DEVICE OVERVIEW

320LC549 is very similar to 320LC548. The only architectural difference between these two is the size of the on–chip ROM: 16K words for 320LC549 and 2K words for 320LC548. 320LC549 will also use a more advanced fabrication technology called C10 to get higher DSP speed performance. The list of the main features of 320LC549 is shown below:

- **32K x 16–Bit on–chip RAM made of :**
  - **Four blocks of 2K x 16–bit on–chip dual access program/data RAM**
  - **Three blocks of 8K x 16–bit on–chip single access program/data RAM**
- **16K x 16–Bit on–chip ROM configurable to program/data memory**
- **Extended addressing mode for 4M x 16–bit maximum addressable external program space**
- **Two Buffered Serial Ports (BSPs)**
- **Time–Division Multiplexed (TDM) Serial Port**
- **8–bit parallel Host Port Interface (HPI)**
- **16–bit Timer**
- **On–chip programmable PLL**
- **On–chip scan–based emulation and JTAG boundary scan logic**
- **144–pin Thin Quad Flat Pack, 20x20x1.4mm package body dimension with 0.5mm lead pitch. Same pinout as 320LC548.**
  A 144–pin Micro–star BGA 12x12mm package is in study at that date. More information will be provided in the next revision of the specifications.
- **80 Mips Performance target at industrial temperature range for the 3.3V device (3.3V for both I/O and core power supplies).**
- **100 Mips Performance target at industrial temperature range for the dual voltage device (3.3V for I/O power supply and 2.5V for core power supply).**

SPECIFICATION CONTENTS

- **Pinout and Pin Functions**
- **Signal Description**
- **Memory**
- **Description of Peripherals**
- **Interrupts**

ADVANCE INFORMATION

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

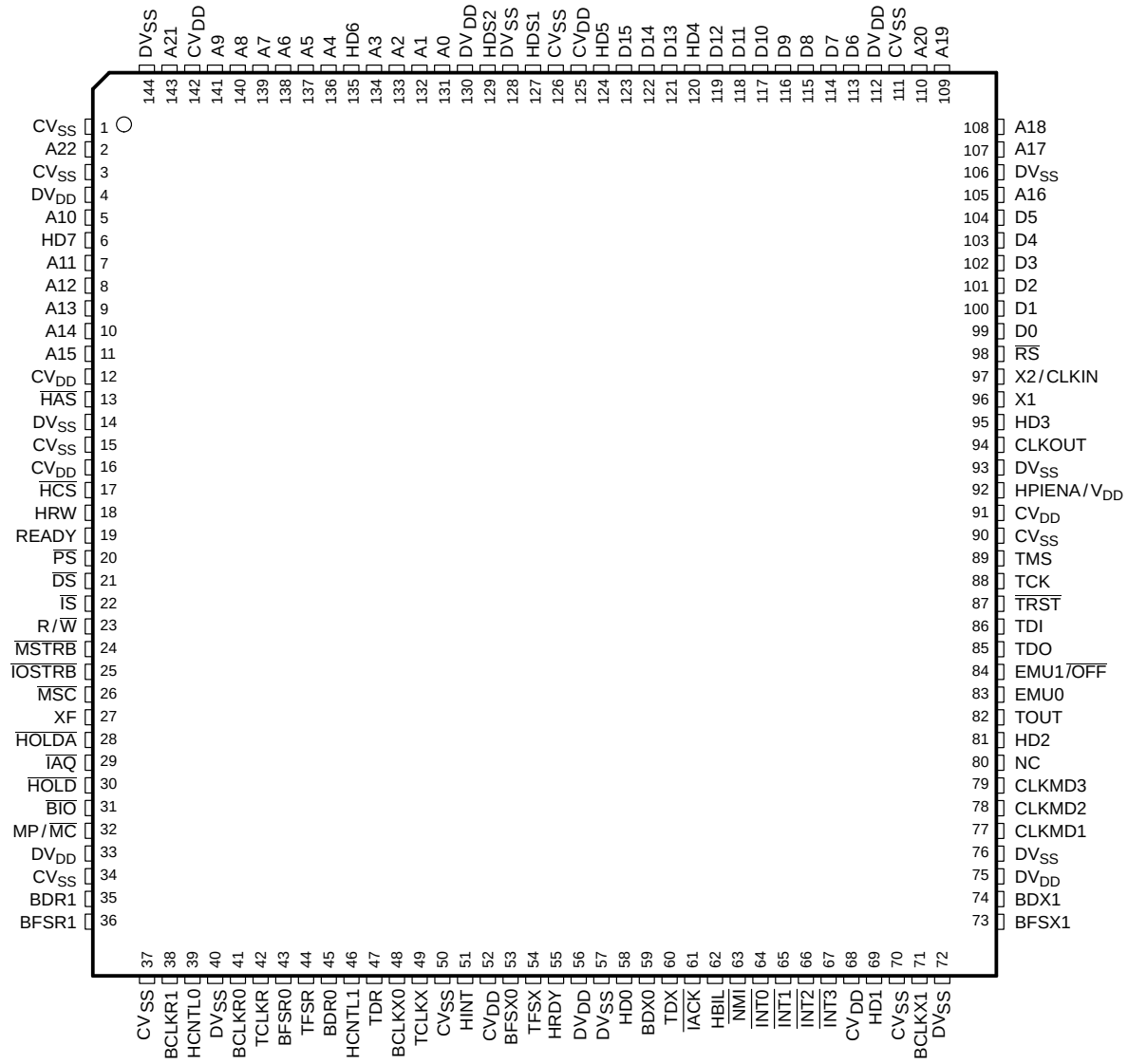
ADVANCE INFORMATION concerns new products in the sampling or preproduction phase of development. Characteristic data and other specifications are subject to change without notice.

TEXAS INSTRUMENTS

SPECIFICATION REVISION HISTORY

- **Between 1.0 and 1.1 : Add more detailed description of extended program addressing, wait–state generator, bank switching and clock generator.**
- **Between 1.1 and 1.2 : Add word mis–alignment detection feature in BSP, add description of upgraded wait state generator, add more detailed description of PLL, and mention Micro–Star–BGA package.**
- **Between 1.2 and 1.3 : eliminate the x2 upgrade in the wait state generator, clarify memory map address 0xFF00–0xFF7F, increase max BSP data rate to 50 MBit/sec.**
- **Between 1.3 and 1.4 : a) the direct switch from DIV2 to DIV4 clock mode is now supported. b) change in the clock mode table at Reset (when all CLKMD pins are set to 1, clock mode is now div 1/1 with external clock). c) the TDM serial port operates in TDM mode with external clock and external frame only.**
- **Between 1.4 and 1.5 : define two LC549 device grades, which are a) a single 3.3V power supply 80 MIPS device and b) a dual 2.5V–3.3V power supply 100 MIPS device.**
- **Between 1.5 and 1.6 : define program memory 0x018000–0x01FFFF, 0x028000–0x02FFFF, and 0x038000–0x03FFFF, as external space instead of reserved space.**

TMS320LC549 PGE PACKAGE
(TOP VIEW)



† NC = No connection

‡ DV_{SS} and DV_{DD} are power supplies for I/O pins while CV_{SS} and CV_{DD} are power supplies for core CPU.

The pin function table on the following pages lists each pin number, signal, function, and operating mode(s) for the TMS320LC549PGE (144-pin) package.

Pin Functions for the TMS320LC549PGE (144-Pin TQFP Package)

PIN NAME	PIN NO	FUNCTION†	DESCRIPTION
CV_{SS}	1	Supply	Ground
A22	2	O/Z	Parallel port address bus
CV_{SS}	3	Supply	Ground
DV_{DD}	4	Supply	$+V_{DD}$
A10	5	O/Z	Parallel port address bus
HD7	6	I/O/Z	Parallel bidirectional data bus (HPI)
A11–A15	7–11	O/Z	Parallel port address bus
CV_{DD}	12	Supply	$+V_{DD}$
$\overline{HAS}$	13	I	Address data strobe (HPI)
DV_{SS}	14	Supply	Ground
CV_{SS}	15	Supply	Ground
CV_{DD}	16	Supply	$+V_{DD}$
$\overline{HCS}$	17	I	Chip select input (HPI)
HRW	18	I	Read/Write (HPI)
READY	19	I	External access ready to complete
$\overline{PS}$	20	O/Z	Program space select
$\overline{DS}$	21	O/Z	Data space select
$\overline{IS}$	22	O/Z	I/O select
R/$\overline{W}$	23	O/Z	Read/write
$\overline{MSTRB}$	24	O/Z	External memory access strobe
$\overline{IOSTRB}$	25	O/Z	External I/O access strobe
$\overline{MSC}$	26	O/Z	Microstate complete
XF	27	O/Z	External flag
$\overline{HOLDA}$	28	O/Z	Hold acknowledge
$\overline{IAQ}$	29	O/Z	Instruction acquisition
$\overline{HOLD}$	30	I	Request access of local memory
$\overline{BIO}$	31	I	Bit I/O pin
MP/$\overline{MC}$	32	I	Microprocessor/microcomputer
DV_{DD}	33	Supply	$+V_{DD}$
CV_{SS}	34	Supply	Ground
BDR1	35	I	Serial data receive input (BSP #1)
BFSR1	36	I	Frame synchronization pulse for receive (BSP #1)
CVss	37	Supply	Ground
BCLKR1	38	I	Receive clock input (BSP #1)
HCNTL0	39	I	Control inputs (HPI)
DV_{SS}	40	Supply	Ground
BCLKR0	41	I	Receive clock input (BSP #0)
TCLKR	42	I	Receive clock input (TDM)
BFSR0	43	I	Frame synchronization pulse for receive (BSP #0)
TFSR/TADD	44	I/O	Receive frame synchronization (TDM)

† Legend:
I = Input
O = Output
Z = High-impedance

Pin Functions for the TMS320LC549PGE (144-Pin TQFP Package) (Continued)

PIN NAME	PIN NO	FUNCTION†	DESCRIPTION
BDR0	45	I	Serial data receive input (BSP #0)
HCNTL1	46	I	Control inputs (HPI)
TDR	47	I	Serial data receive input (TDM)
BCLKX0	48	I/O/Z	Serial port transmit clock (BSP #0)
TCLKX	49	I/O/Z	Serial port transmit clock (TDM)
CVss	50	Supply	Ground
HINT	51	O/Z	Interrupt output (HPI)
CV_{DD}	52	Supply	$+V_{DD}$
BFSX0	53	I/O/Z	Frame synchronization pulse for transmit (BSP #0)
TFSX/TFRM	54	I/O/Z	Transmit frame synchronization (TDM)
HRDY	55	O/Z	Ready output (HPI)
DV_{DD}	56	Supply	$+V_{DD}$
DV_{SS}	57	Supply	Ground
HD0	58	I/O/Z	Parallel bidirectional data bus (HPI)
BDX0	59	O/Z	Serial data transmit output (BSP #0)
TDX	60	O/Z	Serial data transmit output (TDM)
$\overline{IACK}$	61	O/Z	Interrupt acknowledge
HBIL	62	I	Byte identification input (HPI)
$\overline{NMI}$	63	I	Nonmaskable interrupt
$\overline{INT}0-\overline{INT}3$	64–67	I	Interrupt 0 through Interrupt 3
CV_{DD}	68	Supply	$+V_{DD}$
HD1	69	I/O/Z	Parallel bidirectional data bus (HPI)
CVss	70	Supply	Ground
BCLKX1	71	I/O/Z	Serial port transmit clock (BSP #1)
DVss	72	Supply	Ground
BFSX1	73	I/O/Z	Frame synchronization pulse for transmit (BSP #1)
BDX1	74	O/Z	Serial data transmit output (BSP #1)
DV_{DD}	75	Supply	$+V_{DD}$
DV_{SS}	76	Supply	Ground
CLKMD1	77	I	Clock mode pin 1
CLKMD2	78	I	Clock mode pin 2
CLKMD3	79	I	Clock mode pin 3
NC	80	N/A	No connection
HD2	81	I/O/Z	Parallel bidirectional data bus (HPI)
TOUT	82	O/Z	Timer output
EMU0	83	I/O/Z	Emulator interrupt 0
EMU1/$\overline{OFF}$	84	I/O/Z	Emulator interrupt 1/shut off
TDO	85	O/Z	Test data output (IEEE standard 1149.1)

† Legend:
I = Input
O = Output
Z = High-impedance

Pin Functions for the TMS320LC549PGE (144-Pin TQFP Package) (Continued)

PIN NAME	PIN NO	FUNCTION†	DESCRIPTION
TDI	86	I	Test data input (IEEE standard 1149.1)
$\overline{TRST}$	87	I	Test reset (IEEE standard 1149.1)
TCK	88	I	Test clock (IEEE standard 1149.1)
TMS	89	I	Test mode select (IEEE standard 1149.1)
CVss	90	Supply	Ground
CV_{DD}	91	Supply	$+V_{DD}$
HPIENA/V_{DD}	92	I	HPI module select input
DVss	93	Supply	Ground
CLKOUT	94	O/Z	Machine clock output
HD3	95	I/O/Z	Parallel bidirectional data bus (HPI)
X1	96	O	Oscillator output
X2/CLKIN	97	I	Oscillator/external clock input
$\overline{RS}$	98	I	Device reset
D0–D5	99–104	I/O/Z	Parallel data port
A16	105	O/Z	Parallel port address bus
DVss	106	Supply	Ground
A17–A20	107–110	O/Z	Parallel port address bus
CVss	111	Supply	Ground
DV_{DD}	112	Supply	$+V_{DD}$
D6–D12	113–119	I/O/Z	Parallel data port
HD4	120	I/O/Z	Parallel bidirectional data bus (HPI)
D13–D15	121–123	I/O/Z	Parallel data port
HD5	124	I/O/Z	Parallel bidirectional data bus (HPI)
CV_{DD}	125	Supply	$+V_{DD}$
CVss	126	Supply	Ground
HDS1	127	I	Data strobe input (HPI)
DVss	128	Supply	Ground
HDS2	129	I	Data strobe input (HPI)
DV_{DD}	130	Supply	$+V_{DD}$
A0–A3	131–134	O/Z	Parallel port address bus
HD6	135	I/O/Z	Parallel bidirectional data bus (HPI)
A4–A9	136–141	O/Z	Parallel port address bus
CV_{DD}	142	Supply	$+V_{DD}$
A21	143	O/Z	Parallel port address bus
DV_{SS}	144	Supply	Ground

† Legend:
I = Input
O = Output
Z = High-impedance

The following tables list each signal, function, and operating mode(s) grouped by function.

Signal Descriptions

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

† Legend:
I = Input
O = Output
Z = High-impedance

Signal Descriptions (Continued)

PIN		DESCRIPTION
NAME	TYPE†	
INITIALIZATION, INTERRUPT AND RESET OPERATIONS (CONTINUED)		
$\overline{NMI}$	I	Nonmaskable interrupt. $\overline{NMI}$ is an external interrupt that cannot be masked via the INTM or the IMR. When $\overline{NMI}$ is activated, the processor traps to the appropriate vector location.
$\overline{RS}$	I	Reset input. $\overline{RS}$ causes the DSP to terminate execution and forces the program counter to 0FF80h. When $\overline{RS}$ is brought to a high level, execution begins at location 0FF80h of program memory. $\overline{RS}$ affects various registers and status bits.
MP/$\overline{MC}$	I	Microprocessor/microcomputer mode select pin. If active low at reset (microcomputer mode), MP / $\overline{MC}$ causes the internal program ROM to be mapped into the upper 16K words of program memory space. In the microprocessor mode, off-chip memory and its corresponding addresses instead of internal program ROM are accessed by the DSP.
MULTIPROCESSING SIGNALS		
$\overline{BIO}$	I	Branch control input. A branch can be conditionally executed when $\overline{BIO}$ is active. If low, the processor executes the conditional instruction. The $\overline{BIO}$ condition is sampled during the decode phase of the pipeline for XC instruction, and all other instructions sample $\overline{BIO}$ during the read phase of the pipeline.
XF	O/Z	External flag output (latched software-programmable signal). XF is set high by the SSBX XF instruction, set low by RSBX XF instruction or by loading ST1. XF is used for signaling other processors in multiprocessor configurations or as a general-purpose output pin. XF goes into the high-impedance state when $\overline{OFF}$ is low, and is set high at reset.
MEMORY CONTROL SIGNALS		
$\overline{DS}$ $\overline{PS}$ $\overline{IS}$	O/Z	Data, program, and I/O space select signals. $\overline{DS}$, $\overline{PS}$, and $\overline{IS}$ are always high unless driven low for communicating to a particular external space. Active period corresponds to valid address information. Placed into a high-impedance state in hold mode. $\overline{DS}$, $\overline{PS}$, and $\overline{IS}$ also go into the high-impedance state when $\overline{OFF}$ is low.
$\overline{MSTRB}$	O/Z	Memory strobe signal. $\overline{MSTRB}$ is always high unless low-level asserted to indicate an external bus access to data or program memory. Placed in high-impedance state in hold mode. $\overline{MSTRB}$ also goes into the high-impedance state when $\overline{OFF}$ is low.
READY	I	Data ready input. READY indicates that an external device is prepared for a bus transaction to be completed. If the device is not ready (READY is low), the processor waits one cycle and checks READY again. Note that the processor performs ready detection if at least two software wait states are programmed. The READY signal is not sampled until the completion of the software wait states.
R/$\overline{W}$	O/Z	Read/write signal. R/$\overline{W}$ indicates transfer direction during communication to an external device. Normally in read mode (high), unless asserted low when the DSP performs a write operation. Placed in high-impedance state in hold mode. R/$\overline{W}$ also goes into the high-impedance state when $\overline{OFF}$ is low.
$\overline{IOSTRB}$	O/Z	I/O strobe signal. $\overline{IOSTRB}$ is always high unless low level asserted to indicate an external bus access to an I/O device. Placed in high-impedance state in hold mode. $\overline{IOSTRB}$ also goes into the high-impedance state when $\overline{OFF}$ is low.
$\overline{HOLD}$	I	Hold input. $\overline{HOLD}$ is asserted to request control of the address, data, and control lines. When acknowledged by the 'C54x, these lines go into high-impedance state.
$\overline{HOLDA}$	O/Z	Hold acknowledge signal. $\overline{HOLDA}$ indicates to the external circuitry that the processor is in a hold state and that the address, data, and control lines are in a high-impedance state allowing them to be available to the external circuitry. $\overline{HOLDA}$ also goes into the high-impedance state when is $\overline{OFF}$ low.
$\overline{MSC}$	O/Z	Microstate complete signal. $\overline{MSC}$ goes low when the last wait state of two or more internal software wait states programmed is executed. If connected to the READY line, it forces one external wait state after the last internal wait state has been completed. $\overline{MSC}$ also goes into the high-impedance state when $\overline{OFF}$ is low.
$\overline{IAQ}$	O/Z	Instruction acquisition signal. $\overline{IAQ}$ is asserted (active low) when there is an instruction address on the address bus and goes into the high-impedance state when $\overline{OFF}$ is low.

† Legend:
I = Input
O = Output
Z = High-impedance

Signal Descriptions (Continued)

PIN		DESCRIPTION
NAME	TYPE†	
OSCILLATOR/TIMER SIGNALS		
CLKOUT	O/Z	Master clock output signal. CLKOUT cycles at the machine-cycle rate of the CPU. The internal machine cycle is bounded by the rising edges of this signal. CLKOUT also goes into the high-impedance state when $\overline{OFF}$ is low.
CLKMD1 CLKMD2 CLKMD3	I	Clock mode external/internal input signals. They allow you to select and configure different clock modes such as crystal, external clock, various PLL factors.
X2/CLKIN	I	Input pin to internal oscillator from the crystal. If the internal (crystal) oscillator is not being used, a clock can become input to the device using this pin. The internal machine cycle time is determined by the clock operating mode pins (CLKMD1, CLKMD2 and CLKMD3).
X1	O	Output pin from the internal oscillator for the crystal. If the internal oscillator is not used, X1 should be left unconnected. X1 does not go into the high-impedance state when $\overline{OFF}$ is low.
TOUT	O	Timer output. TOUT signals a pulse when the on-chip timer counts down past zero. The pulse is a CLKOUT cycle wide. TOUT also goes into the high-impedance state when $\overline{OFF}$ is low.
BUFFERED SERIAL PORT 0 (BSP #0) and BUFFERED SERIAL PORT 1 (BSP #1) SIGNALS		
BCLKR0 BCLKR1	I	Receive clock input. BCLKR serves as the serial shift clock for the buffered serial port receiver.
BDR0 BDR1	I	Serial data receive input.
BFSR0 BFSR1	I	Frame synchronization pulse for receive input. The BFSR pulse initiates the receive data process over BDR.
BCLKX0 BCLKX1	I/O/Z	Transmit clock. BCLKX serves as the serial shift clock for the buffered serial port transmitter. If $\overline{RS}$ is asserted when BCLKX is configured as output, then BCLKX is turned into input mode by reset operation. BCLKX goes into the high-impedance state when $\overline{OFF}$ is low.
BDX0 BDX1	O/Z	Serial data transmit output. BDX is placed in the high-impedance state when not transmitting, when $\overline{RS}$ is asserted or when $\overline{OFF}$ is low.
BFSX0 BFSX1	I/O/Z	Frame synchronization pulse for transmit input/output. The BFSX pulse initiates the transmit data process over BDX. If $\overline{RS}$ is asserted when BFSX is configured as output, then BFSX is turned into input mode by reset operation. BFSX goes into the high-impedance state when $\overline{OFF}$ is low.
TDM SERIAL PORT SIGNAL		
TCLKR	I	Receive clock input.
TDR	I	Serial data receive input.
TFSR/TADD	I/O	Receive frame synchronization or address
TCLKX	I/O/Z	Transmit clock.
TDX	O/Z	Serial data transmit output.
TFSX/TFRM	I/O/Z	Transmit frame synchronization.

† Legend:
I = Input
O = Output
Z = High-impedance

Signal Descriptions (Continued)

PIN		DESCRIPTION
NAME	TYPE†	
MISCELLANEOUS PIN		
NC		No connection.
HOST PORT INTERFACE SIGNALS		
HD0–HD7	I/O/Z	Parallel bidirectional data bus. Placed in high-impedance state when not outputting data. The signals go into the high-impedance state when $\overline{OFF}$ is low.
HCNTL0 HCNTL1	I	Control inputs.
HBIL	I	Byte identification input.
$\overline{HCS}$	I	Chip select input.
$\overline{HDS1}$ $\overline{HDS2}$	I	Data strobe inputs.
$\overline{HAS}$	I	Address strobe input.
HRW	I	Read/write input.
HRDY	O/Z	Ready output. This signal goes into the high-impedance state when $\overline{OFF}$ is low.
HINT	O/Z	Interrupt output. When the DSP is in reset, this signal is driven high. The signal goes into the high-impedance state when $\overline{OFF}$ is low.
HPIENA/V_{DD}	I	HPI module select input. This signal must be tied to V_{DD} to have HPI selected. If this input is left open or connected to ground, HPI module will not be selected, internal pull-up for HPI input pins are enabled and HPI data bus has keepers set.
SUPPLY PNS		
CV_{SS}	Supply	Ground. Dedicated power supply for the core CPU.
CV_{DD}	Supply	$+V_{DD}$. Dedicated power supply for the core CPU.
DV_{SS}	Supply	Ground. Dedicated power supply for I/O pins.
DV_{DD}	Supply	$+V_{DD}$. Dedicated power supply for I/O pins.
TEST PINS		
TCK	I	IEEE standard 1149.1 test clock. This is normally a free-running clock signal with a 50% duty cycle. The changes on 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.
TD	I	IEEE standard 1149.1 test data input, pin with internal pull-up device. TDI is clocked into the selected register (instruction or data) on a rising edge of TCK.
TDO	O/Z	IEEE standard 1149.1 test data output. The contents of the selected register (instruction or data) is shifted out of TDO on the falling edge of TCK. TDO is in the high-impedance state except when scanning of data is in progress. TDO also goes into the high-impedance state when $\overline{OFF}$ is low.
TMS	I	IEEE standard 1149.1 test mode select. Pin with internal pull-up device. This serial control input is clocked into the test access port (TAP) controller on the rising edge of TCK.
$\overline{TRST}$	I	IEEE standard 1149.1 test reset. $\overline{TRST}$, when high, gives the IEEE standard 1149.1 scan system control of the operations of the device. If $\overline{TRST}$ is not connected or driven low, the device operates in its functional mode, and the IEEE standard 1149.1 signals are ignored. Pin with internal pull-up device.

† Legend:
I = Input
O = Output
Z = High-impedance

Signal Descriptions (Continued)

PIN		DESCRIPTION
NAME	TYPE†	
TEST PINS (CONTINUED)		
EMU0	I/O/Z	Emulator 0 pin. When $\overline{\text{TRST}}$ is driven low, EMU0 must be high for activation of the $\overline{\text{OFF}}$ condition. When $\overline{\text{TRST}}$ is driven high, EMU0 is used as an interrupt to or from the emulator system and is defined as input/output via IEEE standard 1149.1 scan system.
EMU1/$\overline{\text{OFF}}$	I/O/Z	Emulator 1 pin/disable all outputs. When $\overline{\text{TRST}}$ is driven high, EMU1/$\overline{\text{OFF}}$ is used as an interrupt to or from the emulator system and is defined as input/output via IEEE standard 1149.1 scan system. When $\overline{\text{TRST}}$ is driven low, EMU1/$\overline{\text{OFF}}$ is configured as $\overline{\text{OFF}}$. The EMU1/$\overline{\text{OFF}}$ signal, when active low, puts all output drivers into the high-impedance state. Note that $\overline{\text{OFF}}$ is used exclusively for testing and emulation purposes (not for multiprocessing applications). Thus, for $\overline{\text{OFF}}$ condition, the following conditions apply: $\overline{\text{TRST}}$ = low, EMU0 = high EMU1/$\overline{\text{OFF}}$ = low

† Legend:
I = Input
O = Output
Z = High-impedance

1. MEMORY

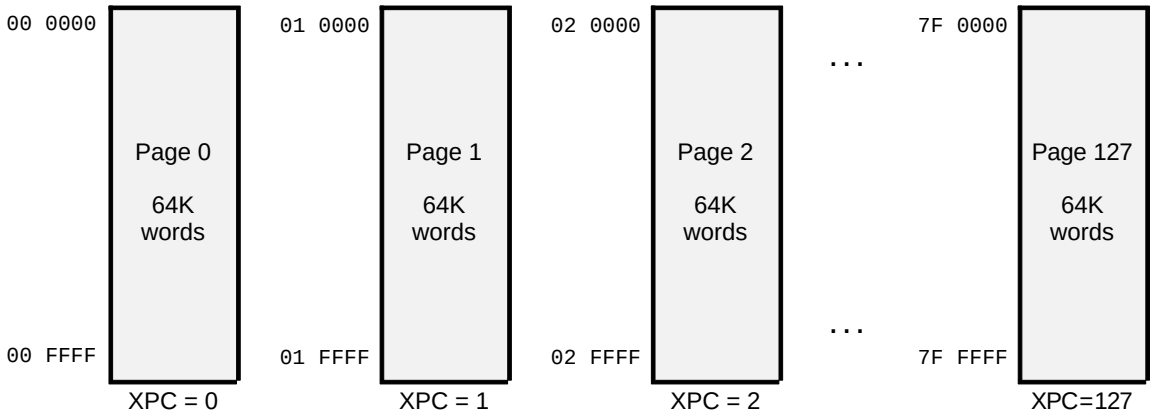
1.1 Extended Program Memory

The ’C549 uses a paged extended memory scheme in program space to allow access of up to 8192K of program memory. In order to implement this scheme, the ’C549 includes several features which are also present on ’C548 :

- 23 address lines, instead of 16
- An extra memory-mapped register, the XPC
- Six extra instructions for addressing extended program space

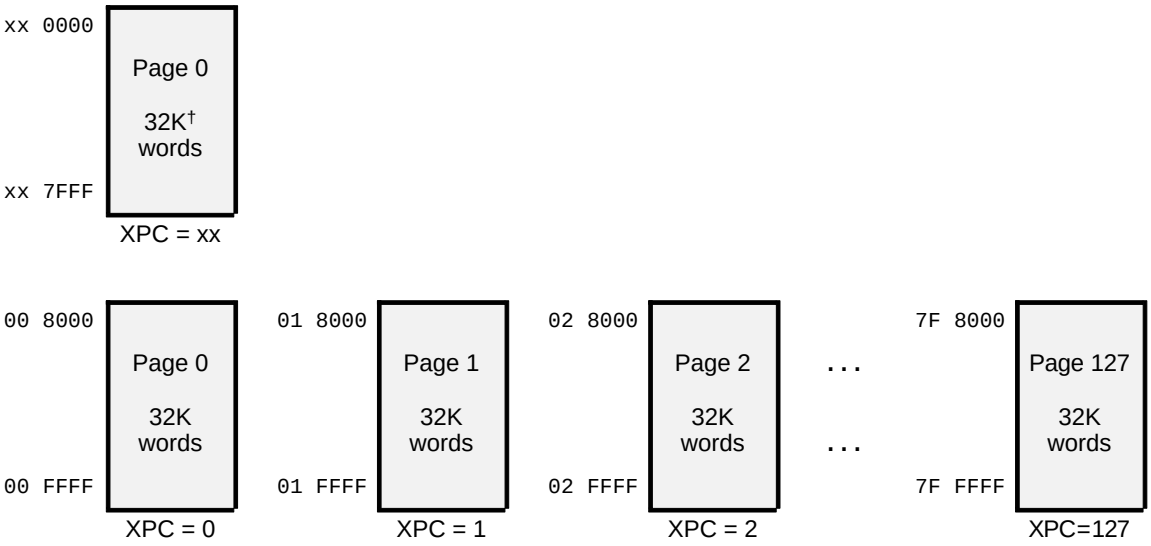
Program memory in the ’C549 is organized into 128 pages that are each 64K in length, as shown in Figure 1.

Figure 1. Extended Program Memory (On-Chip RAM Not Mapped in Program Space and Data Space, OVLY = 0)



When the on-chip RAM is enabled in program space, each page of program memory is made up of two parts: a common block of 32K words and a unique block of 32K words. The common block is shared by all pages and each unique block is accessible only through its assigned page. Figure 2 shows the common and unique blocks.

Figure 2. Extended Program Memory (On-Chip RAM Mapped in Program Space and Data Space, OVLY = 1)



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

† See Figure 3 for more information about this on-chip memory region.

If the on-chip ROM is enabled (MP/$\overline{MC}$ = 0), it is enabled only on page 0. It is not mapped to any other page in program memory.

The value of the XPC register defines the page selection. This register is memory-mapped into data space to address 001Eh. At a hardware reset, the XPC is initialized to 0.

To facilitate page switching through software, the ’C549 has six special instructions that affect the XPC:

- *FB[D]* pmad (23 bits) – Far branch
- *FBACC[D]* Accu[22:0] – Far branch to the location specified by the value in accumulator A or accumulator B
- *FCALL[D]* pmad (23 bits) – Far call
- *FCALA[D]* Accu[22:0] – Far call to the location specified by the value in accumulator A or accumulator B
- *FRET[D]* – Far return
- *FRETE[D]* – Far return with interrupts enabled

In addition to these new instructions, two ’54x instructions are extended to use 23 bits in the ’C549:

- READA data_memory (using 23-bit accumulator address)
- WRITA data_memory (using 23-bit accumulator address)

All other instructions do not modify the XPC register and access only memory within the current page.

1.2 On-chip ROM

The ’C549 features a 16K-word × 16-bit on-chip maskable ROM which can be mapped into program and data memory space if the DROM bit in the processor mode status (PMST) register is set. This allows an instruction to use data stored in the ROM as an operand.

Customers can arrange to have the ROM of the ’C549 programmed with contents unique to any particular application.

A boot loader is available in the standard ’C549 on-chip ROM. This boot loader can be used to automatically transfer user code from an external source to anywhere in the program memory at power-up. If MP/$\overline{MC}$ of the device is sampled low during a hardware reset, execution begins at location FF80h of the on-chip ROM. This location contains a branch instruction to the start of the boot loader program. The standard ’C549 devices provide different ways to download the code to accomodate various system requirements:

- Parallel from 8-bit of 16-bit wide EPROM
- Parallel from I/O space 8-bit of 16-bit mode
- Serial boot from serial ports 8-bit of 16-bit mode
- Host port interface boot
- Warm boot

1.3 On-chip RAM

The ’C549 device contains 8K-word × 16-bit of on-chip Dual Access RAM (DARAM) and 24K–word x 16–bit of on–chip Single Access RAM (SARAM).

The DARAM is composed of four blocks of 2K words each. Each of these four blocks can be dual accessed. It can be read from and written to in the same cycle, or it can be double read from in the same cycle. The DARAM is located in the address range 80h–1FFFh in data space, and can be mapped into program/data space if the OVLY bit is set to one.

The SARAM is composed of three blocks of 8K words each. Each of these three blocks is a single access memory. For instance an instruction word can be fetched from one SARAM block in the same cycle as a data word is written to another SARAM block. The SARAM is located in the address range 2000h–7FFFh in data space, and can be mapped into program/data space if the OVLY bit is set to one.

1.4 On-chip memory security

The ’C549 device has a maskable option to protect the contents of on-chip memories. When the related bit is set, no externally originating instruction can access the on-chip memory spaces.

1.5 Memory map

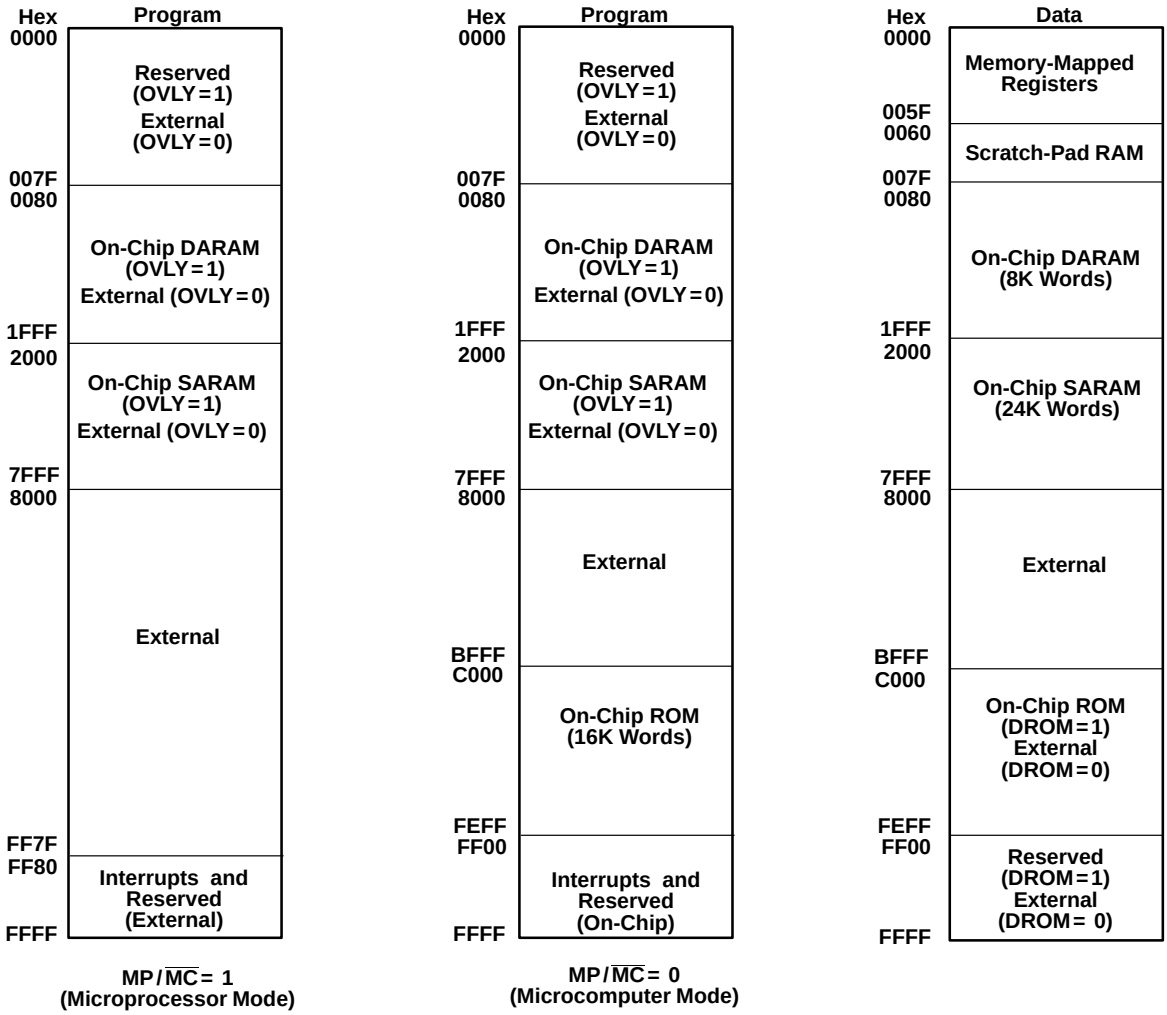


Figure 3. Memory Map of ’C549

2. ON–CHIP PERIPHERALS

The ’C549 device has the same set of peripherals as the ’C548:

- Software-programmable wait-state generator
- Programmable bank switching
- A host port interface
- Two Buffered Serial Ports
- A TDM Serial Port
- A hardware timer
- A clock generator with a multiple Phase–Locked Loop (PLL)

2.1 Software-programmable wait-state generator

Upgraded from ’C548. A detailed description follows.

The software-programmable wait-state generator can extend external bus cycles by up to seven machine cycles, providing a convenient means to interface the ’C549 to slower external devices. Devices that require more than seven wait states can be interfaced using the hardware READY line. When all external accesses are configured for zero wait states, the internal clocks to the wait-state generator are shut off; shutting off these paths from the internal clocks allows the device to run with lower power consumption.

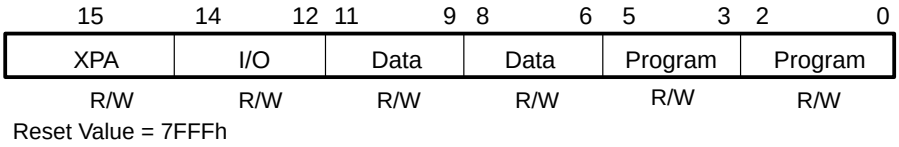
The software-programmable wait-state generator is controlled by the 16-bit software wait-state register (SWWSR), which is memory-mapped to address 0028h in data space.

The program and data spaces each consist of two 32K-word blocks; the I/O space consists of one 64K-word block. Each of these blocks has a corresponding 3-bit field in the SWWSR. These fields are shown in Figure 4 and described immediatly following the figure.

The value of a 3-bit field in SWWSR specifies the number of wait states to be inserted for each access in the corresponding space and address range.

At reset the SWWSR is set to 7FFFh, making the number of wait states for all external accesses to be 7.

Figure 4. Software Wait-State Register (SWWSR)



Reset Value = 7FFFh

The following are the fields in the SWWSR:

Bit	Name	Reset Value	Function
15	XPA	0	Extended program address control bit. Selects the address ranges selected by the program fields.
14–12	I/O	1	I/O space. The field value (0–7) corresponds to the number of wait states for I/O space 0000–FFFFh.
11–9	Data	1	Data space. The field value (0–7) corresponds to the number of wait states for data space 8000–FFFFh.
8–6	Data	1	Data space. The field value (0–7) corresponds to the number of wait states for data space 0000–7FFFh.
5–3	Program	1	Program space. The field value (0–7) corresponds to the number of wait states for: ❑ XPA = 0: xx8000 – xxFFFFh ❑ XPA = 1: 400000h–7FFFFF
2–0	Program	1	Program space. The field value (0–7) corresponds to the number of wait states for: ❑ XPA = 0: xx0000–xx7FFFh ❑ XPA = 1: 000000–3FFFFFh

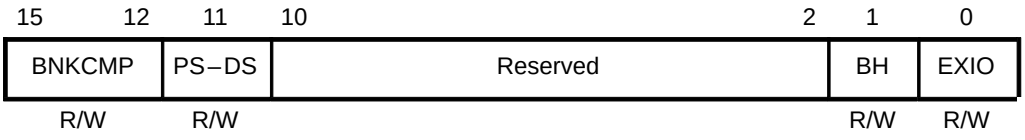
2.2 Programmable bank switching

Same a ’C548. More detailed description follows.

Programmable bank-switching logic allows the ’C549 to switch between external memory banks without requiring external wait states for memories that need several cycles to turn off. The bank-switching logic automatically inserts one cycle when accesses cross memory-bank boundaries inside program or data space.

Bank switching is defined by the bank-switching control register (BSCR), which is memory-mapped at address 0029h. Figure 5 shows the BSCR and its fields are described immediately following the figure.

Figure 5. Bank Switching Control Register (BSCR)



The following are the fields in the BSCR:

Bit	Name	Function
0	EXIO	**External bus interface off**. The EXIO bit controls the external-bus-off function. EXIO is cleared to 0 at reset. EXIO = 0 The external-bus-off function is disabled. EXIO = 1 The external-bus-off function is enabled. The address bus, data bus, and control signals become inactive after completing the current bus cycle. The following table lists the state of the signals when EXIO=1.

Signal	State	Signal	State
A(22–0)	Previous state	R/$\overline{W}$	High level
D(15–0)	High impedance	$\overline{MSC}$	High level
$\overline{PS}$, $\overline{DS}$, $\overline{IS}$	High level	$\overline{IAQ}$	High level
$\overline{MSTRB}$, $\overline{IOSTRB}$	High level		

1	BH	**Bus holder.** Controls the bus holder:BH is cleared to 0 at reset. BH = 0 The bus holder is disabled. BH = 1 The bus holder is enabled. The data bus, D(15–0), is held in the previous logic level.
10–2	Reserved	
11	PS–DS	**Program read–data read access.** 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 accesses of program read and data read, or data read and program read.
15–12	BNKCMP	**Bank compare.** Determines the external memory-bank size. BNKCMP is used to mask the four MSBs of an address. For example, if BNKCMP = 1111_2, the four MSBs (bits 12–15) are compared, resulting in a bank size of 4K words. Bank sizes from 4K words to 64K words are allowed.

The following table summarizes the relationship between BNKCMP, the address bits to be compared, and the bank size. BNKCMP values not listed in the table are not allowed.

BNKCMP	MSBs to Compare	Bank Size (16-Bit Words)
0000	None	2^{16} = 64K
1000	15	2^{15} = 32K
1100	15–14	2^{14} = 16K
1110	15–13	2^{13} = 8K
1111	15–12	2^{12} = 4K

The ’C549 has an internal register that contains the MSBs (as defined by the BNKCMP field) of the last address used for a read or write operation in program or data space. If the MSBs of the address used for the current read do not match those contained in this internal register, the $\overline{\text{MSTRB}}$ (memory strobe) signal is not asserted for one CLKOUT cycle. During this extra cycle, the address bus switches to the new address. The contents of the internal register are replaced with the MSBs for the read of the current address. If the MSBs of the address used for the current read match the bits in the register, a normal read cycle occurs.

If repeated reads are performed from the same memory bank, no extra cycles are inserted. When a read is performed from a different memory bank, memory conflicts are avoided by inserting an extra cycle. An extra cycle is inserted only if a read memory access is followed by another read memory access. This feature can be disabled by clearing BNKCMP to 0.

The ’C549 bank-switching mechanism automatically inserts one extra cycle in the following cases:

- A memory read followed by another memory read from a different memory bank.
- A program-memory read followed by a data-memory read when the PS–DS bit is set to 1.
- A data-memory read followed by a program-memory read when the PS–DS bit is set to 1.
- A program-memory read followed by another program-memory read from a different page.

2.3 Host Port Interface

Same a ’C548. More detailed description follows.

The host port interface is an 8-bit parallel port used to interface a host processor to the DSP device. Information is exchanged between the DSP device and the host processor through on-chip memory that is accessible by both the host and the DSP device. The DSP devices have access to the HPI control register (HPIC) and the host can address the HPI memory via the HPI address register (HPIA). HPI memory is a 2K 16-bit dual-access RAM block which can also be used as general-purpose on-chip data or program dual-access RAM. This memory is located at address range 1000h–17FFh .

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

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

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

Host Port Interface (continued)

The HPI supports high-speed back-to-back accesses. In the shared-access mode, it can handle one byte every five DSP device periods, that is, 128 Mb/s with a 80-MIPS DSP. The HPI is designed so that the host can take advantage of this high bandwidth and run at frequencies up to (f × n) ÷ 5, where n is the number of host cycles for an external access and f is the DSP device frequency. In host-only mode, the HPI supports even higher speed back-to-back host accesses: 1 byte every 50 ns, that is, 160 Mb/s, independently of the DSP device clock rate.

2.4 Serial Ports

Same a 'C548, except two differences: a) there is an extra feature on the BSP that allows detection of word misalignment in transmit or receive buffer (a more detailed description follows) and b) when the TDM serial port operates in TDM mode it must be with an external clock and an external frame only .

The 'C549 devices provide high-speed full-duplex serial ports that allow direct interface to other 'C549 devices, codecs, and other devices in a system. There is a time-division-multiplexed (TDM) serial port, and two auto-buffered serial port (BSP).

The TDM port allows the device to communicate through time-division multiplexing with up to seven other 'C549 devices. Time-division multiplexing is the division of time intervals into a number of subintervals with each subinterval representing a prespecified communication channel. The TDM port serially transmits 16-bit words on a single data line (TDAT) and destination addresses on a single address line (TADD). Each device can transmit data on a single channel and receive data from one or more of the eight channels providing a simple and efficient interface for multiprocessing applications. A frame synchronization pulse occurs once every 128 clock cycles corresponding to transmission of one 16-bit word on each of the eight channels.

The TDM port can also be configured in software to operate as a general-purpose serial port . In this general–purpose mode, the serial port utilizes two memory-mapped registers for data transfer: the data transmit register (DXR) and the data receive register (DRR). Both of these registers can be accessed in the same manner as any other memory location. The transmit and receive sections of the serial port each have associated clocks, frame-synchronization pulses, and serial shift registers, and serial data can be transferred either in bytes or in 16-bit words. Serial port receive and transmit operations can generate their own maskable transmit and receive interrupts (TXNT and TRNT), allowing serial port transfers to be managed via software. The 'C549 serial ports are double buffered and fully static.

Both types of ports described above are capable of operating at up to one-eighth the machine cycle rate (CLKOUT).

The buffered serial port (BSP) consists of a full-duplex double-buffered serial port interface (SPI) and an auto-buffering unit (ABU). The SPI block of the BSP is an enhanced version of the general-purpose serial port. The auto-buffering unit allows the SPI to read/write directly to 'C549 internal memory using a dedicated bus independently of the CPU. This results in minimum overhead for SPI transactions and faster data rates.

When auto-buffering capability is disabled (standard mode), transfers with SPI are performed under software control through interrupts. In this mode, the ABU is transparent and the word-based interrupts (WXINT and WRINT) provided by the SPI are sent to the CPU as transmit interrupt (XINT) and receive interrupt (RINT). When auto buffering is enabled, word transfers are done directly between the SPI and the 'C549 internal memory using ABU-embedded address generators.

Serial Ports (Continued)

The ABU has its own set of circular addressing registers with corresponding address-generation units. Memory for the buffers reside in an on–chip dual–access RAM block of the ’C549. The size of this BSP memory is 2K words. The length and starting addresses of the buffers are user programmable. A buffer empty/full interrupt can be posted to the CPU. Buffering is easily halted thanks to an auto-disabling capability. Auto-buffering capability can be separately enabled for transmit and receive sections. When auto buffering is disabled, operation is similar to the general-purpose serial port.

The SPI allows transfer of 8-, 10-, 12-, or 16-bit data packets. In burst mode, data packets are directed by a frame synchronization pulse for every packet. In continuous mode, the frame synchronization pulse occurs when the data transmission is initiated and no further pulses occur. The frame and clock strobes are frequency and polarity programmable. The SPI is fully static and operates at arbitrarily low clock frequencies. The maximum operating frequency is 50 MBits/sec.

The on–chip RAM memory that can be used by BSP #0 is at address range 0800h–0FFFh.
The on–chip RAM memory that can be used by BSP #1 is at address range 1800h–18FFh.

Description of the new mis–alignment detection feature :
This feature allows the BSP to detect when a word or some words are lost in the serial data line. For example, words can be lost when the external serial clock stops to run for a while and when data bits continue to be sent through BDX pin or received through BDR pin. The DSP is not responsible for this type of off–chip incident due to a dirty external clock signal, nevertheless the DSP can help solving that type of problem.

. Conditions : This feature is applicable when the BSP is in ABU and continuous mode with FIG bit set to 0 and with external serial clocks and external frames.

This feature can be used properly if the address register is initialized to the top of the buffer: ARR must be initialize to point to the top of the receive buffer, AXR must be initialized to point to the top of the transmit buffer. If not, the BSP operates correctly, but does not take advantage of this new feature. In this case, the new interrupt signal BMINTmust be masked in IMR register so that the DSP ignores this interrupt.
Also, this feature can be used properly if the frame signals occur periodically with a period equal to the receive or to the transmit buffer. For example with a receive buffer of size 64, BFSR signal must be active every 64 words.If not, the BSP operates correctly, but does not take advantage of this new feature. In this case, the new interrupt signal BMINTmust be masked in IMR register so that the DSP ignores this interrupt.

. Description : each time the BSP receives a frame signal on BFSR, it internally checks that the ARR register points to the right position in the receive buffer (where it should normally points to when there is no error), so that the first received word coming with this frame is stored at the top of the receive buffer.
Each time the BSP receives a frame signal on BFSX, it internally checks that the AXR register points to the right position in the transmit buffer (where it should normally points to when there is no error), so that the first transmitted word going out with this frame is taken from the top of the transmit buffer.
If one of these two verifications by the BSP shows that the receive buffer pointer or the transmit buffer pointer is mis–aligned, then the BSP generates an interrupt signal to the DSP CPU.

. Types of error detection : mis–alignment of one word or more when a frame comes is detected. The mis–alignment can be detected either for an advanced position of the address pointer to the buffer (the address pointer has already passed the top of the buffer, this can be caused by dirty glitches on the external serial clock, or by an external frame coming too late) or for a delayed position of the address pointer to the buffer (the address pointer has not rewound yet to the top of the buffer, this can be caused by missing edges on the external serial clock or by an external frame coming too early).

2.5 Hardware Timer

Same a ’C548. More detailed description follows.

The ’C549 device features a 16-bit timing circuit with a four-bit prescaler The timer counter is decremented by 1 at every CLKOUT cycle. Each time the counter decrements to 0, a timer interrupt is generated. The timer can be stopped, restarted, reset, or disabled by specific status bits.

2.6 Clock Generator

Same a ’C548. More detailed description follows.

2.6.1 Functional description

This clock generator allows system designers to select the clock source. The sources that drive the clock generator are:

- A crystal resonator circuit. The crystal resonator circuit is connected across the X1 and X2/CLKIN pins of the ’LC549 to enable the internal oscillator.
- An external clock. The external clock source is directly connected to the X2/CLKIN pin, and X1 is left unconnected.

The clock generator on the ’LC549 devices consists of an internal oscillator and a phase-locked loop (PLL) circuit. The PLL circuit is software-programmable, and can be configured in one of two clock modes:

- PLL mode. The input clock (CLKIN) is multiplied by one of 31 possible ratios from 0.25 to 15. These ratios are achieved with an analog voltage controlled oscillator (VCO).
- DIV (divider) mode. The input clock (CLKIN) is divided by 2 or 4.

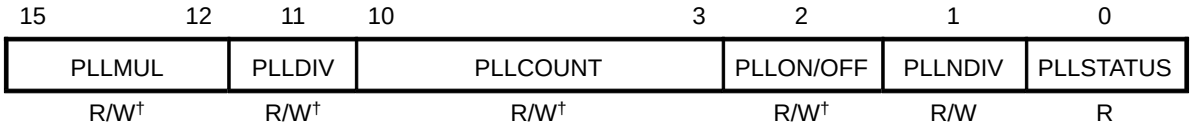
The clock mode is initially determined by the values of the three external pins, CLKMD1, CLKMD2, and CLKMD3. The modes corresponding to the CLKMD pins are shown in Figure 6.

Figure 6. Clock Mode Settings at Reset

CLKMD1	CLKMD2	CLKMD3	CLKMD Reset Value	Clock Mode
0	0	0	0000h	1/2 with external source
0	0	1	1000h	1/2 with external source
0	1	0	2000h	1/2 with external source
1	0	0	4000h	1/2 with internal source
1	1	0	6000h	1/2 with external source
1	1	1	7000h	1/1 with external source
1	0	1	0007h	PLL × 1 with external source
0	1	1	—	Stop mode

The programming of the PLL is loaded in the 16-bit memory-mapped (address 58h) clock mode register (CLKMD). The CLKMD is used to define the clock configuration of the PLL clock module. The CLKMD bit fields are shown Figure 7 and described in Table 8.

Figure 7. Clock Mode Register (CLKMD) Layout



† PLLMUL, PLLDIV, PLLCOUNT, and PLLON/OFF are write, only when PLLSTATUS is low.

Table 8. Clock Mode Register (CLKMD) Bit Summary

Bit	Name	Function
15–12	PLLMUL	PLL multiplier. Defines the frequency multiplier in conjunction with PLLDIV and PLLNDIV, as shown in Table 9 .
11	PLLDIV	PLL divider. Defines the frequency multiplier in conjunction with PLLMUL and PLLNDIV, as shown in Table 9. PLLDIV=0 means that an integer multiply factor is used. PLLDIV=1 means that a non–integer multiply factor is used.
10–3	PLLCOUNT	PLL counter value. Specifies the number of cycles for the PLL timer to count before the PLL clocks the processor when the PLL is programmed. See subsection *PLL Clock Programmable-Timer*, for more information about PLLCOUNT.
2	PLLON/OFF	PLL on/off. Enables or disables the analog part of the clock generator. PLLON/OFF = 0 Analog components disabled. PLLON/OFF = 1 Analog components enabled. PLLON/OFF and PLLNDIV both force the VCO to operate; when PLLON/OFF is high, the VCO runs independently of the state of PLLNDIV:
1	PLLNDIV	PLL clock generator select. Determines whether the clock generator works in PLL mode or in divider (DIV) mode, thus defining the frequency multiplier in conjunction with PLLMUL and PLLDIV. PLLNDIV = 0 Divider mode is used. PLLNDIV = 1 PLL mode is used. Note : writing 1 to this bit starts the decrement of the counter set with PLLCOUNT bits.
0	PLLSTATUS	PLL status. Indicates the mode in which the clock generator is operating. PLLSTATUS = 0 DIV mode PLLSTATUS = 1 PLL mode

PLLON/OFF	PLLNDIV	VCO State
0	0	off
1	0	on
0	1	on
1	1	on

Table 9. Multiplier Related to PLLNDIV, PLLDIV, and PLLMUL

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

† CLKOUT = CLKIN × Multiplier

PLL Clock Programmable-Timer

Because the PLL clock is analog, the PLL requires a lockup time before it is stable.

A programmable timer is included in the PLL to release the PLL clock mode to the DSP after the VCO is locked up; this timer is a counter (set by PLLCOUNT in CLKMD) which decrements from a preset value to 0. The timer can be preset from 0 to 255 and its input clock is CLKIN divided by 16. The corresponding lockup time can be set from 0 to 255 × 16 CLKIN cycles.

The decimal preset value, PLLCOUNT, is:

$$PLLCOUNT = \frac{LockupTime}{16 \times T_{CLKIN}}$$

where T_{CLKIN} is the input reference clock period and LockupTime is the VCO lockup time which needs to be higher than 50 µSec.

2.6.2 How to change from older style C54x PLL to LC549 software programmable PLL

This section is specially addressed to the users of older C54x devices and its hardware programmable PLL. It describes what modifications in the user software are needed to use the LC549 software programmable PLL.

Start–up

The start–up mode depends on CLKMD1/2/3 pin state as given by Figure 6. Most of the start–up modes at device reset are divide–by–two clock. Switching from a divide–by–two to a PLL mode is simple to implement in software, it requires one additional instruction for initialization of the CLKMD register.
For example this additional instruction needed to switch from DIV to PLL x3 with 13MHz CLKIN is :
STM #0010000101001111b,CLKMD

For further description, check 'Switching clock mode from DIV to PLL' paragraph.

Idle1/Idle2

No change is needed in the software to go to or to come from IDLE1 or IDLE 2.

Idle 3

A few extra code lines are needed to keep a Idle 3 behavior and wake–up latency similar with older hardware programmable PLL C54x devices. Before executing 'IDLE 3' instruction, the user needs to switch from PLL to DIV mode (check 'Switching clock mode from PLL to DIV' paragraph for detailed decription). After the wake–up from Idle 3, the user needs to initialize CLKMD register to switch from DIV to PLL mode (check 'Switching clock mode from DIV to PLL' paragraph for detailed description).

The following example of instructions illustrates the case of a LC549 device running with PLL x3, with 13MHz CLKIN, and with a IDLE3 wake–up time of 3072 CLKOUT cycles (3072 CLKOUT cycles was the IDLE3 wake–up time existing with hardware programmable PLL C54x device for PLL x3).
Before IDLE3 :

```
              STM  #0b,CLKMD             ; switch to DIV
TstStatu :    LDM  CLKMD,A
              AND  #01b,A                ; poll STATUS bit
              BC   TstStatu,ANEQ
              STM  #0b,CLKMD             ; reset PLLON_OFF when STATUS is DIV mode
              IDLE 3
```

After IDLE3 wake–up :

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

2.6.3 How to program the LC549 software programmable PLL

This section describes in details how to program the PLL when changing between clock modes:

- Switching clock mode from DIV to PLL mode.
- Switching clock mode from PLL to DIV mode.
- Switching clock mode from PLL mode to a PLL mode with a different multiplying factor.
- Going to IDLE 1 or IDLE 2
  –> simple way, same as hardware programmble PLL C54x device
  –> lower power way (new enhanced way)
  –> much lower power but higher latency way (new enhanced way)
- Going to IDLE 3

Switching clock mode from DIV to PLL mode

Access the CLKMD register to write :

- in the PLLMUL[3:0] and PLDIV fields the values according to the expected multiplication ratio,
- in the PLLCOUNT[7:0] fields the value according to the calculated lockup time,
- in the PLLNDIV bit in order for the PLLCOUNT[7:0] to start decrementing from its preset value.

Then, PLLCOUNT begins to decrement from its preset value and as soon as PLLCOUNT reaches 0, PLL clock mode is provided to the output 6 CLKIN cycles plus 3.5 PLL(VCO frequency) cycles later. During this PLL lock–up period, the DSP continues to operate in DIV mode.

NOTE: Writing to PLMUL,PLDIV,PLCOUNT and PLON_OFF fields is possible only in DIV mode.
PLON_OFF field of CLKMD register is forced to high when a switch to PLL mode is requested (PLLNDIV goes high).

Example : Switching clock mode from DIV to PLL*3, when CLKIN frequency is 13MHz (PLLCOUNT=41 in decimal)

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

Switching clock mode from PLL to DIV mode

1) Access CLKMD register to write 0 in the PLLNDIV fields of CLKMD register.

2) The DIV clock mode will be enabled to the device 12 CLKIN cycles plus 3.5 PLL (VCO frequency) cycles later this last update of CLKMD register if PLLMUL=1111b or 6 CLKIN cycles plus 3.5 PLL cycles for any other PLLMUL values. When DIV clock mode is enabled, the Status bit is set to LOW.

3) Reset PLON_OFF in order to stop VCO (optional, needed when lowest power is desired).

Example : Switching clock mode from PLL to DIV

```
            STM #0b,CLKMD
TstStatu :  LDM CLKMD,A
            AND #01b,A
            BC  TstStatu,ANEQ
            STM #0b,CLKMD
```

Switching ratio from PLL mode to PLL mode

1) Access CLKMD register to write 0 in the PLLNDIV fields of CLKMD register, to enable the DIV clock mode.

2) Access CLKMD register to read STATUS bit. If status is LOW then DIV mode is enabled and PLMUL, PLDIV, PLCOUNT fields can be updated.

3) Access CLKMD register to write in the PLLMUL[3:0] and PLLDIV fields the values according to expected multiplication ratio, in the PLLCOUNT[7:0] field the required counter value, and write 1 in PLLNDIV bit to start decrementing PLLCOUNT[7:0].

4) As soon as PLLCOUNT reaches 0, PLL clock mode is provided to the output 6 CLKIN plus 3.5 PLL (VCO frequency) cycles later.

Example : Switching clock mode from PLL*x to PLL*1

```
              STM #0b,CLKMD
  TstStatu :  LDM CLKMD,A
              AND #01b,A
              BC  TstStatu,ANEQ
              STM #0000001111101111b,CLKMD
```

Going from PLL to Idle1/2 mode

There are three possible ways to go to Idle 1 or Idle 2 modes.

Way #1 : same as hardware programmable PLL C54X, ie. simply run 'IDLE 1' or 'IDLE 2' instruction.

Way #2 : lower power. It is a new and enhanced option compared to way #1. The DSP clock is in DIV mode instead of PLL mode. A detailed procedure follows :

1) Switch from PLL mode to DIV mode by resetting PLLNDIV field with a CLKMD register access in write mode and keep VCO running to minimize latency time for VCO lockup time.

2) Execute Idle1/2 instruction.

3) Upon interrupt, the device is in DIV mode. Select PLL mode by setting PLLNDIV field of CLKMD register. PLL clock mode with a multiplication ratio is provided to the output 6 CLKIN cycles plus 3.5 PLL (VCO frequency) cycles later this last update of CLKMD register (there is no VCO lockup time since VCO has been kept running during Idle 1 or Idle 2 mode).

Way #3 : even lower power, but with higher latency. It is a new and enhanced option compared to way #1. The DSP clock is in DIV mode instead of PLL mode and the VCO is stopped. A detailed procedure follows :

1) Switch from PLL mode to DIV mode by resetting PLLNDIV field with a CLKMD register access in write mode. The DIV clock mode will then be enabled.

2) Access CLKMD register to read STATUS bit : if status is LOW, DIV mode is enabled and PLLMUL, PLLDIV, PLLCOUNT fields can be updated.

3) Program PLLMUL[3:0], PLLDIV and PLLCOUNT[7:0] according to expected multiplication ratio when DSP goes out Idle1/2 mode, and disable analog part to stop VCO running by resetting PLLON_OFF field.

4) Execute Idle1/2 instruction.

5) Upon interrupt, select PLL mode by setting PLLNDIV field of CLKMD register ; then PLLCOUNT[7:0] begins to decrement from its preset value stored in step 4), and as soon as PLLCOUNT reaches 0, the PLL is provided to the output 6 CLKIN plus 3.5 PLL (VCO frequency) cycles later.

Going from PLL to Idle3 mode

In Idle3 mode, a gate disables the reference input clock at the PLL input : so no reference signal is left at the Phase Frequency Detector input and the VCO is in an unknown state.

Before going in to Idle3 mode, PLL needs to run in DIV mode and VCO needs to be stopped.

Device is assumed to be supplied by PLL clock mode with a certain multiplication ratio between input (CLKIN) and output (CLKOUT) frequency (PLLON_OFF=1, and VCO running).

1) Switch from PLL mode to DIV mode by resetting PLLNDIV field with a CLKMD register access in write mode. The DIV clock mode will be enabled to the device 12 CLKIN cycles plus 3.5 PLL (VCO frequency) cycles later this last update of CLKMD register PLLMUL=1111b or 6 CLKIN cycles plus 3.5 PLL cycles for any other PLLMUL values.

2) Access CLKMD register to read STATUS bit : if status is LOW, DIV mode is enabled and PLLON_OFF bit can then be updated.

3) Reset PLLON_OFF to disable analog part and stop VCO. PLLMUL[3:0], PLDIV and PLLCOUNT[7:0] can be programmed either at that stage or after Idle 3 wake–up (step 5) to prepare the multiplication ratio of the PLL when the DSP will be out of Idle 3. This is a user decision.

4) Execute IDLE 3 instruction.

5) Upon interrupt, device will start in DIV mode. Select PLL mode by setting PLLNDIV field of CLKMD register ; If not already done in step 3, initialize also PLLMUL, PLLDIV and PLLCOUNT. As soon as PLLCOUNT reaches 0, PLL clock mode is provided to the output 6 CLKIN cycles plus 3.5 PLL cycles later.

3. Memory–mapped registers

The ’C549 has 27 memory-mapped CPU registers, which are mapped in data memory space address 0h to 1Fh. Each of these devices also has a set of memory-mapped registers associated with peripherals. Table 11 gives a list of CPU memory-mapped registers (MMR) available on ’C549. Table 12 shows additional peripheral MMRs associated with the ’C549.

Table 11. Core Processor Memory-Mapped Registers

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

Table 12. Peripheral Memory-Mapped Registers

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

† Auto-buffering unit (ABU)

4. INTERRUPTS

Vector–relative locations and priorities for all internal and external interrupts are show in Table 13 .

Table 13. ’C549 Interrupt Locations and Priorities

Name	Location		Priority	Function
	Decimal	Hex		
$\overline{RS}$, SINTR	0	00	1	Reset (Hardware and software reset)
$\overline{NMI}$, 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
$\overline{INT0}$, SINT0	64	40	3	External user interrupt #0
$\overline{INT1}$, SINT1	68	44	4	External user interrupt #1
$\overline{INT2}$, SINT2	72	48	5	External user interrupt #2
TINT, SINT3	76	4C	6	External timer interrupt
BRINT0, SINT4	80	50	7	BSP #0 receive interrupt
BXINT0, SINT5	84	54	8	BSP #0 transmit interrupt
TRNT, SINT6	88	58	9	TDM receive interrupt
TXNT, SINT7	92	5C	10	TDM transmit interrupt
$\overline{INT3}$, SINT8	96	60	11	External user interrupt #3
HINT, SINT9	100	64	12	HPI interrupt
BRINT1, SINT10	104	68	13	BSP #1 receive interrupt
BXINT1, SINT11	108	6C	14	BSP #1 transmit interrupt
BMINT0, SINT12	112	70	15	BSP #0 mis–alignment detection interrupt
BMINT1, SINT13	116	74	16	BSP #1 mis–alignment detection interrupt
—	120–127	78–7F	—	Reserved

The IFR and IMR registers are laid out as follows :

15–14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	BMINT1	BMINT0	BXINT1	BRINT1	HINT	$\overline{INT3}$	TXNT	TRNT	BXINT0	BRINT0	TINT	$\overline{INT2}$	$\overline{INT1}$	$\overline{INT0}$