

The following code answers the quiz question.

In this code, $n = 20$ and $m = 5$.

```
#define PULSE_PERIOD 20 // n
#define PULSE_ONE 5 // m

typedef enum{zero, one} states_type;
void m_cycle_regular_pulses(bool &periodic_pulses) {
#pragma HLS INTERFACE ap_none port=periodic_pulses
#pragma HLS INTERFACE ap_ctrl_none port=return

    static states_type state = zero;
    static unsigned int counter = PULSE_PERIOD-1;

    states_type next_state;
    unsigned int next_counter;

    bool periodic_pulses_local;

    switch(state) {
    case zero:
        if (counter == PULSE_ONE) {
            next_counter = PULSE_PERIOD-1;
            next_state = one;
        } else {
            next_counter = counter-1;
            next_state = zero;
        }
        periodic_pulses_local = 0;
        break;
    case one:
        if (counter == PULSE_PERIOD - PULSE_ONE) {
            next_counter = PULSE_PERIOD-1;
            next_state = zero;
        } else {
            next_counter = counter-1;
            next_state = one;
        }
        periodic_pulses_local = 1;

        break;
    default:
        break;
    }

    state = next_state;
    counter = next_counter;
    periodic_pulses = periodic_pulses_local;
}
```

2 Digital System Design with High-Level Synthesis for FPGA

Sequential Circuits

Utilities: Single Cycle Regular Pulses: Quiz Solution

www.highlevel-synthesis.com

}

This is the output waveform after RTL/C Co-simulation

