Technical notes on using Analog Devices DSPs, processors and development tools
Contact our technical support at processor.support@analog.com and dsptools.support@analog.com
Or visit our on-line resources http://www.analog.com/ee-notes and http://www.analog.com/processors

ADSP-BF537 Blackfin® Highlights for ADSP-BF533 Users

Contributed by Glen Ouellette and Benno Kusstatscher

Rev 2 – May 10, 2005

Introduction

The Blackfin® family of processors from Analog Devices have recently been enhanced by three new members, the ADSP-BF534, the ADSP-BF536, and the ADSP-BF537 derivatives. These new parts, which are code compatible with the ADSP-BF531 / ADSP-BF532 / ADSP-BF533 devices, provide a different set of peripherals on the identical platform, including:

- CAN 2.0B controller
- I^2C-compatible TWI interface
- One additional UART controller
- Five more timers
- 32 additional GPIOs
- Handshaking Memory DMA capability

In addition, ADSP-BF536 and ADSP-BF537 derivatives also feature a 10/100 Mbps Ethernet MAC.

Besides all these new peripherals, several architectural enhancements have been made to further improve system performance. The intention of this application note is not to explain the new peripherals, but rather highlight the improvements that have been included in the new ADSP-BF537 devices.

The target audience for this document is expected to be familiar with ADSP-BF533 processors.

Where ever this document refers to ADSP-BF533 processors, it implicitly relates to the memory derivatives, ADSP-BF531 and ADSP-BF532, as well.

Similarly, the name "ADSP-BF537" as used in this document represents all of the ADSP-BF536 memory derivatives and pin-compatible ADSP-BF534 parts.

General-Purpose Ports

The General-purpose Ports is probably the most significant change from the ADSP-BF533. Due to the number of on-chip peripherals featured by ADSP-BF537 devices, there was a need to multiplex pin functions.

The peripherals are organized into General-purpose Ports designated as Port F, Port G, Port H, and Port J. In default mode, all pins of Port F, Port G, and Port H are in general-purpose I/O (GPIO) mode. Port J does not provide GPIO functionality.

To enable Peripheral functionality, the Function Enable registers (`PORTF_FER`, `PORTG_FER`, and `PORTH_FER`) must be explicitly written. Certain pins on Port F, Port G, and Port H, are multiplexed with more than one peripheral. In these cases, the required peripheral is controlled by the Multiplexer Control register (`PORT_MUX`).

Copyright 2005, Analog Devices, Inc. All rights reserved. Analog Devices assumes no responsibility for customer product design or the use or application of customers' products or for any infringements of patents or rights of others which may result from Analog Devices assistance. All trademarks and logos are property of their respective holders. Information furnished by Analog Devices applications and development tools engineers is believed to be accurate and reliable, however no responsibility is assumed by Analog Devices regarding technical accuracy and topicality of the content provided in Analog Devices Engineer-to-Engineer Notes.

When porting code from ADSP-BF533 devices to ADSP-BF537 devices, ensure that startup code manages the function enable and port muxing registers accordingly, prior to initializing on-chip peripherals.

Flag Pins vs. GPIOs

The ADSP-BF533 Flag Pins have not been physically changed on ADSP-BF537 processors. However, the nomenclature changed. *Flag Pins* are now called *GPIOs*. The register names have also been changed.

While ADSP-BF533 processors have one module featuring 16 Flag Pins, ADSP-BF537 processors feature three such modules, providing 48 GPIOs in total. Each GPIO can also operate as an interrupt input. GPIOs can control up to five interrupt channels.

Most GPIOs are multiplexed with other functions. After reset, the respective pins are configured to operate as GPIOs. By default, both the transmit and receive drivers are disabled. The `PORTxIO_DIR`, `PORTxIO_INEN`, and `PORTx_FER` register reset to zero.

Additionally, unlike the ADSP-BF533, the ADSP-BF537 has eight high-current source/sink pins to ease system cost and reduce part count. These pins are found on Port F, `PF7 - PF0`, and function as such regardless whether in GPIO or peripheral function mode.

System Interrupts

Although the Core Event Controller (CEC) remains the same, the System Interrupt Controller (SIC) has minor enhancements.

The ADSP-BF533 processor features 23 system interrupt sources. The ADSP-BF537 features 47 of them! As a result, all 32 bits of the `SIC_IWR`, `SIC_ISR`, and `SIC_IMASK` registers are now populated. In addition, a fourth interrupt assignment register (`SIC_IAR3`) was added, enabling you to fully control 32 system interrupt inputs.

Some of the interrupt sources are hard-wired together. All DMA error and status interrupts share one interrupt channel, requiring that you to install one global DMA error handler for all DMA channels. Similarly, all peripheral error interrupt sources are ORed together. See Figure 4-1 of the *ADSP-BF537 Blackfin Processor Hardware Reference* [1] for details. Note that the manual dedicates a chapter to system interrupts.

Finally, the polarity of the `NMI` input has changed. Refer to NMI Polarity for details.

DMA Enhancements

Compared to ADSP-BF533 devices, the new ADSP-BF537 parts have four more DMA channels to support the numerous additional peripherals. The new channels are identical to others. All twelve peripheral DMA channels can be assigned to any of the connected peripherals. Consequently, the `PMAP` field in the `DMAx_PERIPHERAL_MAP` registers takes any value between 0 and 11, now.

The introduction of the four new DMA channels required a shift in the Memory DMA channels' register addresses. If used in a symbolic manner, this has no impact to user code as the ADSP-BF537 header files will resolve the addresses.

SYNC Bit

The operation of Bit 5 of the DMA configuration registers was enhanced on the ADSP-BF537. On ADSP-BF533 processors, this bit (called the `RESTART` bit) controls receive (memory write) operation of the DMA channels only.

On ADSP-BF537 processors, this bit is called the `SYNC` bit. Though its meaning has almost remained the same for receive operation, it now plays an important role in transmit (memory read) DMA modes of operation.

When the SYNC bit is cleared on transmit DMAs, the ADSP-BF537 behaves the same as the ADSP-BF533. That is, the DMA engine continues immediately when the last data of a DMA work unit is transferred from memory into the DMA FIFO.

This makes perfect sense when the current DMA descriptor is followed by another DMA descriptor. The new DMA descriptor is already fetched while former descriptors final data is still pending in the DMA FIFO, guaranteeing the highest throughput.

If, however, a DMA is terminating in stop mode (FLOW = 0) with interrupts enabled, the interrupt is issued immediately after the final data have been moved from the memory into the DMA FIFO. By the time the interrupt service routine is invoked, the final data are most likely still pending in the 4-stage DMA FIFO. Therefore, the service routine is not allowed to start a new DMA sequence yet as old data still reside in the DMA FIFO. The DMA_RUN bit in the DMAx_IRQ_STATUS registers indicates whether data still pending in the FIFO. In many cases the service routine is required to poll this bit until it goes low.

On the ADSP-BF537, the new SYNC bit enables you to control the interrupt timing on transmit DMAs: when set, it delays the interrupt until the content of the DMA FIFO is drained to the transmitting peripheral. In other words, the SYNC bit controls whether the interrupt aligns to memory side or to the peripheral side of the DMA FIFO.

In the case of a transmit DMA running in stop mode (FLOW = 0), the interrupt does not issue until the DMA_RUN bit goes low. With the SYNC bit set, the service routine can safely start a new DMA sequence or reset the DMA configuration. It is, however, still too early to disable the connected peripheral, as this may have its own transmit buffers that are still processing the data.

Digging deeper, the SYNC bit controls more than just the interrupt timing, it impacts also the DMA state machine. If the SYNC bit is cleared, neither receive nor transmit DMAs are allowed to change certain settings, such as the transfer word size between work units. Setting the SYNC bit eliminates the restrictions for transmit channels.

For receive DMAs, the same recommendation is valid as for ADSP-BF533 DMAs: do not set the SYNC bit except in the first work unit of a descriptor chain.

Descriptor Chains to Varying Memory Spaces

Similar to ADSP-BF533 devices, work units of a receive DMA descriptor chain are permitted to write to different memory spaces (i.e., one work unit writes to internal L1 memory and the subsequent one writes to any off-chip memory), regardless of the SYNC setting. Transmit DMAs, however, are traditionally not permitted to write to different memory spaces. Although the memory space is specified by the DMA's start address only, this operation is not permitted on ADSP-BF533 processors.

On ADSP-BF537 DMAs, the SYNC bit also enables a transmit work unit to read data from a different memory space than the previous work unit; if the previous one had the SYNC bit set.

It is worth more than just a side note, that with the help of the SYNC bit, Memory DMAs that transfer data from on-chip L1 memory to external memories (or vice versa) can now change the transfer direction without stopping an on-going descriptor chain.

The memory write channel of a Memory DMA behaves like a receive DMA and does not require the SYNC bit to be set. The memory read channel, however, behaves like a transmit DMA. Set the SYNC bit in descriptors there, if the start address of subsequent descriptor points to a different memory space.

Peripheral-Mastered DMA Channels

On ADSP-BF533 processors, there is a single control strobe from a peripheral to the DMA controller: DATA REQUEST. With the addition of the Ethernet MAC, the ADSP-BF537 DMA controller has been enhanced to provide the MAC further control over the assigned DMA channels. The Ethernet MAC can emit four different commands to the DMA channel:

- DATA REQUEST
- DATA REQUEST URGENT
- RESTART WORK UNIT
- FINISH WORK UNIT

For example, with the DATA REQUEST URGENT command, the peripheral can signal to the DMA channel that a certain high-water level in its (receive) buffer has been reached and further data reception would cause data loss, if the DMA does not serve the peripheral in time.

With the RESTART and FINISH commands, the peripheral can master the associated DMA channel. Consider the case of a processor DMAing in a data stream, and the peripheral detects an incorrect checksum condition once done. The peripheral can skip the data stream by issuing a RESTART command to the DMA channel. The DMA simply reloads its current registers again. Upon further DATA REQUEST commands, it overwrites the recently received data again. Although the peripheral had used the processor's memory for buffering, the processor's core is not impacted by the failing data stream.

The FINISH command helps the peripheral handle data streams with variable length. Typically, the DMA count register is preloaded with a maximum value. The peripheral starts receiving data until it detects that the data unit is complete. Then it issues a FINISH command. The core is interrupted optionally, and the DMA channel starts a new work unit for the new data set.

Handshaking Memory DMA

Memory-to-memory DMAs usually run at maximal speed as permitted by system throughput. The ADSP-BF537 processors' Handshaking Memory DMA capability enables the two Memory DMA units to be synchronized by external hardware.

There are two edge-sensitive DMA request inputs: DMAR0 and DMAR1. Each is associated with one of the Memory DMA units: MDMA0 and MDMA1.

Although DMA request inputs can be used to control the timing of normal memory-to-memory DMAs, more than likely, they control transfers, such as asynchronous FIFOs or off-chip interface controllers, between Blackfin memory and hardware buffers.

When handshake operation is enabled, the Memory DMA no longer runs freely. Rather, the DATA REQUEST commands are gated by the HMDMA unit. Options, such as whether to initialize every individual DMA transfer by a DMAR0 or DMAR1 edge, or whether such an event triggers the transfer of an entire block of data, are user- programmable.

The new handshake operation mode not only reduces the need of core interaction, it also increases data throughput, since GPIO polling or GPIO-driven interrupts can be avoided in many cases.

External Bus Interface Unit

Three minor, but important, enhancements to the 16-bit wide parallel interface to external SDRAM and asynchronous memories have been made:

- The ADSP-BF537 SDRAM controller now supports SDRAM devices up to 512 MB.
- The ARDY input does not have to be synchronous with CLKOUT. This has changed also on ADSP-BF533 revision 0.4.

- The SDRAM can still be refreshed in hibernate mode. See Hibernate Mode and SDRAM Refresh for details.

Besides this modification to the EBIU module itself, the new handshake operation of the Memory DMA improves the value of the EBIU interface when talking to off-chip buffer and FIFOs. See Handshaking Memory DMA for details.

PPI Enhancements

In general, the PPI implementation on the ADSP-BF533 was carried over to the ADSP-BF537 with the exception of a few minor changes and additions, as described below.

With the introduction of the General-purpose Ports on the ADSP-BF537, you must ensure that the PPI clock, data, and control signals are correctly routed to the appropriate pins. For the PPI data signals, this requires setting the appropriate bits of `PORTG_FER` register and clearing respective bits of the `PORT_MUX` register. For the PPI clock and frame sync signals, this requires setting bits in `PORTF_FER` register and clearing respective bits of the `PORT_MUX` register.

With regard to frame syncs on an existing ADSP-BF533 design, replace each reference to *Timer 1* with *Timer 0* and replace each reference to *Timer 2* with *Timer 1*. This will account for the shift in functionality on the TMR pins (`TMR1 > TMR0`, `TMR2 > TMR1`). Note that this also affects the Timer MMRs (`TIMER1_WIDTH > TIMER0_WIDTH`, etc.) accesses in legacy code ported from the ADSP-BF533 to the ADSP-BF537.

New to ADSP-BF537 processors is the addition of Line tracking error tracking mechanism. This is addressed with the addition of two new status bits (`LT_ERR_OVR`, `LT_ERR_UNDR`) in the `PPI_STATUS` register. When set, the `LT_ERR_OVR` and `LT_ERR_UNDR` bits, which are sticky, indicate that a Line Track Error has occurred. These bits are valid for RX modes with recurring frame syncs only. If one of these bits is set, the programmed number of samples in `PPI_COUNT` did not match the actual number of samples counted between assertions of `PPI_FS1` (for General Purpose modes) or "Start of Active Video (SAV)" codes (for ITU-R 656 modes). If the PPI Error interrupt is enabled in the `SIC_IMASK` register, an interrupt request will be generated when one of these bits is set.

The `LT_ERR_OVR` flag signifies that a horizontal tracking overflow has occurred, where the value in `PPI_COUNT` was reached before a new SAV code was received. This flag does not apply to non-ITU-R 656 modes; in this case, once the value in `PPI_COUNT` is reached, the PPI simply stops counting until receiving the next `PPI_FS1` frame sync.

The `LT_ERR_UNDR` flag signifies that a horizontal tracking underflow has occurred, where a new SAV code or `PPI_FS1` assertion occurred before the value in `PPI_COUNT` was reached.

UART Controllers

ADSP-BF537 parts have two UART modules. Both UART modules are identical to the one UART featured in the ADSP-BF533; however, some signals are now multiplexed with other peripherals. Both UARTs have IrDA support. Timer 1 can perform autobaud detection on UART0, and Timer 6 can do the same for UART1.

UART register names on ADSP-BF533 devices begin with "`UART_`". On ADSP-BF537 devices, UART register names begin with "`UART0_`" or "`UART1_`".

All UART signals, which are found on Port F, are multiplexed with other signals. See General-Purpose Ports for details.

On 0.1 silicon, all known UART anomalies have been fixed. Five are still present on the current 0.4 revision of ADSP-BF533 processors.

General-Purpose Timers

ADSP-BF537 parts have eight PWM timers (five more than on ADSP-BF533 processors). The timers themselves have not changed; however, some signals are now multiplexed with other peripherals. Five more bits are now populated in the TIMER_ENABLE and TIMER_DISABLE registers. All eight timers can still be started and stopped synchronously. Therefore, the 32-bit-wide TIMER_STATUS register is read in atomic manner to report a consistent status snapshot across all eight timers.

On the ADSP-BF533 parts, all three timers can be clocked alternatively by the common PF2 pin. All can capture the UART RX input.

On ADSP-BF537 parts, individual timers use different pins for alternate clocking. Additionally, the TMRCLK input is common to all eight timers.

Timers can capture the UART0 RX, the UART1 RX, and the CAN RX pins for autobaud detection.

Boot ROM

ADSP-BF533 processors have two BMODE pins. ADSP-BF537 processors have three pins, enabling additional boot modes as shown in Table 1.

Due to a silicon anomaly, BMODE=0 and BMODE=1 only are functional on silicon revision 0.0. On 0.1 revision, the content of the Boot ROM has experienced major changes. The following discussion does not apply to 0.0 silicon.

ADSP-BF537 processors have 2 Kbytes of on-chip Boot ROM – twice the amount available on ADSP-BF533 parts. The Boot ROM not only provides new boot modes, but also provides enhanced features.

BMODE	Description
000	**No-boot** Unchanged. The program counter resets to address 0x2000 0000 and starts code execution from there, assuming a 16-bit non-volatile memory is connected to the /AMS0 chip select strobe.
001	**Boot from parallel flash/EPROM** Enhanced. The Boot Kernel starts fetching boot blocks from address 0x2000 0000 assuming an 8-bit or a 16-bit non-volatile memory has been connected to the /AMS0 chip select strobe. True 16-bit DMA mode is now supported.
010	**Boot from 16-bit FIFO (slave mode)** New boot mode. Similar to BMODE=1, except that an asynchronous FIFO device is to be connected to the /AMS3 strobe. Every 16-bit transfer is requested by a rising edge on the DMA request input DMAR1. Note that BMODE=2 has been SPI slave mode on ADSP-BF533 devices.
011	**Boot from SPI memory** Enhanced. Boot from 8- / 16- / 24-bit addressable memories or Atmel DataFlash® devices. SPI SSEL1 (PF10) is used as chip select.
100	**Boot from SPI host (slave mode)** Equivalent to BMODE=2 mode on ADSP-BF533 silicon revision 0.3 and higher.
101	**Boot from TWI memory** New boot mode. Read boot stream from I2C-compatibly memory devices.
110	**Boot from TWI host (slave mode)** New boot mode. Consume boot stream from I2C-compatible master device.
111	**Boot from UART host (slave mode)** New boot mode. After receiving an initial 0x40 byte ('@') the Boot Kernel performs a bit rate auto-detection and sends four bytes back: 0xBF, followed by DLL and DLH and a trailing 0x00 byte. Then, the kernel is ready to receive the complete boot stream (8 data bits, no parity)

Table 1. ADSP-BF537 Boot Modes

The ADSP-BF537 Boot Kernel processes at reset interrupt priority, the one of the ADSP-BF533 processors degrades itself down to IVG15 priority. As a result, the ADSP-BF537 kernel is robust against NMI events at boot time without requiring a special NMI handler. The user program is requested to install proper handlers before leaving reset state.

Host Wait Strobe

On ADSP-BF533 processors, the Host Wait strobe (HWAIT) is available by the SPI slave boot mode. On ADSP-BF537 devices, HWAIT is available in all boot modes except when BMODE=0.

Since all 48 GPIO pins can be used for this purpose, the 4-bit PFLAG field in the boot blocks' headers is now accompanied by a 2-bit field, called PPORT, whose meaning is defined by the following truth table:

PPORT=00: HWAIT is disabled (default)

PPORT=01: HWAIT operates at Port F

PPORT=10: HWAIT operates at Port G

PPORT=11: HWAIT operates at Port H

The PFLAG setting specifies which GPIO pin of the specified port is used.

Although HWAIT has fixed polarity on ADSP-BF533 processors (high=wait, low=continue), polarity is programmable on ADSP-BF537 parts. If the chosen HWAIT pin is pulled down by a resistor, the HWAIT polarity is the same as on ADSP-BF533 processors. A pull-up resistor, however, inverts the signal polarity.

Furthermore, the timing of HWAIT has been changed. Although ADSP-BF533 processors activate HWAIT only when processing IGNORE or ZEROFILL blocks, ADSP-BF537 parts activate HWAIT by default and release it only when ready to consume data (i.e., as soon as the respective DMA has been enabled). This conservative timing enables host devices to operate at higher bit rate.

32-Bit Block Count

The byte count field of the Blackfin boot stream block headers has always been 32 bits wide. The ADSP-BF533 Boot Kernel, however, can process 16-bit counts only. IGNORE blocks in the flash boot mode is the only block type that can handle true 32-bit byte counts.

The ADSP-BF537 Boot Kernel supports 32-bit counts for all block types in all boot modes. Zero count is valid always.

16-Bit DMA

If BMODE=1, the ADSP-BF533 processors support 8- and 16-bit Flash devices. In either case, 8-bit DMA is performed.

On ADSP-BF537 devices, users select whether 8- or 16-bit DMA is used for flash booting. The first byte of the boot stream instructs the Boot Kernel to operate in one of the following modes:

0x20: 16-bit device, 16-bit DMA

0x40: 8-bit device, 8-bit DMA

0x60: 16-bit device, 8-bit DMA

The 0x20 mode is not present on ADSP-BF533 devices yet. Typically, it shortens the time required for booting. It would fail, however, if the block count or the target address is an odd value. The VisualDSP++® elfloader utility guarantees that this never occurs.

For FIFO booting (BMODE=2) only, the 0x20 mode is functional. TWI, SPI, and UART modes support the 0x40 mode only.

Combination of Flag Bits

The ADSP-BF537 Boot Kernel enables multiple Flag bits to be set for the same block. For example, a block can be an IGNORE block, an INIT block, and a FINAL block at the same time.

When the ZEROFILL flag is set, the IGNORE and INIT bits are not evaluated. These flag combinations are reserved for future use.

Supported SPI Memory Devices

Similar to ADSP-BF533 processors, ADSP-BF537 processors support SPI memory devices with 8-, 16-, and 24-bit addressing in SPI master mode (BMODE=3). The ADSP-BF537 Boot Kernel does, however, support more Atmel DataFlash derivatives, now. Furthermore, the ADSP-BF537 not only supports the devices shown in the table below, but also supports all devices with the same *density code* as the listed devices, granted the *Bit Level Addressing Sequence*, as shown in the respective data sheet, is also the same.

Atmel DataFlash	Size [Bytes]	ADSP-BF533
AT45DB041B	4 M	✓
AT45DB081B	8 M	✓
AT45DB161B	16 M	✓
AT45DB321C	32 M	
AT45DB642	64 M	
AT45DB1282D	128 M	

Table 2. DataFlash Supported by ADSP-BF537

Run-Time Functions

Although the Boot ROM of the ADSP-BF533 devices is used only for booting after reset, the Boot ROM of ADSP-BF537 processors contains functions that can be called at runtime from user programs as well.

The primary purpose of these functions is to manage multiple boot streams in the boot memory device. In slave configurations, the host device may pull the Blackfin processor's /RESET pin and supply different boot data, enabling the Blackfin to boot a different application (DXE).

Further support is required only for the three master modes when booting from Flash, SPI, or TWI memories.

Two user-callable functions are provided for each master boot mode. One function simply boots a new application. It has one parameter (the start address of the boot stream that needs to be booted, which resides in off-chip memory). For SPI boot mode, a second parameter informs the function as to the GPIO pin of Port F that is to serve as the slave select strobe. This way, individual boot streams may reside in different SPI memory devices. Also, the bit rate can be controlled by the user program.

For TWI boot mode, a different parameter informs the function as to the address select of the targeted TWI memory.

A second set of functions helps to determine the start address of the individual boot streams. The input parameter simply passes the sequential number of the DXE to be booted, and the function returns the boot stream's start address. For SPI/TWI boot modes, a second parameter is passed, which selects the targeted memory device, as explained above.

These run-time functions help to implement boot management or second-stage loader scenarios without forcing the user to understand the secrets of booting in detail. Fractions of the Boot Kernel are reused for user purposes, avoiding the need to reserve RAM for customized second-stage loaders.

System Management

NMI Polarity

The polarity of the Non-Maskable Interrupt (/NMI) pin on the ADSP-BF537 is active low. This is inverted from that of the ADSP-BF533. The /NMI pin is used frequently in host applications as a power-down indicator to initiate an orderly shutdown.

PLL

ADSP-BF537 processors, similar to ADSP-BF533 processors, can be clocked by an external crystal, a sine wave input, or a buffered, shaped clock derived from an external clock oscillator.

On ADSP-BF537 devices, however, the maximum allowable VCO rate (CLKIN) has been raised from that of the ADSP-BF533 to a maximum of 50 MHz.

Buffered Input Clock

ADSP-BF537 processors provide a new output pin (CLKBUF), which optionally provides a buffered version the input clock signal (CLKIN). This pin allows another device, most likely an Ethernet PHY, and the Blackfin processor to run from a single crystal oscillator. For example, in this application, a single 25 MHz or 50 MHz crystal may be applied directly to the ADSP-BF537. The 25 MHz or 50 MHz output of CLKBUF may then be connected to an external Ethernet MII or RMII PHY device.

The functionality of the CLKBUF pin is controlled by the CLKIN Buffer Output Enable (CLKBUFOE) control bit, which is found in the Voltage Regulator Control register (VR_CTL).

Switching Voltage Regulator

The on-chip switching regulator that generates the internal supply voltage (VDDINT) is identical to the regulator used on ADSP-BF533 silicon revision 0.4.

Besides the on-chip power-on reset generator, the regular /RESET input pin resets the VR_CTL register to its default value. Brown-out conditions are expected to be detected by an external power-fail circuit. Pull /RESET down as soon as VDDEXT goes below 2.25 V.

Hibernate Mode and SDRAM Refresh

The internal voltage supply regulator of the ADSP-BF537 can be shut off by writing b#00 to the FREQ bits of the VR_CTL register. This disables both CCLK and SCLK. Furthermore, it sets the internal power supply voltage (VDDINT) to 0 V, eliminating leakage currents. On ADSP-BF533 devices the internal voltage supply regulator can be awakened by either:

- Assertion of the RESET pin
- RTC event

Additionally, ADSP-BF537 devices can wake up on the following events:

- Activity on the CANRX pin
- Ethernet PHY event
- External GPIO event on PH6 pin

When the core is powered down, VDDINT is set to 0 V, and thus the internal state of the processor is not maintained, with the exception of the VR_CTL and RTC registers. Therefore, write critical information stored internally (memory contents, register contents, and so on) to a non-volatile storage device prior to removing power.

Optionally, a new control bit (CKELOW) has been added to the VR_CTL register to control the functionality of the CKE pin on the SDRAM. By setting the CKELOW bit, the CKE pin remains driven during reset and maintains the self-refreshing state of the SDRAM. This ensures that the contents of the SDRAM are maintained during a reset.

Common Modules

For completeness, the modules that have not been changed between ADSP-BF533 and ADSP-BF537 product generations are listed below. Of course, some modules might have experienced maintenance in terms of bug fixes. As a result, anomalies listed on the ADSP-BF533 anomaly sheets may not exist on ADSP-BF537 processors.

Core

ADSP-BF537 derivatives feature the same core as the ADSP-BF533 parts. Therefore, the parts are fully code compatible.

Memory Architecture

Although the population of L1 memory blocks on the new parts varies from derivative to derivative, the L1 memory/cache architecture is identical to that of the ADSP-BF533 processors.

SPORT Controllers

The two SPORT controllers have not changed; however, some signals are now multiplexed with other peripherals. The SPORT1 signals are now multiplexed with PPI and GPIO pins. Also, the secondary data signals of SPORT0 are multiplexed with the CAN signals. See General-Purpose Ports for details.

SPI Controller

The SPI controller has not changed; however, some signals are now multiplexed with other peripherals. The primary SPI signals are multiplexed with GPIO pins, and seven slave select signals are spread out over Port F and Port J. See General-Purpose Ports for details.

Real-Time Clock

The RTC has not changed, and its signals are not multiplexed.

Core Timer

The Core Timer has not changed.

Watchdog Timer

The Watchdog Timer has not changed.

Conclusion

Besides the new on-chip peripherals, the new Blackfin derivatives have experienced a set of improvements that might stand out from the feature list on first page of the data sheet.

Nevertheless, core and system architecture are identical to ADSP-BF533 devices. By default, all the enhanced peripherals still behave in the known manner. If you are already familiar with the ADSP-BF533 parts, you can transfer that same knowledge to the new ADSP-BF537 derivatives. This application note identifies the areas of the Hardware Reference manual and data sheet that you may need to focus on.

Similarly, code porting is an easy task. The only challenge is the new port muxing scheme and the changed register names in the GPIO modules. A header file accompanies this EE-Note to help you from having to change some register names in your source code when switching between devices.

References

[1] *ADSP-BF537 Blackfin Processor Hardware Reference.* Preliminary Rev 1.1, January 2005. Analog Devices Inc.

[2] *ADSP-BF533 Blackfin Processor Hardware Reference.* Rev 3, September 2004. Analog Devices Inc.

[3] *ADSP-BF536/ADSP-BF537 Blackfin Embedded Processor Data Sheet.* Rev PrD, January 2005. Analog Devices Inc.

[4] *ADSP-BF534 Blackfin Embedded Processor Data Sheet.* Rev PrD, January 2005. Analog Devices Inc.

[5] *ADSP-BF531/ADSP-BF532/ADSP-BF533 Blackfin Embedded Processor Data Sheet.* Rev 0, March 2005. Analog Devices Inc.

[6] *ADSP-BF537 Blackfin Processor Anomaly List.* Rev B, March 2005. Analog Devices Inc.

[7] *ADSP-BF534 Blackfin Processor Anomaly List.* Rev B, March 2005. Analog Devices Inc.

[8] *ADSP-BF533 Blackfin Processor Anomaly List.* Rev M, March 2005. Analog Devices Inc.

[9] *ADSP-BF533 Blackfin Booting Process (EE-240).* Rev 3, January 2005. Analog Devices Inc.

Document History

Revision	Description
Rev 2 – May 10, 2005 *by Benno Kusstatscher* *and Jorge Manguane*	Updated Boot ROM section to reflect new features of 0.1 silicon revision
Rev 1 – January 24, 2005 *by Glen Ouellette* *and Benno Kusstatscher*	Initial release