Lab5 – Løsningsforslag:

Oppgave 1:

```
when "0011" =>
   -- LDI Load Immidiate
   control_RW <= '1';
   control_MD2 <= '1';
```

Oppgave 3:

```
mem_storage(0) <= X"30001400";     -- LDI R4, 1
mem_storage(1) <= X"30002500";     -- LDI R5, 2
mem_storage(2) <= X"50000104";     -- IN  R1, R4
mem_storage(3) <= X"50000205";     -- IN  R2, R5
mem_storage(4) <= X"40002000";     -- JMP 2
```

Oppgave 4

```
-- Initial values
mem_storage(0) <= X"30060100";    -- LDI R1, 50 (current volume)

-- Read key 1
mem_storage(1) <= X"30001400";    -- LDI R4, 1 (Port 1 is key 1)
mem_storage(2) <= X"50000204";    -- IN R2, R4 (Read port "R4" and store the result in R2
                                  -- 0 means that the key has been pressed.
mem_storage(3) <= X"30001400";    -- LDI R4, 1
mem_storage(4) <= X"10800242";    -- Sub R2, R4, R2 ( R2 = R2 - R4)
mem_storage(5) <= X"20030000";    -- BNZ 30 (if the substraction was not zero, then the key
                                  -- was pressed). Jump to 0x30 (48) to handle it
-- Read key 2
mem_storage(6) <= X"30002400";    -- LDI R4, 2 (Port 2 is key 2)
mem_storage(7) <= X"50000304";    -- IN R3, R4 (Read port "R4" and store the result in R2
                                  -- 0 means that the key has been pressed.
mem_storage(8) <= X"30001400";    -- LDI R4, 1
mem_storage(9) <= X"10800343";    -- Sub R3, R4, R2 ( R3 = R3 - R4)
mem_storage(10)<= X"20050000";    -- BNZ 50 (if the substraction was not zero, then the key
                                  -- was pressed). Jump to 0x50 (80) to handle it
-- Jump back to top
mem_storage(11)<= X"40001000";    -- JMP 1   (Repeat until something is pressed)


-- Handle keypress 1
-- First, wait til user lets go of the key
```

```
mem_storage(48) <= X"30001400";    -- LDI R4, 1 (Port 1 is key 1
mem_storage(49) <= X"50000204";    -- IN R2, R4 (Read port "R4" and store the result in R2
                                   -- 1 means that the key has been released.
mem_storage(50) <= X"30001400";    -- LDI R4, 0
mem_storage(51) <= X"10800242";    -- Sub R2, R4, R2 ( R2 = R2 - R4) (status is only updated
                                   -- on arithmetic operations)
mem_storage(52) <= X"20030000";    -- BNZ 30 (if the substraction was not zero, then the key
                                   -- was not released).

-- Don't increase it beyond 7F (max volume)
mem_storage(53)<= X"3007F400";     -- LDI R4, 7F (max volume)
mem_storage(54)<= X"10800941";     -- Sub R9, R4, R1 ( R9 = R1 - R4) (this will be zero if
                                   -- RI is at max volume (7F)
mem_storage(55)<= X"20039000";     -- BNZ 39 (57) (This combination implements Branch if zero)
mem_storage(56)<= X"40001000";     -- JMP 1   (Back to start)

mem_storage(57)<= X"30001400";    -- LDI R4, 1
mem_storage(58)<= X"10700141";    -- ADD R1, R4, R1
mem_storage(59)<= X"4006E000";    -- JMP 6E (110) (Adjust volume)

-- Handle keypress 2
-- First, wait til user lets go of the key
mem_storage(80) <= X"30002400";    -- LDI R4, 1 (Port 2 is key 2)
mem_storage(81) <= X"50000204";    -- IN R2, R4 (Read port "R4" and store the result in R2
                                   -- 1 means that the key has been released.
mem_storage(82) <= X"30001400";    -- LDI R4, 0
mem_storage(83) <= X"10800242";    -- Sub R2, R4, R2 ( R2 = R2 - R4) (status is only updated
                                   -- on arithmetic operations)
mem_storage(84) <= X"20050000";    -- BNZ 50 (if the substraction was not zero, then the key
                                   -- was not released).

-- Don't increase it below 1 (min volume)
mem_storage(85)<= X"30060400";     -- LDI R4, 01 (min volume)
mem_storage(86)<= X"10800941";     -- Sub R9, R4, R1 ( R9 = R1 - R4) (this will be zero if
                                   -- RI is at max volume (7F)
mem_storage(87)<= X"20059000";     -- BNZ 59 (89) (This combination implements Branch if zero)
mem_storage(88)<= X"40001000";     -- JMP 1   (Back to start)



mem_storage(89)<= X"30001400";    -- LDI R4, 1
mem_storage(90)<= X"10800141";    -- SUB R1, R4, R1
mem_storage(91)<= X"4006E000";    -- JMP 6E (110) (Adjust volume)

-- Adjust volume
mem_storage(110)<= X"30003400";    -- LDI R4, 3 (Port 3 is volume)
mem_storage(111)<= X"60000014";    -- OUT R4, R1 (Write value R1 into R4)
mem_storage(112)<= X"40001000";    -- JMP 1 (Back to main)
```