Exponential-Golomb coding

From Wikipedia, the free encyclopedia

An **exponential-Golomb code** (or just **Exp-Golomb code**) is a type of universal code. To encode any nonnegative integer *x* using the exp-Golomb code:

1. Write down $x+1$ in binary
2. Count the bits written, subtract one, and write that number of starting zero bits preceding the previous bit string.

The first few values of the code are:

```
0 ⇒ 1 ⇒ 1
1 ⇒ 10 ⇒ 010
2 ⇒ 11 ⇒ 011
3 ⇒ 100 ⇒ 00100
4 ⇒ 101 ⇒ 00101
5 ⇒ 110 ⇒ 00110
6 ⇒ 111 ⇒ 00111
7 ⇒ 1000 ⇒ 0001000
8 ⇒ 1001 ⇒ 0001001
...[1]
```

This is identical to the Elias gamma code of $x+1$, allowing it to encode 0.[2]

Contents [hide]

Extension to negative numbers [edit]

Exp-Golomb coding for $k = 0$ is used in the H.264/MPEG-4 AVC and H.265 High Efficiency Video Coding video compression standards, in which there is also a variation for the coding of signed numbers by assigning the value 0 to the binary codeword '0' and assigning subsequent codewords to input values of increasing magnitude (and alternating sign, if the field can contain a negative number):

```
 0 ⇒ 0 ⇒ 1 ⇒ 1
 1 ⇒ 1 ⇒ 10 ⇒ 010
−1 ⇒ 2 ⇒ 11 ⇒ 011
 2 ⇒ 3 ⇒ 100 ⇒ 00100
−2 ⇒ 4 ⇒ 101 ⇒ 00101
 3 ⇒ 5 ⇒ 110 ⇒ 00110
−3 ⇒ 6 ⇒ 111 ⇒ 00111
 4 ⇒ 7 ⇒ 1000 ⇒ 0001000
−4 ⇒ 8 ⇒ 1001 ⇒ 0001001
...[1]
```

In other words, a non-positive integer $x \le 0$ is mapped to an even integer $-2x$, while a positive integer $x>0$ is mapped to an odd integer $2x-1$.

Exp-Golomb coding is also used in the Dirac video codec.[3]

Generalization to order *k* [edit]

To encode larger numbers in fewer bits (at the expense of using more bits to encode smaller numbers), this can be generalized using a nonnegative integer parameter *k*. To encode a nonnegative integer *x* in an order-*k* exp-Golomb code:

1. Encode $\lfloor x/2^k \rfloor$ using order-0 exp-Golomb code described above, then
2. Encode $x \bmod 2^k$ in binary

An equivalent way of expressing this is:

1. Encode $x+2^k-1$ using the order-0 exp-Golomb code (i.e. encode $x+2^k$) using the Elias gamma code), then
2. Delete *k* leading zero bits from the encoding result

Exp-Golomb-*k* coding examples

x	*k*=0	*k*=1	*k*=2	*k*=3
0	1	10	100	1000
1	010	11	101	1001
2	011	0100	110	1010
3	00100	0101	111	1011
4	00101	0110	01000	1100
5	00110	0111	01001	1101
6	00111	001000	01010	1110
7	0001000	001001	01011	1111

x	*k*=0	*k*=1	*k*=2	*k*=3
10	0001011	001100	01110	010010
11	0001100	001101	01111	010011
12	0001101	001110	0010000	010100
13	0001110	001111	0010001	010101
14	0001111	00010000	0010010	010110
15	000010000	00010001	0010011	010111
16	000010001	00010010	0010100	011000
17	000010010	00010011	0010101	011001

x	*k*=0	*k*=1	*k*=2	*k*=3
20	000010101	00010110	0011000	011100
21	000010110	00010111	0011001	011101
22	000010111	00011000	0011010	011110
23	000011000	00011001	0011011	011111
24	000011001	00011010	0011100	00100000
25	000011010	00011011	0011101	00100001
26	000011011	00011100	0011110	00100010
27	000011100	00011101	0011111	00100011

8	0001001	001010	01100	010000	18	000010011	00010100	0010110	011010	28	000011101	00011110	000100000	00100100
9	0001010	001011	01101	010001	19	000010100	00010101	0010111	011011	29	000011110	00011111	000100001	00100101

See also [edit]

- Elias gamma coding
- Elias delta coding
- Elias omega coding
- Universal code

References [edit]

1. ^ a b Richardson, Iain (2010). *The H.264 Advanced Video Compression Standard*. Wiley. pp. 208,221. ISBN 978-0-470-51692-8.
2. ^ Rupp, Markus (2009). *Video and Multimedia Transmissions over Cellular Networks: Analysis, Modelling and Optimization in Live 3G Mobile Networks*. Wiley. p. 149.
3. ^ "Dirac Specification" (PDF). BBC. Retrieved 9 March 2011.