"miroCPU186"

iAPX186-CPU Slave-Europakarte

Handbuch

Version: 1.3a
Stand: 22-Mar-1984

miro - Datensysteme GmbH
Madamenweg 162
D-3300 Braunschweig
Tel: 0531 - 81058

I N H A L T S V E R Z E I C H N I S

Anhang

Warenzeichen:
CP/M, MP/M, CP/M-86 und CP/M Graphics sind eingetragene Warenzeichen der Firma Digital Research, Pacific Grove, California, USA.
Multibus und iRMX sind eingetragene Warenzeichen der Firma Intel Corp., Santa Clara, California, USA.
WordStar ist ein eingetragenes Warenzeichen der Firma MicroPro International Corp., San Rafael, California, USA.

Verzichtsleistung:
Bezüglich des Inhalts dieses Handbuchs und gegenüber jeglicher auferlegter Händler- oder Eignungsgarantie für besondere Zwecke übernimmt miro Datensysteme keinerlei Haftung oder Garantie. Weiterhin behält sich miro Datensysteme das Recht vor, das Handbuch zu überarbeiten und in geeigneten Zeitabständen Änderungen des Inhalts vorzunehmen, ohne die Verpflichtung, irgendeine Person oder Organisation von einer derartigen Revision zu benachrichtigen.

1. Kurzbeschreibung

Die miroCPU186 wurde zum Betrieb als passive "Slave-CPU"-Baugruppe am ECB-Systembus entwickelt. Die Kopplung mit der "Master-CPU" des ECB-Systembus erfolgt durch einen 128 Byte FIFO-Buffer als zweiseitiges I/O-Port, das nur je zwei Adressen im I/O-Adreßbereich von Master und Slave belegt. Die mit dem ECB-Systembus kompatible Einfach-Europakarte ermöglicht auf einfache Weise den Aufbau eines modularen Multi-Mikrocomputer-Systems mit einem Z80-Prozessor und einem oder mehreren 80186-Prozessoren im "Master-Slave"-Betrieb. Da der bzw. die "Slave"-Mikrocomputer miroCPU186 über genügend große lokale Programm- und Datenspeicher verfügen, können auch komplexe Programme oder Programm-Module vom "Master" in den bzw. die "Slaves" verlagert werden und in diesen nun völlig autonom und zeitlich echt parallel bearbeitet werden.

Der Einsatz der 8 MHz iAPX 186 CPU erlaubt eine äußerst leistungsfähige und schnelle 16-Bit Datenverarbeitung durch die "Slave"-Module miroCPU186. Der Betrieb der miroCPU186 als eigenständige CPU-Baugruppe ist ebenfalls möglich ("stand-alone"-Betrieb).

Als zentrale Mikroprozessoren verfügen "Master"-CPU-Baugruppen wie z.B. Z80-Zentraleinheiten für den ECB-Systembus uneingeschränkt über den Systembus, während "Slave"-CPU-Baugruppen wie die miroCPU186 von "Master"-CPU-Baugruppen gesteuert werden, sie belegen nicht den Systembus und ermöglichen so echten Parallelbetrieb mehrerer CPU's. Nicht multiprozessorfähige Baugruppen wie Z80-Zentraleinheiten dagegen erlauben nur Zeitmultiplexbetrieb mit anderen aktiven Bausteinen wie z.B. DMA-Baugruppen, indem stets nur eine einzige CPU oder DMA aktiv ist und alle anderen in den "Hold-Mode" geschaltet werden.

"Slave"-CPU-Baugruppen wie die miroCPU186 besitzen genügend lokale Speicher und serielle Schnittstellen und können daher in echtem Simultanbetrieb beliebig komplexe Aufgaben bearbeiten. Durch das Verlagern von Programmteilen des zentralen Mikroprozessors ("Master") in die Peripherie-Mikrocomputer ("Slaves") wird aus einem zentralisierten Mikrocomputersystem ein dezentralisiertes Multi-Mikrocomputersystem mit allen seinen Vorteilen. Die Verteilung der Aufgaben zwischen "Master" und "Slaves" ist frei wählbar; komplexe und zeitkritische Berechnungen, Programm-Module ("Tasks") oder ganze Prozesse können leistungsfähigen "Slave"-Modulen übertragen werden, während das "Master"-Modul nur noch Steuerungs-, Überwachungs- und Ein-/Ausgabe-Aufgaben ausführt und so die Funktion eines Peripherie-Prozessors, also eigentlich eines "Slaves" übernimmt.

Besitzen die "Slave"-Module wie die miroCPU186 genügend eigenen Speicherraum und Ein-/Ausgabeperipherie, ist ein direkter Zugriff zu den Speicher- und Ein-/Ausgabebausteinen am Systembus nicht erforderlich und die Kommunikation zum "Master" kann durch eine lose Kopplung der nun autonomen Bussysteme von "Master" und "Slaves" realisiert werden. Bei dieser Kopplungsart ist eine Multiprozessorfähigkeit des

Systembus nicht zwingend nötig, so daß auch einfachere Systeme wie der (nicht erweiterte) ECB-Bus mit mehreren "Slave"-Prozessoren ausgestattet werden können.

Da die "Slave"-Prozessoren den Systembus nicht belegen, muß die Z80 -"Master-CPU" nun auch nicht über /BUSRQ vom Bus in den "hold-mode", also abgeschaltet und so in ihrer Aktivität blockiert werden, wie dies z.B. für DMA-Bausteine am ECB-Bus der Fall ist; die Benutzung eines gemeinsamen Bussystems durch eine CPU, DMA-Bausteine oder aktive Subprozessoren erzwingt eine Aufteilung der zur Verfügung stehenden Buszeit im Multiplexbetrieb und eine Prioritätsregelung z.B. durch eine BAI/BAO-Kette oder durch einen sog. "Bus Arbiter" für die Buszuteilung.

Die miroCPU186-Baugruppe verfügt über ein eigenes 16-Bit-Bussystem "on-board" und kann daher unabhängig von der "Master"-CPU des ECB-Bussystems im echten Simultanbetrieb völlig autonom und auch zeitlich echt parallel arbeiten; der Systembus wird nicht mehr zur Engpaßstelle des Gesamtsystems. Zeitkritische Operationen können ohne Wartezeiten auf Zugriff zum Systembus ausgeführt und die Effizienz und Geschwindigkeit der Datenverarbeitung erheblich gesteigert werden ohne die sonst übliche überproportionale Kostensteigerung.

Die Kommunikation zwischen "Master"- und "Slave"-Mikrocomputer-Modulen erfolgt asynchron zum Systemtakt durch eine einfache "Handshake"-Prozedur per Interrupt oder Softwareflags, der "Slave"-interne Takt ist unabhängig von den Zeitanforderungen auf dem Systembus. Schnelle Speichertransfers von oder zu den "Slave-CPU"-Baugruppen können wahlweise mit beidseitiger DMA-Unterstützung durchgeführt werden. "Slave-CPU"-Baugruppen können von der "Master-CPU" via Systembus gestartet werden (Software-Reset) oder in der laufenden Programmbearbeitung unterbrochen werden (Software-NMI).

Die Schnittstelle besteht aus einem bidirektional ("dual ported") arbeitendem 128 Byte FIFO ("first in - first out") Pufferspeicher, der voll interruptfähig ist ("daisy-chain"-Logik, Z80 Interrupt Mode 2), und als zweiseitiges I/O-Port im I/O-Adreßbereich des "Masters" und der "Slaves" liegt, so daß keine Speicheradressen besetzt werden.

Kommunikationswünsche untereinander bzw. Fertigmeldungen der "Slave"-CPU's werden der "Master"-CPU via Systembus durch Interrupts und/oder Status-Bits ("Polling") übermittelt; der Transport der Datenbytes erfolgt sequentiell über den FIFO Pufferspeicher mit einer max. Übertragungsrate von 800 kByte/s. Zusätzlich ist der Austausch von Kontroll-Parametern ohne Benutzung des FIFO Pufferspeicher über sog. "mailbox"- oder "message"-Register möglich; ausschließlich die lokale CPU kann in ihre "mailbox" schreiben, alle anderen können nur daraus lesen. Jede miroCPU186-Baugruppe verfügt über zwei "mailbox"-Register für die Kommunikation mit der Z80 "Master-CPU"; das eine Register kann nur von der 80186 CPU, das andere nur von der Z80 CPU beschrieben werden, während beide Register von beiden CPU's gelesen werden können. Übertragungswünsche von Kontroll-Parametern können gegenseitig ebenfalls durch Interrupts übermittelt werden.

2. Technische Daten

Version:
1.3 vom 16-Mar-1983 (0.0 vom 12-Apr-1983)

Leiterplatte:
Einfach-Europaformat 100 x 160 mm nach DIN 41494, Teil 2,
Multilayer, galvanisch verzinnt, Lötstopplack

Bus-Stecker:
64-polige VG-Messerleiste nach DIN 41612, Teil 2,
Bauform 'C' (Reihen a & c bestückt)

Busbelegung:
ECB-Bus (Standard-Version), voll gepuffert;
voll interruptfähig: "daisy-chain"-Logik, Z80 Interrupt Mode 2

Interface-Stecker:
Serielle Schnittstellen: 2 x 16-polige Stiftleisten
Piggy-Pack Erweiterung: 2 x 40-polige Buchsenleisten

Stromversorgung:
+5V, +- 5%; 1.7A typ.
+12V, 18mA typ. (nur für RS 232C)
-12V, 16mA typ. (nur für RS 232C)

Bus-Eingänge (Schmitt-Trigger Puffer, Datenbus):
Vih(min) = 2.0V, Vil(max) = 0.8V
Iih(max) = 0.02mA, Iil(max) = 0.2mA

Bus-Ausgänge (Tri-State Puffer, Datenbus):
Voh(min) = 2.4V, Vol(max) = 0.4V
Ioh(max) = 15mA, Iol(max) = 24mA

Basisadresse (I/O auf Systembus):
A7, A6, A5, A4, A3, A2, A1 einstellbar, standard 50h

Adreßbereich (I/O auf Systembus):
Basisadresse + 0 ... Basisadresse + 1, standard 50h...51h

CPU:
Prozessor: Intel iAPX 186
Taktfrequenz: 8 MHz (5 MHz optional)
Zykluszeit: 125 ns
Befehlsausführungszeit: min. 500 ns
Adressierbarer Speicherbereich: 1 MByte (1048576 Bytes)
Adressierbarer I/O-Adreßbereich: 64 kBytes (65536 Bytes)
3 interne Timer (2 mit TTL-kompatiblen Ein-/Ausgängen)
2 interne DMA-Kanäle
Interner Interrupt-Controller mit 5 externen Ein-/Ausgängen

Interner 256 Byte Kontrollblock (standard):
FF00h: control register, I/O-mapped

Interne Chip-Select Register (standard):
FFA0h: UMCS register
FFA2h: LMCS register
FFA4h: PACS register
FFA6h: MMCS register
FFA8h: MPCS register

Interner Timer 0 (standard):
FF52h: timer 0 max. count A
FF54h: timer 0 max. count B
FF56h: timer 0 mode register

Interner Timer 1 (standard):
FF5Ah: timer 1 max. count A
FF5Ch: timer 1 max. count B
FF5Eh: timer 1 mode register

Interner Timer 2 (standard):
FF62h: timer 2 max. count A
FF66h: timer 2 mode register

Interner Interrupt-Controller (standard):
FF20h...FF3Eh

Interner DMA (standard):
FFC0h...FFCAh: DMA Descriptors Channel 0
FFD0h...FFDAh: DMA Descriptors Channel 1

Internes Relocation Register (standard):
FFFEh

Schreib-/Lese-Speicher (RAM):

Standardmäßige Bestückung:	128 kByte (16 x 64 kBit DRAMs)
Optionale Bestückung:	512 kByte (16 x 256 kBit DRAMs)
Speichererweiterung:	max. 1 MByte durch Piggy-Pack
Zugriffszeit:	max. 200 ns (ohne Wartezyklen)

Festwertspeicher (EPROM):

Einsetzbare EPROM-Typen:	2732, 2764, 27128, 27256, 27512
Anzahl der Steckplätze:	2 (für 16-Bit Datenbus)
Standardmäßige Bestückung:	16 kByte (2 x 2764 EPROMs)
Maximale EPROM-Kapazität:	128 kByte (2 x 27512 EPROMs)
Zugriffszeit:	max. 250 ns (ohne Wartezyklen)
OE high to Output Float:	max. 85 ns (TDF)

Interne DRAM-Speicheradressierung (on-board, programmierbar):
64 kBit DRAMs: 00000h...1FFFFh (standard)
256 kBit DRAMs: 00000h...7FFFFh (optional)

Refresh-Adreßbereich (standard):
80000h...8FFFFh, /MCS0 chip-select

Interne EPROM-Speicheradressierung (on-board, programmierbar):
32 kBit EPROMs, 2 x 2732, 8 kByte: FE000h...FFFFFh
64 kBit EPROMs, 2 x 2764, 16 kByte: FC000h...FFFFFh
128 kBit EPROMs, 2 x 27128, 32 kByte: F8000h...FFFFFh
256 kBit EPROMs, 2 x 27256, 64 kByte: F0000h...FFFFFh
512 kBit EPROMs, 2 x 27512, 128 kByte: E0000h...FFFFFh

Memory Piggy-Pack:
Maximal 512 kByte mit 16 x 256 kBit DRAMs
Adressierung: 80000h...FFFFFh
Durch Einsatz des Memory Picky-Packs wird der gesamte Speicherbereich von 1 MByte ausgenutzt.
Die EPROMs werden ausgeblendet, d.h. beim Einsatz von z.B. 32 kBit EPROMs ist der Speicherbereich von FE000h bis FFFFFh nicht durch den Schreib-/Lesespeicher benutzbar.

Interne I/O-Adressierung (on-board, programmierbar):
0000h: Basis-Adresse für Peripherie-Bausteine,
standard, I/O-mapped, PBA = peripheral base address

Seriell I/O:
Multi Protocol Serial Controller 8274 MPSC;
Zwei asynchrone oder synchrone serielle Schnittstellen;
Interne oder externe Takterzeugung;
Baudraten (mit internem Takt) von DC bis 880 kBaud programmierbar;
I/O-Port Adressierung (on-board, programmierbar, standard):
0000h: 8274 data port channel a
0004h: 8274 status port channel a
0002h: 8274 data port channel b
0006h: 8274 status port channel b

FIFO I/O-Interface:
FIFO Input/Output Interface Unit Z8038-FIO;
I/O-Port Adressierung (on-board, programmierbar, standard):
0200h: Z8038 FIO data register
0202h: Z8038 FIO status register

3. Einsatzgebiete

Auf dem weitverbreiteten ECB-Bus ist durch den Einsatz einer miroCPU186-Baugruppe zusätzlich zur Z80/8080-Software auch das wachsende 8086/8088/80186-Softwareangebot mit entsprechend erhöhtem Durchsatz ablauffähig. Die Z80 "Master"-CPU wird von zeitkritischen Abläufen entlastet, wobei Speicher- und Programmverwaltungsaufgaben bei einfacheren Überwachungsmöglichkeiten relativ reduziert werden.

3.1. CP/M-86 Implementation auf ECB-Bus

Die "Slave-CPU's" können intern unter CP/M-86 (auch MS-DOS oder Concurrent-CP/M) betrieben werden, wobei das gesamte Programmangebot zu diesem Betriebssystem zur Verfügung steht. Die Umschaltung zwischen CP/M-80 (ausgeführt von der "Master-CPU") und CP/M-86 ("Slave-CPU") erfolgt automatisch. Console-I/O erfolgt wahlweise über die Treiber im CP/M-80 der "Master-CPU" oder die seriellen Schnittstellen der "Slave-CPU's". Diskettenzugriffe und sonstige Ein-/Ausgabe werden über die ohnehin vorhandenen BIOS-Routinen der "Master-CPU" (CP/M-80) abgewickelt. Damit ist das gesamte Ein-/Ausgabe-Handling vollkommen hardwareunabhängig ohne Komforteinbußen.

Die Aktionen des Anwenders beschränken sich auf das Einstecken der "Slave-CPU"-Baugruppe(n) und das Anstarten eines mitgelieferten Initialisierungsprogramms. Einzige Voraussetzung ist ein installiertes CP/M-80 System.

Aufgrund der Aufwärtskompatibilität des iAPX 186 zum 8086 kann bestehende 8086-Software ohne jede Änderung übernommen werden. Damit auch die neuen Befehle des iAPX186 verwendbar sind, werden der Assembler ASM-86 sowie die Compiler für die höheren Programmiersprachen PLM-86, PASCAL-86 und FORTRAN-86 in neuen Versionen zur Verfügung gestellt.

3.2. Parallelrechner

Durch den Einsatz mehrerer "Slave-CPU's" können parallelisierbare Programme, auch in strukturierten höheren Programmiersprachen (PASCAL, ADA usw.), echt parallel abgearbeitet werden. Der Datendurchsatz erhöht sich in etwa proportional mit der Anzahl der "Slave-CPU's" bis in die Reichweite von Großrechnern ohne die sonst übliche überproportionale Kostensteigerung.

4. Funktionsbeschreibung

4.1. Timer

Auf dem 80186-Chip bereits integriert sind drei unabhängige Timer (programmierbar); Timer 0 und Timer 1 besitzen je zwei Register für den maximalen Zählwert, die Ein- und Ausgänge sind extern verfügbar, während Timer 2 intern den 1/4 CPU-Clock als Takt verwendet und über nur ein Register für den maximalen Zählwert verfügt; Timer 2 kann als Vorteiler für die Timer 0 und 1 verwendet werden oder zyklisch interne DMA-Anforderungen generieren; Ein- und Ausgänge für Timer 2 sind nicht vorhanden.

4.2. Interrupt-Controller

Auf dem 80186-Chip bereits integriert ist auch ein Interrupt-Controller für vier maskierbare externe Interrupt-Anforderungen über die Leitungen INT0 bis INT3, den nicht maskierbaren Interrupt NMI sowie für die internen Interruptquellen Timer 0 bis Timer 3 und die DMA-Kanäle 0 und 1.

4.3. Wartezyklen-Generator

Alle Chip-Select Leitungen (CS) können unabhängig voneinander mit 0-3 Wartezyklen (programmierbar) belegt werden; zusätzlich ist programmierbar, ob die externen Ready-Leitungen ignoriert werden oder nicht, so daß von außerhalb beliebig viele Wartezyklen für langsame Peripheriebausteine eingefügt werden können.

4.4. DMA-Einheit

Auf dem 80186-Chip bereits integriert sind zwei unabhängige schnelle (max. 2 MBytes/s) DMA-Kanäle. Timer 2 kann intern DMA-Anforderungen erzeugen (programmierbar), außerdem sind über zwei DRQ-Leitungen externe Anforderungen möglich. DMA-Acknowledge Signale werden nicht erzeugt, jedoch können hierfür nötigenfalls entsprechende CS-Leitungen verwendet werden. Die Transfers können unsynchronisiert oder auch synchronisiert erfolgen.

4.5. Serielle Schnittstellen

Die miroCPU186 verfügt über zwei unabhängige serielle Schnittstellen mit dem 8274 MPSC für asynchrone, Bit- oder Byte-synchrone Obertragungsprotokolle. Die Baudraten sind programmierbar von DC bis 880K Baud, Receiver/Transmit Clock sind intern von Timer 0 bzw. Timer 1 verfügbar oder extern anschließbar. Ober den CPU internen Interrupt Controller sind Interrupt gesteuerte Obertragungen (non-vectored mode), über die DMA-Kanäle DMA gesteuerte Obertragungen möglich; unabhängig hiervon kann jederzeit in der Betriebsart "polled mode" gearbeitet werden. Zusätzlich zu den Signalen des 8274 MPSC werden die für den Betrieb mit Modems erforderlichen Hilfssignale TF/RS, DSR und RI zur Verfügung gestellt. Das Signal

'Ring Indicator' RI ist an den Interrupteingang INT2 bzw. INT3 geführt.

4.6. FIFO-Interface

Der 128-Byte FIFO Pufferspeicher ("first in - first out") ermöglicht eine bidirektionale, asynchrone Kopplung zwischen der miroCPU186 und einem Z80 Prozessor (ECB-Bus). Das Interface kann wahlweise in der Betriebsart "polled mode" und/oder Interrupt-gesteuert arbeiten, auch DMA Unterstützung durch die DMA Kanäle des iAPX 186 ist wählbar; mit äußerster Vorsicht ("CPU hangup" möglich!) kann auch die Betriebsart "wait" angewendet werden, in der der jeweilige Prozessor in den Wartezustand gesteuert wird, um eine Synchronisation zu erzielen.

4.7. Speicheradressierung

UCS
FC000h...FFFFFh: EPROM Speicherbereich, standard
C0000h...FFFFFh: max. Speicherbereich (256 kByte)

/LCS
00000h...1FFFFh: DRAM Speicherbereich, standard
00000h...3FFFFh: max. Speicherbereich (256 kByte)

/MCS0
80000h...8FFFFh: Refresh Adreßbereich, standard

/MCS1.../MCS3
nicht benutzte CS-Leitungen

Alle CS-Leitungen bis auf /UCS dürfen nicht mit Wartezyklen belegt werden!
Da der maximal adressierbare Speicherbereich für /LCS 256 kByte groß ist, müssen die mit 256 kBit DRAMs (512 kByte) bestückten miro186CPUs eine externe Dekodierung benutzen.

4.8. Peripherieadressierung

PBA
Basis-Adresse (programmierbar) für Peripheriebausteine,
Standard: 0000h, I/O-mapped (IN/OUT-Befehle)

/PCS0
PBA + 0...PBA + 127: Intel 8274 MPSC

/PCS1
PBA + 128...PBA + 255: TF/RSA (Schreiben)

/PCS2
PBA + 256...PBA + 383: TF/RSB (Schreiben)

/PCS3
PBA + 384...PBA + 511: nicht benutzt

/PCS4
PBA + 512...PBA + 639: Zilog Z8038-FIO

/PCS5
PBA + 640...PBA + 767: DSR (Lesen)

/PCS6
PBA + 768...PBA + 895: nicht benutzt

Die Leitungen /PCS0.../PCS3 sind standardmäßig mit 2 Wartezyklen , die Leitungen /PCS4.../PCS6 mit 3 Wartezyklen belegt.

4.9. Refresh für DRAMs

Die dynamischen RAM's erfordern alle 2 ms (4ms) 128 (256) Refresh-Impulse, die von Timer 2 gesteuert werden. Angewendet wird der Modus RAS-only Refresh. DRAMs mit Pin1-Refresh dürfen zwar eingesetzt werden, jedoch müssen die Refresh-adressen trotzdem extern durch die CPU erzeugt werden, da diese Betriebsart nicht vorgesehen ist.

Erzeugung der Refresh-Adressen:
Standardmäßig werden die Refresh-Adressen durch eine Interrupt-Service-Routine erzeugt, die alle 2ms (4ms) durch Timer 2 angefordert wird. Ist versehentlich der Interrupt maskiert worden, löst nach etwa 6,5 ms ein Watchdog-Timer einen NMI aus, um die Erzeugung der Refreshadressen sicherzustellen.
Sollen zeitkritische Programme nicht durch die Timer-Interrupts für den Refresh unterbrochen werden, können die Refresh-Adressen auch durch DMA-Übertragungen erzeugt werden. Angefordert wird die Übertragung der Refresh-Adressen alle 15 us ebenfalls durch Timer 2, so daß in 2 ms (4ms) alle 128 (256) Refresh-Adressen übertragen sind. Die internen Zähler des DMA-Kanals erzeugen nach jedem Transfer automatisch eine neue Adresse. Programme, die nicht auf die Ausführung der Timer-Interrupts für den Refresh warten können, werden nun nur noch durch eine einzelne DMA-Übertragung alle 15 us unterbrochen.

5. Die iAPX 186 CPU

5.1. Obersicht

Der in der neuen HMOS-III-Technologie gefertigte Baustein iAPX 186 kombiniert auf einem einzigen Chip integriert die Funktions-Einheiten eines fast vollständigen CPU-Boards und enthält:
- eine erweiterte und verbesserte 8086-2 CPU
- einen Taktgenerator für eine Standardtaktfrequenz von 8 MHz
- einen Interruptcontroller für je 5 int./ext. Anforderungen
- drei 16-Bit-Timer/Zähler
- zwei DMA-Kanäle mit einer Obertragungsrate bis zu 2 MByte/s
- die Bussteuereinheit
- Logik zur Chip-Select- und Wartezyklen-Generierung

Die Bussteuereinheit des iAPX 186 führt wie auch bereits beim 8086 sämtliche Speicherzugriffe parallel und unabhängig von der eigentlichen Ausführungseinheit durch. Während die CPU noch mit der Ausführung des aktuellen Befehls beschäftigt ist, holt die Buseinheit ("prefetching") bereits den Code für den/die nächsten Befehl(e) aus dem Speicher in eine 6 Byte lange Warteschlange ("instruction queue"). Die Berechnung der effektiven Adresse eines Operanden dauert nur noch ein bis zwei Takte im Gegensatz zu fünf bis zwölf Takten beim 8086, da die Buseinheit des iAPX 186 ein eigenes Adreßrechenwerk erhalten hat. Die Adreßarithmetik muß nicht mehr in Mikrocode vorgenommen werden, um die vier Komponenten zu verknüpfen, aus denen die Adresse eines Operanden gebildet wird:
- Segmentadresse
- Index
- Basis
- Displacement

Die Buseinheit führt auch alle Buszugriffe für die auf dem Chip befindlichen DMA-Kanäle aus; bei gleichzeitigen Zugriffswünschen von CPU und einem DMA-Kanal wird dem DMA-Kanal immer der Vorrang erteilt.

Der Baustein iAPX 186 zeigt im wesentlichen dasselbe Verhalten auf dem Bus wie der 8086, auch hier sind Daten- und Adreßbus gemultiplext; außerdem liefert der iAPX 186 auch dieselben Steuersignale. Aus der Sicht des Programmierers ist der CPU-Teil des iAPX 186 identisch mit dem 8086; alle Befehle, Register und Adreßmodi sind in gleicher Weise vorhanden. Damit kann bestehende Software ohne jede Änderung im Objektcode übernommen werden. Zusätzlich enthält der iAPX 186 jedoch einige neue Befehle, die Ein-/Ausgabevorgänge, Interruptroutinen und höhere Programmiersprachen noch besser unterstützen; diese Befehle sind auch beim iAPX 286, der anderen Weiterentwicklung der 8086-Architektur, implementiert. Damit hat der 16-Bit-Anwender ein durchgängiges Wachstumskonzept.

Zum Ansprechen der auf dem Chip befindlichen Peripherie- und Controllerfunktionen sind keine neuen Befehle notwendig. Diese Funktionseinheiten werden über auf dem Baustein befindliche Steuerregister angesprochen, die in einem 256 Byte langen Block zusammengefaßt sind. Ober einen "Control Block

Pointer" kann die Lage dieses Blocks beliebig irgendwo im Speicheradreßraum von einem Megabyte oder im E/A-Adreßraum von 64kByte festgelegt, und damit über normale MOVE- oder IN/OUT-Befehle programmiert werden. Nach dem Rücksetzen des Bausteins durch RESET liegt der Kontrollblock immer bei 0FF00h im E/A-Adreßraum. Da der "Control Block Pointer" selbst Teil des Kontrollblocks ist, kann der Kontrollblock danach durch Neubesetzen des Pointers beliebig verschoben werden.

Im Vergleich zum 8086 wurden beim iAPX 186 einige Verbesserungen vorgenommen, die die Leistung des Prozessors erhöhen. Neben einem eigenen Adreßrechenwerk für die Adreßarithmetik, wodurch die Berechnung von effektiven Adressen drei- bis sechsmal schneller abläuft, steht für Multiplikation und Division ebenfalls ein eigenes Rechenwerk zur Verfügung; diese Operationen können drei- bis viermal schneller als beim 8086 ausgeführt werden, während Shiftoperationen jetzt mit einer Geschwindigkeit von einem Takt pro Position durchgeführt werden.

Alle Stringtransfers laufen mit der maximalen Busgeschwindigkeit von 2 MByte/s ab und sind damit etwa doppelt so schnell als beim 8086; der Transfer von Datenstrings ist beim iAPX 186 jetzt auch im E/A-Adreßbereich möglich.

Neben einigen anderen Verbesserungen sorgen auch die neuen Instruktionen für eine Erhöhung der Verarbeitungsleistung. Eine Leistungserhöhung von 30% beim iAPX 186 im Vergleich zum 8 MHz 8086 darf für alle Programme erwartet werden - für Standardbenchmarks, die den 8086 bei unterschiedlichen Aufgaben untersuchen, ergibt sich für den iAPX 186 sogar eine noch günstigere Leistungszunahme von 65%.

5.2. Taktgenerator

Der Taktgenerator ist beim iAPX 186 mit auf dem Chip integriert und kann von einem direkt an den Baustein angeschlossenen 16 MHz-Quarz oder auch von einer externen Taktquelle gespeist werden. Der interne Taktgenerator erzeugt einen symmetrischen 8 MHz-Takt, der die interne Steuerung übernimmt und über den Anschluß CLKOUT auch als Systemtakt nach außen zur Verfügung steht.

Die Signale ARDY und SRDY erlauben den Anschluß von langsameren Speicher- oder E/A-Bausteinen an den iAPX 186. Das asynchrone Ready-Signal ARDY kann zu beliebigen Zeitpunkten angelegt werden, während SRDY synchron zum Prozessortakt erwartet wird. Zusätzlich zu den externen Ready-Signalen steht außerdem noch die programmierbare automatische Wartezyklengenerierung zur Verfügung, die mit der Chip-Select-Logik zusammenarbeitet. Das RESET-Signal wird vom Taktgenerator synchronisiert ausgegeben und kann als System-Reset-Signal verwendet werden.

5.3. Timer

Der iAPX 186 enthält drei voneinander unabhängige programmierbare 16-Bit-Timer/Counter; dadurch ergibt sich ein max. Zählwert von 65535, die max. Zählfrequenz beträgt 2MHz. Die einzelnen Timer zählen aufwärts, was ihre Verwendung als Softwarezähler einfacher macht. Bei Erreichen des maximalen Zählwertes können die Timer über den internen Interrupt-Controller eine Unterbrechung erzeugen. Die Timer können aber auch in einem Abfragemodus ("Polling") betrieben werden, da allle Timerregister zu jedem Zeitpunkt gelesen oder geschrieben werden können.

Timer 2 wird immer mit der internen Taktfrequenz von 2 MHz weitergeschaltet und kann damit Zeiten bis etwa 32 ms messen. Bei Erreichen des Maximalwertes kann er unterschiedliche Aktionen auslösen:

- er kann einen internen Interrupt generieren;
- er kann eine Anforderung an den DMA-Controller stellen, ein Byte/Wort zu übertragen;
- er kann Timer Ø oder 1 weiterzählen und dient so als Vorskalierer für einen der beiden anderen Timer.

Unabhängig von den generierten Signalen kann Timer 2 auch noch so programmiert werden, daß er beim Erreichen des Maximalwertes stehenbleibt oder sofort wieder bei Ø zu zählen anfängt.

Timer Ø und 1 haben dagegen zwei Maximalwertregister und stehen zudem über ein IN- und ein OUT-Signal mit der Außenwelt in Verbindung. Als mögliche Betriebsmodi für die Timer Ø und 1 seien hier nur folgende beschrieben:

- Die Zählerfrequenz kann intern (2 MHz oder Oberlaufsignal von Timer 2) und extern über das IN -Signal bestimmt werden. Im Externmodus kann der entsprechende Timer damit als Zähler für äußere Ereignisse verwendet werden;

- der Timer kann zwischen Maximalwert A und B alternieren. Bei Erreichen von Maximalwert A wird das OUT-Signal auf LOW gesetzt, beim Erreichen des Maximalwerts B auf HIGH, anschließend wird wieder bis zum Maximalwert A gezählt, usw.. Dadurch können beliebige symmetrische (z.B. Baudraten), aber auch sich nicht wiederholende Signalfolgen generiert werden, da die Software ja den Inhalt eines Maximalwertregisters ändern kann, während das andere gerade verwendet wird;

- das IN-Signal kann dazu verwendet werden, das Weiterschalten des Zählers durch den internen Takt zu verhindern (LOW blockiert den Zähler, HIGH läßt ihn weiterlaufen). Dadurch kann der Timer die Zeitdauer externer Zustände messen;

- ein Puls auf dem IN-Signal kann außerdem den Zähler auf Null setzen. Dieser Modus dient zum Synchronisieren der internen Zeit mit externen Ereignissen.

5.4. DMA-Einheit

Der iAPX 186 enthält zwei voneinander unabhängige programmierbare DMA-Kanäle. Quelle und Ziel jedes DMA-Transfers können beliebig innerhalb des 1 MByte großen Speicheradreßraumes oder des 64 kByte großen E/A-Adreßraumes liegen. Ein einzelner Transfer besteht immer aus einem Lesezyklus auf die aktuelle Quelladresse und einem Schreibzyklus auf die Zieladresse. Es lassen sich damit folgende Übertragungsrichtungen programmieren:
- Speicher nach Speicher
- Speicher nach E/A
- E/A nach Speicher
- E/A nach E/A.

Quell- und Zieladreßzeiger können unabhängig voneinander inkrementiert, dekrementiert oder konstant belassen werden. Es kann zu einem Zeitpunkt zwar nur zu/von zwei E/A-Geräten übertragen werden, aber nach jedem Block kann die E/A-Adresse verändert und damit ein anderes Gerät selektiert werden. Deswegen erreicht man mit diesen zwei Kanälen sogar mehr Flexibilität als mit einem separaten DMA-Controller mit vier Kanälen, da diese meist fest mit je einem E/A-Gerät verbunden sind. Soll jedoch wirklich gleichzeitig auf mehr als zwei Kanälen übertragen werden - dabei ist aber immer der max. mögliche Busdurchsatz im Auge zu behalten -, sind externe DMA-Controller über die HOLD- und HLDA-Signale an den iAPX 186 anschließbar.

DMA-Transfers können in Bytes oder Worten erfolgen; Worte können beliebig auf geraden oder ungeraden Adressen stehen. Beim Worttransfer ergibt sich eine max. Übertragungsrate von 2 MByte/s (ein Zyklus mit 4 Takten lesen, ein Zyklus schreiben). Das Übertragen eines Blockes wird automatisch beendet, wenn der 16 bit breite Übertragungszähler auf Null gelaufen ist; zu diesem Zeitpunkt ist auch Interrupt generierbar. Die einzelnen DMA-Anforderungen können über die Anschlüsse DRQ 0 und DRQ 1 von außen kommen oder intern von Timer 2 generiert werden. Dies erlaubt den einfachen Anschluß von Geräten mit konstanter Übertragungsrate oder das Erfassen von Meßwerten in regelmäßigen Zeitabständen.

Alle Buszugriffe der beiden DMA-Kanäle werden von der Buseinheit ausgeführt. Um Konflikte bei gleichzeitigen Anforderungen auflösen zu können, ist die Priorität der Kanäle von der Software definierbar. Falls einem der Kanäle die höhere Priorität zugewiesen ist, hat er im Konfliktfall immer Vorrang vor dem anderen Kanal; bei gleicher Priorität werden Konflikte alternierend entschieden. DMA-Zugriffe haben jedoch generell Vorrang vor Buszugriffen durch die CPU, und dies hat wiederum Vorrang vor dem Prefetching von Codes durch die Buseinheit.

5.5. Interrupt-Controller

Der auf dem iAPX 186 integrierte Interrupt-Controller verarbeitet die internen Anforderungen von den drei Timern und den zwei DMA-Kanälen, sowie die Anforderungen von fünf externen Anschlüssen. Einer der fünf Eingänge entspricht dem NMI (Non-maskable Interrupt) des 8086, die Bedeutung der anderen vier ist per Software programmierbar:

- Modus 1:
  Jede der vier Leitungen dient als eigenes Interrupt-Signal; das Generieren des Interruptvektors geschieht dann auf dem Chip. Jede der vier Leitungen kann dabei unabhängig auf Flanken- oder Pegeltriggerung programmiert werden.

- Modus 2:
  Je zwei Leitungen werden zum Anschluß eines externen Interruptcontrollers 8259A zusammengefaßt. Ein Anschluß dient dabei als Interrupt-Request, der andere als Interrupt-Acknowledge-Signal. Die verbleibenden zwei Leitungen sind entweder im Modus 1 verwendbar, oder man kann auch an sie einen weiteren Interruptcontroller anschließen. Da der externe 8259A selbst wieder bis zu acht 8259A's im Slave-Modus steuern kann, sind in diesem Modus bis zu 128 verschiedene externe Interrupts vom iAPX 186 erkennbar.

Für beide Modi gilt, daß jede Interruptquelle einzeln maskierbar ist und mit einer von acht Prioritätsstufen belegt werden kann. Die internen und externen Interrupts können, wie bereits vom 8259A bekannt, ineinander verschachtelt abgearbeitet werden, was für die Software ein elegantes Abarbeiten von in Folge auftretenden Interrupts erlaubt.

- Modus 3:
  In diesem Modus kann der interne Interruptcontroller als "Slave" an einen externen 8259A angeschlossen werden, wozu alle vier Anschlüsse nötig sind. Bei dieser Betriebsart kann das Betriebssystem RMX/86 ohne Änderung die internen Interruptquellen ansprechen.

Im Vergleich zum 8086 ist die Interrupt-Reaktionszeit wesentlich kürzer geworden, da
- die längsten Befehle, nämlich Multiplikation und Division, jetzt drei- bis viermal so schnell sind;
- die Bearbeitung der Interruptsequenz von 61 Takten beim 8086 auf 42 Takte für interne und und 55 Takte für externe Interrupts beim iAPX 186 verkürzt worden ist,
- sich mit dem PUSHALL-Befehl alle Register in 4,5 us auf den Stack retten lassen.

Eine "Worst-Case"-Betrachtung kann also jetzt von etwa 15 us vom Anlegen des Interruptsignals bis zum Ausführen des ersten Befehls in der Interruptroutine ausgehen.

Die Interrupt-Vektortabelle hat beim iAPX 186 einige neue Einträge bekommen; so führen jetzt z.B. unbenutzte Befehlscodes auf eine Trap. Gleiches gilt für ESCAPE-Befehle, wenn kein 8087-Numerik-Coprozessor angeschlossen ist.

5.6. Chip-Select Logik

Die auf den Chip des 80186 integrierte Chip-Select-Logik ersetzt 10 bis 15 externe TTL-Bausteine und schnelle Dekodier-PROMs sowie ein Verdrahtungsfeld. Der iAPX 186 hat sechs Chip-Select-Leitungen für Speicherbereiche und sieben für Peripheriebausteine. Die auf jede Leitung entfallenden Adreßbereiche sowie die Anzahl der intern zu generierenden Wartezyklen sind durch Software definierbar.

Da heute Speicherbausteine sehr hoher Dichte zur Verfügung stehen, ist mit den sechs Leitungen für den Speicheradreßbereich der Arbeitsspeicherbereich der meisten Systeme ansteuerbar.

Für jeden der drei Adreßbereiche lassen sich bis zu drei Wartezyklen festlegen; außerdem kann noch angegeben werden, ob die interne Wartezeit noch durch ein externes Ready-Signal verlängert werden kann.

Jedes der sieben /PCS-Signale zum Ansteuern von Peripherie-Bausteinen deckt einen Bereich von je 128 Byte ab, was auch für sehr komplexe Controller ausreichend ist; die sieben Bereiche sind direkt aufeinanderfolgend. Die Anfangsadresse des ersten Bereichs kann auf jede beliebige 1-K-Grenze gelegt werden, und zwar nicht nur im E/A-Adreßraum, sondern auch im Speicheradreßraum. Damit kann die Ein-/Ausgabe sowohl mit IN/OUT-Befehlen im "I/O-mapped"-Modus als auch mit MOVE-Befehlen im "Memory-mapped"-Modus erfolgen. Auch hier können wieder - getrennt für /PCS0.../PCS3 und /PCS4.../PCS6 - bis zu drei automatisch einzufügende Wartezyklen definiert werden.

Da Timer, DMA- und Interrupt-Controller bereits auf dem Chip integriert sind, sind für viele Anwendungen sieben externe Chip-Select-Leitungen zur Auswahl von Peripheriebausteinen mehr als ausreichend. Deswegen kann man /PCS5 und /PCS6 auch so programmieren, daß sie den Wert der Adreßleitungen A0 und A1 über den ganzen Buszyklus herauslegen. Dadurch werden 8-Bit-Peripherie-Controller, die meist vier interne Register enthalten, direkt ansteuerbar gemacht.

6. Schreib-/Lesespeicher

Die miroCPU186 ist standardmäßig mit 16 dynamischen Speicher-Bausteinen 64kx1 mit einer maximalen Zugriffszeit von 200 ns bestückt. Ein speziell für die miroCPU186 entwickelter RAM-Controller ist durch ein schnelles Schottky PAL realisiert, das die optimale Ausnutzung der CPU-Geschwindigkeit - ohne Einfügung von Wartezyklen! - bei einer Taktfrequenz von 8 MHz ermöglicht.

6.1. 64 kBit DRAMs

Der Einsatz von Speichertypen 64kx1 anderer Hersteller ist ebenfalls möglich, wenn diese die Betriebsart RAS-only Refresh aufweisen; die max. Zugriffszeit für TRAC darf 200 ns, für TCAC darf 135 ns nicht überschreiten. Langsamere Speichertypen dürfen nicht benutzt werden, auch nicht bei Programmierung von Wartezyklen! Nur genügend schnelle Speichertypen lassen sich direkt von der CPU wie auf der miroCPU186-Baugruppe steuern, und erzielen durch Vermeidung der zusätzlichen Laufzeit von Bustreiber-Bausteinen die beste Ausnutzung der CPU-Geschwindigkeit. Da die Speicher NMOS-Bausteine sind, muß die CPU nur die Kapazität der Speicher genügend schnell umladen können, wie es für die iAPX 186 CPU gegeben ist, die statisch zu treibenden Ströme sind gering.

Das Programmieren von Wartezyklen durch den CPU-internen Wartezykle-Generator ist ohnehin für den DRAM-Speicherbereich auf Leitung /LCS generell nicht zulässig, da die Programmable Array Logic (PAL) des RAM-Controllers ebenfalls auf Wartezyklen umprogrammiert werden muß, d.h. der vorhandene PAL ist durch einen anderen PAL mit einer auf 1-3 Wartezyklen geänderten Logik zu ersetzen. Dies ist natürlich nicht sinnvoll, da ja nun die CPU-Geschwindigkeit gerade für den wichtigen Arbeitsspeicher herabgesetzt wird und wird deshalb nicht angeboten.

Die DRAM's benötigen alle 2 ms (4ms) 128 (256) Refresh-Zyklen; angewendet wird der RAS-only Refreshmodus. Die Erzeugung der 128 (256) Refreshadressen erfolgt in einer alle 2 ms (4ms) von Timer 2 angeforderten Interruptservice-Routine. Der PAL-RAM-Controlller erzeugt bei jedem Lesen oder Schreiben einer Refreshadresse einen Ausgangsimpuls, der auf die Leitung /RAS für den RAS-only Refresh geschaltet wird, während die CAS-Impulse beim Refreshen unterdrückt werden.

Die während des Refreshvorganges von der CPU erzeugten Adressen müssen innerhalb des durch /MCS0 programmierten Refresh-Adreßbereichs liegen, da diese Leitung als Freigabe- bzw. Acknowledge-Signal für den Refresh verwendet wird. Die Freigabe erfolgt sowohl beim Lesen als auch beim Schreiben in dem durch /MCS0 definierten Refresh-Adreßbereich.

Während des Refreshvorgangs werden die /CAS-Impulse durch den PAL-RAM-Controller unterdrückt, so daß auch beim Schreiben keine Änderung des Speicherinhalts erfolgen kann; die Lese- oder Schreibbefehle in der Refreshroutine dienen lediglich zum Aktivieren des Freigabesignals /MCS0.

Um kompatibel mit einer evt. späteren Speichererweiterung durch ein Memory Piggy-Pack zu sein, sollten die Refreshadressen durch die Interrupt-Serviceroutine dennoch nur gelesen werden, da sich bei voller Ausnutzung des 1 MByte Speicherbereichs die Refreshadressen innerhalb des benutzten DRAM Speicherbereichs befinden und beim Schreiben in diesem Bereich der Speicherinhalt zerstört würde.

6.2. DMA-Refresh

Sollen zeitkritische Programme nicht durch die Interrupt-Service-Routine für den Refresh unterbrochen werden, kann die Erzeugung der Refreshadressen auch durch den Kanal Ø des CPU-internen DMA-Controllers erfolgen. Die internen Zähler des DMA-Kanals erzeugen nach jedem Transfer automatisch neue Adressen, die sich wiederum alle in dem durch /MCSØ definierten Refresh-Adreßbereich befinden müssen.

Angefordert wird die DMA-Obertragung der Refreshadressen alle 15 us ebenfalls durch Timer 2. Nach 128 (256) Transfers innerhalb des Refresh-Adreßbereichs ist der Refresh aller 128 (256) ROW-Adressen erfolgt, und zwar innerhalb der geforderten 2 (4) ms. Bei jedem Transfer erfolgt der Refresh nur einer ROW-Adresse, da dieselbe Ziel- wie Quelladresse benutzt werden sollte wegen der Kompatibilität mit einer Speichererweiterung auf 1 MByte.

Unerheblich ist die Obertragungsrichtung, es muß lediglich sichergestellt sein, daß alle Refreshadressen während eines Refreshzyklus einmal angesprochen werden, ob mit einem Lese- oder mit einem Schreibbefehl, ist ohne Belang; auch in der Betriebsart RAS-only-Refresh wird der Speicherinhalt nicht angetastet, da der PAL-RAM-Controller während der Refreshzyklen das /CAS-Signal unterdrückt.

Wird dieselbe Ziel- wie Quelladresse benutzt und wird der Adreßbereich für /MCSØ wie standardmäßig vorgesehen auf 64 KByte programmiert, ist dieser nach 65536 Transfers gerade einmal durchlaufen. Die Interruptservice-Routine kann nun gleichzeitig die Pointer auf die Anfangsadressen zurücksetzen und die DMA-Obertragung neu starten. Der Transferzähler braucht nicht zurückgesetzt werden, da er bereits auf Null steht, wenn die vorherige Obertragung beendet und dadurch ein Interrupt ausgelöst wurde.

Sollte es dennoch einmal passieren, daß die Anforderung von DMA-Transfers für den Refresh unterbrochen wurde, weil der Obertragungszähler nach 65536 Zyklen stehengeblieben ist, und zu diesem Zeitpunkt die Interruptservice-Routine zum Starten einer neuen DMA-Obertragung nicht oder nicht rechtzeitig bedient werden kann, wird durch einen sog. "Watchdog-Timer" etwa 65 us nach dem letzten Refresh-Impuls ein NMI angefordert. Die NMI-Service-Routine kann nun anstelle des zu diesem Zeitpunkt gerade maskierten DMA-Interrupts die neue DMA-Obertragung starten und so den Refresh der DRAM's sicherstellen.

Nach 65536 DMA-Zyklen wird die DMA-Obertragung automatisch beendet und von der DMA-Einheit ein Interrupt ausgelöst, weil der 16 Bit breite Obertragungszähler auf Null gelaufen ist. Die DMA-Obertragung muß nun durch eine Interrupt-Routine neu gestartet werden. Der Anwender sollte sicherstellen, daß die Interruptservice-Routine zum Starten der DMA-Obertragung nicht für längere Zeit gesperrt wird, daß also der Befehl CLI zur Maskierung von Interrupts möglichst effektiv nur in wirklich zu schützenden Programmteilen verwendet wird.

Der DMA-Kanal 0 für den Refresh sollte die höchste oder zumindest die gleiche Priorität wie Kanal 1 erhalten, damit die Anforderung der Refresh-Transfers durch Timer 2 auf jeden Fall angenommen wird. Der DMA-Transfer für die Refresh-Zyklen durch Kanal 0 erfolgt gegebenenfalls alternierend mit dem Transfer auf Kanal 1.

6.3. 256 kBit DRAMs

Der Einsatz von 256 kBit DRAM's - max. 200 ns Zugriffszeit - ist ebenfalls vorgesehen. Da der Pin 1 der 256kx1 Chip's für eine zusätzliche gemultiplexte Adresse benötigt wird, ist für diese Bausteine kein Pin 1-Refresh möglich, die Refresh-Adressen müssen extern erzeugt werden. Da die Betriebsart RAS-only Refresh für Speicher und PAL-RAM-Controller bereits fest eingestellt ist, braucht mit J1 nur noch die gemultiplexte zusätzliche Adresse A9/A18 an Pin 1 gelegt werden. Da die Speicherauswahl-Leitung LCS nur auf eine maximale Speichergrösse von 256 KByte programmiert werden kann, muß für den 512 KByte-Block der 16 x 256 kBit DRAMs eine externe Dekodierung benutzt werden; diese wird mit J3 eingestellt. Außerdem muß die Adresse AD9 am Multiplexer für die RAS/CAS-Adressen auf A17 umgejumpert werden, J2 ist entsprechend einzustellen.

6.4. DRAMs mit 4 ms Refreshzyklus

Die Verwendung von dynamischen Speichern mit 256 Refresh-zyklen während 4 ms ist ohne Änderungen der Schaltung und ohne Umprogrammieren der zyklischen Interrupt-Anforderung oder auch DMA-Anforderung durch Timer 2 möglich.

Voraussetzungen sind wieder DRAM's mit 1 Bit-Organisation, 64 kBitx1 oder 256 kBitx1, einer max. Zugriffszeit von 200 ns und den Betriebsmodi RAS-only-Refresh oder auch Pin 1-Refresh.

6.5. Watchdog-Timer

Zur Oberprüfung der Funktion der gesamten miroCPU186-Baugruppe wird ein sog. Watchdog-Timer verwendet, der die Existenz der Refreshimpulse alle 15 us bzw. 2 ms (4ms) überprüft. Sollte die Interruptservice-Routine zum Starten der DMA-Transfers für den Refresh der DRAM's nicht bedient worden sein, weil z.B. ausgerechnet in diesem Moment der Befehl CLI verwendet wurde, oder weil ein fehlerhaftes Programm vorliegt, kann der Watchdog-Timer durch die Auslösung eines nicht maskierbaren Interrupts NMI eine neue DMA-Obertragung anfordern und somit die weitere Ausführung der Refreshzyklen erzwingen.

Der Watchdog-Timer ermöglicht also die Benutzung auch von CLI ohne Einschränkung, da durch NMI auf jeden Fall sichergestellt ist, daß die Service-Routine für den DMA-Refresh rechtzeitig bedient wird. Diese Service-Routine muß lediglich alle 2,048 s durchlaufen werden, um eine neue DMA-Obertragung zu starten; erst nach dieser Zeit ist im Normalfall das Auslösen eines NMI möglich, wenn zu diesem Zeitpunkt gerade der maskierbare Interrupt des DMA-Kanals gesperrt sein sollte. Im übrigen sollte diesem Interrupt möglichst die höchste Priorität zugewiesen werden.

Standardmäßig wird allerdings nicht der DMA-Refresh, sondern eine Timerinterrupt-Routine für den Refresh verwendet, die Zeitkonstante des Watchdog-Timers muß durch Entfernen von Jumper J23 auf 6,5 ms vergrößert werden, da die Refreshimpulse jetzt nicht mehr alle 15 us, sondern nur noch alle 2 ms (4 ms) erzeugt werden. Bleiben die Refreshimpulse aus, wird nach 6,5 ms ein NMI ausgelöst, der die Ausführung der Refreshzyklen erzwingt.

6.6. Typenliste für DRAM-Bestückung

64 kBit x 1 DRAMs mit Pin-1 Refresh:

Mitsubishi	M5K4164P-15
Mitsubishi	M5K4164P-20
Fujitsu	MB8265-15
Fujitsu	MB8265-20
Motorola	MCM6664
Motorola	MCM6664A
Mostek	MK4164

64 kBit x 1 DRAMs mit RAS-only Refresh:

Mitsubishi	M5K4164NS-15
Motorola	MCM6665A
Mostek	MK4564
Intermetall	ITT4164-15
Intermetall	ITT4164-20
Toshiba	TMM4164C-3
Toshiba	TMM4164C-4
NEC	uPD4164-2
NEC	uPD4164-3

256 kBit x 1 DRAMs mit RAS-only Refresh:
NEC uPD41256-15
NEC uPD41256-20

7. Buskopplung mit Z8038 FIO

Der 128-Byte FIFO Pufferspeicher des Z8038 FIO ("first in - first out") ermöglicht eine bidirektionale, asynchrone Kopplung zwischen der miroCPU186 und einem Z80 Prozessor (ECB-Bus). Als zweiseitiges ("dual ported") I/O-Port liegt der FIFO im I/O-Adreßbereich der "Master-CPU" Z80 und der "Slave-CPU" miroCPU186 und belegt somit keine Speicheradressen. Der FIO organisiert den Datentransfer zwischen den CPU's in der Betriebsart "non-Z-Bus microprocessor" für beide Ports; die CPU's können unabhängig voneinander mit unterschiedlichen Geschwindigkeiten, Datenbusbreiten und Befehlssätzen arbeiten.

Die beiden Ports des Z8038 FIO werden durch je 15 bzw. 16 programmierbare, direkt adressierbare Kontrollregister gesteuert. Diese Register spezifizieren die Betriebsmodi des FIO und enthalten "message"-Register für die Kommunikation von CPU zu CPU ohne Benutzung des FIFO Pufferspeichers.

Die mit der iAPX 186 verbundene Port 1-Seite des FIO kontrolliert die Richtung der Datenübertragung, das Löschen des FIFO-Pufferspeichers und den Reset des gesamten Bausteins; die Port 2-Seite ist mit der Z80 "Master-CPU" verbunden und muß nach dem Reset auf die Freigabe durch den iAPX 186 "Slave"-Prozessor über Port 1 warten.

7.1. Software-Reset

Um den Aufwand zu reduzieren, erfolgt der Reset des Z8038 FIO nicht durch Hardware, sondern allein durch Software-Befehle. Um einen definierten Anfangszustand zu erhalten, wird durch ein Reset des Port 1 auch die Port 2-Seite des FIO zurückgesetzt, indem alle Kontroll-Bits auf Null gesetzt werden. Der Reset des FIO wird durch das Schreiben einer 1 in das Kontrollregister 0 der Port 1-Seite erreicht; danach muß der Reset-Zustand durch Schreiben einer 0 wieder verlassen werden. Erst dann kann mit der Programmierung des Bausteins auf beiden Seiten begonnen werden. Die CPU auf Port 2-Seite des FIO kann durch Lesen des Kontrollregister 0 das Verlassen des Reset-Zustands und die Freigabe durch Port 1 erkennen.

7.2. DMA-Betrieb

Der Z8038 FIO unterstützt DMA -Transfers auf beiden Seiten auch gleichzeitig und asynchron. Die Port 1-Seite des Z8038 FIO ist an die untere Hälfte des Datenbus der miroCPU186 angeschlossen, sie kann deswegen nur mit Byte Transfers durch den Kanal 1 der DMA-Einheit auf dem 80186-Chip erreicht werden. Bei Worttransfers kann die obere Hälfte des 16 Bit-Wortes nicht übertragen werden, da der Z8038 FIO diese ignoriert.

Die Anforderung einer DMA-Übertragung durch den FIO erfolgt über die /REQ-Leitung an DMA-Kanal 1 des iAPX 186, wenn der FIFO Pufferspeicher voll oder leer ist - je nach Richtung der Datenübertragung. Ein einzelner Transfer besteht sequentiell immer aus einem Lesezyklus auf die aktuelle Quelladresse und einem Schreibzyklus auf die Zieladresse, ein DMA-Acknowledge-signal (/DACK) für den FIO wird nicht benötigt.

Die DMA-Übertragung zwischen dem Arbeitsspeicher der miroCPU186 und dem FIO erfolgt immer unter Zwischenschaltung des DMA-Kanals, die Betriebsart "flyby" ist mit der internen DMA-Einheit des 80186 nicht möglich.

Die Port 2-Seite des Z8038 FIO ist mit den Datenleitungen des Systembus verbunden und kann durch Bausteine wie den Z80 DMA Controller bedient werden, wobei auch die Betriebsart "flyby" möglich ist.

In dieser Betriebsart des DMA-Controllers passiert das zu transferierende Byte den DMA-Kanal nicht, sondern wird direkt zwischen FIO und Speicher oder anderen Peripheriebausteinen übertragen. Zur Synchronisation ist jetzt nicht nur das Signal /REQ, das eine DMA-Übertragung für den FIO vom DMA-Controller anfordert, sondern auch das DMA-Acknowledge-Signal /DACK vom DMA-Controller notwendig, das die Anforderung bestätigt. Eine sequentielle Übertragung wie auf der Port 1-Seite ist aber ebenso möglich, in diesem Fall ist wiederum kein /DACK-Signal erforderlich.

Die maximale Geschwindigkeit der Datenübertragung über den FIFO Pufferspeicher beträgt 800 kByte/s. Wird die Übertragung auf beiden Seiten und gleichzeitig durch DMA unterstützt, lassen sich auch längere Blöcke als 128 Byte ohne Wartezeiten übertragen, da die von einer Seite eingeschriebenen Bytes sofort wieder von der anderen Seite ausgelesen werden.

Bei wenig unterschiedlichen Übertragungs-Geschwindigkeiten wird der Pufferspeicher zu einem "scratchpad memory" für kontinuierliche DMA-Übertragungen, die Länge des übertragenen DMA-Blocks kann also wesentlich größer werden als die Länge des FIFO Pufferspeichers.

7.3. Interrupt-Betrieb

Kommunikationswünsche zwischen den "Slave-CPU's" und der "Master-CPU" können per Interrupt übermittelt werden. Ausgelöst werden kann ein Interrrupt z.B. durch das Schreiben in ein "message"-Register, durch die Umschaltung der Datenübertragungsrichtung sowie durch weitere fünf Bedingungen, die im Interrupt-Vektor-Register des Z8038 gelesen werden können.

Die Bearbeitung von Interruptsequenzen durch den internen Interrupt-Controller der iAPX 186 CPU erfolgt für interne Interrupts 30% schneller als für externe Interrupts, weil in dieser Betriebsart keine Taktzyklen für INTA benötigt werden, sondern das Generieren des Interruptvektors auf dem Chip erfolgt.

Die Port 1-Seite des Z8038 FIO darf also selbst keinen Vektor erzeugen, das NV-Bit ("no vektor on interrupt") muß 1 gesetzt werden. Zu Beginn der Interrupt-Serviceroutine kann nun durch Lesen einzig des Interruptvektor-Registers des FIO festgestellt werden, welche Bedingung den Interrupt ausgelöst hat. Dies bedeutet, daß gegenüber einer direkten Obertragung des Interruptvektors auf dem Datenbus vom FIO zur CPU die Interrupt-Reaktionszeit nicht verschlechtert wird, da ja die Bearbeitung für interne Interrupts bis zu diesem Zeitpunkt wie gesagt 30% schneller erfolgt und lediglich ein zusätzlicher IN-Befehl für das Lesen des Interruptvektor-Registers benötigt wird.

Die ebenfalls voll interruptfähige Port 2-Seite des FIO ist in die "daisy-chain"-Logik der Z80 Master-CPU, Interrupt mode 2, eingebunden und kann daher nur im Modus NV = 0 ("vektor on interrupt") betrieben werden. Außerdem sollte der Modus VIS=1 ("vektor includes status") gewählt werden, der das zusätzliche Lesen des Interruptvektor-Registers erspart, weil der Interruptvektor selbst die Ursache der Interruptauslösung beinhaltet.

Die während eines Interrupt-Zyklus von der Z80 CPU automatisch erzeugten Wartezyklen sind für das Lesen des vom FIO zu liefernden Vektors nicht ausreichend, so daß ein Interface-PAL weitere Wartezyklen einfügen muß. Das Interface-PAL schaltet außerdem während eines Interrupt-Acknowledge-Zyklus den Bustreiber des FIO-Systembus-Interface in Leserichtung, gibt diesen aber nur frei, wenn gleichzeitig feststeht, daß der Interrupt durch den FIO ausgelöst wurde und kein anderer Baustein den Datenbus benutzt: Bedingung IEI=1 und IEO=0.

Zur Minimierung der Laufzeit der "daisy-chain"-Priorisierungskette besitzt die miroCPU186-Baugruppe eine "Look-ahead"-Logik. Zusätzlich wird mit dieser Logik erreicht, daß in der IEI/IEO-Kette nachfolgende Bausteine rechtzeitig gesperrt werden, indem der Ausgang IEO der miroCPU186-Baugruppe bereits durch das /INT-Signal des FIO auf Low geschaltet wird; die Laufzeit für das IEO-Signal durch den FIO entfällt dadurch.

Im Gegensatz zu den Z80-I/O-Bausteinen kann der FIO nicht feststellen, wann der Prozessor die Instruktion RETI ausführt; die Rückkehr zum Hauptprogramm muß für den FIO durch Rücksetzen des "Interrupt under service"-Bit (IUS) im jeweiligen Interrupt Status Register des FIO eingeleitet werden.

Um minimale Verzögerungszeiten zu erzielen, sollte die Baugruppe am Anfang der "daisy-chain" liegen, da während eines Befehlshole-Zyklus keine Dekodierung von "ED-4D" durch den FIO und somit keine Änderung des IEO-Ausgangs erfolgt.

7.4. Wait-Betrieb

Für beide Ports des Z8038 FIO ist durch Jumper auch die Betriebsart "wait operation" wählbar, die eine Synchronisierung von Datenübertragungen mit dem FIO durch Einfügen von beliebig vielen Wartezyklen durch die jeweilige CPU erreicht. Diese Betriebsart ist aber nur mit äußerster Vorsicht zu verwenden, da sie zu leicht die CPU's für unzulässig lange Zeit oder auch vollständig blockieren kann ("CPU hangup").

A. Belegung der Steckbrücken

Durch Programmierung der auf der miroCPU186 befindlichen Jumperfelder läßt sich die Baugruppe auf spezielle Eigenschaften einstellen und so an verschiedene Anforderungen anpassen. Hierzu werden Steckbrücken entfernt und neu gesetzt oder Wire-Wrap Verbindungen gezogen.

Alle für die jeweilige Betriebsart nicht relevanten Verbindungen sind mit "d.c." ("don't care") gekennzeichnet, alle Steckbrücken mit "n.c." ("not connected") sind zu entfernen bzw. dürfen nicht bestückt werden; die Jumper sind durchnummeriert von J1 bis J25.

A.1. Basis I/O-Adresse auf Systembus

Die miroCPU186 Baugruppe belegt im I/O-Adreßbereich des Systemprozessors 2 Adressen; die Basisadresse ist durch ein Wrapfeld in Doppelschritten frei einstellbar. Mögliche Basisadressen sind somit 00h, 02h, 04h, 06h, ..., FEh. Die Standardeinstellung ist 50h, die Baugruppe belegt also die I/O-Adressen 50h und 51h. Die für das Wrapfeld W1 angegebenen Pins sind miteinander zu verbinden, um das jeweilige Adreßbit auf "0" zu legen. Offenliegende Verbindungen bedeuten "1" für den Adreßvergleicher:

Adreßbit	Wrapfeld W1
A7 = "0"	7-8
A6 = "0"	6-8
A5 = "0"	5-8
A4 = "0"	4-8
A3 = "0"	3-8
A2 = "0"	2-8
A1 = "0"	1-8

Standard = 1-8, 2-8, 3-8, 5-8, 7-8
(Basisadresse = 50h; A7, A5, A3, A2, A1 = "0"; A4, A6 = "1")

A.2. EPROM Speicherkapazität

Die beiden auf der miroCPU186 befindlichen Sockel für EPROM's können mit den EPROM-Typen 2732 bis 27512 bestückt werden, jedoch stets nur mit je zwei gleichen Typen; es müssen auch stets beide Sockel bestückt werden! EPROM's mit 24-poligen Gehäusen müssen wie üblich mit ihrem Pin 1 in die Buchse 3 der 28-poligen Fassung gesteckt werden, die oberen vier Anschlüsse des Sockels bleiben frei. Nur Bausteine, die nach JEDEC-JC42-Norm ausgelegt sind, können eingesetzt werden, d.h. Typen, die pinkompatibel zu den angegebenen sind.

Die Einstellung der EPROM-Kapazitäten erfolgt mit den Jumperfeldern J4, J5 und J25:

EPROMs	J4	J5	J25
2732A-2, 2732A-20	1-2	1-2	1-2
2764, 2764-2, 2764-25 = Standard	n.c	1-2	1-2
27128, 27128-2	2-3	1-2	1-2
27256, 27256-1, 27256-2, 27256-20, 27256-25	2-3	2-3	1-2
27512, 27512-1, 27512-2, 27512-20, 27512-25	2-3	2-3	1-2

Die EPROM's benutzen den gemultiplexten Adreß-/Datenbus der CPU, sind also direkt mit der CPU zusammengeschaltet. Es dürfen daher nur Typen verwendet werden, die ihre Datenausgänge schnell genug hochohmig schalten, weil sonst Buskonflikte entstehen können.

Die Auswahl der verwendbaren EPROM's muß also in erster Linie nach dem Wert für TDF = OE High to Output Float (Data Float, Output in High-Z) erfolgen, alle Typen mit TDF = 85 ns max. können eingesetzt werden. Eingehalten wird dieser Wert von fast allen EPROM's mit einer Zugriffszeit von max. 250 ns; diese können problemlos und ohne Programmierung von Wartezyklen eingesetzt werden.

Auch einige CMOS-EPROM's mit Zugriffszeiten größer als 250 ns halten den Wert für TDF = 85 ns max. ein; diese dürfen ebenfalls verwendet werden, wenn entsprechend Wartezyklen für UCS (Upper Chip Select, Auswahlsignal EPROM) programmiert werden.

Alle EPROM's, die den Wert TDF = 85 ns max. nicht einhalten, können auch bei Programmierung von Wartezyklen nicht eingesetzt werden! Die Wartezyklen haben keinen Einfluß auf die Geschwindigkeit der Ausgangstreiber der EPROM's, die die Daten eines vorhergehenden Lesezyklus noch auf den Bus treiben, wenn die CPU bereits die Adressen des nächsten Zyklus auf den Bus treibt, so daß Buskonflikte auftreten.

1.3. DRAM Speicherkapazität

Die Standardversion der miroCPU186 ist mit 16 Stück 64 kBit DRAMs bestückt, was einer Kapazität des Schreib-/Lesespeichers von 128 kByte entspricht. Die Speicherkapazität kann optional um den Faktor vier auf 512 kByte erhöht werden, wenn die Karte mit Spezifikations-kompatiblen 256 kBit DRAMs, z.B. NEC 41256D-20, bestückt wird. An den Pin 1 muß eine zusätzliche gemultiplexte Adresse gelegt werden, und es muß die externe Dekodierung der Speicherauswahl-Leitung eingestellt werden. Außerdem muß die Adresse AD9 am Multiplexer für die RAS/CAS-Adressen auf A17 umgejumpert werden. Dazu müssen die Pins 1-2 des Jumperfeldes J1 und die Pinreihen 1-2-3 der Jumperfelder J2 und J3 entsprechend belegt werden.

DRAMs	J1	J2	J3
64 kBitx1 = Standard	n.c.	1-2	1-2
256 kBitx1	1-2	2-3	2-3

Es dürfen nur 1-Bit organisierte DRAM's mit einer Zugriffszeit von TRAC = 200 ns max. und TCAC = 135 ns max. und der Betriebsart RAS-only Refresh oder Pin1-Refresh eingesetzt

werden.

A.4. DMA Selektion

DMA Kanäle	J10	J11
Kanäle 0 und 1: Seriell I/O Kanal A	1-2	2-3
Kanal 1: FIO Port 1 Side DMA	d.c.	3-4
Kanal 0: Timer 2 Refresh DMA	n.c.	d.c.

(Standard: J10 = n.c., J11 = n.c.)

Standardmäßig werden die Kanäle der DMA-Einheit nicht benutzt.

A.5. Seriell I/O

8274 MPSC:

Interne oder externe Takterzeugung	J12	J13	J14	J15	J16
TxCA/RxCA extern	1-2	1-2	d.c.	d.c.	d.c.
TxCB/RxCB extern	d.c.	d.c.	1-2	1-2	d.c.
TxCA/RxCA intern TMR 0 = Standard	2-3	2-3	d.c.	d.c.	d.c.
TxCB/RxCB intern TMR 1 = Standard	d.c.	d.c.	2-3	2-3	1-2
TxCB/RxCB intern TMR 0	d.c.	d.c.	2-3	2-3	2-3
TxC/RxC both TMR 0	2-3	2-3	2-3	2-3	2-3

Wait Selektion für 8274 MPSC:

	J10	J11
RDYA	d.c.	1-2
RDYB	2-3	d.c.

A.6. Zeitkonstante für Watchdog-Timer

	J23
65 us	1-2
5,5 ms	n.c.

Standardmäßig wird die Anforderung von Refreshzyklen durch den Timer 2 alle 4 ms vorgenommen. Ist der Timer-Interrupt für den Refresh gesperrt worden und bleiben deshalb die Refreshimpulse aus, fordert der Watchdog-Timer ein NMI an, um die weitere Anforderung von Refreshzyklen zu erzwingen.

Wird die Anforderung von Refreshzyklen durch den DMA-Kanal 0 alle 15 us vorgenommen, sollte die Zeitkonstante für den Watchdog-Timer auf 65 us eingestellt werden; bleibt der Übertragungszähler des DMA-Kanals auf 0 stehen, und wird der DMA-Transfer nicht nach spätestens 65 us durch eine Interrupt-Service-Routine neu gestartet, fordert der Watchdog-Timer ebenfalls ein NMI an, um eine weitere Anforderung von Refreshzyklen zu erzwingen.

A.7. Reset Selektion

	J24
Power on Reset	1-2
Reset extern = Standard	n.c.

Standardmäßig erfolgt der Reset der miroCPU186 durch das PWRCLR-Signal des Systembus. Im "stand-alone"-Betrieb kann durch Stecken von Jumper 24 auch ein eigenes "Power on Reset" erzeugt werden, und/oder mit einem Taster am Anschluß S3-1 gegen Masse wird ein Reset der Baugruppe erreicht.

A.8. HLDA Selektion

	J19
HLDA enabled	1-2
HLDA disabled = Standard	2-3

A.9. DMA/Wait Selektion für Z8038 FIO

	J11	J20	J21
Port 1 Wait	d.c.	1-2	d.c.
Port 1 DMA (= standard)	3-4	2-3	d.c.
Port 2 Wait	d.c.	d.c.	1-2
Port 2 DMA	d.c.	d.c.	2-3

A.10. 8087 Selektion

Numeric Coprocessor	J22
8087 disabled = Standard	1-2
8087 enabled	2-3

B. Bestückung

B.1. Bestückungsplan

B.2. Lageplan der Steckbrücken

B.3. Stückliste

ICs
```
 1  x  Intel 80186 CPU, 8 MHz
 1  x  Intel 8274 MPSC, 4 MHz
 1  x  Zilog Z8038 FIO, 4 MHz
 2  x  Intel 2764-2, 64 kBit EPROM, 200ns
16  x  Mitsubishi M5K4164AP-15, 64 kBit DRAM, 150ns
 1  x  Intel 8288
 1  x  PAL 16R4CN
 1  x  PAL 16R6ACN
 1  x  MC 8T97, N 8T97 oder MC 6887
 1  x  NE 555, SE 555 oder ICM 7555
```

TTL-ICs
```
 1  x  74  F  04
 1  x  74 LS  09 oder ALS
 1  x  74  F  20
 1  x  74  F  32 oder S
 1  x  74 LS  74 oder ALS
 1  x  74  F  86
 1  x  74 LS 125
 2  x  74 LS 245
 4  x  74  S 531 oder 74  S 373
 1  x  74 LS 682
```

Dioden & Transistoren
```
 1  x  BC 547 oder MPS 2368
 1  x  BC 557 oder A5T3644
 1  x  1N 4148
 1  x  Unitrode TVS 305 oder BZW22/C5V6, 1.3W
```

Kondensatoren (Miniatur, RM 2.54)
```
 1  x  33u, 16V, Tantal
 1  x  4.7u, 16V, Tantal
23  x  330n, Vielschichtkeramik, Emcon 5020 ES 50 RD 334
21  x  100n, Vielschichtkeramik, Emcon 5020 ES 50 RD 104
 2  x  10n, Vielschichtkeramik, Emcon 5020 ES 50 RD 103
 1  x  270p, Keramik, EDPU/06
 2  x  100p, Keramik, EDPU/06
 2  x  22p, Keramik, EDPU/06
```

Widerstände (Miniatur, RM 7.62, 1/8W)
```
 1  x   100
 5  x  1.0k
 3  x  2.7k
 1  x   12k
 1  x   82k
 1  x  680k
```

Widerstandsnetzwerke (SIL, RM 2.54)
```
 2  x    22 x 5 SIL, L10-3 R22
 1  x    39 x 4 SIL, L08-3 R39
 3  x  1.0k x 7 SIL, L08-1 R1K
 1  x  4.7k x 8 SIL, L09-1 R4.7K
 1  x   10k x 7 SIL, L08-1 R10K
```

Quarze
1 x 16 MHz, Grundton, Parallelresonanz, HC-18/U

Elektromechanik
1 x VG-Messerleiste, 64-polig, Reihe a-c (S1a, S1c)
2 x Schrauben M2.5x12 & Muttern M2.5
1 x DIL-8 Präzisionsfassung
8 x DIL-14 Präzisionsfassung
17 x DIL-16 Präzisionsfassung
10 x DIL-20 Präzisionsfassung
2 x DIL-28 Präzisionsfassung
2 x DIL-40 Präzisionsfassung
1 x Textool 68-pol. Lead Chip Carrier Socket
1 x Stift-Feld, 1x1
1 x Stift-Feld, 1x2
3 x Buchsenleiste, 1x2
10 x Buchsenleiste, 1x3
1 x Buchsenleiste, 1x4
2 x Stift-Feld, 2x6
1 x Stift-Feld, 1x8
2 x Stift-Feld, 2x8
2 x Buchsenleiste, 2x20

...e benötigt von den Anschlüssen des ECB-Bus die
...D7 (Datenbus), A0...A7 (Adreßbus für I/O-
sowie mehrere Steuersignale.

...1. Signalbeschreibung

Stromversorgung

+5V — +5 Volt, Versorgungsspannung
+15V — +12...+15 Volt, Hilfsspannung
-15V — -12...-15 Volt, Hilfsspannung
GND — 0 Volt, Ground, Masse

dreß- und Datenbus

...D7 — Datenbus (8 Bit, Input/Output, aktiv "H")
..A7 — Adreßbus (8 Bit, Input, aktiv "H")

ffs-Steuersignale

I/O Request, Ein-/Ausgabe Zyklus,
Input, aktiv "L"
Write, Schreibsynchronisation,
Input, aktiv "L"
Read, Lesesynchronisation,
Input, aktiv "L"
Memory Request, Speicher-Zyklus,
Input, aktiv "L"
Machine Cycle One,
Interrupt Acknowledge Indicator,
put, aktiv "L"
wait Request,
Warte-Zyklus Anforderung,
Jutput, aktiv "L"
'stem Clock, Input

Acknowledge,
Anforderungsbestätigung,
aktiv "L"
t Request,
hungsanforderung,
tiv "L"
nable Input (Daisy Chain)
"H"
le Output (Daisy Chain)
"H"

C.1.2. Steckerbelegung

Bezeichnung	Pin
+5V	a1 & c1
+15V	a19
-15V	c15
GND	a32 & c32
D0	c2
D1	c14
D2	c4
D3	a4
D4	a5
D5	a2
D6	a3
D7	c3
A0	c5
A1	c7
A2	a6
A3	c6
A4	a7
A5	a8
A6	a9
A7	c9
IORQ/	a27
WR/	c22
RD/	c24
MREQ/	c30
M1/	a20
WAIT/	a10
CLK	c29
RESET/	c31
INT/	c21
IEI	c11
IEO	c16

D. Literaturverzeichnis

Die im folgenden aufgeführten Schriftstücke wurden durch Stichworte ergänzt, um bei Bedarf ein schnelles Auffinden bestimmter Themengruppen zu ermöglichen.

Ciarcia, Steve
"Build the Circuit Cellar MPX-16 Computer Sytem"
Part 1:
"Any peripheral device designed to be installed in the IBM Personal Computer can be plugged into this 8088-based system."
BYTE, Nov(1982), 78 - 114
/Design Concepts, Design Characteristics, Pragmatic Considerations, MPX-16 Overview (Figure: Simplified, high-level block diagram of the Circuit Cellar MPX-16 computer system),(Figure: Complete, detailed flow diagram of the MPX-16 system), . Intel 8088 Processor (Figure: Functional block diagram of the Intel 8088 microprocessor), Architecture of the 8088 (Figure: Programmer's model of the 8088's fourteen 16-bit registers), (Figure: Memory organization in the 8088), (Figure: Programmer's model of the 8088/8087 coprocessor combination), The 8087 Numeric Processor, MPX-16 Bus Structures (Figure: Timing diagram showing the relationship of the four major system clock signals), Reset and Clock-Generator Circuits, Nonmaskable-Interrupt Logic, DMA Controller and Bus Arbitration, To Be Continued, Next Month and Thereafter/.
Part 2:
"A continued description of an 8088-based system that shares its principles of operation with the IBM Personal Computer."
BYTE, Dec(1982), 42 - 78
/System Features (Figure: Map of memory-address-space allocation in the MPX-16, in 64K-byte increments), System Memory, ROM Configuration (Figure: Section 2 of the schematic diagram of the MPX-16 computer's main circuit board.), RAM Configuration, Interrupt Advantages, MPX-16 Interrupt Logic, Handling Interrupts, Interrupt Priorities, I/O-Expansion Channels, Oscillator Clock (BSYSCLK1), System Clock (BSYSCLKØ), System Reset (BRESET), Address Latch Enable (SYSALE), System Address Enable (SYSAEN), I/O Channel Ready (IOCHNLDRY/WAIT), System Memory Read (SYSMEMRD), System Memory Write (SYSMEMWR), System I/O Read (SYSIORD), System I/O Write (SYSIOWR), I/O-Channel (Parity) Error (I/OCHNLERR), System Address Bus (SYSAØ through SYSA19), System Data Bus (SYSDATØ throug SYSDA17), I/O Channel Interrupt Requests (IRQST2 through IRQST7), DMA Requests (DMARQST1 through DMARQST3), DMA Acknowledge Lines (DMACK1 through DMACK3), DMA Acknowledge Ø (BDMACKØ), DMA Termminal Count (TCNT), Peripheral Power, I/O-Decoder Logic, Next Month/.
Part 3:
"The final installment describing the design of the MPX-16, which is I/O-compatible with the IBM Personal Computer."
Byte Jan(1983), 54 - 68, 72 - 80
/MPX-16 Features, Parallel I/O Interface, Serial Interface (Figure: Section 4 of the schematic diagram of the MPX-16 computer), Counter/Timers, Floppy-Disk Drive Controller,

Clock-Signal Generation (Figure: Section 5 of the schematic diagram), Motor Control, Drive-Control Logic, Data-Write Logic, Data-Recovery Logic, CP/M-86 BIOS, BIOS Organization (Figure: Memory map if the CP/M-86 operating system as configured for the MPX-16), Disk I/O Routines, Disk-Definition Tables, In Conclusion, Next Month/.

Ciarcia, Steve:
"The Intel 8086"
BYTE Nov(1979), 14 - 24
/The Intel 8086 (Figures: An internal block diagram and pinout specifications of the Intel 8086, the 8086 register modelustrating the differences between the 8086 and the 8080, functional block diagram of internal data paths of the 8086), The Intel SDK-86, The SDK-86 Monitor (Figure: A detailed block diagram of the SDK-86 evalution board), In Conclusion/.

Cantrell, Thomas Woodward:
"An 8088 Procesoor for the S-100 Bus"
Part 2:
BYTE Oct(1980), 62 - 64, 68 - 88
/Interface, The Address Bus, Data Buses, The Control Bus, The Status Bus (Figure: Schematic diagram of the 8088 processor board for the S-100 bus), (Figure: The 8282 8-bit buffer/latch), Design Details, Other Signals: Pins 1 to 50 (Figure: An 8259A Programmable Interrupt Controller can be added to the system by following this logis diagram), Other signals: Pins 51 to 100, Implementation, The Board, The Front Panel, Debuggging the Front Panel (Figure: The front panel), Debugging the Processor Board, Hints/.
Part 3:
BYTE Nov(1980), 340 - 360
/MON88 Philoosophy, MON88 Organization (Figure: Memory map for the MON88 monitor), (Figure: High-level flowchart for MON88 program), Environment Dependence, Command Summary, Commands (Figure: Relative performance of several 8- and 16-bit microprocessors), I/O Drivers, Adding or Removing Commands, Notes on Performance, Finale/.

Casper, Rolf:
"Entwicklung einer universellen Mikrorechnerkarte mit der CPU 8086 und dem Arithmetik-Baustein 8087"
Diplomarbeit TU Braunschweig b. Prof. Dr.-Ing. W. Leonhard
WS 1981/82
/Der Mikroprozessor 8086, Die Ausführungseinheit, Der Registersatz, Die Bus-Interface-Einheit, Die Speicheradressierung, Der Bus-Zyklus, Die Behandlung von Interrupts, Die Multiprozessorfähigkeiten des 8086, Der Initialzustand der CPU, Der Arithmetik-Prozessor 8087, Das Hardware-Interface, Interne Struktur des 8087, Die Numerische Ausführungseinheit, Die Kontroll-Einheit, Datenformate des 8087, Der Einkarten-Mikrorechner, Anforderungen, Schaltungsbeschreibung, Der Taktbaustein 8284, Der Bus-Controller 8288, Die Adressdekodierung, Die Wartesteuerung, Der Timer 8253, Der Interrupt-Controller 8259, Die Speicherbausteine, Der Pulsgeber, Interface zur Rechnererweiterung, Die Multi-Prozessor-Schnittstelle, Vorgänge während eines INTA-Zyklus, Programmierhinweise,

Praktischer Aufbau eines Mikrorechners, Programmbeispiele, Testprogramm zur Rechnerkopplung, Demonstrationsprogramm und Testhilfe für den Arithmetik-Prozessor 8087, Zusammenfassung, Literaturverzeichnis, Anhang: Layout des Mikro-

Praktischer Aufbau eines Mikrorechners, Programmbeispiele, Testprogramm zur Rechnerkopplung, Demonstrationsprogramm und Testhilfe für den Arithmetik-Prozessor 8087, Zusammenfassung, Literaturverzeichnis, Anhang: Layout des Mikrorechners, Bestückungspläne, Stückliste, Kontaktbelegungen, Gesamtschaltplan/.

Geyer, Johann:
"Der Nachfolger, 16-bit-Mikroprozessor ersetzt CPU-Board", elektronik praxis, August 1982, H. 8.

Geyer, Johann:
"Schneller Mikroprozessor arbeitet mit virtuellem Adreßbereich", Elektronik 1982, H. 5.

Huse, Horst:
"Multi-Mikrocomputer-Systeme modular aufgebaut", Elektronik 1982, H. 2, S. 76.

Info:
"iAPX 188-CPU - Leiterplatte auf einem Chip integriert", Elektronik Informationen, 1982, Nr. 10.

Intel Corporation:
"The 8086 Family User's Manual", Technical Manual, Intel Corporation, Santa Clara, 1982.

Intel Corporation:
"Asynchronous Communication with the 8274", Intel Application Note AP-134, Intel Corporation, Santa Clara, 1982.

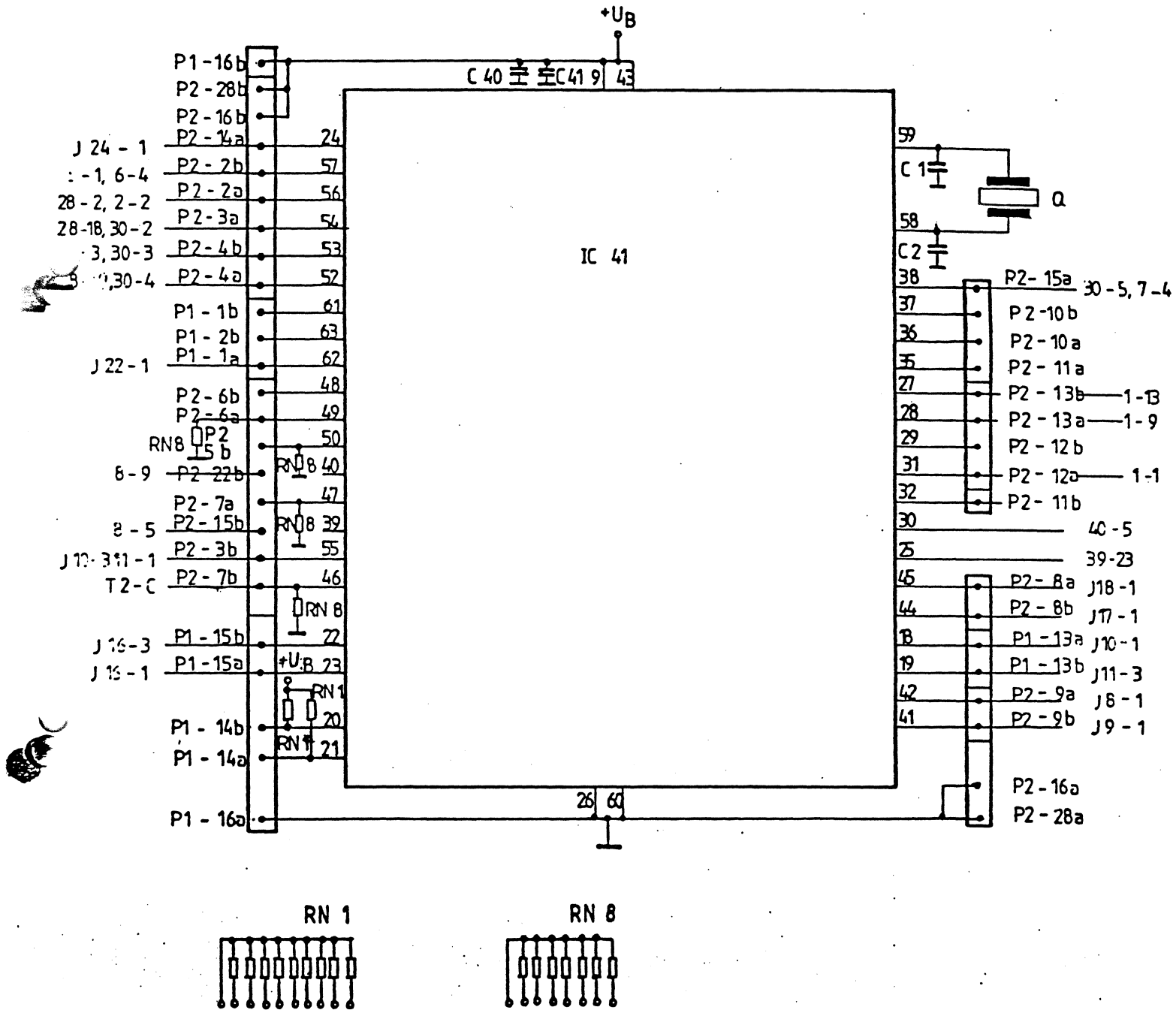
Schmidt, Hermann:
"Multi-Mikroprozessor-Systeme", Elektronik 1982, H. 2, S. 87-95.

Sichtling, Jürgen:
"Entwurf und Aufbau einer Mikrorechnerkarte auf der Basis des 8086 mit 'Multimaster-System-Bus' und 'Resident Bus'", Diplomarbeit TU Braunschweig bei Prof. Dr.-Ing. H. Weh, 1982.

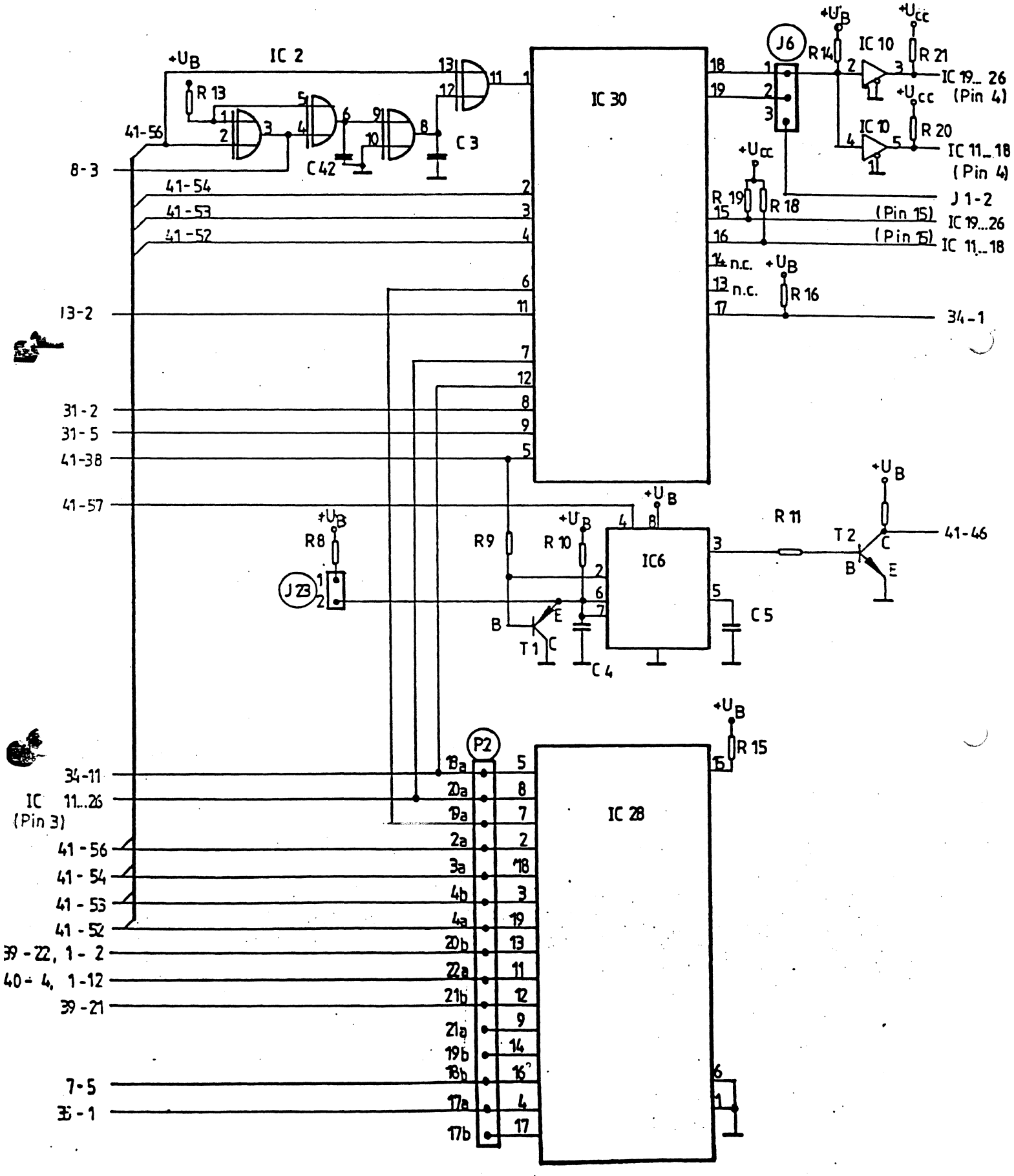
Zilog Incorporation:
"Technical Manual: Z8038 Z8000 Z-FIO FIFO Input/Output Interface Unit", Zilog Incorporation, Campbell, 1981.

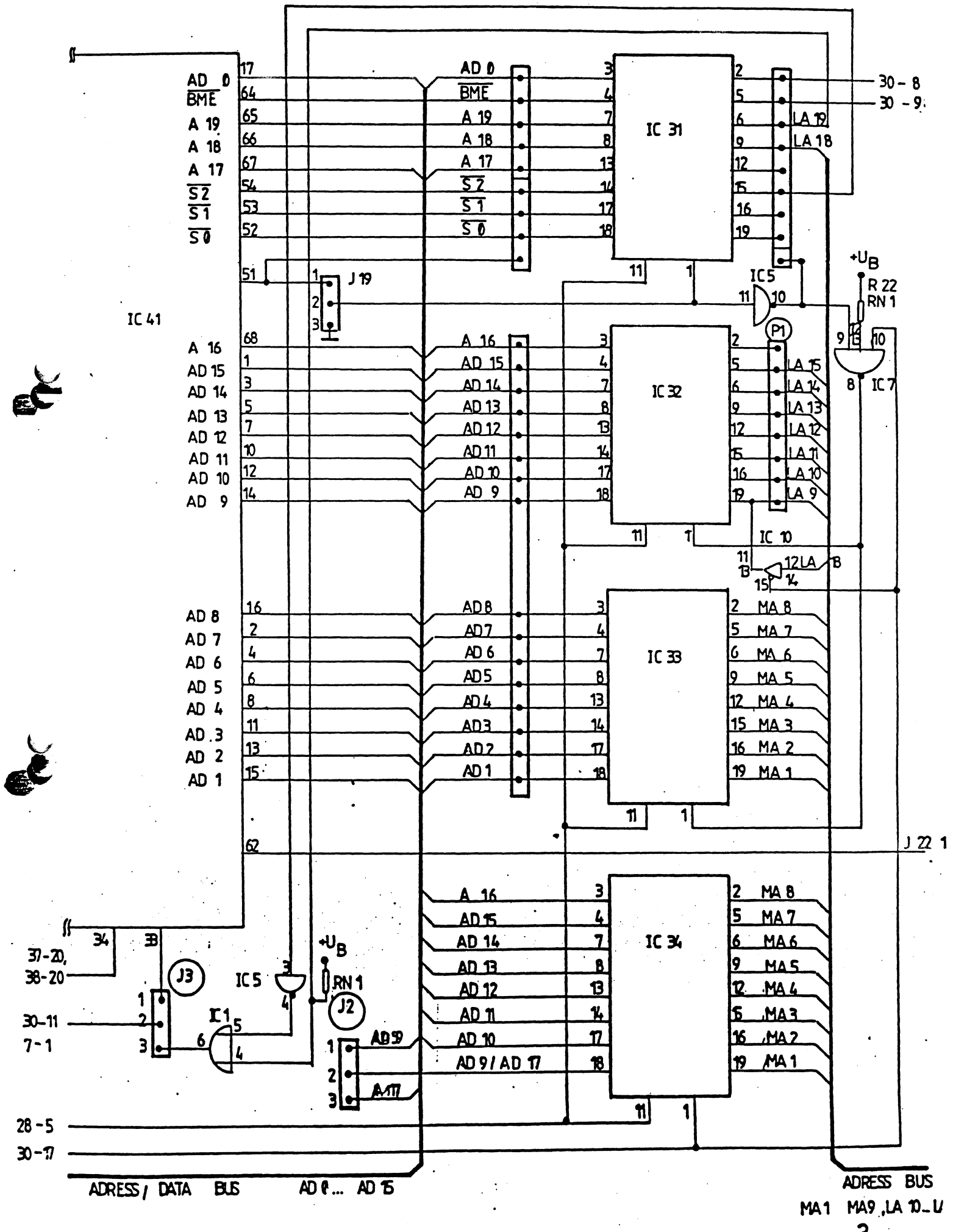
Zingal, Tony:
"Intel's 80186: A 16-Bit Computer on a Chip", BYTE, April 1983, S. 132-146.

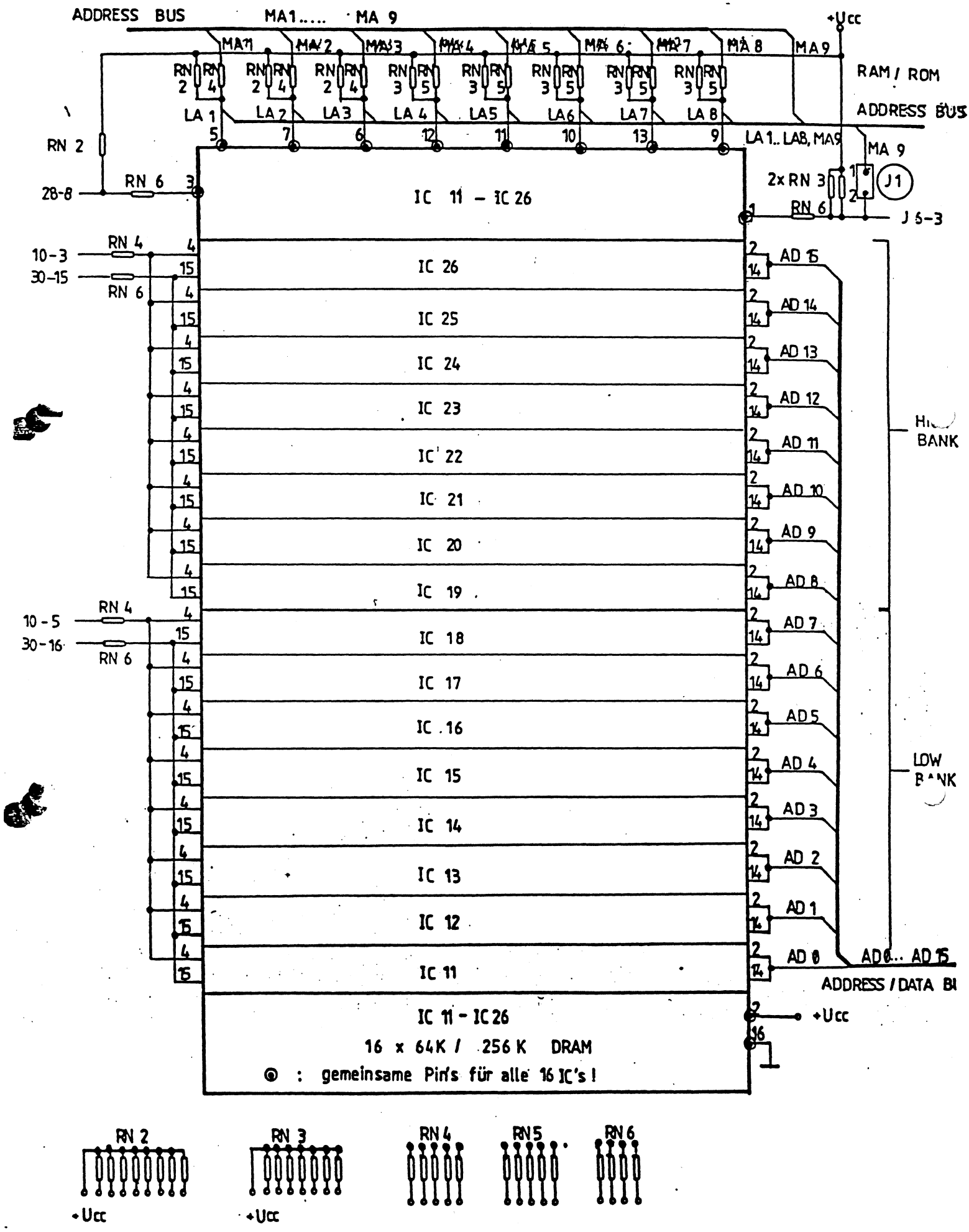
E. Stichwortverzeichnis

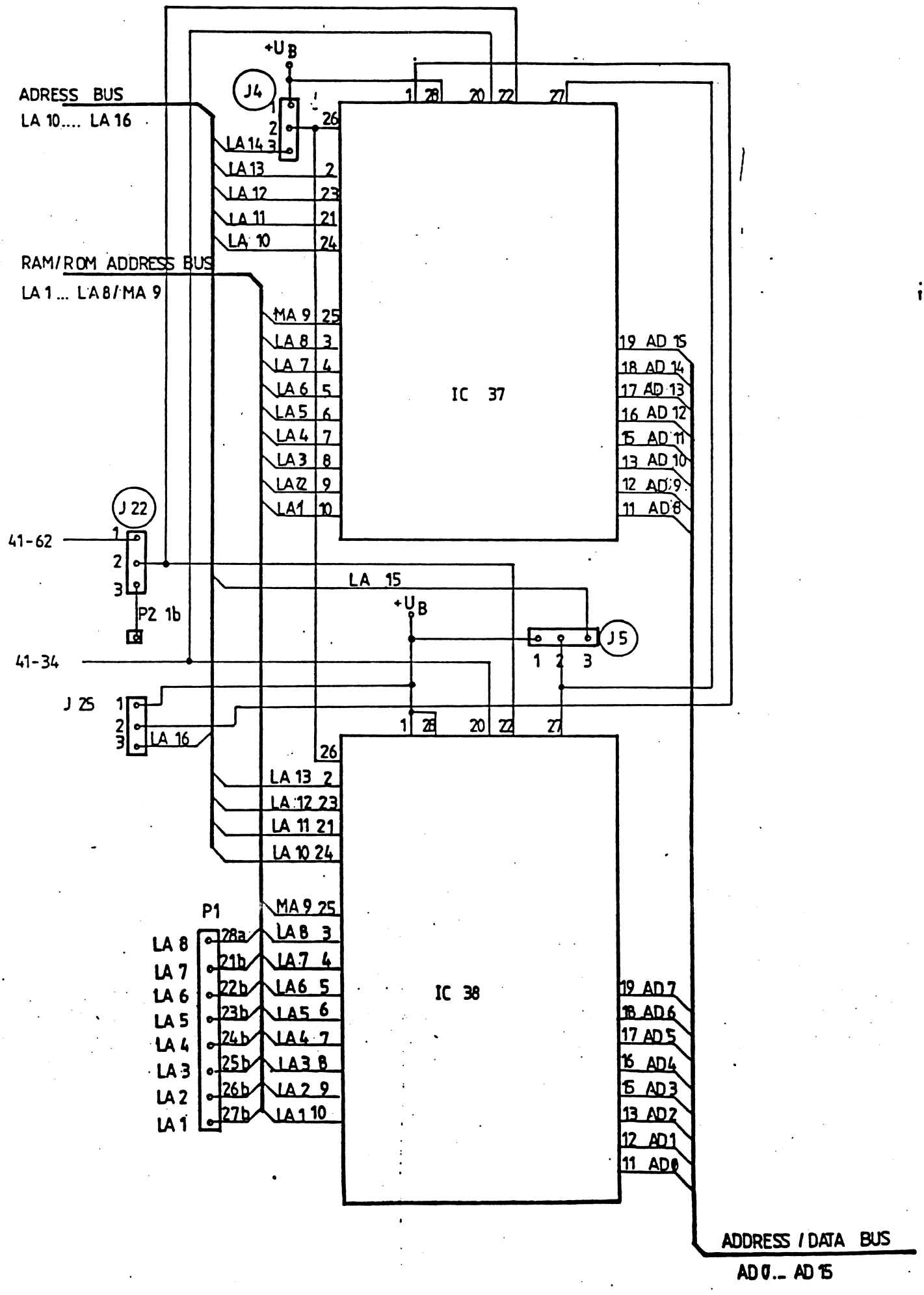

+UB
C 40
C41
9
43
IC 41
P1 - 16b
P2 - 28b
P2 - 16b
P2 - 14a
P2 - 2b
P2 - 2a
P2 - 3a
P2 - 4b
P2 - 4a
P1 - 1b
P1 - 2b
P1 - 1a
P2 - 6b
P2 - 6a
RN8
P2 15 b
P2 - 7a
P2 - 15b
P2 - 3b
P2 - 7b
RN 8
P1 - 15b
P1 - 15a
+UB
RN 1
P1 - 14b
P1 - 14a
P1 - 16a
24
57
56
54
53
52
61
63
62
48
49
50
40
47
39
55
46
22
23
20
21
26
60
59
C 1
Q
58
C2
38
37
36
35
27
28
29
31
32
30
25
45
44
18
19
42
41
P2- 15a
P2 -10 b
P2 - 10 a
P2 - 11 a
P2 - 13b
P2 - 13a
P2 - 12 b
P2 - 12a
P2 - 11b
P2 - 8a
P2 - 8b
P1 - 13a
P1 - 13b
P2 - 9a
P2 - 9b
P2 - 16a
P2 - 28a
RN 1
RN 8

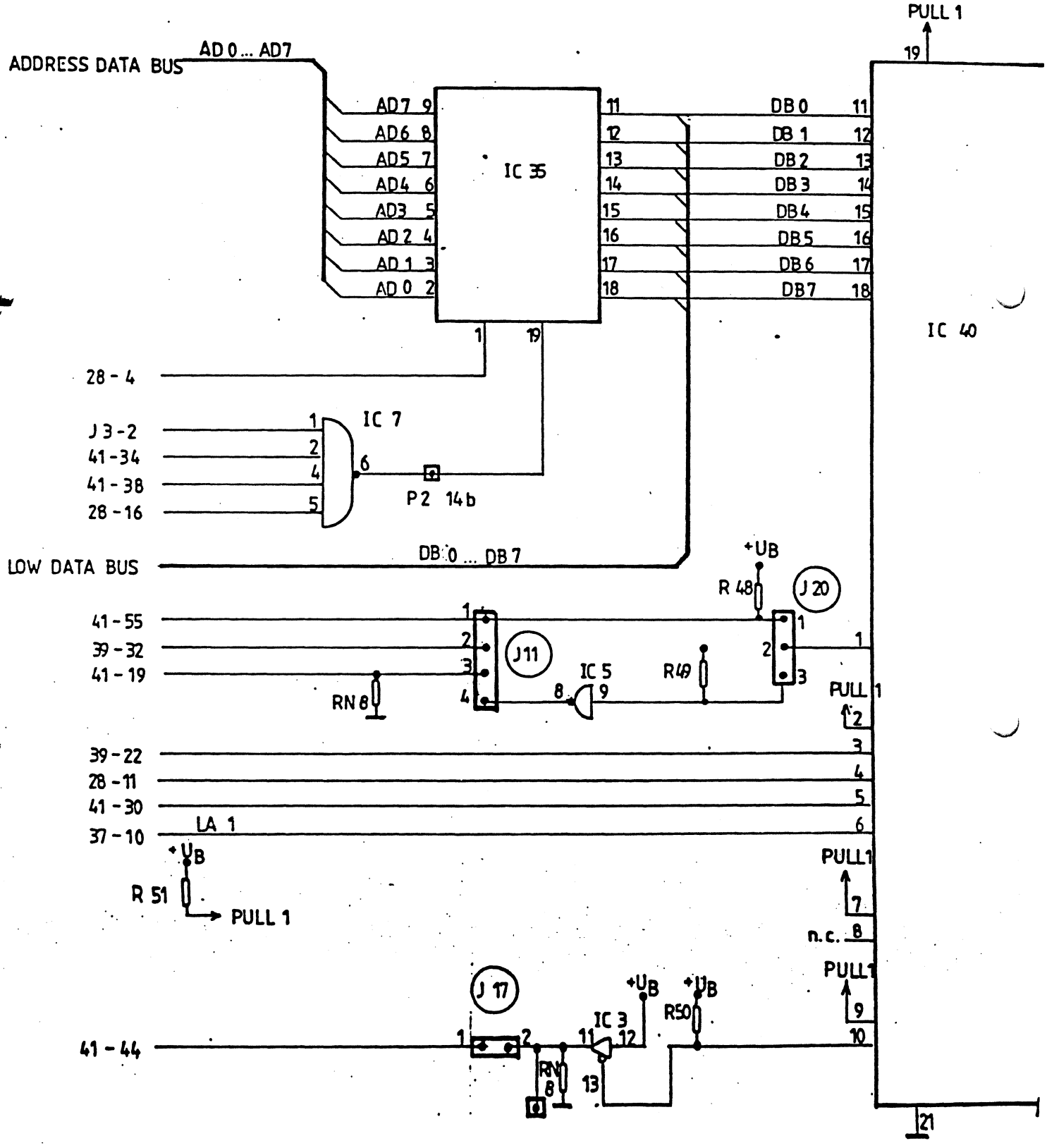
PAL-RAM-Controller
BUS Controller
Watchdog Timer

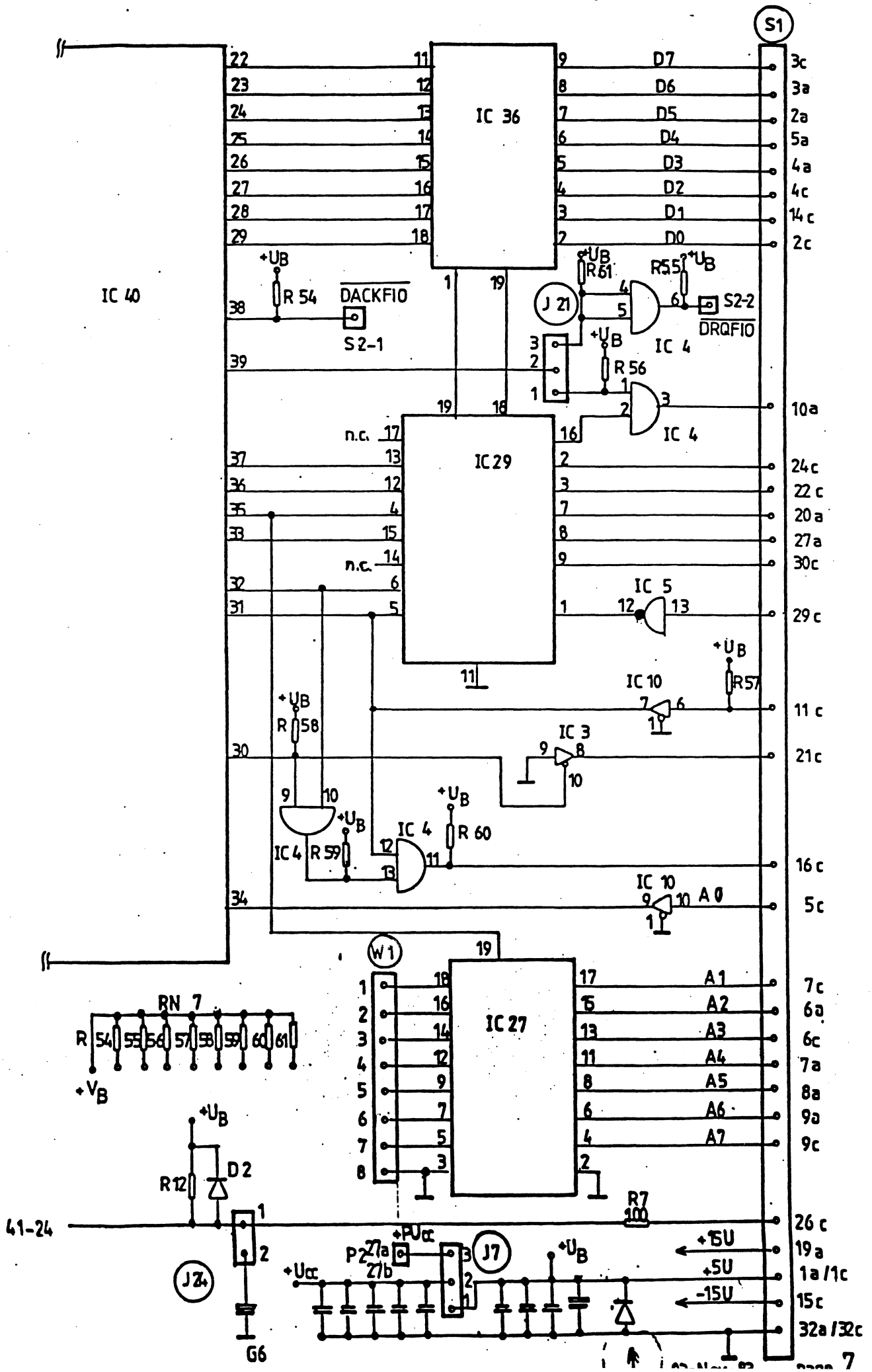


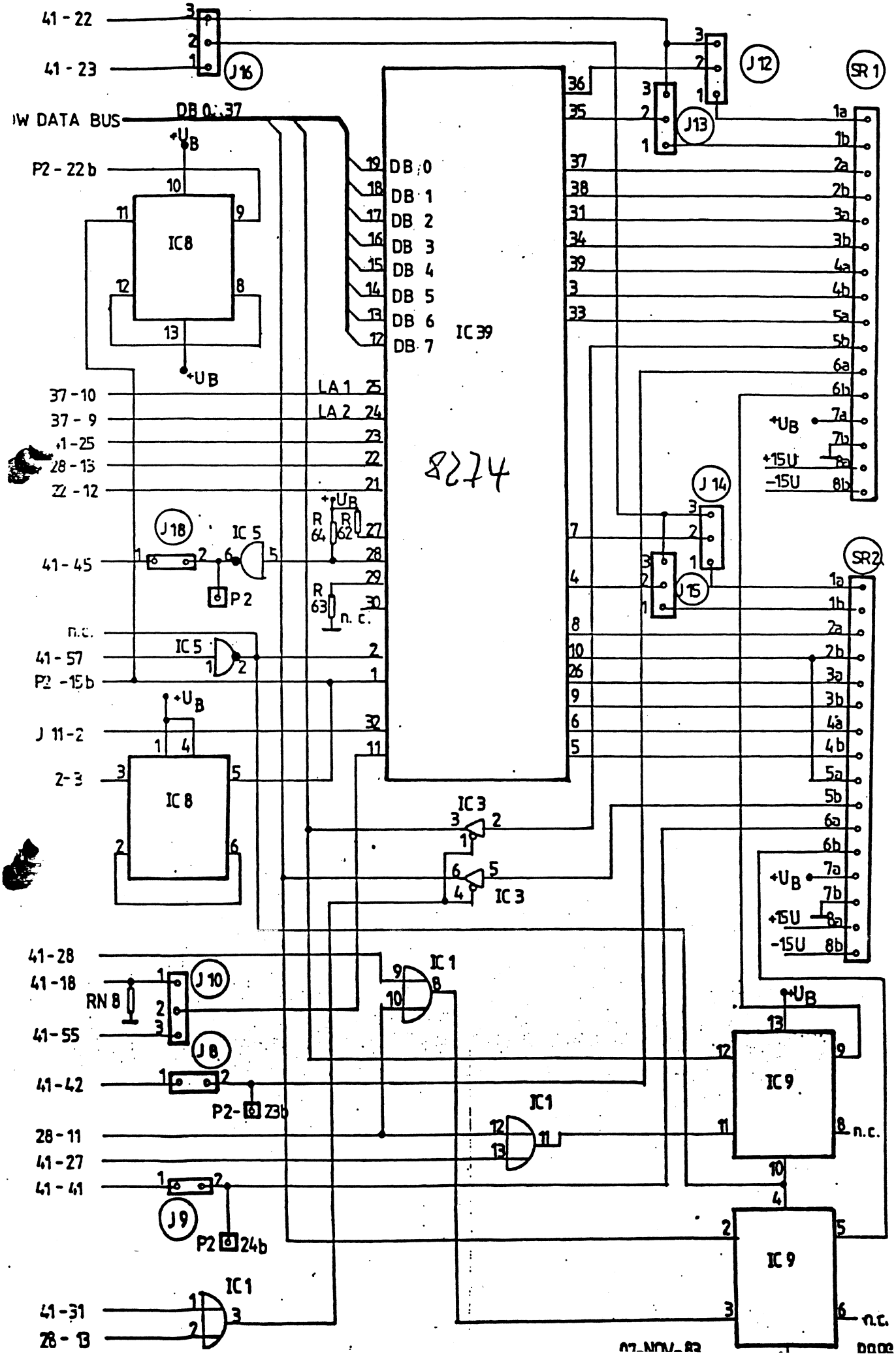

AD 0
BME
A 19
A 18
A 17
S2
S1
S0
IC 41
A 16
AD 15
AD 14
AD 13
AD 12
AD 11
AD 10
AD 9
AD 8
AD 7
AD 6
AD 5
AD 4
AD 3
AD 2
AD 1
J 19
IC 31
IC 32
IC 33
IC 34
IC 5
IC 10
IC 7
P1
+UB
R 22
RN 1
LA 19
LA 18
LA 15
LA 14
LA 13
LA 12
LA 11
LA 10
LA 9
MA 8
MA 7
MA 6
MA 5
MA 4
MA 3
MA 2
MA 1
30 - 8
30 - 9
J 22 1
J3
J2
IC 1
RN 1
AD 0
AD 17
37-20,
38-20
30-11
7-1
28-5
30-17
ADRESS / DATA BUS
AD 0 ... AD 15
ADRESS BUS
MA 1 MA 9, LA 10-19


ADDRESS BUS MA1..... MA 9
+Ucc
MA1 MA2 MA3 MA4 MA5 MA6 MA7 MA8 MA9
RAM / ROM ADDRESS BUS
RN 2 RN 4 RN 3 RN 5
LA 1 LA 2 LA 3 LA 4 LA5 LA6 LA7 LA 8
LA 1.. LA8, MA9
MA 9
2x RN 3
J1
RN 6
J 5-3
28-8
10-3
30-15
10-5
30-16
IC 11 – IC 26
IC 26
IC 25
IC 24
IC 23
IC 22
IC 21
IC 20
IC 19
IC 18
IC 17
IC .16
IC 15
IC 14
IC 13
IC 12
IC 11
AD 15
AD 14
AD 13
AD 12
AD 11
AD 10
AD 9
AD 8
AD 7
AD 6
AD 5
AD 4
AD 3
AD 2
AD 1
AD 0
AD0... AD 15
ADDRESS / DATA BU
HIGH BANK
LOW BANK
IC 11 - IC 26
16 x 64K / 256 K DRAM
◎ : gemeinsame Pin's für alle 16 IC's !
RN 2
RN 3
RN 4
RN 5
RN 6
+Ucc
+Ucc


+UB
J4
ADRESS BUS
LA 10.... LA 16
LA 14
LA 13
LA 12
LA 11
LA 10
RAM/ROM ADDRESS BUS
LA 1 ... LA 8/MA 9
MA 9
LA 8
LA 7
LA 6
LA 5
LA 4
LA 3
LA 2
LA 1
IC 37
19 AD 15
18 AD 14
17 AD 13
16 AD 12
15 AD 11
13 AD 10
12 AD 9
11 AD 8
J 22
41-62
P2 1b
LA 15
+UB
J 5
41-34
J 25
LA 16
IC 38
P1
LA 8
LA 7
LA 6
LA 5
LA 4
LA 3
LA 2
LA 1
19 AD 7
18 AD 6
17 AD 5
16 AD 4
15 AD 3
13 AD 2
12 AD 1
11 AD 0
ADDRESS / DATA BUS
AD 0... AD 15


PULL 1
19
ADDRESS DATA BUS
AD 0 ... AD7
AD7 9
AD6 8
AD5 7
AD4 6
AD3 5
AD2 4
AD1 3
AD0 2
IC 35
11
12
13
14
15
16
17
18
DB 0 11
DB 1 12
DB 2 13
DB 3 14
DB 4 15
DB 5 16
DB 6 17
DB 7 18
IC 40
1
19
28 - 4
IC 7
J 3 - 2
41 - 34
41 - 38
28 - 16
P2 14b
LOW DATA BUS
DB 0 ... DB 7
+UB
R 48
J 20
41 - 55
39 - 32
41 - 19
J11
RN 8
IC 5
R49
PULL 1
39 - 22
28 - 11
41 - 30
37 - 10
LA 1
+UB
R 51
PULL 1
PULL1
n.c.
PULL1
J 17
+UB
+UB
R50
IC 3
41 - 44
RN 8
21


S1
IC 36
IC 40
IC 29
IC 27
IC 4
IC 5
IC 10
IC 3
D7
D6
D5
D4
D3
D2
D1
D0
DACKFIO
DRQFIO
S2-1
S2-2
J 21
J24
J7
W1
RN 7
R 54
R 55
R 56
R 57
R 58
R 59
R 60
R 61
R12
R7
D2
G6
A0
A1
A2
A3
A4
A5
A6
A7
+UB
+VB
+Ucc
+PUcc
+15U
+5U
-15U
41-24
n.c.
10a
24c
22c
20a
27a
30c
29c
11c
21c
16c
5c
7c
6a
6c
7a
8a
9a
9c
26c
19a
1a/1c
15c
32a/32c


41 - 22
41 - 23
J16
J12
J13
SR1
DB 0..37
+UB
P2 - 22b
IC8
+UB
37 - 10
37 - 9
41 - 25
28 - 13
22 - 12
DB 0
DB 1
DB 2
DB 3
DB 4
DB 5
DB 6
DB 7
IC 39
LA 1
LA 2
8274
+15U
-15U
J14
J15
SR2
J18
IC 5
41 - 45
P2
R 64
R 62
R 63
n. c.
41 - 57
P2 - 15b
J 11-2
2-3
IC 3
41 - 28
41 - 18
RN 8
J10
41 - 55
J8
41 - 42
P2 - 23b
IC 1
28 - 11
41 - 27
41 - 41
J9
P2 24b
IC 9
41 - 31
28 - 13
n.c.