RUHR
UNIVERSITÄT
BOCHUM

RUB

Name: **Tim Barsch**
Student ID: **108019210718**
Assignment Number: **2**
Date: November 3, 2021
Course: **ARM Processors for Embedded Cryptography**

Text Assignments

1.1 The Cortex-M4 processor features a load-store architecture. Research on the accumulator architecture and name the differences.

A load-store or register based architecture has one or more general purpose register, which can be used to load and store results. An accumulator architecture has only one general purpose register, the accumulator. "The main advantage/s of "more that one general purpose register" is that the compiler doesn't have to "spill" as many temporary values onto the stack; and it's easier for the CPU to do more independent instruction in parallel." [1]

1.2 Explain why it is helpful to have the two subtract instructions SUB and RSB. How could RSB be realized if only SUB was available?

The usually SUB instruction is the intuitively algebraic subtraction of two values. But the ARM specific architecture has the property that the second operand has always a wider range than the first operand. To use subtract from operand two the value of operand one, the reverse subtraction - `RSB` - can be used. Without the RSB instruction, each subtraction which use the wide of operand two has to be stored into a register first.

1.3 Why are instructions not automatically updating the status register flags? Give an example for a situation where the automatic behavior would cause a problem.

The status register flags are not automatically updated at each instruction to prevent misbehaving of the control flow. In Listing 1 is a control flow shown without updating the status flag after instruction two. In Listing 2 the opposite. The compare instruction at line one will the zero flag be set to one if R0 is equal R1 in both examples.

```
1 CMP R0, R1
2 SUB R0, =#42
```

Listing 1: status register flag not updated

```
1 CMP R0, R1
2 SUBS R0, =#42
```

Listing 2: status register flag updated

In Listing 1, the second instruction will not change the zero flag, it will stay at one. The opposite happens in the example of Listing 2. Adopted the values in R0 and R1 are equal, so will the zero flag set to one. But the second instruction will change the flag to zero (except the case that R0 stores 42). So the resulting zero status flag is different if we update the status flag after each instruction.

1.4 Why is one meaning of zero flag Z=1 that both operands were equal? How is the zero flag updated (arithmetically) when comparing two registers?

The compare instruction is evaluating the zero flag as a SUB instruction. If the subtraction of the second operand from the value in the first operand zero, the zero flag is set to one. Another obvious interpretation is, that both values are equal, otherwise the subtraction wouldn't result zero.

1.5 Explain the IT instruction. Why is the maximum number of conditionally executed instructions after an IT 4?

The `IT` instruction is a Thumb2 specific conditional jump without empties the pipeline. It is a "if-then" condition and can be used to control the flow of execution. There are just four conditionally executed instructions after an IT instruction, because these instructions have to be encode. The encoding is done with a four bit execution state and leads a limit to the conditionally executed instructions.

References

[1] "Brendan", Feb. 27, 2017, softwareengineering.stackexchange.com/questions/343036/the-difference-between-accumulator-based-and-register-based-cpu-architecture

[2] ARM Architecture Reference Manual - Thumb-2 Supplement, 2005 ARM Limited., https://class.ece.iastate.edu/cpre288/resources/docs/Thumb-2SupplementReferenceManual.pdf#E5.CIHEAEHE