The PCMCIA DSP Card: An All-in-One Communications System

Application Report

Raj Chirayil
Digital Signal Processing Applications — Semiconductor Group

SPRA145
October 1994

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

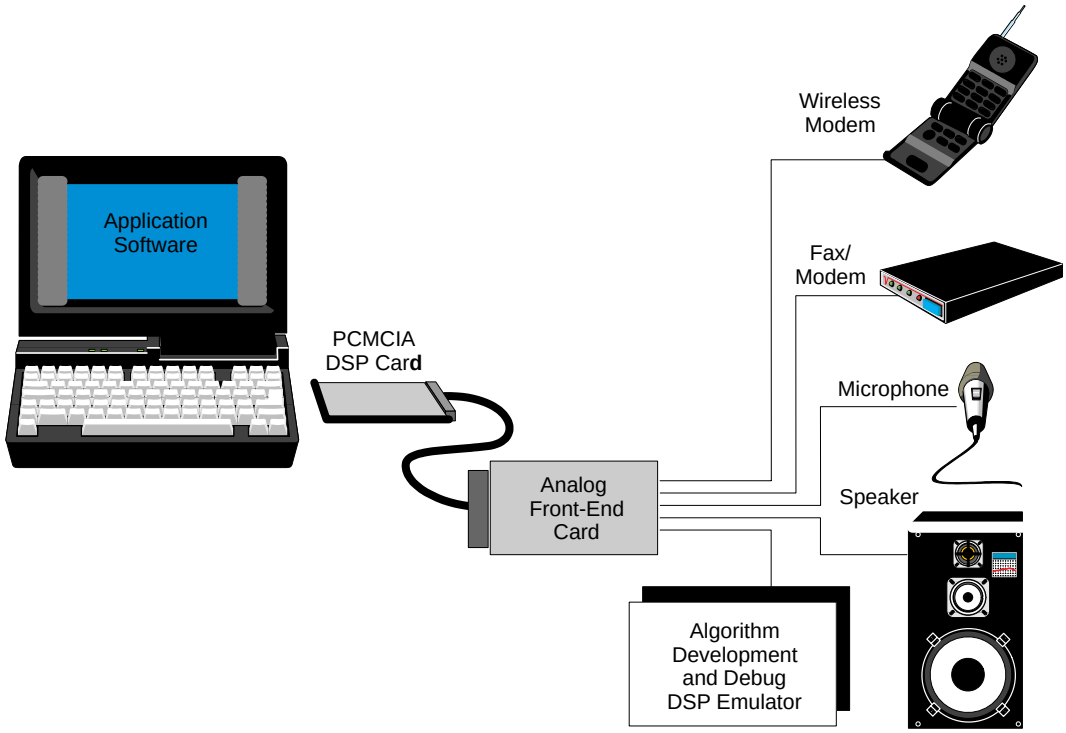
In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Introduction

With the advent of subnotebook computers and personal digital assistants (PDA), there is an ever increasing need for a universal communications engine that is compact, simple to use, and dynamically configurable to suit various operating environments. In the desktop computer world, there is rarely a need for portability, whereas in the world of PDAs, portability is everything. This includes not only the computer itself but also any peripherals that go with it. The Personal Computer Memory Card Interface Association (PCMCIA) standard has made a significant contribution toward meeting this requirement.

Figure 1. DSP Card Block Diagram



All of the PCMCIA cards available today are single-function products and lack the flexibility to be dynamically reconfigured to support multiple applications. The PCMCIA DSP card described here was defined to be used by the host CPUs as a generic coprocessor or as a storage device. For larger data storage applications, the on-board SRAM can be replaced by lower cost, low-power DRAM devices. However, advanced digital signal processing applications such as V.Fast modems and digital cellular phones require higher speed SRAMs to allow full-speed DSP operation. When appropriate algorithms are loaded, the host can transfer data to the on-board memory and command the DSP to perform specific tasks, such as handwriting recognition, image or voice data compression, or music synthesis. An external analog front-end (AFE) card can be connected to the DSP card if the application requires external analog input/output capability.

The architecture and design described here allow users to configure the card as a data/fax modem, speakerphone, telephone answering machine, note taker, character recognition system, or business audio card by merely downloading the appropriate DSP algorithm to the card. Replacing the wireline telephone

interface circuit on the AFE card with an RF circuit and antenna allows the same DSP card to support wireless data or voice communications.

A key requirement for any portable system is low power consumption. This DSP card uses a TMS320C51 DSP, which is ideal for the PCMCIA application because of its very low power consumption, high MIPS rate, and very low cost. Another important system requirement for a portable multifunction DSP card is the ability to provide processing power on demand. A card running a simple speech compression algorithm for a note taker may need less than 5 MIPS, whereas a voice-over-data system running a V.Fast or V.32bis modem and higher quality speech compression algorithm may need 40 MIPS or more. Because of the flexibility of the C5x DSP's clock input scheme, this design allows the host PC to configure the DSP to run faster or slower via s/w commands.

System Architecture

The DSP card system architecture is defined so that any algorithm developed for a particular DSP could be run on non-PCMCIA platforms, provided that particular DSP is available. The host system can treat the DSP card as a programmable function coprocessor. The system applications software needs only to know which DSP a particular algorithm is developed for. This allows the DSP card system to be integrated onto a notebook PC or PDA motherboard by merely replacing the PCMCIA interface with the appropriate host system interface.

The PCMCIA interface logic and additional logic needed for the communications and control are implemented in less than 5,000 gates in an FPGA. For highly integrated systems and motherboard applications, these functions can be easily integrated with the DSP as a single device through TI's TEC320 cDSP approach. The hardware- and host-independent architecture supports a Windows™ [1] application, using the DSP Resource Manager to run the same application and DSP algorithm on multiple platforms and hosts.

The architecture defines any host memory provided on the card to be shared between the host PC and the local DSP. This eliminates data bandwidth limitations and facilitates fast block transfer of data or downloading of DSP algorithms. Because of this dynamic algorithm loading capability, the host can treat the on-board DSP as a programmable function coprocessor. A real-time memory paging scheme makes it possible to load different application algorithms into different pages for fast reconfiguration and task switching.

The on-board FPGA arbitrates any conflicts for access of shared memory between the host PC and the DSP, with the host access having higher priority. The FPGA also implements all necessary host system interface and control logic. Several dedicated communication registers are provided in the FPGA to allow the host PC and the DSP to communicate without interrupting DSP operation. Buffered registers are provided in the FPGA for the required programmable bit I/O.

The PCMCIA DSP card interfaces to the host as a PCMCIA memory card and an I/O card. The PCMCIA specification supports up to 64MB of PCMCIA common memory in addition to a separate attribute memory space. For a 16-bit fixed-point DSP such as the TMS320C5x, this translates into 32M (16-bit) words of external program/data space. The attribute memory can be used by the DSP as 32M (8-bit) bytes of global data space. Both memories must obviously be paged by a DSP with only a 16-bit address.

1 Windows is a trademark of Microsoft Corp.

The DSP card architecture is expandable to support the full extent of PCMCIA memory, which the DSP can access in paged mode under software control. The paging feature allows users to load different application algorithms into different pages for dynamic reconfiguration and task switching. The PCMCIA common memory is mapped into the DSP's data/program space, and the PCMCIA attribute memory is mapped into the DSP's global data space.

This particular implementation limits the DSP's paged external data and program space to 3M words and global data space to 128K bytes. The DSP card is populated with two sets of fast (15-ns) SRAMs — one 64K-byte × 16-bit SRAM in one set and two 128K-byte × 8-bit SRAMs in the other. When the 64K-byte × 16-bit SRAM is enabled, it is used by the DSP as combined data and program memory. When the 128K-byte × 8-bit SRAM is enabled, it is used as separate 64K words of data and 64K words of program memory. The entire 256K bytes of memory are accessible by the PC in byte mode or word mode. However, the DSP can access only one of these memories at a time, as enabled by the system configuration register. The card also has 128K bytes of flash memory, which can be programmed by the PC. The DSP can be configured to boot load from this flash memory upon reset. Although the entire flash memory is mapped into the PC's attribute memory space, only 32K bytes are mapped into the DSP global data space at any given time.

The host PC can access the card as a 16-bit I/O device by writing to the configuration registers. The I/O address for the card is selectable by the PC in the card configuration registers. When the DSP card is configured as an I/O-mapped peripheral, the host communication registers are dual-mapped into the PC's common memory and I/O space.

Figure 2. DSP Card Architecture

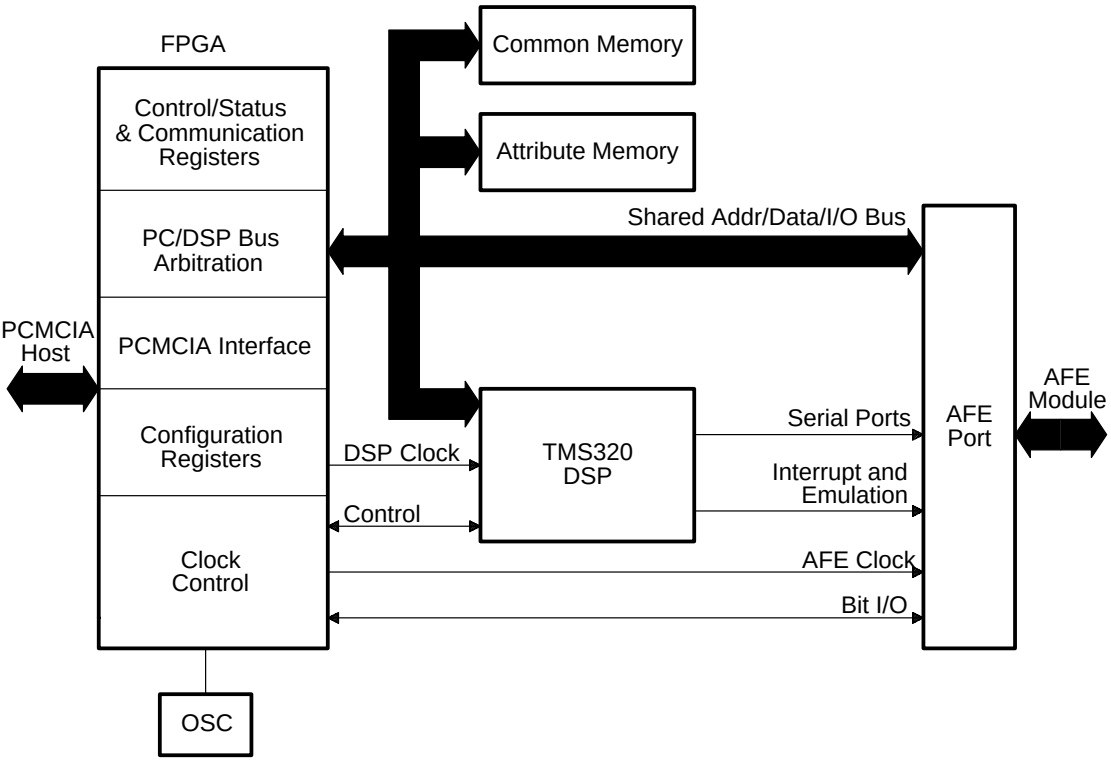


Figure 2 shows a block diagram of the DSP card. The PCMCIA connector appears on the left side of the board. The FPGA integrates all discrete logic in the system. The system clock to the DSP is provided by the FPGA for the control of the DSP's operating speeds. A 48-pin analog front-end (AFE) connector provides an external interface as well as system development and debug. The DSP serial port signals are available at this connector along with programmable input and output pins and DSP interrupt pins to monitor, configure, and control A/D converters, D/A converters, and other external devices. The connector also provides DSP emulation control pins to help DSP algorithm development on the card using the TI XDS510 emulation system.

Operation

The card's operating mode is controlled via the control, status, and communication registers in the FPGA. Some of these registers are accessible only by the PC, some only by the DSP, and some by both the PC and the DSP. These registers are mapped in the common memory space of the PC and I/O space of the DSP.

The DSP card can operate in two modes, the standard mode and the smart mode. In the standard mode, only the signature register (SIGR) is accessible to the PC. Other registers exist only when the card is in the smart mode.

Table 1. DSP Card Registers

	Memory Address		Access Type		
Register Names	**PCMCIA Common Memory**	**C5x (I/O)**	**PC**	**DSP**	**Register Definitions**
SIGR	000400h	----	R/W	----	Signature register — shadowed in the FPGA
Reserved	000000h	–----	----	----	
DSPCR	000002h	----	R/W	----	PC writes to DSPCR to control and configure the DSP card
DSPSR	000004h	----	R/W	----	DSP status register holds DSP operation and communication status
DSPTXD	000006h	0050h	R	W	PC/DSP communication register — buffers DSP's transmit data
DSPRXD	000008h	0051h	W	R	PC/DSP communication register — buffers data to be received by DSP
Reserved	0Ah–0Fh	----	----	----	
PCSR	----	0052h	----	R	DSP reads status of host communication from this PC status register
BIOR	----	0053h	----	R/W	DSP reads/writes this buffered register to create up to 16 bits of I/O
SYSCFG	----	0054h	----	R/W	DSP selects memory pages and clock speeds by writing to SYSCFG
Reserved	----	055h–058h	----	----	

Standard Mode

In the standard mode of operation, the DSP is reset and the clocks are turned off, disabling the DSP. This reduces the standby power and also gives the PC uncontrolled access to the shared memory. In this default

mode, the card appears to the PC and is used by the PC as a standard memory card only. The host can download various communications signal processing (CSP) algorithms to the card without enabling the DSP. The host can also program the flash memory with a DSP initialization code or even a real-time DSP operating system before enabling the DSP. The DSP does not become active until it is specifically made active by the host PC.

Smart Mode

In the smart mode, the communications registers become active and available to the host and DSP. The host continues to have full access to the entire memory on the card. However, when the host PC accesses the shared memory, the DSP operation is temporarily halted because the arbitration logic must put the DSP in a hold condition to give the PC access to the memory bus. Control and communication between the DSP and the PC are implemented via the host communication registers. Although the host PC accesses these registers as regular shared memory, they are physically located in the FPGA. This allows the PC and DSP to access these registers without halting the DSP operation.

Switching Between Standard Mode and Smart Mode

When the PC writes the DSP signature pattern (A320), the DSP is activated and the card is switched from standard mode to smart mode. Once a valid signature is detected, the corresponding bit is set in the DSP control register, DSPCR. Resetting this bit automatically deactivates the DSP and switches the card to standard mode. An alternate method of switching modes is writing to a user-defined register in the PCMCIA attribute memory space.

Memory Organization

The PCMCIA DSP cards provide two separate memory spaces for the common memory and attribute memory. Both memory spaces are accessible by the DSP and the PC. The DSP accesses the common memory in its program and data space and the attribute memory in its global data memory space.

The ability to switch efficiently between various DSP tasks without having to reinitialize or reload is critical for any multifunction communications system. Such a system needs a common memory area that DSP operating systems and the host applications can always access to save system parameters and the operating system itself. Page 0 of the DSP data and program memory is defined to be always active. Thus, DSP operating systems can use page 0 as system memory and additional pages as application-specific memory.

Bus Arbitration

Both the DSP and the PC can access the shared memory on the card. The PC always has higher priority for accessing the memory bus on the card. During PC accesses to the memory bus, the DSP operation is halted. The arbitration logic in the FPGA asserts the $\overline{\text{HOLD}}$ signal to the DSP and extends the PC memory bus access cycle by asserting the $\overline{\text{WAIT}}$ signal. Once the DSP acknowledges the hold by asserting $\overline{\text{HOLDA}}$, the PC $\overline{\text{WAIT}}$ is released and access to shared memory is completed. As soon as the PC completes its access, control of the shared memory is returned to the DSP. Since communication, control, and control registers are not resident in the shared memory, any PC access to these registers will not halt the DSP operation.

Memory Access by the PC

When the PC accesses the shared memory, the DSP is put on hold to grant control of the bus to the PC. The PC's memory access is extended by using the $\overline{\text{WAIT}}$ signal until the DSP puts its bus in the high-impedance state, as indicated by $\overline{\text{HOLDA}}$ signal. There is a time-out if $\overline{\text{HOLDA}}$ is not granted in time. When the card is in smart mode, the PC cannot access the first 16 bytes of the shared memory (also note that the PC cannot access DSP internal memory). This could be used as protected memory for the DSP. PC accesses to this

block do not cause the DSP to be put on hold. The PC must load the DSP reset and interrupt vectors and the application algorithm before switching the card into smart mode. Since the PC can access the entire memory on the card without consideration of the DSP page sizes, memory pages not used by the DSP can be dedicated exclusively for the PC.

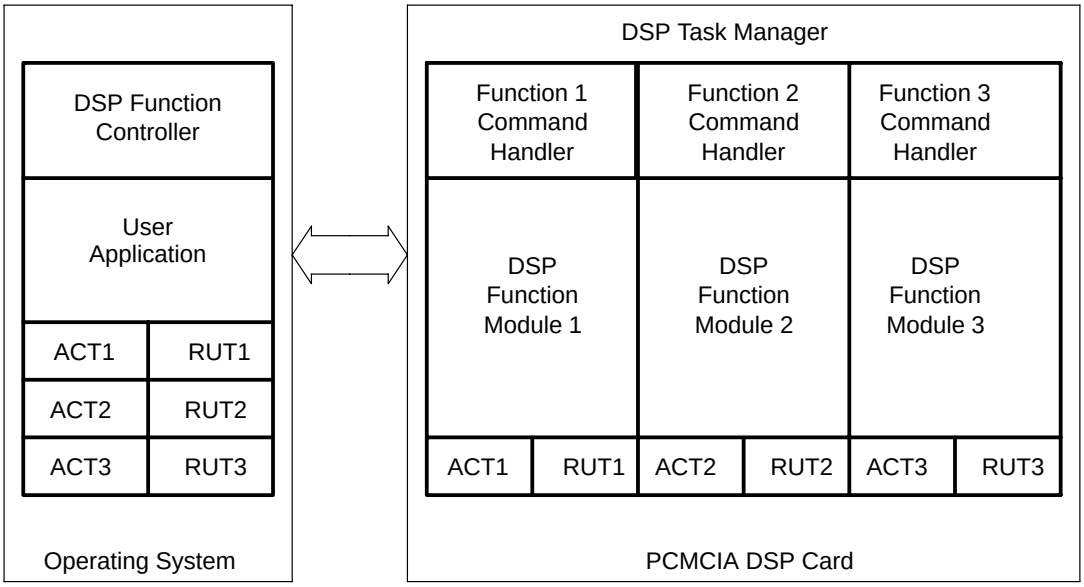
Memory Access by the DSP

The ’C5x versions of the DSP cards can address a maximum of 3M words of common memory. The DSP address range is expanded by using the page selects. Page size for the ’C5x DSP is 32K ($\times$ 16) words. Page 0 (both program and data memory for the ’C5x) is always enabled and cannot be deselected via page select bits in the SYSCFG register. This allows DSP operating systems to use this memory without affecting any memory dedicated for DSP applications.

Loading and Executing a Single Algorithm

Initially, the PC loads the desired algorithm to the DSP memory and initializes the DSP. Then the PC enables itself to be interrupted by setting the appropriate enable bits in the DSP control register (DSPCR). This interrupt can be generated by the AFE card (voice activated switch, ring detect, etc.), depending on the application.

Figure 3. Loading and Executing a Single Algorithm



Once the algorithm is loaded and the system is initialized, the host PC can reduce power consumption by turning off the DSP clock, which automatically puts the DSP in a hold condition, placing its buses into the high-impedance state and allowing the PC quicker access to the remaining unused memory on the card.

When the desired external event occurs (indicated by the interrupt), the PC turns the DSP clock on, and the DSP starts executing the algorithm. Since the algorithm is already loaded into DSP memory, there is no delay in loading the algorithm; this makes fast system response time possible.

Note that the code may also be written into global data memory, and the DSP may be bootloaded by the PC to force the DSP to run any preselected default application.

Loading and Executing Multiple Algorithms

First, the host PC initializes the DSP card and loads the DSP operating system. Now the operating system can load multiple DSP algorithms into the DSP's local memory by using the paging scheme. Each 32K-word page could be used for a specific application. Algorithms that require more than 32K words of memory can use multiple pages. Since the paging scheme is needed only for DSPs with a 16-bit address reach, the host PC or other 32-bit DSP, such as the TMS320C3x, can ignore the paging scheme. Also note that a real-time DSP operating system, such as SPOX 2.0, can be loaded into the DSP's on-chip RAM or mask-programmable ROM, freeing the entire external memory for an applications program or data.

The PC and the DSP must follow a predetermined handshake protocol. Commands and data can be passed easily by using the communication registers without halting DSP operation. The DSP operating system controls enabling of DSP program/data pages and transmission on processed data to the PC.

Host Communication

The host PC and the DSP communicate to each other via dedicated host communications registers. These registers are dual-mapped into the common memory space and I/O space of the of the host PC. They are always mapped into the I/O space for the DSP.

The appropriate control and status registers can be programmed to allow an interrupt-based handshake between the host and the DSP. Both the DSP and the host PC can also poll the appropriate bits in the status registers, where interrupts are not available. This could be true in some motherboard applications, where a single integrated device may share the local memory with the host CPU.

Conclusion

With a real-time DSP operating system such as SPOX 2.0, which is small enough to be executed from a DSP's on-chip ROM, and application algorithm modules loaded into the shared memory as needed by the host PC, the PCMCIA DSP card could become the universal communications system for the emerging mobile office environment. With the TEC320 cDSP available today, the same set of CSP software modules could run on a PDA motherboard or PC add-on card if the required analog interface is provided. With such a universal communications platform and standardized user applications interface, the hardware dependencies and porting nightmares should be a thing of the past.

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.