QBUS/Unibus Storage and Input/Output Card (QSIC/USIC) Programming Manual

David Bridgham Noel Chiappa

August 27, 2020

Contents

Chapter 1

QBUS/UNIBUS Storage and Input/Output Card

1.1 Introduction

The QSIC and USIC are cards which provide emulation of various QBUS and UNIBUS controllers and disks, using MicroSD cards or (eventually) USB disks as the actual storage media. They use NOS original bus drivers (DS8641s), and then level converters to interface to a modern FPGA.

Eventually, they are likely be extended to allow emulation of other controllers, e.g. the Interlan NI1010 and NI2010 Ethernet, via devices plugged into the USB port.

They currently implement upwardly compatible, but extended, versions of the old DEC RK11 and RP11 disk controllers. Although the emulation is not exact, it is good enough that un-modified operating system images (UNIX V6) are able to boot and run.

The emulation limitations are in part because many of the control register bits only make sense with an actual physical drive; also, exact emulation, including delays (e.g. for now non-existent seeks) would limit the performance obtainable. (It is possible some systems that will not work without better emulation of such delays; if so, an option could be added to better emulate them.)

The QSIC is a dual-height QBUS card; it can hold two microSD cards, allowing direct card->card backup. The controllers are denominated as the RKV11-F and RPV11-D; the extension is allowing DMA to the entire 2^{22}

byte QBUS address space.

The USIC is the same functionality in a quad-height SPC card for the UNIBUS. The USIC will optionally adds the Able ENABLE functionality, which allows processors with only 18-bit addressing to have access to 2^{22} bytes of memory.

When this is enabled, the RK11-F and RP11-D (as they are denominated here) can be set to be 'MASSBUS' controllers (notionally), with full direct access to the entire memory (which is on the USIC), without going through a UNIBUS Map. With that turned off, they emulate the originals; i.e. they do DMA cycles on the UNIBUS.

Both cards have provision for adding indicator panels, as close as possible to the DEC originals, to display internal state and datal; this will help invoke the feel of the older machines. They might even be useful for debugging from time to time!

1.2 Basic operation

The space on the microSD cards ('storage devices') is divided into 'packs', described by a 'pack table' on the card (because it applies only to that card), which gives their location and size. Packs can be 'loaded' on 'drives'; in other words, everything works much like the original hardware.

(The term 'mount' is reserved for the operation of letting the operating system add a pack to the visible file system – again, like the existing UNIX, etc, terminology.)

There are also 'load tables', which record which packs are loaded on which drives; a non-volatile instance of these allows a system to be cold-booted without going through a pack loading phase.

A storage device is 'inserted' into a microSD slot; removing one without previously un-mounting (and un-loading) the packs is an error which can damage storage contents, just as switching a physical pack without unmounting it would have on the original hardware.

Removing a storage device will auto-unload any packs still loaded. Before any further disk operations can happen, any packs on a new storage device which are to be used have to be loaded; attempting to use them without that step will produce 'disk not loaded' faults (e.g. clears 'Drive Ready' on the RK11).

Each controller supports the maximum 8 drives of the original. It will

be possible to configure more than one instance of each controller, should simultaneous access to more packs/drives be needed.

1.3 Configuration

The QSIC and USIC contain, potentially, a multitude of devices all of which need configuration. If we did it in the traditional manner with jumpers and DIP switches, the boards would be a mess of DIP switches and difficult to change as we update the FPGA load. Therefore, configuration of these devices is handled through two I/O registers which give access to a series of internal configuration registers inside each of the emulated devices.

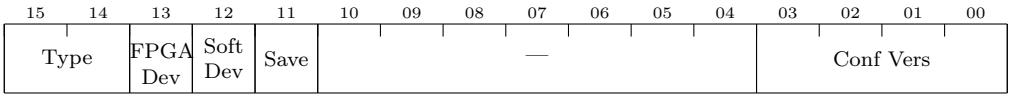
This configuration may be saved to internal flash memory where it will be restored at startup. Some configuration information is more dynamic, such as the disk pack load tables, and needs to be re-computed at each boot.[1]

1.3.1 Bus Registers

Access to the internal configuration registers is through two I/O registers on the UNIBUS or QBUS located at 777720_8 and 777722_8. The first register is the address register. Setting this selects which internal configuration register the second register accesses. Reading or writing the second bus register then accesses the specified configuration register.

1.3.2 Top-Level Configuration Table

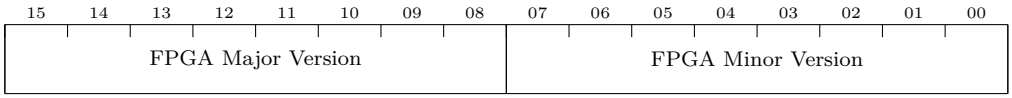
The internal configuration begins at fixed location 0. It gives some information about the USIC or QSIC and then indexes all of the rest of the configuration for the rest of the devices.



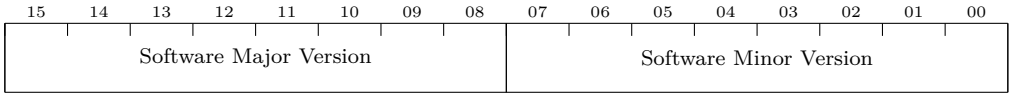

```
Address = 0
```

[1] Still to be designed.

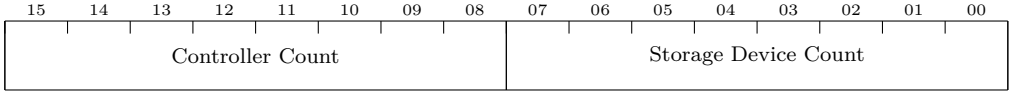
Bit	Designation	Description and Operation
0-3	Conf Vers	Version of the configuration format. This document describes version 0.
11	Save	Set this bit to cause the current configuration to be saved to flash memory. While saving, this bit will read as 1. Do not modify the configuration while saving is in progress.
12	Soft Dev	The Software version shown is under development. This bit will be cleared when it is a released version.
13	FPGA Dev	The FPGA version shown is under development. This bit will be cleared when it is a released version.
14-15	Type	Type of board. USIC 00 QSIC 01



Address = 1



Address = 2

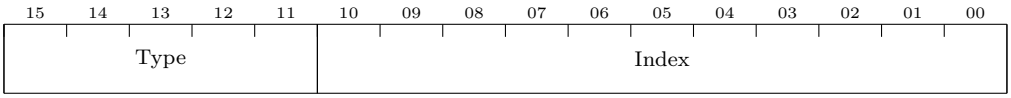


Address = 3

Bit	Designation	Description and Operation
0-7	Storage Device Count	A count of how many entries are in the storage devices table that follows the controller table.
8-15	Controller Count	A count of how many entries are in the controller table that follows.

Beginning at word 4, there is a table of Controllers followed by a table of

Storage Devices. Each entry in the tables is one word long.



Bit	Designation	Description and Operation
0-10	Index	The start address of the configuration block for the specified device.
11-15	Type	The type of device referenced. For controllers, types are:

```
Indicator Panels  0
RK11-F            1
RP11-D            2
Enable+           3
Interlan 1010     4
```

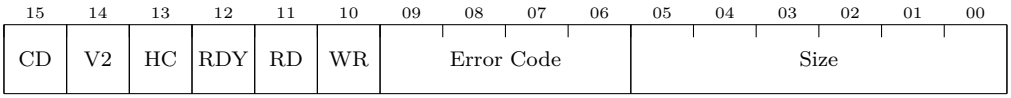
For Storage Devices:

```
SD Card   0
RAM Disk  1
USB       2
```

1.3.3 Storage Devices

The storage devices have no configuration but they show up in the configuration system as a way of reporting status and diagnostics.

SD Card

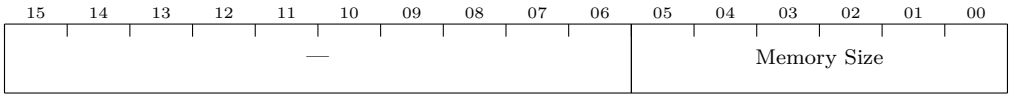


Bit	Designation	Description and Operation
0-5	Size	The $\log_2$ of the size of the SD Card in 512 byte blocks.
6-9	Error Code	If non-zero, the SD card controller is indicating an error.[2]

[2]Will fill out a table of error codes once they're set in the code.

Bit	Designation	Description and Operation
10	WR	The card is in the middle of a write operation.
11	RD	The card is in the middle of a read operation.
12	RDY	The device is ready.
13	HC	High Capacity
14	V2	SD Version 2
15	CD	Card Detect

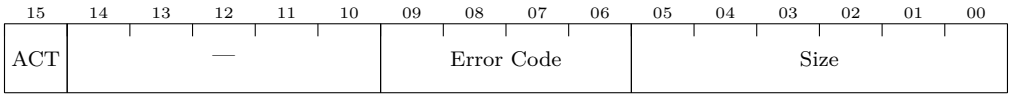
RAM Disk



Bit	Designation	Description and Operation
0-5	Size	The $\log_2$ of the size of memory in 512 byte blocks.[3] Zero (0) indicates there is no RAM available.[4]

USB Device

This is mostly a placeholder for now, since we have not even begun to implement USB devices. I'm expecting to have some small, fixed number of USB devices in the FPGA that get mapped to actual USB storage devices as they're plugged in.



Bit	Designation	Description and Operation
0-5	Size	The $\log_2$ of the size of the USB device in 512 byte blocks.
6-9	Error Code	If non-zero, the USB Controller is indicating an error.
15	ACT	This USB device is active.

[3] Current value is 19. 256 MiBytes = 2^{28} Bytes = 2^{19} blocks.

[4] Wait, if there's no RAM then would we even have an entry for the memory?

Load Tables

The load tables in each disk controller need to specify on which storage device a disk pack resides. These are the values that are used. Note that the SD Cards are physical locations while the different USB devices are logical.

```
SD Card 0  0
SD Card 1  1
 RAM Disk  2
    USB 0  3
    USB 1  4
    USB 2  5
    USB 3  6
    USB 4  7
```

1.4 RAM disks

In addition to storing pack images on removable media, the QSIC/USIC also supports 'RAM drives' – packs which are kept in on-board RAM on the card. The contents do not survive power cycles, but RAM drives are desirable for uses which do a lot of writing, none of which needs to survive a system re-boot, e.g. swapping and paging.

The reason is that microSD cards (unlike the original magnetic media) cannot perform un-limited write cycles; they wear out after a finite number of writes. So, if there is no need to keep writes over re-boots (e.g. swapping or paging data), those operations should be assigned to a RAM disk for long-term use. On most operating systems, it is fairly easy to configure them to do swapping/paging to particular devices.

1.4.1 microSD card selection

There have been reports of cheap commodity microSD cards failing after relatively small numbers of writes. So, these should be avoided; also, backups of the data on microSD cards should be kept fairly meticulously. There are "industrial grade" microSD cards available, which are more robust than good consumer-grade cards, so those are perhaps worth using.

1.4.2 UNIX pipes

The UNIX pipe device can also generate lots of "temporary" writes. Unfortunately, on the early versions of UNIX, pipes are created on the root device – probably the last place you want them, if there are microSD card issues!

This is simple to fix, though; in pipe(), in pipe.c, change the line:

ip = ialloc(rootdev);

to

ip = ialloc(pipedev);

and then go into c.c and add a line underneath the declaration of "rootdev" to add a "pipedev".

Don't forget that you will need to create a file system on any RAM pack you are using to hold pipes, before you can use a pipe, though! Probably the safest thing is to start the system with pipedev set to the root device, and then reset pipedev once the pipe pack has a filesystem on it. (This is quite OK, because once a pipe is created, it stays on the device it was created on, and it's possible to have open pipes on more than one device.) A program to set 'pipedev' is available for V6.

1.5 Upgrading drivers

The devices on QSIC, and USIC with Enable+ on, can do DMA to all 22 bits of address space, but are program compatible for 18 bits (i.e. existing software will run, but can only use the low 2^{18} bytes of memory). To use more than 2^{18} bytes – generally only needed to use the QSIC/USIC devices for swapping/paging – the device driver will have to be fixed – not too complicated, but only if one has the capability!

The 18-bit program compatability is, however, useful for extending operating systems to use the 22-bit capability. (For systems like UNIX, which restrict block device I/O to buffers in the low 2^{16} bytes of memory, the 22-bit capability is not needed if they are not being used for swapping/paging.) Just simply boot the OS with only 2^{18} bytes of memory; modify the device driver to be able to use the 22-bit addressing capability; and restart. (This technique was used to give Unix V6 access to the 22-bit capability.)

NOTE: On OSs that auto-size memory, if you boot the system with more than 256KB of memory, if it tries to swap to high memory with the existing 18-bit driver, either i) the system will crash if it knows that RK11's and/or

RP11's are 18-bit only (well, technically, RKV11-D's are actually only 16 bits), or ii) the transfer will go someplace else in memory from where the OS thought it was going to go. (E.g. UNIX V6 would do this – although the address in the I/O request is 22 bits long, the existing RK11/RP11 drivers only look at the low 18.) The solution is to boot the system with only 256KB of memory installed; then update the driver to be able to do 22 bit DMA; then the rest of the memory can be added back.

Chapter 2

Indicator Panels

The QSIC and USIC support indicator panels which look exactly like the old DEC ones. These comprise (like the originals) a bezel, a captioned inlay, a light shield, and a PCB holding the lamps ("warm white" LEDs, which look identical to the original units, when seen through the inlay); the whole mounts to a 19″ rack.

As on the originals, the bezel and inlay go on the standard "latch mouldings" (the flat plastic units with the two posts with a spherical ball on the top) mounted to the rack; the light shield ("Benelex" in DEC parts jargon) mounts to the rack with a pair of brackets, and the lamp PCB mounts to that.

Although the new panels are mechanically compatible with the originals (so that original bezels, inlays, etc can be used, and vice versa), the hardware interface to the lamp PCB is totally different (bit-serial via a 4-wire interface – data, clock, latch and ground), rather than 'wire per bulb', as in the originals. As a result, the PCBs are also completely different. The new PCBs are limited in size (for ease of fabrication); each only holds 12 columns of lamps, and a set of 3 plug together to make the full 36-wide array of the originals.

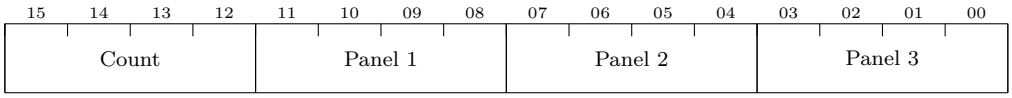
The maximum number of panels a single QSIC/USIC can support is not yet determined, but should be at least 4.

We can supply complete units (mostly newly-fabricated), but we have only a limited supply of original bezels.

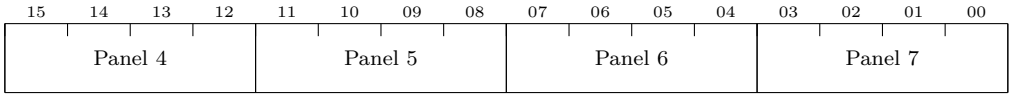
2.1 Configuration

The configuration registers allow the user to set how many indicator panels are attached and what should be displayed on them.

All the indicator panels are driven by a single, differential serial line running at approximately 100 kHz. This serial line is daisy-chained from one panel to the next. The more indicator panels you configure, the slower their update. The current configuration allows for up to seven but that maximum has not yet been tested.



`Address = Start+0`



`Address = Start+1`

Bit	Designation	Description and Operation
	Count	How many panels are active.
	Panel N	The type of each panel. That is, what to display. Panel 1 is first in the chain (that is, closest to the QSIC or USIC).

```
    lamptest  0   All lights on
 Bus Monitor  1   Unibus or QBUS[1]
     RK11 #0  2
     RK11 #1  3
     RP11 #0  4
     RP11 #1  5
     Enable+  6
    Interlan  7[2]
   debugging  15
```

[1] Also includes some SD Card and USB status lights.

[2] Do we even want a full indicator panel for the Interlan Ethernet board? Maybe just grab two or three lights off the bus monitor display.

2.2 Inlays

Although we could do exact duplicates of original panels, the old DEC panels have lots of lights that don't make sense without an actual physical disk (e.g. 'write current on'), and also leave off other ones that would be useful (e.g. memory address). So, we default to a new layout we have designed, which makes the best use of the available set of lamps.

The new layout works with both the RK11 and RP11; the RK11 does not drive all the disk address lights, but is otherwise identical.

We might be able to provide exact copies of the old ones for people who are crazy for authenticity; it will require custom FPGA loads to drive their lamps correctly, though.

In addition to the RK/RP inlays, there is also an inlay to monitor QBUS activity.

Chapter 3

RK11-F

The RK11-F is the implementation of the RK11 disk controller inside the QSIC and USIC. It is substantially compatible with the RK11-D with extensions for extended addressing on the QBUS and some reinterpretations of some of the error bits to better match the flash media the QSIC/USIC uses for storage devices.

To provide 22-bit addressing, the RK11-F has recycled the maintenance register from the RK11-C (which was unused in the RK11-D) to hold the extra address bits. On the Unibus, the RK11-F with extended addressing is notionally a MASSBUS device, giving direct access to the full 2^{22} byte physical address space supported by the ENABLE+ functionality.

On the QBUS, the RK11-F only has 22-bit mode though if you ignore the extended address register it will act mostly like an 18-bit device.[1]

DEC did build an RKV11-D that was an RK11-D modified to work on the QBUS. For some reason, it only support 16-bit addresses even though the RK11-D supported 18-bit addresses on the Unibus. We think the RK11-F, even though in 22-bit mode, ought to work with legacy software but if this turns out to not be the case we'll return to this issue.

[1]The exception has to do with the address register wrapping around at the end of the address space. This shouldn't be an issue in practice as it would involve DMA operations into the I/O page which would almost certainly generate NXM errors.

3.1 Configuration

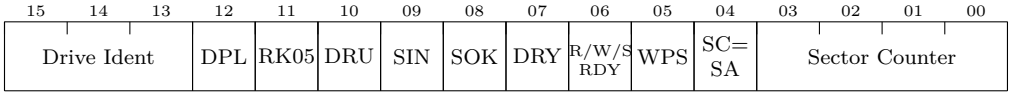
By default, eight I/O registers begin at 777400_8[2]. The default interrupt vector is 220_8 and the interrupt priority is 4. The USIC can be configured for 18-bit or 22-bit addresses. The RK11-F as a whole may be disabled if it's not wanted.

3.2 Programming

The address shown for each register is the default address for the first RK11 controller.

3.2.1 Drive Status Register (RKDS)

```
Address = 777400
```



Bit	Designation	Description and Operation
00-03	Sector Counter (SC)	The current sector address of the selected drive. On the QSIC/USIC, this is just a free-running counter clocked at 312.5kHz ($3.2\mu s$).[3] All disks share a single Sector Counter.
04	Sector Counter Equals Sector Address (SC=SA)	Set when the Sector Counter is equal to the Sector Address (RKDA 03-00).
05	Write Protect Status (WPS)	Set when the selected disk is in the write-protected mode.

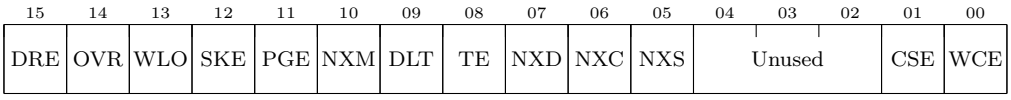
[2]Assuming we include two RK11s by default, I should add the default values for the second RK11 too.

[3]Check me that this is about the right speed.

Bit	Designation	Description and Operation
06	Read/Write/Seek Ready (R/W/S RDY)	Indicates a storage device is loaded and ready to accept commands. In the QSIC/USIC, a storage device may be serving multiple disks at once and so unable to accept commands right now because it's otherwise busy. However, the disk controller can still accept a command and it will wait until the storage device is ready.[4]
07	Drive Ready (DRY)	A storage device is loaded for the selected disk.
08	Sector Counter OK (SOK)	Unused, set to 1.
09	Seek Incomplete (SIN)	Set to 0. Seeks always complete.
10	Drive Unsafe (DRU)	Unused, set to 0.[5]
11	RK05	Set to 1 to indicate this is an RK05.
12	Drive Power Low (DPL)	Unused, set to 0.
13-15	Identification of Drive (ID)	Set to the drive number that caused an interrupt.[6]

3.2.2 Error Register (RKER)

Address = 777402



[4]This is a description of what it does right now. As I wrote this, I realized that another possible implementation, perhaps better, would for it to be the AND of DRY (Drive Ready) and RDY (Control Ready).

[5]Could be used to indicate storage device initialization failure, perhaps.

[6]Currently I set this from state CMD_DONE regardless of whether I generate an interrupt or not. Also, I should check that it sets ID for all commands that could generate an interrupt.

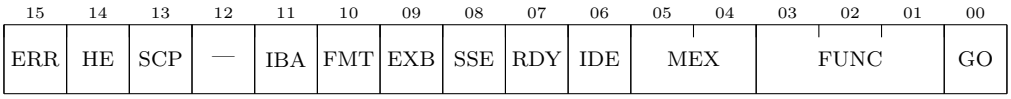
Bit	Designation	Description and Operation
00	Write Check Error (WCE)	Indicates that the data comparison didn't match during a Write Check function.[7]
01	Checksum Error (CSE)	Indicates a checksum error while reading data during a Read Check or Read function. The RK11-F does not do its own checksums on the data and this bit reflects the checksum from the SD Card or USB checksum.[8]
02-04	Unused	
05	Nonexistent Sector (NXS)	Indicates that an attempt was made to initiate a transfer to a sector larger than 13_8.
06	Nonexistent Cylinder (NXC)	Indicates that an attempt was made to initiate a transfer to a cylinder larger than 312_8.
07	Nonexistent Disk (NXD)	Indicates that an attempt was made to initiate a function on a nonexistent drive.[9]
08	Timing Error (TE)	Unused, set to 0.
09	Data Late (DLT)	Unused, set to 0.
10	Nonexistent Memory (NXM)	Set if memory does not respond within the but timeout on the memory cycle.
11	Programming Error (PGE)	Unused, set to 0.
12	Seek Error (SKE)	Unused, set to 0.
13	Write Lockout Violation (WLO)	Set if an attempt is made to write to a disk that is currently write-protected.[10]
14	Overrun (OVR)	Indicates that, during a Read, Write, Read Check, or Write Check function, operations on sector 13_8, surface 1, and cylinder address 312_8 were finished, and the RKWC has not yet overflowed. This is essentially an attempt to overflow out of a disk drive.

[7]Not yet implemented.
[8]Not yet implemented.
[9]Not yet implemented.
[10]Not yet implemented.

Bit	Designation	Description and Operation
15	Drive Error (DRE)	Unused, set to 0.

3.2.3 Control Status Register (RKCS)

Address = 777404



Bit	Designation	Description and Operation
00	GO	When set, causes the RK11-F to act on the function contained in bits 01 through 03 of the RKCS.[11]
01-03	Function	The function to be executed when GO is set. Control Reset 000 Write 001 Read 010 Write Check 011 Seek 100 Read Check 101 Drive Reset 110 Write Lock 111
04-05	Memory Extension (MEX)	A 2-bit extension to RKBA giving an 18-bit bus address. If 22-bit addresses are enabled (QSIC only), these two bits are replicated as bits 00 and 01 of RKXA.
06	Interrupt on Done Enable (IDE)	When set, causes an interrupt to be issued on various condition.[12] The interrupt priority and vector are configurable.
07	Control Ready (RDY)	Control is ready to perform a function.
08	Stop on Soft Error	Currently not implemented.

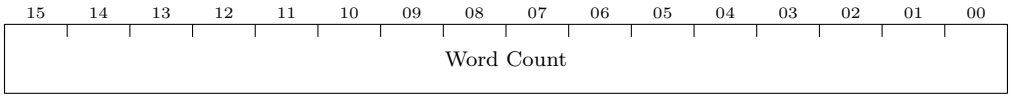
[11]The RK11-D and RK11-E manual lists GO as write-only. The RK11-F currently allows it to be read. Oh wait, I know why they did that. I need to fix the code.

[12]Should audit the code and list all the conditions that can generate an interrupt.

Bit	Designation	Description and Operation
09	Extra bit (EXB)	Unused.
10	Format (FMT)	Not applicable to the QSIC/USIC.[13]
11	Inhibit Incrementing the RKBA (IBA)	Inhibits the RKBA from incrementing during a normal transfer. This allows data transfers to occur to or from the same memory location throughout the entire transfer operation.
12	Unused	
13	Search Complete (SCP)	Indicates that the previous interrupt was the result of some previous Seek or Drive Reset function. Cleared at the initiation of any new function.[14]
14	Hard Error	Set when any of RKER 05-15 are set.
15	Error (ERR)	Set when any of RKER is set.

3.2.4 Word Count Register (RKWC)

Address = 777406

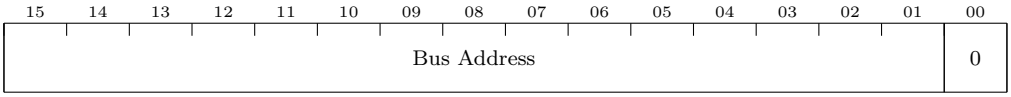


Bit	Designation	Description and Operation
00-15	Word Count	The 2's complement of the number of words to be transferred by a function. The register increments by one after each word transfer. When the register overflows to 0, the transfer is completed and the RK11 function is terminated.

3.2.5 Current Bus Address Register (RKBA)

Address = 777410

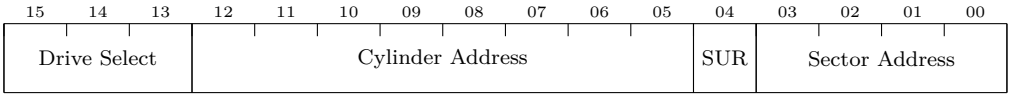
[13]Currently the FMT bit is ignored but it probably should generate some sort of error.
[14]Not yet implemented.



Bit	Designation	Description and Operation
00-15	BA00-BA15	The low 16-bits of the bus address to be used for data transfers. The MEX bits (bits 04 and 05 of RKCS) extend the address to 18-bits and, if enabled, the BAE bits (bits 00-05 of RKXA) extend the address to 22-bits (QSIC only). Bit 00 is always 0 as all transfers are a full word.

3.2.6 Disk Address Register (RKDA)

```
Address = 777412
```



Bit	Designation	Description and Operation
00-03	Sector Address (SA)	The disk sector to be addressed for the next function.
04	Surface (SUR)	Upper or lower surface has no meaning for SD cards or USB flash drives so this maps to just another bit of cylinder addressing.
05-12	Cylinder Address (CYL ADDR)	The cylinder address currently being selection. The largest valid cylinder is 312_8.
13-15	Drive Select (DR SEL)	The logical drive number currently being selected.

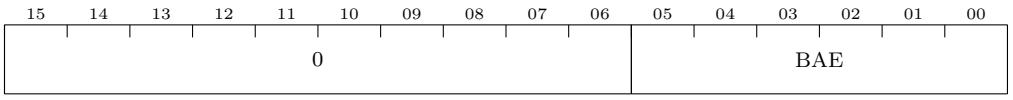
3.2.7 Extended Address Register (RKXA)

```
Address = 777414
```

On the RK11-C this register was a maintenance register and on the RK11-D it was unused. If addressing is set to 18-bits, this register reads as 0 and

writing has no effect, like on the RK11-D.

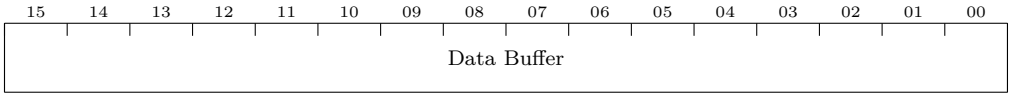
If addressing is set to 22-bits, this register extends the Bus Address register to a full 22-bits. On the Unibus, this only makes sense in the presence of the ENABLE+ and the address is then a physical address rather than being mapped by the ENABLE+. On the QBUS it's always a physical address anyway.



Bit	Designation	Description and Operation
00-05	Bus Address Extension (BAE)	If 22-bit addressing is enabled, these bits extend the Bus Address Register to 22-bits. Bits 00 and 01 are duplicates of MEX (bits 04 and 05 of RKCS) and may be read or written through either register.

3.2.8 Data Buffer Register (RKDB)

`Address = 777416`



Bit	Designation	Description and Operation
00-15	Data Buffer (DB00-DB15)	This register reads from the read end of the FIFO connecting the RK11-F to its storage device. Writing to the Data Buffer has no effect.

Chapter 4

RP11-D

The RP11-D is the implementation of the RP11 disk controller inside the QSIC and USIC. The RP11 never existed on the QBUS at all, but it is being supported there because it's a nice simple controller, and can be very simply extended to provide very large disks.

The RP11-D has been extended to support 22-bit addressing, both on the QSIC (for the 22-bit QBUS), and on the USIC (where it's notionally a MASSBUS device), to be able to have access to the 2^{22} bytes of main memory supported by the ENABLE+ functionality available on the USIC.

The extended address needs more bits alongside the existing Bus Address Register; these are stored in a new register, the 'Extended Address Register' (RPXA). To find a place for this, the RP11-D takes advantage of the fact that the original RP11-C hardware actually responds from 776700_8 up to 776736_8, but there are no actual registers from 776700_8 to 776706_8, so we can put any additional registers needed in that range.

We've also extended it further to allow for much larger disks. This is done by the simple expedient of extending the track and cylinder fields into all the adjacent unused bits, and allowing the sector, head, and cylinder fields of the disk address to take on any value that fits. (These bits are not enabled unless the RP11-D is configured for extended packs.) The result is 28 bits of linear block address, or a maximum disk size of 2^{37} bytes or 128 GiBytes. Obviously you'll need the ability to modify your disk driver to take advantage of this. Another added register allows the device driver to see the configured pack size of each disk pack.

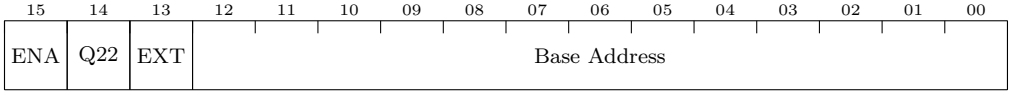
4.1 Configuration

The RP11-D has a set of configuration registers that take the place of the jumpers and DIP switches of earlier disk controllers. Additionally, there are configuration registers that make up a "load table" that maps logical disk drives to disk packs located on the various storage media.

This block of registers is referenced from the top-level configuration block as described in §1.3, page 3. In this way, multiple RP11-Ds can each have its own set of configuration registers and they can move around in the configuration address space without needing to redefine anything here. Addresses shown here are relative to the beginning of this block of RP11-D configuration.

4.1.1 Device

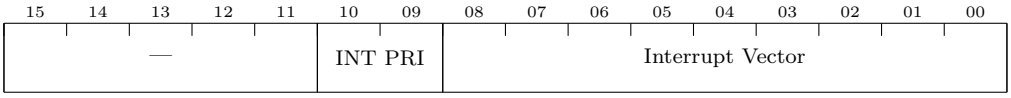
`Address = Start+0`



Bit	Designation	Description and Operation
00-12	Base Address	The base address of the RP11-D's registers in the I/O page on the QBUS or Unibus. The default for the first RP11 is 776700_8.
13	Extended	Enables extended disk packs. This breaks compatibility with previous RP11s but allows for much larger disks if you're able to modify your disk driver.

Bit	Designation	Description and Operation
14	Q22	Enables 22-bit operation, otherwise the RP11-D acts as an 18-bit device. On the QBUS, Q22 would be normal while on the Unibus, 18-bits would be normal. Selecting Q22 on the Unibus while the Enable+ is also enabled, directs the RP11-D to DMA directly to 22-bit memory with physical addresses, not going through the Enable+ Unibus mapping registers. An ersatz MASSBUS disk, if you will. Enabling 22-bit addressing on the Unibus without the Enable+ is currently undefined.
15	Enable	Enables this RP11-D.

```
Address = Start+1
```



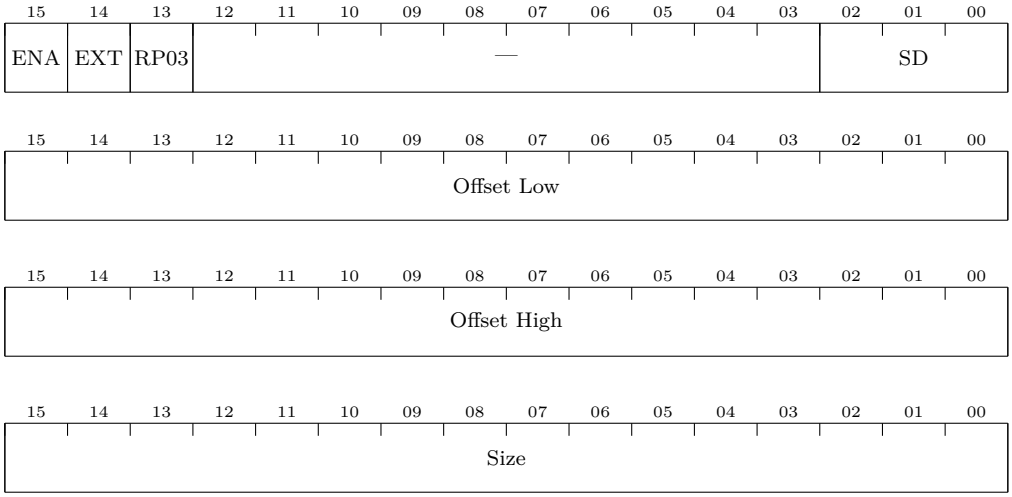
Bit	Designation	Description and Operation
00-08	Interrupt Vector	The interrupt vector to use. The default for the first RP11 is 254_8.

09-10	INT PRI	The interrupt priority. The default is interrupt priority 5.
		`Priority 4 00`
		`Priority 5 01`
		`Priority 6 10`
		`Priority 7 11`

4.1.2 Load Table

The Load Table immediately follows the device configuration and tells the RP11-D about simulated "disk packs" that are loaded into its "drives". It has eight entries, corresponding to the eight disk drives the disk controller supports. Each entry is four consecutive configuration words.

```
Address = Start+2,6,10,14,18,22,26,30
```



Bit	Designation	Description and Operation
00-02	SD	The Storage Device the pack lives on. See the table at §1.3.3, page 7.
13	RP03	If this is not an extended disk pack, this specifies an RP03 disk (406 cylinders, ~40MB) otherwise an RP02 disk (203 cylinders, ~20MB).
14	EXT	This is an extended disk pack. Only allowed if the RP11-D is configured to allow extended packs.[1]
15	ENA	This pack is enabled. Whether the pack shows as loaded depends on both this bit and whether the associated storage device is active.
	Offset	The offset, in blocks, of the pack from the beginning of its Storage Device.

[1] Considering using the File Unsafe error (RPDS bit 09) to indicate that an extended pack was configured on a non-extended RP11-D.

Bit	Designation	Description and Operation
	Size	For extended disk packs, this is the largest cylinder number (one less than the number of cylinders). It is set to 0 for legacy RP02 or RP03 disk packs, allowing for a mix of legacy and extended disk packs on the same RP11-D controller.

On extended disk packs, the RP11-D has 4 bits to specify the sector address and 8 bits for the track. All values are used (16 sectors per track and 256 tracks per cylinder) so there are 2^{12} or 4,096 sectors per cylinder.

The smallest disk supported is 4 MiB with 2 cylinders (since a size of 0, 1 cylinder, specifies a legacy disk pack) while the largest is 65,536 cylinders or 128 GiB.

4.2 Programming

The RP11-D is substantially compatible with the RP11-C, with extensions for extended addressing, the optional extended disk addresses, and again, some different meanings to error bits to better match the flash media the QSIC/USIC uses for storage devices.

Initiate functions (Idle, Seek and Home Seek) only require short periods; execute functions (the rest) tie up the controller until it is finished with the operation. No other operation (initiate or execute) can be started until the execute function has completed. Plain Read and Write include an implicit seek, if needed; the Seek command allows specifying the desired head (track) as well.

After reading or writing the last sector in a track, the RP11-D automatically advances to the next track; if the track that overflowed was the last track in the cylinder, the cylinder automatically advances to the next cylinder. If the cylinder that overflowed was the last cylinder, End of Pack in the RPER is set.

Neither the 36-bit mode, nor header commands (i.e. Header bit in the RPCS set), nor either parity, is supported.

The default interrupt vector is 254_8 and the interrupt priority is 5. Both the interrupt priority and vector are configurable.

4.3 Registers

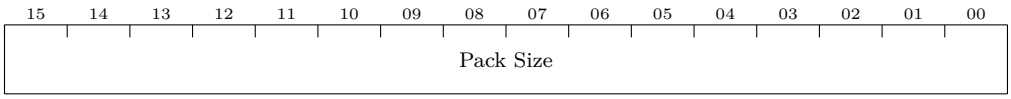
The address shown for each register is the default address for the first RP11 controller.

4.3.1 Pack Size Register (RPPS)

```
Address = 776704
```

If the larger pack size is not enabled, this register reads as 0, and writing has no effect, like on the RP11-C.

If it is enabled, this register give the size of the pack currently mounted on the drive selected by the 'Drive Select' field of the RPCS. If a regular RP02 or RP03 pack is loaded on a drive, the pack size will show as 0 and the disk addressing for that disk will be the regular cylinder/track/sector rather than a linear block address.



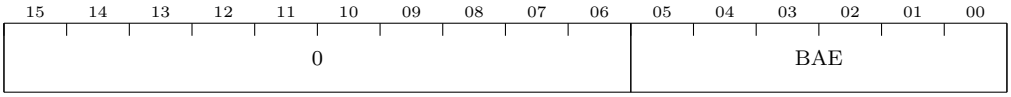
Bit	Designation	Description and Operation
00-15	Pack Size (PS)	Contains the largest valid cylinder number of the drive selected by the 'Drive Select' field of the RPCS; read-only.

4.3.2 Extended Address Register (RPXA)

```
Address = 776706
```

If addressing is set to 18-bits, this register reads as 0 and writing has no effect, like on the RP11-C.

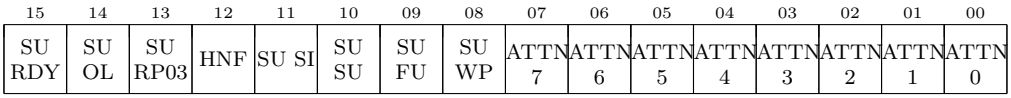
If addressing is set to 22-bits, this register extends the Bus Address register to a full 22-bits. On the UNIBUS, this only makes sense in the presence of the ENABLE+ and the address is then a physical address rather than being mapped by the ENABLE+. On the QBUS it's always a physical address anyway.



Bit	Designation	Description and Operation
00-05	Bus Address Extension (BAE)	If 22-bit addressing is enabled, these bits extend the Bus Address Register to 22-bits. Bits 00 and 01 are duplicates of MEX (bits 04 and 05 of RPCS) and may be read or written through either register.

4.3.3 Drive Status Register (RPDS)

Address = 776710



The Attention bits are read-write (and may be written by a 'write byte' bus cycle), the rest are read-only.

Bit	Designation	Description and Operation
00-07	Drive Attention	Set when the drive completes a seek.
08	Selected Unit Write Protected	Set when the selected drive is in write-protected mode.
09	Selected Unit File Unsafe	Unused, set to 0.[2]
10	Selected Unit Seek Underway	Unused, set to 0.[3]
11	Selected Unit Seek Incomplete	Set to 0; seeks always complete.
12	Header Not Found	Unused, set to 0.
13	Selected Unit RP03	Set to 1 to indicate this is an RP03.

[2]Could be used to indicate storage device initialization failure, perhaps.
[3]May depend on what we do with seeks.

Bit	Designation	Description and Operation
14	Selected Unit Online	Set to 1 when i) an SD card has been installed; ii) it has successfully completed initialization; iii) a pack partition on that card has been assigned to this drive.
15	Selected Unit Ready	Set to 1 as with SU OL, except during a read or write operation to this disk, when is it 0.

4.3.4 Error Register (RPER)

Address = 776712

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
WPV	FUV	NXC	NXT	NXS	PROG	FMTE	MODE	LPE	WPE	CSME	TIMEE	WCE	NXME	EOP	DSK ERR

The RPER is a read-only register (except in maintenance mode, which is not currently supported).

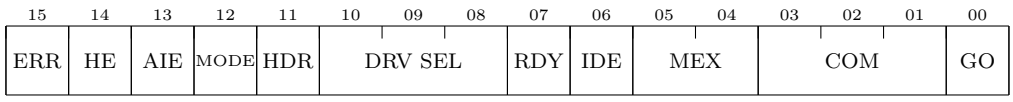
Bit	Designation	Description and Operation
00	Disk Error	OR of HNF and SU SI, so always 0.
01	End of Pack	Indicates that, during a Read, Write, Read Check, or Write Check function, operations on sector 11_8, track 23_8, and cylinder 625_8 were finished, and the RPWC has not yet overflowed. This is essentially an attempt to overflow out of a drive.
02	Nonexistent Memory	Set if memory does not respond within the bus timeout on a memory cycle.
03	Write Check Error	Indicates that the data comparison didn't match during a Write Check function.[4]
04	Timing Error	Unused, set to 0.
05	Checksum Error	Indicates a checksum error while reading data during a Read Check or Read function. The RP11-D does not do its own checksums on the data and this bit reflects the checksum from the SD Card or USB checksum.[5]

[4]Not yet implemented.
[5]Not yet implemented.

Bit	Designation	Description and Operation
06	Word Parity Error	Unused, set to 0.
07	Longitudinal Parity Error	Unused, set to 0.
08	Mode Error	Unused, set to 0.
09	Format Error	Unused, set to 0.
10	Programming Error	OR of transfer attempted with the RPWC set to 0; an operation was attempted on a drive which was not online; an operation was attempted while another was still in progress.
11	Non-existent Sector	Indicates that an attempt was made to initiate an operation to a sector larger than 11_8.
12	Non-existent Track	Indicates that an attempt was made to initiate an operation to a track larger than 23_8.
13	Non-existent Cylinder	Indicates that an attempt was made to initiate a transfer to a cylinder larger than 625_8.
14	File Unsafe Violation	Unused, set to 0.
15	Write Protect Violation	Set if an attempt is made to write to a disk that is currently write-protected.[6]

4.3.5 Control Status Register (RPCS)

Address = 776714



[6]Not yet implemented.

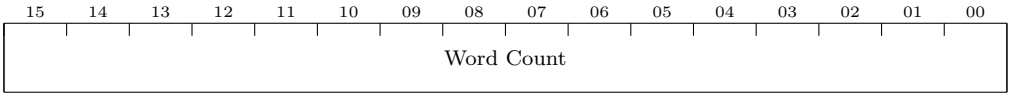
Bit	Designation	Description and Operation
00	Go	When set, causes the RP11-D to act on the function contained in bits 01 through 03 of the RPCS; when set, it sets Not Ready bit (which does not appear to be in any register; perhaps the Ready bit in the CSR is the inversion of that). Write-only, always reads as 0 (it is not stored; rather, the level during the bus write operation is used as a pulse).
01-03	Function	The function to be executed when Go is set. `Idle/Reset` 000 `Write` 001 `Read` 010 `Write Check` 011 `Seek` 100 `Write (no seek)` 101 `Home Seek` 110 `Read (no seek)` 111
04-05	Memory Extended Address	A 2-bit extension to RPBA giving an 18-bit bus address. If 22-bit addresses are enabled, these two bits are replicated as bits 00 and 01 of RPXA.
06	Interrupt on Done (Error) Enable	When set, causes an interrupt to be issued on various conditions.[7]
07	Ready	Controller is ready to perform a new function; read-only.
08-10	Drive Select	Specify the drive for any controller command.
11	Header	Not applicable to the QSIC/USIC.[8]
12	Mode	Not applicable to the QSIC/USIC.[9]
13	Attention Interrupt Enable	Allows the RP11-D to generate an interrupt when any Attention bit (in RPDS) is set.[10]

[7] Should audit the code and list all the conditions that can generate an interrupt.
[8] Currently the Header bit is ignored but it probably should generate some sort of error.
[9] Currently the Mode bit is ignored but it probably should generate some sort of error.
[10] Not yet implemented.

Bit	Designation	Description and Operation
14	Hard Error	Set when any error other than a data error is set; read-only.
15	Error (ERR)	Set when any error is set; read-only.

4.3.6 Word Count Register (RPWC)

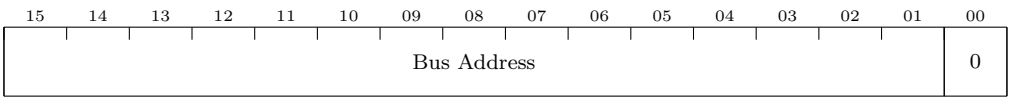
Address = 776716



Bit	Designation	Description and Operation
00-15	Word Count	The 2's complement of the number of words to be transferred by a function. The register increments by one after each word transfer. When the register overflows to 0, the transfer is completed and the RP11 function is terminated.

4.3.7 Bus Address Register (RPBA)

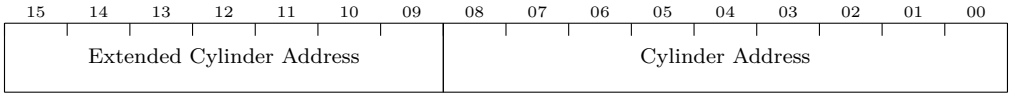
Address = 776720



Bit	Designation	Description and Operation
00-15	BA00-BA15	The low 16-bits of the bus address to be used for data transfers. The MEX bits (bits 04 and 05 of RPCS) extend the address to 18-bits and, if enabled, the BAE bits (bits 00-05 of RPXA) extend the address to 22-bits. Bit 00 is always 0 as all transfers are a full word.

4.3.8 Cylinder Address Register (RPCA)

Address = 776722

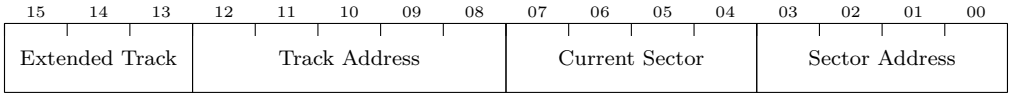


When in RP02/RP03 emulation mode, bits 0-8 are read-write, and 9-15 are unused.

Bit	Designation	Description and Operation
00-08	Cylinder 00-08	The cylinder number when emulating an RP02/03.
09-15	Extended Cylinder 09-15	The high bits of the cylinder number, when emulating an extended pack.

4.3.9 Disk Address Register (RPDA)

```
Address = 776724
```

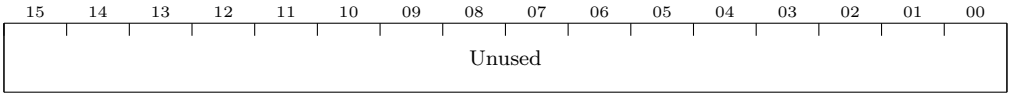


Used for all operations other than Home Seek. Seek uses only the Track Address.

Bit	Designation	Description and Operation
00-03	Sector Address	The disk sector to be addressed for the next function.
04-07	Current Sector	Notionally, the current sector address of the currently selected drive; read-only. On the RP11-D, connected to a free-running counter; the data is of no validity.
08-12	Track 00-04	The track number, when emulating an RP02/03.
13-15	Extended Track 05-07	The extended track number, when emulating an extended pack.

4.3.10 Maintenance 1 Register (RPM1)

```
Address = 776726
```



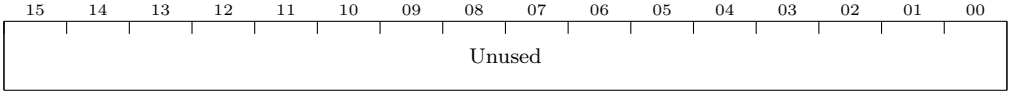
This register is currently unimplemented in the RP11-D.

Bit	Designation	Description and Operation
00-15	Unused	Unused

4.3.11 Maintenance 2 Register (RPM2)

Address = 776730

This register is currently unimplemented in the RP11-D.

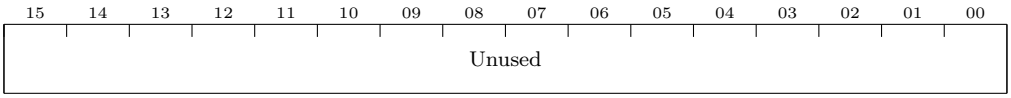


Bit	Designation	Description and Operation
00-15	Unused	Unused

4.3.12 Maintenance 3 Register (RPM3)

Address = 776732

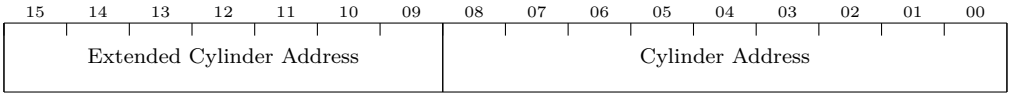
This register is currently unimplemented in the RP11-D.



Bit	Designation	Description and Operation
00-15	Unused	Unused

4.3.13 Selected Unit Cylinder Address(SUCA)

Address = 776734

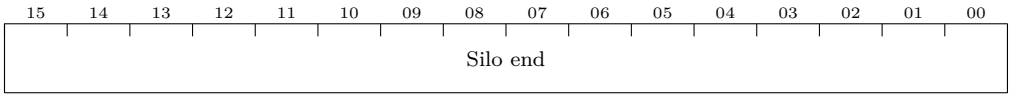


This register appears to be read-only. (Need to check the prints.)

Bit	Designation	Description and Operation
00-08	Cylinder 00-08	Contains the cylinder address of the selected drive (I think!) when emulating an RP02/03.
09-15	Extended Cylinder 09-15	Contains the extended cylinder address when emulating an extended pack.

4.3.14 Silo Memory Buffer Register (SILO)

Address = 776736



Bit	Designation	Description and Operation
00-15	Silo	This register, when enabled by maintenance (currently un-implemented) allows reading from and writing to the FIFO connecting the RP11-D to its storage device.

Chapter 5

Enable+

The Enable+ is an implementation of the Able ENABLE board inside the USIC. The ENABLE board gave access to 2^{22} bytes of main memory much like the PDP-11/70 for processors that only had 18 bits of addressing.

The Able ENABLE board worked by having three bus connections, one UNIBUS coming in, one UNIBUS out, and a memory bus. The USIC simplifies this by having all memory on-board and doesn't actually need a UNIBUS out.

There are two maps in the Enable+, one for accesses from the processor and the other used when devices are doing DMA operations.