2018-08-13

DAT093
Introduction to Electronic System Design
Some comments on starting a process

In many cases, like the adder/subtractor and multiplier in lab assignment 1 we have a clocked process for the execution and the process is started by a start signal. What requirements should we put on this start signal? How can it be used? We have some options

- Start the execution when the start pulse goes high
- Start the execution when the start pulse has gone high and then goes low again

In the first case this statement is not enough. The reason is that we should not trigger the process on the rising edge of the start pulse but start the process if the start pulse is high when the clock signal triggers the process. As a result of this we must make sure that the start pulse has gone low again before the execution sequence has ended otherwise the execution will start again without a new start pulse. This can be done in two ways

- We look at the execution sequence and determine how many clock pulses we can allow the start pulse to be high
- We only activate the start pulse during one clock period

Both options require control of how the start pulse is generated which is unnecessary.
If we on the other hand start our execution when the start pulse goes low again then the length of the start pulse doesn´t matter, as long as it is a pulse and not a step and this makes the implementation easier. One thing we must keep in mind though is that we have to make sure that the start pulse really has gone high first so we don´t just start on the low level of the start pulse. This can be done in two ways

- We can set a flag when the start pulse is high and check if that flag is set when the start pulse is low. If the flag is set, then we know that there has been a pulse
- We can implement the design as a state machine (FSM)

Let´s give examples of the two options. The first one as code and the second one as a flow chart.
Using a flag the code might look something like

```
IF RISING_EDGE(clk) THEN
   IF (start='1') THEN
      started_signal<=’1’;
      --initialize the execution process
   ELSIF (started_signal='1') THEN
      --continue with the execution
      --don´t forget
      started_signal<=’0’;
      --before the execution is finished
```

Using a state machine we have

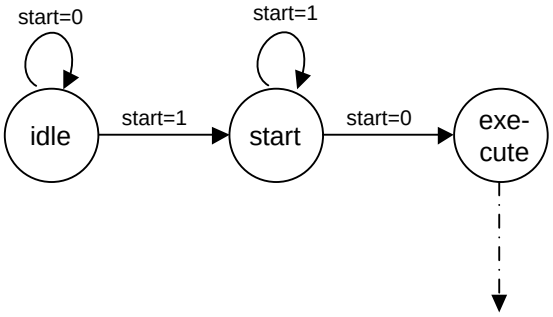

Figure 1 State machine implementation