AAEC/E657

AAEC/E657

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

A VIDEO IMAGING SYSTEM AND RELATED CONTROL HARDWARE FOR NUCLEAR SAFEGUARDS SURVEILLANCE APPLICATIONS

by

J. V. WHICHELLO

MARCH 1987

ISBN 0 642 59865 7

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

A DATA TRANSMISSION SYSTEM FOR THE CENTRIFUGE ENRICHMENT PLANT SAFEGUARDS PROJECT

by

E.W. HESSE

ABSTRACT

A novel data transmission system was developed to enable the Australian Safeguards Office (ASO) in Sydney to monitor remotely the running of an experimental enrichment facility at the Lucas Heights Research Laboratories, using Telecom telephone lines. The system allowed ASO to monitor the enrichment, flows and activity in sensitive areas of the plant by means of a master computer at ASO and a slave computer at Lucas Heights. The slave computer sent requested data and scanned video pictures to determine intrusive changes to the plant. Detailed descriptions of the transmission system and method of operation are given, together with an outline of experience obtained during a three-month surveillance of the facility.

ISBN 0 642 59883 5

The following descriptors have been selected from the INIS Thesaurus to describe the subject content of this report for information retrieval purposes. For further details please refer to IAEA-INIS-12 (INIS: Manual for Indexing) and IAEA-INIS-13 (INIS: Thesaurus) published in Vienna by the International Atomic Energy Agency.

CENTRIFUGE ENRICHMENT PLANTS; COMPUTER CODES; DATA TRANSMISSION; DATA TRANSMISSION SYSTEMS; DEC COMPUTERS; IMAGE PROCESSING; INSPECTION; PATTERN RECOGNITION; SAFEGUARDS; VIDEO TAPES

EDITORIAL NOTE

The Australian Nuclear Science and Technology Organisation replaced the Australian Atomic Energy Commission on 27 April 1987. Reports issued after April 1987 have the prefix ANSTO with no change of the symbol (E, M, S or C) or numbering sequence. Several reports issued in April 1987 have retained the AAEC prefix, their publication having been delayed by unavoidable production problems.

CONTENTS

1. INTRODUCTION

A data transmission system was developed as part of the centrifuge plant surveillance project to allow the Australian Safeguards Office (ASO), at Kings Cross, Sydney, to obtain information on the running of an experimental enrichment facility operated by the Australian Atomic Energy Commission* at the Lucas Heights Research Laboratories. This information was to be in the form of

(a) flows and enrichments in the feed and product lines,

(b) power being supplied or not supplied to the cascade, and

(c) regular video pictures of five areas of the enrichment plant.

By Ministerial direction, operation of the uranium gas centrifuge plant at Lucas Heights Research Laboratories ceased on February 28, 1986. However, work continued on checking the reliability of the surveillance project.

The transmission system consisted of two computer-controlled systems coupled *via* a Telecom telephone connection. The master computer, located at ASO, organised the requests for and the storing of data, and the slave computer at Lucas Heights collected and sent back the requested data. The slave computer had a secondary function of monitoring some of the video pictures to detect image changes, and the ASO computer checked on the status of all devices in the system.

As outlined by Mercer [1987], emphasis was placed on making the system and computer programs as simple as possible to help overcome time and budgetary constraints. Consequently, improved methods of data transfer, inputting commands and other refinements were not done, the intention being that once the system had been fully demonstrated, a better examination could be made of alternative techniques to improve those sections of coding or hardware which limited the performance of this surveillance technique.

2. DESCRIPTION OF COMPUTER SYSTEM HARDWARE

Both computer systems were based on a Digital Equipment Corporation (DEC) LSI 11/23 computer with 265 K bytes of memory. Each system was mounted in a moveable cabinet as shown schematically in **figures 1** and **2**. Common to both systems were

- a dual floppy disk drive,
- a computer screen and keyboard,
- a Concord Data Systems 2400 bit/second V22 *bis* auto-dial telephone modem connected *via* a Randata Data encryptor and serial line interface,
- a Colorado Data Systems Video 274 frame store connected *via* a direct memory access (DMA) parallel line interface, and
- a video monitor connected to the frame store.

In addition to the above components, the ASO system had

- a video cassette recorder (VCR) connected *via* a control unit and parallel line interface, and
- a printer connected direct to the screen.

The AAEC slave system had several additional devices:

- a Doric Digitrend 220 data logger connected *via* two parallel line interfaces,
- a video camera switcher connected *via* a parallel line interface,

---

* Now the Australian Nuclear Science and Technology Organisation (ANSTO)

- a time caption generator,
- a serial line connection to the DEC LSI 11/03 computer, controlling a Balzers quadrupole mass spectrometer (QMS; see Evans [1987]), and
- a non-interruptable power supply.

Each computer could read from or write to any group of the 64 luminance values in the 256 by 256 pixel video picture array in the frame store. The frame store in the slave system could be controlled by the video camera switcher to capture a frame from any of the five cameras monitoring the cascade. The captured frame contained the time of day generated by the time caption generator. The ASO VCR was controlled by the computer to tape the frame store picture received from the slave.

The encryptors could be withheld by the ASO computer to either the encryption or the bypass mode. The Doric data logger allowed the slave computer to request the logger to measure any channel and to receive the reading as a BCD output. The program in the QMS 11/03 computer allowed the slave computer to request measurement of the enrichment in either the feed or product lines [Rutherford and Mercer 1987]. The video monitoring screens were required only for checking the alignment of the cameras and to see if the system was working. The disc drives were used only to load the programs.

3. OPERATION OF THE SYSTEM

The programs in the two computers were run independently and written in RT11 FORTRAN which has the facility to access device handlers under interrupt control. (For additional program details see **appendices B** and **C**.) Although the slave program was written to respond immediately to an ASO request, it was independently responsible for monitoring changes in video pictures and the performance of the data logger and QMS. It would continue these tasks even when the communication link with the ASO computer was broken. The computer programs cycled through a set of flags to check if any tasks needed to be done, regardless of whether a telephone connection had been made. The flags were set either by the RT11 interrupt control (either completion or time) or after a section of each task was completed. The programs were arranged to test the flags in order of importance after any task was completed, thus minimising the time spent in servicing telephone interrupts. Both programs had inbuilt checks to ensure that they continued to operate and were able to reboot and restart under error conditions.

The programs used a special RT11 MACRO Input/Output(I/O) interrupt completion routine to flag the arrival of a legitimate communication string on the telephone line. When set, the flag caused the program to start a high priority task to check and interpret the string. To enable communication, the ASO computer initiated command strings and the slave computer acknowledged those command strings that it understood. The slave never initiated a communication and the ASO computer could only deduce changes occurring in the slave by obtaining regular status reports. Except for video data, every slave reply carried with it the slave status conditions. All communicating strings included cyclical redundancy checksum (CRC) error checking.

The ASO program periodically checked if communication between the two machines was satisfactory; if not (no reply or too many line errors), it started a task to re-establish a telephone link by sending a series of commands to its modem to re-dial the slave computer. In addition, both programs had keyboard interrupt control to allow an operator to enter commands or settings on the ASO computer or to allow the testing of devices serviced by the slave computer.

The slave program was written to run a task that continually stored and scanned three of the five video camera pictures and so check if the latest image had changed significantly; if so, it would notify the ASO computer that it had detected the change (*via* a status flag). The task was written so that it did not interfere with the transmission of requested data back to ASO. To speed up notification, the slave computer would terminate transmission of any routine picture.

When a request for flow and power data was received, the slave initiated an interactive task to carry out the sequence of writing to, and reading from the Doric data logger to obtain measurements from four channels. Requests for enrichment values were passed on to the QMS computer which was then monitored by the slave until it received the result. The slave program also monitored the continual operation of the QMS computer and would try to reboot it should

regular communication cease.

The ASO program was written so that in addition to passing commands to, and receiving data from the slave, it would run a task which automatically sends commands to complete a predetermined sequence of events. The task obtained picture data from each camera in turn, interspersed with data of the cascade flows and, at set intervals, enrichment readings. This task, which could easily be extended, took a low priority, so that other tasks, such as the one to request a 'change detected' picture, would take precedence. The cycle time for this task was approximately seven minutes.

The ASO program could also start a task to record on the VCR the picture it had loaded from the modem in the frame store after receiving the 'end of picture' reply. In addition, the program checked the flags in the slave computer responses and initiated tasks to request either a 'change detected' picture or a QMS result.

4. COMMUNICATION

As stated above, ASO initiated all requests and the slave replied to known requests. This practice was adopted after experimentation had shown that it would be difficult to predict and handle all possible error sequences in the two machines when the telephone link became noisy. As the surveillance program required the ASO computer to be running and in complete charge, there was no necessity for the slave to initiate tasks. This concept had the advantage that only the ASO computer could open its modem input buffer when it expected a predictable reply, *i.e.* after it has sent a command. This avoided the need to process of random spurious noise on the line and, in addition, allowed ASO to ignore the unexpected. On the other hand the slave, whose line had to remain open, could simply ignore anything that it did not understand, knowing that ASO would retransmit the command after it had failed to receive a reply within a set time. This method produced a very stable system.

The following procedures were adopted:

(i) The ASO computer sent a short command containing two characters to indicate which particular type of data the slave should prepare to send.

(ii) The slave would send either an acknowledgement (*i.e.* ACK plus data character) if it could initiate the data collection or, if this was not possible, a negative acknowledgement (*i.e.* NAK plus data character together with a character indicating the reason for the NAK).

(iii) ASO would then send a command with a request character for the data to be sent.

(iv) The slave would send those data, if they were ready, or send an NAK together with a character indicating the reason for non-availability.

As the data for the video picture were sent in 250 lots of 256-byte blocks, ASO had to make a request for each block number, thereby repeating steps (iii) and (iv) until the slave had no more blocks to send; the slave then sent an end of data response (*i.e.* SYN plus data character).

Every ASO command was eight bytes in length and the lengths of all the slave replies were 18 bytes, except for the video data block which was 266. The first byte of the string was set to '1' on output and the last two bytes contained the checksum. The RT11 input handler checked for the '1' start and, if it did not receive the required number of characters in a continuous stream, replaced the first character with '0'. Both the ASO and slave programs ignored strings with '0' (except for increasing the phone-link error count). If the slave determined a checksum error, it replied with an NAK plus 'ET' to indicate a transmission error. If ASO received a checksum error, it was ignored and, if no reply was received in three seconds, it would automatically re-transmit the previous string.

This procedure prevented the communication from getting out of synchronisation because of either computer answering erroneous data. Additional checking was achieved by confirming that the slave's response contained the expected data request character or block number. Also, each new ASO command string contained an incremented number (byte 5) to distinguish it from a repeated timeout command. The slave would only re-transmit its previous response and would not re-activate a data task when it received a command with a repeated number. This prevented the slave from losing track of interrupts, by re-starting a device in mid-stream.

To save time, the checksumming was done only on the first 16 bytes of the picture data. This allowed the occasional pixel to be wrong without really disturbing the picture. However, if too many link errors occurred, the program started a task to re-establish a new telephone link with the hope of getting a better line. This task was also started if there were 10 repeated timeouts (approximately half-minute outages).

The set of ASO commands incorporated during this work (starting byte 2) were as follows:

CD To request a logger measurement.
CQx To initiate the QMS,
where x = F for feed enrichment,
= P for product enrichment,
= B for rebooting the QMS computer.

CPn To obtain video picture from camera number, n,
where n = 1 feed station camera,
= 2 product station camera,
= 3 waste station camera,
= 4 cascade room (wall),
= 5 cascade room (roof),
= 0 greyscale generator.

CLn Similar to CPn except that the picture cannot be terminated for a 'change detected' picture.
CT To transmit logger data.
CU To transmit QMS data.
Bno To transmit picture data for block number no (two-byte integer).
CS To request acknowledge (*i.e.* send status flags).

Additional commands were incorporated during the developmental stage to allow ASO to request a computer program file (load module) from the slave's disc drive. This used an extension of the facility for transmitting a video picture:

CF To open the disc drive file called NEWASO.SAV.
Ano To transmit file data for half block number number (two bytes).

The description and format of the slave's replies are given in **appendix A**.

5. KEYBOARD INPUT

As well as entering the command strings which were to be sent to the slave (*e.g.* CP3 carriage return), the ASO program accepted the following setting or checkout commands:

R To start the automatic command sequence.
H To halt the automatic command sequence.
M To start the automatic regular status commands.
S To make a new telephone connection.
E To close down the telephone connection.
W To force the VCR to tape the picture in the frame store.
T To set the automatic QMS requests of frequency and ratio of product to feed readings.
? To obtain display of ASO status flags and settings.

The slave computer also accepted keyboard input to enable the checkout of the system. The program accepted the following commands:

D	To force the Doric logger to make a measurement sequence.
QF or QP	To initiate a QMS measurement.
QG	To halt the QMS.
QB	To reboot the QMS computer.
Pn	To switch and display camera number n.
Fn	To display stored 'change detected' picture number n.
X	To allow the re setting of luminance error and number of allowable pixel changes.
S	To display slave status flags and results from the last measurement.

6. PRESENTATION OF OUTPUT

All ASO commands and data received by the slave computer were displayed on the ASO screen together with an appended time of day. This information was normally typed on a printer which was connected directly to the screen. The date was also printed every 60 lines. As all slave data replies included the slave status flags, a record was available of all information obtained from the slave computer, together with operational performance of the equipment. If any information or slave equipment were unavailable, a warning message was printed. The number of pictures recorded on the video recorder, together with a message for any 'changed detected' picture, was also recorded. Telephone connection progress messages were printed as well. Thus, from the printed time, it is easy to establish when changes occurred either in plant operation or equipment performance.

7. VIDEO PICTURE MONITORING

When the slave computer received a command from ASO, it first set bits 0 to 2 on the parallel line controlling the camera switcher to connect the required camera to the frame store. It also kept bit 3 on for half a second to allow the picture and the time caption to be properly locked into the frame store. The contents of the frame store were then copied into a 'transmission' store in virtual memory, from where it was subsequently extracted, a column at a time, and sent to ASO. To allow interaction of other tasks, the copying was done in 10-column packets. For programming convenience, only 250 columns (starting at column 3) were stored and transferred to ASO.

The time taken to transmit a complete picture to ASO was just over five minutes. To achieve maximum speed, the slave program prepared the next column in advance, in a second buffer, the first remaining in case ASO did not receive it correctly and asked for it again. To save time, ASO loaded the received strip only after requesting the next block. When the end of picture reply was received by ASO, a task was started which operated the record mode on the VCR for seven seconds. If the VCR controller did not acknowledge either the SWITCH ON or OFF command, an error message was generated.

Once the slave had copied the requested picture to the 'transmission' store, the frame store was free to be switched to one of the three motion detection cameras to check for a changed picture. For this, only part of the picture was extracted and copied into a new store (also virtual). Because of storage limits, and because only some part of the picture included a critical area, the program only selected a 200 x 148 or 160 x 182 pixel area (starting at the top left 30 by 30 pixel); this area excluded the variable time caption. This procedure requires three old plus one new store of 30 000 bytes each, above the 64 000 byte storage required for the 'transmission' store.

While being transferred, each pixel luminance was compared with that previously stored to establish how many had changed by more than a set limit. If more than this limit had changed in any eighth of the picture, a 'change detected' signal was triggered. However, to eliminate temporary changes such as those caused by passing shadows, the program first tried to confirm the change by reloading the frame store with a new image and comparing it with the stored picture. Only then, if it still detected a change, would it take action, holding the picture in the frame store until the ASO computer could request the 'change detected' picture which was then transferred to the transmission store. If the picture being transferred was not already a 'change detected' picture, the slave could terminate it by changing the block number to the end number to speed up the notification. Once the 'change detected' picture was in the 'transmission' store, the program was free to continue scanning the three cameras. Subsequent changes on the same camera were ignored until that picture has been transferred to ASO. A detected change on a different camera was held in the frame store until it was requested by ASO, on completion of the first picture. In this case, the program could not continue to monitor changes because of the lack

of core storage. This limitation is not serious as the objective is to monitor long-term changes which would take more than six minutes to implement. (However, see **section 10.3** for limits to the detection and system operation.)

Extraction and comparison were done 10 columns at a time to reduce the time that the slave program was unavailable to service ASO requests. Although this detection task was given the lowest priority, it was still possible to cycle the three cameras in less than 35 seconds without increasing the transmission time to ASO.

The slave program included a task which allowed the 'change detected' picture to be written to the frame store and then viewed on the video monitor. This enabled the video cameras to be positioned to cover the critical area in the detection area.

8. DORIC DATA LOGGER MONITORING

When the slave received a Doric measurement request, it started an interrupt driven task to carry out the following steps:

(a) Send a command *via* the parallel line controlling the logger to halt and put the logger into random request mode.

(b) Send the logger the required channel in BCD form when it received the logger acknowledgement.

(c) Read, convert and store the BCD value in floating point form when the logger acknowledged the measurement completion.

(d) Repeat steps (a) to (c) for the required three flows and one power level.

(e) Report completion of the task by setting the Doric flag appropriately. This allowed the slave to send the data to ASO on receipt of the next ASO transfer request.

If the logger failed to acknowledge a request in a given time, the Doric flag would be set to indicate that the device is out of order.

9. QMS MONITORING

The program for running the QMS [Rutherford and Mercer 1987] had been written to send a regular 20-byte string to the slave, where the first byte contained a character indication of the QMS state. The slave program received this string through an RT11 interrupt completion routine and checked that it was continually receiving it. When the slave received a QMS command from ASO, it carried out the following steps:

(a) Write the P or F to the QMS serial line.

(b) Check that the first character of the QMS string changes to the required P or F.

(c) Wait for the first character to change back to the non-measurement state, S, then store the average enrichment and statistical error values. This took approximately five minutes.

(d) Change the slave's QMS flag to its completion value to signal ASO that the measurement is finished.

If the regular response string was not received within one minute, the slave program sent a break signal followed by a re-boot command (173000G). If it still could not obtain a response, it would indicate an out-of-order condition in the slave's QMS flag. The program has also been extended for local use to display the time and the five enrichment values which produce the average value.

10. INITIAL COMMUNICATION SEQUENCE

To establish a telephone link to the slave computer, the ASO program started the following tasks:

(a) Send four 'cntrl C' bytes to ensure that the encryptors are in the bypass mode, followed by a break signal for two seconds to get the modems to disengage the telephone line.

(b) Send a command to get the modem in the command phase ('carriage return', 'carriage return', 'B', 'carriage return').

(c) Send the telephone number and type of dialing required when the response from the modem is received.

(d) Check the progress of the modem and, once a connection has been made, send a command to switch the encryptors from bypass to encrypt mode and also to make the first contact with the slave ('"01', '$', '$', '$', and four 'cntrl E' bytes).

(e) Check for the 'encryption on' message, followed by acknowledgement by the slave and, if correct, a report that communication has started.

Should any of these steps fail, the program would re-start at step 1, and try six times before reporting a failure. In the case of failure, the task would be automatically restarted in 15 minutes.

Because the encryptors did not always work reliably, the program was arranged to allow communications to proceed un-encrypted, only if both encryptors were manually switched to the bypass mode. The program would determine this condition because it would not receive an encryptor response if both encryptors were in manual bypass mode but would receive the correct slave response. If the encryptor at the slave end was not in the manual bypass mode, it would receive the encryptor switching acknowledgement normally trapped by the ASO encryptor and treat it as an error.

11. PROBLEMS ENCOUNTERED AND SUGGESTIONS FOR IMPROVEMENTS

This feasibility study was set up to demonstrate not only that it could work but also to obtain experience on its reliability and on the problems encountered under real working conditions. Although the system worked extremely well, five aspects of the system were found which could be improved.

11.1 Computer Outage

Although the system included two checks to ensure that the program continued to run, heavy thunderstorms of the type experienced at Lucas Heights were sufficient to stop the slave computer. Although more time could be spent on improving the system with better power supplies, this would not guarantee the system against all unforeseen faults.

With hindsight it is obvious that a better proposition would be for ASO to download the slave program, *via* the telephone, when it detected that the slave was not responding. This would have the advantage that neither the potentially unreliable slave disc drive nor the uninterruptable power supply would be required. An additional advantage would be that changes could be made to the slave program without entering the plant. The feasibility of sending the program down the telephone line was demonstrated during development when different versions of the ASO program were up-loaded to ASO. The QMS computer program could similarly be loaded from ASO.

11.2 Motion Detection Errors

Much work was done to reduce the number of false 'change detected' pictures initially thrown up by the system. This occurred because each pixel luminance changed to some degree between scans, depending on the accuracy of the alignment and the light intensity conversion in the frame store. Although the problem was partly due to the camera electronics and can be overcome [Whichello 1987], a large part of the problem was the poor location of the cameras and the inadequate lighting. In a better planned system, it should be possible to position the cameras well away from passing operators and sources of vibration. Also, the cameras should have narrow angle lenses and special narrow angle lighting to prevent shadows from affecting the picture. Coolant filling points should be placed away from the monitored valves and pipework to prevent normal actions by operators from interfering with the picture. If these changes were implemented, it is probable that the default pixel luminance error of 32 could be significantly reduced.

11.3 'Change Detected' Picture Notification Problem

During surveillance of the AAEC cascade, the telephone line was kept open for most of the time to obtain the maximum data from the enrichment facility. In this type of operation, the detection of a 'changed' picture ensured that ASO was immediately notified. However, if the line was momentarily closed down, any 'changed' picture would stop the surveillance until the next ASO request was initiated to free the frame store. If the system were used as part of an international surveillance system, the cost of ISD telephone calls would dictate that the line be closed for most of the time.

To overcome this problem there are a number of options. One simple solution would be to increase the memory (and upgrade the LSI 11/23 computer to a PDP 11/73) so that the slave program could store at least one 'changed' picture per camera. In this way, the surveillance agency could obtain a picture and time of the first disturbance, and compare the existing pipe and valve arrangement with those shown on its previous pictures to establish if there had been any change since the last transmission. The largest available memory was four megabytes which allowed the storage of an additional 60 'changed' pictures, thus making it feasible to store one 'changed' picture per camera every 30 minutes during a 24-hour non-transmission break.

An alternative solution would be for the slave computer to dial the ASO computer when it detected a change. Although this would lead to some changes in communication philosophy (see **section 4**), the larger problem would be to devise strategies to overcome the reluctance of the surveillance agency to give too much independent control to the slave computer. For example, it would be necessary to know how to prevent continual motion or an error in detection logic from tying up the telephone line, making it difficult for the agency to regain control.

As the design of this system is flexible, the adopted method would really depend on the needs of the surveillance agency in terms of data and reliability, and also on the funds available to pay for hardware and running costs.

11.4 Encryptor Problems

The encryption of communications did not work satisfactorily. It was found that the encryptors would occasionally get out of sequence and one would drop out of encryption mode. It was then be impossible, without switching off the power to both encryptors, to get them both in an un-encryption mode. The encryptors only worked for a period of two days in the surveillance system before the problem became too severe. Obviously, the specially developed software supplied with the encryptor should be improved to remove this problem. As the fault also occurred in bench tests, no blame can be attached to the telephone link.

11.5 Transmission Speed

Although the speed of transmission was adequate for the ASO-AAEC surveillance system, there were a few times during enrichment transient monitoring when a shorter picture transmission time would have allowed more QMS readings to be requested. Also, for ISD calls, a shorter transmission time would have been better. Since the time of purchase, 9600 baud/second modems have become available which would reduce the picture transmission time from approximately 5 to 1.5 minutes. The time of computer data preparation, VCR recording and Doric data logging would still be the same for the overall cycle time. A smaller but significant decrease in time could be obtained by using synchronous instead of asynchronous transmission.

12. CONCLUSIONS

Despite the fact that time and budgetary constraints prevented many refinements, the system worked beyond ASO's best expectations. The system performed reliably and produced a lot of useful surveillance data over a period of three months of intense activity before termination of the enrichment program. The few times that the system failed were due mainly to the electrical storms which disrupted the computer or Telecom line. The procedure for determining possible intrusion into sensitive plant proved to be very effective.

13. ACKNOWLEDGEMENTS

Acknowledgement is given to D.J. Mercer who wrote the special RT11/MACRO device handlers.

14. REFERENCES

Evans, P.J. [1987] - A quadrupole mass spectrometer system for nuclear safeguards application. ANSTO/E660

Mercer, D.J. [1987] - Centrifuge enrichment plant surveillance project. ANSTO/E report, in preparation.

Rutherford, C.J., Mercer, D.J. [1987] - Computer programs for controlling a Balzer's quadrupole mass spectrometer. AAEC internal report.

Whichello, J.V. [1987] - A video imaging system and related control hardware for nuclear safeguards surveillance applications. AAEC/E657.

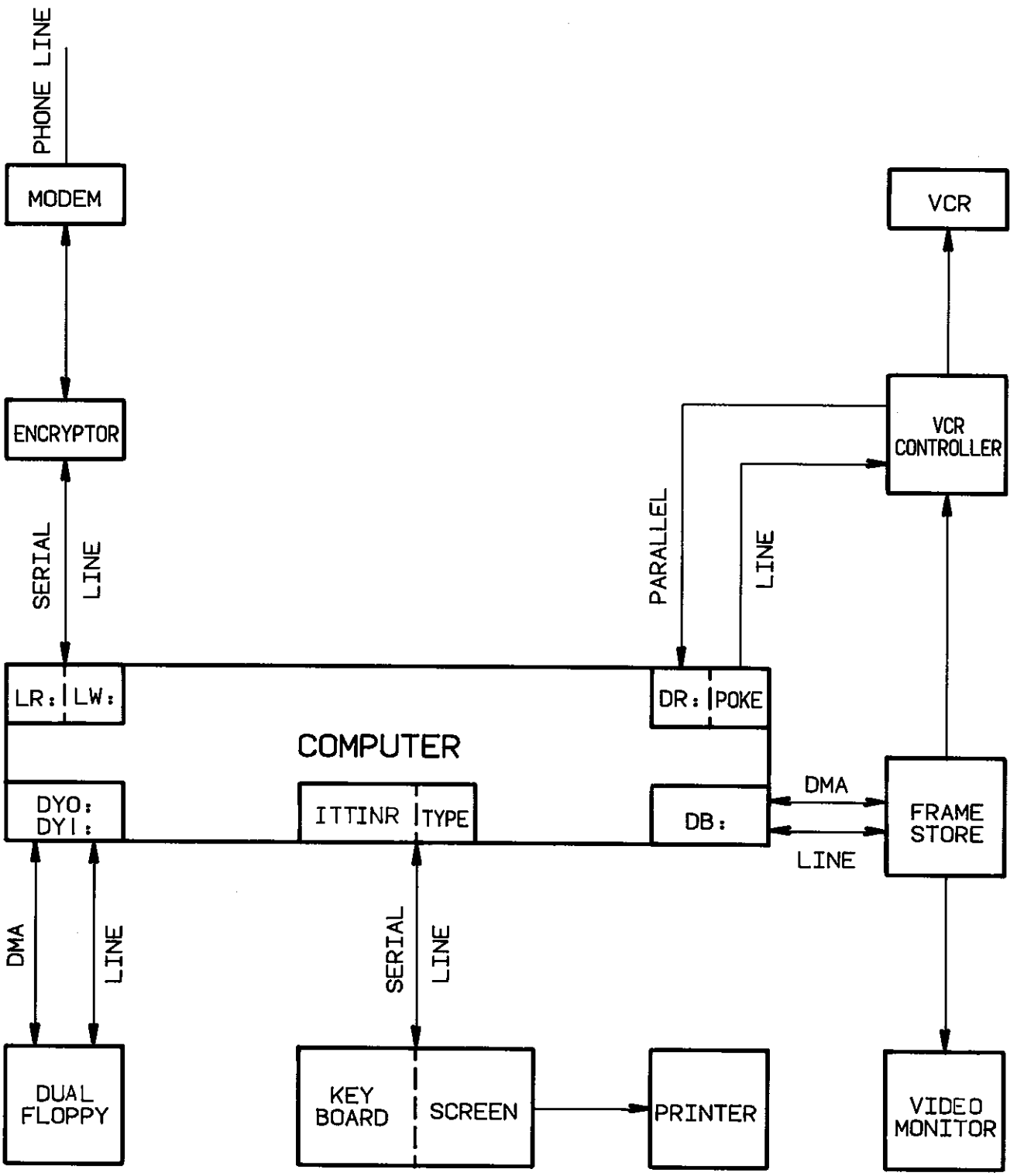

Figure 1 Schematic layout of the ASO computer system

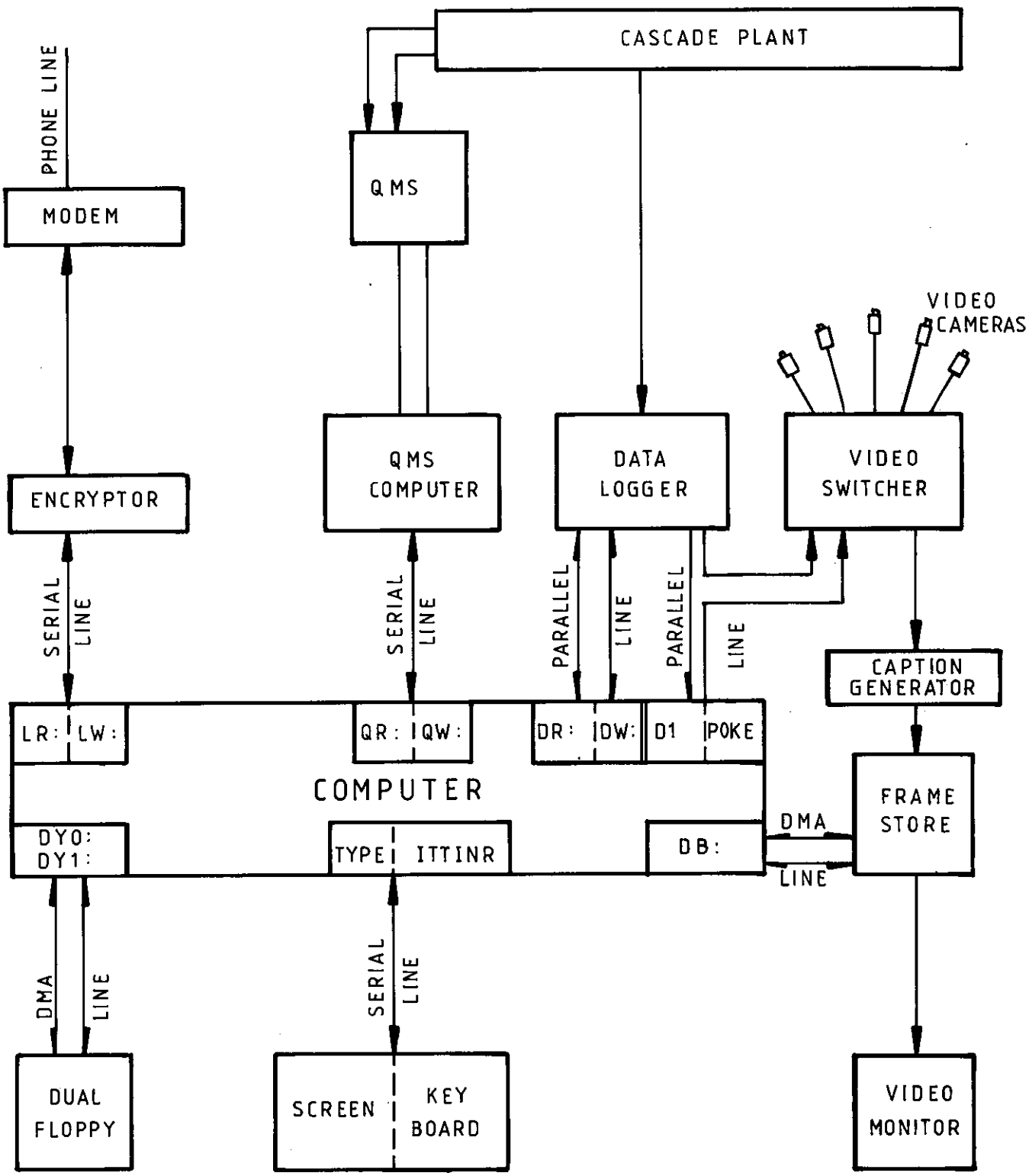

Figure 2 Schematic layout of the slave computer system

APPENDIX A
SLAVE REPLY FORMATS

The formats of the slave replies are described below:

(a) Acknowledgement replies have ACK ("06) followed by the two command characters, *e.g.* QF, P2 or D.

(b) Negative acknowledgement replies have NAK ("025) followed by a command character, and a character indicating the reason:

B	Measuring device is not working.
W	Measurement is not currently possible, *e.g.* there is no gas flow for the QMS or the device is recovering from some outage.
?	Command incorrect or not understood.

The above replies also contain the slave device status flags in bytes 5 to 12.

(c) Data logger reply

Byte		
2	T	
3	D	
4	Y	if cascade power on or N if not.
5 to 8		Feed flow rate (floating point).
9 to 12		Product flow rate (floating point).
13 to 16		Waste flow rate (floating point).

(d) QMS data reply

Byte		
2	U	
3	Q	
4	F	(feed) or P (product).
5 to 6		QMS status word (octal).
7 to 10		Average enrichment value (floating point).
11 to 14		Statistical error (floating point).

The above replies have the checksum number in bytes 17 and 18.

(e) Picture data reply

Byte		
2	B	
3 to 4		Block number (integer).
5 to 6		Frame store vertical strip number (integer).
7 to 8		Not used.
9 to 264		Luminance level for 256 pixels.
265 to 266		Checksum number calculated for bytes 1 to 16 only.

(f) Floppy disc data reply

Byte		
2	A	
2 to 4		Block number.
5 to 260		Program byte data.
261 to 264		Not used.
265 to 266		Checksum number calculated for bytes 1 to 260.

APPENDIX B
ASO PROGRAM

B1 SUBROUTINES

MAIN
- SETASO
- DIALUP
  - UNDIAL
- RECEPT
  - CHKREP
    - CRC
  - ANALYS
  - EXTRAC
  - QCONV
  - FRAMST
  - DISKST
  - DISKC
- SENSLA
- VCRSTR
- DISKO
- LOGM

B2 COMPLETION ROUTINES

TIMRTA LRRTA VRCRTN

B3 RT11 ROUTINES

IREADF	IWRITE	ITIMER	ICMKT
CONCAT	IPOKE	IPEEK	ISLEEP
IRAD50	LOOKUP	IFETCH	IENTER

B4 DEVICE HANDLERS

LR	LW	(Telephone)
DR	DW	(VCR controller)
DB		(Framestore)

B5 OUTLINE OF MAJOR ROUTINES

B5.1 MAIN Line

The main line first calls SETASO to initialise variables, load device handlers, then continuously tests the following:

(i) If a new telephone connection is required (call DIALUP if mdial < 0).

(ii) If a slave reply requires processing (call RECEPT if recflg changed).

(iii) If a command requires sending again after a timeout (call SENSLA if senflg > 0).

(iv) If a VCR controller requires servicing (call VCRSTR if vcrflg > 0).

(v) If the QMS data are ready to be requested (call SENSLA('C','U') if quaflg = 2).

(vi) If a 'change detected' picture requires requesting (call SENSLA('C','P',movflg) if movflg > 0).

(vii) If a new automatic command is ready to be sent (call PROCES if proflg > 0).

(viii) If the keyboard data string requires servicing (if com(1) = ITTINR() > 0).

(ix) If a regular status request requires sending (call SENSLA('C','S') if com(1) = 'M' and timflg > 0).

B5.2 SETASO

This subroutine initialises variables and loads the following special device handlers:

DR	To communicate with the VCR controller parallel line.
DB	To communicate with the frame store (DMA device).
LR and LW	To communicate with the telephone serial line.

LR has two modes, depending on the number of words (n) requested:

if n=0	it will return on the first CR;
if n>0	it will collect n*2 characters starting with a '1' and any delay will terminate collection and replace the first '1' with '0'.

B5.3 DIALUP

This subroutine carries out the functions required to establish an encrypted telephone link, as described in **section 9**, and consists of seven interrupt driven steps indicated by the modflg flag. It makes use of the following:

(i) ITIMER and completion routine TIMRTA for returns after a required delay.

(ii) IREADF, LR and completion routine LRRTA for return when the modem acknowledges.

(iii) IWRITE and LW to send modem command.

(iv) INDEX to check modem response.

(v) LOGM to write out appropriate message.

(vi) UNDIAL to break the telephone link.

UNDIAL in turn, uses IPOKE to add a 1 to the status register for two seconds to force the modem to disengage the line.

B5.4 RECEPT

This subroutine processes the slave replies as outlined in **section 4** and expects the string to be in the form given in **appendix A**. The following steps are involved:

(i) Check if the string is of correct length; if not, check if a SYN (end of picture data) reply; otherwise ignore.

(ii) Cancel timeout timer (call ICMKT).

(iii) Check if picture type data (*i.e.* 'B' or 'A'); if so, see step (vi).

(iv) Check if CRC is OK (call CHKREP); if not, call SENSLA(,) to send command again.

(v) Test type of reply, and if byte 2 equals

(a) 'T' or 'Q'....call EXTRAC (Doric) or QCONV (QMS) to write out the data in correct form.

(b) NAK....examine bytes 3 and 4 to decide on action. All except 'TL' will result in a call to LOGM to write out appropriate message. 'TL' reply will result in a call to LOGM to write out appropriate message. 'TL' reply will result in a 'sleep' of half a second followed by a call to SENSLA to repeat command.

(c) ACK....examine byte 3 and if it equals
'S' or 'Q'...return
'P' or 'L' or 'F'...call SENSLA(,) to request first block.

'D'...call SENSLA to request Doric data.

(d) SYN....examine byte 3 and if it equals
'B'...set VCR record flag vcrflg.
'A'...close floppy file (call DISKC)

The above non-data replies call ANALYS to get an update of slave status flags and to set QMS data ready and 'change detected' picture flags.

(vi) Check on type of long reply; if byte 2 equals

(a) 'B'....call CHKREP for CRC check; call SENSLA('B', next block no); and call FRAMST to load strip into frame store.

(b) 'A'...similar to 'B' except call DISKST to load half the disc block to the floppy disc and do a full CRC check.

B5.5 SENSLA

This subroutine is called to prepare and send a command string to the slave. The string is made up as outlined in **section 4**. The routine adds a count and checksum number to the string before sending. It uses

(i) IWRITE and LW to send string;

(ii) IREADF, LR and completion routine LRRTA to await reply;

(iii) ITIMER and completion routine TIMRTA for return if no reply;

(iv) LOGM to write out if more than 10 repeats are made.

B5.6 VCRSTR

This subroutine controls the VCR controller as outlined in **section 6** and consists of three interrupt driven steps. It uses

(i) IREADF and DR and completion routine VCRRTN for return when the controller acknowledges;

(ii) IPOKE to write command to the controller's parallel line ("045 for record and "02 for stop);

(iii) ITIMER and completion routine TIMRTA to set recording time of nine seconds;

(iv) LOGM to write out the appropriate message.

B5.7 PROCES

This subroutine sends commands automatically, as outlined in **section 2**, doing one of the following steps:

(i) Call SENSLA('C','D') for Doric measurement.

(ii) Check if it is time for QMS measurement, and if so, call SENSLA('C','Q','P or F') depending on product to feed ratio; re-set next QMS command with call to ITIMER with requested time interval.

(iii) Call SENSLA('C','P',kcam); increment camera number, kcam.

APPENDIX C
SLAVE PROGRAM

C1 SUBROUTINES

```
MAIN
    SETSLA
    ANSWER
        CHKCOM
            CRC
        ANALYS
        QCONV
        SENASO
        WOUT
        SENDOR
        SENQUA
        DISKO
        DISKRD
        DISKC
        PSELEC
    PSTR
    PSTRIP
    DORIC
    QUBOOT
    PPRT
    PSELEC
    PCOMP
        DISKST
    SENSLA
```

C2 COMPLETION ROUTINES

TIMRTS	LRRTN	QRRTN
DORRT1	DORRT2	DORRT3

C3 RT11 ROUTINES

IREADF	IWRITE	ITIMER	ICMKT
CONCAT	IPOKE	IPEEK	ISLEEP
IRAD50	LOOKUP	IFETCH	IENTER

C4 DEVICE HANDLERS

LR	LW	(telephone)
QR	QW	(QMS computer console)
DR	DW	(Doric)
D1		(Doric (input) and video switcher (output))
DB		(framestore)

C5 OUTLINE OF MAJOR ROUTINES

C5.1 MAIN line

The main line first calls SETSLA to initialise variables and load device handlers, then continuously tests the following:

(i) If an ASO command requires processing (call ANSWER if recflg > 0), and on on return from answer test, if

(a) picture transmission store requires loading from frame store (call PSTR if newp = 3);

(b) the next picture strip requires loading from transmission store to output buffer (call PSTRIP if ipcx > 0).

(ii) If Doric requires processing (call DORIC if dorflg > 0).

(iii) If QMS requires rebooting (call QUBOOT if qualflg > 0).

(iv) If a detection picture store requires loading into frame store (call PPRT if lkflg > 0).

(v) If the keyboard data string requires servicing (if com(1) = ITTINR() > 0).

(vi) If a detection camera requires switching to the frame store (call PSELEC if comflg < 0 and picflg > 0).

(vii) If a detection picture comparison is required (call PCOMP if comflg < 0).

C5.2 SETSLA

This subroutine initialises variables and loads the following special device handlers:

DR and D1	to communicate with the DORIC logger parallel line.

DR allows either interrupt a or b to be set, and D1 returns three words consisting of low byte in D1, and two bytes in DR:

DB	To communicate with the frame store (DMA device).
QR and QW	To communicate with the QMS console serial line.
LR and LW	To communicate with the telephone serial line.

C5.3 ANSWER

This subroutine processes the ASO commands, as outlined in **section 4** and expects the string to be in the form given in that section. The following steps are involved:

(i) Check if CRC is OK (call CHKCOM); if not see step (vi)

(ii) Check type of command; if byte 2 <> 'C' see step (v)

(iii) Check if a repeated command (*i.e.* byte 5 incremented); if repeated, call WOUT to send previous reply again.

(iv) Test type of command and if byte 3 equals

(a) 'S'... call SENASO to acknowledge.

(b) 'D'...set dorflg = 1 if Doric available (dorflg > -11), call SENASO to acknowledge; if not, call SENASO to send NAK.

(c) 'T'...check if Doric ready(dorflg = -10), call SENDOR to send data; if not, call SENASO to send NAK.

(d) 'Q'...check if QMS ready (quaflg = 1) and there is feed flow (Doric data); if OK, send command to QMS DW; wait for QMS to acknowledge; call SENASO to acknowledge ; if not, call SENASO to send NAK.

(e) 'U'...check if quaflg = 5, call SENQUA to send QMS data; if not, call SENASO to send NAK.

(f) 'P' or 'L'...call PSELEC to switch camera (only if not a 'change detected' picture request (comflg < 1)); set flag to load transmission store (newp = 3); set flag to load first strip into output buffer (ipcx = 1); if 'L', set comflg to prevent termination; call SENASO to acknowledge.

(g) 'F' ...call DISKO to open floppy file; call SENASO to acknowledge.

(h) Otherwise call SENASO with NAK and '?'.

(v) If command is not of type 'C' then test if byte 2 equals

(a) 'B'...call IWRITE to send picture strip in output buffer (previous buffer if previous strip required); set ipcx to load next strip into buffer; or

(b) 'A'...call IWRITE to send output buffer; call DISKRD to load next half floppy data block into buffer.

(vi) (CRC failed)...check if new telephone connection has been made (*i.e.* string starts with "'01','$','$','$'); call SENASO to acknowledge after five second 'sleep'; if not call SENASO with NAK and 'ET'.

C5.4 SENASO

This subroutine is called to prepare a reply string for ASO. The string is made up as outlined in **appendix A**. The subroutine calls WOUT to send the string.

C5.5 WOUT

The subroutine is called by SENASO, SENDOR and SENQUA to send non-data replies to ASO. It uses

(i) IWRITE and LW to send string.

(ii) IREADF, LR and completion routine LRRTN to enable the next ASO command to interrupt the slave.

C5.6 DORIC

This subroutine controls the DORIC data logger as outlined in **section 7**, and consists of 12 interrupt driven steps indicated by flag dorflg. It converts the complex logger BCD output and uses

(i) IPOKE to write commands and channel numbers to the logger's parallel line;

(ii) ITIMER, ICMKT and completion routine TIMRTS to time out if logger fails to respond; and

(iii) IREADF, DR and D1 and completion routines DORRT1, DORRT2 and DORRT3 for return when the logger acknowledges.

C5.7 QRRTN

The completion routine carries out the QMS monitoring, as outlined in **section 8**. The subroutine is entered every time QR receives a string from the QMS. When a measurement is initiated, the subroutine increments the flag quaflg, as the first byte changes through the cycle:

'S' to 'F or P' to 'M' to 'C' to 'S'

quaflg = S Indicates measurement ready.

The subroutine also stores intermediate values as each 'C' is obtained. The subroutine uses

(i) ITIMER, ICMKT and completion routine TIMRTS to re-set the watch-dog timer; and

(ii) IREADF to QR and itself to get the next QMS write.

C5.8 PSELEC

This subroutine is used to switch the frame store to any of the five cameras. Bits 0 to 2 of the parallel line controlling the video switcher carry the camera number and bit 3 is kept on during a 0.3 second 'sleep'.

C5.9 PSTR, PPRT, PSTRIP and PCOMP

These subroutines use virtual store to hold the frame store data. The data are compressed into words rather than bytes to allow addressing within the range 0 to 32000. PSTR, PCOMP and PPRT use IREADF, IWRITE and DB to access the frame store. PCOMP checks on pixel change, as outlined in **section 6**.