DESIGN CONSIDERATIONS FOR THE EMBEDDED PC

Embedded Systems Conference West
San Jose, California
by Eric Auzas
09/14/95

INTRODUCTION

The acceptance of the PC Architecture into the business world has made it ideal for embedded PCs that are low cost and quick time-to-market. The vast availability of both PC hardware and software, development tools, and PC expertise provides engineers with a well defined platform. The design of an embedded PC requires understanding of all the hardware and software pieces. Depending on the application, the embedded PC may or may not require the complete functionality of a desktop PC. Some applications will require full PC compatibility while others may require only a subset. This paper describes the development tools, hardware and software design considerations for developing a cost-effective and quick time-to-market embedded PC.

WHAT IS AN EMBEDDED PC?

The embedded PC can be very different from a traditional desktop PC. It can be completely hidden from the user with no display and no user input. Examples include single-line LCD screens with a keypad input. Even though these PCs appear to be no different than a microcontroller-based design they have the distinct advantage of being based on the PC platform. This allows the designer to use PC expertise, PC development tools, and desktop PC s for both software and hardware development.

BASIC PC SYSTEM REQUIREMENTS AND COMPATIBILITY

The minimum configuration required for a DOS-based embedded PC is an Intel architecture processor, an 8254 timer, an 8259 interrupt controller and memory. A system based on only these components may not run MS-DOS* but could run General Software Embedded DOS* or Datalight ROM-DOS*. This would still allow the software to be developed on a desktop PC.

There are several considerations that must be taken in determining if full PC compatibility is required in an embedded PC. For instance, in a portable design, it may not be necessary to support a floppy disk, thereby eliminating the requirement for the DMA channels typically used for floppy disk data transfers. Early understanding of the requirements of the design and possible PC compatibility tradeoffs can provide for a more cost-effective design.

It is important to determine if off-the-shelf software applications or software developed in-house will be used in the design. Software developed in-house provides more flexibility in the hardware design, whereas the ability to execute all off-the-shelf software requires full PC compatibility. Not all applications have the same hardware requirements as seen in Table 1 which allows hardware design flexibility in some off-the-shelf software applications.

Functionality	**8254**	**8237**	**8259**	**RTC**	**8242**	**8250**	**PC Video**
MSDOS*	X	X	X	X	X		X
Embedded DOS*	X		X				
ROM-DOS*	X		X				
MS Windows* 3.1	X		X	X	X		EGA,VGA
Smarterm/Procomm	X		X		X	X	X

Table 1. Example of a Compatibility Table

EMBEDDED PC SOFTWARE CONSIDERATIONS

The embedded PC architecture is composed of three layers. The bottom layer consists of the PC hardware, for example the CPU, Real Time Clock, DMA, Interrupt Controller, and various other devices depending on the compatibility required. One level up from the hardware is the BIOS which provides low-level drivers to interface to the hardware. Above the BIOS is the Disk Operating System (DOS) which provides a service of organizing files, disk functions, I/O functions, and launching applications. On top of these three layers resides the application. Due to the amount of time it takes to access the hardware through the BIOS or DOS, many software applications access the hardware directly. Figure 1 illustrates the three layer model and how the application software bypasses the other layers.

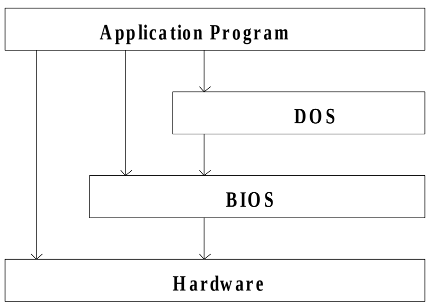


Figure 1. Three Layer Model

BIOS

The BIOS is hardware dependent and typically requires some amount of changes for the embedded design. It is important to understand some of the basic components that make up a BIOS.

Typically a BIOS consists of seven main components; the Boot Vector, the Power On Self Test referred to as POST, the Boot-Strap, the BIOS compatibility address map, the BIOS Interrupt Service Routines (ISRs), the BIOS Device Service Routines (DSRs) and some configuration tables. The standard PC BIOS is 64K in size, located from F0000H to FFFFFH at the top 64K of the real mode address region(1MB). Many embedded PC BIOS vendors allow a range of sizes from 8K to 64K depending on the options required.

Boot Vector

The Boot Vector is typically found at the top of the processor address space. After a reset due to either a software reset or a hardware reset, the processor jumps to the processor specific reset vector. The BIOS will contain a jump instruction (4 bytes) to pass control to the POST. The remaining bytes are used to indicate the BIOS manufacturer date and the PCs ID number.

Power On Self Test

The POST contains tests and initialization routines for the hardware components. The POST can vary in size depending on the system hardware. For example, if the system does not require an 8042 keyboard controller, the keyboard/keyboard controller initialization code does not need to be compiled into the BIOS. Two POST tests that are worth mentioning are the CMOS Shutdown byte and the Optional ROM extension scan. The CMOS shutdown byte is a location in the CMOS RAM area of the Real Time Clock that provides information in determining the cause of a reset. The BIOS uses this shutdown byte to exit protected mode and skip the hardware tests. More information on this byte is discussed in the RTC section under EMBEDDED PC HARDWARE CONSIDERATIONS.

Another important feature in the POST is the ROM extension scan which allows easy software upgradability or additions to a BIOS without any knowledge of the current POST. These ROM extensions are typically used for ISA bus cards to install BIOS functions and initialize hardware on the card. During the POST, a search is conducted for ROM extensions located between C8000H to E0000H. The BIOS searches every 2K boundary for the word 55AAH which indicates a valid ROM extension. If a valid ROM extension is found, a ROM checksum is conducted to validate the contents, and then the ROM extension code is executed. The ROM checksum requires byte 0 to the length minus one of the ROM extension to have a sum of zero. The last byte forces the checksum addition equal to zero.

Offset in the ROM Extension	Data
C8000-C8001	55AA
C8002	Length of extension. Number of 512 byte blocks, divisible by 4
C8003-End-1	Code
End	8 bit value to force the checksum value equal to zero.

Table 2. ROM Extension Example

These ROM extensions are used quite frequently for installing ROM-based DOS like Datalight's ROM-DOS* or Microsoft's DOS* in ROM. Microsoft's ROM Windows* also uses this method for placing a 16K stub in real mode for installation. M-Systems has an option of installing their TFFS driver as a ROM extension.   The ROM extension allows for a modular method of adding software to the embedded system.

After the POST, the BOOT strap loader or INT19 is executed and will search for a bootable disk in the system. The equipment list will indicate the number of floppy disks and hard disks available. If a flash disk is available for boot, it can install itself at the end of the list during its ROM extension installation. The first sector of the bootable disk is loaded into memory and the BIOS passes control to this data which in turn loads DOS.

BIOS Compatibility Table

The BIOS Compatibility table is a table of BIOS entry points that date back to the original PC/XT BIOS. This table needs to be installed to maintain compatibility with application software that calls on these entry points. The table resides from FE000H to FFFFFH and is optional with some BIOS manufacturers..

BIOS Interrupt Service Routines

BIOS Interrupt Service Routines (ISRs) are invoked by hardware interrupts from peripheral devices. They handle the low level software interface between peripheral requests and the BIOS.

BIOS Device Service Routines

BIOS Device Service Routines (DSRs) handle software generated interrupts. These interrupts can be generated either by the BIOS, DOS, MS Windows*, or the application software. Each interrupt number provides a device service with many sub-functions below it.   The function number desired for a particular interrupt is placed in the AH register and any other information required is placed in the remaining registers.

In an embedded PC BIOS, depending on the hardware, specific functions can be excluded to reduce the size of the BIOS if not needed. Obviously there would be a tradeoff with off-the-shelf software compatibility.

If power management is needed, there is an Advanced Power Management (APM) specification that defines functions for the BIOS. APM defines five modes of operation; Full On, APM Enabled, APM Standby, APM Suspend, and Off.

Full On: No system power management is being performed. All devices on.

APM Enabled: System is operating but power management is active. System clocks may be slowed or disabled and unused devices may not be powered.

APM Standby: After a short period with no activity, Standby is entered. Most power management features are active. The current operating parameters are retained, allowing rapid recovery to the APM Enabled state when activity resumes.

APM Suspend: After a long period with no activity, Suspend is entered. All power management functions are active for minimal power consumption (clocks stopped, etc.). The current operating state is saved, resulting in a slow recovery to the Enabled state.
Off: System power supply is off. Operational parameters are not stored. A full system reset is performed before reentering the Full On state.

Currently Microsoft DOS and Microsoft Windows* have the ability to use these features.

The following table is a comparison of third party vendor BIOSs.

Features	Phoenix Technologies	Award Software	System Soft	AMI	General Software	Eurosoft	Annabooks	USA Teknik
PCMCIA	Yes	Yes	Yes	Yes	Planned	Planned	Planned	Yes
FFS/FTL	Yes	Yes	Yes	Yes	Q1 95	Yes	No	Yes
APM	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Min Size	12KB ROM	64KB ROM	N/A	64KB ROM 1MB RAM	8KB ROM 4KB RAM	N/A	48KB ROM 4KB RAM	N/A
Free Source	Optional	Optional	No	Optional	Yes	Optional	Yes	Partial
Remote Floppy	No	No	No	No	Yes	Yes	Yes	Yes
Video/KBD to Serial	No	No	Yes	Yes	Yes	Yes	Yes	Yes
OEM Configurable	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Debugger	Yes	Yes	No	Paradigm	Yes + SSI	Yes	Yes	No

Table 3. BIOS Comparison

Disk Operating Systems (DOS)

The DOS operating system offers functions for I/O communication, floppy/hard disk, video, keyboard, program handling, memory management, and network support. DOS consists of many functions that are available to the embedded user. These functions are out of the scope of this paper and are found in several books listed in the Reference section.

Two companies that have written their own version of DOS specifically for embedded designs are Datalight and General Software. Both are compatible with a version of Microsoft DOS. Microsoft offers both a ROM version and a disk version of DOS. The Microsoft DOS is distributed through Annabooks.

Generally no modifications need to be made to the DOS since it is not hardware dependent like the BIOS. Many vendors bundle a mini-command.com with an approximate size of 10K bytes if size is an issue. Below is a list of the DOS vendors:

	General Software Embedded DOS*	Microsoft MS DOS ROM	Datalight 6.0 ROM DOS*
MS DOS Compatible	MS6.22 Features	MS3.3, MS5.0,MS6.X	MS6.2
Support APM	No	Yes MS5.0, MS6.X	Yes
ROMable	Yes	Yes	Yes
Size Min	32KB ROM 8KB RAM	MS3.3 45K MS5.0 61K ROM 256K RAM	39KB ROM 8KB RAM
Source	Available	Parts Available	Parts Included
OEM Configurable	Yes	Yes	Yes
Remote Disk via Serial	Yes	No	Yes
Option Disk Compression	N/A	Double Space* (6.22)	Stacker*

			Optional

Table 4. DOS Comparison

Another feature for increased performance and memory savings is eXecute In Place (XIP). It allows code to be executed directly from where it is stored. MS DOS* in ROM ROM Windows 3.1*, and GEOWORKS supports XIP.

Graphical User Interface (GUI)
MS ROM Windows* and GEOWORKS exist today if a Graphical User Interface is a requirement.

Microsoft ROM Windows

Microsoft ROM Windows* is very similar to the disk based version found on many desktops. The Microsoft Windows* 3.1 ROM Development Kit (RDK) is available from Annabooks.   The ROM based Windows contains a small amount of XIP code in the real mode address space and a large amount of XIP code above the 1MB address region. The disk based Windows contains no XIP code since all code is loaded from disk.   Two modes of operation exist for ROM Windows, standard mode and enhanced mode. In standard mode Windows, shell programs, applets, fonts and other Windows resources all execute from the XIP memory. In Enhanced mode, Windows executes both from RAM and XIP memory.

Windows Mode	**Min RAM**	**Min XIP**	**Min Disk Space**
Standard	1MB	2MB	0MB
Enhanced	2MB	3MB	2MB

Table 5. ROM Windows requirements

The XIP memory can be provided with Flash Memory like the 28F016 or 28F008. If this is the case, a software utility will be needed to load the ROM Windows files into the flash. Since disk space is a requirement for enhanced mode, a flash file system software is required. The appnote "Implementing Mobile Intel486™ SX CPU PC Designs Using FlashFile™ Components" (order number 292149-001) goes into great detail on the hardware and software specifics for a Microsoft ROM Windows*/Microsoft DOS* in ROM based design using the Microsoft Windows* 3.1 RDK.

GEOWORKS

The GEOS* System Software by Geoworks is a windowed based OS that is targeted to consumer based products. It executes only in real mode and uses XIP windows to access programs. It is a very compact OS with many OEM configuration features. The GEOS OS requires a BIOS and a DOS to function.

Flash File Systems

There are two methods of implementing a file system in flash, one is the Flash File System (FFS) developed by Microsoft and the other is the File Translation Layer (FTL) that is supported by several companies.

FFS uses linked lists to keep track of files. The system can be broken down into three parts. First is the File System Redirector (FSR) which intercepts DOS disk operations from an application and translates them before sending them on to the File System Driver (FSD). Second is the File System Driver, which accepts operations from the FSR. The FSD organizes the data according to the storage architecture and passes low level commands like Read, Write, Copy, and Erase to the Device Driver. Finally the Device Driver accepts low level commands from the FSD and interfaces to the hardware.

The FFS makes the flash drive appear like a network drive to the system. Network drives do not use the standard BIOS function call INT13 to talk to the disk. This causes some problems for applications that perform direct calls to the INT13 BIOS function as they will not be supported by FFS.

FFS also requires an ISA sliding window to access the flash. The window size can be 8KB, 16KB or 32KB and located in the C0000H to DFFFFH address range..

FTL is a sector based file system, like DOS, which allows the software to treat the flash as a normal sector based drive with a sector size of 512 bytes. When modifing a sector, the software remaps the sectors or block to a free area of flash while invalidating the old area. The location of the remapped block is also recorded. Typically an FTL implementation is approximately 20K in size. FTL is also defined in the PCMCIA specification. Depending on the hardware, three methods are available for implementing FTL.

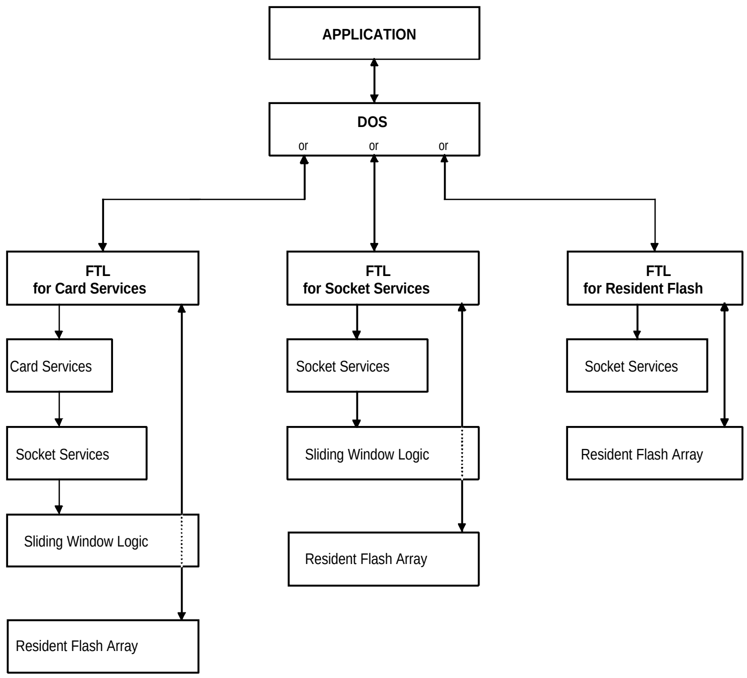


Figure 2. FTL Implementations

The first method uses PCMCIA and requires Card Services, Socket Services, and ISA Sliding Window logic (PCMCIA Controller). The second method requires Socket Services, and ISA Sliding Window logic (PLD) for the FTL to communicate to the flash array. The third method uses protected mode to access the flash and does not require an ISA sliding window. It is also able to communicate directly to the flash because of integrated flash drivers for the 28F008SA, 28F016SA, etc. FTL implementations intercept INT13 BIOS calls allowing for a higher level of compatibility with application software.Below is a list of various flash file system vendors:

	Microsoft FFS	**SCM Microsystems**	**Datalight Cardtrick**	**M-Systems True FFS**
Type	FFS	FTL	FTL/FFS	FTL
Device Driver	Yes	Yes	Yes	Yes
BIOS Extension	No	Yes	Yes	Yes

Table 6. FTL Comparison

EMBEDDED PC HARDWARE CONSIDERATIONS

Intel Architecture Processor

For full PC compatibility an i386™ processor with an 82C206 peripheral chipset provides basic PC compatibility. The 206 includes the 8259, 8237, 8254 and in some cases the RTC. The 206 is available from a number of vendors including Siemens SAB82C206, and PicoPower PT82C206F-LV. In addition, a chipset for DRAM, and ISA bus, is also required like the PicoPower PT86C378, or Opti 82C283. A one device solution with many of the above features (206 + DRAM + ISA) includes the Western Digital WD8110LV, the Chips and Technology F82C836, or the Samsung KS82C388A chipsets.

The Intel386™ EX processor embedded processor is a highly integrated 386 core with both PC and embedded peripherals. The Intel386 EX processor has a 26-bit address bus providing a 64MB address space. The interrupt controller, timers, DMA channels, and serial ports are all PC-AT compatible. The embedded functionality of the Intel386 EX processor consists of a synchronous serial port, DRAM refresh control, chip selects, power management, I/O ports, a watchdog timer, and a JTAG interface. The Intel386 EX processor alone can run a variety of BIOSs and DOSs. Implementing MS Windows* on the EX will require a keyboard controller, real time clock, video controller, and a DRAM controller.

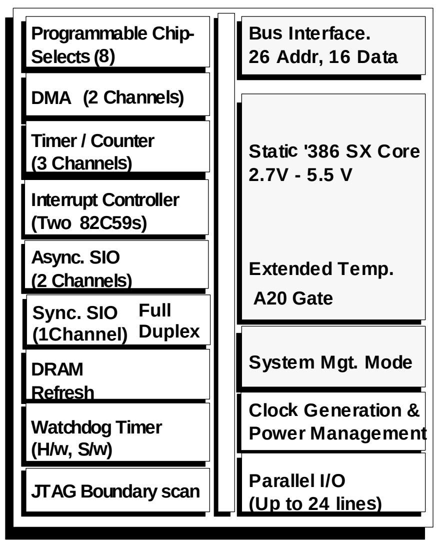


Figure 3. Intel386™ EX Block Diagram

The Intel386 EX Processor Point of Sale reference design is a good example of an embedded design that supports MS Windows*.

Display

If system display is required, several possibilities exist with varying degrees of PC support. The standard PC displays consist of Monochrome, CGA, EGA, and VGA. The VGA controller is the most widely available due to the PC market. VGA controllers exist for CRTs or LCDs or both simultaneously.

If a CRT display is required, a VGA controller is the most widely available with resolutions ranging from 640x480 to 1024x768. VGA controllers are backwards compatible thereby providing all of the PC display standards. A VGA BIOS is required for controller initialization and video functions. The size of the VGA BIOS is typically 32K. Possible tradeoffs include, leaving a video controller out of a design if no display is required. If only a text display is required, the display can be rerouted through the serial port or a small LCD single line character display can be added. A full LCD controller can also provide VGA resolution. Below are the PC compatible video modes that have PC software support:

Display	ROM	Colors	Resolution	RAM	Mem Address
MDA	Integrated in Sys BIOS	None	Char only	4K	B0000-B0FFF
HGC	Integrated in Sys BIOS	None	720x348	64K	B0000-BFFFF

CGA	Integrated in Sys BIOS	16	640x200	16K	B8000-BFFFF
EGA	16K C0000H-C3FFF	64	640x350	64K-256K	A0000-BFFFF
VGA	32K C0000H-C7FFF	256K	720x480	256K	A0000-BFFFF
Super VGA	32K C0000H-C7FFF	256K	1024x768	512K-1MB	A0000-BFFFF

Table 7. PC Video Modes

RTC

In the PC architecture the Real Time Clock contains a battery clock and CMOS RAM. The clock keeps time after power has been removed and also stores the system configuration in CMOS RAM. If the system is never to be powered off, then an RTC may not be required. If the system is not expandable, then the configuration information can be stored in ROM.

CMOS RAM Function	**RAM Locations**
Time/Date	00H-09H
RTC Control Regs	0AH-0DH
Diagnostic Byte	0EH
Reset Code Byte	0FH
Diskette Type	10H
Reserved	11H
Hard Disk Type	12H
Reserved	13H
Equipment Installed Byte	14H
Base & Extended Memory	15H-18H
Reserved	19H-2CH
Additional Flags	2DH
Checksum Value	2EH-2FH
Memory above 1MB	30H-31H
Century	32H
System Information	33H
Reserved	34H-3FH

The original RTC used in the PC AT was a Motorola MC146818. Now several manufacturers supply equivalent products like Dallas Semiconductors DS1287 and BenchMarq. The RTC contains 64 bytes of CMOS RAM that are accessed using two I/O locations port 70 and 71. Port 70 is the address register and port 71 is the data register. Valid address values are 0 to 3FH for 64 CMOS locations. The first 10 locations are used by the RTC to update the time and date, the next four locations are control registers for the RTC, and the remaining 50 locations are used to store system configuration.

The Reset Code Byte(0FH) was originally used in the PC-AT to allow the 80286 processor to return from protected mode to real mode by using the processor reset. This byte would indicate why the processor was reset and has several possible values:

00H Normal power up reset or <CTRL> <ALT> <DEL> reset.
04H Skip POST
05H Skip POST, preserve memory, send an EOI to the Interrupt controller, and then jump to reset vector 0040:0067. .
09H Block move return
0AH Jump to reset vector 0040:0067 without issuing EOI.

The location 0040:0067 contains the address where Real Mode execution should resume. On a Intel386™ processor or Intel486™ processor based system the switch from protected to real mode can be made without resetting the processor, instead the Protected Enable(PE) bit in the Processors MSW register can be disabled. Some BIOSs allow this selection.

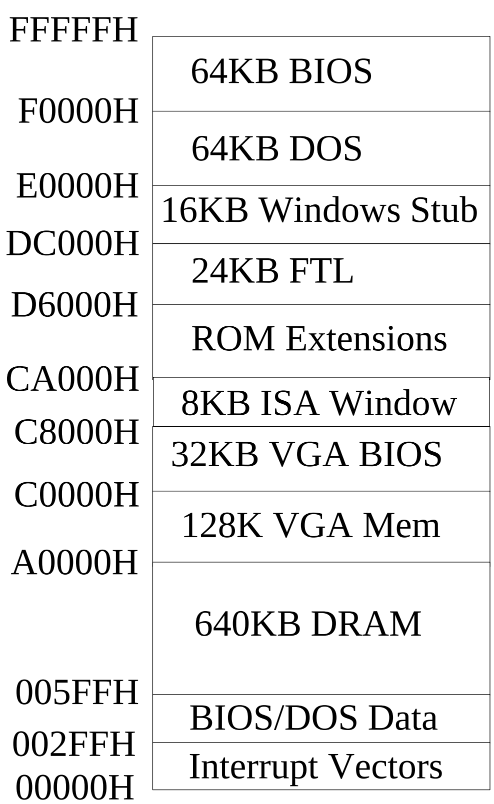
If the design does not require a real time clock then it is possible to use the PE bit to switch from protected mode to real mode. To maintain compatibility the BIOS can be hard coded with the hardware configuration. If the design requires MS-DOS* or MS Windows* than an RTC is a requirement.

Keyboard or Keypad

If data or user entry is not required a keyboard controller like the Intel8242PC/WA/WB is not required. For development purposes the data or user entry can be temporarily rerouted through the serial port for debug. MS DOS* and MS Windows* both require a PC standard keyboard controller and keyboard, although for limited data entry a keypad may suffice. This requires both non-standard hardware(keyboard scanner) and software (scanner driver).

Memory

Memory size is dependent on the application. Typically the BIOS and DOS will require 10K of RAM with 2K for the interrupt vectors.   MS DOS* requires a minimum of 256K while MS Windows* requires 2MB.



Storage

There are several types of media for data storage and include Hard disk, floppy, PCMCIA, Flash, or ROM. It should be determined if the media need read, or read/write capability. Storage requirements depand on the amount of data to be stored, size constraints, power, reliability, removability, and other factors. Below is a partial table of storage possibilities:

Media	**R/W**	**Storage**	**Removable**
Hard Disk	R/W	5MB-1GB	No

PCMCIA	R/W	10MB-40MB	Yes
Flash Array	R/W		No
Floppy	R/W	360K-2.88MB	Yes
ROM	R		No
CD-ROM	R	Huge	Yes

Boot Block Flash/ROM

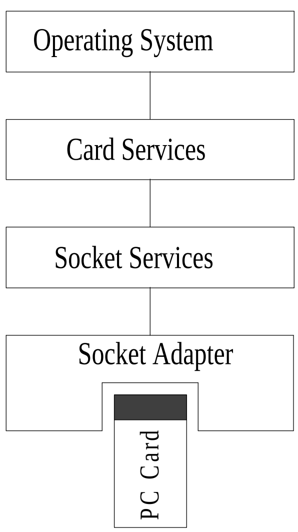
Boot block Flash or ROM is used for storing the BIOS. The boot block flash has the advantage of programming the BIOS after the board has been manufactured, thus allowing longer software development time and field upgrades. To control the BIOS programming on the flash, a flash utility is typically placed in the boot block of the flash prior to placement on the board. An example of this is the EV386EX evaluation board which comes with a flash utility call iBOOTLDR. Source code for this program is available on the Intel BBS (916-356-3600).

Flash Array

A Flash array allows for data storage and is located above the 1MB real mode address region or can be paged with ISA Sliding Window logic. Access to the flash can be accomplished in either protected mode or virtual mode. This memory can be used to store ROM Windows or a flash file system.

PCMCIA

The Personal Computer Memory Card International Association (PCMCIA) specification defines both memory and I/O cards. The cards, referred to as PC cards, are approximately 3.5" by 2" with a thickness that can vary from 3.3mm to 10.5mm depending on the type. These PC cards are ideal for portable embedded devices or embedded instruments. They allow users to upgrade software, increase data storage, and add I/O devices. To interface these PC cards to an embedded design, a PCMCIA controller is required as well as a software interface. Below is the software model.

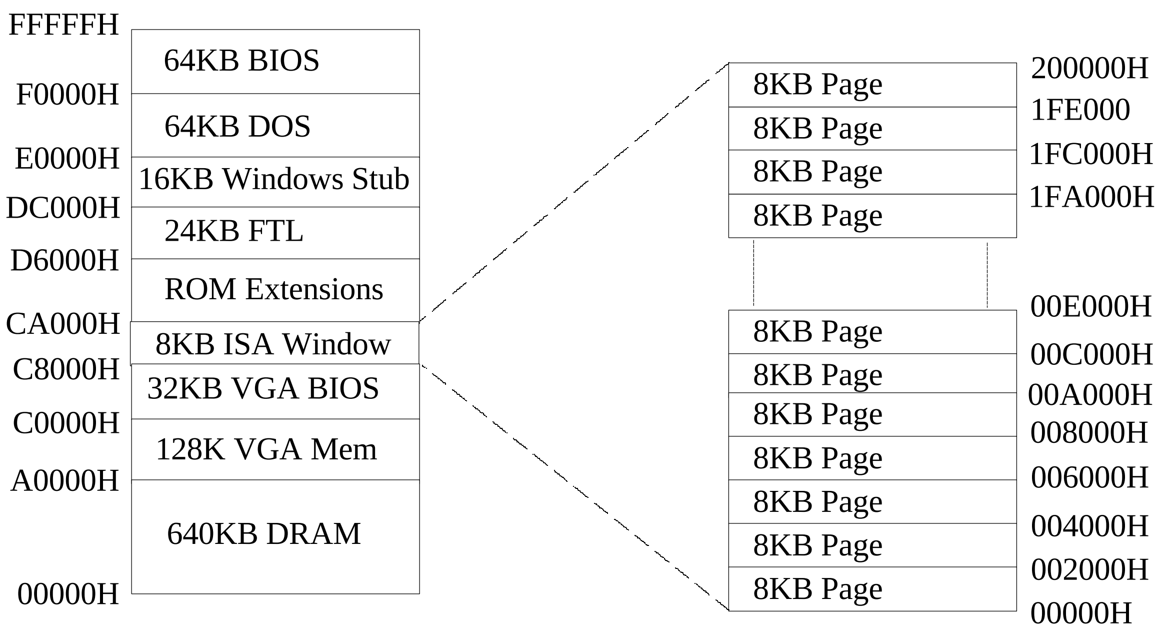


A variety of controllers exist from companies like Cirrus, Databook, and Vadem. Most interface to either the ISA, EISA, or PCI bus. Most controllers are based on the Intel 82365SL which is also compliant with the Exchangeable Card Architecture (ExCA™) or QuickSwap™ specifications. QuickSwap has replaced ExCA as the standard for Intel Architecture. QuickSwap guarantees that a mimimum hardware and software interface exists in the system. This allows the PC architecture to have the ability to communicate with a wide variety of cards. Software drivers like Socket Services and Card Services are required for the interface to operate correctly.

The Socket Services provides a standardized interface to the socket hardware, but is independent of the hardware implementation. This software resides in the ROM BIOS and provides a variety of functions. The Card Services provides a higher level of functions that applications or operating systems can call on for interfacing to the card. To be fully compatible with most cards, the system software must support PCMCIA version 2.1 and QuickSwap. If the system is using flash memory, the drivers may also want to support the Memory Technology Drivers interface. This provides future support for updates when new types of Flash memory cards appear on the market. It is also possible to use a flash file system with a flash card.

ISA Sliding Window

The ISA sliding window allows real mode programs access to memory above the 1MB address space. Typically this window is built around the LIM (Lotus-Intel-Microsoft) Expanded Memory Specification (EMS). Special hardware and software is required to support EMS. The ISA sliding window can range from 4KB to 64KB in size and there can also be multiple windows in real mode.



Flash file systems can use a window like this to access a flash array.

CONCLUSION

There are a variety of options available for the embedded PC designer today. Selecting and understanding the correct configuration is very important to a successful design. Issues like PC compatibility, storage, memory map, hardware requirements are just some of the factors involved. The subjects covered in this paper were designed to provide you with the basic building blocks to design an embedded PC.

REFERENCES

Vendor	Product	Vendor	Product
American Megatrends, Inc Contact: Rajesh Motwani 42808 Christy St Suite 114 Freemont CA 94539 Tel: (510) 249-1000 Fax: (510) 770-0423	BIOS	Needham Electronics Contact: Brian Neal 4630 Beloit Dr. Suite 20 Sacramento CA 95838 Tel: (916) 924-8059 Fax: (916) 924-8065	Flash Programmer
Annabooks University 11848 Bernardo Plaza Ct., #110 San Diego, CA 92128-2417 Tel: (800) 462-1042 Fax: (619) 673-1432	BIOS/DOS	Paradigm Systems Contact: Rick Narrow 3301 Country Club Dr. Suite 2214 Fendwell, NY 13760 Tel: (607) 748-5966 Fax: (607) 748-5968	Debugger
Award Software 777 East Middlefield Rd Mountain View, CA 94043-4023 Tel: (415) 968-4433 Fax: (415) 968-0274	BIOS	Phoenix Technologies Contact: Peter Baldwin 40 Airport Parkway San Jose, CA 95110 Tel: (408) 452-6581 Fax: (408) 452-1985	BIOS
Concurrent Sciences 530 S. Asbury P.O. Box 9666 Moscow Idaho 83843 Tel: (208) 882-0445 Fax: (208) 882-9774	Debugger	Promice ROM Emulator Grammar Engine Inc. 921 Eastwind Dr. Suite 122 Westerville, OH 43081 Tel: (614) 899-7878 Fax: (614) 899-7888	ROM Emulator
Datalight Tech Contact: Drew Gislason 307 N. Olympic Ave Suite 201 Arlingtion WA 98223 Tel: (206) 435-8086 Fax: (206) 435-0253 BBS (206) 435-8734	mini-BIOS/DOS CardTrick*	System Soft Contact: Sue Agranoff 313 Speen St Natick MA 01760 Tel: (508) 651-0088 Ext 230	BIOS/FTL
EuroSoft 4th Floor Hanover House 136 Old Christ Church Rd. Bournemouth BH11NL UK Tel: 44(0)202 297315 Fax: 44(0)202 558280	BIOS	Systems & Software Inc. 18012 Cowan , Suite 100 Irvine, CA 92714-6809 Tel: (714) 833-1700 Fax: (714) 833-1900	Debugger/ Simulator
General Software P.O. Box 2571 Redmond WA 98073 Tel: (206) 454-5755 Fax: (206) 454-5744 BBS: (206) 454-5894	BIOS/DOS	USA Teknik 9030 Viscount Row Dallas, Texas 75247-5412 Tel: (214) 630-6600 Fax: (214) 630-6691	BIOS
GeoWorks 960 Atlantic Ave Alameda CA 94501 Tel: (510) 814-1660	OS	M-Systems 4655 Old Ironsides Dr. Suite #200 Santa Clara, CA 95054 Tel: (408) 654-5820	True FFS*
SCM Microsystems, Inc 985 University Ave, Suite #7 Los Gatos, CA 95030 Tel: (408) 395-9292	FTL-FFS*		

Intel386™ EX Embedded Microprocessor Specification
(Order Number 272420)
Intel Corporation
Literature Distribution Center
P.O. Box 7641
Mt. Prospect, IL 60056-7641
Tel: (800) 548-4725

Intel386™ EX Processor Users Manual
(Order Number 272485)
Intel Corporation
Literature Distribution Center
P.O. Box 7641
Mt. Prospect, IL 60056-7641
Tel: (800) 548-4725

Embedded Microprocessors
(Order Number 272396-002)
Intel Corporation
Literature Distribution Center
P.O. Box 7641
Mt. Prospect, IL 60056-7641
Tel: (800) 548-4725

Embedded Intel386™ Buyer's Guide
(Order Number 272474-001)
Intel Corporation
Literature Distribution Center
P.O. Box 7641
Mt. Prospect, IL 60056-7641
Tel: (800) 548-4725

Advanced Power Management (APM) BIOS Interface Specification Revision 1.1
(Order # 241704-001)
Intel Corporation
Literature Distribution Center
P.O. Box 7641
Mt. Prospect, IL 60056-7641
Tel: (800) 548-4725

Implementing Mobile PC Designs Using High Density FlashFile™ Components
Intel Application Note: AP-362
Intel Corporation
Literature Distribution Center
P.O. Box 7641
Mt. Prospect, IL 60056-7641
Tel: (800) 548-4725

PCMCIA PC Card Standard & Socket Services Interface Specification
PCMCIA
1030B East Duane Avenue
Sunnyvale, CA 94086
Tel: (408) 720-0107
Fax: (408) 720-9416

System BIOS for IBM PCs, Compatibles, and EISA Computers
Phoenix Technologies LTD.
Addison-Wesley Publishing Company

The XT-AT HANDBOOK
John P. Choisser & John O. Foster
Annabooks
15010 Avenue of Science, Suite 101
San Diego, CA 92128
Tel: (800) 462-1042
Fax: (619) 673-1432

Designing with FLASH MEMORY
Brian Dipert and Markus Levy
Annabooks
15010 Avenue of Science, Suite 101
San Diego, CA
Tel: (800) 462-1042
Fax: (619) 673-1432

PC INTERRUPTS
Ralf Brown & Jim Kyle
Addison - Wesley Publishing Company
Available from Annabooks

The Programmer's PC Sourcebook
Thom Hogan
Microsoft Press
16011 N.E. 36th Way
Box 97017
Redmond WA 98073-9717

Technical Reference Personal Computer AT
International Business Machines Corporation (IBM)
P.O. Box 1328-W
Boca Raton, Florida 33429-1328

The IBM PC from the Inside Out
Murray Sargent III & Richard L. Shoemaker
Addison Wesley
Available from Annabooks

AT BUS DESIGN
Edward Solari
Annabooks
15010 Avenue of Science, Suite 101
San Diego, CA
Tel: (800) 462-1042
Fax: (619) 673-1432

Peter Norton Programmers Guide to the IBM PC
Peter Norton
Microsoft Press
16011 N.E. 36th Way
Box 97017
Redmond WA 98073-9717

Intel386DX Microprocessor Hardware Reference Manual
Intel Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641
Tel: (800) 548-4725

PC Intern
Abacus
5370 52nd Street, S.E.
Grand Rapids MI 49512
Tel: (800) 451-4319

Microprocessor Report PC Chipsets Volume I & II, October 1993
MicroDesign Resources
874 Gravenstein Hwy. So., Suite 14
Sebastopol, CA 95472
Tel: (707) 824-4004

The PCMCIA Developers Guide
Michael T. Mori
Sycard Technology
Sunnyvale, CA 94086
Tel: (408) 247-0730

PCMCIA System Architecture
Mindshare Inc.
2202 Buttercup Dr.
Richardson, TX 75082
Tel: (214) 231-2216