Exercise Sheet 2

Embedded AES

Obligatory homework this week: Exercise 3. Hand-in by September 26, 23:59.

Exercise 1: MixColumns on ATMega16
Create a new ATMega16 project (solution) with AVR Studio. Take your MixColumns operation from week 1 as the C source of your project. Build the project and report its code size (program memory), RAM consumption (data memory), and the number of clock cycles needed to execute the MixColumns operation (use the simulator for counting cycles).

Exercise 2: Optimized 8-bit MixColumns
In your project in AVR Studio, implement the 8-bit optimized MixColumns operation as explained in the lecture. Build the project and report its code size (program memory), RAM consumption (data memory), and the number of needed clock cycles to execute the MixColumns operation. Compare it to the result of Exercise 1.

Exercise 3: Optimized 8-bit AES
For this exercise, you will create a lightweight implementation of the AES-128 encryption on the ATMega16. When doing this, there are three possible optimization goals:

1. Low RAM consumption,
2. Small code,
3. Low execution time (clock cycles).

Note that these are *distinct* and sometimes *conflicting* goals! For this exercise, **pick one** of these goals and implement the AES encryption with this goal in mind. You are free to choose your own approaches and tricks to optimize your lightweight AES implementation for your chosen goal. Remember to test the correctness of your code, using for example the test vectors below.

Key	`00000000000000000000000000000000`
Plaintext	`f34481ec3cc627bacd5dc3fb08f273e6`
Ciphertext	`0336763e966d92595a567cc9ce537f5e`
Key	`10a58869d74be5a374cf867cfb473859`
Plaintext	`00000000000000000000000000000000`
Ciphertext	`6d251e6944b051e04eaa6fb4dbf78465`

You should hand in the C code, as well as a short report in PDF format containing a brief documentation of your code, which optimization goal you chose, and the implementation choices made. Your report should also include the code size (program memory in AVR Studio), RAM consumption (data memory in AVR Studio), and the number of needed clock cycles (in AVR Studio, use the simulator for counting cycles) for your AES implementation.

Group work is strongly encouraged but please make sure to hand in individually.