PDP 1
SUPPLEMENT
(PDP-1D-45)
PDP-1D-45
SUPPLEMENT
Copyright 1964 by Digital Equipment Corporation
FOREWORD

This supplement describes special instructions added to PDP-1D-45 at Bolt Beranek and Newman. They are grouped as follows:

Memory Reference Instructions
- Load Character (LCH)
- Deposit Character (DCH)
- Twos Complement Add (TAD)

The Skip Group

The Special Operate Group

The Input-Output Transfer Group
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Reference Instructions</td>
<td>1</td>
</tr>
<tr>
<td>LCH</td>
<td>1</td>
</tr>
<tr>
<td>LCH Octal Code 13</td>
<td>2</td>
</tr>
<tr>
<td>DCH Octal Code 14</td>
<td>4</td>
</tr>
<tr>
<td>DCH Octal Code 15</td>
<td>5</td>
</tr>
<tr>
<td>TAD Octal Code 36</td>
<td>6</td>
</tr>
<tr>
<td>The Skip Group</td>
<td>7</td>
</tr>
<tr>
<td>The Special Operate Group</td>
<td>7</td>
</tr>
<tr>
<td>In/Out Transfer Group Instruction List</td>
<td>11</td>
</tr>
<tr>
<td>Memory and Processor Control IOT's</td>
<td>11</td>
</tr>
<tr>
<td>Miscellaneous Processor IOT's</td>
<td>11</td>
</tr>
<tr>
<td>Type 23 Drum IOT's</td>
<td>12</td>
</tr>
<tr>
<td>Data Communication System Type 630</td>
<td>12</td>
</tr>
<tr>
<td>Display IOT's</td>
<td>12</td>
</tr>
<tr>
<td>In/Out Transfer Group Detailed Description</td>
<td>12</td>
</tr>
<tr>
<td>Precision CRT Display 30</td>
<td>12</td>
</tr>
<tr>
<td>Character Generator 33</td>
<td>13</td>
</tr>
<tr>
<td>Parallel Drum 23</td>
<td>15</td>
</tr>
<tr>
<td>Data Communication System</td>
<td>17</td>
</tr>
</tbody>
</table>
Programmed Data Processor PDP-1D-45 System
MEMORY REFERENCE INSTRUCTIONS

LCH - Octal Code 12 - Load accumulator with a character from memory.
DCH - Octal Code 14 - Deposit a character from accumulator in memory.

Each of these instructions is interpreted as being deferred, hence requiring three memory cycles for execution. The MB and AC are divided into three sections of six bits each. Bits 0-5 = character one (1), bits 6-11 = character two (2), and bits 12-17 = character three (3).

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

The instructions are sub-decoded from MB bits 0 and 1 during the defer cycle. MB bits 0 and 1 are placed in the load-deposit register (LD) and decoded:

- Octal Code 12 and LD - 01 = LC1 - Load character one loads accumulator from memory bits 0-5 and places in accumulator bits 0-5. (AC0-5)
- Octal Code 12 and LD - 10 = LC2 - Load character two loads accumulator from memory bits 6-11, and shifts into AC bits 0-5.
- Octal Code 12 and LD - 11 = LC3 - Load character three loads accumulator from memory bits 12-17, and shifts into AC bits 0-5.
- Octal Code 14 and LD - 01 = DC1 - Deposit character one deposits accumulator bits 0-5 in memory bits 0-5.
- Octal Code 14 and LD - 10 = DC2 - Deposit character two deposits accumulator bits 0-5 in memory bits 6-11.
- Octal Code 14 and LD - 11 = DC3 - Deposit character three deposits accumulator bits 0-5 in memory bits 12-17.

LCH

The registers below show a single step sequence through the LCH instruction if 1008 is the starting location, and it contains a 128 to address 10008. The LCH instruction automatically forces a defer cycle. During the defer cycle, the memory address (MA) contains 10008, and the contents of the MB contain a 01 in the XX position and 20008 in the address portion of the
MB. During cycle one, 2000₈ would be the address. If the MB at this time is assumed to contain A, B, and C, the character A is transferred into the accumulator and the remaining 12 bits are cleared as shown in Figure a. During the defer cycle, if the XX portion of the MB contains 10, the character is transferred into the accumulator and the last 12 bits are cleared as shown in Figure b. If the XX portion of the MB contains 11 during the defer cycle, the results would appear as shown in Figure c.

The LCH instruction clears AC bits 6-17 and leaves the single character in AC bits 0-5.
When the defer bit is a 1 during cycle zero, it sets a one to the increment flip-flop (INC) placing the instruction in the automatic increment mode. In the defer cycle, this takes the first two bits of the MB and effectively adds one (+1) to them. The first time this is used or to enter the automatic mode, the first two bits of the MB should be zeros as the incrementing takes place before the character handling cycle (cycle one).

When entering the defer cycle if the address contains a:

```
00X XXX
```

It is incremented to contain:

```
01
```

If entered with a 01, it is incremented to:

```
10
```

A 10 is incremented to a:

```
11
```

In the last situation, an 11 causes the character bits to be forced to a 01 and the address portion of the MB to be incremented by one.

```
01 plus one
```

Summary: In the automatic mode a sequence performs as follows:

- 00 - Loads character one
- 01 - Loads character two
- 10 - Loads character three
- 11 - Increments the address (+1) and loads character one in the new address
- 01 - Character two
- 10 - Character three
- 11 - Plus one to address and loads character one in new address
- 01 - Character two
- 10 - Character three
- 11 - Plus one to address and loads character one in new address
NOTE: If the automatic mode is entered with a 01 in the first two bits of the memory location brought out during the defer cycle, the first character is skipped.

If a 00 is used in the non-automatic instruction, it is interpreted as a LC1 (01) and loads the accumulator from memory bits 0-5.

In the automatic mode a mid-instruction break is not allowed between the defer cycle and cycle one. (No sequence breaks can occur between the defer cycle and cycle one).

**DCH Octal Code 14**

Assuming a sequence of cycles as used in the LCH instruction, if the accumulator contains a series of characters thus:

```
A  B  C
```

and the memory location addressed during the defer cycle contains a DC1 (01) in the first two bits, at the end of cycle one the AC would contain:

```
B  C  A
```

The MB:

```
A  XXX  XXX  XXX  XXX
```

The X's indicate the information originally contained here remains unchanged.

If the memory location addressed during the defer cycle contains a DC2 (10), and the AC initially contains

```
A  B  C
```

The result in the AC would be

```
B  C  A
```

In the MB

```
XXX  XXX  A  XXX  XXX
```

A DC3 (11) provides the following results in the AC and MB if the AC initially contains the ABC.

AC

```
B  C  A
```

MB

```
XXX  XXX  XXX  XXX  A
```
The following is the result left in the AC and MB if a sequence of DCH instructions is used (non-automatic) and the AC initially contains an ABC in that order:

**DC1**

| AC | B | C | A |
| MB | A | X | X |

**DC2**

| AC | C | A | B |
| MB | A | B | X |

**DC3**

| AC | A | B | C |
| MB | A | B | C |

**Summary:** The DCH instruction always takes the character that is in the first six bits of the AC and places it in the character position designated by the first two bits decoded in the defer cycle: first character to first position, first character to second position, or first character to third position.

**DCH Octal Code 15**

The DCH instruction, using the indirect address bit (bit 5) of the word as a 1, sets the INC flip-flop and during the defer cycle increments the sub-instruction through the same sequence as shown for LCH. The automatic mode should be entered with a 00 in the location addressed in the defer cycle. (Reference from AC to MB)

- **DCH 00** - Deposits first character in first position.
- **DCH 01** - Deposits first character in second position.
- **DCH 10** - Deposits first character in third position.
- **DCH 11** - Increments the address and deposits the first character in the first position of the new address.
If the alphabet were typed in by a program sequence it might resemble this:

```
Start  cl a V c l f  /clear accumulator and flag 1
sz f i (1)     /listen loop
j m p -1
ty i      /bring in typed character
r c r (6)    /move from I/O to AC
s a d (77)   /compare for end character (77)
h l t
  d c h i s t o r e
  j m p s t a r t
```

The MB storage locations would be packed thus:

```
  1  A  B  C
  2  D  E  F
  3  G  H  I
  4  J  K  L
  5  M  N  O
  6  P  Q  R
  7  S  T  U
 10  V  W  X
 11  Y  Z
```

Summary: The DCH instruction deposits the accumulator bits 0-5 into the character location of the memory buffer specified by the bits 0, 1 of the location addressed in the defer cycle, and rotates the next character or that character contained in accumulator bits 6-11 into accumulator bits 0-5 so that it might be deposited in memory on the next use of this same DCH instruction.

If the automatic mode is entered with other than a 00, a character is skipped. A 00 used in the non-automatic mode is interpreted as a 01.

**TAD Octal Code 36**

**TAD - 2's complement add**

The state of the link is sensed, and if a ONE, one is added to the AC (+1 to AC). The C(Y) are then added to the C(AC). The result is left in the AC and the original C(AC) are lost. The C(Y) are unchanged. A carry out of bit 0 is retained in the link flip-flop.
THE SKIP GROUP

644000 SNI Skip on non-zero I/O
Tests the I/O register for the presence of a one. Skips the next instruction in sequence if any of the I/O bits is a one.

654000 SZI Skip on zero I/O
Tests the I/O register for the all-zero condition. Skips the next instruction in sequence if it exists.

760020 LIA Load I/O register from AC
Load the I/O register from the accumulator

760040 LAI Load accumulator from I/O
Loads the accumulator from the I/O register

760060 SWP Exchange AC and I/O
Places the original contents of the AC into the I/O and the original contents of the I/O into the AC

770000 CMI Complement the I/O
Is the logical NOT of the contents of the I/O register

THE SPECIAL OPERATE GROUP

The special operate group of instructions is a new set of microprogram instructions. It uses octal code (74). Execution time is 5 microseconds.

The ring mode is also handled by the special operate group. The ring mode flip-flop (RNG) is set, cleared, or sampled with the program flag instructions. Its condition is transferred to I/O bit 11, and it can be set by the condition of I/O bit 11.

Ring mode is the condition whereby the address portion of the word can be caused to loop repetitively over a section of memory. Ring mode is an eight location loop, starting with the three least significant bits in the address portion of the word. It is indexed to seven \((111_2)\) and then back to zero \((000_2)\). See figure at end of SPC group on page 11.
Three instructions are affected by the ring mode: the load or deposit a character (LCH + DCH); index a character (IDC); and index the accumulator (IDA). Ring mode does not affect the add or normal index instructions (ADD or IDX). Setting, clearing, or sampling of the RNG flip-flop can be thought of as program flag zero.

The link flip-flop has been added to the accumulator to receive the carry out of AC₀ in 2's complement add (TAD). The link flip-flop is placed in I/O bit 10 when transferring the contents of the program flags to the I/O. It is set when transferring the contents of the I/O to the program flags by the condition of I/O bit 10.

<table>
<thead>
<tr>
<th>Event Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Time 1        SCI  SCF  CLL  SZL</td>
</tr>
<tr>
<td>Event Time 2        SCM  IFI  IIF</td>
</tr>
<tr>
<td>Event Time 3        IDA  CML</td>
</tr>
<tr>
<td>Event Time 4        IDC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>5  6  7  8  9  10  11  12  13  14  15  16  17</td>
</tr>
<tr>
<td>Reverse Skip IIF IFI IDC IDA SCM SCI SCF SZL CLL CML X X</td>
</tr>
</tbody>
</table>

744000 IIF Inclusive OR of the I/O from flags. Forms the inclusive OR of the link, RNG, and program flags 1 through 6 in I/O register bits 10 through 17

- Link - I/O₁₀
- Ring Mode - I/O₁₁
- Program Flag 1 - I/O₁₂
- Program Flag 2 - I/O₁₃
- Program Flag 3 - I/O₁₄
- Program Flag 4 - I/O₁₅
- Program Flag 5 - I/O₁₆
- Program Flag 6 - I/O₁₇

If used with SCI, the I/O is cleared prior to readin.
IFI 742000  
Inclusive OR of the flags from I/O. Forms the inclusive OR of I/O register bits 10-17 and leaves in the link, RNG, and program flags 1 through 6 respectively.

- I/O_{10} = Link
- I/O_{11} = Ring Mode
- I/O_{12} = Program Flag 1
- I/O_{13} = Program Flag 2
- I/O_{14} = Program Flag 3
- I/O_{15} = Program Flag 4
- I/O_{16} = Program Flag 5
- I/O_{17} = Program Flag 6

IDC 741000  
Index character
Indexes bits 0 and 1 of the AC. Operates as a pointer word for the LCH or DCH instructions

\[
\begin{align*}
AC_{0-1} \neq & 11 \rightarrow \text{to } AC_1 \\
AC_{0-1} = & 11 \rightarrow \text{to } AC_0. \text{ The end around carry then causes the address portion of the word to be indexed.}
\end{align*}
\]

See figure below.
<table>
<thead>
<tr>
<th>Code</th>
<th>mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>740400</td>
<td>IDA</td>
<td>Index Accumulator, Adds one to the contents of the AC (no effect on MB).</td>
</tr>
<tr>
<td>740200</td>
<td>SCM</td>
<td>Special Complement&lt;br&gt;Complements the accumulator and adds one to the accumulator if the link bit was a one. (Does not perform the 2's complement in itself.) ANDed with IDA the 2's complement of a number is obtained.</td>
</tr>
<tr>
<td>740100</td>
<td>SCI</td>
<td>Special Clear I/O&lt;br&gt;Clears the I/O register at the first event time.</td>
</tr>
<tr>
<td>740040</td>
<td>SCF</td>
<td>Special Clear Flags&lt;br&gt;Clears the link, RNG and the six program flags.</td>
</tr>
<tr>
<td>740020</td>
<td>SZF</td>
<td>Skip on zero link&lt;br&gt;Skips the next instruction in sequence if the link is a zero. 750020 will skip the next instruction in sequence if the link is a one.</td>
</tr>
<tr>
<td>740010</td>
<td>CLL</td>
<td>Clear Link&lt;br&gt;Clears the link flip-flop at event time one.</td>
</tr>
<tr>
<td>740004</td>
<td>CML</td>
<td>Complement Link&lt;br&gt;Forms the logical negation of the link. If a one it is changed to a zero. If a zero it is changed to a one.</td>
</tr>
</tbody>
</table>
RNG

Address portion of MB

<table>
<thead>
<tr>
<th>Ixxx</th>
<th>XXX XXX XXX 000</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
</tr>
<tr>
<td>011</td>
<td></td>
</tr>
<tr>
<td>010</td>
<td></td>
</tr>
<tr>
<td>001</td>
<td></td>
</tr>
</tbody>
</table>

***IN/OUT TRANSFER GROUP***

Memory and Processor Control IOT's

<table>
<thead>
<tr>
<th>720011</th>
<th>ERG</th>
<th>Enter Ring Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>720010</td>
<td>LRG</td>
<td>Leave Ring Mode</td>
</tr>
<tr>
<td>720064</td>
<td>LRM</td>
<td>Leave restrict mode</td>
</tr>
<tr>
<td>720065</td>
<td>ERM</td>
<td>Enter restrict mode</td>
</tr>
<tr>
<td>720066</td>
<td>RNM</td>
<td>Rename memory</td>
</tr>
<tr>
<td>720067</td>
<td>RSM</td>
<td>Reset memory banks</td>
</tr>
</tbody>
</table>

Miscellaneous Processor IOT's

<table>
<thead>
<tr>
<th>72XX32</th>
<th>RCK</th>
<th>Read clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>72XX35</td>
<td>CTB</td>
<td>Clear trap buffer</td>
</tr>
<tr>
<td>72XX17</td>
<td>RRO</td>
<td>Rem-rand out</td>
</tr>
<tr>
<td>72XX37</td>
<td>RRI</td>
<td>Rem-rand in</td>
</tr>
</tbody>
</table>
Type 23 Drum IOT's

72XX61* DIA Drum initial address
72XX62* DWC Drum word count
720063* DCL Drum core location
722061 DBA Drum break address
722062 DRA Drum request address

Data Communication System Type 630

720022 RCH Receive a character
721022 RCR Receive a character and release the scanner
725022 TCC Transmit a character from receiver counter
724022 TCB Transmit a character from send buffer
720122 RRC Read the receiver counter
724122 SSB Set the send buffer
721122 RSC Clear flag and release scanner

Display IOT's

720007 DPY Display one point (intensify)
722027 GPL Generator plot left
720027 GPR Generator plot right
720127 GCF Reset
722026 GLF Load format
720026 GSP Space
722007 SDB Load buffer, no intensify

Precision CRT Display (30)

720007 DPY Display one point. Clears the light pen status bit and displays one point using bits 0 through 9 of the AC to represent the (signed) X coordinate of the point and bits 0 through 9 of the I/O as the (signed) Y coordinate.

AC bits 0-9 and I/O bits 0-9 are loaded before the DPY instruction is given. The line specified by the AC is the signed X coordinate. Plus (0) in AC bit 0 plots points from the center of the screen.

*Above must be given in sequence shown.
cathode ray tube to the right 4-5/8 inches. Minus (1) in AC bit 0 plots the points from the center of the tube to the left 4-5/8 inches. The line specified by the I/O is the signed Y coordinate. Plus (0) in I/O bit 0 plots points vertically up from the center. Minus (1) in I/O bit 0 plots the points from the center down.

\[
\begin{align*}
\text{x} &= 1000 \quad \text{•} \quad \text{y} = 0777 \\
\text{x} &= 0000 \quad \text{•} \quad \text{y} = 0777 \\
\text{x} &= 0777 \quad \text{•} \\
\end{align*}
\]

\[
\begin{align*}
\text{x} &= 1000 \quad \text{•} \quad \text{y} = 0000 \\
\text{x} &= 0000 \quad \text{•} \quad \text{y} = 0000 \\
\text{x} &= 0777 \quad \text{•} \\
\end{align*}
\]

\[
\begin{align*}
\text{x} &= 1000 \quad \text{•} \quad \text{y} = 1000 \\
\text{x} &= 0000 \quad \text{•} \quad \text{y} = 0000 \\
\text{x} &= 0777 \quad \text{•} \quad \text{y} = 1000 \\
\end{align*}
\]

Character Generator (33)

722027 GPL Generator plot left. Transfers the contents of the I/O register of the symbol generator and initiates plotting of the first 17 dots. I/O[17] of this word sets or resets the subscript control as the bit is a 1 or a 0.

720027 GPR Generator plot right. Transfers the contents of the I/O register to the shift register of the symbol generator and initiates plotting of the last 18 dots. The "Clear" is inhibited by MB[7] to prevent losing the count contained in the horizontal and vertical counter which controls dot position.

720127 GCF Reset. Clears the light pen status flip-flop in the display. The light pen status will also be cleared when a normal point plot (IOT-07) is performed.
Load format. The three least significant bits of the I/O register are sent to the character-size control flip-flops and the spacing control circuits. Bits 16 and 17 specify one of four symbol sizes as shown in the tables below. Bit 15, if a 1, specifies automatic spacing between symbols. A completion pulse is not generated by the display when this instruction is performed.

Matrix size, and hence character size, is determined by the number of increments separating the dots on the matrix, when an increment is defined as 1/1024th of the width of height of the display area. The relationship between character size and incremental separation of dots is given below.

<table>
<thead>
<tr>
<th>Character Size</th>
<th>Bit 16</th>
<th>Bit 17</th>
<th>Number of Increments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

Space. Increments the X buffer-counter to position the beam one character position to the right. Since the contents of the I/O register are transferred to the shift register by this instruction, the I/O register should be cleared before performing the "Space" instruction.

Load buffer - intensify

Load buffer - no intensify. By use of the MB₁₂ bit, the normal point plotting instruction (IOT-07) can be used to load the position coordinates of the first character to be displayed without illuminating that point. When the "No Intensify" instruction is performed, the display will not generate a completion pulse; therefore, the computer must allow at least 25 microseconds before executing a gpl instruction.

Except for the gcf, glf, and sdb instructions, which do not cause the generation of a completion pulse, the preceding iot instructions can be coded to perform the in-out wait operations.
Parallel Drum (23)

72X061* DIA
Drum initial address
Transfers the contents of the I/O register to the drum and
is decoded to mean:

IO\_0 = Read
IO\_1-5 = Field to be read
IO\_6-7 = Drum initial address

720062* DWC
Drum word count
Transfers the contents of the I/O register to the drum and
is decoded to mean:

IO\_0 = Write
IO\_1-5 = Field to be written
IO\_6-17 = Words to be transferred

720063* DCL
Drum core location

IO\_2 - IO\_3 = Select memory 0
IO\_2 - IO\_2 = Select memory 1
IO\_2 - IO\_3 = Select memory 2
IO\_2 - IO\_3 = Select memory 3
IO\_4-17 = Starting core address and GO →

722061 DBA
Drum break address
Is decoded the same as DIA. When a DBA instruction is
given, a sequence break occurs when drum address = drum
initial address. If programming consideration can accept
the break, DBA can be used in place of DIA in the drum
sequence of instructions.

720064 LRM
Leave restrict mode
Zeros the restricted mode flip-flops. No memories are
protected. All instructions are legal except incorrect OP
code selections.

*Above must be given in sequence shown.
Enter restrict mode

When entering restrict mode, the i/O register should be preloaded with the desired memories to be protected.

- \( IO^0 \) = Protect memory 0
- \( IO^1 \) = Protect memory 1
- \( IO^2 \) = Protect memory 2
- \( IO^3 \) = Protect memory 3

When in the restrict mode if an incorrect operation code, in-out transfer (IOT), a HLT or any protected memory is addressed, it causes the restrict mode logic to initiate a break to channel 168 in the sequence break logic.

Rename memory

Rename memory takes the memory designated by X and renames it to the number contained in Y. There are four memories and they can be named in any of 16 different configurations. All addressed memories are checked for name. Memory rename logic cannot be bypassed. See RSM.

Reset memory

Restores the physical name to all memories. Zero is a zero, etc.

Read clock

The I/O is cleared and the clock buffer is read into the I/O register.

The clock register is automatically synchronized to the computer timing, and it is not necessary to read clock register more than once. The clock is a pulse at a 1 kc rate and can be read as often as desired.

Clear trap buffer

The trap buffer (which is loaded at the time a restrict mode trap occurs) is cleared by this IOT.
72XX17  RRO  Rem-rand out
Transfers the condition of the I/O register bits 0-17 to
the Rem-Rand Control.

72XX37  RRI  Rem-rand in
Clears the I/O and reads the conditions of the Rem-Rand
Control into the I/O register bits 0-17.

720011  ERG  Enter ring mode
Places the computer in the ring mode. (See special operate
group for detailed description.)

720010  LRG  Leave ring mode
Zeros the ring mode flip-flop.

Data Communication System

The 630 Data Communication System is assigned one basic IOT instruction, octal code 720022.
(Bits 0-17)

The basic instruction is microprogrammed to form a set of useful computer instructions for oper­
at­ing the DCS. Adding or ORing $2000_8$ to the octal equivalent causes the I/O to be cleared
before the operation is executed.

The following instructions control the scanner, the teletype transmitters and teletype receivers.
For convenience, bit configurations are assigned mnemonics as follows:

720022  RCH  Receive a character to I/O 10-17 (8 bits) (13-17, 5 bits)
using the receiver counter. The OR function occurs. Clear
the receiver flag. I/O bits 10-17 must be zeros prior to
operation execution.

721022  RCC  Same as RCH. Clear the scanner flag (release the scanner).

725022  TCC  Transmit a character using the receiver counter (I/O 10-17,
8 bits; I/O 13-17, 5 bits, to the transmitter). Clear the
receiver flag. Clear the scanner flag (release the scanner).
724022 TCB Transmit a character using the send buffer (I/O 10-17, 8 bits; I/O 13-17, 5 bits, to the transmitter). Clear the receiver flag.

720122 RRC Read the receiver counter (counter to I/O 12-17). The OR function occurs.

724122 SSB Set the send buffer (I/O 12-17 to send buffer). Used to select an idle station for transmission.

721122 RSC Clear the scanner flag (release the scanner).

The state of the scanner flag may be read into I/O register bit 16, using the check status instruction (1 = flag on).

Initialization procedures must at least include clearing of the scanner flag. (Actually all receiver flags should be cleared.)

The priority level to which the scanner flag is assigned is dependent upon the equipment configuration of your system.