Lawrence Berkeley National Laboratory
Recent Work

Title
SERIAL DATA TRANSMISSION IN SMALL COMPUTER SYSTEMS

Permalink
https://escholarship.org/uc/item/1g40b884

Author
Meng, J.D.

Publication Date
1970-02-01
SERIAL DATA TRANSMISSION
IN SMALL COMPUTER SYSTEMS

J. D. Meng

February 1970

AEC Contract No. W-7405-eng-48
DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.
Abstract—Serial input/output connections for small computer systems based on fast serial data transmission are described. Input/output cabling problems are minimized with no sacrifice in system performance. Transistor-transistor micrologic operating at 20 MHz is used for transmission and reception. Included is a description of an operational system on a 24-bit word length machine in use at a small cyclotron.

I. INTRODUCTION

During the past several years, integrated circuit technology has developed explosively. Consequently, small-to-medium-sized computers have become physically smaller and dramatically less expensive. Peripheral connection techniques have remained essentially the same during this period, however. Data bits are wired in parallel to each device. If 24 bits of data per word must be transmitted, 24 wires plus individual ground returns are often run between the device and the computer. To handle the input/output for a 24-bit
word length computer, we have estimated a cable containing more than 200 wires is required (24-bit memory data input, 24-bit memory data output; 24-bit central processor input, and 24-bit central processor output plus various control signals). Some economy can be realized by making data busses bi-directional, and this has been done in many computer systems.

However, the previously mentioned integrated circuit technology advances have made straight-forward single bit serial data transmission between a computer and its peripherals a competitive possibility. Such a scheme potentially can reduce cabling from 200 wires per input/output device to about four coaxial cables per system.

Serial transmission is practical because ferrite core memories, which are still in general use by virtually all small computers, typically have a read-restore cycle time of 1 - 2 µs. This time (once again assuming a relatively small, inexpensive computer system) will dictate the maximum possible input/output data rate. If we assume a 20 MHz clock rate (50 ns per bit) of serial data transmission, a 12-bit data word can be transferred serially in 12 x 50 = 600 ns. Twenty-four bits can be transmitted in 1.2 µs. These times are close to the memory cycle times available. Transmitting data serially is not, therefore, a time bottleneck in the system. 20 MHz shift registers and logic are inexpensive, off-the-shelf items today. A 20 MHz bit transfer rate is, consequently, not only practical, but inexpensive and relatively simple.
II. TRANSMISSION SCHEMES

Numerous serial data transmission schemes have been in common use for many years. The common teletype connection is serial, as is data for disk and delay-line memories (where recording is 1-bit serial).

Figures 2, 3 and 4 are block diagrams of three practical serial transmission schemes. Each of these is a possibility for general computer input/output use. The first two are single-wire systems. That is, data is transferred over a single coaxial cable. The third uses two cables, one for clock transmission and the other for data transmission.

The key element in all these schemes is the shift register - a device which will present data serially at its output at a rate of one bit every 50 ns. The particular device which we are using in our system is packaged four bits per unit and each unit can be loaded in a parallel mode, the data then being shiftable at the 20 MHz rate (Fig. 1). Each bit also has an individual output wire. The device is serial or parallel in and serial or parallel out, and requires a single phase clock for shifting. In the receive mode, the serial-input/parallel-output capabilities are used. For transmitters, the parallel-input/serial-output capabilities are used.
III. SINGLE - CABLE SCHEMES

I will mention two single cable schemes briefly. These schemes are, in our opinion, excellent candidates for future systems. For the system which we are currently developing, however, several considerations precluded their use.

First, although none of the schemes could be considered complex, the single cable ones are not as straightforward in implementation as is the two wire scheme on which we settled. Second, although we looked upon the use of 200 wires for system inter-communications as impractical, the difference in economy between using one cable as opposed to that of using two was not significant. Third, transmission time for the one cable schemes considered is longer per bit than that for the two cable scheme selected. Using two cables, our system performance was not degraded. A slight degrading of total system performance would occur because of the lower effective bit-transmission rate of the single cable schemes.

Figure 2 is a detailed block diagram of pulse-width modulation. Figure 2 (a) illustrates the signal transmitted. Figure 2 (b) is the logic used to translate a shift-register data output and a shift clock output into the transmitted signal. The shift-register signals are separated into "1" pulses and "0" pulses. These are then used to trigger monostables having periods corresponding to the "1" pulse width and the "0" pulse width, respectively. The monostable outputs are OR-ed together to form the transmitted output signal.
Figure 2 (c) is the logic for receiving this pulse-width modulated signal and translating it into a clock and data signal suitable for driving the serial input on a shift register. The rising edge of the incoming signal triggers a monostable having a period exceeding that of a "0" pulse but less than that of a "1" pulse. The falling edge of the incoming signal is the shift strobe for the receiving shift register. The inverted output of the monostable (rising-edge triggered) becomes the data input for the receiving shift register. If it is low at the time of the input pulse falling edge, a "0" is shifted into the shift register.

Figure 3 is a similar set of drawings for another single-cable system. It has the potential advantage of using equal width bi-polar pulses for data transmission (clock pulses could all be made positive; data pulses negative. For the sake of block-logic simplicity, all the pulses are shown positive in Fig. 3). This allows simple pulse transformer coupling in systems where complete dc isolation is required between the peripheral device and the computer.

Figure 3 (a) illustrates the output pulse train. Odd pulses in a burst are clock pulses. Even pulses are data. "One's" are arbitrarily specified by a shorter time between the clock and data pulses than "zeros".

* A realistic example of this requirement exists in the author's laboratory where interconnecting the 60 Hz ac power grounds among experimental caves in a cyclotron deals a fatal blow to the sensitive analog equipment in use. All the caves, however, are required to communicate digitally with a single central computer.
Figure 3 (b) is the coding logic for translating shift register signals into a pulse-frequency modulated pulse train. The translation begins identically to that for pulse width modulation (Fig. 1). That is, "1" and "0" pulses are separated and used to trigger monostables set for different periods. The monostable outputs are OR-ed together and at the end of the resulting pulse, a 50 ns pulse is produced by another monostable. This pulse, the data pulse, is OR-ed with the clock pulse and the result is transmitted. (Where it would be desirable to use bi-polar pulse transmission, a special driver must be used in place of the simple OR gate shown here.)

Figure 3 (c) illustrates the translation of this pulse train into a separate clock and data signal for use in driving the receiving shift register. The incoming signal drives the toggle input on a bi-stable. The output of this bi-stable is a pulse width modulated signal and is decoded as in Fig. 2 (c).

Figure 4 illustrates the two-cable scheme we are now using. Both clock and data are transmitted, and no translation is required at either the transmitting or receiving end. The most significant problem for this system is maintenance of the relative times between clock pulses and data transitions. The delay time from the falling edge of a clock pulse to a resulting data transition determines the maximum shift rate for the shift register. This is shown in Fig. 5 (a).
Operating margins can be improved significantly by using different coaxial cable lengths to trim the signal timing to make it look like that in Fig. 5 (b) to the receiver. Additional problems associated with maintaining these relative delays will be considered in greater detail when we discuss our operating system. The following table summarizes advantages and disadvantages of these serial schemes.

TABLE I

COMPARISON OF SERIAL TRANSMISSION SCHEMES

<table>
<thead>
<tr>
<th>Pulse-Width Modulation</th>
<th>Pulse-Frequency Mod.</th>
<th>Clock &amp; Data Transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Cable</td>
<td>Single Cable</td>
<td>Two Cables Required</td>
</tr>
<tr>
<td>Lower Bit-Transmission Rate</td>
<td>Lower Bit-Transmission Rate</td>
<td>High Bit-Transmission Rate</td>
</tr>
<tr>
<td>dc-Level Pulses Make Isolation More Complex</td>
<td>Possibility of Using Bipolar Pulse Transmission to make Isolation Simple.</td>
<td>dc-Level Data and Clock Pulses Make Isolation More Complicated</td>
</tr>
<tr>
<td>Simple Shift Register to Line Translation Required</td>
<td>More Complicated Shift Register to Line Translation Required</td>
<td>No Shift Register to Line Translation Required</td>
</tr>
<tr>
<td>Delays are not Critical</td>
<td>Delays are not Critical</td>
<td>Maintenance of Clock Delays Relative to Data Delays is Critical</td>
</tr>
</tbody>
</table>
This is not meant to be a glossary of such schemes, but merely enough of a sketch to demonstrate the ready availability of many serial data transmission schemes to the imaginative system designer. We will proceed with some details relating to a computer input/output system designed around the above mentioned two wire scheme.

IV. DESIGNING A COMPUTER INPUT/OUTPUT SYSTEM AROUND SERIAL DATA TRANSMISSION

Some History

Six years ago we installed a small computer for simple data analysis in a cyclotron environment. This was a 12-bit machine, very small by most standards, but the wiring and cabling problems were formidable since typical installations in such an environment involve many peripherals - a plotter, a tape machine, a disc and many in-house produced input/output devices [1].

Next we stepped up to a larger machine - one with 18-bit words, and with the larger machine naturally came more complicated input/output [2,3]. The back sides of our equipment racks began to look like solid cables. We couldn't find circuit cards anymore for the cables running up and down the back of the computer, (see Fig. 6).

Finally, we stepped up to a 24-bit machine, gazed for a few solemn moments at the cabling problems, and settled into black despair. It was at just this time that a 20 MHz shift register became
readily available [4]. This triggered our serious consideration of serial data transmission schemes. The memory read-restore cycle time for our computer is 1.75 μs. To transmit 29 bits of digital information at a 20 MHz bit rate required 1.45 μs; less than the cycle time of our computer memory. Consequently, serial input/output data transmission was not in danger of becoming a time bottleneck. Were this not the case, we would still not have to suffer the consequences of going completely parallel to decrease transmission times. It is possible to trade off number-of-cables and transmission time. By using two data cables instead of one, and half the number of clock pulses per word transmission, the time can be halved. In the limit, the number of data cables could be increased to one-per-bit and the number of clock pulses decreased to one. This of course, is the fully parallel case.

A 4-bit shift register costs about $5.00. A 32-bit shift register costs 8 x 5 = $40.00. One is required at each remote point and one is required at the computer. Cabling costs would easily exceed this figure.

Table II summarizes our consideration. It was apparent that the only advantage to using parallel data transmission was transmission time. With a 20 MHz shift frequency being practical, this advantage diminishes considerably. Consequently we decided to try serial data transmission on our proposed system.
<table>
<thead>
<tr>
<th>CRITERION</th>
<th>PARALLEL</th>
<th>SERIAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Number of Wires</td>
<td>Approximately 200N, N being number of remote terminals. Increases Linearly with word size.</td>
<td>4 + N Coaxial Cables</td>
</tr>
<tr>
<td>2) Transmission Time</td>
<td>Typically 0.5 μs for any number of bits*</td>
<td>1.45 μs for 24-bit word with 20 MC clock. Increases linearly with word size*.</td>
</tr>
<tr>
<td>4) Flexibility in physical loc. of remote stations (for trouble shooting or to accommodate changing environments).</td>
<td>Extremely cumbersome to move any remote station. Requires special cables and connectors.</td>
<td>Trivial since the number of cables is very small and the cables and connectors are standard items.</td>
</tr>
<tr>
<td>5) Cost of making connection.</td>
<td>Substantial labor involved in making cables. Special drivers and receivers often required because of crosstalk and noise.</td>
<td>Very little labor involved in cable construction. No need for special receivers since noise and crosstalk are negligible. Special cable drivers may be required but very few are needed. A shift register is required at the computer and at each remote terminal.</td>
</tr>
</tbody>
</table>

\* For typical distances to peripheral devices, transit time will not be significant.
V. SYSTEM DETAILS

Figure 7 is the word structure decided upon. Twenty-four bits are devoted to information. A three-bit tag accompanies each word to identify it. The two most significant bits are set to "0", "1" before each transmission and checked when received in order to detect gross data transmission errors (Fig. 8).

Figure 9 illustrates the logic for generating clock bursts and connecting two shift registers; one transmitting and the other receiving.

Figure 10 is a block drawing of the entire system. The computer, as with most small computers, has two memory input/output routes. One is controlled by the central processor, which specifies a memory location and direction of data transfer. The other is controlled by the device, which specifies a memory address and direction of data transfer. In our system, the central processor may initiate a serial data transfer from memory with either of two instructions. The ACTivate instruction causes the instruction word itself to be transmitted, and causes the three tag bits to be set to code 7. The address field of this instruction contains the six-bit code assigned the device for which it is intended, and may contain additional information encoded in its least significant six bits. The write parallel (WTP) instruction addresses a computer memory location, and causes the contents of this location to be transmitted with a code 5 in the tag bits.
The normal sequence of operation is for the central processor to send a word (or several words) to a device to initiate an operation. The first word is an ACTivate command which will specify a device station, followed immediately by a write parallel (WTP) of a word-count/command word. Each station (unless it were in the midst of an operation) would receive these in its shift register, decode them, and either ignore them or act, depending on the results of the decoding. The station addressed would then periodically require access to the core memory. It would gate off the input to its shift register, possibly load a data word into its shift register and then request service from the multiplexer.

The multiplexer waits until it is not busy, and then responds by triggering a burst of clock pulses, preceded by a return signal on the direct connection to the remote terminal. The clock shifts the 29-bit word from the remote station into the 29-bit registers at the computer for a direct memory write. For a direct memory read, the computer memory is accessed first and the 24-bit data word is loaded into the computer shift register. The clock burst is then triggered.

The computer processor initiates only two modes of data transmission (ACT and WTP). Remote devices each have an individual connection back to the multiplexer at the computer. By initiating a signal on this connection, any device may trigger a clock burst from the computer, and may use this clock burst to return a word of data back to the computer. When the device loads such a data word, it also
loads a three-bit tag specifying what is to be done with the data word. It may be a word destined directly for some memory location (code 2) or it may be a status word to be read by the central processor (code 4) or it may be a status word which will interrupt the computer (code 6). It may also be an address in the computer memory which is to be incremented (code 3).

Code 1, the input/output clear code, is transmitted by the computer when the master clear button is pressed on the computer console, and serves to initialize all input/output devices.

Direct memory (DMA) operations require that the device supply a 15-bit address and a one-bit transfer direction code along with each 24-bit data word. If the address word were stored at the remote device, two serial transmissions would be required for each direct memory operation. We have instead included at the computer one 16-bit register for each DMA device (bits 0 - 14 are memory address, bit 15 is the read/write bit). When such a device requests a clock burst via its single unique connection to the computer and is given control, its address - register at the computer is gated onto the 15 computer memory-address lines. The sixteenth bit (bit 15) is gated onto the computer memory read/write line, and if it is set, a computer memory read is done before the 20 MHz clock burst is started. The clock burst then transmits the 24-bit data word read from memory. If a memory-write were demanded (by bit 15 of the register at the computer being reset) then the memory would not be accessed until
after the 20 MHz clock had returned with the data word.

There is the possibility, of course, that the device never wanted to perform a DMA operation, but instead wanted to interrupt the computer with a device-status word, or perhaps wanted to trigger one of the other options. The computer serial interface contains two shift registers. When one is transmitting, the other is always receiving. Consequently, the serial interface at the computer will detect any non-DMA code in the returning tag-bits and respond accordingly.

Referring to Fig. 10, note that all the remote stations are in series with each other. Each contains an amplifier for clock signals and one for data signals. The amplifiers are, of course, just transistor-transistor micrologic (T^2L) elements. For line drivers we are using discrete transistor circuitry (Fig. 11).

After traversing the loop, the clock is returned to the computer. If a remote station wishes to act as a data source, it operates its gate on the data line passing through it. The gate switches the remote station's shift register output onto the data line emerging from the station, disconnects the incoming data, and uses the incoming clock to shift out its data.

Gating onto this data line by any device is controlled by the "daisy-chain" multiplexer located at the computer. Figure 13 is a drawing of two stages of such a multiplexer. The function of the multiplexer is, of course, the prevention of more than one input
getting control at any one time. "Daisy-chain" refers to the fact that the stages are autonomous and more stages can be added on (or "daisy-chained") simply by connecting the priority line and busy line to the added stages.

Bi-directional communication is required between the computer and the remote station. The station must signal its desire for control and the computer must signal when control has been granted. This is accomplished via one cable per remote station, and some simple circuitry at each end. Referring to Fig. 12, a request for control is made by raising the input line (driving the base of Q1). This drags the voltage across the coaxial cable down to about 1 volt, and lowers the sense line connected to the collector of Q3. This is detected at the computer and becomes the input to a multiplexer stage. The computer acknowledges control by hogging current from the collector circuit of Q1 via the cable connection between the computer and the station. This current hogging eliminates most of the emitter current flowing in Q1 and Q2, thereby turning Q3 off and raising the sense line. This rising edge is the start of the time slice granted to this remote station. The end of the slice occurs when the end of the shift burst is detected, as shown in Fig. 12.

Figure 14 contains two photographs of the operation of this communication scheme. The bottom traces in both photographs are the signal on the bi-directional communication line (in this case, 35 feet of cable). The top trace in Fig. 14(a) is the request-in-progress
signal at the computer (Fig. 12). The top trace in Fig. 14 (b) is the signal at the collector of transistor Q3 (Fig. 12).

Figure 15 contains a photograph of a complete request cycle. The top trace is the 20 MHz burst (containing 29 pulses) which was initiated by the signal on the bi-directional communication line (bottom trace).

Figure 16 contains waveform photographs for the burst-detector in Fig. 12. In Fig. 16 (a), the top trace is the burst; the bottom trace is the signal at the output of the slow (D.T.L.) inverter. Figure 16 (b), top trace, is the end of the burst. The bottom trace is the pulse generated to signal the end of the burst.

Figure 17 (b) is a closeup of a data word returning from a device. The least significant bit is transmitted first. Consequently, to read the word, read from the right-hand side of the picture. The data word is: ¥1 (error-check) 110 (code 6; interrupt request) ¥10 ¥00 ¥00 ¥00 ¥00 ¥00 ¥00 ¥00 (the data word). Figure 17 (c) is a closeup of a part of this word.

Finally, Fig. 17 (a) is another data word (tag code 7, ACT), this one emerging from the computer. The bottom trace in Fig. 18 is the clock emerging from the computer. The top trace is the clock returning after passing through five devices.
Currently operational devices in the system at this time are a CAMAC bin, an IBM compatible tape controller, a remote editing facility (display, keyboard, joy stick), a card reader, a printer, and a disk memory. We have had considerable experience with serial operation at this point and have been able to substantiate our original contentions about its inherent elegance. In addition to the considerable advantages which we originally hypothesized to convince ourselves to try the scheme, many operating conveniences have appeared.

1) Although any defective station in the chain can shut the system down by preventing the flow of data or clock, bypassing such a station is simply a matter of disconnecting and joining four coaxial cables - a very trivial operation!

2) Debugging at any station is simplified by having to worry about only five coaxial cables when pulling a unit out of its rack for troubleshooting.

3) The computer communicates directly only with shift registers, which are located at the computer! The advantage is the greatly reduced probability of fouling the computer from cable ends at a remote terminal. Wires which can easily foul normal computer operation are all terminated at the computer shift register in a carefully designed common interface which prevents such interference. In other words, peripheral devices now all have to deal with a single box which in
turn must deal with the computer. This extra level of isolation adds protection to the operating central processor.

Direct connections to one particular machine which is in very general use today can set bits in the accumulator at any time with the simplest sort of malfunction. Such gross interference has been effectively prevented.

4) The elimination of noise and crosstalk problems seems to have been total - a result of being forced to use coaxial cables for communicating.

5) We can once again see our way through our input/output cabling (Fig. 6).
ACKNOWLEDGMENTS

I wish to acknowledge the valuable assistance of Fred Goulding, Don Landis, John Kopf, Fong Gin and Vicky Donelson. Photographs by Steve Gerber.

FOOTNOTE AND REFERENCES

* Work done under the auspices of the U. S. Atomic Energy Commission.


FIGURE CAPTIONS

Fig. 1. Capabilities of a readily available 20 Mhz shift register.

Fig. 2. Pulse width modulation (one-wire system).

Fig. 3. Pulse frequency modulation (one-wire system).

Fig. 4. Two-wire (clock and data) transmission.

Fig. 5. Hor: 50 ns/cm, Vert: 2V/cm - Part (a) is the untrimmed time relationship between clock and data. Part (b) shows the same signals after delaying the clock relative to the data with a section of transmission cable. This improves operating margins for the receiver.

Fig. 6. (a) Before - input/output cabling from an 18-bit word length computer to four peripheral devices (fully parallel).

(b) After - input/output cabling from a 24-bit word length computer to five devices (two-wire serial).

(c) Fully parallel connections at the device end of the computer system in Fig. 6 (a).

(d) Connection to a CAMAC bin via serial data transmission. Computer cables are at the left.

Fig. 7. Shift register word structure.

Fig. 8. Error Checking.

Fig. 9. Basic serial data transmission scheme.

Fig. 10. Complete input/output scheme.
Fig. 11. In-line data and clock amplifiers at remote stations.

Fig. 12. Multiplexer - remote station communication.

Fig. 13. Two stages of a "daisy-chain" multiplexer.

Fig. 14. Hor: 100 ns/cm, Vert: 2V/cm - Part (a) is the relationship between a remote device driving its line to the multiplexer (bottom trace) and the output of the receiver at the multiplexer (top trace). Part (b) is the response of the receiver at the remote station end of this bi-directional connection (top trace).

Fig. 15. Hor: 500 ns/cm, Vert: 2V/cm - Bottom trace is the signal on the bi-directional communication line to the multiplexer. The top trace is the burst triggered by this signal. (Note that we are not operating quite at 20 MHz.)

Fig. 16. (a) Hor: 500ns/cm, Vert: 2V/cm
(b) Hor: 200ns/cm, Vert: 2V/cm
Part (a) is the burst (top trace) driving a slow (DTL) inverter. The bottom trace is the output of the inverter. Part (b) is the end of the clock (top trace) and the end-of-clock signal (bottom trace).

Fig. 17. (a) and (b) Hor: 200 ns/cm, Vert: 2V/cm
(c) Hor: 50 ns/cm, Vert: 2V/cm
Parts (a) and (b) are data words emerging from the computer and a device, respectively. Error checking bits are the right-most two bits ($\emptyset$,1) and the tag bits appear next
(lll in (a), llø in (b). Part (c) is a closeup of a part of one of the data words.

Fig. 18. Hor: 250 ns/cm, Vert: 2V/div. - This picture is the clock transmitted from the computer (bottom trace) and the same clock returning (top trace) after passing through five devices. Transit time is approximately 500 ns.
KEY WORDS

FIG. 1
CAPABILITIES OF A READILY AVAILABLE
20 MHz SHIFT REGISTER

XBL 702-325
TIME
PULSE WIDTHS (A)

MONO-STABLE SHORT PERIOD
MONO-STABLE LONG PERIOD

CLOCK
SHIFT REGISTER
SERIAL DATA
CODING

BOTH FALLING EDGE TRIGGERED (B)

MONO-STABLE OUTPUT
DECODING
STROBE TIMES

RISING EDGE TRIGGERED
WIDTH SHORTER THAN "1" TIME, LONGER THAN "0" TIME

MONO-STABLE
DATA
SHIFT REGISTER

SHIFT OCCURS ON TRAILING EDGE
CLOCK (C)

Fig. 2
PULSE WIDTH MODULATION (ONE WIRE SYSTEM)

XBL 702-322
Pulse Frequency Modulation (One-Wire System)

Fig. 3

**A**
- Clock pulses
- Data pulses
- Rising-edge triggered
- Clock pulse (could be made negative pulses)
- Output pulse train

**B**
- Coding
- Shift register
- Mono-stable long period
- Mono-stable short period
- Both falling triggered

**C**
- Input pulse train
- Toggle bistable
- Mono-stable output
- Strobe times

**Description**
- The diagram illustrates the operation of a shift register and clock pulses in pulse frequency modulation.
- The clock pulses trigger the shift register, which changes states based on the data pulses.
- The diagram shows the timing and relationship between clock and data pulses, emphasizing the rising-edge triggering mechanism.
- The output pulse train results from this interaction, with the possibility of using negative pulses instead of positive ones.

**Notes**
- The diagram is a detailed representation of pulse frequency modulation, a method for transmitting information over a single wire.
- The key components include the clock pulse generator, the data input, and the mono-stable stages that control the shift register.

**Technical Details**
- The diagram is labeled with symbols and notation typical in electronic circuit diagrams.
- The wiring and logical flow are clearly depicted, allowing for a comprehensive understanding of the system's operation.

**Reference**
- XBL 702-326
FIG. 4
TWO-WIRE (CLOCK AND DATA) TRANSMISSION

XBL 702 328
Fig. 6(a)
Fig. 6(b)
TWO BITS FOR ERROR CHECKING

THREE BITS FOR FUNCTION DEFINITION

\[ \phi \ 1 \]

24 INFORM. BITS

DECODED AT RECEIVERS

\( \phi \) - NO OPERATION
1 - INPUT/OUTPUT CLEAR
2 - DIRECT MEMORY OPERATION
3 - INCREMENT A MEMORY LOC
4 - 24 BIT XFER/CPU CONTROL/WRITE MEMORY
5 - 24 BIT XFER/CPU CONTROL/READ MEMORY
6 - ASK FOR INTERRUPT
7 - ACTIVATE A DEVICE

FIG. 7
SHIFT REGISTER WORD STRUCTURE

XBL 702-324
WHEN THIS SHIFT OCCURS, THE ERROR BITS ARE PASSED THROUGH THE HOME SHIFT REGISTER.

WHEN THIS SHIFT OCCURS, THE REMOTE REGISTER IS CHECKED.

THESE BITS SHOULD ALWAYS BE $\phi$, I AT THE END OF EVERY TRANSMISSION AND IN EVERY SHIFT REGISTER USED.

FIG. 8:
ERROR CHECKING

XBL 702-323
AT THE COMPUTER

29 BIT
SHIFT REGISTER
CLOCK

20 MC
GATED
OSC.

5 BIT
COUNTER

RESET TO
START BURST

DECODE

TURN ON
OSCILLATOR
WHEN
COUNTER IS RESET.

TURN OFF
CLOCK AFTER
29 COUNTS.

AT THE REMOTE TERMINAL
(TAPE MACHINE, DISPLAY
TERMINAL, PLOTTER,
DISC, ETC.)

29 BIT
SHIFT REGISTER
CLOCK

CLOCK - BURSTS OF
29 PULSES
AT 20 MC.

FIG. 9
BASIC SERIAL DATA TRANSMISSION SCHEME

XBL 702-317
FIG. 10
COMPLETE INPUT/OUTPUT SCHEME
FIG. II
IN-LINE DATA AND CLOCK AMPLIFIERS AT REMOTE STATIONS
FIG. 12
MULTIPLEXER - REMOTE STATION COMMUNICATION
FIG. 13
TWO STAGES OF A "DAISY-CHAIN" MULTIPLEXER

XBL 702-320
Fig. 17 (c)
This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.