SMT374

User Manual

User Manual (QCF42); Version 3.0, 5/2/01; © Sundance Multiprocessor Technology Ltd. 2001

Revision History

Date	Comments	Engineer	Version
07/01/03	First rev, based on 376	GP	1.0.0
08/01/03	Tidied Virtex memory map	GP	1.0.1
07/03/03	Updated to match V1.2 of the Firmware	J.V.	1.0.1
30/06/03	SDB added V1.6 of the firmware.	J.V.	1. 0.3
07/07/03	DSP named corrected to DSPA and DSPB. Comport performances. V1.6 of the firmware.	J.V.	1. 0.4
06/08/03	Update Comport naming. Update reprogramming and external interface switch register to use the SMT6001 software	J.V.	1. 0.5
22/08/03	Clarify EMIF register table (i.e. memory map) Links added to firmware description document Version converted to 3 digits number	J.V.	1. 0.6
26/08/03	SDBC memory map corrected	J.V.	1.0.7
04/02/04	Reference to SMT376 page 26 removed	J.V.	1.0.8
14/04/04	Update to the external interface switch register to add a warning about dynamic setting in multiprocessor systems.	J.V.	1.0.9
29/09/04	Added address of the DPRAM for each processor	J.V.	1.1
04/08/05	Update: the user manual supports the new firmware implementation	SM	2.0
18/10/05	Corrected: SHB architecture	SM	2.1
14/02/06	Minor change: SHB architecture	SM	2.2
19/07/06	Updates for the SMT374_300	EW	2.3

Table of Contents

Contacting Sundance

You can contact Sundance for additional information by login onto the **Sundance support forum**.

Notational Conventions

DSP

The term DSP will be used throughout this document in place of TMS320C6211, 6711 or 6713.

SDB

The term SDB will be used throughout this document to refer to the Sundance Digital Bus interface.

SHB

The term SHB will be used throughout this document to refer to the Sundance High-speed Bus interface.

Register Descriptions

The format of registers is described using diagrams of the following form:

31–24	23–16	15–8	7–0
▨	OFLAGLEVEL	▨	▨
R,00000000	RW,10000000	R,00000000	R,10000000

The digits at the top of the diagram indicate bit positions within the register and the central section names bits or bit fields. The bottom row describes what may be done to the field and its value after reset. Shaded fields are reserved and should only ever be written with zeroes.

R	Readable by the CPU
W	Writeable by the CPU
RW	Readable and writeable by the CPU

Binary digits indicate the value of the field after reset

Outline Description

The SMT374 is a dual DSP, size 1 TIM offering the following features:

- TMS320C6211 integer processors running at 150MHz
- TMS320C6711 floating point processors running at 150MHz
- TMS320C6713 floating point processors running at 225MHz
- TMS320C6713-300 floating point processors running at 300MHz
- Six 20MB/s external communication ports (Comports)
- 256MB of SDRAM (100MHz, 128MB per DSP)
- 8MB Flash ROM
- Global Bus connector
- 2 SHB connectors for high-speed data transfer

Intended Audience

There are two existing versions of the firmware for the SMT374. These two versions differ by the number and the type of communication resources (comport and SDB interfaces) provided.

For each of the versions of the different firmware is loaded in the FPGA:

- Firmware version 1.8 or
- Firmware version 2.0

This user manual covers the version 2.0 of the firmware for the SMT374 implemented with the model described in the SMT6500 help file.

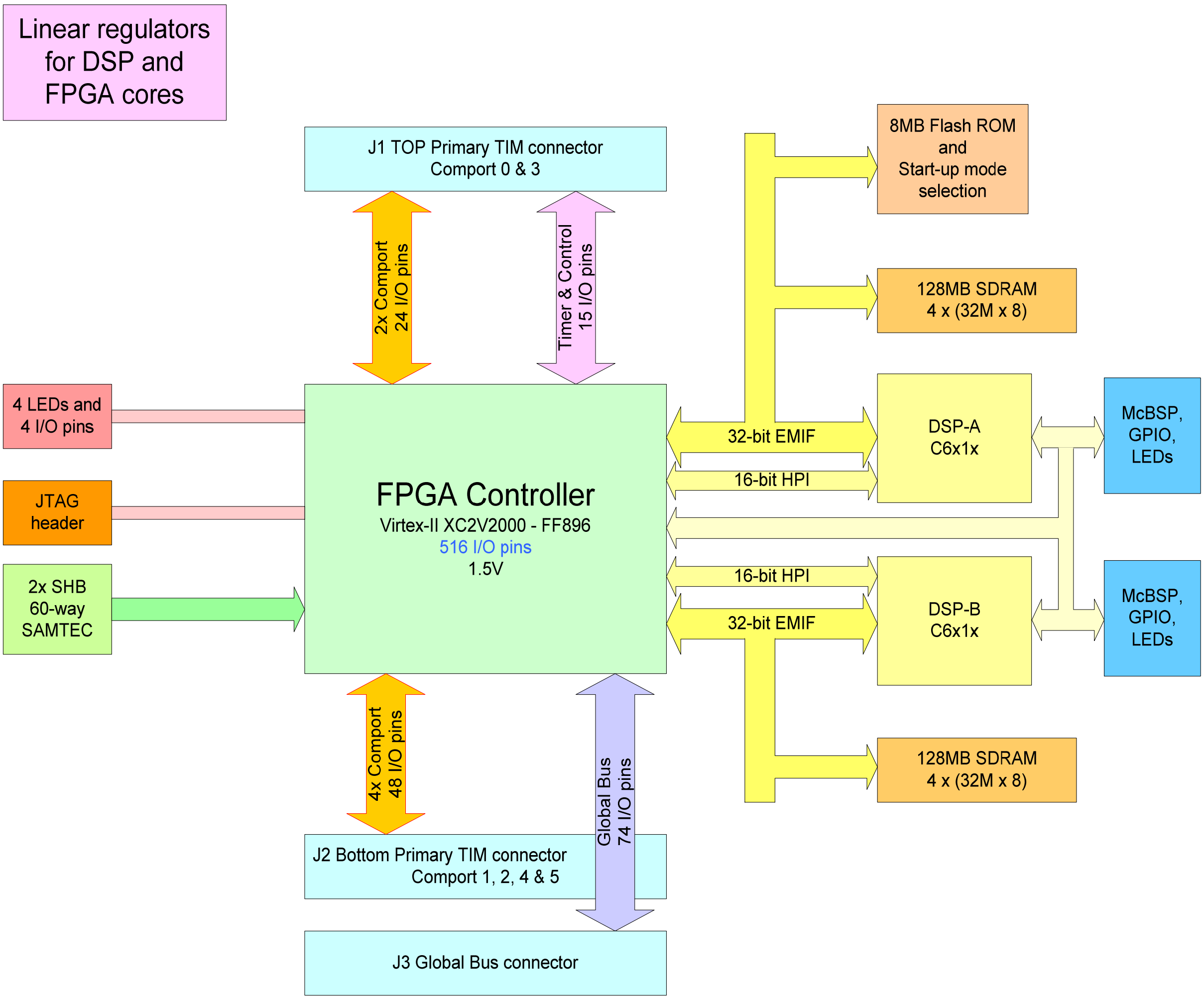
The changes between the firmware version 1.8 and version 2.0 are described in the section Firmware versions.

Block Diagram

The following drawing shows the block diagram of the SMT374 module.

The main components of the SMT374 are:

- Two Texas Instruments DSPs
- One Xilinx FPGA Virtex-II device
- 256MB of SDRAM



Architecture Description

DSPs

The two Texas Instruments DSPs can run up to 300MHz. Each of them is doted of 128MB of Synchronous DRAM (SDRAM).

The DSPs can be of two types:

- **TMS320C6211 or C6711**

  This is a fixed-point digital signal processor provided by Texas Instruments. The processor will run with zero wait states from internal SRAM. The internal memory is 64KB in size and can be partitioned between normal SRAM and L2 cache.

  A 37.5MHz on-board crystal oscillator provides the clock used for the DSP which then multiplies this by four internally. So the DSPs run at 150MHz.

- **TMS320C6713**

  The processor will run with zero wait states from internal SRAM. The internal memory is 256KB in size and can be partitioned between normal SRAM and L2 cache.

  A 37.5MHz on-board crystal oscillator provides the clock used for the DSP which then multiplies this by a programmable amount internally to provide the required core and EMIF clocks. In this case the DSPs will run at 225MHz.

- **TMS320C6713-300**

  The processor will run with zero wait states from internal SRAM. The internal memory is 256KB in size and can be partitioned between normal SRAM and L2 cache.

  A 50MHz on-board crystal oscillator provides the clock used for the DSP which then multiplies this by a programmable amount internally to provide the required core and EMIF clocks. In this case the DSPs will run at 300MHz.

Remark: SMT374_300 built before june 2006 may be fitted with a 37.5MHz instead of a 50MHz crystal oscillator. The DSPs will then run at 225MHz with the default bootloader. The SMT374_300 can be programmed with a “special” bootloader to make the DSPs run at 300MHz. Sundance will provide this “special” bootloader on demand:

http://support.sundance.com/viewtopic.php?p=19911#19911

Boot Mode

The two DSPs are called DSP-A and DSP-B. DSP-A is connected to the on-board flash ROM that contains the Sundance bootloader and the FPGA bitstream.

Following reset, DSP-A will automatically load the first 1KB from the flash ROM into its internal memory at address 0 and then start executing from there; DSP-B remains held in reset. DSP-A now explicitly loads the next 3KB from ROM, giving the effect of an initial load of 4KB. All this code is the Sundance *bootloader*, and it is made up of three parts: FPGA configuration, processor configuration, and the *Comport boot procedure*. FPGA configuration uses data in the ROM to configure the FPGA. Processor configuration sets the processor into a standard state, copies its comport boot procedure into a 2KB dual-port RAM (DPRAM) implemented in the FPGA, and releases DSP-B from reset. DSP-B is configured to boot from this DPRAM, and this leaves both DSPs executing their own copies of the *Comport boot procedure*.

The DPRAM is managed by writing to one of the *board control registers* (BCR) implemented in a CPLD. The BCR bit functions are described in the SMT6001 help file.

Board Control Registers (BCRs)

DSP-A will take approximately 800ms to configure the FPGA following reset, assuming a 150MHz clock. The external devices implemented in the FPGA (such as comports) must not be used during this configuration.

It is safest to wait for the configuration to complete. Note that comports will appear to be "not ready" until the FPGA has been configured.

The FPGA programming algorithm is not described here. It can be found in the boot code.

DPRAM

The DPRAM in the FPGA is only intended to be used during this boot process; more general use is not recommended. The DPRAM is accessible from the following locations:

- DSPA has access to the DPRAM from address 0xB0100000
- DSPB has access to the DPRAM from address 0x90000000

EMIF Control Registers

The DSP has a single external memory interface (EMIF) which is 32 bits wide.

A full description of the registers used to control the EMIF can be found in the *DSP C6000 Peripherals Reference Guide*.

The standard bootstrap will initialise these registers to use the following resources:

Memory space	Setting	Resource DSP-A	Resource DSP-B	Address range
CE0	0x00000030	SDRAM 128MB	SDRAM 128MB	0x80000000 - 0x87FFFFFF
CE1	0xFFFFFF23	Flash 8MB divided in 4x2MB pages	Virtex (2KB DPRAM with the Virtex)	0x90000000 - 0x901FFFFF
CE2	0x00100020	Control register		0xA0000000 - 0xA0000010
CE3	0x00000030	Virtex	Virtex	0xB0000000 - 0xB7FFFFFF

SDRAM

Memory spaces CE0 are used to access 128MB of SDRAM over the EMIF.

The speed of the SDRAM is dependent on the processor variant. Using the C6x11, the SDRAM will operate at 100MHz.

Using the C6713, the SDRAM operates at a programmable rate up to the maximum allowed on the EMIF (TI data sheet = TBD[1]).

The EMIF CE0 & CE3 memory space control registers should be programmed with the value 0x00000030.

FLASH

An 8MB Flash ROM is connected to DSP-A in the EMIF CE1 memory space. It cannot be accessed by DSP-B. The ROM holds boot code for the DSP, configuration data for the FPGA, and optional user-defined code.

A software protection algorithm is in place to prevent programs accidentally altering the ROM's contents. Please contact Sundance for further information about re-programming this device.

The CE1 memory space control register should be programmed with the value 0xFFFFFF23. This is the same value for DSP-B, even though this DSP does not have access to the flash. When DSP-B accesses memory on CE1, the Virtex will respond with data within its internal dual port RAM (using the standard Sundance Virtex configuration).

[1] To Be Defined

FPGA

The FPGA (Field Programmable Gate Array) is a Xilinx Virtex-II device.

It implements the following communication resources:

- Six comport interfaces
- Two 32-bit Sundance digital bus interfaces
- One global bus interface

Version control

Revision numbers for both the boot code and FPGA firmware are stored in the Flash ROM during programming as zero-terminated ASCII strings.

The SMT6001 utility can be used to display the version numbers of the bootloader and the FPGA data.

Firmware versions

The SMT6001 utility includes the latest version of the bootloader and the latest version of the FPGA data that implements the FPGA architecture described in the SMT6500 help file.

Note that the new firmware does not support the DSPA_SWITCH: only two 32-bit SDB interfaces are presented on the TIM connectors and their configuration is fixed. The configuration of the six internal comports is also fixed.

Customers whom wish to use the old firmware that supported the 16-bit options and the DSPA_SWITCH can obtain it from our support web forum.

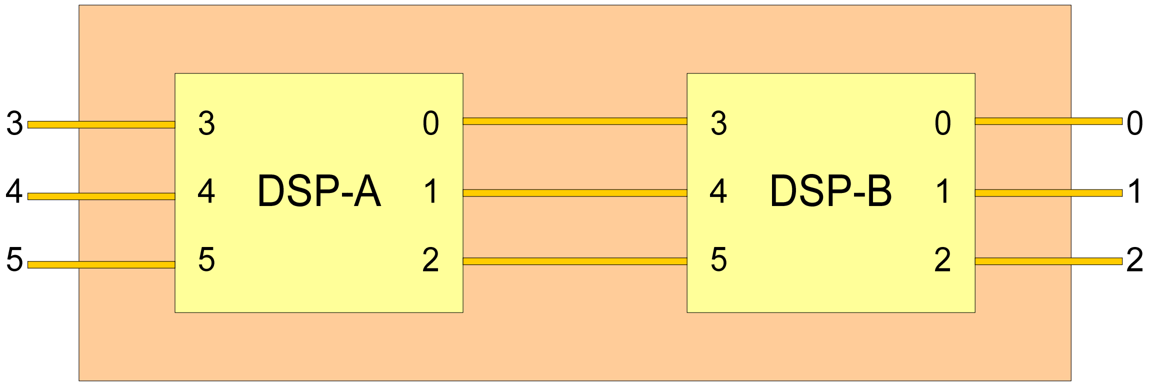
Reprogramming the firmware and boot code

The contents of the flash ROM are managed using the SMT6001 utility. This includes the latest firmware and bootloader along with complete documentation on how to reprogram the ROM. The utility assumes that you have Code Composer Studio installed and that it has been configured correctly for the installed TIMs. The Sundance Wizard can help you with this.

As DSP-B has no connection to the ROM, the programming must be done using DSP-A. To confirm that the ROM has been programmed correctly, you should run the confidence test in the BoardInfo utility (SMT6300).

Comports

Both DSP-A and DSP-B have six Comports, numbered 0 to 5. Each DSP has three Comports connected to the other DSP and three connected to the TIM connectors, as shown in the following diagram:



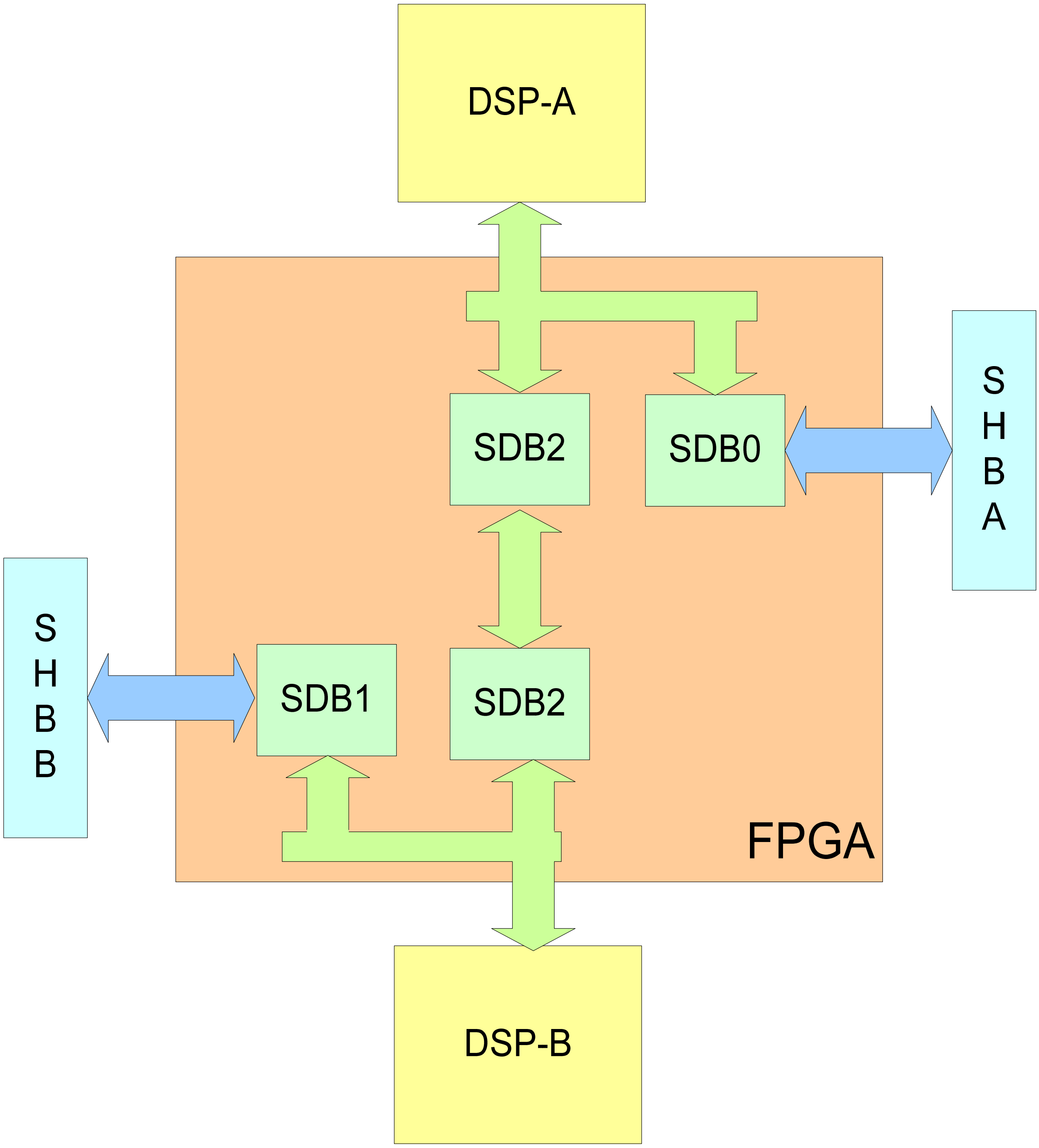
The addresses of the Comport registers are shown in the Virtex Memory Map, and are described in the SMT6400 help file.

SHB

The SMT374 has two SHB connectors. These interfaces operate with a fixed clock rate of 100MHz.

Architecture

SDB0 on DSP-A and SDB1 on DSP-B are presented on the TIM's SHB connectors, SHBA and SHBB respectively. SDB2 of DSP-A has a fixed internal connection to SDB2 of DSP-B.



The addresses of the SDB registers are shown in the Virtex Memory Map, and are described in the SMT6400 help file.

Note that TI's C6713 processors are unable to move data fast enough to use the full bandwidth of 32-bit SDBs. Observed maximum transfer rates using EDMA are 340MB/s (internal memory) and 130MB/s (external memory). TI claims a potential bandwidth of 400MB/s. This is currently under investigation.

Global bus

The SMT374 provides a single global bus interface. This is only accessible from DSP-A. The addresses of the global bus registers are shown in the Virtex Memory Map, and are described in the SMT6400 help file.

LED Setting

The SMT374 has 5 LEDs. One shows the FPGA configuration status and the other 4 are under DSP control (2 each).

Two output TTL I/O pins are available on connector JP2 for control or debugging. Their values can be controlled by bits in the LED register.

LED D2 always displays the state of the FPGA DONE pin. This LED is off when the FPGA is configured (DONE=1) and on when it is not configured (DONE=0).

This LED should go on when the board is first powered up and go off when the FPGA has been successfully programmed (this is the standard operation of the boot code resident in the flash memory device). If the LED does not light at power-on, check that you have the mounting pillars and screws fitted properly. If it stays on, the DSP is not booting correctly, or is set to boot in a non-standard way.

LED Register DSP-A

(0xB00D0000)

31–4	3	2	1	0
	TTL1	TTL0	LED D4	LED D3
	RW,0	RW,0	RW,0	RW,0

LED Register DSP-B

(0xB00D0000)

31–4	3	2	1	0
	TTL3	TTL2	LED D6	LED D5
	RW,0	RW,0	RW,0	RW,0

CONFIG & NMI

The TIM specification describes the operation of an open-collector type signal CONFIG that is driven low after reset.

This signal, on a standard C4x based TIM, is connected to the processor's IIOF3 pin. On the SMT374, the CONFIG signal is asserted after power on, and can be released by writing the value (1<<6) to the config register. Conversely, CONFIG may be re-asserted by writing 0 to this bit. It is not possible for software to read the state of the CONFIG signal.

The NMI signal from the TIM connector can be routed to the DSP NMI pin.

WARNING: Several software components include code sequences that assume setting GIE=0 in the DSP CSR will inhibit all interrupts; NMI violates that assumption. If an NMI occurs during such code sequences it may not be safe to return from the interrupt. This may be particularly significant if you are using the compiler's software pipelining facility.

Each DSP has access to a separate config register.

Config Register

31–8	7	6	5–0
	NMI	CONFIG	

Field	Description	
CONFIG	0	drive CONFIG low
	1	tri-state CONFIG
NMI	0	Disconnect NMI from the DSP
	1	Connect NMI from TIM to the DSP.

Config and NMI DSP lines are described in the SMT6400 help file.

Timer

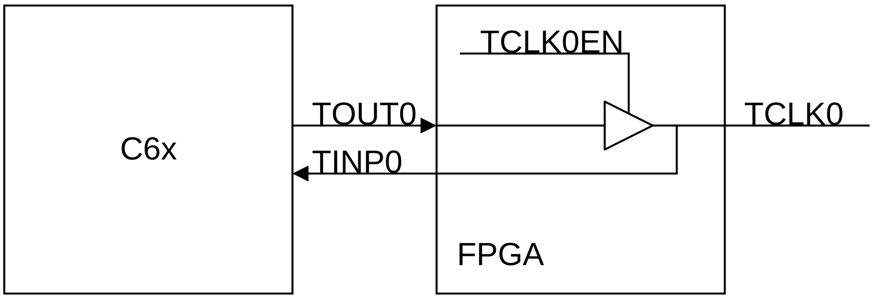
The TIM TCLK0 and TCLK1 signals can be routed to the DSP’s TOUT/TINP pins. The signal direction must be specified, together with the routing information in the timer control register.

Timer Control Register

31–6	5	4	3–0
Reserved	TCLK1	TCLK0	Reserved

Field	Description	
TCLK0	0	TIM TCLK0 is an input
	1	Enable TIM TCLK0 as an output
TCLK1	0	TIM TCLK1 is an input
	1	Enable TIM TCLK1 as an output

If the TIM TCLKx pin is selected as an output, the DSP TOUTx signal will be used to drive it. The TIM TCLKx pin will always drive the DSP TINPx input.



The Timer control register is described in the SMT6400 help file.

Code Composer

This module is fully compatible with the Code Composer Studio debug and development environment. This extends to both the software and JTAG debugging hardware. The driver to use is the *tixds6x1x.dvr* and each SMT374 should be declared as **two DSPs**. The ROM reprogramming utility, SMT6001, requires CCS version 3.0 or later.

Application Development

Depending on the complexity of your application, you can develop code for SMT374 modules in several ways.

SMT6400

For simple applications, the Sundance SMT6400 software support package (project examples) and its associated header files (SmtTim.h and ModSup.h) can suffice.

The SMT6400 product is installed by the Sundance Wizard and it is free of charge.

3L Diamond

This module is fully supported by 3L Diamond, which Sundance recommends for all but the simplest of applications. An SMT374 has to be declared in configuration files as two processors of type SMT374_6711 or SMT374_6713, as appropriate.

SMT6500

This is the support package for the FPGA. It may be used to develop your application in the FPGA of the module.

Operating Conditions

Safety

The module presents no hazard to the user.

EMC

The module is designed to operate within an enclosed host system that provides adequate EMC shielding. Operation within the EU EMC guidelines is only guaranteed when the module is installed within an appropriate host system.

The module is protected from damage by fast voltage transients introduced along output cables from outside the host system.

Short-circuiting any output to ground does not cause the host PC system to lock up or reboot.

General Requirements

The module must be fixed to a TIM40-compliant carrier board.

The SMT374 TIM is in a range of modules that must be supplied with a 3.3V power source. In addition to the 5V supply specified in the TIM specification, these new generation modules require an additional 3.3V supply to be presented on the two diagonally-opposite TIM mounting holes. The lack of this 3.3V power supply should not damage the module, although it will obviously be inoperable; prolonged operation under these circumstances is not recommended.

This module is not directly compatible with earlier generations of TIM motherboards, although the 3.3V supply can be provided from a separate source. It is, however, compatible with the latest generation of Sundance TIM carrier boards such as the SMT310Q and subsequent versions (PCI), and SMT328 (VME), which present the 3.3V via conductive mounting pillars.

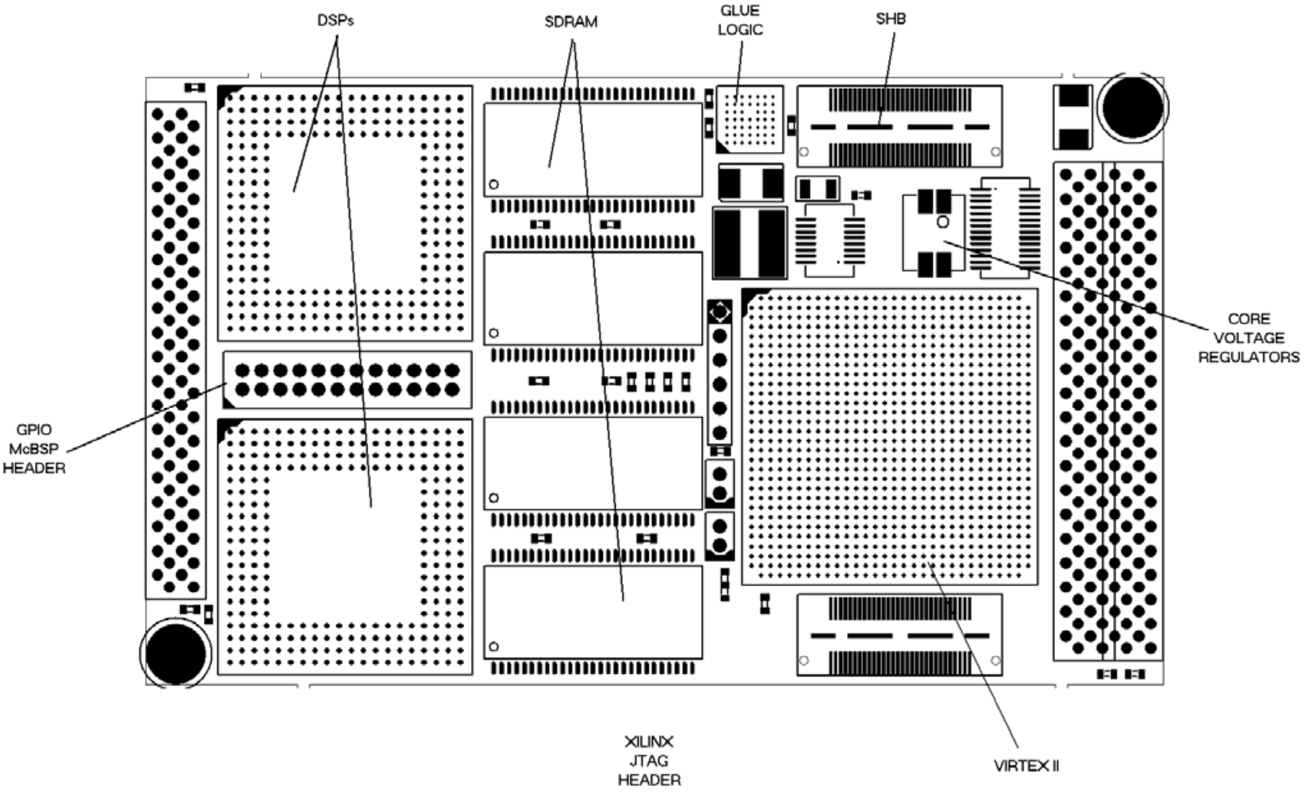
Use of the TIM on SMT327 (cPCI) motherboards may require a firmware upgrade. If LED D6 on the SMT374 remains illuminated once the TIM is plugged in and powered up, the SMT327 needs the upgrade. The latest firmware is supplied with all new boards shipped. Please contact Sundance directly if you have an older board and need the upgrade.

The external ambient temperature must remain between 0°C and 40°C, and the relative humidity must not exceed 95% (non-condensing).

Power Consumption

The power consumption of this TIM is dependent on the operating conditions in terms of core activity and I/O activity. The maximum power consumption is 6W.

Connector Positions



Virtex Memory Map

The memory mapping is as follows:

```
#define DSPA_CP0                    (volatile unsigned int *)0xB0000000
#define DSPA_CP1                    (volatile unsigned int *)0xB0008000
#define DSPA_CP2                    (volatile unsigned int *)0xB0010000
#define DSPA_CP3                    (volatile unsigned int *)0xB0018000
#define DSPA_CP4                    (volatile unsigned int *)0xB0020000
#define DSPA_CP5                    (volatile unsigned int *)0xB0028000
#define DSPA_CP0_STAT               (volatile unsigned int *)0xB0004000
#define DSPA_CP1_STAT               (volatile unsigned int *)0xB000C000
#define DSPA_CP2_STAT               (volatile unsigned int *)0xB0014000
#define DSPA_CP3_STAT               (volatile unsigned int *)0xB001C000
#define DSPA_CP4_STAT               (volatile unsigned int *)0xB0024000
#define DSPA_CP5_STAT               (volatile unsigned int *)0xB002C000
#define DSPA_GBSTAT                 (volatile unsigned int *)0xB0034000
#define DSPA_SDBSTAT                (volatile unsigned int *)0xB0038000
#define DSPA_STAT                   (volatile unsigned int *)0xB003C000
#define DSPA_SDBA                   (volatile unsigned int *)0xB0040000
#define DSPA_SDBA_STAT              (volatile unsigned int *)0xB0048000
#define DSPA_SDBA_INPUTFLAG         (volatile unsigned int *)0xB0044000
#define DSPA_SDBA_OUTPUTFLAG        (volatile unsigned int *)0xB004C000
#define DSPA_SDBB                   (volatile unsigned int *)0xB0050000
#define DSPA_SDBB_STAT              (volatile unsigned int *)0xB0058000
#define DSPA_SDBB_INPUTFLAG         (volatile unsigned int *)0xB0054000
#define DSPA_SDBB_OUTPUTFLAG        (volatile unsigned int *)0xB005C000
#define DSPA_SDBC                   (volatile unsigned int *)0xB0060000
#define DSPA_SDBC_STAT              (volatile unsigned int *)0xB0068000
#define DSPA_SDBC_INPUTFLAG         (volatile unsigned int *)0xB0064000
#define DSPA_SDBC_OUTPUTFLAG        (volatile unsigned int *)0xB006C000
#define DSPA_GLOBAL_BUS             (volatile unsigned int *)0xB00A0000
#define DSPA_GLOBAL_BUS_CTRL        (volatile unsigned int *)0xB0080000
#define DSPA_GLOBAL_BUS_START       (volatile unsigned int *)0xB0088000
#define DSPA_GLOBAL_BUS_LENGTH      (volatile unsigned int *)0xB0090000
#define DSPA_TCLK                   (volatile unsigned int *)0xB00C0000
#define DSPA_TIMCONFIG              (volatile unsigned int *)0xB00C8000
#define DSPA_LED                    (volatile unsigned int *)0xB00D0000
#define DSPA_IIOF                   (volatile unsigned int *)0xB00D8000
```

```
#define DSPA_INTCTRL4                  (volatile unsigned int *)0xB00E0000
#define DSPA_SDBINTCTRL4               (volatile unsigned int *)0xB00E4000
#define DSPA_INTCTRL5                  (volatile unsigned int *)0xB00E8000
#define DSPA_SDBINTCTRL5               (volatile unsigned int *)0xB00EC000
#define DSPA_INTCTRL6                  (volatile unsigned int *)0xB00F0000
#define DSPA_SDBINTCTRL6               (volatile unsigned int *)0xB00F4000
#define DSPA_INTCTRL7                  (volatile unsigned int *)0xB00F8000
#define DSPA_SDBINTCTRL7               (volatile unsigned int *)0xB00FC000
#define DSPA_DPRAM                     (volatile unsigned int *)0xB00FC000
#define DSPB_CP0                       (volatile unsigned int *)0xB0000000
#define DSPB_CP1                       (volatile unsigned int *)0xB0008000
#define DSPB_CP2                       (volatile unsigned int *)0xB0010000
#define DSPB_CP3                       (volatile unsigned int *)0xB0018000
#define DSPB_CP4                       (volatile unsigned int *)0xB0020000
#define DSPB_CP5                       (volatile unsigned int *)0xB0028000
#define DSPB_CP0_STAT                  (volatile unsigned int *)0xB0004000
#define DSPB_CP1_STAT                  (volatile unsigned int *)0xB000C000
#define DSPB_CP2_STAT                  (volatile unsigned int *)0xB0014000
#define DSPB_CP3_STAT                  (volatile unsigned int *)0xB001C000
#define DSPB_CP4_STAT                  (volatile unsigned int *)0xB0024000
#define DSPB_CP5_STAT                  (volatile unsigned int *)0xB002C000
#define DSPB_STAT                      (volatile unsigned int *)0xB003C000
#define DSPB_SDBSTAT                   (volatile unsigned int *)0xB0038000
#define DSPB_SDBA                      (volatile unsigned int *)0xB0040000
#define DSPB_SDBA_STAT                 (volatile unsigned int *)0xB0048000
#define DSPB_SDBA_INPUTFLAG            (volatile unsigned int *)0xB0044000
#define DSPB_SDBA_OUTPUTFLAG           (volatile unsigned int *)0xB004C000
#define DSPB_SDBB                      (volatile unsigned int *)0xB0050000
#define DSPB_SDBB_STAT                 (volatile unsigned int *)0xB0058000
#define DSPB_SDBB_INPUTFLAG            (volatile unsigned int *)0xB0054000
#define DSPB_SDBB_OUTPUTFLAG           (volatile unsigned int *)0xB005C000
#define DSPB_SDBC                      (volatile unsigned int *)0xB0060000
#define DSPB_SDBC_STAT                 (volatile unsigned int *)0xB0068000
#define DSPB_SDBC_INPUTFLAG            (volatile unsigned int *)0xB0064000
#define DSPB_SDBC_OUTPUTFLAG           (volatile unsigned int *)0xB006C000
#define DSPB_TCLK                      (volatile unsigned int *)0xB00C0000
#define DSPB_TIMCONFIG                 (volatile unsigned int *)0xB00C8000
#define DSPB_LED                       (volatile unsigned int *)0xB00D0000
```

```
#define DSPB_IIOF                    (volatile unsigned int *)0xB00D8000
#define DSPB_INTCTRL4                (volatile unsigned int *)0xB00E0000
#define DSPB_SDBINTCTRL4             (volatile unsigned int *)0xB00E4000
#define DSPB_INTCTRL5                (volatile unsigned int *)0xB00E8000
#define DSPB_SDBINTCTRL5             (volatile unsigned int *)0xB00EC000
#define DSPB_INTCTRL6                (volatile unsigned int *)0xB00F0000
#define DSPB_SDBINTCTRL6             (volatile unsigned int *)0xB00F4000
#define DSPB_INTCTRL7                (volatile unsigned int *)0xB00F8000
#define DSPB_SDBINTCTRL7             (volatile unsigned int *)0xB00FC000

#define DSPB_DPRAM                   (volatile unsigned int *)0x90000000
```

Connector Pin-outs

Serial Ports & Other DSP I/O (JP2 connector)

Pin number	Signal	Signal	Pin number
1	DSPA MCBSP CLKS1 (input)	DSPA MCBSP CLKR1 (input/output)	2
3	DSPA MCBSP CLKX1 (input/output)	DSPA MCBSP DR1 (input)	4
5	DSPA MCBSP DX1 (input/output)	DSPA MCBSP FSR1 (input/output)	6
7	DSPA MCBSP FSX1 (input/output)	GND	8
9	DSPB MCBSP CLKS1 (input)	DSPB MCBSP CLKR1 (input/output)	10
11	DSPB MCBSP CLKX1 (input/output)	DSPB MCBSP DR1 (input)	12
13	DSPB MCBSP DX1 (input/output)	DSPB MCBSP FSR1 (input/output)	14
15	DSPB MCBSP FSX1 (input/output)	3.3V	16
17	DSPA TTL0	DSPA TTL1	18
19	DSPB TTL2	DSPB TTL3	20
21	DSPA GPIO0	DSPA GPIO1	22
23	DSPA GPIO2	DSPA GPIO3	24

SHB generic pin-out

W	Hw		QSH Pin number	QSH Pin number	W	Hw	
CLK	Hw0	CLK	1	2	D0	Hw0	D0
D1		D1	3	4	D2		D2
D3		D3	5	6	D4		D4
D5		D5	7	8	D6		D6
D7		D7	9	10	D8		D8
D9		D9	11	12	D10		D10
D11		D11	13	14	D12		D12
D13		D13	15	16	D14		D14
D15		D15	17	18			USERDEF0
		USERDEF1	19	20			USERDEF2
		USERDEF3	21	22	WEN		WEN
REQ		REQ	23	24	ACK		ACK
			25	26			
			27	28			
			29	30			
			31	32			
			33	34			
			35	36			
	Hw1	CLK	37	38	D16	Hw1	D0
D17		D1	39	40	D18		D2
D19		D3	41	42	D20		D4
D21		D5	43	44	D22		D6
D23		D7	45	46	D24		D8
D25		D9	47	48	D26		D10
D27		D11	49	50	D28		D12
D29		D13	51	52	D30		D14
D31		D15	53	54			USERDEF0
		USERDEF1	55	56			USERDEF2
		USERDEF3	57	58			WEN
		REQ	59	60			ACK

Note: - W is a short for Full Word (i.e. 32-bit Word)

- Hw is a short for Half-word (i.e. 16-bit Word)

SHBA pin-out

Pin	Signal	Signal	Pin
1	SDB0_CLK	SDB0_D0	2
3	SDB0_D1	SDB0_D2	4
5	SDB0_D3	SDB0_D4	6
7	SDB0_D5	SDB0_D6	8
9	SDB0_D7	SDB0_D8	10
11	SDB0_D9	SDB0_D10	12
13	SDB0_D11	SDB0_D12	14
15	SDB0_D13	SDB0_D14	16
17	SDB0_D15	SDB0_U0	18
19	SDB0_U1	-	20
21	-	SDB0_WEN1	22
23	SDB0_REQ1	SDB0_ACK1	24
25	-	-	26
27	-	-	28
29	-	-	30
31	-	-	32
33	-	-	34
35	-	-	36
37	SDB1_CLK	SDB1_D0	38
39	SDB1_D1	SDB1_D2	40
41	SDB1_D3	SDB1_D4	42
43	SDB1_D5	SDB1_D6	44
45	SDB1_D7	SDB1_D8	46
47	SDB1_D9	SDB1_D10	48
49	SDB1_D11	SDB1_D12	50
51	SDB1_D13	SDB1_D14	52
53	SDB1_D15	SDB1_U0	54
55	SDB1_U1	-	56
57	-	SDB1_WEN	58
59	SDB1_REQ	SDB1_ACK	60

▨ Not implemented

SHBB pin-out

Pin	Signal	Signal	Pin
1	SDB2_CLK	SDB2_D0	2
3	SDB2_D1	SDB2_D2	4
5	SDB2_D3	SDB2_D4	6
7	SDB2_D5	SDB2_D6	8
9	SDB2_D7	SDB2_D8	10
11	SDB2_D9	SDB2_D10	12
13	SDB2_D11	SDB2_D12	14
15	SDB2_D13	SDB2_D14	16
17	SDB2_D15	SDB2_U0	18
19	SDB2_U1	-	20
21	-	SDB2_WEN1	22
23	SDB2_REQ1	SDB2_ACK1	24
25	-	-	26
27	-	-	28
29	-	-	30
31	-	-	32
33	-	-	34
35	-	-	36
37	SDB3_CLK	SDB3_D0	38
39	SDB3_D1	SDB3_D2	40
41	SDB3_D3	SDB3_D4	42
43	SDB3_D5	SDB3_D6	44
45	SDB3_D7	SDB3_D8	46
47	SDB3_D9	SDB3_D10	48
49	SDB3_D11	SDB3_D12	50
51	SDB3_D13	SDB3_D14	52
53	SDB3_D15	SDB3_U0	54
55	SDB3_U1	-	56
57	-	SDB3_WEN	58
59	SDB3_REQ	SDB3_ACK	60

▨ Not implemented

This standard is implemented using SAMTEC QSTRIP 0.50mm Hi-speed connectors.

To improve electrical performances, a ground plane is embedded in each QSTRIP connector.

For long distances micro-coax ribbon cable is used to connect 2 QSTRIP connectors.

An SHB interface can be 8, 16 or 32-bit wide.

FPGA PROG Pin Control (JP3 connector)

In	PROG asserted
Out	PROG under control of DSP

FPGA JTAG

The following shows the pin-outs for JP4 (FPGA) JTAG connector:

1	2	3	4	5	6
V33	TCK	GND	TMS	TDO	TDI

The JTAG chain includes the FPGA and the CPLD (XC9636XL).

Data Sheets (Hyperlinks)

1. Sundance help file

2. TMS320C6201/C6701 Peripherals Reference Guide (literature number SPRU190)
   It describes the common peripherals available on the TMS320C6201/C6701 digital signal processors. This book includes information on the internal data and program memories, the external memory interface (EMIF), the host port, multichannel-buffered serial ports, direct memory access (DMA), clocking and phase-locked loop (PLL), and the power-down modes.

3. SMT6400 help file and SMT6500 help file

4. TIM-40 MODULE SPECIFICATION Including TMS320C44 Addendum

5. SDB Technical Specification

6. SHB Technical Specification

7. TMS320C4x User's Guide (literature number SPRU063)
   It describes the C4x 32-bit floating-point processor, developed for digital signal processing as well as parallel processing applications. Covered are its architecture, internal register structure, instruction set, pipeline, specifications, and operation of its six DMA channels and six communication ports. Software and hardware applications are included.

8. Xilinx Virtex-II data sheet

9. Texas Instruments TMS320C6211B data sheet

10. Texas Instruments TMS320C6711 data sheet

11. Texas Instruments TMS320C6713 data sheet

Index