- **200-MIPS Dual-Core DSP Consisting of Two Independent Subsystems**
- **Each Core Has an Advanced Multibus Architecture With Three Separate 16-Bit Data Memory Buses and One Program Bus**
- **40-Bit Arithmetic Logic Unit (ALU) Including a 40-Bit Barrel-Shifter and Two 40-Bit Accumulators Per Core**
- **Each Core Has a 17- × 17-Bit Parallel Multiplier Coupled to a 40-Bit Adder for Non-Pipelined Single-Cycle Multiply/ Accumulate (MAC) Operations**
- **Each Core Has a Compare, Select, and Store Unit (CSSU) for the Add/Compare Selection of the Viterbi Operator**
- **Each Core Has an Exponent Encoder to Compute an Exponent Value of a 40-Bit Accumulator Value in a Single Cycle**
- **Each Core Has Two Address Generators With Eight Auxiliary Registers and Two Auxiliary Register Arithmetic Units (ARAUs)**
- **16-Bit Data Bus With Data Bus Holder Feature**
- **256K × 16 Extended Program Address Space**
- **Total of 200K × 16 Dual- and Single-Access On-Chip RAM**
- **Single-Instruction Repeat and Block-Repeat Operations**
- **Instructions With 32-Bit Long Word Operands**
- **Instructions With 2 or 3 Operand Reads**
- **Fast Return From Interrupts**
- **Arithmetic Instructions With Parallel Store and Parallel Load**
- **Conditional Store Instructions**
- **Output Control of CLKOUT**
- **Output Control of TOUT**
- **Power Consumption Control With IDLE1, IDLE2, and IDLE3 Instructions**
- **Dual 1.8-V (Core) and 3.3-V (I/O) Power Supplies for Low Power, Fast Operation**
- **10-ns Single-Cycle Fixed-Point Instruction Execution**
- **Interprocessor Communication via Two Internal 8-Element FIFOs**
- **12 Channels of Direct Memory Access (DMA) for Data Transfers With No CPU Loading (6 Channels Per Subsystem)**
- **Six Multichannel Buffered Serial Ports (McBSPs) (Three McBSPs Per Subsystem)**
- **16-Bit Host-Port Interface (HPI16) Multiplexed With External Memory Interface Pins**
- **Software-Programmable Phase-Locked Loop (PLL) Provides Several Clocking Options (Requires External TTL Oscillator)**
- **On-Chip Scan-Based Emulation Logic**
- **Two Software-Programmable Timers (One Per Subsystem)**
- **Software-Programmable Wait-State Generator (14 Wait States Maximum)**
- **Provided in 144-pin BGA Ball Grid Array (GGU Suffix) and 144-pin Thin Quad Flatpack (TQFP) (PGE Suffix) packages**

NOTE: This data sheet is designed to be used in conjunction with the *TMS320C5000 DSP Family Functional Overview* (literature number SPRU307).

Table of Contents

description

The TMS320VC5420 fixed-point digital signal processor (DSP) is a dual CPU device capable of up to 200-MIPS performance. The ’5420 consists of two independent ’54x subsystems capable of core-to-core communications.

Each subsystem CPU is based on an advanced, modified Harvard architecture that has one program memory bus and three data memory buses. The processor also provides an arithmetic logic unit (ALU) that has a high degree of parallelism, application-specific hardware logic, on-chip memory, and additional on-chip peripherals.

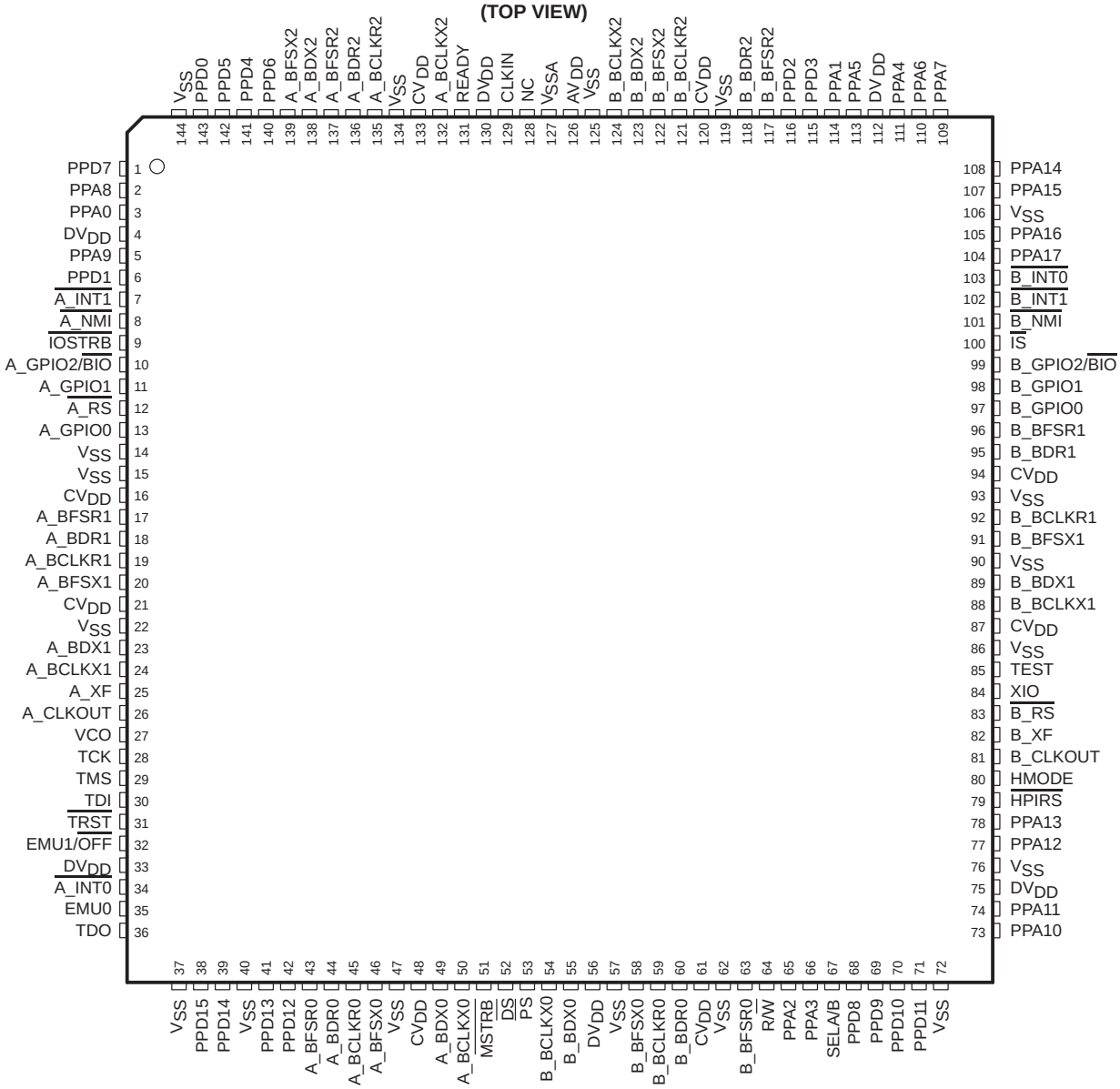
Each subsystem has separate program and data spaces, allowing simultaneous accesses to program instructions and data. Two read operations and one write operation can be performed in one cycle. Instructions with parallel store and application-specific instructions can fully utilize this architecture. Furthermore, data can be transferred between program and data spaces. Such parallelism supports a powerful set of arithmetic, logic, and bit manipulation operations that can be performed in a single machine cycle. In addition, the ’5420 includes the control mechanisms to manage interrupts, repeated operations, and function calls.

The ’5420 is offered in two temperature ranges and individual part numbers as shown below. (Please note that the industrial temperature device part numbers do not follow the typical numbering tradition.)

Commercial temperature devices (0°C to 85°C)
- TMS320VC5420PGE200 (144-pin LQFP)
- TMS320VC5420GGU200 (144-pin BGA)

Industrial temperature range devices (–40°C to 100°C)
- TMS320C5420PGEA200 (144-pin LQFP)
- TMS320C5420GGUA200 (144-pin BGA)

TMS320VC5420 PGE PACKAGE†‡§
(TOP VIEW)

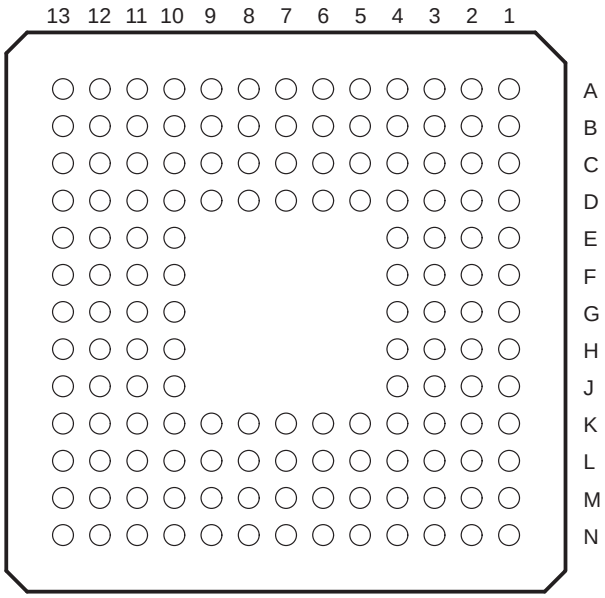


† NC = No internal connection

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

§ Pin configuration shown for nonmultiplexed mode only. See the Pin Assignments for the TMS320VC5420PGE table for multiplexed functions of specific pins.

TMS320VC5420 GGU PACKAGE
(BOTTOM VIEW)



The pin assignments table for the TMS320VC5420GGU lists each pin name and its associated pin number for this 144-pin ball grid array (BGA) package.

pin assignments

The ’5420 pin assignments tables list each pin name and corresponding pin number for the two package types. Some of the ’5420 pins can be configured for one of two functions. For these pins, the primary pin name is listed in the primary column. The secondary pin name is listed in the secondary column and is shaded grey.

Pin Assignments for the TMS320VC5420PGE (144-Pin Thin Quad Flatpack)

PRIMARY SIGNAL NAME	SECONDARY SIGNAL NAME	PIN NO.	PRIMARY SIGNAL NAME	SECONDARY SIGNAL NAME	PIN NO.
PPD7	HD7	1	PPA8		2
PPA0	$\overline{\text{A\_HINT}}$	3	DV_{DD}		4
PPA9		5	PPD1	HD1	6
$\overline{\text{A\_INT1}}$		7	$\overline{\text{A\_NMI}}$		8
$\overline{\text{IOSTRB}}$	A_GPIO3 A_TOUT	9	A_GPIO2/$\overline{\text{BIO}}$		10
A_GPIO1		11	$\overline{\text{A\_RS}}$		12
A_GPIO0		13	V_{SS}		14
V_{SS}		15	CV_{DD}		16
A_BFSR1		17	A_BDR1		18
A_BCLKR1		19	A_BFSX1		20
CV_{DD}		21	V_{SS}		22
A_BDX1		23	A_BCLKX1		24
A_XF		25	A_CLKOUT		26
VCO		27	TCK		28
TMS		29	TDI		30
$\overline{\text{TRST}}$		31	EMU1		32
DV_{DD}		33	$\overline{\text{A\_INT0}}$		34
EMU0		35	TDO		36
V_{SS}		37	PPD15	HD15	38
PPD14	HD14	39	V_{SS}		40
PPD13	HD13	41	PPD12	HD12	42
A_BFSR0		43	A_BDR0		44
A_BCLKR0		45	A_BFSX0		46
V_{SS}		47	CV_{DD}		48
A_BDX0		49	A_BCLKX0		50
$\overline{\text{MSTRB}}$	$\overline{\text{HCS}}$	51	$\overline{\text{DS}}$	$\overline{\text{HDS2}}$	52
$\overline{\text{PS}}$	$\overline{\text{HDS1}}$	53	B_BCLKX0		54
B_BDX0		55	DV_{DD}		56
V_{SS}		57	B_BFSX0		58
B_BCLKR0		59	B_BDR0		60
CV_{DD}		61	V_{SS}		62
B_BFSR0		63	R/$\overline{\text{W}}$	HR/$\overline{\text{W}}$	64
PPA2	HCNTL1	65	PPA3	HCNTL0	66
SELA/B		67	PPD8	HD8	68
PPD9	HD9	69	PPD10	HD10	70
PPD11	HD11	71	V_{SS}		72

Pin Assignments for the TMS320VC5420PGE (144-Pin Thin Quad Flatpack) (Continued)

PRIMARY SIGNAL NAME	SECONDARY SIGNAL NAME	PIN NO.	PRIMARY SIGNAL NAME	SECONDARY SIGNAL NAME	PIN NO.
PPA10		73	PPA11		74
DV_{DD}		75	V_{SS}		76
PPA12		77	PPA13		78
$\overline{\text{HPIRS}}$		79	HMODE		80
B_CLKOUT		81	B_XF		82
$\overline{\text{B\_RS}}$		83	XIO		84
TEST		85	V_{SS}		86
CV_{DD}		87	B_BCLKX1		88
B_BDX1		89	V_{SS}		90
B_BFSX1		91	B_BCLKR1		92
V_{SS}		93	CV_{DD}		94
B_BDR1		95	B_BFSR1		96
B_GPIO0		97	B_GPIO1		98
B_GPIO2/$\overline{\text{BIO}}$		99	$\overline{\text{IS}}$	B_GPIO3	100
$\overline{\text{B\_NMI}}$		101	$\overline{\text{B\_INT1}}$		102
$\overline{\text{B\_INT0}}$		103	PPA17		104
PPA16		105	V_{SS}		106
PPA15		107	PPA14		108
PPA7		109	PPA6		110
PPA4	$\overline{\text{HAS}}$	111	DV_{DD}		112
PPA5		113	PPA1	$\overline{\text{B\_HINT}}$	114
PPD3	HD3	115	PPD2	HD2	116
B_BFSR2		117	B_BDR2		118
V_{SS}		119	CV_{DD}		120
B_BCLKR2		121	B_BFSX2		122
B_BDX2		123	B_BCLKX2		124
V_{SS}		125	AV_{DD}		126
V_{SSA}		127	NC		128
CLKIN		129	DV_{DD}		130
READY	HRDY	131	A_BCLKX2		132
CV_{DD}		133	V_{SS}		134
A_BCLKR2		135	A_BDR2		136
A_BFSR2		137	A_BDX2		138
A_BFSX2		139	PPD6	HD6	140
PPD4	HD4	141	PPD5	HD5	142
PPD0	HD0	143	V_{SS}		144

Pin Assignments for the TMS320VC5420GGU (144-Pin MicroStar Ball Grid Array)

PRIMARY SIGNAL NAME	SECONDARY SIGNAL NAME	BALL NO.	PRIMARY SIGNAL NAME	SECONDARY SIGNAL NAME	BALL NO.
PPD7	HD7	A1	PPA8		B1
DV_{DD}		C1	$\overline{A\_NMI}$		D1
$\overline{A\_RS}$		E1	CV_{DD}		F1
A_BDR1		G1	CV_{DD}		H1
A_XF		J1	TMS		K1
EMU1/$\overline{OFF}$		L1	EMU0		M1
V_{SS}		N1	PPD0	HD0	A2
V_{SS}		B2	PPA0	$\overline{A\_HINT}$	C2
$\overline{A\_INT1}$		D2	A_GPIO1		E2
V_{SS}		F2	A_BFSR1		G2
V_{SS}		H2	A_CLKOUT		J2
TDI		K2	DV_{DD}		L2
TDO		M2	PPD15	HD15	N2
PDD6	HD6	A3	PPD4	HD4	B3
PPD5	HD5	C3	PPD1	HD1	D3
A_GPIO2/$\overline{BIO}$		E3	V_{SS}		F3
A_BCLKR1		G3	A_BDX1		H3
VCO		J3	$\overline{TRST}$		K3
$\overline{A\_INT0}$		L3	PPD14	HD14	M3
V_{SS}		N3	A_BFSR2		A4
A_BDX2		B4	A_BFSX2		C4
PPA9		D4	$\overline{IOSTRB}$	A_GPIO3 A_TOUT	E4
A_GPIO0		F4	A_BFSX1		G4
A_BCLKX1		H4	TCK		J4
PPD13	HD13	K4	PPD12	HD12	L4
A_BFSR0		M4	A_BDR0		N4
CV_{DD}		A5	V_{SS}		B5
A_BCLKR2		C5	A_BDR2		D5
A_BCLKR0		K5	A_BFSX0		L5
V_{SS}		M5	CV_{DD}		N5
CLKIN		A6	DV_{DD}		B6
READY	HRDY	C6	A_BCLKX2		D6
A_BDX0		K6	A_BCLKX0		L6
$\overline{MSTRB}$	$\overline{HCS}$	M6	$\overline{DS}$	$\overline{HDS2}$	N6
AV_{DD}		A7	V_{SS}		B7
V_{SSA}		C7	NC		D7
DV_{DD}		K7	B_BDX0		L7

Pin Assignments for the TMS320VC5420GGU (144-Pin MicroStar Ball Grid Array) (Continued)

PRIMARY SIGNAL NAME	SECONDARY SIGNAL NAME	BALL NO.	PRIMARY SIGNAL NAME	SECONDARY SIGNAL NAME	BALL NO.
$\overline{PS}$	$\overline{HDS1}$	M7	B_BCLKX0		N7
B_BCLKX2		A8	B_BDX2		B8
B_BFSX2		C8	B_BCLKR2		D8
B_BDR0		K8	B_BCLKR0		L8
B_BFSX0		M8	V_{SS}		N8
CV_{DD}		A9	V_{SS}		B9
B_BDR2		C9	B_BFSR2		D9
R/$\overline{W}$	HR/$\overline{W}$	K9	B_BFSR0		L9
V_{SS}		M9	CV_{DD}		N9
PPD2	HD2	A10	PPD3	HD3	B10
PPA1	$\overline{B\_HINT}$	C10	PPA5		D10
$\overline{IS}$	B_GPIO3 B_TOUT	E10	B_BFSR1		F10
B_BCLKR1		G10	TEST		H10
B_CLKOUT		J10	PPA12		K10
SELA/B		L10	PPA3	HCNTL0	M10
PPA2	HCNTL1	N10	DV_{DD}		A11
PPA4	$\overline{HAS}$	B11	V_{SS}		C11
$\overline{B\_INT0}$		D11	B_GPIO2/$\overline{BIO}$		E11
B_BDR1		F11	B_BFSX1		G11
V_{SS}		H11	B_XF		J11
PPA13		K11	PPD10	HD10	L11
PPD9	HD9	M11	PPD8	HD8	N11
PPA6		A12	PPA14		B12
PPA16		C12	$\overline{B\_INT1}$		D12
B_GPIO1		E12	CV_{DD}		F12
B_BDX1		G12	CV_{DD}		H12
$\overline{B\_RS}$		J12	$\overline{HPIRS}$		K12
DV_{DD}		L12	V_{SS}		M12
PPD11	HD11	N12	PPA7		A13
PPA15		B13	PPA17		C13
$\overline{B\_NMI}$		D13	B_GPIO0		E13
V_{SS}		F13	V_{SS}		G13
B_BCLKX1		H13	XIO		J13
HMODE		K13	V_{SS}		L13
PPA11		M13	PPA10		N13

signal descriptions

The ’5420 signal descriptions table lists each pin name, function, and operating mode(s) for the ’5420 device. Some of the ’5420 pins can be configured for one of two functions; a primary function and a secondary function. The names of these pins in secondary mode are shaded in grey in the following table.

Signal Descriptions

NAME	TYPE†	DESCRIPTION
DATA SIGNALS		
PPA17 (MSB) PPA16 PPA15 PPA14 PPA13 PPA12 PPA11 PPA10 PPA9 PPA8 PPA7 PPA6 PPA5 PPA4‡§ PPA3 PPA2 PPA1 PPA0 (LSB)	I/O/Z	Parallel port address bus. The DSP can access the external memory locations by way of the external memory interface using PPA[17:0] in external memory interface (EMIF) mode when the XIO pin is logic high. The PPA[17:0] pins are also multiplexed with the HPI interface. In HPI mode (XIO pin is low), the external address pins PPA[17:0] are used by a host processor for access to the memory map by way of the on-chip HPI. Refer to the HPI section of this table for details on the secondary functions of these pins. These pins are placed into the high-impedance state when $\overline{\text{OFF}}$ is low.
PPD15 (MSB) PPD14 PPD13 PPD12 PPD11 PPD10 PPD9 PPD8 PPD7 PPD6 PPD5 PPD4 PPD3 PPD2 PPD1 PPD0 (LSB)	I/O/Z¶	Parallel port data bus. The DSP uses this bidirectional data bus to access external memory when the device is in external memory interface (EMIF) mode (the XIO pin is logic high). This data bus is also multiplexed with the 16-bit HPI data bus. When in HPI mode, the bus is used to transfer data between the host processor and internal DSP memory via the HPI. Refer to the HPI section of this table for details on the secondary functions of these pins. The data bus includes bus holders to reduce power dissipation caused by floating, unused pins. The bus holders also eliminate the need for external pullup resistors on unused pins. When the data bus is not being driven by the ’5420, the bus holders keep data pins at the last driven logic level. The data bus keepers are disabled at reset and can be enabled/disabled via the BH bit of the BSCR register. These pins are placed into high-impedance state when $\overline{\text{OFF}}$ is low.
INITIALIZATION, INTERRUPT, AND RESET OPERATIONS		
$\overline{\text{A\_INT0}}$§ $\overline{\text{B\_INT0}}$§ $\overline{\text{A\_INT1}}$§ $\overline{\text{B\_INT1}}$§	I	External user interrupts. $\overline{\text{INT0}}$–$\overline{\text{INT3}}$ are prioritized and are maskable by the interrupt mask register (IMR) and the interrupt mode bit. $\overline{\text{INT0}}$ –$\overline{\text{INT3}}$ can be polled and reset by way of the interrupt flag register (IFR).
$\overline{\text{A\_NMI}}$§ $\overline{\text{B\_NMI}}$§	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.

† I = Input, O = Output, S = Supply, Z = High Impedance
‡ This pin has an internal pullup resistor.
§ These pins have Schmitt trigger inputs.
¶ This pin has an internal bus holder controlled by way of the BSCR register in subchip A.
# This pin is used by Texas Instruments for device testing and should be left unconnected.
|| This pin has an internal pulldown resistor.

Signal Descriptions (Continued)

NAME	TYPE†	DESCRIPTION
INITIALIZATION, INTERRUPT, AND RESET OPERATIONS (CONTINUED)		
$\overline{\text{A\_RS}}$§ $\overline{\text{B\_RS}}$§	I	Reset. $\overline{\text{RS}}$ causes the digital signal processor (DSP) to terminate execution and causes a reinitialization of the CPU and peripherals. When $\overline{\text{RS}}$ is brought to a high level, execution begins at location 0FF80h of program memory. $\overline{\text{RS}}$ affects various registers and status bits.
XIO	I	The XIO pin is used to configure the parallel port as a host-port interface (HPI mode when XIO pin is low), or as an asynchronous memory interface (EMIF mode when XIO pin is high). At device reset, the logic combination of the XIO, HMODE, and SELA/B pin levels determines the initialization value of the MP/MC bit (a bit in the processor mode status (PMST) register) Refer to the memory section for details.
GENERAL-PURPOSE I/O SIGNALS		
A_XF B_XF	O	External flag output (latched software-programmable output-only signal). Bit addressable. A_XF and B_XF are placed into the high-impedance state when $\overline{\text{OFF}}$ is low.
A_GPIO0 B_GPIO0 A_GPIO1 B_GPIO1	I/O	General-purpose I/O pins (software-programmable I/O signal). Values can be latched (output) by writing into the GPIO register. The states of GPIO pins (inputs) can be read by reading the GPIO register. The GPIO direction is also programmable by way of the DIRn field in the GPIO register.
A_GPIO2/$\overline{\text{BIO}}$ B_GPIO2/$\overline{\text{BIO}}$		General-purpose I/O. These pins can be configured in the same manner as GPIO0–1; however in input mode, the pins also operate as the traditional branch control bit ($\overline{\text{BIO}}$). If application code does not perform $\overline{\text{BIO}}$-conditional instructions, these pins operate as general inputs.
A_GPIO3 *(A_TOUT)* B_GPIO3 *(B_TOUT)*	I/O	**PRIMARY** $\overline{\text{IOSTRB}}$ / $\overline{\text{IS}}$ — O When the device is in HPI mode and HMODE = 0 (multiplexed), these pins are controlled by the general-purpose I/O control register. TOUT bit must be set to “1” to drive the timer output on the pin. IF TOUT = 0, then these pins are general-purpose I/Os. In EMIF mode (XIO pin high), these signals serve their primary functions and are active during external I/O space accesses.
MEMORY CONTROL SIGNALS		
$\overline{\text{PS}}$‡	O	Program space select signal. The $\overline{\text{PS}}$ signal is asserted during external program space accesses. This pin is placed into the high-impedance state when $\overline{\text{OFF}}$ is low. This pin is also multiplexed with the HPI, and functions as the $\overline{\text{HDS1}}$ data strobe input signal in HPI mode. Refer to the HPI section of this table for details on the secondary function of this pin.
$\overline{\text{DS}}$‡	O	Data space select signal. The $\overline{\text{DS}}$ signal is asserted during external data space accesses. This pin is placed into the high-impedance state when $\overline{\text{OFF}}$ is low. This pin is also multiplexed with the HPI, and functions as the $\overline{\text{HDS2}}$ data strobe input signal in HPI mode. Refer to the HPI section of this table for details on the secondary function of this pin.
$\overline{\text{IS}}$	O	I/O space select signal. The $\overline{\text{IS}}$ signal is asserted during external I/O space accesses. This pin is placed into the high-impedance state when $\overline{\text{OFF}}$ is low. This pin is also multiplexed with the general purpose I/O feature, and functions as the B_GPIO3 (B_TOUT) input/output signal in HPI mode. Refer to the General Purpose I/O section of this table for details on the secondary function of this pin.
$\overline{\text{MSTRB}}$‡§	O	Program and data memory strobe (active in EMIF mode). This pin is placed into the high-impedance state when $\overline{\text{OFF}}$ is low.

† I = Input, O = Output, S = Supply, Z = High Impedance
‡ This pin has an internal pullup resistor.
§ These pins have Schmitt trigger inputs.
¶ This pin has an internal bus holder controlled by way of the BSCR register in subchip A.
# This pin is used by Texas Instruments for device testing and should be left unconnected.
‖ This pin has an internal pulldown resistor.

Signal Descriptions (Continued)

NAME	TYPE†	DESCRIPTION
MEMORY CONTROL SIGNALS (CONTINUED)		
READY	I	Data-ready input signal. READY indicates that the external device is prepared for a bus transaction to be completed. If the device is not ready (READY = 0), the processor waits one cycle and checks READY again. The processor performs the READY detection if at least two software wait states are programmed. This pin is also multiplexed with the HPI, and functions as the Host-port data ready (output) in HPI mode. Refer to the HPI section of this table for details on the secondary function of this pin.
R/$\overline{W}$	O	Read/write output signal. R/$\overline{W}$ indicates transfer direction during communication to an external device. R/$\overline{W}$ is normally in the read mode (high), unless it is asserted low when the DSP performs a write operation. This pin is also multiplexed with the HPI, and functions as the Host-port Read/write input in HPI mode. Refer to the HPI section of this table for details on the secondary function of this pin. This pin is placed into the high-impedance state when $\overline{OFF}$ is low.
$\overline{IOSTRB}$	O	I/O space memory strobe. External I/O space is accessible by the CPU and not the direct memory access (DMA) controller. The DMA has its own dedicated I/O space that is not accessible by the CPU. This pin is also multiplexed with the general pupose I/O feature, and functions as the A_GPIO3(A_TOUT) signal in HPI mode. Refer to the general purpose I/O section of this table for details on the secondary function of this pin. This pin is placed into the high-impedance state when $\overline{OFF}$ is low.
SELA/B	I	The SELA/B pin designates which DSP subsystem has access to the parallel-port interface. Furthermore, this pin determines which subsystem is accessible by the host via the HPI. For external memory accesses (XIO pin high), when SELA/B is low subsystem A has control of the external memory interface. Similarly, when SELA/B is high subsystem B has control. See Table 7 for a truth table of SELA/B, HMODE and XIO pins and functionality. At device reset, the logic combination of the XIO, HMODE, and SELA/B pin levels determines the initialization value of the MP/MC bit (a bit in the processor mode status (PMST) register) Refer to the memory section for details.
CLOCKING SIGNALS		
A_CLKOUT B_CLKOUT	O	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. The CLKOUT pin can be turned off by writing a “1” to the CLKOFF bit of the PMST register. CLKOUT goes into the high-impedance state when EMU1/$\overline{OFF}$ is low.
CLKIN§	I	Input clock to the device. CLKIN connects to an oscillator circuit/device.
VCO	O	VCO is the output of the voltage-controlled oscillator stage of the PLL. This is a 3-state output during normal operation. Active in silicon test/debug mode.
MULTICHANNEL BUFFERED SERIAL PORT 0, 1, AND 2 SIGNALS		
A_BCLKR0‡§ B_BCLKR0‡§ A_BCLKR1‡§ B_BCLKR1‡§ A_BCLKR2‡§ B_BCLKR2‡§	I/O/Z	Receive clocks. BCLKR serves as the serial shift clock for the buffered serial-port receiver. Input from an external clock source for clocking data into the McBSP. When not being used as a clock, these pins can be used as general-purpose I/O by setting RIOEN = 1. BCLKR can be configured as an output by the way of the CLKRM bit in the PCR register.

† I = Input, O = Output, S = Supply, Z = High Impedance
‡ This pin has an internal pullup resistor.
§ These pins have Schmitt trigger inputs.
¶ This pin has an internal bus holder controlled by way of the BSCR register in subchip A.
# This pin is used by Texas Instruments for device testing and should be left unconnected.
|| This pin has an internal pulldown resistor.

Signal Descriptions (Continued)

NAME	TYPE†	DESCRIPTION
MULTICHANNEL BUFFERED SERIAL PORT 0, 1, AND 2 SIGNALS (CONTINUED)		
A_BCLKX0‡§ B_BCLKX0‡§ A_BCLKX1‡§ B_BCLKX1‡§ A_BCLKX2‡§ B_BCLKX2‡§	I/O/Z	Transmit clocks. Clock signal used to clock data from the transmit register. This pin can also be configured as an input by setting the CLKXM = 0 in the PCR register. BCLKX can be sampled as an input by way of the IN1 bit in the SPC register. When not being used as a clock, these pins can be used as general-purpose I/O by setting XIOEN = 1. These pins are placed into the high-impedance state when $\overline{OFF}$ is low.
A_BDR0 B_BDR0 A_BDR1 B_BDR1 A_BDR2 B_BDR2	I	Buffered serial data receive (input) pin. When not being used as data-receive pins, these pins can be used as general-purpose I/O by setting RIOEN = 1.
A_BDX0 B_BDX0 A_BDX1 B_BDX1 A_BDX2 B_BDX2	O/Z	Buffered serial-port transmit (output) pin. When not being used as data-transmit pins, these pins can be used as general-purpose I/O by setting XIOEN = 1. These pins are placed into the high-impedance state when $\overline{OFF}$ is low.
A_BFSR0 B_BFSR0 A_BFSR1 B_BFSR1 A_BFSR2 B_BFSR2	I/O/Z	Frame synchronization pin for buffered serial-port input data. The BFSR pulse initiates the receive-data process over BDR pin. When not being used as data-receive synchronization pins, these pins can be used as general-purpose I/O by setting RIOEN = 1.
A_BFSX0 B_BFSX0 A_BFSX1 B_BFSX1 A_BFSX2 B_BFSX2	I/O/Z	Buffered serial-port frame synchronization pin for transmitting data. The BFSX pulse initiates the transmit-data process over BDX pin. If $\overline{RS}$ is asserted when BFSX is configured as output, then BFSX is turned into input mode by the reset operation. When not being used as data-transmit synchronization pins, these pins can be used as general-purpose I/O by setting XIOEN = 1. These pins are placed into the high-impedance state when $\overline{OFF}$ is low.
HOST-PORT INTERFACE SIGNALS		
HA[0:17]	I	**PRIMARY**: PPA[0:17] O — HPI address inputs. HA[0:17] are used by the host device, in the HPI non-multiplexed mode (HMODE pin is high), to address the on-chip RAM of the ’5420. These pins are shared with the external memory interface and are only used by the HPI when the interface is in HPI mode (XIO pin is low).
HD[0:15]	I/O/Z	PPD[0:15] I/O/Z — Parallel bidirectional data bus. HD[0:15] are used by the host device to transfer data to and from the on-chip RAM of the ’5420. These pins are shared with the external memory interface and are only used by the HPI when the interface is in HPI mode (XIO pin is low). The data bus includes bus holders to reduce power dissipation caused by floating, unused pins. The bus holders also eliminate the need for external pullup resistors on unused pins. When the data bus is not being driven by the ’5420, the bus holders keep data pins at the last driven logic level. The data bus keepers are disabled at reset and can be enabled/disabled via the BH bit of the BSCR register. These pins are placed into the high-impedance state when $\overline{OFF}$ is low.

† I = Input, O = Output, S = Supply, Z = High Impedance
‡ This pin has an internal pullup resistor.
§ These pins have Schmitt trigger inputs.
¶ This pin has an internal bus holder controlled by way of the BSCR register in subchip A.
# This pin is used by Texas Instruments for device testing and should be left unconnected.
‖ This pin has an internal pulldown resistor.

Signal Descriptions (Continued)

NAME	TYPE†	DESCRIPTION		
HOST-PORT INTERFACE SIGNALS (CONTINUED)				
HCNTL0 HCNTL1	I	**PRIMARY** PPA3 PPA2	O	HPI control inputs. The HCNTL0 and HCNTL1 values between HPIA, and HPID registers during HPI reads and writes. These signals are only used in HPI multiplexed address/data mode (HMODE pin is low). These pins are shared with the external memory interface and are only used by the HPI when the interface is in HPI mode (XIO pin is low).
$\overline{\text{HAS}}$‡§	I	PPA4‡§	O	Address strobe input. Hosts with multiplexed address and data pins require $\overline{\text{HAS}}$ to latch the address in the HPIA register. This signal is only used in HPI multiplexed address/data mode (HMODE pin is low). This pin is shared with the external memory interface and is only used by the HPI when the interface is in HPI mode (XIO pin is low).
$\overline{\text{HCS}}$‡§	I	$\overline{\text{MSTRB}}$‡§	O	HPI chip-select signal. Thissignal must be active during HPI transfers, and can remain active between concurrent transfers. This pin is shared with the external memory interface and is only used by the HPI when the interface is in HPI mode (XIO pin is low).
$\overline{\text{HDS1}}$‡§ $\overline{\text{HDS2}}$‡§	I	$\overline{\text{PS}}$‡§ $\overline{\text{DS}}$‡§	O	HPI data strobes. $\overline{\text{HDS1}}$ and $\overline{\text{HDS2}}$ are driven by the host read and write strobes to control transfer HPI transfers. These pins are shared with the external memory interface and are only used by the HPI when the interface is in HPI mode (XIO pin is low).
HR/$\overline{\text{W}}$	I	R/$\overline{\text{W}}$	O	HPI read/write signal. This signal is used by the host to control the direction of an HPI transfer. This pin is shared with the external memory interface and is only used by the HPI when the interface is in HPI mode (XIO pin is low).
HRDY	O	READY	I	HPI data-ready output. The ready output informs the host when the HPI is ready for the next transfer. This pin is shared with the external memory interface and is only used by the HPI when the interface is in HPI mode (XIO pin is low). HRDY is placed into the high-impedance state when $\overline{\text{OFF}}$ is low.
$\overline{\text{A\_HINT}}$ $\overline{\text{B\_HINT}}$	O	PPA0 PPA1	O	Host interrupt pin. HPI can interrupt the host by asserting this low. The host can clear this interrupt by writing a “1” to the $\overline{\text{HINT}}$ bit of the HPIC register. Only supported in HPI multiplexed address/data mode (HMODE pin is low). These pins are placed into the high-impedance state when $\overline{\text{OFF}}$ is low.
$\overline{\text{HPIRS}}$§	I	Host-port interface (HPI) reset pin. This signal resets the host port interface and both subsystems.		
HMODE	I	Host mode select. When this pin is low it selects the HPI multiplexed address/data mode. The multiplexed address/data mode allows hosts with multiplexed address/data lines access to the HPI registers HPIC, HPIA, and HPID. Host-to-DSP and DSP-to-host interrupts are supported in this mode. When HMODE is high, it selects the HPI nonmultiplexed mode. HPI nonmultiplexed mode allows hosts with separate address/data buses to access the HPI address range by way of the 18-bit address bus and the HPI data (HPID) register via the 16-bit data bus. Host-to-DSP and DSP-to-host interrupts are not supported in this mode.		
SUPPLY PINS				
AV_{DD}	S	Dedicated power supply that powers the PLL. AV_{DD} = 1.8 V. AV_{DD} can be connected to CV_{DD}.		
CV_{DD}	S	Dedicated power supply that powers the core CPUs. CV_{DD} = 1.8 V		
DV_{DD}	S	Dedicated power supply that powers the I/O pins. DV_{DD} = 3.3 V		
V_{SS}	S	Digital ground. Dedicated ground plane for the device.		

† I = Input, O = Output, S = Supply, Z = High Impedance
‡ This pin has an internal pullup resistor.
§ These pins have Schmitt trigger inputs.
¶ This pin has an internal bus holder controlled by way of the BSCR register in subchip A.
# This pin is used by Texas Instruments for device testing and should be left unconnected.
‖ This pin has an internal pulldown resistor.

Signal Descriptions (Continued)

NAME	TYPE†	DESCRIPTION
SUPPLY PINS (CONTINUED)		
V_{SSA}	S	Analog ground. Dedicated ground for the PLL. V_{SSA} can be connected to V_{SS} if digital and analog grounds are not separated.
TEST PIN		
TEST#		No connection
EMULATION/TEST PINS		
TCK‡§	I	Standard test clock. This is normally a free-running clock signal with a 50% duty cycle. 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	Test data input. Pin with an internal pullup device. TDI is clocked into the selected register (instruction or data) on a rising edge of TCK.
TDO	O	Test data pin. The contents of the selected register is shifted out of TDO on the falling edge of TCK. TDO is in high-impedance state except when the scanning of data is in progress. These pins are placed into high-impedance state when $\overline{OFF}$ is low.
TMS‡	I	Test mode select. Pin with internal pullup device. This serial control input is clocked into the TAP controller on the rising edge of TCK.
$\overline{TRST}$‖	I	Test reset. When high, $\overline{TRST}$ gives the scan system control of the operations of the device. If $\overline{TRST}$ is driven low, the device operates in its functional mode and the emulation signals are ignored. Pin with internal pulldown device.
EMU0	I/O/Z	Emulator interrupt 0 pin. When $\overline{TRST}$ is driven low, EMU0 must be high for the activation of the EMU1/$\overline{OFF}$ condition. When $\overline{TRST}$ is driven high, EMU0 is used as an interrupt to or from the emulator system and is defined as I/O.
EMU1/$\overline{OFF}$	I/O/Z	Emulator interrupt 1 pin. When $\overline{TRST}$ is driven high, EMU1/$\overline{OFF}$ is used as an interrupt to or from the emulator system and is defined as I/O. When $\overline{TRST}$ transitions from high to low, then EMU1 operates as $\overline{OFF}$. EMU/$\overline{OFF}$ = 0 puts all output drivers into the high-impedance state. Note that $\overline{OFF}$ is used exclusively for testing and emulation purposes (and not for multiprocessing applications). Therefore, for the $\overline{OFF}$ condition, the following conditions apply: $\overline{TRST}$ = 0, EMU0 = 1, EMU1 = 0

† I = Input, O = Output, S = Supply, Z = High Impedance
‡ This pin has an internal pullup resistor.
§ These pins have Schmitt trigger inputs.
¶ This pin has an internal bus holder controlled by way of the BSCR register in subchip A.
# This pin is used by Texas Instruments for device testing and should be left unconnected.
‖ This pin has an internal pulldown resistor.

functional overview

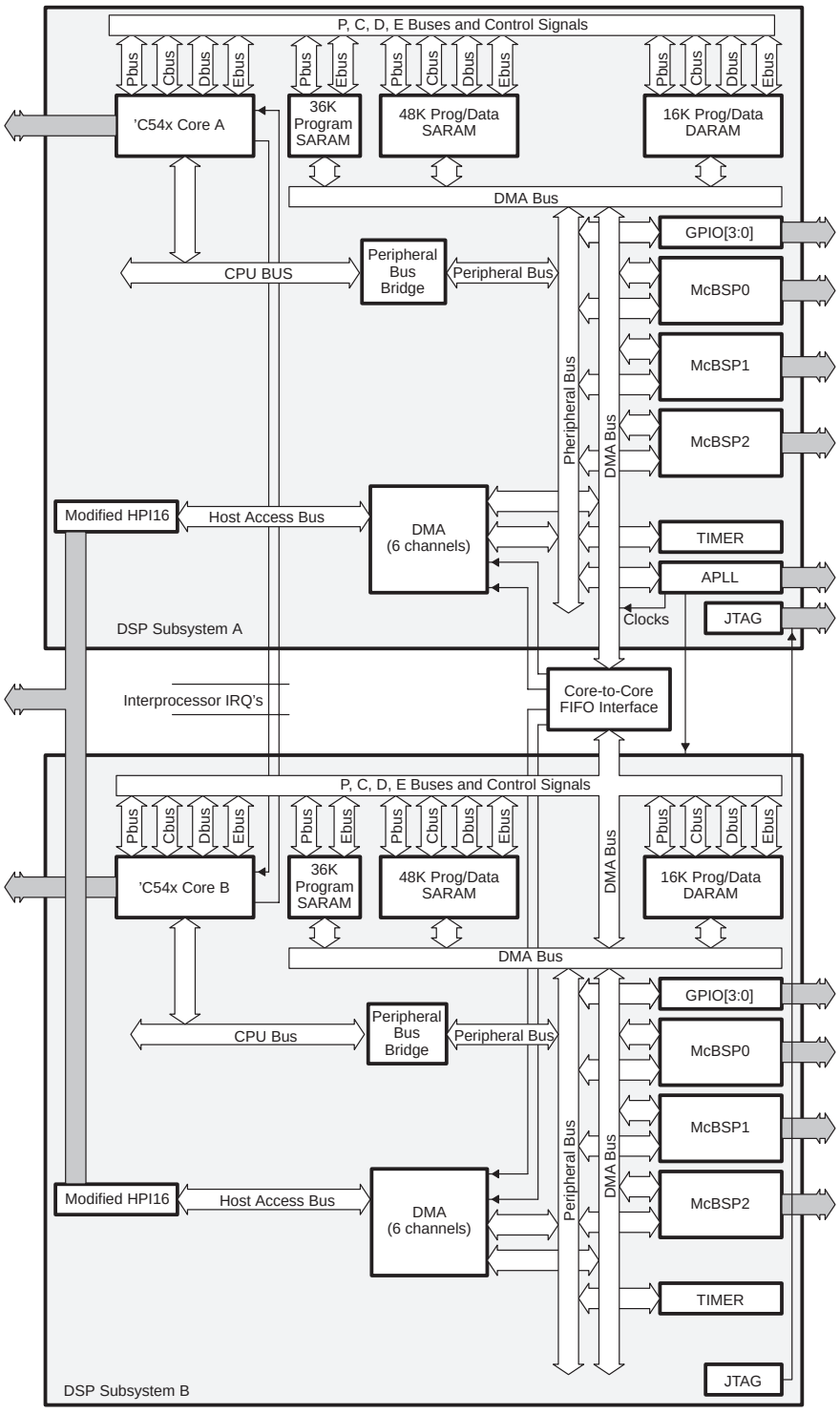


Figure 1. Functional Block Diagram

memory

The total memory address range for each ’5420 subsystem is 384K 16-bit words. The memory space is divided into three specific memory segments: 256K-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 is used to interface to external memory-mapped peripherals and can also serve as extra data storage space. The CPU I/O space should not be confused with the DMA I/O space, which is completely independent and not accessible by the CPU.

on-chip dual-access RAM (DARAM)

The ’5420 subsystems A and B each have 16K $\times$ 16-bit on-chip DARAM (2 blocks of 8K words).

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 ’5420 subsystems A and B each have 84K-word $\times$ 16-bit on-chip SARAM (ten blocks of 8K words each and one block of 4K words).

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 (4000h–7FFFh) is mapped into data memory space. This memory range can be mapped into program/data memory space by setting the OVLY bit in the PMST register. The SARAM at 8000h–FFFFh is program memory at reset and can be configured as program/data memory by setting the DROM bit. SARAM spaces18000h–1FFFFh and 2F000h–2FFFFh are mapped as program memory only.

program memory

The ’5420 device features a paged extended memory scheme in program space to allow access of up to 256K of program memory relative to each subsystem. This extended program memory (each subsystem) is organized into four pages (0–3), each 64K in length. A hardware pin is used to select which DSP subsystem (A or B) has control of the external memory interface. To implement the extended program memory scheme, the ’5420 device includes the following features:

- Two additional address lines (for a total of 18)
- A pin (SELA/B) for external memory interface arbitration between subsystem A and B

data memory

The data memory space on each ’5420 subsystem contains up to 64K 16-bit word addresses. The device automatically accesses 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.

parallel I/O ports

Each subsystem of the ’5420 has a total of 64K I/O ports. These ports can be addressed by PORTR and PORTW. The $\overline{\text{IS}}$ signal indicates the read/write access 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 logic. The SELA/B pin selects which subsystem has access to the external I/O space.

external memory interface

The ’5420 has a single external memory interface shared between both subsystems. The external memory interface enables the ’5420 subsystems to connect to external memory devices or other parallel interfaces. The SELA/B pin is used to determine which subsystem has access to the external memory interface. When the SELA/B pin is low, subsystem A has access to the external memory interface, and when it is high, subsystem

external memory interface (continued)

B has access to the interface. The external memory interface is also shared with the host port interface (HPI). The XIO pin is used to select between the external memory interface and the hostport interface. When the XIO pin is high, the external memory interface is active, and when it is low, the host port interface is active.

processor mode status register (PMST)

Each subsystem has a processor-mode status register (PMST) that controls memory configuration. The bit layout of the PMST register is shown in Figure 1

15 – 7	6	5	4	3	2	1	0
IPTR	MP/$\overline{\text{MC}}$	OVLY	AVIS	DROM	CLKOFF	SMUL	SST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LEGEND: R = Read, W = Write

Figure 1. Processor Mode Status Register (PMST) Bit Layout

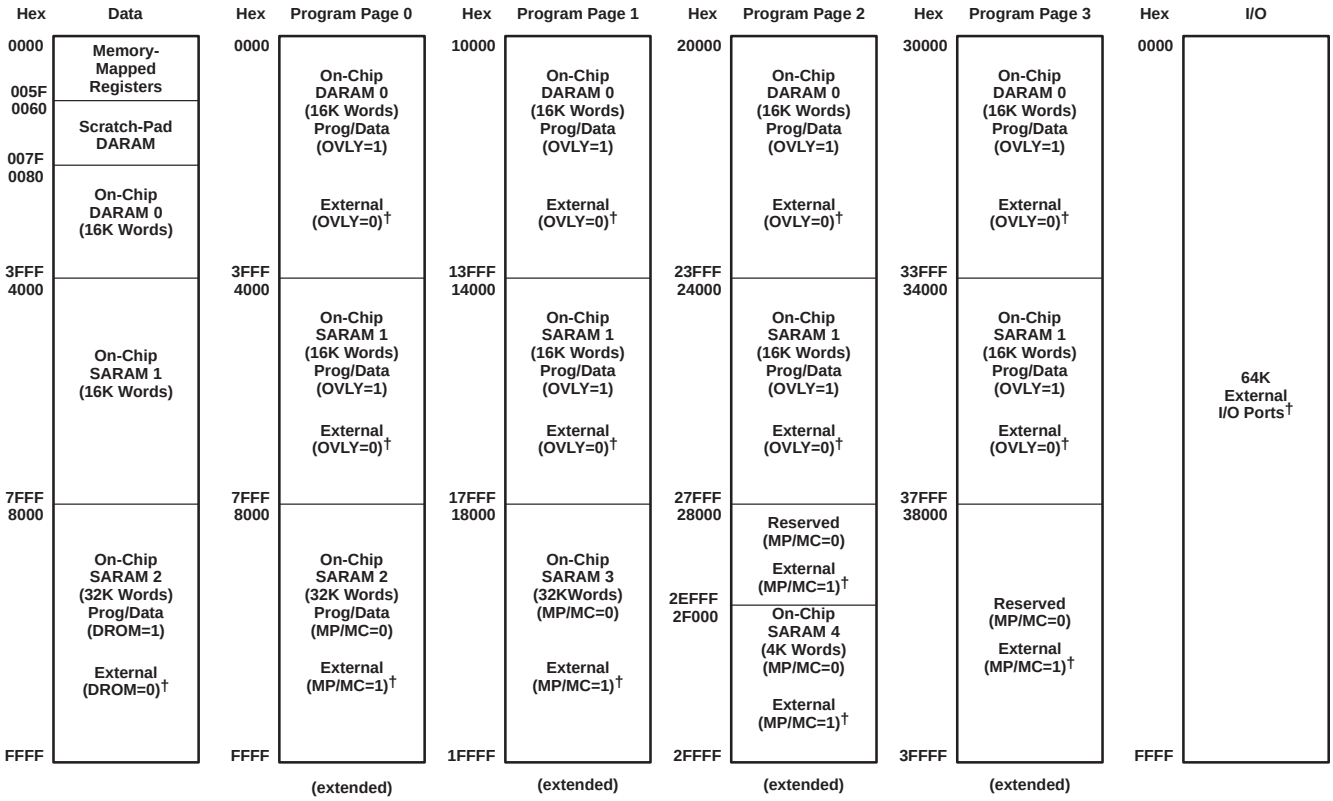
The functions of the PMST register bits are illustrated in the memory map. The MP/MC bit is used to map the upper address range of all program space pages (x8000–xFFFF) as either external or internal memory. The OVLY bit is used to overlay the on-chip DARAM0 and SARAM1 blocks from dataspace onto to program space. Similarly, the DROM bit is used to overlay the SARAM2 block from program space onto data space. See the *TMS320C54x DSP CPU and Peripherals Reference Set, Volume 1* (literature number SPRU131) for a description of the other bits of the PMST register.

Due to the dual-processor configuration and the several EMIF/HPI options available, the MP/MC bit is initialized at the time of device reset to a logic level that is dependent on the XIO, HMODE, and SELA/B pins. Table 1 shows the initialized logic level of the MP/MC bit and how it depends on these pins.

Table 1. MP/MC Bit Logic Levels at Reset

’5420 PINS			MP/MC BIT	
XIO	HMODE	SELA/B	SUBSYSTEM A	SUBSYSTEM B
0	X	X	0	0
1	0	X	1	1
1	1	0	1	0
1	1	1	0	1

memory map



† The external memory interface must be enabled by driving the XIO pin high, in order for external memory accesses to occur.

Figure 2. Memory Map for Each CPU Subsystem

multicore reset signals

The ’5420 device includes three reset signals: $\overline{\text{A\_RS}}$, $\overline{\text{B\_RS}}$, and $\overline{\text{HPIRS}}$. The $\overline{\text{A\_RS}}$ and $\overline{\text{B\_RS}}$ pins function as the CPU reset signal for subsystem A and subsystem B, respectively. These signals reset the state of the CPU registers and upon release, initiates the reset function. Additionally, the $\overline{\text{A\_RS}}$ signal resets the on-chip PLL and initializes the CLKMD register to bypass mode.

The HPI reset signal ($\overline{\text{HPIRS}}$) places the HPI peripheral into a reset state. It is necessary to wait three clock cycles after the rising edge of $\overline{\text{HPIRS}}$ before performing an HPI access.

reset vector initialization

The ’5420 device does not have on-chip ROM and therefore does not contain bootloader routines/software. Consequently, the user must have a valid reset vector in place before releasing the reset signal. This is referred to as *reset vector initialization*. After reset, the ’5420 device fetches the reset vector at address 0xFF80 in program memory and begins to execute the instructions found in memory. The application code is raw program and data words and does not require the traditional *boot-table* or *boot-packet* format.

reset vectorinitialization (continued)

The selection of the reset initialization option is determined by the state of three pins; XIO, XMODE, and SELA/B. The options include:

- HPI (host-dependent)
- EMIF-to-HPI (stand-alone)
- Simultaneous EMIF (stand-alone)
- Sequential EMIF (stand-alone)

HPI

The HPI method is only valid when the level of the XIO pin is low. The ’5420 acts as a slave to an external master host. The host device must keep the ’5420 device in reset as it downloads code to the subsystem that is determined by the logic level of the SELA/B pin. When SELA/B is low, the master downloads code to subsystem A. By driving SELA/B high, the master host can subsequently download code to subsystem B. The HMODE pin determines the configuration of the HPI (multiplexed or nonmultiplexed) and is an asynchronous input. Therefore, HMODE can be changed to the desired configuration while $\overline{\text{A\_RS}}$ and $\overline{\text{B\_RS}}$ are low prior to the transfer. Once the subsystem(s) have been loaded and are ready to execute, the master host can release the reset pin(s).

There are two valid options for controlling the reset function of the subsystems. The first option is to hold the $\overline{\text{A\_RS}}$ and $\overline{\text{B\_RS}}$ pins low while the $\overline{\text{HPIRS}}$ pin transitions from low to high. This keeps the cores in reset while allowing the HPI full access to download the application code. The host can now drive the $\overline{\text{A\_RS}}$ and $\overline{\text{B\_RS}}$ signals high simultaneously or separately to release the respective subsystem from reset. The subsystems then fetch their respective reset vector. If the subsystems are released from reset seperately, subsystem A should be released from reset first, since the $\overline{\text{A\_RS}}$ pin resets the on-chip PLL that is common to both subsystems.

Another valid option is to keep the $\overline{\text{A\_RS}}$ and $\overline{\text{B\_RS}}$ pins high while the host transitions the $\overline{\text{HPIRS}}$ pin from low to high. Special internal logic causes the HPI to be fully operable and the cores remain in reset. As a result, after the host processor has downloaded the application code via the HPI, it must perform an additional HPI write (any value) to address 0x2F. This releases the respective subsystem from reset. By changing the value of SELA/B, the host can write to 0x2F via the HPI to release the other subsystem from reset.

EMIF-to-HPI

In this particular vector initialization method, the host processor controlling the HPI is one of the subsystems. The master host is subsystem A if SELA/B is low and subsystem B when SELA/B is high. As described in the signal descriptions table, the address, data, and control signals of the program space are multiplexed with the HPI signals. In a special mode when XIO is high (EMIF mode) and HMODE is high (HPI nonmultiplexed mode), these multiplexed signals are connected, making it possible for the master subsystem’s EMIF to initialize the slave subsystem via the slave’s HPI. The master subsystem then releases the slave from reset either by transitioning the hardware reset signal ($\overline{\text{x\_RS}}$) high, or in software, by writing to memory location 0x2F via the HPI. As a result, the slave core fetches the reset vector.

simultaneous EMIF

The simultaneous EMIF vector initialization option allows both subsystems to access external memory simultaneously. The subsystems are designed to operate synchronized with one another while accessing the same locations simultaneously. In this mode, when XIO is high and HMODE is low, one subsystem is given full control of the EMIF while the other subsystem relies on the synchronization of the two subsystems. Instructions fetched by one subsystem are ready for both subsystems to execute. After the application code is executed or transferred to internal memory, write accesses to external memory are prohibited.

This method requires the $\overline{\text{A\_RS}}$ and $\overline{\text{B\_RS}}$ pins to be tied high while $\overline{\text{HPIRS}}$ transitions from low to high. When $\overline{\text{HPIRS}}$ transitions high, both subsystems fetches the same reset vector.

sequential EMIF

The sequential EMIF option allows one master subsystem to run from external memory while controlling the slave subsystem's $\overline{RS}$ signal and the SELA/B pin. At system reset, only the master subsystem is actually reset. Upon a low-to-high transition of the master's $\overline{RS}$ signal, the master subsystem fetches the reset vector and proceeds to copy external application code to internal memory space. The master subsystem begins executing the application code, then changes the state of SELA/B, relinquishing the external EMIF to the slave subsystem. The master then releases the slave $\overline{RS}$ signal. As a result, the slave fetches the reset vector and begins to copy the external application code to internal memory space. Note, GPIO pins on the master subsystem can be used to control the SELA/B and slave reset ($\overline{x\_RS}$) pins externally.

on-chip peripherals

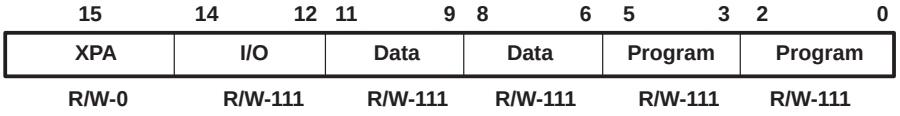
All the '54x devices have the same CPU structure; however, they have different on-chip peripherals connected to their CPUs. The on-chip peripheral options provided on each subsystem of the '5420 are:

- Software-programmable wait-state generator
- Programmable bank-switching
- 16-bit host-port interface (HPI16)
- Multichannel buffered serial ports (McBSPs)
- A hardware timer
- A software-programmable clock generator with a phase-locked loop (PLL)

software-programmable wait-state generators

The Software-programmable wait-state generator can be used to extend external bus cycles up to fourteen machine cycles to interface with slower off-chip memory and I/O devices. Note that all external memory accesses on the '5420 require at least one wait state. The software wait-state register (SWWSR) controls the operation of the wait-state generator. The SWWSR of a particular DSP subsystem (A or B) is used for the external memory interface, depending on the logic level of the SELA/B pin.The 14 LSBs of the SWWSR specify the number of wait states (0 to 7) to be inserted for external memory accesses to five separate address ranges. This allows a different number of wait states for each of the five address ranges.

Additionally, the software wait-state multiplier (SWSM) bit of the software wait-state control register (SWCR) defines a multiplication factor of 1 or 2 for the number of wait states. At reset, the wait-state generator is initialized to provide seven wait states on all external memory accesses. The SWWSR bit fields are shown in Figure 3 and described in Table 2.



LEGEND: R=Read, W=Write, 0=Value after reset

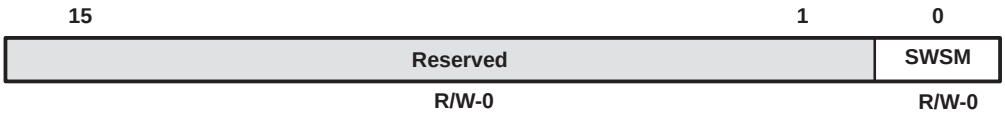
Figure 3. Software Wait-State Register (SWWSR) [Memory-Mapped Register (MMR) Address 0028h]

software-programmable wait-state generator (continued)

Table 2. Software Wait-State Register (SWWSR) Bit Fields

BIT NO.	BIT NAME	RESET VALUE	FUNCTION
15	XPA	0	Extended program address control bit. XPA is used in conjunction with the program space fields (bits 0 through 5) to select the address range for program space wait states.
14–12	I/O	1	I/O space. The field value (0–7) corresponds to the base number of wait states for I/O space accesses within addresses 0000–FFFFh. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.
11–9	Data	1	Upper data space. The field value (0–7) corresponds to the base number of wait states for external data space accesses within addresses 8000–FFFFh. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.
8–6	Data	1	Lower data space. The field value (0–7) corresponds to the base number of wait states for external data space accesses within addresses 0000–7FFFh. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.
5–3	Program	1	Upper program space. The field value (0–7) corresponds to the base number of wait states for external program space accesses within the following addresses: ❑ XPA = 0: x8000 – xFFFFh ❑ XPA = 1: The upper program space bit field has no effect on wait states. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.
2–0	Program	1	Program space. The field value (0–7) corresponds to the base number of wait states for external program space accesses within the following addresses: ❑ XPA = 0: x0000–x7FFFh ❑ XPA = 1: 00000–3FFFFh The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states.

The software wait-state multiplier bit of the software wait-state control register (SWCR) is used to extend the base number of wait states selected by the SWWSR. The SWCR bit fields are shown in Figure 4 and described in Table 3.



LEGEND: R = Read, W = Write

Figure 4. Software Wait-State Control Register (SWCR) [MMR Address 002Bh]

Table 3. Software Wait-State Control Register (SWCR) Bit Fields

PIN NO.	PIN NAME	RESET VALUE	FUNCTION
15–1	Reserved	0	These bits are reserved and are unaffected by writes.
0	SWSM	0	Software wait-state multiplier. Used to multiply the number of wait states defined in the SWWSR by a factor of 1 or 2. ❑ SWSM = 0: wait-state base values are unchanged (multiplied by 1). ❑ SWSM = 1: wait-state base values are mulitplied by 2 for a maximum of 14 wait states.

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. This extra cycle allows memory devices to release the bus before other devices start driving the bus; thereby avoiding bus contention. The size of the memory bank for the bank-switching is defined by the bank-switching control register (BSCR). The BSCR of a particular DSP subsystem (A or B) is used for the external memory interface depending on the logic level of the SELA/B pin.

15 – 12	11	10 – 9	8	7 – 3	2	1	0
BNKCMP	PS-DS	Reserved	IPIRQ	Reserved	BH	Reserved	EXIO
R/W	R/W	R/W	R/W		R/W		R/W

LEGEND: R = Read, W = Write

Figure 5. BSCR Register Bit Layout for Each DSP Subsystem

Table 4. BSCR Register Bit Functions for Each DSP Subsystem

BIT NO.	BIT NAME	RESET VALUE	FUNCTION
15–12	BNKCMP	1111	Bank compare. BNKCMP determines the external memory-bank size. BNKCMP is used to mask the four MSBs of an address. For example, if BNKCMP = 1111b, the four MSBs (bits 12–15) are compared, resulting in a bank size of 4K words. Bank sizes of 4K words to 64K words are allowed.
11	PS-DS	1	Program read – data read access. PS-DS inserts an extra cycle between consecutive accesses of program read and data read or data read and program read. PS-DS = 0 No extra cycles are inserted by this feature. PS-DS = 1 One extra cycle is inserted between consecutive data and program reads.
10–9	Reserved	0	These bits are reserved and are unaffected by writes.
8	IPIRQ	0	The IPIRQ bit is used to send an interprocessor interrupt to the other subsystem. IPIRQ=1 sends the interrupt. IPIRQ must be cleared before subsequent interrupts can be made. Refer to the interrupts section for more details
7–3	Reserved	0	These bits are reserved and are unaffected by writes.
2	BH	0	Bus holder. BH controls the data bus holder feature: BH is cleared to 0 at reset. BH = 0 The bus holder is disabled. BH = 1 The bus holder is enabled. When not driven, the data bus (PPD[15:0]) is held in the previous logic level.
1	Reserved	0	These bits are reserved and are unaffected by writes.
0	EXIO	0	External bus interface off. The EXIO bit controls the external bus-off function. EXIO = 0 The external bus interface functions as usual. EXIO = 1 The address bus, data bus, and control signals become inactive after completing the current bus cycle. Note that the DROM, MP/$\overline{MC}$, and OVLY bits in the PMST and the HM bit of ST1 cannot be modified when the interface is disabled.

16-bit host-port interface (HPI16)

The HPI16 is an enhanced 16-bit version of the ’C54x 8-bit host-port interface (HPI). The HPI16 is designed to allow a 16-bit host to access the DSP on-chip memory, with the host acting as the master of the interface. Figure 6 illustrates the available memory accessible by the HPI. It should be noted that neither the CPU nor DMA I/O spaces can be accessed using the host-port interface.

16-bit bidirectional host-port interface (HPI16)

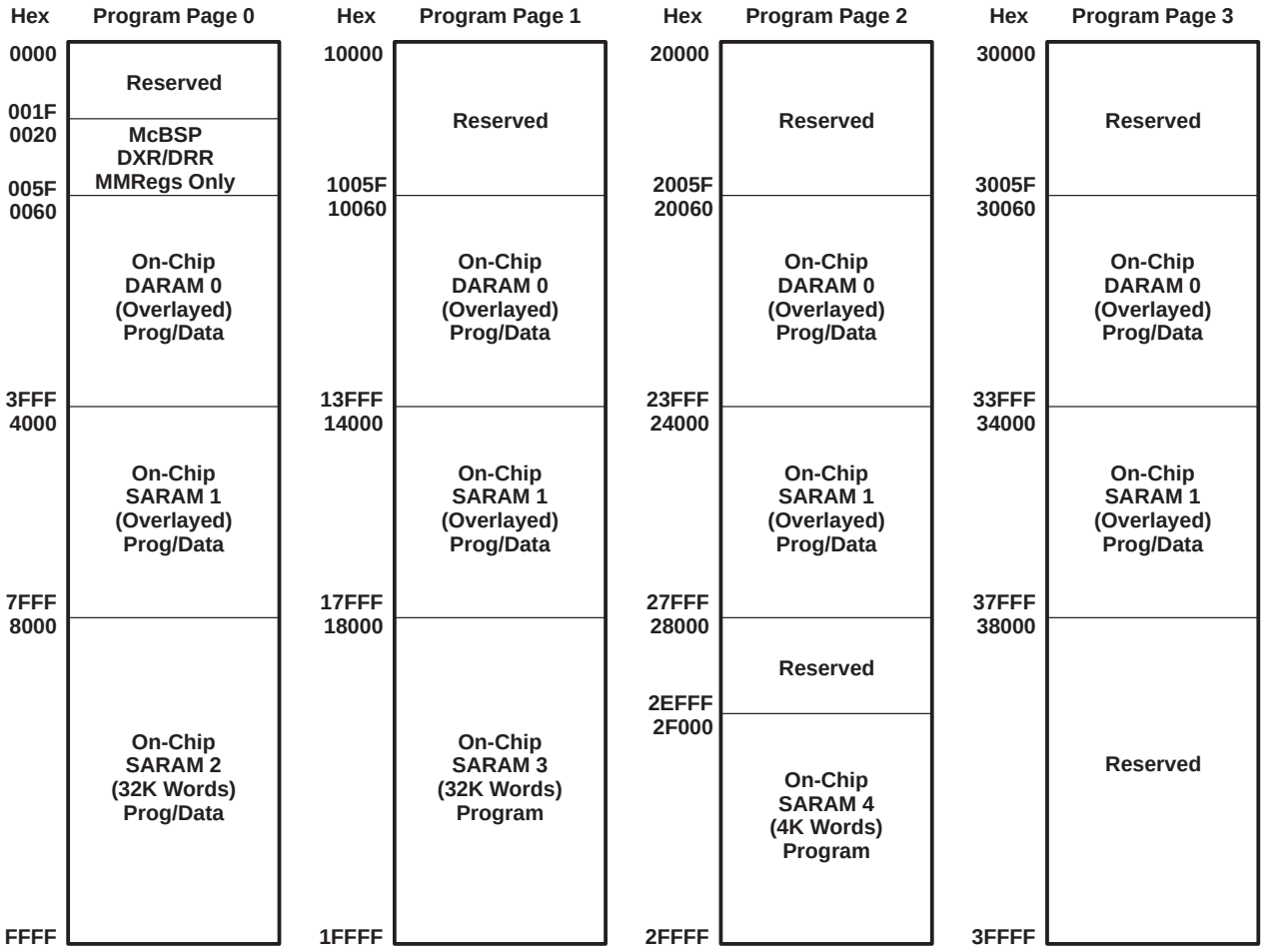


Figure 6. Memory Map Relative to Host-Port interface

16-bit bidirectional host-port interface (HPI16) (continued)

Some of the features of the HPI16 include:

- 16-bit bidirectional data bus
- Multiple data strobes and control signals to allow glueless interfacing to a variety of hosts
- Multiplexed *and* nonmultiplexed address/data modes
- 18-bit address bus used in nonmultiplexed mode to allow access to all internal memory (including internal extended address pages)
- 18-bit address register used in multiplexed mode. Includes address autoincrement feature for faster accesses to sequential addresses
- Interface to on-chip DMA module to allow access to entire internal memory space
- HRDY signal to hold off host accesses due to DMA latency
- Control register available in *multiplexed* mode only. Accessible by either host or DSP to provide host/DSP interrupts, extended addressing, and data prefetch capability

The HPI16 acts as a slave to a 16-bit host processor and allows access to the on-chip memory of the DSP. There are two modes of operation as determined by the HMODE signal: *multiplexed* mode and *nonmultiplexed* mode.

HPI multiplexed mode

In *multiplexed* mode, HPI16 operation is very similar to the standard 8-bit HPI, which is available with other ’C54x products. A host with a multiplexed address/data bus can access the HPI16 data register (HPID), address register (HPIA), or control register (HPIC) via the HD bidirectional data bus. The host initiates the access with the strobe signals ($\overline{\text{HDS1}}$, $\overline{\text{HDS2}}$, $\overline{\text{HCS}}$) and controls the type of access with the HCNTL, HR/$\overline{\text{W}}$, and $\overline{\text{HAS}}$ signals. The DSP can interrupt the host via the $\overline{\text{HINT}}$ signal, and can stall host accesses via the HRDY signal.

host/DSP interrupts

In *multiplexed* mode, the HPI16 offers the capability for the host and DSP to interrupt each other through the HPIC register.

For host-to-DSP interrupts, the host must write a “1” to the DSPINT bit of the HPIC register. This generates an interrupt to the DSP. This interrupt can also be used to wake the DSP from any of the IDLE 1,2, or 3 states. Note that the DSPINT bit is always read as “0” by both the host and DSP.

For DSP-to-host interrupts, the DSP must write a “1” to the $\overline{\text{HINT}}$ bit of the HPIC register to interrupt the host via the $\overline{\text{HINT}}$ pin. The host acknowledges and clear this interrupt by also writing a “1” to the $\overline{\text{HINT}}$ bit of the HPIC register. Note that writing a “0” to the $\overline{\text{HINT}}$ bit by either host or DSP has no effect.

HPI nonmultiplexed mode

In *nonmultiplexed* mode, a host with separate address/data buses can access the HPI16 data register (HPID) via the HD 16-bit bidirectional data bus, and the address register (HPIA) via the 18-bit HA address bus. The host initiates the access with the strobe signals ($\overline{\text{HDS1}}$, $\overline{\text{HDS2}}$, $\overline{\text{HCS}}$) and controls the direction of the access with the HR/$\overline{\text{W}}$ signal. The HPI16 can stall host accesses via the HRDY signal. Note that the HPIC register is not available in *nonmultiplexed* mode since there are no HCNTL signals available. All host accesses initiate a DMA read or write access.

other HPI16 system considerations

operation during IDLE2

The HPI16 can continue to operate during IDLE1 or IDLE2 by using special clock management logic that turns on relevant clocks to perform a synchronous memory access, and then turns the clocks back off to save power. The DSP CPU does not wake up from the IDLE mode during this process.

downloading code during reset

The HPI16 can download code while the DSP is in reset. However, the system provides a pin ($\overline{\text{HPIRS}}$) that provides a way to take the HPI16 module out of reset while leaving the DSP in reset.

emulation considerations

The HPI16 can continue operation even when the DSP CPU is halted due to debugger breakpoints or other emulation events.

5420 boundary scan implementation

The ’5420 does not implement a fully compliant IEEE1149.1 boundary scan capability. Observe-only boundary scan cells are used on all of the device pins that allow the pins to be observed (read) but not controlled (driven) using boundary scan. Driving nodes to perform board interconnect test must be accomplished using other boundary scan capable devices on the board. Although this implies some reduction in testability, compared to full boundary scan, this implementation is still compatible with the boundary scan automatic test pattern generation (ATPG) tools.

multichannel buffered serial port (McBSP)

The ’5420 device provides high-speed, full-duplex serial ports that allow direct interface to other ’C54x devices, codecs, and other devices in a system. There are six multichannel buffered serial ports (McBSPs) on chip (three per subsystem).

The McBSP is based on the standard serial port interface found on the ’54x devices. Like its predecessors, the McBSP provides:

- Full-duplex communication
- Double-buffer data registers, which allow a continuous data stream
- Independent framing and clocking for receive and transmit

In addition, the McBSP has the following capabilities:

- Direct interface to:
  - T1/E1 framers
  - MVIP switching-compatible and ST-BUS compliant devices
  - IOM-2 compliant device
  - Serial peripheral interface devices
- Multichannel transmit and receive of up to 128 channels
- A wide selection of data sizes, including: 8, 12, 16, 20, 24, or 32 bits
- μ-law and A-law companding
- Programmable polarity for both frame synchronization and data clocks
- Programmable internal clock and frame generation

The McBSP consists of a data path and control path. The six pins, BDX, BDR, BFSX, BFSR, BCLKX, and BCLKR, connect the control and data paths to external devices. The pins can be programmed as general-purpose I/O pins if they are not used for serial communication.

Like the standard serial port interface on the McBSP, the data is communicated to devices interfacing to the McBSP by way of the data transmit (BDX) pin for transmit and the data receive (BDR) pin for receive. Control information in the form of clocking and frame synchronization is communicated by way of BCLKX, BCLKR, BFSX, and BFSR. The device communicates to the McBSP by way of 16-bit-wide control registers accessible via the internal peripheral bus. The CPU or DMA reads the received data from the data receive register (DRR) and writes the data to be transmitted to the data transmit register (DXR). Data written to the DXR is shifted out

multichannel buffered serial port (McBSP) (continued)

to BDX by way of the transmit shift register (XSR). Similarly, receive data on the BDR pin is shifted into the receive shift register (RSR) and copied into the receive buffer register (RBR). RBR is then copied to DRR, which can be read by the CPU or DMA. This allows internal data movement and external data communications simultaneously. The control block consists of internal clock generation, frame synchronization signal generation, and their control, and multichannel selection. This control block sends notification of important events to the CPU and DMA by way of two interrupt signals, XINT and RINT, and two event signals, XEVT and REVT.

The on-chip companding hardware allows compression and expansion of data in either μ-law or A-law format. When companding is used, transmitted data is encoded according to specified companding law and received data is decoded to 2's complement format.

The sample rate generator provides the McBSP with several means of selecting clocking and framing for both the receiver and transmitter. Both the receiver and transmitter can select clocking and framing independently.

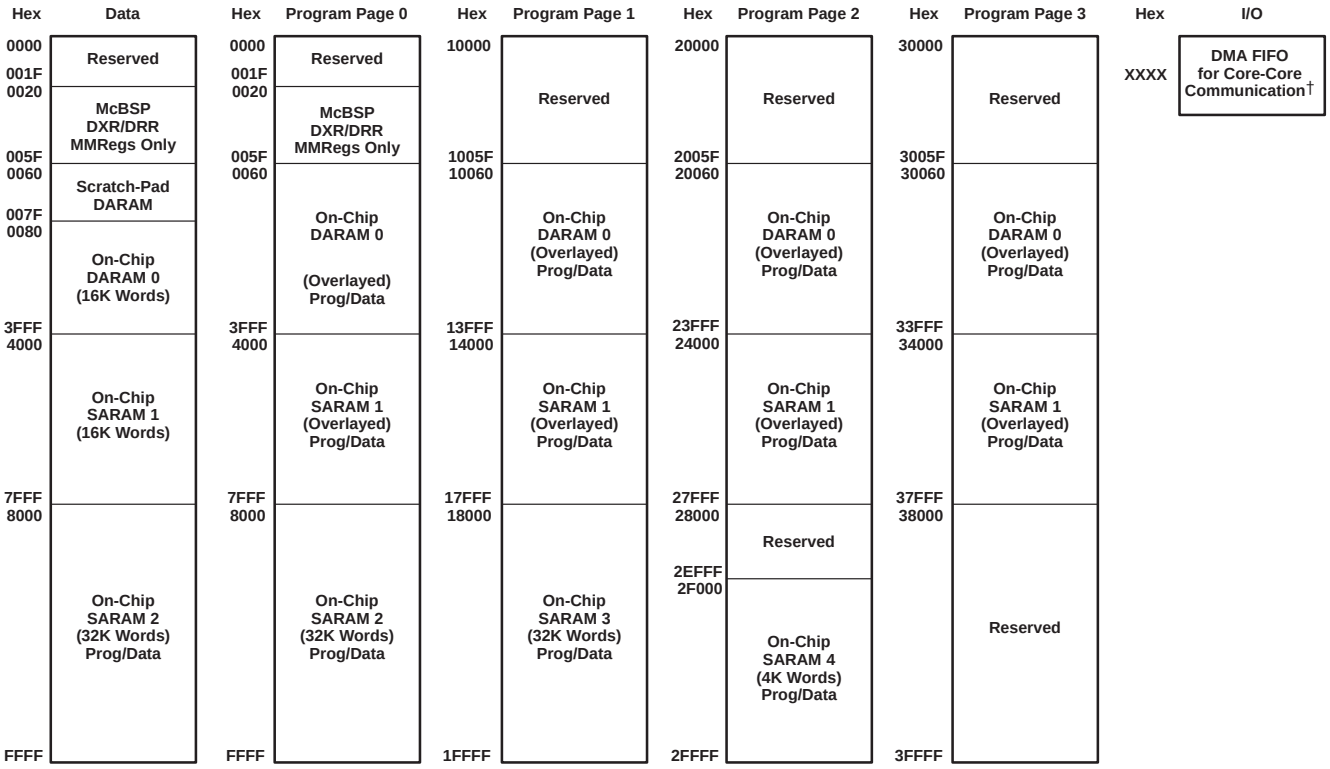
The McBSP allows the multiple channels to be independently selected for the transmitter and receiver. When multiple channels are selected, each frame represents a time-division multiplexed (TDM) data stream. In using time-division multiplexed data streams, the CPU may only need to process a few of them.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

The clock stop mode (CLKSTP) in the McBSP provides compatibility with the SPI protocol. Clock stop mode works with only single-phase frames and one word per frame. The word sizes supported by the McBSP are programmable for 8-, 12-, 16-, 20-, 24-, or 32-bit operation. When the McBSP is configured to operate in SPI mode, both the transmitter and the receiver operate together as a master or as a slave.

direct memory access unit (DMA)

The '5420 direct memory access (DMA) controller transfers data between points in the memory map without intervention by the CPU. The DMA allows movements of data to and from internal program/data memory, internal peripherals, such as the McBSPs and the HPI to occur in the background of the CPU operation. Each subsystem has its own independent DMA with six programmable channels, allowing six different contexts for DMA operation. The HPI has a dedicated auxiliary DMA channel. Figure 7 illustrates the memory map accessible by the DMA.

direct memory access unit (DMA) (continued)



† When the source or destination for a DMA channel is programmed for I/O space, the channel accesses the core-to-core FIFO irrespective of the address specified.

Figure 7. Memory Map Relative to DMA

features

The ’5420 DMA has the following features:

- The DMA operates independently of the CPU.
- The DMA has six channels. The DMA can keep track of the contexts of six independent block transfers.
- The DMA has higher priority than the CPU for internal accesses.
- Each channel has independently programmable priorities.
- Each channel’s source and destination address registers can have configurable indexes through memory on each read and write transfer, respectively. The address can remain constant, postincrement, postdecrement or be adjusted by a programmable value.
- Each read or write transfer can be initialized by selected events.
- On completion of a half-block or full-block transfer, each DMA channel can send an interrupt to the CPU.
- An on-chip RAM DMA transfer requires 4 clock cycles to complete. External transfers are not supported.
- The DMA can perform double word transfers (a 32-bit transfer of two16-bit-words).

DMA controller synchronization events

The transfers associated with each DMA channel can be synchronized to one of several events. The DSYN bit field of the DMA channel x sync select and frame count (DMSFCx) register selects the synchronization event for a channel. The list of possible events and the DSYN values are shown in Table 5.

Table 5. DMA Synchronization Events

DSYN VALUE	DMA SYNCHRONIZATION EVENT
0000b	No synchronization used
0001b	McBSP0 Receive Event
0010b	McBSP0 Transmit Event
0011b	McBSP2 Receive Event
0100b	McBSP2 Transmit Event
0101b	McBSP1 Receive Event
0110b	McBSP1 Transmit Event
0111b	FIFO Receive Buffer Not Empty Event
1000b	FIFO Transmit Buffer Not Full Event
1001b – 1111b	Reserved

DMA channel interrupt selection

The DMA controller can generate a CPU interrupt for each of the six channels. However, channels 0, 1, 2, and 3 are multiplexed with other interrupt sources. DMA channels 0 and 1 share an interrupt line with the receive and transmit portions of McBSP2 (IMR/IFR bits 6 and 7), and DMA channels 2 and 3 share an interrupt line with the receive and transmit portions of McBSP1 (IMR/IFR bits 10 and 11). When the ’5402 is reset, the interrupts from these four DMA channels are deselected. The INTSEL bit field in the DMA channel priority and enable control (DMPREC) register can be used to select these interrupts, as shown in Table 6.

Table 6. DMA Channel Interrupt Selection

INTSEL Value	IMR/IFR[6]	IMR/IFR[7]	IMR/IFR[10]	IMR/IFR[11]
00b (reset)	BRINT2	BXINT2	BRINT1	BXINT1
01b	BRINT2	BXINT2	DMAC2	DMAC3
10b	DMAC0	DMAC1	DMAC2	DMAC3
11b	Reserved			

subsystem communications

The ’5420 device provides two options for efficient core-to-core communications:

- Core-to-core FIFO communications
- EMIF-to-HPI communications (asynchronous external memory interface-to host-port interface)

FIFO data communications

The subsystems’ FIFO communications interface is shown in the ’5420 functional block diagram (Figure 1). Two unidirectional 8-word-deep FIFOs are available in the device for efficient interprocessor communication: one configured for core A-to-core B data transfers, and the other configured for core B-to-core A data transfers. Each subsystem, by way of DMA control, can write to its respective output data FIFO and read from its respective input data FIFO. The FIFOs are accessed using the DMA’s I/O space, which is completely independent of the CPU I/O space. The DMA transfers to or from the FIFOs can be synchronized to “receive FIFO not empty” and “transmit FIFO not full” events providing protection from overflow and underflow. Subsystems can interrupt each

FIFO data communications (continued)

other to flag when the FIFOs are either full or empty. The interprocessor interrupt request bit (IPIRQ) (bit 8 in the BSCR register) is set to “1” to generate an interprocessor interrupt (IPINT) in the other subsystem. See the *interrupts* section for more information.

EMIF-to-HPI data communication

The ’5420 also provides the capability for one subsystem to act as a master and transfer data to the other subsystem via an EMIF-to-HPI connection. The master device is configured in EMIF mode (XIO pin is high); while by default, when HMODE=1, the slave device external interface is configured to operate as an HPI (nonmultiplexed mode). The data-transfer direction is defined by the logic level of SELA/B. See Table 7 for a complete description of HMODE, SELA/B, and XIO pin functionality. The EMIF-to-HPI option is bidirectional, but does not permit full duplex communication without external SELA/B arbitration. This mode does not offer master/slave interrupts due to the nonmultiplexed HPI configuration.

Table 7. EMIF/HPI Modes

HMODE	SELA/B	HPI MODES (XIO PIN =0)	EMIF MODES (XIO PIN = 1)
0	0	HPI multiplexed address/data Subsystem A slave to host	Subsystem A can access EMIF Subsystem B has no access to EMIF or HPI
0	1	HPI multiplexed address/data Subsystem B slave to host	Subsystem B can access EMIF Subsystem A has no access to EMIF or HPI
1	0	HPI nonmultiplexed address/data Subsystem A slave to host	EMIF-to-HPI master is subsystem A, slave is subsystem B
1	1	HPI nonmultiplexed address/data Subsystem B slave to host	EMIF-to-HPI master is subsystem B, slave is subsystem A

general-purpose I/O

In addition to the standard XF and $\overline{\text{BIO}}$ pins, the ’5420 has eight general-purpose I/O pins. These pins are:

A_GPIO0, A_GPIO1, A_GPIO2, A_GPIO3

B_GPIO0, B_GPIO1, B_GPIO2, B_GPIO3

Each subsystem’s CPU has one general-purpose I/O register located at address 0x3c in data memory. Each I/O register controls four general-purpose I/O pins. Figure 8 shows the bit layout of the general-purpose I/O control register and Table 8 describes the bit functions.

general-purpose I/O (continued)

15	14 12	11	10	9	8	7 4	3	2	1	0
TOUT	Reserved	DIR3	DIR2	DIR1	DIR0	Reserved	DAT3	DAT2	DAT1	DAT0
R/W		R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W

LEGEND: R = Read, W = Write

Figure 8. General-Purpose I/O Control Register Bit Layout

Table 8. General-Purpose I/O Control Register Bit Functions

BIT NO.	BIT NAME	BIT VALUE	FUNCTION
15	TOUT	0	Timer output disable. Uses GPIO3 as general-purpose I/O. ***(Reset value)***
		1	Timer output enable. Overrides DIR3. Timer output is driven on GPIO3 and readable in DAT3.
14-12	Reserved	X	Register bit is reserved.
11–8	DIRn†	0	GPIOn pin is used as an input. ***(Reset value)***
		1	GPIOn pin is used as an output.
7–4	Reserved	X	Register bit is reserved.
3–0	DATn†	0	GPIOn is driven with a 0 (DIRn=1). GPIOn is read as 0 (DIRn=0).
		1	GPIOn is driven with a 1 (DIRn=1). GPIOn is read as 1 (DIRn=0).

† n = 0, 1, 2, or 3

The TOUT bit is used to multiplex the output of the timer and GPIO3. DIR3 has no affect when TOUT = 1. All pins are programmable as an input or output via the direction bit (DIRn). Data is either driven or read from the data bit field (DATn).

GPIO2 is a special case where the logic level determines the operation of $\overline{\text{BIO}}$-conditional instructions on the CPU. GPIO2 is always mapped as a general-purpose I/O, but the $\overline{\text{BIO}}$ function exists when this pin is configured as an input.

hardware timer

Each subsystem of the ’5420 features a 16-bit timing circuit with a 4-bit prescaler. The timer counter decrements 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. The timer output pulse is driven on GPIO3 when the TOUT bit is set to one in the general-purpose I/O control register. The device must be in HPI mode (XIO = 0) to drive TOUT on the GPIO3 pin.

software-programmable phase-locked loop (PLL)

The clock generator provides clocks to the ’5420 device, and consists of a phase-locked loop (PLL) circuit. The clock generator requires a reference clock input, which must be provided by using an external clock source. The reference clock input is then divided by two (DIV mode) to generate clocks for the ’5420 device. Alternately, the PLL circuit can be used (PLL mode) 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 default startup mode for the PLL on the ’5420 device is bypass (multiply-by-1).

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. Only subsystem A controls the PLL. Subsystem B cannot access the PLL registers.

software-programmable phase-locked loop (PLL) (continued)

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 (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 by the 16-bit memory-mapped (address 0058h) clock mode register (CLKMD) in subsystem A. The CLKMD register is used to define the clock configuration of the PLL clock module.

memory-mapped registers

Each ’5420 subsystem has 27 memory-mapped CPU registers, which are mapped in data memory space addresses 0h to 1Fh. Table 9 gives a list of CPU memory-mapped registers (MMRs) available on ’5420. Each subsystem device also has a set of memory-mapped registers associated with peripherals. Table 10, and Table 11 show additional peripheral MMRs associated with the ’5420.

Table 9. Processor Memory-Mapped Registers for Each DSP Subsystem

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
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 Register
XPC	30	1E	Extended Program Counter
—	31	1F	Reserved

memory-mapped registers (continued)

Table 10. Peripheral Memory-Mapped Registers for Each DSP Subsystem

NAME	ADDRESS DEC	ADDRESS HEX	DESCRIPTION
DRR20	32	20	McBSP 0 Data Receive Register 2
DRR10	33	21	McBSP 0 Data Receive Register 1
DXR20	34	22	McBSP 0 Data Transmit Register 2
DXR10	35	23	McBSP 0 Data Transmit Register 1
TIM	36	24	Timer Register
PRD	37	25	Timer Period Register
TCR	38	26	Timer Control Register
—	39	27	Reserved
SWWSR	40	28	Software Wait-State Register
BSCR	41	29	Bank-Switching Control Register
—	42	2A	Reserved
SWCR	43	2B	Software Wait-State Control Register
HPIC	44	2C	HPI Control Register (HMODE=0 only)
—	45–47	2D–2F	Reserved
DRR22	48	30	McBSP 2 Data Receive Register 2
DRR12	49	31	McBSP 2 Data Receive Register 1
DXR22	50	32	McBSP 2 Data Transmit Register 2
DXR12	51	33	McBSP 2 Data Transmit Register 1
SPSA2	52	34	McBSP 2 Subbank Address Register†
SPSD2	53	35	McBSP 2 Subbank Data Register†
—	54–55	36–37	Reserved
SPSA0	56	38	McBSP 0 Subbank Address Register†
SPSD0	57	39	McBSP 0 Subbank Data Register†
—	58–59	3A–3B	Reserved
GPIO	60	3C	General-Purpose I/O Register
—	61–63	3D–3F	Reserved
DRR21	64	40	McBSP 1 Data Receive Register 2
DRR11	65	41	McBSP 1 Data Receive Register 1
DXR21	66	42	McBSP 1 Data Transmit Register 2
DXR11	67	43	McBSP 1 Data Transmit Register 1
—	68–71	44–47	Reserved
SPSA1	72	48	McBSP 1 Subbank Address Register†
SPSD1	73	49	McBSP 1 Subbank Data Register†
—	74–83	4A–53	Reserved
DMPREC	84	54	DMA Priority and Enable Control Register
DMSA	85	55	DMA Subbank Address Register‡
DMSDI	86	56	DMA Subbank Data Register with Autoincrement‡
DMSDN	87	57	DMA Subbank Data Register‡
CLKMD	88	58	Clock Mode Register (CLKMD)
—	89–95	59–5F	Reserved

† See Table 11 for a detailed description of the McBSP control registers and their subaddresses.
‡ See Table 12 for a detailed description of the DMA sub-bank addressed registers.

McBSP control registers and subaddresses

The control registers for the multichannel buffered serial port (McBSP) are accessed using the subbank addressing scheme. This allows a set or subbank of registers to be accessed through a single memory location. The McBSP subbank address register (SPSA) is used as a pointer to select a particular register within the subbank. The McBSP data register (SPSDI) and the DMA autoincrement subaddress register (SPSDN) register are used to access (read or write) the selected register. Table 11 shows the McBSP control registers and their corresponding subaddresses.

Table 11. McBSP Control Registers and Subaddresses

McBSP0		McBSP1		McBSP2		SUB-ADDRESS	DESCRIPTION
NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS		
SPCR10	39h	SPCR11	49h	SPCR12	35h	00h	Serial port control register 1
SPCR20	39h	SPCR21	49h	SPCR22	35h	01h	Serial port control register 2
RCR10	39h	RCR11	49h	RCR12	35h	02h	Receive control register 1
RCR20	39h	RCR21	49h	RCR22	35h	03h	Receive control register 2
XCR10	39h	XCR11	49h	XCR12	35h	04h	Transmit control register 1
XCR20	39h	XCR21	49h	XCR22	35h	05h	Transmit control register 2
SRGR10	39h	SRGR11	49h	SRGR12	35h	06h	Sample rate generator register 1
SRGR20	39h	SRGR21	49h	SRGR22	35h	07h	Sample rate generator register 2
MCR10	39h	MCR11	49h	MCR12	35h	08h	Multichannel register 1
MCR20	39h	MCR21	49h	MCR22	35h	09h	Multichannel register 2
RCERA0	39h	RCERA1	49h	RCERA2	35h	0Ah	Receive channel enable register partition A
RCERB0	39h	RCERB1	49h	RCERA2	35h	0Bh	Receive channel enable register partition B
XCERA0	39h	XCERA1	49h	XCERA2	35h	0Ch	Transmit channel enable register partition A
XCERB0	39h	XCERB1	49h	XCERA2	35h	0Dh	Transmit channel enable register partition B
PCR0	39h	PCR1	49h	PCR2	35h	0Eh	Pin control register

DMA subbank addressed registers

The direct memory access (DMA) controller has several control registers associated with it. The main control register (DMPREC) is a standard memory-mapped register. However, the other registers are accessed using the subbank addressing scheme. This allows a set or subbank of registers to be accessed through a single memory location. The DMA subbank address (DMSA) register is used as a pointer to select a particular register within the subbank, while the DMA subbank data (DMSD) register or the DMA subbank data register with autoincrement (DMSDI) is used to access (read or write) the selected register.

When the DMSDI register is used to access the subbank, the subbank address is automatically postincremented so that a subsequent access affects the next register within the subbank. This autoincrement feature is intended for efficient, successive accesses to several control registers. If the autoincrement feature is not required, the DMSDN register should be used to access the subbank. Table 12 shows the DMA controller subbank addressed registers and their corresponding subaddresses.

DMA subbank addressed registers (continued)

Table 12. DMA Subbank Addressed Registers

NAME	ADDRESS	SUB-ADDRESS	DESCRIPTION
DMSRC0	56h/57h	00h	DMA channel 0 source address register
DMDST0	56h/57h	01h	DMA channel 0 destination address register
DMCTR0	56h/57h	02h	DMA channel 0 element count register
DMSFC0	56h/57h	03h	DMA channel 0 sync select and frame count register
DMMCR0	56h/57h	04h	DMA channel 0 transfer mode control register
DMSRC1	56h/57h	05h	DMA channel 1 source address register
DMDST1	56h/57h	06h	DMA channel 1 destination address register
DMCTR1	56h/57h	07h	DMA channel 1 element count register
DMSFC1	56h/57h	08h	DMA channel 1 sync select and frame count register
DMMCR1	56h/57h	09h	DMA channel 1 transfer mode control register
DMSRC2	56h/57h	0Ah	DMA channel 2 source address register
DMDST2	56h/57h	0Bh	DMA channel 2 destination address register
DMCTR2	56h/57h	0Ch	DMA channel 2 element count register
DMSFC2	56h/57h	0Dh	DMA channel 2 sync select and frame count register
DMMCR2	56h/57h	0Eh	DMA channel 2 transfer mode control register
DMSRC3	56h/57h	0Fh	DMA channel 3 source address register
DMDST3	56h/57h	10h	DMA channel 3 destination address register
DMCTR3	56h/57h	11h	DMA channel 3 element count register
DMSFC3	56h/57h	12h	DMA channel 3 sync select and frame count register
DMMCR3	56h/57h	13h	DMA channel 3 transfer mode control register
DMSRC4	56h/57h	14h	DMA channel 4 source address register
DMDST4	56h/57h	15h	DMA channel 4 destination address register
DMCTR4	56h/57h	16h	DMA channel 4 element count register
DMSFC4	56h/57h	17h	DMA channel 4 sync select and frame count register
DMMCR4	56h/57h	18h	DMA channel 4 transfer mode control register
DMSRC5	56h/57h	19h	DMA channel 5 source address register
DMDST5	56h/57h	1Ah	DMA channel 5 destination address register
DMCTR5	56h/57h	1Bh	DMA channel 5 element count register
DMSFC5	56h/57h	1Ch	DMA channel 5 sync select and frame count register
DMMCR5	56h/57h	1Dh	DMA channel 5 transfer mode control register
DMSRCP	56h/57h	1Eh	DMA source program page address (common channel)
DMDSTP	56h/57h	1Fh	DMA destination program page address (common channel)
DMIDX0	56h/57h	20h	DMA element index address register 0
DMIDX1	56h/57h	21h	DMA element index address register 1
DMFRI0	56h/57h	22h	DMA frame index register 0
DMFRI1	56h/57h	23h	DMA frame index register 1
DMGSA	56h/57h	24h	DMA global source address reload register
DMGDA	56h/57h	25h	DMA global destination address reload register
DMGCR	56h/57h	26h	DMA global count reload register
DMGFR	56h/57h	27h	DMA global frame count reload register

interrupts

Vector-relative locations and priorities for all internal and external interrupts are shown in Table 13.

Table 13. ’5420 Interrupt Locations and Priorities for Each DSP Subsystem

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	Reserved
TINT, SINT3	76	4C	6	External Timer Interrupt
BRINT0, SINT4	80	50	7	McBSP #0 Receive Interrupt
BXINT0, SINT5	84	54	8	McBSP #0 Transmit Interrupt
BRINT2 (DMAC0), SINT6	88	58	9	McBSP #2 Receive Interrupt (default) or DMA Channel 0 interrupt. The selection is made in the DMPREC register.
BXINT2 (DMAC1), SINT7	92	5C	10	McBSP #2 Receive Interrupt (default) or DMA Channel 1 interrupt. The selection is made in the DMPREC register.
$\overline{\text{INT3}}$, SINT8	96	60	11	Reserved
HPINT, SINT9	100	64	12	HPI Interrupt (from DSPINT in HPIC)
BRINT1 (DMAC2), SINT10	104	68	13	McBSP #1 Receive Interrupt (default) or DMA Channel 2 interrupt. The selection is made in the DMPREC register.
BXINT1 (DMAC3), SINT11	108	6C	14	McBSP #1 transmit Interrupt (default) or DMA channel 3 interrupt. The selection is made in the DMPREC register.
DMAC4, SINT12	112	70	15	DMA Channel 4
DMAC5, SINT13	116	74	16	DMA Channel 5
IPINT, SINT14	120	78	17	Interprocessor Interrupt
—	124–127	7C–7F	—	Reserved

interrupts (continued)

Figure 9 shows the bit layout of the interrupt mask register (IMR) and the interrupt flag register (IFR). Table 14 describes the bit functions.

The interprocessor interrupt (IPINT) bit of the interrupt mask register (IMR) and the interrupt flag register (IFR) allows the subsystem to perform interrupt service routines based on the other subsystem activity. Incoming IPINT interrupts are latched in bit 14 of the IFR. Generating an interprocessor interrupt is performed by writing a “1” to the IPIRQ field of the bank-switching control register (BSCR). Subsequent interrupts must first clear the interrupt by writing “0” to the IPIRQ field.

For example, if subsystem A is required to notify subsystem B of a completed task, subsystem A must write a “1” to the IPIRQ field to generate a IPINT interrupt on subsystem B. On subsystem B, the IPINT interrupt is latched in bit 14 of the IFR.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	IPINT	DMAC5	DMAC4	BXINT1 or DMAC3	BRINT1 or DMAC2	HPINT	RES	BXINT2 or DMAC1	BRINT2 or DMAC0	BXINT0	BRINT0	TINT	RES	INT1	INT0

Figure 9. Bit Layout of the IMR and IFR Registers for Each Subsystem

Table 14. Bit Functions for IMR and IFR Registers for Each DSP Subsystem

BIT		FUNCTION
NUMBER	NAME	
15	–	Reserved
14	IPINT	Interprocessor IRQ.
13	DMAC5	DMA channel 5 interrupt flag/mask bit
12	DMAC4	DMA channel 4 interrupt flag/mask bit
11	BXINT1/DMAC3	This bit can be configured as either the McBSP1 transmit interrupt flag/mask bit, or the DMA channel 3 interrupt flag/mask bit. The selection is made in the DMPREC register.
10	BRINT1/DMAC2	This bit can be configured as either the McBSP1 receive interrupt flag/mask bit, or the DMA channel 2 interrupt flag/mask bit. The selection is made in the DMPREC register.
9	HPINT	Host to ’54x interrupt flag/mask
8	–	Reserved
7	BXINT2/DMAC1	This bit can be configured as either the McBSP2 transmit interrupt flag/mask bit, or the DMA channel 1 interrupt flag/mask bit. The selection is made in the DMPREC register.
6	BRINT2/DMAC0	This bit can be configured as either the McBSP2 receive interrupt flag/mask bit, or the DMA channel 0 interrupt flag/mask bit. The selection is made in the DMPREC register.
5	BXINT0	McBSP0 transmit interrupt flag/mask bit
4	BRINT0	McBSP0 receive interrupt flag/mask bit
3	TINT	Timer interrupt flag/mask bit
2	–	Reserved
1	INT1	External interrupt 1 flag/mask bit
0	INT0	External interrupt 0 flag/mask bit

IDLE3 power-down mode

The IDLE1 and IDLE2 power-down modes operate as described in the *TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals* (literature number SPRU131). The IDLE3 mode is special in that the clocking circuitry is shut off to conserve power. The ’5420 cannot enter an IDLE3 mode unless both subsystems execute an IDLE3 instruction. The power-reduced benefits of IDLE3 cannot be realized until both subsystems enter the IDLE3 state and the internal clocks are automatically shut off. The order in which subsystems enter IDLE3 does not matter.

emulating the ’5420 device

The ’5420 is a single device, but actually consists of two independent subsystems that contain register/status information used by the emulator tools. The emulator tools must be informed of the multicore device by modifying the **board.cfg** file. The board.cfg file is an ASCII file that can be modified with most editors. This provides the emulator with a description of the JTAG chain. The board.cfg file must identify two processors when using the ’5420. The file contents would look something like this:

“CPU_B” TI320C5xx

“CPU_A” TI320C5xx

Use the compose program to make this file into a binary file (**board.dat**), readable by the emulation tools. Place the board.dat file in the directory that contains the emulator software.

The subsystems are serially connected together internally. Emulation information is serially transmitted into the device using TDI. The device responds with serial scan information transmitted out the TDO pin.

documentation support

Extensive documentation supports all TMS320 family generations of devices from product announcement through applications development. The following types of documentation are available to support the design and use of the ’C5000 family of DSPs:

- *TMS320C5000 DSP Family Functional Overview* (literature number SPRU307)
- Device-specific data sheets (such as this document)
- Complete User Guides
- Development-support tools
- Hardware and software application reports

The five-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)
- *Volume 5: Enhanced Peripherals* (literature number SPRU302)

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 Texas Instruments (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.

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

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

Supply voltage I/O range, DV_{DD}‡ . . . . . – 0.5 V to 4.0 V
Supply voltage core range, CV_{DD}‡ . . . . . – 0.5 V to 2.0 V
Supply voltage analog PLL, AV_{DD}‡ . . . . . – 0.5 V to 2.0 V
Input voltage range, V_I . . . . . – 0.5 V to DV_{DD} + 0.5 V
Output voltage range, V_o . . . . . – 0.5 V to DV_{DD} + 0.5 V
Operating case temperature range, T_C . . . . . – 40°C to 100°C
Storage temperature range T_{stg} . . . . . – 65°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		3.0	3.3	3.6	V
CV_{DD}	Device supply voltage, core		1.71	1.80	1.98	V
AV_{DD}	Device supply voltage, PLL		1.71	1.80	1.98	V
V_{SS}	Supply voltage, GND			0		V
V_{IH}	High-level input voltage, I/O	Schmitt trigger inputs, TRST, SELA/B DV_{DD} = 3.3 ± 0.3 V	$0.7DV_{DD}$		DV_{DD}	V
		All other inputs	2		DV_{DD}	
V_{IL}	Low-level input voltage, I/O	Schmitt trigger inputs DV_{DD} = 3.3 ± 0.3 V	0		$0.3DV_{DD}$	V
		All other inputs	0		0.8	
I_{OH}	High-level output current				–300	µA
I_{OL}	Low-level output current				1.5	mA
T_C	Operating case temperature, industrial		–40		100	°C
	Operating case temperature, commercial		0		85	

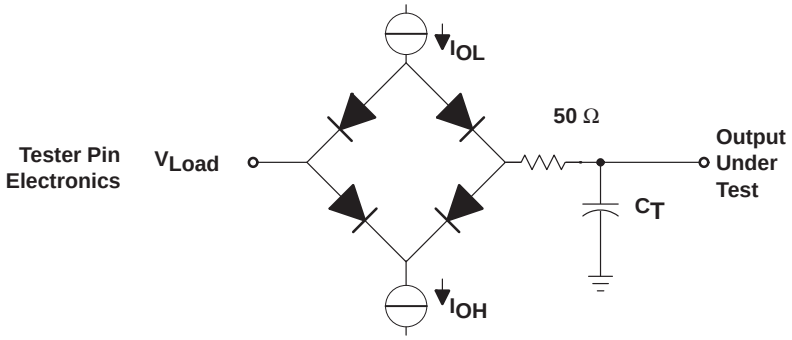
Refer to Figure 10 for 3.3-V device test load circuit values.

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

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage‡		DV_{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		DV_{DD} = MAX, V_O = V_{SS} to DV_{DD}	–10		10	µA
I_I	Input current (V_I = V_{SS} to V_{DD})	TRST	With internal pulldown	–10		35	µA
		See pin descriptions	With internal pullups	–35		10	
		PPD[15:0]	Bus holders enabled, DV_{DD} = MAX, V_I = V_{SS} to V_{IL} (MAX); V_{IH} (MIN) to DV_{DD}	–200		200	
		All other input-only pins		–10		10	
I_{DDC}	Supply current, both core CPUs		CV_{DD} = 1.8 V, f_X=100 MHz§, T_C=25°C#		180		mA
I_{DDP}	Supply current, pins		DV_{DD} = 3.3 V, f_{clock} = 100 MHz¶, T_C = 25°C‖		54		mA
I_{DDA}	Supply current, PLL				5		mA
I_{DDC}	Supply current, standby	IDLE2	PLL × n mode, 20 MHz input		2		mA
		IDLE3	PLL x n mode, 20 MHz input		600		µA
C_i	Input capacitance				5		pF
C_o	Output capacitance				5		pF

† All values are typical unless otherwise specified.
‡ All input and output voltage levels except A_RS, B_RS, INT0 INT1, NMI, CLKIN, BCLKX, BCLKR, HAS, HCS, TCK, TRST, SELA/B, HDS1, HDS2, and HPIRS are LVTTL-compatible.
§ Clock mode: PLL × 1 with external source
# This value represents the current consumption of the CPU, on-chip memory, and on-chip peripherals. Conditions include: program execution from on-chip RAM, with 50% usage of MAC and 50% usage of NOP instructions. Actual operating current varies with program being executed.
‖ This value was obtained using the following conditions: external memory writes at a rate of 20 million writes per second, CLKOFF=0, full-duplex operation of all six McBSPs at a rate of 10 million bits per second each, and 15-pF loads on all outputs. For more details on how this calculation is performed, refer to the *Calculation of TMS320C54x Power Dissipation Application Report* (literature number SPRA164).

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 10. 3.3-V Test Load Circuit

external multiply-by-N clock option

An external frequency can be used by injecting the frequency directly into CLKIN. 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 11 and the recommended operating conditions table)

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

† N is the PLL multiplier. N = 1 – 15

timing requirements (see Figure 11)

			MIN	MAX	UNIT
$t_{c(CI)}$	Cycle time, CLKIN	Integer PLL multiplier N	10N	200	ns
		PLL multiplier N = x.5	10N	100	
		PLL multiplier N = x.25, x.75	10N	50	
$t_{f(CI)}$	Fall time, CLKIN			8	ns
$t_{r(CI)}$	Rise time, CLKIN			8	ns

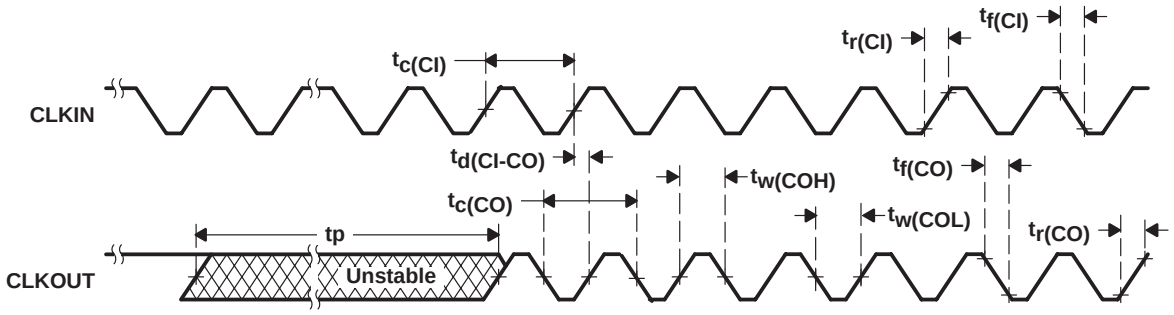


Figure 11. External Multiply-By-One Clock

bypass option

An external frequency can be used by injecting the frequency directly into CLKIN.

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 11 and the recommended operating conditions table)

PARAMETER		MIN	TYP	MAX	UNIT
$t_{c(CO)}$	Cycle time, CLKOUT	10	$t_{c(CI)}$		ns
$t_{d(ClH-CO)}$	Delay time, CLKIN high/low to CLKOUT high/low	4	10	16	ns
$t_{f(CO)}$	Fall time, CLKOUT		2		ns
$t_{r(CO)}$	Rise time, CLKOUT		2		ns
$t_{w(COL)}$	Pulse duration, CLKOUT low	H – 2	H – 1	H	ns
$t_{w(COH)}$	Pulse duration, CLKOUT high	H – 2	H – 1	H	ns

timing requirements (see Figure 11)

		MIN	MAX	UNIT
$t_{c(CI)}$	Cycle time, CLKIN	10	†	ns
$t_{f(CI)}$	Fall time, CLKIN		8	ns
$t_{r(CI)}$	Rise time, CLKIN		8	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.

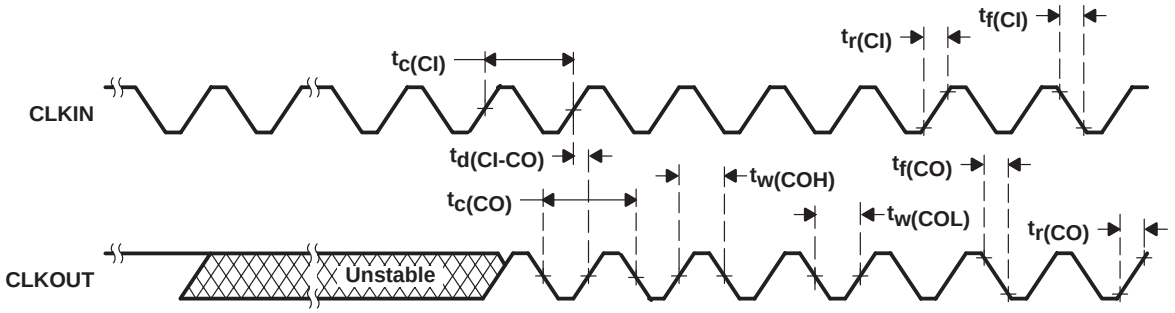


Figure 12. Timing Diagram for Bypass Mode

external memory interface timing for one wait state

switching characteristics over recommended operating conditions for a one-wait-state memory read ($\overline{\text{MSTRB}}$ = 0)† (see Figure 13)

PARAMETER		MIN	MAX	UNIT
$t_{d(CLKL-A)}$	Delay time, CLKOUT low to address valid‡	–1	5	ns
$t_{d(CLKH-A)}$	Delay time, CLKOUT high (transition) to address valid§	–1	6	ns
$t_{d(CLKL-MSL)}$	Delay time, CLKOUT low to $\overline{\text{MSTRB}}$ low	–1	4	ns
$t_{d(CLKL-MSH)}$	Delay time, CLKOUT low to $\overline{\text{MSTRB}}$ high	–1	4	ns
$t_{h(CLKL-A)R}$	Hold time, address valid after CLKOUT low‡	–1	5	ns
$t_{h(CLKH-A)R}$	Hold time, address valid after CLKOUT high§	–1	6	ns

† Address, $\overline{\text{PS}}$, and $\overline{\text{DS}}$ timings are all included in timings referenced as address.
‡ In the case of a memory read preceded by a memory read
§ In the case of a memory read preceded by a memory write

timing requirements for a one-wait-state memory read ($\overline{\text{MSTRB}}$ = 0) [H = 0.5 $t_{c(CO)}$]† (see Figure 13)

		MIN	MAX	UNIT
$t_{a(A)M}$	Access time, read data access from address valid (1 wait state required)		4H–15	ns
$t_{a(MSTRBL)}$	Access time, read data access from $\overline{\text{MSTRB}}$ low		4H–14	ns
$t_{su(D)R}$	Setup time, read data before CLKOUT low	12		ns
$t_{h(D)R}$	Hold time, read data after CLKOUT low	0		ns
$t_{h(A-D)R}$	Hold time, read data after address invalid	0		ns
$t_{h(D)MSTRBH}$	Hold time, read data after $\overline{\text{MSTRB}}$ high	0		ns

† Address, $\overline{\text{PS}}$, and $\overline{\text{DS}}$ timings are all included in timings referenced as address.

external memory interface timing for one wait state (continued)

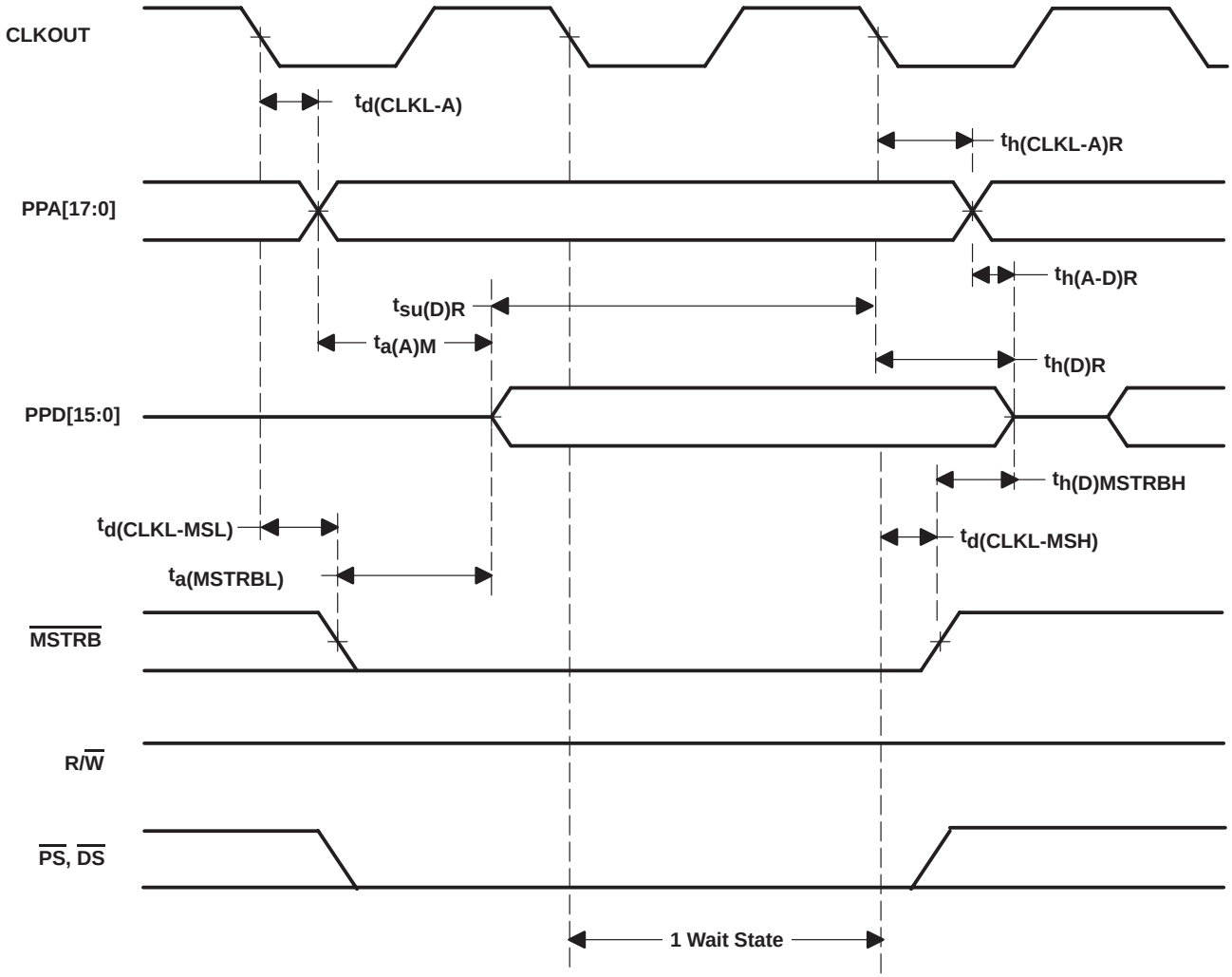


Figure 13. Memory Read ($\overline{\text{MSTRB}}$ = 0)

external memory interface timing for a memory write for one wait state

switching characteristics over recommended operating conditions for a memory write ($\overline{MSTRB}$ = 0) [H = 0.5 $t_{c(CO)}$]† (see Figure 14)

PARAMETER		MIN	MAX	UNIT
$t_{d(CLKH-A)}$	Delay time, CLKOUT high to address valid‡	– 1	6	ns
$t_{d(CLKL-A)}$	Delay time, CLKOUT low to address valid§	–1	5	ns
$t_{d(CLKL-MSL)}$	Delay time, CLKOUT low to $\overline{MSTRB}$ low	–1	4	ns
$t_{d(CLKL-D)W}$	Delay time, CLKOUT low to data valid	0	12	ns
$t_{d(CLKL-MSH)}$	Delay time, CLKOUT low to $\overline{MSTRB}$ high	– 1	4	ns
$t_{d(CLKH-RWL)}$	Delay time, CLKOUT high to R/$\overline{W}$ low	0	4	ns
$t_{d(CLKH-RWH)}$	Delay time, CLKOUT high to R/$\overline{W}$ high	–1	4	ns
$t_{d(RWL-MSTRBL)}$	Delay time, R/$\overline{W}$ low to $\overline{MSTRB}$ low	H – 2	H + 2	ns
$t_{h(A)W}$	Hold time, address valid after CLKOUT high‡	–1	6	ns

† Address, $\overline{PS}$, and $\overline{DS}$ timings are all included in timings referenced as address.
‡ 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.

timing requirements for a memory write ($\overline{MSTRB}$ = 0) [H = 0.5 $t_{c(CO)}$]† (see Figure 14)

		MIN	MAX	UNIT
$t_{h(D)MSH}$	Hold time, write data valid after $\overline{MSTRB}$ high	H – 3	H +3§	ns
$t_{w(SL)MS}$	Pulse duration, $\overline{MSTRB}$ low§	4H–4		ns
$t_{su(A)W}$	Setup time, address valid before $\overline{MSTRB}$ low	H–4		ns
$t_{su(D)MSH}$	Setup time, write data valid before $\overline{MSTRB}$ high	4H–10	4H+5§	ns

† Address, $\overline{PS}$, and $\overline{DS}$ timings are all included in timings referenced as address.
§ In the case of a memory write preceded by an I/O cycle.

external memory interface timing for a memory write for one wait state (continued)

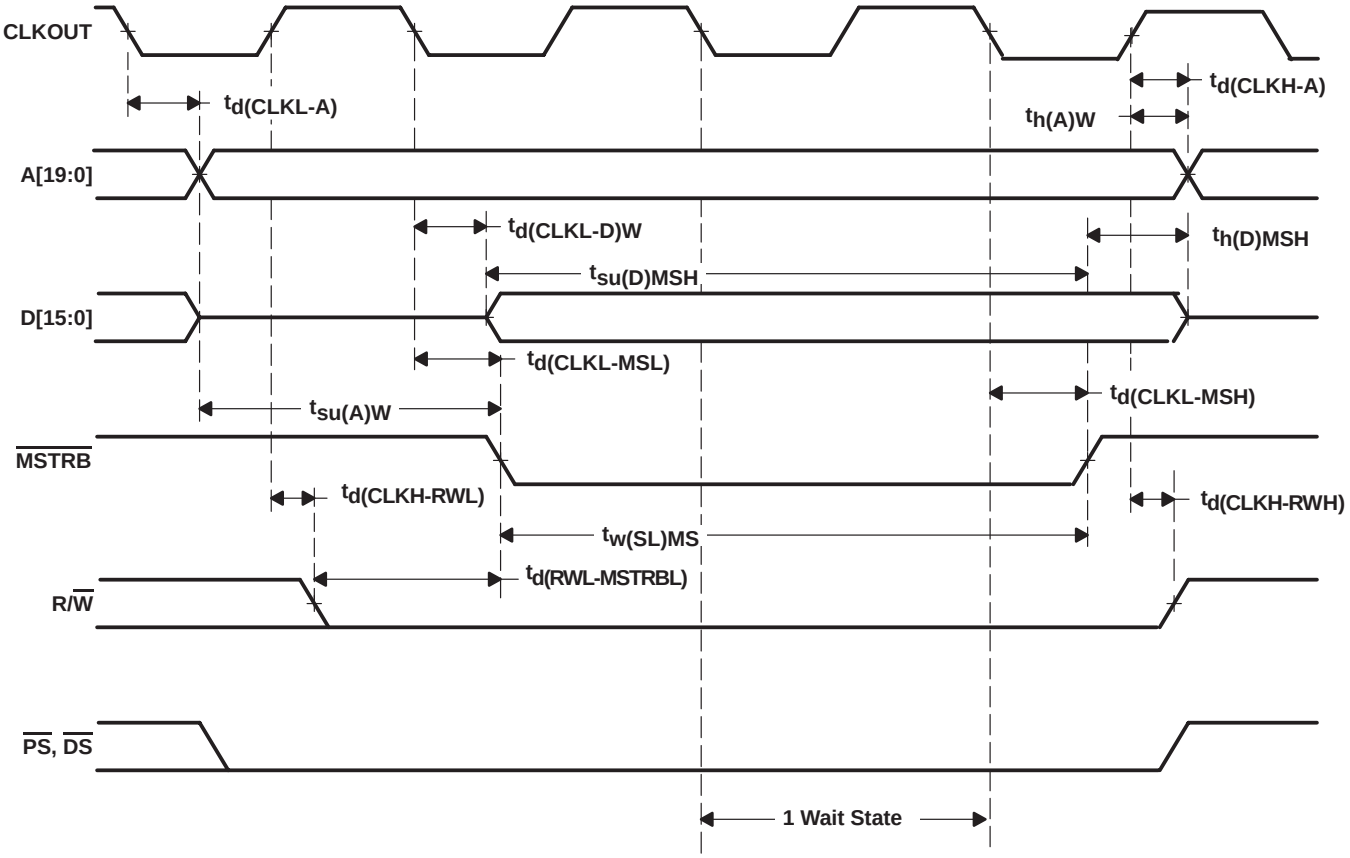


Figure 14. Memory Write ($\overline{\text{MSTRB}}$ = 0)

ready timing for externally generated wait states

timing requirements for externally generated wait states [H = 0.5 $t_{c(CO)}$]† (see Figure 15 and Figure 16)

		MIN	MAX	UNIT
$t_{su(RDY)}$	Setup time, READY before CLKOUT low	7		ns
$t_{h(RDY)}$	Hold time, READY after CLKOUT low	0		ns
$t_{v(RDY)MSTRB}$	Valid time, READY after $\overline{\text{MSTRB}}$ low‡		4H–8	ns
$t_{h(RDY)MSTRB}$	Hold time, READY after $\overline{\text{MSTRB}}$ low‡	4H		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.
‡ These timings are included for reference only. The critical timings for READY are those referenced to CLKOUT

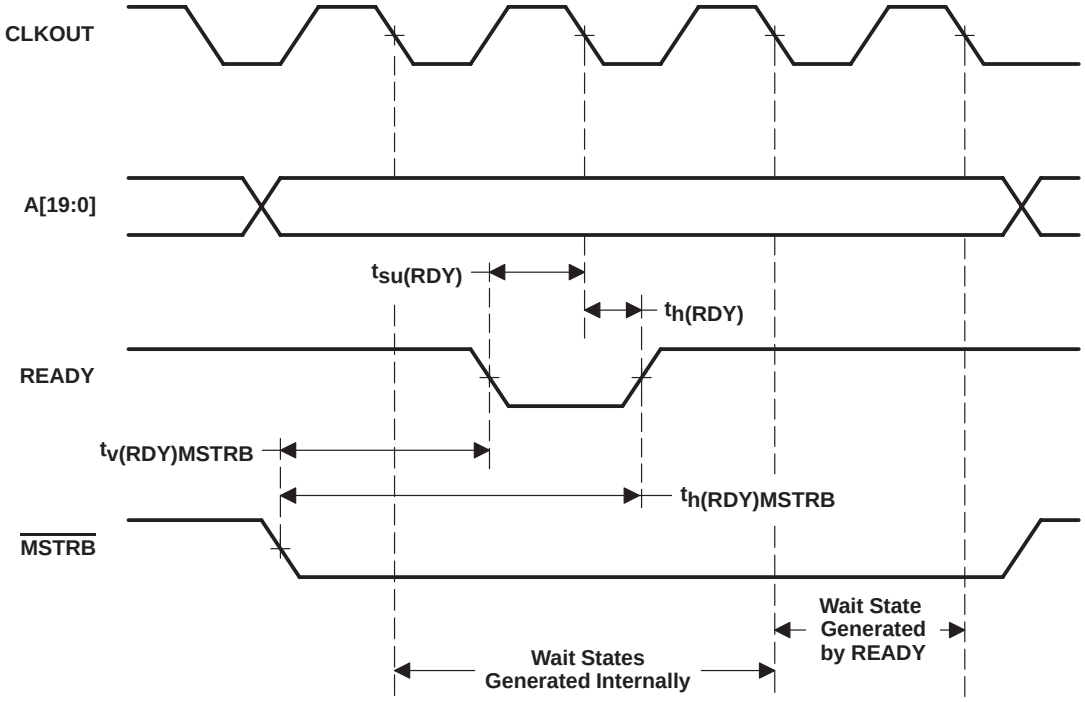


Figure 15. Memory Read With Externally Generated Wait States

ready timing for externally generated wait states (continued)

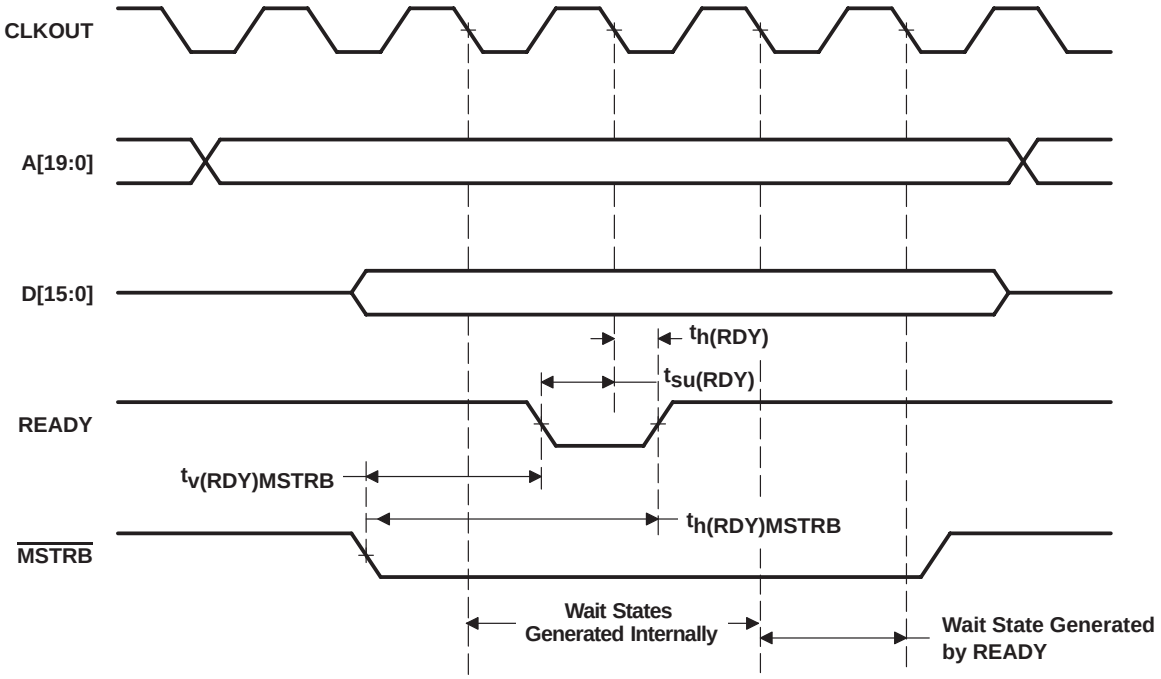


Figure 16. Memory Write With Externally Generated Wait States

parallel I/O interface timing

switching characteristics over recommended operating conditions for a parallel I/O port read ($\overline{IOSTRB}$ = 0)† (see Figure 17)

PARAMETER		MIN	MAX	UNIT
$t_{d(CLKL-A)}$	Delay time, CLKOUT low to address valid	–1	5	ns
$t_{d(CLKH-ISTRBL)}$	Delay time, CLKOUT high to $\overline{IOSTRB}$ low	0	5	ns
$t_{d(CLKH-ISTRBH)}$	Delay time, CLKOUT high to $\overline{IOSTRB}$ high	0	5	ns
$t_{h(A)IOR}$	Hold time, address after CLKOUT low	–1	5	ns

† Address and $\overline{IS}$ timings are included in timings referenced as address.

timing requirements for a parallel I/O port read ($\overline{IOSTRB}$ = 0) [H = 0.5 $t_{c(CO)}$]† (see Figure 17)

		MIN	MAX	UNIT
$t_{a(A)IO}$	Access time, read data access from address valid		5H–15	ns
$t_{a(ISTRBL)IO}$	Access time, read data access from $\overline{IOSTRB}$ low		4H–14	ns
$t_{su(D)IOR}$	Setup time, read data before CLKOUT high	10		ns
$t_{h(D)IOR}$	Hold time, read data after CLKOUT high	0		ns
$t_{h(ISTRBH-D)R}$	Hold time, read data after $\overline{IOSTRB}$ high	0		ns

† Address and $\overline{IS}$ timings are included in timings referenced as address.

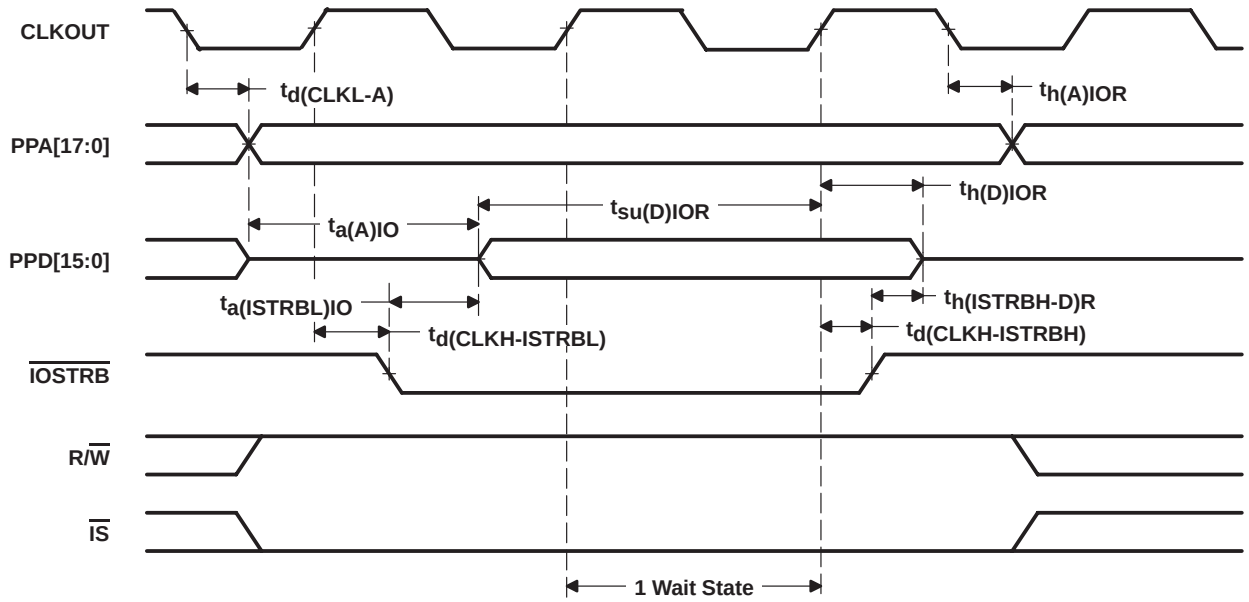


Figure 17. Parallel I/O Port Read ($\overline{IOSTRB}$=0)

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 18)

PARAMETER		MIN	MAX	UNIT
$t_{d(CLKL-A)}$	Delay time, CLKOUT low to address valid	–1	5	ns
$t_{d(CLKH-ISTRBL)}$	Delay time, CLKOUT high to $\overline{IOSTRB}$ low	0	5	ns
$t_{d(CLKH-D)IOW}$	Delay time, CLKOUT high to write data valid	H–5	H+11	ns
$t_{d(CLKH-ISTRBH)}$	Delay time,CLKOUT high to $\overline{IOSTRB}$ high	0	5	ns
$t_{d(CLKL-RWL)}$	Delay time, CLKOUT low to R/$\overline{W}$ low	0	4	ns
$t_{d(CLKL-RWH)}$	Delay time, CLKOUT low to R/$\overline{W}$ high	0	4	ns
$t_{h(A)IOW}$	Hold time, address valid after CLKOUT low	–1	5	ns
$t_{h(D)IOW}$	Hold time, write data after $\overline{IOSTRB}$ high	H–3	H+5	ns
$t_{su(D)IOSTRBH}$	Setup time, write data before $\overline{IOSTRB}$ high	3H–9	3H+5	ns
$t_{su(A)IOSTRBL}$	Setup time, address valid before $\overline{IOSTRB}$ low	H–3	H+3	ns

† Address and $\overline{IS}$ timings are included in timings referenced as address.

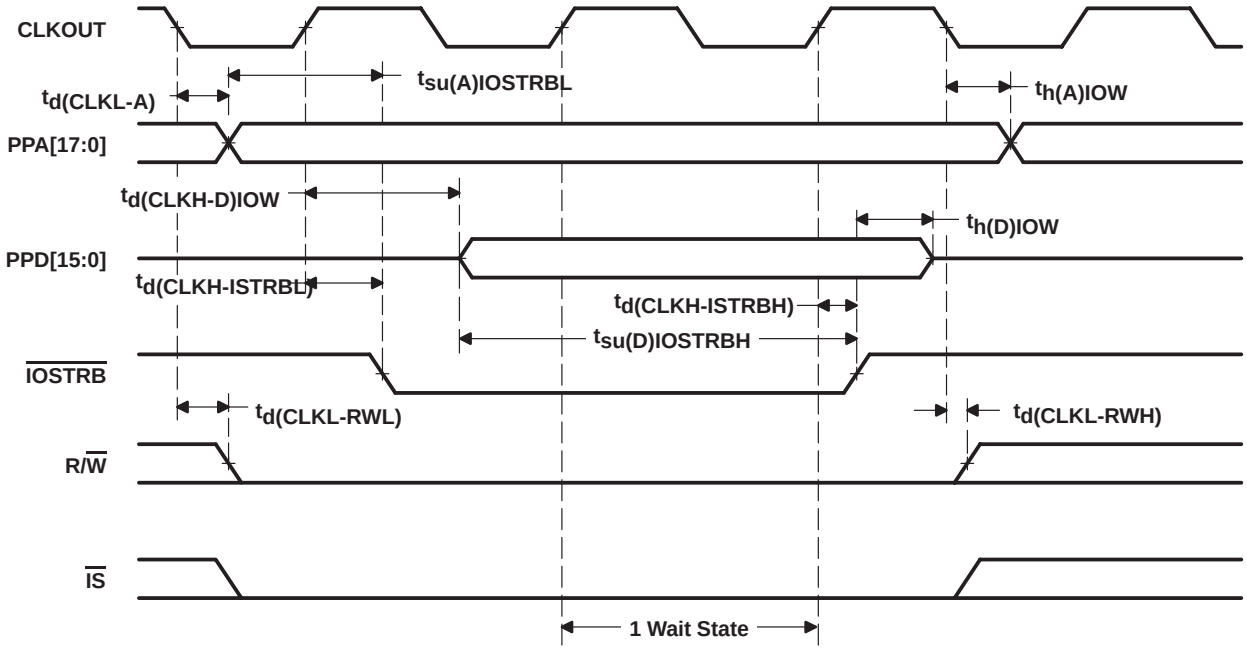


Figure 18. Parallel I/O Port Write ($\overline{IOSTRB}$=0)

ready timing for externally generated wait states

timing requirements for externally generated wait states [H = 0.5 $t_{c(CO)}$]† (see Figure 19 and Figure 20)

		MIN	MAX	UNIT
$t_{su(RDY)}$	Setup time, READY before CLKOUT low	7		ns
$t_{h(RDY)}$	Hold time, READY after CLKOUT low	0		ns
$t_{v(RDY)IOSTRB}$	Valid time, READY after $\overline{IOSTRB}$ low‡		5H–8	ns
$t_{h(RDY)IOSTRB}$	Hold time, READY after $\overline{IOSTRB}$ low‡	5H		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 using READY, at least two software wait states must be programmed.
‡ These timings are included for reference only. The critical timings for READY are those referenced to CLKOUT.

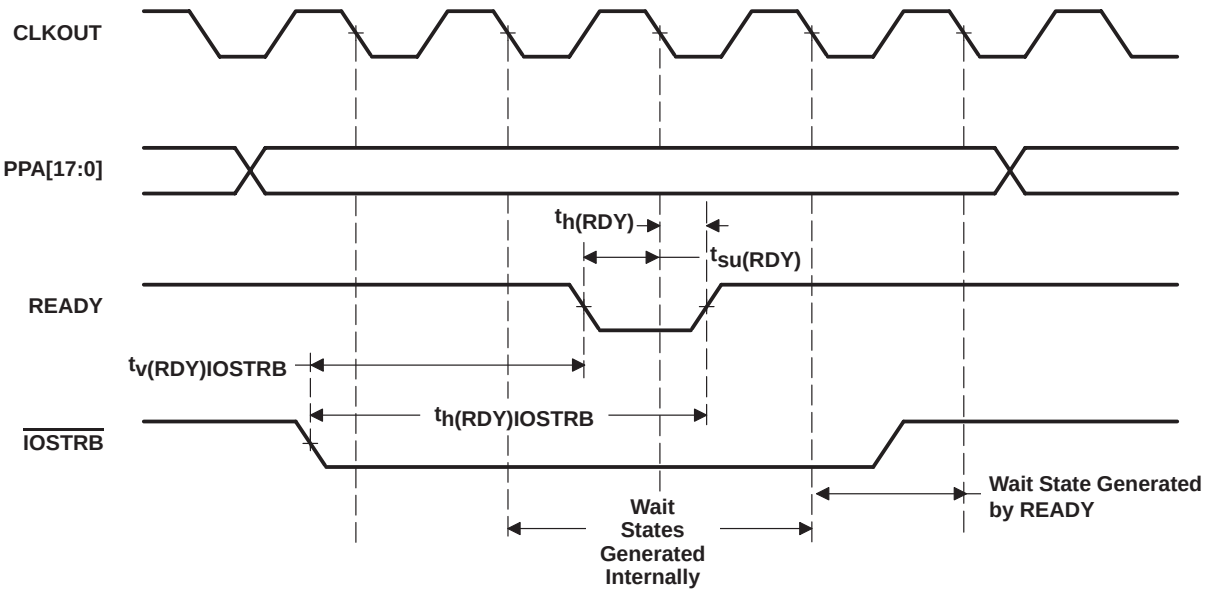


Figure 19. I/O Port Read With Externally Generated Wait States

ready timing for externally generated wait states (continued)

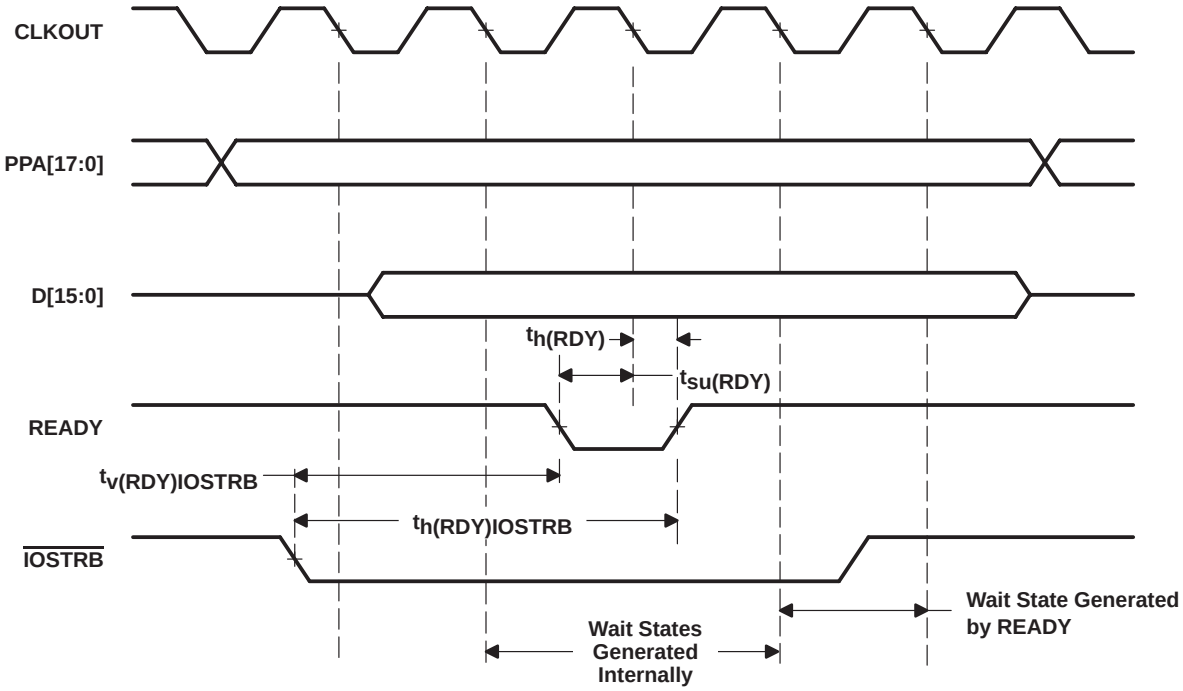


Figure 20. I/O Port Write With Externally Generated Wait States

reset, $\overline{\text{BIO}}$, interrupt, and XIO timing

timing requirements for reset, $\overline{\text{BIO}}$, interrupt, and XIO [H = 0.5 $t_{c(CO)}$] (see Figure 21, Figure 22, and Figure 23)

		MIN	MAX	UNIT
$t_{h(RS)}$	Hold time, $\overline{\text{A\_RS}}$ or $\overline{\text{B\_RS}}$ after CLKOUT low	0		ns
$t_{h(BIO)}$	Hold time, $\overline{\text{BIO}}$ after CLKOUT low	0		ns
$t_{h(INT)}$	Hold time, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$, after CLKOUT low†	0		ns
$t_{h(XIO)}$‡	Hold time, XIO after CLKOUT low	0		ns
$t_{w(RSL)}$	Pulse duration, $\overline{\text{A\_RS}}$ or $\overline{\text{B\_RS}}$ low§¶	4H+5		ns
$t_{w(BIO)A}$	Pulse duration, $\overline{\text{BIO}}$ low, asynchronous	5H		ns
$t_{w(INTH)A}$	Pulse duration, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$ high (asynchronous)†	4H		ns
$t_{w(INTL)A}$	Pulse duration, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$ low (asynchronous)†	4H		ns
$t_{w(INTL)WKP}$	Pulse duration, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$ low for IDLE2/IDLE3 wakeup†	8		ns
$t_{su(RS)}$	Setup time, $\overline{\text{A\_RS}}$ or $\overline{\text{B\_RS}}$ before CLKIN low¶	7		ns
$t_{su(BIO)}$	Setup time, $\overline{\text{BIO}}$ before CLKOUT low	9		ns
$t_{su(INT)}$	Setup time, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$, $\overline{\text{A\_RS}}$ or $\overline{\text{B\_RS}}$ before CLKOUT low†	9		ns
$t_{su(XIO)}$‡	Setup time, XIO before CLKOUT low	10		ns

† The external interrupts ($\overline{\text{INT0}}$–$\overline{\text{INT1}}$, $\overline{\text{NMI}}$) 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 a three-CLKOUT sampling sequence.

‡ Once the setup and hold times are met for XIO, the following falling edge of CLKOUT is either an HPI or EMIF cycle.

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

¶ Note that $\overline{\text{A\_RS}}$ can cause a change in clock frequency, therefore changing the value of H (see the *software-programmable phase-locked loop (PLL)* section).

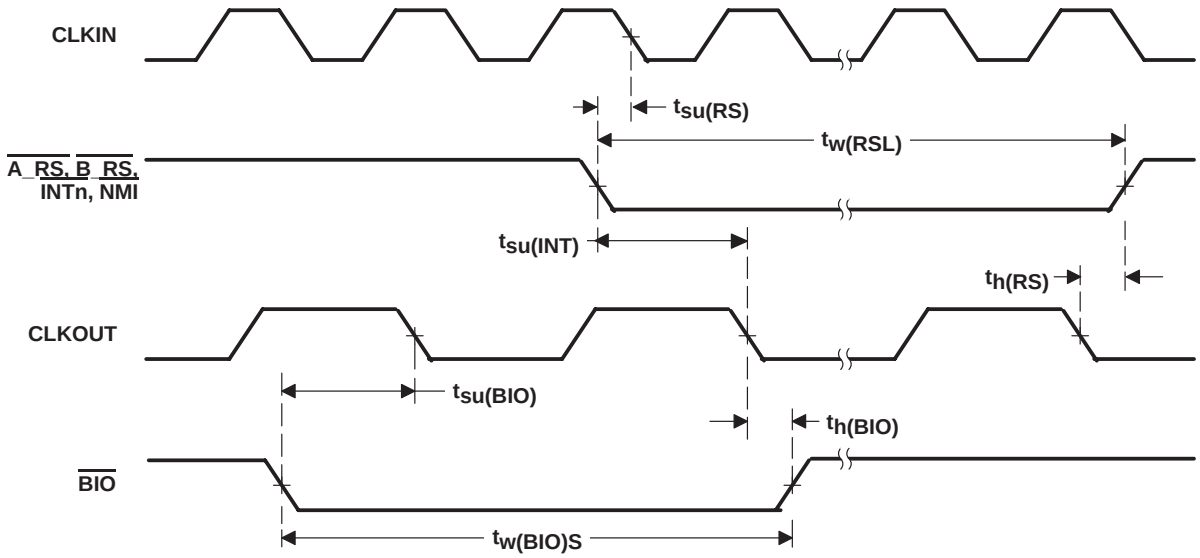


Figure 21. Reset and $\overline{\text{BIO}}$ Timings

reset, $\overline{\text{BIO}}$, interrupt, and XIO timing (continued)

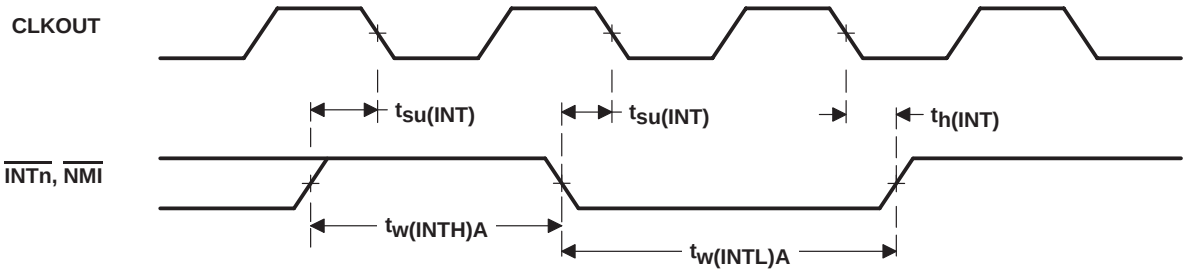


Figure 22. Interrupt Timing

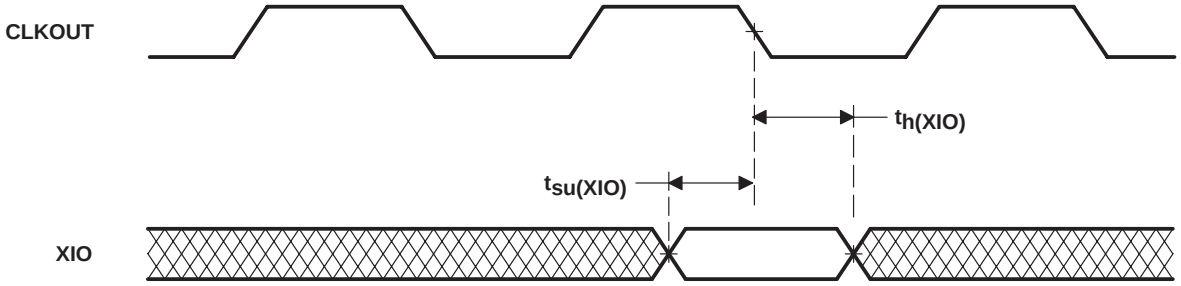


Figure 23. XIO Timing

external flag (XF), and timer output (TOUT) timing

switching characteristics over recommended operating conditions for external flag (XF) and TOUT [H = 0.5 $t_{c(CO)}$] (see Figure 24 and Figure 25)

PARAMETER		MIN	MAX	UNIT
$t_{d(XF)}$	Delay time, CLKOUT low to XF high	0	3	ns
	Delay time, CLKOUT low to XF low	0	3	
$t_{d(TOUTH)}$	Delay time, CLKOUT high to TOUT high	0	5	ns
$t_{d(TOUTL)}$	Delay time, CLKOUT high to TOUT low	0	5	ns
$t_{w(TOUT)}$	Pulse duration, TOUT	2H–2		ns

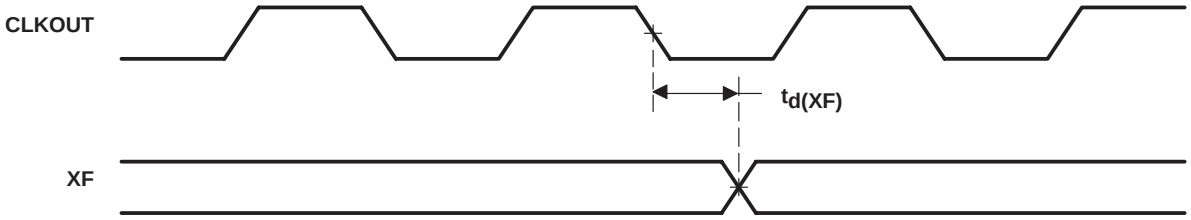


Figure 24. External Flag (XF) Timing

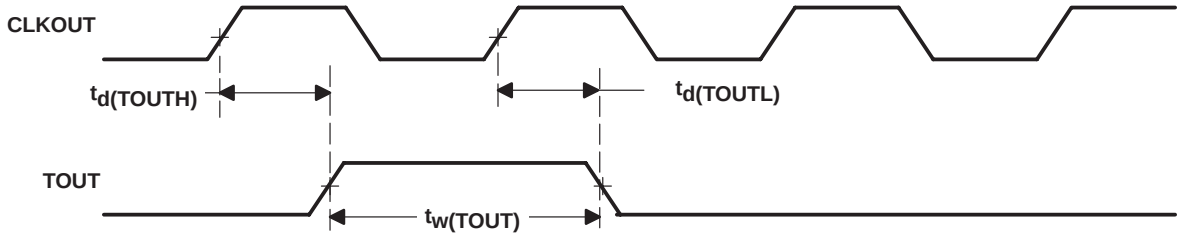


Figure 25. Timer (TOUT) Timing

general purpose input output (GPIO) timing

timing requirements for GPIO (see Figure 26)

		MIN	MAX	UNIT
$t_{su(GPIO-COH)}$	Setup time, GPIOx input valid before CLKOUT high, GPIOx configured as general-purpose input.	7		ns
$t_{h(GPIO-COH)}$	Hold time, GPIOx input valid after CLKOUT high, GPIOx configured as general-purpose input.	0		ns

switching characteristics for GPIO (see Figure 26)

PARAMETER		MIN	MAX	UNIT
$t_{d(COH-GPIO)}$	Delay time, CLKOUT high to GPIOx output change. GPIOx configured as general-purpose output.	0	5	ns

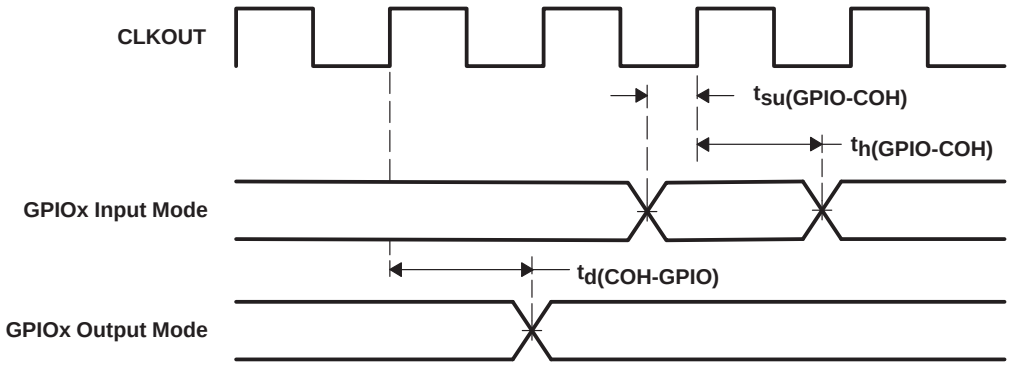


Figure 26. GPIO Timings

SELA/B timing

switching characteristics in XIO = 1 mode for SELA/B (see Figure 27)

PARAMETER		MIN	MAX	UNIT
$t_{d(SELA/B-ABUS)}$	Delay time, SELA/B to address bus valid in XIO = 1 mode.	3	10	ns

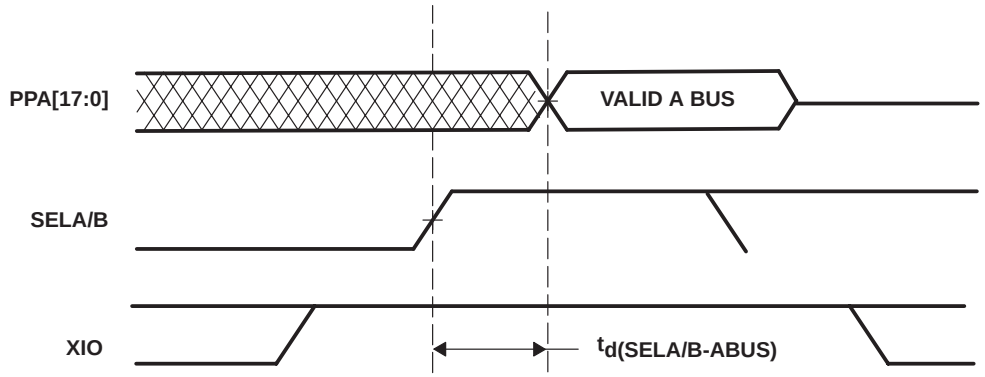


Figure 27. SELA/B Timing in XIO = 1 Mode

multichannel buffered serial port timing

timing requirements for the McBSP† [H=0.5$t_{c(CO)}$] (see Figure 28 and Figure 29)

			MIN	MAX	UNIT
$t_{c(BCKRX)}$	Cycle time, BCLKR/X	BCLKR/X ext	4H		ns
$t_{w(BCKRX)}$	Pulse duration, BCLKR/X or BCLKR/X high	BCLKR/X ext	6		ns
$t_{h(BCKRL-BFRH)}$	Hold time, external BFSR high after BCLKR low	BCLKR int	0		ns
		BCLKR ext	4		
$t_{h(BCKRL-BDRV)}$	Hold time, BDR valid after BCLKR low	BCLKR int	0		ns
		BCLKR ext	5		
$t_{h(BCKXL-BFXH)}$	Hold time, external BFSX high after BCLKX low	BCLKX int	0		ns
		BCLKX ext	4		
$t_{su(BFRH-BCKRL)}$	Setup time, external BFSR high before BCLKR low	BCLKR int	10		ns
		BCLKR ext	4		
$t_{su(BDRV-BCKRL)}$	Setup time, BDR valid before BCLKR low	BCLKR int	10		ns
		BCLKR ext	3		
$t_{su(BFXH-BCKXL)}$	Setup time, external BFSX high before BCLKX low	BCLKX int	10		ns
		BCLKX ext	6		
$t_{r(BCKRX)}$	Rise time, BCKR/X	BCLKR/X ext		8	ns
$t_{f(BCKRX)}$	Fall time, BCKR/X	BCLKR/X ext		8	ns

† Polarity bits CLKRP = CLKXP = FSRP = FSXP = 0. If the polarity of any of the signals is inverted, then the timing references of that signal are also inverted.

multichannel buffered serial port timing (continued)

switching characteristics for the McBSP† [H=0.5$t_{c(CO)}$] (see Figure 28 and Figure 29)

PARAMETER				MIN	MAX	UNIT
$t_{c(BCKRX)}$	Cycle time, BCLKR/X		BCLKR/X int	4H		ns
$t_{w(BCKRXH)}$	Pulse duration, BCLKR/X high		BCLKR/X int	D–4‡	D+1‡	ns
$t_{w(BCKRXL)}$	Pulse duration, BCLKR/X low		BCLKR/X int	C–4‡	C+1‡	ns
$t_{d(BCKRH-BFRV)}$	Delay time, BCLKR high to internal BFSR valid		BCLKR int	–3	3	ns
$t_{d(BCKXH-BFXV)}$	Delay time, BCLKX high to internal BFSX valid		BCLKX int	–3	8	ns
			BCLKX ext	2	15	
$t_{dis(BCKXH-BDXHZ)}$	Disable time, BCLKX high to BDX high impedance following last data bit		BCLKX int	–8	5	ns
			BCLKX ext	1	19	
$t_{d(BCKXH-BDXV)}$	Delay time, BCLKX high to BDX valid. This applies to all bits except the first bit transmitted.		BCLKX int	0	11	ns
			BCLKX ext	5	20	
	Delay time, BCLKX high to BDX valid.§ Only applies to first bit transmitted when in Data Delay 1 or 2 (XDATDLY=01b or 10b) modes	DXENA = 0	BCLKX int		11	
			BCLKX ext		20	
		DXENA = 1	BCLKX int		25	
			BCLKX ext		27	
$t_{e(BCKXH-BDX)}$	Enable time, BCLKX high to BDX driven.§ Only applies to first bit transmitted when in Data Delay 1 or 2 (XDATDLY=01b or 10b) modes	DXENA = 0	BCLKX int	–4		ns
			BCLKX ext	2		
		DXENA = 1	BCLKX int	6		
			BCLKX ext	12		
$t_{d(BFXH-BDXV)}$	Delay time, BFSX high to BDX valid.§ Only applies to first bit transmitted when in Data Delay 0 (XDATDLY=00b) mode.	DXENA = 0	BFSX int		9	ns
			BFSX ext		12	
		DXENA = 1	BFSX int		25	
			BFSX ext		26	
$t_{e(BFXH-BDX)}$	Enable time, BFSX high to BDX driven.§ Only applies to first bit transmitted when in Data Delay 0 (XDATDLY=00b) mode	DXENA = 0	BFSX int	–1		ns
			BFSX ext	2		
		DXENA = 1	BFSX int	9		
			BFSX ext	13		

† Polarity bits CLKRP = CLKXP = FSRP = FSXP = 0. If the polarity of any of the signals is inverted, then the timing references of that signal are also inverted.

‡ T=BCLKRX period = (1 + CLKGDV) * 2H
C=BCLKRX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * 2H when CLKGDV is even
D=BCLKRX high pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2 + 1) * 2H when CLKGDV is even

§ See the *TMS320C54x Enhanced Peripherals Reference Set, Volume 5* (literature number SPRU302) for a description of the DX enable (DXENA) and data delay features of the McBSP.

multichannel buffered serial port timing (continued)

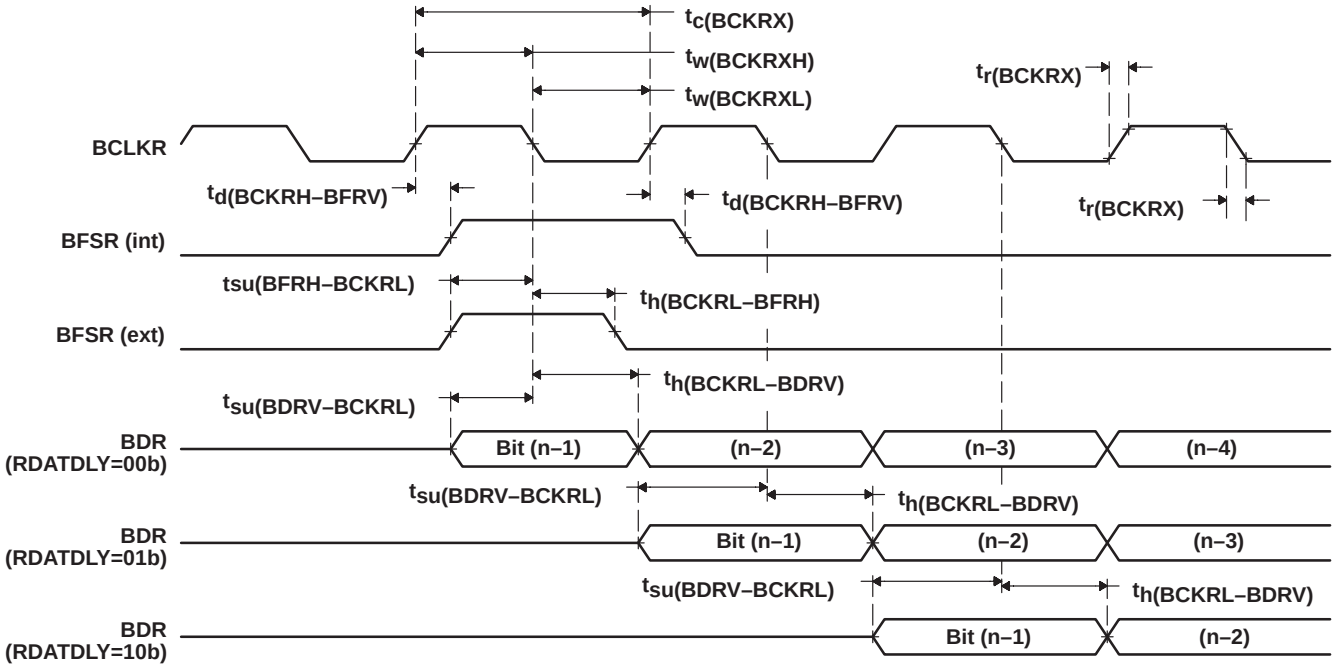


Figure 28. McBSP Receive Timings

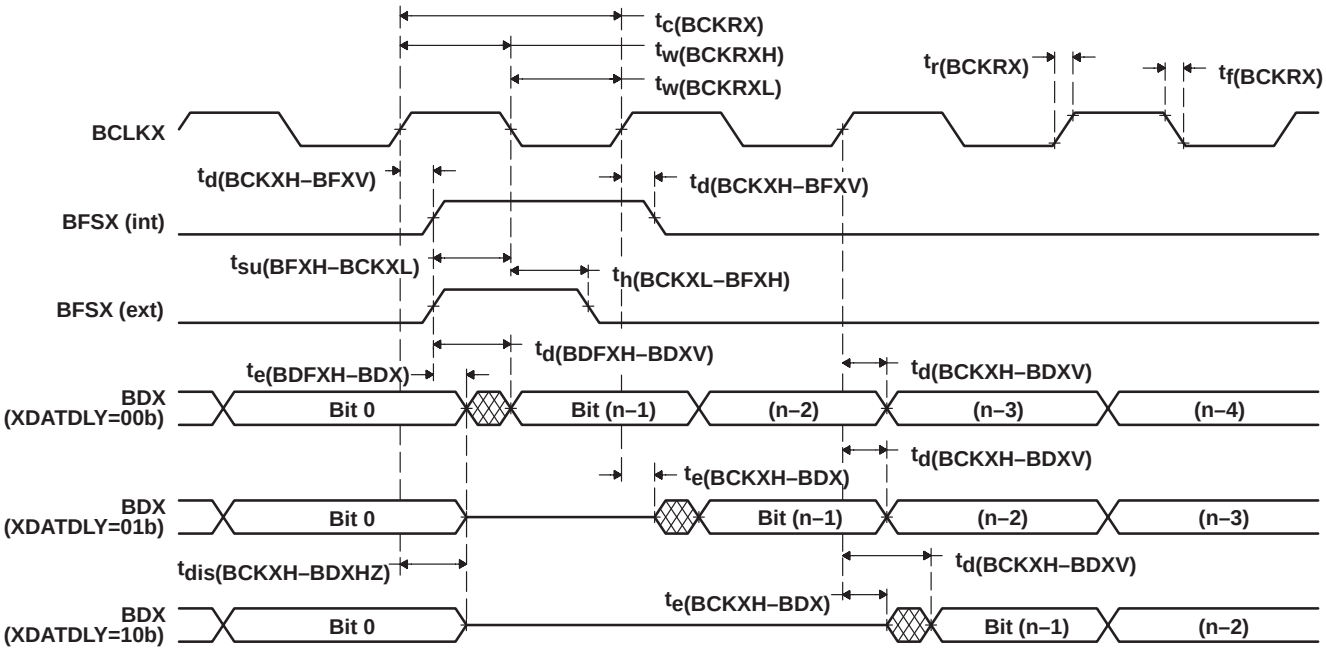


Figure 29. McBSP Transmit Timings

multichannel buffered serial port timing (continued)

timing requirements for McBSP general-purpose I/O (see Figure 30)

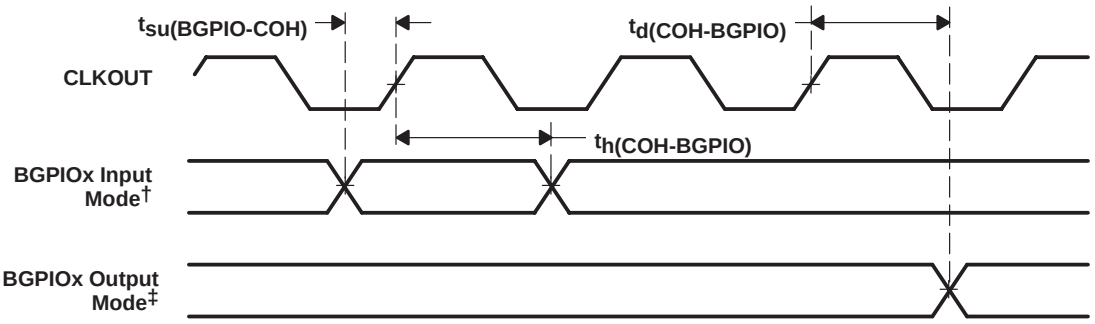
		MIN	MAX	UNIT
$t_{su(BGPIO-COH)}$	Setup time, BGPIOx input mode before CLKOUT high†	9		ns
$t_{h(COH-BGPIO)}$	Hold time, BGPIOx input mode after CLKOUT high†	0		ns

† BGPIOx refers to BCLKRx, BFSRx, BDRx, BCLKXx, or BFSXx when configured as a general-purpose input.

switching characteristics for McBSP general-purpose I/O (see Figure 30)

PARAMETER		MIN	MAX	UNIT
$t_{d(COH-BGPIO)}$	Delay time, CLKOUT high to BGPIOx output mode‡	–10	10	ns

‡ BGPIOx refers to BCLKRx, BFSRx, BCLKXx, BFSXx, or BDXx when configured as a general-purpose output.



† BGPIOx refers to BCLKRx, BFSRx, BDRx, BCLKXx, or BFSXx when configured as a general-purpose input.
‡ BGPIOx refers to BCLKRx, BFSRx, BCLKXx, BFSXx, or BDXx when configured as a general-purpose output.

Figure 30. McBSP General-Purpose I/O Timings

multichannel buffered serial port timing (continued)

timing requirements for McBSP as SPI master or slave: [H=0.5$t_{c(CO)}$] CLKSTP = 10b, CLKXP = 0† (see Figure 31)

		MASTER		SLAVE		UNIT
		MIN	MAX	MIN	MAX	
$t_{su(BDRV-BCKXL)}$	Setup time, BDR valid before BCLKX low	12		2 – 12H		ns
$t_{h(BCKXL-BDRV)}$	Hold time, BDR valid after BCLKX low	4		6 + 12H		ns
$t_{su(BFXL-BCKXH)}$	Setup time, BFSX low before BCLKX high			10		ns
$t_{c(BCKX)}$	Cycle time, BCLKX	12H		32H		ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics for McBSP as SPI master or slave: [H=0.5$t_{c(CO)}$] CLKSTP = 10b, CLKXP = 0† (see Figure 31)

PARAMETER		MASTER‡		SLAVE		UNIT
		MIN	MAX	MIN	MAX	
$t_{h(BCKXL-BFXL)}$	Hold time, BFSX low after BCLKX low§	T – 5	T + 5			ns
$t_{d(BFXL-BCKXH)}$	Delay time, BFSX low to BCLKX high¶	C – 5	C + 5			ns
$t_{d(BCKXH-BDXV)}$	Delay time, BCLKX high to BDX valid	–2	12	6H + 4	10H + 19	ns
$t_{dis(BCKXL-BDXHZ)}$	Disable time, BDX high impedance following last data bit from BCLKX low	C – 2	C +10			ns
$t_{dis(BFXH-BDXHZ)}$	Disable time, BDX high impedance following last data bit from BFSX high			4H+ 4	8H + 17	ns
$t_{d(BFXL-BDXV)}$	Delay time, BFSX low to BDX valid			4H + 4	8H + 17	ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ T = BCLKX period = (1 + CLKGDV) * 2H
C = BCLKX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * 2H when CLKGDV is even
§ FSRP = FSXP = 1. As a SPI master, BFSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on BFSX and BFSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
¶ BFSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (BCLKX).

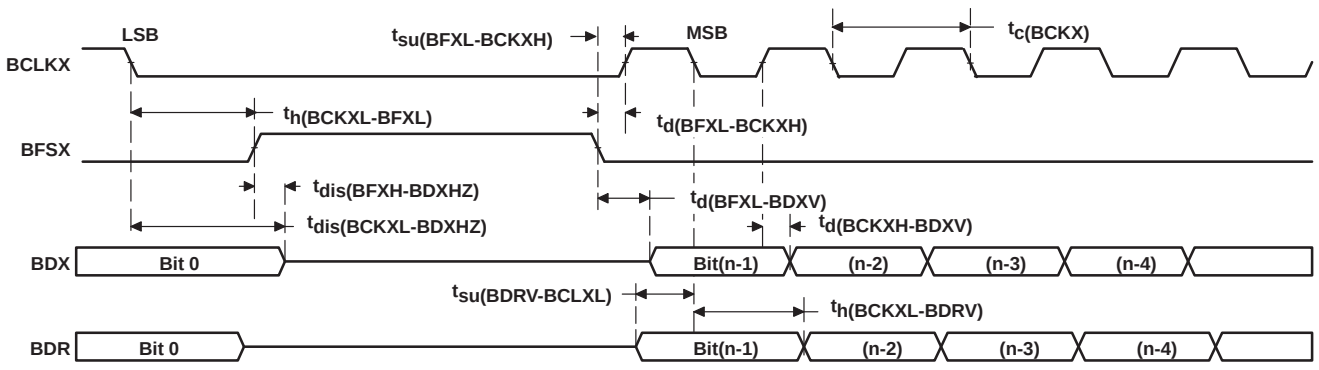


Figure 31. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

multichannel buffered serial port timing (continued)

timing requirements for McBSP as SPI master or slave: [H=0.5$t_{c(CO)}$] CLKSTP = 11b, CLKXP = 0† (see Figure 32)

		MASTER		SLAVE		UNIT
		MIN	MAX	MIN	MAX	
$t_{su(BDRV-BCKXH)}$	Setup time, BDR valid before BCLKX high	12		2 – 12H		ns
$t_{h(BCKXH-BDRV)}$	Hold time, BDR valid after BCLKX high	4		5 +12H		ns
$t_{su(BFXL-BCKXH)}$	Setup time, BFSX low before BCLKX high			10		ns
$t_{c(BCKX)}$	Cycle time, BCLKX	12H		32H		ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics for McBSP as SPI master or slave: [H=0.5$t_{c(CO)}$] CLKSTP = 11b, CLKXP = 0† (see Figure 32)

PARAMETER		MASTER‡		SLAVE		UNIT
		MIN	MAX	MIN	MAX	
$t_{h(BCKXL-BFXL)}$	Hold time, BFSX low after BCLKX low§	C – 5	C + 5			ns
$t_{d(BFXL-BCKXH)}$	Delay time, BFSX low to BCLKX high¶	T – 5	T + 5			ns
$t_{d(BCKXL-BDXV)}$	Delay time, BCLKX low to BDX valid	–2	12	6H + 4	10H + 19	ns
$t_{dis(BCKXL-BDXHZ)}$	Disable time, BDX high impedance following last data bit from BCLKX low	–2	10	6H + 4	10H + 17	ns
$t_{d(BFXL-BDXV)}$	Delay time, BFSX low to BDX valid	D – 2	D +10	4H – 4	8H + 17	ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ T = BCLKX period = (1 + CLKGDV) * 2H
C = BCLKX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * 2H when CLKGDV is even
D = BCLKX high pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2 + 1) * 2H when CLKGDV is even
§ FSRP = FSXP = 1. As a SPI master, BFSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on BFSX and BFSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
¶ BFSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (BCLKX).

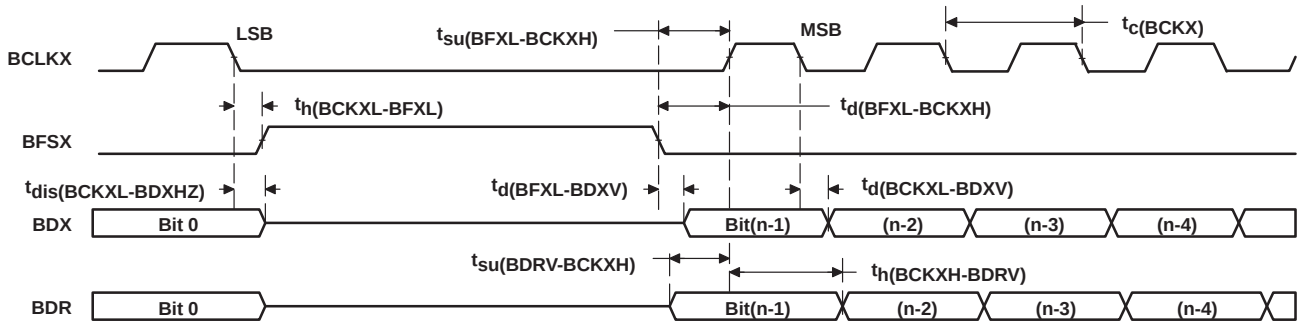


Figure 32. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

multichannel buffered serial port timing (continued)

timing requirements for McBSP as SPI master or slave: [H=0.5$t_{c(CO)}$] CLKSTP = 10b, CLKXP = 1† (see Figure 33)

		MASTER		SLAVE		UNIT
		MIN	MAX	MIN	MAX	
$t_{su(BDRV-BCKXH)}$	Setup time, BDR valid before BCLKX high	12		2 – 12H		ns
$t_{h(BCKXH-BDRV)}$	Hold time, BDR valid after BCLKX high	4		6 + 12H		ns
$t_{su(BFXL-BCKXL)}$	Setup time, BFSX low before BCLKX low			10		ns
$t_{c(BCKX)}$	Cycle time, BCLKX	12H		32H		ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics for McBSP as SPI master or slave: [H=0.5$t_{c(CO)}$] CLKSTP = 10b, CLKXP = 1†‡ (see Figure 33)

PARAMETER		MASTER		SLAVE		UNIT
		MIN	MAX	MIN	MAX	
$t_{h(BCKXH-BFXL)}$	Hold time, BFSX low after BCLKX high§	T – 5	T + 5			ns
$t_{d(BFXL-BCKXL)}$	Delay time, BFSX low to BCLKX low¶	D – 5	D + 5			ns
$t_{d(BCKXL-BDXV)}$	Delay time, BCLKX low to BDX valid	–2	12	6H + 4	10H + 19	ns
$t_{dis(BCKXH-BDXHZ)}$	Disable time, BDX high impedance following last data bit from BCLKX high	D – 2	D +10			ns
$t_{dis(BFXH-BDXHZ)}$	Disable time, BDX high impedance following last data bit from BFSX high			4H + 4	8H + 17	ns
$t_{d(BFXL-BDXV)}$	Delay time, BFSX low to BDX valid			4H – 4	8H + 17	ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ T = BCLKX period = (1 + CLKGDV) * 2H
D = BCLKX high pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2 + 1) * 2H when CLKGDV is even
§ FSRP = FSXP = 1. As a SPI master, BFSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on BFSX and BFSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
¶ BFSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (BCLKX).

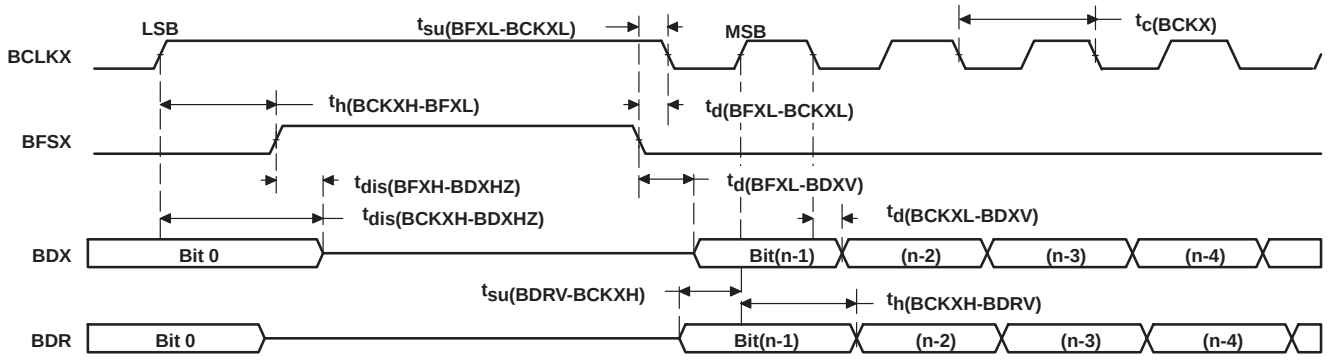


Figure 33. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

multichannel buffered serial port timing (continued)

timing requirements for McBSP as SPI master or slave: [H=0.5$t_{c(CO)}$] CLKSTP = 11b, CLKXP = 1† (see Figure 34)

		MASTER		SLAVE		UNIT
		MIN	MAX	MIN	MAX	
$t_{su(BDRV-BCKXL)}$	Setup time, BDR valid before BCLKX low	12		2 – 12H		ns
$t_{h(BCKXL-BDRV)}$	Hold time, BDR valid after BCLKX low	4		6 + 12H		ns
$t_{su(BFXL-BCKXL)}$	Setup time, BFSX low before BCLKX low			10		ns
$t_{c(BCKX)}$	Cycle time, BCLKX	12H		32H		ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics for McBSP as SPI master or slave: [H=0.5$t_{c(CO)}$] CLKSTP = 11b, CLKXP = 1†‡ (see Figure 34)

PARAMETER		MASTER‡		SLAVE		UNIT
		MIN	MAX	MIN	MAX	
$t_{h(BCKXH-BFXL)}$	Hold time, BFSX low after BCLKX high§	D – 5	D + 5			ns
$t_{d(BFXL-BCKXL)}$	Delay time, BFSX low to BCLKX low¶	T – 5	T + 5			ns
$t_{d(BCKXH-BDXV)}$	Delay time, BCLKX high to BDX valid	–2	12	6H + 4	10H + 19	ns
$t_{dis(BCKXH-BDXHZ)}$	Disable time, BDX high impedance following last data bit from BCLKX high	–2	10	6H + 4	10H + 17	ns
$t_{d(BFXL-BDXV)}$	Delay time, BFSX low to BDX valid	C – 2	C +10	4H – 4	8H + 17	ns

† For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
‡ T = BCLKX period = (1 + CLKGDV) * 2H
C = BCLKX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * 2H when CLKGDV is even
D = BCLKX high pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2 + 1) * 2H when CLKGDV is even
§ FSRP = FSXP = 1. As a SPI master, BFSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on BFSX and BFSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
¶ BFSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (BCLKX).

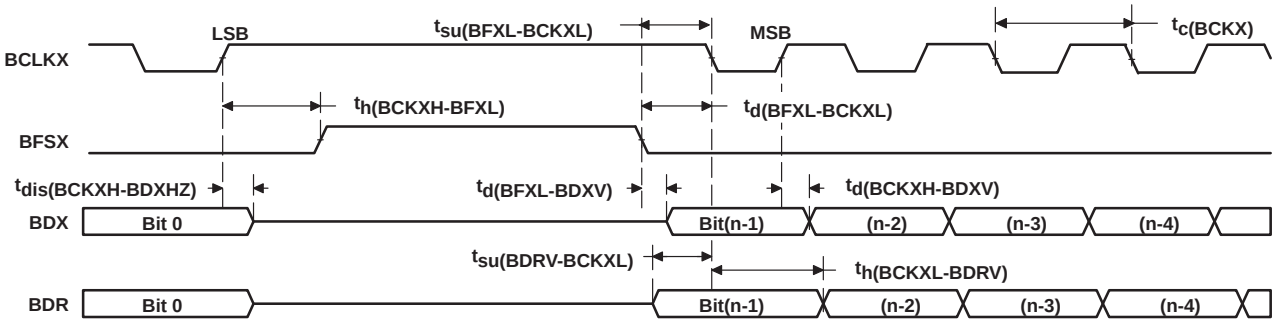


Figure 34. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

HPI16 timing

switching characteristics over recommended operating conditions[†‡§] [H = 0.5$t_{c(CO)}$] (see Figure 35 – Figure 42)

PARAMETER			MIN	MAX	UNIT
$t_{d(DSL-HDD)}$	Delay time, DS low to HD driven[§]		3	20	ns
$t_{d(DSL-HDV1)}$	Delay time, DS low to HDx valid for first byte of an HPI read	Case 1a: Memory accesses when DMAC is active in 16-bit mode and $t_{w(DSH)}$ < 18H		18H+20 – $t_{w(DSH)}$	ns
		Case 1b: Memory accesses when DMAC is active in 16-bit mode and $t_{w(DSH)}$ ≥ 18H		20	
		Case 1c: Memory access when DMAC is active in 32-bit mode and $t_{w(DSH)}$ < 26H		26H+20 – $t_{w(DSH)}$	
		Case 1d: Memory access when DMAC is active in 32-bit mode and $t_{w(DSH)}$ ≥ 26H		20	
		Case 2a: Memory accesses when DMAC is inactive and $t_{w(DSH)}$ < 10H		10H+20 – $t_{w(DSH)}$	
		Case 2b: Memory accesses when DMAC is inactive and $t_{w(DSH)}$ ≥ 10H		20	
		Case 3: Register accesses		20	
$t_{d(DSL-HDV2)}$	Multiplexed reads with autoincrement. Prefetch completed.		3	20	ns
$t_{d(DSH-HYH)}$	Delay time, DS high to HRDY high[§] (writes and autoincrement reads)	No DMA channel active		12H+5	ns
		One or more 16-bit DMA channels active		18H+5	
		One or more 32-bit DMA channels active		26H+5	
		Writes to DSPINT and $\overline{\text{HINT}}$		4H + 5	
$t_{v(HYH-HDV)}$	Valid time, HDx valid after HRDY high			7	ns
$t_{h(DSH-HDV)R}$	Hold time, HD valid after DS rising edge, read[§]		0	10	ns
$t_{d(COH-HYH)}$	Delay time, CLKOUT rising edge to HRDY high			5	ns
$t_{d(DSH-HYL)}$	Delay time, HDS or $\overline{\text{HCS}}$ high to HRDY low[‡]			12	ns
$t_{d(COH–HTX)}$	Delay time, CLKOUT rising edge to $\overline{\text{HINT}}$ change			5	ns

† HAD stands for HCNTL0, HCNTL1, and HR/$\overline{\text{W}}$.
‡ HDS refers to either $\overline{\text{HDS1}}$ or $\overline{\text{HDS2}}$.
§ DS refers to the logical OR of $\overline{\text{HCS}}$ and $\overline{\text{HDS}}$.

HPI16 timing (continued)

timing requirements [H = 0.5$t_{c(CO)}$] (see Note 1 and Figure 35 – Figure 42)

				MIN	MAX	UNIT
$t_{su(HBV-DSL)}$	Setup time, HAD valid before DS falling edge†‡			5		ns
$t_{h(DSL-HBV)}$	Hold time, HAD valid after DS falling edge†‡			5		ns
$t_{su(HBV-HSL)}$	Setup time, HAD valid before $\overline{HAS}$ falling edge†			5		ns
$t_{h(HSL-HBV)}$	Hold time, HAD valid after $\overline{HAS}$ falling edge†			5		ns
$t_{su(HAV-DSH)}$	Setup time, address valid before DS rising edge (nonmultiplexed write)			5		ns
$t_{su(HAV-DSL)}$	Setup time, address valid before DS falling edge (nonmultiplexed mode)‡			–4H – 5		ns
$t_{h(DSH-HAV)}$	Hold time, address valid after DS rising edge (nonmultiplexed mode)‡			1		ns
$t_{su(HSL-DSL)}$	Setup time, $\overline{HAS}$ low before DS falling edge‡			5		ns
$t_{h(HSL-DSL)}$	Hold time, $\overline{HAS}$ low after DS falling edge‡			2		ns
$t_{w(DSL)}$	Pulse duration, DS low‡			30		ns
$t_{w(DSH)}$	Pulse duration, DS high‡			10		ns
$t_{c(DSH-DSH)}$	Cycle time, DS rising edge to next DS rising edge‡	Nonmultiplexed or multiplexed mode (no increment) with no DMA activity.	Reads	12H		ns
			Writes	14H		ns
		Nonmultiplexed or multiplexed mode (no increment) with 16-bit DMA activity.	Reads	18H		ns
			Writes	20H		
		Nonmultiplexed or multiplexed mode (no increment) with 32-bit DMA activity.	Reads	26H		ns
			Writes	28H		
	Cycle time, DS rising edge to next DS rising edge‡ (In autoincrement mode, WRITE timings are the same as READ timings.)	Multiplexed (autoincrement) with no DMA activity.		12H		ns
		Multiplexed (autoincrement) with 16-bit DMA activity.		18H		ns
		Multiplexed (autoincrement) with 32-bit DMA activity.		26H		ns
	Cycle time, DS rising edge to next DS rising edge writes to DSPINT and $\overline{HINT}$			8H		ns
$t_{su(HDV-DSH)}$	Setup time, HD valid before DS rising edge‡			10		ns
$t_{h(DSH-HDV)W}$	Hold time, HD valid after DS rising edge, write‡			0		ns
$t_{su(SELV-DSL)}$	Setup time, SELA/B valid before DS falling edge‡			5		ns
$t_{h(DSH-SELV)}$	Hold time, SELA/B valid after DS Rising edge‡			0		ns

† HAD stands for HCNTL0, HCNTL1, and HR/$\overline{W}$.
‡ DS refers to the logical OR of $\overline{HCS}$ and $\overline{HDS}$.

HPI16 timing (continued)

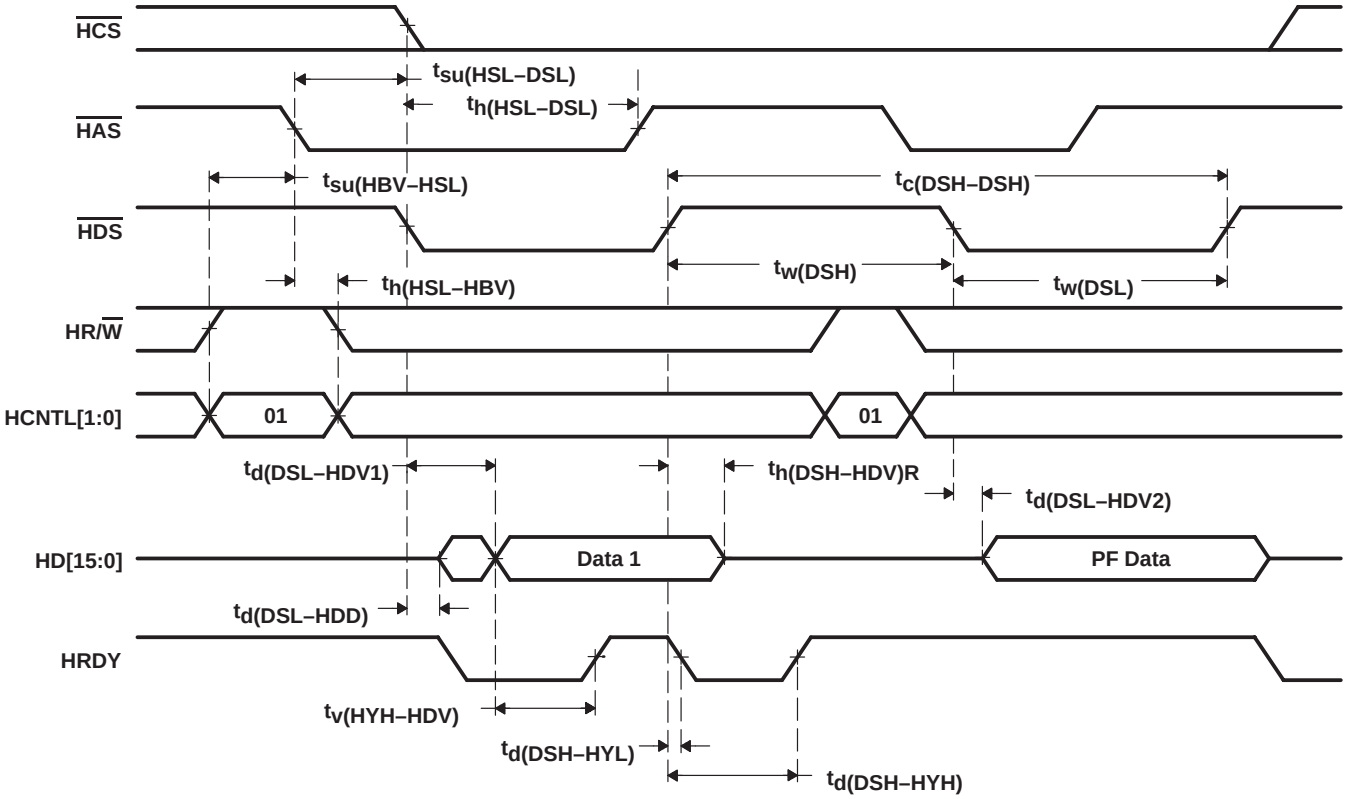


Figure 35. Multiplexed Read Timings Using $\overline{\text{HAS}}$

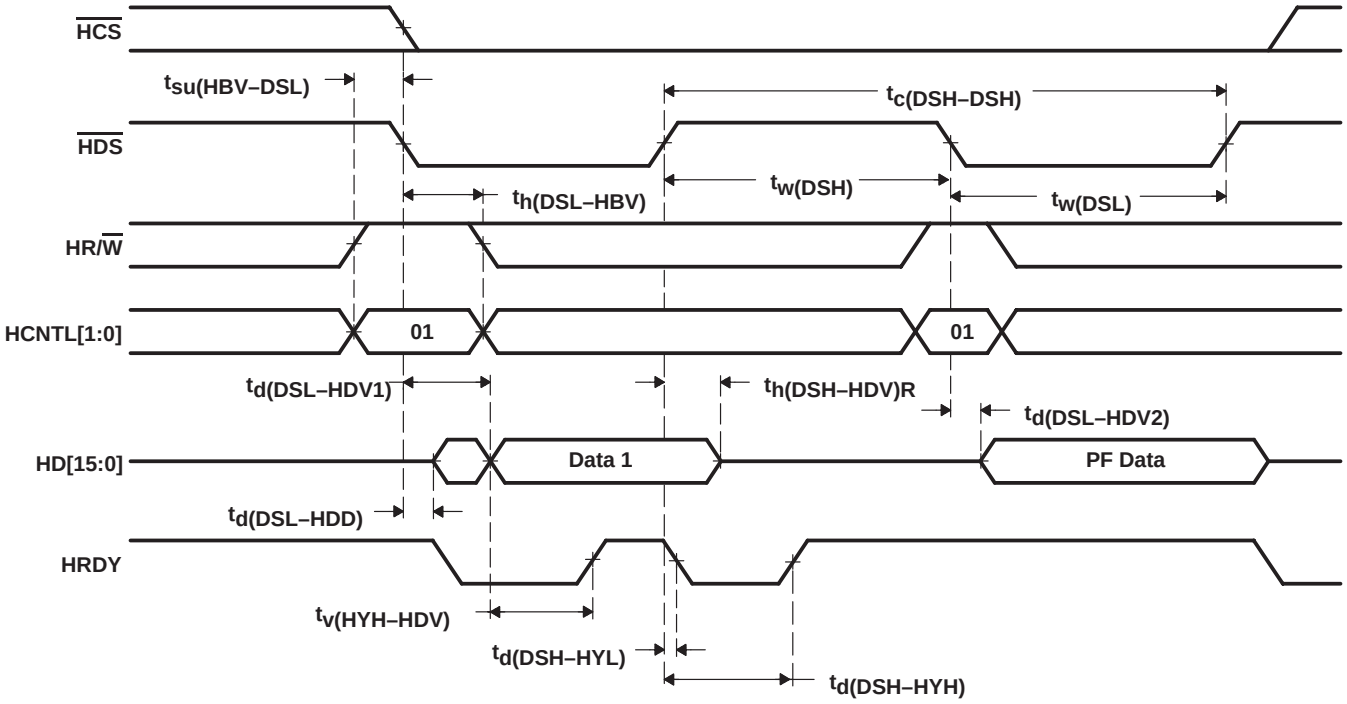


Figure 36. Multiplexed Read Timings Without $\overline{\text{HAS}}$

HPI16 timing (continued)

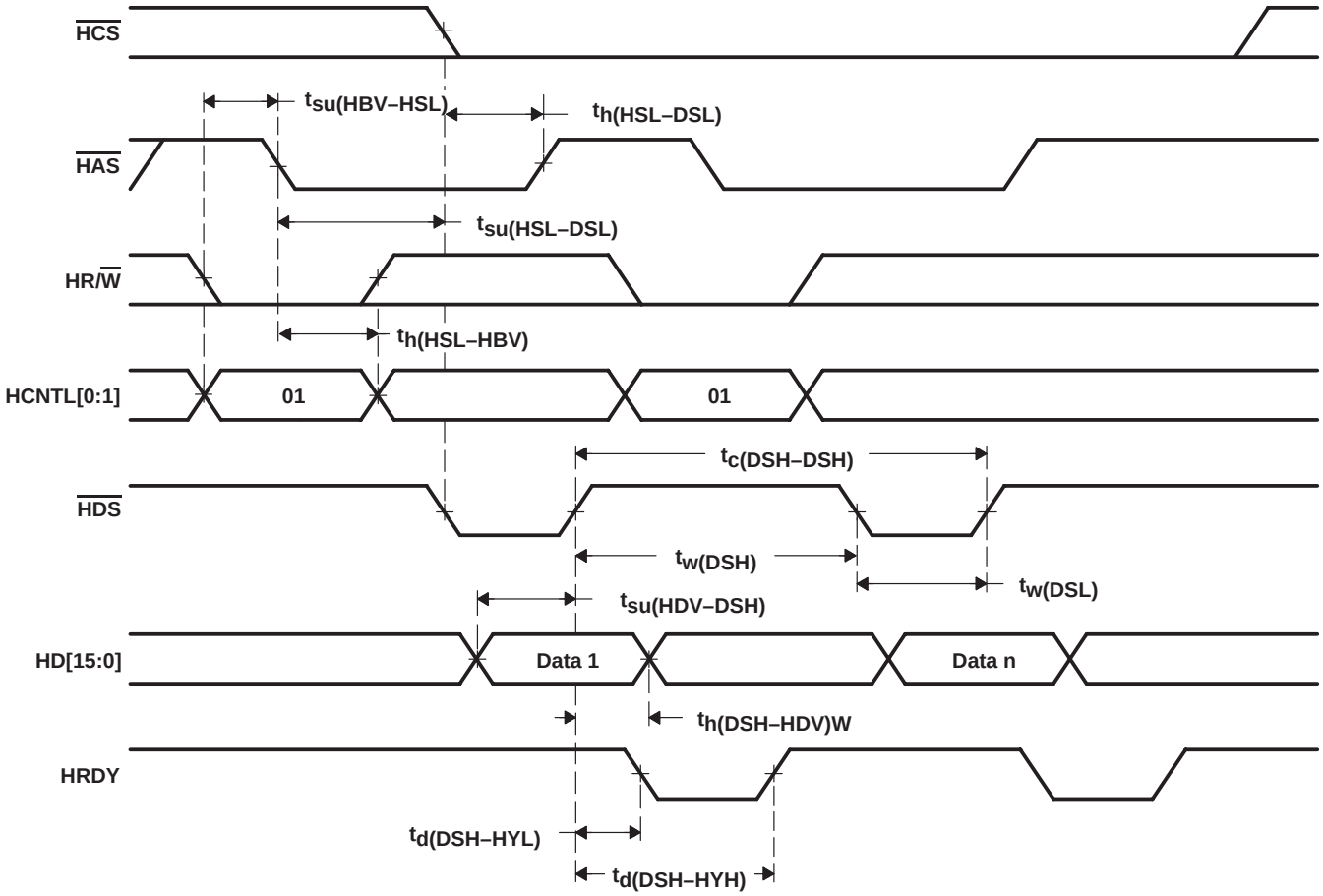


Figure 37. Multiplexed Write Timings Using $\overline{\text{HAS}}$

HPI16 timing (continued)

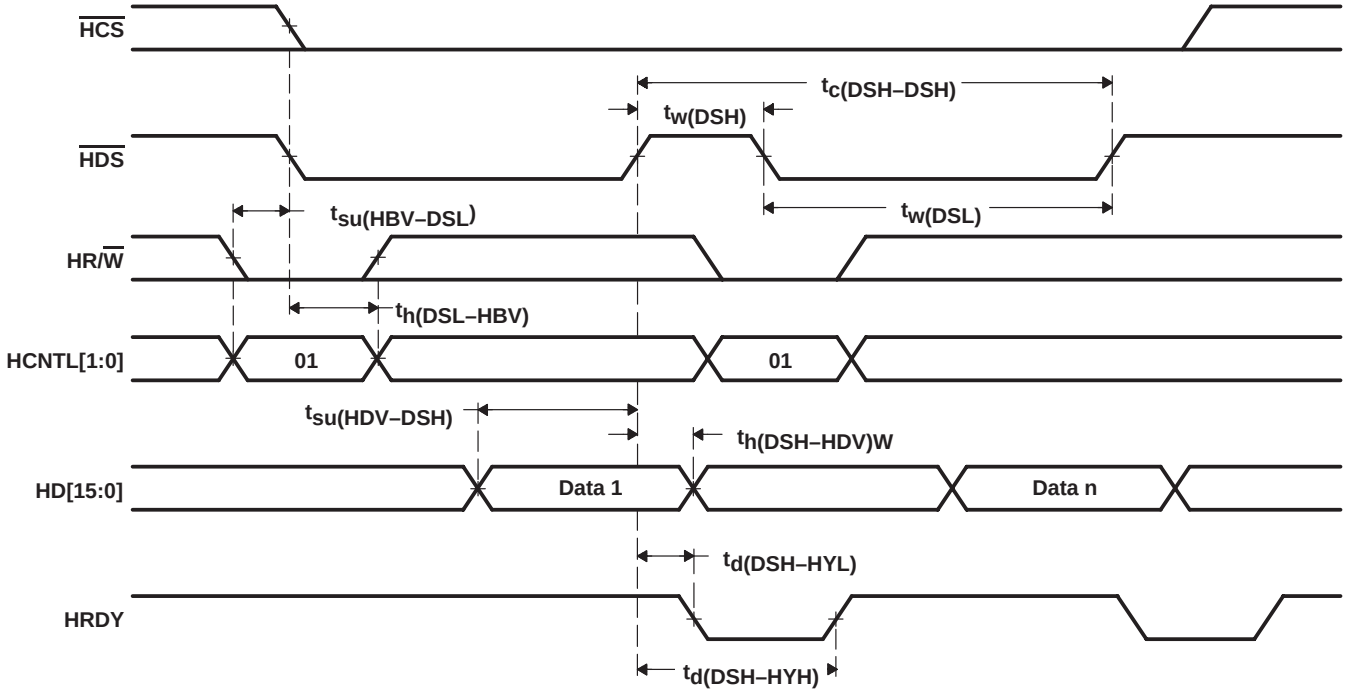


Figure 38. Multiplexed Write Timings Without $\overline{\text{HAS}}$

HPI16 timing (continued)

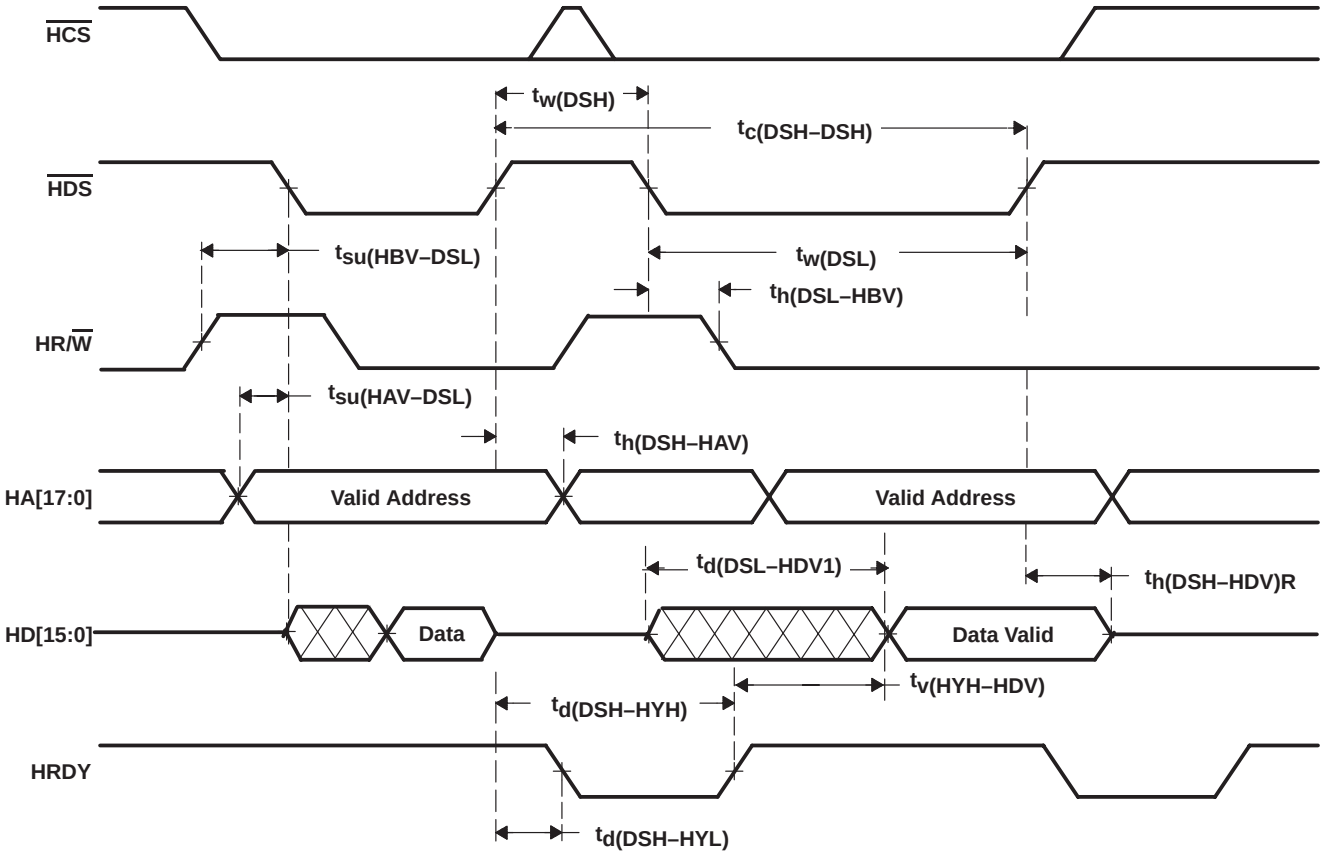


Figure 39. Nonmultiplexed Read Timings

HPI16 timing (continued)

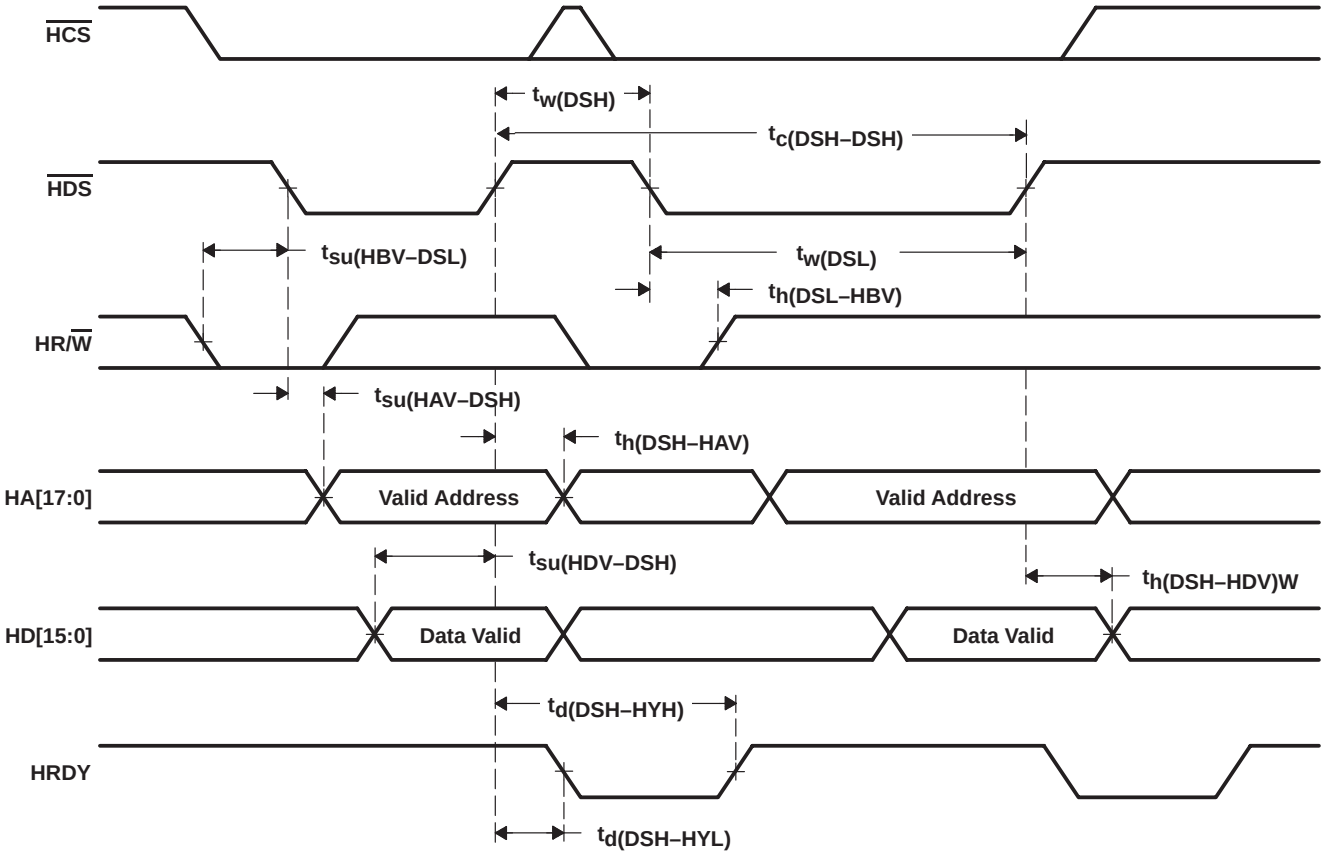


Figure 40. Nonmultiplexed Write Timings

HPI16 timing (continued)

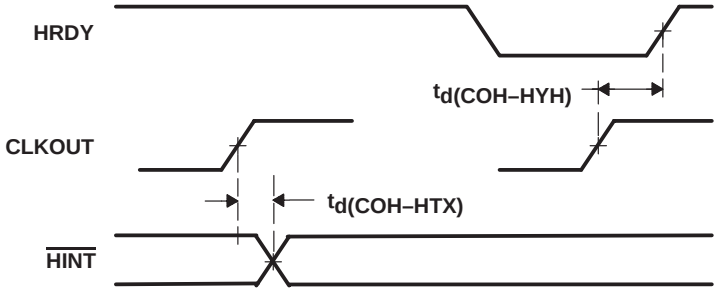


Figure 41. HRDY and HINT Relative to CLKOUT

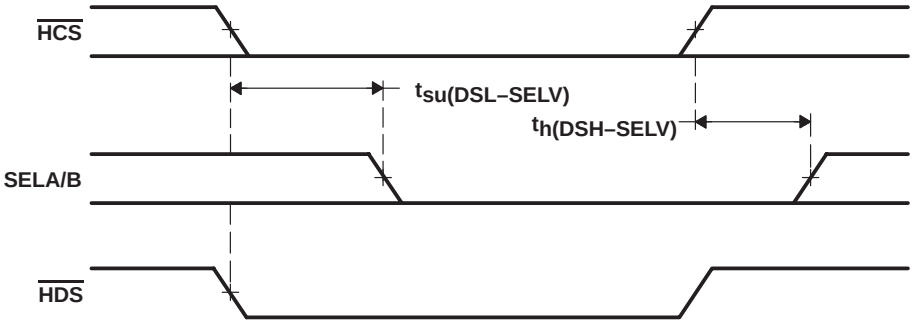
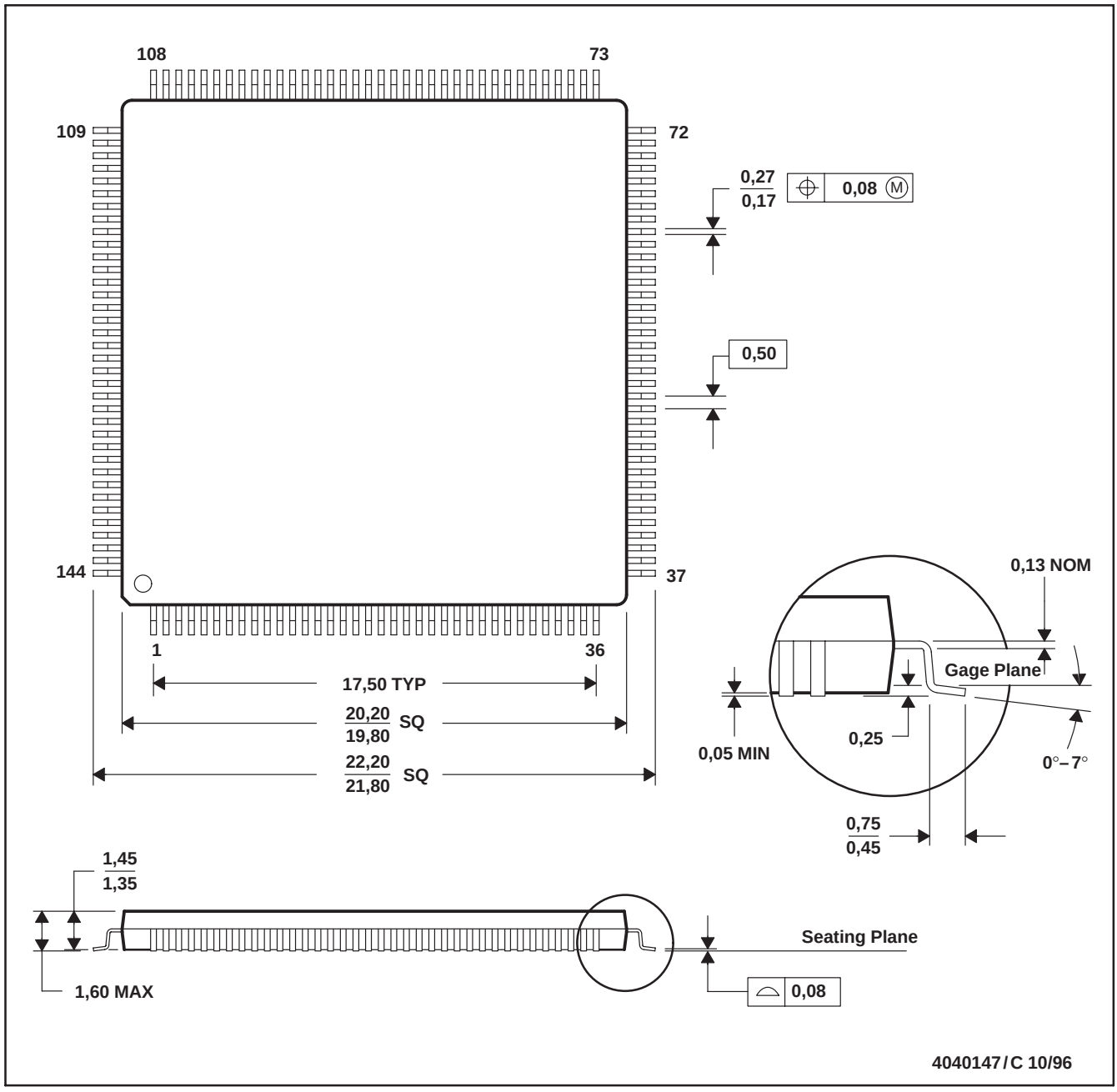


Figure 42. SELA/B Timing

MECHANICAL DATA

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



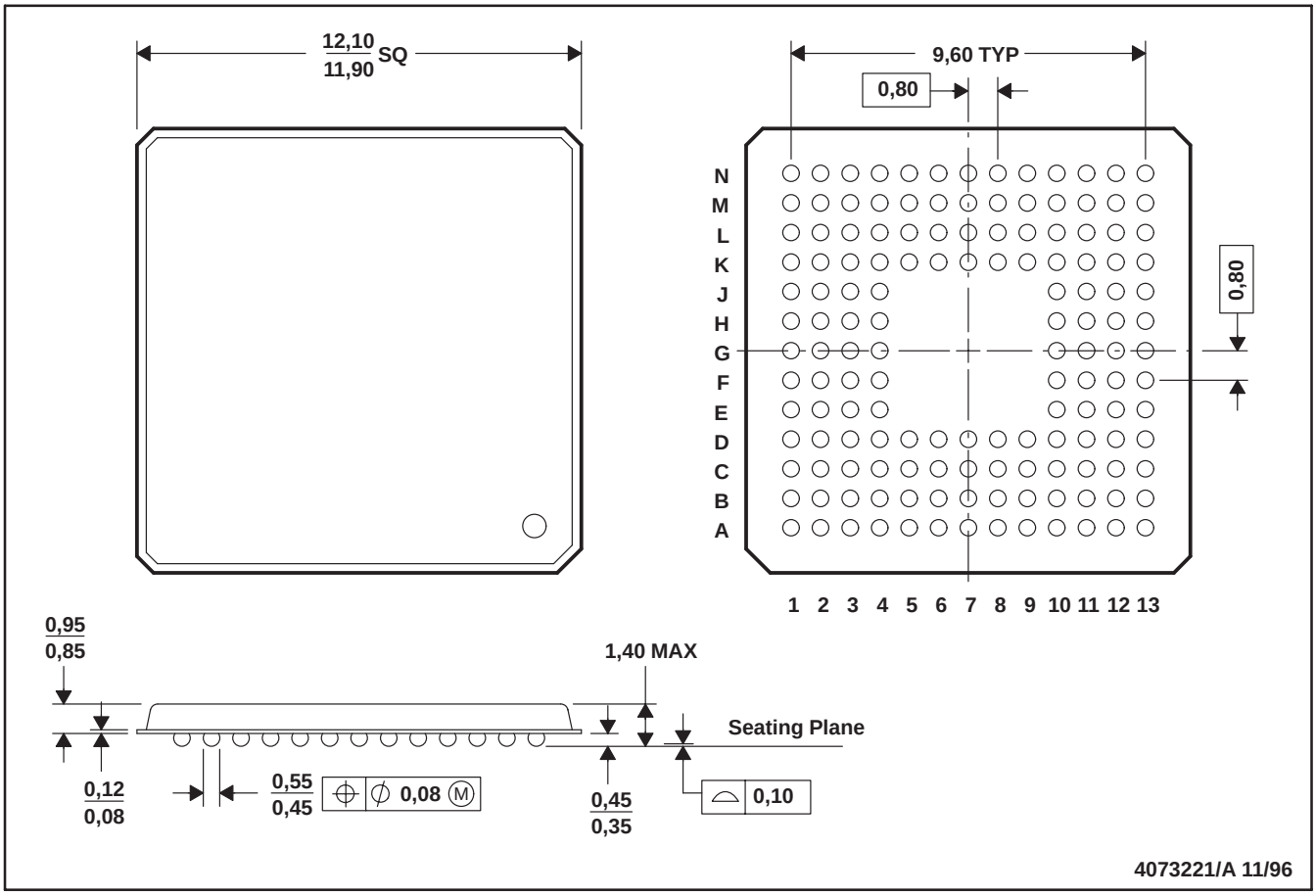
NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026

Thermal Resistance Characteristics

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

MECHANICAL DATA

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



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. MicroStar BGA™ configuration

Thermal Resistance Characteristics

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

MicroStar BGA is a trademark of Texas Instruments Incorporated.

IMPORTANT NOTICE

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

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

Customers are responsible for their applications using TI components.

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

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.