

## Fachhochschule Esslingen – Hochschule für Technik

Wintersemester	2005/2006	Blatt-Nr.:	1 von 7
Fachbereich:	IT/NT	Semester:	NT 5
Prüfungsfach:	Rechnertechnik 2	Fachnummer:	5074
Hilfsmittel:	alle	Zeit:	90 Minuten

### Aufgabe 1 (30 Punkte):

a) Beschreiben Sie die fünf wesentlichen Komponenten eines jeden Computersystems:

Lösung zu Aufgabe 2a):

- CPU mit
  - Datenpfad (Rechenwerk, ALU)
  - Steuerung (Control, Steuerwerk)
- Speicher (Memory)
- Eingabe und Ausgabe (Input/Output, Peripherie)
- Adress-, Daten- und Steuerbussystem

b) Beschreiben Sie fünf mögliche Adressierungsarten eines 68HCS12-Rechners und geben Sie für jede Variante ein Beispiel:

Lösung zu Aufgabe 2b):

#### Bezeichnung

#### Beispiel

- |  |                    |
|--|--------------------|
| • Unmittelbare Adressierung (Immediate)                | LDAA #0            |
| • Implizite Adressierung                               | CLRA               |
| • Registeradressierung                                 | TFR B, A           |
| • Direkte Adressierung                                 | STAA variable      |
| • Register-Indirekte Adressierung                      |                    |
| Mit Index (mit Offset)                                 | STAA 0, X          |
| Mit Prä/Post-Inkrement/Dekrement                       | STAA 1, X+         |
| • Speicher-Indirekte-Adressierung                      |                    |
| Mit Index  | STAA [# \$1000, X] |
| • Relative Adressierung                                | BRA start          |
| (sieht für Programmierer aus wie direkte Adressierung) |                    |

## Fachhochschule Esslingen – Hochschule für Technik

Wintersemester	2005/2006	Blatt-Nr.:	2 von 7
Fachbereich:	IT/NT	Semester:	NT 5
Prüfungsfach:	Rechnertechnik 2	Fachnummer:	5074
Hilfsmittel:	alle	Zeit:	90 Minuten

c) Welche Funktion hat ein Compiler? Bitte ausführlich erklären.

Lösung zu Aufgabe 2c):

Übersetzen des Programmcodes von Hochsprache (z.B. C) in hexadezimal codierte Maschinensprache (Objektcode) bzw. der Daten in Speicherplatzreservierungen und Initialisierung des Speicherplatzes.

d) Welche Funktion hat ein Linker? Bitte ausführlich erklären.

Lösung zu Aufgabe 2d):

Zusammenbinden des Objektcodes verschiedener Programmmodule sowie Bibliotheken (dabei Auflösung von Referenzen) zu relocierbarem Objektcode.

Konvertierung des relocierbaren Objektcodes in absoluten Objectcode, dh. Zuweisen von absoluten Adressen, erfolgt durch den Locator (bei Embedded Systems, Locator ist bei manchen Toolketten Teil des Linkers) oder durch den Lader des Betriebssystems (bei Computern).

## Fachhochschule Esslingen – Hochschule für Technik

Wintersemester	2005/2006	Blatt-Nr.:	3 von 7
Fachbereich:	IT/NT	Semester:	NT 5
Prüfungsfach:	Rechnertechnik 2	Fachnummer:	5074
Hilfsmittel:	alle	Zeit:	90 Minuten

- e) Erläutern Sie den Unterschied zwischen einer von-Neumann-Architektur und einer Harvard-Architektur.

Lösung zu Aufgabe 2d:

Von-Neumann:

Programmcode und Daten im selben Speicher.  
Zugriff über ein gemeinsames Bussystem.

Harvard:

Programmcode und Daten in getrennten Speichern.  
Zugriff über getrennte Bussysteme.

- f) Erläutern Sie die Funktion der I, N-, Z- und V-Flags im CCR des 68HCS12.

Lösung zu Aufgabe 2f:

I=1 Interrupts gesperrt (maskiert)  
N=1 Ergebnis (als 2er-Komplement-Zahl interpretiert) einer Operation war negativ  
Z=1 Ergebnis einer Operation war Null  
V=1 Überlauf bei einer Operation (Werte als 2er-Komplement-Zahl interpretiert)

## Fachhochschule Esslingen – Hochschule für Technik

Wintersemester	2005/2006	Blatt-Nr.:	4 von 7
Fachbereich:	IT/NT	Semester:	NT 5
Prüfungsfach:	Rechnertechnik 2	Fachnummer:	5074
Hilfsmittel:	alle	Zeit:	90 Minuten

### Aufgabe 2 (35 Punkte):

Im folgenden Ausschnitt aus einem Assemblerprogrammlisting für einen Freescale 68HCS12-Rechner (Codewarrior-Entwicklungsumgebung und Aufrufkonventionen) sehen Sie ein Unterprogramm mit einem 16-Bit Integer-Wert als Parameter und einem 16-Bit Integer-Wert als Rückgabewert. Sie sollen herausfinden, was es macht.

MYFUNCTION:		<code>; unsigned int MYFUNCTION(unsigned int n)</code>	
STD 4, -SP		<code>; Lokale Variable: unsigned int k = n (A)</code>	
LDD #-1		<code>; Lokale Variable: unsigned int N = -1</code>	
STD 2, SP			
LDX 0, SP		<code>; im Unterprogramm: k → SP+0, N → SP+2</code>	
1: BNE else1		<code>; if (k==0)</code>	
NEGB		<code>; { k = +1;</code>	
CLRA			
STD 0, SP			
else1:		<code>; }</code>	
LDX 0, SP		<code>; if (k&lt;=8)</code>	
DEX			
CPX #7			
BHI return1			
LDAB #1		<code>; { N = 1;</code>	<code>Reg. D = 1</code>
CLRA			
LDY 0, SP			<code>Reg. Y = k</code>
dowhile1:		<code>; do</code>	
TFR Y, X		<code>; {</code>	<code>Reg X = Y</code>
EMUL		<code>; N = N * k;</code>	<code>(Y,D) = D x Y</code>
DEX		<code>; k = k - 1;</code>	<code>Reg X = X - 1</code>
CPX #1			
TFR X, Y			<code>Reg. Y = X</code>
BGT dowhile1		<code>; } while (k &gt; 1) ;</code>	<code>Reg. X &gt; 1</code>
STD 2, SP		<code>; N = Reg. D</code>	
return1:		<code>; }</code>	
LDD 2, SP		<code>; return N;</code>	
2: LEAS 4, SP		<code>; Lokale Variable vom Stack entfernen</code>	
RTS			

- a) An das Unterprogramm wird ein Parameter übergeben. Wie wird dieser Parameter übergeben?

Antwort:

**Übergabe n im Register D, wird bei (A) in lokale Variable auf Stack kopiert**

- b) Wie wird das Ergebnis des Unterprogramms an den Aufrufer zurückgegeben?

Antwort:

**Rückgabewert N in Register D (siehe Zeile nach Label return1)**

- c) Welcher Wert steht im X-Register, wenn der Programmzeiger auf Label 1: zeigt und der Parameterwert n=6 übergeben wurde?

Antwort (nächste Seite):

## Fachhochschule Esslingen – Hochschule für Technik

Wintersemester	2005/2006	Blatt-Nr.:	5 von 7
Fachbereich:	IT/NT	Semester:	NT 5
Prüfungsfach:	Rechnertechnik 2	Fachnummer:	5074
Hilfsmittel:	alle	Zeit:	90 Minuten

**X = 6 = k = n**

- d) Die Funktion soll mit dem Parameterwert 3 aufgerufen werden. Zeichnen Sie den Stack mit allen Werten (numerisch wenn möglich, ein Byte pro Reihe), wenn der Programmzeiger auf Label 2: zeigt.

Adresse	Wert	Bedeutung
\$2FFF	PC LSB	Return-Adresse vom ersten Call von MYFUNCTION
\$2FFE	PC MSB	
<b>\$2FFD</b>	<b>\$06 (LSB)</b>	<b>Lokale Variable unsigned int N = n!</b>
<b>\$2FFC</b>	<b>\$00 (MSB)</b>	
<b>\$2FFB</b>	<b>\$03 (LSB)</b>	<b>Lokale Variable unsigned int k = n (Kopie des Parameters n)</b>
<b>\$2FFA</b>	<b>\$00 (MSB)</b>	<b>(im Unterprogramm SP+0)</b>

- e) Schreiben Sie auf ein separates Blatt die C-Routine, die zum oben gezeigten Assembler-Programm gehört.
- f) Geben Sie hier für die folgenden Parameterwerte die Ergebnisse des Unterprogramms an:

Parameterwert:	Ergebnis
0 .....	<b>N = 0! = 1</b>
-6 .....	<b>N = (unsigned int) -1 = 65535</b>
6 .....	<b>N = 6! = 720</b>
10 .....	<b>N = (unsigned int) -1 = 65535</b>

- g) Geben Sie den gültigen Wertebereich für den Parameter an, innerhalb dessen das Ergebnis korrekt dargestellt wird. Was passiert außerhalb dieses Bereiches?  
Antwort:

**-1 < n < 9 ; außerhalb: N = (unsigned int) -1 = 65535**  
**d.h. n = 0 ... 8 → N = n! = 1 ... 40320 (9!=362 880 nicht mehr mit 16bit darstellbar)**

## Fachhochschule Esslingen – Hochschule für Technik

Wintersemester	2005/2006	Blatt-Nr.:	6 von 7
Fachbereich:	IT/NT	Semester:	NT 5
Prüfungsfach:	Rechnertechnik 2	Fachnummer:	5074
Hilfsmittel:	alle	Zeit:	90 Minuten

### Aufgabe 3 (35 Punkte):

Der Beeper auf unserem Lautsprecherboard hängt am Ausgang des Timers Kanal 5. In dieser Aufgabe sollen Sie in der Assemblersprache die notwendige Software erstellen, so dass beim Drücken auf den Taster SW1 an Port H der Beeper ertönt. Dazu gehen Sie wie folgt vor:

- a) Sie schreiben eine Prozedur `initBeeper`, die den Timer initialisiert. Dazu müssen Sie die Register `TSCR1`, `TIOS` und `TIE` geeignet beschreiben. Alle anderen Register spielen zunächst keine Rolle. Der Timer wird im Output Compare Mode betrieben.

Lösung zu Aufgabe 3a):

```
initBeeper:
    BSET TSCR1, #$80      ; Enable timer module

    MOVB #$07, TSCR2      ; Timer clock period  $T_{TCNT,max} = 5,3\mu s$ 
                          ; (nicht gefordert, Default-Wert 42ns)

    BSET TIOS, #$20       ; Set timer channel 5 to output
                          ; compare mode

    BSET TIE, #$20        ; Enable interrupt for timer ch. 5

    RTS
```

- b) Schreiben Sie zwei Routinen `beeperOn` und `beeperOff`, die den Ausgang des Timer-Kanals 5 aktivieren bzw. stilllegen. Dazu verwenden Sie lediglich das Register `TCTL1`. Alle anderen Timer-Ausgänge dürfen nicht verändert werden! Setzen Sie den Ausgang so, dass er toggelt, d.h. zwischen 0 und 1 wechselt.

Lösung zu Aufgabe 3b):

```
beeperOn:
    BCLR TCTL1, #%00001000 ; optional, da Default-Wert Bit 3 = 0
    BSET TCTL1, #%00000100 ; On next timer 5 event toggle
    RTS

beeperOff:
    BCLR TCTL1, #%00001100 ; On timer 5 event no output
    RTS
```

## Fachhochschule Esslingen – Hochschule für Technik

Wintersemester	2005/2006	Blatt-Nr.:	7 von 7
Fachbereich:	IT/NT	Semester:	NT 5
Prüfungsfach:	Rechnertechnik 2	Fachnummer:	5074
Hilfsmittel:	alle	Zeit:	90 Minuten

- c) Jetzt erstellen Sie die Interrupt-Routine. Diese inkrementiert TC5 immer um einen bestimmten Wert, der die Frequenz festlegt. Wird der Wert vom Zeitgeber erreicht, wird ein Interrupt erzeugt. Die Interrupt-Routine muss also nur den Vergleichswert in TC5 um ein Delta erhöhen und das Interrupt-Flag zurücksetzen.

Lösung zu Aufgabe 3c):

```
.vect: SECTION
      ORG $FFE4
int13: DC.W timer5Isr      ; Interrupt vector for timer channel 5

period: EQU 214            ; Signal period  $2 * 214 * 5,3\mu s = 2,27ms$ 
                          ; (ca.440Hz), in Aufgabe nicht vorgegeben

.init: SECTION
timer5Isr:
      LDD #period          ; Add period
      ADDD TC5              ; Set timer for next interrupt
      STD TC5

      BSET TFLG1, #$20      ; Reset interrupt flag for timer ch. 5
      RTI
```

- d) Jetzt schreiben Sie das Hauptprogramm. Zunächst schalten Sie alle Interrupts aus. Sie setzen Port H als Eingang und schalten seine Interrupts aus.

Lösung zu Aufgabe 3d):

```
Main:
Entry:
      LDS #__SEG_END_SSTACK; Initialize stack pointer
      SEI                      ; Disable interrupts
      MOVB #$00, DDRH          ; Port H as inputs (optional, da Default)
      MOVB #$00, PIEH          ; Port H interrupts off ( - " - )
```

- e) Nun initialisieren Sie den Beeper-Timer und schalten die Interrupts wieder ein. Danach gehen Sie in eine Endlosschleife, in der Sie beim Drücken der Taste SW1 den Beeper ertönen lassen.

Lösung zu Aufgabe 3d):

```
      JSR initBeeper
      CLI

Loop:
      BRSET PTH, $01, off      ; Button not pressed = 1
      JSR beeperOn             ; Turn beeper on, if button pressed
      BRA Loop

Off:
      JSR beeperOff            ; Turn beeper off, if button not pressed
      BRA Loop
```