TECHNISCHE UNIVERSITÄT WIEN

Institut für Computertechnik
Institute of Computer Technology

MELODI

Mass E-Learning of design, test, and prototyping DIgital hardware

LEVEL 5

LSR (LED Stepper Rotation)

In this task, you have to control a stepper motor on which a raster disk is mounted. The teeth of the raster disk can be detected by two light barriers. One of these light barriers has to be used to detect the rotation of the stepper motor. The details of the stepper motor control can be found in the task *[Level 2] Stepper: Spin the disk!*. There are also four LEDs built in that you can control independently of each other. *Figure 1* below shows the details of the of the LSR task.

Your task is to generate a **PWM** with **20KHz** and a duty cycle of **40%**. The **stepping frequency** should be **100Hz** starting with a clockwise rotation (see *Table 1*). Establish a **counter** with an initial value of 0. Whenever the **light-barrier [1]** is interrupted, the counter should be incremented and the counter's actual value has to be displayed on the four LEDs in a binary style (LSB = LED[1]). In addition, display those four bits on MELODI's webpage using the VIRTUAL FLAGs.

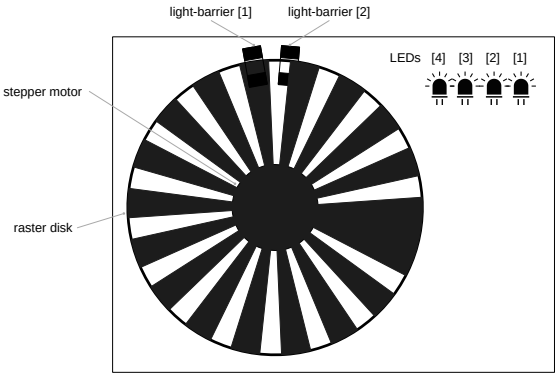


Figure 1: Details of the LSR Task.

As soon as the counter reaches the value of 12 (=0b1100), the counter and the LEDs shall be reset and the stepper motor must be rotated counter clockwise. Now count the interrupts again and drive the LEDs and VIRTUAL FLAGs accordingly. After 12 interrupts, reset the counter and the LEDs, turn the stepper again in clockwise direction and so on...

Step	O[0]	O[1]	O[2]	O[3]
1	PWM	low	PWM	low
2	low	PWM	PWM	low
3	low	PWM	low	PWM
4	PWM	low	low	PWM
1	PWM	low	PWM	low
...	...	..	..	..
change direction after 12 interrupts				
1	PWM	low	low	PWM
2	low	PWM	low	PWM
3	low	PWM	PWM	low
4	PWM	low	PWM	low
1	PWM	low	low	PWM
..	..	..	..	..

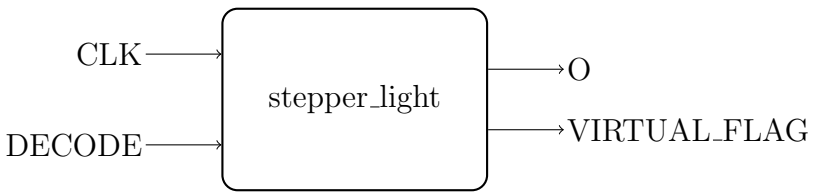
Table 1: Unipolar stepper control.

Important Note: Increase the counter only when a light barrier is being interrupted (i.e. at the rising edge of the signal of the light barrier). Also, observe the signal at the transitions of the raster disk carefully and perform debouncing if necessary.

Entity

Your task is to program the behavior of an entity called "stepper_light". This entity is declared in the attached file "stepper_light.vhdl" and has the following properties:

- Input: CLK (100MHz) with type std_logic
- Input: DECODE with type std_logic_vector of length 2
- Output: O with type std_logic_vector of length 8
- Output: VIRTUAL_FLAG with type std_logic_vector of length 4



Do not change the file "stepper_light.vhdl".

LSR Control		
Component	Port	Value
stepper motor	O[0]	PWM = '0' / '1'
stepper motor	O[1]	PWM = '0' / '1'
stepper motor	O[2]	PWM = '0' / '1'
stepper motor	O[3]	PWM = '0' / '1'
led [1]	O[4]	'1'=ON / '0'=OFF (LSB)
led [2]	O[5]	'1'=ON / '0'=OFF
led [3]	O[6]	'1'=ON / '0'=OFF
led [4]	O[7]	'1'=ON / '0'=OFF (MSB)
light-barrier [1]	DECODE[0]	'1'=DETECTED / '0'=NOT_DETECTED
light-barrier [2]	DECODE[1]	'1'=DETECTED / '0'=NOT_DETECTED
virtual flag 1	VIRTUAL_FLAG[0]	'1'=ENABLE / '0'=DISABLE (LSB)
virtual flag 2	VIRTUAL_FLAG[1]	'1'=ENABLE / '0'=DISABLE
virtual flag 3	VIRTUAL_FLAG[2]	'1'=ENABLE / '0'=DISABLE
virtual flag 4	VIRTUAL_FLAG[3]	'1'=ENABLE / '0'=DISABLE (MSB)

Table 2: LSR control.

This behavior has to be programmed in the attached file "stepper_light_beh.vhdl". To turn in your solution write an email to vhdlabgabe+e384@tuwien.ac.at with subject "Result Task 5" and attach your behavior file(s): **stepper_light_beh.vhdl**

Good Luck and May the Force be with you.