AN2397
Rev. 1, 12/2004

ITU-T G.729I Recommendation on the StarCore™ SC140/SC1400 Cores

By Corneliu Margina, Emilian Medve, and Bogdan Costinescu

This application note describes the process of implementing the ITU-T G.729I Recommendation on Freescale DSPs based on the StarCore™ SC140/SC1400 DSP cores. The implementation consists of two phases: porting the code onto the SC140 core and optimizing the most time/size-consuming functions to achieve performance requirements. The ITU-T G.729I Recommendation [7] presents a reference fixed-point implementation for integrating the ITU-T G.729 Recommendation with Annexes D, E, and B (silence compression scheme). The Annex B is used for each of the three integrated speech coding rates, so the G.729I Recommendation is composed of the following:

- G.729 (8 kbps rate) [1]
- G.729B (G.729 + Annex B) [2]
- G.729D (6.4 kbps rate) [3]
- G.729F (G.729D + Annex B) [4]
- G.729E (11.8 kbps rate) [5]
- G.729G (G.729E + Annex B) [6]

Some optimizations in the ITU-T G.729 [8] and ITU-T G.729B [11] implementations are integrated into this project. The Metrowerks® StarCore compiler reduced the effort required in the optimization phase. Therefore, most of the functions are optimized in C and only a few are optimized in assembly language. The bit-exactness tests were performed using all available ITU test vectors for the implemented rates with or without Annex B. The project achieved high performance on the SC140 core. Thirty channels on the 300 MHz SC140 core required 229 KB of memory for all channels.

CONTENTS

© Freescale Semiconductor, Inc., 2002, 2004. All rights reserved.

1 ITU-T G.729I Recommendation

The ITU-T G.729I Recommendation is a Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP) speech codec that runs at 6.4 kbps (Annex D [3]), 8 kpbs (G.729 Recommendation [1]) and 11.8 kbps (Annex E [4]) using the silence compression scheme (Annex B). The Annex B [2] adds the functionality for discontinuous transmission (DTX), voice activity detection (VAD), and comfort noise generator (CNG) at each integrated speech codec rate. The integrated rates have the following functionality:

- *8 kbps* (G.729). The analog voice signal is passed through a 300 Hz–3400 Hz bandpass filter and sampled at 8 kHz to yield digital data that is converted to a 16-bit linear PCM speech signal. The encoder analyzes the speech signal to extract the parameters of the CELP model. These parameters are encoded and transmitted into a bitstream. The decoder uses the received parameters to retrieve the synthesis filter coefficients. Thus, the speech is reconstructed by filtering the excitation codebook. The vocoder operates on 10 ms frames with a 5 ms look-ahead for linear-prediction (LP) analysis. The overall algorithmic delay is 15 ms.
- *6.4 kbps* (G.729D). Annex D represents the lower rate extension to the G.729 Recommendation and handles the overload conditions. It does not provide the same level of quality as the G.729 Recommendation, but it provides significantly higher quality than the G.726 Recommendation at 24 kbps. Annex D is based on the G.729 Recommendation with a few modifications:
  - The ACELP codebook is modified to use two signed pulses in two overlapping tracks of different lengths.
  - The conjugate-structured codebook for the gains is modified to use only 6 bits.
  - Only 4 bits are used for coding the pitch delay in the second subframe.
  - An additional postfiltering technique reduces the effects of the sparser algebraic codebook.
  - The pitch delay parity bit is removed.
- *11.8 kbps* (G.729E): Annex E represents the higher rate extension to the G.729 Recommendation. It accommodates a wide range of input signals, such as speech with background noise and even music. Modifications to this annex are as follows:
  - A backward LP analysis is added for music signals and stationary background noises. The backward/forward decision selects speech (forward mode) or music (backward mode). The backward/forward procedure also reduces the number of switches needed to perform smooth switching between filters. The LP mode and the related information better adapts postfiltering and perceptual weighting to either music or speech. This mode is also used for the error concealment.
  - Two algebraic excitation codebooks are added to extend the bit rate up to 11.8 kbps.

2 ITU-T G.729I Implementation Process

This section describes the process of implementing the ITU-T G.729I Recommendation on the SC140 core. This process consists of the following phases:

- *Porting the code to the SC140 core.* Modifying the data type definitions, adding pragmas and intrinsic functions, and building channel data.
- *Project level optimizations*. Function inlining, data alignment, and channel data modifications.
- Algorithmic changes. Adapting the algorithms to perform efficiently on the SC140 core.
- Function level optimizations. Applying C optimization techniques and optimizing a few functions by writing them in assembly language.

A performance results table is presented for each of these phases. The values listed are for the three rates using Annex B. The values for G.729, G.729D, and G.729E are lower because of the Annex B computations that are not performed. Tests were performed on both the SC140 simulator and the MSC8101ADS board using the Metrowerks CodeWarrior® for StarCore IDE Release 1.5 with available patches.

2.1 Porting ITU-T G.729I to the SC140 Core

The steps in the ITU-T G.729I porting methodology presented in [8], [9], [10], [11], and [12] are as follows:

1. Use the `prototype.h` file to define all data types for the SC140 architecture.
2. Replace the emulated DSP operations with intrinsic functions provided by the Metrowerks SC140 C compiler.
3. Handle the overflow flag with intrinsic functions provided by the Metrowerks SC140 C compiler.
4. Modify the code for multi-channel environments.
5. Split one function per file.

Table 1summarizes the performance of G.729I on the SC140 core. These results are obtained using only C code compiled with the –O3 and –O3 –Og (global) optimization flags, respectively. Individual figures are provided for each rate: 8 kbs, 6.4 kbs, and 11.8 kbs.

Table 1. Performance After ITU-T G.729I is Ported to the SC140 Core

Speed (MCPS)			ROM		RAM		Optimization Level
G.729B	G.729F	G.729G	Program (Bytes)	Tables (Bytes)	Channel (Bytes)	Stack (Bytes)	
24.64	20.37	36.4	68688	7840	5060	3528	–O3
23.29	18.87	33.68	66226	7840	5060	3408	–O3 –Og

2.2 Project-Level Optimizations

The global project optimizations benefit from the G.729 [8] and G.729B [11] optimized implementations, considerably reducing the effort required and improving both performance and program size. Some optimized functions (C and/or assembly) can be directly integrated into the current project, but others must be adapted due to the increased order (30) of the LPC polynomial in G.729E. **Table 2** summarizes the functions integrated from the G.729 [8] and G.729B [11] optimized implementations.

Table 2. Reused Functions from G.729 [8] and G.729B [11] Optimized Implementations

Common	Encoder	Decoder
Calc_exc_rand()	Autocorrcp()	compute_ltp_l()
Copy()	Az_lsp()	Post_Process()
Gauss()	Chebps_10	search_del()
Get_lsp_pol()	Chebps_11	
Inv_sqrt()	Convolve()	
Levinsong()	cor_h()	
Pred_lt_3()	corr_xy2()	
Random()	D4i40_17()	

Table 2. Reused Functions from G.729 [8] and G.729B [11] Optimized Implementations (Continued)

Common	Encoder	Decoder
Residue()	Lag_max()	
Syn_filte()	Lsp_pre_select()	
	Norm_corr()	
	Pitch_ol()	
	Pre_Process()	
	vad()	

The optimizations in this phase apply to the new and changed functions from Annexes D and E:

- Channel data alignment to an 8-byte boundary to benefit from the packed moves of the SC140 core.
- Alignment of the global static tables and local temporary vectors to benefit from the packed moves of the SC140 core.
- Replacement of 32-bit DPF operations with implementations that use the efficient support provided by the Metrowerks C compiler.
- Move of program code and global static tables into custom named sections for link-time flexibility.
- Removal and propagation of constant parameters.

Table 3 shows the performance after the project-level optimization phase. The results are based only on C optimized functions and on assembly and optimized C functions compiled with the –O3 compiler optimization flag.

Table 3. Performance After the Project-Level Optimizations Phase

Speed (MCPS)			ROM		RAM		Implementation
G.729B	G.729F	G.729G	Program (Bytes)	Tables (Bytes)	Channel (Bytes)	Stack (Bytes)	
16.83	11.95	18.03	69248	7864	4976	5032	C only
16.32	11.78	16.78	66888	7864	4976	5016	C and assembly

2.3 Algorithmic Changes

Functions were selected for optimization on the basis of profiler data and experience with similar applications, with a focus on the most time consuming functions. Functions selected for optimization are collectively called the “G1 set.” Algorithmic changes applied on selected functions are:

- Reordering of vectors.
- Packed index computation.
- Sequential non-dependent computations performed in parallel.

We selected the following encoder functions selected for algorithmic changes:

- `cor_h_e()`
- `cor_h_vec()`
- `D2i40_11()`
- `search_ixiy()`

Table 4 presents the performance results after the algorithmic changes. Results are based on optimized C functions and on assembly and optimized C functions compiled with the –O3 compiler optimization flag.

Table 4. Performance Results After Algorithmic Changes

Speed (MCPS)			ROM		RAM		Implementation
G.729B	G.729F	G.729G	Program (Bytes)	Tables (Bytes)	Channel (Bytes)	Stack (Bytes)	
13.33	9.74	14.35	76754	8072	4984	4912	C only
10.2	8.51	11.36	70414	7988	4984	4896	C and assembly

2.4 Function-Level Optimizations

The functions to be optimized are selected from the G1 set on the basis of profile results. Function-level optimizations involve the following optimization techniques [15]: multi-sampling; loop merging, splitting, and unrolling; and split computation.

A few small functions are manually inlined to eliminate the call overhead. Other functions are merged due to similar computation to decrease the code size. A scratch space for reducing the stack size was added as a more efficient way to manage the RAM space. **Table 5** lists the functions optimized only in C.

Table 5. Functions Optimized Only in C

Common	Encoder	Decoder
ener_dB()	ACELP_Codebook()	calc_st_filte()
Lsp_expand_1()	ACELP_Codebook64()	Dec_gaine()
Lsp_expand_2()	ACELP_10i40_35bits()	Dec_gaine_6k()
Lsp_expand_1_2()	ACELP_12i40_44bits()	Decod_ld8c()
	Encoder_ld8c()	filt_mu()
	Gbk_presel()	pst_ltpe()
	Gbk_presel_6k()	scale_st()
	Lag_window()	
	Lsp_select_1()	
	Lsp_select_2()	
	perc_vare()	
	Qua_gain()	
	Qua_gain_6k()	
	update_bwd()	

Table 6 presents the performance results after the function-level C optimizations. Results are obtained using C-only functions compiled with the –O3 optimization flag of the compiler.

Table 6. Performance of C-Optimized Implementation

Speed (MCPS)			ROM		RAM		
G.729B	G.729F	G.729G	Program (Bytes)	Tables (Bytes)	Channel (Bytes)	Stack (Bytes)	Scratch (Bytes)
11.89	8.38	13.25	79856	7936	4984	2640	2512

Table 7 lists the functions optimized in assembly because they are time critical functions from the G1 set. These functions are also optimized in C.

Table 7. Assembly Optimized Functions

Common	Encoder	Decoder
autocorr_hyb_window()	cor_h_e()	
Lag_window_bwd()	cor_h_vec()	
Weight_Az()	cor_h_x_e()	
	search_ixiy()	

Table 8 presents the performance results after the assembly function-level optimizations. Results are based on assembly and C functions compiled with the –O3 optimization flag.

Table 8. Performance of C- and Assembly-Optimized Implementation

Speed (MCPS)			ROM		RAM		
G.729B	G.729F	G.729G	Program (Bytes)	Tables (Bytes)	Channel (Bytes)	Stack (Bytes)	Scratch (Bytes)
8.59	6.98	9.91	70914	7932	4984	2640	2512

3 Performance

Table 9 summarizes the performance results of the G.729I optimized SC140 implementation. Results are based on assembly and C optimized functions compiled with the –O3 optimization flag of the Metrowerks StarCore C compiler. The encoded bitstream is packed in the RTP format specified in the `draft-ietf-avt-profile-new-12.txt` file (a revision to RFC1890): "RTP Profile for Audio and Video Conferences with Minimal Control." Performance results show that the ITU-T G.729I standard achieves high performance on the SC140 core: 30 channels of G.729I on a 300 MHz SC140 core uses 229 KB of memory for all channels. **Figure 1** shows the person-months required. **Figure 1** presents the MCPS figures of the G.729G module, which is the most time-consuming part. The midpoints represent the intermediate values obtained during integration and optimization.

Table 9. Performance Results of the G.729I Optimized Implementation on the SC140 Core

Speed (MCPS)			ROM		RAM			Implementation
G.729B	G.729F	G.729G	Program (Bytes)	Tables (Bytes)	Channel (Bytes)	Stack (Bytes)	Scratch (Bytes)	
11.89	8.38	13.25	79856	7936	4984	2640	2512	C only
8.59	6.98	9.91	70914	7932	4984	2640	2512	C and assembly

The following extra memory is required to run the encoder and decoder modules:

- Encoder (454 bytes):
  - Input speech vector: 160 bytes
  - Analysis parameters vector: 38 bytes
  - Analysis parameters vector using RTP format: 16 bytes
  - Output bitstream vector: 240 bytes
- Decoder (458 bytes)
  - Input bitstream vector: 240 bytes
  - Synthesis parameters vector: 42 bytes
  - Synthesis parameters vector using RTP format: 16 bytes
  - Postfilter output vector: 160 bytes

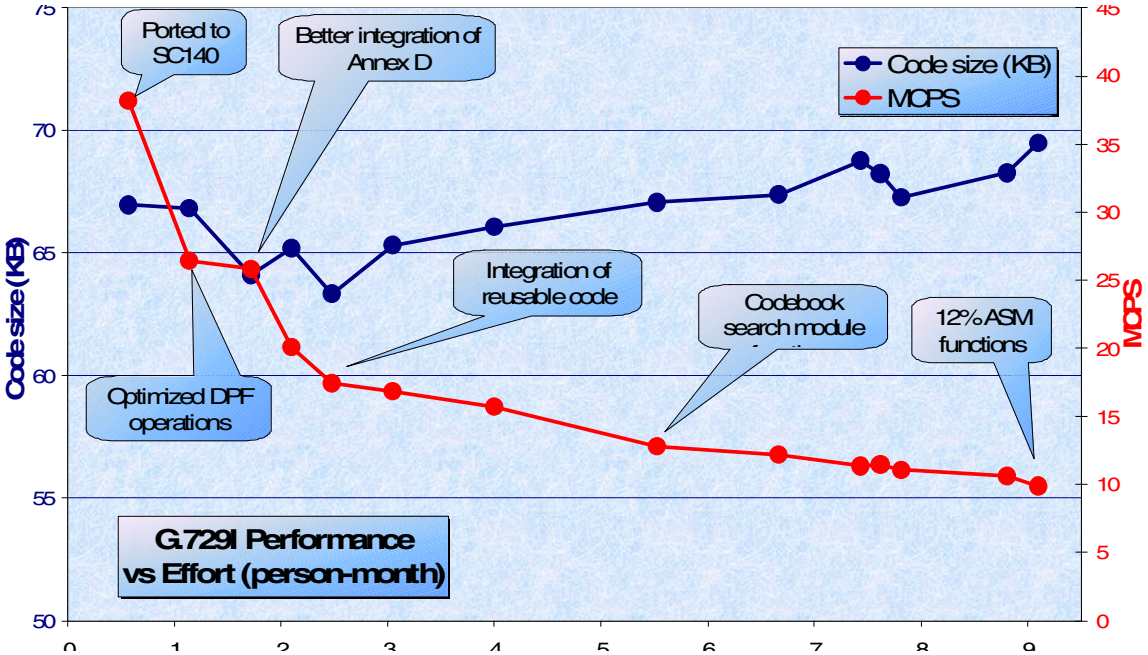

Figure 1. Processing Load Versus Implementation Effort

4 References

[1] ITU-T Recommendation G.729. *Coding of Speech at 8 kbit/s using Conjugate Structure Algebraic Code-Excited Linear Prediction (CS-ACELP)*, March 1996

[2] ITU-T Recommendation G.729 Annex B. *A Silence Compression Scheme for G.729 Optimized for Terminals Conforming to Recommendation V.70*, November 1996.

[3] ITU-T Recommendation G.729 Annex D. *6.4 kbit/s CS-ACELP Speech Coding Algorithm*, Sept. 1998.

[4] ITU-T Recommendation G.729 Annex F. *Reference Implementation of G.729 Annex B DTX functionality for Annex D*, February 2000.

[5] ITU-T Recommendation G.729 Annex E. *11.8 kbit/s CS-ACELP Speech Coding Algorithm*, Sept. 1998.

[6] ITU-T Recommendation G.729 Annex G. *Reference implementation of G.729 Annex B DTX Functionality for Annex E*, February 2000.

[7] ITU-T Recommendation G.729 Annex I. *Reference Fixed-point Implementation for Integrating G.729 CS-ACELP Speech Coding Main Body with Annexes B, D and E*, February 2000.

[8] *ITU-T G.729 Implementation on StarCore SC140*, Freescale Semiconductor (AN2094).

[9] *ITU-T G.729A Implementation on StarCore SC140*, Freescale Semiconductor (AN2151).

[10] *ITU-T G.729AB Implementation on the StarCore SC140 Core*, Freescale Semiconductor (AN2261).

[11] *ITU-T G.729B Implementation on the StarCore SC140 Core*, Freescale Semiconductor (AN2278).

[12] *3GPP-AMR-NB with ETSI-EFR Implementation on the StarCore SC140 Core*, Freescale Semiconductor (AN2280).

[13] *Speech Coder Filters Using the StarCore SC140*, Freescale Semiconductor (AN2152/D).

[14] *Implementing the Levinson-Durbin Algorithm on the SC140*, Freescale Semiconductor (AN2197).

[15] *Developing Optimized Code for Both Size and Speed on the StarCore SC140 Core*, Freescale Semiconductor (AN2266).

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations not listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GMBH
Technical Information Center
Schatzbogen 7
81829 München, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
+800 2666 8080

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

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

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 Freescale Semiconductor 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. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor 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 Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor 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 Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. StarCore is a trademark of StarCore LLC. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2002, 2004.

Document Order No.: AN2397
Rev. 1
12/2004