ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

Companding Techniques using SysGen

Sheetal Patil [1], Satheesh Rao[2], D Saravanan[3]

M.Tech Student, Dept. of Electronics and Communication Engineering, NMAMIT, Nitte, Karnataka, India [1]

Assistant Professor, Dept. of Electronics and Communication Engineering, NMAMIT, Nitte, Karnataka, India[2]

Scientist 'F', Centre for Air Borne Systems-DRDO, Bengaluru, Karnataka, India[3]

ABSTRACT: Although humans are well equipped for analog communications, analog transmission is not particularly efficient. Digital signals, having only "one-bit" and "zero-bit" states, are more easily separated from noise. It can be amplified without corruption but are more prone to noise and data loss on long distance communications. The world's communication systems have converted to a digital transmission format called pulse code modulation (PCM). PCM is called "waveform" coding because it creates a coded form of the original voice waveform. It is defined in the ITU-T G.711 specification. Since most voice signals generated have noise, to improve voice quality companding is used.

KEYWORDS: Companding, PCM, ITU-T, SysGen.

I. INTRODUCTION

Companding refers to the process of first compressing an analog signal at the source, and then expanding this signal back to its original size when it reaches its destination. The term companding is created by combining the two terms, compressing and expanding, into one word. At the time of the companding process, input analog signal samples are compressed into logarithmic segments. Each segment is then quantized and coded using uniform quantization. The compression process is logarithmic. The compression increases as the sample signals increase. By companding many advantages can be achieved, such as improved noise treatment. In order to prevent the signal from distortions caused by channel these techniques are used. The ITU-T standards for companding are called A-Law and μ-Law.

A-Law and μ-Law Companding

A-Law and μ-Law are audio compression schemes (codecs) defined by Consultative Committee for International Telephony and Telegraphy (CCITT) G.711 which compress 16-bit linear PCM data down to eight bits of logarithmic data. They convert 16 bit digital audio signal into 8 bits by a logarithmic formula.

A-Law Companding

Across Europe the CCITT suggested A-Law as standard companding technique. The linear sample values are restricted to 12 magnitude bits and the A-Law compression can be more precisely explained by the equation (1), the parameter used for compression is A i.e A=87.7 in Europe and y is the normalized integer to be compressed.

$$G(x)=\begin{cases} \frac{(A*|z|)}{(1+\ln(A))}, & 0 \le |z| \le \left(\frac{1}{A}\right) \\ \frac{(1+\ln(A|z|)*(\mathrm{sgn}(z))}{(1+\ln(A))}, & \left(\frac{1}{A}\right) \le |z| \le 1 \end{cases} \quad (1)$$

After encoding the given input data, an inversion pattern is applied to the obtained result which is of 8-bit that increases the density of transitions which is a benefit to hardware performance. The decoding is done by xoring the 8-bit code with 0x55.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

μ-Law Companding

The United States and Japan use μ-Law companding. The linear sample values are restricted to 13 magnitude bits and the μ-Law compression can more precisely be explained by equation (2), the parameter used for compression is μ i.e μ=255 and z is the normalized integer to be compressed.

$$F(x) = \left\{ \frac{(\mathbf{sgn}(z) * (1 + \ln(\mu|z|)))}{(1 + \ln(\mu))} , \quad 0 \leq |z| \leq 1 \right\} \tag{2}$$

After encoding the input data, an inversion pattern is applied to the obtained result which is of 8-bit to increase the density of transitions which is a benefit to hardware performance and the decoded output is obtained. The decoding is done by xoring the 8-bit code with 0xFF.

II. LITERATURE SURVEY

Charles.W.Brokish made a report of companding techniques which are A-Law and μ-Law and explained the same in brief [1]. Arazi et al., proposed an efficient compander design which converts 20 bit data to 8 bit with 1000 gates using Xilinx Virtex –II Pro FPGA device [2]. A.klein et al., explains how to instantaneously compand the signals using Matlab/Simulink [3].

III. METHODOLOGY

The audio signal is given as input to the Encoder and decoded back through Decoder. The audio input given is "M1F1-int16-AFsp.wav". It has samples of around 11234 when it was read using audioread command. First for the reference Matlab codes are written for both algorithms. Then using the reference code System Generator blocks where implemented. The System Generator has many inbuilt logical blocks and there are some blocks which cannot be implemented by it directly. So those codes are written in Matlab as function and made as Mcodes.The A-Law and μ-Law use System Generator to implement the Matlab code in blocks.

IV. EXPERIMENTAL SETUP

A-Law implementation using System Generator

The A-Law involves an Encoder and a Decoder. The samples of the audio file, "M1F1-int16-AFsp.wav" is sent to Encoder and the output is obtained at the Decoder.

A-Law Encoder

The A-Law Encoder System Generator block can be given in Figure 1. It consists of many logical blocks and Mcodes. The audio file samples are given as input to Encoder.

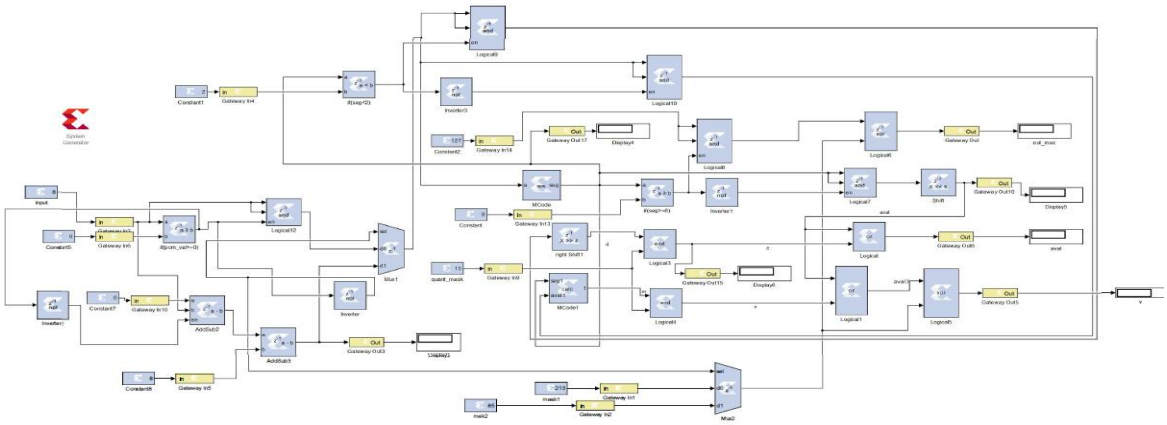

Figure 1: A-Law Encoder implementation using System Generator

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

A-Law Decoder

The A-Law Decoder implementation is shown in Figure.2. The Encoder output is fed as input to Decoder block.

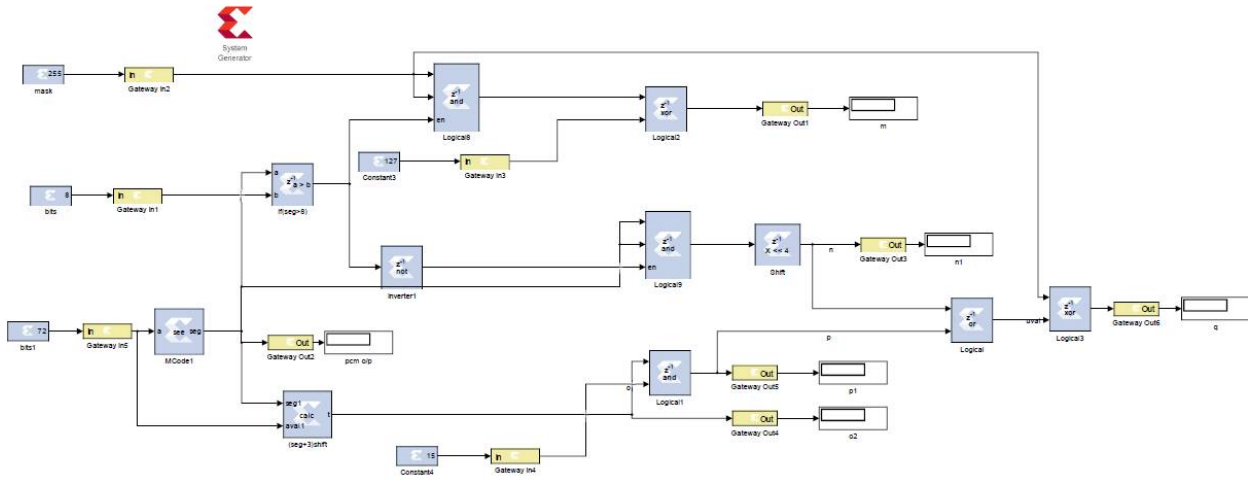

Figure 2: A-Law Decoder implementation using System Generator.

μ-Law Encoder and Decoder

The μ-Law also involves Encoder and Decoder whose blocks are implemented using System Generator.

μ-Law Encoder

The μ-Law Encoder's System Generator block is implemented as shown in Figure 3. The audio file samples are given as input Encoder.

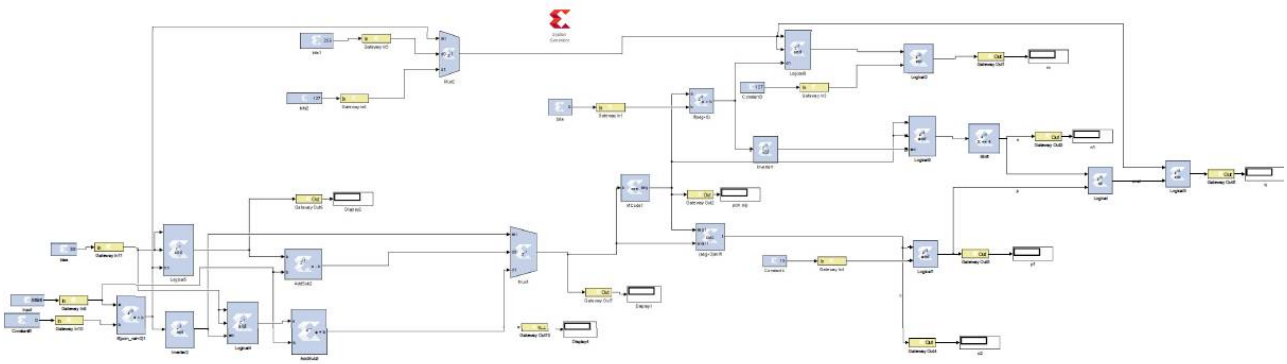

Figure 3: μ-Law Encoder implementation using System generator

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

μ-Law Decoder

The μ-Law Decoder implementation is given below in Figure 4. The Encoder output is fed as input to Decoder block.

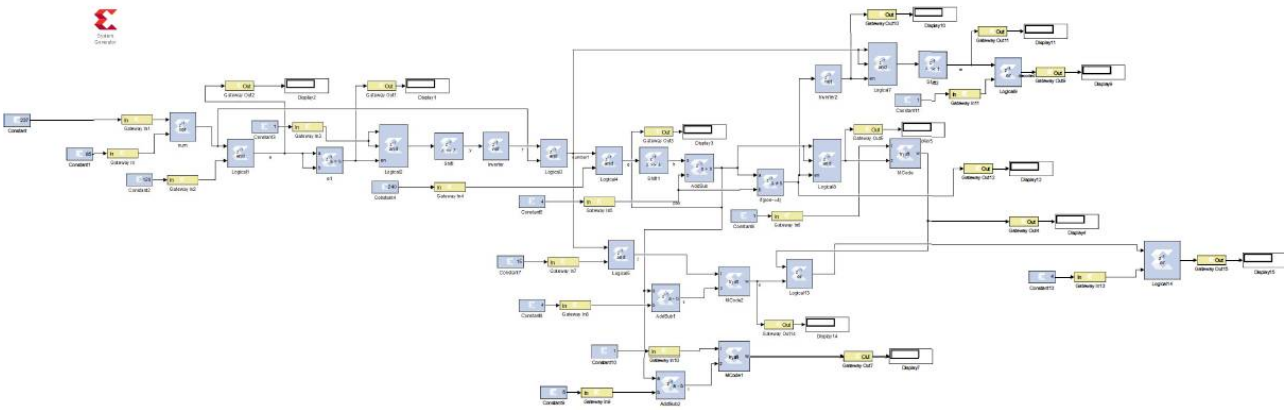

Figure 4: μ-Law Decoder implementation using System Generator.

IV. RESULT AND DISCUSSION

The results consist of A-Law and μ-Law implementation using System Generator in the Simulink.

A-Law implementation using System Generator

The Simulink has Available System Generator block of G.711 A-Law and μ-Law to which the 16 bit audio file samples are given and the output is noted. The Available System Generator block of G.711 cannot be converted into VHDL code. So these algorithms (A-Law and μ-Law) are implemented using the System Generator blocks which include Mcode and logical blocks that help in implementation of algorithms easily. The inbuilt block is shown in Figure 5.

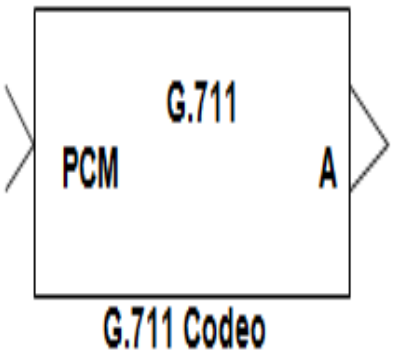

Figure 5: Available block in System Generator

A-Law Encoder Output

Some of the samples obtained from audio file are given as input to Encoder. In Table 1 the output of Available block in System Generator and implemented A-Law Outputs are compared.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

A-Law Decoder Output

The encoder output is given as input to the decoder. In Table 2 the output of Available block in System Generator and implemented A-Law Outputs are compared.

Table 1 A-Law Encoder Output

Serial No.	Encoder Input	Output of Available block in System Generator	Implemented A-Law Encoder Output
1	705	131	135
2	1230	182	183
3	-369	18	23
4	-1310	49	55
5	-356	19	23
6	657	134	135
7	371	146	151
8	-343	16	23
9	72	247	247
10	1551	189	182
11	2146	165	167
12	2225	164	167
13	1955	187	182
14	-13	83	69
15	-1754	62	54
16	-1544	61	54
17	-124	123	118
18	579	135	135
19	-276	20	23
20	-1407	48	55
21	-1997	58	54
22	-2129	37	39
23	-995	10	6
24	855	143	134
25	2153	165	167
26	1230	182	183
27	1290	177	183
28	860	143	134
29	995	138	134
30	-790	13	6
31	-948	8	6
32	-288	20	23
33	538	133	135
34	5435	170	170

Table 2 A-Law Decoder Output

Serial No.	Decoder Input	Output of Available block in System Generator	Implemented A-Law Decoder Output
1	135	592	634
2	183	1184	1119
3	23	-296	-319
4	55	-1184	-1218
5	23	-296	-319
6	135	592	634
7	151	296	326
8	23	-296	-319
9	247	74	72
10	182	1248	1331
11	167	2368	2291
12	167	2368	2291
13	182	1248	1331
14	69	-33	-19
15	54	-1248	-1282
16	54	-1248	-1282
17	118	-78	-111
18	135	592	634
19	23	-296	-319
20	55	-1184	-1218
21	54	-1248	-1282
22	39	-2368	-2460
23	6	-624	-529
24	134	624	579
25	167	2368	2291
26	183	1184	1119
27	183	1184	1119
28	134	624	579
29	134	624	579
30	6	-624	-529
31	6	-624	-529
32	23	-296	-319
33	135	592	634
34	170	4032	5436

μ-Law implementation using System Generator

μ-Law Encoder Output

The audio samples which are obtained is given as input to Encoder. In Table 3 the output of Available block in System Generator and implemented μ -Law Outputs are compared.

μ- Law Decoder Output

The Encoder output is given as input to the Decoder. In the Table 4.4, the Available System Generator block and System Generator outputs are compared. The Simulink has an Available System Generator block of G.711 A-Law and μ-Law to which the obtained Encoder result of 8 bit samples are given and the output is noted. The System Generator blocks include Mcode and logical blocks which help in implementation of algorithms easily. In Table 4, the output of Available block in System Generator and implemented μ-Law Outputs are compared.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

Table 3 µ-Law Encoder Output

Serial No.	Encoder Input	Output of Available block in System Generator	Implemented µ-Law Encoder Output
1	705	184	173
2	1230	172	157
3	-369	70	60
4	-1310	43	29
5	-356	71	60
6	637	187	173
7	371	198	188
8	-343	72	61
9	72	229	220
10	1551	167	156
11	2146	158	141
12	2225	158	141
13	1955	160	156
14	-13	120	109
15	-1754	36	28
16	-1544	39	28
17	-124	92	77
18	579	188	173
19	-276	76	61
20	-1407	41	29
21	-1997	32	28
22	-2129	31	13
23	-995	47	29
24	855	180	172
25	2153	158	141
26	1230	172	157
27	1290	171	157
28	860	180	172
29	995	175	157
30	-790	54	44
31	-948	49	44
32	-288	75	61
33	538	190	173
34	5435	138	125

Table 4 µ-Law Decoder Output

Serial No.	Decoder Input	Output of Available block in System Generator	Implemented µ-Law Decoder Output
1	173	655	639
2	157	1343	1311
3	60	-327	-352
4	29	-1343	-1311
5	60	-327	-319
6	173	655	639
7	188	327	319
8	61	-311	-336
9	220	57	59
10	156	1407	1375
11	141	2719	2655
12	141	2719	2655
13	156	1407	1375
14	109	-10	-9
15	28	-1407	-1375
16	28	-1407	-1375
17	77	-139	-135
18	173	655	639
19	61	-311	-366
20	29	-1343	-1311
21	28	-1343	-1311
22	13	-2719	-2655
23	29	-1343	-1311
24	172	687	671
25	141	2719	2655
26	157	1343	1311
27	157	1343	1311
28	172	687	671
29	157	1343	1311
30	44	-687	-671
31	44	-687	-671
32	61	-311	-303
33	173	655	639
34	125	5471	5343

V. CONCLUSION

The audio companding helps in increasing the Signal to Noise Ratio (SNR) by suppressing the noisy data. The outputs of the Available System Generator block and the implemented System Generator are compared and are almost same. The Encoder and Decoder table depict that the signal sent as input to Encoder and the output derived from the Decoder are approximately equal. The audio companding is employed as a basic building block in the hardware module of ADMS.

REFERENCES

[1] Charles.W.Brokish, "A law and µ law Companding using the TMS320C54X", Application Report SPRA163a, Texas Instruments, 1997.
[2] Arazi, Elhanany, "A scalable architecture for high speed digital companding", 48th IEEE International Midwest Symposium on Circuits and Systems, Vol.1, pp. 488 – 490, 2005.
[3] A. Klein and Y Tsividis, "Instantaneously companding digital signal processors", Proceding of IEEC ICASSP, Vol 3, pp.1433-1436, 2007.