ID-7047

HDLC/SDLC COMMUNICATION MODULE

DANSK DATA ELEKTRONIK APS

October 1978

1. General Description:

The ID-7047 HDLC communication module is used to implement HDLC/SDLC protocols in the ID-7000 microprocessor system. The module uses the INTEL 8273 HDLC/SDLC controller chip as described in appendix 1. The module is flexible and allows either DMA or non-DMA data transfers, interrupt or non-interrupt driven. If DMA tranfers are used, an ID-7009 DMA priority module must be present. The ID-7047 uses separate channels for input and output. If interrupt driven I/O is used an ID-7003 Interrupt Priority module must be present.

Fig. 1 is a blocked schematic of the board.

Appendix 2 is a complete logic schematic of the module.

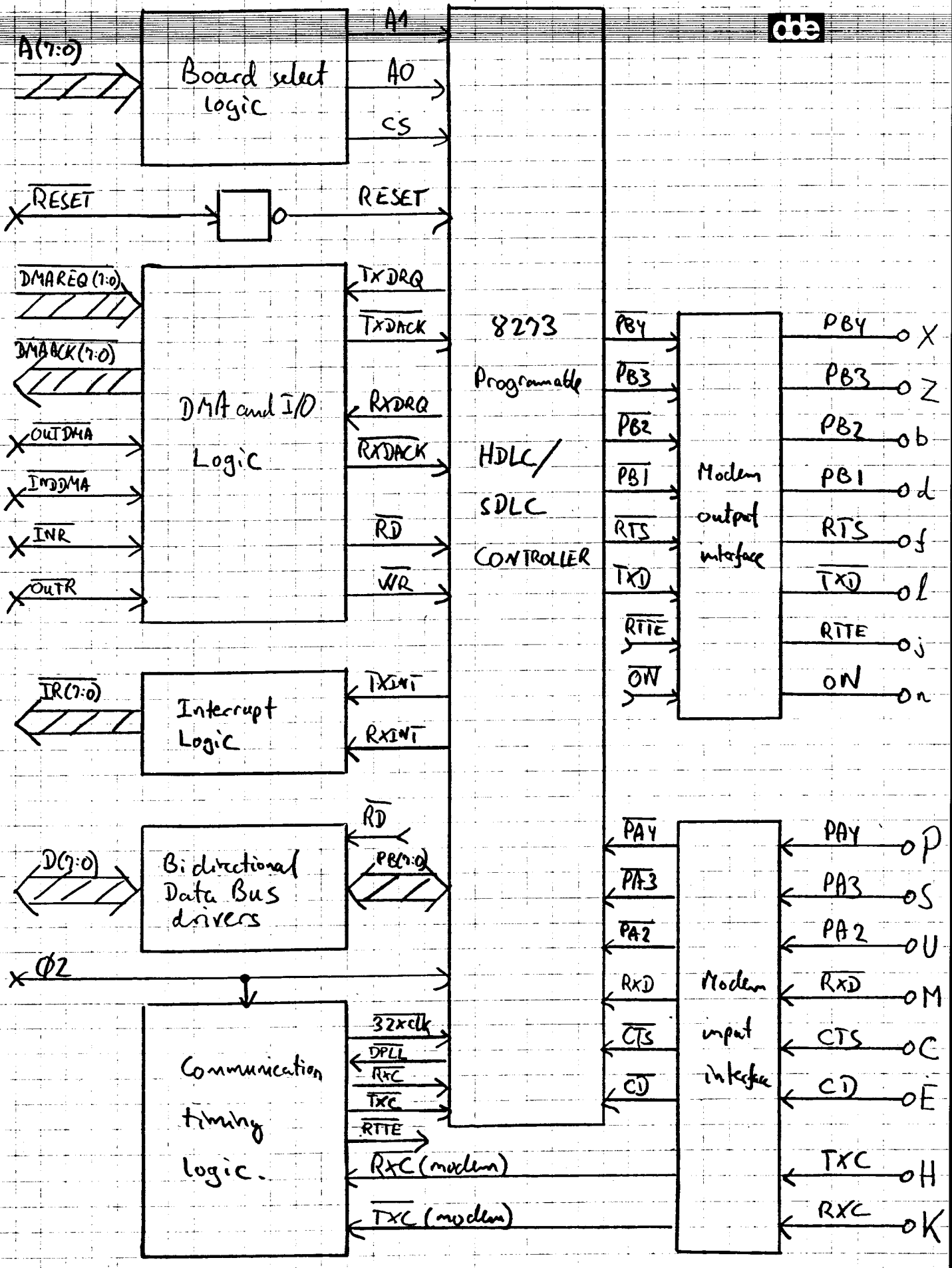


Fig. 1 ID-7047 Blocked Schematic.

2. Adressing. The module uses 8 concecutive I/O-addresses. A(7:3) is compared with an on board switch register to generate a card select signal.

A2, A1 and A0 are used to specify the following data-transfers between controller and CPU.
(Refer to appendix for detailed information).

A2	A1	A0	Input	Output
0	0	0	STATUS	COMMAND
0	0	1	RESULT	PARAMETER
0	1	0	TXINTResult	TESTMODE
0	1	1	RXINTResult	-
1	0	0	PROGRAMMED RXDATA	-
1	0	1	-	-
1	1	0	-	PROGRAMMED TXDATA
1	1	1	-	-

Addresses for PROGRAMMED RXDATA and PROGRAMMED TXDATA should only be used if NON DMA data tranfer mode is used.

3. Interrupts. This TXINT and the RXINT from the 8273 controller chip may be connected to the ID-7000 IR-bus. A 16 pin component board is used for this purpose (fig.2).

Interrupts strapping

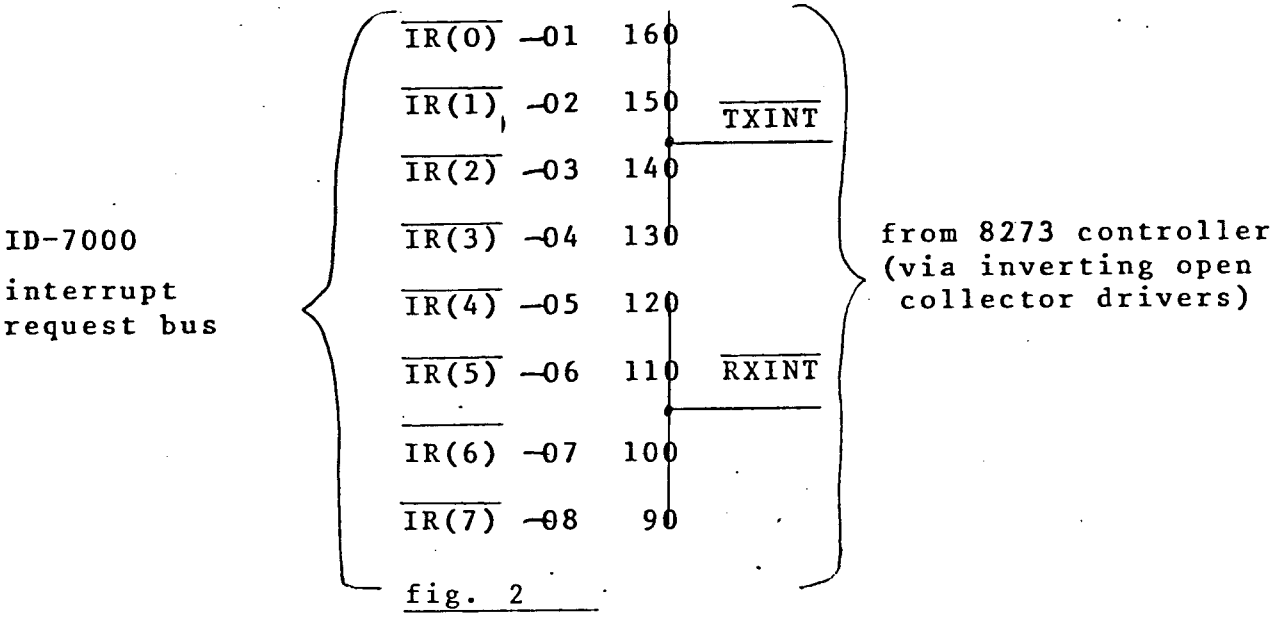


fig. 2

The interrupt sources may share common $\overline{IR}$-lines as the drivers are open collector.

Refer to appendix 1 for use of TXINT and RXINT

4. DMA Transfers. Using DMA, a complete datatransfer, when initalized by the CPU, can take place without using the CPU. The DMA-logic generates a $\overline{DMAREQ}$ for each 8 bit data element and the transfers takes place after the $\overline{DMAACK}$ signal has been received. Separate DMA channels are used for input and output.

Selection of DMA-channels are performed using two 16 pin component boards. One board is used for the connection to the $\overline{DMAREQ}$-bus and one board is used for connection to the $\overline{DMAACK}$-bus. The channel number should correspond for $\overline{DMAREQ}$ and $\overline{DMAACK}$. This is obtained by using identical strap boards. Fig. 3 shows the DMA strapping. If programmed datatransfers are wanted (running the 8273 in NON-DMA-mode) connections 9-10 and 15-16 are performed

in the DMAACK-strap area. In this case no connections to the $\overline{\text{DMAACK}}$-bus should be present.

DMA-strapping

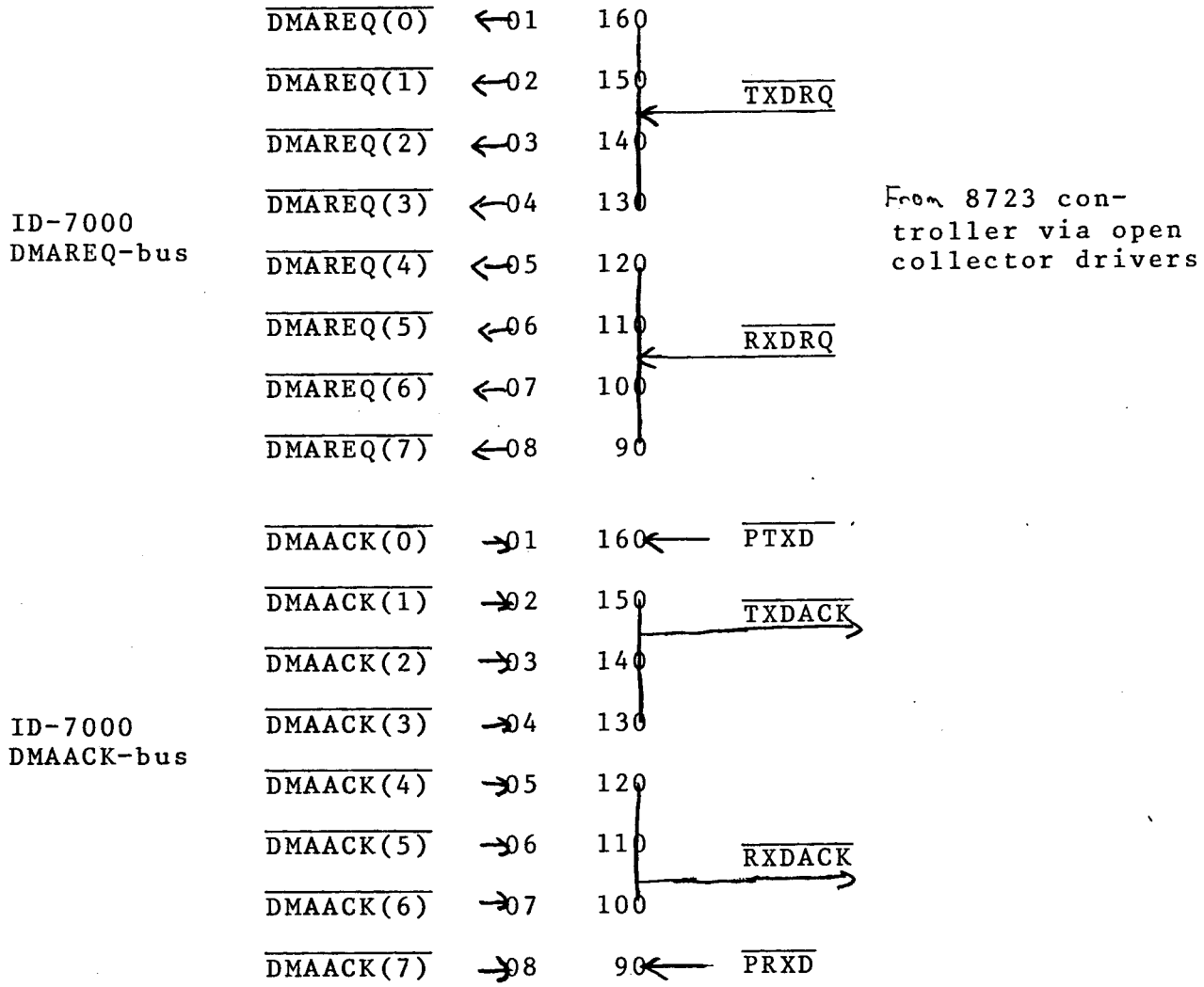


fig. 3

Refer to appendix 1 for detailed information concerning the DMA operation of the 8273 controller. Information concerning the DMA-priority module (ID-7009) is available in the manual for this module.

NB! It should be noticed, that the byte count is loaded to the 8273 controller prior to a complete datatransfer. The end interrupt is given when this number of bytes are transferred (refer to appendix 1). Prior to the DMA-transfer a memory

start address and a byte count should also be loaded to the ID-7009 DMA-priority module. The programmer should specify a byte count to the ID-7009 big enough to ensure that the DMA channel doesn't terminate before the data transfer is completed.

Baud rate selection.

The ID-7047 HDLC-communication module is primarily intended for synchronous datacommunication.

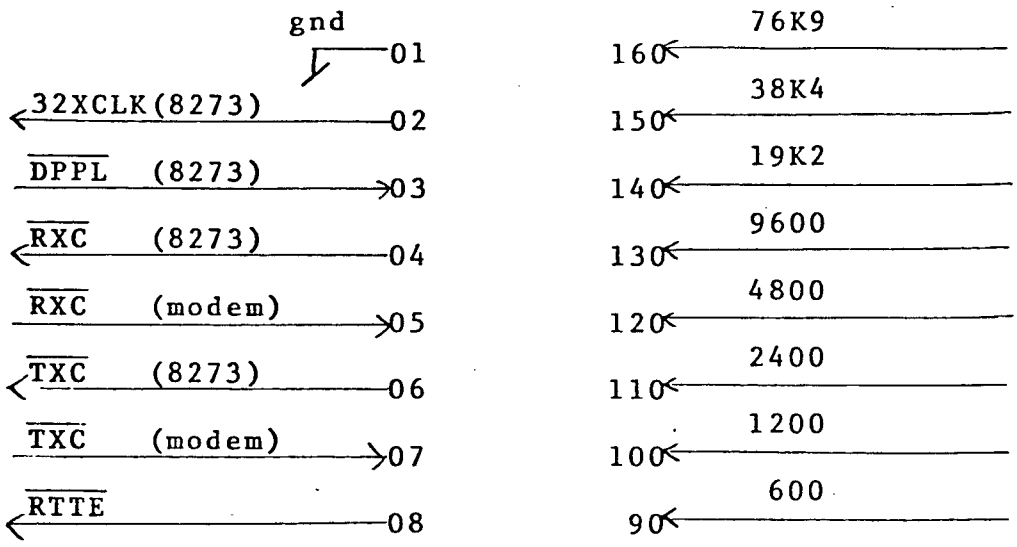
If synchronous modems are used for the communication channel, receiver and transmitter clocks are generated by the modem.

In some applications where the distance between the terminal equipment is short, a connection without modems could be used. In this case, a clock is available from the ID-7047 module. The available baud rates are: 76K9, 38K4, 19K2, 9600, 4800, 2400, 1200 and 600.

By using NRZI coding of data, it is possible also to use an asynchronous data channel by using an internal phase-lock-loop oscillator to generate the clock. In this case an input pin on the 8273 controller should contain a clock 32 times the used baud rate. This facility is also used in SDLC-loops for the loop terminals. Further information concerning this matter is available in appendix 1.

A component board is used to select the appropriate clock scheme:

Clock-strapping:



notes: 1. For normal synchronous operation where clocks are generated from modem, pins 4 and 5 are connected and pins 6 and 7 are connected. (Also connect pin 1→2).

2. If on board baud rate generator is used, connect pin 4 and pin 6 to appropriate baud rate (9—16). If this clock should be used in the connected equipment, it is available in the top connector with V24 signal levels. This is obtained by connecting pin 8 to appropriate baud rate. (Also connect pin 1 ← 2).

3. If asynchrounous or SDLC-loop operation is wanted, a clock should be connected to pin 2. In this case a clock phase-locked to received data is available on pin 3. This signal can be used as transmitter and/or receiver clock for the 8273 controller.

NB! If the phase locked loop facility is not used, the $\overline{32Xclk}$ pin should be grounded by connecting pin 1 and 2 together.

fig. 4

5. Modem interface.

The following signals are available in the top connector of the module for interface to modem or other equipment. The signal levels and loading capacity of these signals correspond to the CCITT V24 recommandation.

Inputs:

C:	CTS	(clear to send)
E:	CD	(carrier detect)
H:	TXC	(transmitter clock)
K:	RXC	(receiver clock)
M:	$\overline{\text{RXD}}$	(received data)
P:	PA4	user defined inputs
S:	PA3	
V:	PA2	

Outputs:

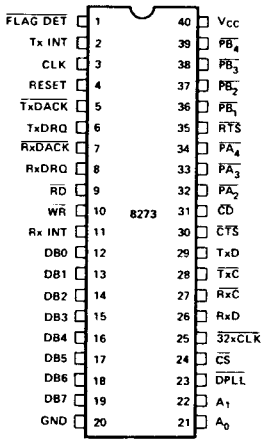
f:	RTS	(request to send)
l:	$\overline{\text{TXD}}$	(transmitted data)
j :	RTTE	(clock from internal baud rate generator)
n:	ON	(V24-ON signal)
X:	PB4	user defined outputs
Z:	PB3	
b	PB2	
d	PB1	

8273
PROGRAMMABLE HDLC/SDLC PROTOCOL CONTROLLER

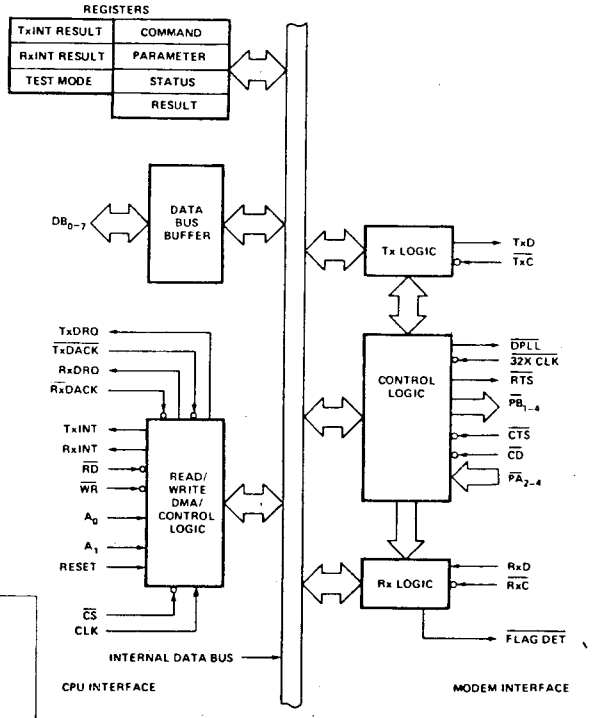
- HDLC/SDLC Compatible
- Frame Level Commands
- Full Duplex, Half Duplex, or Loop SDLC Operation
- Up to 64K Baud Transfers
- Two User Programmable Modem Control Ports
- Automatic FCS (CRC) Generation and Checking
- Programmable NRZI Encode/Decode
- N-Bit Reception Capability
- Digital Phase Locked Loop Clock Recovery
- Minimum CPU Overhead
- Fully Compatible with 8080/8085 CPUs
- Single +5V Supply
- 40-Pin Package

The Intel® 8273 Programmable HDLC/SDLC Protocol Controller is a dedicated device designed to support the ISO/C-CITT's HDLC and IBM's SDLC communication line protocols. It is fully compatible with Intel's new high performance microcomputer systems such as the MCS-85™. A frame level command set is achieved by a unique microprogrammed dual processor chip architecture. The processing capability supported by the 8273 relieves the system CPU of the low level real-time tasks normally associated with controllers.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN NAMES

DB0–DB7	DATA BUS (8 BITS)	$\overline{CS}$	CHIP SELECT
$\overline{FLAG\ DET}$	FLAG DETECT	$\overline{32xCLK}$	32 TIMES CLOCK
TxINT	TRANSMITTER INTERRUPT	Rx D	RECEIVER DATA
CLK	CLOCK INPUT	$\overline{Rx\ C}$	RECEIVER CLOCK
RESET	RESET	$\overline{Tx\ C}$	TRANSMITTER CLOCK
$\overline{Tx\ DACK}$	TRANSMITTER DMA ACKNOWLEDGE	Tx D	TRANSMITTER DATA
TxDRQ	TRANSMITTER DMA REQUEST	$\overline{CTS}$	CLEAR TO SEND
$\overline{RD}$	READ INPUT	$\overline{CD}$	CARRIER DETECT
$\overline{WR}$	WRITE INPUT	$\overline{PA_2}$–$\overline{PA_4}$	GP INPUT PORTS
$\overline{Rx\ DACK}$	RECEIVER DMA ACKNOWLEDGE	$\overline{PB_1}$–$\overline{PB_4}$	GP OUTPUT PORTS
Rx DRQ	RECEIVER DMA REQUEST	$\overline{RTS}$	REQUEST TO SEND
Rx INT	RECEIVER INTERRUPT	V_{CC}	+5 VOLT SUPPLY
A0–A1	COMMAND REGISTER SELECT ADDRESS	GND	GROUND
$\overline{DPLL}$	DIGITAL PHASE LOCKED LOOP		

MPU PERIPHERALS

A BRIEF DESCRIPTION OF HDLC/SDLC PROTOCOLS

General

The High Level Data Link Control (HDLC) is a standard communication link protocol established by International Standards Organization (ISO). HDLC is the discipline used to implement ISO X.25 packet switching systems.

The Synchronous Data Link Control (SDLC) is an IBM communication link protocol used to implement the System Network Architecture (SNA). Both the protocols are bit oriented, code independent, and ideal for full duplex communication. Some common applications include terminal to terminal, terminal to CPU, CPU to CPU, satellite communication, packet switching and other high speed data links. In systems which require expensive cabling and interconnect hardware, any of the two protocols could be used to simplify interfacing (by going serial), thereby reducing interconnect hardware costs. Since both the protocols are speed independent, reducing interconnect hardware could become an important application.

Network

In both the HDLC and SDLC line protocols, according to a pre-assigned hierarchy, a PRIMARY (Control) STATION controls the overall network (data link) and issues commands to the SECONDARY (Slave) STATIONS. The latter comply with instructions and respond by sending appropriate RESPONSES. Whenever a transmitting station must end transmission prematurely it sends an ABORT character. Upon detecting an abort character, a receiving station ignores the transmission block called a FRAME. Time fill between frames can be accomplished by transmitting either continuous frame preambles called FLAGS or an abort character. A time fill within a frame is not permitted. Whenever a station receives a string of more that fifteen consecutive ones, the station goes into an IDLE state.

Frames

A single communication element is called a FRAME which can be used for both Link Control and data transfer purposes. The elements of a frame are the beginning eight bit FLAG (F) consisting of one zero, six ones, and a zero, an eight bit ADDRESS FIELD (A), an eight bit CONTROL FIELD (C), a variable (N-bit) INFORMATION FIELD (I), a sixteen bit FRAME CHECK SEQUENCE (FCS), and an eight bit end FLAG (F), having the same bit pattern as the beginning flag. In HDLC the Address (A) and Control (C) bytes are extendable. The HDLC and the SDLC use three types of frames; an Information Frame is used to transfer data, a Supervisory Frame is used for control purposes, and a Non-sequenced Frame is used for initialization and control of the secondary stations.

Frame Characteristics

An important characteristic of a frame is that its contents are made code transparent by use of a zero bit insertion and deletion technique. Thus the user can adopt any format or code suitable for his system — it may even be a computer word length or a "memory dump". The frame is bit oriented that is, bits, not characters in each field, have specific meanings. The Frame Check Sequence (FCS) is an error detection scheme similar to the Cyclic Redundancy Checkword (CRC) widely used in magnetic disk storage devices. The Command and Response information frames contain sequence numbers in the control fields identifying the sent and received frames. The sequence numbers are used in Error Recovery Procedures (ERP) and as implicit acknowledgement of frame communication, enhancing the true full-duplex nature of the HDLC/SDLC protocols.

In contrast, BISYNC is basically half-duplex (two way alternate) because of necessity to transmit immediate acknowledgement frames. HDLC/SDLC therefore saves propagation delay times and have a potential of twice the throughput rate of BISYNC.

It is possible to use HDLC or SDLC over half duplex lines but there is a corresponding loss in throughput because both are primarily designed for full-duplex communication. As in any synchronous system, the bit rate is determined by the clock bits supplied by the modem, protocols themselves are speed independent.

A byproduct of the use of zero-bit insertion-deletion technique is the non-return-to-zero invert (NRZI) data transmission/reception compatibility. The latter allows HDLC/SDLC protocols to be used with asynchronous data communication hardware in which the clocks are derived from the NRZI encoded data.

References

IBM Synchronous Data Link Control General Information, IBM, GA27-3093-1.

Standard Network Access Protocol Specification, DATAPAC, Trans-Canada Telephone System CCG111

Recommendation X.25, ISO/CCITT March 2, 1976.

IBM 3650 Retail Store System Loop Interface OEM Information, IBM, GA 27-3098-0

Guidebook to Data Communications, Training Manual, Hewlett-Packard 5955-1715

IBM Introduction to Teleprocessing, IBM, GC 20-8095-02

System Network Architecture, Technical Overview, IBM, GA 27-3102

System Network Architecture Format and Protocol, IBM GA 27-3112

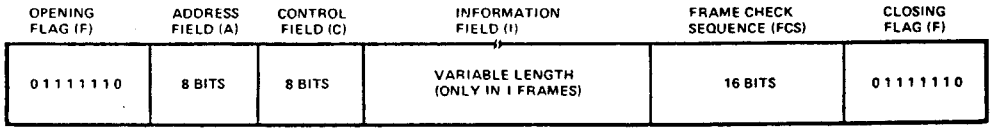


Figure 1. Frame Format

FUNCTIONAL DESCRIPTION

General

The Intel® 8273 HDLC/SDLC controller is a microcomputer peripheral device which supports the International Standards Organization (ISO) High Level Data Link Control (HDLC), and IBM Synchronous Data Link Control (SDLC) communications protocols. This controller minimizes CPU software by supporting a comprehensive frame-level instruction set and by hardware implementation of the low level tasks associated with frame assembly/disassembly and data integrity. The 8273 can be used in either synchronous or asynchronous applications. In asynchronous applications the data can be programmed to be encoded/decoded in NRZI code. The clock is derived from the NRZI data using a digital phase locked loop. The data transparency is achieved by using a zero-bit insertion/deletion technique. The frames are automatically checked for errors during reception by verifying the Frame Check Sequence (FCS); the FCS is automatically generated and appended before the final flag in transmit.

The 8273 recognizes and can generate flags (01111110), Abort, Idle, and GA (EOP) characters.

The 8273 can assume either a primary (control) or a secondary (slave) role. It can therefore be readily implemented in an SDLC loop configuration as typified by the IBM 3650 Retail Store System by programming the 8273 into a one-bit delay mode. In such a configuration, a two wire pair can be effectively used for data transfer between controllers and loop stations. The digital phase locked loop output pin can be used by the loop station without the presence of an accurate Tx clock.

Hardware Description

The 8273 is packaged in a 40 pin DIP. The following is a functional description of each pin.

Pin Name (No.)	I/O	Description
V_{CC} (40)		+5V Supply
GND (20)		Ground
RESET (4)	I	A high signal on this pin will force the 8273 to an idle state. The 8273 will remain idle until a command is issued by the CPU. The modem interface output signals are forced high. Reset must be true for a minimum of 10 TCY.
$\overline{CS}$ (24)	I	The $\overline{RD}$ and $\overline{WR}$ inputs are enabled by the chip select input.
DB_7-DB_0 (19-12)	I/O	The Data Bus lines are bidirectional three-state lines which interface with the system Data Bus.
$\overline{WR}$ (10)	I	The Write signal is used to control the transfer of either a command or data from CPU to the 8273.
$\overline{RD}$ (9)	I	The Read signal is used to control the transfer of either a data byte or a status word from the 8273 to the CPU.
TxINT (2)	O	The Transmitter interrupt signal indicates that the transmitter logic requires service.
RxINT (11)	O	The Receiver interrupt signal indicates that the Receiver logic requires service.
TxDRQ (6)	O	Requests a transfer of data between memory and the 8273 for a transmit operation.
RxRDQ (8)	O	Requests a transfer of data between the 8273 and memory for a receive operation.
$\overline{TxDACK}$ (6)	I	The Transmitter acknowledge signal notifies the 8273 that the TxDMA cycle has been granted.
$\overline{RxDACK}$ (7)	I	The Receiver DMA acknowledge signal notifies the 8273 that the RxDMA cycle has been granted.
A_1-A_0 (22-21)	I	These two lines are CPU Interface Register Select lines.
TxD (29)	O	This line transmits the serial data to the communication channel.
$\overline{TxC}$ (28)	I	The transmitter clock is used to synchronize the transmit data.
RxD (26)	I	This line receives serial data from the communication channel.
$\overline{RxC}$ (27)	I	The Receiver Clock is used to synchronize the receive data.
$\overline{32X\ CLK}$ (25)	I	The 32X clock is used to provide clock recovery when an asynchronous modem is used. In loop configuration the loop station can run without an accurate 1X clock by using the 32X CLK in conjunction with the DPLL output. (This pin must be grounded when not used).
$\overline{DPLL}$ (23)	O	Digital Phase Locked Loop output can be tied to RxC and/or TxC when 1X clock is not available. DPLL is used with 32X CLK.
$\overline{FLAG\ DET}$ (1)	O	Flag Detect signals that a flag (01111110) has been received by an active receiver.
$\overline{RTS}$ (35)	O	Request to Send signals that the 8273 is ready to transmit data.
$\overline{CTS}$ (30)	I	Clear to Send signals that the modem is ready to accept data from the 8273.
$\overline{CD}$ (31)	I	Carrier Detect signals that the line transmission has started and the 8273 may begin to sample data on RxD line.
$\overline{PA_{2-4}}$ (32-34)	I	General purpose input ports. The logic levels on these lines can be Read by the CPU through the Data Bus Buffer.
$\overline{PB_{1-4}}$ (36-39)	O	General purpose output ports. The CPU can write these output lines through Data Bus Buffer.
CLK (3)	I	A square wave TTL clock.

MPU PERIPHERALS

CPU Interface

The CPU interface is optimized for the MCS-80/85™ bus with an 8257 DMA controller. However, the interface is flexible, and allows either DMA or non-DMA data transfers, interrupt or non-interrupt driven. It further allows maximum line utilization by providing early interrupt mechanism for buffered (only the information field can be transferred to memory) Tx command overlapping. It also provides separate Rx and Tx interrupt output channels for efficient operation. The 8273 keeps the interrupt request active until all the associated interrupt results have been read.

The CPU utilizes the CPU interface to specify commands and transfer data. It consists of seven registers addressed via $\overline{CS}$, A_1, A_0, $\overline{RD}$ and $\overline{WR}$ signals and two independent data registers for receive data and transmit data. A_1, A_0 are generally derived from two low order bits of the address bus. If an 8080 based CPU is utilized, the $\overline{RD}$ and $\overline{WR}$ signals may be driven by the 8228 $\overline{I/OR}$ and $\overline{I/OW}$. The table shows the seven register select decoding:

Address Inputs		Control Logic Inputs	
A1	A0	$\overline{CS} \cdot \overline{RD}$	$\overline{CS} \cdot \overline{WR}$
0	0	Status	Command
0	1	Result	Parameter
1	0	TxINT Result	Test Mode
1	1	RxINT Result	—

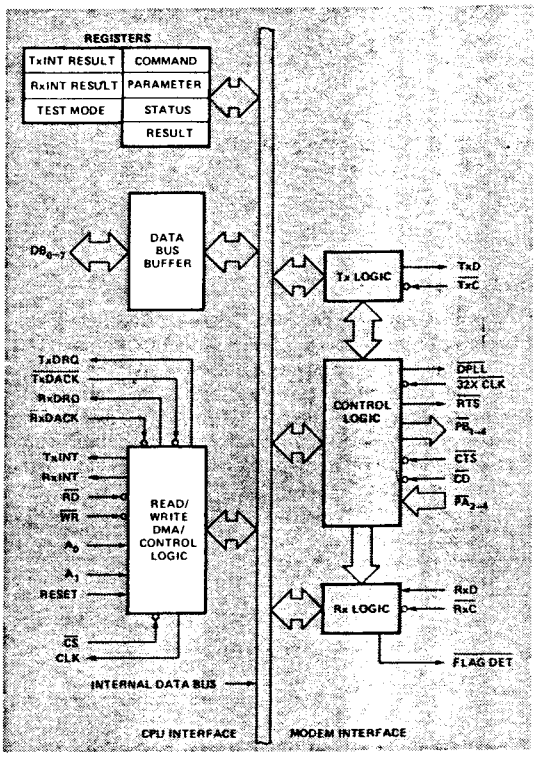


Figure 2. 8273 Block Diagram Showing CPU Interface Functions

Register Description

Command

Operations are initiated by writing an appropriate command in the Command Register.

Parameter

Parameters of commands that require additional information are written to this register.

Result

Contains an immediate result describing an outcome of an executed command.

Transmit Interrupt Result

Contains the outcome of 8273 transmit operation (good/bad completion).

Receive Interrupt Result

Contains the outcome of 8273 receive operation (good/bad completion), followed by additional results which detail the reason for interrupt.

Status

The status register reflects the state of the 8273 CPU Interface.

DMA Data Transfers

The 8273 CPU interface supports two independent data interfaces: receive data and transmit data. At high data transmission speeds the data transfer rate of the 8273 is great enough to justify the use of direct memory access (DMA) for the data transfers. When the 8273 is configured in DMA mode, the elements of the DMA interfaces are:

TxDRQ: Transmit DMA Request

Requests a transfer of data between memory and the 8273 for a transmit operation.

$\overline{TxDACK}$: Transmit DMA Acknowledge

The $\overline{TxDACK}$ signal notifies the 8273 that a transmit DMA cycle has been granted.

RxDRQ: Receive DMA Request

Requests a transfer of data between the 8273 and memory for a receive operation.

$\overline{\text{RxDACK}}$: Receive DMA Acknowledge

The $\overline{\text{RxDACK}}$ signal notifies the 8273 that a receive DMA cycle has been granted.

$\overline{\text{RD}}$, $\overline{\text{WR}}$: Read, Write

The $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals are used to specify the direction of the data transfer.

DMA transfers require the use of a DMA controller such as the Intel 8257. The function of the DMA controller is to provide sequential addresses and timing for the transfer, at a starting address determined by the CPU. Counting of data block lengths is performed by the 8273.

To request a DMA transfer the 8273 raises the appropriate DMA REQUEST. DMA ACKNOWLEDGE and READ enables DMA data onto the bus (independently of CHIP SELECT). DMA ACKNOWLEDGE and WRITE transfers DMA data to the 8273 (independent of CHIP SELECT).

It is also possible to configure the 8273 in the non-DMA data transfer mode. In this mode the CPU module must pass data to the 8273 in response to non-DMA data requests indicated by the status word.

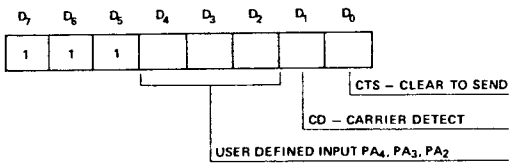
Modem Interface

The 8273 Modem interface provides both dedicated and user defined modem control functions. All the signals are active low so that EIA RS-232C inverting drivers (MC 1488) and inverting receivers (MC 1489) may be used to interface to standard modems. For asynchronous operation, this interface supports programmable NRZI data encode/decode, a digital phase locked loop for efficient clock extraction from NRZI data, and modem control ports with automatic $\overline{\text{CTS}}$, $\overline{\text{CD}}$ monitoring and $\overline{\text{RTS}}$ generation. This interface also allows the 8273 to operate in PRE-FRAME SYNC mode in which the 8273 prefixes 16 transitions to a frame to synchronize idle lines before transmission of the first flag.

It should be noted that all the 8273 port operations deal with logical values, for instance, bit D0 of Port A will be a one when $\overline{\text{CTS}}$ (Pin 30) is a physical zero (logical one).

Port A — Input Port

During operation, the 8273 interrogates input pins $\overline{\text{CTS}}$ (Clear to Send) and $\overline{\text{CD}}$ (Carrier Detect). $\overline{\text{CTS}}$ is used to condition the start of a transmission. If during transmission $\overline{\text{CTS}}$ is lost the 8273 generates an interrupt. During reception, if $\overline{\text{CD}}$ is lost, the 8273 generates an interrupt.



The user defined input bits correspond to the 8273 PA_4, PA_3 and PA_2 pins. The 8273 does not interrogate or manipulate these bits.

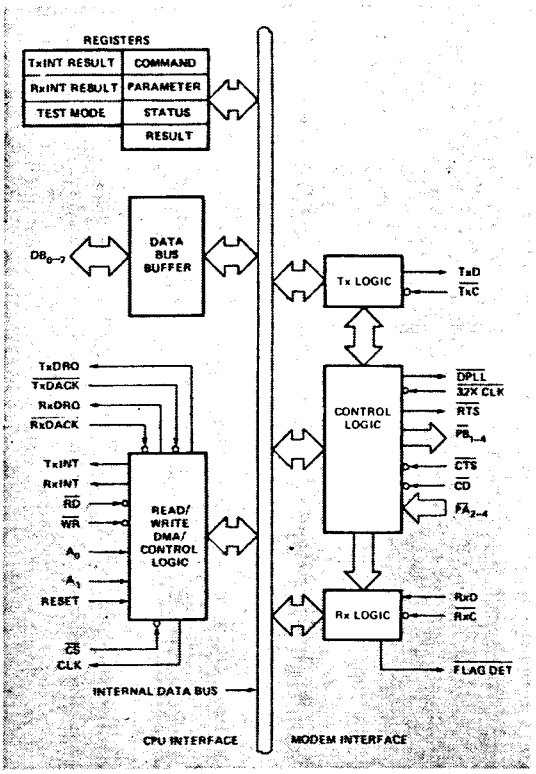
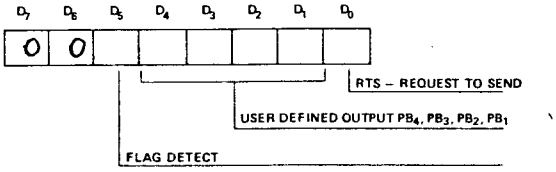


Figure 3. 8273 Block Diagram Showing Control Logic Functions

Port B - Output Port

During normal operation, if the CPU sets $\overline{\text{RTS}}$ active, the 8273 will not change this pin; however, if the CPU sets $\overline{\text{RTS}}$ inactive, the 8273 will activate it before each transmission and deactivate it one byte time after transmission. While the receiver is active the flag detect pin is pulsed each time a flag sequence is detected in the receive data stream. Following an 8273 reset, all pins of Port B are set to a high, inactive level.



The user defined output bits correspond to the state of PB_4-PB_1 pins. The 8273 does not interrogate or manipulate these bits.

MPU PERIPHERALS

Serial Data Logic

The Serial data is synchronized by the user transmit ($\overline{TxC}$) and receive ($\overline{RxC}$) clocks. The leading edge of $\overline{TxC}$ generates new transmit data and the trailing edge of $\overline{RxC}$ is used to capture receive data. The NRZI encoding/decoding of the receive and transmit data is programmable.

The diagnostic features included in the Serial Data logic are programmable loop back of data and selectable clock for the receiver. In the loop-back mode, the data presented to the TxD pin is internally routed to the receive data input circuitry in place of the RxD pin, thus allowing a CPU to send a message to itself to verify operation of the 8273.

In the selectable clock diagnostic feature, when the data is looped back, the receiver may be presented incorrect sample timing by the external circuitry. The user may select to substitute the $\overline{TxC}$ pin for the $\overline{RxC}$ input on-chip so that the clock used to generate the loop back data is used to sample it. Since TxD is generated off the leading edge of $\overline{TxC}$ and RxD is sampled on the trailing edge, the selected clock allows bit synchronism.

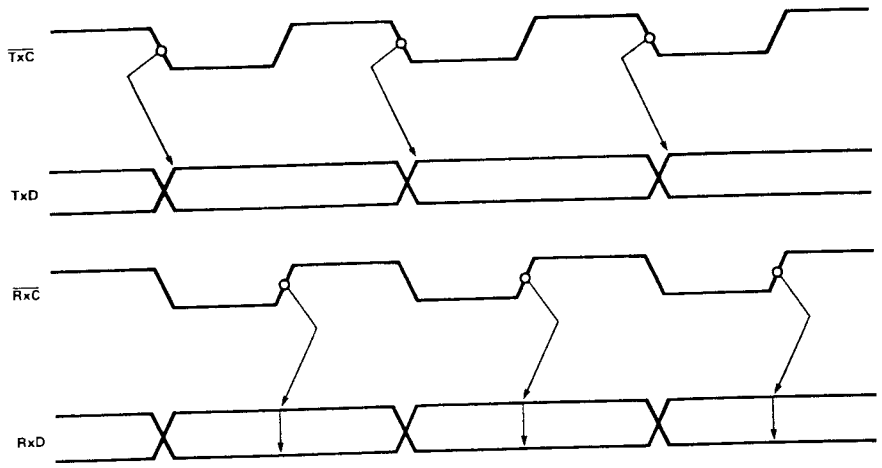


Figure 4. Transmit/Receive Timing

Asynchronous Mode Interface

Although the 8273 is fully compatible with the HDLC/SDLC communication line protocols, which are primarily designed for synchronous communication, the 8273 can also be used in asynchronous applications by using this interface. The interface employs a digital phase locked loop (DPLL) for clock recovery from a receive data stream and programmable NRZI encoding and decoding of data. The use of NRZI coding with SDLC transmission guarantees that within a frame, data transitions will occur at least every five bit times — the longest sequence of ones which may be transmitted without zero-bit insertion. The DPLL should be used only when NRZI coding is used since the NRZI coding will transmit zero sequence as line transitions. The digital phase locked loop also facilitates full-duplex and half-duplex asynchronous implementation with, or without modems.

Digital Phase Locked Loop

In asynchronous applications, the clock is derived from the receiver data stream by the use of the digital phase locked loop (DPLL). The DPLL requires a clock input at 32 times the required baud rate. The receive data (RxD) is sampled with this $\overline{32X\ CLK}$ and the 8273 DPLL supplies a sample pulse nominally centered on the RxD bit cells. The DPLL has a built-in "stiffness" which reduces sensitivity to line noise and bit distortion. This is accomplished by making phase error adjustments in discrete increments. Since the nominal pulse is made to occur at 32 counts of the $\overline{32X\ CLK}$, these counts are subtracted or added to the nominal, depending upon which quadrant of the four error quadrants the data edge occurs in. For example if an RxD edge is detected in quadrant A1, it is apparent that the $\overline{DPLL}$ sample "A" was placed too close to the trailing edge of the data cell; sample "B" will then be placed at T = ($T_{nominal}$ - 2 counts) = 30 counts of the $\overline{32X\ CLK}$ to move the sample pulse "B" toward the nominal center of the next bit cell. A data edge occuring in quadrant B1 would cause a smaller adjustment of phase with T = 31 counts of the $\overline{32X\ CLK}$. Using this technique the $\overline{DPLL}$ pulse will converge to nominal bit center within 12 data bit times, worst case, with constant incoming RxD edges.

A method of attaining bit synchronism following a line idle is to use PRE-FRAME SYNC mode of transmission.

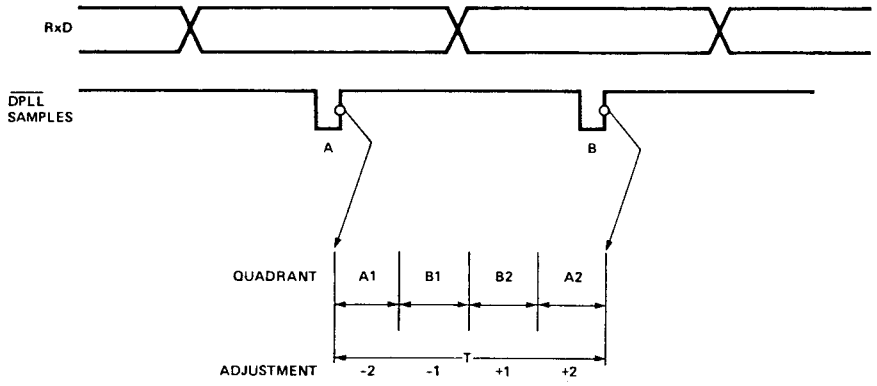
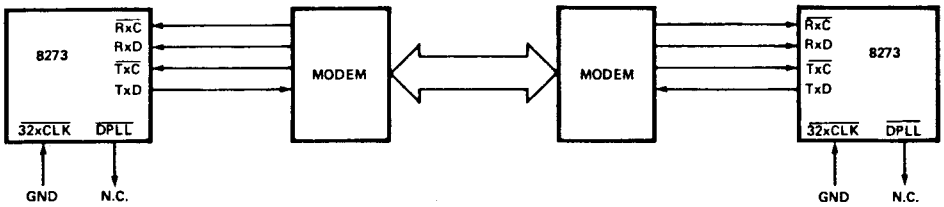


Figure 5. DPLL Sample Timing

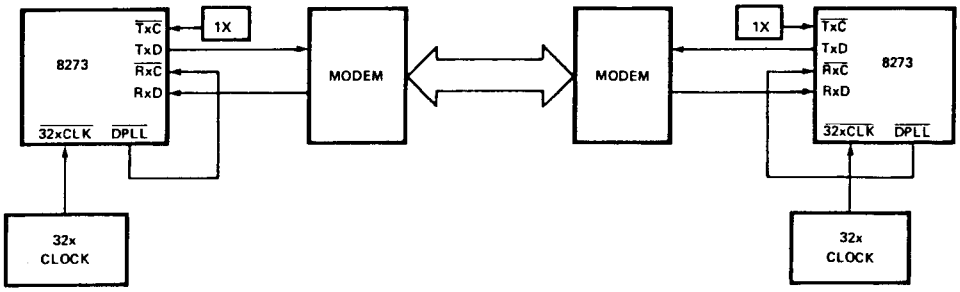
MPU PERIPHERALS

A1.8

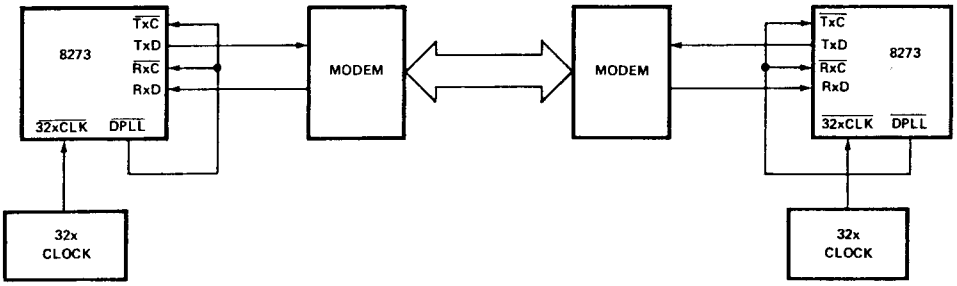
Synchronous Modem — Duplex or Half Duplex Operation



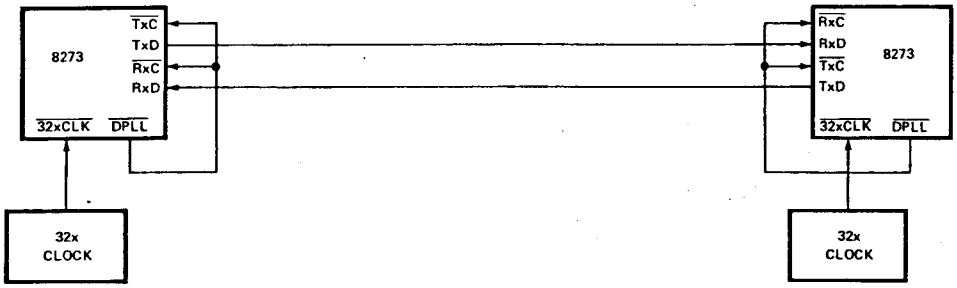
Asynchronous Modems — Duplex Operation



Asynchronous Modems — Half Duplex Operation



Asynchronous — No Modems — Duplex or Half Duplex



SDLC Loop

The DPLL simplifies the SDLC loop station implementation. In this application, each secondary station on a loop data link is a repeater set in one-bit delay mode. The signals sent out on the loop by the loop controller (primary station) are relayed from station to station then, back to the controller. Any secondary station finding its address in the A field captures the frame for action at that station. All received frames are relayed to the next station on the loop.

Loop stations are required to derive bit timing from the incoming NRZI data stream. The DPLL generates sample Rx clock timing for reception and uses the same clock to implement Tx clock timing.

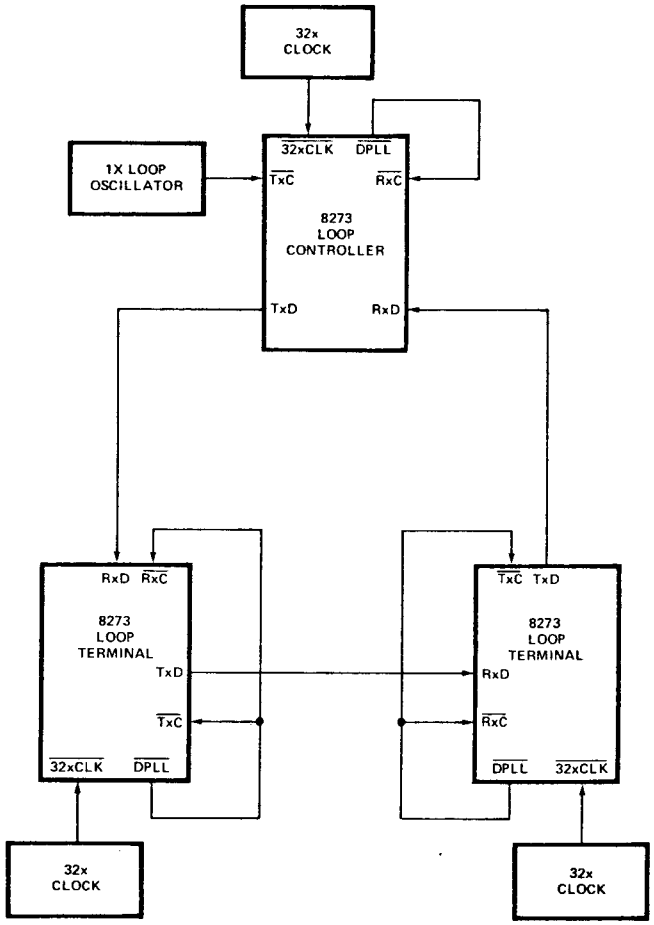
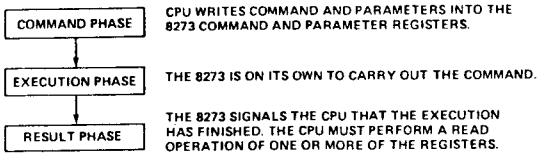


Figure 6. SDLC Loop Application

PRINCIPLES OF OPERATION

The 8273 is an intelligent peripheral controller which relieves the CPU of many of the rote tasks associated with constructing and receiving frames. It is fully compatible with the MCS-80/85™ system bus. As a peripheral device, it accepts commands from a CPU, executes these commands and provides an Interrupt and Result back to the CPU at the end of the execution. The communication with the CPU is done by activation of $\overline{CS}$, $\overline{RD}$, $\overline{WR}$ pins, while the A_1, A_0 select the appropriate registers on the chip as described in the Hardware Description Section.

The 8273 operation is composed of the following sequence of events:



The Command Phase

During the command phase, the software writes a command to the command register. The command bytes provide a general description of the type of operation requested. Many commands require more detailed information about the command. In such a case up to four parameters are written into the parameter register. The flowchart of the command phase indicates that a command may not be issued if the Status Register indicates that the device is busy. Similarly if a parameter is issued when the Parameter Buffer shows full, incorrect operation will occur.

The 8273 is a duplex device and both transmitter and receiver may each be executing a command or passing results at any given time. For this reason separate interrupt pins are provided. However, the command register must be used for one command sequence at a time.

Status Register

The status register contains the status of the 8273 activity. The description is as follows.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CBSY	CBF	CPBF	CRBF	RxINT	TxINT	RxIRA	TxIRA

Bit 7 CBSY (Command Busy)

Indicates in-progress command, set for CPU poll when Command Register is full, reset upon command phase completion. It is improper to write a command when CBSY is set; it results in incorrect operation.

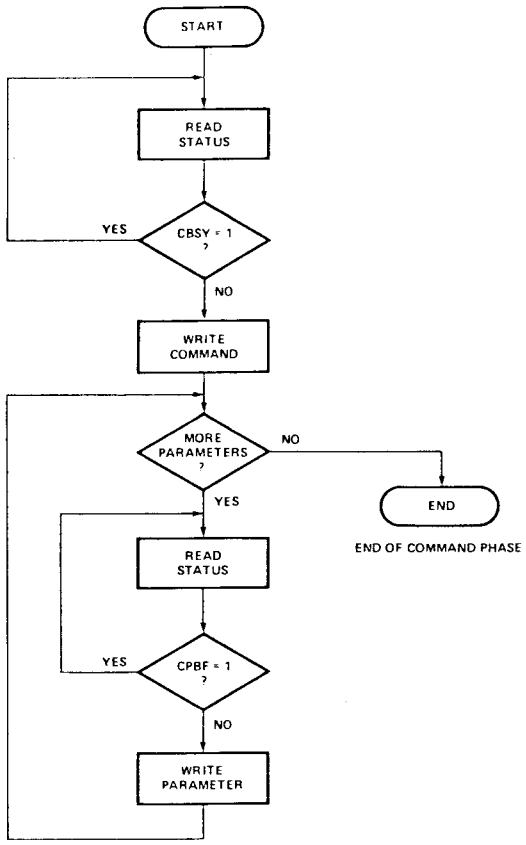


Figure 7. Command Phase Flowchart

Bit 6 CBF (Command Buffer Full)

Indicates that the command register is full, it is reset when the 8273 accepts the command byte but does not imply that execution has begun.

Bit 5 CPBF (Command Parameter Buffer Full)

CPBF is set when the parameter buffer is full, and is reset by the 8273 when it accepts the parameter. The CPU may poll CPBF to determine when additional parameters may be written.

Bit 4 CRBF (Command Result Buffer Full)

Indicates that an executed command immediate result is present in the Result Register. It is set by 8273 and reset when CPU reads the result.

Bit 3 RxINT (Receiver Interrupt)

RxINT indicates that the receiver requires CPU attention. It is identical to RxINT (pin 11) and is set by the 8273 either upon good/bad completion of a specified command or by Non-DMA data transfer. It is reset only after the CPU has read the result byte or has received a data byte from the 8273 in a Non-DMA data transfer.

Bit 2 TxINT (Transmitter Interrupt)

The TxINT indicates that the transmitter requires CPU attention. It is identical to TxINT (pin 2). It is set by 8273 either upon good/bad completion of a specified command or by Non-DMA data transfer. It is reset only after the CPU has read the result byte or has transferred transmit data byte to the 8273 in a Non-DMA transfer.

Bit 1 RxIRA (Receiver Interrupt Result Available)

The RxIRA is set by the 8273 when an interrupt result byte is placed in the RxIRA register. It is reset after the CPU has read the RxIRA register.

Bit 0 TxIRA (Transmitter Interrupt Result Available)

The TxIRA is set by the 8273 when an interrupt result byte is placed in the TxIRA register. It is reset when the CPU has read the TxIRA register.

The Execution Phase

Upon accepting the last parameter, the 8273 enters into the Execution Phase. The execution phase may consist of a DMA or other activity, and may or may not require CPU intervention. The CPU intervention is elliminated in this phase if the system utilizes DMA for the data transfers, otherwise, for non-DMA data transfers, the CPU is interrupted by the 8273 via TxINT and RxINT pins, for each data byte request.

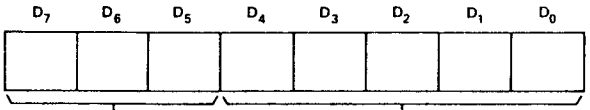
The Result Phase

During the result phase, the 8273 notifies the CPU of the execution outcome of a command. This phase is initiated by:

1. The successful completion of an operation
2. An error detected during an operation.

To facilitate quick network software decisions, two types of execution results are provided:

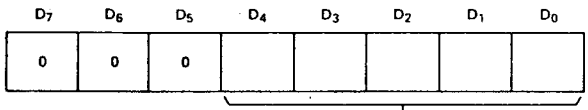
1. An Immediate Result
2. A Non-Immediate Result



D_7	D_6	D_5	
1	1	1	All 8-bits received
0	0	0	D0 received
1	0	0	D_1-D_0 received
0	1	0	D_2-D_0 received
1	1	0	D_3-D_0 received
0	0	1	D_4-D_0 received
1	0	1	D_5-D_0 received
0	1	1	D_6-D_0 received

D_4	D_3	D_2	D_1	D_0	
0	0	0	0	0	A_1 match or general receive
0	0	0	0	1	A_2 match
0	0	0	1	1	CRC error
0	0	1	0	0	Abort detected
0	0	1	0	1	Idle detect
0	0	1	1	0	EOP detected
0	0	1	1	1	Frame less than 32-bits
0	1	0	0	0	DMA overrun detected
0	1	0	0	1	Memory buffer overflow
0	1	0	1	0	Carrier detect failure
0	1	0	1	1	Receive interrupt overrun

Figure 8. Rx Interrupt Result Byte Format



D_4	D_3	D_2	D_1	D_0	
0	1	1	0	0	Early transmit interrupt
0	1	1	0	1	Frame transmit complete
0	1	1	1	0	DMA underrun
0	1	1	1	1	Clear to Send (CTS) error
1	0	0	0	0	Abort complete

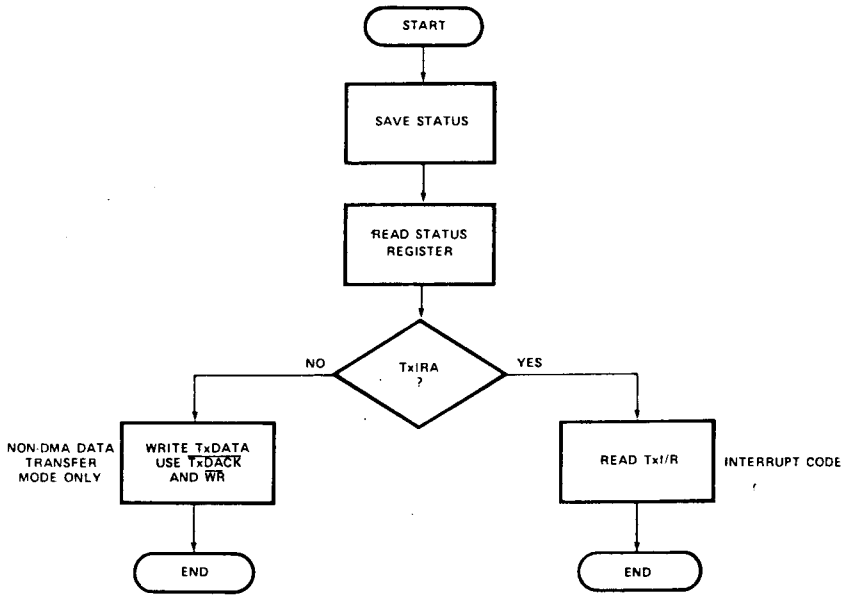
Figure 9. Tx Interrupt Result Byte Format

Immediate result is provided by the 8273 for commands such as Read Port A and Read Port B which have information ($\overline{CTS}$, $\overline{CD}$, $\overline{RTS}$, etc.) that the network software needs to make quick operational decisions.

A command which cannot provide an immediate result will generate an interrupt to signal the beginning of the Result phase. The immediate results are provided in the Result Register; all non-immediate results are available upon device interrupt, through Tx Interrupt Result Register TxI/R or Rx Interrupt Result Register RxI/R. The result may consist of a one-byte interrupt code indicating the condition for the interrupt and, if required, one or more bytes which detail the condition.

Tx and Rx Interrupt Result Registers

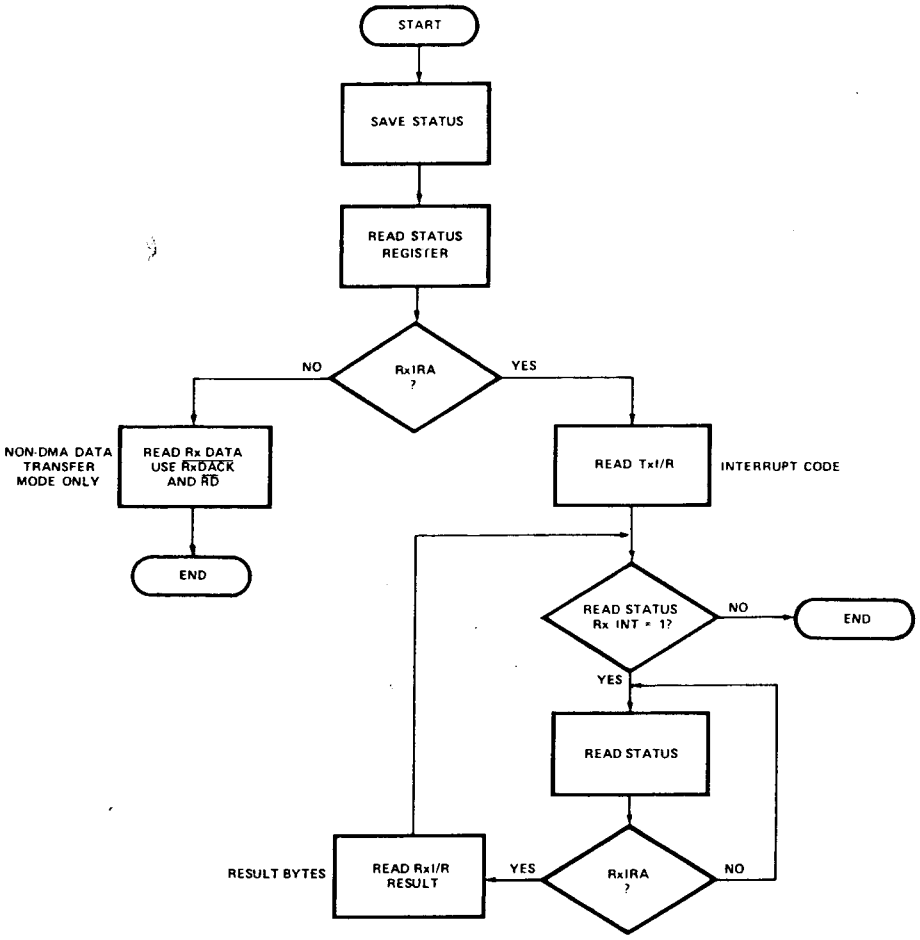
The Result Registers have a result code, the three high order bits D_7-D_5 of which are set to zero for all but the receive command. This command result contains a count that indicates the number of bits received in the last byte. If a partial byte is received, the high order bits of the last data byte are indeterminate.



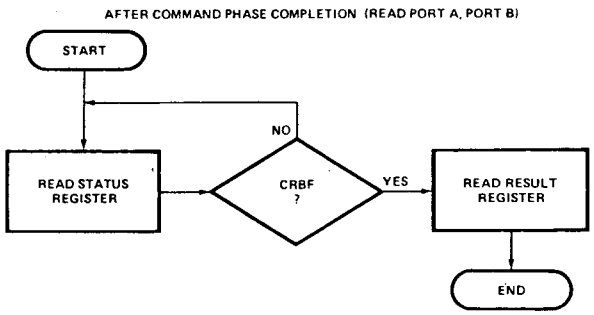
RESULT PHASE FLOWCHART — Tx INTERRUPT RESULTS

Figure 10. Tx Interrupt Service

A1.13



RESULT PHASE FLOWCHART — INTERRUPT RESULTS



RESULT PHASE FLOWCHART — IMMEDIATE RESULTS

Figure 11. Rx Interrupt Service

DETAILED COMMAND DESCRIPTION

General

The 8273 HDLC/SDLC controller supports a comprehensive set of high level commands which allows the 8273 to be readily used in full-duplex, half-duplex, synchronous, asynchronous and SDLC loop configuration, with or without modems. These frame-level commands minimize CPU and software overhead. The 8273 has address and control byte buffers which allow the receive and transmit commands to be used in buffered or non-buffered modes.

In buffered transmit mode, the 8273 transmits a flag automatically, reads the Address and Control buffer registers and transmits the fields, then via DMA, it fetches the information field. The 8273, having transmitted the information field, automatically appends the Frame Check Sequence (FCS) and the end flag. Correspondingly, in buffered read mode, the Address and Control fields are stored in their respective buffer registers and only Information Field is transferred to memory.

In non-buffered transmit mode, the 8273 transmits the beginning flag automatically, then fetches and transmits the Address, Control and Information fields from the memory, appends the FCS character and an end flag. In the non-buffered receive mode the entire contents of a frame are sent to memory with the exception of the flags and FCS.

HDLC Implementation

HDLC Address and Control field are extendable. The extension is selected by setting the low order bit of the field to be extended to a one, a zero in the low order bit indicates the last byte of the respective field.

Since Address/Control field extension is normally done with software to maximize extension flexibility, the 8273 does not create or operate upon contents of the extended HDLC Address/Control fields. Extended fields are transparently passed by the 8273 to user as either interrupt results or data transfer requests. Software must assemble the fields for transmission and interrogate them upon reception.

However, the user can take advantage of the powerful 8273 commands to minimize CPU/Software overhead and simplify buffer management in handling extended fields. For instance buffered mode can be used to separate the first two bytes, then interrogate the others from buffer. Buffered mode is perfect for a two byte address field.

The 8273 when programmed, recognizes protocol characters unique to HDLC such as Abort, which is a string of seven or more ones (01111111). Since Abort character is the same as the GA (EOP) character used in SDLC Loop applications, Loop Transmit and Receive commands are not recommended to be used in HDLC. HDLC does not support Loop mode.

Initialization Set/Reset Commands

These commands are used to manipulate data within the 8273 registers. The Set commands have a single parameter which is a mask that corresponds to the bits to be set. (They perform a logical-OR of the specified register with the mask provided as a parameter). The Register commands have a single parameter which is a mask that has a zero in the bit positions that are to be reset. (They perform a logical-AND of the specified register with the mask).

Set One-Bit Delay (CMD Code A4)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	0	1	0	0	1	0	0
PAR:	0	1	1	0	0	0	0	0	0	0

When one bit delay is set, 8273 retransmits the received data stream one bit delayed. This mode is entered at a receiver character boundary, and should only be used by Loop Stations.

Reset One-Bit Delay (CMD Code 64)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	0	1	1	0	0	1	0	0
PAR:	0	1	0	1	1	1	1	1	1	1

The 8273 stops the one bit delayed retransmission mode.

Set Data Transfer Mode (CMD Code 97)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	0	0	1	0	1	1	1
PAR:	0	1	0	0	0	0	0	0	0	1

When the data transfer mode is set, the 8273 will interrupt when data bytes are required for transmission or are available from a receive. If a transmit interrupt occurs and the status indicates that there is no Transmit Result (TxIRA = 0), the interrupt is a transmit data request. If a receive interrupt occurs and the status indicates that there is no receive result (RxIRA = 0), the interrupt is a receive data request.

Reset Data Transfer Mode (CMD Code 57)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	0	1	0	1	0	1	1	1
PAR:	0	1	1	1	1	1	1	1	1	0

If the Data Transfer Mode is reset, the 8273 data transfers are performed through the DMA requests without interrupting the CPU.

A1_15

Set Operating Mode (CMD Code 91) */

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	0	0	1	0	0	0	1
PAR:	0	1	0	0						

1 = FLAG STREAM MODE
1 = PREFRAME SYNC MODE
1 = BUFFERED MODE
1 = EARLY INTERRUPT MODE
1 = EOP INTERRUPT MODE
1 = HDLC MODE

Reset Operating Mode (CMD Code 51)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	0	1	0	1	0	0	0	1
PAR:	0	1	1	1						

Any mode switches set in CMD code 91 can be reset using this command by placing zeros in the appropriate positions.

(D5) HDLC Mode

In HDLC mode, a bit sequence of seven ones (01111111) is interpreted as an abort character. Otherwise, eight ones (011111111) signal an abort.

(D4) EOP Interrupt Mode

In EOP interrupt mode, an interrupt is generated whenever an EOP character (01111111) is detected by an active receiver. This mode is useful for the implementation of an SDLC loop controller in detecting the end of a message stream after a loop poll.

(D3) Transmitter Early Interrupt Mode (Tx)

The early interrupt mode is specified to indicate when the 8273 should generate an end of frame interrupt. When set, an early interrupt is generated when the last data character has been passed to the 8273. If the user software responds with another transmit command before the final flag is sent, the final flag interrupt will not be generated and a new frame will immediately begin when the current frame is complete. This permits frames to be separated by a single flag. If no additional Tx commands are provided, a final interrupt will follow.

If this bit is zero, the interrupt will be generated only after the final flag has been transmitted.

(D2) Buffered Mode

If the buffered mode bit is set to a one, the first two bytes (normally the address (A) and control (C) fields) of a frame are buffered by the 8273. If this bit is a zero the address and control fields are passed to and from memory.

(D1) Preframe Sync Mode

If this bit is set to a one the 8273 will transmit two characters before the first flag of a frame.
To guarantee sixteen line transitions, the 8273 sends two bytes of data $(00)_H$ if NRZI is set or data $(55)_H$ if NRZI is not set.

(D0) Flag Stream Mode

If this bit is set to a one, the following table outlines the operation of the transmitter.

TRANSMITTER STATE	ACTION
Idle	Send Flags immediately.
Transmit or Transmit-Transparent Active	Send Flags after the transmission complete
Loop Transmit Active	Ignore command.
1 Bit Delay Active	Ignore command.

If this bit is reset to zero the following table outlines the operation of the transmitter.

TRANSMITTER STATE	ACTION
IDLE	Send Idles on next character boundary.
Transmit or Transmit-Transparent Active	Send Idles after the transmission is complete.
Loop Transmit Active	Ignore command.
1 Bit Delay Active	Ignore command.

Set Serial I/O Mode (CMD Code A0)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	0	1	0	0	0	0	0
PAR:	0	1	0	0	0	0	0			

1 = NRZ1 MODE
1 = TxC → RxC
1 = LOOP BACK TxD → RxD

Reset Serial I/O Mode (CMD Code 60)

This command allows bits set in CMD code A0 to be reset by placing zeros in the appropriate positions.

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	0	1	1	0	0	0	0	0
PAR:	0	1	1	1	1	1	1			

(D2) Loop Back

If this bit is set to a one, the transmit data is internally routed to the receive data circuitry.

(D1) TxC → RxC

If this bit is set to a one, the transmit clock is internally routed to the receive clock circuitry. It is normally used with the loop back bit (D2).

(D0) NRZI Mode

If this bit is set to a one, NRZI encoding and decoding of transmit and receive data is provided. If this bit is a zero, the transmit and receive data is treated as a normal positive logic bit stream.

NRZI encoding specifies that a zero causes a change in the polarity of the transmitted signal and a one causes no polarity change. NRZI is used in all asynchronous operations. Refer to IBM document GA27-3093 for details.

*/ This command should be sent 12.156 the first command after reset command

A1.16

Reset Device Command

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
TMR:	1	0	0	0	0	0	0	0	0	1
TMR:	1	0	0	0	0	0	0	0	0	0

An 8273 reset command is executed by outputing a $(01)_H$ followed by $(00)_H$ to the test mode register (TMR). See 8273 AC timing characteristics for Reset pulse specifications.

The reset command emulates the action of the reset pin.

1. The modem control signals are forced high (inactive level).
2. The 8273 status register flags are cleared.
3. Any commands in progress are terminated immediately.
4. The 8273 enters an idle state until the next command is issued.
5. The Serial I/O and Operating Mode registers are set to zero and DMA data register transfer mode is selected.
6. The device assumes a non-loop SDLC terminal role.

Receive Commands

The 8273 supports three receive commands: General Receive, Selective Receive, and Selective Loop Receive.

General Receive (CMD Code C0)

General receive is a receive mode in which frames are received regardless of the contents of the address field.

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	0	0	0	0
PAR:	0	1	LEAST SIGNIFICANT BYTE OF THE RECEIVE BUFFER LENGTH (B0)							
PAR:	0	1	MOST SIGNIFICANT BYTE OF RECEIVE BUFFER LENGTH (B1)							

NOTES:

1. If buffered mode is specified, the R0, R1 receive frame length (result) is the number of data bytes received.
2. If non-buffered mode is specified, the R0, R1 receive frame length (result) is the number of data bytes received plus two (the count includes the address and control bytes).
3. The frame check sequence (FCS) is not transferred to memory.
4. Frames with less than 32 bits between flags are ignored (no interrupt generated) if the buffered mode is specified.
5. In the non-buffered mode an interrupt is generated when a less than 32 bit frame is received, since data transfer requests have occurred.
6. The 8273 receiver is always disabled when an Idle is received after a valid frame. The CPU module must issue a receive command to re-enable the receiver.
7. The intervening ABORT character between a final flag and an IDLE does not generate an interrupt.
8. If an ABORT Character is not preceded by a flag and is followed by an IDLE, an interrupt will be generated for the ABORT followed by an IDLE interrupt one character time later. The reception of an ABORT will disable the receiver.

Selective Receive (CMD Code C1)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	0	0	0	1
PAR:	0	1	LEAST SIGNIFICANT BYTE OF THE RECEIVE BUFFER LENGTH (B0)							
PAR:	0	1	MOST SIGNIFICANT BYTE OF RECEIVE BUFFER LENGTH (B1)							
PAR:	0	1	RECEIVE FRAME ADDRESS MATCH FIELD ONE (A1)							
PAR:	0	1	RECEIVE FRAME ADDRESS MATCH FIELD TWO (A2)							

Selective receive is a receive mode in which frames are ignored unless the address field matches any one of two address fields given to the 8273 as parameters.

When selective receive is used in HDLC the 8273 looks at the first character, if extended, software must then decide if the message is for this unit.

Selective Loop Receive (CMD Code C2)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	0	0	1	0
PAR:	0	0	LEAST SIGNIFICANT BYTE OF THE RECEIVE BUFFER LENGTH (B0)							
PAR:	0	1	MOST SIGNIFICANT BYTE OF RECEIVE BUFFER LENGTH (B1)							
PAR:	0	1	RECEIVE FRAME ADDRESS MATCH FIELD ONE (A1)							
PAR:	0	1	RECEIVE FRAME ADDRESS MATCH FIELD TWO (A2)							

Selective loop receive operates like selective receive except that the transmitter is placed in flag stream mode automatically after detecting an EOP (01111111) following a valid received frame. The one bit delay mode is also reset at the end of a selective loop receive.

Receive Disable (CMD Code C5)

Terminates an active receive command immediately.

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	0	1	0	1
PAR:	NONE									

MPU PERIPHERALS

Transmit Commands

The 8273 supports three transmit commands: Transmit Frame, Loop Transmit, Transmit Transparent.

Transmit Frame (CMD Code C8)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	1	0	0	0
PAR:	0	1	LEAST SIGNIFICANT BYTE OF FRAME LENGTH (L0)							
PAR:	0	1	MOST SIGNIFICANT BYTE OF FRAME LENGTH (L1)							
PAR:	0	1	ADDRESS FIELD OF TRANSMIT FRAME (A)							
PAR:	0	1	CONTROL FIELD OF TRANSMIT FRAME (C)							

Transmits one frame including: initial flag, frame check sequence, and the final flag.

If the buffered mode is specified, the L0, L1, frame length provided as a parameter is the length of the information field and the address and control fields must be input.

In unbuffered mode the frame length provided must be the length of the information field plus two and the address and control fields must be the first two bytes of data. Thus only the frame length bytes are required as parameters.

Loop Transmit (CMD Code CA)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	1	0	1	0
PAR:	0	1	LEAST SIGNIFICANT BYTE OF FRAME LENGTH (L0)							
PAR:	0	1	MOST SIGNIFICANT BYTE OF FRAME LENGTH (L1)							
PAR:	0	1	ADDRESS FIELD OF TRANSMIT FRAME (A)							
PAR:	0	1	CONTROL FIELD OF TRANSMIT FRAME (C)							

Transmits one frame in the same manner as the transmit frame command except:

1. This command should be given only in one-bit delay mode.
2. If the flag stream mode is not active transmission will begin after a received EOP has been converted to a flag.
3. If the flag stream mode is active transmission will begin at the next flag boundary for buffered mode or at the third flag boundary for non-buffered mode.
4. At the end of a loop transmit the one-bit delay mode is entered and the flag stream mode is reset.

Transmit Transparent (CMD Coded C9)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	1	0	0	1
PAR:	0	1	LEAST SIGNIFICANT BYTE OF FRAME LENGTH (L0)							
PAR:	0	1	MOST SIGNIFICANT BYTE OF FRAME LENGTH (L1)							

The 8273 will transmit a block of raw data without protocol, i.e., no zero bit insertion, flags, or frame check sequences.

Abort Transmit Commands

An abort command is supported for each type of transmit command. The abort commands are ignored if a transmit command is not in progress.

Abort Transmit Frame (CMD Code CC)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	1	1	0	0

PAR: NONE

After an abort character (eight contiguous ones) is transmitted, the transmitter reverts to sending flags or idles as a function of the flag stream mode specified.

Abort Loop Transmit (CMD Code CE)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	1	1	1	0

PAR: NONE

After a flag is transmitted the transmitter reverts to one bit delay mode.

Abort Transmit Transparent (CMD Code CD)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	1	1	0	0	1	1	0	1

PAR: NONE

The transmitter reverts to sending flags or idles as a function of the flag stream mode specified.

MPU PERIPHERALS

Modem Control Commands

The modem control commands are used to manipulate the modem control ports.

When read Port A or Port B commands are executed the result of the command is returned in the result register. The Bit Set Port B command requires a parameter that is a mask that corresponds to the bits to be set. The Bit Reset Port B command requires a mask that has a zero in the bit positions that are to be reset.

Read Port A (CMD Code 22)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	0	0	1	0	0	0	1	0

PAR: NONE

Read Port B (CMD Code 23)

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	0	0	1	0	0	0	1	1

PAR: NONE

Set Port B Bits (CMD Code A3)

This command allows user defined Port B pins to be set.

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	0	1	1	0	0	0	1	1
PAR:	0	1	0	0						

RTS – REQUEST TO SEND
USER DEFINED
FLAG DETECT

NB SET MASK: 1

(D5) Flag Detect

This bit can be used to set the flag detect pin. However, it will be reset when the next flag is detected.

(D_4-D_1) User Defined Outputs

These bits correspond to the state of the PB_4-PB_1 output pins.

(D_0) Request to Send

This is a dedicated 8273 modem control signal, and reflects the same logical state of RTS pin.

Reset Port B Bits (CMD Code 63)

This command allows Port B user defined bits to be reset.

	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CMD:	0	0	0	1	1	0	0	0	1	1
PAR:	0	1	1	1						

RTS – REQUEST TO SEND
USER DEFINED
FLAG DETECT

This command allows Port B (D_4-D_1) user defined bits to be reset. These bits correspond to Output Port pins (PB_4-PB_1).

NB! Reset MASK: 0

8273 Command Summary

Command Description	Command (HEX)	Parameter	Results	Result Port	Completion Interrupt
Set One Bit Delay	A4	Set Mask	None	—	No
Reset One Bit Delay	64	Reset Mask	None	—	No
Set Data Transfer Mode	97	Set Mask	None	—	No
Reset Data Transfer Mode	57	Reset Mask	None	—	No
Set Operating Mode	91	Set Mask	None	—	No
Reset Operating Mode	51	Reset Mask	None	—	No
Set Serial I/O Mode	A0	Set Mask	None	—	No
Reset Serial I/O Mode	60	Reset Mask	None	—	No
General Receive	C0	B0,B1	IC,R0,R1,A,C	RXI/R ***	Yes
Selective Receive	C1	B0,B1,A1,A2	IC,R0,R1,A,C	RXI/R	Yes
Selective Loop Receive	C2	B0,B1,A1,A2	IC,R0,R1,A,C	RXI/R	Yes
Receive Disable	C5	None	None	—	No
Transmit Frame	C8	L0,L1,A,C	IC	TXI/R **	Yes
Loop Transmit	CA	L0,L1,A,C	IC	TXI/R	Yes
Transmit Transparent	C9	L0,L1	IC	TXI/R	Yes
Abort Transmit Frame	CC	None	IC	TXI/R	Yes
Abort Loop Transmit	CE	None	IC	TXI/R	Yes
Abort Transmit Transparent	CD	None	IC	TXI/R	Yes
Read Port A	22	None	Port Value	Result *	No
Read Port B	23	None	Port Value	Result	No
Set Port B Bit	A3	Set Mask	None	—	No
Reset Port B Bit	63	Reset Mask	None	—	No

*/ BASE+1, **/ BASE+2, ***/ BASE+3

8273 Command Summary Key

B0 — Least significant byte of the receive buffer length.
B1 — Most significant byte of the receive buffer length.
L0 — Least significant byte of the Tx frame length.
L1 — Most significant byte of the Tx frame length.
A1 — Receive frame address match field one.
A2 — Receive frame address match field two.
A — Address field of received frame. If non-buffered mode is specified, this result is not provided.
C — Control field of received frame. If non-buffered mode is specified this result is not provided.
RXI/R — Receive interrupt result register.
TXI/R — Transmit interrupt result register.
R0 — Least significant byte of the length of the frame received.
R1 — Most significant byte of the length of the frame received.
IC — Interrupt result code (see table).

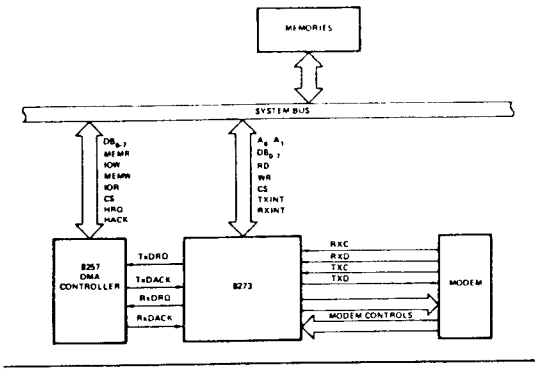


Figure 14. 8273 System Diagram

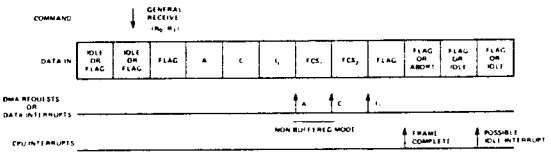


Figure 12. Typical Frame Reception

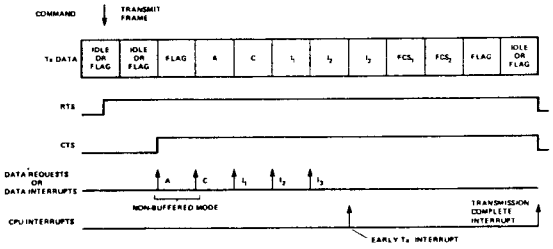


Figure 13. Typical Frame Transmission

MPU PERIPHERALS

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin With Respect to Ground	-0.5V to +7V
Power Dissipation	1 Watt

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5.0V ±5%

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	Volts	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	Volts	
V_{OL}	Output Low Voltage		0.45	Volts	I_{OL} = 2.0mA
V_{OH}	Output High Voltage	2.4		Volts	I_{OH} = -200μA
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OZ}	Off-State Output Current		±10	μA	V_{OUT} = V_{CC} to 0V
I_{CC}	V_{CC} Supply Current		160	mA	

CAPACITANCE

T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	t_c = 1MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured Pins Returned to GND

A1.21

A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5.0V ±5%

Read Cycle

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AC}	Select Setup to $\overline{RD}$	0		ns	
t_{CA}	Select Hold from $\overline{RD}$	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	250		ns	
t_{AD}	Data Delay from Address		200	ns	
t_{RD}	Data Delay from $\overline{RD}$		150	ns	C_L = 150pF
t_{DF}	Output Float Delay	20	100	ns	C_L = 20pF for Minimum; 150pF for Maximum

Write Cycle

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AC}	Select Setup to $\overline{WR}$	0		ns	
t_{CA}	Select Hold from $\overline{WR}$	0		ns	
t_{WW}	$\overline{WR}$ Pulse Width	250		ns	
t_{DW}	Data Setup to $\overline{WR}$	150		ns	
t_{WD}	Data Hold from $\overline{WR}$	-20		ns	

DMA

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CQ}	Request Hold from $\overline{WR}$ or $\overline{RD}$ (for Non-Burst Mode)		150	ns	

Other Timing

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{RSTW}	Reset Pulse Width	10		t_{CY}	
t_r	Input Signal Rise Time		20	ns	
t_f	Input Signal Fall Time		20	ns	
t_{RSTS}	Reset to First $\overline{IOWR}$	2		t_{CY}	
t_{CY}	Clock	250			Note 3
t_{CL}	Clock Low	T_{BS}			Note 2
t_{CH}	Clock High	T_{BS}			Note 2
t_{DCL}	Data Clock Low				
t_{DCH}	Data Clock High	200		ns	
t_{DCY}	Data Clock	15625		ns	Note 3
t_{TD}	Transmit Data Delay		100	ns	
t_{DS}	Data Setup Time	100		ns	
t_{DH}	Data Hold Time	0		ns	
t_{DPLL}	$\overline{DPLL}$ Output Low	200		ns	
t_{FLD}	$\overline{FLAG\ DET}$ Output Low	$8 \cdot t_{CY} \pm 50$		ns	

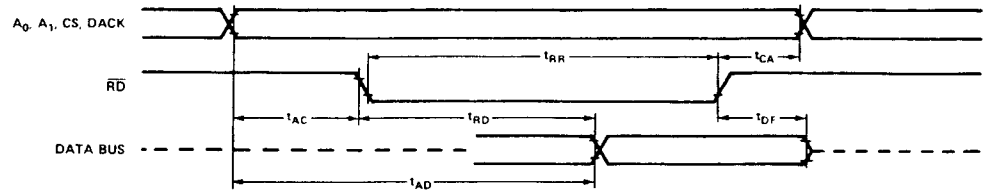
MPU PERIPHERALS

NOTES:

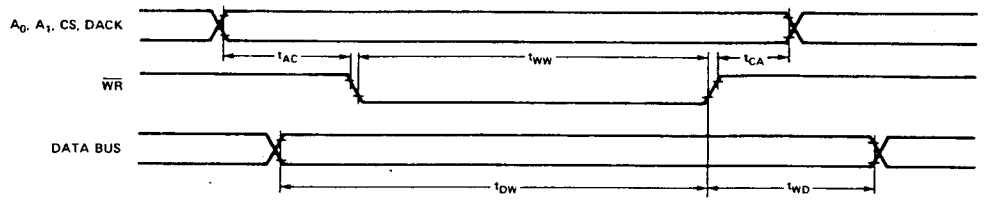
1. All timing measurements are made at the reference voltages unless otherwise specified:
   Input "1" at 2.0V, "0" at 0.8V
   Output "1" at 2.0V, "0" at 0.8V
2. To be specified.
3. 64K baud maximum operating rate.

WAVEFORMS

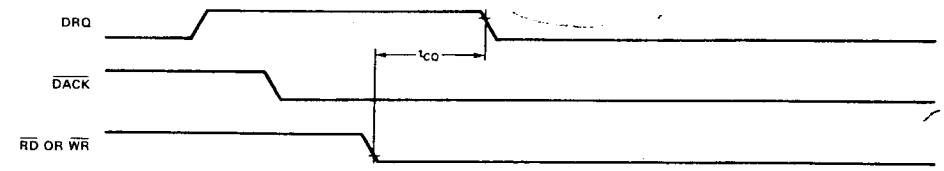
Read Waveforms

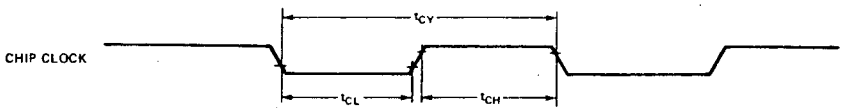


Write Waveforms

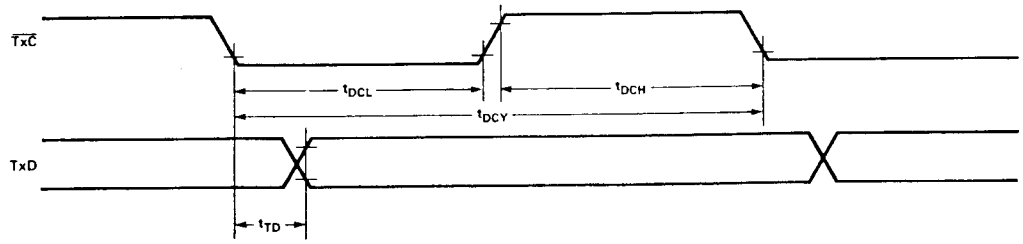


DMA Waveforms

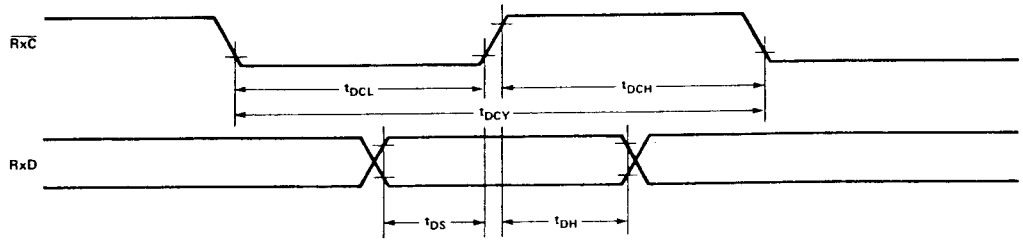




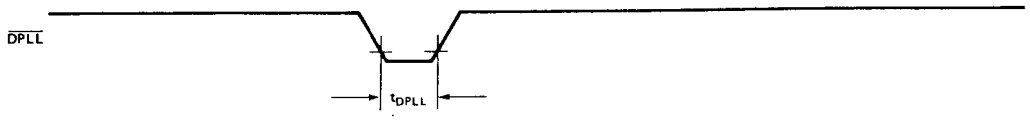
Transmit Data Waveforms



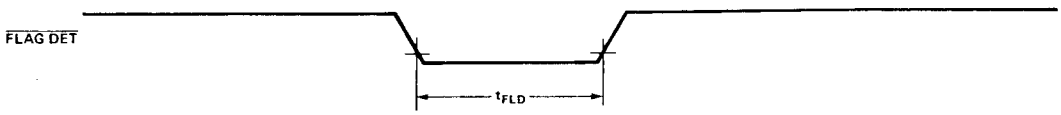
Receive Data Waveforms

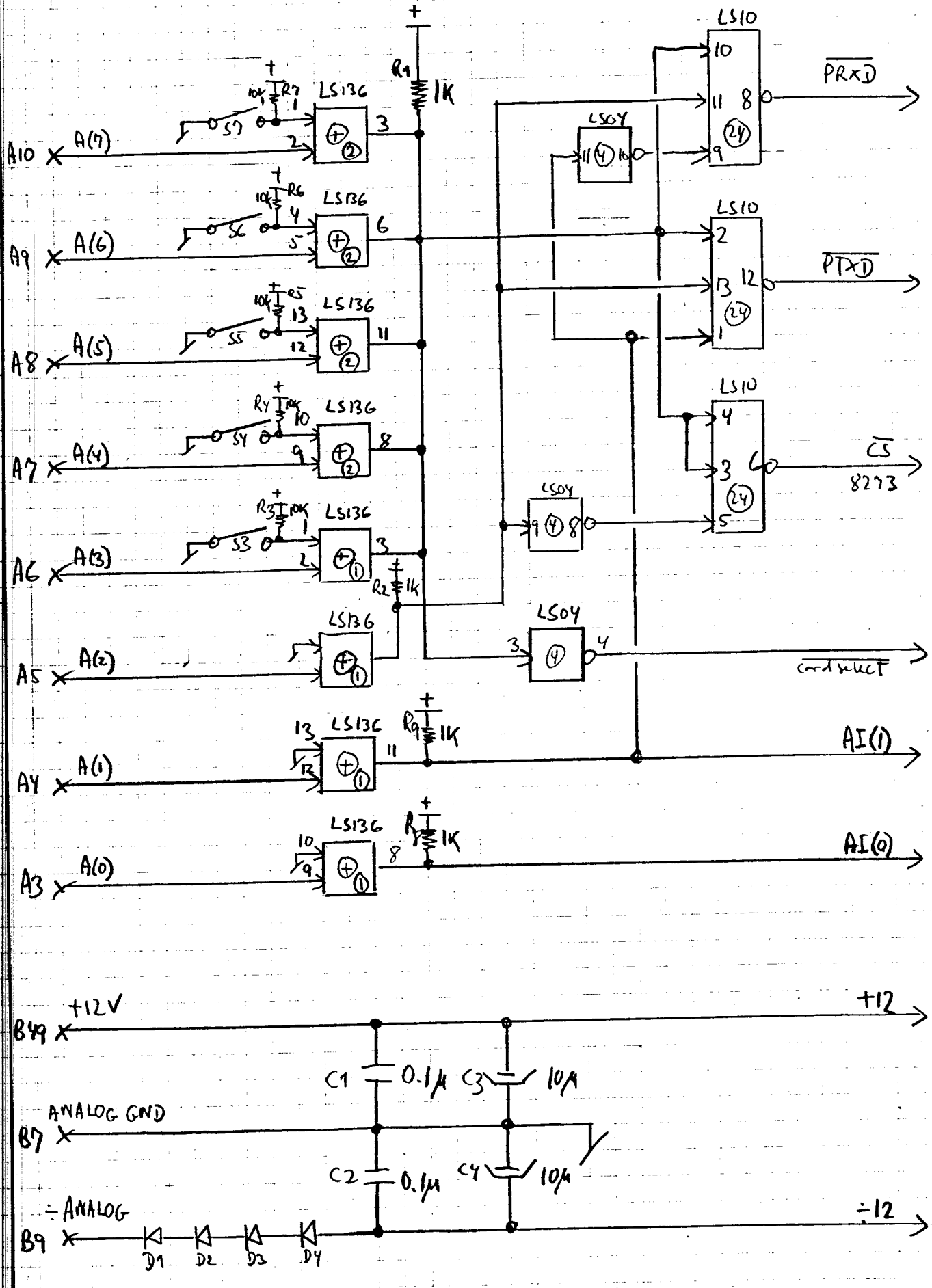


DPLL Output Waveform



Flag Detect Output Waveform



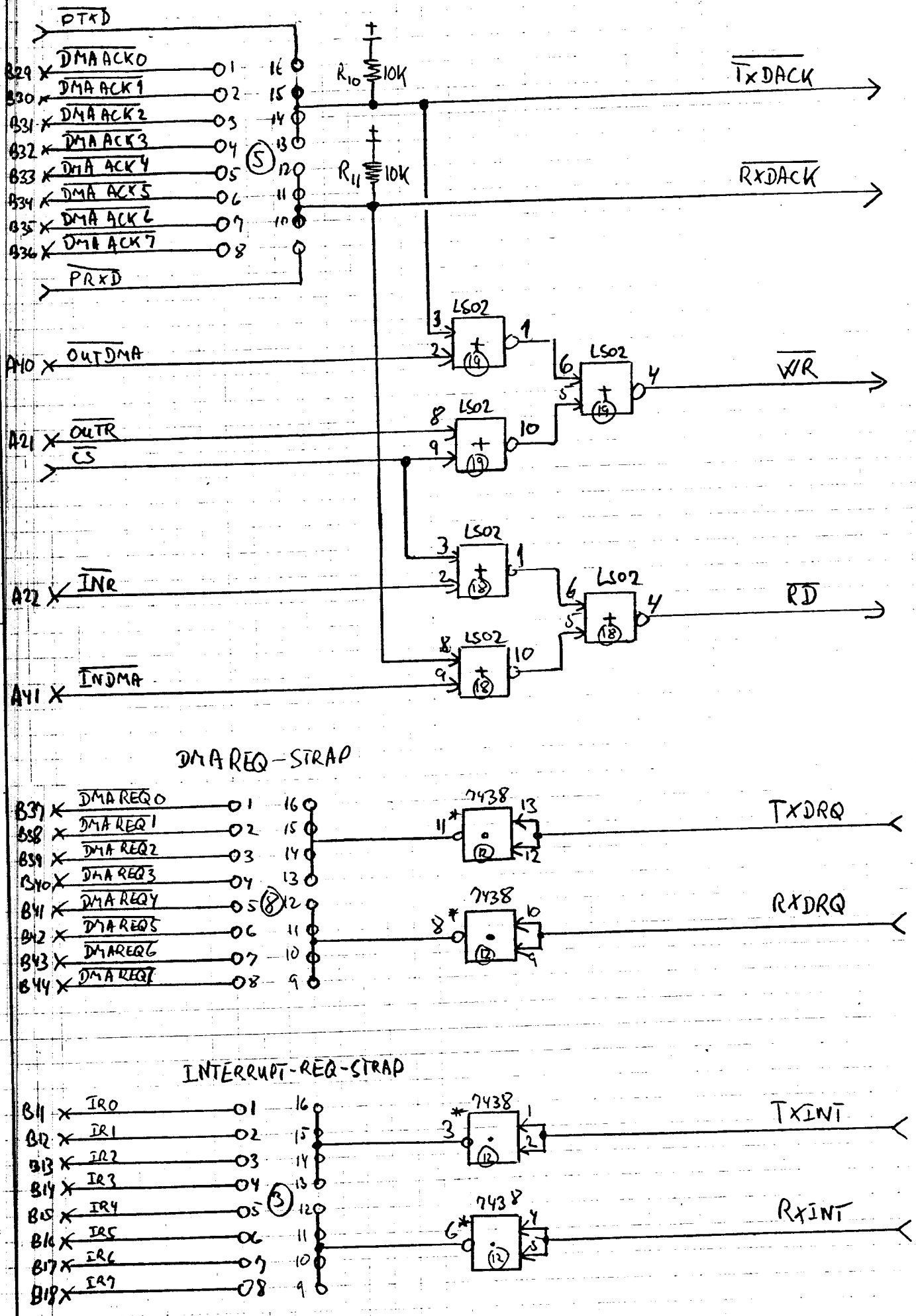

A10
A9
A8
A7
A6
A5
A4
A3
A(7)
A(6)
A(5)
A(4)
A(3)
A(2)
A(1)
A(0)
LS136
LS10
LS04
R1 1K
R2 1K
R3 10K
R4 10K
R5 10K
R6 10K
R7 10K
S1
S2
S3
S4
S5
PRXD
PTXD
CS
8273
card select
AI(1)
AI(0)
+12V
+12
B49
C1 0.1µ
C3 10µ
ANALOG GND
B7
C2 0.1µ
C4 10µ
÷ANALOG
÷12
B9
D1
D2
D3
D4

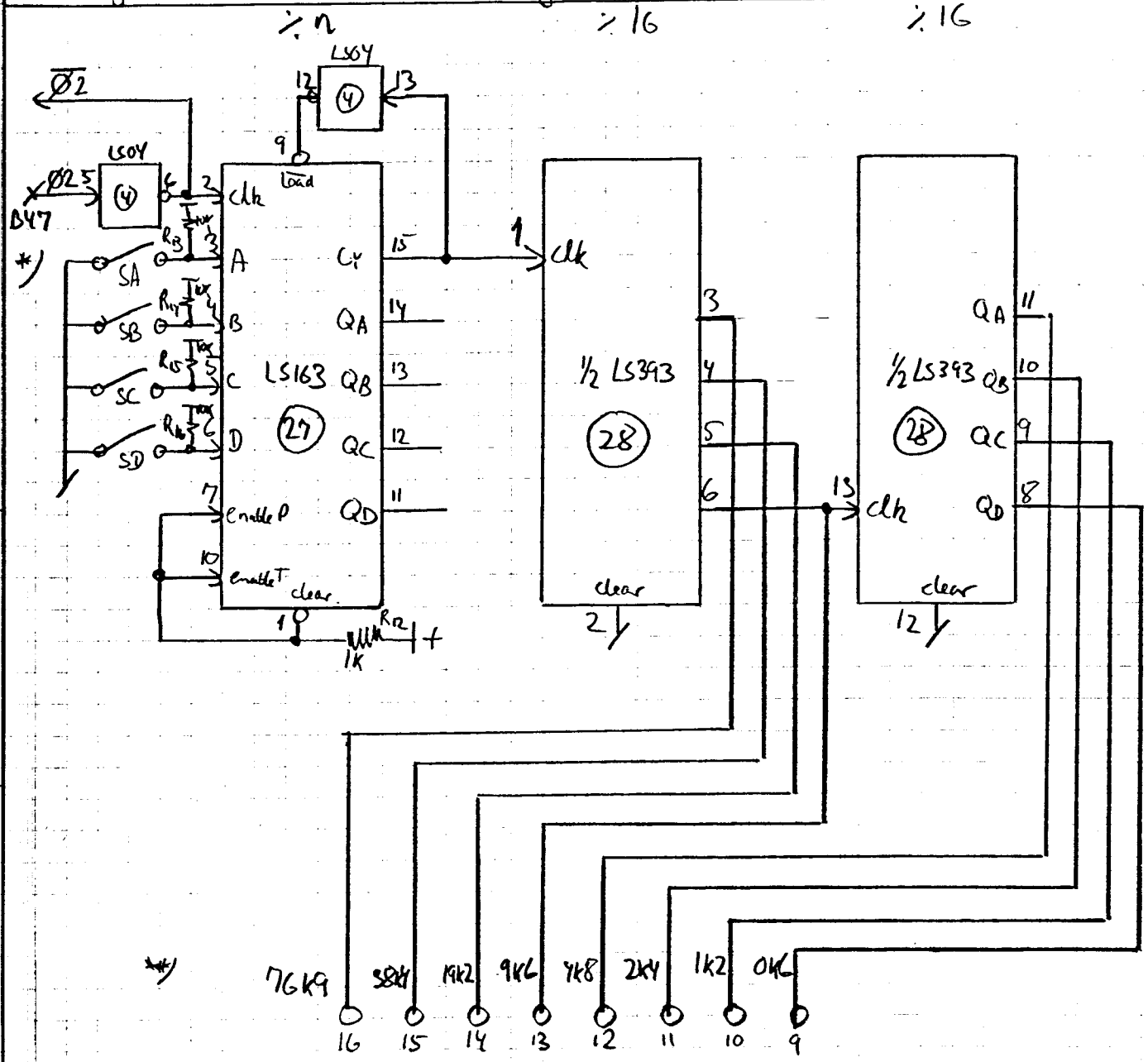
dansk data elektronik aps

ID-7047 HDLC-communication module.

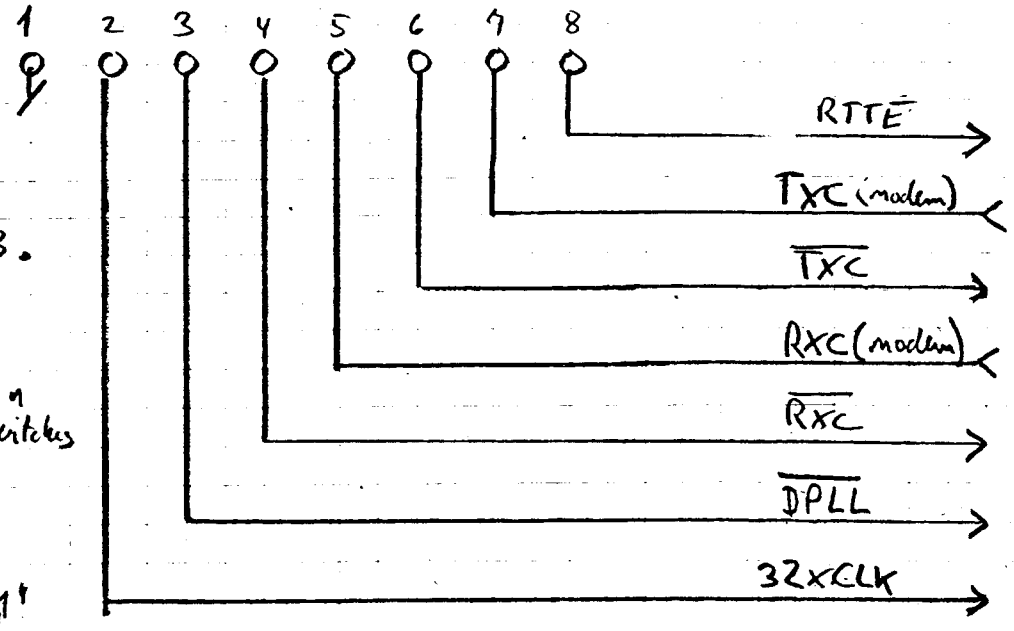
Initialer/dato	Side
OL 1-10-78	A 2.2
Revideret	Projekt

Logic schematic: Read/Write Logic, Interrupt logic.





CLOCK-STRAP: (29)

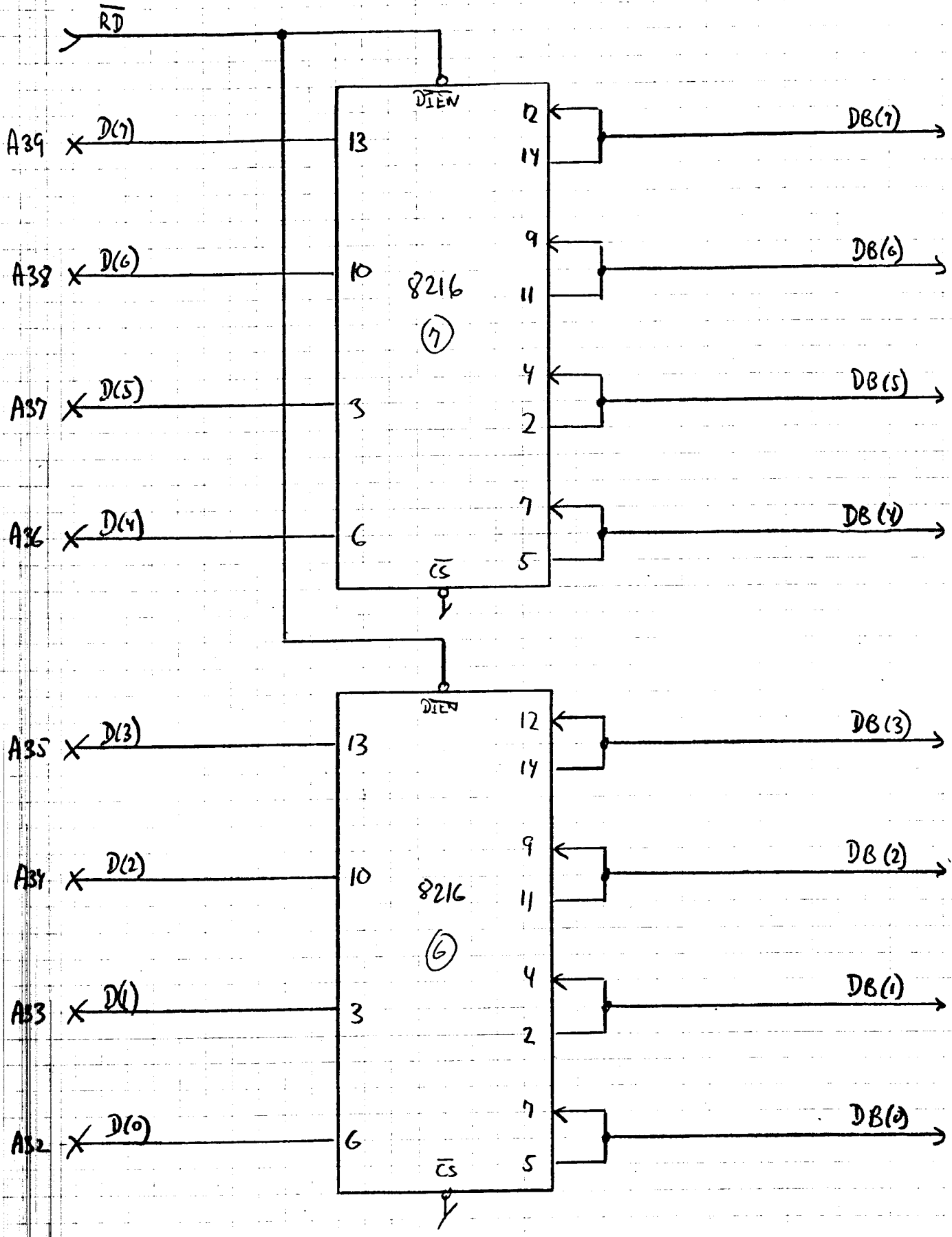


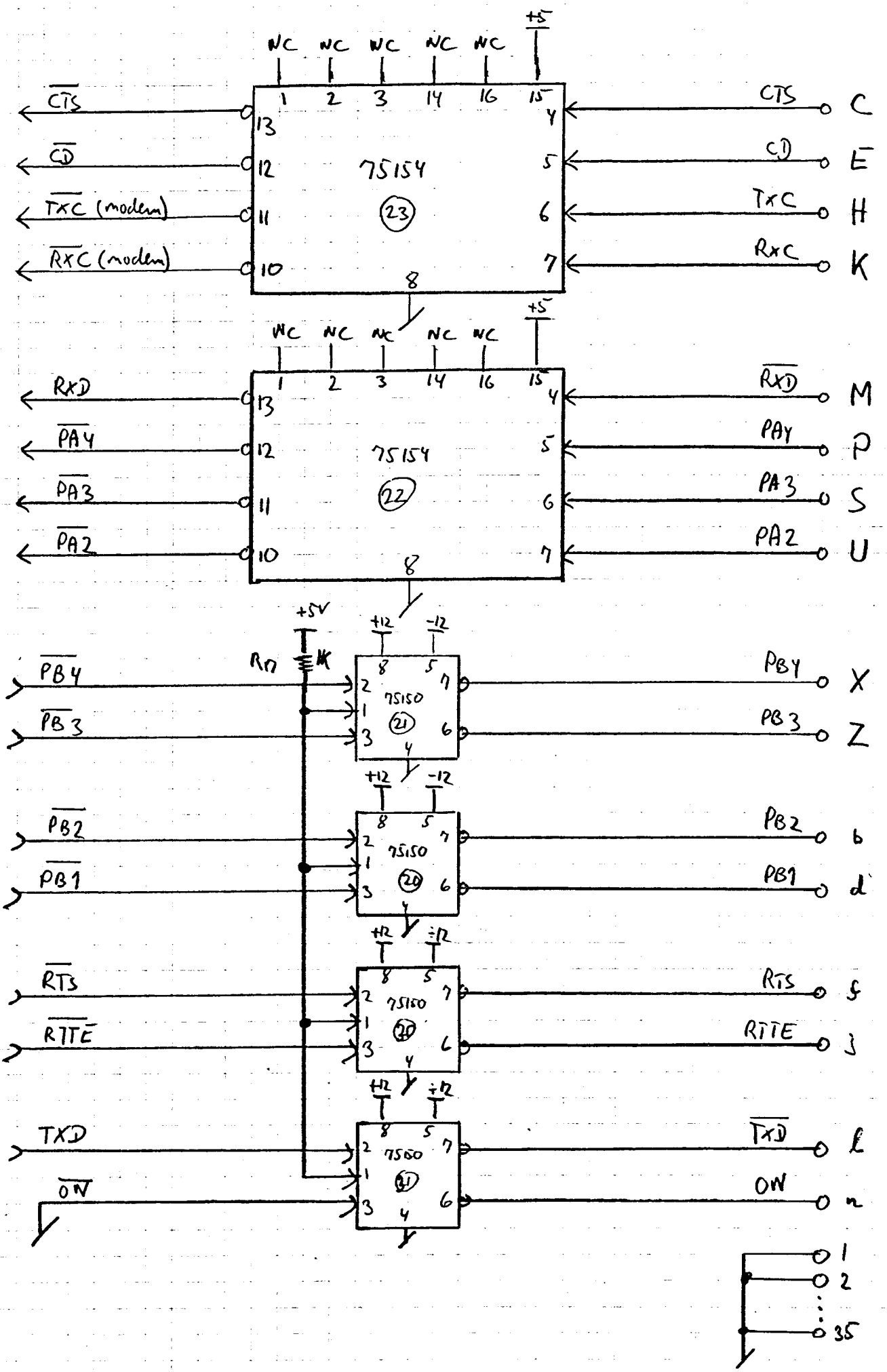
**) shown baud rates corresponds to n=13.

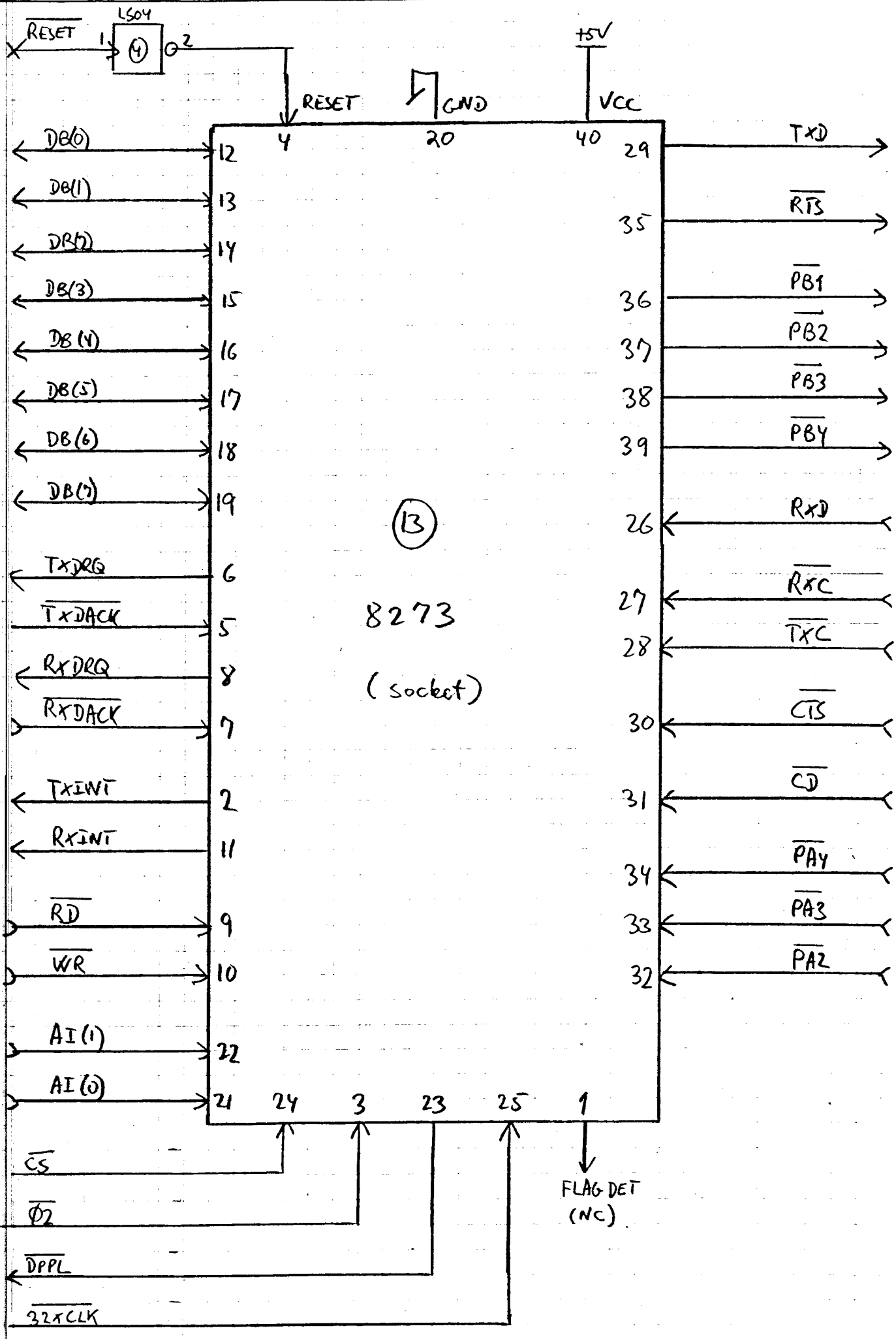
*) The divide ratio n is determined by switches SD, SC, SB, SA:

$SD\,SC\,SB\,SA = 16 - n$

open switch (white) = "1"
closed switch (red) = "0"


RD
DIEN
A39
D(7)
13
12
14
DB(7)
A38
D(6)
10
9
11
DB(6)
8216
(7)
A37
D(5)
3
4
2
DB(5)
A36
D(4)
6
7
5
DB(4)
CS
DIEN
A35
D(3)
13
12
14
DB(3)
A34
D(2)
10
9
11
DB(2)
8216
(6)
A33
D(1)
3
4
2
DB(1)
A32
D(0)
6
7
5
DB(0)
CS


NC NC NC NC NC +5
1 2 3 14 16 15
CTS
CD
TxC (modem)
RxC (modem)
13 12 11 10
75154
(23)
4 5 6 7
CTS C
CD E
TxC H
RxC K
8
NC NC NC NC NC +5
1 2 3 14 16 15
RxD
PA4
PA3
PA2
13 12 11 10
75154
(22)
4 5 6 7
RxD M
PA4 P
PA3 S
PA2 U
8
+5V
R1
1K
+12 -12
PB4
PB3
2 1 3
75150
(21)
8 5 7 6 4
PB4 X
PB3 Z
PB2
PB1
(20)
PB2 b
PB1 d
RTS
RTTE
(19)
RTS f
RTTE j
TXD
ON
(18)
TxD L
ON n
1
2
35

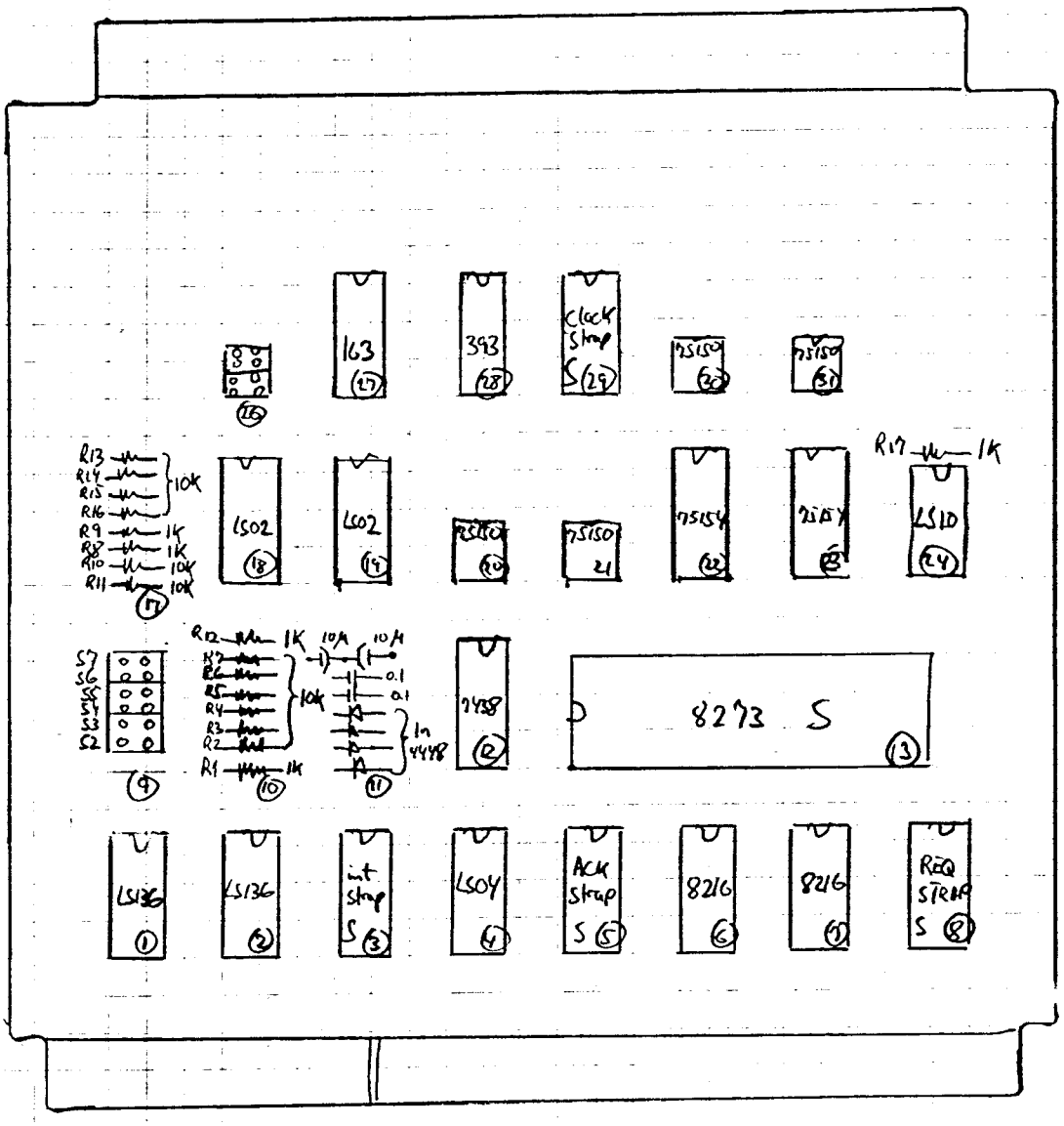

RESET
LS04
RESET
GND
+5V
VCC
4
20
40
DB(0)
DB(1)
DB(2)
DB(3)
DB(4)
DB(5)
DB(6)
DB(7)
12
13
14
15
16
17
18
19
TXDRQ
TXDACK
RXDRQ
RXDACK
TXINT
RXINT
RD
WR
AI(1)
AI(0)
6
5
8
7
2
11
9
10
22
21
24
3
23
25
1
CS
Φ2
DPPL
32xCLK
FLAG DET
(NC)
TXD
RTS
PB1
PB2
PB3
PB4
RXD
RXC
TXC
CTS
CD
PA4
PA3
PA2
29
35
36
37
38
39
26
27
28
30
31
34
33
32
13
8273
(socket)

dansk data elektronik aps

ID-7047 HDLC-communication module

Component placement.

Initialer/dato	Side
OL 1-10-78	A 2.7
Revideret	Projekt



```
0001 0000 ;TESTPROGRAM FOR ID-7047 HDLC/SDLC-CONTROLLER.
0002 0000 ;*****************************************************
0003 0000 ;
0004 0000 ;TESTPROGRAM USES ID-7047 IN A NON-DMA AND NON-INTERRUPT MODE.
0005 0000 ;
0006 0000 ;BASEADRESS FOR MODULE IS SET TO X.00 (ADDRESS SWITCHES OFF.)
0007 0000 ;
0008 0000 ;THE DIVIDE RATIO IN THE BAUD RATE CIRCUITS IS SET TO 13
0009 0000 ;(SD,SC OFF - SB,SA ON)
0010 0000 ;
0011 0000 ;A BAUD RATE STRAP WITH THE FOLLOWING CONNECTIONS ARE PLACED
0012 0000 ;IN STRAP POSITION 29:
0013 0000 ;
0014 0000 ;                2 - 1    :        32XCLK - GND
0015 0000 ;                5 - 4    :        RXC(MODEM) - RXC
0016 0000 ;                7 - 6    :        TXC(MODEM) - TXC
0017 0000 ;               13 - 8    :        9600 - RTTE
0018 0000 ;
0019 0000 ;A STRAP WITH THE FOLLOWING CONNECTIONS ARE PLACED IN STRAP
0020 0000 ;POSITION 5:
0021 0000 ;
0022 0000 ;               16 - 15   :        PTXD - TXDACK
0023 0000 ;                9 - 10   :        PRXD - RXDACK
0024 0000 ;
0025 0000 ;
0026 0000 ;
0027 0000 ;A TOP CONNECTOR WITH FOLLOWING CONNECTIONS ARE INSTALLED:
0028 0000 ;
0029 0000 ;                n - C - E          ON - CTS - CD
0030 0000 ;                J - H - K      RTTE - TXC - RXC
0031 0000 ;                l - M            TXD - RXD
0032 0000 ;                X - P            PB4 - PA4
0033 0000 ;                Z - S            PB3 - PA3
0034 0000 ;                b - U            PB2 - PA2
0035 0000 ;
0036 0000 ;
0037 0000 ;IMPORTANT NOTICE:  THE TEST PROGRAM RETURNS TO DEBUGGER
0038 0000 ;IN CASE OF ERRRORS, A DEBUGGER SHOULD BE PRE-
0039 0000 ;SENT IN THE COMPUTER BEFORE TEST PROGRAM IS STARTED.
0040 0000 BASE:   EQU     0          ;BASEADRESS FOR MODULE
0041 0000 ;
0042 0000 STAT:   EQU     BASE+0     ;STATUSREGISTER ADDRESS
0043 0000 COM:    EQU     BASE+0     ;COMMAND ADDRESS
0044 0000 RES:    EQU     BASE+1     ;RESULT ADDRESS
0045 0000 PAR:    EQU     BASE+1     ;PARAMETER ADDRESS
0046 0000 TXIR:   EQU     BASE+2     ;TXINTRESULT ADDRESS
0047 0000 TSTM:   EQU     BASE+2     ;TESTMODE ADDRESS
0048 0000 RXIR:   EQU     BASE+3     ;RXINTRESULT ADDRESS
0049 0000 PRXD:   EQU     BASE+4     ;PROGRAMMED RXDATA ADDRESS
0050 0000 PTXD:   EQU     BASE+6     ;PROGRAMMED TXDATA ADDRESS
0051 0000 ;
0052 0000 LSBBL:  EQU     00         ;RX BUFFERLENGTH LSB
0053 0000 MSBBL:  EQU     30         ;RX BUFFERLENGTH MSB
0054 0000 ;
0055 0000 LSBFL:  EQU     00         ;TX FRAMELENGTH LSB - EQUAL TO LSBBL.
0056 0000 MSBFL:  EQU     30         ;TX FRAMELENGTH MSB - EQUAL TO MSBBL.
0057 0000 ;
0058 0000 TXBS:   EQU      0         ;TXBUFFER START IN MEMORY.
0059 0000 ;
0060 0000 ;
0061 0000 HDLC:   DI
0062 0001         MVI     A,1        ;RESET HDLC-CONTROLLER.
0063 0003         OUT     TSTM
0064 0005         MVI     A,0
```

```
0065 0007          OUT     TSTM
0066 0009 ;
0067 0009 ;
0068 0009          MVI     C,91     ;SET OPERATING MODE:
0069 000B          MVI     B,25     ;HDLC, BUFF-MODE,FLAG STREAM.
0070 000D          CALL    COMP     ;IMPORTANT: THIS COMMAND MUST BE
0071 0010                           ;ISSUED AS THE FIRST ONE.
0072 0010 ;
0073 0010 ;TEST OF USER DEFINED MODEM INTERFACE (EXCL. PB1-OUTPUT)
0074 0010 ;
0075 0010          MVI     C,063    ;RESET ALL PORT B-BITS
0076 0012          MVI     B,0C0    ;INCLUDING PB4,PB3,PB2,PB1.
0077 0014          CALL    COMP
0078 0017 M02:     IN      STAT     ;WAIT WHILE COMMAND BUSY.
0079 0019          ANI     80
0080 001B          JNZ     M02
0081 001E          MVI     A,22     ;SEND READ PORT A COMMAND.
0082 0020          OUT     COM
0083 0022 M03:     IN      STAT     ;WAIT WHILE COMMAND BUSY.
0084 0024          ANI     80
0085 0026          JNZ     M03
0086 0029          IN      RES      ;GET RESULT REGISTER
0087 002B          CPI     0E3      ;TEST PA4,PA3,PA2=0 - CD,CTS=1
0088 002D          CNZ     0F008    ;IF NOT JUMP TO DEBUGGER.
0089 0030 ;
0090 0030          MVI     C,0A3    ;SET ALL PORT B-BITS
0091 0032          MVI     B,03F    ;INCLUDING PB4,PB3,PB2,PB1
0092 0034          CALL    COMP
0093 0037 M04:     IN      STAT     ;WAIT WHILE COMMAND BUSY
0094 0039          ANI     80
0095 003B          JNZ     M04
0096 003E          MVI     A,22     ;SEND READ PORT A COMMAND.
0097 0040          OUT     COM
0098 0042 M05:     IN      STAT     ;WAIT WHILE COMMAND BUSY.
0099 0044          ANI     80
0100 0046          JNZ     M05
0101 0049          IN      RES      ;GET RESULT REGISTER
0102 004B          CPI     0FF      ;TEST PA4,PA3,PA2,CD,CTS=1
0103 004D          CNZ     0F008    ;IF NOT JUMP TO DEBUGGER.
0104 0050 ;
0105 0050 ;NOW DATATRANSFER IS INITIALIZED:
0106 0050 ;
0107 0050 ;
0108 0050          MVI     C,97     ;SET DATATRANSFER MODE (NON DMA)
0109 0052          MVI     B,1
0110 0054          CALL    COMP
0111 0057 ;
0112 0057 ;
0113 0057          MVI     C,0C0    ;GENERAL RECEIVE COMMAND.
0114 0059          MVI     B,LSBBL  ;LSB-BUFFER LENGTH.
0115 005B          CALL    COMP
0116 005E          MVI     B,MSBBL  ;MSB-BUFFER LENGTH.
0117 0060          CALL    COMP1
0118 0063 ;
0119 0063 ;
0120 0063          MVI     C,0C8    ;TRANSMIT FRAME COMMAND.
0121 0065          MVI     B,LSBFL  ;LSB-FRAME LENGTH.
0122 0067          CALL    COMP
0123 006A          MVI     B,MSBFL  ;MSB-FRAME LENGTH
0124 006C          CALL    COMP1
0125 006F          MVI     B,0AA    ;ADDRESS-FIELD
0126 0071          CALL    COMP1
0127 0074          MVI     B,55     ;CONTROL-FIELD
0128 0076          CALL    COMP1
```

```
0129 0079 ;
0130 0079 ;
0131 0079 ;END INITIALIZATION.NOW DATA TRANSFER IS STARTED:
0132 0079 ;
0133 0079 ;
0134 0079         LXI     H,TXBS  ;TXBUFFER START IN MEMORY.
0135 007C         LXI     D,RXBS  ;RXBUFFER START IN MEMORY.
0136 007F ;
0137 007F TSTAT:  IN      STAT    ;TEST RXIRA AND TXIRA
0138 0081         ANI     3       ;(FRAME TX AND RX CONCLUDED).
0139 0083         CPI     3
0140 0085         JZ      SLUT    ;CONCLUDED - JUMP
0141 0088 ;
0142 0088         IN      STAT    ;TEST TXINT
0143 008A         ANI     4
0144 008C         CNZ     TXCH    ;TXREADY, SEND CHARACTER.
0145 008F ;
0146 008F         IN      STAT    ;TEST RXINT
0147 0091         ANI     8
0148 0093         CNZ     RXCH    ;RECEIVER READY, GET CHARACTER.
0149 0096 ;
0150 0096         JMP     TSTAT
0151 0099 ;
0152 0099 ;
0153 0099 SLUT:   LXI     H,TXBS
0154 009C         LXI     D,RXBS
0155 009F         MVI     C,LSBFL
0156 00A1         MVI     B,MSBFL
0157 00A3 SLUT1:  LDAX    D       ;COMPARE TX AND RX MEMORY BUFFERS.
0158 00A4         CMP     M
0159 00A5         CNZ     0F008   ;CALL DEBUG IF ERROR.
0160 00A8         INX     H
0161 00A9         INX     D
0162 00AA         DCX     B
0163 00AB         MOV     A,C
0164 00AC         ORA     B
0165 00AD         JNZ     SLUT1
0166 00B0 ;
0167 00B0         IN      TXIR    ;GET TXINT-RESULT
0168 00B2         MOV     B,A     ;SAVE IN B-REGISTER.
0169 00B3         IN      RXIR    ;GET RXINT-RESULT
0170 00B5         MOV     C,A     ;SAVE IN C-REGISTER.
0171 00B6         MVI     A,0D    ;TEST TXINTERRUPT RESULT
0172 00B8         CMP     B
0173 00B9         CNZ     0F008   ;NOT OK, CALL DEBUG.
0174 00BC         MVI     A,0E0   ;TEST RXINTERRUPT RESULT
0175 00BE         CMP     C
0176 00BF         CNZ     0F008   ;NOT OK, CALL DEBUG.
0177 00C2         JMP     HDLC    ;OK, START TESTPROGRAMMET FORFRA.
0178 00C5 ;
0179 00C5 ;
0180 00C5 ;SUBROUTINE TXCH:
0181 00C5 ;
0182 00C5 TXCH:   MOV     A,M
0183 00C6         OUT     PTXD
0184 00C8         INX     H
0185 00C9         RET
0186 00CA ;
0187 00CA ;
0188 00CA ;SUBROUTINE RXCH:
0189 00CA ;
0190 00CA RXCH:   IN      PRXD
0191 00CC         STAX    D
0192 00CD         INX     D
```

```
0193 00CE            RET
0194 00CF ;
0195 00CF ;
0196 00CF ;SUBROUTINE COMP, COMP1:
0197 00CF ;(LOADS COMMANDS AND PARAMETERS).
0198 00CF ;
0199 00CF COMP:   IN       STAT     ;TEST COMMAND BUSY
0200 00D1         ANI      80
0201 00D3         JNZ      COMP
0202 00D6         MOV      A,C
0203 00D7         OUT      COM
0204 00D9 COMP1:  IN       STAT
0205 00DB         ANI      20       ;TEST COMMAND PAR-BUFFER FULL.
0206 00DD         JNZ      COMP1
0207 00E0         MOV      A,B
0208 00E1         OUT      PAR
0209 00E3         RET
0210 00E4 ;
0211 00E4 ;
0212 00E4 ;
0213 00E4 RXBS:   DS       3000
0214 30E4 ;
0215 30E4 ;
0216 30E4         END      HDLC
```

LINKING OF "HDLC   " 0001

MODULE LIST.

XECUTION STARTS IN ADDRESS 4000

SYMBOL	ABSOLUTE ADDRESS	TAG	ROM BASE	RAM BASE
DLC			4000	70E4

TARTING ADDRESS 4000
NDING ADDRESS 70E4