AUTOMATIC ADDRESSING OF TELEMETRY CHANNELS[*]

L. A. Lucero
Sandia National Laboratories
Livermore, California 94550

ABSTRACT

To simplify telemetry software development, a design that elminates the use of software instructions to address telemetry channels is being implemented in our telemetry systems. By using the DMA function of the RCA 1802 microprocessor, once initialized, addressing of telemetry channels is automatic, requiring no software. In this report, the automatic addressing scheme is compared with an earlier technique that uses software to address telemetry channels. In comparison, the automatic addressing scheme effectively increases the software capability of the microprocessor, simplifies telemetry dataset encoding, eases data set changes, and may decrease the electronic hardware count. The software addressing technique uses at least three instructions to address each channel. The automatic addressing technique requires no software instructions. Instead, addressing is performed using a direct memory access cycle stealing technique. Application of an early version of this addressing scheme opened up the capability to execute 400 more microprocessor instructions than could be executed using the software addressing scheme. The present version of the automatic addressing scheme uses a section of PROM reserved for telemetry channel addresses. Encoding for a dataset is accomplished by programming the PROM with channel addresses in the order they are to be monitored. Software for one of our telemetry units was written using the software addressing scheme, then rewritten using the automatic addressing scheme. While 1000 bytes of memory were required by the software addressing scheme, the automatic addressing scheme required only 396 bytes. A number of prototypes using AATC have been built and tested in a telemetry lab unit. All have worked successfully.

INTRODUCTION

To provide for a more efficient method of gathering information, a microprocessor-based telemetry system is used to select only relevant information at any given time based on in-flight analysis of data. The first programs used software instructions to address each channel transmitted and synchronized channel selection with the telemetry system. In

[*] This work supported by the U. S. Department of Energy.

developing these first software programs, it became apparent that addressing telemetry channels through software instructions was awkward; it required many software instructions and tedious data formatting.

In software addressing of telemetry channels (SATC), software is used to send an address stored in Programmable Read-Only Memory (PROM) to the Data Acquisition Systems (DASs) to select an analog input channel for analog-to-digital conversion. This requires reading the address from the PROM into the microprocessor, and then outputting the address from the microprocessor to the DASs.

Obviously, it would be more efficient to send the addresses stored in PROM directly to the DASs. A search for such a method resulted in the automatic addressing of telemetry channels (AATC) scheme.

MICROPROCESSOR-BASED TELEMETRY

Our telemetry systems use the RCA 1802 microprocessor and two custom large scale integrated circuits (LSIs), the DAS, and the control LSI as basic building blocks. Transmission is in pulse-code modulation. Figure 1 shows a telemetry system block diagram.

The control LSI formats the output data of the DASs for transmission. The DAS handles the telemetry data channels, performing analog-to-digital conversions on analog information from transducers. The DAS and control LSI are initialized and controlled by the microprocessor, the brain of the system.

Each DAS is capable of handling 24 telemetry DAS channels, designated 1 through 24. A DAS channel is selected for conversion by sending its address to the DAS multiplexer. The parallel load (PRLD) signal is a control LSI-generated timing signal. To maintain a continuous flow of data, each DAS in the system is addressed to do a conversion immediately after a PRLD signal.

Addresses to the DASs are assigned so they can be addressed independently. An address can be sent out for each DAS, each address selecting a different channel, or one address may be sent to a group of DASs for selection of a common channel. For example, if Channel 3 of each of the DASs in a two-DAS system is to be selected, one address will handle both DASs.

Telemetry information is gathered through a sequence of datasets. Each dataset contains a group of ordered sampled telemetry channels. The set of sampled channels is transmitted repetively using a “sync” word for identification and synchronization. During a flight the telemetry sequences through a series of different datasets. The microprocessor switches from one dataset to another based on the signals it monitors.

AUTOMATIC ADDRESSING OF TELEMETRY CHANNELS

AATC does telemetry channel addressing under hardware control by using the 1802 direct memory access (DMA) function. When the 1802-DMA function is asserted, the contents of the R∅ reqister are placed on the address lines, then the contents of R∅ are incremented by one. (The microprocessor performs a DMA cycle in 4 µs.) All of this occurs invisible to the software. Software will continue executing as if DMA never occurred. The DMA function is asserted by placing a low logic level (0) on the DMA-IN pin of the 1802(l). Thus, by using the PRLD signal to drive the DMA-IN pin, the DMA function is asserted each time a PRLD signal occurs (see Figure 2).

A repeated sequence of contiguous addresses A_M, A_{M+1}, ... , A_N is generated as follows: 1) R∅ is initialized by software to the first address in the sequence, A_M; 2) each time PRLD occurs DMA is asserted, placing the contents of R∅ on the address lines, the contents of R∅ are incremented by one; 3) software is used to determine when the last address, A_N, in the sequence occurs. Immediately after A_N is placed on the address lines, software repeats step one, reinitializing R∅, and repeating the sequence.

Early telemetry units used this scheme, in selected datasets, to generate a sequence of continuous addresses to the DASs, using the standard configuration shown in Figure 1. This has the advantage of using no memory to address channels. The disadvantage is that it is restricted to datasets that can be arranged so they are addressed by a contiguous sequence of addresses, one address per PRLD. Unfortunately, most datasets do not fall into this category.

Adding a PROM to the telemetry system between the 1802 address lines and the DAS address lines (see Figure 3) eliminates the restrictions imposed by addressing the DASs directly with the 1802 address lines. The PROM is programmed with addresses to the DASs in the order they are to be generated. The output of the PROM goes to the DAS address lines. A repeated sequence of contiguous addresses, generated as described previously, is used to address the PROM.

To send more than one address to the DASs between PROMS, one of the bits, b_n, in each PROM memory location is reserved as a flag to reassert DMA (see Figure 4). If this bit is programmed high at the PROM address location A_M, THE output bit b_n will be high when A_M is selected. DMA is then immediately repeated, generating the DAS address at location A_{M+l} of the PROM. If bit b_n is programmed low, DMA will not occur until the next PRLD. In a four-DAS system, between any two PRLDs each DAS may be addressed to monitor a different channel by programming the bit b_n so four DAS addresses are generated. Any dataset can easily be handled using this scheme. This configuration has the added advantage of neatly separating the software program from the addresses that select DAS channels, makinq dataset changes easy and having a minimal effect on the software program.

AATC handles sync words by using a special mapping for addresses. Generation of a sync word requires two pieces of information; the actual sync word data, and the DAS that is to receive this data. The DAS address is stored in the DAS PROM, the sync word data is stored in the program PROM. Addresses are mapped so that one 1802 address simultaneously selects the program PROM location containing the sync word data and the DAS PROM location containing the address for DAS selection. Normally, the program PROM is selected by the MRD line during a fetch or execute cycle of the microprocessor. To select the program PROM during DMA, the 1802 state code lines are decoded to indicate a DMA cycle. The output of this decode logic is OR'd with MRD to the chip select of the program PROM. To output a set of sync words, RØ is initialized to the 1802 address that selects the first sync word data and DAS address. A set of sync words are assigned contigious address locations in memory. PRLD will assert DMA, providing the 1802 address to select the first sync word. Subsequent sync words are addressed by using the DMA repeat bit of the DAS PROM.

Figure 5 shows the contents of a hypothetical PROM, programmed for AATC for a four-DAS AATC system. To simplify Figure 5, rather than showing the actual address to each DAS and channel, a description is given (e.g., "address of Channel 1, DAS 1 and 2" rather than "1000 0000 1100 0001"). The column of bits to the right of the DAS PROM in Figure 5 represents the locations in PROM reserved to indicate if DMA is to be repeated immediately after the DMA cycle. Initializing the 1802 register RØ to line 1, the microprocessor sequences through the PROM as follows:

1. PRLD asserts DMA, addressing line 1 of the DAS PROM and line 1 of the program PROM. The output of the DAS PROM selects DAS 1 to receive sync data output from the program PROM. The DMA repeat bit is high, line 2 will be addressed immediately after line 1.

2. DMA repeat bit asserts DMA, addressing line 2 of the DAS PROM and line 2 of the program PROM. This provides sync data to DAS 2. The DMA repeat bit is high, line 3 will be addressed immediately after line 2.

3. DMA repeat bit asserts DMA, addressing line 3 providing sync data to DAS 3. The DMA repeat bit is high, line 4 will be addressed immediately after line 3.

4. DMA repeat bit asserts DMA, addressing line 4 providing sync data to DAS 4. DMA repeat bit is low, the next DMA cycle will be asserted by PRLD.

5. PRLD asserts DMA, addressing line 5 of the DAS PROM. The output of the DAS PROM addresses all four DASs to select Channel 5.

6. The next PRLD asserts DMA, addressing line 6 of the DAS PROM. Channel 12 of all DASs is selected.

7. The next PRLD asserts DMA, addressing line 7 of the DAS PROM. Channel 1 of DASs 1 and 2 is selected. The DMA repeat bit is high, line 8 of the PROM is immediately addressed after line 7.

8. The DMA repeat bit asserts DMA, addressing line 8 and selecting Channel 23 of DASs 3 and 4.

9. The next PRLD asserts DMA, addressing line 9 of the DAS PROM. Selecting the corresponding channel and DASs.

10. The pattern is repeated until line 9 is selected. Software is then used to reinitialize RØ to line 1.

A flow chart of the software used by AATC to handle a dataset is shown in Figure 6. This software requires the six software instructions (also shown in Figure 6). PRLD and the DMA repeat bit automatically send out sync words and address DASs. The telemetry system uses a 128-word dataset and runs the 1802 at 2 MHz. Once RØ is initialized, the microprocessor is free to execute software for 4 milliseconds. this is enough time to execute 500 software instructions. RØ must be reinitialized every 4 milliseconds. the period of one frame. The last instructions executed in the dataset are a software loop that determines when RØ addresses the last DAS PROM location for that dataset. When the last DAS PROM location is addressed, the program counter branches to the beginning of the dataset software, reinitializing RØ. Detecting the end of the dataset, branching to the beginning of the dataset software, and initializing RØ takes only six instructions (48 μs) , so there is plenty of time to detect the end of the frame and reinitialize RØ without interferring with the AATC process.

Figure 7 shows how software executed during the dataset may be used to detect when a dataset change is to occur. Changing to a different dataset simply requires software to poll a flag or data that will indicate when to enter the next dataset. To maintain sync, the first step in switching the datasets is to finish transmission of the present frame. This is accomplished by waiting until RØ selects the last address for dataset 1. Dataset 2 is then transmitted by branching to the corresponding software.

COMPARISON OF AATC AND SATC TECHNIQUES

This section compares AATC with SATC on the following features: 1) software capability, 2) software complexity, 3) dataset formatting, encoding, and changes, and 4) hardware count.

The extra software capability provided by AATC over SATC can best be illustrated by comparing the microprocessor execution time required for addressing channels in each case. This comparison used the Tektronix 8002 to do a real time trace of the microprocessor as it executed instructions. The flow charts for the two datasets compared, dataset 3 and dataset 1, are shown in Figures 8 and 9. Dataset 1 used 104 instructions to address 16 datasets; one can see that the software program was very busy addressing channels. Dataset 3 employed eight instructions to initialize and start AATC. Had there been a need, over 200 instructions handling other jobs could have been executed during the time the 16 datasets were addressed in Dataset 3.

Software complexity is high in SATC; software to do other jobs must be intermingled with software to address channels. As a result, any job requiring more than five instructions is broken up into small noncontiguous sections. The situation is further complicated because software for SATC is optimized by incorporating software loops to address channels. These software loops are broken up when other software jobs are introduced. All this makes such software difficult to write, test, and debug. By using AATC, once initialization is completed, software for other jobs can be executed without interference by the automatic channel addressing.

In using SATC, unless an effort is made to assign data channels to contiguous addresses ordered as they appear in a dataset, the software program becomes very lengthy and cumbersome. Imposing contiguous addresses to data channels as they appear in a dataset requires that data channels be carefully assigned to DASs and DAS channels. This is very time consuming. A modification to a dataset may require rewriting a considerable amount of software.

AATC simplifies the coding and formatting of a dataset by eliminating restrictions posed by SATC. AATC has the actual channel addresses stored in the PROM. As a result, addresses need not occur in a contiguous order. A dataset may be encoded by first assigning data channels to DASs, then programming the DAS PROM with addresses corresponding to each channel in the order they are to appear in the datset. AATC separates the software program from addresses to the DAS, with each stored in different

PROMs. Changing a channel in a dataset only involves changing a location in the DAS PROM, and has no impact on the software program.

AATC may reduce the amount of hardware necessary to address channels, as compared to SATC. Applying AATC to address the four datasets of the telemetry system required 198 addresses (396 bytes). To address the same four datasets, SATC required 1000 bytes of memory. In this case, AATC offers the potential of using one PROM, where SATC requires two PROMs.

In conclusion, AATC increases software capability, decreases software complexity, eases dataset format and encoding, and may provide a savings in hardware.

REFERENCES

1. "User Manual for the CDP1802 COSMAC Microprocessor," RCA Corporation, Somerville, NJ, 1976, P. 86.

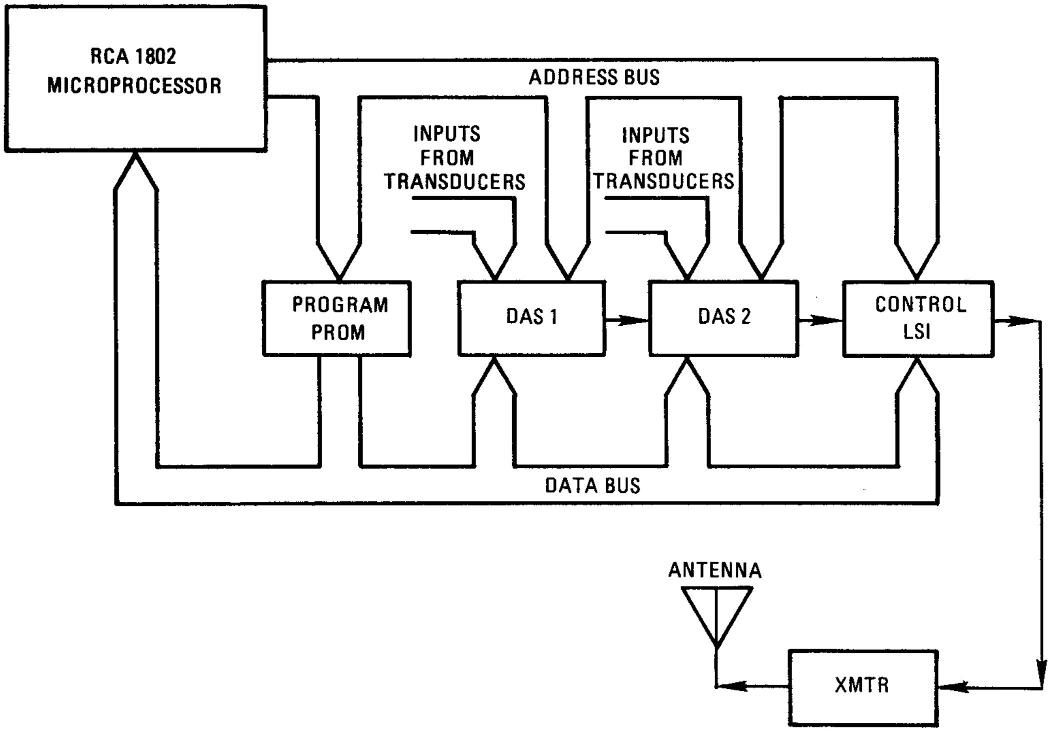


Figure 1. Telemetry System Block Diagram

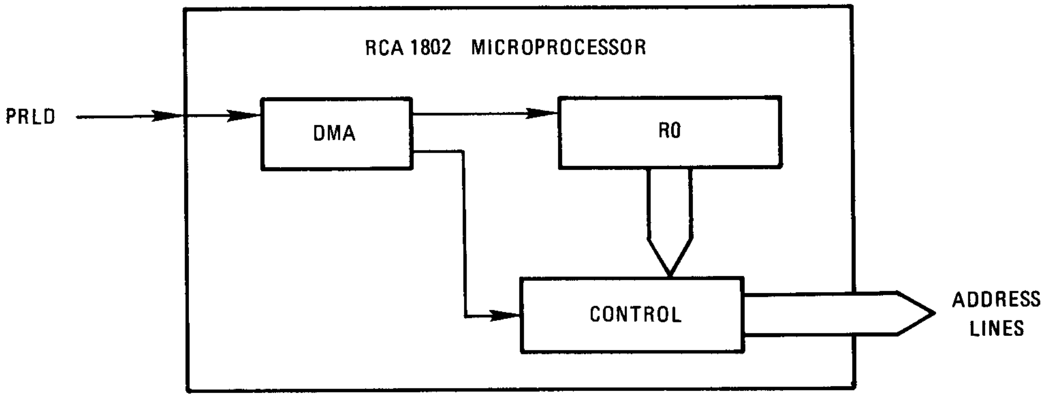


Figure 2. PRLD Used to Drive DMA

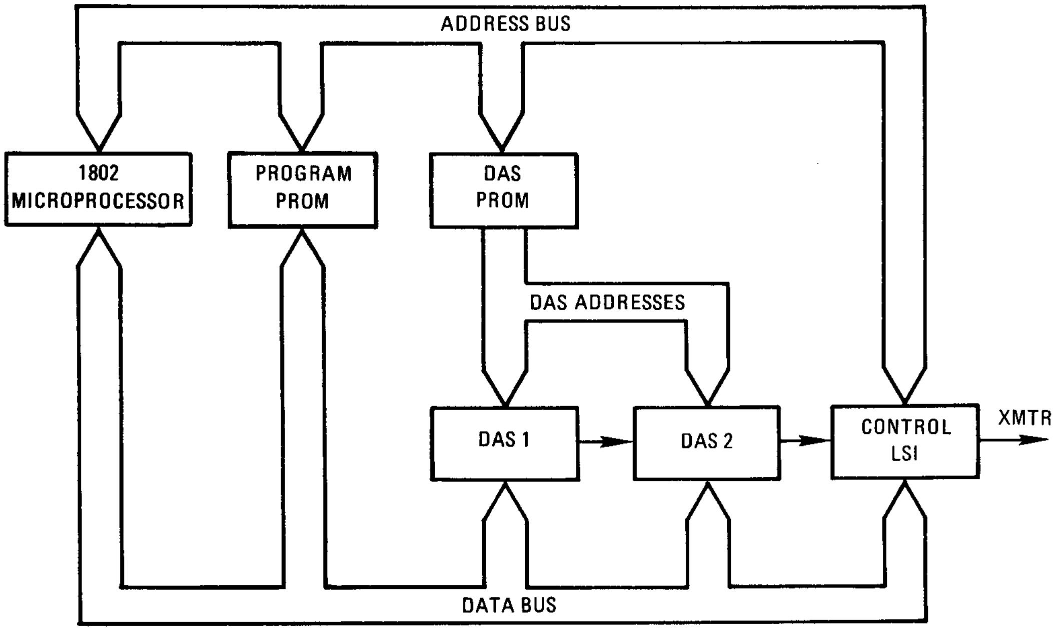


Figure 3. Block Diagram of AATC Using PROM to Address DAS

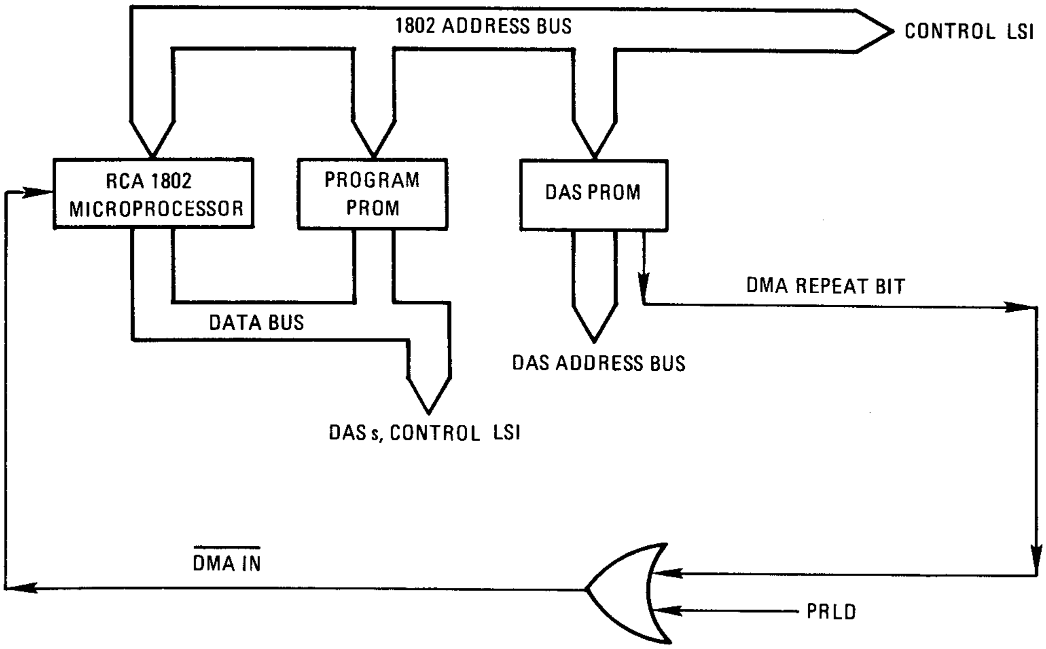
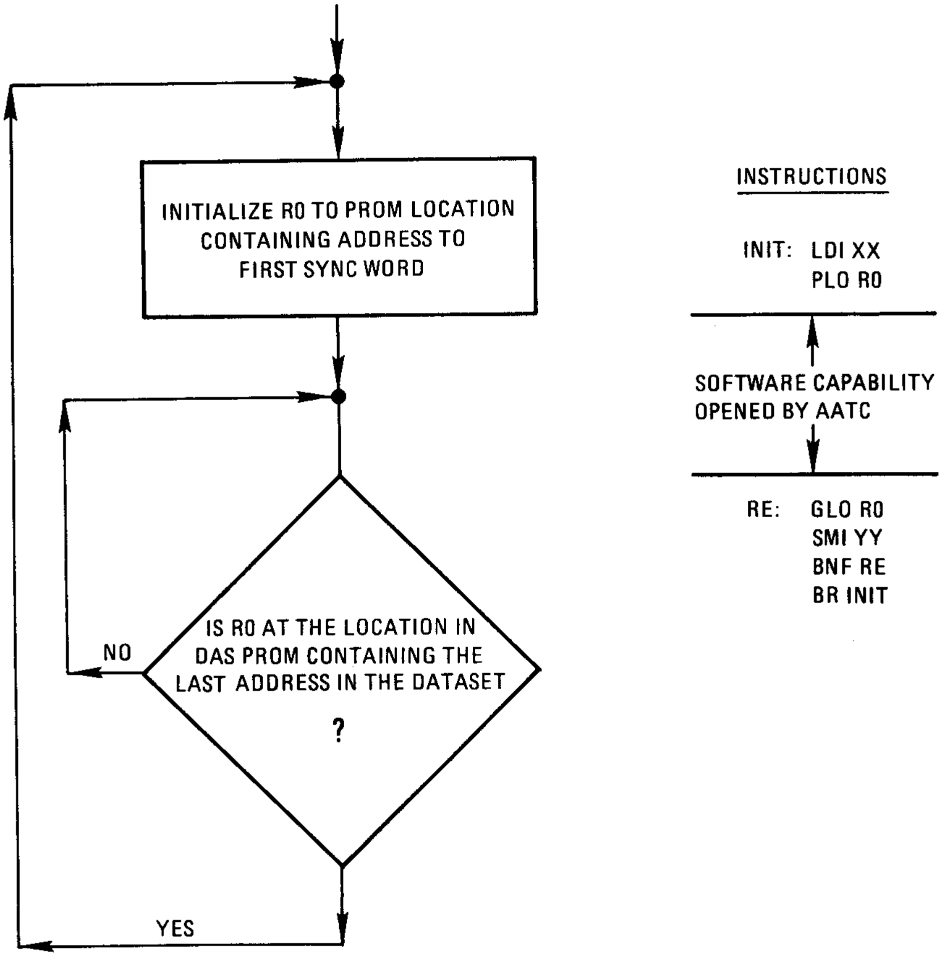


Figure 4. Block Diagram of DMA Repeat Bit Capability

	DAS PROM		
Line	Address for:	DMA Repeat Bit	Program PROM
1	DAS 1, Sync	1	Sync Word 1
2	DAS 2, Sync	1	Sync Word 2
3	DAS 3, Sync	1	Sync Word 3
4	DAS 4, Sync	0	Sync Word 4
5	DASs 1,2,3,4 Channel 5	0	--
6	DASs 1,2,3,4 Channel 12	0	--
7	DASs 1,2 Channel 1	1	--
8	DASs 3,4 Channel 23	0	--
9	DASs 1,2,3,4 Channel 8	0	--

Figure 5. Contents of PROMs Implementing AATC


INITIALIZE R0 TO PROM LOCATION CONTAINING ADDRESS TO FIRST SYNC WORD
IS R0 AT THE LOCATION IN DAS PROM CONTAINING THE LAST ADDRESS IN THE DATASET ?
NO
YES
SOFTWARE CAPABILITY OPENED BY AATC

```
INSTRUCTIONS

INIT:   LDI XX
        PLO R0

RE:     GLO R0
        SMI YY
        BNF RE
        BR INIT
```

Figure 6. Software Flowchart and Instructions for a Dataset Using AATC

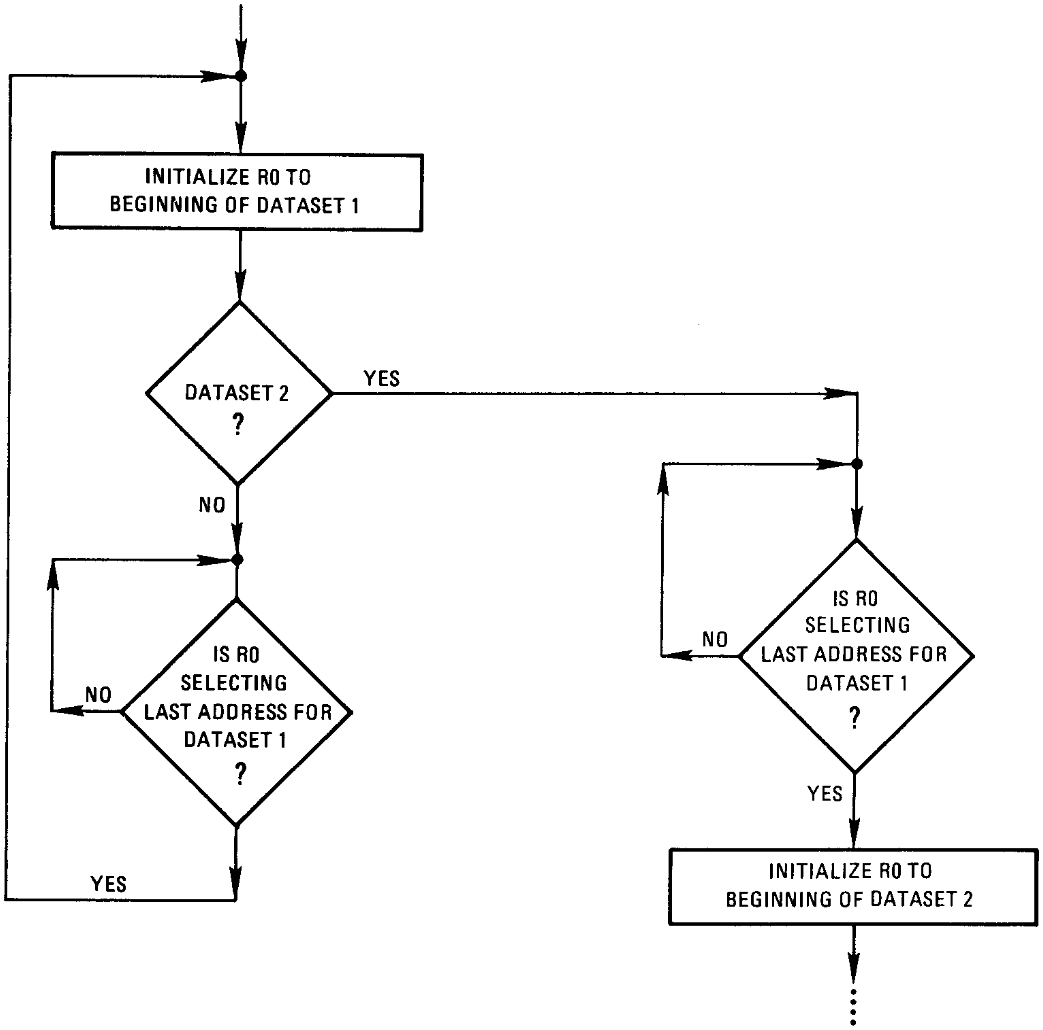

INITIALIZE R0 TO BEGINNING OF DATASET 1
DATASET 2 ?
YES
NO
IS R0 SELECTING LAST ADDRESS FOR DATASET 1 ?
NO
YES
IS R0 SELECTING LAST ADDRESS FOR DATASET 1 ?
NO
YES
INITIALIZE R0 TO BEGINNING OF DATASET 2

Figure 7. Software Flowchart Illustrating How to Change Datasets

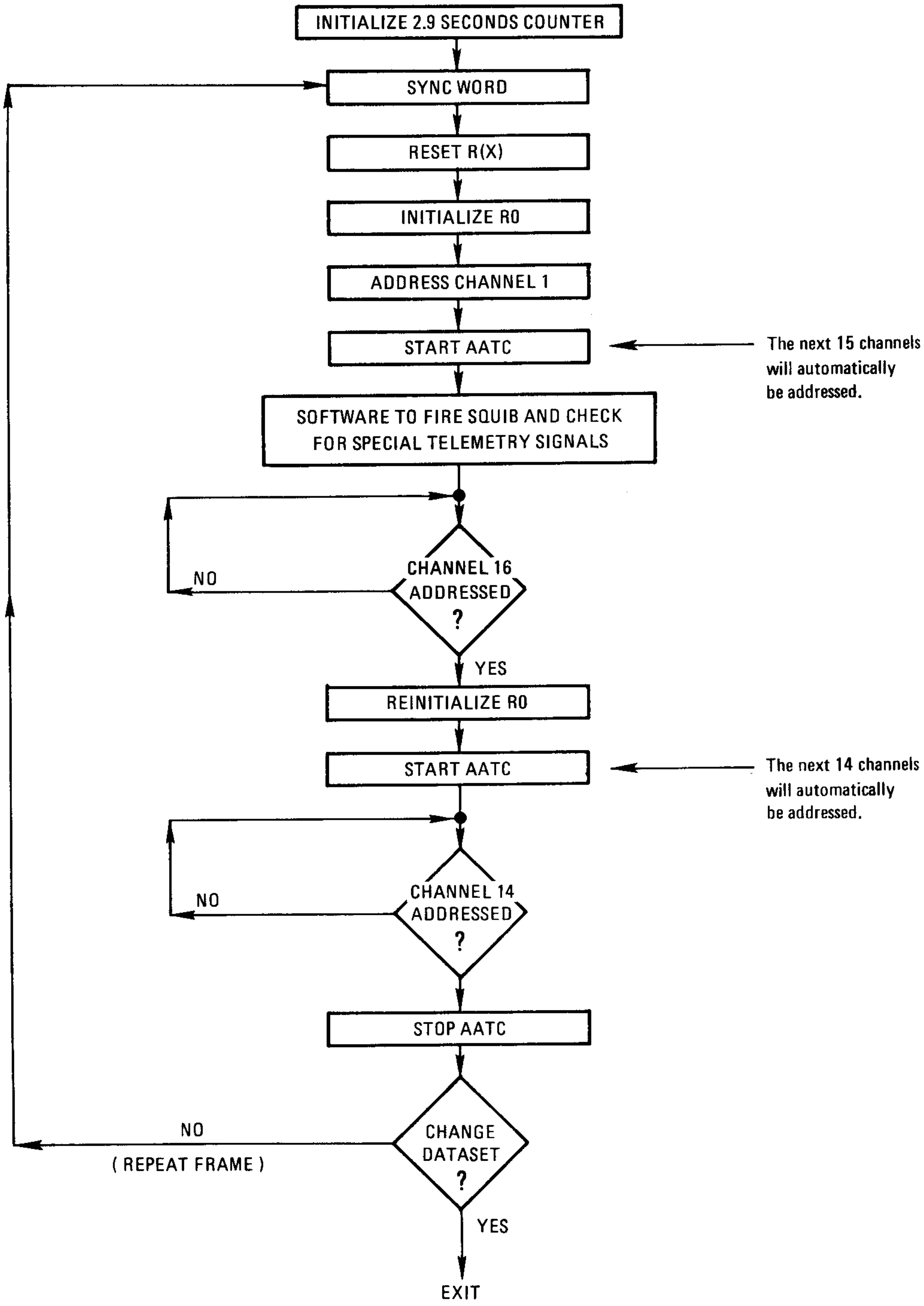

INITIALIZE 2.9 SECONDS COUNTER
SYNC WORD
RESET R(X)
INITIALIZE R0
ADDRESS CHANNEL 1
START AATC
The next 15 channels will automatically be addressed.
SOFTWARE TO FIRE SQUIB AND CHECK FOR SPECIAL TELEMETRY SIGNALS
CHANNEL 16 ADDRESSED ?
NO
YES
REINITIALIZE R0
START AATC
The next 14 channels will automatically be addressed.
CHANNEL 14 ADDRESSED ?
NO
STOP AATC
CHANGE DATASET ?
NO
(REPEAT FRAME)
YES
EXIT

Figure 8. Flowchart/Execution of Dataset 3

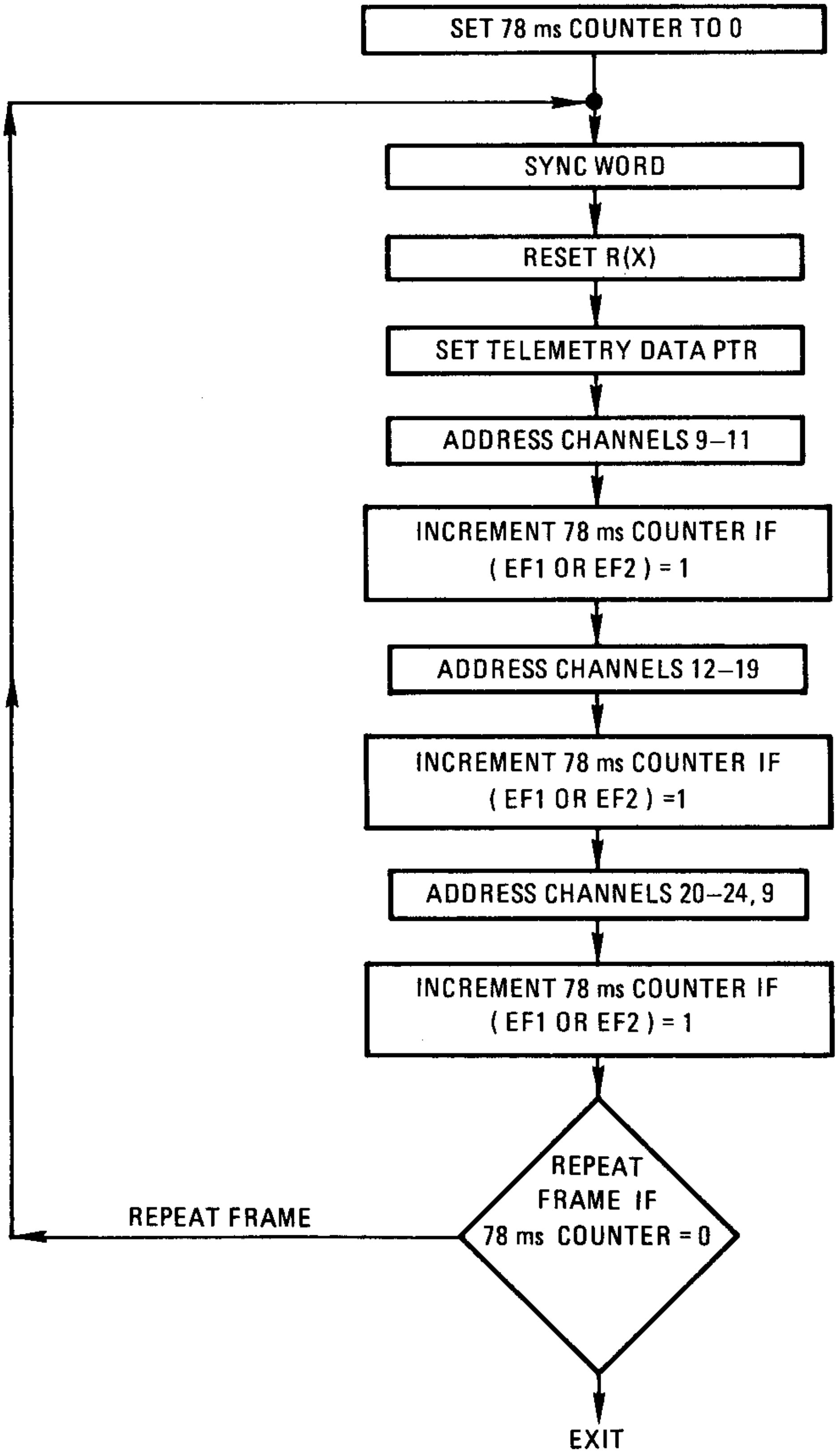

SET 78 ms COUNTER TO 0
SYNC WORD
RESET R(X)
SET TELEMETRY DATA PTR
ADDRESS CHANNELS 9–11
INCREMENT 78 ms COUNTER IF (EF1 OR EF2) = 1
ADDRESS CHANNELS 12–19
INCREMENT 78 ms COUNTER IF (EF1 OR EF2) =1
ADDRESS CHANNELS 20–24, 9
INCREMENT 78 ms COUNTER IF (EF1 OR EF2) = 1
REPEAT FRAME IF 78 ms COUNTER = 0
REPEAT FRAME
EXIT

Figure 9. Flowchart/Execution of Dataset 1