Performance Analysis of Line Echo Cancellation Implementation Using TMS320C6201

Zhaohong Zhang
Gunter Schmer

October, 1997

1. Introduction

This memo summarizes the results of a performance analysis on the LMS filter based line echo canceler (LEC) implementation using TMS320C6201 digital signal processor. The CPU throughput, internal data memory, and power dissipation requirements for processing are discussed.

2. Overview of Line Echo Cancellation

Telephone calls are open subjected to distortion or echo as they go through various network components. The primary cause of line echo is a an analog device called hybrid. Due to the electric current leakage in the hybrid, a part of the signal energy is reflected back to the source of the signal, which causes the talkers on each end of the connection hear an echo of their voices.

To improve the quality of telephone conversation, a line echo canceler needs to be placed in the network. In a line echo canceler, a transversal FIR filter is typically used to predict the echo from the history of the far-end signal, and the echo residue is calculated as:

$$e(n) = d(n) - \sum_{k=1}^{M} H_k(n)x(n-k) \qquad (1)$$

where *e(n)* is the value of residue at time *n*, *d(n)* is the value of echo at time *n*, $H_k(n)$ is the *k*th filter coefficient at time *n*, and *x(n-k)* is the value of the far-end signal at time *n-k*. *M* is the length of the filter, which is determined by the echo tail length.

The filter is usually updated by the LMS (least mean square) algorithm as:

$$H_k(n+1) = H_k(n) + \mu e(n)x(n-k) \qquad (2)$$

where $\mu \geq 0$ is the adaptation step size.

For the leaky LMS algorithm, the filter is updated by:

$$H_k(n+1) = \beta H_k(n) + \mu e(n)x(n-k) \qquad (3)$$

where $0 \leq \beta \leq 1$ is the leaky factor, which is introduced to gain more control of the filter response.

The performance requirements of a line echo canceler is given in ITU G.165/G.168 specifications. It generally requires that the echo residue is reduced to 30 dB below the far-end signal level at the convergence of the cancellation filter.

3. ‘C6201 Performance in Echo Cancellation

The TMS320C6201 currently delivers up to 1600 MIPS at 200 MHz with a roadmap to double this performance by the end of the decade. This high performance solution is driven by the key features such as dual data paths from 8 functional units including 2 multipliers and 6 arithmetic units allowing execution of 8 32-bit instructions in parallel, and 5 DMA channels with automatic address generation features. Those key features enable TMS320C6201 to deliver up to 10 times the performance of the previous DSPs, providing an ideal solution for multi-channel telephony applications such as line echo cancellation.

In the analysis of TMS320C6201 performance in line echo cancellation, we assume a normal sampling rate of 8 KHz (125 us looptime), and 16 bit filter coefficients. The performance is primarily limited by the available internal data memory (64 Kbytes) and CPU throughput. Multi-channel operation and variation of echo tail length have no effect on the small program memory required for the LMS algorithm. The effectiveness of using 16 bit filter coefficients are discussed in another application report.

3.1 Data Memory Requirement

In general, the data memory requirement for processing a channel with N ms echo tail is shown in the table below. We assume that the algorithm is based on a normalized LMS or leaky LMS. For completeness we also reserved 30 variables per channel for Modem Tone Detection, Phase Reversal Detection, Double Talk Detection, and Non-Linear-Processing (NLP). For the existing code, Modem Tone Detection uses 16 variables and the data memory for the rest procedures are negligible.

Processing Variables	30 (16 bit)
Circular Data Buffer	$2^K \geq 8N$ (16 bit)
Filter Coefficients	$8N$ (16 bit)
Total	$8N + 30 + 2^K$ (16 bit)

Table 1: Data memory requirement for a channel with N ms echo tail.

The data memory required for each channel to process echoes with tail length of 32 ms, 48 ms, and 64 ms are listed in Table 2.

	32 ms Echo Tail	48 ms Echo Tail	64 ms Echo Tail
Processing Variables	60 bytes	60 bytes	60 bytes
Circular Data Buffer	512 bytes	1024 bytes	1024 bytes
Filter Coefficients	512 bytes	768 bytes	1024 bytes
Total	1084 bytes	1852 bytes	2108 bytes

Table 2: Data memory requirement per channel.

Table 3 illustrates the maximum number of echo cancellation channels that can be implemented on 200 MHz ‘C6201’s 64 Kbytes (65,536 bytes) internal data memory. In calculating the data in Table 3, we reserve 1000 byte for data I/O buffer. Again, the table reflects the data requirements for normalized LMS/leaky LMS, Modem Tone Detect, Phase Reversal Detect, Double Talk Detect, and NLP.

	32 ms Echo Tail	48 ms Echo Tail	64 ms Echo Tail
Number of Channels	59	34	30

Table 3: Maximum number of channels can be implemented on internal data memory.

3.2 CPU Throughput Requirement

There are two existing LMS codes implemented on ‘C6201. One handles normal LMS algorithm[1], and the other one performs leaky LMS algorithm[2]. Their throughput characteristics are listed in Table 4.

	Throughput of Existing Code	Ideal Throughput
LMS	1.125M + overhead (cycles)	1.0M + overhead (cycles)
Leaky LMS	1.5M + overhead (cycles)	1.5M + overhead (cycles)

Table 4: Throughput characteristics of the existing ‘C6201 LMS code

where M is the number of the filter taps.

It can be seen from Equation 1 and 2 that two multiplies are required for processing each LMS filter tap. One is for calculating the echo prediction, and the other one is used to update the filter coefficients. If two multiply units are used simultaneously, the ideal throughput for the LMS algorithm will be M plus overhead. Considering the limitation due to the available number of registers on the chip, we think that the throughput achieved by the existing code (1.125M) is a realistic upper limit. It can also be seen from Equation 3 that the leaky LMS algorithm requires one more multiply for filter update, which makes the ideal throughput for the leaky LMS algorithm to be 1.5M. The existing code has reached the ideal limit.

The ‘C6201CPU throughput time required for each echo canceler channel is listed in Table 5. In Table 5, we assume a processing overhead of 100 cycles for Modem Tone Detect, Phase Reversal Detect, Double Talk Detect, NLP, and loop initialization codes. This is a very conservative assumption and is consistent with the code we used in the simulations. For the code we used in simulation, NLP takes 16 cycles and Modem tone Detection takes 26 cycles.

	32 ms Echo Tail	48 ms Echo Tail	64 ms Echo Tail
LMS	388 cycles	532 cycles	676 cycles

Leaky LMS	484 cycles	676 cycles	868 cycles

Table 5: CPU throughput requirement for each channel.

Table 6 displays the maximum numbers of echo canceler channels that can be implemented given the 200MHz ‘C6201CPU throughput and zero wait state memory.

	32 ms Echo Tail	48 ms Echo Tail	64 ms Echo Tail
LMS	64	46	36
Leaky LMS	51	36	28

Table 6: Maximum number of channels can be implemented due to throughput limit.

Combining the internal data memory analysis in Table 3 and throughput analysis in Table 6, we determine the upper bound of the number of LEC channels a single ‘C6201 can support as in Table 7.

	32 ms Echo Tail	48 ms Echo Tail	64 ms Echo Tail
LMS	59	34	30
Leaky LMS	51	34	28

Table 7: Maximum number of channels can be implemented on ‘C6201.

3.3 Power Dissipation

The power consumption analysis is based on the TI white paper “TMS320C6201 Projected Power Dissipation on TI’s TImeLine™ Technology”[3].

Assume that the only I/O function is the serial port and the host interface. Since the power consumed in a serial port is very small (< 20 mW @20 MHz) and the host interface activity usually occurs at very low rate, the I/O power consumption is considered negligible in this analysis. The power consumption in ‘C6201 is primarily determined by the following factors:

- Level of CPU activity (number and type of instructions in parallel),
- On-chip program memory access rate,
- On-chip data memory access rate,
- Level of activities of the “other” circuits (peripherals, host interface, external memory interface, etc.).

In order to characterize the power consumption, the execution of a program can be expressed in terms of combination of very high, high, and low levels of activity. The very high DSP activity model fits the intensive operation within a loop and is defined as

CPU is running 6-8 instructions in parallel.
Program memory access is 100%, one fetch every cycle.
Data memory fetch is greater than 90%.

“Other” activities are 1.4 times baseline.

The “baseline” activity is the activity due to the clock switching with all other signals in static state. The high DSP activity model fits the intensive operation within a loop with less memory access and is defined as

CPU is running 6-8 instructions in parallel.
Program memory access is greater then 90%, about one fetch every cycle.
Data memory fetch is about 50%.
“Other” activities are 1.2 times baseline.

The low DSP activity model fits the codes for control and initialization and is defined as

CPU is running 2-4 instructions in parallel.
Program memory access is no more than 40%.
Data memory access is no more than 20%.
“Other” activities are in the baseline.

The average power consumption of these two modes at 200MHz for the Rev. 3 device (1.8V core) is listed in Table 8. One can see that the maximum power consumed by a single ‘C6201 will be less than 2 watts.

DSP Activity	Very High[1]	High	Low
CPU	0.686 W	0.686 W	0.368 W
Program Memory	0.205 W	0.185 W	0.073 W
Data Memory	0.279 W	0.155 W	0.015 W
“Other” Circuits	0.823 W	0.686 W	0.588 W
Total	1.993 W	1.71 W	1.04 W

Table 8: Power data for the two DSP activity modes.

In this analysis, we treat DSP activity during the LMS/leaky LMS loops as “very high” and others as “low”. For the LMS algorithm, there will by 1.125 *M* cycles (looping) in “very high” activity and 100 cycles (overhead) in “low” activity. Consider the 125 us (25000 cycles) loop time, the power required to process one echo channel is given by

$$P_{channel} = \frac{1.125M}{25000} P_{very\_high} + \frac{100}{25000} P_{low}$$

where *M* is the number of filter taps. For the leaky LMS algorithm, there will by 1.5 *M* cycles (looping) in high activity and 100 cycles (overhead) in low activity. The power required to process one echo channel is given by

[1] The power estimate for the “very high” activity mode is extrapolated from the data for the ‘high” and ‘low” activity mode.

$$P_{channel} = \frac{1.5M}{25000} P_{very\_high} + \frac{100}{25000} P_{low}$$

The average power consumption per channel for processing 32 ms, 48 ms, and 64 ms echo tails are listed in Table 9.

	32 ms Echo Tail	48 ms Echo Tail	64 ms Echo Tail
Power Per Channel (LMS)	27.1 mW	38.6 mW	50.1 mW
Power Per Channel (leaky LMS)	34.8 mW	50.1 mW	65.4 mW

Table 9: The average power consumption for each LEC channel.

3.4 Board Level Consideration

The maximum performance parameters of a single ‘C6201 in line echo cancellation are summarized in Table 10. The spare DSP throughput is defined as the percentage of the loop time (125 us).

Algorithm	Echo Tail (ms)	Channels/ DSP	Spare Data Memory[2]	Spare DSP Throughput	Power /DSP
LMS	32	59	579 bytes	8.4 %	1.599 W
LMS	48	34	1,568 bytes	27.6 %	1.312 W
LMS	64	30	1,296 bytes	18.9 %	1.503 W
Leaky LMS	32	51	9,252 bytes	0.0 %	1.775 W
Leaky LMS	48	34	1,568 bytes	8.0 %	1.703 W
Leaky LMS	64	28	5,512 bytes	0.0 %	1.831 W

Table 10. Maximum ‘C6201 performance parameters in line echo cancellation.

[2] Additional 1,000 bytes of the internal data memory are reserved as buffer for data I/O.

Channels/ DSP	Channels/ Board	Spare Data Memory/DSP	Spare DSP Throughpu t	Power/ DSP	Power/ Board	Channels/ in^2
34	850	1,568 bytes	8.0 %	1.703 W	42.575 W	8.854
32	800	5,272 bytes	17.5 %	1.603 W	40.075 W	8.333
30	750	8,976 bytes	22.5 %	1.503 W	37.575 W	7.812
28	700	12,680 bytes	27.5 %	1.403 W	35.075 W	7.292
26	650	16,384 bytes	32.6 %	1.303 W	32.575 W	6.771
24	600	20,088 bytes	37.8 %	1.202 W	30.050 W	6.250

Table 11. Echo cancellation performance for 48 ms tail from a board with 25 ‘C6201.

Due to the small package size of TMS320C6201 (35mm by 35 mm), power requirement is usually the primary limiting factor to how many ‘C6201 can be put on the board. Consider a board with 25 ‘C6201 and that the leaky LMS algorithm with 16 bit filter coefficients is used to process 48 ms echo tails, the echo cancellation performance delivered by this board is illustrated in Table 11. Assume the spacing between the processors is 13mm, the size of this board is about 96 square inches. With this board, a user can support leaky LMS echo cancellation anywhere from 600 channels (and have 37.8 % spare throughput) to 850 channels (and have 8 % spare throughput).

4. System Considerations

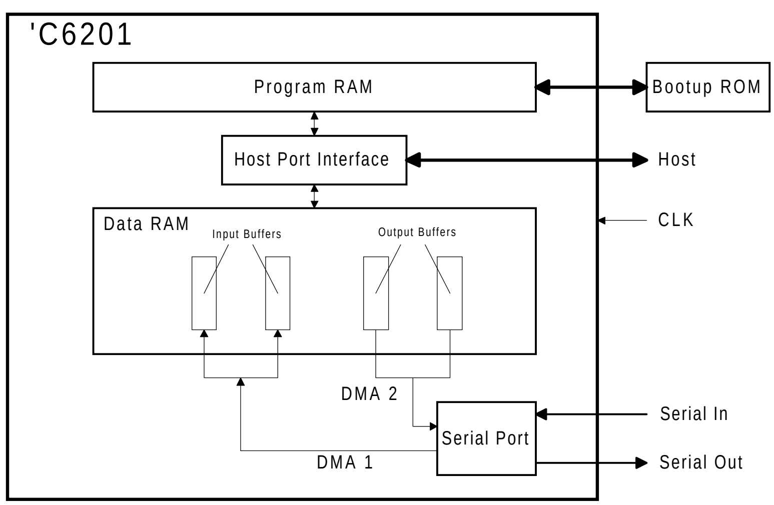

Figure 1: A block diagram for system implementation.

Figure 1 illustrates some thoughts of system implementation on a single ‘C6201. The processing should be synchronized with the input data frame. This can be realized by starting processing serving the interrupt on detection of the serial data frame synchronization pulses. Data I/O is handled through two double buffers. On the receive end, the data is passed from the serial port to one buffer through one DMA channel, while the CPU is processing the data saved in another buffer during the past data frame. On the transmission side, DMA is moving the processed data from one buffer to the serial port, while CPU is saving data to another buffer. This approach will result in a 250 us (two frame) delay between the input and output. The channel switching can be accomplished by directing CPU to read data from predetermined addresses in the input buffer and write to a predetermined address in the output buffer.

5. Summary

This analysis shows that TMS320C6201 offers a high performance and cost effective solution to multi-channel line echo cancellation. A processor board with 25 ‘C6201 can process up to 850 echo channels of 48 ms tail. Compared to other dedicated processor for echo cancellation, ‘C6201 provides much more flexibility in algorithm design so that users can tailor the implementation to meet the special requirements for their application. The existing ‘C6201 LMS and leaky LMS modules will also greatly reduce the software development time.

6. References

[1] LMSFIR8, ‘C6xx assembly benchmark from TI’s external web site.

[2] “Implementation of Echo Control for G165/DECT on Texas Instruments TMS320C62xx Processors”, Texas Instruments application report, August 1997

[3] “TMS320C6201 Projected Power Dissipation on TI’s TImeLine™ Technology”, Texas Instruments White Paper by Linda Hurd, October 1997