1 TMS320C6201 Revision 1 to 2 Migration Document

TABLE OF CONTENTS

FIGURES

This document highlights the differences between revision 1 and revision 2 of TMS320C6201 and discusses the newly added functions and peripherals. It also lists the pin-out changes and the new pin assignment tables.

The previous version of this document was dated Aug 12, 1997.

New in this version of the Migration Document:

- RSV1 should now be pulled up instead of down
- HBE[1:0] changed polarity to HBE[1:0]_
- formerly unconnected pin C21 is now a test input RSV6 and should be pulled up
- formerly unconnected pin B22 is now a test input RSV7 and should be pulled up
- formerly unconnected pin A23 is now a test input RSV8 and should be pulled up

1.1 Overview

The following is the high level summary of the functional changes implemented in revision 2 of the 320C6201 as compared to revision 1:

- New functional modules added:
  - Host Port Interface (HPI)
  - 2 Multi-channel Serial Ports (MCSP0, MCSP1)
  - 2 Timers (TIMER0, TIMER1)
  - Interrupt Selector

- Existing functional modules that were modified:
  - DMAC (DMA Controller)
  - EMIF (External Memory Interface)
  - Data Memory Controller
  - PBUS (Peripheral Bus Controller)

- Changes to the External Memory Interface:
  - “External Controller” interface removed from EMIF
  - “Ready” signal added to “Asynchronous Memory Interface”
  - Burst functionality added to the SDRAM interface
  - External Address pins EA have been renamed
  - 3 high order EA pins have been removed

- Changes to the DMA Controller:
  - Increased number of channels from 2 to 4 plus 1 auxiliary channel
  - Increased the maximum transfer rate to one word/cycle
  - Added Split Mode
  - Added Auto-initialization between transfers
  - Added Automatic Data Sorting
  - Added Status Pins

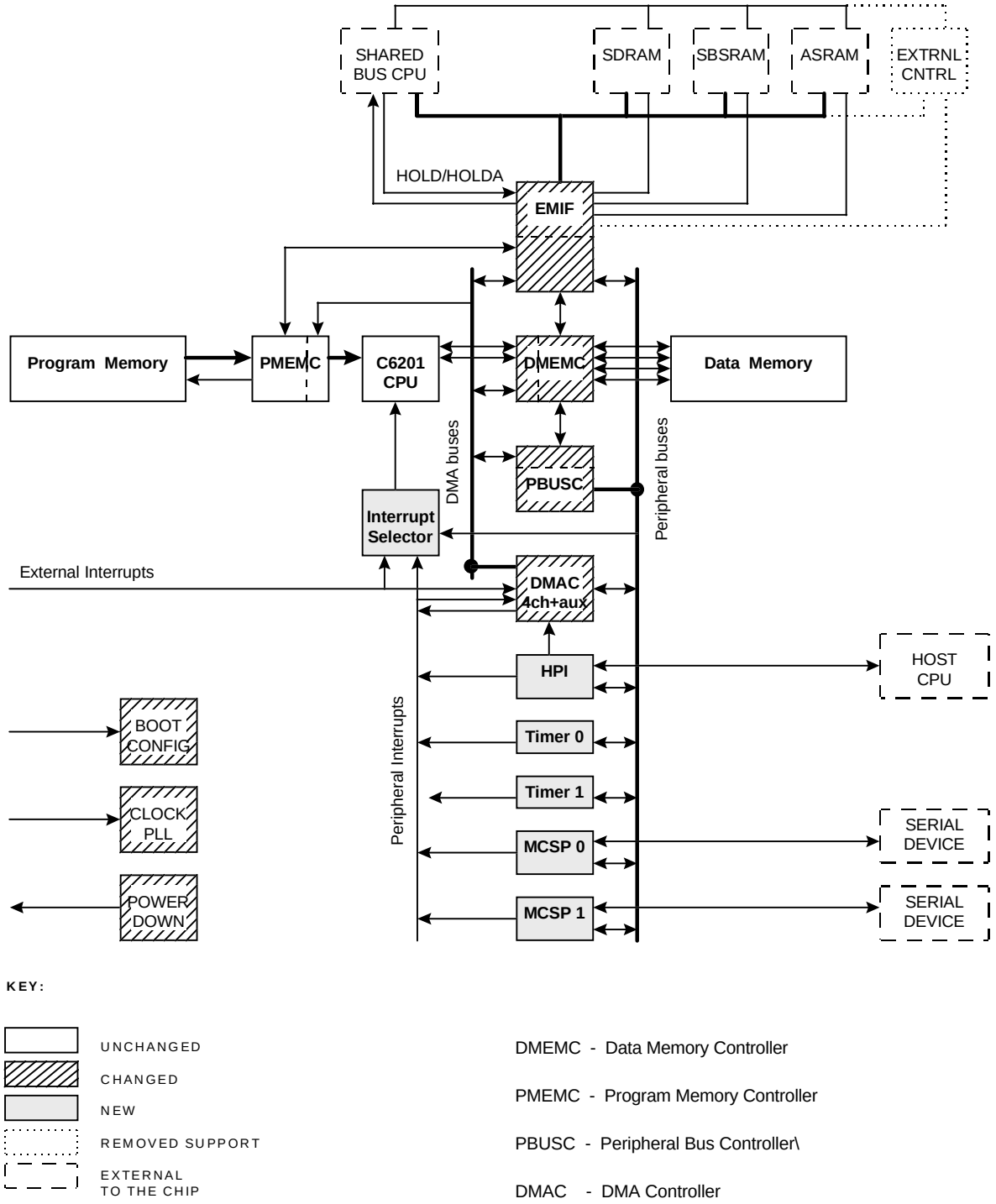
- Changes to the Data Memory Controller:
  - Added access to internal peripherals from CPU side A
  - Added high speed DMA access to internal data memory
  - Added programmable priority arbitration between DMA and CPU requests

- Changes to the Instruction Set:
  - The STP instruction was removed

- Other changes:
  - The 32Mbyte CE2 space has been split into 2 16Mbyte spaces - CE2 and a newly added CE3.
  - Removed the x2 clock mode
  - Changed the Power-down modes

- Existing functional blocks that were not changed:
  - CPU and PMEMC (Program Memory Controller / Cache)

Figure 1-1. Overview Of Revision 2 changes



1.2 Pin Assignment Changes

The following is the summary of the pin-out changes implemented in rev 2:

- New pins that replaced the revision 1 non-connected pins:

PD	Power-down mode 2 and 3
RSV4 BOOTMODE4 BOOTMODE3 BOOTMODE2 BOOTMODE1 BOOTMODE0	device boot configuration
TINP0	Timer0 / General Purpose Input
TOUT0	Timer0 / General Purpose Output
TOUT1	Timer1 / General Purpose Output
DMAC3 DMAC2 DMAC1 DMAC0	DMA Action Complete Status
CLKR1 CLKX1 DX1 FSR1 FSX1	MCSP1
CLKS0 CLKR0 CLKX0 DR0 DX0 FSR0 FSX0	MCSP0
RSV5	reserved

- New pins that replaced (discontinued) revision 1 pins:

TINP1	(ea20)	Timer1/General Purpose Input
CLKOUT1	(ssadv)	full speed clock
RSV3	(cmptb)	reserved
HINT_	(ea21)	

HCNTL1	(ha14)	
HCNTL0	(ha13)	
HHIL	(ha12)	HPI
HBE1	(ha11)	
HBE0	(ha10)	
HR/W_	(ha9)	
HAS_	(ha5)	
HCS_	(ha6)	
HDS1_	(ha8)	HPI
HDS2_	(ha7)	
HRDY_	(ea22)	
CE3_	(sdcke)	Memory Space 3 Enable
ARE_	(xreq_)	Async memory read strobe
SSCLK	(clkout1)	SBRAM clock
SDCLK	(clkout2)	SDRAM clock
CLKS1	(ha16)	MCSP1
DR1	(ha15)	MCSP1
C21	(ha1)	reserved input
B22	(ha2)	reserved input
A23	(ha3)	reserved input

- Non-connected revision 1 pins that remain non-connected in revision 2:

  K3
  J2
  H1
  H2
  G1
  E4
  R2

- Former pins that are now Non-Connects:

D21	(ha4)
D10	(hreq)
C9	(hr_)
A8	(hw_)
B8	(hack)

- Pins that retained the same function and location but changed names:

  EA2 (ea0)
  EA3 (ea1)
  EA4 (ea2)
  EA5 (ea3)
  EA6 (ea4)
  EA7 (ea5)
  EA8 (ea6)
  EA9 (ea7)
  EA10 (ea8)
  EA11 (ea9)
  EA12 (ea10)
  EA13 (ea11)
  EA14 (ea12)
  EA15 (ea13)
  EA16 (ea14)
  EA17 (ea15)
  EA18 (ea16)
  EA19 (ea17)
  EA20 (ea18)
  EA21 (ea19)

- Pins that changed location displacing a former non-connected pin:

  HOLDA

- Pins that changed location displacing another pin that was moved somewhere else:

  CLKOUT2

- Pins that retained the same function, location and name:

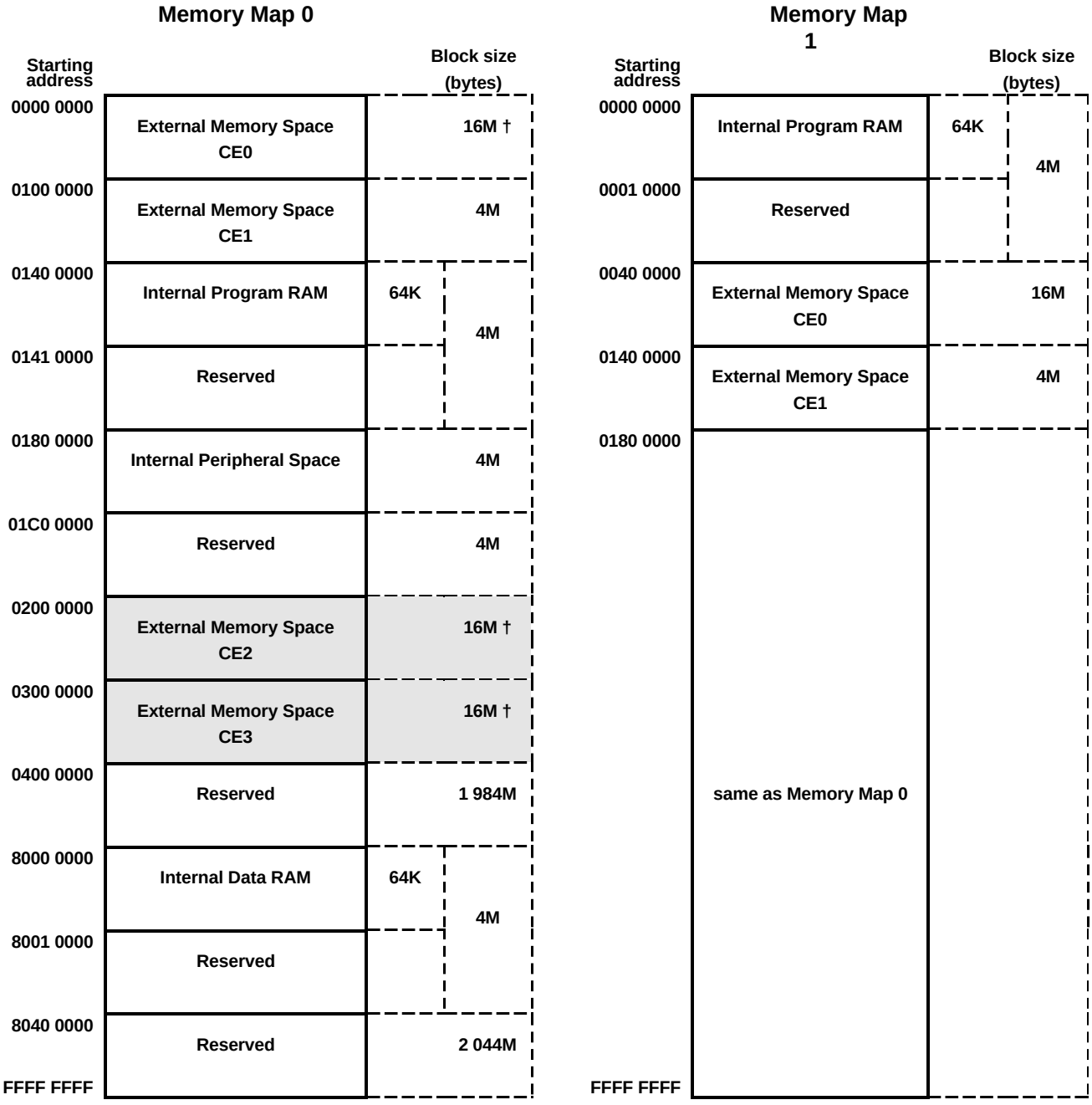
  The rest of original pins not mentioned above

1.3 Memory Map

- Number of CE spaces

  The number of CE spaces was increased from 3 to 4 by adding the CE3 space. The 32Mbyte revision 1 CE2 space was reduced to 16Mbytes, and the upper 16Mbytes are now assigned to CE3. All CE spaces are now less then or equal to 16Mbytes in size. The 20-bit external word address (which has been reduced by 3 bits from 23bits) corresponds to 4Mbyte external address reach for all memory types except SDRAM. The SDRAM uses multiplexed row and column address for a much greater external address reach (actual address reach depends on the SDRAM device).

Figure 1-2. Memory Map changes



Note: The Memory Map mode 0 or 1 is selected via the BOOTMODE[4:0] pins

† Maximum address range when using SDRAM (multiplexed address)

1.4 DMA

- Number of channels:

  The number of channels was increased from 2 to 4.

  Each channel can effectively become 2 channels if used in split mode (up to 6 channels total), where the write side transfers data from a peripheral to memory and the read side transfers data from memory to a peripheral. Since in each case the peripheral address remains unchanged, the DMA source ALU modifies the address for the receive channel and DMA destination ALU modifies the address transmit channel. Normally (in non-split mode i.e. memory to memory transfers), the source and destination addresses of each channel are both modified between each transferred data element.

  The Auxiliary Channel is provided to allow another peripheral (HPIF) access to the DMA bus to control data transfer between that peripheral and an arbitrary location of the Memory Map.

- Scope and Performance

  In addition to servicing the internal RAM and external memory, the DMA access was increased to include virtually the entire Memory Map. The DMA Controller can now transfer data through the Peripheral Bus Controller (PBUSC) to access any of the internal peripherals.

  The maximum transfer rate involving the internal RAM and EMIF have been increased to one transfer per cycle. Doubling of temporary data holding registers combined with separate read and write buses makes it possible to sustain simultaneous reads and writes after the startup overhead.

  The maximum transfer rate involving the internal peripherals is defined by the 3 cycle per read or write performance of the Peripheral Bus (PBUS).

In order to buffer resource contention due to CPU activity and to support synchronous burst memories, a single 8-word deep FIFO (existing inside the DMA Controller) can be assigned to a highly loaded DMA channel. Additionally, each DMA channel features a programmable priority bit that can override a CPU resource request inside the EMIF, DMEMC or PMEMC for faster processing of a high priority DMA requests.

The data sizes supported by the DMAC were expanded to include bytes and half-words in addition to word data transfers.

- Transfer Block Structure:

  The data block structure has been expanded to also include “Channel Sorting”. A previously monolithic block of data can be now composed of multiple frames of equal number of data elements. The sorting capability allows to group first elements of each frame into contiguous blocks of memory - Channels. The second elements of each frame are sorted into a separate Channel buffer, as are the third elements, etc.

- Address Control:

  In order to support channel sorting, the former DMA counter has been replaced with a composite frame/element counter, and source and destination addresses are now adjusted with frame/element indexes.

  The address is adjusted with an element index for each tick of the data element counter, except the one for the last element in the frame which is adjusted by the frame index. Each time the element counter counts out, the frame counter is decremented once. The block transfer completes when the frame counter counts out causing a DMA channel interrupt.

- Auto-initialization:

  While previously the CPU had to re-initialize and restart the DMA for each transferred block, the new auto-initialization mode can automatically recycle each DMA channel in between multiple blocks of continuous transfer.

  Once a block transfer is completed, another block transfer can be automatically started by reloading the transfer counter and source/destination addresses with values stored by the CPU in the Global Data Register File. The initialization values can remain the same for multiple blocks, or the CPU may change the reload values for the next block while the current block is being transferred.

- Synchronization, Control/Status:

  As before, the individual data transfers can be synchronized to specific events/interrupts for reads, writes or both. In addition to synchronizing data elements, frame synchronization can stop data transfer after the current frame in anticipation of an event identifying the start of the next frame.

  Element and frame synchronization elements can be triggered by interrupts or set/reset manually by accessing bits in the DMA Secondary Control Register.

  The synchronization status for read and write operations can be monitored by reading the DMA Secondary Control Register bits.

  The specific synchronization status may also be presented on the DMA status pins - one for each channel. The DMA Secondary Control Register can be used to specify which synchronization status information is presented on the DMA status pins.

- DMA Non-compatibility with rev1:

    1. On rev 1 START=11b bits in the DMA Primary Control Configuration start the DMA without auto-initialization. Same bit combination in rev2 starts the DMA in auto-initialization mode in which the DMA will restart itself after each transfer. If a program checks STATUS bits for 00b, it will never happen, because DMA won't stop until 00b is written to START bits.

    2. The DMA interrupt signals driving the Interrupt Selector do not automatically deassert after each interrupt. The Interrupt Service Routine is now required to clear the active interrupt status within the DMA to make it possible for the next DMA interrupt to be recognized.

    3. The DMA interrupt status will be active after a boot startup following reset since the boot transfer is performed by the DMA. The boot DMA interrupt status needs to be cleared by software before other DMA interrupts can be processed.

1.5 Host Port Interface

The revision 2 HPIF is a significantly changed module, similar to the C5x/C54x HPI but 16-bit wide. The following are the major differences:

- Architecture:

  While the revision 1 HPIF had a direct connection to the internal data memory, the revision 2 uses the DMA auxiliary channel to transfer the data to and from the Host. Under Host control, the DMA auxiliary channel stops the normal DMA activity in order to use the DMA buses to access any location within the Memory Map. In contrast to revision 1 HPIF's separate address and data ports, the revision 2 HPI features a single 16-bit bus that serves as data, address and control port.

- Performance:

  The new HPIF can transfer 16-bit data in 4 clock cycles as compared to a minimum of 10 cycles per transfer using the revision 1 HPIF. Removal of external rdy/ack handshaking improves the best case first access latency by another 7 cycles. HPIF latency can be minimized by prioritizing the auxiliary DMA channel above the other DMA channels.

- Host Control:

  While other internal peripherals are controlled by the CPU or DMA, all data transfers through the HPI are originated by an external Host Processor. A single 16-bit data port is used by a Host to transfer data, address and control commands. This is in contrast to the revision 1 HPI where after establishing control over the internal data memory (by req/ack handshaking signals), the Host would simultaneously use address and data terminals to transfer data. In revision 2, the Host uses a set of control signals to select access to one of HPIF's 3 internal registers - Control register, Address register or Data register. If Data register is chosen, additional control signals identify the data size and location of a particular data element within the 32-bit data register. Instead of separate read and write strobes, the new interface features a read/write, 3 strobes and a "ready" output.

- Address Control:

  Instead of presenting a dedicated address for each data element, all data transfers now start by the Host writing to the 32-bit HPIF word address register. Dedicated control lines instruct the HPIF if each data transfer is preceded by a corresponding address, or weather multiple data transfers follow a single address load which is then auto-incremented for each subsequently transferred data element.

- Data Sizes:

  While the revision 1 HPI could only transfer 16-bit data aligned on half-word address boundaries, the revision 2 HPI can transfer word, half-word or byte data anywhere within the Memory Map. Half-word and word data addresses are aligned with half-word and word address boundaries.

- Interrupts:

  While the revision 1 HPIF had no interrupt support of any kind, the revision 2 HPIF can generate interrupts to both the CPU and the Host by setting corresponding bits in the HPI Control Register.

- Initialization:

  The new Host Port Interface requires initialization of the HPIF Control Register to define the correct positioning of address half-words and data half-words within a word for subsequent data transfer. The revision 1 HPIF did not require any initialization to prepare for transfer.

- Reset:

  The address pins of revision 1 HPIF were used for device initialization during active Reset. In revision 2, other dedicated pins have been allocated for device initialization removing the need for sharing of the HPIF pins for that purpose.

1.6 Multi-channel Serial Ports (MCSP0, MCSP1)

The Multi-channel Serial Port (MCSP) is a new peripheral of which two are added to the C6201 revision 2 silicon. Both MCSP channels feature built-in hardware functions to directly support industry standard serial port protocols. Those include ST-bus standard, T1/E1 framer chips, MVIP-switching compliant devices, SPI devices, AC97 compliant devices and IOM2 compliant devices. It also directly interfaces to Codecs and AICs.

Below is a list of MCSP features, followed by a more detailed description of the most significant functions:

1. Multi-channel operation of up to 128 channels
2. Programmable Clock and Frame Sync pulses and their polarities
3. Clocks and Frame Syncs can be inputs or outputs
4. Supported word sizes are 8-,12-,16-,20-,24- and 32-bit wide
5. Data Frames may be divided into 2 phases with different word size in each phase.
6. u-law/A-law companding
7. Data packing using Frame Sync Ignore bits for efficient utilization of bandwidth
8. Internal Digital Loop-back between the Transmit and Receive modules for efficient s/w testing
9. Detection of Serial Port Exception Conditions
10. Configurable justification of received data
11. If not used as serial port, the MCSP pins can be configured as general purpose I/O
12. Programmable interrupt modes

- Multi-Channel Operation:

The Multi-Channel Serial Port directly supports communications applications where multiple processors may operate on different sections (channels) of a duplex multi-channel serial data stream. In addition to partitioning the data stream among multiple processors, the multi-channel features can also be used for multi-processor communication.

The multi-channel data stream may be composed of up to 128 words (channels) of data per frame (only single phase frames allowed). The 128 words can be arranged in up to 8 contiguous blocks of data, each containing 16 channels.

In a multi-processor system, each C6201 processor may be programmed to process a different group of channels by filtering out the unwanted blocks/channels during the receive operation and by transmitting out only the assigned blocks/channels. The processors in the system keep track of the individual blocks/channels by counting the transferred words since the most recent Frame Sync pulse.

Multi-channel operation is implemented by grouping of channels inside a frame in 16 channel blocks for the purpose of “masking out” portions of the frame. In a multi-processor application, one processor may want to ignore certain blocks in the frame, because another processor in the system may be assigned to those blocks. The maximum number of blocks in a single frame is 8, which corresponds to the maximum frame size of 128 words (channels). In multi-channel operation, individual blocks can be selected and individual channels can either be masked out or selected for transfer by the MCSP for all transmit or receive operations. The maximum number of individually selected words is 32 (2 adjacent blocks, each 16 words wide).

- Frame Synchronization:

All data transferred via the serial ports is organized in frames. The beginning of each frame is marked by a Frame Sync pulse. The first bit of a word in a the transfer appears on the DR or DX pin after 0,1 or 2 programmable shift clock periods following the Frame Sync pulse. The maximum frame size is 128 words for single-Phase frame and 256 words for a dual-Phase frame. The frame length (in words), word length (in bits) and the data bit delay following the Frame Sync are programmable. The period between consecutive frames is programmable as well.

The receive and transmit Frame Sync pins can be programmed to be an input or an output. The widths of the receive and transmit Frame Sync pulses are also programmable.

The words inside a frame can be grouped by Phases. Each Phase can be programmed for a number of words comprising that phase and the width of those words (in bits). There can be a maximum of 2 Phases in a frame.

In Multi-channel operation, the words inside frames can be organized in up to 8 blocks of 16 words each yielding a maximum of 128 words (channels). All words inside all blocks must be of the same size (one Phase only).

- Programmable Shift Clock:

The receive and transmit clocks, that shift the data in and out of the MCSP, can be driven by an external source or generated internally by the Sample Rate Generator. The input to the Sample Rate Generator can be selected to be driven either by the CPU clock or the external MCSP clock source pin (CLKS). The Sample Rate Generator divides down the input clock, as programmed by the user, to generate the receive and transmit clocks. The polarity of the external clock source can be programmed to be reversed before it enters the Sample Rate Generator.

The MCSP clocks, appearing on the transmit and receive clock pins, can be programmed to be either inputs or outputs. Their polarities are also programmable. The transmit and receive clocks shift the bits in and out of the MCSP receive and transmit modules.

- Clock Stop Mode:

The Clock Stop Mode can be used to connect to SPI compliant devices. Per SPI protocol, the master clock is used to drive both the input and output data to and from the SPI devices. The presence or absence of this clock effectively dictates transmission and reception of data between master(s) and slaves, in absence of dedicated receive and transmit Frame Sync pulses. The MCSP transmit section drives out the transmit clock to shift data in and out of the SPI slave device. The same clock is also looped back to the receive section of the master serial port to shift in the data transmitted out of the SPI device. The MCSP transmit Frame Sync output in this mode is used to enable the slave SPI device during active clock periods.

1.7 Timers (TIMER0, TIMER1)

Two 32-bit general purpose Timers, similar to the C3x/C4x Timers, were added in the revision 2 of the C6201 silicon. The Timers can be driven by the internal clock or by the external clock source through the timer pins. The timer pins (TINP0, TINP1, TOUT0, TOUT1) can also be used for general purpose I/O if not used for clocking of the Timers. With an internal clock, for example, the Timer can signal an external A/D converter to start the data conversion, or it can trigger the DMA controller to begin data transfer. With an external clock, for example, the timer can count external events and interrupt the CPU after a specified number of events has elapsed.

Three 32-bit memory mapped registers are used to configure and control the operation of each Timer. The Timer Control register determines the operation mode of the Timer, monitors the Timer status and controls the function of the TIM pin. The Timer Counter register contains the current value of the counter. The Timer Period register contains the number of timer input clock cycles to count before generating an interrupt and/or a pulse. This number controls the Timer output signal frequency. The Timer pulses appearing on a TOUT pin (if selected) can be programmed to be one clock wide or have a 50/50 duty cycle.

1.8 Interrupt Selector Module

8 new interrupt have been added (2 Timer, 2 DMA and 4 Serial Port). This brings the total # of interrupt sources to 16. The CPU, however, only supports 12 interrupts at a time. The Interrupt Selector module was created in order to map the 16 available sources into 12 possible interrupts recognizable by the CPU. The interrupt selector features 2 registers where desired mappings of interrupt sources to vectored interrupts may be assigned. A third register allows polarity configuration for the 4 available external interrupts. The interrupt configuration registers inside the Interrupt Selector Module are accessible by the CPU through the Peripheral Bus.

1.9 Start-up Configuration after Reset

The most significant startup configuration differences in revision 2 include addition of boot from HPI mode and changing the method by which external pins are used to initialize device configuration:

- Control Pins:

  While the revision 1 device startup configuration was defined by driving 13 HPI address pins during Reset, the revision 2 configuration is primarily controlled by software setting bits in control registers. The device configuration for Boot startup is still provided by external pins. The 5 dedicated Boot Mode pins are sampled on the rising edge of the Reset signal. After boot has commenced, the rest of the device is then configured by software.

- Memory Map:

  The 5 Boot Mode pins select one of two possible Memory Map modes - mode 0 and mode 1. As before, Memory Map 0 is used in conjunction with direct execution startup from external space CE0, while Memory Map 1 is associated with the boot from external memory space CE1 or HPI to internal program memory (followed by program execution from internal program memory, address 0). Both memory maps (0 and 1) have not changed with the exception of the new chip enable space CE3 added to both maps starting at address location 300 0000h.

- Direct Execution:

  As before, the direct execution commences from the external memory space CE0 starting at address 0h (Memory Map 0). The 5 Boot Mode pins assign the CE0 memory space type to either SDRAM, SBSRAM or Asynchronous Memory.

- Boot From Memory:

  In this mode, 16K of 32-bit words are automatically transferred (using DMA0) from external memory space CE1 to internal program memory starting at address 0. The fixed size of transferred data is in contrast to revision 1 boot where the length of transferred data could be selected from 8 available sizes. The width of the CE1 memory holding the boot source is now encoded in the 5 Boot Mode pins.

- Boot from HPI:

  This is a new boot mode added in revision 2. Following deassertion of the Reset signal, the CPU stays in reset while the rest of the chip is taken out of reset. The HPI can at that point write to any location within the Memory Map. Once the necessary program and data have been loaded, the Host sends the HPI interrupt to the CPU to take it out or reset. The CPU then starts program execution from the internal program memory at address 0h.

1.10 External Memory Interface (EMIF)

The EMIF has been extensively redesigned. Major changes include adding a fourth Chip Enable memory space (CE3 - can be assigned to ASRAM, SBSRAM or SDRAM), removing the External Controller interface, adding burst to the SDRAM interface, adding a discrete read strobe and “ready” signal to the Asynchronous Memory interface and choice of full or half clock for the SBSRAM interface. The burst performance (PMEMC or DMA accessing SBSRAM or SDRAM) has improved substantially. The External Memory Word address reach has been reduced to 20 bits, which corresponds to 4Mbytes (1Mword x 4 Byte Enables).

- EMIF Request Priority:

  The following is the new request priority order that determines which request is accepted by the EMIF in the case of multiple requesters simultaneously competing for access to external memory:

  1) HOLD (highest priority)
  2) SDRAM REFRESH
  3) DMEMC
  4) PMEMC
  5) DMA (lowest priority)

  Each DMA request can be accompanied with a priority override bit which bumps it up in priority to between DMEMC and SDRAM REFRESH, above all CPU requests.

  The burst requests originating from any requester can now be clumped together into blocks of 8 transfers that can not be interrupted by the normally higher priority DMEMC requests. This was done in order to preserve the continuity of program fetch packets from SDRAM.

- Asynchronous Memory Controller:

  The Asynchronous Memory access was reduced from minimum of 4 to 2 cycles per each read or write.

  An additional discrete read strobe was added to complement the output enable signal which stays active for the duration of multiple read cycles. The read strobe comes up between individual cycles in order to simplify the interface to external peripheral devices such as FIFOs.

  The external “ready” signal was added to allow a slow external device to extend a read or write cycle already in progress. The “ready” signal features 3 cycle pipeline and it may need to be asserted early by the external logic in order to be recognized in minimum number of cycles.

  A programmable “turn-around” time was built in between consecutive read and write cycles in order to prevent data bus contention by slow external devices.

- Synchronous Burst SRAM (SBSRAM) Controller:

  The Burst Advance signal was removed, since EMIF drives the correct address for each SBSRAM read and write cycle issued.

  The half-rate clock was added to accommodate slower SBSRAM devices without slowing down the DSP. The SBSRAM clock rate can now be chosen to run at full device speed or half the CLKOUT1 speed.

- Synchronous DRAM (SDRAM) Controller:

  The SDRAM interface has been significantly enhanced with new features and increased performance. The interface now supports burst read and write cycles in addition to the individual reads and writes. The SDRAM clock rate remains at half the instruction cycle speed for maximum performance of 1 transfer every 2 instruction cycles.

  As in revision 1, the SDRAM interface still features page boundary detection where page misses are automatically recognized causing bank deactivation and new row address to be strobed. Automatic bank deactivation is now also performed when switching between reads and writes, in advance of each refresh cycle and before the SDRAM MRS cycle after device reset.

  The SDRAM refresh requests are now buffered using a 2 bit counter in order to minimize data burst interruptions due to refresh. The SDRAM refresh controller will attempt to squeeze the refresh cycles in between the bursts, however, if the refresh count reaches 3, the burst will be interrupted and all pending refresh cycles will be serviced without any further delay.

  Following reset, the SDRAM controller now will automatically issue the MRS cycles to initialize the SDRAM devices. The Write Burst Length, Read Latency, Burst Type and Burst Length SDRAM initialization parameters are programmed to the fixed values of 1,3, Serial and 1 respectively. Other parameters that may change with SDRAM device type and speed are programmable inside the SDRAM Control Register as before.

  The SDRAM interface now features the internal address shifter that supports the popular SDRAM sizes and organizations. The shift amount can be programmed by the CPU in the EMIF Global Control Register. The shift amount can be also initialized at Reset by the 5 BOOT_MODE device initialization pins.

1.11 Electrical, Power Dissipation and Thermal changes

These characteristics have not changed significantly other then a small increase in power dissipation due to added new peripherals.

1.12 Pin Assignment Tables

The following pin assignment tables have been shaded according to how the pin assignments changes from rev1 to rev2.

NAME	PIN NO.	TYPE	DESCRIPTION
Clock / PLL			
CLKIN	C10	I	Clock Input
CLKOUT1 (ssadv_)[1]	AF22	O	Clock output at full device speed
CLKOUT2	AF20	O	Clock output at half of device speed
CLKMODE1	C6	I	Clock mode select
CLKMODE0	C5	I •	Selects whether the output clock frequency = input clock freq x4 or x1
PLLFREQ3	A9	I	PLL Frequency Range
PLLFREQ2	D11	I •	Selects one of 5 frequency ranges bounding the CLKOUT1 signal.
PLLFREQ1	B10	I •	CLKOUT1 frequency determines the 3-bit value for the PLLFREQ pins.
RSV2	C11	I	Reserved (pull-high with a dedicated 20KΩ resistor)
PLLV	D12	A †	PLL analog VCC connection for the low-pass filter
PLLG	C12	A †	PLL analog GND connection for the low-pass filter
PLLF	A11	A †	PLL low-pass filter connection to external components and a bypass capacitor
RSV5	C8	O	Reserved (leave unconnected, do not connect to power or ground)
JTAG Emulation			† A = Analog Signal (PLL Filter)
TMS	L3	I	JTAG test port mode select (features an internal pull-up)
TDO	W2	O/Z	JTAG test port data out
TDI	R4	I	JTAG test port data in (features an internal pull-up)
TCK	R3	I	JTAG test port clock
TRST-	T1	I	JTAG test port reset (features an internal pull-down)
EMU1	Y1	I/O/Z	Emulation pin 1, pull-up with a dedicated 20KΩ resistor
EMU0	W3	I/O/Z	Emulation pin 0, pull-up with a dedicated 20KΩ resistor
Control			
RSV0	T2	I	Reserved for testing, pull-up with a dedicated 20KΩ resistor
RSV1	G2	I	Reserved for testing, pull-up with a dedicated 20KΩ resistor
RSV3 (cmptb)	B9	I	Reserved. Pull-up with a 20KOhm resistor
RESET-	K2	I	Device Reset
NMI	L2	I	Non-Maskable interrupt
EXT_INT7	U3	I	External interrupts Edge driven (programmable edge)
EXT_INT6	V2	I	
EXT_INT5	W1	I •	
EXT_INT4	U4	I	
IACK	Y2	O	Interrupt acknowledge for all active interrupts serviced by the CPU
INUM3	AA1	O	Active interrupt identification number Valid during IACK for all active interrupts (not just external) Encoding order follows the Interrupt Service Fetch Packet ordering
INUM2	W4	O •	
INUM1	AA2	O •	
INUM0	AB1	O	
LENDIAN	H3	I	If high, selects Little Endian byte/half-word addressing order within a word If low, selects Big Endian addressing
PD-	D3	O	Power-down mode2 or 3 active if low

Key

REV2 PIN THAT HAS SAME FUNCTION AND LOCATION AS IT DID IN REV1
EXISTING PIN THAT CHANGED LOCATION DISPLACING A NON-CONNECTED PIN
NEW REV2 PIN THAT REPLACES A DISCONTINUED REV1 PIN (old signal name)

	PIN		

[1] CLKOUT1 is a supplemental CPU rate clock output. This pin provides a signal to drive external logic in addition to SSCLK (formerly CLKOUT1 on rev 1). SSADV_ is not used by rev 2. The user should tie the ADV_ inputs to the SBSRAMs inactive-high.

NAME	NO.	TYPE	DESCRIPTION

Host Port Interface (HPI)[2]

HINT- (ea21)	H26	O/Z	Host Interrupt (from DSP to Host)
HCNTRL1 (ha14)	F23	I	Host control - selects between Control, Address or Data registers
HCNTRL0 (ha13)	D25	I	Host control - selects between Control, Address or Data registers
HHWIL (ha12)	C26	I	Host halfword select - first or second halfword (not necessarily high or low order)
HBE1_ (ha11)	E23	I	Host byte select within word or half-word
HBE0_ (ha10)	D24	I	Host byte select within word or half-word
HR/W- (ha9)	C23	I	Host read or write select

HD15	B13	I/O/Z	Host Port Data (used for transfer of data, address and control)
HD14	B14	I/O/Z	
HD13	C14	I/O/Z	
HD12	B15	I/O/Z	
HD11	D15	I/O/Z	
HD10	B16	I/O/Z	
HD9	A17	I/O/Z	
HD8	B17	I/O/Z	
HD7	D16	I/O/Z	
HD6	B18	I/O/Z	
HD5	A19	I/O/Z	
HD4	C18	I/O/Z	
HD3	B19	I/O/Z	
HD2	C19	I/O/Z	
HD1	B20	I/O/Z	
HD0	B21	I/O/Z	

HAS- (ha5)	C22	I	Host address strobe
HCS- (ha6)	B23	I	Host chip select
HDS1- (ha8)	D22	I	Host data strobe 1
HDS2- (ha7)	A24	I	Host data strobe 2
HRDY- (ea22)	J24	O	Host ready (from DSP to Host)

Boot mode[3]

RSV4	A6	I	Reserved for testing, pull-down with a dedicated 20KΩ resistor
BOOTMODE4	D8	I	Boot mode
BOOTMODE3	B4	I	
BOOTMODE2	A3	I	
BOOTMODE1	D5	I	
BOOTMODE0	C4	I	

Key

REV2 PIN THAT HAS SAME FUNCTION AND LOCATION AS IT DID IN REV1
NEW REV2 PIN THAT REPLACES A DISCONTINUED REV1 PIN (old signal name)
NEW REV2 PIN THAT REPLACES A “NOT CONNECTED” REV1 PIN
EXISTING PIN THAT CHANGED LOCATION DISPLACING ANOTHER PIN (THAT MOVED SOMEWHERE ELSE)

NAME	PIN NO.	TYPE	DESCRIPTION

[2] Due to its limited functionality the rev 1 host port has been fully redesigned in rev 2 and is not compatible with rev 1.

[3] A method of boot configuration on rev 2 that uses fewer pins which are not shared with functional pins replaces the boot configuration method through the HA pins on rev 1.

EMIF - control signals common to all types of memory

CE3- (sdcke)[4]	AE22	O/Z	Memory space enables
CE2-	AD26	O/Z	
CE1-	AB24	O/Z	(enabled by bits 24 and 25 of the word address)
CE0-	AC26	O/Z	(only one asserted during any external data access)
BE3-	AB25	O/Z	Byte enable control (decoded from the 2 lowest bits of the internal address)
BE2-	AA24	O/Z	byte write enables for most types of memory
BE1-	Y23	O/Z	can be directly connected to SDRAM read and write mask signal (SDQM)
BE0-	AA26	O/Z	

EMIF - Address

EA21	J26	O/Z	External Address (word address)
EA20	K25	O/Z	
EA19	L24	O/Z	
EA18	K26	O/Z	
EA17	M26	O/Z	
EA16	M25	O/Z	
EA15	P25	O/Z	
EA14	P24	O/Z	
EA13	R25	O/Z	
EA12	T26	O/Z	
EA11	R23	O/Z	
EA10	U26	O/Z	
EA9	U25	O/Z	
EA8	T23	O/Z	
EA7	V26	O/Z	
EA6	V25	O/Z	
EA5	W26	O/Z	
EA4	V24	O/Z	
EA3	W25	O/Z	
EA2	Y26	O/Z	

Key

REV2 PIN THAT HAS SAME FUNCTION AND LOCATION AS IT DID IN REV1
NEW REV2 PIN THAT REPLACES A DISCONTINUED REV1 PIN (old signal name)
REV2 PIN THAT RETAINED THE SAME FUNCTION AND LOCATION BUT CHANGED NAME FROM REV1

[4] SDCKE is not used by rev 2. The user should tie the CKE inputs to the SDRAMs active-high.

NAME	PIN NO.	TYPE	DESCRIPTION
EMIF - Data			
ED31	AB2	I/O/Z	External Data
ED30	AC1	I/O/Z	
ED29	AA4	I/O/Z	
ED28	AD1	I/O/Z	
ED27	AC3	I/O/Z	
ED26	AD4	I/O/Z	
ED25	AF3	I/O/Z	
ED24	AE4	I/O/Z	
ED23	AD5	I/O/Z	
ED22	AF4	I/O/Z	
ED21	AE5	I/O/Z	
ED20	AD6	I/O/Z	
ED19	AE6	I/O/Z	
ED18	AD7	I/O/Z	
ED17	AC8	I/O/Z	
ED16	AF7	I/O/Z	
ED15	AD9	I/O/Z	
ED14	AD10	I/O/Z	
ED13	AF9	I/O/Z	
ED12	AC11	I/O/Z	
ED11	AE10	I/O/Z	
ED10	AE11	I/O/Z	
ED9	AF11	I/O/Z	
ED8	AE14	I/O/Z	
ED7	AF15	I/O/Z	
ED6	AE15	I/O/Z	
ED5	AF16	I/O/Z	
ED4	AC15	I/O/Z	
ED3	AE17	I/O/Z	
ED2	AF18	I/O/Z	
ED1	AF19	I/O/Z	
ED0	AC17	I/O/Z	

Key

REV2 PIN THAT HAS SAME FUNCTION AND LOCATION AS IT DID IN REV1

NAME	PIN NO.	TYPE	DESCRIPTION
EMIF - Asynchronous Memory control			
ARE- (xreq-)[5]	Y24	O/Z	Asynchronous Memory read strobe
AOE-	AC24	O/Z	Asynchronous Memory output enable
AWE-	AD23	O/Z	Asynchronous Memory write enable
ARDY	W23	I	Asynchronous Memory ready signal
EMIF - Synchronous Burst SRAM control			
SSADS-	AC20	O/Z	SBSRAM address strobe
SSOE-	AF21	O/Z	SBSRAM output enable
SSWE-	AD19	O/Z	SBSRAM write enable
SSCLK (clkout1)[6]	AD17	O/Z	SBSRAM clock
EMIF - Synchronous DRAM control			
SDA10	AD21	O/Z	SDRAM address 10 (separate for refresh)
SDRAS-	AF24	O/Z	SDRAM row address strobe
SDCAS-	AD22	O/Z	SDRAM column address strobe
SDWE-	AF23	O/Z	SDRAM write enable
SDCLK (clkout2)[7]	AE20	O/Z	SDRAM clock
EMIF - Bus arbitration			
HOLD-	AA25	I	Hold request from the host
HOLDA-	A7	O	Hold request acknowledge to the host

Key

REV2 PIN THAT HAS SAME FUNCTION AND LOCATION AS IT DID IN REV1
NEW REV2 PIN THAT REPLACES A DISCONTINUED REV1 PIN (old signal name)

[5] Due to difficulty of interfacing the external controller mode has been removed. This functionality has been replaced with AXRDY also supplying ready to asynchronous accesses.

[6] SSCLK like rev 1 CLKOUT1 is the clock for SBSRAMs. However, its default rate is ½ CPU clock.. In this way, it is pin-compatible.

[7] SDCLK is a ½ rate CPU clock used to drive SDRAMs just like in rev 1. In this way, it is pin compatible.

NAME	PIN NO.	TYPE	DESCRIPTION
Timers			
TOUT1	H24	O/Z	Timer 1 or general purpose output
TINP1 (ea20)	K24	I	Timer 1 or general purpose input
TOUT0	M4	O/Z	Timer 0 or general purpose output
TINP0	K4	I	Timer 0 or general purpose input
DMA Action Complete			
DMAC3	D2	O	
DMAC2	F4	O	DMA Action Complete
DMAC1	D1	O	
DMAC0	E2	O	
Multi Channel Serial Port 1 (MCSP1)			
CLKS1 (ha16)	E25	I	External clock source(as opposed to internal)
CLKR1	H23	I/O/Z	Receive clock
CLKX1	F26	I/O/Z	Transmit clock
DR1 (ha15)	D26	I	Receive data
DX1	G23	O/Z	Transmit data
FSR1	E26	I/O/Z	Receive frame sync
FSX1	F25	I/O/Z	Transmit frame sync
Multi Channel Serial Port 0 (MCSP0)			
CLKS0	L4	I	External clock source(as opposed to internal)
CLKR0	M2	I/O/Z	Receive clock
CLKX0	L1	I/O/Z	Transmit clock
DR0	J1	I	Receive data
DX0	R1	O/Z	Transmit data
FSR0	P4	I/O/Z	Receive frame sync
FSX0	P3	I/O/Z	Transmit frame sync
Reserved for test			
RSV6 (ha1)	C21	I	Reserved for testing, pull-up with a dedicated 20KΩ resistor
RSV 7 (ha2)	B22	I	Reserved for testing, pull-up with a dedicated 20KΩ resistor
RSV8 (ha3)	A23	I	Reserved for testing, pull-up with a dedicated 20KΩ resistor

Key

NEW REV2 PIN THAT REPLACES A “NOT CONNECTED” REV1 PIN
NEW REV2 PIN THAT REPLACES A DISCONTINUED REV1 PIN (old signal name)

NAME	PIN NO.	TYPE	DESCRIPTION

Remaining pins

	K3		
	J2		
	H1		
	H2		
	G1		
	E4		
	R2		

NAME	PIN NO.	TYPE	DESCRIPTION

former pins - now No-Connects

(ha4)	D21		
(hreq)	D10		
(hr_)	C9		
(hw_)	A8		
(hack)	B8		

Key

NEW REV2 PIN THAT REPLACES A “NOT CONNECTED” REV1 PIN
FORMER REV1 PIN THAT IS NOW NOT CONNECTED

3.3V Supply Voltage	PIN NO.
DVDD	B7
DVDD	A10
DVDD	A15
DVDD	A18
DVDD	D17
DVDD	A21
DVDD	A22
DVDD	G24
DVDD	G25
DVDD	H25
DVDD	J25
DVDD	L25
DVDD	N23
DVDD	R26
DVDD	T24
DVDD	U24
DVDD	W24
DVDD	AB26
DVDD	AC25
DVDD	AC22
DVDD	AC21
DVDD	AE21
DVDD	AC19
DVDD	AD18
DVDD	AE18
DVDD	AF17
DVDD	AD15
DVDD	AD13
DVDD	AD11
DVDD	AC10
DVDD	AF6
DVDD	AF5
DVDD	AC6
DVDD	AB4
DVDD	AB3
DVDD	Y4
DVDD	N3
DVDD	M3
DVDD	F3
DVDD	C1

2.5V Supply Voltage	PIN NO.
CVDD	D6
CVDD	D7
CVDD	A5
CVDD	B6
CVDD	B11
CVDD	B12
CVDD	A12
CVDD	C15
CVDD	A16
CVDD	A20
CVDD	C20
CVDD	D20
CVDD	K23
CVDD	M24
CVDD	N25
CVDD	AD14
CVDD	AF12
CVDD	AE12
CVDD	AC12
CVDD	AE8
CVDD	AD8
CVDD	T4
CVDD	T3
CVDD	U1
CVDD	P2
CVDD	M1
CVDD	K1
CVDD	F1
CVDD	H4
CVDD	E1

2.5V Supply Voltage	PIN NO.
CVDD	B2
CVDD	C3
CVDD	D4
CVDD	AE2
CVDD	AD3
CVDD	AC4
CVDD	D23
CVDD	C24
CVDD	B25
CVDD	AC23
CVDD	AD24
CVDD	AE25
CVDD	D9
CVDD	D14
CVDD	D18
CVDD	J23
CVDD	P23
CVDD	V23
CVDD	J4
CVDD	N4
CVDD	V4
CVDD	AC9
CVDD	AC13
CVDD	AC18

Note: PLLV and PLLG signals are not part of External Voltage Supply or Ground. See the CLOCK/PLL documentation for information on how to connect those pins.

Key

PIN THAT HAS SAME FUNCTIONALITY IN REV 1 & REV2

Ground	PIN NO.
VSS	A4
VSS	B5
VSS	C7
VSS	D13
VSS	C13
VSS	C16
VSS	C17
VSS	D19
VSS	E24
VSS	F24
VSS	G26
VSS	L23
VSS	M23
VSS	L26
VSS	N24
VSS	R24
VSS	T25
VSS	U23
VSS	Y25
VSS	AA23
VSS	AB23
VSS	AE23
VSS	AD20
VSS	AE19
VSS	AC16
VSS	AD16
VSS	AE16
VSS	AC14
VSS	AE13
VSS	AD12
VSS	AF10
VSS	AE9
VSS	AF8
VSS	AE7
VSS	AC7
VSS	AC5
VSS	AC2
VSS	AA3
VSS	Y3
VSS	V3

Ground	PIN NO.
VSS	V1
VSS	U2
VSS	N2
VSS	J3
VSS	G3
VSS	F2
VSS	G4
VSS	E3

Ground	PIN NO.
VSS	A1
VSS	B1
VSS	A2
VSS	C2
VSS	B3
VSS	AE1
VSS	AF1
VSS	AD2
VSS	AF2
VSS	AE3
VSS	A25
VSS	A26
VSS	B26
VSS	C25
VSS	B24
VSS	AF25
VSS	AF26
VSS	AE24
VSS	AE26
VSS	AD25
VSS	A13
VSS	A14
VSS	AF13
VSS	AF14
VSS	N1
VSS	P1
VSS	N26
VSS	P26

Note: PLLV and PLLG signals are not part of External Voltage Supply or Ground. See the CLOCK/PLL documentation for information on how to connect those pins.

Note: PLLV and PLLG signals are not part of External Voltage Supply or Ground. See the CLOCK/PLL documentation for information on how to connect those pins.

Key

PIN THAT HAS SAME FUNCTIONALITY IN REV 1 & REV2

APPENDIX A - Chip Level Preview

During every clock cycle the CPU can fetch a 256-bit Fetch Packet (8 instructions) and access 64 bits of data.

The CPU can execute 8 instruction per every clock cycle.

The size of the internal program memory is 64Kbytes (or 2048 Fetch Packets)

The size of the internal data memory is 64Kbytes (or 16Kwords).

DMA can access internal or external memory at a rate of 1 transfer per clock cycle (simultaneous reads and writes)

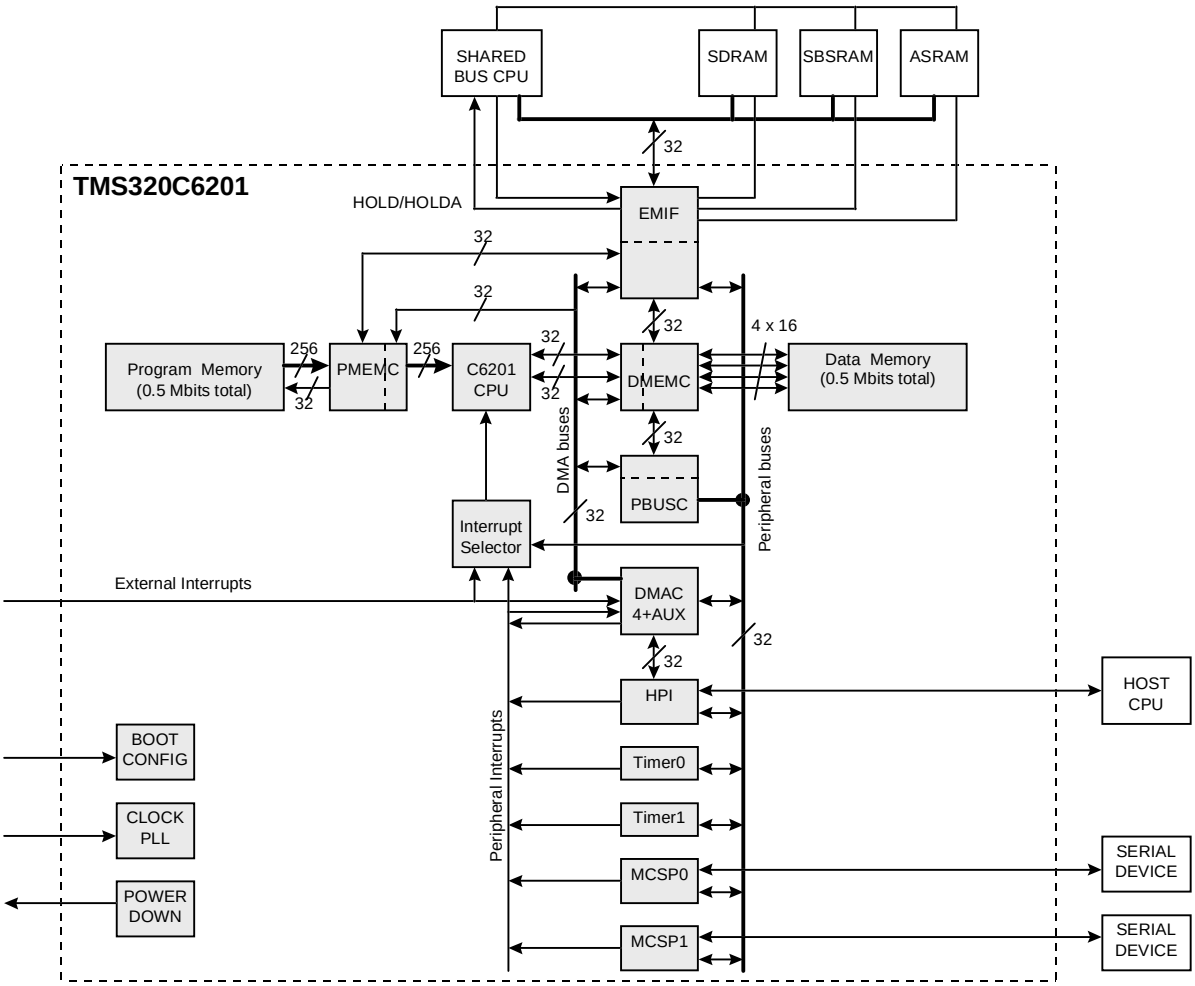
DMA can access internal peripherals at a rate of 1 read or write per 3 clock cycles (for peripheral source and destination - 6 cycles).

DMA supports hardware channel sorting.

Multi-Channel Serial Ports can operate at 100 MHz bit rate using internal or external clock source.

Interrupt Selector assigns 16 possible interrupt sources to maximum of 12 interrupts that the CPU can recognize at any given time.

External Memory Interface fully supports Synchronous DRAM, Sync-Burst SRAM as well as many types of Asynchronous Memories and I/O Devices.



DMEMC - Data Memory Controller

PMEMC - Program Memory Controller

PBUSC - Peripheral Bus Controller\

DMAC - DMA Controller

HPI - HOST PORT INTERFACE

APPENDIX B - CPU

The CPU can fetch a 256-bit wide (8 instructions) Fetch Packet every clock cycle.

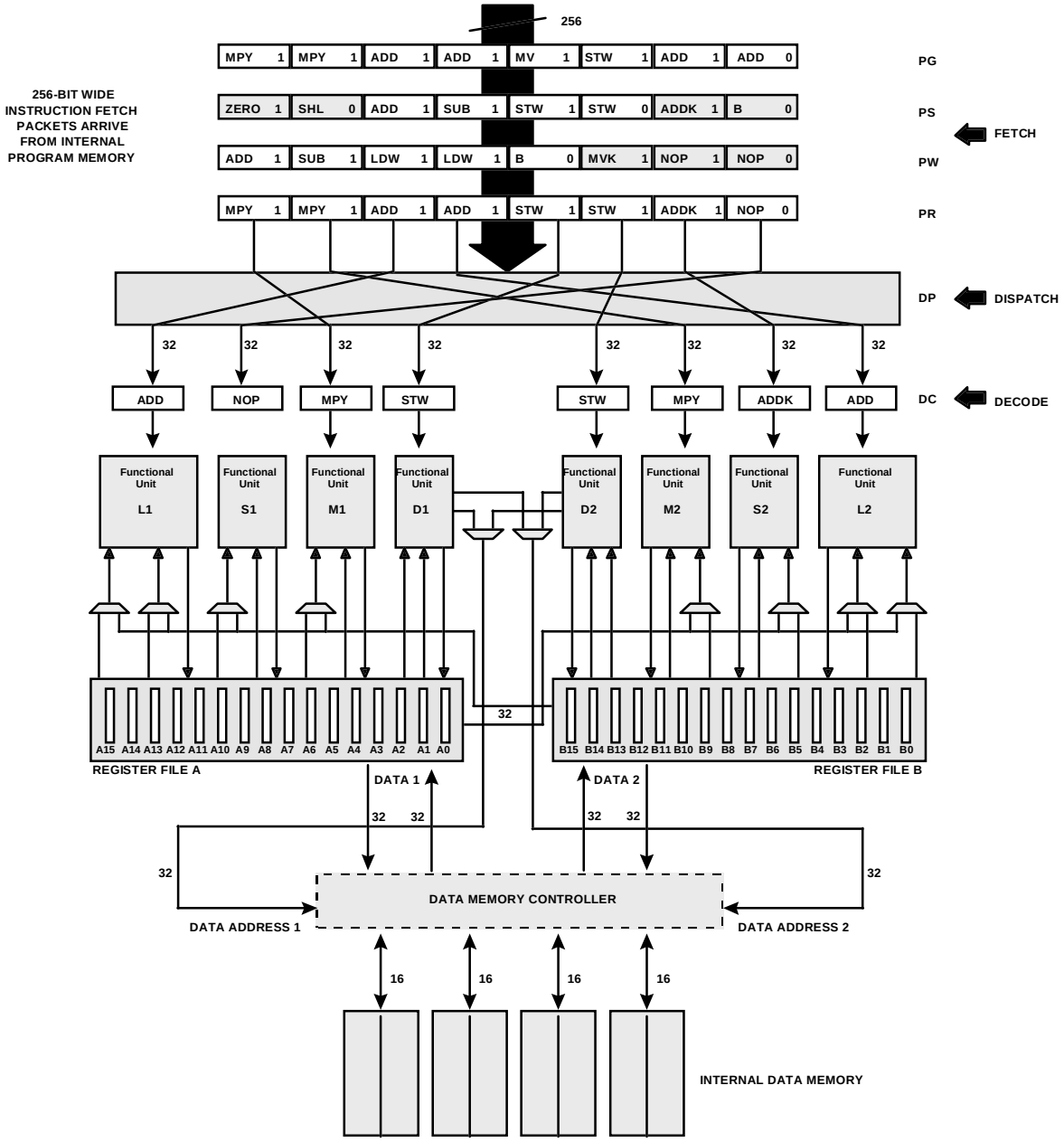
Most instructions have latency of 6 cycles (from fetch to execution, with some exceptions), however instructions can be completed at the rate of 8 per clock cycle.

The instructions in the Fetch Packet can execute in parallel or serially depending on the state of the P bit (lowest order bit in the instruction opcode). Parallel and serial instructions inside a Fetch Packet constitute Execute Packets. There can be from one to 8 Execute Packets inside a Fetch Packets.

The individual instructions composing Execute Packets are simultaneously dispatched to the decoders of their respective functional units. A new Fetch Packet is requested as soon as all Execute Packets composing the current Fetch Packet have been dispatched.

The 8 functional units are grouped in two CPU sides - A and B. Each side features 4 functional units coupled to 16 32-bit register for that side. Most functional units on either side of the CPU can simultaneously access any of the registers on the same side. The CPU also features a cross-path for transferring data between functional units and registers on the opposite sides of the CPU. Only one cross-load can be performed per instruction cycle.

The CPU has a register based architecture where all operations are performed on data residing in registers and where the results are also stored in registers. Two of the eight functional units can be used to transfer data between CPU registers and memory. Data can be transferred at a rate of 2 32-bit loads or stores every clock cycle.


256-BIT WIDE INSTRUCTION FETCH PACKETS ARRIVE FROM INTERNAL PROGRAM MEMORY
256
MPY 1 MPY 1 ADD 1 ADD 1 MV 1 STW 1 ADD 1 ADD 0
ZERO 1 SHL 0 ADD 1 SUB 1 STW 1 STW 0 ADDK 1 B 0
ADD 1 SUB 1 LDW 1 LDW 1 B 0 MVK 1 NOP 1 NOP 0
MPY 1 MPY 1 ADD 1 ADD 1 STW 1 STW 1 ADDK 1 NOP 0
PG
PS
PW
PR
FETCH
DP
DISPATCH
32
ADD NOP MPY STW STW MPY ADDK ADD
DC
DECODE
Functional Unit L1
Functional Unit S1
Functional Unit M1
Functional Unit D1
Functional Unit D2
Functional Unit M2
Functional Unit S2
Functional Unit L2
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
REGISTER FILE A
REGISTER FILE B
DATA 1
DATA 2
DATA MEMORY CONTROLLER
DATA ADDRESS 1
DATA ADDRESS 2
16
INTERNAL DATA MEMORY

APPENDIX C - External Memory Interface Preview

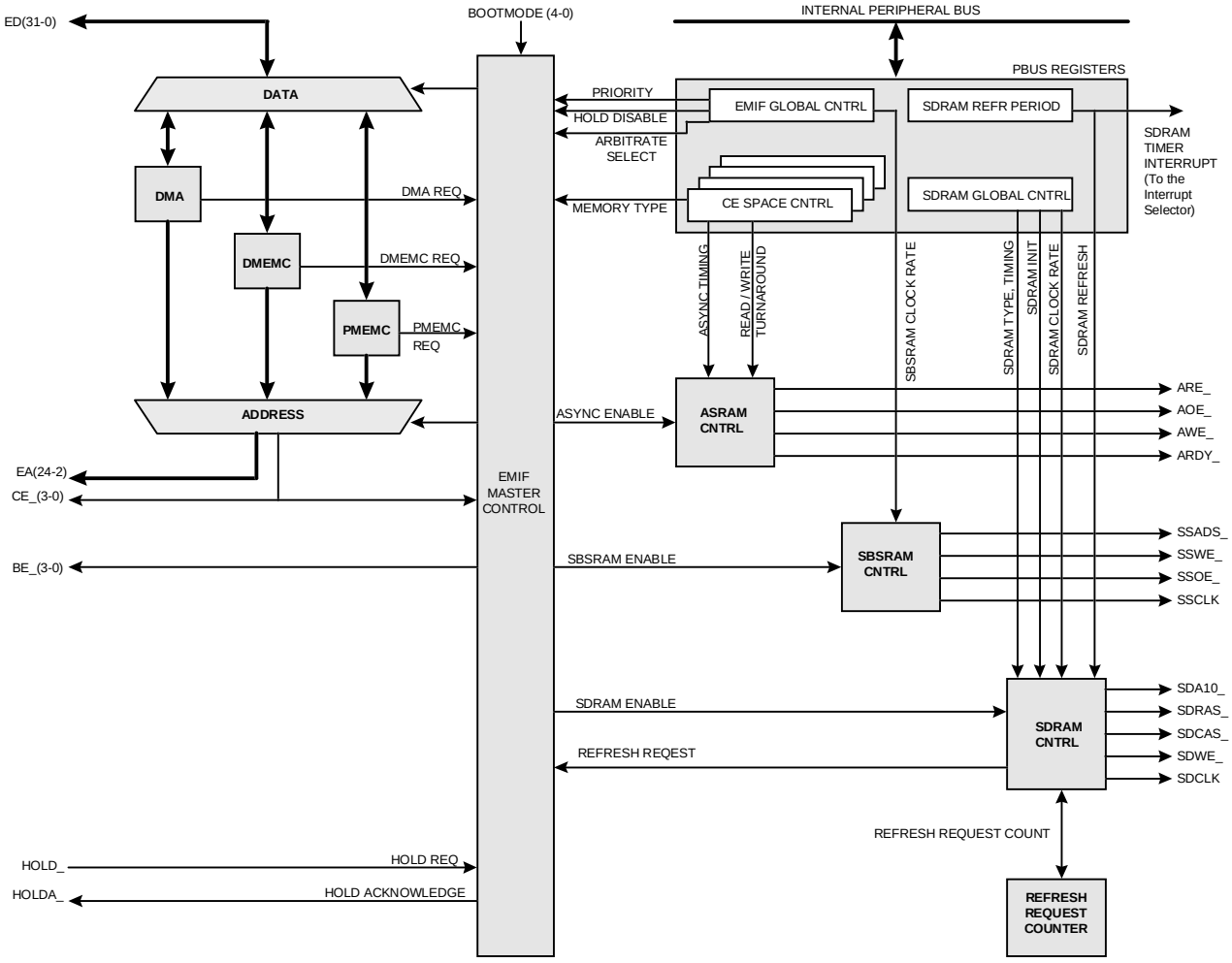
The External Memory Interface (EMIF) controls access to the external memory. EMIF supports 3 different types of memory - Asynchronous Memory, Sync-Burst SRAM (SBSRAM) and Synchronous DRAM (SDRAM). Requests to EMIF can originate from 5 different sources - PMEMC, DMEMC, DMA, Hold (by External Host) and SDRAM refresh. If two or more requesters simultaneously require access to external memory, EMIF uses the predetermined priority scheme to service higher priority requests before the lower priority requests.

The three types of memory supported by EMIF are each represented by a corresponding control block generating all the external control signals for a given memory. The three memory controllers are individually enabled by the EMIF Master Controller in response to requests for access to external memory. The Master Controller matches the request address to the corresponding CE Space Control register in order to decide which one of the memory controllers should be enabled for a given request.

The SDRAM refresh request has normally the lowest priority of all EMIF requesters. Refresh requests that are temporarily blocked by other requesters with higher priority, increment the 2-bit Refresh Request Counter. The counter is then decremented each time a pending refresh request is serviced. When 3 consecutive missed refresh cycles increment the counter to 3, the refresh becomes urgent and is raised in priority above the memory requests from CPU and DMA.

An arbitration priority bit in EMIF Global Control register determines whether a continuous lower priority access (burst) can be interrupted by a higher priority request immediately, or after a minimum of 8 transfers.

The priority of DMA requests to EMIF is normally lower then CPU requests (PMEMC and DMEMC). In some instances, however, the DMA controller can decide to grant the DMA higher priority then the CPU. In those cases, the DMA request will be accompanied by an “override bit” which will signal the EMIF Master Controller to service the DMA request ahead of any CPU requests.



DMEMC - Data Memory Controller

PMEMC - Program Memory Controller

PBUSC - Peripheral Bus Controller\

DMAC - DMA Controller

APPENDIX D - Host Port Interface Preview

The Host Port Interface (HPI) allows a host CPU to issue and control read and write cycles between the host and any C6x Memory Map location.

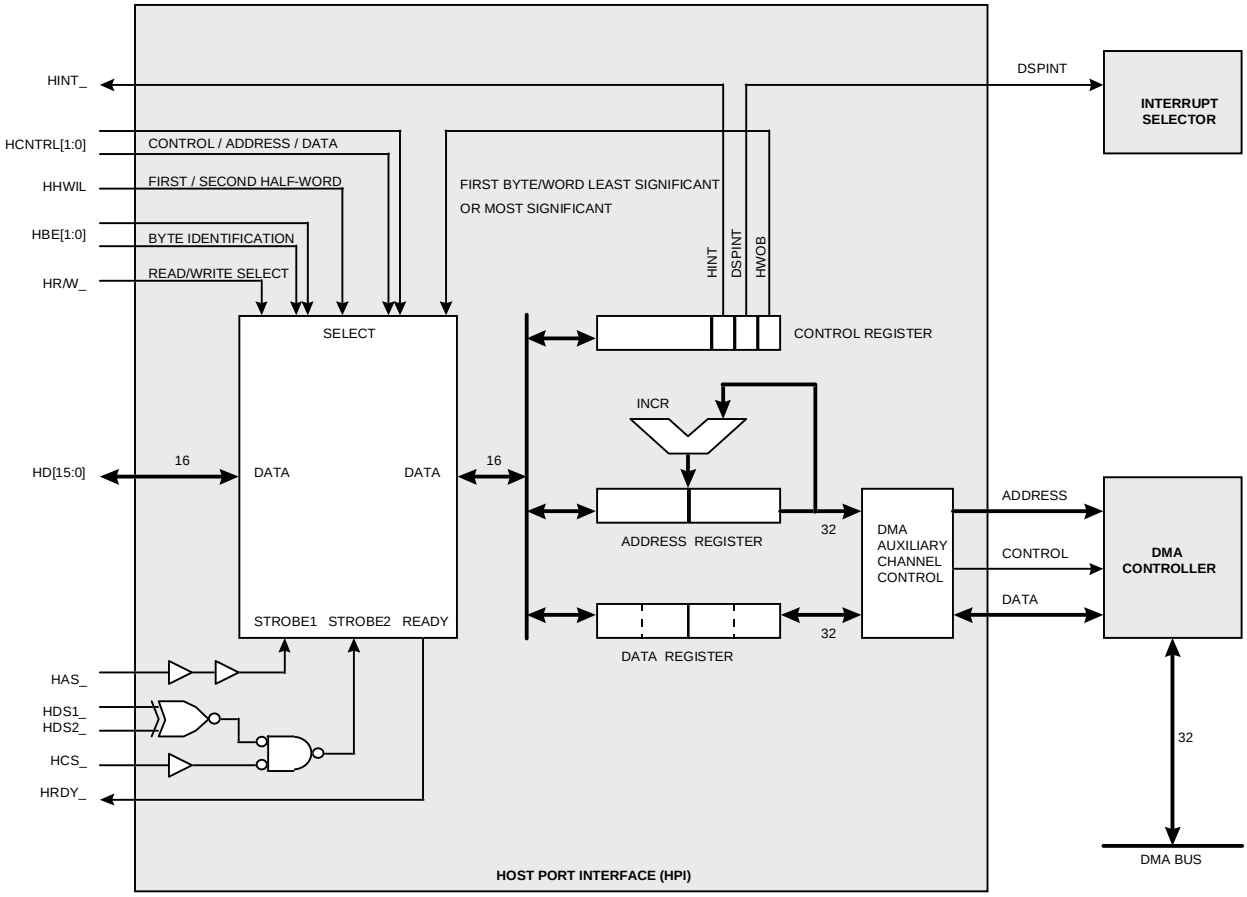
A single 16-bit data bus is used to access the HPI Control Register, Address Register and Data register. The 2 HCNTL signals are used to select which register is being accessed during all HPI read or write cycles.

The Address register is word aligned with bottom 2 bits always zero.

The host can access 8-, 16- or 32-bit data. The ordering (address) of bytes and half-words within the Data register is determined by the Control register HWOB bit to be increasing in the direction of the most significant bit or the least significant bit.

First and second half-word accesses to the Data register are identified by the HHWIL signal. During HPI read cycles the first half-word triggers the DMA Auxiliary Channel to load the Data register with source data. During the HPI write cycles the second half-word triggers the DMA Auxiliary Channel to move the contents of the Data register to its destination. The first and second half-word accesses can both involve either the low 16 bits of the Data register or the high 16 bits, depending on the state of the HBOB bit (in the Control register).

Each half-word read and write uses the two HBE signals to identify the specific bytes to be accessed within the half-words.


HINT_
DSPINT
INTERRUPT SELECTOR
HCNTRL[1:0]
CONTROL / ADDRESS / DATA
HHWIL
FIRST / SECOND HALF-WORD
FIRST BYTE/WORD LEAST SIGNIFICANT
OR MOST SIGNIFICANT
HBE[1:0]
BYTE IDENTIFICATION
HR/W_
READ/WRITE SELECT
HINT
DSPINT
HWOB
SELECT
CONTROL REGISTER
INCR
HD[15:0]
16
DATA
DATA
16
ADDRESS REGISTER
32
DMA AUXILIARY CHANNEL CONTROL
ADDRESS
CONTROL
DATA
DMA CONTROLLER
STROBE1 STROBE2 READY
DATA REGISTER
32
HAS_
HDS1_
HDS2_
HCS_
HRDY_
32
DMA BUS
HOST PORT INTERFACE (HPI)

APPENDIX E - DMA Controller Preview

The DMA Controller transfers data between any two Memory Mapped locations without interrupting the CPU.

Each DMA transfer consists of a read cycle from the source address to a temporary DMA register, followed by a write cycle from the temporary register to the destination address. The minimum latency for each transfer is 2 cycles, but following startup delay, reads and writes can occur simultaneously resulting in single cycle transfers.

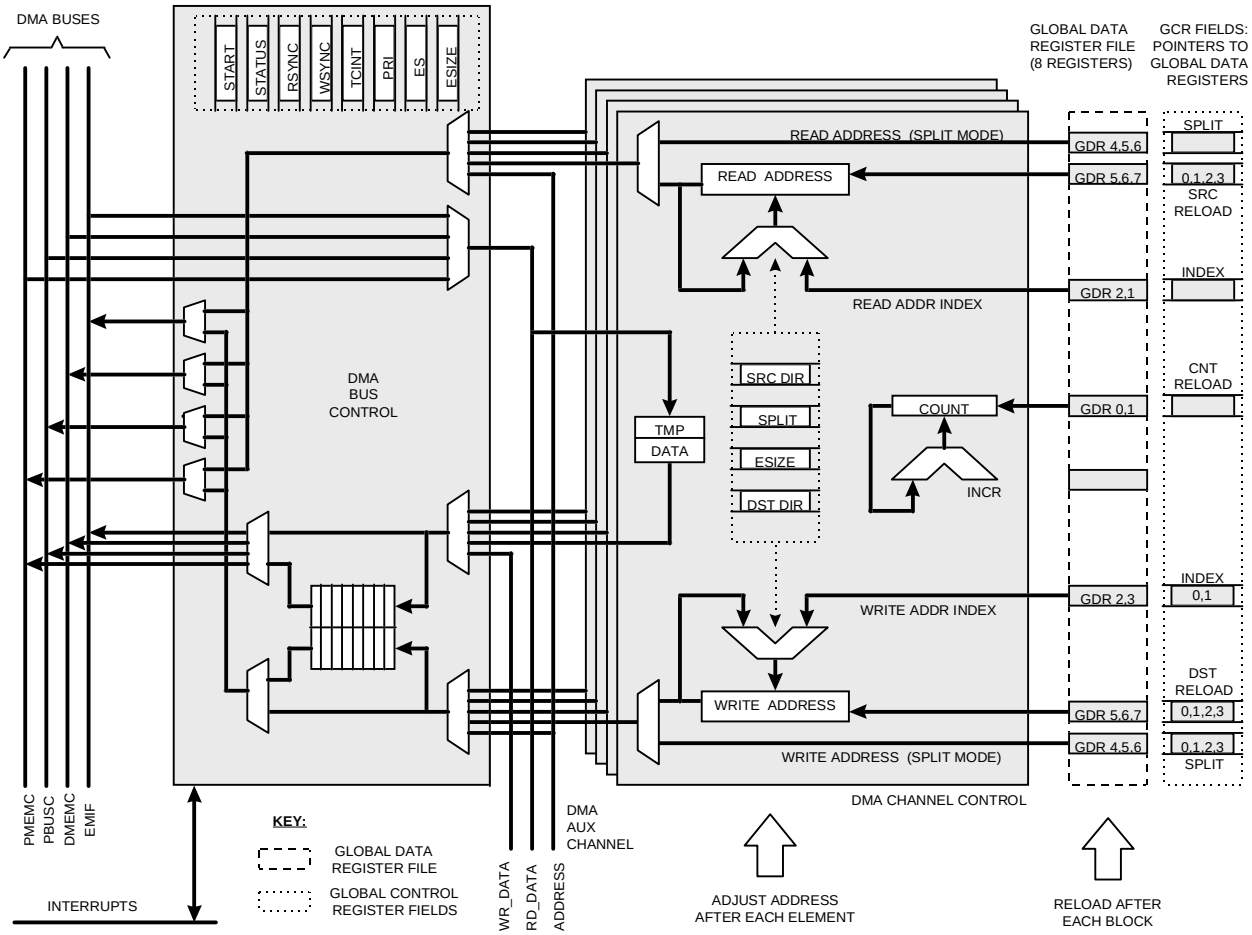
The individual DMA Channel Controllers update the source and destination addresses between every element of the transfer, referencing Global Data Register File registers for directions on how to update the addresses. Specific fields inside the DMA Global Control Register identify the assignment of the Global Data registers to the corresponding address modifiers.

The DMA Channel Controllers can also reload the read/write address and counter registers with initial/updated values at the beginning of every transfer block. The address/counter reload registers are identified by the corresponding fields inside the Global Control Register.

The DMA Channel Controllers request read and write cycles from the DMA Bus Controller which then prioritizes the individual requests before issuing DMA Bus Cycles to service them.

In addition to the 4 internal channels, the DMA Bus Controller can receive DMA Bus Cycle requests from an external source such as the DMA Auxiliary Channel inside the Host Port Interface peripheral (HPI).

The DMA Bus Controller features an 8-deep FIFO for buffering destination data/address pairs that follow a stalled write to a temporarily busy destination.

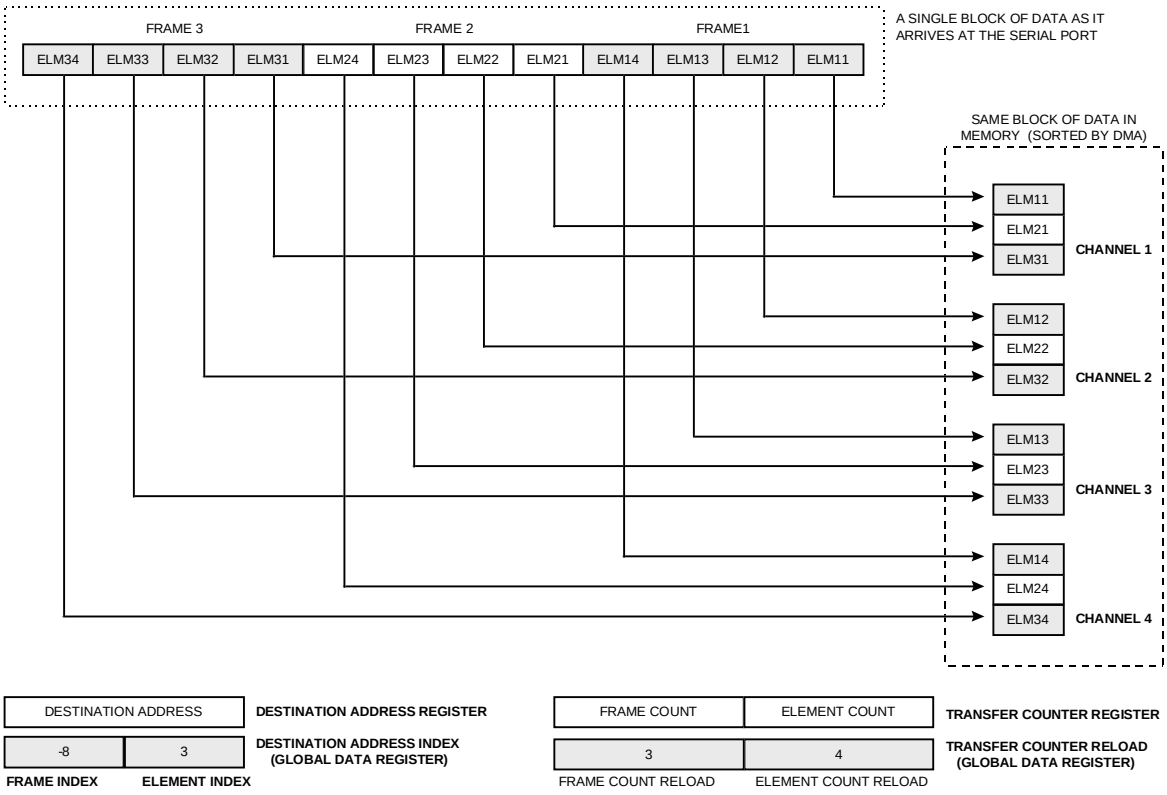

DMA BUSES
START
STATUS
RSYNC
WSYNC
TCINT
PRI
ES
ESIZE
GLOBAL DATA REGISTER FILE (8 REGISTERS)
GCR FIELDS: POINTERS TO GLOBAL DATA REGISTERS
READ ADDRESS (SPLIT MODE)
READ ADDRESS
READ ADDR INDEX
GDR 4,5,6
GDR 5,6,7
GDR 2,1
GDR 0,1
GDR 2,3
GDR 5,6,7
GDR 4,5,6
SPLIT
0,1,2,3
SRC RELOAD
INDEX
CNT RELOAD
INDEX
0,1
DST RELOAD
0,1,2,3
0,1,2,3
SPLIT
DMA BUS CONTROL
SRC DIR
SPLIT
ESIZE
DST DIR
TMP DATA
COUNT
INCR
WRITE ADDR INDEX
WRITE ADDRESS
WRITE ADDRESS (SPLIT MODE)
DMA CHANNEL CONTROL
PMEMC
PBUSC
DMEMC
EMIF
INTERRUPTS
KEY:
GLOBAL DATA REGISTER FILE
GLOBAL CONTROL REGISTER FIELDS
DMA AUX CHANNEL
WR_DATA
RD_DATA
ADDRESS
ADJUST ADDRESS AFTER EACH ELEMENT
RELOAD AFTER EACH BLOCK

APPENDIX F - DMA Channel Sorting Operation

Channel Sorting is a feature of the DMA Controller that enables automating data buffering according to the position of the data element within the transfer frame. For example, if an incoming block of data consisted of 3 frames of 4 data element each, the DMAC would place the first element of every frame (representing channel 1) into one contiguous buffer in destination memory. Similarly, the 2nd, 3rd and 4th data element of each of 3 frames would be also assembled into their own respective contiguous buffers representing channels 2, 3 and 4.

Channel Sorting is implemented by splitting the DMA Transfer Counter into its Element and Frame components as well as splitting the DMA Destination Address Index into its own Element and Frame components. At the start of the transfer, for each time the Transfer Element Counter is decremented, the Element Index is added to the destination address. The Element Counter will reach 0 at the end of the first frame at which time the Element Counter will be reloaded with the original value and the Frame Counter will be decremented by 1. Also, at the end of each frame, the destination address is incremented (or decremented) by the value stored in the Frame Index (instead the Element Index). When the Frame Count reaches zero, the DMAC can either reload the Transfer Counter to receive another block, or it can stop the transfer until restarted again by the CPU.

The DMAC can be programmed for transfer of non-structured blocks of data as well, by disabling the Channel Sorting mode.


FRAME 3
FRAME 2
FRAME1
A SINGLE BLOCK OF DATA AS IT ARRIVES AT THE SERIAL PORT
ELM34
ELM33
ELM32
ELM31
ELM24
ELM23
ELM22
ELM21
ELM14
ELM13
ELM12
ELM11
SAME BLOCK OF DATA IN MEMORY (SORTED BY DMA)
ELM11
ELM21
ELM31
CHANNEL 1
ELM12
ELM22
ELM32
CHANNEL 2
ELM13
ELM23
ELM33
CHANNEL 3
ELM14
ELM24
ELM34
CHANNEL 4
DESTINATION ADDRESS
DESTINATION ADDRESS REGISTER
-8
3
DESTINATION ADDRESS INDEX (GLOBAL DATA REGISTER)
FRAME INDEX
ELEMENT INDEX
FRAME COUNT
ELEMENT COUNT
TRANSFER COUNTER REGISTER
3
4
TRANSFER COUNTER RELOAD (GLOBAL DATA REGISTER)
FRAME COUNT RELOAD
ELEMENT COUNT RELOAD

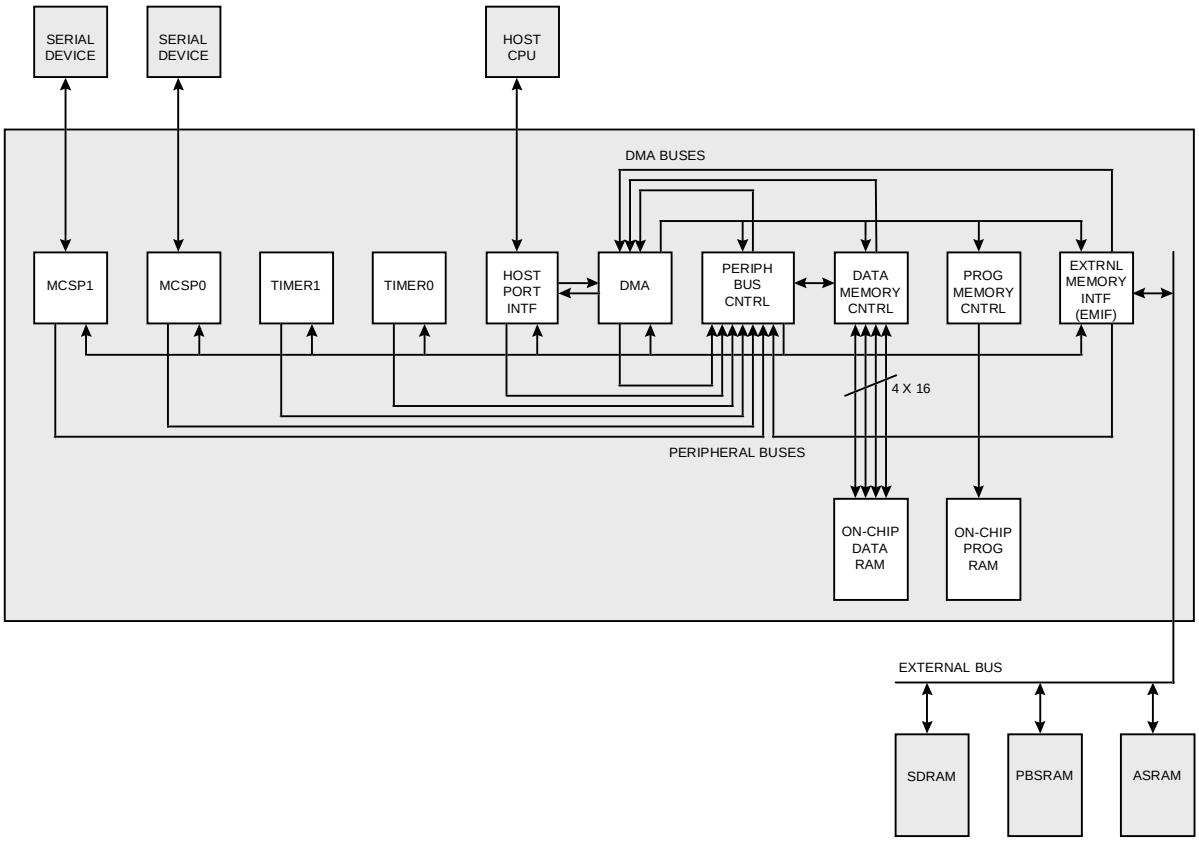
APPENDIX G - DMA And Peripheral Buses

DMA buses provide the DMAC (DMA Controller) a high performance path to the Internal Memory Controllers (DMEMC and PMEMC), the Internal Peripheral Controller (PBUSC) and the External Memory Interface (EMIF).

The maximum DMA transfer rate through the DMA buses is one transfer per clock cycle when accessing internal memory or external memory.

The maximum DMA transfer rate is one transfer per 3 clock cycles (each way) when accessing internal peripherals via the PBUSC (individual PBUS transfers always take 3 cycles or 6 cycles if both source and destination are internal peripherals).

The Host Port Interface (HPI) uses the DMA Auxiliary Channel to access any Memory Mapped location via the DMA and Peripheral buses.


SERIAL DEVICE
SERIAL DEVICE
HOST CPU
DMA BUSES
MCSP1
MCSP0
TIMER1
TIMER0
HOST PORT INTF
DMA
PERIPH BUS CNTRL
DATA MEMORY CNTRL
PROG MEMORY CNTRL
EXTRNL MEMORY INTF (EMIF)
4 X 16
PERIPHERAL BUSES
ON-CHIP DATA RAM
ON-CHIP PROG RAM
EXTERNAL BUS
SDRAM
PBSRAM
ASRAM