

Low Memory Addresses

Plan for PowerPC system

Tue, Apr 4, 2000

Introduction

The PowerPC version of the system code will use many low memory addresses that are actually unavailable in that system because of the needs of VxWorks. In order to maintain a compatible set of addresses between the PowerPC and 68K version of the system, low memory addresses are mapped to higher memory on the PowerPC version. To an outside user, accessing low memory results in actual access to high memory. To achieve this, a request for memory data that using sufficiently small addresses is converted into addresses in the corresponding high memory area that holds analogous data structures. This note describes which accesses need to be supported in this way.

Overall plan

The lowest megabyte of 68K memory is to be treated in the above method, so that accesses to the first megabyte of memory are mapped into higher memory, say, 0x00E00000–0x00FFFFFF. This will mean that the first 14MB of memory is available to VxWorks, which is expected to be sufficient for its needs. Sticking to 6-digit addresses has a practical advantage when entering addresses into the analog control field of the ADESC table, and also when using the memory dump page.

Low memory structures

Here is a list of various structures that will be included in low memory, along with their 68K low memory addresses.

<i>Address</i>	<i>Name</i>	<i>Size</i>	<i>Use</i>
000400	VRAMIMAG	0200	Ascii buffer for little console displays
000600	ABORTDP	0060	Abort diagnostics (??)
000660	GIDPTR	0020	Generally Interesting Datapool (??)
000680			
0006C0	SERTABL	0008	Serial port connections table SERTABL
0006C8			
000700	TASKTABL	0100	Task Table (??)
000780	TTDPTR	0004	Ptr to task timing diagnostics table
000784	GLOBALS	0004	Base of system globals (68K A5 register)
000788	DATETIME	0008	Current time in BCD yr,mo,da,hr,mn,sc,cy,ms
000790	INTSLEDA	0004	I/O address of interrupt LEDs
000794	TASKLEDA	0004	I/O address of task LEDs
000798	CNT2000	0001	2000Hz decremting counter byte
00079A	LAESAV	0002	Enable Bit# saved around call to Loc Appl
00079C	SYSDATE	0004	System version date in BCD form yyyyymmdd
0007A0	MOTABLE	0060	Motor table (room for 10 motors)
000800	SBugArea	0200	Not needed for VxWorks
000A00	ZEROS	0200	Source of zeros for invalid idents
000C00	DCTABLE	0100	Digital Control table (room for 31)
000D00	TABLEDIR	0100	System table directory copy
000E00	BETABLE	0100	Bus error table
000E80	Globals	0180	Global variables (68K A5=000F80)
001000			

0011C0	SRMTABLE	0240	SRM Status table
001400	XRNODES	0400	Scratch ident array for ACReq (??)
001800	ALLOCD	1000	ALLOC, FREE diagnostics (??)
002800	PROTO	0100	Foreign protocol table
002900			
002F80	CLKEVTB	0040	Clock event bit map tables
002FC0	IPSCT_C	0010	A/D scan diagnostics for IP_C
002FD0	IPSCT_D	0010	A/D scan diagnostics for IP_D
002FE0	IPEVT_B	0020	Event diagnostics for IP_B
003000	CLKEVTS	0800	Clock event times
003800	SETLOG	0800	Settings Log data stream
004000	SWFTCMND	0800	Swift digitizer commands data stream
004800	ETRDATA	0400	Elapsed times for Data Access Table entries
005000			
005800	TFTPLOG	0800	TFTP Log data stream
006800	FNCAPT	0800	Foreign node capture diagnostics buffer
007000	TASKLOG	1000	Task Activity Log data stream
008000	AERSLOG	1000	AERS Acnet Alarms Log data stream
009000	RETDLOG	1000	RETDAT Log data stream
00A000			
010000	FMONLOG	10000	FMON network capture diagnostics area
020000			
060000	NETRCV	20000	Ethernet Datagram Receive buffer (128K)
080000	NETXMT	20000	Ethernet Datagram Transmit buffer (128K)
0A0000	ARCRCV	20000	Arcnet Receive buffer (128K)
0C0000	ARCXMT	C000	Arcnet Transmit buffer (48K)
0CC000	ARCTPLH	2000	Arcnet TPL area (8K)
0CE000	ARCPQ	2000	Arcnet OUTPQ area (8K)
0D0000	NAMETAB	10000	Hashed Name Table (64K)
0E0000			

All the cases of data streams listed above are established via the DSTRM table entries. The system code only knows about one or two data stream queue addresses, in case it has to initialize nonvolatile memory from scratch, upon finding a completely zero table directory.

The network tables are placed at addresses according to an algorithm that is used by the network diagnostic page application PAGENETF to determine which datagram is a received datagram and which is a transmitted datagram, as both are recorded in the NETFRAME data stream. The algorithm is to AND the address of the datagram with 0x20000; if the result is nonzero, it is a receive buffer, else it is a transmit buffer. In order to reserve 128K memory for the datagram buffer, we use NETRCV at 0x60000 and NETXMT at 0x80000. We use ARCRCV at 0xA0000 and ARCXMT at 0xC0000, with ARCTPL and ARCPQ located at 0xCC000 and 0xCE000. This uses 256K for the transmit and receive buffers for both ethernet and arcnet.

The address of the circular diagnostic buffer used by the FMON local application is specified via two parameter words. This is only a diagnostic. We must be careful.

System Tables

The system table directory is full of addresses, by definition. Each entry contains the base address of that system table. But all these addresses will have to be made correct for the PowerPC system. If we do it manually, we have to be careful to maintain the proper checksum. It might be easier to use the appropriate listype for writing to such entries. The correct addresses must be installed somehow.

Certain system tables may have need for low memory addresses in some of their fields. These may include ADESC, RDATA, BADDR, LATBL, DSTRM, TRING.

The ADESC table may have fields that hold low memory addresses in the analog control field. This would be unusual, but there may be a few such cases. The routine MAPTO32 is used for most mappings of the analog control field into 32-bit addresses. This routine could also check for low memory accesses and map them appropriately. We could also modify each such case manually, which would also include modifying the appropriate DABEL input SDR records.

The RDATA table may include memory addresses in the second long word of an entry. It may be rare to find low memory addresses in such entries, and they would likely only be read from, and not written to, so we can probably check for such occurrences manually before admitting them into a PowerPC system. As an example, checking through the entries in node0610 for memory addresses, there is a VME I/O address 0xFFFF6100 that must be modified—not for low memory mapping, but rather for VME I/O space mapping. There is a use of the address 0xF35 that holds the length of a 15Hz cycle in half ms units. A similar case is the address 0xF68 that has something to do with the little console global variable data. There is also a low memory address of 0x2FA2 that is in the clock event bit map table; this address occurs in the third long word of the type 0x2A entry that copies memory words.

The BADDR table of binary byte addresses may have low memory addresses. It would be easy to modify the RBINARY routine to check for such addresses, applying the offset necessary to reach what is the likely intent. The other attitude is to make sure that we check the content of this table before using it in a PowerPC system.

The LATBL includes parameters of local application instances. In a few cases, there can be pairs of parameter words that comprise memory addresses. We need to check for such cases and adjust them before enabling such local application instances.

A DSTRM table entry includes the address of the corresponding data stream queue. These are often diagnostic queues that may be in low memory. During initialization, each DSTRM entry is checked, and the queue area is prepared for use. This code could be modified to check for a low memory address and replace it for subsequent use. This would actually alter what is in the DSTRM entry, so that it would no longer be suitable for a 68K system, but it simplifies the rest of the data stream access code.

The TRING table does not seem to have low memory addresses within it. The parts that have to do with token ring will not be used in the PowerPC system.

For all these system tables, the proper approach is probably to manually modify low memory addresses to the mapped high memory address. This will limit the modifications in the system code that will be needed. We have always had to be careful before in entering proper addresses in all nonvolatile system tables as a matter of course during system configuration. There is no strong reason why we should not continue to take such care with the PowerPC version.

Settings

There is a group of setting routines that are invoked to cause settings to memory. They are SETMB, SETMW, SETMFIFO, SETMMAP, SETMLVF, and SETMWVF. After retrieving the expected memory address, each will need to check for low memory accesses to see whether an adjustment to higher memory is needed. This case is more significant than some others, because it involves writing to memory. Accidentally writing into VxWorks memory may cause us to endure a kind of debugging session we would not want to endure.