Published: 06/30/67
(Supersedes: BC.4.01, 10/21/66;
BC.4.01, 04/14/67)

## Identification

System Initialization and Bootload
A. Bensoussan

## Purpose

This document describes the system initialization and bootload procedure for GE 645 system.

"Initialize" and "Initialize and Bootload" functions are provided using two distinct pushbuttons.

The Initialize function will place all of the active and passive units of the system in an initial state. The peripherals are not affected.

The bootload function initializes the system and then loads a program, stored in the GIOC hardware, into core memory. The processor executes the bootload program, which is designed to read a tape written in the Multics standard format; it reads the beginning of the tape and transfers to a conventional location.

## Initialize

Pressing the Initialize Button causes the following sequence:

1. A logic signal is sent from GIOC to all Memory Controllers

2. The Memory Controllers will
   Initialize the controller
   Send an initial signal to all ports

3. All active units will reset their control logic to an initial state

   Memory Controllers: All Interrupt Cells reset
   No memory cycle or request existing
   The masks are set so that all interrupts are allowed.

   Processors : All fault conditions reset
   DIS instruction in CPU (wait for interrupt)

| | |
|---|---|
| GIOC | : All adapters reset<br>No channel service requests existing |
| Drum | : Controller in disconnected state |

Bootload

The bootload is activated by a single pushbutton. One of the 2 GIOC peripheral adapter channels implemented within the bootload as the loader program channel must have been preselected by a panel switch in the bootload GIOC controller cabinet.

The GIOC contains a Bootload channel with a fixed store memory from which 64 words are transferred into selected locations in core. The bootload channel is part of the controller and operates as a direct channel; no channel number is assigned and no command can be issued to it; the channel is only activated by depressing the bootload button.

Before pushing the bootload button.

A certain number of switches have to be set before pushing the bootload button, so that the hardware assumptions made by the bootload program are respected.

1. GIOC base address Go

The GIOC has an 18 bit panel switch to determine the base address Go of the GIOC mailboxes. Setting this address defines the memory controller that is "control memory" for this GIOC. Let m be the number associated with this memory controller (m = 1,2,3,4,5,6,7 or 8).

Because of the mailbox address formation mechanism, this address must be zero modulo a power of two which is greater than twice the highest assigned channel number: if n is the highest assigned channel number, then Go must be zero modulo a power of two which is greater than 2 x n.

2. Control processor

There is an 8-position panel switch in each memory controller, to define what processor will be designated as control processor for this memory controller. Let p be the value of this switch in memory controller (m)

3. Processor base address Po

The 18 bit panel switch in processor p is set with the value So that defines the base address Po of the fault vector the following way: Po = So x 64. Po has to be zero modulo 1024. This assumption is made in order to be able to determine the base address of the fault vector without having to know the port number through which processor p receives the interrupt from memory controller m.

4. Bootload base address Bo

This address is set by an 18 bit panel switch on the GIOC. It must be equal to the location to which the control processor will be forced when it receives the interrupt at the end of the loading.

$$Bo = Po + k \times 64$$

where k is the port number through which the processor p is connected to the memory controller m.

5. Bootload channel selection

A separate switch within the GIOC will select one of the 2 peripheral adapter channels implemented within the bootload as the loader program channel.

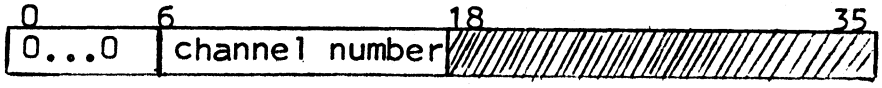
This switch is read by the GIOC when the diodes are transferred into core; its value is 1 or 2. If the value is 2, then the GIOC "OR"s a one in word Go + 17, bit 17.

When the switch is set to 1, the bootload program described below assumes that the tape to be read is on channel 31. When the switch is set to 2, the bootload program reads the channel number from the processor switches, the value of which is given by the operator on the processor, before pushing the bootload button.

The processor switches are a 36 bit word; the channel number is in bits 6-17. Bits 18-35 are not used by the bootload program.

The channel number must be the channel number of the LPW, that is bit 17 must be zero.

Processor Switches:



6. Interrupt cell assignment

Each status channel used by a GIOC is assigned one of the 32 interrupt cells in the memory controller (m) which contains the base address Go of the GIOC mailboxes. The assignment of a status channel to a particular execute interrupt cell is accomplished by a patch plug. Status channel 0 should always be assigned a higher priority than any other status channel.

The bootload assumes that

GIOC status channel 0 is assigned cell i ($0 \leqslant i \leqslant 3$)

GIOC status channel 1 is assigned cell j ($4 \leqslant j \leqslant 11$)

7. Channel number of the peripheral device

The bootload assumes that the device is the tape handler on channel #30, device #0, unless the bootload channel switch is set to 2 (see bootload channel selection).

Pushing the bootload button

When the bootload button is depressed

The Initialize signal is activated, causing all active and passive units to be initialized.

Then the sequence of events described below occurs.

1. 64 words are loaded from GIOC into core

   a. Interrupt vector

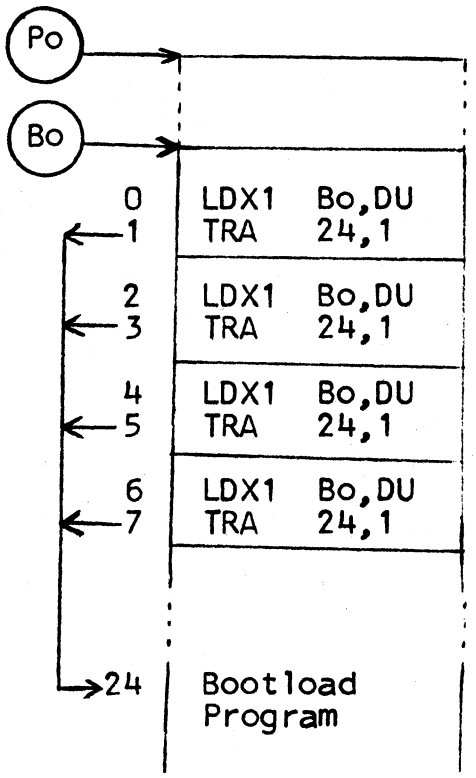
      8 words are loaded at Bo to Bo + 7, providing four interrupt entries associated with cells 0,1,2,3. All of them transfer to the beginning of the bootload program (Bo + 24)

   b. Bootload program

      The bootload program is loaded starting at Bo + 24

c. Mailboxes

Control words are stored in the mailboxes. All addresses contained in these control words are relative to Go; they will be adjusted to absolute addresses by the bootload program before the Connect GIOC is issued.




2. Interrupt 0

When the 64 words are loaded, GIOC sends an SXC (Set Execute Interrupt Cell) to the Memory Controller (m) containing the base address Go. The interrupt cell i, associated with status channel #0 is set (status channel 0 has been assigned interrupt cell 0,1,2 or 3). The Memory Controller (m) interrupts its control processor (p); the interrupt vector associated with this processor and this Memory Controller is located at Bo, where 4 interrupt entries have been stored. Control goes to the beginning of bootload program, at START0 (Bo+24).

3. Bootload program execution: START0

Tests the value of the bootload channel selection switch, using the bit 17 of word Go+17. If this bit is ONE, the CCW is read from the 36 switches and the channel number is stored in Go+CCW bits 0-17.

Sets interrupt vector entries 0-11 so that they will transfer to START1 after the tape label is skipped.

Initializes control words in the mailboxes for the first connect.

Sends a connect to GIOC through connect channel specified by the Connect Operand Word (COW) located at Go+0, for skipping the tape label until a physical EOF mark is encountered.

Waits for interrupt 1 which will occur when the physical EOF mark is found.

4. Interrupt 1

When the tape label is skipped, status channel 1 is activated and interrupt cell j is set ON ($4 \leq j \leq 11$). The interrupt vector is entered that transfers to START1.

5. Bootload program execution: START1

Sets interrupt vector entries 0-11 so that they will transfer to START2 after the next physical record of the MST is read

Initializes control words in the mailboxes for the second connect.

Sends a connect to GIOC for reading one physical record on the tape into location Lo=Go+1024

Waits for interrupt2 which will occur when the physical record is read

6. Interrupt 2

When the physical record is read, status channel 1 is activated, and the interrupt vector is entered that transfers to START2.

7. Bootload program execution: START2

If major status is ready, transfers to Lo+32. The constant 32 has been chosen to allow flexibility in the logical format of the tape read by the bootload program.

If not, stop.

Bootload program code and Mailboxes

The bootload program code is shown in Figure 1. The contents of the mailboxes is shown in Figure 2. In both figures 1 and 2,

(x) stands for zero when the 64 words are in the diodes and stands for the value x when the 64 words are in core.

[ x ] stands for zero when the 64 words are in the diodes and stands for x during the connect operation. The value x is stored by the bootload program.

1. Program Code

This program relocates all internal references by index register 1, which is loaded upon entry to contain the base address of the bootload (Bo). It relocates all external references by index register 0 which is loaded in the program, to contain the base address of the mailboxes (Go). Thus, this program written for loading at zero will operate anywhere, the mailboxes being anywhere in memory.

2. Mailboxes

Figure 2 shows the Mailboxes as they are in the diodes. All addresses of control words (except the absolute address 10 in word 1) are zero in the diodes and are stored by the bootload program before the connect is issued.

COW: Has to be at Go+0
Bits 33-35 specify the Memory Controller port number to which the GIOC is connected. This port number is stored in the diodes. Bits 31-32 are zero and specify the connect channel #8.

SCW0: Has to be at Go+1
It is the SCW used when status channel #0 is activated, after the diodes have been transferred into core.

The status is stored in ABSOLUTE location 10.

SCW1: Has to be at Go+3
It is the SCW used when status channel #1 is activated. The address is zero in the diodes.

CCW: Channel Command Word.
The CCW is at Go+10. Figure 2 shows the CCW as it is stored in the diodes, that is:

The LPW channel number is 30; bit 35=1 indicates that the multiple physical record (MPR) mode is set; bits 32-33=10 set the active mode. The device number (bits 18-23) and the operation code (bits 24-29) are not used in the CCW when the multiple physical record mode is set. They are found in the command DCW (DCW1).

MPR mode has to be set because the command SKIP does not cause any data transfer; if the mode was single physical record (Bit 35=0), no LPW list would be activated; as a consequence no status channel could be specified and no interrupt would occur after completion of the command SKIP.

Using the MPR mode, the LPW is activated even if there is no data transfer, and we can specify the status channel by a pointer in the command DCW (bits 3-5 of DCW1).

Before the second connect is issued, the bootload program sets bits 18-23 of the CCW with the value 000510 (octal):

| | |
|---|---|
| Bits 18-23 = 000 000 | define the device #0 |
| Bits 24-29 = 000 101 | define the operation code READ |
| Bits 30-31 = 00 | they must be zero |
| Bits 32-33 = 10 | set the active mode |
| Bit 34 = 0 | it must be zero |
| Bit 35 = 0 | set the single physical record mode. |

CPW: Has to be in Go+16
The address contained in Go+17 is zero in the diodes. GIOC will "OR" a one in bit 17 of word Go+17 if the bootload channel switch is set to 2.

DCW1: It is a command DCW (bits 0-2 = 4). The device number (bits 18-23) is 0: the operation code (bits 24-29) is 45 octal = SKIP. When SKIP is completed, the EOF status forces termination; the terminate signal occurs and the status word is stored using status channel #1 (bits 3-5 = 001 in DCW1). Correct operation depends on EOF forcing termination.

DCW2: It is a data transfer DCW (bits 0-2 = 0) for use during the READ operation. If a terminate signal occurs, status channel 1 is activated. Bit 17 = 1 which indicates the last DCW. Exhaust, external and internal signals should not occur, and if they do, it is an error.

TRAP: Words TRAP and TRAP+1 contain two instructions:

```
EAX7      3,7
TRA  START1-3,7
```

These instructions are stored by the bootload program into each pair of the interrupt vector. Index register 7 is initialized with the value Bo. When the interrupt occurs after the label is skipped the interrupt vector is entered. The first instruction causes register 7 to be incremented by 3 (X7=Bo+3); the second instruction transfers to START1.

When the next interrupt occurs after the physical record is read, the first instruction sets register 7 to the value Bo+6 and the second instruction transfers to START1+3 which is START2.

TEMP: Words TEMP and TEMP+1 are used to manufacture the information that has to be stored in the LPW. When the program has computed the address of the LPW, it stores the contents of TEMP and TEMP+1 in LPW and LPW+1.

LPW: The LPW is not stored in the diodes since its location is not known (the channel number specified by the CCW is 30 in the diodes, but it can be changed by the operator).

Before the first connect is issued for skipping the label, LPW contains zero, LPW+1 points to DCW1 and the tally is 2.

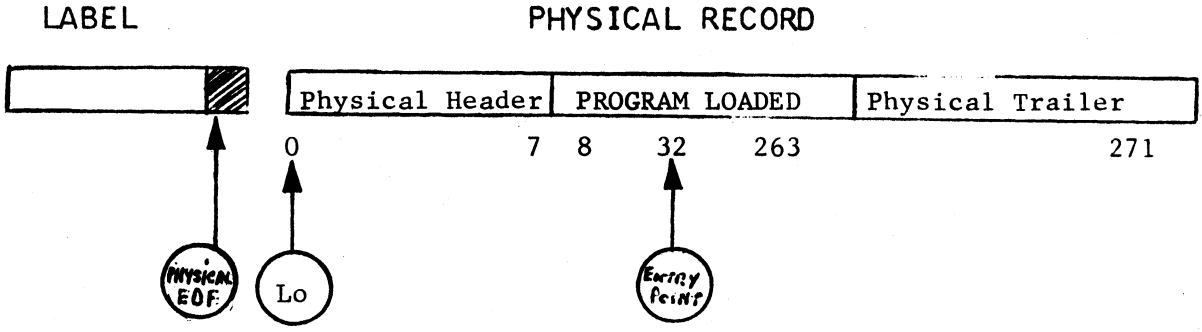
When the first connect is given, the multiple physical record mode is set (Bit 35=1 in CCW), the LPW is activated, the command DCW (DCW1) is executed which initiates a SKIP operation. When SKIP is completed, the EOF status will force termination and leave the LPW pointing at DCW2.

Before the second connect is issued, the CCW is set to contain the device #0, the operation code READ (05 octal) and the single physical record mode bit (Bit 35=0). The LPW is activated which still points to DCW2; the GIOC initiates the READ operation and interprets DCW2 as the last data transfer DCW (Bit 17=1) for the connect in single physical record mode.

Communication between bootload and the program loaded

When the bootload program transfers control to the program it has loaded:

1. Go: GIOC base address : saved in index register 0
2. Bo: Bootload base address : saved in index register 1
3. Lo: Program loaded address : saved in index register 2
4. Po: Processor base address : is not known, so it cannot be passed to the program loaded. If the convention that Po is 0 modulo 1024 has been followed, Po can be computed from Bo.
5. The COW used by the bootload is at location Go+0
6. The entry point of the loaded program must be at location Lo+32 as shown below; this program may be the bootstrap initializer, or a dump program or a diagnostic program




If the tape cannot be read, the bootload will halt on the instruction DIS 1

Memory Map

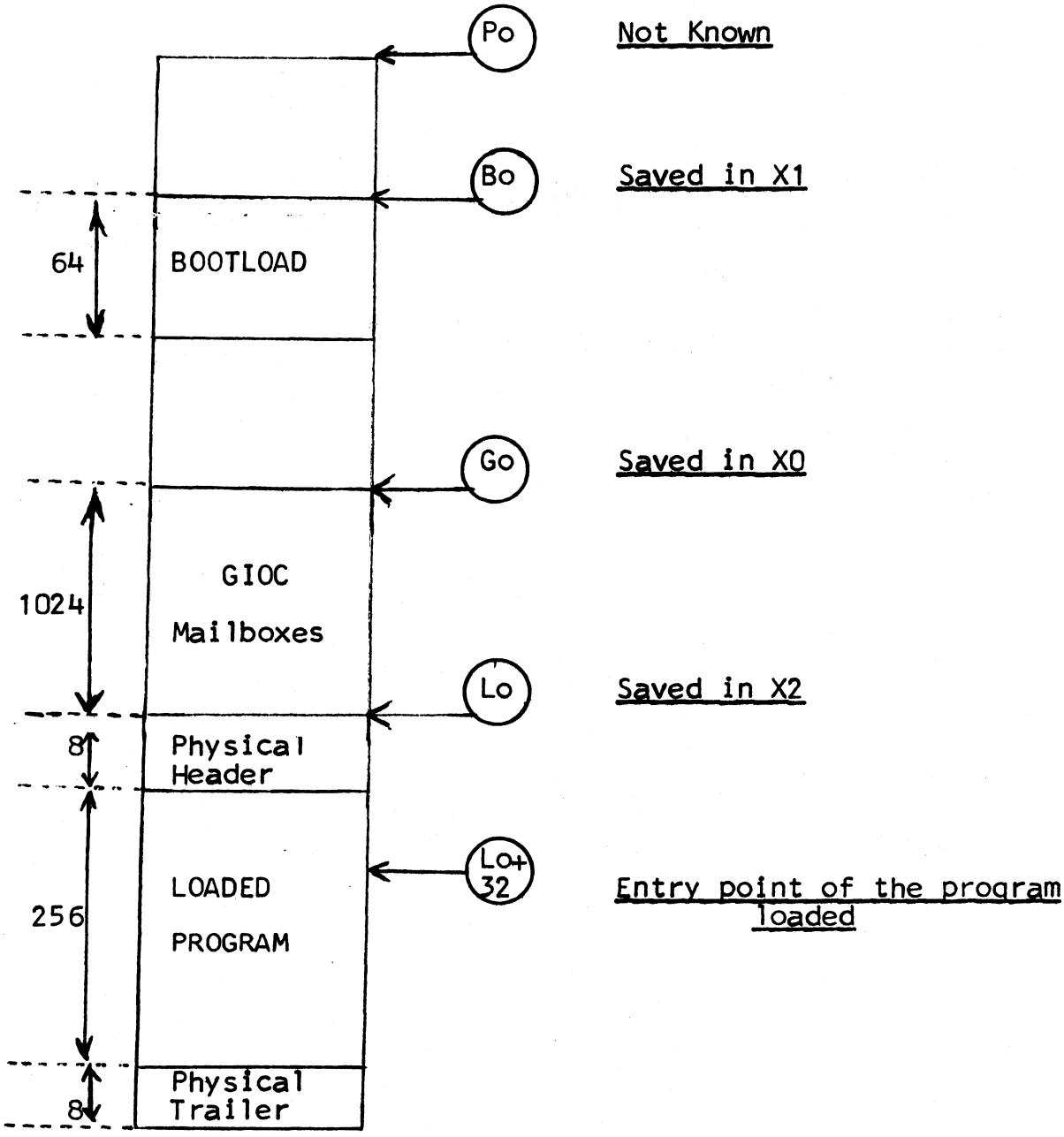

Po
Not Known
Bo
Saved in X1
64
BOOTLOAD
Go
Saved in X0
1024
GIOC
Mailboxes
Lo
Saved in X2
8
Physical
Header
Lo+
32
Entry point of the program
loaded
LOADED
PROGRAM
256
Physical
Trailer
8

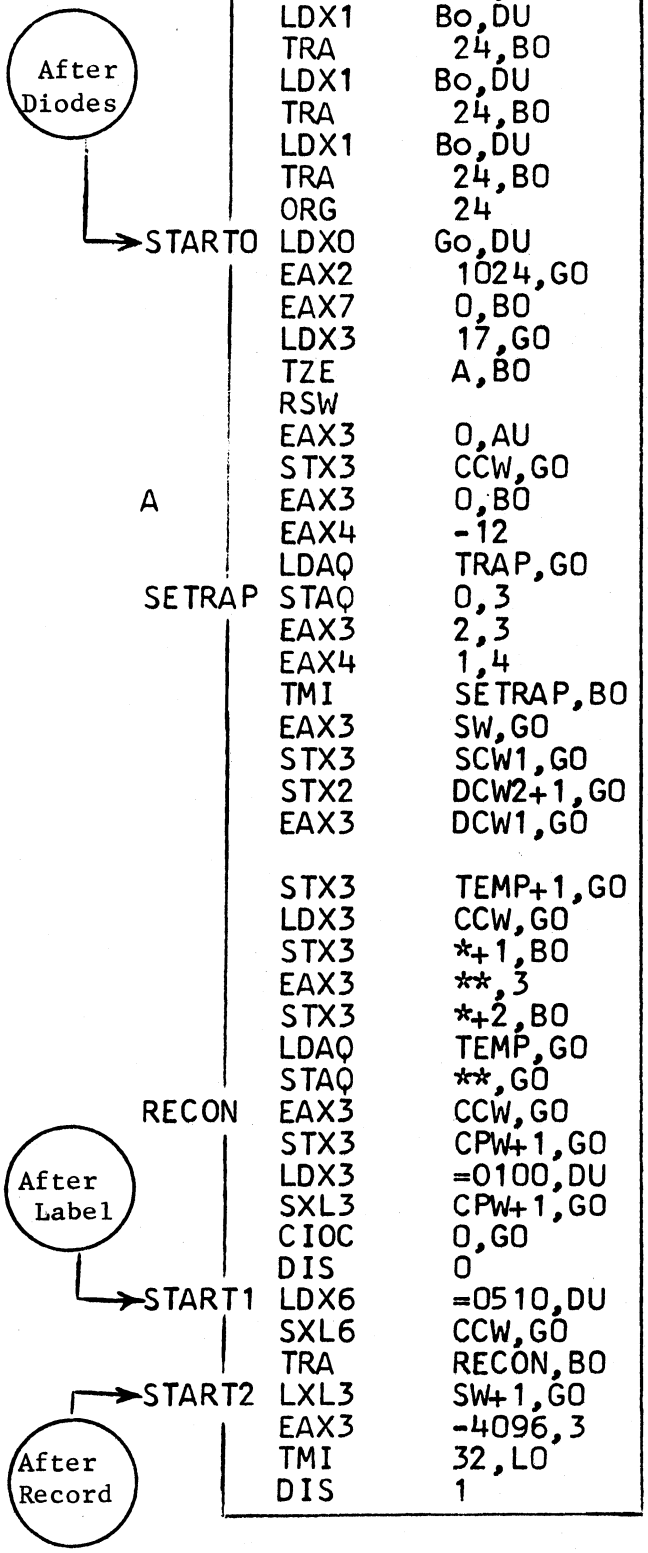
Figure 1: The Bootload Program

```
GO      EQU     0       GO is the symbolic name for index
                          register 0 that contains Go
BO      EQU     1       BO is the symbolic name for index
                          register 1 that contains Bo
LO      EQU     2       LO is the symbolic name for index
                          register 2 that contains Lo

CØW     EQU     0       Relative position of CØW in the
                          mailbox area
SCW1    EQU     3       Relative position of SCW1 in the
                          mailbox area
SW      EQU     4       Relative position of SW in the
                          mailbox area
CCW     EQU     10      Relative position of CCW in the
                          mailbox area
CPW     EQU     16      Relative position of CPW in the
                          mailbox area
DCW1    EQU     20      Relative position of DCW1 in the
                          mailbox area
DCW2    EQU     22      Relative position of DCW2 in the
                          mailbox area
TRAP    EQU     26      Relative position of TRAP in the
                          mailbox area
TEMP    EQU     28      Relative position of TEMP in the
                          mailbox area

        EVEN
```

CONTINUED
NEXT PAGE




```
        LDX1    Bo,DU       INTERRUPT VECTOR FOR INTERRUPT CELL 0
        TRA      24,BO
        LDX1    Bo,DU       INTERRUPT VECTOR FOR INTERRUPT CELL 1
        TRA      24,BO
        LDX1    Bo,DU       INTERRUPT VECTOR FOR INTERRUPT CELL 2
        TRA      24,BO
        LDX1    Bo,DU       INTERRUPT VECTOR FOR INTERRUPT CELL 3
        TRA      24,BO
        ORG      24
START0  LDX0    Go,DU       X0=BASE ADD GIOC = GO
        EAX2     1024,GO     X2=BASE ADD BOOTSTRAP LOADER
        EAX7     0,BO        SET X7 WITH BOOTLOAD BASE ADDRESS Bo
        LDX3     17,GO
        TZE      A,BO        TEST Go+17 BIT 17
        RSW                  READ SWITCHES IF BIT 17 ON
        EAX3     0,AU
        STX3     CCW,GO      STORE CHANNEL NUMBER FROM SWITCHES
A       EAX3     0,BO        X3=BOOTLOAD BASE ADDRESS=Bo
        EAX4     -12         FOR INTERRUPT CELLS 0 to 11
        LDAQ     TRAP,GO     ALL INT. VECTOR GO TO START1
SETRAP  STAQ     0,3
        EAX3     2,3         SET NEXT PAIR
        EAX4     1,4         INCREMENT COUNT
        TMI      SETRAP,BO   LOOP
        EAX3     SW,GO
        STX3     SCW1,GO     SET ADD IN SCW1
        STX2     DCW2+1,GO   SET ADD IN DCW DATA
        EAX3     DCW1,GO

        STX3     TEMP+1,GO   STORE Go+DCW1 IN TEMPORARY LPW
        LDX3     CCW,GO
        STX3     *+1,BO      CALCULATE THE LPW ADDRESS
        EAX3     **,3        DEPENDING ON THE CHANNEL NUMBER
        STX3     *+2,BO      STORE LPW ADDRESS IN *+2
        LDAQ     TEMP,GO     LOAD TEMPORARY LPW
        STAQ     **,GO       STORE TEMPORARY LPW IN REAL LPW
RECON   EAX3     CCW,GO      INIT. CPW FOR EACH CONNECT
        STX3     CPW+1,GO
        LDX3     =0100,DU    SET TALLY OF 1
        SXL3     CPW+1,GO
        CIOC     0,GO        INITIATE CONNECT OPERATION
        DIS      0           WAIT FOR I/O
START1  LDX6     =0510,DU    INITIALIZE CCW WITH READ COMMAND
        SXL6     CCW,GO      BEFORE ISSUING SECOND CONNECT
        TRA      RECON,BO    GO TO ISSUE ONE OTHER CONNECT
START2  LXL3     SW+1,GO     EXAMINE STATUS FROM READ
        EAX3     -4096,3
        TMI      32,LO       TRANSFER TO LOADED PROGRAM
        DIS      1           MAJOR STATUS NOT O.K.
```

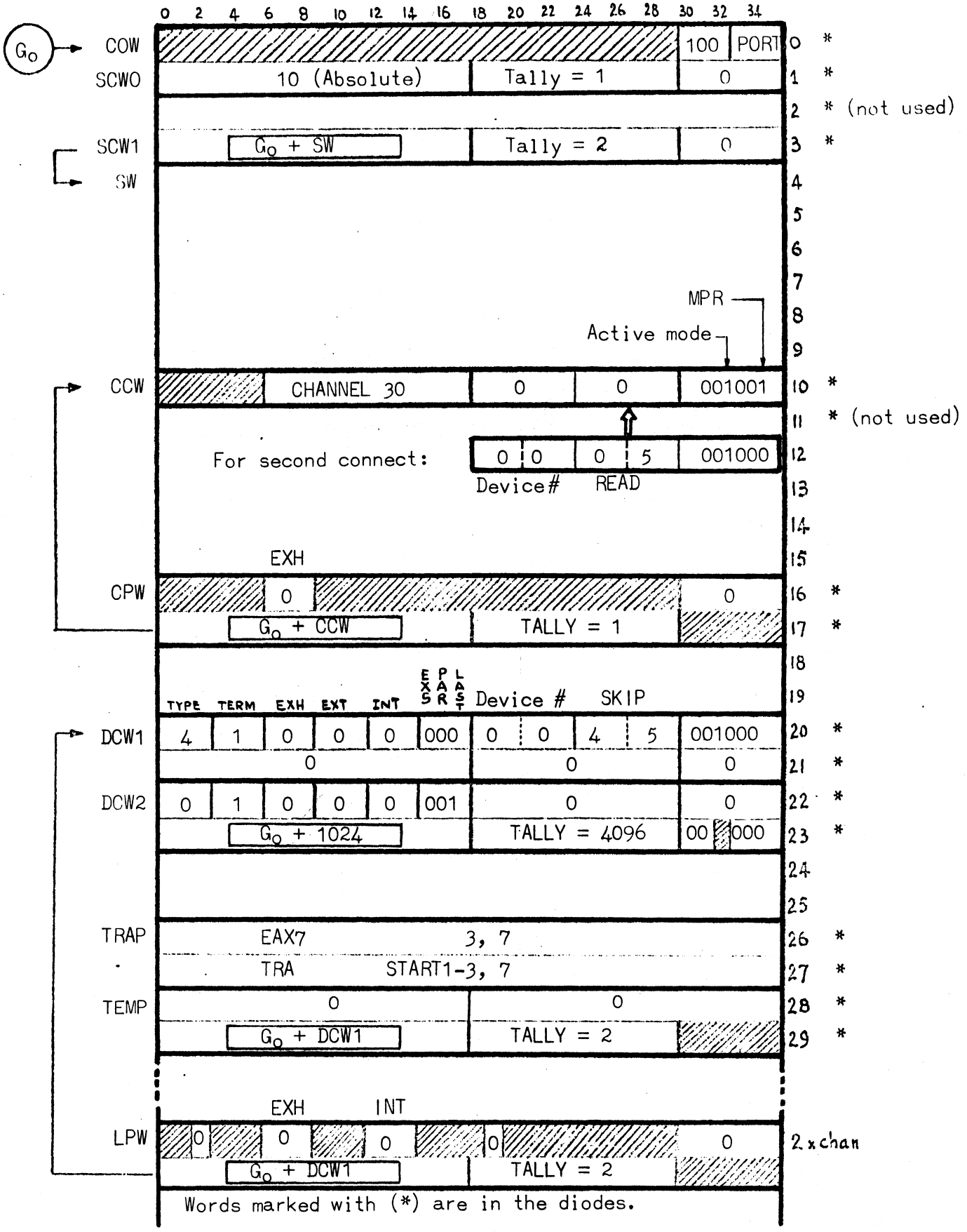



Figure 2: Mailboxes