

Rechnersysteme SS 2009

Aufgabenblatt 6

Vorzustellen vom 2.–8. Juni

Aufgabe 1

Schreiben Sie ein 68k-Assemblerprogramm, das für positive Ganzzahlen in einem benutzerdefinierten Intervall feststellt, welches die Primzahlen darin sind. Die obere und untere Grenze des Intervalls soll bei Programmstart vom Nutzer eingegeben werden. Geben Sie die Primzahlen am Bildschirm aus. Nutzen Sie den Trap-#15-Befehl.

Aufgabe 2

Nachfolgend finden Sie das Listing eines Assemblerprogramms in M68K-Assembler. Veranschaulichen Sie den Aufbau und die Belegung des Stacks einschließlich der zugehörigen Adressen. Berücksichtigen Sie dabei auch Stack-Veränderungen durch die Programmierung. Hinweis: Entkommentieren Sie die Zeile nach dem Symbol START entsprechend des von Ihnen genutzten Simulators.

```
*-----*
* Program      : stackpointer
* Description:  analyse der belegungen des stack
*-----*

                ORG      $0                Prozessor Init

                DC.L     $4010             Stackpointer initialisieren
                DC.L     START            Adr. für 1. ausführbaren Befehl

                ORG      $2000            Startadresse für Programmcode

START:
*              MOVE.L   #$4010,SP         START mit EASY68k
                MOVE.L   #1,D0           Testwert für case
                MOVE.L   D0,-(A7)        push c
                BSR      CASE            UP Case
                MOVE.L   (A7)+,D1        pop rückgabeparameter
                STOP     #$2700

CASE           MOVE.L   D0,-(A7)         benutzte Register sichern
                MOVE.L   A0,-(A7)
                MOVE.L   12(A7),A0       c nach A0
                MOVE.L   #0,D0           Initialisieren mit 0, Wertebereich grenz.
                CMP.L    #11,A0          Begrenzung des Wertebereichs
                BLS.S    USETAB          wenn c<=11 dann USETAB
                MOVE.L   D0,12(A7)       Ergebnis zurückgeben
                MOVE.L   (A7)+,A0        Register wiederherstellen
                MOVE.L   (A7)+,D0
                RTS

USETAB        MOVE.B    Tab(A0),D0       Zurückspringen
                MOVE.L   D0,12(A7)       Tabellenwert an index c, zurückgeben
                MOVE.L   (A7)+,A0        Ergebnis zurückgeben
                MOVE.L   (A7)+,D0        Register wiederherstellen
                RTS

                ORG      $1000           Zurückspringen
                DC.B     0,7,3,0,0,0     Startadresse für Daten
                DC.B     0,2,0,0,0,12    Look-up-Tabelle
                END     START
```

Aufgabe 3

Wozu dient ein Sequenzer? Welche Art von Flipflop findet in diesem Anwendung?
Zu welchem Zweck erhält dieses Flipflop ein Reset-Signal? An welcher Stelle wird es ausgelöst?

Aufgabe 4

Nennen und erläutern Sie Designprinzipien, die mit der RISC-Prozessorarchitektur entstanden!

Aufgabe 5

Welcher Lösungsweg wurde beim Entwurf von RISC-Systemen eingeschlagen, um mit der Problematik einer hohen Schachtelungstiefe von Unterprogrammen, einer hohen Anzahl von Variablen und einer Begrenztheit an Registern umzugehen?