

## Low Level C-programming and microcomputer architecture

### Task 9 CPU performance analyze

#### Versionshistorik

Version	Datum	Ansvarig	Beskrivning
0.0	2013-03-26	LL	Preliminär version
0.1	2013-04-22	ML	Reviderad version
0.2	2013-12-16	LL	Lagt till bilaga C, olika SOF filer
0.3	2014-01-28	LL	Lagt till en loop i testprogrammet
0.4	2015-09-25	LL	Bytt till nyare version av Nios CPU
0.5	2016-08-15	ML	uppdaterad



Figur 1: Utnyttjandegraden av CPU:n

## Innehåll

1 Inledning.....	1
2 Kravspecifikation.....	2
Bilagor .....	4
Bilaga A Testprogram.....	4
Bilaga B Tabell för testfall .....	5
Bilaga C Olika SOF filer för att programmera FPGA:n .....	5

## 1 Inledning

Ingenjören kan nu konstruera enkla datorsystem och har genomfört CASE 3. Ingenjören får ett uppdrag från en kund att prestanda-testa ett program och analysera hur dataarkitekturer påverkar exekveringshastighet. Programmet ska optimeras på olika sätt, sparas i olika typer av minnen och därefter ska mätningar göras på hur lång tid det

tar att exekvera programmet. Analysen ska vara kortfattad och beskriva varför det blir skillnader.

## 2 Kravspecifikation

Följande ska genomföras för att uppgiften ska anses slutförd:

**Tabell 1** Kravspecifikation

Krav	Beskrivning	Utfört Ja/nej
<b>Förstudie</b>		
Krav_001	<p>Gå igenom teorielektionen (Teori_9 och länk*) och skriv en kort sammanfattning vilken ska ingå i rapporten enligt krav_007 (eget kapitel).</p> <p>Beskriv följande begrepp och hur de påverkar prestanda (övergripande):</p> <ul style="list-style-type: none"> <li>- Data and instruction cash</li> <li>- Pipeline</li> <li>- Beskriv gärna andra CPU-mekanismer</li> <li>- Svara på frågan; Vad är det för skillnad när det gäller prestanda (accestid) på internal RAM och external RAM (utanför chippet)?</li> </ul> <p>*<a href="https://www.altera.com/products/processors/support.html">https://www.altera.com/products/processors/support.html</a></p>	
Krav_002	<p>Beskriv de olika HW-arkitekturerna (eget kapitel)</p> <ul style="list-style-type: none"> <li>- 50MHz Economy (HW_CASE_4)</li> <li>- 50 MHz Fast (HW_CASE_4_F)</li> </ul>	
<b>Funktionskrav</b>		
Krav_003	Använd testprogrammet i bilaga 1.	
Krav_004	<p>Följande testfall ska gås igenom med de två olika processorerna (economy och fast):</p> <p><b>Testfall 1</b>  <b>Mjukvara (all programvara):</b> Interna FPGA RAM, enable_reduced_device_drivers, enable_small_c_library.</p> <p><b>Testfall 2</b>  <b>Mjukvara (all programvara):</b> SRAM minnet/Ingen C-kods optimering</p> <p><b>Testfall 3</b>  <b>Mjukvara (all programvara):</b> SRAM minnet/enable_reduced_device_drivers,</p>	



	enable_small_c_library.	
Krav_005	Tabell 1 enligt bilaga 2 ska fyllas i med antal klockcykler och tid.	
Krav_006	Gör kortfattad analys av siffrorna. (eget kapitel)	
<b>Dokumentationskrav</b>		
Krav_007	Sammanfoga dokumentationen från krav_001 till krav_006 till en läsbar rapport med separata kapitel. Framsida med titel, en kort sammanfattning, innehålls-förteckning och kapitel. Lägg även till eventuella slutsatser och referenser.	
<b>Leveranskrav</b>		
Krav_008	Leveransen ska ske till plattformen Itslearning. Leveransen ska vara en rapport. Namnet på filen ska vara "förnamn_efternamn_C_task_9". Sista leveransdag se kursschema (för VG).	



## Bilagor

### Bilaga A Testprogram

```
#include <stdio.h>
#include <altera_avalon_timer_regs.h>

int main(){
    int offset, I;
    int value_d,value_c,value_b,value_a,tids_bruk,time;
    int sum, results_a, results_b, results_c, results_d;

    while(1){

        TIMER_RESET;
        TIMER_START;
        offset=TIMER_READ;

        TIMER_RESET;
        TIMER_START;

        for(I=0;I<100;++I)
        {
            value_a=I;
            value_b= I+5;
            value_c=7;
            value_d=7;
            results_a= value_a*145+value_c*346;
            results_b = value_c*value_a*120;
            results_c = value_d+value_b;
            results_d = value_b*value_a+value_c;
            sum = sum + value_b;
        }

        time=TIMER_READ;
        tids_bruk=time-offset;
        printf("offset = %d\ntime = %d\ntidsbruk = %d\n" ,offset,time,tids_bruk);
    }
    return 0;
}
```

## Bilaga B Tabell för testfall



Mät exekveringstid samt redovisa mätvärden enligt tabell 1 nedan. I de fall resultaten är olika ska detta faktum förklaras (i grova drag). *Economy* ska inte ha någon cash, *standard* ska ha instruktions cash och *fast* ska ha både instruktion och data cash.

Tabell 2: Mätvärden från dina testfall

Testfall	Programmets placering	HW Download	
		Economy ingen cashe <b>HW_CASE_4</b>	Fast Instruktion 4kb och Data 2kb cashe <b>HW_CASE_4_F</b>
1	Interna FPGA RAM/ enable_reduced_device_drivers, enable_small_c_library	<i>Antal Clock cykler (tids_bruk) och absolut tid</i>	
2	SRAM minnet/ Ingen C-kods optimering		
3	SRAM minnet/ enable_reduced_device_drivers, enable_small_c_library		

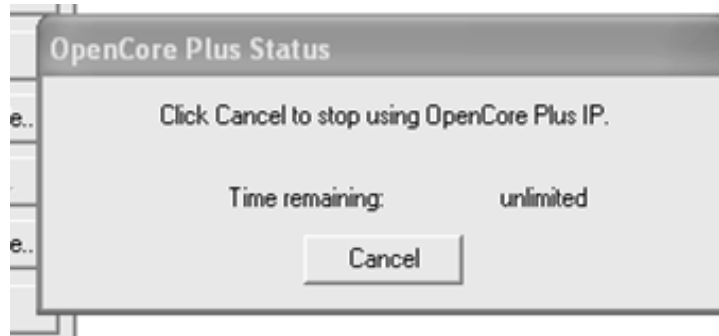
Absolut tid = Antal clock cykler \* period tid (system klockan kopplad till timer är 80 MHz).

## Bilaga C Olika SOF filer för att programmera FPGA'n

 <a href="#">case.sof</a>	2013-05-20 17:45	SOF-fil	3 461 kB
 <a href="#">case_time_limited.sof</a>	2012-10-29 04:39	SOF-fil	3 459 kB

Figur 2: case.sof är NIOS/E och de andra två är case\_time\_limited.sof

Nios/e är licens fri och då programmeras FPGA'n med xxx.sof filen. Nios/f behövs licens för att kunna ladda ner och göra en produkt. Nios/f fungerar bara en bestämd tid, eftersom inuti finns det en timer (väl gömd) som låser CPU 'n efter en bestämd tid. Vidare behövs en uppkoppling med USB blaster och "OpenCore Plus Status" window för Nios/f och Nios/s ska inte stängas ner.



**Figure 3 "Cancel" if this window appears for Nios/e not for Nios/f and Nios/s**

Alla Nios IP komponenter är krypterade, konstruktionen kan inte ses.

Det som inte heller går med de CPU-typer som har licens är att skapa en nätlista av hela konstruktionen, för att exportera konstruktionen till ett annat verktyg. FPGA:n programmeras med `case_time_limited.sof`, eftersom den är skapad före "Netlist generator".

Följande fel i Quartus kan bortses från

Can't generate netlist):

Error (204012): Can't generate netlist output files because the file

"C:/Users/Lennart/Desktop/temp/case\_restored/db/ip/niosII/submodules/niosII\_cpu.v" is an OpenCore Plus time-limited file

Error (204009): Can't generate netlist output files because the license for encrypted file

"C:/Users/Lennart/Desktop/temp/case\_restored/db/ip/niosII/submodules/niosII\_cpu.v" is not available

Error (204012): Can't generate netlist output files because the file

"C:/Users/Lennart/Desktop/temp/case\_restored/db/ip/niosII/submodules/niosII\_cpu.v" is an OpenCore Plus time-limited file

Error (204009): Can't generate netlist output files because the license for encrypted file

"C:/Users/Lennart/Desktop/temp/case\_restored/db/ip/niosII/submodules/niosII\_cpu.v" is not available

Error (204012): Can't generate netlist output files because the file

"C:/Users/Lennart/Desktop/temp/case\_restored/db/ip/niosII/submodules/niosII\_cpu.v" is an OpenCore Plus time-limited file

Error (204009): Can't generate netlist output files because the license for encrypted file

"C:/Users/Lennart/Desktop/temp/case\_restored/db/ip/niosII/submodules/niosII\_cpu.v" is not available