

Questions and tasks in Lecture 4

Task 4-1: Write a program that counts three registers up until they reach a target value. Use the assembler functions BYTE1, BYTE2 and BYTE3 to derive the three bytes from the constant:

```
; Deriving single bytes from a constant  
.equ constant = 1234567 ; = hexadecimal 0x12D687  
.equ b1 = BYTE1(constant) ; Yields 0x87  
.equ b2 = BYTE2(constant) ; Yields 0xD6  
.equ b3 = BYTE3(constant) ; Yields 0x12
```

Use the instruction CPI (ComPare with Immediate) and the Z flag to find out, if the value has been reached. Try if you can derive a formula for the clock cycles and verify with the simulator if your formula is working correct and exact.

(Note: This task is solvable!)

Questions and tasks in Lecture 4 - Continued

Task 4-2: Try to change the code so that the three registers down-count the initial constant value.

Hint: Do not use the instruction DEC (for DECREASE) because it does not affect the overflow flag C (for Carry). Use the instructions SUBI (for SUBtract Immediate) and BRCC (BRanch on Carry flag Clear) for conditional jumps.

Try to derive the formula for the number of clock cycles.

Another hint: The formula to derive the consumed clock cycles for that might get a little complicated, so give up if unsuccessful for longer than two hours. Genius approaches are welcome!

Questions and tasks in Lecture 4 - Continued

Task 4-3: If your crystal (next lecture) would have a frequency of 4 MHz, what would be the constants for

- a) cOuter, cMiddle and cInner for the method first introduced here,**
- b) for the up-counting, and**
- c) for the down-counting.**

Try verification with avr_sim, by inputting the clock frequency and disabling the CLKDIV8 fuse (but use a very fast computer for that, place breakpoints, let avr_sim work over night and disable the operating system's sleep features!)