This memo describes the Janus Boot operation using HIPO (Hierarchy-Input-Process-Output) diagrams. These diagrams will describe in general what will occur and in what order, and will show the firmware/software structure planned for implementation. Two types of diagrams will be used:

1. Functional diagrams showing the processes performed and their ordering, and

2. Structure diagrams showing the interrelationships between identifiable firmware modules.

A list of the diagrams is presented below:

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Boot Operation, HIPO Functional Diagram</td>
</tr>
<tr>
<td>2</td>
<td>EPROM Microdiagnostic, HIPO Functional Diagram</td>
</tr>
<tr>
<td>3</td>
<td>System Initialization Microdiagnostic, HIPO Functional Diagram</td>
</tr>
<tr>
<td>4</td>
<td>System Initialization Firmware, HIPO Functional Diagram</td>
</tr>
<tr>
<td>5</td>
<td>EPROM Microdiagnostic, HIPO Structure Diagram</td>
</tr>
<tr>
<td>6</td>
<td>System Initialization Microdiagnostic, HIPO Structure Diagram</td>
</tr>
<tr>
<td>7</td>
<td>System Initialization Firmware, HIPO Structure Diagram</td>
</tr>
</tbody>
</table>

The majority of the Boot operation will be controlled by two packages stored on the system software storage media:

* System Initialization Microdiagnostic
A microdiagnostic designed to test system electronics without requiring user intervention, and

* System Initialization Firmware

Firmware designed to establish Mesa emulation and initialize the system for Pilot.

These two packages will run in sequence after the EPROM Microdiagnostic has been completed successfully. If critical errors are encountered, the Boot operation will be aborted (halted). Critical errors are those errors that would negate or seriously impair subsequent processing by Pilot and application software.

The remainder of this memo will be presented according to the following table of contents. It will describe those modules identified in the HIPO Structure Diagrams.
Janus Boot Operation

I  EPROM Microdiagnostic
II  System Initialization Microdiagnostic
   II.1  Microdiagnostic Controller
   II.2  Processor/Control Store Test
   II.3  Extended Processor Test
   II.4  Main Memory/Memory Map Test
   II.5  Extended Memory Test
   II.6  Hardware Configurator
   II.7  Serial Interface Test
   II.8  Rigid Disk Test
   II.9  Floppy Disk Test
   II.10  Full Page & 1/4 Page Display Unit Test
   II.11  Magnetic Card Unit Test
   II.12  Cassette Tape Unit Test
   II.13  Xerox Wire/RS232C Test
III  System Initialization Firmware
   III.1  Emulator Load/Verification
   III.2  Memory Map Load/Verification
   III.3  Machine Parameter Initialization
   III.4  Pilot ILM Load/Verification
I EPROM Microdiagnostic

This microdiagnostic is permanently a part of the processor hardware and will be used to initiate a Boot operation from an appropriate load device containing the system software media.

A Boot operation will commence when one of the following occurs:

1. Power-up,
2. Hardware reports a TYPE 3 error,
3. Watchdog timer runout,
4. An operator pushes the START button, or
5. Software initiates a Boot operation.

When a Boot operation is initiated by one of these actions, the contents of a EPROM memory (currently 512 words) will be loaded into the first portion of the control memory and control transferred to it. This is referred to as the EPROM Microdiagnostic. This microdiagnostic will be responsible for initiating the loading and verifying of the System Initialization Microdiagnostic.

II System Initialization Microdiagnostic

This microdiagnostic will receive control from the EPROM bootstrap microdiagnostic after it has completed successfully. It will perform the following major functions:

1. Verify the operability of the applicable hardware hardcore, i.e.,
   a. processor and main memory,
   b. data path from the load device and the control and main memories, and
   c. the microcode loaded from the load device.
2. Test the remainder of the system electronics without invoking user intervention, and
3. Provide fault isolation to a minimum number of FRUs, where possible.

The major goal of this microdiagnostic will be to detect and isolate a maximum number of hardware failures.

II.1 Microdiagnostic Controller

This firmware will control Boot operation activities while the System Initialization Microdiagnostic is active. It will perform the following functions:
1. Select the hardware devices and microdiagnostic tests to be tested and run.

2. Support an interface to the system disk so that the contents of the control store can be overlayed, where necessary, and new programs and data can be retrieved.

3. Determine when a system has no available K/D unit (full page or 1/4 page display unit), or when all available K/D units have encountered errors, and take appropriate action (e.g., present a reserved code in the D0 Maintenance Panel indicating these conditions), and

4. Will load and initially verify the System Initialization Firmware control program.

II.2 Processor/Control Store Test

This test will complete verification of the processor and control store. It will be performed immediately after the Microdiagnostic Control Program is loaded, and will detect solid and, where feasible, intermittent hardware failures.

II.3 Extended Processor Test

This test will be executed only if:

1. Intermittent errors have been encountered during the Processor/Control Store Test, or

2. When directed by a user after Pilot and application software have been established.

The major goal of this test is to provide, where possible, fault isolation to a small number of components on individual processor FRUs.

II.4 Main Memory/Memory Map Test

This test will perform main memory testing in a manner that will minimize the amount of time expended in testing the main memory and main memory map. Extensive main memory and main memory map testing will be performed by the Extended Memory Test.

This test will perform the following functions:

1. Writing a fixed test pattern in all of main memory and read/verifying that pattern looking for the occurrence of multiple memory failures.

2. Verifying the main memory map by an memory access test.
II.5 Extended Memory Test

This test will perform the following functions:

1. Random and worst-case pattern main memory testing,
2. Extensive main memory map tests, and
3. Main memory chip failure isolation.

This test will be executed when:

1. During a Boot operation intermittent errors are encountered during execution of the Main Memory/Memory Map Test,
2. Directed by a user after Pilot and the application software have been established.

II.6 Hardware Configurator

This routine will establish and maintain the run-time configuration used during the Boot operation. It will supply and maintain information on hardware devices tested and untested, and those found to be faulty.

II.7 Serial Interface Test

This test will Poll all devices on the serial interface and execute simple "turnaround" tests where available. The goal of this test is to determine if the serial interface can be used by subsequent tests without error.

II.8 Rigid Disk Test

This test will perform head positioning tests on the rigid disk, and limited data storage and retrieval testing to verify the ability of the device/controller to correctly access storage areas on the disk, and reliably transfer and maintain data.

II.9 Floppy Disk Test

This test will perform head positioning tests on a floppy disk, and will verify the ability of the device/controller to transfer data from the storage media to memory without error. No attempt will be made to write on the peripheral storage media.

II.10 Full Page & 1/4 Page Display Unit Test
Janus Boot Operation

This test will utilize hardware diagnostic features, specifically those included in the keyboard, to determine if data can be sent to and received from a full page or 1/4 page display unit.

II.11 Magnetic Card Unit Test

This test will perform head positioning tests on a magnetic card unit, and will verify the ability of the device/controller to transfer data from the storage media to memory without error. No attempt will be made to write on the peripheral storage media.

II.12 Cassette Tape Unit Test

This test will perform device electronics tests, where appropriate, to determine if solid failures are present. No attempt will be made to move the storage media.

II.13 Xerox Wire/RS232C Test

This firmware will test the Xerox Wire and RS232C interface hardware to the extent possible without requiring user intervention, avoiding possible interaction problems with other connected systems.

III System Initialization Firmware

This package will receive control from the system initialization microdiagnostic only after it has completed without encountering critical errors. Control will be transferred to Pilot only after:

1. The memory map has been loaded and verified,
2. The Mesa emulation has been established, and
3. The initial Pilot load module has been loaded and verified.

III.1 Mesa Emulator Load/Verification

The Mesa emulation microcode will be loaded from the system disk and verified (checksummed) to ensure that it was loaded correctly.

III.2 Memory Map Load/Verification

The main memory map will be established for use by Pilot. A short verification of the map will ensure that solid map parity errors are not detected by the hardware.
III.3 Machine Parameter Initialization

This will be written in microcode and will establish all parameters expected by *Pilot* that are settable.

III.4 Pilot ILM Load/Verification

The *Pilot* ILM (Initial Load Module) will be loaded and verified (checksummed). This ILM will then be responsible for bringing in the rest of *Pilot*. Additionally, data accumulated during the Boot operation will be loaded into specific main memory locations so that the data can be accessed by *Pilot*. 
Distribution:

D. Stottlemyre
P. Jarvis
E. Reber
C. Hankins
W. Kennedy
J. Weaver
Input | Process | Output
--- | --- | ---
Power-up | After hardware has loaded the 512 location EPROM memory into the control store, perform: | DO Maintenance Panel
TYPE 3 Error | 1. Run EPROM Microdiagnostic | 
Watchdog Timer | 2. Run System Initialization Microdiagnostic | 
START Button Pushed | 3. Run System Initialization Firmware | 
Software-Initiated Boot | Note: HALT on critical errors | Pilot Initial Load Module

Notes:

1. Upon the occurrence of any of the five conditions listed above, the hardware will load into the control store the contents of a 512 location EPROM memory (EPROM Microdiagnostic).

2. The System Initialization Microdiagnostic and the System Initialization Firmware are resident on system load device.

3. The boot-time firmware will test all attached and available electronic hardware that does not require user intervention. Hardware errors encountered will be classified as either critical or non-critical. Critical errors are those that will negate or seriously impair the ability of the system to perform user processing. Non-critical errors are those that can exist and still allow some level of user processing. Critical error will result in an ABORT of the boot operation. Non-critical errors will be reported to Pilot at the end of the boot operation.

Figure 1. Boot Operation, HIPO Functional Diagram
Notes:

1. The major goal of this microdiagnostic is to check the load path between the control store and the system load device so that the System Initialization Microdiagnostic can be loaded without errors.
Input
- Power-up
- TYPE 3 Error
- Watchdog Timer
- START Button Pushed
- Software-Initiated Boot

Process
- EPROM Microdiagnostic
  1. Test internal data paths and ability to branch and test.
  2. Test ability to write & read control store
  3. Test ability to access rigid disk
  4. If OKAY, initiate read; else, initiate read from alternate device (floppy disk); if not available, HALT.
  5. Verify data read into the control store; if in error, HALT
  6. Test for correct load module; if incorrect, not the proper boot module, so HALT

Output
- DO Maintenance Panel

System Initialization Microdiagnostic

Notes:

1. The major goal of this microdiagnostic is to check the load path between the control store and the system load device so that the System Initialization Microdiagnostic can be loaded without errors.

Figure 2. EPROM Microdiagnostic, HIPO Functional Diagram
Notes:

1. At this point the 'boot path' should have been tested enough for solid hardware failures that it can be relied upon in further load operations. The firmware loaded will also have been checked for data checksum errors, making it reasonable to assume that the previous operations have been performed correctly.
System Initialization Microdiagnostic

1. Perform testing on the serial interface; CONTINUE on error
2. Test all magnetic card devices; CONTINUE on error
3. Test all cassette tape devices; CONTINUE on error
4. Test all printers; CONTINUE on error

(see note 2)

System Initialization Firmware

Notes:

2. At this point all the electronics testable at boot-time have been tested. The remainder of the boot operation will be concerned with machine initialization and the establishment of Mesa emulation and Pilot.

3. No user intervention has been required up to this point, and the only information displayed has been placed in the D0 Maintenance Panel. This will continue to be true until Pilot has established.

Figure 3. System Initialization Microdiagnostic, HIPO Functional Diagram (Continued)
Input Process Output

System Initialization
Microdiagnostic

<table>
<thead>
<tr>
<th>Sys. Init. Firmware</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Load main memory map and report physical memory configuration</td>
</tr>
<tr>
<td>2. Initialize machine parameters</td>
</tr>
<tr>
<td>3. Load and verify (checksum) Mesa emulator</td>
</tr>
<tr>
<td>4. Load and verify (checksum) Pilot initial load module</td>
</tr>
</tbody>
</table>

Output

DO Maintenance Panel

Pilot Initial Load Module

Notes:

1. Data obtained during the boot operation will tentatively be stored in main memory in locations known to Pilot. The amount of information so saved will be limited to indicating the cause of the boot operation, summary information on non-critical errors encountered (not: critical errors will cause the boot operation to HALT), and main memory map data.

Figure 4. System Initialization Firmware, HIPO Functional Diagram
Hardware-Controlled
Start

Verify Branching Capability

Checksum EPROM
Data Loaded

Verify remaining
Control Store

Verify Internal
Data Paths

Determine System
Load Device

Initiate Load Operation

System Initialization
Microdiagnostic

Figure 5. EPROM Microdiagnostic, HIPO Structure Diagram
Notes:
1. The initial load module from the rigid disk or alternate load device is considered to be part of the Microdiagnostic Controller

2. Each routine that must display a code in the D0 Maintenance Panel will be responsible for loading that code

Figure 6. System Initialization Microdiagnostic, HIPO Structure Diagram
Mesa Emulator
Load/Verification

Memory Map
Load/Verification

Machine Parameter
Initialization

Pilot ILM
Load/Verification

Legend:

ILM - Initial Load Module

Figure 7. System Initialization Firmware, HIPO Structure Diagram