

# Embedded Systems - Exercises

Ingo Sander

October 5, 2004

## Contents

<b>1</b>	<b>Instruction Set and Architecture</b>	<b>2</b>
<b>2</b>	<b>CPUs</b>	<b>4</b>
<b>3</b>	<b>Embedded Computing Platform</b>	<b>7</b>
<b>4</b>	<b>Program Design and Analysis</b>	<b>8</b>
<b>5</b>	<b>Processes and Operating Systems</b>	<b>10</b>

The main base for this collection of exercises are the exams from the previous year. However they are not fully representative, since in this years course version more focus has been put onto chapter 5 and 6. Thus please study also the exercises in the Wolf book, mainly Q5-3 to Q5-16, Q6-13 to Q6-15 and Q6-17 to Q6-26.

# 1 Instruction Set and Architecture

1. Describe the *von Neumann* and the *Harvard architectures* and point out differences and similarities. What are the strengths and weaknesses of each architecture for embedded systems? (*Exam 2003-10-24*)
2. Write the equivalent of the following C-code in ARM assembly. Describe your data structures and use comments to describe your design. (*Exam 2003-10-24*)

```
int i , N=10;

sum = 0;

for ( i = 0; i < N; i++)
{
    sum = sum + array[ i ];
}
```

3. Write the ARM assembly code needed for the following function. Follow the *ARM procedure call standard* (APCS) so that it can be called from a C program. Motivate your answer. (*Exam 2003-10-24*)

```
foo(int a , b)
{
    int x,y,z; /* allocated in r5 , r6 and r7 */

    y = a + b;
    z = a - b;
    x = y * z;

    return x;
}
```

4. Explain the ARM Procedure Call Standard (APCS) and why it is needed. (*Exam 2004-01-14*)
5. Write a, well commented, subroutine in ARM assembly which performs a multiplication of two positive 32-bit numbers and whose result also is contained in a 32-bit number. Use repetitive addition (not a combination of shifts and addition). The subroutine has two in-parameters and returns a result. (*Exam 2004-01-14*)

6. Modify your subroutine from 5 so that the time for a multiplication depends on the smallest factor. Also analyze your code and determine the execution time in terms of the number of instructions. (*Exam 2004-01-14*)

## 2 CPUs

1. Explain busy-wait I/O vs. interrupt driven I/O. (*Exam 2003-10-24*)
2. What are the main differences between an interrupt routine and a function (subroutine)? (*Exam 2003-10-24*)
3. What is an exception?
4. What is the advantage of using the Fast Interrupt mode (FIQ) in an embedded system design, over the normal interrupt (IRQ) mode? (*Exam 2003-10-24*)
5. Explain the steps taken in an ARM processor when responding to an interrupt. (*Exam 2003-10-24*)
6. Explain the concept of Direct Memory Access, DMA. Your explanation should contain what it is, the advantage compared to other I/O methods and the basic steps in performing a DMA operation. (*Exam 2003-10-24*)
7. Derive a formula for the average memory access time in an embedded computer system where a single level of unified cache memory is used. Explain all terms used. (*Exam 2004-01-14*)
8. Consider a direct-mapped cache which is 4 kbyte large and with a block size of 8 bytes. Explain how the address mapping works and how the processor can find whether a particular memory access can be satisfied in the cache or in the main memory. (*Exam 2004-01-14*)
9. What is the advantage of multi-level cache systems? (*Exam 2004-01-14*)
10. What are the tradeoffs in doubling the cache block size? (*Exam 2004-01-14*)

11. Eight switches and eight LEDs are connected via memory-mapped I/O with an ARM CPU (Figure 1). The address for the switches is \$0x70001000 and for the LEDs \$0x70010004. An interrupt is generated, whenever a switch is activated or deactivated.

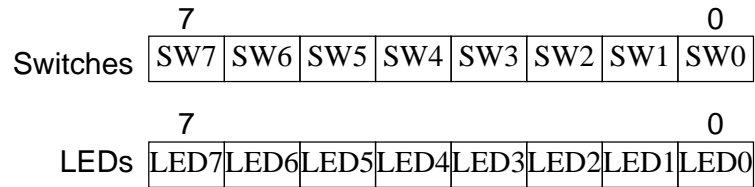


Figure 1: Switches and LEDs

The main program is a C-program that makes some heavy calculation.

Write an interrupt routine that calculates the sum of two unsigned binary numbers, where the first number is given by the bit pattern of the first four switches (SW7-SW4) and the second number by the bit pattern of the last four switches (SW3-SW0). The sum shall then be shown on the LEDs.

12. Describe the principle of cache memories. From the description, it should be clear what problem it solves, a brief description on how it works, the most important design considerations and why it works. (Exam 2003-10-24)

13. An byte-addressable embedded computer system has a main memory consisting of 16384 32-bit words. It also has a small data cache capable of holding eight 32-bit words, where each cache block consists of two words. When a given program is executed the processor reads data from the following sequence of hex addresses:

8000, 8008, 801C, 6E00, 6E0C, 6E08, 8010, 8004, 6E08, 6E0C, 7520, 7524.

This pattern is repeated four times.

- (a) Consider a direct-mapped cache.
  - i. How many bits are used for the tag, set (block), and word (offset) fields for the representation of a memory address?
  - ii. Show the contents of the cache at the end of the execution. Assume the cache is empty at the start of the program.
  - iii. Compute the hit-rate for this example. Assume that the cache is initially empty.
- (b) Repeat 13a for a two-way-set associative-mapped cache that uses a least-recent-used replacement algorithm.

### **3 Embedded Computing Platform**

1. What are the main differences between SRAM and DRAM? (*Exam 2003-10-24*)
2. What is the meaning of a burst mode transaction on a bus and what is the advantage? (*Exam 2003-10-24*)
3. What is the purpose of a bus bridge? (*Exam 2003-10-24*)
4. What is the meaning of a wait state and what is it good for? (*Exam 2003-10-24*)

## 4 Program Design and Analysis

1. A small ARM program is divided into three source files.

- Source-File 1

```
GLOBAL A, D

AREA inst_A , CODE
ENTRY
EXTERN num1
A   ADR r1 , num1
    LDR r2 , [ r1 ]
    EXTERN num3
    ADR r1 , num3
    LDR r3 , [ r1 ]
    EXTERN X
    BL  X
D   B   D

END
```

- Source-File 2

```
GLOBAL X, Y, Z

AREA inst_B , CODE
X   CMP r2 , r3
Y   BLE Z
    SUB r2 , r2 , #1
    B   X
Z   MOV pc , lr
```

- Source-File 3

```
GLOBAL num1, num2, num3

AREA num_A, DATA
num1 DCD 3 , 400 , 300
num2 DCB 6 , 9 , 12 , 1
num3 DCD 1
```

(a) Explain briefly the two passes normally done in assembler

- (b) Determine for each source file the symbol table that is generated by the assembler.
  - (c) Does the linker have sufficient information to generate an executable program. Motivate!
2. How does a compiler allocate memory for a two-dimensional matrix? Show how the correct address to the element with indices  $i$  and  $j$  is calculated for a matrix with 64-bit double precision elements. (*Exam 2004-01-14*)
  3. What is loop unrolling and what is the advantage/disadvantage? (*Exam 2004-01-14*)
  4. What do you need to know to be able to calculate the execution time of a program without executing it? (*Exam 2004-01-14*)

## 5 Processes and Operating Systems

1. Which of the following systems of periodic tasks are schedulable by (1) a rate monotonic scheduling<sup>1</sup> or (2) by earliest deadline first? Motivate your answer! The characteristics of a process  $P$  are described as  $P = (\textit{computation time}, \textit{period}, \textit{deadline})$ .

(a)  $P_1 = (4, 8, 8); P_2 = (4, 12, 12); P_3 = (4, 20, 20)$

(b)  $P_1 = (4, 8, 8); P_2 = (2, 10, 10); P_3 = (3, 12, 12)$

(c)  $P_1 = (3, 8, 8); P_2 = (3, 9, 9); P_3 = (3, 15, 15)$

2. Two processes with different priorities in a preemptive multitasking system  $P_1$  and  $P_2$  want to use an output device  $OD$ , which is a shared resource. A part of the code for both processes is shown below.

```
    ; Code for Process 1
    ;
    ...
    ; use of resource OD
    ADR r1 , OD
    STR r2 , [ r1 ]
    ...

    ; Code for Process 2
    ;
    ...
    ; use of resource OD
    ADR r1 , OD
    STR r2 , [ r1 ]
    ...
```

Can you add protection mechanisms, so that there is not any longer a risk for collision.

---

<sup>1</sup>Scheduling with static priorities is feasible, if  $U \leq n(2^{1/n} - 1)$