Manual Update Sheet

DATE: June 1, 1998

Document Being Updated: ***TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals***

Literature Number Being Updated: SPRU131D

Manual Included in a Kit: No

This Manual Update Sheet (SPRZ117A) ships with the *TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals*, literature number SPRU131D.

Updates within paragraphs appear in a **bold typeface**.

Page: **Change or Add:**

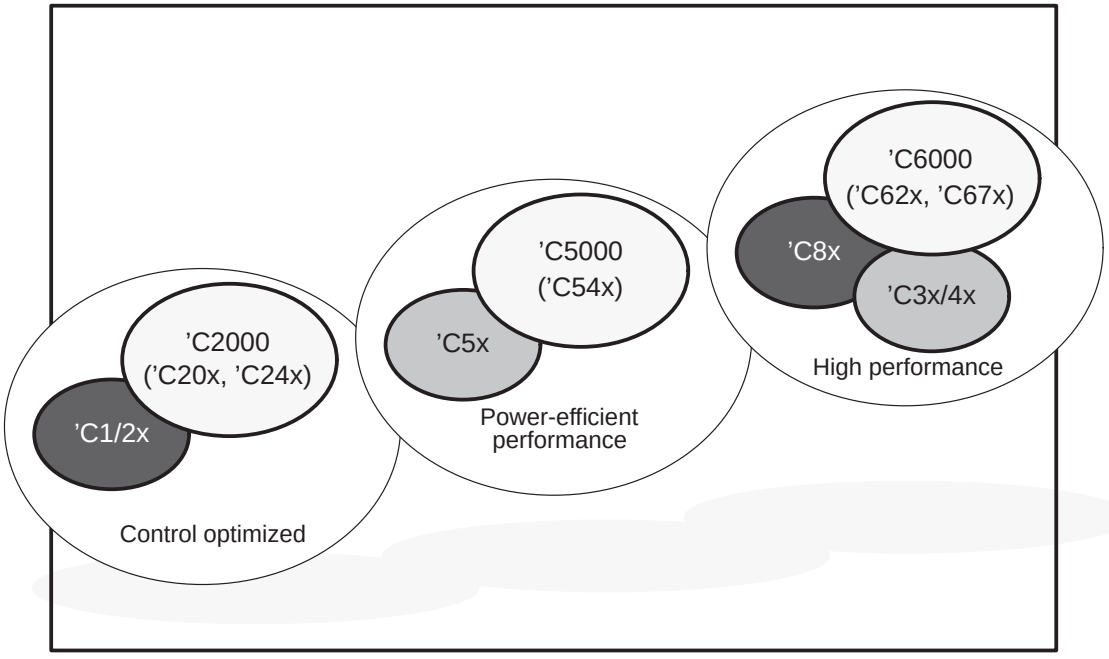
vi Add the following sentence to the end of the first paragraph:

> Many of these documents are located on the internet at http://www.ti.com. Click DSPS Solutions, then click DSP Literature.

vii Add the following document reference:

> ***TMS320C548/C549 Bootloader Technical Reference*** (literature number SPRU288) describes the process the bootloader uses to transfer user code from an external source to the program memory at power up. (Presently available only on the internet.)

1–3 Change Figure 1–1 , *Evolution of the TMS320 Family*, to the following updated graphic:



1–6 Change the first second-level bullet under Memory to:

- 192K words × 16-bit addressable memory space (64K-words program, 64K-words data, and 64K-words I/O), with extended program memory (8M words) in the '548 **and '549**.

Page: **Change or Add:**

1–6 Add the following ’549 information to the configuration table at the bottom of the page:

Device	Program ROM	Program/Data ROM	DARAM†	SARAM‡
’549	16	16	8	24

1–7 Change the first sentence of the note below the option table to:

†The **’541B**, ’545**A**, ’546**A**, ’548, and **’549** are designated as LP-type devices.

1–8 Add the following ’549 information to the ports table at the top of the page:

Device	Host Port Interface	Serial Ports		
		Synchronous	Buffered	Time-Division Multiplexed
’549	1	0	2	1

Delete the ’548 3-V power supply and 12.5-ns/10-ns speed designation in the table that lists power supply, speed, and package type information. Add ’541B, ’545A, ’546A, and ’549 information as noted in the revised table below. Delete the note below the table regarding ’548 speed.

Device	Power Supply	Speed	Package
’541	5 V	25 ns	100-pin TQFP
	3 V / 3.3 V	25 ns/20 ns	100-pin TQFP
’541B	**3 V / 3.3 V**	**15 ns**	**100-pin TQFP**
’542	5 V	25 ns	144-pin TQFP
	3 V / 3.3 V	25 ns/20 ns	128-pin/144-pin TQFP
’543	3 V / 3.3 V	25 ns/20 ns	100-pin TQFP
’545	3 V / 3.3 V	25 ns/20 ns	128-pin TQFP
’545A	**3 V / 3.3 V**	**15 ns**	**128-pin TQFP**
’546	3 V / 3.3 V	25 ns/20 ns	100-pin TQFP
’546A	**3 V / 3.3 V**	**15 ns**	**100-pin TQFP**
’548	3.3 V	20 ns/15 ns	144-pin TQFP
’549	**3.3 V**	**15 ns/12.5 ns**	**144-pin TQFP/144-pin Micro Star™ BGA**
’VC549	**3.3 V (2.5 V core)**	**10 ns**	**144-pin TQFP/144-pin Micro Star™ BGA**

Page: **Change or Add:**

2–5 Add the following ’549 information to Table 2–2, *Program and Data Memory on the TMS320C54x Devices*:

Memory Type	’549
ROM:	**16K**
Program	**16K**
Program/data	**16K**
DARAM†	**8K**
SARAM†	**24K**

Change the second paragraph in section 2.2.1, *On-Chip ROM*, to the following:

> On devices with a small amount of ROM (2K words), the ROM contains a bootloader that is useful for booting to faster on-chip or external RAM. For bootloading details on all ’54x devices except the ’548 and ’549, see *TMS320C54x DSP Reference Set, Volume 4: Applications Guide.* For bootloading details on the ’548 and ’549, see *TMS320C548/549 Bootloader Technical Reference.*

2–6 Change the second sentence in section 2.2.2, *On-Chip Dual-Access RAM (DARAM)*, to:

> Because each DARAM block can be accessed twice per machine cycle, the central processing unit (CPU) **and peripherals such as the buffered serial port (BSP) and host port interface (HPI)** can read from and write to a DARAM memory address in the same cycle.

2–11 Change the first sentence in section 2.7.2, *Software-Programmable Wait-State Generator*, to:

> The software-programmable wait-state generator extends external bus cycles up to seven machine cycles **(14 machine cycles in the ’549)** to interface with slower off-chip memory and I/O devices.

2–12 Add the following ’549 information to Table 2–3, *Host Port Interfaces on the TMS320C54x Devices*:

On-Chip Peripheral	’549
Host port interface	**1**

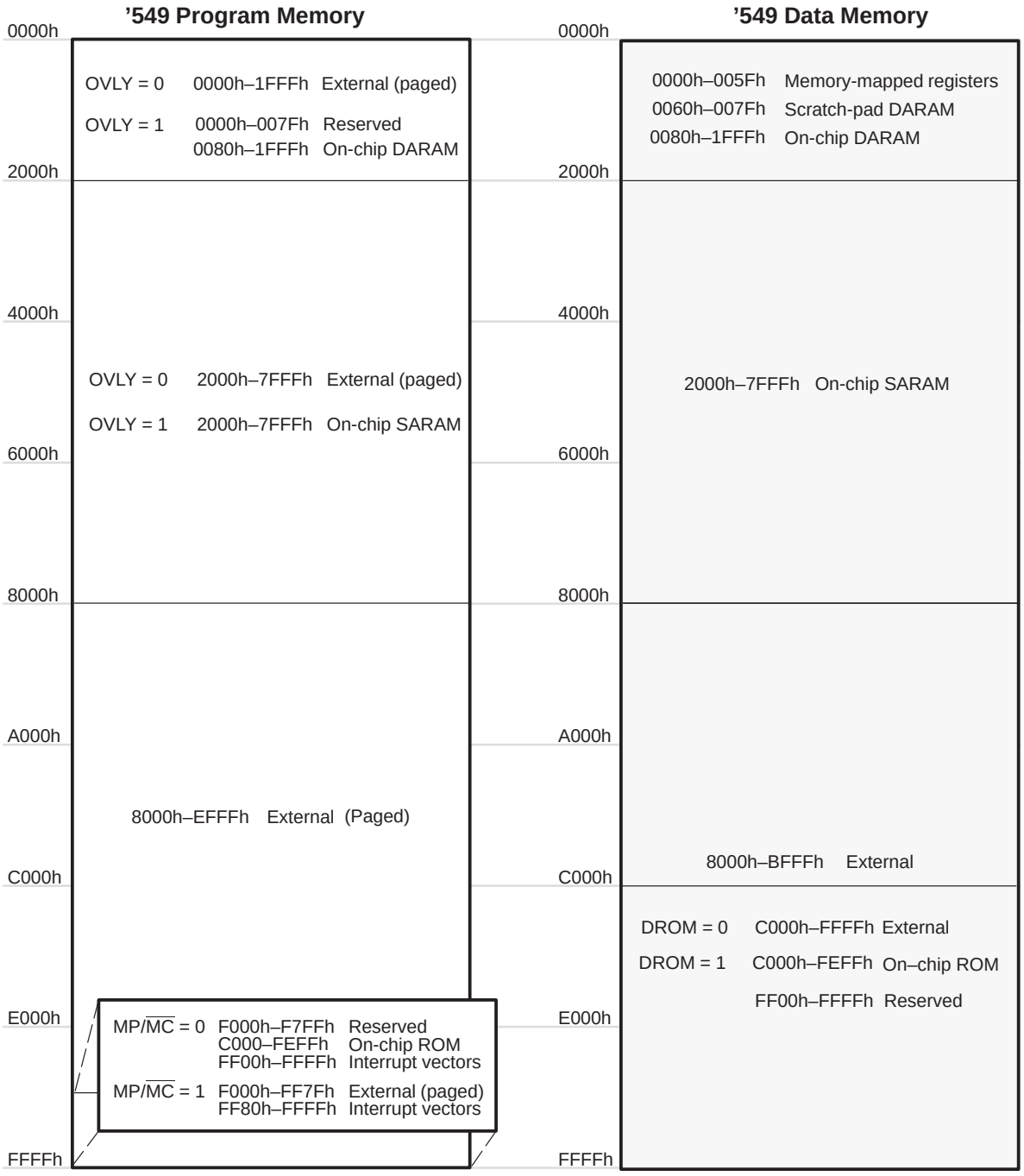
2–13 Add the following ’549 information to Table 2–4, *Serial Port Interfaces on the TMS320C54x Devices*:

Serial Ports	’549
Synchronous	**0**
Buffered	**2**
TDM	**1**

Page:	Change or Add:
2–14	Change the first sentence in section 2.9, *External Bus Interface*, to: The ’54x can address up to 64K words of data memory, 64K words of program memory (8M words in the ’548 and **’549**), and up to 64K words of 16-bit parallel I/O ports.
3–1	Change the fourth sentence in the first paragraph to: In some devices, such as the ’548 and **’549**, the memory structure has been modified through overlay and paging schemes.
3–2	Change the third sentence in the first paragraph in section 3.1, *Memory Space*, to: Together, these three spaces provide a total address range of 192K words (except in the ’548 and ’**549**).

Page: **Change or Add:**

3–6 Add the following graphic: *Memory Maps for the TMS320C549,* after Figure 3–4.



Page: **Change or Add:**

3–7 Change the heading of section 3.1.1 to:

Extended Program Memory (Available on TMS320C548**/549**)

Change the first paragraph to:

The ’548 **and ’549** use a paged extended memory scheme in program space to allow access of up to 8192K words of program memory. In order to implement this scheme, the ’548 **and ’549** include several additional features:

3–7 Change the second paragraph to:

Program memory in the ’548 **and ’549** is organized into 128 pages that are each 64K words in length, as shown in Figure 3–5.

3–8 Change the sentence above the first bulleted list to:

To facilitate page switching through software, the ’548 **and ’549** have six special instructions that affect the XPC:

Change the sentence above the second bulleted list to:

In addition to these new instructions, two ’54x instructions are extended in the ’548 **and ’549** to use 23 bits.

3–9 Change the first sentence of the first paragraph to:

The external program memory on the ’54x devices (except on the ’548 **and ’549**) addresses up to 64K 16-bit words.

Add the following ’549 information to Table 3–1, *On-Chip Program Memory Available on the TMS320C54x Devices*:

Device	ROM (MP/$\overline{MC}$ = 0)	DARAM (OVLY = 1)	SARAM (OVLY = 1)
’549	16K	8K	24K

3–10 Add the following ’549 information to Figure 3–7, *On-Chip ROM Block Organization*:

’549
C000–CFFF
D000–DFFF
E000–EFFF
F000–FFFF

Page: **Change or Add:**

3–11 Change the first paragraph in section 3.2.4, *On-Chip ROM Code Contents and Mapping*, to:

> ’54x devices have either large (**16K**, 24K, 28K, or 48K words) on-chip ROM or 2K words of on-chip ROM. A large on-chip ROM can be programmed with your code, while the content of a 2K-word on-chip ROM is defined by Texas Instruments. On the ’54x devices **with on-chip bootloader** ROM, the 2K words (at F800h to FFFFh) **may** contain **one or more of the following, depending on the specific device:**

3–12 Change the label above the second memory map to:

> ’542/543/548***/549***

3–13 Add the following ’549 information to Table 3–2, *On-Chip Data Memory Available on the TMS320C54x Devices:*

Device	Program/Data ROM (DROM = 1)	DARAM	SARAM
’549	16K	8K	24K

3–15 Change the label above the RAM block on the far right to:

> ’548***/549***

3–16 Add 03FFh as the last address at DARAM block DP = 7.

3–18 Change the XPC description at address 1E in Table 3–3, *CPU Memory-Mapped Registers,* to:

Address	Name	Description
1E	XPC	Program counter extension register (’548 **and ’549**)

3–20 Change the heading of section 3.3.4.11 to:

> *Program Counter Extension Register (XPC, Available on ’548**/549**)*

4–3 Delete the following from the third sentence of the Bit 9 function description:

> **. . . .** or when an expression that writes to B does not overflow.

5–6 Change the second sentence in the note at the bottom of the page to:

> However, because the ’548 **and ’549** have 23 address lines, the program-memory location in a ’548 **or ’549** device is specified by the lower 23 bits of accumulator A.

6–2 Change the second sentence of the first paragraph to:

> The ’54x can address a total of 64K words of program memory using the program address bus (PAB); the ’548 **and ’549** have an additional seven address lines to provide external access to 128 64K-word pages.

Page:	Change or Add:
6–2	Change the sentence below the fifth bulleted item to: One additional register is used in the ’548 **and ’549** to address extended memory:
6–5	Change the first sentence in the first paragraph to: XPC is a 7-bit register that selects the current page of program memory for the ’548 **and ’549**. For more information about extended program memory in the ’548 **and ’549**, see section 3.1.1, *Extended Program Memory*, on page 3–8.
6–8	Change the heading of section 6.3.3 to: Far Branches (Available on TMS320C548**/549**) Change the first sentence of section 6.3.3 to: To allow branches to extended memory, the ’548 **and ’549** include two far branch instructions:
6–8	Change the first sentence below the bulleted list to: Table 6–5 shows the far branch instructions in the ’548 **and ’549** (both nondelayed and delayed) and the number of cycles needed to execute these instructions.
6–11	Change the heading of section 6.4.3 to: Far Calls (Available on TMS320C548**/549)** Change the first sentence in section 6.4.3 to: To allow calls to extended memory, the ’548 **and ’549** include two far call instructions: Change the first sentence of the paragraph below the bulleted list to: Table 6–8 shows the far call instructions in the ’548 **and ’549** (nondelayed and delayed) and the number of cycles needed to execute these instructions.
6–12	Change the second paragraph in section 6.5, *Returns*, to: The ’548 **and ’549** offer an additional return instruction: an unconditional far return, both nondelayed and delayed.
6–14	Change the heading of section 6.5.3 to: Far Returns (Available on TMS320C548**/549)** Change the first sentence in section 6.5.3 to: To allow returns from extended memory, the ’548 **and ’549** include two far return instructions:

Page: **Change or Add:**

6–15 Change the first sentence to:

Table 6–11 shows the far return instructions in the ’548 **and ’549** (nondelayed and delayed) and the number of cycles needed to execute these instructions.

6–25 Change the fifth bulleted item to:

- XPC is cleared (’548 **and ’549**)

6–28 Add the following register to Figure 6–2, *Interrupt Flag Register (IFR) Diagram*:

(g) ’549 IFR

15–14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resvd	BMINT1	BMINT0	BXINT1	BRINT1	HPINT	INT3	TXINT	TRINT	BXINT0	BRINT0	TINT	INT2	INT1	INT0

6–29 Change the first paragraph to:

Figure 6–3 shows how the ’54x uses a memory-mapped IMR for masking external and internal interrupts. If INTM = 0 in ST1, a 1 in any IMR bit enables the corresponding interrupt. Neither $\overline{\text{NMI}}$ nor $\overline{\text{RS}}$ is included in the IMR, because IMR has no effect on these interrupts. You can read or write to the IMR.

Add the following register to Figure 6–3, *Interrupt Mask Register (IMR) Diagram*:

(g) ’549 IMR

15–14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resvd	BMINT1	BMINT0	BXINT1	BRINT1	HPINT	INT3	TXINT	TRINT	BXINT0	BRINT0	TINT	INT2	INT1	INT0

6–32 Add the following parenthetical sentence below item number 1 in section 6.10.5, *Phase 3: Execute Interrupt Service Routine (ISR)*:

(The program counter extension register, XPC, does not get pushed to the top of the stack; that is, it does not get saved on the stack.)

Page: **Change or Add:**

6–43 Add the following table after Table 6–23: *TMS320C549 Interrupt Locations and Priorities*

TRAP/INTR Number (K)	Priority	Name	Location	Function
0	1	$\overline{\text{RS}}$/SINTR	0	Reset (hardware and software reset)
1	2	$\overline{\text{NMI}}$/SINT16	4	Nonmaskable interrupt
2	–	SINT17	8	Software interrupt #17
3	–	SINT18	C	Software interrupt #18
4	–	SINT19	10	Software interrupt #19
5	–	SINT20	14	Software interrupt #20
6	–	SINT21	18	Software interrupt #21
7	–	SINT22	1C	Software interrupt #22
8	–	SINT23	20	Software interrupt #23
9	–	SINT24	24	Software interrupt #24
10	–	SINT25	28	Software interrupt #25
11	–	SINT26	2C	Software interrupt #26
12	–	SINT27	30	Software interrupt #27
13	–	SINT28	34	Software interrupt #28
14	–	SINT29	38	Software interrupt #29
15	–	SINT30	3C	Software interrupt #30
16	3	$\overline{\text{INT0}}$/SINT0	40	External user interrupt #0
17	4	$\overline{\text{INT1}}$/SINT1	44	External user interrupt #1
18	5	$\overline{\text{INT2}}$/SINT2	48	External user interrupt #2
19	6	TINT/SINT3	4C	Internal timer interrupt
20	7	BRINT0/SINT4	50	Buffered serial port 0 receive interrupt
21	8	BXINT0/SINT5	54	Buffered serial port 0 transmit interrupt
22	9	TRINT/SINT6	58	TDM serial port receive interrupt
23	10	TXINT/SINT7	5C	TDM serial port transmit interrupt
24	11	$\overline{\text{INT3}}$/SINT8	60	External user interrupt #3
25	12	$\overline{\text{HINT}}$/SINT9	64	HPI interrupt
26	13	BRINT1/SINT10	68	Buffered serial port 1 receive interrupt
27	14	BXINT1/SINT11	6C	Buffered serial port 1 transmit interrupt
28	15	BMINT0/SINT12	70	BSP #0 misalignment detection interrupt (’549 only)
29	16	BMINT1/SINT13	74	BSP #1 misalignment detection interrupt (’549 only)
30–31	–		78–7F	Reserved

Page: **Change or Add:**

7–28 Add the following ’549 information to Table 7–1, *DARAM Blocks*:

Device	Block Size	Number of Blocks
’549	2K words	4

8–1 Change the fourth bulleted item to:

- ❑ Host port interface (’542, ’545, ’548, **and ’549**)

Change the sixth bulleted item to:

- ❑ Buffered serial port (’542, ’543, ’545, ’546, ’548, **and ’549**)

Change the seventh bulleted item to:

- ❑ Time-division multiplexed (TDM) serial port (’542, ’543, ’548, **and ’549**)

8–3 Change the label above the memory-mapped register block on the far right of Figure 8–1, *TMS320C54x Peripheral Memory-Mapped Registers*, to:

’548**/549**

8–4 Change the title of Table 8–1 to include the ’541B device and make the following changes to the table:

*TMS320C541**/541B** Peripheral Memory-Mapped Registers*

Address	Name	Description
33–**57**	–	Reserved
58	**CLKMD**	**Clock mode register (’541B only)**
59–57	**–**	**Reserved**

8–7 Change the title of Table 8–4 to include the ’545A device and add the following to address 58:

*TMS320C545**/545A** Peripheral Memory-Mapped Registers*

Address	Name	Description
58	CLKMD	Clock mode register **(’545A only)**

8–8 Change the title of Table 8–5 to include the ’546A device and add the following to address 58:

*TMS320C546**/546A** Peripheral Memory-Mapped Registers*

Address	Name	Description
58	CLKMD	Clock mode register **(’546A only)**

Page: **Change or Add:**

8–9 Add the following table after Table 8–6: *TMS320C549 Peripheral-Memory Mapped Registers*

Address	Name	Description
20	BDRR0	Buffered serial port 0 data receive register
21	BDXR0	Buffered serial port 0 data transmit register
22	BSPC0	Buffered serial port 0 control register
23	BSPCE0	Buffered serial port 0 control extension register
24	TIM	Timer count register
25	PRD	Timer period register
26	TCR	Timer control register
27	–	Reserved
28	SWWSR	External interface software wait-state register
29	BSCR	External interface bank-switching control register
2A	–	Reserved
2B	XSWR	Extended software wait-state register
2C	HPIC	Host port interface control register
2D-2F	–	Reserved
30	TRCV	TDM serial port data receive register
31	TDXR	TDM serial port data transmit register
32	TSPC	TDM serial port control register
33	TCSR	TDM serial port channel select register
34	TRTA	TDM serial port receive transmit register
35	TRAD	TDM serial port receive address register
36-37	–	Reserved
38	AXR0	ABU 0 transmit address register
39	BKX0	ABU 0 transmit buffer-size register
3A	ARR0	ABU 0 receive address register
3B	BKR0	ABU 0 receive buffer-size register
3C	AXR1	ABU 1 transmit address register
3D	BKX1	ABU 1 transmit buffer-size register
3E	ARR1	ABU 1 receive address register
3F	BKR1	ABU 1 receive buffer-size register
40	BDRR1	Buffered serial port 1 data receive register
41	BDXR1	Buffered serial port 1 data transmit register
42	BSPC1	Buffered serial port 1 control register
43	BSPCE1	Buffered serial port 1 control extension register
44-57	–	Reserved
58	CLKMD	Clock-mode register
59-5F	–	Reserved

Page: **Change or Add:**

8–16 Change the fifth sentence in the paragraph below the bulleted list to:

The ’545A, ’546A, ’548 **and ’549** devices use a software-programmable PLL and are referred to in this section as *LP* devices.

8–17 Change the heading of section 8.4.2 to:

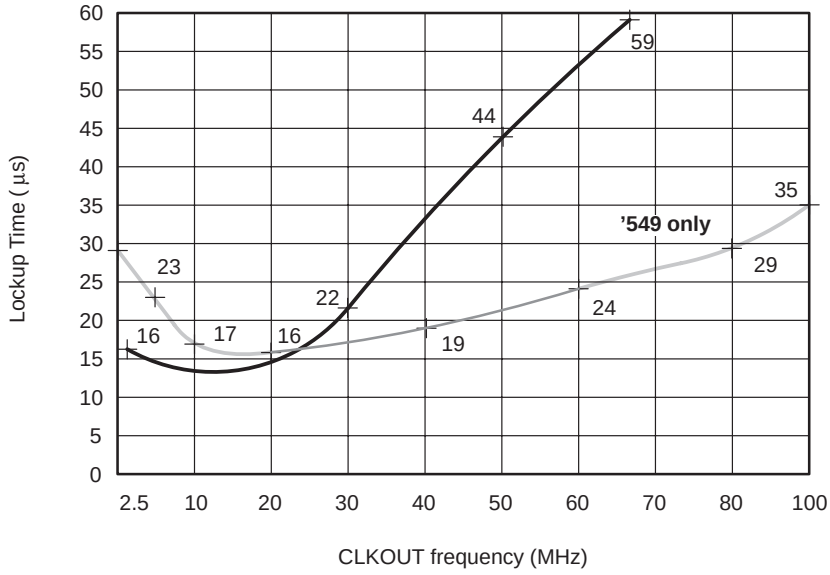
Software-Programmable PLL (Available on TMS320C**541B**/545A/546A/548**/549**)

8–18 Add the following to Table 8–10, *Clock Mode Setting at Reset*:

CLKMD1	CLKMD2	CLKMD3	CLKMD Reset Value	Clock Mode
1	1	1	7000h	Divide-by-2 with internal source†

† **Reserved on ’549**

8–21 Replace Figure 8–7, *PLL Lockup Time Versus CLKOUT Frequency,* with the following revised graphic:



8–26 Change the first sentence of the first paragraph to:

The host port interface (HPI) is available on the ’542, ’545, ’548, **and ’549** devices.

Page: **Change or Add:**

8–27 Change the third sentence in the second paragraph to:

> In this mode, asynchronous host accesses are resynchronized internally and, in the case of a conflict between a ’54x and a host cycle **(where both accesses are reads or writes)**, the host has access priority and the ’54x waits one cycle.

9–2 Add the following ’549 information to Table 9–1, *Serial Ports on the TMS320C54x Devices*:

Device	Standard Synchronous Serial Ports	Buffered Serial Ports	Time-Division Multiplexed Serial Ports
’549	0	2	1

9–3 Change the third and fourth sentences in the second paragraph to:

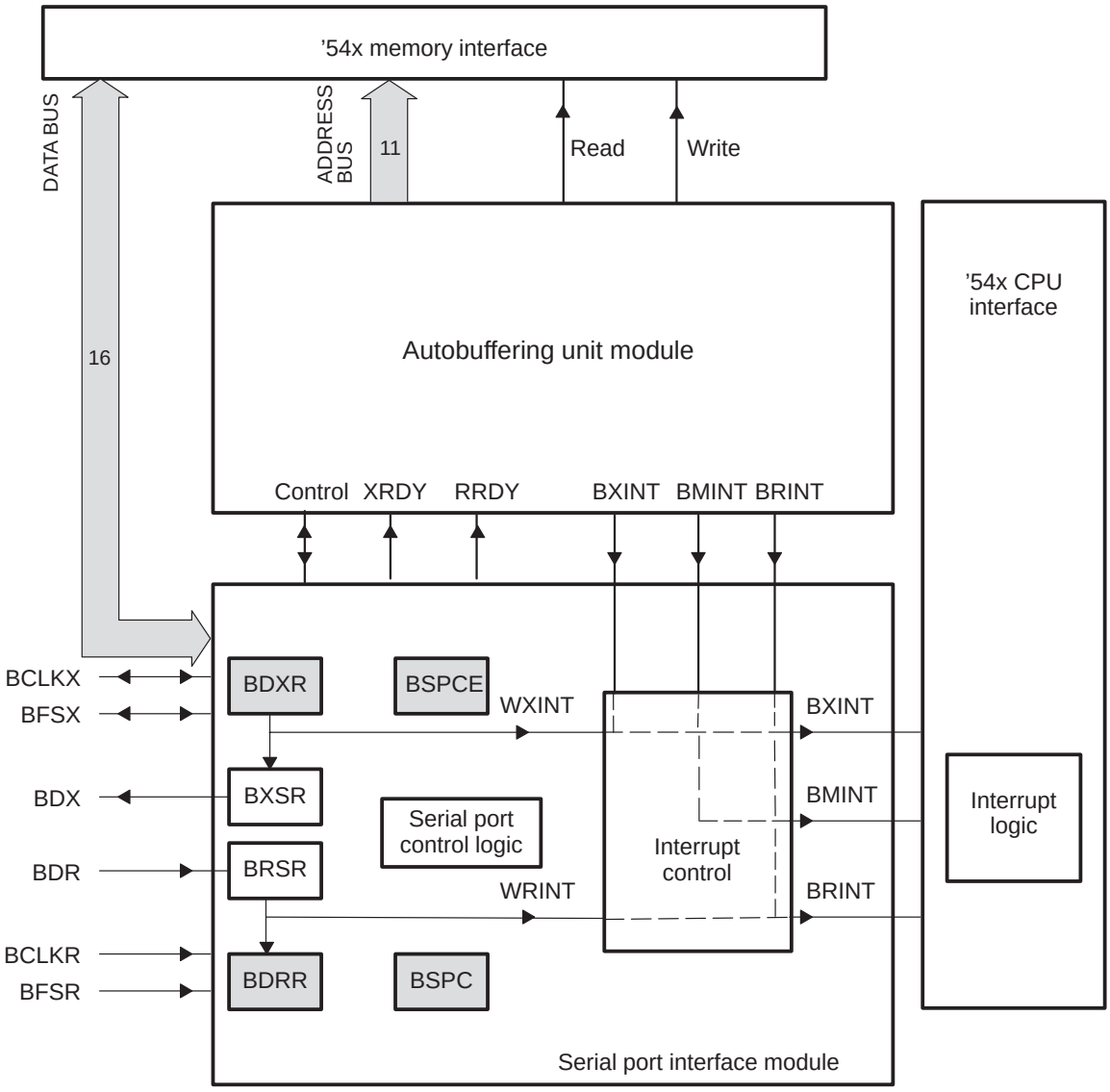
> The TDM serial port interface is implemented on the ’542, ’543, ’548, **and ’549** devices. The ’542, ’543, ’545, ’546, ’548, **and ’549** devices include a buffered serial port (BSP) that implements an automatic buffering feature, which greatly reduces CPU overhead required in handling serial data transfers.

Change the second sentence of the fourth paragraph to:

> Therefore, when using the ’542, ’543, ’545, ’546, ’548, **and ’549** devices, you should consult section 9.3.

Page: **Change or Add:**

9–33 Add a signal for the buffer misalignment interrupt (BMINT) to Figure 9–21, *BSP Block Diagram,* as in the following graphic:

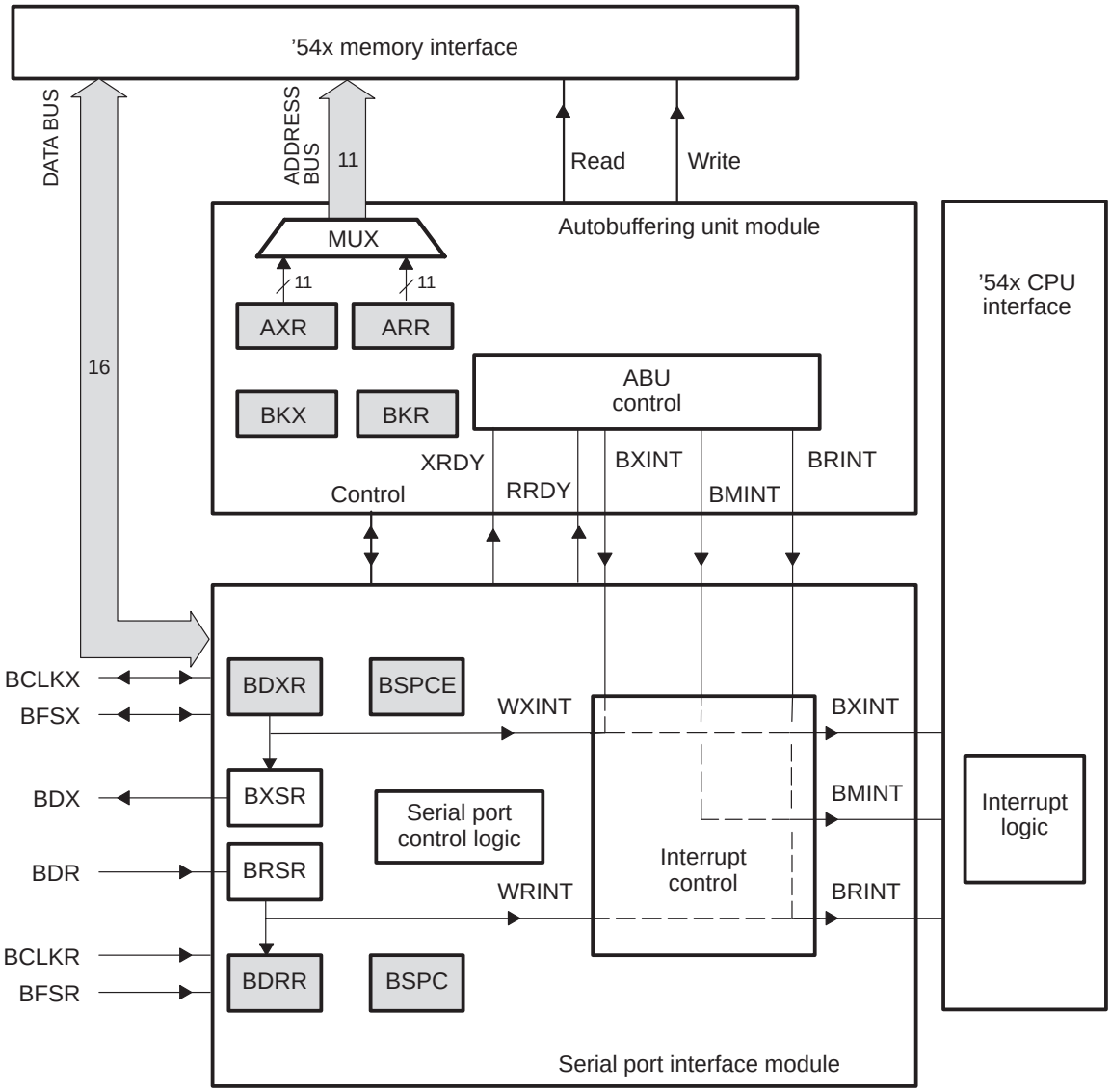


9–40 Change the last paragraph to:

The ABU also implements CPU interrupts when transmit and receive buffers have been halfway or entirely filled or emptied. These interrupts take the place of the transmit and receive interrupts in standard mode operation (the receive interrupt is the CPU). They are not generated in autobuffering mode. This mechanism features an autodisabling capability which can be used to automatically terminate autobuffering when either the half-of-buffer or bottom-of-buffer boundary is crossed. These features are described in detail later in this section.

Page: **Change or Add:**

9–41 Add a signal for the buffer misalignment interrupt (BMINT) to Figure 9–24, *ABU Block Diagram,* as in the following graphic:

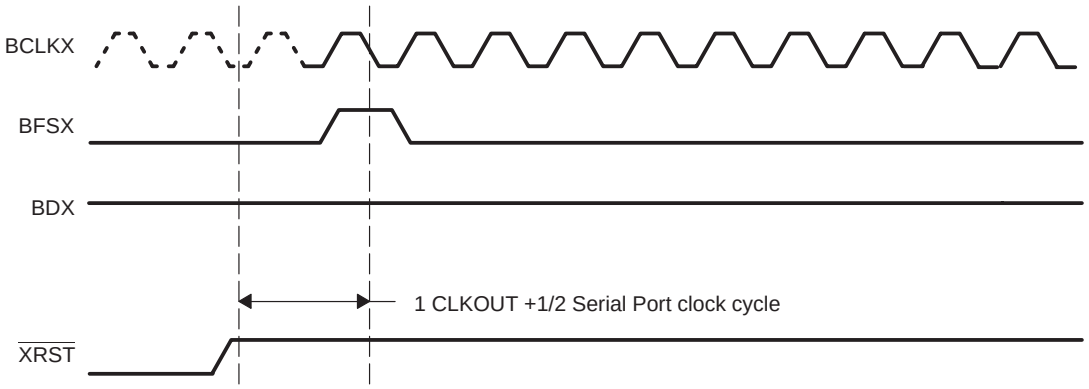


9–49 Change the last sentence in the last paragraph to:

In this example, if the BFSX pulse occurs during the first two BCLKXs **after the transmit section is taken out of reset**, the transmit frame is ignored and BDX is placed in the high-impedance state.

Page: **Change or Add:**

9–49 Remove the graphical representation of data from the BDX line in figure 9–28, *Standard Mode BSP Initialization Timing,* as follows:



9–51 Change the first sentence of the fifth paragraph to:

In Example 9–3 and Example 9–4, the transmit and receive interrupts used are those that the BSP occupies on the ’542, ’543, ’545, ’546, ’548, **and ’549**, the devices that include the BSP.

9–53 Add the following BMINT section between sections 9.3.3.2, *Initialization Examples*, and section 9.3.4, *BSP Operation in Power-Down Mode*:

Buffer Misalignment Interrupt (BMINT) – ’549 Only

BMINT is generated when a frame sync occurs and the ABU transmit or receive buffer pointer is not at the top of the the buffer address. This is useful for detecting several potential error conditions on the serial interface, including extraneous and missed clocks and frame sync pulses. A BMINT interrupt, therefore, indicates that one or more words may have been lost on the serial interface.

BMINT is useful for detecting buffer misalignment only when the buffer pointer(s) are initially loaded with the top of the buffer address and a frame of data contains the same number of words as the buffer length. These are the only conditions under which a frame sync occurring at a buffer address other than the top of the buffer constitute an error condition. In cases where these conditions are met, a frame sync always occurs when the buffer pointer is at the top of the buffer address (if the interface is functioning properly).

If BMINT is enabled under conditions other than those described, interrupts can be generated under circumstances other than actual buffer misalignment. For this reason, BMINT should generally be masked in the IMR so that the processor ignores this interrupt.

BMINT is available when the device is operating in the auto-buffering mode with continuous transfers, the FIG bit cleared to 0, and with external serial clocks or frames.

Page: **Change or Add:**

10–2 Change the heading row in Table 10–1, *Key External Interface Signals*, to include the ’549 device as follows:

Signal Name	’541’542, ’543, ’545, ’546	’548, ’549	Description
A0–A15	15–0	22–0	Address bus

10–5 Change the first sentence of the first paragraph in section 10.3.1, *Wait State Generator, to:*

The software-programmable wait-state generator can extend external bus cycles by up to seven machine cycles **(14 machine cycles on ’549 devices)**, providing a convenient means to interface the ’54x to slower external devices.

Change the last sentence of the third paragraph in section 10.3.1, *Wait State Generator,* to:

These fields are shown in Figure 10–2 and described in Table 10–2. The ’548 **and ’549** are described in Table 10–3.

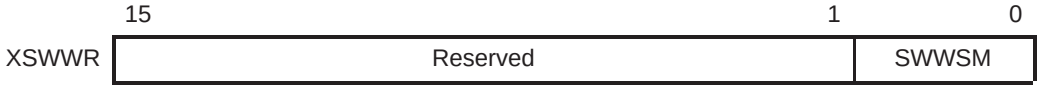
Change the note below Figure 10–2, *Software Wait-State Register (SWWSR) Diagram*, to:

†XPA bit on ’548 **and ’549** only

Add the following sentence after Figure 10–2, *Software Wait-State Register (SWWSR) Diagram*:

The ’549 has an extra bit (software wait-state multiplier, SWWSM) that resides in XSWWR, which is memory-mapped to address 002Bh in data space.

Add the following XSWWR register after the SWWSR register:



Add the following sentence below the XSWWR register:

When SWWSM is set to 1, the wait states are multiplied by two, extending the number of wait states from 7 to 14.

10–6 Change the bit 15 function in Table 10–2, *Software Wait-State Register (SWWSR) Bit Summary,* to:

Bit	Name	Reset Value	Function
15	Reserved	0	Reserved. In the ’548 **and ’549**, this bit changes the operation of the program fields (see Table 10–3).

Page: **Change or Add:**

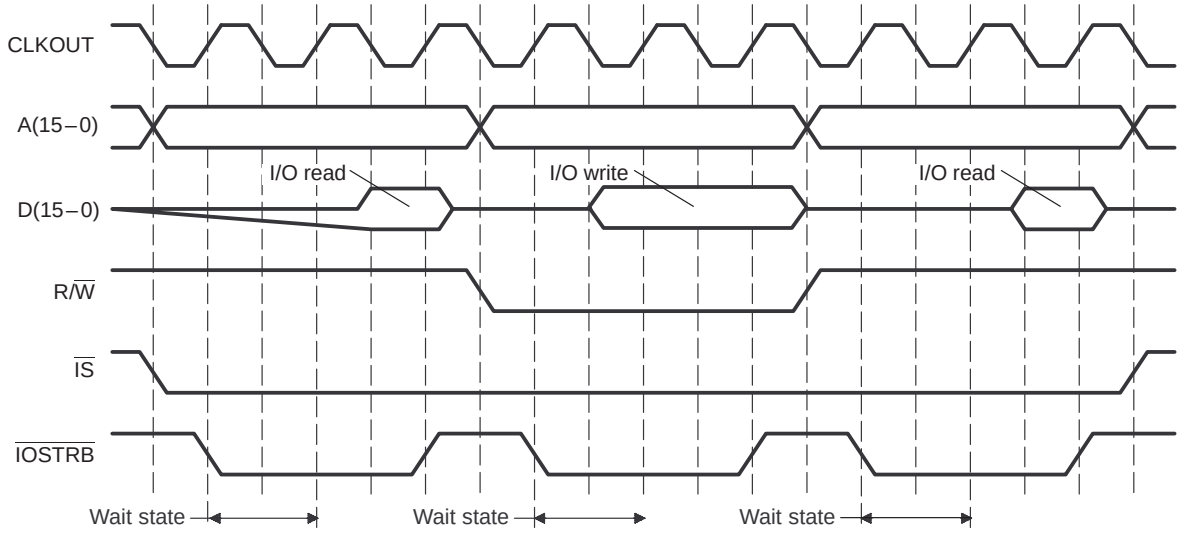
10–6 Change the title of Table 10–3 to:

*TMS320C548**/549** Software Wait-State Register (SWWSR) Bit Summary*

10–10 Change the third bulleted item to:

- ❑ A program-memory read followed by another program-memory read from a different page (with the ’548 **and ’549**)

10–17 Change the I/O write at D(15–0) in Figure 10–11, *Parallel I/O Operation for Read-Write-Read (I/O-Space Wait States)*, as follows:



A-1 Add the following term and definition to the table:

BMINT1, BMINT0	**Buffer misalignment interrupt**

A-4 Add the following register to Figure A–5, *Interrupt Flag Register (IFR) Diagram*:

(g) ’549 IFR

15–14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resvd	BMINT1	BMINT0	BXINT1	BRINT1	HPINT	INT3	TXINT	TRINT	BXINT0	BRINT0	TINT	INT2	INT1	INT0

A-5 Add the following register to Figure A–6, *Interrupt Mask Register (IMR) Diagram*:

(g) ’549 IMR

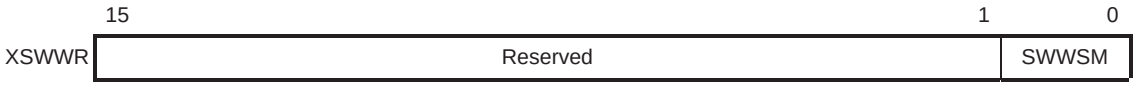
15–14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resvd	BMINT1	BMINT0	BXINT1	BRINT1	HPINT	INT3	TXINT	TRINT	BXINT0	BRINT0	TINT	INT2	INT1	INT0

A-6 Change the note at the bottom of Figure A–9, *Software Wait-State Register (SWWSR),* to:

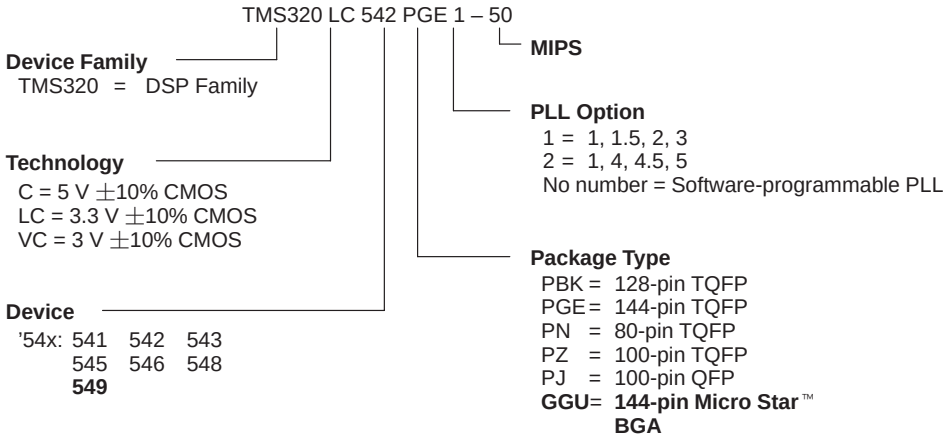
†XPA bit on ’548 **and ’549** only

Page: **Change or Add:**

A-6 Add the following *Extended Software Wait-State Register (XSWWR)* below the SWWSR register in Figure A–9, *Software Wait-State Register (SWWSR)*:



C-6 Add ’549 to the list of devices and MicroStar™ BGA to the list of package types in Figure C–1, *TMS320C54x Device Nomenclature,* as follows:



E-4 Add the following glossary definition below **block-repeat start address register (RSA)**:

BMINT: See *buffer misalignment interrupt*.

E-5 Add the following glossary definition below **BSPCE, BSPCE0, BSPCE1**:

buffer misalignment interrupt (BMINT): A ’549 feature that detects potential error conditions and indicates lost words on a serial port interface.

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 acknowledgement, 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.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

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.