Interfacing a Graphics LCD to a DSP via a Micro-controller – Simple Distributed Processing used to Enhance the Integration of DSP and Micro-controller Courses in an EET Program.

Anthony J. A. Oxtoby, Christopher S. Arndt
Purdue University, West Lafayette IN/Xilinx Inc., Boulder CO

Abstract

This paper describes a hardware arrangement that allows data transfer between the 16-bit fixed point ADSP2181 digital signal processor and a Seiko G1216 graphical LCD via an 8-bit 80C552 micro-controller. In this application, the LCD is used to display the results of a 128-point DFT implemented on the ADSP2181 operating at a sampling frequency of 22.05kHz. The interface, however, provides a more broadly useful mechanism to connect slow external devices to the DSP processor without incurring the wait states that are necessary when interfacing directly to the DSP memory spaces and to offload the processing associated with data manipulation and handshaking with these devices. It also provides a link connecting content of the two required microprocessor courses in the EET undergraduate program.

Introduction

The sequence of required microprocessor based courses in the EET program at Purdue University consists of an introductory sophomore level course based on the 80C552 micro-controller and an introductory DSP course at junior/senior level using the ADSP2181[1] digital signal processor. The DSP course includes the operation and programming of a fixed-point processor and then moves on to cover the theory and implementation of common DSP applications such as filtering, audio effects and the DFT/FFT.

Because of the hardware emphasis in the course, interfacing to I/O devices has been included, permitting some external interaction to take place with the DSP algorithms. Directly interfacing such devices to the processor memory spaces reduces algorithm speed because wait states are needed in any transactions with these slower I/O devices. One alternative approach is to use serial data transfer through the processor's serial ports and perform all the necessary scaling in the DSP. Another is to use an external controller to handle the acquisition and scaling of data and the appropriate handshaking for slower I/O devices along with relaying data to and from the DSP processor via direct memory access. Simple circuitry to implement the data transfer and handshaking has been developed and is now included in selected laboratory exercises. This way one can extend the interfacing options in the course, take advantage of a wider range of I/O devices such as push buttons, LCD displays, multi-channel ADCs etc. and provide a direct link to earlier micro-processor courses. Student projects can also take advantage of this technique.

Microprocessor Hardware

The DSP course uses the ADSP-2181 based EZ-Kit Lite, an inexpensive yet versatile development system from Analog Devices. Incorporated onto the EZ_KIT Lite is a 16-bit fixed point, 30ns ADSP-2181 processor with 16k words of internal data memory, 16k words of program memory and several additional off-chip memory spaces. In addition, there is an AD1847 stereo audio CODEC with programmable sampling rate. Access is provided to all external data, address and control bus lines through expansion connectors on the board. Additional analog peripherals can be connected to the processor through the 2k word 16-bit I/O memory space but would require individual glue logic circuitry to decode bus signals along with wait state insertions to obtain reliable data exchange with slow I/O devices.

In contrast, the introductory microprocessor course uses the 80C552 micro-controller along with a variety of I/O devices such as push buttons, potentiometers, LEDs, and LCD displays. All devices connect easily to the various ports on the 80C552 and students have a semester's experience of working with the system before taking the DSP class. It was in an effort to take advantage of both the 80C552 hardware and the students' experience and thus provide some continuity between the two courses that the interface between the two systems was developed.

The ADSP-2181 IDMA Port

The ADSP-2181 provides several means to interface with peripheral processors and devices. In addition to the interface buses used to access the various memory spaces, it contains an Internal Direct Memory Access (IDMA) port. The port supports boot loading and run-time access to the program memory (PM) and data memory (DM) spaces within the ADSP-2181 but does not permit access to internal memory mapped control registers.

Control of the IDMA port is achieved through an internal register mapped into DM(0x3FE0) accessible to both the ADSP-2181and the host device (80C552). In this application, only the 80C552 host accesses this register. Data transfer through the IDMA port takes place via the 16-bit IDMA port address/data bus and is controlled using the handshaking signals listed in table 1 below.

Table 1 IDMA Handshaking Signals

Pin Name(s)	Active State	Input/Outpt	Function
IRD	Low	Input	IDMA Port Read Strobe
IWR	Low	Input	IDMA Port Write Strobe
IAL	High	Input	IDMA Port Address Latch Enable
IS	Low	Input	IDMA Port Select
IAD15-0		Input/Output	IDMA Port Address/Data Bus
IACK	Low	Output	IDMA Port Access Ready Acknowledge

Data transfer is initiated by writing, to the IDMA control register at DM(0x3FE0), a 16-bit word which defines the memory space to be accessed, PM or DM, and a 14-bit starting address. In the next IDMA bus cycle, data is either written to or read from the specified address in the ADSP-2181 memory, depending on the whether the port write (IWR) or port read (IRD) strobe lines respectively, are asserted. Following the memory access the address in the IDMA control register is automatically incremented thus avoiding additional address latch cycles when accessing contiguous memory locations. In all IDMA transfers, the port select (IS) must also be asserted.

The 80C552 to ADSP-2181 IDMA Interface

The IDMA port on the ADSP-2181 is specifically designed for a seamless interface to another 16-bit device whereas the 80C552 micro-controller is an 8-bit device. In order to transfer 16-bit data via the IDMA port, two pairs of 8-bit tri-state latches are used to create two bi-directional ports to temporary hold the upper and lower bytes of data during transfers and also to provide isolation and thereby reduce noise transfer from the 80C552 system to the ADSP2181. A block diagram of the interface is shown below in figure 1.

Two complete cycles of the 80C552 are needed for each cycle of the IDMA port. The operation when writing to the ADSP2181 data memory or the IDMA control register is as follows. During the first cycle, the LS byte of the address to be written to the IDMA control register is written to latch C. During the second cycle the MS byte is written to latch A and then the output enable pins of latches A and C are asserted along with the IS control line. Either the IWR or the IAL control lines are also asserted depending on whether data or an address is being written to the IDMA port. Performing the transfers in this order supplies the IDMA port with the complete 16-bit value at one time.

When conducting an IDMA read cycle to transfer data to the 80C552, the process described above is reversed. During the first 80C552 external bus cycle, the IS and IRD control lines of the IDMA port are asserted and the MS byte of the data is written to latch B. The output enable for latch B is asserted so that the 80C552 can read the MS bits and the 8 LS bits from the IDMA are latched into latch D. For the second 80C552 cycle, the output of latch B is disabled and the LS byte of the data is read from latch D. All four tri-state latches and the glue logic for the interface is programmed into an Altera 7182 CPLD to reduce chip count.

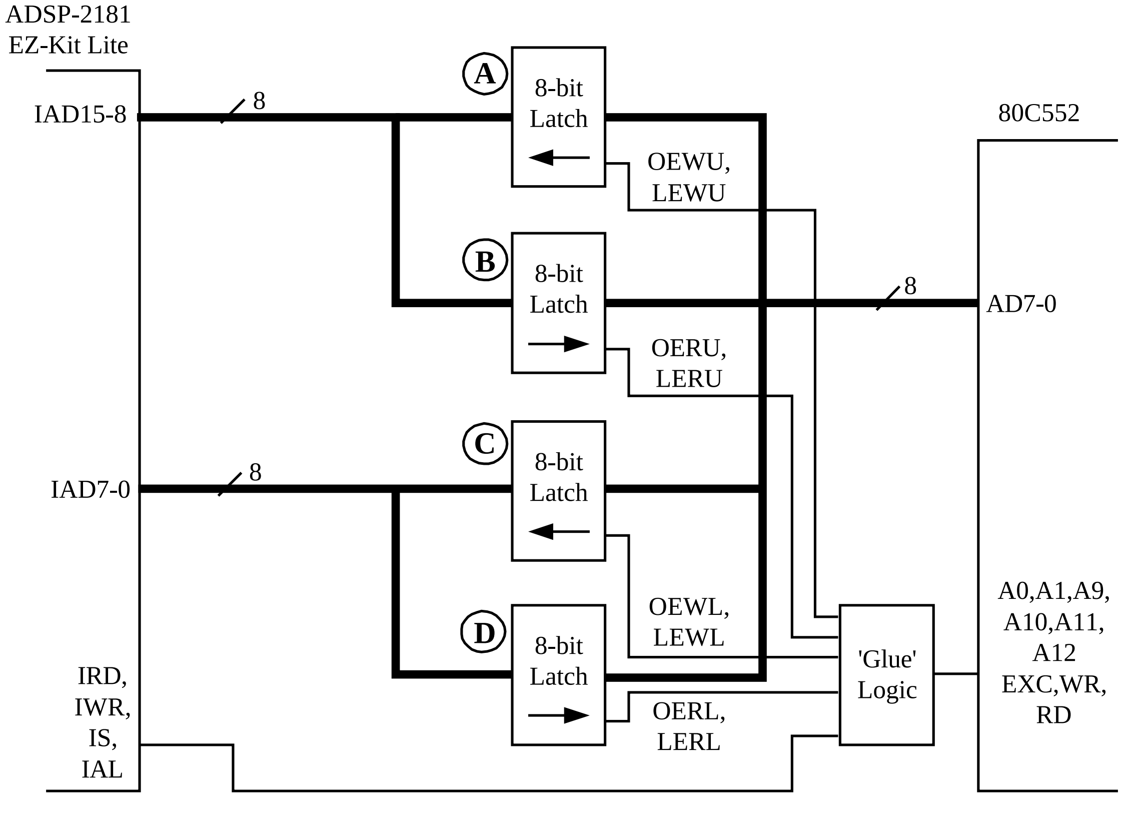

Figure 1 Block diagram of the 80C552 to ADSP-2181 IDMA port interface

The DFT Algorithm

An overview of the structure of the real-time DFT algorithm is shown in figure 2 below. The algorithm is an example of block processing and uses a ping-pong operation in which one data buffer is being processed through the DFT algorithm whilst the other is being refreshed at the sampling rate from the CODEC. The roles are switched when 128 new samples have been collected. Note that the DFT operations must have been completed before switchover in order to preserve real-time operation. At the output end of the algorithm the 128 values associated with the short term power spectrum of the signal are stored in one output data buffer whilst the previous block of DFT results stored in the alternate output buffer are transferred to the 80C552 via the IDMA port. Several blocks of memory, all of length 128, are required with each being accessed using the modulo addressing mode of the set of 8 pointer and modifier registers built into the pair of data address generators of the ADSP2181. Central to the DFT algorithm is the multiply/accumulate operation, so common to DSP algorithms.

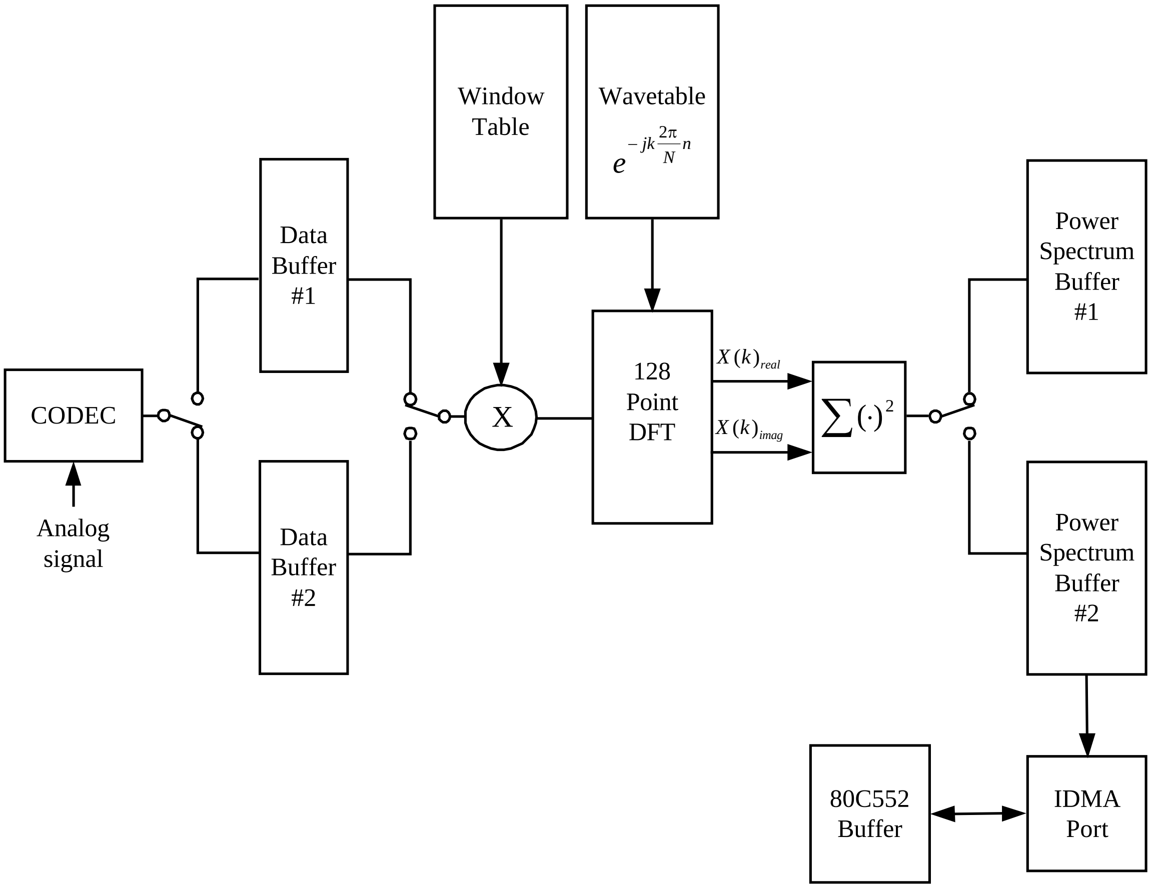

Figure 2 Real-time DFT Algorithm

LCD Hardware

The Seiko g1216 LCD[2] is a 128x64 pixel display that is configured in two halves with each side consisting of a 64x64 pixel block as shown in figure 3 below. A built-in controller is included along with two 8-bit wide RAM chips to hold the display data for each half. Each byte of data stored in the RAM represents the pixel pattern for a particular column on a particular page.

The 8 data lines and 6 control lines (RST, R/W, D/I, E, CS1, CS2) associated with the LCD are driven from two ports on the 80c552 controller. CS1 and CS2 are used to select each of the LCD

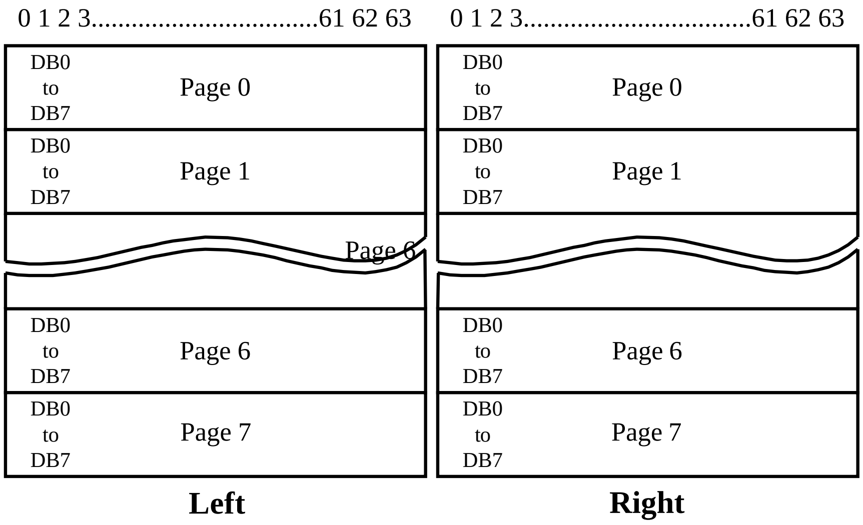

Figure 3 LCD Layout

RAM chips and the RST reset line turns the display off, and resets the row and column address counters to 0. The R/W line is used to determine whether data is written to, or read from the display RAM and the D/I line indicates data or instruction. If data is selected, the display RAM can be either read or written to. When instruction is selected, a new command can be written to the screen, or the status flags in the control registers can be read. Finally, the enable (E) line must be asserted in order to execute any read or write operation. Full handshaking using the LCD status flags was not used here since the PU-552 is much slower than the LCD on-board controller.

80C552 Micro-controller Software
This software consists of two parts namely IDMA control for reading DFT data and, secondly, DFT data processing into a form suitable for display on the LCD.

As described earlier, two circular buffers, each of length 128, are used, in ping-pong fashion, to store the power spectrum data generated by each successive implementation of a 128-point DFT. The key in synchronizing the extraction of one block of DFT data by the 80C552 while not compromising the real-time nature of the DFT algorithm is to transfer the starting address of the latest block of DFT data to the 80C552 immediately that the DFT algorithm has finished writing to that block, and to also allow sufficient time for the whole block of data to be transferred through the IDMA port to an array in the 80C552 before the DFT algorithm starts to overwrite the buffer contents. Since a 128-point DFT (not the FFT) takes a little over 0.5mS to implement on the 30nS ADSP2181 and it takes 6.38mS to collect the next 128 signal samples at 20.05kHz sampling frequency, then the 80C552 has approximately 12.25mS in which to extract 128 data words since every other DFT buffer is transferred for display on the LCD. Hence, because there is adequate time to make the transfer based on the available time and speed of the data transfer through the IDMA, the 80C552 simply polls the content of a variable in the ADSP2181 and waits until the variable changes from 0x0000 (DFT data not available) to the base address of the valid DFT data. Once this address is read it is echoed back to the IDMA control register and a counter loop controls the transfer of the 128 DFT data values to the 80C552 memory.

Having obtained the DFT data from the ADSP2181 the data must then be analyzed in order to generate the pixel pattern corresponding to a bar-graph representation of the DFT bin values. There are several approaches. Typically one might start by scaling (shifting) the original 16 bit data down to 6-bit resolution. Then select one half of the LCD to write to and commence to write 64 bytes of pixel pattern data to each page starting at page0. Pixel pattern data may be obtained by detecting those DFT data above the thresholds separating successive pages, outputting a pattern corresponding to the height of the data above the threshold and setting all patterns to 1 for the rest of that particular DFT bin. Patterns for those DFT data not above a particular threshold are set to 0 until this is no longer true.

Writing the pixel pattern data to the LCD requires that the screen be first reset thus zeroing the address counters and turning both sides of the screen off. Both halves of the screen must then be turned on and finally the page and column set to the address where writing is to start. Once the page and column have been set, data is simply written sequentially since with each write, the column counter is incremented automatically. After writing out the data for a page, the next page must be selected before continuing to write, to avoid overwriting the previous data. Having filled one half of the screen the other half must first be selected and can then be written to page by page as described previously. It recommended that one half be filled before switching to the other.

Conclusion

The implementation of the interface between the 80C552 micro-controller and the ADSP-2181 digital signal processor provides an opportunity for students to put to good use some of the experience gained in their first microprocessor class. The ability to exercise some control over the DSP algorithms they develop, or display information especially in the exercise described above, provides a genuinely motivating experience. Students benefit from the exposure to a multiprocessor solution to realistic applications that serves to demonstrate the complementary aspects to using two processors. Other applications include the control of audio effects such as reverberation and the control of parameters associated with a wave-table based function generator or with digital filtering applications such as graphic or parametric equalizers.

The laboratory exercise based on the DFT and display is typically spread over three 3-hour lab sessions with half of the time devoted to the DFT algorithm development and the rest for the display.

Bibliography

1. Analog Devices, ADSP-2100 Family User’s Manual. 1995.
2. Seiko Instruments, Modules with Built-in Data RAM

ANTHONY J. A. OXTOBY

Tony Oxtoby is an Associate Professor of Electrical Engineering Technology at Purdue University in West Lafayette, Indiana and was educated in England at the University of Hull and the University of Manchester Institute of Science and Technology. He developed the course in Digital Signal Processing now required at three Purdue campuses. In addition, he has presented DSP workshops for industry and academics.

CHRISTOPHER S. ARNDT

Chris Arndt is a graduate of the Electrical Engineering Technology department at Purdue University in West Lafayette, Indiana. He is currently employed with Xilinx Corporation as a development engineer.