

## 1- (BCD Counter)

The following design top-function in HLS solves the problem.

```
typedef enum {s0, s1, s2, s3, s4, s5, s6, s7, s8, s9} bcd_counter_state_type;
void bcd_counter(ap_uint<4> &count) {
#pragma HLS INTERFACE ap_ctrl_none port=return
#pragma HLS INTERFACE ap_none port=count

    static bcd_counter_state_type state = s0;

    ap_uint<4> count_tmp;
    bcd_counter_state_type next_state = state;
    switch(state) {
    case s0:
        next_state = s1;
        count_tmp = 0;
        break;
    case s1:
        next_state = s2;
        count_tmp = 1;
        break;
    case s2:
        next_state = s3;
        count_tmp = 2;
        break;
    case s3:
        next_state = s4;
        count_tmp = 3;
        break;
    case s4:
        next_state = s5;
        count_tmp = 4;
        break;
    case s5:
        next_state = s6;
        count_tmp = 5;
        break;
    case s6:
        next_state = s7;
        count_tmp = 6;
        break;
    case s7:
        next_state = s8;
        count_tmp = 7;
        break;
    case s8:
        next_state = s9;
        count_tmp = 8;
        break;
    case s9:
        next_state = s0;
        count_tmp = 9;
        break;
    }
```

```
    default:
        break;
}

state = next_state;
count = count_tmp;
}
```

2- The following code solves the problem.

```
typedef enum {A, B, C, D} exercise02_fsm_state_type;
void exercise02_fsm(bool x, bool &y) {
#pragma HLS INTERFACE ap_none port=x
#pragma HLS INTERFACE ap_none port=y
#pragma HLS INTERFACE ap_ctrl_none port=return

    static exercise02_fsm_state_type state = A;

    exercise02_fsm_state_type next_state;

    bool y_tmp;

    switch(state) {

    case A:
        if (x == 0) {
            next_state = D;
            y_tmp = 1;
        } else {
            next_state = B;
            y_tmp = 0;
        }
        break;
    case B:
        if (x == 0) {
            next_state = C;
            y_tmp = 0;
        } else {
            next_state = D;
            y_tmp = 0;
        }
        break;
    case C:
        if (x == 0) {
            next_state = A;
            y_tmp = 1;
        } else {
            next_state = B;
        }
    }
```

```

        y_tmp = 0;
    }
    break;
case D:
    if (x == 0) {
        next_state = C;
        y_tmp = 0;
    } else {
        next_state = A;
        y_tmp = 1;
    }
    break;
default:
    break;
}

state = next_state;
y = y_tmp;
}

```

3- The following code solves the problem.

```

typedef enum {A, B, C, D} exercise02_fsm_state_type;
void exercise03_fsm(bool x, ap_uint<2> y) {
#pragma HLS INTERFACE ap_none port=x
#pragma HLS INTERFACE ap_none port=y
#pragma HLS INTERFACE ap_ctrl_none port=return

    static exercise02_fsm_state_type state = A;

    exercise02_fsm_state_type next_state;

    bool y_tmp;

    switch(state) {

    case A:
        if (x == 0) {
            next_state = A;
        } else {
            next_state = B;
        }
        y_tmp = 0b00;
        break;
    case B:
        if (x == 0) {
            next_state = C;
        } else {
            next_state = D;
        }
        y_tmp = 0b01;
    }
}

```

```
        break;
    case C:
        if (x == 0) {
            next_state = C;
        } else {
            next_state = D;
        }
        y_tmp = 0b10;
        break;
    case D:
        if (x == 0) {
            next_state = B;
        } else {
            next_state = A;
        }
        y_tmp = 0b11;
        break;
    default:
        break;
}

state = next_state;
y = y_tmp;
}
```