TEIS
Teknisk rapport

TEIS TEKNISK RAPPORT

Författare Lasse Karagiannis	Uppgift Task7
E-post adress lasse.l.karagiannis@gmail.com	Filnamn Lasse_Karagiannis_C_task_7pdf

Code reduction

Lasse Karagiannis

16-10-23

Innehåll

1 KRAVSPECIFIKATION....3

2 Sammanfattning teorilektion 7 (Krav_001)....4

3 Beskriv minnestyperna i ett separat kapitel i rapporten (Krav_004)....4

4 Redovisa resultaten från stegen ovan i ett separat kapitel i rapporten Frågeställning: Varför ”minskar”/”minskar inte” kodmängden?
Gör en kort analys (Krav_005)....4

1 KRAVSPECIFIKATION

Tabell 1 Kravspecifikation

Krav	**Beskrivning**	**Utfört Ja/nej**
	Förstudie	
Krav_001	Skriv en kort sammanfattning av teorilektion 7 (Theory_7) vilken ska ingå i rapporten enligt krav_006.	
	Funktionskrav	
Krav_002	Skapa ett applikationsprojekt med C-programmet i bilaga 2.	
Krav_003	Genomför följande steg och fyll i tabellen som finns i bilaga 1. Steg 5 har alla optimeringarna, alltså bygg på optimeringarna i steg. **Ej reducerad kod** Ingen optimering, utgå ifrån ”hello world”. **Steg 1** I Nios II BSP Editorn/Main/Settings/Advanced aktivering av enable *”reduced_device_drivers”*. **Steg 2** I Nios II BSP Editorn/Main/Settings/Advanced aktivering av *enable_small_c_library* **Steg 3** I Nios II BSP Editorn/Main/Settings/Advanced aktivering av *enable_lightweight_device_driver_api* deaktivering av *enable_c_plus_plus*, *enable_clean_exit* och *enable_exit*. **Steg 4** *printf* i C-koden ersattes med *alt_printf* **Steg 5** I Nios II BSP Propertys sätts *Optimization Level* till *Size*.	
Krav_004	Beskriv minnestyperna i ett separat kapitel i rapporten (krav_007).	
Krav_005	Redovisa resultaten från stegen ovan i ett separat kapitel i rapporten (krav_007). Frågeställning: Varför ”minskar”/”minskar inte” kodmängden? Gör en kort analys.	
	Dokumentationskrav	
Krav_006	Sammanfoga dokumentationen från krav_001 till krav_005 till en läsbar rapport. Framsida med titel, en kort sammanfattning, innehållsförteckning och separata kapitel enligt krav_001 till krav_005 ovan. Lägg även till eventuella slutsatser och referenser.	
	Leveranskrav	
Krav_007	Leveransen ska ske till plattformen Itslearning. Leveransen ska vara en kort rapport. Namnet på filen ska vara ”förnamn_efternamn_C_task_7”. Sista leveransdag se kursschema (för VG).	

2 Sammanfattning teorilektion 7 (Krav_001)

Teoriavsnittet handlade om hur man kan optimera koden med avseende på kodstorlek för Nios II. Sammanfattningsvis kan sägas att verktyget erbjuder möjligheten att kontrollera storleken på .elf-filen. Verktyget erbjuder dessutom optimerade kodbibliotek med alternativa storleksoptimerade rutiner för funktioner såsom printf(), scanf() etc.
Vidare att inte utan vidare ta med funktioner för periferikretsar om de inte används, att inte ta med debug versionens monitor-system när man ska gå till produktion utan run-versionen

3 Genomför följande steg och fyll i tabellen som finns i bilaga 1. Steg 5 har alla optimeringarna, alltså bygg på optimeringarna i steg

Den minnesmängd som behövs efter varje steg har hämtats (xxx.objdump).

Tabell 2 Minnesmängd (exempel, kan minskas ner med minnestyper)

Minnestyp	Ej reducerad kod	steg 1	steg 2	Steg 3	Steg 4	Steg 5
.entry + .exceptions	0x20 + 0x210	0x20 + 0x00	0x20 + 0x00	0x20+ 0x00	0x20+ 0x00	0x20+ 0x00
.text	0xf720	0xe884	0x1dd0	0xdc4	0x61c	0x42c
.rodata	0x300	0x300	0x38	0x0c	0x0c	0x0c
.rwdata	0x1aec	0xaac	0x2d4	0xf0	0x04	0x04
.bss	0x160	0x50	0x10	0x10	0x0c	0x0c
Total summa	0x1189C= 71836	0xF6A0 = 63136	0x210C = 8460	0xEF0 =3824	0x6B4 =1716	0x468 = 1128
Reducering [%] OBS, det är hex!	100%	12.1% enable *"reduced_device_drivers"*	88.2% *enable_small_c_library*	94.7% *enable_lightweight_device_driver_api* deaktivering av *enable_c_plus_plus+ relaterad*	97.6% *alt_printf*	98.4% *Size optimering*

4 Beskriv minnestyperna i ett separat kapitel i rapporten (Krav_004)

Nios II genrerar ett länkar skript som gör fördelningen av minnet i kod och data och detta I olika subsektioner. Exempelvis så innheåller .entry segmentet innehåller adressen till resetvektorn, vilken är en 32 bitars address.
.exceptions är segmentet innehållande interrupt relaterad information avseende cpu:n och dess periferikretsar. Troligtvis innehåller den basadressen för interruptvektorerna, samt jmp instruktioner till de minnespositioner där länkaren själva interrupt-koden.
.text innehåller startadressen för applikationskoden och koden självt
.rodata innhåller information om konstanter dess minnesadresser i dataminnet. Read-only data.
.rw innehåller information om variabler och dess placeringar i minnesrymden. Read-and-Write data.
.bss sektionen innehåller information om variabler och minnesadresser som ska initieras till noll.

5 Redovisa resultaten från stegen ovan i ett separat kapitel i rapporten Frågeställning: Varför ”minskar”/”minskar inte” kodmängden? Gör en kort analys (Krav_005)

Man ser att ju man skalar av koden desto mindre blir .elf-filen. Aktivering av reduced_device-drivers gav en minsking av koden med 12.1%, därefter aktiverade vi small_c_library som gaven en minnesreduktion på totalt 88%, och som ensamt alltså bidrar med en minsking på 67%. Därefeter aktiverar vi lightweight device-drivers, vilket inte gör så stor skillnad då vi inte använder särsklit många perfiferi-enheter PIO-koden är t.ex. Bortkommenterad, så den enda device-drivern som optimeras måste vara JTAG. Borttagandet av stöd för C++ gav en reduktion av minnesutrymmet, förmodligen därför att C++ använder sig av dynamiska objekt till väldigt mycket, som naturligtvis måste beredas plats på heapen. Användning en alt_printf istället för printf sparade oss sedan ca 2kB, därefeter gjorde kompilatorn sitt med utrymmes optimeringen.

6 VG-uppgift

“fatal error: altera_avalon...h: No such file”
Lösning:
1) Finns altera_avalon….h i din BSP?

2) Finns det någon PIO i din adress map&Akitektur?
3) Vad anser du att du ska göra?
Svar altera_avalon_pio-regs.h finns inte så jag kopierar in den I mappstrukturen. Vet var den ska finnas genom att studera ett projekt mapp där PIO fungerar. Kopierar över system.h och bygger koden. Kör till synes utan några fel. Har tagit bort minnesoptimering.