Politecnico di Torino

1859

Signal Processing and Wireless Transmission Lab DSP Module

Pro. Andrea Migliorati

Group 4:
Zhang Ziteng (289309)
Kang Yali (287831)
Tong Lin (287649)
Cen Jialuo (287781)
Hassan Abeyat (305435)

May 14, 2023

1 Lab 1 Audio filtering

1.1 Filtering Structures

In this task, we implemented different FIR and IIR filters by using floating-point representation. According to the given references and materials, we created our project and modified the code.

- As shown in Table 1, $output1f$ is for the linear FIR and $output2f$ is for the circular FIR. We verified that the linear buffer and circular buffer versions have the same output.

Expression	Type	Value
err1	float	58360452.0
err2	float	0
output1f	float[256]	[1671.54565,3008.16797,...]
output2f	float[256]	[1671.54565,3008.16797,...]

Table 1: The results for outputs and errors

- The results of CPU clock cycles for both versions and different optimization levels are below:

Optimization	Disable	0	1	2	3
Linear FIR	6,404,195.72	6,187,784.78	6,247,599.00	5,751,219.00	5,751,219.00
Circular FIR	6,012,996.94	5,901,036.00	5,965,194.72	5,733,415.84	5,733,415.84

Table 2: The floating-point FIR CPU clock cycles

With the increment of the optimization level, the number of clock cycles decreases. Compared with the linear FIR, the circular one runs faster. This results from that the circular filter does not access the memory as much as the linear filter. The two different results tend to be similar in the end. When the optimization level is greater than or equal to 2, the number of CPU clock cycles does not change any more.

- As shown in Table 3, the terms $output1f$ and $output2f$, respectively represent the output value of IIR biquad I and II, which only have very slight differences regarding the different structures.

Expression	Type	Value
err1	float	0
err2	float	5.18073264e-08
output1f	float[256]	[453.172485,118.953827,-158.411102...]
output2f	float[256]	[453.172699,118.953842,-158.411194...]

Table 3: The results for outputs and errors

- After we implemented IIR direct form I and II versions, the results are shown in the table below. The IIR biquad II runs always faster than the direct I. The number of the clock cycle becomes gradually similar as the optimization level increases. Compared with the disabled one, the difference at level 3 stays only about 10,000 rather than 20,000. When the optimization level is greater than or equal to 2, the number of CPU clock cycles does not change.

Optimization	Disable	0	1	2	3
IIR biquad I	601,660.47	596,581.38	582,864.94	576,331.94	576,331.94
IIR biquad II	584,590.81	583,950.59	565,815.06	560,793.72	560,793.72

Table 4: The floating-point IIR CPU clock cycles

1.2 Fixed-point implementation

In this task, we post the previous filter structures from floating-point to fixed-point arithmetic and compare floating-point and fixed-point versions in terms of complexity and precision.

- As shown in Table 5, the mean squared error (MSE) between floating-point output and fixed-point output for IIR and FIR are computed with the given coefficients. The MSE of IIR-biquad1 is very small at about 2.94, with the coefficients of af and bf. Whereas the error from IIR-biquad2 is quite large and in order of 10^9, because of the diverged output for this kind of structure. The MSEs of linear and circular FIR are the same and very small.

- When using the coefficient afq and bfq, all those previously described MSEs are reduced, except for the term IIR2(even with different coefficient, the value is also quit huge). The last two MSEs of the FIR ones are the same and become smaller than before.

Coefficient	IIR 1	IIR 2	FIR Linear	FIR Circular
af,bf	2.94634032	358114112	0.350714147	0.350714147
afq,bfq	0.808875144	358113504	0.0837360248	0.0837360248

Table 5: MSE without noise shaping for different filters

- The CPU clock cycles of fixed-point versions for the circular FIR and the linear FIR are measured in this part. As shown in Table 6, the number of the clock cycle becomes gradually similar as the optimization level increases.

  All these CPU clock cycles of the fixed-point versions are smaller than the floating-point ones, according to Table 2, whose difference is more than 1,000,000. The optimization effect is more pronounced than floating-point ones. When the optimization level is greater than or equal to 2, the number of CPU clock cycles does not change anymore.

Optimization	Disable	0	1	2	3
Linear FIR	538,130.00	329,488.00	354,573.00	3,3047.00	3,3047.00
Circular FIR	311,875.19	186,182.03	176,448.16	25,533.41	25,533.41

Table 6: The fixed-point FIR CPU clock cycles

- For the type of IIR (direct form I & II), the measured CPU clock cycles for the fix-point versions are shown in Table 7. The number of the clock cycle becomes gradually similar as the optimization level increases.

  All these CPU clock cycles of the fixed-point versions are smaller than the floating-point ones, according to Table 4, whose difference is more than 500,000. The optimization effect is more pronounced for the situation of fixed-point than floating-point ones. When the optimization level is greater than or equal to 2, the number of CPU clock cycles does not change.

Optimization	Disable	0	1	2	3
IIR biquad I	45,587.00	38,679.00	25,624.00	8,217.00	8,217.00
IIR biquad II	43,283.00	36,115.00	26,900.00	12,821.00	12,821.00

Table 7: The fixed-point IIR CPU clock cycles

Coeffieient	af,bf	afq,bfq
IIR biquad I	2.25959	0.08313

Table 8: MSE with noise shaping for IIR biquad I

- Here, we implement noise shaping (NS) for the IIR direct form I with c1 = a1 and c2 = a2.

  From Table 8, we found that the accuracy has improved with noise shaping under the afq and bfq coefficients.

Optimization	Disable	0	1	2	3
IIR biquad I	58,447.00	51,739.00	32,792.00	12,310.00	12,310.00

Table 9: The IIR biquad I CPU clock cycles with NS

Compared the Table 4 and Table 9(for direct form I), the *CPU clock cycles* decreases, which means the noise shaping block leads to a trade-off about performance.

1.3 Real-time filtering

- Verification of the real-time operations:

  In this part, we use different optimization levels (Disable and 1), FIR implementations (linear and circular), IIR implementations (biquad I and biquad II) and sampling frequencies to verify the function of the system.

	Fir_linear		Fir_circular		IIR-1		IIR-2	
Frequency(kHz)	16	48	16	48	16	48	16	48
Disable	10400	10400	20000	18000	32000	54272	32000	30720
1	19456	19400	32000	34560	32000	94976	32000	46592

Table 10: Samples per second

  Through analysis, we conclude that the value of the sample per second should be twice the frequency value, as there are two channels. We can notice that circular FIR can run correctly in level 1 optimization for a sampling frequency of 16kHz. Both the linear and circular FIR are required to operate with higher optimization to meet the requirement.
  We can find that the sample per second of IIR-1 is larger than IIR-2 at the same optimization level. Both of them can be used in 16kHz. And if we want to use it in 48kHz, we also need a deeper optimization, and IIR biquad I is easier to meet the requirements.

- Filter analysis:

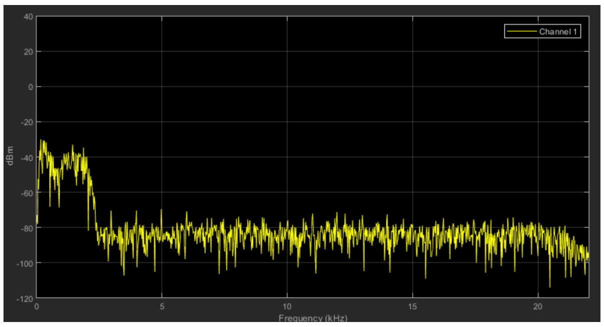

Figure 1: The FIR filter at 48MHz

  As shown in Figure.1, the FIR filter can be seen as a low-pass filter.

- The spectrum of IIR (biquad I and biquad II) at 16kHz and 48kHz:

  For the IIR filter, the spectrum of biquad I and biquad II are quite similar. Low-frequency signals can pass through the system, as in Figure 2 and Figure 3, and when the frequency of the signal is higher than the cut-off frequency of the filter, the higher part of the signal is filtered, as shown in Figure 4 and Figure 5.

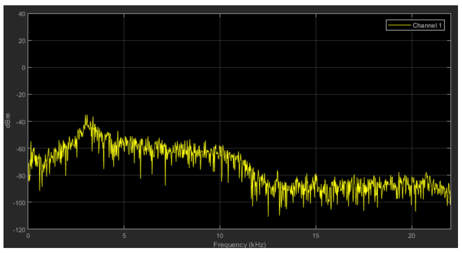

Figure 2: IIR (biquad I) at 16kHz

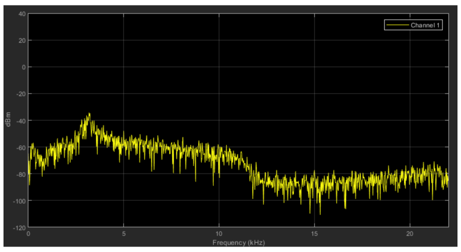

Figure 3: IIR (biquad II) at 16kHz

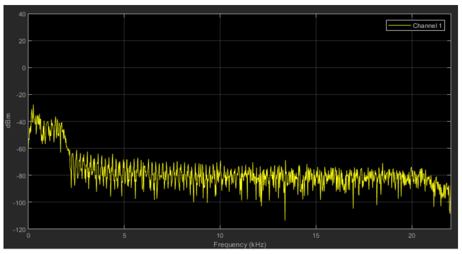

Figure 4: IIR (biquad I) at 48kHz

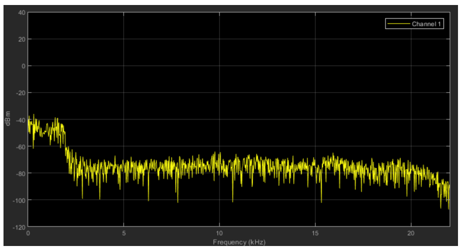

Figure 5: IIR (biquad II) at 48KHz

- Delay subtraction:

  For the FIR filter, using a delay equal to $\frac{FIR\_LEN-1}{2}$ and it is a complementary high-pass filter. But for the IIR filter, as we can see it is not a true high-pass filter.

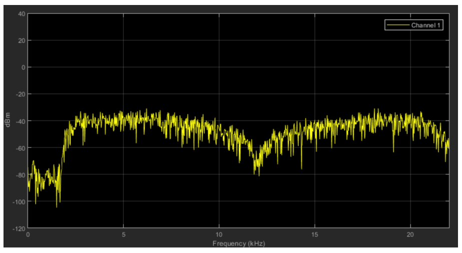

Figure 6: Fir delay subtraction

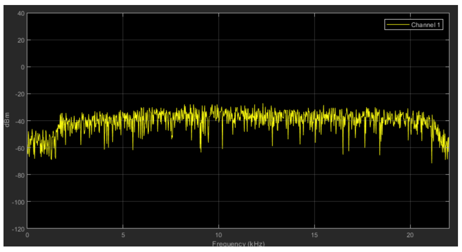

Figure 7: IIR delay subtraction

It is because that the characteristic of this filter is different from the FIR filter, and the IIR is not linear, which is based on recursion due to feedback mechanisms. Therefore, it cannot be seen as a high-pass filter with the delay.

2 Lab 2 audio compression

2.1 Testing the Audio Codec

In this task, we implement prediction and entropy coding functions and test them by using a reference of the data set and a reference of the implementation of the decoder.

- After completing the implementation and verifying the correctness of the function, we measure the number of output bits when we encode the referenced 8192 samples for different choices of the predictor, with different values of k for the Rice-Golomb code (5, 7, 9), as shown in Table 11,

	k=5	k=7	k=9
no predictor	-	289894	141110
polynomial 1	165737	97869	93526
polynomial 2	111755	85564	90751
5^{th}-order predictor	93204	79941	90230

Table 11: The number of output bits

it is obvious that the value of the output sample without predictors is very large, while the values of the output sample with predictors are significantly less. For the term of no predictor, the value for $k = 9$ is larger than $k = 7$. However, the output bits do not keep decreasing with the increase of k, as in the situation of $polynomial1$. According to the data of $polynomial2$ and 5^{th}-order predictor, the number of output bits begins to increase when reaching $k = 7$. Moreover, the value of 5^{th}-order predictor is the smallest when $k = 7$.

- Then we measure the number of CPU clock cycles per encoded sample for different versions of the optimization level, respectively using no optimization level and optimization level 2, as shown in Table 12.

	k=5		k=7		k=9	
Optimization level	No	2	No	2	No	2
No predictor	3676	2295	1156	680	532	274
Polynomial 1	769	418	456	213	384	165
Polynomial 2	650	330	493	226	460	203
5^{th}-order predictor	807	387	703	317	684	304

Table 12: CPU clock cycles/8192 samples

By analyzing the results, we can conclude that, when $k = 9$ and the optimization level is 2, the highest efficiency can be found in the situation of *polynomial* 1.

2.2 Real-time audio coding

- The bit rate of the encoder for different configurations:

	Disable	1	2	3
12 KHz	201325	201325	201325	201325
16 KHz	268058	268058	268058	268058
24 KHz	303249	376257	376257	376257
48 KHz	303249	485312	688362	688362

Table 13: The bit rate of the encoder for different configurations

At low frequencies, the system can operate correctly in real-time without optimization. However, optimization is necessary when reaching 48 kHz.
Overall, the bit rate increases with the increment of the sampling frequency, while the optimization levels seldom affect the results.

- The average bit rate of the encoder

	No predictor	Polynomial 1	Polynomial 2	5^{th}-order predictor
k=5	2818272	1015030	950640	748300
k=7	1628396	988915	908950	687358
k=9	1150481	994166	991812	810278

Table 14: Bit Rate for different predictors

The uncompressed Bit Rate is given by:

$$Uncompressed\, Bit\, Rate = 44.1MHz \times 16bits/samples \times 2channels = 1411200bits/s \tag{1}$$

The best compression ratio is at $k = 7$, $5^{th} - order\, predictor$:

$$The\, best\, compression\, ratio = \frac{Uncompressed\, Bit\, Rate}{BitRate} = 2.05 \tag{2}$$

- The changes in bit rate is corresponding with different parts (soft, loud) of the song.

	Quiet	Energy
Loud	835000	1130000
Soft	783000	1050000

Table 15: Bit rate (approximate integer) of the song

We found that the bit rate increases with the increment of the value of volume, and it is higher in more dynamic parts of the song.