Using the HS-3282 ARINC Bus Interface Circuit

Application Note *March 1997* **AN400.1**

Author: Daniel B. Clifton

Introduction

The Intersil HS-3282 is a high performance CMOS programmable bus interface circuit that was designed to meet the requirements of ARINC Specification 429, and similarly encoded, time multiplexed serial data protocols. Its simple but efficient design allows the HS-3282 to be used without major complications in a variety of applications. By setting an internal control register, the HS-3282 can be programmed to operate at different data rates and with different word lengths, and to transmit either even or odd parity. The device can also be programmed to operate with or without a unique address (SDI — source/destination identifier), and with or without its self test mode implemented. Although the HS-3282 was designed to transmit and receive high-speed data rates (100K BPS or 12.5K BPS), standard baud rates of 9600 or 1200 BPS can be implemented by reducing the input clock frequency. The timing requirements of the HS-3282 make it compatible with 8086 microprocessor or other similarly based systems operating at 5MHz or less. Designed to meet the critical needs of today's advanced aircraft, the HS-3282 is a cornerstone of reliability for systems that cannot tolerate a wide margin of error.

Functional Operation of the HS-3282

In order to clarify and expand on the basic details given in the data sheet, the following information has been provided to point out some of the particularities of the HS-3282.

Clock Frequencies

The two receiver output signals, $\overline{\text{D/R1}}$ and $\overline{\text{D/R2}}$, have a minimum pulse width of one clock period. Because of this, when using a slower clock rate than 1MHz to drive the HS-3282, these outputs may remain in a low state for a few microseconds after the data in the receivers is fetched. This could cause the same data to be fetched more than once if these outputs are used to drive state sensitive interrupt requests; therefore, it is necessary to use edge sensitive interrupt requests, as mentioned in the typical application of the HS-3282 below.

Bi-Directional Data Bus

Data transfer to and from the host is accomplished via a 16-bit bidirectional, three-stated bus. The control of this bus is completely internal to the HS-3282. When data is written to the transmitter or the control word register, the bus is automatically enabled as an input; when data is read from the receivers, the bus is automatically enabled as an output; at all other times, the bus is in a high impedance state and will not interfere with external operations.

Setting Up the Control Word Register

When a low to high transition occurs on the $\overline{\text{CWSTR}}$ pin, the data on the eleven most significant bits of the bidirectional bus is latched into the control word register. The location and function of each of these bits is shown in the data sheet.

Receiver Operation

Incoming data from the line receiver (or from the self test circuit) is shifted into the data shift register by the word gap timer on an edge sensitive basis. This results in a high data rate tolerance; although ARINC specification 429 requires at least a ±1% tolerance, the HS-3282 has at least a ±10% tolerance at all data rates. In order to prevent reception errors, the word gap timer is designed to disable and reset the receiver upon reception of two consecutive null times (or two consecutive data times), and re-enable the receiver after four additional null times. If a word of the proper length (and SDI) has been received by this time, the word will be latched and the $\overline{\text{D/R}}$ flag of the corresponding receiver will go low to signal the host that a valid word is ready to be fetched. (For the most efficient operation, the $\overline{\text{D/R}}$ flags should be used to generate interrupt requests to the host system.) It should be noted that the parity bit that is stored in the receiver latch may be different than the parity bit of the word that was received. This is because the parity bit that is stored in the receiver latch is actually a parity flag, indicating by its status the parity of the word that was received: if the parity of the word received was odd, the parity flag will be a logic "0"; if the parity of the word received was even, the parity flag will be a logic "1".

Once a word is ready to be fetched, it may be read by the host system in two 16-bit parts over the bi-directional bus by strobing the appropriate $\overline{\text{EN}}$ line low twice, once with the SEL line low to read "word 1" and once with the SEL line high to read "word 2". (If preferred, "word 2" may be read before "word 1", or the $\overline{\text{EN}}$ line may be held low for one long pulse while the SEL input is toggled to select first one word and then the other.) The actual ARINC contents of these two 16-bit "words" (or the contents of the 25-bit word length) is shown in the data sheet. It should be noted that the $\overline{\text{D/R}}$ flag will not be reset unless both words are read.

Transmitter Operation

As mentioned in the data sheet, the transmitter has a FIFO that can hold up to eight data words. Although the HS-3282 has two inputs, $\overline{\text{PL1}}$ and $\overline{\text{PL2}}$, used to write data into the FIFO, the data is not actually entered into the FIFO until the second input ($\overline{\text{PL2}}$) is pulsed low. Therefore, the first half of each data word must be written to the HS-3282 first. Then, when the $\overline{\text{PL2}}$ input is pulsed low to write the second half of the word, the proper data will be transferred into the FIFO.

The HS-3282 transmitter is designed to transmit data in sets containing from one to eight 32 or 25-bit words. It is of primary importance that the transmitter FIFO not be disturbed while this transmission is taking place; therefore, systems should be designed to disallow writing to the FIFO while transmission is taking place. The only exception to this rule is that words can be written into the FIFO while the first word (only) is being transmitted; if transmission of a second word is started before the TX/R has returned to a high state, the FIFO must remain undisturbed until the entire transmission sequence is completed and the TX/R flag goes high.

The TX/R flag becomes useful here as an interrupt request output to the host system, signaling that the FIFO is ready for another set of data words.

It is also of primary importance that the ENTX input remain high for the duration of the transmission sequence or the integrity of the data in the FIFO will be broken. This can best be accomplished by feeding the TX/R flag through an inverter and back into the ENTX input. This application will enable the transmitter as soon as the first word is written into the FIFO, and keep it enabled until the transmission sequence is completed; since most host systems operate at much greater speeds than the transmitter, an additional seven words could easily be written into the FIFO while the first word is still being transmitted.

The value of the parity bit as written into the FIFO makes no difference since the transmitter sets the parity bit at transmission time according to the type of parity that it has been programmed to transmit.

Lightning Protection

Although the bus driver has been protected by a 100mW fuse capable of sinking up to 1A for short periods of time (100ms) and internal zener diodes which saturate at about 8.7V, the bus interface circuit has no such protection for its receiver inputs. Because of the possibility of a lighting strike to aircraft, additional protection should be used to protect both the HS-3282 and the bus driver from high voltage spikes. External avalanche diodes with high power ratings (five or ten watts) should be used to clamp the bus at about ±6.8V. This will prevent the fuse and the zeners in the driver from being burned out by current surges, and it will keep the voltage level on the inputs of the receiver within acceptable limits.

A Typical Application of the HS-3282

The following example shows one possible way to interface the HS-3282 with a host system. Although different applications may require different approaches, most systems will have requirements similar to those that have been met here.

Logical Control

As shown in Figure 1, the support circuitry necessary to integrate the HS-3282 into a system primarily involves a logic circuit to drive the control inputs. In most cases, this can be achieved using microprocessor $\overline{RD}$ and $\overline{WR}$ bus control signals and two address lines in conjunction with a decoded chip select line and a DEN (data enable) line. During READ operations, the particular function ($\overline{EN1}$ or $\overline{EN2}$) is selected when the proper address is present while the $\overline{RD}$ line is low; note that the second LSB of the address bus is used to select either word 1 or word 2. The DEN line is used to signal the BIU to place its data on the bus.

Therefore, for read operations, the only timing requirement the host system must meet is a minimum DEN pulse of 200ns plus the propagation delay of the enable gates. During write operations, the $\overline{WR}$ line is inverted and used instead of the DEN line to enable the particular function ($\overline{CWSTR}$, $\overline{PL1}$, or $\overline{PL2}$) previously selected by the address bus. Since the minimum data hold time of the HS-3282 is 0ns, the timing requirement for write operations, besides the minimum 200ns $\overline{WR}$ pulse, is a minimum data disenable delay equal to the total propagation delay caused by the function enable gates.

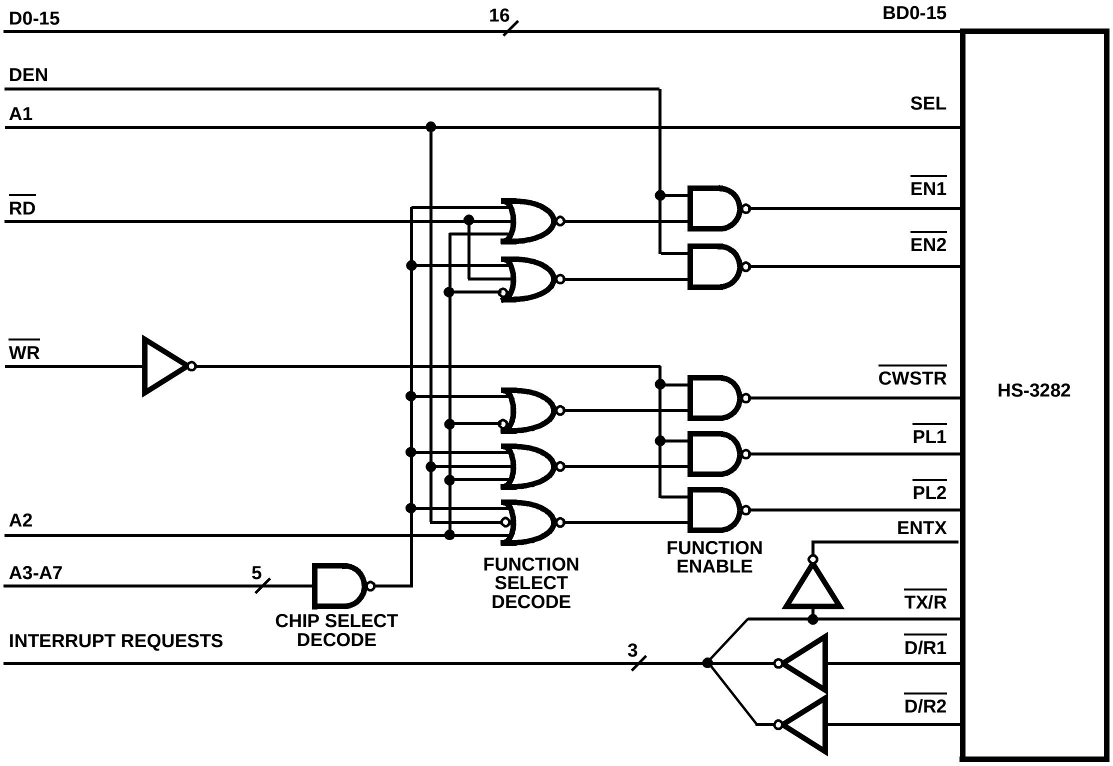

FIGURE 1. TYPICAL HOST TO HS-3282 INTERFACE LOGIC

Address Decoding

In the system shown, the transmitter FIFO is decoded as addresses F8 (first word) and FA (second word), and the control word register is at FC; writing to these addresses will load the corresponding registers of the HS-3282. Receiver latch #1 is at addresses F8 (first word) and FA (second word), and Receiver latch #2 is at addresses FC (first word) and FE (second word); reading these addresses will load the data from the corresponding receiver latch of the HS-3282. Note that an address line can be connected directly to the SEL input of the HS-3282 to perform the receiver latch word select function. Since it is impossible to write to the receiver latches or to read the transmitter FIFO and the control word register, the addresses of the read functions can overlap the addresses of the write functions without presenting a problem. See Table 1.

TABLE 1. TYPICAL HS-3282 FUNCTIONAL DECODING

		ADDRESS			
$\overline{RD}$	$\overline{WR}$	F8	FA	FC	FE
1	0	$\overline{PL1}$	$\overline{PL2}$	CWSTR	CWSTR
0	1	$\overline{EN1}$ (1st Word)	$\overline{EN1}$ (2nd Word)	EN2 (1st Word)	EN2 (2nd Word)

Interrupts

For the most efficient operation, any system incorporating the HS-3282 should provide conditions by which the device can generate interrupt requests to the host system. If both receivers are being used, then a minimum of three interrupt vectors are needed: one (TX/R) to signal the end of a data set transmission, and one for each of the receivers ($\overline{D/R}$) to signal the presence of a valid word ready to be fetched. Since the TX/R signal can remain high for indeterminate periods of time while the transmitter is inactive, the requests should be received by the host on an edge sensitive basis.

Software Requirements

A flowchart of a simple algorithm that the host system could use to exercise and monitor the functions of the HS-3282 is shown in Figure 2. In order to begin, the HS-3282 must be initialized with a control word, then the host should set some flag in system memory to indicate that the transmitter is available. If data is ready to be transmitted, the host should store the data in a temporary buffer until the transmitter becomes available. At this time, the data is taken from the buffer and loaded into the transmitter FIFO (eight 32-bit words maximum). After loading the FIFO, the host should reset the flag in system memory mentioned previously to indicate that 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 $\overline{D/R}$ interrupt is received, data can be read from the receiver latch and any corresponding action taken. If a TX/R interrupt is received, the transmitter available flag should be set once again, and any ready data in the buffer could be written into the FIFO.

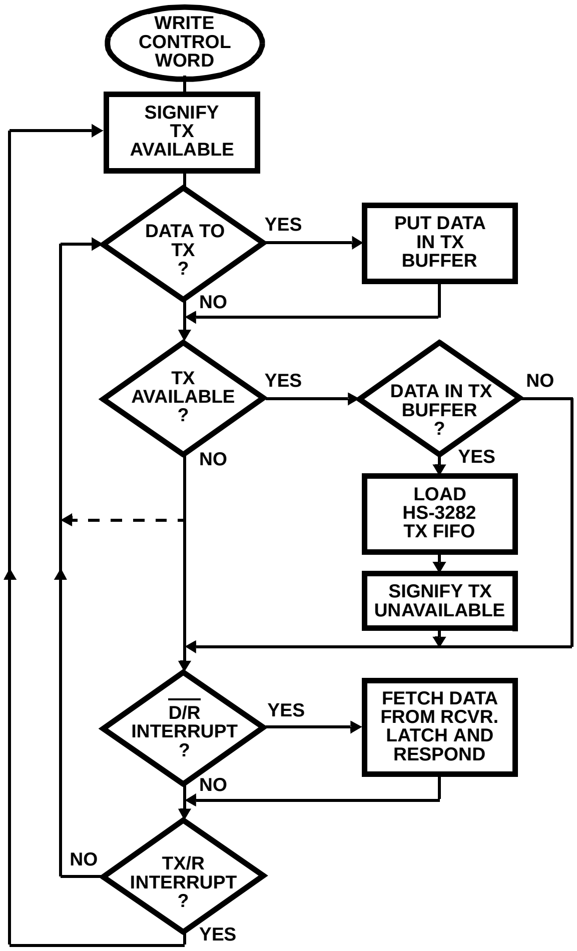

FIGURE 2. SIMPLE HS-3282 DRIVER ALGORITHM

Adapting the HS-3282 to an 8-Bit Data Bus

Although designed for a 16-bit data bus, the HS-3282 can be used in systems with an 8-bit data bus by adding a few external components. The following example shows how this can be done.

Logical Control

Figure 3 shows the circuitry necessary to integrate the HS-3282 into a system that utilizes an 8-bit data bus. The circuit is very similar to the one in Figure 1, the primary difference being the addition of the logic circuitry necessary to drive two input/output latches. The primary purpose of this additional circuitry is to latch the high order byte of data so that the host system can accomplish each 16-bit data transfer with the HS-3282 by using two separate 8-bit operations. Each of the two latches serves a specific purpose: the output latch provides the HS-3282 with the high order byte during write operations ($\overline{PL1}$, $\overline{PL2}$, and $\overline{CWSTR}$), and the input latch receives the high order byte during read operations ($\overline{EN1}$ and $\overline{EN2}$). The outputs on these latches should be three-stated to avoid bus contention. The least significant bit of the address bus is used to activate the latches onto the host system data bus whenever the host system is performing high order byte operations, and this same address bit is inverted and added as an input to each of the function enable gates to prevent the HS-3282 from being activated at the same time. During low order byte operations, the outputs of the function enable gates are used to activate the latches onto the high order byte of the bidirectional bus of the HS-3282, allowing the BIU to instantaneously transfer a full 16-bit word. Since the latches are automatically activated in unison with the BlU during low order byte operations, the high order byte must be handled first during write operations so that the proper data will be present in the output latch when the lower byte is written to the BlU. Conversely, the low order byte must be handled first during read operations since the high order byte is automatically strobed into the input latch when the low order byte is read. Timing requirements are increased by the additional logic gates; read operations now require a minimum DEN pulse equal to 200ns plus the propagation delays of two logic gates, a one-shot, and the input latch; write operations now require a minimum delay from address valid to $\overline{\text{WR}}$ enable equal to the propagation delays of two gates and a minimum $\overline{\text{WR}}$ pulse of 200ns plus the propagation delays of two gates and the output latch.

Note that one shots are used to drive the strobes on the latches. This is necessary so that the falling edge of the strobe occurs while the data is still active on the bus; otherwise, the propagation delays of the additional gates would cause the latches to close after the data had been disenabled.

Address Decoding

The system in Figure 3 will have the same functional addresses as the system in Figure 1 except that in this case each particular byte of each register has its own address. Therefore, the FIFO is still located at addresses F8 (first word, low byte) and FA (second word, low byte); however, the high bytes must be addressed as F9 (first word) and FB (second word). The receivers are addressed in the same way as shown in Table 2.

Conclusion

Although it was designed for ARINC applications, the HS-3282 bus interface circuit is a very versatile device, capable of serving any type of communications purpose. Its high speed capability and its high reliability make it especially useful in scientific and real time operations where large volume data gathering and time critical transmission of control signals is required. These qualities, in combination with the simplicity with which the device may be incorporated into a system, make the HS-3282 a wise choice for a wide spectrum of applications.

ARINC Specification 429 A Brief Overview

ARINC Specification 429, otherwise known as the Mark 33 Digital Information Transfer System (DITS), is a definition of standards used extensively by the air transport industry for the transfer of digital data between avionics systems elements. Systems utilizing this standard have been installed in a wide range of aircraft including the Boeing 737, 747, 757, and 767; the European Airbus; Bell Helicopter; and a large number of small aircraft. Replacing the earlier ARINC Specification 419 which had standardized the various forms of serial transmission developed during the emergent period of digital avionics technology, Specification 429 eliminates much of the previous confusion by defining the standard for a single form of serial transmission.

According to Specification 429, digital data is transmitted via a differential signal over a uni-directional bus composed of two twisted and shielded wires. The data is sent in 32-bit words, each word containing a parity bit and an eight-bit label that defines the flight function to which the remaining data pertains. This data, encoded in either numeric (binary or BCD) or alphabetic (ISO No. 5) format, is further divided into various fields according to label type. In order to completely standardize communication and prevent conflicts, all flight functions have been assigned a particular label and data format.

The ARINC Specification 429, by defining a single standard for the transfer of digital information, eliminates the need for complex interfaces between avionics systems produced by different manufacturers. This provides those avionics components that conform to this standard with a virtual “plug-in” capability, and gives such components a certain measure of universality. For more information about this specification, contact Aeronautical Radio, Inc., 2551 Riva Road, Annapolis, Maryland 21401.

TABLE 2. TYPICAL HS-3282 FUNCTIONAL DECODING USING AN 8-BIT DATA BUS

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

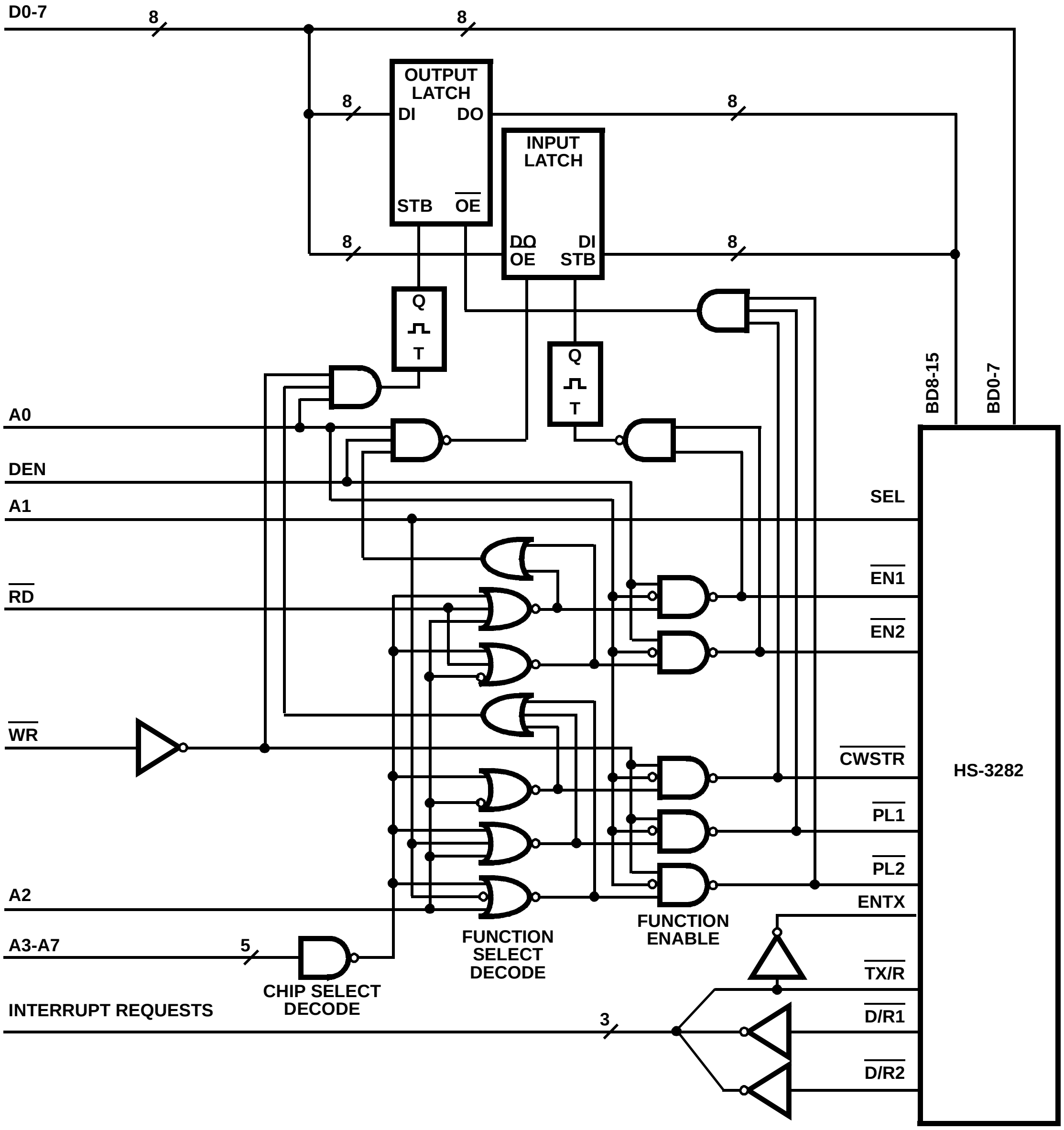

FIGURE 3. TYPICAL HOST TO HS-3282 INTERFACE LOGIC USING AN 8-BIT DATA BUS