Sandpoint Linux Memory Size Considerations

by Maurie Ommerman
CPD Application
Freescale Semiconductor, Inc.
Austin, TX

This application note describes the process of selecting the correct memory size value for various Sandpoint PCI mezzanine card (PMC) processors. This information also applies to the Freescale PCI mezzanine card (MPMC), which is the same as the PMC. This application note describes the steps for selecting the appropriate memory size for Sandpoint Linux. Sandpoint is an evaluation board for the PowerPC™ processors.
1 Memory Selection

The Linux kernel for Sandpoint does not automatically size memory. Memory size is fixed by a statically defined variable in both the 2.4 and 2.6 kernels. It is necessary first to determine the physical memory size of the PMC on the system, either by looking at the splash screen for DINK32 (a small operating system debugger for the Sandpoint evaluation board) or through the DINK32 command, ab. An example is shown as follows:

```
I/O system initialized...
Memory Enabled: [ 128MB at CL=3 ]
Caches Enabled: [ L1I(16K), L1D(16K) ]
Register Inits: [ 32 GPRs, 32 FPRs, 222 SPRs ]
    #    #    #
    #    #    #
    #    #    #
    #    #    #    #    #
    #    #    #    #    #    #
    #    #    #    #    #    #
    #    #    #    #    #    #
    #    #    #    #    #    #
                        #    #
                        #    #
                        #    #
                        #    #
                        #    #
                        #    #
                        #    #
Version : 13.1.1, Metaware Build
Written by : Freescale’s RISC Applications Group, Austin, TX
System : Sandpoint X3 with Unity (MPMC8245)
Processor : MPC8245 V1.1 @ 400 MHz, 133 MHz memory
Memory : Map B (CHRP) 128MB at CL=3

As shown in this example, the physical memory size is 128 Mbytes. PMCs continue to be released with more and more physical memory. The first PMC was released with 32 Mbytes, the MPC7xx series was released with 64 Mbytes, and the MPC74xx and MPC824x were released with 128 Mbytes. Newer PMCs will be released with 256 Mbytes or more. When a Linux kernel is built, it is necessary to specify the actual memory size on the physical device when setting the memory size in the kernel. If they do not match, system crashes, Linux segmentation dumps, or other unpredictable results may occur. Note that a discrepancy does not necessarily cause a system to fail immediately.

1.1 Linux Kernel 2.4

The sandpoint_find_end_of_memory(void) function sets the memory size. In the 2.4 kernel, this function is found in arch/ppc/platforms/sandpoint_setup.c. Find the line that returns the memory size, usually at line number 430. Change the line ‘return 32*1024*1024;’ to the physical memory size. For example, if there are 128 Mbytes of physical memory, change the line to ‘return 128*1024*1024;’.

1.2 Linux Kernel 2.6

The sandpoint_find_end_of_memory(void) function sets the memory size. In the 2.6 kernel, this function is found in arch/ppc/platforms/sandpoint.c. Find the line that returns the memory size, usually at line

Sandpoint Linux Memory Size Considerations, Rev. 1

Freescale Semiconductor
number 486. Change the line ‘return 32*1024*1024;’ to the physical memory size. For example, if there are 128 Mbytes of physical memory, change the line to ‘return 128*1024*1024;’.

1.3 Summary

In both the 2.4 and 2.6 kernels, the sandpoint_find_end_of_memory(void) function must be modified to return the actual physical memory size.

2 References

2. Unity X4 (MPMC8240/MPMC8245) Configuration Guide
3. Valis X3(MPMC745x) Configuration Guide
4. Creating a Linux ‘Out of the Box’ Experience on a Sandpoint Platform (AN2578)
5. Adding new PowerPC Processors and Bridge Chips to Linux (AN2650)
6. Understanding the Linux Kernel, Daniel P. Bovet and Marco Cesati, available from O’Reilly Publishing

The first five references are available on the Freescale web site listed on the back cover of this document.

3 Document Revision History

Table 1 provides a revision history for this application note.

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Change(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Initial release.</td>
</tr>
<tr>
<td>1</td>
<td>Non-substantive formatting.</td>
</tr>
</tbody>
</table>