Exercise 2: Combinational Logic

Prof. Dr. Sven Simon

1 Golomb encoder

The Golomb-Rice codes are used for entropy encoding. In Golomb code with a parameter $m = 2^k$, we represent an positive number n using two numbers q and r, where $q = \lfloor \frac{n}{m} \rfloor$ and $r = n - qm$. The quotient q is represented by the *unary code* of q. The unary code for a positive integer x is simply x 0s followed by a 1. The remainder r is represented in $\log_2 m = k$ bits. The following table shows the results of encoding numbers 0–15, with $m = 2^2$:

Symbol	Quotient	Remainder	Unary(Quotient)	Code
0	0	0	1	1 00
1	0	1	1	1 01
2	0	2	1	1 10
3	0	3	1	1 11
4	1	0	01	01 00
5	1	1	01	01 01
6	1	2	01	01 10
7	1	3	01	01 11
8	2	0	001	001 00
9	2	1	001	001 01
10	2	2	001	001 10
11	2	3	001	001 11
12	3	0	0001	0001 00
13	3	1	0001	0001 01
14	3	2	0001	0001 10
15	3	3	0001	0001 11

Given the following entity (given in `gol_enc.vhd`), VHDL describe a Golomb encoder. Assume that k is always 2 for simplicity. Left align the codeword.

```
entity gol_enc is
  port (
    symbol   : in  std_logic_vector(3 downto 0);
    codeword : out std_logic_vector(5 downto 0));
end entity gol_enc;
```

2 Golomb decoder

Golomb-rice codes are uniquely decodable. Design a Golomb decoder. The entity is given below (given also in `gol_dec.vhd`):

```
entity gol_dec is
  port (
    codeword : in  std_logic_vector(5 downto 0);
    symbol   : out std_logic_vector(3 downto 0)
    );
end entity gol_dec;
```

For simplicity, assume `codeword` has always only one valid left-aligned codeword.

3 Verification of the designs

For the sake of validation. Run the test bench `top_tb.vhd`. The testbench cascades the encoder and decoder and compares the streamed input symbols with the streamed output symbols. If the test fails, identify the errors and fix it!