Exercise sheet 13 – I/O

Goals:

- Programmed I/O
- Interrupt driven I/O
- DMA

Exercise 13.1: Synchronisation commands

(a) Describe a machine instruction for synchronisation and its interaction with the application, OS (operating system), and bus level. *Hint: You may consider the TAS command and semaphores, learned in the Operating Systems (Betriebssysteme) lecture as well as in this lecture. Use some pseudo-code to describe your ideas.*

Exercise 13.2: Programmed I/O (single transfer) with busy wait (pseudo C code)

Consider a system with the *F-Bus serial interface* (FSI). You want to receive data (characters) from the FSI with the busy wait approach. Compare the lecture for that.

(a) Update the `CA_exercises` repository with `git pull`.

(b) In `CA_exercises/sheet_13_io/io_pc_prog_io_busy_wait/io_pc_prog_io_busy_wait.c` you will find a skeleton file.

(c) Complete the skeleton with pseudo C to read 16 bytes (characters) from the *F-Bus serial interface* (FSI) into the memory buffer.

Exercise 13.3: Programmed I/O (single transfer) with polling (pseudo C code)

Consider a system with the *F-Bus serial interface* (FSI). You want to receive data (characters) from the FSI with the polling approach. Compare the lecture for that.

(a) In `CA_exercises/sheet_13_io/io_pc_prog_io_polling/io_pc_prog_io_polling.c` you will find a skeleton file.

(b) Complete the skeleton with pseudo C to read 16 bytes (characters) from the *F-Bus serial interface* (FSI) into the memory buffer.

Exercise 13.4: Interrupt driven I/O (single transfer) (pseudo C code)

Consider a system with the *F-Bus serial interface* (FSI). You want to receive data (characters) from the FSI with the interrupt control approach. Compare the lecture for that.

(a) In `CA_exercises/sheet_13_io/io_pc_interrupt_io/io_pc_interrupt_io.c` you will find a skeleton file.

(b) Complete the skeleton with pseudo C to read 16 bytes (characters) from the *F-Bus serial interface* (FSI) into the memory buffer.

Exercise 13.5: DMA programming (pseudo C code)

Consider a system with the *F-Bus DMA disk* (FDD). You want to write data (some words) from the memory with the DMA approach to the disk. Compare the lecture for that.

(a) In
`CA_exercises/sheet_13_io/io_pc_dma/io_pc_dma.c`
you will find a skeleton file.

(b) Complete the skeleton with pseudo C to write 16 words (4 bytes per word) from the memory to the *F-Bus DMA disk* (FDD).

- Source (memory) starting address: 0x400000
- Target (disk) starting address: 0x4711

Hint: Source, destination, how much, GO!

Exercise 13.6: DMA programming (coding)

Please use the Arduino MKR WiFi 1010 for this exercise, as the Arduino Mega 2560 does not support DMA. This exercise is based on the simple DMA example of Adafruit. Take the provided Arduino skeleton sketch to copy data using *direct memory access (DMA)* . *Hint: Check out Adafruit_ZeroDMA.h.*

(a) Open the provided
`CA_exercises/sheet_13_io/io_prog_dma_io/io_prog_dma_io.ino`
skeleton file and fill in the TODOs.

(b) Flash your sketch on the provided *Arduino MKR Wifi 1010* and open a serial monitor. Please make sure, that you set the right baud rate according to your sketch.