TMS320C55x™ DSP Functional Overview

Literature Number: SPRU312
June 2000

PRINTED WITH SOY INK™

Texas Instruments

Printed on Recycled Paper

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products 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.

Customers are responsible for their applications using TI components.

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

TI assumes no liability for applications assistance or customer product design. TI does not 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. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Preface

Read This First

About This Manual

This document provides a functional overview of the devices included in the TMS320C55x™ generation of digital signal processors (DSPs). Included are descriptions of the CPU architecture, bus structure, memory structure, and on-chip peripherals. Detailed descriptions of device-specific characteristics such as package pinouts, package mechanical data, and device electrical characteristics are included in separate device-specific data sheets.

Trademarks

TMS320C55x is a trademark of Texas Instruments Incorporated.

C55x is a trademark of Texas Instruments Incorporated.

MicroStar BGA is a trademark of Texas Instruments Incorporated.

XDS and XDS510 are trademarks of Texas Instruments Incorporated.

TMS320 is a trademark of Texas Instruments Incorporated.

C6000 is a trademark of Texas Instruments Incorporated.

C54x is a trademark of Texas Instruments Incorporated.

Code Composer Studio is a trademark of Texas Instruments Incorporated.

TMS320C5000 is a trademark of Texas Instruments Incorporated.

Contents

Figures

Tables

Chapter 1

Features

This chapter lists characteristics of the CPU, on-chip memory, on-chip peripherals, and power conservation features of the TMS320C55x™ DSP. Emulation, test, and packaging information is also shown.

Topic **Page**

1.1 TMS320C55x Processor Characteristics

Table 1–1 lists characteristics for members of the TMS320C55x generation of fixed-point digital signal processors (DSPs). All references to C55x™ DSP hereafter indicate TMS320C55x unless otherwise specified.

1.1.1 CPU

Features for the high performance, low-power C55x CPU include:

❑ Advanced multiple-bus architecture with one internal program memory bus and five internal data buses (three dedicated to reads and two dedicated to writes)

❑ Unified program/data memory architecture

❑ Dual 17-bit x17-bit multipliers coupled to 40-bit dedicated adders for non-pipelined single-cycle multiply accumulate (MAC) operations

❑ Compare, select and store unit (CSSU) for the add/compare section of the Viterbi operator

❑ Exponent encoder to compute an exponent value of a 40-bit accumulator value in a single cycle

❑ Two address generators with eight auxiliary registers and two auxiliary register arithmetic units

❑ Data buses with bus holders

❑ 8M x 16-bit (16M-bytes) total addressable memory space

❑ Single-instruction repeat or block repeat operations for program code

❑ Conditional execution

❑ Seven-stage pipeline for high instruction throughput

❑ Instruction Buffer Unit that loads, parses, queues and decodes instructions to decouple the program fetch function from the pipeline

❑ Program Flow Unit that coordinates program actions among multiple parallel CPU functional units

❑ Address Data Flow Unit that provides data address generation and includes a 16-bit arithmetic unit capable of performing arithmetic, logical, shift, and saturation operations

❑ Data Computation Unit containing the primary computation units of the CPU including a 40-bit arithmetic logic unit, two multiply-accumulate units (MACs), and a shifter

1.1.2 On-chip Memory

Features include:

- ❑ Dual-access RAM (DARAM) that supports two memory accesses per cycle
- ❑ Single-access RAM (SARAM) that supports one memory access per cycle
- ❑ ROM that provides non-volatile storage for program or data

1.1.3 On-chip Peripherals

Features include:

- ❑ Digital phase-locked loop (DPLL) clock generator (frequency multiplier) with external clock source
- ❑ Full-duplex multichannel buffered serial ports (McBSP)
- ❑ Direct memory access (DMA) controller
- ❑ 16-bit Enhanced Host Port Interface (EHPI)
- ❑ 16-bit timers with 4-bit prescalers
- ❑ External Memory Interface (EMIF) supporting access to asynchronous SRAM and EPROM, synchronous burst SRAM (SBSRAM) and synchronous DRAM (SDRAM)
- ❑ Configurable Instruction Cache
- ❑ Dedicated General-purpose I/O (GPIO) pins

1.1.4 Power Conservation

Features include:

- ❑ Software-programmable Idle Domains that provide configurable low-power modes
- ❑ Automatic power management
- ❑ Advanced low-power CMOS process

1.1.5 Emulation, Test, and Packaging

Features include:

- ❑ On-chip scan-based emulation capability with program history tracking of recent program counter values and discontinuities (Trace FIFO)
- ❑ IEEE 1149.1 (JTAG) Boundary Scan Test Capability
- ❑ Space-saving MicroStar BGA™ (Ball Grid Array) Packaging

Table 1–1. Characteristics of the C55x Processors

	VC5510
Memory	
On-chip SARAM	32K words (64K bytes)
On-chip DARAM	128K words (256K bytes)
On-chip ROM	16K words (32K bytes)
Total addressable Memory Space (internal + external)	8M words (16M bytes)
On-chip Bootloader (in ROM)	√
Peripherals	
McBSPs	3
DMA controller	√
EHPI (16-bit)	√
Configurable Instruction Cache	24K bytes
Timers	2
Programmable DPLL Clock Generator	√
General Purpose I/O pins	
Dedicated input/output	√
XF - dedicated output	1
Multiplexed with McBSP (input/output)	21
Multiplexed with Timer (output only)	2
CPU Cycle Time/Speed	
160 MHz (6.25 ns)	√
200 MHz (5 ns)	√
Package Types	
240-pin BGA	√

Architecture

This chapter discusses the C55x^{TM} DSP architecture which is built around four functional units. These units exchange program and data information with each other and with memory through dedicated internal buses. The chapter also discusses the seven-stage instruction pipeline as well as the six power-conserving IDLE domains.

Topic **Page**

2.1 Overview

The C55x architecture achieves power-efficient performance through increased parallelism and complete focus on reduction in power dissipation. The CPU supports an internal bus structure composed of:

- one program bus
- three data read buses
- two data write buses
- additional buses dedicated to peripheral and DMA activity

These buses provide the ability to perform up to three data reads and two data writes in a single cycle. In parallel, the DMA controller can perform up to two data transfers per cycle independent of CPU activity.

The C55x CPU provides two multiply-accumulate (MAC) units each capable of 17-bit x 17-bit multiplication in a single cycle. A central 40-bit arithmetic/logic unit (ALU) is supported by an additional 16-bit ALU. Use of ALUs is subject to instruction set control. This programmability provides the capacity to optimize parallel activity and power consumption. These resources are managed in the Address Data Flow Unit (AU) and Data Computation Unit (DU) of the C55x CPU.

The C55x architecture supports a variable byte width instruction set for improved code density. The Instruction Buffer Unit (IU) performs 32-bit program fetches from internal or external memory and queues instructions for the Program Unit (PU). The Program Unit decodes the instructions, directs tasks to AU and DU resources, and manages the fully-protected pipeline. Configurable instruction cache is also available to minimize external memory accesses improving data throughput and conserving system power.

Four Functional Units

The C55x architecture is built around four primary blocks: the Instruction Buffer unit (IU), the Program Flow unit (PU), the Address Data Flow Unit (AU) and the Data Computation Unit (DU). These functional units exchange program and data information with each other and with memory through multiple dedicated internal buses.

Figure 2–1 shows the principal blocks and bus structure in the C55x devices. Refer to Table 2–1 for a quick reference to bus usage.

Program fetches are performed using the 24-bit program address bus (PAB) and the 32-bit Program Read Bus (PB). The functional units read data from memory via three 16-bit Data Read Buses named B-bus (BB), C-bus(CB) and D-bus(DB). Each data read bus also has an associated 24-bit Data Read Address Bus (BAB, CAB and DAB). Single operand reads are performed on the D-bus. Dual-operand reads use C-bus and D-bus. B-bus provides a third read path and can be used to provide coefficients for dual-multiply operations.

Program and data writes are performed on two 16-bit Data Write Buses called E-bus (EB) and F-bus (FB). The write buses also have associated 24-bit Data Write Address Buses (EAB and FAB). Additional buses are present on the C55x devices to provide dedicated service to the DMA controller and the Peripheral Controller.

Figure 2–1. Functional Block Diagram

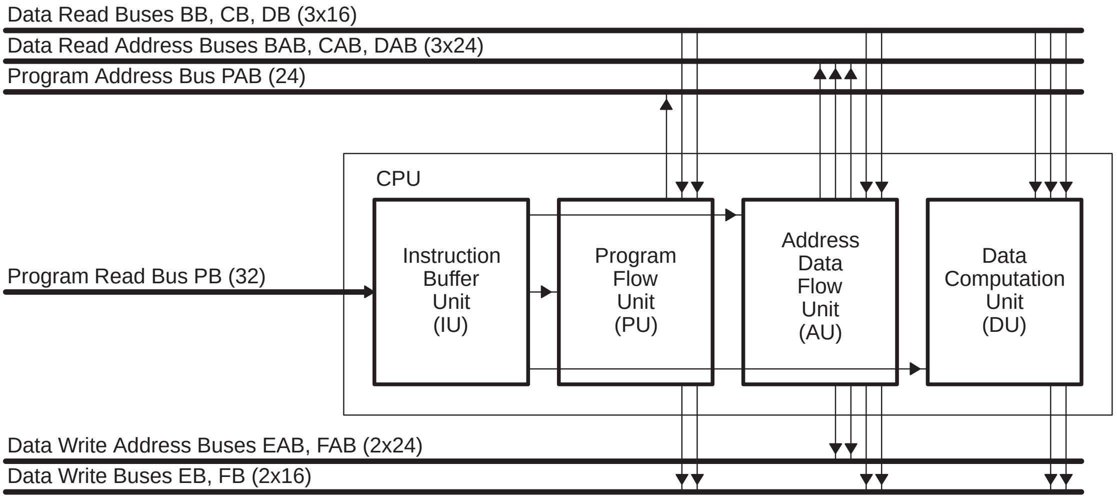

Data Read Buses BB, CB, DB (3x16)
Data Read Address Buses BAB, CAB, DAB (3x24)
Program Address Bus PAB (24)
CPU
Program Read Bus PB (32)
Instruction Buffer Unit (IU)
Program Flow Unit (PU)
Address Data Flow Unit (AU)
Data Computation Unit (DU)
Data Write Address Buses EAB, FAB (2x24)
Data Write Buses EB, FB (2x16)

Table 2–1. Bus Usage

Action Performed	Bus Used
Program fetches	24-bit Program Address Bus (**PAB**) 32-bit Program Read Bus (**PB**)
IU, AU, DU, and PU read data from memory	Three 16-bit Data Read Buses: **BB**, **CB,** and **DB** Associated 24-bit Data Read Address Buses: **BAB**, **CAB**, **DAB**
Single-operand reads	**D bus**
Dual-operand reads	**C bus** and **D bus**
Coefficient reads for dual-multiply operations	**B bus**
Program and data writes	Two 16-bit Data Write Buses: **EB** and **FB** Associated 24-bit Data Write Address Buses: **EAB** and **FAB**

2.2 Instruction Buffer Unit (IU or I Unit)

The Instruction Buffer Unit receives program code into its instruction buffer queue and decodes instructions. The I Unit then passes the appropriate information to the Program Flow Unit, Address Data Flow Unit and Data Computation Units for execution. The CPU fetches 32-bit packets from memory into the Instruction Buffer Queue (IBQ). The IBQ holds up to 64 bytes of instructions in queue to be decoded. The IBQ provides 6 bytes at a time to the instruction decoder which then dispatches actions to the other primary functional units in the CPU.

In addition to facilitating pipelining of instructions, the IBQ enables execution of a block of code stored completely within the queue (local repeat instruction) and speculative fetching of instructions while a condition is being tested for conditional goto, call, and return instructions.

2.3 Program Flow Unit (PU or P Unit)

The Program Flow Unit receives instructions from the I Unit and coordinates program flow actions including:

❑ Interpreting conditions for conditional instructions

❑ Determining branch (goto) addresses

❑ Initiating interrupt servicing when an interrupt is requested

❑ Managing single- and block-repeat operations

❑ Managing execution of parallel instructions

2.4 Address Data Flow Unit (AU or A Unit)

The Address Data Flow Unit is responsible for generating all of the addresses for data reads and writes. It can generate addresses based on immediate data from the I Unit or from pointers contained in auxiliary registers inside the A Unit. There are eight auxiliary registers for use as address pointers and coefficient data pointer registers to provide a dedicated pointer to a coefficient table. The registers for control of circular addressing are also managed by the A Unit.

The A Unit also contains a 16-bit ALU capable of performing arithmetical, logical, shift, and saturation operations.

2.5 Data Computation Unit (DU or D Unit)

The Data Computation Unit contains the primary computational units of the CPU. The D Unit is composed of a shifter, a 40-bit ALU, two MAC units, four 40-bit accumulator registers and two transition registers.

2.5.1 Shifter

The D-Unit shifter performs the following actions:

- ❑ Shifts 40-bit accumulator values up to 31 bits left or up to 32 bits right. The shift count can be read from a temporary register or can be supplied as a constant in the instruction.
- ❑ Shifts 16-bit register, memory or I/O space values up to 31 bits left or up to 32 bits right.
- ❑ Shifts 16-bit immediate values up to 15 bits left. The shift count is contained in the instruction as a constant.
- ❑ Normalizes accumulator values
- ❑ Extracts and expands bit fields, and performs bit counting
- ❑ Rotates register values
- ❑ Rounds and/or saturates accumulator values before they are stored to data memory

2.5.2 D-Unit Arithmetic Logic Unit (D-Unit ALU)

The D Unit contains a 40-bit ALU that accepts values from the I Unit and communicates calculation results to all of the other primary functional units. The functions of the D-Unit ALU include:

- ❑ Performs additions, subtractions, comparisons, rounding, saturation, Boolean logic operations, and absolute value calculations
- ❑ Performs two arithmetical operations simultaneously when a dual 16-bit arithmetic instruction is executed
- ❑ Tests, sets, clears and complements D-Unit register bits
- ❑ Moves register values

2.5.3 Multiply-Accumulate Units (MACs)

Two MAC units support multiplication and addition/subtraction. In a single cycle, each MAC can perform a 17-bit x 17-bit multiplication (fractional or integer) and a 40-bit addition or subtraction with optional 32-/40-bit saturation. The accumulators receive all the results or MAC operations. The presence of three dedicated read buses and two dedicated write buses provides the capability for sustained, single-cycle dual-MAC operations.

2.5.4 D-Unit Registers

The D Unit contains four 40-bit accumulators (AC0–AC3) that are used as source/destination for calculations performed by the MAC units and the ALU. The D Units also contains two 16-bit transition registers (TRN0, TRN1) that hold the transition decision path to new metrics to perform the Viterbi algorithm.

2.6 Instruction Pipeline

The C55x DSPs perform instruction fetching, decoding, and execution in seven stages as follows:

- ❑ **Fetch stage** reads program data from memory into the instruction buffer queue
- ❑ **Decode stage** decodes instructions and dispatches tasks to the other primary functional units
- ❑ **Address stage** computes addresses for data accesses and branch addresses for program discontinuities
- ❑ **Access1/Access 2 stages** send data read addresses to memory
- ❑ **Read stage** transfers operand data on B bus, C bus, and D bus
- ❑ **Execute stage** executes operation in the A unit and D unit, and performs writes on the E bus and F bus

The C55x pipeline is *protected* meaning it will *automatically insert cycles as necessary to prevent pipeline conflicts*. Pipeline protection cycles are inserted when:

- ❑ An instruction is supposed to write to a location but a previous instruction has not yet read that location (extra cycles are inserted so the read occurs first), or ...
- ❑ An instruction is supposed to read from a location but a previous instruction has not yet written to that location (extra cycles are inserted so the read occurs first)

2.7 IDLE Domains for Power Conservation

The flexible architecture of C55x devices provides a means to dynamically conserve power through software-programmable Idle Domains. Blocks of circuitry on the device are organized into idle domains. Each domain can operate normally or can be placed in a low-power idle state. The Idle Control Register (ICR) determines which domains will be placed in the idle state when the execution of the next IDLE instruction occurs. The six domains are**:**

- CPU domain
- DMA domain
- Peripherals domain
- Clock generator domain
- Instruction cache domain
- EMIF domain

Each domain can be placed in a low-power state when its capabilities are not required. This control provides the user the capability to dynamically modify the power consumption of the device based on activity. **Note that when each domain is in the idle state, the functions of that particular domain are not available**. An *exception* to this exists in the *peripheral domain*. In the peripheral domain, each peripheral has an Idle Enable bit which controls whether or not the peripheral will respond to the changes in the idle state. Thus, peripherals can be individually configured to idle or remain active when the peripheral domain is idled.

The idle state can be exited by modifying the ICR (if the CPU and clock generation domains were not idled) or by an external interrupt.

Chapter 3

Memory

This chapter discusses the available on-chip memory including DARAM, SARAM, and ROM.

Topic **Page**

3.1 On-chip Memory

The C55x™ DSP architecture provides access to a maximum of 8M words (16M bytes) of memory, including on-chip and external memory, organized as a single program/data space. General information about the available on-chip memory types is included in the subsections of this chapter.

Refer to Chapter 4, *Peripherals*, for information about the External Memory Interface (EMIF). For specific memory maps and configurations, refer to the device-specific data sheet.

3.1.1 On-chip Dual-Access RAM (DARAM)

DARAM supports two accesses (read or write) per memory block in a single cycle. Refer to the device-specific data sheet for DARAM block sizes on each device.

3.1.2 On-chip Single-Access RAM (SARAM)

SARAM supports one access (read or write) per memory block in a single cycle. Refer to the device-specific data sheet for SARAM block sizes on each device.

3.1.3 On-chip ROM

ROM provides non-volatile memory storage for program or data information. The on-chip ROM is single-access, meaning only one read can be performed at a time. Each ROM access requires two cycles.

Each device in the C55x generation is available with a standard ROM which may contain various components including a bootloader, data tables and software support functions. Refer to the device-specific data sheet for ROM contents descriptions on each device.

Customers may also arrange to have the C55x's ROM programmed with contents unique to their applications.

3.1.4 Bootloader

Bootloaders, stored in on-chip ROM, are available. At power-up, these bootloaders can automatically transfer user code from an external source to anywhere in memory. The bootloader may support transfer of program information from any of the following sources: EHPI, external asynchronous memory, or serial ports. Refer to the device-specific data sheet for descriptions of bootloader functions on each device.

Chapter 4

On-Chip Peripherals

All C55x™ DSP devices use the same CPU structure but are capable of supporting different on-chip peripherals and memory configurations. This chapter discusses the on-chip peripherals which include:

❑ Digital phase-locked loop (DPLL) clock generation
❑ Instruction Cache
❑ External Memory Interface (EMIF)
❑ Direct Memory Access (DMA) controller
❑ 16-bit Enhanced Host Port Interface (EHPI)
❑ Multichannel serial ports (McBSPs)
❑ 16-bit timers with 4-bit prescalers
❑ General-purpose I/O (GPIO) pins
❑ Trace FIFO (for emulation purposes only)

Peripheral control registers are mapped to an I/O space separate from the main memory space. Mapping of peripherals is device-specific as indicated in the data sheet. The Peripheral Bus Controller handles exchange of data between peripherals and the CPU via dedicated peripheral buses.

Topic **Page**

4.1 Digital Phase-Locked Loop (DPLL) Clock Generation

4.1.1 Clock Generator

The C55x devices include an on-chip clock generator which provides the ability to divide or multiply the input clock frequency for the desired CPU clock speed. Input clock multiplication is achieved by synthesizing the CPU clock using the DPLL. The clock generator has that has two modes of operation:

❑ Bypass mode

❑ Lock mode

In the *bypass mode*, the digital PLL is disabled, and the output clock is equal to the input clock divided by 1, 2 or 4. This mode can be used to save power when lower frequency operation is acceptable, since the digital PLL is disabled. This mode is also present while the digital PLL circuitry is locking.

In the *lock mode*, the DPLL is enabled, and the clock generator provides a synthesized output frequency which is locked to the input reference. The synthesized frequency is determined by a multiplication factor and a division factor. The multiplication factor can range from 2 to 31. The division factor can be 1, 2, 3 or 4. When lock mode is initiated, the device remains in bypass mode until the DPLL achieves lock on the input clock. The clock generator then automatically switches to lock mode. If lock on the input clock is lost at any time, the clock generator automatically switches back to bypass mode until the DPLL is relocked on the input clock.

The clock generator is is *bypass mode* after reset.

4.2 Instruction Cache

The C55x instruction cache provides 24k bytes of cache space for program information (code/data) fetched by the program unit. The instruction cache can be configured to utilize combinations of up to three cache structure types. The available cache types are:

❑ 2-way Set Associative

❑ 1-way Set Associative (Direct-mapped)

❑ Ramset

In the *2-way set associative mode*, the cache is structured similar to two arrays of direct-mapped cache. Each cache line has two tags. The 2-way block size is 16k bytes.

In *1-way set associative mode*, the cache is *a single direct-mapped array*. Each cache line has one tag. The 1-way block size is 8k bytes. The 2-way and 1-way modes provide more efficient cache performance for code that does not execute in a linear stream (branching code or calls/returns). In each of these modes, cache lines are replaced based on a least-recently-used algorithm.

In *Ramset mode*, the value of the tag is fixed and for each tag there is an array of 256 cache lines. In this mode, cache lines are not replaced unless the tag is changed or the cache is flushed. This mode provides better cache performance for frequently-used linear blocks of code (such as subroutines) that will fit within the Ramset 4k bytes block size. Multiple Ramset blocks are available depending on the cache configuration. See the device-specific data sheet for information on available cache configurations on each device.

4.3 External Memory Interface (EMIF)

The EMIF supports a glueless interface from the C55x to a variety of external memory devices. The supported memory types are shown in Table 4–1.

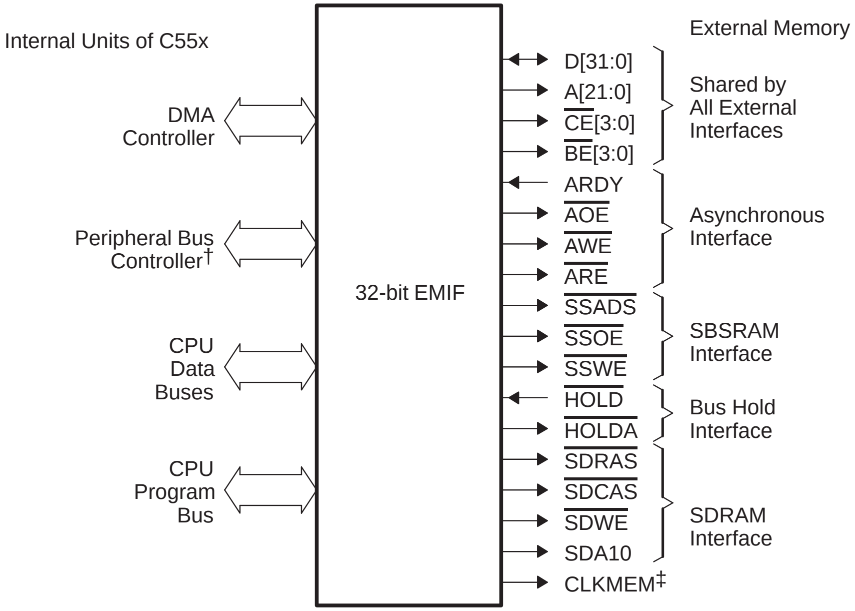
Table 4–1. Memory Types Supported by the C55x EMIF

Device Type	Memory	EMIF Timing Features	Benefit
Asynchronous	SRAM, ROM, Flash	Programmable strobe timing	Provides a high degree of flexibility to accommodate memory device timings
Synchronous	Synchronous burst SRAM (SBSRAM)	Runs at either 1/2× or 1× the CPU clock rate	Interface to high-speed memory with sustained throughput
	Synchronous DRAM (SDRAM)	Runs at either 1/2× or 1× the CPU clock rate	Interface to high-speed, high-density SDRAM

For each memory type, the EMIF supports 8-bit, 16-bit, and 32-bit accesses for both reads and writes. For writes, the EMIF controls the byte enable signals ($\overline{BE}$[3:0]) and the data bus to perform 8-bit transfers or 16-bit transfers. For reads, the entire 32-bit bus is read. Then, it is internally parsed by the EMIF.

The EMIF block diagram, Figure 4–1, shows the interface between external memory and the internal resources of the C55x.

Figure 4–1. Block Diagram of EMIF



[†] This connection allows the CPU to access the EMIF registers.
[‡] The CLKMEM signal is shared by the SDRAM and SBSRAM interfaces.

CPU configuration of the EMIF is performed through the peripheral bus controller. The EMIF services data transfer requests from three internal sources:

❑ Program fetches from the CPU

❑ Data accesses from the CPU

❑ Data accesses from the on-chip DMA controller

If multiple requests arrive simultaneously, the EMIF services each request in priority order, as shown in Table 4–2.

Table 4–2. EMIF Request Priority

Priority	Requestor
Highest	External Hold (initiated by the HOLD signal)
	Urgent Refresh
	CPU E bus
	CPU F Bus
	CPU D Bus
	CPU C Bus
	CPU P bus
	Cache
	DMA Controller
Lowest	Trickle SDRAM Refresh

4.3.1 EMIF Signal Descriptions

The EMIF signals can be grouped into functional categories as shown in Table 4–3. The address bus, the data bus, the chip enables and the byte enables are shared by all memory types. Other signals are dedicated to specific memory types as shown (with the exception of the CLKMEM signal, which is shared by both synchronous memory types).

The external memory spaces are selected by the CE signals. Each CE signal acts as a select for a different section of the external memory space. A memory type (the MTYPE field in the CE space control register) is selected for the entire address space associated with each CE signal. After configuration, when an access is made to a particular memory space, the EMIF signals perform as the selected memory type for that space. For instance, if space CE0 is configured for SDRAM, any time an access is made to that space, the access is performed as an SDRAM access. For information about the address ranges covered by each CE signal, consult the device-specific data sheet.

Table 4–3. EMIF Signal Descriptions

Signal Groups	EMIF Signal	I/O/Z	Description
Shared by All Memory Types	D[31:0]	I/OZ	32-bit data bus
	A[21:0]	O/Z	External address bus
	$\overline{\text{CE0}}$	O/Z	Active-low chip select for memory space CE0
	$\overline{\text{CE1}}$	O/Z	Active-low chip select for memory space CE1
	$\overline{\text{CE2}}$	O/Z	Active-low chip select for memory space CE2
	$\overline{\text{CE3}}$	O/Z	Active-low chip select for memory space CE3
	$\overline{\text{BE}}$[3:0]	O/Z	Active-low byte enables. Individual bytes and half-words can be selected for both read and write cycles.
Asynchronous Interface	ARDY	I	Ready. Active-high ready input used to insert wait states for slow memories.
	$\overline{\text{AOE}}$	O/Z	Active-low output enable for asynchronous memory interface
	$\overline{\text{AWE}}$	O/Z	Active-low write strobe for asynchronous memory interface
	$\overline{\text{ARE}}$	O/Z	Active-low read strobe for asynchronous memory interface
SBSRAM Interface	$\overline{\text{SSADS}}$	O/Z	Active-low address strobe/enable for SBSRAM interface Causes a new address to be registered
	$\overline{\text{SSOE}}$	O/Z	Active-low output buffer enable for SBSRAM interface
	$\overline{\text{SSWE}}$	O/Z	Active-low write enable for SBSRAM interface
Shared by SBSRAM and SDRAM	CLKMEM	O/Z	SDRAM memory interface clock 1/2× or 1× the CPU clock rate
SDRAM Interface	$\overline{\text{SDRAS}}$	O/Z	Active-low row strobe for SDRAM memory interface
	$\overline{\text{SDCAS}}$	O/Z	Active-low column strobe for SDRAM memory interface
	$\overline{\text{SDWE}}$	O/Z	Active-low write enable for SDRAM memory interface
	SDA10	O/Z	SDRAM A10 address line. Address line / autoprecharge disable for SDRAM memory. Serves as a row address bit (logically equivalent to A12) during ACTV commands and also disables the autoprecharging function of SDRAM during read or write operations
$\overline{\text{HOLD}}$ Interface	$\overline{\text{HOLD}}$	I	Active-low external bus hold (high-Z) request
	$\overline{\text{HOLDA}}$	O	Active-low external bus hold acknowledge

4.3.2 EMIF Registers

Within the EMIF, the following set of registers control EMIF configuration and behavior:

❑ EMIF Global Control Register

❑ EMIF Global Reset Register

❑ EMIF CE Space Control Registers 1,2,3

❑ EMIF SDRAM Control Register

❑ EMIF SDRAM Period Register

4.3.2.1 CE Space Control Registers

There are three CE space control registers per CE space. The MTYPE field identifies the memory type for the corresponding CE space. If MTYPE selects SBSRAM or SDRAM, the remaining fields in the register do not apply. If an asynchronous type is selected, the remaining fields specify the timing behavior of the address and control signals for access to that space.

A time-out period can also be configured for each space for the EMIF to generate a bus error interrupt to the CPU if the external memory fails to respond.

4.3.2.2 EMIF Global Control Register

The EMIF Global Control Register configures parameters common to all the CE spaces.

4.3.2.3 EMIF Global Reset Register

This register is used to reset the EMIF to a known condition without changing the current configuration values.

4.3.2.4 EMIF SDRAM control register

The SDRAM Control Register controls SDRAM parameters for all CE spaces that specify an SDRAM memory type in the MTYPE field of its associated CE Space Control Register. Since the SDRAM Control Register controls all SDRAM spaces, each space must contain SDRAM with the same refresh, timing and page characteristics. This register should not be modified while accessing SDRAM.

4.3.2.5 EMIF SDRAM Period and Counter Register

The SDRAM Refresh Period Register controls the refresh period for SDRAM in numbers of CLKMEM cycles.

4.3.2.6 EMIF SDRAM Initialization Register

Any write in this register, will bring about SDRAM initialization, in each CE space configured for SDRAM.

4.3.2.7 EMIF Bus Error Status Register

When an asynchronous access fails to respond within the time-out period an interrupt can be generated to the CPU. The Bus Error Status Register indicates the source of the failure as one of the CPU buses, the DMA bus, or an external CE space.

4.3.3 Asynchronous Interface

The asynchronous interface offers configurable memory cycle types that interface to a variety of memories such as SRAM, EPROM, and Flash memory.

4.3.3.1 Programmable Asynchronous Memory Parameters

The EMIF allows a high degree of timing flexibility for asynchronous accesses. The programmable parameters that allow this are:

- **Setup**: The time between the beginning of a memory cycle ($\overline{\text{CE}}$ low, address valid) and the activation of the read or write strobe
- **Strobe**: The time between the activation and deactivation of the read ($\overline{\text{ARE}}$) or write strobe ($\overline{\text{AWE}}$)
- **Hold**: The time between the deactivation of the read or write strobe and the end of the cycle (which can be either an address change or the deactivation of the $\overline{\text{CE}}$ signal)
- **Extended Hold**: The time after the last access (burst transfer or single access) that $\overline{\text{CE}}$ stays active.

These parameters are programmable in terms of CPU clock cycles via fields in the EMIF CE Space Control Registers. Separate setup, strobe, and hold parameters are available for read and write accesses. The SETUP and STROBE fields have a minimum count of 1 (a 0 in these fields will still be interpreted as a 1). For the first access in a set of consecutive accesses or a single access, the setup period will have a minimum of two. Programming the STROBE period provides the equivalent to software wait-states. In addition, extra cycles can be inserted into the strobe period by activating the ARDY input. The ready input is internally synchronized to the CPU clock and is sampled on each clock cycle. As long as the ARDY input remains low, wait states will be inserted into the STROBE period of the memory transfer. HOLD can be set to zero cycles. Extended Hold can be set to a minimum of one cycle.

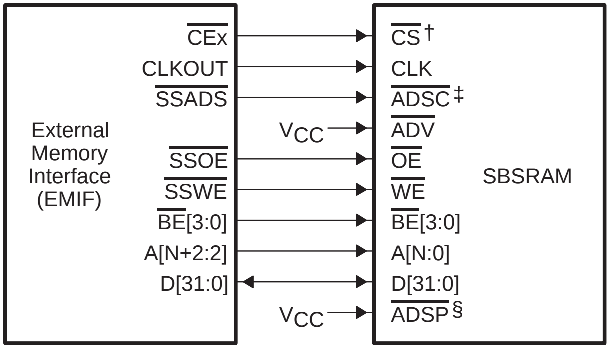
4.3.4 SBSRAM Interface

The EMIF interfaces directly to 32-bit wide industry-standard pipelined SBSRAMs. Flow-through SBSRAMs are not currently supported.

A typical EMIF to SBSRAM interface is shown in Figure 4–2.The three SBSRAM control signals are latched by the SBSRAM on the rising edge of

CLKMEM to determine the current operation. These signals are only valid if the chip select line for the SBSRAM is low. Typically, on the SBSRAM device, the ADV signal allows the SBSRAM to generate addresses internally for interface to controllers which cannot provide addresses quickly enough. However, the C55x EMIF does not use this signal because the device is capable of generating addresses on each cycle.

Figure 4–2. EMIF to SBSRAM Interface



[†] Chip Select

[‡] Synchronous Address Status Controller – Active low input which interrupts any outgoing burst, causing a new external address to be registered. If CEx is low, a read or write is performed using this new address. The ADSC is also used to place the chip into a power-down state when CEx is high.

[§] Synchronous Address Status Processor (ADSP) – Active low input interrupts any outgoing burst, causing a new external address to be registered. A read is performed using the new address, independent of byte write enables and ADSC.

SBSRAMs are *latent* by their architecture, meaning that *read data follows address and control information*. Consequently, the EMIF inserts cycles between read and write commands to ensure that no conflict exists on the D[31:0] bus. The initial three-cycle latency is present when changing directions on the bus, or on the first access of a burst sequence. Subsequent accesses in the burst sequence are single-cycle.

For detailed information on SBSRAM interface timing, see the device specific data sheet.

4.3.5 SDRAM Interface

The EMIF supports a glueless interface to 64-Mbit SDRAM, offering an interface to high speed and high-density memory. The EMIF supports the SDRAM commands shown in Table 4–4.

Table 4–4. EMIF SDRAM Commands

Command	Function
DCAB	Deactivate (also known as precharge) all banks
ACTV	Activate the selected bank and select the row
READ	Input the starting column address and begin the read operation
WRT	Input the starting column address and begin the write operation
MRS	Mode Register Set, configures SDRAM mode register
REFR	Auto refresh cycle with internal address

4.3.5.1 SDRAM Initialization

The EMIF performs the necessary functions to initialize SDRAM if any of the CE spaces are configured for SDRAM. An SDRAM initialization is requested by a write of 1 to the INIT in the EMIF SDRAM Control Register. This should not be done if an SDRAM access is occurring.

The actual sequence of events of an initialization is as follows:

1) Send a DCAB command to all CE spaces configured as SDRAM
2) Send three REFR refresh commands
3) Send an MRS command to all CE spaces configured as SDRAM

The DCAB cycle is performed immediately after reset, provided the $\overline{\text{HOLD}}$ input is not active. If $\overline{\text{HOLD}}$ is active, the DCAB command is not performed until the hold condition is removed. The external requester should not attempt to access any SDRAM banks in this case, unless it performs SDRAM initialization and control itself.

4.3.5.2 Monitoring Page Boundaries

Because SDRAM is a paged memory type, the EMIF SDRAM controller monitors the active row of SDRAM so that row boundaries are not crossed during the course of an access. To accomplish this monitoring, the EMIF stores the address of the open page, and performs compares against that address for subsequent accesses to the SDRAM bank. This storage and comparison is performed independently for each CE space. The number of address bits compared is a function of the page size programmed in the SDWID field in the EMIF SDRAM Control Register.

If, during the course of an access, a page boundary is crossed, the EMIF performs a DCAB command and starts a new row access. Simply ending the current access is not a condition, which forces the active SDRAM row to be closed. The EMIF speculatively leaves the active row open until it becomes necessary to close it. This feature decreases the deactivate-reactivate overhead and allows the interface to fully capitalize on address locality of memory accesses.

4.3.5.3 SDRAM Refresh

The RFEN bit in the SDRAM Control Register selects the SDRAM refresh mode of the EMIF. A value of 0 in the RFEN field disables all EMIF refreshes. The user must ensure that refreshes are implemented in an external device. A value of 1 in RFEN field enables the EMIF to perform refreshes of SDRAM.

The refresh command (REFR) is sent to all CE spaces configured to use SDRAM by the MTYPE field of the corresponding CE Space Control Register. REFR is automatically preceded by a DCAB command. This ensures all CE spaces selected with SDRAM are deactivated before refresh occurs. Page information is always invalid before and after a REFR command; thus a refresh cycle always forces a page miss on the next access. Following the DCAB command, the EMIF begins performing “trickle” refreshes at a rate defined by the PERIOD value in the EMIF SDRAM Control register, provided no other SDRAM access is pending. If refresh operations are delayed by the EMIF activity, the EMIF can automatically detect when the need to refresh becomes urgent and it will temporarily elevate the priority of refreshes over other EMIF accesses. When the refresh is completed, the priority returns to a “trickle” refresh that is only performed when the EMIF is not otherwise busy.

4.3.6 $\overline{\text{HOLD}}$ Interface

The EMIF responds to hold requests for the external bus. The hold handshake allows an external device and the EMIF to share the external bus. The handshake mechanism uses two signals:

- $\overline{\text{HOLD}}$: hold request input. The external device drives this pin low to request bus access. $\overline{\text{HOLD}}$ is the highest priority request that the EMIF can receive during active operation. When the hold is requested, the EMIF stops driving the bus at the earliest possible moment, which may entail completion of the current accesses and device deactivation. The external device must continue to drive $\overline{\text{HOLD}}$ low for as long as it wants to drive the bus. The $\overline{\text{HOLD}}$ input is internally synchronized to the CPU clock.

- $\overline{\text{HOLDA}}$: hold acknowledge output. The EMIF asserts this signal active after it has placed its signal outputs in the high-impedance state. The external device may then drive the bus as required. The EMIF places all outputs in the high-impedance state with the exception of the CLKOUT output.

External hold can be prevented by setting the NOHOLD bit in the EMIF global control register.

4.4 Direct Memory Access (DMA) Controller

The direct memory access (DMA) controller transfers data between locations in the memory space without intervention by the CPU. This movement of data to and from internal memory, external memory, and peripherals occurs in background of CPU operation.

The DMA has six independent, programmable channels allowing six different contexts for DMA operation. Each of the channels are executed in a time division multiplexed fashion.

Functional features of the DMA include:

- ❑ (After configuration) Operation independent of the CPU
- ❑ Six channels to keep track of the context of six independent block transfers
- ❑ Two levels of programmable priority between channels
- ❑ Data transfer sizes of 8-bits, 16-bits, and 32-bits (Capable of burst transfers of four 32-bit words to external memory)
- ❑ Configurable indices for each channel source and destination address modification
  - ■ Address may remain constant, post increment, post decrement, or be adjusted by a programmable value
- ❑ Individual transfers that may be triggered by selected events including McBSP receive or transmit events, timer events or external interrupts
- ❑ Individual DMA channels that may send an interrupt to the CPU on completion of a entire block, and/or of an entire frame, and/or half a frame transfer, and/or a synchronization drop
- ❑ Four ports, one for each set of data resources: SARAM, DARAM, EMIF, and the peripheral bus controller
- ❑ An additional dedicated port to the EHPI that allows direct transfers between EHPI and memory
- ❑ Automatic reload capability for the channel configuration registers that eliminates the need for the CPU to configure the channel after each block of transfers is complete (As long as the channel context has not changed, the channel will continue to automatically reinitialize. The context can be changed during operation and the new context will become valid when the current block transfer completes.)
- ❑ Maximum data transfer rate of two 16-bit words per cycle (two reads and two writes)
  - ■ Transfer rate can be achieved when using four different ports for the two data transfers

4.5 Enhanced Host Port Interface (EHPI)

The EHPI is a 16-bit wide parallel port through which a host processor can directly access the DSP’s memory space, with the host acting as master of the interface.

The EHPI communicates with memory via a dedicated auxiliary DMA channel and internal DMA buses that provide connectivity to the entire DSP’s internal memory and part of the DSP’s external memory.

Features of the EHPI include:

- ❑ 20-bit address bus used in non-multiplexed mode to allow access to all internal/external memory (See device-specific data sheet for memory maps.)
- ❑ 16-bit bidirectional data bus
- ❑ Multiple data strobes and control signals to allow glueless interfacing to a variety of hosts
- ❑ Multiplexed and non-multiplexed address/data modes
- ❑ 20-bit address register used in multiplexed mode (Includes address autoincrement feature for faster access to sequential addresses)
- ❑ Interface to on-chip DMA controller that provides access to memory space
- ❑ HRDY signal that provides host handshaking due to DMA latency
- ❑ Control register (available in multiplexed mode only) that is accessible by either the DSP or the host to provide host/DSP interrupts, handshaking, error conditions, and DSP reset control by the host
- ❑ Shared-access mode (SAM) where the EHPI, CPU and DMA all have access to the same memory space

4.5.1 Modes of Operation: Multiplexed and Non-multiplexed

There are two modes of operation as determined by the HMODE signal: multiplexed mode and non-multiplexed mode.

In *non-multiplexed mode*, address and data have dedicated buses. The internal control registers (HPIA, HPIC) are not available. Address is supplied via the 20-bit address bus.

In *multiplexed mode*, address and data information are multiplexed on the EHPI data bus. The EHPI address bus is not used. Two control pins, HCNTL0 and HCNTL1 indicate the type of transaction being performed. The internal address register (HPIA) contains the address for memory reads and writes. The internal EHPI control register (HPIC) contains configuration information and provides a path for control communication between the DSP and the host. Handshaking and error information are posted in the HPIC. The EHPI data register (HPID) contains that actual data read from or written to memory.

4.5.2 Operation During Low-Power Modes

By using automatic clock management logic, the EHPI can continue to operate during C55x idle states The clock management logic supplies the necessary clocks during EHPI/DMA transactions, and then stops the clocks when transactions are complete.

4.5.3 Loading Memory During Reset

The EHPI can download code and other memory contents while the DSP is in reset. When the EHPI is active, a device reset on the DSP puts the device in a reset state. During this state, the EHPI can load memory and then, when the initialization is complete, release the device from reset.

4.5.4 Emulation Considerations

The EHPI can continue operation even when the DSP CPU is halted due to debugger breakpoints or other emulation events.

4.6 Multichannel Buffered Serial Port (McBSP)

The McBSPs are high-speed, full-duplex serial ports that allow direct interface to other devices in a system such as other DSPs and codecs. McBSPs provide direct interface to:

- T1/E1 framers
- MVIP switching compatible and ST-BUS compliant devices
- IOM-2 compliant devices
- AC97 compliant devices
- IIS compliant devices
- SPI devices

The McBSPs are very similar to (and compatible with) those found on the C54x™ DSP and C6000™ DSP devices. McBSP capabilities include:

❑ Full-duplex communication

❑ Double-buffered data registers that allow a continuous data stream

❑ Independent framing and clocking for receives and transmits

❑ External shift clock generation or an internal programmable frequency shift clock

❑ Multichannel transmits and receives of up to 128 channels

❑ A wide selection of *data sizes* (see note below) including 8, 12, 16, 20, 24, or 32 bits

❑ μ-Law and A-Law companding

❑ 8-bit data transfers with LSB or MSB first

❑ Programmable polarity for both frame synchronization and data clocks

❑ Highly programmable internal clock and frame generation

Note:

Data sizes are referred to as ‘word’ or ‘serial word’ throughout the McBSP section. Therefore when ‘word’ is used, it can be either 8, 12, 16, 20, 24, or 32 bits in contrast to the true definition of word as being 16 bits.

The McBSP consists of a data path and control path. Seven pins (BCLKS, BDX, BDR, BFSX, BFSR, BCLKX and BCLKR) connect the control and data paths to external devices. The implemented pins can be programmed as GPIO pins if they are not used for serial communication.

The data is communicated to devices interfacing to the McBSP via the data transmit (BDX) pin for transmit and the Data Receive (BDR) pin for receives. Control information in the form of clocking and frame synchronization, is communicated via the BCLKX, BCLKR, BFSX, and BFSR pins. The device communicates to the McBSP via 16-bit wide control registers accessible via the peripheral bus. The CPU or DMA reads the received data from the Data Receive Register (DRR) and writes the data to be transmitted to the Data Transmit Register (DXR). The control block consists of internal clock generation, frame synchronization signal generation, and their control, and multichannel selection. This control block sends notification of important events to the CPU and DMA via two interrupt signals (XINT and RINT) and four event signals (XVET, RVET, XVETA and RVETA).

The on-chip companding (COMpress and exPAND) hardware allows compression and expansion of data in either μ-law or A-law format. When companding is used, transmit data is encoded according to specified companding law, and receive data is decoded to 2’s complement format.

The sample rate generator on the McBSP provides programmable clock and frame sync generation. Clocks and frame syncs can be programmed independently. The CPU clock, the CLKS pin, the CLKR pin, or the CLKX pin can be selected as reference clocks for the sample rate generator.

The McBSP allows the multiple channels to be independently selected for the transmitter and receiver. When the multiple channels are selected, each frame represents a time-division multiplexed (TDM) data stream. In using TDM data streams, the CPU may only need to process a few of them. Thus, to save memory and bus bandwidth, multichannel selection allows independent enabling of particular channels for transmission and reception. Up to 128 channels in a bit stream can be independently enabled, either fully independently or as blocks of 32 channels.

The clock stop mode (CLKSTP) in the McBSP provides compatibility with the SPI protocol. Clock stop mode works with only single-phase frames and one word per frame. When the McBSP is configured to operate in SPI mode, both the transmitter and the receiver operate together as a master or as a slave.

The McBSP provides a bit-field extraction and expansion mode called A-bis. In this mode, the McBSP can receive and transmit up to 128 8-bit channels on a PCM link. The receive section can extract all 1024 bits from a 1024 bit PCM frame according to a given receiving pattern. It then generates an interrupt to the CPU when the DRR is compacted with a 16-useful-bit word, or when a receive frame is ended. Likewise, the transmit section can expand up to 1024 bits into a 1024-bit PCM frame at a specific position according to a given transmitting pattern. It then generates an interrupt when a 16-bit word is transmitted or a transmit frame is ended.

The McBSP is fully static and operates at arbitrary low-clock frequency. The maximum frequency is CPU clock frequency divided by two.

4.7 Timers

The C55x devices include software-programmable 16-bit countdown timers with 4-bit prescalers providing a total 20-bit dynamic range. After programming the timer mode, period and prescaler values, the timer is enabled and counts down on system clock events. The timer can use the CPU clock or an external clock (provided through the TIN/TOUT pin) as a reference clock. When the timer count reaches zero, the timer can generate interrupt events to the CPU or the DMA controller and/or generate a pulse on the TIN/TOUT pin. If so configured, the timer can automatically reload its period and prescaler values to generate continuous periodic events.

Each timer has an associated TIN/TOUT pin. This pin can be configured as an input and used as an alternate timer clock source (instead of the CPU clock), or it can be configured as an output responding to time-out events. If configured as an output, the TIN/TOUT pin can either toggle or generate a single pulse on each time-out event. The pulse width and polarity are programmable through the Timer Control Register (TCR). If the TIN/TOUT pin is not used for a timer function, it can be configured as an additional general-purpose output pin.

4.8 General Purpose I/O (GPIO) Pins

The C55x devices provide individually selectable general-purpose input/output pins. Each pin can be individually configured as an input or an output. The pins are configured through two registers: IODIR and IODATA.

The IODIR register determines whether each GPIO pin functions as an input or an output. Each GPIO pin has a corresponding bit in the IODIR that controls its function.

The IODATA register contains the read or write data associated with each GPIO pin. If a GPIO pin is configured as an input in the IODIR register, the state of the pin can be read through the IODATA register. If a GPIO pin is configured as an output in the IODIR register, the contents of the IODATA register determine the state that will be driven on the associated pin.

The XF pin is a dedicated general-purpose output that is controlled by the XF bit in status register ST1.

In addition certain pins on the McBSPs and the Timers can be configured for use as general-purpose inputs or outputs. Refer to the *TMS320C55x Peripherals Reference Guide* (SPRU317) for more information.

4.9 Trace FIFO

The Trace FIFO is on-chip circuitry that is used exclusively by the emulator to save the last 16 PC discontinuities and the last 32 PC values to retrieve data values for impending operations. It requires an XDS510™ emulator to access the trace data and requires no direct software intervention by the user.

Chapter 5

Development Support

This chapter discusses tools available to support development of applications based on C55x™ DSPs. The device and support tool evolutionary cycle is also covered.

Topic **Page**

5.1 Development Tools

Texas Instruments offers an extensive line of development tools for the C55x generation of DSPs, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules.

5.1.1 Software Development Tools

The following products support development of applications based on C55x DSPs:

- ❏ Assembler/Linker
- ❏ Simulator
- ❏ Optimizing ANSI C compiler
- ❏ Application algorithms
- ❏ C/Assembly debugger and code profiler

5.1.2 Hardware Development Tools

Extended development system (XDS™) emulator (supports C55x multiprocessor system debug)

Detailed information on TI DSP development tool packages is available at: **www.ti.com/sc/docs/tools/dsp/index.html** on the web. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

The *TMS320 Third Party Support Reference Guide* (SPRU052) contains information from other companies in the industry regarding products related to the TMS320™ DSPs . To receive copies of TMS320™ DSP literature, contact the Literature Response Center at 800-477-8924.

For further information, call the Product Information Center (PIC) at 1-800-336-5236.

5.2 Device and Development Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all TMS320™ DSP devices and support tools. Each TMS320™ DSP device member has one of three prefixes: TMX, TMP, or TMS. Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

5.2.1 Device Development Evolutionary Flow

Devices are designated as follows :

TMX Experimental device that is not necessarily representative of the final device's electrical specifications

TMP Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification

TMS Fully-qualified production device

5.2.2 Support Tool Development Evolutionary Flow

Support tools are designated as follows:

TMDX Development support product that has not yet completed Texas Instruments internal qualification testing.

TMDS Fully qualified development support product

5.2.3 Points to Consider

❑ TMX and TMP devices and TMDX development support tools are shipped against the following disclaimer:

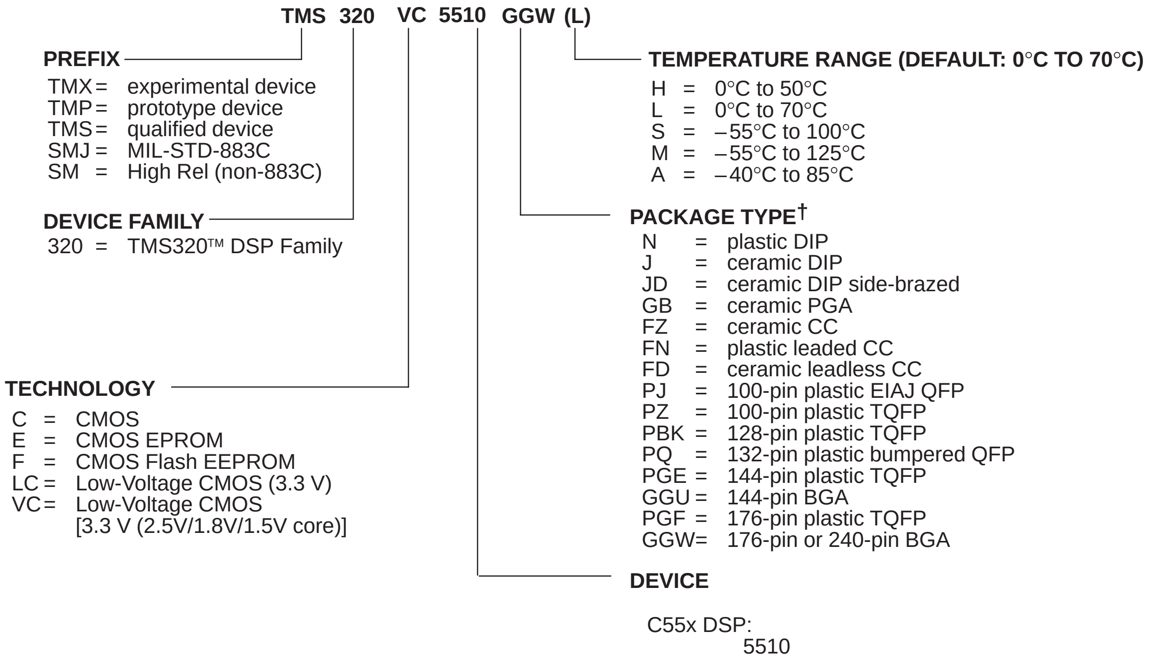
"Developmental product is intended for internal evaluation purposes."

❑ TMS devices and TMDS development support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

❑ Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

- ❑ TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, PZ, PGE, PBK, or GGU) and temperature range (for example, L). Figure 5–1 provides a legend for reading the complete device name for any TMS320™ DSP family member.

Figure 5–1. TMS320 DSP Device Nomenclature



TMS 320 VC 5510 GGW (L)

PREFIX

TMX	=	experimental device
TMP	=	prototype device
TMS	=	qualified device
SMJ	=	MIL-STD-883C
SM	=	High Rel (non-883C)

DEVICE FAMILY

320	=	TMS320™ DSP Family

TECHNOLOGY

C	=	CMOS
E	=	CMOS EPROM
F	=	CMOS Flash EEPROM
LC	=	Low-Voltage CMOS (3.3 V)
VC	=	Low-Voltage CMOS [3.3 V (2.5V/1.8V/1.5V core)]

TEMPERATURE RANGE (DEFAULT: 0°C TO 70°C)

H	=	0°C to 50°C
L	=	0°C to 70°C
S	=	–55°C to 100°C
M	=	–55°C to 125°C
A	=	–40°C to 85°C

PACKAGE TYPE†

N	=	plastic DIP
J	=	ceramic DIP
JD	=	ceramic DIP side-brazed
GB	=	ceramic PGA
FZ	=	ceramic CC
FN	=	plastic leaded CC
FD	=	ceramic leadless CC
PJ	=	100-pin plastic EIAJ QFP
PZ	=	100-pin plastic TQFP
PBK	=	128-pin plastic TQFP
PQ	=	132-pin plastic bumpered QFP
PGE	=	144-pin plastic TQFP
GGU	=	144-pin BGA
PGF	=	176-pin plastic TQFP
GGW	=	176-pin or 240-pin BGA

DEVICE

C55x DSP:
5510

† DIP = Dual-In-Line Package
PGA = Pin Grid Array
CC = Chip Carrier
QFP = Quad Flat Package
TQFP = Thin Quad Flat Package
BGA = Ball Grid Array

Chapter 6

Documentation Support

This chapter lists a variety of reference sources and document types available regarding TMS320™ DSP devices.

Topic **Page**

6.1 Related Documentation

Extensive documentation supports all TMS320™ DSP devices from product announcement through applications development. The following types of documentation are available to support the design and use of the C55x™ generation of DSPs:

❑ Functional Overview (such as this document)
❑ Device-specific data sheets
❑ Complete User Guides
❑ Development support tools documentation
❑ Hardware and software application reports

Printed versions of TI DSP documentation are available from the Literature Response Center at 1-800-477-8924.

6.1.1 Worldwide Web

Detailed documentation regarding the C55x generation of DSPs can be found at: **http://www.ti.com** on the Worldwide Web.

From the Texas Instruments home page, click the down arrow on the **Products** box and choose **Digitial Signal Processors**. Scroll down to the *TMS320C5000™ DSP Platform* listing and choose the **Parametric Table**. Locate the device you're interested in, and click on it. From the device-specific page, you can access data sheets, user guides, application reports, and development tools. Titles include:

6.1.1.1 Data Sheets

❑ *TMS320VC5510 Fixed-Point Digital Signal Processor (DSP)*, (SPRS076)

6.1.1.2 User Manuals

❑ *TMS320C55x DSP CPU Reference Guide*, (SPRU371)

❑ *TMS320C55x DSP Peripherals Reference Guide*, (SPRU317)

❑ *TMS320C55x DSP Programmer's Guide*, (SPRU376)

❑ *TMS320C55x DSP Mnemonic Instruction Set Reference Guide*, (SPRU374)

❑ *TMS320C55x DSP Algebraic Instruction Set Reference Guide*, (SPRU375)

❑ *TMS320C55x Optimizing C Compiler User's Guide*, (SPRU281)

❑ *TMS320C55x Assembly Language Tools User's Guide*, (SPRU280)

❑ *TMS320C55x DSP Library Programmer's Reference*, (SPRU422)

6.1.1.3 Development Tools

Code Composer Studio™ contains the following online guides:

❑ *TMS320C55x DSP Instruction Sets Online Reference Guide*

❑ *TMS320C55x DSP Registers Online Guide*

❑ *TMS320C55x DSP CPU Online Guide*

❑ *TMS320C55x DSP Peripherals Online Guide*

6.1.2 Publications

For general background information on DSPs and TI devices, see the three-volume publication *Digital Signal Processing Applications with the TMS320 Family* (literature numbers SPRA012, SPRA016, and SPRA017).

A series of DSP textbooks is published by Prentice-Hall and John Wiley & Sons to support digital signal processing research and education. The TMS320™ DSP newsletter, *Details on Signal Processing*, is published quarterly and distributed to update TMS320™ DSP customers on product information.

6.1.3 Comments and Questions

To send comments regarding the *TMS320C55x DSP Functional Overview* (SPRU312), use the **comments@books.sc.ti.com** email address, which is a repository for feedback. For questions and support, contact the Product Information Center (PIC) at 1-800-336-5236.

Appendix A

Glossary

A

accumulator: A register that stores the results of an operation and provides an input for subsequent arithmetic logic unit (ALU) operations.

ACTV: SDRAM command that activates the selected bank and selects the row.

address: The location of program code or data stored; an individually accessible memory location.

Address Data Flow Unit (AU, A Unit): Functional CPU unit that generates all addresses for data reads and writes.

A-law: Companding standard used in Europe.

ALU: See *arithmetic logic unit*.

arithmetic logic unit (ALU): The hardware of the CPU that performs arithmetic and logic functions.

AU: See *Address Data Flow Unit*.

A Unit: See *Address Data Flow Unit*.

B

byte enable: A signal controlled by the EMIF to perform 8-bit or 16-bit transfers for writes.

C

cache: A fast storage buffer for program code.

central processing unit (CPU): The unit that coordinates the functions of a processor.

chip enable: A signal that acts as a select for a different section of external memory space. Used during EMIF operation.

circular addressing: An address mode in which a finite set of addresses is reused by linking the largest address back to the smallest address.

clock cycles: A periodic or sequence of events based on the input from the external clock.

code: A set of instructions written to perform a task; a computer program or part of a program.

companding: Compressing and expanding of data in either μ-law or A-law format.

CPU: See *central processing unit.*

D

Data Computation Unit (DU, D Unit): Functional CPU unit that contains the primary computational units.

data memory: A region of memory used for storing or manipulating data, separate from the region used for storing program code.

DCAB: SDRAM command that deactivates all banks. Also known as precharge.

direct memory access (DMA): Memory access that does not use the CPU; used for data transfer directly between memory and a peripheral.

direct memory access (DMA) controller: Specialized circuitry that transfers data from memory to memory without using the CPU.

DMA: See *Direct-memory Access.*

DU: See *Data Computation Unit*.

D Unit: See *Data Computation Unit*.

E

EHPI: See *Enhanced Host Port Interface*.

Enhanced Host Port Interface: A 16-bit wide parallel port through which a host processor can directly access the DSP’s memory space, with the host acting as master of the interface.

EPROM: Erasable programmable ROM.

extended hold: The time after the last access (burst transfer or single access) that $\overline{\text{CE}}$ stays active during asynchronous accesses.

external memory interface (EMIF): Microprocessor hardware which is used to read from and write to off-chip memory.

F

fixed-point processor: A processor which does arithmetic operations using integer arithmetic with no exponents.

FLASH: Non-volatile asynchronous memory.

H

Hold: The time between deactivation of the read or write strobe and the end of the cycle (address change or deactivation of $\overline{\text{CE}}$ signal) during asynchronous accesses.

$\overline{\textbf{HOLD}}$: (pin) External bus hold (High-Z) request signal.

I

IDLE: A power-down mode.

IDLE domain: Sections of a device which can be selectively enabled or disabled under software control. When disabled, a domain enters a very low-power state in which register or memory contents are still maintained.

indirect addressing: An addressing mode in which an address points to another pointer rather than to the actual data.

Instruction Buffer Unit (IU, I Unit): Functional CPU unit that receives program code into its buffer queue and decodes instructions. It then passes information to the Program Flow Unit, the Address Data Flow Unit, and the Data Computation Unit for execution.

interrupt: A signal sent by hardware or software to request a processor's attention. An interrupt tells the processor to suspend its current operation, save the current task status, and perform a particular set of instructions. Interrupts communicate with the operating system and prioritize tasks to be performed.

IU: See *Instruction Buffer Unit*.

I Unit: See *Instruction Buffer Unit*.

L

latency: The delay between the occurrence of a condition and the reaction of the device. Also, in a pipeline, the necessary delay between the execution of two potentially conflicting instructions to ensure that the values used by the second instruction are correct.

M

MAC: See *multiply-accumulate unit*.

McBSP: See *multichannel buffered serial port*.

mode register set (MRS): Configures the SDRAM mode register.

MRS: See *mode register set*.

μ-law: Companding standard used in the United States and Japan.

multichannel buffered serial port (McBSP): High-speed, full-duplex serial ports that allow direct access to devices and codecs.

multiplexed mode: During EHPI operation, address and data information are multiplexed on the EHPI data bus.

multiplier: A CPU component that multiplies the contents of two registers.

multiply-accumulate unit (MAC): CPU unit capable of 17-bit x 17-bit multiplication in a single cycle.

N

non-multiplexed mode: During EHPI operation, address and data have dedicated buses.

P

parallelism: Sequencing events to occur simultaneously. Parallelism is achieved in a CPU by using instruction pipelining.

peripheral: A device connected to and usually controlled by a host device.

pipeline: A method of executing instructions in which the output of one process serves as the input to another, much like an assembly line. These processes become the stages or phases of the pipeline.

program fetch unit: The CPU hardware that retrieves program instructions.

Program Flow Unit (PU, P Unit): Functional CPU unit that receives instructions from the Instructional Buffer Unit and coordinates program flow actions.

program memory: A memory region used for storing and executing programs, separate from the region used for storing data.

PU: See *Program Flow Unit*.

P Unit: See *Program Flow Unit*.

R

Ramset mode: Cache structure that provides better cache performance for frequently-used linear blocks of code (subroutines) that fit within the Ramset 4k-bytes block size.

register: A small area of high speed memory, located within a processor or electronic device, that is used for temporarily storing data or instructions. Each register is given a name, contains a few bytes of information, and is referenced by programs.

reset: A means of bringing the CPU to a known state by setting the registers and control bits to predetermined values and signaling execution to start at a specified address.

S

saturation: A state where any further input no longer results in the expected output.

SBSRAM: Synchronous burst SRAM.

SDRAM: Synchronous DRAM.

set associative (1-way or direct mapped) Cache structure that provides more efficient cache performance for code that does not execute in a linear stream (branching code or calls/returns). Cache is a single direct-mapped array.

set associative (2-way) Cache structure that provides more efficient cache performance for code that does not execute in a linear stream (branching code or calls/returns). Cache is structured similar to two arrays of direct-mapped cache.

setup: The time between the beginning of a memory cycle and the activation of the read or write strobe during asynchronous accesses.

shifter: A hardware unit that shifts bits in a word to the left or to the right.

SRAM: Static RAM.

strobe: The time between activation and deactivation of the read or write strobe during asynchronous accesses.

T

Trace FIFO: On-chip circuitry that saves the last 16 PC discontinuities and the last 32 PC values to retrieve data values for impending operations. It is used exclusively by the emulator.

W

word: A set of bits that is stored, addressed, transmitted, or operated on as a unit.

Index