Digital Signal Processing Design Using TMS 320C5X Processor

Subra Ganesan
Department of Computer Science and Engineering
Oakland University, Rochester, MI 48309. U.S.A.
Phone: (810) 370 2206 Email: ganesan@oakland.edu

ABSTRACT

This paper describes the design of a Digital Signal Processing (DSP) system and a set of laboratory experiments to understand and /or teach real time system applications. Digital Signal processors have high performance and in chip integration and are highly suitable for real time control applications. TMS 320C5X is an advanced fixed point DSP processor and is widely used for communication, multimedia and control applications. This paper describes a top down approach for the selection of DSP processor, the hardware board design using TMS320C5X processor, software development using the custom board which can be interfaced with the personal computer (PC). A typical course on this topic starts with an introduction to the DSP chip and then provides a step by step instruction with hands on training using several lab experiments.

Some of the laboratory experiments described here are: FIR filter and IIR filter implementation, word recognition, touch-tone phone dialer and Fuzzy logic based engine control Application of DSP in Automotive control and use of application development tools such as Matlab, Filter design software design packages are also described.

INTRODUCTION

DSP processors are reduced instruction processors optimized for the fastest possible execution of addition, subtraction, multiplication and shifting instructions. DSPs have on-chip multipliers, compute a pipelined running sum of products and perform data scaling, memory shifts and pointer incrementing in parallel with the other operations. Advanced control algorithms requiring very short time loop can be easily implemented using DSPs.

Rapid advances in digital signal microprocessors (DSP) and application specific DSPs are causing enormous impact in the design of real time embedded control systems. Advances in circuit technology, architecture and algorithms, increased speed, availability of fixed point and floating point processors with on-chip DMA (direct memory access), timer, interconnect ports, pipelined and parallel arithmetic capabilities are contributing to novel applications for DSP chips.

Selection of DSP processor

A number of DSP processors are in the market. Some of the key manufacturers of DSP chips are Texas Instruments, Motorola, Analog Devices, and AT&T. Selection of a DSP processor for any application needs careful analysis. A top down approach to DSP selection is briefly given here.

1. Select DSP based on the application's performance requirements. DSPs are classified as fixed point or floating point. The processor selection is narrowed in choice to a few processors based on the width of the internal data bus (16,32,40 bits). The processors with higher data bus width normally have higher performance, but higher cost.
2. Select the processor based on the system interface factors such as, memory requirement (internal RAM, ROM, EPROM, external memory), ease of interface to external slow memory, internal or external analog input and output interface features, interrupts, internal timer, counter, and general or specific I/O pins.
3. Select the processor based on the availability of development tools such as assembler, simulator, hardware emulators, and evaluation boards.
4. Consider the cost, power consumption, second sourcing, third party software availability, printed circuit board layout, flexibility, application support, and software compatibility with future generation processors.

For most of the real time control applications a fixed point DSP processor is the ideal one. TMS320C5X generation processors consists of C50, C51, C52 and other processors. Since C5X processors are widely used and also easy to use, it is selected by the author for teaching the DSP system concepts.

The TMS320C5X processor

Some of the key features of this DSP processor are:

- 35-50 ns single-cycle fixed point instruction execution time.
- 2K x 16 single_cycle on-chip boot ROM
- 8K x 16 single_cycle on-chip program ROM
- 1056 x 16 dual-access on-chip data RAM
- 224K x 16 external memory (64K program, 64K data, 64K I/O and 32 K global)
- 32 bit arithmetic logic unit, 32 bit accumulator, 32 bit accumulator buffer
- 16 bit parallel logic unit
- 16 x 16 parallel multiplier with a 32 bit product capability
- single_cycle multiply and accumulate instructions
- 8 Auxiliary registers
- 11 context switch registers
- 8 level hardware stack
- 0 to 16 bit left and right shift barrel shifters and 64-bit incremental data shifter
- 2 indirectly addressed circular buffers for circular addressing
- Single instruction repeat and block repeat operations
- block memory move instructions
- Full duplex synchronous serial port for direct communication
- Time division multiple access serial port
- interval timer with period, control, and counter registers for software stop, start and reset

- 64K parallel I/O ports, 16 of which is memory mapped
- 16 software programmable wait state generators for program, data and I/O memory spaces.
- Extended hold operation for concurrent external DMA
- Index addressing mode, Bit-reversed index addressing for radix 2 -FFT
- 4 deep pipelined operation for delayed branch, call, and return instructions
- on chip clock, JTAG boundary scan logic

In the following section the details of a Custom C52 board is given.

DSP C52 board and DSP manager software

The C52 board provides a low cost method for developing software, debugging and evaluating the performance of DSP for the required application. The board has analog input and output interface, RS232 in/out for interfacing to a personal computer, 9V AC power input, 8k to 64 K RAM, Monitor/ Debugger EPROM, and expansion connectors. The TLC 320AC02 device provides the analog interface (14 bit ADC and 14 bit DAC, maximum sampling rate of 43.2 kHz for ADC and 25 KHz for DAC channel). I/ O board provides interface to 3 seven segment Display with drivers, 4 row x 3 column keypad for DTMF generation and DSP input, microphone and preamplifier circuit, cassette player input, audio amp and speaker with volume control, and digital output for motor control.

The Manager software provides communication from the personal computer to the DSP board. The presentation screen, which has eight areas and various pull down menus. Commands allow accessing files, Assembling the file, editing, loading the Hex file from the PC to the board., disassembling, memory dump etc.

Lab experiments:

A variety of laboratory experiments can be done with the lab station described here.
A suggested list of experiments are [1]:

1. Introduction to the board interfaces and software set up. Assembly language and Instruction set questions.
2. Interface to I/O like 7 segment display, key board, ADC, DAC.
3. Digital Oscillator, A-Law Compandor, Random number generator, All pole filter, All zero filter, Radix-2 butterfly, Pseudorandom bit sequence generator
4. Filter design using DFDP package and implementation of FIR filters, simulation of hearing impairment using real time FIR filtering
5. IIR filter design and implementation
6. FFT software development
7. Touch tone phone dialer and decoder
8. Vowel recognition

In this paper one of the experiments is given in detail and the others may be found in reference 1.

Pseudorandom Sequences:

The Pseudorandom sequences have the properties of random sequence. The random properties satisfied by these sequences are:

1. The balance property: In each run of the sequence the number of 1s differ from the number of 0s by at most one.
2. The run property: Among the runs of 1s and 0s in each period of the sequence, one half of the runs of each kind in a period of the sequence are of length 1, one fourth of length two, $1/2^k$ if length k, as long as these numbers have any meaning.
3. The correlation property: If term by term comparison of the sequence with a cyclic shift of itself is done, then the number of agreements will differ from the number of disagreements by, at most 1.

Basically the Pseudorandom encoder uses a shift register as shown in Figure 1. Table 1 gives the feedback connections in the sequential mode. The maximal length shift register sequence has a length $n = 2^K-1$ and the legendre code has a length $n=4m-1$ where n takes on any positive integer value such that n is a prime.

The students write a PRBS sequence generator program in DSP assembly language and down load it to the board and the output is passed through the analog interface board to the speaker. The output is the white noise sound through the speaker. In the following section the application of DSP for real time Fuzzy logic based control of engine idle speed is given.

FUZZY CONTROL

Fuzzy control has been applied successfully to a wide variety of difficult control problems. The advantages of fuzzy control are:

- Complete mathematical knowledge of the controlled object is not required.
- Provides great flexibility and simple solutions to complex problems which needs descriptive and intuitive thinking.
- The fuzzy controller is implemented with simple IF - THEN rules.

In fuzzy control, values for inputs and outputs are defined as fuzzy sets. The fuzzy sets are described by membership functions as illustrated in Figure 2. The fuzzy set values are labeled NM (negative medium), NS (negative small), Z (zero), PS(positive small) and PM(positive medium).

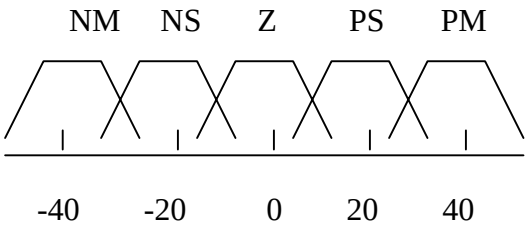

Figure 2: Fuzzy Membership Function

The inputs to a fuzzy controller are assigned to a fuzzy variable with a degree of membership given by the membership functions. The degree of membership is in the range of 0 to 1. A degree of 0 implies that input value does not belong to the set and a value of 1 implies that the input value belongs to the set entirely. The input value can also have a degree between 0 and 1. For example, in Figure 2, an input value of 10 would have a degree of membership in both Z and PS of 0.5 and a degree of 0 for other membership functions. On the other hand, an input value of -40 would belong to NM with degree of membership 1 and 0 for other membership functions. This process is called fuzzification. The typical membership function shapes are triangular, trapezoidal and exponential. The shape of the membership functions affects the time and space requirements of the microcontroller. The next process is Rule Evaluation. The fuzzy controller consists of a set of IF - THEN rules of the form:

IF {input conditions}
THEN {action on output }

The rules implicitly expresses the knowledge of the system in consideration in an intuitive manner. After applying all of the fuzzy rules to a given set of inputs the output will also belong to more than one membership function with different degrees of membership. The next step is called defuzzification which involves computing the crisp output value. The output value is a weighted average of the fuzzy output sets. The functional diagram of the fuzzy controller is shown in Figure 3.

FUZZY CONTROLLER IMPLEMENTATION

The inputs to the fuzzy controller are RPM error and Error rate. RPM error is the difference between current and set idle speed. Error rate is the difference between current and previous engine speed. The outputs of the fuzzy controller are Stepper Motor Position (air flow) and Spark Advance (SA).

The stepper motor output has a wide dynamic range (0-3000 rpm), but its action on the idle speed is retarded by the intake manifold capacity as well as the intake and compression phases. The spark advance output has an immediate action on the compressed mixture in the combustion chamber, but only a limited dynamic range (+/- 100 rpm).

Figure 4 shows a fuzzy K-map which maps out the rules for the fuzzy controller. The column corresponds to the fuzzy set RPM Error and the rows correspond to the fuzzy sets Error Rate. The entries in the cells of the fuzzy K-map are the fuzzy sets for the output spark advance. For example,

If RPM error is NM and Error rate is NS Then Spark Advance (output) will be PS. A similar table exists for Stepper motor position. The output is then calculated using the expression

$$\text{output} = \frac{\sum \text{centroids x weights}}{\sum \text{weights}}$$

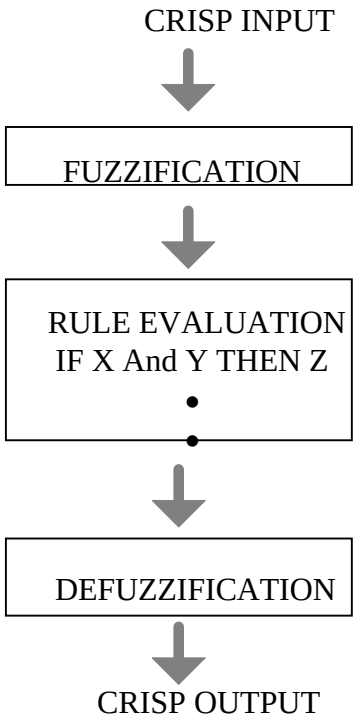

Figure 3 Functional Diagram of Fuzzy Controller

RPM Error

Error Rate	NM	NS	Z	PS	PM
NM	PM	PS	PM	PS	PS
NS	PS	PS	PS	Z	Z
Z	PM	PS	Z	NS	NM
PS	Z	Z	NS	NS	NS
PM	NS	NS	NM	NS	NM

Figure 4 Fuzzy K-Map

SOFTWARE AND TESTING

The fuzzy controller software is written in Assembly Language to run on 68HC11 microcontroller and DSP TMS320C50 processors. The algorithm is tested in the lab using MATLAB and tested with simulated input/outputs to the microprocessors. The fuzzy controller is also tested on a hardware loop simulator(Engine Simulator).The Engine speed was plotted for no load and for torque. We obtained a better response with the Fuzzy controller when compared to the response of a conventional controller which uses lookup tables. Performance on DSP processor was faster and accurate because of its

special features like: internal hardware multiplier, 32-bit accumulator, single cycle instructions, multiply and accumulate instruction, and repeat instructions.

SUMMARY

This paper describes the design of a Digital Signal Processing (DSP) system and a set of laboratory experiments to understand and /or teach real time system applications. The lab experiments and the real time control application are chosen to provide the proper understanding of DSP processor based systems. DSP processors are ideally suited for Fuzzy real time controllers.

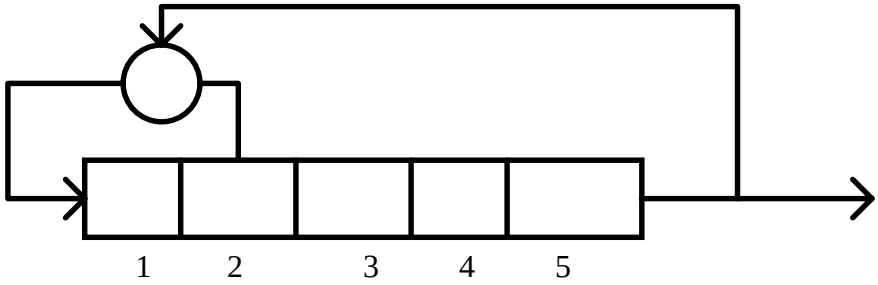

Figure 1. A Pseudorandom sequence encoder using sequential circuit.

TABLE 1 Pseudorandom Binary sequence generators-feedback tap connections

Register length	Feedback tap	Sequence length
2	1,2	3
3	2,3	7
4	3,4	15
5	3,5	31
6	5,6	63
7	6,7	127
8	2,3,4,8	255
9	5,9	511
10	7,10	1023
11	9,11	2047
12	2,10,11,12	4095
13	1,11,12,13	8191
14	2,12,13,14	16383
15	14,15	32767
16	11,13,14,16	65535
17	14,17	131071
18	11,18	252143
19	14,17,18,19	524287
20	17,20	1048575

References

1. S. Ganesan, R. Sudhakar, Raj Shah, “Digital Signal Processing using TMS 320C5X,” Advanced Microcomputer System education division, 1996, ISBN 0-9642962-5-X
2. S. Ganesan “ Advanced digital signal processors and automotive applications,” ASEE annual conference proceedings, Session 1620, pp. 300-305, 1992 .
3. Tretter, “ Communication system design using DSP algorithms” Plenum Publishing, 1995.
4. T.W. Parks and C.S.Burrus, “ Digital Filter Design “ Wiley.
5. J.G. Proakis and D.G. Manolakis, “ DSP principles, algorithms and applications “ 2nd edition, Macmillan.
6. M.J.T Smith and R.M. Mersereau, “ Introduction to DSP - A computer laboratory text book” John Wiley.
7. R.M. Merseraeau and M.J.T.Smith, “ Digital Filtering - a computer laboratory text book “ John Wiley.
8. P. Papamichalis, “ A practical approach to speech coding”, Prentice Hall.
9. Ralph Chassiang, “ DSP with C 30 “ John Wiley.
10. Digital control applications with TI TMS 320 processors, Texas Instruments
11. Applications books Vol. 1, 2, 3 for TMS 320 by Texas Instruments
12. Ahmed and R. B. Kline, “ Recent advances in DSP systems “ IEEE communication Magazine, May 1991.
13. Takahashi et. al, “A Simple Engine Model for Idle Speed Control,” SAE 850291, 1985.
14. Woods, Robert, Et. Al, “Development of an Idle Speed Control System for Spark Ignition Engines,” Proceedings of the 31st Conference on Decision and Control, 1982.
15. Sans Mariano, “A Second Order Idle Speed Controller,” SAE880184, 1988.
16. Hrovat, D., Bodenheimer, B., “Robust Automotive Idle Speed Control Based on μ-synthesis,” Proceedings of the American Control Conference, 1983.
17. Morris, R.L., Powell, B.K., “Modern Control Applications in Idle Speed Control,” Proceedings of the 1983 American Control Conference, Vol. 1, 79-85, June, 1983.
18. Abate, N. Dosio, “Use of Fuzzy Logic for Engine Idle Speed Control,” SAE900594, 107-114.
19. Murukami, S., “Application of Fuzzy Controller to Automobile Speed Control System,” IFAC 1983, 43-49.
20. E. Haskell, “Object Oriented Implementation of Fuzzy Control,” Tech. Rep. No. 9501-1, Oakland University, USA, March, 95.
21. Puskorius, G.V., Feldkamp, L.A., “Automotive Engine Idle Speed Control with Recurrent Neural Networks,” Proceedings of the American Control Conference, 1983, 311-316.
22. Pedrycz etal., “A reconfigurable Fuzzy Neural Network with In-Situ learning”, IEEE Micro, August 95, pp. 19-30