

<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

**Unterschrift:** \_\_\_\_\_

**Als Hilfsmittel sind nur zugelassen:**

- 1 doppelseitig handbeschriebenes DIN A4 -Blatt
- Schreibstifte

**Das Deckblatt ist mit Angabe des Namens, Matrikelnr. und der Unterschrift abzugeben. Alle zusätzlichen Blätter, die abgegeben werden, sind mit Namen und Matrikelnr. zu kennzeichnen.**

Note	Punkte	Aufgabe	Mögliche Punkte	Erreichte Punkte
1.0	70	1	8	
1.3	66	2	8	
1.7	62	3	12	
2.0	58	4	12	
2.3	53	5	7	
2.7	48	6	6	
3.0	43	7	6	
3.3	38	8	10	
3.7	34	9	4	
4.0	30	10	6	
		Summe	79	

Punkte: \_\_\_\_\_ Note: \_\_\_\_\_

<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

### Aufgabe 1:

8 Punkte

Übersetzen Sie folgende beiden C-Programme in ein codeoptimiertes Assemblerprogramm:

a)

```

int main(void) {
    int x = 1;
    int y = 2;
    return(0);
}

```

(2 Punkte)

Lösung:  
**main:**

---



---

b)

```

int main(void) {
    volatile int x = 1;
    volatile int y = 2;
    return(0);
}

```

(6 Punkte)

Lösung:  
**main:**

```

sub    sp, sp, #8

```

---



---



---



---



---



---



---

```

mov    pc, lr

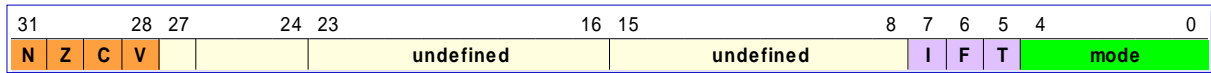
```

<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

## Aufgabe 2:

8 Punkte

Das Current Program Status Register (CPSR) hat folgende Struktur:



Benutzen Sie unter anderem die folgenden Befehle msr und mrs um die beiden Teilaufgaben zu lösen.

Syntax:

```

msr{<cond>} <psr>_<feld>, Rm
msr {<cond>} <psr>_<feld>, #<immediate>
<psr> = CPSR oder SPSR
<feld> = Kombination von f, s, x, c
mrs{<cond>} Rd, <psr>
<psr> = CPSR oder SPSR
  
```

a)

Setzen des Zero Flags auf den Wert 1 (ohne die anderen N,C,V Flags zu verändern). Das Flag Feld (Bits 24-31 vom CPSR) kann mit CPSR\_f angesprochen werden.

(4 Punkte)

Lösung:

---



---



---

b)

Prozessor umschalten vom IRQ Mode (Bitmuster 10010) in den USER Mode (Bitmuster 10000), ohne die I,F,T Bits zu verändern. Das Kontroll Feld (Bits 0-7 vom CPSR) kann mit CPSR\_c angesprochen werden.

(4 Punkte)

Lösung:

---



---



---



---

<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

### Aufgabe 3:

12 Punkte

a)

Welche Werte haben die Register R0-R3, R9 jeweils nach Ausführung der folgenden Befehle. Tragen Sie die Lösung in die Tabelle jeweils in den Zeilen mit dem stm und ldm Befehl ein. Geben Sie die Register R0 – R4 dezimal und das Register R9 hexadezimal an.

(10 Punkte)

Befehl	R0	R1	R2	R3	R9
Initialisierung	0	1	2	3	0x100
stmda r9!, {r0-r2}					
ldmib r9, {r2,r3}					
Initialisierung	0	1	2	3	0x100
stmdb r9!, {r0-r3}					
ldmib r9!, {r0-r2}					

b)

Welche Stackbefehle würde man vor dem Rücksprung aus einer Funktion nutzen, um den Stack korrekt abzubauen, wenn beim Eintritt in diese Funktion folgender Stackbefehl benutzt wurde. Tragen Sie die Lösung in die Tabelle ein:

(2 Punkte)

nach dem Eintritt in die Funktion	vor dem Austritt aus der Funktion
stmdb sp!, {r0, r1, r2, lr}	
stmfd sp!, {r0-r4, r7, r9, lr}	

<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

#### Aufgabe 4:

12 Punkte

Beantworten Sie folgende Fragen zur Prozessorarchitektur:

a) Was versteht man unter einer LOAD/STORE Architektur ?

(2 Punkte)

---



---

b) Aus wieviel Stufen besteht die Pipeline des Prozessors und wie werden diese Stufen bezeichnet ?

(3 Punkte)

---



---

c) Wieviel Prozessorzyklen benötigt ein Programm ausgeführt, das aus 5 Datenverarbeitungsbefehlen (add, sub, mov) besteht ?

(2 Punkte)

---



---



---

d) Wieviel Speicher benötigt ein Befehl ?

(1 Punkt)

---

e) Aus wievielen Registern besteht ein Registersatz (ohne Statusregister) ? Welches Register ist der Program Counter (pc), Stack Pointer (sp) und das Link Register (lr) ?

(2 Punkte)

---



---

f) Was versteht man unter bedingter Ausführung von Befehlen ? Geben Sie ein Beispiel mit Beschreibung an ?

(2 Punkte)

---



---



---

<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

### Aufgabe 5:

7 Punkte

Eine Make-Datei mit dem Dateinamen *makefile1* hat folgenden Inhalt.

*FILE = Termin3Aufgabe1*

*Opti = 1*

*all:*

```

arm-elf-gcc -c ../boot/swi.S
arm-elf-gcc -c ../boot/boot.s
arm-elf-gcc -g -c -O$(OPTI) $(FILE).c
arm-elf-gcc -S -O$(OPTI) $(FILE).c
arm-elf-ld -Ttext 0x2000000 swi.o boot.o $(FILE).o -o $(FILE).elf

```

Wie wird diese Make-Datei von der Konsole aus aufgerufen ?

(1 Punkt)

Lösung:

---

Nennen Sie die Dateien, welche nach einer fehlerfreien Ausführung erzeugt wurden.

(5 Punkte)

Lösung:

---



---



---



---



---



---

Was bewirkt die Angabe von *-c* beim Aufruf des *gcc* ?

(1 Punkt)

Lösung:

---

<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

### Aufgaben 6:

6 Punkte

Wie muß der Aufruf über einen Softwareinterrupt aussehen, damit *getchar* durch den abgebildeten *SWIHandler* ausgeführt wird?

(2 Punkte)

Lösung:

#### *SWIHandler:*

```

stmfd    sp!, {lr}
ldr      ip,[r14, #-4]
bic      ip, ip, #0xff000000
ldr      lr, =SWIJumpTable
ldr      ip, [lr, ip]
mov      lr, pc
mov      pc, ip
ldmfd    sp!, {pc}^

```

#### *SWIJumpTable:*

```

.word    init_ser
.word    putchar
.word    getchar

```

Welches ist die nächste Adresse im Programmcounter nach Ausführen des folgenden Befehls?

(2 Punkte)

```
0x2000010    SWI        #0x1
```

Lösung:

Welcher Befehl steht an der angegebenen Adresse?

(2 Punkte)

Lösung:

<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

### Aufgabe 7:

6 Punkte

Ein Timer soll so initialisiert werden, damit er unten abgebildetes Signal erzeugt. Das System wird mit 5MHz getaktet. Die Initialisierungsroutine ist bis auf die Werte für die Compare-Register fertig.  
 Berechnen Sie die fehlenden Werte und tragen diese ins Listing ein. Dokumentieren Sie Ihren Lösungsweg.

```

// Timer3 initialisieren
void Timer3_init( void )
{
    StructTC* timerbase3 = TCB3_BASE; // Basisadresse TC Block 1
    StructPIO* piobaseA = PIOA_BASE; // Basisadresse PIO B

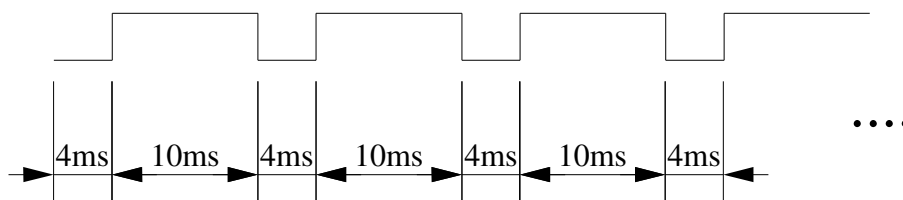
    timerbase3->TC_CCR = TC_CLKDIS; // Disable Clock

    // Initialize the mode of the timer 3
    timerbase3->TC_CMR =
        TC_ACPC_CLEAR_OUTPUT | //ACPC : Register C clear TIOA
        TC_ACPA_SET_OUTPUT | //ACPA : Register A set TIOA
        TC_WAVE | //WAVE : Waveform mode
        TC_CPCTRIG | //CPCTRIG : Register C compare trigger enable
        TC_CLKS_MCK2; //TCCLKS : MCKI / 2

    // Initialize the counter:
    timerbase3->TC_RA = _____; //__
    timerbase3->TC_RC = _____; //__

    // Start the timer :
    timerbase3->TC_CCR = TC_CLKEN ;
    timerbase3->TC_CCR = TC_SWTRG ;
}
  
```

(2 Punkte)



Lösungsweg:

---



---



---

(4 Punkte)



<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

### Aufgabe 9:

**4 Punkte**

Im Folgenden handele es sich um eine Routine, welche auch aus einem Software-Interrupt-Handler aufgerufen wird.

Ergänzen Sie die erste und letzte Zeile

*getchar:*

```

_____@_____ (2 Punkte)
1:  ldr    r2, =USART0_BASE
    ldr    r1, [r2, #US_CSR]
    ands  r1, r1, #US_RXRDY
    beq   1b
    ldr    r0, [r2, #US_RHR]
_____@_____ (2 Punkte)

```

<b>FHD</b> Dr. Th. Horsch Dr. K. Frank Fachbereich I	<b>Mikroprozessortechnik 2</b> Klausur vom 23.01.2004	Name: _____ Matrikel-Nr.: _____
---	--	------------------------------------

### Aufgabe 10:

**6 Punkte**

Zur initialisierung einer seriellen Schnittstelle sollen die ab der Adresse L1 abgelegten hexadezimalen Werte an die ebenfalls dort abgelegten Adressen in einer Schleife abgelegt werden.

Ergänzen Sie die entsprechende Routine.

```
#include "../h/pmc.inc"
#include "../h/pio.inc"
#include "../h/usart.inc"
```

```
DEFAULT_BAUD = 38400
CLOCK_SPEED = 25000000
US_BAUD = CLOCK_SPEED / (16*(DEFAULT_BAUD))
```

@ Funktion

```
.text
.align 2
.global      init_ser
.type        init_ser,function
init_ser:
    stmfd    sp!, {r0-r3, lr}           @ Register retten
    adr      r0, L1
    adr      r1, L1_end
init_ser_loop:
    ldmia    r0!, {r2-r3}
```

---



---



---



---

```
ldmfd      sp!, {r0-r3, pc}           @ Rücksprung
```

L1:

```
.word PMC_BASE+PMC_PCER, 0x4
.word PIOA_BASE+PIO_PDR, 0x18000
.word USART0_BASE+US_CR, 0xa0
.word USART0_BASE+US_MR, 0x8c0
.word USART0_BASE+US_BRGR, 0x29
.word USART0_BASE+US_CR, 0x50
```

L1\_end: