DEC 11/45
RTE

RSX-11D SPEC

TO: RSX-11D Distribution

FROM: H, Krejci

DATE: 19 Jun 72

SUBJ: SYSTEM LISTS

DOC: 130-101-037-00

> The material included in this functional specification, including but not limited to, instruction times and operating speeds is for information purposes only, All such material is subject to change without notice, Concequently DEC makes no claim and shall not be liable for its accuracy,

> Unless specified otherwise, the terms "RSX" and "RSX-11" imply "RSX-11D",

The virtual address space 100000-157777 (ASRs 4,5,6) for the executive (kernel mode), and for Privileged Tasks (user mode), is mapped into an area of real memory containing (1) a System Communications area, called SCOM, (2) the system's tables & lists, and (3) a set of subroutines, called System Subroutines, that manipulate System Lists and provide other system level functions, This Spec describes the system's Tables & Lists,

RSX-11D uses linked lists, and fixed length tables, to maintain system information,

The fixed length tables are lists with elements that reside in consecutive memory locations, This format is used when lists are static, when list scan time is critical, or both,

Most linked lists are linked as Double Ended Queues, and are called DEQUES (pronounced decks), An RSX-11D deque consists of a listhead and list elements, or nodes, circularly linked by both backward and forward pointers, The first word of a node (or listhead) is a forward pointer, i.e., the address of the next node (or listhead) looking forward. The second word of a node (or listhead) is a backward pointer, i.e., the address of the next node (or listhead) looking backward. A listhead is a degenerate node, consisting of only a forward and a backward pointer, and is used as a reference point, i.e., A listhead identifies a deque, and indicates both the beginning and end of the circularly linked list. All nodes consist of two pointers followed by thirteen data words.

The following is a description of the major lists used in RSX-11D

THE TASK PARTITION DIRECTORY

The Task Partition Directory (TPD) is a table of six word entries for each Task Partition [1] defined at System Configuration (SGEN). Each Partition's entry is of the following format:

```
     Wd. ØØ -- Partition name (first half).
     Wd. Ø1 -- Partition name (second half).
     Wd. Ø2 -- 1/64th Base address of partition.
     Wd. Ø3 -- 1/64th Partition size in bytes.
     Wd. Ø4 -- Hole Pointer/Occupied Indicator.
     Wd. Ø5 -- Partition Type indicator.
```

The Partition name consists of six characters in Radix-5Ø representation.

Memory Partition bounds are restricted to lie on 32-word boundaries, and may be located anywhere within 124K of real memory space. Their base & size are recorded in terms of 32-word memory blocks.

A Partition type (wd. 5) zero is a "System Controlled" Partition, i.e., The System will load as many Task as possible in the Partition. For System Controlled Partitions, Wd. 4 is a pointer (1/64th real address) to the first "hole" (if any) within the Partition. Each hole is an unused block of memory, and is a multiple of 32-words in length. The holes (if any) in a System Controlled Partition are circularlly linked from Wd. Ø4. The first two words of

[1] A "Task Partition" is an area of memory dedicated to the execution of Tasks.

each hole contain, (1) a pointer to tne next hole, or to the partition's TPD entry Wd. Ø4, and (2) the hole's size in 32-word blocks. Wd. Ø4 & Wd. Ø5 appear as a hole of zero size, and are used as a listhead for this singlarly linked list.

A Partition type (Wd. 5) one is a "User Controlled" Partition, i.e., The System will only load one Task at a time in the Partition. For User Controlled Partitions, Wd. 4 is used as a "Partition occupied flag" -- zero, available; non-zero, occupied.

THE GLOBAL COMMON DIRECTORY

The Global Common Directory (GCD) is a table of four word entries for each GLOBAL COMMON BLOCK defined at System Configuration. Each entry is of the following format:

Wd. ØØ -- Name of COMMON Block (first half),
Wd. Ø1 -- Name of COMMON Block (second half),
Wd. Ø2 -- 1/64th Base address of COMMON Block,
Wd. Ø3 -- 1/64th size of COMMON Block.

The COMMON Block name consists of six characters in Radix-5Ø representation.

Global Common Block bounds must lie on 32-word boundaries, and may be loacated anywhere within 124K of real memory space. Their base & size are recorded in terms of 32-word memory blocks.

THE PHYSICAL UNIT DIRECTORY

The Physical Unit Directory (PUD) is a table of eight word entries for each physical unit specified at System Configuration. [When a Logical I/O Unit is assigned to a physical unit, the physical unit is represented the address of the corresponding PUD entry.] Each device-unit entry is of the following format:

Wd. ØØ -- Device name (2 ASCII Chars),
Wd. Ø1 -- Unit number & Flags byte,
Wd. Ø2 -- Attached flag,
Wd. Ø3 -- redirected flag,
Wd. Ø4 -- ATL node adr of Handler Task,
Wd. Ø5 -- Requests queued counter,
Wd. Ø6 -- Unit request deque forward linkage,
Wd. Ø7 -- Unit request deque backward linkage,

The bits of the Flags Byte are used as follows:

```
  bit-0 -- set when handler task is resident.
  bit-1 -- set when a file is open.
  bit-5 -- set if device-unit is file-oriented.
  bit-6 -- set if device-unit is readable.
  bit-7 -- set if device-unit is writeable.
```

When the unit is ATTACHed to a task, the Attach Flag is set to the Task's STD entry address. When the unit is not Attached to a task, this flag is set to zero.

When I/O requests have been REDIRECTED to another unit, the Redirect Flag is set to the PUD entry address for that unit. When I/O is not REDIRECTED, the Redirect Flag is set the address of its PUD entry.

THE POOL

Empty nodes for use in any deque are initially provided by the System Configuration Routine in the form of a deque called "The Pool of available list elements", or the POOL. When a node is needed to expand a list, it is taken from the pool. When a node is no longer needed, it is returned [2] to the pool.

While the most common usage of pool nodes is with linked lists (Deques), they are also used by the system to create the data portions of "System Task Directory" entries.

THE SYSTEM TASK DIRECTORY

The System Task Directory (STD) provides information about each Task that has been INSTALLED into a system. The information recorded in a Task's STD entry is either (1) information that is required when the task is not active (viz., RECEIVE Deque listhead), and (2) information that is required to load a Task into memory (viz., Task name, disk address of image).

Under RSX-11, Tasks are referenced by name, and the STD is searched for an indicated Task name at each reference. The STD is structured such that this search can be performed rapidly, without imposing naming conventions, orders of

[2] "Taken" and "returned" imply changing node pointers and not moving data.

installation, or dedication of a large memory area,

The STD consists of a table of entry pointers for t maximum number of INSTALLed tasks, and a fifteen word ent for each Task that is INSTALLED, The table is maintained the programs that INSTALL and REMOVE Tasks such that t number of entries is known, and consecutive table wor point to Task STD entries ordered alphabetically by Ta Name,

The fourteen word block of memory for a STD entry is tak from the POOL when a Task is INSTALLed, and returned wher Task is REMOVED,

Thus, a Task name may be found rapidly (via binary search and memory is not dedicated for STD entries until it needed,

Each STD entry is of the following format:

```
     Wd, ØØ -- Task name (first half).
     Wd, Ø1 -- Task name (second half),
     Wd, Ø2 -- Default partition (SPD entry address),
     Wd, Ø3 -- Default Priority & unused Byte,
     Wd, Ø4 -- Flags Word.
     Wd, Ø5 -- Size of resident code (load image),
     Wd, Ø6 -- Max Task size,
     Wd, Ø7 -- Max Pool usage,
     Wd, 1Ø -- Pool usage,
     Wd, 11 -- Task RECEIVE Deque forward linkage,
     Wd, 12 -- Task RECEIVE Deque backward linkage,
     Wd, 13 -- Disk adr of Task Image (first half),
     Wd, 14 -- Disk adr of Task Image (second half),
     Wd, 15 -- Disk adr of Checkpoint Image (first half),
     Wd, 16 -- Disk adr of Checkpoint Image (second half)
```

The Task name consists of six characters in radix representation,

The default Partition is represented by a pointer (addre of an entry in the "System Partition Directory", wh provides the Partition's name & definition,

The bits of the Flags Word are used as follows:

```
  Bit-Ø -- Set when Task is active,
  Bit-1 -- Set when Task is Fixed-in-memory,
  Bit-2 -- Set when Task is checkpointed,
  Bit-3 -- Set when Task is disabled,
  Bit-4 -- Set when task is being fixed-in-memory,
  Bit-5 -- Set if Task is not-fixable,
  Bit-6 -- Set if Task may be checkpointed,
  Bit-7 -- Set if Task may be disabled,
```

THE ACTIVE TASK LIST

The Active Task List (ATL) is a priority ordered list active Tasks. It consists of a deque with listhead in S and a node of the following format for each active task:

Wd. 00 -- Forward linkage,
Wd. 01 -- Backward linkage,
Wd. 02 -- STD entry pointer (Task ID),
Wd. 03 -- Run Priority, & unused byte,
Wd. 04 -- Run Partition (TPD entry address),
Wd. 05 -- 1/64th real address of load image,
Wd. 06 -- Task Status & I/O In Process Count,
Wd. 07 -- I/O Requests Pending Count,
Wd. 10 -- Task's Event Flags 1-16,
Wd. 11 -- Task's Event Flags 17-32,
Wd. 12 -- Event Flags mask,
Wd. 13 -- Event Flags mask,
Wd. 14 -- Event Flags mask,
Wd. 15 -- Event Flags mask,
Wd. 16 -- Unused.

This list is ordered by the priority of the active Tasks is used to drive the system. The order in which Tasks considered is determined by scanning the list, and action to be taken is determined by examining the T Status word. Following is a list of Task status values their meanings.

00 -- Task needs memory
02 -- Task LOAD request queued
04 -- Task loaded successfully
06 -- Task NOT successfully loaded
10 -- Task is running
12 -- Task RECORD request queued
14 -- Task recorded successfully
16 -- Task NOT successfully recorded
20 -- Task waiting for an Event Flag 1-16 to be set
22 -- Task waiting for an Event Flag 17-32 to be set
24 -- Task waiting for an Event Flag 33-48 to be set
26 -- Task waiting for an Event Flag 49-64 to be set
30 -- Task waiting for an Event Flag 1-64 to be set
32 -- Task SUSPENDED

THE CLOCK QUEUE

The Clock Queue is a deque consisting of one node for e operation to be performed at some time in the future. "ticks till due" count in the first node of the Clock Qu is decremented at each clock tick until "node becomes d

(count is zero), at which time the indicated operation
performed. The nodes are linked in the order in which t
will come due, and are of the following format:

Wd. Ø0 -- Forward linkage.
Wd. Ø1 -- Backward linkage.
Wd. Ø2 -- STD node adr of requestor.
Wd. Ø3 -- ATL node adr of requestor.
Wd. Ø4 -- Ticks behind previous req (high order).
Wd. Ø5 -- Ticks behind previous req (low order).
Wd. Ø6 -- Request type indicator & unused byte.
Wd. Ø7 -- [MT] BIS source operand to set flag.
Wd. 1Ø -- [MT] BIS dest address (kernel virtual) to
set flag.
Wd. 11 -- [TS] STD entry adr of scheduled Task.
Wd. 12 -- [TS] Run Priority & unused byte.
Wd. 13 -- [TS] Run Partition (TPD entry adr).
Wd. 14 -- [TS] Reschedule interval (high order).
Wd. 15 -- [TS] Reschedule interval (low order).
Wd. 16 -- Unused.

[MT] -- Mark Time entry only
[TS] -- Task Scheduling entry only

The Request Type (Wd. 4, Byte-Ø) is used as follows:
Ø -- Mark Time request
1 -- Task schedule request
2 -- Nullified request

The Event Flag Range (wd. 4, Byte-1) is used to indi
which range of 16 Event Flags is to be set (by
instruction) if an Event Flag is to be set. The ranges
defined as follows:
Ø -- Event Flag 1-16
1 -- Event Flag 17-32
2 -- Event Flag 33-48
3 -- Event Flag 49-64

The BIS instruction source operand (wd. 5) indicates whe
an Event Flag is to be set, and if so, which flag within
range indicated in the previous byte.

Words 1Ø-15 are not used if the reqeust is Mark Time.

THE I/O REQUEST QUEUE

The I/O Request Queue is a Deque with listhead in
Physical Unit Directory entry for the unit to which
request has been queued. Each I/O Request node is of
following format:

```
Wd. ØØ -- Forward linkage,
Wd. Ø1 -- Backward linkage,
Wd. Ø2 -- STD entry adr of requestor,
Wd. Ø3 -- Request Priority,
Wd. Ø4 -- Logical Unit Number & Event Flag Number,
Wd. Ø5 -- I/O Function Code,
Wd. Ø6 -- Address (virtual) of I/O Status Word,
Wd. Ø7 -- Parameter #1,
Wd. 10 -- Parameter #2,
Wd. 11 -- Parameter #3,
Wd. 12 -- Parameter #4,
Wd. 13 -- Parameter #5,
Wd. 14 -- Parameter #6,
Wd. 15 -- Parameter #7,
Wd. 16 -- Parameter #8,
```

THE SEND/RECEIVE QUEUES

The Receive Queues are Deques with listheads in the S entries for each Task. Entries are created and queu (FIFO) by SEND or SEND AND REQUEST Directives, and remov by RECEIVE or RECEIVE OR EXIT Directives. Each Receive no is of the following format:

```
Wd. ØØ -- Forward linkage,
Wd. Ø1 -- Backward linkage,
Wd. Ø2 -- STD entry adr of SENDing Task,
Wd. Ø3 -- Data Word #1,
Wd. Ø4 -- Data Word #2,
Wd. Ø5 -- Data Word #3,
Wd. Ø6 -- Data Word #4,
Wd. Ø7 -- Data Word #5,
Wd. 10 -- Data Word #6,
Wd. 11 -- Data Word #7,
Wd. 12 -- Data Word #8,
Wd. 13 -- Data Word #9,
Wd. 14 -- Data Word #10,
Wd. 15 -- Data Word #11,
Wd. 16 -- Data Word #12,
```

THE INTERRUPT CONNECT NODE

Interrupt service routines are a part of Privileged Tas (normally I/O Handler Tasks), and therefore, are n necessiarly always memory resident. These service routin may reside anywhere in 124K of real address space, and a executed in kernel mode under kernel ASR3.

Whenever an interrupt service routine is "connected" to
hardware interrupt, a node is taken from the pool and se
as follows, and the interrupt's trap PC is set to
address of the fourth word (Wd. Ø3) of the node.

Wd. ØØ -- Unused
Wd. Ø1 -- Unused
Wd. Ø2 -- STD entry adr of connecting task
Wd. Ø3 -- "MOV ASR3,-(SP)" instruction
Wd. Ø4 -- (second word of MOV instruction)
Wd. Ø5 -- "MOV (PC)+,ASR3" instruction
Wd. Ø6 -- 1/64th base of service routine's adr space
Wd. Ø7 -- (third word of MOV instruction)
Wd. 1Ø -- "JMP ENTRY" instruction
Wd. 11 -- (second word of JMP instruction)
Wd. 12 -- Unused
Wd. 13 -- Unused
Wd. 14 -- Unused
Wd. 15 -- Unused
Wd. 16 -- Unused

The STD entry address is used to identify the connecti
Task, so that when the interrupt is "disconnected" and
connect node is returned to the pool, the Task's pool us
count can be decremented.