# HANDLING VARIABLE DATA RATES WITH SERDEX

## FAST-TO-SLOW, SLOW-TO-FAST, STEADY-TO-BURST, BURST-TO-STEADY

by David G. Larsen and Dr. Peter Rony

The use of on-line data acquisition and control in the research laboratory has experienced dramatic changes since the introduction of the general-purpose small computer in the early 1960's. The cost was, of course, prohibitive for the first few years, for processing, memory, and -- to a lesser extent -- for analog/digital data-acquisition equipment. Then, as the cost tumbled at the rate of about 40% per year for a constant price/performance ratio, the use of a minicomputer became acceptable to more and more laboratories.

There were still rather serious limitations on the use of a minicomputer for the average research laboratory. The basic minicomputer was, and still is, only a small part of the system's cost. In addition to the central processing unit (CPU), interface hardware, and data-acquisition circuitry, software was also required. These additional requirements were so expensive, if purchased as a turnkey system, that in many instances a decision was made to build a "homemade" interface system. For most laboratories, this almost always proved to be disastrous: the amount of effort needed to write software at an assembler level and build interface hardware was, and will continue to be, underestimated by factors of five to ten times. Some of the very large corporate research labs were able to absorb the time, cost, and range of talent (from analog circuit design to digital software) and have very viable automation systems. However, most groups who built "homemade" systems discovered that their final operating system required an overall investment that had been substantially underestimated.

Recently[1], the SERDEX™ SERial Data EXchange modules have become available.* It is clear that the use of asynchronous serial ASCII-coded data, along with optically-isolated standard 20mA current loops, the two most important characteristics of SERDEX, will permit substantial savings of time and money associated with computer interfacing. With the SERDEX modules, only digital signals are transmitted; only a single twisted pair of wire is required for each remote location; both data and control commands can be either transmitted or received on each channel; no hardware interface to teletypewriters or to teletypewriter ports is required; all programming may be done in a high-level language, such as BASIC; high noise-immunity exists; and -- finally -- serial data can be transmitted at the rate of 20,000 bits per second, which corresponds to 1800 ASCII characters per second.

The advantages of using asynchronous serial ASCII-coded data are almost overwhelming. In fact, the only disadvantage that such a technique has, when compared to parallel data-transmission systems, is speed. With serial ASCII-coded data, we will occasionally encounter situations where a loss of data, owing to a mismatch in the speed at which data can be handled, occurs, somewhere between the point at which such data is acquired and the point at which it enters the minicomputer.

An obvious source of rate-limiting is the teletypewriter itself, which can only handle data at the rate of 10 ASCII characters/second. (However, most minicomputers have available as options telecommunications ports that will operate at up to 20kbit/s.) A second rate-limiting step can occur when high-level languages are employed; such languages can require appreciable time to perform several digital multiplications and divisions, and this time will naturally limit the rate at which data can be accepted by the minicomputer.

For the resolution of dilemmas such as these, the user may find *First-In First-Out serial memories (FIFO's)* to be convenient buffers between digital devices whose data-handling speeds are mismatched. The principle of operation of the FIFO is straightforward:

- Digital data bits are independently clocked into the FIFO memory via the input clock. The data entry proceeds as long as the memory remains not completely full.
- The data bits, upon entering the FIFO, immediately ripple through the memory and stack up at the output terminals.
- The data bits are subsequently clocked out of the FIFO at a clock rate that is usually (and can be substantially) different from the input clock rate.
- A strict sequential order is maintained between the entering and leaving data bits: the first data bits to enter the FIFO are the first ones to leave.

The term, "first-in first-out memory" is certainly an apt description of this interesting LSI chip.

One of the popular FIFO's is the Fairchild Semiconductor 3341 64-word by 4-bit FIFO, which can operate at 1MHz shift-in shift-out data rates. According to the manufacturer:[2] "An ideal use of the 3341 FIFO's is with two systems or subsystems of differing data or unsynchronized data rates that must talk to each other. Data can be entered or removed in steady streams, bursts, or irregular patterns, or a combination of these." In our example, which involves the transmission of

*THE AUTHORS are with the Scientific Telecommunications and Computer Interfacing Laboratories, Virginia Polytechnic Institute and State University, Blacksburg, Virginia. Mr. Larsen is in the Department of Chemistry, and Dr. Rony is in the Department of Chemical Engineering.*

[1] "SERDEX—The Near-Ideal System Interface," *Analog Dialogue* 7-2, 1973.

*For complete information on SERDEX, use the reply card. Request M17.

[2] Fairchild Semiconductor: OPTIMOS manual, September, 1972, pp. 74-79.

digital information from a digital instrument, via a SERDEX transmitter module, to either a teletypewriter, teletypewriter port, or telecommunication serial minicomputer port, we can envision the following situations:

A. Irregular data can be collected from a remote location and stored in a FIFO for transmission by a SERDEX transmitter module when it is convenient.

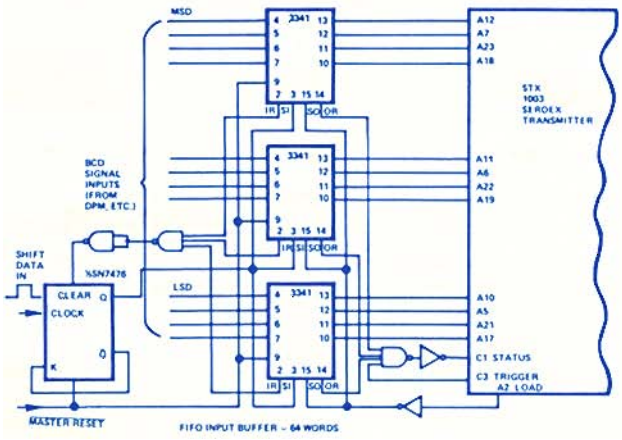
B. High-speed data can be collected in bursts from a remote location and stored in a FIFO for transmission by a SERDEX transmitter module at considerably slower data rates.

C. A slow flow of steady data can be collected from a remote location and stored in a FIFO which, with the aid of a SERDEX transmitter module (STX1003), re-transmits the data in bursts to the minicomputer, thus promoting more-efficient use of the CPU.

D. Data for printout on a teletypewriter can be loaded into a FIFO, freeing the SERDEX transmitter module and allowing the teletypewriter to proceed at its own slower pace.

To illustrate item C above, let us consider a laboratory that has a number of slow data-generating systems that require on-line data logging (but perhaps minimal immediate control response). It would be very desirable to use FIFO's between the sources of data (including converters or multiplexers) and the parallel *n*-digit BCD (binary-coded decimal) input to the SERDEX transmitter module; these *n* FIFO's would store parallel BCD data and thus reduce the number of times the SERDEX transmitter would need to be serviced by the minicomputer.

For example, let us assume that a liquid chromatograph output is such that one data-point per second is an adequate sampling rate. If the A/D converter output were 4 digits BCD (0.01% resolution), then, by multiplexing, 16 data points could be stored in a single 3341 FIFO without losing any data. It would then take a CPU asynchronous port, operating at 20,000 bits/second, a total of 35.2ms to read the 16 ASCII characters transmitted by the SERDEX transmitter module, operating at a similar 20,000b/s data rate. This corresponds to one 35.2ms transmission in a 16s period, which is a much more efficient way of handling data than to cause a CPU to service the transmitter module once a second. Using external memories at the parallel BCD data source, as suggested in this example, allows us much more flexibility in timing constraints when we write software, to service both a number of on-line instruments that are slow, and also those instruments that generate data at a very fast rate, but intermittently.




The figure shows a FIFO memory -- which can accept three BCD data bits and store 64 words -- used with the SERDEX transmitter. Of course, a FIFO can be added for each additional BCD character that is desired. The system works in the following manner (a more-concrete functional description appears in the Appendix):

- To input data into the memory: A clock pulse (*shift data in*) must be provided when the desired ("ready") BCD data is present at the FIFO inputs. The data may be entered at word rates from dc to 1MHz. At any time the FIFO is full, additional data points will simply be lost, until data has been shifted out to make room for new data.

- To transmit data, the STX1003 is used in the normal way: When an ASCII "?" is received, trigger C3 will initiate transmission by being *nand*-ed with the FIFO data-ready flags, and trigger the STX transmitter at C1. If there is no data in the FIFO, no transmission will occur when the ASCII "?" is received.

## APPENDIX

First Data In, First Data Out (FIFO) Memory

The Fairchild 3341 is built with MOS technology and mounted in a 16-pin dual in-line integrated circuit package, and it is fully TTL-compatible. The name truly defines the system operation in terms of data flow. The 3341 is a 4-bit, 64-word FIFO: the first word entered will bubble to the output and be the first data removed as a 4-bit parallel word. Both input and output are asynchronous and independent in operation. The maximum data transfer rate of 4-bit words is 1MHz; however the input and the output can operate at the same or at different rates simultaneously.

DATA INPUT: When the Input Ready Flag (pin 2, "IR") is *high*, data on the 4 Input lines (pins 4, 5, 6, 7) may be strobed in by externally applying a *low-to-high* transition at the Shift Input (pin 3, "SI"). The "IR" input goes *low* after the *low-to-high* at "SI," and when "SI" is returned to *low*, the data will flow from the first input position towards the output, stopping in the last empty word location in the memory. All previously-entered data that has not been shifted out will be ahead of the last 4-bit word entered into the memory.

DATA OUTPUT: When the FIFO has data ready at its Output pins (10, 11, 12, and 13), the Output Ready Flag (pin 14, "OR"), will go *high*. New data can be shifted into the output by going from *low* to *high* on the Shift Output (pin 15, "SO"). "OR" will then go *low*, and when "SO" returns to a *low* state, new data will be shifted into the output, provided that there is data in the FIFO.

OTHER CONSIDERATIONS: If it is necessary to know exactly how much data the FIFO memory contains at any one instant, an external register can be built, using *up-down* counters, to perform this task. The 3341 can be used to build FIFO memories which accommodate large parallel words, in increments of 4 bits; it can also be easily expanded to hold more than 64 words, in increments of 64 words. See the manufacturer's application notes for these and other possibilities. ▶▶▶