



PowerPC™

Application Note **Echo Canceller Implementation with Motorola AltiVec™ Technology**

*Mike Phillip and Perry He
risc10@email.sps.mot.com*

This document contains the following parts:

Part	Page
• Part I, "Overview"	1
• Part II, "Echoes and Echo Cancellation"	2
• Part III, "Echo Cancellation Algorithm"	4
• Part IV, "G.165 and G.168 Recommendations"	8
• Part V, "AltiVec Implementation Design"	11
• Part VI, "Performance Analysis"	17

Part I Overview

Echo in the telephone network is a well-known phenomenon in long distance telephonic communication. Long-delayed and noticeable echo may create significant or even unbearable disturbance in the telephone conversation. Echo cancellation is a technique to reduce the echo in the telephone network to an acceptable level. International Telecommunication Union; Telecommunication Standardization Sector (ITU-T) standardizes the requirements for echo cancellers by establishing a set of recommendations, namely, G.165—Echo Cancellers and G.168—Digital Network Echo Cancellers. Some digital echo cancellation techniques, such as normalized Least Mean Square (LMS) adaptive echo cancellation, have been developed in compliance with the ITU-T standards. The goal of an echo canceller implementation is to reduce the per

Echoes and Echo Cancellation

voice channel processing time while achieving satisfactory echo cancellation. Better echo cancellation algorithms, faster processors, and better implementation techniques may all contribute to this goal. Motorola AltiVec Technology, a high performance PowerPC implementation with parallel processors in a single-instruction-multiple-data (SIMD) architecture to vectorize data, can provide a new way to process more voice channels in a shorter period of time for echo cancellation.

This document describes the various aspects of echo cancellation, including its fundamentals, algorithms, requirements and designs with the Motorola AltiVec technology:

- Part 2 outlines the basics of echo cancellation.
- Part 3 details the echo cancellation algorithms.
- Part 4 summarizes the echo cancellation requirements made by ITU in two recommendations, G.165 and G.168.
- Part 5 describes a general implementation design for echo cancellers in AltiVec technology.
- Part 6 analyzes the overall performance of the echo canceller.

Part II Echoes and Echo Cancellation

This part contains two sections, echo sources and echo cancellation.

2.1 Echo Sources

The basic telephone network consists of two types of wire segments: four-wire central network and two-wire local network as shown in Figure 1. The two-wire network includes a subscriber loop and some portion of the local network. The choice of two wires in the subscriber loop, rather than four wires as in the central network, is mainly economic. The four-wire central network separates the two directions of signal transmission, using one pair of wires for each transmission direction. The two-wire local network, on the other hand, carries signal transmission in both directions in the same pair of wires. A converting device, called a hybrid, is needed at the junction of the two-wire to four-wire segments. The impedance mismatch of the converting device is one source of echo.

As shown in Figure 1, when a far-end user talks, the speech signal travels through the near-end echo path, and a portion of the signal is reflected back to the far-end listener, due to the impedance mismatch in the hybrid. This type of the echo is called electric echo or circuit echo. G.165 standardizes the design requirements for echo cancellers operating on such electric echoes.

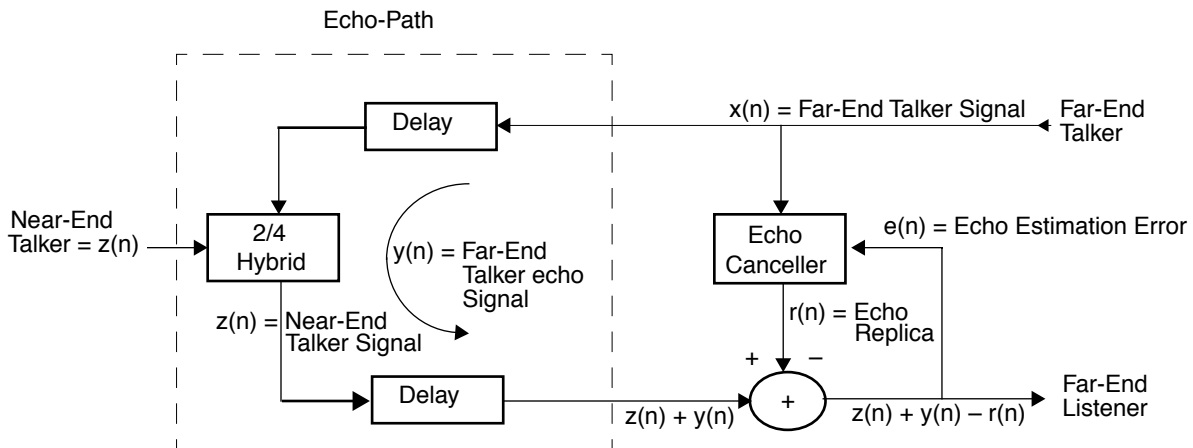


Figure 1. Four-Wire Central Network and Two-Wire Local Network

Acoustic echo is another type of echo. The source of the acoustic echo is acoustic environments, in which the echo path is the acoustic path from earphone to microphone. ITU Recommendation G.168 standardizes the design requirements for echo cancellers to minimize both electric echoes and acoustic echoes in a digital telecommunication network.

Figure 1 shows various signals involved in an echo cancellation process. The far-end talker signal $x(n)$ is transmitted through the echo path. As the signal travels, the strength of the signal gets weaker or attenuates. The echo signal $y(n)$ is a portion of the delayed $x(n)$, and $y(n)$ is reflected back to the far-end listener by the 2/4 hybrid. Without an echo canceller, the far-end listener receives the mixture of the desired near-end talker signal $z(n)$ and the undesired far-end talker echo $y(n)$. The effect of the echo on the quality of communication depends on the delay around the transmission network. For short delays (that is, < 20 milliseconds), the echo $y(n)$ represents an insignificant impairment if the signal attenuation is relatively large (that is, 6 dB or more). The short delays make the echoes indistinguishable from the normal side-tone in the telephone, and sufficient echo attenuation prevents a significant amount of the echo energy from causing further echo loops. For long delays (that is, 250-500 milliseconds, as in satellite communication), the echo may create a significant disturbance to the talker and may even make it very difficult to carry on a normal conversation.

2.2 Echo Cancellation

The principle of echo cancellation is to use the far-end talker signal $x(n)$ as a reference signal to generate an echo replica $r(n)$. This replica is subtracted from the signal $z(n) + y(n)$ to yield the transmitted near-end talker signal, $z(n) + y(n) - r(n)$. In theory, total echo cancellation can be achieved if the error signal $y(n) = r(n)$. The practical problems are that the echo of the far-end talker signal is mixed with the near-end talker signal, and the actual transfer function of the echo path is unknown. Therefore, estimating a proper $r(n)$, is essential for echo cancellation. A common technique to generate $r(n)$ is the normalized Least Mean Square (LSM) adaptive filter.

Part III Echo Cancellation Algorithm

This section contains two sections:

- Basic algorithm of the normalized LMS adaptive echo canceller
- Detailed description of echo cancellation process

3.1 Basic Algorithm of the Normalized LMS Adaptive Echo Canceller

The echo canceller works on the assumption that the echo in the communication system can be estimated in an adaptive process by an adaptive filter based on the reference talker signal. The deviation of the estimated echo from the actual echo is calculated and used to update the filter coefficients for more accurate echo estimation in the next iteration. The algorithm works on the following three equations:

1. Echo replica estimation $r(n)$

Let $h_i(n)$ = the adaptive filter (FIR) coefficients at time n (where $0 \leq i < L$, L is the filter length), $r(n)$ = the estimated echo replica at time n , and $x(n)$ = the sample of the far-end talker signal at time n , then the estimated echo replica is as follows:

$$r(n) = \sum_{i=0}^{L-1} h_i(n)x(n-i) \quad (1)$$

2. Echo estimation error $e(n)$

Let $y(n)$ = the electric echo received at the point before the summation with the echo replica (see Figure 1), the echo estimation error at time n is given as follows:

$$e(n) = y(n) - r(n) \quad (2)$$

3. Filter coefficient adaptation, $h_i(n+1)$

The echo estimation error $e(n)$ is used to adjust the filter coefficients for later echo estimation. The set of filter coefficients used in the next echo estimation are calculated by the following:

$$h_i(n+1) = h_i(n) + \mu e(n)x(n-i) \quad (3)$$

where μ is the adaptation gain given by the following:

$$\mu = \frac{\alpha}{r_x(0)_{\max} + c} \quad (\alpha: \text{experimental constant}) \quad (3.1)$$

Equation (3.1) shows that each coefficient is adjusted by adding a fraction of the product of the error term $e(n)$ and reference signal $x(n-i)$ to the previous $h_i(n)$. The fraction is controlled by the adaptation gain, that is the combination of α (the adaptation constant), $r_x(0)_{\max}$ (the estimated maximum average power of the reference signal $x(n)$), and c (the limiting constant). The adaptation gain μ is chosen to be as large as possible to achieve the rapid filter convergence while retaining the filter stability.

Besides the three equations described above, the actual implementation of the echo cancellers needs to handle some other tasks. The next sub-section provides a complete description of the step-by-step process for implementing the echo canceller with the parameters used in the actual implementation.

3.2 Detailed Description of Echo Cancellation Process

The procedure is extracted from C reference code using a floating-point implementation.

The filter length L of the adaptive echo canceller is 512. This filter is able to generate an echo replica with maximum transmission time delay of the echo path up to 64 msec. L can also be other values that corresponding to different echo path delays for various applications. For 8 kHz sampling rate, $L = (\text{echo path delay in msec}) * 8$. The filter length L is the major factor in the total processing time.

NOTE:

A shorter filter length may be chosen if the maximum time delay of the echo path for a specified communication system is known in advance. In fact, one echo canceller is installed at each side of the echo path. The echo canceller shown in Figure 1 is to operate on the far-end talker echo. By the same principle, there is another echo canceller installed at the far-end to operate on the near-end talker echo. As a result of the arrangement, the echo cancellers are implemented close to the echo sources to minimize the echo path delay.

The following steps are needed for echo cancellation:

1. Take a frame of 80 samples of the echo signal, denoted as $s(0)$, $s(1)$, ... $s(79)$. The sample rate is 8 kHz, so the sample duration or frame length is 10 msec. Then apply a high-pass filter to the frame to remove the non-voice low frequency noises. The outputs of the filter, $y(0)$ to $y(79)$, are given by the following:

For $n = 0$ to 79

$$y(n) = b(0)s(n) + \sum_{j=4}^1 (b(j)s(n-j) - a(j)y(n-j)) \quad (4)$$

where, $b(j)$ and $a(j)$ high-pass filter coefficients are:

$$b(0) \dots b(4) = (0.898, -3.590, 5.384, -3.590, 0.898)$$

$$a(1) \dots a(4) = (-3.782, 5.374, -3.397, 0.806)$$

Steps 2 to 15 are for every sample within the frame of 80 samples:

2. Load a speech sample (the far-end talker signal $x(n)$) into the last element of a buffer $xref[512]$, (referred as the reference signal for echo cancellation) and push the rest of the samples up by 1 location (assume the buffer contains 512 previous samples).

for $i = 511$ to 1

$xref[i] = xref[i-1]$

then take a new y

for $n = 0$ to 79

$xref[0] = x[n]$

Echo Cancellation Algorithm

- For every 32 samples, calculate the average energy of the echo signals $P_{\text{echo}}(n)$. Note that the filtered echo signals have been stored in y :

$$P_{\text{echo}}(n) = \frac{1}{32} \sum_{i=0}^{31} (y(n-i))^2$$

This average energy will be used later for double-talk detection.

- For every 32 samples, calculate the average energy of the reference signals:

$$P_{\text{ref}}(n) = \frac{1}{32} \sum_{i=0}^{31} (x_{\text{ref}}(n-i))^2$$

$P_{\text{ref}}(n)$ and its 15 previous values are stored into a reference energy buffer, x_{buf} .

for $i = 15$ to 1

$x_{\text{buf}}(i) = x_{\text{buf}}(i-1)$

$x_{\text{buf}}(0) = P_{\text{ref}}(n)$

- Find the maximum value among the current and past 15 P_{ref} s (this is done only when a new P_{ref} is calculated, that is, every 32 samples):

for $i = 0$ to 15

$r_x(0)_{\text{max}} = \max(x_{\text{buf}}(0), \dots, x_{\text{buf}}(15))$

$r_x(0)_{\text{max}}$ will be used in the filter coefficient adaptation as shown in equation 3.1.

- Check for impulsive reference signal for every sample $x(n)$:

if $(x(n)^2 / 16 > r_x(0)_{\text{max}})$

then $r_x(0)_{\text{max}} = x(n)^2$

- Calculate filter adaptation factor whenever $r_x(0)_{\text{max}}$ is changed:

$$\frac{1}{r_x(0)_{\text{max}} + c} \quad \text{where } c = -36\text{dB}$$

- Detect the existence of double talk:

if $P_{\text{echo}}(n) > -48\text{dB} \ \&\& \ P_{\text{echo}}(n) > (0.5 * r_x(0)_{\text{max}})$

Increment a fixed double-talk counter by 1 where 0.5 is considered as the fixed double-talk threshold. The fixed double-talk counter is one of two indicators of possible double-talk.

if $P_{\text{echo}}(n) > -48 \text{ dB} \ \&\& \ P_{\text{echo}}(n) > dt_thsh * r_x(0)_{\text{max}}$

Increment an adaptive double-talk counter by 1 where dt_thsh is the adaptive double-talk threshold, that is adjusted every 160 samples in Step 18. The adaptive double-talk counter is the other indicator for possible double-talk.

- In case of double talk, freeze the filter coefficient adaptation and use the previous (auxiliary) filter coefficients. The set of auxiliary coefficients is stored every 160 samples in Step 17. This will prevent the divergence of the filter coefficient adaptation when double talk exists.
- Calculate the echo replica

$$\text{echo replica } r(n) = \sum_{i=0}^{L-1} h_i(n) x_{\text{ref}}(n-i)$$

11. Calculate the error of the echo replica

$$\text{estimation error } e(n) = y(n) - r(n)$$

12. Update the filter coefficient h_i . This is done only when the following two conditions are met:

- No double-talk—That ensures that the adaptation process will be stopped when double-talk is detected
- $r_x(0)_{\max} > -45$ dB—That ensures that the adaptation is done only when the reference signal is strong enough, that is, the far-end talker is actively talking.

for $i = 0$ to 511

$$h_i(n+1) = h_i(n) + \mu e(n) x_{\text{ref}}(n-i)$$

where

$$\mu = \frac{1}{L(r_x(0)_{\max} + c)}$$

13. Do the coefficient leakage.

14. Calculate the residual energy ($re(n-1)$ = the previous estimated residual energy)

$$re(n) = \beta * re(n-1) + (1 - \beta)(e(n))^2$$

where β is a leaky integrator constant ($\beta = 0.96$).

15. Nonlinear processing

The nonlinear processing is carried out only when the nonlinear processor is enabled and double-talk is not detected. If $re(n) < rx0_{\max} * (-18$ dB) and the comfort noise is enabled, add the comfort noise to the output of the echo canceller output in such a way that

$$e(n) = k * e(n) + (1 - k) * \text{comfort_noise}$$

where k is a constant and the comfort noise is random noise with the maximum level of -72 dB generated by a comfort noise generator.

Steps 16 to 18 are only done once every 160 samples (2 frames).

16. Ensure double-talk detection.

The occurrences of possible double-talk have been accumulated in Step 8 of the process. This step checks for apparent false double-talk detection due to artificially low adaptive double-talk threshold.

If the adaptive double-talk counter > a certain count

while the fixed double-talk counter < 1

no double-talk is detected

reset the adaptive double-talk threshold to a fixed threshold

reset both adaptive and fixed double-talk counters.

17. Back up the filter coefficients in an auxiliary buffer. The coefficients are going to be used when double-talk is detected.

G.165 and G.168 Recommendations

18. Adjust the adaptive double-talk detection threshold.

The threshold needs to be adjusted when a predetermined portion of the filter coefficients are changed and double-talk is not detected.

$$dt_thsh = 0.97 * dt_thsh + 0.03 * 2.238 * \sum_{i=0}^{L-1} (h_i(n))^2$$

where $2.238 = 3.5$ dB.

Part IV G.165 and G.168 Recommendations

The following three sub-sections are the summary of ITU-T G.165 and G.168 related to the scope of this project.

4.1 Definitions

Figure 2 shows a functional diagram of an echo canceller along with some key parameters defined as follows:

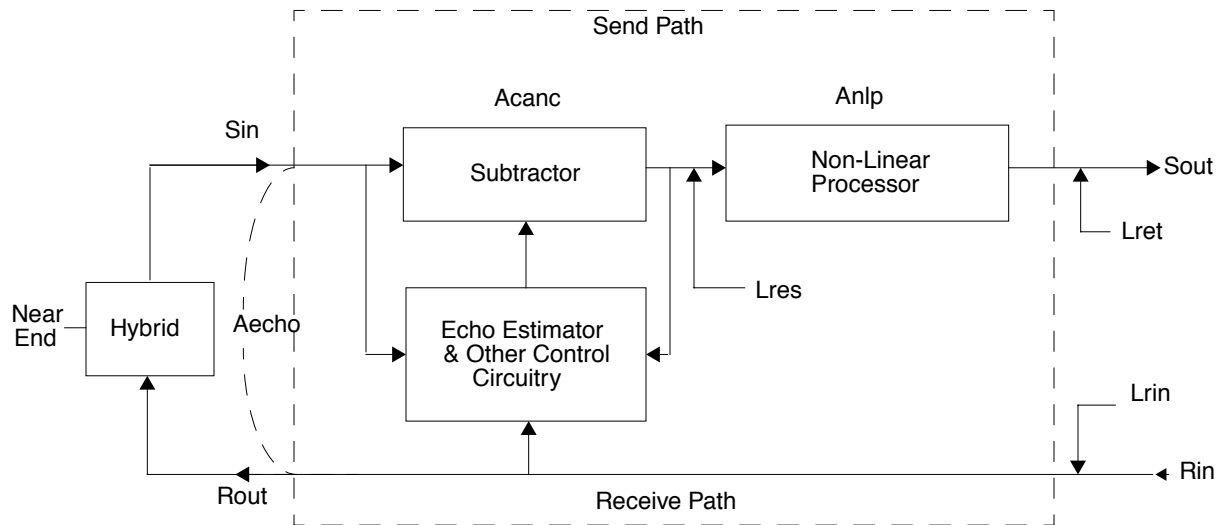


Figure 2. Echo Canceller Definitions

- R_{in} : Receive-in port.
- R_{out} : Receive-out port.
- S_{in} : Send-in port.
- S_{out} : Send-out port.
- Echo path: The transmission path between R_{out} and S_{in} of an echo canceller.
- Near-end: The side of an echo canceller that contains the echo path on which the echo canceller is intended to operate.
- Far-end: The side of an echo canceller, that does not contain the echo path on which the echo canceller is intended to operate.

- Echo loss (A_{echo}): The attenuation of a signal from R_{out} to S_{in} .
- Echo cancellation (A_{canc}): The attenuation of the echo signal as it passes through the send path of an echo canceller.
- Send-in signal level (L_{rin}): The level of the far-end talker signal at S_{in} .
- Residual echo level (L_{res}): The level of the echo signal, that remains at S_{out} after imperfect echo cancellation. $L_{\text{res}} = L_{\text{rin}} - A_{\text{echo}} - A_{\text{canc}}$.
- Nonlinear processor (NLP): A device having a defined suppression threshold level. The device will suppress all signals to some minimum value when the signals are detected to be below the threshold.
- Nonlinear processing loss (A_{nlp}): Additional attenuation of residual echo level by a nonlinear process placed in the send path of an echo canceller.
- Combined loss (A_{com}): The sum of echo loss, cancellation loss and nonlinear processing loss, that is, $A_{\text{com}} = A_{\text{echo}} + A_{\text{canc}} + A_{\text{nlp}}$.
- Returned echo level (L_{ret}): The level of the echo signal at S_{out} . This signal will be returned to the far-end talker.
 - $L_{\text{ret}} = L_{\text{rin}} - (A_{\text{echo}} + A_{\text{canc}} + A_{\text{nlp}})$, if nonlinear processing is included.
 - $L_{\text{ret}} = L_{\text{res}}$, if nonlinear processing is excluded.
- Echo pure delay (t_p): The delay from R_{out} to S_{in} due to the delays in the near-end echo path transmission facilities.
- Echo path delay (t_d): The sum of pure delay and dispersion time, t_d is the time required to accommodate the band-limiting, multiple reflection, and hybrid transit effect.
- Convergence: The process of developing a model of the echo path that will be used in the echo estimator to produce the estimate of the circuit echo.
- Convergence time: The interval between the instant a defined test signal is applied to R_{in} of an echo canceller and the instant the returned echo level at S_{out} reaches a defined level.
- Comfort noise: Insertion of pseudo-random noise during silent intervals. Comfort noise prevents the annoyance of intervals of speech with background noise followed by intervals of silence.

4.2 ITU-T Recommendation G.165—Echo Cancellers

The ITU-T Recommendation G.165 applies to echo canceller designs using either digital or analog techniques and is intended for use in an international circuit. Echo cancellers designed to G.165 will be compatible with each other. G.165 defines performance requirements for an echo canceller but leaves the design details (such as the implementation algorithms) to the designers. The recommendation also specifies a set of tests for verification of the echo cancel design.

4.2.1 Characteristics of Echo Cancellers

This section includes the purpose of echo cancellers, the minimum echo loss, and the minimum order of echo cancellers.

4.2.1.1 Purpose of Echo Cancellers

Echo cancellers are designed to have cancellation take place only in the send path due to signals present in the receive path (see Figure 2), that is, the echo canceller is designed to cancel the far-end talker echo. There is a separate echo canceller located near the far-end echo path to cancel the near-end talker echo.

4.2.1.2 Minimum Echo Loss

One of the requirements for communication system hardware design is that the echo loss (A_{echo}) from R_{out} to S_{in} is 6 dB or greater. Echo cancellers are designed to perform properly for the echo loss of 6 dB or greater.

4.2.1.3 Minimum Order of Echo Cancellers

An echo canceller must have sufficient storage capacity (the order of the echo canceller) for the required number of signal samples in order to produce a proper replica of the echo path impulse response. In general, too small an order of the echo canceller will not be able to produce a replica that is adequate for all echo paths. On the other hand, too large an order of the echo canceller will create undesirable additional noise. For a sampling rate of 8 kHz, the order of the echo cancellers is 8 times the echo path delay ($8 \cdot t_d$).

4.2.2 Echo Canceller Requirements

1. Rapid convergence rate.
2. Low returned echo level during single talk
3. Low divergence during double talk

4.2.3 Echo Canceller Requirement Tests

The performance of an echo canceller is tested in the following 11 tests. This document only summarizes the general requirement for each of the tests without listing the detailed testing conditions and numerical specifications. Refer to G.165 recommendation for detailed requirements.

1. Test 1—Steady state residual and returned echo level test
The test ensures that the steady state cancellation (A_{canc}) is sufficient to produce a residual echo level below a certain threshold. The level should be sufficiently low to permit the use of nonlinear processing without undue reliance on it.
2. Test 2—Convergence test
This test ensures that the echo canceller converges rapidly (within a half second) for all combinations of input signal levels and echo paths and that the returned echo level is sufficiently low.
3. Test 3—Double-talk detection test
The double talk detector should be able to correctly detect double talk, that occurs when the near-end talker is also talking. The double-talk detection is tested in two ways. First, the false detection of double talk should not occur. Second, the double talk detector should be sufficiently sensitive and be able to operate fast enough to detect double talk when it does occur, and to prevent large divergence during the period of double-talking.
4. Test 4—Leak rate test
If the far-end talker signal is removed from R_{in} after the echo canceller reaches the fully converged state, the impulse response of the echo canceller will gradually converge to zero. This test ensures that this does not happen too fast. The test requires that two minutes after the removal of the R_{in} signal, the residual echo level (L_{res}) should not increase more than 10 dB over the steady state L_{res} .
There are an additional 7 tests listed in the G.165 recommendation. These tests are either provisional, optional, not defined (under study), or beyond the scope of the project. The names of the tests are listed for reference only.
5. Test 5—Infinite return loss convergence test (provisional)

6. Test 6—Nondivergence on narrow-band signals (optional)
7. Test 7—Nonconvergence of echo cancellers on mono or bifrequency signals transmitted in a handshaking protocol (optional)
8. Test 8—Overload test for Type A and Type D cancellers
9. Test 9—Comfort noise test (provisional)
10. Test 10—Facsimile test (under study)
11. Test 11—Tandem echo canceller test (under study)

In addition there are some requirements for echo canceller tone (2100 Hz) disabler and non-linear processor.

4.3 ITU-T Recommendation G.168—Digital Network Echo Cancellers

The ITU-T Recommendation G.168 applies to echo canceller design using digital techniques and is intended for use in circuits with relatively long delays. The recommendation ensures that echo canceller performance is adequate under wider network conditions that are specified in G.165, (such as performance on voice, FAX, residual acoustic echo signal)s and in mobile networks. Echo cancellers designed to G.168 will be compatible with each other, with echo cancellers designed in accordance with G.165, and with echo suppressors designed in accordance with the ITU-T Recommendation G.164 echo suppressors recommendation. Like G.165, G.168 defines performance and test requirements for an echo canceller but leaves the design details to the designers.

4.3.1 Echo Canceller Requirements

1. Rapid convergence rate
2. Low returned echo level during single talk
3. Low divergence during double talk
4. Assured double talk detection
5. Proper operation during facsimile and low speed (<9.6 Kbit/sec) voiceband data transmissions

4.3.2 Echo Canceller Requirement Tests

The performance of an echo canceller is tested in 14 tests, plus echo canceller tone disabler and non-linear processor. Tests 1 – 11, echo canceller tone disable and non-linear processor testing requirements are similar to the ones in G.165 as described above. Tests 12 & 13 are under study, and Test 14 is for C-Series low-speed data modems. Therefore, the test details are not described in this document.

Part V AltiVec Implementation Design

This section includes details about the AltiVec implementation design.

5.1 Requirements and Priorities

The implementation is based on the following facts or simulation conditions:

1. Speech signals are sampled at 8 kHz with 16-bit precision.
2. Each frame contains 80 samples (10 msec sample period).
3. The output of the echo canceller is 16 bit.

Altivec Implementation Design

The 18 steps of the echo cancellation process described in Section 3.2 need to be implemented, but some of them are computationally intensive and characterized by typical multiply-accumulate (MAC) operations. These operations have great impact on the overall execution time of the echo canceller. Therefore, it is necessary to identify those “key” functional operations. Table 1 lists 10 such computationally intensive functions classified by functionality. Each functional module is not necessarily a separate function in the implementation. The steps in Section 3.2, “Detailed Description of Echo Cancellation Process” not listed here are those that take only a small fraction of execution time. These non-key functions, however, may contribute significantly to the code-size of an echo canceller.

Table 1 summarizes the following characteristics of each functional module:

1. Execution frequencies per frame (Column 2)
2. Number of multiply-accumulates per execution (Column 3)
3. Number of multiply-accumulates per frame (the product of the first two items) (Column 4)
4. Estimated execution time percentage with respect to the overall execution time of the echo canceller (Column 5)

Table 1. Computationally Intensive Functions

Function	Execution Frequency/ Frame (2)	Macs/Execution (3)	# of Macs/ Frame (4)	% in Total Macs (5)
High-pass filter	80	9	720	<1
Estimate echo power	3	32	96	<1
Estimate reference power	3	32	96	<1
Check impulsive signal	80	2	160	<1
Estimate echo replica	80	512	40960	48.8
Adapt filter coefficients	80	512	40960	48.8
Estimate residual energy	80	3	240	<1
Nonlinear processing	80	3	240	<1
Adapt double talk threshold	0.5	515	258	<1
Generate comfort noise	80	3	240	<1
Total	—	—	83970	—

It is clear that two functions, echo replica estimation and filter coefficient adaptation, are the most important ones, including more than 97% of the total MACs.

5.2 Vector Alignment Adjustment

The 512 filter coefficients (h_0 to h_{511}) are stored in 64 vectors $H[0]$ to $H[63]$ with alignment. All the h and x values are in Q15 format.

$H[0]$	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7
--------	-------	-------	-------	-------	-------	-------	-------	-------

...

The corresponding 512 reference samples (x_n to x_{n-511}) to be processed per sample echo cancellation are

$H[63]$	h_{504} to h_{511}
---------	------------------------

stored in another 64 vectors $X[0]$ to $X[63]$ with alignment.

$X[0]$	x_n	x_{n-1}	x_{n-2}	x_{n-3}	x_{n-4}	x_{n-5}	x_{n-6}	x_{n-7}
--------	-------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

...

$X[63]$	x_{n-504} to x_{n-511}
---------	----------------------------

For each new speech sample, all 511 previous samples have to be shifted rightward by 1 (time delay of 1). Therefore, all the elements in the 64 vectors need to be shifted rightward by 1 with the most right element of $X[63]$ to be dropped, and a new speech sample to occupy the first element in $X[0]$. This is done by executing the `vec_perm` instruction 64 times. The vector alignment is done here, rather than later, because the X vectors are going to be used twice in the later process.

5.3 High-Pass Filter

The high-pass filter operation needs 5 echo samples (stored in a S vector), 4 previous filtered y results (stored in a Y vector), 11 coefficients (named as b and a in Section 3.2). The coefficient of a_0 is never used in this operation, so it is omitted. The b and a coefficients are stored in two separate vectors, B and A , with the coefficients being negated.

B	b_4	b_3	b_2	b_1	b_0	0	0	0
-----	-------	-------	-------	-------	-------	---	---	---

A	$-a_4$	$-a_3$	$-a_2$	$-a_1$	0	0	0	0
-----	--------	--------	--------	--------	---	---	---	---

Use `lvsl` and `perm` instructions to fetch s and y elements into S and Y vectors

S	s_{n-4}	s_{n-3}	s_{n-2}	s_{n-1}	s_n	0	0	0
-----	-----------	-----------	-----------	-----------	-------	---	---	---

Y	y_{n-4}	y_{n-3}	y_{n-2}	y_{n-1}	0	0	0	0
-----	-----------	-----------	-----------	-----------	---	---	---	---

Use `vec_msums` to get $X*B$ and $Y*A$, use `vec_adds` followed by `vec_sums` to get the total multiply-sum. The process is repeated 80 times for all 80 echo-samples.

5.4 Average Echo Power Calculation

The 32 echo samples are stored in 4 vectors E[0] to E[3]. Let P_{echo} = the accumulated-sum of the echo power, then the result of each `vec_msums` instruction will yield:

E [0]	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7
*								
E [0]	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7
=								
P_{echo}	$e_0 * e_0 + e_1 * e_1$		$e_2 * e_2 + e_3 * e_3$		$e_4 * e_4 + e_5 * e_5$		$e_6 * e_6 + e_7 * e_7$	

Do the same calculation for E[1] to E[3]. Two P_{echo} s will be used in the implementation to interleave `vec_msums`, and that, in combination with load instructions of E vectors, will utilize the 3 clock latencies for each `vec_msums` instruction. The final accumulated-sum is obtained by adding the two P_{echo} s together (`vec_adds`) and adding across (`vec_sums`). The average echo power is the final accumulated-sum divided by 32 ($>>5$). The final result is in Q30 format (Q15*Q15).

5.5 Average Reference Power Calculation

The implementation of the calculation of the average reference power is exactly the same as the calculation of the average echo power (see Section 5.4), except the signal to be operated on is the reference signal x .

5.6 Maximum among Sixteen Average Reference Powers

The maximum value of the 16 most recent average reference powers calculated in Section 5.5 will be needed for checking impulsive reference signals (see Step 6 in Section 3.2). The values of the 16 average reference powers are stored in 4 vectors, P[0]-P[3]. The maximum is the result of applying `vec_max` and `vec_sld` in the following steps:

1. Do `vec_max` on P[0] and P[1] to find 4 candidates of the maximum.

P[0]	p_0	p_1	p_2	p_3
P[1]	p_4	p_5	p_6	p_7

`vec_max` will produce:

M0	$\max(p_0, p_4)$	$\max(p_1, p_5)$	$\max(p_2, p_6)$	$\max(p_3, p_7)$
----	------------------	------------------	------------------	------------------

2. Do `vec_max` on P[2] and P[3] and store the results to M1.
3. Do `vec_max` on M0 and M1 and store the results to M0.
4. Left shift M0 by 8 bytes and save the result into M1.
5. Do `vec_max` among M0 and M1 and save the results to M0.
6. Left shift M0 by 4 bytes and save the result into M1.
7. Do `vec_max` among M0 and M1 and save the results to M0.

The maximum value among all the 16 candidates now is the 1st element of M0.

5.7 Echo Replica Estimates

The 512 filter coefficients and 512 reference samples are stored in 64 H vectors and 64 X vectors, respectively, as described in Section 5.1. Let R store the estimated echo replica, set R = 0 to start.

H[0]	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7
------	-------	-------	-------	-------	-------	-------	-------	-------

*

X[0]	x_n	x_{n-1}	x_{n-2}	x_{n-3}	x_{n-4}	x_{n-5}	x_{n-6}	x_{n-7}
------	-------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

+

R	r_0	r_1	r_2	r_3
---	-------	-------	-------	-------

=

R	$h_0 * x_n + h_1 * x_{n-1} + r_0$	$h_2 * x_{n-2} + h_3 * x_{n-3} + r_1$	$h_4 * x_{n-4} + h_5 * x_{n-5} + r_2$	$h_6 * x_{n-6} + h_7 * x_{n-7} + r_3$
---	-----------------------------------	---------------------------------------	---------------------------------------	---------------------------------------

The above process repeats for the rest of H and X vectors.

This function is one of the two most time consuming functions in an echo canceller. For each echo replica estimation, the following instructions are needed:

- 64 loads for H and 64 loads for X
- 64 `vec_msums`
- 1 `vec_adds` and 1 `vec_sums` at the very end to get the final accumulated-sum
- 1 `vec_sra` for scaling the final result (from Q30 to Q15)

Two R vectors are used in the calculation to interleave `vec_msums` operation in order to utilize the 3 clock latencies of each `vec_msums`.

5.8 Filter Coefficient Adaptations

This function modifies all elements of the 64 H vectors based on the echo estimation error $e(n)$, the 64 X vectors (the reference signals) and μ

$$(\mu = \frac{1}{(r_x(0)_{\max} + c)}).$$

At a given time n , e and μ are fixed, so $e*\mu$ can be calculated once in advance and then splatted by `vec_splat` into a constant vector C . Then use `vec_mradds` to get the updated h values.

C	c	c	c	c	c	c	c	c
	*	*	*	*	*	*	*	*
X[0]	x_n	x_{n-1}	x_{n-2}	x_{n-3}	x_{n-4}	x_{n-5}	x_{n-6}	x_{n-7}
	+	+	+	+	+	+	+	+
H[0] at n	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7
	=	=	=	=	=	=	=	=
H[0] at n+1	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7

This is done for all the h coefficients. The operations may be interleaved because the coefficient updates are independent of each other.

This operation is the other most time consuming function modules. For each speech sample, the following instructions are needed:

- 64 loads for H and 64 loads for X vectors
- 64 `Vec_adds`
- 64 stores for H vectors

5.9 Residual Energy Estimates

This operation is calculated once per sample. Each term in the calculation is in Q30 (Q15*Q15) format, and the result needs to be shifted right by 15 bit to convert it back to Q15.

5.10 Non-Linear Processing

This operation is calculated once per sample and is done in scalar code.

5.11 Double-Talk Threshold Adaptation

The accumulated-sum of $\sum_{i=0}^{L-1} (h_i(n))^2$ can be calculated in the similar way as described in Section 5.4.

Part VI Performance Analysis

The performance of the AltiVec echo canceller implementation is measured in MIPS by Maxim on a Macintosh. The MIPS is for single channel with cache warm up. R_{in} signal (reference) for the echo canceller is from a file containing 2800 frame of speech samples (80 samples per frame). S_{in} signal (the mixture of the echo and the near-end talker signal) is from a separate file of the same length. The MIPS numbers listed in Table 2 are the average within a frame of 80 samples. The average is calculated by:

$$\text{MIPS} = \text{Total number of clocks for processing the frame} / 10000$$

The maximum and minimum MIPS numbers are obtained from two frames, that are believed to be the most and the least computation-intensive ones. The two frames are chosen from a pilot run on 400 frames of the speaker samples. The run showed a bimodal behavior of the echo canceller, mostly due to the filter coefficient update process. When double talk is absent, the filter coefficients need to be updated after each S_{out} is generated. On the other hand, when double talk is detected, the adaption freezes and the filter coefficients are not updated. The maximum MIPS represents, or is close to the worst case scenario; and the minimum MIPS is for comparison and analysis purpose only.


Table 2. MIPS Numbers from Run on Speaker Samples

Echo Delay (msec)	Maximum in MIPS (Most Computational Intensive)	Minimum in MIPS (Least Computational Intensive)
64	6.41	3.40
32	4.52	2.22
16	3.42	1.56
8	2.51	1.04

DigitalDNA and Mfax are trademarks of Motorola, Inc.

The PowerPC name, the PowerPC logotype, and PowerPC 603e are trademarks of International Business Machines Corporation used by Motorola under license from International Business Machines Corporation.

Information in this document is provided solely to enable system and software implementers to use PowerPC microprocessors. There are no express or implied copyright licenses granted hereunder to design or fabricate PowerPC integrated circuits or integrated circuits based on the information in this document.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and  are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:

USA/EUROPE: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1-303-675-2140 or 1-800-441-2447
1-303-675-2140 or 1-800-441-2447

JAPAN: Motorola Japan Ltd.; SPS, Technical Information Center, 3-20-1, Minami-Azabu, Minato-ku, Tokyo 106-8573 Japan. 81-3-3440-3569

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong. 852-26668334

Customer Focus Center: 1-800-521-6274

Mfax™: RMFAX0@email.sps.mot.com

Motorola Fax Back System

- TOUCHTONE 1-602-244-6609

- US & Canada ONLY <http://sps.motorola.com/mfax/>

HOME PAGE: <http://motorola.com/semiconductors>

Document Comments: FAX (512) 895-2638, Attn: RISC Applications Engineering

World Wide Web Addresses: <http://www.motorola.com/PowerPC>
<http://www.motorola.com/netcomm>
<http://www.motorola.com/ColdFire>



MOTOROLA

AN1842/D