HOCHSCHULE ESSLINGEN FAKULTÄT INFORMATIONSTECHNIK

Sample Exam H		Blatt Nr.:	**1 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript, Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

Insert your name here: Total: 100 Points

Given name (Vorname):	Last name (Nachname):	Student ID (Matrikelnummer):
Solution hints (no guarantee for correctness)		

Please use the free space on these sheets for your solution. Solutions may be in English or German. If space is not sufficient, please use the backside or additional sheets.

Bitte tragen Sie Ihre Lösungen in Deutsch oder Englisch in die Aufgabenblätter ein. Wenn der vorgesehene Platz nicht ausreicht, verwenden Sie bitte die Rückseite bzw. Zusatzblätter. Kennzeichnen Sie, auf welche Fragen sich die Zusatzblätter beziehen.

Viel Erfolg - Good luck!

Problem 1: Miscellaneous (Σ 25 points)

1.1 (5 points)

Which problems/bugs does the following C function have?

```
int * enterPinNumber (void)
{     char pin[8] = "12345678";
      int  pinNum = 0;
      printf("Enter your PIN: ");
      gets(pin);                  //Enter PIN from keyboard as string
      pinNum = atoi(pin);         //Convert string to number
      return &pinNum;
}
```

Initalization string too long, 7 characters only or declare char pin[9] required.

gets() may cause a buffer overflow, if the user enters 8 or more characters.

Function returns a pointer to a local variable, which is no longer available, when the function returns to the caller.

Note: int pinNum only allows values in the range -32768 … +32767. If the user enters an out-of-range value or non-numeric characters, this will be handled by atoi().

Sample Exam H		Blatt Nr.:	**2 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript,** **Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

1.2 (4 points)

Describe the structure of a "Stack Frame" used by HCS12's C compiler when calling subroutines:

Local Variables <<--- TOS
Save Registers (if any)
Return Address
Function Parameters

1.3 (10 points)

A student writes the following C program. The program shall toggle the LEDs on the Dragon12 board approximately once every 5 seconds via an ECT timer interrupt service routine.

The code compiles and runs, but the LEDs do not blink as expected. Correct **all** bugs and add instructions (if required), to make the program work as specified.

```
#include <mc9s12dp256.h>

#define COUNT 10                    <<<1876  correct value for timer

void interrupt 8 timerISR(void)     <<< Interrupt vector table entry
{   static int i = 0;

    if (++i >= 500)                 // Toggle every 500th interrupt event
    {   PORTB = ~ PORTB;
        i = 0;
    }

    TC0 = TC0 + COUNT;
    TFLG1 = TFLG1 | 0x01;           <<< Reset interrupt flag
}

void main(void)
{   EnableInterrupts;               <<< Global interrupt enable

    DDRB  = 0xFF;                   // Configure LED ports
    DDRJ  = DDRJ | 0x02;
    PORTB = 0x00;
    PTJ   = PTJ  & 0xFC;
```

Sample Exam H		Blatt Nr.:	**3 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript,** **Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

```
    TSCR1 = 0x80;                  // Configure timer channel 0
    TSCR2 = 0x07;

    TIOS  = 0x01;

    TCTL1 = 0x00;
    TCTL2 = 0x00;

    TC0 = TCNT + COUNT;
    TIE = 0x01;                    <<<  Enable timer interrupt

    for (;;);
}
```

1.4 (6 points)

What is the contents of variables `S1,S2,…, S6`, when the following C code has been executed:

```
char a = 0x7E, b = 0x81, c=0x7F, S1,S2,S3,S4,S5,S6;
     S1 = !a;
     S2 = a | b;
     S3 = a && b;
     S4 = a ^ c;
     S5 = ~a;
     S6 = a > b ? 1 : 0;
```

```
S1 =      0 = FALSE

S2 =      -1 = 0xFF

S3 =      1 = TRUE

S4 =      1 = 0x01

S5 =      -127 = 0x81

S6 =      1 = 0x01
```

Sample Exam H		Blatt Nr.:	**4 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript, Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

Problem 2: Addressing Modes (Σ 25 points)

2.1 (5 points)

Specify the addressing mode for all operands of the following HCS12 instructions.
One (or more) instructions may contain bugs. Mark the instruction(s) and explain the problem.

Instruction	1st operand	2nd operand
`LDD #var1`	`Implicit register address for D`	`Immediate`
`LDD 6, -Y`	`Implicit register address for D`	`Register-indirect address wit pre decrement`
`STD #var3`	`Implicit register address for D`	`Invalid address: Cannot store into constant`
`LDD $C000`	`Implicit register address for D`	`Direct address $C000`
`MOVB 0,Y,var3`	`Register-indirect address with index/offset`	`Direct address for var3`

2.2

A HCS12 program defines some global variables and constants:

```
.const:     SECTION
            ORG     $D010

c1:         DC.W    $0102, $0304, $0506, $0708
c2:         DC.B    $09, $10, $11, $12, $13, $14, $15, $16
c3:         DC.L    $76543210, $FEDCBA98

.data:      SECTION
            ORG     $2080
p:          DS.W    1
v:          DS.W    1
```

For each row in the following table specify the contents of CPU registers D, X and Y and of variables p and v, after the instructions in the left field of the row have been executed. Fields, where values do not change, may be left empty. Mark unknown values as “???” if necessary.

HOCHSCHULE ESSLINGEN FAKULTÄT INFORMATIONSTECHNIK

Sample Exam H		Blatt Nr.:	**5 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript,** **Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

Note: All values in Hex (20 points)

HCS12 Instructions	D	X	Y	p	v
`Initial values`	0	0	0	0	0

`LDD c1`
`LDX c2`
`LDY c3`
`MOVW #$2456, v`
`MOVW #v, p`
	$0102	$0910	$7654	&v=
				$2082
					$2456

`PSHX`
`PSHD`
`PSHY`
`PULB`
`PULA`
`PULX`
`LDY 2,SP+`
	$5476	$0102	$0910	=	=

`LDX #c1`
`LDY #c2`
`LDD 2, X`
	$0304	$D010	$D018	=	=

`LDAA 3,+X` `LDAB 1, Y-`	$0409	$D013	$D017	=	=
`LDX #p` `LDD [0,X]`	$2456	$2080	=	=	=

`LDD #$7080`
`TFR A, X`
`TFR B, Y`
`BMI L0`
`LDD c3`
`L0: NOP`
	$7654	$0070	$FF80	=	=

`LDD #$2080`
`LDX $D010`
`LDY c3+2`
	$2080	$0102	$3210	=	=

`MOVW #v, p`
`LDD #$ABCD`
`STD p+2`
	$ABCD	=	=	=	$ABCD

Sample Exam H		Blatt Nr.:	**6 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript,** **Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

Problem 3: Code Analysis (Σ 35 points)

The following C program performs a mathematical operation with two vectors:

```
char vectorA[8] = { 1, 2, 3, 4, 5, 6, 7, 8 };
char vectorB[8] = { 8, 7, 6, 5, 4, 3, 2, 1 };
int result;

int function(char *a, char *b, int n);

void main(void)
{   . . .
    result = function(vectorA, vectorB, 3);
    . . .
}
```

function() is a HCS12 assembler subroutine:

```
function:       STD   6,-SP          ; Line 1   Comments see
                LDD   #0             ; Line 2   sample Codewarrior
                STD   4, SP          ; Line 3   project
                STD   2, SP          ; Line 4
                BRA   m2             ; Line 5

m1:             LDD   10,SP          ; Line 6
                ADDD  2, SP          ; Line 7
                TFR   D, X           ; Line 8
                LDAB  0, X           ; Line 9
                SEX   B, D           ; Line 10
                PSHD                 ; Line 11

                LDD   10,SP          ; Line 12
                ADDD  4, SP          ; Line 13
                TFR   D, X           ; Line 14
                LDAB  0, X           ; Line 15
                SEX   B, Y           ; Line 16
                PULD                 ; Line 17

                EMUL                 ; Line 18
                ADDD  4, SP          ; Line 19
                STD   4, SP          ; Line 20

                LDX   2, SP          ; Line 21
                INX                  ; Line 22
                STX   2, SP          ; Line 23

m2:             LDD   2, SP          ; Line 24
                CPD   0, SP          ; Line 25
                BLT   m1             ; Line 26

                LDD   4, SP          ; Line 27
                LEAS  6, SP          ; Line 28
                RTS                  ; Line 29
```

Sample Exam H		Blatt Nr.:	**7 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript,** **Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

3.1
Into which HCS12 assembler instructions does the C compiler translate the subroutine call

```
result = function(vectorA, vectorB, 3)
```

(6 points)

```
LDD #vectorA     ; Params on stack
PSHD
LDD #vectorB
PSHD
LDD #3           ; Last param in register D
JSR function     ; Call subroutine
STD result       ; Store result
LEAS 4,SP        ; Remove params from stack
```

3.2
Specify the state of the stack, when the CPU reaches line 5. Use the table on the next page.

3.3
What is the purpose of the following lines of code?

(6 points)

Lines 1…4: Allocate and initialize local variables on the stack

Line 27: Return function result to caller

Line 28: Release local variables

3.4
How often does the program execute lines 24…25, when the function is called as in 3.1?

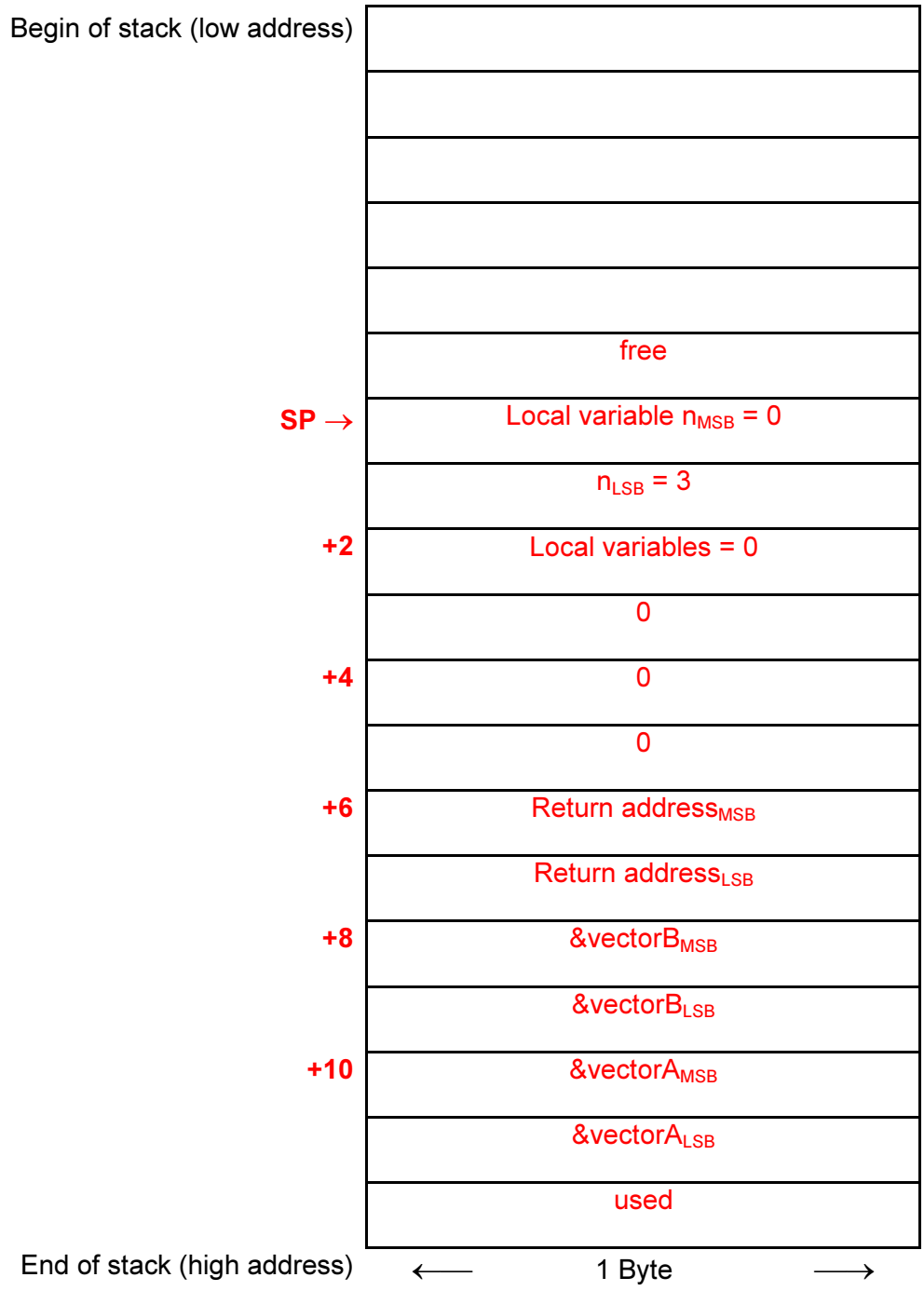
(2 points)

Executes n+1 = 4 times

Sample Exam H		Blatt Nr.:	**8 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript,** **Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

Stack status for question 3.2:

(8 points)



Sample Exam H		Blatt Nr.:	**9 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript, Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

3.5

Lines 21…26 implement a well-known C/C++ programming construct, which is used to control program flow. Which one?

(2 points)

Implements a for()-loop with a counter variable (local variable addressed via 2,SP)

3.6

Which value is in register D, when the program reaches line 11 for the first time?

(2 points)

D = 1 = 0x0001

Which value is in register Y, when the program reaches line 17 for the first time?

(2 points)

Y = 8 = 0x0008

Which value is in register D, when the program reaches line 20 for the first time?

(2 points)

D = 8 = 0x0008

Which value does the function return and what is the purpose of this function?

(5 points)

Return value: $a_1 b_1 + \ldots + a_n b_n = 40 = 0x28$

Function purpose: Compute the scalar product of the two vectors a and b for the first n elements

Sample Exam H		Blatt Nr.:	**10 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript,** **Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

Problem 4: String to Integer Number Conversion (Σ 15 points)

You shall write a subroutine

```
unsigned char asciiToNum(char *string)
```

which converts a string into a number. The ASCII zero terminated string contains two characters, which are ASCII-coded decimal digits. The subroutine shall convert this string into the respective unsigned 8 bit number, e.g. `string = "37" → num = 37 = 0x25`.

If the string contains non-numeric characters, i.e. other characters than '0','1',…,'9', the function shall return 0 = 0x00.

The subroutine shall be written in HCS12 assembler or C (without using any standard library functions) in such a way, that it can be called from a C program.

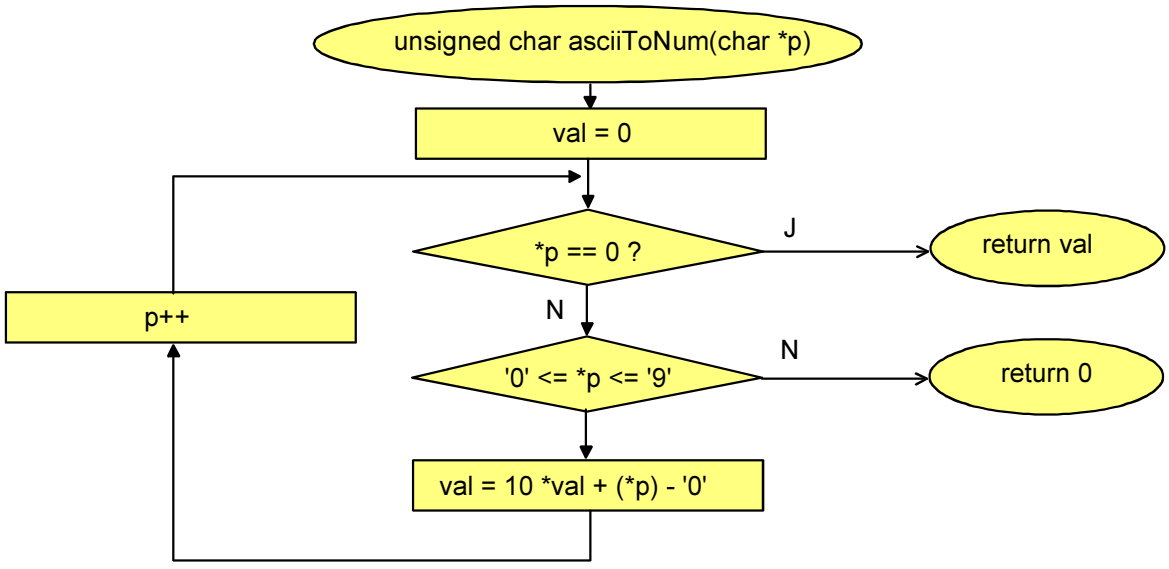
ASCII table:

ASCII character	*ASCII code in hex*
Non-numeric characters	0x00 … 0x2F
'0'	0x30
'1'	0x31
'2'	0x32
. . .	. . .
'9'	0x39
Non-numeric characters	0x3A … 0xFF

4.1

Describe your subroutine design with a program flow chart:

(7 points)



REMOVE

Sample Exam H		Blatt Nr.:	**11 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript,** **Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

4.2
Write the subroutine in HCS12 assembler or in C without any C standard library functions. The subroutine shall be placed into a separate module (=file). Don't forget to add comments to your code!

(4 points for a solution in C OR 8 points for a solution in HCS12 assembler)

```
// Solution in C
// Note: These solutions work for strings of arbitrary length. The
number is computed modulo 256

unsigned char asciiToNumC(char *p)
{   unsigned char val = 0;

    while (*p != 0)                          // while not end of string
    {      if ((*p < '0') || (*p > '9'))     // check for non-numeric
      {   return 0;
      } else
      {   val = val * 10 + (*p - '0');       // convert char to number
      }
      p++;
    }
    return val;
}
```

Sample Exam H		Blatt Nr.:	**12 von 12**
Studiengang:	**Softwaretechnik SWB** **Technische Informatik TIB** **Ingenieurpädagogik IEP**	Semester:	IT4
Prüfungsfach:	**Computerarchitektur**	Fachnummer:	**1054003**
Hilfsmittel:	**Vorlesungs- und Labormanuskript,** **Fachliteratur, Taschenrechner** **Lecture manuscript, books, pocket calculator**	Dauer: Dozent: :	**90 min** Zimmermann

```
// Solution in HCS12 assembler

            XDEF asciiToNum       ; Export function

.init:      SECTION

; unsigned char asciiToNum(char *p)


asciiToNum:PSHX       ; Save register (optional)

            TFR D, X    ; Move pointer --> reg X

            CLRB         ;Initialize return val = 0 in reg B

loop:       LDAA 0,X    ; Load character into reg A

            TSTA        ; Check for end of string
            BEQ  return

            CMPA #'0'   ; Check, if character is numeric
            BLO  nonnum
            CMPA #'9'
            BHI  nonnum

            LDAA #10    ; val = val * 10 + *p - '0'
            MUL

            ADDB 0,X
            SUBB #'0'

            INX         ; p++
            BRA loop

nonnum:     CLRB        ; Non numeric character, so return 0

return:     PULX        ; Restore register (optional)

            RTS         ; Return value already in B
```