Answers for Session 1

Group 00:

`Stephan Holljes`
`Yannick Keller`
`Tim Buecher`

Ex.: 1

a) The MUL instruction causes the high number of cycles. It is a multi-cycle instruction.

b) It is an instruction for the simulator so it knows when to stop.

Ex. 2:

a) To get the adress of _A. In general to get the adress of a data semgment. To load a 32 bit value we have to split it in two instructions with 16 bit

   Value after addi : The high 16 bits of the address

   Value after lsoi: The full address (32 bit)

b) In this example the first 16 bits can be omit, because there are 0. So we can omit ADDI (and not LSOI). In gerenal it is posssible they are not 0. SO we need booth instructions to get the full adress.

Ex. 3:

a) if-else

b) max function

c) Because without the NOP the next instruction will also be excecuted

d) We can remove the first NOP with the instruction ADD R24, R0, R22. Because if we jump, R24 will be overwritten. So the result in R24 will be correct

Ex. 4:

a) Multiplication of R21 with R22 and result stored in R23

b)

c) 31 times: (0x7fff << 16)

Ex. 5:

a) Switch case.

   Calculate the offset to the address (base offset to _label1) and then calculate the offset to the right JP

b) Because in _label1 the jumps have a gap of 8 bytes

c) Branch: Conditional

   Jumps: Not conditional

Ex. 6:

a) (attached code to mail: ‘6_for.s’)

b) Number assembly instructions: 18

c) Number clock cycles: 110 (-pf0), 101 (-pf1)

d) Number memory accesses: 21