

Design a 10-minute timer in HLS and evaluate that on the Basys3 board.

10 minutes is equal to $10 * 60 = 600s$.

If we assume the design clock period of $10ns$ then the timer should count for $600 \times \frac{10^9}{10} = 60,000,000,000$ clock cycles.

The following code shows this timer.

```
#include "ten_minutes_timer.h"

typedef enum{idle, running} timer_state_type;
void ten_minutes_timer(bool start_timer, bool reset_timer, bool &end_timer) {
#pragma HLS INTERFACE ap_none port      = start_timer
#pragma HLS INTERFACE ap_none port      = reset_timer
#pragma HLS INTERFACE ap_none port      = end_timer
#pragma HLS INTERFACE ap_ctrl_none port= return

    static timer_state_type state = idle;
    static unsigned long long int timer_variable = 60000000000; //number of
10ns clocks in 6 minutes

    timer_state_type      next_state = state;
    unsigned long long int next_timer_variable = timer_variable;

    bool end_local;

    switch(state) {
    case idle:
        if (start_timer == 1) {
            next_state      = running;
        } else {
            next_state      = idle;
        }
        end_local = 0;
        break;
    case running:

        if (timer_variable == 0) {
            if (reset_timer == 1) {
                next_state      = idle;
                end_local      = 0;
                next_timer_variable = 60000000000;
            } else {
                next_state      = running;
                end_local      = 1;
                next_timer_variable = 0;
            }
        }
    }
```

```
        } else {
            next_timer_variable = timer_variable - 1;
            next_state
            = running;
            end_local
            = 0;
        }
        break;
    default:
        break;
    }

    state
    = next_state;
    timer_variable
    = next_timer_variable;
    end_timer
    = end_local;
}
```

The timer has two inputs *start_timer* and *reset_timer* and one output *end_time*. The timer will start when we press the *start_timer*. When it finishes, it assigns 1 to *end_timer*. As long as we have not pressed the *reset_timer*, the output is one.

After pressing the *reset_timer*, it is ready to start again.