oettle & reichler
datentechnik GmbH

Völkstraße 27
8900 Augsburg 1

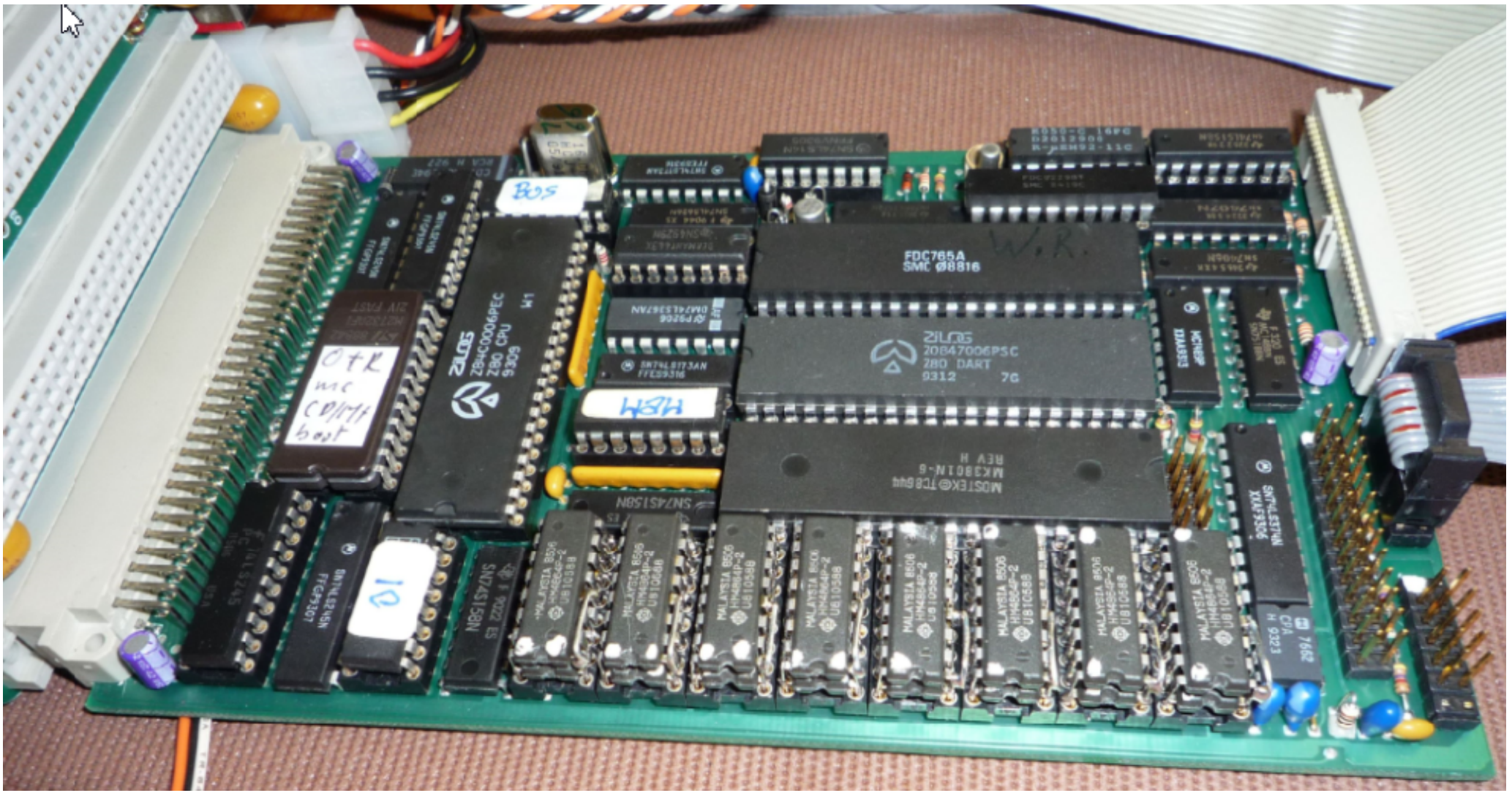
Tel.: (0821) 15 46 32

E P C

Einplatinencomputer

H A N D B U C H

EPC ist ein EinPlatinenComputer

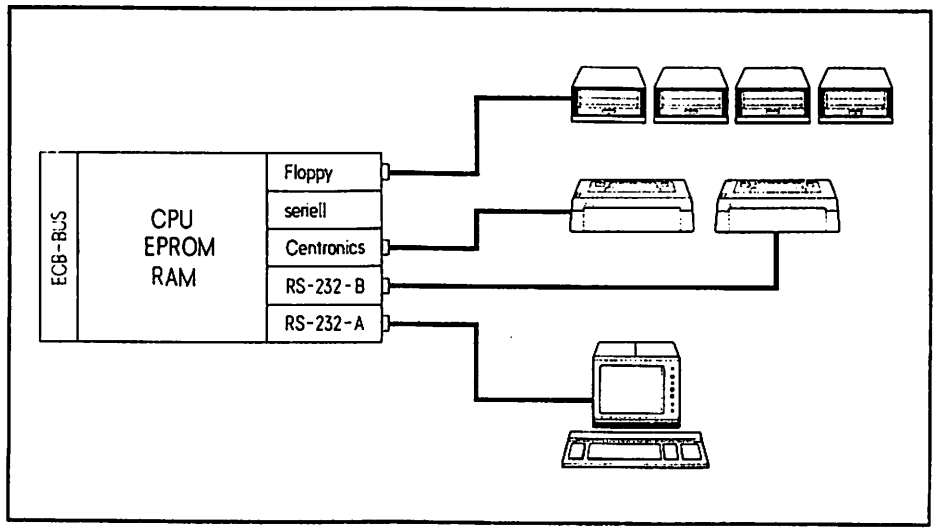


Inhaltsverzeichnis

Einsatzmöglichkeiten

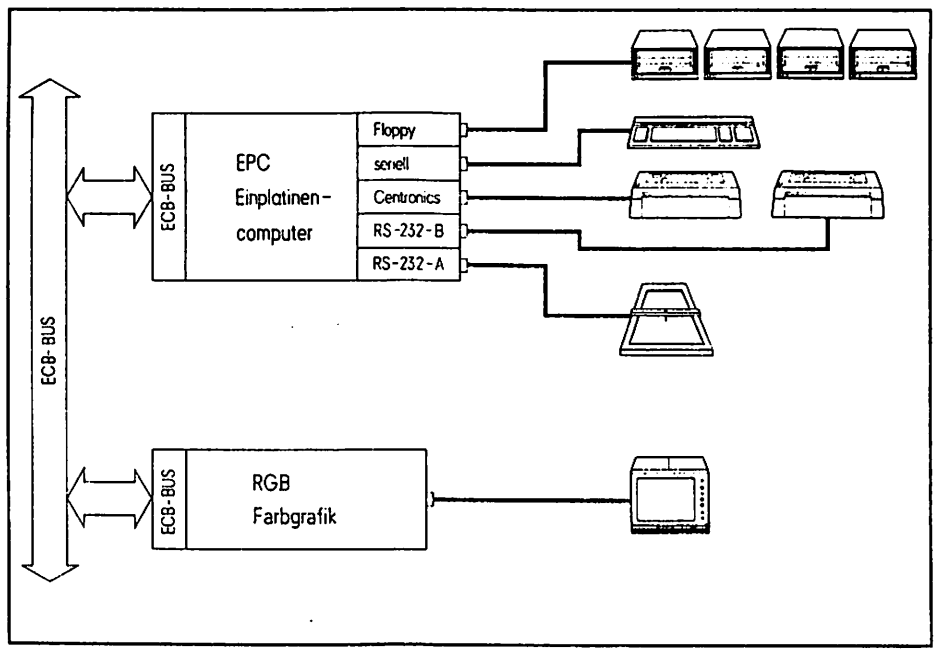
stand-alone System

Der EPC stellt sämtliche Funktionen für ein komplettes CP/M Plus Computersystem zur Verfügung. An Peripherie wird nur ein handelsübliches Terminal und mindestens ein Floppy Laufwerk benötigt.



Master Baugruppe

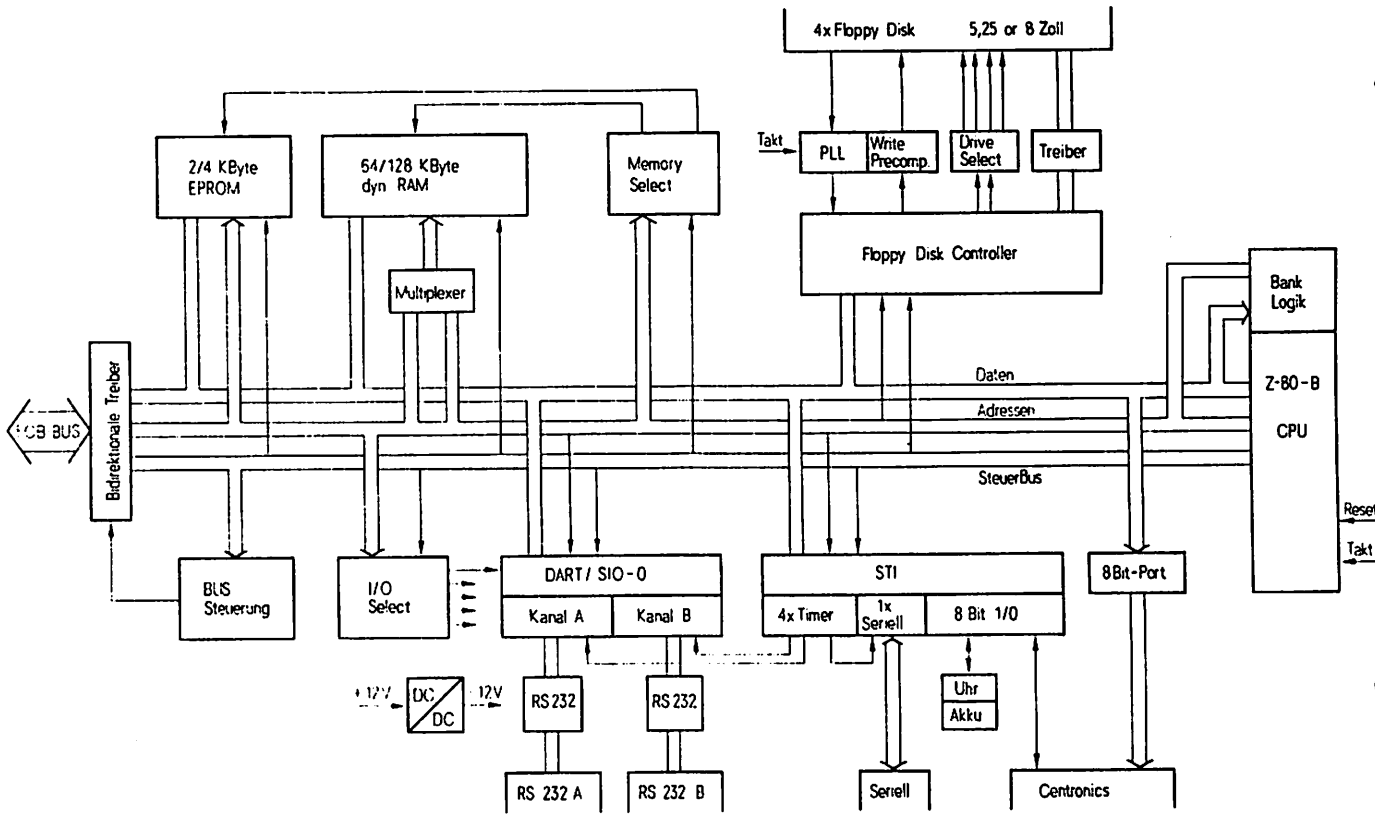
Dank der ECB-BUS Schnittstelle kann der EPC aus einer reichhaltigen Palette von Zusatzkarten erweitert werden. Zusätzlich notwendig wird dann eine ECB-BUS Rückwandverdrahtung und ein Einschubgehäuse. Es kann wahlweise ein Terminal über RS-232 oder über den ECB-BUS eine Video-Karte mit seperatem Monitor und Tastatur angeschlossen werden.



I/O Baugruppe

Sämtliche auf dem EPC befindlichen Funktionseinheiten wie RAM, Floppy und I/O können auch von einer externen CPU angesprochen werden. Der EPC läßt sich so in ein bereits bestehendes ECB-System einstecken und als I/O und RAM Karte ansprechen. Dies ermöglicht einen gezielten Test der Baugruppe beim Aufbau. Ebenso kann auf diese Weise spezielle Software für den EPC in einem bestehenden System entwickelt und getestet werden.

Schaltungsübersicht



1 M-Byte Adressbereich

Eine Banking-Schaltung erzeugt vier zusätzliche Adressen A16-A19 und vergrößert somit den maximalen Adressbereich auf 1 M-Byte. Über einfache I/O-Befehle kann zwischen 16 verschiedenen 64K Pages umgeschaltet werden.

Der Speicher

Der EPC verfügt über getrennte ROM- und RAM-Bereiche. Der RAM-Speicher läßt sich in zwei Schritten zu 64 K-Byte auf maximal 128

K-Byte ausbauen. Er belegt standardmäßig die untersten beiden Pages im 1 M-Byte Adressbereich. Eine 24 polige Fassung nach JEDEC kann wahlweise mit einem 2 bzw. 4 K-Byte großen Festwertspeicherbestückt werden und belegt normalerweise die untersten Adressen. Der verfügbare Speicher läßt sich in Blöcken zu 16 K-Byte beliebig im 1 M-Byte Bereich anordnen.

Die Disketten Steuerung

Der EPC ist in der Lage maximal 4 Laufwerke gemischt zwischen 3 und 8 Zoll zu verwalten. Alle gängigen Formate wie FM/MFM, SS/DS sowie Größe und Anzahl der Sektoren können softwaremäßig eingestellt werden. Am vorderen Kartenrand wird eine Shugart kompatible Schnittstelle bereitgestellt, die den direkten Anschluß der Laufwerke über ein Flachbandkabel gestattet.
Besonderer Wert wurde auf die sichere Umwandlung des Datenstroms von und zur Diskette gelegt. Beim Lesen gewährleistet eine integrierte PLL-Schaltung, beim Schreiben eine Precompensation-Schaltung den sicheren Datentransport . Eine Motor ein/aus Schaltung sorgt für eine lange Lebensdauer der Disketten und einen geräuscharmen Betrieb.

I/O Schnittstellen

Alle Schnittstellen sind am vorderen Kartenrand der Baugruppe auf 2-reihigen Postensteckern herausgeführt. Die Belegung wurde dabei so gewählt, daß Normstecker über Flachbandkabel in Schneidklemmtechnik direkt 1:1 gequetscht werden können.
Eine parallele Schnittstelle nach der Centronics Norm erlaubt den Anschluß eines Druckers.
Zwei serielle, nach RS-232/V24 gepufferte, Schnittstellen sind in der Lage ein asynchrones Datenformat in Vollduplexbetrieb zu liefern. Die Pufferung umfaßt neben dem Datenstrom auch die Signale CTS und RTS. Die dafür notwendige -12 Volt Spannung wird auf der Karte über einen DC-DC Wandler erzeugt. Die Baudrate läßt sich für beide Kanäle getrennt frei programmieren. Wahlweise ist auch eine synchrone Datenübertragung für die Formate IBM BiSynch, HDLC oder SDLC möglich.
Eine dritte serielle Schnittstelle mit TTL-Pegel eignet sich ebenfalls für asynchrone vollduplex Datenübertragung.

Die Echtzeituhr

Ein akku gepufferter Uhrenbaustein stellt dem System Echtzeit zur Verfügung. Diese kann unter CP/M Plus in Dateien eingebunden werden. Folgende Angaben stehen zur Verfügung: (Schalt-)Jahr, Monat, Wochentag, Tag, Stunde, Minute und Sekunde.

Hardwarebeschreibung

Stromversorgung

Die Karte benötigt zwei Betriebsspannungen: +5 und +12 Volt, wobei die +12 Volt nur für die RS-232 Schnittstelle erforderlich sind. -12 V werden von einem Spannungsinverter ICL 7660 (IC 33) erzeugt. Dem RS-232 Treiber MC 1488 (IC 36) werden somit ca. +/- 9 Volt bereit gestellt.

Takt

Auf der Karte befinden sich zwei Quarzoszillatoren aufgebaut aus einem LS 626 Baustein und zwei Quarzen.
Der Systemtakt (6 MHz) wird direkt aus dem Oszillator abgeleitet und über eine Transistor-Treiberstufe aktiv auf High- Potential gezogen. IC 05 (74 LS 194) teilt den Systemtakt durch den Faktor zwei und versorgt damit den Timer Eingang TCLK des STI Bausteins (IC 26).
Der zweite Oszillator erzeugt den Takt (16 MHz) für den Floppy Interface Baustein FDC 9229 (IC 23).

Reset

Ein Power-up Reset wird mittels eines RC Gliedes und zwei LS 14 Gatter (IC 19) erzeugt und über eine Open-Collector-Stufe auf PWCLR (Pin 26c am BUS-Stecker) geführt. Dieses Signal wird zum Zurücksetzen der CPU, STI, DART, FDC und dem LS 173 (Banking) benutzt. Der Eingang RESET am BUS-Stecker (31c) kann zum Anschluß eines Reset Schalters verwendet werden.

Wait

Die Wait Schaltung kann bei I/O- und Speicheroperationen einen Wait-Stait einfügen. Der Eingang (Pin 10) des 74 LS 194 (IC 05) ist mit dem Wait-Request Signal des I/O- und des Memory-PROMs beschaltet. Wird dieses aktiv, wechselt IC 05 vom Load- in den Shift-Zustand über. Mit der nächsten steigenden Flanke des Systemtaktes wird ein Wait Stait ausgegeben.
Durch Programmieren des I/O PROM's kann festgelegt werden bei welchem der insgesamt 256 I/O Adreßen ein Wait-Zyklus generiert wird. Der Memory PROM erlaubt es jedem 16 K-Byte Speicherblock innerhalb des 1 M-Byte Speicherbereichs ein Wait zuzuordnen.
Standardmäßig wird bei jedem I/O-Zugriff (Ausnahme uPD 765) und bei Speicherzugriffen auf den BOOT-EPROM jeweils ein Wait-Stait eingefügt. Speicherzugriffe auf externe Speicherbaugruppen und den internen dynamischen RAM-Speicher verursachen keine Wait's.

Banking Speicher Verwaltung

Diese Schaltung erlaubt es der CPU 1 M-Byte Speicher direkt zu adressieren. Ein 4 Bit Ausgabeport IC 14 (74 LS 173) erzeugt die vier Adressen A16-A19 (Pageadressen). Über I/O-Ausgabe-Befehle (D0 = A16, D3 = A19) läßt sich der Inhalt dieses Ports und damit die Bank-Adresse bestimmen.
Der Memory-PROM ist standardmäßig so ausgelegt, daß die obersten 16 K-Byte im 64 K-Byte Adreßraum der CPU unabhängig von den Adressen A16-A19 eingeblendet sind. Beim Wechsel auf eine andere Page werden also nur die untersten 48 K-Byte ausgetauscht.
Nach einem Reset ist der LS 173 Baustein zurückgesetzt und die unterste 64 K-Byte Page wird angesprochen.

Memory Select

Zur Adreß-Dekodierung des 24 poligen Byte-Wide Sockel und der 128 K-Byte RAM wird ein 256x4 PROM (IC 13) verwendet. Durch entsprechende Programmierung kann die Adreßlage der beiden Speicher bestimmt werden. Die Adreßlage läßt sich dabei in 16 K-Byte Schritten frei im maximal möglichen Adreßraum von 1 M-Byte bestimmen. Der BOOT Eingang am PROM erlaubt das Ein- und Ausschalten einzelner Speicherbereiche über die System-Software. Der Memory PROM ist standardmäßig so programmiert, daß der auf der Karte befindliche RAM die untersten 128 K-Byte belegt. Der EPROM ist den untersten 16 K-Byte überlagert. Führt der BOOT Eingang High Pegel, ist der EPROM eingeblendet. Wird I7 des STI Bausteins auf Low Pegel programmiert, wird der BOOT-EPROM abgeschaltet und der adressgleiche 16 K-RAM Speicher aktiviert. Der überhalb von 04000H liegende RAM ist unabhängig von dem BOOT Eingang immer aktiv.

In Abhängigkeit der 8 Eingangssignale und der beiden CE Signale werden die 4 Ausgänge RAS1, RAS2, MEM-SEL und CE-EPROM aktiv (Low) geschaltet:
Zwei der Ausgänge werden direkt als RAS Signale für die beiden 64 K-Byte RAM Blöcke benützt. Diese sind aktiv, wenn ein Speicherzugriff auf die entsprechende Adresslage oder ein Refresh erfolgt. Der Ausgang M-SEL führt Low Pegel, wenn auf einen internen Speicher des EPC zugegriffen wird und dient zur BUS-Steuerung. Dieses Signal wird als Open Collector Ausgang auch an den BUS-Stecker DESLCT (Pin 26a) geführt. Das CAS Signal für den dynamischen Speicher wird ebenfalls aus diesem Signal abgeleitet.
Das CE-Signal für den 24 poligen Sockel wird an die Wait Schaltung weitergleitet, welche einen Wait-Stait generiert.

Signalbelegung des Memory PROMs IC 13 (24 SA 10):

A7	Pin 15	A15	D3	Pin 9	CE EPROM
A6	Pin 1	BOOT	D2	Pin 10	RAS 1
A5	Pin 2	A14	D1	Pin 11	Memory Select
A4	Pin 3	A19	D0	Pin 12	RAS 0
A3	Pin 4	A16			
A2	Pin 7	RFSH			
A1	Pin 6	A18			
A0	Pin 5	A17			

Der dynamische Speicher

Als Massenspeicher werden kostengünsige dynamische 64K * 1 Speicher eingesetzt. Wichtig ist, daß nur RAM's mit einem 7 Bit Refresh (128 Refresh-zyklen) und bei 6 MHz Systemfrequenz 150ns Typen verwendet werden. Wahlweise können 8 (64 K-Byte) oder 16 (128 K-Byte) Speicher bestückt werden.
Um 128 K-Byte RAM auf diesem kleinen Raum zu realisieren, werden jeweils zwei 64 K-Bit Speicher übereinander montiert. Pin 4 (RAS) des oberen Speichers ist nicht mit dem unteren verbunden, sondern wird getrennt über den unbelegten Pin 1 herausgeführt. Der Anschluß 1 darf dabei keine interene Verbindung haben. Alle anderen Anschlüsse werden direkt mit dem unteren IC verbunden.

Ansteuerung

Die dynamischen Speicher werden im 'early Write' Modus betrieben. Das heißt, bei einem Schreibzugriff ist das Write Signal bereits mit der fallenden Flanke von CAS aktiv (Low). Dies hat den Vorteil, daß die Datentreiber (D out) beim Schreiben im Tri-State Zustand bleiben. Din und Dout müssen deshalb nicht getrennt gepuffert werden und lassen sich so einfach verbinden.
Die Adressen A0-A15 werden über zwei invertierende Adressmultiplexer auf die Speicher geschaltet. Wichtig ist, daß hier unbedingt Schottky Bausteine eingesetzt werden.

Precharge Schaltung

Um bei 6 MHz Systemfrequenz einen einwandfreien Betrieb mit den dynamischen Speichern zu gewährleisten ist eine Precharge-Extension-Schaltung erforderlich. Diese hat zur Aufgabe den Precharge beim M1 Zyklus zwischen Speicherzugriff (Opcode Fetch) und Refreshzugriff während T3 aufzuweiten. Wird auf den dynamischen Speicher zugegriffen, beendet IC 18 (74 LS 173) den Opcode Fetch mit der steigenden Flanke von T3 vorzeitig.

Festwertspeicher
Als Festwertspeicher kann wahlweise ein 2- bzw. 4-K-Byte EPROM eingesetzt werden. Der 24 polige Sockel ist dabei nach der JEDEC Norm ausgeführt. Pin 21 wird wahlweise mit +5V (2 K-Byte) oder A11 (4 K-Byte) beschaltet (am Pin 21 auf der Lötseite). Standardmäßig ist Pin 21 mit +5V verbunden.

Die Floppy Steuerung

Der Controller

Als Floppy Controller wurde der uPD 765 von NEC ausgewählt. Er ist in der Lage die unter CP/M 3.0 notwendigen Multi-Sector und Multi-Track Operationen zu unterstützen. Seine hohe Intelliegenz entlastet die Software erheblich. So ist er in der Lage selbstständig Disketten zu formatieren und Informationen über den Zustand von 4 Laufwerken zwischenzuspeichern. Das Einstellen von irgendwelchen Trimmern entfällt vollständig.

Während des Transfers von und zur Diskette wird der uPD 765 im Polling Verfahren betrieben. Der Systemtakt von 6 MHz erlaubt es 8 Zoll Double Density Disketten mit diesem Verfahren einwandfrei zu bearbeiten.

Der Interface Baustein

Den einwandfreien Datentransfer von und zur Diskette übernimmt der Baustein FDC 9229 (IC 23). Beim Lesen von Diskette sorgt eine integrierte "Phase Lock Loop" Schaltung für die Aufbereitung des seriellen Datenstroms. Beim Schreiben gewährleistet eine Write Precompensations Schaltung die sichere Aufzeichnungen. Der Precompensations Wert beträgt bei MINI Laufwerken 250 ns und bei Standard Drives 125 ns. Prinzipiell können über die Eingänge P0 - P2 auch andere Werte eingestellt werden.
Der Eingang MINI am 9229 erlaubt es zwischen 5.25 und 8 Zoll Laufwerken umzuschalten. Im wesentlichen wird dabei die Datentransferrate von 250 KBit/sec bei 5.25 Zoll auf 500 KBit bei 8 Zoll umgeschaltet. Der Zustand des MINI Eingangs läßt sich über eine Lötbrücke (unter dem Akku auf der Lötseite) bestimmen und über den Ausgang I5 des STI Bausteins von der Software schalten.

Signalbelegung des 34 poligen Floppysteckers:

!	01	02	Head Load
!	03	04	Head Load
!	05	06	Drive Select 3
!	07	08	Index
!	09	10	Drive Select 0
!	11	12	Drive Select 1
	13	14	Drive Select 2
G	15	16	Motor on
N	17	18	Direction
D	19	20	Step
	21	22	Write Data
!	23	24	Write Enable
!	25	26	Track 00
!	27	28	Write Protect
!	29	30	Read Data
!	31	32	Side Select
!	33	34	Ready

I/O Select

Die Dekodierung der I/O Bausteine übernimmt ein 256x4 PROM (IC 08) und ein 74 LS 139 TTL Baustein (IC 22). Durch Programmierung des PROM's lassen sich die I/O-Bausteine beliebig im maximalen I/O-Adreßbereich (max. 256) anordnen.

Signalbelegung des I/O PROMs IC 08 (24 SA 10):

A7	Pin 15	M1	D3	Pin 9	CE Cent,DART,173,FDC
A6	Pin 1	A1	D2	Pin 10	Wait Request
A5	Pin 2	A3	D1	Pin 11	CE STI
A4	Pin 3	A2	D0	Pin 12	I/O Select
A3	Pin 4	A7			
A2	Pin 7	A4			
A1	Pin 6	A5			
A0	Pin 5	A6			

In Abhängigkeit der acht Eingangssignale M1 und A1 - A7 werden vier Ausgangssignale erzeugt:
Das Signal IO-SEL führt Low Pegel, wenn auf einen internern I/O Baustein zugegriffen wird und dient zur BUS-Steuerung.
Ein Wait-Request-Signal das an die Wait-Schaltung geführt ist gestattet jedem der 256 I/O Adressen gezielt einen Wait-Stait zuzuordnen.
Die beiden verbleibenden Ausgänge dienen als Chip Enable Signale für die I/O Bausteine.
Ein CE wird direkt an den STI Baustein geführt der 16 I/O-Adressen belegt. Die oberen vier Adreßbits A7-A4 können frei durch Programmierung des PROMs bestimmt werden.
Die Z80-DART, der FDC, der Centronics Datenport und die Banking Schaltung belegen ebenfalls 16 I/O Adressen. Die oberen vier Adressen dieses Blocks können über den PROM festgelegt werden.
Die Dekodierung der Adressen A2 und A3 übernimmt IC 22 (74 LS 139). Innerhalb des 16 Byte Blocks haben die vier I/O-Baugruppen folgende Adreßlage:

XXXX 00xx	Centronics Datenport
XXXX 01xx	Z-80 DART
XXXX 10xx	Banking Port (LS 173)
XXXX 11xx	FDC 765

Die I/O Bausteine belegen standardmäßig folgende Adressen:

00-0F	Z-80 STI
10	Centronics Data-Output
14-17	Z-80 DART/SIO
18	Banking Output (LS 173)
1C,1D	FDC Controller uPD 765

Schnittstellen

a, Centronics

Die Centronics Schnittstelle erlaubt es auf einfache Weise einen Drucker anzuschließen. Der 8-Bit Datenstrom zum Printer wird über 8 D-Flip-Flops einem LS 374 Baustein (IC 34) erzeugt. Der Strobe Ausgang wird von dem STI-Baustein erzeugt und über ein Gatter (74 LS 367) gepuffert. Die Eingänge Paper empty und Acknowledge sind ebenfalls mit der STI verbunden. Busy ist an den DART-Baustein geführt. Alle Signale können somit Interrupts auslösen.
Sämtliche Signale sind auf einen 26 poligen Stecker nach Centronics Norm geführt.

Belegung des 26 poligen Centronics Steckers:

Strobe	01	02	!
D0	03	04	!
D1	05	06	!
D2	07	08	
D3	09	10	G
D4	11	12	N
D5	13	14	D
D6	15	16	
D7	17	18	!
Acknowledge	19	20	!
Busy	21	22	!
Paper Empty	23	24	
	25	26	

b, RS-232

Es stehen zwei serielle asynchrone vollduplex Kanäle zur Verfügung. Alternativ ist durch Austausch der Z80-DART durch eine Z80-SIO-0 auch synchrone Datenübertragung möglich. Die Baudrate ist softwaremäßig getrennt für beide Kanäle einstellbar. Channel A der DART wird von Timer D (STI) und Channel B von Timer C mit dem Baudrate Takt versorgt.
Receive- und Transmit-Data, Clear to Send und Request to Send sind bei beiden Kanälen auf RS-232 Pegel gepuffert und getrennt für jeden Kanal auf zwei 14 polige Pfostenstecker geführt. Die Belegung dieser Stecker entspricht der RS-232 Norm. Über ein Kabel in Schneid-Klemm-Technik lassen sie sich direkt mit einem 25 poligen D-Stecker verbinden.

Belegung der beiden 14 poligen RS-232 Stecker:

	1	01	02
Receive Data	2	03	04
Transmit Data	3	05	06
Request to Send	4	07	08
Clear to Send	5	09	10
	6	11	12
GND	7	13	14

c,seriell 5 Volt TTL

Eine weitere serielle asynchrone vollduplex Schittstelle mit TTL Pegel wird auf einem 10 poligen Stecker bereit gestellt. Selbige läßt sich z. B. zum Anschluß einer Tastatur verwenden. Die Baudrate ist auch hier softwaremäßig einstellbar (Timer B, STI).

Die Belegung ist wie folgt:

!	01	02	Vcc
G	03	04	NMI
N	05	06	GND
D	07	08	seriell in
!	09	10	seriell out

Echtzeituhr

Der Uhrenbaustein E 050-16 erlaubt die Angabe von Sekunden, Minuten, Stunden, Tag, Wochentag, Monat und (Schalt-)Jahr. Ein implementierter Akku ermöglicht den Betrieb auch bei ausgeschaltetem Gerät. Über den Trimmer C27 läßt sich die Quarzfrequenz exakt auf 32,768 kHz abgleichen. Die Steuersignale des Uhrenbausteins werden über die Z-80 DART und STI beschaltet. Der Datenaustausch geschieht dabei seriell über die Leitung DATA O/P. Ein Sekunden Takt ist an den Eingang I4 der STI geführt. I4 dient Timer A als Eingang im Event count mode. Entsprechend programmiert erzeugt der Ausgang TA des Timers das Motor on/off Signal für die Laufwerke, indem er den an I4 anliegenden Sekundenpuls herabzählt und bei Time-Out den Motor ausschaltet.

Beschaltung MEM E 050-16

OUT STOP:	Vcc	Inactiv	Input
RESET:	Vcc	Inactiv	Input
OUTSEL:	GND	Activ	Input
CHIP SELECT:	DTRA	Dart A Data Terminal Ready	Input
SECONDS:	STI I4	Timer A erzeugt Motor on/off	Output
DATA O/P:	STI I0	Serieller Datenstrom	In/Output
CLOCK:	DTRB	Dart B Data Terminal Ready	Input
MIN,HRS:	unbeschaltet		Outputs

Serial Timer Interrupt Controller STI

Der STI-Baustein stellt 8 parallel I/O-Kanäle, 1 USART und 4 Timer-Zeitgeber zur Verfügung.

Belegung der I/O Leitungen I0 - I7:

Echtzeituhr			
I0	Pin 8	I/O	serieller Datenstrom
I4	Pin 14	Input	Sekunden Puls für Timer A Input
Centronics			
I1	Pin 9	Output	Strobe, Low active
I3	Pin 11	Output	Paper empty, High active
I6	Pin 14	Input	Acknowledge, Low active
Floppy Controller			
I2	Pin 10	Output	Terminal Count 765, High active
I5	Pin 13	Output	MINI, High = MINI, Low = Std.
BOOT			
I7	Pin 15	Output	BOOT, High = EPROM ein, Low = EPROM aus
Timers			
TA	Pin 1	Output	Motor on/off FDD's
TB	Pin 2	Output	Baudrate for STI USART
TC	Pin 3	Output	Baudrate for DART Channel B
TD	Pin 4	Output	Baudrate for DART Channel A

Der STI Baustein ist derzeit nur in der A Version (4 MHz) erhältlich. Es ist deshalb bei Systemfrequenzen von 6 MHz nicht möglich die Vektor Interupt Fähigkeiten des Bausteins auszunützen.

BUS Schnittstelle

Große Bedeutung wurde dem BUS-Anschluß zugemessen. So sind Daten, Adressen und Steuersignale über 4 Treiber Bausteine (74 LS 245) bidirektional gepuffert. Das heißt es können externe Baugruppen wie eine CPU oder eine DMA auf den interner Speicher bzw. I/O Baugruppen zugreifen. Der DART und STI Baustein sind in der Lage einen Vektor-Interrupt an eine externe CPU weiterzuleiten.
Die Steuerung des Daten- und Adreß-bus wird von einem 256x4 PROM (IC 06) durchgeführt. In Abhängigkeit der acht Eingangssignale werden vier Ausgangssignale erzeugt, welche die Richtung und den Tri-State Zustand der Daten- und Adreß-treiber bestimmen.
Die Signalbelegung des BUS-Steuerproms IC 06 ist wie folgt:

A7	Pin 15	BUSAK	D3	Pin 9	EN 173
A6	Pin 1	RD	D2	Pin 10	DIR DATA
A5	Pin 2	IEI	D1	Pin 11	EN BUS
A4	Pin 3	IO-SEL	D0	Pin 12	DIR Adress
A3	Pin 4	M1			
A2	Pin 7	IEO			
A1	Pin 6	Memory-SEL			
A0	Pin 5	IORQ			

Die vier Treiber Bausteine sind grundsätzlich 'enabled'. Wird die Karte als Slave eingesetzt besteht die Möglichkeit die Treiber auch in den Tri-State Zustand zu versetzen.
Das Signal BUSAK bestimmt die Richtung der Adress und Steuerleitungen. Ist BUSAK inaktiv (High) sind die Treiber nach außen auf den BUS geschaltet. Die oberen vier Adressen A16-A19 werden getrennt angesprochen. Diese können bei aktiven BUSAK wahlweise in den Tri-State Zustand versetzt werden oder nicht (standardmäßig Tri-State).
Um die Richtung des Datenstromes festzulegen sind eine Reihe von Informationen notwendig. Hierbei muß unterschieden werden, ob die interne CPU aktiv ist oder nicht, ob auf einen internen oder externen Speicher oder I/O Baugruppe zugegriffen wird und ob ein Vektor Interrupt vorliegt.
Grundsätzlich ist der Datenstrom nach außen geschaltet. Außnahmen sind Lesebefehle, die auf externe Baugruppen zugreifen und Interrupt Acknowlege Zyklen die von externen I/O Baugruppen ausgelöst wurden.
Ist die interne CPU inaktiv geht der Datenstrom zum Karteninneren. Außnahmen sind Lesebefehle von einer externen Einheit auf Baugruppen im Karteninneren.

Einsatz als I/O Karte

Sämtliche auf der CPU Karte befindlichen Baugruppen können auch von einer externen CPU angesprochen werden. D. h. die EPC-Baugruppe kann auch in einem bestehenden System als I/O Karte eingesetzt werden.
Zum Einsatz als I/O-Karte sind folgende Punkte zu beachten:

1, Der Systemtakt ist extern zuzuführen. D.h. der interne Taktgenerator muß durch Entfernen des Widerstandes R05 deaktiviert werden.
2, Der BUSAK Ausgang ist vom BUS abzutrennen.
3, Die CPU muß aus der Fassung genommen und durch eine Brücke von GND (Pin 29) nach BUSAK (Pin 23) ersetzt werden.
4, Es ist zu überprüfen ob das PWRCLR Signal auf dem Systembus als open Collector ausgeführt wurde. Wenn nicht, ist es vom BUS abzutrennen.

Einsatz als Slave Prozessor

Ebenfalls ist die CPU Karte als Slave in einem Multiprozessor-Multi-User System einsetzbar. Die Karte arbeitet dabei parallel zu anderen Prozessoren in dem System ohne die Aktivitäten des Masters auf dem Systembus zu stören. Der Datenaustausch zwischen Slave und Master kann dabei mit der vollen Geschwindigkeit des Systembuses erfolgen. Der Datenaustausch erfolgt über den RAM auf der EPC Karte.
Folgende Änderungen sind vorzunehmen:

1, Pin 2 des Memory PROMs wird mit BUSAK beschaltet
2, Ein spezieller Memory PROM ist eizusetzen
3, Ein spezieller BUS PROM ist einzusetzen
4, Die Leitungen IEI, IEO, CLK , DESLCT, WAIT, INT, NMI sind nicht auf des MASTER Bus zu schalten.
5, Die PWRCLR Leitung muß als open Collector ausgeführt sein
6, BUSREQ und BUSAK sind nicht auf den BUS zu legen, sondern über zwei parallele I/O Leitungen mit dem Master zu verbinden.
Die BUSREQ Leitung dient dann als Slave Request Leitung,
die BUSAK Leitung als Slave Acknowledge Signal.
Des weiteren kann eine I/O Leitung auf der EPC-Karte mit einer I/O Leitung des Masters verbunden werden die als Master Request Signal dient.

Bei mehreren Slave Baugruppen in einem System empfielt es sich eine spezielle Rückwandverdrahtung anzufertigen.

Funktionsweise als Slave Baugruppe

Nach einem System Reset sind sämtliche Leitungstreiber zum BUS nach beiden Richtungen im Tri-State-Zustand. Sowohl Master als auch Slave beginnen simultan und autonom zu arbeiten. Nun kann z.B. der Slave über die Master-Request Leitung die einen Interrupt im Master System auslöst Bearbeitung beantragen. Der Master muß das Slave-Reqest-Signal, das an BUSREQ der Slave CPU herangeführt ist, aktivieren. Die Slave CPU antwortet mit BUSAK (Slave Acknowledge), das gleichzeitig an den Memory PROM herangeführt wird. Dies hat zur Folge, daß der auf der Slave Karte befindliche Speicher in eine Adreßlage gebracht werden kann, die sich nicht mit dem Speicher des Masters überlagert. Ebenfalls werden sämtliche Treiberbausteine aktiviert. Der Master kann nun ungehindert auf den Speicher der Slave Karte zugreifen.
Nach Zurücknehmen des Slave-Request Signals und des daraus resultierenden inaktiv werden des Slave Acknowledge Signals können sowohl Master als auch Slave wieder autonom arbeiten.

Inbetriebnahme

Terminalanschluß

Wahlweise kann an den EPC unsere RGB Grafikkarte oder über RS-232 ein handelsübliches Terminal angeschlossen werden. Die Software überprüft beim Einschalten ob die RGB Baugruppe am BUS vorhanden ist. Falls nicht erfolgt die Ausgabe über RS-232 Channel A.
Das Übertragungsprotokoll ist standardmäßig folgendes:
Baudrate 9600, 1 Stoppbit, keine Parität

Tastaturanschluß

Wird die RGB Grafikkarte als Ausgabeeinheit verwendet, erwartet die Standardsoftware an der seriellen TTL Schnittstelle auf dem EPC die Tastatureingabe.
Das Übertragungsprotokoll ist standardmäßig folgendes:
Baudrate 9600, 1 Stoppbit, keine Parität

Laufwerksanschluß

Head Load

Der EPC stellt ein Head-Load Signal auf Pin 2 und 4 der Floppy-Steckerleiste zur Verfügung. Dieses Signal kann zum Laden der Köpfe herangezogen werden. Werden TEAC- Laufwerke eingesetzt, so ist die Steckbrücke **HL** und **SM** zu setzen.

Alternativ kann der Kopf der Laufwerke auch durch das Motor-On-Signal des EPC-Computers geladen werden. Dadurch werden die Kopf-Lade-Geräusche stark herabgesetzt, der Kopf bleibt jedoch länger auf der Diskette abgesenkt (trotzdem empfehlenswert !). Bei den TEAC-Laufwerken sind hierzu die Steckbrücken **HM** und **SM** zu setzen.

Drive-Select

Je nach gewünschter Laufwerksnummer sind die entsprechen den Jumper auf den Laufwerken zu setzen (DS0 - DS3). Laufwerk A (DS0) muß auf jeden Fall vorhanden sein. Dabei ist zu beachten, daß der Floppy-Controller uPD 765 beim 'Steppen' die Drive-Select-Leitung des gewünschten Lauf werks nur für die Zeit des 'Step'-Pulses aktiviert, da nach werden nacheinander auch die restlichen drei Drive- Select-Leitungen angesprochen, um den 'Ready-Zustand' dieser Laufwerke abzufragen. Dies kann vereinzelt zu Problemen bei Einsatz älterer 8 Zoll Laufwerke führen.

Pin Description

B/$\overline{A}$. *Channel A Or B Select* (input, High selects Channel B). This input defines which channel is accessed during a data transfer between the CPU and the Z-80 DART.

C/$\overline{D}$. *Control Or Data Select* (input, High selects Control). This input specifies the type of information (control or data) transferred on the data bus between the CPU and the Z-80 DART.

$\overline{CE}$. *Chip Enable* (input, active Low). A Low at this input enables the Z-80 DART to accept command or data input from the CPU during a write cycle, or to transmit data to the CPU during a read cycle.

CLK. *System Clock* (input). The Z-80 DART uses the standard Z-80 single-phase system clock to synchronize internal signals.

$\overline{CTSA}$, $\overline{CTSB}$. *Clear To Send* (inputs, active Low). When programmed as Auto Enables, a Low on these inputs enables the respective transmitter. If not programmed as Auto Enables, these inputs may be programmed as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow-risetime signals.

D_0-D_7. *System Data Bus* (bidirectional, 3-state) transfers data and commands between the CPU and the Z-80 DART.

$\overline{DCDA}$, $\overline{DCDB}$. *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if the Z-80 DART is programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered.

$\overline{DTRA}$, $\overline{DTRB}$. *Data Terminal Ready* (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be programmed as general-purpose outputs.

IEI. *Interrupt Enable In* (input, active High) is used with IEO to form a priority daisy chain when there is more than one interrupt-driven device. A High on this line indicates that no other device of higher priority is being serviced by a CPU interrupt service routine.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an interrupt from this Z-80 DART. Thus, this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

$\overline{INT}$. *Interrupt Request* (output, open drain, active Low). When the Z-80 DART is requesting an interrupt, it pulls $\overline{INT}$ Low.

$\overline{M1}$. *Machine Cycle One* (input from Z-80 CPU, active Low). When $\overline{M1}$ and $\overline{RD}$ are both active, the Z-80 CPU is fetching an instruction from memory; when $\overline{M1}$ is active while $\overline{IORQ}$ is active, the Z-80 DART accepts $\overline{M1}$ and $\overline{IORQ}$ as an interrupt acknowledge if the Z-80 DART is the highest priority device that has interrupted the Z-80 CPU.

$\overline{IORQ}$. *Input/Output Request* (input from CPU, active Low). $\overline{IORQ}$ is used in conjunction with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$ and $\overline{RD}$ to transfer commands and data between the CPU and the Z-80 DART. When $\overline{CE}$, $\overline{RD}$ and $\overline{IORQ}$ are all active, the channel selected by B/$\overline{A}$ transfers data to the CPU (a read operation). When $\overline{CE}$ and $\overline{IORQ}$ are active, but $\overline{RD}$ is inactive, the channel selected by B/$\overline{A}$ is written to by the CPU with either data or control information as specified by C/$\overline{D}$.

$\overline{RxCA}$, $\overline{RxCB}$. *Receiver Clocks* (inputs). Receive data is sampled on the rising edge of $\overline{RxC}$. The Receive Clocks may be 1, 16, 32 or 64 times the data rate.

$\overline{RD}$. *Read Cycle Status.* (input from CPU, active Low). If $\overline{RD}$ is active, a memory or I/O read operation is in progress.

RxDA, RxDB. *Receive Data* (inputs, active High).

$\overline{RESET}$. *Reset* (input, active Low). Disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem controls High and disables all interrupts.

$\overline{RIA}$, $\overline{RIB}$. *Ring Indicator* (inputs, Active Low). These inputs are similar to $\overline{CTS}$ and $\overline{DCD}$. The Z-80 DART detects both logic level transitions and interrupts the CPU. When not used in switched-line applications, these inputs can be used as general-purpose inputs.

$\overline{RTSA}$, $\overline{RTSB}$. *Request to Send* (outputs, active Low). When the RTS bit is set, the $\overline{RTS}$ output goes Low. When the RTS bit is reset, the output goes High after the transmitter empties.

$\overline{TxCA}$, $\overline{TxCB}$. *Transmitter Clocks* (inputs). TxD changes on the falling edge of $\overline{TxC}$. The Transmitter Clocks may be 1, 16, 32 or 64 times the data rate; however, the clock multiplier for the transmitter and the receiver must be the same. The Transmit Clock inputs are Schmitt-trigger buffered. Both the Receiver and Transmitter Clocks may be driven by the Z-80 CTC Counter Time Circuit for programmable baud rate generation.

TxDA, TxDB. *Transmit Data* (outputs, active High).

$\overline{W/RDYA}$, $\overline{W/RDYB}$. *Wait/Ready* (outputs, open drain when programmed for Wait function, driven High and Low when programmed for Ready function). These dual-purpose outputs may be programmed as Ready lines for a DMA controller or as Wait lines that synchronize the CPU to the Z-80 DART data rate. The reset state is open drain.

Functional Description

The functional capabilities of the Z-80 DART can be described from two different points of view: as a data communications device, it transmits and receives serial data, and meets the requirements of asynchronous data communications protocols; as a Z-80 family peripheral, it interacts with the Z-80 CPU and other Z-80 peripheral circuits, and shares the data, address and control buses, as well as being a part of the Z-80 interrupt structure. As a peripheral to other microprocessors, the Z-80 DART offers valuable features such as non-vectored interrupts, polling and simple handshake capability.

The first part of the following functional description introduces Z-80 DART data communications capabilities; the second part describes the interaction between the CPU and the Z-80 DART.

The Z-80 DART offers RS-232 serial communications support by providing device signals for external modem control. In addition to dual-channel Request To Send, Clear To Send, and Data Carrier Detect ports, the Z-80 DART also features a dual channel Ring Indicator (RIA, RIB) input to facilitate local/remote or station-to-station communication capability.

Communications Capabilities. The Z-80 DART provides two independent full-duplex channels for use as an asynchronous receiver/transmitter. The following is a short description of receiver/transmitter capabilities. For more details, refer to the Asynchronous Mode section of the *Z-80 SIO Technical Manual*. The Z-80 DART offers transmission and reception of five to eight bits per character, plus optional even or odd parity. The transmitter can supply one, one and a half or two stop bits per character and can provide a break output at any time. The receiver break detection logic interrupts the CPU both at the start and end of a received break. Reception is protected from spikes by a transient spike rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the Receive Data input. If the Low does not persist—as in the case of a transient—the character assembly process is not started.

Framing errors and overrun errors are detected and buffered together with the character on which they occurred. Vectored interrupts allow fast servicing of interrupting conditions using dedicated routines. Furthermore, a built-in checking process avoids interpreting a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit is begun.

The Z-80 DART does not require symmetric Transmit and Receive Clock signals—a feature that allows it to be used with a Z-80 CTC or any other clock source. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32 or 1/64 of the clock rate supplied to the Receive and Transmit Clock inputs. When using Channel B, the bit rates for transmit and receive operations must be the same because $\overline{RxC}$ and $\overline{TxC}$ are bonded together ($\overline{RxTxCB}$).

I/O Interface Capabilities. The Z-80 DART offers the choice of Polling, Interrupt (vectored or non-vectored) and Block Transfer modes to transfer data, status and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

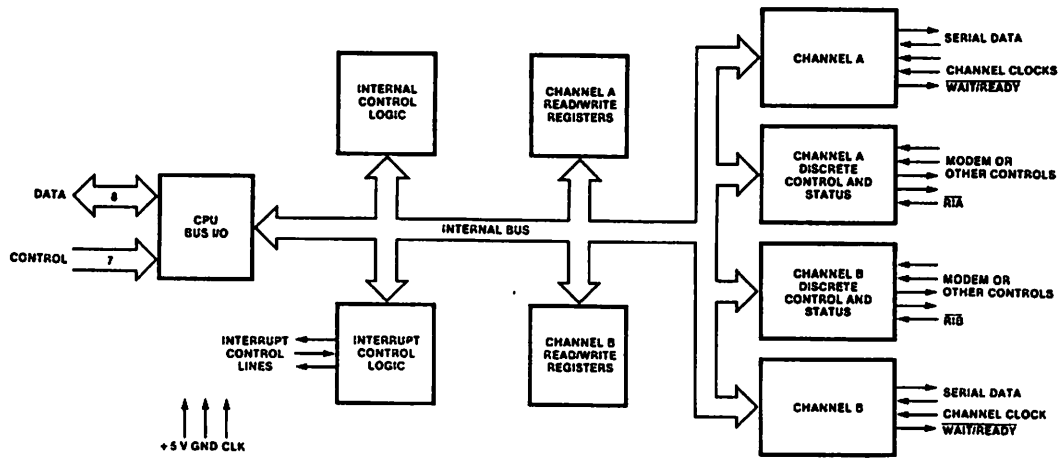


Figure 3. Block Diagram

Functional Description (Continued)

POLLING. There are no interrupts in the Polled mode. Status registers RR0 and RR1 are updated at appropriate times for each function being performed. All the interrupt modes of the Z-80 DART must be disabled to operate the device in a polled environment.

While in its Polling sequence, the CPU examines the status contained in RR0 for each channel; the RR0 status bits serve as an acknowledge to the Poll inquiry. The two RR0 status bits D_0 and D_2 indicate that a data transfer is needed. The status also indicates Error or other special status conditions (see "Z-80 DART Programming"). The Special Receive Condition status contained in RR1 does not have to be read in a Polling sequence because the status bits in RR1 are accompanied by a Receive Character Available status in RR0.

INTERRUPTS. The Z-80 DART offers an elaborate interrupt scheme that provides fast interrupt response in real-time applications. As a member of the Z-80 family, the Z-80 DART can be daisy-chained along with other Z-80 peripherals for peripheral interrupt-priority resolution. In addition, the internal interrupts of the Z-80 DART are nested to prioritize the various interrupts generated by Channels A and B. Channel B registers WR2 and RR2 contain the interrupt vector that points to an interrupt service routine in the memory. To eliminate the necessity of writing a status analysis routine, the Z-80 DART can modify the interrupt vector in RR2 so it points directly to one of eight interrupt service routines. This is done under program control by setting a program bit (WR1, D_2) in Channel B called "Status Affects Vector." When this bit is set, the interrupt vector in RR2 is modified according to the assigned priority of the various interrupting conditions.

Transmit interrupts, Receive interrupts and External/Status interrupts are the main sources of interrupts. Each interrupt source is enabled under program control with Channel A having a higher priority than Channel B, and with Receiver, Transmit and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted by the transmit buffer *becoming* empty. (This implies that the transmitter must have had a data character written into it so it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on the first received character
- Interrupt on all received characters
- Interrupt on a Special Receive condition

Interrupt On First Character is typically used with the Block Transfer mode. Interrupt On All Receive Characters can optionally modify the interrupt vector in the event of a parity error. The Special Receive Condition interrupt can occur on a character basis. The Special Receive condition can cause an interrupt only if the Interrupt On First Receive Character or Interrupt On All Receive Characters mode is selected. In Interrupt On First Receive Character, an interrupt can occur from Special Receive conditions (except Parity Error) after the first receive character interrupt (example: Receive Overrun interrupt).

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$ and $\overline{RI}$ pins; however, an External/Status interrupt is also caused by the detection of a Break sequence in the data stream. The interrupt caused by the Break sequence has a special feature that allows the Z-80 DART to interrupt when the Break sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Break condition.

CPU/DMA BLOCK TRANSFER. The Z-80 DART provides a Block Transfer mode to accommodate CPU block transfer functions and DMA block transfers (Z-80 DMA or other designs). The Block Transfer mode uses the $\overline{W/RDY}$ output in conjunction with the Wait/Ready bits of Write Register 1. The $\overline{W/RDY}$ output can be defined under software control as a Wait line in the CPU Block Transfer mode or as a Ready line in the DMA Block Transfer mode.

To a DMA controller, the Z-80 DART Ready output indicates that the Z-80 DART is ready to transfer data to or from memory. To the CPU, the Wait output indicates that the Z-80 DART is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle.

Internal Architecture

The device internal structure includes a Z-80 CPU interface, internal control and interrupt logic, and two full-duplex channels. Each channel contains read and write registers, and discrete control and status logic that provides the interface to modems or other external devices.

The read and write register group includes five 8-bit control registers and two status registers. The interrupt vector is written into an additional 8-bit register (Write Register 2) in Channel B that may be read through Read Register 2 in Channel B. The registers for both channels are designated as follows:

WR0-WR5 — Write Registers 0 through 5
RR0-RR2 — Read Registers 0 through 2

The bit assignment and functional grouping of each register is configured to simplify and organize the programming process.

The logic for both channels provides formats, bit synchronization and validation for data transferred to and from the channel interface. The modem control inputs Clear to Send ($\overline{CTS}$), Data Carrier Detect ($\overline{DCD}$) and Ring Indicator ($\overline{RI}$) are monitored by the control logic under program control. All the modem control signals are general purpose in nature and can be used for functions other than modem control.

For automatic interrupt vectoring, the interrupt control logic determines which channel and which device within the channel has the highest priority. Priority is fixed with Channel A assigned a higher priority than Channel B; Receive, Transmit and External/Status interrupts are prioritized in that order within each channel.

Data Path. The transmit and receive data path illustrated for Channel A in Figure 4 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service a Receive Character Available interrupt in a high-speed data transfer.

The transmitter has an 8-bit transmit data register that is loaded from the internal data bus, and a 9-bit transmit shift register that is loaded from the transmit data register.

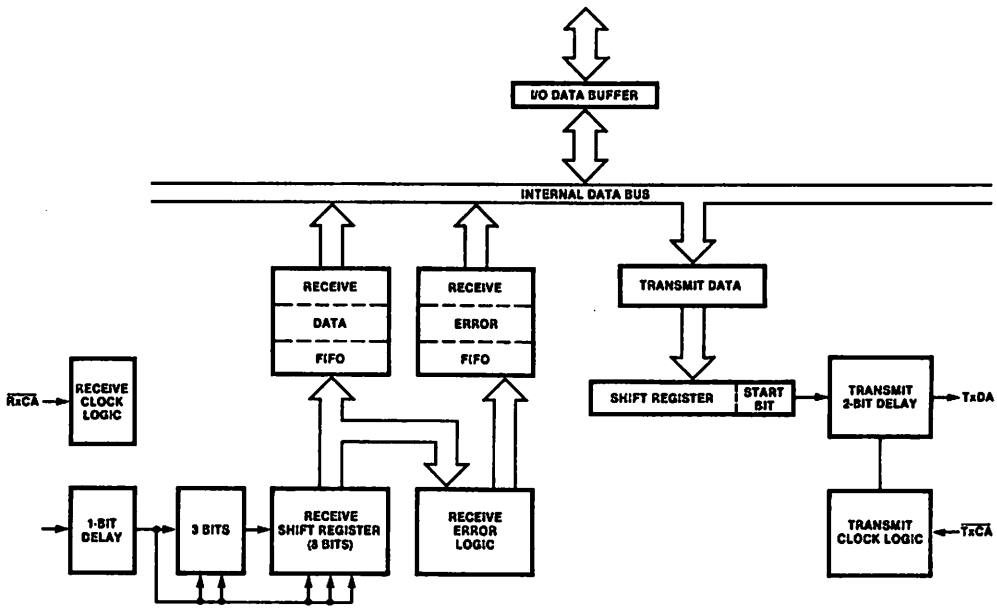


Figure 4. Data Path

Z-80 DART Programming

To program the Z-80 DART, the system program first issues a series of commands that initialize the basic mode and then other commands that qualify conditions within the selected mode. For example, the character length, clock rate, number of stop bits, even or odd parity are first set, then the Interrupt mode and, finally, receiver or transmitter enable.

Both channels contain command registers that must be programmed via the system program prior to operation. The Channel Select input (B/$\overline{A}$) and the Control/Data input (C/$\overline{D}$) are the command structure addressing controls, and are normally controlled by the CPU address bus.

Write Registers. The Z-80 DART contains six registers (WR0-WR5) in each channel that are programmed separately by the system program to configure the functional personality of the channels (Figure 4). With the exception of WR0, programming the write registers requires two bytes. The first byte contains three bits (D_0-D_2) that point to the selected register; the second byte is the actual control word that is written into the register to configure the Z-80 DART.

WR0 is a special case in that all the basic commands (CMD_0-CMD_2) can be accessed with a single byte. Reset (internal or external) initializes the pointer bits D_0-D_2 to point to WR0. This means that a register cannot be pointed to in the same operation as a channel reset.

Write Register Functions	
WR0	Register pointers, initialization commands for the various modes, etc.
WR1	Transmit/Receive interrupt and data transfer mode definition.
WR2	Interrupt vector (Channel B only)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls

Read Registers. The Z-80 DART contains three registers (RR0-RR2) that can be read to obtain the status information for each channel (except for RR2, which applies to Channel B only). The status information includes error conditions, interrupt vector and standard communications-interface signals.

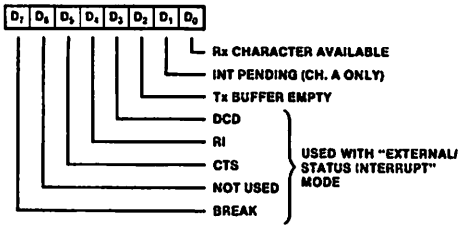
To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing an input instruction, the contents of the addressed read register can be read by the CPU.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring. For example, when the interrupt vector indicates that a Special Receive Condition interrupt has occurred, all the appropriate error bits can be read from a single register (RR1).

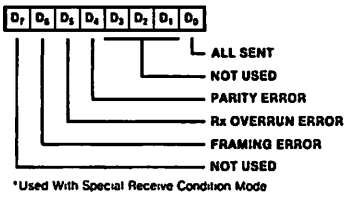
Read Register Functions	
RR0	Transmit/Receive buffer status, interrupt status and external status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only)

Z-80 DART Read and Write Registers

READ REGISTER 0

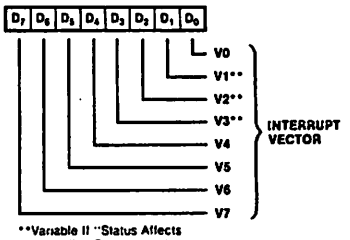


READ REGISTER 1*



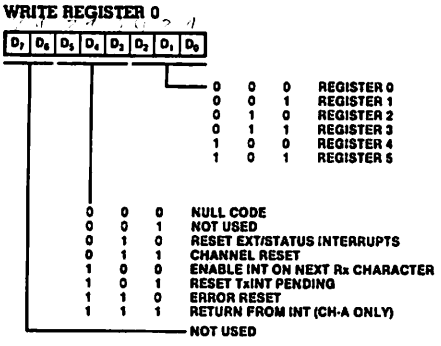
*Used With Special Receive Condition Mode

READ REGISTER 2

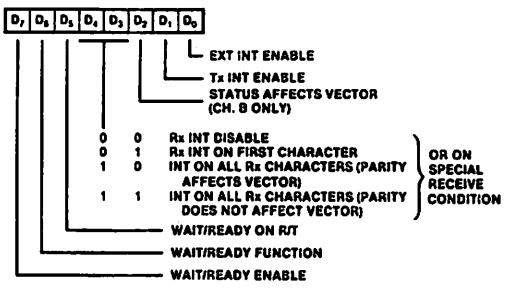


**Variable if "Status Affects Vector" Is Programmed

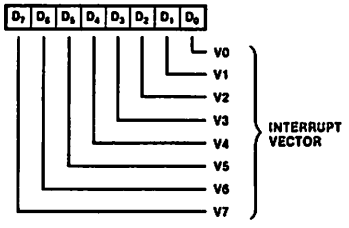
WRITE REGISTER 0



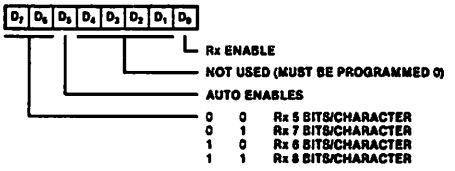
WRITE REGISTER 1



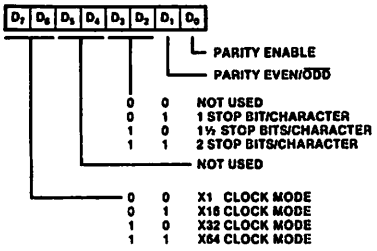
WRITE REGISTER 2 (CHANNEL B ONLY)



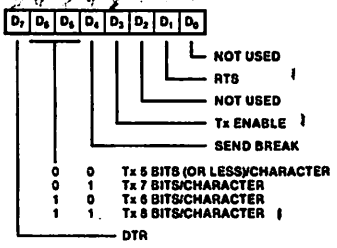
WRITE REGISTER 3



WRITE REGISTER 4



WRITE REGISTER 5



SIGNAL NAME	DESCRIPTION
V_{SS}	Ground
V_{CC}	+5 volts (± 5 percent)
$\overline{CE}$	Chip Enable (Input, active low)
$\overline{RD}$	Read Enable (Input, active low)
$\overline{WR}$	Write Enable (Input, active low)
A_0-A_3	Address Inputs. Used to address one of the internal registers during a read or write operation
D_0-D_7	Data Bus (bi-directional)
$\overline{RESET}$	Device Reset (negative true). When activated, all internal registers (except for Timer or USART Data registers) will be cleared, all timers stopped, USART turned off, all interrupts disabled and all pending interrupts cleared, and all I/O lines placed in tri-state input mode.
I_0-I_7	General purpose I/O and interrupt lines
$\overline{INT}$	Interrupt Request (Output, active low, open drain)
$\overline{IORQ}$	Input/Output Request from Z80-CPU (input, active low). The $\overline{IORQ}$ signal is used in conjunction with $\overline{M1}$ to signal the MK3801 that the CPU is acknowledging its interrupt.
IEI	Interrupt Enable In, active High
IEO	Interrupt Enable Out, active High
SO	Serial Output
SI	Serial Input
RC	Receiver Clock Input
TC	Transmit Clock Input
TAO-TDO	Timer Outputs
TCLK	Timer Clock Input
$\overline{M1}$	Z80 Machine Cycle One (negative true)

PIN DESCRIPTION

Figure 1 illustrates the pinout of the MK3801. The functions of these individual pins are described above.

INTERNAL ORGANIZATION

Figure 2 illustrates the MK3801 internal organization, which supports the full set of timing, communications, parallel I/O, and interrupt processing functions available in the device.

CPU BUS I/O

The CPU BUS I/O provides the means of communications between the system and the MK3801. Data, Status, and Control Registers in the MK3801 are accessed by the bus in order to establish device parameters, assert control, and transfer status and data between the system and the MK3801.

Each register in the MK3801 is addressed over the address bus in conjunction with Chip Enable ($\overline{CE}$), while data is transferred over the eight bit Data bus under control of Read ($\overline{RD}$) and Write ($\overline{WR}$) signals.

REGISTER ACCESSES

All register accesses are independent of any system clock. To read a register, both $\overline{CE}$ and $\overline{RD}$ must be active. The internal read control signal is essentially the combination of both $\overline{CE}$ and $\overline{RD}$ active; thus the read operation will begin when the later of these two signals goes active and will end when the first signal goes inactive. The address bus must be stable prior to the start of the operation and must remain stable until the end of the operation. Unless a read operation or an interrupt acknowledge cycle is in progress, the data bus (D_0-D_7) will remain in the tri-state condition.

To write a register, both $\overline{CE}$ and $\overline{WR}$ must be active. The address must be stable prior to the start of the operation and must remain stable until the end of the operation. The data must be stable prior to the end of the operation and must remain stable until the end of the operation. The data presented on the bus will be latched into the register shortly after either $\overline{WR}$ or $\overline{CE}$ goes inactive.

INTERNAL REGISTERS

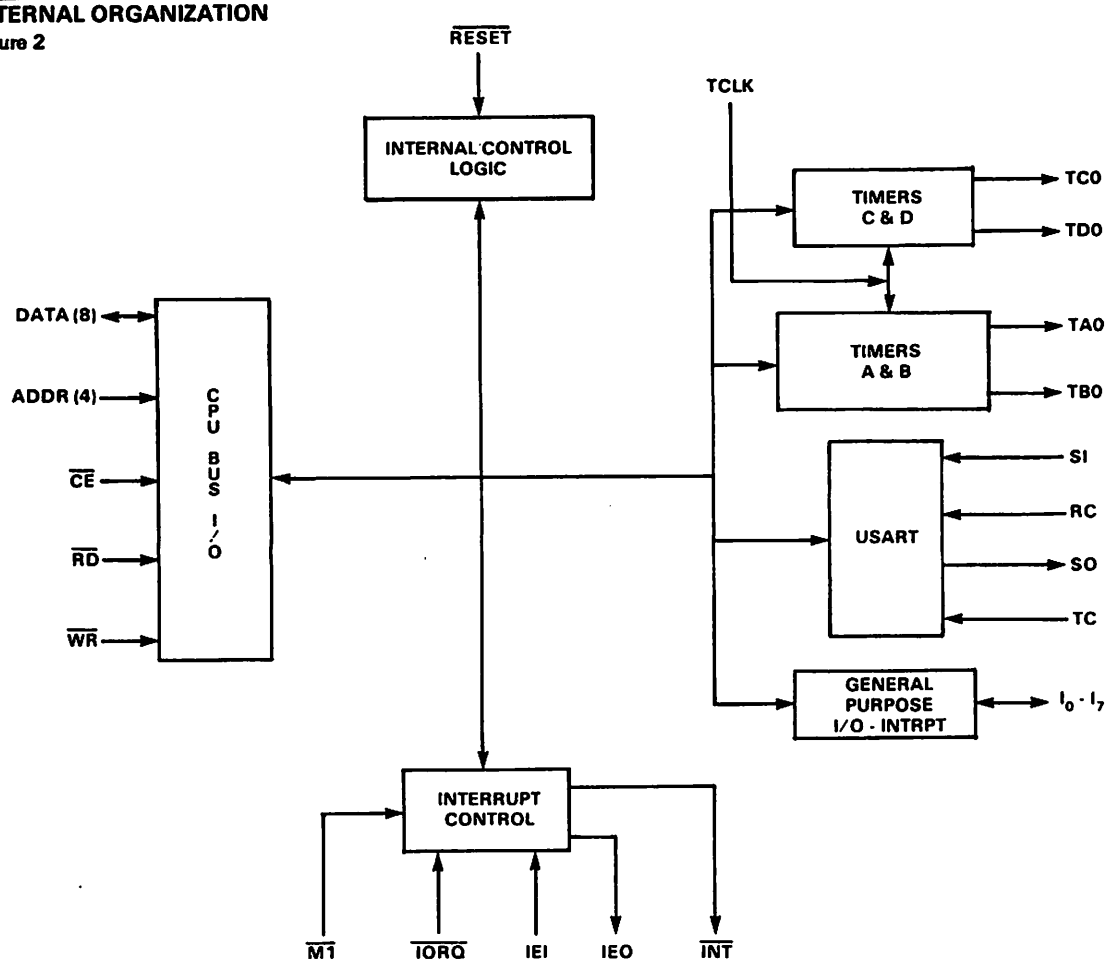
There are 24 internal registers used to control the operation of the STI. Sixteen of these registers are directly addressable and accessible. Eight registers are indirectly addressable via the Pointer/Vector Register and accessible via the Indirect Data Register.

DIRECTLY ADDRESSABLE REGISTERS

The Directly Addressable Registers are accessed by placing the address of the desired register on the address lines (A_0-A_3) during a write or read cycle. Figure 3 lists the Directly Addressable Registers.

INTERNAL ORGANIZATION

Figure 2



DIRECTLY ACCESSIBLE REGISTERS

Figure 3

ADDRESS	ABBREVIATION	REGISTER NAME
0	IDR	Indirect Data Register
1	GPIP	General Purpose I/O-Interrupt
2	IPRB	Interrupt Pending Register B
3	IPRA	Interrupt Pending Register A
4	ISRB	Interrupt in-Service Register B
5	ISRA	Interrupt in-Service Register A
6	IMRB	Interrupt Mask Register B
7	IMRA	Interrupt Mask Register A
8	PVR	Pointer/Vector Register
9	TABCR	Timers A and B Control Register

DIRECTLY ACCESSIBLE REGISTERS (Continued)
Figure 3

ADDRESS	ABBREVIATION	REGISTER NAME
A	TBDR	Timer B Data Register
B	TADR	Timer A Data Register
C	UCR	USART Control Register
D	RSR	Receiver Status Register
E	TSR	Transmitter Status Register
F	UDR	USART Data Register

INDIRECTLY ADDRESSABLE REGISTERS
Figure 4

INDIRECT ADDRESS	ABBREVIATION	REGISTER NAME
0	SCR	Sync Character Register
1	TDDR	Timer D Data Register
2	TCDR	Timer C Data Register
3	AER	Active Edge Register
4	IERB	Interrupt Enable Register B
5	IERA	Interrupt Enable Register A
6	DDR	Data Direction Register
7	TCDCR	Timers C and D Control Register

INDIRECTLY ADDRESSABLE REGISTERS

Indirectly Addressable Registers are addressed by placing the indirect address in bits IA0-IA2 of the Pointer/Vector Register, as defined in Figure 5. Data may be written to or read from the register indicated by these Indirect Register Address bits by a write or read access of the Indirect Data Register (selected when A_0-A_3 are all zero). The indirect address bits of the Pointer/Vector Register will remain unchanged after an indirect access. Repeated accesses of the Indirect Data Register will access the same indirect register as long as the indirect address in the Pointer/Vector Register remains unchanged. The Indirectly Addressable Registers are listed in Figure 4.

INTERRUPT VECTOR DEFINITION

Each individual function in the MK3801 is provided with a unique interrupt vector that is presented to the system during the interrupt acknowledge cycle. The interrupt vector returned during interrupt acknowledge is formed as shown in Figure 6. There are 16 vector addresses generated internally by the MK3801, one for each of the 16 interrupt channels. The three most significant bits of these vector addresses correspond to the three most significant bits of the Pointer/Vector Register shown in Figure 5. The least significant bit of each vector address is always 0, thus producing even vector addresses. The remaining 4 bits (IV_1 through IV_4) identify each of the 16 interrupt channels individually. The lowest priority channel responds with 0000 for IV_4-IV_1 respectively. The next higher priority channel responds with 0001, and so on in binary order, with the highest priority channel responding with 1111. Figure 7 lists each of the 16 interrupt channels in order of descending priority.

INTERRUPT CONTROL REGISTERS

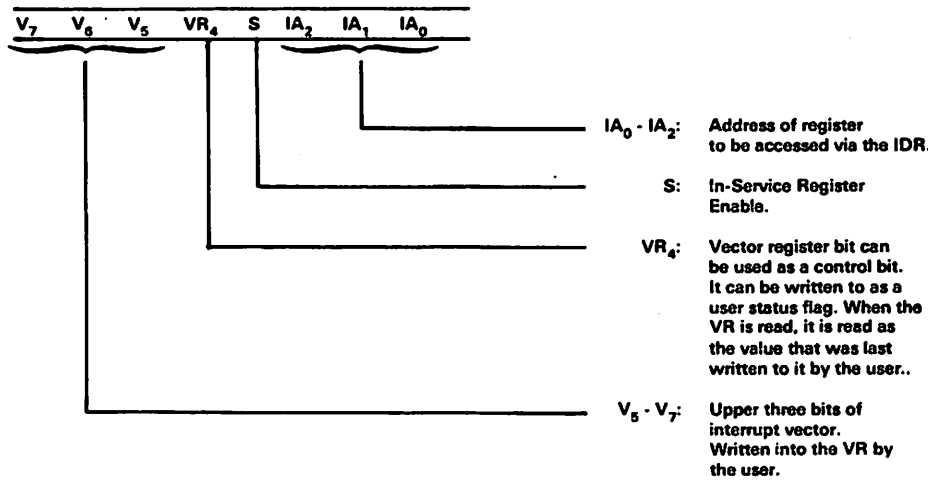
The Interrupt Control Registers provide control of interrupt processing for all I/O facilities of the MK3801. These registers allow the programmer to enable or disable any or all of the 16 interrupts, provide masking for any interrupts, and access to the pending or in-service status of the interrupts. Optional End-of-Interrupt modes are available under software control. The format of each of the Interrupt Control Registers is presented in Figure 8.

INTERRUPT OPERATION

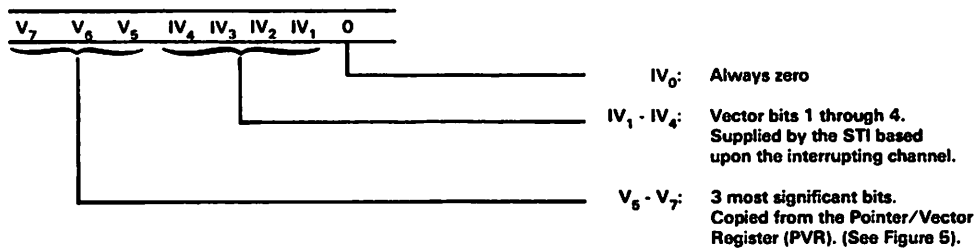
The Interrupt Enable Registers enable or disable the setting of an interrupt in the Interrupt Pending Registers. A 0 in a bit of the Interrupt Enable Registers disables the interrupt for the associated channel while a 1 enables the interrupt.

Once an interrupt is enabled, the occurrence of an interrupting condition on that channel will cause the

POINTER/VECTOR REGISTER (PVR) Port 08
Figure 5



INTERRUPT VECTOR
Figure 6



corresponding bit in the Interrupt Pending Register to be set. This indicates that an interrupt is pending in the MK3801.

Pending interrupts are presented to the Z80 CPU in order of priority (see Figure 1) unless they have been masked off. This is done by clearing the bit in the Interrupt Mask Register corresponding to the channel whose interrupt is to be masked. The channel's interrupt will remain pending until the mask bit for that channel is set, at which time the interrupt for that channel will be processsed in order of priority.

When an interrupt vector is generated for a pending interrupt and passed to the Z80 CPU, the bit in the Interrupt Pending Register, associated with the channel generating the interrupt, will be cleared. At this time, no history of the interrupt remains in the MK3801.

In order to retain historical evidence of an interrupt being serviced by the Z80, the In-Service Register may be enabled by setting the S-bit in the Pointer/Vector Register (see Figure 5). If the In-Service Register is enabled, the bit of the In-Service Register corresponding to the interrupting channel will be set when the interrupt vector is passed to the Z80. At the same time, the Interrupt Pending bit will be cleared since the interrupt is now in service. The In-Service bit will be cleared on execution of a Return-from-Interrupt (H'ED4D') instruction. The In-Service Registers are directly addressable, and the In-Service bit for any interrupt may be cleared by writing to the In-Service Register if the Return-from-Interrupt instruction is not used.

INTERRUPT CONTROL REGISTER DEFINITIONS

Figure 7

There are sixteen interrupt channels on the STI arranged in the following priority:

PRIORITY	CHANNEL	DESCRIPTION	ALTERNATE USAGE
HIGHEST	1111	General Purpose Interrupt 7 (I_7)	
	1110	General Purpose Interrupt 6 (I_6)	
	1101	Timer A	
	1100	Receive Buffer Full	
	1011	Receive Error	
	1010	Transmit Buffer Empty	
	1001	Transmit Error	
	1000	Timer B	
	0111	General Purpose Interrupt 5 (I_5)	
	0110	General Purpose Interrupt 4 (I_4)	TA (PW-Event)
	0101	Timer C	
	0100	Timer D	
	0011	General Purpose Interrupt 3 (I_3)	TB (PW-Event)
	0010	General Purpose Interrupt 2 (I_2)	
	0001	General Purpose Interrupt 1 (I_1)	DMA (TR)TX
LOWEST	0000	General Purpose Interrupt 0 (I_0)	DMA (RR)REC

INTERRUPT CONTROL REGISTERS

Figure 8

INTERRUPT ENABLE REGISTERS

ADDRESS		7	6	5	4	3	2	1	0
Indirect Port 5	A (IERA)	GPIP 7	GPIP 6	TIMER A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TIMER B
Indirect Port 4	B (IERB)	GPIP 5	GPIP 4	TIMER C	TIMER D	GPIP 3	GPIP 2	GPIP 1	GPIP 0

INTERRUPT MASK REGISTERS

		7	6	5	4	3	2	1	0
Port 7	A (IMRA)	GPIP 7	GPIP 6	TIMER A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TIMER B
Port 6	B (IMRB)	GPIP 5	GPIP 4	TIMER C	TIMER D	GPIP 3	GPIP 2	GPIP 1	GPIP 0

1 = UNMASKED, 0 = MASKED

INTERRUPT PENDING REGISTERS

		7	6	5	4	3	2	1	0
Port 3	A (IPRA)	GPIP 7	GPIP 6	TIMER A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TIMER B
Port 2	B (IPRB)	GPIP 5	GPIP 4	TIMER C	TIMER D	GPIP 3	GPIP 2	GPIP 1	GPIP 0

WRITING 0 = CLEAR
WRITING 1 = UNCHANGED

INTERRUPT CONTROL REGISTERS (Continued)

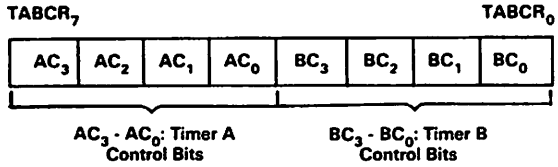
Figure 8

INTERRUPT SERVICE REGISTERS

ADDRESS		7	6	5	4	3	2	1	0
Port 5	A (ISRA)	GPIP 7	GPIP 6	TIMER A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TIMER B
Port 4	B (ISRB)	GPIP 5	GPIP 4	TIMER C	TIMER D	GPIP 3	GPIP 2	GPIP 1	GPIP 0

TIMER A and B CONTROL REGISTER (TABCR) Port 9

Figure 9

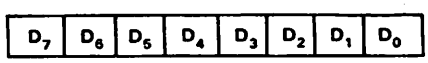


The four control bits are used to select the timer mode and prescale value, as follows:

CONTROL BIT DEFINITION

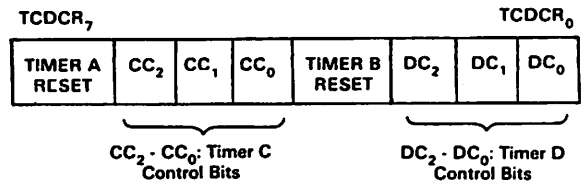
C_3	C_2	C_1	C_0	
0	0	0	0	Timer Stopped
0	0	0	1	Delay Mode, ÷4 Prescale
0	0	1	0	Delay Mode, ÷10 Prescale
0	0	1	1	Delay Mode, ÷16 Prescale
0	1	0	0	Delay Mode, ÷50 Prescale
0	1	0	1	Delay Mode, ÷64 Prescale
0	1	1	0	Delay Mode, ÷100 Prescale
0	1	1	1	Delay Mode, ÷200 Prescale
1	0	0	0	Event Count Mode
1	0	0	1	Pulse Width Mode, ÷4 Prescale
1	0	1	0	Pulse Width Mode, ÷10 Prescale
1	0	1	1	Pulse Width Mode, ÷16 Prescale
1	1	0	0	Pulse Width Mode, ÷50 Prescale
1	1	0	1	Pulse Width Mode, ÷64 Prescale
1	1	1	0	Pulse Width Mode, ÷100 Prescale
1	1	1	1	Pulse Width Mode, ÷200 Prescale

TIMER A DATA REGISTER AND TIMER B DATA REGISTER (TADR, TBDR) Port B & Port A



TIMER C and D CONTROL REGISTER (TCDCR) Indirect Port 7

Figure 10

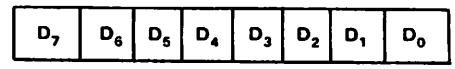


Three control bits are used to control each timer, as defined below:

CONTROL BIT DEFINITION

C_2	C_1	C_0	
0	0	0	Timer Stopped
0	0	1	Delay Mode, ÷4 Prescale
0	1	0	Delay Mode, ÷10 Prescale
0	1	1	Delay Mode, ÷16 Prescale
1	0	0	Delay Mode, ÷50 Prescale
1	0	1	Delay Mode, ÷64 Prescale
1	1	0	Delay Mode, ÷100 Prescale
1	1	1	Delay Mode, ÷200 Prescale

TIMER C DATA REGISTER and TIMER D DATA REGISTER (TCDR, TDDR) Indirect, Port 2 and Indirect Port 1



TIMERS

Four timers are available on the MK3801. Two provide full service features including delay timer operation, event counter operation, pulse width measurement operation, and pulse generation. The two other timers provide delay timer features only, and may be used for baud rate generators for use with the USART.

All timers are prescaler/counter timers, with a common independent clock input, and are not required to be operated from the system clock. In addition, all timers have a time-out output function that toggles each time the timer times out.

TIMER CONTROL REGISTERS

The 4 timers (A,B,C, and D) are programmed via 2 control registers and 4 timer data registers. Timers A and B are controlled by a single register (TABCR) and two timer data registers (TADR,TBDR). Timers C and D are controlled by a second control register (TCDCR) and two timer data

registers (TCDR, TDDR). Bits in the control registers allow the selection of operational mode, prescale, and control, while the data registers are used to read the timer or write the time constant register. General Purpose I/O Interrupt pins 3 (TB) and 4 (TA) are used for timer B and A inputs in event and pulse width modes. Figure 9 illustrates the Control and Data Register for timers A and B, while Figure 10 illustrates the Control and Data registers for timers C and D.

USART

Serial Communication is provided by the USART, which is capable of either asynchronous or synchronous operation. Variable word width and start/stop bit configurations are available under software control for asynchronous operation. For synchronous operation, a Sync Word is provided to establish synchronization during receive operations. The Sync Word will also be repeatedly transmitted when no other data is available for transmission. Operational modes exist to allow stripping of all Sync Words received in synchronous operation, and to allow the operation of DMA control handshake lines by the USART through General Purpose I/O Port lines 0 and 1. Separate receive and transmit clocks are available, and separate receive and transmit status and data bytes allow independent operation of the transmit and receive sections.

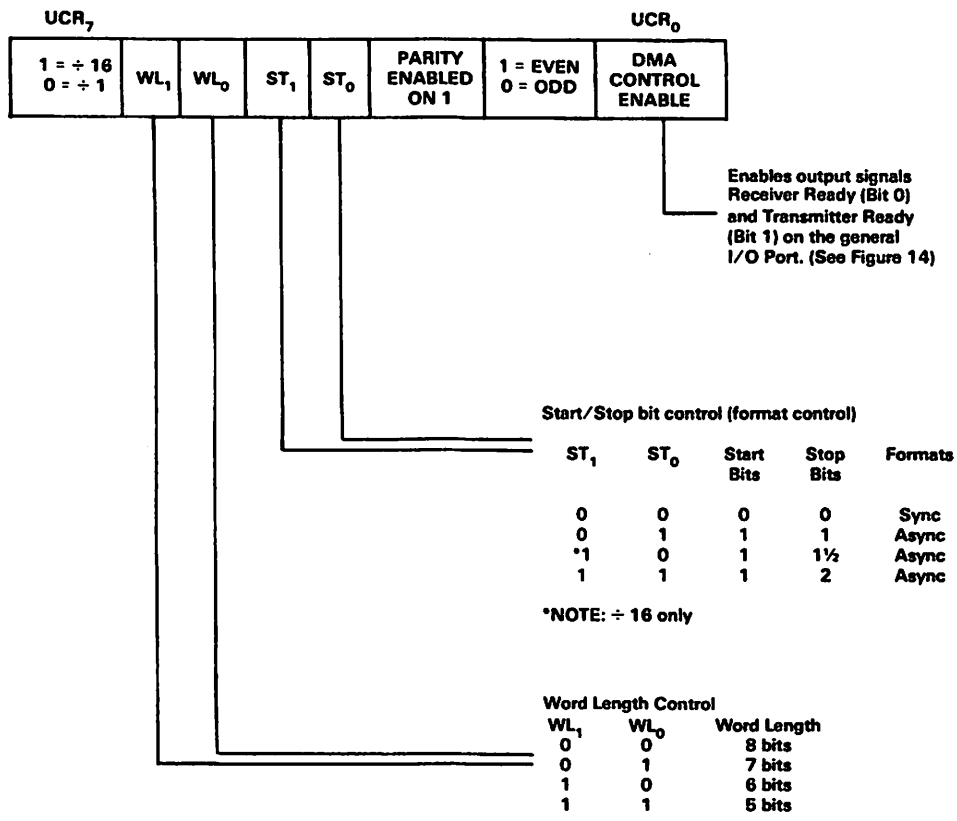
USART CONTROL REGISTERS

The USART is provided with 3 control/status registers and a data register. The programmer may specify operational parameters for the USART via the Control Register, as shown in Figure 11. Status of both the Receiver and Transmitter sections is accessed by means of the 2 Status Registers, as shown in Figure 12. Data written to the Data Register is passed to the transmitter, while reading the data register will access data received by the USART. The USART Data Register form is illustrated in Figure 13.

ERROR CONDITIONS

Error conditions in the USART are determined by monitoring the Receive Status Register (Port D) and the Transmitter Status Register (Port E). These error conditions are only valid for each word boundary and are not latched. When executing block transfers of data, it is necessary to save any errors so that they can be checked at the end of a block. In order to save error conditions during data transfer, the STI interrupt controller may be used by enabling error

USART CONTROL REGISTER (UCR) Port C
Figure 11



RECEIVER STATUS REGISTER (RSR) Port D

Figure 12

RSR_7							RSR_0
BUFFER FULL	OVERRUN ERROR	PARITY ERROR	FRAME ERROR	FOUND/SEARCH OR BREAK DETECT	MATCH/CHARACTER IN PROGRESS	SYNC STRIP ENABLE	RECEIVER ENABLE

TRANSMITTER STATUS REGISTER (TSR) Port E

TSR_7							TSR_0
BUFFER EMPTY	UNDERRUN ERROR	AUTO TURNAROUND	END OF TRANSMISSION	BREAK	HIGH	LOW	TRANSMITTER ENABLE

H	L	Serial Output State
0	0	Hi-Z
0	1	Low ("0")
1	0	High
1	1	Loop*

*Connects transmitter output to receiver input. In loopback mode, transmitter goes high when disabled. Also connects clocks with TC given priority.

USART DATA REGISTER (UDR) Port F

Figure 13

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

GENERAL PURPOSE I/O CONTROL REGISTERS

Figure 14

ACTIVE EDGE CONTROL REGISTER (AER) Indirect Port 3

1 = RISING
0 = FALLING

GPIP 7	GPIP 6	GPIP 5	GPIP 4	GPIP 3	GPIP 2	GPIP 1	GPIP 0

DATA DIRECTION REGISTER (DDR) Indirect Port 6

1 = OUTPUT
0 = INPUT

GPIP 7	GPIP 6	GPIP 5	GPIP 4	GPIP 3	GPIP 2	GPIP 1	GPIP 0

GENERAL PURPOSE I/O DATA REGISTER (GPIP) Port 1

GPIP 7	GPIP 6	GPIP 5	GPIP 4	GPIP 3	GPIP 2	GPIP 1 (TR)	GPIP 0 (RR)
			TIMER A INPUT	TIMER B INPUT			

interrupts (Port 5, Indirect) for the desired channel (Receive error or Transmit error) and by masking these bits off (Port 7). Once the transfer is complete, the Interrupt Pending Register (Port 3) can be polled to determine the presence of a pending error interrupt, and therefore an error.

GENERAL PURPOSE I/O - INTERRUPT PORT

The General Purpose I/O - Interrupt Port provides eight I/O lines that may be operated either as inputs or outputs under software control. In addition, each line may generate an interrupt on either a positive going edge or a negative going edge of the input signal.

Two of the lines in this port provide auxiliary input functions for the timers in the pulse width measurement mode and the event counter mode. Two others serve as auxiliary output lines for the USART, one indicating the Receive Buffer Full condition (RR) and the other indicating the Transmitter Buffer Empty condition (TR). These may be used as handshake signals for a DMA controller or other external control circuitry.

GENERAL PURPOSE I/O CONTROL REGISTERS

The General Purpose I/O and Interrupt Port has 2 control registers. One allows the programmer to specify the Active Edge for each bit that will trigger the interrupt associated with that bit. The other register specifies the Data Direction (input and output) associated with each bit. The third register is the actual data I/O register used to input or output data to the port. When the USART is programmed to use DMA signals, this overrides the GPIP data and the DDR. The General Purpose I/O Control and Data Registers are illustrated in Figure 14.

μPD765A

INSTRUCTION SET ① ②

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
READ DATA										
Command	W	MT	MF	SK	0	0	1	1	0	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	——— C ———								Sector ID information prior to Command execution. The 4 bytes are commanded against header on Floppy Disk
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— DTL ———								
Execution										Data-transfer between the FDD and main-system
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								Sector ID information after Command execution
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								
READ DELETED DATA										
Command	W	MT	MF	SK	0	1	1	0	0	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	——— C ———								Sector ID information prior to Command execution. The 4 bytes are commanded against header on Floppy Disk
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— DTL ———								
Execution										Data-transfer between the FDD and main-system
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								Sector ID information after Command execution
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								
WRITE DATA										
Command	W	MT	MF	0	0	0	1	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	——— C ———								Sector ID information prior to Command execution. The 4 bytes are commanded against header on Floppy Disk.
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— DTL ———								
Execution										Data-transfer between the main-system and FDD
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								Sector ID information after Command execution
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								
WRITE DELETED DATA										
Command	W	MT	MF	0	0	1	0	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	——— C ———								Sector ID information prior to Command execution. The 4 bytes are commanded against header on Floppy Disk.
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— DTL ———								
Execution										Data-transfer between the FDD and main-system
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								Sector ID information after Command execution
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								
READ A TRACK										
Command	W	0	MF	SK	0	0	0	1	0	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	——— C ———								Sector ID information prior to Command execution
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— DTL ———								
Execution										Data transfer between the FDD and main-system. FDC reads all data fields from index hole to EOT.
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								Sector ID information after Command execution
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								
READ ID										
Command	W	0	MF	0	0	1	0	1	0	Commands
	W	X	X	X	X	X	HD	US1	US0	
Execution										The first correct ID information on the Cylinder is stored in Data Register
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								Sector ID information read during Execution Phase from Floppy Disk
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								
FORMAT A TRACK										
Command	W	0	MF	0	0	1	1	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	——— N ———								Bytes/Sector
	W	——— SC ———								Sectors/Track
	W	——— GPL ———								Gap 3
	W	——— D ———								Filler Byte
Execution										FDC formats an entire track
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								In this case, the ID information has no meaning
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								
SCAN EQUAL										
Command	W	MT	MF	SK	1	0	0	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	——— C ———								Sector ID information prior to Command execution
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— STP ———								
Execution										Data-compared between the FDD and main-system
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								Sector ID information after Command execution
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								

Note: ① Symbols used in this table are described at the end of this section.
② A_0 should equal binary 1 for all operations.
③ X = Don't care usually made to equal binary 0

INSTRUCTION SET (CONT.)

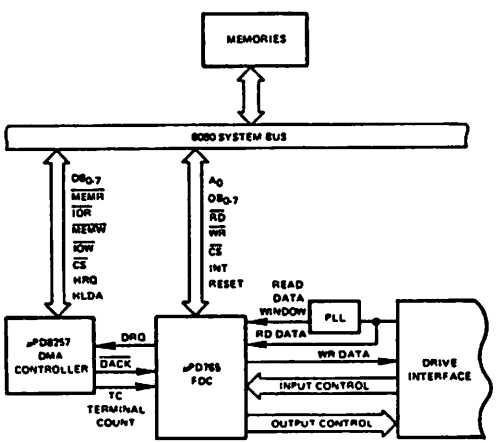
PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
SCAN LOW OR EQUAL										
Command	W	MT	MF	SK	1	1	0	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	——— C ———								Sector ID information prior Command execution
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— STP ———								
Execution										Data-compared between the FDD and main-system
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								Sector ID information after Command execution
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								
SCAN HIGH OR EQUAL										
Command	W	MT	MF	SK	1	1	1	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	——— C ———								Sector ID information prior Command execution
	W	——— H ———								
	W	——— R ———								
	W	——— N ———								
	W	——— EOT ———								
	W	——— GPL ———								
	W	——— STP ———								
Execution										Data-compared between the FDD and main-system
Result	R	——— ST 0 ———								Status information after Command execution
	R	——— ST 1 ———								
	R	——— ST 2 ———								
	R	——— C ———								Sector ID information after Command execution
	R	——— H ———								
	R	——— R ———								
	R	——— N ———								

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
RECALIBRATE										
Command	W	0	0	0	0	0	1	1	1	Command Codes
Execution	W	X	X	X	X	X	0	US1	US0	Head retracted to Track 0
SENSE INTERRUPT STATUS										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R	——— STO ———								Status information at the end of each operation about the FDC
	R	——— PCN ———								
SPECIFY										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	——— SRT ———						——— HUT ———		
	W	——— HLT ———							ND	
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
Result	R	——— ST 3 ———								Status information about FDD
SEEK										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
Execution	W	——— NCN ———								Head is positioned over proper Cylinder on Diskette
INVALID										
Command	W	——— Invalid Codes ———								Invalid Command Codes (Disable – FDC goes into Standby State)
Result	R	——— ST 0 ———								ST 0 = 80 (16)

COMMAND SYMBOL DESCRIPTION

SYMBOL	NAME	DESCRIPTION
A_0	Address Line 0	A_0 controls selection of Main Status Register (A_0 = 0) or Data Register (A_0 = 1)
C	Cylinder Number	C stands for the current/selected Cylinder (track) number 0 through 76 of the medium.
D	Data	D stands for the data pattern which is going to be written into a Sector.
D_7-D_0	Data Bus	8-bit Data Bus, where D_7 stands for a most significant bit, and D_0 stands for a least significant bit.
DTL	Data Length	When N is defined as 00 DTL stands for the data length which users are going to read out or write into the Sector.
EOT	End of Track	EOT stands for the final Sector number on a Cylinder. During Read or Write operation FDC will stop data transfer after a sector # equal to EOT.
GPL	Gap Length	GPL stands for the length of Gap 3. During Read/Write commands this value determines the number of bytes that VCOs will stay low after two CRC bytes. During Format command it determines the size of Gap 3.
H	Head Address	H stands for head number 0 or 1, as specified in ID field.
HD	Head	HD stands for a selected head number 0 or 1 and controls the polarity of pin 27. (H = HD in all command words.)
HLT	Head Load Time	HLT stands for the head load time in the FDD (2 to 254 ms in 2 ms increments).
HUT	Head Unload Time	HUT stands for the head unload time after a read or write operation has occurred (16 to 240 ms in 16 ms increments).
MF	FM or MFM Mode	If MF is low, FM mode is selected, and if it is high, MFM mode is selected.
MT	Multi-Track	If MT is high, a multi-track operation is to be performed. (If MT = 1 after finishing Read/Write operation on side 0 FDC will automatically start searching for sector 1 on side 1.

SYMBOL	NAME	DESCRIPTION
N	Number	N stands for the number of data bytes written in a Sector.
NCN	New Cylinder Number	NCN stands for a new Cylinder number, which is going to be reached as a result of the Seek operation. Desired position of Head.
ND	Non-DMA Mode	ND stands for operation in the Non-DMA Mode.
PCN	Present Cylinder Number	PCN stands for the Cylinder number at the completion of SENSE INTERRUPT STATUS Command. Position of Head at present time.
R	Record	R stands for the Sector number, which will be read or written.
R/W	Read/Write	R/W stands for either Read (R) or Write (W) signal.
SC	Sector	SC indicates the number of Sectors per Cylinder.
SK	Skip	SK stands for Skip Deleted Data Address Mark.
SRT	Step Rate Time	SRT stands for the Stepping Rate for the FDD. (1 to 16 ms in 1 ms increments.) Stepping Rate applies to all drives, (F = 1 ms, E = 2 ms, etc.).
ST 0 ST 1 ST 2 ST 3	Status 0 Status 1 Status 2 Status 3	ST 0-3 stand for one of four registers which store the status information after a command has been executed. This information is available during the result phase after command execution. These registers should not be confused with the main status register (selected by A_0 = 0). ST 0-3 may be read only after a command has been executed and contain information relevant to that particular command.
STP		During a Scan operation, if STP = 1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA); and if STP = 2, then alternate sectors are read and compared.
US0, US1	Unit Select	US stands for a selected drive number 0 or 1.



PROCESSOR INTERFACE

During Command or Result Phases the Main Status Register (described earlier) must be read by the processor before each byte of information is written into or read from the Data Register. After each byte of data read or written to Data Register, CPU should wait for 12 μs before reading MSR. Bits D6 and D7 in the Main Status Register must be in a 0 and 1 state, respectively, before each byte of the command word may be written into the μPD765. Many of the commands require multiple bytes, and as a result the Main Status Register must be read prior to each byte transfer to the μPD765. On the other hand, during the Result Phase, D6 and D7 in the Main Status Register must both be 1's (D6 = 1 and D7 = 1) before reading each byte from the Data Register. Note, this reading of the Main Status Register before each byte transfer to the μPD765 is required in only the Command and Result Phases, and NOT during the Execution Phase.

During the Execution Phase, the Main Status Register need not be read. If the μPD765 is in the NON-DMA Mode, then the receipt of each data byte (if μPD765 is reading data from FDD) is indicated by an Interrupt signal on pin 18 (INT = 1). The generation of a Read signal ($\overline{RD}$ = 0) or Write signal ($\overline{WR}$ = 0) will reset the Interrupt as well as output the Data onto the Data Bus. If the processor cannot handle Interrupts fast enough (every 13 μs) for MFM and 27 μs for FM mode, then it may poll the Main Status Register and then bit D7 (RQM) functions just like the Interrupt signal. If a Write Command is in process then the WR signal performs the reset to the Interrupt signal.

If the μPD765 is in the DMA Mode, no Interrupts are generated during the Execution Phase. The μPD765 generates DRQ's (DMA Requests) when each byte of data is available. The DMA Controller responds to this request with both a $\overline{DACK}$ = 0 (DMA Acknowledge) and a $\overline{RD}$ = 0 (Read signal). When the DMA Acknowledge signal goes low ($\overline{DACK}$ = 0) then the DMA Request is reset (DRQ = 0). If a Write Command has been programmed then a $\overline{WR}$ signal will appear instead of $\overline{RD}$. After the Execution Phase has been completed (Terminal Count has occurred) or EOT sector was read/written, then an Interrupt will occur (INT = 1). This signifies the beginning of the Result Phase. When the first byte of data is read during the Result Phase, the Interrupt is automatically reset (INT = 0).

It is important to note that during the Result Phase all bytes shown in the Command Table must be read. The Read Data Command, for example has seven bytes of data in the Result Phase. All seven bytes must be read in order to successfully complete the Read Data Command. The μPD765 will not accept a new command until all seven bytes have been read. Other commands may require fewer bytes to be read during the Result Phase.

The μPD765 contains five Status Registers. The Main Status Register mentioned above may be read by the processor at any time. The other four Status Registers (ST0, ST1, ST2, and ST3) are only available during the Result Phase, and may be read only after completing a command. The particular command which has been executed determines how many of the Status Registers will be read.

The bytes of data which are sent to the μPD765 to form the Command Phase, and are read out of the μPD765 in the Result Phase, must occur in the order shown in the Command Table. That is, the Command Code must be sent first and the other bytes sent in the prescribed sequence. No foreshortening of the Command or Result Phases are allowed. After the last byte of data in the Command Phase is sent to the μPD765, the Execution Phase automatically starts. In a similar fashion, when the last byte of data is read out in the Result Phase, the command is automatically ended and the μPD765 is ready for a new command.

POLLING FEATURE OF THE μPD765

After the Specify command has been sent to the μPD765, the Unit Select line US0 and US1 will automatically go into a polling mode. In between commands (and between step pulses in the SEEK command) the μPD765 polls all four FDD's looking for a change in the Ready line from any of the drives. If the Ready line changes state (usually due to a door opening or closing) then the μPD765 will generate an interrupt. When Status Register 0 (ST0) is read (after Sense Interrupt Status is issued), Not Ready (NR) will be indicated. The polling of the Ready line by the μPD765 occurs continuously between commands, thus notifying the processor which drives are on or off line. Each drive is polled every 1.024 ms except during the Read/Write commands.

FUNCTIONAL DESCRIPTION OF COMMANDS

READ DATA

A set of nine (9) byte words are required to place the FDC into the Read Data Mode. After the Read Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the Specify Command), and begins reading ID Address Marks and ID fields. When the current sector number ("R") stored in the ID Register (IDR) compares with the sector number read off the diskette, then the FDC outputs data (from the data field) byte-to-byte to the main system via the data bus.

After completion of the read operation from the current sector, the Sector Number is incremented by one, and the data from the next sector is read and output on the data bus. This continuous read function is called a "Multi-Sector Read Operation." The Read Data Command may be terminated by the receipt of a Terminal Count signal. TC should be issued at the same time that the DACK for the last byte of data is sent. Upon receipt of this signal, the FDC stops outputting data to the processor, but will continue to read data from the current sector, check CRC (Cyclic Redundancy Count) bytes, and then at the end of the sector terminate the Read Data Command.
The amount of data which can be handled with a single command to the FDC depends upon MT (multi-track), MF (MFM/FM), and N (Number of Bytes/Sector). Table 1 below shows the Transfer Capacity.

Multi-Track MT	MFM/FM MF	Bytes/Sector N	Maximum Transfer Capacity (Bytes/Sector) (Number of Sectors)	Final Sector Read from Diskette
0	0	00	(128) (26) = 3,328	26 at Side 0
0	1	01	(256) (26) = 6,656	or 26 at Side 1
1	0	00	(128) (52) = 6,656	26 at Side 1
1	1	01	(256) (52) = 13,312	
0	0	01	(256) (15) = 3,840	15 at Side 0
0	1	02	(512) (15) = 7,680	or 15 at Side 1
1	0	01	(256) (30) = 7,680	15 at Side 1
1	1	02	(512) (30) = 15,360	
0	0	02	(512) (8) = 4,096	8 at Side 0
0	1	03	(1024) (8) = 8,192	or 8 at Side 1
1	0	02	(512) (16) = 8,192	8 at Side 1
1	1	03	(1024) (16) = 16,384	

Table 1. Transfer Capacity

The "multi-track" function (MT) allows the FDC to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector 1, Side 0 and completing at Sector L, Side 1 (Sector L = last sector on the side). Note, this function pertains to only one cylinder (the same track) on each side of the diskette.

When N = 0, then DTL defines the data length which the FDC must treat as a sector. If DTL is smaller than the actual data length in a Sector, the data beyond DTL in the Sector, is not sent to the Data Bus. The FDC reads (internally) the complete Sector performing the CRC check, and depending upon the manner of command termination, may perform a Multi-Sector Read Operation. When N is non-zero, then DTL has no meaning and should be set to FF Hexidecimal.

At the completion of the Read Data Command, the head is not unloaded until after Head Unload Time Interval (specified in the Specify Command) has elapsed. If the processor issues another command before the head unloads then the head settling time may be saved between subsequent reads. This time out is particularly valuable when a diskette is copied from one drive to another.

If the FDC detects the Index Hole twice without finding the right sector, (indicated in "R"), then the FDC sets the ND (No Data) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

After reading the ID and Data Fields in each sector, the FDC checks the CRC bytes. If a read error is detected (incorrect CRC in ID field), the FDC sets the DE (Data Error) flag in Status Register 1 to a 1 (high), and if a CRC error occurs in the Data Field the FDC also sets the DD (Data Error in Data Field) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

If the FDC reads a Deleted Data Address Mark off the diskette, and the SK bit (bit D5 in the first Command Word) is not set (SK = 0), then the FDC sets the CM (Control Mark) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command, after reading all the data in the Sector. If SK = 1, the FDC skips the sector with the Deleted Data Address Mark and reads the next sector. The CRC bits in the deleted data field are not checked when SK = 1.

During disk data transfers between the FDC and the processor, via the data bus, the FDC must be serviced by the processor every 27 μs in the FM Mode, and every 13 μs in the MFM Mode, or the FDC sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command.

If the processor terminates a read (or write) operation in the FDC, then the ID Information in the Result Phase is dependent upon the state of the MT bit and EOT byte. Table 2 shows the values for C, H, R, and N, when the processor terminates the Command.

MT	HD	Final Sector Transferred to Processor	ID Information at Result Phase			
			C	H	R	N
0	0	Less than EOT	NC	NC	R + 1	NC
	0	Equal to EOT	C + 1	NC	R = 01	NC
	1	Less than EOT	NC	NC	R + 1	NC
	1	Equal to EOT	C + 1	NC	R = 01	NC
1	0	Less than EOT	NC	NC	R + 1	NC
	0	Equal to EOT	NC	LSB	R = 01	NC
	1	Less than EOT	NC	NC	R + 1	NC
	1	Equal to EOT	C + 1	LSB	R = 01	NC

Notes: 1 NC (No Change): The same value as the one at the beginning of command execution.

2 LSB (Least Significant Bit): The least significant bit of H is complemented.

WRITE DATA

A set of nine (9) bytes are required to set the FDC into the Write Data mode. After the Write Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified Head Settling Time (defined in the Specify Command), and begins reading ID Fields. When all four bytes loaded during the command (C, H, R, N) match the four bytes of the ID field from the diskette, the FDC takes data from the processor byte-by-byte via the data bus, and outputs it to the FDD.

After writing data into the current sector, the Sector Number stored in "R" is incremented by one, and the next data field is written into. The FDC continues this "Multi-Sector Write Operation" until the issuance of a Terminal Count signal. If a Terminal Count signal is sent to the FDC it continues writing into the current sector to complete the data field. If the Terminal Count signal is received while a data field is being written then the remainder of the data field is filled with 00 (zeros).

The FDC reads the ID field of each sector and checks the CRC bytes. If the FDC detects a read error (incorrect CRC) in one of the ID Fields, it sets the DE (Data Error) flag of Status Register 1 to a 1 (high), and terminates the Write Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

The Write Command operates in much the same manner as the Read Command. The following items are the same, and one should refer to the Read Data Command for details:

- Transfer Capacity
- EN (End of Cylinder) Flag
- ND (No Data) Flag
- Head Unload Time Interval
- ID Information when the processor terminates command (see Table 2)
- Definition of DTL when N = 0 and when N ≠ 0

In the Write Data mode, data transfers between the processor and FDC, via the Data Bus, must occur every 27 μs in the FM mode, and every 13 μs in the MFM mode. If the time interval between data transfers is longer than this then the FDC sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Write Data Command. (Status Register 0 also has bit 7 and 6 set to 0 and 1 respectively.)

WRITE DELETED DATA

This command is the same as the Write Data Command except a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark.

READ DELETED DATA

This command is the same as the Read Data Command except that when the FDC detects a Data Address Mark at the beginning of a Data Field (and SK = 0 (low)), it will read all the data in the sector and set the CM flag in Status Register 2 to a 1 (high), and then terminate the command. If SK = 1, then the FDC skips the sector with the Data Address Mark and reads the next sector.

READ A TRACK

This command is similar to READ DATA Command except that this is a continuous READ operation where the entire data field from each of the sectors are read. Immediately after encountering the INDEX HOLE, the FDC starts reading all data fields on the track, as continuous blocks of data. If the FDC finds an error in the ID or DATA CRC check bytes, it continues to read data from the track. The FDC compares the ID information read from each sector with the value stored in the IDR, and sets the ND flag of Status Register 1 to a 1 (high) if there is no comparison. Multi-track or skip operations are not allowed with this command.

This command terminates when number of sectors read is equal to EOT. If the FDC does not find an ID Address Mark on the diskette after it encounters the INDEX HOLE for the second time, then it sets the MA (missing address mark) flag in Status Register 1 to a 1 (high), and terminates the command. (Status Register 0 has bits 7 and 6 set to 0 and 1 respectively.)

FUNCTIONAL DESCRIPTION OF COMMANDS (CONT.)

READ ID

The READ ID Command is used to give the present position of the recording head. The FDC stores the values from the first ID Field it is able to read. If no proper ID Address Mark is found on the diskette, before the INDEX HOLE is encountered for the second time then the MA (Missing Address Mark) flag in Status Register 1 is set to a 1 (high), and if no data is found then the ND (No Data) flag is also set in Status Register 1 to a 1 (high). The command is then terminated with Bits 7 and 6 in Status Register 0 set to 0 and 1 respectively. During this command there is no data transfer between FDC and the CPU except during the result phase.

FORMAT A TRACK

The Format Command allows an entire track to be formatted. After the INDEX HOLE is detected, Data is written on the Diskette, Gaps, Address Marks, ID Fields and Data Fields, all per the IBM System 34 (Double Density) or System 3740 (Single Density) Format are recorded. The particular format which will be written is controlled by the values programmed into N (number of bytes/sector), SC (sectors/cylinder), GPL (Gap Length), and D (Data Pattern) which are supplied by the processor during the Command Phase. The Data Field is filled with the Byte of data stored in D. The ID Field for each sector is supplied by the processor; that is, four data requests per sector are made by the FDC for C (Cylinder Number), H (Head Number), R (Sector Number) and N (Number of Bytes/Sector). This allows the diskette to be formatted with non-sequential sector numbers, if desired.

The processor must send new values for C, H, R, and N to the μPD765 for each sector on the track. If FDC is set for DMA mode, it will issue 4 DMA requests per sector. If it is set for interrupt mode, it will issue four interrupts per sector and the processor must supply C, H, R and N load for each sector. The contents of the R register is incremented by one after each sector is formatted, thus, the R register contains a value of R when it is read during the Result Phase. This incrementing and formatting continues for the whole track until the FDC encounters the INDEX HOLE for the second time, whereupon it terminates the command.

If a FAULT signal is received from the FDD at the end of a write operation, then the FDC sets the EC flag of Status Register 0 to a 1 (high), and terminates the command after setting bits 7 and 6 of Status Register 0 to 0 and 1 respectively. Also the loss of a READY signal at the beginning of a command execution phase causes bits 7 and 6 of Status Register 0 to be set to 0 and 1 respectively.

Table 3 shows the relationship between N, SC, and GPL for various sector sizes.

		8" STANDARD FLOPPY					5¼" MINI FLOPPY			
FORMAT	SECTOR SIZE	N	SC	GPL ①	GPL ②	SECTOR SIZE	N	SC	GPL ①	GPL ②
FM Mode	128 bytes/Sector	00	1A	07	1B	128 bytes/Sector	00	12	07	09
	256	01	0F	0E	2A	128	00	10	10	19
	512	02	08	1B	3A	256	01	08	18	30
	1024 bytes/Sector	03	04	47	8A	512	02	04	46	87
	2048	04	02	C8	FF	1024	03	02	C8	FF
	4096	05	01	C8	FF	2048	04	01	C8	FF
MFM Mode	256	01	1A	0E	36	256	01	12	0A	0C
	512	02	0F	1B	54	256	01	10	20	32
	1024	03	08	35	74	512	02	08	2A	50
	2048	04	04	99	FF	1024	03	04	80	F0
	4096	05	02	C8	FF	2048	04	02	C8	FF
	8192	06	01	C8	FF	4096	05	01	C8	FF

Table 3

Note: ① Suggested values of GPL in Read or Write Commands to avoid splice point between data field and ID field of contiguous sections.
② Suggested values of GPL in format command.
③ In MFM mode FDC can not perform a read/write/format operation with 128 bytes/sector. (N = 00)
④ All the values are hexidecimal.

SCAN COMMANDS

The SCAN Commands allow data which is being read from the diskette to be compared against data which is being supplied from the main system. The FDC compares the data on a byte-by-byte basis, and looks for a sector of data which meets the conditions of $D_{FDD} = D_{Processor}$, $D_{FDD} < D_{Processor}$, or $D_{FDD} > D_{Processor}$. The hexidecimal byte of FF either from memory or from FDD can be used as a mask byte because it always meet the condition of the compare. Ones complement arithmetic is used for comparison (FF = largest number, 00 = smallest number). After a whole sector of data is compared, if the conditions are not met, the sector number is incremental (R = STP + R), and the scan operation is continued. The scan operation continues until one of the following conditions occur; the conditions for scan are met (equal, low, or high), the last sector on the track is reached (EOT), or the terminal count signal is received.

If the conditions for scan are met then the FDC sets the SH (Scan Hit) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. If the conditions for scan are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), then the FDC sets the SN (Scan Not Satisfied) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. The receipt of a TERMINAL COUNT signal from the Processor or DMA Controller during the scan operation will cause the FDC to complete the comparison of the particular byte which is in process, and then to terminate the command. Table 4 shows the status of bits SH and SN under various conditions of SCAN.

COMMAND	STATUS REGISTER 2		COMMENTS
	BIT 2 = SN	BIT 3 = SH	
Scan Equal	0	1	$D_{FDD} = D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan Low or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} < D_{Processor}$
	1	0	$D_{FDD} > D_{Processor}$
Scan High or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} > D_{Processor}$
	1	0	$D_{FDD} < D_{Processor}$

Table 4

If the FDC encounters a Deleted Data Address Mark on one of the sectors (and SK = 0), then it regards the sector as the last sector on the cylinder, sets CM (Control Mark) flag of Status Register 2 to a 1 (high) and terminates the command. If SK = 1, the FDC skips the sector with the Deleted Address Mark, and reads the next sector. In the second case (SK = 1), the FDC sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) in order to show that a Deleted Sector had been encountered.

When either the STP (contiguous sectors = 01, or alternate sectors = 02 sectors are read) or the MT (Multi-Track) are programmed, it is necessary to remember that the last sector on the track must be read. For example, if STP = 02, MT = 0, the sectors are numbered sequentially 1 through 26, and we start the Scan Command at sector 21; the following will happen. Sectors 21, 23, and 25 will be read, then the next sector (26) will be skipped and the Index Hole will be encountered before the EOT value of 26 can be read. This will result in an abnormal termination of the command. If the EOT had been set at 25 or the scanning started at sector 20, then the Scan Command would be completed in a normal manner.

During the Scan Command data is supplied by either the processor or DMA Controller for comparison against the data read from the diskette. In order to avoid having the OR (Over Run) flag set in Status Register 1, it is necessary to have the data available in less than 27 μs (FM Mode) or 13 μs (MFM Mode). If an Overrun occurs the FDC ends the command with bits 7 and 6 of Status Register 0 set to 0 and 1, respectively.

SEEK

The read/write head within the FDD is moved from cylinder to cylinder under control of the Seek Command. FDC has four independent Present Cylinder Registers for each drive. They are clear only after Recalibrate command. The FDC compares the PCN (Present Cylinder Number) which is the current head position with the NCN (New Cylinder Number), and if there is a difference performs the following operation:

PCN < NCN: Direction signal to FDD set to a 1 (high), and Step Pulses are issued. (Step In.)
PCN > NCN: Direction signal to FDD set to a 0 (low), and Step Pulses are issued. (Step Out.)

The rate at which Step Pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY Command. After each Step Pulse is issued NCN is compared against PCN, and when NCN = PCN, then the SE (Seek End) flag is set in Status Register 0 to a 1 (high), and the command is terminated. At this point FDC interrupt goes high. Bits $D8_0$-$D8_3$ in Main Status Register are set during seek operation and are cleared by Sense Interrupt Status command.

During the Command Phase of the Seek operation the FDC is in the FDC BUSY state, but during the Execution Phase it is in the NON BUSY state. While the FDC is in the NON BUSY state, another Seek Command may be issued, and in this manner parallel seek operations may be done on up to 4 Drives at once. No other command could be issued for as long as FDC is in process of sending Step Pulses to any drive.

If an FDD is in a NOT READY state at the beginning of the command execution phase or during the seek operation, then the NR (NOT READY) flag is set in Status Register 0 to a 1 (high), and the command is terminated after bits 7 and 6 of Status Register 0 are set to 0 and 1 respectively.

If the time to write 3 bytes of seek command exceeds 150 μs, the timing between first two Step Pulses may be shorter than set in the Specify command by as much as 1 ms.

RECALIBRATE

The function of this command is to retract the read/write head within the FDD to the Track 0 position. The FDC clears the contents of the PCN counter, and checks the status of the Track 0 signal from the FDD. As long as the Track 0 signal is low, the Direction signal remains 0 (low) and Step Pulses are issued. When the Track 0 signal goes high, the SE (SEEK END) flag in Status Register 0 is set to a 1 (high) and the command is terminated. If the Track 0 signal is still low after 77 Step Pulse have been issued, the FDC sets the SE (SEEK END) and EC (EQUIPMENT CHECK) flags of Status Register 0 to both 1s (highs), and terminates the command after bits 7 and 6 of Status Register 0 is set to 0 and 1 respectively.

The ability to do overlap RECALIBRATE Commands to multiple FDDs and the loss of the READY signal, as described in the SEEK Command, also applies to the RECALIBRATE Command.

SENSE INTERRUPT STATUS

An Interrupt signal is generated by the FDC for one of the following reasons:

1. Upon entering the Result Phase of:
   - a. Read Data Command
   - b. Read a Track Command
   - c. Read ID Command
   - d. Read Deleted Data Command
   - e. Write Data Command
   - f. Format a Cylinder Command
   - g. Write Deleted Data Command
   - h. Scan Commands
2. Ready Line of FDD changes state
3. End of Seek or Recalibrate Command
4. During Execution Phase in the NON-DMA Mode

Interrupts caused by reasons 1 and 4 above occur during normal command operations and are easily discernible by the processor. During an execution phase in NON-DMA Mode, D5S in Main Status Register is high. Upon entering Result Phase this bit gets clear. Reason 1 and 4 does not require Sense Interrupt Status command. The interrupt is cleared by reading/writing data to FDC. Interrupts caused by reasons 2 and 3 above may be uniquely identified with the aid of the Sense Interrupt Status Command. This command when issued resets the interrupt signal and via bits 5, 6, and 7 of Status Register 0 identifies the cause of the interrupt.

SEEK END	INTERRUPT CODE		CAUSE
BIT 5	BIT 6	BIT 7	
0	1	1	Ready Line changed state, either polarity
1	0	0	Normal Termination of Seek or Recalibrate Command
1	1	0	Abnormal Termination of Seek or Recalibrate Command

Table 5

Neither the Seek or Recalibrate Command have a Result Phase. Therefore, it is mandatory to use the Sense Interrupt Status Command after these commands to effectively terminate them and to provide verification of where the head is positioned (PCN).

Issuing Sense Interrupt Status Command without interrupt pending is treated as an invalid command.

SPECIFY

The Specify Command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the Execution Phase of one of the Read/Write Commands to the head unload state. This timer is programmable from 16 to 240 ms in increments of 16 ms (01 = 16 ms, 02 = 32 ms . . . . 0F = 240 ms). The SRT (Step Rate Time) defines the time interval between adjacent step pulses. This timer is programmable from 1 to 16 ms in increments of 1 ms (F = 1 ms, E = 2 ms, D = 3 ms, etc.). The HLT (Head Load Time) defines the time between when the Head Load signal goes high and when the Read/Write operation starts. This timer is programmable from 2 to 254 ms in increments of 2 ms (01 = 2 ms, 02 = 4 ms, 03 = 6 ms . . . 7F = 254 ms).

The time intervals mentioned above are a direct function of the clock (CLK on pin 19). Times indicated above are for an 8 MHz clock, if the clock was reduced to 4 MHz (mini-floppy application) then all time intervals are increased by a factor of 2.

The choice of DMA or NON-DMA operation is made by the ND (NON-DMA) bit. When this bit is high (ND = 1) the NON-DMA mode is selected, and when ND = 0 the DMA mode is selected.

SENSE DRIVE STATUS

This command may be used by the processor whenever it wishes to obtain the status of the FDDs. Status Register 3 contains the Drive Status information stored internally in FDC registers.

INVALID

If an invalid command is sent to the FDC (a command not defined above), then the FDC will terminate the command after bits 7 and 6 of Status Register 0 are set to 1 and 0 respectively. No interrupt is generated by the μPD765 during this condition. Bit 6 and bit 7 (DIO and RQM) in the Main Status Register are both high ("1") indicating to the processor that the μPD765 is in the Result Phase and the contents of Status Register 0 (ST0) must be read. When the processor reads Status Register 0 it will find a 80 hex indicating an invalid command was received.

A Sense Interrupt Status Command must be sent after a Seek or Recalibrate Interrupt, otherwise the FDC will consider the next command to be an Invalid Command.

In some applications the user may wish to use this command as a No-Op command, to place the FDC in a standby or no operation state.

BIT NO.	NAME	SYMBOL	DESCRIPTION
STATUS REGISTER 0			
D_7	Interrupt Code	IC	$D_7 = 0$ and $D_6 = 0$ Normal Termination of Command, (NT). Command was completed and properly executed.
D_6			$D_7 = 0$ and $D_6 = 1$ Abnormal Termination of Command, (AT). Execution of Command was started, but was not successfully completed.
			$D_7 = 1$ and $D_6 = 0$ Invalid Command issue, (IC). Command which was issued was never started.
			$D_7 = 1$ and $D_6 = 1$ Abnormal Termination because during command execution the ready signal from FDD changed state.
D_5	Seek End	SE	When the FDC completes the SEEK Command, this flag is set to 1 (high).
D_4	Equipment Check	EC	If a fault Signal is received from the FDD, or if the Track 0 Signal fails to occur after 77 Step Pulses (Recalibrate Command) then this flag is set.
D_3	Not Ready	NR	When the FDD is in the not-ready state and a read or write command is issued, this flag is set. If a read or write command is issued to Side 1 of a single sided drive, then this flag is set.
D_2	Head Address	HD	This flag is used to indicate the state of the head at Interrupt.
D_1	Unit Select 1	US 1	These flags are used to indicate a Drive Unit. Number at Interrupt.
D_0	Unit Select 0	US 0	
STATUS REGISTER 1			
D_7	End of Cylinder	EN	When the FDC tries to access a Sector beyond the final Sector of a Cylinder, this flag is set.
D_6			Not used. This bit is always 0 (low).
D_5	Data Error	DE	When the FDC detects a CRC error in either the ID field or the data field, this flag is set.
D_4	Over Run	OR	If the FDC is not serviced by the main-systems during data transfers, within a certain time interval, this flag is set.
D_3			Not used. This bit always 0 (low).
D_2	No Data	ND	During execution of READ DATA, WRITE DELETED DATA or SCAN Command, if the FDC cannot find the Sector specified in the IDR Register, this flag is set.
			During executing the READ ID Command, if the FDC cannot read the ID field without an error, then this flag is set.
			During the execution of the READ A Cylinder Command, if the starting sector cannot be found, then this flag is set.

BIT NO.	BIT NAME	BIT SYMBOL	DESCRIPTION
			STATUS REGISTER 1 (CONT.)
D_1	Not Writable	NW	During execution of WRITE DATA, WRITE DELETED DATA or Format A Cylinder Command, if the FDC detects a write protect signal from the FDD, then this flag is set.
D_0	Missing Address Mark	MA	If the FDC cannot detect the ID Address Mark after encountering the index hole twice, then this flag is set.
			If the FDC cannot detect the Data Address Mark or Deleted Data Address Mark, this flag is set. Also at the same time, the MD (Missing Address Mark in Data Field) of Status Register 2 is set.
			STATUS REGISTER 2
D_7			Not used. This bit is always 0 (low).
D_6	Control Mark	CM	During executing the READ DATA or SCAN Command, if the FDC encounters a Sector which contains a Deleted Data Address Mark, this flag is set.
D_5	Data Error in Data Field	DD	If the FDC detects a CRC error in the data field then this flag is set.
D_4	Wrong Cylinder	WC	This bit is related with the ND bit, and when the contents of C on the medium is different from that stored in the IDR, this flag is set.
D_3	Scan Equal Hit	SH	During execution, the SCAN Command, if the condition of "equal" is satisfied, this flag is set.
D_2	Scan Not Satisfied	SN	During executing the SCAN Command, if the FDC cannot find a Sector on the cylinder which meets the condition, then this flag is set.
D_1	Bad Cylinder	BC	This bit is related with the ND bit, and when the content of C on the medium is different from that stored in the IDR and the content of C is FF, then this flag is set.
D_0	Missing Address Mark in Data Field	MD	When data is read from the medium, if the FDC cannot find a Data Address Mark or Deleted Data Address Mark, then this flag is set.
			STATUS REGISTER 3
D_7	Fault	FT	This bit is used to indicate the status of the Fault signal from the FDD.
D_6	Write Protected	WP	This bit is used to indicate the status of the Write Protected signal from the FDD.
D_5	Ready	RY	This bit is used to indicate the status of the Ready signal from the FDD.
D_4	Track 0	T0	This bit is used to indicate the status of the Track 0 signal from the FDD.
D_3	Two Side	TS	This bit is used to indicate the status of the Two Side signal from the FDD.
D_2	Head Address	HD	This bit is used to indicate the status of Side Select signal to the FDD.
D_1	Unit Select 1	US 1	This bit is used to indicate the status of the Unit Select 1 signal to the FDD.
D_0	Unit Select 0	US 0	This bit is used to indicate the status of the Unit Select 0 signal to the FDD.

PIN IDENTIFICATION

PIN NO.	PIN SYMBOL	PIN NAME	INPUT/ OUTPUT	CONNECTION TO	FUNCTION
1	RST	Reset	Input	Processor	Places FDC in idle state. Resets output lines to FDD to "0" (low). Does not affect SRT, HUT or HLT in Specify command. If ROY pin is held high during Reset, FDC will generate interrupt. 1-35 ms later. To clear this interrupt use Sense Interrupt Status command.
2	$\overline{RD}$	Read	Input①	Processor	Control signal for transfer of data from FDC to Data Bus, when "0" (low).
3	$\overline{WR}$	Write	Input①	Processor	Control signal for transfer of data to FDC via Data Bus, when "0" (low).
4	$\overline{CS}$	Chip Select	Input	Processor	IC selected when "0" (low), allowing $\overline{RD}$ and $\overline{WR}$ to be enabled.
5	A_0	Data/Status Reg Select	Input①	Processor	Selects Data Reg (A_0=1) or Status Reg (A_0=0) contents of the FDC to be sent to Data Bus.
6-13	DB_0-DB_7	Data Bus	Input① Output	Processor	Bi-Directional 8-Bit Data Bus.
14	DRQ	Data DMA Request	Output	DMA	DMA Request is being made by FDC when DRQ="1".
15	$\overline{DACK}$	DMA Acknowledge	Input	DMA	DMA cycle is active when "0" (low) and Controller is performing DMA transfer
16	TC	Terminal Count	Input	DMA	Indicates the termination of a DMA transfer when "1" (high). It terminates data transfer during Read/Write/Scan command in DMA or interrupt mode
17	IDX	Index	Input	FDD	Indicates the beginning of a disk track.
18	INT	Interrupt	Output	Processor	Interrupt Request Generated by FDC.
19	CLK	Clock	Input		Single Phase 8 MHz Squarewave Clock.
20	GND	Ground			D.C. Power Return.
21	WCK	Write Clock	Input		Write data rate to FDD. FM = 500 kHz, MFM = 1 MHz, with a pulse width of 250 ns for both FM and MFM.
22	RDW	Read Data Window	Input	Phase Lock Loop	Generated by PLL, and used to sample data from FDD.
23	RDD	Read Data	Input	FDD	Read data from FDD, containing clock and data bits.
24	VCO	VCO Sync	Output	Phase Lock Loop	Inhibits VCO in PLL when "0" (low), enables VCO when "1."
25	WE	Write Enable	Output	FDD	Enables write data loop FDD.
26	MFM	MFM Mode	Output	Phase Lock Loop	MFM mode when "1," FM mode when "0."
27	HD	Head Select	Output	FDD	Head 1 selected when "1" (high). Head 0 selected when "0" (low).
28,29	US_1,US_0	Unit Select	Output	FDD	FDD Unit Selected.
30	WDA	Write Data	Output	FDD	Serial clock and data bits to FDD.
31,32	PS_1,PS_0	Precompensation (pre-shift)	Output	FDD	Write precompensation status during MFM mode. Determines early, late, and normal times.
33	FLT/TP_0	Fault/Track 0	Input	FDD	Senses FDD fault condition. In Read/Write mode; and Track 0 condition in Seek mode.
34	WP/TS	Write Protect/ Two-Side	Input	FDD	Senses Write Protect status in Read/Write mode; and Two Side Mode in Seek mode.
35	RDY	Ready	Input	FDD	Indicates FDD is ready to send or receive data.
36	HDL	Head Load	Output	FDD	Command which causes read/write head in FDD to contact diskette.
37	FR/STP	Fit Reset/Step	Output	FDD	Resets fault F.F. in FDD in Read/Write mode, contains step pulses to move head to another cylinder in Seek mode.
38	LCT/DIR	Low Current/ Direction	Output	FDD	Lowers Write current on inner tracks in Read/Write mode, determines direction head will step in Seek mode. A fault reset pulse is issued at the beginning of each Read or Write command prior to the occurrence of the Head Load signal.
39	R/W/SEEK	Read Write/SEEK	Output	FDD	When "1" (high) Seek mode executed and when "0" (low) Read/Write mode selected.
40	V_{CC}	+5V			DC Power.

Note: ① Disabled when CS = 1.

CAPACITANCE

T_a = 25°C; f_c = 1 MHz; V_{CC} = 0V

PARAMETER	SYMBOL	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT	TEST CONDITIONS
Clock Input Capacitance	$C_{IN(\Phi)}$			20	pF	All Pins Except Pin Under Test Tied to AC Ground
Input Capacitance	C_{IN}			10	pF	
Output Capacitance	C_{OUT}			20	pF	

OPERATION

BI-DIRECTIONAL DATA-TRANSFER (13)

Data-transfer to and from the chip is accomplished through a single I/O data-line (13), using external clock-pulses. (15) After the chip-select-input (9) has been activated (to a low level), a serial 3 bit address-word will first be accepted by the chip to select particular timing-information. A fourth bit selects read or write-mode, thus specifying whether the selected timing-information should be written into the circuit or read from it at pin 13 . Using the following clock-pulses, data-transfer with the chip at pin 13 may be accomplished.

When chip select is high the I/O line is high impedance, neither accepting nor giving out data.

Addressing the data

For serial data-transfer into or out of the chip, the address- and data-line must first be activated by putting chip select to a low level.
When chip-select goes low the first 3 bits clocked into the chip by external clock-pulses will be interpreted as an address. This determines which time information is selected.

address-word	bit configuration MSB		LSB	selected time-information	counting capability
0	0	0	0	second	00...59
1	0	0	1	minute	00...59
2	0	1	0	hour	00...23
3	0	1	1	date	01...28/29.30/31
4	1	0	0	month	01...12
5	1	0	1	day of the week	01...07
6	1	1	0	year	00...99
7	1	1	1	continous data-transfer in the following sequence: hours, minutes, date, month, year, day of the week, seconds.	

Read/Write - select

The fourth bit clocked into the chip by external clock will decide whether the time information has to be read or written:

X X X 1 reads time information from the chip

X X X 0 writes time information into the chip to set the time.

$\overline{\text{Outsel}}$ (6)

If this input is set to logic "0" and if the "READ" mode is selected, the first bit of data will be sent onto the I/O PAD (13) at the negative edge of the 5 th clock pulse. When this input is left open or set to logic "1" the first bit of data is sent onto the I/O PAD (13) at the negative edge of the 4 th pulse.
(See timing diagrams)

Read-out mode

a) Selective data read-out (addresses 0...6)

If the I/P data at the positive edge of the 4th clock-pulses is "1" (for read), the selected time-data is sent from the internal time-counter into the output-shift-registers. The following 8 external negative edges will clock out 8 bit time data at pin 13 organized in two 4-bit BCD-format-words starting with the least-significant bit of the selected time-information. The time information is clocked out serially in synchronism with the negative edge of the external clock. All further clock-pulses are then ignored.

b) Continuous data read-out (address 7)

If the I/P data at the positive edge of the 4th clock-pulses is "1" (for read) and the address 7 was selected, all available time-information (from second to year) will be sent out with the following 56 external negative edges of the clock, i.e. in synchronism with the negative edge of the external clock. Data-transfer is then available in an 8 bit by 7 word serial format. The first bit of each time-data is again the least-significant bit.

If chip-select is kept at a low level (activated), the time data will be available every second, i.e. new timing-information is available once a second.
Interruption of dataflow is accomplished by returning chip-select to a high level state.

Write-in mode

a) Selective data write-in (addresses 0...6)

If the I/P data at the positive edge of the 4th clock-pulses is "0" (for write), time-information can be clocked into the chip via pin 13 by means of external clock-pulses at the clock I/P pin 15.

The following 8 external clock-pulses will clock the new timing information into the internal I/O shift register; all further clock-pulses are then ignored until chip-select goes high and then low again.
The first bit clocked into the chip, should be the most significant bit. The time-information is transfered from the internal I/O shift-register into the time-counting circuit with the rising edge of the chip-select signal; this defines the exact time at which the time information is updated.

b) Continuous data write-in (address 7)

This mode is similar to addresses 0...6, except that all 8 bit by 7 word time information is clocked into the chip in a predefined sequence; thus 60 clock-pulses are required for a complete cycle after chip-select goes low. Every 8 bit-word should be available in the format that the most significant bit is clocked in first. In this mode it is not necessary to return chip-select to a high level in order to write the information into the chip. This information is written into the chip only once for each low going chip-select-signal.

$\overline{\text{Chip-select}}$ (9).

Normally chip-select is high.

Chip-select going low enables the data I/O line, and allows all data to be clocked in synchronism with the external clock.

In the selective data write-in mode, the high going chip-select edge transfers the data from the internal I/O shift register into the time-counting network, thus specifying the exact time at which the time information is updated.

WHENEVER CHIP SELECT GOES HIGH THE I/O LINE (13) GOES INTO A HIGH IMPEDANCE STATE NEITHER ACCEPTING NOR GIVING OUT DATA.

Clock-In (15)

The positive going edge of the clock-signal controls address and data input. The negative going edge controls data output. The clock is internally gated by the chip-select-signal (9). Clock-pulses may be continuously connected to the chip without shifting any data in or out when chip-select is at a high level.

DESCRIPTION

This C-MOS circuit may be used as a real time clock or as an absolute time-counter with battery back up and an external 32 KHz quartz as time-refence. The information transfer takes place serially on a 1-bit bi-directional I/O data line in synchronism with an external clock. Data-exchange is controlled by the chip-select-signal. The internal time-counting circuit operates at a minimum voltage of 1.5 V. When a second supply of 5 V is connected to another input terminal, all outputs are then available. All of the outputs are capable of driving one T.T.L. unit load. The data I/O line is tri-state to enable it to work in microprocessor type environments.

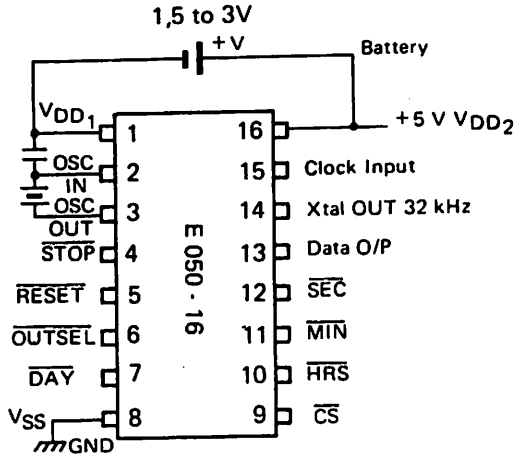
The data are in BCD format words.

FEATURES

- internal time-counter requires one external battery-cell only. Time is still running, even when supply-voltage is removed from circuit
- All of the outputs lines can drive one T.T.L. unit load with 5 V supply connected
- all input/time setting and output/time reading is performed through a single I/O-line for minimum wiring costs
- accepts low impedance 32 KHz-xtals as time reference
- choice of: a) read/write
  b) selected time information
  c) continuous reading of all time-data
- may be used in all µP-controlled or conventional electronic equipment, such as office calculators, electronic typewriters, industrial machine-control and TV-equipment or others
- counts seconds, minutes, hours, date of the month, day of the week, month and year
  every 4th year, February has 29 days
- available in a single 16 pin dual in-line package
- to write or read time-information, an external clock frequency of up to 150 KHz may be used.
- independent stop-input for external control of time-counting, for use as relative time-counter.
- may also be used as a preset time-counter, 24 hours by use of the time pulse outputs.
- easy connection to serial-in LCD-drivers to read out all time-data
- independent output-pins for second/minute/hour/day time-pulses
- Xtal frequency may be used to clock the data I/O and external circuits.

PACKAGE

16 Pin Dual in Line



Pin Connections and functions

1. V_{DD1} (Battery back up negative terminal)
2. OSC IN
3. OSC OUT
4. $\overline{STOP}$ (negative going I/P stop internal counting)
5. $\overline{RESET}$ (negative going I/P to reset)
6. $\overline{OUTSEL}$ (negative going I/P to read)
7. $\overline{DAY}$ (negative pulse every day)
8. VSS (GND)
9. $\overline{CHIP\ SELECT}$ (negative going I/P to start READ or WRITE FUNCTION)
10. $\overline{HRS}$ (negative pulse every hour)
11. $\overline{MIN}$ (negative pulse every minute)
12. $\overline{SEC}$ (negative pulse every second)
13. DATA IN/OUT
14. XTAL OUT 32 kHz
15. CLOCK (external frequency input)
16. VDD2 (positive supply + 5V)

Address information will be clocked into the chip via pin 13 by the first 3 clock-pulses after chip-select goes low. At the fourth clock-pulses a decision is made whether the selected time-information should be read or written. The fifth and following clock-pulses will then clock-in or clock-out the time data. During selective read-write modes, the thirteenth and following clock-pulses are ignored until the next chip-select high-low excursion. During continuous read-write modes the 61st and following clock-pulses are also ignored until the next chip-select high-low excursion.

Power-on-reset

Upon connection of a battery between pins 1 and 16, an internal power-on-reset signal resets all counters to a specified condition. The O/P at the I/O pin (13) remains at a logic "1" until the chip-select pin (9) is taken low and then high again; after which the next chip-select can be used in the normal manner. This continuous logic "1" at the I/O pin indicates battery voltage has been removed at some time and that since this time the chip has never been accessed. This condition is a non-realistic time read out and can easily be detected by a µP as a power turn off signal.
Time counting commences as soon as the xtal starts oscillating.

Battery Operation (1,16)

A single cell battery must be connected between pins 1 and 16 of the package. This voltage keeps the internal time-counter running even when the 5 V supply is removed. The battery may be replaced by a chargeable cell and appropriate current limiting circuitry and charging of the battery can be accomplished by the 5 V supply.

$\overline{\text{Stop}}$ Input (4)

The circuit E 050/16 may also be used as a time-counter with start/stop operation.

When pin (4) is connected to logic "1" or left open circuit, the circuit will count time. When pin 4 is connected to logic "0" the circuit will stop counting time but will retain the most recent timing information.

$\overline{\text{Reset}}$-Input (5)

Reset of the internal time-counting is performed by a negative-going pulse at pin 5. When this pin is left open an internal pull-up resistor keeps the circuit in the counting mode. Reset to the individual registers is as shown:

00 second
00 minute
00 hour
01 date
01 month
01 day of the week
00 year

The WIDTH of the reset pulse should be not less than 14 uS. Time counting will remain static WHILST the reset input is low.

Time-Pulses Output (7, 10, 11, 12)

When $\overline{\text{Stop}}$-input (4) is left open or connected to a logic "1" continuous output timing-pulses are typically 32 µs wide and may be used to clock external circuitry every second, minute, hour or day, depending on the output-pin.

When address 7 is selected (continuous data), the pulse-length is automatically changed, it then becomes 56 clock-pulses long. The timepulse may then be used to gate or strobe the data from pin 13 into other external circuitry, i.e. display-drivers with serial data input. Timing-pulses will not be available, when $\overline{\text{stop}}$ (4) is at a low level. In order to return to 32 µs wide output pulses one of the selective data readouts must be performed.

32 KHz Output (14)

This output is connected via an internal driver directly from the xtal-oscillator.

This squareware may be used to clock external C.M.O.S. circuitry, i.e. serial-input display-drivers connected to the data-output pin. Pin 14 may be connected to clock-in (15) in order to clock the internal address and data I/O. Pin 14 may also be used to measure the xtal-frequency without loading the oscillator.

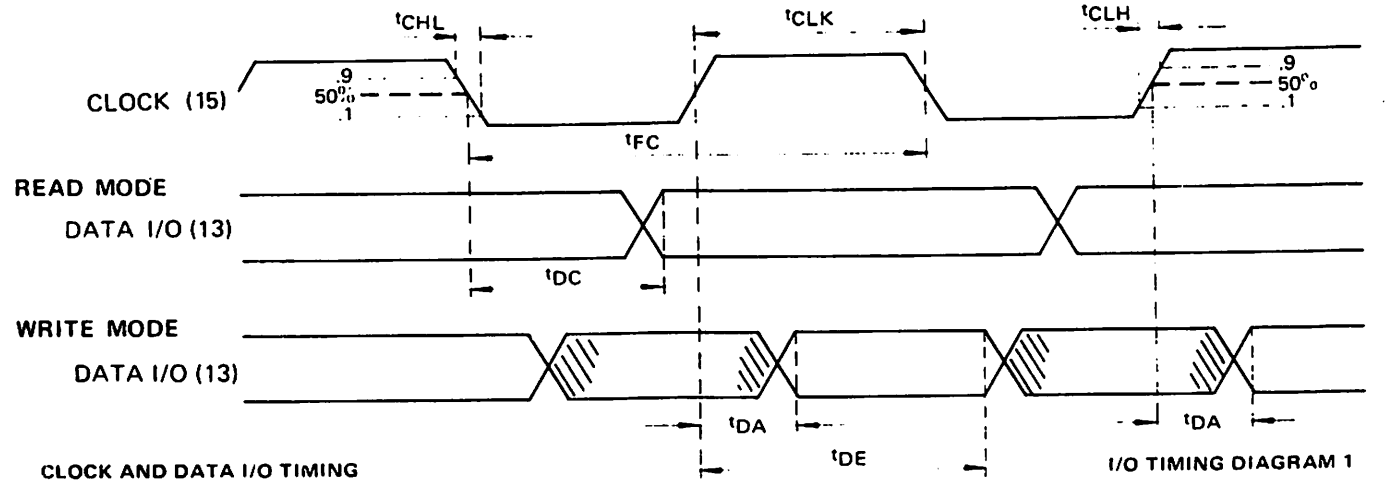
Supply-Voltage (16)

A supply-voltage of 5 V may be supplied at pin 16, in order to input or output time information. If this voltage is removed the internal time-counting still goes on when pin 16 has a battery-cell connected and when using the xtal time-base.

When this voltage is removed, all input- and output-levels (except the data I/O line) remain at the level of the +Ve line through internal pull-up resistors. Thus, no zero going time output-pulses are generated. The data I/O line is in a high impedance state without the 5 V supply connected.

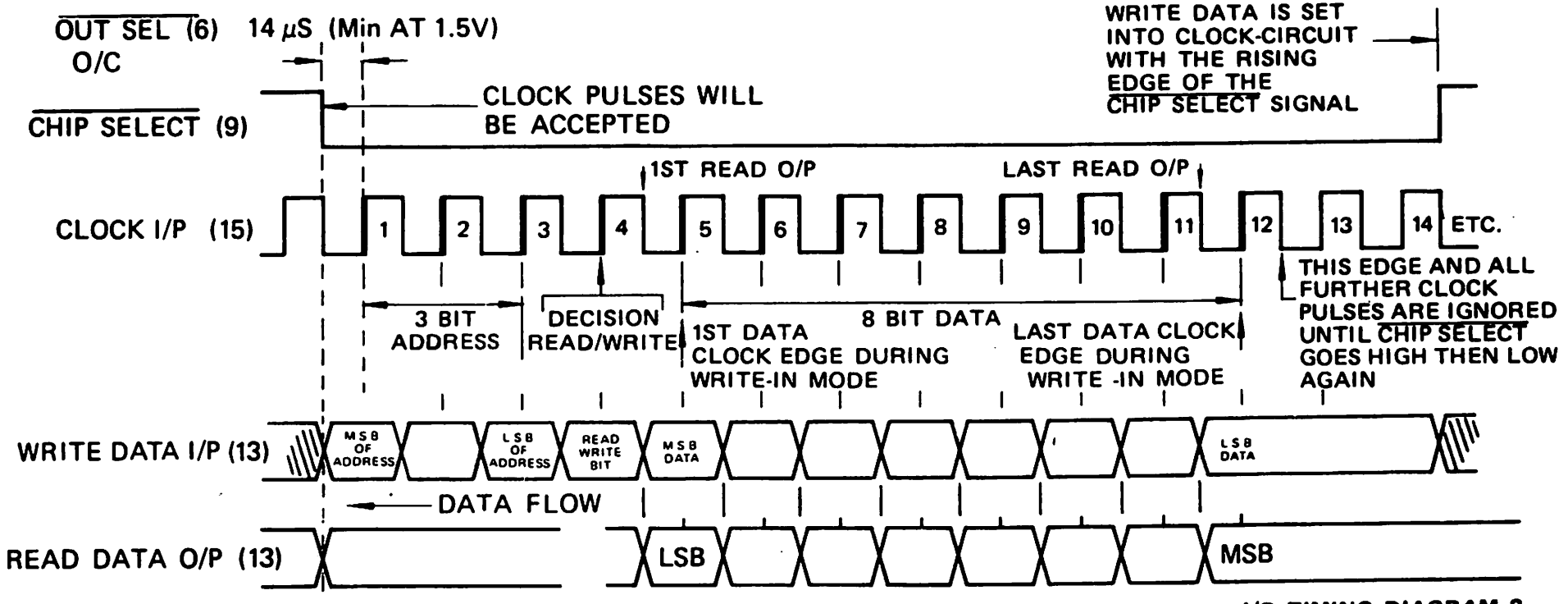
Xtal (2+3) Quartz: f = 32 768 Hz
Rquartz = 40 kΩ typ, 60 kΩ max.
C_O = 1.5 pF
C_I = 1.7 fF

Frequency adjustment = ± 10 ppm Trimmer: C_T : 3 to 12 pF



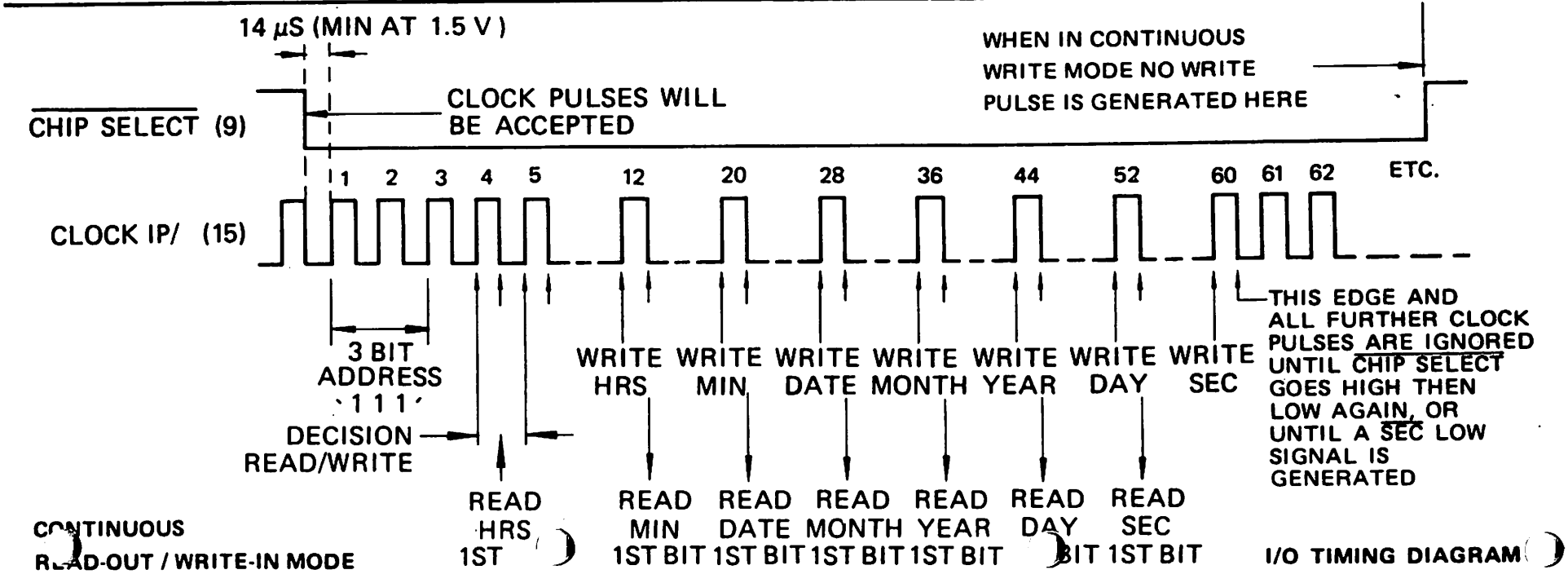
CLOCK AND DATA I/O TIMING

I/O TIMING DIAGRAM 1



SINGLE READ-OUT/WRITE-IN MODE

I/O TIMING DIAGRAM 2



CONTINUOUS READ-OUT / WRITE-IN MODE

I/O TIMING DIAGRAM

Programm Beispiel:

Die im Folgendem aufgeführten Programmbeispiele sollen die prinzipielle Programmierung des EPC-Moduls verdeutlichen. Besondere Beachtung sind dabei den Initialisierungsroutinen, die die prinzipielle Programmierung der I/O-Bausteine zeigt, zu schenken:

```
CP/M MACRO ASSEM 2.0    £001    EPC-Modul utility subroutines

                        MACLIB Z80
 0100                   ORG 100H
 0011 =         XON     EQU 11H

                ; PORTADRESS EQUATES EPC-MODUL
                ; ============================
                ;
                ;       CHIP BASE
                ;       =========

 0000 =         P$STI           EQU     00H     ; STI 00H - 0FH
 0010 =         P$CENT          EQU     10H
 0014 =         P$SIO           EQU     14H
 0018 =         P$BANK          EQU     18H
 001C =         P$765$CMD       EQU     1CH
 001D =         P$765$DAT       EQU     1DH

                ;       DART CHANNELS:
                ;       ==============
 0014 =         P$SIOA$DAT              EQU     P$SIO+0H        ;DART A DATA
 0015 =         P$SIOB$DAT              EQU     P$SIO+1H        ;DART B DATA
 0016 =         P$SIOA$CTR              EQU     P$SIO+2H        ;DART A CTR.
 0017 =         P$SIOB$CTR              EQU     P$SIO+3H        ;DART B CTR.

                ;       STI CHANNELS:
                ;       =============
 0000 =         P$IDR           EQU     P$STI+0         ;INDIRECT DATA REG
 0001 =         P$GPIP          EQU     P$STI+1         ;GENERAL PURPOSE I/O LINES
 0002 =         P$IPRB          EQU     P$STI+2         ;INT PENDING REG B
 0003 =         P$IPRA          EQU     P$STI+3         ;INT PENDING REG A
 0004 =         P$ISRB          EQU     P$STI+4         ;INT SERVICE REG B
 0005 =         P$ISRA          EQU     P$STI+5         ;INT SERVICE REG A
 0006 =         P$IMRB          EQU     P$STI+6         ;INT    MASK REG B
 0007 =         P$IMRA          EQU     P$STI+7         ;INT    MASK REG A
 0008 =         P$PVR           EQU     P$STI+8         ;POINTER/VECTOR REG
 0009 =         P$TABCR         EQU     P$STI+9         ;TIMER A&B CTR REG
 000A =         P$TBDR          EQU     P$STI+10        ;TIMER B DATA
 000B =         P$TADR          EQU     P$STI+11        ;TIMER A DATA
 000C =         P$UCR           EQU     P$STI+12        ;USART CTR
 000D =         P$RSR           EQU     P$STI+13        ;RECEIVER STATUS
 000E =         P$TSR           EQU     P$STI+14        ;TRANSMITTER STATUS
 000F =         P$UDR           EQU     P$STI+15        ;USART DATA
```

```
                ;************************************
                ;*       INIT TABLE DART            *
                ;************************************

                ;       * INIT DART'S *
                        ; DART A
                INIDART:
0100 0E16               MVI C,P$SIOA$CTR
0102 0607               MVI B,7
0104 211001             LXI H,V24ATAB
                        OUTIR
0107+EDB3               DB      0EDH,0B3H

                        ; DARTB
0109 0E17               MVI C,P$SIOB$CTR
010B 0607               MVI B,7
                        OUTIR
010D+EDB3               DB      0EDH,0B3H
010F C9                 RET

                V24ATAB:
0110 18                 DB 18H          ; CHANNEL RESET
0111 03C1               DB 3,0C1H       ; WR3: RXENABLE, 8 BITS
0113 0444               DB 4,44H        ; WR4: X16 CLOCK MODE, 1 STOP, NO PARITY
0115 05                 DB 5
0116 6A         AWR5:   DB 06AH         ; WR5: 8 BITS, TXENABLE, RTS
                                        ; DTR = 0 -> OUTPUT=1 -> WATCH CS = 1
                V24BTAB:
0117 18                 DB 18H
0118 0341               DB 3,041H       ; WR3: RXENABLE, 7 BITS
011A 044F               DB 4,04FH       ; WR4: X16 CLOCK MODE, 2 STOP, PARITY EVEN
011C 05                 DB 5
011D AA         BWR5:   DB 0AAH         ; WR5: RTS, 7 BITS, TXENABLE
                                        ; DTR = 1 => OUTP. = 0 -> WATCH CLOCK = LOW

                ;*************** RS 232 A ******************
                ;INPUT/OUTPUT A CHARACTER VIA DART CHANNEL A
                ;*******************************************

                V24A$OUST:

011E DB16               IN P$SIOA$CTR                   ; TEST OUTPUT STATE
0120 E604C8F6FF         ANI 04H ! RZ ! ORI -1 ! RET     ; RET Z=1 IF NOT READY

                V24A$INST:

0126 DB16               IN P$SIOA$CTR                   ; TEST INPUT STATE
0128 E601C8F6FF         ANI 1 ! RZ ! ORI -1 ! RET       ; RET Z=1 IF NOT READY

                V24A$IN:

012E CD2601             CALL V24A$INST                  ; TEST INPUT STATA
                        JRZ V24A$IN                     ; LOOP BACK IF NOT RDY
0131+28FB               DB      28H,V24A$IN-$-1
0133 DB140000           IN P$SIOA$DAT ! NOP ! NOP       ; ROOM TO CLEAR PARITY
0137 C9                 RET

                V24A$OUT:

0138 CD1E01             CALL V24A$OUST                  ; TEST OUTPUT STATE
                        JRZ V24A$OUT                    ; WAIT IF BUSY
013B+28FB               DB      28H,V24A$OUT-$-1
013D 79D314C9           MOV A,C ! OUT P$SIOA$DAT ! RET  ; OUTPUT CHARACTER IN <C>
```

```
                ;****** INIT SERIAL TIMER INTERRUPT CONTROLER STI **********
                ;************************************************************

                STIINIT:
0141 216101             LXI H,STITAB                    ;POINTER TO TABLE
0144 0E01060F           MVI C,P$GPIP ! MVI B,15         ;15 DIRECT REGISTERS
                STIDIR:
0148 7E                 MOV A,M ! OUTP A                ;OUT DATA
0149+ED79               DB      0EDH,A*8+41H
014B 230C               INX H ! INR C                   ;NEXT DATA
                        DJNZ STIDIR
014D+10F9               DB      10H,STIDIR-$-1

014F 0608               MVI B,8                         ;8 INDIRECT REGISTERS
                STIINDR:
0151 DB08E6F8           IN P$PVR ! ANI 0F8H
0155 05B004             DCR B ! ORA B ! INR B
0158 D308               OUT P$PVR                       ;INDIRECT ADDRESS TO PVR
015A 7ED300             MOV A,M ! OUT P$IDR             ;CONSTANT IN INDIRECT REG
015D 23                 INX H ! DJNZ STIINDR            ;NEXT INDIRECT REGISTER
015E+10F1               DB      10H,STIINDR-$-1
0160 C9                 RET


                ;************** INIT TABLE STI ****************
                ;**********************************************

                STITAB:

                        ; 15 DIRECT ADRESSABLE REGISTER STARTING WITH GPIP

0161 73                 DB 0111$0011B   ;GPIB BOOT AUS, RESET, TC = 0, WATCH DATA = 1
0162 0000               DB 0,0          ;IPRB, IPRA CLEAR PENDING INTERRUPTS
0164 0000               DB 0,0          ;ISRB, ISRA
0166 0000               DB 0,0          ;IMRB, IMRA
0168 08                 DB 08           ;PVR, SOFTWARE END OF INT
0169 02                 DB 0000$0010B   ;TABCR-TA STOP, TB DELAY PRESC./10
016A 01                 DB 1            ;TBDR->TC= 3.0 MHZ / (2X10X16X9600)
016B 0A                 DB 10           ;TADR -> INPUT = 1 HZ -> MOTO = 10 S
016C 88                 DB 1000$1000B   ;UCR /16; 8 BITS; 1 STOP; NO PARITY ; NON DMA
016D 01                 DB 01           ;RCR RECEIVER ENABLE
016E 05                 DB 0000$0101B   ;TSR TRANSMITTER ENABLE
016F 11                 DB 011H         ;UDR -> XON

                        ; 8 INDIRECT ADR. REGISTERS STARTING WITH TCDR

0170 A2                 DB 1010$0010B   ;TCDCR- DELAY MODE,PRESCALE /10, TA RES.
0171 87                 DB 1000$0111B   ;DDR - IN = BIT 6,5,4,3
0172 60                 DB 0110$0000B   ;IERA TIMER A, ACKN. CENTR.
0173 02                 DB 0000$0010B   ;IERB
0174 00                 DB 0000$0000B   ;AER-ACTIVE EDGE REGISTER ALL FALLING
0175 08                 DB 8            ;TCDR -> TC= 3.0 MHZ / (2X16X10X1200)
0176 01                 DB 1            ;TDDR -> TC= 3.0 MHZ / (2X16X10X9600)
0177 AA                 DB 0AAH         ;SCR-SYNCH CHARACTER


                ; ******* TEST KEYBOARD STATUS **********
                ; SWITCH MOTOR FDC OFF IF INT FROM TIMER
                ; ****************************************

                CSTS:
0178 DB03               IN P$IPRA               ; TIME OUT FROM TIMER A ?
                        BIT 5,A
```

```
017A+CB6F               DB      0CBH,5*8+A+40H
                        JRZ KEY$STAT
017C+280A               DB      28H,KEY$STAT-$-1

                        RES 5,A
017E+CBAF               DB      0CBH,5*8+A+80H
0180 D303               OUT P$IPRA              ; RESET PENDING INTERRUPT
0182 DB09               IN P$TABCR              ; IF TIME OUT STOP TIMER
0184 E60FD309           ANI 0FH ! OUT P$TABCR


                ;********** KEYBOARD ************
                ; INPUT/OUTPUT CHARACTERS TO KEY
                ; USE REVERSE XON-PROTOCOLL
                ;*********************************

                KEY$STAT:                               ; TEST RECEIVER STATE
0188 DB0DE680C8         IN P$RSR ! ANI 80H ! RZ         ; RET WITH Z=1 IF NO CHAR. AVIAI
018D F6FFC9             ORI -1 ! RET                    ; ELSE RET Z=0

                KEYINP:                                 ; INPUT A CHARACTER FORM KEYBOAI
0190 DB0DE6F0           IN P$RSR ! ANI 0F0H             ; RECEIVER STATUS. CHARACTER AV]
                        JRZ KEYINP                      ; NO -> LOOP BACK
0194+28FA               DB      28H,KEYINP-$-1
                        BIT 7,A                         ; BUFFER FULL ?
0196+CB7F               DB      0CBH,7*8+A+40H
                        JRNZ KEYREC                     ; SKIP IF SO
0198+200B               DB      20H,KEYREC-$-1
019A 3E01D30D           MVI A,01H ! OUT P$RSR           ; ELSE ERRORS, RESET IT
019E 0E11CDAF01         MVI C,XON ! CALL KEYOUTP        ; OUTPUT XON TO KEYBOARD
                        JR KEYINP                       ; NEW RETRY
01A3+18EB               DB      18H,KEYINP-$-1

01A5 DB0F47     KEYREC: IN P$UDR ! MOV B,A              ; SAVE CHARACTER
01A8 0E11CDAF01         MVI C,XON ! CALL KEYOUTP        ; SEND XON TO KEY
01AD 78C9               MOV A,B ! RET                   ; RET WITH INPUT CHARACTER IN <A

                KEY$OUTP:                               ; OUTPUT CHARACTER IN <C> TO KE)
01AF DB0E               IN P$TSR ! BIT 7,A              ; TRANSMITTER EMPTY ?
01B1+CB7F               DB      0CBH,7*8+A+40H
                        JRZ KEY$OUTP                    ; NOT EMPTY ? -> LOOP BACK
01B3+28FA               DB      28H,KEY$OUTP-$-1
01B5 79D30FC9           MOV A,C ! OUT P$UDR ! RET       ; EMPTY - OUTPUT CHARACTER


                ; ***************** CENTRONICS ******************
                ; OUTPUT A CHARACTER IN A TO CENTRONICS PRINTER
                ;***********************************************

                CENTOUT:
01B9 3E10D316           MVI A,10H ! OUT P$SIOA$CTR      ; RESET EXT. STATUS CHANGE INT.
01BD CDDD01             CALL CENT$STAT ! JRZ CENT$OUT   ; CENTRONICS READY ?
01C0+28F7               DB      28H,CENT$OUT-$-1
01C2 79D310             MOV A,C! OUT P$CENT             ; SEND CHAR
01C5 DB01               IN P$GPIP ! RES 1,A
01C7+CB8F               DB      0CBH,1*8+A+80H
01C9 CDD601             CALL CENT$DELAY
01CC D301               OUT P$GPIP                      ; GIVE STROBE
                        SETB 1,A
01CE+CBCF               DB      0CBH,1*8+A+0C0H
01D0 CDD601D301         CALL CENTDELAY ! OUT P$GPIP     ; RESET STROBE
01D5 C9                 RET
```

```
                CENT$DELAY:

01D6 C5060A             PUSH B! MVI B,10
                        DJNZ CENTDELAY+3
01D9+10FE               DB      10H,CENTDELAY+3-$-1
01DB C1C9               POP B! RET

                CENT$STAT:

01DD DB16E608           IN P$SIOA$CTR ! ANI 1000B       ; CENTRONICS BUSY ?  -> DCD = L(
01E1 C8                 RZ
01E2 F6FFC9             ORI -1 ! RET                    ; READY IF DCD = 1 -> /DCD = BU


                ;*********************************
                ; READ/WRITE TIME ENTRY POINTS
                ; INTERFACE TO MEM E 050 - 16
                ;*********************************

                RDTIME:

01E5 CD3802             CALL CONTCMD                    ; GIVE CONT. CMD WORD
01E8 CD4B02             CALL CLKTIME                    ; ONE MORE CLOCK => READ DATA

                        ; MAKE DATA PIN = INPUT
01EB DB08E6F8           IN P$PVR ! ANI 0F8H
01EF F606D308           ORI 06H ! OUT P$PVR             ; SELECT IDR 6 => DDR
01F3 DB00               IN P$IDR ! RES 0,A              ; BIT 0 DDR = 0 -> INPUT
01F5+CB87               DB      0CBH,0*8+A+80H
01F7 D300               OUT P$IDR

01F9 CD4B02             CALL CLKTIME                    ; ONE MORE CLOCK 5

                        ; NOW SET UP COUNTERS
01FC 0607               MVI B,7                         ; 7 BYTES TO READ
01FE 213102             LXI H,DATE
                RDTIM0:
0201 0E011600           MVI C,1 ! MVI D,0               ; BYTE PATTERN YET ZERO, START W

                RDTIM1:

0205 DB01               IN P$GPIP ! BIT 0,A             ; TEST FOR ZERO
0207+CB47               DB      0CBH,0*8+A+40H
                        JRZ RDTIM2                      ; SKIP OUT IF SO
0209+2803               DB      28H,RDTIM2-$-1
020B 7AB157             MOV A,D ! ORA C ! MOV D,A       ; ELSE SET BIT POSITION

                RDTIM2:
020E CD4B02             CALL CLKTIME                    ; NEXT BIT
                        RLCR C                          ; NEXT POSITION
0211+CB01               DB      0CBH, 00H + C
                        JRNC RDTIM1                     ; LOOP BACK IF MORE POSITIONS
0213+30F0               DB      30H,RDTIM1-$-1

0215 77                 MOV M,A                         ; SAVE BYTE
0216 23                 INX H                           ; NEXT POSITION
                        DJNZ RDTIM0
0217+10E8               DB      10H,RDTIM0-$-1

                        ; MAKE DATA PIN TO OUTPUT
0219 DB00               IN P$IDR ! SETB 0,A             ; BIT 0 DDR = 1 => OUTPUT
021B+CBC7               DB      0CBH,0*8+A+0C0H
```

```
021D D300               OUT P$IDR

                        ; RESET CS
021F 3E05D316  PREMEM:  MVI A,5 ! OUT P$SIOA$CTR
0223 3A1601             LDA AWR5                        ; GET WR 5 SIO A
                        RES 7,A
0226+CBBF               DB      0CBH,7*8+A+80H
0228 D316               OUT P$SIOA$CTR                  ; RESET CS

                        ; MAKE DATA PIN = 1
022A DB01               IN P$GPIP ! SETB 0,A
022C+CBC7               DB      0CBH,0*8+A+0C0H
022E D301               OUT P$GPIP
0230 C9                 RET

0231           DATE:    DS 7

               CONTCMD:         ; GIVE CONTINUOUS COMMAND TO MEM
                                ; ACTIVATE CS FORM MEM
0238 3E05D316           MVI A,5 ! OUT P$SIOA$CTR        ; SELECT WR5 SIOA
023C 3A1601             LDA AWR5                        ; GET WR5 SIOA
                        SETB 7,A                        ; ACTIVATE DTR -> CS
023F+CBFF               DB      0CBH,7*8+A+0C0H
0241 D316               OUT P$SIOA$CTR                  ; ACTIVATE IT
0243 0603               MVI B,3
               CONTLOP:
0245 CD4B02             CALL CLKTIME                    ; 3 TIMES CLOCKS WHEN DATA = 1
                        DJNZ CONTLOP                    ; -> CONTINUOUS COMMAND
0248+10FB               DB      10H,CONTLOP-$-1
024A C9                 RET


               CLKTIME:         ; GIVE ONE POSITIVE GOING CLOCK PULSE
024B F5C5               PUSH PSW ! PUSH B
024D 0607               MVI B,7                         ; DELAY
               CLKTIM1:DJNZ CLKTIM1
024F+10FE               DB      10H,CLKTIM1-$-1
0251 3E05D317           MVI A,5 ! OUT P$SIOB$CTR        ; SELECT WR5
0255 3A1D01             LDA BWR5 ! RES 7,A
0258+CBBF               DB      0CBH,7*8+A+80H
025A D317               OUT P$SIOB$CTR                  ; CLOCK = HIGH

025C 0607               MVI B,7                         ; DELAY
               CLKTIM2:DJNZ CLKTIM2
025E+10FE               DB      10H,CLKTIM2-$-1
0260 3E05D317           MVI A,5 ! OUT P$SIOB$CTR        ; SELECT WR5
0264 3A1D01             LDA BWR5 ! SETB 7,A
0267+CBFF               DB      0CBH,7*8+A+0C0H
0269 D317               OUT P$SIOB$CTR                  ; CLOCK = LOW
026B C1F1C9             POP B ! POP PSW ! RET


026E CD3802    WRTIME:  CALL CONTCMD
0271 DB01               IN P$GPIP ! RES 0,A             ; SELECT WRITE
0273+CB87               DB      0CBH,0*8+A+80H
0275 D301CD4B02         OUT P$GPIP ! CALL CLKTIME

                        ; SET UP COUNTERS
027A 0607213102         MVI B,7 ! LXI H,DATE            ; 7 BYTES
               WRTIM0:
027F C50608             PUSH B ! MVI B,8
0282 7E                 MOV A,M                         ; 8 BITS, GET BYTE
```



```
0283 07F5       WRTIM1: RLC ! PUSH PSW                  ; MSB INTO CARRY, SAVE PATT.
0285 DB01               IN P$GPIP ! RES 0,A             ; ASSUME BIT IS ZERO
0287+CB87               DB      0CBH,0*8+A+80H
                        JRNC WRTIM2                     ; SKIP IF SO
0289+3002               DB      30H,WRTIM2-$-1
                        SETB 0,A                        ; ELSE SET BIT
028B+CBC7               DB      0CBH,0*8+A+0C0H
028D D301CD4B02WRTIM2: OUT P$GPIP ! CALL CLKTIME ! POP PSW
                        DJNZ WRTIM1                     ; NEXT BIT
0293+10EE               DB      10H,WRTIM1-$-1

0295 C123               POP B ! INX H                   ; NEXT BYTE
                        DJNZ WRTIM0
0297+10E6               DB      10H,WRTIM0-$-1
0299 C31F02             JMP PREMEM



                ;*****************************************************
                ;FLOPPY DISK INTERFACE ROUTINES USING UPD 765 CTRL.
                ;*****************************************************


                        ;INIT UPD 765
                INI765:
029C 0600               MVI B,0
                        DJNZ INI765+2                   ; DELAY
029E+10FE               DB      10H,INI765+2-$-1
02A0 DB1CFE80           IN P$765$CMD ! CPI 80H          ; REQUEST FOR MASTER ?
                        JRZ SPECIFY                     ; NO -> SKIP
02A4+2804               DB      28H,SPECIFY-$-1
02A6 DB1D               IN P$765$DAT                    ; ELSE READ DATA
                        JR INI765                       ; AND TRY IT AGAIN
02A8+18F2               DB      18H,INI765-$-1

                SPECIFY:
02AA 21CC02             LXI H,SPEC5-1                   ; SPECIFY PARAMETER TEAC FD55-B
02AD 010303             LXI B,0303H                     ; SPECIFY CMD. 3 BYTES
02B0 CD2A03             CALL CMDFD1                     ; SEND COMMAND

                        ; RECALIBRATE DRIVE TO TRACK ZERO

                RECAL:
02B3 010702             LXI B,0207H                     ; RECAL.CMD =07 - 2 BYTES
02B6 CDF802             CALL MOTO

                        ; SENSE DRIVE INTERUPT STATUS

02B9 010801     SENSE:  LXI B,0108H                     ; COMMAND SENSED.=08 - 1 BYTE
02BC CD2703             CALL    CMDFD
02BF CD3A03             CALL    NEXT                    ; FETCH RESULT
02C2 47                 MOV     B,A                     ; SAVE IT
02C3 FE80               CPI     80H                     ; INVALID CMD.?
02C5 C43A03             CNZ     NEXT                    ; NO -> NEXT RESULT
                        BIT     5,B                     ; SEEK FINISHED ?
02C8+CB68               DB      0CBH,5*8+B+40H
                        JRZ     SENSE                   ; NO -> LOOP BACK
02CA+28ED               DB      28H,SENSE-$-1
02CC C9                 RET                             ; YES -> ALL DONE
```

```
                ;*********** SPECIFY PARAMETER FOR TEAC FD-55B *****************
                ;*****************************************************************

02CD DF         SPEC5   DB      1101$1111B
02CE 13                 DB      0001$0011B      ;SPECIFY PARAM. 5 1/4 TEAC B
                                                ;HLT = 36 MS
                                                ;HUT = 480 MS
                                                ;SRT = 6 MS
                                                ;NON-DMA MODE

                ;********** COMMAND TABLE FOR UPD 765 ***************
                ;****************************************************

02CF 46         CMDTAB: DB      046H    ; MFM READ COMMAND
02D0 00         UNIT:   DB      0       ; HEAD0/UNIT A
02D1 00         TRACK:  DB      0       ; TRACK 0
02D2 00         HEAD:   DB      0       ; HEAD 0
02D3 01         SECTOR: DB      1       ; SECTOR 1
02D4 02         N:      DB      2       ; N=2 512BYTES
02D5 0A         EOT:    DB      10      ; EOT, 10 SECTORS PER TRACK
02D6 10         GPL:    DB      10H     ; GPL GAP LENGTH
02D7 FF         DTL:    DB      -1      ; DTL DATA TERMINAL LENGTH

02D8            RESLTB: DS 7            ; RESULT TABLE FOR FDC
02DF            CMDCODE:DS 1            ; COMMAND CODE FOR 765
02E0            ZDMA:   DS 2            ; CURRENT DMA ADRESS



                ;*********** SENSE DRIVE STATUS ***************
                ;**********************************************

                SDRS:
02E2 010402             LXI     B,0204H         ; SDRS.CMD = 04 - 2 BYTES
02E5 CD2703             CALL    CMDFD
02E8 CD3A03             CALL    NEXT
02EB C9                 RET

                ;************* SEEK ****************
                ; SEEK TO TRACK IN A, DRIVE IN UNIT
                ;***********************************

                SEEK:
02EC 32D102             STA TRACK
02EF 010F03             LXI B,030FH             ; COMMAND SEEK = 0F - 2 BYTES
02F2 CDF802             CALL MOTO
02F5 C3B902             JMP SENSE


                ;****************** MOTOR ON ***************************
                ; WAIT UNTIL DISK READY, THEN TRANSMIT CMD. IN BC TO FDD
                ;*******************************************************

02F8 C5         MOTO:   PUSH    B               ; SAVE CMD.
                MOTO1:
                        ; STOP TIMER A
02F9 DB09               IN P$TABCR              ; GET TIMER AB CTR
02FB E60F               ANI 0FH                 ; STOP TIMER A
02FD D309               OUT P$TABCR

                        ; RESET TIMER A
```

```
02FF DB08               IN P$PVR
0301 F607D308           ORI 07H ! OUT P$PVR     ; ACCESS IDR NR. 7
0305 DB00               IN P$IDR                ; GET TCDCTR
                        SETB 7,A
0307+CBFF               DB      0CBH,7*8+A+0C0H
0309 D300               OUT P$IDR               ; RESET TIMER A -> MOTOR GOES ON
                        RES 7,A
030B+CBBF               DB      0CBH,7*8+A+80H
030D D300               OUT P$IDR               ; RUN TIMER A

                        ; SET TIMER A
030F 3E0A               MVI A,10                ; 10 SECONDS FOR MOTOR ON
0311 D30B               OUT P$TADR

                        ; ENABLE TIMER A
0313 DB09               IN P$TABCR              ; GET TIMER AB CTR. REGISTER
0315 F680               ORI 80H                 ; SET TIMER A TO EVENT COUNT MODE
0317 D309               OUT P$TABCR

                        ; RESET PENDING INTERRUPT FROM TIMER A
0319 DB03               IN P$IPRA               ; PENDING INT. REG A
                        RES 5,A
031B+CBAF               DB      0CBH,5*8+A+80H
031D D303               OUT P$IPRA

031F CDE202             CALL SDRS ! BIT 5,A     ; DRIVE READY ?
0322+CB6F               DB      0CBH,5*8+A+40H
                        JRZ MOT01               ; NO->WAIT
0324+28D3               DB      28H,MOT01-$-1
0326 C1                 POP B

                ; ************ COMMAND TO UPD 765 **************
                ; TRANSMITT COMAND IN BC TO FLOPPY DISK CONTROLER
                ; B=NUMBER OF BYTES TO TRANSMITT, C = COMMAND
                ; ********************************************

                CMDFD:
0327 21CF02             LXI H,CMDTAB

                CMDFD1:
032A DB1C               IN P$765$CMD
032C E6C0FE80           ANI 0C0H! CPI 80H
                        JRNZ CMDFD1             ;REQUEST FOR MASTER ?
0330+20F8               DB      20H,CMDFD1-$-1
0332 79D31D             MOV A,C! OUT P$765$DAT  ;SEND COMMAND
0335 234E               INX H ! MOV C,M
                        DJNZ CMDFD1             ;NEXT COMMAND
0337+10F1               DB      10H,CMDFD1-$-1
0339 C9                 RET


                ;********** NEXT RESULT ************
                ; READ NEXT RESULT BYTE FROM FDC
                ;************************************

                NEXT:
033A DB1C               IN P$765$CMD
033C E6C0FEC0           ANI 0C0H! CPI 0C0H
                        JRNZ NEXT               ; REQUEST FOR MASTER ?
0340+20F8               DB      20H,NEXT-$-1
0342 DB1D               IN P$765$DAT
0344 C9                 RET
```



```
                ;********* FETCH RESULTS FROM FDC ************
                ;       CHECK FOR R/W ERRORS, IF ANY Z=0
                ;*****************************************

                RESULT:
0345 0606               MVI     B,6     ; 7 RESULTS
0347 CD3A03             CALL    NEXT    ; FETCH IT
034A 21D802             LXI     H,RESLTB
034D 77                 MOV     M,A     ; STORE IT
034E E6C0               ANI     0C0H    ; ANY ERROR IN STATUS 0
0350 4F                 MOV     C,A
                RESLOP:
0351 CD3A03             CALL    NEXT
0354 23                 INX     H
0355 77                 MOV     M,A
                        DJNZ    RESLOP
0356+10F9               DB      10H,RESLOP-$-1
0358 79                 MOV     A,C     ; ERROR REPORT STATUS 0
0359 B7                 ORA     A
035A C9                 RET             ; RET WITH ERROR IN A


                ; ************************************************
                ; ****** SIMPLE READ & WRITE ENTRY POINTS *********
                ; R/W IN NON DMA MODE A 512 BYTE SECTOR
                ; R/W ONE SECTOR IN <SECTOR>,
                ; R/W TRACK IN <TRACK>, SEEK-COMMAND GIVEN
                ; R/W HEAD IN <HEAD> AND <UNIT>
                ; R/W DRIVE NUMBER IN <UNIT>
                ; DESTINATION/SOURCE IN <ZDMA>
                ; ************************************************************

                ;******* WRITE SELECTED SECTOR TO FDD *******
                ;*******************************************

                WRITEIT:
035B 11                 DB 11H
                        OUTI            ; WRITE DATA
035C+EDA3               DB      0EDH,0A3H
035E 3E05               MVI A,05H       ; 765 WRITE COMMAND IN A
                        JR RWIT
0360+1805               DB      18H,RWIT-$-1


                ;******* READ SELECTED RECORD FROM FDD ******
                ;*******************************************

                READIT:
0362 11                 DB 11H
                        INI             ; READ DATA
0363+EDA2               DB      0EDH,0A2H
0365 3E06               MVI A,6         ; 765 READ COMMAND IN A

                RWIT:
0367 32DF02             STA CMDCODE             ; COMAND CODE FOR READ/WRITE
                        SDED INIMFM1            ; SET INI / OUTI OP-CODES
036A+ED53               DB      0EDH,53H
036C+C003               DW      INIMFM1
                        SDED INIMFM2
036E+ED53               DB      0EDH,53H
0370+CD03               DW      INIMFM2
```

```
                RWLOP:
0372 060A               MVI B,10                ; RETRY-COUNTER
0374 C5         RWOP:   PUSH B
0375 F3                 DI
0376 119E03D5           LXI D,TERMI ! PUSH D    ; SET TERMI ONTO STACK
                        LXIX RW765M             ; START ADRESS
037A+DD21               DB      0DDH,21H
037C+B603               DW      RW765M
                        LXIY ENDRW              ; RETURN ADRESS
037E+FD21               DB      0FDH,21H
0380+9703               DW      ENDRW
0382 0609               MVI B,9                 ; 9 COMMANDS
0384 3ADF02             LDA CMDCODE
0387 4F                 MOV C,A                 ; COMMAND IN C
0388 3ACF02B14F         LDA CMDTAB ! ORA C ! MOV C,A    ; COMMAND READ/WRITE IN C
038D CDF802             CALL MOTO               ; TRANSFER COMMAND

0390 2AE002             LHLD ZDMA               ; DMA ADRESS IN <HL>
0393 0E1D               MVI C,P$765$DAT

                        PCIX                    ; JUMP TO READ/WRITE 765
0395+DDE9               DB      0DDH,0E9H

                ENDRW:
0397 DB01F604           IN P$GPIP ! ORI 0100B
039B D301               OUT P$GPIP              ; TERMINATE COUNT
039D D1                 POP D                   ; RESET STACK ADRESS IF NOT USED

                TERMI:
039E DB01F604           IN P$GPIP ! ORI 0100B   ; ENTRY POINT TO ABORT R/W
03A2 D301               OUT P$GPIP              ; GIVE TERMINATE COUNT
                        RES 2,A
03A4+CB97               DB      0CBH,2*8+A+80H
03A6 000000             NOP! NOP! NOP           ; DELAY
03A9 D301               OUT P$GPIP              ; RESET TC
03AB FB                 EI                      ; ENABLE INT.

03AC CD4503             CALL RESULT             ; ENTER RESULT PHASE
03AF C1C8               POP B ! RZ              ; NO ERRORS -> RET

                        DJNZ RWOP               ; TRY IT AGAIN IF ERRORS
03B1+10C1               DB      10H,RWOP-$-1
03B3 3E01C9             MVI A,1 ! RET           ; UNRECOVERABLE ERROR


                ;****************************************
                ; PHYSICAL READ/WRITE ENTRYS NON-DMA MODE
                ;****************************************

                ;*******************************************
                RW765M: ; HL = DESTINATION, C = PORT,
                ;*******************************************

03B6 0600               MVI B,0         ; 256 BYTES
                RWMFM:
03B8 DB1C               IN P$765$CMD
03BA 07                 RLC             ; BIT 7 IN CARRY
                        JRNC RWMFM      ; REQUEST FOR MASTER ?
03BB+30FB               DB      30H,RWMFM-$-1
03BD 0707               RLC ! RLC
03BF D0                 RNC             ; EXECUTION MODE ?
                INIMFM1:
                        INI             ; INI IF READ DATA, OUTI IF WRITE DATA
```

```
03C0+EDA2               DB      0EDH,0A2H
03C2 C2B803             JNZ RWMFM       ; REPEAT TIL DONE

                RWMFM2:
03C5 DB1C               IN P$765$CMD    ; NEXT 256 BYTES, REG B IS ZERO
03C7 07                 RLC             ; BIT 7 IN CARRY
                        JRNC RWMFM2     ; REQUEST FOR MASTER ?
03C8+30FB               DB      30H,RWMFM2-$-1
03CA 0707               RLC ! RLC
03CC D0                 RNC             ; EXECUTION MODE ?
                INIMFM2:
                        INI
03CD+EDA2               DB      0EDH,0A2H
03CF C2C503             JNZ RWMFM2      ; REPEAT TIL DONE
                        PCIY            ; JUMP TO ENDRW
03D2+FDE9               DB      0FDH,0E9H

03D4                    END
```

IC's

IC 00,01	74 LS 245
IC 02	2716/32
IC 03,04	74 LS 245
IC 05	74 LS 194
IC 06	24 S 10 (BU/0)
IC 07	Z80 B CPU
IC 08	24 SA 10 (IO/0)
IC 09	74 S 158
IC 10,11	4164, 150 ns
IC 12	74 S 158
IC 13	24 SA 10 (ME/0)
IC 14	74 LS 173
IC 15	74 LS 367
IC 16	4929
IC 17	74 LS 626
IC 18	74 LS 173
IC 19	74 LS 14
IC 20	MEM E050-16
IC 21	74 LS 158
IC 22	74 LS 139
IC 23	FDC 9229 BT
IC 24	upD 765
IC 25	Z80 B DART
IC 26	Z80 A STI
IC 27-32	4164, 150 ns
IC 33	ICL 7660
IC 34	74 LS 374
IC 35	MC 1489
IC 36	MC 1488
IC 37	74 06
IC 38	74 07

Dioden

D 00-02	AA 143
D 03	1 N 4148
D 04	AA 143
D 05	1 N 4148
D 06	Z 2,7
D 07	1 N 4148

Netzwerke

SIL 0	7 x 4,7 K
SIL 1	7 x 680

Widerstände

R 00	2,2 K
R 01	680
R 02	2,2 K
R 03	82 K
R 04	47 K
R 05	22
R 06	680
R 07-10	4,7 K
R 11-15	330
R 16,17	4,7 K
R 18	1,2 K
R 19	4,7 K

Transistor

T 00	2 N 2907 A

Kondensatoren

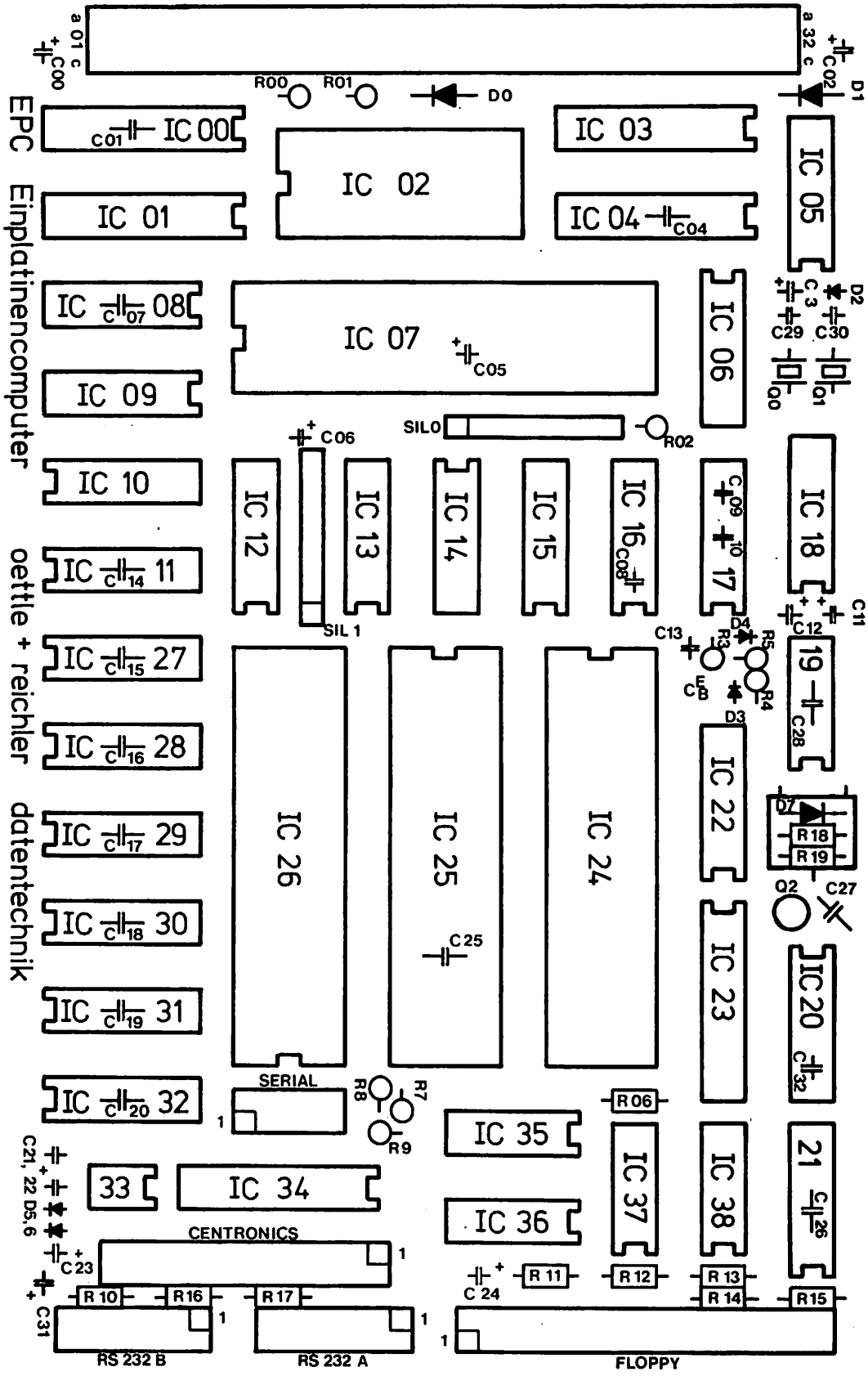
C 00	4,7 uF, 10V
C 01	47 nF
C 02,03	4,7 uF, 10V
C 04,05	47 nF
C 06	4,7 uF, 10V
C 07	47 nF
C 08	470 pF
C 09,10	entfaellt
C 11	4,7 uF, 10V
C 12	10 uF, 6,3V
C 13	12 pF
C 14-20	47 nF
C 21-23	10 uF, 16V
C 24	4,7 uF, 10V
C 25,26	47 nF
C 27	3-12 pF Trimmer
C 28	47 nF
C 29,30	12 pF
C 31	4,7 uF, 10V
C 32	470 pF

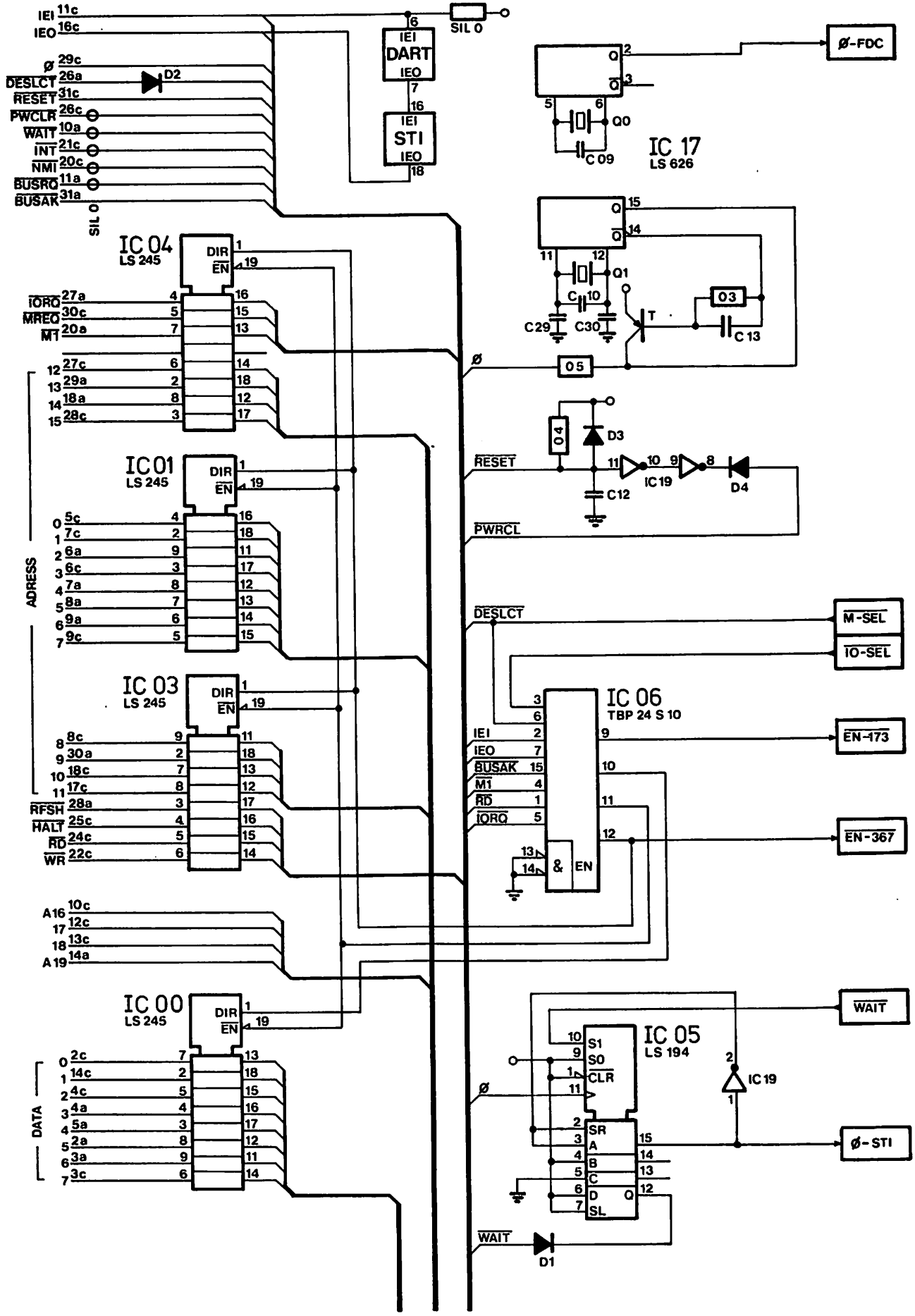
Quarze

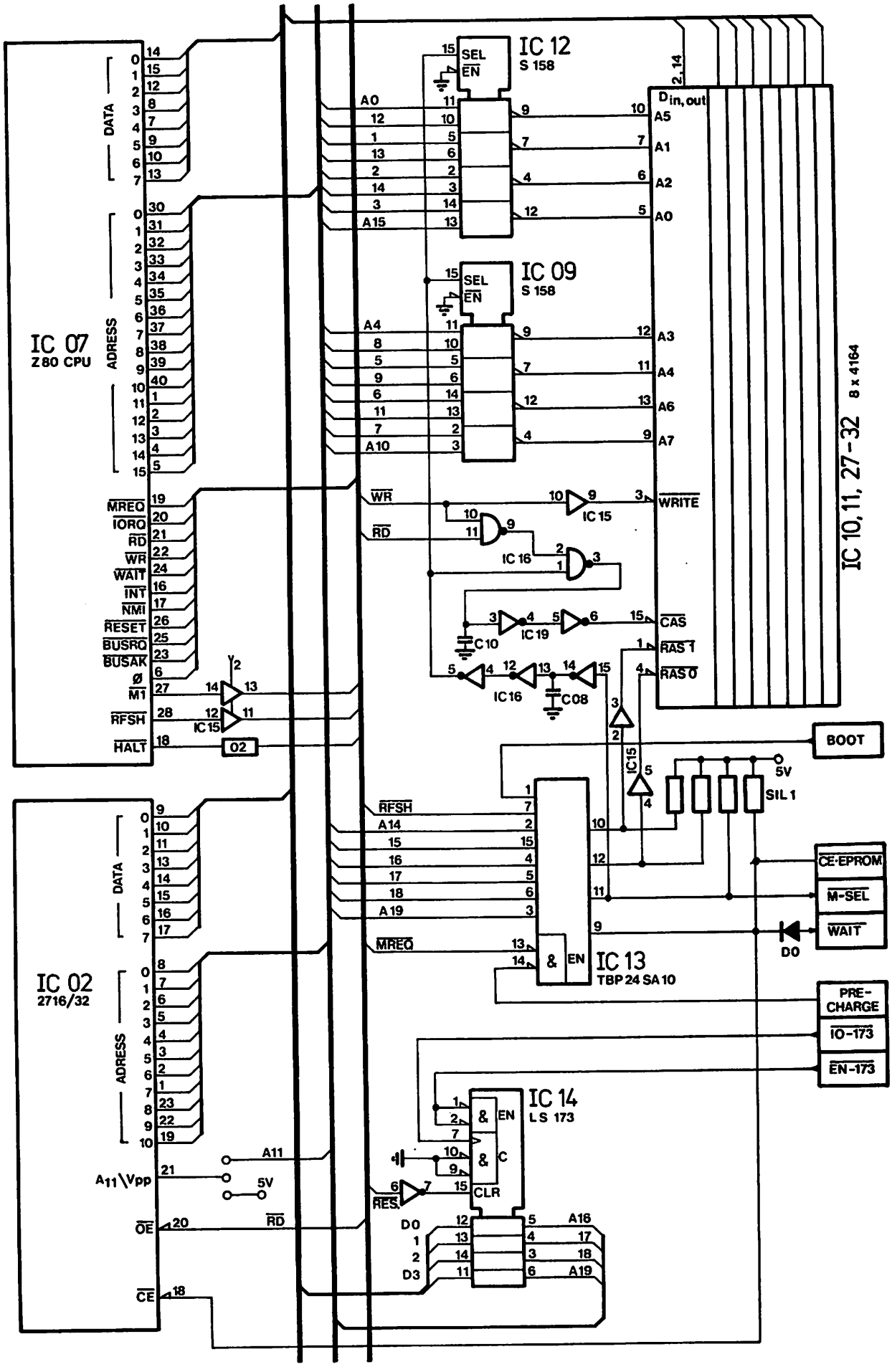
Q0	16 MHz
Q1	6 MHz
Q2	32,768 KHz

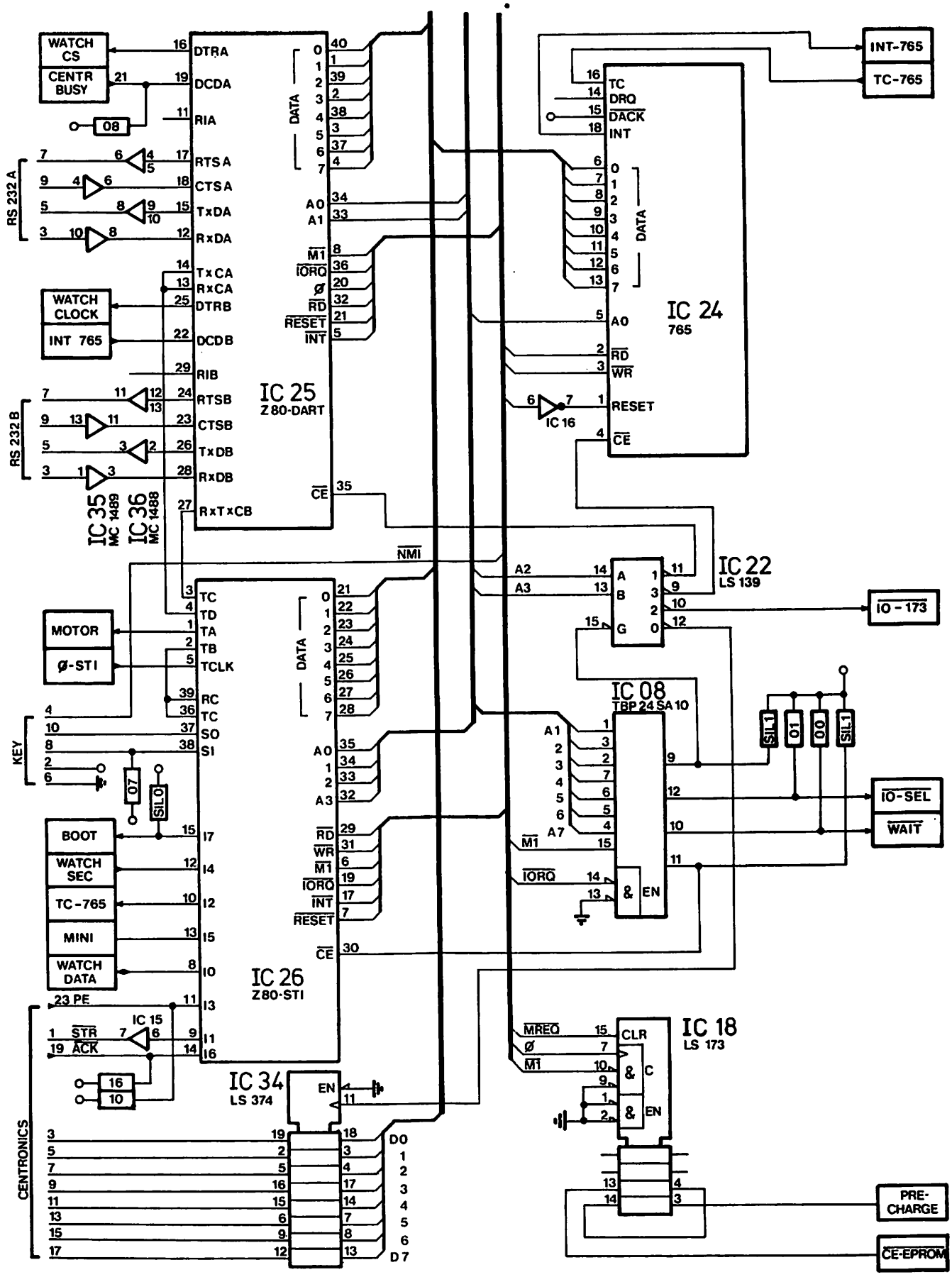
Akku

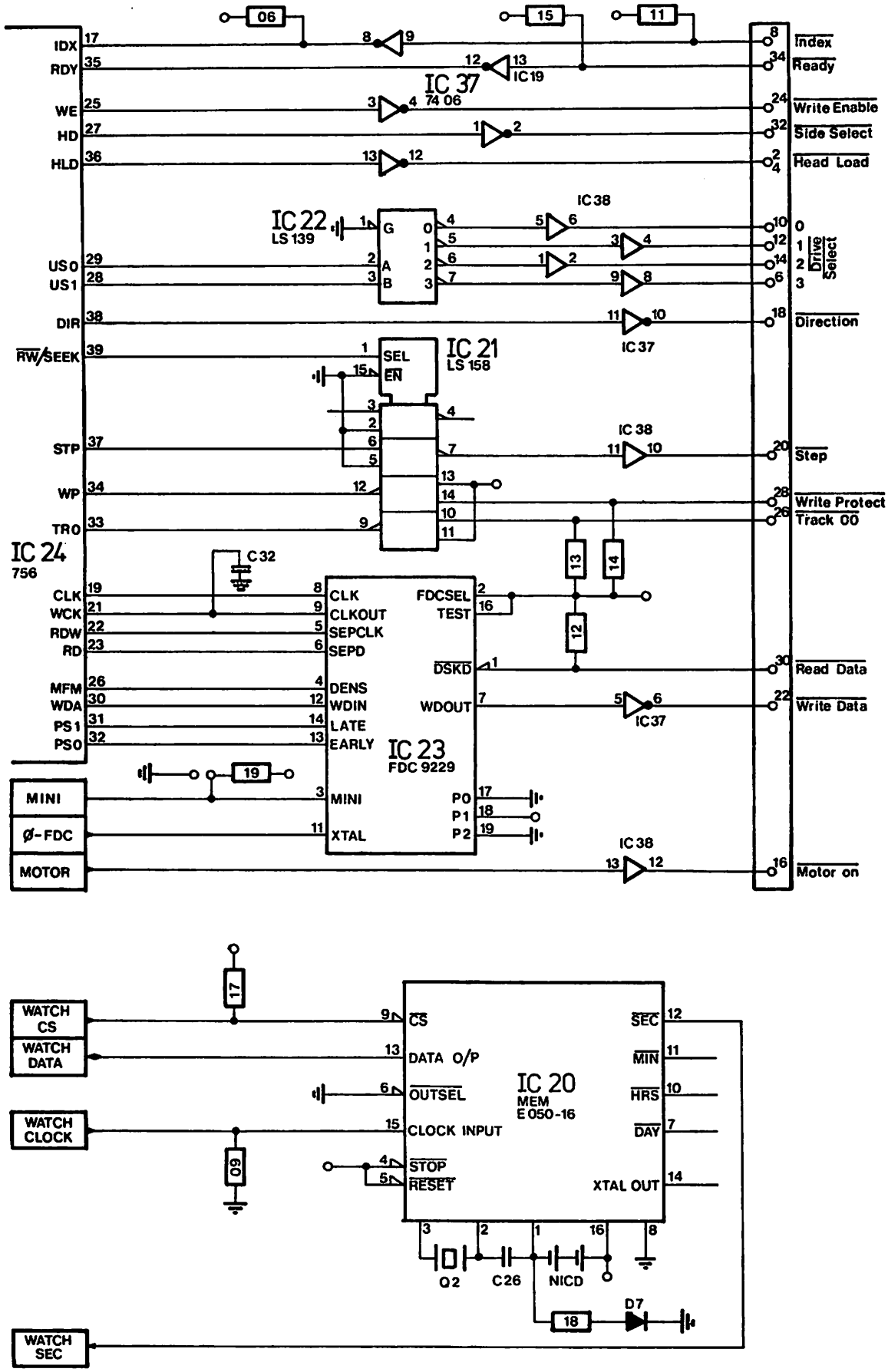
NICD	2x 20 DK-F











EPC - Einplatinencomputer

Betreff Einsatz von Eproms Typ 2732 4 kByte:

Der letzte Update des CP/M-Plus Betriebssystems machte den Einsatz von 32k Eproms des Typs 2732 für den Boot-Ladevorgang erforderlich. Der EPC-Einplatinencomputer ist jedoch standardmäßig für den Betrieb von Eproms 2716 eingestellt. Bitte beachten Sie hierzu daß Adresse A11 an Pin 21 des 24 poligen ROM-Sockel herangeführt werden muß (s. EPC-Handbuch Seite 59). Auf der Lötseite der Platine sind an Pin 21 die Adresse A11 und 5 V herangeführt. Die 5V Leitung ist vom Pin 21 zu unterbrechen (scharfes Messer), stattdessen ist die dem Pin gegenüberliegende Adresse A11 zu brücken (Lötpunkt).

Ansicht von der Lötseite PIN 21 IC 02: