

The following code shows the usage of the *show\_display(...)* function in the *bcd\_counter*.

The resulted RTL code is a single-cycle design. Therefore, running the code on the Basys3 board does not miss any input.

```
#include "bcd_counter_multicycle.h"

void show_display(
    ap_uint<4> counter_state,
    ap_uint<8> &seven_segment_data,
    ap_uint<4> &seven_segment_enable)
{
    switch (counter_state) {
        case 0:
            seven_segment_data = seven_segment_code[0];
            break;
        case 1:
            seven_segment_data = seven_segment_code[1];
            break;
        case 2:
            seven_segment_data = seven_segment_code[2];
            break;
        case 3:
            seven_segment_data = seven_segment_code[3];
            break;
        case 4:
            seven_segment_data = seven_segment_code[4];
            break;
        case 5:
            seven_segment_data = seven_segment_code[5];
            break;
        case 6:
            seven_segment_data = seven_segment_code[6];
            break;
        case 7:
            seven_segment_data = seven_segment_code[7];
            break;
        case 8:
            seven_segment_data = seven_segment_code[8];
            break;
        case 9:
            seven_segment_data = seven_segment_code[9];
            break;
        default:
            break;
    }
    seven_segment_enable = 0b1110;
}

void bcd_counter(bool pulse, ap_uint<8> &seven_segment_data, ap_uint<4> &seven_segment_enable) {

#pragma HLS INTERFACE ap_none port=seven_segment_enable
#pragma HLS INTERFACE ap_none port=seven_segment_data
#pragma HLS INTERFACE ap_none port=pulse
#pragma HLS INTERFACE ap_ctrl_hs port=return

    static ap_uint<4> counter_state = 0;
```

## 2 Digital System Design with High-Level Synthesis for FPGA

### Sequential Circuits

Function Pipelining-Multi-Cycle Design: Quiz Solution

[www.highlevel-synthesis.com](http://www.highlevel-synthesis.com)

```
if (pulse == 1){
    if (counter_state == 9)
        counter_state = 0;
    else
        counter_state++;
}

show_display(counter_state, seven_segment_data, seven_segment_enable);
}
```