Digital Design and Computer Architecture LU

Lab Protocol

Exercise III

Group 2

Markus Bichl, Matr. Nr. 01625745

markus.bichl@student.tuwien.ac.at

Dominik Dallinger, Matr. Nr. 01529357

e1529357@student.tuwien.ac.at

Mathias Hinterndorfer, Matr. Nr. 01612420

e1612420@student.tuwien.ac.at

Vienna, May 24, 2018

<u> </u>	Msgs																					
 ∳ pc_in	14'	14'h0000																	[1	4'h000C	[14'h000	0
■ ◆ pc_out		14'h0000				14	[14'h0	008 (14	rhoooc	(14'h001	0 [14'h0	014 (14h	018 (14)	1001C (14'h00	20 (14	10024	(14'h0	028 [1	4'h000C	14'h001	0 (14'h0014
■ ♦ instr	32'	32 (32'h20010000					[32'h0	0000000		(32'h202	32'h0	0000000	(32)	nAC0 (32'h08	0 (32	1000000	000			32'h20	(32'h0000
mem_out		{21'hXXXXXX}} U U {4'h	X) {32'hXXXXX	00001					0 0 {4'hf	} {32'h0	0000000}				{21'h0				{4 {) \(\) {21'h000
🗓 🔷 address	21'	21'hXXXXXX					21'h0	00000						(:	21'h00	0 21'	100000			1'h000	. [21'h00) (21'h0000
◆ wr	0						_															
📮 💠 byteena		4'hX					4'hF			_												
📺 🔷 wrdata		32'dX					32'd0												13	2'd1	32'd0	
<u> </u>	Msgs																					
a		14'h0000				(14'h(000C)	14'h000	0								(14'h(000C	14'h00	00		
	14'		(14'h0020)14'h002	4 (14'h00					0014 [14	4'h0018)14'h001C	(14'h0020	[14'h00	24 (1	4'h0028					h0014 (1	4'h0018 [14'
■ ♦ pc_in	14' 14'	14'h0000					000C	14'h001	0)(14'h0	0014 [14)14'h001C)32'hACO						000C	14'h00	0 (14)	h0014 (1	
■	14' 14' 32'	14'h0000 14'h0018 (14'h001C	(32'h080)32'h000		28 (14'h)	000C	14'h001	0 (14'h) (32'h)	0000000			32'h080		00000	0	(14'h)	000C	14'h00 32'h20	l0	h00000000	
■- pc_in ■- pc_out ■- instr	14' 14' 32' {21	14'h0000 14'h0018 (14'h001C 32'h000 (32'hACO	(32'h080) 32'h000) 32'h000	00000	28 (14'h)	000C)	14'h001 32'h202 {21'h00	0 (14'h) (32'h)	0000000		32'hAC0	32'h080	[32'h00	00000	0	(14'h)	000C	14'h00 32'h20 {21'h0	l0	h00000000	32'
pc_in pc_out pd_pc_out pd_instr pd_mem_out pd_address pd_rd	14' 14' 32' {21	14'h0000 14'h0018	(32'h080 ({21'h00) 32'h000) 32'h000	00000	28 (14'h)	000C)	14'h001 32'h202 {21'h00	0 (14'h0 (32'h0 ({21'h	0000000		32'hAC0	X 32 h080 X {21 h00	[32'h00	00000	0	(14'h)	000C	14'h00 32'h20 {21'h0	l0	h0000000 'h000000	32'
□ → pc_in □ → pc_out □ → instr □ ← mem_out □ ← address □ ← wr	14' 14' 32' {21 0	14'h0000 14'h0018 (14'h001C 32'h000 (32'hACO (21'h0000001 0 0 (4 21'h000000	(32'h080 ({21'h00) 32'h000) 32'h000	00000	28 (14'h)	000C)	14'h001 32'h202 {21'h00	0 (14'h0 (32'h0 ({21'h	0000000		32'hAC0	X 32 h080 X {21 h00	[32'h00	00000	0	(14'h)	000C	14'h00 32'h20 {21'h0	l0	h0000000 'h000000	32'
pc_in pc_out pd_pc_out pd_instr pd_mem_out pd_address pd_rd	14' 14' 32' {21 0 0 4'hF	14'h0000 14'h0018	(32'h080 ({21'h00) 32'h000) 32'h000	00000	28 (14'h)	000C)	14'h001 32'h202 {21'h00	0 (14'h0 (32'h0 ({21'h	0000000		32'hAC0	X 32 h080 X {21 h00	[32'h00	00000	0	(14'h)	000C	14'h00 32'h20 {21'h0	l0	h0000000 'h000000	32'

Figure 1: Simulation screenshot of the first 2 cycles of the program.

<u> </u>	Msgs																
■ ◆ pc_in	14'	14'h000C		14'h0000												114'h000c	
	14'	14'h000C		14'h0010		14'h0014		14'h0018		14'h001C	14'h0020	14'h0024		14'h0028		14'h000C	
		32'h00000000		32'h202100		32'h000000				32'hAC0100	32'h080000	32'h000000					
■ ← mem_out		{21'h000010}	01 {	{21'h00002	4} 0 0 {	{21 h00000		} {32'h000	00000}		{21 h00000		0} 0 0 {4'h	} {32'h000i	10000}	{21 h00001	
🗖 🔷 address	21'	21'h00001d		21'h000024		21'h000000	1				21 h000002	21'h000000	i			121'h000010	
- rd	0																
→ WF	0																
🛕 🔷 byteena	4'hF	4'hF															
🚊 💠 wrdata	32'd0	32'd1		32'd0							32'd1	32'd0				32'd2	

Figure 2: Simulation screenshot for Listing 1. One complete cycle.

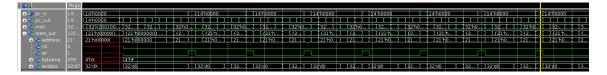


Figure 3: Simulation screenshot for Listing 1. Seven complete cycles.

At the beginning of Figure 1 the value of register 0b00000 (this value is 0) gets added by 1 and written into register 0b00001. After that, in a never ending loop, the value of register 0b00001 gets added by 1 and written into the same register. Additionally, the value of register 0b00001 gets written into the memory register at 0h0010.

Listing 2: Assembler example machine code

```
WIDTH=32;
ADDRESS_RADIX=HEX;
DATA_RADIX=HEX;
DEPTH=11;
CONTENT BEGIN
        00000000 : 20010000;
        00000001 : 00000000;
        00000002 : 00000000;
        00000003 : 20210001;
        00000004 : 00000000;
        00000005 : 00000000;
        00000006 : ac010010;
        00000007 : 08000003;
        00000008 : 00000000;
        00000009 : 00000000;
        0000000a : 00000000;
END;
```