

## Dekodiranje

Dekodiranje je analogno kodiranju. Na vhodu dobimo množico bitov in enako tabelo simbolov kot pri kodiranju (Tabela 1). Iz množice vhodnih bitov z istimi preslikavami dekodiramo množico simbolov, ki jih pošljemo na izhod kot rezultat dekodiranja. Uporabimo enako inicializacijo kot za kodiranje in enake enačbe Enačba 1, Enačba 2 in Enačba 3. Pri preslikavah E1, E2 in E3 ne zapisujemo bitov na izhod, ampak beremo bit po bit iz vhodnega bitnega niza, kot je zapisano spodaj:

// Preslikava E1

```
spMeja = 2 · spMeja  
zgMeja = 2 · zgMeja + 1  
polje = 2 · polje + naslednjiBit()
```

// Preslikava E2

```
spMeja = 2 · (spMeja – drugaCetrtna)  
zgMeja = 2 · (zgMeja – drugaCetrtna) + 1  
polje = 2 · (polje – drugaCetrtna) + naslednjiBit()
```

// Preslikava E3

```
spMeja = 2 · (spMeja – prvaCetrtna)  
zgMeja = 2 · (zgMeja – prvaCetrtna) + 1  
polje = 2 · (polje – prvaCetrtna) + naslednjiBit()
```

Pri inicializaciji uporabimo še pomožno spremenljivko *polje*, ki jo napolnimo z št. bitov - 1 vhodnimi biti (npr. če smo kodirali z 8 biti v spremenljivko *polje* preberemo prvih 7 bitov iz vhodnega niza bitov. Glej spodnji primer dekodiranja).

Za dekodiranje simbola uporabimo naslednjo enačbo:

$$vrednost = \frac{polje - spMeja}{korak}$$

Ko izračunamo **vrednost**, jo poiščemo v Tabeli 1, v katerem intervalu leži (npr. če izračunamo vrednost je 0.5, vidimo, da ima simbol G  $spMeja = 0$  in  $zgMeja = 1$ , kar pomeni, da vrednost leži v intervalu simbola G in tako dekodiramo simbol G ter ga pošljemo na izhod).

Metoda *naslednjiBit()* vrne 1 ali 0 na naslednji poziciji vhodnega niza bitov.

## Primer dekodiranja:

### Inicializacija:

$$spMeja = 0000000[0]$$

$$zgMeja = 2^{n-1} - 1 = 2^{8-1=7} - 1 = 1111111[127]$$

$$drugaCetrtna = \left\lfloor \frac{zgMeja + 1}{2} \right\rfloor = 1000000[64]$$

$$prvaCetrtna = \left\lfloor \frac{drugaCetrtna}{2} \right\rfloor = 0100000[32]$$

$$tretjaCetrtna = \lfloor prvaCetrtna \cdot 3 \rfloor = 1100000[96]$$

Ker smo pri kodiranju uporabili 8 bitov za predstavitev globalnega intervala, preberemo prvih 8-1=7 bitov iz vhodnega niza bitov (če bi delali z 32 biti bi uporabili prvih 31 bitov iz vhoda).

$$polje = 00010000111$$

Polje		Korak	Vrednost	Sim	Meje	E1/E2	Polje	E3	Polje
0001000	L	25	$\frac{8-0}{25} = 0$	G $0 \leq G < 1$	$0 + 25 * 0 = 0$	0	0100001	0	0100001
	H				$0 + 25 * 1 - 1 = 24$	99		99	
0100001	L	20	$\frac{33-0}{20} = 1$	E $1 \leq E < 2$	$0 + 20 * 1 = 20$	40	1000011	16	1000111
	H				$0 + 20 * 2 - 1 = 39$	79		95	
1000111	L	16	$\frac{71-16}{16} = 3$	M $2 \leq M < 4$	$16 + 16 * 2 = 48$	48	1000111	0	1011100
	H				$16 + 16 * 4 - 1 = 79$	79		127	
1011100	L	25	$\frac{92-0}{25} = 3$	M $2 \leq M < 4$	$0 + 25 * 2 = 50$	50	1011100	50	1011100
	H				$0 + 25 * 4 - 1 = 99$	99		99	
1011100	L	10	$\frac{92-50}{10} = 4$	A $4 \leq A < 5$					
	H								

Rezultat: **GEMMA**