

HI-8282 Overview

The HI-8282 is a silicon gate device for interfacing the ARINC 429 serial data bus to a 16-bit parallel data bus. It contains two receivers and an independent transmitter. The receiver input circuitry and logic are designed to meet the ARINC 429 specifications for loading, level detection, timing, and protocol. The transmitter section provides the ARINC 429 communication protocol. The 5 Volt logic outputs are translated to ARINC 429 drive levels by one of Holt's companion bus drivers (HI-3182, HI-8585 or HI-8586). The HI-8282 also contains an internal control register that is used for programming the data rate, parity type, source-to-destination identifier, and to enable or disable internal self test mode. The timing requirements of the device make it compatible with 8086 microprocessors or similarly based systems operating at 5 MHz or less. The high reliability of the HI-8282 enables it to meet the narrow error margin tolerances of today's advanced avionics.

Device Operation

The HI-8282 is a true ARINC 429 compatible device; that is, it was designed to overcome the deficiencies of some of the older standard devices of this type.

This application note is designed to highlight and clarify the details of these improvements, as well as to provide a succinct explanation of operations of the device.

Master Clock Frequency

The HI-8282 data sheet provides information on the correct operation of the device with a 1 MHz master clock (CLK). Since the device uses sampling-based logic instead of edge sensitive logic in the receiver circuitry and the fact that all internally generated clocks are proportional to the master clock, it is possible to operate the HI-8282 at frequencies lower than 1 MHz. For example, if a 500 KHz master clock is used, then all data clocks, sampling clocks and the bit rate will be cut in half; while the pulse widths will double.

The 16-Bit Data Bus

A 16-bit bidirectional, tristate data bus is used to transfer data to and from the host. The control of this bus is completely handled internally by the HI-8282. The bus is automatically enabled as an output when data is read from the receivers and as an input when data is written to the transmitter or control word register.

The Control Word Register

The control word register is 10 bits in length and allows for the programming of various data transfer parameters. When a low to high transition occurs on the Control Word Strobe (CWSTR) pin, data bits BD14-BD05 of the bidirectional bus are latched into the control word register. All other bits are don't cares. Refer to the HI-8282 data sheet for the functions of the individual bits in the register.

In accordance with the ARINC 429 specification, the HI-8282 allows for words 32 bits in length only, and hence, only a 10-bit control word register is required. Some older standard devices of this type allowed the use of an 11-bit control word register and selection between 25-bit and 32-bit words.

The Receivers

The receivers use three 10-bit sampling shift registers to determine if valid data is being received, if the minimum pulse width is maintained, and if the bit rate is acceptable.

When looking for valid data bits, the internal logic looks to find three consecutive Ones or Zeros in the five upper bits of the sampling registers and three consecutive Nulls in the five lower bits within the data time. For a Null in the word gap, three consecutive Nulls in the upper five bits of the registers must be followed by three consecutive Nulls in the lower five bits.

The device also checks the bit rate by sampling the shift registers. If a data bit follows its predecessor within 8 to 12 samples, then the bit rate is acceptable. This ensures a tighter data rate tolerance than that obtained with other devices of this type which use edge sensitive receiver logic. The HI-8282 data rate tolerance is $\pm 2\%$.

The word gap timer samples the Null shift register every 10 input clocks (80 for low speed) after the last data bit of a valid reception. If the Null is present, the word gap counter is incremented. A count of three Null periods will enable the next data reception. Upon a Master Reset (\overline{MR}), the word gap counter is set. This is done to insure correct operation in self-test mode after a reset.

When the error detection circuitry determines that a bad bit has been received or a data framing error has occurred, the receivers will reject the data word.

The parity circuit generates the correct parity flag and inserts it as bit 32 in the received word.

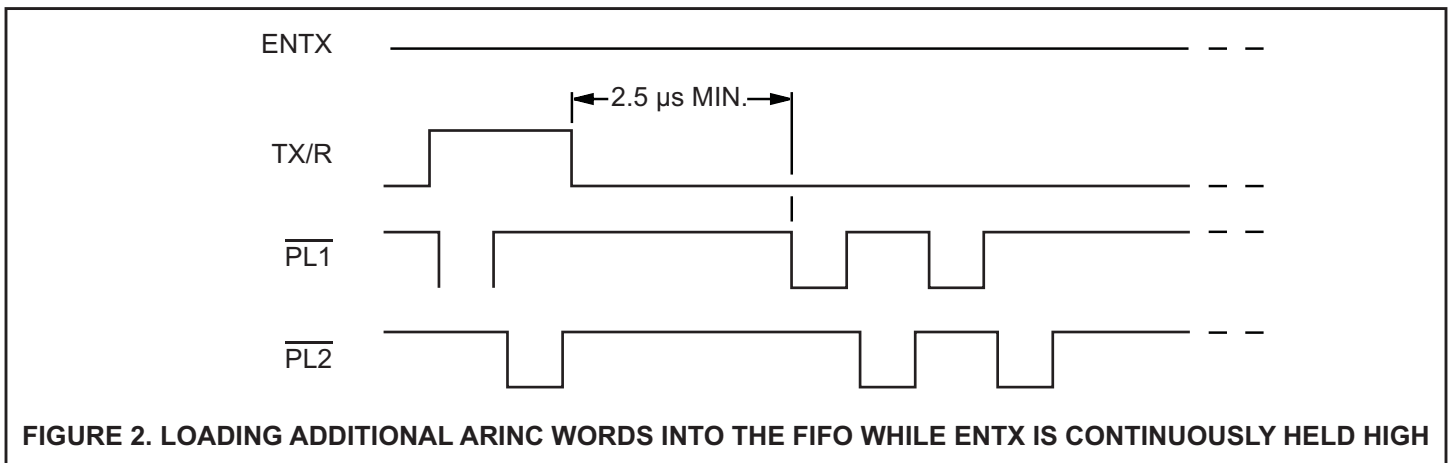
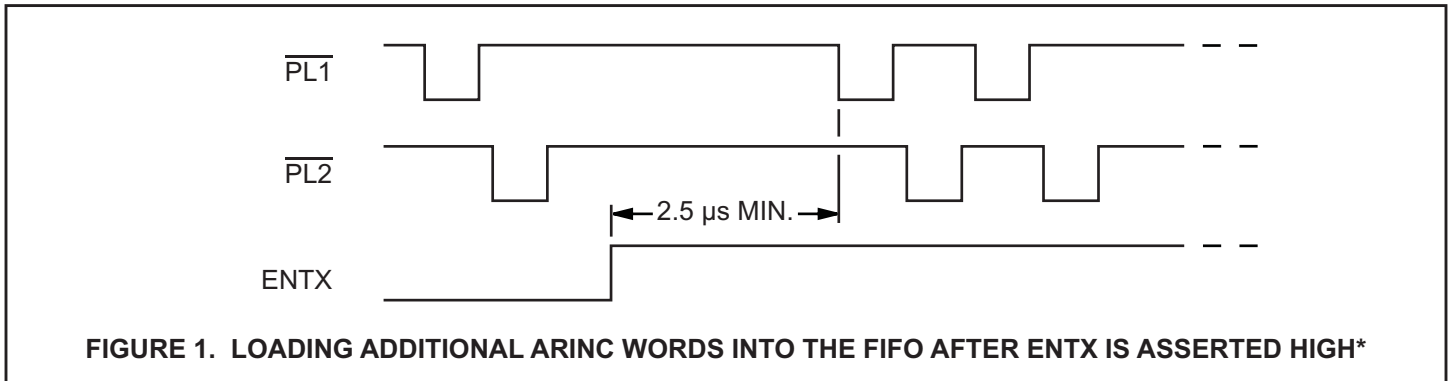
The Transmitter

The transmitter has an 8-word by 31-bit FIFO. The $\overline{PL1}$ and $\overline{PL2}$ control lines are used to load the ARINC data words into the FIFO. The first byte of each 31-bit data word is loaded with a negative going pulse on $\overline{PL1}$ and the second byte by a negative going pulse on $\overline{PL2}$. The first byte of each data word must always be loaded into the FIFO before the second byte. The 31-bit ARINC data word is not completely loaded into the FIFO until the second byte is latched by the corresponding $\overline{PL2}$ pulse.

The HI-8282 transmitter can hold from one to eight data words and is designed to transmit the words as a block of data. The optimum design is one in which the system is prevented from writing to the FIFO at all times during transmission (\overline{ENTX} high); however, data may be loaded into the FIFO during transmission of the first word, providing the conditions in Figures 1 and 2 are met.

The procedure used to load the FIFO during transmission of the first word is as follows. Transmission begins

HI-8282 TIMING SPECIFICATION



* Allowed **ONLY** during the transmission of the **FIRST** ARINC word.

when the first word is loaded into the transmitter and ENTX goes high or when ENTX is held high. It is necessary to allow a 2.5 μ sec time period following either ENTX going high or, if ENTX is always held high, $\overline{\text{TX/R}}$ going low, before loading the first byte of the next word. Once 2.5 μ sec has elapsed, an additional eight words may be loaded into the FIFO. If an attempt is made to load more than eight words into the FIFO, all additional words are ignored. This ensures a higher level of data integrity than with some other devices of this type, where the eighth word is overwritten by each additional word. When transmission of the second word has started, ENTX must remain high until the entire transmission sequence is completed and the TX/R line goes high, or data within the FIFO may be corrupted. The Transmitter Clock (TX CLK) is aligned with the transmitter output data and TX/R goes high at the end of the last transmitter output data bit.

An ARINC data word consists of 32 bits; however, the FIFO contained in the HI-8282 is 31 bits wide. The 32nd bit is the parity bit appended to the word as it is being transmitted, in accordance with the parity type programmed in the control word register.

Self Test

The device can be placed into self test mode by placing a Zero in bit BD05 of the control word. During self test, the data being transmitted by the transmitter is connected directly to the receiver shift registers, bypassing the receiver input interface circuitry. Receiver 1 receives Data True and receiver 2 receives the complementary data. The transmitter output remains active during self test.

Lightning Protection

Details of external lightning protection circuits for the HI-8282 family of products may be found in Holt's AN-300 Applications Note.

A Summary of the ARINC Specification 429

ARINC Specification 429, also known as Mark 33 Digital Information Transfer System (DITS), is a standard that is widely used in the avionics industry for the transfer of digital data between avionics system elements. Specification 429 replaces the earlier ARINC Specification 419 and was developed to alleviate much of the previous confusion by defining the standard for a single form of serial transmission.

ARINC 429 requires that the digital data is transmitted as a differential signal over a uni-directional bus composed of two twisted and shielded wires. The 32-bit data word contains a parity bit and an eight-bit label that defines the flight functions to which the remaining data bits pertain. This data is encoded in either numeric (binary or BCD) or alphabetic (ISO No. 5) format which is further divided into various fields according to the label type. All flight functions have been assigned a particular data or data format by the specification in order to prevent conflicts.

The ARINC 429 specification eliminates the need for complex interfaces between avionics systems produced by different manufacturers and provides a certain amount of "plug-in" compatibility and universality. For more information about this specification, contact Aeronautical Radio, Inc., 2551 Riva Road, Annapolis, Maryland 21401, www.arinc.com.

Applications

Interfacing the HI-8282 to a 16-Bit Data Bus

The circuit shown in Figure 3 illustrates a possible interface between the HI-8282 and a 16-bit host system. The approach to designing the interface will vary with the application; however, many systems will have similarities to the one shown here.

Logical Control

The primary interface circuitry that must be developed to implement the HI-8282 is the logic for controlling the device control lines. This can usually be accomplished using the \overline{RD} , \overline{WR} , and DEN (Data Enable) lines of the microprocessor and two address lines. In a read cycle, $\overline{EN1}$ or $\overline{EN2}$ is selected when the proper address is present while the \overline{RD} line is low. The second LSB of the address bus is used to select either the first or second byte. The DEN line is used to signal the HI-8282 to place its data on the bus. This means that the only timing requirement the host system must meet during a read cycle is a minimum pulse of 200ns plus the propagation delay of the read enable logic.

For a write cycle, the \overline{WR} line is inverted and used instead of the DEN line to enable the \overline{CWSTR} , $\overline{PL1}$, and $\overline{PL2}$ previously selected by the address bus. The timing requirement for write operations is a 200ns \overline{WR} pulse plus a minimum data disenable time equal to the total propagation delay of the write function enable gates.

Address Decoding

With the application shown in Figure 3, decoded address F8 (first byte) and FA (second byte) are for the FIFO and address FC is the control word register. During a write cycle, writing to these addresses will load the corresponding registers. During a read cycle, receiver 1 data is at address F8 (first byte) and FA (second byte). Likewise, receiver 2 data is at address FC (first byte) and FE (second byte). Reading these addresses will load the data from the corresponding receiver latch onto the data bus. Having the same addresses for the receivers and the latches presents no problem since it is impossible to write to the receivers or read the FIFO and control word register at the same time. Therefore, an address line (i.e. A1) can be connected directly to the SEL input of the device to perform the receiver latch word function. See Table 1.

\overline{RD}	\overline{WR}	ADDRESSES			
		F8	FA	FC	FE
1	0	$\overline{PL1}$	$\overline{PL2}$	\overline{CWSTR}	\overline{CWSTR}
0	1	$\overline{EN1}$ 1st Byte	$\overline{EN1}$ 2nd Byte	$\overline{EN2}$ 1st Byte	$\overline{EN2}$ 2nd Byte

TABLE 1. TYPICAL HI-8282 FUNCTIONAL DIAGRAM

Interrupts

If the device is being used in an interrupt driven system, a minimum of three interrupt vectors are needed. These include one to signal the end of a data set transmission ($\overline{TX/R}$) and one for each of the receivers to signal the presence of a valid word ready to be fetched ($\overline{D/R}$).

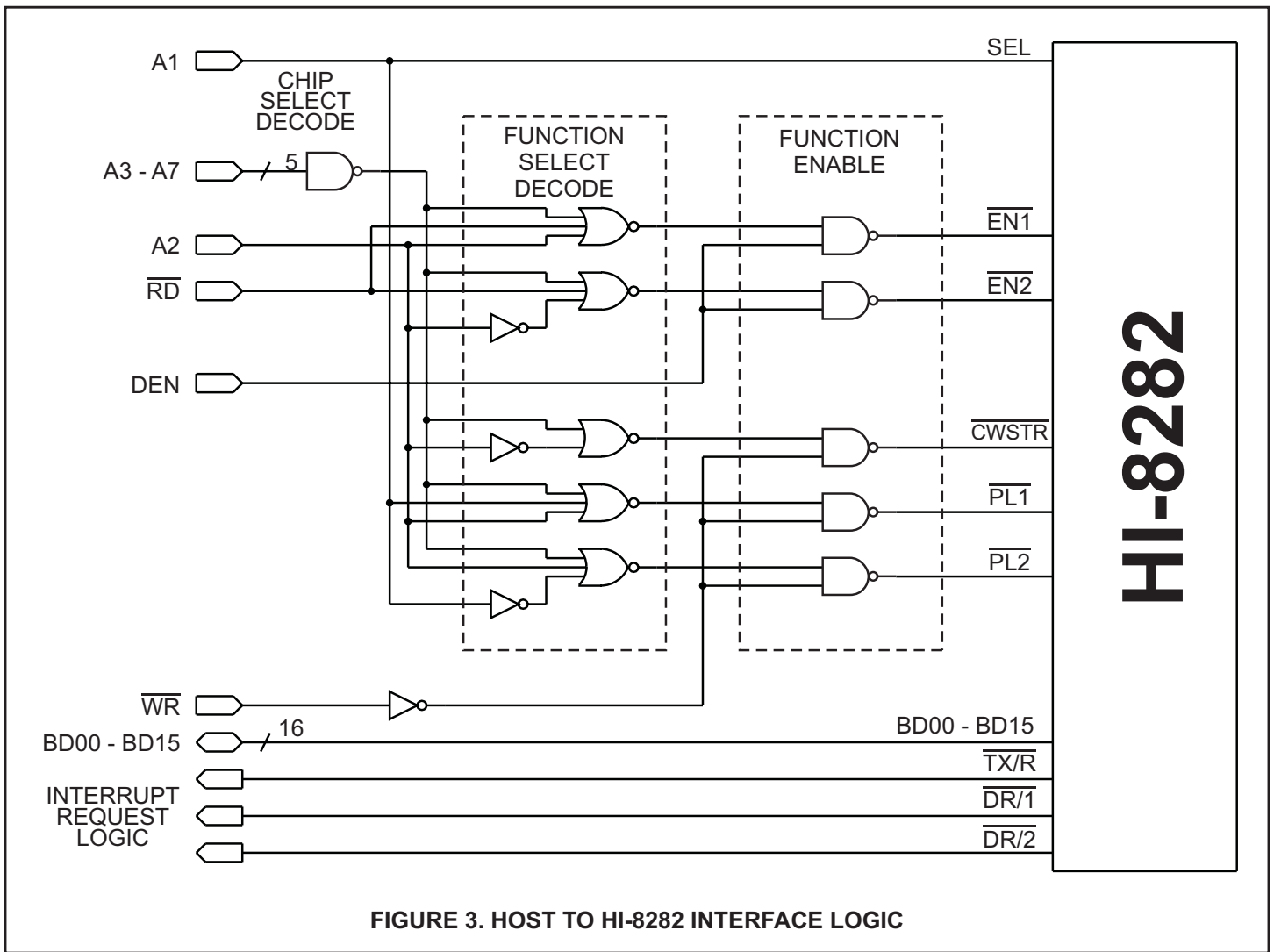


FIGURE 3. HOST TO HI-8282 INTERFACE LOGIC

Adapting the HI-8282 to an 8-Bit Data Bus

The HI-8282 is designed for use with a 16-bit data bus; however, with the addition of a few external components in the host interface circuitry, it is easily adaptable to a system with an 8-bit data bus.

Logical Control

The circuitry necessary for using the HI-8282 in a system that has an 8-bit data bus is illustrated in Figure 4. The circuit is very similar to the one in Figure 3; the primary difference being the addition of the logic circuitry necessary to drive two input/output latch devices. This additional circuitry is used to latch the high 8-bits of data while the host system uses two 8-bit operations to complete each 16-bit transfer with the HI-8282. Each of the two latch devices serves a dedicated purpose. The output latch device provides the HI-8282 with the high 8-bits during write cycles (application of signals $\overline{PL1}$, $\overline{PL2}$, and \overline{CWSTR}), and the input latch device receives the high 8-bits during read cycles (application of signals $\overline{EN1}$ and $\overline{EN2}$). The latches should have tristate outputs to avoid bus contention.

The LSB of the address bus (A0) is used to gate the latched data onto the host system data bus whenever the host system is performing high 8-bit operations. It is also inverted and used as an input to the function enable logic to prevent the HI-8282 functions from simultaneously being activated.

For a host read operation, the HI-8282 places a 16-bit word upon its bidirectional data bus. The upper 8-bits of the 16-bit byte are automatically strobed into the input latch device, while the host simultaneously receives the lower 8-bits. On the next host read cycle, the function enable logic enables the outputs of the input latch device to present the upper 8-bits of the word to the host.

During a write operation, the upper 8-bits of the 16-bit byte being written to the HI-8282 are presented to the bus interface logic. The output latch device is used to temporarily store the 8-bits until the host writes the lower 8-bits on the next write cycle. When the host presents the lower 8-bits, the output latch device is enabled by the function enable logic and both the upper and the lower 8-bits are presented in unison to the HI-8282 as a single 16-bit byte.

To Summarize:

1. To READ a 16-bit byte from the HI-8282, the host receives the LOW order 8-bits first.
2. To WRITE a 16-bit byte to the HI-8282, the host presents the HIGH order 8-bits first.

Time delays are increased by the additional logic gates. Read cycles require a minimum DEN pulse equal to 200ns plus the propagation delays of two logic gates and the input latch device. Write operations require a minimum delay from address valid to \overline{WR} enable equal to the propagation delay of two gates and a minimum \overline{WR} pulse of 200ns plus the propagation delays of two gates and the output latch device.

The latches used in this interface are positive-edge triggered. This eliminates the need for devices such as one-shots that would be necessary to drive the clock/strobe inputs on negative-edge triggered latches and thus provides a higher degree of system stability.

Address Decoding

The system in Figure 4 will have the same functional addresses as the system in Figure 3; except that, for an 8-bit bus, each particular 8-bits of each register has its own address. Therefore, the FIFO is still located at address F8 (first byte, low 8-bits) and FA (second byte, low 8-bits); however, the high 8-bits must be addressed as F9 (first byte) and FB (second byte). The receivers are addressed as shown in Table 2.

\overline{RD}	\overline{WR}	ADDRESS							
		F8	F9	FA	FB	FC	FD	FE	FF
1	0	$\overline{PL1}$ Low 8-bits	$\overline{PL1}$ High 8-bits	$\overline{PL2}$ Low 8-bits	$\overline{PL2}$ High 8-bits	\overline{CWSTR} Low 8-bits	\overline{CWSTR} High 8-bits	\overline{CWSTR} Low 8-bits	\overline{CWSTR} High 8-bits
0	1	$\overline{EN1}$ Low 8-bits 1st Byte	$\overline{EN1}$ High 8-bits 1st Byte	$\overline{EN1}$ Low 8-bits 2nd Byte	$\overline{EN1}$ High 8-bits 2nd Byte	$\overline{EN2}$ Low 8-bits 1st Byte	$\overline{EN2}$ High 8-bits 1st Byte	$\overline{EN2}$ Low 8-bits 2nd Byte	$\overline{EN2}$ High 8-bits 2nd Byte

TABLE 2. HI-8282 FUNCTIONAL DECODING USING AN 8-BIT DATA BUS

Software Driver for the HI-8282

A flowchart of a simple driver the host system could use to control and monitor the HI-8282 is shown in Figure 5. Initially, the host must configure the device by writing the control word followed by setting of the flag in system memory to indicate the transmitter is available. If data is ready to be transmitted, the host should store the data in a temporary buffer. Provided the transmitter is available, the data is taken from the buffer and loaded into the transmitter FIFO (eight 31-bit words

maximum). After loading the FIFO, the host must reset the flag in system memory to indicate the transmitter is no longer available. Unless an interrupt is received by this time, the host can move on to other tasks or continue storing data in the temporary buffer (as shown by the broken arrow). If a TX/R interrupt is received, the transmitter available flag must be set once again and data in the temporary buffer can be written into the FIFO.

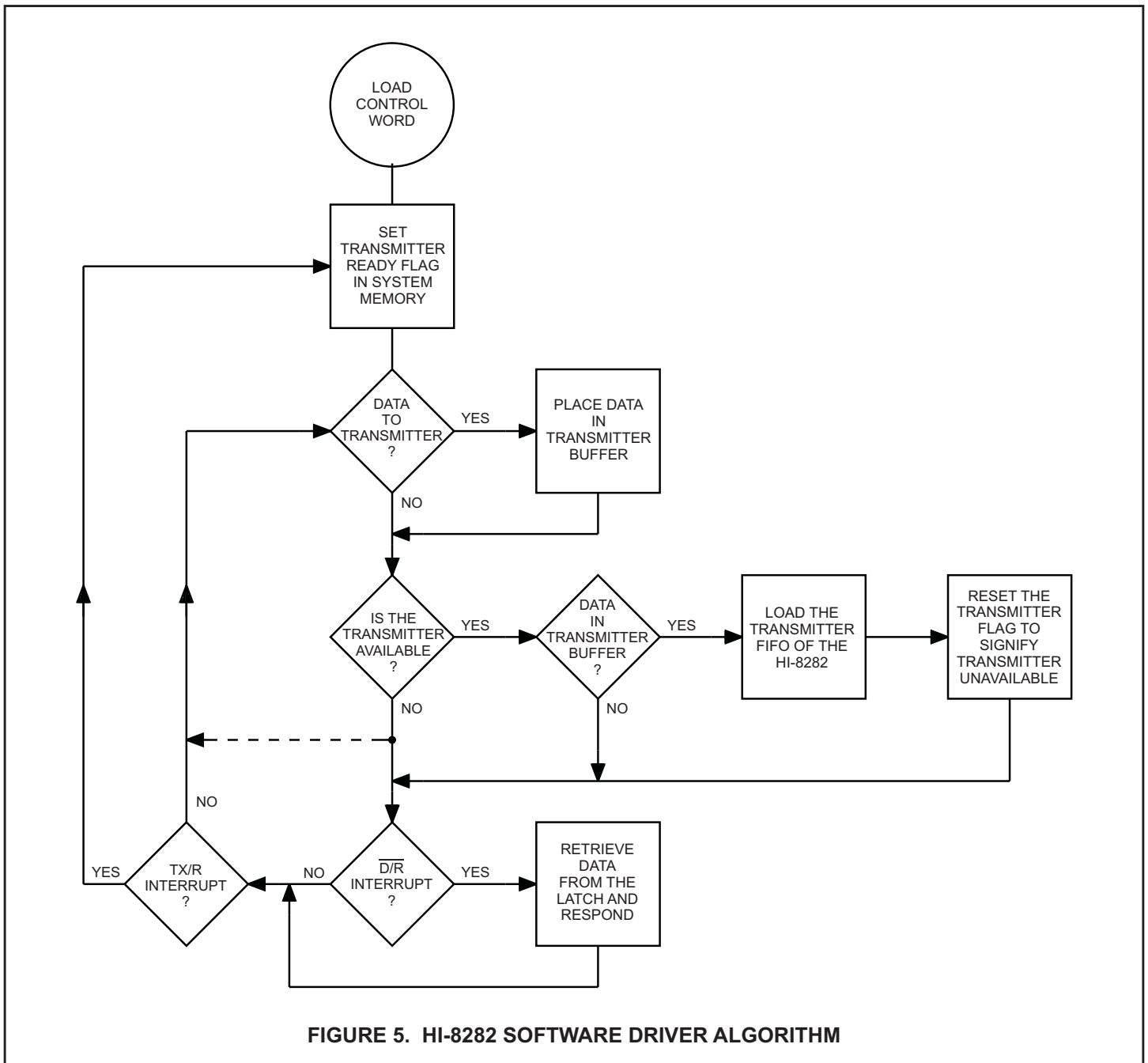


FIGURE 5. HI-8282 SOFTWARE DRIVER ALGORITHM