- Advanced Multibus Architecture With Three Separate 16-Bit Data Memory Buses and One Program Memory Bus
- 40-Bit Arithmetic Logic Unit (ALU) Including a 40-Bit Barrel Shifter and Two Independent 40-Bit Accumulators
- 17- × 17-Bit Parallel Multiplier Coupled to a 40-Bit Dedicated Adder for Non-Pipelined Single-Cycle Multiply/Accumulate (MAC) Operation
- Compare, Select, and Store Unit (CSSU) for the Add/Compare Selection 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 (ARAUs)
- Data Bus With a Bus Holder Feature
- Address Bus With a Bus Holder Feature (’548 and ’549 Only)
- Extended Addressing Mode for 8M × 16-Bit Maximum Addressable External Program Space (’548 and ’549 Only)
- 192K × 16-Bit Maximum Addressable Memory Space (64K Words Program, 64K Words Data, and 64K Words I/O)
- On-Chip ROM with Some Configurable to Program/Data Memory
- Dual-Access On-Chip RAM
- Single-Access On-Chip RAM (’548/’549)
- Single-Instruction Repeat and Block-Repeat Operations for Program Code
- Block-Memory-Move Instructions for Better Program and Data Management
- Instructions With a 32-Bit Long Word Operand
- Instructions With Two- or Three-Operand Reads
- Arithmetic Instructions With Parallel Store and Parallel Load
- Conditional Store Instructions
- Fast Return From Interrupt
- On-Chip Peripherals
  - Software-Programmable Wait-State Generator and Programmable Bank Switching
  - On-Chip Phase-Locked Loop (PLL) Clock Generator With Internal Oscillator or External Clock Source
  - Full-Duplex Serial Port to Support 8- or 16-Bit Transfers (’541, ’LC545, and ’LC546 Only)
  - Time-Division Multiplexed (TDM) Serial Port (’542, ’543, ’548, and ’549 Only)
  - Buffered Serial Port (BSP) (’542, ’543, ’LC545, ’LC546, ’548, and ’549 Only)
  - 8-Bit Parallel Host Port Interface (HPI) (’542, ’LC545, ’548, and ’549)
  - One 16-Bit Timer
  - External-Input/Output (XIO) Off Control to Disable the External Data Bus, Address Bus and Control Signals
- Power Consumption Control With IDLE1, IDLE2, and IDLE3 Instructions With Power-Down Modes
- CLKOUT Off Control to Disable CLKOUT
- On-Chip Scan-Based Emulation Logic, IEEE Std 1149.1† (JTAG) Boundary Scan Logic
- 25-ns Single-Cycle Fixed-Point Instruction Execution Time [40 MIPS] for 5-V Power Supply (’C541 and ’C542 Only)
- 20-ns and 25-ns Single-Cycle Fixed-Point Instruction Execution Time (50 MIPS and 40 MIPS) for 3.3-V Power Supply (’LC54x)
- 15-ns Single-Cycle Fixed-Point Instruction Execution Time (66 MIPS) for 3.3-V Power Supply (’LC54xA, ’548, ’LC549)
- 12.5-ns Single-Cycle Fixed-Point Instruction Execution Time (80 MIPS) for 3.3-V Power Supply (’LC549)
- 10-ns Single-Cycle Fixed-Point Instruction Execution Time (100 MIPS) for 3.3-V Power Supply (2.5-V Core) (’VC549)

ADVANCE INFORMATION

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

† IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

ADVANCE INFORMATION

description

The TMS320C54x, TMS320LC54x, and TMS320VC54x fixed-point, digital signal processor (DSP) families (hereafter referred to as the ’54x unless otherwise specified) are based on an advanced modified Harvard architecture that has one program memory bus and three data memory buses. These processors also provide an arithmetic logic unit (ALU) that has a high degree of parallelism, application-specific hardware logic, on-chip memory, and additional on-chip peripherals. These DSP families also provide a highly specialized instruction set, which is the basis of the operational flexibility and speed of these DSPs.

Separate program and data spaces allow simultaneous access to program instructions and data, providing the high degree of parallelism. Two reads and one write operation can be performed in a single cycle. Instructions with parallel store and application-specific instructions can fully utilize this architecture. In addition, data can be transferred between data and program spaces. Such parallelism supports a powerful set of arithmetic, logic, and bit-manipulation operations that can all be performed in a single machine cycle. In addition, the ’C54x, ’LC54x, and ’VC54x versions include the control mechanisms to manage interrupts, repeated operations, and function calls.

Table 1 provides an overview of the ’54x generation of DSPs. The table shows significant features of each device including the capacity of on-chip RAM and ROM memories, the peripherals, the execution time of one machine cycle, and the type of package with its total pin count.

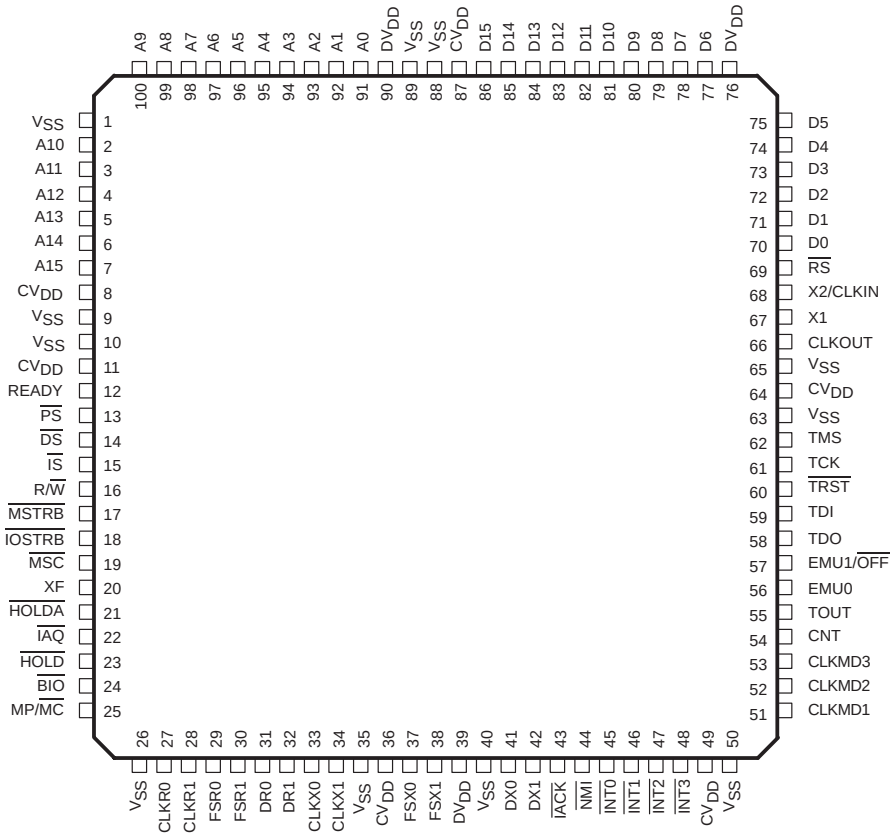
Table 1. Characteristics of the ’54x Processors

DSP TYPE	NOMINAL VOLTAGE (V)	ON-CHIP MEMORY		PERIPHERALS			CYCLE TIME (ns)	PACKAGE TYPE
		RAM† (Word)	ROM (Word)	SERIAL PORT	TIMER	HPI		
TMS320C541	5.0	5K	28K‡	2§	1	No	25	100-pin TQFP
TMS320LC541	3.3	5K	28K‡	2§	1	No	20/25	100-pin TQFP
TMS320C542	5.0	10K	2K	2¶	1	Yes	25	144-pin TQFP
TMS320LC542	3.3	10K	2K	2¶	1	Yes	20/25	128-pin TQFP/144-pin TQFP
TMS320LC543	3.3	10K	2K	2¶	1	No	20/25	100-pin TQFP
TMS320LC545	3.3	6K	48K#	2‖	1	Yes	20/25	128-pin TQFP
TMS320LC545A	3.3	6K	48K#	2‖	1	Yes	15/20/25	128-pin TQFP
TMS320LC546	3.3	6K	48K#	2‖	1	No	20/25	100-pin TQFP
TMS320LC546A	3.3	6K	48K#	2‖	1	No	15/20/25	100-pin TQFP
TMS320LC548	3.3	32K	2K	3☆	1	Yes	15/20	144-pin TQFP/144-pin μstar™ BGA
TMS320LC549	3.3	32K	16K	3☆	1	Yes	12.5/15	144-pin TQFP/144-pin μstar BGA
TMS320VC549	3.3 (2.5 core)	32K	16K	3☆	1	Yes	10	144-pin TQFP/144-pin μstar BGA

Legend:
TQFP = Thin Quad Flatpack
BGA = Ball Grid Array
† The dual-access RAM (single access RAM on ’548 and ’549 devices) can be configured as data memory or program/data memory.
‡ For ’C541/’LC541, 8K words of ROM can be configured as program memory or program/data memory.
§ Two standard (general purpose) serial ports
¶ One TDM and one BSP
# For ’LC545/’LC546, 16K words of ROM can be configured as program memory or program/data memory.
‖ One standard and one BSP
☆ One TDM and two BSPs

μstar is a trademark of Texas Instruments Incorporated.

TMS320C541, TMS320LC541
PZ PACKAGE†
(TOP VIEW)



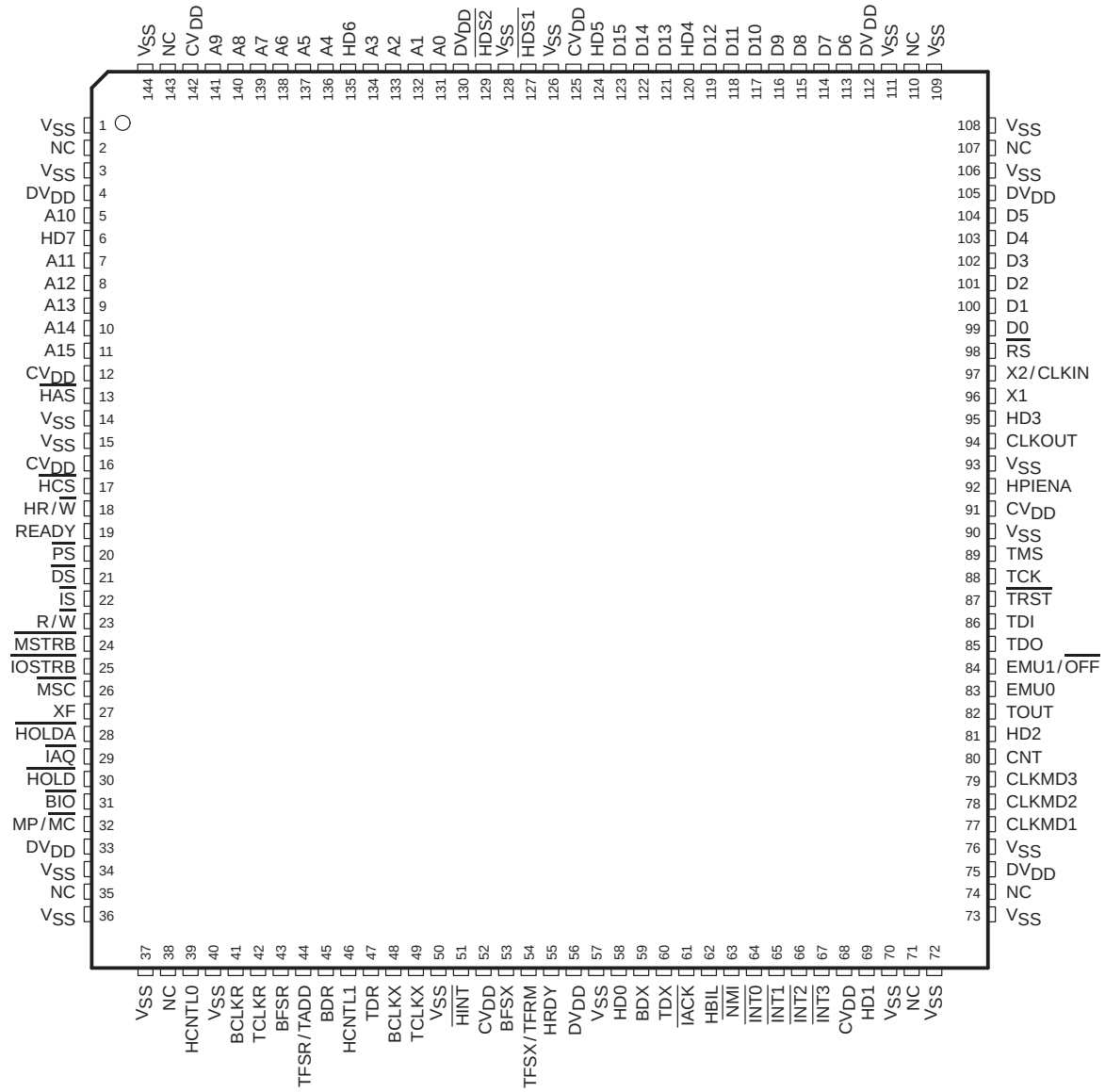
† DV_{DD} is the power supply for the I/O pins while CV_{DD} is the power supply for the core CPU, and V_{SS} is the ground for both the I/O pins and the core CPU.

The ’54x signal descriptions table lists each terminal name, function, and operating mode(s) for the TMS320C541PZ/TMS320LC541PZ (100-pin TQFP packages).

For the ’C541/’LC541 (100-pin packages), no letter in front of CLKRn, FSRn, DRn, CLKXn, FSXn, and DXn pin names denotes standard serial port (where n = 0 or 1 port).

ADVANCE INFORMATION

TMS320C542/TMS320LC542
PGE PACKAGE†‡
(TOP VIEW)



† NC = No connection

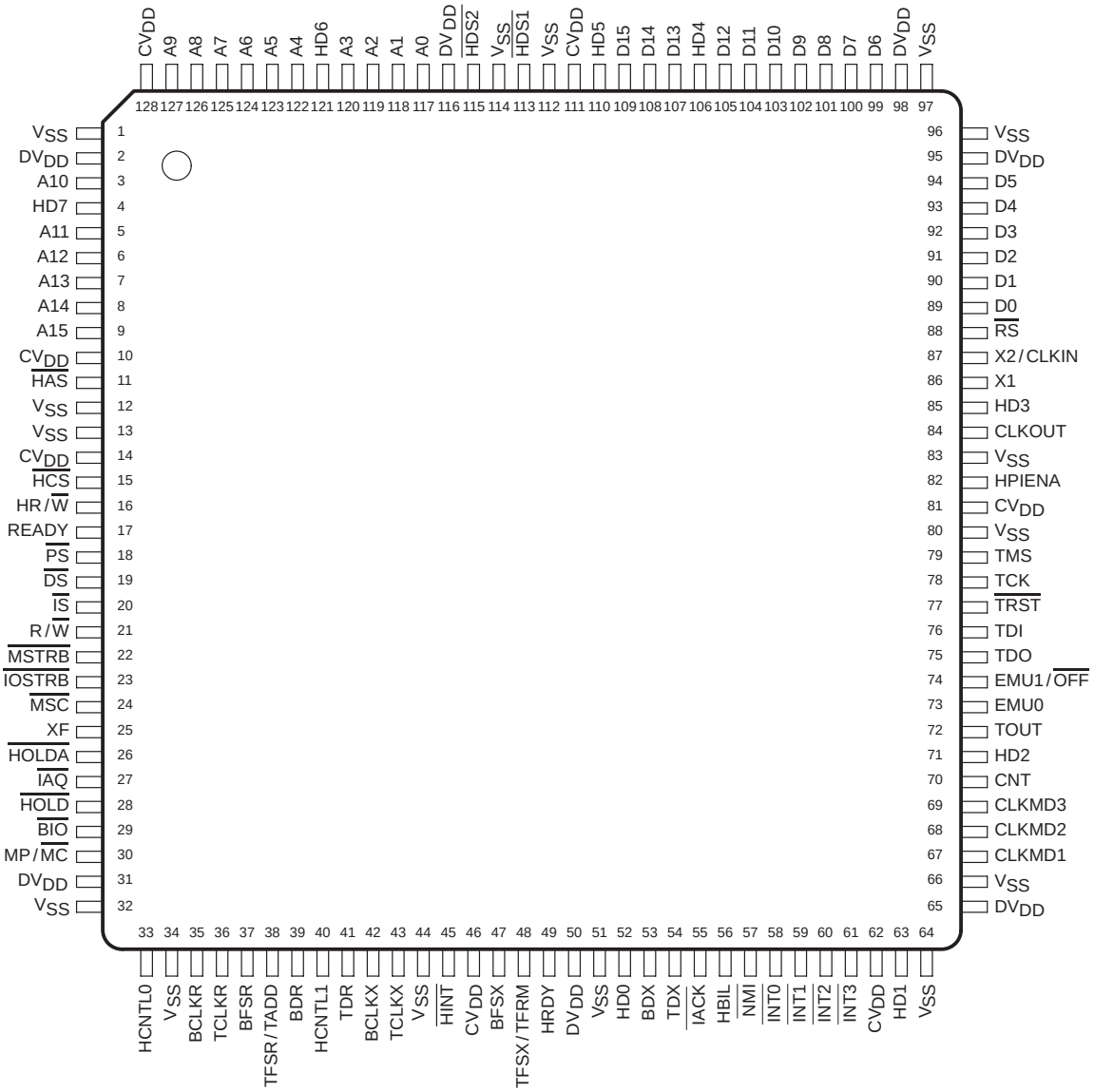
‡ DV_{DD} is the power supply for the I/O pins while CV_{DD} is the power supply for the core CPU, and V_{SS} is the ground for both the I/O pins and the core CPU.

The ’54x signal descriptions table lists each terminal name, function, and operating mode(s) for the TMS320C542PGE/’LC542PGE (144-pin TQFP packages).

For the ’C542/’LC542 (144-pin TQFP packages), the letter B in front of CLKR, FSR, DR, CLKX, FSX, and DX pin names denotes buffered serial port (BSP). The letter T in front of CLKR, FSR, DR, CLKX, FSX, and DX pin names denotes time-division multiplexed (TDM) serial port.

ADVANCE INFORMATION

TMS320LC542
PBK PACKAGE†
(TOP VIEW)



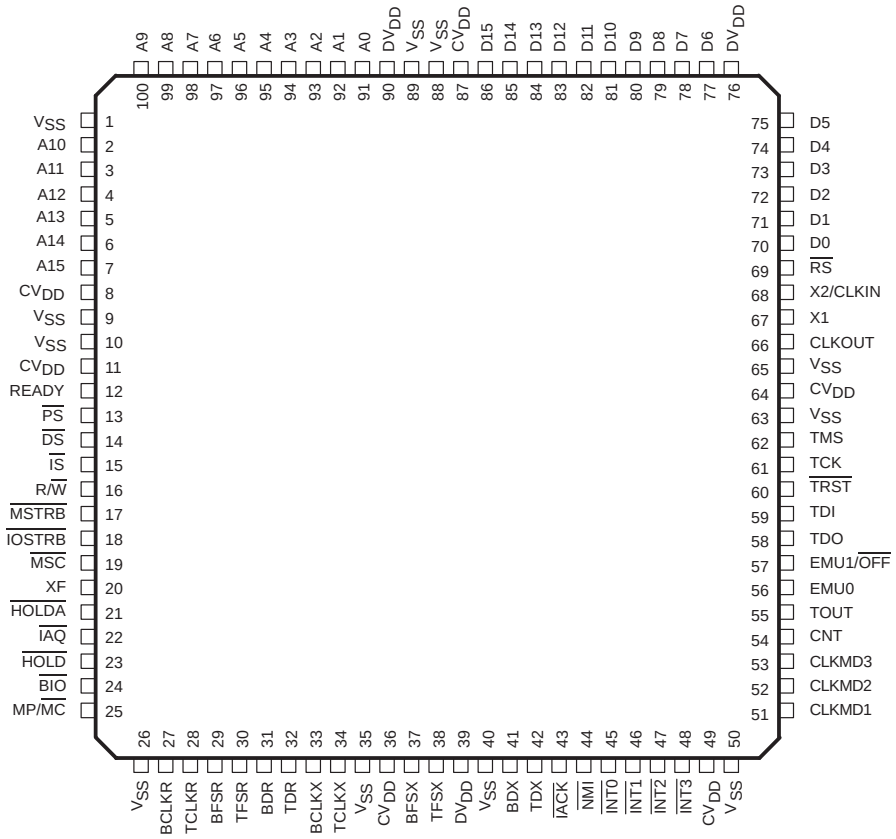
† DV_{DD} is the power supply for the I/O pins while CV_{DD} is the power supply for the core CPU, and V_{SS} is the ground for both the I/O pins and the core CPU.

The ’54x signal descriptions table lists each terminal name, function, and operating mode(s) for the TMS320LC542PBK (128-pin TQFP package).

For the ’LC542 (128-pin TQFP package), the letter B in front of CLKR, FSR, DR, CLKX, FSX, and DX pin names denotes buffered serial port (BSP). The letter T in front of CLKR, FSR, DR, CLKX, FSX, and DX pin names denotes time-division multiplexed (TDM) serial port.

ADVANCE INFORMATION

TMS320LC543
PZ PACKAGE†
(TOP VIEW)



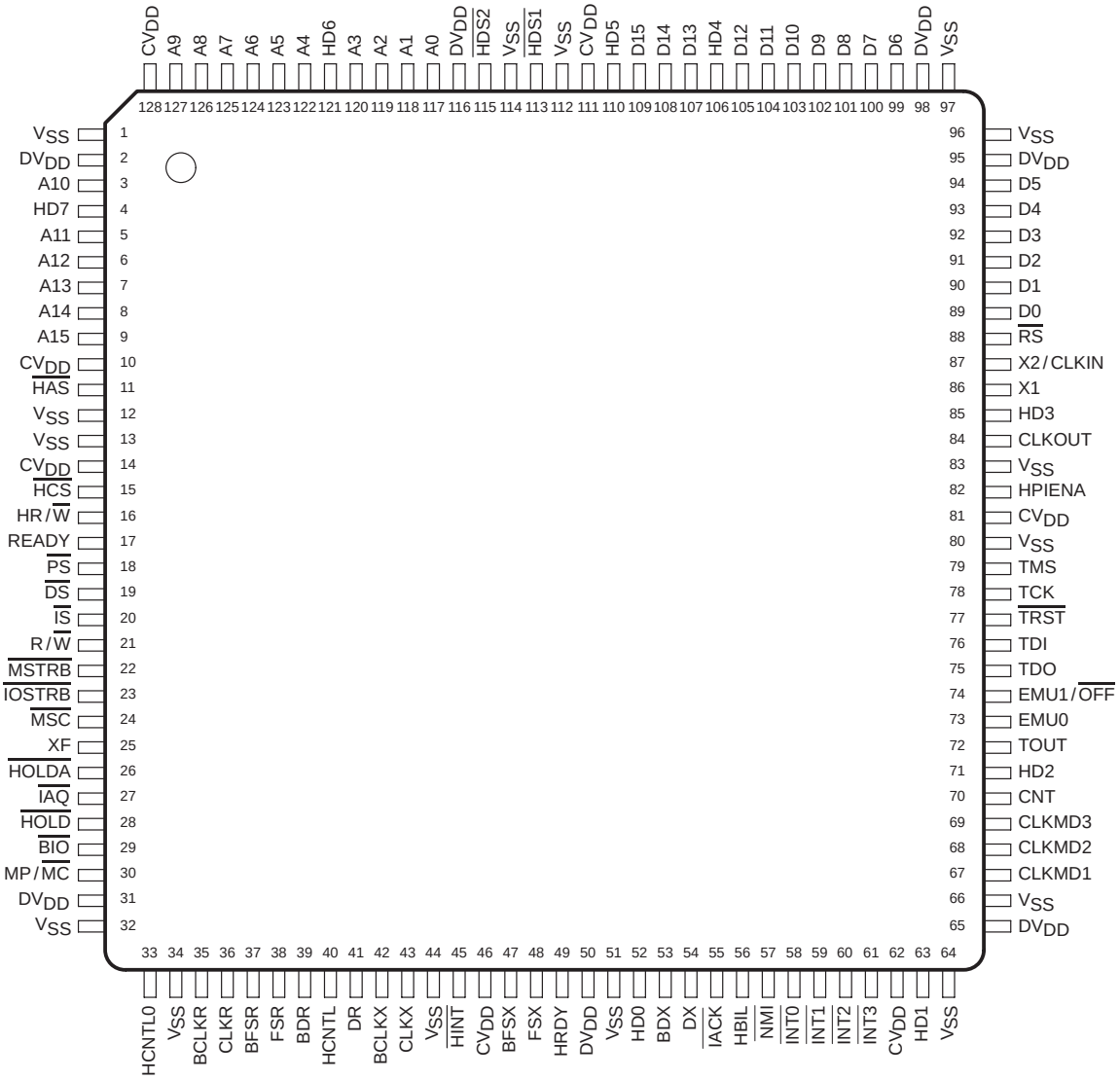
† DV_{DD} is the power supply for the I/O pins while CV_{DD} is the power supply for the core CPU, and V_{SS} is the ground for both the I/O pins and the core CPU.

The ’54x signal descriptions table lists each terminal name, function, and operating mode(s) for the TMS320LC543PZ (100-pin TQFP package).

For the ’LC543 (100-pin TQFP package), the letter B in front of CLKR, FSR, DR, CLKX, FSX, and DX denotes buffered serial port (BSP). The letter T in front of CLKR, FSR, DR, CLKX, FSX, and DX denotes time-division multiplexed (TDM) serial port.

ADVANCE INFORMATION

TMS320LC545
PBK PACKAGE†
(TOP VIEW)



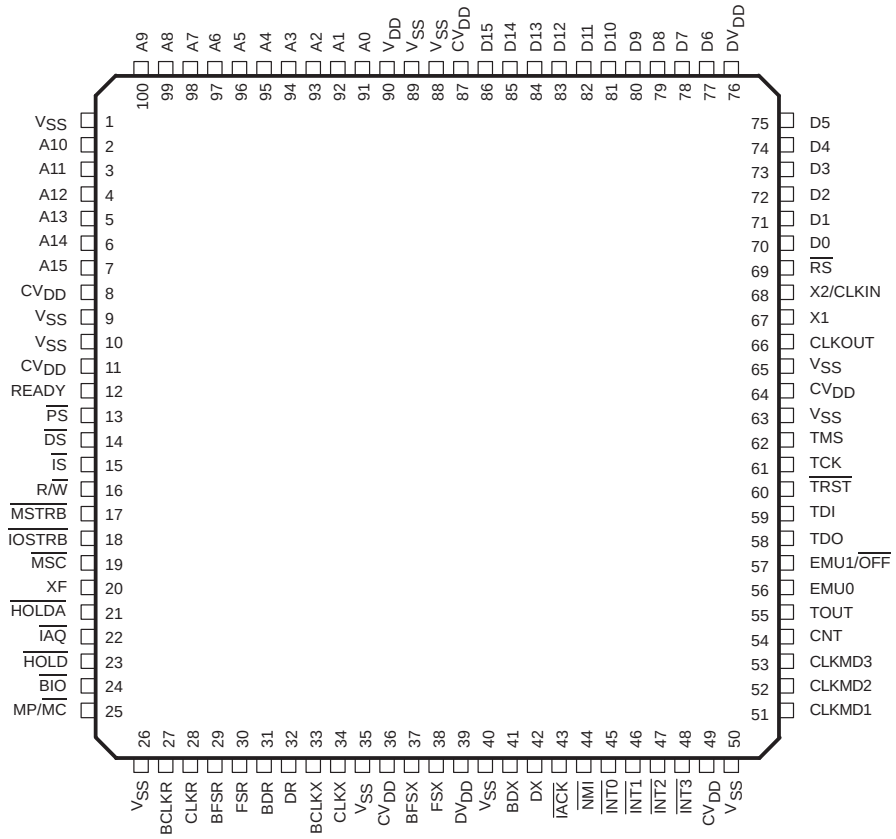
† DV_{DD} is the power supply for the I/O pins while CV_{DD} is the power supply for the core CPU, and V_{SS} is the ground for both the I/O pins and the core CPU.

The ’54x signal descriptions table lists each terminal name, function, and operating mode(s) for the for the TMS320LC545PBK (128-pin TQFP package).

For the ’LC545 (128-pin TQFP package), the letter B in front of CLKR, FSR, DR, CLKX, FSX, and DX pin names denotes buffered serial port (BSP). No letter in front of CLKR, FSR, DR, CLKX, FSX, and DX pin names denotes standard serial port.

ADVANCE INFORMATION

TMS320LC546
PZ PACKAGE†
(TOP VIEW)



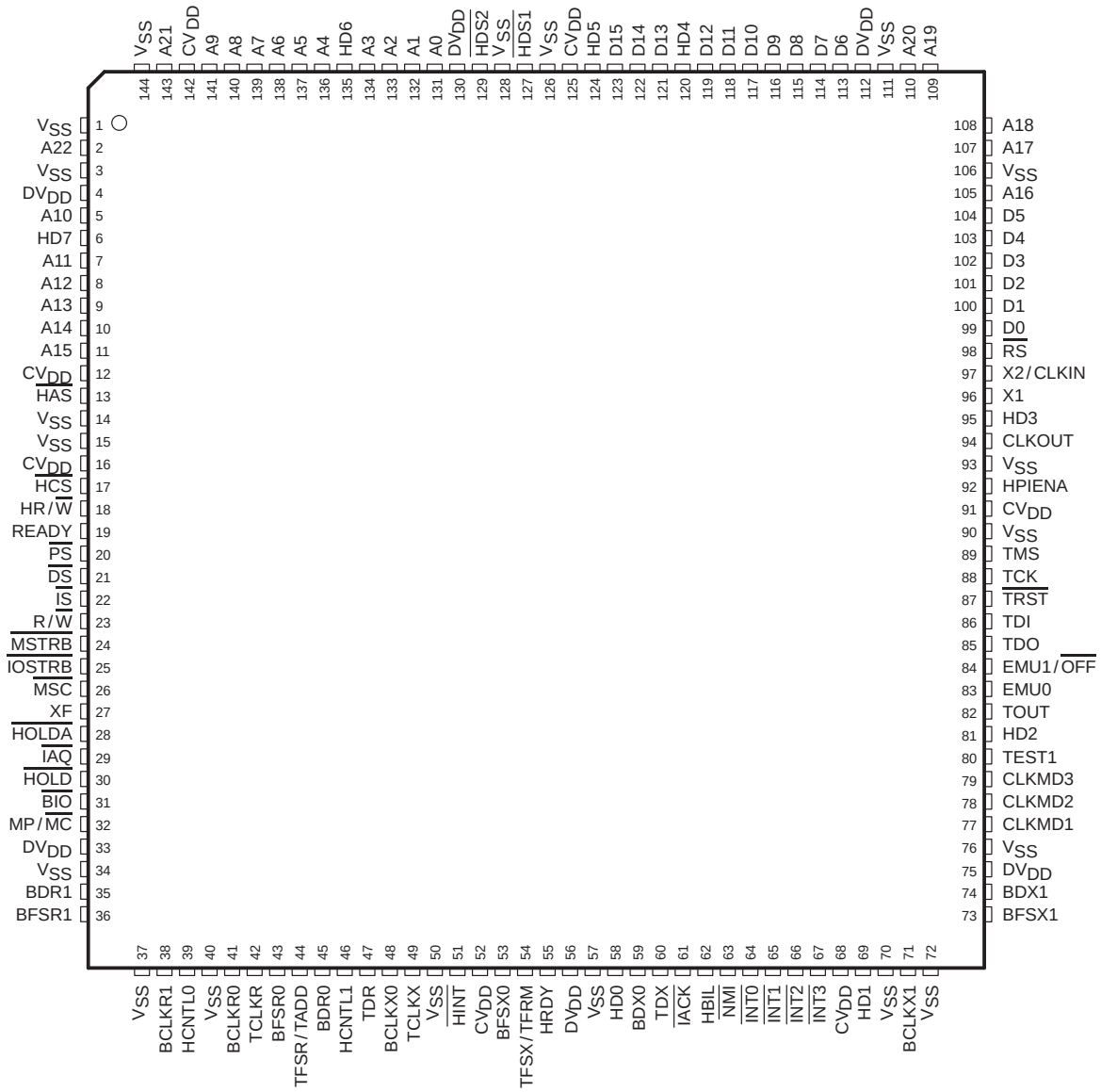
† DV_{DD} is the power supply for the I/O pins while CV_{DD} is the power supply for the core CPU, and V_{SS} is the ground for both the I/O pins and the core CPU.

The ’54x signal descriptions table lists each terminal name, function, and operating mode(s) for the for the TMS320LC546PZ (100-pin TQFP package).

For the ’LC546 (100-pin TQFP package), the letter B in front of CLKR, FSR, DR, FSX, and DX denotes buffered serial port (BSP). No letter in front of CLKR, FSR, DR, FSX, and DX denotes standard serial port.

ADVANCE INFORMATION

TMS320LC548, TMS320LC549, and TMS320VC549
PGE PACKAGE†‡
(TOP VIEW)



† NC = No connection

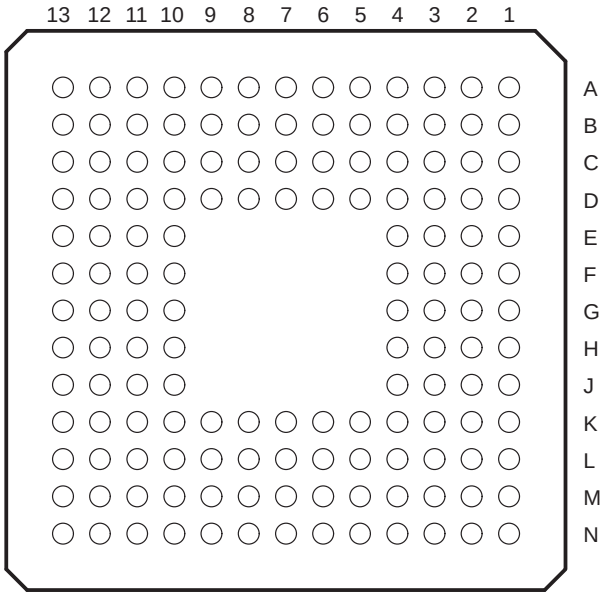
‡ DV_{DD} is the power supply for the I/O pins while CV_{DD} is the power supply for the core CPU, and V_{SS} is the ground for both the I/O pins and the core CPU.

The ’54x signal descriptions table lists each terminal name, function, and operating mode(s) for the TMS320LC548PGE (144-pin TQFP package).

For the ’LC548, ’LC549 and ’VC549 (144-pin TQFP package), the letter B in front of CLKRn, FSRn, DRn, CLKXn, FSXn, and DXn pin names denotes buffered serial port (BSP), where n = 0 or 1 port. The letter T in front of CLKR, FSR, DR, CLKX, FSX, and DX pin names denotes time-division multiplexed (TDM) serial port.

ADVANCE INFORMATION

TMS320LC548, TMS320LC549, TMS320VC549
GGU PACKAGE
(BOTTOM VIEW)



The pin assignments table to follow lists each signal quadrant and BGA ball pin number for the TMS320LC548, TMS320LC549, and TMS320VC549 (144-pin BGA package).

The ’54x signal descriptions table lists each terminal name, function, and operating mode(s) for the TMS320LC548GGU, TMS320LC549GGU, and TMS320VC549GGU.

ADVANCE INFORMATION

Pin Assignments for the TMS320LC548GGU, TMS320LC549GGU, and TMS320VC549GGU (144-Pin BGA Package)†

SIGNAL QUADRANT 1	BGA BALL #	SIGNAL QUADRANT 2	BGA BALL #	SIGNAL QUADRANT 3	BGA BALL #	SIGNAL QUADRANT 4	BGA BALL #
V_{SS}	A1	BFSX1	N13	V_{SS}	N1	A19	A13
A22	B1	BDX1	M13	BCLKR1	N2	A20	A12
V_{SS}	C2	DV_{DD}	L12	HCNTL0	M3	V_{SS}	B11
DV_{DD}	C1	V_{SS}	L13	V_{SS}	N3	DV_{DD}	A11
A10	D4	CLKMD1	K10	BCLKR0	K4	D6	D10
HD7	D3	CLKMD2	K11	TCLKR	L4	D7	C10
A11	D2	CLKMD3	K12	BFSR0	M4	D8	B10
A12	D1	TEST1	K13	TFSR/TADD	N4	D9	A10
A13	E4	HD2	J10	BDR0	K5	D10	D9
A14	E3	TOUT	J11	HCNTL1	L5	D11	C9
A15	E2	EMU0	J12	TDR	M5	D12	B9
CV_{DD}	E1	EMU1/$\overline{OFF}$	J13	BCLKX0	N5	HD4	A9
$\overline{HAS}$	F4	TDO	H10	TCLKX	K6	D13	D8
V_{SS}	F3	TDI	H11	V_{SS}	L6	D14	C8
V_{SS}	F2	$\overline{TRST}$	H12	$\overline{HINT}$	M6	D15	B8
CV_{DD}	F1	TCK	H13	CVDD	N6	HD5	A8
$\overline{HCS}$	G2	TMS	G12	BFSX0	M7	CV_{DD}	B7
HR/$\overline{W}$	G1	V_{SS}	G13	TFSX/TFRM	N7	V_{SS}	A7
READY	G3	CV_{DD}	G11	HRDY	L7	$\overline{HDS1}$	C7
$\overline{PS}$	G4	HPIENA	G10	DV_{DD}	K7	V_{SS}	D7
$\overline{DS}$	H1	V_{SS}	F13	V_{SS}	N8	$\overline{HDS2}$	A6
$\overline{IS}$	H2	CLKOUT	F12	HD0	M8	DV_{DD}	B6
R/$\overline{W}$	H3	HD3	F11	BDX0	L8	A0	C6
$\overline{MSTRB}$	H4	X1	F10	TDX	K8	A1	D6
$\overline{IOSTRB}$	J1	X2/CLKIN	E13	$\overline{IACK}$	N9	A2	A5
$\overline{MSC}$	J2	$\overline{RS}$	E12	HBIL	M9	A3	B5
XF	J3	D0	E11	$\overline{NMI}$	L9	HD6	C5
$\overline{HOLDA}$	J4	D1	E10	$\overline{INT0}$	K9	A4	D5
$\overline{IAQ}$	K1	D2	D13	$\overline{INT1}$	N10	A5	A4
$\overline{HOLD}$	K2	D3	D12	$\overline{INT2}$	M10	A6	B4
$\overline{BIO}$	K3	D4	D11	$\overline{INT3}$	L10	A7	C4
MP/$\overline{MC}$	L1	D5	C13	CV_{DD}	N11	A8	A3
DV_{DD}	L2	A16	C12	HD1	M11	A9	B3
V_{SS}	L3	V_{SS}	C11	V_{SS}	L11	CV_{DD}	C3
BDR1	M1	A17	B13	BCLKX1	N12	A21	A2
BFSR1	M2	A18	B12	V_{SS}	M12	V_{SS}	B2

† DV_{DD} is the power supply for the I/O pins while CV_{DD} is the power supply for the core CPU, and V_{SS} is the ground for both the I/O pins and the core CPU.

ADVANCE INFORMATION

’54x Signal Descriptions

TERMINAL NAME	TERMINAL TYPE†	DESCRIPTION
DATA SIGNALS		
A22 (MSB) A21 A20 A19 A18 A17 A16 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 (LSB)	O/Z	Parallel port address bus A22 (MSB) through A0 (LSB). The sixteen LSBs (A15–A0) are multiplexed to address external data/program memory or I/O. A15–A0 are placed in the high-impedance state in the hold mode. A15–A0 also go into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low. The seven MSBs (A22 to A16) are used for extended program memory addressing (’548 and ’549 only). On the ’548 and ’549 devices, the address bus have a feature called bus holder that eliminates passive components and the power dissipation associated with it. The bus holder keeps the address bus at the previous logic level when the bus goes into a high-impedance state.
D15 (MSB) D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 (LSB)	I/O/Z	Parallel port data bus D15 (MSB) through D0 (LSB). D15–D0 are multiplexed to transfer data between the core CPU and external data/program memory or I/O devices. D15–D0 are placed in the high-impedance state when not output or when $\overline{\text{RS}}$ or $\overline{\text{HOLD}}$ is asserted. D15–D0 also go into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low. The data bus has a feature called bus holder that eliminates passive components and the power dissipation associated with it. The bus holder keeps the data bus at the previous logic level when the bus goes into a high-impedance state.
INITIALIZATION, INTERRUPT AND RESET OPERATIONS		
$\overline{\text{IACK}}$	O/Z	Interrupt acknowledge signal. $\overline{\text{IACK}}$ indicates the receipt of an interrupt and that the program counter is fetching the interrupt vector location designated by A15–0. $\overline{\text{IACK}}$ also goes into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
$\overline{\text{INT0}}$ $\overline{\text{INT1}}$ $\overline{\text{INT2}}$ $\overline{\text{INT3}}$	I	External user interrupt inputs. $\overline{\text{INT0}}$–$\overline{\text{INT3}}$ are prioritized and are maskable by the interrupt mask register and the interrupt mode bit. $\overline{\text{INT0}}$ –$\overline{\text{INT3}}$ can be polled and reset by the interrupt flag register.

† I = Input, O = Output, Z = High impedance

’54x Signal Descriptions (Continued)

TERMINAL NAME	TERMINAL TYPE†	DESCRIPTION
INITIALIZATION, INTERRUPT AND RESET OPERATIONS (CONTINUED)		
$\overline{\text{NMI}}$	I	Nonmaskable interrupt. $\overline{\text{NMI}}$ is an external interrupt that cannot be masked by way of the INTM or the IMR. When $\overline{\text{NMI}}$ is activated, the processor traps to the appropriate vector location.
$\overline{\text{RS}}$	I	Reset input. $\overline{\text{RS}}$ causes the DSP to terminate execution and forces the program counter to 0FF80h. When $\overline{\text{RS}}$ is brought to a high level, execution begins at location 0FF80h of the program memory. $\overline{\text{RS}}$ affects various registers and status bits.
MP/$\overline{\text{MC}}$	I	Microprocessor/microcomputer mode-select pin. If active-low at reset (microcomputer mode), MP/$\overline{\text{MC}}$ causes the internal program ROM to be mapped into the upper program memory space. In the microprocessor mode, off-chip memory and its corresponding addresses (instead of internal program ROM) are accessed by the DSP.
CNT	I	I/O level select. For 5-V operation, all input and output voltage levels are TTL-compatible when CNT is pulled down to a low level. For 3-V operation with CMOS-compatible I/O interface levels, CNT is pulled to a high level.
MULTIPROCESSING SIGNALS		
$\overline{\text{BIO}}$	I	Branch control input. A branch can be conditionally executed when $\overline{\text{BIO}}$ is active. If low, the processor executes the conditional instruction. The $\overline{\text{BIO}}$ condition is sampled during the decode phase of the pipeline for the XC instruction, and all other instructions sample $\overline{\text{BIO}}$ during the read phase of the pipeline.
XF	O/Z	External flag output (latched software-programmable signal). XF is set high by the SSBX XF instruction, set low by RSBX XF instruction or by loading the ST1 status register. XF is used for signaling other processors in multiprocessor configurations or as a general-purpose output pin. XF goes into the high-impedance state when $\overline{\text{OFF}}$ is low, and is set high at reset.
MEMORY CONTROL SIGNALS		
$\overline{\text{DS}}$ $\overline{\text{PS}}$ $\overline{\text{IS}}$	O/Z	Data, program, and I/O space select signals. $\overline{\text{DS}}$, $\overline{\text{PS}}$, and $\overline{\text{IS}}$ are always high unless driven low for communicating to a particular external space. Active period corresponds to valid address information. Placed into a high-impedance state in hold mode. $\overline{\text{DS}}$, $\overline{\text{PS}}$, and $\overline{\text{IS}}$ also go into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
$\overline{\text{MSTRB}}$	O/Z	Memory strobe signal. $\overline{\text{MSTRB}}$ is always high unless low-level asserted to indicate an external bus access to data or program memory. Placed in high-impedance state in hold mode. $\overline{\text{MSTRB}}$ also goes into the high-impedance state when $\overline{\text{OFF}}$ is low.
READY	I	Data-ready input. READY indicates that an external device is prepared for a bus transaction to be completed. If the device is not ready (READY is low), the processor waits one cycle and checks READY again. Note that the processor performs ready-detection if at least two software wait states are programmed. The READY signal is not sampled until the completion of the software wait states.
R/$\overline{\text{W}}$	O/Z	Read/write signal. R/$\overline{\text{W}}$ indicates transfer direction during communication to an external device and is normally high (in read mode), unless asserted low when the DSP performs a write operation. Placed in the high-impedance state in hold mode, R/$\overline{\text{W}}$ also goes into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
$\overline{\text{IOSTRB}}$	O/Z	I/O strobe signal. $\overline{\text{IOSTRB}}$ is always high unless low level asserted to indicate an external bus access to an I/O device. Placed in high-impedance state in hold mode. $\overline{\text{IOSTRB}}$ also goes into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
$\overline{\text{HOLD}}$	I	Hold input. $\overline{\text{HOLD}}$ is asserted to request control of the address, data, and control lines. When acknowledged by the ’54x, these lines go into high-impedance state.
$\overline{\text{HOLDA}}$	O/Z	Hold acknowledge signal. $\overline{\text{HOLDA}}$ indicates to the external circuitry that the processor is in a hold state and that the address, data, and control lines are in a high-impedance state, allowing them to be available to the external circuitry. $\overline{\text{HOLDA}}$ also goes into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
$\overline{\text{MSC}}$	O/Z	Microstate complete signal. $\overline{\text{MSC}}$ goes low when the last wait state of two or more internal software wait states programmed are executed. If connected to the READY line, $\overline{\text{MSC}}$ forces one external wait state after the last internal wait state has been completed. $\overline{\text{MSC}}$ also goes into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.

† I = Input, O = Output, Z = High impedance

ADVANCE INFORMATION

’54x Signal Descriptions (Continued)

TERMINAL NAME	TERMINAL TYPE†	DESCRIPTION
MEMORY CONTROL SIGNALS (CONTINUED)		
$\overline{IAQ}$	O/Z	Instruction acquisition signal. $\overline{IAQ}$ is asserted (active low) when there is an instruction address on the address bus and goes into the high-impedance state when EMU1/$\overline{OFF}$ is low.
OSCILLATOR/TIMER SIGNALS		
CLKOUT	O/Z	Master clock output signal. CLKOUT cycles at the machine-cycle rate of the CPU. The internal machine cycle is bounded by the falling edges of this signal. CLKOUT also goes into the high-impedance state when EMU1/$\overline{OFF}$ is low.
CLKMD1 CLKMD2 CLKMD3	I	Clock mode external/internal input signals. CLKMD1, CLKMD2, and CLKMD3 allow you to select and configure different clock modes, such as crystal, external clock, and various PLL factors. Refer to PLL section for a detailed functional description of these pins.
X2/CLKIN	I	Input pin to internal oscillator from the crystal. If the internal (crystal) oscillator is not being used, a clock can become input to the device using this pin. The internal machine cycle time is determined by the clock operating-mode pins (CLKMD1, CLKMD2 and CLKMD3).
X1	O	Output pin from the internal oscillator for the crystal. If the internal oscillator is not used, X1 should be left unconnected. X1 does not go into the high-impedance state when EMU1/$\overline{OFF}$ is low.
TOUT	O/Z	Timer output. TOUT signals a pulse when the on-chip timer counts down past zero. The pulse is a CLKOUT-cycle wide. TOUT also goes into the high-impedance state when EMU1/$\overline{OFF}$ is low.
BUFFERED SERIAL PORT 0 AND BUFFERED SERIAL PORT 1 SIGNALS		
BCLKR0 BCLKR1	I	Receive clocks. External clock signal for clocking data from the data-receive (DR) pin into the buffered serial port receive shift registers (RSRs). Must be present during buffered serial port transfers. If the buffered serial port is not being used, BCLKR0 and BCLKR1 can be sampled as an input by way of IN0 bit of the SPC register.
BCLKX0 BCLKX1	I/O/Z	Transmit clock. Clock signal for clocking data from the serial port transmit shift register (XSR) to the data transmit (DX) pin. BCLKX can be an input if MCM in the serial port control register is cleared to 0. It also can be driven by the device at 1/(CLKDV + 1) where CLKDV range is 0–31 CLKOUT frequency when MCM is set to 1. If the buffered serial port is not used, BCLKX can be sampled as an input by way of IN1 of the SPC register. BCLKX0 and BCLKX1 go into the high-impedance state when $\overline{OFF}$ is low.
BDR0 BDR1	I	Buffered serial-data-receive input. Serial data is received in the RSR by BDR0/BDR1.
BDX0 BDX1	O/Z	Buffered serial-port-transmit output. Serial data is transmitted from the XSR by way of BDX. BDX0 and BDX1 are placed in the high-impedance state when not transmitting and when EMU1/$\overline{OFF}$ is low.
BFSR0 BFSR1	I	Frame synchronization pulse for receive input. The falling edge of the BFSR pulse initiates the data-receive process, beginning the clocking of the RSR.
BFSX0 BFSX1	I/O/Z	Frame synchronization pulse for transmit input/output. The falling edge of the BFSX pulse initiates the data-transmit process, beginning the clocking of the XSR. Following reset, the default operating condition of BFSX is an input. BFSX0 and BFSX1 can be selected by software to be an output when TXM in the serial control register is set to 1. This pin goes into the high-impedance state when EMU1/$\overline{OFF}$ is low.
SERIAL PORT 0 AND SERIAL PORT 1 SIGNALS		
CLKR0 CLKR1	I	Receive clocks. External clock signal for clocking data from the data receive (DR) pin into the serial port receive shift register (RSR). Must be present during serial port transfers. If the serial port is not being used, CLKR0 and CLKR1 can be sampled as an input via IN0 bit of the SPC register.
CLKX0 CLKX1	I/O/Z	Transmit clock. Clock signal for clocking data from the serial port transmit shift register (XSR) to the data transmit (DX) pin. CLKX can be an input if MCM in the serial port control register is cleared to 0. It also can be driven by the device at 1/4 CLKOUT frequency when MCM is set to 1. If the serial port is not used, CLKX can be sampled as an input via IN1 of the SPC register. CLKX0 and CLKX1 go into the high-impedance state when EMU1/$\overline{OFF}$ is low.
DR0 DR1	I	Serial-data-receive input. Serial data is received in the RSR by DR.

† I = Input, O = Output, Z = High impedance

’54x Signal Descriptions (Continued)

TERMINAL NAME	TERMINAL TYPE†	DESCRIPTION
SERIAL PORT 0 AND SERIAL PORT 1 SIGNALS (CONTINUED)		
DX0 DX1	O/Z	Serial port transmit output. Serial data is transmitted from the XSR via DX. DX0 and DX1 are placed in the high-impedance state when not transmitting and when EMU1/$\overline{\text{OFF}}$ is low.
FSR0 FSR1	I	Frame synchronization pulse for receive input. The falling edge of the FSR pulse initiates the data-receive process, beginning the clocking of the RSR.
FSX0 FSX1	I/O/Z	Frame synchronization pulse for transmit input/output. The falling edge of the FSX pulse initiates the data transmit process, beginning the clocking of the XSR. Following reset, the default operating condition of FSX is an input. FSX0 and FSX1 can be selected by software to be an output when TXM in the serial control register is set to 1. This pin goes into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
TDM SERIAL PORT SIGNALS		
TCLKR	I	TDM receive clock input
TDR	I	TDM serial data-receive input
TFSR/TADD	I/O	TDM receive frame synchronization or TDM address
TCLKX	I/O/Z	TDM transmit clock
TDX	O/Z	TDM serial data-transmit output
TFSX/TFRM	I/O/Z	TDM transmit frame synchronization
HOST PORT INTERFACE SIGNALS		
HD0–HD7	I/O/Z	Parallel bidirectional data bus. HD0–HD7 are placed in the high-impedance state when not outputting data. The signals go into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
HCNTL0 HCNTL1	I	Control inputs
HBIL	I	Byte-identification input
$\overline{\text{HCS}}$	I	Chip-select input
$\overline{\text{HDS1}}$ $\overline{\text{HDS2}}$	I	Data strobe inputs
$\overline{\text{HAS}}$	I	Address strobe input
HR/$\overline{\text{W}}$	I	Read/write input
HRDY	O/Z	Ready output. This signal goes into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
$\overline{\text{HINT}}$	O/Z	Interrupt output. When the DSP is in reset, this signal is driven high. The signal goes into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
HPIENA	I	HPI module select input. This signal must be tied to a logic 1 state to have HPI selected. If this input is left open or connected to ground, the HPI module will not be selected, internal pullup for the HPI input pins are enabled, and the HPI data bus has keepers set. This input is provided with an internal pull-down resistor which is active only when $\overline{\text{RS}}$ is low. HPIENA is sampled when $\overline{\text{RS}}$ goes high and ignored until $\overline{\text{RS}}$ goes low again. Refer to the Electrical Characteristics section for the input current requirements for this pin.
SUPPLY PINS		
CV_{DD}	Supply	$+V_{DD}$. CV_{DD} is the dedicated power supply for the core CPU.
DV_{DD}	Supply	$+V_{DD}$. DV_{DD} is the dedicated power supply for I/O pins.
V_{SS}	Supply	Ground. V_{SS} is the dedicated power ground for the device.

† I = Input, O = Output, Z = High impedance

ADVANCE INFORMATION

’54x Signal Descriptions (Continued)

TERMINAL NAME	TERMINAL TYPE†	DESCRIPTION
IEEE1149.1 TEST PINS		
TCK	I	IEEE standard 1149.1 test clock. This is normally a free-running clock signal with a 50% duty cycle. The changes on the test-access port (TAP) of input signals TMS and TDI are clocked into the TAP controller, instruction register, or selected test data register on the rising edge of TCK. Changes at the TAP output signal (TDO) occur on the falling edge of TCK.
TDI	I	IEEE standard 1149.1 test data input. Pin with internal pullup device. TDI is clocked into the selected register (instruction or data) on a rising edge of TCK.
TDO	O/Z	IEEE standard 1149.1 test data output. The contents of the selected register (instruction or data) is shifted out of TDO on the falling edge of TCK. TDO is in the high-impedance state except when the scanning of data is in progress. TDO also goes into the high-impedance state when EMU1/$\overline{\text{OFF}}$ is low.
TMS	I	IEEE standard 1149.1 test mode select. Pin with internal pullup device. This serial control input is clocked into the TAP controller on the rising edge of TCK.
$\overline{\text{TRST}}$	I	IEEE standard 1149.1 test reset. $\overline{\text{TRST}}$, when high, gives the IEEE standard 1149.1 scan system control of the operations of the device. If $\overline{\text{TRST}}$ is not connected or driven low, the device operates in its functional mode, and the IEEE standard 1149.1 signals are ignored. Pin with internal pulldown device.
EMU0	I/O/Z	Emulator interrupt 0 pin. When $\overline{\text{TRST}}$ is driven low, EMU0 must be high for the activation of the EMU1/$\overline{\text{OFF}}$ condition. When $\overline{\text{TRST}}$ is driven high, EMU0 is used as an interrupt to or from the emulator system and is defined as input/output by way of IEEE standard 1149.1 scan system.
EMU1/$\overline{\text{OFF}}$	I/O/Z	Emulator interrupt 1 pin/disable all outputs. When $\overline{\text{TRST}}$ is driven high, EMU1/$\overline{\text{OFF}}$ is used as an interrupt to or from the emulator system and is defined as input/output by way of IEEE standard 1149.1 scan system. When $\overline{\text{TRST}}$ is driven low, EMU1/$\overline{\text{OFF}}$ is configured as $\overline{\text{OFF}}$. The EMU1/$\overline{\text{OFF}}$ signal, when active low, puts all output drivers into the high-impedance state. Note that $\overline{\text{OFF}}$ is used exclusively for testing and emulation purposes (not for multiprocessing applications). Therefore, for the $\overline{\text{OFF}}$ condition, the following conditions apply: $\overline{\text{TRST}}$ = low, EMU0 = high EMU1/$\overline{\text{OFF}}$ = low
DEVICE TEST PIN		
TEST1	I	Test1 – Reserved for internal use only (’LC548, ’LC549, and ’VC549 only). **This pin must not be connected (NC).**

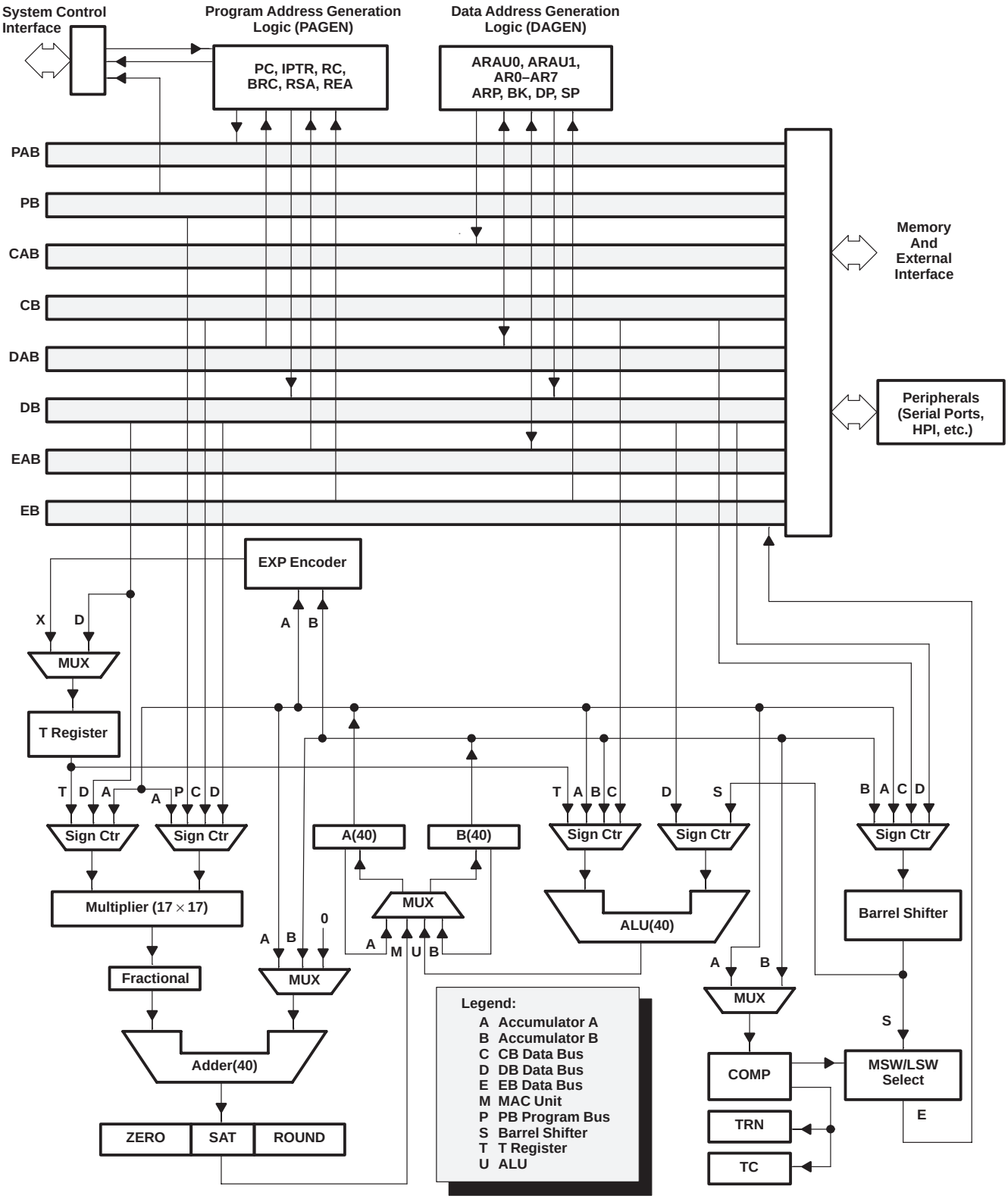
† I = Input, O = Output, Z = High impedance

architecture

The ’54x DSPs use an advanced, modified Harvard architecture that maximizes processing power by maintaining three separate bus structures for data memory and one for program memory. Separate program and data spaces allow simultaneous access to program instructions and data, providing a high degree of parallelism. For example, two read and one write operations can be performed in a single cycle. Instructions with parallel store and application-specific instructions fully utilize this architecture. In addition, data can be transferred between data and program spaces. Such parallelism supports a powerful set of arithmetic, logic, and bit-manipulation operations that can all be performed in a single machine cycle. In addition, the ’54x include the control mechanisms to manage interrupts, repeated operations, and function calls.

The functional block diagram includes the principal blocks and bus structure in the ’54x devices.

functional block diagram of the ’54x internal hardware



ADVANCE INFORMATION

central processing unit (CPU)

The CPU of the ’54x devices contains:

- A 40-bit arithmetic logic unit (ALU)
- Two 40-bit accumulators
- A barrel shifter
- A 17 × 17-bit multiplier/adder
- A compare, select and store unit (CSSU)

arithmetic logic unit (ALU)

The ’54x devices perform 2s-complement arithmetic using: a 40-bit arithmetic logic unit (ALU) and two 40-bit accumulators (ACCA and ACCB). The ALU also can perform Boolean operations.

The ALU can function as two 16-bit ALUs and perform two 16-bit operations simultaneously when the C16 bit in status register 1 (ST1) is set.

accumulators

The accumulators, ACCA and ACCB, store the output from the ALU or the multiplier / adder block; the accumulators can also provide a second input to the ALU or the multiplier / adder. The accumulators are divided into three parts:

- Guard bits (bits 32–39)
- A high-order word (bits 16–31)
- A low-order word (bits 0–15)

Instructions are provided for storing the guard bits, the high- and the low-order accumulator words in data memory, and for manipulating 32-bit accumulator words in or out of data memory. Also, any of the accumulators can be used as temporary storage for the other.

barrel shifter

The ’54x’s barrel shifter has a 40-bit input connected to the accumulator, or data memory (CB, DB) and a 40-bit output connected to the ALU, or data memory (EB). The barrel shifter produces a left shift of 0 to 31 bits and a right shift of 0 to 16 bits on the input data. The shift requirements are defined in the shift-count field (ASM) of ST1 or defined in the temporary register (TREG), which is designated as a shift-count register. This shifter and the exponent detector normalize the values in an accumulator in a single cycle. The least significant bits (LSBs) of the output are filled with 0s and the most significant bits (MSBs) can be either zero-filled or sign-extended, depending on the state of the sign-extended mode bit (SXM) of ST1. Additional shift capabilities enable the processor to perform numerical scaling, bit extraction, extended arithmetic, and overflow prevention operations.

multiplier/adder

The multiplier / adder performs 17 × 17-bit 2s-complement multiplication with a 40-bit accumulation in a single instruction cycle. The multiplier / adder block consists of several elements: a multiplier, adder, signed / unsigned input control, fractional control, a zero detector, a rounder (2s-complement), overflow / saturation logic, and TREG. The multiplier has two inputs: one input is selected from the TREG, a data-memory operand, or an accumulator; the other is selected from the program memory, the data memory, an accumulator, or an immediate value. The fast on-chip multiplier allows the ’54x to perform operations such as convolution, correlation, and filtering efficiently.

In addition, the multiplier and ALU together execute multiply/accumulate (MAC) computations and ALU operations in parallel in a single instruction cycle. This function is used in determining the Euclid distance, and in implementing symmetrical and least mean square (LMS) filters, which are required for complex DSP algorithms.

compare, select and store unit (CSSU)

The compare, select and store unit (CSSU) performs maximum comparisons between the accumulator's high and low word, allows the test/control (TC) flag bit of status register 0 (ST0) and the transition (TRN) register to keep their transition histories, and selects the larger word in the accumulator to be stored in data memory. The CSSU also accelerates Viterbi-type butterfly computation with optimized on-chip hardware.

program control

Program control is provided by several hardware and software mechanisms:

- The program controller decodes instructions, manages the pipeline, stores the status of operations, and decodes conditional operations. Some of the hardware elements included in the program controller are the program counter, the status and control register, the stack, and the address-generation logic.
- Some of the software mechanisms used for program control include branches, calls, conditional instructions, a repeat instruction, reset, and interrupts.

power-down modes

There are three power-down modes, activated by the IDLE1, IDLE2, and IDLE3 instructions. In these modes, the '54x devices enter a dormant state and dissipate considerably less power than in normal operation. The IDLE1 instruction is used to shut down the CPU. The IDLE2 instruction is used to shut down the CPU and on-chip peripherals. The IDLE3 instruction is used to shut down the '54x processor completely. This instruction stops the PLL circuitry as well as the CPU and peripherals.

bus structure

The '54x device architecture is built around eight major 16-bit buses:

- One program-read bus (PB), which carries the instruction code and immediate operands from program memory
- Two data-read buses (CB, DB) and one data-write bus (EB), which interconnect to various elements, such as the CPU, data-address generation logic, program-address generation logic, on-chip peripherals, and data memory
  - The CB and DB carry the operands read from data memory.
  - The EB carries the data to be written to memory.
- Four address buses (PAB, CAB, DAB, and EAB), which carry the addresses needed for instruction execution

The '54x devices have the capability to generate up to two data-memory addresses per cycle, which are stored into two auxiliary register arithmetic units (ARAU0 and ARAU1).

The PB can carry data operands stored in program space (for instance, a coefficient table) to the multiplier for multiply/accumulate operations or to a destination in data space for the data move instruction. This capability allows implementation of single-cycle three-operand instructions such as FIRS.

The '54x devices also have an on-chip bidirectional bus for accessing on-chip peripherals; this bus is connected to DB and EB through the bus exchanger in the CPU interface. Accesses using this bus can require more than two cycles for reads and writes depending on the peripheral's structure.

The '54x devices can have bus keepers connected to the data bus. Bus keepers ensure that the data bus does not float. When bus keepers are enabled, the data bus maintains its previous level. Setting bit 1 of the bank switching control register (BSCR) enables bus keepers and clearing bit 1 disables the bus keepers. A reset automatically disables the bus keepers.

bus structure (continued)

The ’548 and ’549 devices also have equivalent bus keepers connected to the address bus. The bus keepers ensure the address bus does not float when in high-impedance. For the ’548 and ’549 devices, the bus keepers are always enabled.

Table 2 summarizes the buses used by various types of accesses.

Table 2. Bus Usage for Accesses

ACCESS TYPE	ADDRESS BUS				PROGRAM BUS	DATA BUS		
	PAB	CAB	DAB	EAB	PB	CB	DB	EB
Program read	√				√			
Program write	√							√
Data single read			√				√	
Data dual read		√	√			√	√	
Data long (32-bit) read		√(hw)	√(lw)			√(hw)	√(lw)	
Data single write				√				√
Data read/data write			√	√			√	√
Dual read/coefficient read	√	√	√		√	√	√	
Peripheral read			√				√	
Peripheral write				√				√

Legend:
hw = high 16-bit word
lw = low 16-bit word

memory

The total memory address range for the host of ’54x devices is 192K 16-bit words. The ’548 and ’549 devices have 8M-word program memory. The memory space is divided into three specific memory segments: 64K-word program, 64K-word data, and 64K-word I/O. The program memory space contains the instructions to be executed as well as tables used in execution. The data memory space stores data used by the instructions. The I/O memory space interfaces to external memory-mapped peripherals and can also serve as extra data storage space.

The parallel nature of the architecture of these DSPs allows them to perform four concurrent memory operations in any given machine cycle: fetching an instruction, reading two operands, and writing an operand. The four parallel buses are the program-read bus (PB), the data-write bus (EB) and the two data-read buses (CB and DB). Each bus accesses different memory spaces for different aspects of the DSP’s operation. Additionally, this architecture allows dual-operand reads, 32-bit-long word accesses, and a single read with a parallel store.

The ’54x DSPs include on-chip memory to aid in system performance and integration.

on-chip ROM

The ’C541 and ’LC541 feature a 28K-word × 16-bit on-chip maskable ROM. 8K words of the ’C541 and ’LC541 ROM can be mapped into program and data memory space if the data ROM (DROM) bit in the processor mode status (PMST) register is set. This allows an instruction to use data stored in the ROM as an operand.

The ’LC545/’LC546 all feature a 48K-word × 16-bit on-chip maskable ROM. 16K words of the ROM on these devices can be mapped into program and data memory space if the DROM bit in the PMST register is set.

The ’C542/’LC542/’LC543/ ’LC548 all feature 2K-word × 16-bit on-chip ROM.

The ’LC549 and ’VC549 feature 16K-word x 16-bit on-chip ROM.

on-chip ROM (continued)

Customers can arrange to have the ROM of the ’54x programmed with contents unique to any particular application.

on-chip dual-access RAM (DARAM)

The ’541 devices have a 5K-word × 16-bit on-chip DARAM (5 blocks of 1K-word each).

The ’542 and ’543 devices have a 10K-word × 16-bit on-chip DARAM (5 blocks of 2K-word each).

The ’545 and ’546 devices have a 6K-word × 16-bit on-chip DARAM (3 blocks of 2K-word each).

The ’548 and ’549 devices have a 8K-word × 16-bit on-chip DARAM (4 blocks of 2K-word each).

Each of these RAM blocks can be accessed twice per machine cycle. This memory is intended primarily to store data values; however, it can be used to store program as well. At reset, the DARAM is mapped into data memory space. DARAM can be mapped into program/data memory space by setting the OVLY bit in the PMST register.

on-chip single-access RAM (SARAM)

The ’548 and ’549 devices have a 24K word × 16 bit on-chip SARAM (three blocks of 8K words each).

Each of these SARAM blocks is a single-access memory. This memory is intended primarily to store data values; however, it can be used to store program as well. At reset, the SARAM is mapped into data memory space (2000h–7FFFh). SARAM can be mapped into program/data memory space by setting the OVLY bit in the PMST register.

on-chip memory security

The ’54x devices have a maskable option to protect the contents of on-chip memories. When the related bit is set, no externally originating instruction can access the on-chip memory spaces.

memory (continued)

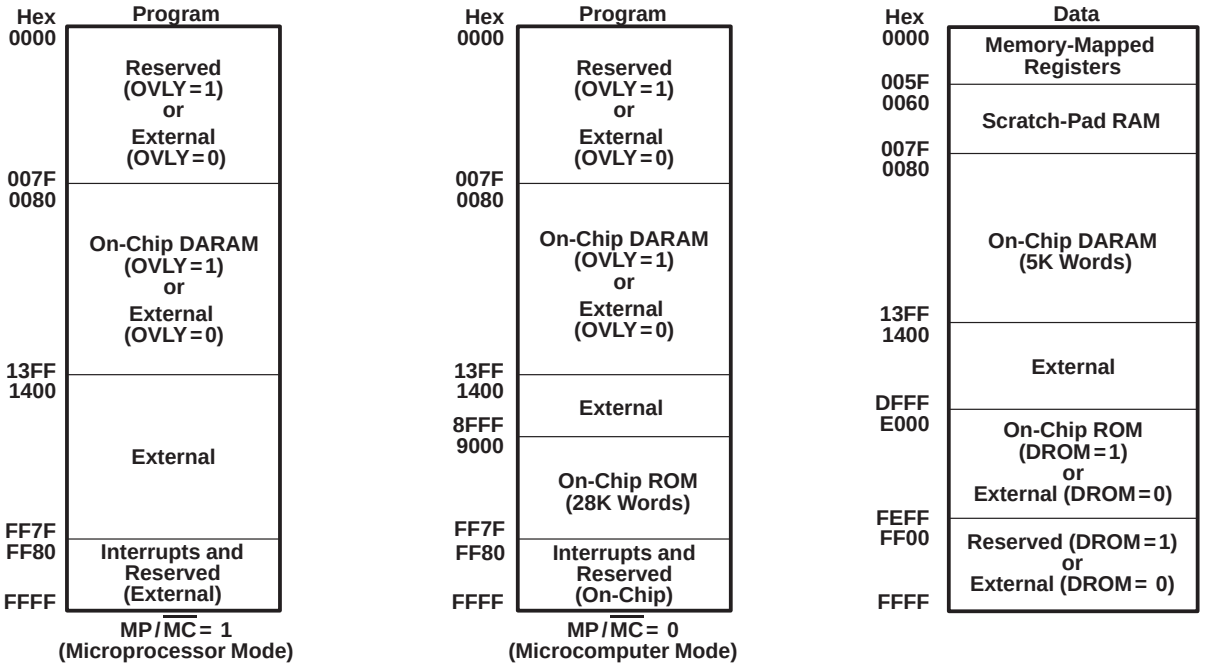


Figure 1. Memory Map (’541 only)

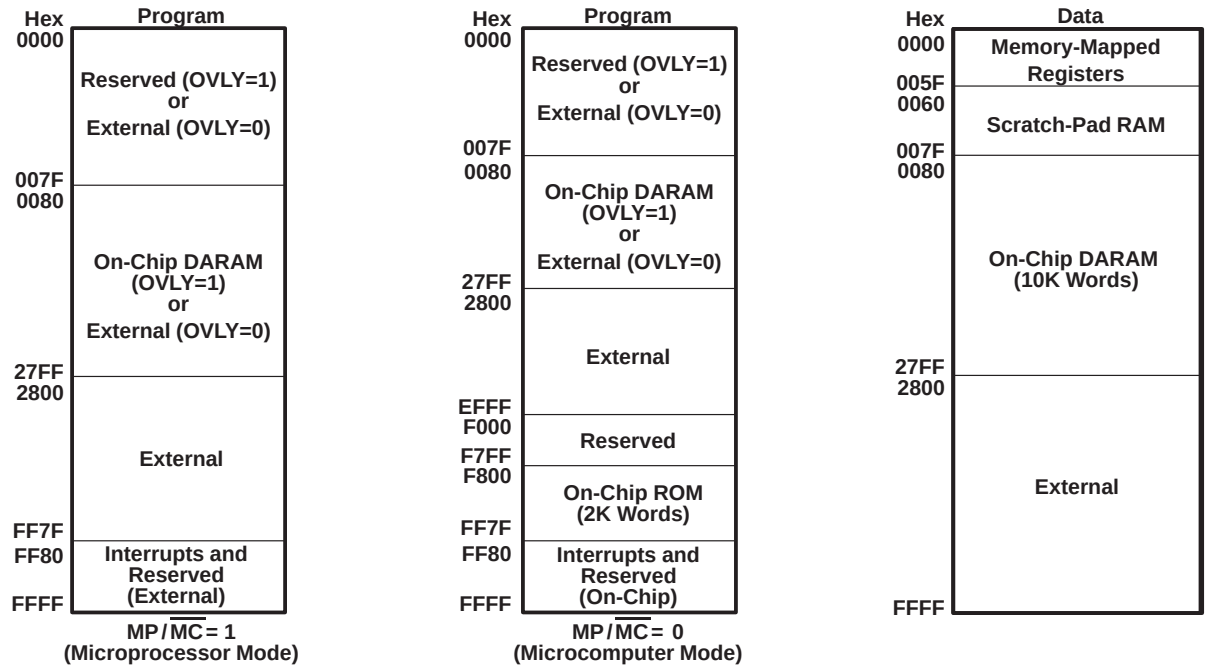


Figure 2. Memory Map (’542 and ’543 only)

ADVANCE INFORMATION

memory (continued)

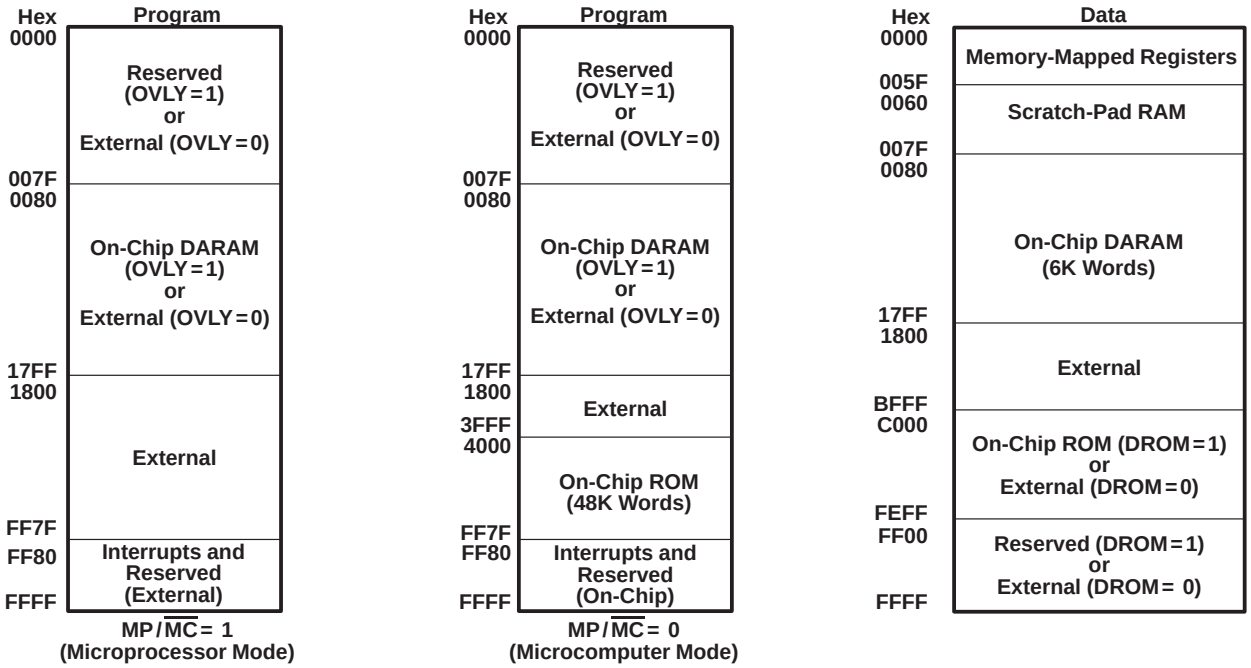


Figure 3. Memory Map (’545 and ’546 only)

ADVANCE INFORMATION

memory (continued)

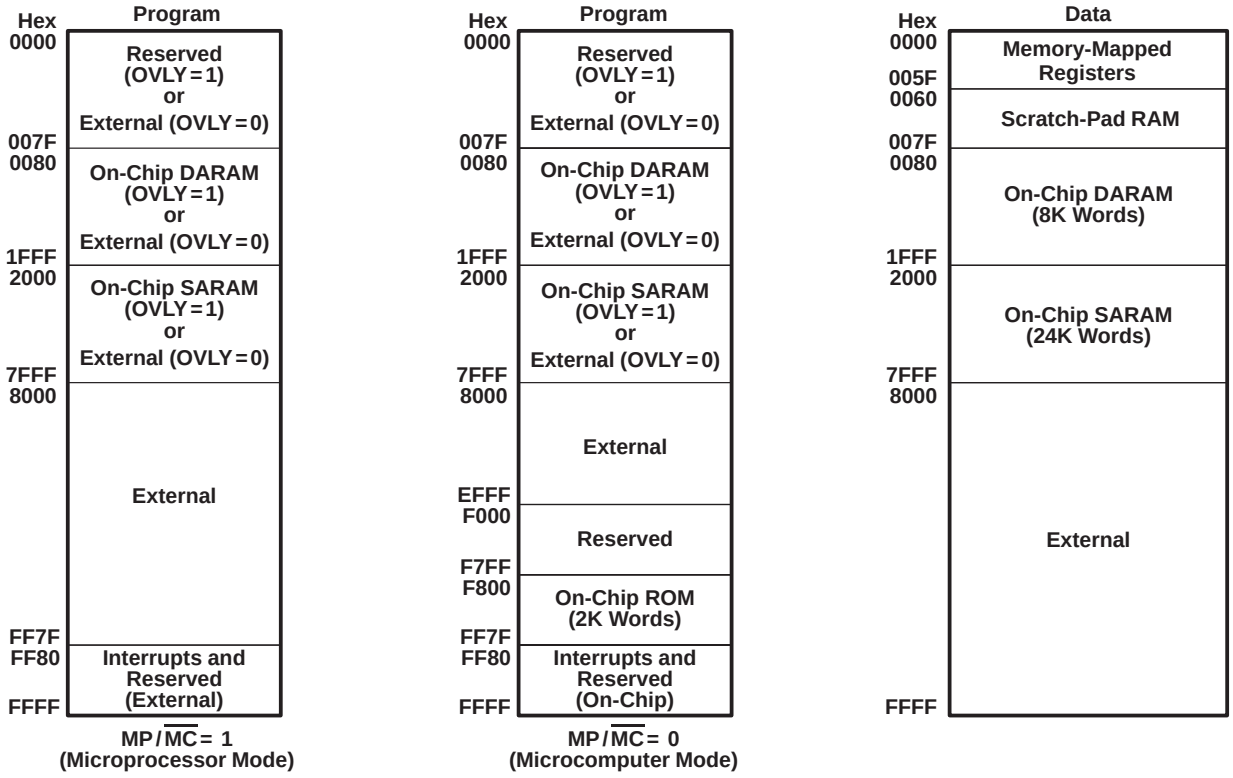


Figure 4. Memory Map (’548 only)
(In the case of a 64K Program Word Address Reach)

ADVANCE INFORMATION

memory (continued)

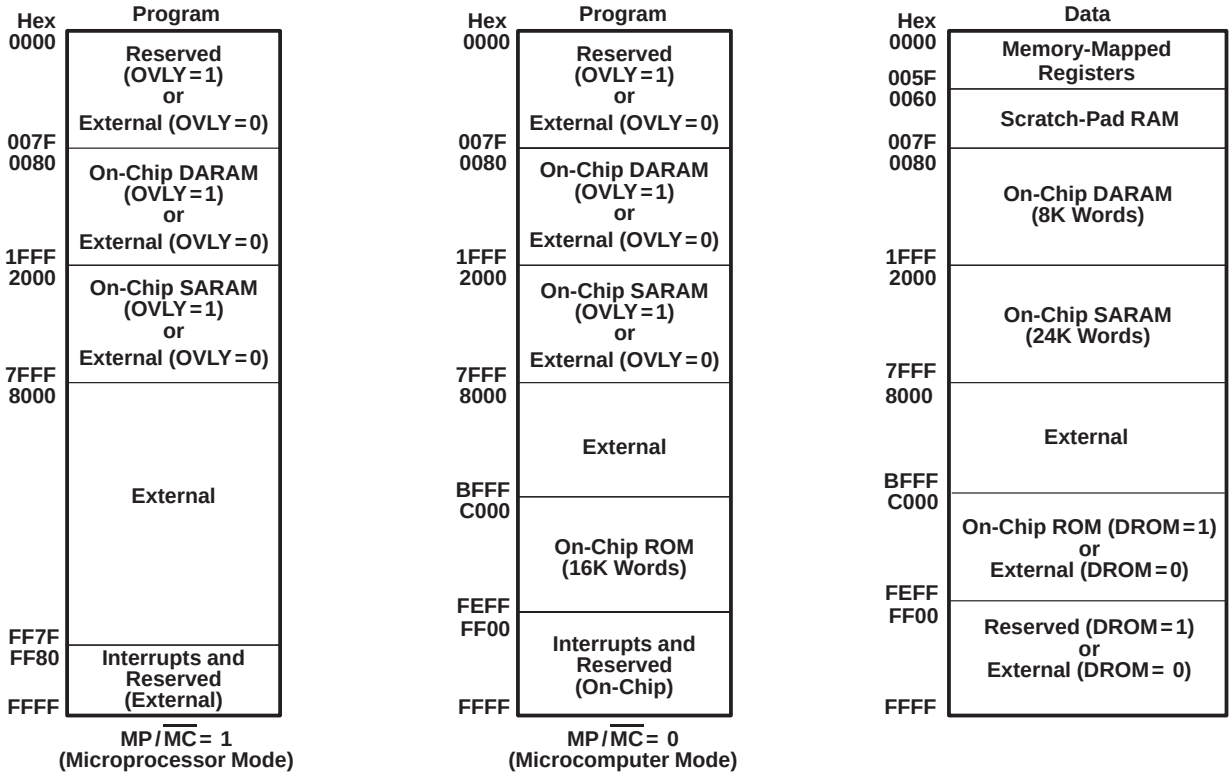
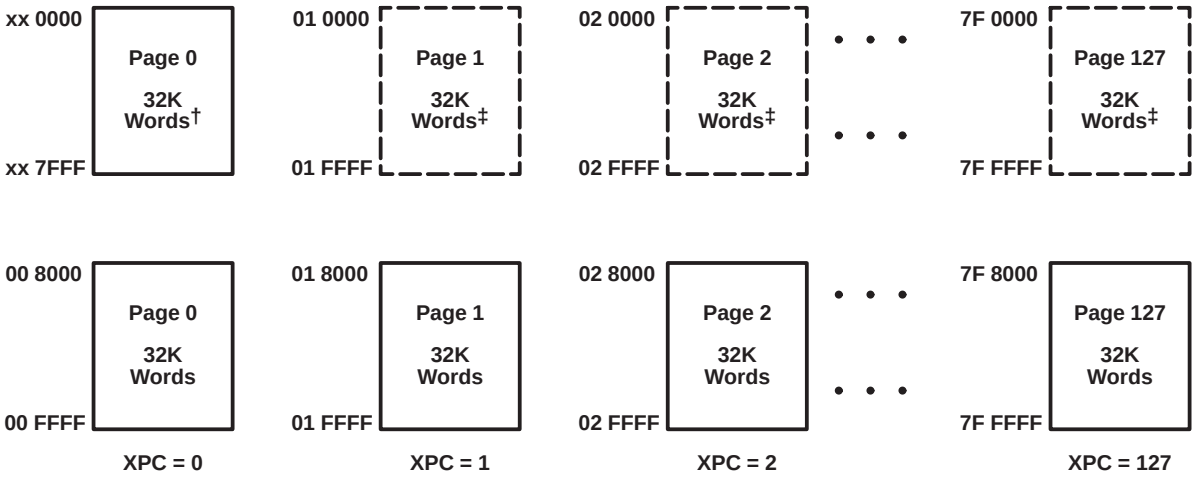


Figure 5. Memory Map (’549 only)



† See Figure 4 and Figure 5 for more information about this on-chip memory region.

‡ These pages available when OVLY = 0 when on–chip RAM is not mapped in program space or data space. When OVLY = 1 the first 32K words are all on page 0 when on–chip RAM is mapped in program space or data space.

NOTE A: When the on-chip RAM is enabled in program space, all accesses to the region xx 0000 – xx 7FFF, regardless of page number, are mapped to the on-chip RAM at 00 0000 – 00 7FFF.

Figure 6. Extended Program Memory (’548 and ’549 only)

ADVANCE INFORMATION

program memory

The external program memory space on the ’54x devices addresses up to 64K 16-bit words. Software can configure their memory cells to reside inside or outside of the program address map. When the cells are mapped into program space, the device automatically accesses them when their addresses are within bounds. When the program-address generation (PAGEN) logic generates an address outside its bounds, the device automatically generates an external access. The advantages of operating from on-chip memory are as follows:

- Higher performance because no wait states are required
- Lower cost than external memory
- Lower power than external memory

The advantage of operating from off-chip memory is the ability to access a larger address space.

program memory address map

The reset, interrupt, and trap vectors are addressed in program space. These vectors are soft — meaning that the processor, when taking the trap, loads the program counter (PC) with the trap address and executes the code at the vector location. Four words are reserved at each vector location to accommodate a delayed branch instruction, and either two 1-word instructions or one 2-word instruction, which allows branching to the appropriate interrupt service routine without the overhead.

At device reset, the reset, interrupt, and trap vectors are mapped to address FF80h in program space. However, these vectors can be remapped to the beginning of any 128-word page in program space after device reset. This is done by loading the interrupt vector pointer (IPTR) bits in the PMST register with the appropriate 128-word page boundary address. After loading IPTR, any user interrupt or trap vector is mapped to the new 128-word page. For example:

```
STM    #05800h,PMST     ;Remapped vectors to start at 5800h.
```

This example moves the interrupt vectors to program space at address 05800h. Any subsequent interrupt (except for a device reset) fetches its interrupt vector from that new location. For example, if, after loading the IPTR, an $\overline{\text{INT2}}$ occurs, the interrupt service routine vector is fetched from location 5848h in program space as opposed to location FFC8h. This feature facilitates moving the desired vectors out of the boot ROM and then removing the ROM from the memory map. Once the system code is booted into the system from the boot-loader code resident in ROM, the application reloads the IPTR with a value pointing to the new vectors. In the previous example, the STM instruction is used to modify the PMST. Note that the STM instruction modifies not only the IPTR but other status/control bits in the PMST register.

NOTE: The hardware reset ($\overline{\text{RS}}$) vector cannot be remapped, because the hardware reset loads the IPTR with 1s. Therefore, the reset vector is always fetched at location FF80h in program space. In addition, for the ’54x, 128 words are reserved in the on-chip ROM for device-testing purposes. Application code written to be implemented in on-chip ROM must reserve these 128 words at addresses FF00h–FF7Fh in program space.

extended program memory (’548 and ’549 only)

The ’548 and ’549 devices use a paged extended memory scheme in program space to allow access of up to 8M of program memory. This extended program memory is organized into 128 pages (0–127), each 64K in length. To implement the extended program memory scheme, the ’548 and ’549 device includes the following additional features:

- Seven additional address lines (for a total of 23)
- An extra memory-mapped register [program counter extension register (XPC)]

extended program memory (’548 and ’549 only) (continued)

- Six new instructions for addressing extended program memory space:
  - FB[D] — Far branch
  - FBACC[D] — Far branch to the location specified by the value in accumulator A or accumulator B
  - FCALA[D] — Far call to the location specified by the value in accumulator A or accumulator B
  - FCALL[D] — Far call
  - FRET[D] — Far return
  - FRETE[D] — Far return with interrupts enabled
- Two ’54x instructions are extended to use the 23 bits in the ’548 and ’549 devices:
  - READA — Read program memory addressed by accumulator A and store in data memory
  - WRITA — Write data to program memory addressed by accumulator A

For more information on these six new instructions and the two extended instructions, refer to the instruction set summary table in this data sheet and to the *TMS320C54x DSP Reference Set, Volume 2, Mnemonic Instruction Set*, literature number SPRU172. And for more information on extended program memory, refer to the *TMS320C54x DSP Reference Set, Volume 1, CPU and Peripherals*, literature number SPRU131.

data memory

The data memory space on the ’54x device addresses contains up to 64K of 16-bit words. The ’devices automatically access the on-chip RAM when addressing within its bounds. When an address is generated outside the RAM bounds, the device automatically generates an external access.

The advantages of operating from on-chip memory are as follows:

- Higher performance because no wait states are required
- Higher performance because of better flow within the pipeline of the CALU
- Lower cost than external memory
- Lower power than external memory

The advantage of operating from off-chip memory is the ability to access a larger address space.

bootloader

A bootloader is available in the standard ’54x on-chip ROM. This bootloader can be used to transfer user code from an external source to anywhere in the program memory at power up automatically. If MP/$\overline{\text{MC}}$ of the device is sampled low during a hardware reset, execution begins at location FF80h of the on-chip ROM. This location contains a branch instruction to the start of the boot-loader program. The standard ’54x devices provide different ways to download the code to accommodate various system requirements:

- Parallel from 8-bit or 16-bit-wide EPROM
- Parallel from I/O space 8-bit or 16-bit mode
- Serial boot from serial ports 8-bit or 16-bit mode
- Host port interface boot (’542, ’545, ’548, and ’549 devices only)
- Warm boot

bootloader (continued)

The bootloader provided in the on-chip ROM of the ’548 and ’549 devices implements several enhanced features. These include the addition of BSP and TDM boot modes. To accommodate these new boot modes, the encoding of the boot-mode selection word has been modified.

For a detailed description of bootloader functionality, refer to the *TMS320C54x DSP Reference Set, Volume 4: Applications Guide* (literature number SPRU173). For a detailed description of the enhanced bootloader functionality, refer to the *TMS320x548/’549 Bootloader Technical Reference*.

on-chip peripherals

All the ’54x devices have the same CPU structure; however, they have different on-chip peripherals connected to their CPUs. The on-chip peripheral options provided are:

- Software-programmable wait-state generator
- Programmable bank switching
- Parallel I/O ports
- Serial ports (standard, TDM, and BSP)
- A hardware timer
- A clock generator [with a multiple phase-locked loop (PLL) on ’549 devices]

software-programmable wait-state generators

Software-programmable wait-state generators can be used to extend external bus cycles up to seven machine cycles to interface with slower off-chip memory and I/O devices. The software wait-state generators are incorporated without any external hardware. For off-chip memory access, a number of wait states can be specified for every 32K-word block of program and data memory space, and for one 64K-word block of I/O space within the software wait-state (SWWSR) register.

programmable bank-switching

Programmable bank-switching can be used to insert one cycle automatically when crossing memory-bank boundaries inside program memory or data memory space. One cycle can also be inserted when crossing from program-memory space to data-memory space (’54x) or one program memory page to another program memory page (’548 and ’549 only). This extra cycle allows memory devices to release the bus before other devices start driving the bus; thereby avoiding bus contention. The size of memory bank for the bank-switching is defined by the bank-switching control register (BSCR).

parallel I/O ports

Each ’54x device has a total of 64K I/O ports. These ports can be addressed by the PORTR instruction or the PORTW instruction. The $\overline{IS}$ signal indicates a read/write operation through an I/O port. The devices can interface easily with external devices through the I/O ports while requiring minimal off-chip address-decoding circuits.

host-port interface (’542, ’545, ’548, and ’549 only)

The host-port interface (HPI) is an 8-bit parallel port used to interface a host processor to the DSP device. Information is exchanged between the DSP device and the host processor through on-chip memory that is accessible by both the host and the DSP device. The DSP devices have access to the HPI control (HPIC) register and the host can address the HPI memory through the HPI address register (HPIA). HPI memory is a 2K-word DARAM block that resides at 1000h to 17FFh in data memory and can also be used as general-purpose on-chip data or program DARAM.

host-port interface (’542, ’545, ’548, and ’549 only) (continued)

Data transfers of 16-bit words occur as two consecutive bytes with a dedicated pin (HBIL) indicating whether the high or low byte is being transmitted. Two control pins, HCNTL1 and HCNTL0, control host access to the HPIA, HPI data (with an optional automatic address increment), or the HPIC. The host can interrupt the DSP device by writing to HPIC. The DSP device can interrupt the host with a dedicated $\overline{\text{HINT}}$ pin that the host can acknowledge and clear.

The HPI has two modes of operation, shared-access mode (SAM) and host-only mode (HOM). In SAM, the normal mode of operation, both the DSP device and the host can access HPI memory. In this mode, asynchronous host accesses are resynchronized internally and, in case of conflict, the host has access priority and the DSP device waits one cycle. The HOM capability allows the host to access HPI memory while the DSP device is in IDLE2 (all internal clocks stopped) or in reset mode. The host can therefore access the HPI RAM while the DSP device is in its optimal configuration in terms of power consumption.

The HPI control register has two data strobes, $\overline{\text{HDS1}}$ and $\overline{\text{HDS2}}$, a read/write strobe HR/$\overline{\text{W}}$, and an address strobe $\overline{\text{HAS}}$, to enable a glueless interface to a variety of industry-standard host devices. The HPI is interfaced easily to hosts with multiplexed address/data bus, separate address and data buses, one data strobe and a read/write strobe, or two separate strobes for read and write.

The HPI supports high-speed back-to-back accesses.

- In the SAM, the HPI can handle one byte every five DSP device periods—that is, 64 MBps with a 40-MIPS DSP, or 160 MBps with a 100-MIPS DSP. The HPI is designed so that the host can take advantage of this high bandwidth and run at frequencies up to $(f \times n) \div 5$, where n is the number of host cycles for an external access and f is the DSP device frequency.
- In HOM, the HPI supports high-speed back-to-back host accesses at 1 byte every 50 ns—that is, 160 MBps with a -40 or faster DSP.

serial ports

The ’54x devices provide high-speed full-duplex serial ports that allow direct interface to other ’54x devices, codecs, and other devices in a system. There is a standard serial port, a time-division-multiplexed (TDM) serial port, and a buffered serial port (BSP). The ’549 devices provides a misalignment detection feature to that allows the device to detect when a word or words are lost in the serial data line.

The general-purpose serial port utilizes two memory-mapped registers for data transfer: the data-transmit register (DXR) and the data-receive register (DRR). Both of these registers can be accessed in the same manner as any other memory location. The transmit and receive sections of the serial port each have associated clocks, frame-synchronization pulses, and serial-shift registers; and serial data can be transferred either in bytes or in 16-bit words. Serial port receive and transmit operations can generate their own maskable transmit and receive interrupts (XINT and RINT), allowing serial-port transfers to be managed through software. The ’54x serial ports are double-buffered and fully static.

The TDM port allows the device to communicate through time-division multiplexing with up to seven other ’54x devices with TDM ports. Time-division multiplexing is the division of time intervals into a number of subintervals with each subinterval representing a prespecified communications channel. The TDM port serially transmits 16-bit words on a single data line (TDAT) and destination addresses on a single address line (TADD). Each device can transmit data on a single channel and receive data from one or more of the eight channels, providing a simple and efficient interface for multiprocessing applications. A frame synchronization pulse occurs once every 128 clock cycles, corresponding to the transmission of one 16-bit word on each of the eight channels. Like the general-purpose serial port, the TDM port is double-buffered on both input and output data.

The buffered serial port (BSP) consists of a full-duplex double-buffered serial-port interface and an auto-buffering unit (ABU). The serial port block of the BSP is an enhanced version of the standard serial port. The ABU allows the serial port to read/write directly to the ’54x internal memory using a dedicated bus independent of the CPU. This results in minimal overhead for serial port transactions and faster data rates.

serial ports (continued)

When auto-buffering capability is disabled (standard mode), serial port transfers are performed under software control through interrupts. In this mode, the ABU is transparent and the word-based interrupts (WXINT and WRINT) provided by the serial port are sent to the CPU as transmit interrupt (XINT) and receive interrupt (RINT). When auto buffering is enabled, word transfers are done directly between the serial port and the ’54x internal memory using ABU-embedded address generators.

The ABU has its own set of circular-addressing registers with corresponding address-generation units. Memory for the buffers resides in 2K words of the ’54x internal memory. The length and starting addresses of the buffers are user-programmable. A buffer-empty/buffer-full interrupt can be posted to the CPU. Buffering is easily halted by an auto-disabling capability. Auto-buffering capability can be enabled separately for transmit and receive sections. When auto buffering is disabled, operation is similar to that of the general-purpose serial port.

The BSP allows transfer of 8-, 10-, 12-, or 16-bit data packets. In burst mode, data packets are directed by a frame synchronization pulse for every packet. In continuous mode, the frame synchronization pulse occurs when the data transmission is initiated and no further pulses occur. The frame and clock strobes are frequency- and polarity-programmable. The BSP is fully static and operates at arbitrarily low clock frequencies. The maximum operating frequency for ’54x devices up to 50 MIPs is CLKOUT. For higher-speed ’54x devices, the maximum operating frequency is 50 MBps at 20 ns.

buffer misalignment (BMINT) interrupt (’549 only)

The BMINT interrupt is generated when a frame sync occurs and the ABU transmit or receive buffer pointer is not at the top of 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 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 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 buffer address, if the interface is functioning properly.

If BMINT is enabled under conditions other than those stated above, interrupts may be generated under circumstances other than actual buffer misalignment. In these cases, BMINT should generally be masked in the IMR register so that the processor will ignore this interrupt.

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

The BSP0 and BSP1 BMINT bits in the IMR and IFR registers are bits 12 and 13, respectively, (bit 15 is the MSB), and their interrupt vector locations are 070h and 074h, respectively.

serial ports (continued)

Table 3 provides a comparison of the serial ports available in the ’54x devices.

Table 3. Serial Port Configurations for the ’54x

DEVICE	NO. OF STANDARD SERIAL PORTS	NO. OF BSPs (BSP ADDRESS RANGES)	NO. OF TDM SERIAL PORTS
TMS320C541 TMS320LC541	2	–	–
TMS320C542 TMS320LC542	–	1 (0800h–0FFFh)	1
TMS320LC543	–	1 (0800h–0FFFh)	1
TMS320LC545 TMS320LC545A	1	1 (0800h–0FFFh)	–
TMS320LC546 TMS320LC546A	1	1 (0800h–0FFFh)	–
TMS320LC548	–	2 (0800h–0FFFh and 1800h–1FFFh)	1
TMS320LC549 TMS320VC549	–	2 (0800h–0FFFh and 1800h–1FFFh)	1

hardware timer

The ’54x devices feature a 16-bit timing circuit with a four-bit prescaler. The timer counter is decremented by one at every CLKOUT cycle. Each time the counter decrements to zero, a timer interrupt is generated. The timer can be stopped, restarted, reset, or disabled by specific status bits.

clock generator

The clock generator provides clocks to the ’54x device, and consists of an internal oscillator and a phase-locked loop (PLL) circuit. The clock generator requires a reference clock input, which can be provided by using a crystal resonator with the internal oscillator, or from an external clock source. The reference clock input is then either divided by two (or by four on the ’545A, ’546A, ’548, and ’549) to generate clocks for the ’54x device, or the PLL circuit can be used to generate the device clock by multiplying the reference clock frequency by a scale factor, allowing use of a clock source with a lower frequency than that of the CPU.

The PLL is an adaptive circuit that, once synchronized, locks onto and tracks an input clock signal. When the PLL is initially started, it enters a transitional mode during which the PLL acquires lock with the input signal. Once the PLL is locked, it continues to track and maintain synchronization with the input signal. Then, other internal clock circuitry allows the synthesis of new clock frequencies for use as master clock for the ’54x device.

Two types of PLL are available: a hardware-programmable PLL and a software-programmable PLL. All ’54x devices have the hardware-programmable PLL except the ’545A, ’546A, ’548, and ’549, which have the software-programmable PLL. On the hardware-programmable PLL, an external delay must be provided before the device is released from reset in order for the PLL to achieve lock. With the software-programmable PLL, a lock timer is provided to implement this delay automatically. Note that both the hardware- and the software-programmable PLLs require the device to be reset after power up to begin functioning properly.

hardware-programmable PLL

The ’54x can use either the internal oscillator or an external frequency source for an input clock. The clock generation mode is determined by the CLKMD1, CLKMD2 and CLKMD3 clock mode pins except on the ’545A, the ’546A, the ’548, and the ’549 (see software-programmable PLL description below). Table 4 outlines the selection of the clock mode by these pins. Note that both the hardware- and the software-programmable PLLs require the device to be reset after power up to begin functioning properly.

ADVANCE INFORMATION

hardware-programmable PLL (continued)

Table 4. Clock Mode Configurations

MODE-SELECT PINS			CLOCK MODE	
CLKMD1	CLKMD2	CLKMD3	OPTION 1†	OPTION 2†
0	0	0	PLL × 3 with external source	PLL × 5 with external source
1	1	0	PLL × 2 with external source	PLL × 4 with external source
1	0	0	PLL × 3, internal oscillator enabled	PLL × 5, internal oscillator enabled
0	1	0	PLL × 1.5 with external source	PLL × 4.5 with external source
0	0	1	Divide-by-two with external source	Divide-by-two with external source
0	1	1	Stop mode‡	Stop mode‡
1	0	1	PLL × 1 with external source	PLL × 1 with external source
1	1	1	Divide-by-two, internal oscillator enabled	Divide-by-two, internal oscillator enabled

† Option: Option 1 or option 2 is selected when ordering the device.
‡ Stop mode: The function of the stop mode is equivalent to that of the power-down mode of IDLE3; however, the IDLE3 instruction is recommended rather than stop mode to realize full power saving, since IDLE3 stops clocks synchronously and can be exited with an interrupt.

software-programmable PLL ('545A, '546A, '548, and '549)

The software-programmable PLL features a high level of flexibility, and includes a clock scaler that provides various clock multiplier ratios, capability to directly enable and disable the PLL, and a PLL lock timer that can be used to delay switching to PLL clocking mode of the device until lock is achieved.

Devices that have a built-in software-programmable PLL can be configured in one of two clock modes:

- PLL mode. The input clock (X2/CLKIN) is multiplied by 1 of 31 possible ratios. These ratios are achieved using the PLL circuitry.
- DIV (divider) mode. The input clock is divided by 2 or 4. Note that when DIV mode is used, the PLL can be completely disabled in order to minimize power dissipation.

The software-programmable PLL is controlled using the 16-bit memory-mapped (address 0058h) clock mode register (CLKMD). The CLKMD register is used to define the clock configuration of the PLL clock module. The CLKMD register fields are shown in Figure 7 and described below. Note that upon reset, the CLKMD register is initialized with a predetermined value dependent only upon the state of the CLKMD1 – CLKMD3 pins (see Table 6).

Bit #	15–12	11	10–3	2	1	0
	PLLMUL	PLLDIV	PLLCOUNT	PLLON/OFF	PLLNDIV	PLLSTATUS
	R/W†	R/W†	R/W†	R/W†	R/W	R

R = read, W = write

† When in DIV mode (PLLSTATUS is low), PLLMUL, PLLDIV, PLLCOUNT, and PLLON/OFF are don't cares, and their contents are indeterminate.

Figure 7. Clock Mode Control Register (CLKMD)

ADVANCE INFORMATION

software-programmable PLL ('545A, '546A, '548, and '549) (continued)

Bits 15–12 PLLMUL. PLL multiplier. Defines the frequency multiplier in conjunction with PLLDIV and PLLNDIV, as shown in Table 5.

Bit 11 PLLDIV. PLL divider. Defines the frequency multiplier in conjunction with PLLMUL and PLLNDIV, as shown in Table 5.

0 = an integer multiply factor is used.

1 = a non-integer multiply factor is used.

Bits 10–3 PLLCOUNT. PLL counter value. Specifies the number of input clock cycles (in increments of 16 cycles) for the PLL lock timer to count before the PLL begins clocking the processor after the PLL is started. The PLL counter is a down-counter, which is driven by the input clock divided by 16; therefore, for every 16 input clocks, the PLL counter decrements by one.

The PLL counter can be used to ensure that the processor is not clocked until the PLL is locked, so that only valid clock signals are sent to the device.

Bit 2 PLLON/OFF. PLL on/off. Enables or disables the PLL part of the clock generator in conjunction with the PLLNDIV bit. Note that PLLON/OFF and PLLNDIV can both force the PLL to run; when PLLON/OFF is high, the PLL runs independently of the state of PLLNDIV.

PLLON/OFF	PLLNDIV	PLL STATE
0	0	Off
1	0	On
0	1	On
1	1	On

Bit 1 PLLNDIV. PLL clock generator select. Determines whether the clock generator works in PLL mode or in divider (DIV) mode, thereby defining the frequency multiplier in conjunction with PLLMUL and PLLDIV.

0 = Divider mode is used

1 = PLL mode is used

Bit 0 PLLSTATUS. PLL status. Indicates the mode in which the clock generator is operating.

0 = DIV mode

1 = PLL mode

Table 5. PLL Multiplier Ratio as a Function of PLLNDIV, PLLDIV, and PLLMUL

PLLNDIV	PLLDIV	PLLMUL	MULTIPLIER†
0	x	0–14	0.5
0	x	15	0.25
1	0	0–14	PLLMUL + 1
1	0	15	1
1	1	0 or even	(PLLMUL + 1) ÷ 2
1	1	odd	PLLMUL ÷ 4

† CLKOUT = CLKIN x multiplier

ADVANCE INFORMATION

software-programmable PLL ('545A, '546A, '548, and '549) (continued)

Immediately following reset, the clock mode is determined by the values of the three external pins: CLKMD1, CLKMD2, and CLKMD3. The modes corresponding to the CLKMD pins are shown in Table 6.

Table 6. Clock Mode Settings at Reset

CLKMD1	CLKMD2	CLKMD3	CLKMD REGISTER RESET VALUE	CLOCK MODE
0	0	0	0000h	Divide-by-two, with external source
0	0	1	1000h	Divide-by-two, with external source
0	1	0	2000h	Divide-by-two, with external source
1	0	0	4000h	Divide-by-two, internal oscillator enabled
1	1	0	6000h	Divide-by-two, with external source
1	1	1	7000h	Divide-by-two, internal oscillator enabled†
1	0	1	0007h	PLL × 1 with external source
0	1	1	—	Stop mode

† Reserved mode ('549 only). Do not use in normal operation.

Following reset, the software-programmable PLL can be programmed to any configuration desired, as described above. Note that when the PLL × 1 with external source option (CLKMD[1–3]=101) is selected during reset, the internal PLL lock-count timer is not active; therefore, the system must delay releasing reset in order to allow for the PLL lock-time delay. Also, note that both the hardware- and the software-programmable PLLs require the device to be reset after power up to begin functioning properly.

programming considerations when using the software-programmable PLL

The software-programmable PLL offers many different options in startup configurations, operating modes, and power-saving features. Programming considerations and several software examples are presented here to illustrate the proper use of the software-programmable PLL at start-up, when switching between different clocking modes, and before and after IDLE1/IDLE2/IDLE3 instruction execution.

use of the PLLCOUNT programmable lock timer

During the lockup period, the PLL should not be used to clock the '54x. The PLLCOUNT programmable lock timer provides a convenient method of automatically delaying clocking of the device by the PLL until lock is achieved.

The PLL lock timer is a counter, loaded from the PLLCOUNT field in the CLKMD register, that decrements from its preset value to 0. The timer can be preset to any value from 0 to 255, and its input clock is CLKIN divided by 16. The resulting lockup delay can therefore be set from 0 to 255 × 16 CLKIN cycles.

The lock timer is activated when the clock generator operating mode is switched from DIV to PLL (see the section describing switching from DIV mode to PLL mode). During the lockup period, the clock generator continues to operate in DIV mode; after the PLL lock timer has decremented to zero, the PLL begins clocking the '54x.

Accordingly, the value loaded into PLLCOUNT is chosen based on the following relationship:

$$\text{PLLCOUNT} > \text{Lockup Time} / (16 \times t_{CLKIN})$$

where t_{CLKIN} is the input reference clock period and lockup time is the required PLL lockup time as shown in Figure 8.

ADVANCE INFORMATION

use of the PLLCOUNT programmable lock timer (continued)

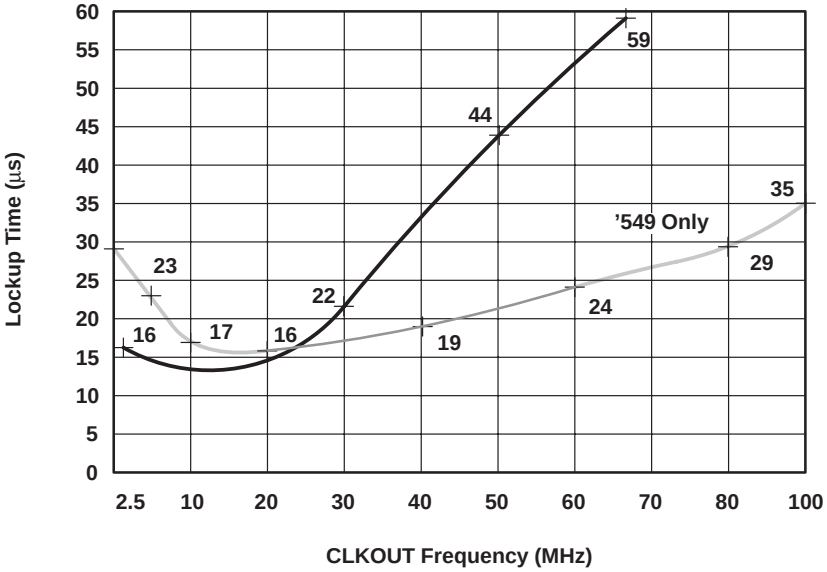


Figure 8. PLL Lockup Time Versus CLKOUT Frequency

switching from DIV mode to PLL mode

Several circumstances may require switching from DIV mode to PLL mode; however, note that if the PLL is not locked when switching from DIV mode to PLL mode, the PLL lockup time delay must be observed before the mode switch occurs to ensure that only proper clock signals are sent to the device. It is, therefore, important to know whether or not the PLL is locked when switching operating modes.

The PLL is unlocked on power-up, after changing the PLLMUL or PLLDIV values, after turning off the PLL (PLLON/OFF = 0), or after loss of input reference clock. Once locked, the PLL remains locked even in DIV mode as long as the PLL had been previously locked and has not been turned off (PLLON/OFF stays 1), and the PLLMUL and PLLDIV values have not been changed since the PLL was locked.

Switching from DIV mode to PLL mode (setting PLLNDIV to 1) activates the PLLCOUNT programmable lock timer (when PLLCOUNT is preloaded with a non-zero value), and this can be used to provide a convenient method for implementing the lockup time delay. The PLLCOUNT lock timer feature should be used in the situations described above, where the PLL is unlocked unless a reset delay is used to implement the lockup delay, or the PLL is not used.

Switching from DIV mode to PLL mode is accomplished by loading the CLKMD register. The following procedure describes switching from DIV mode to PLL mode when the PLL is not locked. When performing this mode switch with the PLL already locked, the effect is the same as when switching from PLL to DIV mode, but in the reverse order. In this case, the delays of when the new clock mode takes effect are the same.

When switching from DIV to PLL mode with the PLL unlocked, or when the mode change will result in unlocked operation, the PLLMUL[3–0], PLLDIV, and PLLNDIV bits are set to select the desired frequency multiplier as described in Table 5, and the PLLCOUNT[7–0] bits are set to select the required lockup time delay. Note that PLLMUL, PLLDIV, PLLCOUNT, and PLLON/OFF can only be modified when in DIV mode.

switching from DIV mode to PLL mode (continued)

Once the PLLNDIV bit is set, the PLLCOUNT timer begins being decremented from its preset value. When the PLLCOUNT timer reaches zero, the switch to PLL mode takes effect after six CLKIN cycles plus 3.5 PLL cycles (CLKOUT frequency). When the switch to PLL mode is completed, the PLLSTATUS bit in the CLKMD register is read as 1. Note that during the PLL lockup period, the ’54x continues operating in DIV mode.

The following software example shows an instruction that can be used to switch from DIV mode to PLL × 3, with a CLKIN frequency of 13 MHz and PLLCOUNT = 41 (decimal).

```
        STM     #0010000101001111b, CLKMD
```

switching clock mode from PLL to DIV

When switching from PLL mode to DIV mode, the PLLCOUNT delay does not occur, and the switch between the two modes takes place after a short transition delay.

The switch from PLL mode to DIV mode is also accomplished by loading the CLKMD register. The PLLNDIV bit is set to 0, selecting DIV mode, and the PLLMUL bits are set to select the desired frequency multiplier as shown in Table 5.

The switch to DIV mode takes effect in 6 CLKIN cycles plus 3.5 PLL cycles (CLKOUT frequency) for all PLLMUL values except 1111b. With a PLLMUL value of 1111b, the switch to DIV mode takes effect in 12 CLKIN cycles plus 3.5 PLL cycles (CLKOUT frequency). When the switch to DIV mode is completed, the PLLSTATUS bit in the CLKMD register is read as 0.

The following software example shows a code sequence that can be used to switch from PLL × 3 to divide-by-two mode. Note that the PLLSTATUS bit is polled to determine when the switch to DIV mode has taken effect, and then the STM instruction is used to turn off the PLL at this point.

```
        STM     #0b, CLKMD              ;switch to DIV mode
TstStatu: LDM   CLKMD, A
        AND     #01b, A                 ;poll STATUS bit
        BC      TstStatu, ANEQ
        STM     #0b, CLKMD              ;reset PLLON_OFF when STATUS
                                        ;is DIV mode
```

switching mode from one PLL multiplier to another

When switching from one PLL multiplier ratio to another is required, the clock generator must be switched from PLL mode to DIV mode before selecting the new multiplier ratio; switching directly from one PLL multiplier ratio to another is not supported.

In order to switch from one PLL multiplier ratio to another, the following steps must be followed:

1. Set the PLLNDIV bit to 0, selecting DIV mode.
2. Poll the PLLSTATUS bit until a 0 is obtained, indicating that DIV mode is enabled and that PLLMUL, PLLDIV, and PLLCOUNT can be updated.
3. Modify the CLKMD register to set the PLLMUL[3–0], PLLDIV, and PLLNDIV bits to the desired frequency multiplier as defined in Table 5, and the PLLCOUNT[7–0] bits to the required lock-up time.

When the PLLNDIV bit is set to one in step three, the PLLCOUNT timer begins decrementing from its preset value. Once the PLLCOUNT timer reaches zero, the new PLL mode takes effect after six CLKIN cycles plus 3.5 PLL cycles (CLKOUT frequency).

switching mode from one PLL multiplier to another (continued)

Also, note that a direct switch between divide-by-two mode and divide-by-four mode is not possible. To switch between these two modes, the clock generator must first be set to PLL mode with an integer-only (non-fractional) multiplier ratio, and then set back to DIV mode in the desired divider configuration (see previous sections for details on switching between DIV and PLL modes).

The following software example shows a code sequence that can be used to switch clock mode from PLL × X to PLL × 1.

```
            STM     #0b, CLKMD                  ;switch to DIV mode
TstStatu:   LDM     CLKMD, A
            AND     #01b, A                     ;poll STATUS bit
            BC      TstStatu, ANEQ
            STM     #0000001111101111b, CLKMD   ;switch to PLL × 1 mode
```

programmable clock generator operation immediately following reset

Immediately following reset, the operating mode of the clock generator is determined only on the basis of the CLKMD1/2/3 pin state as described in Table 6. All but two of these operating modes are 'divide-by-two with external source'. Switching from divide-by-two to a PLL mode can easily be accomplished by changing the CLKMD register contents. Note that if use of the internal oscillator is desired, either the 100 or the 111 state of the CLKMD1–CLKMD3 pins must be selected at reset (as shown in Table 6) since the internal oscillator cannot be programmed through software.

The following software example shows an instruction that can be used to switch from divide-by-two mode to the PLL × 3 mode.

```
            STM     #0010000101001111b, CLKMD
```

considerations when using IDLE1/IDLE2/IDLE3

When using one of the IDLE instructions to reduce power requirements, proper management of the PLL is important. The clock generator consumes the least power when operating in DIV mode with the PLL disabled. Therefore, if power dissipation is a significant consideration, it is desirable to switch from PLL to DIV mode, and disable the PLL, before executing the IDLE1/IDLE2/IDLE3 instructions. This is accomplished as explained above in the section describing switching clock mode from PLL to DIV. After waking up from IDLE1/IDLE2/IDLE3, the clock generator can be reprogrammed to PLL mode as explained above in the section describing switching clock mode from DIV to PLL.

Note that when the PLL is stopped during an IDLE state, and the '54x device is restarted and the clock generator is switched back to PLL mode, the PLL lockup delay occurs in the same manner as in a normal device startup. Therefore, in this case, the lockup delay must also be accounted for, either externally or by using the PLL lockup counter timer.

The following software example illustrates a code sequence that switches the clock generator from PLL × 3 mode to divide-by-two mode, turns off the PLL, and enters IDLE3. After waking up from IDLE3, the clock generator is switched back from DIV mode to PLL × 3 mode using a single STM instruction, with a PLLCOUNT of 64 (decimal) used for the lock timer value.

considerations when using IDLE1/IDLE2/IDLE3 (continued)

```
                STM     #0b, CLKMD                  ;switch to DIV mode
TstStatu:       LDM     CLKMD, A
                AND     #01b, A                     ;poll STATUS bit
                BC      TstStatu, ANEQ
                STM     #0b, CLKMD                  ;reset PLLON_OFF when STATUS
                                                    ;is DIV mode
                IDLE3

                (After IDLE3 wake-up – switch the PLL from DIV mode to PLL × 3 mode)

                STM     #0010001000000111b, CLKMD   ;PLLCOUNT = 64 (decimal)
```

PLL considerations when using the bootloader

The ROM on the ’545A and ’546A contains a bootloader program that can be used to load programs into RAM for execution following reset. When using this bootloader with the software-programmable PLL, several considerations are important for proper system operation.

On the ’545A and ’546A, for compatibility, the bootloader configures the PLL to the same mode as would have resulted if the same CLKMD1–3 input bits had been provided to the option-1 or option-2 hardware-programmable PLL (see Table 4), according to whether the ’545A or ’546A is an option-1 or option-2 device. Once the bootloader program has finished executing, and control is transferred to the user’s program, the PLL can be reprogrammed to any desired configuration.

memory-mapped registers

Most ’54x devices have 26 (except ’548 and ’549 have 27) memory-mapped CPU registers, which are mapped into data memory located at addresses 0h to 1Fh. Each of these devices also has a set of memory-mapped registers associated with peripherals. Table 7 gives a list of CPU memory-mapped registers (MMR) common to all ’54x devices. Table 8 shows additional peripheral MMRs associated with the ’541 devices, Table 9 shows those associated with the ’545/’546 devices, Table 10 shows those associated with the ’542/’543 devices, and Table 11 shows those associated with the ’548/’549 devices.

Table 7. Core Processor Memory-Mapped Registers

NAME	ADDRESS		DESCRIPTION
	DEC	HEX	
IMR	0	0	Interrupt mask register
IFR	1	1	Interrupt flag register
–	2–5	2–5	Reserved for testing
ST0	6	6	Status register 0
ST1	7	7	Status register 1
AL	8	8	Accumulator A low word (15–0)
AH	9	9	Accumulator A high word (31–16)
AG	10	A	Accumulator A guard bits (39–32)
BL	11	B	Accumulator B low word (15–0)
BH	12	C	Accumulator B high word (31–16)
BG	13	D	Accumulator B guard bits (39–32)
TREG	14	E	Temporary register
TRN	15	F	Transition register
AR0	16	10	Auxiliary register 0
AR1	17	11	Auxiliary register 1
AR2	18	12	Auxiliary register 2
AR3	19	13	Auxiliary register 3
AR4	20	14	Auxiliary register 4
AR5	21	15	Auxiliary register 5
AR6	22	16	Auxiliary register 6
AR7	23	17	Auxiliary register 7
SP	24	18	Stack pointer register
BK	25	19	Circular buffer size register
BRC	26	1A	Block-repeat counter
RSA	27	1B	Block-repeat start address
REA	28	1C	Block-repeat end address
PMST	29	1D	Processor mode status (PMST) register
XPC	30	1E	Extended program counter (’548 and ’549 only)
–	31	1F	Reserved

memory-mapped registers (continued)

Table 8. Peripheral Memory-Mapped Registers ('541 Only)

NAME	ADDRESS		DESCRIPTION
	DEC	HEX	
DRR0	32	20	Serial port 0 data-receive register
DXR0	33	21	Serial port 0 data-transmit register
SPC0	34	22	Serial port 0 control register
—	35	23	Reserved
TIM	36	24	Timer register
PRD	37	25	Timer period register
TCR	38	26	Timer control register
—	39	27	Reserved
SWWSR	40	28	S/W wait-state register
BSCR	41	29	Bank-switching control register
—	42–47	2A–2F	Reserved
DRR1	48	30	Serial port 1 data-receive register
DXR1	49	31	Serial port 1 data-transmit register
SPC1	50	32	Serial port 1 control register
—	51	33	Reserved
—	52–95	34–5F	Reserved

ADVANCE INFORMATION

memory-mapped registers (continued)

Table 9. Peripheral Memory-Mapped Registers (’545 and ’546 Only)†

NAME	ADDRESS		DESCRIPTION
	DEC	HEX	
BDRR	32	20	BSP data-receive register
BDXR	33	21	BSP data-transmit register
BSPC	34	22	BSP serial-port control register
BSPCE	35	23	BSP control extension register
TIM	36	24	Timer register
PRD	37	25	Timer period counter
TCR	38	26	Timer control register
—	39	27	Reserved
SWWSR	40	28	External bus S/W wait-state register
BSCR	41	29	External bus bank-switching control register
—	42 – 43	2A – 2B	Reserved
HPIC	44	2C	HPI control register‡
—	45 – 47	2D – 2F	Reserved
DRR	48	30	Data-receive register
DXR	49	31	Data-transmit register
SPC	50	32	Serial-port control register
—	51 – 55	33 – 37	Reserved
AXR	56	38	BSP ABU transmit-address register
BKX	57	39	BSP ABU transmit-buffer-size register
ARR	58	3A	BSP ABU receive-address register
BKR	59	3B	BSP ABU receive-buffer-size register

† BSP = Buffered serial port
ABU = Auto-buffering unit
‡ Host port interface (HPI) on ’LC545 only

ADVANCE INFORMATION

memory-mapped registers (continued)

Table 10. Peripheral Memory-Mapped Registers ('542 and '543 Only)†

NAME	ADDRESS		DESCRIPTION
	DEC	HEX	
BDRR	32	20	BSP data-receive register
BDXR	33	21	BSP data-transmit register
BSPC	34	22	BSP serial-port control register
BSPCE	35	23	BSP control extension register
TIM	36	24	Timer register
PRD	37	25	Timer period counter
TCR	38	26	Timer control register
—	39	27	Reserved
SWWSR	40	28	External bus S/W wait-state register
BSCR	41	29	External bus bank-switching control register
—	42 – 43	2A – 2B	Reserved
HPIC	44	2C	HPI control register‡
—	45 – 47	2D – 2F	Reserved
TRCV	48	30	TDM data-receive register
TDXR	49	31	TDM data-transmit register
TSPC	50	32	TDM serial-port control register
TCSR	51	33	TDM channel-select register
TRTA	52	34	TDM receive/transmit register
TRAD	53	35	TDM receive address register
—	54 – 55	36 – 37	Reserved
AXR	56	38	BSP ABU transmit-address register
BKX	57	39	BSP ABU transmit-buffer-size register
ARR	58	3A	BSP ABU receive-address register
BKR	59	3B	BSP ABU receive-buffer-size register

† BSP = Buffered serial port
TDM = Time-division multiplexed
ABU = Auto-buffering unit
‡ Host port interface (HPI) on '542 only

ADVANCE INFORMATION

memory-mapped registers (continued)

Table 11. Peripheral Memory-Mapped Registers (’548 and ’549 Only)†

NAME	ADDRESS		DESCRIPTION
	DEC	HEX	
BDRR0	32	20	BSP 0 data-receive register
BDXR0	33	21	BSP 0 data-transmit register
BSPC0	34	22	BSP 0 control register
BSPCE0	35	23	BSP 0 control extension register
TIM	36	24	Timer count register
PRD	37	25	Timer period register
TCR	38	26	Timer control register
—	39	27	Reserved
SWWSR	40	28	External interface software wait-state register
BSCR	41	29	External interface bank-switching control register
—	42	2A	Reserved
—	43	2B	Reserved
HPIC	44	2C	HPI control register
—	45–47	2D–2F	Reserved
TRCV	48	30	TDM port data-receive register
TDXR	49	31	TDM port data-transmit register
TSPC	50	32	TDM serial port control register
TCSR	51	33	TDM channel-select register
TRTA	52	34	TDM receive/transmit register
TRAD	53	35	TDM receive/address register
—	54–55	36–37	Reserved
AXR0	56	38	ABU 0 transmit-address register
BKX0	57	39	ABU 0 transmit-buffer-size register
ARR0	58	3A	ABU 0 receive-address register
BKR0	59	3B	ABU 0 receive-buffer-size register
AXR1	60	3C	ABU 1 transmit-address register
BKX1	61	3D	ABU 1 transmit-buffer-size register
ARR1	62	3E	ABU 1 receive-address register
BKR1	63	3F	ABU 1 receive-buffer-size register
BDRR1	64	40	BSP 1 data-receive register
BDXR1	65	41	BSP 1 data-transmit register
BSPC1	66	42	BSP 1 control register
BSPCE1	67	43	BSP 1 control extension register
—	68–87	44–57	Reserved
CLKMD	88	58	Clock mode register
—	89–95	59–5F	Reserved

† BSP = Buffered serial port
ABU = Auto-buffering unit
HPI = Host-port interface

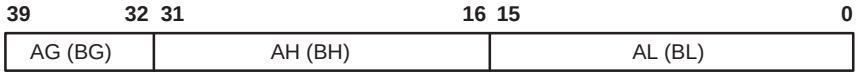
ADVANCE INFORMATION

status registers (ST0, ST1)

The status registers, ST0 and ST1, contain the status of the various conditions and modes for the ’54x devices. ST0 contains the flags (OV, C, and TC) produced by arithmetic operations and bit manipulations in addition to the data page pointer (DP) and the auxiliary register pointer (ARP) fields. ST1 contains the various modes and instructions that the processor operates on and executes.

accumulators (AL, AH, AG, and BL, BH, BG)

The ’54x devices have two 40-bit accumulators: accumulator A and accumulator B. Each accumulator is memory-mapped and partitioned into accumulator low-word (AL, BL), accumulator high-word (AH, BH), and accumulator guard bits (AG, BG).



auxiliary registers (AR0–AR7)

The eight 16-bit auxiliary registers (AR0–AR7) can be accessed by the CALU and modified by the auxiliary register arithmetic units (ARAUs). The primary function of the auxiliary registers is generating 16-bit addresses for data space. However, these registers also can act as general-purpose registers or counters.

temporary register (TREG)

The TREG is used to hold one of the multiplicands for multiply and multiply/accumulate instructions. It can hold a dynamic (execution-time programmable) shift count for instructions with shift operation such as ADD, LD, and SUB instructions. It also can hold a dynamic bit address for the BITT instruction. The EXP instruction stores the exponent value computed into the TREG, while the NORM instruction uses the TREG value to normalize the number. For ACS operation of Viterbi decoding, TREG holds branch metrics used by the DADST and DSADT instructions.

transition register (TRN)

The TRN is a 16-bit register that is used to hold the transition decision for the path to new metrics to perform the Viterbi algorithm. The CMPS (compare, select, max, and store) instruction updates the contents of the TRN based on the comparison between the accumulator high word and the accumulator low word.

stack-pointer register (SP)

The SP is a 16-bit register that contains the address at the top of the system. The SP always points to the last element pushed onto the stack. The stack is manipulated by interrupts, traps, calls, returns, and the PUSHD, PSHM, POPD, and POPM instructions. Pushes and pops of the stack predecrement and postincrement, respectively, all 16 bits of the SP.

circular-buffer-size register (BK)

The 16-bit BK is used by the ARAUs in circular addressing to specify the data block size.

block repeat registers (BRC, RSA, REA)

The block-repeat counter (BRC) is a 16-bit register used to specify the number of times a block of code is to be repeated when performing a block repeat. The block-repeat start address (RSA) is a 16-bit register containing the starting address of the block of program memory to be repeated when operating in the repeat mode. The 16-bit block repeat-end address (REA) contains the ending address if the block of program memory is to be repeated when operating in the repeat mode.

interrupt registers (IMR, IFR)

The interrupt-mask register (IMR) is used to mask off specific interrupts individually at required times. The interrupt-flag register (IFR) indicates the current status of the interrupts.

processor-mode status register (PMST)

The processor-mode status register (PMST) controls memory configurations of the ’54x devices.

interrupts

Vector-relative locations and priorities for all internal and external interrupts are shown in Table 12.

Table 12. ’54x Interrupt Locations and Priorities

NAME	LOCATION		PRIORITY	FUNCTION
	DECIMAL	HEX		
$\overline{\text{RS}}$, SINTR	0	00	1	Reset (Hardware and software reset)
$\overline{\text{NMI}}$, SINT16	4	04	2	Nonmaskable interrupt
SINT17	8	08	–	Software interrupt #17
SINT18	12	0C	–	Software interrupt #18
SINT19	16	10	–	Software interrupt #19
SINT20	20	14	–	Software interrupt #20
SINT21	24	18	–	Software interrupt #21
SINT22	28	1C	–	Software interrupt #22
SINT23	32	20	–	Software interrupt #23
SINT24	36	24	–	Software interrupt #24
SINT25	40	28	–	Software interrupt #25
SINT26	44	2C	–	Software interrupt #26
SINT27	48	30	–	Software interrupt #27
SINT28	52	34	–	Software interrupt #28
SINT29	56	38	–	Software interrupt #29
SINT30	60	3C	–	Software interrupt #30
$\overline{\text{INT0}}$, SINT0	64	40	3	External user interrupt #0
$\overline{\text{INT1}}$, SINT1	68	44	4	External user interrupt #1
$\overline{\text{INT2}}$, SINT2	72	48	5	External user interrupt #2
TINT, SINT3	76	4C	6	External timer interrupt
BRINT0, SINT4	80	50	7	BSP #0 receive interrupt†
BXINT0, SINT5	84	54	8	BSP #0 transmit interrupt†
TRINT, SINT6	88	58	9	TDM receive interrupt‡
TRINT, SINT7	92	5C	10	TDM transmit interrupt‡
$\overline{\text{INT3}}$, SINT8	96	60	11	External user interrupt #3
$\overline{\text{HINT}}$, SINT9	100	64§	12	HPI interrupt (’542, ’545, ’548, ’549 only)
BRINT1, SINT10	104	68§	13	BSP #1 receive interrupt (’548, ’549 only)
BXINT1, SINT11	108	6C§	14	BSP #1 transmit interrupt (’548, ’549 only)
BMINT0, SINT12	112	70§	15	BSP #0 misalignment detection interrupt (’549 only)
BMINT1, SINT13	116	74§	16	BSP #1 misalignment detection interrupt (’549 only)
—	120–127	78–7F§	–	Reserved

† On ’541 devices, these interrupt locations are serial port 0 interrupts (RINT0/XINT0).
‡ On ’541, ’545, and ’546 devices, these interrupt locations are serial port 1 interrupts (RINT1/XINT1).
§ On ’541, ’543, and ’546 devices, interrupt locations 64h – 7Fh are reserved. On ’542 and ’545 devices, interrupt locations 68h – 7Fh are reserved. On ’548 devices, interrupt locations 70h – 7Fh are reserved.

ADVANCE INFORMATION

interrupts (continued)

The IFR and IMR registers are laid out as shown in Figure 9.

15–14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	BMINT1	BMINT0	BXINT1	BRINT1	HINT	$\overline{\text{INT3}}$	TXNT	TRNT	BXINT0	TRINT0	TINT	$\overline{\text{INT2}}$	$\overline{\text{INT1}}$	$\overline{\text{INT0}}$

Figure 9. IFR and IMR Registers

instruction set summary

This section summarizes the syntax used by the mnemonic assembler and the associated instruction set opcodes for the ’54x DSP devices (see Table 13). For detailed information on instruction operation, see the *TMS320C54x DSP Reference Set, Volume 2: Mnemonic Instruction Set* (literature number SPRU172); and for detailed information on the algebraic assembler, see the *TMS320C54x DSP Reference Set, Volume 3: Algebraic Instruction Set* (literature number SPRU179).

instruction set summary (continued)

Table 13. ’54x Instruction Set Opcodes

MNEMONIC SYNTAX	DESCRIPTION	WORDS/ CYCLES†	OPCODE MSB			LSB
ARITHMETIC INSTRUCTIONS						
ABDST *Xmem*, *Ymem*	Absolute distance	1/1	1110	0011	XXXX	YYYY
ABS *src* [, *dst*]	Absolute value of ACC	1/1	1111	01SD	1000	0101
ADD *Smem, src*	Add operand to ACC	1/1	0000	000S	IAAA	AAAA
ADD *Smem,* **TS**, *src*	Add (shifted by TREG[5:0]) operand to ACC	1/1	0000	010S	IAAA	AAAA
ADD *Smem,* **16**, *src* [, *dst*]	Add (shifted by 16 bits) operand to ACC	1/1	0011	11SD	IAAA	AAAA
ADD *Smem* [, *SHIFT*], *src* [, *dst*]	Add shifted operand to ACC (2-word opcode)	2/2	0110 0000	1111 11SD	IAAA 000S	AAAA HIFT
ADD *Xmem, SHFT*, *src*	Add shifted operand to ACC	1/1	1001	000S	XXXX	SHFT
ADD *Xmem, Ymem, dst*	Add dual operands, shift result by 16	1/1	1010	000D	XXXX	YYYY
ADD *#lk* [, *SHFT*], *src* [, *dst*]	Add shifted long-immediate value to ACC	2/2	1111	00SD	0000	SHFT
ADD *#lk*, **16**, *src* [, *dst*]	Add (shifted by 16 bits) long-immediate to ACC	2/2	1111	00SD	0110	0000
ADD *src* [, *SHIFT*], [, *dst*]	Add ACC(s) (A/B), then shift result	1/1	1111	01SD	000S	HIFT
ADD *src,* **ASM** [, *dst*]	Add ACC(s) (A/B), then shift result by ASM value	1/1	1111	01SD	1000	0000
ADDC *Smem, src*	Add to accumulator with carry	1/1	0000	011S	IAAA	AAAA
ADDM *#lk*, *Smem*	Add long-immediate value to memory	2/2	0110	1011	IAAA	AAAA
ADDS *Smem, src*	Add to ACC with sign-extension suppressed	1/1	0000	001S	IAAA	AAAA
DADD *Lmem, src* [, *dst*]	Double/dual add to accumulator	1/1	0101	00SD	IAAA	AAAA
DADST *Lmem*, *dst*	Double/dual add/subtract of T, long operand	1/1	0101	101D	IAAA	AAAA
DELAY *Smem*	Memory delay	1/1	0100	1101	IAAA	AAAA
DRSUB *Lmem*, *src*	Double/dual 16-bit subtract from long word	1/1	0101	100S	IAAA	AAAA
DSADT *Lmem, dst*	Double/dual, subtract/add of T, long operand	1/1	0101	111D	IAAA	AAAA
DSUB *Lmem*, *src*	Double-precision/dual 16-bit subtract from ACC	1/1	0101	010S	IAAA	AAAA
DSUBT *Lmem*, *dst*	Double/dual, subtract/subtract of T, long operand	1/1	0101	110D	IAAA	AAAA
EXP *src*	Accumulator exponent	1/1	1111	010S	1000	1110
FIRS *Xmem*, *Ymem*, *pmad*	Symmetrical finite impulse response filter	2/3	1110	0000	XXXX	YYYY
LMS *Xmem*, *Ymem*	Least mean square	1/1	1110	0001	XXXX	YYYY
MAC[**R**] *Smem*, *src*	Multiply by TREG, add to ACC, round if specified	1/1	0010	10RS	IAAA	AAAA
MAC[**R**] *Xmem*, *Ymem*, *src* [, *dst*]	Multiply dual, add to ACC, round if specified	1/1	1011	0RSD	XXXX	YYYY
MAC *#lk*, *src* [, *dst*]	Multiply TREG by long-immediate, add to ACC	2/2	1111	00SD	0110	0111
MAC *Smem*, *#lk*, *src* [, *dst*]	Multiply by long-immediate value, add to ACC	2/2	0110	01SD	IAAA	AAAA
MACA[**R**] *Smem* [, *B*]	Multiply by ACCA, add to ACCB [round]	1/1	0011	01R1	IAAA	AAAA
MACA[**R**] **T**, *src* [, *dst*]	Multiply TREG by ACCA, add to ACC [round]	1/1	1111	01SD	1000	100R
MACD *Smem*, *pmad*, *src*	Multiply by program memory, accumulate/delay	2/3	0111	101S	IAAA	AAAA
MACP *Smem*, *pmad*, *src*	Multiply by program memory, then accumulate	2/3	0111	100S	IAAA	AAAA
MACSU *Xmem*, *Ymem*, *src*	Multiply signed by unsigned, then accumulate	1/1	1010	011S	XXXX	YYYY
MAS[**R**] *Smem, src*	Multiply by T, subtract from ACC [round]	1/1	0010	11RS	IAAA	AAAA

† Values for words and cycles assume the use of DARAM for data. Add one word and one cycle when using long-offset indirect addressing or absolute addressing with a single data-memory operand.
‡ Delayed Instruction
§ Condition true
¶ Condition false

ADVANCE INFORMATION

instruction set summary (continued)

Table 13. ’54x Instruction Set Opcodes (Continued)

MNEMONIC SYNTAX	DESCRIPTION	WORDS/ CYCLES†	OPCODE MSB			LSB
ARITHMETIC INSTRUCTIONS (CONTINUED)						
MAS[**R**] *Xmem*, *Ymem*, *src* [, *dst*]	Multiply dual, subtract from ACC [round]	1/1	1011	1RSD	XXXX	YYYY
MASA *Smem* [, *B*]	Multiply operand by ACCA, subtract from ACCB	1/1	0011	0011	IAAA	AAAA
MASA[**R**] **T**, *src* [, *dst*]	Multiply ACCA by T, subtract from ACC [round]	1/1	1111	01SD	1000	101R
MAX *dst*	Accumulator maximum	1/1	1111	010D	1000	0110
MIN *dst*	Accumulator minimum	1/1	1111	010D	1000	0111
MPY[**R**] *Smem*, *dst*	Multiply TREG by operand, round if specified	1/1	0010	00RD	IAAA	AAAA
MPY *Xmem*, *Ymem*, *dst*	Multiply dual data-memory operands	1/1	1010	010D	XXXX	YYYY
MPY *Smem*, #*lk*, *dst*	Multiply operand by long-immediate operand	2/2	0110	001D	IAAA	AAAA
MPY #*lk*, *dst*	Multiply TREG value by long-immediate operand	2/2	1111	000D	0110	0110
MPYA *Smem*	Multiply single data-memory operand by ACCA	1/1	0011	0001	IAAA	AAAA
MPYA *dst*	Multiply TREG value by ACCA	1/1	1111	010D	1000	1100
MPYU *Smem*, *dst*	Multiply unsigned	1/1	0010	010D	IAAA	AAAA
NEG *src* [, *dst*]	Negate accumulator	1/1	1111	01SD	1000	0100
NORM *src* [, *dst*]	Normalize	1/1	1111	01SD	1000	1111
POLY *Smem*	Evaluate polynomial	1/1	0011	0110	IAAA	AAAA
RND *src* [, *dst*]	Round accumulator	1/1	1111	01SD	1001	1111
SAT *src*	Saturate accumulator	1/1	1111	010S	1000	0011
SQDST *Xmem*, *Ymem*	Square distance	1/1	1110	0010	XXXX	YYYY
SQUR *Smem, dst*	Square single data-memory operand	1/1	0010	011D	IAAA	AAAA
SQUR A, *dst*	Square ACCA high	1/1	1111	010D	1000	1101
SQURA *Smem, src*	Square and accumulate	1/1	0011	100S	IAAA	AAAA
SQURS *Smem, src*	Square and subtract	1/1	0011	101S	IAAA	AAAA
SUB *Smem, src*	Subtract operand from accumulator	1/1	0000	100S	IAAA	AAAA
SUB *Smem,* **TS**, *src*	Shift by TREG[5:0], then subtract from ACC	1/1	0000	110S	IAAA	AAAA
SUB *Smem,* **16**, *src* [, *dst*]	Shift operand 16 bits, then subtract from ACC	1/1	0100	00SD	IAAA	AAAA
SUB *Smem* [, *SHIFT*], *src* [, *dst*]	Shift operand, then subtract from ACC (2-word opcode)	2/2	0110 0000	1111 11SD	IAAA 001S	AAAA HIFT
SUB *Xmem, SHFT, src*	Shift operand, then subtract from ACC	1/1	1001	001S	XXXX	SHFT
SUB *Xmem, Ymem, dst*	Shift dual operands by 16, then subtract	1/1	1010	001D	XXXX	YYYY
SUB #*lk* [, *SHFT*], *src* [, *dst*]	Shift long-immediate, then subtract from ACC	2/2	1111	00SD	0001	SHFT
SUB #*lk,* **16**, *src* [, *dst*]	Shift long-immediate 16 bits, subtract from ACC	2/2	1111	00SD	0110	0001
SUB *src* [, *SHIFT*], [, *dst*]	Subtract shifted ACC from ACC	1/1	1111	01SD	001S	HIFT
SUB *src,* **ASM** [, *dst*]	Subtract ACC shifted by ASM from ACC	1/1	1111	01SD	1000	0001
SUBB *Smem, src*	Subtract from accumulator with borrow	1/1	0000	111D	IAAA	AAAA
SUBC *Smem, src*	Subtract conditionally	1/1	0001	111S	IAAA	AAAA
SUBS *Smem, src*	Subtract from ACC, sign-extension suppressed	1/1	0000	101S	IAAA	AAAA

† Values for words and cycles assume the use of DARAM for data. Add one word and one cycle when using long-offset indirect addressing or absolute addressing with a single data-memory operand.
‡ Delayed Instruction
§ Condition true
¶ Condition false

instruction set summary (continued)

Table 13. ’54x Instruction Set Opcodes (Continued)

MNEMONIC SYNTAX	DESCRIPTION	WORDS/ CYCLES†	OPCODE MSB			LSB
CONTROL INSTRUCTIONS						
B[**D**] *pmad*	Branch unconditionally with optional delay	2/4,2‡	1111	00Z0	0111	0011
BACC[**D**] *src*	Branch to address in ACC, optional delay	1/6,4‡	1111	01ZS	1110	0010
BANZ[**D**] *pmad*, *Sind*	Branch on AR(ARP) not zero, optional delay	2/4§,2¶,2‡	0110	11Z0	IAAA	AAAA
BC[**D**] *pmad*, *cond* [, *cond* [, *cond*]]	Branch conditionally, optional delay	2/5§,3¶,3‡	1111	10Z0	CCCC	CCCC
CALA[**D**] *src*	Call subroutine at address in ACC, optional delay	1/6,4‡	1111	01ZS	1110	0011
CALL[**D**] *pmad*	Call unconditionally, optional delay	2/4,2¶	1111	00Z0	0111	0100
CC[**D**] *pmad*, *cond* [, *cond* [, *cond*]]	Call conditionally, optional delay	2/5§,3¶,3‡	1111	10Z1	CCCC	CCCC
FB[**D**] *extpmad*	Far branch unconditionally (optional delay)	2/4,2‡	1111	10Z0	1KKK	KKKK
FBACC[**D**] *src*	Far branch to address in ACC, optional delay	1/6,4‡	1111	01ZS	1110	0110
FCALA[**D**] *src*	Far call to address in ACC, optional delay	1/6,4‡	1111	01ZS	1110	0111
FCALL[**D**] *extpmad*	Far call unconditionally, optional delay	2/4,2‡	1111	10Z1	1KKK	KKKK
FRAME *K*	Stack pointer immediate offset	1/1	1110	1110	KKKK	KKKK
FRET[**D**]	Far return (FRETD is for delayed return)	1/6,4‡	1111	01Z0	1110	0100
FRETE[**D**]	Far return, enable interrupts, optional delay	1/6,4‡	1111	01Z0	1110	0101
IDLE *K*	Idle until interrupt	1/4	1111	01NN	1110	0001
INTR *K*	Software interrupt	1/3	1111	0111	110K	KKKK
MAR *Smem*	Modify auxiliary register	1/1	0110	1101	IAAA	AAAA
NOP	No operation	1/1	1111	0100	1001	0101
POPD *Smem*	Pop top of stack to data memory	1/1	1000	1011	IAAA	AAAA
POPM *MMR*	Pop top of stack to memory-mapped register	1/1	1000	1010	IAAA	AAAA
PSHD *Smem*	Push data-memory value onto stack	1/1	0100	1011	IAAA	AAAA
PSHM *MMR*	Push memory-mapped register onto stack	1/1	0100	1010	IAAA	AAAA
RC[**D**] *cond* [, *cond* [, *cond*]]	Return conditionally, optional delay	1/5§,3¶,3‡	1111	11Z0	CCCC	CCCC
RESET	Software reset	1/3	1111	0111	1110	0000
RET[**D**]	Return, optional delay	1/5,3‡	1111	11Z0	0000	0000
RETE[**D**]	Return and enable interrupts, optional delay	1/5,3‡	1111	01Z0	1110	1011
RETF[**D**]	Return fast and enable interrupts, optional delay	1/3,1‡	1111	01Z0	1001	1011
RPT *Smem*	Repeat next instruction, count is in operand	1/1	0100	0111	IAAA	AAAA
RPT *#K*	Repeat next instruction, count is short immediate	1/1	1110	1100	KKKK	KKKK
RPT *#lk*	Repeat next instruction, count is long immediate	2/2	1111	0000	0111	0000
RPTB[**D**] *pmad*	Block repeat, optional delay	2/4,2‡	1111	00Z0	0111	0010
RPTZ *dst, #lk*	Repeat next instruction and clear accumulator	2/2	1111	000D	0111	0001
RSBX *N*, *SBIT*	Reset status-register bit	1/1	1111	01N0	1011	SBIT
SSBX *N*, *SBIT*	Set status-register bit	1/1	1111	01N1	1011	SBIT
TRAP *K*	Software interrupt	1/3	1111	0100	110K	KKKK
XC *n*, *cond* [, *cond* [, *cond*]]	Execute conditionally	1/1	1111	11N1	CCCC	CCCC

† Values for words and cycles assume the use of DARAM for data. Add one word and one cycle when using long-offset indirect addressing or absolute addressing with a single data-memory operand.
‡ Delayed Instruction
§ Condition true
¶ Condition false

ADVANCE INFORMATION

instruction set summary (continued)

Table 13. ’54x Instruction Set Opcodes (Continued)

MNEMONIC SYNTAX	DESCRIPTION	WORDS/ CYCLES†	OPCODE MSB			LSB
I/O INSTRUCTIONS						
PORTR *PA, Smem*	Read data from port	2/2	0111	0100	IAAA	AAAA
PORTW *Smem, PA*	Write data to port	2/2	0111	0101	IAAA	AAAA
LOAD/STORE INSTRUCTIONS						
CMPS *src, Smem*	Compare, select and store maximum	1/1	1000	111S	IAAA	AAAA
DLD *Lmem, dst*	Long-word load to accumulator	1/1	0101	011D	IAAA	AAAA
DST *src, Lmem*	Store accumulator in long word	1/2	0100	111S	IAAA	AAAA
LD *Smem, dst*	Load accumulator with operand	1/1	0001	000D	IAAA	AAAA
LD *Smem*, **TS**, *dst*	Shift operand by TREG[5:0], then load into ACC	1/1	0001	010D	IAAA	AAAA
LD *Smem,* **16***, dst*	Shift operand by 16 bits, then load into ACC	1/1	0100	010D	IAAA	AAAA
LD *Smem* [, *SHIFT*], *dst*	Shift operand, then load into ACC (2-word opcode)	2/2	0110 0000	1111 110D	IAAA 010S	AAAA HIFT
LD *Xmem, SHFT, dst*	Shift operand, then load into ACC	1/1	1001	010D	XXXX	SHFT
LD *#K, dst*	Load ACC with short-immediate operand	1/1	1110	100D	KKKK	KKKK
LD *#lk* [, *SHFT*], *dst*	Shift long-immediate, then load into ACC	2/2	1111	000D	0010	SHFT
LD *#lk*, **16**, *dst*	Shift long-immediate 16 bits, load into ACC	2/2	1111	000D	0110	0010
LD *src*, **ASM** [, *dst*]	Shift ACC by value in ASM register	1/1	1111	01SD	1000	0010
LD *src* [, *SHIFT*] [, *dst*]	Shift accumulator	1/1	1111	01SD	010S	HIFT
LD *Smem,* **T**	Load TREG with single data-memory operand	1/1	0011	0000	IAAA	AAAA
LD *Smem,* **DP**	Load DP with single data-memory operand	1/3	0100	0110	IAAA	AAAA
LD *#k9,* **DP**	Load DP with 9-bit operand	1/1	1110	101K	KKKK	KKKK
LD *#k5,* **ASM**	Load ACC shift-mode register with 5-bit operand	1/1	1110	1101	000K	KKKK
LD *#k3,* **ARP**	Load ARP with 3-bit operand	1/1	1111	0100	1010	0KKK
LD *Smem,* **ASM**	Load operand bits 4–0 into ASM register	1/1	0011	0010	IAAA	AAAA
LD *Xmem, dst* **|| MAC**[**R**] *Ymem* [*, dst_*]	Parallel load, multiply/accumulate [round]	1/1	1010	10RD	XXXX	YYYY
LD *Xmem, dst* **|| MAS**[**R**] *Ymem* [*, dst_*]	Parallel load, multiply/subtract [round]	1/1	1010	11RD	XXXX	YYYY
LDM *MMR, dst*	Load memory-mapped register	1/1	0100	100D	IAAA	AAAA
LDR *Smem, dst*	Load memory value in ACC high with rounding	1/1	0001	011D	IAAA	AAAA
LDU *Smem, dst*	Load unsigned memory value	1/1	0001	001D	IAAA	AAAA
LTD *Smem*	Load TREG and insert delay	1/1	0100	1100	IAAA	AAAA
SACCD *src, Xmem, cond*	Store accumulator conditionally	1/1	1001	111S	XXXX	COND
SRCCD *Xmem, cond*	Store block-repeat counter conditionally	1/1	1001	1101	XXXX	COND
ST T, *Smem*	Store TREG	1/1	1000	1100	IAAA	AAAA
ST TRN, *Smem*	Store TRN	1/1	1000	1101	IAAA	AAAA
ST *#lk*, *Smem*	Store long-immediate operand	2/2	0111	0110	IAAA	AAAA
STH *src, Smem*	Store accumulator high to data memory	1/1	1000	001S	IAAA	AAAA

† Values for words and cycles assume the use of DARAM for data. Add one word and one cycle when using long-offset indirect addressing or absolute addressing with a single data-memory operand.
‡ Delayed Instruction
§ Condition true
¶ Condition false

instruction set summary (continued)

Table 13. ’54x Instruction Set Opcodes (Continued)

MNEMONIC SYNTAX	DESCRIPTION	WORDS/ CYCLES†	OPCODE MSB			LSB
LOAD / STORE INSTRUCTIONS (CONTINUED)						
STH *src*, **ASM**, *Smem*	Shift ACC high by ASM, store to data memory	1 / 1	1000	011S	IAAA	AAAA
STH *src*, *SHFT*, *Xmem*	Shift ACC high, then store to data memory	1 / 1	1001	101S	XXXX	SHFT
STH *src* [, *SHIFT*], *Smem*	Shift ACC high, then store to data memory (2-word opcode)	2 / 2	0110 0000	1111 110S	IAAA 011S	AAAA HIFT
ST *src*, *Ymem* **|| ADD** *Xmem*, *dst*	Store ACC with parallel add	1 / 1	1100	00SD	XXXX	YYYY
ST *src*, *Ymem* **|| LD** *Xmem*, *dst*	Store ACC with parallel load into accumulator	1 / 1	1100	10SD	XXXX	YYYY
ST *src*, *Ymem* **|| LD** *Xmem*, **T**	Store ACC with parallel load into TREG	1 / 1	1110	01S0	XXXX	YYYY
ST *src*, *Ymem* **|| MAC[R]** *Xmem*, *dst*	Parallel store and multiply ACC [round]	1 / 1	1101	0RSD	XXXX	YYYY
ST *src*, *Ymem* **|| MAS[R]** *Xmem*, *dst*	Parallel store, multiply, and subtract	1 / 1	1101	1RSD	XXXX	YYYY
ST *src*, *Ymem* **|| MPY** *Xmem*, *dst*	Parallel store and multiply	1 / 1	1100	11SD	XXXX	YYYY
ST *src*, *Ymem* **|| SUB** *Xmem*, *dst*	Parallel store and subtract	1 / 1	1100	01SD	XXXX	YYYY
STL *src*, *Smem*	Store ACC low to data memory	1 / 1	1000	000S	IAAA	AAAA
STL *src*, **ASM**, *Smem*	Shift ACC low by ASM, store to data memory	1 / 1	1000	010S	IAAA	AAAA
STL *src*, *SHFT*, *Xmem*	Shift ACC low, then store to data memory	1 / 1	1001	100S	XXXX	SHFT
STL *src* [, *SHIFT*], *Smem*	Shift ACC low, then store to data memory (2-word opcode)	2 / 2	0110 0000	1111 110S	IAAA 100S	AAAA HIFT
STLM *src*, *MMR*	Store accumulator low to memory	1 / 1	1000	100S	IAAA	AAAA
STM *#lk*, *MMR*	Store ACC low into memory-mapped register	2 / 2	0111	0111	IAAA	AAAA
STRCD *Xmem*, *cond*	Store TREG conditionally	1 / 1	1001	1100	XXXX	COND
LOGICAL INSTRUCTIONS						
AND *Smem*, *src*	AND single data-memory operand with ACC	1 / 1	0001	100S	IAAA	AAAA
AND *#lk* [, *SHFT*], *src* [, *dst*]	Shift long-immediate operand, AND with ACC	2 / 2	1111	00SD	0011	SHFT
AND *#lk*, **16**, *src* [, *dst*]	Shift long-immediate 16 bits, AND with ACC	2 / 2	1111	00SD	0110	0011
AND *src* [, *SHIFT*], [, *dst*]	AND accumulator(s), then shift result	1 / 1	1111	00SD	100S	HIFT
ANDM *#lk*, *Smem*	AND memory with long-immediate operand	2 / 2	0110	1000	IAAA	AAAA
BIT *Xmem*, *BITC*	Test bit	1 / 1	1001	0110	XXXX	BITC
BITF *Smem*, *#lk*	Test bit field specified by immediate value	2 / 2	0110	0001	IAAA	AAAA
BITT *Smem*	Test bit specified by TREG	1 / 1	0011	0100	IAAA	AAAA
CMPL *src* [, *dst*]	Complement accumulator	1 / 1	1111	01SD	1001	0011
CMPM *Smem*, *#lk*	Compare memory with long-immediate operand	2 / 2	0110	0000	IAAA	AAAA
CMPR *CC*, *ARx*	Compare auxiliary register with AR0	1 / 1	1111	01CC	1010	1ARX
OR *Smem, src*	OR single data-memory operand with ACC	1 / 1	0001	101S	IAAA	AAAA

† Values for words and cycles assume the use of DARAM for data. Add one word and one cycle when using long-offset indirect addressing or absolute addressing with a single data-memory operand.
‡ Delayed Instruction
§ Condition true
¶ Condition false

ADVANCE INFORMATION

instruction set summary (continued)

Table 13. ’54x Instruction Set Opcodes (Continued)

MNEMONIC SYNTAX	DESCRIPTION	WORDS/ CYCLES†	OPCODE MSB			LSB
LOGICAL INSTRUCTIONS (CONTINUED)						
OR *#lk* [, *SHFT*], *src* [, *dst*]	Shift long-immediate operand, then OR with ACC	2/2	1111	00SD	0100	SHFT
OR *#lk,* **16**, *src* [, *dst*]	Shift long-immediate 16 bits, then OR with ACC	2/2	1111	00SD	0110	0100
OR *src* [, *SHIFT*], [, *dst*]	OR accumulator(s), then shift result	1/1	1111	00SD	101S	HIFT
ORM *#lk, Smem*	OR memory with constant	2/2	0110	1001	IAAA	AAAA
ROL *src*	Rotate accumulator left	1/1	1111	010S	1001	0001
ROLTC *src*	Rotate accumulator left using TC	1/1	1111	010S	1001	0010
ROR *src*	Rotate accumulator right	1/1	1111	010S	1001	0000
SFTA *src*, *SHIFT* [, *dst*]	Shift accumulator arithmetically	1/1	1111	01SD	011S	HIFT
SFTC *src*	Shift accumulator conditionally	1/1	1111	010S	1001	0100
SFTL *src*, *SHIFT* [, *dst*]	Shift accumulator logically	1/1	1111	00SD	111S	HIFT
XOR *Smem, src*	XOR operand with ACC	1/1	0001	110S	IAAA	AAAA
XOR *#lk* [, *SHFT*], *src* [, *dst*]	Shift long-immediate, then XOR with ACC	2/2	1111	00SD	0101	SHFT
XOR *#lk,* **16**, *src* [, *dst*]	Shift long-immediate 16 bits, then XOR with ACC	2/2	1111	00SD	0110	0101
XOR *src* [, *SHIFT*] [, *dst*]	XOR accumulator(s), then shift result	1/1	1111	00SD	110S	HIFT
XORM *#lk, Smem*	XOR memory with constant	2/2	0110	1010	IAAA	AAAA
MOVE INSTRUCTIONS						
MVDD *Xmem*, *Ymem*	Move within data memory, X/Y addressing	1/1	1110	0101	XXXX	YYYY
MVDK *Smem*, *dmad*	Move data, destination addressing	2/2	0111	0001	IAAA	AAAA
MVDM *dmad*, *MMR*	Move data to memory-mapped register	2/2	0111	0010	IAAA	AAAA
MVDP *Smem*, *pmad*	Move data to program memory	2/4	0111	1101	IAAA	AAAA
MVKD *dmad*, *Smem*	Move data with source addressing	2/2	0111	0000	IAAA	AAAA
MVMD *MMR*, *dmad*	Move memory-mapped register to data	2/2	0111	0011	IAAA	AAAA
MVMM *MMRx*, *MMRy*	Move between memory-mapped registers	1/1	1110	0111	MMRX	MMRY
MVPD *pmad*, *Smem*	Move program memory to data memory	2/3	0111	1100	IAAA	AAAA
READA *Smem*	Read data memory addressed by ACCA	1/5	0111	1110	IAAA	AAAA
WRITA *Smem*	Write data memory addressed by ACCA	1/5	0111	1111	IAAA	AAAA

† Values for words and cycles assume the use of DARAM for data. Add one word and one cycle when using long-offset indirect addressing or absolute addressing with a single data-memory operand.
‡ Delayed Instruction
§ Condition true
¶ Condition false

ADVANCE INFORMATION

development support

Texas Instruments offers an extensive line of development tools for the ’54x 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.

The following products support development of ’54x-based applications:

Software Development Tools:
Assembler/Linker
Simulator
Optimizing ANSI C compiler
Application algorithms
C/Assembly debugger and code profiler

Hardware Development Tools:
Extended development system (XDS™) emulator (supports ’54x multiprocessor system debug)
’54x EVM (Evaluation Module)
’54x DSK (DSP Starter Kit)

The *TMS320 Family Development Support Reference Guide* (SPRU011) contains information about development support products for all TMS320 family member devices, including documentation. Refer to this document for further information about TMS320 documentation or any other TMS320 support products from Texas Instruments. There is an additional document, the *TMS320 Third Party Support Reference Guide* (SPRU052), which contains information about TMS320-related products from other companies in the industry. To receive copies of TMS320 literature, contact the Literature Response Center at 800/477-8924.

See Table 14 for complete listings of development support tools for the ’54x. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

Table 14. Development Support Tools

DEVELOPMENT TOOL	PLATFORM	PART NUMBER
Software		
Assembler/Linker	PC-DOS™, OS/2™	TMDS324L850-02
Compiler/Assembler/Linker	PC-DOS, OS/2	TMDS324L855-02
Compiler/Assembler/Linker	SPARC™	TMDS324L555-09
Simulator	PC-DOS, WIN™	TMDS324L851-02
Simulator	SPARC, WIN	TMDS324L551-09
Digital Filter Design Package for PC	PC-DOS	DFDP
XDS510™ Debugger/Emulation Software	PC-DOS, OS/2, WIN	TMDS32401L0
XDS510WS™ Debugger/Emulation Software	SPARC, WIN	TMDS32406L0
Hardware		
XDS510 Emulator†	PC-DOS, OS/2	TMDS00510
XDS510WS Emulator‡	SPARC, WIN	TMDS00510WS
3 V/5 V PC/SPARC JTAG Emulation Cable	N/A	TMDS3080002
EVM Evaluation Module	PC-DOS, WIN	TMDX3260051
DSK DSP Starter Kit	PC-DOS	TMDX32400L0

† Includes XDS510 board and JTAG emulation cable; TMDS32401L0 C-source debugger conversion software not included
‡ Includes XDS510WS box, SCSI cable, power supply, and JTAG emulation cable; TMDS32406L0 C-source debugger conversion software not included

PC-DOS and OS/2 are trademarks of International Business Machines Corp.
SPARC is a trademark of SPARC International, Inc.
WIN is a trademark of Microstate Corporation.
XDS, XDS510, and XDS510WS are trademarks of Texas Instruments Incorporated.

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 devices and support tools. Each TMS320 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). This development flow is defined below.

Device development evolutionary flow:

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

Support tool development evolutionary flow:

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

TMDS Fully qualified development support product

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 has 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 10 provides a legend for reading the complete device name for any TMS320 family member.

device and development support tool nomenclature (continued)

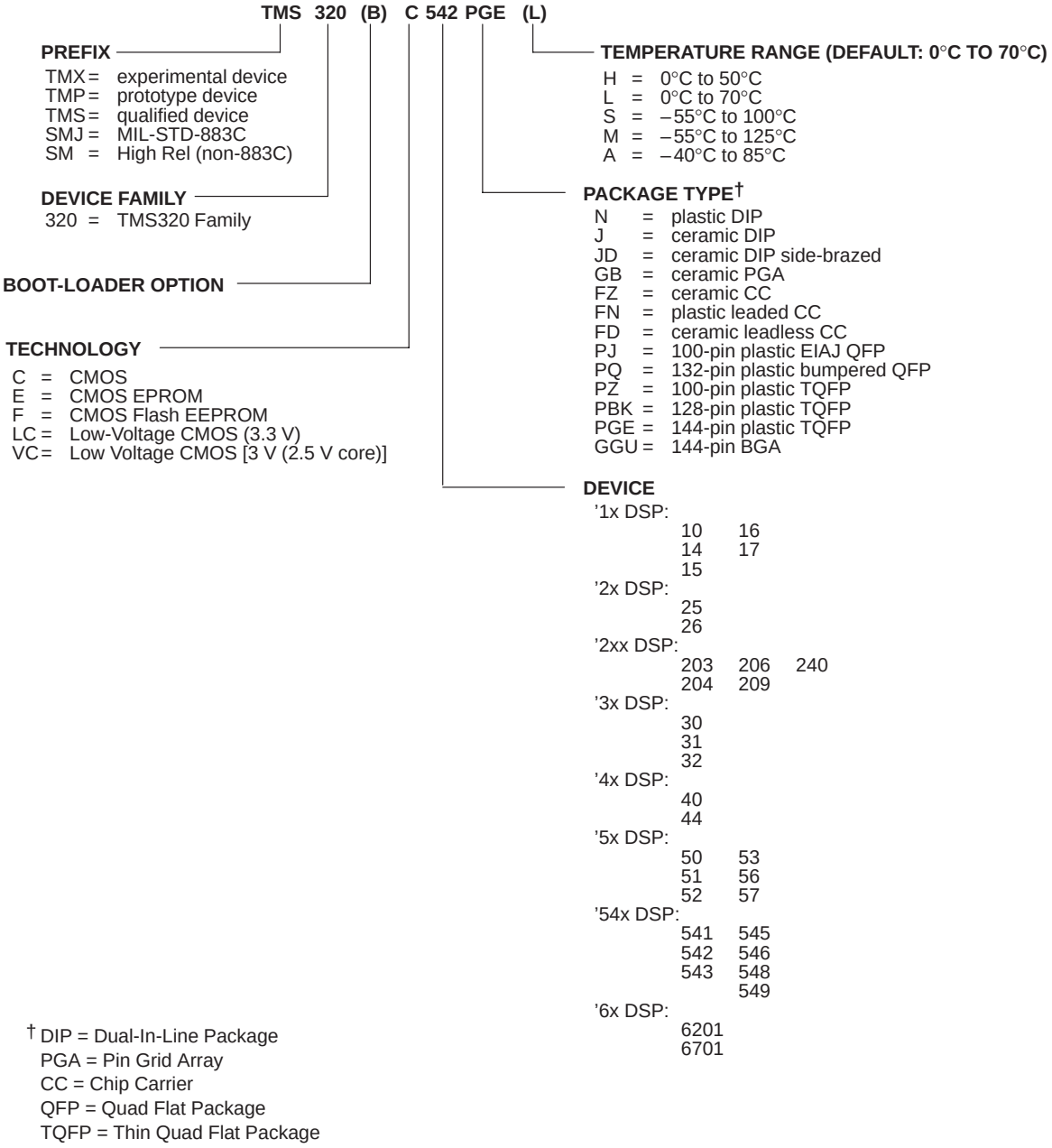


Figure 10. TMS320 DSP Device Nomenclature

ADVANCE INFORMATION

documentation support

Extensive documentation supports all TMS320 family generations of devices from product announcement through applications development. The types of documentation available include: data sheets, such as this document, with design specifications; complete user's guides for all devices; development support tools; and hardware and software applications.

The four-volume *TMS320C54x DSP Reference Set* (literature number SPRU210) consists of:

- *Volume 1: CPU and Peripherals* (literature number SPRU131)
- *Volume 2: Mnemonic Instruction Set* (literature number SPRU172)
- *Volume 3: Algebraic Instruction Set* (literature number SPRU179)
- *Volume 4: Applications Guide* (literature number SPRU173)

The reference set describes in detail the '54x TMS320 products currently available and the hardware and software applications, including algorithms, for fixed-point TMS320 devices.

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 newsletter, *Details on Signal Processing*, is published quarterly and distributed to update TMS320 customers on product information. The TMS320 DSP bulletin board service (BBS) provides access to information pertaining to the TMS320 family, including documentation, source code and object code for many DSP algorithms and utilities. The BBS can be reached at 281/274-2323.

Information regarding TI DSP products is also available on the Worldwide Web at http://www.ti.com uniform resource locator (URL).

electrical characteristics and operating conditions (’541, ’542)

absolute maximum ratings over specified temperature range (unless otherwise noted)†

Supply voltage range, V_{DD}‡ . . . . . –0.3 V to 7 V
Input voltage range . . . . . –0.3 V to 7 V
Output voltage range . . . . . –0.3 V to 7 V
Operating case temperature range, T_C . . . . . –40°C to 100°C
Storage temperature range, T_{stg} . . . . . –55°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
‡ All voltage values are with respect to V_{SS}.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{DD}	Supply voltage		4.75	5	5.25	V
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage	$\overline{RS}$, $\overline{INTn}$, $\overline{NMI}$,CNT,CLKMDn X2/CLKIN	3		V_{DD} + 0.3	V
		All other inputs	2		V_{DD} + 0.3	
V_{IL}	Low-level input voltage		–0.3		0.8	V
I_{OH}	High-level output current				–300	µA
I_{OL}	Low-level output current				2	mA
T_C	Operating case temperature		–40		100	°C

Refer to Figure 11 for 5-V device test load circuit values.

ADVANCE INFORMATION

electrical characteristics and operating conditions (’541, ’542) (continued)

electrical characteristics over recommended operating case temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage‡		I_{OH} = –300 µA	2.4			V
V_{OL}	Low-level output voltage‡		I_{OL} = 2 mA			0.6	V
I_{IZ}	Input current in high impedance		V_{DD} = MAX, V_O = V_{SS} to V_{DD}	–20		20	µA
I_I	Input current (V_I = V_{SS} to V_{DD})	$\overline{TRST}$	With internal pulldown	–10		800	µA
		HPIENA	With internal pulldown, $\overline{RS}$ = 0	–10		400	µA
		TMS, TCK, TDI, HPI‖	With internal pullups	–500		10	
		D[15:0], HD[7:0]	Bus holders enabled, V_{DD} = MAX, V_I = V_{SS} to V_{DD}	–175		175	
		All other input-only pins		–10		10	
I_{DDC}	Supply current, core CPU		V_{DD} = 5 V, f_x = 40 MHz,§ T_C = 25°C		47¶		mA
I_{DDP}	Supply current, pins		V_{DD} = 5 V, f_x = 40 MHz,§ T_C = 25°C		18#		mA
I_{DD}	Supply current, standby	IDLE2	PLL × 1 mode, 40 MHz input		4		mA
		IDLE3	Divide-by-two mode, CLKIN stopped		5		µA
C_i	Input capacitance				10		pF
C_o	Output capacitance				10		pF

† All typical values are at V_{DD} = 5 V, T_C = 25°C.
‡ All input and output voltage levels except $\overline{RS}$, $\overline{INT0}$–$\overline{INT3}$, $\overline{NMI}$, CNT, X2/CLKIN, CLKMD0–CLKMD3 are TTL-compatible.
§ Clock mode: PLL × 1 with external source
¶ This value was obtained with 50% usage of MAC and 50% usage of NOP instructions. Actual operating current varies with program being executed.
# This value was obtained with single-cycle external writes, CLKOFF = 0 and load = 15 pF. For more details on how this calculation is performed, refer to the *Calculation of TMS320C54x Power Dissipation* application report (literature number SPRA164).
‖ HPI input signals except for HPIENA.

ADVANCE INFORMATION

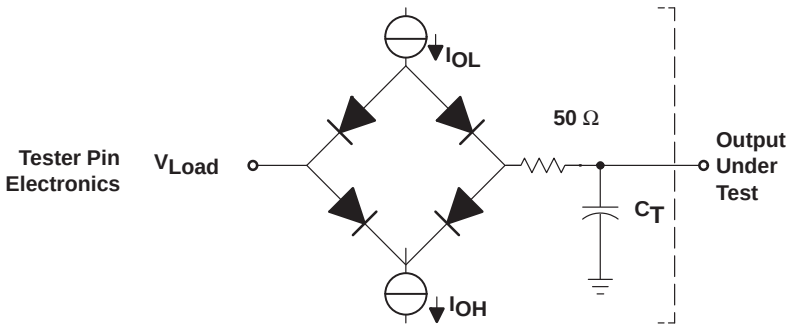
PARAMETER MEASUREMENT INFORMATION

timing parameter symbology

Timing parameter symbols used are created in accordance with JEDEC Standard 100-A. To shorten the symbols, some of the pin names and other related terminology have been abbreviated as follows:

Lowercase subscripts and their meanings:	
a	access time
c	cycle time (period)
d	delay time
dis	disable time
en	enable time
f	fall time
h	hold time
r	rise time
su	setup time
t	transition time
v	valid time
w	pulse duration (width)
X	Unknown, changing, or don't care level

Letters and symbols and their meanings:	
H	High
L	Low
V	Valid
Z	High impedance



Where: I_{OL} = 2 mA (all outputs)
I_{OH} = 300 μA (all outputs)
V_{Load} = 1.5 V
C_T = 40 pF typical load circuit capacitance.

Figure 11. 5-V Test Load Circuit

ADVANCE INFORMATION

electrical characteristics and operating conditions (’LC54x, ’VC54x)

absolute maximum ratings over specified temperature range (unless otherwise noted)†

Supply voltage I/O range, DV_{DD}‡ . . . . . –0.3 V to 4.6 V
Supply voltage core range, CV_{DD}‡ . . . . . –0.3 V to 3.75 V
Input voltage range . . . . . –0.3 V to 4.6 V
Output voltage range . . . . . –0.3 V to 4.6 V
Operating case temperature range, T_C . . . . . –40°C to 100°C
Storage temperature range, T_{stg} . . . . . –55°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
‡ All voltage values are with respect to V_{SS}.

recommended operating conditions

			MIN	NOM	MAX	UNIT
DV_{DD}	Device supply voltage, I/O (’LC54x -40, -50, -66, -80, -100)		3	3.3	3.6	V
CV_{DD}	Device supply voltage, core (’VC549-100)		2.4	2.5	2.75	V
V_{SS}	Supply voltage, GND			0		V
V_{IH}	High-level input voltage, I/O (’LC54x -40, -50, -66, -80, -100)	$\overline{RS}$,$\overline{INTn}$,$\overline{NMI}$,CNT, X2/CLKIN, CLKMDn, DV_{DD} = 3.3 ± 0.3 V	2.5		DV_{DD} + 0.3	V
		All other inputs	2		DV_{DD} + 0.3	
V_{IL}	Low-level input voltage		–0.3		0.8	V
I_{OH}	High-level output current				–300	µA
I_{OL}	Low-level output current				1.5	mA
T_C	Operating case temperature		–40		100	°C

Refer to Figure 12 for 3.3-V device test load circuit values.

ADVANCE INFORMATION

electrical characteristics and operating conditions (’LC54x, ’VC54x) (continued)

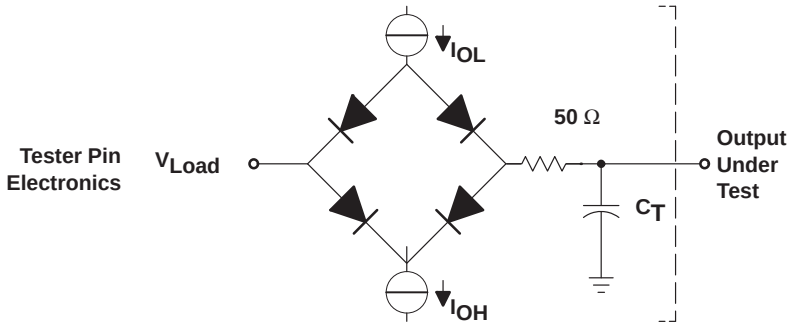
electrical characteristics over recommended operating case temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage‡		V_{DD} = 3.3 ± 0.3 V, I_{OH} = MAX	2.4			V
V_{OL}	Low-level output voltage‡		I_{OL} = MAX			0.4	V
I_{IZ}	Input current in high impedance	A[22:0] (’548/’549 only)	V_{DD} = MAX, V_O = V_{SS} to V_{DD}	–175		175	µA
		All other ’54x devices	V_{DD} = MAX, V_O = V_{SS} to V_{DD}	–10		10	
I_I	Input current (V_I = V_{SS} to V_{DD})	$\overline{TRST}$	With internal pulldown	–10		800	µA
		HPIENA	With internal pulldown, $\overline{RS}$ = 0	–10		400	
		TMS, TCK, TDI, HPI‖	With internal pullups	–400		10	
		D[15:0], HD[7:0]	Bus holders enabled, V_{DD} = MAX, V_I = V_{SS} to V_{DD}	–175		175	
		All other input-only pins		–10		10	
I_{DDC}	Supply current, core CPU		DV_{DD} = 3.0 V, f_X = 40 MHz,§ T_C = 25°C		28¶		mA
I_{DDC}	Supply current, core CPU (’549 only)		CV_{DD} = 2.5 V, f_X = 40 MHz,§ T_C = 25°C		20¶		mA
I_{DDP}	Supply current, pins		V_{DD} = 3.0 V, f_X = 40 MHz,§ T_C = 25°C		10.8#		mA
I_{DD}	Supply current, standby	IDLE2	PLL × 1 mode, 40 MHz input		2		mA
		IDLE3	Divide-by-two mode, CLKIN stopped		5☆		µA
C_i	Input capacitance				10		pF
C_o	Output capacitance				10		pF

† All values are typical unless otherwise specified.
‡ All input and output voltage levels except $\overline{RS}$, $\overline{INT0}$–$\overline{INT3}$, $\overline{NMI}$, CNT, X2/CLKIN, CLKMD0–CLKMD3 are LVTTL-compatible.
§ Clock mode: PLL × 1 with external source
¶ This value was obtained with 50% usage of MAC and 50% usage of NOP instructions. Actual operating current varies with program being executed.
# This value was obtained with single-cycle external writes, CLKOFF = 0 and load = 15 pF. For more details on how this calculation is performed, refer to the *Calculation of TMS320C54x Power Dissipation* application report (literature number SPRA164).
‖ HPI input signals except for HPIENA.
☆ Values derived from characterization data and not tested.

ADVANCE INFORMATION

PARAMETER MEASUREMENT INFORMATION



Where: I_{OL} = 1.5 mA (all outputs)
I_{OH} = 300 μA (all outputs)
V_{Load} = 1.5 V
C_T = 40 pF typical load circuit capacitance.

Figure 12. 3.3-V Test Load Circuit

ADVANCE INFORMATION

internal divide-by-two clock option with external crystal

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN. The frequency of CLKOUT is one-half the crystal's oscillating frequency. The crystal should be either fundamental or overtone operation, and parallel resonant, with an effective series resistance of 30 Ω and power dissipation of 1 mW; it should be specified at a load capacitance of 20 pF. Note the circuit shown in Figure 13 represents fundamental mode operation. For overtone mode operation, additional components are generally required.

recommended operating conditions (see Figure 13)

		’C54x-40 ’LC54x-40			’LC54x-50			’54x-66			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
f_X	Input clock frequency	0†		80	0†		100	0†		133.33‡	MHz
C1, C2			10			10			10		pF

		’LC54x-80			’VC54x-100			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
f_X	Input clock frequency	0†		160‡§	0†		200‡§	MHz
C1, C2			10			10		pF

† This device utilizes a fully static design and therefore can operate with $t_{c(CI)}$ approaching ∞. The device is characterized at frequencies approaching 0 Hz.
‡ Values derived from characterization data and not tested.
§ It is recommended that the PLL clocking option be used for maximum frequency operation.

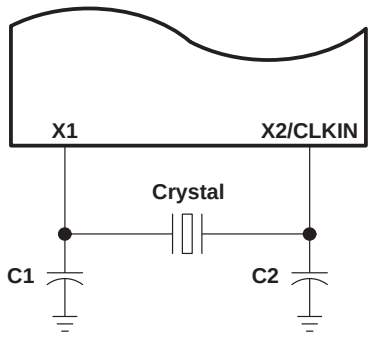


Figure 13. Internal Divide-by-Two Clock Option With External Crystal

ADVANCE INFORMATION

external divide-by-two clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected, CLKMD1 and CLKMD2 set low, and CLKMD3 set high. This external frequency is divided by two to generate the internal machine cycle.

The external frequency injected must conform to specifications listed in the timing requirements table.

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Figure 13 and Figure 14, and the recommended operating conditions table†)

PARAMETER		’C54x-40 ’LC54x-40			’LC54x-50			’54x-66			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
$t_{c(CO)}$	Cycle time, CLKOUT	25	$2t_{c(CI)}$	‡	20	$2t_{c(CI)}$	‡	15	$2t_{c(CI)}$	‡	ns
$t_{d(CIH-CO)}$	Delay time, X2/CLKIN high to CLKOUT high/low	6	12	18	6	12	18	4	10	16	ns
$t_{f(CO)}$	Fall time, CLKOUT†		2			2			2		ns
$t_{r(CO)}$	Rise time, CLKOUT†		2			2			2		ns
$t_{w(COL)}$	Pulse duration, CLKOUT low†	H–4	H–2	H	H–4	H–2	H	H–4	H–2	H	ns
$t_{w(COH)}$	Pulse duration, CLKOUT high†	H–4	H–2	H	H–4	H–2	H	H–4	H–2	H	ns

PARAMETER		’LC54x-80			’VC54x-100			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
$t_{c(CO)}$	Cycle time, CLKOUT	12.5†§	$2t_{c(CI)}$	‡	10†§	$2t_{c(CI)}$	‡	ns
$t_{d(CIH-CO)}$	Delay time, X2/CLKIN high to CLKOUT high/low	4	10	16	4	10	16	ns
$t_{f(CO)}$	Fall time, CLKOUT†		2			2		ns
$t_{r(CO)}$	Rise time, CLKOUT†		2			2		ns
$t_{w(COL)}$	Pulse duration, CLKOUT low†	H–3	H–1	H	H–2	H–1	H	ns
$t_{w(COH)}$	Pulse duration, CLKOUT high†	H–3	H–1	H	H–2	H–1	H	ns

† Values derived from characterization data and not tested.
‡ This device utilizes a fully static design and therefore can operate with $t_{c(CI)}$ approaching ∞. The device is characterized at frequencies approaching 0 Hz.
§ It is recommended that the PLL clocking option be used for maximum frequency operation.

external divide-by-two clock option (continued)

timing requirements over recommended operating conditions (see Figure 14)

		’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{c(CI)}$	Cycle time, X2/CLKIN	12.5	†	10	†	7.5	†	ns
$t_{f(CI)}$	Fall time, X2/CLKIN‡		4		4		4	ns
$t_{r(CI)}$	Rise time, X2/CLKIN‡		4		4		4	ns
$t_{w(CIL)}$	Pulse duration, X2/CLKIN low	3	†	3	†	3	†	ns
$t_{w(CIH)}$	Pulse duration, X2/CLKIN high	3	†	3	†	3	†	ns

		’LC54x-80		’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{c(CI)}$	Cycle time, X2/CLKIN	6.25	†	5	†	ns
$t_{f(CI)}$	Fall time, X2/CLKIN‡		1		1	ns
$t_{r(CI)}$	Rise time, X2/CLKIN‡		1		1	ns
$t_{w(CIL)}$	Pulse duration, X2/CLKIN low	2	†	2	†	ns
$t_{w(CIH)}$	Pulse duration, X2/CLKIN high	2	†	2	†	ns

† This device utilizes a fully static design and therefore can operate with $t_{c(CI)}$ approaching ∞. The device is characterized at frequencies approaching 0 Hz.

‡ Values assured by design but not tested.

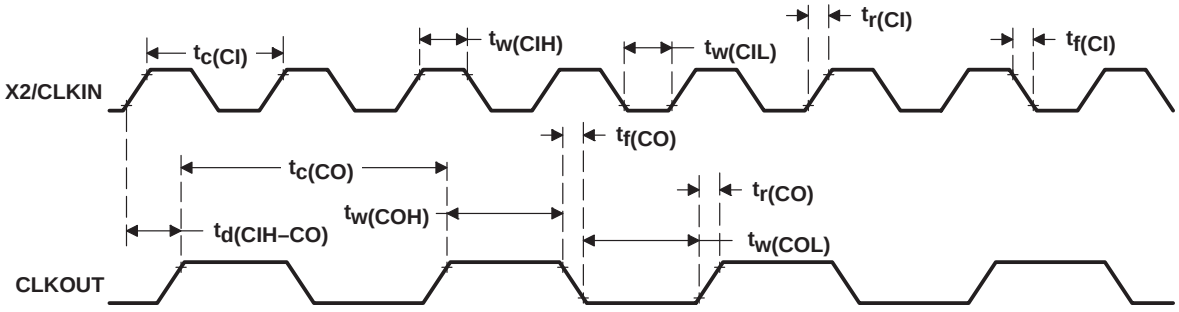


Figure 14. External Divide-by-Two Clock Timing

external multiply-by-N clock option

An external frequency can be used by injecting the frequency directly into X2/CLKIN, with X1 left unconnected, CLKMD1, CLKMD2, and CLKMD3 set according to the clock mode configuration table (Table 4). This external frequency is multiplied by N to generate the internal machine cycle.

The external frequency injected must conform to specifications listed in the timing requirements table.

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Figure 13 and Figure 15, and the recommended operating conditions table)

PARAMETER		’C54x-40 ’LC54x-40			’LC54x-50			’54x-66			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
$t_{c(CO)}$	Cycle time, CLKOUT	25	$t_{c(CI)}$/N		20	$t_{c(CI)}$/N		15	$t_{c(CI)}$/N		ns
$t_{d(CIH-CO)}$	Delay time, X2/CLKIN high/low to CLKOUT high/low	6	12	18	6	12	18	4	10	16	ns
$t_{f(CO)}$	Fall time, CLKOUT		2			2			2		ns
$t_{r(CO)}$	Rise time, CLKOUT		2			2			2		ns
$t_{w(COL)}$	Pulse duration, CLKOUT low	H–4	H–2	H	H–4	H–2	H	H–4	H–2	H	ns
$t_{w(COH)}$	Pulse duration, CLKOUT high	H–4	H–2	H	H–4	H–2	H	H–4	H–2	H	ns
t_p	Transitory phase, PLL lock-up time			50†			50†			50†	µs

PARAMETER		’LC54x-80			’VC54x-100			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
$t_{c(CO)}$	Cycle time, CLKOUT	12.5	$t_{c(CI)}$/N		10	$t_{c(CI)}$/N		ns
$t_{d(CIH-CO)}$	Delay time, X2/CLKIN high/low to CLKOUT high/low†	4	10	16	4	10	16	ns
$t_{f(CO)}$	Fall time, CLKOUT		2			2		ns
$t_{r(CO)}$	Rise time, CLKOUT		2			2		ns
$t_{w(COL)}$	Pulse duration, CLKOUT low	H–3	H–1	H	H–2	H–1	H	ns
$t_{w(COH)}$	Pulse duration, CLKOUT high	H–3	H–1	H	H–2	H–1	H	ns
t_p	Transitory phase, PLL lock-up time			29†			35†	µs

† Values derived from characterization data and not tested

external multiply-by-N clock option (continued)

timing requirements over recommended operating conditions (see Figure 15)

			’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
$t_{c(CI)}$	Cycle time, X2/CLKIN	Integer PLL multiplier N (N = 1–15)	25N	400N	20N	400N	15N	400N	ns
		PLL multiplier N = x.5	25N	200N	20N	200N	15N	200N	
		PLL multiplier N = x.25, x.75	25N	100N	20N	100N	15N	100N	
$t_{f(CI)}$	Fall time, X2/CLKIN†			4		4		4	ns
$t_{r(CI)}$	Rise time, X2/CLKIN†			4		4		4	ns
$t_{w(CIL)}$	Pulse duration, X2/CLKIN low		8		6		3.5		ns
$t_{w(CIH)}$	Pulse duration, X2/CLKIN high		8		6		3.5		ns

			’LC54x-80		’VC54x-100		UNIT
			MIN	MAX	MIN	MAX	
$t_{c(CI)}$	Cycle time, X2/CLKIN	Integer PLL multiplier N (N = 1–15)	12.5N	400N	10N	400N	ns
		PLL multiplier N = x.5	12.5N	200N	10N	200N	
		PLL multiplier N = x.25, x.75	12.5N	100N	10N	100N	
$t_{f(CI)}$	Fall time, X2/CLKIN†			2		2	ns
$t_{r(CI)}$	Rise time, X2/CLKIN†			2		2	ns
$t_{w(CIL)}$	Pulse duration, X2/CLKIN low			3		2	ns
$t_{w(CIH)}$	Pulse duration, X2/CLKIN high			3		2	ns

† Values derived from characterization data and not tested.

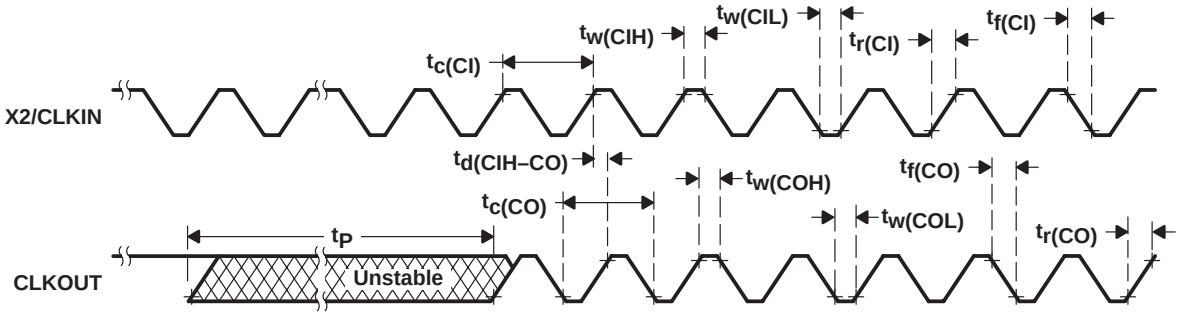


Figure 15. External Multiply-by-One Clock Timing

memory and parallel I/O interface timing

switching characteristics over recommended operating conditions for a memory read ($\overline{MSTRB}$ = 0) [H = 0.5 $t_{c(CO)}$]†‡ (see Figure 16)

PARAMETER		’LC542-40 ’LC543-40		’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
$t_{d(CLKL-A)}$	Delay time, address valid from CLKOUT low§	0¶	5	0¶	5	0¶	5	0¶	5	ns
$t_{d(CLKH-A)}$	Delay time, address valid from CLKOUT high (transition)#	0¶	5	0¶	5	0¶	5	– 2¶	3	ns
$t_{d(CLKL-MSL)}$	Delay time, $\overline{MSTRB}$ low from CLKOUT low	0	5	0	5	0	5	0	5	ns
$t_{d(CLKL-MSH)}$	Delay time, $\overline{MSTRB}$ high from CLKOUT low	– 2	3	– 2	3	– 2	3	– 2	3	ns
$t_{h(CLKL-A)R}$	Hold time, address valid after CLKOUT low§	0	5¶	0	5¶	0	5¶	0	5¶	ns
$t_{h(CLKH-A)R}$	Hold time, address valid after CLKOUT high#	0	5¶	0	5¶	0	5¶	– 2	3¶	ns

PARAMETER		’LC54x-80		’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{d(CLKL-A)}$	Delay time, address valid from CLKOUT low§	0¶	4	0¶	3	ns
$t_{d(CLKH-A)}$	Delay time, address valid from CLKOUT high (transition)#	0¶	4	0¶	3	ns
$t_{d(CLKL-MSL)}$	Delay time, $\overline{MSTRB}$ low from CLKOUT low	0	4	0	3	ns
$t_{d(CLKL-MSH)}$	Delay time, $\overline{MSTRB}$ high from CLKOUT low	– 2	2	– 2	2	ns
$t_{h(CLKL-A)R}$	Hold time, address valid after CLKOUT low§	0	4¶	0	3¶	ns
$t_{h(CLKH-A)R}$	Hold time, address valid after CLKOUT high#	0	4¶	0	3¶	ns

† Address, $\overline{PS}$, and $\overline{DS}$ timings are all included in timings referenced as address.
‡ See Table 15, Table 16, and Table 17 for address bus timing variation with load capacitance.
§ In the case of a memory read preceded by a memory read
¶ Values derived from characterization data and not tested.
# In the case of a memory read preceded by a memory write

ADVANCE INFORMATION

memory and parallel I/O interface timing (continued)

timing requirements over recommended operating conditions for a memory read ($\overline{\text{MSTRB}}$ = 0) [H = 0.5 $t_{c(CO)}$]†‡ (see Figure 16)

		’LC542-40 ’LC543-40		’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
$t_{a(A)M}$	Access time, read data access from address valid		2H–12		2H–10		2H–10		2H–10	ns
$t_{a(MSTRBL)}$	Access time, read data access from $\overline{\text{MSTRB}}$ low		2H–12		2H–10		2H–10		2H–10	ns
$t_{su(D)R}$	Setup time, read data before CLKOUT low	7		5		5		5		ns
$t_{h(D)R}$	Hold time, read data after CLKOUT low	0		0		0		2		ns
$t_{h(A-D)R}$	Hold time, read data after address invalid	0		0		0		1		ns
$t_{h(D)MSTRBH}$	Hold time, read data after $\overline{\text{MSTRB}}$ high	0		0		0		0		ns

		’LC54x-80		’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{a(A)M}$	Access time, read data access from address valid		2H–7.5		2H–6	ns
$t_{a(MSTRBL)}$	Access time, read data access from $\overline{\text{MSTRB}}$ low		2H–7.5		2H–6	ns
$t_{su(D)R}$	Setup time, read data before CLKOUT low	5		4		ns
$t_{h(D)R}$	Hold time, read data after CLKOUT low	2		2		ns
$t_{h(A-D)R}$	Hold time, read data after address invalid	1		1		ns
$t_{h(D)MSTRBH}$	Hold time, read data after $\overline{\text{MSTRB}}$ high	0		0		ns

† Address, $\overline{\text{PS}}$, and $\overline{\text{DS}}$ timings are all included in timings referenced as address.
‡ See Table 15, Table 16, and Table 17 for address bus timing variation with load capacitance.

memory and parallel I/O interface timing (continued)

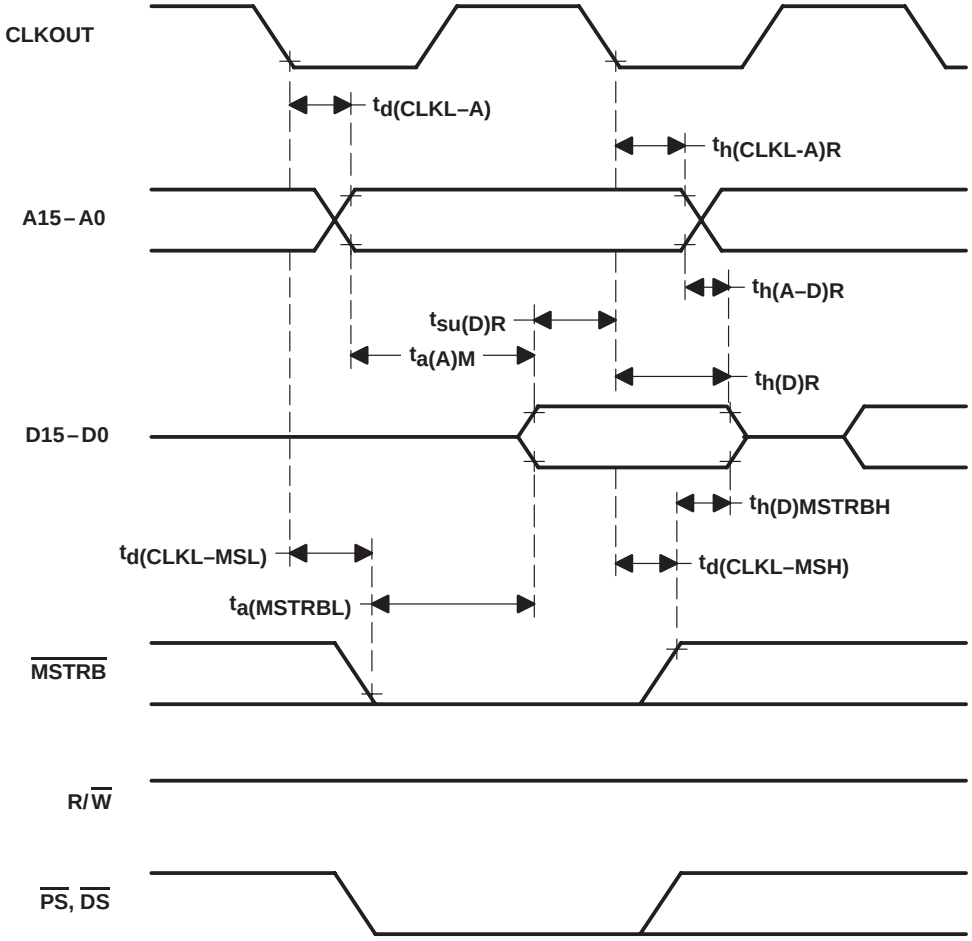


Figure 16. Memory Read ($\overline{\text{MSTRB}}$ = 0)

ADVANCE INFORMATION

memory and parallel I/O interface timing (continued)

switching characteristics over recommended operating conditions for a memory write ($\overline{MSTRB}$ = 0) [H = 0.5 $t_{c(CO)}$]†‡ (see Figure 17)

PARAMETER		'C54x-40 'LC54x-40 'LC54x-50		'54x-66		UNIT
		MIN	MAX	MIN	MAX	
$t_{d(CLKH-A)}$	Delay time, address valid from CLKOUT high§	0#	5	– 2#	3	ns
$t_{d(CLKL-A)}$	Delay time, address valid from CLKOUT low¶	0#	5	0#	5	ns
$t_{d(CLKL-MSL)}$	Delay time, $\overline{MSTRB}$ low from CLKOUT low	0	5	0	5	ns
$t_{d(CLKL-D)W}$	Delay time, data valid from CLKOUT low	0#	10	0#	6	ns
$t_{d(CLKL-MSH)}$	Delay time, $\overline{MSTRB}$ high from CLKOUT low	– 2	3	– 2	3	ns
$t_{d(CLKH-RWL)}$	Delay time, R/$\overline{W}$ low from CLKOUT high	0	5	– 2	3	ns
$t_{d(CLKH-RWH)}$	Delay time, R/$\overline{W}$ high from CLKOUT high	– 2	3	– 2	3	ns
$t_{d(RWL-MSTRBL)}$	Delay time, $\overline{MSTRB}$ low after R/$\overline{W}$ low#	H – 2	H + 3	H – 2	H + 3	ns
$t_{h(A)W}$	Hold time, address valid after CLKOUT high§	0	5	0	5	ns

PARAMETER		'LC54x-80		'VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{d(CLKH-A)}$	Delay time, address valid from CLKOUT high§	– 2#	3	– 2#	3	ns
$t_{d(CLKL-A)}$	Delay time, address valid from CLKOUT low¶	0#	4	0#	3	ns
$t_{d(CLKL-MSL)}$	Delay time, $\overline{MSTRB}$ low from CLKOUT low	0	4	0	3	ns
$t_{d(CLKL-D)W}$	Delay time, data valid from CLKOUT low	0#	5	0#	3	ns
$t_{d(CLKL-MSH)}$	Delay time, $\overline{MSTRB}$ high from CLKOUT low	– 2	3	– 2	3	ns
$t_{d(CLKH-RWL)}$	Delay time, R/$\overline{W}$ low from CLKOUT high	– 2	3	– 2	3	ns
$t_{d(CLKH-RWH)}$	Delay time, R/$\overline{W}$ high from CLKOUT high	– 2	3	– 2	3	ns
$t_{d(RWL-MSTRBL)}$	Delay time, $\overline{MSTRB}$ low after R/$\overline{W}$ low#	H – 2	H + 2	H – 2	H + 2	ns
$t_{h(A)W}$	Hold time, address valid after CLKOUT high§	0	3	0	3	ns

† Address, $\overline{PS}$, and $\overline{DS}$ timings are all included in timings referenced as address.
‡ See Table 15, Table 16, and Table 17 for address bus timing variation with load capacitance.
§ In the case of a memory write preceded by a memory write.
¶ In the case of a memory write preceded by an I/O cycle.
# Values derived from characterization data and not tested.

memory and parallel I/O interface timing (continued)

timing requirements over recommended operating conditions for a memory write ($\overline{\text{MSTRB}}$ = 0) [H = 0.5 $t_{c(CO)}$]†‡ (see Figure 17)

		’C54x-40 ’LC54x-40 ’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	
$t_{h(D)MSH}$	Hold time, write data valid after $\overline{\text{MSTRB}}$ high	H–5	H+5§¶	H–5	H+5§¶	ns
$t_{w(SL)MS}$	Pulse duration, $\overline{\text{MSTRB}}$ low¶	2H–5		2H–5		ns
$t_{su(A)W}$	Setup time, address valid before $\overline{\text{MSTRB}}$ low	2H–5		2H–5		ns
$t_{su(D)MSH}$	Setup time, write data valid before $\overline{\text{MSTRB}}$ high	2H–10	2H+10§¶	2H–10	2H+8§¶	ns

		’LC54x-80		’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{h(D)MSH}$	Hold time, write data valid after $\overline{\text{MSTRB}}$ high	H–4	H+4§¶	H–3	H+3§¶	ns
$t_{w(SL)MS}$	Pulse duration, $\overline{\text{MSTRB}}$ low¶	2H–5		2H–4		ns
$t_{su(A)W}$	Setup time, address valid before $\overline{\text{MSTRB}}$ low	2H–5		2H–4		ns
$t_{su(D)MSH}$	Setup time, write data valid before $\overline{\text{MSTRB}}$ high	2H–7	2H+7§¶	2H–5	2H+5§¶	ns

† Address, $\overline{\text{PS}}$, and $\overline{\text{DS}}$ timings are all included in timings referenced as address.
‡ See Table 15, Table 16, and Table 17 for address bus timing variation with load capacitance.
§ In the case of a memory write preceded by an I/O cycle.
¶ Values derived from characterization data and not tested.

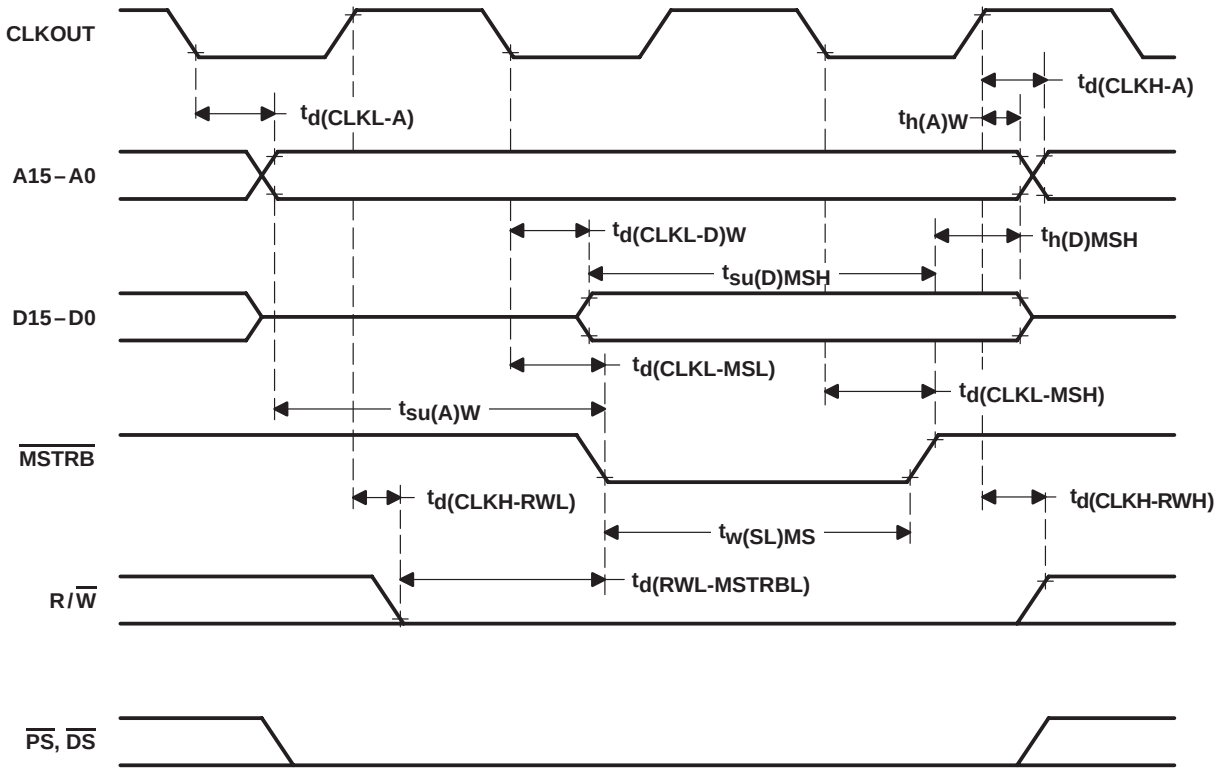


Figure 17. Memory Write ($\overline{\text{MSTRB}}$ = 0)

memory and parallel I/O interface timing (continued)

switching characteristics over recommended operating conditions for a parallel I/O port read ($\overline{IOSTRB}$ = 0) [H = 0.5 $t_{c(CO)}$]†‡ (see Figure 18)

PARAMETER		’LC542-40 ’LC543-40		’C54x-40 ’LC54x-40 ’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{d(CLKL-A)}$	Delay time, address valid from CLKOUT low	0§	5	0§	5	0§	5	ns
$t_{d(CLKH-ISTRBL)}$	Delay time, $\overline{IOSTRB}$ low from CLKOUT high	0	5	0	5	– 2	3	ns
$t_{d(CLKH-ISTRBH)}$	Delay time, $\overline{IOSTRB}$ high from CLKOUT high	– 2	3	– 2	3	– 2	3	ns
$t_{h(A)IOR}$	Hold time, address after CLKOUT low	0	5§	0	5§	0	5§	ns

PARAMETER		’LC54x-80		’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{d(CLKL-A)}$	Delay time, address valid from CLKOUT low	0§	4	0§	3	ns
$t_{d(CLKH-ISTRBL)}$	Delay time, $\overline{IOSTRB}$ low from CLKOUT high	– 2	3	– 2	3	ns
$t_{d(CLKH-ISTRBH)}$	Delay time, $\overline{IOSTRB}$ high from CLKOUT high	– 2	3	– 2	3	ns
$t_{h(A)IOR}$	Hold time, address after CLKOUT low	0	4§	0	3§	ns

† Address and $\overline{IS}$ timings are included in timings referenced as address.
‡ See Table 15, Table 16, and Table 17 for address bus timing variation with load capacitance.
§ Values derived from characterization data and not tested.

memory and parallel I/O interface timing (continued)

timing requirements over recommended operating conditions for a parallel I/O port read ($\overline{IOSTRB}$ = 0) [H = 0.5 $t_{c(CO)}$]†‡ (see Figure 18)

		’LC542-40 ’LC543-40		’C54x-40 ’LC54x-40 ’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{a(A)IO}$	Access time, read data access from address valid		3H–12		3H–10		3H–10	ns
$t_{a(ISTRBL)IO}$	Access time, read data access from $\overline{IOSTRB}$ low		2H–12		2H–10		2H–10	ns
$t_{su(D)IOR}$	Setup time, read data before CLKOUT high	7		5		5		ns
$t_{h(D)IOR}$	Hold time, read data after CLKOUT high	0		0		0		ns
$t_{h(ISTRBH-D)R}$	Hold time, read data after $\overline{IOSTRB}$ high	0		0		0		ns

		’LC54x-80		’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{a(A)IO}$	Access time, read data access from address valid		3H–5		3H–3	ns
$t_{a(ISTRBL)IO}$	Access time, read data access from $\overline{IOSTRB}$ low		2H–5		2H–3	ns
$t_{su(D)IOR}$	Setup time, read data before CLKOUT high	4		4		ns
$t_{h(D)IOR}$	Hold time, read data after CLKOUT high	0		0		ns
$t_{h(ISTRBH-D)R}$	Hold time, read data after $\overline{IOSTRB}$ high	0		0		ns

† Address and $\overline{IS}$ timings are included in timings referenced as address.
‡ See Table 15, Table 16, and Table 17 for address bus timing variation with load capacitance.

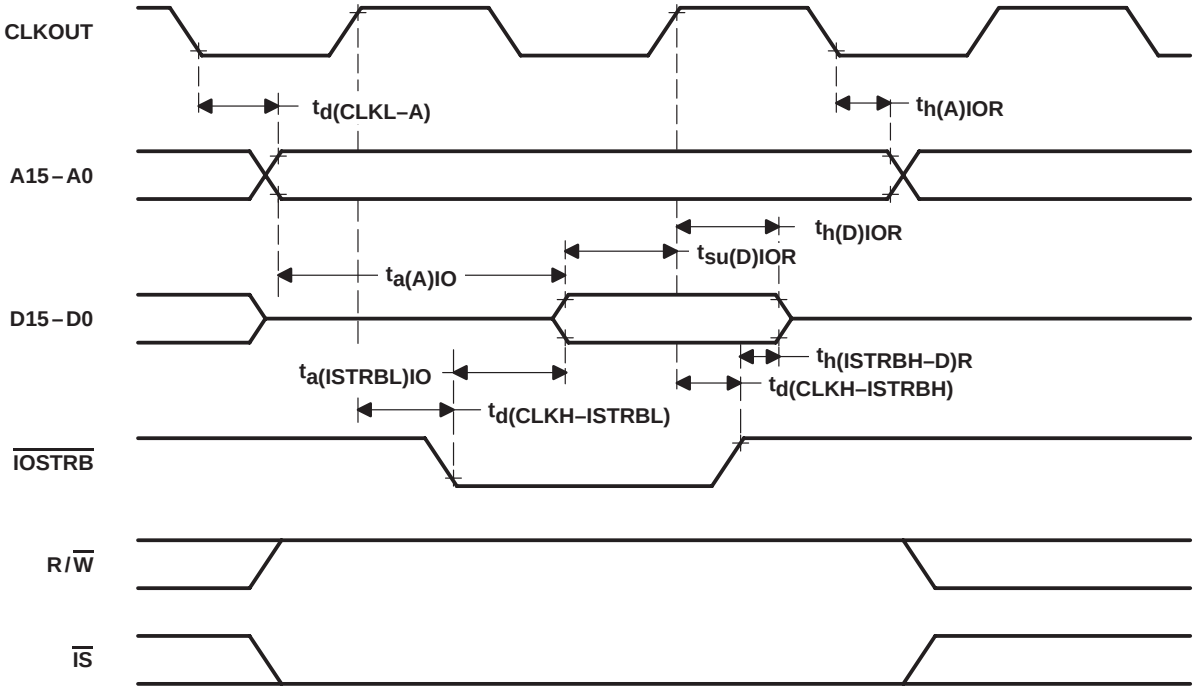


Figure 18. Parallel I/O Port Read ($\overline{IOSTRB}$ = 0)

memory and parallel I/O interface timing (continued)

switching characteristics over recommended operating conditions for a parallel I/O port write ($\overline{IOSTRB}$ = 0) [H = 0.5 $t_{c(CO)}$] (see Figure 19)†

PARAMETER		’C54x-40 ’LC54x-40 ’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	
$t_{d(CLKL-A)}$	Delay time, address valid from CLKOUT low‡	0§	5	0§	5	ns
$t_{d(CLKH-ISTRBL)}$	Delay time, $\overline{IOSTRB}$ low from CLKOUT high	0	5	–2	3	ns
$t_{d(CLKH-D)IOW}$	Delay time, write data valid from CLKOUT high	H–5§	H+10	H–5§	H+8	ns
$t_{d(CLKH-ISTRBH)}$	Delay time, $\overline{IOSTRB}$ high from CLKOUT high	– 2	3	–2	3	ns
$t_{d(CLKL-RWL)}$	Delay time, R/$\overline{W}$ low from CLKOUT low	0	5	0	5	ns
$t_{d(CLKL-RWH)}$	Delay time, R/$\overline{W}$ high from CLKOUT low	– 2	3	–2	3	ns
$t_{h(A)IOW}$	Hold time, address valid from CLKOUT low‡	0	5§	0	5§	ns
$t_{h(D)IOW}$	Hold time, write data after $\overline{IOSTRB}$ high	H–5	H+5§	H–5	H+5§	ns
$t_{su(D)IOSTRBH}$	Setup time, write data before $\overline{IOSTRB}$ high	H–7	H	H–5	H	ns
$t_{su(A)IOSTRBL}$	Setup time, address valid before $\overline{IOSTRB}$ low§	H–5	H+5	H–5	H+5	ns

PARAMETER		’LC54x-80		’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{d(CLKL-A)}$	Delay time, address valid from CLKOUT low‡	0§	4	0§	3	ns
$t_{d(CLKH-ISTRBL)}$	Delay time, $\overline{IOSTRB}$ low from CLKOUT high	– 2	3	– 2	3	ns
$t_{d(CLKH-D)IOW}$	Delay time, write data valid from CLKOUT high	H–5§	H+5	H–5§	H+3	ns
$t_{d(CLKH-ISTRBH)}$	Delay time, $\overline{IOSTRB}$ high from CLKOUT high	– 2	3	–2	3	ns
$t_{d(CLKL-RWL)}$	Delay time, R/$\overline{W}$ low from CLKOUT low	0	4	0	3	ns
$t_{d(CLKL-RWH)}$	Delay time, R/$\overline{W}$ high from CLKOUT low	– 2	2	–2	2	ns
$t_{h(A)IOW}$	Hold time, address valid from CLKOUT low‡	0	4§	0	3§	ns
$t_{h(D)IOW}$	Hold time, write data after $\overline{IOSTRB}$ high	H–4	H+4§	H–3	H+3§	ns
$t_{su(D)IOSTRBH}$	Setup time, write data before $\overline{IOSTRB}$ high	H–4	H+1	H–3	H+1	ns
$t_{su(A)IOSTRBL}$	Setup time, address valid before $\overline{IOSTRB}$ low§	H–5	H+5	H–3	H+3	ns

† See Table 15, Table 16, and Table 17 for address bus timing variation with load capacitance.
‡ Address and $\overline{IS}$ timings are included in timings referenced as address.
§ Values derived from characterization data and not tested

memory and parallel I/O interface timing (continued)

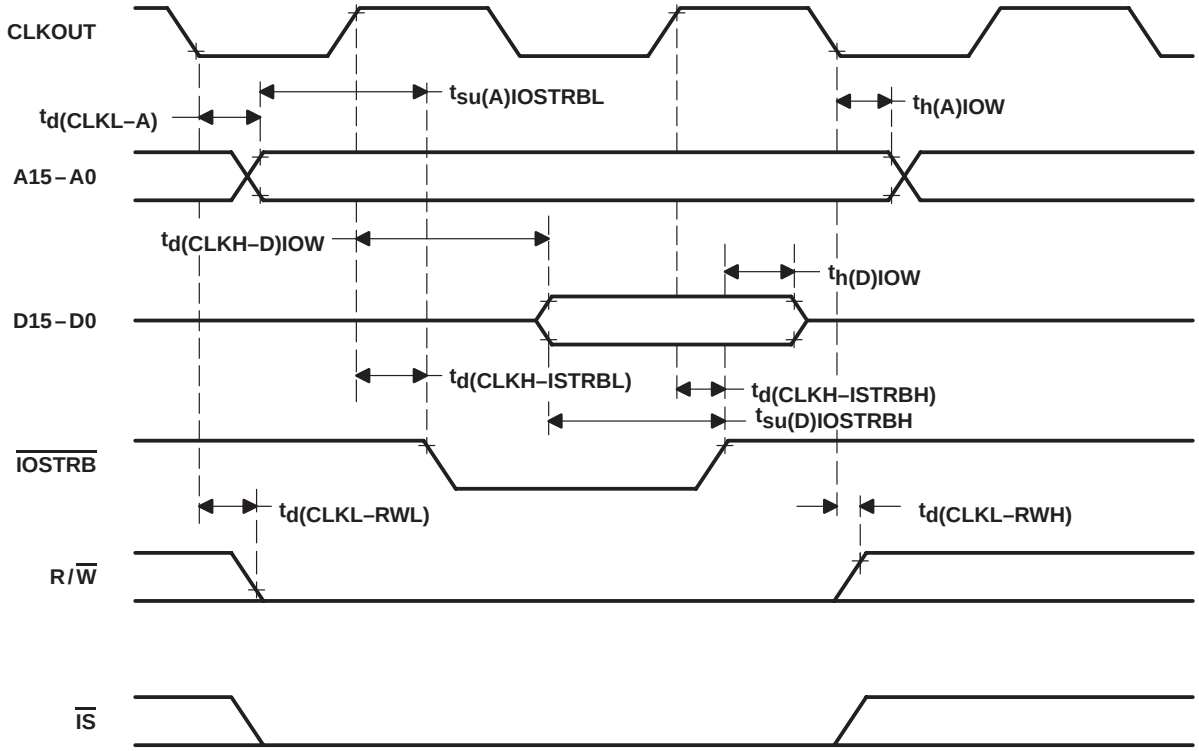


Figure 19. Parallel I/O Port Write ($\overline{\text{IOSTRB}}$ = 0)

I/O timing variation with load capacitance: SPICE simulation results

Condition:	**Temperature**	**: 125° C**
	Capacitance	**: 0 – 100pF**
	Voltage	**: 2.7 / 3.0 / 3.3 V**
	Model	**: Weak / Nominal / Strong**

Figure 20. Rise and Fall Time Diagram

I/O timing variation with load capacitance: SPICE simulation results (continued)

Table 15. Timing Variation With Load Capacitance: [2.7 V] 10% – 90%

	WEAK		NOMINAL		STRONG	
	RISE	FALL	RISE	FALL	RISE	FALL
0 pF	0.476 ns	0.457 ns	0.429 ns	0.391 ns	0.382 ns	0.323 ns
10 pF	1.511 ns	1.278 ns	1.386 ns	1.148 ns	1.215 ns	1.049 ns
20 pF	2.551 ns	2.133 ns	2.350 ns	1.956 ns	2.074 ns	1.779 ns
30 pF	3.614 ns	3.011 ns	3.327 ns	2.762 ns	2.929 ns	2.512 ns
40 pF	4.664 ns	3.899 ns	4.394 ns	3.566 ns	3.798 ns	3.264 ns
50 pF	5.752 ns	4.786 ns	5.273 ns	4.395 ns	4.655 ns	4.010 ns
60 pF	6.789 ns	5.656 ns	6.273 ns	5.206 ns	5.515 ns	4.750 ns
70 pF	7.817 ns	6.598 ns	7.241 ns	6.000 ns	6.442 ns	5.487 ns
80 pF	8.897 ns	7.531 ns	8.278 ns	6.928 ns	7.262 ns	6.317 ns
90 pF	10.021 ns	8.332 ns	9.152 ns	7.735 ns	8.130 ns	7.066 ns
100 pF	11.072 ns	9.299 ns	10.208 ns	8.537 ns	8.997 ns	7.754 ns

Table 16. Timing Variation With Load Capacitance: [3 V] 10% – 90%

	WEAK		NOMINAL		STRONG	
	RISE	FALL	RISE	FALL	RISE	FALL
0 pF	0.436 ns	0.387 ns	0.398 ns	0.350 ns	0.345 ns	0.290 ns
10 pF	1.349 ns	1.185 ns	1.240 ns	1.064 ns	1.092 ns	0.964 ns
20 pF	2.273 ns	1.966 ns	2.098 ns	1.794 ns	1.861 ns	1.634 ns
30 pF	3.226 ns	2.765 ns	2.974 ns	2.539 ns	2.637 ns	2.324 ns
40 pF	4.168 ns	3.573 ns	3.849 ns	3.292 ns	3.406 ns	3.013 ns
50 pF	5.110 ns	4.377 ns	4.732 ns	4.052 ns	4.194 ns	3.710 ns
60 pF	6.033 ns	5.230 ns	5.660 ns	4.811 ns	5.005 ns	4.401 ns
70 pF	7.077 ns	5.997 ns	6.524 ns	5.601 ns	5.746 ns	5.117 ns
80 pF	8.020 ns	6.899 ns	7.416 ns	6.336 ns	6.559 ns	5.861 ns
90 pF	8.917 ns	7.709 ns	8.218 ns	7.124 ns	7.323 ns	6.498 ns
100 pF	9.885 ns	8.541 ns	9.141 ns	7.830 ns	8.101 ns	7.238 ns

ADVANCE INFORMATION

I/O timing variation with load capacitance: SPICE simulation results (continued)

Table 17. Timing Variation With Load Capacitance: [3.3 V] 10% – 90% [3 V] 10% – 90%

	WEAK		NOMINAL		STRONG	
	RISE	FALL	RISE	FALL	RISE	FALL
0 pF	0.404 ns	0.361 ns	0.371 ns	0.310 ns	0.321 ns	0.284 ns
10 pF	1.227 ns	1.081 ns	1.133 ns	1.001 ns	1.000 ns	0.892 ns
20 pF	2.070 ns	1.822 ns	1.915 ns	1.675 ns	1.704 ns	1.530 ns
30 pF	2.931 ns	2.567 ns	2.719 ns	2.367 ns	2.414 ns	2.169 ns
40 pF	3.777 ns	3.322 ns	3.515 ns	3.072 ns	3.120 ns	2.823 ns
50 pF	4.646 ns	4.091 ns	4.319 ns	3.779 ns	3.842 ns	3.466 ns
60 pF	5.487 ns	4.859 ns	5.145 ns	4.503 ns	4.571 ns	4.142 ns
70 pF	6.405 ns	5.608 ns	5.980 ns	5.234 ns	5.301 ns	4.767 ns
80 pF	7.284 ns	6.463 ns	6.723 ns	5.873 ns	5.941 ns	5.446 ns
90 pF	8.159 ns	7.097 ns	7.560 ns	6.692 ns	6.740 ns	6.146 ns
100 pF	8.994 ns	7.935 ns	8.300 ns	7.307 ns	7.431 ns	6.822 ns

ADVANCE INFORMATION

ready timing for externally generated wait states

timing requirements over recommended operating conditions for externally generated wait states [H = 0.5 $t_{c(CO)}$]† (see Figure 21, Figure 22, Figure 23, and Figure 24)

		’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{su(RDY)}$	Setup time, READY before CLKOUT low	10		8		7		ns
$t_{h(RDY)}$	Hold time, READY after CLKOUT low	0		0		0		ns
$t_{v(RDY)MSTRB}$	Valid time, READY after $\overline{MSTRB}$ low§		4H – 15		4H – 12		4H–10	ns
$t_{h(RDY)MSTRB}$	Hold time, READY after $\overline{MSTRB}$ low§	4H		4H		4H		ns
$t_{v(RDY)IOSTRB}$	Valid time, READY after $\overline{IOSTRB}$ low§		5H – 15		5H – 12		5H–10	ns
$t_{h(RDY)IOSTRB}$	Hold time, READY after $\overline{IOSTRB}$ low§	5H		5H		5H		ns
$t_{v(MSCL)}$	Valid time, $\overline{MSC}$ low after CLKOUT low	0‡	5	0‡	5	0‡	5	ns
$t_{v(MSCH)}$	Valid time, $\overline{MSC}$ high after CLKOUT low	–2‡	3	–2‡	3	–2‡	3	ns

		’LC54x-80		’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{su(RDY)}$	Setup time, READY before CLKOUT low	6		5		ns
$t_{h(RDY)}$	Hold time, READY after CLKOUT low	0		0		ns
$t_{v(RDY)MSTRB}$	Valid time, READY after $\overline{MSTRB}$ low§		4H – 10		4H–8	ns
$t_{h(RDY)MSTRB}$	Hold time, READY after $\overline{MSTRB}$ low§	4H		4H		ns
$t_{v(RDY)IOSTRB}$	Valid time, READY after $\overline{IOSTRB}$ low§		5H – 10		5H–8	ns
$t_{h(RDY)IOSTRB}$	Hold time, READY after $\overline{IOSTRB}$ low§	5H		5H		ns
$t_{v(MSCL)}$	Valid time, $\overline{MSC}$ low after CLKOUT low	0‡	4	0‡	3	ns
$t_{v(MSCH)}$	Valid time, $\overline{MSC}$ high after CLKOUT low	–2‡	2	–2‡	2	ns

† The hardware wait states can be used only in conjunction with the software wait states to extend the bus cycles. To generate wait states by READY, at least two software wait states must be programmed. READY is not sampled until the completion of the internal software wait states.
‡ Values derived from characterization data and not tested.
§ These timings are included for reference only. The critical timings for READY are those referenced to CLKOUT.

ready timing for externally generated wait states (continued)

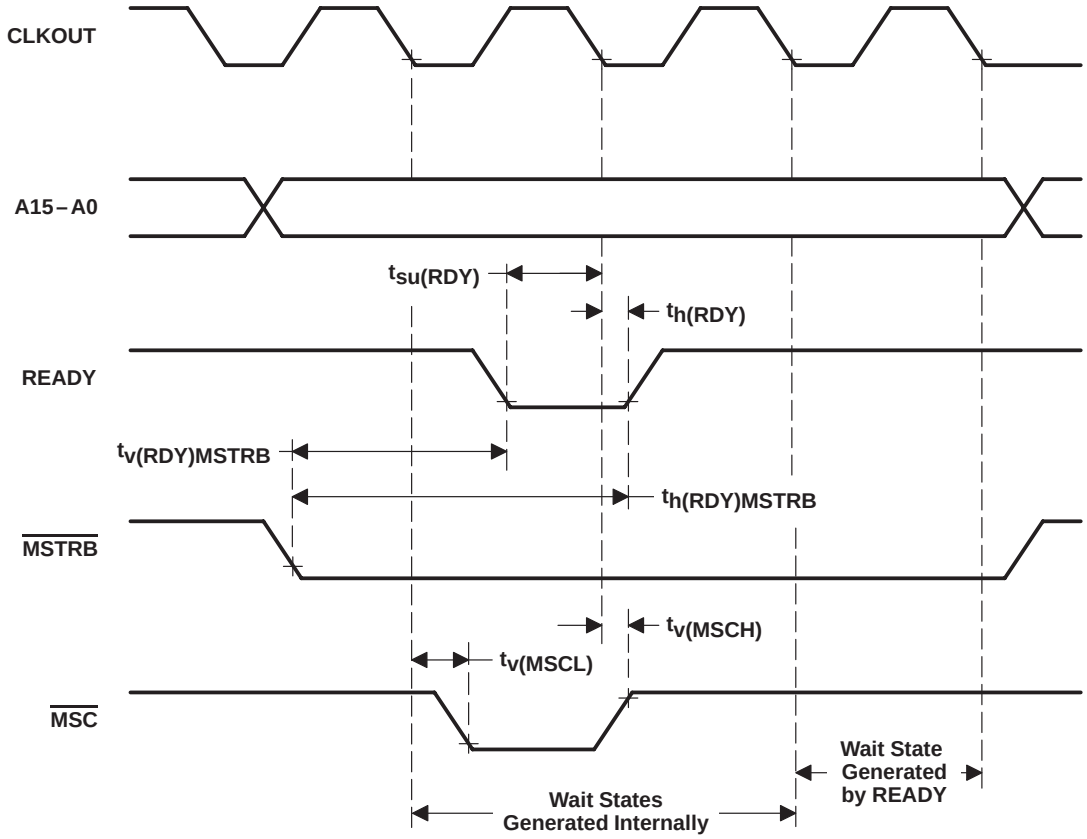


Figure 21. Memory Read With Externally Generated Wait States

ADVANCE INFORMATION

ready timing for externally generated wait states (continued)

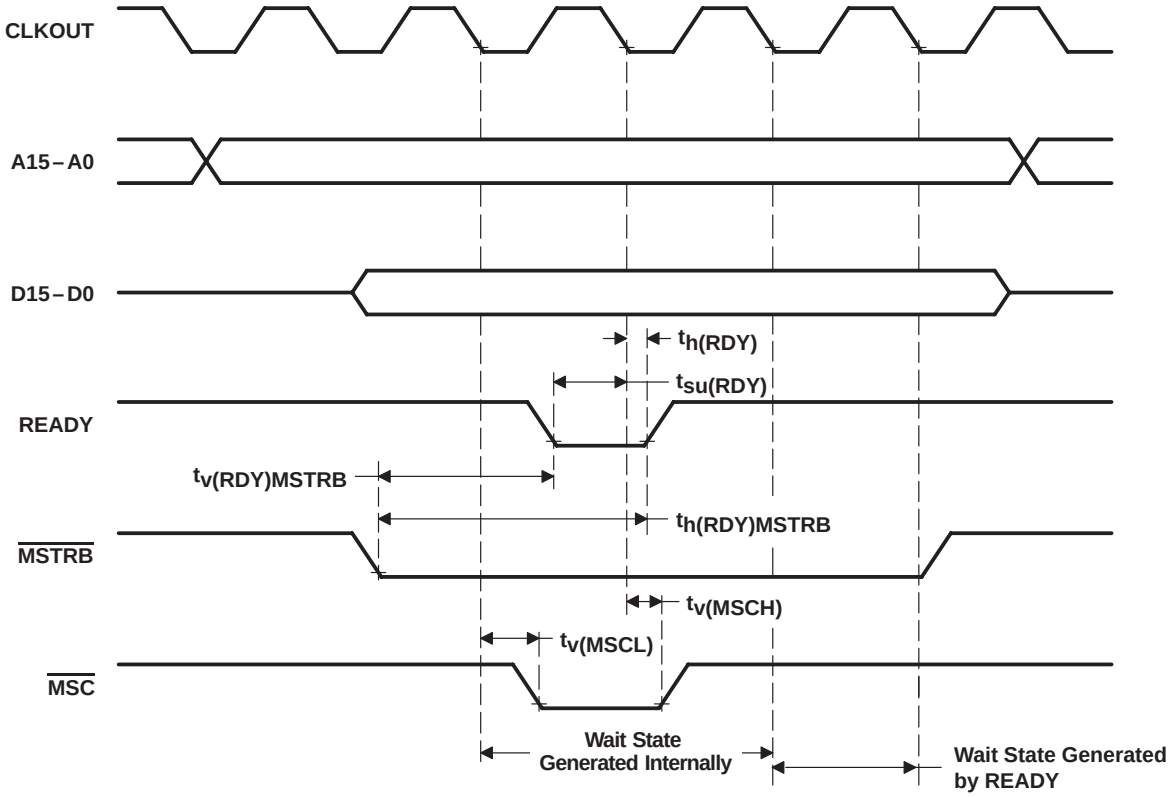


Figure 22. Memory Write With Externally Generated Wait States

ready timing for externally generated wait states (continued)

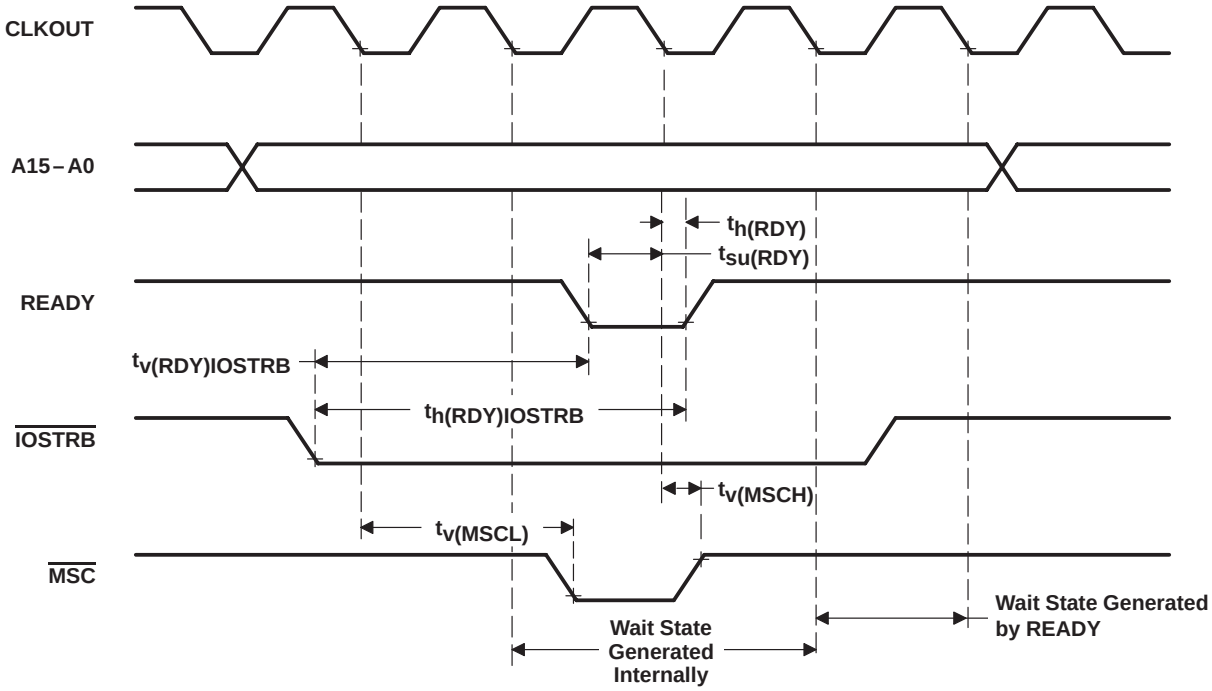


Figure 23. I/O Read With Externally Generated Wait States

ready timing for externally generated wait states (continued)

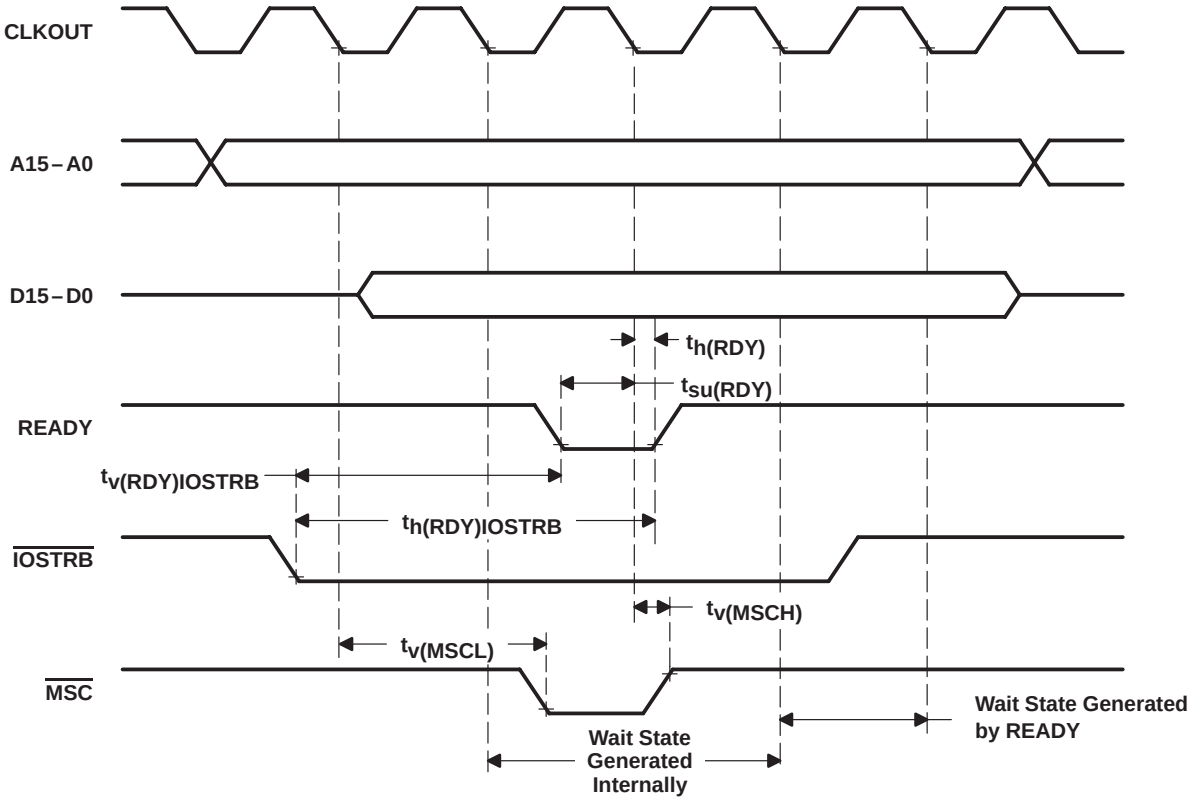


Figure 24. I/O Write With Externally Generated Wait States

HOLD and HOLDA timing

switching characteristics over recommended operating conditions for memory control signals and HOLDA [H = 0.5 $t_{c(CO)}$] (see Figure 25)

PARAMETER		’C54x-40 ’LC54x-40 ’LC54x-50		’54x-66		’LC54x-80 ’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{dis(CLKL-A)}$	Disable time, CLKOUT low to address, $\overline{PS}$, $\overline{DS}$, $\overline{IS}$ high impedance‡		5		5		5	ns
$t_{dis(CLKL-RW)}$	Disable time, CLKOUT low to R/$\overline{W}$ high impedance ‡		5		5		5	ns
$t_{dis(CLKL-S)}$	Disable time, CLKOUT low to $\overline{MSTRB}$, $\overline{IOSTRB}$ high impedance‡		5		5		5	ns
$t_{en(CLKL-A)}$	Enable time, CLKOUT low to address, $\overline{PS}$, $\overline{DS}$, $\overline{IS}$‡		2H+5		2H+5		2H+5	ns
$t_{en(CLKL-RW)}$	Enable time, CLKOUT low to R/$\overline{W}$ enabled‡		2H+5		2H+5		2H+5	ns
$t_{en(CLKL-S)}$	Enable time, CLKOUT low to $\overline{MSTRB}$, $\overline{IOSTRB}$ enabled‡		2H+5		2H+5	2	2H+5	ns
$t_{v(HOLDA)}$	Valid time, $\overline{HOLDA}$ low after CLKOUT low	– 2	5	0†	5	0†	5	ns
	Valid time, $\overline{HOLDA}$ high after CLKOUT low	– 2	5	– 2	3†	– 2	3†	ns
$t_{w(HOLDA)}$	Pulse duration, $\overline{HOLDA}$ low duration	2H–3		2H–3		2H–3		ns

† Values derived from characterization data and not tested.
‡ Values assured by design but not tested.

timing requirements over recommended operating conditions for HOLD [H = 0.5 $t_{c(CO)}$] (see Figure 25)

		’C54x-40 ’LC54x-40 ’LC54x-50		’54x-66		’LC54x-80 ’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	MAX	MIN	
$t_{w(HOLD)}$	Pulse duration, $\overline{HOLD}$ low duration	4H+10		4H+10		4H+10		ns
$t_{su(HOLD)}$	Setup time, $\overline{HOLD}$ before CLKOUT low	10		10		10		ns

$\overline{\text{HOLD}}$ and $\overline{\text{HOLDA}}$ timing (continued)

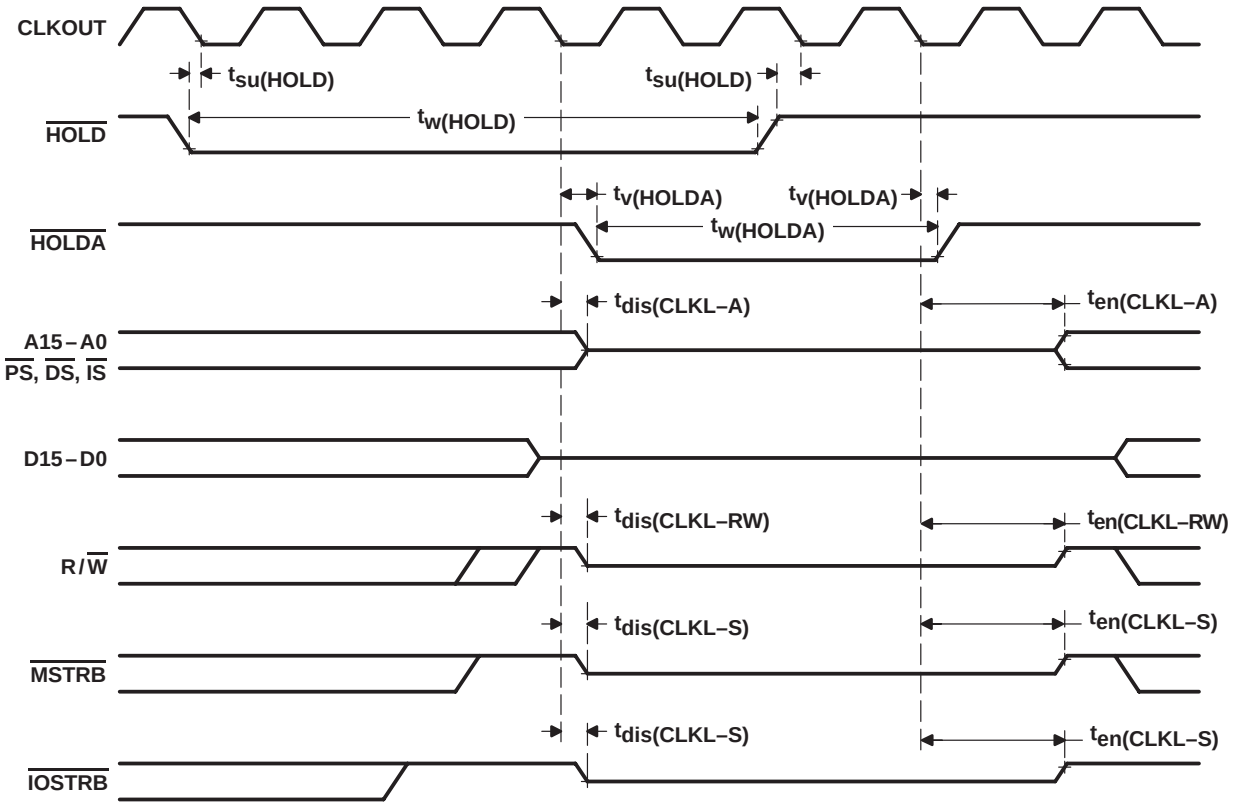


Figure 25. $\overline{\text{HOLD}}$ and $\overline{\text{HOLDA}}$ Timing (HM = 1)

ADVANCE INFORMATION

reset, $\overline{BIO}$, interrupt, and MP/$\overline{MC}$ timings

timing requirements over recommended operating conditions for reset, interrupt, $\overline{BIO}$, and MP/$\overline{MC}$ [H = 0.5 $t_{c(CO)}$] (see Figure 26, Figure 27, and Figure 28)

		’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{h(RS)}$	Hold time, $\overline{RS}$ after CLKOUT low	0		0		0		ns
$t_{h(BIO)}$	Hold time, $\overline{BIO}$ after CLKOUT low	0		0		0		ns
$t_{h(INT)}$	Hold time, $\overline{INTn}$, $\overline{NMI}$, after CLKOUT low†	0		0		0		ns
$t_{h(MPMC)}$	Hold time, MP/$\overline{MC}$ after CLKOUT low‡	0		0		0		ns
$t_{w(RSL)}$	Pulse duration, $\overline{RS}$ low§¶‖	4H + 10		4H + 10		4H+10		ns
$t_{w(BIO)S}$	Pulse duration, $\overline{BIO}$ low, synchronous¶	2H+15		2H+12		2H+10		ns
$t_{w(BIO)A}$	Pulse duration, $\overline{BIO}$ low, asynchronous‡	4H		4H		4H		ns
$t_{w(INTH)S}$	Pulse duration, $\overline{INTn}$, $\overline{NMI}$ high (synchronous)¶	2H+15		2H+12		2H+10		ns
$t_{w(INTH)A}$	Pulse duration, $\overline{INTn}$, $\overline{NMI}$ high (asynchronous)‡	4H		4H		4H		ns
$t_{w(INTL)S}$	Pulse duration, $\overline{INTn}$, $\overline{NMI}$ low (synchronous)¶	2H+15		2H+12		2H+10		ns
$t_{w(INTL)A}$	Pulse duration, $\overline{INTn}$, $\overline{NMI}$ low (asynchronous)‡	4H		4H		4H		ns
$t_{w(INTL)WKP}$	Pulse duration, $\overline{INTn}$, $\overline{NMI}$ low for IDLE2/IDLE3 wakeup‡	10		10		10		ns
$t_{su(RS)}$	Setup time, $\overline{RS}$ before X2/CLKIN low#	5		5		5		ns
$t_{su(BIO)}$	Setup time, $\overline{BIO}$ before CLKOUT low	15		12		10		ns
$t_{su(INT)}$	Setup time, $\overline{INTn}$, $\overline{NMI}$, $\overline{RS}$ before CLKOUT low	15		12		10		ns
$t_{su(MPMC)}$	Setup time, MP/$\overline{MC}$ before CLKOUT low‡	10		10		10		ns

† The external interrupts ($\overline{INT0}$ – $\overline{INT3}$, $\overline{NMI}$) are synchronized to the core CPU by way of a two flip-flop synchronizer which samples these inputs with consecutive falling edges of CLKOUT. The input to the interrupt pins is required to represent a 1–0–0 sequence at the timing that is corresponding to three CLKOUTs sampling sequence.

‡ Values assured by design but not tested.

§ If the PLL mode is selected, then at power-on sequence, or at wakeup from IDLE3, $\overline{RS}$ must be held low for at least 50 μs to assure synchronization and lock-in of the PLL.

¶ Values derived from characterization data and not tested.

Divide-by-two mode

‖ Note that $\overline{RS}$ may cause a change in clock frequency, therefore changing the value of H (see the PLL section).

ADVANCE INFORMATION

reset, $\overline{\text{BIO}}$, interrupt, and MP/$\overline{\text{MC}}$ timings (continued)

timing requirements over recommended operating conditions for reset, interrupt, $\overline{\text{BIO}}$, and MP/$\overline{\text{MC}}$ [H = 0.5 $t_{c(CO)}$] (see Figure 26, Figure 27, and Figure 28) (continued)

		’LC54x-80		’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{h(RS)}$	Hold time, $\overline{\text{RS}}$ after CLKOUT low	0		0		ns
$t_{h(BIO)}$	Hold time, $\overline{\text{BIO}}$ after CLKOUT low	0		0		ns
$t_{h(INT)}$	Hold time, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$, after CLKOUT low†	0		0		ns
$t_{h(MPMC)}$	Hold time, MP/$\overline{\text{MC}}$ after CLKOUT low‡	0		0		ns
$t_{w(RSL)}$	Pulse duration, $\overline{\text{RS}}$ low§¶‖	4H+7		4H+5		ns
$t_{w(BIO)S}$	Pulse duration, $\overline{\text{BIO}}$ low, synchronous¶	2H+7		2H+5		ns
$t_{w(BIO)A}$	Pulse duration, $\overline{\text{BIO}}$ low, asynchronous‡	4H		4H		ns
$t_{w(INTH)S}$	Pulse duration, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$ high (synchronous)¶	2H+7		2H+7		ns
$t_{w(INTH)A}$	Pulse duration, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$ high (asynchronous)‡	4H		4H		ns
$t_{w(INTL)S}$	Pulse duration, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$ low (synchronous)¶	2H+7		2H+7		ns
$t_{w(INTL)A}$	Pulse duration, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$ low (asynchronous)‡	4H		4H		ns
$t_{w(INTL)WKP}$	Pulse duration, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$ low for IDLE2/IDLE3 wakeup‡	10		8		ns
$t_{su(RS)}$	Setup time, $\overline{\text{RS}}$ before X2/CLKIN low#	5		5		ns
$t_{su(BIO)}$	Setup time, $\overline{\text{BIO}}$ before CLKOUT low	10		8		ns
$t_{su(INT)}$	Setup time, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$, $\overline{\text{RS}}$ before CLKOUT low	10		8		ns
$t_{su(MPMC)}$	Setup time, MP/$\overline{\text{MC}}$ before CLKOUT low‡	10		8		ns

† The external interrupts ($\overline{\text{INT0}}$ – $\overline{\text{INT3}}$, $\overline{\text{NMI}}$) are synchronized to the core CPU by way of a two flip-flop synchronizer which samples these inputs with consecutive falling edges of CLKOUT. The input to the interrupt pins is required to represent a 1–0–0 sequence at the timing that is corresponding to three CLKOUTs sampling sequence.

‡ Values assured by design but not tested.

§ If the PLL mode is selected, then at power-on sequence, or at wakeup from IDLE3, $\overline{\text{RS}}$ must be held low for at least 50 µs to assure synchronization and lock-in of the PLL.

¶ Values derived from characterization data and not tested.

Divide-by-two mode

‖ Note that $\overline{\text{RS}}$ may cause a change in clock frequency, therefore changing the value of H (see the PLL section).

ADVANCE INFORMATION

reset, $\overline{\text{BIO}}$, interrupt, and MP/$\overline{\text{MC}}$ timings (continued)

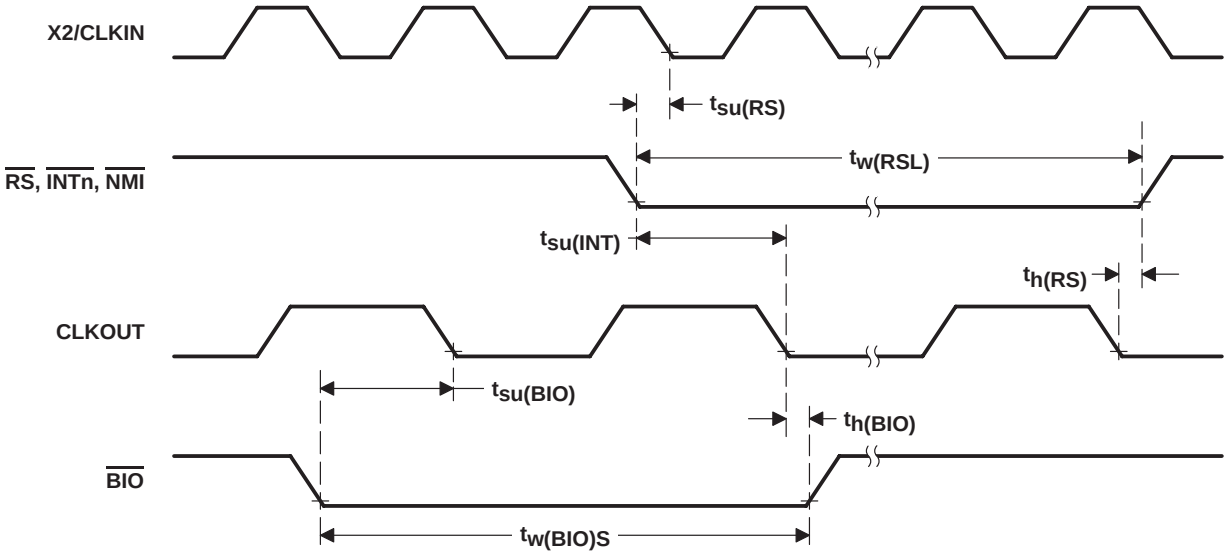


Figure 26. Reset and $\overline{\text{BIO}}$ Timings

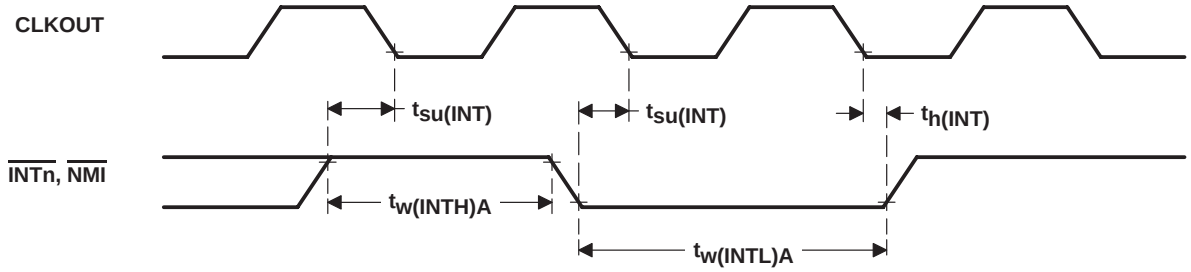


Figure 27. Interrupt Timing

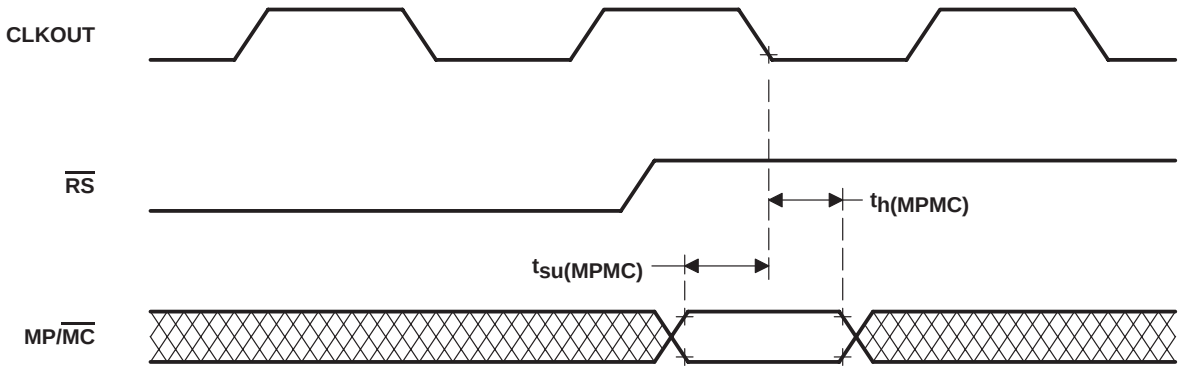


Figure 28. MP/$\overline{\text{MC}}$ Timing

ADVANCE INFORMATION

instruction acquisition ($\overline{IAQ}$), interrupt acknowledge ($\overline{IACK}$), external flag (XF), and TOUT timing

switching characteristics over recommended operating conditions for $\overline{IAQ}$ and $\overline{IACK}$ [H = 0.5 $t_{c(CO)}$] (see Figure 29)

PARAMETER		’C54x-40 ’LC54x-40 ’LC54x-50 ’54x-66		’LC54x-80 ’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{d(CLKL-IAQL)}$	Delay time, $\overline{IAQ}$ low from CLKOUT low	0	5	0	5	ns
$t_{d(CLKL-IAQH)}$	Delay time, $\overline{IAQ}$ high from CLKOUT low	– 2	3	– 2	3	ns
$t_{d(A)IAQ}$	Delay time, address valid before $\overline{IAQ}$ low†		4		4	ns
$t_{d(CLKL-IACKL)}$	Delay time, $\overline{IACK}$ low from CLKOUT low	– 2	3	– 2	3	ns
$t_{d(CLKL-IACKH)}$	Delay time , $\overline{IACK}$ high from CLKOUT low	– 2†	3	– 2†	3	ns
$t_{d(A)IACK}$	Delay time, address valid before $\overline{IACK}$ low†		3		3	ns
$t_{h(A)IAQ}$	Hold time, address valid after $\overline{IAQ}$ high†	0		0		ns
$t_{h(A)IACK}$	Hold time, address valid after $\overline{IACK}$ high†	0		0		ns
$t_{w(IAQL)}$	Pulse duration, $\overline{IAQ}$ low†	2H – 10		2H – 10		ns
$t_{w(IACKL)}$	Pulse duration, $\overline{IACK}$ low†	2H – 10		2H – 10		ns

† Values derived from characterization data and not tested.

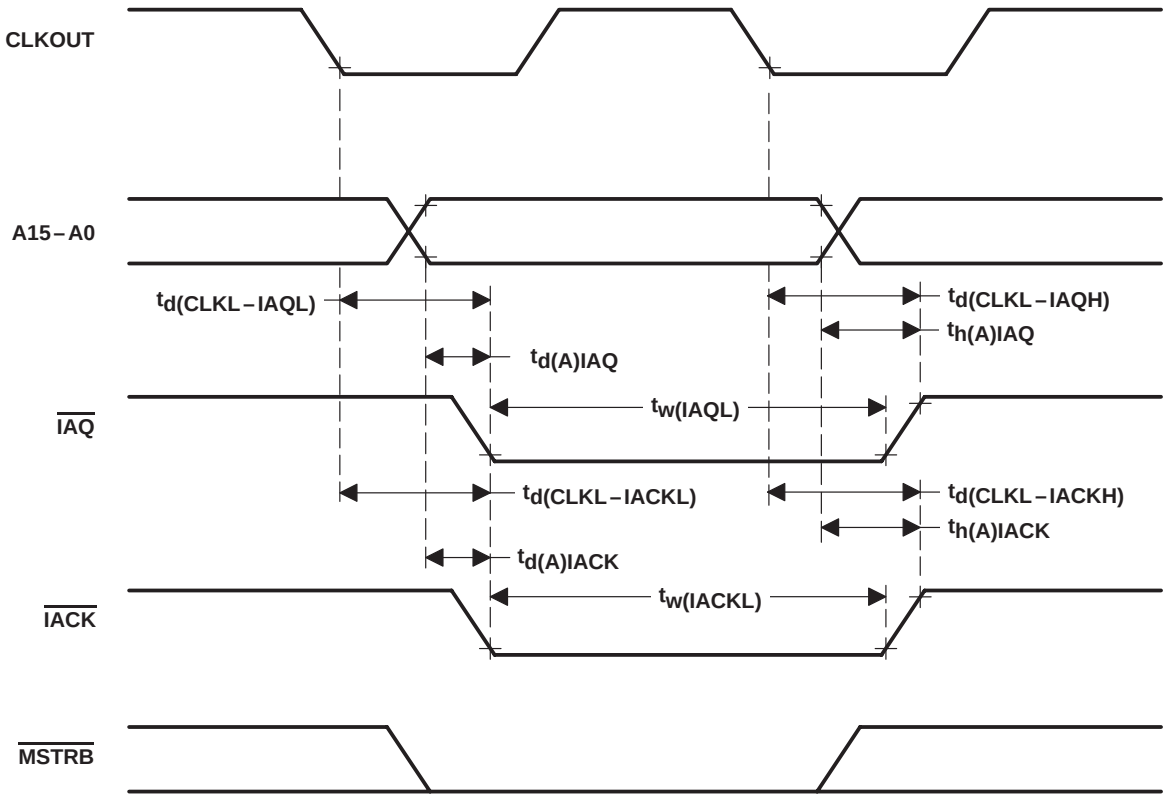


Figure 29. Instruction Acquisition ($\overline{IAQ}$) and Interrupt Acknowledge ($\overline{IACK}$) Timing

instruction acquisition ($\overline{IAQ}$), interrupt acknowledge ($\overline{IACK}$), external flag (XF), and TOUT timing (continued)

switching characteristics over recommended operating conditions for external flag (XF) and TOUT [H = 0.5 $t_{c(CO)}$] (see Figure 30 and Figure 31)

PARAMETER		’C54x-40 ’LC54x-40 ’LC54x-50 ’54x-66		’LC54x-80 ’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{d(XF)}$	Delay time, XF high after CLKOUT low	– 2	3†	– 2	3†	ns
	Delay time, XF low after CLKOUT low	0†	5	0†	5	
$t_{d(TOUTH)}$	Delay time, TOUT high after CLKOUT low	– 2	3	– 2	3	ns
$t_{d(TOUTL)}$	Delay time, TOUT low after CLKOUT low	–2	3	–2	3	ns
$t_{w(TOUT)}$	Pulse duration, TOUT	2H – 10†		2H – 10†		ns

† Values derived from characterization data and not tested.

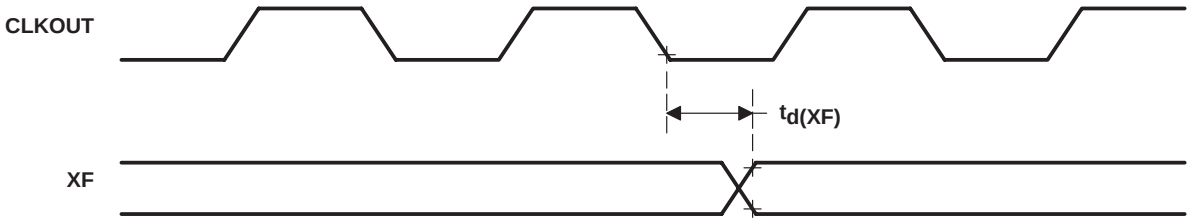


Figure 30. External Flag (XF) Timing

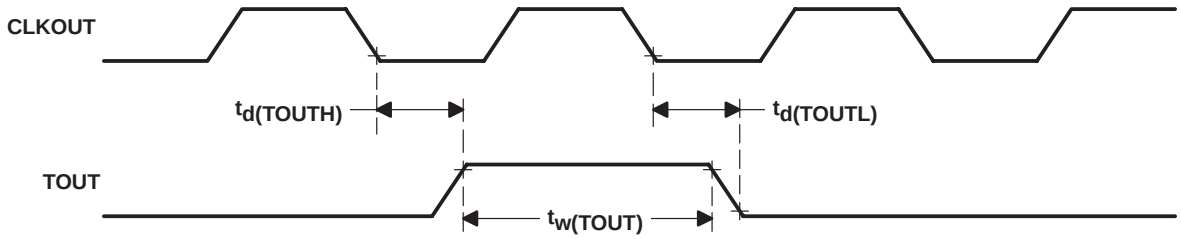


Figure 31. TOUT Timing

serial port receive timing

switching characteristics over recommended operating conditions for serial port receive [H = 0.5 $t_{c(CO)}$] (see Figure 32)

PARAMETER		’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{h(FSR)}$	Hold time, FSR after CLKR falling edge	7		6		6		ns
$t_{h(DR)}$	Hold time, DR after CLKR falling edge	7		6		6		ns

timing requirements over recommended operating conditions for serial port receive [H = 0.5 $t_{c(CO)}$] (see Figure 32)

		’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{c(SCK)}$	Cycle time, serial port clock	6H	†	6H	†	6H	†	ns
$t_{f(SCK)}$	Fall time, serial port clock‡		6		6		6	ns
$t_{r(SCK)}$	Rise time, serial port clock‡		6		6		6	ns
$t_{w(SCK)}$	Pulse duration, serial port clock low/high	3H		3H		3H		ns
$t_{su(FSR)}$	Setup time, FSR before CLKR falling edge	7		6		6		ns
$t_{su(DR)}$	Setup time, DR before CLKR falling edge	7		6		6		ns

† The serial port design is fully static and, therefore, can operate with $t_{c(SCK)}$ approaching ∞. It is characterized approaching an input frequency of 0 Hz but tested at a much higher frequency to minimize test time.
‡ Values assured by design but not tested.

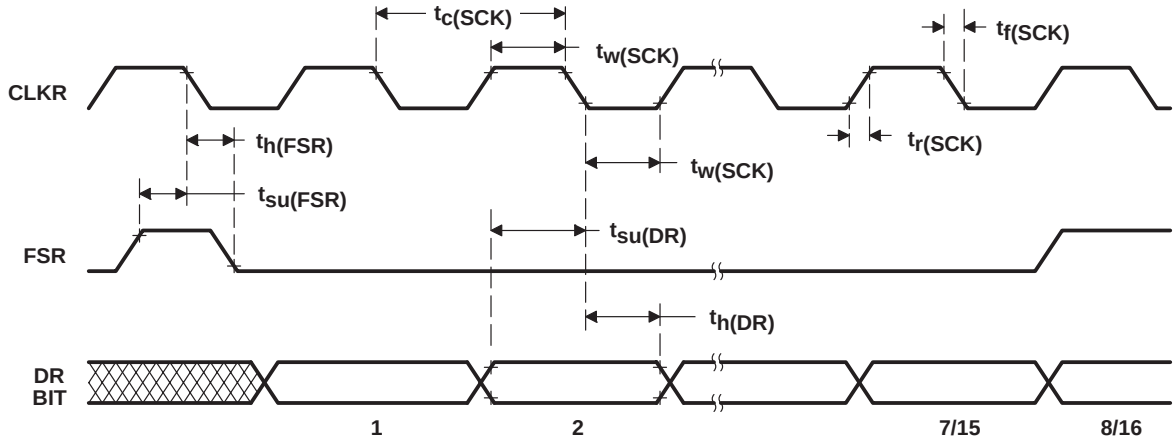


Figure 32. Serial Port Receive Timing

ADVANCE INFORMATION

serial port transmit timing

switching characteristics over recommended operating conditions for serial port transmit with external clocks and frames [H = 0.5$t_{c(CO)}$] (see Figure 33)

PARAMETER		’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{d(DX)}$	Delay time, DX valid after CLKX rising		25		25		25	ns
$t_{d(FSX)}$	Delay time, FSX after CLKX rising edge		2H – 8		2H – 5		2H–5	ns
$t_{dis(DX)}$	Disable time, DX after CLKX rising†		40		40		40	ns

† Values derived from characterization data and not tested

timing requirements over recommended operating conditions for serial port transmit with external clocks and frames [H = 0.5$t_{c(CO)}$] (see Figure 33)

		’C54x-40 ’LC54x-40		’LC54x-50		’54x-66		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{c(SCK)}$	Cycle time, serial port clock	6H	‡	6H	‡	6H	‡	ns
$t_{h(DX)}$	Hold time, DX valid after CLKX rising	–5		–5		– 5		ns
$t_{h(FSX)}$	Hold time, FSX after CLKX falling edge (see Note 1)	7		6		6		ns
$t_{h(FSX)H}$	Hold time, FSX after CLKX rising edge (see Note 1)		2H – 8§		2H – 5§		2H–5§	ns
$t_{f(SCK)}$	Fall time, serial port clock¶		6		6		6	ns
$t_{r(SCK)}$	Rise time, serial port clock¶		6		6		6	ns
$t_{w(SCK)}$	Pulse duration, serial port clock low/high	3H		3H		3H		ns

‡ The serial port design is fully static and, therefore, can operate with $t_{c(SCK)}$ approaching ∞. It is characterized approaching an input frequency of 0 Hz but tested at a much higher frequency to minimize test time.

§ If the FSX pulse does not meet this specification, the first bit of serial data is driven on DX until the falling edge of FSX. After the falling edge of FSX, data is shifted out on DX pin. The transmit buffer-empty interrupt is generated when the $t_{h(FSX)}$ and $t_{h(FSX)H}$ specification is met.

¶ Values assured by design but not tested.

NOTE 1: Internal clock with external FSX and vice versa are also allowable. However, FSX timings to CLKX always are defined depending on the source of FSX, and CLKX timings always are dependent upon the source of CLKX. Specifically, the relationship of FSX to CLKX is independent of the source of CLKX.

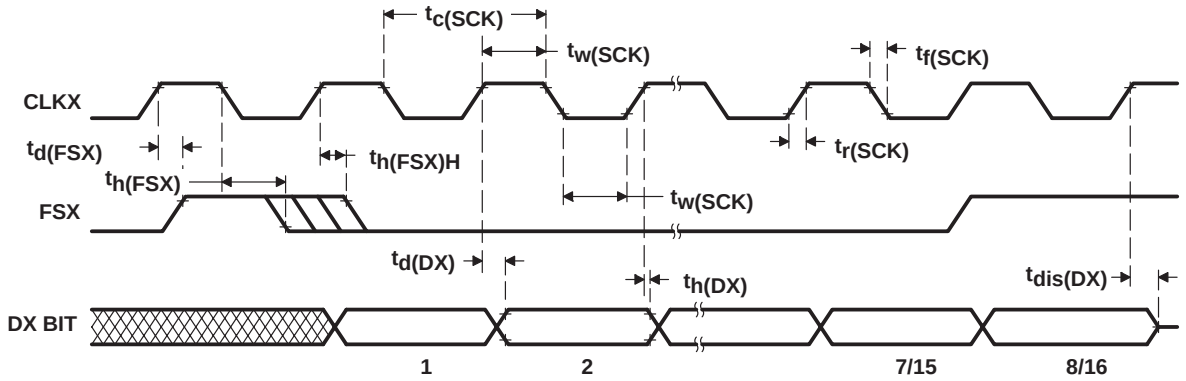


Figure 33. Serial Port Transmit Timing With External Clocks and Frames

serial port transmit timing (continued)

switching characteristics over recommended operating conditions for serial port transmit with internal clocks and frames [H = 0.5$t_{c(CO)}$] (see Figure 34)

PARAMETER		’C54x-40 ’LC54x-40 ’LC54x-50			’54x-66			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
$t_{c(SCK)}$	Cycle time, serial port clock		8H			8H		ns
$t_{d(FSX)}$	Delay time, CLKX rising to FSX			15			15	ns
$t_{d(DX)}$	Delay time, CLKX rising to DX			15			15	ns
$t_{dis(DX)}$	Disable time, CLKX rising to DX†			20			20	ns
$t_{h(DX)}$	Hold time, DX valid after CLKX rising edge	–5			–5			ns
$t_{f(SCK)}$	Fall time, serial port clock		4			4		ns
$t_{r(SCK)}$	Rise time, serial port clock		4			4		ns
$t_{w(SCK)}$	Pulse duration, serial port clock low/high	4H–8			4H–8			ns

† Values derived from characterization data and not tested.

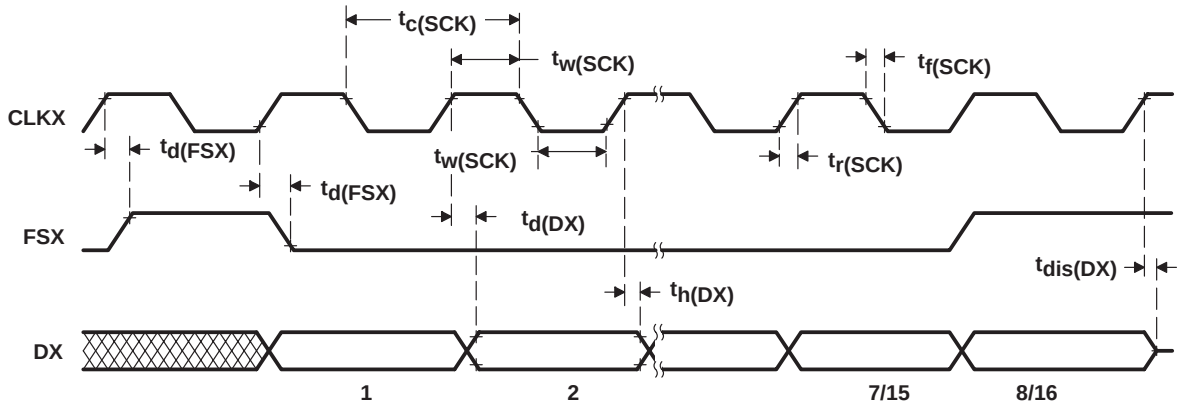


Figure 34. Serial Port Transmit Timing With Internal Clocks and Frames

ADVANCE INFORMATION

buffered serial port receive timing

timing requirements over recommended operating conditions (see Figure 35)

		’C54x-40 ’LC54x-40		’LC54x-50 ’54x-66		’LC54x-80 ’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{c(SCK)}$	Cycle time, serial port clock	25	†	20	†	20	†	ns
$t_{f(SCK)}$	Fall time, serial port clock‡		4		4		4	ns
$t_{r(SCK)}$	Rise time, serial port clock‡		4		4		4	ns
$t_{w(SCK)}$	Pulse duration, serial port clock low/high‡	8.5		6		6		ns
$t_{su(BFSR)}$	Setup time, BFSR before BCLKR falling edge (see Note 2)	2		2		2		ns
$t_{h(BFSR)}$	Hold time, BFSR after BCLKR falling edge (see Note 2)	10	$t_{c(SCK)}-2$§	10	$t_{c(SCK)}-2$§	10	$t_{c(SCK)}-2$§	ns
$t_{su(BDR)}$	Setup time, BDR before BCLKR falling edge	0		0		0		ns
$t_{h(BDR)}$	Hold time, BDR after BCLKR falling edge	10		10		10		ns

† The serial port design is fully static and therefore can operate with $t_{c(SCK)}$ approaching infinity. It is characterized approaching an input frequency of 0 Hz but tested at a much higher frequency to minimize test time.
‡ Values assured by design but not tested.
§ First bit is read when BFSR is sampled low by BCLKR clock.
NOTE 2: Timings for BCLKR and BFSR are given with polarity bits (BCLKP and BFSP) set to 0.

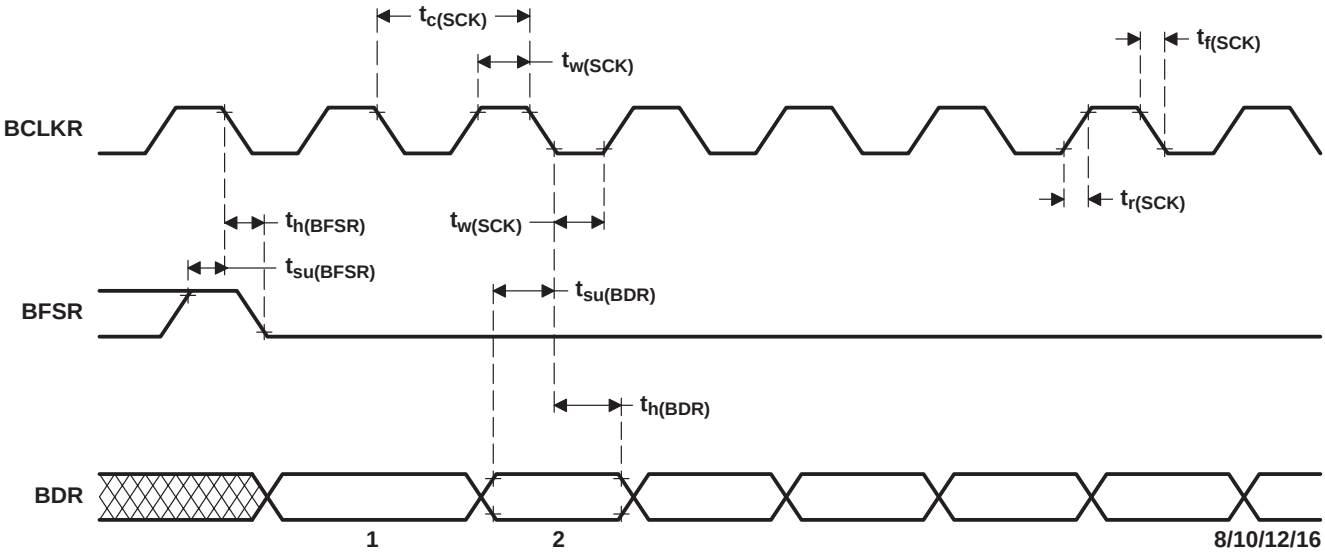


Figure 35. Buffered Serial Port Receive Timing

buffered serial port transmit timing of external frames

switching characteristics over recommended operating conditions (see Figure 36)

PARAMETER		’C54x-40 ’LC54x-40 ’LC54x-50		’54x-66		’LC54x-80 ’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{d(BDX)}$	Delay time, BDX valid after BCLKX rising		18		18		18	ns
$t_{dis(BDX)}$	Disable time, BDX after BCLKX rising†	4	6	4	6	4	6	ns
$t_{dis(BDX)pcm}$	Disable time, PCM mode, BDX after BCLKX rising†		6		6		6	ns
$t_{en(BDX)pcm}$	Enable time, PCM mode, BDX after BCLKX rising†	8		8		8		ns
$t_{h(BDX)}$	Hold time, BDX valid after BCLKX rising	4		2		2		ns

† Values assured by design but not tested.

timing requirements over recommended operating conditions (see Figure 36)

		’C54x-40 ’LC54x-40		’LC54x-50 ’54x-66		’LC54x-80 ’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{c(SCK)}$	Cycle time, serial port clock	25	‡	20	‡	20	‡	ns
$t_{f(SCK)}$	Fall time, serial port clock†		4		4		4	ns
$t_{r(SCK)}$	Rise time, serial port clock†		4		4		4	ns
$t_{w(SCK)}$	Pulse duration, serial port clock low/ high†	8.5		6		6		ns
$t_{h(BFSX)}$	Hold time, BFSX after CLKX falling edge (see Notes 3 and 4)	6	$t_{c(SCK)}-6$§	6	$t_{c(SCK)}-6$§	6	$t_{c(SCK)}-6$§	ns
$t_{su(BFSX)}$	Setup time, FSX before CLKX falling edge (see Notes 3 and 4)	6		6		6		ns

† Values assured by design but not tested.

‡ The serial port design is fully static and therefore can operate with $t_{c(SCK)}$ approaching infinity. It is characterized approaching an input frequency of 0 Hz but tested at a much higher frequency to minimize test time.

§ If BFSX does not meet this specification, the first bit of the serial data is driven on BDX until BFSX goes low (sampled on falling edge of BCLKX). After falling edge of the BFSX, data will be shifted out on the BDX pin.

NOTES: 3. Internal clock with external BFSX and vice versa are also allowable. However, BFSX timings to BCLKX always are defined depending on the source of BFSX, and BCLKX timings always are dependent upon the source of BCLKX.

4. Timings for BCLKX and BFSX are given with polarity bits (BCLKP and BFSP) set to 0.

buffered serial port transmit timing of external frames (continued)

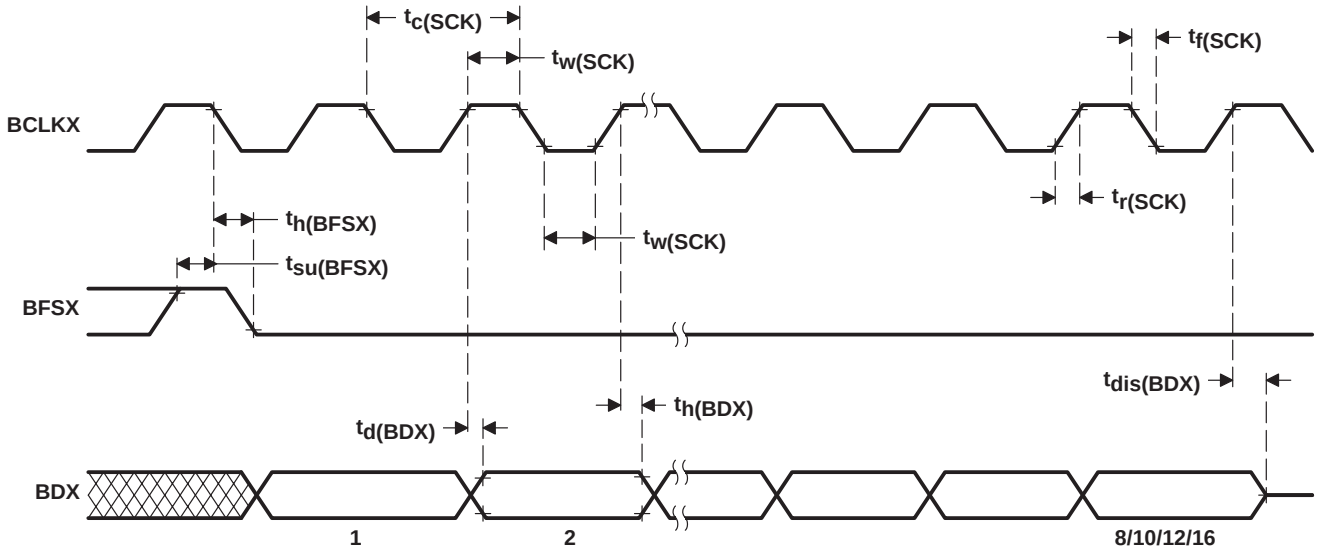


Figure 36. Buffered Serial Port Transmit Timing of External Clocks and External Frames

ADVANCE INFORMATION

buffered serial port transmit timing of internal frame and internal clock

switching characteristics over recommended operating conditions [H = 0.5tc(CO)] (see Figure 37)

PARAMETER		’C54x-40 ’LC54x-40 ’LC54x-50 ’54x-66		’LC54x-80 ’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{c(SCK)}$	Cycle time, serial port clock, internal clock	2H	62H†	2H	62H†	ns
$t_{d(BFSX)}$	Delay time, BFSX after BCLKX rising edge (see Notes 3 and 4)		10		10	ns
$t_{d(BDX)}$	Delay time, BDX valid after BCLKX rising edge		8		8	ns
$t_{dis(BDX)}$	Disable time, BDX after BCLKX rising edge†	0	5	0	5	ns
$t_{dis(BDX)pcm}$	Disable time, PCM mode, BDX after BCLKX rising edge†		5		5	ns
$t_{en(BDX)pcm}$	Enable time, PCM mode, BDX after BCLKX rising edge†	7		7		ns
$t_{h(BDX)}$	Hold time, BDX valid after BCLKX rising edge	0		0		ns
$t_{f(SCK)}$	Fall time, serial port clock†		4		4	ns
$t_{r(SCK)}$	Rise time, serial port clock†		4		4	ns
$t_{w(SCK)}$	Pulse duration, serial port clock low/high†	H–4		H–4		ns

† Values assured by design but not tested.

NOTES: 3. Internal clock with external BFSX and vice versa are also allowable. However, BFSX timings to BCLKX always are defined depending on the source of BFSX, and BCLKX timings always are dependent upon the source of BCLKX.
4. Timings for BCLKX and BFSX are given with polarity bits (BCLKP and BFSP) set to 0.

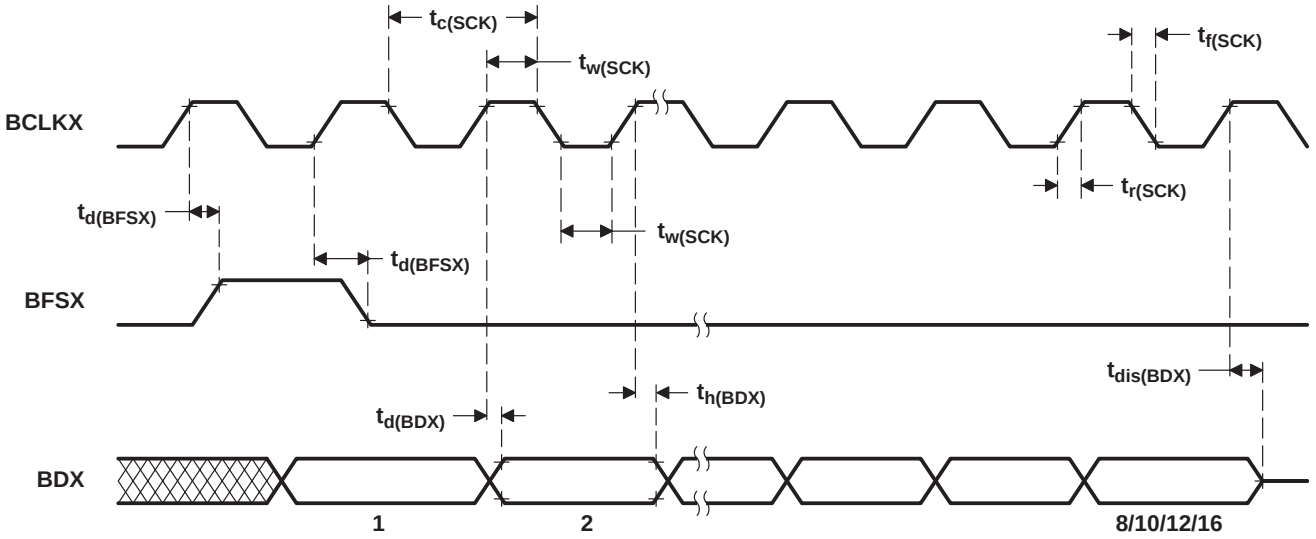


Figure 37. Buffered Serial Port Transmit Timing of Internal Clocks and Internal Frames

serial-port receive timing in TDM mode

timing requirements over recommended ranges of supply voltage and operating free-air temperature [H = 0.5$t_{c(CO)}$] ('542/'543 only) (see Figure 38)

		'542 '543		UNIT
		MIN	MAX†	
$t_{c(SCK)}$	Cycle time, serial-port clock	8H	‡	ns
$t_{f(SCK)}$	Fall time, serial-port clock		6	ns
$t_{r(SCK)}$	Rise time, serial-port clock		6	ns
$t_{w(SCK)}$	Pulse duration, serial-port clock low/high	4H		ns
$t_{su(TD-TCL)}$	Setup time, TDAT/TADD before TCLK falling edge	–(3H–9)		ns
$t_{h(TCH-TD)}$	Hold time, TDAT/TADD after TCLK rising edge, $t_{w(SCKL)}$ < 5H	0		ns
$t_{h(TCL-TD)}$	Hold time, TDAT/TADD after TCLK falling edge, $t_{w(SCKL)}$ > 5H	5H+5		ns
$t_{su(TF-TCH)}$	Setup time, TFRM before TCLK rising edge§	10		ns
$t_{h(TCH-TF)}$	Hold time, TFRM after TCLK rising edge§	10		ns

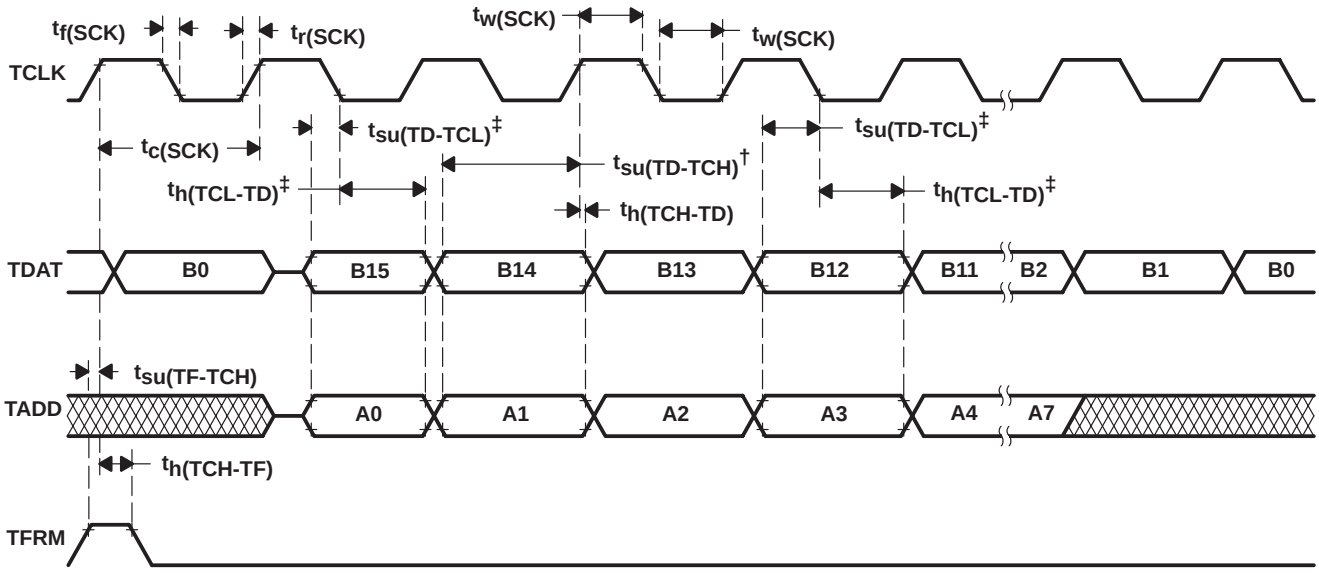
† Values assured by design and are not tested.
‡ The serial-port design is fully static and, therefore, can operate with $t_{c(SCK)}$ approaching infinity. It is characterized approaching an input frequency of 0 Hz but tested at a much higher frequency to minimize test time.
§ TFRM timing and waveforms shown in Figure 38 are for external TFRM. TFRM can also be configured as internal. The TFRM internal case is illustrated in the transmit timing diagram in Figure 39.

timing requirements over recommended ranges of supply voltage and operating free-air temperature [H = 0.5$t_{c(CO)}$] (all other 'C54x/'LC54x/'VC54x devices) (see Figure 38)

		'C54x-40 'LC54x-40 'LC54x-50		'54x-66		'LC54x-80 'VC54x-100		UNIT
		MIN	MAX†	MIN	MAX†	MIN	MAX†	
$t_{c(SCK)}$	Cycle time, serial-port clock	8H	‡	8H	‡	16H	‡	ns
$t_{f(SCK)}$	Fall time, serial-port clock		6		6		6	ns
$t_{r(SCK)}$	Rise time, serial-port clock		6		6		6	ns
$t_{w(SCK)}$	Pulse duration, serial-port clock low/high	4H		4H		8H		ns
$t_{su(TD-TCH)}$	Setup time, TDAT/TADD before TCLK rising edge	25		10		10		ns
$t_{h(TCH-TD)}$	Hold time, TDAT/TADD after TCLK rising edge	– 6		1		1		ns
$t_{su(TF-TCH)}$	Setup time, TFRM before TCLK rising edge§	10		10		10		ns
$t_{h(TCH-TF)}$	Hold time, TFRM after TCLK rising edge§	10		10		10		ns

† Values assured by design and are not tested.
‡ The serial-port design is fully static and, therefore, can operate with $t_{c(SCK)}$ approaching infinity. It is characterized approaching an input frequency of 0 Hz but tested at a much higher frequency to minimize test time.
§ TFRM timing and waveforms shown in Figure 38 are for external TFRM. TFRM can also be configured as internal. The TFRM internal case is illustrated in the transmit timing diagram in Figure 39.

serial-port receive timing in TDM mode (continued)



† All devices except ’542/’543
‡ ’542/’543 only

Figure 38. Serial-Port Receive Timing in TDM Mode

ADVANCE INFORMATION

serial-port transmit timing in TDM mode

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Figure 39)

PARAMETER		'542 '543		'C54x-40 'LC54x-40 'LC54x-50		UNIT
		MIN	MAX	MIN	MAX	
$t_{h(TCH-TDV)}$	Hold time, TDAT / TADD valid after TCLK rising edge, TCLK external	3†		0		ns
$t_{h(TCH-TDV)}$	Hold time, TDAT/TADD valid after TCLK rising edge, TCLK internal	1†		– 5		ns
$t_{d(TCH-TFV)}$	Delay time, TFRM valid after TCLK rising edge TCLK ext‡	H – 3	3H + 22§	H – 3	3H + 22§	ns
	Delay time, TFRM valid after TCLK rising edge, TCLK int‡	H – 3	3H + 12§	H – 3	3H + 12§	
$t_{d(TC-TDV)}$	Delay time, TCLK to valid TDAT/TADD rising edge, TCLK ext		18		25	ns
	Delay time, TCLK to valid TDAT/TADD rising edge, TCLK int		18		18	

PARAMETER		'54x-66		'LC54x-80 'VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{h(TCH-TDV)}$	Hold time, TDAT / TADD valid after TCLK rising edge, TCLK external	1		1		ns
$t_{h(TCH-TDV)}$	Hold time, TDAT/TADD valid after TCLK rising edge, TCLK internal	1		1		ns
$t_{d(TCH-TFV)}$	Delay time, TFRM valid after TCLK rising edge, TCLK ext‡	H – 3	3H+22§	H – 3	3H+22§	ns
	Delay time, TFRM valid after TCLK rising edge, TCLK int‡	H – 3	3H+12§	H – 3	3H+12§	
$t_{d(TC-TDV)}$	Delay time, TCLK to valid TDAT/TADD rising edge, TCLK ext		25		25	ns
	Delay time, TCLK to valid TDAT/TADD rising edge, TCLK int		18		18	

† Values derived from characterization data but not tested.
‡ TFRM timing and waveforms shown in Figure 39 are for internal TFRM. TFRM can also be configured as external. The TFRM external case is illustrated in the receive timing diagram in Figure 38.
§ Values assured by design but are not tested.

serial-port transmit timing in TDM mode (continued)

timing requirements over recommended ranges of supply voltage and operating free-air temperature [H = 0.5$t_{c(CO)}$] (see Figure 39)

		’C54x-40 ’LC54x-40 ’LC54x-50 ’54x-66		’LC54x-80 ’VC54x-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{c(SCK)}$	Cycle time, serial-port clock	8H‡	§	16H‡	§	ns
$t_{f(SCK)}$	Fall time, serial-port clock		6†		6†	ns
$t_{r(SCK)}$	Rise time, serial-port clock		6†		6†	ns
$t_{w(SCK)}$	Pulse duration, serial-port clock low/high	4H‡		8H‡		ns

† Values assured by design but are not tested.
‡ When SCK is generated internally, this value is typical.
§ The serial-port design is fully static and, therefore, can operate with $t_{c(SCK)}$ approaching ∞. It is characterized approaching an input frequency of 0 Hz but tested as a much higher frequency to minimize test time.

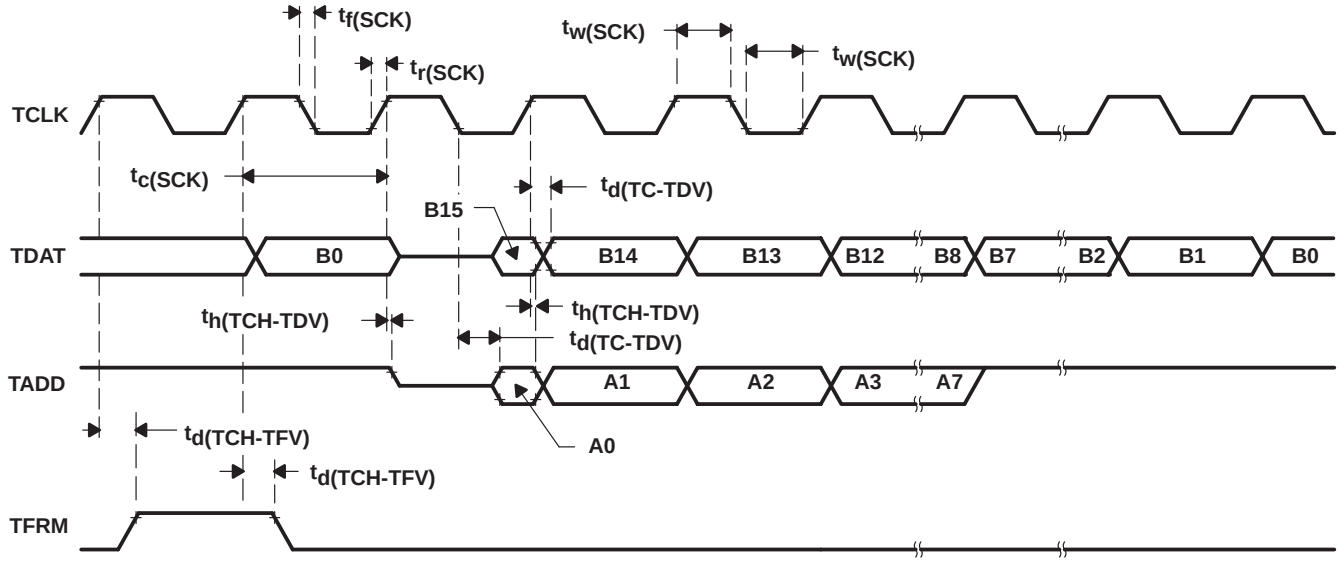


Figure 39. Serial-Port Transmit Timing in TDM Mode

host port interface timing

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Notes 5 and 6) (see Figure 40 through Figure 43)

PARAMETER			’C54x-40 ’C54x-50 ’C54x-66		UNIT
			MIN	MAX	
$t_{d(DSL-HDV)}$	Delay time, $\overline{DS}$ low to HD driven		5‡	12‡	ns
$t_{d(HEL-HDV1)}$	Delay time, HDS falling to HD valid for first byte of a non-subsequent read: → max 20 ns§¶	Case 1: Shared-access mode if $t_{w(DSH)}$ < 7H		7H+20–$t_{w(DSH)}$	ns
		Case 2: Shared-access mode if $t_{w(DSH)}$ > 7H		20‡	
		Case 3: Host-only mode if $t_{w(DSH)}$ < 20 ns		40–$t_{w(DSH)}$	
		Case 4: Host-only mode if $t_{w(DSH)}$ > 20 ns		20‡	
$t_{d(DSL-HDV2)}$	Delay time, $\overline{DS}$ low to HD valid, second byte‡		5¶	20	ns
$t_{d(DSH-HYH)}$	Delay time, $\overline{DS}$ high to HRDY high			10H+10‡	ns
$t_{su(HDV-HYH)}$	Setup time, HD valid before HRDY rising edge		3H–10‡		ns
$t_{h(DSH-HDV)R}$	Hold time, HD valid after $\overline{DS}$ rising edge, read		0†	12	ns
$t_{d(COH-HYH)}$	Delay time, CLKOUT rising edge to HRDY high			10‡	ns
$t_{d(DSH-HYL)}$	Delay time, $\overline{HDS}$ or $\overline{HCS}$ high to HRDY low			12‡	ns
$t_{d(COH-HTX)}$	Delay time, CLKOUT rising edge to $\overline{HINT}$ change†			15	ns

† Values derived from characterization data and not tested.
‡ Values assured by design but not tested.
§ Host-only mode timings apply for read accesses to HPIC or HPIA, write accesses to BOB, and resetting DSPINT or HINT to 0 in shared-access mode. HRDY does not go low for these accesses.
¶ Shared-access mode timings will be met automatically if HRDY is used.

NOTES: 5. SAM = shared-access mode, HOM = host-only mode
$\overline{HAD}$ stands for HCNTRL0, HCNTRL1, and HR/$\overline{W}$.
$\overline{HDS}$ refers to either $\overline{HDS1}$ or $\overline{HDS2}$.
$\overline{DS}$ refers to the logical OR of $\overline{HCS}$ and $\overline{HDS}$.
6. On host read accesses to the HPI, the setup time of HD before $\overline{DS}$ rising edge depends on the host waveforms and cannot be specified here.

ADVANCE INFORMATION

host port interface timing (continued)

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Notes 5 and 6) (see Figure 40 through Figure 43) (continued)

PARAMETER			’LC54x-80 ’VC54x-100		UNIT
			MIN	MAX	
$t_{d(DSL-HDV)}$	Delay time, $\overline{DS}$ low to HD driven		5‡	12‡	ns
$t_{d(HEL-HDV1)}$	Delay time, HDS falling to HD valid for first byte of a non-subsequent read: → max 20 ns§¶	Case 1: Shared-access mode if $t_{w(DSH)}$ < 7H		7H+20–$t_{w(DSH)}$	ns
		Case 2: Shared-access mode if $t_{w(DSH)}$ > 7H		20‡	
		Case 3: Host-only mode if $t_{w(DSH)}$ < 20 ns		40–$t_{w(DSH)}$	
		Case 4: Host-only mode if $t_{w(DSH)}$ > 20 ns		20‡	
$t_{d(DSL-HDV2)}$	Delay time, $\overline{DS}$ low to HD valid, second byte‡		5¶	20	ns
$t_{d(DSH-HYH)}$	Delay time, $\overline{DS}$ high to HRDY high			10H+10‡	ns
$t_{su(HDV-HYH)}$	Setup time, HD valid before HRDY rising edge		3H–10‡		ns
$t_{h(DSH-HDV)R}$	Hold time, HD valid after $\overline{DS}$ rising edge, read		0†	12	ns
$t_{d(COH-HYH)}$	Delay time, CLKOUT rising edge to HRDY high			10‡	ns
$t_{d(DSH-HYL)}$	Delay time, $\overline{HDS}$ or $\overline{HCS}$ high to HRDY low			12‡	ns
$t_{d(COH-HTX)}$	Delay time, CLKOUT rising edge to $\overline{HINT}$ change†			15	ns

† Values derived from characterization data and not tested.
‡ Values assured by design but not tested.
§ Host-only mode timings apply for read accesses to HPIC or HPIA, write accesses to BOB, and resetting DSPINT or HINT to 0 in shared-access mode. HRDY does not go low for these accesses.
¶ Shared-access mode timings will be met automatically if HRDY is used.

NOTES: 5. SAM = shared-access mode, HOM = host-only mode
$\overline{HAD}$ stands for HCNTRL0, HCNTRL1, and HR/$\overline{W}$.
$\overline{HDS}$ refers to either $\overline{HDS1}$ or $\overline{HDS2}$.
$\overline{DS}$ refers to the logical OR of $\overline{HCS}$ and $\overline{HDS}$.
6. On host read accesses to the HPI, the setup time of HD before $\overline{DS}$ rising edge depends on the host waveforms and cannot be specified here.

host port interface timing (continued)

timing requirements over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Note 5) (see Figure 40 through Figure 43)

			’C54x-40 ’C54x-50 ’C54x-66		’LC54x-80 ’VC54x-100		UNIT
			MIN	MAX	MIN	MAX	
$t_{su(HBV-DSL)}$	Setup time, HAD/HBIL valid before $\overline{DS}$ falling edge		10		10		ns
$t_{h(DSL-HBV)}$	Hold time, HAD/HBIL valid after $\overline{DS}$ falling edge		5		5		ns
$t_{su(HSL-DSL)}$	Setup time, $\overline{HAS}$ low before $\overline{DS}$ falling edge		12		12		ns
$t_{w(DSL)}$	Pulse duration, $\overline{DS}$ low		30†		30†		ns
$t_{w(DSH)}$	Pulse duration, $\overline{DS}$ high		10		10		ns
$t_{c(DSH-DSH)}$†	Cycle time, $\overline{DS}$ rising edge to next $\overline{DS}$ rising edge	Case 1: When using HRDY (see Access Timing With HRDY)	50		50		ns
		Case 2a: SAM accesses and HOM active writes to DSPINT or HINT without using HRDY (see Access Timings Without HRDY)	10H‡		10H‡		
		Case 2b: When not using HRDY for other HOM accesses	50		50		
$t_{su(HDV-DSH)}$	Setup time, HD valid before $\overline{DS}$ rising edge		12		12		ns
$t_{h(DSH-HDV)W}$	Hold time, HD valid after $\overline{DS}$ rising edge, write		3		3		ns

† A host not using HRDY should meet the 10H requirement all the time unless a software handshake is used to change the access rate according to the HPI mode.

‡ Values assured by design but not tested.

NOTE 5: SAM = shared-access mode, HOM = host-only mode
$\underline{HAD}$ stands for HCNTRL0, HCNTRL1, and HR/$\overline{W}$.
$\underline{HDS}$ refers to either $\overline{HDS1}$ or $\overline{HDS2}$.
$\overline{DS}$ refers to the logical OR of $\overline{HCS}$ and $\overline{HDS}$.

host port interface timing (continued)

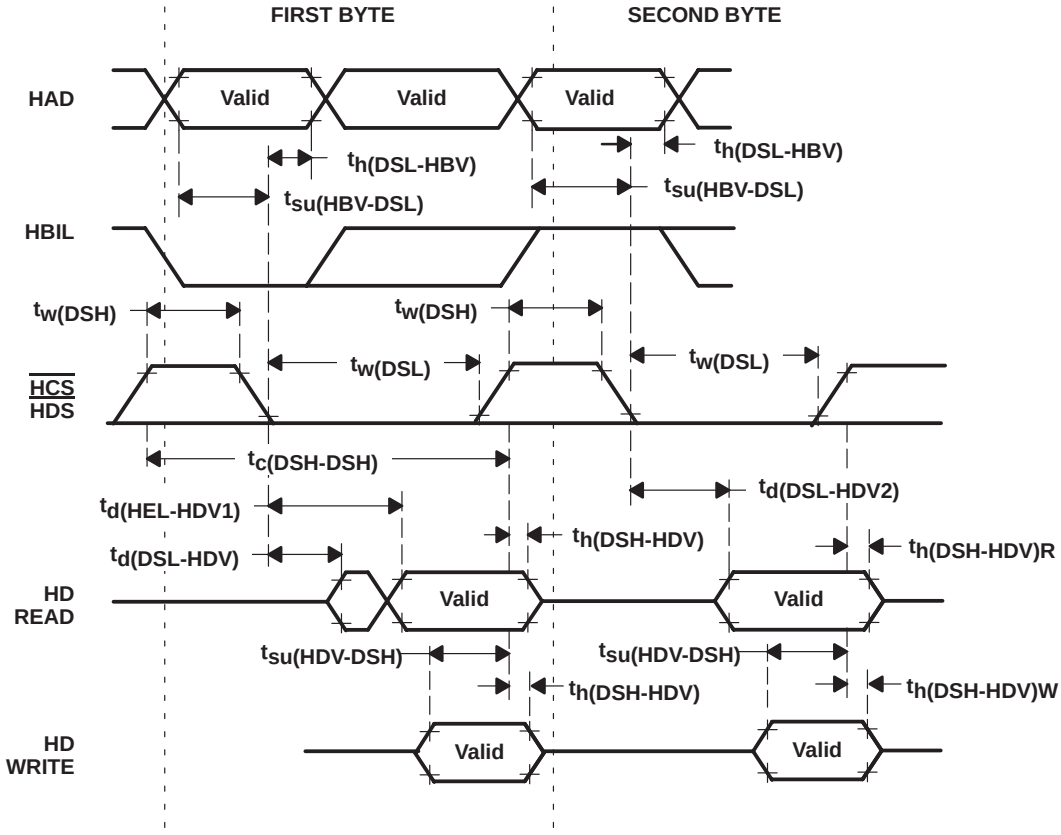


Figure 40. Read/Write Access Timings Without HRDY or $\overline{\text{HAS}}$

ADVANCE INFORMATION

host port interface timing (continued)

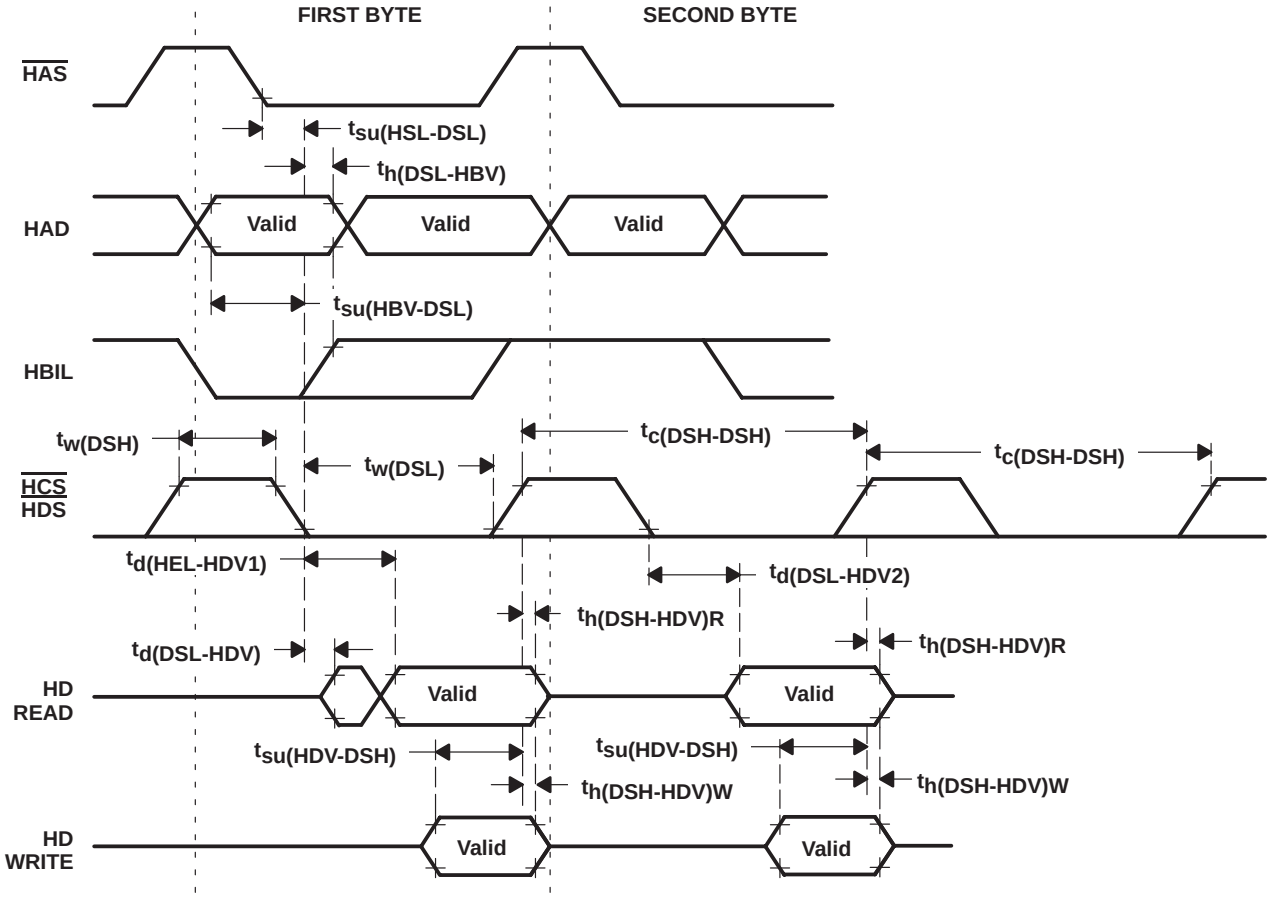
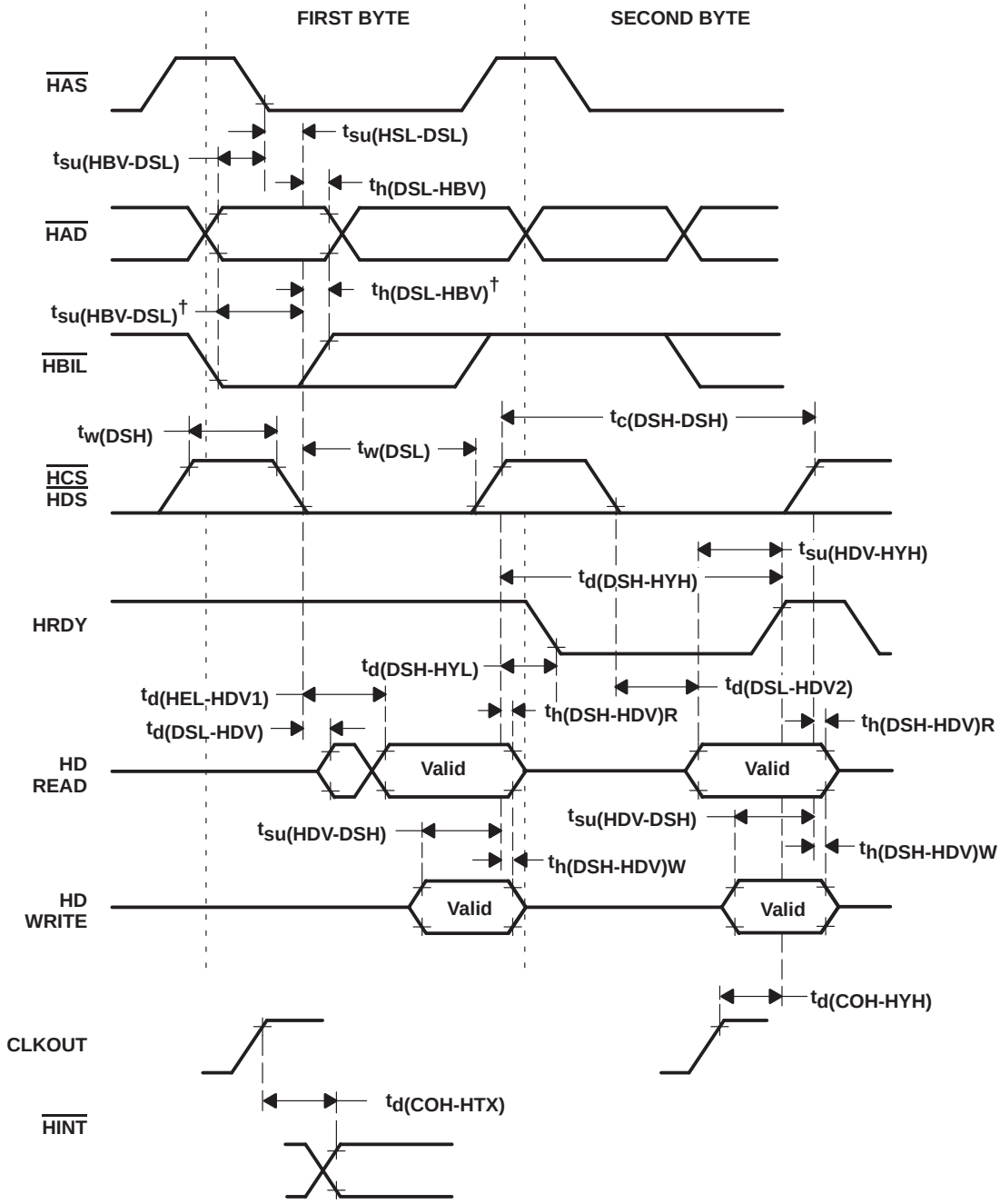


Figure 41. Read/Write Access Timings Using $\overline{\text{HAS}}$ Without HRDY

ADVANCE INFORMATION

host port interface timing (continued)



† When $\overline{\text{HAS}}$ is tied to V_{DD}

Figure 42. Read/Write Access Timing With HRDY

ADVANCE INFORMATION

host port interface timing (continued)

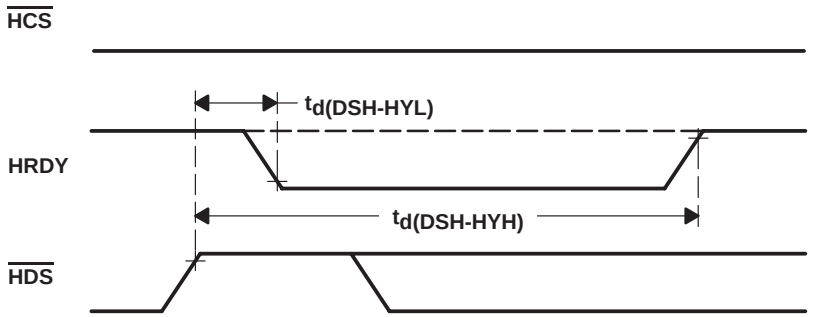
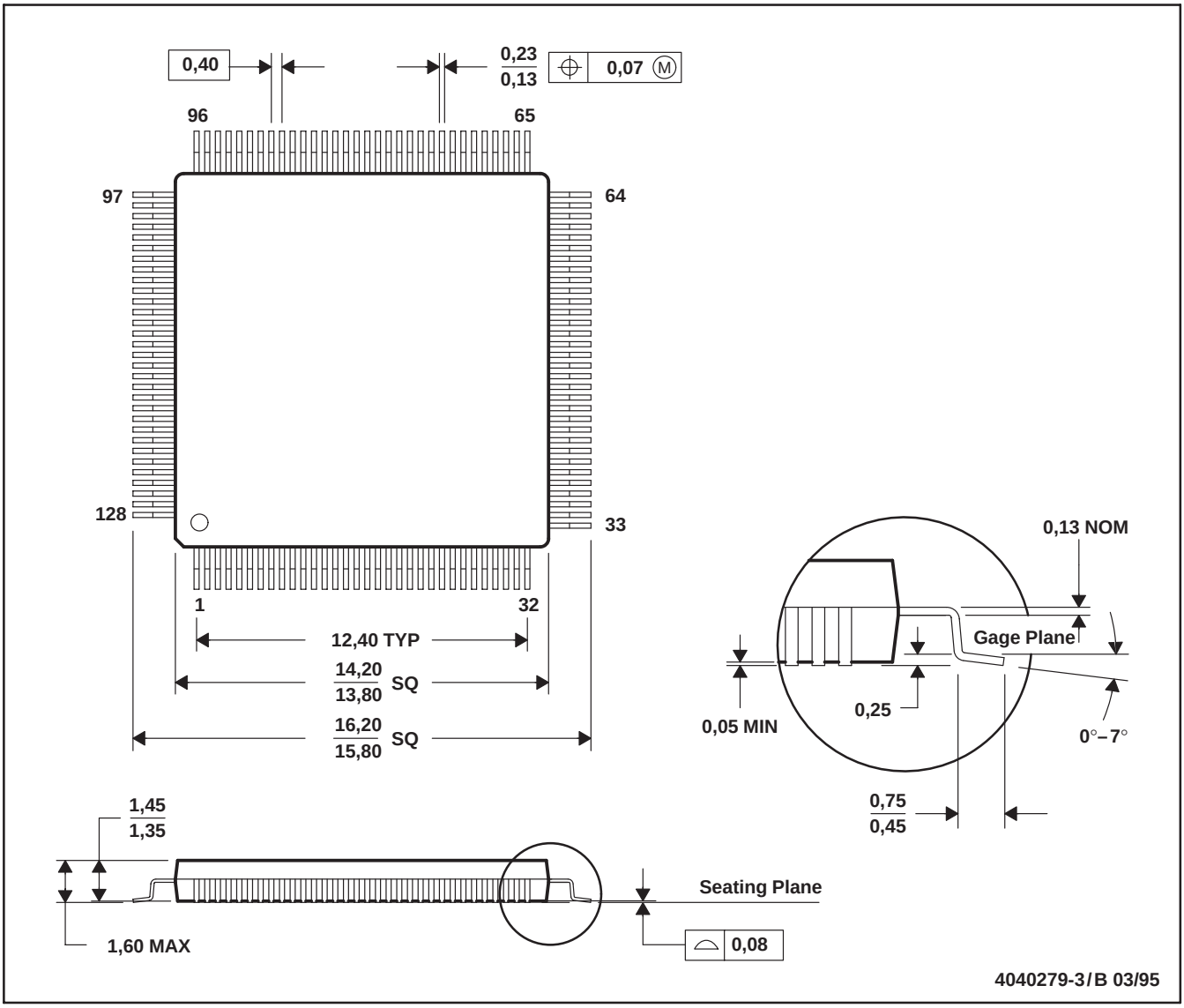


Figure 43. HRDY Signal When $\overline{\text{HCS}}$ is Always Low

MECHANICAL DATA

TMS320LC542/'LC545 128-Pin Thin Plastic Quad Flatpack (TQFP)

PBK (S-PQFP-G128) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

Thermal Resistance Characteristics

PARAMETER	°C/W
$R_{\Theta JA}$	58
$R_{\Theta JC}$	10

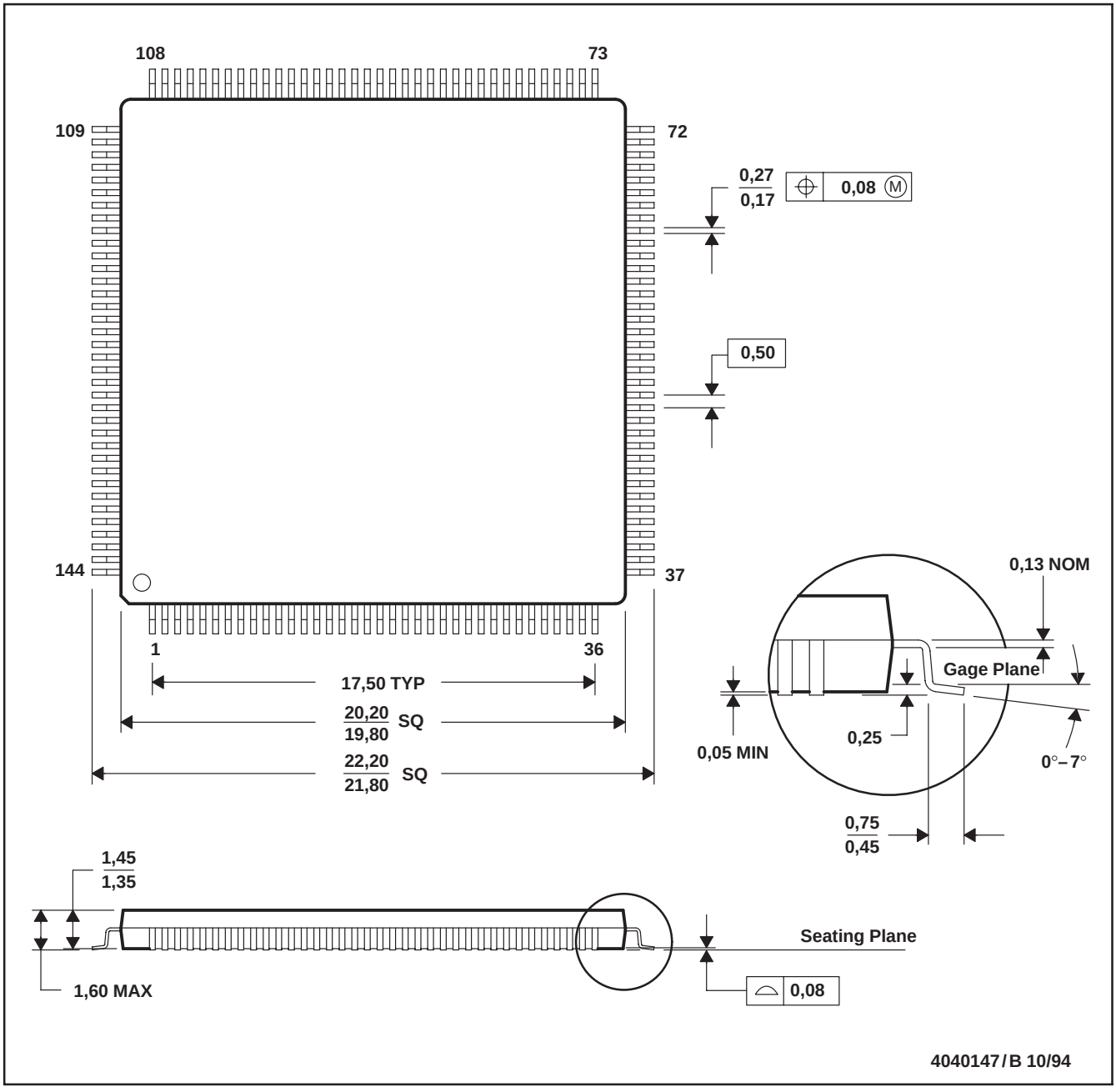
ADVANCE INFORMATION

ADVANCE INFORMATION

MECHANICAL DATA

TMS320C542/'LC542/'LC548, 'LC549, 'VC549 144-Pin Thin Plastic Quad Flatpack (TQFP)

PGE (S-PQFP-G144) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MO-136

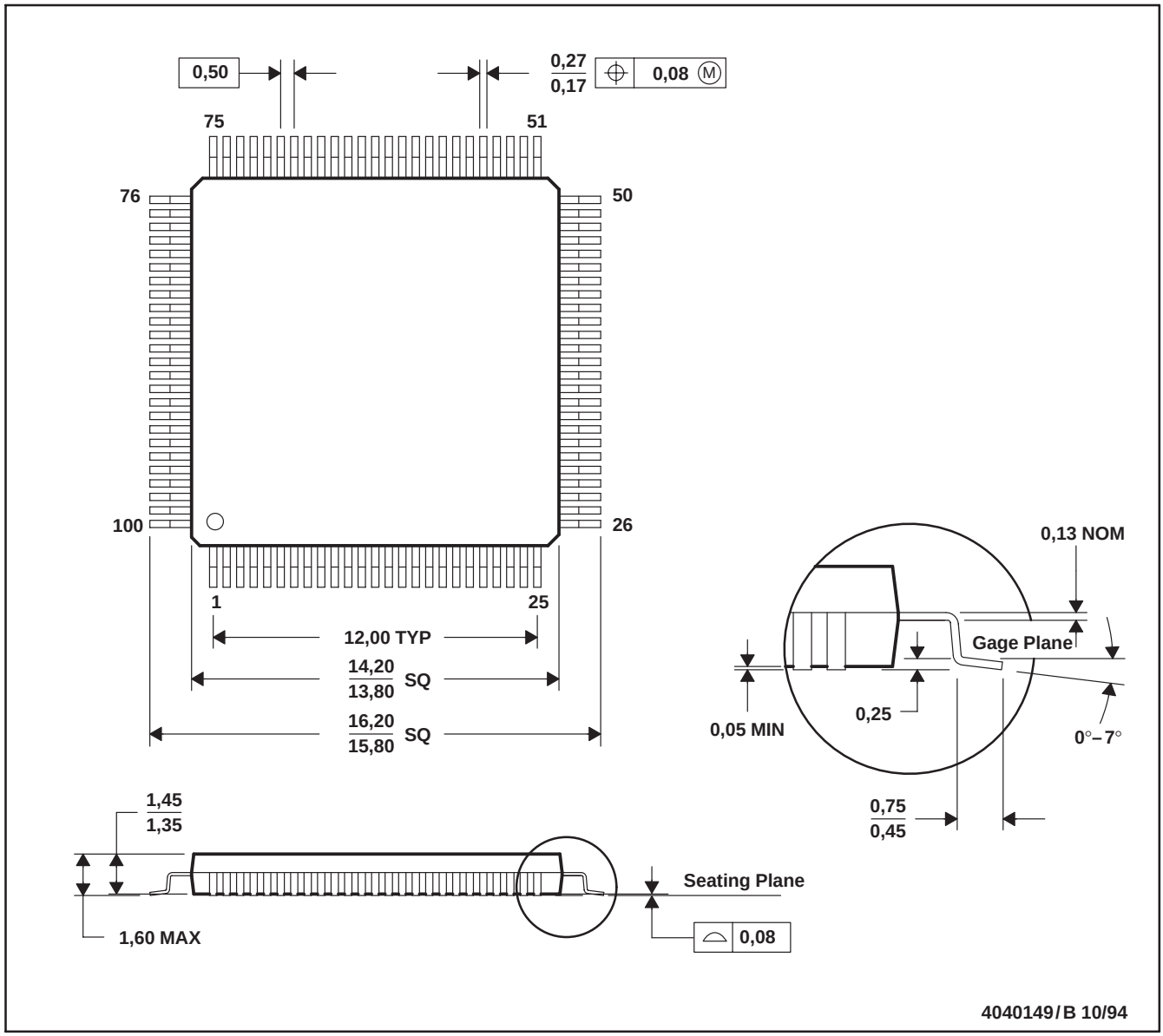
Thermal Resistance Characteristics

PARAMETER	°C/W
$R_{\Theta JA}$	56
$R_{\Theta JC}$	5

MECHANICAL DATA

TMS320C541/'LC541/'LC543/'LC546 100-Pin Thin Plastic Quad Flatpack (TQFP)

PZ (S-PQFP-G100) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MO-136

Thermal Resistance Characteristics

PARAMETER	°C/W
$R_{\Theta JA}$	58
$R_{\Theta JC}$	10

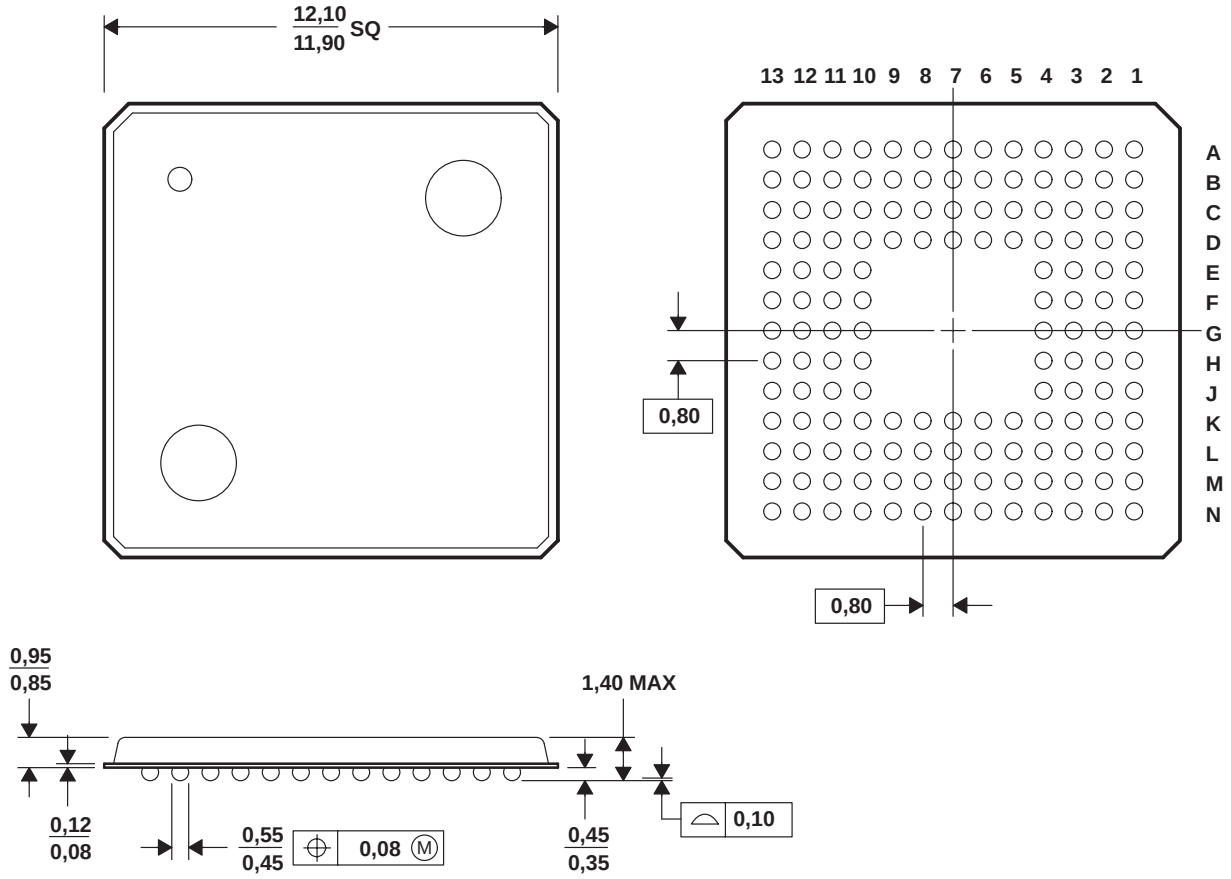
ADVANCE INFORMATION

ADVANCE INFORMATION

MECHANICAL DATA

TMS320LC548, TMS320LC549, and TMS320VC549 144-Pin Plastic Ball Grid Array Package (BGA)

GGU (S-PBGA-N144) **PLASTIC BALL GRID ARRAY PACKAGE**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

Thermal Resistance Characteristics

PARAMETER	°C/W
$R_{\Theta JA}$	38
$R_{\Theta JC}$	5

IMPORTANT NOTICE

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

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

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

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

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

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

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