Running Windows CE "XIP" on the ÉlanSC400 Evaluation Platform

Dave Tobias, AMD

Introduction and Goal of Document

There has been a lot of recent interest on how to use the Microsoft Windows CE operating system with x86 based platforms. This article provides the steps for generating an XIP (eXecute In Place) image of the CEPC DEMO7 project for use on the AMD ÉlanSC400 **version 3.x** evaluation board. The ÉlanSC400 is a highly integrated, system on a chip which boasts 486/100 MHz CPU performance, fully integrated PC/AT compatibility, power management, PCMCIA, and much more. The ÉlanSC400 **version 3.x** evaluation board is a software development platform, which is available directly from AMD.

The information given here assumes that the release version of the Windows CE 2.0 Embedded Toolkit (ETK) for Visual C++ 5.0 has been freshly installed onto an NT based build station. From this base installation it will be shown how to build a Flash-resident CE image of the DEMO7 project, burn it into the ÉlanSC400 evaluation board application Flash banks, and then run it XIP on this platform. A set of files, including a working boot loader for CE running on the ÉlanSC400 evaluation board, is available for use in conjunction with this document. All supporting files may be obtained by requesting them via email from EPD.SUPPORT@AMD.COM. In the interest of brevity, is assumed that you have a working knowledge of the basic CE setup and build process. Further detail on this subject is available by sending a request referencing this article to the above email address.

Some History

One of Microsoft's ETK goals was to provide a fast-path build process for several demo projects for each supported CPU. Engineers could use these samples as a starting point for their own products. To support the ETK, a standard platform was developed for the demos: the Hitachi D9000 (ODO) board. Through the use of a swappable CPU card, this single hardware platform could run CE on several CPU types such as the SH3, MIPS, PPC, etc. However, until shortly after CE 2.0 was released, there was no X86 CPU card available for the ODO platform. Thus, while all other CPU types were tested with the ODO board, the x86 CE kernel was tested on a standard PC using drivers written especially for the standard PC/AT compatible peripherals.

The term "CEPC" is often used loosely to refer to the operating system itself, and this has sometimes lead to confusion. CEPC is really several things bundled together. First and foremost, however it is the same 32 bit, multitasking Windows CE operating system that runs on the ODO platform, HP/Cs, and many other existing platforms. It is not a DOS program, an emulation of CE, or any other such thing. In fact, the main difference between the CEPC and ODO environments is that CEPC relies on PC/AT BIOS to initialize the system, and PC DOS (along with LOADCEPC.EXE) to place the CE image into RAM.

The decision to use a PC as the standard platform for X86 had some decided advantages for getting x86 support into the ETK quickly. While running from RAM is a great sanity check, and can be used to develop drivers on a standard PC while the target hardware is being built, a "real world" product running CE may not want to include BIOS or DOS. Standard CEPC requires both of these, and the ETK does not explain how to eliminate the requirement for them. This can be a real problem for those x86 customers wishing to test Windows CE in a "real product" type environment on platforms such as the AMD ÉlanSC400 evaluation board. What's needed is the option to have CE boot up and run XIP from Flash on an x86 platform. The goal of this document and associated software collateral is to enable this as a new x86 build option. Individual steps are given for performing each task followed by more detail as necessary.

Baseline the Basic OS Build Process by Creating a "Run From RAM" Image

GOAL: To build the standard ETK CEPC DEMO7 image, and run it from RAM on the ÉlanSC400 evaluation board as a sanity check that the ETK is installed properly.

1. On the NT-based build machine create a CEPC DEMO7 build window shortcut by copying the x86 DEMO1 shortcut which was installed by the ETK, and then renaming it to "CEPC DEMO7". Now open the shortcut properties, and change WINCE.BAT's "DEMO1" command line parameter to "DEMO7".

2. Double click on the new shortcut to open a CEPC build window for DEMO7. At the DOS command prompt, set the environment variable CEPC_DDI_VGA8BPP=1.

3. Build the operating system image by running BLDDEMO.BAT from the \WINCE directory. The resulting operating system image file is called NK.BIN.

4. Ensure your ÉlanSC400 evaluation board has a standard keyboard, PS/2 mouse, and ISA VGA card installed, and then boot it to DOS. Note that the LOADCEPC.EXE utility requires HIMEM.SYS to be in your CONFIG.SYS.

5. Copy the NK.BIN image to the ÉlanSC400 evaluation board.

6. Use the ETK's LOADCEPC.EXE utility to load the O/S image into memory and then pass control to it as follows: LOADCEPC NK.BIN [enter]. After a few seconds you should see the full handheld PC (H/PC) demo run.

The above steps take you through the basic process of building the DEMO7 project on the ETK for the CEPC platform. Before changing anything on the ETK to support Flash based operations, you should familiarize yourself with building the ETK CEPC DEMO7 project for x86. Note that a standard CEPC NK.BIN image will run on the ÉlanSC400 evaluation board with no modifications since the ÉlanSC400 device is fully PC/AT compatible. The resulting image assumes that you do not have an ISA version of one of the graphics cards (such as those based on the S3 Trio64 chip, Tseng labs, or ATI) for which high resolution CE drivers exist. Thus, the CEPC_DDI_VGA8BPP environment is used to configure a lower resolution graphics mode of 320x200x8BPP.

The NK.BIN file for the DEMO7 project can be 5 MB or larger. Perhaps the best file transfer method is to use a network card (such as the AMD PCNET-ISA card) in one of the ÉlanSC400's ISA slots to communicate with the build machine. Instructions for getting the PC NET-ISA card set up on the ÉlanSC400 evaluation board under DOS may be obtained from EPD.SUPPORT@AMD.COM.

Build and Install the CE Boot Loader

GOAL: To build and install the Windows CE boot loader that is supplied by AMD for the ÉlanSC400 evaluation platform, version 3.x.

1. Use the supplied MAKE.BAT file to build the boot loader using Borland TASM/TLINK, and Paradigm Locate.

2. Configure JP 36 on the ÉlanSC400 rev 3.x evaluation board so that ROMCS0 is routed to the 4MB Resident Flash Array (RFA). Now run E4FFLASH.EXE as follows: E4FLASH cs0:F0000 BOOTLOAD.BIN [enter], and then E4FLASH cs0:3F0000 BOOTLOAD.BIN [enter].

3. Configure the boot ROM width (SW3[2]) to the 16-bit position.

4. Cycle power to test the installation. If properly installed, you should see the port 680h/80h LEDs be set to 01C5. Following this, you should see the sign-on screen from your graphics card's VGA BIOS (if applicable). If you don't get 01C5 on the LEDs, or if the system hangs, power off the evaluation board and wait 3 seconds, then power back on.

By the time that Windows CE gets control of the system, it expects the x86 CPU to be in 32 bit protect mode, and the system hardware to be initialized. Standard CEPC relies on system BIOS to handle the details of coming up at reset. Removing the BIOS requirement is nontrivial since the default CE 2.0 ETK tools and build environment does not easily support x86 CPUs at boot time. To work around this, a stand-alone boot loader was developed for the

ÉlanSC400 evaluation platform. This boot loader is built outside of the Windows CE tools environment. Building the example boot loader requires Borland Turbo assembler 4.0+, Borland TLINK.EXE 7.0+ and Paradigm Locate version 5.10a+. After getting control directly from the x86 reset vector, the boot loader initializes the CPU, chip set, and graphics card before passing control to CE.

The boot loader code was intentionally minimized for quicker understanding. Thus, the sample boot loader may require additions to the ÉlanSC400 initialization tables, etc. to support any new features that you may add. The boot loader code targets the ÉlanSC400 version 3.x evaluation board, and requires changes to run on version 2.x boards. The boot loader performs the following tasks:

- Places the ÉlanSC400's 486 CPU core into 32-bit protect mode using a flat memory model.

- Enables system DRAM. This is hard-coded to support the DRAM that comes standard with the 3.0 evaluation board. A dynamic memory sizing routine for the ÉlanSC400 is available upon request from EPD.SUPPORT@AMD.COM.

- Performs main ÉlanSC400 CSC register initialization which handles ISA bus setup, PIRQ routing, etc.

- Sets up the system memory map. The ÉlanSC400's MMS windows E and F are used to provide contiguous access to Flash banks 1 and 2 starting at 32 MB in system address space. After MMS window initialization, the system memory map appears as follows:

Start Address	Stop Address	Resource
FFFFFFF0	FFFFFFFF	Boot vector hook: Far jmp to boot loader code below 1MB
02800000	FFFFFFEF	Unused
02400000	027FFFFF	4 MB accessed via MMSF, which generates ROMCS2
02000000	023FFFFF	4 MB accessed via MMSE, which generates ROMCS1
01000000	01FFFFFF	Space for up to 16 MB more of system DRAM
00100000	00FFFFFF	15 MB system DRAM for windows CE
000F0000	000FFFFF	Boot loader code/data (ROMCS0)
000D0000	000EFFFF	Maps to ISA bus
000C0000	000CFFFF	Reserved for graphics adapter ROM
000B0000	000BFFFF	Maps to ISA bus
000A0000	000AFFFF	VGA graphics buffer
00000000	0009FFFF	System RAM used only by boot loader

- Initializes VGA graphics. The boot loader assumes the use of an ISA based graphics adapter that has an onboard BIOS extension ROM. The VGA BIOS on the graphics adapter is called to initialize the graphics adapter.

- Sets up a pointer to the CEPC parameters. The LOADCEPC.EXE utility takes some command line parameters. Since the boot loader replaces BIOS, DOS, and LOADCEPC.EXE, it must perform all applicable tasks of these software components. The boot loader maintains a 4 byte global array containing the emulated LOADCEPC.EXE parameters.

- Passes control to the CE image. This is done by jumping to offset 0 of the CE image at absolute physical address 32MB.

Since the boot loader will reside in ROM/Flash the use of a locator is required. The file BOOTLOAD.CFG has been

provided for use with Paradigm Locate. Once the boot loader is built, the E4FLASH.EXE utility can be used to write it into one of the ÉlanSC400 evaluation board's ROMCS0 boot Flash banks. For brevity, only the installation of the boot loader to the 4MB ROMCS0 RFA is discussed. This allows system BIOS to remain in the 128Kb DIP device so that you can easily boot either CE or DOS. The 4MB ROMCS1 and ROMCS2 RFAs (8MB total) are available for the XIP Windows CE image.

Upgrade the Base CEPC Files

GOAL: As shipped from the factory, the ETK does not expose the x86 page tables, which are required in order to support XIP operation. The following steps apply a patch that expose the page tables to the programmer.

1.   Patch the ETK. First, change to the \WINCE directory, then: PKUNZIP –D PAGETABL.ZIP [enter] .

2.   Do a complete re-build of the ETK so that you can verify the build environment is still OK. To do this, delete the \WINCE\RELEASE directory, and then close the CE build window. Now re-open the window, and re-build per the steps given earlier.

Step 1 above applies a Microsoft-supplied patch to the ETK to allow programmer access to the x86 page tables. This is required in order to support XIP operation because the pre-patch ETK page tables are hard coded and do not provide mappings for the Flash where CE will XIP from. The patch copies several files into the \WINCE directory structure including \WINCE\PLATFORM\CEPC\KERNEL\HAL\x86\OEMINIT.ASM. This new file contains the code that sets up the x86 page tables. Note that the patch simply exposes the page tables for potential modification. No SC400 specific mappings are included with the patch's OEMINIT.ASM.

It is advised that you do a full rebuild of the ETK after applying the patch to ensure sanity. Assuming a successful build, ascertain that OEMINIT.ASM was really built by verifying the presence of the following file: \WINCE\PLATFORM\SC400\KERNEL\HAL\OBJ\x86\i486\CE\RETAIL\OEMINIT.OBJ.

Create an ÉlanSC400 Specific Build Window and Platform Directory

GOAL: Reduce impact of SC400 specific changes on the rest of the build tree by creating a new SC400 "platform" directory. Also, make an ÉlanSC400 specific DEMO7 build window shortcut.

1.   Create an ÉlanSC400-specific DEMO7 build window shortcut by copying the existing CEPC DEMO7 shortcut to one named "SC400", and then replacing the new shortcut's WINCE.BAT "CEPC" parameter with "SC400".

2.   Create a new platform subdirectory for the ÉlanSC400 by going to the \WINCE\PLATFORM directory, and typing in: XCOPY CEPC SC400 /e /i [enter]

3.   Update the \WINCE\PLATFORM\SC400\KERNEL\HAL\X86\OEMINIT.ASM file with the ÉlanSC400 evaluation board specific version of this file supplied by AMD.

The easiest way to create the shortcut mentioned in step 1 is to copy the CEPC DEMO7 shortcut that you made earlier, rename it, and then change the properties. In the copy, change the WINCE.BAT command line's "CEPC" parameter to "SC400". Note that "SC400" MUST be entered verbatim. If not, the case sensitive comparisons that are done in the build environment's batch files will fail. Now you have an icon that will open an SC400-specific DEMO7 build window.

The reason for step 2 above is to allow your SC400 changes to exist in their own subdirectory. This way, if you make a mistake you can easily start over from a known good platform. For this step, the case of the directory name doesn't matter.

Step 3 above is required because the default OEMINIT.ASM that's provided by the ETK patch doesn't know that the ÉlanSC400 evaluation board's memory map will include 8 MB of Flash starting at 32MB. The modified version supplied by AMD contains the appropriate page table entries.

Upgrade CONFIG.BIB and BLDDEMO.BAT

GOAL: The default CEPC DEMO7 CONFIG.BIB and BLDDEMO.BAT files will not support XIP images, so update them with files provided by AMD.

1. Copy the CONFIG.BIB file supplied by AMD into the \WINCE\PLATFORM\SC400\FILES directory.

2. Copy the BLDDEMO.BAT file supplied by AMD into the \WINCE\PUBLIC\DEMO7\OAK\MISC directory.

Some ETK build environment changes were required in order to support x86 platform XIP operation, and these changes affect the CONFIG.BIB and BLDDEMO.BAT files. The following text explains the changes that were made to the original files to support XIP operation. **The changes do not detract from the original ability to build an image that will run from RAM**.

The CONFIG.BIB file tells the CE locator where in the memory map the executable code (i.e. kernel, etc.) will be located, where system RAM resides, and provides other information pertinent to the build process. The locator uses this address information when providing internal code reference fixups. For a given hardware platform, the locator must use different addresses to fix up the internal references when running XIP from Flash as opposed to when running from RAM. The IMGFLASH environment variable was added to CONFIG.BIB to allow build time selection of RAM versus Flash memory maps. When the environment variable IMGFLASH=1, the image will be built with the Flash based memory map. Otherwise the RAM memory map will be used.

Also, the new CONFIG.BIB file causes a "flat" binary image (NK.NB0) to be generated when CE is built with IMGFLASH=1 (see the CONFIG.BIB "ABX" option in the ETK docs for more info). Note that NK.BIN is not a simple, flat binary image. Instead, it's a collection of binary images, each with its own header, concatenated into a single file. The AMD E4FLASH.EXE utility does not know about the .BIN file format, so a true "flat" image is required.

The BLDDEMO.BAT file must also be updated to remove the explicit setting of the IMGFLASH environment variable to NULL. We will want the option to set this environment variable manually from the DOS command line. Also, the original ETK build environment knows about 2 platforms: "ODO" and "CEPC", and BLDDEMO.BAT compares an environment variable that holds the platform name to these string constants. If the platform name is not "CEPC", then the ODO platform is assumed. Thus the comparison must also look for the string "SC400", or the build will fail. It should be noted that because BLDDEMO.BAT is not located beneath the SC400 platform directory, it is not an ÉlanSC400 specific file. However, the changes made will not negatively effect the build of DEMO7 on other platforms.

At this point you should be able to build the RAM based x86 based CE image using the **SC400** build window. As a sanity check it is suggested that you do a from-scratch rebuild (and test run) of the RAM based version of the OS in order to test the installation of the changes. The rebuild process is similar to the CEPC baseline build procedure described earlier. Note that a RAM based image built using the SC400 DEMO7 build window will be identical to a RAM based image built with the original using the CEPC DEMO7 build window. The SC400 images only differ when you set IMGFLASH=1.

Build a Flash Based Image

GOAL: You have verified that you can build a RAM based CE image using an SC400-specific build window. Now build the XIP image from the still-open build window as follows:

1. Configure the environment variable to enable XIP builds: set IMGFLASH=1[enter].

2. From the \WINCE\RELEASE directory, just type MAKEIMG[enter]. The XIP image will be made in about 30 seconds, and will be 8MB in size. You can find it in \WINCE\RELEASE, and it will be called NK.NB0.

Building a Flash based image at this point is simply a matter of running the locator again using the Flash-specific memory mappings from CONFIG.BIB. That's why the above steps tell you to set the environment variable IMGFLASH=1 and then run MAKEIMG. MAKEIMG runs the locator, and then combines the resulting binaries into the CE NK.NB0 image.

Burn the ÉlanSC400 Evaluation Board RFA and Boot the Image

GOAL: To write NK.NB0 to the ÉlanSC400 evaluation board RFA banks and then boot and execute CE from Flash.

1. Either copy the image to the ÉlanSC400 evaluation board, or (from the ÉlanSC400 board) mount your \WINCE build directory as a remote disk. Change directory so the current directory on your evaluation board contains the NK.NB0 image.

2. Make sure that the ROMCS1 and ROMCS2 jumpers (JP37 and JP38) point to their respective RFA banks, not to the DIP Flash.

3. Using E4FLASH.EXE version 1.01 or later, enter the following commands at the ÉlanSC400 evaluation board's DOS command: E4FLASH CS1:0 –s0 –f3FFFFF nk.nb0 [enter]. Then type E4FLASH CS2:0 –s400000 – f7FFFFF nk.nb0 [enter].

4. Ensure that the CS0 jumper and x8/x16 boot ROM switch are properly set (see instructions above).

5. Connect a null modem cable from the 9 pin "D" connector that's silk-screened "SERIAL" on the ÉlanSC400 to a terminal emulator which is set up for 115.2k, N, 8, 1.

6. Power the ÉlanSC400 evaluation board down, wait 3 seconds, and then power it up. Watch the terminal emulator as CE boot-up messages are displayed. The evaluation board should boot to the H/PC desktop GUI within about 15 seconds.

The NK.NB0 file will be burnt into the x16-1 and x16-2 Flash banks on the ÉlanSC400 revision 3 evaluation board. Due to the large file sizes involved with CE builds, it's recommended that you network the evaluation board to the build station, and mount the build station's \WINCE directory as a remote drive on the ÉlanSC400 evaluation board. Now the image can be written to Flash directly from the mounted disk.

NK.NB0 is an 8MB image that needs to get burnt across two 4 MB Flash banks. E4FLASH.EXE version 1.01 (included in the files provided by AMD) and later can burn portions of a file into Flash. Before attempting to burn in the image you must ensure that the ROMCS1 and ROMCS2 jumpers are properly installed to route these chip selects to the x16-1 and x16-2 Flash banks respectively. To ascertain this, invoke E4FLASH.EXE with no parameters. At the bottom of the resulting help screen E4FLASH.EXE will display what devices it has auto-detected for each of the ÉlanSC400 ROMCSx signals.

Once all of the above steps are completed you should see lots of messages coming from CE as it boots. Some of the messages may indicate that errors have occurred but this is normal. Experience will tell you which error messages really require corrective action.

cexipsml.doc amd/dft 09/20/1998