Termin 4

WS2012

LOAD, STORE, bedingte Befehle, Speicherbereiche





HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES



RECHNERARCHITEKTUR

WS2012

Termin 4

LOAD, STORE, bedingte Befehle, Speicherbereiche

Name, Vorname	Matrikelnummer	Anmerkungen
Datum	Raster (z.B. Mi3x)	Testat/Datum

Legende: V:Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

h-da / fbi / I-PST Termin4WS2012.odt 06.08.2012 gedruckt: 09.02.10 1 / 6

Termin 4LOAD, STORE, bedingte Befehle, Speicherbereiche

Ziele:

Verständnis für LOAD und STORE Befehle, bedingte Befehle und die verschiedenen Speicherbereiche. Ziel ist die Implementierung mit möglichst geringer Codegröße sowie der Umgang mit einem Debugger/Simulator und der Entwicklungsumgebung.

Arbeitsverzeichnis:

Kopieren Sie sich das Verzeichnis, welches Ihnen im Praktikum zur Verfügung gestellt wird, in Ihr persönliches Verzeichnis. Dort stehen Ihnen dann alle benötigten Dateien zur Verfügung.

Vorbereitung

Arbeiten Sie sich in folgende Befehle des ARM-Prozessors ein:

Instruktion	Bedeutung
ADDNE R1, R2, #1	R1 := R2 + 1, falls das Z-Bit im Prozessorstatuswort nicht gesetzt ist
LDR R1, [R2]	R1 := $mem_{32}[R2]$
LDREQ R1, [R2]	R1 := mem ₃₂ [R2], falls das Z-Bit im Prozessorstatuswort gesetzt ist
LDRB R1, [R2]	R1 := mem ₈ [R2]
STR R1, [R2]	mem ₃₂ [R2] := R1
STRB R1, [R2]	mem ₈ [R2] := R1
ADR R1, Marke	R1:=PC+(Offset zur Marke)
B Marke	PC wird auf Adresse der Marke gesetzt
BEQ Marke	PC wird auf Adresse der Marke gesetzt, falls das Z-Bit im
	Prozessorstatuswort gesetzt ist
BNE Marke	PC wird auf Adresse der Marke gesetzt, falls das Z-Bit im
	Prozessorstatuswort nicht gesetzt ist

Aufgabe 1:

Auf welchen Adressen wird der Inhalt von Register r1 gespeichert? Ergänzen Sie sie Kommentarzeilen.

mov	r0, #0	
str	r1, [r0], #4	// Inhalt von r1 auf Adresse 0x danach steht in r0 0x
eor	r0, r0, r0	
str	r1, [r0, #4]	// Inhalt von r1 auf Adresse 0x danach steht in r0 0x
mov	r0, #0	
str	r1, [r0]!	// Inhalt von r1 auf Adresse 0x danach steht in r0 0x
sub	r0, r0, r0	
str	r1, [r0, #4]!	// Inhalt von r1 auf Adresse 0x danach steht in r0 0x
and	r0, r0, #0	
strb	r1, [r0, #1]!	// Inhalt von r1 auf Adresse 0x danach steht in r0 0x
mov	r1, #4	
strb	r1, [r0, r1]!	// Inhalt von r1 auf Adresse 0x danach steht in r0 0x

h-da / fbi / I-PST Termin4WS2012.odt 06.08.2012 gedruckt: 09.02.10 2 / 6

Fb Informatik Technische Fachgruppe

Termin 4LOAD, STORE, bedingte Befehle, Speicherbereiche

Aufgabe 2:

Bearbeiten Sie schriftlich die Zusatzfragen.

- a) Auf welche Weise kann man die Condition-Code-Flags NZCV (Bedingungsbits) des Prozessorstatuswort (CPSR) setzen?
- b) Wie wird die Pseudoinstruktion "ADR R1, Marke" vom Assembler umgesetzt? Schreiben Sie hierzu den Befehl in einen der vorgegebenen Programmrahmen und schauen Sie ihn sich im Debugger in der Mixed-Darstellung an (rechts oben → Source auswählen). Vollziehen Sie die Umsetzung des Compiler nach und informieren Sie sich auch über Pipelining.
- c) Das Prozessorstatuswort hat den Wert 0xa00013, wenn der Befehl "BNE Marke" ausgeführt wird. Würde dann der Sprung an die (symbolische) Adresse Marke ausgeführt? Weisen Sie Ihre Antwort mit einem Programm nach.

Aufgabe 3:

Es ist ein Programm zu entwickeln, welches alle Werte einer Liste1 nach Liste2 kopiert. In den Listen steht an erster Stelle die Anzahl der Elemente der jeweiligen Liste. Liste1 ist, bis auf die erste Zahl (Anzahl der Elemente max. 255) eine Liste mit 8Bit großen vorzeichenbehafteten Zahlen (-128 bis +127). In Liste2 sollen die Zahlen aus Liste1, ausser die Anzahl der Elemente (die bleibt vorzeichenlos), als 32Bit große vorzeichenbehaftete Zahlen abgelegt werden.

Aufgabe 4:

Nach dem Kopiervorgang soll in einem weiteren Schritt die Liste2 aufsteigend sortiert werden. Hierzu erweitern Sie Ihr Programm von Aufgabe 3. Es gibt verschiedene Sortieralgorithmen (z.B. Bubblesort). Denken Sie daran, dass die Länge der Liste an erster Stelle unverändert stehen bleiben muss.

Bericht

Der erforderliche Praktikumsbericht dient zu Ihrer Nachbereitung des Praktikums und wird stichprobenhaft überprüft. Er beinahltet auch den zeilenweisen kommentierten Quelltext. Haben Sie Ihre Berichte zu den Praktikumsterminen dabei.

h-da / fbi / I-PST Termin4WS2012.odt 06.08.2012 gedruckt: 09.02.10 3 / 6

Fb Informatik Technische Fachgruppe

// End of File

Termin 4LOAD, STORE, bedingte Befehle, Speicherbereiche

WS2012

```
// Name:
                    Matrikelnummer:
// Name:
                    Matrikelnummer:
// Datum:
                    "aufgabe1.S"
          .file
                              @ legt eine Textsection fuer PrgrammCode + Konstanten an
          .text
          .align
                              @ sorgt dafuer, dass nachfolgende Anweisungen auf einer durch 4 teilbaren Adresse liegen
                                        @ unteren 2 Bit sind 0
                                        @ nimmt das Symbol main in die globale Sysmboltabelle auf
          .global
                    main
                    main,function
          .type
main:
                    r0, #0
          mov
                    r1, [r0], #4
                                        // Inhalt von r1 auf Adresse 0x____ danach steht in r0 0x_
          str
          eor
                    r0, r0, r0
                    r1, [r0, #4]
                                        // Inhalt von r1 auf Adresse 0x____ danach steht in r0 0x_
          str
          mov
                    r0, #0
                    r1, [r0]!
                                        // Inhalt von r1 auf Adresse 0x____ danach steht in r0 0x_
          str
          sub
                    r0, r0, r0
                    r1, [r0, #4]!
                                        // Inhalt von r1 auf Adresse 0x____ danach steht in r0 0x__
          str
          and
                    r0, r0, #0
          strb
                    r1, [r0, #1]!
                                        // Inhalt von r1 auf Adresse 0x
                                                                          danach steht in r0 0x
          mov
                    r1, #4
          strb
                    r1, [r0, r1]!// Inhalt von r1 auf Adresse 0x_____ danach steht in r0 0x_
          bx
.Lfe1:
                    main,.Lfe1-main
          .size
```

h-da / fbi / I-PST Termin4WS2012.odt 06.08.2012 gedruckt: 09.02.10 4 / 6

Fb Informatik Technische Fachgruppe

// End of File

Termin 4LOAD, STORE, bedingte Befehle, Speicherbereiche

WS2012

// Name: Matrikelnummer: // Name: Matrikelnummer: // Datum: // .file "aufgabe2.S" .text @ legt eine Textsection fuer PrgrammCode + Konstanten an .align @ sorgt dafuer, dass nachfolgende Anweisungen auf einer durch 4 teilbaren Adresse liegen @ unteren 2 Bit sind 0 .global @ nimmt das Symbol main in die globale Sysmboltabelle auf main .type main,function main: bx lr .Lfe1: .size main,.Lfe1-main

h-da / fbi / I-PST Termin4WS2012.odt 06.08.2012 gedruckt: 09.02.10 5 / 6

Fb Informatik Termin 4

Liste1Ende:

// End of File

// .comm-Section für nicht initialisierte Daten

.comm

Technische Fachgruppe LOAD, STORE, bedingte Befehle, Speicherbereiche

// Name: Matrikelnummer: // Name: Matrikelnummer: // Datum: .file "aufgabe3.S" .text @ legt eine Textsection fuer PrgrammCode + Konstanten an .align 2 @ sorgt dafuer, dass nachfolgende Anweisungen auf einer durch 4 teilbaren Adresse liegen @ unteren 2 Bit sind 0 .global @ nimmt das Symbol main in die globale Sysmboltabelle auf main .type main,function main: sp!, {r4, r5, lr} @ Ruecksprungadresse und Register sichern stmfd kopieren: @ hier Ihr Programm zum Kopieren einer Byte-Tabelle (je 8Bit) in eine Word-Tabelle (je 32Bit) einfuegen ... vorzeichen: @ hier Ihr Programm um die 8Bit-Zahlen auf vorzeichenrichtige 32Bit-Zahlen zu wandeln ... sortieren: @ hier Ihr Programm um die vorzeichenrichtige Zahlen in Liste2 zu sortieren fertig: ldmfd sp!, {r4, r5, pc} @ Ruecksprungadresse und Register TAB2: .word Liste2 @ Beispiel um an Adressen aus anderen Segmenten zu kommen .Lfe1: main,.Lfe1-main .size // .data-Section für initialisierte Daten // Erster Wert der Tabelle steht fuer die Anzahl (max. 64) der Werte der Tabelle Liste1: .byte (Liste1Ende-Liste1), -9, 8, -7, 6, -5, 4, -3, 2, -1, 0, 127, 128

WS2012

h-da / fbi / I-PST Termin4WS2012.odt 06.08.2012 gedruckt: 09.02.10 6 / 6

Liste2, (Liste1Ende-Liste1)*4) @ Speicherbereich mit der Groesse*4 von Liste1 reservieren