TMS320 DSP
DESIGNER'S NOTEBOOK
Number 45
TEXAS INSTRUMENTS

Fast TMS320C5x External Memory Interface

Contributed by Joe George

Design Problem

How can I connect zero-wait-state SRAM to a fast 'C5x without external decode logic?

Solution

Introduction

The 'C5x provides the hardware engineer two sets of signals for external memory interface. The first is documented in Section 6 of the 'C5x User's Guide using $\overline{RD}$ and $\overline{WE}$ (Figure 6-13). These signals allow the glueless interface to the $\overline{OE}$ and $\overline{WE}$ respectively of various memories. In essence, the 'C5x supplies the decode for you. As can be seen in Appendix B of the 'C5x User's Guide, the $\overline{RD}$ and $\overline{WE}$ signals change a half cycle later than the address. But as the 'C5x gets faster, the setup and hold timings for $\overline{RD}$ and $\overline{WE}$ as shown on A-14, become more constrained and therefore, more difficult for a memory to satisfy. Thus the second set of signals, R/$\overline{W}$ and $\overline{STRB}$, become useful for memory interface.

As with the 'C2x and 'C3x, the R/$\overline{W}$ and $\overline{STRB}$ signals are usually decoded by external logic (alternative shown later in this document). The most important timings for a memory to meet when using R/$\overline{W}$ and $\overline{STRB}$ is read data access from address valid (ta_A) as seen on page A-14 of the 'C5x User's Guide. If this timing is satisfied, then by examining Appendix B and page A-14, it can be proven by inspection that the write timing is satisfied. The following table summarizes ta_A for the following 'C5x speeds.

Name	Instruction Cycle	ta_A
TMS320C5x-40	50 ns	32 ns
TMS320C5x-57	35 ns	20 ns
TMS320C5x-80	25 ns	15 ns

ta_A represents the amount of time required for the memory/logic to drive valid data once the 'C5x has generated a valid address on its external bus during a memory access. For example the 'C5x EVM, which has a TMS320C50-40, uses 25-ns SRAM and 7-ns PALs to deliver valid data in 25+7=32 ns, thus satisfying the device's 32-ns requirement.

However, as one approaches using a 25-ns 'C5x in a similar fashion, one finds that 10-ns memories and 5-ns PALs are quite expensive. Therefore it is advantageous to examine a "No Decode" memory interface option.

No Decode Memory Interface

The no decode memory interface is seen on page 12-5 of the 1992 'C3x Users Guide. There are some additional features of the 'C5x, such as partitionable software wait states, which makes the no decode memory interface an attractive solution. The memory interface still uses R/$\overline{W}$ and $\overline{STRB}$, but in a different way. The basic concept is to allow the external SRAMs to be on continuously so that 0-wait-state operation is achieved. There is also no decode logic to inhibit bus cycle speed. When other devices on the DSP's bus are accessed (in I/O space or even other sub-64K banks of program or data space), the fast SRAMs are removed from the bus using the chip selects. The ' C5x's on-chip software-programmable wait-state generator becomes an ideal device for giving external logic enough time to juggle the chip selects of various devices on the external bus.

Let us now revisit the issue of ta_A timing. As an example, let's take a 20-ns 'C5x device with ta_A=15 ns max. Figure 1 below illustrates the no decode zero-wait-state connection of external memory. Since ta_A is 15 ns, a 15-ns memory would be sufficient to satisfy ta_A. The SRAMs in Figure 1 must have a $\overline{WE}$ controllable access feature allowing the RAM $\overline{OE}$ to be tied low. The DSP memory strobe ($\overline{STRB}$) is connected to the memory chip select ($\overline{CS}$ or $\overline{CE}$), and DSP read-write signal (R/$\overline{W}$) is connected to RAM write enable ($\overline{WE}$). Appendix B of the 'C5x User's Guide verifies this configuration works for all bus cycles. In order for other devices to hang on the DSP's external bus, the SRAM should have a second chip select used to remove it from the bus. This is shown below:

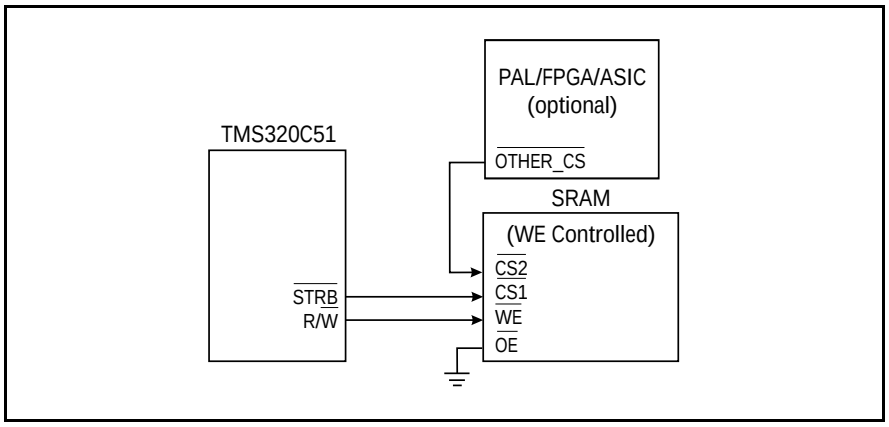

Figure 1. 'C5x no decode SRAM interface

The configuration above permits two types of memory schemes when using address and memory strobes. Notice in Figure 1, address lines, $\overline{PS}$, and $\overline{DS}$ are not shown. The first possible configuration would use a 64K RAM (i.e., address lines A0 to A15) where the corresponding DSP and RAM address lines connected. The $\overline{PS}$ and $\overline{DS}$ are left unconnected. This gives the combined program and data scheme (dubbed the CPD scheme) as seen in a 'C5x EVM. This means program and data spaces will overlap in the external RAM. There is no differentiation of program and data in the DSP's external memory. This allows flexible program/data allocation and also makes memory paging quite easy (assuming you get higher-density SRAMs. If you need to cascade SRAMs, then chips selects and S/W wait-states need to be used (as seen in Figure 1). In either case, it is assumed

that the action of switching pages is not zero-wait-state). The disadvantage of CPD is that the address range of the DSP has been cut in half.

The second method of no decode logic address mapping is to use 128K memories and connect A16 to one of the memory space strobes $\overline{PS}$ or $\overline{DS}$. Thus the single bank of 128K RAMs are divided into separate 64K blocks of program and data (dubbed the SPD scheme). This allows full-speed operation and switching between two "banks" of program and data that are actually one bank of SRAM.

A disadvantage of both these schemes is subtly apparent in the name, no decode. This means that sub-64K (or whatever maximum size) blocks cannot be paged, enabled, etc. Also mixing CPD and SPD schemes on the same bus have disadvantages for paging (see Designer Note #46). The SPD scheme requires A16 to be tied to a DSP memory strobe. The CPD scheme leaves the memory strobes unconnected.

The above configurations should allow a hardware designer to use the slowest, thus cheapest, memories possible when interfacing external memory to a fast 'C5x. This is becoming increasingly important as DSPs become faster.