Digital Signal Processing with the TMS320C31 DSK

Bill Bitler, Rulph Chassaing, Peter Martin
InfiMed/Roger Williams University/ECC International

Abstract

This paper introduces some applications in digital signal processing (DSP) using the Digital signal processing Starter Kit (DSK). The DSK is based on Texas Instruments' TMS320C31 (C31) floating-point processor. Available support tools for the DSK are described. A home-made wire-wrapped board was built to interface the Crystal CS4216 16-bit stereo audio codec to the C31-DSK, allowing a sampling rate of 48 kHz for audio signals. A home-made daughter board was also built with 32K words of zero wait sate SRAM and 128K of flash memory.

Introduction

DSP-based systems can be readily reprogrammed for a different application. A wide range of applications such as modems and speech recognition can be less expensive using DSP techniques. DSP techniques have become very successful because of the continued development and availability of low-cost hardware and software tools.

The TMS320C31-based DSK[1,2] is a relatively powerful, yet inexpensive ($99) development board for real-time digital signal processing. The DSK includes the C31, the TLC32040 analog interface circuit (AIC) chip for I/O, and comes with an assembler and debugger. Examples such as FFT and tone generation are included with the DSK. Programming examples using both C and assembly code for the TMS320C30[3] can be readily modified to run on the DSK.

DSK's based on the fixed-point processors TMS320C2x and TMS320C5x are also available. In this article, we will only discuss the recent DSK based on the floating-point C31.

The TMS320C31 DSK

The C31 is a member of the third-generation family of 32-bit floating-point digital signal processors capable of performing floating-point, integer, and logical operations. It has 2K words of on-chip or internal memory, and a 24-bit address bus, making it capable of addressing 16 million words (32-bit) of memory space for program, data, and input/output. With a 40 ns instruction cycle time, it provides capabilities for 50 million floating-point operations per second (MFLOPS) or 25 million instructions per second (MIPS). While there are two serial ports available on the C30, only one serial port is available on the C31. While the C31 has 2K words of on-chip memory, the last 256 words of internal memory locations in the C31 on the DSK are used for the communications kernel and vectors[1]. The DSK board communicates with the PC host through a dB25 parallel printer port interface connector cable.

The AIC has 14-bit ADC and DAC and variable sampling rate of up to 20 kHz. Higher sampling rates such as 44.1 kHz for audio applications can be achieved, although not to TI's specifications. It has two inputs and connects to the serial port on the C31. It includes a switched capacitor input filter for antialiasing (by-passable) and output reconstruction filter. All the C31 pins, the AIC I/O pins, and power are available through expansion connectors on the DSK board, which provides four 32-pin DIL footprints. This allows for additional circuitry such as external SRAM and flash memory, and alternative ADC converters which been connected to the C31's serial port.

The assembler provided with the DSK does not create a common object file format (COFF). A program in C (or assembly) can be compiled/assembled and linked to create an executable COFF file using the TMS320 floating-point tools[4-6]. The resulting COFF file must be properly linked so that it can be directly downloaded into the C31 on the DSK. Code is assembled at an absolute address using certain assembler directives to create an executable object file, and does not use a linker. An executable file can be loaded into the DSK using either the debugger or a bootloader. The C31 on-chip memory addresses 0x809800 and 0x809801 are avoided since they are used by the bootloader for stack space (they can be used after the communication kernel is downloaded).

Several assembler directives are used to control the starting addresses of different sections, thereby eliminating the need for a linker. For example, several source files can be appended using the .include assembler directive, as in C programming. The .include assembler directive serves the function of a linker to include or chain several files together.

Support Software and Hardware Tools

Interfacing the DSK to External SRAM and Flash Memory

A daughter board was built with 32 Kwords of SRAM and 128 Kbytes of flash memory. This daughter board connects to (fits underneath) the DSK through the four connectors JP2-3 and JP5-6 on the DSK. Appropriate signals (address, data, V+, GND, R/W, INT0-3, STRB) from the C31 used for the SRAM and the flash memory are available through those four connectors.

The address lines A22 and A23 (both low) provide the memory decode range 0-3FFFFFh. The external SRAM can be accessed in any 32K boundary within this range. A starting address of 100000h was chosen. The signal STRB from the C31 is used for a valid external memory address on the external bus. To obtain zero wait state operations, the access time is calculated to be faster than 15.5 ns. Four 32K x 8 bit SRAM with 15 ns speed rating for zero wait state operations were used. To test the SRAM, different patterns of data were written to memory locations 100000h-107FFFh (32K x 32) on the SRAM and then the same data read back.

The AM29F010 flash memory from AMD is divided into eight sections of 16 Kbytes each. The bytes can be programmed one byte at a time using EPROM programming technique of hot electron injection. A starting memory address of 400000h was chosen on the flash, which corresponds to the 2nd boot region in the C31 memory map. The flash was tested using 2 wait states. A specific application program can run from the flash without any connection to a PC.

Interfacing the DSK to the CS4216 Stereo Audio Codec

A CS4216 Evaluation Board was interfaced to the DSK. This board includes the CS4216 and contains jacks for line and microphone inputs. Figure 1 shows a partial schematic of a home-made wire-wrapped board which contains the Crystal's CS4216 16-bit stereo audio codec and supporting hardware for interfacing with the DSK. The CS4216 uses Delta-Sigma A/D and D/A converters with internal 64x oversampling, and internal input anti-aliasing and output smoothing filters. JP1 on the wire-wrapped board is connected to JP1 on the DSK through a ribbon cable connector. The interface provides communications and control for the CS4216 codec. SSYNC, SCLK, SDIN, and SDOUT pins are used for the serial communications interface with FSX0, FSR0, CLKR0, and DR0 pins on the C31. The interface includes the codec reset signal that is supplied from XF0 on the C31. The TCLK0 on the C31 is used as the codec master clock, which controls the A/D and D/A conversion rates. Different sampling rates can be achieved by setting the appropriate value in the period register, or

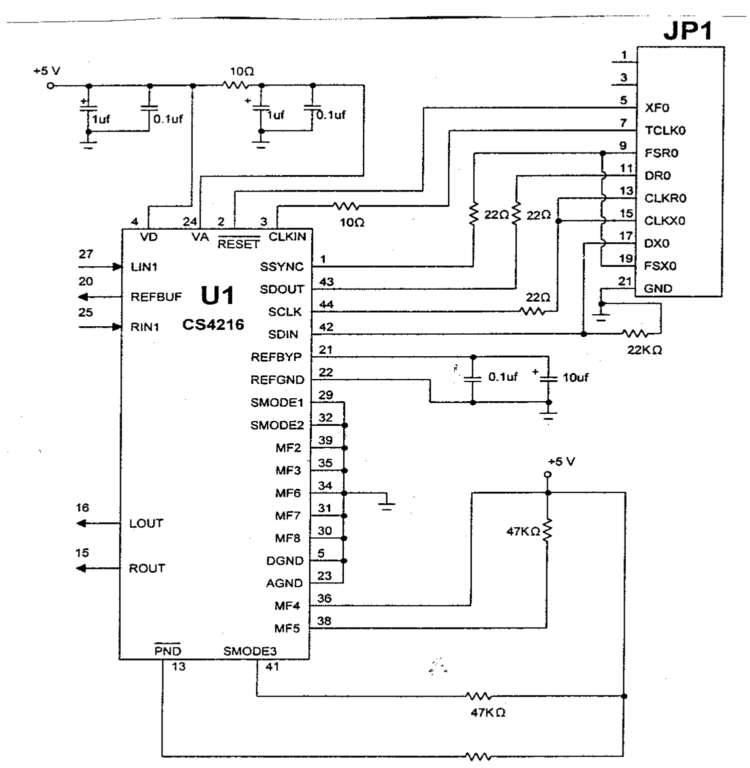

Figure 1. Partial DSK-CS4216 interface schematic

Fs = (Master Clock/4)/(Period register*256)

The sampling rate is 48,828 Hz by setting the period register with a value of one (Fs = 8,138 Hz with a period register set at 6). SMODE1-3 pins are set to allow 32-bit length serial port transfers; MF2-8 set to allow the maximum sampling rate and configure the codec control serial port for default control settings. The pins on the left of the schematic are connected to amplification control circuits for line level I/O, microphone input, and speaker output. LIN1 and RIN1 pins can be connected to either a 1/2-gain (regular or line input) or to a 16-gain (microphone input) amplifier using jumpers. The REBUF pin is used for biasing the amplifiers to allow input signals with no DC offset; and LOUT and ROUT are connected to a coupling circuit (line or regular output) and amplifier to drive a speaker.

Commercial Tools

A design package which interacts directly with the C31 DSK is available from **MultiDSP**, e-mail: multidsp@aol.com. FIR as well as IIR filters can be very quickly designed. Plots of poles and zeros, magnitude and phase, etc. can be obtained. Effects of filter length and window functions can be monitored. The designed filter can then be implemented readily (within seconds) in real-time on the DSK. It is reasonably priced at $40 for students.

Code Explorer, a debugger which supports the C31 DSK and available from **GoDSP**, is free through the web at http://www.go-dsp.com. GoDSP sells the popular Code Composer, with documentation. Code explorer is very similar to Code Composer but no documentation, does not accept a COFF file (Code Composer does). One can retrieve (send) an input (output) from (to) a file. Results from an output file can then be plotted in both time and frequency.

SigLab is a virtual instrument available from **DSPTechnology,** e-mail: sm@dspt.com. It interfaces to a PC via an SCSI connector. The SigLab includes a function generator, oscilloscope, spectrum analyzer. The function generator can generate random noise, two-tone sinusoid, arbitrary waveform, etc. One of three DSP chips used by SigLab is a TMS320C31. The maximum sampling rate is 51.2 kHz for a max. bandwith display of 20 kHz (fine for audio applications). The University price is reasonable at $3500 (considering the cost of a dedicated spectrum analyzer, for example).

RIDE is a software package available from **Hyperception**, e-mail:info@hyperception.com. It is a visual design tool which can be used to implement DSP algorithms. It contains many functional blocks for FFT, correlation, filtering, etc. and can be used for both simulation and real-time (Block Diagram is a simulation version). Results can be displayed on the PC monitor or to an external device such as a scope in real-time. The TMS320-based EVM can be used as a driver for real-time. It does not currently support the C31 DSK.

Implementation

Several assembly and C coded programs[3] have been slightly modified to run on the DSK.

1. FIR Filter in C and Assembly

a) The C coded FIR program[3] was modified to handle the interrupt vector used, before being compiled. An appropriate linker command file did not use the first two C31 internal memory locations since they are reserved on the DSK for the bootloader. This linker command file also specifies the memory location 0x809FC0 as the starting address for interrupt vectors. The following file which defines the interrupt used is also linked:

```
        .ref        _c_int05                    ;define XINT0 for interrupt
        .sect       "vecs"                      ;section for interrupt vectors
        br          _c_int05                    ;branch to use XINT0 for interrupt
        .end
```

The FIR program can then be compiled and linked to produce an executable COFF file which can be directly downloaded into the C31 on the DSK and run. Similarly, the mixed coded (C program calling an assembly function) FIR filter example[3] was also implemented with the C31.

b) The following partial FIR assembly coded program illustrates the use of a number of assembly directives:

```
        .start      ".text",0x809900            ;where text begins
        .start      ".data",0x809C00            ;where data begins
        .include    "AICCOMA.ASM"               ;include AIC communication routines
        .include    "BP55.COF"                  ;include coefficients file
        .data                                   ;data section
        .brstart    "XN_BUFF",64                ;align a buffer for input samples
XN      .sect       "XN_BUFF"                   ;buffer section for input samples
        .loop       LENGTH                      ;loop length times to
        .float      0                           ;initialize buffer samples to zero
        .endloop                                ;end of loop
```

The .start assembler serves the function of a linker file and used to specify the starting address of data and text sections. The .include is used as in a C program to "include" a file. The files to be included are "AICCOMA.ASM" which contains the AIC communications routines and the file "BP55.COF" which contains the coefficients of a specific bandpass FIR filter. A different filter can be readily implemented by "including" a different coefficients file in the generic FIR filter program. The .brstart assembler directive is used to create an aligned circular buffer. The buffer "XN_BUFF", creaated with the .sect assembler directive, is reserved for the input samples. The .loop directive creates a loop to initialize the input samples to zero. This method is preferable since an assembler directive is resolved during the assembling process and does not occupy program memory space as is done with an instruction.

2. PC Host/C31 Communication Example

This example illustrates communication between a C program executed by the PC host and an assembly program executed by the C31 on the DSK. The C host program, compiled with Turbo C++, prompts the user to enter a number between 1 and 8 which corresponds to one of eight sets of FIR filter coefficients to be selected (bandstop, 3 bands, etc). The value entered by the user is sent to the C31 assembly program which runs the selected filter. Communication between the PC host and the C31 can be accomplished using putmem ( ) to transmit the user's selected number (corresponding to the desired filter) to the C31-based DSK. Putmem ( ) and getmem ( ) are utility functions provided with the DSK to send data from the PC host to the DSK, and to receive data from the C31, respectively.

3. Adaptive FIR Filter

This example implements the adaptive filter structure described in references[3,7] for the cancellation of an additive noise. The AIC on-board the DSK contains two inputs. One input is used for a desired signal with an additive unwanted noise. A reference noise, correlated with the unwanted additive noise, uses the second AIC input (available through one of the expansion connectors on the DSK). The AIC communication routines[3] allow data to be accessed through both input ports. With voice as input, the filter is adapted so that the sinusoidal noise is cancelled. The adaptive filter was tested using different filter length and rate of convergence.

Conclusion

The C31-DSK provides the opportunity for DSP development with a floating-point processor at a relatively low cost. The C30, C31, and the lower-cost C32 (with less internal memory), and higher-cost TMS320C4x[8] (with 4 to 6 serial ports) which are code compatible with the TMS320C3x processors are currently being used in a wide range of applications from communications and controls to multimedia.

Acknowledgement

The support of the Roger Williams University Research Foundation (RWURF) is appreciated. Two National Science Foundation (NSF) equipment grants (1986, 1988) contributed to the development and enhancement of the DSP Lab at RWU. Two recent grants from NSF's Undergraduate Faculty Enhancement (UFE) Program (1996, 1997) to offer three DSP and Applications workshops at RWU for 60 faculty members further supported RWU's DSP Lab .

References

1. TMS320C3x DSP Starter Kit User's Guide, Texas Instruments Inc., 1996.

2. TMS320C31 Embedded Control Technical Brief, Texas Instruments Inc., 1992.

3. R. Chassaing, Digital Signal Processing with C and the TMS320C30, J. Wiley, 1992.

4. TMS320C3x User's Guide, Texas Instruments Inc., 1994.

5. TMS320 Floating-Point DSP Optimizing C Compiler User's Guide, Texas Instruments Inc., 1991.

6. TMS320 Floating-Point DSP Assembly Language Tools User's Guide, Texas Instruments, 1991.

7. B. Widrow and S. D. Stearns, "Adaptive Signal Processing", Prentice Hall, 1985.

8. R. Chassaing and P. Martin, "Parallel Processing with the TMS320C40", Proceedings of the 1995 ASEE Annual Conference, June, 1995.

9. R. Chassaing and Bill Bitler, " Digital Signal Processing for Video Line Rate Analysis", International Journal of Engineering Education, 1995.

10. Chassaing, J. Prime, V. Caramadre, R. Siipola, A. Malone, C. Foreman, R. Groppe, A. Ataya, "Student Projects on Digital Signal Processing with the TMS320C30", Proceedings of the 1995 ASEE Annual Conference, June, 1995.

11. R. Chassaing, W. Anakwa, and A. Richardson, "Real-Time Digital Signal Processing in Education", Proceedings of the 1993 International Conference on Acoustics, Speech and Signal Processing (ICASSP), April, 1993.

12. R. Chassaing, "The need for a Laboratory Component in DSP Education **-** A Personal Glimpse", Digital Signal Processing, Academic Press, January 1993.

13. R. Chassaing and B. Bitler (contributors), "Signal Processing Chips and Applications", The Electrical Engineering Handbook, CRC Press, 1993.

14. R. Chassaing, Digital Signal Processing with the TMS320C25, J. Wiley, 1990.

BILL BITLER is employed by InfiMed, in New York, as a Senior Systems Engineer working with imaging products and digital image processors for medical diagnostic imaging. He received his BS in Computer Engineering in 1992 from Roger Williams University. Bill authored and coauthored technical papers for the SPIE conferences on medical imaging, and coauthored several articles on DSP.

RULPH CHASSAING received the PhD (EE) from the Polytechnic Institute of New York. He is a professor at Roger Williams University, the author of the text *"Digital Signal Processing with C and the TMS320C30"* and coauthored with Dr. D. W. Horning *"Digital Signal Processing with the TMS320C25"*, both published by Wiley (1992, 1990). He published a number of articles on real-time applications in DSP.

PETER MARTIN received his BS in Computer Engineering in 1989 from Roger Williams University, and is a Software Engineer at ECC International Corp. where he is developing software for real-time training devices. He coauthored several articles on the TMS320 processors and contributed in the research for the texts, *"Digital Signal Processing with C and the TMS320C30"*, and *"Digital Signal Processing with the TMS320C25"*