"STELLA" Project:

Communications Interface Module (CIM)
"STELLA" PROJECT: COMMUNICATIONS INTERFACE MODULE (CIM).

K.S. Olofsson
B.M. Segal

SUMMARY.

STELLA is an experiment on the use and performance of 1 Mbit/sec. satellite data links between 7 European laboratories using the OTS-2 communications satellite. Each collaborating laboratory has its own Earth Station (ES) and Link Driving Computer (LDC), and these are of several different types; to provide the necessary high speed ES-LDC interfaces, a special Communications Interface Module (CIM) has been designed, providing the following features:

-- Design modularity, adaptable to each ES/LDC and capable of later enhancement, e.g. addition of buffer memory, etc.
-- Full-duplex HDLC line protocol, and standard CCITT (V35 type) modem control and clock signals.
-- Basic timing control for the "master" transmitting station and for synchronization of the "slave" receiver station(s).

The design has been established by CERN in collaboration with the other laboratories, in particular with Rutherford who are responsible for the LDC software, and ESA who have fixed the earth station specification. CERN has produced a small series of CIM's to equip each of the participating laboratories with a CIM which they can interface easily to their particular ES/LDC.
I. INTRODUCTION & SYSTEM PRINCIPLES.

1. The CIM (Communications Interface Module) for the STELLA Project is a full-duplex data communications interface between each Link Driving Computer (LDC) and associated Earth Station (ES). It provides 7 standard CCITT modem circuits: TX (103), RX (104), RTS (105), DSR (107), CD (109), TSET (114), and RSET (115). It also automatically performs the link-level HDLC framing/deframing, bit insertion/deletion and CRC checking of all data blocks which are transmitted/received by the LDC, and provides associated status words to allow the LDC to monitor operation and intervene for error-recovery. It also provides the LDC with certain timing services which relieve it of critical constraints during the communication cycle. All high-speed (1 Mbit/sec) data is input/output via LDC Direct Memory Access (DMA), to further relieve the LDC processor.

A schematic view of the CIM is given in Figure 1.

2. The CIM contains a microprocessor (Motorola M6800), and two LSI special-purpose chips for the HDLC function (SMC COM5025). The microprocessor handles initialization of the COM chips, ensures transmission of enough HDLC frame flags before/after each block to satisfy modem and codec hardware requirements, and provides end-of-block interrupts and status to the LDC. Within each HDLC frame, data flows directly between the LDC memory and the modem, via the COM chips, without intervention by the M6800 MPU.

3. The CIM’s timing function ensures that all transmissions (from master or slave stations) are made during appropriate time windows in the basic operational cycle (nominally 75 msec.). These “transmit windows” are set (one for each station) before each operating period, and each CIM’s window is loaded into it when it is enabled for operation by its LDC.

The basic 75 msec. timing cycle is synchronized (and periodically re-synchronized/tuned) in each slave CIM, relative to detected incoming master station data bursts. Once started, slave timing cycles can run autonomously (subject to small drifts) during any temporary absence of incoming master bursts.

A small number of programmable timing markers within each CIM’s timing cycle may also be enabled by its LDC, each with its own LDC maskable interrupt, to further relieve the LDC of critical timing functions.

4. The CIM hardware is of modular design, particularly as concerns the LDC interface, thus allowing different model LDC’s to be connected by simple substitution of circuit boards. Provision for expansion and enhancements has also been foreseen.

A block diagram of the CIM hardware is given in Figure 2. Common to every CIM will be the portion of Figure 2 falling below the “Line of Demarcation”, i.e. the LDC-independent portion.

5. Above the demarcation line, CIM circuitry provides a 16-bit wide full-duplex interface to an LDC PIO (Programmed Input-Output) channel, used for exchange of control/status information between the LDC and the CIM. Also provided are two high-speed parallel interfaces (> 1 Mbit/sec) connecting the COM chips and the LDC memory, either:

   (1) Via an LDC DMA multiplexer (e.g. PDP-11 case),
   or: (2) Directly to LDC memory ports (e.g. GEC 4080 case).

In case (1), only the control/status dialogue passes over the PIO channel and all data blocks pass over the DMA channel. In case
(2) however, the CIM itself contains the DMA channel logic, and the PIO dialogue must be extended to allow the LDC to load the CIM's DMA registers with the memory block address and byte count for each data transfer.

6. In either case, the CIM microprocessor intervenes only for setting up block transfers (by means of the PIO dialogue), and as each block transfer is begun it breaks the bus connection to the affected COM chip so as to isolate the bus data and address lines during the transfer.

7. The CIM has comprehensive self-test software, permitting troubleshooting to be carried out progressively on the CIM itself, the earth-station equipment and the LDC DMA/PIO connections. A special hardware feature is the "Maintenance Mode", which when selected internally loops Tx to Rx serial data and lowers the data rate to 125 kbits/sec. to simplify measurements. The CIM can be set up/controlled in a "Stand-Alone Test Mode" by setting CIM option bits on two front-panel hexadecimal switches.

8. For software development, the CIM has 2 communication ports, one for connection of an asynchronous terminal, the other connecting to a serial interface from a host computer (not necessarily the LDC). CIM firmware permits the terminal to address either the CIM itself (via the standard Motorola MINIBUG system which is included in the CIM firmware), or the host computer, also allowing down-line loading of the CIM's RAM from the host. Down-line loading of the RAM from the LDC, via the PIO channel, is also provided. Enough RAM (8K bytes) is supplied so that all or part of CIM software can be executed out of RAM, though in production CIM's most or all software will reside in EPROM.
II. CIM TIMING PRINCIPLES.

1. The CIM timing system has, as a principal objective, the need to supervise data transmission and reception of blocks with typical inter-block guard times of a few milliseconds. The CIM's processing power is limited by the time needed to respond to and usefully service an interrupt; this time has been restricted to be of the order of 100 microseconds, so that useful amounts of work can be performed by the CIM in 1 millisecond.

2. For an M6800 with 1MHz clocking, this restriction translates to a requirement to limit interrupt handling (and any other processing with interrupts disabled) to sequences of less than about thirty MPU instructions; this has been achieved in the CIM software. Of course this has to be combined with priority schemes to deal with the cases of work arriving in very short bursts. Tests have shown that the CIM can simultaneously transmit and receive data blocks a few milliseconds in length with guard times of 1 millisecond, in a realistic environment.

3. Timing precision of individual events is limited by hardware (1MHz clocking and MC6840 programmable timers) to 2 microseconds. This seems perfectly adequate for the CIM's application.

4. Certain CIM processes depend on the LDC's own timing limitations, such as PIO input/output. For the PIO case, the CIM limits the LDC to respond within a few hundred microseconds per word, in order to prevent blocking of its other scheduled work; this is therefore an LDC system requirement.

III. TIMING FRAME & FORMAT.

The basic STELLA time frame and data format is shown, relative to the OTS-2 satellite, in Figure 3. A long data burst from the "master" station (about 40 milliseconds in length) is followed by shorter ACK/NAK responses from the "slave" station(s), all in a basic timing cycle of nominally 75 milliseconds. The detailed HDLC link-level data structure of each burst is also shown.

The basic CIM timing cycle is shown in Figure 4. The cycle time is set by succeeding R0 interrupts, within which occur the R1, EF, and R2 interrupts corresponding to the single transmission sequence permitted any CIM in one basic cycle (where R1-R2 delimit the "Transmit-Window"). R3 and R4 are optionally-set timing marks.

IIB. TIMER FUNCTION.

The timer hardware consists basically of 3 (or more) MC6840 PTM (Programmable Timer Module) chips, each containing a set of three 16-bit down-counters with associated initial-value registers, together with external circuitry to provide specialized control and interrupt logic. These nine (or more) counters are conceptually named and used as follows:

- **R0** -- Basic timing-cycle counter, STARTING MODE
- **R1-R4** -- Timing Markers within cycle (4 or more).
  - **RC** -- Re-synchronization Counter (Running-mode).
  - **W1-W2** -- Window Markers for use with RC.

All counters have a count rate exactly 1/2 that of the M6800 clock (which is 1.024 MHz), but this could be changed if necessary;
this chosen value allows for a maximum basic timing cycle length of 256 milliseconds (75 msec is the nominal value).

For all stations, an "End-Flags" (EF) interrupt is generated at nominally 80 character-times after the start of the Transmit Window to permit this number of HDLC Flags to precede all transmissions and allow the modem and Linkabit decoder to start up correctly. This time is adjusted by the hardware to take account of the two possible transmit bit speeds (1.0 or 0.5 Mbit/sec.), and the bit speed used in Maintenance Mode (125 kbit/sec.).

IIIC. TIMER SYNCHRONIZATION.

The Timer hardware has two operating modes, "STARTING MODE" and "RUNNING MODE", used respectively for SYNCHRONIZING and RESYNCHRONIZING a slave station's timer relative to the master station.

1. STARTING MODE.

In this mode, the default at start of operation, the counter R0* runs freely modulo the contents of its Initial-Value Register (IVR), which is preset to the nominal value 19200 decimal (corresponding to 75 msec.), and generates an interrupt each time it passes zero (and resets to its IVR value); at this time the timer hardware also restarts counters R1–R4 which simply count down to zero from their individual IVR values, then stop and generate their own interrupts. Thus a basic timing cycle (R0*) and a set of programmable timing markers within that cycle are established.

However, this cycle is unrelated to the period and phase of the master station: in order to SYNCHRONIZE, the counter R0 is used as follows: each time a leading edge is sensed on the Receive-Data line from the earth-station, R0 is reset to its IVR value (also preset to the nominal 75 msec. value), and a test made when the next trailing edge is sensed on Receive-Data. If this edge falls within a WINDOW delimited by counters W1–W2 (preset to bracket a time value of approx. 35–45 msec.), then it is assumed that R0 is synchronized to the master's data burst, R0 is allowed to run freely modulo 75 msec. and is substituted in function for R0*.

Thus the system of counters R0, R1–R4 is now in phase with the master station, but with nominal period; at this point the hardware switches automatically to "RUNNING MODE".

If the trailing edge fails to fall within the W1–W2 window, an interrupt is generated on the RC interrupt line as an indication of the failure ("NO HIT"); no further action occurs until the next leading edge is sensed, R0 is reset, and the process repeated.

2. RUNNING MODE.

In this mode, the Resynchronization process is made by software "tuning" of the period of R0 as follows: the Window counters W1–W2 are reloaded to bracket the nominal basic timing cycle of 75 msec., and the Resynchronization Counter RC has its IVR set to about 80 msec. The hardware restarts RC upon every SECOND restart of R0, and tests for a leading edge of Receive-Data sensed within the window W1–W2; if this occurs, RC is halted immediately, otherwise RC halts when it reaches its IVR value. An interrupt is generated whenever RC halts; thus the software can record successive values of the period of the master station (or note the lack of valid master transmissions), and can periodically readjust R0's IVR contents to tune the station's basic timing cycle to that of the master station.

For the special case of the master station itself, the timer is left in Running Mode permanently, with R0's IVR value fixed at 75 msec. and no tuning procedure is performed.
III. CIM SOFTWARE PRINCIPLES.

1. The CIM is restricted to only FIVE sources of external input:

<table>
<thead>
<tr>
<th>PRIORITY</th>
<th>NATURE OF INPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (top)</td>
<td>TIMER EVENTS</td>
</tr>
<tr>
<td>2.</td>
<td>PIO MESSAGES FROM LDC</td>
</tr>
<tr>
<td>3.</td>
<td>RX COM/DMA ACTIVITY</td>
</tr>
<tr>
<td>4.</td>
<td>TX COM/DMA ACTIVITY</td>
</tr>
<tr>
<td>5.</td>
<td>MODEM (AND OTHER) PHYSICAL EVENTS</td>
</tr>
</tbody>
</table>

2. Multi-tasking is not possible for the CIM: processing is SERIAL.

3. Interrupts are only produced by TIMER and DMA END-OF-BLOCK events. Only the most time-critical work is done at interrupt time, while interrupts are still disabled; less critical work is scheduled for later (interruptible) servicing.

4. In the light of the above, the overall strategy is to serialize the CIM's work into a consistent and effective order, as follows:

   'SCAN' AS OFTEN AS POSSIBLE OVER THE INPUT SOURCES 1-5, AND SET THE CORRESPONDING BIT (OF BITS 7-3 RESPECTIVELY IN THE CIM STATUS WORD) IF WORK IS TO BE SCHEDULED ON ANY LEVEL.

   'SERVICE' AS QUICKLY AS POSSIBLE, AND IN ORDER OF PRIORITY, ALL SCHEDULED WORK:
   I.E. PERFORM TO COMPLETION THE SERVICE ROUTINE FOR THE TOP-PRIORITY LEVEL FLAGGED BY 'SCAN' AND CLEAR THE FLAG BIT AFTER SERVICE ENTRY. AFTER EACH SUCH SERVICE, CHECK FOR ANY FURTHER SCHEDULED WORK, THEN RETURN TO SCAN.

5. Interrupt Handlers work as follows:

There are 8 Timer interrupt lines (IR20-7), each having one bit in the Timer Status Byte (TSB); each interrupt causes its TSB bit to be set, and in some cases also sets a service-level Flag bit if further work needs to be done. A ninth interrupt line (IR0) is provided for Rx and Tx End-of-Block (EOB) events.
(e.g. an EOB interrupt will cause setting of service-level 3 or 4, and a Timer Register R3 interrupt will cause setting of the R3 bit in the TSB, as well as the level 1 Flag bit if the LDC has requested that R3 be enabled as a timing mark).

Thus interrupt-handlers represent a second source of service-level inputs (for levels 1, 3 and 4), in addition to the 'SCAN' process, and for this reason, all service code and scanning code (being interruptible) must disable interrupts while modifying the TSB or the service Flag bits in the CIM Status Word.

6. Service Routine Details:

   The 5 Service Routines correspond to Priorities 1-5 and to Bits 7-3 respectively of the CIM Status Word. In order they are:

   TIMER SERVICE.
   Quaueed by Timer Interrupt Handler when work remains, e.g. will send a Timer Status message to LDC if current interrupt is a timer mark set by LDC; will do watchdog/housekeeping services after each N (N >= 1) basic cycles of R0 interrupts.
PIO MESSAGE SERVICE.
Queued when input is found waiting on the PIO channel from LDC. Will read the current single- or multi-word message from the input PIO interface, analyze it and execute it to completion, i.e., if a reply message is needed, will issue it on the PIO write interface.

RX COM/DMA SERVICE.
Queued as a result of an LDC Rx DMA Request message, or subsequently by an Rx End-of-Block interrupt handler. Will set up Rx COM and the Rx DMA channel, and handle the EOB condition, sending the LDC any necessary status messages on the PIO channel. Will send an error message if Rx DMA not requested when an incoming block is received, as this data will be lost.

TX COM/DMA SERVICE.
Queued as a result of an LDC Tx DMA Request message, or subsequently by a Tx End-of-Block interrupt handler. Will set up Tx COM and the Tx DMA channel, and handle the EOB condition, sending the LDC any necessary status messages on the PIO channel. Will send an error message if Tx DMA requests are made too frequently by the LDC. Ensures that transmission occurs within the correct transmit window, else generates an error status.

MODEM & PHYSICAL EVENTS SERVICE.
Queued by 'SCAN' encountering a modem or other external physical status change requiring service. Will update any status information and issue any necessary PIO messages to the LDC.

7. Initialization & Startup:

DEADSTART.
At deadstart (entry 'START0'), all necessary RAM locations are initialized, including the MINIBUG stack and RAM interrupt vectors and default values for certain RAM blocks which at Restart will be used for hardware and system initialization (e.g., the Enable-Block and the Timer Latch Block). Note that these values can be altered before a restart if non-default values are preferred, for example to change some of the Self-Test parameters.

RESTART.
At restart (entry 'START'), all the CIM hardware is initialized and all operational counts are cleared. Operation begins. This entry can also be activated by an LDC command (RESTART).

8. Memory Map:

<table>
<thead>
<tr>
<th>RAM SOFTWARE</th>
<th>EPROM SOFTWARE</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-0002</td>
<td>same</td>
<td>Jump to RESTART entry.</td>
</tr>
<tr>
<td>0003-00FF</td>
<td>same</td>
<td>DLL workspace.</td>
</tr>
<tr>
<td>0010-00FF</td>
<td>same</td>
<td>CIM Global workspace.</td>
</tr>
<tr>
<td></td>
<td>E000-E002</td>
<td>Jump to DEADSTART entry.</td>
</tr>
<tr>
<td>0100-03FF</td>
<td>E100-E3FF</td>
<td>Deadstart + Interrupt handlers.</td>
</tr>
<tr>
<td>0400-07FF</td>
<td>E400-E7FF</td>
<td>Restart + Scheduler + Scan Loop.</td>
</tr>
<tr>
<td>0800-0BFF</td>
<td>E800-EBFF</td>
<td>Service Routines + Commands.</td>
</tr>
<tr>
<td>0C00-0FFF</td>
<td>EC00-EFFF</td>
<td>Miscellaneous Subroutines.</td>
</tr>
<tr>
<td>1000-1FFF</td>
<td>F000-F7FF</td>
<td>Space.</td>
</tr>
<tr>
<td></td>
<td>F800-FBFF</td>
<td>DLL EPROM.</td>
</tr>
<tr>
<td></td>
<td>FC00-FFF</td>
<td>MINIBUG EPROM.</td>
</tr>
</tbody>
</table>
III A. CIM-LDC MESSAGE FORMATS.

The CIM and LDC communicate via the 16-bit Programmed I/O (PIO) channel, by reading/writing messages to each other. Each message is made up of one or more 16-bit words; the 1st word is a header, and if header-word Bit 15 is set, further message words follow the header:

1. SINGLE-WORD MESSAGE:

```
  15 14 11 8 7 0
-heavy-----^-----|---OPCODE---|----DATA BYTE, d-----| (HEADER)
       | 0=Receive,
       | 1=Transmit.
```

2. MULTI-WORD MESSAGE:

```
  15 14 13 11 8 7 0
-heavy-----|---OPCODE---|-----BYTE COUNT, b-----| (HEADER)
       | BYTE 1 | BYTE 2 | (TEXT...
       | etc.... |

-byte b-1 (or b) | BYTE b (or spare) | (see NOTE)

TERMINATOR WORD (if TW =1) for debugging
``` 

NOTE: The Terminator Word (e.g. all 1's) is a redundant item for use in debug phase, to reduce the danger of a header-word being lost (e.g. due to a wrong message Byte Count). It can follow Single-Word or Multi-Word messages.
### IIIB. CIM-LDC MESSAGE TYPES.

<table>
<thead>
<tr>
<th>OPCODE</th>
<th>FORM</th>
<th>VALUE.</th>
<th>DIRECTION</th>
<th>LDC-to-CIM</th>
<th>CIM-to-LDC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(SW/MW)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>--------</td>
<td>------------------</td>
<td>------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>0</td>
<td>Single-Wd</td>
<td>spare</td>
<td>Error (d=error no.)</td>
<td>spare</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Multi-Wd.</td>
<td>spare</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SW</td>
<td>Restart-Request</td>
<td>Restart/Enable Ack (d=0/1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MW</td>
<td>Enable-Request Block (see Sec. IIIC)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SW</td>
<td>DMA Request (Rx/Tx)</td>
<td>spare</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MW</td>
<td>DMA Request Block (see Sec. IIIC)</td>
<td>DMA EOB Status Block (see Sec. IIIC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SW</td>
<td>Status Req.(d=type)</td>
<td>1-Byte CIM Status (d=status)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MW</td>
<td>spare</td>
<td>d-Byte CIM Status Block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SW</td>
<td>Set/Clr Timer Marks (d=Enable bits)</td>
<td>Timer Status (d=status)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MW</td>
<td>Timer Marks Block (see Sec. IIIC)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>SW</td>
<td>CIM Dump Request (not implemented)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MW</td>
<td>CIM Reload Block (d=15; followed by PIO down-load) (see Sec. IIIC)</td>
<td>CIM Dump Block (not implemented)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IIIC. SPECIAL MESSAGE BLOCK FORMATS.

1. ENABLE-REQUEST BLOCK (LDC TO CIM):

Byte 0: CIM TYPE/ADDRESS.
        Upper nibble = 0(NORD), 1(GEC), 2(PDP), etc.
        Lower nibble = Site Address (0-15).
Byte 1: ENABLE-OPTIONS.
        Bit 0: If set, CIM is SLAVE.
        Bit 1: If set, CIM is MASTER.
        Bit 2: If set, TX EOB Status only sent if EOB error.
        Bit 3: If set, Rx COM uses Byte 0 as SECONDARY ADDRESS.
        Bit 4: If set, DMA requests set only by H/W, not PIO.
        Bit 5: If set, Rx COM runs in MAINTENANCE MODE.
        Bit 6: spare
        Bit 7: If set, CIM runs in MAINTENANCE MODE.

Word 1: OPERATING CYCLE LENGTH (in MSEC units).
Word 2: , , , , TOLERANCE ( , , ) .
Word 3: MASTER BURST LENGTH ( , , ) .
Word 4: , , , , TOLERANCE ( , , ) .
Word 5: TRANSMIT WINDOW OPEN TIME ( , , ) .
Word 6: , , , , SHUT ( , , ) .

2. DMA EOB STATUS BLOCKS (CIM TO LDC):

Rx CASE--
Byte 0: PIARTDB BITS AT RX EOB TIME
Byte 1: Rx COM STATUS , , , ,
Byte 2: LINKABIT ERRCNT , ,
Byte 3-4: Rx DMA RESIDUAL WD CNT (GEC only).

Tx CASE--
Byte 0: PIARTDB BITS AT TX EOB TIME
Byte 1: Tx COM STATUS , , , ,
Byte 2-3: Tx DMA RESIDUAL WD CNT (GEC only).

3. DMA REQUEST BLOCK (LDC TO CIM -- GEC only):

Rx OR Tx CASE--
Word 0: Rx OR Tx DMA WORD COUNT
Word 1: Rx OR Tx DMA BLOCK ADDRESS

4. TIMER MARKS BLOCK (LDC TO CIM):

Word 0: TIMER MARK R1 VALUE (in MSEC units).
Word 1: , , , , R2 ( , , ) .
Word 2: , , , , R3 ( , , ) .
Word 3: , , , , R4 ( , , ) .

NOTE:-- A zero word leaves the corresponding Timer Mark unchanged.

5. PIO DOWN-LINE LOAD BLOCK (LDC TO CIM):

The format of the reload block is standard MOTOROLA ASCII-CODED HEX:
Each 16-bit PIO word transferred is considered as two bytes, each
being a single hex character; thus each PIO word corresponds to one
hex-coded byte of data, with the high bits of each word containing
the most significant hex nibble of the data byte.
Each valid record starts with 'Sncc' where cc is a hex count of data
bytes after the count in the record; the last data byte in each
record is a checksum of all data bytes after the count. A loadable
record has n=1 and its first two data bytes after the count set to
the initial load address for the remaining bytes (except checksum).
The final record has n=9.
Note that in response to a Status-Request from the LDC, the CIM sends a 1-byte response (if the Request-type d does not exceed a limit value, currently 16), else sends a block response d bytes in length. In all cases the "status bytes" sent are just the CIM's RAM workspace area contents, the d-th byte in the 1-byte case, or the first d bytes in the d-byte case. The order of the bytes in the workspace area are given below. Some additional details of certain bytes are given in Sec. IIIE.

**STATUS**

**BYTE 0:** Timer Status Byte  
1: CIM Status Word (High Byte)  
2: CIM Status Word (Low Byte)  
3: COM Rx Status  
4: COM Tx Status  
5: Rx Error Count  
6: Tx Error Count  
7: Timer Interrupt Bits Enabled by CIM  
8: Timer Registers Enabled for LDC Messages  
9: End-of-Period Timer Status  
10: Contents of Timer IMR register  
11: Contents of Timer AUX register  
12: Flag to queue Timer Services (Level 1)  
13: Contents of CIM front-panel Switches  
14: Contents of Display  
15: Panel Switch Options being set up  
16: Panel Switch Options In Force.

"..see software listing for rest of Global Area."
III. SPECIAL STATUS FORMATS.

1. TIMER STATUS:

```
0  6  5  4  3  2  1  0
| R0  | R4  | R3  | R2  | R1  | R0* | RC  | EF  |
```

Note that all bits in the Timer Status Byte (TSB) are cleared at the start of each basic timing cycle; thus within each cycle the bits set in the byte indicate all interrupt lines which have had any activation(s) since the start of the current cycle. It is up to the LDC to keep track of the sequence and significance of these interrupt events, by comparing TSB's received in successive Timer Status messages.

2. CIM STATUS WORD:

```
Hi Byte:  15  14  13  12  11  10  9  8
          | ENAB | DSR | RTS | CD  | TD  | RD  | TR  | RR  |

--if set: CIM  Modem Request Carr.  Tx-DMA  Rx-DMA  Tx-Req.  Rx-Req
Enabled  OK  to-Send -Det.  Busy  Busy  Active  Active

Lo Byte:  7  6  5  4  3  2  1  0

          | Timer| P/I0 | RX-DMA| TX-DMA| Modem| Service Level No. |

--if set: ...Service Levels 1-5 Requested...
```

3. COM STATUS BYTES:

```
Rx Byte:  7  6  5  4  3  2  1  0
          | RERR | C   | B   | A   | ROR | RAB/GA | REOM | RSOM |

Tx Byte:  7  6  5  4  3  2  1  0
          | RERR | 0   | 0   | 0   | TXGA| TXAB  | TEOH | TSOH |
```
IIIIF. CIM ADDRESSING SCHEME.

1. RX/TX PIA.
Addresses C000-3:

PA0: Prog. Reset
PA1: Rx DMA On
PA2: Tx DMA On
PA3: Rx EOB set
PA4: CLR Linkabit Err-rate Count
PA5: SEL-CLR Rx EOB
PA6: , , Tx , ,
PA7: Tx GO

CA2: Rx DMA ATTN

PB0: Tx WDCNT Zero (self-test) or(GEC): MPFF
PB1: Rx DMA REQ (H/W) or(GEC): CLR MPFF
PB2: Tx , , , or(GEC): Store Active
PB3: Tx DMA ATTN or(GEC): Store Interlock
PB4: CD from E.S.
PB5: RTS to , ,
PB6: DSR from , ,
PB7: Linkabit Error Counter Overflow

CB2: Rx DMA STATUS

Addresses C004-7: spare.

2. MAINTENANCE PIA.
Addresses C008-B:

PA0: Maintenance Select
PA1: Rx CRC (CERN-Loeche)
PA2: Tx CRC , ,
PA3: LESP , ,
PA4: Slave/Master Select (Slave=1)
PA5: spare
PA6: 70 MHz OSC ON (self-test)
PA7: Error-Insert (Invert Tx bits)

PB0-7: Hex Display/Switches

CB2: Select Hex Switches.

3. LINKABIT ERROR-RATE COUNTER ADDRESS:
C00C: , , , , , ,
Addresses C00D-F: spare.

4. DMA/COM CONTROL.
Addresses C010-3F:

10-11: Rx Block Addr. (MS-LS)
18-19: Tx , , , ,
20-21: Rx Word Count , ,
28-29: Tx , , , ,
30-37: Rx COM Reg. Select
38-3F: Tx , , , ,

5. PIO PIA's.
Addresses C100-B:

0-3: PIA0
PA0-7: INL0-7  
PB0-7: INM0-7

CA1: Data Transmitted (PDP only)  
CA2: Input Requested (,,)

4-7: PIA1:

PA0-7: OUTL0-7  
PB0-7: OUTM0-7

CA1: Output Requested (PDP only)  
CA2: Output Accepted (,,)

8-B: PIA2: (not PDP case)

PA0: Output Active  
PA1: Input,,  
PA2-7: ..spare..

PB0: Output Request  
PB1: Input,,  
PB2: Output Status  
PB3: Input,,  
PB4-7: ..spare..

CA1: Data Ready  
CB1: Data Transmitted

6. TIMER.  
Addresses C200-3A:

00-01: CR0 (Timer Chip 0)  
02-03: RC (MS-LS)  
04-05: W1,,  
06-07: W2,,  
08-09: CR1  
0A-0B: R0*,,  
0C-0D: R0,,  
0E-0F: R1,,  
10-11: CR2  
12-13: R2,,  
14-15: R3,,  
16-17: R4,,  
18-19, 20-21, 28-29, 30-31: ..spare CR3-6..  
1A-1B, ..,. .,. 36-37: .., R5-R16..  
39: IMR Reg.  
3A: IRR Reg.

7. MEMORY.  
Addresses 0000-1FFF: 8KB RAM for workspace + RAM code.  
Addresses A000-A0FF: 256B RAM for MPU stack + Interrupt vectors.  
Addresses E000-F7FF: 6KB EPROM for CIM EPROM code (6 2708's).  
Addresses F800-FBFF: 1KB EPROM for DLL EPROM code (1 2703).  
Addresses FC00-FFFF: 1KB EPROM for MINIBUG + Interrupt vectors.

8. ACIA's.  
Addresses 8008-8009: ACIA (Terminal) -- STATUS & DATA (resp.)  
Addresses 800A-800B: ACIA (Host) --,,,
IV. HARDWARE CHARACTERISTICS.

IV.A. MECHANICAL.

A CERN-developed DEC-based card/crate system is used for the CIM; a basic crate has 32 dual connector slots and each connector has 36 pins (DEC #9111). The crate is of 19-inch type and does not include power supplies; power is applied at the side of the crate and is distributed using metal bars across the crate soldered to the appropriate pins. A printed-circuit motherboard has been developed connecting all bus signals to all connector slots. The back side of the crate can also be wire-wrapped.

A smaller-size crate than this maximum is actually used for the CIM, with a capacity of 24 dual slots; it is believed that this will suffice for any CIM configuration, but if needed an additional crate could be connected, thus expanding the number of connector slots to 56; the second crate would connect to the bus via a bus-extender card buffering the bus signals.

A standard CERN wire-wrap card (DD 4939) connects into the crate; it has a capacity of 70 16-pin DIL integrated circuit packages and it allows a circuit layout to include IC's with 8, 14, 16, 20, 28, or 40 pins in a mixed fashion. A software package exists at CERN to prepare paper tape, used in turn to control a semi-automatic wire-wrap machine. Wire-wrap techniques are used normally only for prototypes, as such cards need 2 slots instead of 1 to physically accommodate the wire-wrap pins. However, the technique has been found to be so reliable that for CIM's without space problems many of the cards can be of wire-wrap type. Cards common to all CIM's (e.g. microprocessor, RAM, ACIA) are manufactured by printed-circuit techniques.

The CIM logic is laid out on circuit boards in a modular fashion that allows for easy reconfiguration when adapting the CIM to each different LDC. (See Figure 2). A list of circuit boards follows:

1. M6800 CPU/EPROM BOARD.
   This board contains the M6800 MPU, the clock circuitry for both MPU and Tx COM chip, the Priority Interrupt Controller (PIC), and 256 bytes of RAM for the stack. There are eight (8) sockets for 2708 type EPROM's, each 1K byte in capacity. The board is of printed circuit construction.

2. M6800 RAM MEMORY BOARD.
   This printed circuit board contains 8K bytes of 2114 type static RAM and address decoding logic allowing easy reconfiguration. The board can be tested using the MINIBUG 'W' command (after setting up start and end test addresses in locations #A002-3 and #A004-5).

3. ACIA BOARD.
   This printed circuit board supports two asynchronous CCITT V24 I/O channels. It uses the Motorola M6850 Asynchronous Communications Interface Adapter (ACIA) and has common address decoding logic for both channels. The required four addresses must be contiguous as follows:

   Address XXXX    ACIA 1 -- STATUS
   ,, XXXX+1      ,, -- DATA
   ,, XXXX+2      ACIA 2 -- STATUS
   ,, XXXX+3      ,, -- DATA

   Address XXXX is switch-selectable, using on-board switches for the address-selection. There is also an on-board baud-rate switch-selection feature allowing 14 different I/O speeds as shown below. Note that this baud-rate selection is common to both channels and to both transmit and receive directions.
IC 15 (see drawing DD4984-1-2)
PIN 5 6 7 8 SPEED (bps)
0 0 0 0 --
0 0 0 1 --
0 0 1 0 50
0 0 1 1 75
0 1 0 0 134.5
0 1 0 1 200
0 1 1 0 600
0 1 1 1 2400
1 0 0 0 9600
1 0 0 1 4800
1 0 1 0 1800
1 0 1 1 1200
1 1 0 0 2400
1 1 0 1 300
1 1 1 0 150
1 1 1 1 110

4. TIMER BOARD.
The timer board handles CIM timing functions. It contains three (3) Motorola M6840 Programmable Timers, Interrupt Mask and Interrupt Request Registers, Slave/Master logic and bus address-decoding logic. More M6840 timers could in principle be accommodated. A Preamble Counter is also included which provides an interrupt ("End-Flags") to the MPU after transmission of 80 HDLC frame flags. This counter is reset at the end of the Transmit Window (R2). Reprogramming of the value of this counter can be achieved by a simple wiring change.
The timer board provides circuitry for operating in either Master or Slave Mode. When in Slave Mode the timer R0 will synchronize on master reference bursts by means of the logic on drawing DD4954-12-2. Since all counters run in continuous mode, at the end of their respective periods a gating signal is used to disable each counter until the next End-Cycle (R0); each counter's end of period also causes a bit to be set in the Interrupt Request Register (IRR), causing an MPU vectored interrupt via the PIC provided that the corresponding bit in the Interrupt Mask Register (IMR) has been set. As reset of bits in the IRR (performed by a read of the IRR) is not selective, serviced interrupts are masked off for the rest of the R0 cycle by using the IMR and the IRR is only reset at End-Cycle time.
The board is of wire-wrap construction.

5. TX DMA/COM BOARD.
The TX board handles communication between the LDC's DMA channel (or memory port) and the satellite earth station. It basically gets 16-bit words from the LDC memory, splits them into two bytes each, and stores them in a 32-byte long FIFO; then the COM5025 chip, which handles the HDLC protocol, requests a byte at a time from the FIFO and transmits it in an HDLC serial frame to the earth station. The COM chip has connections via a bus switch to the MPU bus for control/status purposes, but during a Tx DMA transfer (signal TXDMAON true) the COM chip is isolated from the MPU bus and gets data from the FIFO without MPU intervention. COM initialization and start-of-frame are controlled by the MPU. The actual time of transmission is determined by a signal TX-WINDOW derived from the Timer Board; after a preamble of 80 HDLC flags (also determined by the Timer Board), the MPU commands the COM to start sending data. The first byte (which for timing reasons has been pre-loaded into the COM data register) is then sent and the transfer proceeds with the COM chip isolated until the last byte from the FIFO is being loaded. This byte has a "tag bit" set, causing logic on the board to load a Transmit-End-of-Message (TEOM) command into the COM which in turn sends the frame CRC bytes followed by trailing flag bytes until the end of TX-WINDOW. After sending the two CRC bytes, the board also generates an IR0 interrupt to the MPU to signal End-of-Tx-Block, allowing the MPU to drop TXDMAON, thereby
reconnecting the COM to the MPU bus and permitting COM status to be read and forwarded to the LDC.

For the case (e.g. GEC 4080) where LDC DMA logic is absent, the TX board also contains Tx Block Address/Word Count Registers which are incremented/decremented during transfers, plus logic to set the tag bit of the final FIFO byte at end-of-block. These DMA registers are MPU-accessible as follows: Block-Address 16-bits write-only; Word-Count 8-bits read-write.

The board is of wire-wrap construction.

6. RX DMA/COM BOARD 1.

The RX DMA board 1 handles reception of HDLC frames from the satellite earth station, performs the necessary operations to extract and check validity of the data, and transfers the data to the LDC via the RX DMA channel. The HDLC serial data are received together with the receive clock by the RX COM chip, which performs frame detection, bit extraction, and CRC and overrun checking; data are staticized and passed on bytewise to a 32 byte long receive FIFO, after which assembly into 16-bit words takes place. Control logic then transfers the words to the LDC DMA channel. A ninth "tag" bit is appended to each data byte when it is written into the FIFO, but only if the COM signal RX-ACTIVE is present. As this signal is present for the whole HDLC frame except during the last byte, the final byte appears on the FIFO output without a tag bit, and this is used to send an End-of-Receive-Block IRQ interrupt to the MPU; COM status can now be read by the MPU and further control action taken. (Similarly to the Tx case, the RX COM is disconnected from the MPU bus during RX DMA activity and reconnected afterwards by the signal RXDMAON). The Rx IRQ interrupt is distinguished from the Tx IRQ interrupt by setting a PIA bit; other bits on this "RX/TX" PIA on RX board 1 are used for earth station control and both Rx and Tx DMA control.

The board is of wire-wrap construction.

7. RX DMA/COM BOARD 2.

The RX DMA board 2 has three main functions:
-- Bus address decoding for TX, RX1 and RX2 boards.
-- Control of hexadecimal switches/display on CIM front panel.
-- Holding DMA Block-address/Word-count registers for the GEC 4080.

The bus address decoding is carried out centrally because all three DMA boards have adjacent addresses; bits 15-6 are decoded vs. switch-settings and are common for each board, whereas bits 5-2 provide the board addresses and bits 1-0 are used for subaddresses e.g. PIA register addresses.

The two-digit hex display and thumbwheel switches on the CIM panel are connected to a "Maintenance" PIA (data registers) and control bit CB2 is used to select switch/display mode for program control of these two devices, since both use the same data lines.

For the case (e.g. GEC 4080) where LDC DMA logic is absent, the RX board 2 also contains Rx Block Address/Word Count Registers which are incremented/decremented during transfers, plus logic to set tag bits of all FIFO bytes except at end-of-block. These DMA registers are MPU-accessible as follows: Block-Address 16-bits write-only; Word-Count 8-bits read-write.

The board is of wire-wrap construction.

8. PIO BOARD.

The Programmed I/O (PIO) board provides a full-duplex 16-bit data channel between the CIM and the LDC. It contains a PIA for LDC Input, another PIA for LDC Output, and in cases (e.g. NORD, GEC) where extra control signals are used, a third PIA is added. The board also holds bus address decoding logic plus drivers/receivers for the LDC connection.

The board is of wire-wrap construction.
**IVC. ELECTRICAL.**

A bus standard has been specified in collaboration with Rutherford, the list of signals and pin allocations is given in Sec. IVD. The bus is TTL-compatible and loading of signals should not exceed a unit Schottky-TTL. The specification is rather unrestrictive: the bus includes enough address and data lines to handle 16-bit as well as 8-bit microprocessors; the data path is 16-bits wide (bidirectional) and the address field 24-bits wide. There is provision for eight additional bus signals and nine signals for card-card communication; the latter can also be achieved via Berg connectors at the back side of the cards, possibly using flat cable.

**IVC. PACKAGING.**

The crate is fitted into a 19-inch wide box with the required power supplies. Important CIM hardware conditions are displayed on a front panel, and two hex digits of input (switches) and output (display) are also provided for CIM control. All external connections are handled via rear panel connectors.

**IVD. CIM BUS STANDARD.**

<table>
<thead>
<tr>
<th>UPPER CONNECTOR</th>
<th>LOWER CONNECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 Spare</td>
<td>A1 A16</td>
</tr>
<tr>
<td>B1 PHI-1</td>
<td>B1 A17</td>
</tr>
<tr>
<td>C1 INQ</td>
<td>C1 A19</td>
</tr>
<tr>
<td>D1 VMA</td>
<td>D1 A20</td>
</tr>
<tr>
<td>E1 D0</td>
<td>E1 A22</td>
</tr>
<tr>
<td>F1 D2</td>
<td>F1 D8</td>
</tr>
<tr>
<td>H1 D4</td>
<td>H1 D10</td>
</tr>
<tr>
<td>J1 D6</td>
<td>J1 D12</td>
</tr>
<tr>
<td>K1 A0</td>
<td>K1 D14</td>
</tr>
<tr>
<td>L1 A2</td>
<td>L1 12kHz</td>
</tr>
<tr>
<td>M1 A4</td>
<td>M1 2 MHz</td>
</tr>
<tr>
<td>N1 A6</td>
<td>N1 1 MHz</td>
</tr>
<tr>
<td>P1 R8</td>
<td>P1 0.5MHz</td>
</tr>
<tr>
<td>R1 A10</td>
<td>R1 Note 1</td>
</tr>
<tr>
<td>S1 A12</td>
<td>S1 0.5V</td>
</tr>
<tr>
<td>T1 GND</td>
<td>T1 GND</td>
</tr>
<tr>
<td>U1 A14</td>
<td>U1 Note 2</td>
</tr>
<tr>
<td>V1 R/W</td>
<td>V1 TSO</td>
</tr>
</tbody>
</table>

Case: NORD10 PDP11 GEC

Note 1: LOWER CONNECTOR SIGNAL R1: TxDMA Rq Go-Tx ---

Note 2:  , U1: Tx Abort AttnTx RxDING
V. CIM OPERATING PROCEDURES.

VL. EPROM/RAM ENTRY POINTS.

1. POWERON (MINIBUG):
   Upon power-on/reset, software control of the CIM is taken by the
   Motorola MINIBUG monitor, which initializes the ACIA of the terminal
   port and prints an asterisk prompt '*' on the screen. Basic MINIBUG
   terminal commands are:

   M hhhh  -- Display memory location hhhh (hex)
   G hhhh  -- Go to (execute from) location hhhh.

2. EPROM DEADSTART (#E000/#E100):
   To initialize (deadstart) the CIM, type: G E000 or: G E100
   as the EPROM deadstart entry point is at #E100 and #E000 contains a
   jump to #E100. This initializes the CIM hardware and also sets up
   all necessary RAM locations with suitable default values.

3. RAM DEADSTART (#0100):
   If downline loading of the RAM software has previously been done,
   then to deadstart the CIM with this software, type: G 0100
   as the RAM deadstart entry point is at #0100. This initializes the
   CIM hardware and also sets up all necessary RAM locations with suit-
   able default values. Control is taken away from EPROM software if this
   has previously been running.

4. DOWNLINE RAM LOADING:
   If alternatively it is desired to downline load and execute out of
   RAM instead of EPROM, then from MINIBUG type the command:

   G FA00  -- ACIA Down-Line Loader
   or:
   G F900  -- PIO Down-Line Loader (manual entry).

   -- ACIA DLL (#FA00):
   In the ACIA DLL case, the CIM will permit connection to be made to a
   remote host (which need not be the LDC) via the second ACIA port,
   and the terminal on the first ACIA port. Keyboard entries are made to
   the host in the normal way to perform login, prepare the load file
   and so on. RAM loading is made typically with a LIST command (of an
   appropriate load module in standard Motorola format) followed by the
   keyboard character CONTROL-B in place of carriage-return. After the
   load is over, control returns to MINIBUG; loading errors are flagged
   by printing a '?' on the screen before the MINIBUG return. After a
   successful RAM load, the command:

   G 0100

   will cause the CIM to be deadstarted with the RAM software; for the
   first entry (only) after load, the command G 0000 will also do this.

   -- PIO DLL (#F900):
   In the PIO DLL case, the CIM will display 'OF' on the panel display,
   to indicate it is in "Fill" mode, and wait for the LDC to perform
   the load via the PIO channel. In this case, after a successful load,
   control is immediately given to the RAM deadstart address, but if
   a loading error occurs control is given to CIM EPROM software (if
   fitted) else to MINIBUG. In the error case, the LDC receives no CIM
   message after loading, whereas after a good load and RAM execution
   has begun, a PIO restart-acknowledge message is issued to the LDC
   by the CIM.
5. EPROM/RAM RESTARTS (#0000/NMI):

After deadstarting, restart entries are made either by pressing
the Non-Maskable-Interrupt (NMI) button on the front panel, or by
the terminal command:

```
G 0000
```
as the NMI RAM vector location, and the RAM address #0000, are both
rewritten after deadstart so as to contain jumps to the restart (not
the deadstart) address of the EPROM or RAM software last deadstarted.

6. CIM WITHOUT A CONTROL TERMINAL:

A special version of MINIBUG can be fitted to avoid the need for
terminal 'G hhhh' commands to deadstart the CIM. At reset time, a
test is made for presence of a control terminal (by checking for the
presence/absence of the CD and CTS signals on the terminal ACIA). If
the terminal is found to be absent (or off) at poweron/reset time,
then MINIBUG presets the NMI RAM vector location so that THE FIRST
TIME THE NMI BUTTON IS PRESSED AFTER POWERON/RESET, either an EPROM
deadstart is performed (if CIM EPROM software is fitted) or a wait
is made for Remote Fill of RAM by the LDC via the PIO channel (while
displaying 'OF' on the CIM panel display).

All NMI operations after the first cause restarts, not deadstarts,
as described in the preceding paragraph.

7. EPROM CHECKSUMMING:

By typing the command:

```
G FBAD
```
a separate checksum is calculated for each of the 8 1K EPROMS which
can be mounted on the CIM MPU board, followed by an overall checksum.
These 9 results are printed on the terminal and also displayed (with
a 5-second pause between each of them, to allow each to be noted) on
the CIM front-panel display; each result is a single byte. If the
byte is 'FF' this indicates an empty EPROM socket. The 'FBAD' command
is a mnemonic for "Find BAD" EPROM's.
The CIM has a front-panel switch containing two hex digits, and a corresponding display, providing elementary manual and visual control information to be input/output between CIM and operator. In normal operation the display simply REFLECTS the switch setting (with the exception that a setting of '00' is reflected with 'FF', because the display '00' can be produced by a CIM malfunction). It is recommended that normally the switches are left at '00' or 'FF', so that the CIM displays 'FF'. Then different values displayed indicate ERRORS, as listed in paragraph 1 below. Once an error is noted, the display can be reset by making any switch change, unless the error is persistent in which case it will immediately be re-displayed.

The switches are also used to set up various operating options in the manner described in paragraph 2.

1. CIM ERROR DISPLAYS:
   All errors except #06 are also sent as single-word PIO messages to the LDC.
   Error Number  Error Condition  
   #01  PIO Input: nonexistent opcode.  (from LDC)
   #02  , ,  wrong byte count in message.
   #03  , ,  missing terminator word.
   #04  , ,  extra  , ,  .
   #05  , ,  premature termination by LDC.
   #06  PIO Output: premature termination (or ( to LDC)  no response at all) by LDC.
   #10  CIM Enable: DSR is down at enable time.
   #11  , ,  enable block has format error.
   #12  , ,  , ,  , ,  bad type-nibble.
   #13  , ,  , ,  , ,  master/slave bit in disagreement with CIM panel switch.
   #20  DMA Error: LDC has DMA non-operational.
   #21  , ,  DMA request while request still active.
   #22  , ,  DMA req. block has wrong no. of params.
   #23  , ,  DMA s/w req. invalid as h/w option set.
   #31  Status Req: Status request of multi-word format.
   #41  Timer Req.: Attempt to set invalid interrupt bits.
   #42  , ,  Req. block with wrong no. of params.
   #0A  spare
   #05  Burst Err.: missing/late/absent Master Burst.
   #0C  CD Error: Carrier-Detect with no Rx DMA active.
   #0D  DSR Error: Data-Set-Ready dropped while operating.
   #0E  Error DLL: Error condition during PIO Remote-Fill.
   #0F  Filling: (not an error)--in Remote-Fill Status.

2. SWITCH-SETTING OF CIM OPTIONS:
   Special switch manipulations are needed to set up CIM operating options. The CIM has 1 byte (8 bits) of switchable options, whose individual bits are as follows:
   (Lower nibble)--Bit 0: If set, inhibits PIO Outputs to LDC.
   --Bit 1: , ,  , ,  PIO Inputs from .
   --Bit 2: , ,  adds terminator word to Outputs.
   --Bit 3: spare
   (Upper nibble)--Bit 4: If set, inverts Tx Data bits (Error-insert)
   --Bit 5: , ,  activates CIM self-test.
   --Bit 6: , ,  disables modem checks (CD, DSR, RTS).
   --Bit 7: , ,  selects Maintenance Mode.

The option byte in force at any time can be DISPLAYED by the switch setting 'DD'. To CHANGE the option byte, the Lower and Upper nibbles of the byte must first be set up as follows:
Switch setting 'Aa' PRE-SETS LowerNibble to hex value a.
'Bb', UpperNibble b.
'CC' CHECKS preset nibbles by displaying 'ba'.

Finally:
Switch setting 'EE' ENTERS preset into operating nibbles.
(and displays the new operating byte).
VC. RAM WORKSPACE USE.

By using the MINIBUG 'M' command, CIM memory locations may be examined and altered if desired. The lower 256 RAM locations have special importance as they are used as a global workspace area. The locations are used as follows:

- #0000-0002 Jump instruction to current RESTART address.
- #0003-000F Private for use of DLL routines.
- #0010-0020 CIM 1-byte status area (see IIID above).
- #0021-00FF CIM global status area (remainder)...

Locations of frequent interest are, for example:

- #0015-0016 RX/Tx Error Counts
- #0020 Switch Option Byte in force.
- #0021-0029.. RX/Tx DMA EOB Status Blocks:
- #0021 RX PIARTDB Status (latest)
- #0022 RX COM ",",
- #0023 Rx Linkabit Errors ",",
- #0026 Tx PIARTDB Status ",",
- #0027 Tx COM ",",
- #0030-0034 R0,R1,R2,R3,R4 Interrupt counts
- #003B-003D Block Counts: (Total, Rx and Tx).
- #0050-005E Enable Block in force.
- #005B-005C Value of R0 Timer Latch (in MSEC units)
- #006D-0074 ,,, R1-R4 Latches..
- #007D-0080 RX DMA Wd Cnt & Block Addr Requested (GEC only)
- #0081-0084 Tx ",",

FIG. 2. COMMUNICATIONS INTERFACE MODULE — BLOCK DIAGRAM HARDWARE
Relative to satellite:

**Fig. 3 - Basic Time Frame & Format**
Relative to E.S.:

R0 R1 EF R2 (R4) (R3) R0...

TW1 TW2 (optional)

→ TRANSMIT WINDOW ←

OVERALL BASIC TIMING CYCLE

NOTE: EF marker occurs a fixed number of BIT-TIMES after TW1

FIG. 4 — CIM PROGRAMMABLE TIMERS