

IIIIII	PPPPPPPP	LL
IIIIII	PPPPPPPP	LL
II	PP PP	LL
II	PP PP	LL
II	PP PP	LL
II	PP PP	LL
II	PPPPPPPP	LL
II	PPPPPPPP	LL
II	PP	LL
II	PP	LL
II	PP	LL
II	PP	LL
IIIIII	PP	LLLLLLLLLL
IIIIII	PP	LLLLLLLLLL

LL	PPPPPPPP	TTTTTTTTTT	777777777
LL	PPPPPPPP	TTTTTTTTTT	777777777
LL	PP PP	TT	77
LL	PP PP	TT	77
LL	PP PP	TT	77
LL	PP PP	TT	77
LL	PPPPPPPP	TT	77
LL	PPPPPPPP	TT	77
LL	PP	TT	77
LL	PP	TT	77
LL	PP	TT	77
LL	PP	TT	77
LLLLLLLLLL	PP	TT	77
LLLLLLLLLL	PP	TT	77

START Job IPL Req #114 for WAW Date 23-Jul-82 21:36:58 Monitor: Rational Ma
 File RM:<WAW.IPL>IPL.LPT.7, created: 26-Apr-82 21:49:25
 printed: 23-Jul-82 21:36:58
 Job parameters: Request created:23-Jul-82 21:29:31 Page limit:45 Forms:NORMAL
 File parameters: Copy: 1 of 1 Spacing:SINGLE File format:ASCII Print mode:A

Power Up, Initial Program Load and System Initialization

DRAFT 1

by
Walter A. Wallach

Rational Machines proprietary document.

1. Summary

This document discusses the power up, initial program load and system initialization procedures and mechanisms for the R1000. These procedures include power up sequencing, initial bootload of one IOP, remote bootload of the other IOPs and the CPUs, running system verification diagnostics and error reporting and reconfiguration, peripheral power up, remote loading of system initialization firmware and software, and system initialization. This document also discusses operator communications interface for normal system environments, operator interface for diagnostic procedures, error logging and system performance statistics and environment display.

The reader is assumed familiar with the R1000 system concepts and system topology issues.

2. System Configurations

An R1000 cluster consists of one or more R1000 processors (CPUs), one or more input/output processors (IOPs), one or more fixed media moving head disks for paging support, one or more magnetic tape devices and one or more communications controllers with support for from 8 to 128 asynchronous communications lines. Support for a line printer (600-900 lines per minute) may also be included.

All I/O devices connect to an IOP. The IOPs and CPUs communicate over the system bus, a high speed word-oriented message bus. The IOPs provide driver support software, request queueing, error handling and interrupt buffering for the I/O devices, and offer a high level task-oriented interface to tasks running on the CPU.

Each IOP connects to the system bus via an I/O adapter card (IOA).

2.1. Power Distribution

The R1000 cluster is subdivided into several power distribution groups, each with its own power supply:

- CPU group, consisting of the CPU and memory cards and the sysbus adapter card for a single R1000 processor. There may be up to four of these groups in a single cluster, each with six processor boards and from two to four memory boards and a power supply.
- IOP group, consisting of a PDP11/24 processor and four slot expansion backplane, 256KB of MOS memory, device controller cards.
- Peripheral group, consisting of exactly one disk, tape, line

printer or terminal device. Each of these units has its own power supply, and certain of these may require manual intervention by a human operator for power up (this also provides the alternative of powering down such devices when they are not in use). Presently, only the disk units are automatically powered up as part of the IPL procedure, and these are powered one at a time. Tape units remain powered off until an operator manually resets the unit. Terminals will power up depending upon the setting of their individual power switches.

- IOA and Power Sequencer group, the final power distribution group type, consists of all IOAs (UNIBUS and sysbus sides), a power supply and cabling to the remote power on connector of each IOP.

The sysbus must be designed so that it will work even though some of the devices on it are not powered up.

2.2. Powerup Scenario

Powerup begins with an operator actuating a single power up switch. This switch powers the IOAs and Power Sequencer group. The IOA in slot zero is defined to be the master, and the IOP it is connected to is termed the master IOP. This IOP controls IPL for the rest of the cluster.

2.2.1. Communications between IOPs

Each IOA card has an 8031 microprocessor which provides a data channel between the console ports of the IOPs (via the diagnostic bus). The 8031 on the master IOA connects the diagnostic bus to the secondary serial console port (SLU2) of the master IOP. For the remaining IOPs, their 8031s connect their primary serial console port (SLU1) to the diagnostic bus. This is what makes an IOA a master: the UNIBUS side of the adapter is cabled to SLU2 while the system console is cabled to SLU1. The other end of this IOA is plugged into IOA slot zero. This interface is described in the section on special hardware features. The master uses this link to send commands to the remote IOPs, and to receive status and error reports from them.

These 8031s are in addition to the 8051 diagnostic processor, which is also connected to the diagnostic bus. The 8031s provide communications between IOPs. Those on the nonmaster IOAs implement a dedicated channel to the master, while the one on the master IOA is polled for inter-IOP messages, which are buffered and fed to SLU2.

2.2.2. IPL of the Master IOP

All 8031s run a small diagnostic program, then the master 8031 powers up its IOP and instructs it to IPL (how, since the system console is SLU1, but this 8031 is connected to SLU2?). The system is always boot loaded from the system disk device, which is one of the disks connected to the master IOP.

When powered, the master IOP performs a bootload from its system disk device (normal boot load can be done only from a disk device). A failed system may be diagnosed by manually powering any selected group and boot loading a diagnostic program or microprogram. This procedure is also controlled by a master IOP, but this IOP may be boot loaded from any mass storage unit (disk or tape). These procedures are beyond the scope of this document.

2.2.3. Master IOP

Once the master IOP has been powered, it runs a small verification diagnostic program to determine if it should continue with IPL. If this diagnostic fails, the operator is informed that he must select another IOP to load from. The master IOP then determines that it is the only IOP powered. This is done by reading a power monitor register in its IOA (on the sysbus side). This register contains one bit for each group which may be remotely powered by the master IOP.

The master IOP bootloads the first part of the system initialization program from its system disk device. The purpose of this software is to power the remainder of the system and initiate system verification diagnostics within each group.

First, the master IOP runs the system verification diagnostic on itself. Upon passing this, it powers up its remaining peripheral units. It then powers up each of the other IOPs by setting the appropriate bit in the power register (a companion to the power monitor register introduced above).

2.2.4. Nonmaster IOPs

As each remaining IOP is powered, it remains halted. The master loads a program image of the IOP verification diagnostic via the sysbus and IOA into each of the remaining IOPs. These IOPs are then instructed to start via the 8031 link to their console port (this takes advantage of the soft console support of the PDP11/24). Each remaining IOP then independently runs its verification program and, upon passing, powers up its peripheral units, then awaits the next program image from the master. If it fails the system verification diagnostic, it is declared incapable of executing, and a later system initialization program logically reconfigures to remove that IOP (this reconfiguration support is probably not included in the initial

release, since it involves logically removing any virtual processors whose address spaces are stored on disks connected to the failed IOP).

2.2.5. IOP Initialization

The master IOP loads a program image of initialization software, the normal IOP service software and the RSX11S kernal image via the sysbus and IOA. This program is structured such that, after initialization, the memory used by the initialization code is reclaimed as buffer space.

During this phase, all devices are initialized, data structures, request queues and device drivers initialized and error reporting channels are established. When this phase is complete, these IOPs are able to receive and service requests in the normal way.

2.2.6. CPU Power and Loading

While the other IOPs are running their verification programs, the master IOP powers each R1000 processor, and loads a system verification microdiagnostic program via the diagnostic bus. Each processor in the cluster independently runs the verification diagnostic, reporting any errors to the master IOP. As for the nonmaster IOPs, if an error is encountered, that CPU is logically removed from the cluster. Removing a CPU at IPL time is easier than removing an IOP, since no VPs need be removed.

2.2.7. CPU Initialization

Each CPU is initialized by loading the initialization firmware (via the diagnostic bus), then loading the virtual memory pages required for system initialization (via the sysbus). The tag stores are initialized. The memory management data base initial contents are sent using a special sysbus message, which loads the pages (they are prewired in the data base, so they will never be displaced). All wired code and module pages are loaded and locked into memory. The initialization firmware is replaced by normal firmware, and the initialization software is executed. Finally, the initialization modules terminate and are deleted from memory, and the CPU is ready to begin processing.

2.2.8. Assigning Virtual Processors to CPUs

The default assignment of VPs to CPUs is included in the initial state loaded into the CPU during CPU initialization. These defaults may be superceeded if one of the CPUs is found to be defective, or if the operator explicitly requests an alternate configuration. The master IOP prompts the operator before continuing with VP assignment.

2.2.9. Initialization of the Master IOP

After all processors and IOPs have been initialized, the master loads its initialization code, normal service code and RSX11S over the IPL software and begins its initialization process. When initialized, it continues to serve as system control processor, in that it owns the system console device and receives all error messages from the other IOPs (see the discussion of the diagnostic bus in a later section).

2.2.10. Recovery of Closed Tasks

After the system has been initialized, the medium term scheduler on each CPU attempts to recover modules which may be restarted. This is done by interrogating the file system on each CPU and paging in the control stack header page of each restartable module. These modules are scheduled and, when they begin executing, demand load the remaining virtual memory pages they need to run. The R1000 scheduler includes some form of load control which limits the number of modules actively loading pages during module initialization transients.

3. Special Hardware Considerations

Each IOA contains the two registers introduced above, the power monitor and power registers. The standard boot PROM is used on the IOA to handle IPL.

3.0.1. Communications between IOPs

Each IOA includes an 8031 microprocessor which interfaces the diagnostic bus to the IOP's serial interface. For the nonmaster IOPs, the 8031 interfaces SLU1 to the diagnostic bus, while for the master IOP, SLU2 is interfaced. These serial interfaces are run at 19.2K baud, with the 8031 providing buffering of messages to and from the 18.2KByte diagnostic bus. The master IOP's SLU1 is connected to the system console device. These 8031s are in addition to the normal 8051 diagnostic processor included on the IOA card.

The 8031 and serial interfaces provide an alternate path for loading nonmaster IOPs. The soft console feature of the 11/24 is used by the master IOP during remote IOP bootload. The 8031 and serial links are used by the IOPs to send system status information and error reports to the master IOP. Note that, because of the order of magnitude speed difference between the serial lines and the diagnostic bus, messages are limited to 64 bytes, the buffer capacity of the 8031.

3.0.2. Booting Remote IOPs

Software images are transmitted to the other IOPs via the UNIBUS, since all IOPs are connected via their IOAs and the system bus. The nonmaster IOP WAITs until the initial software image is loaded, then begins execution. As each phase completes, the software loops until the next phase of software is loaded.

All IOPs run the standard DEC powerup diagnostic, then halt. The master IOP bootloads from its system disk. The nonmaster IOPs remain halted until started by the master. If a failure occurs in a remote IOP, the IOP is halted by forcing the 11/24 into ODT, the soft console emulator microcode. This is done by emulating a "break" function over SLU1 (this is the normal terminal break function, inverting the transmit line to force a framing error).

With the remote IOP halted, the master does the following, using the remote IOP's ODT function unless otherwise stated:

- Clear the remote IOP's PSW. This clears any funny state the CPU may be in.
- Set the SP away from the area of memory to be DMA loaded, such that stack overflows do not occur.
- Mask all interrupts except priority 7 (NPR).
- Load a WAIT instruction into memory.
- START the remote IOP at the WAIT instruction.
- Use the IOA to DMA a program image into memory.
- Use the break function again to halt the remote IOP.
- START the remote IOP at the proper address.

As each IPL phase completes execution, it HALTs, leaving the remote IOP ready for the next phase to be DMA loaded into its memory.

Software loaded into IOP memory via the sysbus and IOA use the DMA capability of the IOA. Such software is not verified after being loaded, since all busses and interfaces involved are parity checked. Software and firmware loaded via the diagnostic bus is verified prior to being executed.

4. Device Configurations

Each IOP requires a data base reflecting the devices configured to that UNIBUS. These data bases must be dynamic in the sense that IOP software is shipped preconfigured and is not generated in the field for each customer configuration. A data base is built each time the device complement of an IOP changes, and transmitted as part of that IOP's initialization image. The initialization code uses this table to set up the initial request queues and device drivers, initialize the devices and controllers and assign buffers.

5. Operator Interface

The operator's console is an asynchronous terminal device identical to any other terminal except perhaps for speed (the operator's console may run at 19.2 Kbaud). As indicated in a previous section, this device must be accessible to the master IOP for prompting the operator and reporting errors. During IPL, this device looks like and is handled like a line-oriented terminal.

5.1. Normal Operation

During normal system operation, the operator's console contains separate windows for each R1000 CPU, perhaps a window for each IOP, and a window for system messages and operator-entered commands. The processor windows are used to report performance, utilization and scheduling statistics. These windows do not scroll, but are updated periodically with information derived from the R1000 accounting, scheduler and memory manager resource usage counts.

The message and command window scrolls as messages are sent and displayed. This window is logged either to disk or to a hard copy printing device. The statistical windows are not logged.

5.2. System Diagnosis

When a failed system is being diagnosed, the operator's console is accessed as a line-oriented device, reporting results of diagnostic tests and errors. These results are also logged to a hard copy device (the hard copy operator's console printer or the system line printer). For logging reasons, the operator's console and either a hard copy operator's printer or the system printer must be connected to the master IOP. Diagnosing a system where the IOP connected to these devices has failed requires some recabling.

As noted above, the IOPs may be accessed via the diagnostic bus and their serial interfaces, should the sysbus interfaces be suspect. The UNIBUS and memory must be operational before an IOP can be diagnosed.

6. Error Logging

Errors encountered during normal system operation are logged to an address space on the system disk by the master IOP. This address space is invisible to the R1000 CPU while it is being accessed. A command is provided which closes the current logging address space and opens a new one. The R1000 then can access the data in the error log. The error log is not reflected in the R1000's file system; each disk page of this file contains a pointer to the next disk page. A task must traverse error log pages, copying them to a data stack before the log may be processed by an R1000 software task.

The R1000s send errors to the master IOP via the sysbus. Nonmaster IOPs may send error messages via the sysbus, or via their serial lines and the diagnostic bus (probably hardware error reports during diagnostic execution).

Table of Contents

1. Summary	1
2. System Configurations	1
2.1. Power Distribution	1
2.2. Powerup Scenario	2
2.2.1. Communications between IOPs	2
2.2.2. IPL of the Master IOP	3
2.2.3. Master IOP	3
2.2.4. Nonmaster IOPs	3
2.2.5. IOP Initialization	4
2.2.6. CPU Power and Loading	4
2.2.7. CPU Initialization	4
2.2.8. Assigning Virtual Processors to CPUs	4
2.2.9. Initialization of the Master IOP	5
2.2.10. Recovery of Closed Tasks	5
3. Special Hardware Considerations	5
3.0.1. Communications between IOPs	5
3.0.2. Booting Remote IOPs	6
4. Device Configurations	7
5. Operator Interface	7
5.1. Normal Operation	7
5.2. System Diagnosis	7
6. Error Logging	8