





**Question 3 [3 points].** *FPGA and VHDL: synchronizing clock enable with data*

A signal called `clk_en_in` is high the first rising edge of `clk` after `data_in` is ready and is low the next rising edge. We want to design an entity in order to update `data_out` from `data_in` every two `clk_en_in` in the following way; if `data_in` (coded as unsigned) is lower than 8 then `data_out` is equal to `data_in`, otherwise all bits of from `data_out` take the value one. In addition, a signal called `clk_en_out` must be high the first rising edge of `clk` after `data_out` is ready and must be low the next rising edge.

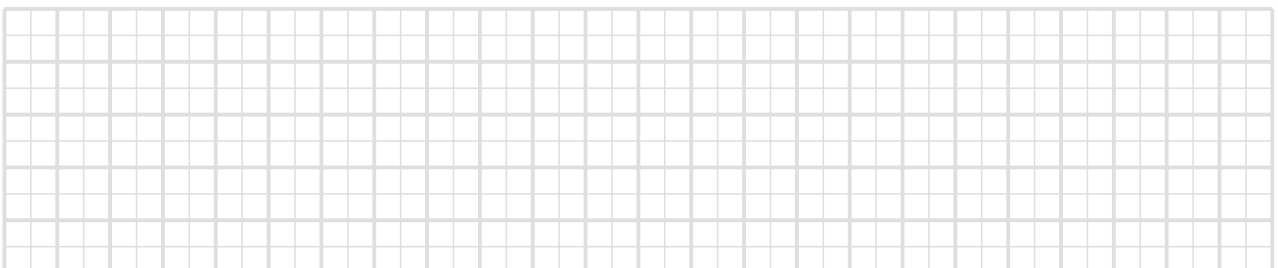
The following VHDL code tries to do that.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity se_exam_2023 is
  port (clk, clk_en_in : in std_logic;
        data_in : in std_logic_vector(3 downto 0);
        clk_en_out : out std_logic;
        data_out : out std_logic_vector(3 downto 0));
end entity;

architecture arch_1 of se_exam_2023 is
  signal n : unsigned(1 downto 0) := to_unsigned(0,2);
begin
  process(clk)
  begin
    if rising_edge(clk) then
      if clk_en_in = '1' then
        clk_en_out <= '0';
        if n = 2 then
          if unsigned(data_in) < 8 then
            clk_en_out <= '1';
            data_out <= data_in;
            n <= to_unsigned(0,n'length);
          else
            data_out <= (others => '1');
          end if;
        end if;
        n <= n+1;
      end if;
    end if;
  end process;
end architecture;
```

- a) Unfortunately, `clk_en_out` is not well generated. Draw the actual digital waveform (`clk`, `clk_en_in` and `clk_en_out`). Consider that there is one `clk_en_in` for every three `clk`, and that `data_in` is "0000".



- b) Modify the code. Make the modifications next to the code of this page.

c) Draw the correct digital waveform.



**Question 4 [2 points].** *Qualifiers and fixed-width integer types in C*

The next code, *code<sub>1</sub>*, belongs to one of your classmates who is taking the *PBN* subject. The code is used in one of the modules of the course project.

```
#include ...
#define ...

typedef enum { ABoff, Aclear, Bclear, AtoB, BtoA } state_t;
static volatile uint8_t ticks;
static volatile state_t state;
static semaph_t semaphA, semaphB;

ISR(TIMER1_COMPA_vect) {
    ticks++;
    if (ticks == 20) {
        if (state == AtoB) state = Bclear;
        if (state == BtoA) ...
    } else if (ticks == 80) {
        if (state == Aclear) ...
    } else if (ticks == 100) {
        ...
    }
}

void control_init(void) {
    ticks = 0;
    state = ABoff;
    ...
}

void control_force(street_t t) {
    if (state == ABoff){
        state = ...
        ticks = ...
        ...
    }
    ...
}
```

This other code, *code<sub>2</sub>*, belongs to another of your classmates.

```
#include ...
#define ...

static int ticks = 0;
static enum { Aclear, AtoB, Bclear, BtoA, ABoff} state;
static semaph_t SemA;
static semaph_t SemB;
```

```

ISR(TIMER1_COMPA_vect){
  --ticks;
  if(ticks == 0){
    if(state == Aclear){
      ...
      ticks = ...
      state = ...
    }
    else if ...
  }
}

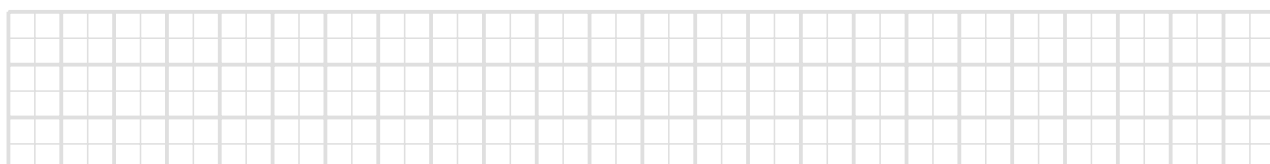
void control_init(void){
  ticks = 0;
  state = ABoff;
  ...
}

void control_force(street_t t){
  if(state != ABoff){
    ...
    if(state == Bclear){
      ticks = ...
      state = ...
    }
    ...
  }
  ...
}

```

First, focus on the variable *ticks*.

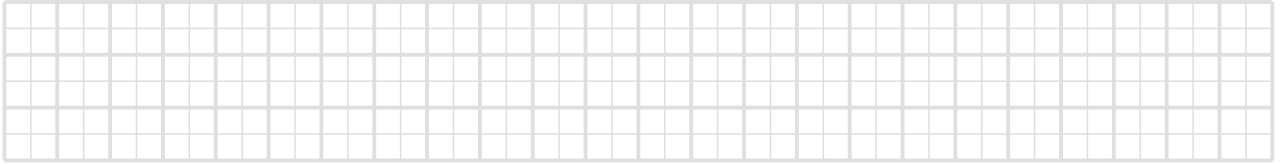
- a) Comment on the validity of this sentence: *the variable ticks should be uint8\_t and not int*. Ignore this difference in all the remaining questions.



Next, focus on the definition of the variables *ticks* and *state* and answer the following questions:

- a) Mark the true sentence/sentences:
- If both codes prove to work as expected then both definitions are correct
  - None of the codes will work because the definitions are not correct
  - One of the codes will work and the other will not work
  - One code will work and the other cannot be said
  - None of the previous sentences is true

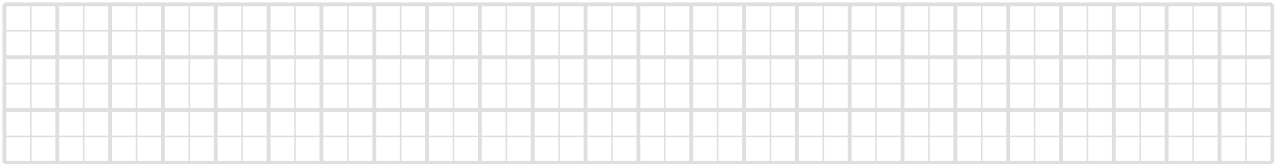
b) Justify your previous answer.

A rectangular grid consisting of 20 columns and 10 rows, intended for writing a justification.

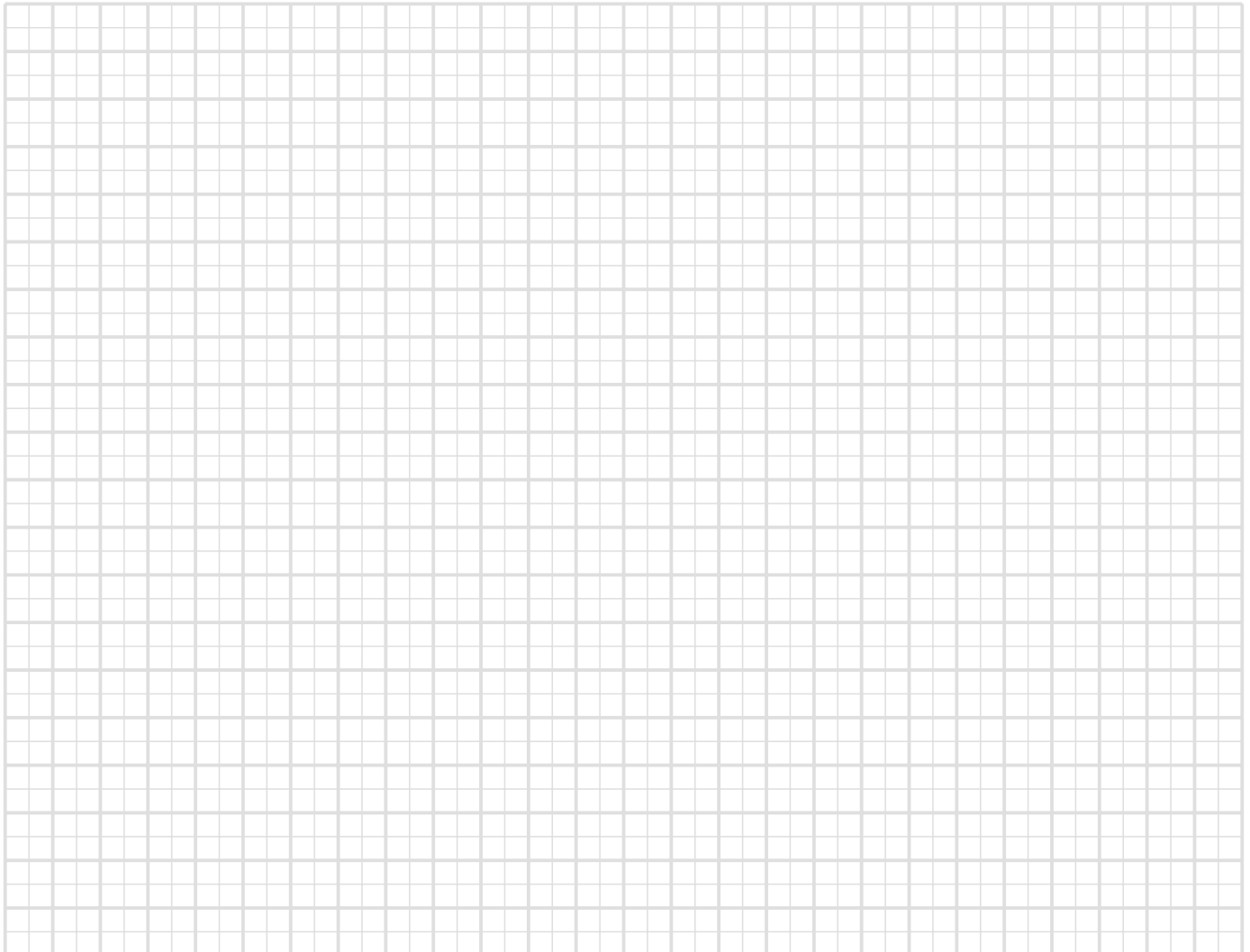
c) Mark the FALSE sentence/sentences:

- The definitions in one code are fine and the definitions in the other code are not
- The definitions in one code are fine and the definitions in the other code cannot be said
- The definitions in both codes are fine
- One or more of the previous sentences is false

d) Justify your previous answer.

A rectangular grid consisting of 20 columns and 10 rows, intended for writing a justification.

Draft area

A large rectangular grid consisting of 20 columns and 40 rows, intended for writing a draft.



