Engineer-to-Engineer Note EE-286

ANALOG DEVICES **Technical notes on using Analog Devices DSPs, processors and development tools**
Visit our Web resources http://www.analog.com/ee-notes and http://www.analog.com/processors or e-mail processor.support@analog.com or processor.tools.support@analog.com for technical support.

Interfacing SDRAM Memories to SHARC® Processors

Contributed by Aseem V., J. Jegadeesan and D. Venkataraman

Rev 5 – November 22, 2010

Introduction

Third and fourth generation high-end SHARC® processors have been specifically designed to simplify product development, speed up time to market, and reduce product costs for a variety of audio applications including audio/video receivers (AVRs), professional mixing consoles, and digital synthesizers.

In this EE-note, third generation refers to ADSP-21367, ADSP-21368, ADSP-21369 (hereafter referred to as ADSP-21368), and ADSP-2137x SHARC processors. Fourth generation refers to ADSP-2147x, and ADSP-2148x (hereafter referred to as ADSP-214xx) SHARC processors.

These processors, permit glue less interface to various SDRAM memories available, and allow embedded audio designers to take advantage of the most cost-effective memory technology. The result is lower overall system costs for applications that require lots of memory to store large amount of audio data such as lip sync and delay lines. These SHARC processors help audio product manufacturers reach the market with new, validated and tested, high-performance products at an unprecedented speed and with lower system costs.

This EE-Note discusses the SDRAM interface of the above mentioned devices, and differences between the SDRAM controllers (SDCs) on these processors. It also describes the shared memory feature, uniquely supported by ADSP-21368 processors. Furthermore, it discusses execution from external SDRAM on ADSP-2137x, and ADSP-214xx processors. Example code is provided in the associated `.ZIP` file. Finally, the EE-note also discusses SDRAM throughput optimization guidelines.

The SDRAM memories used in this particular document are MT48LC4M32B2, MT48LC8M16A2 and MT48LC16M16A2 from Micron Technology. Note that other SDRAM memories may be used as well.

External Port and SDRAM Controller (SDC)

The ADSP-21368, ADSP-2137x, and ADSP-214xx external port supports asynchronous memory devices like SRAM and flash, as well as synchronous memory devices like SDRAM. These processors have a dedicated on-chip programmable SDRAM controller (SDC), allowing a glueless interface to a variety of SDRAM memory devices.

The SDC supports a glueless interface with any standard SDRAM (32-Mbit, 64-Mbit, 128-Mbit, 256-Mbit, and 512-Mbit) with x4, x8, x16, and x32 configurations for ADSP-21368 and ADSP-2137x processors, with x4, x8, x16 for ADSP-214xx processors. The SDC can support up to 254 Mwords of SDRAM in four banks. Bank 0 can accommodate up to 62 Mwords, and banks 1, 2, and 3 can accommodate up to 64 Mwords each. The SDC includes timing options to support additional buffers between the processors and SDRAM devices. This allows the processor to handle the capacitive loads of large memory arrays. The processors support 16- and 32-bit-wide SDRAM devices.

The SDC for the different processors support the following features:

- Supports different page sizes that are programmable with column address widths
- Uses a programmable refresh counter to coordinate between varying clock frequencies and the SDRAM's required refresh rate
- Uses a separate pin (SDA10) that enables the SDC to pre-charge SDRAM before issuing an auto-refresh or self-refresh command while the external address pins of the other chip are being used by the AMI controller to access asynchronous memory
- Provides multiple timing options to suppo rt additional buffers between the processors and SDRAM
- Supports self-refresh mode
- Provides two SDRAM power-up options
- Supports 32-bit data access by the processor core
- Supports SDRAM memory access in DMA mode
- Provides a throughput of one word per SDCLK cycle for read operations in both DMA mode and core mode for ADSP-21368 and ADSP-2137x processors and a throughput of one word per two SDCLK cycles for all reads in ADSP-214xx processors.
- Provides a throughput of one and two words per SDCLK cycle for write operations using direct core and DMA access for ADSP-21368 processors, one word per SDCLK cycle for DMA access for ADSP-2137x processors and 1 word per two SDCLK cycles for all writes for ADSP-214xx processors.
- ADSP-2137x processors support optional *full page burst*. If the optional full page burst is selected in the SDCTL register, the SDC posts an address on the bus for every read and write. It does not burst, but instead, causes every start address of the burst to be interrupted with another start address. This mode is equivalent to no burst mode. However, if a non-SDRAM access is latched, the SDC interrupts the full page burst protocol by executing a burst stop command, and the specific page remains open. Not supported in ADSP-21368, ADSP-214xx processors.
- ADSP-2137x, and ADSP-214xx processors support *multibank operation* within SDRAM, which allows one page from each bank to remain open at any time and results in an effective page size of up to four pages. It reduces pre-charge and activation cycles by mapping opcode/data among different internal SDRAM banks. For ADSP-2137x processors this operation is supported for external bank 0 and 1 only. ADSP-214xx processors support this feature on all external memory banks.
- ADSP-2137x, and ADSP-214xx processors support execution of instructions from external memory bank 0.
- ADSP-21368 processor supports only the 'conflict' cache for internal memory code execution where it caches only those instructions whose fetching conflicted with access of a data operand from memory over the PM bus. For ADSP-2137x, And ADSP-214xx processors, the cache is enhanced such that the same cache behaves as instruction cache for external memory execution as well. Unlike the 'conflict'

cache for the internal memory code, every instruction that is fetched from external memory into the program sequencer is also simultaneously loaded into the cache. This feature enables faster external memory code execution throughput.

- ADSP-214xx processors support VISA (Variable Instruction Set Architecture) for instruction fetch from Bank 0, which drops redundant/unused bits within the 48-bit instruction to create more efficient and compact code. The program sequencer supports fetching these 16-bit and 32-bit instructions from SDRAM. This can reduce the size of code in a binary executable by 20 and 30% without any degradation of performance.
- ADSP-214xx processors support SIMD from external SDRAM memory which allows access to the complementary registers on the PEy unit in the normal word space (NW) unlike earlier SHARC processors which supported SIMD from internal memory. This improves performance since there is no need to explicitly load the complimentary registers as in SISD mode.
- ADSP-214xx processors support page-interleaving and bank interleaving
- ADSP-214xx processors support DMA from SPORT to external memory. Earlier the data sent to the SPORTs (say from an ADC), needed to be first transferred to internal memory, before finally transferring the data to external memory. Thus the new feature reduces the DMA cycles and allows internal memory to be used for other purposes.
- ADSP-2137x, and ADSP-214xx processors support single pre-charge command. For a page miss during reads or writes in any specific internal SDRAM bank, the SDC uses the single pre-charge command to close that bank. All other internal banks are untouched. For ADSP-2137x processors this command is supported only for bank 0 and 1. For ADSP-214xx processors, this command is supported on all external banks.
- ADSP-214xx processors support optional DQM signals, which are the I/O enable signals for SDRAM. This allows all the signals of the SDRAM to be connected directly to these processors.

The SDRAM controller on ADSP-21368 processors can run up to 166 MHz with silicon revision 0.2 and higher. For 0.0 and 0.1 silicon revision, the SDRAM controller can run up to 133 MHz only. The SDRAM controller on ADSP-2137x processors (for all silicon revisions) runs up to 133 MHz. The SDRAM controller on ADSP-2147x runs up to 133MHz and on ADSP-2148x runs up to 166 MHz.

ⓘ For ADSP-2148x processors, to run the SDRAM controller at 133 MHz, use SDRAM speed grade of 143 MHz or above. To run the SDRAM controller at 143 MHz, use SDRAM speed grade of 166 MHz or above. To run the SDRAM controller at 166 MHz, use SDRAM speed grade of 183 MHz or above.

Table 1 highlights differences between the SDC on the ADSP-21368, ADSP-2137x, and ADSP-214xx processors.

ADSP-21368 SDC	ADSP-2137x SDC	ADSP-214xx SDC
16/32 SDRAM	16/32 SDRAM	16 bit SDRAM
Only No burst mode (BL = 1) supported.	Optional full page burst mode also supported, this also includes the optional burst stop command.	Only No burst mode (BL = 1) supported.
Multibank operation not supported. Only one page can remain open at any time. When all the four external memory banks are connected to the SDRAM the effective page size is increased up to four pages.	Multibank operation within SDRAM is supported, allowing one page from each bank to remain open at any time and results in an effective page size of up to four pages. Since external bank 0 and 1 support multibank operation only, when all the four external memory banks are connected to the SDRAM the effective page size is increased up to ten pages.	Multibank operation within SDRAM is supported, on all the external memory banks. The maximum number of open pages is increased to sixteen pages.
Only data can be stored in external SDRAM.	Supports execution from external memory bank 0. Both data and code can be stored in external SDRAM.	Supports execution from external memory bank 0. Both data and code can be stored in external SDRAM. Additionally supports 16 bit packed instruction fetch from external back 0 using VISA (Variable Instruction Set Architecture) in order to save memory space.
Supports instruction cache for internal memory code execution	Supports instruction cache for internal and external memory code execution	Supports instruction cache for internal and external memory code execution
SDRAM does not support SIMD from external banks.	SDRAM does not support SIMD from external banks.	SDRAM supports SIMD from all external banks. The controller also supports SIMD mode from external memory. Two subsequent Normal words (16-bit packed data) are unpacked and loaded into primary and secondary data registers.
Supports only bank interleaving	Supports only bank interleaving	Supports page interleaving and bank interleaving
No support for data transfer from SPORT to SDRAM memory	No support for data transfer from SPORT to SDRAM memory	Supports direct DMA from SPORT to SDRAM memory
Shared memory access is supported	Shared memory access not supported	Shared memory access not supported
No support for DQM signals	No support for DQM signals	Supports DQM signals
No support for single pre-charge	Supports single pre-charge in external bank 0 and 1	Supports single pre-charge on all banks

Table 1. SDC features comparison table for ADSP-21368, ADSP-2137x, and ADSP-214xx processors

Hardware Interface

The processors support 16- and 32-bit SDRAM devices. In both cases, internal addressing is 32-bit addressing. The mapping of internal addresses to the external address pins depends on the row address width, column address width, and the X16DE bit setting. ADDR0 of the processor is not used for 32-bit SDRAM devices. For 32-bit SDRAM devices, connect ADDR1 of the processor to A0 of the memory device. For 16-bit SDRAM devices, connect ADDR0 of the processor to A0 of the memory device. Connect SDA10 of processor to A10 of the SDRAM.

Figure 1and Figure 2 show hardware interfaces with 32-bit SDRAM and 16-bit SDRAM devices, respectively. For details, refer to the ADSP-21369 EZ-KIT Lite® evaluation board design, which has a 32-bit SDRAM interface. Refer to the ADSP-21375 EZ-KIT Lite evaluation board design, which has a 16-bit SDRAM interface.

The ADSP-21368 and ADSP-21375 processor’s SDC do not support the SDRAM’s DQM signals, which are the I/O enable signals for the SDRAM device. ADSP-214xx processors support the DQM signals. The DQM signal of these processors should be connected to all the DQM signals of the SDRAM device. Refer to ADSP-2147x/ADSP-2148x EZ-Board® evaluation board design to see how the DQM signals are connected.

For write operations, data is written to memory if the corresponding DQM signals are registered logic low. If the DQM signal is registered high, the corresponding data inputs are ignored. For memory read operations, valid data is driven on data lines if the corresponding DQM signals are registered low; otherwise, the data lines are in a high-impedance state. Also please note that, the DQM latency in the read cycle is two clock cycles regardless of /CAS latency. In other words, the DQM signal is tri-stated 2 cycles after the read command. Since ADSP-21368 and ADSP-2137x processors do not provide DQM support, hard-wire these signals to digital ground on the board to enable the data I/O signals for memory read or write operations.

Some SDRAM devices (such as the W9864G2GH-7 from Winbond Electronics and the K4S643232H from Samsung Electronics) require their DQM signals to be driven logic high during power-up initialization. These SDRAMs can be interfaced as is with ADSP-214xx processors. For these processors the DQM pin on the SDC is driven high during power-up and till the SDRAM controller is initialized in the software. For connecting these SDRAMs to ADSP-21368 and ADSP-2137x processors, use the processor’s FLAG pin to drive the SDRAMs high during power-up as described in their data sheets.

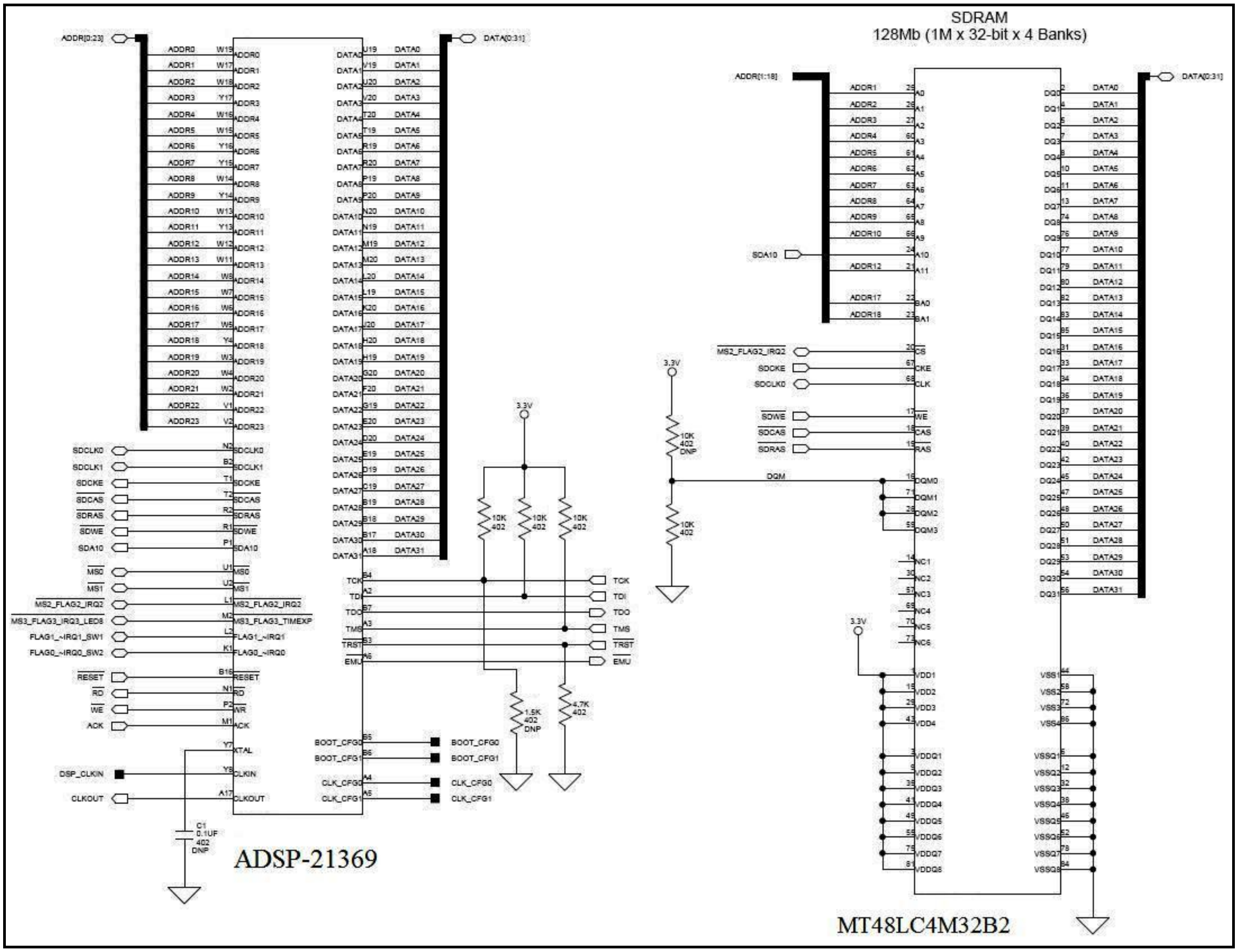


Figure 1. Address mapping and other signal connections for a 32-bit SDRAM device

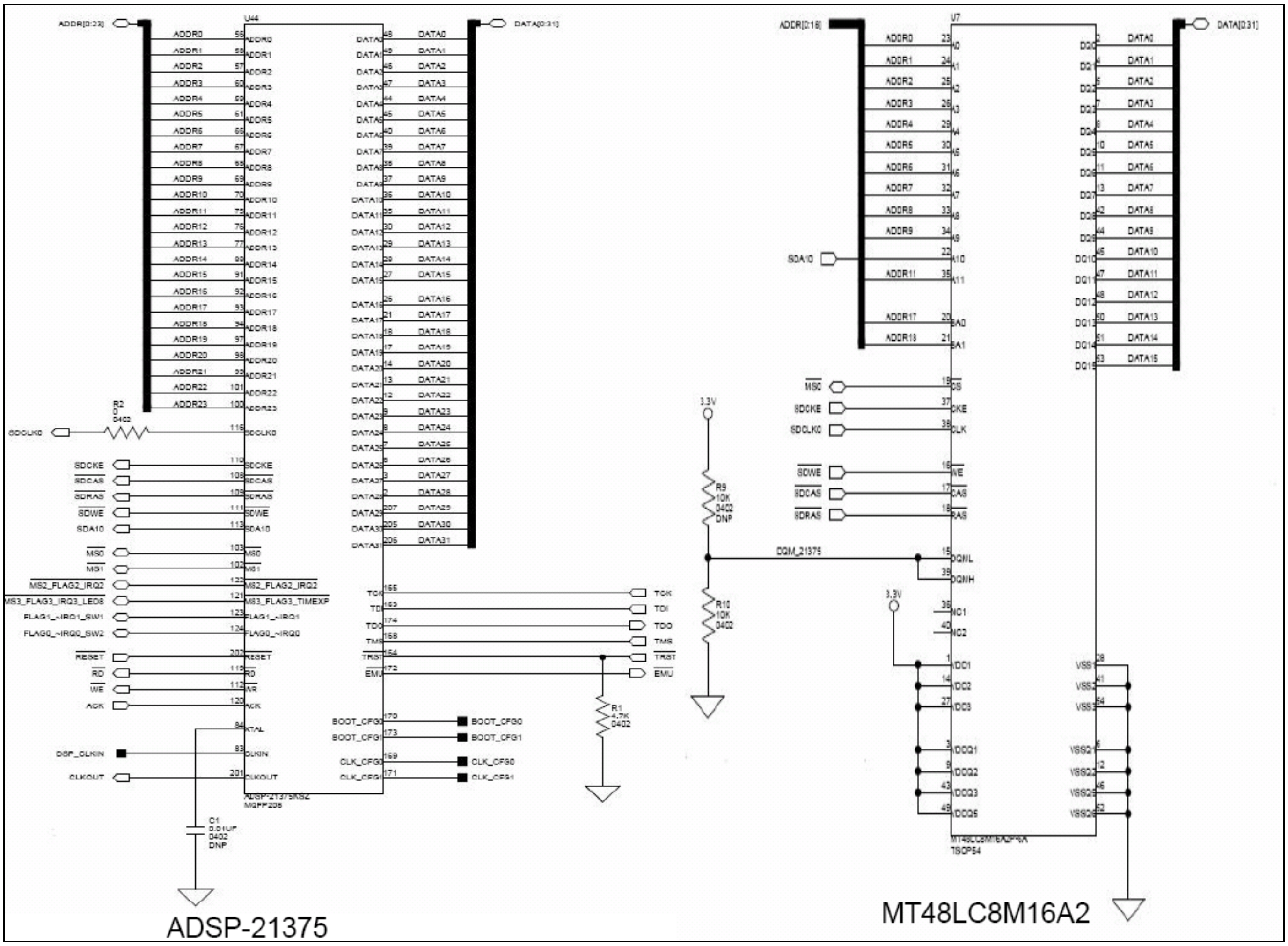


Figure 2. Address mapping and other signal connections for 16-bit SDRAM

The SDRAM can be mapped to any of the processor’s four external memory banks. Each memory bank has its memory select (`MSx/`) line, which is used to drive the memory device’s chip select (`CS/`) signal. For ADSP-21368 and ADSP-2137x processors the external memory address space (Table 2) is supported in normal word addressing mode only. ADSP-2137x processors supports external memory execution from external memory Bank 0 and (Table 3) shows the memory map. Note that ADSP-214xx processors also support short word addressing (Table 4) (VISA mode) for instruction fetch from the external memory Bank 0 apart from normal word addressing.

Bank	Size	Address Range
Bank 0	62 Mwords	0x0020 0000 – 0x03FF FFFF
Bank 1	64 Mwords	0x0400 0000 – 0x07FF FFFF
Bank 2	64 Mwords	0x0800 0000 – 0x0BFF FFFF
Bank 3	64 Mwords	0x0C00 0000 – 0x0FFF FFFF

Table 2. External memory address space for SDRAM memories

Access Type	Size	Address Range
ISA (NW)	8 Mwords	0x0080 0000 – 0x00FF FFFF

Table 3. External Bank 0 Instruction Fetch for ADSP-2137x processors

Access Type	Size	Address Range
ISA (NW)	4 Mwords	0x0020 0000 - 0x005F FFFF
VISA (SW)	10 Mwords	0x0060 0000 - 0x00FF FFFF

Table 4. External Bank 0 Instruction Fetch for ADSP-214xx processors

For a 16-bit wide memory, the external port interface effectively translates the Logical Addresses in the range 0x20 0000 – 0x5F FFFF to Physical Addresses in the range 0x60 0000 – 0x11F FFFF, when accessing 48-bit instructions in legacy, normal word mode (ISA) encoding from external memory. The external port performs three accesses to form one 48-bit word before forwarding it to the instruction fetch data unit. Note that external port interface passes the addresses in the range 0x60 0000 – 0xFF FFFF as is to external memory. As in the previous case, the external port accesses three short words to return a 48-bit word to the IAB for each access requested by the sequencer. The short words for a VISA section of code are packed in such a way that lowest of the addresses pertaining to a given instruction has the most significant short word of that instruction and the highest address has the least significant short word.

ⓘ The packed instruction when fetched in VISA operation is internally rotated before it reaches the instruction alignment buffer and cache. However, if this instruction is fetched in normal word ISA operation, this rotation does not happen and instruction will be cached without rotation. Eventually if the cache gives out the instruction to the processor, this would be a corrupted instruction. To avoid this, codes should not be placed in both the following ranges 0x5F FFFD-0x5F FFFF and 0x60 0000- 0x60 0008. Place code only in one of the above ranges.

Setting Up the SDC

Follow these steps to properly configure the SDRAM controller.

Configure the Core-clock-to-SDRAM-clock ratio

The SDRAM clock (SDCLK) is derived from the core clock (CCLK). Five fixed ratios are supported: 1:2.0, 1:2.5, 1:3.0, 1:3.5, and 1:4.0. To obtain a desired SDCLK, configure the PLL for the appropriate CCLK frequency. The DIVEN bit should be set while setting the CCLK-to-SDCLK ratio bit in PMCTL register. Listing 1 shows how to set up the CCLK -to-SDCLK ratio in the PMCTL register.

```
Set_CoreCLK_SDRAMClk_Ratio:

     // Set the Core clock to SDRAM clock ratio
     ustat1 = dm(PMCTL);
     #ifdef CCLK_SDCLK_RATIO_2_0
           bit clr ustat1 SDCKR2 | SDCKR2_5 | SDCKR3 | SDCKR3_5 | SDCKR4;
           bit set ustat1 SDCKR2 | DIVEN;
     #endif
     #ifdef CCLK_SDCLK_RATIO_2_5
           bit clr ustat1 SDCKR2 | SDCKR2_5 | SDCKR3 | SDCKR3_5 | SDCKR4;
           bit set ustat1 SDCKR2_5 | DIVEN;
     #endif
     #ifdef CCLK_SDCLK_RATIO_3_0
           bit clr ustat1 SDCKR2 | SDCKR2_5 | SDCKR3 | SDCKR3_5 | SDCKR4;
           bit set ustat1 SDCKR3 | DIVEN;
     #endif
     #ifdef CCLK_SDCLK_RATIO_3_5
           bit clr ustat1 SDCKR2 | SDCKR2_5 | SDCKR3 | SDCKR3_5 | SDCKR4;
           bit set ustat1 SDCKR3_5 | DIVEN;
     #endif
     #ifdef CCLK_SDCLK_RATIO_4_0
           bit clr ustat1 SDCKR2 | SDCKR2_5 | SDCKR3 | SDCKR3_5 | SDCKR4;
           bit set ustat1 SDCKR4 | DIVEN;
     #endif
     dm(PMCTL) = ustat1;

     RTS;
```

Listing 1. Configuring the CCLK-to-SDCLK ratio

Configure the Processor's PLL

As mentioned before, SDRAM clock (SDCLK) is derived from core clock (CCLK). Program the PLL correctly for the desired CCLK and the desired SDCLK. There should be at least one core idle cycle between setting the DIVEN bit in the PMCTL register and putting the PLL in bypass mode. When writing into the PMCTL register to set or clear the bypass mode, clear the DIVEN bit; this ensures correct PLL set up. Listing 2 shows the suggested PLL configuration code. Disable SDCLK output before re-programming the PLL and re-enable it after completing the PLL configuration.

Map SDRAM to one of the external memory banks

Map SDRAM to one of the four external memory banks by setting appropriate BxSD bits of external port control (EPCTL) register. Refer to Listing 3.

Configure the SDC

This consists of two steps. The first step configures the SDRRC register with the correct value of the refresh count (RDIV) and sets up the SDRAM read optimization. The second step writes various control parameters to the SDRAM control (SDCTL) register. These are mainly the timing requirements of SDRAM device. The SDCLK period determines the values of these various timing requirements in addition to the RDIV value. Set the power-up sequence bit (SDPSS) in the processor's SDCTL register with ID = 00 (single processor system) or processor with ID = 01 (multiple processor system for shared memory). After setting up the power sequence bit, perform a dummy read or write access to external SDRAM. This initiates the power-up sequence for SDRAM. Listing 4 shows SDC configuration code.

```
PLL_Configuration:

     ustat2 = dm(PMCTL);
     bit clr ustat2 PLLM63 | PLLD8; //Clear multiplier and divisor bits

     #ifdef MHz266
     bit set ustat2 INDIV |PLLM43;  /* set a multiplier of 64 and a divider of 3 */
     #endif

     #ifdef MHz325
     bit set ustat2 PLLM26;  /* set a multiplier of 26 */
     #endif

     dm(PMCTL) = ustat2;
     bit set ustat2 PLLBP;   /* Put PLL in bypass mode. */
     dm(PMCTL) = ustat2;     /* The DIVEN bit should be cleared while placing the PLL in bypass mode */

     //Delay -
    //wait for PLL to lock at new rate (requirement for modifying multiplier only)
     lcntr = PLLSetlDelay, do pllwait until lce;
     pllwait: nop;
     //Delay - Done

     ustat2 = dm(PMCTL);     /* DIVEN bit returns as zero */
     bit clr ustat2 PLLBP;   /* take PLL out of Bypass, PLL is now at CLKIN*4
                               (CoreCLK = CLKIN * M/N =        CLKIN* 16/4) */
     dm(PMCTL) = ustat2;     /* DIVEN bit should be cleared while taking the PLL out of bypass mode */
     RTS;
```

Listing 2. PLL configuration code

```
SDRAM_Bank_Mapping:

     // Mapping Bank 0 to 32-bit SDRAM
     // Make sure that jumper is set appropriately so that MS0 is connected to
     // chip select of 32-bit SDRAM device
     ustat1 = dm(EPCTL);
     bit set ustat1 B0SD;
     bit clr ustat1 B1SD|B2SD|B3SD;
     dm(EPCTL) = ustat1;
     RTS;
```

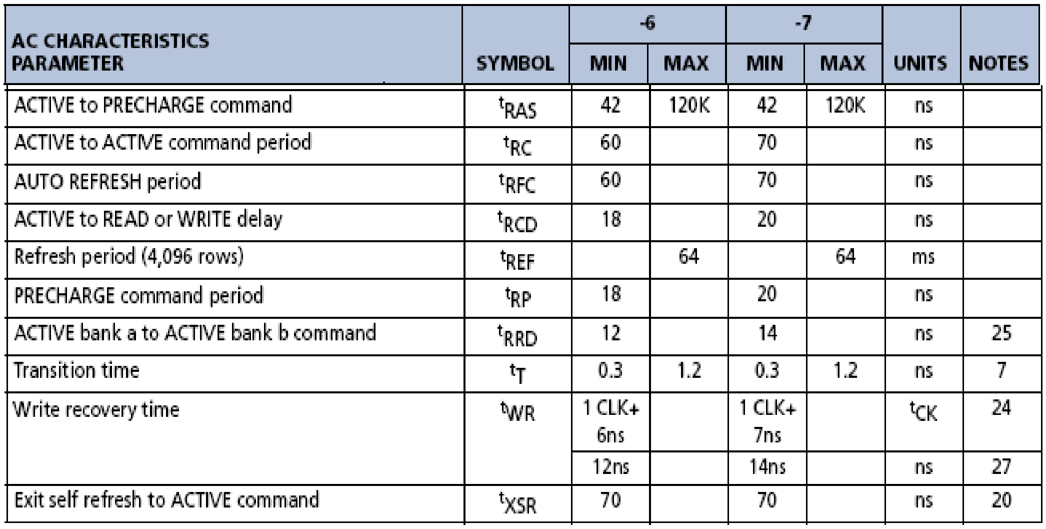
Listing 3. SDRAM mapping to one of the external memory banks

```
SDC_Configuration:

     // Programming SDRAM control registers.
     // RDIV = ((f SDCLK X t REF )/NRA) - (tRAS + tRP )
     ustat1 = 0x815; // (133*(10^6)*64*(10^-3)/4096) - (7+3)
     // SDRAM clock at 133MHz
     /*For other CCLK_SDCLK ratio values, the RDIV value needs to be re-calculated based
       on the SDRAM clock frequency. Above formula should be used for to calculate
       the RDIV value */

     bit set ustat1 SDROPT | BIT_17; // Enabling SDRAM read optimization
     // Setting the Modify to 1
     dm(SDRRC) = ustat1;

     ustat1 = 0;
     bit set ustat1 SDCL3|DSDCLK1|SDTRAS6|SDTRP3|SDCAW8|SDPSS|
                    SDTWR2|SDTRCD3|SDRAW12;
     /*SDCTL value will depend on the SDRAM clock and hence, these values need to be
       re-calculated for a different SDRAM clock frequency */

     dm(SDCTL) = ustat1;

     RTS;
```

Listing 4. Configuring the SDC

Setting Up the SDRRC and SDCTL Registers

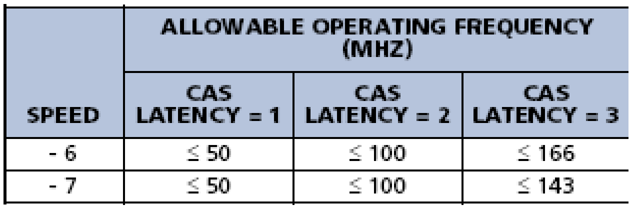
The SDRAM auto refresh rate and other timing requirement parameters are configured in the SDRRC and SDCTL registers. The SDRRC register contains the auto refresh count value, and the SDCTL register contains other timing requirement parameters as specified by the SDRAM. All timing parameters are programmed in terms of number of SDCLK cycles. All parameters depend on the frequency/period of SDCLK, which are based on the minimum timing specifications provided by the SDRAM data sheet. This section describes how to derive these parameter values and is based on the device specifications for the MT48LC4M32B2 from Micron Technology.



		-6		-7			
AC CHARACTERISTICS PARAMETER	**SYMBOL**	**MIN**	**MAX**	**MIN**	**MAX**	**UNITS**	**NOTES**
ACTIVE to PRECHARGE command	t_{RAS}	42	120K	42	120K	ns	
ACTIVE to ACTIVE command period	t_{RC}	60		70		ns	
AUTO REFRESH period	t_{RFC}	60		70		ns	
ACTIVE to READ or WRITE delay	t_{RCD}	18		20		ns	
Refresh period (4,096 rows)	t_{REF}		64		64	ms	
PRECHARGE command period	t_{RP}	18		20		ns	
ACTIVE bank a to ACTIVE bank b command	t_{RRD}	12		14		ns	25
Transition time	t_{T}	0.3	1.2	0.3	1.2	ns	7
Write recovery time	t_{WR}	1 CLK+ 6ns		1 CLK+ 7ns		t_{CK}	24
		12ns		14ns		ns	27
Exit self refresh to ACTIVE command	t_{XSR}	70		70		ns	20

Table 5. SDRAM timing requirements for MT48LC4M32B2 at 133 MHz

The timing requirement parameters needed to program the SDC are t_{RP}, t_{RAS}, t_{RC}, t_{RFC}, t_{REF}, t_{WR}, and t_{RCD}. These timing requirements are specified in the SDRAM data sheet. Refer to Table 5 for timing requirements and to Table 6 for CAS latency values for different frequency ranges.



SPEED	ALLOWABLE OPERATING FREQUENCY (MHZ)		
	CAS LATENCY = 1	CAS LATENCY = 2	CAS LATENCY = 3
- 6	≤ 50	≤ 100	≤ 166
- 7	≤ 50	≤ 100	≤ 143

Table 6. CAS latency

With the core operating at 333 MHz, the `CCLK`-to-`SDCLK` ratio set at 1:2.5, the derived `SDCLK` is 133 MHz. This yields an `SDCLK` cycle period of approximately 7.5 ns. Therefore, the required timing parameters are shown in Table 7.

SDC Timing Parameters		
t_{RP}	18 ns/7.5 ns	3 cycles
t_{RAS}	42 ns/7.5 ns	6 cycles
t_{RC}	60 ns/7.5 ns	8 cycles
t_{RFC}	60 ns/7.5 ns	8 cycles
t_{REF}	64 ms	
t_{RCD}	18 ns/7.5 ns	3 cycles
t_{WR}	1 CLK + 6 ns = 13.5 ns	2 cycles

Table 7. Required SDC timing parameters

Program the `DIV` field of the `SDRRC` register with the correct value of SDRAM refresh count. Use Equation 1 to derive this value.

$$RDIV \le \left(\frac{f_{SDCLK} \times t_{REF}}{NRA}\right) - (t_{RAS} + t_{RP})$$

Equation 1. Deriving the RDIV count value

Where:

f_{SDCLK} = `SDCLK` frequency

t_{REF} = SDRAM refresh period

NRA = Number of row addresses in SDRAM (refresh cycles to refresh whole SDRAM)

t_{RAS} = Active-to-precharge time (`SDTRAS` bits in the SDRAM memory control [`SDCTL`] register) in number of clock cycles

t_{RP} = RAS-to-precharge time (in the SDRAM memory control [`SDCTL`] register) in number of clock cycles

Therefore, for the device considered in this example:

f_{SDCLK} = 133 MHz

t_{REF} = 64 ms

NRA = 4096

t_{RAS} = 6

t_{RP} = 3

Thus, per Equation 1, R_{DIV} should be <= 2069 or 0x815.

At 133 MHz, the CAS latency to be used is 3 cycles; hence bit definition `SDCL3` should be set.

Per Table 7, the other bit definitions to be set in `SDCTL` are: `SDTRAS6`, `SDTRP3`, `SDTWR2`, and `SDTRCD3`. The column address width is 256 (8 bits); hence, `SDCAW8` should be set. The row address width is 12 bits; hence, `SDRAW12` should be set. These bit definitions are specified in the processor's header file (for example, `def21369.h` for ADSP-21369 processors). The above explained calculations can be used as reference for ADSP-2137x, ADSP-214xx processors.

Note that the external port on ADSP-21375 processors is 16 bits wide and the external port on ADSP-21371 processors is 32 bits wide. The `X16DE` bit of the `SDCTL` register must be set to 1 to configure the external port of the ADSP-21371 to be 16 bits wide. Similarly the external port on ADSP-214xx processors is 16 bit wide, therefore, `X16DE` bit of the `SDCTL` register must be set to 1 to configure the external port for 16 bits wide.

The following figures show logic analyzer screen captures taken during various SDRAM accesses for the ADSP-21368 processor.

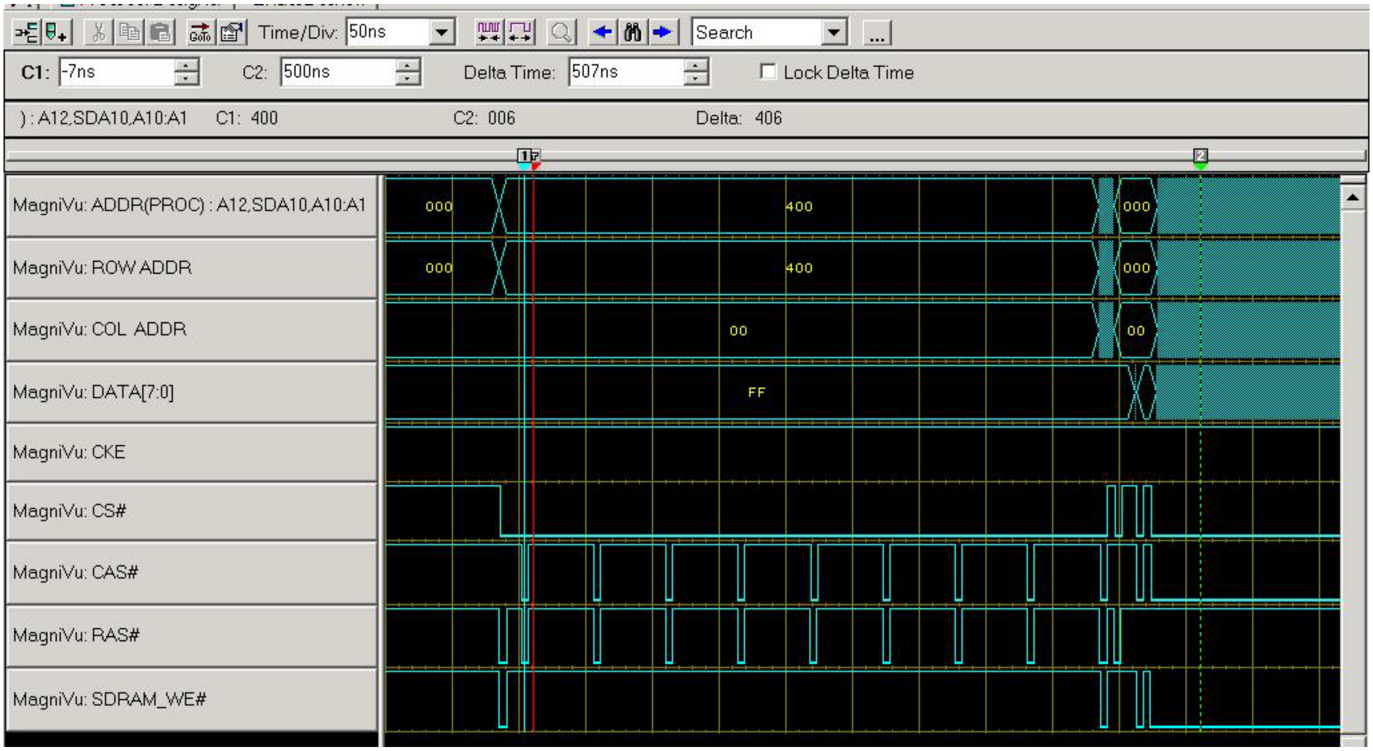
Figure 3 shows the SDRAM power-up sequence. After setting the `SDPSS` bit in `SDCTL`, a dummy read or write access to SDRAM memory triggers the power-up sequence (Please note that the dummy access is an explicit user access). The sequence depends on the SDRAM power-up mode (`SDPM`) bit. When this bit is cleared (=0), the sequence of commands that follows is:

Precharge → 8 CBR cycles → Mode register set command → Bank activation command→ dummy read/write operation.

Figure 4 shows SDRAM sequential read operations without the optimization bit (`SDROPT`) set in the `SDRRC` register. Each read operation takes multiple `SDCLK` cycles (7 cycles for each read with CL = 3, and 6 cycles for each read with CL = 2).

Precharge All → 8 auto refresh command cycles → Mode register set command → Bank activation command → Dummy write operation

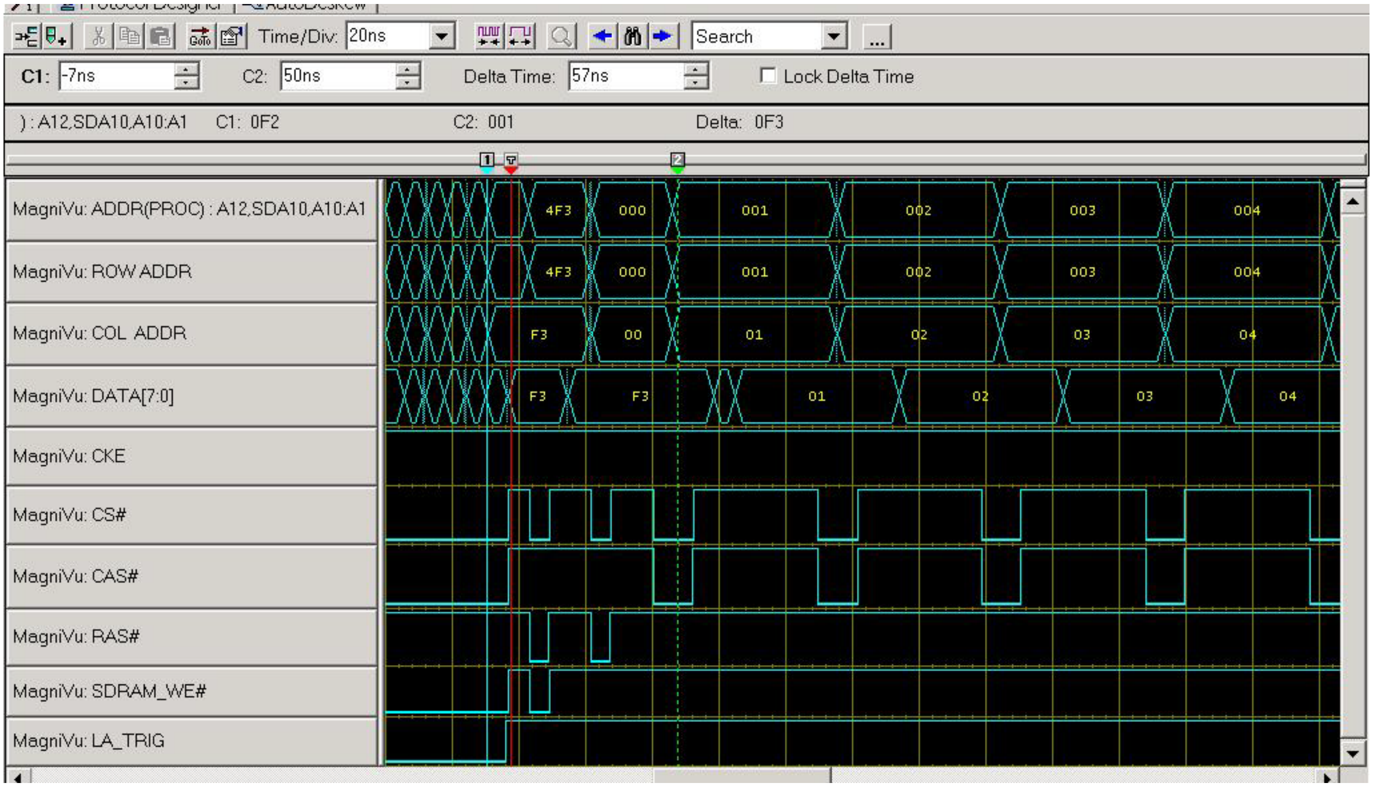
Figure 5 shows SDRAM sequential read operations with the read optimization bit (`SDROPT`) set. With optimization enabled, read operations occur every `SDCLK` cycle for every 32 words for core or DMA access.


Time/Div: 50ns

C1: -7ns C2: 500ns Delta Time: 507ns □ Lock Delta Time

) : A12,SDA10,A10:A1 C1: 400 C2: 006 Delta: 406

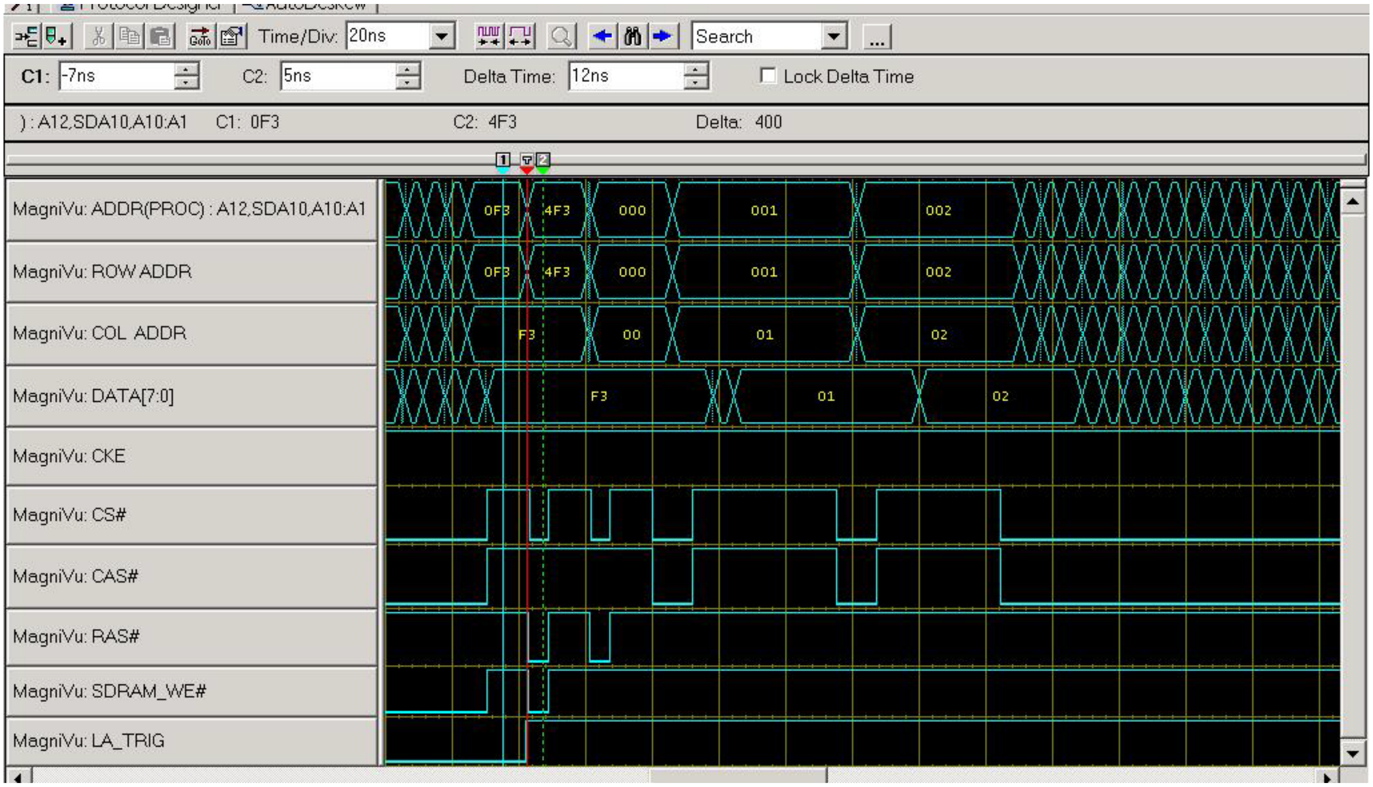
Figure 3. SDRAM power-up sequence on ADSP-21368 processors


Time/Div: 20ns

C1: -7ns C2: 50ns Delta Time: 57ns □ Lock Delta Time

) : A12,SDA10,A10:A1 C1: 0F2 C2: 001 Delta: 0F3

Figure 4. Read optimization not selected on ADSP-21368 processors



Time/Div: 20ns

Search

C1: -7ns

C2: 5ns

Delta Time: 12ns

□ Lock Delta Time

) : A12,SDA10,A10:A1 C1: 0F3 C2: 4F3 Delta: 400

Figure 5. Read optimization enabled on ADSP-21368 processors

Shared Memory Support

ADSP-21368 processors support shared memory on the external port. These processors have on-chip bus arbitration logic, enabling them to share the external bus. The shared external memory feature can also be used with SDRAM memories also. The on-chip bus arbitration logic permits the connection of up to four other processors to create a shared external bus system.

Shared memory support does not apply for ADSP-21367, ADSP-21369, ADSP-2137x, and ADSP-214xx processors.

Consider the following points when designing a shared SDRAM system:

- Configure the processor with ID = 01 to perform the power-up sequencing for the shared SDRAM
- When another processor in the shared bus system wants to access SDRAM, it should initiate a force auto refresh command (Refer to SDCTL register) for the SDRAM. This should be done only once.
- The shared SDRAM memory interface can run up to 133 MHz, provided the load conditions on the printed circuit board (PCB) are met.
- The `CCLK`-to-`CLKIN` ratio should always be an integer.
- The `CCLK` -to-`SDCLK` ratios of two and four are only supported.

- Pull the unused BRx (bus request signal) pins high. They are used by the external port bus arbitration logic.
- The external bus priority configured using the RPBA signal should be the same for all the processors.

Figure 6 shows a typical shared memory system consisting of two processors and one SDRAM memory device.

In the shared memory system, since only integer ratios are allowed for (CCLK-to-CLKIN and CCLK-to-SDCLK), the CCLK would be limited to 266 MHz for the SDCLK operation at 133 MHz. In a shared memory system, whose CCLK is set for 333 MHz, the minimum CCLK-to-SDCLK clock ratio of 4 results in an 83.25 MHz SDCLK.

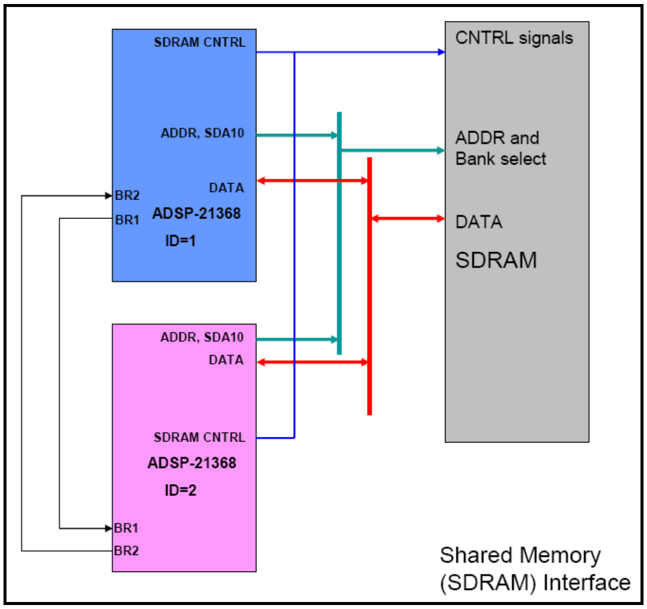


Figure 6. SDRAM as shared memory for two ADSP-21368 processors

Execution from External Memory

ADSP-21375 and ADSP-214xx processors support execution of instructions from external memory bank 0 via the 16-bit external port, while ADSP-21371 processors support the same via the 32-bit external port. This external memory can be SDRAM or asynchronous memories (such as SRAM or flash).

For the SDRAM, the processor supports 16-bit to 48-bit packing and 32-bit to 48-bit packing. The instruction throughput when executing code from external SDRAM memory is 2 instructions every 3 SDCLK (peripheral) clock cycles over a 32-bit-wide external port, and 2 instructions every 6 SDCLK clock cycles over a 16-bit-wide external port.

For ADSP-214xx processors have an additional optional feature to use VISA encoding to store instruction into SDRAM memory to reduce the overall code size.

General PCB Design Guidelines

The following are general PCB guidelines for designing an SDRAM interface with ADSP-21368, ADSP-2137x, and ADSP-214xx processors.

- Keep the address and data lines as short as possible and ensure equal length. Pay special attention to SDCLK and the SDRAM control lines as they are used for SDRAM refresh and data signals as they toggle almost every SDCLK cycle. For ADSP-21368 and ADSP-2137x processors, limit the SDCLK trace length between 2 and 4 inches. The DATA and ADDR trace lengths can be between 4 and 8 inches. The trace width can be 5 millimeters. For ADSP-2147x processors keep the trace length to a maximum of 4 inches for all frequencies. For ADSP-2148x processors, if the frequency of operation is above 133 MHz, trace length can be up to 4 inches. If the frequency of operation is less than or equal to 133 MHz, SDCLK trace length can be up to 4inches and ADDR, DATA and Command signal trace length can be up to 6 inches.
- Terminations models (serial or parallel) for SDCLK, DATA, and ADDR are not explicitly recommended. Use signal integrity (SI) tools to identify any needed terminations. Excessive termination can cause slower rise and fall times and may increase delays. Analog Devices recommends that system/board designers verify all design aspects such as signal integrity, electrical timings, and so on using simulation tools *before* PCB fabrication.
- The processor's I/O drivers can support nominally up to 30 pF of load on each I/O. If the load is higher a timing delay comes into play (Refer to datasheet for further details).
- Use sufficient decoupling for memory devices. Figure 7 shows decoupling for each SDRAM device on the application PCB. Use proper decoupling for the processor as mentioned in the system design chapter of the processor's *Hardware Reference* manual [1].
- For ADSP-21368 processors use the prescribed filter circuitry for the A_{VDD} supply as shown and described in the processor's data sheet. The details of this are described in the processors data sheets.

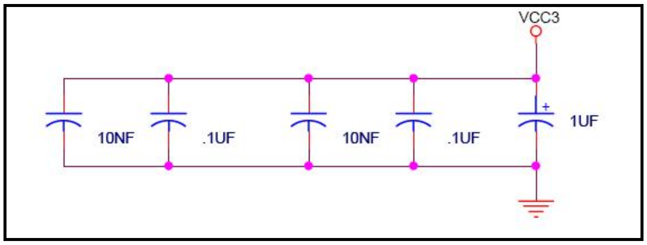


Figure 7. Decoupling for each memory device

- Also consult the application notes of the specific SDRAM manufacturer in regards to current requirements (depending on operation types) and surrounded bypass filter requirements.

Software Code

Example codes are supplied in a `.ZIP` file associated with this EE-Note. It contains code for both 16- and 32-bit-wide SDRAMs in core mode and DMA mode for the ADSP-21368 and ADSP-2137x processors. One of the code examples demonstrates shared memory using 32-bit SDRAM on ADSP-21368 processors. Also, two other example codes demonstrate the execution from SDRAM on ADSP-2137x processors. Other two examples implement DMA transfer between SPORTs and SDRAM and using SIMD from external memory for ADSP-2147x processors (applicable for ADSP-2148x processors also).

The first example code places the code in the external memory using the `.LDF` file. For this case, the SDRAM controller must be initialized before loading the application. The VisualDSP++® tools initialize the SDRAM controller with the default values for the SDRAM on the ADSP-21375 EZ-KIT Lite board before loading the application. In case the SDRAM used on the target board is different, the SDRAM controller initialization value can be edited to modify the default value configured in the `.xml` file (located in the processor's `System\ArchDef` folder).

In the second example code, the function that must be executed from the external SDRAM is placed as part of the internal memory in a `.dat` file. This data is copied to the external SDRAM before executing the specific function from the SDRAM. Use this approach when the part of the application that must be executed is available in the external memory bank in which the execution from external memory is not supported.

The third example implements DMA transfer between SPORTs and SDRAM for ADSP-21479 processors. This code transfers data from internal memory to SDRAM through SPORTs and transfers the received data at the external memory back to internal memory through SPORTs and finally compares the transmitted and the received data at the internal memory.

The fourth example implements the SIMD from external memory. The input samples are stored in external memory and accessed using SIMD.

Example to implement the VISA feature for ADSP-21479 and ADSP-21489 processors can be found under the VisualDSP++ 5.0 tools directory:

```
C:\...\Analog Devices\VisualDSP 5.0\214xx\Examples\ADSP-21479 EZ-Board\VISA_example
```

SDRAM Throughput Optimization Guidelines

The following is a list of SDRAM throughput optimization techniques to help improve performance when accessing SDRAM memory devices:

- For SDRAM reads (core or DMA), the best case throughput is achieved only when the accesses are sequential and uninterrupted. Also, the read optimization should be enabled and the read optimization modifier should be equal to the modifier between the successive accesses.
- For SIMD core reads, the best case throughput is achieved only when the accesses are sequential and uninterrupted and the modifier between the successive accesses should be equal to ‘2’. The reason is that for SIMD accesses the SDRAM controller uses read optimization with modifier equal to ‘1’ irrespective of the SDROPT, and modifier settings in the SDRRC register.
- For writes, the throughput is always the same whether the accesses are sequential or non sequential.
- Frequent page switching in the same bank can reduce the SDRAM throughput as it requires additional overheads for closing the previous page and opening the new page. It can occur because of frequent calls/jumps in case of external code execution. It case of data accesses, it can also occur because of arbitration between two sources (core/DMA) requesting for the SDRAM accesses in the same bank but different pages. The throughput might improve by using page interleaving in case of code execution and by placing the buffers in different banks for data accesses.
- User should be careful while using the rotating priority for arbitration between two sources (core or DMA) as it may cause breaks in the sequential reads. In case using the fixed priority is not possible, one must consider using arbitration freezing with maximum possible freeze length. This assures that the accesses are sequential at least for the freeze length and no bus master occupies the bus for a very long period as well.

References

[1] *ADSP-214xx SHARC Processor Hardware Reference,* Rev 0.3, July 2010. Analog Devices, Inc.

[2] *ADSP-2137x SHARC Processor Hardware Reference, Rev 2.1, May 2010.* Analog Devices, Inc.

[3] *ADSP-21478/ADSP-21479 SHARC Processor Data Sheet (Rev PrC, 10/2010).* Analog Devices, Inc.

[4] *ADSP-21483/21486/21487/21488/21489 SHARC Processor Data Sheet (Rev PrA, 04/2010).* Analog Devices, Inc.

[5] *ADSP-21371/ADSP-21375 SHARC Processor Data Sheet (Rev C, 09/2009) . Analog Devices, Inc.*

[6] *ADSP-21367/ADSP-21368/ADSP-21369 SHARC Processors Data Sheet (Rev E, 07/2009).* Analog Devices, Inc.

[7] *ADSP-21479 EZ-Board Evaluation System Manual*, Rev 1.0, April 2010. Analog Devices, Inc.

[8] *ADSP-21489 EZ- Board Evaluation System Manual*, Rev 1.0, April 2010. Analog Devices, Inc.

[9] *ADSP-21369 EZ-KIT Lite Manual (Rev 2.2, September 2009)* . Analog Devices, Inc.

[10] *ADSP-21375 EZ-KIT Lite Evaluation System Manual*, Rev 1.1, September 2007. Analog Devices, Inc.

[11] *ADSP-21371 EZ-KIT Lite Evaluation System Manual*, Rev. 1.0, September 2009. Analog Devices, Inc.

[12] *MT48LC4M32B2TG SDRAM Data Sheet, Rev. L 1/09 EN,* Micron Technology, Inc.

[13] *MT48LC16M16A2SDRAM Data Sheet, Rev. N 1/10 EN,* Micron Technology, Inc.

[14] *MT48LC8M16A2 SDRAM Data Sheet, Rev. N 1/09 EN*. Micron Technology, Inc.

[15] *K4S643232H SDRAM Data Sheet*, Rev. 1.1, August 2004. Samsung Electronics.

[16] *W9864G2GH-7 SDRAM Data Sheet*, Rev. A01, August 2006. Winbond Electronics Cor.

Document History

Revision	**Description**
Rev 5 – November 22, 2010, *by Deepa Venkataraman*	Renamed EE-Note from: Interfacing SDRAM Memories to ADSP-21368, ADSP-2137x SHARC® Processors to Interfacing SDRAM Memories to SHARC® Processors. Updated the document to add the ADSP-2147x and ADSP-2148x SDRAM controller details, re-organized the contents accordingly, and removed register bit descriptions. Added the SDRAM throughput optimization guidelines section.
Rev 4 – January 9, 2008, *by Deepa Venkataraman*	Removed references to obsolete Winbond W986432DH and modified the PLL configuration code.
Rev 3 – April 19, 2007, *by Jeyanthi Jegadeesan*	Updated the document to add the ADSP-2137x SDRAM interface details and modified document title accordingly.
Rev 2 – March 14, 2006, *by Jeyanthi Jegadeesan*	Revised info pertaining to 133 MHz SDCLK operation and updated document title accordingly.
Rev 1 – November 28, 2005, *by Aseem Vasudev Prabhugaonkar*	Initial release as *Interfacing 166 MHz SDRAM Memory to ADSP-21367/21368/21369 SHARC Processors (EE-286).*