Exercise 6: Plotter

G. Kemnitz*, TU Clausthal, Institute of Computer Science

May 29, 2012

Abstract

Starting from the circuit design of a fan control and a control of a VGA monitor a simple plotter should be designed that displays the speed of the fan in dependence of the time on a monitor screen.

1 Overall function

A plotter displays a function of time as a graphic. Our plotter should be built out of:

- a circuit to measure the fan speed,
- a memory for storing captured signal values and
- a VGA controller.

The measuring circuit for the fan speed produces in discrete time steps speed values and stores them in the memory. Is the signal memory full, new values override old values circularly. The VGA controller displays the captured speed values in the memory in a graphic. To simplify readability in addition to the wave form of the speed signal a grid should be displayed and the newest measured value should be marked by a vertical read line that moves as a cursor from left to right over the screen. Figure 1 shows how the monitor screen should look.

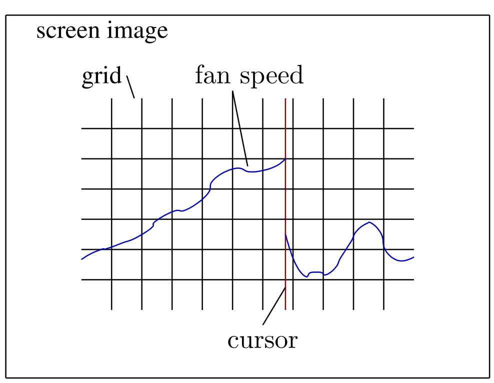

Figure 1: Screen output of the plotter

*Tel. 05323/727116

The horizontal and vertical grid lines can be drawn on the screen by comparing the actual pixel and line counter with the coordinates of the lines and displaying if equal the grid color [see exercise 5 (VGA)]. The wave form can be drawn by reading for each column the corresponding value from the signal memory and comparing it with the line number. If signal value and line number are equal the pixel has to be drawn in line color (blue). Else if it is a grid point, it has to be drawn with the grid color (black) and else with the background color (white):

```
if Zeile = AZ_RAM(to_integer(unsigned(Spalte))) then
   rgb <= signal_color;
elsif (<grid_point>)
   rgb <= grid_color;
else
   rgb <= background_color;
end if;
```

Pay attention: The here used line and pixel numbers are related to the coordinate system for signal presentation and differ from the values of the pixel and the line counter. Transformations are necessary.

2 Measuring fan speed

The fan speed should be measured by a light barrier on the expansion board. The fan has nine blades and rotates with a maximum speed of 100 revolutions per second. The displayed value for this speed should be approximately 220. Both, the rising and the falling edges of the of the light barrier output should be counted. These are 18 count steps per revolution and for the maximum speed approximately 18 steps per 10 ms. To detect all transitions the light barrier signal must be sampled at least twice per step. So the sampling frequency must be at least:

$$f_a > 2 \cdot \frac{18 \text{ (steps\_per\_revolution)}}{10\,\text{ms}}$$

The required measuring time is about:

$$t_{mess} = \frac{\text{display\_value} \cdot \text{time\_per\_revolution}}{\text{steps\_per\_revolution}} \approx \frac{220 \cdot 10\,\text{ms}}{18} = 122\,\text{ms}$$

It should be recommended to use a 12 kHz sampling clock. To get the 122 ms measuring window the sampling clock should be further divided 2^{11}. In the first clock of the measuring circle the step counter should be cleared, in the last clock the counter value should be stored in the signal memory and in all other steps, if the last two sampled values distinguish, the step counter should be increased by one.

The signal memory should have 512 memory places with a size of one byte. The type declaration could be:

```
type AZ_RAM_type is array(0 to 511) of std_logic_vector(7 downto 0)
```

The signal memory should be written circularly: the first value to address zero, the second to address one etc. up to the 512th value to address 511. The 513th should be written again to address null. To allow the system to select a synchronous block memory for the signal memory, the write access must be described in a sampling process:

```
signal AZ_RAM: AZ_RAM_typ;
signal write_counter: natural range 0 to 511;
signal sampl_counter:  std_logic_vector(7 downto 0);
...
process(Takt_12kHz)
```

```
beginn
  if rising_edge(Takt_12kHz) then
    ...
    if <letzer_Takt_des_Messfensters> then
      AZ_RAM(write_counter) <= sampl_counter;
      -- modulo 512 is only necessary for simulation.
      -- In the circuit the write_counter becomes a
      -- 9 bit register, dthat otherwise also counts modulo 512.
      write_counter <= (write_counter + 1) mod 512;
    end if;
  end if;
end process;
```

2.1 Monitor control

The VGA controller circuit can mostly taken from exercise before. Clock frequency also should be 25 MHz and the image resolution 640 pixel x 480 lines. The background color should be black. In the middle of the screen a white grid-window of 513 pixel x 257 lines with 32 pixel line distance should be presented (9 horizontal and 17 vertical grid lines). The cursor line should be drawn red as in figure. The color for the waveform should be cyan.

The first vertical grid has the the pixel counter value 207. In this column also the spot for the value from block memory address zero has to be presented. Correspondingly, the next column is reserved for the second block memory value and generally column n for the value on address $n - 207$. It ends at column 207+511=718. The value zero should be presented at line 398 and values $w > 0$ at line $398 - w$. The algorithm for drawing wave form is correspondingly:

```
signal hCounter : integer range 0 to 799 := 0;
signal vCounter : integer range 0 to 520 := 0;
...
process (Takt_25MHz)
begin
  ...
  if <column in range from 207 to 718> then
    Wert <= AZ_RAM(hCounter-207);
    if to_integer(unsigned(Wert)) = 398 - vCounter then
      rgb <= cyan;
    elsif <grid point> then grb <= <grid_color>;
    else rb <= <background_color>;
    end if;
  end if;
end process;
```

A concurrent addressing of the block memory – the writing in the 12 kHz process and the reading in the 25 MHz process – is in our design system possible and will be translated in a dual port RAM. In rare cases the circuit may write and read simultaneously at the same address. For the data read from the memory are only displayed an not processed potential malfunction caused by the simultaneous access will almost not disturb the overall operation.

3 Additional exercise

Add a presentation of the waveform of the temperature, measured by the sensor behind the fan. For this the measuring circuit from exercise 4 has to be adapted to sample temperature values and store them in a second block memory of the same size. The VGA controller has to be extended to draw a second waveform.

4 Check list for the compliance test

Presentation of a working circuit. it should be visible If the fan is switched on, how speed increases and after switch of how it decreases.