



Exercise sheet 12 – Bus sequences

Goals:

- Program sequence and resulting bus cycles
- Cache influence on bus cycles
- Isolated I/O
- Memory mapped I/O

Exercise 12.1: Program sequence and resulting bus cycles

Consider a 32-bit CPU **without** caches.

Given is following instruction-sequence:

Word 1:	Code for SUB R1, X	; X = X - R1
Word 2:	Address of X	
Word 3:	Code for ADD #4711, R2	; R2 = R2 + 4711
Word 4:	Operand 4711	; Direct operand
Word 5:	Code for MOVE (R0)+, (R1)	; (R1) = (R0)
		; R0 and R1 may contain addresses
		; (R0)+: Post increment of R0

- 32 bit word in each memory line
- Rx stands for data- or address registers

Hint: You may want to draw a table. A spreadsheet software (Excel, LibreOffice) or a paper is your friend.

Nr.	Master	Cycle	Comment	α	β	γ_1	γ_2
1							
...							

(a) State a possible sequence of resulting bus cycles.

Exercise 12.2: Cache influence on bus cycles

Consider a 32-bit CPU **with** caches.

State the changes for *exercise 12.1* resulting in the usage of different caches.

Hint: Addresses of variables (direct addresses) and direct operands are considered as instructions.

Consider following cases:

- Common cache for data and instructions (**perfectly filled**): Which cycles may be obsolete now? *Hint: Mark them with α .*
- Cache for instructions (**perfectly filled**): Which cycles may be obsolete now? *Hint: Mark them with β .*
- Cache for data with *write through* (**perfectly filled**): Which cycles may be obsolete now? *Hint: Mark them with γ_1 .*

- (d) Cache for data with *write back* (**perfectly filled**): Which cycles may be obsolete now? *Hint: Mark them with γ_2 .*

Exercise 12.3: Isolated I/O (coding)

The idea is to continuously toggle the built-in LED of the *Arduino Mega*. For that, the isolated I/O functions should be used.

Hint: You may find the PIN Mapping, the ATMEGA 2560 Datasheet, and the AVR Instruction Set Manual useful.

- (a) On the *Arduino Mega*, the built-in LED is on digital pin 13. On which physical pin is the digital pin 13 mapped and how is it called? Use the PIN Mapping for that.
- (b) The physical pin is part of a register with 8 bits. How is this called and on which position in the 8 bit register is the physical pin mapped? You may use the ATMEGA 2560 Datasheet to find this. *Hint: Look at page 96, section 13.4.5.*
- (c) Which value do you have to write into this register, to enable (switch on)/disable (switch off) the built-in LED?
- (d) Find the register address where the physical pin of the built-in LED is contained. You may again use the ATMEGA 2560 Datasheet to find this. *Hint: Look at page 96, section 13.4.5.: the first HEX value.*
- (e) Find an assembler instruction with which you can directly write to the I/O register. You may use the AVR Instruction Set Manual to find this. *Hint: You may have a look on page 134.*
- (f) Open the provided
`CA_exercises/sheet_12_bus/io_prog_isolated_io/io_prog_isolated_io.ino`
skeleton file and use the collected information about the registers, addresses, values, and assembler instructions to complete the code.
- (g) Flash your sketch on the provided *Arduino Mega*. The built-in LED should toggle now.

Exercise 12.4: Memory mapped I/O (coding)

The idea is to continuously toggle the built-in LED of the *Arduino Mega*. For that memory mapped I/O should be used.

Hint: You may find the PIN Mapping, the ATMEGA 2560 Datasheet, and the AVR Instruction Set Manual useful.

- (a) Find the memory address of the register address where the physical pin of the built-in LED is connected. You may again use the ATMEGA 2560 Datasheet to find this. *Hint: Look at page 96, section 13.4.5.: the second HEX value inside the parenthesis.*
- (b) Find an assembler instruction with which you can write data from a register into the memory (data space/SRAM). You may use the AVR Instruction Set Manual to find this. *Hint: You may have a look on page 179.*
- (c) Open the provided
`CA_exercises/sheet_12_bus/io_prog_memory_mapped_io/io_prog_memory_mapped_io.ino`
skeleton file and use the collected information about the registers, addresses, values, and assembler instructions to complete the code.
- (d) Flash your sketch on the provided *Arduino Mega*. The built-in LED should toggle now.