Setting Commander/Servant Hierarchies

In a VXIbus system, a commander is a plug-in module which controls other plug-in modules. "Control" can be a commander such as the Agilent E1406A Command Module translating SCPI commands, and/or serving as the GPIB interface for (servant) modules within its servant area.

During the configuration sequence, the resource manager assigns servant modules to a commander module based on the servants' logical addresses and the commander's servant area. The concept of the servant area is shown in Figure 2-1. The *C-Size VXIbus Systems Configuration Guide* shows how to set the command module's servant area.

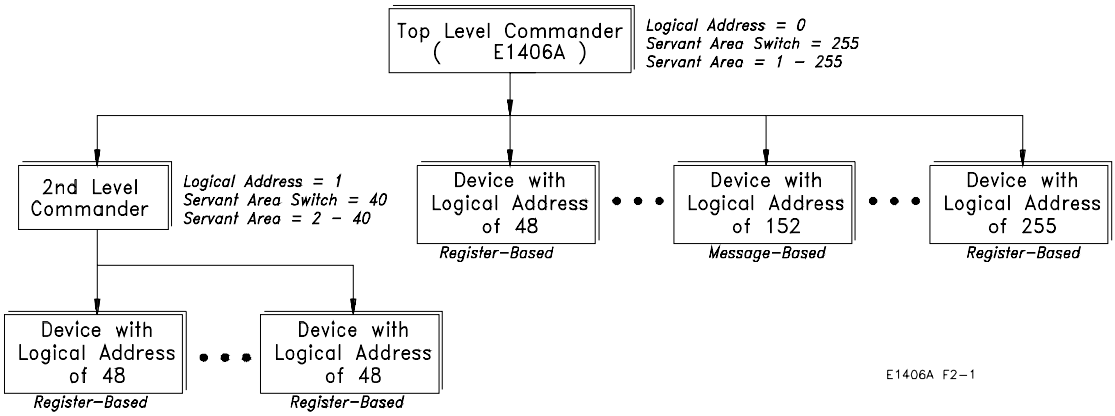

Figure 2-1. Example of Commander/Servant Hierarchy

Note the following regarding commander/servant relationships:

- A commander's servant area is its logical address + 1, through its logical address + its servant area switch setting.
- If within a given commander's servant area (Figure 2-1) there is another lower-level commander(s) (logical address 1), the given commander will control the lower-level commander. However, all modules within the servant area of the lower-level commander (logical addresses 2 - 41) will be controlled by the lower-level commander.
- If there is a commander outside the servant area of the command module/resource manager, that commander becomes a top level commander. The resource manager will assign all modules within the commander's servant area to that commander, or to that commander's lower-level commanders.
- The command module will always be the commander for IBASIC even if IBASIC's logical address (240) is outside the module's servant area. There can be multiple IBASICs in the same system since each is a servant to its respective command module. Note that there are no VXIbus registers for IBASIC.

User-Defined Commander/Servant Hierarchies

In some systems you may need to assign a servant to a commander that is outside the commander's servant area. In other systems, it may be necessary to change a module's secondary GPIB address, or assign secondary addresses to modules whose logical addresses are not instrument identifiers. These tasks can be accomplished with the user-defined commander/servant hierarchy table described in this section.

Note

Register-based instrument drivers that support multiple card sets normally require that the cards in the set have sequential logical addresses. When instrument drivers support non-sequential logical addresses, instruments that consist of non-sequential card sets must be created using the user-defined commander/servant hierarchy table. There must be an entry in the table for every card in the instrument card set.

The User-Defined Commander/Servant Hierarchy Table

User-defined commander/servant hierarchies and secondary GPIB addresses are specified with a commander/servant hierarchy table created in the command module. The table is created as follows:

1. Table space in the command module's non-volatile user RAM is made available by allocating a segment of RAM with the command:

   DIAGnostic:NRAM:CREate <*size*>

2. Reset the command module. NRAM is created during the boot-up process:

   DIAGnostic:BOOT:WARM

3. The location (starting address) of the table in RAM is determined with the command:

   DIAGnostic:NRAM:ADDRess?

4. Data is downloaded into the table with the command:

   DIAGnostic:DOWNload <*address*>, <*data*>

5. The table is linked to the appropriate algorithm in the command module processor with the command:

   VXI:CONFigure:CTABle <*address*>

Table Format

The format of the commander/servant hierarchy table is shown in Table 2-5.

Table 2-5. Commander/Servant Hierarchy Table Format

Valid Flag/ Number of Modules		
Laddr	Cmdr Laddr	Sec Addr
Laddr	Cmdr Laddr	Sec Addr
•	•	•
Laddr	Cmdr Laddr	Sec Addr

The table parameters are:

- **Valid Flag (1/0)** 1 indicates the table is valid and the modules should be configured accordingly. 0 (zero) will cause an error message (Error 38). Valid Flag is part of the table header and is represented by the upper eight bits of the header word.
- **Number of Modules (1 - 254)** is the number of entries in the table. Number of Modules is part of the table header and is represented by the lower eight bits of the header word.
- **Laddr** is the logical address of the module which is assigned a new commander or new secondary GPIB address. Field is one word.
- **Cmdr Laddr** is the logical address of the commander to which the module specified by **Laddr** is assigned. If -1 is specified, the module is not assigned to a commander. Field is one word.
- **Sec Addr (1 - 30)** is the secondary GPIB address assigned to the module specified by **Laddr**. If -1 is specified, the secondary address is assigned by default. Field is one word.

Determining the Table Size

The commander/servant hierarchy table has a one word header and three one word fields. The amount of RAM allocated with DIAGnostic:NRAM:CREate is specified in bytes. Since one word is two bytes, the amount of RAM to allocate is computed as:

```
2 + 6(N)
```

where N is the number of modules to be configured. For example, to assign three modules to a particular commander, the table size would be:

```
2 + 6(3) = 20 bytes
```

DIAGnostic:NRAM:CREate would be executed as:

```
OUTPUT  @E1406;"DIAG:NRAM:CRE  20"
```

Data Format

Data can be sent to the commander/servant hierarchy table in any convenient format, as long as the binary data is preserved. This can be accomplished using DIAGnostic:PEEK? and DIAGnostic:POKE, by reading the data into a variable in the computer and then downloading the data to the table using the Arbitrary Block Program Data format, and so forth. In the following example, this is accomplished by reading the data into 16 bit integer variables in the computer and then downloading the data to the table using the ANSI/IEEE 488.2-1987 Arbitrary Block Program Data format. More information on the Arbitrary Block Program format can be found on page 121 of this manual and in the *ANSI/IEEE 488.2-1987* document.

The table header is sent as a single 16-bit word which must contain the Valid Flag and the number of modules involved. **For a valid table, the header is 256 plus the number of modules.** For example, to indicate a valid table with seven entries, the header is 263 (256 + 7 = 263).

CAUTION **When downloading data into the commander/servant hierarchy table, DIAGnostic:DOWNload does not determine if the table is large enough to store the data. If the amount of data sent by DIAGnostic:DOWNload is greater than the (table) space allocated by DIAGnostic:NRAM:CREate, system errors will occur. You can recover from these errors by executing DIAGnostic:BOOT:COLD, or by pressing the "Ctrl-R" keys on an RS-232 terminal while cycling mainframe power.**

Example: Assigning a Secondary GPIB Address

The following program assigns secondary GPIB address 01 to the Agilent E1411B 5½-Digit Multimeter at logical address 25. The program notes each of the steps used to create and load the table.

```
10  !Assign an I/O path and allocate a variable to store commander/servant
20  !hierarchy data to be downloaded to the command module.
30  ASSIGN  @E1406 TO 70900;EOL  CHR$(10)  END
40  INTEGER  Cs_hier(1:4)
50  !
60  !Allocate a segment of non-volatile user RAM on the command module
70  !to store the commander/servant hierarchy table.
80  OUTPUT  @E1406;"DIAG:NRAM:CRE  8"
90  !
100 !Restart the system instrument to allocate the user RAM.  Wait for the
110 !restart to complete before continuing.
120 OUTPUT  @E1406;"DIAG:BOOT"
130 ON TIMEOUT 7,.1 GOTO Complete
140 Complete:    B=SPOLL(70900)
150 OFF TIMEOUT 7
160 !
170 !Return the starting address of the table in non-volatile user RAM.
180 OUTPUT  @E1406;"DIAG:NRAM:ADDR?"
190 ENTER  @E1406;A
200 !
210 !Download the following: the table is valid and one module is being
220 !assigned a secondary address, the logical address of the module is 25,
230 !its commander's logical address is 0, the secondary address is 01.
240 DATA  257,25,0,1
250 READ  Cs_hier(*)
260 OUTPUT  @E1406 USING    "#,3(K)";"DIAG:DOWN    ";A;" ,#0"
270 OUTPUT  @E1406 USING    "W";Cs_hier(*)
280 !
290 !Link the commander/servant hierarchy table to the appropriate algorithm.
300 OUTPUT  @E1406;"VXI:CONF:CTAB    ";A
310 !
320 !Restart the system instrument to set the user-defined configuration.
330 OUTPUT  @E1406;"DIAG:BOOT"
340 END
```

Comments

- The following errors are associated with the commander/servant hierarchy table:

  ERROR 12: INVALID UDEF COMMANDER LADD
  This error occurs when the user-defined commander logical address specified in the table (Cmdr Laddr) is not a valid commander. Either the commander does not exist, or it is not a message-based device.

  ERROR 14: INVALID UDEF SECONDARY ADDRESS
  This error occurs when the user-defined secondary address (Sec Addr) is invalid in the commander/servant hierarchy table. Valid secondary addresses are -1, 1 - 30. The error also occurs if the device to which the secondary address is assigned is outside the servant area of the command module.

  ERROR 15: DUPLICATE SECONDARY ADDRESS
  This error occurs when the same secondary address is specified for more than one module in the commander/servant hierarchy table.

  ERROR 18: INVALID COMMANDER LADD
  This error occurs when the commander specified in the user-defined commander/servant hierarchy table is not a valid message-based commander, or the device does not exist.

  ERROR 37: INVALID UDEF CNFG TABLE
  This error occurs when the user-defined commander/servant hierarchy table is not true (valid flag does not equal 1).

  ERROR 38: INVALID UDEF CNFG TABLE DATA
  This error occurs when there are 0 or greater than 254 entries in the user-defined commander/servant hierarchy table.

- The secondary GPIB addresses (and/or commanders) assigned by the commander/servant hierarchy table are used by the system until DIAGnostic:BOOT:COLD or VXI:CONFigure:CTABle 0 is executed.