Lab 08

R. Ferrero & E. Giusto

Politecnico di Torino

Dipartimento di Automatica e Informatica (DAUIN)

Torino - Italy

Ex. 1

- Write a subroutine `myUADD8` implementing the following instruction:

  ```
  UADD8 <Rd>, <Rn>, <Rm>
  ```

- To check how `UADD8` works, please check Lab06, Ex.1.
- Routine has to fulfill AAPCS standard:
  - Parameters passed using registers: `Rn` = `r0`, `Rm` = `r1`
  - Return Value passed using register: `Rd` = `r0`
  - Registers `r4`-`r11` have to be preserved.

Ex. 2

- Write a subroutine `myUSAD8` implementing the following instruction:

  `USAD8 <Rd>, <Rn>, <Rm>`

- To check how `USAD8` works, please check Lab06, Ex.2.
- Parameters and return value are saved in a memory area `DATA READWRITE`, which address is passed to `myUSAD8` using `r6`
- The subroutine does not have to modify registers.

Ex. 3

- Write two subroutines, `mySMUAD` and `mySMUSD` implementing the following instructions:

  `SMUAD <Rd>, <Rn>, <Rm>`

  `SMUSD <Rd>, <Rn>, <Rm>`

- To check how `SMUAD` and `SMUSD` work, please check Lab06, Ex.3.
- Parameters and return value are passed using the stack.
- The subroutines do not have to modify registers.

Exercise 4

- Write the handler of a Supervisor call.
- The handler needs to check the immediate value of `SVC`.
- If the immediate is equal to 3, the handler implements `SMUAD`; if it is equal to 8, it implements `SMUSD`.
- In both cases, the two parameters are popped from stack and the result is pushed into the stack (as done in exercise 3).

Exercise 5

- Add a C file to a new Keil project.
- The new C file contains `int main(void)`
- In the `Reset_Handler`, branch to the `main`
- In the `main`:
  - define three int variabiles `var1`, `var2`, `sum`
  - initialize `var1`, `var2`
  - call the assembly function developed in exercise 1, passing `var1`, `var2` and storing the result in `sum`

Exercise 6

- Write a C function that computes the square root of the sum of the squares of 2 numbers.
- Suggestion: you can use the `hypot` function in `math.h` library
- In the `Reset_Handler`, call the C function with a branch and link.