

BALLISTIC MISSILE
DEFENSE ORGANIZATION
7100 Defense Pentagon
Washington, D.C. 20301-7100

**PARALLEL FUNCTION PROCESSOR
HARDWARE OPERATION**

**SPECIAL TECHNICAL REPORT
REPORT NO. STR-0142-90-005**

January 23, 1990

**GUIDANCE, NAVIGATION AND CONTROL
DIGITAL EMULATION TECHNOLOGY LABORATORY**

Contract No. DASG60-89-C-0142

Sponsored By

The United States Army Strategic Defense Command

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30332 - 0540

Contract Data Requirements List Item A004

Period Covered: Not Applicable

Type Report: As Required

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20010829 013

UL10300



DISCLAIMER

DISCLAIMER STATEMENT - The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

DISTRIBUTION CONTROL

- (1) DISTRIBUTION STATEMENT - Approved for public release; distribution is unlimited.
- (2) This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227 - 7013 October, 1988.

**PARALLEL FUNCTION PROCESSOR
HARDWARE OPERATION**

January 23, 1990

Author

Michael B. Woods

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30332-0540

Eugene L. Sanders

USASDC

Contract Monitor

Cecil O. Alford

Georgia Tech

Project Director

Copyright 1990

Georgia Tech Research Corporation

Centennial Research Building

Atlanta, Georgia 30332

TABLE OF CONTENTS

1. Scope	1
1.1 Identification	1
1.2 System Overview	1
1.3 Document Overview	1
2. Referenced Documents	1
3. Computer System Operation	2
3.1 Computer System Preparation and Shutdown	2
3.1.1 Power On and Off	2
3.1.2 Initiation	2
3.1.3 Shutdown	2
3.2 Operating Procedures	3
3.2.1 Input and Output Procedures	3
3.2.2 Monitoring Procedures	3
3.2.3 Recovery Procedures	4
3.2.4 Off-line Routine Procedures	4
3.2.5 Other Procedures	5
4. Diagnostic Procedures	5
4.1 Diagnostic Features Summary	5
4.2 Diagnostic Procedures	5
4.2.1 Multibus Repeater System Test	5
4.2.2 Crossbar Transfer Test	6
4.2.3 Sequencer Handshake Test	6
4.2.4 Crossbar and Sequencer Memory Test	7
4.2.5 Hardware Assisted Function Test	7
4.3 Diagnostic Tools	8
4.3.1 Digital Oscilloscope	8
4.3.2 Digital Multimeter	8
4.3.3 Logic Analyzer	8
5. Notes	8
5.1 Acronym Glossary	8
Appendix A Figures	9
Figure 3.1-1 Location of Ckt Breakers & Pwr Cables	9
Figure 3.1-2 Location of Power Switches	9
Appendix B Debug Examples Using PFP Displays	10
Example 1 Two Sends Each With Multiple Receivers	10
Example 2 One Processor Broadcasting to 31 Others	11
Example 3 Five Simultaneous Sends With Multiple Receives	12

1. SCOPE

1.1 Identification

This Computer System Operator's Manual (CSOM) applies to Georgia Tech's Parallel Function Processor (PFP), Georgia Tech part number CERL002-0757-000.0.

1.2 System Overview

The PFP is a 64 processor computing engine for use in computationally intensive applications that can be partitioned into functional blocks. The processors are grouped in two 32 processor clusters running from one common host. Each 32 processor cluster is connected by a crossbar switch. All inter-processor communication takes place over the crossbar(s). Simultaneous transfers may take place independently and switch patterns may be changed every cycle. In order to program the machine correctly, all inter-processor communication and data transfer lengths must be known beforehand.

The PFP has been designed to accommodate "hardware in the loop" simulations running in real time. Actual hardware components may first be simulated on one or more processors and later replaced with actual hardware interfaced to specified crossbar ports. The inputs and outputs to/from the device will appear identical to those it would see in an actual system.

1.3 Document Overview

The purpose of this manual is to give the reader a basic understanding of how the PFP works, its capabilities, and how to use it. The manual contains all instructions for operating the unit and a brief architectural overview. Refer to document #CERL002-0757-100.1 for detailed programming instructions.

2. REFERENCED DOCUMENTS

PFP Programmer's Guide

PFP Technical Data Package

Intel System 310 Hardware Literature

Intel iRMX Operating System User's Manual

Intel iMDDX Diagnostics User's Guide

3. COMPUTER SYSTEM OPERATION

3.1 Computer System Preparation and Shutdown

3.1.1 Power On and Off

Power is supplied to the PFP through seven (7) separate AC power cables located at the bottom rear side of the unit. Each cable must be plugged into a circuit capable of supplying 115V, 60 Hz, power at 15 amperes.

Each of the seven AC power circuits is controlled by a circuit breaker located in the rear of the unit. All seven circuit breakers must be turned on for full power to the PFP. Selected breakers can be off if the full PFP is not being used, or for maintenance purposes. An orange light next to each breaker indicates that it is on. See Figure 3.1-1 for location of the breakers.

There are seven (7) power switches located on the front of the machine (See Figure 3.1-2). Six of the switches turn on power to the processor racks. The seventh switch turns on power to the crossbars. Turn on all seven switches to activate full power to the unit.

The Intel 310 host may be powered on its own AC circuit or from the same circuit as the center rack. The power switch to the 310 is located on the rear side of the host case. Turn the switch to the on position to activate power to the host.

3.1.2 Initiation

After power up or system reset, the Intel 310 host automatically runs through self diagnostics and a bootstrap program. The system then prompts the user to log on with the following prompt:

LOGON:

The host is now initialized and ready to use.

On power up or system reset, each iSBC 286/12 processor in the PFP automatically runs through self diagnostics. Each processor has 4 LEDs on the front to indicate status conditions. When the green LED is on and the other three are off, that processor has completed its self diagnostics. When all processors have completed their self diagnostics the system is ready to use.

The PFP does not run an operating system and does not require any special initialization. All programs are down loaded from the host as executable code and given a start command by the host. Data is read back by the host during program execution or after the program has finished.

3.1.3 Shutdown

Prior to shutting down the Intel 310 or the PFP, all data that may be needed later must be saved. Data on the PFP must be read by the host and saved on one of the host's storage media (i.e., tape, floppy, or hard disk). After data has been saved, the host is shut down by turning off the power switch located at the rear of the unit.

The PFP is shut down by turning off the seven power switches located on the front of the center rack on the PFP.

3.2 Operating Procedures

3.2.1 Input and Output Procedures

All input and output media for the PFP system are accessed through the host Intel 310. The host contains a 40 MB hard disk drive, a 360 KB floppy disk drive, and a 60 MB cartridge tape drive. The host runs Intel's iRMX I operating system. In order to store data generated by the PFP, it must be read by the host and then stored to the selected device using the appropriate iRMX system command. See Volume 4, "Operator's Guide to the Human Interface", of the iRMX I Operating System Manual for detailed instructions.

The host is set up as a single user system. All system prompts and messages appear at this console. After the host is powered up or the system is reset, the host will run through self diagnostics then come up with the log on prompt. To use the PFP in conjunction with the 310 host, enter the following responses:

LOGON: pfp

PASSWORD: kirk

To exit the system, type the command "logoff".

3.2.2 Monitoring Procedures

Write statements may be sent to the console screen to monitor the operation of software running on the host 310. If the host is running in conjunction with the PFP, it may be used to read data from the PFP and display it (or save it) while the program is executing.

The PFP has two devices for monitoring a program during execution, 1) the crossbar status displays and 2) the sequencer/processor transition boards. These displays are most valuable for debugging new application programs as they are being developed. The crossbar status display shows the current switch patterns set in the crossbar. When a program is executing properly, the display appears as a bunch of flickering lights. When the program stops or hangs up, it appears as a set of arrows on a rectangular grid showing which processors are sending on that cycle, which processors are receiving, and the data paths being used. Two displays exist in the front of the center rack of the system. The top

display shows the crossbar patterns for the left bank of 32 processors. The bottom display shows the switch patterns for the right bank of 32 processors.

The sequencer/processor transition boards are located inside the rear doors of the two processor racks. Each board has LED's indicating the status of the four handshake lines coming from each processor. The four lines from each processor are DAV/ (data is available at the processor and ready to send to the network), ROT/ (read data out of the processor into the network), RFI/ (processor is ready for input from the network), and WIN/ (write data into the processor from the network). All lines are asserted low. An LED is off if the line is asserted and on if it is not. As with the crossbar display, the display appears as flickering lights during program execution. When a program hangs up, the lights can be used for program diagnosis.

The crossbar status display and sequencer/processor transition boards actually function as a pair when debugging a program. The crossbar status display allows the user to tell which program cycle the machine has stopped on. The sequencer/processor transition boards allow the user to tell which processor hung up the system on that cycle. That processor can then be interrogated by the host to determine its status. The code for that block can be modified to send more debug information to the host until the problem is resolved. Appendix B contains 3 example crossbar transfer cycles and the display patterns (both crossbar display and sequencer/processor transition boards) that go with them. Each example is explained in detail and should be read to learn what to look for on the displays.

3.2.3 Recovery Procedures

In the event the operation of the Intel host is interrupted or aborted, the system can be re-booted by pressing the button labeled "R" on the front lower left corner of the host case. The machine will run through its self diagnostics and come up with the logon prompt.

In the event the operation of the PFP is interrupted, aborted, or hangs up, the system may be reset in one of two ways. A software reset may be issued by the host, or a hardware reset may be issued by pressing the reset button on the PFP. The RESET button is located on the center rack directly below the power switches. (See Figure 3.1-1) The result is identical. The software command merely allows the user to reset the machine from a remote location. After the machine has been reset, the application must be re-loaded and re-started in order to run correctly.

3.2.4 Off-line Routine Procedures

The Intel 310 host requires no off-line routine procedures other than routine cleaning of the floppy disk drive and cartridge tape drive. See the Intel 310 Hardware User's Manual for instructions and maintenance intervals.

The PFP requires the four air filters located on the front of the machine and the air filter located beneath the crossbar to be taken out and cleaned monthly (more often if the filters appear dirty).

3.2.5 Other Procedures

To be added later.

4. DIAGNOSTIC PROCEDURES

4.1 Diagnostic Features Summary

The Intel 310 runs internal diagnostics on power up or system reset which test the processor subsystem, memory subsystem, and boot subsystem. Comprehensive diagnostic programs for the 310 are available from Intel in their iMDDX diagnostic package (order number 174032-001).

Each of the iSBC 286/12 processors in the PFP run their own diagnostics on power up or system reset. The processor and memory subsystems are tested similarly to the processor and memory in the host. While the test is running a red LED located on the front of the processor remains on. After the test is passed a green light comes on. The test runs approximately five seconds. If the red light remains on longer than this time, the processor has failed and should be replaced.

Comprehensive diagnostics for the PFP are stored on the host 310. Programs are available to test all major portions of the system including the Multibus repeater system, crossbar data paths, sequencer handshake cables, sequencer and crossbar memory, and the hardware assisted functions on the processor. Each test runs in an infinite loop until it is stopped at the host. Errors can be reported to a disk file or to the printer. Each test also has output to the system terminal to indicate the current state of the test while it is running.

4.2 Diagnostic Procedures

4.2.1 Multibus Repeater System Test

The Multibus Repeater Test checks the master Multibus repeater located in the host, the slave repeaters located in each processor rack, the Multibus backplanes located in each processor rack, the Multibus interface on each processor, and the associated cabling. The host first sends a bit pattern of 0000 (hex) to each processor and reads it back, then repeats this series of instructions for the patterns 5555, AAAA, and FFFF. Errors are reported when the value received does not match the value sent. When reporting an error, the value sent, the value received, the difference, and the processor involved are all listed. Test results may be routed to the printer or to a disk file.

In order to run the test, follow these steps:

- 1) Sign on the Intel 310 using the logon name "pfp" and the password "kirk".
- 2) Move to the test directory by typing "cd test/mu".

3) Type "submit mu" to run the test. The terminal screen will come up with a display indicating that the test is running. The test will run in an infinite loop until it is manually stopped.

4) To stop the test, type a "control C" at the terminal.

4.2.2 Crossbar Transfer Test

The Crossbar Transfer Test checks all data paths through both crossbars, the sequencer/crossbar interface, all processor/crossbar interfaces, and associated cabling. The test sends a data word to one processor over the Multibus. That processor then broadcasts that word to the other 31 processors on that crossbar. The host then reads that word back from all 32 processors (over the Multibus) to make sure that the correct data was received. Each processor takes a turn broadcasting the data word in a round-robin fashion. The data words sent through the crossbar are 0000 (hex), 5555, AAAA, and FFFF. The test will run in a continuous loop until it is stopped manually. Test results may be sent to the printer or a disk file.

In order to run the test, follow these steps:

1) Sign on the Intel 310 using the logon name "pfp" and the password "kirk".

2) Move to the test directory by typing "cd test/T3".

3) Type "submit T3" to run the test. The terminal screen will come up with a display indicating that the test is running. The test will run in an infinite loop until it is manually stopped.

4) To stop the test, type a "control C" at the terminal.

4.2.3 Sequencer Handshake Test

The Sequencer Handshake Test checks the sequencer handshake circuitry to each processor, the handshake circuitry on each processor, the sequencer/processor transition board, and associated cabling. The sequencer/processor handshake circuits and associated cabling are checked on an individual basis. The test sets up processor 0 as a sending processor. Processor 0 sends a data word (all 0's) to processor 1 and the host reads it back. Processor 0 then sends the word to processor 2 and the host reads it. Processor 0 sends to processor 3, etc. A screen display is output during the test to indicate the test is running. If the test hangs up on a faulty circuit or connection, it can be identified as the last transfer displayed on the screen. The test runs in an infinite loop until it is stopped manually.

In order to run the test, follow these steps:

1) Sign on the Intel 310 using the logon name "pfp" and the password "kirk".

- 2) Move to the test directory by typing "cd test/T2".
- 3) Type "submit T2" to run the test. The terminal screen will come up with a display indicating that the test is running. The test will run in an infinite loop until it is manually stopped.
- 4) To stop the test, type a "control C" at the terminal.

4.2.4 Crossbar and Sequencer Memory Test

The Crossbar and Sequencer Memory Test checks all crossbar and sequencer memory, the sequencer/crossbar interface board, and associated cabling. The host writes data to all crossbar and sequencer memory locations and then reads them back for verification. The data patterns initially sent are 0000 (hex), 5555, AAAA, and FFFF. Random numbers are then generated at the host, sent out, and read back for verification. The test will run in a continuous loop until it is stopped manually. Test results may be sent to the printer or a disk file. The test may be run on the sequencer memory individually, the crossbar memory individually, or both sequencer and crossbar combined.

In order to run the test, follow these steps:

- 1) Sign on the Intel 310 using the logon name "pfp" and the password "kirk".
- 2) Move to the test directory by typing "cd test/xm".
- 3) Type "submit xm(s)" to run the test on just the sequencer, "submit xm(x)" to run the test on just the crossbar, or "submit xm(b)" to run the test on both. The terminal screen will come up with a display indicating that the test is running. The test will run in an infinite loop until it is manually stopped.
- 4) To stop the test, type a "control C" at the terminal.

4.2.5 Hardware Assisted Function Test

Special "piggyback" boards are available to speed the calculations of certain functions through the use of hardware. Exponentials, square roots, natural logarithms, reciprocals, sines, cosines, tangents, arcsines, arccosines, and arctangents are all supported. This test has each processor compute each function over a given range and then compares the answers from each processor to known values stored at the host. At the present time, this test only supports the GT-FPP/3 processor. The test will run in a continuous loop until it is stopped manually. Test results may be sent to the printer or a disk file.

In order to run the test, follow these steps:

- 1) Sign on the Intel 310 using the logon name "pfp" and the password "kirk".

2) Move to the test directory by typing "cd test/function".

3) Type "submit function" to run the test. The terminal screen will come up with a display indicating that the test is running. The test will run in an infinite loop until it is manually stopped.

4) To stop the test, type a "control C" at the terminal.

4.3 Diagnostic Tools

4.3.1 Digital Oscilloscope

A Hewlett-Packard digitizing oscilloscope, model 54201D, or equivalent is recommended for on-site debugging of problems encountered when running system diagnostics.

4.3.2 Digital Multimeter

A Fluke digital Multimeter, model 77, or equivalent is recommended for on-site debugging of problems encountered when running system diagnostics.

4.3.3 Logic Analyzer

An NWIS Microanalyst Logic Analyzer, model 2200 , or equivalent, is recommended for on-site debugging of problems encountered when running system diagnostics.

5 NOTES

5.1 Acronym Glossary

CSOM - Computer System Operator's Manual

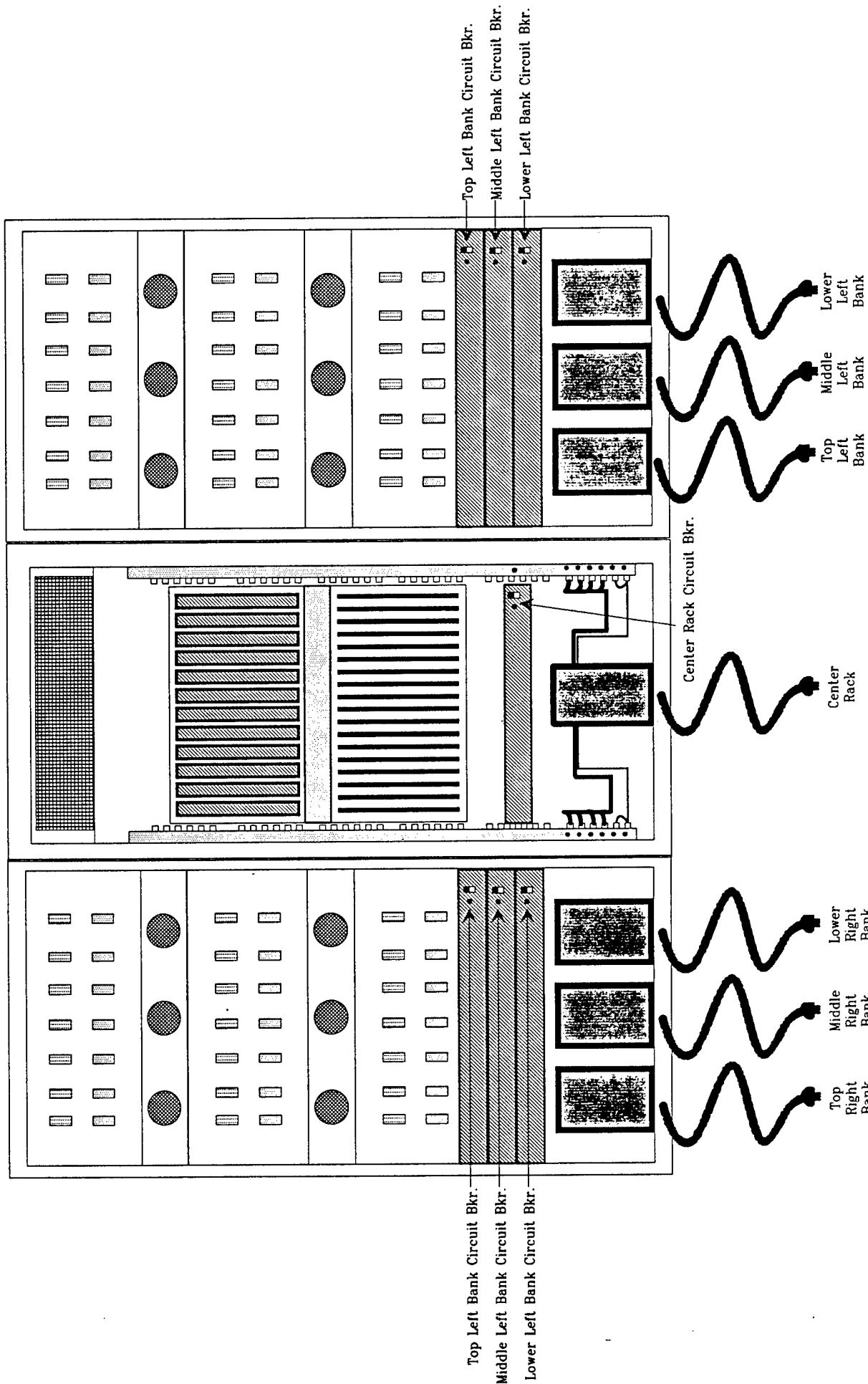
LED - Light Emitting Diode

PFP - Parallel Function Processor

APPENDIX A FIGURES

Figure 3.1-1 Location of Ckt Breakers & Pwr Cables

Figure 3.1-2 Location of Power Switches



(Rear view of system)

Figure 3.1-1 Location of Circuit Breakers and Power Cables

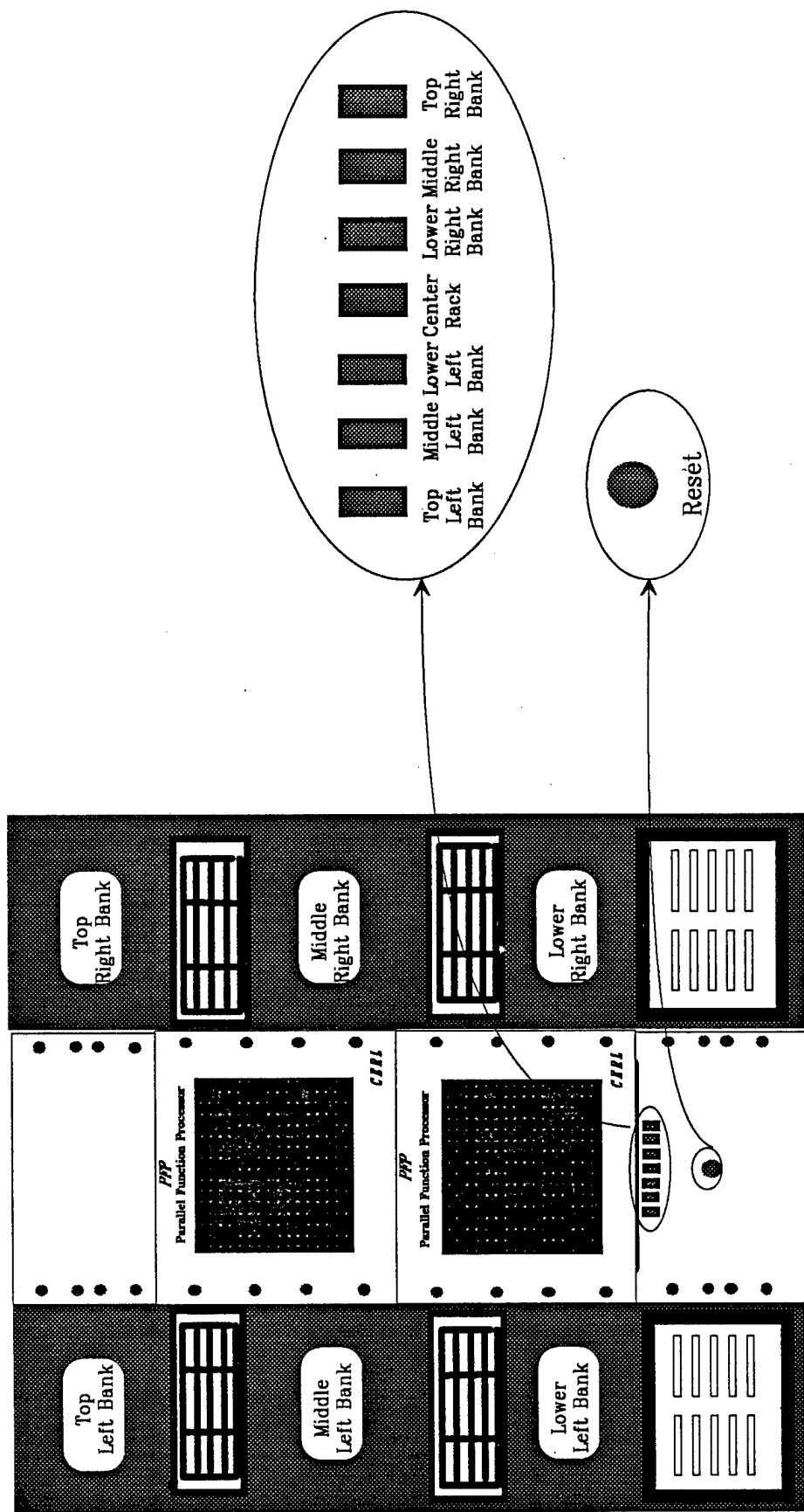


Figure 3.1-2 Location of Power Switches

APPENDIX B DEBUG EXAMPLES USING PFP DISPLAYS

There are 5 major steps involved in a crossbar transfer. They are:

1. The sequencer must set the appropriate data transfer paths in the crossbar. The paths being used can be identified by reading the crossbar display.
2. The sequencer asserts ROT/ (read out) on all processors that are sending data on that cycle.
3. The Sequencer waits to see DAV/ (data available) from all sending processors and RFI/ (ready for input) on all processors receiving on that cycle. Status lines from processors not involved in that cycle are ignored.
4. The sequencer asserts WIN/ on all receiving processors.
5. The sequencer negates the active ROT/ and WIN/ lines to terminate the cycle.

Example 1 Two Sends Each With Multiple Receivers

Example 1 has processor P1 sending to processors P16, P18, P19, and P21 while processor P23 sends to P0, P4, and P8.

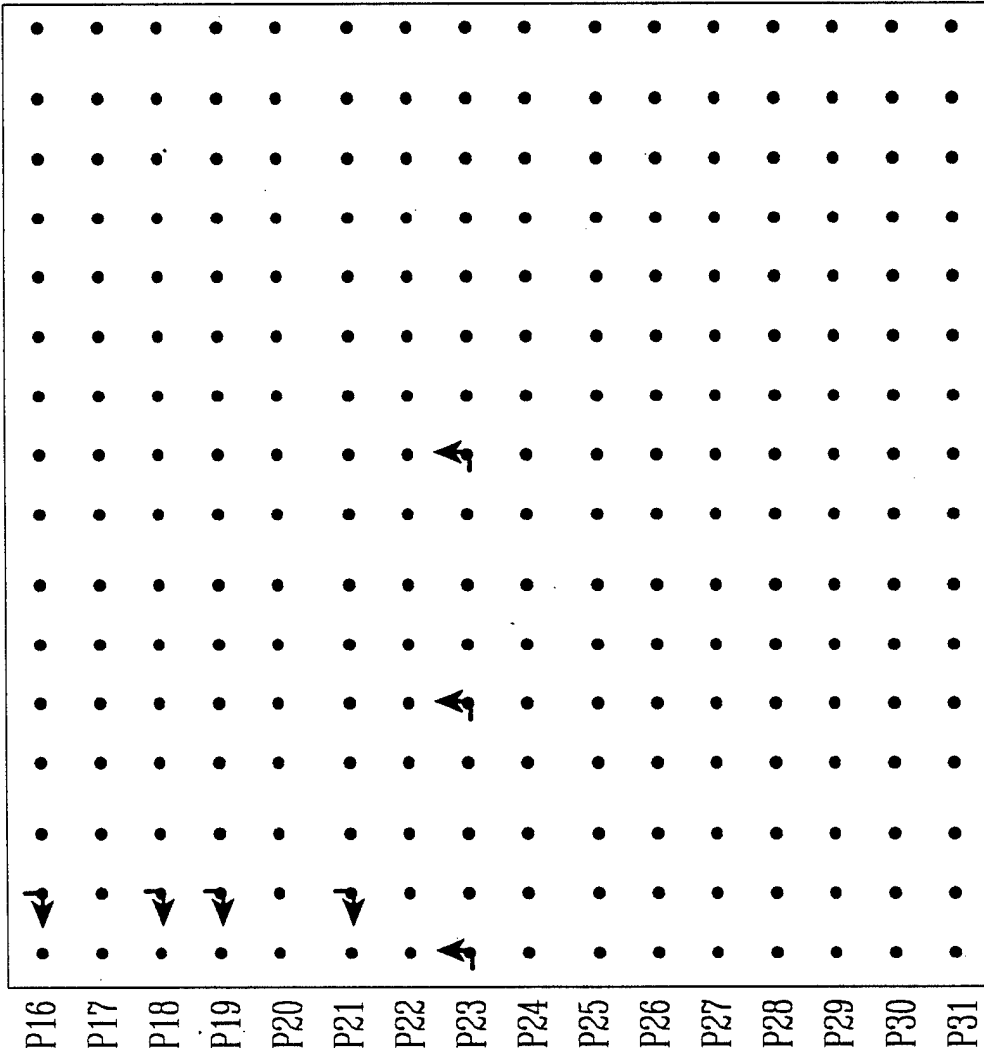
The five transfer steps are shown on the next 5 figures. They are:

1. The sequencer sets up the crossbar data paths.
2. The sequencer asserts ROT/ on the sending processors (P1 and P23).
3. The Sequencer waits to see DAV/ (data available) from all sending processors (P1 and P23) and RFI/ (ready for input) on all processors receiving on that cycle (P0, P4, P8, P16, P18, P19, and P21). Status lines from processors not involved in that cycle are ignored.
4. The sequencer asserts WIN/ on all receiving processors (P0, P4, P8, P16, P18, P19, and P21).
5. The sequencer negates the active ROT/ (P1 and P23) and WIN/ (P0, P4, P8, P16, P18, P19, and P21) lines to terminate the cycle.

PFEP

PARALLEL FUNCTION PROCESSOR

P0 P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15



CYCLE

P16,P18,P19,P21 := P1
P0,P4,P8 := P23

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off

Example 1 Step 2

$\overline{\text{DAV}}$	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off
 Example 1 Step 3

$\overline{\text{DAV}}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	○	●	●	○	○	●	○	○	○	○	○	○	○	○	○	○
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

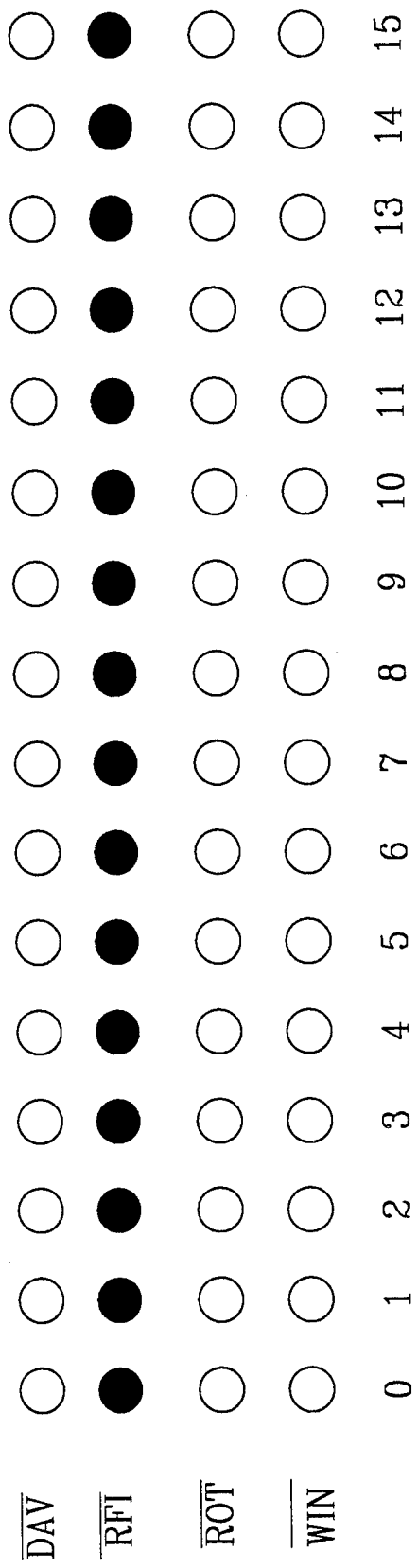
LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

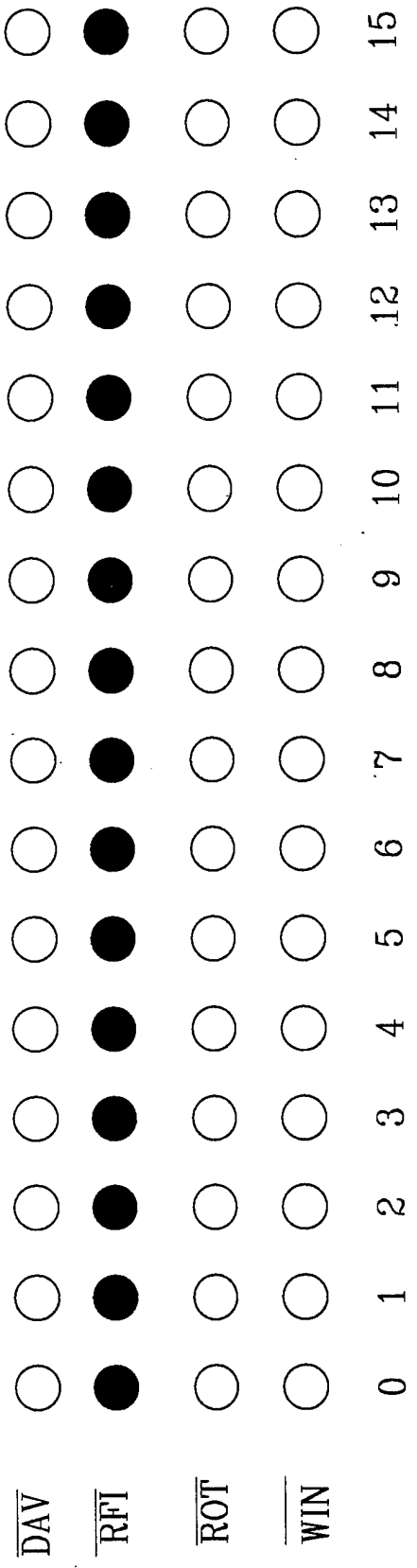
LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off

Example 1 Step 4



LEDS FOR P0 THROUGH P15



LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off

Example 1 Step 5

Example 2 One Processor Broadcasting to 31 Others

Example 2 shows processor P17 broadcasting a data word to the other 31 processors on the crossbar. All 31 receive the same word at the same time.

The five transfer steps are shown on the next 5 figures. They are:

1. The sequencer sets up the crossbar data paths.
2. The sequencer asserts ROT/ on the sending processor (P17).
3. The Sequencer waits to see DAV/ (data available) from all sending processors (P17) and RFI/ (ready for input) on all processors receiving on that cycle (all others).
4. The sequencer asserts WIN/ on all receiving processors (all but P17).
5. The sequencer negates the active ROT/ (P17) and WIN/ (all others) lines to terminate the cycle.

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off

Example 2 Step 2

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off

Example 2 Step 3

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			

LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	●	○	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15			

LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off

Example 3 Step 4

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off

Example 2 Step 5

Example 3 Five Simultaneous Sends With Multiple Receives

Example 3 shows processor P5 sending to processors P25, P26, P27, and P28, while processor P2 sends to P24, processor P23 sends to P3, processor P22 sends to P21, and processor P12 sends to P7, P10, and P15.

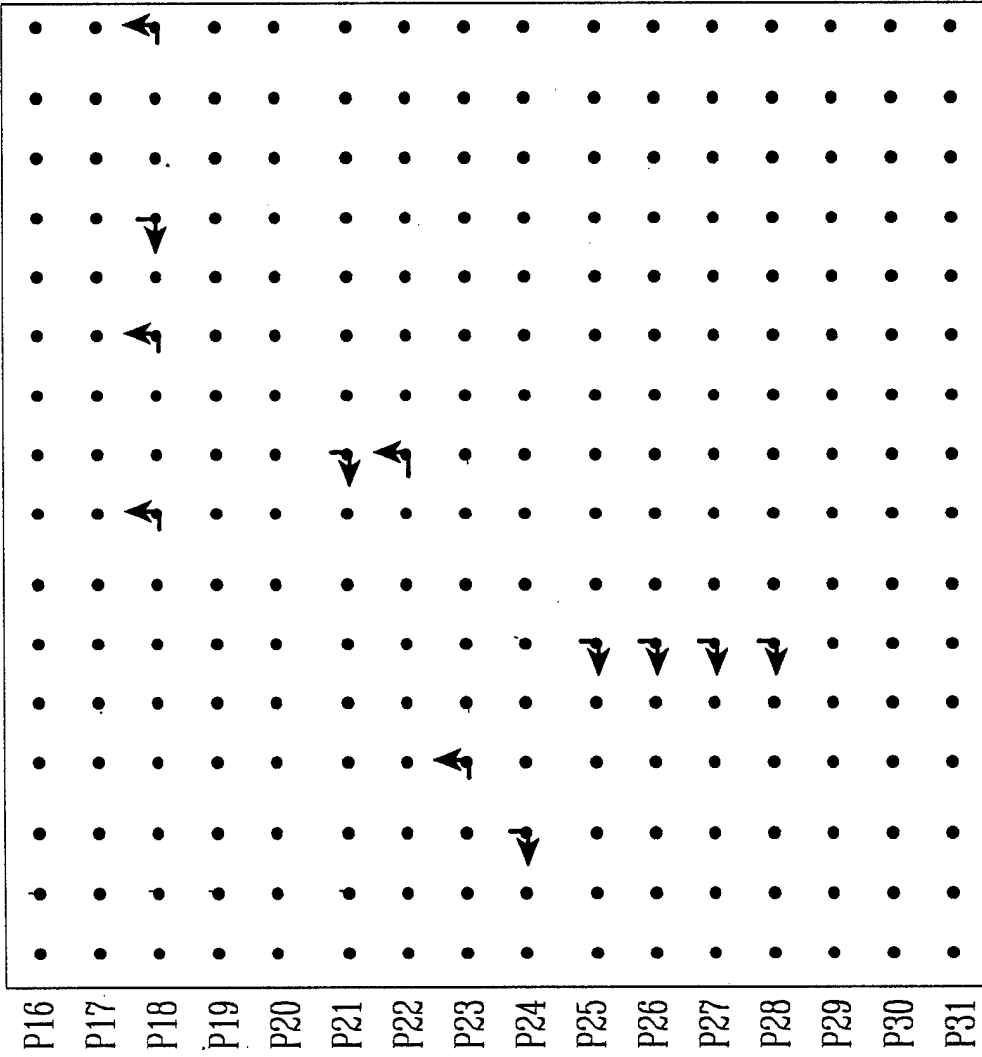
The five transfer steps are shown on the next 5 figures. They are:

1. The sequencer sets up the crossbar data paths.
2. The sequencer asserts ROT/ on the sending processors (P2, P5, P22, P23, and P12).
3. The Sequencer waits to see DAV/ (data available) from all sending processors (P2, P5, P22, P23, and P12) and RFI/ (ready for input) on all processors receiving on that cycle (P3, P7, P10, P15, P21, P24, P25, P26, P27, and P28).
4. The sequencer asserts WIN/ on all receiving processors (P3, P7, P10, P15, P21, P24, P25, P26, P27, and P28).
5. The sequencer negates the active ROT/ (P2, P5, P22, P23, and P12) and WIN/ (P3, P7, P10, P15, P21, P24, P25, P26, P27, and P28) lines to terminate the cycle.

PFP

PARALLEL FUNCTION PROCESSOR

P0 P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15



CYCLE

P25,P26,P27,P28 := P5
 P24 := P2
 P3 := P23
 P21 := P22
 P7,P10,P15 := P12

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off

Example 3 Step 2

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	

LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	

LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off
 Example 3 Step 3

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	

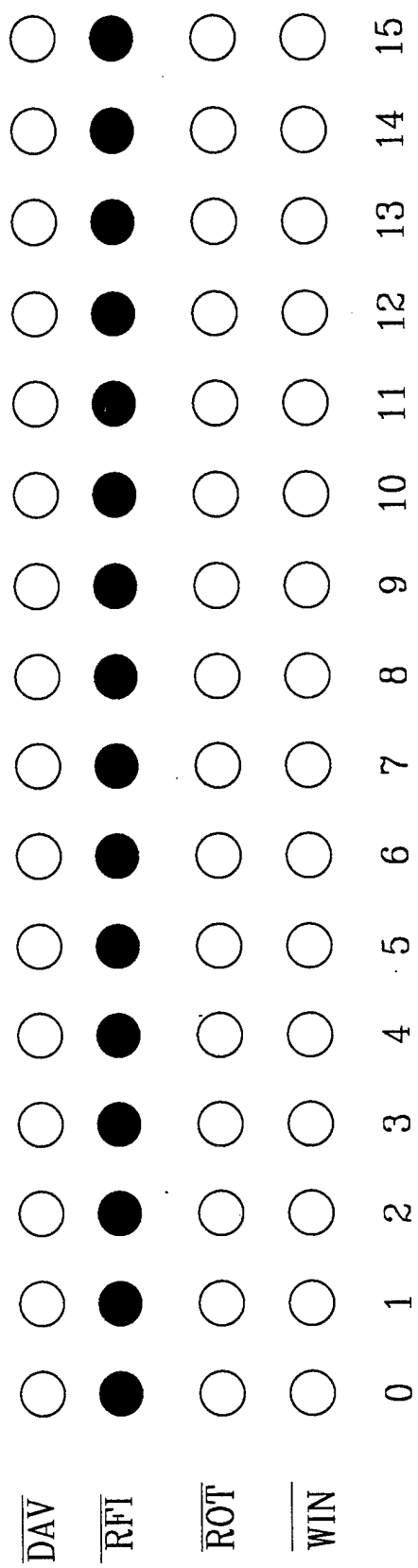
LEDS FOR P0 THROUGH P15

$\overline{\text{DAV}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{RFI}}$	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{ROT}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
$\overline{\text{WIN}}$	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	

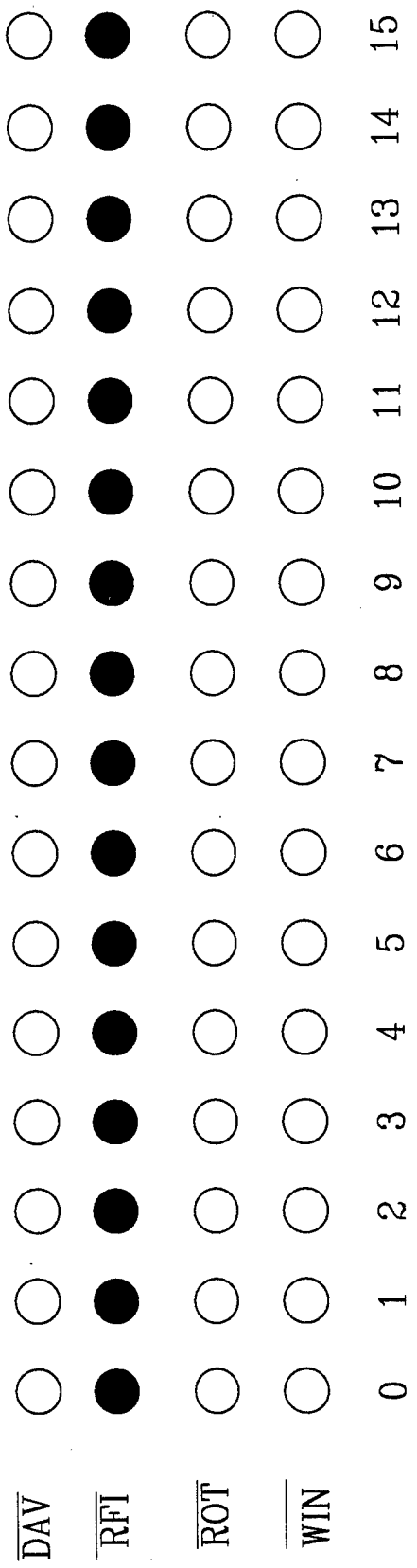
LEDS FOR P16 THROUGH P31

○ LED is on ● LED is off

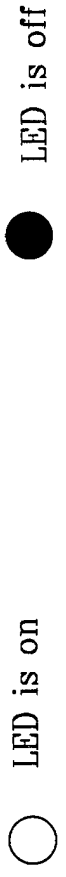
Example 3 Step 4



LEDS FOR P0 THROUGH P15



LEDS FOR P16 THROUGH P31



Example 3 Step 5