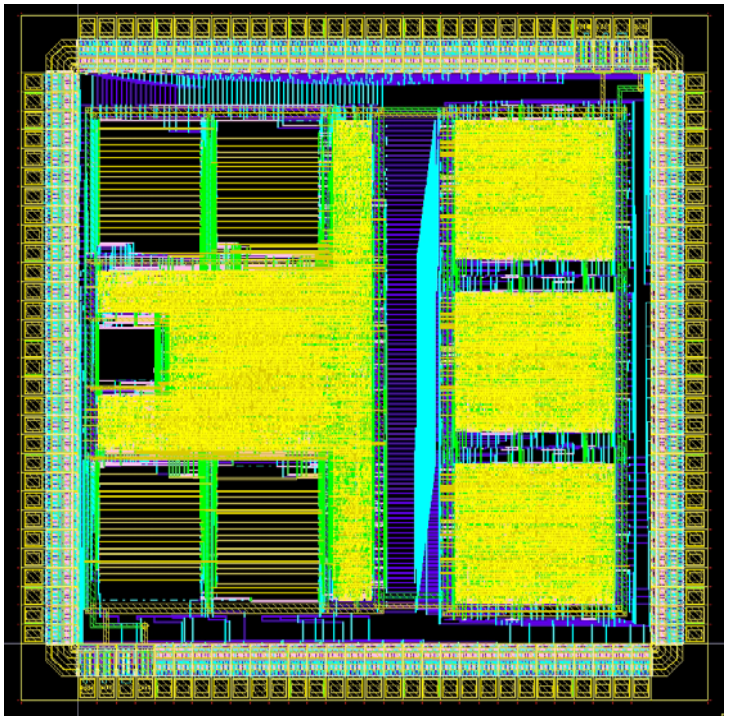
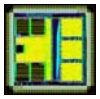
Digital Systems and Microprocessor Design (H7068)



10. Exercises / Laboratory

Daniel Roggen
d.roggen@sussex.ac.uk



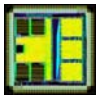
Objective / success criteria

- Prerequisite: lab of week 8
- The objective of this laboratory is to:
  - Understand the memory addressing mode of the educational processor
- Success criteria:
  - You are able to understand and predict the action of the move memory instructions

Tutorial I

- You are going to try simple programs using the mov r,[src] instruction to understand its behavior
- Both immediate and register sources will be tested

Instructions	instruction(15..8)								Instruction(7..0)							
Move	Opcode			$\overline{R}/I$	dd#m	sd#m	dreg		src							
mov r, r	0	0	0	0	0	0	r	r	-	-	-	-	-	-	r	r
mov r, i	0	0	0	1	0	0	r	r	i	i	i	i	i	i	i	i
mov r, [r]	0	0	0	0	0	1	r	r	-	-	-	-	-	-	r	r
mov r, [i]	0	0	0	1	0	1	r	r	i	i	i	i	i	i	i	i
mov [r], r	0	0	0	0	1	0	r	r	-	-	-	-	-	-	r	r
mov [r], i	0	0	0	1	1	0	r	r	i	i	i	i	i	i	i	i



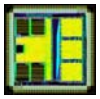
Tutorial I

- Consider the following program:

```
00    mov   ra,[1d]    141D
02    mov   rb,[1e]    151E
04    mov   rc,1f      121F
06    mov   rd,[rc]    0702
```

- In addition, fill in the memory with the following values at these locations:

Addr	Data
1D	33
1E	44
1F	55



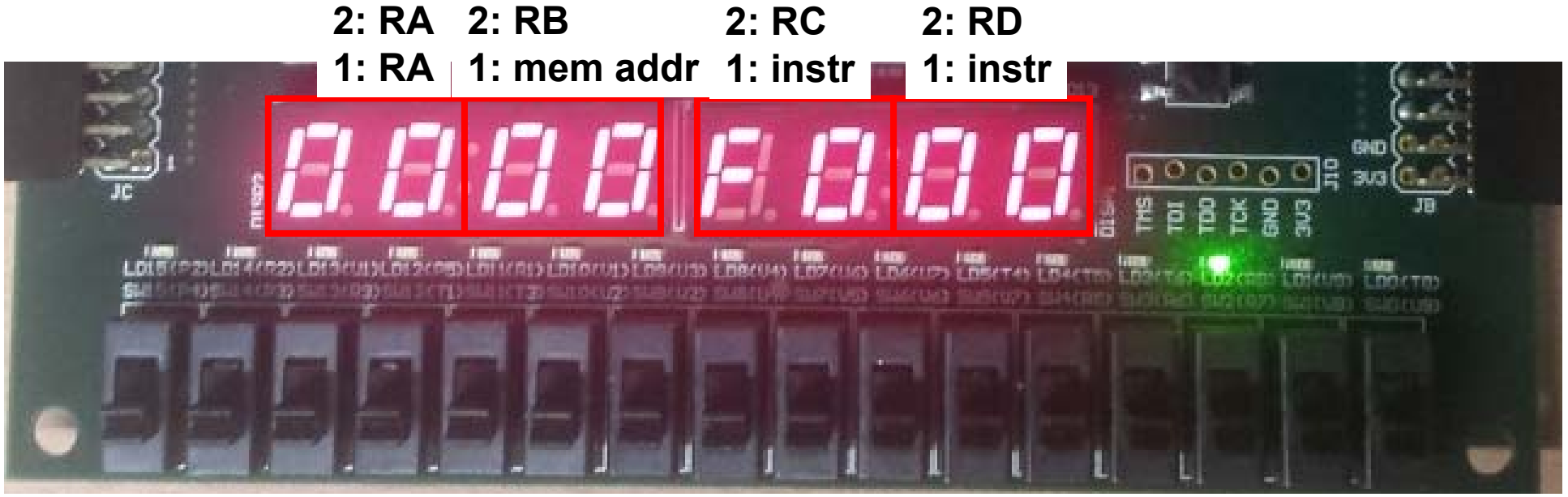
Tutorial I

- Answer the following questions running the instructions in your head and verifying on the board:
- What is the resulting content of registers RA, RB, RC, RD?
- Explain in your words what the mov dst,[src] instruction does
- How is the behaviour different than this program:

```
00    mov   ra,1d      101D
02    mov   rb,1e      111E
04    mov   rc,1f      121F
06    mov   rd,rc      0302
```

Tutorial I

- Remember the 3 clock cycles per instruction: fetch high, fetch low, and execute. The board shows the address placed on the memory bus during each cycle (mem addr)

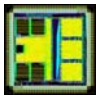


- Explain what is placed on mem_addr at each clock cycle and why?
- How is what is on mem_addr different during the execute cycle when the instruction is a move memory (e.g. at address 0, 2 and 6 of the first program) v.s. when the instruction is not a mov memory (at address 4 of the first program)

Tutorial II

- You are going to try simple programs using the mov [r],src instruction to understand its behavior
- Both immediate and register sources will be tested

Instructions	instruction(15..8)								Instruction(7..0)							
Move	Opcode			$\overline{R}/I$	dd#m	sd#m	dreg		src							
mov r, r	0	0	0	0	0	0	r	r	-	-	-	-	-	-	r	r
mov r, i	0	0	0	1	0	0	r	r	i	i	i	i	i	i	i	i
mov r, [r]	0	0	0	0	0	1	r	r	-	-	-	-	-	-	r	r
mov r, [i]	0	0	0	1	0	1	r	r	i	i	i	i	i	i	i	i
mov [r], r	0	0	0	0	1	0	r	r	-	-	-	-	-	-	r	r
mov [r], i	0	0	0	1	1	0	r	r	i	i	i	i	i	i	i	i



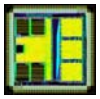
Tutorial II

- Consider the following program:

```
00    mov   ra,1e       101E
02    mov   rb,99       1199
04    mov   [ra],rb     0801
06    add   ra,01       3001
08    mov   [ra],88     1888
```

- In addition, fill in the memory with the following values at these locations:

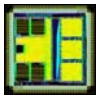
```
Addr        Data
1E          44
1F          55
```



Tutorial II

- Answer the following questions running the instructions in your head and verifying on the board:
- What is the content of memory locations 1E and 1F after executing the program?
- Explain in your words what the mov dst,[src] instruction does
- How is the behaviour different than this program:

```
00    mov   ra,1e      101E
02    mov   rb,99      1199
04    mov   ra,rb      0001
06    add   ra,01      3001
08    mov   ra,88      1088
```

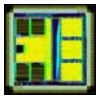


Tutorial II

- In tutorial I and II we used some memory locations to store data. Discuss what makes the difference (if any) between a memory location corresponding to an instruction or to data.
- How does the processor know if a memory location is instruction or data?

- Explain why are there no mov instructions to do at the same time a read from memory and a write to memory (e.g. mov [a],[1Eh])?

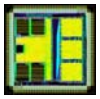
- Discuss what modifications to the processor would be required to have such a mov instruction reading and writing to memory at the same time



Tutorial III

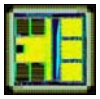
- Consider the following program

```
00    mov   ra,07      1007
02    mov   rb,00      1100
04    mov   [ra],rb    0801
06    out   00         D000
08    add   rb,1       3101
0A    jmp   04         B004
```



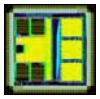
Tutorial III

- What do you expect the out instruction to do on the LEDs?
- Execute the program for ~50 clock cycles: what happens on the LEDs?
- Can you explain the result?



Tutorial III

- Go into the memory edit mode, and check the content of memory locations 4-8; does it correspond to the program you entered?
- With this knowledge, to what kind of program are you dealing with?
- What could be the uses for such kind of programs?



Coursework assignment

- Take time to advance the coursework assignment!

- If you want to create a custom program discuss with me if the complexity is adequate to get a full mark.