CLID DETECTOR
R2.9

Data Sheet

The Data contained in the document is preliminary and is subject to change.

TABLE OF CONTENTS

1 Introduction

Caller ID (CLID) means the identification of information about the calling party, which is sent by Central Office (CO) to the called party (its Customer Premises Equipment (CPE)), either off-hook or on-hook. The data transmission and reception is governed by the following BellCore documents:

- TR-TSY-000030;
- TR-NWT-000030;
- SR-TSV-002476;
- As well as additional technical reference and special report bulletins.

This R2.9 revision of the CLID detector provides the TYPE 1 (on-hook, between first and second ring, or before first ring) signal detection and returns the message raw byte data without parsing of particular fields such as Message Type, Parameter(s) Type(s), etc. The decoding of the message meaning should be performed by the user application.

The CLID detector design provides for extendibility (TYPE2, CAS) when and if required by Licensee.

The C54x version of CLID was not optimized for MIPS performance because it never shares processor with other MIPS consuming signal-processing components as echo canceller or signal detectors (they are idle during on-hook state).

1.1 Features

- Designed as a telecom system component rather than a stand-alone algorithm.
- Can be configured on the fly, and such parameters as twists, frequency acceptance, spectrum cleanness, and signal duration thresholds (for CAS detection) can be altered during a call. The granularity of each parameter is about 0.25 dB (and 0.5 Hz for frequency acceptance margins).
- Provides estimates on the frequency and energy of detected CAS tones, which may help to identify system –wide problems.
- Wide dynamic range,
- High noise immunity.

1.2 Component List

1. CLID detector data sheet (this document)
2. Iclid.h xDAIS-compliant header file.
3. Iclid.c xDAIS-compliant C file with default configuration data.
4. clid_miket.h xDAIS-compliant header file.
5. clid_miket.l54 xDAIS-compliant library.
6. clid.exe PC-based demo program (TBD).
7. clid_readme1st.txt documentation to the demo program (TBD).

1.3 Abbreviations

TBD

1.4 References

TBD

2 Functional Overview

The proposed "Express DSP" compliant C55x /C54x CLID detector is based on multi-rate signal processing and modern complex domain applied spectral analysis methods and approaches. Frequency content analysis is optimized for short signals, and it provides very reliable results. It takes advantage of C55x architecture with high degree of parallelism, dual MAC, and three simultaneous data read paths.

The CLID detector consists of the following blocks:

- Band-pass filter;
- Re-sampling and smoothing filter;
- Matched filter – based amplitude-insensitive frequency discriminator;
- Maximum – likelihood based bit and byte processors;
- Control logic.

The band-pass filter is designed in accordance with recommendations given by The BellCore Technical Reference of September 1993, containing changes to TR-NTW-000030 and SR-TSV-002476. The band-pass filter eliminates both low and high frequency interference, characteristic for local loops without proper AC termination. The band-pass filter transforms the input signal into complex domain to facilitate further processing.

The re-sampling and smoothing filter was introduced to improve the overall CLID detector performance. It was optimized via stochastic simulation of the entire CLID detector. The optimization criterion was to provide 'the best performance in the worst conditions': no more than 5% packet loss ratio in conditions of the worst SNR for the local loop with approximately 6 dB twist between 1200 and 2200 Hz tones.

- As the result of optimization, the CLID detector outperforms competitive products by at least 4 dB of SNR.
- The CLID detector requires only 6 dB of SNR between the lowest tone amplitude and AWGN to provide at least 95% packet detection rate (50 byte packets). This corresponds to BER of 10^{-4}.
- The BER of CLID detector (and its packet loss ratio) improves drastically with SNR increase. If SNR is only 2 dB better, the BER drops to 10^{-8} what corresponds to practically failure-free operations.

The re-sampled data is fed into the amplitude-insensitive frequency discriminator based on the matched filter (MF) theory. This design allows avoiding use of AGC and ensuring full dynamic range coverage. The CLID detector operates even if the signal level equals to the 1 LSB of the codec resolution.

The MF output is fed into bit and byte detectors, which:

- Adjust the byte sampling position to the maximal eye opening
- Take into account the stop and start bit conditioning.
- Chose the bits of the current message word to provide the best possible fit to the incoming waveform.

The control logic block provides overall control over CLID detector operations and interfaces to the user application. If a valid CLID message has been detected, the control logic block signals user application and keeps the message unmodified until the user explicitly reset the CLID detector.

The CLID detector performs 5 ms frame size processing.

3 API

3.1 Data structures

3.1.1 Data Base

Database is an object that keeps data, which should be preserved between calls to process incoming data. It should be properly initialized and shall not be overwritten by any other application. It may be moved into another location without limitations. The ‘best’ placement for database is internal DARAM, but performance shall not suffer drastically even if the database is placed in external RAM. If any performance problem arises, the object may be copied via DMA (in and out) into a temporary location in internal RAM.

Database size:

- 260 Words (16 bit), long word (32 bit) alignment

The database starts from 128 W of message storage space. The consecutive message bytes are packed in words, first in a LS byte and next in MS byte.

The content and meaning of following fields of the database are undisclosed in this document.

3.1.2 Scratch pad

Scratch pad is an object that is used during calls to process incoming data. It does not need to be initialized. It may be overwritten by any other application. It may reside in a new place each time a processing function is called: i.e. it may be moved into another location without limitations. The ‘best’ placement for database is internal DARAM. Performance will suffer drastically even if the database is placed in internal SARAM.

Scratch pad size:

- 430 Words, long word alignment

The content and meaning of particular fields of the scratch pad are undisclosed in this document.

3.1.3 Configuration

Configuration is an object, which contain important decision-making parameters. The data there is READ-ONLY by CLID functions. The configuration data may be altered by external application on the fly, but excluding periods of time when a CLID detector processes a valid CLID signal.

Note that `Int` is defined in TI’s supplied `ialg.h` file as 16 bit signed word.

```
typedef struct ICPD_tCfg {
    Int sNormShift;
    Int sMinEn;
    Int sMaxEn;
    Int sMaxTwist;
    Int sMaxFreqDev;
} ICPD_tCfg;
```

All relevant parameters are expressed in dB, scaled up with a coefficient, so that 3.0103 dB corresponds to 512. Thus 1 dB will correspond to approximately 170.083 ~ 170 = ICPD_1DB. The same definition is applied to Frequency deviation, but here 1% of frequency deviation corresponds to 170.083 (the same value).

The fields of configuration have the following meaning:

Parameter	Description	Recommended Range
NormShift	'Normalisation': number of left shifts to align data so that 0dBm0 corresponds to 2048 DC. If right shifts are required, use negative values	0
MinEn	Relevant for CAS reception only. Minimum energy of CAS signal to be detected, in dBm	-35…-32 dBm
MaxEn	Relevant for CAS reception only. Maximum energy of CAS signal to be detected, in dBm.	-14…–11 dBm
MaxTwist	Relevant for CAS reception only. Maximum twist between tones.	6…9 dB
MaxFreqDev	Relevant for CAS reception only. Maximum frequency deviation in percents	0.5…0.8 %

3.1.4 CAS Statistics

CLID detector provides following statistics of the incoming signal, to facilitate in field testing:

Parameter	Description	Precision
Start	The number of times a valid start of a CLID signal was detected.	~1 dB
End	The number of times a valid CLID signal was detected and reported.	~1 dB
Abort	The number of times a valid start of a CLID signal was detected but some errors occurred and the data was not reported.	~0.2Hz
CAS	The number of times a valid CAS signal was detected and reported.	~0.2Hz

The signal measurements are reported within the following structure:

```
typedef struct CLID_MIKET_tStts {

    Int sStart;
    Int sEnd;
    Int sAbort;
    Int sCas;

} CLID_MIKET_tStts;
```

3.2 IALG API

XDAIS compliant IALG interface is fully supported.

3.3 Vendor specific API

3.3.1 Initialization

```
extern void CLID_MIKET_init_db (void *pDb, ICLID_tCfg *pCfg);
```

pDb shall refer to the user-allocated properly aligned memory block.
PCfg shall refer to a valid configuration structure.

3.3.2 Control

```
extern void CLID_MIKET_control
(void *pDb, Int Cmd, ICLID_Status *pStatus );
```

Cmd can constructed by OR-ed flags:

```
#define ICLID_CMD_OFF          (1)   // disable detector from running.
#define ICLID_CMD_RESET        (2)   // reset database variables, data saving buffers
#define ICLID_CMD_CFG          (4)   // update configuration
#define ICLID_CMD_TYPE1        (8)   // start CLID TYPE 1 detection
#define ICLID_CMD_TYPE2        (16)  // start CLID TYPE 2 detection
#define ICLID_CMD_CAS          (32)  // start CLID CAS detection
#define ICLID_CMD_RESET_STTS   (64)  // reset statistics
```

It is not recommended to apply OFF without RESET due to the obvious consequences of calling _control(pDb, 0); afterwards. Asynchronous calling _control() and _process() from different tasks is not recommended. All other flags will be ignored.

3.3.3 Process

```
extern Int CLID_MIKET_process(void *pDb, void *pSc, Int *pIn);
```

Reports Int word: MSByte as event:

Event	Value	Meaning
ICLID_EV_NONE	0	No changes happened this frame
ICLID_EV_EARLY_ON	(1<<8)	Some valid reversals were detected; a valid CLID sequence is expected to follow.
ICLID_EV_EARLY_OFF	(2<<8)	The previous event was erroneous
ICLID_EV_START	(3<<8)	The first byte of the data was detected.
ICLID_EV_END	(4<<8)	Complete message was detected, and check sum matches.
ICLID_EV_ABORT	(5<<8)	Something went wrong with this message.

Normally, user shall get a sequence of

- EV_EARLY_ON,
- EV_START,
- EV_END.

Note that:

- pDb must point to initialized database.
- pSc must point to scratch pad. pSc shall point to DARAM if max speed is required, otherwise MIPS will almost double.
- pIn shall point to a frame (40) of continuous data.
- If CLID was turned off, then pSc and pIn can point anywhere.

3.4 C54x Linking

NOTE: The sections: .clida, .clidb, .clidc must be linked in program space. The preferable location is internal RAM (OVLY=1).

3.5 Using CLID detector

The user starts the CLID detector by applying ICLID_CMD_TYPE1 command via _control function. The CLID detector reserves 128 words in its database for the maximum length message to be stored there.

The CLID detector reports its progress with events _EARLY_ON, _EARLY_OFF, _START, _END, _ABORT. The _END event signals that the entire message is ready for retrieval. The primary use of other events is troubleshooting and field problem resolution.

The CLID detector will wait indefinitely and will not modify the received message data after it signals the user application with _END event. Thus the user application can retrieve the data (stored at the very start of database) at their convenience. The message length byte is MSB of word 0.

The user application may issue _OFF command immediately after receiving _END event. This will zero MIPS consumption.

The CLID detector may be reset after the data is retrieved (data storage will be zeroed). The CLID detector shall be reset if it fails to detect the message before the second ring comes. Otherwise it will consume MIPS unnecessarily.

The CLID detector provides simple statistics (number of messages received and lost) to facilitate system testing and field troubleshooting. The statistics may be expanded in future releases if and when required by the customer.

4 Specifications

4.1 General

Parameter	C54x version	C55x version
MIPS / Instance	3.2 MIPS	TBD MHz
Algorithm Memory (DARAM & SARAM), words	1.5 kW	TBD kW
Instance Memory (SARAM), words	230 W	182 W

4.2 Performance characterization

There may be no warranty given that it will operate perfectly under all and any life circumstances. Note that CLID detector operates with real-world data, it can be 'close to perfect', but it cannot ever possibly be absolutely perfect. The performance of CLID detector depends on the configuration parameter settings. CLID detector may become dysfunctional if those settings are inappropriate. Ensure that a user understands the meaning of parameters and consequences of wrong settings before any changes are applied.

There are numberless variations of test conditions, and only some of them may be of particular interest to a Licensee. The demo program provided is a bit-exact simulation of CLID detector. It may be used to characterize the performance of CLID detector in conditions of particular interest, if different from test cases provided.

4.2.1 Default Configuration

The following table shows the default CP detector's settings.

Parameter	Value
NormShift	0
MinEn	-34.0 dBm
MaxEn	-12 dB
Twist	8 dB
MaxFreqDev	0.7%

4.2.2 Test Setup.

CLID detector passes all tests with signals generated by either Consultronics or other similar equipments. The performance of CLID detector was further evaluated in simulator with

- a CLID generator (-25dBm default),
- followed by signal filtering (5 types)
- and AWGN (varied in 1 dB steps).

The frequency response of the filters used is given on Figure 1.

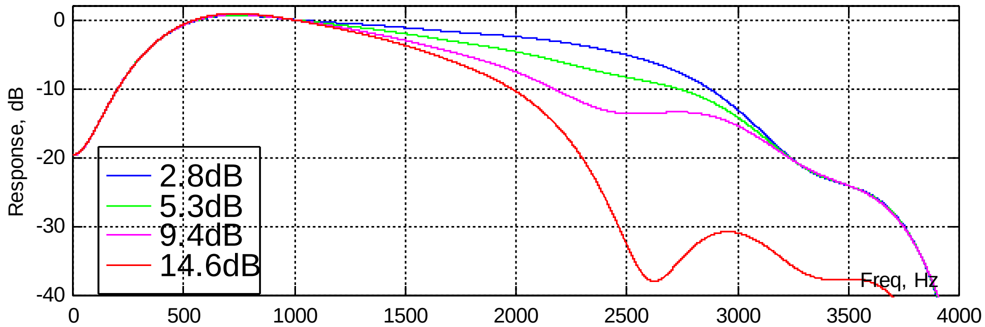

Figure 1. Application of the filters results in twist, increasing from approximately 3 dB to 15 dB. [1] states that maximal twist observed in field was 5.6 dB.

4.2.3 Packet Error Rate

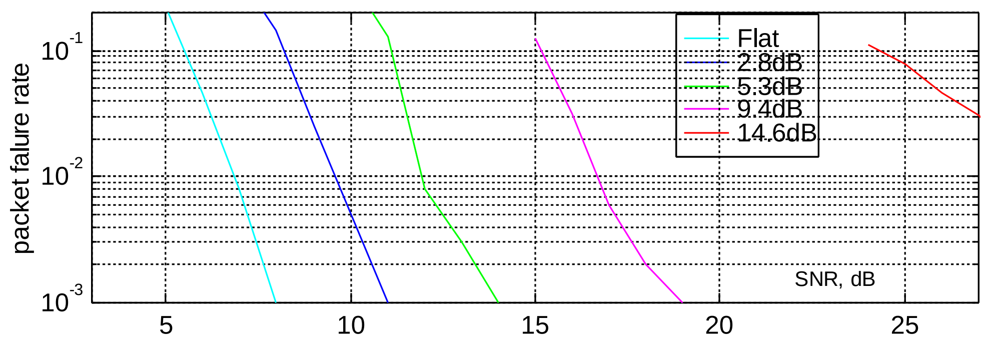

Figure 2. The corresponding packet failure rate for 50 byte long packets grows in proportion to the increasing twist. To get bit error rate, divide this packet failure ratio by 400.

The packet error ratio is fairly stable across dynamic range:

Signal Level, dBm	Noise level, dBm	Racket Failure Ratio
-15	-21	0.046
-25	-31	0.046
-35	-41	0.037
-45	-51	0.023
-55	-61	0.010

4.2.4 Flat (no twist)

The corresponding packet failure ratio is:

Noise level, dBm	Racket Failure Ratio
-33	<0.001
-32	0.008
-31	0.046
-30	0.227
-29	0.753

4.2.5 Low twist (3dB)

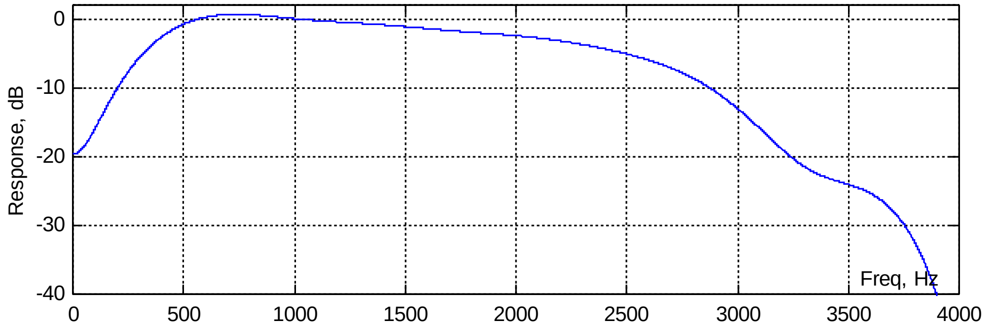

Figure 3. This filter provides about 2.75 dB difference between 1200 and 2200 Hz. The corresponding packet failure ratio is:

Noise level, dBm	Racket Failure Ratio
-36	<0.001
-35	0.005
-34	0.025
-33	0.146
-32	0.412

4.2.6 Medium Twist (5 dB)

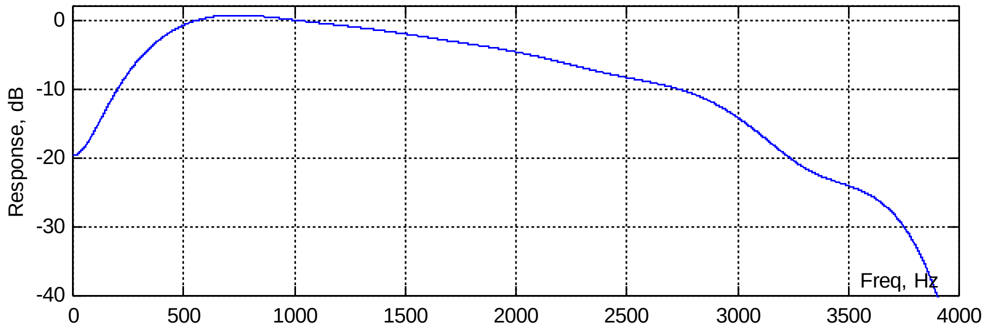

Figure 4. This filter provides about 5.3 dB difference between 1200 and 2200 Hz. The corresponding packet failure ratio is:

Noise level, dBm	Racket Failure Ratio
-39	<0.001
-38	0.003
-37	0.008
-36	0.130
-35	0.380

4.2.7 High twist (10 dB):

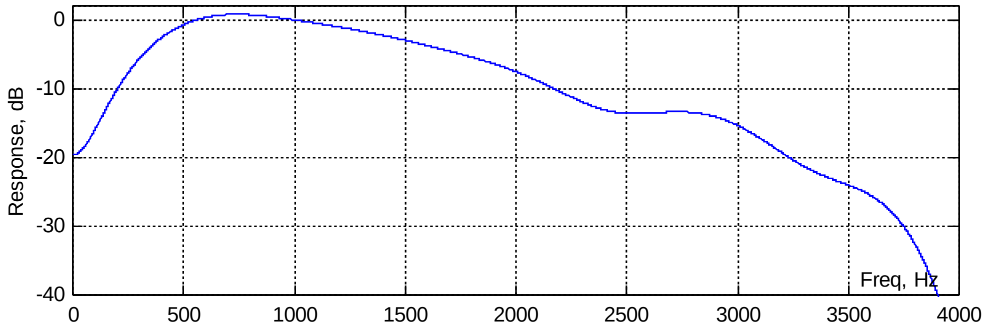

Figure 5. This filter provides about 9.4 dB difference between 1200 and 2200 Hz. The corresponding packet failure ratio is:

Noise level, dBm	Racket Failure Ratio
-45	<0.001
-44	0.002
-43	0.006
-42	0.032
-41	0.126
-40	0.275

4.2.8 Very High Twist (15 dB)

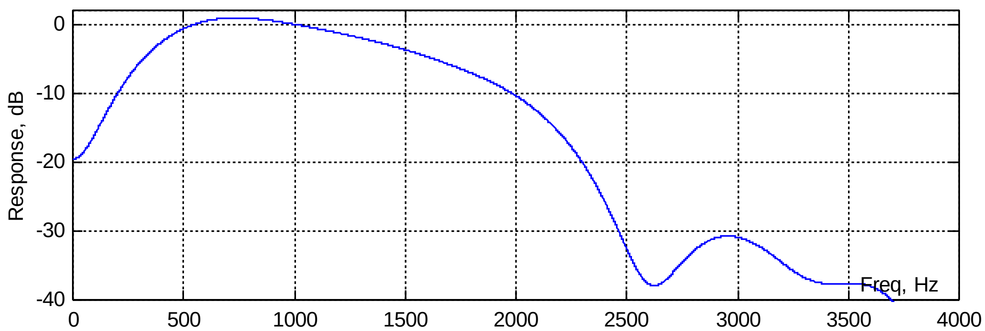

Figure 6. This filter provides about 14.6 dB difference between 1200 and 2200 Hz. The corresponding packet failure ratio is:

Noise level, dBm	Racket Failure Ratio
-54	0.015
-53	0.031
-52	0.046
-51	0.078
-50	0.112