Application of a TMS320C31 chip for DSP/Embedded System

Shuchen Kate Feng, Rolf Olsen, Phil Pietraski, and D. Peter Siddon

National Synchrotron Light Source, Brookhaven National Laboratory

Upton, NY 11973, USA

The TMS320C31 chip from Texas Instruments is ideal for DSP and Embedded System Applications. Its powerful instruction set, high speed and innovative architecture enhances the performance of DSP applications. Its JTAG IEEE 1149.1 emulation port and microcomputer/boot-loader function facilitate computer simulation, debugging and embedded-system development. To date, the speed of this low-cost chip ranges from 27MHz to 50MHz.

At NSLS, we have incorporated a TMS320C31 chip and an SN74ACT8990 test bus-controller with an open-target-interfaced emulation porting kit for VME-bus-hosted emulation and AT-bus-hosted emulation. Using the AT-bus-hosted emulator for development, we have designed and implemented a fifteen-channel-counting device for NSLS beam line data acquisition. The implemented device is a stand-alone counter which can be controlled via front panel keypads with display or by a host computer over an IEEE-488 interface. Some technical hardware and software details in designing the emulation as well as the counter will be described. Other commercial scan-path emulators for the TMS320C31 will be listed as alternatives which could be considered depending on a project time frame, financial plan and design needs.

1. Introduction

At the beginning of this project, we decided to update the processor used for the embedded systems developed by the NSLS Beam Line Group from a 2MHZ 6502 microprocessor we used in our in-house-designed motor controller [1] to a faster microprocessor. Making good use of the existing resources in NSLS was a sensible approach to the upgrade. NSLS had a site-wide-licensed emulation porting software kit [2] for the Texas Instruments TMS320 DSP family. The kit emulated the TMS320-based device through the JTAG IEEE 1149.1 emulation port in conjunction with an SN74ACT8990 test bus controller. The kit contains source codes, which allows users to apply emulation technology (i.e. scan and debug) directly to their product in diverse hardware and software platforms. In our application, we designed an AT-bus-hosted emulation board installed on a PC to interface the AT-bus with the JTAG bus. As to the emulation software, we compiled the emulation porting software on a 486 PC MS-DOS platform. We adapted the TMS320C31 chip to utilize its microprocessor mode during development for on-line emulation and its boot-loader mode for a stand-alone, embedded device. Using that emulation scheme, we developed a stand-alone, fifteen-channel-counting device for the NSLS beam line data acquisition.

2. The emulation hardware and software architecture

The emulation porting kit written in C language can be compiled easily under MS-DOS/Windows, OS2, SUN-OS, SUN-solaris, HP -UX, etc, by changing the compiling option in a makefile. The emulation hardware platform can be based on an AT bus, SBus, VME bus, and so on. A few definitions in header files can be tailored easily to fit different emulation hardware platforms. The TMS320C31 chip uses a modular port scan device (MPSD) technology to allow complete emulation via a serial scan path to the SN74ACT8990 chip through a 12-pin header. One needs to design interface hardware to communicate between the emulation hardware platform and the JTAG IEEE 1149.1 emulation port using a 44-pin surface-mounted SN74ACT8990 test bus controller[2]. The block diagram in Fig. 1 depicts the design concept of our TMS320C31 emulation hardware hosted on an AT-bus.

This in-house-designed emulation system is inexpensive and versatile, particularly in the situation when developers need to implement the application on diverse emulation hardware and software platform. However, there are other commercially-built emulators for the TMS320C31 on certain emulation hardware platforms. Texas Instruments markets DSP development hardware and software for PC and SCSI-hosted workstations. White Mountain DSP [3] markets emulators for PC and SBus-hosted Sun workstations. Those commercially built emulators provide convenience and time-saving for development on a fixed platform.

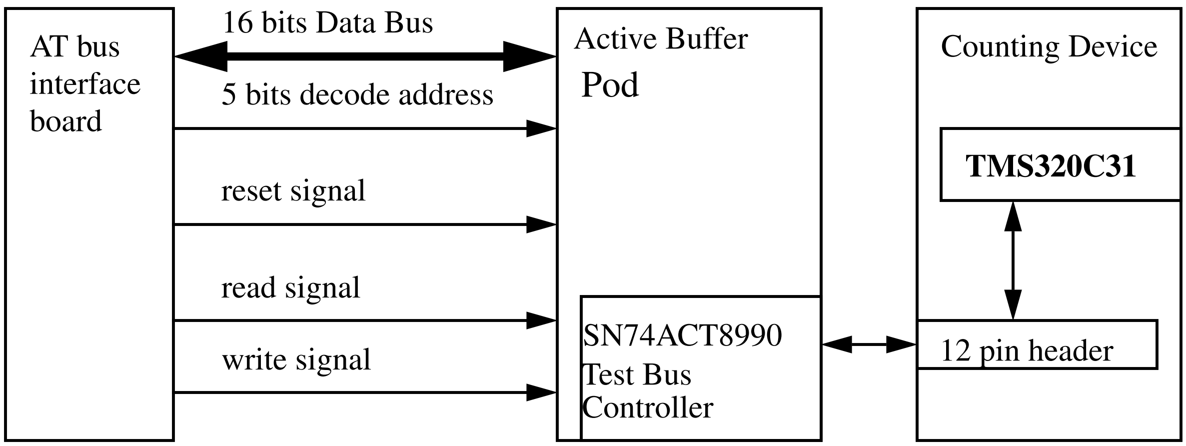

Figure 1: AT-bus-hosted emulation hardware for a TMS320C31 based device

All the emulation systems, custom-built or commercially-built, have similar graphical-user interfaces that help users to develop, test and refine C programs and assembly language programs. It provides effective and flexible functions such as breakpoints, single-step, user halt and memory display.

3. Features of the developed counting device

The developed fifteen-channel-counting device can be configured as one group of fifteen-channel counters, or three independent groups of five-channel counters. Each group can count a specified amount of time based on the internal clocks, or a specified amount of external events from any channel in the same group. The counters can be controlled via a front panel keyboard with display, or by a host computer over an IEEE-488 interface. The following is a list of the additional features:

- Flexible choice of a preset channel in any group
- Clocking time based in milliseconds from 20 to 1000000 (e.g. 10^6), or in seconds (default) from 1 to 10000000 (e.g. 10^7)
- Maximum external event counts is 0xffffffff (32 bits resolution ~= 4×10^9)
- Readout of counts during counting
- GPIB data transfer interrupt rate is capable of up to 1.2 MHz
- Overflow flag displayed on a front panel and readable through GPIB
- Maximum counting frequency for input is 4MHz

4. The design of the developed counting device

Combining the advantages of hardware, firmware, and software designs, a GPIB data transfer interrupt rate for the counting device of up to 1.2 MHz has been achieved. Figure 2 shows the prototype unit now in use on beam line X16B at the NSLS.

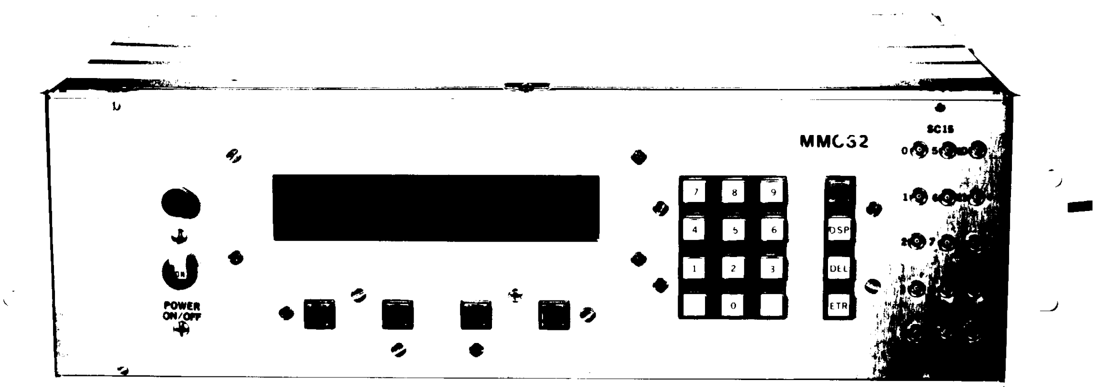

Figure 2: The prototype counter now in use on beam line X16B at NSLS

4.1 Hardware Design

We chose the 33MHz TMS320C31 chip as the system microprocessor not only because of its low-cost but also because of its performance. It is an ideal chip for DSP applications as well as general-purpose microprocessor-based applications. Its powerful instruction sets reduce the software overhead drastically by a factor of fifty as compared with the previous 2MHz microprocessor we used.

We use six surface-mounted AM9513A System Timing Controllers [4] to master the fifteen-channel counting. In addition, the system has several programmable logic devices[5] to program the counting synchronization, timing, overflow latching, device decoding and a wait state mechanism. The propagation delays of the programmable logic devices are all under 35 nanoseconds. Using PLDs with direct I/O, propagation delays are more favorable in this high-speed application. The wait-state mechanism was designed to handle slow timing peripherals such as the display unit and keypads which are used for local display and control.

A surface-mounted National Instrument GPIB-TNT4882 chip [6] on the board communicates between the counting device and a host computer via a GPIB connection. Its 8-bit 16-deep FIFO buffers data between the GPIB and the CPU, which increases the GPIB data transfer throughput.

All the hardware components are mounted on a 6.35 inch x 9.25 inch, six-layered board. The ground and 5-volt signal each occupy an individual plane. Figure 3 shows the hardware structure of the counter board.

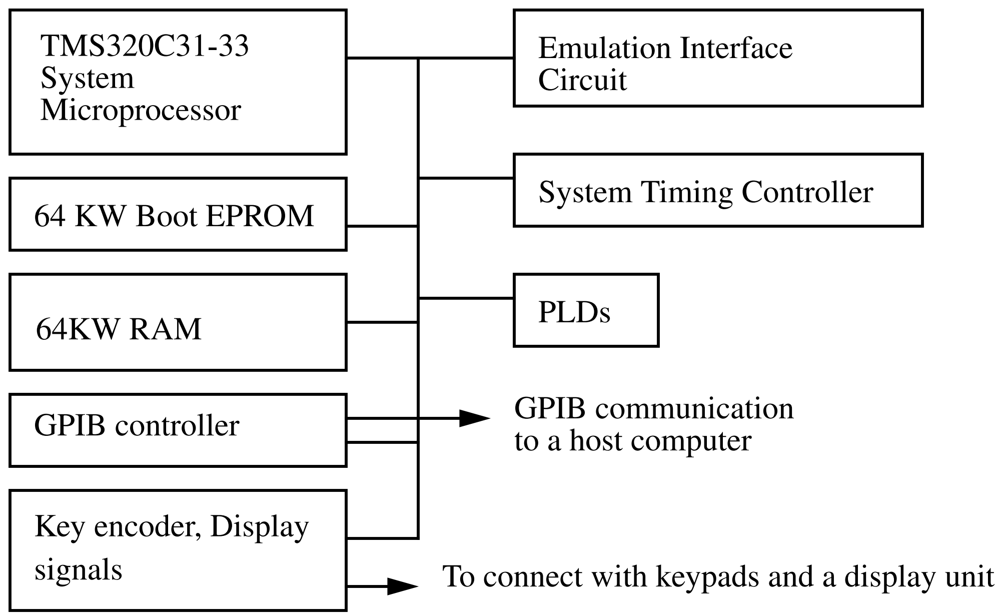

Figure 3: Hardware structure of the counter board

4.2 Firmware Design

There are four 64Kx8 27512 EPROMs and one 64Kx32 RAM module[7] with 20ns of access time. During the emulation/development stage the software is downloaded to the fast RAM directly from the PC host for on-line simulation and debugging. After development, the finished code and a bootloader were programmed into the inexpensive, slow, erasable EPROM. On power-up, the TMS320C31 downloads the codes resident on the EPROM to the fast RAM for real-time execution. The boot loading time for a 64K 32-bit word code is only 60 milliseconds.

4.3 Software Design

SPOX [8] is a real-time DSP operating system for TMS320 DSP chips. We decided not to use the SPOX OS because of its large memory requirements (43K 32-bit words) and the possibility of increased software overhead from features we did not need. Rather, a primitive microkernel is sufficient for our counter application. We wrote our own real-time kernel in assembly language in the time-critical lower layers (e.g. I/O drivers and interrupt handlers) to maximize efficiency. The use of C for the uppermost layers facilitated sophisticated function development without significant software overhead. The C crosscompiler from Texas Instruments includes mathematical functions, type-conversion functions and other general utilities in its library. However, it did not contain functions for stream I/O interfaces (e.g. scanf, and printf). One could get the source code from public domain sites or purchase the C compiler for TMS320C31 from Tartan [9]. Including kernels, stack pointers and dynamic variables, the complete codes for the counting device take 12.5K 32-bit words, which leaves plenty of room in the RAM for further development. Figure 4 shows the software development flow.

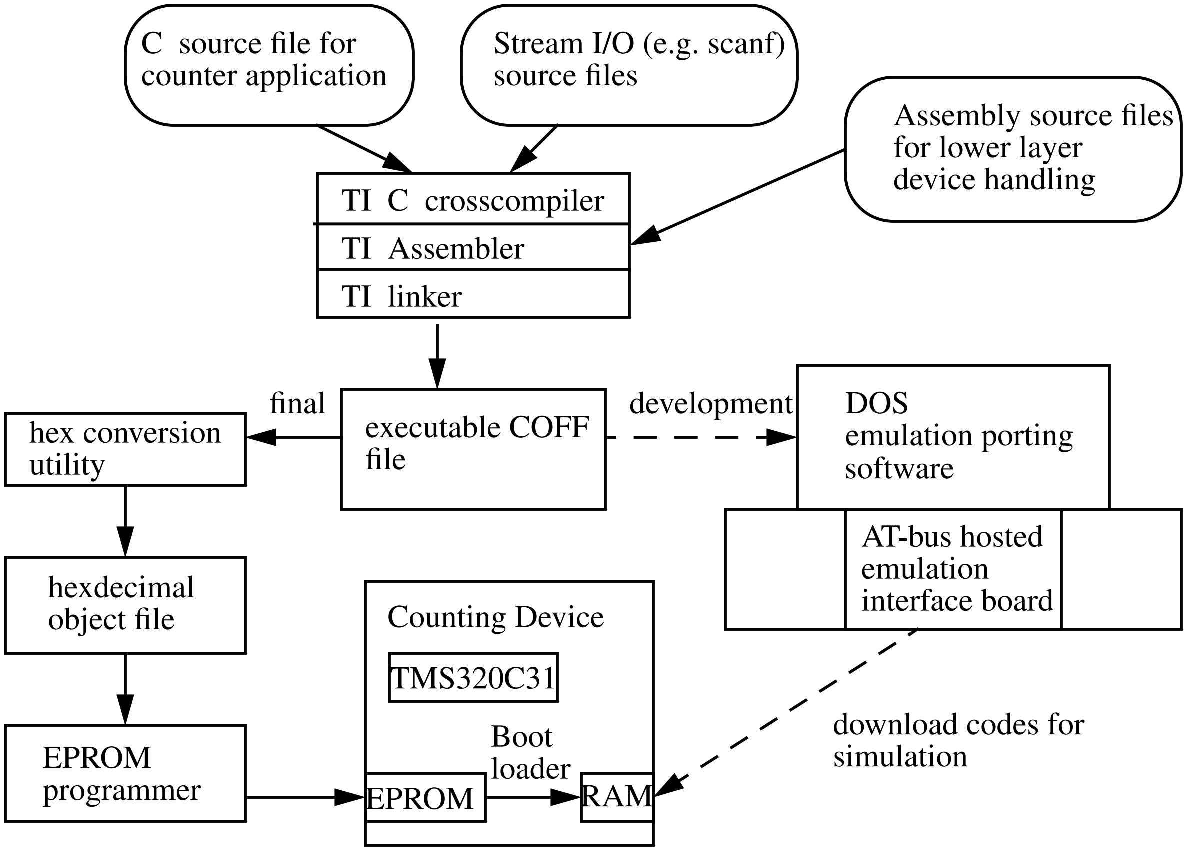

Figure 4: Software development flow of the counting device

5. Discussion

5.1 Emulation Scheme

The current emulation system has a restriction of 27K 32-bit words of memory for emulation due to the 640K memory limitation of DOS. If one uses Microsoft Windows, the amount of memory one can configure in the memory map corresponds directly to the amount of memory in the PC. However, one can easily migrate the AT-bus-hosted emulation board from a PC to an HP workstation equipped with an EISA slot. As soon as one recompiles the emulation porting kit under HP-UX, the application software for the counting device can be ported to HP-UX without many changes because of the portability of C. To do so, one needs to get a C crosscompiler for TMS320C31 running under HP-UX from TI. Migrating the emulation system to an HP workstation not only overcomes memory restrictions but also allows multi-tasking, which facilitates development and testing. The use of a NSLS PC-based UNIX system (e.g. LINUX [10]) would require TI to port their crosscompiler, which so far has not been done.

5.2 Expansion of the counting device

One can expand the counting device by adding our previously in-house-designed motor controller boards into the same chassis. An optimal solution is to design additional interface circuits to the motor boards on the counter board without modifying the motor boards. That will result in a device with a maximum configuration consisting of a fifteen-channel counter and a thirty-two channel motor controller.

Acknowledgments

We greatly appreciate Professor John Murray of Electrical Engineering Department of SUNY at Stony Brook for discussions about the emulation porting software. This work was supported by the U.S. Department of Energy under Contract DE-AC02-76CH00016 with Associated Universities, Inc.

References

[1] S. K. Feng, D. P. Siddons, "A Versatile Stepping Motor Controller For Systems With Many Motors", Nuclear Instrument and Methods in Physics Research A291, 1990, 422-426.

[2] Texas Instruments, U.S.A., (214)644-5580.

[3] White Mountain DSP, U.S.A., (603)883-2430.

[4] Advanced Micro Devices, (800)862-0020.

[5] Philips Semiconductors, U.S.A., (619)560-0242.

[6] National Instruments, U.S.A., (800)433-3488.

[7] Cypress Semiconductor, U.S.A., (408)943-2600.

[8] Spectron Microsystems Inc., U.S.A., (805)968-5100

[9] Tartan Compilers, available from Tartan Inc., (412)856-3600.

[10] S.K. Feng et al., "NSLS Beamline Control and Data Acquisition Computer-System Upgrade", these proceedings.