

## Procesul de codare pentru legea $\mu$

Așa cum se vede în Figura 1, semnalul vocal se aplica la intrarea unui convertor analog-numeric linear pe 14 biți, la ieșirea căruia se obțin codurile, pe 14 biți, ale eșantioanelor semnalului de intrare. Compresorul digital alege, conform caracteristicii de compresie, codul de 8 biți, care va fi transmis pe linie, corespunzător codului de 14 biți.

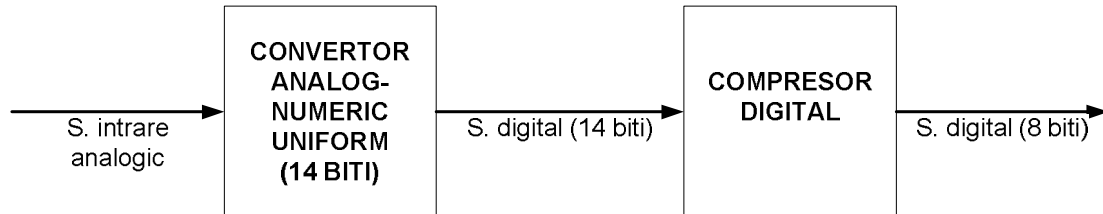


Figura 1 - Procesul de codare pentru legea  $\mu$

Putem calcula gama dinamică a nivelului semnalului de intrare pe baza notei 1 (anexa 1), care ne indică puterea maximă,  $T_{max}=3.17dBm0$ . Deoarece în calculul puterii se folosește valoarea efectivă a semnalului, vom calcula gama dinamică pentru un semnal sinusoidal. Folosind această valoare vom calcula nivelul maxim acceptat al unui semnal sinusoidal, considerând valoarea impedanței  $|Z|=600\Omega$  :

$$10 \cdot \lg\left(\frac{V_{max}^2}{2 \cdot 600 \cdot 10^{-3}}\right) = 3.17 \Rightarrow V_{max} = \sqrt{1.2 \cdot 10^{0.317}} = 1.57794V \quad (1)$$

Nivelele semnalului vocal nu au o distribuție uniformă, probabilitatea de apariție a nivelelor mici fiind mult mai mare, și deci probabilitatea ca un cod să conțină biți de zero este mare. Codul de linie folosit în cazul legii  $\mu$  este B8ZS (Bipolar with 8 Zeros Substitution), care este o variantă a codului AMI (Alternate Mark Inversion) care înlocuiește secvențele de 8 biți de zero consecutivi cu o violare a regulii de codare. Deoarece codul AMI codează biții de zero cu nivele de zero, secvențele lungi de zero afectează sincronizarea. Deși codul B8ZS înlocuiește secvențele de 8 biți de zero consecutivi rămâne posibilitatea de apariție a secvențelor de 7 biți de zero consecutivi. Din această cauză și ținând cont de faptul că probabilitatea de apariție a codurilor care conțin mulți biți de zero este mai mare, la ieșirea compresorului digital cei 8 biți obținuți sunt inversați.

Se poate observa (anexa 1) că în urma inversării biților codul obținut pentru primul subsegment pozitiv (semn:0, seg:000, subseg:0000) este 1111111b (FFh), cod care este identic cu cel de alarmă albastră; din această cauză acest cod nu se transmite, ci se transmite codul primului sub-segment negativ (semn:1, seg:000, subseg:0000) al cărui cod obținut după inversare este 0111111b (7Fh). Acesta din urmă este codul transmis când linia nu este utilizată. (Codurile din coloana 6 a tabelului din anexa 1 sunt cele obținute după inversare)

Pentru a simplifica implementarea compresorului digital se adaugă valoarea 33 la valoarea semnalului de intrare. Această operație are rolul de a face ca capetele segmentelor să fie puteri al lui 2 (vezi coloana 3, anexa 1:  $31+33=64$ ,  $95+33=128$ ,  $223+33=256, \dots$ ). În urma acestei operații, se poate observa în tabelul de codare (Tabel 1)

că decizia asupra segmentului în care se găsește eșantionul al cărui cod se aplică la intrarea compresorului digital se poate lua pe baza poziției primului bit de 1 (de la MSB spre LSB), după ce a fost extras bitul de semn.

Intrarea compresorului digital (după adăugarea valorii 33)													Cod comprimat								
													Segment			Subsegment					
bit:	12	11	10	9	8	7	6	5	4	3	2	1	0	bit:	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	1	a	b	c	d	x		0	0	0	a	b	c	d
	0	0	0	0	0	0	1	a	b	c	d	x	x		0	0	1	a	b	c	d
	0	0	0	0	0	1	a	b	c	d	x	x	x		0	1	0	a	b	c	d
	0	0	0	0	1	a	b	c	d	x	x	x	x		0	1	1	a	b	c	d
	0	0	0	1	a	b	c	d	x	x	x	x	x		1	0	0	a	b	c	d
	0	0	1	a	b	c	d	x	x	x	x	x	x		1	0	1	a	b	c	d
	0	1	a	b	c	d	x	x	x	x	x	x	x		1	1	0	a	b	c	d
	1	a	b	c	d	x	x	x	x	x	x	x	x		1	1	1	a	b	c	d

Tabel 1 – Tabelul de codare pentru legea  $\mu$

Observăm în Tabelul 1 cum poziția primului bit de 1 ne indică codul segmentului:

$COD\ SEGMENT_{10} = 7 - nr.\ zerouri\ din\ fața\ primului\ bit\ de\ 1 = -5 + poziția\ primului\ bit\ de\ 1$  (2), iar codul subsegmentului îl reprezintă următorii 4 biți, după primul bit de 1.

De exemplu dacă la intrarea compresorului, după adăugarea valorii 33, avem codul 10000111011100 putem folosi relația (2) ca să determinăm codul segmentului:

$$COD\ SEGMENT_{10} = 7 - 4 = 3 \Rightarrow COD\ SEGMENT_2 = 011$$

iar codul sub-segmentului sunt chiar următorii 4 biți :

$$COD\ SUBSEGMENT_2 = 1101$$

Bitul de semn rămâne nemodificat:

$$COD\ SEMN_2 = 1$$

Astfel codul obținut este 10111101<sub>2</sub>. Acesta se inversează și se transmite pe linie. La destinație codul preluat de pe linie este din nou inversat și se obține codul 10111101<sub>2</sub>. Pe baza codului segmentului determinăm poziția primului bit de 1, și obținem 00001xxxxxxx, după care completăm următorii 4 biți cu codul sub-segmentului, 000011101xxxx. Au mai rămas de completat ultimii 4 biți, despre care nu avem nici un fel de informație. Astfel pentru a reduce la jumătate eroarea de cuantizare maximă se alege valoarea din mijlocul domeniului care ar putea fi reprezentat (Figura 2) și obținem codul 0000111011000<sub>2</sub>, la care se adaugă bitul de semn și se obține 10000111011000<sub>2</sub>.

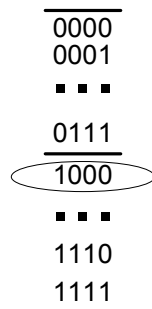


Figura 2 – Înlocuirea la recepție a biților eliminați

## Procesul de codare pentru legea A

În cazul legii A, se schimbă dimensiunea segmentelor, iar primul segment are 32 de subsegmente, ceea ce duce la un total de doar 7 segmente, atât pe partea pozitivă cât și pe cea negativă. Primul segment însă se poate împărți în două segmente, fiecare a câte 16 subsegmente și astfel structura codului rămâne aceeași ca la legea  $\mu$ . Cuantizarea uniformă se face doar pe 13 biți, iar adăugarea valorii 33 sau a oricărei alte valori nu mai este necesară deoarece capetele segmentelor sunt puteri ale lui 2 (coloana 3, anexa 2). Codarea se face conform tabelului de codare (Tabel 2), iar codului obținut îi sunt inversați biții de pe pozițiile pare. Codul de linie care se folosește este HDB3 (High Density Bipolar 3 Zeros), care este tot un cod AMI modificat, care înlocuiește secvențele de 4 biți de zero consecutivi cu violări ale reguli de codare.

Intrarea compresorului digital														Segment				Subsegment			
bit:	11	10	9	8	7	6	5	4	3	2	1	0	bit:	6	5	4	3	2	1	0	
	0	0	0	0	0	0	0	a	b	c	d	x		0	0	0	a	b	c	d	
	0	0	0	0	0	0	1	a	b	c	d	x		0	0	1	a	b	c	d	
	0	0	0	0	0	1	a	b	c	d	x	x		0	1	0	a	b	c	d	
	0	0	0	0	1	a	b	c	d	x	x	x		0	1	1	a	b	c	d	
	0	0	0	1	a	b	c	d	x	x	x	x		1	0	0	a	b	c	d	
	0	0	1	a	b	c	d	x	x	x	x	x		1	0	1	a	b	c	d	
	0	1	a	b	c	d	x	x	x	x	x	x		1	1	0	a	b	c	d	
	1	a	b	c	d	x	x	x	x	x	x	x		1	1	1	a	b	c	d	

Tabel 2 – Tabelul de codare pentru legea A

Se poate observa că diferența care apare în tabelul de codare, față de legea  $\mu$ , este la primele două segmente.

### Exemplu

În cazul legii A, pentru o gamă dinamică a semnalului de intrare între -1.5V și 1.5V, să se determine codul obținut la ieșirea compresorului digital pentru o tensiune de intrare  $V_{in} = -0.45779V$ .

### Codarea

Metoda 1 – folosind caracteristica de compresie :

- deoarece caracteristicile de compresie sunt date normalizat, primul pas ar fi să normalizăm tensiunea de intrare :

$$V_{in_{normalizat}} = \frac{-0.45779}{1.5} = -0.305193$$

- în continuare stabilim valoarea bitului de semn  $b_{semn} = 0$  și vom lucra în continuare cu valoarea în modul a tensiunii de intrare normalizate  $|V_{in_{normalizat}}| = 0.305193$

- următorul pas este determinarea segmentului în care cade această tensiune de intrare. Pentru aceasta ne uităm pe axa X a caracteristicii de compresie (legea A) și observăm că această valoare cade în segmentul 6, delimitat de valorile 0.25, 0.5. Astfel obținem codul segmentului :  $110_2$ .
- ultimul pas este de determinare a subsegmentului în care cade valoarea tensiunii de intrare. Pentru aceasta trebuie să determinăm dimensiunea unui subsegment al segmentului 6,  $q_6 = \frac{0.5 - 0.25}{16} = 0.015625$ . Folosind această valoare putem să determinăm subsegmentul în care se găsește tensiunea de intrare:
- $subsegment = \frac{0.305193 - 0.25}{0.015625} = 3.53 \Rightarrow cod\_subsegment = 0011_2$
- Codul obținut la ieșirea compresorului : **01100011**

### Metoda 2 – folosind tabelul de codare

- într-o primă fază vrem să determinăm codul de la intrarea compresorului. Pentru aceasta determinăm cuanta elementară:  $q_e = \frac{1}{2^{12}}$ , după care determinăm codul zecimal returnat de convertorul analog-numeric  $val = \frac{|V_{in\_normalizat}|}{q_e} = 0.305193 \cdot 2^{12} = 1250.070528 \Rightarrow val = 1250$ , valoare scrisă în binar  $010011100010_2$
- bitul de semn este  $b_{semn}=0$ , și folosind relația (2) sau din Tabelul 2 putem determina COD SEGMENT =  $110_2$ , iar COD SUBSEGMENT =  $0011_2$
- Codul obținut la ieșirea compresorului este același ca la metoda 1: **01100011**.

## **Decodarea**

### Metoda 1 - folosind caracteristica de compresie

La destinație valoarea tensiunii de ieșire poate fi determinată folosind relația :

$$V_{out} = Limita\_inf_{seg\_i} + cod\_subseg \cdot q_i + \frac{1}{2} \cdot q_i$$

și obținem  $|V_{out}| = 0.25 + 3 \cdot 0.015625 + \frac{1}{2} \cdot 0.015625 = 0.3046875$ , având o eroare de cuantizare de  $eroare\_cuantizare = 0.305193 - 0.3046875 = 0.0005055$ .

### Metoda 2 – folosind tabelul de codare

Din codul primit, pe baza Tabelului 2 putem determina semnul, poziția primului bit de 1 și următorii 4 biți **0010011xxxxxx**. Completăm ultimi 6 biți și obținem **0010011100000**<sub>2</sub>. Transformăm în zecimal și obținem valoarea 1248. Calculăm tensiunea de ieșire,  $|V_{out}| = 1248 \cdot q_e = 1248 \cdot 2^{-12} = 0.3046875$ .

Anexa 1 – Legea  $\mu$  (sursa: Recomandarea G.711)

TABLEAU 2a / G.711  
 $\mu$ -law, positive input values

1	2	3	4	5	6	7	8
Segment number	Number of intervals $\times$ interval size	Value at segment end points	Decision value number $n$	Decision value $x_n$ (see Note 1)	Character signal	Quantized value (value at decoder output) $y_n$	Decoder output value number
					Bit number 1 2 3 4 5 6 7 8		
8	$16 \times 256$	8159	(128)	(8159)	-----	8031	127
			127	7903	1 0 0 0 0 0 0 0		
7	$16 \times 128$	4063	113	4319	(see Note 2)	4191	112
			112	4063	1 0 0 0 1 1 1 1		
6	$16 \times 64$	2015	97	2143	(see Note 2)	2079	96
			96	2015	1 0 0 1 1 1 1 1		
5	$16 \times 32$	991	81	1055	(see Note 2)	1023	80
			80	991	1 0 1 0 1 1 1 1		
4	$16 \times 16$	479	65	511	(see Note 2)	495	64
			64	479	1 0 1 1 1 1 1 1		
3	$16 \times 8$	223	49	239	(see Note 2)	231	48
			48	223	1 1 0 0 1 1 1 1		
2	$16 \times 4$	95	33	103	(see Note 2)	99	32
			32	95	1 1 0 1 1 1 1 1		
1	$15 \times 2$	31	17	35	(see Note 2)	33	16
			16	31	1 1 1 0 1 1 1 1		
↓	$1 \times 1$	31	2	3	(see Note 2)	2	1
			1	1	1 1 1 1 1 1 1 0		
			0	0	1 1 1 1 1 1 1 1	0	0

Note 1 – 8159 normalized value units correspond to  $T_{\max} = 3.17$  dBm0.

Note 2 – The character signal corresponding to positive input values between two successive decision values numbered  $n$  and  $n + 1$  (see column 4) is  $(255 - n)$  expressed as a binary number.

Note 3 – The value at the decoder output is  $y_0 = x_0 = 0$  for  $n = 0$ , and  $y_n = \frac{x_n + x_{n+1}}{2}$  for  $n = 1, 2, \dots, 127$ .

Note 4 –  $x_{128}$  is a virtual decision value.

Note 5 – In Tables 1/G.711 and 2/G.711 the values of the uniform code are given in columns 3, 5 and 7.

Anexa 2 – Legea A (sursă: Recomandarea ITU-T G.711)

TABLE 1a/G.711

A-law, positive input values

1	2	3	4	5	6	7	8
Segment number	Number of intervals × interval size	Value at segment end points	Decision value number $n$	Decision value $x_n$ (see Note 1)	Character signal before inversion of the even bits	Quantized value (value at decoder output) $y_n$	Decoder output value number
					Bit number 1 2 3 4 5 6 7 8		
		4096	(128)	(4096)	-----		
7	16 × 128		127	3968	1 1 1 1 1 1 1 1	4032	128
					(see Note 2)		
6	16 × 64	2048	113	2176	1 1 1 1 0 0 0 0	2112	113
			112	2048	(see Note 2)		
5	16 × 32	1024	97	1088	1 1 1 0 0 0 0 0	1056	97
			96	1024	(see Note 2)		
4	16 × 16	512	81	544	1 1 0 1 0 0 0 0	528	81
			80	512	(see Note 2)		
3	16 × 8	256	65	272	1 1 0 0 0 0 0 0	264	65
			64	256	(see Note 2)		
2	16 × 4	128	49	136	1 0 1 1 0 0 0 0	132	49
			48	128	(see Note 2)		
1	32 × 2	64	33	68	1 0 1 0 0 0 0 0	66	33
			32	64	(see Note 2)		
			1	2			
			0	0	1 0 0 0 0 0 0 0	1	1

Note 1 – 4096 normalized value units correspond to  $T_{\max} = 3.14 \text{ dBm0}$ .

Note 2 – The character signals are obtained by inverting the even bits of the signals of column 6. Before this inversion, the character signal corresponding to positive input values between two successive decision values numbered  $n$  and  $n + 1$  (see column 4) is  $(128 + n)$  expressed as a binary number

Note 3 – The value at the decoder output is  $y_n = \frac{x_{n-1} + x_n}{2}$  for  $n = 1, \dots, 127, 128$ .

Note 4 –  $x_{128}$  is a virtual decision value.

Note 5 – In Tables 1/G.711 and 2/G.711 the values of the uniform code are given in columns 3, 5 and 7.