HAUTE ÉCOLE D'INGÉNIERIE ET DE GESTION DU CANTON DE VAUD

Prof. Yann Thoma

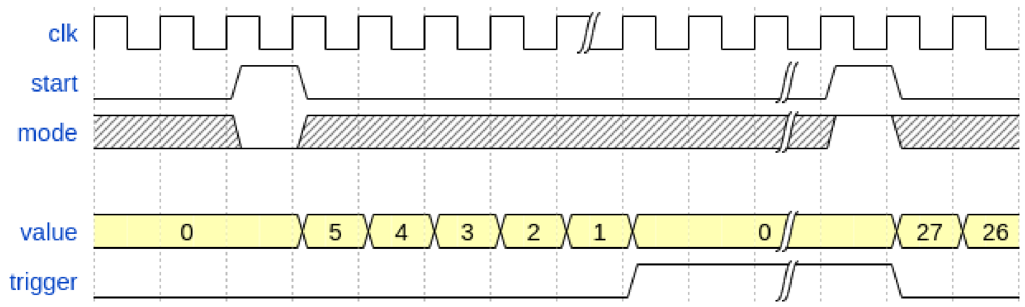
Formal verification
Verification of a timer
CERN training - July 2025

Context

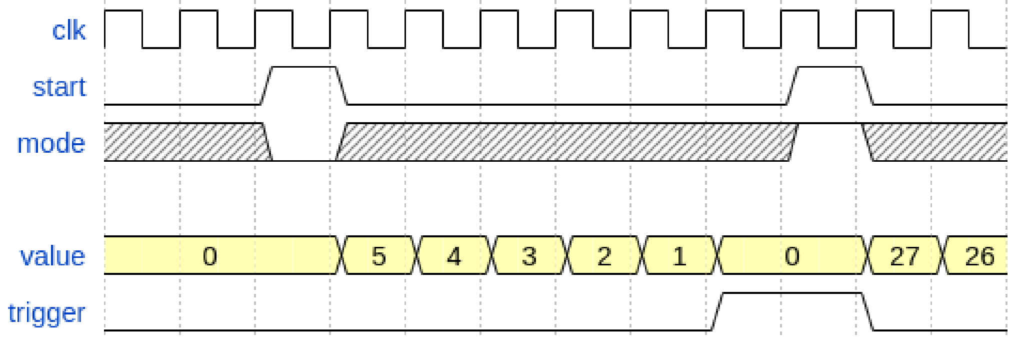
We are interested in verifying the correctness of a timer having the following interface :

```
entity timer is
port(
  clk_i     : in  std_logic;
  rst_i     : in  std_logic;
  start_i   : in  std_logic;
  mode_i    : in  std_logic;
  value_o   : out std_logic_vector(7 downto 0);
  trigger_o : out std_logic
);
end timer;
```

This timer is capable, from its start, of counting a certain number of cycles before generating a signal indicating its completion. The timer is started by activating the `start_i` signal for one clock cycle. When `start_i` is active, the `mode_i` input indicates the duration of the timer. If `mode_i` is set to `'0'`, then the timer must count 6 cycles, and if it is set to `'1'`, it must count 28 cycles. When `start_i` is activated, the timer begins decrementing until it reaches 0. When it reaches 0, the `trigger_o` output changes to '1' and remains so until the timer restarts (when `start_i` returns to '1').
The current value of the counter is also available as the `value_o` output.
The specifications of the timer state the the start input should not be activated before the timer is triggered.
The following waveform illustrates an example of how it works. The grayed-out mode indicates that the current value is not important.



And the next one the minimum time between two activations of `start` :



Propose a set of assertions (and maybe assumptions) that allow you to use formal verification to prove the correct behavior of this timer. Make a good use of `assume` and `assert` to formally verify this timer.
The timer is able to generate errors, so you can test your assertions by changing the following statement in the `timer.sby` file :
`-gERRNO=0`.
`ERRNO` values :

- 0 : valid behavior
- in $[1, 5]$: wrong behavior