Lawrence Berkeley National Laboratory
Recent Work

Title
SERIAL INPUT/OUTPUT FOR SMALL COMPUTERS

Permalink
https://escholarship.org/uc/item/890561n2

Author
Meng, J.D.

Publication Date
1969-07-01

eScholarship.org Powered by the California Digital Library
University of California

Submitted to IEEE Transactions
on Computers

UCRL-19252
Preprint

c. 2

SERIAL INPUT/OUTPUT FOR SMALL COMPUTERS

RECEIVED
LAWRENCE
RADIATION LABORATORY
AUG 25 1969
DOCUMENTS SECTION

J. D. Meng

July 1969

AEC Contract No. W-7405-eng-48

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 5545

LAWRENCE RADIATION LABORATORY
UNIVERSITY of CALIFORNIA BERKELEY

UCRL-19252

2

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.

SERIAL INPUT/OUTPUT FOR SMALL COMPUTERS

J. D. Meng

Lawrence Radiation Laboratory
University of California
Berkeley, California

July 1969

Abstract——A computer input/output scheme is described which has at its heart serial data transmission to and from a computer installation. The scheme is practical today because of inexpensive, fast, available shift-registers and associated logic. Its major advantage is the use of only four + n coaxial cables to connect the computer to n peripherals or remote terminals. The CAMAC system is discussed as a standard digital interface for peripherals in a small computer system.

I. CONSIDERATIONS

Five 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, 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.

Finally, we stepped up to a 24-bit machine, gazed for a few solemn moments at the cabling problems, and settled into black despair.

A 20 MC shift register came to our attention about this time,[4] however, and it became apparent that we would be able to handle all our 24-bit input/output over just five coaxial cables, as opposed to the 200 or so individual wires which would be required in a straightforward, parallel installation.

Some simple arithmetic signalled the practicality of the scheme. Memory cycle time on our 24-bit machine is 1.75 μs. Transmission of a 29-bit word serially at 20 MC requires (29 x 50) ns - 1.45 μs, less than one memory cycle-time. 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 I, Comparison of Serial and Parallel Data Transfers, weights the argument heavily in favor of serial data communication. The practicality in any system is in fact determined by the transmission time of a word of data. Until very recently, this was prohibitively long. Indeed, in some applications, it may still be marginally too long. It is possible, of course, to trade off number-of-cables and transmission time. By using two data coaxes instead of one, and half the number of clock pulses per burst, the transmission 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 is the fully-paralled case. For any system, an optimum data-transmission scheme may be something between these endpoints. Fully-parallel transmission is expensive but fast. Fully-serial transmission is inexpensive but slower. Today, when 20 MC clock rates are practical, the scale for measuring optimum performance per dollar is tilted in the serial direction.

TABLE I

COMPARISON OF SERIAL AND PARALLEL DATA TRANSFERS

CRITERION	PARALLEL	SERIAL
1) Number of Wires	Approximately 200N, N being number of remote terminals. Increases Linearly with word size.	4 + N Coaxial Cables - Constant with word size change
2) Transmission Time	Typically .5 μs for any number of bits*	1.45 μs for 24-bit word with 20 MC clock. Increases linearly with word size*.
3) Crosstalk and Noise	Major problem. Severely limits transmission distances.	Not a problem for any practical transmission distance. Delay of clock relative to data is fairly critical.
4) Flexibility in physical loc. of remote stations (for trouble shooting or to accomodate changing environments).	Extremely cumbersum to move any remote station. Requires special cables and connectors.	Trivial since the number of cables is very small and the cables and connectors are standard items.
5) Cost of making connection.	Substantial labor involved in making cables. Special drivers and receivers often required because of crosstalk and noise.	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.

* For typical distances to peripheral devices, transit time will not be significant.

A. Practical Operating Example

The basic transmission scheme is blocked out in Fig. 1. It consists of a gated oscillator, a 5-bit counter and decoding gate, and the shift registers. The technological developments which have made this practical are:

1) Availability of 20 MC, 4-bit TTL shift registers in a single can. This shift register may be loaded or unloaded either serially or in parallel, and in fact has gating on the parallel inputs, reducing in many cases the number of external gates required. The current price of these shift registers is not prohibitive, being in the range of $5 - $10. A 29-bit register (4 bits per can) costs about $50.00.

2) Availability of inexpensive TTL micrologic which is compatable with the shift registers. The standard TTL output is 130 Ω in the collector of a saturated emitter - follower. This is nearly a perfect match to a 125 Ω coaxial cable, a standard, readily-available coax. We use 29-bit registers to transmit 24 bits of data to allow for transmission error-checking, and also to allow three bits for sending an identifying tag with the 24-bit data word.

Our computer, similar to many machines, has two basic input/output routes. One is via the central-processor (which supplies the memory address), and the other is directly into or out of memory (the external station must supply both an address and the data). We are using the same cables for both routes. The entire system is blocked out in Fig. 2. There are two cables returning with data and the shift clock. Then there is one cable required to connect each remote station to the daisy-chain multiplexer at the computer. The total number of cables from the computer equals: $4+n$,

where n is the number of remote stations.

The critical part of the system, of course, is the relative delay in the signal and clock channels. Gross inequalities in cable lengths must be avoided, as well as gross differences in signal paths through amplifiers.

The amplifiers at the remote stations (Fig. 2a) are just TTL gates, making them simple and inexpensive. The input-gate boxes, similarly, are TTL circuits. Gates driving the data coax must be discrete since it is necessary to drive data from any of the station points (including the computer), and when a station is not transmitting, it must not load the line.

Any defective station in the chain can shut the system down by preventing the flow of data or clock. However, by-passing such a station is simply a matter of disconnecting and joining four coaxial cables - a very trivial operation!

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.

More detailed considerations will reveal more advantages. 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 word structure is depicted in Fig. 3. To initialize a device, the programmer might give an ACTIVATE command which will specify which station, followed immediately by a memory-location word, and then followed by a word-count/command word. Each station (unless it were in the midst of an operation) would receive these words 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, load a memory address into the shift register with the proper tag in the three function bits, and then request service from the multiplexer.

The multiplexer waits until it is not busy, and then responds with a burst of clock pulses, preceeded 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. The three function bits with the word are decoded after being received, and the desired computer operation is then initialized.

At the completion of the operation required by the command sequence, the remote station signals the computer by putting the interrupt code into its shift register (bits 2-4, Fig. 3) and asking the multiplexer for control. The computer responds with a clock burst when control is granted, and interprets the word which reaches it as an interrupt request. The remote station is identified from the multiplexer (not from coding in the data word) and the 24-bit data word may be read by the computer to determine the specifics of the interrupt request (such as Ready, End of File, End of Tape, etc.).

The point here is that the computer communicates directly only with shift registers, which are located at the computer! The advantage is the relative impossibility 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, peripherals now all have to deal with a single box which in turn must deal with the computer. The 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 is very effectively prevented using this serial scheme).

In addition to the interesting features of the serial data-transmission scheme, the multiplexer is interesting in that a single cable connection is used to request service from the computer interface by the remote station, return an answer to the remote station, and also to signal the computer from the remote station when a data transfer is required. This is all done with the circuits in Fig. 4. The request is made by the remote terminal clamping the voltage down on the line. The request is acknowledge by having the computer end hog current from the remote-terminal transmitter. The computer then releases the line, and the remote station signals data-ready by removing the request.

Some comments should be made concerning the error-checking scheme we have tentatively adopted. We considered a parity tree at each shift register, but thought perhaps this involved too much hardware to be practical - particularly since our data motions were all serial. A straight-forward scheme of using data-signal transitions to toggle a flip-flop as the shifting occurred was also considered. (This is equivalent to a parity check, but on serial data.) This technique does not involve much extra hardware, and in fact we have left provisions for using it should our present scheme prove inadequate. What we have decided on is the use of two bits which are always preset to (Ø, 1) at the initiation of a transmission (Fig. 5). They are checked at the end of each transmission for the same code.

The two bits are passed completely through the transmitting shift register at each transmission. We are assuming, of course, that failures will be more substantial than sporadic (noise), and the use of coax for all our interconnections supports this contention.

A detected error at the computer sets a flag which can be sensed by program. At the remote station, the technique for the handling of errors is chosen by the station designer. The tape station, for example, sends back to the computer an error-flag message with the interrupt request.

One last point for consideration is the counter decoder which determines the number of shift pulses (Fig. 1). Our gated oscillator produces one pulse after the gate has been lowered. Consequently, the decoder must not turn the pulser back on when this extra count occurs, and must detect a "$\geq$" condition rather than just an "=" condition. We have done this simply by using 28 + 1 pulses.

$$28_{10} = 34_8 = 011\ 1ØØ_2$$

Our decoder detects the Ø111 and ignores the least significant two bits.

This has been an over view of our system - not a detailed description or analysis. We felt it should be given consideration because of its inherent elegance and simplicity, and our consideration has at this point paid substantial dividends.

We are now developing a general interface for use with the CAMAC standard (formerly called IANUS) for nuclear instrumentation [5].

II. USE OF THE CAMAC STANDARD FOR PERIPHERAL INTERFACES

The features of the CAMAC standard are covered fairly completely in the Hooton article [5]. Consequently, I will not attempt to detail the scheme. Generally, though, it is a bin containing 25 slots and a well-defined set of busses and interconnections. It is designed so that the modules in the end position can multiplex operations on the busses, and can also control bin interactions with any computer. The bin is becoming a nuclear instrumentation standard, having already won acceptance in Europe. The need for such a standard is, of course, dictated by the wide variety of special-purpose equipment (developed in various laboratories) which could be used in many other laboratories. By defining the interface (which can be driven by any commercial computer), such special-purpose equipment becomes instantly operable in all laboratories using the standard.

In typical small-computer operation where the variety of special peripheral connections may be very large, such a standard interface has substantial value. If, for example, a particular disc or a particular tape machine were interfaced to a standard rather than being interfaced to some individual computer's idiosyncrosies, the controller for the peripheral could be used unchanged with any computer which was interfaced to the standard.

Thus, the number of computer interfaces necessarily designed for each different computer type is reduced to one rather than one-per-peripheral. From the user's standpoint, this is an ideal situation.

He is free to choose the best available computer instead of merely the one with the best peripheral complement. He is freed from the tedium of redesigning all his individual interfaces everytime he changes computer types. He is also able to use interface designs from a larger group of users - not just those who by chance have the same type of computer he has. Of course, computer manufacturers might be expected to vehemently oppose such a user-dictated standard, since they clearly would like to have their particular input/output scheme adopted as a standard, with the accompanying competitive advantage when such a user goes out to purchase his next machine.

Defining a standard for use in interfacing peripherals poises more problems than defining a standard for use just with nuclear instruments. However, CAMAC has been designed with enough care that it is easily adaptable to peripheral interfaces.

First, peripherals are not always located close enough to each other to make it desirable for controllers for each device to be in very close proximity to each other. Second, the computer must be interfaced to the bin standard, and the bin controller multiplexes the contents of its bin. For high-data transfer-rate devices (such as a disc), and where multiple bins must be themselves multiplexed at the computer (in addition to multiplexing within each bin), the multiplexing time build-up may get prohibitive. If a device requests computer service from a position within a bin and must wait (1) for the bin to be free, (2) for the bin to request service from the computer and (3) for the computer to be free, the time interval from the initial request to securing computer service may get too long for the device to tolerate. The obvious solution to these problems is the use of "single-device bins". These need not be bins at all, but merely device controllers containing standard CAMAC connectors for connection to the computer - CAMAC interface. Thus, we may retain the multiple advantages of standardization without sacrificing flexibility or response time.

Another problem is that generated by differing word length of different computers. How is it possible to build a peripheral controller to a standard interface without first knowing the word length required?

The answer is that we build two boxes - one for register functions and one for device-control functions. Register functions are word-length oriented (not computer oriented) and may include a word counter, computer-address counter and word buffer. This box may be common to most of the devices in a system. The device-control functions form the interface of the particular device to the CAMAC standard.

Thus far this section has been hypothetical and seems little related to our serial input/output scheme, it is very much related however, because we have in our system under development elected to develop a single interface - serial to CAMAC - and then produce our device controllers (disc, card reader, printer, micro-tape) to CAMAC standards.

This practical application of the CAMAC standard has proven itself nicely workable. Indeed, we anticipate progressively greater advantages as CAMAC becomes more familiar as a design standard and as we move on to our next computer.

FOOTNOTE AND REFERENCES

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

[1] L. B. Robinson, "Applications of Small Computers in Nuclear Spectroscopy," IEEE Transactions on Nuclear Science, Vol. NS-13, No. 1, pp. 161-166, February 1966.

[2] L. B. Robinson, J. D. Meng, "On-Line Real-Time Time-Sharing Operation of a PDP-7," Digital Equipment Computer User's Society' (DECUS) proceedings, Fall 1966, pp. 53-56. UCRL-17220, Lawrence Radiation Laboratory, Berkeley, California, March 1967.

[3] L. B. Robinson, "The Computer in your future, Part II," Scientific Research, pp. 65-74, September 1967.

[4] Texas Instruments, New Product Bulletin on Types SN5495N, SN7495, SN7495N "4-Bit Right-Shift Left-Shift Register,"March 1968.

[5] I. N. Hooton and R. C. M. Barnes, "A Standardized Data Highway for On-Line Computer Applications," Proceedings of the Fall Joint Computer Conference, Vol. II, pp. 1077-1087, 1968.

FIGURE CAPTIONS

Fig. 1. Basic Serial Data Transmission Scheme.

Fig. 2. Computer Input/Output Scheme.

Fig. 3. In-Lin Data and Clock Amplifiers at Remote Stations

Fig. 4. Shift Register Word Structure

Fig. 5. Multiplexer-Remote Station Communication.

Fig. 6. Error Checking.

KEY PHRASES

1) Small computer installations.

2) Serial Input/Output system.

3) Computer central-processor isolation.

4) Bi-directional DC communications.

5) Daisy-chain multiplexer.

6) Noise levels and crosstalk.

7) CAMAC standard digital interface for peripherals.

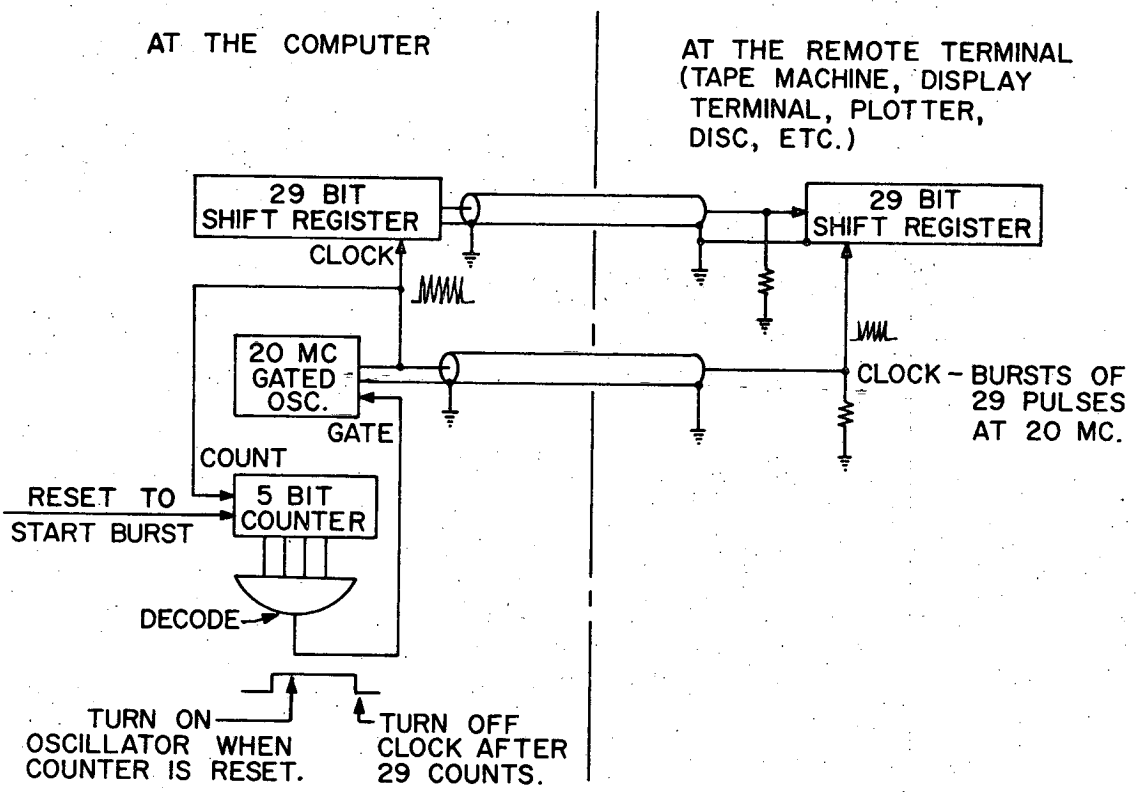


FIG. 1
BASIC SERIAL DATA TRANSMISSION SCHEME

XBL 692-185

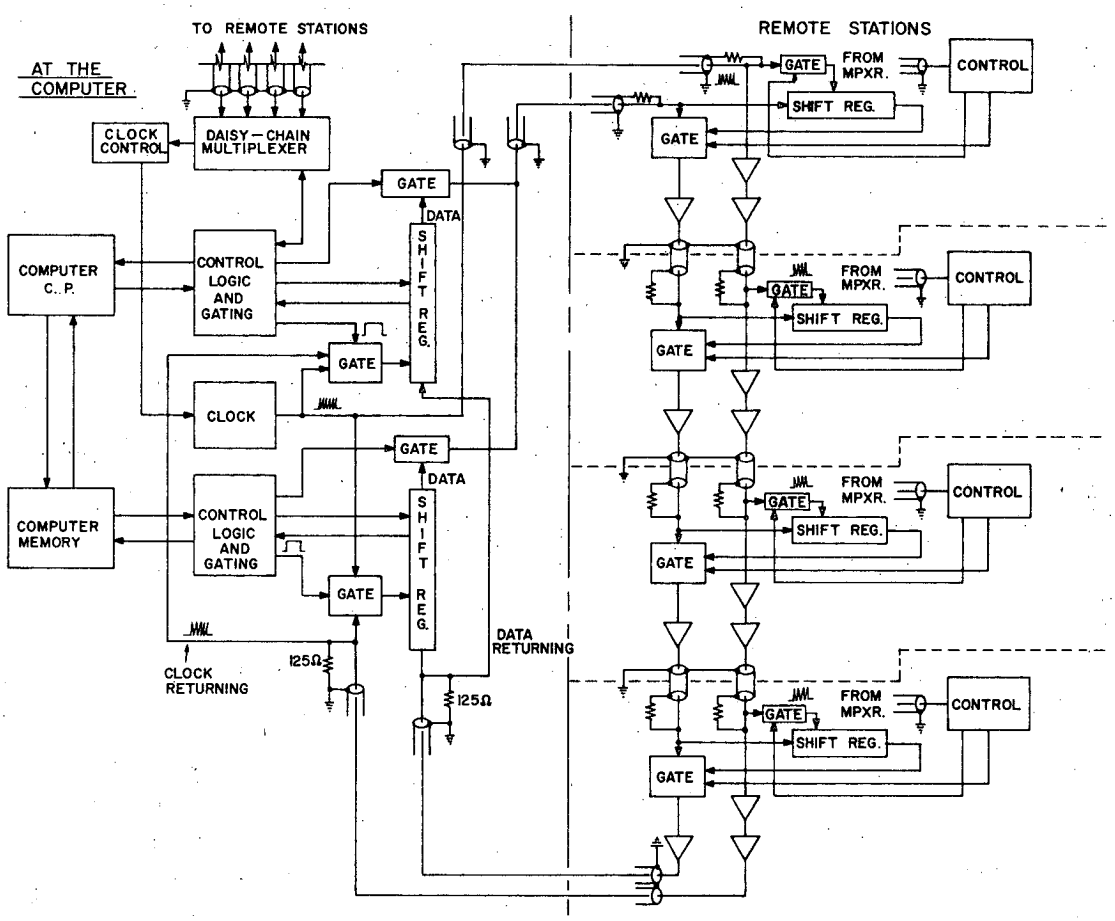


FIG. 2:
COMPLETE INPUT/OUTPUT SCHEME

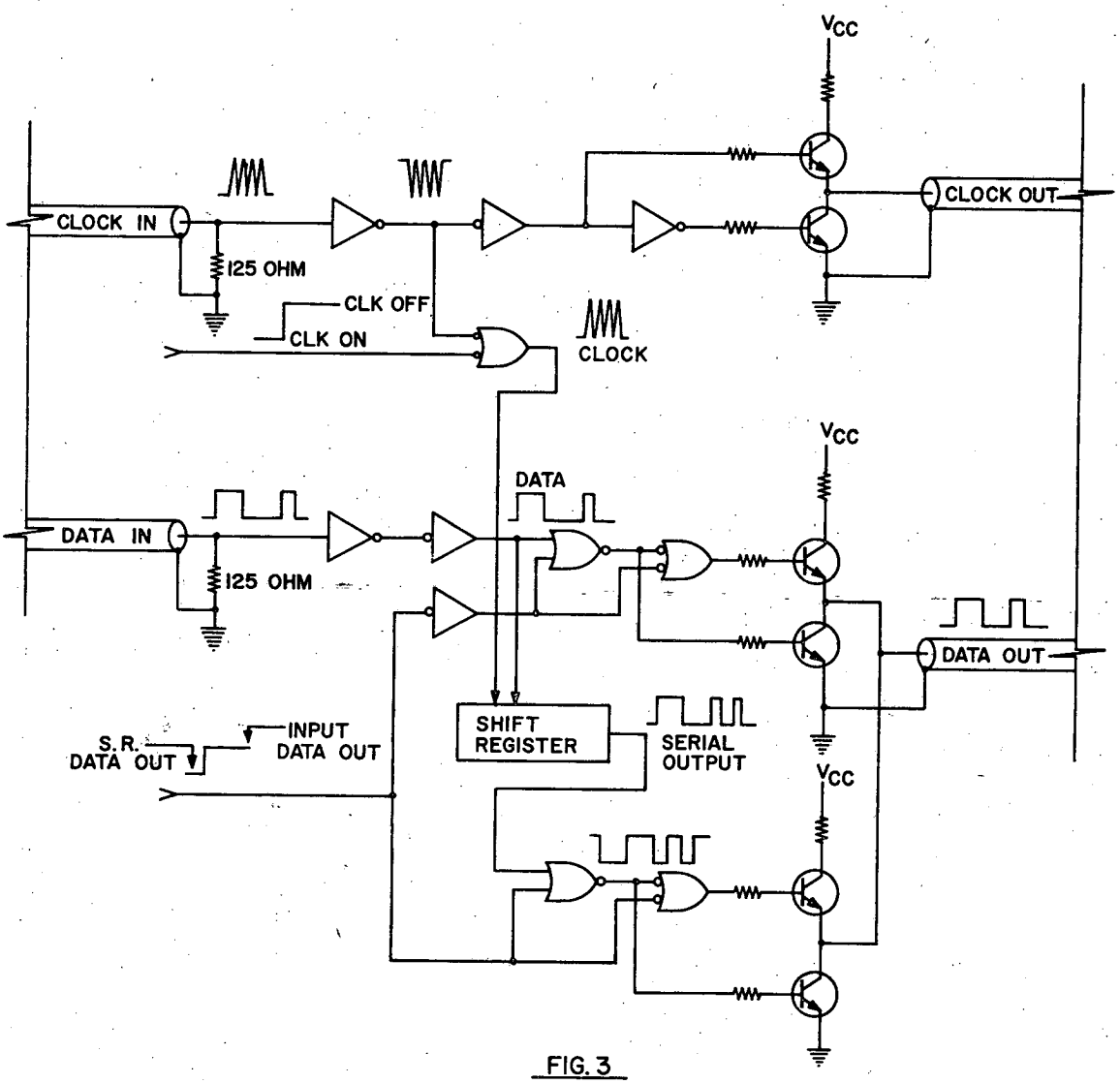
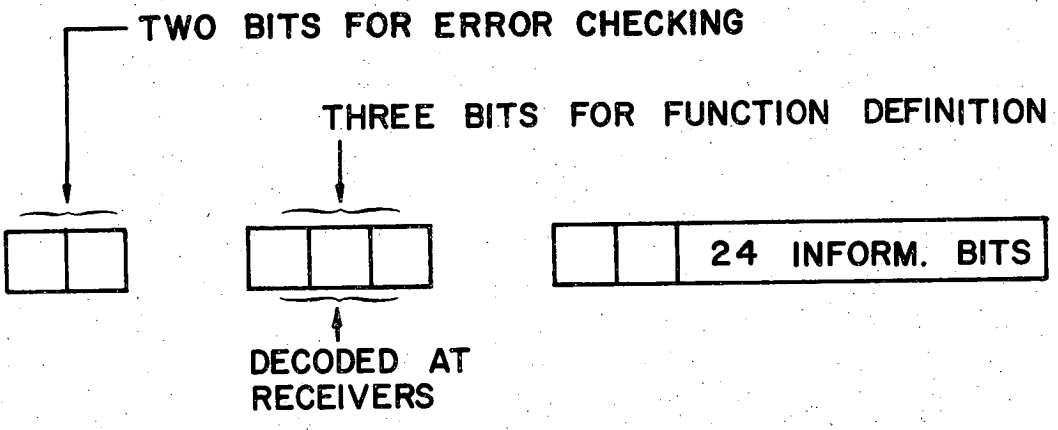


FIG. 3
IN—LINE DATA AND CLOCK AMPLIFIERS AT REMOTE STATIONS

XBL 697-868



ϕ - NO OPERATION
1 - DIRECT MEMORY READ
2 - DIRECT MEMORY WRITE
3 - INCREMENT A MEMORY LOC
4 - 24 BIT XFER/CPU CONTROL/READ MEMORY
5 - 24 BIT XFER/CPU CONTROL/WRITE MEMORY
6 - ASK FOR INTERRUPT
7 - ACTIVATE A DEVICE

FIG. 4
SHIFT REGISTER WORD STRUCTURE

XBL 692-183

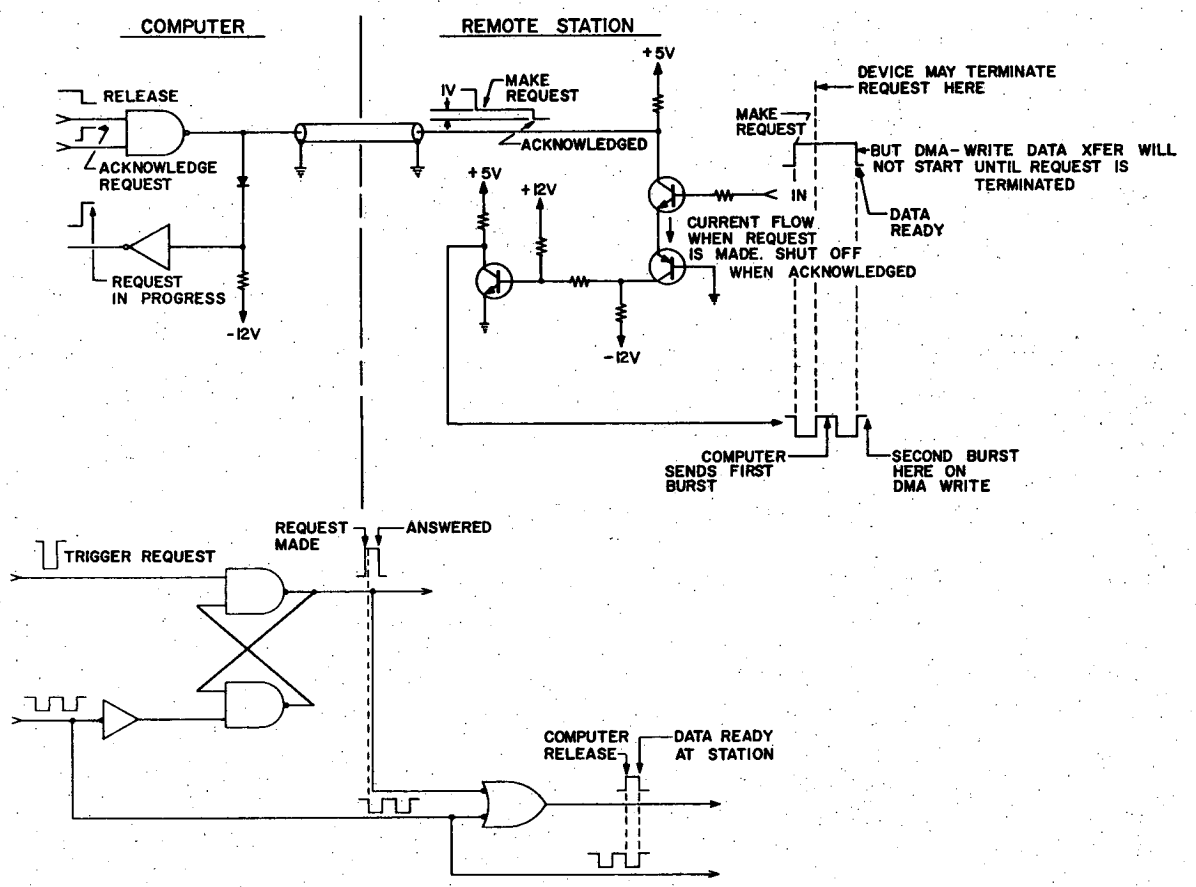


FIG. 5
MULTIPLEXER – REMOTE STATION COMMUNICATION

XIII 692-184

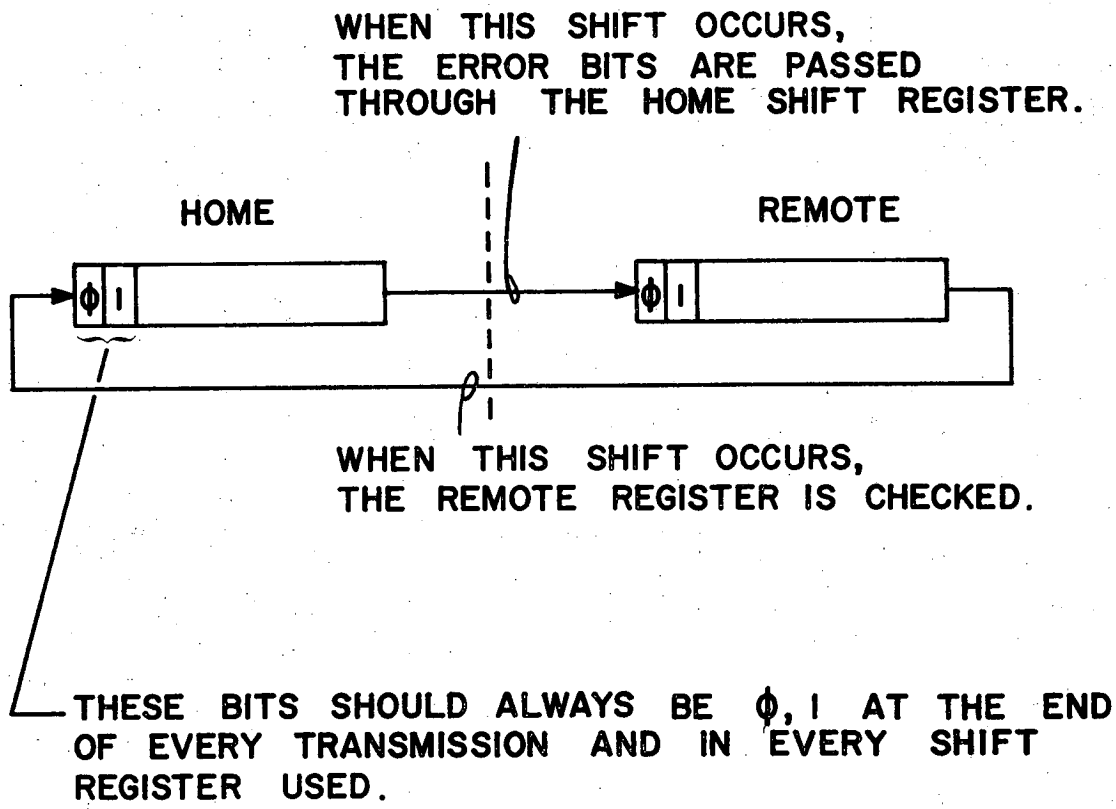


FIG. 6:

ERROR CHECKING

XBL 692-219

LEGAL NOTICE

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.

TECHNICAL INFORMATION DIVISION
LAWRENCE RADIATION LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720