

Rekenaarstelsels 245 - Prakties 7

Computer Systems 245 - Practical 7

2015-09-25

In vandag se prakties gaan ons jou saamsteltaal kode van prakties 4 en prakties 5 hergebruik.

In today's practical we are going to reuse your assembly code for practical 4 and practical 5.

1 Studentenommer funksies

In prakties 4 en prakties 5 het ons die volgende vier funksies geskryf:

calculateUniqueNumber het die grootste 4 getalle opgetel, kleinste 4 getalle opgetel, en daarna die twee antwoorde met mekaar gemaak.

generateChecksum het 'n algoritme gebruik om 'n toetsom van 'n studentenommer te bereken.

validateChecksum het 'n modulus operator geïmplementeer om die toetsom na te gaan.

bubbleSort het die getalle in jou studentenommer gesorteer volgens grootte.

Verwys na prakties 4 en prakties 5 se opdrag en voltooi die selfde vier funksies deur dit in C te skryf.

'n Raamwerk vir die C-funksies wat voltooi moet word is beskikbaar op learn.sun.ac.za. Wanneer jy die projek in e2studio begin, sal jy jou .asm-lêer van prakties 5 ook moet byvoeg onder die bronkode vouer.

2 Spoedverskil tussen ASM en C

In prakties 1 het ons kode geskryf om 'n LED aan en af te skakel, en ook om sy toestand te verander. Gebruik hierdie kode en voeg dit by tot hierdie prakties. Ons wil die LED gebruik om te kan sien hoe lank 'n funksie neem om uit te voer. 'n Voorbeeld hoe om dit te doen is deur die LED se toestand om te swaai, daarna 'n funksie te roep, en alles in 'n oneindige lus te plaas. Die LED sal dus flits met 'n aan en af tyd van die duur van die funksie.

Verander jou kode om die LED op hierdie manier te laat puls, maar doen dit slegs vir een funksie op 'n slag. Gebruik dan die ossilloskoop om die presiese tydsduur wat die LED aan en af is te meet. Herhaal die prosedure vir al 4 ASM- en al 4 C-funksies. Stel 'n tabel op wat die tyd wat elke funksie, en sy ekwivalent neem aandui.

1 Student number functions

In practical 4 and practical 5 we wrote the following four functions:

calculateUniqueNumber added the smallest four numbers, added the largest four number, and then multiplied the answers.

generateChecksum implemented an algorithm to calculate the checksum for a student number.

validateChecksum implemented a modulus operator to check if the checksum is correct.

bubbleSort sorted the numbers in your student number according to their values.

Refer to practical 4 and practical 5's assignments and implement the same four functions, but this time using the C language.

A C file with the incomplete functions is available on learn.sun.ac.za. During setup of your project in e2studio, you will have to add your .asm answer file from practical 5 to the source code directory.

2 Speed difference between ASM and C

In practical 1 we wrote code to turn an LED on and off, and to toggle its state. Add this code to the current project. We want to use the LED to see how long a function takes to execute. To do this, toggle the LED state and call a function, all inside an endless loop. The execution time can be determined by measuring how long the LED is on or off.

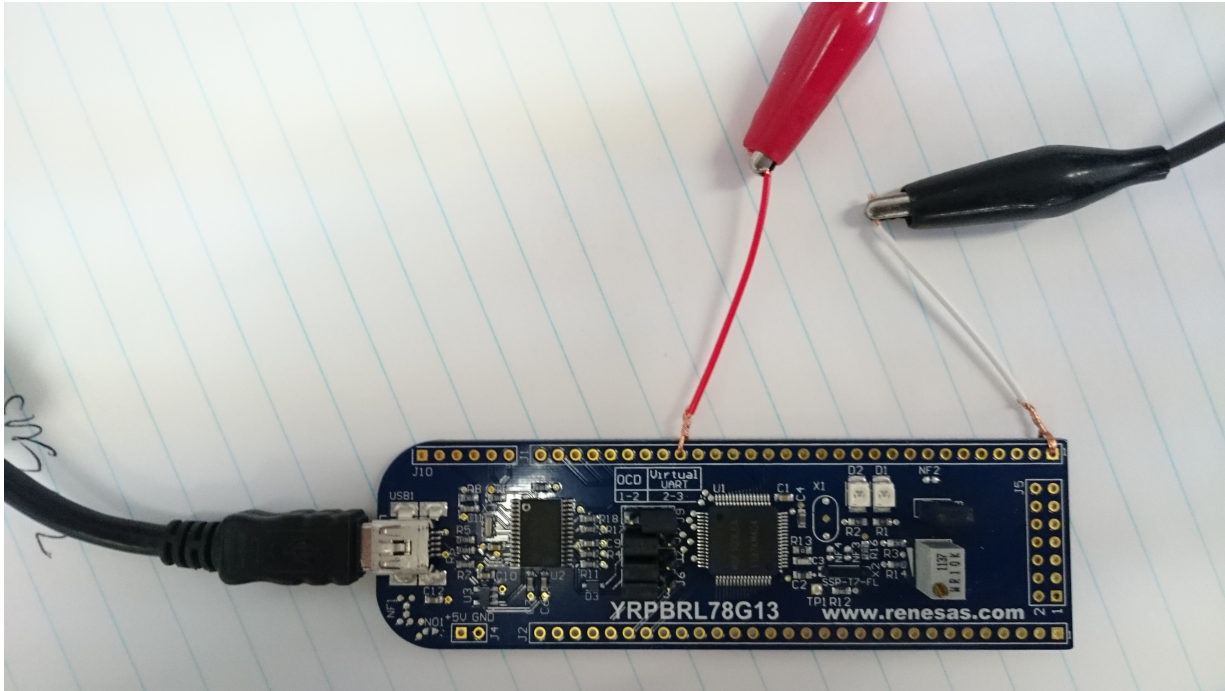
Change your code to toggle the LED in this way. Only measure one function at a time. Use an oscilloscope to measure the precise on and off time of the LED. Repeat this for all 4 assembly and all 4 C functions. Create a table that indicates the time for each function.

Om die ossilloskoop aan die ontwikkelingsbordjie te koppel gaan jy draadjies deur die regte gaatjies op die bord moet plaas, hulle vas te draai en die ossilloskoop se krokodilknypers daaraan te verbind. Sodat e2studio se ontfoutter nie ons eksperiment beïnvloed nie, programmeer eers die bordjie, daarna verwyder dit vanuit die rekenaar se USB poort, en gebruik dan die een op die ossilloskoop om krag aan hom te voorsien. Die program sal nie dadelik uitvoer nie. Jy sal die brugskakelaar J9 op die 2-3 posisie moet plaas. Kyk na die foto hier onder vir hoe die opstelling moet lyk.

Die LED sal baie vinnig aan en af skakel, so dit sal net lyk of dit aan is.

The oscilloscope probes are too big to measure on the board. To make it easier connect wires to the correct holes on the board, twist them to fasten them and measure using the probes. The debugger in e2studio might interfere with the timing of the microcontroller. To avoid this, and to be certain the code is running on the microcontroller, unplug the usb and plug it into the oscilloscope usb port. The program will not start immediately. Move the jumper J9 to position 2-3. See the photo for the entire setup.

The LED will turn on and off very quickly, so it will just look like the LED is on.



Skryf 'n verslag oor jou waarnemings uit hierdie prakties. Handig die verslag as 'n PDF-dokument in op learn.sun.ac.za voor **Donderdag 1 Oktober om 23:55**.

Jou verslag moet ten minste die volgende bevat:

- Die ASM en C kode vir al vier funksies.
- Die tabel van tye hoe lank elke funksie neem.
- 'n Staafgrafiek van die 8 funksies se tydsdure.
- Waarnemings en gevolgtrekkings, met verduidelikings hoekom daar 'n tydverskil tussen die ASM- en C-funksies is. Verwys na prakties 1 se verduideliking hoekom ons ASM per hand geskryf het.
- Vanuit jou waarnemings verduidelik:
 - hoekom sal 'n mens eerder kode in C skryf?
 - wanneer sal 'n mens eerder kode in ASM skryf?

Write a report about your observations in this practical. Hand the report in as a PDF document on learn.sun.ac.za before **Thursday 1 October at 23:55**

The report must contain the following:

- The ASM and C code for all functions.
- The table of execution times for each of the functions.
- A bar graph showing each of the execution times.
- Observations and conclusions with explanations of why there are time differences between the ASM and C functions. Refer to the explanation in practical 1 to explain why we write ASM functions by hand.
- From your explanations explain:
 - Why would you rather code in C?
 - Why would you rather code in assembly?

Verwys na die tydverskil in die uitvoer van die kode, die verskil in aantal lyne kode, en die verskil in tyd wat dit neem om die ASM- en C-kode te skryf. Jy kan ook praat oor die verstaanbaarheid van die kode wanneer iemand anders dit moet lees en interpreteer.

Refer to the time differences in the execution of the code, the number of lines of code, and the difference in time to write the assembly and C code. You can also discuss the readability of the code when someone else has to read and interpret it.