
Motorola Digital Signal Processors

DSP56001 Interface Techniques and Examples

by
Roman Robles
Digital Signal Processor Operation

ARCHIVED FROM MOTOROLA SEMICONDUCTOR, INC. 2005


Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and  are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

Table of Contents

SECTION 1	1.1 DSP56001 Memory I/O Basics	1-4
Interfacing	1.2 Memory Subsystem Overview	1-5
Motorola's	1.3 Circuit Description	1-8
DSP56001 to	1.4 Summary	1-11
Pseudo Static RAM		
SECTION 2	2.1 Circuit Overview	2-6
A Simple Dynamic	2.2 Circuit Description	2-9
RAM Interface for	2.3 Summary	2-12
the DSP56001		
SECTION 3	3.1 Interface Circuit Overview	3-1
A Simple ISA Bus	3.2 Detailed Circuit Description	3-3
Interface for the	3.3 Timing	3-4
DSP56001		
SECTION 4	4.1 Introduction	4-1
Communicate with	4.2 Example Program	4-1
the DSP56000 Host		
Interface Using		
C Language		
INDEX		Index-1
REFERENCES		Reference-1

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Illustrations

Figure 1-1	Pseudo Static RAM Auto Refresh Timing	1-3
Figure 1-2	DSP56001-to-PSRAM Schematic	1-6
Figure 1-3	PSRAM Interface State Diagram	1-9
Figure 1-4	DSP56001-to-PSRAM Timing	1-10
Figure 1-5	DSP56001-to-PSRAM PLD Definition	1-13
Figure 1-6	PSRAM Interface Initialization Code	1-14
Figure 2-1	DRAM Memory Address Multiplexing	2-3
Figure 2-2	DRAM Refresh Modes	2-4
Figure 2-3	DSP56001-to-DRAM Timing	2-5
Figure 2-4	DSP56001-to-DRAM Schematic	2-8
Figure 2-5	DRAM Interface State Diagram	2-10
Figure 2-6	PLD Design File -DRAM Interface	2-13
Figure 2-7	DRAM Interface Initialization Code	2-15
Figure 3-1	DSP56001-to-ISA Bus Interface Schematic	3-2
Figure 3-2	PLD Definition for the ISA Bus Interface	3-6
Figure 3-3	DSP56001-to-ISA Bus Interface Timing	3-8
Figure 4-1	Example Program of DSP56000 Host Interface Using C Language	4-4

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

SECTION 1

Interfacing Motorola's DSP56001 to Pseudo Static RAM

***“PSRAM
combines the
economies of
DRAM with the
straightforward
interface of fully
Static RAM to
provide 128K
bytes in a 32-pin
DIP.”***

When the design definition of a DSP subsystem calls for a large memory space, the cost of populating this space with static RAM (SRAM) can be prohibitive. Although SRAM offers the advantages of high speed and a very simple interface, the complex structure of the SRAM storage cell results in SRAM price/density ratios which are inferior to those of dynamic RAM. Pseudo Static RAM (PSRAM) presents one possible compromise between the contradictory requirements of high density, low cost, high speed and interface simplicity.

This section presents a simple implementation of a PSRAM interface to the DSP56001. Using an array of three 128K x 8 PSRAMs, the circuit provides access to 128K 24-bit words of data space. With the DSP56001 operating from a 33MHz clock, this memory subsystem will operate with 2 wait states for non-consecutive accesses.

Pseudo-Static RAM

PSRAM combines the economies of DRAM with the straightforward interface of fully Static RAM to provide 128K bytes in a 32-pin DIP. Internally, the device contains a dynamic RAM array with on-board address multiplexing, an internal refresh row counter, and an internal refresh timer. The memory array is divided into eight sections, each consisting of a of 512 (row) x 256 (column) matrix of storage cells, forming a byte-wide memory which is 128K locations deep.

The device is pin compatible with the 128Kx8 SRAM JEDEC pinout with the exception of pin 1 (on standard SRAMs, this pin would be a no-connect; on PSRAM, it is the refresh strobe \bar{F}). These features enable the PSRAM to replace fully static RAM in many applications with a minimum amount of "glue".

PSRAM has two complementary enable lines, $\bar{E}1$ and $E2$. During read and write operations, these enable lines must *strobe* the address into the device. This is another difference between PSRAM and fully static RAM.

Since the PSRAM is based on DRAM storage elements, it requires a precharge delay between successive accesses and a periodic refresh. PSRAM supports three different refresh modes; CE-only refresh, auto refresh and self refresh.

- CE-only refresh requires external hardware or software to provide periodic addressing of each of the 512 rows. Use of this method would add a considerable amount of interface hardware or would cause significant degradation to software performance.
- Self refresh can be entered after 8 ms in standby mode. In this mode the on-board refresh timer and refresh counter are used to provide the refresh sequencing. A delay slightly greater than one access cycle is required when leaving this mode before data read/write operations can proceed. This mode is useful for long standby periods, but is not suitable for device refresh during periods of normal DSP activity due to the unique timing requirements. To use this mode during idle periods would require mode selection logic as well as the circuitry associated with one of the other "active access" modes.

- Auto refresh occurs when the PSRAM is disabled by either of the device select inputs going false followed by the refresh pin \bar{F} going active. For each transition of \bar{F} , one row of each section is refreshed and the refresh row counter is advanced in preparation for the next refresh cycle. The example presented in this note uses this mode because it requires the least amount of external logic and impacts the normal DSP software only when a data transfer contends with a refresh cycle.

Figure 1-1 depicts an auto refresh cycle in which two rows are refreshed in succession. Note that either $\bar{E}1$ or $\bar{E}2$ can disable the device. Refer to the data sheets specific to the PSRAMs selected for any particular application.

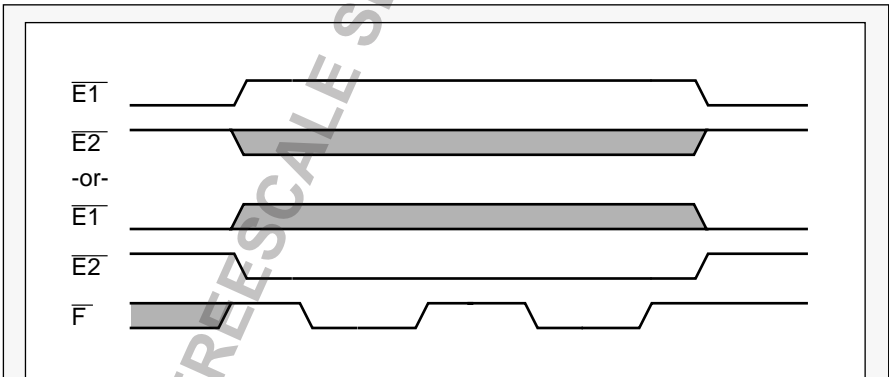


Figure 1-1 Pseudo Static RAM Auto Refresh Timing in which 2 rows are refreshed in succession. Note that either $\bar{E}1$ or $\bar{E}2$ can disable the device. Please refer to the data sheets specific to the PSRAMs selected for any particular application.

1.1 DSP56001 Memory I/O Basics

Memory interface to the DSP56001 occurs over Port A of the processor. Port A consists of 24 bi-directional data lines (D0-D23), 16 address lines (A0-A15), three memory reference lines (\overline{PS} , \overline{DS} , X/\overline{Y}) and two data strobes (\overline{RD} , \overline{WR}). Additionally, a pair of bus access control signals, $\overline{\text{Bus Request/Bus Grant}}$ ($\overline{BR/BG}$), can be used to synchronize access requests between the processor and another device attempting to gain mastership of the bus. The bus access pins have alternate functions, $\overline{\text{Bus Strobe /Wait}}$ ($\overline{BS/WT}$), which allow external circuitry to insert additional wait states in external bus cycles. To minimize power consumption, the address lines remain stable until the beginning of the next external access. The memory reference signals (\overline{PS} , \overline{DS} and X/\overline{Y}) are deasserted during periods when the external bus is idle, but are **not** deasserted during successive accesses to the same external memory space.

Setting bit 7 of the processor's Operating Mode Register (OMR) causes the bus access control bits to assume the $\overline{\text{Bus Strobe/Wait}}$ ($\overline{BS/WT}$) mode. In this mode, the \overline{BS} pin is asserted at the beginning of every external access and is released during T3 of each external cycle. Assertion of the \overline{WT} pin during T2 **while \overline{BS} is asserted** adds wait states to the bus cycle.

Wait states will continue to be inserted until two falling edges of EXTAL occur in succession with the release of \overline{WT} . \overline{WT} should never be asserted when \overline{BS} is inactive.

When the DSP56001 is reading data from the bus, the data must be stable for the specified setup and hold periods before and after (respectively) the rising edge of the read strobe \overline{RD} . During processor write operations to the external bus, the data is valid for a specified time before and after the rising edge of the write strobe \overline{WR} .

These relationships are shown in the simplified PSRAM timing diagram of Figure 1-4. (For DRAM timing see Figure 2-3.) For more detailed information, refer to the **DSP56001 User's Manual** and the **DSP56001 Data Sheet**.

1.2 Memory Subsystem Overview

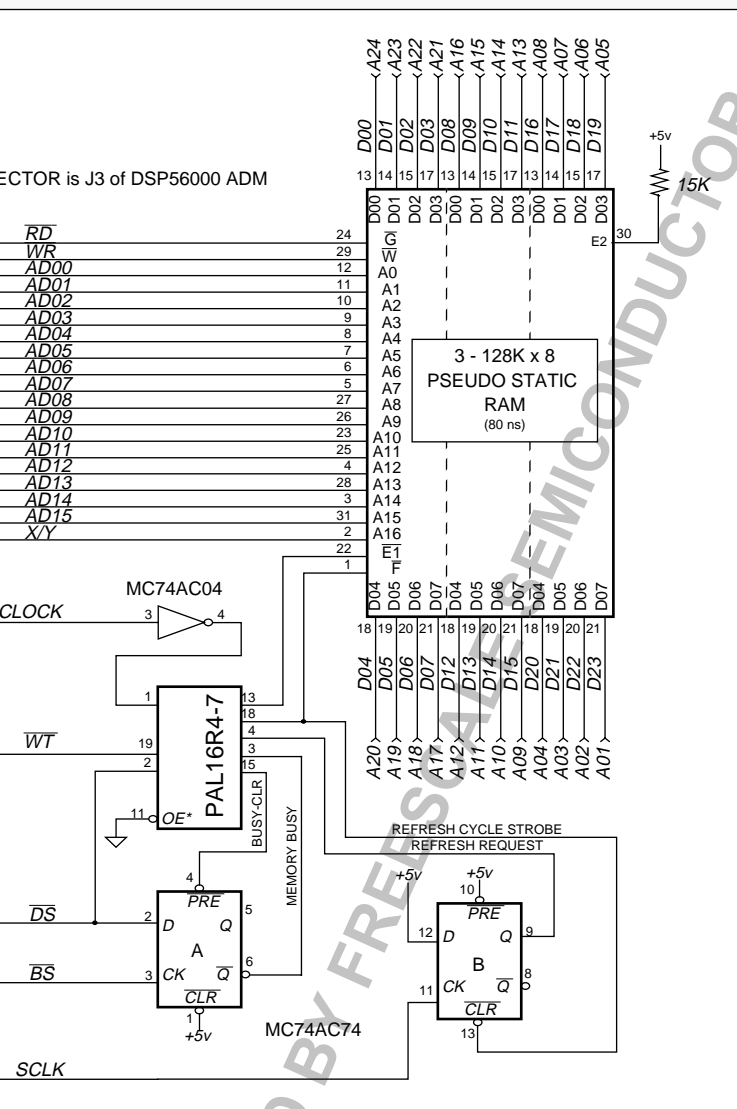
The circuit in Figure 1-2 is designed to serve as an extension of the Motorola DSP56000 ADS Application Development Module (ADM). The Static RAM on the ADM should be configured to reside solely within the DSP56001 program space. The PSRAMs and their interface circuitry are attached to the DSP56001's Data and Address Buses via ADM connector J3.

The PSRAM bank consists of three devices. Each device provides 128K storage cells for each of 8 data bits, forming an array of 128K 24-bit words. The DSP56001 can address 64K 24-bit words in each of its two data spaces, X:memory and Y: memory. Therefore, this PSRAM array fully populates both of the processor's data spaces.



ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

ECTOR is J3 of DSP56000 ADM



-2 DSP56001-to-PSRAM Schematic provides two functions: it controls the refresh cycles and it generates precharge delays. This is a schematic depiction of the interface circuit.

In order to minimize the component count, the refresh request timing is supplied by the processor's Serial Control Interface (SCI) clock, SCLK. Initialization software configures this clock to provide a pulse train with a 15 μ s period. Once initialized, the generation of this signal is completely transparent to any code executing on the processor. Figure 1-6 is a listing of the initialization code and a short "pass/fail" memory test routine. The value loaded into the SCI Clock Control Register (SCCR) at X:\$FFF2 will vary as a function of the system clock frequency. For a 33 MHz clock, a value of \$107F will yield the desired refresh rate of 15.6 μ s per row.

A second task of the initialization software is the selection of the $\overline{BS}/\overline{WT}$ mode of operation. This mode allows an external source to insert wait states into bus cycles, and is employed by the interface when precharge and refresh delays are needed.

The interface operates from the same clock which drives the processor. In systems operating from an external clock source, this should be easy to provide. In this example, the DSP56001 clock is buffered by a CMOS inverter which subsequently drives the interface circuitry. It is essential that the device used to buffer this clock has a very high input impedance. The oscillator on the DSP56001 **cannot** drive a TTL input load.

1.3 Circuit Description

Figure 1-2 is a schematic depiction of the interface circuit. Basically, the interface provides two functions: it controls the refresh cycles and it generates pre-charge delays.

Section "B" of the MC74AC74 generates a refresh request on the rising edge of SCLK and holds the request until the PAL16R4-7 controller executes a refresh cycle and resets the Flip-Flop. As shown in Figure 1-3, the controller defers a refresh cycle until any access currently in progress completes. If the subsequent DSP56001 instruction cycle does not access this PSRAM array, this refresh is transparent. If the subsequent cycle does access this area of memory, then wait states are inserted until the refresh completes.

Section "A" of the MC74AC74 is clocked by the rising edge of \overline{BS} , which occurs at the end of each external bus cycle. In the event that the bus cycle which has just ended was an active cycle for the PSRAM array, the PSRAM address decode (\overline{DS} in this example) will be latched into Flip-Flop "A". The PLD will receive MEMORY BUSY status, indicating that a pre-charge cycle is in progress. The PLD will hold off further PSRAM activity until sufficient precharge delay has elapsed. Note that no extra delay is seen by the DSP56001 if the subsequent cycle does not access this particular PSRAM. If multiple banks of PSRAM are used, bank interleaving strategies can result in most (or all) of the precharge cycles being hidden

