VS1005 VSOS3 EQUALIZER FTEQU

VS1005 VSOS3

Project Code:
Project Name: VSMPG

All information in this document is provided as-is without warranty. Features are subject to change without notice.

Revision History

Rev.	Date	Author	Description
1.05	2017-03-14	HH	Corrected memory corruption bug in conjunction with AUODAC.
1.04	2016-02-01	HH	Documentation update.
1.03	2015-09-09	HH	Might make audio choppy, fixed.
1.02	2015-09-04	HH	In some cases would only set first audio channel, fixed.
1.01	2015-08-31	HH	Created this PDF document. Software is unchanged.
1.00	2015-08-28	HH	Initial release.

Contents

List of Figures

1 Introduction

This document is an instruction manual on how to use the VS1005 VSOS equalizer.

Chapter 4 describes the application.

Chapter 5 show how to set it up and run it.

Chapter 6 presents some C data structures needed by those who use a VSOS version older than v3.23.

VLSI's recommendations for how to set up the filters are given in Chapter 7.

Finally, Chapter 8 contains VLSI Solution's contact information.

2 License

This code may be used freely in any product containing one or more ICs by VLSI Solution.

3 Disclaimer

No guarantee is given for the usability of this code.

4 The VS1005 VSOS3 Equalizer FtEqu

The FtEqu package offers upto 16 channels of high-quality equalization filtering. To use the package, copy the .DL3 files to your VS1005 system disk SYS/ folder, then follow the instructions in this manual. The equalizer can be accessed both from C programs using ioctl() calls, or from the VSOS Shell command line.

The package consists of the following programs:

Name	Description
FTOEQU	Equalization driver that connects to stdaudioout
SETEQU	User program to set parameters for the equalizer

The processing power requirement if approximately 3.5 MIPS + 4.25 MIPS for each stereo filter at 48 kHz (double that at 96 kHz). If more than one filter channel is used, it may be required to use a higher clock speed than VS1005's default 60 MHz. Clock speed can be set e.g. with VSOS Shell command SETCLOCK.

5 Loading and Unloading the Equalizer

5.1 Loading the Equalizer from config.txt

Loading FTOEQU.DL3 in config.txt and setting first filter to attenuate bass:

```
# Starts equalizer and connects to stdaudioout
FTOEQU
# Set L+R @ 300 Hz to -12 dB, Q factor 0.7
RUN SETEQU 1 3 130 -12 0.7
```

Note: FTOEQU is loaded and stays in memory. Because SETEQU doesn't need to stay in memory, it is started with the RUN command.

5.2 Loading the Equalizer Using the VSOS Shell

The example below loads the equalizer driver, and sets the first filter in L+R mode @ 300 Hz to -12 dB, Q factor 0.7:

```
S:>driver +ftoequ
S:>setequ 1 3 130 -12 0.7
```

5.3 Unloading the Equalizer Using the VSOS Shell

```
S:>driver -ftoequ
```

5.4 Loading the Equalizer Manually from C Code

```
// Change name as necessary
#define LIB_NAME "ftoequ"

u_int16 *equLib = NULL;
FILE *equFP = NULL;

equLib = LoadLibrary(LIB_NAME);
if (!equLib) {
  printf("Cannot load " LIB_NAME ".DL3 library\n");
  goto finally;
}
equFP = stdaudioout;
```

Note that if you have started FTOEQU from config.txt, you don't need to open it from your software, but instead you can directly send ioctl() calls to stdaudioout.

5.5 Unload the Equalizer from C Code

If loaded using C code, FTOEQU can be unloaded as follows:

```
finally:
if (equFP) {
  /* Close file */
  RunLoadedFunction(i2sLib, ENTRY_4, (void *)equFP);
  equFP = NULL;
}
if (equLib) {
  DropLibrary(equLib);
  adcLib = NULL;
}
```

5.6 Controlling the Equalizer from the VSOS Shell

From the VSOS Shell, you can e.g. Set L+R @ 300 Hz to -12 dB, Q factor 0.7:

```
S:>setequ 1 3 130 -12 0.7
```

To read the status of the first filter:

```
S:>setequ 1
```

To clear the first filter:

```
S:>setequ 1 0
```

To read all filters:

```
S:>setequ
```

For more options:

```
S:>setequ -h
```

Example: Set the Bass/Treble control shown in Chapter 7.1:

```
S:>setequ 1 3 100   XXX 0.7
S:>setequ 2 3 10000 YYY 0.7
```

where you must replace XXX and YYY with the user-selected gains in dB.

5.7 Controlling the Equalizer from config.txt

Example: Set the Bass/Treble control shown in Chapter 7.1:

```
run setequ 1 3 100   XXX 0.7
run setequ 2 3 10000 YYY 0.7
```

where you must replace XXX and YYY with the user-selected gains in dB.

5.8 Controlling the Equalizer from C Code Using ioctl()

All ioctl controls except those listed here are forwarded to the underlying audio driver.

5.8.1 Read How Many Filters Are Available

If there is no equalizer in the audio path, S_ERROR will be returned.
Example:

```
#include <aucommon.h>
#include <vsos.h>
s_int16 n = ioctl(equFP, IOCTL_GET_EQU_MAX_FILTERS, NULL);
```

5.8.2 Read Status of One Equalizer Filter

The example below shows how to read filter 0.

```
#include <aucommon.h>
#include <vsos.h>
struct FilterEqualizer fltEqu;
fltEqu.filterNumber = 0; /* 0 .. MAX_FILTERS-1 */
if (ioctl(fp, IOCTL_AUDIO_GET_EQU_FILTER, (void *)(&fltEqu)) == S_OK) {
  printf("%2d: flags 0x%04x, %10.2f Hz, %6.2f dB, Q %5.2f\n",
         fltEqu.filterNumber+1, fltEqu.flags, fltEqu.centerFrequencyHz,
         fltEqu.gainDB, fltEqu.qFactor);
} else {
  printf("%2d: ERROR\n", fltEqu.filterNumber+1);
}
```

5.8.3 Set an Equalizer Filter

The example below shows how to set the first filter both to left and right audio channel:

```
#include <aucommon.h>
#include <vsos.h>
struct FilterEqualizer fltEqu;
fltEqu.filterNumber = 0;
fltEqu.flags = FLT_EQUF_LEFT | FLT_EQUF_RIGHT;
fltEqu.centerFrequencyHz = 130.0;
fltEqu.gainDB = -12.0;
fltEqu.qFactor = 0.7;
ioctl(fp, IOCTL_AUDIO_SET_EQU_FILTER, (void *)(&fltEqu));
```

(Note: to disable a filter, set fltEqu.flags to 0.)

5.8.4 Example

Example: Set the Bass/Treble control shown in Chapter 7.1:

```
#include <aucommon.h>
#include <vsos.h>

struct FilterEqualizer fltEqu;
double XXX, YYY;

fltEqu.filterNumber = 0;
fltEqu.flags = FLT_EQUF_LEFT | FLT_EQUF_RIGHT;
fltEqu.centerFrequencyHz = 100.0;
fltEqu.gainDB = XXX;
fltEqu.qFactor = 0.7;
ioctl(fp, IOCTL_AUDIO_SET_EQU_FILTER, (void *)(&fltEqu));

fltEqu.filterNumber = 1;
fltEqu.flags = FLT_EQUF_LEFT | FLT_EQUF_RIGHT;
fltEqu.centerFrequencyHz = 10000.0;
fltEqu.gainDB = YYY;
fltEqu.qFactor = 0.7;
ioctl(fp, IOCTL_AUDIO_SET_EQU_FILTER, (void *)(&fltEqu));
```

where you must replace XXX and YYY with the user-selected gains in dB.

6 Data Structures

If using an earlier version than VSOS 3.23, the following data structures are not yet in aucommon.h, so you need to add this to your source code:

```
#ifndef IOCTL_AUDIO_GET_EQU_FILTER
#define IOCTL_AUDIO_GET_EQU_FILTER 233
#define IOCTL_AUDIO_SET_EQU_FILTER 234
#define IOCTL_AUDIO_GET_EQU_MAX_FILTERS 235

#define FLT_EQUF_LEFT 1
#define FLT_EQUF_RIGHT 2

struct FilterEqualizer {
  s_int16 filterNumber; /* 0 .. MAX_FILTERS-1 */
  u_int16 flags; /* FLT_EQUF_ flags */
  double centerFrequencyHz;
  double gainDB;        /* Recommended -12.0 .. +12.0 dB */
  double qFactor;       /* Recommended 0.1 .. 4.0 (higher is steeper) */
};

#endif /* !IOCTL_AUDIO_GET_EQU_FILTER */
```

7 Recommended Filter Values

This Chapter gives some VLSI recommendations for how to set the filters for a Bass/Treble control, or 3-channel, 5-channel, and 7-channel equalizers.

While these are by no means definitive rules, people making filters are recommended to set them up so that the frequencies are in ascending order. This will make it possible to later build a system control that can control equalizers even if the system control has not set it up to begin with.

Note that when you set filters, it is recommended to keep the gain between -12 and +12 dB, and the Q factor between 0.5 and 4.0.

7.1 2-Band Bass/Treble Control

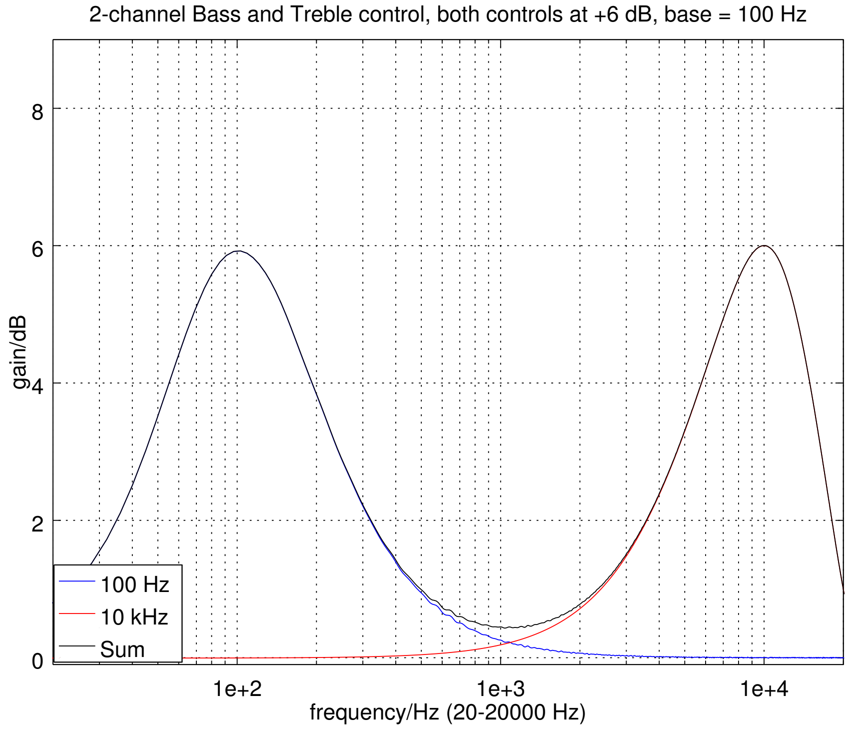

Figure 1: 2-Band Bass/Treble Control.

Filter Number	Frequency/Hz	Q Factor
0	100	0.7
1	10000	0.7

Figure 1 shows the frequency responses for the individual bands of a 2-band equalizer when both bands are set to +6 dB, as well as their total. This filter set is useful for a small speaker.

The exact frequency values should be adapted to the application. For instance, the best frequency for a bass control for a large Hi-Fi speaker may be downto 60 Hz, while a small PC table speaker might sound better with the bass control at 150 Hz. The exact Q factor values are also a matter of taste. Generally, the best range is between 0.5 (gentle curves) and 1.0 (steeper curves).

The processing power requirement for this filter at 48 kHz, in stereo, is approximately 12.0 MIPS.

7.2 3-Band Equalizer

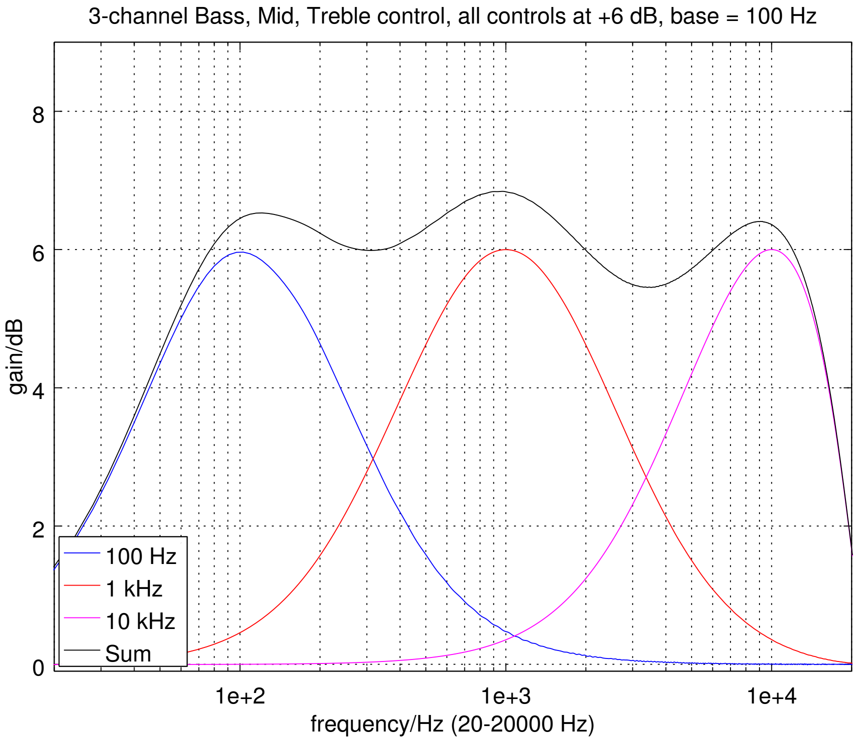

Figure 2: 3-Band Equalizer with 100 Hz bass control.

Filter Number	Frequency/Hz	Q Factor
0	100	0.5
1	1000	0.5
2	10000	0.5

Figure 2 shows the frequency responses for the individual bands of a 3-band Bass/Mid/Treble control when all of the bands are set to +6 dB, as well as their total. This filter set is useful for a small speaker.

The exact frequency values should be adapted to the application. For instance, the best frequency for a bass control for a large Hi-Fi speaker may be downto 60 Hz, while a small PC table speaker might sound better with the bass control at 150 Hz. The Q values here have been selected to give the most even frequency response possible when all controls are driven in the same direction.

The processing power requirement for this filter at 48 kHz, in stereo, is approximately 16.2 MIPS.

7.3 5-Band Equalizer, Bass = 60 Hz

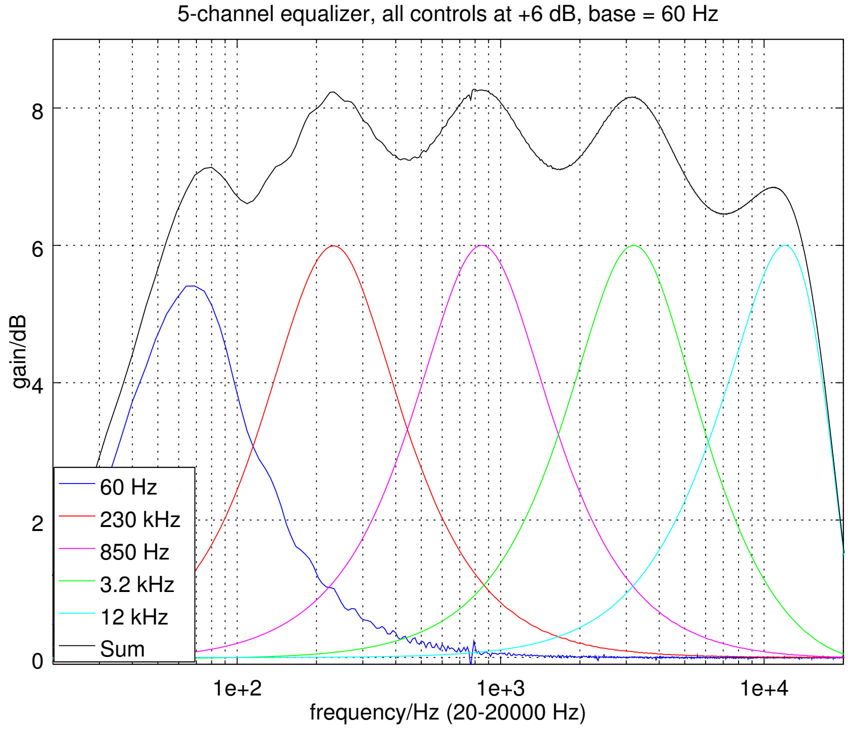

Figure 3: 5-Band Equalizer with 60 Hz bass control.

Filter Number	Frequency/Hz	Q Factor
0	60	0.9
1	230	0.9
2	850	0.9
3	3200	0.9
4	12000	0.7

Figure 3 shows the frequency responses for the individual bands of a 5-band equalizer when all of the bands are set to +6 dB, as well as their total. This filter set is useful for a large speaker.

The processing power requirement for this filter at 48 kHz, in stereo, is approximately 24.7 MIPS.

7.4 5-Band Equalizer, Bass = 80 Hz

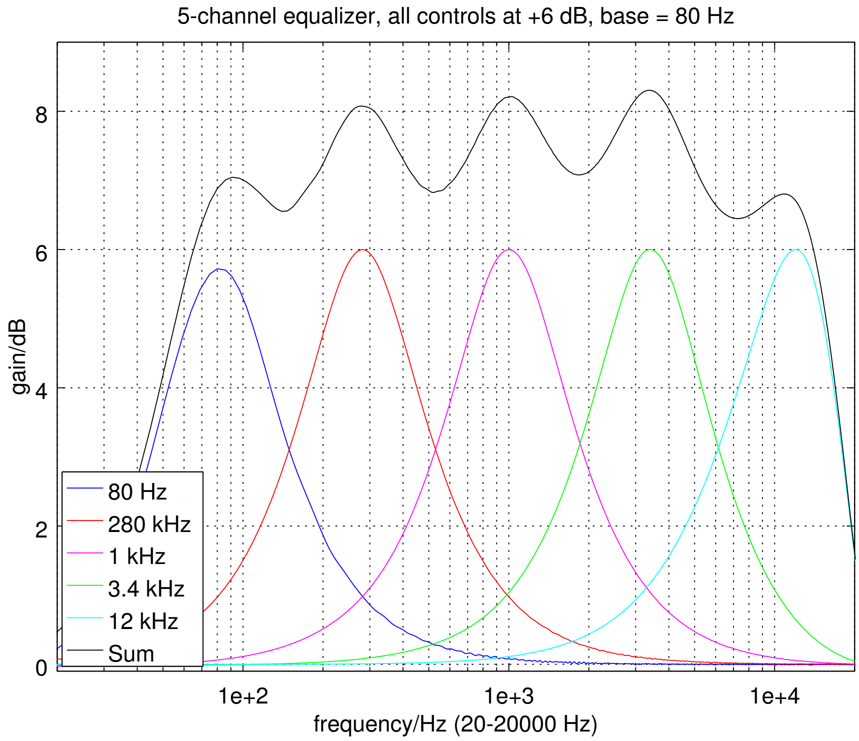

Figure 4: 5-Band Equalizer with 80 Hz bass control.

Filter Number	Frequency/Hz	Q Factor
0	80	1.0
1	280	1.0
2	1000	1.0
3	3400	1.0
4	12000	0.7

Figure 4 shows the frequency responses for the individual bands of a 5-band equalizer when all of the bands are set to +6 dB, as well as their total. This filter set is useful for a medium size speaker.

The processing power requirement for this filter at 48 kHz, in stereo, is approximately 24.7 MIPS.

7.5 5-Band Equalizer, Bass = 100 Hz

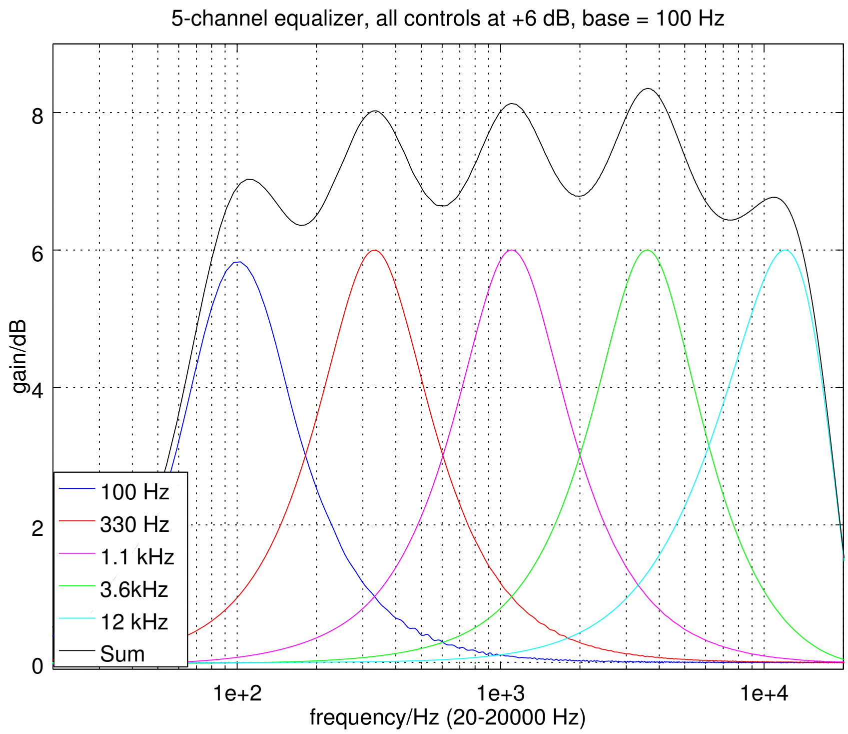

Figure 5: 5-Band Equalizer with 100 Hz bass control.

Filter Number	Frequency/Hz	Q Factor
0	100	1.1
1	330	1.1
2	1100	1.1
3	3600	1.1
4	12000	0.7

Figure 5 shows the frequency responses for the individual bands of a 5-band equalizer when all of the bands are set to +6 dB, as well as their total. This filter set is useful for a small speaker.

The processing power requirement for this filter at 48 kHz, in stereo, is approximately 24.7 MIPS.

7.6 7-Band Equalizer, Bass = 60 Hz

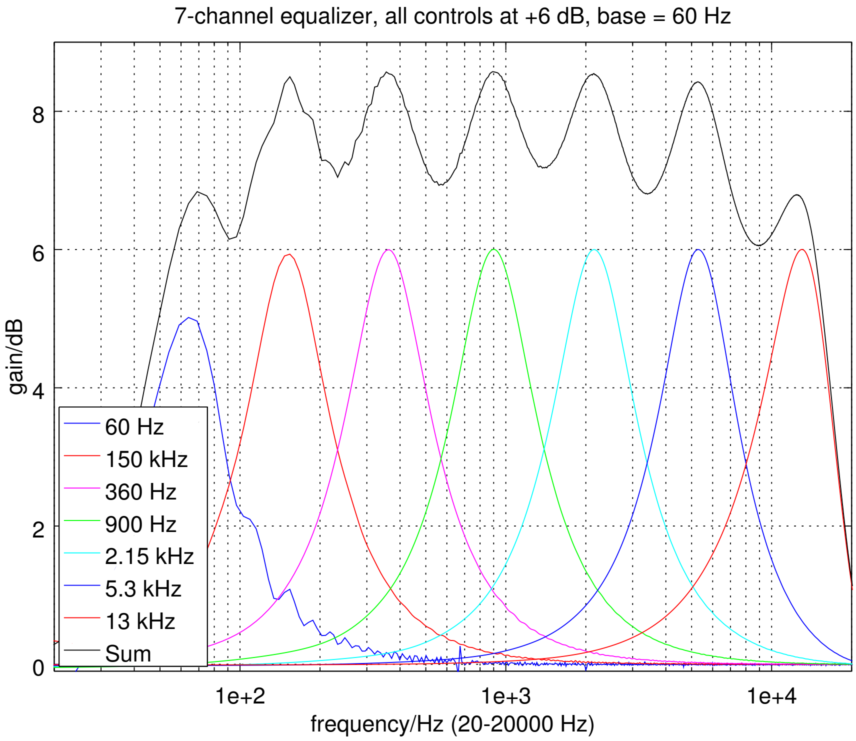

Figure 6: 7-Band Equalizer with 60 Hz bass control.

Filter Number	Frequency/Hz	Q Factor
0	60	1.5
1	150	1.5
2	360	1.5
3	900	1.5
4	2150	1.5
5	5300	1.5
6	13000	1.0

Figure 6 shows the frequency responses for the individual bands of a 7-band equalizer when all of the bands are set to +6 dB, as well as their total. This filter set is useful for a large speaker.

The processing power requirement for this filter at 48 kHz, in stereo, is approximately 33.2 MIPS.

7.7 7-Band Equalizer, Bass = 80 Hz

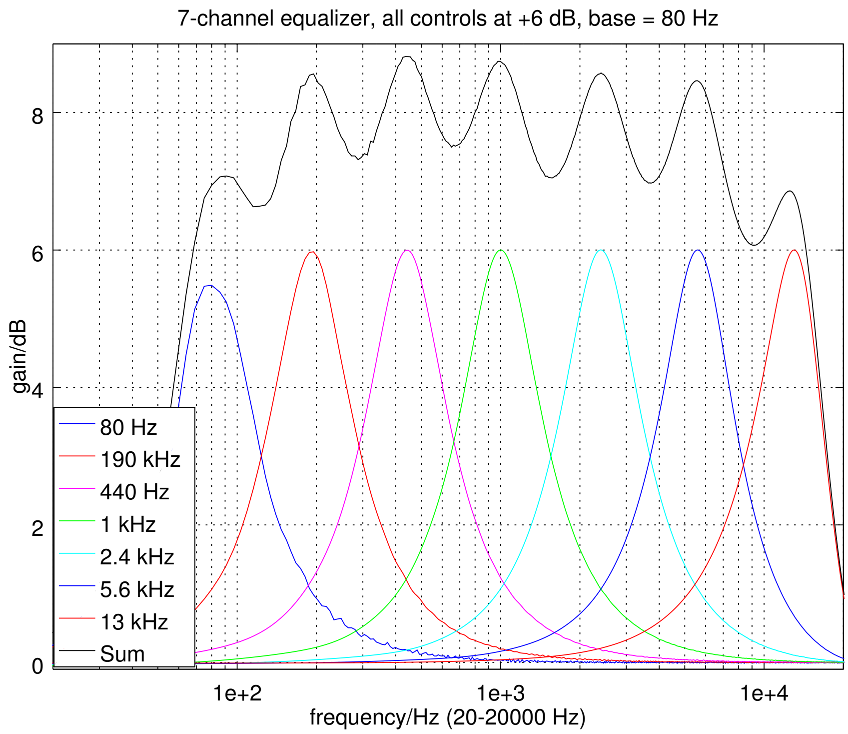

Figure 7: 7-Band Equalizer with 80 Hz bass control.

Filter Number	Frequency/Hz	Q Factor
0	80	1.55
1	190	1.55
2	440	1.55
3	1000	1.55
4	2400	1.55
5	5600	1.55
6	13000	1.10

Figure 7 shows the frequency responses for the individual bands of a 7-band equalizer when all of the bands are set to +6 dB, as well as their total. This filter set is useful for a medium size speaker.

The processing power requirement for this filter at 48 kHz, in stereo, is approximately 33.2 MIPS.

7.8 7-Band Equalizer, Bass = 100 Hz

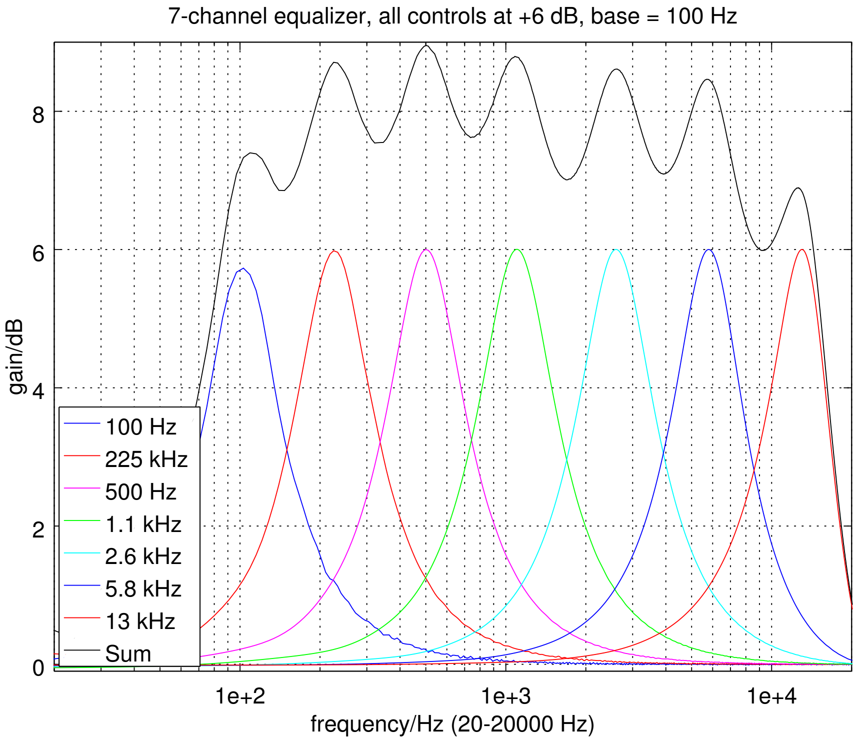

Figure 8: 7-Band Equalizer with 100 Hz bass control.

Filter Number	Frequency/Hz	Q Factor
0	100	1.6
1	225	1.6
2	500	1.6
3	1100	1.6
4	2600	1.6
5	5800	1.6
6	13000	1.2

Figure 8 shows the frequency responses for the individual bands of a 7-band equalizer when all of the bands are set to +6 dB, as well as their total. This filter set is useful for a small speaker.

The processing power requirement for this filter at 48 kHz, in stereo, is approximately 33.2 MIPS.

8 Contact Information

VLSI Solution Oy
Entrance G, 2nd floor
Hermiankatu 8
FI-33720 Tampere
FINLAND

URL: http://www.vlsi.fi/
Phone: +358-50-462-3200
Commercial e-mail: sales@vlsi.fi

For technical support or suggestions regarding this document, please participate at
http://www.vsdsp-forum.com/
For confidential technical discussions, contact
support@vlsi.fi