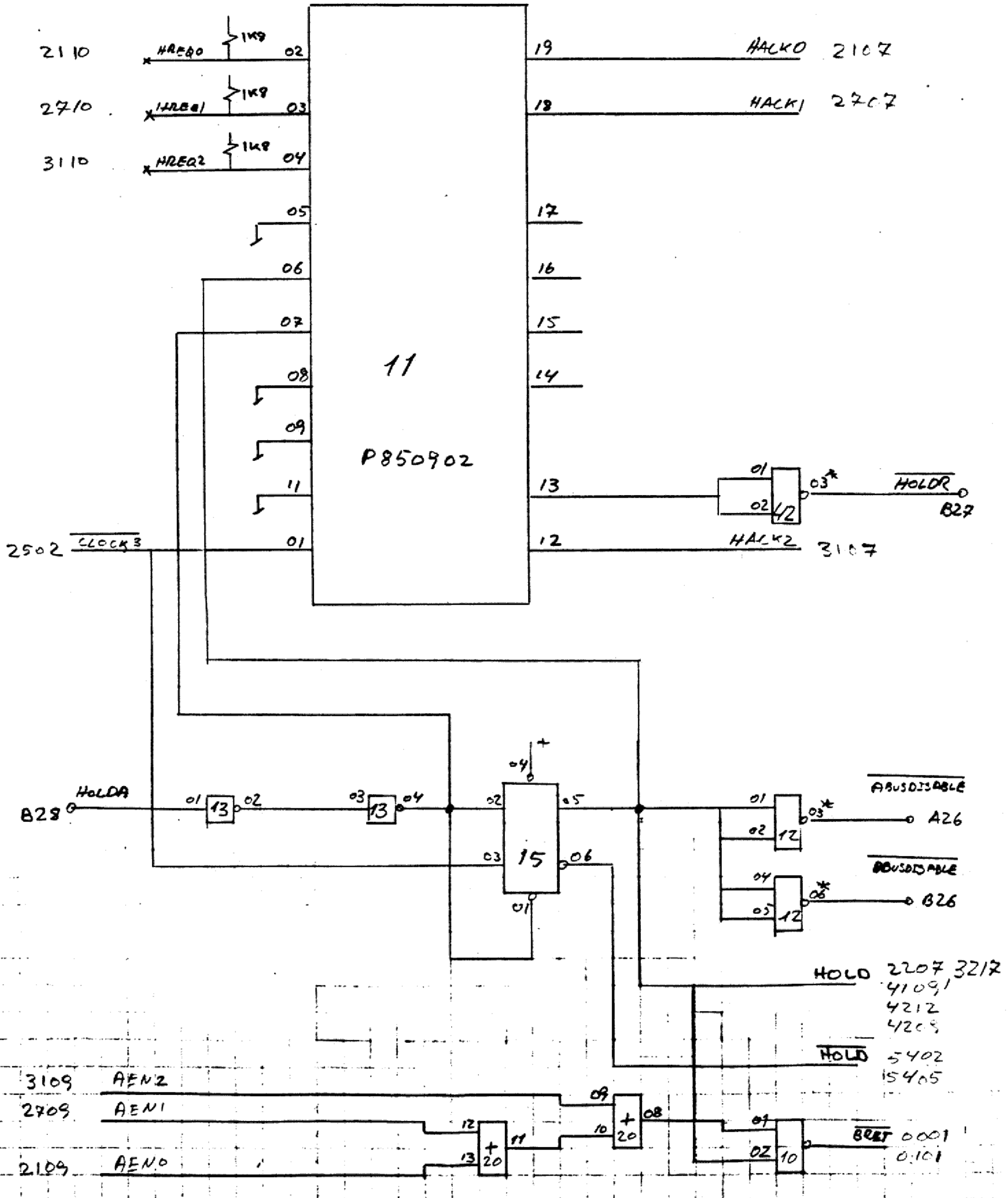


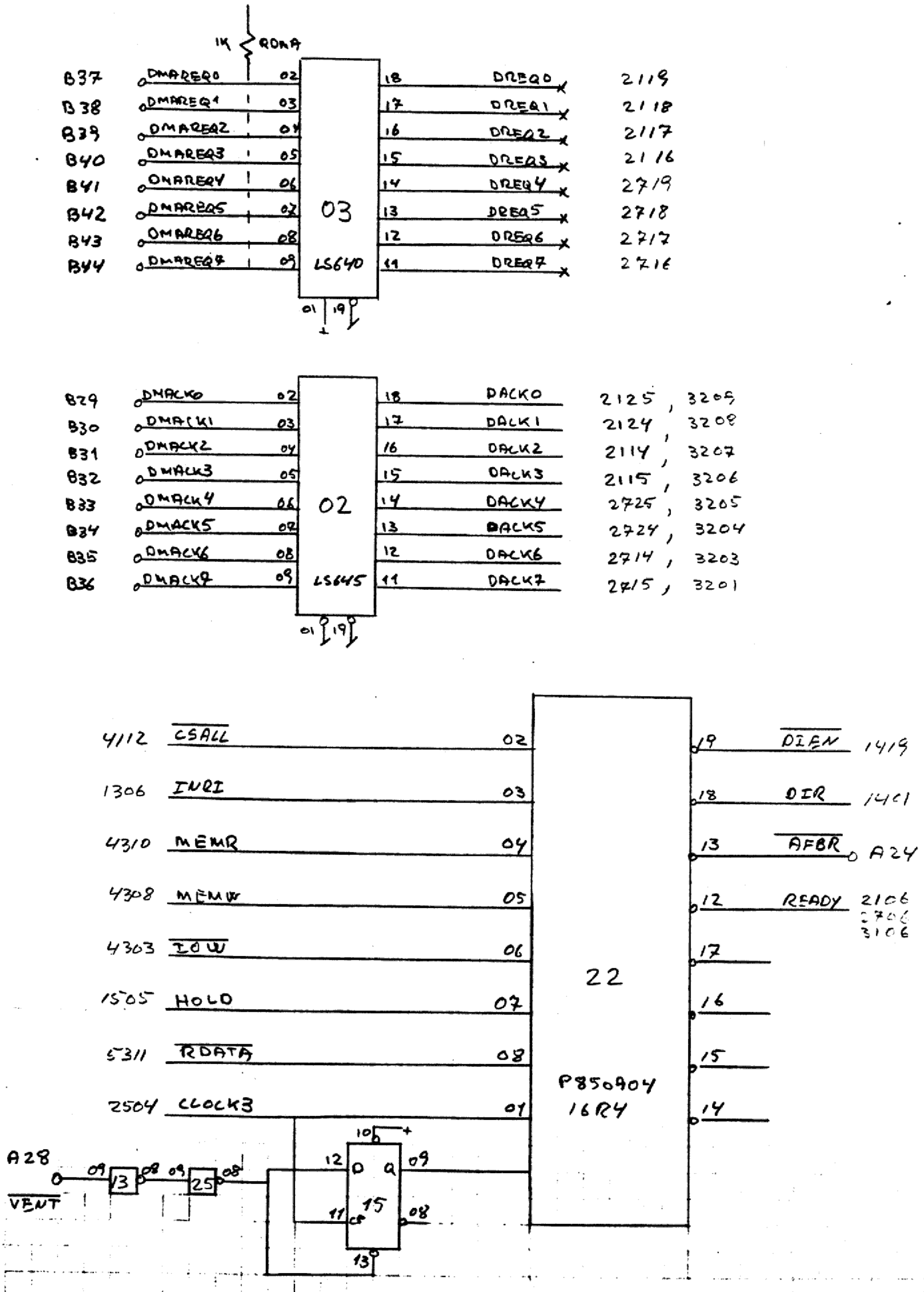

0011 AI7 08
0012 AI6 07
0013 AI5 06
0014 AI4 05
0015 AI3 04
0016 AI2 03
0017 AI1 02
0018 AI0 01
1505 HOLD 09
11
41
B850901
16L8
19 DCS0 2111 (20x)
18 DCS1 2711 (30x)
17 DCS2 3111 5302 (40x)
16 MCS3 5611 (FBx)
15 MCS4 5101 (FB&FDx)
14 ZCS5 1010 (FEx)
13 MEL NC
12 CSALL 2202
1506 HOLD
RIDM
A41 INDMA 11 42 13 12
02 43 01
RIOR
IOR 2101 2701 3101 5312
A22 INR 05 13 06
01 02 54 03
RODM
A40 OUTDMA 08 42 10 09
04 43 03
RIOW
IOW 2102 2702 3102 2206 5309
A21 OUTR 13 13 12
04 05 54 06
INRI 2203,5504
05 43 06 INRI
OUTI 5102,5501,1009
13 43 12 OUTI 5301

31	27	21					21	27	31
4301	4301	4301	01	IOR	A7	40	0011	0011	0011
4303	4303	4303	02	IOW	A6	39	0012	0012	0012
4311.	4311	4311	03	MEMR	A5	38	0013	0013	0013
4309	4309	4309	04	MEMW	A4	37	0014	0014	0014
+	+	+	05	TEST	EOP	36	3013	3011	5305
2212	2212	2212	06	READY	A3	35	0015	0015	0015
1112	1118	1119	07	HACK	A2	34	0016	0016	0016
2005	2002	2001	08	ADSTB	A1	33	0017	0017	0017
2009	2012	2013	09	AEN	A0	32	0018	0018	0018
1104	1103	1102	10	HREQ	Vcc	31	+	+	+
4117	4118	4119	11	CS	DB0	30	1418	1418	1418
2504	2504	2504	12	CLK	DB1	29	1417	1417	1417
2512	2512	2512	13	RESET	DB2	28	1416	1416	1416
3214	0212	0216	14	DACK2	DB3	27	1415	1415	1415
3213	0211	0215	15	DACK3	DB4	26	1414	1414	1414
NC	0311	0315	16	DREQ3	DACK0	25	0218	0214	5310
NC	0312	0316	17	DREQ2	DACK1	24	0217	0213	5313
5209	0313	0317	18	DREQ1	DB5	23	1413	1413	1413
5208	0314	0318	19	DREQ0	DB6	22	1412	1412	1412
			20	Vss	DB7	21	1411	1411	1411

21
27
31

Initialer/dato	Side 6.
Revideret TW-J 29/9-82	Projekt

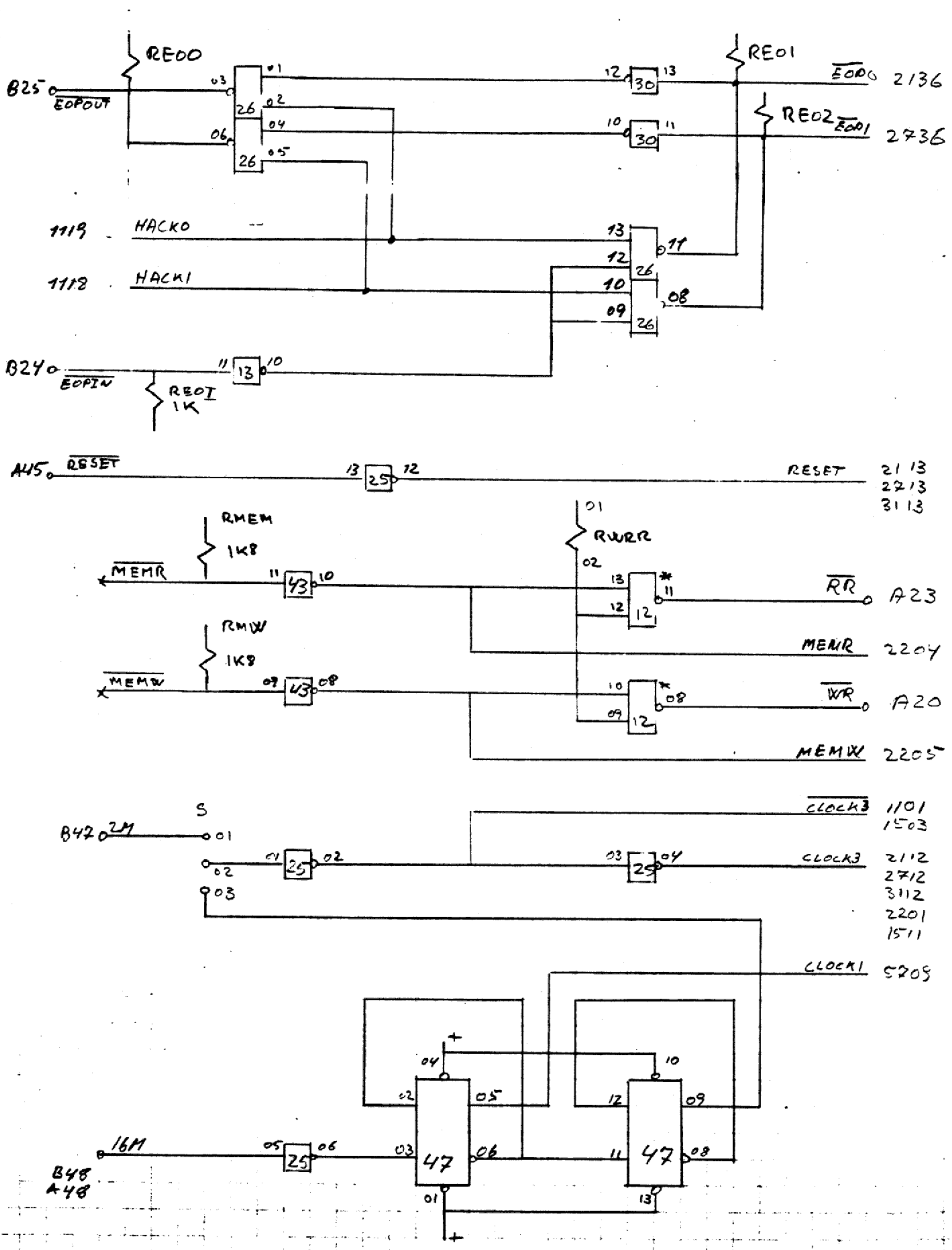


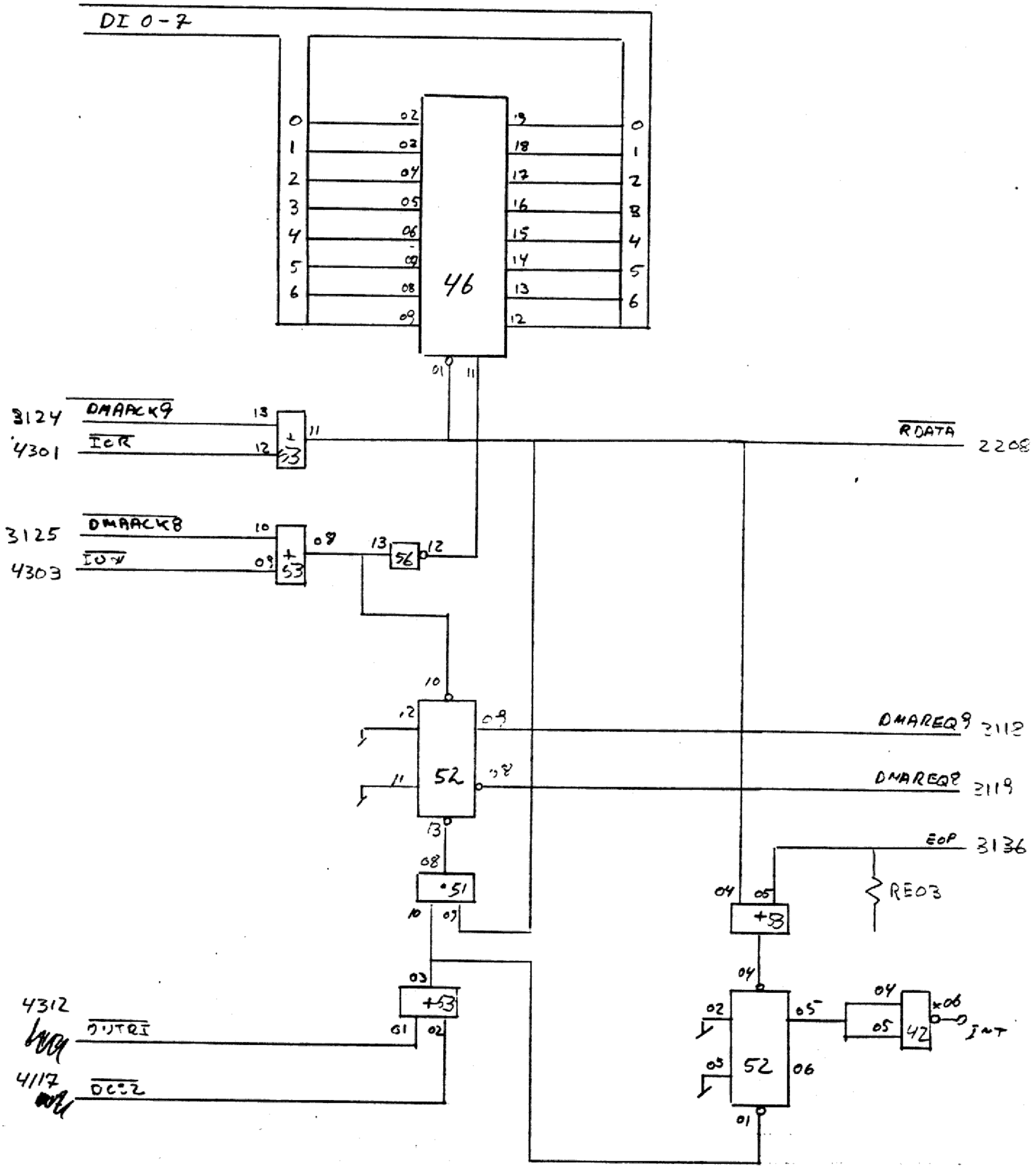

1K
RDMA
B37 DMAREQ0 02 18 DREQ0 2119
B38 DMAREQ1 03 17 DREQ1 2118
B39 DMAREQ2 04 16 DREQ2 2117
B40 DMAREQ3 05 15 DREQ3 2116
B41 DMAREQ4 06 14 DREQ4 2719
B42 DMAREQ5 07 13 DREQ5 2718
B43 DMAREQ6 08 12 DREQ6 2717
B44 DMAREQ7 09 11 DREQ7 2716
03
LS640
01 19
B29 DMACK0 02 18 DACK0 2125, 3209
B30 DMACK1 03 17 DACK1 2124, 3208
B31 DMACK2 04 16 DACK2 2114, 3207
B32 DMACK3 05 15 DACK3 2115, 3206
B33 DMACK4 06 14 DACK4 2725, 3205
B34 DMACK5 07 13 DACK5 2724, 3204
B35 DMACK6 08 12 DACK6 2714, 3203
B36 DMACK7 09 11 DACK7 2715, 3201
02
LS645
01 19
4112 CSALL 02
1306 INRI 03
4310 MEMR 04
4308 MEMW 05
4303 IOW 06
1505 HOLD 07
5311 RDATA 08
2504 CLOCK3 01
22
P850904
16R4
19 DIEN 1419
18 DIR 1401
13 AFBR A24
12 READY 2106 2706 3106
17
16
15
14
A28
VENT
09 13 08
09 25 08
10 +
12 D Q 09
11 CP 15 Q̄ 08
13

3 2,7

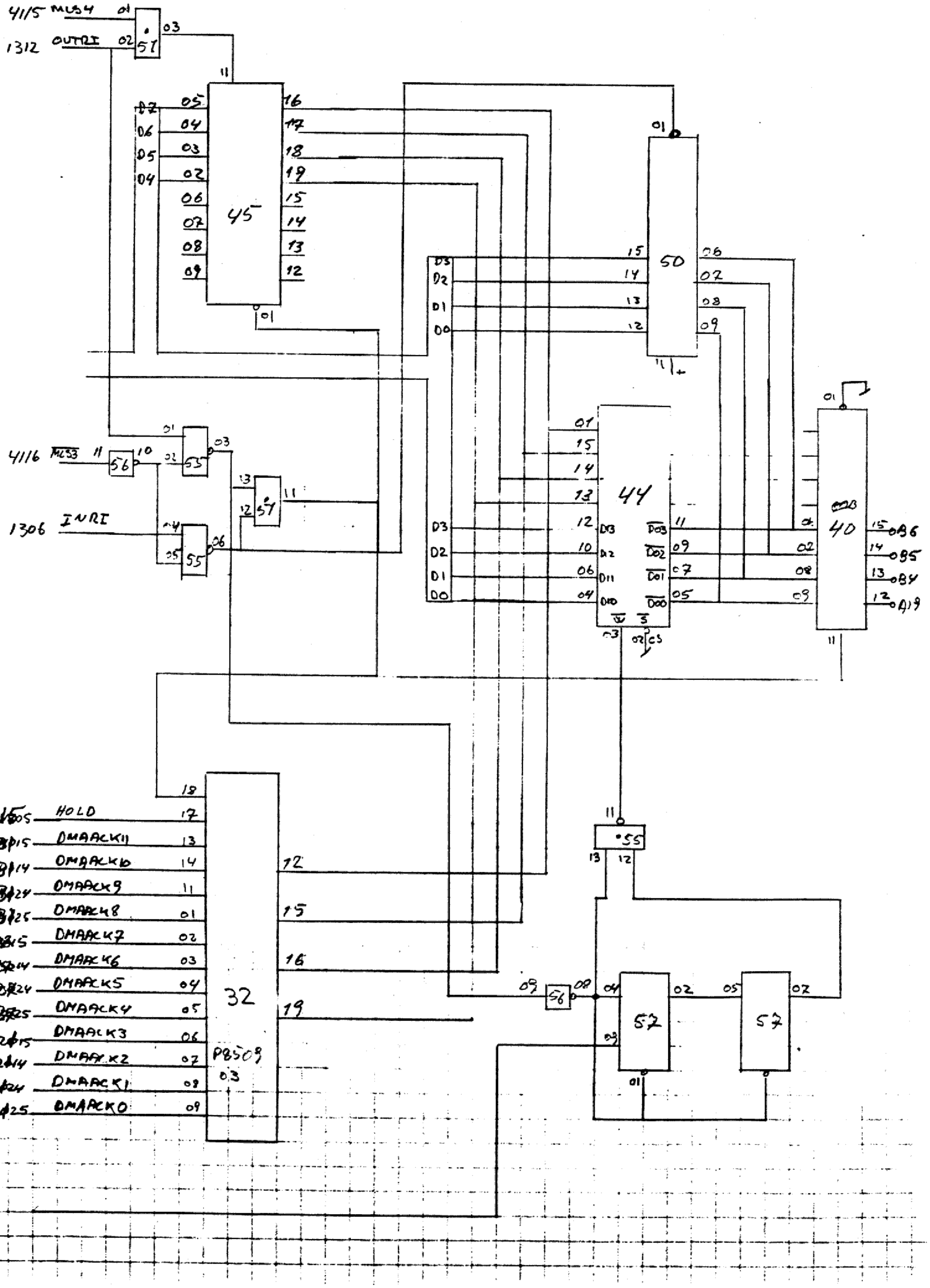
dde dansk data elektronik aps

Initialer/dato	Side 8
Revideret 7N-3 20/9-82	Projekt



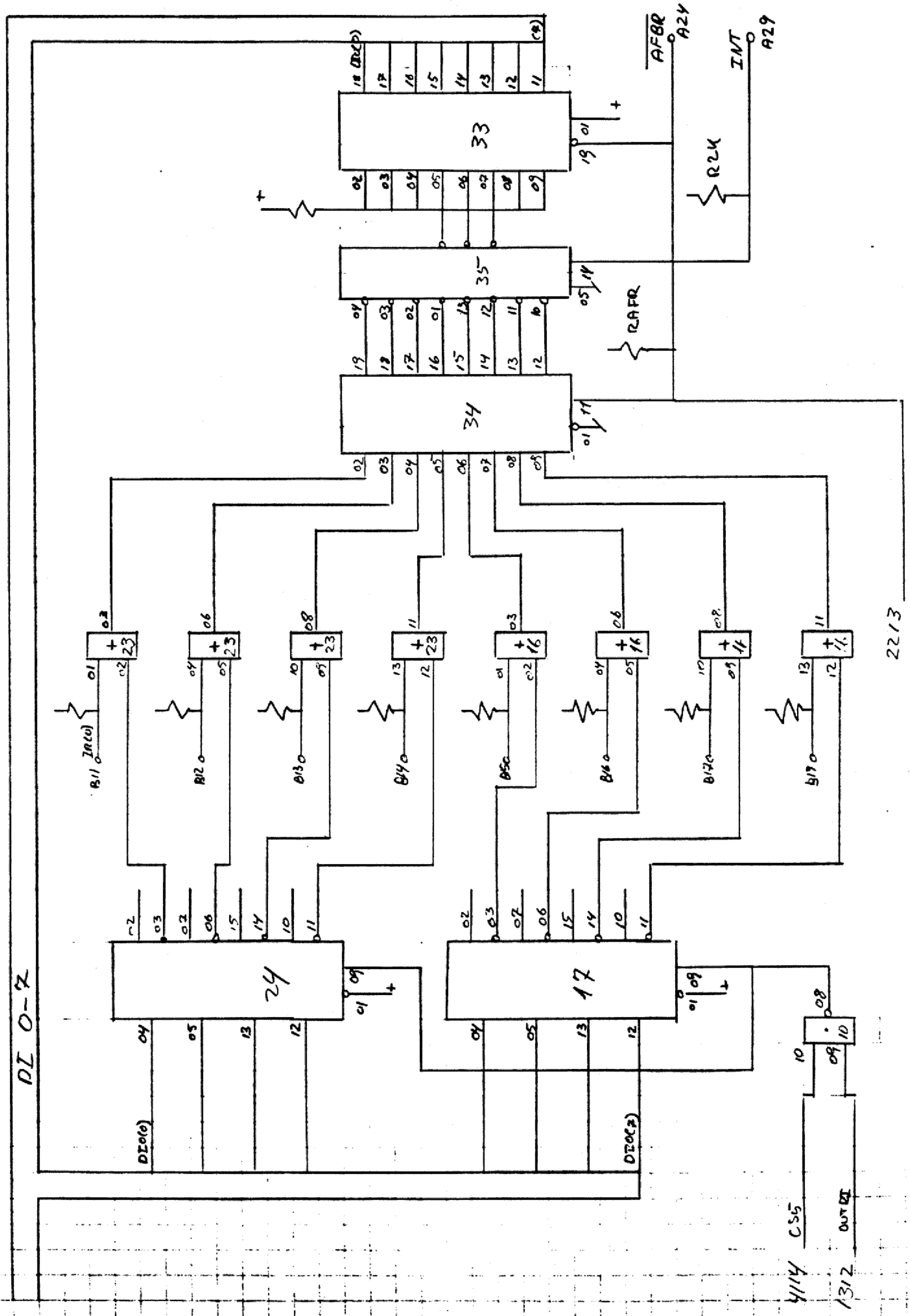

DI 0-7
0
1
2
3
4
5
6
02
03
04
05
06
07
08
09
46
19
18
17
16
15
14
13
12
0
1
2
3
4
5
6
01
11
3124 DMAACK9
4301 ICR
13
12
11
53
RDATA 2208
3125 DMAACK8
4303 IOW
10
09
08
53
13
56
12
10
12
11
52
09
08
13
DMAREQ9 3118
DMAREQ8 3119
08
51
10
09
03
+53
01
02
4312 OUTRI
4117 OCE2
EOP 3136
RE03
04
05
+8
04
02
03
52
05
06
01
04
05
42
06
INT

	Initialer/dato	Side 10
	Revideret	Projekt



dansk data elektronik aps

Initialer/dato	Side 11
Revideret	Projekt



Am9517A
Multimode DMA Controller

DISTINCTIVE CHARACTERISTICS

- Four independent DMA channels, each with separate registers for Mode Control, Current Address, Base Address, Current Word Count and Base Word Count.
- Transfer modes: Block, Demand, Single Word, Cascade
- Independent autoinitialization of all channels
- Memory-to-memory transfers
- Memory block initialization
- Address increment or decrement
- Master system disable
- Enable/disable control of individual DMA requests
- Directly expandable to any number of channels
- End of Process input for terminating transfers
- Software DMA requests
- Independent polarity control for DREQ and DACK signals
- Compressed timing option speeds transfers – up to 2.5M bytes/second
- +5 volt power supply
- Advanced N-channel silicon gate MOS technology
- 40 pin Hermetic DIP package
- New 9517A-5 5MHz version for higher speed CPU compatability

GENERAL DESCRIPTION

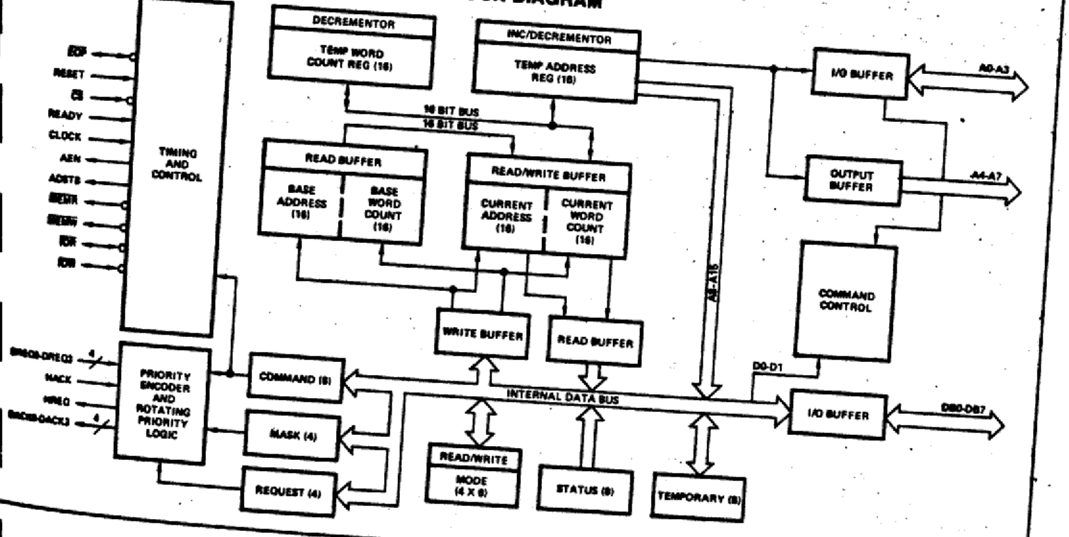
The Am9517A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information to or from the system memory. Memory-to-memory transfer capability is also provided. The Am9517A offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The Am9517A is designed to be used in conjunction with an external 8-bit address register such as the Am74LS373. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips.

The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process ($\overline{EOP}$).

Each channel has a full 64K address and word count capability. An external $\overline{EOP}$ signal can terminate a DMA or memory-to-memory transfer. This is useful for block search or compare operations using external comparators or for intelligent peripherals to abort erroneous services.

BLOCK DIAGRAM



03040B-1

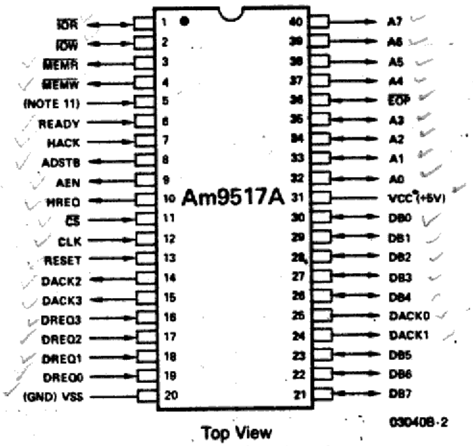
ORDERING INFORMATION

Package Type	Ambient Temperature	Maximum Clock Frequency		
		3MHz	4MHz	5MHz
Hermetic DIP/ Molded DIP	0°C ≤ T_A ≤ +70°C	AM9517ADC/PC AM9517A-1DC/PC	AM9517A-4DC/PC	AM9517A-5DC/PC
Hermetic DIP	−40°C ≤ T_A ≤ +85°C	AM9517ADI AM9517A-1DI	AM9517A-4DI/PI	
Hermetic DIP	−55°C ≤ T_A ≤ +125°C	AM9517ADMB		

9517

SYMBOL NAVN.

Figure 1. Connection Diagram
D-40, P-40



Top View
*Pin 1 is marked for orientation.

INTERFACE SIGNAL DESCRIPTION

VCC: +5 Volt Supply
VSS: Ground

CLK (Clock, Input)

This input controls the internal operations of the Am9517A and its rate of data transfers. The input may be driven at up to 3MHz for the standard Am9517A, up to 4MHz for the Am9517A-4, and up to 5MHz for the Am9517A-5.

$\overline{CS}$ (Chip Select, Input)

Chip Select is an active low input used to select the Am9517A as an I/O device during an I/O Read or I/O Write by the host CPU. This allows CPU communication on the data bus. During multiple transfers to or from the Am9517A by the host CPU, $\overline{CS}$ may be held low providing $\overline{IOR}$ or $\overline{IOW}$ is toggled following each transfer.

RESET (Reset, Input)

Reset is an asynchronous active high input which clears the Command, Status, Request and Temporary registers. It also clears the First/Last Flip/Flop and sets the Mask register. Following a Reset the device is in the Idle cycle.

READY (Ready, Input)

Ready is an input used to extend the memory read and write pulses from the Am9517A to accommodate slow memories or I/O peripheral devices.

HACK (Hold Acknowledge, Input)

The active high Hold Acknowledge from the CPU indicates that control of the system buses has been relinquished.

DREQ0-DREQ3 (DMA Request, Input)

The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In Fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. Polarity of DREQ is programmable. Reset initializes these lines to active high.

DB0-DB7 (Data Bus, Input/Output)

The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled during the I/O Read by the host CPU, permitting the CPU to examine the contents of an Address register, the Status register, the Temporary register or a Word Count register. The Data Bus is enabled to input data during a host CPU I/O write, allowing the CPU to program the Am9517A control registers. During DMA cycles the most significant eight bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations data from the source memory location comes into the Am9517A's Temporary register on the read-from-memory half of the operation. On the write-to-memory half of the operation, the data bus outputs the Temporary register data into the destination memory location.

$\overline{IOR}$ (I/O Read, Input/Output)

I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the Am9517A to access data from a peripheral during a DMA Write transfer.

$\overline{IOW}$ (I/O Write, Input/Output)

I/O Write is a bidirectional active low three-state line. In the Idle cycle it is an input control signal used by the CPU to load information into the Am9517A. In the Active cycle it is an output control signal used by the Am9517A to load data to the peripheral during a DMA Read transfer.

Write operations by the CPU to the Am9517A require a rising $\overline{IOW}$ edge following each data byte transfer. It is not sufficent to hold the $\overline{IOW}$ pin low and toggle $\overline{CS}$.

$\overline{EOP}$ (End of Process, Input/Output)

$\overline{EOP}$ is an active low bidirectional open-drain signal providing information concerning the completion of DMA service. When a channel's Word Count goes to zero, the Am9517A pulses $\overline{EOP}$ low to provide the peripheral with a completion signal. $\overline{EOP}$ may also be pulled low by the peripheral to cause premature completion. The reception of $\overline{EOP}$, either internal or external, causes the currently active channel to terminate the service, to set its TC bit in the Status register and to reset its request bit. If Autoinitialization is selected for the channel, the current registers will be updated from the base registers. Otherwise the channel's mask bit will be set and the register contents will remain unaltered.

During memory-to-memory transfers, $\overline{EOP}$ will be output when the TC for channel 1 occurs. $\overline{EOP}$ always applies to the channel with an active DACK; external $\overline{EOP}$s are disregarded when DACK0-DACK3 are all inactive if the DMA is in state SI.

In situations where two or more Am9517A DMAs are cascaded, the $\overline{EOP}$ pins should be logically OR'ed (not wire-OR'ed).

Because $\overline{EOP}$ is an open-drain signal, an external pullup resistor is required. Values of 3.3K or 4.7K are recommended; the $\overline{EOP}$ pin cannot sink the current passed by a 1K pullup.

A0-A3 (Address, Input/Output)

The four least significant address lines are bidirectional 3-state signals. During DMA Idle cycles they are inputs and allow the host CPU to load or read control registers. When the DMA is active, they are outputs and provide the lower 4-bits of the output address.

A4-A7 (Address, Output)

The four most significant address lines are three-state outputs and provide four bits of address. These lines are enabled only during DMA service.

HREQ (Hold Request, Output)

The Hold Request to the CPU is used by the DMA to request control of the system bus. Software requests or unmasked DREQs cause the Am9517A to issue HREQ.

DACK0-DACK3 (DMA Acknowledge, Output)

The DMA Acknowledge lines indicate that a channel is active. In many systems they will be used to select a peripheral. Only one DACK will be active at a time and none will be active unless the DMA is in control of the bus. The polarity of these lines is programmable. Reset initializes them to active-low.

AEN (Address Enable, Output)

Address Enable is an active high signal used to disable the system bus during DMA cycles to enable the output of the external latch which holds the upper byte of the address. Note that during DMA transfers HACK and AEN should be used to deselect all other I/O peripherals which may erroneously be accessed as programmed I/O during the DMA operation. The Am9517A automatically deselects itself by disabling the $\overline{CS}$ input during DMA transfers.

ADSTB (Address Strobe, Output)

The active high Address Strobe is used to strobe the upper address byte from DB0-DB7 into an external latch.

Figure 2. Am9517A Internal Registers.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	8 bits	1
Mask Register	6 bits	4
Request Register	4 bits	1
	4 bits	1

$\overline{MEMR}$ (Memory Read, Output)

The Memory Read signal is an active low three-state output used to access data from the selected memory location during a memory-to-peripheral or a memory-to-memory transfer.

$\overline{MEMW}$ (Memory Write, Output)

The Memory Write signal is an active low three-state output used to write data to the selected memory location during a peripheral-to-memory or a memory-to-memory transfer.

FUNCTIONAL DESCRIPTION

The Am9517A block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The Am9517A contains 344 bits of internal memory in the form of registers. Figure 2 lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

The Am9517A contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the Am9517A. The Program Command Control block decodes the various commands given to the Am9517A by the microprocessor prior to servicing a DMA Request. It also decodes each channel's Mode Control word. The Priority Encoder block resolves priority contention among DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input. In Am9080A systems this input will usually be the $\phi 2$ TTL clock from an Am8224. However, any appropriate system clock will suffice.

DMA Operation

The Am9517A is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The Am9517A can assume seven separate states, each composed of one full clock period. State I (SI) is the inactive state. It is entered when the Am9517A has no valid DMA requests pending. While in SI, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State 0 (S0) is the first state of a DMA service. The Am9517A has requested a hold but the processor has not yet returned an acknowledge. An acknowledge from the CPU will signal that transfers may begin. S1, S2, S3 and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted before S4 by the use of the Ready line on the Am9517A.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for each complete transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23 and S24) for the write-to-memory half of the transfer. The Temporary Data register is used for intermediate storage of the memory byte.

IDLE Cycle

When no channel is requesting service, the Am9517A will enter the Idle cycle and perform "SI" states. In this cycle the Am9517A will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample $\overline{CS}$, looking for an attempt by the microprocessor to write or read the internal registers of the Am9517A. When

$\overline{\text{CS}}$ is low and HACK is low, the Am9517A enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0-A3 are inputs to the device and select which registers will be read or written. The $\overline{\text{IOR}}$ and $\overline{\text{IOW}}$ lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip/flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip/flop is reset by Master Clear or Reset. A separate software command can also reset this flip/flop.

Special software commands can be executed by the Am9517A in the Program Condition. These commands are decoded as sets of addresses when both $\overline{\text{CS}}$ and $\overline{\text{IOW}}$ are active and do not make use of the data bus. Functions include Clear First/Last Flip/Flop and Master Clear.

ACTIVE CYCLE

When the Am9517A is in the Idle cycle and a channel requests a DMA service, the device will output a HREQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place, in one of four modes:

Single Transfer Mode: In Single Transfer mode, the Am9517A will make a one-byte transfer during each HREQ/HACK handshake. When DREQ goes active, HREQ will go active. After the CPU responds by driving HACK active, a one-byte transfer will take place. Following the transfer, HREQ will go inactive, the word count will be decremented and the address will be either incremented or decremented. When the word count goes to zero a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

To perform a single transfer, DREQ must be held active only until the corresponding DACK goes active. If DREQ is held continuously active, HREQ will go inactive following each transfer and then will go active again and a new one-byte transfer will be made following each rising edge of HACK. In 8080A/9080A systems this will ensure one full machine cycle of execution between DMA transfers. Details of timing between the Am9517A and other bus control protocols will depend upon the characteristics of the microprocessor involved.

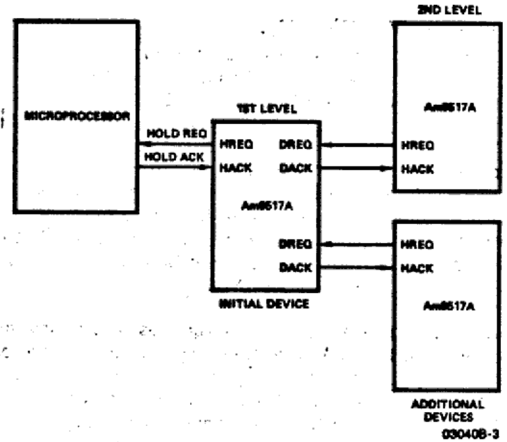
Block Transfer Mode: In Block Transfer mode, the Am9517A will continue making transfers until a TC (caused by the word count going to zero) or an external End of Process ($\overline{\text{EOP}}$) is encountered. DREQ need be held active only until DACK becomes active. An autoinitialize will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode: In Demand Transfer mode the device will continue making transfers until a TC or external $\overline{\text{EOP}}$ is encountered or until DREQ goes inactive. Thus, the device requesting service may discontinue transfers by bringing DREQ inactive. Service may be resumed by asserting an active DREQ once again. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count may be read from the Am9517A Current Address and Current Word Count registers. Autoinitialization will only occur following a TC or $\overline{\text{EOP}}$ at the end of service. Following Autoinitialization, an active-going DREQ edge is required to initiate a new DMA service.

Cascade Mode: This mode is used to cascade more than one Am9517A together for simple system expansion. The HREQ and HACK signals from the additional Am9517A are connected to the DREQ and DACK signals of a channel of the initial Am9517A. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel in the initial device is used only for prioritizing the additional device, it does not output any address or control signals of its own. These would conflict with the outputs of the active channel in the added device. The Am9517A will respond to DREQ with DACK but all other outputs except HREQ will be disabled.

Figure 3 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More Am9517As could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level devices forming a third level.

Figure 3. Cascaded Am9517As.



TRANSFER TYPES

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from an I/O device to the memory by activating $\overline{\text{IOR}}$ and $\overline{\text{MEMW}}$. Read transfers move data from memory to an I/O device by activating $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$. Verify transfers are pseudo transfers; the Am9517A operates as in Read or Write transfers generating addresses, responding to $\overline{\text{EOP}}$, etc., however, the memory and I/O control lines remain inactive.

Memory-to-Memory: The Am9517A includes a block move capability that allows blocks of data to be moved from one memory address space to another. When Bit C0 in the Command register is set to a logical 1, channels 0 and 1 will operate as memory-to-memory transfer channels. Channel 0 forms the source address and channel 1 forms the destination address. The channel 1 word count is used. A memory-to-memory transfer is initiated by setting a software DMA request for channel 0. Block Transfer Mode should be used for memory-to-memory. When channel 0 is programmed for a fixed source address, a single source word may be written into a block of memory.

When setting up the Am9517A for memory-to-memory operation, it is suggested that both channels 0 and 1 be masked out.

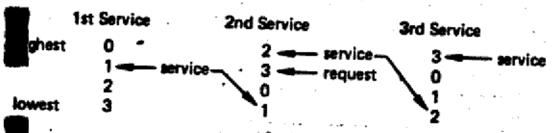
Further, the channel 0 word count should be initialized to the same value used in channel 1. No DACK outputs will be active during memory-to-memory transfers.

The Am9517A will respond to external $\overline{EOP}$ signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers may be found in Timing Diagram 4.

Autoinitialize: By programming a bit in the Mode register a channel may be set up for an Autoinitialize operation. During Autoinitialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word Count registers of that channel following $\overline{EOP}$. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not set by $\overline{EOP}$ when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to repeat its service without CPU intervention.

Priority: The Am9517A has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly. With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.



The priority encoder selects the highest priority channel requesting service on each active-going HACK edge. Once a channel is started, its operation will not be suspended if a request is received by a higher priority channel. The high priority channel will only gain control after the lower priority channel releases HREQ. When control is passed from one channel to another, the CPU will always gain bus control. This ensures generation of rising HACK edge to be used to initiate selection of the new highest-priority requesting channel.

Compressed Timing: In order to achieve even greater throughput where system characteristics permit, the Am9517A can compress the transfer time to two clock cycles. From Timing Diagram 3 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3 the read pulse width is made equal to the write pulse width and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur when A8-A15 need updating (see Address Generation). Timing for compressed transfers is found in Timing Diagram 6.

Extended Write: For Flyby Transactions late write is normally used, as this allows sufficient time for the $\overline{IOR}$ signal to get data from the peripheral onto the bus before $\overline{MEMW}$ is activated. In some systems, performance can be improved by starting the write cycle earlier. This is especially true for memory-to-memory transactions.

Address Generation: In order to reduce pin count, the Am9517A multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a 3-state enable. The lower order address bits are output by the Am9517A directly. Lines A0-A7 should be connected to the address bus. Timing Diagram 3 shows the time relationships between CLK, AEN, ADSTB, DB0-DB7 and A0-A7.

During Block and Demand Transfer mode services which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the Am9517A executes S1 states only when updating of A8-A15 in the latch is necessary. This means for long services, S1 states may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

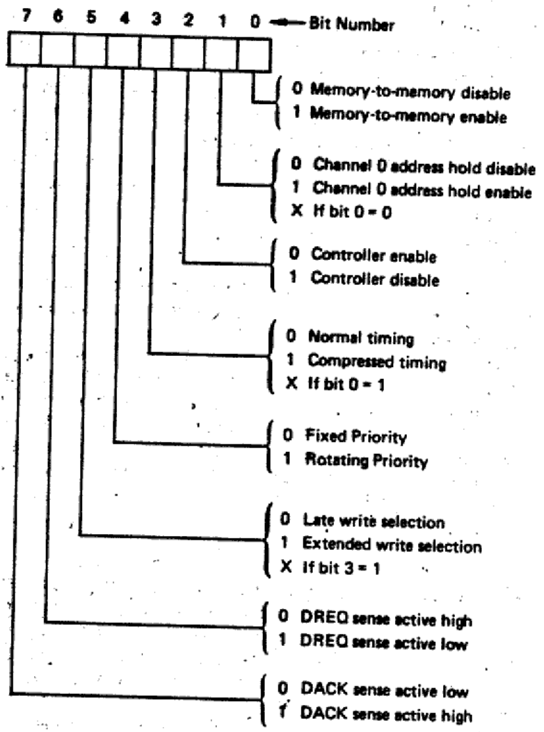
REGISTER DESCRIPTION

Current Address Register: Each channel has a 16-bit Current Address register. This register holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialization takes place only after an $\overline{EOP}$.

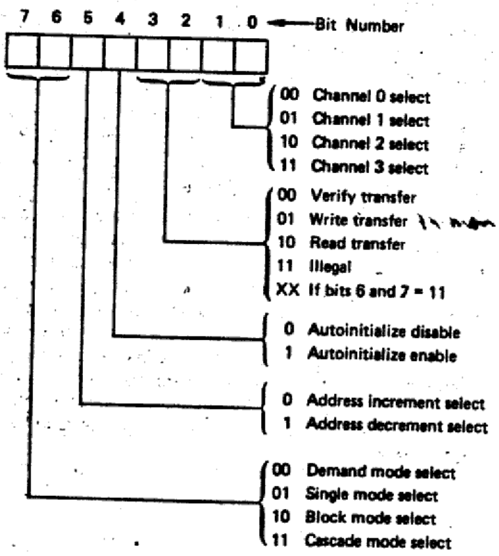
Current Word Count Register: Each channel has a 16-bit Current Word Count register. This register should be programmed with, and will return on a CPU read, a value one less than the number of words to be transferred. The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes to zero, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service it may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize can occur only when an $\overline{EOP}$ occurs. Note that the contents of the Word Count register will be FFFF (hex) following on internally generated $\overline{EOP}$.

Base Address and Base Word Count Registers: Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original values of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes during DMA programming by the microprocessor. Accordingly, writing to these registers when intermediate values are in the Current registers will overwrite the intermediate values. The Base registers cannot be read by the microprocessor.

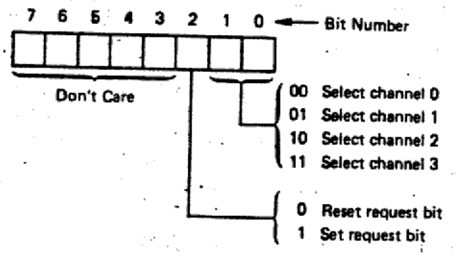
Command Register: This 8-bit register controls the operation of the Am9517A. It is programmed by the microprocessor in the Program Condition and is cleared by Reset. The following table lists the function of the command bits. See Figure 4 for address coding.



Mode Register: Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register it to be written.

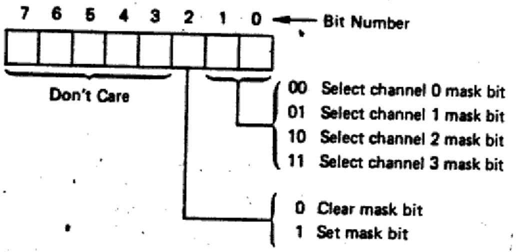


Request Register: The Am9517A can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are nonmaskable and subject to prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external EOP. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 4 for address coding.

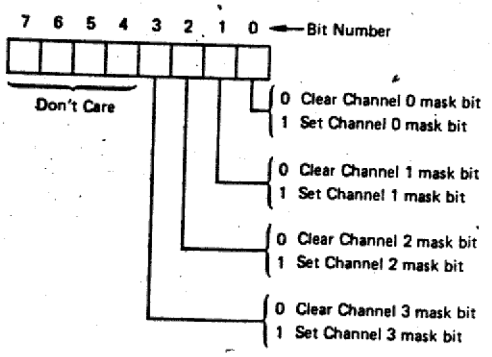


Software requests will be serviced only if the channel is in Block mode. When initiating a memory-to-memory transfer, the software request for channel 0 should be set.

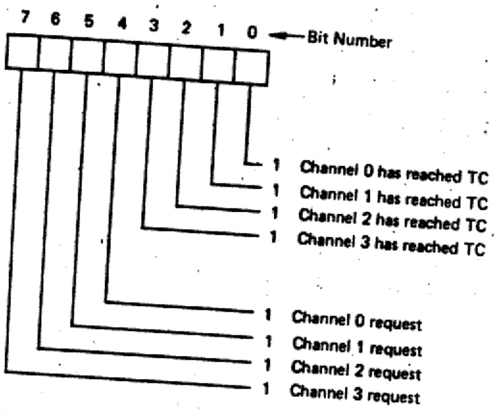
Mask Register: Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an EOP if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 4 for instruction addressing.


7 6 5 4 3 2 1 0 ← Bit Number
Don't Care
00 Select channel 0 mask bit
01 Select channel 1 mask bit
10 Select channel 2 mask bit
11 Select channel 3 mask bit
0 Clear mask bit
1 Set mask bit

All four bits of the Mask Register may also be written with a single command.


7 6 5 4 3 2 1 0 ← Bit Number
Don't Care
0 Clear Channel 0 mask bit
1 Set Channel 0 mask bit
0 Clear Channel 1 mask bit
1 Set Channel 1 mask bit
0 Clear Channel 2 mask bit
1 Set Channel 2 mask bit
0 Clear Channel 3 mask bit
1 Set Channel 3 mask bit

Status Register: The Status registers may be read out of the Am9517A by the microprocessor. It indicates which channels have reached a terminal count and which channels have pending DMA requests. Bits 0-3 are set each time a TC is reached by that channel, including after each Autoinitialization. These bits are cleared by Reset and each Status Read. Bits 4-7 are set whenever their corresponding channel is requesting service.


7 6 5 4 3 2 1 0
Bit Number
1 Channel 0 has reached TC
1 Channel 1 has reached TC
1 Channel 2 has reached TC
1 Channel 3 has reached TC
1 Channel 0 request
1 Channel 1 request
1 Channel 2 request
1 Channel 3 request

Temporary Register: The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands: There are two special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The two software commands are:

Clear First/Last Flip/Flop: This command may be issued prior to writing or reading Am9517A address or word count information. This initializes the Flip/Flop to a known state so that subsequent accesses to register contents by the microprocessor will address lower and upper bytes in the correct sequence. When the Flip/Flop is cleared it addresses the lower byte and when set it addresses the upper byte.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary and Internal First/Last Flip/Flop registers are cleared and the Mask register is set. The Am9517A will enter the Idle cycle.

Figure 4 lists the address codes for the software commands.

Figure 4. Register and Function Addressing.

Interface Signals						Operation
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Illegal
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 5. Word Count and Address Register Command Codes.

Channel	Register	Operation	Signals							Internal Flip/Flop	Data Bus DB0–DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base & Current Address	Write	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	0	0	1	0	W0-W7
			0	1	0	0	0	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0-W7
			0	0	1	0	0	0	1	1	W8-W15
1	Base & Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	0	1	1	0	W0-W7
			0	1	0	0	0	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0-W7
			0	0	1	0	0	1	1	1	W8-W15
2	Base & Current Address	Write	0	1	0	0	1	0	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	1	0	1	0	W0-W7
			0	1	0	0	1	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0-W7
			0	0	1	0	1	0	1	1	W8-W15
3	Base & Current Address	Write	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0-W7
			0	0	1	0	1	1	1	1	W8-W15

MAXIMUM RATINGS above which useful life may be impaired	**Am9517A**
Storage Temperature	−65 to +150°C
VCC with Respect to VSS	−0.5 to +7.0V
All Signal Voltages with Respect to VSS	−0.5V to +7.0V
Power Dissipation (Package Limitation)	1.5W

The products described by this specification include internal circuitry designed to protect input devices from damaging accumulations of static charge. It is suggested, nevertheless, that conventional precautions be observed during storage, handling and use in order to avoid exposure to excessive voltages.

OPERATING RANGE

Part Number	T_A	V_{CC}
Am9517ADC/PC	0 to +70°C	5.0V ±5%
Am9517A-1DC/PC	0 to +70°C	5.0V ±5%
Am9517A-4DC/PC	0 to +70°C	5.0V ±5%
Am9517A-5DC/PC	0 to +70°C	5.0V ±5%
Am9517ADI	−40 to +85°C	5.0V ±10%
Am9517A-1DI	−40 to +85°C	5.0V ±10%
Am9517A-4DI	−40 to +85°C	5.0V ±10%
Am9517ADMB	−55 to +125°C	5.0V ±10%

ELECTRICAL CHARACTERISTICS over operating range (Note 1)

Parameter	Description	Test Conditions	Min	Typ	Max	Unit
VOH	Output HIGH Voltage	IOH = −200µA	2.4			Volts
		IOH = −100µA, (HREQ Only)	3.3			
VOL	Output LOW Voltage	IOL = 3.2mA			0.45	Volts
VIH	Input HIGH Voltage		2.0		VCC+0.5	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC	−10		+10	µA
IOZ	Output Leakage Current	VCC ≤ VO ≤ VSS+.40	−10		+10	µA
ICC	VCC Supply Current	T_A = +25°C		65	130	mA
		T_A = 0°C		75	150	
		T_A = −55°C			175	
CO	Output Capacitance	fc = 1.0MHz, Inputs = 0V		4	8	pF
CI	Input Capacitance			8	15	pF
CIO	I/O Capacitance			10	18	pF

NOTES:

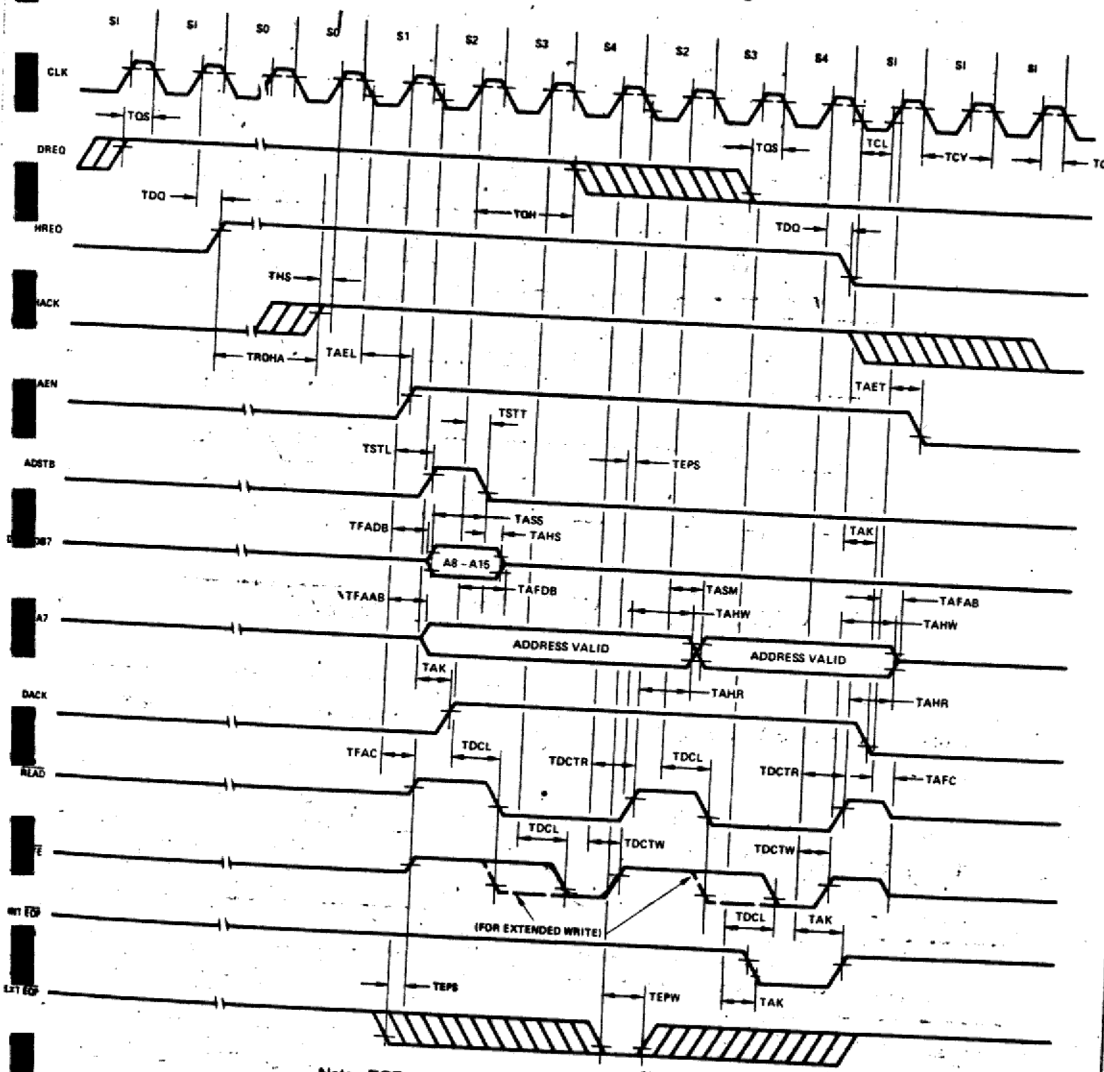
1. Typical values are for T_A = 25°C, nominal supply voltage and nominal processing parameters.
2. Input timing parameters assume transition times of 20ns or less. Waveform measurement points for both input and output signals are 2.0V for High and 0.8V for Low, unless otherwise noted.
3. Output loading is 1 Standard TTL gate plus 50pF capacitance unless noted otherwise.
4. The new $\overline{IOW}$ or $\overline{MEMW}$ pulse width for normal write will be TCY-100ns and for extended write will be 2TCY-100ns. The net $\overline{IOR}$ or $\overline{MEMR}$ pulse width for normal read will be 2TCY-50ns and for compressed read will be TCY-50ns.
5. TDQ is specified for two different output HIGH levels. TDQ1 is measured at 2.0V. TDQ2 is measured at 3.3V. The value for TDQ2 assumes an external 3.3kΩ pull-up resistor connected from HREQ to VCC.
6. DREQ should be held active until DACK is returned.
7. DREQ and DACK signals may be active high or active low. Timing diagrams assume the active high mode.
8. Output loading on the data bus is 1 Standard TTL gate plus 15pF for the minimum value and 1 Standard TTL gate plus 100pF for the maximum value.
9. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600ns for the Am9517A or Am9517A-1, at least 450ns for the Am9517A-4 and 400ns for the Am9517A-5 as recovery time between active read or write pulses.
10. Parameters are listed in alphabetical order.
11. Pin 5 is an input that should always be at a logic high level. An internal pull-up resistor will establish a logic high when the pin is left floating. Alternatively, pin 5 may be tied to VCC.
12. Signals $\overline{READ}$ and $\overline{WRITE}$ refer to $\overline{IOR}$ and $\overline{MEMW}$ respectively for peripheral-to-memory DMA operations and to $\overline{MEMR}$ and $\overline{IOW}$ respectively for memory-to-peripheral DMA operations.
13. If N wait states are added during the write-to-memory half of a memory-to-memory transfer, this parameter will increase by N (TCY).

Am9517A
SWITCHING CHARACTERISTICS
ACTIVE CYCLE (Notes 2, 3, 10, 11 and 12)

Parameters	Description	Am9517A Min	Am9517A Max	Am9517A-1 Min	Am9517A-1 Max	Am9517A-4 Min	Am9517A-4 Max	Am9517A-5 Min	Am9517A-5 Max	Units
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		300		300		225		200	ns
TAET	AEN LOW from CLK HIGH (S1) Delay Time		200		200		150		130	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		150		150		120		90	ns
TAFC	$\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ Float from CLK HIGH		150		150		120		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		250		250		190		170	ns
TAHR	ADR from $\overline{\text{READ}}$ HIGH Hold Time	TCY-100		TCY-100		TCY-100		TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	50		50		40		30		ns
TAHW	ADR from WRITE HIGH Hold Time	TCY-50		TCY-50		TCY-50		TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time		280		280		220		170	ns
	$\overline{\text{EOP}}$ HIGH from CLK HIGH Delay Time		250		250		190		170	ns
	$\overline{\text{EOP}}$ LOW to CLK HIGH Delay Time		250		250		190		100	ns
TASM	ADR Stable from CLK HIGH		250		250		190		170	ns
TASS	DB to ADSTB LOW Setup Time	100		100		100		100		ns
TCH	Clock High Time (Transitions ≤ 10ns)	120		120		100		80		ns
TCL	Clock Low Time (Transitions ≤ 10ns)	150		150		110		68		ns
TCY	CLK Cycle Time	320		320		250		200		ns
TDCL	CLK HIGH to $\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ LOW Delay (Note 4)		270		270		200		190	ns
TDCTR	$\overline{\text{Read}}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		270		270		210		190	ns
TDCTW	$\overline{\text{WRITE}}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		200		200		150		130	ns
TDQ1	HREQ Valid from CLK HIGH Delay Time (Note 5)		160		160		120		120	ns
TDQ2			250		250		190		120	ns
TEPS	$\overline{\text{EOP}}$ LOW from CLK LOW Setup Time	60		60		45		40		ns
TEPW	$\overline{\text{EOP}}$ Pulse Width	300		300		225		220		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		250		250		190		170	ns
TFAC	$\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ Active from CLK HIGH		200		200		150		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		300		300		225		200	ns
THS	HACK Valid to CLK HIGH Setup Time	100		100		75		75		ns
TIDH	Input Data from $\overline{\text{MEMR}}$ HIGH Hold Time	0		0		0		0		ns
TIDS	Input Data to $\overline{\text{MEMR}}$ HIGH Setup Time	250		250		190		170		ns
TODH	Output Data from $\overline{\text{MEMW}}$ HIGH Hold Time	20		20		20		10		ns
TODV	Output Data Valid to $\overline{\text{MEMW}}$ HIGH (Note 13)	200		200		125		125		ns
TQS	DREQ to CLK LOW (S1, S4) Setup Time	120		0		0		0		ns
TRH	CLK to READY LOW Hold Time	20		20		20		20		ns
TRS	READY to CLK LOW Setup Time	100		100		60		60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		200		200		150		130	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		140		140		110		90	ns
TQH	DREQ from DACK Valid Hold Time	0		0		0		0		ns
TRQHA	HREQ to HACK Delay Time	1		1		1		1		clk

SWITCHING WAVEFORMS

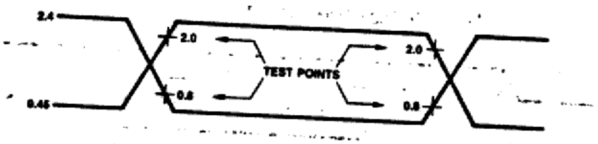
Timing Diagram 1. Active Cycle Timing Diagram



Note: EOP must precede AEN in single transfer mode.

03040B-4

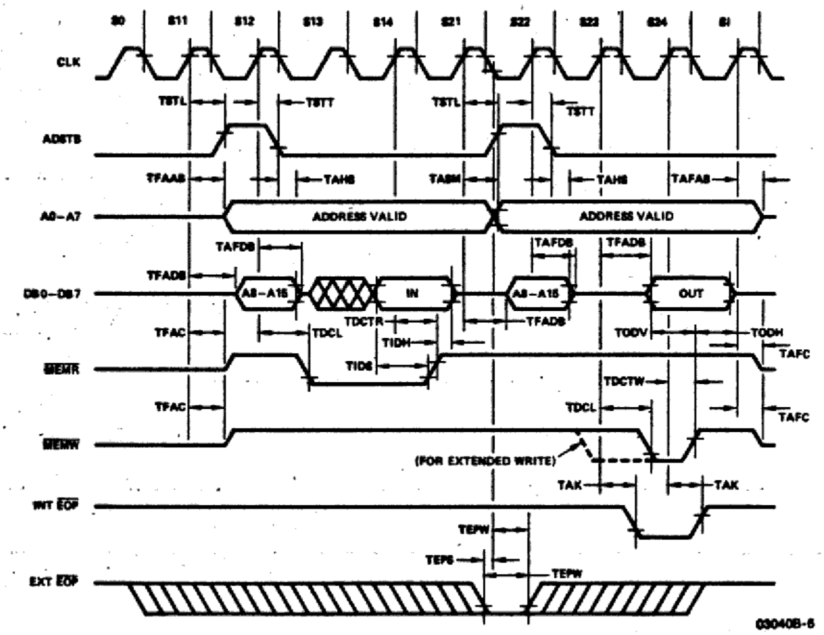
INPUT WAVEFORMS FOR AC TESTS



03040B-5

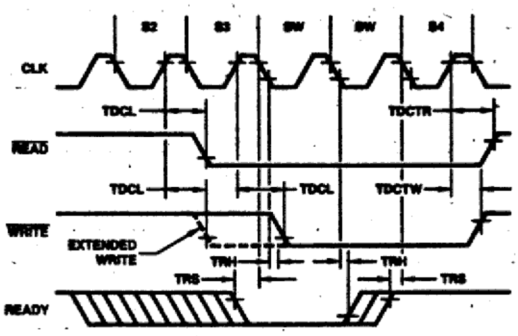
SWITCHING WAVEFORMS (Cont.)

Timing Diagram 2. Memory-to-Memory



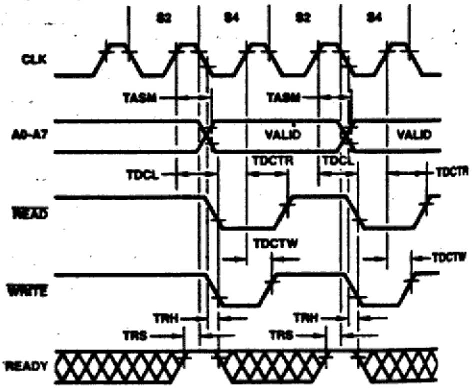
03040B-6

Timing Diagram 3. Ready Timing



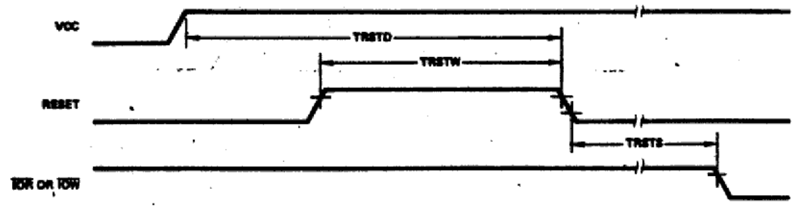
03040B-7

Timing Diagram 4. Compressed Timing



03040B-8

Timing Diagram 5. Reset Timing



03040B-9

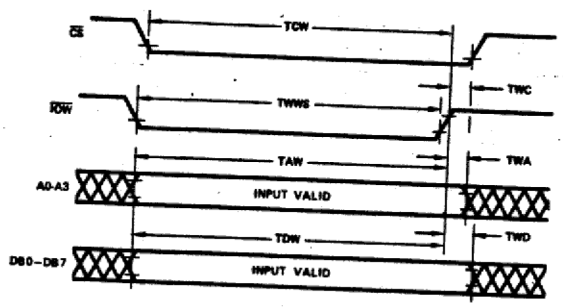
SWITCHING CHARACTERISTICS

PROGRAM CONDITION (IDLE CYCLE)
(Notes 2, 3, 10, and 11)

Parameters	Description	Am9517A Min	Am9517A Max	Am9517A-1 Min	Am9517A-1 Max	Am9517A-4 Min	Am9517A-4 Max	Am9517A-5 Min	Am9517A-5 Max	Units
TAR	ADR Valid or $\overline{CS}$ LOW to $\overline{READ}$ LOW	50		50		50		50		ns
TAW	ADR Valid to $\overline{WRITE}$ HIGH Setup Time	200		200		150		130		ns
TCW	CS LOW to $\overline{WRITE}$ HIGH Setup Time	200		200		150		130		ns
TDW	Data Valid to $\overline{WRITE}$ HIGH Setup Time	200		200		150		130		ns
TRA	ADR or CS Hold from $\overline{READ}$ HIGH	0		0		0		0		ns
TRDE	Data Access from $\overline{READ}$ LOW (Note 8)		300		200		200		140	ns
TRDF	DB Float Delay from $\overline{READ}$ HIGH	20	150	20	100	20	100	0	70	ns
TRSTD	Power Supply HIGH to RESET LOW Setup Time	500		500		500		500		ns
TRSTS	RESET to First $\overline{IOWR}$	2TCY		2TCY		2TCY		2TCY		ns
TRSTW	RESET Pulse Width	300		300		300		300		ns
TRW	$\overline{READ}$ Width	300		300		250		200		ns
TWA	ADR from $\overline{WRITE}$ HIGH Hold Time	20		20		20		20		ns
TWC	CS HIGH from $\overline{WRITE}$ HIGH Hold Time	20		20		20		20		ns
TWD	Data from $\overline{WRITE}$ HIGH Hold Time	30		30		30		30		ns
TWWS	Write Width	200		200		200		160		ns

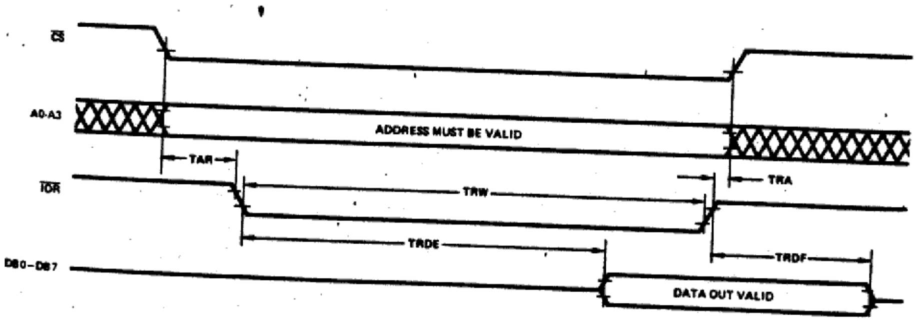
SWITCHING WAVEFORMS (Cont.)

Timing Diagram 6. Program Condition Write Timing (Note 9)



03040B-10

Timing Diagram 7. Program Condition Read Cycle (Note 9)

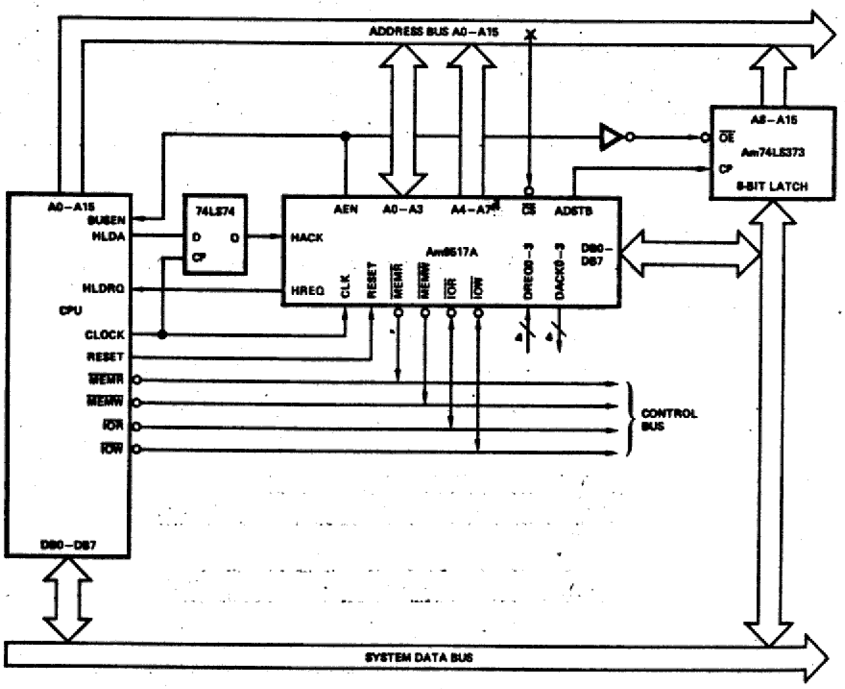


03040B-11

APPLICATION INFORMATION

Figure 6 shows a convenient method for configuring a DMA system with the Am9517A Controller and a microprocessor system. The Multimode DMA Controller issues a Hold Request to the processor whenever there is at least one valid DMA Request from a peripheral device. When the processor replies with a Hold Acknowledge signal, the Am9517A takes control of the Address Bus, the Data Bus and the Control Bus. The address for the first transfer operation comes out in two bytes – the least significant eight bits on the eight Address outputs and the most significant eight bits on the Data Bus. The contents of the Data Bus are then latched into the Am74LS373 register to complete the full 16 bits of the Address Bus. The Am74LS373 is a high speed, low power, 8-bit, 3-state register in a 20-pin package. After the initial transfer takes place, the register is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one Am9517A is used.

Figure 6. Basic DMA Configuration.



03040B 12

Field Change Note for the 8509 module.
level 0 date 26-11-1982.

document date
24-07-1984

Subject: 16Mhz clock.

The 8509 module get the 16Mhz clock from terminal A48 and terminal B48, since the 16Mhz clock is only present on terminal B48, it is recommended that the connection to the A48 tertminal is cut, to lessen the load on the clock signal.

TWJ.

MODULE 8509 SOLDER SIDE