ABSTRACT

The purpose of this application note is to provide a detailed description of the Multichannel G711 speech codec optimized for performances on the STMicroelectronics (ST\textsuperscript{TM}) ST122 DSP-MCU system, which is based on the ST100\textsuperscript{®} architecture.

The ST122 DSP-MCU is a 32-bit scoreboarded long instruction word dual MAC DSP-MCU architecture suitable for signal processing algorithms as well as for the control based algorithms execution.

The topics include a G711 speech codec technical overview, an Application Program Interface (API) and performance evaluation in terms of speed and Data/Program memory and an ST122 overview. Multichannel implementation is also addressed.
TABLE OF CONTENTS

1 INTRODUCTION ........................................................................................................ 3
2 TERMINOLOGY ........................................................................................................ 4
3 ST122 OVERVIEW ..................................................................................................... 5
   3.1 MAIN TARGET APPLICATIONS OF ST122 DSP CORES ........................................ 5
   3.2 ST100 DSP CORES KEY ADVANTAGES ................................................................. 5
   3.3 ST122 DSP CORE MAIN FEATURES LIST .............................................................. 6
4 G.711 SPEECH CODER (A-LAW / µ-LAW CODING) OVERVIEW .......................... 8
5 MULTICHANNEL IMPLEMENTATION OF G711 SPEECH CODEC FOR ST122 .... 8
6 API OVERVIEW .......................................................................................................... 9
   6.1 INPUT AND OUTPUT DATA FORMAT DESCRIPTION ..................................... 9
      6.1.1 Input Buffer and Format for the G711 encoder .............................................. 9
      6.1.2 Output Buffer and Format for the G711 encoder ............................................ 9
      6.1.3 Input Buffer and Format for the G711 decoder .............................................. 9
      6.1.4 Output Buffer and Format for the G711 decoder ........................................... 9
   6.2 NAME CONVENTION AND FILE ORGANIZATION ............................................. 10
   6.3 ENCODER INTERFACE FUNCTIONS ................................................................ 10
      6.3.1 G711_alaw_encode() .................................................................................... 10
      6.3.2 G711_ulaw_encode() .................................................................................... 11
   6.4 DECODER INTERFACE FUNCTIONS ............................................................... 12
      6.4.1 G711_alaw_decode() ..................................................................................... 12
      6.4.2 G711_ulaw_decode() .................................................................................... 13
   6.5 CONSTANT DATA ............................................................................................... 13
7 PERFORMANCE RESULTS ....................................................................................... 14
   7.1 EXECUTION TIME ............................................................................................... 14
   7.2 CODE SIZE AND DATA SIZE(BYTES) ................................................................ 14
8 REFERENCES ............................................................................................................ 15
1 - INTRODUCTION

The G711 is a voice compression and decompression standard that has been specified by the ITU-T Recommendation G711 A-Law/µ-Law coding of speech at 64kbits/s. The first ITU-T Recommendation on speech coding G711 was completed in 1972. ITU-T Recommendation G711 was published in Fascicule III.4 of the Blue Book.

G711 speech codec is widely used for voice compression in telecommunications systems such as public switched telephone network (PSTN) telephony. The idea behind digitalization of the network was to take benefit as far as possible of the existing infrastructure in spite of limited bandwidth inherent to the conventional PSTN resulting in a limited communication quality. A rate of 64kbits/s was found to give good quality. G711 is also present in Multimedia for lower audio storage overhead.

The G711 is provided with a set of testing sequences for both encoder and decoder. The code implemented on the STMicroelectronics ST122 Core is fully bit compatible with the bit-exact reference code on all the testing sequences provided in the ITU-T Software Tool Library release of 1996.

The evolution of digital systems is increasingly requiring use of Digital Signal Processors (DSP). At the same time, constraints in cost effectiveness and the need for faster and increased functionality is leading to super integration of DSP, MCU, memory and specific logic into a single multi-tasking System-On-Chip (SOC). The ST100 DSP Core’s family provides flexible, high performance and low power cores, suitable for integration into application specific embedded solutions either as a memory-based DSP data processor or as part of a multiprocessor-based System-On-Chip (SOC).

The ST100 DSP Core’s key advantages include, a super-scalar based architecture with scalable multiple Computational Units working in parallel. The ST100 DSP family aims at bringing the DSP platform of choice wherever high performance, power consumption and time to market are essential. The following section will give more information on the ST122 features.

This application note is addressed to ST100 architecture based programmers, system engineers, tool developers and project managers.
2 - TERMINOLOGY

**G.711**: G.711 (A-law/µ-law coding) speech codec algorithm uses logarithm companding which can compress a 16-bit linear sample down to a 8-bit sample.

**ITU-T**: International Telecommunication Union – Telecommunication Standardization Sector

**Reentrant Program**: A program is said to be reentrant if its execution is independent of all its other executions or this program may be entered repeatedly. In general, this implies that non-constant static data and global data are removed from this code. Reentrance is required for multichannel implementation.

**Multichannel system**: system capable of running more than one algorithm at the same time. One DSP algorithm is applied to multiple channels. The type of algorithm and the channels being applied can change during run-time.

**ST1xx**: DSP family based of the ST100 architecture. This architecture defines scalable DSP’s for telecom, peripheral (e.g. Printer), audio and video applications.

**ST122**: DSP based on a 2 MAC architecture.

**ST140**: DSP based on a 4 MAC architecture.

**3GPP2**: 3rd Generation Partnership Project 2

**Vocoder**: Encoder + Decoder for voice signals
3 - ST122 OVERVIEW

The ST122 DSP is based on the ST100 family of DSP processors. The **ST100 DSP Cores family** provides flexible, high performance and low power cores, suitable for integration into application specific embedded solutions either as a memory-based DSP data processor or as part of a multiprocessor-based System-On-Chip (SOC). Building on ST100 DSP family advanced features, the ST122 is a 32-bit MCU/16bit DSP Load/Store architecture, which provides advanced dual-MAC/dual-ALU DSP capabilities.

3.1 - Main target applications of ST122 DSP Cores

- Wireless Handsets, including advanced mobile Multimedia functions;
- Wireless infrastructures;
- Connectivity, xDSL modems, Voice Over IP;
- Computer peripherals: Data Storage / Printer;
- Automotive;
- Emerging applications requiring high performance signal processing.

3.2 - ST100 DSP Cores key advantages

Based on super-scalar architectures with scalable multiple processing Units (AU and DU) working in parallel, the ST100 DSP family aims at bringing the DSP platform of choice wherever High Performance, Power Consumption and Time to Market are essential.

When defining the ST100 DSP architecture, the focus was put on some important key features:

- **High DSP Performance** combined with efficient control oriented features;
- **Fine balance tuning between Performance and Cost** through flexible instruction modes;
- **Tightly coupled Coprocessor interface** enabling to enrich the instructions sets with powerful user’s customizable instructions used to speed up specific algorithms;
- **Support to Program Cache** (and Data Cache in further versions);
- **Efficient compilers** and tool chains allowing advanced software development methodology using Universal High Level Languages (C/C++);
- **Integration into complex System-On-Chip.** The ST100 DSP Cores are designed to be embedded with other system components on silicon or to be a memory-based data-processing tool. The ST122 can be combined with the extensive cell libraries available from ST, including high-speed logic and memory technologies to provide a true one-chip embedded processor.
3.3 - ST122 DSP Core Main Features List

- **Advanced Load/store Architecture**
  Regular and efficient,
  Optimized for programming in ‘C/C++/EC++’ languages

- **Two Instruction Sets**
  GP32, 32-bit for high performance microcontroller code,
  GP16, 16-bit for compact microcontroller code.

- **One Operating Mode**
  SLIW 4 x 32-bit for high performance vector code (DSP loops).

- **Forward-binary compatible with the ST140 DSP (Four MAC version) at the instruction set level**

- **Flexible Processing**
  2-way superscalar in GP16, GP32.
  One SLIW (4 x GP32) per cycle.

- **Predicated Execution For Most Instructions**
  Removes need for conditional branches.
  Compact coding and Increased instruction level parallelism.

- **Flexible Data Format**
  The ST122 supports the following data types:
  - 16-bit, 32-bit and 40-bit Unsigned/Signed Integer,
  - 16-bit, 32-bit, and 40-bit Signed Fractional,
  - Signed and Unsigned Byte,
  - Supports little Endian for data and program.

- **Circular And Bit-reversed DSP Addressing Modes**
  Facilitates FIR and FFT data processing.

- **Arithmetic Capability**
  40-bit and 32-bit arithmetic,
  SIMD (Packed Arithmetic 2 x 16-bit),
  Saturating (Clamping) and/or Rounding options.
• **Application Oriented Instructions**
  Useful instructions for ETSI primitives in GP32 and GP16:
  – NORMW, Half Word CLAMP, SHLCW, VITERBI...
  General usage instructions:
  – Minimum/Maximum, absolute value, interleave/de-interleave,

• **Hardware Loop Controllers**
  Zero cycles overhead for continuous data processing.
  Three nestable loops.

• **Memory Space**
  One 128-bit Program bus and two data buses,
  32-bit addressing range, 4Gbytes of memory space,
  Flexible memory space.

• **Interrupt, Trap And Context Switching**
  Fast response to external events or system errors,
  32 Interrupt (5-bit) levels,
  Software interrupts allowed by writing to the interrupt controller register.

• **Power Saving**
  Four " IDLE" modes performing power saving operations.
  The Program Cache also implements a low consumption mode (2- cycle access)
  Static design – Clocks can be shut down without loss of data.

• **Tightly Coupled Coprocessor Interface**
  Extension of the ST122 core data unit.

• **Program Cache**
  Configurable Instruction Cache.
4 - G.711 SPEECH CODER (A-LAW / µ-LAW CODING) OVERVIEW

The G711 ITU-T Recommendation covers A-Law/µ-Law pulse code modulation (PCM) at 64kbits/s. The principle is simple: voice input data are sampled in uniform PCM format at 8 KHz. The PCM data are then compressed down to 8 bits per sample. With this combination, each voice channel requires a bit rate of 64kbits/s. G.711 coding uses logarithmic compression: the advantage of such non-linear quantizer is that 8 bits per sample is sufficient to achieve good quality of speech, with uniform signal to noise ratio over a wide range of input levels.

ITU-T Recommendation G.711 specifies two common companding methods in use:
– µ-Law coding for North American and Japanese standards
– A-Law compression for European standards.

The companding codecs in this specification are actually piecewise linear approximation to the non-linear µ-Law and A-Law defining equations [1].

The µ-Law compander compresses a 14-bit signal (linear PCM) down to an 8-bit signal (logarithmic PCM data) and expands it back to 14 bits. An A-Law compander compresses and expands a 13-bit signal (linear PCM) to 8 bits and back again. A-law compander code includes the reversion of even bits.

For detailed description of the G.711 vocoder, you may refer to [1] and [2].

5 - MULTICHANNEL IMPLEMENTATION OF G711 SPEECH CODEC FOR ST122 SOLUTION

This section presents the multichannel solution for G711 Speech codec for the ST122 based communication systems.

Typically a single DSP processor may be required to run not only many algorithms but also more than one channel of a single algorithm at a time. Having separate DSP processors for different algorithms and channels does not really make any sense, as it will increase cost, power and the board space. Hence we need to write the source code for algorithm in such a way that it is possible to:

• Run more than one channel of this algorithm without replicating the whole code. However the global data, which is specific to a particular channel, need to be separately allocated for each channel. This data is usually called context data. A code fulfilling this requirement is called Reentrant. A reentrant code must not be self-modifying.

• To change the type of algorithm and channel during run time. For this, the whole program and data should be made Relocatable.

G.711 implementation includes independent user-callable functions that perform all of the µ-Law, A-Law, encoding and decoding operations. This implementation uses the look-up tables approach for A-Law and µ-Law coding. These look-up tables are defined as global variables in the application.

G.711 vocoder is multichannel enabled in the sense that both functions of G.711, encode and decode are reentrant. There are no context data which are channels dependent. Constant data, that are channel independent, deal with tables for A-Law and µ-Law decoding. These tables are not relocatable in the current implementation and represent 512 bytes (256 bytes for A-law and 256 bytes for µ-Law).
6 - API OVERVIEW
The present G711 API has been designed to define a simple interface for G711 vocoder and is suitable for multichannel. The API defined requires four functional modules: two modules for encoder and two modules for decoder, to address either US standard (µ-Law) or European standard (A-Law).

6.1 - Input and Output data format Description

6.1.1 - Input Buffer and Format for the G711 encoder
The number of samples $Ns$ in the input buffer is in fact a parameter than can be passed on the command line when launching the executable. Those samples are assumed to be quantized in an uniform PCM format with at least 13 Most Significant Bits (MSBs) of dynamic range (for A-Law) and 14 MSBs (for µ-Law). Then the input buffer size is $2 * Ns$ Bytes, each sample being expanded on one 16-bit Word (proper scaling factor have been applied in the implementation to accept 16 bits linear PCM samples). By default, the value of $Ns$ is set to 256 samples in the source file.
For the test performed, the input buffer consists of 256 samples sampled at 8 kHz. Then one speech frame lasts 32 ms.

6.1.2 - Output Buffer and Format for the G711 encoder
For each speech frame, composed of $Ns$ samples, the G711 coder produces an encoded output composed of samples each coded on 8 Least Significant Bit (LSBs). The size of this output is then $Ns$ bytes.

6.1.3 - Input Buffer and Format for the G711 decoder
The input buffer of the decoder has the same format as the encoder output buffer.

6.1.4 - Output Buffer and Format for the G711 decoder
The output buffer of the decoder contains $Ns$ 16-bit synthesized PCM samples.
6.2 - Name convention and File organization
The multichannel layer of the G711 vocoder file organization is the following:

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G711_codec.h</td>
<td>Prototypes of</td>
</tr>
<tr>
<td></td>
<td>- G711_alaw_encode(),</td>
</tr>
<tr>
<td></td>
<td>- G711_ulaw_encode(),</td>
</tr>
<tr>
<td></td>
<td>- G711_alaw_decode()</td>
</tr>
<tr>
<td></td>
<td>- G711_ulaw_decode()</td>
</tr>
<tr>
<td>G711_codec.c</td>
<td>Definition of the high-level</td>
</tr>
<tr>
<td></td>
<td>- G711_alaw_encode(),</td>
</tr>
<tr>
<td></td>
<td>- G711_ulaw_encode(),</td>
</tr>
<tr>
<td></td>
<td>- G711_alaw_decode()</td>
</tr>
<tr>
<td></td>
<td>- G711_ulaw_decode()</td>
</tr>
<tr>
<td>G711_const.c</td>
<td>Declaration of all the constant G711 look-up tables for A-law</td>
</tr>
<tr>
<td></td>
<td>and µ-law (coder and decoder)</td>
</tr>
<tr>
<td>G711_const.h</td>
<td>Definition of the g711 look-up tables</td>
</tr>
<tr>
<td></td>
<td>(coder and decoder)</td>
</tr>
</tbody>
</table>

6.3 - Encoder Interface functions

6.3.1 - G711_alaw_encode()

SYNOPSIS

void G711_alaw_encode( int smpno, short * linbuf, short * logbuf)

<table>
<thead>
<tr>
<th>Function</th>
<th>Required header</th>
</tr>
</thead>
<tbody>
<tr>
<td>G711_alaw_encode()</td>
<td>G711_codec.h</td>
</tr>
</tbody>
</table>

DESCRIPTION

G711_alaw_encode performs A-Law encoding rule according to ITU-T Recommendation G.711: it compresses 13 MSBs of linear PCM to 8 LSBs logarithmic PCM.

Notes: output samples are 8-bit right-justified.
6.3.2 - G711_ulaw_encode()

SYNOPSIS

void G711_ulaw_encode( int smpno, short * linbuf, short * logbuf)

DESCRIPTION

G711_ulaw_encode performs µ-Law encoding rule according to ITU-T Recommendation G.711: it compresses 14 MSBs of linear PCM to 8 LSBs logarithmic PCM.

Notes: output samples are 8-bit right-justified.

PARAMETERS

<table>
<thead>
<tr>
<th>Name</th>
<th>In/Out</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smpno</td>
<td>In</td>
<td>Number of samples in linbuf to be encoded</td>
</tr>
<tr>
<td>linbuf</td>
<td>In</td>
<td>Input sample’s buffer; each short sample shall contain linear PCM (2’s complement, 16-bit wide) samples, left-justified</td>
</tr>
<tr>
<td>logbuf</td>
<td>Out</td>
<td>Output sample’s buffer; each short sample will contain right-justified 8-bit wide valid µ-law samples</td>
</tr>
</tbody>
</table>

RETURN VALUE None
6.4 - Decoder interface functions

6.4.1 - G711_alaw_decode()

SYNOPSIS

```c
void G711_alaw_decode( int smpno, short * logbuf, short * linbuf)
```

<table>
<thead>
<tr>
<th>Function</th>
<th>Required header</th>
</tr>
</thead>
<tbody>
<tr>
<td>G711_alaw_decode()</td>
<td>G711_codec.h</td>
</tr>
</tbody>
</table>

DESCRIPTION

G711_alaw_decode performs A-Law decoding rule according to ITU-T Recommendation G.711: it expands 8-bit (8 LSBs) logarithmic PCM to 13-bit linear PCM (13 MSBs).

Notes: output samples will be 13-bit left-justified, and the input samples will be 8-bit right-justified.

PARAMETERS

<table>
<thead>
<tr>
<th>Name</th>
<th>In/Out</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smpno</td>
<td>In</td>
<td>Number of samples in logbuf to be decoded</td>
</tr>
<tr>
<td>logbuf</td>
<td>In</td>
<td>Input sample's buffer; each short sample will contain right-justified 8-bit wide valid A-law samples</td>
</tr>
<tr>
<td>linbuf</td>
<td>Out</td>
<td>Output sample’s buffer; each short sample will contain linear PCM (2's complement, 16-bit wide) samples, left-justified</td>
</tr>
</tbody>
</table>

RETURN VALUE

None
6.4.2 - G711_ulaw_decode()

SYNOPSIS

```c
void G711_ulaw_decode( int smpno, short * logbuf, short * linbuf);
```

<table>
<thead>
<tr>
<th>Function</th>
<th>Required header</th>
</tr>
</thead>
<tbody>
<tr>
<td>G711_ulaw_decode()</td>
<td>G711_codec.h</td>
</tr>
</tbody>
</table>

DESCRIPTION

G711_ulaw_decode performs µ-Law decoding rule according to ITU-T Recommendation G.711: it expands 8-bit (8 LSBs) logarithmic PCM to 14-bit linear PCM (14 MSBs).

*Notes: output samples will be 14-bit left-justified, and input samples will be 8-bit right-justified.*

PARAMETERS

<table>
<thead>
<tr>
<th>Name</th>
<th>In/Out</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smpno</td>
<td>In</td>
<td>Number of samples in logbuf to be decoded</td>
</tr>
<tr>
<td>logbuf</td>
<td>In</td>
<td>Input samples' buffer; each short sample shall contain right-justified 8-bit wide valid µ-law samples</td>
</tr>
<tr>
<td>linbuf</td>
<td>Out</td>
<td>Output sample's buffer; each short sample will contain linear PCM (2’s complement, 16-bit wide) samples, left-justified</td>
</tr>
</tbody>
</table>

RETURN VALUE

None

6.5 - Constant data

Both modules in Decoder interface use the look-up tables declared as global in the G.711 application (one look-up table of 256 bytes for A-Law decoding and one look-up tables of 256 bytes for µ_Law decoding) which are not relocatable in the current implementation. These read-only tables are defined in G711_const.h and declared in G711_const.c files.
The G711 speech codec was optimized for the ST122 to achieve a target below 0.7 MCPS as per the project requirement. In order to reach this target, the A-Law/µ-Law functions were first optimized at C-level. Then some logic ST122 compiler options characteristic for G711 application were also used. The most important optimizations performed by the compiler are register allocation, unrolling technique and predication that enables the removal of branches and the elimination of the misprediction penalty associated with branches. Predication option, in this code, has a great impact on performances.

7.1 - Execution time

Table 1 gives the worst-case cycle count of one G711 A-Law/µ-Law speech codec for one channel. Performances were obtained using the Toolset version 2.1.0 (stcc compiler 3.4, Cycle-Based Simulator version release 1.1.0, ISS standalone fast functional simulator release 1.1.15) and measured on the current chip (with program cache activated). The codec implementation is a FULL C implementation.

<table>
<thead>
<tr>
<th></th>
<th>A-Law</th>
<th>µ-Law</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encode</td>
<td>0.2014 MCPS</td>
<td>0.257 MCPS</td>
</tr>
<tr>
<td>Decode</td>
<td>0.097 MCPS</td>
<td>0.097 MCPS</td>
</tr>
<tr>
<td>Total MCPS (Worst figures)</td>
<td>0.298</td>
<td></td>
</tr>
<tr>
<td>Total MCPS (Worst figures)</td>
<td>0.354</td>
<td></td>
</tr>
</tbody>
</table>

Notes: *Million of Cycles Per Second of processed speech

7.2 - Code Size and Data Size (Bytes)

The following table summarizes the estimated number of bytes for the G.711 A-Law/µ-Law speech codec program memory and data memory.

<table>
<thead>
<tr>
<th></th>
<th>A-Law</th>
<th>µ-Law</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Size</td>
<td>Encode</td>
<td>Decode</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>80</td>
</tr>
<tr>
<td>Total : 260</td>
<td>Total : 256</td>
<td>Total : 320</td>
</tr>
</tbody>
</table>

Notes: Memory Size Figures (in Bytes)

- Stack size measured through st122 tools << sticg – stack executable >>: 488 bytes
- Data size is equal to 0 Bytes.
8 - REFERENCES


[3] ST122 DSP Core Reference Guide


Table 1. Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Description of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2004</td>
<td>1</td>
<td>First Issue</td>
</tr>
</tbody>
</table>
The present note which is for guidance only, aims at providing customers with information regarding their products in order for them to save time. As a result, STMicroelectronics shall not be held liable for any direct, indirect or consequential damages with respect to any claims arising from the content of such a note and/or the use made by customers of the information contained herein in connection with their products.