DRAFT ONLY
COMPUTERS AND PROCESSING
AT/25.1.1/024
28/9/85

The Culgoora Data Collection System

1.0 INTRODUCTION

In this document an attempt is made to set out in one place brief descriptions of the various components of the Culgoora data collection system and our concept of how these systems will function together. Both hardware and software aspects are covered. This document provides a reference as to the system being provided by the AT computer group and will be updated as changes occur.

Some preliminary specifications for the processes (programs) to run in the synchronous machine were set out in AT/25.1.1/005. This note updates these specifications to reflect current views on how the synchronous software will function. While much of original concept remains unaltered, there are some significant changes in the method of implementation.

2.0 DATA COLLECTION HARDWARE

2.1 Local Area Network

The AT computing system at the Culgoora central control building will comprise a number of processors linked by an Ethernet local area network (see AT/25.3/003).

```
                  THE CULGOORA NETWORK

                       Ethernet
        ---------------------------------------------
       !              !              !              !
       !              !              !              !
   -----------   -----------   ----------   ----------------
  !           ! !           ! !          ! !                !
  !   SYNCH   ! !  ASYNCH   ! !    CCC   ! !      ALC       !
  !           ! !           ! !          ! !                !
   -----------   -----------   ----------   ----------------
                                              ! ! ! ! ! ! ! !
                                              ! ! ! ! ! ! ! !
                    Serial links              ! ! ! ! ! ! ! !
    ------------------------------------------  ! ! ! ! ! ! !
   !         -----------------------------------  ! ! ! ! ! !
   !       !        ------------------------------  ! ! ! ! !
   !       !       !        ------------------------  ! ! ! !
   !       !       !       !        ------------------  ! ! !
   !       !       !       !       !        ------------  ! !
   !       !       !       !       !       !        ------  !
   !       !       !       !       !       !       !        !
 -----   -----   -----   -----   -----   -----     !        !
! ACC ! ! ACC ! ! ACC ! ! ACC ! ! ACC ! ! ACC !    !        !
!ANT#1! !ANT#2! !ACC#3! !ACC#4! !ACC#5! !ACC#6!    !        !
 -----   -----   -----   -----   -----   -----     !        !
                                                   !        !
                                                  ------  -----
                                                 ! PKS.! ! ACC !
                                                 !(TID)! ! S.S.!
                                                  -----   -----
```

The processors on the network are to perform the following functions:

Synchronous Computer (SYNCH): From this machine it is possible to establish communications paths to all array hardware for the purpose of controlling and monitoring observations. Each observing program which runs in SYNCH operates on the basis of a fixed data collection cycle and SYNCH is responsible for all tasks synchronized to this cycle. Often the cycle time will be set by the correlator integration period which is expected to range from 1 to 20 sec (the present software design probably will not support a cycle less than 1 sec). SYNCH must be able to support a number of simultaneously executing observing processes.

Asynchronous Computer (ASYNCH): Programs not tied to the data collection cycle run on this machine, for example, scheduling programs and programs to analyse observations of calibration sources. Normally only a limited amount of validity checking of data takes place in SYNCH, and the main analysis programs execute on ASYNCH. There are two sources of data available for these programs; 1. visibility data from the correlator which ~~is~~ are written directly to disks shared by SYNCH and ASYNCH, and 2. monitor data written to the archival data base by each observing program.

Correlator Control Computer (CCC): Dedicated to control of the AT correlator and its attached array processor. It may be decided that two separate correlator control computers are needed, one for the compact array correlator and one for long baseline correlator. For the present a single CCC is assumed, but whatever the final choice it can readily be accommodated within the general framework proposed in this document.

Antenna Link Controller (ALC): Responsible for handling antenna communications links and also the central site control/monitor bus and its associated data sets.

Present plans are to employ VAX 11/750s operating under VMS for SYNCH and ASYNCH, a microVAX I running VAXELN for ALC, and a microVAX II with microVMS for CCC.

2.2 Wide Area Network

Communication links between Culgoora and other sites involved in AT operations are required in order to satisfy two distinct requirements;

1. Sites at which LBA antennas are located need real-time links with Culgoora to provide for control and monitoring during the course of observations (see next section).

2. A wide area network is required linking the various AT sites and the Divisional headquarters at Epping. This network would need to provide the following capabilities:

- The ability to remotely monitor the current AT observations by providing a subset of the displays available to the array operator at Culgoora,
- Transfer of programs and small data files (eg. observing requests) by telescope users,
- System monitoring by engineering and maintenance staff by allowing access to monitor data stored in the data bases at Culgoora.
- Exchange of electronic mail,
- Maintenance and development of software,
- Transmission of "change" information to allow databases at different sites to be kept current.

The essential sites which must be linked by the wide area network are Epping, Parkes and Culgoora. This network might advantageously be extended to included AAO facilities at Epping and Siding Spring. The network nodes would all be VAX processors and standard DEC interprocessor communication hardware and software would need to be supported(eg. VAXPSI, DECNET). The types of links being considered are Austpac and Telecom private lines.

2.3 Antenna Communication Links

Compact array antennas: The computer network at Culgoora is further extended by providing asynchronous serial links from ALC to antenna control computers (ACCs) located at each of the six compact array (CA) antennas. These links will carry the control/monitor data required for real-time operation of the antennas. The specific software and hardware selected for the ALC and the ACC does not provide support for any standard communications protocol and we are planning to implement a simplified version of DEC's DDCMP (Digital Data Communications Protocol).

Long baseline array antennas: The method of C/M communication with the antennas of the long baseline array (LBA) has yet to be decided. One possibility is to employ the same method as for the compact array but use Telecom leased lines or even dial-up connections over the Telecom public switched network. Other possibilities include a satellite channel (if LO phase stabilization is done via satellite) or else Telecom's public packed switching network (AUSTPAC). Some mixture of the above is also possible. While any of the various possibilities can be accommodated fairly readily, it will be assumed for the purposes of this note that Parkes and Siding Spring are linked to the ALC by Telecom leased or dial-up lines. It is further assumed that C/M traffic to/from Tidbinbilla will go via Parkes and the existing Tidbinbilla-Parkes microwave link.

2.4 C/M Busses

The ALC and each ACC will support one or more special serial control/monitor busses. Data sets provide a standard interface between the C/M bus and individual pieces of equipment which require servicing. The C/M bus on the ALC services central site equipment while antenna equipment is serviced by the ACC.

Whereas Siding Spring will be equipped with an ACC and C/M busses, the type of interface required for control/monitor at Parkes has yet to be decided.

Data Sets: Equipment located at each antenna of the compact array and Siding Spring will be serviced by the ACC via one or more control/monitor (C/M) busses. The proposal is to use a full-duplex multi-drop serial link conforming to RS-422 and operating at 38.4 kbaud.

The 'data set' (DS) provides the standard interface between each individual piece of equipment and the C/M bus. A simple protocol will allow communications between the ACC and the data sets. A DS is implemented as a plug in module with an 8-bit Intel 8031 microcomputer to handle communications with the bus, as well as the collection of digital and analog monitor data and the dissemination of digital control data. An A/D converter and analog multiplexer will provide the ability to monitor a number of analog inputs. An optional secondary board might be used to increase the number of analog monitor points beyond those provided on the primary board.

2.5 SERVO Computer Interface

The antenna control computer(ACC) and the servo control computer(SERVO) will function together in the following way:

- A standard dataset will handle the communications; this may be placed on a second dedicated RS422 C/M bus.
- ACC will send Az-El position commands at 15 times per second. As the servo response bandwidth is about 1Hz, some missed requests would normally be undetectable. If SERVO fails to receive any commands within a one-second interval it will assume a malfunction and will automatically stow the antenna.
- ACC will read a summary status word from SERVO prior to sending each new position command. Provided no malfunction is indicated. a full status read-out will be performed at less frequent intervals, say every sec.
- Two modes of operation are accommodated: track and slew. Slew mode must be explicitly selected by the ACC before a position change in excess of ~1arcmin is requested, otherwise the request will be treated as an error.
- SERVO will form RMS tracking errors, either accumulated over the last second or since the last interrogation.

2.6 The AT Clock System

All Culgoora timing will be disseminated on a serial clock bus. There is a single time source, the AT Clock, which is situated at the central site and generates the time frames transmitted on the clock bus. A time frame is 1-millisecond in length and comprises 1000 bits generated using a pulse width modulation scheme. Each clock frame contains a sync sequence, status information, and CRC characters, together with the following time information;

- BAT (Binary Atomic Time) as a 48-bit binary msec count since Julian Date 2,400,000.5 (1858 Nov 17 0h UT). BAT is the local approximation to IAT, but based on the local frequency standard, possibly a Rubidium oscillator.
- UTC in BCD as YYMMDDMMSS.SSS
- dUTC, in whole binary seconds (increment applied to UTC to give IAT)
- dUT, in whole binary msec (increment applied to UTC to give UT)

A number of different types of clock bus interface boards are provided. These boards plug into a host processors Q-bus and allow the serial clock information to be utilized by the processor. The boards being implemented for the AT are as follows:

Clock Bus Receiver (Time Grabber): The 'time grabber' permits a host processor to read any of the time information contained in the clock frames.

Event Trigger: Allows the generation of formatted pulses at programmable rates from 5Hz to 1kHz. The pulses may be used as trigger pulses for external equipment or to generate regular host interrupts.

Event Generator: Permits the generation of external trigger pulses at any desired times with a 1microsec resolution. The "events" occur when the selected times down-loaded from the host match the time derived from the clock bus.

THE CLOCK SYSTEM

```
  --------   ------    ------    ----------------
 !        ! !      !  !      !  !                ! Other
 ! SYNCH  ! ! CCC  !  ! ALC  !  !       ACC      !:::::>>
 !        ! !      !  !      !  !                ! ACC's
  --------   ------    ------    ----------------
    !Unibus    !         !         !      !      !
    !          !         !         !      !      !
 CONVERTER     !         !         !      !      !
    !Q-bus     !Q-bus    !Q-bus    !Q-bus !Q-bus !Q-bus
  __!_       __!_      __!_      __!_   __!_   __!_
 ! TG !     ! TG !    ! TG !    ! TG ! ! ET ! ! EG !
 !____!     !____!    !____!    !____! !____! !____!
    !          !         !         !      !      !
    !__________!_________!_________!______!______!______
          Clock Bus
```

The central site computers SYNCH, ALC, and CCC are to be equipped with time grabbers and the appropriate software handlers to permit subroutine calls to read the time on the clock bus. SYNCH requires a Unibus to Q-bus converter. This arrangement should allow synchronization of processes running in these machine to an accuracy on the order of ten msec, even allowing for the unpredictable delays which will occur in the operating systems employed. Event generators may also be required on CCC or ALC to provide host interrupt or trigger pulses for external equipment.

The clock bus disseminates time to each compact array ACC which is equipped with a time grabber, event trigger, and event generator. The time information are required for coordinate conversion and to control of fringe rotation and sampling. The event trigger will be employed to control the coordinate conversion cycle (about 15Hz). For accurate absolute timing at each antenna an allowance needs to be made for the transmission time of the clock frames from the central site, which could be as large as 30-microsec for optical fibres.

3.0 OVERVIEW OF THE SYNCHRONOUS COMPUTER SOFTWARE

3.1 Programs Required

The hierarchy of the processes in the synchronous machine will looks like:

```
                          ---------
                         !         !
                         !  OPER   !
                         !         !
                          ---------
                              !
                              !
            ----------------------------------------------
          !           !           !          !          !
 -----------  -----------  ---------  ----------  ---------
! Observing !! Data Base !! Control/!! Hardware !! Array   !
! Processes !!  Server   !! Monitor !! Control  !! Status  !
!           !!           !! Server  !! Processes!! Display !
!   xxxOBS  !!   DBSER   !!  CMSER  !!  xxxCON  !! STATUS  !
 -----------  -----------  ---------  ----------  ---------
```

OPER: Provides the operator interface to the AT. It initiates and controls all other processes.

OBS (Observing Processes): Allow data collection for each type of measurement which needs to be supported. A prime requirement of the software design is the ability to support simultaneously running observing processes. Another requirement is that monitor data from one such process be available to any other simultaneously executing observing process. Observing processes are designated by the generic name OBS, specific observing programs have preceding characters added to make unique names, eg., CAOBS.

SER (Server Processes): Because OBS processes and some other processes must cooperate in their use of resources, a number of server processes are specified (generic name SER) These are capable of accepting and processing requests from more than one source. The need for two specific server processes has been identified:

- DBSER for handling data base requests,
- CMSER for handling control/monitor requests.

CON (Control Processes): A number of different types of array hardware (antennas, correlators, environmental sensors, etc) must be accessed to satisfy the C/M requirements of OBS programs. To fulfill this need, each piece (or type) of hardware is handled by its own control process (generic name CON) which contains all the information specific to that equipment and can collect monitor data from it and send control requests to it. A control process may either be located on the local network node (SYNCH) or on a remote node of the local area network (CCC, ALC). If a wide area network (supported by DECNET or VAXPSI) is implemented for handling Parkes or Siding Spring control/monitor data (as opposed to serial asynchronous lines from ALC), then control processes would run on these nodes as well.

STATUS: Generates displays based on information obtained from executing processes to assist the operator in diagnosing and abnormalities or equipment malfunctions.

3.2 Interprocess Communications

3.2.1 Mailboxes -

General ***~~Advantages:~~*** Mailboxes are to be employed for interprocess communications in instances where only relatively short messages are involved in non time-critical circumstances. It is anticipated that mailboxes can be used for transferring most control/monitor data. The following advantages are offered by mailboxes;

- They can be made to look like simple I/O so that processes can be easily tested in isolation with the mailbox I/O being supplied by a terminal,
- They ensure synchronization between processes without the excessive use of flags often needed with shared common.

Communications between processes running on different nodes of the LAN requires DECnet, but this can be done in a way functionally equivalent to using mailboxes. To simplify the following discussion the term "mailbox" is applied to interprocess communications, even when the processes are on different network nodes.

Mailbox creation: Mailboxes will be created at one point and permanently. This allows all other processes to treat I/O through mailboxes in a device independent way. A set of standard names will have to be assigned to the created mailboxes and a list of the mailboxes to be created will have to be held in a file (which can be modified before running OPER). Each pair of programs wishing to communicate over a mailbox link will have the mailbox name assigned at compile time (via include files). So each new mailbox link will require changes to include files and the OPER mailbox list file.

Termination Mailbox Handling: All processes run by OPER will be launched with a termination mailbox defined. This will be used by OPER to warn of the premature death of any process under its control. OPER will send a message to STATUS (see later) to allow it to clean up display areas allocated to the now deceased process.

3.2.2 Shared Common -

It is anticipated that shared common will be employed for the following purposes:

- Monitor data sent back to CMSER by control processes (via mailboxes) will be placed in a shared common area so as to be accessible to any OBS process. The control process responsible for collecting correlator visibility data (VISCON) may write directly to shared common rather than using a mailbox to CMSER.
- All SYNCH processes have a shared common area which can be accessed by STATUS and used for generating displays at the operator's console. This provides a 'window' through which the detailed functioning of each process can be viewed. No synchronization is required in this situation, the processes does not care if STATUS looks at the data placed in the common area or not.

4.0 OVERVIEW OF THE CONTROL/MONITOR SYSTEM

4.1 General

Essential information about every control/monitor (C/M) point is stored in the AT data base and can be accessed by any process via DBSER. Each point is assigned a unique name which an OBS process can specify without it needing to be concerned with the specific details as to how the point is actually serviced. OBS requests for C/M data are forwarded via mailbox (MBx) to CMSER which handles the task of coordinating these requests with any which are already current from other OBS processes running simultaneously. CMSER accesses the data base via DBSER to determine the particular control (CON) process required and obtain any other information needed for servicing each C/M point. The control/monitor requests are forwarded by private MBx to the appropriate control process and CMSER also handles the monitor data sent back from the control process via a public MBx (or possibly shared common for CON processes in SYNCH). Returned monitor data are placed in a shared common area where it can be accessed by any OBS process.

Control processes provide the means whereby control/monitor requests are ultimately handled. The majority of C/M points are accessed via special C/M busses and their associated data set interfaces to the hardware. In this case a C/M point is uniquely determined by specifying the C/M bus number, the data set number on that bus, and a specific address within the data set address range. AS distinct from "hardware access", there is another class of servicing referred to as "software access". Here the name of the C/M point is recognized by the control process and code is executed to take the appropriate action. This action may involve forming a new monitor value from other monitor data (eg., generating a quality index or a margin indication) or setting a variable to the supplied control value.

4.2 The C/M Data Base

The C/M DB file will contain at least the following fields;

For Control Points

- Unique name assigned to control point
- Name of associated hardware system
- Name of associated control process
- Hardware or Software servicing required
- Hardware access address (for hardware servicing only)

- Type of command value required (integer, real, logical, character)
- Command length (bytes)
- Optional list of permitted values eg.,(ON,OFF), (OPEN,CLOSE), (0,1,2,3.4)
- Validity dates for this point (begin and end)
- Current validity status

For Monitor Points

- Unique name assigned to the monitor point
- Name of associated hardware system
- Name of associated control process
- Hardware or Software servicing required
- Hardware access address (for hardware servicing only)
- Upper and lower margins (optional)
- Type (integer, real, logical, character)
- Length (bytes)
- Validity dates for this point (begin and end)
- Current validity status

The data base needs to be kept current and always reflect the actual conditions in the associated hardware if the system is to function correctly. Thus, when data collection is in progress the DB will be locked against any changes to the entries for control/monitor points currently being serviced.

4.3 Cycling Control and Synchronization Considerations

Following an initial setup phase, an OBS process will normally proceed by calculating a start time (TS) and a time increment (DT) and commence the data collection phase. TS and DT will be referred to as the cycle parameters, the start time of cycle "n" is DS+(n-1)*DT and its end time is DS+n*DT. CON processes also will generally run on a fixed cycle time, which would naturally be synchronized to the OBS process (ie.,it would have the same cycle parameters) for which it is servicing control/monitor requests. However, in the general case any CON process may provide monitor data for a number of different OBS processes, and clearly it can only be synchronized to one of them if OBS cycle parameters can be set arbitrarily. The following procedure is proposed to solve this problem.

The basic simplifying assumption is that each CON process always executes with a single set of cycle parameters and thus can be synchronized to only one OBS process. Other OBS processes may request cyclic monitor data but it will be collected on the cycle determined by the OBS which first set the synchronization. However, all cyclic monitor data is available to all OBS processes because it is all placed in shared common by CMSER. If no OBS process specifies cycle parameters then the CON processes will use default values supplied from the data base via CMSER.

CMSER arbitrates requests from OBS processes to set monitor points and CON cycle parameters. Each request for cyclic servicing of a monitor point includes the type of synchronization required as: "none", "desirable", or "mandatory". For any particular CON process only one OBS process can request monitor data with mandatory synchronization, otherwise a fatal error will be produced.

4.4 Types of Monitor Requests which are supported

Two distinct types of monitor requests can be made by an OBS process. These are;

Cyclic Requests: The specified monitor data are collected on every cycle of the associated CON process, as set by its current cycling parameters. These types of requests contain the cycle number for which the sampling is to start or, in the case where monitor points are being deleted, the last cycle number. Cyclic monitor data are forwarded back to CMSER as a single block from each CON process and placed in shared common, along with the associated CON processes cycle number. Some CON processes may also support a separate 'slow' monitor list which generates blocks of cyclic monitor data every "N" (N is specified) normal monitor cycles.

Immediate Requests: These requests are forwarded to CON processes which are expected to service them immediately and return the resulting monitor data to CMSER, tagged with the cycle number. CMSER then placed the data in shared common and returns it via a MBx to the requesting OBS process.

4.5 Types of Control Requests which are supported

Two distinct types of control requests are possible from an OBS process. These are:

Cyclic Commands: Commands which are sent to CON processes for execution at the beginning of the specified cycle number (normally the next one).

Immediate Commands: These are sent to CON processes which are expected to execute them immediately.

4.6 OBS access to monitor data in shared common

When an OBS process makes a request for cyclic monitor data, CMSER returns the address information via MBx indicating where the monitor data will be placed in shared common when it is received. The OBS process then may access the monitor data in shared common in two possible ways;

- Synchronized. This is the method employed by any OBS process to access monitor data received from CON processes with which it is synchronized. In this case it is usually necessary to check the CON process cycle number which is also placed in shared common to ensure that the monitor data are actually received by the requesting OBS process within the correct OBS cycle period.
- Latest Value. OBS just takes the current value as it stands. This is the only choice for monitor data collected by control processes which are not synchronized to the particular OBS cycle. However, the time interval can be determined by reading the cycle number and converting it to the start and stop times of that cycle, knowing the cycle parameters for the particular CON process (also available from the shared common area).

4.7 Specification of control/monitor points

Each OBS process needs to construct lists of C/M points required both in its initialization and its cycling phases. Some of this information is derived from the command input file (based on the experimenters specifications) and some from the from standard lists applicable to the particular OBS. Each monitor point needs to have the type of synchronization required. A list of monitor data to be archived also has to be supplied.

4.8 Monitor Archive

For the moment it will be assumed that each OBS process is responsible for archival of monitor data it collects and it will write these data directly into the data base. This procedure is relatively straightforward to implement and has the advantage of allowing easy and immediate access to the monitor data via the vendor-supplied DB query and report programs. However if tests show that the DB is overwhelmed, then some alternative scheme for monitor archive will need to be devised.

Before the OBS data collection cycle commences, the DB specification record is set up which contains the names of monitor points to be archived. Thereafter one DB record containing monitor values is written for each OBS cycle. An option will be provided to allow monitor data from a specified number of consecutive cycles to be combined by OBS before being written to the DB so as to reduce the volume of archival data.

5.0 SPECIFICATIONS FOR SYNCHRONOUS COMPUTER PROCESSES

5.1 OPER

The Boss - all programs have their origin in this one.

Functions: OPER performs the following functions:

1. Process creation - OPER is the process that creates the other processes that make up the job tree running in the synchronous machine and on the other nodes of the Culgoora LAN. The server processes, CMSER and DBSER, are initiated first, followed by specified OBS processes. After initialization an OBS process supplies OPER with the names of CON processes it needed for conducting its measurements and these processes are then started by OPER. OPER also will terminate CON processes when all the OBS using them have completed.

2. Operator command input - the array operator passes commands to on-line processes via OPER.

3. Status logging - each on-line process will send back a short status message whenever it has a major state change. This message will be displayed on the terminal used for command input and written to a log file. This file should be open for simultaneous read access by other processes.

4. Script file control - an option will be to have OPER accept commands from a script file. This file will used to start up the system and conducting standard observations. A syntax of the form @file should be used.

5. Mailbox creation - all mailboxes for process communication will be created by OPER (see section on mailboxes). In the following specifications it is assumed that "mailbox" includes the functionally equivalent procedures set up for communicating between processes on different nodes of the Culgoora LAN.

6. Termination mailbox handling - all processes run by OPER will be launched with a termination mailbox defined. This will be used by OPER to warn of the premature death of any process under its control. OPER will send a message to STATUS (see later) to allow it to clean up display areas allocated to the now deceased process.

7. Remote control - terminal input and output must be able to be redirected over the network, including the wide area network. This will enable the on-line software to be run from any node, including Parkes and Epping.

Command Input: SYS$INPUT for OPER is directed to the terminal keyboard used by the array operator. Thus OPER is the primary entry point for all commands to the on-line software.

Input should not require any special escape sequences or control characters which are not readily available on a VT100. No special function keys will be used. Standard terminal independent procedures will be used for command input.

Command syntax will be of the form:

'process_name' 'operation'

'process_name' is the name of the recipient process and 'operation' is the requested action to be performed. The text of 'operation' is passed to the recipient without modification or interpretation.

Command Output: SYS$OUTPUT for OPER is directed to the same terminal as command input. The terminal should have a simple display, compatible with a VT100 with AVO (Advanced Video Option, an add on which gives reverse video etc). OPER will be responsible for deciding where the status messages will be displayed on the screen and what attributes to apply. That is, the process will send a message to be displayed without any display attribute characters. The actual message text will be prefixed by a fixed length field containing a status value. This will be of the form 'FATAL', 'INFORM' or 'WARN'. It will then be up to OPER to display the message with attributes appropriate to the associated status.

Displays: The display (or displays) generated by OPER contains the following elements:

- An area for messages which any subprocesses has tagged as 'WARN' or 'FATAL' so as to inform the operator of the situation,
- Scrolled areas which can be allocated to any of the active processes for displaying the general messages received from them,
- A display of all the processes started by OPER and their current status,
- An area for unsolicited keyboard input from the operator to allow commands to be entered.

Files In: Commands to OPER can be read from a script file. This will be used to start up the whole system automatically and will allow any of the various standard observing programs to be initiated.

Files Out: OPER will log all messages sent to it from its subprocesses. This can be used by the array operator to review operation carried out so far. This file should be open for shared reading so that any observer can read the file to check on the progress of an observation. The file will be a simple text file with fixed fields for date/time, process name, status and message text. This will allow simple record selection by process name and status level.

MBx Links In/Out: OPER establishes mailbox links for SYS$INPUT, SYS$OUTPUT and SYS$ERROR for all processes that it launches. The SYS$INPUT link will be used to pass commands to each process. All commands will be simple text which could be input via a terminal. The SYS$OUTPUT link will be used to return status messages to OPER. All messages will be simple text lines which could be displayed on a terminal.

This requirement of simple text both in and out means that all programs, which are intended to run under OPER, can be tested without OPER.

Shared Common Links: OPER will not use shared common.

5.2 CON

5.2.1 General Specifications -

General: CON is the generic name for control processes. Each control process is responsible for managing control/monitor servicing for a specific hardware system. Since more than one OBS process may be sending/requesting control/monitor data, these operations must be initially coordinated by a server process-CMSER. Thus, control processes have no direct communications with any OBS process. Control processes may reside on any node of the Culgoora LAN.

Functions: All control processes must perform the following functions:

1. Initialization- A CON process is started by OPER when an OBS process requires it.

2. Connection requests - CMSER will send a connection request upon receiving the initial control/monitor request from an OBS process. This will use the private mailbox link between CMSER and the CON process which was established by OPER.

3. Disconnect request - when CMSER has finished with the hardware (ie all OBS which use it have completed) it sends a disconnect request. OPER may send this command on behalf of a failed CMSER.

4. Control requests - these are received from CMSER and permit control over the servicing of control/monitor requests. Some commands needed are the following (detailed formats are as yet undecided):

   o Set cycle parameters (ie., start time and time increment).

   o Add/remove specified monitor points to/from list of cyclic monitor commencing on specified cycle number. The information in this command will provide the means of accessing the monitor data (eg., hardware address). Some control processes will permit both 'fast' and 'slow' monitor lists.

   o Collect and return specified monitor data immediately

   o Execute specified control commands immediately

- o Execute specified control commands at the commencement of specified cycle number.

In response to immediate monitor commands, a control process collect the monitor data and returns it to CMSER via a public MBx. Cyclic monitor data is collected and returned every cycle, usually via the public MBx, but for CON processes in SYNCH it may be written directly to a shared common area if this is specified.

Command Input: SYS$INPUT for control processes points back to OPER. Any needed commands can be sent down this line.

Command Output: SYS$OUTPUT and SYS$ERROR point back to OPER. Status messages are passed back via this link.
Files In/Out: None used.

MBx Links In/Out: CON processes use a public MBx to send monitor data to CMSER and a private MBx to receive control/monitor requests from CMSER.

Shared Common Links: A shared common area with CMSER may be used for CON processes running in SYNCH.

5.2.2 List of required Control Processes -

CON processes currently seen to be necessary are indicated in the following list:

- CORCON. Node=CCC, Correlator control
- VISCON, Node=SYNCH, Accesses visibility data output from the correlator array processor (see below).
- ANTCON, Node=ALC, Control of compact array antennas (possible LBA antennas also)
- CENTCON, Node=ALC, Controls central site hardware serviced via the C/M bus on the ALC
- RECCON, Node=ALC, LBA tape recording control via central site C/M bus on the ALC
- PLAYCON, Node=ALC, LBA tape playback control via central site C/M bus on the ALC
- ENVCON. Node=SYNCH, Controls environmental sensors

- CLKCON, Node=SYNCH (or possibly ALC). AT Clock control via dedicated serial link. UT corrections are supplied when necessary and status is sensed every second.

The exact form of the interface which allows transfer of a specified subset of visibility data from the correlator to the synchronous computer for validity checking is undecided. One proposal (see AT/25.3/003) is via an array processor I/O port and a DR11-W on SYNC's Unibus. VISCON would then be the process responsible for controlling the DR11 and making the visibility data available to OBS processes.

5.3 CMSER

General: CMSER services OBS requests for control/monitor data and forward these requests on to the appropriate CON processes. Monitor data received by CMSER from CON processes are then made available to the requesting OBS processes.

Functions: CMSER performs the following functions;

1. Initialization- Started by OPER

2. Connect request - OBS processes send initial connection requests when control/monitor service is required. These requests utilise the public mailbox link to CMSER.

3. Disconnect request - when an OBS has completed and is ceasing its operation it sends a disconnect request.

4. Control requests - these are sent by OBS processes to set up to required servicing of its control/monitor requirements Some commands needed are:

   - Set cycle parameters (start time and time increment), see below. CMSER sends these parameter to CON processes for which synchronization of cyclic monitor data is required.

   - Add/remove specified monitor points to/from list of cyclic monitor for specified cycle number. Either 'fast' or 'slow' monitor list may be specified for some CON processes.

   - Collect specified monitor data in immediate mode

   - Execute specified control commands immediate mode

   - Execute specified control commands at the commencement of specified cycle number.

5. Control/Monitor Data Handling and Synchronization- See following description.

Monitor Data Collection: Requests for monitor data are received via a public MBx from OBS processes. Monitor data requests are supported in two modes, immediate and cyclic. In response to monitor requests in immediate mode a control process collect the monitor data and returns it to CMSER via a public MBx. Cyclic monitor data is collected and returned every cycle, generally via public MBx, but for CON processes in SYNCH it may be written directly to a shared common area if specified. For each OBS, CMSER also maintains lists of the type of synchronization required (none, desirable, mandatory) for each cyclic monitor request (see below).

Adding monitor points- CMSER checks the monitor lists that it maintains to determine if each specified point is already being serviced or not. For those which are not, the DB is consulted to determine which CON process is required for servicing and to extract any required hardware access information. The monitor point name and access information are sent via private MBx to the appropriate CON process for servicing. Also include are cycle number information, mode of service required (immediate or cyclic, if cyclic then whether slow or fast), also address information if monitor data to be written directly to shared common. The DB is also locked against changes being made to the entries relating to monitor points. currently in use. CMSER allocates addresses in shared common where returned monitor data can be accessed by OBS processes. these address are returned to OBS by the private return MBx.

Deleting monitor points (cyclic mode)- CMSER removes the specified points from its monitor list, sends a MBx message to the appropriate CON process to do the same (along with the last cycle number on which to supply supply the monitor data). The DB is also unlocked so that changes could again be made to the entries relating to these monitor points.

Servicing of control points: Requests for sending control information are received by CMSER via public MBx from OBS processes. Requests are supported in two modes, immediate and cyclic. In response to control requests in immediate mode a client CON process executes the desired action as soon as it is received. Cyclic control requests are execute at the beginning of the cycle specified. CMSER maintains lists of the control requests it services and the cycle number on which they were done (this information is needed for display by STATUS)

CMSER consults the DB to extract the name of the CON process to service control requests and to determine any required hardware access information. An option will be to have CMSER check the supplied control value against the list those permitted values stored in the DB entry to determine its validity. The control point name and access information are sent via private MBx to the appropriate CON process for servicing, along with mode of service require and cycle number if applicable.

Synchronization of CON processes: CMSER has the responsibility for setting the cycle parameters (start time, time increment) for CON processes and for synchronizing these parameters with OBS cycle parameters if needed. Initially (prior to any control/monitor requests from OBS processes) default cycle parameters (possibly from the DB) are supplied to all running CON processes.

The basic simplifying assumption is that each CON process always executes with a single set of cycle parameters and thus can be synchronized to only one OBS process. Other OBS processes may request cyclic monitor data but it will be collected on the cycle determined by the OBS which initially set the synchronization.

An OBS process can supply cycle parameters to CMSER and request synchronization. CMSER will attempt to synchronize CON processes (by sending them the new cycle parameters) for which the particular OBS has requested cyclic monitor data with "desirable" or "mandatory" synchronization. A fatal error must be flagged if two OBS request mandatory synchronization of the same CON process. The current cycle parameter of CON processes can be read by an OBS process from the shared common area.

Command Input: SYS$INPUT for points back to OPER. Any needed commands can be sent down this line.

Command Output: SYS$OUTPUT and SYS$ERROR point back to OPER. Status messages are passed back via this link.

Files In/Out: None used.

MBx Links In/Out: CMSER will have a public input MBx for control monitor requests from OBS processes, and another public input MBx for monitor data returned from CON processes. Private MBxes will handle messages in the other direction, ie., CMSER to OBS processes and CMSER to CON processes.

Shared Common Links: Used for monitor data collected by some CON processes (only on SYNCH) and for making all monitor/control data accessible to all OBS processes.

STATUS also will require shared common access to lists of control/monitor points being serviced and the actual monitor data collected. Thus the operator can view, in real time, current control/monitor parameters for all CON processes via one or more "pages" on the status display.

5.4 DBSER

General: DBSER manages access to the data base files that are common to all programs running in the synchronous system. Many of the function of DBSER may be able to be handled directly by the commercial data base management software which has been purchased.

Functions: Processes send mailbox requests to DBSER. Requests are serviced by DBSER which then returns any information required and a status value showing whether the request was correctly handled. The request and reply should be given as simple text strings. The following operations can be performed:

1. Connect - a process requests a connection to DBSER. The request is made over a public mailbox and the request contains the name of a mailbox to be used to send replies to the connecting process.

2. Disconnect - when a process is cleaning up prior to exiting it sends a disconnect request. This frees all locks owned by a particular process and deletes any connection information.

3. Update - new information is inserted into the data base. The data should be checked for consistency before the data base is modified.

4. Inquire - a process requests the value assigned to an element in the data base. This command must allow for multiple items in the one packet.

5. Allocate - a process can request exclusive access to a given resource. This resource need not be physical, it is simply identified by name. DBSER is used to hold the lock data base for these resources. There must be a mechanism for unlocking resources which were locked by a failed process.

6. Free - A process can free resources that are allocated to it or some other process. All locks owned by a particular process can be freed by a single command. OPER can free resources locked by a failed process.

Command Input: SYS$INPUT points back to OPER. OPER should be able to send requests over this link.

An implicit connection to OPER is made when DBSER starts up. Requests will come in via SYS$INPUT and returns via SYS$OUTPUT. This will allow the operator access to the data base via OPER commands.

Command Output: SYS$OUTPUT points back to OPER. Status messages and the results of OPER requests are returned on this channel.

Files In/Out: The following data base files are handled by DBSER:

1. Source catalog - a version of each source catalog should be available to DBSER. This will ensure that OBS will have access to the latest source positions at run time.

2. Data dictionary - this contains the translation of all resource names to their physical realization. This will enable a process to request the value of a particular physical parameter, such as an antenna monitor or control point, by name rather than, say, a data set number and relative address.

3. Array resource file - this contains the list of resources that can be locked, along with current owner, access modes permitted etc.

4. Configuration file - this lists the values of all parameters which are measured at a configuration change such as telescope position.

5. Calibration catalog - this contains the results of calibration observations on standard sources.

6. Control/Monitor file - contains information on all points which are available.

MBx Links In/Out: DBSER accepts requests over a mail box link and sends data back via another. There is a single public mailbox for input to DBSER and connection commands establish a private return mailbox.

All commands give a process name as part of the request. This allows any process to send a request on behalf of some other process. Hence OPER can send disconnect requests to clean up after a failed process.

Shared Common Links Not used.

5.5 OBS

5.5.1 General Specifications -

General: OBS is the generic observing program. More than one OBS may be running at any one time, provided there is no conflict as to the resources required by each.

Functions: All OBS processes will perform the following functions:

1. Operation control - all control commands to OBS come from OPER. See below for a list of commands supported by OBS.

2. Script file reading - the observations to be undertaken by OBS are selected from a script file. OPER tells OBS the name of the script file to use and, optionally, a starting observation number. OBS works its way through the script file till it is told to stop or reaches the end. At this time it goes idle waiting for the next command.

3. Operator messages - OBS sends messages back to OPER each time it changes state eg. starts a new observation, aborts an observation etc.

4. Resource control - OBS must request all the devices that it needs. These include antennas, correlator modules, receivers etc. These allocation requests are handled by a data base server process, DBSER.

5. Control/Monitor requests - OBS sends these to CMSER which forwards them to the appropriate CON process for servicing. OBS also determines the cycle parameters (start time, integration time) which may be required to synchronize operations with CON processes.

6. Archive - OBS logs specified monitor data directly to the DB.

7. Status messages - OBS periodically places a package of status data in the section of memory it shares with STATUS. These data are used to update the section of the screen reserved for OBS. STATUS can choose to ignore this data, since it is displayed for information purposes only.

Command Input: SYS$INPUT for OBS points at OPER. Simple text commands are sent down this channel.

Control commands will be:

- o START 'file' 'number' - start a sequence of observations.
- o ABORT - stop the current observation and idle.
- o SET 'parameter' 'value' - modify the parameter table.

The SET command enables the operator to modify the execution of OBS in a controlled manner. For example:

SET DATA_FLAG TERRIBLE

This command allows the operator to flag the recorded data as bad during some transient event that may not be picked up by the data checking code.

Command Output: SYS$OUTPUT and SYS$ERROR point back to OPER. Status messages for the operator display are sent on this channel. Each message will have two text fields. The first is a status level, the second is the text of the message.

Files In: OBS reads observing commands from a script file which has been prepared in advance by the scheduler. The name of the file to use is passed down by OPER in the START command.

The script file should be a sequential text file which could be prepared by a text editor. To allow random start points the file should contain numbered markers at the start of the set of records associated with a particular observation. This could be done by making each observation group a namelist.

Files Out: Monitor data will be recorded in an archive file on the DB. This file is to be accessible for simultaneous read by asynchronous processes. The record descriptor initially set up for the DB file by OBS contains the names of the monitor data recorded in the file.

MBx Links In/Out: OBS establishes mailbox links with

1. DBSER - this link is used to interrogate the standard data bases.
2. CMSER - this link is used to requests for control/monitor data.

Shared Common Links: OBS will use shared common to access monitor data collected by CMSER. After receiving requests for monitor data. CMSER will and return to OBS (via the private MBx link) the address in the shared common area where the monitor data will be placed.

OBS will also share a common area with STATUS. This will permit a real-time window by which the workings of OBS can be monitored. The data in this common area will be specific to each OBS program and its current contents will be displayed by STATUS on an appropriate "page" of its display. Information displayed might include : current RA and DEC, last commands from OPER, last commands to CMSER, samples of the latest monitor data from CMSER (samples of visibility data etc.)

5.5.2 List of Required OBS processes -

A partial list of OBS programs which may be required are:

- CAOBS for compact array observations using the correlator. It is anticipated that data required for most array calibration procedures can also be collected using this process, with the appropriate reduction programs being run in ASYNCH.
- RECOBS for recording long baseline array data from a single compact array antenna or from the tied array.
- PLAYOBS for playback and correlation of previously recorded VLB observations.
- IDLEOBS can be run when the array or individual antennas are idle to allow a minimum of monitor data to be collected and displayed (any array resources used by this processes can be taken by another OBS process when required).
- POINTOBS for collecting pointing calibration data on an individual antenna (does not require the correlator).
- MAINTOBS for collecting and displaying operator specified maintenance data on any particular piece of equipment.

5.6 STATUS

General: STATUS is the program which collects and displays status information from all processes associated with the conduct of the observation.

Functions: STATUS performs the following functions:

1. Display update - STATUS maintains a status display which it updates on a regular cycle with any new data obtained from client processes via the shared common links. This display should be multipaged and in colour.

2. Display modification - OPER sends commands to STATUS to modify the selection of pages to be displayed.

Command Input: SYS$INPUT points back to OPER. Commands from the operator come via this channel. One command needed would be DISCONNECT which would be issued to clean up the display after a process failed. Commands to modify the display come down this channel.

Command Output: SYS$OUTPUT and SYS$ERROR point back to OPER. These channels are used for status messages to the OPER display.

Files In/Out: STATUS does not use input or output files.

MBx Links In/Out: STATUS does not use any mailboxes links, except to OPER.

Shared Common Links: STATUS has shared common links with any processes for which a real-time status window is needed. This includes each OBS process, CON processes running in SYNCH, and CMSER.