

## Practical 6 / *Prakties 6*

Hierdie opdrag moet gedemonstreer word vir die dosent of 'n demi voor die einde van jou groepsessie, en 'n punt van 0, 1, of 2 sal toegeken word. U \*.vhd lêer moet ook op SUNLearn gelaai word voor 17h00 7 Mei 2015.

*This assignment must be demonstrated to the lecturer or a demi before the end of your group's session, and a mark of 0, 1, or 2 will be awarded. Your \*.vhd file must also be uploaded onto SUNLearn before 17h00 7 May 2015.*

### Komponente - *Components*

Elke student moet 'n DE0 bordjie uitteken en hou tot die einde van die semester. As die bordjie aan die einde van die semester nie teruggegee word in dieselfde toestand as waarin dit uitgeteken is nie, sal daar geen punte vir hierdie module vir die betrokke student toegeken word nie.

*Every student must sign out a DE0 board and keep it until the end of the semester. If the board is not returned at the end of the semester in the same condition as it was signed out, no marks will be given to the student who signed it out.*

**Werk op jou eie. Elke student moet sy/haar eie kode skryf en op SUNLearn oplaai.**

***Work on your own. Each student must write his/her own code and upload it onto SUNLearn.***

### Opdragte - *Tasks*

Laai die projek templaar op SUNLearn af en unzip dit. Maak die projek oop deur op die .qpf lêer te dubbel-klik. Implementeer al die vrae in die bestaande argitektuur, en programmeer die DE0 bordjie vir demonstrasie. 'n veelvoudige funksie tydhouer/teller moet geïmplementeer word. SW(1) en SW(0) se toestande gaan bepaal watter funksie aktief is:

*Download and unzip the project template on SUNLearn. Open the project by double clicking on the .qpf file. Implement all the questions in the existing architecture, and program them onto the DE0 board for demonstration. A multi-function timer/counter must be implemented. The states of SW(1) and SW(0) will determine which function is currently active:*

SW(1) & SW(0)	Mode	Description
"00"	Running	Display "0000", "1111", "2222", ..., "9999", "0000", at 1Hz
"01"	Duration counter	Determine the duration of a button press
"10"	Event counter	Count the number of times a button is pressed, with clear
"11"	Stopwatch	Perform like a stop watch, with start/stop and clear

Figure 1: The different functions of the timer/counter.

Vir hierdie prakties sal die 7-segment verbindings van prakties 5 uitgebrei moet word om nommers vanaf "0000" tot "9999" te kan ver-

*For this practical you will need to expand the 7-segment display of practical 5 to enable it to display numbers ranging from "0000" up to*

toon. Daar sal ook addisionele klok verdelers geïmplementeer moet word, aangesien daar beide 1 Hz en 100 Hz klokseine nodig word.

1. **Running:** Skryf VHDL kode om HEX0, HEX1, HEX2, en HEX3 teen 1 Hz te laat optel as volg: “0000”, “1111”, “2222”, – > “9999”, waarna dit oorrol na “0000” en weer van voor af begin tel.
2. **Duration Counter:** Skryf VHDL kode om te bepaal hoe lank BUTTON(2) ingehou word. Die teller tel slegs terwyl BUTTON(2) ingehou word, en gaan voort waar dit vantevore opgehou het wanneer dit weer ingehou word. Die 7-segment vertooneenheid vertoon die verlope tyd in sekondes <punt> centisekondes (SS.CC). BUTTON(1) word gebruik om die teller te herstel na 00.00.
3. **Event Counter:** Skryf VHDL kode om te tel hoeveel keer BUTTON(2) gedruk word, en vertoon die resultaat op die 7-segment vertooneenheid. BUTTON(1) herstel die teller na “0000”.
4. **Stopwatch:** Skryf VHDL kode om tyd te hou, soortgelyk aan ’n stophorlosie. Wanneer hierdie funksie aktief is, word BUTTON(2) gedruk (en gelos) om die tyd-opteller te aktiveer, en dan weer gedruk (en gelos) om dit te stop. Die 7-segment vertooneenheid vertoon die verlope tyd in sekondes <punt> centisekondes. BUTTON(1) word gebruik om die teller te herstel na 00.00.

“9999”. *Additional clock dividers will also be needed, since both 1 Hz and 100 Hz clock signals are needed.*

1. **Running:** *Write VHDL code to make HEX0, HEX1, HEX2, and HEX3 count at 1 Hz in the following sequence: “0000”, “1111”, “2222”, – > “9999”, after which it rolls over to “0000” and starts counting from the beginning again.*
2. **Duration Counter:** *Write VHDL code to determine how long BUTTON(2) is pressed. The counter only counts while BUTTON(2) is pressed, and continues from where it left off when it is pressed again. The 7-segment display displays the accumulated time in seconds <full stop> centiseconds (SS.CC). BUTTON(1) is used to reset the counter to 00.00.*
3. **Event Counter:** *Write VHDL code to count how many times BUTTON(2) is pressed, and show the result on the 7-segment display. BUTTON(1) resets the counter to “0000”.*
4. **Stopwatch:** *Write VHDL code to keep time, similar to a stopwatch. When this function is active, BUTTON(2) is pressed (and let go) to activate the timer, and then pressed (and let go) to stop the time-counter. The 7-segment display displays the accumulated time in seconds <full stop> centiseconds. BUTTON(1) is used to reset the counter to 00.00.*

**Onthou om jou kode op SUNLearn op te laai voor 17h00 7 Mei 2015!**

***Remember to upload your code to SUN-Learn before 17h00 7 Mei 2015.***