

```

// DUT ports

module mtm_Alu (
    input logic clk,    // posedge active clock
    input logic rst_n,  // reset active low
    input logic sin,    // serial data input
    output logic sout   // serial data output
);

/* -----
* The DUT is 32-bit ALU with serial input and serial output.
*
* The communication protocol uses the same byte (packet) structure for the ALU input
* and output:
*   - the idle line state is 1 (HIGH);
*   - each byte has the format: 0tbbbbbb1 where
*       0 = start bit, LOW (transferred first)
*       t,b = payload bits (MSB first)
*       1 = stop bit, HIGH (transferred last)
*
* There are two types of packets:
*   - DATA: 00bbbbbb1 (where bit t == 0), used to transfer the data
*   - CMD: 01bbbbbb1 (where bit t == 1), used to transfer the operands, CRC and
*         operation results
*
* Below the following code will be used:
* DATA(x) = sending data packet 00bbbbbb1, where bbbbbbb == x
* CTL(x) = sending command packet 01bbbbbb1, where bbbbbbb == x
* -----
* To process two 32-bit numbers, A and B, with ALU, nine bytes must be sent in
* the following order (via sin input), with bbbbbbb data as follows:
* 1. DATA( B[31:24] )
* 2. DATA( B[23:16] )
* 3. DATA( B[15:8] )
* 4. DATA( B[7:0] )
* 5. DATA( A[31:24] )
* 6. DATA( A[23:16] )
* 7. DATA( A[15:8] )
* 8. DATA( A[7:0] )
* 9. CTL( {1'b0, OP, CRC} )
*   where:
*       OP is 3-bit ALU operation code:
*       3'b000 = AND
*       3'b001 = OR
*       3'b100 = ADD
*       3'b101 = SUB
*       other = not used
*       CRC is 4-bit CRC checksum calculated for 68 bits of {B, A, 1'b1, OP} with
*       initial state == 0;
*       CRC poly is:  $x^4 + x + 1$ , for data width = 68
* -----
* The ALU can send back two types of packets, DATA and CTL.
*
* After processing the input data without any errors, ALU sends the 32-bit C data
* using the following set of bytes:
* 1. DATA( C[31:24] )
* 2. DATA( C[23:16] )
* 3. DATA( C[15:8] )
* 4. DATA( C[7:0] )
* 5. CTL( {1'b0, FLAGS, CRC} )
*   where:
*       FLAGS = { Carry, Overflow, Zero, Negative } - 4 bits of ALU flags
*       CRC is 3-bit CRC checksum calculated for 37 bits of { C, 1'b0, FLAGS },
*       with initial value == 0;
*       CRC poly is:  $x^3 + x + 1$ , for data width = 37
*
* ALU can also send an packet, when a processing error occurred:
* CTL( {1'b1, ERR_FLAGS, PARITY} )

```

```

*      where:
*      ERR_FLAGS is a set of bits (MSB first):
*      - ERR_DATA = incorrect input data, wrong number of DATA frames before
*      CTL frame (should always be 8)
*      - ERR_CRC = bad CRC for the input data
*      - ERR_OP = unknown ALU operation code
*      - ERR_DATA (duplicated)
*      - ERR_CRC (duplicated)
*      - ERR_OP (duplicated)
*      PARITY = parity bit for the data of {1'b1, ERR_FLAGS}; event parity
*      (number of ones in {1'b1, ERR_FLAGS, PARITY} should be even.
*/

```